AUTOSCALING

You can determine which subnets Auto Scaling will launch new instances into.

Launch configuration is the template used to create new EC2 instances and includes parameters such as instance family, instance type, AMI, key pair and security groups.

The AMI must exist on EC2

* If you want to change your launch configurations you have to create a new one, make the required changes, and use that with your auto scaling groups.

You can attach one or more Target Groups to your ASG to include instances behind an ALB.

You can add a running instance to an ASG if the following conditions are met:

* The instance is in a running state.
* The AMI used to launch the instance still exists.
* The instance is not part of another ASG.
* The instance is in the same AZs for the ASG.
* Q. When should I use Amazon EC2 Auto Scaling vs. AWS Auto Scaling?
* You should use AWS Auto Scaling to manage scaling for multiple resources across multiple services. AWS Auto Scaling lets you define dynamic scaling policies for multiple EC2 Auto Scaling groups or other resources using predefined scaling strategies. Using AWS Auto Scaling to configure scaling policies for all of the scalable resources in your application is faster than managing scaling policies for each resource via its individual service console. It’s also easier, as AWS Auto Scaling includes predefined scaling strategies that simplify the setup of scaling policies. You should also use AWS Auto Scaling if you want to create predictive scaling for EC2 resources.
* You should use EC2 Auto Scaling if you only need to scale Amazon EC2 Auto Scaling groups, or if you are only interested in maintaining the health of your EC2 fleet. You should also use EC2 Auto Scaling if you need to create or configure Amazon EC2 Auto Scaling groups, or if you need to set up scheduled or step scaling policies (as AWS Auto Scaling supports only target tracking scaling policies).
* EC2 Auto Scaling groups must be created and configured outside of AWS Auto Scaling, such as through the EC2 console, Auto Scaling API or via CloudFormation. AWS Auto Scaling can help you configure dynamic scaling policies for your existing EC2 Auto Scaling groups.
* Q: What is fleet management and how is it different from dynamic scaling?
* If your application runs on Amazon EC2 instances, then you have what’s referred to as a ‘fleet’. *Fleet management* refers to the functionality that automatically replaces unhealthy instances and maintains your fleet at the desired capacity. Amazon EC2 Auto Scaling fleet management ensures that your application is able to receive traffic and that the instances themselves are working properly. When Auto Scaling detects a failed [health check](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html), it can replace the instance automatically.
* The *dynamic scaling* capabilities of Amazon EC2 Auto Scaling refers to the functionality that automatically increases or decreases capacity based on load or other metrics. For example, if your CPU spikes above 80% (and you have an alarm setup) Amazon EC2 Auto Scaling can add a new instance dynamically
* Q: What is target tracking?
* Target tracking is a new type of scaling policy that you can use to set up dynamic scaling for your application in just a few simple steps. With target tracking, you select a load metric for your application, such as CPU utilization or request count, set the target value, and Amazon EC2 Auto Scaling adjusts the number of EC2 instances in your ASG as needed to maintain that target. It acts like a home thermostat, automatically adjusting the system to keep the environment at your desired temperature. For example, you can configure target tracking to keep CPU utilization for your fleet of web servers at 50%. From there, Amazon EC2 Auto Scaling launches or terminates EC2 instances as required to keep the average CPU utilization at 50%.

Q. What are the benefits of AWS Auto Scaling?

AWS Auto Scaling is a fast, easy way to optimize the performance and costs of your applications.

* Setup scaling quickly: AWS Auto Scaling provides a unified scaling experience for all of the scalable resources powering your application. You can see the average utilization for all of your scalable resources and quickly define target utilization levels for each group of like resources from a single, intuitive interface.
* Make smart scaling decisions: AWS Auto Scaling lets you automate how groups of different resources respond to changes in demand. Easy-to-understand scaling strategies let you choose to optimize availability, costs, or a balance of both. AWS Auto Scaling automatically creates all of the scaling policies and sets targets for you based on your preference.
* Automatically maintain performance: AWS Auto Scaling continually monitors resources underlying your application to make sure that they are operating at your desired performance levels. When demand spikes, AWS Auto Scaling automatically increases the capacity of constrained resources so you maintain a high quality of service.
* Anticipate costs and avoid overspending: AWS Auto Scaling can help you optimize your utilization and cost efficiencies when consuming AWS services so you only pay for the resources you actually need. When demand drops, AWS Auto Scaling will automatically remove any excess resource capacity so you avoid overspending.
* Q. When should I use AWS Auto Scaling?
* You should use AWS Auto Scaling if you have an application that uses one or more scalable resources and experiences variable load. A good example would be an e-commerce web application that receives variable traffic through the day. It follows a standard three tier architecture with Elastic Load Balancing for distributing incoming traffic, Amazon EC2 for the compute layer, and DynamoDB for the data layer. In this case, AWS Auto Scaling will scale one or more EC2 Auto Scaling groups and DynamoDB tables that are powering the application in response to the demand curve

Q: What happens to my Amazon EC2 instances if I delete my ASG?

If you have an EC2 Auto Scaling group (ASG) with running instances and you choose to delete the ASG, the instances will be terminated and the ASG will be deleted.

Q: How do I know when EC2 Auto Scaling is launching or terminating the EC2 instances in an EC2 Auto Scaling group?

When you use Amazon EC2 Auto Scaling to scale your applications automatically, it is useful to know when EC2 Auto Scaling is launching or terminating the EC2 instances in your EC2 Auto Scaling group. [Amazon SNS](https://aws.amazon.com/sns/) coordinates and manages the delivery or sending of notifications to subscribing clients or endpoints. You can configure EC2 Auto Scaling to send an SNS notification whenever your EC2 Auto Scaling group scales. Amazon SNS can deliver notifications as HTTP or HTTPS POST, email (SMTP, either plain-text or in JSON format), or as a message posted to an Amazon SQS queue. For example, if you configure your EC2 Auto Scaling group to use the autoscaling: EC2\_INSTANCE\_TERMINATE notification type, and your EC2 Auto Scaling group terminates an instance, it sends an email notification. This email contains the details of the terminated instance, such as the instance ID and the reason that the instance was terminated.

Q: How can I implement changes across multiple instances in an EC2 Auto Scaling group?

You can use AWS CodeDeploy or CloudFormation to orchestrate code changes to multiple instances in your EC2 Auto Scaling group.

Q: If I have data installed in an EC2 Auto Scaling group, and a new instance is dynamically created later, is the data copied over to the new instances?

Data is not automatically copied from existing instances to new instances. You can use [lifecycle hooks](http://docs.aws.amazon.com/autoscaling/latest/userguide/lifecycle-hooks.html) to copy the data, or an [Amazon RDS](https://aws.amazon.com/rds/) database including replicas.

Q: Can I customize a health check?

Yes, there is an API called *SetInstanceHealth*that allows you to change an instance's state to UNHEALTHY, which will then result in a termination and replacement.

Q: Can I suspend health checks (for example, to evaluate unhealthy instances)?

Yes, you can temporarily suspend Amazon EC2 Auto Scaling health checks by using the SuspendProcesses API. You can use the ResumeProcesses API to resume automatic health checks.

Q: Which health check type should I select?

If you are using Elastic Load Balancing (ELB) with your group, you should select an ELB health check. If you’re not using ELB with your group, you should select the EC2 health check.

Q: Can I use Amazon EC2 Auto Scaling for health checks and to replace unhealthy instances if I’m not using Elastic Load Balancing (ELB)?

You don't have to use ELB to use Auto Scaling. You can use the EC2 health check to identify and replace unhealthy instances.

Q: Is there any way to use Amazon EC2 Auto Scaling to only add a volume without adding an instance?

A volume is attached to a new instance when it is added. Amazon EC2 Auto Scaling doesn't automatically add a volume when the existing one is approaching capacity. You can use the EC2 API to add a volume to an existing instance.

Q: Do the Elastic Load Balancing (ELB) health checks work with Application Load Balancers and Network Load Balancers? Will an instance be marked as unhealthy if any target group associated with it becomes unhealthy?

Yes, Amazon EC2 Auto Scaling works with Application Load Balancers and Network Load Balancers including their health check feature.

Q: What does the term “stateful instances” refer to?

When we refer to a stateful instance, we mean an instance that has data on it, which exists only on that instance. In general, terminating a stateful instance means that the data (or state information) on the instance is lost. You may want to consider using lifecycle hooks to copy the data off of a stateful instance before it’s terminated, or enable instance protection to prevent Amazon EC2 Auto Scaling from terminating it

Q: How does Amazon EC2 Auto Scaling replace an impaired instance?

When an impaired instance fails a health check, Amazon EC2 Auto Scaling automatically terminates it and replaces it with a new one. If you’re using an Elastic Load Balancing load balancer, Amazon EC2 Auto Scaling gracefully detaches the impaired instance from the load balancer before provisioning a new one and attaching it to the load balancer. This is all done automatically, so you don’t need to respond manually when an instance needs replacing.

Q: How do I control which instances Amazon EC2 Auto Scaling terminates when scaling in, and how do I protect data on an instance?

With each Amazon EC2 Auto Scaling group, you control when Amazon EC2 Auto Scaling adds instances (referred to as scaling out) or remove instances (referred to as scaling in) from your group. You can scale the size of your group manually by attaching and detaching instances, or you can automate the process through the use of a scaling policy. When you have Amazon EC2 Auto Scaling automatically scale in, you must decide which instances Amazon EC2 Auto Scaling should terminate first. You can configure this through the use of a termination policy. You can also use instance protection to prevent Amazon EC2 Auto Scaling from selecting specific instances for termination when scaling in. If you have data on an instance, and you need that data to be persistent even if your instance is scaled in, then you can use a service like S3, RDS, or DynamoDB, to make sure that it is stored off the instance.

Q: If Elastic Load Balancing (ELB) determines that an instance is unhealthy, and moved offline, will the previous requests sent to the failed instance be queued and rerouted to other instances within the group?

When ELB notices that the instance is unhealthy, it will stop routing requests to it. However, prior to discovering that the instance is unhealthy, some requests to that instance will fail.

Q: If you don’t use Elastic Load Balancing (ELB) how would users be directed to the other servers in a group if there was a failure?

You can integrate with Route53 (which Amazon EC2 Auto Scaling does not currently support out of the box, but many customers use). You can also use your own reverse proxy, or for internal microservices, can use service discovery solutions.

Q: Are CloudWatch agents automatically installed on EC2 instances when you create an Amazon EC2 Auto Scaling group?

If your AMI contains a CloudWatch agent, it’s automatically installed on EC2 instances when you create an EC2 Auto Scaling group. With the stock Amazon Linux AMI, you need to install it (recommended, via yum).

Q: Can I create a single ASG to scale instances across different purchase options?

Yes. You can provision and automatically scale EC2 capacity across different EC2 instance types, Availability Zones, and On-Demand, RIs and Spot purchase options in a single Auto Scaling Group. You have the option to define the desired split between On-Demand and Spot capacity, select which instance types work for your application, and specify preference for how EC2 Auto Scaling should distribute the ASG capacity within each purchasing model.

Q: Can I use ASGs to launch and manage just Spot Instances or just On-Demand instances and RIs?

Yes. You can configure your ASG specifying all capacity to be only Spot instances or all capacity to be only On-Demand instances and RIs.

Q: What if the instance types I like are not available in an Availability Zone?

If none of the specified instance types are available in an Availability Zone, Auto Scaling will retarget the launches in other Availability Zones associated with the Auto Scaling group. Auto Scaling will always prefer keeping your compute balanced across Availability Zones and retarget if all instance types are not available in an Availability Zone.

**The ASG will terminate the EC2 Instance**

To maintain the same number of instances, Amazon EC2 Auto Scaling performs a periodic health check on running instances within an Auto Scaling group. When it finds that an instance is unhealthy, it terminates that instance and launches a new one. Amazon EC2 Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance.

**ASGAverageCPUUtilization** - This is a predefined metric for target tracking scaling policy. This represents the Average CPU utilization of the Auto Scaling group.

**ASGAverageNetworkOut** - This is a predefined metric for target tracking scaling policy. This represents the Average number of bytes sent out on all network interfaces by the Auto Scaling group.

**ALBRequestCountPerTarget** - This is a predefined metric for target tracking scaling policy. This represents the Number of requests completed per target in an Application Load Balancer target group.

It is important to note that a target tracking scaling policy assumes that it should scale out your Auto Scaling group when the specified metric is above the target value.

 Amazon EC2 Auto Scaling doesn't automatically add a volume when the existing one is approaching capacity. You can use the EC2 API to add a volume to an existing instance.

**Amazon EC2 Auto Scaling works with both Application Load Balancers and Network Load Balancers**

EC2 Auto Scaling groups are regional constructs. They can span Availability Zones, but not AWS regions.

Q. When should I use AWS Auto Scaling vs. Amazon EC2 Auto Scaling?

You should use AWS Auto Scaling to manage scaling for multiple resources across multiple services. AWS Auto Scaling lets you define dynamic scaling policies for multiple EC2 Auto Scaling groups or other resources using predefined scaling strategies. Using AWS Auto Scaling to configure scaling policies for all of the scalable resources in your application is faster than managing scaling policies for each resource via its individual service console. It’s also easier, as AWS Auto Scaling includes predefined scaling strategies that simplify the setup of scaling policies. You should also use AWS Auto Scaling if you want to create predictive scaling for EC2 resources.

You should use EC2 Auto Scaling if you only need to scale Amazon EC2 Auto Scaling groups, or if you are only interested in maintaining the health of your EC2 fleet. You should also use EC2 Auto Scaling if you need to create or configure Amazon EC2 Auto Scaling groups, or if you need to set up scheduled or step scaling policies (as AWS Auto Scaling supports only target tracking scaling policies).

EC2 Auto Scaling groups must be created and configured outside of AWS Auto Scaling, such as through the EC2 console, Auto Scaling API or via CloudFormation. AWS Auto Scaling can help you configure dynamic scaling policies for your existing EC2 Auto Scaling groups.

Q. When should I use AWS Auto Scaling vs. Auto Scaling for individual services?

You should use AWS Auto Scaling to manage scaling for multiple resources across multiple services. AWS Auto Scaling enables unified scaling for multiple resources, and has predefined guidance that helps make it easier and faster to configure scaling. If you prefer, you can instead choose to use the individual service consoles, Auto Scaling API, or Application Auto Scaling API to scale individual AWS services. You should also use the individual consoles or API if you want to setup step scaling policies or scheduled scaling, as AWS Auto Scaling creates target tracking scaling policies only.

Q. How is AWS Auto Scaling different than the scaling capabilities for individual services?

The following table provides a comparison of AWS scaling options.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **AWS Auto Scaling** | **Amazon EC2 Auto Scaling** | **Auto Scaling for Other Services** |
| Resources you can scale | EC2 Auto Scaling groups EC2 Spot Fleets ECS services DynamoDB provisioned capacity for tables & GSIs Aurora Replicas | EC2 Auto Scaling groups | EC2 Spot Fleets ECS services DynamoDB provisioned capacity for tables & GSIs Aurora Replicas EMR clusters Appstream 2.0 fleet Sagemaker endpoint variants |
| Scaling method | Application-wide scaling using a unified interface | One Auto Scaling group at a time | One resource at a time |
| Predictive Scaling | Yes (EC2 Only) | No | No |
| Automatic discovery of all scalable resources in your application | Yes | No | No |
| Ability to scale multiple resources across multiple services with a unified interface | Yes | No | No |
| Guidance and recommendations for setting up scaling policies | Yes | No | No |
| Ability to create and setup Auto Scaling groups | No | Yes | Not applicable |
| Ability to use Auto Scaling only for EC2 Fleet Management | No | Yes | Not applicable |
| Setup intelligent, self-optimizing target tracking scaling policies\* | Yes | Yes | Yes |
| Setup scheduled scaling actions | No | Yes | Yes |
| Setup step scaling policies | No | Yes | Yes |
| Configure a scaling policy with different metrics and thresholds for each resource | No | Yes | Yes |

Q. What can I scale with AWS Auto Scaling?

You can use AWS Auto Scaling to setup scaling for the following resources in your application through a single, unified interface:

* [Amazon EC2](https://aws.amazon.com/ec2/) Auto Scaling groups
* [Amazon Elastic Container Service (ECS)](https://aws.amazon.com/ecs/) services (currently ECS services cannot be discovered using resource tags)
* [Amazon EC2 Spot](https://aws.amazon.com/ec2/spot/) Fleets
* [Amazon DynamoDB](https://aws.amazon.com/dynamodb/) throughput capacity
* Aurora replicas for [Amazon Aurora](https://rds/aurora/)

Q. How do I select an application stack within AWS Auto Scaling?

You can either select an AWS CloudFormation stack or select resources based on common resource tag(s). Please note that currently, ECS services cannot be discovered using tags.

Q. How does AWS Auto Scaling discover what resources can scale?

AWS Auto Scaling will scan your selected AWS CloudFormation stack or resources with the specified tags to identify the supported AWS resource types that can be scaled. Please note that currently, ECS services cannot be discovered using tags.