

Step 1: Launch an EC2 Instance

1. Log in to AWS Management Console:

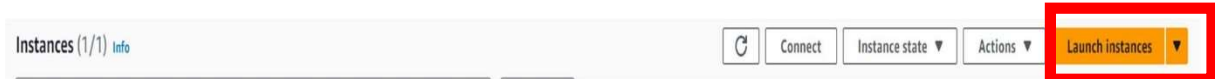
- Go to the AWS Management Console at <https://aws.amazon.com/console/>
- Sign in with your AWS credentials.

2. Navigate to EC2 Dashboard:

- In the AWS Management Console, type "EC2" in the search bar and select EC2 to navigate to the EC2 Dashboard.

3. Launch an Instance:

- Click on the "Launch Instance" button.



- Choose an Amazon Machine Image (AMI): Select "Ubuntu Server 20.04 LTS (HVM), SSD Volume Type".

▼ 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

🔍 Search our full catalog including 1000s of application and OS images

Recents | My AMIs | Quick Start



Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type Free tier eligible ▼
ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

- Choose an Instance Type: Select t2.micro (eligible for the free tier).

▼ Instance type
[Info](#)
[Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

☐ All generations
[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

- Configure Instance:
 - Select an existing key pair or create a new one.
 - Network: Choose the default VPC.
 - Subnet: Choose a subnet in the US-East-1 (N. Virginia) region.
 - Enable Auto-assign Public IP.

▼ Network settings
[Info](#)

Edit

Network [Info](#)

vpc-01f4dd0a574fc4267

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

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.

☒ Create security group
☐ Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

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

- Add Storage: Keep the default settings.
- Add Tags: Add a tag to identify your instance (e.g., Key: Name, Value: Nginx).

4. Review and Launch:

- Review your instance settings and click "Launch".

▼ Summary

Number of instances

[Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 24.04 LTS, ...[read more](#)

ami-04a81a99f5ec58529

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

5. Configure Security Group:

- Add a new security group with the following rules:
 - Type: HTTP, Protocol: TCP, Port Range: 80, Source: 0.0.0.0/0
 - Type: SSH, Protocol: TCP, Port Range: 22, Source: 0.0.0.0/0

i-0c00a91976ab448ec (Nginx)

▼ Inbound rules

| Name | Security group rule ID | Port range | Protocol | Source | Security groups | Description |
|------|------------------------|------------|----------|-----------|---------------------------------|-------------|
| - | sgr-0c96d25f45f8dfbfc | 80 | TCP | 0.0.0.0/0 | launch-wizard-1 | - |
| - | sgr-041906bbea0c8558a | 22 | TCP | 0.0.0.0/0 | launch-wizard-1 | - |

Step 2: Connect to Your Instance

1. **Connect to the EC2 Instance:**

- In the EC2 Dashboard, select your instance.
- Click on "Connect" and follow the instructions to connect to your instance using SSH.

Step 3: Install Apache and PHP

1. **Update the package index:**

```
sudo apt update -y
```

2. **Install Apache:**

```
sudo apt install apache2 -y
```

3. **Start Apache:**

```
sudo systemctl start apache2
```

```
sudo systemctl enable apache2
```

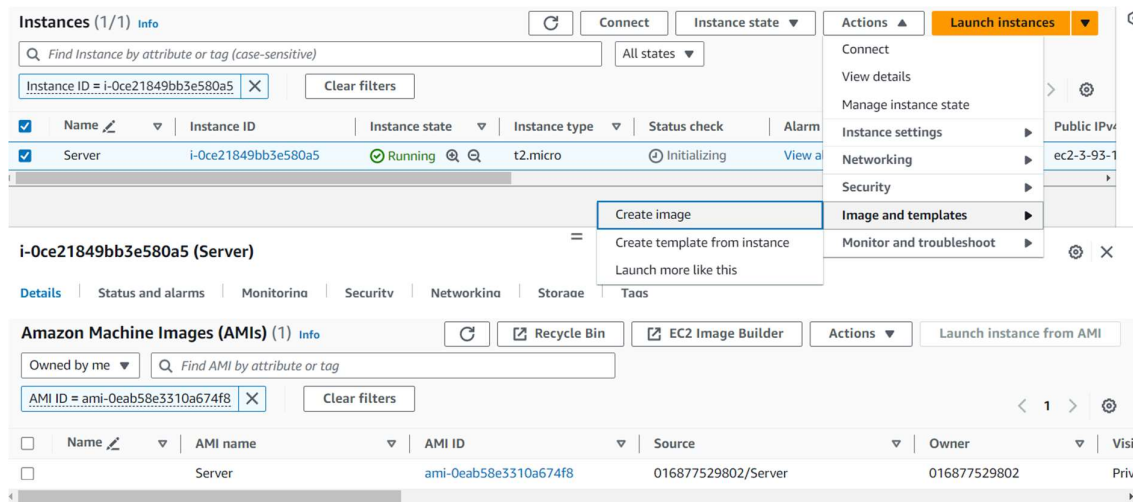
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-39-186:/home/ubuntu# ^[[200~sudo systemctl start apache2~^C
root@ip-172-31-39-186:/home/ubuntu# sudo systemctl start apache2
root@ip-172-31-39-186:/home/ubuntu# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-07-26 03:06:02 UTC; 1min 25s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 2429 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.4M (peak: 5.6M)
        CPU: 41ms
    CGroup: /system.slice/apache2.service
            └─2429 /usr/sbin/apache2 -k start
            └─2432 /usr/sbin/apache2 -k start
            └─2433 /usr/sbin/apache2 -k start
```

4. **Restart Apache:**

```
sudo systemctl restart apache2
```

5. **Create an AMI:**

- After your instance is up and running in US-East-1, go to the EC2 Dashboard, right-click on the instance, and select "Create Image".
- Specify details and create the AMI.



Step 4: Create a Launch Template

1. Navigate to **Launch Templates** in the EC2 dashboard.
2. Click on **Create launch template**.

EC2 > [Launch templates](#) > Create launch template

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

3. Fill in template details and instance configuration.
4. Ensure to use the same AMI, instance type, and security group as your manually launched instance.

| | | | | |
|-----------------------------|----------------------|------------------------|-------------------|------------------|
| <input type="text"/> Search | | | | |
| <input type="checkbox"/> | Launch Template ID ▾ | Launch Template Name ▾ | Default Version ▾ | Latest Version ▾ |
| <input type="checkbox"/> | lt-0bc001d103d2a60b3 | Server | 1 | 1 |

Step 5: Create an Auto Scaling Group:

1. Navigate to **Auto Scaling Groups**.
2. Click on **Create Auto Scaling group**.
3. Choose your launch template.
4. Set the desired capacity to 2, minimum capacity to 1, and maximum capacity to 3.
5. Configure network and subnets.
6. Set up scaling policies (optional).

EC2 > Auto Scaling groups > Server

Server

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details Edit

| | | | |
|---|-----------------------|--|--|
| Auto Scaling group name Server | Desired capacity 1 | Desired capacity type Units (number of instances) | Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:016877529802:autoScalingGroup:82a922b5-d9a0-42ad-808c-b83929040a25:autoScalingGroupName/Server |
| Date created Sat Aug 10 2024 13:45:13 GMT+0530 (India Standard Time) | Minimum capacity 1 | Status - | |
| | Maximum capacity 3 | | |

Launch template Edit

| | | | |
|---|---------------------------------|---------------------------|--|
| Launch template lt-06e014c35a62a66f1 Server | AMI ID ami-0eab58e3310a674f8 | Instance type t2.micro | Owner arn:aws:sts:016877529802:assumed-role/Corestack_Role/shahid199578_gmail |
|---|---------------------------------|---------------------------|--|

Step 6: Create an Application Load Balancer

1. **Navigate to the EC2 Dashboard:**
 - Click on **Load Balancers** under the Load Balancing section.
 - Click on **Create Load Balancer**.
 - Choose **Application Load Balancer**.
 - Configure the load balancer:
 - Name: my-load-balancer.
 - Scheme: Internet-facing.
 - Listeners: HTTP (port 80).
 - Availability Zones: Select the VPC and subnets.

2. **Configure Security Groups** for the load balancer:

- Ensure it allows HTTP traffic.

3. **Configure Routing:**

- Create a target group:
 - Name: my-target-group.
 - Target type: Instances.
 - Protocol: HTTP.
 - Port: 80.
 - Health checks: HTTP.
- Register your instances in the target group.

4. **Configure Routing:**

- Create a new target group.
- Select Instance as the target type.
- Register the same EC2 instances you used with the CLB.

[EC2](#) > [Load balancers](#) > Create Application Load Balancer

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► **How Application Load Balancers work**

Basic configuration

Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)
Scheme can't be changed after the load balancer is created.

☒ **Internet-facing**
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ **Internal**
An internal load balancer routes requests from clients to targets using private IP addresses. Compatible with the IPv4 and Dualstack IP address types.

Load balancer IP address type [Info](#)

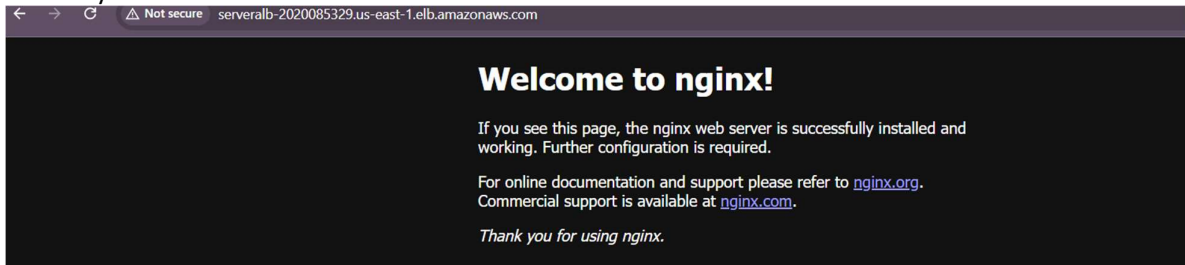
5. **Review and Create** the load balancer.

Step 7: Test the Application Load Balancer

1. **Get the DNS name of the ALB** from the Load Balancers dashboard.

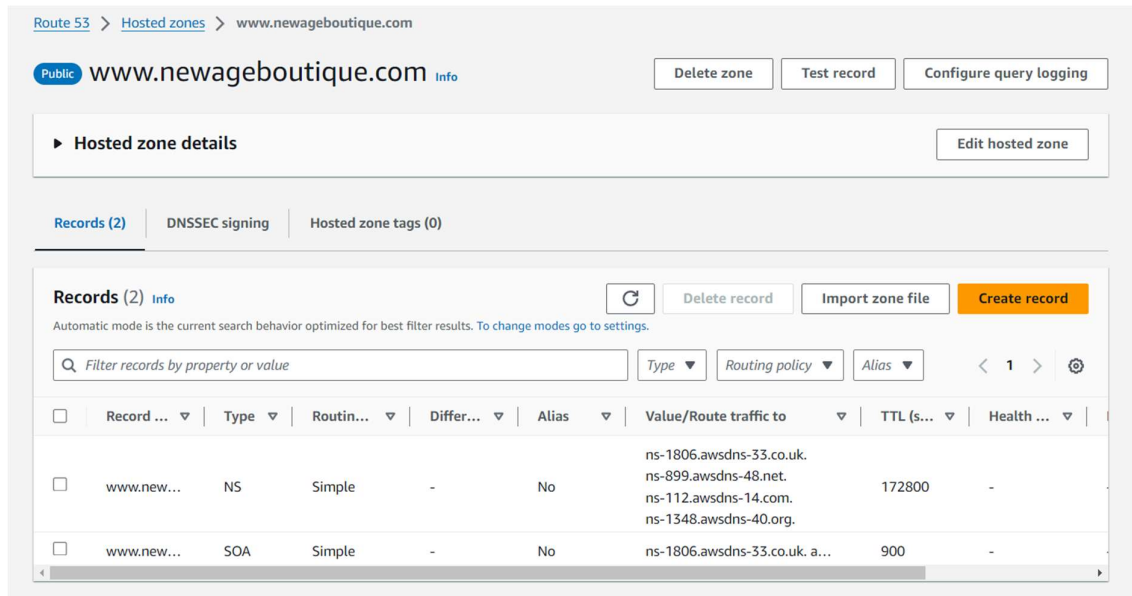
| | |
|---|---|
| Load balancer ARN 📄 <code>arn:aws:elasticloadbalancing:us-east-1:016877529802:loadbalancer/app/ServerALB/c1775d93389d35d8</code> | DNS name Info 📄 <code>ServerALB-2020085329.us-east-1.elb.amazonaws.com (A Record)</code> |
|---|---|

2. Access the DNS name in your web browser to ensure the ALB is routing traffic correctly to your instances.



Steps 8: Route Traffic Using Route 53

- If you're using Route 53, create an A Record pointing to your load balancer.



1. Create an A Record:

- Click on the **Create record** button.
- In the **Record name** field, enter the subdomain or leave it blank for the root domain (e.g., `www` or `@` for the root).
- For **Record type**, select **A** (your load balancer).
- In the **Value** field, enter your load balancer endpoint.
- You can leave the **TTL (Time to Live)** value at its default or set a custom value based on your preferences.

2. Save the Record:

- Click on the **Create records** button to save the new record.