

The screenshot shows the AWS EC2 Dashboard. On the left, a sidebar lists various EC2 services: Global View, Events, Console-to-Code, Instances, Images, Elastic Block Store, Network & Security, and CloudShell. The main area displays a summary of resources in the US East (N. Virginia) Region. It includes a table of running instances, Auto Scaling Groups, Capacity Reservations, Dedicated Hosts, Elastic IPs, Instances, Key pairs, Load balancers, Placement groups, Security groups, Snapshots, and Volumes. Below this, there's a section to launch an instance with buttons for "Launch instance" and "Migrate a server". To the right, the "Service health" section shows the US East (N. Virginia) region is operating normally. The "EC2 Free Tier" section highlights offers for all AWS Regions, noting an end-of-month forecast and exceeding free tier limits. A "Offer usage (monthly)" section shows Linux EC2 Instances and Storage space on EBS both at 0%. At the bottom, a note says "Your instances will launch in the US East (N. Virginia) Region".

Go to the EC2 instance.

The screenshot shows the AWS EC2 Load Balancers page. The sidebar includes options for Images, Elastic Block Store, Network & Security, Load Balancing (with Load Balancers selected), and Auto Scaling. The main content area is titled "Load balancers" and states that elastic load balancing scales capacity automatically. It features a search bar and a table header for Name, DNS name, State, VPC ID, Availability Zones, and Type. Below the table, a message says "No load balancers" and "You don't have any load balancers in us-east-1". A "Create load balancer" button is present. At the bottom, it says "0 load balancers selected" and "Select a load balancer above."

You can see the loadbalancer.

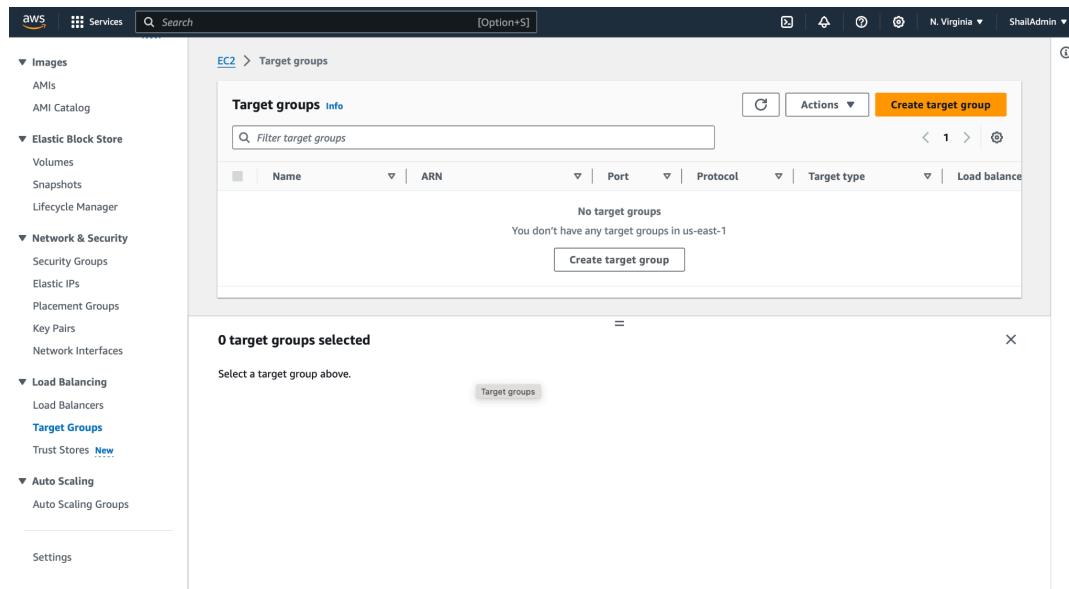
First we need to create a target group.

What is a Target Group?

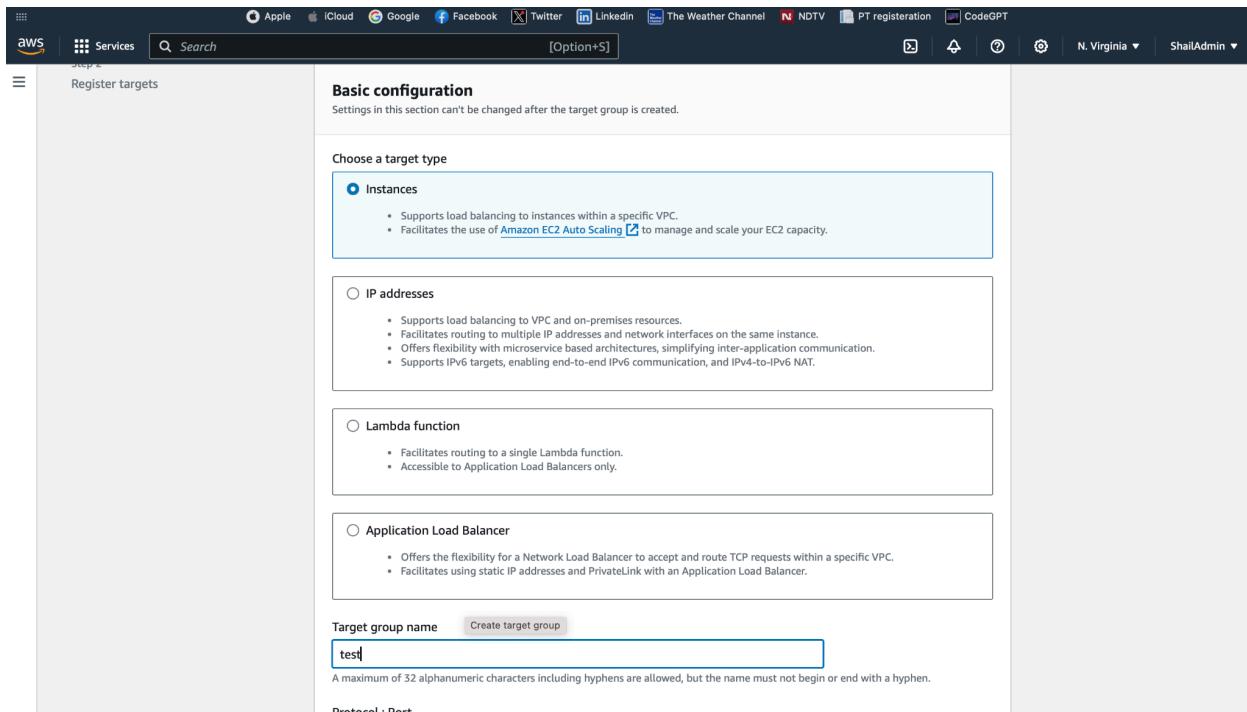
A target group is a logical grouping of targets, such as Amazon EC2 instances, containers, or IP addresses, that you want to route traffic to. It defines how the traffic is distributed and monitored. When a load balancer receives a request, it forwards the request to one of the targets in a target group based on its routing rules and health checks.

Key Concepts

- **Target:** An endpoint that can receive traffic. Targets can be EC2 instances, containers (ECS tasks), IP addresses, or Lambda functions (for ALBs).
- **Health Checks:** Periodic checks to ensure that targets are healthy and able to handle traffic. Unhealthy targets are temporarily removed from the rotation until they recover.
- **Protocol and Port:** The protocol (HTTP, HTTPS, TCP, etc.) and port number that the load balancer uses to communicate with targets.



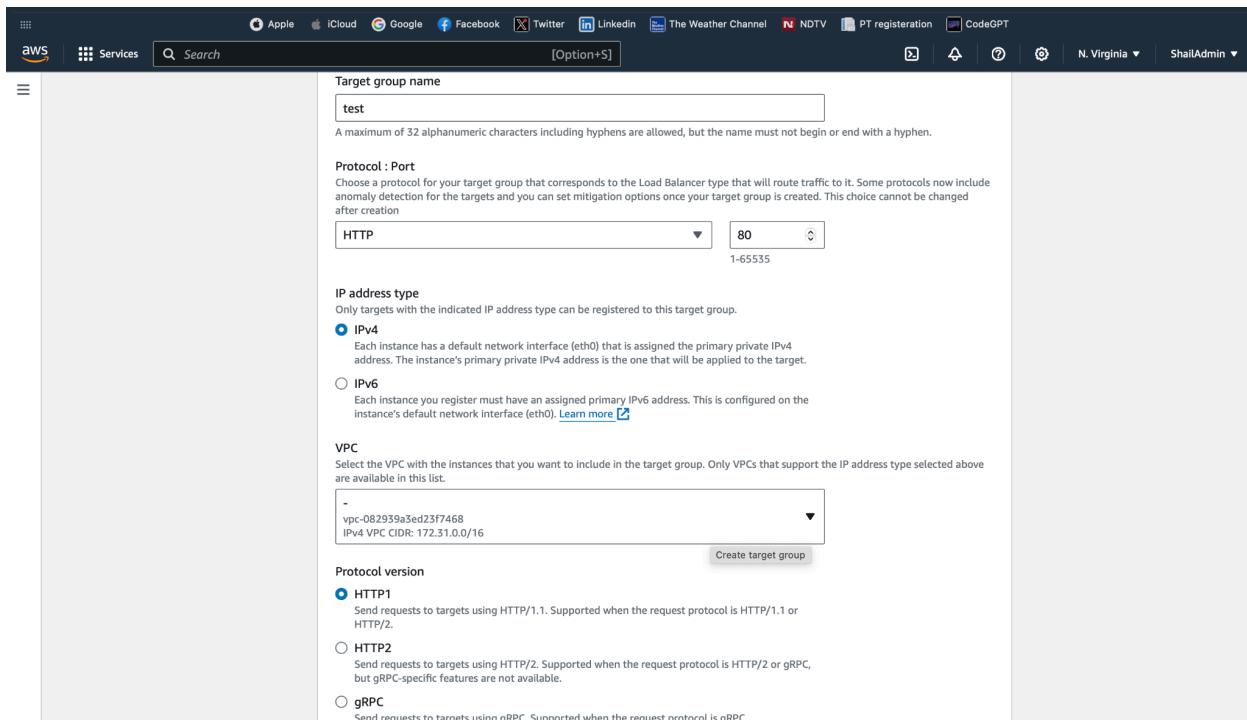
Create Target Group



The screenshot shows the second step of the AWS CloudFormation Create Stack Wizard. The title is "Create stack". The left sidebar shows "CloudFormation" and "Stacks". The main area has a header "Basic configuration" with the note "Settings in this section can't be changed after the stack is created." Below it is a section titled "Choose a target type" with four options:

- Instances
 - Supports load balancing to instances within a specific VPC.
 - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
- IP addresses
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservice based architectures, simplifying inter-application communication.
 - Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.
- Lambda function
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
- Application Load Balancer
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Below the target type section are fields for "Target group name" (containing "test") and "Create target group" button. A note says "A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen." At the bottom is a "Protocol : Port" section with "HTTP" selected and port "80".



This screenshot shows the second step of the AWS CloudFormation Create Stack Wizard, identical to the one above but with different parameter values. The "Target group name" is "test". The "Protocol : Port" section shows "HTTP" selected with port "80". The "IP address type" section shows "IPv4" selected. The "VPC" section shows a dropdown menu with "vpc-082939a3ed23f7468" and "IPv4 VPC CIDR: 172.31.0.0/16". The "Protocol version" section shows "HTTP1" selected. The "Description" field at the bottom is "This stack creates a target group named test." The "Next Step" button is visible at the bottom right.

The screenshot shows the AWS Load Balancer configuration interface. In the 'Protocol version' section, 'HTTP1' is selected. Under 'Health checks', the 'Health check protocol' is set to 'HTTP' and the 'Health check path' is '/'. A note indicates that certain default attributes will be applied to the target group. The 'Attributes' section is also visible.

For health check, by default it is root directory. However if you want to mention you can.

Click on Next

The screenshot shows the 'Register targets' step of the load balancer setup. It displays an empty list of available instances and the ports for selected instances (set to 80). The 'Review targets' section shows an empty list of targets.

Click on Create Target Group

The screenshot shows the AWS EC2 Target Groups page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs), and various other services like Load Balancing, Auto Scaling, and CloudWatch Metrics. The main content area displays a target group named 'test'. The 'Details' section shows the target type as 'Instance', protocol as 'HTTP: 80', and VPC as 'vpc-082939a3ed23f7468'. Below this, a summary table shows 0 total targets, 0 healthy, 0 unhealthy, 0 unused, 0 initial, and 0 draining. The 'Targets' tab is selected. At the bottom, there's a 'Registered targets (0)' section with a 'Filter targets' search bar and buttons for 'Anomaly mitigation: Not applicable', 'Deregister', and 'Register targets'.

Go to loadbalancer and create or we can go to autoscaling groups and create.

The screenshot shows the AWS navigation sidebar. It includes sections for Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The 'Load Balancing' and 'Auto Scaling' sections are expanded, showing their respective sub-links.

Autoscaling group

The screenshot shows the AWS EC2 Auto Scaling landing page. At the top, there's a navigation bar with links to various services like Apple, iCloud, Google, Facebook, Twitter, LinkedIn, The Weather Channel, NDTV, PT registration, and CodeGPT. Below the navigation is a search bar and a user dropdown for 'ShailAdmin'. The main heading is 'Amazon EC2 Auto Scaling' with the subtext 'helps maintain the availability of your applications'. A call-to-action button 'Create Auto Scaling group' is prominent. Below this, a section titled 'How it works' contains a diagram showing an 'Auto Scaling group' containing four instances, with one instance being dashed to indicate scaling. Another section titled 'Pricing' explains that there are no additional fees beyond service fees for EC2, CloudWatch, and other AWS resources. A 'Getting started' link is also present.

Before creating auto scaling we need to create a launch template.

Create custom ami with sample data

The screenshot shows the AWS search interface with the query 'ec2'. The results are categorized under 'Services' and 'Features'. The top result is 'EC2' with a sub-card for 'Top features' including Dashboard, Launch templates, Instances, Spot Instance requests, and Savings plans. Other services listed include EC2 Image Builder, Recycle Bin, and Amazon Inspector.

Create an EC2 instance and host a website.

From there create a custom AMI.

The screenshot shows the 'Launch an instance' wizard. The first step, 'Name and tags', has a 'Name' field set to 'test'. The second step, 'Application and OS Images (Amazon Machine Image)', shows a catalog search bar and a grid of recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A tooltip for the free tier explains it includes 750 hours of t2.micro usage per month, 750 public IPv4 address usages, 30 GB of EBS storage, 2 million I/Os, and 1 GB of snapshots. The final step, 'Summary', shows one instance being launched with the 'Amazon Linux 2023 AMI 2023.5.2...' selected. The 'Launch instance' button is highlighted.

The screenshot shows the AWS Launch Wizard interface for creating a new Amazon Linux 2023.5.2 instance. On the left, under 'Network settings', it shows a VPC (vpc-082939a3ed23f7468) and a subnet (No preference). It includes options for auto-assigning public IP and enabling security groups. A note about additional charges applies when outside of the free tier allowance. Under 'Firewall (security groups)', there are two options: 'Create security group' (selected) and 'Select existing security group'. Below this, it says 'We'll create a new security group called "launch-wizard-1" with the following rules:'.

- Allow SSH traffic from Anywhere (0.0.0.0/0)
- Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

A warning message in a yellow box states: '⚠ Rules with source of 0.0.0.0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

On the right, the 'Summary' section shows 1 instance, the AMI (Amazon Linux 2023.5.2), the instance type (t2.micro), and storage (1 volume(s) - 8 GiB). A callout box highlights the 'Free tier' benefits: 750 hours of t2.micro (or t3.micro in regions where t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and more.

At the bottom, there are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

In advance details write a custom script

The screenshot shows the AWS Launch Wizard interface for creating a new Amazon Linux 2023.5.2 instance. In the 'User data - optional' section, a note says 'will break.' and a warning message in a yellow box says '⚠ User data has already been base64 encoded'. The user data field contains a shell script:

```
#!/bin/bash
yum install httpd -y
systemctl enable httpd
systemctl start httpd
```

On the right, the 'Summary' section shows 1 instance, the AMI (Amazon Linux 2023.5.2), the instance type (t2.micro), and storage (1 volume(s) - 8 GiB). A callout box highlights the 'Free tier' benefits: 750 hours of t2.micro (or t3.micro in regions where t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and more.

At the bottom, there are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

Launch Instance.

Check Status

The screenshot shows the AWS EC2 Instance Details page for an instance with the ID i-0a7c669a416f7b72a. The instance is a t2.micro type running on a private IP (ip-172-31-84-118.ec2.internal). It has an auto-assigned public IP (44.204.22.29) and an IAM Role. The VPC ID is vpc-082939a3ed23f7468. The subnet ID is subnet-01a8b4c483f84bbc5. The instance ARN is arn:aws:ec2:us-east-1:637423339839:instance/i-0a7c669a416f7b72a. The status bar at the bottom indicates "Status checks: 1 passed".

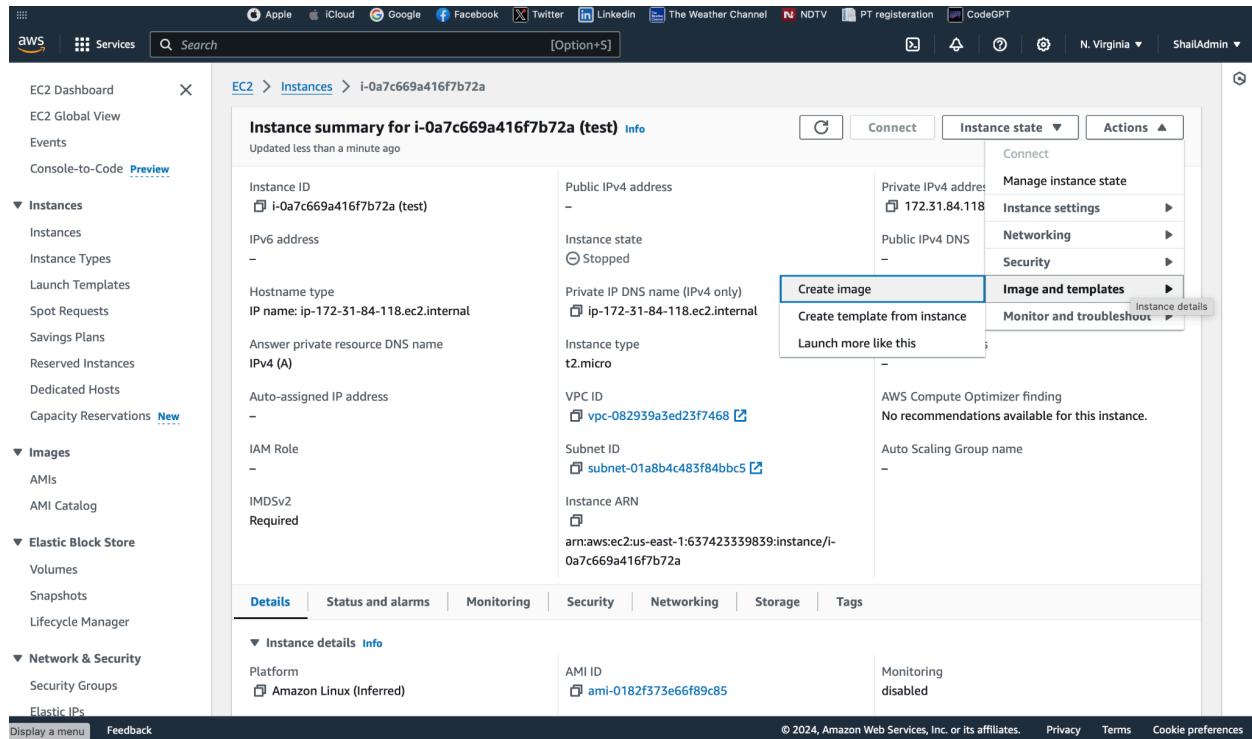
[t works!]

Display a menu

Check with IP.

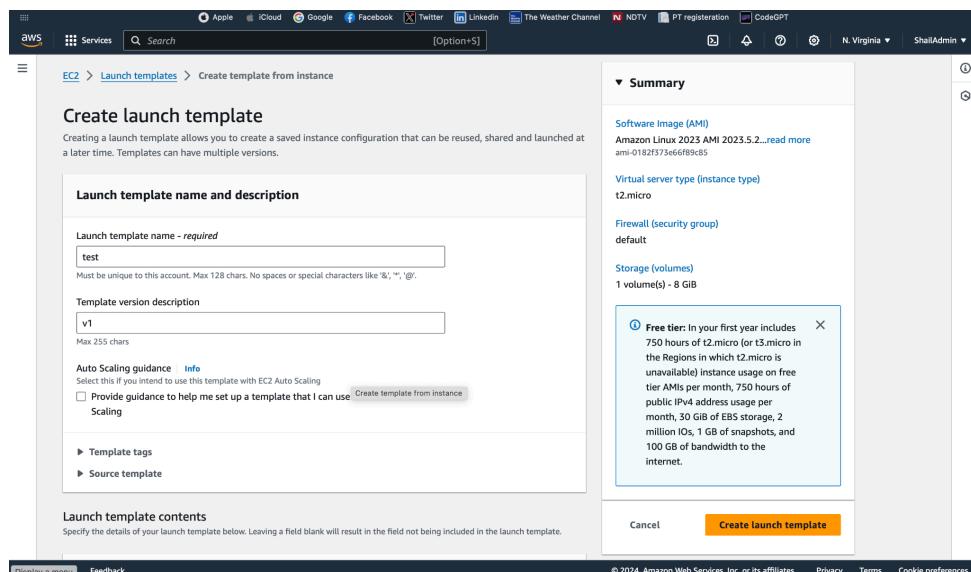
Create an instance from template.

Stop the EC2 instance.



The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with links like EC2 Dashboard, EC2 Global View, Events, and various Instance, Images, and Elastic Block Store categories. The main area displays an instance summary for an instance named i-0a7c669a416f7b72a (test). The instance is currently stopped. A context menu is open on the right, with 'Create image' being the selected option. Other visible options in the menu include 'Manage instance state', 'Instance settings', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'.

Click on Create template from instance.



The screenshot shows the 'Create launch template' wizard. In the 'Summary' step, the user has selected the 'Amazon Linux 2023 AMI 2023.5.2...' software image, the 't2.micro' virtual server type, and '1 volume(s) - 8 GiB' storage. A callout box highlights the 'Free tier' information: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom, there are 'Cancel' and 'Create launch template' buttons.

Select linux instance

AWS CloudFormation console showing the creation of a launch template.

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

[AMI from catalog](#) | [Recents](#) | **Quick Start**

[Don't include in launch template](#) | [Amazon Linux](#) | [macOS](#) | [Ubuntu](#) | [Windows](#) | [Red Hat](#)

[Browse more AMIs](#) | [Create template from instance](#)

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0182f373e66f89c85 (64-bit (x86), uefi-preferred) / ami-0b947c5d5516fa06e (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Summary

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...[read more](#)
ami-0182f373e66f89c85

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) | **Create launch template**

AWS CloudFormation console showing the creation of a launch template.

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

[AMI from catalog](#) | [Recents](#) | **Quick Start**

[Don't include in launch template](#) | [Amazon Linux](#) | [macOS](#) | [Ubuntu](#) | [Windows](#) | [Red Hat](#)

[Browse more AMIs](#) | [Create template from instance](#)

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0182f373e66f89c85 (64-bit (x86), uefi-preferred) / ami-0b947c5d5516fa06e (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Summary

Software Image (AMI)
Amazon Linux 2023 AMI 2023.5.2...[read more](#)
ami-0182f373e66f89c85

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) | **Create launch template**

Select key pair

Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name: gk-macbook [Create new key pair](#)

Network settings [Info](#)

Subnet: [Info](#) Don't include in launch template [Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group [Create security group](#)

Security groups [Info](#)

Select security groups: default sg-0471e51c0a9421150 [X](#) VPC: vpc-082939a3ed23f7468 [Compare security](#) [Create template from instance](#)

[Advanced network configuration](#)

Summary

Software Image (AMI): Amazon Linux 2023.5.2... [read more](#) ami-0182f373e66f89c85

Virtual server type (instance type): t2.micro

Firewall (security group): default

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth to the internet.

[Cancel](#) [Create launch template](#)

Click on Launch templates.

Success Successfully created test(lt-022cc88c1ed107b09).

[Actions log](#)

Next Steps

Launch an instance
With On-Demand Instances, you pay for compute capacity by the second (for Linux, with a minimum of 60 seconds) or by the hour (for all other operating systems) with no long-term commitments or up-front payments. Launch an On-Demand Instance from your launch template.

[Launch instance from this template](#)

Create an Auto Scaling group from your template
Amazon EC2 Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs.

[Create Auto Scaling group](#)

Create Spot Fleet
A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance (of each instance type in each Availability Zone) is set by Amazon EC2, and adjusted gradually based on the long-term supply of and demand for Spot Instances. Spot instances are well-suited for data-analysis, batch jobs, background processing, and optional tasks.

[Create Spot Fleet](#)

[View launch templates](#)

Go to AutoScaling group.

The screenshot shows the AWS EC2 Auto Scaling landing page. On the left, there's a navigation sidebar with sections like Images, AMIs, AMI Catalog, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling (with Auto Scaling Groups selected). The main content area has a title "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below it, a text box explains that Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. To the right, a box titled "Create Auto Scaling group" contains the text "Get started with EC2 Auto Scaling by creating an Auto Scaling group." and a prominent orange "Create Auto Scaling group" button. At the bottom, there are sections for "How it works", "Pricing", and "Getting started".

The screenshot shows the "Create Auto Scaling group" wizard, Step 1: Choose launch template. The left sidebar lists steps from Step 1 to Step 7. The main form is titled "Choose launch template" and includes a "Name" input field where "test-auto" is entered. A note states: "Auto Scaling group name Enter a name to identify the group." Below this is a "Launch template" section with a note: "For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023." It also includes dropdowns for "Launch template" (set to "test") and "Version" (set to "Default (1)").

Selecting subnet will define if you want single az or multi az

The screenshot shows the AWS Step Functions console with the 'Create New State Machine' wizard open. The current step is 'Step 2: Choose instance launch options'. On the left, a sidebar lists optional steps: Step 3 - optional (Configure advanced options), Step 4 - optional (Configure group size and scaling), Step 5 - optional (Add notifications), and Step 6 - optional (Add tags). The main content area shows the 'Instance type requirements' section, which includes a table for a launch template named 'test' (version Default, v1) and an 'Instance type' field set to 't2.micro'. Below this is the 'Network' section, which displays a list of subnets across different availability zones (us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f) with their respective IP ranges and default status. A dropdown menu allows selecting specific availability zones and subnets. At the bottom right of the network section, there is a note about using VPC Lattice.

Click on Next.

Create Loadbalancer.

The screenshot shows the AWS Auto Scaling groups console with the 'Create Auto Scaling group' wizard open. The current step is 'Step 3 - optional: Configure advanced options'. The sidebar lists optional steps: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 - optional (Configure advanced options), Step 4 - optional (Configure group size and scaling), Step 5 - optional (Add notifications), and Step 6 - optional (Add tags). The main content area shows the 'Configure advanced options - optional' section. It includes a 'Load balancing' section with three options: 'No load balancer' (selected), 'Attach to an existing load balancer', and 'Attach to a new load balancer'. Below this is a 'VPC Lattice integration options' section with two options: 'No VPC Lattice service' (selected) and 'Attach to VPC Lattice service'. A note at the bottom explains that VPC Lattice facilitates communication between AWS services.

Screenshot of the AWS Auto Scaling console showing the configuration of a new load balancer.

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Attach to a new load balancer
 Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type
 Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, visit the [Load Balancing console](#).

Application Load Balancer
 HTTP, HTTPS

Network Load Balancer
 TCP, UDP, TLS

Load balancer name
 Name cannot be changed after the load balancer is created.

Create Auto Scaling group

Load balancer scheme
 Scheme cannot be changed after the load balancer is created.

Internal

Internet-facing

Network mapping
 Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
[vpc-082939a3ed23f7468](#)

Display a menu Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS Auto Scaling console showing the configuration of a new load balancer.

Internal

Internet-facing

Network mapping
 Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC
[vpc-082939a3ed23f7468](#)

Availability Zones and subnets
 You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

<input checked="" type="checkbox"/> us-east-1b	<input type="text" value="subnet-01a8b4c483f84bbc5"/>
<input checked="" type="checkbox"/> us-east-1a	<input type="text" value="subnet-0330084cd3e4f88d4"/>
<input type="checkbox"/> us-east-1e	<input type="text" value="Select a subnet"/> Create Auto Scaling group
<input type="checkbox"/> us-east-1f	<input type="text" value="Select a subnet"/>
<input type="checkbox"/> us-east-1c	<input type="text" value="Select a subnet"/>
<input type="checkbox"/> us-east-1d	<input type="text" value="Select a subnet"/>

Listeners and routing
 If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	test HTTP

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

Display a menu Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select target Group.

Health check is chargeable. Please take a note.

The screenshot shows the AWS Auto Scaling configuration interface. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Choose launch template', Step 2 is 'Choose instance launch options', Step 3 is 'Configure advanced options', Step 4 is 'Configure group size and scaling' (which is currently selected), Step 5 is 'Add notifications', and Step 6 is 'Add tags'. Step 7 is 'Review'. The main content area is titled 'Configure group size and scaling - optional'. It includes sections for 'Group size' (with a note about desired capacity type being vCPUs or Memory(GiB) for mixed instances), 'Desired capacity' (set to 1), 'Scaling' (info about resizing manually or automatically), 'Scaling limits' (min and max desired capacity both set to 1), and 'Automatic scaling' (info about target tracking policy). A 'Create Auto Scaling group' button is at the bottom right of the scaling section.

Configure autoscaling parameters.

The screenshot shows the AWS Auto Scaling configuration interface, specifically the 'Instance maintenance policy' step. The sidebar shows steps 1 through 6. The main content area has a note that the group will not dynamically resize to meet demand. It includes a section for 'Instance maintenance policy' with a note about availability during replacement events. It then asks to choose a replacement behavior based on availability requirements, offering four options: 'Mixed behavior' (selected), 'Prioritize availability', 'Control costs', and 'Flexible'. Under 'Mixed behavior', it says 'No policy' (selected) for rebalancing events. Under 'Prioritize availability', it says 'Launch before terminating'. Under 'Control costs', it says 'Terminate and launch'. Under 'Flexible', it says 'Custom behavior'. Below this is a section for 'Instance scale-in protection' with a note about preventing termination. A checkbox for 'Enable instance scale-in protection' is present. At the bottom are 'Cancel', 'Skip to review', 'Previous', and 'Next' buttons, with 'Next' being highlighted in orange. A 'Create Auto Scaling group' button is also at the bottom right.

Click on next > Review. Create Autoscaling Group.

Check from loadbalancer IP.