AWS Auto Scaling

User Guide

AWS Auto Scaling: User Guide

The AWS Documentation website is getting a new look!

Try it now and let us know what you think. Switch to the new look >>

You can return to the original look by selecting English in the language selector above.

Table of Contents

What Is AWS Auto Scaling?

AWS Auto Scaling enables you to configure automatic scaling for the AWS 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 automatic scaling features of multiple AWS services. 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 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 AWS Auto Scaling Works (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.
- AWS Regions and Endpoints in the *AWS General Reference*—This page shows you the regional availability of AWS Auto Scaling and other AWS services.
- 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 AWS resources. You can use AWS CloudFormation sample templates or create your own templates to create the AWS resources, and any associated dependencies or runtime parameters, required to run your application. You can also create templates of scaling plans using AWS CloudFormation.

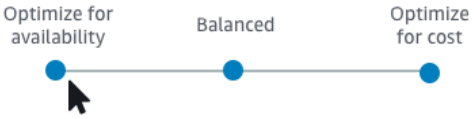
Amazon CloudWatch is a monitoring service for AWS Cloud resources and the applications you run on AWS. CloudWatch lets you collect and track metrics, log files, and automatically react to changes in your applications using alarms. You can also publish your own custom metrics to CloudWatch using the AWS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

How AWS Auto Scaling Works

Scaling plans are a set of instructions for scaling your resources. You create one scaling plan per application source, such as an AWS CloudFormation stack, a set of tags, or one or more Amazon EC2 Auto Scaling groups. Your scaling plan blends dynamic scaling and predictive scaling methods together to support your scaling strategy.

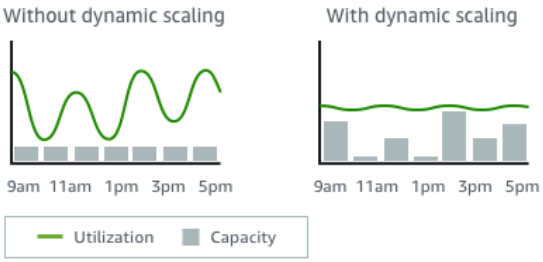
What is a scaling strategy?

The scaling strategy tells AWS Auto Scaling how to optimize the utilization of the 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?

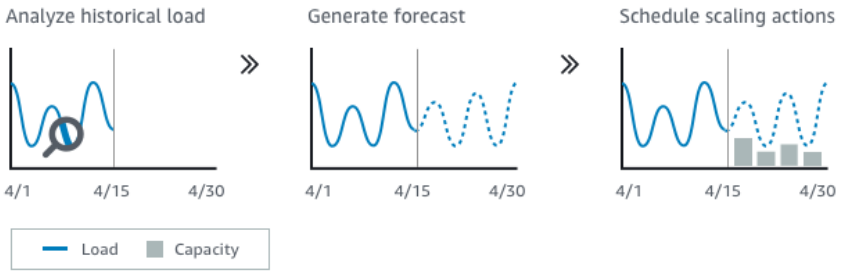
Dynamic scaling creates target tracking scaling policies for the resources in your scaling plan. These scaling policies adjust resource capacity in response to live changes in resource utilization. The intention is to provide enough capacity to maintain utilization at the target value specified by the 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, similar to how weather forecasts works. Using the forecast, it 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 the 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 use the AWS Management Console to create your first scaling plan, and 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 AWS Cloud. Take note of the following:

- How long it takes to launch and configure a server.
- Whether you have existing scaling policies created from other consoles.
- What metrics have the most relevance to your application's performance.
- The target utilization that makes sense for each scalable resource in your application based on the resource as a whole, for example, the full Amazon EC2 Auto Scaling group instead of individual instances.
- Whether the metric history is sufficiently long to use with predictive scaling (if using newly created Amazon EC2 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.

Tasks

Best Practices for AWS Auto Scaling

The following best practices can help you make the most of AWS Auto Scaling:

- 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*.
- 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.

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.
- You can create scaling policies from various AWS consoles, but AWS Auto Scaling does not overwrite these other scaling policies or create new ones by default. You can optionally replace the existing scaling policies with target tracking scaling policies created from the AWS Auto Scaling console by enabling the **Replace external scaling policies** setting in the scaling plan. For more information, see Dynamic Scaling Settings (p. 11).
- 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.

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. The Getting Started walkthrough focuses on the

most straightforward configuration for a scaling plan. It also describes the Create Scaling Plan wizard and all of the ways that you can use it to configure a scaling plan.

The first step in the wizard asks you to find your scalable resources. There are two ways to locate the resources for a new scaling plan:

- You specify an application source (an AWS CloudFormation stack or a set of tags) for AWS Auto Scaling to use to automatically discover your scalable resources. As you define your scaling plan, you can then choose which of these resources to include or exclude.
- Alternatively, you can choose one or more Auto Scaling groups of Amazon EC2 instances to use in your scaling plan.

For your scalable resources from multiple services to be discoverable, you must have an AWS CloudFormation stack or a set of tags. When you use a CloudFormation stack, 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, AWS Auto Scaling needs to 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 service name must also be unique across each of these ECS clusters.

Tags can be assigned in a number of ways. Use the console for each individual service by accessing the **Tags** tab on the relevant resource screen, or use the Tag Editor. Currently, ECS services and Spot Fleet requests cannot be discovered using tags.

> **Important**
> For a beginner-friendly tutorial, you can start more simply by choosing an Auto Scaling group to add to your scaling plan. 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 advanced features that are available in your scaling plan.

Prerequisites

Before you begin, use the Amazon EC2 console to create a new Auto Scaling group by following the steps for Creating an Auto Scaling Group Using a Launch Template in the *Amazon EC2 Auto Scaling User Guide*. You can choose any group, but let's use a new group for this tutorial, so that you can delete it afterwards. As soon as the group is deleted, you stop incurring charges for the Amazon EC2 instances it ran.

You can optionally configure your Auto Scaling group to ensure it performs optimally by following these guidelines:

- Enable detailed monitoring to get metric data for individual 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 at a 1-minute frequency. 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, configure your instances as `unlimited` so that they can sustain high CPU performance as required in this tutorial. 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*.

You need IAM permissions to create a scaling plan and enable predictive scaling. For information about the required IAM permissions, see the Example Policies (p. 17) in this guide.

Discovering or Choosing Your Scalable Resources

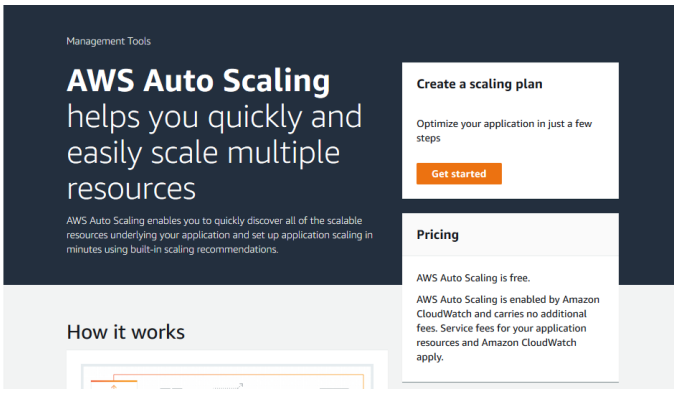
Complete the following procedure to find your scalable resources.

> **Note**
> For this tutorial, choose **Choose EC2 Auto Scaling groups** and then choose the Auto Scaling group that you created in the previous section.

To find scalable resources to add to your 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 scalable resources.

   > **Note**
   > You can choose any Region that supports AWS Auto Scaling. For a list of available Regions, see the AWS Regions and Endpoints documentation in the *AWS General Reference*.
3. From the welcome page, choose **Get started**.



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

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 AWS Auto Scaling Works (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**, type 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**.

   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 customized scaling metric to use instead of the scaling 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 customized load metric to use instead of the load 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.

Customized Metrics Specification

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 customized metric. Before you follow these instructions, we recommend that you become familiar with the Amazon CloudWatch User Guide.

To specify a customized 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. Cooldown periods are the amount of time after a scale-out or scale-in activity completes before another activity can start. The intention is to give newly provisioned resources time to start handling demand before triggering a new scaling action. This setting is not available if the resource is an Auto Scaling group. For more information, see Cooldown Period in the *Application Auto Scaling User Guide*.
- **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 run time 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.

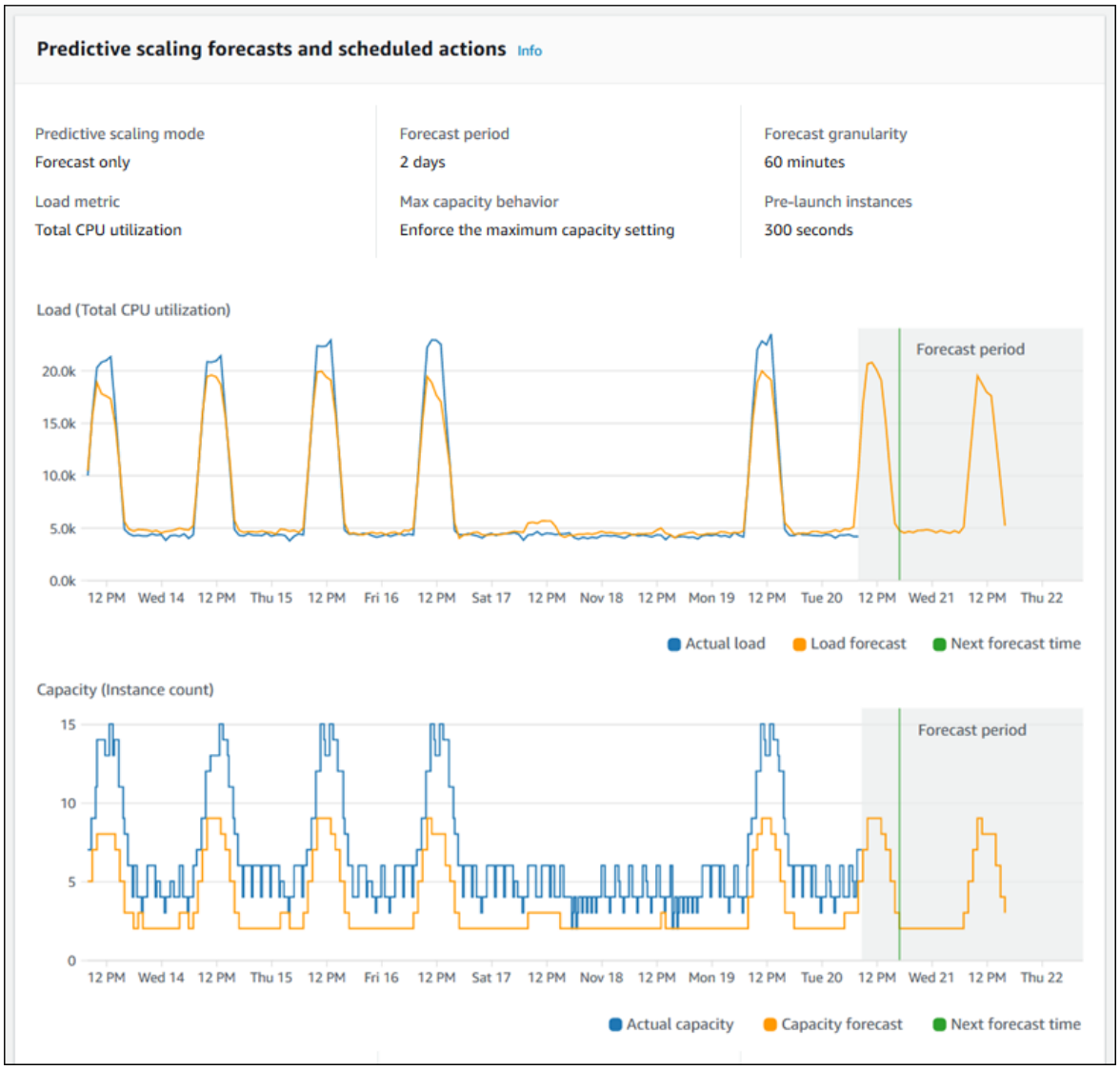
To view scaling information for a resource

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.



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*.

Authentication and Access Control for AWS Auto Scaling

Access to AWS Auto Scaling requires credentials that AWS can use to authenticate your requests. Those credentials must have permissions to perform AWS Auto Scaling actions, such as creating scaling plans.

This topic provides details on how you can use AWS Identity and Access Management (IAM) to help secure your resources by controlling who can perform AWS Auto Scaling actions.

By default, a brand new IAM user has no permissions to do anything. To grant permissions to call AWS Auto Scaling actions, you attach an IAM policy to the IAM users or groups that require the permissions it grants.

Specifying Actions in a Policy

You can specify any and all AWS Auto Scaling actions in an IAM policy. For more information, see Actions in the *AWS Auto Scaling API Reference*.

To specify a single policy, you can use the following prefix with the name of the action: `autoscaling-plans:`. For example:

```
"Action": "autoscaling-plans:DescribeScalingPlans"
```

Wildcards are supported. For example, you can use `autoscaling-plans:*` to specify all AWS Auto Scaling actions.

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

You can also use `Describe*` to specify all actions whose names start with `Describe`.

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

In addition to the permissions for calling AWS Auto Scaling actions, users also require permissions to create a service-linked role.

When users create a scaling plan with predictive scaling enabled, AWS Auto Scaling creates a service-linked role in your account, if the role does not exist already. 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"
```

For more information, see Service-Linked Roles for AWS Auto Scaling (p. 19).

Specifying the Resource

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 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.

Specifying Conditions in a Policy

When you grant permissions, you can use IAM policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. To express conditions, use predefined condition keys.

For a list of condition keys supported by each AWS service, see Actions, Resources, and Condition Keys for AWS Services in the *IAM User Guide*. For a list of condition keys that can be used in multiple AWS services, see AWS Global Condition Context Keys in the *IAM User Guide*.

AWS Auto Scaling does not provide additional condition keys.

Example Policies

To create a scaling plan, users must have permission to use the actions in the following example policy.

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

To configure predictive scaling for Auto Scaling groups, users must also have permission to use the actions in the following example policy.

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

```
            "Resource": "*"
        }
    ]
}
```

Additional IAM Permissions

Users must have additional permissions for each type of resource they must add to a scaling plan. You specify the following actions in the `Action` element of an IAM policy statement.

Auto Scaling groups

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

Auto Scaling groups that use predictive scaling

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

Resource types other than Auto Scaling groups

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

ECS services

- `ecs:DescribeServices`
- `ecs:UpdateServices`

Spot Fleet requests

- `ec2:DescribeSpotFleetRequests`
- `ec2:ModifySpotFleetRequest`

DynamoDB tables or global indexes

- `dynamodb:DescribeTable`
- `dynamodb:UpdateTable`

Aurora DB clusters

- `rds:AddTagsToResource`
- `rds:CreateDBInstance`
- `rds:DeleteDBInstance`
- `rds:DescribeDBClusters`
- `rds:DescribeDBInstances`

Service-Linked Roles for AWS Auto Scaling

AWS Auto Scaling uses service-linked roles for the permissions that it requires to call other AWS services 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 AWS 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*.

> **Note**
> For information about the service-linked roles created by Amazon EC2 Auto Scaling and Application Auto Scaling, see Service-Linked Roles in the *Amazon EC2 Auto Scaling User Guide* and Service-Linked Roles in the *Application Auto Scaling User Guide*.

Service-Linked Role Permissions for AWS Auto Scaling

AWS Auto Scaling uses the following service-linked role to manage predictive scaling of Amazon EC2 Auto Scaling groups on your behalf.

The `AWSServiceRoleForAutoScalingPlans_EC2AutoScaling` 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.amazonaws.com` service to assume it.

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Using Service-Linked Roles in the *IAM User Guide*.

Create Service-Linked Roles (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 required user permissions (p. 16) in this guide.

Edit the Service-Linked Roles

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 Roles

If you no longer use AWS Auto Scaling, we recommend that you delete the service-linked role. You can delete a service-linked role only after first deleting the related AWS 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 because you cannot inadvertently remove permissions 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 AWS Auto Scaling Service-Linked Roles

AWS Auto Scaling supports using service-linked roles in all of the regions where the service is available. For more information, see AWS Regions and Endpoints.

AWS Auto Scaling Limits

Your AWS account has the following limits related to AWS Auto Scaling. To request a limit increase, use the Auto Scaling Limits form.

Default Limits Per Region Per Account

Item	Default Limit	Notes
Maximum number of scalable resources per resource type	Amazon DynamoDB: 2000 Amazon EC2 Auto Scaling groups: 200 All other resource types: 500	Make sure that you specify the type of resource with your request for a limit increase, for example, Amazon EC2 Auto Scaling, Amazon ECS, or DynamoDB.
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	

For more information on the service limits for other AWS services, see AWS Service Limits 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 AWS.

- **Classes & Workshops** – Links to role-based and specialty courses as well as 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 web page 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
Support for increasing maximum capacity above forecast capacity, plus guide changes (p. 23)	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. 23)	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. 23)	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. 23)	This release adds support for specifying a set of tags as an application source.	April 23, 2018
New service (p. 23)	Initial release of AWS Auto Scaling.	January 16, 2018