ArcGIS 10.3 Server on Amazon Web Services

Table of Contents

What is ArcGIS Server on Amazon Web Services?

ArcGIS Server on Amazon Web Services allows you to deploy ArcGIS Server on the Amazon Web Services (AWS). ArcGIS Server runs on Amazon's hardware and is administered through web services.

Advantages of deploying your server on AWS include the following:

- **No installation required—**You don't have to install ArcGIS Server yourself. Instead, you use a downloadable utility, ArcGIS Server Cloud Builder on Amazon Web Services, to create your ArcGIS Server site on AWS. Once you create your site, you can immediately connect to it and begin publishing services from ArcMap.
- **Scalable on demand—**You can configure your site so that additional GIS servers are added in response to certain triggers, such as CPU usage. New servers can be created in a matter of minutes, allowing your site to gracefully respond to abrupt spikes in traffic. When you no longer need the instances, you can destroy them and incur no further charges for them.
- **No hardware infrastructure to maintain—**Deploying ArcGIS Server on Amazon Web Services requires no special hardware; you just have to be able to connect to the Internet. After creating your site, you can gain fine-grained management of your server through the AWS Management Console, a web application provided by Amazon. You can log in to your server through Windows Remote Desktop Connection to fine-tune your data, services, and applications.

  Deploying your server in a cloud environment allows you to use as many or as few computing resources as necessary without committing to a long-term purchase of hardware or other IT infrastructure.

About this help

The help section you are reading focuses on how to deploy ArcGIS Server and use enterprise geodatabases on Amazon Web Services. It also includes special topics about maintaining and scaling your deployment on AWS.

If you need general information about ArcGIS Server or enterprise geodatabases, you should visit those respective sections in the ArcGIS Help. The ArcGIS Help describes how to publish services, configure security users and roles, create web applications, build geodatabases, and so on.

Links to the ArcGIS Help are provided throughout this help system where appropriate.

Quick start guide

Do the following to start running ArcGIS Server on Amazon Web Services:

1. Create an Amazon account and make sure it includes Elastic Compute Cloud (EC2) access.
2. Contact Esri Customer Service or your local distributor and provide them with your Amazon account number. Customer Service will then share the ArcGIS for Server Amazon Machine Images (AMIs) with your account.
3. Download and install ArcGIS Server Cloud Builder on Amazon Web Services. You can access this download when logged in to My Esri. It is listed with your other Esri software downloads. Install Cloud Builder on your own desktop machine.
4. Use ArcGIS Server Cloud Builder on Amazon Web Services to create your site on Amazon.
5. In the **Catalog** window in ArcMap, add an ArcGIS Server connection to your new site.
6. Use ArcGIS for Desktop to publish a test map service to your new site. Because you haven't yet registered any data locations with the server, the data will be copied to the server automatically when publishing. That's okay for this exercise as long as you don't put a lot of data in your test map.
7. Connect to the Services Directory of your site to verify that the service published successfully.
8. Optionally, perform more detailed administrative tasks using the AWS Management Console. The console allows you to change the specs of your GIS server machines, detach and attach storage drives, alter security groups, and so forth.

Introduction to Amazon Web Services terminology

Before you begin working with ArcGIS Server on Amazon Web Services, familiarize yourself with some related terms and concepts.

ArcGIS Server Cloud Builder on Amazon Web Services

ArcGIS Server Cloud Builder on Amazon Web Services is a free downloadable app that helps you create an ArcGIS Server site in Amazon Web Services. The app runs on your desktop and requires your Amazon Access Key and Secret Access Key to log in. Once you've logged in to the app, you choose the operating system you want to use for the site and how many machines it can contain, along with some other parameters. You then click a button and the site is created for you. You can also edit existing sites that you've created with the app.

AWS Management Console

The AWS Management Console is Amazon's web-based interface for managing your Amazon Elastic Compute Cloud (EC2) instances. Although you can, and should, perform most of your site administration tasks in Cloud Builder, the AWS Management Console allows you to wield finer control over your ArcGIS site architecture by adding storage volumes, adjusting security groups, and so on.

The AWS Management Console requires you to log in with an account that you've set up with Amazon to use its cloud. The AWS Management Console allows you to view your account information and accumulated charges.

EC2 instance

An EC2 instance is a virtual machine that you create in Amazon Elastic Compute Cloud. You create the instance using an Amazon Machine Image (AMI) that is preconfigured with an operating system and ArcGIS software already installed. When you create a site with ArcGIS Server Cloud Builder on Amazon Web Services, EC2 instances are created behind the scenes.

The following AMIs are available. Each AMI includes a database management system (DBMS) that can be used for your ArcGIS Server site's managed databases.

- **ArcGIS for Server (Windows) with Microsoft SQL Server Express—**Ideal for workgroup geodatabases (but can be used for enterprise geodatabases). Runs on Windows Server 2012.
- **ArcGIS for Server (Windows) with Microsoft SQL Server—**Ideal for enterprise geodatabases. Runs on Windows Server 2012.
- **ArcGIS for Server (Linux) with PostgreSQL—**Runs on Ubuntu Linux.

Once you create the instance, you can log in to it. For example, you can access a Windows instance using a Windows Remote Desktop Connection. If you've worked with ArcGIS before, this is where things start to look familiar. For example, once logged in to an instance, you can use ArcMap to create and manage services.

You can stop and start EC2 instances in much the same way you can shut down and start a computer. Instances can also be terminated when they are no longer needed. Once you terminate an instance, all information about it is lost. For this reason, it's recommended that you store your data on Amazon Elastic Block Store (EBS) volumes attached to your instance and optionally back up your data using Amazon Simple Storage Service (S3).

After creating an instance, you'll add your data, services, and other programs. Once you've configured your instance the way you want, you can use ArcGIS Server Cloud Builder on Amazon Web Services to create a site template reflecting your machine's current state. Use this template to launch a production site whose instances are preconfigured with your data and services.

EBS volume

Amazon Elastic Block Store (EBS) volumes are virtual disk drives that you can attach to your EC2 instance to add more storage. The ArcGIS Server Windows AMIs create an EBS volume and attach it to the instance as the D: drive. The ArcGIS Server Ubutu AMI also creates an EBS volume named /gisdata that is mounted to the instance. The size of this volume is configurable in Cloud Builder, or it is 10 GB if you launch your site from the AWS Management Console.

You can choose to place your data on these preattached volumes, replace them with volumes of a different size, or attach more volumes.

If you create your own EBS volume, you must do the work of formatting the drive and attaching it (for example, you can configure it as drive E: on Windows). You can configure the drive to contain as much or as little space as you need.

EBS volumes are an excellent way to manage your data in the cloud independently from your software. For example, you can easily detach an EBS volume from one instance and attach it to another. When you terminate an instance, any previously attached EBS volumes are left behind, allowing you to attach them to other instances, if desired.

You can create snapshots of your EBS volumes, which allow you to quickly generate identical volumes. Snapshots are useful if you need to share data across Amazon availability zones, or if you want to make the same data available to several Amazon accounts. Finally,

snapshots provide a means of data backup. If a volume fails for some reason, you can deploy a new volume from the original snapshot without much loss.

Elastic Load Balancer

Amazon Elastic Load Balancers (ELBs) are a way to distribute work across multiple EC2 instances. All requests to your server go through the load balancer, which then evenly distributes the requests to the available EC2 instances. You can add or remove participating EC2 instances from the load balancer at any time. When you build a site with ArcGIS Server Cloud Builder on Amazon Web Services, an ELB is configured for you and your GIS server instances are placed beneath it.

Regions and Availability Zones

Amazon EC2 consists of multiple data centers around the world that are designed to address enterprise architecture challenges.

- Amazon **Regions** represent data center facilities in dispersed geographic locations such as the United States, Europe, and Asia.
- Amazon **Availability Zones** are distinct locations within a region that are engineered to be isolated from failures in other Availability Zones. Configuring your application in multiple zones provides an extra layer of availability in case one of the zones should fail.

Relational Database Service

Amazon Relational Database Service (RDS) is a web service that provides you access to a database instance. Amazon patches the database software for you and creates backups of your databases, which are retained for one day by default. When you build an ArcGIS Server site with ArcGIS Server Cloud Builder on Amazon Web Services, you have an option to include an Amazon RDS for Microsoft SQL Server instance with your site.

S3

Amazon Simple Storage Service (S3) is an Amazon service designed specifically for data storage in the cloud. This storage option has the lowest potential for data failure or loss. You can use S3 as a place for data backup or as a middle ground for data transfer between your on-premises deployment and your EBS volumes. Also, any snapshots you create of your EBS volumes are stored on S3.

ArcGIS Server architectures on Amazon Web Services

ArcGIS Server Cloud Builder on Amazon Web Services (AWS) gives you various options for your ArcGIS Server site architecture. For example, you can choose to put multiple GIS server instances in your site to handle large processing loads, and you can choose to put the geodatabase on its own instance to isolate it from the GIS server tier. This topic discusses the architecture options available when building your site with Cloud Builder. It also explains architecture guidelines and limitations if you're building your site manually using the AWS Management Console.

Architectures available with Cloud Builder

ArcGIS Server Cloud Builder on Amazon Web Services allows you to build a simple site with one GIS server to a complex site with many GIS servers. It also allows you to include an enterprise geodatabase and determine where the enterprise geodatabase is placed.

ArcGIS Server without an enterprise geodatabase

You can use Cloud Builder to create an ArcGIS Server site running on Windows or Linux without an enterprise geodatabase. Such a site can have one or many GIS servers. All the participating GIS servers must use the same operating system. In all cases, the GIS server or servers are placed beneath an Elastic Load Balancer (ELB).

ArcGIS Server with an enterprise geodatabase on the same AWS instance

You can optionally choose to create a site that has an enterprise geodatabase running on the same AWS instance as ArcGIS Server. If the site contains multiple GIS servers, the enterprise geodatabase runs on just one of those servers.

If you built your ArcGIS Server site on Linux, the enterprise geodatabase runs on PostgreSQL. If you build your ArcGIS Server site on Windows, the enterprise geodatabase can run on Microsoft SQL Server Express or SQL Server Standard.

Running the enterprise geodatabase on the same AWS instance as ArcGIS Server can help you save hardware costs (the hourly fees you pay to Amazon for use of the instances). This architecture also reduces latency between the server and the database. However, when you run the server and the database on the same AWS instance, you are sharing hardware resources and run a greater risk of site disruption if either component is overloaded or goes offline.

ArcGIS Server with an enterprise geodatabase on its own AWS instance

You can choose to create a site that has an enterprise geodatabase running on its own AWS instance, separate from the ArcGIS Server AWS instance(s). If you built your ArcGIS Server site on Linux, the enterprise geodatabase runs on PostgreSQL. If you built your site on Windows, the enterprise geodatabase runs on SQL Server Standard. If you choose to create a site with an Amazon Relational Database Service (RDS) for Microsoft SQL Server, it will always run on an instance separate from your ArcGIS Server instances.

Running the enterprise geodatabase on a separate instance from ArcGIS Server can result in increased fees for instance usage. However, separating the GIS server and database tiers gives you more freedom to adjust or fix one tier without affecting the other.

Architectures available when building your site manually

When you build your site manually using the AWS Management Console, you can launch one or many AWS instances running ArcGIS Server. If you launch multiple instances, you must also launch an ELB and add the instances to it. All ArcGIS Server instances you add beneath the ELB must be identical.

Each ArcGIS Server instance beneath the ELB operates independently of the other instances beneath the ELB. In other words, each instance has its own configuration store, server directories, logs, and so forth. This limits the types of things you can do with the server, such as asynchronous geoprocessing or distributed map caching jobs. If you want a genuine distributed architecture that can do those types of things, use Cloud Builder to create your site.

There are three Amazon Machine Images (AMIs) you can choose when launching instances for your site, or you can use a custom AMI that you have derived from one of these three AMIs. For descriptions of the AMIs, see The ArcGIS Server AMIs. All the AMIs include ArcGIS Server and an enterprise geodatabase. If you want to run the enterprise geodatabase on a separate instance from ArcGIS Server, it's recommended that you use Cloud Builder to create your site.

Do I need to install the ArcGIS Web Adaptor?

ArcGIS Server Cloud Builder on Amazon Web Services does not install or configure the ArcGIS Web Adaptor because the Elastic Load Balancer (ELB) plays many of the same roles, distributing incoming requests among the GIS servers in your site. However, after you create your site, you can optionally log in to your instance and download the Web Adaptor setup from My Esri. This is necessary if you'll

be hosting ArcGIS API for JavaScript applications on a Windows instance. See Deploy a web application on a Windows site for more details.

The ArcGIS Server AMIs

At version 10.3 of ArcGIS for Server, Esri provides three Amazon Machine Images (AMIs) you can use when building a site with the Amazon Web Services (AWS) Management Console. These AMIs are also used behind the scenes when you build sites with ArcGIS Server Cloud Builder on Amazon Web Services. Before you can build a site using either technique, you must provide Esri Customer Service with your Amazon account number so they can make the AMIs accessible to you.

The AMIs are

- Esri ArcGIS 10.3 Server on Ubuntu with PostgreSQL
- Esri ArcGIS 10.3 Server with SQL Server
- Esri ArcGIS 10.3 Server with SQL Express

Below is a summary of the software and components included with each AMI.

	Esri ArcGIS 10.3 Server with PostgreSQL	Esri ArcGIS 10.3 Server with SQL Server	Esri ArcGIS 10.3 Server with SQL Express
Operating system	Ubuntu 12.04	Windows Server 2012	Windows Server 2012
Database	PostgreSQL 9.3.4	Microsoft SQL Server 2012 (Standard Edition)	Microsoft SQL Server 2012 (Express Edition)
Web server	Apache	IIS 8	IIS 8
Web browser	N/A	Internet Explorer 10	Internet Explorer 10
Root storage	60 GB EBS volume	60 GB EBS volume	60 GB EBS volume
Attached storage	10 GB EBS volume on /gisdata	10 GB EBS volume on D:	10 GB EBS volume on D:
ArcGIS software	ArcGIS 10.3 for Server	ArcGIS 10.3 for Desktop ArcGIS 10.3 for Server	ArcGIS 10.3 for Desktop ArcGIS 10.3 for Server
ArcGIS Server account name	arcgis	arcgis	arcgis
Other software	N/A	Microsoft Security Essentials	Microsoft Security Essentials

Typically, it's easier to build your site using ArcGIS Server Cloud Builder on Amazon Web Services. See Build an ArcGIS Server site on Amazon Web Services for setup instructions. However, if you will be using the AWS Management Console to create a site from these AMIs, follow the instructions in Launch an Amazon Web Services instance running ArcGIS Server instead.

Use Cloud Builder to create and maintain a site

Install Cloud Builder

ArcGIS Server Cloud Builder on Amazon Web Services is a free downloadable app that helps you create an ArcGIS Server site on Amazon Web Services (AWS). This app is the recommended way to get started with building a site on AWS, although in some limited workflows you can alternatively use the AWS Management Console.

Cloud Builder requires the Windows operating system and an Internet connection. You can install Cloud Builder on your own desktop machine; you don't have to install it on a server.

Follow these steps to download and install Cloud Builder.

Note: ArcGIS Server Cloud Builder on Amazon Web Services requires Adobe AIR. The setup will install this component if it is not detected. If necessary, consult your system administrator to ensure that your system settings will not restrict the download or installation of Adobe AIR.

1. Download ArcGIS Server Cloud Builder on Amazon Web Services.
   This download is available to anyone who has purchased ArcGIS for Server. To get to the download, you must log in to My Esri and view the product download page. The ArcGIS Server Cloud Builder on Amazon Web Services download is listed next to the ArcGIS for Server download. If you do not see the download, contact Esri Customer Service (U.S. customers) or distributor (international customers).
2. Launch the installer executable (`.exe`) file you downloaded.
3. Choose the installation location and set the installation preferences such as whether or not you want a desktop shortcut.
   You might notice that the default installation location is the 32-bit Program Files directory `C:\Program Files (x86)`. Cloud Builder is a 32-bit application that works in both 32-bit and 64-bit environments that have a 32-bit compatibility mode.

   When you have finished adjusting these settings, click **Continue**.
4. Evaluate the Adobe AIR license agreement.
   If you agree with the Adobe AIR license terms, click **I Agree** to begin the Cloud Builder installation. Depending on your Windows settings, you may be prompted to allow administrator permissions before installation begins.

ArcGIS Server Cloud Builder

ArcGIS Server Cloud Builder on Amazon Web Services helps you build and maintain an ArcGIS Server site in Amazon EC2. It is a lightweight desktop application downloadable by anyone who has purchased ArcGIS for Server.

When you build a site in Amazon EC2, there are various components that you have to create, connect, and maintain, such as an EC2 instance, a security group, a key pair file, an Elastic Load Balancer (ELB). You can use Cloud Builder, which hides the complexity of assembling and administering your site.

This topic briefly introduces the things you can do with Cloud Builder.

Build a site

Cloud Builder helps you build a complete ArcGIS Server site in Amazon EC2. The site can be simple or complex. It can have one GIS server or many GIS servers. It can have an enterprise geodatabase attached. It can expand or shrink in response to CPU utilization.

When you create a site using Cloud Builder, you provide a license file and your primary site administrator credentials for ArcGIS Server. This allows Cloud Builder to authorize ArcGIS Server for you and create a site. When Cloud Builder has finished creating the site, it gives you a link you can use to access Manager. You can immediately begin publishing services to the site.

See Build an ArcGIS server site on Amazon Web Services to get started.

Build a custom site

Many cloud users like to save their own set of services and data to a machine image that they can use to launch identical instances in the future. This is helpful when scaling out a site with more machines or when replacing a machine that has been taken offline. Using Cloud Builder, you can create site templates that preserve the state of your machine for future use. When launching future sites, you can use the template to ensure that your services and data are configured from the beginning.

See Create a site template.

Maintain a site

Using Cloud Builder, you can easily stop or start your site. Stopping a site is like turning off the power when you go home at night. A stopped site incurs fewer charges than a started site, and you might choose to stop your site on nights, weekends, or other off-hours. This can be a good way to cut costs if you are not running mission-critical applications or if your site is just used for development or testing purposes.

See Stop and start your site.

Back up a site

Cloud Builder allows you to make a backup that preserves the current state of your site, including services, apps, and data. If your site becomes corrupted or experiences other problems, you can use the backup to return to a working state.

See Make a backup of your site.

Build an ArcGIS Server site on Amazon Web Services

ArcGIS Server Cloud Builder on Amazon Web Services is a downloadable desktop application that helps you create an ArcGIS Server site on the Amazon Web Services (AWS). The steps below explain how to create a site using Cloud Builder.

Configure ArcGIS Server Cloud Builder on Amazon Web Services

You need the following to use ArcGIS Server Cloud Builder on Amazon Web Services:

- An Amazon account
- The ArcGIS for Server Amazon Machine Images (AMIs) shared to your Amazon account
- The **Access Key** and **Secret Access Key** of a user who is an administrator for your Amazon account
- The ArcGIS Server Cloud Builder on Amazon Web Services application
- An ArcGIS for Server license file

1. Download, install, and launch ArcGIS Server Cloud Builder on Amazon Web Services.
   You can access the Cloud Builder download when logged in to My Esri. It is listed with your other Esri software downloads. You need to have purchased ArcGIS for Server before you can see the download.
2. Log in using the **Access Key** and **Secret Access Key** belonging to an administrator user of your Amazon account.
   Amazon Identity and Access Management (IAM) services allow you to create users and groups and assign them various levels of privileges to your account. Each user has an associated Access Key and Secret Access Key. You need to create at least one administrator user and supply that user's Access Key and Secret Access Key when you log in to Cloud Builder.

   If you don't have a user or keys defined, log in to the AWS Management Console and click **Services** > **IAM**. Follow the on-screen prompts to create a group, assign it administrative access, and create a user. Once you create a user, you can download the Access Key and Secret Access Key. You can only download these once, at the time you create the user.

   Be aware that ArcGIS Server Cloud Builder on Amazon Web Services will not allow you to log in if the ArcGIS for Server Amazon Machine Images (AMIs) have not been shared with your Amazon account. Contact Esri Customer Service (U.S customers) or your local distributor to get the AMIs shared with your Amazon account.

Launch an AWS instance

The ArcGIS Server Cloud Builder on Amazon Web Services will take you through the steps to create an AWS instance.

General settings

1. Click the **Sites** tab and click **Create Site**.
2. Type a name and description for your site.
   The name will be applied to various resources that the app creates for you in AWS so you can easily identify them.
3. Use the **Software** drop-down list to choose the AMI used to launch your instance. You can use the Windows or Ubuntu Linux AMIs Esri shared with our account, or you can use a site template that you've previously configured to contain your own software, data, and services.
4. Browse to the Esri license file (with extension `.prvc`) that should be used to license your site. If you need help creating a `.prvc` file, see Frequently asked questions.
5. Click **Next** to proceed to the next panel.

Amazon Web Services settings

Define AWS settings for your site.

1. Choose the AWS region to be used for your site.
   A region represents an Amazon data center complex in a certain area of the world. A site cannot be distributed across regions, although you can maintain a second site in an alternate region for redundancy.
2. Choose whether you want to launch your site in **Amazon EC2** or **Amazon VPC** (Virtual Private Cloud).
   - If you launch in Amazon EC2, you must choose an availability zone. These represent physically isolated data centers within regions. You can spread a site across zones for higher fault tolerance. This is accomplished by choosing **No**

preference for the zone, thereby allowing EC2 to pick a zone for each instance (virtual machine) that is launched in your site.

- Amazon VPC is a mechanism for configuring or extending a private network within EC2. You must have configured a VPC subnet in an AWS region and availability zone to choose this option. Cloud Builder detects the existing subnets for your account and displays them in the **Subnet ID** drop-down list.

3. Choose a key pair to use for this site. A key pair is required if you ever want to log in to one of the instances in your site. You can use an existing key pair or create a new one.
   - To use an existing key pair, select one from the drop-down list. This list only contains key pairs from the region in which you are working.
   - If you create a new key pair, a .pem file is placed in your Windows Documents folder under ArcGISCloudBuilder. For example, the path to your key pair file might look like `C:\Users\username\Documents\ArcGISCloudBuilder\arcgis-TestSite.pem`. You should move this file to a secure location and keep it available for future use.
4. Click **Next** to proceed to the next panel.

ArcGIS Server instance settings

Specify settings for the instances that comprise your ArcGIS Server.

Caution: The instance type, as well as the minimum and maximum number of instances you choose, can greatly affect the amount of money that you are charged by AWS. Before choosing these settings, carefully estimate your site usage and the server power you'll need to accommodate that usage.

1. Choose the instance type to be used in your site.
   The instances available to you will vary slightly depending on which region you are using. However, micro instances are not available in any region because they do not meet the minimum memory requirement for ArcGIS Server.
2. Choose the size of the Amazon Elastic Block Store (EBS) volume to attach to each GIS server instance in your site. This EBS volume will be visible as a local disk drive to which you can upload your data. Your ArcGIS Server configuration store and server directories will also be placed on this drive on one of the instances.
3. Choose whether the root drive and attached EBS volume should be deleted when the site is terminated.
   Preserving the drives allows you to attach it to other sites in the future if you choose. However, you must remember to delete the drives manually when they are no longer needed. You will incur charges for the drives as long as they exist.

The next two steps apply to ArcGIS for Server enterprise licensed sites only.

4. Choose the number of AWS instances that will participate in your site.
   The **Number of instances** property represents the minimum number of instances that will be launched when the site is created.

   License: You must not run ArcGIS Server on a greater number of CPU cores than you have licensed with Esri, regardless of whether the cores are in the cloud or on-premises. Also be aware that you are responsible for all Amazon Web Services charges you incur for AWS instances and other resources launched using Cloud Builder.

5. Optionally check **Enable auto-scaling** to launch or terminate new AWS instances automatically based on usage triggers that you specify.
   Autoscaling monitors site usage and adds or removes instances based on the CPU usage and duration. When more people and applications access your site, more instances are added to it. When demand decreases, instances are removed. Instances will never be fewer than the number you specify for **Number of instances**, and will never exceed the number you specify for **Maximum number of instances**.

   Amazon CloudWatch is the service that provides CPU monitoring on your instances and makes the autoscaling triggers possible. You will see an extra fee applied for the CloudWatch service if you choose to enable autoscaling.

   a. Set the **Maximum number of instances** that can be launched.
   b. Specify the percent CPU usage and duration that will cause an instance to be removed from your site.
   c. Specify the percent CPU usage and duration that will cause an instance to be added to your site.

   For example, if you keep the default settings, when your site experiences over 80 percent CPU usage for five consecutive minutes, a new GIS server instance will be added to the site unless you have reached the maximum number of instances you

specified. When your CPU usage goes below 20 percent for five consecutive minutes, a GIS server instance will be removed from your site and terminated unless your site has been reduced to the original number of instances you specified.

6. Click **Next** to proceed to the next panel.

Geodatabase settings

You can choose to include geodatabases in your site by checking **Include enterprise geodatabase**. Relational database management system options vary depending on which AMI and ArcGIS Server license you are using to launch your site.

- If you are using an ArcGIS for Server Workgroup license, you can create a site that includes **Microsoft SQL Server Express on an ArcGIS Server EC2 instance**.
- If you are using an ArcGIS for Server Enterprise license and ArcGIS for Server (Ubuntu Linux) AMI, you can create a site that includes either **PostgreSQL on an ArcGIS Server EC2 instance** or **PostgreSQL on its own dedicated EC2 instance**.
- If you are using an ArcGIS for Server Enterprise license and ArcGIS for Server (Windows) AMI, you can create a site that includes one of the following:
  - **Microsoft SQL Server Express on an ArcGIS Server EC2 instance**
  - **Microsoft SQL Server Standard on an ArcGIS Server EC2 instance**
  - **Microsoft SQL Server Standard on its own dedicated EC2 instance**
  - **Amazon RDS for Microsoft SQL Server**

Relational database management systems running on their own dedicated EC2 instance require that you specify the instance type, EBS volume size, and whether the volume should be deleted at the time the site is terminated. Be aware that all these settings affect the amount charged to your Amazon account.

Amazon RDS for Microsoft SQL Server always runs on its own dedicated instance, for which you must specify the instance class and allocated storage size. You must additionally choose the subnet group to use if you are running the Amazon RDS for Microsoft SQL Server instance in Amazon VPC.

Click **Next** to proceed to the next panel once you have made your selections

Security settings

You must create a primary site administrator for ArcGIS Server. You can also specify an SSL certificate for encrypted communication.

1. Type the user name and password to use for the ArcGIS Server primary site administrator account.
   The primary site administrator is not an operating system account; it is an account built into ArcGIS Server that you create at this time. You'll use this account for logging in to Server Manager and making connections to your server until you are able to further configure security on your site.

   The password must be 8 to 16 characters in length, and cannot contain a forward slash (/) or at sign (@).

   If you choose to include an enterprise geodatabase in Amazon RDS for SQL Server in your site, the password you use for the primary site administrator will also be used for the database users that the geodatabase requires.
2. Choose whether to install an SSL certificate when the site is created. This allows encrypted communication with your site. See Set up SSL using Cloud Builder to learn more about this option.
3. Click **Next** to proceed to the summary panel.

Review settings and create the site

Review the settings for the site. Click **Back** to make changes on a previous panel; otherwise, click **Finish** to create the site.

It can take a while for the site to be created. A series of messages will appear explaining what the application is doing while it is creating the site.

⚠ **Caution:** You must remain logged in to Cloud Builder until site creation is complete.

The site with the resources you specified are created on EC2. These include one or more instances with Amazon CloudWatch enabled, EBS volumes, and an Elastic Load Balancer (ELB). You immediately begin incurring charges for these. To learn more about the costs of these resources, see http://aws.amazon.com/pricing/ec2.

The next steps

You'll need to connect to the server to load and publish data. See Use your site for more information.

Remote access to your instances is not enabled by default. See the following topics for help if you want to log in to one of your EC2 instances:

- Open an Amazon EC2 security group for ArcGIS Server
- Administer your Amazon EC2 instance with Windows Remote Desktop
- Administer your Amazon EC2 Ubuntu Linux instance with remote access from Windows
- Administer your Amazon EC2 Ubuntu Linux instance with remote access from Linux

Sites that include geodatabases in PostgreSQL also require that you change passwords for security reasons. You can also change the password of the administrator login on Windows instances.

Use your site

Use your site

You connect and publish to ArcGIS Server on Amazon Web Services the same way you connect and publish to an ArcGIS server site on premises. This topic contains a few things to remember that will make it easier to work with your site on Amazon EC2.

Connect to the server

When you connect to a site that you built with ArcGIS Server Cloud Builder on Amazon Web Services, you do not use the address of any of the GIS server machines; instead, you use the URL of the Amazon Elastic Load Balancer. This load balancer has been configured to receive incoming traffic to the site through the common port 80, so there is no need to use the typical ArcGIS port 6080 in the URL. The load balancer distributes requests to the different GIS server machines in the site.

If you're not sure of the load balancer URL, open ArcGIS Server Cloud Builder on Amazon Web Services and click the name of your site. You'll see an expanded list of metadata, including an item called **Manager URL**. The first part of this URL contains the load balancer address (it typically contains .elb.amazonaws.com).

See Connect to your site from ArcGIS for Desktop for step-by-step connection instructions.

It is not necessary for you to install the ArcGIS Web Adaptor on your instances. The Elastic Load Balancer fills many of the roles typically played by the Web Adaptor.

Copy data to the server

There are two ways you can make your GIS data accessible to your Amazon EC2-based site:

- You can log in to your instance and copy the data to it manually, using a technology like FTP or Remote Desktop Copy and Paste.
- You can choose to have ArcGIS copy the data to the server automatically when you publish a service. This option does not allow data to be shared between services, nor does it allow data synchronization between the cloud and your on-premises deployment; however, this option is required if you don't have permissions to log in to the server.

Often, you may be publishing from a machine on premises that has different data paths than you use on your EC2 instance. Registering your data locations with ArcGIS Server helps you ensure that your paths will be automatically corrected when you publish.

Once your data is on the server, you may want to keep it in sync with a geodatabase on premises. An ArcGIS Server geodata service can help with this. For more information, see Use a geodata service and connected replica.

Connect to ArcGIS Server on AWS from ArcMap

Connect to your ArcGIS Server site running on Amazon Web Services from the **Catalog** window in ArcMap to view and work with your services in the desktop environment. For example, you can drag and drop a service from the **Catalog** window into the ArcMap table of contents.

Note: Instructions are only provided here for connecting to ArcGIS Server, not your enterprise geodatabase. You should avoid connecting to your geodatabases from your on-premises environment, primarily for performance reasons.

Follow these steps to connect to your server.

1. In the Catalog tree, expand the **GIS Servers** node.
2. Double-click **Add ArcGIS Server**.
3. Choose whether you want to make an administrator, publisher, or user connection and click **Next**.
   The steps in this topic assume an administrator connection, but the process is similar for the other types of connection.
4. For the **Server URL**, type the URL of the ArcGIS Server site you want to connect to.
   The URL takes the format `http://<Elastic Load Balancer address>/arcgis`. One way to get the URL is to display the **My Sites** list in Cloud Builder and click the name of your site. Select and copy the **Manager URL**, paste it in the **Server URL** text box, and delete `/manager/` from the end.
5. Choose **ArcGIS Server** from the **Server Type** drop-down list.
6. During the publishing process, a service definition file is created and temporarily stored locally on disk. When the publishing process completes, the service definition is uploaded to the server and the local file is deleted. You can optionally change the default location by deselecting **Use ArcGIS Desktop's staging folder** and browsing to a new location. For more information, see Changing the default staging folder in ArcGIS for Desktop.
7. Enter the **User Name** and **Password** of an account that has permissions to ArcGIS Server.
   The privileges associated with the account you enter need to correspond to the type of connection you are making. For example, if you are attempting to make a publisher connection, the account needs to have publisher (or administrator) privileges to ArcGIS Server.

   If you've not created any accounts for ArcGIS Server, use the primary site administrator account that you specified when you defined the site in Cloud Builder.

   Optionally, you can choose to save your user name and password information so that you don't have to enter it each time you connect.
8. Click **Finish**. Your connection appears in the **GIS Servers** node in the Catalog tree.
9. Optionally, rename your connection. To do so, right-click the connection in the Catalog tree and choose **Rename**.

Administer your Amazon EC2 instance with Windows Remote Desktop Connection

Once you've built a site with ArcGIS Server Cloud Builder on Amazon Web Services or you've launched an Amazon EC2 instance running ArcGIS Server, you can log in and begin configuring data and services on the instance. The instructions below explain how to log in to a Windows instance using Remote Desktop Connection.

⚠ **Caution:** Before you attempt these steps, you must open port 3389 on the Amazon security group used by your instance. For instructions, see Open an Amazon EC2 security group for ArcGIS Server. You should also confirm with your system administrator that your computer is allowed to make Remote Desktop connections to machines outside your organization's firewall—specifically, machines on the Amazon subnets.

1. Log in to the AWS Management Console and click **Instances**.
2. Find your instance in the list and make sure it is started.
3. In the instances list, click your instance and examine the detailed information that appears below it. Note or copy the name given for **Public DNS**.
4. Right-click your instance and click **Get Windows Password**.
5. Click **Choose File** and browse to the key pair file (.pem) that you specified when you created the instance.
6. Click **Decrypt Password** and save or note the password in a secure place.
7. Right-click your instance and click **Get System Log**. This page shows the status of your instance. You should not continue with the remaining steps until about 5 minutes after the message **Windows is ready to use** appears in this log. If you used Cloud Builder to build your site, you should also wait until your site appears in Cloud Builder's list of running sites.
   Note the following about using the system log to look for the **Windows is ready to use** message:
   - You may have to repeatedly open the log to check for this message.
   - The time it takes for this message to appear can vary based on the responsiveness of Amazon EC2. Factors affecting response time may include the time of day, your location, and the health of the Amazon EC2 zone and region you are using. Expect to wait at least 10 minutes.
   - If the message does not appear after a long period of time (for example, 30 minutes), Amazon is experiencing a problem starting the instance. Consider terminating the instance and launching a new one.
8. About 5 minutes after the **Windows is ready to use** message appears in the system log, open **All Programs** > **Accessories** > **Remote Desktop Connection**.
9. In the **Computer** text box, enter the public DNS name that you copied or wrote down previously and click **Connect**.
10. Choose to connect using the **Administrator** account.
11. Type the administrator password you retrieved earlier in these steps.
    You are now logged in to the instance and should see the Windows desktop.
12. Change your Administrator password.
    a. In Windows, open **User Accounts**.
    b. Click **Manage User Accounts**.
    c. Choose the Administrator account in the list and click **Reset Password**.
    d. Type the new password twice to confirm, and click **OK**.

    ⚠ **Caution:** It is strongly recommended that you follow the above steps to change the initial Administrator password.
    Your randomly generated Administrator password is not always going to be available for retrieval through the AWS Management Console. After stopping and starting the instance, **Get Windows Password** no longer shows the password retrieval dialog box. Instead, a message box appears informing you that no password was found.
13. Authorize ArcGIS for Desktop.
    a. Start ArcGIS Administrator.

b. Click the **Desktop** folder and choose a licensing option. If you choose a Concurrent Use license, supply the address of a license manager that is visible to the instance. If you choose a Single Use license, click **Authorize Now** and proceed through the wizard.

There is no separate step to authorize ArcGIS for Server, since you were already required to supply your license information when you launched the instance.

14. Open ArcGIS Server Manager and choose to create a new site. You can now create a site, move data to your instance, and begin publishing services.

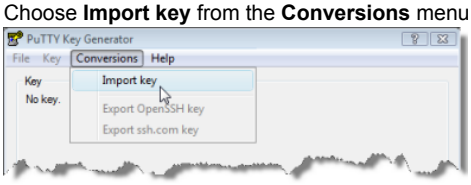
Administer your Amazon EC2 Ubuntu Linux instance with remote access from Windows

Once you have built a site with ArcGIS Server Cloud Builder on Amazon Web Services, you can use a terminal emulator application to make an SSH connection from your local Windows computer to your Ubuntu Linux Amazon EC2 instance.

Caution: Before you attempt an SSH connection to your instance, you must add a rule to your security group that opens port 22 (the SSH port) to your network. See Open an Amazon EC2 security group for ArcGIS Server for instructions.

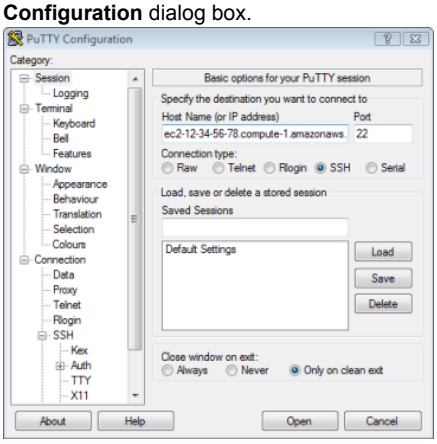
The following steps provide an example of connecting from a local Windows computer using PuTTY, an open source terminal emulator.

1. Use PuTTYgen to make a private key (.ppk) file from the Amazon key pair (.pem) file that was generated when you launched your site using ArcGIS Server Cloud Builder on Amazon Web Services.
   The .ppk file is PuTTY's SSH private key format.
   a. Open PuTTYgen.
   b. Choose **Import key** from the **Conversions** menu.



   c. Navigate to and choose the .pem file that was generated when you launched your site with ArcGIS Server Cloud Builder on Amazon Web Services.
      The file will have the same name as your site and is located in the ArcGISCloudBuilder folder on the machine where you installed Cloud Builder.
   d. Type a passphrase in the **Key passphrase** text box.
   e. Type the same phrase in the **Confirm passphrase** text box.
   f. Click **Save private key**.
   g. Choose the location in which to create the file.
   h. Type a name for the file.
   i. Be sure you are saving as type **PuTTY Private Key Files (*.ppk)** and click **Save**.

   The .ppk file is created in the directory you specified.
2. Open PuTTY and create a session.
3. Specify the public DNS of your site in the **Host Name (or IP address)** text box in the **Session** category of the **PuTTY Configuration** dialog box.



4. Expand the **Connection** category, expand **SSH**, and click **Auth**.

5. Click **Browse**, navigate to the location where you saved your .ppk file, choose the file, then click **Open**.
6. Click **Open** to connect to the Linux instance.
   If this is the first time you are connecting to this DNS or you did not cache the host key in the registry, a **PuTTY Security Alert** dialog box appears. Click **Yes** if you want to cache the server host key and connect, or click **No** if you want to connect but do not want to cache the server host key.
7. Log in as the arcgis user.
   All ArcGIS for Server sites have a user named arcgis. The private key is used in place of a password for the arcgis user when connecting directly to a Linux instance.

   The arcgis user is not a super user on the Linux instance. If you need to perform tasks such as mounting a drive, you must log in as a super user.
8. Type the passphrase you set for the .ppk file.

You are now connected to the Ubuntu Linux Amazon EC2 instance.

Administer your Amazon EC2 Ubuntu Linux instance with remote access from Linux

If you use the Amazon Web Services (AWS) console to create an AWS instance running ArcGIS for Server, you can log in to authorize ArcGIS for Server, and configure data and services on the instance.

1. Log in to the AWS Management Console and open the instance list for your account and region.
2. Make sure your instance is running.
3. Choose your instance from the list and examine its Description information. Note or copy the name given for **Public DNS**.
4. Open any SSH client (such as PuTTY).
5. Locate your private key file that you specified when you created the instance, for example, arcgisserver.pem.
6. Use chmod to make sure your key file isn't publicly viewable; otherwise, SSH won't work:

   ```
   chmod 400 arcgisserver.pem
   ```

7. Connect to your AWS instance using its public DNS.
   To perform ArcGIS Server tasks, such as authorizing ArcGIS for Server, connect as the arcgis user. To perform operating system-level tasks, such as starting the web server, connect as the ubuntu user.

   The following example makes an ssh connection as the arcgis user to an instance with a public DNS of ec2-50-16-11-231.compute-1.amazonaws.com using the key file arcgisserver.pem:

   ```
   ssh -i arcgisserver.pem arcgis@ec2-50-16-11-231.compute-1.amazonaws.com
   ```

8. Authorize ArcGIS for Server by running the authorizeSoftware script. You'll need to reference an Esri license file.

   ```
   arcgis@ip-10-82-222-58:$ cd /arcgis/server/tools/
   arcgis@ip-10-82-222-58:/arcgis/server/tools$ ./authorizeSoftware <prvc_license_file> [email]
   ```

   Tip: You can run the authorizeSoftware tool with the `--help` option to see all possible syntax.

9. ArcGIS for Server is ready for you to create a site.

   Dive-in: Every time you stop and start the instance, Amazon assigns your instance a new machine name and Public DNS name/IP Address. Therefore, it is not helpful to create shortcuts or hard-coded references to your instance unless you have configured an Amazon Elastic IP for your instance and you reassociate this IP every time you start the instance.

Change the ArcGIS Server account password (Windows only)

When you launch an Amazon Web Services (AWS) instance using the ArcGIS for Server Windows AMI, the ArcGIS Server account is preconfigured on the instance with the name arcgis and a randomly generated password that is unknown to anyone.

You don't have to change the account password, but if you want to change it, follow these steps:

1. Using the Administrator account, make a remote desktop connection to the ArcGIS Server site instance. In a multiple instance site, this is referred to as the SITEHOST.
   Remember that to remotely connect to an AWS instance, you must open the RDP port in your security group.
2. Set a new password for the arcgis operating system account.
   For instructions on changing Windows account passwords, see the Microsoft documentation.
3. Once you have reset the account password, you must update the properties of the ArcGIS Server service so that it runs using the new password.
   The ArcGIS Server service should be logging on with the **.\arcgis** account. You can reset the password for this account from the **Services** Microsoft Management Console (MMC).
   a. Right-click the service **ArcGIS Server** and click **Properties**.
   b. Click the **Log On** tab and make sure **This account: .\arcgis** is chosen for the log on.
   c. Type the new password twice to confirm and click **OK**.
4. Repeat the above steps on all GIS servers in your site, using the same password for the ArcGIS Server account.

Update site properties

You can use ArcGIS Server Cloud Builder on Amazon Web Services to modify some of the properties of your existing ArcGIS site. For example, you can update your site to adjust the computing power allocated to your GIS Servers, enable or disable autoscaling, or change the description of your site.

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services and click the **Update** button next to your site name.
2. Navigate the wizard using the **Next** and **Back** buttons, and apply the desired changes.
   Some of the site settings in ArcGIS Server Cloud Builder on Amazon Web Services may be disabled. This indicates the setting cannot be updated. The following table describes which settings you should expect to be available for updating:

	ArcGIS site is stopped	ArcGIS site is running	Comments
Description	Yes	Yes	
Region and Availability zone	No	No	Although you cannot change the region and availability zone of an existing site, you can use templates to launch identical sites in different regions and availability zones.
Key pair	No	No	
Instance type	Yes	No	You can change your instance type to increase or decrease the computing power of EC2 instances used for your ArcGIS site.
Launch as EBS-Optimized instance	Yes	No	If the instance type you selected supports EBS-optimized, you can disable or enable it.
EBS volume Size and Deletion policy	No	No	
Number of instances	Yes	Yes	Use this setting to adjust the number of GIS Servers powering your site.
Autoscaling	Yes	Yes	You can enable autoscaling policies to dynamically adjust the computing power of your site, or disable this Amazon Web Services feature.
Geodatabase	No	No	
SSL certificate	Yes	No	You can enable SSL or change the SSL certificate if it's already enabled.

3. Advance to the **Summary** panel of the wizard and click **Finish** to apply the changes.

For more fine-grained control over your site, you can build it manually using the AWS Management Console.

Create custom sites using templates

What is a site template?

ArcGIS Server Cloud Builder on Amazon Web Services uses a set of predefined Amazon Machine Images (AMI) when it builds a site. If you want to use an image with your own set of services and applications, you can build a site and preserve it as a template. You can then use the template to launch future sites, saving you some configuration work on each site.

Templates preserve the licensing of your site. For example, you cannot use a template created with a Workgroup license to launch a site with more than one GIS server. Similarly, templates preserve the geodatabase configuration. If your template does not include a geodatabase running on its own instance, you cannot choose to include one when you create a new site from the template.

The template is stored using Amazon Web Services and incurs a storage fee. You will see charges from both Amazon EC2 and Amazon S3 that pertain to the template.

Sharing templates

Cloud Builder allows you to share your templates with users of other Amazon accounts. Likewise, you can import templates into Cloud Builder that other Amazon account owners have shared with you. The ability to share templates is useful if you are building a custom solution for a client, if there are multiple Amazon accounts in your organization, or if you have to set up a lab or classroom wherein each participant will use their own Amazon account.

License: When working with templates, keep in mind that you are not permitted to share your licensed Esri software with users who fall outside the realm of your license agreement.

The remainder of this help book explains how to create, share, import, and launch sites from templates.

Related topics

- Create a site template
- Create a site from a template
- Share a template with other accounts
- Import a shared template

Create a site template

The instructions below explain how to create a site template using ArcGIS Server Cloud Builder on Amazon Web Services. You can later use the template to launch identical sites.

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services and create a site according to the following guidelines:
   - Create the site in the region where you'll be using the template. Templates are region-specific; they can only launch sites in the region where they were created.
   - If there's any possibility that your template will be used to launch multiple machine sites, create the site with two instances. One of these instances will become the image for the site server (the GIS server containing the configuration store and the server directories) and the other will become the image for the additional GIS servers. For example, if you are hosting data and web apps on your site, you might want to configure these on the site server only, and configure the additional GIS servers without the web apps and data.

   See Build an ArcGIS Server site on Amazon Web Services if you need help creating the site.
2. Log in to the EC2 instances in your site and use the operating system tools to change the administrator password to something you can remember. This password will be applied to any site created with the template.
3. Configure any services, applications, or third-party software that you want to have on the instances. You can also reset the ArcGIS Server account password on Windows instances.
   If you used two instances, both the site server (SITEHOST) and the additional GIS server are saved in the template. When you create a site with this template, the resulting site server is launched from the site server image, and all the additional GIS servers are launched from the additional GIS server image.

   Always reset the password of the ArcGIS Server account on SITEHOST first, and then reset the password on the other instances in your site. See Change the ArcGIS Server account password for more information.

   When preparing to create a template, one thing you might want to do with the additional GIS server is remove the default attached Elastic Block Store (EBS) volume or replace it with something smaller. This reduces the storage costs you have to pay if you launch many GIS servers from this template.

   You can tell which instance is the site server by examining the instance names in the AWS Management Console.

   If you prepare and save just one instance in your template and you launch a multiple machine site from that template, the additional GIS servers (beyond the site server) are created using the basic ArcGIS Server AMIs.
4. In ArcGIS Server Cloud Builder on Amazon Web Services, click the **Templates** tab.
5. Click **Create Template**.
6. Choose your site from the drop-down list and type a name and description for the template.
7. Click **Finish**.
   This begins the creation of your template, which can take some time. It's recommended that you wait at least 30 minutes before attempting to use your template.
8. If you plan on sharing this template with other accounts and you also plan on keeping the parent site up and running, change the administrator password of the instances in the parent site.

Now you can create a site using this template.

You can delete your template by clicking the **Templates** tab, and then clicking the **Delete** button next to your template name. If you have shared the template with other Amazon accounts, they will no longer be able to build sites with it.

Create a site from a template

A site template gives you an easy way to launch a site containing your desired set of data, services, applications, and third-party software. Create a site template, and then follow the instructions below to create new sites from the template.

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services, click the **Sites** tab, and click **Create Site**.
2. Follow the steps in Build an ArcGIS server site on Amazon Web Services to create a site, making sure to set the following properties:
    - On the **General** panel, set the **Software** property to **My Own ArcGIS Server Template**.
    - On the **Amazon Web Services** panel, set the **Region** property to the region in which your template is stored.
    - On the **ArcGIS Server** panel, set the **ArcGIS Server templates** property to the name of the template you want to use.
    - On the **Security** panel, specify any user name and password you would like to use for the primary site administrator account for this site. You don't have to enter the same primary site administrator user name and password that you used when you originally created the template.

    You may notice a few restrictions as you create the site using the template. You cannot use a template created with a Workgroup license to launch a site with more than one GIS server. Also, you must use the geodatabase included in the template, although you can change its instance type (size). You cannot choose to include a separate geodatabase instance if the template did not include one.

    If the template includes two images, both your site server (the instance with the server directories and configuration store) and your additional GIS servers may contain customized software, data, and other configurations set by the template author. If the template includes just one image, only the site server machine will have customizations; the remaining GIS servers will be launched using the Esri ArcGIS Server AMIs. Contact the template author if you have questions about the architecture used to build the template.

Once the site is created, you can choose to log in to its EC2 instances and make further customizations. To log in, you will need to know the administrator password that was used on the original instances from which the template was created. Contact the author of the template to get the administrator password. As a best practice for security, it's recommended you change this password after you create your site.

Share a template with other accounts

Once you've created a site template, you can share it with other Amazon accounts. The owners of these accounts can then import the template into ArcGIS Server Cloud Builder on Amazon Web Services and build sites with it.

The ability to share templates is useful if you are building a custom solution for a client, if there are multiple Amazon accounts in your organization, or if you have to set up a lab or classroom wherein each participant will use their own Amazon account.

License: When working with templates, keep in mind that you are not permitted to share your licensed Esri software with users who fall outside the realm of your license agreement.

The steps below explain how to share an existing template with one or more Amazon accounts.

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services and click the **Template** tab.
2. Find your template in the list and click the **Permissions** button.
3. Click **Add Account**.
4. In the **AWS Account Number** input box, type the identifier of the Amazon account with which you want to share the template. See Find your Amazon account identifier to review how to find the identifier for an Amazon account.
5. In the **AWS Account Email** input box, type the e-mail address associated with the Amazon account. This is the e-mail address the account users enter when logging in to the AWS Management Console.
6. Click **Finish** and repeat the above process for any other accounts with which you want to share your template.
7. Return to the **Templates** tab and expand your template information by clicking the arrow to the left of the template name.
8. Copy or write down the **Template ID** and provide it to the owners of the Amazon accounts with whom you shared the template. The Amazon account owners need this ID to import the template into Cloud Builder.
   You also need to share the administrator password used by the instances in this template. You should encourage the person receiving the template to change the administrator password after he or she creates a site from the template.

To view the steps for importing a template, see Import a shared template.

If you share a template and later delete it, users cannot build any sites with the template.

Related topics

Import a shared template

Import a shared template

If someone has shared a site template with your Amazon account, you can import the template into ArcGIS Server Cloud Builder on Amazon Web Services and use it to build your own sites.

Before you can import a template, you need to get the ID from the template owner. The template ID is visible when you expand the template details in Cloud Builder.

Before you can log in to any instances that you launch with the template, you'll also need to get the administrator password from the template owner.

Follow these steps to import a template:

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services and click the **Template** tab.
2. Click **Import Template**.
3. In the **Template ID** input box, type the ID of a template that has been shared with you. You must obtain this ID from the template owner.
4. In the **New Name** input box, type a name for this template.
5. Click **Finish**.

Now you can build a site from this template, following the guidelines in Create a site from a template.

Once the template owner deletes the template, you cannot launch new sites from it. Your existing sites will not be affected.

Related topics

Share a template with other accounts

Make a backup of your site

Options for AWS-based ArcGIS Server site backups

To achieve a backup of your Amazon Web Services (AWS)-based ArcGIS Server site, you can use the ArcGIS Server backup and restore utilities, the backups provided directly from Cloud Builder, or Cloud Builder templates. The combination of backup technologies you choose depends on what you need to back up.

What do you need to back up?	What should you use to make the backup?
ArcGIS Server service configurations, server object extensions, registered data store item locations, user and role information, list of machines in the site, clusters and their lists of machines, and log settings	ArcGIS Server backup and restore utilities
ArcGIS Server settings not mentioned above such as caches, logs, Web Adaptor configurations, primary site administrator name and password, and so forth.	Cloud Builder backup
GIS data, instance specs, or third-party software on the SITEHOST instance	Cloud Builder backup
GIS data, instance specs, or third-party software on the SITEHOST instance and additional GIS server instances in the site.	Cloud Builder template

You can use any combination of the above techniques to meet your needs. For example, you might want to mix occasional Cloud Builder backups with more frequent site configuration backups by the ArcGIS Server backup utility. The site configuration backups don't preserve as much information, but they also don't require downtime and can be stored in a separate physical location in case your instances are corrupted.

Make a Cloud Builder backup of your site

ArcGIS Server Cloud Builder on Amazon Web Services allows you to create a backup that preserves the current state of your site. If an enterprise geodatabase instance is included in your site, it also participates in the backup. Restoring a backup is an easy way to return your site to its original state if your site becomes corrupted, or if you are doing development and prototyping work and you want to start something over.

When you create a Cloud Builder backup, the current state of one GIS server instance in your site is preserved (the site server instance named SITEHOST in the AWS Management Console, which holds the configuration store). The backup preserves all services, caches, and data that were attached to your site server instance. If you moved the configuration store or your datsets to another instance, you'll need to use some other means to back them up.

When the backup is restored, the image of the one GIS server is used to launch as many GIS servers as originally participated in the site. If you want to launch multiple sites that are architected the same way, use a site template instead of a backup.

The backup is saved using Amazon Web Services and incurs a storage fee. You will see charges from both Amazon EC2 and Amazon S3 that pertain to the backup.

You cannot use a backup to restore a site that has been deleted. When you delete a site, its backups are also deleted.

Follow these steps to create a backup:

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services.
2. Find your site in the list and click its **Backups** button.
3. Click **New**.
4. Type a name and description for your backup and click **Finish**.
   The name and description are important because you can have multiple backups of the same site, each taken at different times. Well-formed names and descriptions can help you find the correct backup to restore.

This begins the creation of your backup, which can take some time. It's recommended that you wait up to 30 minutes before attempting to restore a site using a backup.

To restore your site from this backup, see Restore your site from a Cloud Builder backup.

You can delete your backup by clicking **Backups**, selecting the backup, and clicking **Delete**.

Restore site from a Cloud Builder backup

Restoring a Cloud Builder backup is an easy way to return your site to its original state if your site becomes corrupted or you are doing development and prototyping work and you want to start over.

A site backup contains the image of one GIS server that participated in the site (the site server instance that contained the configuration store). When the backup is restored, the image of this GIS server is used to launch as many GIS servers as originally participated in the site.

Restoring a backup affects only your existing site; it does not create a new site. If you want to create a new site just like your existing one, use a site template instead.

Note: You cannot use a backup to restore a site that has been deleted. When you delete a site, its backups are also deleted.

Follow these steps to restore a site from a backup:

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services.
2. In the list of sites, find the site you want to restore and click its **Backups** button.
3. Select the backup you want to restore and click **Restore**.

Run ArcGIS Server backup and restore utilities on AWS

ArcGIS Server includes backup and restore utilities that preserve properties of your site configuration. You can use these utilities in combination with Cloud Builder backups if you want a basic backup of your site properties that can be stored in a different physical location and restored without site downtime.

When you run the ArcGIS Server backup and restore utilities on your Amazon Web Services site, make sure to do the following:

- When you provide the utility parameters, use the public DNS or LOCALHOST in the server URL; do not use the Elastic Load Balancer name.
- Copy the backup file .agssite to a location apart from the EC2 instance, such as Amazon S3. This allows you to recover your site configuration even if you lose the original EC2 instances or your Cloud Builder backup.
- After running the restore utility, use Cloud Builder to stop your site, then start it again. This restores your original number of instances to the site.

You may see error messages related to the Elastic Load Balancer when you run the restore utility. This is expected.

Stop and start your site

You can cut costs by stopping your EC2 instances when no one is using your site. ArcGIS Server Cloud Builder on Amazon Web Services provides a convenient command to stop all the instances in your site and scale down to just one instance. It also has a command to start your site when you are ready to use it again.

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services.
2. Find your site in the list and click its **Stop** icon.
   This stops all EC2 instances in your site, including any instances running an enterprise geodatabase. Additionally, all the ArcGIS Server instances in your site are terminated except for the site server (the instance containing the configuration store).
3. When you're ready to start your site again, find your site in the list and click its **Start** icon.
   This starts your site server EC2 instance and your enterprise geodatabase instance. Additionally, it adds ArcGIS Server instances to your site until you reach the minimum number of instances you specified when you built your site in Cloud Builder. After running the Start command, you should wait several minutes before attempting to use your site.

Stopping your site makes it unavailable for use until it is started again. While your site is stopped, you do not have to pay the hourly fee for running your instance; however, you still must pay the fees for any disk space taken by your instance.

If you no longer need the AWS instances, you can delete your site.

Delete your site

Deleting your site terminates all instances participating in the site. Use caution with this option, since you cannot restore a deleted site (all backups of the site are also deleted).

To delete your site, follow the steps below:

1. Log in to ArcGIS Server Cloud Builder on Amazon Web Services.
2. Find the site you want to delete and click its **Delete** icon.
3. Confirm whether you want to delete the Elastic Load Balancer (ELB), then click **OK**.
   Cloud Builder lets you choose whether to keep the ELB in case you want to use it with a future site. In some scenarios it may be useful to keep the ELB to preserve existing URLs.

Deleting a site terminates all resources included in the site except for your key pair and possibly your ELB (as described above) and your attached EBS volumes. Your EBS volumes will be only preserved if you unchecked **Delete the root drive and storage drives when the site is terminated** when you built the site in Cloud Builder. Although this checkbox mentions the root drive, the root drive is always deleted when you terminate a site.

You can delete these leftover resources at any time using the AWS Management Console.

Troubleshooting Cloud Builder

If you encounter issues when using ArcGIS Server Cloud Builder on Amazon Web Services, you can create a text file that captures error logs and other information to help you troubleshoot the problem.

Follow these steps to enable debug-level logging for ArcGIS Server Cloud Builder on Amazon Web Services:

1. If ArcGIS Server Cloud Builder on Amazon Web Services is running, close it.
2. Create a plain text file named `logfile` in the `%USERPROFILE%\documents\ArcGISCloudBuilder` folder on the machine where ArcGIS Server Cloud Builder on Amazon Web Services is installed.
   Do not include any extension on the file name; for example, the file must be named `logfile`, not `logfile.txt`.
3. Open the file and type `DEBUG` on the first line.
4. Save and close the file.
5. Restart ArcGIS Server Cloud Builder on Amazon Web Services and repeat the steps that led to the problem you are trying to troubleshoot.

Debug-level logs will now be written to the log file. When ArcGIS Server Cloud Builder on Amazon Web Services writes to the log file, the file name will have the date appended to the end of it in the format YYYY-MM-DD.

A new log file is created each day ArcGIS Server Cloud Builder on Amazon Web Services is used. Be sure to clean up old log files.

Get your services and data onto a site

Strategies for data transfer to Amazon Web Services

Creating a GIS deployment with Amazon Web Services requires you to transfer some or all of your GIS data over the Internet to locations on the cloud. This topic lists some options of where you can store your data on the cloud and how you can transfer the data. It also discusses some factors that affect data transfer time.

Places to store the data

Once you create an EC2 instance running ArcGIS Server, you need to prepare to transfer your data to the cloud. There are several places you can store your data. All the following options incur charges from Amazon that are subject to change and that you should research before making your choice.

- **EBS volumes—**Amazon Elastic Block Store (EBS) volumes are virtual disk drives that you can attach to your EC2 instance to add more storage. In fact, a volume is always attached for you as part of the ArcGIS Server Amazon Machine Images (AMIs). You can configure the size of this attached volume when you build the site in ArcGIS Server Cloud Builder on Amazon Web Services. The ArcGIS server directories are configured on this drive, so when you publish services with the option to copy data to the server, the data goes onto this EBS volume. You can also create other directories on this volume to hold your data.

  Read Amazon's EBS overview

- **Amazon S3—**Amazon Simple Storage Service (S3) is an Amazon service designed specifically for data storage in the cloud. This storage option has the lowest potential for data failure or loss. You can use S3 as a place for data backup or as a middle ground for data transfer between your on-premises deployment and your EBS volumes. Also, any snapshots you create of your EBS volumes are stored on S3.

  Read Amazon's S3 overview

- **EC2 instance—**It's possible to transfer data directly onto your EC2 instance; however, if the instance is terminated, your data from the C: or root drive will be immediately lost. The ArcGIS Server AMI apportions a relatively small amount of space (60 GB on Windows) on the C: drive to discourage data storage on this drive. In contrast, attached EBS volumes such as the D: drive persist when the instance terminates and are a safer option for data storage.

  ⚠ **Caution:** Do not store GIS data or map caches on the C: or root drive of your EC2 instance in a production deployment.

Options for transferring data to the cloud

Transferring data from your on-premises deployment into the cloud takes time and, in some cases, coordination with your IT security staff. Exporting data to a location on the Internet (in other words, the cloud) is often not as fast or secure as the common data transfers that you do within your local network.

There are many strategies you can use to get data onto the cloud, but if you work with sensitive data, you'll want to make sure you coordinate with your IT staff to make sure your method is secure and approved by your organization. Following are some of your options:

- **Configure ArcGIS to copy the data when you publish a service**—You can configure ArcGIS so that whenever you publish a service, the data for that service is copied to the server. The data is packaged into a service definition file (.sd), transferred into the ArcGIS server uploads directory, and finally unpacked into the ArcGIS server input directory or a database you have registered with ArcGIS Server (as ArcGIS Server's Managed Database). Be aware that this can take a long time and result in the transfer of large amounts of data if you do not limit the extents and datasets used in your map or other resource.

  This option does not allow data to be shared between services, nor does it allow data synchronization between the cloud and your on-premises deployment.

- **Remote Desktop Connection copy and paste—**Windows Remote Desktop Connection allows file system redirection wherein your local drives can be mapped to the remote computer. While logged into your EC2 instance on Windows through Remote Desktop, you can open Windows Explorer and copy data from your local drives to your EBS volumes.

  To enable file system redirection, in the **Remote Desktop Connection** window, click the **Local Resources** tab and check the check box to make your drives available. The wording varies depending on which version of Windows you are using. In Windows 7, you have to click the **More** button to see the option to make drives available.

  If you choose to transfer sensitive data using Remote Desktop Connection, you should ensure that additional layers of security are in place. Older versions of Remote Desktop Connection have been shown to contain security vulnerabilities wherein a computer posing as the server can gain access to your data (sometimes known as man-in-the-middle attacks).

Note: Copy and paste can take a while to transfer data. Do not copy any other file or data before the paste procedure is complete. If you do, the paste terminates and you have to start over.

- **S3 client utilities—**Amazon S3 can be used as a middle ground for moving data from your on-premises deployment to your EBS volumes. To get data into S3, you can use the AWS Management Console or one of the many third-party apps that are designed for easily moving files between S3 and your own computers. Once your data is on S3, you can use the same utility on your EC2 instance to transfer data from S3 onto the instance.
- **Your own web server—**Any data available on the web through HTTP is accessible to your EC2 instance. If you have a web-facing server in your organization, you can place your data on it, then download the data from your EC2 instance. The advantage of this approach is that you can configure security on your web server to limit who can download the data and to encrypt the transaction through SSL.
- **FTP—**You can enable file transfer protocol (FTP) to upload files directly onto your EC2 instance. Beware that standard FTP does not encrypt information and sends passwords in clear text. To safely use FTP, you need to take additional security measures, such as encrypting your FTP sessions with SSL, limiting which users are allowed to transfer data to your instance through FTP, and disabling FTP after your initial data transfer. Some third-party products are designed to help you set up secure FTP connections.
- **AWS Import/Export—**If you need to transfer an enormous amount of data to Amazon, it may be faster and/or more cost effective to ship the data to Amazon on a portable storage device and pay Amazon to load the data directly into S3. Amazon offers this service as AWS Import/Export.

  If you consider using AWS Import/Export, you'll need to decide if it's appropriate for your organization's data sensitivity. Any time you put a device in the mail, you run the risk, however small, of the physical destruction or interception of your data. You can mitigate these risks by backing up and encrypting the data. If you still have concerns about whether AWS Import/Export is an appropriate choice for your data, contact Amazon directly.

Amazon works with many Solution Providers, some of whom provide data transfer, storage, and security solutions. See Find an AWS Solution Provider to understand whether one of these companies can help with your cloud strategy. Esri itself is one of these providers and offers various project and implementation services for deploying GIS in the Amazon cloud.

Factors that affect data transfer time

Performance of the above data transfer options can vary based on your physical proximity to the Amazon cloud, the time of day, and the quality of your connection to the Internet.

GIS datasets, especially imagery and map caches, can take large amounts of space and may need to be zipped before transfer, either to reduce the size of the file or to reduce the total number of files for more efficient transfer (especially in the case of map caches). Some S3 client utilities may place limits on the size of any one file you can transfer or the number of individual files you can store. Also, some zipping programs have limits on the amount of data that can be zipped. The zipping time and effort should be taken into account when you choose a data transfer option.

Finally, if using S3, be aware of the limitations on the number of buckets you can create and other restrictions on S3 buckets. Amazon lists these in Bucket Restrictions and Limitations.

Maintaining the integrity of data paths

Any time you move data to a new location, you need to be aware of any paths referencing the data that may also need to be updated. This is a concern with map documents, which may reference dozens of data layers at different paths.

Registering your Amazon EC2 data location with your ArcGIS server can help reduce the effort of fixing broken data paths after publishing. See Registering your data with ArcGIS Server using ArcGIS for Desktop.

Another option is to log in to your instance and use ArcMap to repair the out-of-date paths. ArcGIS for Desktop is included on the ArcGIS Server AMI so that you can easily make the repairs.

Another way to reduce the need to repair data connections is to use relative paths in your map documents and store your maps and data in a common folder.

Replace the default EBS volume on Windows

Any EC2 instance you create using the ArcGIS Server Amazon Machine Images (AMIs) has an EBS volume attached. The size of this volume is set at 100 GB by default when you build your site with ArcGIS Server Cloud Builder on Amazon Web Services. If you build your site with the AWS Management Console, the size is 10 GB. If you later decide to change the drive size, you can detach it and replace it with another drive.

Tip: If you just need more space, it might be easier to add another drive instead of replacing the existing one.

Log in to your instance and follow the steps below to replace the attached drive with one of a different size:

1. Stop the ArcGIS Server Windows Service.
   a. Open the Windows **Services** manager.
   b. Right-click the service **ArcGIS Server** and click **Stop**.
2. Perform the following substeps if you are working with an EC2 instance that has a SQL Server Express geodatabase installed along with ArcGIS Server. Otherwise skip this step.
   a. Start ArcCatalog.
   b. In the Catalog tree, expand **Database Servers**, then expand your database server instance.
   c. Right-click the database **egdb**, point to **Administration**, then click **Detach**.
   d. Right-click the database **geodata**, point to **Administration**, then click **Detach**.
3. Perform the following substeps if you are working with an EC2 instance that has a SQL Server Standard geodatabase installed along with ArcGIS Server. Otherwise skip this step.
   a. Start SQL Server Management Studio.
   b. Log in to your SQL Server instance using operating system authentication.
   c. Expand the **Databases** folder.
   d. Right-click the database **egdb**, point to **Tasks**, then click **Detach**.
   e. Leave the default detachment options and click **OK**.
   f. Repeat the above two substeps to detach the database **geodata**.
   g. In the **Object Explorer** window of SQL Server Management Studio, right-click your database instance (it is probably called localhost) and click **Stop**.
   Click **Yes** when prompted to confirm.
   h. Close SQL Server Management Studio.
4. Copy all the contents of the original attached volume into a folder on your root system (Linux) or C drive (Windows).
   For example, on Windows, you could create a new folder `C:\temp`, and copy all folders from `D:\` into `C:\temp`.

   If the root system or C drive is not big enough, you must copy the files to a different location.
5. In Windows, click **Start** and type `diskmgmt.msc` in the search box. Then press Enter.
6. Right-click your D drive, click **Delete Volume**, and click **Yes** when prompted to confirm.
7. Log in to the AWS Management Console and display the EC2 page corresponding to your Amazon region.
8. Click **Volumes** and find the original attached volume.
9. Right-click the volume and click **Detach Volume**. Click **Yes, Detach** when prompted.
   Detaching may take some time. If the volume does not detach, right-click the volume and choose **Force Detach**.
10. Create a new EBS volume of your desired size and attach it to your instance. This process is described in Add disk space to your Windows site and Add disk space to your Linux site.
11. Copy all the original contents of your volume back onto the new volume at the same paths used before.
    For example, if you copied the contents from `D:\` into `C:\temp`, move them back to `D:\`.
12. Grant read and write permission for the ArcGIS Server account for all the folders on your new volume.
13. In the AWS Management Console, click **Instances**, right-click your instance, and click **Reboot**. Wait several minutes for your instance to reboot.

14. Perform the following substeps if you are working with an EC2 instance that has a SQL Server Express geodatabase installed along with ArcGIS Server. Otherwise, skip this step.
    a. In ArcCatalog, expand **Database Servers** and, if necessary, double-click your database server instance to connect to it.
    b. Right-click your database server instance and click **Attach**.
    The **Attach Geodatabase** dialog box opens.
    c. Browse to the location of your **egdb** database, which is probably in `D:\data\geodatabase\egdb.mdf` if you configured your new volume as the D drive. Click **OK**.
    d. Repeat the above two substeps to attach your **geodata** database. The `.mdf` file is probably in `D:\data\geodatabase\geodata.mdf`.
15. Perform the following substeps if you are working with an EC2 instance that has a SQL Server Standard geodatabase installed along with ArcGIS Server. Otherwise skip this step.
    a. Log in to SQL Server Management Studio, right-click the **Databases** folder and click **Attach**.
    b. Click **Add** on the **Attach Databases** dialog box.
    c. Browse to the location of your **egdb** database, which is probably in `D:\data\geodatabase\egdb.mdf` if you configured your new volume as the D drive. Then click **OK**.
    d. Repeat the above two substeps to add your **geodata** database. The `.mdf` file is probably in `D:\data\geodatabase\geodata.mdf`.
    e. Click **OK** to attach both databases.
    f. Close SQL Server Management Studio and your Windows Remote Desktop connection.
    g. In the AWS Management Console, click **Instances**, right-click your instance, and click **Reboot**. Wait several minutes for your instance to reboot.
16. Preview the sample service, or publish a new service and verify that your site is working as expected.
17. Return to the **Volumes** page of the AWS Management Console.
18. Right-click your original volume (the one you had detached) and click **Delete Volume**.

Replace the default EBS volume on Linux

Any EC2 instance you create using the ArcGIS Server Amazon Machine Images (AMIs) has an EBS volume attached. The size of this volume is set at 100 GB by default when you build your site with ArcGIS Server Cloud Builder on Amazon Web Services. If you build your site with the AWS Management Console, the size is 10 GB. If you later decide to change the drive size, you can detach it and replace it with another drive.

Tip: If you just need more space, it might be easier to add another drive instead of replacing the existing one.

Follow the workflow below to replace the attached drive (referred to as Volume A) with one of a different size (Volume B). If you need detailed steps on how to create, attach, and detach EBS volumes using the AWS Management Console, see the Amazon Web Services documentation.

1. Stop your site using ArcGIS Server Cloud Builder on Amazon Web Services (not the AWS Management Console).
2. Use the AWS Management Console to create and attach Volume B with the size you want. You attach it alongside Volume A at this point. When you attach, you cannot specify `/dev/sdx` as the device because this is already in use by Volume A.
3. Start your site using Cloud Builder (not the AWS Management Console).
4. Log in to your instance. For example, you can SSH into your instance:
   ```
   ssh –i <your key pair file> ubuntu@<Public DNS of your EC2 instance>
   ```
   Make sure you have opened port 22 in your instance's Amazon security group before you attempt this.
5. Stop ArcGIS Server as the arcgis user. For example:
   ```
   sudo -u arcgis /arcgis/server/stopserver.sh
   ```
6. If your ArcGIS Server instance also has a PostgreSQL geodatabase running on the same instance, stop the PostgreSQL database by running:
   ```
   sudo -u postgres -i pg_ctl stop
   ```
7. Copy all the data from Volume A onto Volume B. In the following steps /mnt/data-store is used as the attach location and /dev/sdf is the device field.

   Caution: This procedure assumes you are mounting an empty volume. If you are mounting a volume that already has data on it, don't use mkfs before mounting the volume; otherwise you will format the volume and delete the existing data.

   ```
   sudo mkfs -t ext4 /dev/sdf
   sudo mkdir /mnt/data-store
   sudo mount /dev/sdf /mnt/data-store
   sudo cp -rp /gisdata/* /mnt/data-store
   ```

8. Stop your site using Cloud Builder.
9. Log in to the AWS Management Console and display the EC2 page corresponding to your Amazon region.
10. Using the AWS Management Console, detach both Volume A and Volume B from the instance.
11. Once you are certain that the volumes are detached, use the AWS Management Console to attach Volume B back onto your instance. Use /dev/sdx as the device, or your instance will not start properly.
12. Once you are certain that Volume B has been attached, use Cloud Builder to start your site.
13. Using the AWS Management Console, delete Volume A.

Add disk space to your Windows site

Any EC2 instance you create using the ArcGIS Server Amazon Machine Images (AMIs) has an EBS volume attached. The size of this volume is set at 100 GB by default when you build your site with ArcGIS Server Cloud Builder on Amazon Web Services. If you build your site with the AWS Management Console, the size is 10 GB. If you need more space, you can create additional volumes and attach them to your EC2 instance.

The steps below explain how to create a new EBS volume and attach it to your EC2 instance.

1. Open the AWS Management Console and display the EC2 page corresponding to your Amazon region.
2. On the left menu, click **Volumes**.
3. Click the **Create Volume** button.
4. Type a value for **Size**. Remember that you will be charged by Amazon for the size you choose, and you cannot change this size once you've created the volume.
5. Choose a value for **Availability Zone**. You need to match the zone of the EC2 instance to which you will attach the volume.

   Tip: If you don't know the zone of the instance to which you want to attach the volume, return to the AWS Management Console, click **Instances**, click your instance, then examine the **Description** tab to find the zone.

6. Optionally, choose a snapshot. Snapshots are a way of preformatting a disk with certain datasets. To create a new empty disk, choose **No Snapshot**.
7. Click **Yes, Create** to create the volume. This can take a while. When the volume is ready, you'll see its status change from **creating** to **available** in AWS Management Console.

   You've successfully created your EBS volume, but it's not attached to your EC2 instance yet. The remaining steps explain how to format the disk and attach it to your instance.

8. Right-click the volume and click **Attach Volume**.
9. Choose your instance from the drop-down list. Remember that you can only attach your volume to instances that reside in the same availability zone.
10. Type the four-letter code for an available Windows device, such as `xvdf`, and click **Attach**.
    It may take a few minutes to attach your volume. You can see the status in the Attachment Information column in the AWS Management Console list of volumes. After a while, **attaching** changes to **attached.**
11. Log in to your EC2 instance using Windows Remote Desktop.
12. Click **Start** and type `diskmgmt.msc` in the search box. Then press Enter.
    At this point, a message may appear about choosing a partition style. If this happens, you can go to Step 16 and continue from there.
13. In the bottom section of the **Disk Management** console, scroll through the list of disks until you find an unallocated disk matching the size of the volume you just attached.
14. Right-click the disk and click **Online**.
15. Right-click the disk again and click **Initialize Disk**.
16. Ensure your new disk is checked and choose a partition style (if you are unsure which partition style to use, choose **MBR**). Then click **OK**.
    Now you can choose to partition all or part of your disk and assign the partition(s) a drive letter.
17. Right-click the box containing the label **Unallocated** (which represents your disk's unallocated space) and click **New Simple Volume**.
18. Follow the prompts in the **New Simple Volume** wizard to select the amount of space you want to assign to a drive and the drive letter you want to use. If you want the whole disk to be available as one standard drive, you can leave most of the defaults in this wizard. You may want to change **Volume label** to give the drive a name.

Add disk space to your Linux site

The ArcGIS for Server AMI comes with an attached EBS volume. The size of this volume is set at 100 GB by default when you build your site with ArcGIS Server Cloud Builder on Amazon Web Services. If you build your site with the AWS Management Console, the size is 10 GB. If you need more space, you can create additional volumes and attach them to your EC2 instance.

The steps below explain how to create a new EBS volume and attach it to your EC2 instance.

1. Open the AWS Management Console and display the EC2 page corresponding to your Amazon region.
2. From the left menu, click **Volumes**.
3. Click the **Create Volume** button.
4. Type a value for **Size**. Remember that you will be charged by Amazon for the size you choose, and you cannot change this size once you've created the volume.
5. Choose a value for **Availability Zone**. You need to match the zone of the EC2 instance to which you will attach the volume.
6. If you don't know the zone of the instance to which you want to attach the volume, return to the AWS Management Console, click **Instances**, click your instance, then examine the **Description** tab to find the zone.
7. Optionally, choose a snapshot. Snapshots are a way of preformatting a disk with certain datasets. To create a new empty disk, choose **No Snapshot**.
8. Click **Yes, Create** to create the volume. This can take a while. When the volume is ready, you'll see its status change from **Creating** to **Available** in the AWS Management Console.
9. You've successfully created your EBS volume, but it's not attached to your EC2 instance yet. The remaining steps explain how to format the disk and attach it to your instance.
10. Right-click the volume and click **Attach Volume**.
11. Choose your instance from the drop-down list. Remember that you can only attach your volume to instances that reside in the same availability zone.
12. Use the Device field to specify how to expose the volume to the instance (/dev/sdb through /dev/sdz). Click **Attach**.
    It may take a few minutes to attach your volume. You can see the status in the Attachment Information column in the AWS Management Console list of volumes. After a while, **Attaching** changes to **Attached.**
13. To make a volume available to the Linux operating system, use the following command (the following command is an example of how to create an ext3 file system and mount it as /mnt/data-store):

```
ssh –i <your key pair file> ubuntu@<Public DNS of your EC2 instance>

sudo yes | mkfs –t ext3 /dev/sdh

Caution: This procedure assumes you want to mount an empty volume. If you're mounting a volume that already has data on it, don't use mkfs before mounting t

sudo mkdir /mnt/data-store
sudo mount /dev/sdh /mnt/data-store
```

Any data written to this file system is written to the Amazon EBS volume and is transparent to applications using the device.

Note: To enable the instance to reconnect to an Amazon EBS volume on reboot, add the device to the fstab or create a script that automatically mounts the volume on startup.

Move EBS volumes between Windows instances

The ArcGIS for Server AMIs use a separate Amazon Elastic Block Storage (EBS) volume for data storage. Therefore, the ArcGIS and database management system (DBMS) software are stored in a different location from the data, and you can move data from one instance to another by creating a snapshot of an existing EBS volume, creating a volume from the snapshot, and attaching the new volume to a different ArcGIS Server on Amazon Web Services site.

You would do this if you want to move your existing data and services to a site created from a newer ArcGIS for Server AMI, or you want to have multiple sites with the same data.

The following workflow explains how to move a volume from one Windows ArcGIS Server on Amazon Web Services site to another.

Guidelines for moving EBS volumes

Be aware of the following when moving volumes between Windows instances:

- You must stop all your services and ArcGIS for Server on your source instance before moving volumes.
- The ArcGIS for Server directory structure on the new volume must be the same as it was on the old volume. This is accomplished by making a snapshot of your existing volume.
- ArcGIS for Server relies on the volume being mapped to the D drive on Windows.
- The operating system must be the same when moving volumes between instances. For example, you cannot move a volume from a Windows instance to a Linux instance.
- Block device names must be unique; therefore, when you attach the new volume to the target instance, be sure to give it a different name from the existing one.
- To move an EBS volume that contains geodatabases in SQL Server Standard or Express, you must first detach the databases from the source SQL Server instance. After you move the EBS volume to another ArcGIS Server on Amazon Web Services instance, you must attach the databases to the new SQL Server instance before you can use them. You cannot attach a newer release database to an older release SQL Server instance.

  Note: This workflow does not apply to SQL Server on Amazon Relational Database Services (RDS).

Prepare source volume

Before you can create a snapshot of the volume, you must stop your services, stop ArcGIS for Server, detach SQL Server databases (if using them), unmount the volume, and stop the existing ArcGIS Server on Amazon Web Services site.

1. In ArcGIS Server Manager or ArcGIS for Desktop, stop all services running on your GIS server.
2. Make a remote desktop connection to the source instance.
   See Administer your Amazon EC2 instance with Windows Remote Desktop Connection for instructions.
3. Stop ArcGIS for Server.
   a. Open the Windows Services dialog box.
   b. Right-click ArcGIS Server in the Services list and click **Stop**.
4. If you have geodatabases in SQL Server, detach them.
   - If your site is licensed with an ArcGIS for Server Enterprise license, use SQL Server Management Studio to detach the databases.
   - If your site is licensed with an ArcGIS for Server Workgroup license, use ArcGIS for Desktop to detach the geodatabases from the database server.

     Note: ArcGIS for Desktop must be licensed for you to use it.

Create snapshot of data volume

Once the source instance is ready, create a snapshot of the volume you want to move. Creating a snapshot basically makes a copy of the content of the existing volume.

1. Log in to the Amazon Web Services Management Console.
2. Click the **EC2** link and display the page corresponding to your Amazon region.

3. Click **Volumes**.
4. Right-click the volume you want to move and click **Create Snapshot**.
5. Type a name and description for your snapshot that will help you find it in the snapshot list.
6. Click **Yes, Create**.

Create target instance

If the target instance has not been created yet, create it now.

Once you have a target instance, record the following information about it:

- The instance ID
- The instance zone (for example, us-east-1d)
- The volume ID of the volume you will be replacing
- The block device name of the existing volume on the target site

Block device names must be unique, so when you attach the new volume to the target site, be sure to give it a different name from the existing one.

By default, the ArcGIS for Server AMIs use xvdg for the EBS volume name.

Create volume from snapshot

You must create a volume from the snapshot you made of your source volume so you can attach it to the target site. The volume you create can be the same size or larger than the original volume, but not smaller. Do the following in the Amazon Web Services Management Console to create a new volume from the snapshot:

1. Navigate to the EC2 page for your region and click **Snapshots**.
2. Right-click the snapshot you created from the source volume and click **Create Volume from Snapshot**.
3. Specify the size of your new volume.
   The new volume can be the same size or larger than the original volume, but not smaller.
4. From the drop-down list, choose the instance zone you recorded for the target instance.
5. Click **Yes, Create**.
6. Click **Volumes**.
   You should see your new EBS volume being created. Record the volume ID for future use.

   Note: It may take a few minutes to create the volume.

Attach new volume to target site

Use the Amazon Web Services Management Console to attach the new volume to your target site. Then, connect to the target site and bring the volume online.

1. In Amazon Web Services Management Console, right-click the new EBS volume and click **Attach Volume**.
2. From the drop-down list, choose the instance ID of the target instance.
3. Type a name for the new block device.
   Remember, the device name must be unique on that instance.
4. Click **Yes, Attach**.

   Bring the new volume online after you attach it.
5. Make a remote desktop connection to the target instance and stop ArcGIS for Server.
   See Administering your Amazon Web Services instance with Windows Remote Desktop Connection for information on making a remote desktop connection to a Windows instance.
6. Open the Microsoft Server Manager on the target instance.
7. Expand the Storage node and click **Disk Management**.
   Notice that the new volume is offline.

8. Right-click the disk volume in the Graphical View and click **Online**.

Replace volume on target site

Remove the empty volume that came with the new ArcGIS Server on Amazon Web Services site, remap your newly attached volume to the D drive, and attach geodatabases (if needed).

⚠ **Caution:** Be sure to remove the empty volume, not the one that you just attached to the new site.

To remove the volume, you must detach geodatabases (if applicable) and stop the services that are pointing to the volume to be detached.

1. If you are using SQL Server Standard, start Management Studio, detach the geodatabases from the SQL Server instance, then stop the SQL Server instance.
2. If you are using SQL Server Express, do the following to detach the geodatabases and stop the database server:
   a. Start ArcGIS for Desktop.
   b. Right-click each geodatabase on the database server, one at a time, and click **Detach**.
   c. Right-click the database server and click **Disconnect** .
   d. Right-click the database server and click **Stop**.
3. Open the Services dialog box and stop the ArcGIS for Server service.

   Next, unmount the old volume.
4. In Microsoft Server Manager on the target instance, expand the Storage node and click **Disk Management**.
5. Right-click the volume that came with the new site and click **Change Drive Letter and Paths**.
6. Choose the drive to unmount and click **Remove**.
7. Click **Yes** when prompted to confirm that you want to remove the drive.

   ArcGIS Server relies on the data store files being on the D drive; therefore, you must remap the new volume to the D drive.
8. Right-click the newly attached volume and click **Change Drive Letter and Paths**.
9. Click **Change**.
10. Choose **D** from the drop-down list and click **OK**.
11. When prompted to confirm, click **Yes**.
12. Close all applications and disconnect from the target instance.

    Stop your site and detach the old volume from the target instance.
13. Stop your site using ArcGIS Server Cloud Builder on Amazon Web Services (not the Amazon Web Services Management Console).
14. Log in to Amazon Web Services Management Console.
15. Click **Volumes**.
16. Take note of the volume ID of the volume you want to delete; you will need this information to identify the volume once it is detached.
17. Right-click the empty volume and click **Detach Volume**.
18. Click **Yes, Detach** to confirm.
19. If you no longer need the detached volume, right-click it and click **Delete Volume**.

    ⚠ **Caution:** You cannot restore a volume once it is deleted; therefore, be sure you are deleting the correct volume.
20. Click **Yes, Delete** to confirm.
21. Start your site using ArcGIS Server Cloud Builder on Amazon Web Services (not the Amazon Web Services Management Console).

Attach SQL Server databases

If you have geodatabases in SQL Server, you can attach them now that you have remapped the replacement drive.

Follow these steps if your instance is licensed with an ArcGIS for Server Enterprise license:

1. Log in to the target instance.
2. Start SQL Server Management Studio.
3. Attach your existing, populated geodatabases from their location on the newly attached volume.
   See the section "Attaching the database and transaction log files to the new instance" in Moving geodatabases in SQL Server to an ArcGIS Server for Amazon Web Services instance for instructions.

Follow these steps if your instance is licensed with an ArcGIS for Server Workgroup license:

1. Start ArcGIS for Desktop on the new instance.
   Remember to authorize ArcGIS for Desktop on the new instance. This is done from the ArcGIS Administrator.
2. Connect to the database server on the new instance.
3. Attach the geodatabases from the new volume to the database server.

Once all your data is accessible, you can restart or republish your services.

Restart services

You can restart the services that got moved with your volume. For geodata services, though, you must first reset the replica properties to point to the geodata service on the new ArcGIS Server on Amazon Web Services site.

Use **Replica Manager** to reset the replica target.

1. Start ArcMap or ArcCatalog.
2. Create a GIS Server connection to the new ArcGIS Server on Amazon Web Services site.
3. Connect to your parent replica geodatabase.
4. Right-click the geodatabase, point to **Distributed Geodatabase**, and click **Manage Replicas**.
   The **Replica Properties** dialog box opens.
5. Right-click the replica in the list and click **Properties**.
6. Click the **Advanced** tab.
7. Set a new relative replica connection by browsing to your geodata service on the new ArcGIS Server on Amazon Web Services site.
8. Click **OK** to close the **Replica Properties** dialog box.
9. Close the **Replica Manager**.
10. Restart your services from ArcGIS for Desktop or ArcGIS Server Manager.

Build map and image caches on Amazon Web Services

ArcGIS Server caches in Amazon EC2

You can create, store, and use ArcGIS map, image, and globe service caches on Amazon EC2 in much the same way that you do on premises. For small caches, you'll notice little difference in your caching workflow, and you can even transfer existing caches to the cloud with relative ease. For larger caches, you'll need to decide whether to transfer an on-premises cache to the cloud or create the cache from scratch on the cloud.

Caching on the cloud offers you access to powerful machines that you can use for just as long as it takes to make the cache. When you're done, you can terminate the instances and incur no further charges for these machines.

Should you move your existing cache to the cloud?

Moving files to Amazon Web Services is not as easy as moving files from one machine to another within your network. As detailed in Strategies for data transfer to Amazon Web Services, you need to decide on a method of file transfer. You also need to appropriately secure that method of file transfer in accordance with the sensitivity of your data, then transfer the files. The speed of the transfer is limited by your Internet connection and often takes much longer than a file transfer within your own network.

The above issues are compounded with map caches since, in the exploded format, these can consist of thousands and sometimes millions of files. Sometimes, the large number of files involved in the transfer can be more of a burden than the total size. If you know that the cache will be transferred between machines, on the cloud or on premises, you should use the compact cache format whenever possible.

Before you move an exploded cache to Amazon, you might want to try transferring just a small portion and extrapolate an estimate of how long it would take to move the entire cache. Similarly, you might re-create a small piece of the cache using an EC2 instance to determine whether re-creating the cache on the cloud would be a faster option. Although regenerating your entire cache may seem like an unattractive prospect, an EC2 instance can often generate tiles much faster than you can copy exploded tiles to EC2.

If you have an extraordinarily large cache that you have invested weeks or months of time to build, you might consider mailing the cache to Amazon and paying them to load the cache directly into the cloud. This type of service is available through AWS Import/Export.

Cache creation in Amazon EC2

Creating an ArcGIS map, image, or globe service cache in the Amazon Elastic Compute Cloud (EC2) differs from caching outside the cloud in several ways:

- You have a number of different instance sizes and prices at your disposal.
- You have a variety of locations in the cloud where you can place the cache.

This topic discusses the above factors in more detail.

Choosing an instance size and price

Amazon EC2 offers a variety of instance sizes and specifications. Each has its own price per hour of usage. The larger instances, especially those with a lot of memory, can generate tiles very quickly. The smaller instances generate tiles more slowly but have a lower cost.

You can create your cache on an attached Amazon Elastic Block Store (EBS) volume using a powerful instance. When the caching completes, you can detach the EBS volume and attach it to your regular instance (which may be smaller and less expensive). You can then terminate the powerful instance that you used for caching. In this way, you can use the power of the cloud to cache while not committing to a relatively expensive instance for any longer than necessary.

You may need to make a decision between economy and speed. Using a low power instance with a low cost per hour is not always the most economical choice, as the total cost of the cache is dependent on the number of hours spent creating tiles. On the other hand, the most powerful instances may also yield a higher total cost of the cache: even though you spend fewer hours caching, you pay a higher price per hour.

Using a small test cache (perhaps the size of a medium-sized city) as well as a custom Amazon Machine Image (AMI) or site template, you can perform relatively inexpensive tests with different instance types to find out which is most economical for your cache.

Powerful EC2 instance types are well suited to scheduled cache updates, since many update workflows are time sensitive.

Choosing the number of map service instances to use when caching

Each EC2 instance has a certain number of virtual CPU cores. This number is visible when you choose the instance type from the **Launch Instance** wizard. The number of cores can help you determine how many instances of the CachingTools geoprocessing service to devote toward your caching. Using too many service instances will overwork your CPUs, while too few service instances will leave your CPUs underutilized.

Although the best number may be reached with some trial and error, a good starting point is to allow a maximum of n + 1 instances of the CachingTools service, where n is the number of virtual cores on a single EC2 instance in your site.

Manual scaling and auto scaling

When building a large cache, you might be tempted to set up auto scaling triggers that automatically increase the number of EC2 instances working on the cache as the CPU usage increases. However, auto scaling is better suited to handling unexpected spikes in traffic. When creating caches, you already know that you will need a great amount of computing power; therefore, it makes more sense to launch all your needed instances before you build the cache, rather than waiting for them to launch sequentially via auto scaling triggers.

Deciding where to place the cache

As described in Strategies for data transfer to Amazon Web Services, there are several types of locations where you can place your data. When you first create the cache, you'll write it to an EBS volume that's attached to your EC2 instance. This volume is attached at the time you build your site, and it's a good place to put the cache if the volume is large enough. If the volume is too small, you need to create and attach another volume and register a server cache directory on it.

Do not build a cache on the C drive of your EC2 instance. If the instance is ever terminated, the cache will be lost.

Ultimately, you might want to move or place a copy of the cache onto Amazon Simple Storage Service (Amazon S3). If you're just interested in keeping a backup on Amazon S3, you can create an EBS snapshot. A snapshot effectively backs up your drive on Amazon S3, and you can quickly use the snapshot to create a new EBS volume if your existing volume fails for any reason.

You can also serve the tiles from Amazon S3 and access them as a custom tile layer using a JavaScript, Flex, or Silverlight application. The advantage of this is that your tiles do not depend on a running service, and you can optionally use Amazon CloudFront to speed the delivery of tiles across the Internet to all parts of the world. If you want to move tiles to Amazon S3 for this purpose, you can transfer the tiles from your EBS volume using either the Amazon Web Services APIs or a third-party front-end application for Amazon S3. You could also do this if you created the cache outside the cloud.

Geodatabases on Amazon Web Services

Geodatabases and ArcGIS Server on Amazon Web Services

Geodatabases store spatial and nonspatial data. Geodatabases on Amazon Web Services (AWS) instances are intended to be used to store data you are serving from your ArcGIS Server AWS instance. You can use file, workgroup, or enterprise geodatabases.

Note: Geodatabases in AWS instances are not intended to be accessed directly from on-premises ArcGIS clients.

Options are available in the ArcGIS Server Cloud Builder on Amazon Web Services to include enterprise or workgroup geodatabases with your site. You can either load data directly to these geodatabases (see Strategies for data transfer to Amazon Web Services and Strategies for loading data into a geodatabase on Amazon Web Services for suggestions on how to get your data to your geodatabases in the cloud), or data stores can be set up so data can be copied at time of publication or synchronized to these geodatabases from your on-premises ArcGIS for Desktop installation.

To understand the data store options available when publishing data, see the following ArcGIS for Server topics:

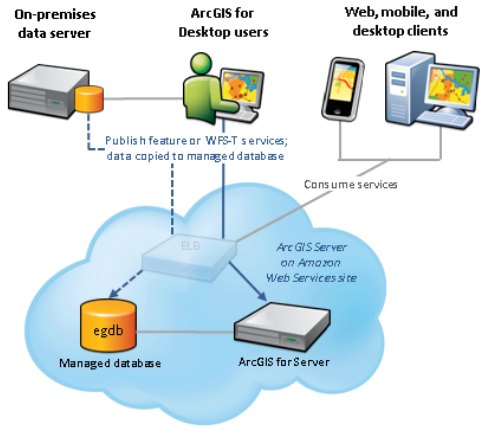
- Sections "If the publisher's machine and the server are working with different databases" and "If the publisher's machine and the server are working out of different folders" in About registering your data with the server
- Copying data to the server automatically when publishing

The following sections describe each type of geodatabase used with ArcGIS Server on Amazon Web Services.

Enterprise geodatabases

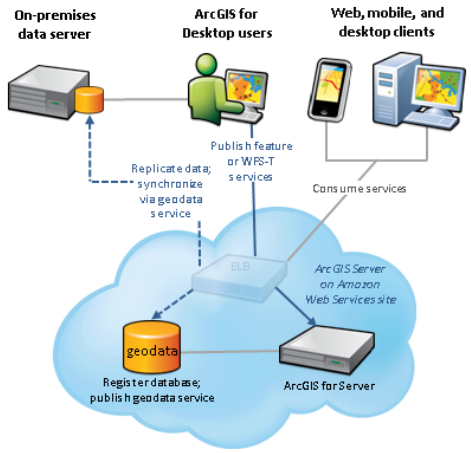
When you launch your site using the ArcGIS Server Cloud Builder on Amazon Web Services application and an ArcGIS for Server Enterprise license, two geodatabases are created automatically: egdb and geodata.

The egdb geodatabase is registered as an ArcGIS Server managed database. When you publish feature or WFS-T services to a site that has a registered managed database, the data can be copied from the geodatabase that contains the map source data (publisher geodatabase) to the registered database (server geodatabase). This data is dependent on the service; when you delete the feature or WFS-T service, the data is deleted from the egdb geodatabase.



The geodata geodatabase is intended for use as a replicated geodatabase. You can register the geodata geodatabase with ArcGIS Server. When you do so, designate the geodata geodatabase as a server database connection that is not the same as your publisher database connection (your on-premises enterprise geodatabase) and create a geodata service. You can replicate data from your on-premises enterprise geodatabase to the geodata geodatabase through the geodata service.

When you publish a feature or WFS-T service that includes the replicated data, edits made to the data through the feature service can be synchronized with the geodata service, updating the data in your on-premises enterprise geodatabase. Similarly, you can continue to edit your on-premises data and use the geodata service to synchronize those changes to the data in the geodata geodatabase.



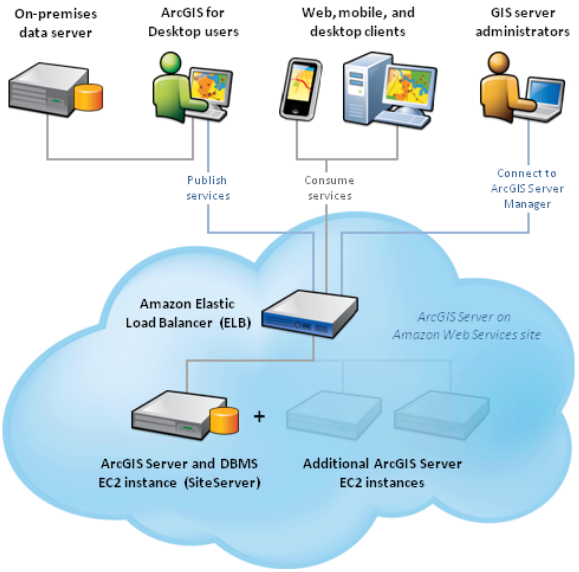
See About registering your data with the server in the ArcGIS help for information on registering databases with ArcGIS Server.

If you have an ArcGIS for Server Enterprise license, you can use any of these Amazon Machine Images (AMIs):

- ArcGIS for Server (Ubuntu Linux)
- ArcGIS for Server (Windows)
- Your own template that was created from the ArcGIS for Server Ubuntu Linux or Windows AMIs

You can choose to include a DBMS that contains the egdb and geodata geodatabases on the same instance as ArcGIS for Server. Alternatively, if you use PostgreSQL or SQL Server Standard, you can have the DBMS and geodatabases created on a separate instance that is part of the ArcGIS Server on Amazon Web Services site.

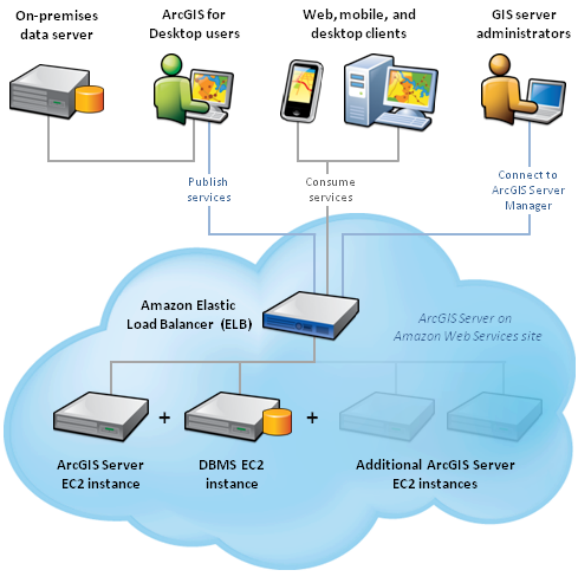
The following diagram shows an ArcGIS Server on Amazon Web Services site with ArcGIS for Server and the DBMS on the same instance, and two additional ArcGIS for Server AWS instances that come online when CPU usage exceeds a specified threshold and go offline when CPU usage drops below a specified threshold:



If you anticipate that your services will require a lot of processing in the DBMS—for example, if you publish many geoprocessing services that use data in the geodatabases or you have editable feature services—you will likely want to create the DBMS on a separate instance.

Note: If you use an Amazon Relational Database Service for SQL Server, the geodatabases are always created on a separate AWS instance.

The following diagram shows an ArcGIS Server on Amazon Web Services site with ArcGIS for Server and the DBMS on separate AWS instances, with two additional ArcGIS for Server AWS instances available when CPU usage exceeds a specified threshold.



See Geodatabases in PostgreSQL included with ArcGIS Server for Amazon Web Services and Geodatabases in SQL Server included with ArcGIS Server for Amazon Web Services for more information on using an enterprise geodatabase with your ArcGIS Server on Amazon Web Services site.

Workgroup geodatabases

If you have an ArcGIS for Server Workgroup license, you can use the ArcGIS Server Cloud Builder on Amazon Web Services application to launch an instance of the ArcGIS for Server for Windows AMI or a template created from this AMI and include an instance of SQL Server Express on the ArcGIS for Server instance.

The SQL Server Express instance (database server) comes with two geodatabases already created: egdb and geodata. As with the enterprise instance, the egdb geodatabase is registered as a database connection that is different from the publisher database connection. When you publish feature or WFS-T services to a site that has a geodatabase registered in this way, the data is copied from your source to the registered geodatabase in the cloud. The geodata geodatabase is intended for use as a replicated geodatabase. You can register the geodata geodatabase as a database connection that is different from the publisher database connection and create a geodata service from it. Through the geodata service, you can synchronize data from your on-premises enterprise or workgroup geodatabase to the geodata geodatabase.

Workgroup geodatabases only support Windows authentication. Two operating system users are automatically added to the SQL Server Express instance as server administrators: Administrator and ArcGIS. Both logins are dbo in the egdb and geodata geodatabases.

See Workgroup geodatabases included with ArcGIS Server for Amazon Web Services for more information on using workgroup geodatabases with an ArcGIS Server on Amazon Web Services instance.

File geodatabases

See File geodatabases used with ArcGIS Server for Amazon Web Services.

Related topics

Geodatabases in PostgreSQL included with ArcGIS Server for Amazon Web Services
Enterprise geodatabases in SQL Server in ArcGIS Server on Amazon Web Services
Workgroup geodatabases included with ArcGIS Server for Amazon Web Services

Strategies for loading data into a geodatabase on Amazon Web Services

You can replicate data using a geodata service published from a geodatabase in the cloud, publish data to a registered database, or move your source data to Amazon Web Services then load the data into your geodatabase.

Replicate

You can replicate data from an on-premises geodatabase to a geodatabase in your ArcGIS Server on Amazon Web Services instance.

The geodata geodatabase that is included with your ArcGIS Server on Amazon Web Services instance is intended for this purpose. Register the geodata geodatabase as a replicated database and create a geodata service from it. Then create a replica of your local data, replicating it to the geodata service. As edits are made, you can synchronize changes through the geodata service. See Using a geodata service and a connected replica for more information.

Publish

If you created your site to include a database management system, the egdb geodatabase that is included in the site is registered automatically with your ArcGIS for Server instance. When you publish a feature or WFS-T service, a copy of the data will be moved to the egdb geodatabase and the service will use the data in the egdb geodatabase as its source data.

When you delete the feature or WFS-T service, the corresponding data is deleted from the egdb geodatabase.

Load data you moved to the cloud

You can also physically move data to your geodatabases in your ArcGIS Server on Amazon Web Services instance.

First, you must move the data to Amazon Web Services. Once your source data has been moved to your instance, you must log in to your ArcGIS for Server instance and use the tools provided in ArcGIS for Desktop to load the data into your geodatabase.

From ArcCatalog or the Catalog window in ArcMap, you can do the following:

- Copy data from a source geodatabase, such as a file geodatabase you transferred to Amazon Web Services, and paste it into your geodatabase. See Copying feature datasets, classes, and tables to another geodatabase for more information.
- Create empty feature classes or tables, then use the Simple Data Loader or Object Loader wizard to add data to them. See About loading data into existing feature classes or tables for information on the two wizards.
- Create an empty feature class or table and use a source XML record set document that you uploaded to Amazon Web Services to load data to the feature class or table.
- Use geoprocessing tools to import source shapefiles, coverages, feature classes, tables, rasters, or XML workspace documents that you moved to Amazon Web Services. See An overview of importing datasets for more information.
- Attach a workgroup geodatabase you moved from another ArcGIS Server on Amazon Web Services instance.

If you are using enterprise geodatabases, you also could connect to the DBMS on your ArcGIS Server on Amazon Web Services instance and restore a dump file from an existing geodatabase in PostgreSQL to your PostgreSQL database cluster or attach a database to SQL Server on Amazon EC2 that you detached from another SQL Server instance.

For information on getting source data onto an Amazon Web Services instance, see Strategies for data transfer to Amazon.

Optional enterprise geodatabase management

The enterprise geodatabases that are created with your ArcGIS Server on Amazon Web Services instance do not require any configuration before you can access them. However, you can perform some administration tasks, such as creating another geodatabase, creating new roles or groups, or moving data between instances, if your workflow requires it. Listed here are some of these optional advanced administration tasks, tools used to perform them, and links to documentation with further instructions.

Manage logins

Your ArcGIS Server on Amazon Web Services instance comes with default roles to administer geodatabases in PostgreSQL and Amazon Relational Database Service for SQL Server and load spatial data to them. (See Geodatabases and ArcGIS Server on Amazon Web Services for a list of these default roles.) You may require additional roles to manage your spatial data. For example, you may want to add a login that has read-only privileges in the database, require a login with a name other than the default name provided with the ArcGIS for Server Amazon Machine Image (AMI), or want additional data owner logins in the database.

If you want additional logins that can own data, you can use the Create Database User geoprocessing tool or a Python script to create a new login.

If you want to create logins that can only view data, use SQL or your database management system (DBMS) tools to create the logins. See PostgreSQL or Microsoft SQL Server's documentation for more information.

Create additional geodatabases

If you choose to include enterprise geodatabases with your ArcGIS for Server instance, the instance comes with two geodatabases: egdb and geodata. To create additional geodatabases in PostgreSQL or SQL Server—for example, if you want a geodatabase with a different name or you require additional geodatabases for different services or applications to access—you can use the Create Enterprise Geodatabase geoprocessing tool. To create another geodatabase in Amazon Relational Database Service for SQL Server, use the Enable Enterprise Geodatabase tool.

After the additional geodatabase is created, connect to it as an administrator and add any schemas or users you require.

To create additional workgroup geodatabases, connect to the database server from ArcGIS for Desktop as a server administrator and create a geodatabase. See Create additional workgroup geodatabases for more information.

Install PostGIS

Esri supports the use of the PostGIS geometry type in enterprise geodatabases in PostgreSQL. You can download a supported version of PostGIS and install it on the ArcGIS Server on Amazon Web Services instance where PostgreSQL resides (either the same instance as ArcGIS for Server or a separate instance). See the Database System Requirements page on the ArcGIS Resource Center for the version of PostGIS supported with the AMI version you are using.

To use PostGIS geometry, the database that contains your geodatabase must be enabled to use PostGIS. Since PostGIS is not part of the basic AMI, the default geodatabase is not enabled to use PostGIS. Therefore, after you install PostGIS, you must enable your geodatabase to use it. Once it is enabled, new data you add to the geodatabase can use the PostGIS geometry type by specifying the PG_GEOMETRY configuration keyword when you create or import the data. Existing data continues to use the ST_Geometry storage type.

After you have installed PostGIS and enabled a database to use it, you may want to create your own AMI to preserve the current state of your machine. See Create your own AMI for more information.

Tune database configuration settings

In most cases, the default configuration settings for the databases that store enterprise geodatabases are sufficient. However, if you have a more complex system with many users and large amounts of data and are using multiple Elastic Block Store (EBS) volumes, you have the option to alter the configuration of the PostgreSQL database cluster or SQL Server Standard instance.

The PostgreSQL database cluster or SQL Server instance provided with the ArcGIS for Server AMI has been configured for an Amazon standard large instance (7.5 GB memory). However, you may choose a different ArcGIS for Server AMI type with a larger memory footprint or have a different workflow for which you require nondefault configuration settings.

To change these configuration settings, you must connect directly to the DBMS using DBMS tools. See PostgreSQL or SQL Server's documentation for information on configuration settings and how to alter them.

Enterprise geodatabases in PostgreSQL

Geodatabases in PostgreSQL included with ArcGIS Server for Amazon Web Services

If you use the ArcGIS for Server (Ubuntu Linux) Amazon Machine Image (AMI) to create an ArcGIS Server site, you have the option to include a PostgreSQL database cluster. The database cluster can be on the ArcGIS for Server instance or on a separate instance that is part of your ArcGIS for Server site.

When you use ArcGIS Server Cloud Builder on Amazon Web Services to launch your ArcGIS Server site, the database cluster contains two pre-created geodatabases: egdb and geodata.

The egdb geodatabase is registered as an ArcGIS Server managed database. When you publish feature or WFS-T services to a site that has a registered managed database, the data can be copied from the geodatabase that contains the map source data (publisher geodatabase) to the registered database (server geodatabase). This data is dependent on the service; when you delete the feature or WFS-T service, the data is deleted from the egdb geodatabase.

The geodata geodatabase is intended for use as a replicated geodatabase. You can register the geodata geodatabase with ArcGIS Server. When you do so, designate the geodata geodatabase as a server database connection that is not the same as your publisher database connection (your on-premises enterprise geodatabase) and create a geodata service. You can replicate data from your on-premises enterprise geodatabase to the geodata geodatabase through the geodata service.

When you publish a feature or WFS-T service that includes the replicated data, edits made to the data through the feature service can be synchronized with the geodata service, updating the data in your on-premises enterprise geodatabase. Similarly, you can continue to edit your on-premises data and use the geodata service to synchronize those changes to the data in the geodata geodatabase.

The PostgreSQL database cluster also contains three database login roles:

- postgres: The database cluster administrator
- sde: The geodatabase administrator*
- owner: A user who owns a schema in the egdb and geodata databases and, therefore, can create data in them*

By default, the passwords for these users are as follows:

- postgres = postgres
- sde = E$ri3774*
- owner = owner*

*These login roles are only created if you use ArcGIS Server Cloud Builder on Amazon Web Services to create your site.

To secure your site, you must log in to the instance that contains the PostgreSQL database cluster and change the passwords. See Changing the passwords of the default PostgreSQL login roles for instructions.

Change default PostgreSQL passwords

Three database login roles are provided with the enterprise geodatabase in PostgreSQL on ArcGIS Server on Amazon Web Services instances. Their passwords are hard coded and well known. To make your instance more secure, you must change the login role passwords.

To change passwords, you must be able to log in to the Amazon Web Services instance. If you create your site using the ArcGIS Server Cloud Builder on Amazon Web Services application, you must specify a key pair when you create the site, then open port 22 in your security group to allow you to connect to the instances. You can close port 22 after you change the passwords.

1. Connect as ubuntu to the instance where PostgreSQL is installed.
   - If you are connecting to your ArcGIS for Server instance from a local Windows computer, make an SSH connection to the instance. See Administer your Amazon EC2 Ubuntu Linux instance with remote access from Windows for an example.
   - If connecting to your ArcGIS for Server instance from another Linux box, see Administer your Amazon EC2 Ubuntu Linux instance with remote access from Linux for an example.
2. Switch to the root user.
   ```
   sudo su -
   ```
3. Log in to psql using the postgres database login role, connecting to the postgres database.
   ```
   cd /data
   psql postgres postgres
   ```
4. Issue the \password command to alter the passwords of the three login roles.
   The syntax for the \password command is \password <username>.

   For example, to change the sde password, type:
   ```
   \password sde
   ```

   You will be prompted to type a new password. Repeat this for the owner and postgres user, giving each a strong, unique password.
5. To exit psql, type `\q`.

   Note: After you change the password of the owner login, you must update the registered databases for your ArcGIS Server on Amazon Web Services instance to use the new password.
6. Connect to the GIS Server from the Catalog tree in your local installation of ArcGIS for Desktop.
7. Right-click the server and click **Server Properties**.
8. Select a registered database in the list and click the **Properties** button.
   The **Edit Registered Database** dialog box opens.
9. Click **Edit** to change the connection information.
   The **Database Connection** dialog box opens.
10. Type the new password for the owner login in the **Password** field and click **OK**.
11. Click **OK** on the **Edit Registered Database** dialog box.
12. If you have registered the geodata or any other geodatabase on your ArcGIS Server on Amazon Web Services instance, repeat steps 8 through 11 for each one to change the password of the owner login role.

Additional security for geodatabases in PostgreSQL

Access to the ArcGIS for Server instance is controlled by the Amazon security group settings. By default, no outside ports are open, thereby securing your database cluster.

Although the Amazon security group provides the control to block rogue connection requests, you can further secure access to the PostgreSQL database cluster itself. For example, if you open ports in your security group that do allow outside access, you would likely want to increase the security on the database cluster.

To do this, you can alter settings in the pg_hba.conf file found in the PGDATA directory on the instance where PostgreSQL is installed. By default, the pg_hba.conf file is set up to allow any user from any machine to connect to the PostgreSQL database cluster. Alter these settings to designate the IP address or addresses of specific machines that you want to have access to the PostgreSQL database cluster. See PostgreSQL documentation for more information on how to configure the pg_hba.conf file.

Related topics

ArcGIS Server security on Amazon Web Services

Move a geodatabase in PostgreSQL to ArcGIS Server on AWS

You can move an existing enterprise geodatabase in PostgreSQL from a local server to an ArcGIS Server on Amazon Web Services instance or from one ArcGIS Server on Amazon Web Services instance to another using a backup file.

To do this, create a backup of the source geodatabase, transfer the backup file to the target ArcGIS Server on Amazon Web Services instance, create a database and login roles in the target PostgreSQL database cluster, and restore the database.

Check for custom variable settings

Any custom settings you had on the source database cluster that you want to retain must be added to the new database cluster. Configuration settings are stored in the postgresql.conf file. Make a copy of this file and move it to the target database cluster on your ArcGIS Server on Amazon Web Services instance.

The postgresql.conf file on the ArcGIS Server on Amazon Web Services instance can be found at /data on the mounted drive.

Create backup of source geodatabase

You can create a backup of the database to transfer the data files to an ArcGIS Server on Amazon Web Services instance.

Use the PostgreSQL pg_dump application to create a dump file.

1. Connect to the server where the source geodatabase is stored.
2. Execute the pg_dump command at a shell command prompt to create a backup of the database.

   ```
   pg_dump -U postgres -F c > /data/spdbbu11012.dump
   ```

   See the PostgreSQL documentation for more information on the pg_dump application.

Move backup file to destination instance

There are several ways to transfer the dump file to the target instance. See Strategies for data transfer to Amazon Web Services for different options for moving data. You can use the same method to move the postgresql.conf file, if necessary.

Be sure to place the dump file on the Elastic Block Store (EBS) volume.

Once the dump file is on the target ArcGIS Server on Amazon Web Services instance, prepare the PostgreSQL database cluster.

Prepare target PostgreSQL database cluster

You must have a database to which you will restore the dump file. Also, any login roles that own data in the source database must exist in the target database cluster.

Place configuration file

If you are using a customized postgresql.conf file, make a backup copy of the default postgresql.conf file on the target instance and place the customized file in /data on the mounted drive.

Create empty database

The database you create must have the same name as the database on the source PostgreSQL database cluster for which you created a dump file. Database names must be unique within a database cluster. That means if you are moving databases from one ArcGIS Server on Amazon Web Services to another, one of the following must be true:

- Your source database cannot be one of the default databases (egdb or geodata).
  or
- You must delete the default database of the same name from the target PostgreSQL database cluster before you can restore the transferred backup file.

1. Log in to the target ArcGIS Server on Amazon Web Services instance as the root user.
2. Open a command shell and log in to psql as the sde user.
3. Create a database into which you will restore the dump file.
   Keep the following information in mind:
   - The name, owner, and encoding of the target database must be the same as those of the source database.

- If you used a nondefault database template for the source database, such as a PostGIS database template, use that template for the target database.
- You can store the new database in an existing tablespace or create a new tablespace specifically for this database. If you want to use a new tablespace, you must create it before you create the database.

```
CREATE DATABASE <dbname>
  WITH OWNER = sde
       TEMPLATE = template0
       ENCODING = '<encoding of db>'
       TABLESPACE =
       LC_COLLATE = ''
       LC_CTYPE = ''
       CONNECTION LIMIT = -1;
```

Create login and group roles

The target database cluster must contain login roles for each user who owns data in the source database.

If you are moving a database from one ArcGIS Server on Amazon Web Services instance to another and you are using the default login roles, you do not need to create new roles in the target database cluster. However, if data in the source database is owned by nondefault login roles, you must create login roles with the same names in the target PostgreSQL database cluster. Similarly, if you want to use group logins in the target database, you must re-create those and grant them to the login roles.

See the PostgreSQL documentation for information on creating group and login roles and adding logins to groups.

Restore database

Use the PostgreSQL pg_restore application to restore the database.

⚠ **Caution:** You must run the pg_restore command twice: once to restore the public schema, which contains the sde_spatial_references system table, and a second time to restore the rest of the data. If you do not do this, no spatial data will be restored.

1. Connect to the destination ArcGIS Server on Amazon Web Services instance (the one to which you moved the dump file).
2. Open a command shell and log in to psql as the sde user.
3. Execute the pg_restore command at a command shell prompt to restore the public schema in the database.
   The sde user must be a superuser in PostgreSQL to execute the pg_restore command.

   ```
   pg_restore -U sde -n public -d spdb /data/spdbbu11012.dump
   Password:
   ```

4. Execute the pg_restore command a second time to restore the entire database.

   ```
   pg_restore -U sde -d spdb /data/spdbbu11012.dump
   Password:
   ```

   See the PostgreSQL documentation for more information on the pg_restore application.

After the database is restored on the target PostgreSQL database cluster, check to be sure the schemas and tables you were expecting are present in the new database. To do this, query the PostgreSQL catalog views that store this information. For example, you could query the pg_tables catalog view to see all the tables in the database and the schema in which they are stored.

Set database variables

Now you will use the custom database variable information you gathered earlier to set variables on the new database.

These steps instruct you on how to add and set the search_path variable. The same basic steps are used to set custom values for other database variables.

1. Connect to the new database from psql as the sde user.
2. From the psql prompt, set the search path for the database to include the sde schema.

   ```
   SET search_path TO "$user",public,sde;
   ```

Related topics

Open Amazon EC2 security group for ArcGIS Server

Upgrade geodatabases in PostgreSQL in ArcGIS Server for Amazon Web Services

To upgrade geodatabase functionality, you can either create an ArcGIS Server 10.3 for Amazon Web Services (AWS) instance, move your data to the new geodatabase, and republish services, or you can update the ArcGIS software on your existing AWS instance and upgrade the geodatabase.

The following sections describe methods for getting your geodatabase to the latest possible state, and when you would use each method.

Create an ArcGIS 10.3 for AWS instance and move your data from an older instance

The easiest way to get all the latest versions of ArcGIS and PostgreSQL is to create an ArcGIS Server for AWS instance from the latest AMI, move your data, and republish services. See The ArcGIS Server AMIs for information on what software and operating system versions are present in the latest ArcGIS Server AMIs.

For information on moving data, see the section "Options for transferring data to the cloud" in Strategies for data transfer to Amazon Web Services.

If you are using an ArcGIS Server 10 AWS instance with data in PostgreSQL, you must use this method to upgrade your geodatabase.

Move from a 10 instance to a 10.3 instance

At ArcGIS 10, there was a separate enterprise geodatabase Amazon Machine Image (AMI) that contained a PostgreSQL 8.3.8 database cluster on Windows. In ArcGIS 10.3, the ArcGIS Server (Linux) with PostgreSQL AMI runs Ubuntu Linux and can optionally contain a PostgreSQL 9.3.4 database cluster. You cannot move directly from a PostgreSQL database on Windows to one on Linux; therefore, if you migrate to ArcGIS 10.3 Server for AWS from ArcGIS 10 Server for AWS, you must move or reload your data.

Move from a 10.1 or 10.2.x instance to a 10.3 instance

The version of PostgreSQL included with the ArcGIS Server AMIs has changed over time. The following lists the version of PostgreSQL included with each Linux AMI:

ArcGIS Server (Linux) with PostgreSQL AMI	PostgreSQL version included
ArcGIS 10.1	9.05
ArcGIS 10.2	9.013
ArcGIS 10.2.1 and 10.2.2	9.2.4
ArcGIS 10.3	9.3.4

If you want to use PostgreSQL 9.3.4, you can move your data to ArcGIS Server 10.3 for AWS.

Update the ArcGIS software on your existing AWS instance and upgrade the geodatabase

You can install the latest version of ArcGIS Server on your existing instance, upgrade your geodatabase (if needed), and continue using your existing data and services. Be aware, though, that this means the operating system and all other software on the instance remain at the version they were when you created your instance. For example, if you created your instance from an ArcGIS Server 10.1 for AWS AMI, you will continue to use a PostgreSQL 9.0.5 database on an Ubuntu 10.04 operating system.

The next two sections explain how to upgrade a geodatabase in PostgreSQL on an existing ArcGIS Server AWS instance.

Update ArcGIS Server on an existing ArcGIS Server for AWS instance and upgrade geodatabases on SITEHOST

If you want to update an ArcGIS Server for AWS instance, follow the steps in this section to upgrade geodatabases in PostgreSQL that reside on the same instance as ArcGIS Server (SITEHOST).

Update ArcGIS for Server, move the new st_geometry.so file to the PostgreSQL lib directory, stop ArcGIS Server, run a Python script to upgrade each geodatabase, and then restart ArcGIS Server.

1. Install the latest version of ArcGIS for Server on your AWS instance.
   See Applying an ArcGIS update to a single-machine site or Applying an ArcGIS update to an Ubuntu multiple-machine site for instructions if you have not already installed the software.

2. Before you upgrade your geodatabases, make backup copies of them. The easiest way to do that is to make a Cloud Builder backup of your site (if you created your site using Cloud Builder), or make a snapshot of the EBS volume that contains your data. If you want to make a snapshot of the EBS volume, see Amazon Web Services documentation for instructions.
3. Also, before upgrading, you must copy the st_geometry.so file from `/arcgis/server/DatabaseSupport/PostgreSQL/Linux64` to `/usr/lib/postgresql/<postgresql version>/lib`.
   a. The st_geometry.so file is owned by root; therefore, after connecting to the Amazon Web Services SITEHOST instance as ubuntu, switch users to root:
   ```
   sudo su -
   ```
   Note: Be aware that you cannot connect as the arcgis user and switch to the root user.
   b. Before you copy the new file, rename the old one.
   ```
   mv /usr/lib/postgresql/<postgresql version>/lib/st_geometry.so /usr/lib/postgresql/<postgresql version>/lib/st_geometryOLD.so
   ```
   c. Copy the new st_geometry.so file.
   ```
   cp /arcgis/server/DatabaseSupport/PostgreSQL/Linux64/st_geometry.so /usr/lib/postgresql/<postgresql version>/lib
   ```
   d. Change permissions on the st_geometry.so file to allow other users to read and execute it.
   ```
   chmod 755 /usr/lib/postgresql/<postgresql version>/lib/st_geometry.so
   ```
4. Switch to the arcgis user.
   ```
   su - arcgis
   ```
5. Open a text editor to create the upgrade Python script.
   ```
   vi upgrade.py
   ```
6. Copy the following script into a text editor.
   Alter the script to use information specific to your site, including the instance name, sde password, connection file name and location, and database name.
   ```
   #Import arcpy module
   import arcpy

   # Create connection file
   arcpy.CreateDatabaseConnection_management("/tmp","egdb_connection.sde","POSTGRESQL","ec2-123-456-789-100.compute-1.amazonaws.com","DATABASE_AUTH", "sde
   # Upgrade geodatabase
   arcpy.UpgradeGDB_management("/tmp/egdb_connection.sde", "PREREQUISITE_CHECK", "UPGRADE")
   ```
   See Create Database Connection and Upgrade Geodatabase for more information on tool syntax.
7. Save the file and close the text editor.
8. Stop ArcGIS Server.
   ```
   /arcgis/server/stopserver.sh
   ```
9. Run the script in Python.
   ```
   /arcgis/server/tools/python <path to the file>/upgrade.py
   ```
   You will likely see multiple notices about type references. You can ignore these; they do not indicate the upgrade failed.

   Information is written to the GDBUpgrade.log in `/arcgis/server/framework/runtime/.wine/drive_c/users/arcgis/LocalAppData/ESRI/Server<#>` if the upgrade does fail.
10. Run the following in Python to be sure your geodatabase is upgraded. Replace `/tmp/egdb_connection.sde` with your connection file location and name.
    First, open an arcpy prompt.
    ```
    cd /arcgis/server/tools
    ./python
    ```

Use `isCurrent` to determine if the geodatabase is upgraded to the current release.

```
import arcpy
isCurrent = arcpy.Describe('/tmp/egdb_connection.sde').currentRelease
print isCurrent
```

If `print isCurrent` returns `True`, your geodatabase is upgraded.

11. To exit the arcpy prompt, type `quit()`.
12. Alter the script and run it again for each geodatabase you need to upgrade.
13. Once all geodatabases are upgraded, restart ArcGIS Server.

```
/arcgis/server/startserver.sh
```

If you have a multiple-machine site, proceed with step 14 in Applying an ArcGIS update to an Ubuntu multiple-machine site.

Update ArcGIS Server on an existing ArcGIS Server for AWS instance and upgrade geodatabases that are on their own AWS instance (EGDBHOST)

If you want to update an ArcGIS Server for AWS instance, follow the steps in this section to upgrade geodatabases in PostgreSQL that are on their own, dedicated AWS instance (EGDBHOST).

Update ArcGIS for Server on the SITEHOST, move the new st_geometry.so file from SITEHOST to the PostgreSQL lib directory on EGDBHOST, stop ArcGIS Server, run a Python script from SITEHOST to upgrade each geodatabase, then restart ArcGIS Server.

1. Install the new version of ArcGIS for Server on your AWS instance.
   See Apply an ArcGIS update to a single-machine site or Apply an ArcGIS update to an Ubuntu multiple machine site for instructions.
2. Before you upgrade your geodatabases, make backup copies of them. The easiest way to do that is to make a snapshot of the EBS volume that contains your data. See Amazon Web Services documentation for instructions on making a snapshot of an EBS volume.
3. Also, before upgrading, you must copy the st_geometry.so file from `/arcgis/server/DatabaseSupport/PostgreSQL/Linux64` to `/usr/lib/postgresql/9.0/lib`.
   a. Before you copy the new file, rename the old one on EGDBHOST. Since the st_geometry.so file is owned by the root user, log in to EGDBHOST as ubuntu and switch to the root user.

   ```
   sudo su -
   mv /usr/lib/postgresql/9.0/lib/st_geometry.so /usr/lib/postgresql/<postgresql version>/lib/st_geometryOLD.so
   ```

   b. You must move the st_geometry.so file from `/arcgis/server/DatabaseSupport/PostgreSQL/Linux64` on SITEHOST to `/usr/lib/postgresql/<postgresql version>/lib` on EGDBHOST.
   Since the two source and destination directories on the different instances are not accessible by the same user, log in to SITEHOST as the arcgis user and copy the file to the gisdata directory on EGDBHOST.

   ```
   cp /arcgis/server/DatabaseSupport/PostgreSQL/Linux64/st_geometry.so /net/EGDBHOST/gisdata/st_geometry.so
   ```

   c. Log in to EGDBHOST as ubuntu.
   d. Switch to the root user.

   ```
   sudo su -
   ```

   e. Copy the st_geometry.so file from `gisdata` to `/usr/lib/postgresql/<postgresql version>/lib`.

   ```
   cp /gisdata/st_geometry.so /usr/lib/postgresql/<postgresql version>/lib/st_geometry.so
   ```

   f. Change permissions on the st_geometry.so file to allow other users to read and execute it.

   ```
   chmod 755 /usr/lib/postgresql/<postgresql version>/lib/st_geometry.so
   ```

4. Log in to SITEHOST as the arcgis user.
5. Open a text editor to create the upgrade Python script.

For example:

```
vi upgrade.py
```

6. Copy the following script into a text editor.
   Alter the script to use information specific to your site, including the instance name, sde password, connection file name and location, and database name.

```
#Import arcpy module
import arcpy

# Create connection file
arcpy.CreateDatabaseConnection_management("/tmp","egdb_connection.sde","POSTGRESQL","ec2-123-456-789-100.compute-1.amazonaws.com","DATABASE_AUTH", "sde
# Upgrade geodatabase
arcpy.UpgradeGDB_management("/tmp/egdb_connection.sde", "PREREQUISITE_CHECK", "UPGRADE")
```

   See Create Database Connection and Upgrade Geodatabase for more information on tool syntax.

7. Save the file and close the text editor.
8. Stop ArcGIS Server.

```
/arcgis/server/stopserver.sh
```

9. Run the script in Python.

```
/arcgis/server/tools/python <path to the file>/upgrade.py
```

   You will likely see several notices about type references. You can ignore those messages.

   If the upgrade fails, information is written to the GDBUpgrade.log in `/arcgis/server/framework/runtime/.wine/drive_c/users/arcgis/LocalAppData/ESRI/Server<#>`.

10. To be sure your geodatabase is upgraded, open Python and run the following, replacing `/tmp/egdb_connection.sde` with your connection file location and name.
    First, open an arcpy prompt.

```
cd /arcgis/server/tools
./python
```

    Use `isCurrent` to determine if the geodatabase is upgraded to the current release.

```
import arcpy

isCurrent = arcpy.Describe('/tmp/egdb_connection.sde').currentRelease

print isCurrent
```

    If `print isCurrent` returns `True`, your geodatabase is upgraded.

11. To exit the arcpy prompt, type `quit()`.
12. Alter the upgrade.py script and run it again for each geodatabase you need to upgrade.
13. Once all geodatabases are upgraded, restart ArcGIS Server.

```
/arcgis/server/startserver.sh
```

If you have a multiple-machine site, proceed with step 14 in Apply an ArcGIS update to an Ubuntu multiple-machine site.

Create geodatabases in PostgreSQL on AWS

If you created an ArcGIS Server on Amazon Web Services (AWS) instance using the Esri AMI that includes PostgreSQL, you can create geodatabases using a Python script that calls the Create Enterprise Geodatabase geoprocessing tool.

When you use ArcGIS Server Cloud Builder on Amazon Web Services to launch your instance, two enterprise geodatabases are created in PostgreSQL. If you want additional geodatabases—for example, if you want to customize the name or location of the geodatabases you use, or you have multiple departments that maintain their own discrete data and require their own geodatabases—you can create additional geodatabases.

When you manually set up your site using the AWS Management Console, no geodatabases are created. If you want to use geodatabases in PostgreSQL with your ArcGIS Server on Amazon Web Services instance, you must create them.

Follow these steps to create a geodatabase in PostgreSQL to use with your ArcGIS Server on Amazon Web Services instance:

1. Make an SSH connection to your ArcGIS for Server site (SITEHOST).
   You must open the SSH port (22) in your security group to do this.
2. Copy the following script into a text editor:

```
#Import arcpy module
import arcpy

arcpy.CreateEnterpriseGeodatabase("PostgreSQL", "<aws instance name>", "<new geodatabase name>", "DATABASE_AUTH", "postgres", "<postgres password>", "", "sd
```

3. Alter the values in brackets (<>) to match the information for your site and save the file with a `.py` extension.
4. Open a command shell and set environment variables to point to the script.
5. Run the script to create the geodatabase.
   If the information you provide is correct, you receive a message indicating the geodatabase was successfully created. If you receive an error message, be sure the information you provided was correct (for example, you used the correct passwords and the database name is valid).

   Tip: Check the PostgreSQL log file at `data/pg_log` for extended error messages.

   You now have a new geodatabase. You must add login roles and user schemas to the database.
6. Copy this script into a text editor to connect to the new geodatabase and create a login role with matching schema. Alternatively, you can specify an existing login role, and a schema for it will be created in the database.

```
#Import arcpy module
import arcpy

arcpy.CreateDatabaseConnection_management("<path to directory where connection file to be created>", "<connection file name>.sde", "POSTGRESQL", "<aws insta

arcpy.CreateDatabaseUser_management("<path to directory with connection file>/<connection file name>.sde", "DB", "<login role name>", "<login password>")
```

7. Alter the values in brackets (<>) to match the database and login role information for your site and save the file with a `.py` extension.
   For example, if you want the default login role created with ArcGIS Server Cloud Builder on Amazon Web Services—owner—to create data in your new geodatabase, specify the owner login role in the script.
8. Open a command shell and set environment variables to point to the script.
9. Run the script to create a login role and schema (or a schema for an existing login role).
   The script grants usage privileges on the schema to public.
10. Repeat the previous step for each login role that will be creating data in the new geodatabase.

Related topics

Strategies for loading data into a geodatabase on Amazon Web Services

Enterprise geodatabases in SQL Server

Enterprise geodatabases in SQL Server in ArcGIS Server on Amazon Web Services

If you use an ArcGIS for Server (Windows) Amazon Machine Image (AMI) to create an ArcGIS Server site, you have the option to include one of the following Microsoft SQL Server instances:

- SQL Server Standard
- Amazon Relational Database Service (RDS) for SQL Server
- SQL Server Express

When you create your instance using ArcGIS Server Cloud Builder on Amazon Web Services, the SQL Server instance contains two precreated geodatabases: egdb and geodata.

The egdb geodatabase is registered as an ArcGIS Server managed database. When you publish feature or WFS-T services to a site that has a registered managed database, the data can be copied from the geodatabase that contains the map source data (publisher geodatabase) to the registered database (server geodatabase). This data is dependent on the service; when you delete the feature or WFS-T service, the data is deleted from the egdb geodatabase.

The geodata geodatabase is intended for use as a replicated geodatabase. You can register the geodata geodatabase with ArcGIS Server. When you do so, designate the geodata geodatabase as a server database connection that is not the same as your publisher database connection (your on-premises enterprise geodatabase) and create a geodata service. You can replicate data from your on-premises enterprise geodatabase to the geodata geodatabase through the geodata service.

When you publish a feature or WFS-T service that includes the replicated data, edits made to the data through the feature service can be synchronized with the geodata service, updating the data in your on-premises enterprise geodatabase. Similarly, you can continue to edit your on-premises data and use the geodata service to synchronize those changes to the data in the geodata geodatabase.

SQL Server Standard

If you choose SQL Server Standard, you can choose to create the DBMS on the same instance as ArcGIS Server or on an instance separate from your ArcGIS Server instance.

The SQL Server Standard instances are set to use mixed-mode authentication. They come with the following two operating system logins:

- Administrator: This login is the ArcGIS Server on Amazon Web Services instance administrator and an administrator in the SQL Server instance.
- ArcGIS: This login is an administrator in the SQL Server instance and is the login under which ArcGIS services run. This login must have access to the data to publish services.

Both logins are members of the SQL Server sysadmin fixed server role; therefore, the egdb and geodata geodatabases that are created by ArcGIS Server Cloud Builder on Amazon Web Services are stored in the dbo schema, and any data loaded while logged in as either of these users is stored in the dbo schema in each geodatabase. As dbo, these users have full privileges on all the data loaded to the geodatabases and the SQL Server instance itself.

The passwords for these logins are set by the system. You can alter the password for the Administrator login. See Change default Windows Administrator password for instructions.

If you manually create your site using the Amazon Web Services console, you must create your own geodatabases. You can use the Create Enterprise Geodatabase geoprocessing tool or Python script to do this. After the geodatabases are created, you must create ArcGIS users in the database for the ArcGIS login so you can publish services.

Amazon RDS for SQL Server

Amazon Relational Database Service (RDS) performs some database maintenance tasks, so you don't have to. For example, RDS instances automatically apply database patches and create backups of your database. They are also designed to be highly available, as they can be run in multiple availabilty zones (referred to as multi-AZ deployments).

Multi-AZ deployments are the default deployment type for Amazon RDS instances; therefore, Amazon RDS for SQL Server instances you launch using ArcGIS Server on Amazon Web Services at 10.3 and newer release AMIs will use this deployment type.

Note: Because they provide enhanced availability, Amazon recommends you use multi-AZ deployments for production systems. Be aware, though, that these deployments cost more to use per hour and for storage than other deployments. Consult Amazon's pricing guide for more information.

When you use an Amazon RDS for SQL Server instance in your ArcGIS Server on AWS site, the Amazon RDS for SQL Server instance is always separate from your ArcGIS Server instance.

SQL Server RDS instances do not support operating system authenticated logins. When you create your site using ArcGIS Server Cloud Builder on Amazon Web Services, two SQL Server logins and database users are created.

- EsriRDSAdmin: The EsriRDSAdmin user is the RDS master user, which creates the databases and sde user. EsriRDSAdmin also owns the data that is copied to EGDBHOST when you publish feature services. ArcGIS Server Cloud Builder on Amazon Web Services sets the password for this account to be the same as the password you provide for the ArcGIS Server site administrator.
- Sde: The sde user owns the geodatabase system objects. ArcGIS Server Cloud Builder on Amazon Web Services sets the password for this account to be the same as the password you provide for the ArcGIS Server site administrator.

If you manually create your site using the Amazon Web Services console and want to use SQL Server RDS for data storage, you need to do the following to get your geodatabase set up:

- Use SQL Server Management Studio or TSQL to create a database.
- Use SQL Server Management Studio or TSQL to create an sde login, database user, and schema.
- Use SQL Server Management Studio or TSQL to grant the sde user sufficient privileges to create a geodatabase.
- Run the Enable Enterprise Geodatabase geoprocessing tool or Python script to create a geodatabase.

  Note: You cannot use the Create Enterprise Geodatabase geoprocessing tool or script to create an sde user and geodatabase in SQL Server RDS.

- Create a login, database user, and matching schema to load data, and grant the user privileges to create data. You can use the Create Database User geoprocessing tool or a Python script to do this, or you can use Management Studio or TSQL.
- Register your geodatabase with ArcGIS Server.

SQL Server Express

SQL Server Express instances are always created on the same instance as ArcGIS Server.

The SQL Server Express instances use Windows-authenticated logins exclusively. They come with the following two operating system logins:

- Administrator: This login is the ArcGIS Server on Amazon Web Services instance administrator and an administrator in the SQL Server Express instance.
- ArcGIS: This login is an administrator in the SQL Server Express instance and is the login under which ArcGIS services run. This login must have access to the data to publish services.

Both logins are members of the SQL Server sysadmin fixed server role; therefore, the egdb and geodata geodatabases are stored in the dbo schema, and any data loaded while logged in as either of these users is stored in the dbo schema in each geodatabase. As dbo, these users have full privileges on all the data loaded to the geodatabases and the SQL Server instance itself.

The passwords for these logins are set by the system. You can alter the password for the Administrator login. See Change default Windows Administrator password for instructions.

Workgroup geodatabases

Change default Windows Administrator password

The operating system administrator login password for ArcGIS Server on Amazon Web Services is randomly generated. If you want to log in to your ArcGIS Server on Amazon Web Services Windows instances through Remote Desktop, you can get the decrypted password using the PEM file generated by your key pair. You can keep this password, or you can log in and change it to something you can more easily remember.

The Administrator login is also a member of the Microsoft SQL Server sysadmin server role for ArcGIS Server on Amazon Web Services sites that include SQL Server.

Note: If your site includes a SQL Server instance on a separate ArcGIS Server on Amazon Web Services instance, be sure you use the same password for the Administrator operating system login on both instances.

If you want to change passwords for the Administrator operating system login, do the following:

1. In the Amazon Web Services Management Console, right-click your ArcGIS for Server instance on Windows and click **Get Windows Password**.
2. In the resultant dialog box, you will need to browse to your PEM file.
   A decrypted password for the Windows Administrator is displayed.
3. Use Remote Desktop to connect to your ArcGIS for Server instance server.

   Tip: Remember that to log in to your ArcGIS Server on Amazon Web Services Windows instance through Remote Desktop, you must specify a key pair when you create the site with the ArcGIS Server Cloud Builder on Amazon Web Services application, and you must open ports 3389 and 6080 in your security group.
4. Once connected to the instance, change the Windows Administrator password. If you need instructions on how to reset the password of a Windows operating system login, see the Microsoft documentation.

Related topics

Administer your Amazon EC2 instance with Windows Remote Desktop Connection

Connect from an ArcGIS Server instance to an enterprise geodatabase

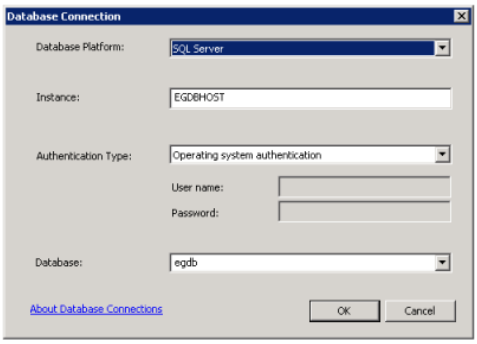
If you deployed an ArcGIS Server on Amazon Web Services on a Windows instance that includes an enterprise geodatabase, you can log in to your ArcGIS for Server instance and connect from ArcGIS for Desktop to the enterprise geodatabase.

1. To connect to the instance, you must open ports in your security group.
   See Administer your Amazon EC2 instance with Windows Remote Desktop Connection for information on opening ports, enabling Remote Desktop, obtaining the Administrator login's password, and authorizing ArcGIS for Desktop. (You can ignore the last step, though; you don't need to open Server Manager.)
2. If your site has SQL Server Standard on its own dedicated EC2 instance, you must change the Windows Administrator passwords to be the same on both the EGDBHOST and SITEHOST instances.
   See Change password for the default Windows Administrator login for instructions.
3. Once you have logged in to your ArcGIS for Server instance as the Administrator and authorized ArcGIS for Desktop, start ArcCatalog.
4. Expand the Database Connections node in the Catalog tree.
5. Double-click **Add Database Connection**.
6. Provide the following connection properties to connect to your enterprise geodatabase in SQL Server:

Field	Value
Database Platform	SQL Server
Instance	The name of the SQL Server instance Use one of the following for the instance name: • `EGDBHOST`—You can use the site name `EGDBHOST` if you launched your site from ArcGIS Server Cloud Builder on Amazon Web Services. • The public DNS of the instance where SQL Server is installed—Only use the DNS if SQL Server is running on a different instance than ArcGIS Server and you created the instance outside of ArcGIS Server Cloud Builder on Amazon Web Services. The following is an example DNS:`ec2-123-45-678-90.compute-1.amazonaws.com` Note that if you stop your site or your EC2 instance, the public DNS will change. Therefore, if you use the DNS to connect, you will have to update this connection information. • `localhost`—You can type `localhost` for the instance if your site has SQL Server on the same instance as ArcGIS Server and ArcGIS for Desktop.
Authentication Type	Operating system authentication
Database	The name of the geodatabase to which you want to connect If you launched your site from ArcGIS Server Cloud Builder on Amazon Web Services, there are two geodatabases present on the SQL Server instance by default: `egdb` and `geodata`.

   Be sure you check **Save user name and password**.

   The **Database Connection** dialog box should look similar to the following:



7. Click **OK** to connect and close the **Database Connection** dialog box.
8. Type a name for the connection.
   For example, type `ags1connection`.

Move enterprise geodatabases in SQL Server between AWS instances

You can move an existing enterprise geodatabase in SQL Server from one ArcGIS Server on Amazon Web Services instance to another by detaching the database files and transaction logs from one SQL Server instance, moving them to the target ArcGIS Server on Amazon Web Services instance, and attaching them to the new SQL Server instance.

Note: This workflow does not apply to Amazon Relational Database Services for SQL Server.

When would you use this method to move data?

You might use this method if you created a new ArcGIS Server on Amazon Web Services instance and you want to move a geodatabase from your existing instance (the source instance) to the new instance (the target instance).

Moving individual geodatabases in SQL Server between instances of ArcGIS Server on Amazon Web Services is relatively simple as long as you haven't added custom logins to the SQL Server instance.

If you plan to move one of the default geodatabases (egdb or geodata), you must delete the corresponding geodatabase from the target instance. Database names on an instance must be unique, and geodatabases cannot be renamed. If the geodatabase on the target instance contains data, you should not use this method to move the geodatabase.

Stop web services

There cannot be any active connections to a database if you want to detach it. Therefore, you must stop the services that use data in the database before you detach the database from the existing instance.

1. Open ArcGIS Server Manager for your existing ArcGIS Server on Amazon Web Services instance.
2. On the **Services** page, choose the services that are connected to the database you want to detach and click **Stop**.

Detach database

Use SQL Server Management Studio to detach the database.

1. Make a remote desktop connection to your existing instance.
   You must open the RDP port in your security group to make remote desktop connections. You can remove this rule from your security group after you move your data.
   - If your SQL Server instance is on the same machine as ArcGIS for Server, connect to your ArcGIS for Server (SITEHOST) instance.
   - If your SQL Server instance is on a different machine, connect to that (EGDBHOST) instance.
2. Start SQL Server Management Studio.
3. Log in to your SQL Server instance using operating system authentication.
4. Expand the **Database** folder.
5. Right-click the database, point to **Tasks**, and click **Detach**.
6. Choose which options you want and click **OK**.

Start new instance

If the destination ArcGIS Server on Amazon Web Services instance to which you want to transfer the database has not yet been created, follow the instructions in Build an ArcGIS server site on Amazon Web Services to create your new site.

Delete existing database

If your ArcGIS Server on Amazon Web Services instance includes an enterprise geodatabase server, it comes with two geodatabases: egdb and geodata. If the geodatabase you are moving to the target ArcGIS Server on Amazon Web Services instance has the same name as one of these, you must delete the existing geodatabase before you can attach the one you moved.

⚠ **Caution:** Do not do this if the geodatabase on the target instance contains data that is still needed. If your destination instance already has a geodatabase of that name that contains data, you must use a different method to move your data from one instance to another; geodatabases cannot be renamed.

Prerequisite:
Delete the databases in SQL Server Management Studio on the new ArcGIS Server on Amazon Web Services instance.

1. Log in to the target ArcGIS Server on Amazon Web Services instance using remote desktop and the operating system Administrator login.
   You must open the RDP port in your new security group to make remote desktop connections. You can remove this rule from your security group after you move your data.

- If your SQL Server instance is on the same machine as ArcGIS for Server, connect to your ArcGIS for Server (SITEHOST) instance.
- If your SQL Server instance is on a different machine, connect to that (EGDBHOST) instance.

2. Start SQL Server Management Studio.
3. Log in to your SQL Server instance using operating system authentication.
4. Expand the **Database** folder.
5. Right-click the database that has the same name as the database you want to attach and click **Delete**.
6. Click **OK** on the **Delete Object** dialog box to drop the database.

Move database and transaction log files

You must move the detached database (.mdf) and log files (.ldf) from the source instance to the target instance. See Strategies for data transfer to Amazon Web Services for ways to move files to your ArcGIS Server on Amazon Web Services instance. Be sure to place the files on the data volume, not the C drive of the target instance.

Attach database and transaction log files to new instance

Use SQL Server Management Studio to attach the files to the new SQL Server instance.

1. If you closed Management Studio, restart it and log in to your target instance.
2. Right-click the **Database** folder and click **Attach**.
3. Click **Add** on the **Attach Databases** dialog box.
4. Browse to the location of your database file on the target ArcGIS Server on Amazon Web Services instance.
5. Choose the .mdf file and click **OK**.
6. Click **OK** to attach the database.
   If the database doesn't appear in the list, refresh the **Database** folder.

Add users to database, if required

If you moved your geodatabase from another ArcGIS Server on Amazon Web Services instance, the database already contains the users needed for ArcGIS. If you had added other logins and users to your source instance and database, and those users own data, you must add the same logins to the new SQL Server instance. If you are using SQL Server-authenticated logins, you might need to synchronize SIDs. See the SQL Server documentation for information on creating and synchronizing logins and users.

Related topics

Open Amazon EC2 security group for ArcGIS Server
Administer your Amazon EC2 instance with Windows Remote Desktop Connection

Create an enterprise geodatabase in SQL Server on AWS

If you created an ArcGIS Server on Amazon Web Services (AWS) instance using the Esri AMI that includes SQL Server and you are using an ArcGIS for Server Enterprise license, you can create geodatabases using the Create Enterprise Geodatabase geoprocessing tool.

Note: The Create Enterprise Geodatabase tool cannot be used to create a geodatabase in Amazon Relational Database Service (RDS) for SQL Server or a SQL Server Express instance licensed with ArcGIS for Server workgroup. To create geodatabases (or additional geodatabases) in a SQL Server RDS, see Create a geodatabase on Amazon Relational Database Service for SQL Server. To create geodatabases in a SQL Server Express instance licensed with ArcGIS for Server workgroup, see Creating additional workgroup geodatabases.

When you use ArcGIS Server Cloud Builder on Amazon Web Services to launch your instance, two enterprise geodatabases are created in SQL Server. If you want additional geodatabases—for example, if you want to customize the name or location of the geodatabases you use, or you have multiple departments that maintain their own discrete data and require their own geodatabases—you can create additional geodatabases.

When you manually set up your site using the AWS Management Console, no geodatabases are created. If you want to use enterprise geodatabases in SQL Server with your ArcGIS Server on Amazon Web Services instance, you must create them.

Follow these steps to create an enterprise geodatabase in SQL Server to use with your ArcGIS Server on Amazon Web Services instance:

1. Make a remote desktop connection to your ArcGIS for Server instance, logging in as the administrator.
   You must open the RDP port (3389) in your security group to do this.
2. Start ArcMap and open the **Catalog** window or start ArcCatalog.
3. Open the Create Enterprise Geodatabase tool.
   You can search for or browse to this tool, which is located in the Geodatabase Administration toolset of the Data Management toolbox.
4. Choose **SQL Server** from the **DBMS Type** drop-down list.
5. In the **Instance** text box, type or paste the public DNS of the instance where SQL Server is installed.
6. In the **Database** text box, type a name for the database you want to create to contain your geodatabase.
7. Check **Operating System Authentication**.
8. Uncheck **Sde Owned Schema**.
9. Browse to the keycodes file on the ArcGIS for Server instance to populate the **Authorization File** text box.
   When you authorized ArcGIS for Server, a keycodes file was written to `Program Files\ESRI\License<release>\sysgen`.
10. Click **OK** to run the tool.
    If you encounter any errors, extended errors will be written to a log file (GDBCreateGeodatabase<#>.log), which is created in the directory specified for your %TEMP% variable on the computer where the tool is run.

You now have a new geodatabase.

Related topics

Strategies for loading data into a geodatabase on Amazon Web Services

Create a geodatabase on Amazon Relational Database Service for SQL Server

When you use ArcGIS Server Cloud Builder on Amazon Web Services to launch an ArcGIS Server on Amazon Web Services (AWS) instance using the Esri AMI that includes Amazon Relational Database Service (RDS) for SQL Server, two enterprise geodatabases are created in the RDS instance. If you want additional geodatabases—for example, if you want to customize the name or location of the geodatabases you use, or you have multiple departments that maintain their own discrete data and require their own geodatabases—you can create additional databases using Microsoft SQL Server Management Studio, add the sde user and schema to them, and run the Enable Enterprise Geodatabase geoprocessing tool to create geodatabases in the databases.

When you manually set up your site using the AWS Management Console, no RDS instance, sde user, databases, or geodatabases are created. If you want to use enterprise geodatabases in an Amazon RDS for SQL Server instance with your ArcGIS Server on Amazon Web Services instance, you must create them and add them to your site.

The first set of steps below describe creating additional geodatabases for an ArcGIS Server on Amazon Web Services site that was created using the ArcGIS Server Cloud Builder on Amazon Web Services. The second set of steps explains how to create enterprise geodatabases in Amazon RDS for SQL Server if you launched the Esri AMI from the AWS Management Console.

Add a geodatabase to an existing Amazon RDS for SQL Server instance

Follow these steps to create an additional enterprise geodatabase in Amazon RDS for SQL Server to use with an ArcGIS Server on Amazon Web Services instance that you launched from ArcGIS Server Cloud Builder on Amazon Web Services:

1. Connect to the SQL Server RDS instance from SQL Server Management Studio using the master login. For instructions, see the AWS topic Connecting to a DB Instance Running the Microsoft SQL Server Database.
   You may need to open ports in your security group to do this.
2. Do all of the following from SQL Server Management Studio:
   a. Create a database. Store the database and log files on the D drive of the instance. Set other database configuration settings as you require.
   b. Create an sde user in the database that is mapped to the sde login.
   c. Create an sde schema owned by the sde user.
   d. Grant the sde user CREATE FUNCTION, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW privileges in the database.
   e. Create a user who will own and publish data. Map this user to the EsriRDSAdmin login.
   f. Create a schema owned by the user created in the previous step.
   g. Grant the new user CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW privileges in the database.

   For instructions on creating a database, login, user, and schema, see the Microsoft Books Online for SQL Server 2012.
3. Make a remote desktop connection to your ArcGIS for Server instance, logging in as the administrator.
   For instructions, see Administer your Amazon EC2 instance with Windows Remote Desktop Connection.
4. Start ArcMap and open the **Catalog** window or start ArcCatalog.
5. Connect to your new database using the sde login.
   See Connect from an ArcGIS Server instance to an enterprise geodatabase for instructions on making a database connection.
6. Open the Enable Enterprise Geodatabase tool and specify the database connection you made in the previous step.
7. Specify the authorization file for ArcGIS Server. This is stored on the C drive of the instance in `\\Program Files\ESRI\License<release#>\sysgen`.
8. Click **OK** to run the tool and create a geodatabase.

Once you have a geodatabase, create a connection to it and register it with ArcGIS Server. For instructions, see Registering your data with ArcGIS Server using ArcGIS for Desktop.

Add an Amazon RDS for SQL Server instance to ArcGIS Server on AWS

Follow these steps to create an Amazon RDS for SQL Server instance to use with your ArcGIS Server on Amazon Web Services instance; add the RDS instance to your ArcGIS Server site; create a database; create logins, users, and schemas; grant the sde user

privileges to create a geodatabase; grant the data owner user privileges to create data; and run the Enable Enterprise Geodatabase tool to create a geodatabase.

1. Follow the instructions in the AWS documentation to create a database instance running SQL Server.
   To create an instance in Amazon VPC, see the AWS topic Using Amazon RDS with Amazon Virtual Private Cloud.
2. Use SQL Server Management Studio to connect to the SQL Server database instance you created in the first step.
   For help with connecting, see the AWS topic Connecting to a DB Instance Running the Microsoft SQL Server Database Engine.
3. Do all of the following from SQL Server Management Studio:
   a. Create a database. Store the database and log files on the D drive of the instance. Set other database configuration settings as you require.
   b. Create a SQL Server authenticated login named sde.
   c. Create an sde user in the database that is mapped to the sde login.
   d. Create an sde schema owned by the sde user.
   e. Grant the sde user CREATE FUNCTION, CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW privileges in the database.
   f. Create a SQL Server authenticated login to be used to store and publish data.
   g. In the database, create a user that is mapped to the login created in the previous step.
   h. Create a schema, owned by the user you created in the previous step. The schema must have the same name as the user.
   i. Grant the user CREATE PROCEDURE, CREATE TABLE, and CREATE VIEW privileges in the database.

   For instructions on creating a database, login, user, and schema, see the Microsoft Books Online for SQL Server 2012.
4. Make a remote desktop connection to your ArcGIS for Server instance, logging in as the administrator.
   For instructions, see Administer your Amazon EC2 instance with Windows Remote Desktop Connection.
5. Start ArcMap and open the **Catalog** window or start ArcCatalog.
6. Connect to your new database using the sde login.
   See Connect from an ArcGIS Server instance to an enterprise geodatabase for instructions on making a database connection.
7. Open the Enable Enterprise Geodatabase tool and specify the database connection you made in the previous step.
8. Specify the authorization file for ArcGIS Server. This is stored on the C drive of the instance in `\\Program Files\ESRI\License<release#>\sysgen`.
9. Click **OK** to run the tool and create a geodatabase.
10. From the Catalog tree, connect to the geodatabase with the login for the data owner.
11. Use the connection file created in the last step to register the geodatabase with ArcGIS Server. For instructions, see Registering your data with ArcGIS Server using ArcGIS for Desktop.

Upgrade enterprise geodatabases in SQL Server in ArcGIS Server for Amazon Web Services

If you applied software updates to the ArcGIS clients on an ArcGIS Server for Amazon Web Services instance, you can upgrade your enterprise geodatabases in SQL Server. You can do this from the **Database Properties** dialog box in ArcGIS for Desktop as described in the following steps:

1. Log in to your ArcGIS Server instance using the administrator login.
2. There cannot be any users other than the administrator connected to the geodatabase when you upgrade. This includes service connections. You must stop any ArcGIS service connections that access data in your geodatabase before you upgrade.
3. Start Microsoft SQL Server Management Studio and log in with Windows authentication.
4. Create a full backup of each of the databases that contain an enterprise geodatabase.
   If you only use the geodatabases provided with your ArcGIS Server for Amazon Web Services instance, the databases are egdb and geodata.
5. On your ArcGIS Server instance, start ArcCatalog or ArcMap and open the **Catalog** window.
6. Connect to one of the enterprise geodatabases.
7. Right-click the geodatabase connection under **Database Connections** in the Catalog tree.
8. Click the **General** tab.
9. Click **Upgrade Geodatabase**.
   The Upgrade Geodatabase geoprocessing tool opens.
10. Leave both the **Pre-requisites check** and **Upgrade geodatabase** options checked and click **OK** to upgrade the geodatabase.
11. When the tool completes successfully, click **Close** to close the geoprocessing tool progress dialog box.
    If the tool does not complete successfully, information is written to the GDBUpgrade.log found in `C:\Users\<user name>\AppData\Local\ESRI\Desktop<#>`.
12. Repeat steps 5 through 10 to upgrade your other enterprise geodatabases.
13. Restart your services.

Related topics

Apply an ArcGIS update to a single-machine site
Apply an ArcGIS update to a Windows multiple-machine site

Workgroup geodatabases included with ArcGIS Server for Amazon Web Services

The ArcGIS Server Amazon Machine Image (AMI) can be used with an ArcGIS for Server Workgroup license. To use this, you authorize your ArcGIS for Server instance with your workgroup license, then you can use the SQL Server Express instance that is included with the ArcGIS for Server AMI. This instance is authorized to store workgroup geodatabases and use with ArcGIS for Server Workgroup.

The SQL Server Express instance (database server) comes with two geodatabases already created: egdb and geodata. As with the enterprise instance, the egdb geodatabase is registered as a database connection that is different from the publisher database connection. When you publish feature or WFS-T services to a site that has a geodatabase registered in this way, the data is copied from your source to the registered geodatabase in the cloud. The geodata geodatabase is intended for use as a replicated geodatabase. You can register the geodata geodatabase as a database connection that is different from the publisher database connection and create a geodata service from it. Through the geodata service, you can synchronize data from your on-premises enterprise or workgroup geodatabase to the geodata geodatabase.

Considerations when using ArcGIS Server Workgroup geodatabases

- If you connect to the database server from ArcGIS for Desktop in the ArcGIS Server on Amazon Web Services instance, use `localhost` in place of the server name.

  Legacy: Prior to ArcGIS 10.1 SP1, use `localhost\sqlexpress` for the server name.

  In the cloud, you are not locked into one specific server; therefore, you cannot specify a server name to connect to the SQL Server Express instance. Using localhost ensures ArcGIS can find the SQL Server Express instance.
- If you want to publish from the ArcMap installation on your ArcGIS Server on Amazon Web Services instance, you must create database connection (.sde) files for each geodatabase you want to use for source data because you cannot publish an ArcGIS service from the Database Servers node (a .gdb file) in the Catalog tree. If the service you are publishing is going to contain data from a workgroup geodatabase, the service must point to a connection under the Database Connections node (.sde file). To create a connection under the Database Connections node, right-click the geodatabase under the database server in the Catalog tree and click **Save Connection**. This creates a .sde file in the default location: `Users\Administrator\AppData\Roaming\ESRI\Desktop<#>\ArcCatalog`. Place the file in a location that the ArcGIS user can access to publish an ArcGIS service using the data in the geodatabase.
- Workgroup geodatabases are limited to a size of 10 GB.
- Workgroup geodatabases use Windows authentication exclusively. Therefore, if you want to add more users to the database server, you must add users to the ArcGIS for Server instance, then add the new users to the database server and grant permissions. Local users must be placed in the Windows Remote Desktop Users group to be able to log in to the EC2 instance.
- If you create a new ArcGIS for Server Workgroup instance and want to move your existing geodatabases to it, you must first detach the geodatabases from the database server in your existing instance, move the .mdf file to the new instance, and attach the geodatabase to the new database server. See Moving workgroup geodatabases between ArcGIS Server on Amazon Web Services instance for instructions.

See the topics in the Administering geodatabases on database servers section of the ArcGIS Help if you want to customize or directly administer these types of geodatabases.

Change default Windows Administrator password

The operating system administrator login password for ArcGIS Server on Amazon Web Services is randomly generated. If you want to log in to your ArcGIS Server on Amazon Web Services Windows instances through Remote Desktop, you can get the decrypted password using the PEM file generated by your key pair. You can keep this password, or you can log in and change it to something you can more easily remember.

The Administrator login is also a member of the Microsoft SQL Server sysadmin server role for ArcGIS Server on Amazon Web Services sites that include SQL Server.

Note: If your site includes a SQL Server instance on a separate ArcGIS Server on Amazon Web Services instance, be sure you use the same password for the Administrator operating system login on both instances.

If you want to change passwords for the Administrator operating system login, do the following:

1. In the Amazon Web Services Management Console, right-click your ArcGIS for Server instance on Windows and click **Get Windows Password**.
2. In the resultant dialog box, you will need to browse to your PEM file.
   A decrypted password for the Windows Administrator is displayed.
3. Use Remote Desktop to connect to your ArcGIS for Server instance server.

   Tip: Remember that to log in to your ArcGIS Server on Amazon Web Services Windows instance through Remote Desktop, you must specify a key pair when you create the site with the ArcGIS Server Cloud Builder on Amazon Web Services application, and you must open ports 3389 and 6080 in your security group.

4. Once connected to the instance, change the Windows Administrator password. If you need instructions on how to reset the password of a Windows operating system login, see the Microsoft documentation.

Related topics

Administer your Amazon EC2 instance with Windows Remote Desktop Connection

Move workgroup geodatabases between AWS instances

Complexity:
Beginner
Data Requirement:
Use your own data

You can move an existing workgroup geodatabase from one ArcGIS Server on Amazon Web Services instance to another by detaching the geodatabases from the source database server, moving the database files to the target ArcGIS Server on Amazon Web Services instance, and attaching them to the new database server.

If the source and destination ArcGIS Server on Amazon Web Services instances are at different releases, you may need to upgrade the geodatabase after moving it.

Stop web services

There cannot be any active connections to a geodatabase if you want to detach it. Therefore, you should stop the services that use the data in the geodatabase before you detach the geodatabase from the database server.

1. Open ArcGIS Server Manager for your existing ArcGIS Server on Amazon Web Services instance.
2. On the **Services** page, choose the services that are connected to the geodatabase you want to detach and click **Stop**.

Detach geodatabase

To move the database files, you must detach the geodatabase from the database server.

Connect to the database server (SQL Server Express instance) from the Catalog tree in ArcGIS for Desktop and detach the geodatabase you want to move.

Note: If this is the first time you have logged in to your instance, you must authorize ArcGIS for Desktop with a valid license before you can open ArcCatalog. Go to **Start** > **All Programs** > **ArcGIS** > **ArcGIS Administrator** on your source instance to authorize ArcGIS for Desktop.

1. Log in to the source ArcGIS Server on Amazon Web Services instance using remote desktop and the operating system Administrator login.
   Remember that to log in to an ArcGIS Server on Amazon Web Services instance, you must open the remote desktop port (3389) in your security group.
2. Start ArcCatalog.
3. Connect to the database server.
4. On the **Contents** tab, right-click the geodatabase you want to detach, point to **Administration**, and click **Detach**.
5. Take note of the location of the database file on the **Confirm Detach** dialog box.
6. Click **Yes** to proceed with detaching the geodatabase.

Start a new instance

If the destination ArcGIS Server on Amazon Web Services instance to which you want to transfer the geodatabase has not yet been created, follow the instructions in Build an ArcGIS Server site on Amazon Web Services to create your new site.

Delete geodatabases from destination database server

Database servers in your ArcGIS Server on Amazon Web Services instance come with two geodatabases: egdb and geodata. If the destination database server has a geodatabase with the same name as the one you detached from the source database server, you must delete the existing geodatabase before you can attach the one you moved.

Caution: Do not do this if the geodatabase on the destination database server contains data that is still needed. If your destination instance already has a geodatabase of that name that contains data, you must use a different method to move your data from one instance to another; geodatabases cannot be renamed. If you know the data is no longer needed, make a backup of the geodatabase before you delete it just to be safe.

If the destination database server contains a geodatabase with the same name as the geodatabase you are moving, connect to the database server (SQL Server Express instance) from the Catalog tree in ArcGIS for Desktop and delete the geodatabase that has the same name.

1. Log in to the destination ArcGIS Server on Amazon Web Services instance using the operating system Administrator login.
2. Start ArcCatalog.
   Remember, if this is a new site, you must authorize your ArcGIS for Desktop installation before you can use it.
3. Connect to the destination database server.
4. Right-click the geodatabase that has the same name as the geodatabase you are moving and click **Delete**.
5. Confirm that you want to delete the geodatabase.

Move .mdf files

You must move the detached database file (.mdf) from the source instance to the destination instance. The .mdf file can be found in the location you noted in the first section of this topic. See Strategies for data transfer to Amazon Web Services for ways to move files to your destination ArcGIS Server on Amazon Web Services instance. Be sure to place the .mdf file on the data volume (the D drive), not the C drive of the destination instance.

Attach geodatabase to new instance

Connect to the destination database server from the Catalog tree in ArcGIS for Desktop and attach the geodatabase.

1. In ArcCatalog, right-click the database server and click **Attach**.
   The **Attach Geodatabase** dialog box opens.
2. Browse to the location on the destination instance where you placed the .mdf file and click **OK** to add it.
3. Click **OK** to attach the geodatabase.

Upgrade workgroup geodatabases in ArcGIS Sever for Amazon Web Services

There are two different scenarios for upgrading a workgroup geodatabase in Amazon EC2:

- When you migrate your ArcGIS Server Workgroup for Amazon Web Services instances to a new, full release of the ArcGIS for Server Amazon Machine Image (AMI) (such as when moving from 10.2 to 10.3), you can move your geodatabases to the new instance. You may also need to upgrade your geodatabases.
- For service packs and patches, you can apply them to ArcGIS for Server and ArcGIS for Desktop in your ArcGIS Server Workgroup for Amazon Web Services instance, then upgrade your workgroup geodatabases if necessary.

1. If you are moving your workgroup geodatabase to a newer release of ArcGIS Server on Amazon Web Services, follow the steps in Moving workgroup geodatabases to an ArcGIS Server instance, then proceed to step 3.
2. If you are updating the ArcGIS software in your ArcGIS Server on Amazon Web Services instance, download the updates for ArcGIS for Server and ArcGIS for Desktop, place them on your instance, and follow the installation instructions provided with the download to install them.
   This will require you to make a remote desktop connection to your existing ArcGIS for Server instance. You must open the RDP port in your security group to make remote desktop connections. You can remove this rule from your security group after you install updates and upgrade geodatabases.
3. After geodatabases have been attached or ArcGIS updates have been applied, open ArcCatalog and connect to the database server in the Catalog tree.
4. Check the upgrade status of the geodatabases.
   a. Right-click the first geodatabase and click **Properties**.
   b. Click the **General** tab.
      If the Upgrade Status indicates an upgrade is not needed, close the **Geodatabase Properties** dialog box for this geodatabase and close your remote desktop session. No further action is required.

      If the Upgrade Status indicates the geodatabase can be upgraded, proceed with the remaining steps.
5. Close the **Geodatabase Properties** dialog box.
6. Make a backup of the geodatabase before upgrading.
   a. Right-click the geodatabase, point to **Administration**, then click **Backup**.
   b. Type a name for the backup file.
   c. Browse to the location where you want to store the backup.
   d. Type a description of the backup.
   e. Click **OK**.
7. Upgrade the geodatabase.
   a. Open the **Geodatabase Properties** dialog box.
   b. Click **Upgrade Geodatabase**.
   c. When the Upgrade Geodatabase geoprocessing tool opens, leave both the **Pre-requisites check** and the **Upgrade geodatabase** options checked, then click **OK** to upgrade the geodatabase.
   d. When the tool completes successfully, click **Close** to close the geoprocessing tool progress dialog box.
8. Repeat steps 4 through 7 to back up and upgrade each geodatabase on your database server.
9. Once all geodatabases are upgraded, close your remote desktop session.

Create additional workgroup geodatabases

To create additional workgroup geodatabases, connect to the SQL Server Express instance from the Database Servers node in the Catalog tree in ArcGIS for Desktop on the ArcGIS for Server site in the cloud.

Some common reasons you would create additional geodatabases include the following:

- You want to customize your geodatabase.
  The default geodatabases (egdb and geodata) cannot be renamed. If you want a geodatabase with a different name, you must create another geodatabase.
- You want to group your data by geodatabase.
  For example, if you have multiple departments that maintain their own discrete data, you might create separate geodatabases for each group.

1. Make a remote desktop connection to your ArcGIS for Server instance, logging in as the administrator.
   You must open the RDP port (3389) in your security group to do this.
2. Start ArcMap and open the **Catalog** window or start ArcCatalog.
3. Expand the **Database Servers** node in the Catalog tree.
4. If you have not already added the database server to the Catalog tree, do so now.
   a. Double-click **Add Database Server**.
   b. Type `LOCALHOST` in the **Database Server** field.
   c. Click **OK** to close the **Add Database Server** dialog box and connect.
5. Double-click the database server to connect to it.
6. Right-click the database server and click **New Geodatabase**.
7. Type a name for your new geodatabase in the **Geodatabase Name** text box.
8. You should store your data on an external volume (not the C: drive); therefore, click the ellipsis button and browse to the location where you want to store the database.
   If storing in the default volume created with your ArcGIS Server site, this would be the data folder on the D: drive.
9. Type an initial size for the database or accept the default value.
10. Click **OK** to create the geodatabase.
    The new geodatabase is added to the database server in the Catalog tree, and the ArcGIS and Administrator operating system users are automatically added to the new geodatabase. You can now register this geodatabase as a data store for your ArcGIS for Server site or transfer data to your ArcGIS for Server instance and load it into this geodatabase.

Related topics

Open Amazon EC2 security group for ArcGIS Server
Administer your Amazon EC2 instance with Windows Remote Desktop Connection
Strategies for loading data into a geodatabase on Amazon Web Services

File geodatabases

File geodatabases used with ArcGIS Server for Amazon Web Services

File geodatabases provide a convenient method for data storage that can be used with both ArcGIS for Server Workgroup and Enterprise licenses. The file geodatabases should be placed or created on an Elastic Block Store (EBS) of your ArcGIS Server on Amazon Web Services instances so they can be moved between instances more easily.

To help you decide whether you want to use a file geodatabase to store your data for use with ArcGIS for Server, see Data storage considerations for an ArcGIS Server site.

Replication

Replication to an Amazon Web Services instance using geodata services

If you are making edits to your GIS data, you can edit the data locally then use replication to synchronize the data through a geodata service published from your enterprise geodatabase in your ArcGIS Server on Amazon Web Services instance.

Replication and synchronization performed directly from your local geodatabase to your enterprise geodatabase in your ArcGIS Server on Amazon Web Services instance would take a great deal of time. For that reason, you should do the following instead:

1. Register your local (publisher) geodatabase and server (Amazon) geodatabase with ArcGIS for Server.
2. Choose to create a geodata service from the server database when you register it.
3. Set the registered data store for your ArcGIS for Server connection to use replication, with the parent being your local enterprise geodatabase and the enterprise geodatabase in Amazon being the child.
4. After edits are made, synchronize data changes.

You should be aware of any restrictions that exist when replicating specific types of data. See the following topics for more information:

Replication and related data
Replication and raster data
Replication and topology
Replication and geometric networks
Replication and terrains, network datasets, parcel fabrics, and representations

See Using a geodata service and a connected replica for instructions.

Use a geodata service and a connected replica

If you have data in an enterprise geodatabase on-site, you might want to replicate some of that data to an enterprise geodatabase in the cloud. To do so, you can use the workflow described in this topic.

The geodata geodatabase that the ArcGIS Server Cloud Builder on Amazon Web Services application creates is intended for use in a replicated data workflow. When you register the geodata database as a replicated data store with ArcGIS for Server, you can create a geodata service. Next, replicate data from your on-premises geodatabase to the geodata service. After edits are made, you can then synchronize through the geodata service.

In most cases, you will want to edit your on-premises geodatabase while leaving the data in your geodatabase in the cloud read-only to the users who access it through the Internet. To do that, you would create a one-way replica; edits made to the on-premises geodatabase are synchronized with the geodatabase in the ArcGIS Server on Amazon Web Services instance through the geodata service. However, you could also have an editable feature service that makes changes to the geodatabase in the ArcGIS Server on Amazon Web Services instance. In that case, you would create a two-way replica; edits made in either geodatabase get synchronized to the other geodatabase through the geodata service.

Prepare on-premises data for replication

Before you can replicate data, it must meet specific requirements:

- All spatial data must be stored in a high-precision spatial reference.
- The parent and child replica geodatabases should be at the same release, though it is possible to use an ArcGIS 10, 10.1, 10.2, 10.2.1, or 10.2.2 geodatabase for the parent and an ArcGIS 10.3 geodatabase to host the child. However, if you use two-way replication, be sure you do not add data or make any edits that introduce functionality to the child replica that is not supported in the parent replica's geodatabase.
- The owner of the data to be replicated must perform the following steps:

1. Grant write access on the data to the user who will create the replica.
2. Register as versioned the data to be included in the replica.
   The data must be fully versioned; it cannot be versioned with the option to move edits to base.
3. Add a global ID column to each dataset that will be part of the replica.

Start an ArcGIS Server on Amazon Web Services instance

If you have not done so already, launch an instance of ArcGIS for Server. See Build an ArcGIS site on Amazon Web Services for instructions. Be sure to specify a key pair and choose to include an enterprise geodatabase when setting up the site. The geodatabase can be on the same instance as ArcGIS for Server or its own separate instance.

Instances created through the ArcGIS Server Cloud Builder on Amazon Web Services application automatically have a geodatabase—geodata—intended for use as a replica geodatabase.

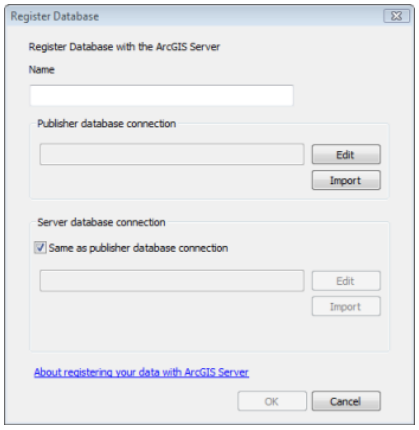
Connect to your ArcGIS Server instance

Create a GIS server publisher connection from ArcGIS for Desktop to your ArcGIS Server on Amazon Web Services instance. See Making a publisher connection to ArcGIS Server in ArcGIS for Desktop for instructions.

Register geodatabases and publish a geodata service

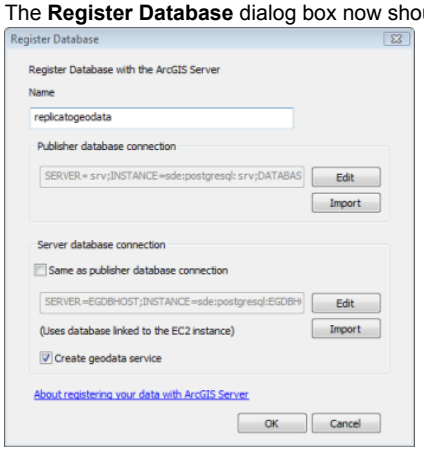
Register the geodata geodatabase as a replicated data store and create a geodata service. This is done from the **Server Properties** dialog box of the GIS server connection you created in ArcGIS for Desktop.

1. Right-click the GIS server connection and click **Server Properties**.
   The **ArcGIS Server Properties** dialog box opens.
2. Click the plus button (**+**) next to **Registered Databases** on the **Data Store** tab.
   The **Register Database** dialog box opens.



3. Type a name for the data store in the **Name** text box.
   This name will be used for the registered database list and the geodata service.
4. Click **Import** next to the **Publisher database connection** text box.
5. Browse to the location of the database connection file of your on-premises geodatabase and click **Select**.
   The **Publisher database connection** text box is populated with the connection information for your on-premises geodatabase.
6. Uncheck **Same as publisher database connection**.
   When connecting to an ArcGIS Server on Amazon Web Services instance, **Server database connection** will be populated with the connection to your geodata geodatabase on Amazon EC2 automatically.
7. Check **Create geodata service**.

   The **Register Database** dialog box now should look similar to the following:



8. Click **OK** to register your databases and create a geodata service.
   Your newly registered database appears in the **Registered Databases** list.
9. Click **OK** to close the **ArcGIS Server Properties** dialog box.
10. Double-click or refresh your ArcGIS Server connection to confirm that the geodata service was created.

Replicate through the geodata service using the Create Replica wizard

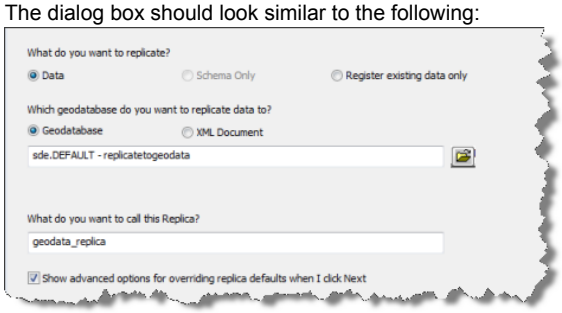
The geodata geodatabase in your ArcGIS Server on Amazon Web Services instance must contain replicas of the data you want to keep synchronized with your on-premises data. To replicate the data to the geodata registered database, use the Create Replica wizard in ArcMap.

1. Start ArcMap.
2. Add to the map the data from your on-premises enterprise geodatabase that you want to replicate to your geodatabase in Amazon EC2.

If you only want one area of the data to be published, zoom to the extent of that area or draw a box on the map that includes all the data you want to replicate. Otherwise, you can replicate all the data or specify an extent envelope when you replicate the data.

3. Open the **Distributed Geodatabase** toolbar.
   Open the toolbar by clicking **Customize** > **Toolbars** > **Distributed Geodatabase**.
4. Click the **Create Replica** button on the **Distributed Geodatabase** toolbar.
5. Choose what type of replica to create.
   - If you will be pushing edits only from your on-premises geodatabase to your geodatabase in Amazon EC2, choose **One way replica**, leave **Parent to child** selected, then click **Next**.
   - If you will be pushing edits from your on-premises geodatabase and will allow users to edit the data in your geodatabase in Amazon EC2, choose **Two way replica** and click **Next**.
6. Click **Next**.
7. Choose **Data** and **Geodatabase** to replicate your data to the geodatabase.
8. Click the **Open** button, navigate to your geodata service, then click **OK**.
9. Type a name for your replica.
10. Check **Show advanced options for overriding replica defaults when I click Next**.

    The dialog box should look similar to the following:



    Tip: If you are using all the default settings, you can uncheck **Show advanced options for overriding replica defaults when I click Next**.

11. Click **Next**.
12. Choose which replication model to use.
    - If the data you are replicating participates in geodatabase functionality, such as geometric networks, topologies, or relationship classes, click **Full Model**. (This is the default model.)
    - If your data does not use any geodatabase functionality and you do not need the data in the child replica to be versioned, choose **Simple Model**.
13. Click **Next**.
14. Choose the extent of the data to replicate.
    - If you want only the data that appears in the current map extent, choose **The current display extent**. (This is the default data extent to replicate.)
    - If you want the full extent of the data in the ArcMap table of contents, choose **The full extent of the data**.
    - If you drew a graphic around the data in the map and want to replicate only the data that is inside that graphic, choose **The boundary of the currently selected graphic**.
    - If you want a specific extent, choose **The following extent** and type the values of the extent you want.
15. Check the **Include** box for the data you want to replicate.
16. If there are relationship classes in the data, you have the option to include or exclude the related data. By default, related data is replicated. If you don't want to include it, uncheck **Replicate related data**.
17. Click **Next**.

18. If you included related data, define how related objects are added to the replica—either with a forward direction (data added to the origin class is added to the destination class) or reverse direction (data added to the destination class is added to the origin class).
19. Click **Next**.
20. You can click **Summary** to see all the information you provided for creating the replica. Click **OK** to close the summary when you have finished reviewing it, or click **Back** if you need to make any changes.
21. If no changes are needed, click **Finish**.

The data is replicated through the service to the geodata geodatabase.

Edit data in the on-premises geodatabase

Edit data in your on-premises enterprise geodatabase as you normally would. If you edit in other geodatabase versions, changes must be posted to the parent version if you want those edits to be synchronized with the child replica.

Synchronize edits

You can synchronize after a set of edits or synchronize on a specific schedule, for example, at the end of each day. Once the edits you want to synchronize are completed and have been posted to the parent version, you can use the **Synchronize Changes** button on the **Distributed Geodatabase** toolbar or script synchronization through your geodata service.

When you synchronize data, you must connect as the same database user that created the replica or as the geodatabase administrator.

For more information on synchronization, see Synchronizing connected replicas in the ArcGIS help.

Synchronize in ArcMap

In this workflow, you have a map open that contains the replicated data that has been edited.

1. Open the **Distributed Geodatabase** toolbar
   Open the toolbar by clicking **Customize** > **Toolbars** > **Distributed Geodatabase**.
2. Click the **Synchronize Changes** button on the **Distributed Geodatabase** toolbar.
   The **Synchronize Changes Wizard** dialog box opens.

   All the information on the **Synchronize Changes Wizard** dialog box is provided automatically. You only need to alter information on this wizard if you have multiple replicas, you re-created your GIS server connection and need to specify a new relative replica, or you are using two-way replication and want to replicate from the geodata service to your on-premises geodatabase.
3. When you confirm the information on the wizard, click **Finish** to start the synchronization.

Script synchronization

You can write a script to synchronize your geodatabases. See How to synchronize a replica in a connected environment in the ArcObjects SDK for the Microsoft .NET Framework for a sample script.

Related topics

Replication to an Amazon Web Services instance using geodata services

Deploy web applications on Amazon Web Services

Strategies for web application deployment on Amazon Web Services

It's likely that you've developed a set of web applications that use your ArcGIS web services. There are several approaches you can take to host these web applications using Amazon Web Services.

Host the application on Amazon S3

You can host static web applications on Amazon Simple Storage Service (S3). In other words, the app must not use any server-side scripting languages such as PHP or JSP. This is an easy and relatively inexpensive way to host a web application, and it allows you to take advantage of the Amazon CloudFront content delivery service. See Web applications on Amazon S3 to learn more.

Host the application on an EC2 instance in your ArcGIS Server site

Another option is to host your web application on one of the EC2 instances in your ArcGIS Server site. If you take this approach, you should use the instance tagged as SITEHOST in the AWS Management Console. This is the instance that is not terminated when the site is stopped. It's also the instance that is preserved when you make a site template or backup.

The IIS and Apache web servers are available on the ArcGIS Server Windows and Linux instances, respectively. You can optionally install the Web Adaptor on your EC2 instances, but this is often not necessary.

See Deploy a web application on a Windows EC2 instance and Deploy a web application on an Ubuntu Linux EC2 instance to learn more.

Host the application on an EC2 instance apart from your ArcGIS Server site

You can launch an EC2 instance and use it for the sole purpose of hosting web applications. You might take this approach if you cannot host the application on S3, but you want to separate the web application tier from the GIS web service tier. In this scenario, you'll need to make sure that your Amazon security group(s) allow the necessary communication between all your instances.

Web applications on Amazon S3

If your web application is static, it may be easier or cheaper to host it on Amazon Simple Storage Service (S3) instead of an EC2 instance. In S3, you can store sets of files in web-accessible folders, called buckets. S3 allows you to designate any bucket as a website. You define an index document (in other words, a start page) and a policy stating who can access the site. You can then interact with your pages from the bucket as if they were hosted on a traditional web server.

Not all web applications are appropriate for S3. If your app uses a server-side scripting language such as PHP, JSP, or ASP.NET, then you should host the app on your own EC2 instance. However, client-side scripting languages such as JavaScript are appropriate for S3. Applications built with the out-of-the-box Esri Web APIs and viewers should work on S3.

A benefit of hosting your page in S3 is the potential to use the Amazon CloudFront delivery service. This is an Amazon web service that hosts your content on various servers around the world, optimizing your file delivery speed among geographically dispersed users.

For full information about how to host a website on S3, see the Amazon Web Services documentation.

Deploy a web application on a Windows AWS instance

Deploying a web application on your Windows ArcGIS Server site on Amazon Web Services (AWS) requires some preparation. Follow these steps to expose your application in a secure and stable manner. You will need to perform some of these steps in the AWS Management Console or a similar third-party client to Amazon Web Services.

1. Create a site using ArcGIS Server Cloud Builder on Amazon Web Services.
2. Add rules to your site's Amazon security group to allow the following:
   - Remote Desktop (RDP) access through port 3389 for your IP address. This allows you to log in and configure your instance.
   - HTTP access (typically through port 80) for the IP address range of your choosing.
3. Using Windows Remote Desktop, log in to the ArcGIS Server AWS instance that is marked as SITEHOST when you view your list of instances in the AWS Management Console. Unless otherwise noted, you'll perform the remainder of these steps while logged in to this instance.
4. Perform the following substeps to start and configure the Internet Information Services (IIS) web server that is included on your instance:
   a. Open **Control Panel** > **Administrative Tools** > **Services**.
   b. Right-click **World Wide Web Publishing Service** and click **Properties**.
   c. Set **Startup type** to **Automatic** and click **OK**.
   d. Right-click **World Wide Web Publishing Service** and click **Start** if it is not started already.
   e. Start IIS Manager.
   f. In the left menu, expand your server node, expand **Sites**, and click **Default Web Site**.
   g. Click the **Start** link.
5. Download and install the ArcGIS Web Adaptor (IIS) from My Esri. Follow the steps in the installation guide to install and configure the Web Adaptor. Configure it to communicate through port 80.

   Note: This step is only required if you are deploying a JavaScript application, because a JavaScript application must access web services from the same origin (web server) on which the application is deployed. If you're deploying a Flex or a Silverlight application, you don't have to install the Web Adaptor.
6. Deploy your web application by copying the web application folder to IIS's root folder `C:\Inetpub\wwwroot`.
7. Using the AWS Management Console or another client to AWS, allocate an Elastic IP address, and associate it with the site server instance in your site (remember this is the instance to which you logged in).
   The Elastic IP address is necessary because Amazon Web Services changes the machine name whenever you stop and start a site. The Elastic IP gives you a constant address that you can use for accessing your GIS server.

   Incoming requests to your web app will go through this Elastic IP, not the Elastic Load Balancer.
8. Optionally, communicate with your network administrator to register a domain name for your site.
   In most cases, you'll want to create a domain name such as mymaps.mycity.gov instead of sending users directly to your Elastic IP. Provide your Elastic IP address to your network administrator and he or she can associate a domain name that points traffic to your site.
9. Update any URLs in your web application code.
   - **JavaScript applications**—The URLs must use your newly defined domain name. For example, `http://mymaps.mycity.gov/arcgis/rest/services/MyService/MapServer`.
   - **Flex and Silverlight applications**—Use the URLs containing the address of the Elastic Load Balancer (ELB) that Cloud Builder placed on your site. For example, `http://<ELB address>/arcgis/rest/services/MyService/MapServer`. Flex and Silverlight can access web services through the ELB address because client access policy files were automatically added to the site.

Deploy a web application on an Ubuntu Linux AWS instance

Deploying a web application on your Linux ArcGIS Server site on Amazon Web Services (AWS) requires some preparation. Follow the steps below to expose your application in a secure and stable manner. You will need to perform some of these steps in the AWS Management Console or a similar third-party client to Amazon Web Services.

1. Create a site using ArcGIS Server Cloud Builder on Amazon Web Services.
2. Add rules to your site's Amazon security group to allow the following:
   - SSH access (typically through port 22) for your IP address. This allows you to log in and configure your instance.
   - HTTP access (typically through port 80) for the IP range of your choosing.
3. Using SSH, log in to the site server instance. This is the instance containing the configuration store. It is marked as SITEHOST when you view your list of instances in the AWS Management Console. Unless otherwise noted, perform the remainder of these steps while logged in to this instance.
   To log in, enter the following command:

   ```
   ssh -i <your key pair file> ubuntu@<Public DNS of your AWS instance>
   ```
4. Start Apache using the following command:
   ```
   sudo service apache2 start
   ```
5. Configure Apache so that it starts whenever the operating system starts. Use the following command:
   ```
   sudo update-rc.d apache2 defaults
   ```
6. Using the AWS Management Console or another client to AWS, allocate an Elastic IP address and associate it with the site server instance in your site (remember, this is the instance to which you logged in).
   The Elastic IP address is necessary because Amazon Web Service changes the machine name whenever you stop and start a site. The Elastic IP gives you a constant address that you can use for accessing your ArcGIS Server.
7. Deploy your web application by copying it to Apache's root folder, `/var/www/`.
8. Optionally, communicate with your network administrator to register a domain name for your site.
   In most cases, you'll want to create a domain name such as mymaps.mycity.gov instead of directing users through your Elastic IP. Provide your Elastic IP address to your network administrator and he or she can associate a domain name that points traffic to your site.
9. Update any URLs in your web application code.
   - **JavaScript applications**—The URLs must use your newly defined domain name. For example, `http://mymaps.mycity.gov/arcgis/rest/services/MyService/MapServer`.
   - **Flex and Silverlight applications**—Use the URLs containing the address of the Elastic Load Balancer (ELB) that Cloud Builder placed on your site. For example, `http://<ELB address>/arcgis/rest/services/MyService/MapServer`. Flex and Silverlight can access web services through the ELB address because client access policy files were automatically added to the site.

Secure ArcGIS Server on Amazon Web Services

ArcGIS Server security on Amazon Web Services

Devising a comprehensive ArcGIS Server security strategy on Amazon EC2 requires you to plan for security at different levels. Consider the following questions:

- Who should be able to create and destroy ArcGIS Server sites using my Amazon account?
- Who should be able to log in to my EC2 instances to install new software or directly administer the server?
- What computers should be able to discover my server once it is running on EC2, and for which purposes?
- Who should be able to connect to my site as a user, publisher, or administrator?
- Are there some users who need to be allowed access to certain services and denied access to others?
- Will my web applications require a login?

You'll need to understand and use a variety of security techniques to make a secure solution that answers all the above questions in a satisfactory way. This topic describes how you could approach each.

Secure your cloud administration environment

Amazon Identity and Access Management (IAM) allows you to manage groups of users who have various levels of permissions to your AWS account. Before you can log in to Cloud Builder, you must use IAM to create at least one user with administrator access to your account. You will then need to download the Access Key and Secret Access Key associated with that user. See Frequently asked questions to learn how to do this. When you first log in to Cloud Builder, you can decide whether to save these keys or require them at every login.

Advanced administration of ArcGIS Server on Amazon Web Services is performed using the AWS Management Console. You must log in to the console with your Amazon account name and password before you can launch or terminate EC2 instances, configure Amazon Elastic Load Balancers (ELBs) and Elastic IPs, and perform other administrative functions of the virtual environment. Logging in also lets you view your account activity and billing information.

Only share your Amazon account name, password, Access Keys, and Secret Access Keys with a small number of people in your organization who understand how to properly launch, edit, and terminate resources using the Cloud Builder or AWS Management Console. Allowing widespread access to untrained personnel makes your deployment vulnerable to severe system disruption and excessive charges on your account. These types of problems may ultimately be more damaging than an assault from an external hacker.

Amazon offers an optional layer of protection for the AWS Management Console beyond your account name and password. This option, AWS Multi-Factor Authentication, requires you to have a six-digit code generated by a small hardware device in your possession. The code frequently changes, such that if a malicious user were to obtain your account name and password, he or she would still not be able to log in to the AWS Management Console.

Secure instance administration

Logging in to the Cloud Builder or AWS Management Console is just one aspect of ArcGIS server administration on Amazon EC2. Another part of setting up your cloud deployment is logging in to your EC2 instances to transfer data and configure GIS services and applications.

You initially log in to your Windows EC2 instance as the machine administrator, using a randomly generated password that you retrieve using your key pair file. Keep your key pair file in a secure location. Then, the first time you log in to the instance, you should change the password to something easier to remember. It is not secure to write down the password or store it in clear text somewhere on your local machine.

Tip: Consider choosing a password that corresponds to the Windows Server 2012 complexity requirements, which include the following:

- Passwords should not contain the user's account name or parts of the user's full name that exceed two consecutive characters.
- Passwords should be at least eight characters in length.
- Passwords should contain characters from three of the following four categories:
  - English uppercase characters (A through Z)
  - English lowercase characters (a through z)
  - Base 10 digits (0 through 9)
  - Nonalphanumeric characters (for example, !, $, #, %)

Once you've logged in to the instance, you can optionally use Windows tools to define nonadministrative users who can log in.

Secure instances against outside attacks

All EC2 instances use a firewall to protect against inappropriate or unknown outside access. You configure the firewall by creating security groups and opening access to a range of IP addresses, ports, and protocols on each group. Every time you launch a new EC2 instance, you need to specify which security group the instance will belong to.

By default, new security groups have no access allowed. At a minimum, you need to allow remote desktop access and HTTP access to log in to your EC2 instance and test your server. See Open an Amazon EC2 security group for ArcGIS for Server for instructions. Also, see Common security group configurations for ideas of security group settings that are appropriate for ArcGIS Server on Amazon Web Services.

When you use ArcGIS Server Cloud Builder on Amazon Web Services to create a site, a security group is created and configured for you. The necessary ports are opened on the security group to allow the site to function, but if needed you can use the AWS Management Console to fine-tune the settings of this security group. For example, if you want to log in to one of the instances using Windows Remote Desktop, you need to open port 3389.

The Amazon Security Center contains white papers and best practice documents for designing a secure architecture for EC2. These guidelines are applicable to ArcGIS Server on Amazon Web Services.

Secure GIS web applications and services

Access to your web services and applications is managed through the same security mechanisms that you use with ArcGIS Server outside Amazon EC2. These are described in the ArcGIS Help book Securing your ArcGIS Server site.

The **Security** tab of ArcGIS Server Manager helps you configure users and roles and choose which users and roles have access to your services. ArcGIS Server has a built-in user and role store that can be an attractive option on a cloud-based site that cannot reach user stores on your local network.

Open Amazon EC2 security group for ArcGIS Server

Amazon provides security groups that allow you to specify who can connect to your EC2 instances. When you build a site using ArcGIS Server Cloud Builder on Amazon Web Services, a security group is created for you, and HTTP access is granted. However, if you intend to work with your EC2 instances using Remote Desktop Connection or SSH, you must add rules allowing those types of connections.

Tip: If you are building a site and Cloud Builder detects that you have a security group named arcgis-<site name>, Cloud Builder applies that security group instead of creating a new one. This behavior means that you can potentially create and configure a security group as described below before you build a site.

If you are building your site manually using the AWS Management Console, you must create a security group yourself and add Remote Desktop and SSH rules. Additionally, you must add an HTTP access rule for users to access your web services. Finally, you need to allow all instances in your security group to access each other. This entire process is described below.

1. Sign in to the AWS Management Console and display the page for the EC2 region hosting your site.
2. On the left pane, click **Security Groups**.
3. Click the box next to the security group you want to modify, and click the **Inbound** tab to examine the list of allowed connections.
4. Click **Edit** to alter the list of inbound connections allowed.
   The **Edit inbound rules** dialog box opens.
5. Click **Add Rule**.
   A new line is added to the bottom of the inbound rules.
6. If you are using a Windows instance, use the drop-down lists and text boxes to add **RDP** as an allowed connection. This opens port `3389`. You'll also need to supply a range of IP addresses that are allowed to make this connection, using Classless Inter-Domain Routing (CIDR) notation. For example, `0.0.0.0/0` allows everyone to connect (not recommended for security reasons), whereas `92.23.32.51/32` allows one specific IP address to connect.
7. If you are using a Linux instance, use the drop-down lists and text boxes to create a new **Custom TCP rule** allowing access to port `22` from an approved IP address or range of IPs. This allows you to interact with your instance through SSH.

   Note: If you built your site using ArcGIS Server Cloud Builder on Amazon Web Services, the next three rules were added automatically. You can click **Apply Rule Changes** and exit this topic.

8. Click **Add Rule**, and add a **Custom TCP rule** with port `6080` as an allowed connection. Optionally, specify a range of IP addresses that are allowed to make this connection.
9. If you'll be using an encrypted connection, click **Add Rule** and add a **Custom TCP rule** with port `6443` as an allowed connection. Optionally, specify a range of IP addresses that are allowed to make this connection.
10. Click **Add Rule** and add a rule to allow all EC2 instances within your group full access to each other. To do this, choose **All ICMP**. Then, in the **Source** text box, type the Group ID of the security group that you are currently editing (for example, sg-xxxxxxxx). If you don't know the ID of your security group, you can switch back to the **Details** tab to see it, but be aware that this will erase the other rules you've set if you have not yet clicked **Save**.
11. If you have not yet done so, click **Save**. Your rule changes take effect immediately.

See Common security group configurations to learn more about these security rules and when to adjust them.

Common security group configurations

An Amazon Elastic Compute Cloud (EC2) instance can only allow network traffic from sources and ports defined in its security group. When you use Amazon EC2, you need to set up some security groups that correspond to the types of things you'll be doing with your EC2 instances. This topic describes some common security groups you can configure for different ArcGIS Server deployments.

By default, a security group is completely locked down. You add rules to a security group specifying the type of traffic allowed, the ports it will be allowed through, and the computers from which communication will be accepted. The ports you decide to open and the type of traffic you need to allow depend on what you are doing with the instance.

The following are suggestions of security group names and rules that you can configure in the AWS Management Console. Allowed ports and protocols may vary based on your organization's IT policies. The suggestions below use the most common port numbers. If your organization has an IT specialist, consider consulting with him or her to devise the best security strategy for your EC2 instances.

ArcGIS Server Development

Consider creating a security group specifically for EC2 instances that are being used for development and testing purposes. This type of group could allow the following access:

- **RDP access through port 3389 for your IP address or a range of approved IP addresses within your organization (Windows only)**. This allows you to administer your EC2 instance through Windows Remote Desktop. You must use Classless Inter-Domain Routing (CIDR) notation to specify a range of IP addresses (or one IP address) that can make connections. For example, 0.0.0.0/0 allows everyone to connect, whereas 92.23.32.51/32 allows one specific IP address to connect. Check with your system administrator if you need help obtaining the external-facing IP address of your local machine.
- **TCP access through port 22 for your IP address or a range of approved IP addresses within your organization (Linux only)**. Opening port 22 allows you to work with your Linux instances through SSH.
- **TCP access through port 6080 for everyone (if not using an Elastic Load Balancer) or the Elastic Load Balancer's security group (if using an Elastic Load Balancer)**. Port 6080 is used for communication with ArcGIS Server. If you're not putting an Elastic Load Balancer in front of your site, then you need to open port 6080 to everyone who will use your ArcGIS Server web services. If you're using an Elastic Load Balancer, you need to open port 6080 to the Elastic Load Balancer's security group (which is discoverable through the AWS Management Console and is most likely a value such as `amazon-elb/amazon-elb-sg`).
- **Access from other machines in this group**. This is required in order for the GIS server machines to communicate with each other. It also facilitates file sharing. You can add a rule permitting this type of access by choosing the **All ICMP** rule type, entering your security Group ID (for example sg-xxxxxxxx) in the **Source** box, and clicking **Add Rule**. When you take this approach, the machines in your group can communicate with each other through all ports and protocols.

ArcGIS Server Production

Once you've developed and tested your application and are ready to move it to a production tier, it's a good idea to disable remote desktop access. If a problem occurs and you need to log in to the machine, you can temporarily change the security group configuration to allow yourself access. An ArcGIS Server Production group could allow the following access:

- **TCP access through port 6080 for everyone (if not using an Elastic Load Balancer) or the Elastic Load Balancer's security group (if using an Elastic Load Balancer)**
- **Access from other machines in this group**

ArcGIS Server Production Secure

If you want to require encrypted communication with your machine, you should configure an Elastic Load Balancer on your site that receives traffic through port 443, the port typically used for encrypted communication through SSL. Then configure the load balancer to forward traffic to port 6443. On your security group, open the ports described above for ArcGIS Server Production.

Enterprise Geodatabase

If you choose to have enterprise geodatabases on a separate instance from your ArcGIS Server instance, You can configure a security group specifically for your enterprise geodatabase instance that allows the following:

- **TCP access through port 22 (Linux) for your IP address or a range of approved IP addresses within your organization** You need to remotely connect to your machine at least once to change the PostgreSQL default passwords. After that, you can remove this remote access rule from the security group if you want.

- **RDP access through port 3389 (Windows)** You can add this rule if you need to remotely connect to your SQL Server or SQL Server Express instance—for example, to add users or additional geodatabases—then remove it when you finish.
- **Access from machines in your ArcGIS server security group** This allows your instances running ArcGIS for Server to view your enterprise geodatabase instance. If machines not participating in your security groups need to connect to your geodatabase, you need to explicitly open port 5432, which allows communication with PostgreSQL.

Commonly used ports

Following are some of the most common ports you may work with as you create security groups. Some of these ports you may not need to explicitly open; rather, you may just decide to give machines within your security group full access to each other. If you want to allow access from machines not participating in your security groups (for example, your desktop workstation in your office), you need to open specific port numbers.

Port	Common purpose
80	HTTP access to IIS web server or load balancer
443	HTTPS access to IIS web server or load balancer
445	Windows file sharing
1433	Connections to Microsoft SQL Server
3389	Connections to Windows Remote Desktop
5432	Connections to PostgreSQL
6080	HTTP access to ArcGIS Server
6443	HTTPS access to ArcGIS Server

Windows Firewall is enabled on any instance that you launch using the Esri-provided AMIs, including on sites that you build with Cloud Builder. If you install a third-party application that requires ports other than those listed above, ensure that Windows Firewall is also configured to allow the port.

Windows Firewall and the ArcGIS Server AMIs

Windows Firewall is enabled and preconfigured when you build a Windows-based ArcGIS Server site using ArcGIS Server Cloud Builder on Amazon Web Services or the AWS Management Console. All ports that are necessary for ArcGIS Server to run are open to inbound connections.

Outbound connections are allowed by default.

Legacy: In the ArcGIS 10.1 AMIs, outbound connections were blocked in Windows Firewall by default.

Windows Firewall and Amazon security groups

The Amazon security groups provide protection against unsolicited incoming traffic. In this role, they complement Windows Firewall. To completely open a port to inbound traffic, the port must be allowed by both the Amazon security group and Windows Firewall.

Set up SSL using Cloud Builder

In some cases you may want to require encrypted communication with your ArcGIS Server site using HTTPS. This requires that your Elastic Load Balancer (ELB) and ArcGIS site use an SSL certificate. The Elastic Load Balancer (ELB) must use an SSL certificate you obtain from a trusted Certificate Authority (CA). ArcGIS Server Cloud Builder on Amazon Web Services can install your SSL certificate into the ELB for you at the time you create a site. Also, by default, it enables SSL on the ArcGIS Server instances using the out-of-the-box, self-signed certificate. You may continue to use this certificate for the ArcGIS Server or replace it with a new CA-issued certificate.

On the **Security** panel of Cloud Builder, you will find options for uploading and installing SSL certificates on your site.

Upload a new certificate for the ELB

Amazon Web Services (AWS) allows you to upload and store SSL certificates in the cloud as part of its Identity and Access Management (IAM) service. You don't have to learn how to use this service directly, because Cloud Builder provides a front end to it. Using Cloud Builder, you can upload one or more SSL certificates to AWS IAM; then you can choose to apply any one of those certificates whenever you build a site. The certificate will be installed for you.

You may upload an existing SSL certificate or create a new SSL certificate. You must ensure that your SSL certificate meets the criteria mentioned in the **To update an SSL certificate for an HTTPS load balancer** section in the topic Update an SSL Certificate for a Load Balancer. When generating the ELB's SSL certificate, you must also ensure that the common name used matches the public DNS (hostname) of the ELB. To generate the private and public keys for a new SSL certificate, you will need to use an SSL management tool or software product such as OpenSSL.

Note: An SSL certificate created using the ArcGIS Server Administrator Directory is managed in an internal read-only keystore and cannot be exported for use with the Amazon ELB.

To upload a new SSL certificate and install it on the ELB, do the following:

1. Copy the private and public keys corresponding to your SSL certificate to a folder on the computer running Cloud Builder.
2. Start creating or updating a site using Cloud Builder.
3. In the **Security** panel of Cloud Builder, check **Install SSL certificate**.
4. From the **Choose SSL certificate** drop-down list, choose **<Upload certificate>**.
5. Supply the **Certificate name** by entering a unique name for the SSL certificate. Do not include the path in this value.
6. Supply the location of the **Private key** corresponding to the SSL certificate you want to upload. The private key must be an RSA key in PEM-encoded format.
7. Supply the location of the **Public key certificate** corresponding to the SSL certificate you want to upload. The public key must be in PEM-encoded format.
8. Click **Upload**.
9. In the **Choose SSL certificate** drop-down list, ensure your new certificate is selected.

Use an existing certificate for the ELB

If you've already uploaded a certificate using Cloud Builder, you can do the following to install it on a site:

1. Start creating or updating a site using Cloud Builder.
2. In the **Security** panel of Cloud Builder, check **Install SSL certificate**.
3. From the **Choose SSL certificate** drop-down list, choose your certificate name.

Configure the ELB health check in HTTPS-only scenarios

When you select an SSL certificate, Cloud Builder configures your site so that it can receive both HTTP and HTTPS requests. If you later modify ArcGIS Server so that it is only allowed to receive HTTPS requests, you must update the Elastic Load Balancer (ELB) health check using the following steps:

1. Log in to the AWS Management Console and display the page for the EC2 region where your site resides.
2. Click **Load Balancers**.
3. Check the check box next to the load balancer named arcgis-<your site>.
4. In the lower panel, click the **Health Check** tab.
5. Click **Edit Health Check**.

6. Change the **Ping Protocol** to **HTTPS**.
7. Change the **Ping Port** to `6443` and click **Save**.

Apply ArcGIS updates on Amazon Web Services

ArcGIS updates on Amazon Web Services

Esri releases new ArcGIS Server Cloud Builder on Amazon Web Services AMIs and an updated Cloud Builder application shortly after each new ArcGIS release. You should use the latest version of Cloud Builder when you build new sites. See Cloud Builder upgrades to learn when and how to upgrade Cloud Builder.

To update existing sites, you must log in to your EC2 instances and apply the update directly. To minimize downtime, you can use Cloud Builder's site templates to launch a copy of your site. Once you have applied the update on the copy, you can redirect traffic into your upgraded site and delete the old site.

If you have a multiple-machine site, you must also create a template of your upgraded site and use the template to launch a new site with the desired number of machines.

The help topics in this section provide instructions for upgrading single-machine and multiple-machine sites.

- Apply an ArcGIS update to a single-machine site
- Apply an ArcGIS update to a Windows multiple-machine site
- Apply an ArcGIS update to an Ubuntu multiple machine site

If you don't want to apply the update on an existing instance, you can either maintain your site at its current version, or you can launch a new site from the latest version of Cloud Builder and redeploy your services and applications on it.

If you did not build your site using Cloud Builder, you should log in to the EC2 instances and apply the update directly. If you need to minimize downtime, you can save an Amazon Machine Image (AMI) of your instance and use it to launch a copy of your instance. You can then apply the update on the copy.

Cloud Builder updates

New versions of Cloud Builder are currently released at the time of every ArcGIS update. The most recent version of Cloud Builder creates new sites with the most recent ArcGIS update applied, unless you are creating a site from a template; when you create a site from a template, the version of software used in the template is applied to the new site.

You can download the latest version of Cloud Builder from My Esri. You can access the download if you have purchased ArcGIS for Server.

Update notices and restrictions

If Cloud Builder detects that you are not running its latest version, you'll see a message that an upgrade (in other words, an update) is available. Get the update only after you have determined that you are finished creating new sites at your current ArcGIS version. Once you have accessed your Amazon account using a particular version of Cloud Builder, you cannot go back and access your account using an older version of Cloud Builder.

When you update Cloud Builder, the version numbers of your existing sites and templates are not affected. You can still manage your existing sites and templates in the updated Cloud Builder. It's possible to see sites and templates of differing versions listed side by side in Cloud Builder.

Apply an ArcGIS update to a single-machine site

It's possible to download ArcGIS updates onto your EC2 instance and apply them directly to ArcGIS for Server and ArcGIS for Desktop.

To minimize downtime, it's recommended that you apply the update on a copy of your site using a site template. The optional step 1 below explains how to do this. Once you have applied the update, you can redirect traffic into your updated site.

Use the steps below to update a site only if you are certain that it will stay a single-machine site. If there is any chance that your site will need to expand to include multiple GIS servers, you need to apply the update on two instances and create a site template to launch your final updated site, as described in Apply an ArcGIS update to a Windows multiple-machine site and Apply an ArcGIS update to an Ubuntu multiple machine site.

1. Optionally, to minimize downtime during the update and test the update on a separate instance, use ArcGIS Server Cloud Builder on Amazon Web Services to create a template of your site. Then launch a new site from the template.
   You'll apply the update to this copy of your site. Once you are ready, you can switch your incoming traffic to the updated site.
2. Install the version of ArcGIS Server Cloud Builder on Amazon Web Services that corresponds to the update you want to apply. Be aware that when you do this, you will not be able to access your sites with an older version of Cloud Builder.
3. Download the updates for ArcGIS for Desktop (Windows only), ArcGIS for Server, and optionally the ArcGIS Web Adaptor. You can download them directly onto the EC2 instance or you can copy them onto the instance from another machine.
4. Set your ArcGIS Server account password to a known value. By default this password is random, but when you apply the update, you need to enter the password. If you need instructions on how to reset the password of a Windows operating system login, see the Microsoft documentation.
5. Install the updates in the following order: ArcGIS for Desktop (Windows only), ArcGIS for Server, and ArcGIS Web Adaptor. Further information about installing the updates is available in the installation guides.

   Note: Beginning with ArcGIS 10.3, ArcGIS for Server and ArcGIS for Desktop must be at the same version. Uninstall or upgrade the previous version of ArcGIS for Desktop before applying ArcGIS for Server updates.

6. If your site includes enterprise or workgroup geodatabases, upgrade the geodatabases. For instructions, see the topic for the type of geodatabase you are using:
   - Upgrade geodatabases in PostgreSQL in ArcGIS Server for Amazon Web Services
   - Upgrade enterprise geodatabases in SQL Server in ArcGIS Sever for Amazon Web Services
   - Upgrade workgroup geodatabases
7. If you performed step 1 to launch your site from a template, switch your organization's address mapping to point at your updated site's URL. If your apps were referencing the site URL directly, you'll need to update any URLs in your apps to incorporate your new Elastic Load Balancer (ELB) address. You can get the ELB address by looking at the **Manager URL** in the list of site details in Cloud Builder.
   Once you verify that the updated site is working, you can delete your original site and the template you created if you performed step 1.

Related topics

Apply an ArcGIS update to a Windows multiple-machine site
Apply an ArcGIS update to an Ubuntu multiple-machine site
Cloud Builder updates

Apply an ArcGIS update to a Windows multiple-machine site

This topic describes how to apply an ArcGIS update to a Windows multiple-machine ArcGIS Server site running on Amazon Web Services (AWS). In the context of this topic, a multiple-machine site is a site that includes more than one EC2 instance running the GIS server component.

To upgrade the site, you need to apply the update on two machines, then use those machines to create a template from which you launch your final updated site.

To minimize downtime, it's recommended that you apply the update on a copy of your site, which you can also make using site templates. The optional step 2 below explains how to do this. Once you have applied the update, you can redirect traffic into your updated site.

If you do not want to perform the steps below, you can launch a new site using the latest version of Cloud Builder and redeploy your data, services, and applications on it.

Follow the steps below to update a multiple-machine site to the latest version of ArcGIS. Do not stop your site at any time while performing these steps.

1. Install the version of ArcGIS Server Cloud Builder on Amazon Web Services that corresponds to the update you want to apply. Be aware that when you do this, you will not be able to access your sites with an older version of Cloud Builder.
2. Optionally, to minimize downtime during the update, use Cloud Builder to create a template of your site. Then launch a new site from the template.
   When you launch this site, configure it to have exactly two EC2 instances running the GIS server component. In other words, you should set the **Number of instances** property to `2` and uncheck **Enable auto-scaling**.

   You'll apply the update to this copy of your site. Once you are ready, you can switch your incoming traffic to the updated site.
3. If you did not perform step 2 above, edit your site in Cloud Builder so that it contains exactly two EC2 instances. In other words, you should set the **Number of instances** property to `2` and uncheck **Enable auto-scaling**.
4. Log in to the EC2 instance containing the ArcGIS Server configuration store and server directories. This instance is marked **SITEHOST** in the AWS Management Console.
5. On the SITEHOST, download the updates for ArcGIS for Desktop, ArcGIS for Server, and (optionally) ArcGIS Web Adaptor.
6. On the SITEHOST, set your ArcGIS Server account (arcgis) password to a known value. By default this password is random, but when you apply the update, you need to enter the password. If you need instructions on how to reset the password of a Windows operating system login, see the Microsoft documentation.
7. On the SITEHOST, install the updates in the following order: ArcGIS for Desktop, ArcGIS for Server, and ArcGIS Web Adaptor. Further information about installing the updates is available in the installation guides.

   Note: Beginning with ArcGIS 10.3, ArcGIS for Server and ArcGIS for Desktop must be at the same version. Uninstall or upgrade the previous version of ArcGIS for Desktop before applying ArcGIS for Server updates.
8. On the other instance, download updates for ArcGIS for Desktop, ArcGIS for Server, and (optionally) ArcGIS Web Adaptor.
9. On the other instance, install the updates for ArcGIS for Desktop.
10. On the other instance, uninstall the existing ArcGIS for Server.
11. On the other instance, set your ArcGIS Server account (arcgis) password to the same password you set on SITEHOST.

    Note: Be sure to set the ArcGIS Server account password to the same value on all instances.
12. Install the new version of ArcGIS for Server.
    Be sure to provide the arcgis user's new password.

    For authorization, you can use the file in this location or use your new license file:

    ```
    C:\Program Files:\ArcGIS\Server\framework\etc\license.ecp
    ```
13. If you are using a web adaptor, install the updates for ArcGIS Web Adaptor.
14. Test and configure your updated site. Make sure it is working exactly the way you want.
    If you were to leave your site in its current state, the second EC2 instance would be lost the next time you stopped your site using Cloud Builder. Therefore, continue with the following steps to preserve your work as a template and launch a site with the desired number of instances.
15. Use Cloud Builder to create a template from your newly updated site.

16. Using the template you created in the previous step, launch a site with the desired number of EC2 instances and auto-scaling rules.
17. Switch your organization's address mapping to point at your updated site's URL. If your apps were referencing the site URL directly, you'll need to modify any URLs in your apps to incorporate your new Elastic Load Balancer (ELB) address. You can get the ELB address by looking at the **Manager URL** in the list of site details in Cloud Builder.
    Once you verify that the updated site is working, you can delete your original site and the templates you created in steps 2.

Related topics

Apply an ArcGIS update to an Ubuntu multiple-machine site
Apply an ArcGIS update to a single-machine site
Cloud Builder updates

Apply an ArcGIS update to an Ubuntu multiple-machine site

This topic describes how to apply an ArcGIS update to an Ubuntu multiple-machine ArcGIS Server site running on Amazon Web Services (AWS). In the context of this topic, a multiple-machine site is a site that includes more than one EC2 instance running the GIS server component.

To upgrade the site, you need to apply the update on two machines, then use those machines to create a template from which you launch your final updated site.

To minimize downtime, it's recommended that you apply the update on a copy of your site, which you can also make using site templates. The optional step 2 below explains how to do this. Once you have applied the update, you can redirect traffic into your updated site.

Follow the steps below to update a multiple-machine site to the latest version of ArcGIS. Do not stop your site at any time while performing these steps.

If you do not want to perform the steps below, you can launch a new site using the latest version of Cloud Builder and redeploy your data, services, and applications on it.

1. Install the version of ArcGIS Server Cloud Builder on Amazon Web Services that corresponds to the update you want to apply. Be aware that when you do this, you will not be able to access your sites with an older version of Cloud Builder.
2. Optionally, to minimize downtime during the update, use Cloud Builder to create a template of your site. Then launch a new site from the template.
   When you launch this site, configure it to have exactly two EC2 instances running the GIS server component. In other words, you should set the **Number of instances** property to `2` and uncheck **Enable auto-scaling**.

   You'll apply the update to this copy of your site. Once you are ready, you can switch your incoming traffic to the updated site.
3. If you did not perform step 2 above, edit your site in Cloud Builder so that it contains exactly two EC2 instances. In other words, you should set the **Number of instances** property to `2` and uncheck **Enable auto-scaling**.
4. Download the ArcGIS for Server update.
5. First, log in with the arcgis user to the EC2 instance containing the ArcGIS Server configuration store and server directories. This instance is marked **SITEHOST** in the AWS Management Console.
   To log in, type the following command:

   ```
   ssh -i <your key pair file> arcgis@<public DNS of AWS instance>
   ```

   Remember that to log in to an instance, you must open the SSH port in the site's security group.
6. Copy or ftp the update's tar file to the SITEHOST instance.
7. Untar the update's setup.
8. Run the installation.

   ```
   ./Setup
   ```

9. After you apply the update to SITEHOST, apply it to the second machine. Log in as the arcgis user to the second machine in your site.
   To log in, type the following command:

   ```
   ssh -i <your key pair file> arcgis@<public DNS of AWS instance>
   ```

10. Copy or ftp the update's installation files to the second machine instance.
11. Uninstall the current ArcGIS Server installation.
    Change directories to the current ArcGIS Server installation, and run the uninstall_ArcGISServer script.

    ```
    cd /arcgis/server
    ./uninstall_ArcGISServer
    ```

12. Change directories to the location of the ArcGIS Server setup you downloaded and run Setup to install it.

    ```
    ./Setup -m silent -l yes -d /arcgis/server
    ```

13. If your site includes a PostgreSQL database cluster on SITEHOST or on EGDBHOST, upgrade your geodatabases. See Upgrade geodatabases in PostgreSQL in ArcGIS Server for Amazon Web Services for instructions.
14. Test and configure your updated site. Make sure it is working exactly the way you want.

If you were to leave your site in its current state, the second EC2 instance would be lost the next time you stopped your site using Cloud Builder. Therefore, continue with the following steps to preserve your work as a template and launch a site with the desired number of instances.

15. Use Cloud Builder to create a template from your newly updated site.
16. Using the template you created in the previous step, launch a site with the desired number of EC2 instances and auto-scaling rules.
17. If you had data replicated to a geodata service on your original site, you must change your replicas to point to the geodata service on the new site.
    a. Open one of the .mxd files that contains data you replicated to your original site.
    b. Under the **GIS Servers** node in the **Catalog** window, create a connection to the new site.
    c. Click the **Manage Replicas** button on the **Distributed Geodatabase** toolbar.
    The **Replica Manager** opens.
    d. Right-click the replica in the list that you need to point to the new site and click **Properties**.
    e. Click the **Open** button next to the **Relative Replica Connection**.
    f. Navigate to the new site, choose the geodata service, then click **Open**.
    g. Click **OK** to apply changes and close the **Replica Properties** dialog box.
    h. Close the **Replica Manager**.
    i. Restart your geodata service.

    Repeat these substeps for every replica you created that includes geodatabases in your original ArcGIS Server on Amazon Web Services site.
18. Switch your organization's address mapping to point at your updated site's URL. If your apps were referencing the site URL directly, you'll need to modify any URLs in your apps to incorporate your new Elastic Load Balancer (ELB) address. You can get the ELB address by looking at the **Manager URL** in the list of site details in Cloud Builder.
    Once you verify that the updated site is working, you can delete your original site and optionally the templates you created in step 2.

Related topics

Upgrade geodatabases in PostgreSQL in ArcGIS Server for Amazon Web Services
Apply an ArcGIS update to a Windows multiple-machine site
Apply an ArcGIS update to a single-machine site
Cloud Builder updates

Build a site manually using the AWS Management Console

Use the AWS Management Console with ArcGIS Server

The Amazon Web Services (AWS) Management Console is a web application that you can use to administer the GIS servers and other resources that you have created on AWS. You should typically use ArcGIS Server Cloud Builder on Amazon Web Services to do basic things like create, stop, and start your site. However, you can use the AWS Management Console to accomplish advanced administrative tasks such as adjusting security group rules or attaching more storage to your site.

You can also build ArcGIS Server sites with the AWS Management Console, although it's easier to create a site with ArcGIS Server Cloud Builder on Amazon Web Services. Any site you create with the AWS Management Console consists of single-machine sites connected under an Amazon Elastic Load Balancer (ELB). These sites are stand alone; they don't work together as part of a single site and, therefore, are not suited for tasks such as asynchronous geoprocessing or distributed map caching jobs. To take advantage of the architecture in which the GIS servers communicate with each other, you must use Cloud Builder.

Tip: When using the AWS Management Console with Internet Explorer, Amazon strongly recommends using at least Internet Explorer 9, as Internet Explorer 8 has a slower JavaScript engine that can result in timeouts.

Build ArcGIS Server site using AWS Management Console

Creating an ArcGIS server site using the AWS Management Console is a multistep process that requires some architectural planning and understanding of Amazon Web Services. One example of a basic workflow is listed below. The approach you take may vary based on your deployment needs.

1. Prepare to deploy ArcGIS Server on Amazon Web Services.
   - Create an Amazon account and make sure it includes Elastic Compute Cloud (EC2) access.
   - Get access to the Esri-created Amazon Machine Images (AMIs) through Esri Customer Service (for US customers) or your local distributor (for customers outside the US).
   - Create the typical Amazon security groups for ArcGIS, configuring at least one security group that allows remote access.
   - Configure an Amazon Elastic IP address for the production AWS instance so it can be recovered or updated with minimal downtime.
2. Launch an AWS instance running ArcGIS Server and connect to it.
   - Use one of the ArcGIS Server AMIs to launch an AWS instance.
   - Connect to your new instance using Windows Remote Desktop, SSH and a terminal emulator from Windows, or an SSH connection from Linux.
3. Set up a place to store your data.
   - Choose where to store your data on the cloud.
   - Optionally, configure additional EBS volumes for local data storage.
   - Optionally, create enterprise geodatabases in PostgreSQL, SQL Server, or SQL Server on Amazon Relational Database Service to store your data.
4. Move your data to AWS.
   - Choose a data transfer method and move your data.
   - Give the ArcGIS Server account permissions to access your data.
5. Create GIS services and applications.
   - Use ArcGIS to create services for mapping, geocoding, editing, and other functionality.
   - Migrate your applications to the AWS instance and repoint them to the services now running on your AWS instance.
6. Customize security on your AWS instance.
   - Change the administrator password for your AWS instance (Windows only).
   - Change the passwords for the database users if you're using a geodatabase in PostgreSQL.
   - Optionally, configure ArcGIS server security on your server.
   - Optionally, make other security adjustments to the AWS instance such as firewall adjustments, SSL configuration, and so on.
7. Prepare your production environment.

- Create a custom AMI containing the services and apps on your site.
- Iterate with larger instance types, if necessary, to arrive at the instance type with the appropriate amount of power for your deployment.
- Alter your AWS security group rules to disable or restrict remote access to the production AWS instance. This increases security on the production instance.
- Optionally, use your custom AMI to launch or terminate instances in response to demand. This can be done programmatically using the Amazon Auto Scaling API, or you can do it manually. Multiple machines can be connected by an Amazon Elastic Load Balancer (ELB).

This help system describes many of the above steps. Another helpful resource is the Amazon Web Services documentation, which explains how to use the AWS Management Console and how to program with Amazon Web Services.

Launch an Amazon Web Services instance running ArcGIS for Server

If you build your ArcGIS Server site manually using the Amazon Web Services (AWS) Management Console, there are several things you must be aware of when creating an AWS instance running ArcGIS Server.

Tip: If you use ArcGIS Server Cloud Builder on Amazon Web Services to build your site, the process of launching an AWS instance running ArcGIS Server is simplified, and your site can take full advantage of the peer-to-peer aspects of the ArcGIS Server architecture.

For specific steps and explanations of Amazon Web Services functionality and how to use the AWS Management Console, read the Amazon Web Services documentatioin. The following is information specific to launching an AWS instance running ArcGIS Server.

Before you launch an instance

You must have an Amazon account enabled for AWS access, and you should create an Elastic IP address to use with your ArcGIS Server for Amazon Web Services instance.

1. Open a web browser and sign in to the AWS Management Console with your Amazon user name and password.
   Your Amazon account must be enabled for AWS access, which is a one-time step that can be performed on the AWS website. Contact your system administrator or Amazon if you are unsure about the account to use.
2. Go to the EC2 section of the AWS console and choose the region in which you want to work.
   Regions represent the AWS data centers available throughout the world. You can use regions to get your deployment closer to customers or to meet legal requirements. Pricing varies among the regions.
3. Allocate a new Elastic IP address to use with your instance. Be sure to record the IP that is created, as you will use that later.

Launch an instance from an Esri ArcGIS Server AMI

Launch an instance from the EC2 section of the AWS Management Console. To find the Amazon Machine Image (AMI), search for `ArcGIS` or `Esri`.

There is an AMI available for Linux and two AMIs available for Windows (one with SQL Server Express and another with SQL Server Standard). If you don't see the AMIs, contact Esri Customer Service (U.S. customers) or local distributor (outside the United States) and provide them with your Amazon account number so they can share the AMIs with you.

Instance type

You must choose an instance type when you launch an AWS instance. Be sure to choose an instance type that meets at least the minimum system requirements for ArcGIS Server.

Be aware that different instance types will incur different charges with Amazon.

Storage options

When you launch an instance from the AWS Management Console, you are given the opportunity to change the storage options. However, don't change the storage settings when you launch the instance. There are settings on the AMIs that configure device mapping, and altering the storage options could cause these mappings to fail.

If you want to change the size of the attached EBS volume, you can do so after launching your instance. See Replace the default EBS volume on Windows and Replace the default EBS volume on Linux. Although these topics give steps for Cloud Builder, their workflows can be adapted for use with the AWS Management Console.

Security groups

Create a security group when you launch your instance to define what incoming traffic can access your instance.

The following are basic security group rules that allow you to log in to the ArcGIS Server for Amazon Web Services instance and test that your services are running. See Common security group configurations for lists of security group rules that you can apply in different scenarios.

Protocol	Port Range (Code)	Source
RDP (Windows instance only)	3389	Specify an IP address or IP address range of the machines allowed to connect to your instance through remote desktop.
SSH (Linux instance only)	22	Specify an IP address or IP address range of the machines allowed to connect to your instance through an SSH client.
Custom TCP rule	6080	If you are not using an Elastic Load Balancer, type `0.0.0.0/0`. If you are using an Elastic Load Balancer, type the Elastic Load Balancer's security group name. This is a value such as `amazon-elb/amazon-elb-sg`.

Create a key pair

You must create a key pair to allow you to log in to your AWS instance. The key pair is needed to retrieve the administrator password of an AWS Windows instance. You must have the key pair to make an SSL connection to an AWS Linux instance.

Download the key pair file, which has a .pem extension. Save the file in a safe place.

After you launch an instance

It will take a few minutes to launch the instance. While the instance is launching, its status is shown as pending in the AWS Management Console. Once the instance has finished launching, its status is shown as started, and you have a running AWS instance for which you will incur charges. You may need to click **Refresh** to determine if your instance has finished launching.

For several minutes after you launch or start an AWS instance, you may notice that you cannot log in even though the instance appears started in the AWS Management Console. It takes a few minutes for the instance and ArcGIS to be completely configured.

To determine whether your instance is ready to accept a connection, right-click the instance in the AWS Management Console and click **Get System Log**. When you first launch or start the instance, you will see a blank log screen when you do this. On a healthy Windows AWS instance, you should see a message that Windows is ready to use if you check the log a few minutes later. To allow time for ArcGIS to configure itself, you should not attempt to connect to your AWS instance until at least five minutes after the system log reports the message Windows is ready to use, although this time can vary based on the health of Amazon Web Services.

See Administer your Amazon EC2 instance with Windows Remote Desktop Connection, Administer your Amazon EC2 instance with remote access to Ubuntu Linux, or Administer your Amazon EC2 instance with remote access from Windows to Ubuntu Linux for additional instructions about logging in to your instance and working with it.

If you plan to store data in enterprise geodatabases on the instance, you must create the geodatabases. See Create enterprise geodatabases in SQL Server on AWS or Create geodatabases in PostgreSQL on AWS for instructions.

Tip: Once your instances are no longer needed, stop or terminate them to avoid unnecessary charges. When you terminate an instance, all data and software on it is lost and cannot be recovered. However, you can stop and start an instance without losing data. You can right-click the status of an instance to see a menu that allows you to stop, start, or terminate the instance.

Elastic IPs

Amazon Elastic IPs and ArcGIS Server

An advantage of using Amazon EC2 is the ability to start, stop, create, and terminate instances at any time. However, this flexibility creates a potential challenge with IP addresses. Restarting a stopped instance (or re-creating an instance after another instance is terminated) results in a new IP address. How do you successfully reference a machine when the IP address is constantly changing?

In response to this problem, Amazon offers the ability to allocate an Elastic IP address. An Elastic IP provides you a single IP address that you can associate with different EC2 instances over time. If your EC2 instance has an Elastic IP and that instance is ever stopped or terminated, you can immediately associate a new EC2 instance with the Elastic IP. Your existing applications will not break because the applications see the IP address they were expecting, even though the back-end EC2 instance has changed.

If you're building your own site manually, you should associate an Elastic IP with your enterprise geodatabase instance if you have the geodatabase on its own dedicated instance. Optionally, you can also associate an Elastic IP with your ArcGIS Server instance if you are not already using an Elastic Load Balancer.

If you use ArcGIS Server Cloud Builder on Amazon Web Services to build your site, no Elastic IPs are created or needed.

If you ever need to stop an instance, you should reassociate it with its Elastic IP after you start the instance again. You can even associate the Elastic IP with a backup instance while the other instance is down. If you don't have an Elastic IP, users' connections to your machines will permanently break if you ever have to stop the instance.

Note: An Elastic IP is not the same as an Elastic Load Balancer. An Elastic Load Balancer helps you scale out your site by associating many EC2 instances at the same time under one web address. An Elastic IP, on the other hand, can only be associated with one EC2 instance at a time.

Allocate an Amazon Elastic IP and associate it with your instance

To allocate an Elastic IP and associate it with an Amazon Web Services (AWS) instance, do the following:

1. Open the AWS Management Console, click the **EC2** link, and display the page associated with your region.
2. Click the **Elastic IPs** link.
3. Click **Allocate New Address** and choose **EC2** or **VPC** from the drop-down list, depending whether you're going to associate this IP with an instance in Amazon EC2 or Amazon Virtual Private Cloud, respectively. Click **Yes, allocate** to confirm your choice.
4. Right-click the newly created Elastic IP and choose **Associate Address**.
5. Choose your desired EC2 instance from the drop-down list of running instances and click **Associate**.

Related topics

Amazon Elastic IPs and ArcGIS Server

Elastic Load Balancers

Amazon Elastic Load Balancers and ArcGIS Server

An Amazon Elastic Load Balancer (ELB) unites multiple Elastic Compute Cloud (EC2) instances under a common web address. The ELB plays the role of the web gateway in the ArcGIS Server architecture and prevents any one of the GIS servers from being overwhelmed with incoming requests. You can then harness the power of all the GIS server instances under the load balancer to respond to requests.

When you build your site with ArcGIS Server Cloud Builder on Amazon Web Services, an ELB is created for you, and all the instances in your site are placed under it. Also, any instances created by your auto scaling rules are placed under the ELB. You will see the ELB address when you examine the URL to your site.

If you build your site using the AWS Management Console, you need to create the ELB yourself and add your instances to it. The suggested workflow is as follows:

1. Create an ELB.
2. Launch one EC2 instance running ArcGIS for Server.
3. Load your data and applications on the instance.
4. Launch more GIS server instances, either manually or through auto scaling.
5. Add your instances to the ELB.

Create an Elastic Load Balancer

You can use the AWS Management Console to create an ELB and add EC2 instances to it. Steps for doing this are as follows:

1. Log in to the AWS Management Console and click **EC2**.
2. Click **Load Balancers**.
3. Click **Create Load Balancer**.
4. Type a load balancer name.
5. Choose whether to create the load balancer in EC2 or a VPC.
6. Click **Add** to add listener configuration rules. Choose the ports and protocols to be used by your load balancer. You need to define the port on which HTTP traffic will enter the load balancer (**Load Balancer Port**) and the port to which HTTP traffic will be forwarded (**Instance Port**).
   In many ArcGIS Server scenarios, you can set the **Load Balancer Port** as 80 and the **Instance Port** as 6080, which is ArcGIS Server's port. To save this rule in the AWS Management Console, you'll need to remove the default rule of 80 forwarding to 80. If you have configured the ArcGIS Web Adaptor on your instance, you would use the web adaptor port for the **Instance Port** instead of 6080.

   When you've finished setting these values, click **Continue**.
7. Set the configuration options for the load balancer health check.
   Your load balancer periodically checks its attached instances to ensure that they are still responsive. The checks are performed by pinging a page on the instance, by default /index.html, and determining whether a valid response was returned. You must either configure a page /index.html on your instances or change the value for **Ping Path** to a page located at an identical location on each instance. A good setting for **Ping Path** is a page related to your GIS server implementation, such as the REST URL of one of your map services.

   When you've finished configuring the health check, click **Continue**.
8. Click **Continue** to skip the wizard panel that adds instances to the load balancer.
   You can add instances to your load balancer at a later time. You don't have to add any instances right now to create the load balancer.
9. Add tags to help you identify your load balancer. Click **Continue** when you finish.
10. Review the information about your load balancer and, if necessary, go back and make any corrections. When you're satisfied with the configuration options, click **Create** to create the load balancer.
11. Click **Close** to close the wizard.
12. Find your new load balancer in the **Load Balancers** section of the AWS Management Console. Note the DNS name of the load balancer. To take advantage of the load balancer, you should now include this in any URLs that you use to reference your GIS server site.

Add AWS instances to an Elastic Load Balancer

Once you've created an Amazon Elastic Load Balancer (ELB), follow these steps to add Amazon Web Services (AWS) instances to it.

1. Log in to the AWS Management Console and click **EC2**.
2. Click **Load Balancers**.
   The **Load Balancers** section of the console is divided into upper and lower panes. When you choose a load balancer from the upper pane, details about the load balancer appear in the lower pane.
3. In the upper pane, click the box next to your load balancer.
4. In the lower pane, click the **Instances** tab.
5. Click **Edit Instances**.
6. Check the boxes next to the instances you want to add to your load balancer and click **Save**.

Tip: It's recommended that you add instances from a variety of availability zones. This way you still have a running instance if a zone becomes weakened or disabled for any reason. Amazon recommends maintaining the same number of instances in each availability zone because the ELB distributes traffic equally across availability zones.

Create your own AMI

An Amazon Machine Image (AMI) defines the programs and settings that will be applied when you launch an EC2 instance. Once you have finished configuring the data, services, and applications on your ArcGIS Server instance, you can save your work as a custom AMI stored in Amazon EC2. You then scale out your site by using this custom AMI to launch additional instances.

Note: When you build a site using ArcGIS Server Cloud Builder on Amazon Web Services, you don't have to know how to save a custom AMI. You create and use site templates, which manage the custom AMIs for you behind the scenes. See Creating a site template to learn how to get started.
If you have decided to build your ArcGIS Server site using the AWS Management Console, you must create the custom AMI and launch the instances manually using the instructions in this topic.

Once you have launched new instances using the custom AMI, you place them beneath an elastic load balancer (ELB), thereby creating a siloed architecture. However, this architecture does not provide any way for the GIS servers to communicate with each other. If you want the full benefits of the newer ArcGIS Server architecture, you should build your site using ArcGIS Server Cloud Builder on Amazon Web Services.

Creating a custom AMI copies any Elastic Block Store (EBS) volumes that you may have attached. Be aware that this will affect your costs when the custom AMI is deployed. For example, you may have a 100 GB attached EBS volume on your current EC2 instance. If you create a custom AMI from your instance and deploy it five times, you will be charged for five new EC2 instances **and** five 100 GB EBS volumes.

Use the following procedure to create your own AMI using the AWS Management Console:

1. Configure an EC2 instance and its attached EBS volumes in the exact way you want them created in the custom AMI.

   Caution: Creating a custom AMI makes an exact copy of the current state of your EC2 instance. Before you continue, remove all personal information from your EC2 instance, including stored information such as web browser cookies and temporary files.

2. Log out of your instance, but do not stop or terminate it.
3. Log in to the AWS Management Console, display the EC2 page for your region, then click **Instances**.
4. Choose the instance from which you want to create a custom AMI.
5. Click **Actions** and click **Create Image**.
6. Type a name for **Image Name** that is easily identifiable to you and, optionally, input text for **Image Description**.
7. Click **Create Image**.
8. Read the message box that appears. To view the AMI status, go to the **AMIs** page. Here you can see your AMI being created. It can take a while to create the AMI. Plan for at least 20 minutes, or longer if you've installed a lot of additional applications or data.

You launch an instance from a custom AMI the same way you launch an instance from the ArcGIS for Server AMI; however, in the first page of the **Request Instances Wizard**, click the **My AMIs** tab and choose your custom AMI. If there are many AMIs in the list, you should be able to discern your AMI using the **Name** field, which includes the image name you typed when you created the AMI.

Caution: AMIs that you create with ArcGIS software are for individual use and are allowed only for the purpose of configuring your own ArcGIS for Server deployment on Amazon. They are not to be redistributed or shared with any other parties.

Add Amazon EC2 instances in response to demand

You can use Amazon EC2 to adjust your ArcGIS server site in response to user demand. You can add multiple EC2 instances to your deployment and connect them under an Elastic Load Balancer (ELB). As demand increases, you can add more instances either manually or programmatically.

Tip: When you build your site with ArcGIS Server Cloud Builder on Amazon Web Services, you can choose settings of the minimum and maximum instances to include in your site, as well as the CPU usage levels that will trigger automatic scaling of your site. This is an easy way to configure auto scaling, since the ELB is configured for you and there is no need for you to write auto scaling commands.
If you do not use Cloud Builder, you must create the instances yourself and add them to the ELB manually using the AWS Management Console or the Amazon Auto Scaling API.

The Elastic Load Balancer and ArcGIS server scaling

If you need more power than one EC2 instance can provide, you can scale out your deployment by creating new EC2 instances as additional GIS servers and placing them under an ELB. All requests to your server go through the ELB, which then evenly distributes the requests to the available EC2 instances.

The AWS Management Console contains a **Load Balancers** link, where you can view your existing ELBs and add new ones using a wizard. You can add and remove instances from an ELB at any time by viewing the load balancer properties and updating a series of check boxes next to a list of instance names.

The ArcGIS Server instances you launch and place beneath the ELB using the AWS Management Console have no knowledge of each other and cannot be used for actions such as asynchronous geoprocessing or distributed map caching. To take full advantage of the architecture in which the GIS servers communicate with each other, build your site using ArcGIS Server Cloud Builder on Amazon Web Services.

Monitor your deployment

Suppose you maintain a public-facing web map application that displays natural disaster information from a state government. This site normally receives 500 visits per hour; however, when major weather events occur, such as tornadoes or flooding, the site receives up to 10,000 visits per hour. You don't want to maintain enough hardware (or even EC2 instances) to always be able to support 10,000 visits per hour, but your site needs to be able to handle this load on random occasions without slowing down.

The first step in scaling out your deployment is to monitor the load on your server. This is not always necessary if you have a pretty good guess of when the traffic is going to occur; however, with events such as tornadoes, you may not have much of a warning. Amazon CloudWatch is a service that you can use to understand the amount of work being done by your EC2 instances. CloudWatch gives you charts of instance usage, and it also provides metrics through optional APIs.

CloudWatch is an option that you must explicitly enable on EC2 instances that you manually create. CloudWatch is already enabled on any instances launched by ArcGIS Server Cloud Builder on Amazon Web Services.

Amazon charges a fee for certain features of CloudWatch. If this is a concern, you can devise your own ways to track certain metrics on your instances, such as CPU and memory usage.

Add more instances

If you detect that your deployment is overworked, you can create additional EC2 instances to add to your deployment. Conversely, if your deployment is underutilized, you can remove instances to cut costs. You can adjust your number of instances either manually or automatically.

If you want to add and remove EC2 instances automatically, you can use ArcGIS Server Cloud Builder on Amazon Web Services to set up rules for adding or removing instances based on levels of CPU usage. Or, if you're building your site manually, you can set up your own rules using the Amazon Auto Scaling API.

Any machines created by Amazon's automatic scaling have CloudWatch enabled, and the corresponding fees will apply.

Delete EBS volumes after terminating instances

When you terminate an instance, the Amazon Elastic Block Store (EBS) volumes attached to that instance are not automatically deleted. If you are not going to reuse the volumes, you should delete them to avoid incurring unnecessary charges. One way to accomplish this is to use the EC2 API to check for EBS volumes with a status of Available and delete any available volumes found. You can perform this check at regular intervals.

Increase your Amazon instance limit

If you plan to run many EC2 instances on your deployment, check your Amazon account to understand the number of instances you are allowed. Most accounts are limited to 20 on-demand or reserved instances and 100 spot instances per region. Amazon provides a web form that you can use to request a different limit for your account.

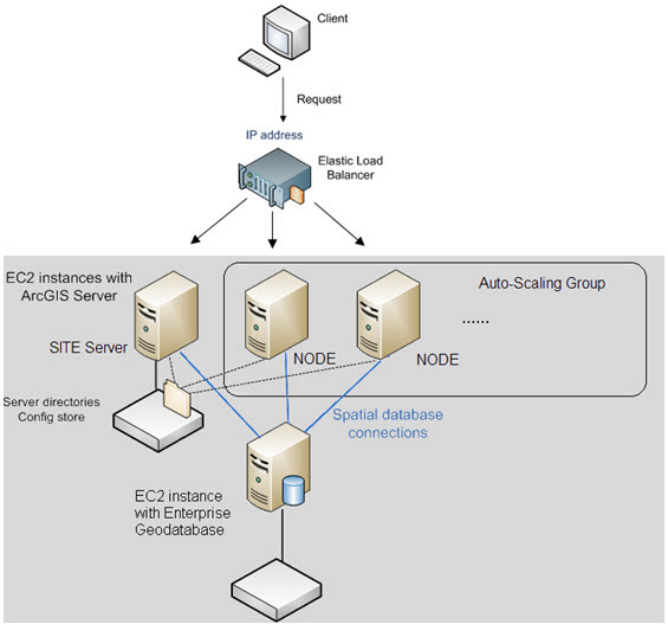
AWS CloudFormation and high availability with ArcGIS Server

AWS CloudFormation and high availability with ArcGIS Server

You can use some features of Amazon Web Services (AWS) CloudFormation to achieve certain high availability benefits when working with ArcGIS Server on AWS.

Some background is necessary to understand the need for this approach. The typical way that people get started using ArcGIS Server on AWS is to use ArcGIS Server Cloud Builder. This is a simple wizard that helps you set up an ArcGIS Server site in the Amazon Elastic Compute Cloud (EC2). The output of the wizard is a site with potentially many machines that can distribute jobs among themselves and take full advantage of the peer-to-peer architecture found in ArcGIS Server. Cloud Builder also offers a framework for making site backups, in addition to templates that can be used to launch additional sites following an identical pattern.

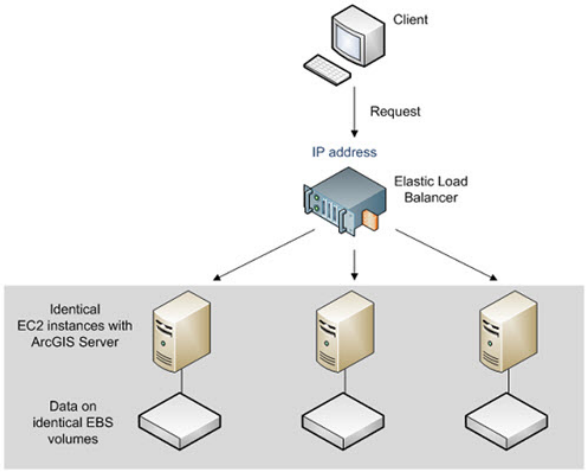
The architecture created by Cloud Builder is pictured in the following diagram.



For all its benefits, the Cloud Builder-created site has a few drawbacks when it comes to high availability. If the GIS server instance running the configuration store is lost, the site will be unusable. Also, it can take 30 minutes or more to restore a backup, which could be an unacceptable amount of downtime for some organizations.

An alternative approach that avoids some of these issues is a siloed architecture, pictured below, wherein many identical but independent ArcGIS Server sites are connected under an elastic load balancer (ELB). This architecture was used exclusively with ArcGIS Server on AWS prior to version 10.1 and can still be used if you are willing to sacrifice ease of administration and the ability to distribute large jobs (such as cache tile building) across servers.

One of the main benefits of the siloed architecture is that it allows you to continue running ArcGIS Server even if you lose any GIS server instance.



The siloed architecture cannot be built with Cloud Builder: you must create it manually using the AWS Management Console or APIs. However, some features of Amazon CloudFormation can help you launch and maintain the siloed architecture much more easily than could be accomplished in the past. CloudFormation ensures that you always maintain a minimum number of machines in your site, allowing for seamless recovery when one of your machines becomes unavailable.

The remainder of this topic explains how to set up a siloed architecture using CloudFormation.

Use CloudFormation to build the siloed architecture

Amazon CloudFormation is a service that helps you define architectures for the Amazon Web Services you use. It is an example of infrastructure as code; meaning you write code that can deploy a particular hardware infrastructure in a cloud environment. In the case of CloudFormation, you use a JavaScript object notation (JSON) template to define a stack of resources that work together in a predetermined way. In this case, the template defines your siloed ArcGIS Server sites and the Elastic Load Balancer that holds them together.

Copy and examine the templates

1. To get started, examine one of these example templates:
   - Example CloudFormation template for Amazon EC2
   - Example CloudFormation template for Amazon Virtual Private Cloud (VPC)
2. Copy the template code to a text file and save it.
3. Examine the JSON properties that are set and modify them as needed.
   There are a number of properties in the Parameters object that you can set when launching the stack, such as the number and types of instances participating in the site and the Amazon Machine Image (AMI) used. Other resources, such as the elastic load balancer (ELB) settings, are burned into the template and can only be changed by the template author.

   Note the auto scaling and CloudWatch alarm settings that define triggers for adding and removing machines from your site based on factors such as CPU usage. You can also define how many machines will be updated at once if you ever make changes to your AMI, allowing you to always maintain an available instance.
4. In addition to the items found in this template, you can introduce an LBCookieStickinessPolicy on the ELB, which determines whether sticky sessions are enabled. These ensure that a user's requests get directed to the same GIS server machine for the duration of his or her session. This can be helpful for working with long-running features like asynchronous geoprocessing jobs.
5. You can also define the SSL certificate to be applied onto the stack. To do this, upload your certificate to AWS Identity and Access Management (IAM) and insert code into the template to specify the SSLCertificateId setting on the ELB.

Create the stack

Once you've selected a template, follow the steps below to launch a multiple machine ArcGIS Server site using the siloed architecture.

1. Use the AWS Management Console to launch an instance using one of the ArcGIS Server AMIs provided by Esri.
2. License ArcGIS Server and create a site on the instance.
3. Configure the instance with your services, data, and any third-party applications you want to install.

> **Caution:** If you have any map services that will not use a cache, set the `supportedImageReturnType` property of the map service to `MIME` instead of `URL` (so the user is not routed back into the stack, and potentially the wrong instance, when trying to retrieve map images). You will need to edit the service properties in the ArcGIS Server Administrator Directory in order to adjust this setting; it is not available in Manager.

4. Save an AMI from your instance. All instances created from this AMI will be the same. Each will be its own ArcGIS Server site with its own configuration store, server directories, and data.
5. Once the AMI is created, take note of the AMI ID. This is needed to create the stack.
6. In the AWS Management Console, click **Services** > **CloudFormation** to open the **CloudFormation Stacks** page.
7. Click **Create Stack**.
8. Provide a **Stack Name**.
9. Choose **Upload a Template File**, browse to one of the template files you saved and modified above, and click **Continue**.
10. Set the following properties for your stack:
    a. Change the **AMI** property to be the ID of your AMI that you created in step 3.
    b. Change the **MaxSize** to the maximum number of instances you want to have in your stack at one time.
    c. If you are using the VPC template, set the options for **VPCId**, **Subnets**, and **AZs** (Availability Zones) to correspond with your desired VPC environment settings.
    d. Adjust the **InstanceType** to reflect the size of instance you want.
    e. Modify the **KeyName** to be the name of one of your Amazon key pairs. Do not include the .pem extension.
    f. Change the **MinSize** to the minimum number of instances you want in your stack at one time. If you keep this value at 2 or more, you protect yourself against loss of your site if a machine becomes unavailable. CloudFormation will immediately create a second instance if only one is found.
11. Click **Continue**.
12. Add tags to easily identify your instance, such as `Key = Name` and `Value = MyCloudFormation`, and click **Continue**.
13. Examine all your parameters and click **Continue** when you're ready to create your stack.

Your stack will be launched. After a short period of time, ArcGIS Server will be accessible through the following URL format: `http://<Elastic Load Balancer address>/arcgis/rest`.

Example CloudFormation template for Amazon EC2

The following CloudFormation template can be used to launch a highly available ArcGIS Server site in Amazon EC2. See AWS Cloud Formation and high availability with ArcGIS Server.

Copy this code, paste it into a text file, and adapt it to your needs. Then browse to the text file when you create an ArcGIS Server stack.

Template code

```
{
	"AWSTemplateFormatVersion": "2010-09-09",

	"Description": "CloudFormation template to launch a highly available ArcGIS Server stack in Amazon EC2",

	"Parameters" : {
		"AMI" : {
			"Description" : "Your ArcGIS Server AMI ID.",
			"Type" : "String"
		},
		"InstanceType" : {
			"Description" : "Type of EC2 instance to launch.",
			"Type" : "String",
			"Default" : "m1.medium",
			"AllowedValues" : [ "m1.medium","m1.large","m1.xlarge","m2.xlarge","m2.2xlarge","m2.4xlarge","m3.xlarge","m3.2xlarge","c1.xlarge","cc2.8xla
			"ConstraintDescription" : "must be a valid EC2 instance type."
		},
		"KeyName" : {
			"Description" : "The EC2 Key Pair to allow Remote Desktop access or SSH to the instances.",
			"Type" : "String",
			"Default" : "Your_KeyPair_Name"
		},
		"MinSize" : {
			"Description" : "Minimum number of EC2 instances.",
			"Type" : "Number",
			"Default" : "2"
		},
		"MaxSize" : {
			"Description" : "Maximum number of EC2 instances.",
			"Type" : "Number",
			"Default" : "4"
		}
	},

	"Resources": {

		"ELB" : {
			"Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
			"Properties" : {
				"AvailabilityZones" : { "Fn::GetAZs" : "" },
				"Listeners" : [ {
					"LoadBalancerPort" : "80",
					"InstancePort" : "6080" ,
					"Protocol" : "HTTP"
				}],
				"HealthCheck" : {
					"Target" : "HTTP:6080/arcgis/rest/info/healthcheck",
					"HealthyThreshold" : "3",
					"UnhealthyThreshold" : "5",
					"Interval" : "30",
					"Timeout" : "5"
				}
			}
		},

		"InstanceSecurityGroup": {
			"Type": "AWS::EC2::SecurityGroup",
			"Properties": {
				"GroupDescription": "Security group for highly available ArcGIS Server architecture",
				"SecurityGroupIngress": [{
                                                "IpProtocol": "tcp",
                                                "FromPort": "6080" ,
                                                "ToPort": "6080" ,
                                                "SourceSecurityGroupName": { "Fn::GetAtt" : [ "ELB" , "SourceSecurityGroup.GroupName" ]},
                                                                                "SourceSecurityGroupOwnerId": { "Fn::GetAtt" : [ "ELB" , "SourceSecurityGr
                                        }]
			}
		},

		"LaunchConfig": {
			"Type": "AWS::AutoScaling::LaunchConfiguration",
			"Properties": {
				"ImageId": {"Ref" : "AMI"},
				"InstanceType": {"Ref": "InstanceType"},
				"KeyName": {"Ref": "KeyName"},
				"SecurityGroups": [{"Ref": "InstanceSecurityGroup"}],
				"InstanceMonitoring": true
			}
		},

		"AutoScalingGroup": {
			"UpdatePolicy" : {
				"AutoScalingRollingUpdate" : {
					"MinInstancesInService" : {"Ref" : "MinSize"},
					"MaxBatchSize" : "1",
					"PauseTime" : "PT15M"
				}
			},
			"Type": "AWS::AutoScaling::AutoScalingGroup",
			"Properties": {
				"AvailabilityZones": { "Fn::GetAZs" : ""},
				"Cooldown": "300",
				"MaxSize": {"Ref" : "MaxSize"},
				"MinSize": {"Ref" : "MinSize"},
				"LaunchConfigurationName": {"Ref": "LaunchConfig"},
				"HealthCheckType" : "EC2",
				"HealthCheckGracePeriod" : "3600",
				"LoadBalancerNames": [{"Ref": "ELB"}],
				"Tags": [{"Key": "Name", "Value": {"Ref": "AWS::StackName"}, "PropagateAtLaunch" : true}]
			}
		},

		"ScaleUpPolicy" : {
			"Type" : "AWS::AutoScaling::ScalingPolicy",
			"Properties" : {
				"AdjustmentType" : "ChangeInCapacity",
				"AutoScalingGroupName" : { "Ref" : "AutoScalingGroup" },
				"Cooldown" : "60",
				"ScalingAdjustment" : "1"
			}
		},

		"ScaleDownPolicy" : {
			"Type" : "AWS::AutoScaling::ScalingPolicy",
```

```
                        "Properties" : {
                                "AdjustmentType" : "ChangeInCapacity",
                                "AutoScalingGroupName" : { "Ref" : "AutoScalingGroup" },
                                "Cooldown" : "60",
                                "ScalingAdjustment" : "-1"
                        }
                },

                "CPUAlarmHigh": {
                        "Type": "AWS::CloudWatch::Alarm",
                        "Properties": {
                                "AlarmDescription": "Scale-up if CPU > 80% for 10 minutes",
                                "MetricName": "CPUUtilization",
                                "Namespace": "AWS/EC2",
                                "Statistic": "Average",
                                "Period": "300",
                                "EvaluationPeriods": "2",
                                "Threshold": "80",
                                "AlarmActions": [ { "Ref": "ScaleUpPolicy" } ],
                                "Dimensions": [
                                        {
                                                "Name": "AutoScalingGroupName",
                                                "Value": { "Ref": "AutoScalingGroup" }
                                        }
                                        ],
                                        "ComparisonOperator": "GreaterThanThreshold"
                        }
                },

                "CPUAlarmLow": {
                        "Type": "AWS::CloudWatch::Alarm",
                        "Properties": {
                                "AlarmDescription": "Scale-down if CPU < 20% for 10 minutes",
                                "MetricName": "CPUUtilization",
                                "Namespace": "AWS/EC2",
                                "Statistic": "Average",
                                "Period": "300",
                                "EvaluationPeriods": "2",
                                "Threshold": "20",
                                "AlarmActions": [ { "Ref": "ScaleDownPolicy" } ],
                                "Dimensions": [
                                        {
                                                "Name": "AutoScalingGroupName",
                                                "Value": { "Ref": "AutoScalingGroup" }
                                        }
                                        ],
                                        "ComparisonOperator": "LessThanThreshold"
                        }
                }
        },

        "Outputs": {
                "RestURL": {
                        "Value": {
                                "Fn::Join": ["", ["http://", {"Fn::GetAtt": ["ELB", "DNSName" ]}, "/arcgis/rest"]]
                        },
                        "Description" : "ArcGIS REST Services Directory URL."
                }
        }

}
```

Parameters in this template

The **AMI** parameter sets the AMI ID that will be used when launching instances in this stack. You point this at your own customized AMI running ArcGIS Server.

The **InstanceType** parameter sets the EC2 instance type that will be used when launching instances in the stack. Set this with caution because it could affect both AWS costs and Esri licensing costs. The costs will be multiplied by the number of instances you launch using your auto scaling groups.

The **KeyName** is the name of a key pair that will allow you to retrieve the administrator password for your instances. You'll need to create the key pair file prior to using the template. This option is necessary if you will be connecting to your instances via Remote Desktop or SSH.

The **MinSize** is the minimum number of instances participating in your stack at any time. This number of instances will immediately be launched when you create your stack. If you keep this value at 2 or more, you protect yourself against ArcGIS Server being unavailable if an instance goes down. CloudFormation will immediately create a second instance if only one is found.

The **MaxSize** is the maximum number of instances that are allowed to participate in your stack at any time, no matter your auto scaling triggers or instance CPU usage.

Resources in this template

The **ELB** resource describes the Elastic Load Balancer (ELB) placed in the stack. The ELB is the point of entry for all web requests to ArcGIS Server. It distributes requests to the available ArcGIS Server instances. You can use properties on the ELB to adjust the load balancer port, the instance port (which could deviate from the default 6080 to be 6443 in the case of an SSL-secured site), and the health check settings.

The **InstanceSecurityGroup** resource determines the access rules that will be applied to any instances launched into this stack. Notice that you can set ingress rules determining which ports are open to the instances. In this template, the Elastic Load Balancer group is allowed to access the instances on port 6080.

The **LaunchConfig** resource brings in some of the parameters set by the template user to determine the type of instance to be launched and the AMI to be applied.

The **AutoScalingGroup** resource sets up rules about when instances will be added and removed from the stack in response to triggers such as CPU usage. This resource also contains an update policy that determines how many instances will be updated at once when you make an update to the AMI. The updates are applied on a rolling basis so that your entire stack is not taken offline by the update. Your MaxBatchSize represents the number of instances updated at once, and should ideally be set lower than your MinSize property for your stack, thereby ensuring that an instance always remains available during an update.

The **ScaleUpPolicy** resource defines how an instance will be added to the stack in case of heavy load. This is referenced later in the template by the CPUAlarmHigh resource.

The **ScaleDownPolicy** resource defines how an instance will be removed from the stack in case of a light load. This is referenced later in the template by the CPUAlarmLow resource.

The **CPUAlarmHigh** resource sets specific parameters for an alarm that will cause an instance to be added to the stack. In this template, an instance is added when CPU usage exceeds 80 percent for 10 minutes.

The **CPUAlarmLow** resource sets specific parameters for an alarm that will cause an instance to be removed from the stack. In this template, an instance is removed when CPU usage goes below 20 percent for 10 minutes.

Detailed examples of all the JSON properties you can put in a CloudFormation template are available in the section Working With Templates in the AWS documentation.

Example CloudFormation template for Amazon VPC

The following CloudFormation template can be used to launch a highly available ArcGIS Server stack in Amazon Virtual Private Cloud (VPC). See AWS Cloud Formation and high availability with ArcGIS Server.

Copy this code, paste it into a text file, and adapt it to your needs. Then browse to the text file when you create your ArcGIS Server stack with CloudFormation.

Template code

```
{        "AWSTemplateFormatVersion": "2010-09-09",

        "Description": "CloudFormation template to launch a highly available ArcGIS Server stack in Amazon VPC",

        "Parameters" : {
                "AMI" : {
                        "Description" : "Your ArcGIS Server AMI ID.",
                        "Type" : "String"
                },
                "VpcId" : {
                        "Type" : "String",
                        "Description" : "VpcId of your existing Virtual Private Cloud (VPC).",
                        "Default" : "vpc-4adfc123"
                },
                "Subnets" : {
                        "Type" : "CommaDelimitedList",
                        "Default" : "subnet-8ec5b8e6,subnet-1edcc277",
                        "Description" : "The list of SubnetIds where the stack will be launched"
                },
            "AZs" : {
                "Type" : "CommaDelimitedList",
                "Default" : "us-west-2b,us-west-2c",
                "Description" : "The list of AvailabilityZones for your Virtual Private Cloud (VPC). It needs to be consistent with the AZ of your subnets."
                },
                "InstanceType" : {
                        "Description" : "Type of EC2 instance to launch.",
                        "Type" : "String",
                        "Default" : "m1.medium",
                        "AllowedValues" : [ "m1.medium","m1.large","m1.xlarge","m2.xlarge","m2.2xlarge","m2.4xlarge","m3.xlarge","m3.2xlarge","c1.xlarge","cc2.8xla
                        "ConstraintDescription" : "must be a valid EC2 instance type."
                },
                "KeyName" : {
                        "Description" : "The EC2 Key Pair to allow Remote Desktop access or SSH to the instances.",
                        "Type" : "String",
                        "Default" : "Your_KeyPair_Name"
                },
                "MinSize" : {
                        "Description" : "Minimum number of EC2 instances.",
                        "Type" : "Number",
                        "Default" : "2"
                },
                "MaxSize" : {
                        "Description" : "Maximum number of EC2 instances.",
                        "Type" : "Number",
                        "Default" : "4"
                }
        },

        "Resources": {

                "LoadBalancerSecurityGroup": {
                        "Type": "AWS::EC2::SecurityGroup",
                        "Properties": {
                                "GroupDescription": "Enable HTTP access on port 80 and 443.",
                                "VpcId" : { "Ref" : "VpcId" },
                                "SecurityGroupIngress": [{
                                                                        "IpProtocol": "tcp",
                                                                        "FromPort": "80" ,
                                                                        "ToPort": "80" ,
                                                                        "CidrIp" : "0.0.0.0/0"
                                                                                                },{
                                                                         "IpProtocol": "tcp",
                                                                        "FromPort": "443" ,
                                                                        "ToPort": "443" ,
                                                "CidrIp" : "0.0.0.0/0"
                                                                        }],
                        "SecurityGroupEgress" : [{                                                                                      "IpProtocol" : "tcp",
                                                                                                                                        "FromPort" : "6080",
                                                                                                                                        "ToPort" : "6080",
                                                                                                                                        "CidrIp" : "0.0.0.0/0"
                                                                                                                                        },{
                                                                        "IpProtocol": "tcp",
                                                                        "FromPort": "6443" ,
                                                                        "ToPort": "6443" ,
                                                                        "CidrIp" : "0.0.0.0/0"
                                                                }]
                                                        }
                        },
                "ELB" : {
                        "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
                        "Properties" : {
                                "Subnets" : { "Ref" : "Subnets" },
                                "SecurityGroups" : [ { "Ref" : "LoadBalancerSecurityGroup" } ],
                                "Listeners" : [ {
                                        "LoadBalancerPort" : "80",
                                        "InstancePort" : "6080" ,
                                        "Protocol" : "HTTP"
                                }],
                                "HealthCheck" : {
                                        "Target" : "HTTP:6080/arcgis/rest/info/healthcheck",
                                        "HealthyThreshold" : "3",
                                        "UnhealthyThreshold" : "5",
                                        "Interval" : "30",
                                        "Timeout" : "5"
                                }
                        }
                },

                "InstanceSecurityGroup": {
                        "Type": "AWS::EC2::SecurityGroup",
                        "Properties": {
                                "GroupDescription": "Enable HTTP access on 6080 and 6443 from ELB.",
                                "VpcId" : { "Ref" : "VpcId" },
                                "SecurityGroupIngress": [{
                                                                        "IpProtocol": "tcp",
                                                                        "FromPort": "6080" ,
                                                                        "ToPort": "6080" ,
                                                                                                "SourceSecurityGroupId": { "Ref" : "LoadBalancerSecurityGroup" }
                                                                },{
                                                                        "IpProtocol": "tcp",
                                                                        "FromPort": "6443" ,
                                                                        "ToPort": "6443" ,
                                                                        "SourceSecurityGroupId": { "Ref" : "LoadBalancerSecurityGroup" }
```

```
                                                }]
                        }
                },

                "LaunchConfig": {
                        "Type": "AWS::AutoScaling::LaunchConfiguration",
                        "Properties": {
                                "ImageId": {"Ref" : "AMI"},
                                "InstanceType": {"Ref": "InstanceType"},
                                "KeyName": {"Ref": "KeyName"},
                                "SecurityGroups": [{"Ref": "InstanceSecurityGroup"}],
                                "InstanceMonitoring": true
                        }
                },

                "AutoScalingGroup": {
                        "UpdatePolicy" : {
                                "AutoScalingRollingUpdate" : {
                                        "MinInstancesInService" : {"Ref" : "MinSize"},
                                        "MaxBatchSize" : "3",
                                        "PauseTime" : "PT15M"
                                }
                        },
                        "Type": "AWS::AutoScaling::AutoScalingGroup",
                        "Properties": {
                                "AvailabilityZones" : { "Ref" : "AZs" },
                                "VPCZoneIdentifier" : { "Ref" : "Subnets" },
                                "Cooldown": "300",
                                "MaxSize": {"Ref" : "MaxSize"},
                                "MinSize": {"Ref" : "MinSize"},
                                "LaunchConfigurationName": {"Ref": "LaunchConfig"},
                "HealthCheckType" : "EC2",
                "HealthCheckGracePeriod" : "3600",
                                "LoadBalancerNames": [{"Ref": "ELB"}],
                                "Tags": [{"Key": "Name", "Value": {"Ref": "AWS::StackName"}, "PropagateAtLaunch" : true}]
                        }
                },

                "ScaleUpPolicy" : {
                        "Type" : "AWS::AutoScaling::ScalingPolicy",
                        "Properties" : {
                                "AdjustmentType" : "ChangeInCapacity",
                                "AutoScalingGroupName" : { "Ref" : "AutoScalingGroup" },
                                "Cooldown" : "60",
                                "ScalingAdjustment" : "1"
                        }
                },

                "ScaleDownPolicy" : {
                        "Type" : "AWS::AutoScaling::ScalingPolicy",
                        "Properties" : {
                                "AdjustmentType" : "ChangeInCapacity",
                                "AutoScalingGroupName" : { "Ref" : "AutoScalingGroup" },
                                "Cooldown" : "60",
                                "ScalingAdjustment" : "-1"
                        }
                },

                "CPUAlarmHigh": {
                        "Type": "AWS::CloudWatch::Alarm",
                        "Properties": {
                                "AlarmDescription": "Scale-up if CPU > 80% for 10 minutes",
                                "MetricName": "CPUUtilization",
                                "Namespace": "AWS/EC2",
                                "Statistic": "Average",
                                "Period": "300",
                                "EvaluationPeriods": "2",
                                "Threshold": "80",
                                "AlarmActions": [ { "Ref": "ScaleUpPolicy" } ],
                                "Dimensions": [
                                        {
                                                "Name": "AutoScalingGroupName",
                                                "Value": { "Ref": "AutoScalingGroup" }
                                        }
                                        ],
                                        "ComparisonOperator": "GreaterThanThreshold"
                        }
                },

                "CPUAlarmLow": {
                        "Type": "AWS::CloudWatch::Alarm",
                        "Properties": {
                                "AlarmDescription": "Scale-down if CPU < 20% for 10 minutes",
                                "MetricName": "CPUUtilization",
                                "Namespace": "AWS/EC2",
                                "Statistic": "Average",
                                "Period": "300",
                                "EvaluationPeriods": "2",
                                "Threshold": "20",
                                "AlarmActions": [ { "Ref": "ScaleDownPolicy" } ],
                                "Dimensions": [
                                        {
                                                "Name": "AutoScalingGroupName",
                                                "Value": { "Ref": "AutoScalingGroup" }
                                        }
                                        ],
                                        "ComparisonOperator": "LessThanThreshold"
                        }
                }
        },

        "Outputs": {
                "RestURL": {
                        "Value": {
                                "Fn::Join": ["", ["http://", {"Fn::GetAtt": ["ELB", "DNSName" ]}, "/arcgis/rest"]]
                        },
                        "Description" : "ArcGIS Server REST Services Directory URL"
                }
        }

}
```

Parameters in this template

The **AMI** parameter sets the AMI ID that will be used when launching instances in this stack. You point this at your own customized AMI running ArcGIS Server.

The **VPCId** parameter is the physical ID of the virtual private cloud that will be used to launch the stack.

The **Subnets** parameter specifies a comma-delimited list of your VPC subnets where you want to launch the stack. For high availability purposes, you can specify subnets in more than one zone. The instances in your stack will be spread among the zones.

The **AZs** parameter is a comma-delimited list of the names of the availability zones housing the subnets that you listed.

The **InstanceType** parameter sets the EC2 instance type that will be used when launching instances in the stack. Set this with caution because it could affect both AWS costs and Esri licensing costs. The costs will be multiplied by the number of instances you launch using your auto scaling groups.

The **KeyName** is the name of a key pair that will allow you to retrieve the administrator password for your instances. You'll need to create the key pair file prior to using the template. This option is necessary if you will be connecting to your instances via Remote Desktop or SSH.

The **MinSize** is the minimum number of instances participating in your stack at any time. This number of instances will immediately be launched when you create your stack. If you keep this value at 2 or more, you protect yourself against ArcGIS Server being unavailable if an instance goes down. CloudFormation will immediately create a second instance if only one is found.

The **MaxSize** is the maximum number of instances that are allowed to participate in your stack at any time, no matter your auto scaling triggers or instance CPU usage.

Resources in this template

The **LoadBalancerSecurityGroup** resource determines the access rules for the Elastic Load Balancer, including the ports through which traffic can be received and forwarded.

The **ELB** resource describes the Elastic Load Balancer (ELB) placed in the stack. The ELB is the point of entry for all web requests to ArcGIS Server. It distributes requests to the available ArcGIS Server instances. You can use properties on the ELB to adjust the load balancer port, the instance port (which could deviate from the default 6080 to be 6443 in the case of an SSL-secured site), and the health check settings.

The **InstanceSecurityGroup** resource determines the access rules that will be applied to any instances launched into this stack. Notice that you can set ingress rules determining which ports are open to the instances. In this template, the Elastic Load Balancer group is allowed to access the instances on port 6080.

The **LaunchConfig** resource brings in some of the parameters set by the template user to determine the type of instance to be launched and the AMI to be applied.

The **AutoScalingGroup** resource sets up rules about when instances will be added and removed from the stack in response to triggers such as CPU usage. This resource also contains an update policy that determines how many instances will be updated at once when you make an update to the AMI. The updates are applied on a rolling basis so that your entire stack is not taken offline by the update. Your MaxBatchSize represents the number of instances updated at once, and should ideally be set lower than your MinSize property for your stack, thereby ensuring that an instance always remains available during an update.

The **ScaleUpPolicy** resource defines how an instance will be added to the stack in case of heavy load. This is referenced later in the template by the CPUAlarmHigh resource.

The **ScaleDownPolicy** resource defines how an instance will be removed from the stack in case of a light load. This is referenced later in the template by the CPUAlarmLow resource.

The **CPUAlarmHigh** resource sets specific parameters for an alarm that will cause an instance to be added to the stack. In this template, an instance is added when CPU usage exceeds 80 percent for 10 minutes.

The **CPUAlarmLow** resource sets specific parameters for an alarm that will cause an instance to be removed from the stack. In this template, an instance is removed when CPU usage goes below 20 percent for 10 minutes.

Detailed examples of all the JSON properties you can put in a CloudFormation template are available in the section Working With Templates in the AWS documentation.

Reference

Amazon Virtual Private Cloud and ArcGIS Server

Amazon Virtual Private Cloud (VPC) allows you to create a subnet of Amazon Elastic Compute Cloud (EC2) instances that can act as your own private network in the cloud. You can work with this subnet independently in the cloud or use it with a VPN connection to expand your organization's internal network.

You need to first create the subnet using the AWS Management Console or one of the AWS APIs. When you create an ArcGIS Server site, Cloud Builder detects the VPC subnets associated with your Amazon account and displays them in a drop-down list on the **Amazon Web Services** panel. You can then select a subnet and launch your site within it.

What is Amazon CloudWatch?

Amazon CloudWatch is an optional service from Amazon EC2 that you can purchase to monitor usage of your EC2 instances.

CloudWatch provides the statistics necessary for Amazon EC2 to perform automatic scaling of a site. Consequently, when you create a site with ArcGIS Server Cloud Builder on Amazon Web Services, CloudWatch is enabled on all instances. Whenever the AWS Auto Scaling API is used to create instances automatically, the instances created have CloudWatch enabled.

CloudWatch produces statistics and graphs describing the loads on your cloud-based services. You can see these graphs in the AWS Management Console when you display the informational details for one or more EC2 instances.

In the AWS Management Console, you can enable CloudWatch when you are completing the wizard to launch a new instance. CloudWatch can also be configured on Amazon Elastic Load Balancers (ELBs), which you would use if you had many EC2 instances working together in one ArcGIS Server site.

Localization and ArcGIS Server on Amazon Web Services

The ArcGIS for Server Amazon Machine Images (AMIs) are built on top of Amazon-provided AMIs that have the operating system language set to English.

You can optionally change the language of the operating system and install additional language packs to localize ArcGIS. In a multiple-machine site, you would do this on the site server machine. In most cases you would not need to log in to the additional GIS server instances, so localizing them would be unnecessary.

Operating system languages provided with the AMI

The ArcGIS for Server AMIs have several operating system language packs preinstalled by Amazon. The preinstalled languages are English, German, Spanish, French, Italian, Chinese (Simplified), Chinese (Traditional), Korean, Japanese.

Install additional operating system languages

If the operating system language you want to use is not included on the AMI, you can download and install the language pack from Microsoft.

Windows Server 2012 language pack

Change the operating system language

Once your language pack is installed, follow these steps to change the language of the operating system:

1. Click **Start** > **All Programs** > **Control Panel**, then double-click **Regional and Language Options**.
2. Click the **Keyboards and Languages** tab.
3. Click the **Choose a display language** drop-down arrow and choose the language you want to use.
4. Configure any other needed options using the other tabs on the **Region and Language** dialog box.
   Be aware that if you choose **Change System Locale**, you will be required to reboot the instance for the changes to take effect. Applications that do not support Unicode may no longer work as expected.

Installing ArcGIS language packs

ArcGIS language packs are available for ArcGIS for Server and ArcGIS for Desktop within a few months of the initial product release in English. Available languages include Arabic, Chinese (Simplified), French, German, Italian, Japanese, Portuguese (Brazilian), Russian, and Spanish.

You can download these language packs onto your EC2 instance by connecting to My Esri.

Changing the language displayed by ArcGIS for Server

ArcGIS Server Manager and the Services Directory use the language set for your web browser. After you install the ArcGIS for Server language pack, set the language for your browser.

Internet Explorer

1. In Internet Explorer, click **Tools** > **Internet Options** > **General** > **Languages**.
2. Add the language you want your browser to display.
3. Move this language to the top of the language list.
4. Click **OK** to close all open dialog boxes.
5. Open ArcGIS Server Manager or the Services Directory.

Firefox

1. In Firefox, click **Firefox** > **Options** > **Options**.
2. Click **Content**.
3. Find the **Languages** section of the dialog box and click **Choose**.
4. Add the language you want the browser to display.
5. Move this language to the top of the language list.

6. Click **OK** to close all open dialog boxes.
7. Open ArcGIS Server Manager or the Services Directory.

Changing the language displayed by ArcGIS for Desktop

After installing a language pack, ArcGIS for Desktop should immediately honor the operating system display language (instructions for configuring the display language are at the beginning of this document). If for any reason you need them, the explicit instructions for setting the display language for ArcGIS for Desktop are below:

1. Open ArcGIS Administrator and click **Start** > **All Programs** > **ArcGIS** > **ArcGIS Administrator**.
2. In the left menu of ArcGIS Administrator, click the root node for ArcGIS containing your machine name.
3. Click the **Advanced** button.
4. Choose a **Display Language** from the drop-down list and click **Save**.
5. Click **OK** to close ArcGIS Administrator.

Limitations of ArcGIS Server on Amazon Web Services

Most ArcGIS functionality is available on Amazon Web Services; however, this topic lists some features and configurations that are not supported.

Server clusters are not supported

ArcGIS Server clusters are not supported on Amazon Web Services. ArcGIS Server sites that you create in Amazon EC2 use the default cluster for all GIS servers. You should not attempt to create additional clusters.

Limitations related to the use of the Elastic Load Balancer (ELB) and the Web ADF

The use of Elastic Load Balancers (ELBs) for scaling in the Amazon cloud constrains a few specific features of ArcGIS Server. ELBs are configured by default so that incoming requests are brokered across any of your instances participating in the GIS server site; that is, there is no guarantee that requests from a particular client (web browser user session, desktop client, or mobile application) will be handled by a particular EC2 instance. This limits the use of stateful web applications.

ArcGIS Server Web Application Developer Framework (ADF) applications rely on user session information that is stored at the web tier (in Internet Information Services). For the Web ADF application to work, incoming requests from a user session (web browser) have to hit the EC2 instance in which the session information is being stored. As described previously, an ELB does not guarantee that.

Advanced users can accommodate this limitation by storing Web ADF session information in SQL Server. Amazon also provides several scripts for configuring sticky sessions in the ELB; however, Esri has not certified these configurations in Amazon EC2.

The Web ADF is deprecated technology, and you are encouraged to transition your web apps to the ArcGIS web APIs for JavaScript.

Amazon-imposed limitations

Amazon enforces some limitations on resources you can create, such as the size of EBS volumes, the number of EBS volumes you can attach to an instance, the number of instances you can run at once, and the number of Elastic IPs you can allocate in one region. Exceptions to the limitations are sometimes available upon request to Amazon.

For the most up-to-date and official information about Amazon limitations, see the AWS Documentation.

Find your Amazon account identifier

Each Amazon account has a name and e-mail address associated with it, as well as a 12-digit account identifier (ID). When you have an Amazon Machine Image (AMI) shared with you, or when you share an AMI with another Amazon account, you may need to provide this number. Below is one way you can find your account number.

1. Log in to the AWS Management Console.
2. At the top of the page, click the link that is your account name.
3. Click **My Account**.
4. Your 12-digit account ID is listed under **Account Settings**.

Frequently asked questions

This topic covers common questions about ArcGIS Server on Amazon Web Services.

Cloud Builder questions

ArcGIS Server site questions

General Amazon Web Services questions

Are my Amazon Access Key and Secret Access Key viewed or stored by Esri when I log in to ArcGIS Server Cloud Builder on Amazon Web Services?

No. When you use ArcGIS Server Cloud Builder on Amazon Web Services, your Amazon Access Key and Secret Access Key are evaluated directly against Amazon's databases and are never sent to Esri servers.

How do I obtain a .prvc file to license my site if I have an Esri Developer Network (EDN) license?

Cloud Builder requires a .prvc file to license ArcGIS Server. EDN users can create a .prvc file in the following way:

1. Log in to My Esri to obtain your authorization number.
2. Using a text editor such as Notepad, create a file from the following template and insert your information, including the authorization number you obtained in the previous step.

```
// User Information
First Name=
Last Name=
Organization=
Department=
Email=
Address 1=
City=
State/Province=
Location=<country or region>
Location Code=<country or region code>
Zip/Postal Code=
Phone Number=
Your Organization=
Your Industry=
Yourself=

// Features and authorization numbers
ArcGIS Server=ECP...
```

3. Save the file with a .prvc extension.
4. In Cloud Builder, browse to the .prvc file when prompted for your **License file**.

When I allow Cloud Builder to create a new key pair file, where does the file get placed on disk?

The key pair is placed in your Windows Documents folder under ArcGISCloudBuilder. For example, the path to your key pair file might look like `C:\Users\username\Documents\ArcGISCloudBuilder\arcgis-TestSite.pem`.

Is the Web Adaptor installed when I create my site with Cloud Builder? Do I need to install the Web Adaptor?

ArcGIS Server Cloud Builder on Amazon Web Services does not install or configure the ArcGIS Web Adaptor because the Elastic Load Balancer (ELB) plays many of the same roles, distributing incoming requests among the GIS servers in your site. However, after you create your site, you can optionally log in to your instance and download the Web Adaptor setup from My Esri. This is necessary if you'll be hosting ArcGIS API for JavaScript applications on a Windows instance. See Deploy a web application on a Windows site for more details.

Cloud Builder displays the following message after I log in: "Failed to get a list of custom configuration templates. Forbidden."

Although this message might have multiple causes, it can appear when your system time is out of sync with the actual current time. For example, if you live in a region where the time moves forward an hour during the summer months, and your computer's clock is not synchronized with this change, Cloud Builder will not be allowed to perform certain actions with Amazon EC2.

Can I use Cloud Builder with a proxy server?

ArcGIS Server Cloud Builder on Amazon Web Services cannot be used to build a site when connecting to the Internet through a proxy server.

I recently created an ArcGIS Server site on Windows and the remote desktop connection fails when I attempt to log in.

First, allow some time for the site to launch. If you used ArcGIS Server Cloud Builder on Amazon Web Services to create the site, wait until all the site information appears in the **My Sites** window and you see the buttons for stopping, updating, deleting, and so forth.

If you launched the site manually using the AWS Management Console, right-click the instance and click **Get System Log**. Once you see a message the Windows is ready to use, wait about 5 additional minutes before you attempt to log in.

Second, make sure you've added a rule to your site's security group allowing remote desktop connections through port 3389. This is described in Open an Amazon EC2 security group for ArcGIS Server

After I launch an EC2 instance, how can I be sure that ArcGIS Server was successfully configured?

You can check the health of ArcGIS Server by viewing and logging in to Manager. Expand your site details in Cloud Builder to see the URL for Manager. You should be able to go to this URL and log in using the primary site administrator account you supplied when stepping through the Cloud Builder wizard. A healthy site will have a running sample map service, along with the preconfigured services in the System and Utilities folders.

Be sure to allow sufficient time for the site to launch before you make the above checks.

Is Windows Firewall enabled on the ArcGIS Server AMIs?

Windows Firewall is enabled on the ArcGIS Server AMIs, and all ports are open that are necessary for ArcGIS Server to run. Windows Firewall compliments the Amazon security groups. To allow inbound access on a port, you must allow the port using both Windows Firewall and the Amazon security group.

See Windows Firewall and the ArcGIS Server AMIs.

Is ArcGIS Server supported in Amazon Virtual Private Cloud (VPC)?

Yes. Beginning at version 10.1 Service Pack 1, ArcGIS Server Cloud Builder on Amazon Web Services gives you the option to launch your site within a VPC subnet. You can also launch ArcGIS Server instances in VPC using the AWS Management Console.

See Amazon Virtual Private Cloud (VPC) and ArcGIS Server.

I see an error about insufficient capacity when I click **Launch** to launch an instance.

This is an error from Amazon EC2 meaning that it does not have the available capacity to meet your request for a new instance. If your deployment architecture permits, you may be able to work around this error by requesting an instance in a different availability zone or letting EC2 choose the availability zone for you. Other options are to try launching a different size of instance or launching the instance later.

When I try to retrieve the administrator password for my EC2 instance on Windows, I get the message "No password was found."

This message can appear if you try to use **Get Windows Password** after you have stopped and started an EC2 instance. To avoid this error, the first time you log in, change the Administrator password to a value you can more easily remember.

How do I find out my Access Key and Secret Access Key?

When configuring client apps that use EC2, including ArcGIS Server Cloud Builder on Amazon Web Services, you may be asked to provide an Access Key and Secret Access Key. These unique keys are used to identify you (or an application acting in your behalf) when you make requests to Amazon Web Services.

Each Amazon account can have multiple users associated with it, each with their own levels of access and their own keys. To log in to Cloud Builder, you need to supply the access keys associated with a user of your account that has administrator access.

If you don't have a user with administrator access defined or you don't have the keys, you need to do the following:

1. Log in to the AWS Management Console and click **Services** > **IAM**.
2. Click **Create a New Group of Users**.
3. Type a group name and click **Continue**.
4. Click the **Select** button next to the **Administrator Access** policy template. Administrator privileges are required in order to use all the functions of Cloud Builder. Power users can do everything in Cloud Builder except manage SSL certificates.
5. Click **Continue** to accept the default security policy for an administrator.
6. Click the **Create New Users** tab and type the name for a user, such as your own name. Then click **Continue**.
7. Review the information about the group and user to be created and click **Continue**.
8. Click **Download Credentials** and save the CSV file to a secure location. You cannot download these keys later, so make sure you get them onto a local machine.
9. Open the CSV file in a spreadsheet or text editor and copy and paste the **Access Key** and **Secret Access Key** into the Cloud Builder login screen.

I got a message in the AWS Management Console that my instance is scheduled for retirement. What does this mean?

You might see this message if your instance happens to be running on degraded hardware that Amazon needs to replace. If you see this message, you should stop your site and start it again using ArcGIS Server Cloud Builder on Amazon Web Services. After this, the message should go away.