

Day 3

Launch a Load Balancer (ALB) with EC2



www.linkedin.com/in/pravinkr011



<https://github.com/pravinkr011/30-days-aws-projects/>

Name and tags [Info](#)

Name

[Add additional tags](#)

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

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose **Browse more AMIs**.

Recents

Quick Start

Amazon
Linux



macOS



Ubuntu



Windows



Red Hat



SUSE Linux



Debian



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

```
root@ip-172-31-28-97:/home/ubuntu# history
 1  apt update -y
 2  sudo apt install apache2 -y
 3  sudo systemctl start apache2
 4  sudo systemctl enable apache2
 5  sudo systemctl status apache2
 6  history
root@ip-172-31-28-97:/home/ubuntu#
```

Command used to install apache2
Enable apache2



systemctl status apache2

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-28-97:/home/ubuntu# sudo systemctl start apache2
root@ip-172-31-28-97:/home/ubuntu# sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
root@ip-172-31-28-97:/home/ubuntu# sudo 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 2025-09-26 03:37:47 UTC; 54s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 2472 (apache2)
    Tasks: 55 (limit: 1121)
   Memory: 5.2M (peak: 5.3M)
      CPU: 34ms
   CGroup: /system.slice/apache2.service
           └─2472 /usr/sbin/apache2 -k start
             └─2475 /usr/sbin/apache2 -k start
               └─2476 /usr/sbin/apache2 -k start

Sep 26 03:37:47 ip-172-31-28-97 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Sep 26 03:37:47 ip-172-31-28-97 systemd[1]: Started apache2.service - The Apache HTTP Server.
root@ip-172-31-28-97:/home/ubuntu#
```



Configured the index.html in /var/www/html/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Day 2 - AWS Project Completion</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: linear-gradient(to right, #0077b6, #00b4d8);
      color: white;
      text-align: center;
      padding: 50px;
    }
    h1 {
      font-size: 2.5rem;
      margin-bottom: 20px;
    }
    p {
      font-size: 1.2rem;
    }
    .box {
      margin: 30px auto;
      padding: 20px;
      max-width: 600px;
      background: rgba(255, 255, 255, 0.1);
      border-radius: 10px;
      box-shadow: 0px 0px 10px rgba(0,0,0,0.3);
    }
  </style>
</head>
<body>
  <h1>🚀 Day 2 Project Completed!</h1>
  <div class="box">
    <p>Successfully deployed a <span class="highlight">Web Application</span> on <span
class="highlight">AWS EC2</span> using <span class="highlight">Apache
Server</span>.</p>
    <p>This marks the <strong>Day 2 milestone</strong> of my <em>30 Days of AWS
Projects</em> journey.</p>
  </div>
  <footer>
    🌐 Pravin Kumar | DevOps & Cloud Engineer
  </footer>
</body>
</html>
```

contd...

```
.highlight {
  font-weight: bold;
  color: #ffd60a;
}
footer {
  margin-top: 50px;
  font-size: 0.9rem;
  opacity: 0.8;
}
</style>
</head>
<body>
  <h1>🚀 Day 2 Project Completed!</h1>
  <div class="box">
    <p>Successfully deployed a <span class="highlight">Web Application</span> on <span
class="highlight">AWS EC2</span> using <span class="highlight">Apache
Server</span>.</p>
    <p>This marks the <strong>Day 2 milestone</strong> of my <em>30 Days of AWS
Projects</em> journey.</p>
  </div>
  <footer>
    🌐 Pravin Kumar | DevOps & Cloud Engineer
  </footer>
</body>
</html>
```





EC2



Dashboard

EC2 Global View

Events

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

▼ Images

AMIs

AMI Catalog

Amazon EC2 Auto Scaling

helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

Create Auto Scaling group

How it works



Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.



Launch template name *-required*

ALB-WEB-TEMPLATE

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

Template version description

v1.0

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

No template tags are currently applied to this template. Add a template tag to apply it to the launch template.

Add new tag

You can add up to 50 more tags.

▼ Source template

You can optionally specify a source template if you would like to create a template from another existing template

Launch template name



CloudShell

Feedback

© 2025,



Choose instance launch options

Step 3 - optional

Integrate with other services

Step 4 - optional

Configure group size and scaling

Step 5 - optional

Add notifications

Step 6 - optional

Add tags

Step 7

Review

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 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.

ALB-WEB-APP-1



Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

Equal or less than desired capacity

Max desired capacity

Equal or greater than desired capacity

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.

Instance maintenance policy [Info](#)

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.



100.24.7.174



in P...



5. Data Structures...



draw.io



Amazon Web Servic...



DevSecOps and Clo...



Star Agile LMS: Log...



Log In | Intellipaat



Project Completed!

Successfully deployed a **Web Application** on **AWS EC2** using **Apache Server**.

This marks the **Day 3 milestone** of my *30 Days of AWS Projects* journey.



Pravin Kumar | DevOps & Cloud Engineer



www.linkedin.com/in/pravinkr011



<https://github.com/pravinkr011/30-days-aws-projects/>