AWS Auto Scaling

User Guide

AWS Auto Scaling: User Guide

Table of Contents

What is AWS Auto Scaling? 1
Features of AWS Auto Scaling 1
Pricing 2
How to get started 2
Related services 2
How scaling plans work 3
Getting started 5
Step 1: Find your scalable resources 5
Prerequisites for your sample scaling plan 6
Add your Auto Scaling group to your sample scaling plan 6
Learn more about discovering your scalable resources 7
Step 2: Specify the scaling strategy 8
Step 3: Configure advanced settings (optional) 10
General settings 10
Dynamic scaling settings 11
Predictive scaling settings 12
Step 4: Create your scaling plan 13
(Optional) View scaling information for a resource 13
Step 5: Clean up 15
Delete your Auto Scaling group 15
Step 6: Next steps 15
Best practices 17
Other considerations 17
Avoiding the ActiveWithProblems error 18
Security 19
Data protection 19
Identity and Access Management 20
Access control 20
How AWS Auto Scaling works with IAM 20
Service-linked roles 23
Identity-based policy examples 24
Logging and monitoring 28
Compliance validation 29
Resilience 30
Infrastructure security 30
Using VPC endpoints for private connectivity 30
Create an interface VPC endpoint 30
Create a VPC endpoint policy 31
Endpoint migration 31
Service quotas 33
Resources 34
Document history 35

What is AWS Auto Scaling?

AWS Auto Scaling enables you to configure automatic scaling for the scalable resources that are part of your application in a matter of minutes. The AWS Auto Scaling console provides a single user interface to use the auto scaling features of multiple services in the AWS Cloud. You can configure automatic scaling for individual resources or for whole applications.

With AWS Auto Scaling, you configure and manage scaling for your resources through a scaling plan. The scaling plan uses dynamic scaling and predictive scaling to automatically scale your application's resources. This ensures that you add the required computing power to handle the load on your application and then remove it when it's no longer required. The scaling plan lets you choose scaling strategies to define how to optimize your resource utilization. You can optimize for availability, for cost, or a balance of both. Alternatively, you can create custom scaling strategies.

AWS Auto Scaling is useful for applications that experience daily or weekly variations in traffic flow, including the following:

- Cyclical traffic such as high use of resources during regular business hours and low use of resources overnight
- On and off workload patterns, such as batch processing, testing, or periodic analysis
- Variable traffic patterns, such as marketing campaigns with periods of spiky growth

Features of AWS Auto Scaling

Use AWS Auto Scaling to automatically scale the following resources:

- **Amazon EC2 Auto Scaling groups**: Launch or terminate EC2 instances in an Auto Scaling group.
- **Amazon EC2 Spot Fleet requests**: Launch or terminate instances from a Spot Fleet request, or automatically replace instances that get interrupted for price or capacity reasons.
- **Amazon ECS**: Adjust the ECS service desired count up or down in response to load variations.
- **Amazon DynamoDB**: Enable a DynamoDB table or a global secondary index to increase or decrease its provisioned read and write capacity to handle increases in traffic without throttling.
- **Amazon Aurora**: Dynamically adjust the number of Aurora read replicas provisioned for an Aurora DB cluster to handle changes in active connections or workload.

The scaling features currently available are dynamic scaling and predictive scaling.

Dynamic scaling creates target tracking scaling policies for the scalable resources in your application. This lets your scaling plan add and remove capacity for each resource as required to maintain resource utilization at the specified target value. The default scaling metrics provided are based on the most commonly used metrics used for automatic scaling.

How predictive scaling works:

- **Load forecasting**: AWS Auto Scaling analyzes up to 14 days of history for a specified load metric and forecasts the future demand for the next two days. This data is available in one-hour intervals and updated daily.
- **Scheduled scaling actions**: AWS Auto Scaling schedules the scaling actions that proactively add and remove resource capacity to reflect the load forecast. At the scheduled time, AWS Auto Scaling updates the resource's minimum capacity with the value specified by the scheduled scaling action.

The intention is to maintain resource utilization at the target value specified by the scaling strategy. If your application requires more capacity than is forecast, dynamic scaling is available to add additional capacity.

- **Maximum capacity behavior**: Each resource has a minimum and a maximum capacity limit between which the value specified by the scheduled scaling action is expected to lie. However, you can control whether your application can add resources beyond their maximum capacity when the forecast capacity is higher than the maximum capacity.

Currently, predictive scaling is only available for Amazon EC2 Auto Scaling groups.

Pricing

AWS Auto Scaling features are enabled by Amazon CloudWatch metrics and alarms. The features are provided at no additional charge beyond the service fees for CloudWatch and the other AWS Cloud resources that you use.

How to get started

For an introduction to AWS Auto Scaling, we recommend that you familiarize yourself with the following:

- How scaling plans work (p. 3)—This introduces the concepts of scaling strategies, dynamic scaling, and predictive scaling to help you get familiar with AWS Auto Scaling.
- AWS Auto Scaling FAQs—The FAQ on the product page provides information about the benefits of this service.
- Regions and endpoints in the *AWS General Reference*—This table shows you the regional availability of AWS Auto Scaling.
- Amazon EC2 Auto Scaling User Guide—This guide shows you how to create and manage the Auto Scaling groups to use when scaling your fleet of Amazon EC2 instances.
- Application Auto Scaling User Guide—This guide provides you with topics and resources related to automatic scaling of resources beyond Amazon EC2. Whenever you need more information specific to scaling an individual scalable resource or service other than Amazon EC2, you can access the technical documentation from this guide.

To get started, complete the getting started tutorial for AWS Auto Scaling in Getting started with AWS Auto Scaling (p. 5).

Related services

AWS CloudFormation allows you to use templates, which are formatted text files in JSON or YAML, to model and provision a collection of related Amazon Web Services resources. You can use AWS CloudFormation sample templates or create your own templates to create the resources, and any associated dependencies or runtime parameters, required to run your application. You can also create templates of scaling plans using AWS CloudFormation.

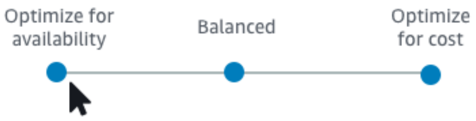
Amazon CloudWatch is a monitoring service for AWS Cloud resources and the applications you run on Amazon Web Services. CloudWatch lets you collect and track metrics, log files, and automatically react to changes in your applications using alarms. You can also publish your own custom metrics to CloudWatch using the AWS CLI or an API.

How scaling plans work

A scaling plan is the core component of AWS Auto Scaling. It's where you configure a set of instructions for scaling your resources. If you work with AWS CloudFormation or add tags to scalable resources, you can set up scaling plans for different sets of resources, per application. AWS Auto Scaling provides recommendations for scaling strategies customized to each resource. After you create your scaling plan, AWS Auto Scaling combines dynamic scaling and predictive scaling methods together to support your scaling strategy.

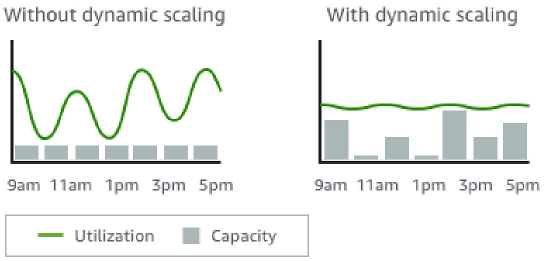
What is a scaling strategy?

The scaling strategy tells AWS Auto Scaling how to optimize the utilization of resources in your scaling plan. You can optimize for availability, for cost, or a balance of both. Alternatively, you can also create your own custom strategy, per the metrics and thresholds you define. You can set separate strategies for each resource or resource type.



What is dynamic scaling?

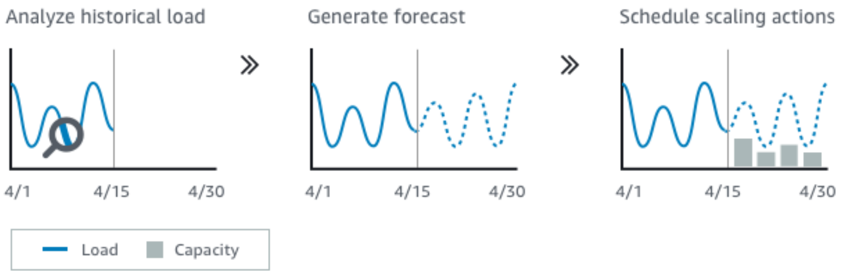
Dynamic scaling creates target tracking scaling policies for the resources in your scaling plan. These scaling policies adjust resource capacity in response to live changes in resource utilization. The intention is to provide enough capacity to maintain utilization at the target value specified by the scaling strategy. This is similar to the way that your thermostat maintains the temperature of your home. You choose the temperature and the thermostat does the rest.



For example, you can configure your scaling plan to keep the number of tasks that your ECS service runs at 75 percent of CPU. When the CPU utilization of your service rises above 75 percent (meaning that more than 75 percent of the CPU that is reserved for the service is being used), this triggers your scaling policy to add another task to your service to help out with the increased load.

What is predictive scaling?

Predictive scaling uses machine learning to analyze each resource's historical workload and regularly forecasts the future load for the next two days. This is similar to how weather forecasts work. Using the forecast, predictive scaling generates scheduled scaling actions to make sure that the resource capacity is available before your application needs it. Like dynamic scaling, predictive scaling works to maintain utilization at the target value specified by the scaling strategy.



For example, you can enable predictive scaling and configure your scaling strategy to keep the average CPU utilization of your Auto Scaling group at 50 percent. Your forecast calls for traffic spikes to occur every day at 8 o'clock in the morning. Your scaling plan creates the future scheduled scaling actions to make sure that your Auto Scaling group is ready to handle that traffic ahead of time. This helps keep the application performance constant, with the aim of always having the capacity required to maintain resource utilization as close to 50 percent as possible at all times.

Getting started with AWS Auto Scaling

This section describes the steps to begin using AWS Auto Scaling. You can use the AWS Management Console to create your first scaling plan. Then you learn the basics of creating scaling plans with predictive scaling and dynamic scaling enabled.

Before you create a scaling plan for use with your application, review your application thoroughly as it runs in the Amazon Web Services Cloud. Take note of the following:

- Whether you have existing scaling policies created from other consoles. You can replace the existing scaling policies, or you can keep them (without being allowed to make any changes to their values) when you create your scaling plan.
- The target utilization that makes sense for each scalable resource in your application based on the resource as a whole. For example, the amount of CPU that the EC2 instances in an Auto Scaling group are expected to use compared to their available CPU. Or for a service like DynamoDB that uses a provisioned throughput model, the amount of read and write activity that a table or index is expected to use compared to the available throughput. In other words, the ratio of consumed to provisioned capacity. You can change the target utilization at any time after you you've created your scaling plan.
- How long it takes to launch and configure a server. Knowing this will help you configure a window for each EC2 instance to warm up after launching to ensure that we don't launch a new server while the previous one is still launching.
- Whether the metric history is sufficiently long to use with predictive scaling (if using newly created Auto Scaling groups). In general, having a full 14 days of historical data translates into more accurate forecasts. The minimum is 24 hours.

The better you understand your application, the more effective you can make your scaling plan.

Topics

- Step 1: Find your scalable resources (p. 5)
- Step 2: Specify the scaling strategy (p. 8)
- Step 3: Configure advanced settings (optional) (p. 10)
- Step 4: Create your scaling plan (p. 13)
- Step 5: Clean up (p. 15)
- Step 6: Next steps (p. 15)

Step 1: Find your scalable resources

In the getting started section, you create a scaling plan and get a hands-on introduction to using AWS Auto Scaling through the AWS Management Console.

When you create a scaling plan from the console, AWS Auto Scaling helps you find your scalable resources as a first step. There are three ways to locate the resources for a new scaling plan from the console:

- You can choose an AWS CloudFormation stack for the AWS Auto Scaling console to use to automatically discover your scalable resources.
- You can choose a set of tags for the AWS Auto Scaling console to use to automatically discover your scalable resources.
- You can choose one or more Amazon EC2 Auto Scaling groups to add to your scaling plan.

If this is your first scaling plan, we recommend that you start by choosing the third option and create a sample scaling plan using an EC2 Auto Scaling group.

Prerequisites for your sample scaling plan

For a beginner-friendly tutorial for using the console to create a scaling plan, we recommend that you start by creating an Auto Scaling group, and then create the scaling plan and add the Auto Scaling group. By using an Auto Scaling group, you can enable the predictive scaling feature and the dynamic scaling feature. You must enable both features to use the full set of features that are available in your scaling plan.

Get started by creating an Auto Scaling group, if you do not already have one. For more information, see Getting started with Amazon EC2 Auto Scaling in the *Amazon EC2 Auto Scaling User Guide*. If you create a new group, you can delete it afterwards. As soon as the group is deleted, you stop incurring charges for the Amazon EC2 instances it ran.

Configure your Auto Scaling group as follows to ensure the scaling plan works as expected:

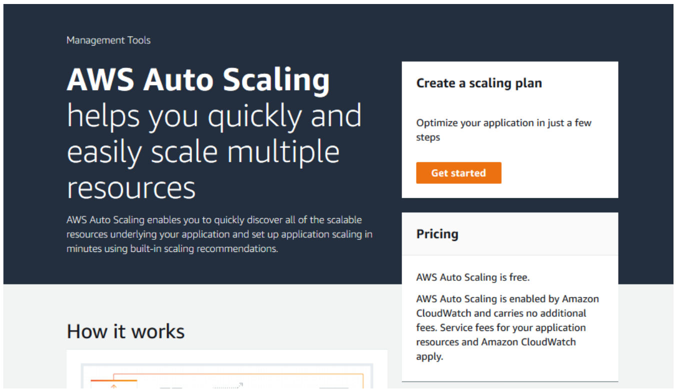
- In the launch template or launch configuration that you associate with the Auto Scaling group, enable detailed monitoring to get CloudWatch metric data for EC2 instances at a 1-minute frequency. Additional charges apply. For more information, see Configure monitoring for Auto Scaling instances in the *Amazon EC2 Auto Scaling User Guide*.
- Enable Auto Scaling group metrics to get aggregated data for your group of instances in CloudWatch. For more information, see Enable Auto Scaling group metrics in the *Amazon EC2 Auto Scaling User Guide*.
- If you use a T2 or T3 instance type, use a launch template to configure your instances as `unlimited` so that they can sustain high CPU performance while you're testing. Additional charges may apply. For more information, see Using an Auto Scaling group to launch a burstable performance instance as Unlimited in the *Amazon EC2 User Guide for Linux Instances*.

Add your Auto Scaling group to your sample scaling plan

Now that you've created your Auto Scaling group, you're ready to create your sample scaling plan using the AWS Management Console.

To add an Auto Scaling group to a new scaling plan

1. Open the AWS Auto Scaling console at https://console.aws.amazon.com/awsautoscaling/.
2. On the navigation bar at the top of the screen, choose the same Region that you used when you created your Auto Scaling group.
3. From the welcome page, choose **Get started**.



Management Tools

AWS Auto Scaling helps you quickly and easily scale multiple resources

AWS Auto Scaling enables you to quickly discover all of the scalable resources underlying your application and set up application scaling in minutes using built-in scaling recommendations.

Create a scaling plan

Optimize your application in just a few steps

Pricing

AWS Auto Scaling is free.

AWS Auto Scaling is enabled by Amazon CloudWatch and carries no additional fees. Service fees for your application resources and Amazon CloudWatch apply.

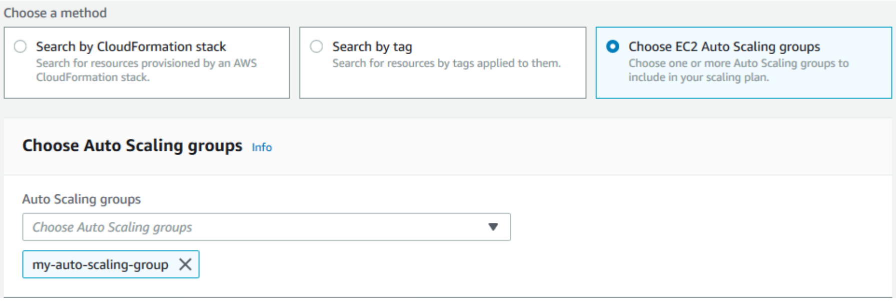
How it works

4. On the **Find scalable resources** page, choose **Search by CloudFormation stack**, **Search by tag**, or **Choose EC2 Auto Scaling groups**.

   Note

   This tutorial assumes you're choosing an Auto Scaling group. Later, you can use this same procedure to create a scaling plan using the **Search by CloudFormation stack** or **Search by tag** option.

   - If you chose **Search by CloudFormation stack**, choose the AWS CloudFormation stack to use.
   - If you chose **Search by tag**, then for each tag, choose a tag key from **Key** and tag values from **Value**. To add tags, choose **Add another row**. To remove tags, choose **Remove**.
   - If you chose **Choose EC2 Auto Scaling groups**, then for **Auto Scaling groups**, choose one or more Auto Scaling groups.



Choose a method

○ Search by CloudFormation stack
Search for resources provisioned by an AWS CloudFormation stack.

○ Search by tag
Search for resources by tags applied to them.

● Choose EC2 Auto Scaling groups
Choose one or more EC2 Auto Scaling groups to include in your scaling plan.

Choose Auto Scaling groups Info

Auto Scaling groups

Choose Auto Scaling groups

my-auto-scaling-group

5. Choose **Next** to add the Auto Scaling group to the scaling plan and to proceed to the next step.

   If you chose the **Search by CloudFormation stack** or **Search by tag** option, then choosing **Next** makes the scalable resources associated with the stack or set of tags available to the scaling plan. As you define your scaling plan, you can then choose which of these resources to include or exclude.

Learn more about discovering your scalable resources

If you have already created a sample scaling plan and would like to create more, the following information explains the scenarios for using a CloudFormation stack or a set of tags in more detail. You can use this section to decide whether to choose the **Search by CloudFormation stack** or **Search by tag** option to discover your scalable resources when using the console to create your scaling plan.

Discovering scalable resources using a CloudFormation stack

When you use CloudFormation, you work with stacks to provision resources. All of the resources in a stack are defined by the stack's template. Your scaling plan adds an orchestration layer on top of the stack that makes it easier to configure scaling for multiple resources. Without a scaling plan, you would need to set up scaling for each scalable resource individually. This means figuring out the order for provisioning resources and scaling policies, and understanding the subtleties of how these dependencies work.

In the AWS Auto Scaling console, you can select an existing stack to scan it for resources that can be configured for automatic scaling. AWS Auto Scaling only finds resources that are defined in the selected stack. It does not traverse through nested stacks.

For your ECS services to be discoverable in a CloudFormation stack, the AWS Auto Scaling console must know which ECS cluster is running the service. This requires that your ECS services be in the same CloudFormation stack as the ECS cluster that is running the service. Otherwise, they must be part of the default cluster. To be identified correctly, the ECS service name must also be unique across each of these ECS clusters.

For more information about CloudFormation, see What is AWS CloudFormation? in the *AWS CloudFormation User Guide*.

Discovering scalable resources using tags

Tags provide metadata that can be used to discover related scalable resources in the AWS Auto Scaling console, using tag filters.

Use tags to find any of the following resources:

- Aurora DB clusters
- Auto Scaling groups
- DynamoDB tables and global secondary indexes

When you search by more than one tag, each resource must have all of the listed tags to be discovered.

Tags can be assigned in a number of ways. For more information, see Tagging AWS resources in the *AWS General Reference*.

Step 2: Specify the scaling strategy

Use the following procedure to specify scaling strategies for the resources that were found in the previous step.

For each type of resource, AWS Auto Scaling chooses the metric that is most commonly used for determining how much of the resource is in use at any given time. You choose the most appropriate scaling strategy to optimize performance of your application based on this metric. When you enable the dynamic scaling feature and the predictive scaling feature, the scaling strategy is shared between them. For more information, see How scaling plans work (p. 3).

The following scaling strategies are available:

- **Optimize for availability**—AWS Auto Scaling scales the resource out and in automatically to maintain resource utilization at 40 percent. This option is useful when your application has urgent and sometimes unpredictable scaling needs.
- **Balance availability and cost**—AWS Auto Scaling scales the resource out and in automatically to maintain resource utilization at 50 percent. This option helps you maintain high availability while also reducing costs.
- **Optimize for cost**—AWS Auto Scaling scales the resource out and in automatically to maintain resource utilization at 70 percent. This option is useful for lowering costs if your application can handle having reduced buffer capacity when there are unexpected changes in demand.

For example, the scaling plan configures your Auto Scaling group to add or remove Amazon EC2 instances based on how much of the CPU is used on average for all instances in the group. You choose whether to optimize utilization for availability, cost, or a combination of the two by changing the scaling strategy.

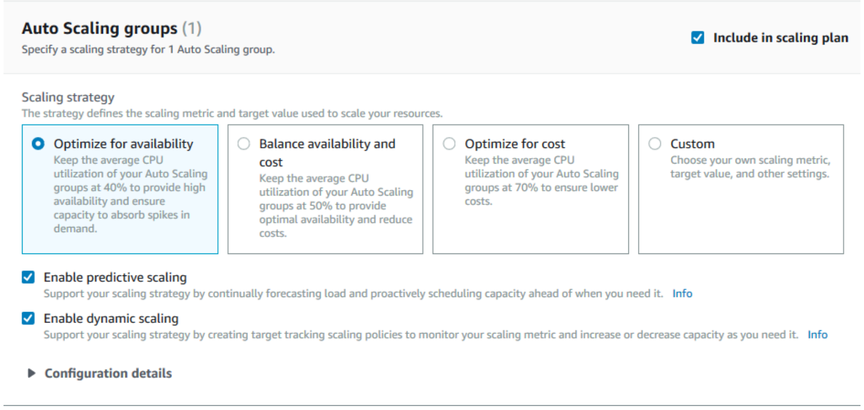
Alternatively, you can configure a custom strategy if an off-the-shelf strategy doesn't meet your needs. With a custom strategy, you can change the target utilization value, choose a different metric, or both.

> **Important**
> For the beginner tutorial, complete only the first step of the following procedure and then choose **Next** to continue. (You can skip the rest of the procedure because the tutorial focuses on using the default scaling strategy, **Optimize for availability**, that keeps the average CPU utilization of your Auto Scaling group at 40 percent.)

To specify scaling strategies

1. On the **Specify scaling strategy** page, for **Scaling plan details**, **Name**, enter a name for your scaling plan. The name of your scaling plan must be unique within your set of scaling plans for the Region, can have a maximum of 128 characters, and must not contain pipes "|", forward slashes "/", or colons ":".
2. For each type of resource, provide the following scaling instructions.

    a. For the **Scaling strategy**, choose one of these options: **Optimize for availability**, **Balance availability and cost**, **Optimize for cost**, or **Custom**.



Auto Scaling groups (1)
Specify a scaling strategy for 1 Auto Scaling group.

☑ **Include in scaling plan**

Scaling strategy
The strategy defines the scaling metric and target value used to scale your resources.

● Optimize for availability
Keep the average CPU utilization of your Auto Scaling groups at 40% to provide high availability and ensure capacity to absorb spikes in demand.

○ Balance availability and cost
Keep the average CPU utilization of your Auto Scaling groups at 50% to provide optimal availability and reduce costs.

○ Optimize for cost
Keep the average CPU utilization of your Auto Scaling groups at 70% to ensure lower costs.

○ Custom
Choose your own scaling metric, target value, and other settings.

☑ Enable predictive scaling
Support your scaling strategy by continually forecasting load and proactively scheduling capacity ahead of when you need it. Info

☑ Enable dynamic scaling
Support your scaling strategy by creating target tracking scaling policies to monitor your scaling metric and increase or decrease capacity as you need it. Info

▶ **Configuration details**

    b. If you chose **Custom** in the previous step, choose your custom settings under **Configuration details**. Here you can find the list of metrics available to you (if any) and related graphs based on data from CloudWatch. The recent metric history is the main focus of the graphs.

    - For **Scaling metric**, choose the desired scaling metric. If there are no other predefined metrics available, this option has no drop-down list to show.
    - For **Target value**, choose the desired target utilization value.
    - For **Load metric** [Auto Scaling groups only], choose an appropriate load metric to use for predictive scaling.
    - For **Replace external scaling policies**, choose whether to delete scaling policies created from outside of the scaling plan (such as from other consoles) and replace them with new target tracking scaling policies created by the scaling plan.

    c. (Optional) By default, predictive scaling is enabled for your Auto Scaling groups. To disable predictive scaling for your Auto Scaling groups, clear **Enable predictive scaling**.

    d. (Optional) By default, dynamic scaling is enabled for all resource types. To disable dynamic scaling for a type of resource, clear **Enable dynamic scaling**.

   e. (Optional) By default, when you specify an application source from which multiple scalable resources are discovered, all resource types are automatically included in your scaling plan. To omit a type of resource from your scaling plan, clear **Include in scaling plan**.

3. When you are finished, choose **Next**.

Step 3: Configure advanced settings (optional)

Now that you have specified the scaling strategy to use for each resource type, you can choose to customize any of the default settings on a per resource basis using the **Configure advanced settings** step. For each resource type, there are multiple groups of settings that you can customize. In most cases, however, the default settings should be optimal, with the possible exception of the values for minimum capacity and maximum capacity, which should be carefully adjusted.

Skip this procedure if you would like to keep the default settings. You can change these settings anytime by editing the scaling plan.

> **Important**
> For the beginner tutorial, let's make a few changes to update the maximum capacity of your Auto Scaling group and enable predictive scaling in forecast only mode. Although you do not need to customize all of the settings for the tutorial, let's also briefly examine the settings in each section.

General settings

Use this procedure to view and customize the settings you specified in the previous step, on a per resource basis. You can also customize the minimum capacity and maximum capacity for each resource.

To view and customize the general settings

1. On the **Configure advanced settings** page, choose the arrow to the left of any of the section headings to expand the section. For the tutorial, expand the **Auto Scaling groups** section.
2. From the table that's displayed, choose the Auto Scaling group that you are using in this tutorial.
3. Leave the **Include in scaling plan** option selected. If this option is not selected, the resource is omitted from the scaling plan. If you do not include at least one resource, the scaling plan cannot be created.
4. To expand the view and see the details of the **General Settings** section, choose the arrow to the left of the section heading.
5. You can make choices for any of the following items. For this tutorial, locate the **Maximum capacity** setting and enter a value of `3` in place of the current value.

   - **Scaling strategy**—Allows you to optimize for availability, cost, or a balance of both, or to specify a custom strategy.
   - **Enable dynamic scaling**—If this setting is cleared, the selected resource cannot scale using a target tracking scaling configuration.
   - **Enable predictive scaling**—[Auto Scaling groups only] If this setting is cleared, the selected group cannot scale using predictive scaling.
   - **Scaling metric**—Specifies the scaling metric to use. If you choose **Custom**, you can specify a custom metric to use instead of the predefined metrics that are available in the console. For more information, see the next topic in this section.
   - **Target value**—Specifies the target utilization value to use.
   - **Load metric**—[Auto Scaling groups only] Specifies the load metric to use. If you choose **Custom**, you can specify a custom metric to use instead of the predefined metrics that are available in the console. For more information, see the next topic in this section.

- **Minimum capacity**—Specifies the minimum capacity for the resource. AWS Auto Scaling ensures that your resource never goes below this size.
- **Maximum capacity**—Specifies the maximum capacity for the resource. AWS Auto Scaling ensures that your resource never goes above this size.

  > **Note**
  > When you use predictive scaling, you can optionally choose a different maximum capacity behavior to use based on the forecast capacity. This setting is in the **Predictive scaling settings** section.

Custom metrics

AWS Auto Scaling provides the most commonly used metrics for automatic scaling. However, depending on your needs, you might prefer to get data from different metrics instead of the metrics in the console. Amazon CloudWatch has many different metrics to choose from. CloudWatch also lets you publish your own metrics.

You use JSON to specify a CloudWatch custom metric. Before you follow these instructions, we recommend that you become familiar with the Amazon CloudWatch User Guide.

To specify a custom metric, you construct a JSON-formatted payload using a set of required parameters from a template. You add the values for each parameter from CloudWatch. We provide the template as part of the custom options for **Scaling metric** and **Load metric** in the advanced settings of your scaling plan.

JSON represents data in two ways:

- An *object*, which is an unordered collection of name-value pairs. An object is defined within left ({) and right (}) braces. Each name-value pair begins with the name, followed by a colon, followed by the value. Name-value pairs are comma-separated.
- An *array*, which is an ordered collection of values. An array is defined within left ([) and right (]) brackets. Items in the array are comma-separated.

Here is an example of the JSON template with sample values for each parameter:

```
{
  "MetricName": "MyBackendCPU",
  "Namespace": "MyNamespace",
  "Dimensions": [
    {
      "Name": "MyOptionalMetricDimensionName",
      "Value": "MyOptionalMetricDimensionValue"
    }
  ],
  "Statistic": "Sum"
}
```

For more information, see Customized scaling metric specification and Customized load metric specification in the *AWS Auto Scaling API Reference*.

Dynamic scaling settings

Use this procedure to view and customize the settings for the target tracking scaling policy that AWS Auto Scaling creates.

To view and customize the settings for dynamic scaling

1. To expand the view and see the details of the **Dynamic scaling settings** section, choose the arrow to the left of the section heading.
2. You can make choices for the following items. However, the default settings are fine for this tutorial.

   - **Replace external scaling policies**—If this setting is cleared, it keeps existing scaling policies created from outside of this scaling plan, and does not create new ones.
   - **Disable scale-in**—If this setting is cleared, automatic scale-in to decrease the current capacity of the resource is allowed when the specified metric is below the target value.
   - **Cooldown**—Creates scale-out and scale-in cooldown periods. The cooldown period is the amount of time the scaling policy waits for a previous scaling activity to take effect. For more information, see Cooldown period in the *Application Auto Scaling User Guide*. (This setting is not shown if the resource is an Auto Scaling group.)
   - **Instance warmup**—[Auto Scaling groups only] Controls the amount of time that elapses before a newly launched instance begins contributing to the CloudWatch metrics. For more information, see Instance warmup in the *Amazon EC2 Auto Scaling User Guide*.

Predictive scaling settings

If your resource is an Auto Scaling group, use this procedure to view and customize the settings AWS Auto Scaling uses for predictive scaling.

To view and customize the settings for predictive scaling

1. To expand the view and see the details of the **Predictive scaling settings** section, choose the arrow to the left of the section heading.
2. You can make choices for the following items. For this tutorial, change the **Predictive scaling mode** to **Forecast only**.

   - **Predictive scaling mode**—Specifies the scaling mode. The default is **Forecast and scale**. If you change it to **Forecast only**, the scaling plan forecasts future capacity but doesn't apply the scaling actions.
   - **Pre-launch instances**—Adjusts the scaling actions to run earlier when scaling out. For example, the forecast says to add capacity at 10:00 AM, and the buffer time is 5 minutes (300 seconds). The runtime of the corresponding scaling action is then 9:55 AM. This is helpful for Auto Scaling groups, where it can take a few minutes from the time an instance launches until it comes in service. The actual time can vary as it depends on several factors, such as the size of the instance and whether there are startup scripts to complete. The default is 300 seconds.
   - **Max capacity behavior**—Controls whether the selected resource can scale up above the maximum capacity when the forecast capacity is close to or exceeds the currently specified maximum capacity. The default is **Enforce the maximum capacity setting**.
     - **Enforce the maximum capacity setting**—AWS Auto Scaling cannot scale resource capacity higher than the maximum capacity. The maximum capacity is enforced as a hard limit.
     - **Set the maximum capacity to equal forecast capacity**—AWS Auto Scaling can scale resource capacity higher than the maximum capacity to equal but not exceed forecast capacity.
     - **Increase maximum capacity above forecast capacity**—AWS Auto Scaling can scale resource capacity higher than the maximum capacity by a specified buffer value. The intention is to give the target tracking scaling policy extra capacity if unexpected traffic occurs.
   - **Max capacity behavior buffer**—If you chose **Increase maximum capacity above forecast capacity**, choose the size of the capacity buffer to use when the forecast capacity is close to or exceeds the maximum capacity. The value is specified as a percentage relative to the forecast capacity. For example, with a 10 percent buffer, if the forecast capacity is 50, and the maximum capacity is 40, then the effective maximum capacity is 55.

3. When you are finished customizing settings, choose **Next**.

   Note
   To revert any of your changes, select the resources and choose **Revert to original**. This resets the selected resources to their last known state within the scaling plan.

Step 4: Create your scaling plan

On the **Review and create** page, review the details of your scaling plan and choose **Create scaling plan**. You are directed to a page that shows the status of your scaling plan. The scaling plan can take a moment to finish being created while your resources are updated.

With predictive scaling, AWS Auto Scaling analyzes the history of the specified load metric from the past 14 days (minimum of 24 hours of data is required) to generate a forecast for two days ahead. It then schedules scaling actions to adjust the resource capacity to match the forecast for each hour in the forecast period.

After the creation of the scaling plan is complete, view the scaling plan details by choosing its name from the **Scaling plans** screen.

(Optional) View scaling information for a resource

Use this procedure to view the scaling information created for a resource.

Data is presented in the following ways:

- Graphs showing recent metric history data from CloudWatch.
- Predictive scaling graphs showing load forecasts and capacity forecasts based on data from AWS Auto Scaling.
- A table that lists all the predictive scaling actions scheduled for the resource.

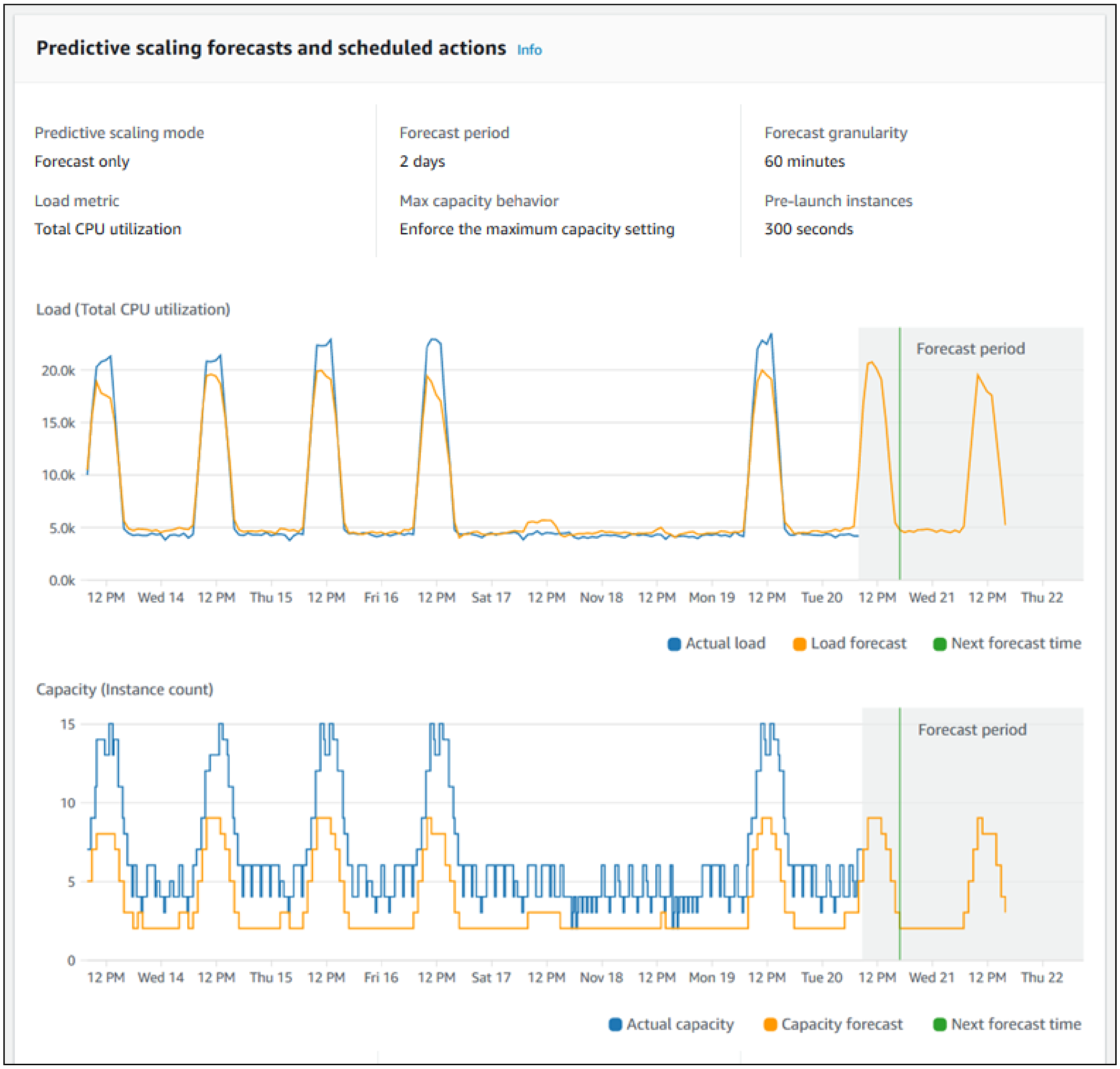
To view scaling information for a resource

1. Open the AWS Auto Scaling console at https://console.aws.amazon.com/awsautoscaling/.
2. On the **Scaling plans** page, choose the scaling plan.
3. On the **Scaling plan details** page, choose the resource to view.

Monitoring and evaluating forecasts

When your scaling plan is up and running, you can monitor the load forecast, the capacity forecast, and scaling actions to examine the performance of predictive scaling. All of this data is available in the AWS Auto Scaling console for all Auto Scaling groups that are enabled for predictive scaling. Keep in mind that your scaling plan requires at least 24 hours of historical load data to make the initial forecast.

In the following example, the left side of each graph shows a historical pattern. The right side shows the forecast that was generated by the scaling plan for the forecast period. Both actual and forecast values (in blue and orange) are plotted.



Predictive scaling forecasts and scheduled actions Info

Predictive scaling mode Forecast only	Forecast period 2 days	Forecast granularity 60 minutes
Load metric Total CPU utilization	Max capacity behavior Enforce the maximum capacity setting	Pre-launch instances 300 seconds

AWS Auto Scaling learns from your data automatically. First, it makes a load forecast. Then, a capacity forecast calculation determines the minimum number of instances that are required to support the application. Based on the capacity forecast, AWS Auto Scaling schedules scaling actions that scale the Auto Scaling group in advance of predicted load changes. If dynamic scaling is enabled (recommended), the Auto Scaling group can scale out additional capacity (or remove capacity) based on the current utilization of the group of instances.

When evaluating how well predictive scaling performs, monitor how closely the actual and forecast values match *over time*. When you create a scaling plan, AWS Auto Scaling provides graphs based on the most recent actual data. It also provides an initial forecast for the next 48 hours. However, when the scaling plan is created, there is very little forecast data to compare the actual data to. Wait until the scaling plan has obtained forecast values for a few periods before comparing the historical forecast values against the actual values. After a few days of daily forecasts, you'll have a larger sample of forecast values to compare with actual values.

For patterns that occur on a daily basis, the time interval between creating your scaling plan and evaluating the forecast effectiveness can be as short as a few days. However, this length of time is insufficient to evaluate the forecast based on a recent pattern change. For example, let's say you are looking at the forecast for an Auto Scaling group that started a new marketing campaign in the past week. The campaign significantly increases your web traffic for the same two days each week. In situations like this, we recommend waiting for the group to collect a full week or two of new data before evaluating the effectiveness of the forecast. The same recommendation applies for a brand new Auto Scaling group that has only started to collect metric data.

If the actual and forecast values don't match after monitoring them over an appropriate length of time, you should also consider your choice of load metric. To be effective, the load metric must represent a reliable and accurate measure of the total load on all instances in the Auto Scaling group. The load metric is core to predictive scaling. If you choose a non-optimal load metric, it can prevent predictive scaling from making accurate load and capacity forecasts and scheduling the correct capacity adjustments for your Auto Scaling group.

Step 5: Clean up

After you have completed the getting started tutorial, you can choose to keep your scaling plan. However, if you are not actively using your scaling plan, you should consider deleting it so that your account does not incur unnecessary charges.

Deleting a scaling plan deletes the target tracking scaling policies, their associated CloudWatch alarms, and the predictive scaling actions that AWS Auto Scaling created on your behalf.

Deleting a scaling plan does not delete your AWS CloudFormation stack, Auto Scaling group, or other scalable resources.

To delete a scaling plan

1. Open the AWS Auto Scaling console at https://console.aws.amazon.com/awsautoscaling/.
2. On the **Scaling plans** page, select the scaling plan that you created for this tutorial and choose **Delete**.
3. When prompted for confirmation, choose **Delete**.

After you delete your scaling plan, your resources do not revert to their original capacity. For example, if your Auto Scaling group is scaled to 10 instances when you delete the scaling plan, your group is still scaled to 10 instances after the scaling plan is deleted. You can update the capacity of specific resources by accessing the console for each individual service.

Delete your Auto Scaling group

To prevent your account from accruing Amazon EC2 charges, you should also delete the Auto Scaling group that you created for this tutorial.

For step-by-step instructions, see Delete your Auto Scaling group in the *Amazon EC2 Auto Scaling User Guide*.

Step 6: Next steps

Now that you have familiarized yourself with AWS Auto Scaling and some of its features, you may want to try creating your own scaling plan template using AWS CloudFormation.

An AWS CloudFormation template is a JSON or YAML-formatted text file that describes the Amazon Web Services infrastructure needed to run an application or service along with any interconnections among infrastructure components. With AWS CloudFormation, you deploy and manage an associated collection of resources as a *stack*. AWS CloudFormation is available at no additional charge, and you pay only for the AWS resources needed to run your applications. Resources can consist of any AWS resource you define within the template. For more information, see AWS CloudFormation concepts in the *AWS CloudFormation User Guide*.

In the *AWS CloudFormation User Guide*, we provide a simple template to get you started. The sample template is available as an example in the AWS::AutoScalingPlans::ScalingPlan section of the AWS CloudFormation template reference documentation. The sample template creates a scaling plan for a single Auto Scaling group and enables predictive scaling and dynamic scaling.

For more information, see Getting started with AWS CloudFormation in the *AWS CloudFormation User Guide*.

Best practices for AWS Auto Scaling scaling plans

The following best practices can help you make the most of scaling plans:

- Wherever possible, you should scale on Amazon EC2 instance metrics with a 1-minute frequency because that ensures a faster response to utilization changes. Scaling on metrics with a 5-minute frequency can result in a slower response time and scaling on stale metric data. By default, EC2 instances are enabled for basic monitoring, which means metric data for instances is available at 5-minute intervals. For an additional charge, you can enable detailed monitoring to get metric data for instances at a 1-minute frequency. For more information, see Configure monitoring for Auto Scaling instances in the *Amazon EC2 Auto Scaling User Guide*.
- We also recommend that you enable Auto Scaling group metrics. Otherwise, actual capacity data is not shown in the capacity forecast graphs that are available on completion of the Create Scaling Plan wizard. To enable Auto Scaling group metrics, open an Auto Scaling group in the Amazon EC2 console, and from the **Monitoring** tab, choose **Enable Group Metrics Collection**. These metrics describe the group rather than any of its instances. For more information, see Enable Auto Scaling group metrics in the *Amazon EC2 Auto Scaling User Guide*.
- Check which instance type your Auto Scaling group uses. Amazon EC2 instances with burstable performance, which are T3 and T2 instances, are designed to provide a baseline level of CPU performance with the ability to burst to a higher level when required by your workload. Depending on the target utilization specified by the scaling plan, you could run the risk of exceeding the baseline and then running out of CPU credits, which limits performance. For more information, see CPU credits and baseline performance for burstable performance instances. To configure these instances as `unlimited`, see Using an Auto Scaling group to launch a burstable performance instance as Unlimited in the *Amazon EC2 User Guide for Linux Instances*.

Other considerations

Keep the following additional considerations in mind:

- Predictive scaling uses workload forecasts to schedule capacity in the future. The quality of the forecasts varies based on how cyclical the workload is and the applicability of the trained forecasting model. Predictive scaling can be run in forecast only mode to assess the quality of the forecasts and the scaling actions created by the forecasts. You can set the predictive scaling mode to **Forecast only** when you create the scaling plan and then change it to **Forecast and scale** when you're finished assessing the forecast quality. For more information, see Predictive scaling settings (p. 12) and Monitoring and evaluating forecasts (p. 13).
- If you choose to specify different metrics for predictive scaling, you must ensure that the scaling metric and load metric are strongly correlated. The metric value must increase and decrease proportionally to the number of instances in the Auto Scaling group. This ensures that the metric data can be used to proportionally scale out or in the number of instances. For example, the load metric is total request count and the scaling metric is average CPU utilization. If the total request count increases by 50 percent, the average CPU utilization should also increase by 50 percent, provided that capacity remains unchanged.
- Before creating your scaling plan, you should delete any previously scheduled scaling actions that you no longer need by accessing the consoles they were created from. AWS Auto Scaling does not create a predictive scaling action that overlaps an existing scheduled scaling action.

- Your customized settings for minimum and maximum capacity, along with other settings used for dynamic scaling, show up in other consoles. However, we recommend that after you create a scaling plan, you do not modify these settings from other consoles because your scaling plan does not receive the updates from other consoles.
- Your scaling plan can contain resources from multiple services, but each resource can be in only one scaling plan at a time.

Avoiding the ActiveWithProblems error

An "ActiveWithProblems" error can occur when a scaling plan is created, or resources are added to a scaling plan. The error occurs when the scaling plan is active, but the scaling configuration for one or more resources could not be applied.

Usually, this happens because a resource already has a scaling policy or an Auto Scaling group does not meet the minimum requirements for predictive scaling.

If any of your resources already have scaling policies from various service consoles, AWS Auto Scaling does not overwrite these other scaling policies or create new ones by default. You can optionally delete the existing scaling policies and replace them with target tracking scaling policies created from the AWS Auto Scaling console. You do this by enabling the **Replace external scaling policies** setting for each resource that has scaling policies to overwrite.

With predictive scaling, we recommend waiting 24 hours after creating a new Auto Scaling group to configure predictive scaling. At minimum, there must be 24 hours of historical data to generate the initial forecast. If the group has less than 24 hours of historical data and predictive scaling is enabled, this results in the scaling plan being unable to generate a forecast until the next forecast period after the group has collected the required amount of data. However, you can also edit and save the scaling plan to restart the forecast process as soon as the 24 hours of data is available.

Security in AWS Auto Scaling

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Auto Scaling, see AWS services in scope by compliance program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Auto Scaling. The following topics show you how to configure AWS Auto Scaling to meet your security and compliance objectives. You also learn how to use other Amazon Web Services that help you to monitor and secure your AWS Auto Scaling resources.

Topics

- AWS Auto Scaling and data protection (p. 19)
- Identity and Access Management for AWS Auto Scaling (p. 20)
- Logging and monitoring in AWS Auto Scaling (p. 28)
- Compliance validation for AWS Auto Scaling (p. 29)
- Resilience in AWS Auto Scaling (p. 30)
- Infrastructure security in AWS Auto Scaling (p. 30)
- AWS Auto Scaling and interface VPC endpoints (p. 30)

AWS Auto Scaling and data protection

The AWS shared responsibility model applies to data protection in AWS Auto Scaling. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.

- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with AWS Auto Scaling or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Identity and Access Management for AWS Auto Scaling

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS Auto Scaling resources. IAM is an AWS service that you can use with no additional charge.

To use AWS Auto Scaling, you need an Amazon Web Services account and credentials. To increase the security of your account, we recommend that you use an *IAM user* to provide access credentials instead of using your Amazon Web Services account credentials. For more information, see Amazon Web Services account root user credentials vs. IAM user credentials in the *AWS General Reference* and IAM best practices in the *IAM User Guide*.

For an overview of IAM users and why they are important for the security of your account, see AWS security credentials in the *AWS General Reference*.

For details about working with IAM, see the *IAM User Guide*.

Access control

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access AWS Auto Scaling resources. For example, you must have permissions to create scaling plans, configure predictive scaling, and so on.

The following sections provide details on how an IAM administrator can use IAM to help secure your AWS resources, by controlling who can perform AWS Auto Scaling actions.

Topics

- How AWS Auto Scaling works with IAM (p. 20)
- AWS Auto Scaling service-linked roles (p. 23)
- AWS Auto Scaling identity-based policy examples (p. 24)

How AWS Auto Scaling works with IAM

Before you use IAM to manage access to AWS Auto Scaling, you should understand what IAM features are available to use with AWS Auto Scaling. To get a high-level view of how AWS Auto Scaling and other AWS services work with IAM, see AWS services that work with IAM in the *IAM User Guide*.

Topics

- AWS Auto Scaling identity-based policies (p. 21)
- AWS Auto Scaling resource-based policies (p. 22)
- Access Control Lists (ACLs) (p. 22)
- Authorization based on AWS Auto Scaling tags (p. 22)
- AWS Auto Scaling IAM roles (p. 22)

AWS Auto Scaling identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources, and the conditions under which actions are allowed or denied. AWS Auto Scaling supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in AWS Auto Scaling use the following prefix before the action: `autoscaling-plans:`. Policy statements must include either an `Action` or `NotAction` element. AWS Auto Scaling defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as shown in the following example.

```
"Action": [
      "autoscaling-plans:DescribeScalingPlans",
      "autoscaling-plans:DescribeScalingPlanResources"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Describe`, include the following action.

```
"Action": "autoscaling-plans:Describe*"
```

To see a list of AWS Auto Scaling actions, see Actions in the *AWS Auto Scaling API Reference*.

Resources

The `Resource` element specifies the object or objects to which the action applies.

AWS Auto Scaling has no service-defined resources that can be used as the `Resource` element of an IAM policy statement. Therefore, there are no Amazon Resource Names (ARNs) for AWS Auto Scaling for you to use in an IAM policy. To control access to AWS Auto Scaling actions, always use an * (asterisk) as the resource when writing an IAM policy.

Condition keys

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. For example, you might want a policy to be applied only after a specific date. To express conditions, use predefined condition keys.

AWS Auto Scaling does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

The `Condition` element is optional.

Examples

To view examples of AWS Auto Scaling identity-based policies, see AWS Auto Scaling identity-based policy examples (p. 24).

AWS Auto Scaling resource-based policies

Other Amazon Web Services, such as Amazon Simple Storage Service, support resource-based permissions policies. For example, you can attach a permissions policy to an S3 bucket to manage access permissions to that bucket.

AWS Auto Scaling does not support resource-based policies.

Access Control Lists (ACLs)

AWS Auto Scaling does not support Access Control Lists (ACLs).

Authorization based on AWS Auto Scaling tags

AWS Auto Scaling has no service-defined resources that can be tagged. Therefore, it does not support controlling access based on tags.

AWS Auto Scaling IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with AWS Auto Scaling

You can use temporary credentials to sign in with federation, to assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.

AWS Auto Scaling supports using temporary credentials.

Service-linked roles

AWS Auto Scaling supports service-linked roles.

Depending on which resources you add to your scaling plan, this automatically creates a service-linked role for Amazon EC2 Auto Scaling and one service-linked role per resource type for Application Auto Scaling.

If you enable predictive scaling, this automatically creates the AWS Auto Scaling service-linked role.

For more information, see the applicable role-specific documentation:

- Service-linked roles for Amazon EC2 Auto Scaling in the *Amazon EC2 Auto Scaling User Guide*
- Service-linked roles for Application Auto Scaling in the *Application Auto Scaling User Guide*
- AWS Auto Scaling service-linked roles (p. 23) in this guide

You can use the following procedure to determine if your account already has a service-linked role.

To determine whether a service-linked role already exists

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Roles**.
3. Search the list for "AWSServiceRole" to find the service linked roles that exist in your account. Look for the name of the service-linked role that you want to check.

Service roles

AWS Auto Scaling has no service roles.

AWS Auto Scaling service-linked roles

> **Important**
> To get complete information about the service-linked roles that are required to use scaling plans, see Service-linked roles (p. 22).

AWS Auto Scaling uses service-linked roles for the permissions that it requires to call other AWS on your behalf. A service-linked role is a unique type of AWS Identity and Access Management (IAM) role that is linked directly to an AWS service.

Service-linked roles provide a secure way to delegate permissions to Amazon Web Services because only the linked service can assume a service-linked role. For more information, see Using service-linked roles in the *IAM User Guide*.

The following sections describe how to create and manage the AWS Auto Scaling service-linked role. Start by configuring permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role.

Permissions granted by the service-linked role

AWS Auto Scaling uses the **AWSServiceRoleForAutoScalingPlans_EC2AutoScaling** service-linked role to manage predictive scaling of Amazon EC2 Auto Scaling groups on your behalf. This role is predefined with permissions to make the following calls on your behalf:

- `cloudwatch:GetMetricData`
- `autoscaling:DescribeAutoScalingGroups`
- `autoscaling:DescribeScheduledActions`
- `autoscaling:BatchPutScheduledUpdateGroupAction`
- `autoscaling:BatchDeleteScheduledAction`

This role trusts the `autoscaling-plans.amazonaws.com` service to assume it.

Create the service-linked role (automatic)

AWS Auto Scaling creates the **AWSServiceRoleForAutoScalingPlans_EC2AutoScaling** role for you the first time that you create a scaling plan with predictive scaling enabled.

Important

Make sure that you have enabled the IAM permissions that allow an IAM entity (such as a user, group, or role) to create the service-linked role. Otherwise, the automatic creation fails. For more information, see Service-linked role permissions in the *IAM User Guide* or the information about Permissions required to create a service-linked role (p. 28) in this guide.

Create the service-linked role (manual)

To create the service-linked role manually, you can use the IAM console, AWS CLI, or IAM API. For more information, see Creating a service-linked role in the *IAM User Guide*.

To create a service-linked role (AWS CLI)

Use the following create-service-linked-role CLI command to create the AWS Auto Scaling service-linked role.

```
aws iam create-service-linked-role --aws-service-name autoscaling-plans.amazonaws.com
```

Edit the service-linked role

With the **AWSServiceRoleForAutoScalingPlans_EC2AutoScaling** role created by AWS Auto Scaling, you can edit only its description and not its permissions. For more information, see Editing a service-linked role in the *IAM User Guide*.

Delete the service-linked role

If you no longer use scaling plans, we recommend that you delete the service-linked role. You can delete a service-linked role only after first deleting the associated dependent resources. If a service-linked role is used with multiple scaling plans, you must delete all scaling plans with predictive scaling enabled before you can delete the role. This protects your scaling plans by preventing you from inadvertently removing the permissions that you need to manage them. For more information, see Step 5: Clean up (p. 15).

You can use IAM to delete the service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

After you delete the **AWSServiceRoleForAutoScalingPlans_EC2AutoScaling** service-linked role, AWS Auto Scaling creates the role again when you create a scaling plan with predictive scaling enabled.

Supported regions for the service-linked role

AWS Auto Scaling supports using service-linked roles in all of the Regions where the service is available.

AWS Auto Scaling identity-based policy examples

By default, a brand new IAM user has no permissions to do anything. An IAM administrator must create IAM policies that grant users and roles permission to perform AWS Auto Scaling actions, such as configuring scaling policies. The administrator must then attach those policies to the IAM users or roles that require those permissions.

To learn how to create an IAM policy using these example JSON policy documents, see Creating policies on the JSON tab in the *IAM User Guide*.

If you are new to creating policies, we recommend that you first create an IAM user in your account and attach policies to the user. You can use the console to verify the effects of each policy as you attach the policy to the user.

Topics

- Policy best practices (p. 25)
- Allow users to create scaling plans (p. 25)
- Allow users to enable predictive scaling (p. 26)
- Additional required permissions (p. 26)
- Permissions required to create a service-linked role (p. 28)

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete AWS Auto Scaling resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using AWS Auto Scaling quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.

Allow users to create scaling plans

The following shows an example of a permissions policy that allows users to create scaling plans.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "autoscaling-plans:*",
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms",
              "cloudwatch:DescribeAlarms",
              "cloudformation:ListStackResources"
            ],
            "Resource": "*"
        }
    ]
}
```

For a user to work with a scaling plan, that user must have additional permissions that allow them to work with specific resources in their account. These permissions are listed in Additional required permissions (p. 26).

Each console user also needs permissions that allow them to discover the scalable resources in their account and to view graphs of CloudWatch metric data from the AWS Auto Scaling console. The additional set of permissions required to work with the AWS Auto Scaling console is listed below:

- `cloudformation:ListStacks`: To list stacks.
- `tag:GetTagKeys`: To find scalable resources that contain certain tag keys.
- `tag:GetTagValues`: To find resources that contain certain tag values.
- `autoscaling:DescribeTags`: To find Auto Scaling groups that contain certain tags.
- `cloudwatch:GetMetricData`: To view data in metric graphs.

Allow users to enable predictive scaling

The following shows an example of a permissions policy that allows users to enable predictive scaling. These permissions extend the features of scaling plans that are set up to scale Auto Scaling groups.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "cloudwatch:GetMetricData",
              "autoscaling:DescribeAutoScalingGroups",
              "autoscaling:DescribeScheduledActions",
              "autoscaling:BatchPutScheduledUpdateGroupAction",
              "autoscaling:BatchDeleteScheduledAction"
            ],
            "Resource": "*"
        }
    ]
}
```

Additional required permissions

When granting permissions for AWS Auto Scaling, you must decide which resources users get permissions for. Depending on which scenarios you want to support, you can specify the following actions in the `Action` element of an IAM policy statement.

Auto Scaling groups

To add Auto Scaling groups to a scaling plan, users must have the following permissions from Amazon EC2 Auto Scaling:

- `autoscaling:UpdateAutoScalingGroup`
- `autoscaling:DescribeAutoScalingGroups`
- `autoscaling:PutScalingPolicy`
- `autoscaling:DescribePolicies`
- `autoscaling:DeletePolicy`

ECS services

To add ECS services to a scaling plan, users must have the following permissions from Amazon ECS and Application Auto Scaling:

- `ecs:DescribeServices`

- `ecs:UpdateService`
- `application-autoscaling:RegisterScalableTarget`
- `application-autoscaling:DescribeScalableTargets`
- `application-autoscaling:DeregisterScalableTarget`
- `application-autoscaling:PutScalingPolicy`
- `application-autoscaling:DescribeScalingPolicies`
- `application-autoscaling:DeleteScalingPolicy`

Spot Fleet

To add Spot Fleets to a scaling plan, users must have the following permissions from Amazon EC2 and Application Auto Scaling:

- `ec2:DescribeSpotFleetRequests`
- `ec2:ModifySpotFleetRequest`
- `application-autoscaling:RegisterScalableTarget`
- `application-autoscaling:DescribeScalableTargets`
- `application-autoscaling:DeregisterScalableTarget`
- `application-autoscaling:PutScalingPolicy`
- `application-autoscaling:DescribeScalingPolicies`
- `application-autoscaling:DeleteScalingPolicy`

DynamoDB tables or global indexes

To add DynamoDB tables or global indexes to a scaling plan, users must have the following permissions from DynamoDB and Application Auto Scaling:

- `dynamodb:DescribeTable`
- `dynamodb:UpdateTable`
- `application-autoscaling:RegisterScalableTarget`
- `application-autoscaling:DescribeScalableTargets`
- `application-autoscaling:DeregisterScalableTarget`
- `application-autoscaling:PutScalingPolicy`
- `application-autoscaling:DescribeScalingPolicies`
- `application-autoscaling:DeleteScalingPolicy`

Aurora DB clusters

To add Aurora DB clusters to a scaling plan, users must have the following permissions from Amazon Aurora and Application Auto Scaling:

- `rds:AddTagsToResource`
- `rds:CreateDBInstance`
- `rds:DeleteDBInstance`
- `rds:DescribeDBClusters`
- `rds:DescribeDBInstances`
- `application-autoscaling:RegisterScalableTarget`

- `application-autoscaling:DescribeScalableTargets`
- `application-autoscaling:DeregisterScalableTarget`
- `application-autoscaling:PutScalingPolicy`
- `application-autoscaling:DescribeScalingPolicies`
- `application-autoscaling:DeleteScalingPolicy`

Permissions required to create a service-linked role

AWS Auto Scaling requires permissions to create a service-linked role the first time any user in your AWS account creates a scaling plan with predictive scaling enabled. If the service-linked role does not exist already, AWS Auto Scaling creates it in your account. The service-linked role grants permissions to AWS Auto Scaling so that it can call other services on your behalf.

For automatic role creation to succeed, users must have permissions for the `iam:CreateServiceLinkedRole` action.

```
"Action": "iam:CreateServiceLinkedRole"
```

The following shows an example of a permissions policy that allows a user to create an AWS Auto Scaling service-linked role.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/autoscaling-
plans.amazonaws.com/AWSServiceRoleForAutoScalingPlans_EC2AutoScaling",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName":"autoscaling-plans.amazonaws.com"
                }
            }
        }
    ]
}
```

For more information, see AWS Auto Scaling service-linked roles (p. 23).

Logging and monitoring in AWS Auto Scaling

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Auto Scaling and your other Amazon Web Services Cloud solutions. You should collect monitoring data from all parts of your Amazon Web Services solution so that you can more easily debug a multi-point failure if one occurs. AWS provides the following tools for logging and monitoring activities, and taking automatic actions when appropriate:

Amazon CloudWatch Alarms

To detect unhealthy application behavior, CloudWatch helps you by automatically monitoring certain metrics for your Amazon Web Services resources. You can configure a CloudWatch alarm and set up an Amazon SNS notification that sends an email when a metric's value is not what you expect or when certain anomalies are detected. For example, you can be notified when network activity is suddenly higher or lower than a metric's expected value. For more information, see the *Amazon CloudWatch User Guide*.

AWS CloudTrail

AWS CloudTrail captures API calls and related events made by or on behalf of your Amazon Web Services account. Then it delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the *AWS CloudTrail User Guide*. For information about the AWS Auto Scaling API calls that are logged by CloudTrail, see Logging AWS Auto Scaling API calls with CloudTrail.

Amazon CloudWatch Logs

Amazon CloudWatch Logs enable you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the *Amazon CloudWatch Logs User Guide*.

Amazon EventBridge

EventBridge delivers a near real time stream of system events that describe changes in Amazon Web Services resources. AWS Auto Scaling currently does not emit events. However, you can write rules that trigger automated actions in other Amazon Web Services as a result of API calls made by AWS Auto Scaling. For more information, see Creating an EventBridge rule that triggers on an AWS API call using AWS CloudTrail in the *Amazon EventBridge User Guide*.

Related topics

- Monitoring your Auto Scaling instances and groups in the *Amazon EC2 Auto Scaling User Guide*
- Application Auto Scaling monitoring in the *Application Auto Scaling User Guide*

Compliance validation for AWS Auto Scaling

The security and compliance of Amazon Web Services (AWS) services is assessed by third-party auditors as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS Artifact.

Your compliance responsibility when using AWS Auto Scaling is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and compliance quick start guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA security and compliance whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS compliance resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating resources with rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.

- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Resilience in AWS Auto Scaling

The AWS global infrastructure is built around AWS Regions and Availability Zones.

AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking.

With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in AWS Auto Scaling

As a managed service, AWS Auto Scaling is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of security processes whitepaper.

You use AWS published API calls to access AWS Auto Scaling through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

AWS Auto Scaling and interface VPC endpoints

You can establish a private connection between your virtual private cloud (VPC) and the AWS Auto Scaling API by creating an interface VPC endpoint. You can use this connection to call the AWS Auto Scaling API from your VPC without sending traffic over the internet. The endpoint provides reliable, scalable connectivity to the AWS Auto Scaling API. It does this without requiring an internet gateway, NAT instance, or VPN connection.

Interface VPC endpoints are powered by AWS PrivateLink, a feature that enables private communication between Amazon Web Services using private IP addresses. For more information, see AWS PrivateLink.

Create an interface VPC endpoint

You can create a VPC endpoint for the AWS Auto Scaling service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). Create an endpoint for AWS Auto Scaling using the following service name:

- **com.amazonaws.*`region`*.autoscaling-plans** — Creates an endpoint for the AWS Auto Scaling API operations.

For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Enable private DNS for the endpoint to make API requests to the supported service using its default DNS hostname (for example, `autoscaling-plans.us-east-1.amazonaws.com`). When creating an endpoint for Amazon Web Services, this setting is enabled by default. For more information, see Accessing a service through an interface endpoint in the *Amazon VPC User Guide*.

You do not need to change any AWS Auto Scaling settings. AWS Auto Scaling calls other Amazon Web Services using either service endpoints or private interface VPC endpoints, whichever are in use.

Create a VPC endpoint policy

You can attach a policy to your VPC endpoint to control access to the AWS Auto Scaling API. The policy specifies:

- The principal that can perform actions.
- The actions that can be performed.
- The resource on which the actions can be performed.

The following example shows a VPC endpoint policy that denies everyone permission to delete a scaling plan through the endpoint. The example policy also grants everyone permission to perform all other actions.

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "autoscaling-plans:DeleteScalingPlan",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

For more information, see Using VPC endpoint policies in the *Amazon VPC User Guide*.

Endpoint migration

On November 22, 2019, AWS Auto Scaling introduced `autoscaling-plans.region.amazonaws.com` as the new default DNS hostname and endpoint for calls to the AWS Auto Scaling API. The new endpoint is compatible with the latest release of the AWS CLI and SDKs. If you have not done so already, install the latest AWS CLI and SDKs to use the new endpoint. To update the AWS CLI, see Installing the AWS CLI using pip in the *AWS Command Line Interface User Guide*. For information about the AWS SDKs, see Tools for Amazon Web Services.

> **Important**
> For backward compatibility, the existing `autoscaling.region.amazonaws.com` endpoint will continue to be supported for calls to the AWS Auto Scaling API. To set up the `autoscaling.region.amazonaws.com` endpoint as a private interface VPC endpoint, see Amazon EC2 Auto Scaling and interface VPC endpoints in the *Amazon EC2 Auto Scaling User Guide*.

Endpoint to Call When Using the CLI or the AWS Auto Scaling API

For the current release of AWS Auto Scaling, your calls to the AWS Auto Scaling API automatically go to the `autoscaling-plans.region.amazonaws.com` endpoint instead of `autoscaling.region.amazonaws.com`.

You can call the new endpoint in the CLI by using the following parameter with each command to specify the endpoint: `--endpoint-url https://autoscaling-plans.region.amazonaws.com`.

Although it is not recommended, you can also call the old endpoint in the CLI by using the following parameter with each command to specify the endpoint: `--endpoint-url https://autoscaling.region.amazonaws.com`.

For the various SDKs used to call the APIs, see the documentation for the SDK of interest to learn how to direct the requests to a specific endpoint. For more information, see Tools for Amazon Web Services.

AWS Auto Scaling service quotas

Your Amazon Web Services account has the following default quotas, formerly referred to as limits, for AWS Auto Scaling.

To request an increase, use the Auto Scaling limits form. Make sure that you specify the type of resource with your request for an increase, for example, Amazon EC2 Auto Scaling, Amazon ECS, or DynamoDB.

Default quotas per Region per account

Item	Default
Maximum number of scalable resources per resource type	Quotas vary depending on resource type. Amazon DynamoDB: 3000 Amazon EC2 Auto Scaling groups: 200 All other resource types: 500
Maximum number of scaling plans	100
Maximum number of scaling instructions per scaling plan	500
Maximum number of target tracking configurations per scaling instruction	10

Keep service quotas in mind as you scale out your workloads. For example, when you reach the maximum number of capacity units allowed by a service, scaling out will stop. If demand drops and the current capacity decreases, AWS Auto Scaling can scale out again. To avoid reaching this service quota limit again, you can request an increase. Each service has its own default quotas for the maximum capacity of the resource. For information about the default quotas for other Amazon Web Services, see Service endpoints and quotas in the *Amazon Web Services General Reference*.

AWS Auto Scaling resources

The following related resources can help you as you work with this service.

- **AWS Auto Scaling** – The primary web page for information about AWS Auto Scaling.
- **AWS Auto Scaling FAQ** – The answers to questions customers ask about AWS Auto Scaling.
- **AWS Auto Scaling discussion forum** – Get help from the community.
- **Tagging Auto Scaling groups and instances** – Get information about tagging your Auto Scaling groups.
- **Tagging for DynamoDB** – Get information about tagging your Amazon DynamoDB tables or global secondary indexes.
- **Tagging Amazon RDS resources** – Get information about tagging your Aurora DB clusters.
- **Working with Tag Editor** – Get information about using Tag Editor, including which resources Tag Editor supports.
- **Target tracking scaling policies** for Amazon EC2 Auto Scaling – Get information about target tracking scaling policies for Amazon EC2 Auto Scaling groups.
- **Target tracking scaling policies** for all other resources – Get information about target tracking scaling policies for resources beyond Amazon EC2, such as DynamoDB indexes and tables and Amazon ECS services.
- **AWS Auto Scaling API and CLI reference guides** – Documentation for the API calls and the AWS CLI commands that you can use to create, modify, and delete Auto Scaling plans.
- **Logging API calls with CloudTrail** – Get information about monitoring calls made to the API for your account, including calls made by the AWS Management Console, command line tools, and other services.

The following additional resources are available to help you learn more about Amazon Web Services.

- **Classes & Workshops** – Links to role-based and specialty courses, in addition to self-paced labs to help sharpen your AWS skills and gain practical experience.
- **AWS Developer Tools** – Links to developer tools, SDKs, IDE toolkits, and command line tools for developing and managing AWS applications.
- **AWS Whitepapers** – Links to a comprehensive list of technical AWS whitepapers, covering topics such as architecture, security, and economics and authored by AWS Solutions Architects or other technical experts.
- **AWS Support Center** – The hub for creating and managing your AWS Support cases. Also includes links to other helpful resources, such as forums, technical FAQs, service health status, and AWS Trusted Advisor.
- **AWS Support** – The primary webpage for information about AWS Support, a one-on-one, fast-response support channel to help you build and run applications in the cloud.
- **Contact Us** – A central contact point for inquiries concerning AWS billing, account, events, abuse, and other issues.
- **AWS Site Terms** – Detailed information about our copyright and trademark; your account, license, and site access; and other topics.

Document history

The following table describes important additions to the AWS Auto Scaling documentation. For notification about updates to this documentation, you can subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
New "Security" chapter (p. 35)	A new Security chapter in the *AWS Auto Scaling User Guide* helps you understand how to apply the shared responsibility model when using AWS Auto Scaling. As part of this update, the user guide chapter "Authentication and Access Control" has been replaced by a new, more useful section, Identity and access management for AWS Auto Scaling.	March 12, 2020
Support for Amazon VPC endpoints (p. 35)	You can now establish a private connection between your VPC and AWS Auto Scaling. For migration considerations and instructions, see AWS Auto Scaling and interface VPC endpoints.	November 22, 2019
Support for increasing maximum capacity above forecast capacity, plus guide changes (p. 35)	Adds console support for allowing the scaling plan to increase maximum capacity above forecast capacity by a specified buffer value. For more information, see Predictive scaling settings in the *AWS Auto Scaling User Guide*. This release also includes several rewritten sections in the Getting started with AWS Auto Scaling tutorial.	March 9, 2019
Predictive scaling and enhancements (p. 35)	You can now use predictive scaling to proactively scale your Amazon EC2 Auto Scaling groups. This release also adds support for replacing scaling policies created outside of the scaling plan (such as from other consoles) and controlling whether you enable your plan's dynamic scaling feature. For more information, see Getting started with AWS Auto Scaling.	November 20, 2018

Support for custom resource settings (p. 35)	Added support for customizing various settings for each individual resource or multiple resources at the same time. For more information, see Getting started with AWS Auto Scaling.	October 9, 2018
Tags as an application source (p. 35)	This release adds support for specifying a set of tags as an application source.	April 23, 2018
New service (p. 35)	Initial release of AWS Auto Scaling.	January 16, 2018