# 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 traffic 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](https://docs.aws.amazon.com/autoscaling/plans/userguide/how-it-works.html)—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](https://aws.amazon.com/autoscaling/faqs/)—The FAQ on the product page provides information about the benefits of this service.
* [AWS Region Table](https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/)—This page shows you the regional availability of AWS Auto Scaling.
* [Amazon EC2 Auto Scaling User Guide](https://docs.aws.amazon.com/autoscaling/ec2/userguide/)—This guide shows you how to create and manage the Auto Scaling groups to use when scaling your fleet of Amazon EC2 instances.

# Getting Started with Amazon EC2 Auto Scaling

When you use Amazon EC2 Auto Scaling, you must use certain building blocks to get started. This tutorial walks you through the process for setting up the basic infrastructure for Amazon EC2 Auto Scaling.

Before you create an Auto Scaling group 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.
* What metrics have the most relevance to your application's performance.
* How many Availability Zones the Auto Scaling group should span.
* What existing resources can be used, such as EC2 instances or Amazon Machine Images (AMIs).
* Do you want to scale to increase or decrease capacity, or do you just want to ensure that a specific number of servers are always running? Keep in mind that Amazon EC2 Auto Scaling can do both simultaneously.

The better you understand your application, the more effective you can make your Auto Scaling architecture.

The following instructions are for a configuration template that defines your EC2 instances, creates an Auto Scaling group to maintain the healthy number of instances, and optionally deletes this basic infrastructure. This tutorial assumes that you are familiar with launching EC2 instances and have already created a key pair and a security group.

**Tasks**

* [Step 1: Create a Launch Template](https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html#gs-create-lt)
* [Step 2: Create an Auto Scaling Group](https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html#gs-create-asg)
* [Step 3: Verify Your Auto Scaling Group](https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html#gs-verify-asg)
* [Step 4: (Optional) Delete Your Scaling Infrastructure](https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html#gs-delete-asg)

## Step 1: Create a Launch Template

For this step, you create a launch template that specifies the type of EC2 instance that Amazon EC2 Auto Scaling creates for you. Include information such as the ID of the Amazon Machine Image (AMI) to use for launching the EC2 instance, the instance type, key pairs, security groups, and block device mappings.

**To create a launch template for an Auto Scaling group**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar, select a region. The Amazon EC2 Auto Scaling resources that you create are tied to the region you specify and are not replicated across regions. For more information, see [Example: Distributing Instances Across Availability Zones](https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-benefits.html#arch-AutoScalingMultiAZ).
3. On the navigation pane, choose **Instances**, **Launch Templates**.
4. Choose **Create launch template**.
5. Choose **Create a new template**. Provide a name and description for the launch template.
6. For **AMI ID**, choose an AMI ID from which to launch an instance. The Amazon Machine Image (AMI) serves as templates for your instances.
7. For **Instance type**, choose a hardware configuration that is compatible with the AMI that you specified.
8. (Optional) For **Key pair name**, type the name of the key pair to use when connecting to your instances.
9. (Optional) For **Network type**, choose whether to launch your instances into a **VPC** or**EC2-Classic**, if applicable.

**Important**

When you create an Auto Scaling group, the following launch template settings are ignored in favor of the Auto Scaling group settings: network type, subnet, and Availability Zone.

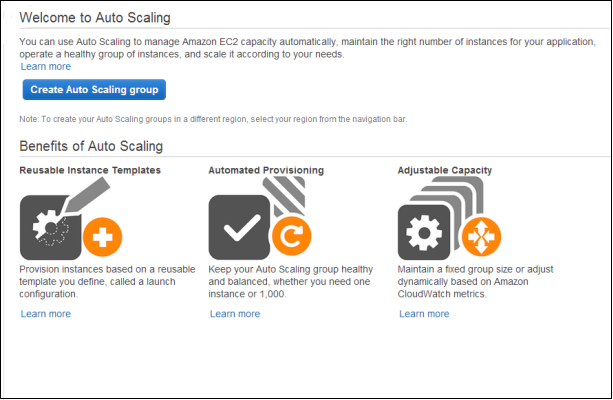
1. Skip **Security Groups** to configure a security group as part of the network interface. You cannot specify security groups in both places.
2. For **Network interfaces**, configure **Auto-assign public IP** and **Security group ID**. To launch instances into a VPC, you must specify a security group that is created for that VPC. You can leave the other fields empty and this creates a primary network interface with IP addresses that we select for your instances.
3. (Optional) For **Storage (Volumes)**, specify volumes to attach to the instances in addition to the volumes specified by the AMI you specified.
4. (Optional) For **Tags**, specify one or more tags to associate with the instances and volumes.
5. Choose **Create launch template**.
6. On the confirmation page, choose **Create Auto Scaling group**.

If you are not currently using launch templates, you can create a launch configuration instead.

A launch configuration is similar to a launch template, in that it specifies the type of EC2 instance that Amazon EC2 Auto Scaling creates for you. Create the launch configuration by including information such as the ID of the Amazon Machine Image (AMI) to use, the instance type, the key pair, security groups, and block device mapping.

**To create a launch configuration**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar, select a Region. The Auto Scaling resources that you create are tied to the Region that you specify and are not replicated across Regions. For more information, see [Example: Distributing Instances Across Availability Zones](https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-benefits.html#arch-AutoScalingMultiAZ).
3. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
4. On the **Welcome to Auto Scaling** page, choose **Create Auto Scaling group**.



1. On the **Create Auto Scaling Group** page, choose **Launch Configuration, Create a new launch configuration**, and then choose **Next Step**.
2. For the **Choose AMI** step, there is a list of basic configurations, called Amazon Machine Images (AMIs), that serve as templates for your instances. Choose **Select** for the Amazon Linux AMI.
3. For the **Choose Instance Type** step, select a hardware configuration for your instances. We recommend that you keep the default, a t2.micro instance. Choose **Next: Configure details**.

**Note**

T2 instances must be launched into a subnet of a VPC. If you select a t2.micro instance but don't have a VPC, one is created for you. This VPC includes a public subnet in each Availability Zone in the region.

1. For the **Configure details** step, do the following:
   1. For **Name**, type a name for your launch configuration (for example, my-first-lc).
   2. For **Advanced Details**, select an IP address type. If you want to connect to an instance in a VPC, you must select an option that assigns a public IP address. If you want to connect to your instance but aren't sure whether you have a default VPC, select **Assign a public IP address to every instance**.
   3. Choose **Skip to review**.
2. For the **Review** step, choose **Edit security groups**. Follow the instructions to choose an existing security group, and then choose **Review**.
3. For the **Review** step, choose **Create launch configuration**.
4. Complete the **Select an existing key pair or create a new key pair** step as instructed. You won't connect to your instance as part of this tutorial. Therefore, you can select **Proceed without a key pair** unless you intend to connect to your instance.
5. Choose **Create launch configuration**. The launch configuration is created and the wizard to create an Auto Scaling group is displayed.

## Step 2: Create an Auto Scaling Group

An Auto Scaling group is a collection of EC2 instances, and the core of Amazon EC2 Auto Scaling. When you create an Auto Scaling group, you include information such as the subnets for the instances and the initial number of instances to start with.

Use the following procedure to continue where you left off after creating the launch template.

**To create an Auto Scaling group**

1. For the **Configure Auto Scaling group details** step, do the following:
   1. For **Group name**, type a name for your Auto Scaling group (for example, my-first-asg).
   2. For **Launch template version**, choose whether the Auto Scaling group uses the default, the latest, or a specific version of the launch template when scaling out.
   3. For **Fleet Composition**, choose **Adhere to the launch template**.
   4. Keep **Group size** set to the default value of 1 instance for this tutorial.
   5. Keep **Network** set to the default VPC for the region, or select your own VPC.
   6. For **Subnet**, choose a subnet for the VPC.
   7. Choose **Next: Configure scaling policies**.
2. On the **Configure scaling policies** page, select **Keep this group at its initial size** and choose **Review**.
3. On the **Review** page, choose **Create Auto Scaling group**.
4. On the **Auto Scaling group creation status** page, choose **Close**.

## Step 3: Verify Your Auto Scaling Group

Now that you have created your Auto Scaling group, you are ready to verify that the group has launched an EC2 instance.

**To verify that your Auto Scaling group has launched an EC2 instance**

1. On the **Auto Scaling Groups** page, select the Auto Scaling group that you just created.
2. The **Details** tab provides information about the Auto Scaling group.


                        Auto Scaling group details
                    

1. On the **Activity History** tab, the **Status** column shows the current status of your instance. While your instance is launching, the status column shows In progress. The status changes to Successful after the instance is launched. You can also use the refresh button to see the current status of your instance.
2. On the **Instances** tab, the **Lifecycle** column shows the state of your instance. You can see that your Auto Scaling group has launched your EC2 instance, and that it is in the InService lifecycle state. The **Health Status** column shows the result of the EC2 instance health check on your instance.


                        Auto Scaling group instances
                    

1. (Optional) If you want, you can try the following experiment to learn more about Amazon EC2 Auto Scaling. The minimum size for your Auto Scaling group is one instance. Therefore, if you terminate the running instance, Amazon EC2 Auto Scaling must launch a new instance to replace it.
   1. On the **Instances** tab, select the ID of the instance. This shows you the instance on the **Instances** page.
   2. Choose **Actions**, **Instance State**, **Terminate**. When prompted for confirmation, choose **Yes, Terminate**.
   3. On the navigation pane, choose **Auto Scaling Groups**. Select your Auto Scaling group and choose the **Activity History** tab. The default cooldown for the Auto Scaling group is 300 seconds (5 minutes), so it takes about 5 minutes until you see the scaling activity. When the scaling activity starts, you see an entry for the termination of the first instance and an entry for the launch of a new instance. The **Instances** tab shows the new instance only.
   4. On the navigation pane, choose **Instances**. This page shows both the terminated instance and the running instance.

Go to the next step if you would like to delete your basic infrastructure for automatic scaling. Otherwise, you can use this infrastructure as your base and try one or more of the following:

* [Maintaining the Number of Instances in Your Auto Scaling Group](https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-maintain-instance-levels.html)
* [Manual Scaling](https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-manual-scaling.html)
* [Dynamic Scaling for Amazon EC2 Auto Scaling](https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html)
* [Getting Amazon SNS Notifications When Your Auto Scaling Group Scales](https://docs.aws.amazon.com/autoscaling/ec2/userguide/ASGettingNotifications.html)

## Step 4: (Optional) Delete Your Scaling Infrastructure

You can either delete your scaling infrastructure or delete just your Auto Scaling group and keep your launch template to use later.

**To delete your Auto Scaling group**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. Select your Auto Scaling group (for example, **my-first-asg**).
4. Choose **Actions**, **Delete**. When prompted for confirmation, choose **Yes, Delete**.

The **Name** column indicates that the Auto Scaling group is being deleted. The **Desired**, **Min**, and **Max** columns show 0 instances for the Auto Scaling group.

Skip this procedure if you would like to keep your launch template.

**To delete your launch template**

1. On the navigation pane, choose **Instances**, **Launch Templates**.
2. Select your launch template (for example, **my-first-lt**).
3. Choose **Actions**, **Delete template**. When prompted for confirmation, choose **Delete launch template**.

Skip this procedure if you would like to keep your launch configuration.

**To delete your launch configuration**

1. On the navigation pane, under **Auto Scaling**, choose **Launch Configurations**.
2. Select your launch configuration (for example, my-first-lc).
3. Choose **Actions**, **Delete launch configuration**. When prompted for confirmation, choose **Yes, Delete**.

# Autoscaling-plans

## Description

Use AWS Auto Scaling to quickly discover all the scalable AWS resources for your application and configure dynamic scaling and predictive scaling for your resources using scaling plans. Use this service in conjunction with the Amazon EC2 Auto Scaling, Application Auto Scaling, Amazon CloudWatch, and AWS CloudFormation services.

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

For more information about AWS Auto Scaling, see the [AWS Auto Scaling User Guide](http://docs.aws.amazon.com/autoscaling/plans/userguide/what-is-aws-auto-scaling.html) .

## Available Commands

* [create-scaling-plan](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/create-scaling-plan.html)
* [delete-scaling-plan](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/delete-scaling-plan.html)
* [describe-scaling-plan-resources](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/describe-scaling-plan-resources.html)
* [describe-scaling-plans](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/describe-scaling-plans.html)
* [get-scaling-plan-resource-forecast-data](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/get-scaling-plan-resource-forecast-data.html)
* [update-scaling-plan](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/update-scaling-plan.html)

# Create-scaling-plan

## Description

Creates a scaling plan.

See also: [AWS API Documentation](https://docs.aws.amazon.com/goto/WebAPI/autoscaling-plans-2018-01-06/CreateScalingPlan)

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Synopsis

create-scaling-plan

--scaling-plan-name <value>

--application-source <value>

--scaling-instructions <value>

[--cli-input-json <value>]

[--generate-cli-skeleton <value>]

--cli-input-json (string) Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values. It is not possible to pass arbitrary binary values using a JSON-provided value as the string will be taken literally.

--generate-cli-skeleton (string) Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for --cli-input-json. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command.

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Output

ScalingPlanVersion -> (long)

The version number of the scaling plan. This value is always 1.

Currently, you cannot specify multiple scaling plan versions.

# Delete-scaling-plan

## Description

Deletes the specified scaling plan.

Deleting a scaling plan deletes the underlying ScalingInstruction for all of the scalable resources that are covered by the plan.

If the plan has launched resources or has scaling activities in progress, you must delete those resources separately.

See also: [AWS API Documentation](https://docs.aws.amazon.com/goto/WebAPI/autoscaling-plans-2018-01-06/DeleteScalingPlan)

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Synopsis

delete-scaling-plan

--scaling-plan-name <value>

--scaling-plan-version <value>

[--cli-input-json <value>]

[--generate-cli-skeleton <value>]

## Options

--scaling-plan-name (string)

The name of the scaling plan.

--scaling-plan-version (long)

The version number of the scaling plan.

--cli-input-json (string) Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values. It is not possible to pass arbitrary binary values using a JSON-provided value as the string will be taken literally.

--generate-cli-skeleton (string) Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for --cli-input-json. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command.

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Output[¶](https://docs.aws.amazon.com/cli/latest/reference/autoscaling-plans/delete-scaling-plan.html#output)

# Describe-scaling-plan-resources

## Description

Describes the scalable resources in the specified scaling plan.

See also: [AWS API Documentation](https://docs.aws.amazon.com/goto/WebAPI/autoscaling-plans-2018-01-06/DescribeScalingPlanResources)

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Synopsis

describe-scaling-plans

[--scaling-plan-names <value>]

[--scaling-plan-version <value>]

[--application-sources <value>]

[--max-results <value>]

[--next-token <value>]

[--cli-input-json <value>]

[--generate-cli-skeleton <value>]

## Options

--scaling-plan-names (list)

The names of the scaling plans (up to 10). If you specify application sources, you cannot specify scaling plan names.

Syntax:

"string" "string" ...

--scaling-plan-version (long)

The version number of the scaling plan. If you specify a scaling plan version, you must also specify a scaling plan name.

--application-sources (list)

The sources for the applications (up to 10). If you specify scaling plan names, you cannot specify application sources.

JSON Syntax:

[

{

"CloudFormationStackARN": "string",

"TagFilters": [

{

"Key": "string",

"Values": ["string", ...]

}

...

]

}

...

]

--max-results (integer)

The maximum number of scalable resources to return. This value can be between 1 and 50. The default value is 50.

--next-token (string)

The token for the next set of results.

--cli-input-json (string) Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values. It is not possible to pass arbitrary binary values using a JSON-provided value as the string will be taken literally.

--generate-cli-skeleton (string) Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for --cli-input-json. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command.

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Output

ScalingPlans -> (list)

Information about the scaling plans.

(structure)

Represents a scaling plan.

ScalingPlanName -> (string)

The name of the scaling plan.

ScalingPlanVersion -> (long)

The version number of the scaling plan.

ApplicationSource -> (structure)

The application source.

CloudFormationStackARN -> (string)

The Amazon Resource Name (ARN) of a AWS CloudFormation stack.

TagFilters -> (list)

A set of tags (up to 50).

(structure)

Represents a tag.

Key -> (string)

The tag key.

Values -> (list)

The tag values (0 to 20).

(string)

ScalingInstructions -> (list)

The scaling instructions.

(structure)

Describes a scaling instruction for a scalable resource.

The scaling instruction is used in combination with a scaling plan, which is a set of instructions for configuring dynamic scaling and predictive scaling for the scalable resources in your application. Each scaling instruction applies to one resource.

AWS Auto Scaling creates target tracking scaling policies based on the scaling instructions. Target tracking scaling policies adjust the capacity of your scalable resource as required to maintain resource utilization at the target value that you specified.

AWS Auto Scaling also configures predictive scaling for your Amazon EC2 Auto Scaling groups using a subset of parameters, including the load metric, the scaling metric, the target value for the scaling metric, the predictive scaling mode (forecast and scale or forecast only), and the desired behavior when the forecast capacity exceeds the maximum capacity of the resource. With predictive scaling, AWS Auto Scaling generates forecasts with traffic predictions for the two days ahead and schedules scaling actions that proactively add and remove resource capacity to match the forecast.

For more information, see the [AWS Auto Scaling User Guide](http://docs.aws.amazon.com/autoscaling/plans/userguide/what-is-aws-auto-scaling.html) .

ServiceNamespace -> (string)

The namespace of the AWS service.

ResourceId -> (string)

The ID of the resource. This string consists of the resource type and unique identifier.

* Auto Scaling group - The resource type is autoScalingGroup and the unique identifier is the name of the Auto Scaling group. Example: autoScalingGroup/my-asg .
* ECS service - The resource type is service and the unique identifier is the cluster name and service name. Example: service/default/sample-webapp .
* Spot Fleet request - The resource type is spot-fleet-request and the unique identifier is the Spot Fleet request ID. Example: spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE .
* DynamoDB table - The resource type is table and the unique identifier is the resource ID. Example: table/my-table .
* DynamoDB global secondary index - The resource type is index and the unique identifier is the resource ID. Example: table/my-table/index/my-table-index .
* Aurora DB cluster - The resource type is cluster and the unique identifier is the cluster name. Example: cluster:my-db-cluster .

ScalableDimension -> (string)

The scalable dimension associated with the resource.

* autoscaling:autoScalingGroup:DesiredCapacity - The desired capacity of an Auto Scaling group.
* ecs:service:DesiredCount - The desired task count of an ECS service.
* ec2:spot-fleet-request:TargetCapacity - The target capacity of a Spot Fleet request.
* dynamodb:table:ReadCapacityUnits - The provisioned read capacity for a DynamoDB table.
* dynamodb:table:WriteCapacityUnits - The provisioned write capacity for a DynamoDB table.
* dynamodb:index:ReadCapacityUnits - The provisioned read capacity for a DynamoDB global secondary index.
* dynamodb:index:WriteCapacityUnits - The provisioned write capacity for a DynamoDB global secondary index.
* rds:cluster:ReadReplicaCount - The count of Aurora Replicas in an Aurora DB cluster. Available for Aurora MySQL-compatible edition.

MinCapacity -> (integer)

The minimum capacity of the resource.

MaxCapacity -> (integer)

The maximum capacity of the resource. The exception to this upper limit is if you specify a non-default setting for **PredictiveScalingMaxCapacityBehavior** .

TargetTrackingConfigurations -> (list)

The structure that defines new target tracking configurations (up to 10). Each of these structures includes a specific scaling metric and a target value for the metric, along with various parameters to use with dynamic scaling.

With predictive scaling and dynamic scaling, the resource scales based on the target tracking configuration that provides the largest capacity for both scale in and scale out.

Condition: The scaling metric must be unique across target tracking configurations.

(structure)

Describes a target tracking configuration. Used with ScalingInstruction and ScalingPolicy .

PredefinedScalingMetricSpecification -> (structure)

A predefined metric.

PredefinedScalingMetricType -> (string)

The metric type. The ALBRequestCountPerTarget metric type applies only to Auto Scaling groups, Spot Fleet requests, and ECS services.

ResourceLabel -> (string)

Identifies the resource associated with the metric type. You can't specify a resource label unless the metric type is ALBRequestCountPerTarget and there is a target group for an Application Load Balancer attached to the Auto Scaling group, Spot Fleet request, or ECS service.

The format is app/load-balancer-name/load-balancer-id/targetgroup/target-group-name/target-group-id, where:

* app/load-balancer-name/load-balancer-idis the final portion of the load balancer ARN.
* targetgroup/target-group-name/target-group-idis the final portion of the target group ARN.

CustomizedScalingMetricSpecification -> (structure)

A customized metric.

MetricName -> (string)

The name of the metric.

Namespace -> (string)

The namespace of the metric.

Dimensions -> (list)

The dimensions of the metric.

(structure)

Represents a dimension for a customized metric.

Name -> (string)

The name of the dimension.

Value -> (string)

The value of the dimension.

Statistic -> (string)

The statistic of the metric.

Unit -> (string)

The unit of the metric.

TargetValue -> (double)

The target value for the metric. The range is 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2).

DisableScaleIn -> (boolean)

Indicates whether scale in by the target tracking scaling policy is disabled. If the value is true , scale in is disabled and the target tracking scaling policy doesn't remove capacity from the scalable resource. Otherwise, scale in is enabled and the target tracking scaling policy can remove capacity from the scalable resource.

The default value is false .

ScaleOutCooldown -> (integer)

The amount of time, in seconds, after a scale-out activity completes before another scale-out activity can start. This value is not used if the scalable resource is an Auto Scaling group.

While the cooldown period is in effect, the capacity that has been added by the previous scale-out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. The intention is to continuously (but not excessively) scale out.

ScaleInCooldown -> (integer)

The amount of time, in seconds, after a scale in activity completes before another scale in activity can start. This value is not used if the scalable resource is an Auto Scaling group.

The cooldown period is used to block subsequent scale in requests until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in, AWS Auto Scaling scales out your scalable target immediately.

EstimatedInstanceWarmup -> (integer)

The estimated time, in seconds, until a newly launched instance can contribute to the CloudWatch metrics. This value is used only if the resource is an Auto Scaling group.

PredefinedLoadMetricSpecification -> (structure)

The predefined load metric to use for predictive scaling. This parameter or a **CustomizedLoadMetricSpecification** is required when configuring predictive scaling, and cannot be used otherwise.

PredefinedLoadMetricType -> (string)

The metric type.

ResourceLabel -> (string)

Identifies the resource associated with the metric type. You can't specify a resource label unless the metric type is ALBRequestCountPerTarget and there is a target group for an Application Load Balancer attached to the Auto Scaling group.

The format is app/load-balancer-name/load-balancer-id/targetgroup/target-group-name/target-group-id, where:

* app/load-balancer-name/load-balancer-idis the final portion of the load balancer ARN.
* targetgroup/target-group-name/target-group-idis the final portion of the target group ARN.

CustomizedLoadMetricSpecification -> (structure)

The customized load metric to use for predictive scaling. This parameter or a **PredefinedLoadMetricSpecification** is required when configuring predictive scaling, and cannot be used otherwise.

MetricName -> (string)

The name of the metric.

Namespace -> (string)

The namespace of the metric.

Dimensions -> (list)

The dimensions of the metric.

(structure)

Represents a dimension for a customized metric.

Name -> (string)

The name of the dimension.

Value -> (string)

The value of the dimension.

Statistic -> (string)

The statistic of the metric. Currently, the value must always be Sum .

Unit -> (string)

The unit of the metric.

ScheduledActionBufferTime -> (integer)

The amount of time, in seconds, to buffer the run time of scheduled scaling actions when scaling out. For example, if the forecast says to add capacity at 10:00 AM, and the buffer time is 5 minutes, then the run time of the corresponding scheduled scaling action will be 9:55 AM. The intention is to give resources time to be provisioned. For example, it can take a few minutes to launch an EC2 instance. The actual amount of time required depends on several factors, such as the size of the instance and whether there are startup scripts to complete.

The value must be less than the forecast interval duration of 3600 seconds (60 minutes). The default is 300 seconds.

Only valid when configuring predictive scaling.

PredictiveScalingMaxCapacityBehavior -> (string)

Defines the behavior that should be applied if the forecast capacity approaches or exceeds the maximum capacity specified for the resource. The default value is SetForecastCapacityToMaxCapacity .

The following are possible values:

* SetForecastCapacityToMaxCapacity - AWS Auto Scaling cannot scale resource capacity higher than the maximum capacity. The maximum capacity is enforced as a hard limit.
* SetMaxCapacityToForecastCapacity - AWS Auto Scaling may scale resource capacity higher than the maximum capacity to equal but not exceed forecast capacity.
* SetMaxCapacityAboveForecastCapacity - AWS Auto Scaling may 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.

Only valid when configuring predictive scaling.

PredictiveScalingMaxCapacityBuffer -> (integer)

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, if the buffer is 10, this means a 10 percent buffer, such that if the forecast capacity is 50, and the maximum capacity is 40, then the effective maximum capacity is 55.

Only valid when configuring predictive scaling. Required if the **PredictiveScalingMaxCapacityBehavior** is set to SetMaxCapacityAboveForecastCapacity, and cannot be used otherwise.

The range is 1-100.

PredictiveScalingMode -> (string)

The predictive scaling mode. The default value is ForecastAndScale . Otherwise, AWS Auto Scaling forecasts capacity but does not create any scheduled scaling actions based on the capacity forecast.

ScalingPolicyUpdateBehavior -> (string)

Controls whether a resource's externally created scaling policies are kept or replaced.

The default value is KeepExternalPolicies . If the parameter is set to ReplaceExternalPolicies , any scaling policies that are external to AWS Auto Scaling are deleted and new target tracking scaling policies created.

Only valid when configuring dynamic scaling.

Condition: The number of existing policies to be replaced must be less than or equal to 50. If there are more than 50 policies to be replaced, AWS Auto Scaling keeps all existing policies and does not create new ones.

DisableDynamicScaling -> (boolean)

Controls whether dynamic scaling by AWS Auto Scaling is disabled. When dynamic scaling is enabled, AWS Auto Scaling creates target tracking scaling policies based on the specified target tracking configurations.

The default is enabled (false ).

StatusCode -> (string)

The status of the scaling plan.

* Active - The scaling plan is active.
* ActiveWithProblems - The scaling plan is active, but the scaling configuration for one or more resources could not be applied.
* CreationInProgress - The scaling plan is being created.
* CreationFailed - The scaling plan could not be created.
* DeletionInProgress - The scaling plan is being deleted.
* DeletionFailed - The scaling plan could not be deleted.
* UpdateInProgress - The scaling plan is being updated.
* UpdateFailed - The scaling plan could not be updated.

StatusMessage -> (string)

A simple message about the current status of the scaling plan.

StatusStartTime -> (timestamp)

The Unix time stamp when the scaling plan entered the current status.

CreationTime -> (timestamp)

The Unix time stamp when the scaling plan was created.

NextToken -> (string)

The token required to get the next set of results. This value is null if there are no more results to return.

# Get-scaling-plan-resource-forecast-data

## Description

Retrieves the forecast data for a scalable resource.

Capacity forecasts are represented as predicted values, or data points, that are calculated using historical data points from a specified CloudWatch load metric. Data points are available for up to 56 days.

See also: [AWS API Documentation](https://docs.aws.amazon.com/goto/WebAPI/autoscaling-plans-2018-01-06/GetScalingPlanResourceForecastData)

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Synopsis

get-scaling-plan-resource-forecast-data

--scaling-plan-name <value>

--scaling-plan-version <value>

--service-namespace <value>

--resource-id <value>

--scalable-dimension <value>

--forecast-data-type <value>

--start-time <value>

--end-time <value>

[--cli-input-json <value>]

[--generate-cli-skeleton <value>]

## Options

--scaling-plan-name (string)

The name of the scaling plan.

--scaling-plan-version (long)

The version number of the scaling plan.

--service-namespace (string)

The namespace of the AWS service.

Possible values:

* autoscaling
* ecs
* ec2
* rds
* dynamodb

--resource-id (string)

The ID of the resource. This string consists of the resource type and unique identifier.

* Auto Scaling group - The resource type is autoScalingGroup and the unique identifier is the name of the Auto Scaling group. Example: autoScalingGroup/my-asg .
* ECS service - The resource type is service and the unique identifier is the cluster name and service name. Example: service/default/sample-webapp .
* Spot Fleet request - The resource type is spot-fleet-request and the unique identifier is the Spot Fleet request ID. Example: spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE .
* DynamoDB table - The resource type is table and the unique identifier is the resource ID. Example: table/my-table .
* DynamoDB global secondary index - The resource type is index and the unique identifier is the resource ID. Example: table/my-table/index/my-table-index .
* Aurora DB cluster - The resource type is cluster and the unique identifier is the cluster name. Example: cluster:my-db-cluster .

--scalable-dimension (string)

The scalable dimension for the resource.

Possible values:

* autoscaling:autoScalingGroup:DesiredCapacity
* ecs:service:DesiredCount
* ec2:spot-fleet-request:TargetCapacity
* rds:cluster:ReadReplicaCount
* dynamodb:table:ReadCapacityUnits
* dynamodb:table:WriteCapacityUnits
* dynamodb:index:ReadCapacityUnits
* dynamodb:index:WriteCapacityUnits

--forecast-data-type (string)

The type of forecast data to get.

* LoadForecast : The load metric forecast.
* CapacityForecast : The capacity forecast.
* ScheduledActionMinCapacity : The minimum capacity for each scheduled scaling action. This data is calculated as the larger of two values: the capacity forecast or the minimum capacity in the scaling instruction.
* ScheduledActionMaxCapacity : The maximum capacity for each scheduled scaling action. The calculation used is determined by the predictive scaling maximum capacity behavior setting in the scaling instruction.

Possible values:

* CapacityForecast
* LoadForecast
* ScheduledActionMinCapacity
* ScheduledActionMaxCapacity

--start-time (timestamp)

The inclusive start time of the time range for the forecast data to get. The date and time can be at most 56 days before the current date and time.

--end-time (timestamp)

The exclusive end time of the time range for the forecast data to get. The maximum time duration between the start and end time is seven days.

Although this parameter can accept a date and time that is more than two days in the future, the availability of forecast data has limits. AWS Auto Scaling only issues forecasts for periods of two days in advance.

--cli-input-json (string) Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values. It is not possible to pass arbitrary binary values using a JSON-provided value as the string will be taken literally.

--generate-cli-skeleton (string) Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for --cli-input-json. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command.

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Output

Datapoints -> (list)

The data points to return.

(structure)

Represents a single value in the forecast data used for predictive scaling.

Timestamp -> (timestamp)

The time stamp for the data point in UTC format.

Value -> (double)

The value of the data point.

# Update-scaling-plan

## Description

Updates the specified scaling plan.

You cannot update a scaling plan if it is in the process of being created, updated, or deleted.

See also: [AWS API Documentation](https://docs.aws.amazon.com/goto/WebAPI/autoscaling-plans-2018-01-06/UpdateScalingPlan)

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Synopsis

update-scaling-plan

--scaling-plan-name <value>

--scaling-plan-version <value>

[--application-source <value>]

[--scaling-instructions <value>]

[--cli-input-json <value>]

[--generate-cli-skeleton <value>]

## Options

--scaling-plan-name (string)

The name of the scaling plan.

--scaling-plan-version (long)

The version number of the scaling plan.

--application-source (structure)

A CloudFormation stack or set of tags.

JSON Syntax:

{

"CloudFormationStackARN": "string",

"TagFilters": [

{

"Key": "string",

"Values": ["string", ...]

}

...

]

}

--scaling-instructions (list)

The scaling instructions.

JSON Syntax:

[

{

"ServiceNamespace": "autoscaling"|"ecs"|"ec2"|"rds"|"dynamodb",

"ResourceId": "string",

"ScalableDimension": "autoscaling:autoScalingGroup:DesiredCapacity"|"ecs:service:DesiredCount"|"ec2:spot-fleet-request:TargetCapacity"|"rds:cluster:ReadReplicaCount"|"dynamodb:table:ReadCapacityUnits"|"dynamodb:table:WriteCapacityUnits"|"dynamodb:index:ReadCapacityUnits"|"dynamodb:index:WriteCapacityUnits",

"MinCapacity": integer,

"MaxCapacity": integer,

"TargetTrackingConfigurations": [

{

"PredefinedScalingMetricSpecification": {

"PredefinedScalingMetricType": "ASGAverageCPUUtilization"|"ASGAverageNetworkIn"|"ASGAverageNetworkOut"|"DynamoDBReadCapacityUtilization"|"DynamoDBWriteCapacityUtilization"|"ECSServiceAverageCPUUtilization"|"ECSServiceAverageMemoryUtilization"|"ALBRequestCountPerTarget"|"RDSReaderAverageCPUUtilization"|"RDSReaderAverageDatabaseConnections"|"EC2SpotFleetRequestAverageCPUUtilization"|"EC2SpotFleetRequestAverageNetworkIn"|"EC2SpotFleetRequestAverageNetworkOut",

"ResourceLabel": "string"

},

"CustomizedScalingMetricSpecification": {

"MetricName": "string",

"Namespace": "string",

"Dimensions": [

{

"Name": "string",

"Value": "string"

}

...

],

"Statistic": "Average"|"Minimum"|"Maximum"|"SampleCount"|"Sum",

"Unit": "string"

},

"TargetValue": double,

"DisableScaleIn": true|false,

"ScaleOutCooldown": integer,

"ScaleInCooldown": integer,

"EstimatedInstanceWarmup": integer

}

...

],

"PredefinedLoadMetricSpecification": {

"PredefinedLoadMetricType": "ASGTotalCPUUtilization"|"ASGTotalNetworkIn"|"ASGTotalNetworkOut"|"ALBTargetGroupRequestCount",

"ResourceLabel": "string"

},

"CustomizedLoadMetricSpecification": {

"MetricName": "string",

"Namespace": "string",

"Dimensions": [

{

"Name": "string",

"Value": "string"

}

...

],

"Statistic": "Average"|"Minimum"|"Maximum"|"SampleCount"|"Sum",

"Unit": "string"

},

"ScheduledActionBufferTime": integer,

"PredictiveScalingMaxCapacityBehavior": "SetForecastCapacityToMaxCapacity"|"SetMaxCapacityToForecastCapacity"|"SetMaxCapacityAboveForecastCapacity",

"PredictiveScalingMaxCapacityBuffer": integer,

"PredictiveScalingMode": "ForecastAndScale"|"ForecastOnly",

"ScalingPolicyUpdateBehavior": "KeepExternalPolicies"|"ReplaceExternalPolicies",

"DisableDynamicScaling": true|false

}

...

]

--cli-input-json (string) Performs service operation based on the JSON string provided. The JSON string follows the format provided by --generate-cli-skeleton. If other arguments are provided on the command line, the CLI values will override the JSON-provided values. It is not possible to pass arbitrary binary values using a JSON-provided value as the string will be taken literally.

--generate-cli-skeleton (string) Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for --cli-input-json. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command.

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

## Output

--scaling-plan-version <value>

[--max-results <value>]

[--next-token <value>]

[--cli-input-json <value>]

[--generate-cli-skeleton <value>]

eton (string) Prints a JSON skeleton to standard output without sending an API request. If provided with no value or the value input, prints a sample input JSON that can be used as an argument for --cli-input-json. If provided with the value output, it validates the command inputs and returns a sample output JSON for that command.

See ['aws help'](https://docs.aws.amazon.com/cli/latest/reference/index.html) for descriptions of global parameters.

Represents a scalable resource.

ScalingPlanName -> (string)

The name of the scaling plan.

ScalingPlanVersion -> (long)

The version number of the scaling plan.

ServiceNamespace -> (string)

The namespace of the AWS service.

ResourceId -> (string)

The ID of the resource. This string consists of the resource type and unique identifier.

* Auto Scaling group - The resource type is autoScalingGroup and the unique identifier is the name of the Auto Scaling group. Example: autoScalingGroup/my-asg .
* ECS service - The resource type is service and the unique identifier is the cluster name and service name. Example: service/default/sample-webapp .
* Spot Fleet request - The resource type is spot-fleet-request and the unique identifier is the Spot Fleet request ID. Example: spot-fleet-request/sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE .
* DynamoDB table - The resource type is table and the unique identifier is the resource ID. Example: table/my-table .
* DynamoDB global secondary index - The resource type is index and the unique identifier is the resource ID. Example: table/my-table/index/my-table-index .
* Aurora DB cluster - The resource type is cluster and the unique identifier is the cluster name. Example: cluster:my-db-cluster .

ScalableDimension -> (string)

The scalable dimension for the resource.

* autoscaling:autoScalingGroup:DesiredCapacity - The desired capacity of an Auto Scaling group.
* ecs:service:DesiredCount - The desired task count of an ECS service.
* ec2:spot-fleet-request:TargetCapacity - The target capacity of a Spot Fleet request.
* dynamodb:table:ReadCapacityUnits - The provisioned read capacity for a DynamoDB table.
* dynamodb:table:WriteCapacityUnits - The provisioned write capacity for a DynamoDB table.
* dynamodb:index:ReadCapacityUnits - The provisioned read capacity for a DynamoDB global secondary index.
* dynamodb:index:WriteCapacityUnits - The provisioned write capacity for a DynamoDB global secondary index.
* rds:cluster:ReadReplicaCount - The count of Aurora Replicas in an Aurora DB cluster. Available for Aurora MySQL-compatible edition.

ScalingPolicies -> (list)

The scaling policies.

(structure)

Represents a scaling policy.

PolicyName -> (string)

The name of the scaling policy.

PolicyType -> (string)

The type of scaling policy.

TargetTrackingConfiguration -> (structure)

The target tracking scaling policy.

PredefinedScalingMetricSpecification -> (structure)

A predefined metric.

PredefinedScalingMetricType -> (string)

The metric type. The ALBRequestCountPerTarget metric type applies only to Auto Scaling groups, Spot Fleet requests, and ECS services.

ResourceLabel -> (string)

Identifies the resource associated with the metric type. You can't specify a resource label unless the metric type is ALBRequestCountPerTarget and there is a target group for an Application Load Balancer attached to the Auto Scaling group, Spot Fleet request, or ECS service.

The format is app/load-balancer-name/load-balancer-id/targetgroup/target-group-name/target-group-id, where:

* app/load-balancer-name/load-balancer-idis the final portion of the load balancer ARN.
* targetgroup/target-group-name/target-group-idis the final portion of the target group ARN.

CustomizedScalingMetricSpecification -> (structure)

A customized metric.

MetricName -> (string)

The name of the metric.

Namespace -> (string)

The namespace of the metric.

Dimensions -> (list)

The dimensions of the metric.

(structure)

Represents a dimension for a customized metric.

Name -> (string)

The name of the dimension.

Value -> (string)

The value of the dimension.

Statistic -> (string)

The statistic of the metric.

Unit -> (string)

The unit of the metric.

TargetValue -> (double)

The target value for the metric. The range is 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2).

DisableScaleIn -> (boolean)

Indicates whether scale in by the target tracking scaling policy is disabled. If the value is true , scale in is disabled and the target tracking scaling policy doesn't remove capacity from the scalable resource. Otherwise, scale in is enabled and the target tracking scaling policy can remove capacity from the scalable resource.

The default value is false .

ScaleOutCooldown -> (integer)

The amount of time, in seconds, after a scale-out activity completes before another scale-out activity can start. This value is not used if the scalable resource is an Auto Scaling group.

While the cooldown period is in effect, the capacity that has been added by the previous scale-out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. The intention is to continuously (but not excessively) scale out.

ScaleInCooldown -> (integer)

The amount of time, in seconds, after a scale in activity completes before another scale in activity can start. This value is not used if the scalable resource is an Auto Scaling group.

The cooldown period is used to block subsequent scale in requests until it has expired. The intention is to scale in conservatively to protect your application's availability. However, if another alarm triggers a scale-out policy during the cooldown period after a scale-in, AWS Auto Scaling scales out your scalable target immediately.

EstimatedInstanceWarmup -> (integer)

The estimated time, in seconds, until a newly launched instance can contribute to the CloudWatch metrics. This value is used only if the resource is an Auto Scaling group.

ScalingStatusCode -> (string)

The scaling status of the resource.

* Active - The scaling configuration is active.
* Inactive - The scaling configuration is not active because the scaling plan is being created or the scaling configuration could not be applied. Check the status message for more information.
* PartiallyActive - The scaling configuration is partially active because the scaling plan is being created or deleted or the scaling configuration could not be fully applied. Check the status message for more information.

ScalingStatusMessage -> (string)

A simple message about the current scaling status of the resource.

NextToken -> (string)

The token required to get the next set of results. This value is null if there are no more results to return.