



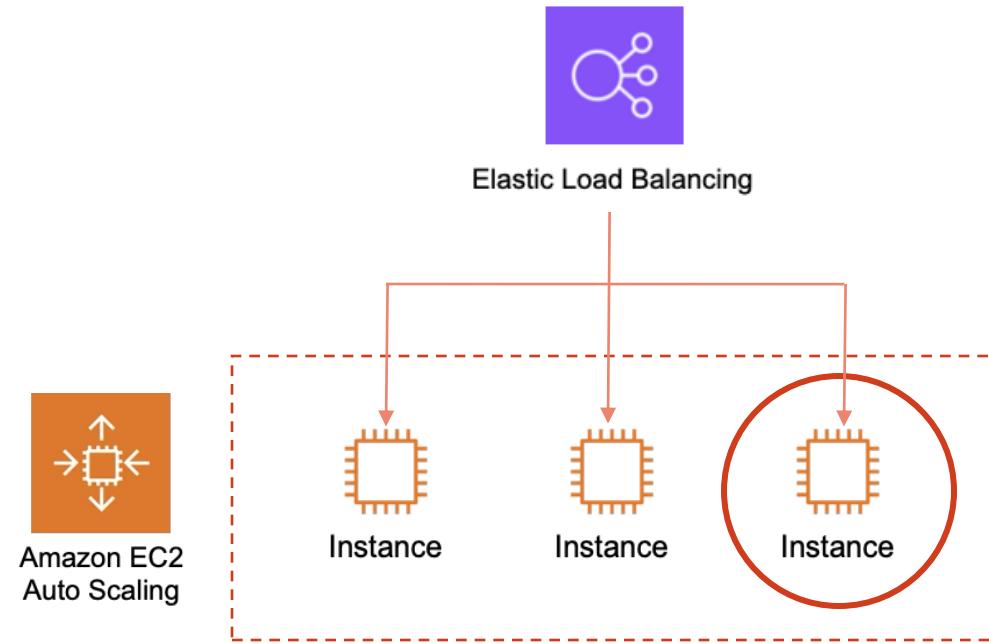
Hands-on Autoscaling and Load-Balancing Using AWS

Experience the scalability and elasticity of cloud computing

Auto-scaling and Load-balancing

High availability, scalability, fault tolerance

- **Load Balancer:** distributes incoming client requests (e.g., web traffic) across multiple backend targets (like EC2 instances).
- **Autoscaling:** automatically adds or removes compute resources (e.g., EC2 instances) based on real-time demand.



Note: **LB** focuses on distributing traffic across **existing** servers, while **AS** adjusts the **number** of servers based on demand.

Step-by-step on load-balancing

- Create VPC

Your VPCs (2) <small>Info</small>						
<small>Last updated 11 minutes ago</small> Actions Create VPC						
<input type="text"/> Find VPCs by attribute or tag						
<input type="checkbox"/>	Name		VPC ID		State	
	Default		vpc-0b2cd...		Available	Off
					IPv4 CIDR	
					172.31.0.0/16	-

- Create at least 2 EC2 instances

- Ensure you have web server for your machines

Instances (8) <small>Info</small>						
<small>Last updated less than a minute ago</small> Connect Instance state Actions Launch instances						
<input type="text"/> Find Instance by attribute or tag (case-sensitive)						
<input type="checkbox"/>	Name		Instance ID		Instance state	
	WebServer		i-0e5c68562822ce7b8		Running	t3.micro
	TutorialDemo		i-06536956a3973f7e3		Running	t3.micro
					Status check	Alarm status
					3/3 checks passed	View alarms +
					3/3 checks passed	View alarms +
						ap-
						ap-

Step-by-step on load-balancing

- Setup **load balancer**

- LoadBalancer type
- Security group
- Listener on target group

Load balancer types

Application Load Balancer [Info](#)

The diagram shows a user connecting to an Application Load Balancer (ALB). The ALB then routes traffic to either an HTTP or HTTPS listener. These listeners then connect to three different targets: a Lambda function, a database, and a microservice.

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

[Create](#)

Network Load Balancer [Info](#)

The diagram shows a user connecting to a VPC, which then connects to a Network Load Balancer (NLB). The NLB then routes traffic to three different protocols: TCP, UDP, and TLS. Each protocol connects to a specific target: ALB for TCP, a database for UDP, and a microservice for TLS.

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

[Create](#)

Gateway Load Balancer [Info](#)

The diagram shows a user connecting to a GWLB, which then connects to a target group. The target group contains three specific targets: a lock icon, a document icon, and a fire icon.

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

[Create](#)

▶ **Classic Load Balancer - previous generation**

Step-by-step on load-balancing

- Setup **load balancer**

- LoadBalancer type
- Security group
- Listener on **target group**

Listeners and routing Info
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener **HTTP:80**

Protocol: HTTP Port: 80
1-65535

Default action | Info
Forward to: TutorialTargetGroup Target type: Instance, IPv4
HTTP

[Create target group](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)

You can add up to 50 more tags.

Targets Monitoring Health checks Attributes Tags

Registered targets (2) Info Anomaly mitigation: Not applicable Deregister Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets < 1 >

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status detail
<input type="checkbox"/>	i-0e5c68562822ce7b8	WebServer	80	ap-southeast-...	Healthy	-
<input type="checkbox"/>	i-06536956a3973f7e3	TutorialDemo	80	ap-southeast-...	Healthy	-

Target type: instances

Target name: TutorialTargetGroup

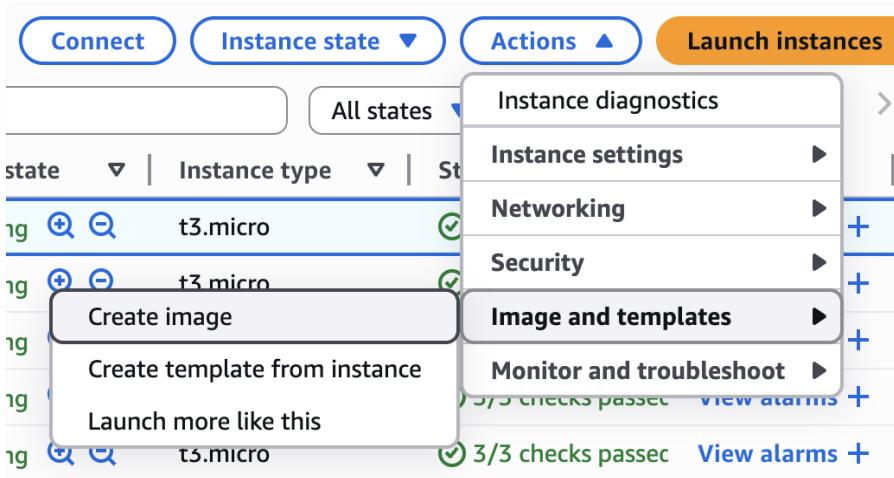
Health check protocol: HTTP

Health check path: /

Register targets: available instances

Step-by-step on autoscaling

- Create my **image**
 - With application, configured settings and dependencies
- Create a **template**
 - Image, instance type...



Note: so, every instance launched by Autoscaling Group starts with the exact same environment.

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance | [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► [Template tags](#)
► [Source template](#)

Step-by-step on autoscaling

- Setup **autoscaling group**

- Attach to a load balancer
- Health checks
- Group size
- Scaling policies: metric, value

Health checks

Health checks increase availability by replacing unhealthy instances. At least one fails, instance replacement occurs.

EC2 health checks

[\(i\) Always enabled](#)

Additional health check types - optional | [Info](#)

Turn on Elastic Load Balancing health checks [Recommended](#)

Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

[\(i\)](#) EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the [Load Balancer console](#) 

Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups  

MyTargetGroup | HTTP 
Application Load Balancer: TutorialLB

Step-by-step on autoscaling

- Setup **autoscaling group**
 - Attach to a load balancer
 - Health checks
 - Group size
 - **Scaling policies**

Automatic scaling - optional

Choose whether to use a target tracking policy | [Info](#)

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

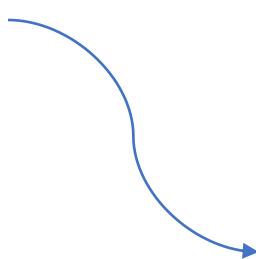
Scaling policy name
Target Tracking Policy

Metric type | [Info](#)
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

Target value
50

Instance warmup | [Info](#)
300 seconds



Average CPU utilization

Search metric types

Average CPU utilization

Average network in (bytes)

Average network out (bytes)

Application Load Balancer request count per target

Custom CloudWatch metric

Autoscaling and Load-balancing of Kubernetes

- **Pod**: the minimum unit represents a single instance of a running process in your cluster
- **Workload**: Kubernetes control plane **automatically manages Pod objects** based on the specification for the workload object you defined
- **Service**: exposes a network application that is running as one or more Pods in your cluster

The **Service** (of type LoadBalancer) provides the load balancing

- distributes incoming traffic to all healthy Pods
- Canary release, A/B testing

The **Horizontal Pod Autoscaler** provides the autoscaling

- tells the workload to adjust the number of Pod replicas
- E-commerce sale event, online game rush hours

