

Building highly scalable and available infrastructure that handles failure on AWS



 Tej Mandaliya

Introduction

Objective:

This project demonstrates the creation of an Auto Scaling group that automatically adjusts the number of EC2 instances based on demand. The EC2 instances are launched using a pre-configured launch template in a custom VPC.

Technologies used:

AWS EC2, Auto Scaling, AWS ELB, IAM, Launch Templates, VPC (Virtual Private Cloud), Security Groups, and Cloud Watch.

This project took me...

Approximately 1 hour



Tej Mandaliya

How I Set Up

First Creates a Custom VPC

- **VPC Creation:**
 - **CIDR block: 10.0.0.0/16**
 - **2 Public subnets (in different Availability Zones):**
 - **Subnet 1: ap-south-1a, CIDR block: 10.0.1.0/24**
 - **Subnet 2: ap-south-1b, CIDR block: 10.0.2.0/24**
- **Internet Gateway:**
 - **Attach to the VPC to allow external access.**
- **Route Tables:**
 - **Route traffic to the Internet Gateway for public subnets.**

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with navigation links: VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, and Managed prefix lists. The main content area is titled 'Your VPCs (1/2)' and shows a table with two VPCs:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
MyVPC	vpc-003820febae15f86	Available	10.0.0.0/16	-
default	vpc-0b4b4ed485ed946d6	Available	172.31.0.0/16	-

Below the table, there's a section for 'vpc-003820febae15f86 / MyVPC' with tabs for Details, Resource map, CIDRs, Flow logs, Tags, and Integrations. The 'Details' tab is selected, showing a table with columns: VPC ID, State, DNS hostnames, and DNS resolution.

The screenshot shows the AWS Subnets dashboard. On the left, there's a sidebar with navigation links: Subnets dashboard, Global View, Filter by VPC, Virtual private cloud, VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, and Managed prefix lists. The main content area is titled 'Your Subnets (1/2)' and shows a table with two subnets:

Name	Subnet ID	State	VPC
-	subnet-0157c279d5a200cb0	Available	vpc-0b4b4ed485ed946d6 def...
PrivateSubnet	subnet-090646267cbe4e63c	Available	vpc-003820febae15f86 MyVPC
PublicSubnet	subnet-0aaecbd45016e738	Available	vpc-003820febae15f86 MyVPC
-	subnet-0e4e05751f51970e0	Available	vpc-0b4b4ed485ed946d6 def...

Below the table, there's a section for 'Subnets: subnet-090646267cbe4e63c, subnet-0aaecbd45016e738'.

Global View

VPC

Private cloud

Subnets

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main
<input type="checkbox"/>	-	rtb-05da7e95acee15cea	-	-	Yes
<input checked="" type="checkbox"/>	PublicRouteTable	rtb-052b138abb1fb1aca	2 subnets	-	No
<input type="checkbox"/>	-	rtb-027859cada5fa6071	-	-	Yes

Second Step Launch EC2 Instances

Launch an EC2 Instance:

- Choose an Amazon Machine Image (AMI) (e.g., Amazon Linux 2).
- Select an Instance Type (e.g., t2.micro for free tier).
- Click Next: Configure Instance Details.
 - Ensure the instance is in the public subnet.

2. Configure Security Group:

- Create a new security group (or select an existing one).
- Add inbound rules for:
 - HTTP: Port 80 (Type: HTTP, Source: Anywhere or specific IP).
 - HTTPS: Port 443 (Type: HTTPS, Source: Anywhere).
 - SSH: Port 22 (Type: SSH, Source: Your IP).

3. After that Launch Instance.

AWS Services [Alt+S] Mumbai Tej

EC2 Dashboard Instances (1/3) Last updated less than a minute ago Connect Instance state Actions Launch instances

EC2 Global View

Events

Instances

Instances

Instance Types

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input checked="" type="checkbox"/>	Myproj	i-0c6785cb7f1e6e93	Running	t2.micro	2/2 checks passed	View alarms

Set Up Elastic Load Balancer (ELB)

1. Create a Load Balancer:

- In the EC2 dashboard, go to Load Balancers > Create Load Balancer.
- Select Application Load Balancer.
- Name it (e.g., MyLoadBalancer).
- Select Internet-facing and your VPC.
- Choose at least two public subnets.

2. Configure Listeners:

- Set the listener (HTTP on port 80).
- Click Next: Configure Security Settings (you can skip HTTPS)

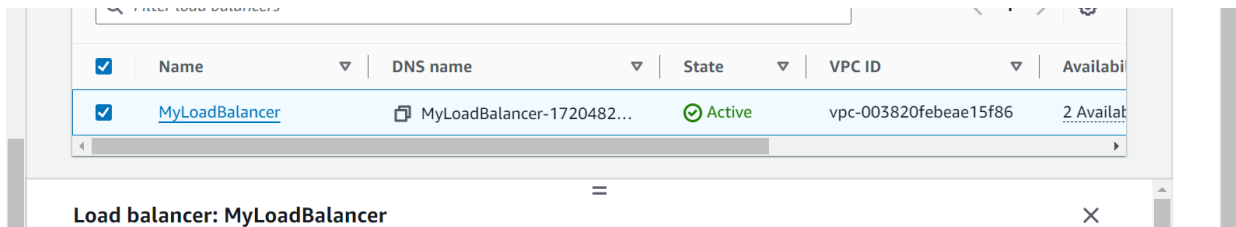
setup for now).

3. Configure Target Groups:

- Create a new target group (e.g., MyTargetGroup):
 - Target type: Instance.
 - Protocol: HTTP.
 - Port: 80.
- Click Next: Register Targets.
- Register your EC2 instance by selecting it and clicking Add to registered.

4. Create Load Balancer:

- Review your configuration and click Create Load Balancer.



Implement Auto Scaling

1. Created an Auto Scaling Group:

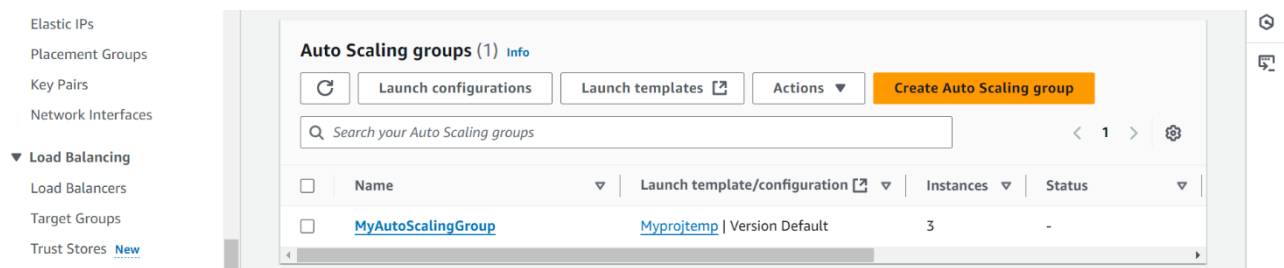
- Created Auto Scaling Group (MyAutoScalingGroup).
- Selected the VPC and at least two subnets (one public, one private).

2. Configure Group Settings:

- Set Desired capacity to 1, Minimum capacity to 1, and Maximum capacity to 3.

3. Configure Scaling Policies:

- Add scaling policies (e.g., scale out if CPU utilization > 70% and scale in if < 30%).



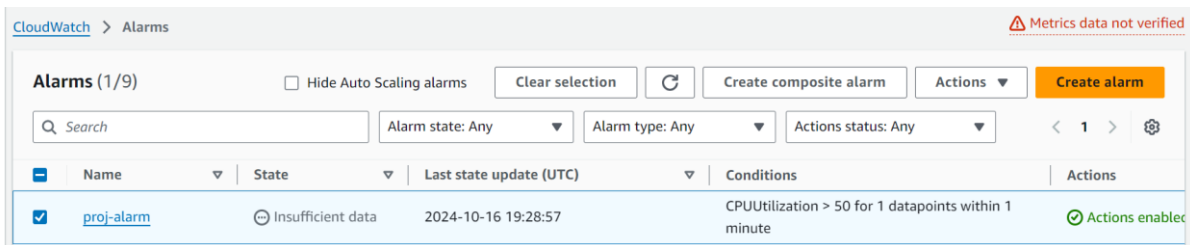
Created Monitor Your Infrastructure with CloudWatch

1. CloudWatch Alarms:

- Click on Alarms > Create Alarm.
- Selected metric and choose EC2 metrics.
- Choose Per-Instance Metrics and selected my instance.
- Set conditions (e.g., whenever CPU Utilization is greater than 70%).
- Configure actions (send a notification to an SNS topic).

2. Create Dashboards:

- We can create dashboards to visualize your metrics. Click on Dashboards > Create dashboard and select the metrics we want to monitor.



The screenshot shows the AWS CloudWatch Alarms console. At the top, there's a breadcrumb 'CloudWatch > Alarms' and a red warning 'Metrics data not verified'. Below this is a header for 'Alarms (1/9)' with a 'Hide Auto Scaling alarms' checkbox, 'Clear selection', a refresh icon, 'Create composite alarm', an 'Actions' dropdown, and a 'Create alarm' button. A search bar and filters for 'Alarm state: Any', 'Alarm type: Any', and 'Actions status: Any' are present. The main table has columns: Name, State, Last state update (UTC), Conditions, and Actions. One alarm is listed: 'proj-alarm' with a state of 'Insufficient data', a last update of '2024-10-16 19:28:57', and a condition of 'CPUUtilization > 50 for 1 datapoints within 1 minute'. The 'Actions' column shows 'Actions enabled' with a green checkmark.

Name	State	Last state update (UTC)	Conditions	Actions
<input checked="" type="checkbox"/> proj-alarm	Insufficient data	2024-10-16 19:28:57	CPUUtilization > 50 for 1 datapoints within 1 minute	Actions enabled

Last Step to Steup Git for version Control.



