

Project Statement :

Implementing High Availability with Auto Scaling: **Description:** Configure an application to automatically scale based on demand to ensure high availability and performance. **Implementation:** Set up an Auto Scaling group and configure scaling policies based on metrics like CPU utilization or request rate. Use Elastic Load Balancer to distribute traffic across instances.

I'm thrilled to share the successful completion of my Project

Here's a brief overview of my journey:

Create an Auto Scaling Group:

Navigate to the Auto Scaling service. Create an Auto Scaling group, specifying the desired instance type, minimum and maximum number of instances, and the availability zones where instances should be launched

Configure Scaling Policies:

Define scaling policies that will trigger the Auto Scaling group to add or remove instances based on specific metrics like CPU utilization, request rate, or custom application metrics.

Set Up Amazon CloudWatch Alarms:

Create CloudWatch alarms to monitor the metrics specified in your scaling policies.

These alarms will trigger scaling actions when the defined thresholds are breached.

Configure Scaling Triggers:

Associate your scaling policies with the CloudWatch alarms to create scaling triggers.

This will tell Auto Scaling when to perform scaling actions.

Create an Elastic Load Balancer (ELB):

Set up an Elastic Load Balancer to distribute incoming traffic across your instances.

Configure the ELB to perform health checks on your instances and route traffic only to healthy instances.

Attach Auto Scaling Group to the Load Balancer:

Attach your Auto Scaling group to the Elastic Load Balancer.

This ensures that traffic is distributed evenly across all instances launched by the Auto Scaling group.

Testing and Monitoring:

Test your setup to ensure that instances are scaling in and out correctly based on demand.

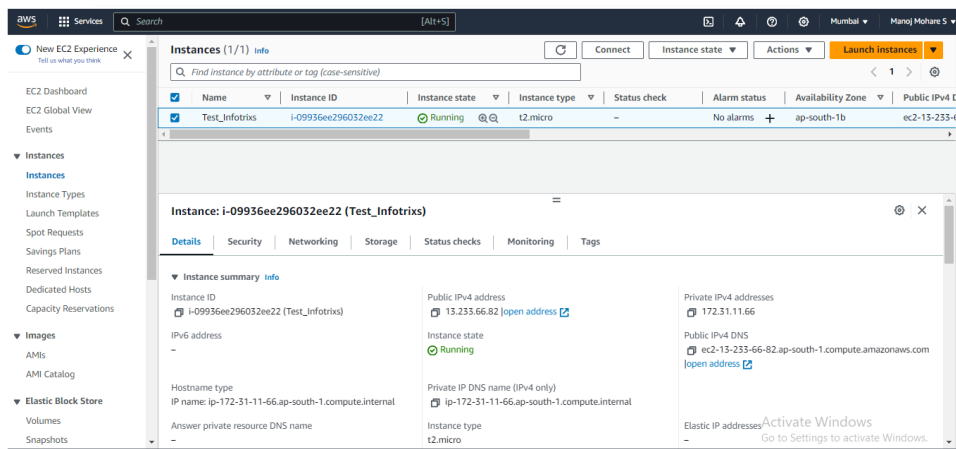
Continuously monitor your application's performance, Auto Scaling events, and CloudWatch metrics to make adjustments as needed.

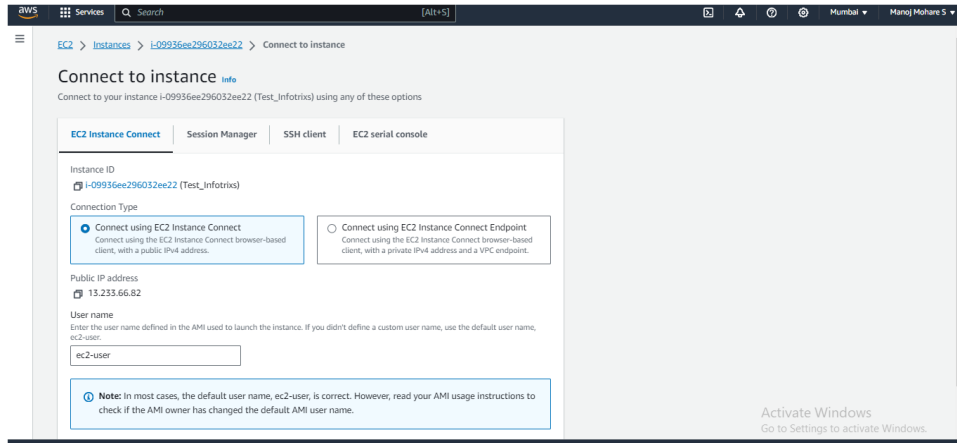
Regular Maintenance:

Periodically review and adjust your scaling policies and alarms to optimize performance and cost.

Keep your application and infrastructure up to date with the latest patches and security updates.

Create a instance





```
Installing      : httpd-core-2.4.56-1.amzn2023.x86_64          7/12
Installing      : mod_lua-2.4.56-1.amzn2023.x86_64           8/12
Installing      : mod_http2-2.0.11-2.amzn2023.x86_64         9/12
Installing      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 10/12
Installing      : libbrotli-1.0.9-4.amzn2023.0.2.x86_64       11/12
Installing      : httpd-2.4.56-1.amzn2023.x86_64             12/12
Running scriptlet: httpd-2.4.56-1.amzn2023.x86_64           12/12
Verifying       : mod_lua-2.4.56-1.amzn2023.x86_64           1/12
Verifying       : httpd-core-2.4.56-1.amzn2023.x86_64        2/12
Verifying       : httpd-2.4.56-1.amzn2023.x86_64            3/12
Verifying       : apr-util-1.6.3-1.amzn2023.0.1.x86_64        4/12
Verifying       : mod_http2-2.0.11-2.amzn2023.x86_64         5/12
Verifying       : apr-1.7.2-2.amzn2023.0.2.x86_64            6/12
Verifying       : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 7/12
Verifying       : httpd-tools-2.4.56-1.amzn2023.x86_64       8/12
Verifying       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64       9/12
Verifying       : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 10/12
Verifying       : httpd-filesystem-2.4.56-1.amzn2023.noarch   11/12
Verifying       : mailcap-2.1.49-3.amzn2023.0.3.noarch        12/12

Installed:
apr-1.7.2-2.amzn2023.0.2.x86_64          apr-util-1.6.3-1.amzn2023.0.1.x86_64      apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  httpd-2.4.56-1.amzn2023.x86_64          httpd-core-2.4.56-1.amzn2023.x86_64
httpd-filesystem-2.4.56-1.amzn2023.noarch  httpd-tools-2.4.56-1.amzn2023.x86_64    libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch      mod_http2-2.0.11-2.amzn2023.x86_64      mod_lua-2.4.56-1.amzn2023.x86_64

Complete!
root@ip-172-31-11-66 ~#
```

i-09936ee296032ee22 (Test_Infotrix) X

PublicIPs: 13.233.66.82 PrivateIPs: 172.31.11.66

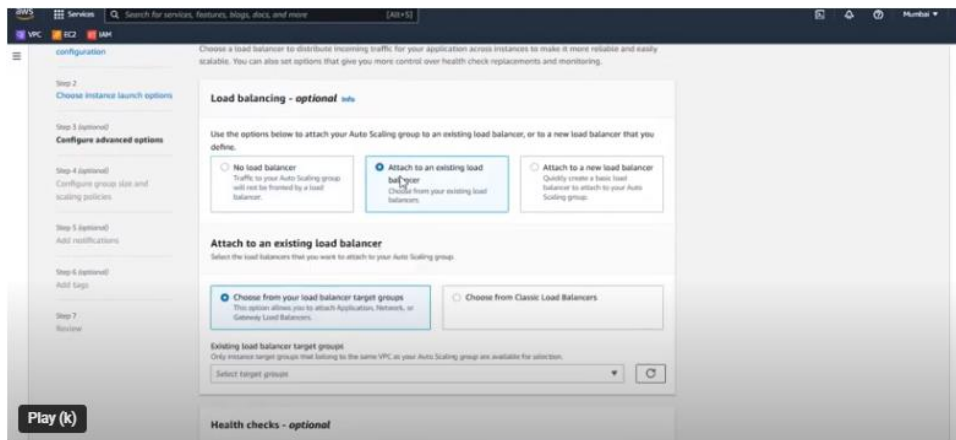
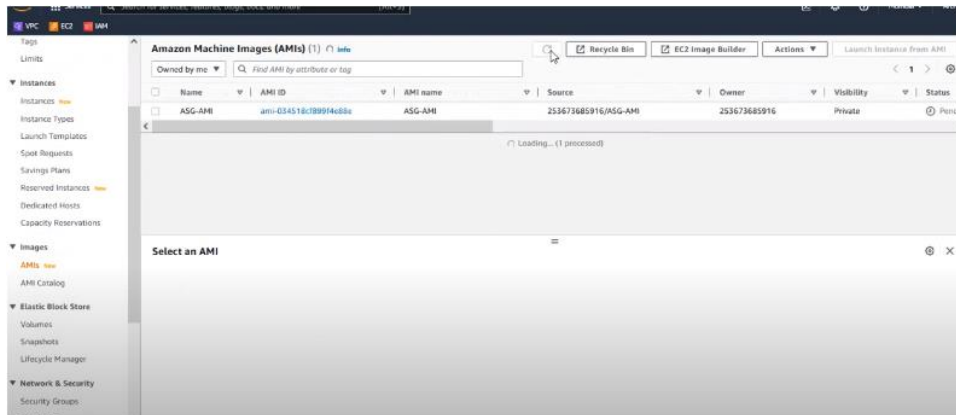
```

AWS  Services  Search  [Alt+S]  Mumbai  Manoj Mohare 5
[root@ip-172-31-11-66 ec2-user]# yum install httpd -y
Last metadata expiration check: 1:39:41 ago on Thu Sep 28 03:23:06 2023.
Package httpd-2.4.56-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-11-66 ec2-user]# systemctl start httpd
[root@ip-172-31-11-66 ec2-user]#

```

i-09936ee296032ee22 (Test_Infortrix)
PublicIPs: 13.233.66.82 PrivateIPs: 172.31.11.66

It works!



Capacity Reservation

Images

- AMI
- 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

Auto Scaling

- Launch Configurations
- Auto Scaling Groups

Create dynamic scaling policy

Policy type: Step scaling

Scaling policy name:

CloudWatch alarm: Choose an alarm that can scale capacity whenever

[Create a CloudWatch alarm](#)

Take the action:

Add ▼

20 ▼ capacity units ▼

[Add step](#)

Instances need: 100 seconds warm up before including in metric

[Cancel](#) [Create](#)

```

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 16 available
run "sudo yum update" to apply all updates.
ec2-user@ip-172-31-11-48 ~$
ec2-user@ip-172-31-11-48 ~$
ec2-user@ip-172-31-11-48 ~$ sudo su -
Last login: Sun Oct 30 08:37:01 UTC 2022 on pts/0
root@ip-172-31-11-48 ~#
root@ip-172-31-11-48 ~#
root@ip-172-31-11-48 ~#
root@ip-172-31-11-48 ~# yum install stress
loaded plugins: extras_suggestions, langpacks, priorities, update-motd
30 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package stress.x86_64 0:1.0.4-16.el7 will be installed
--> Finished Dependency Resolution

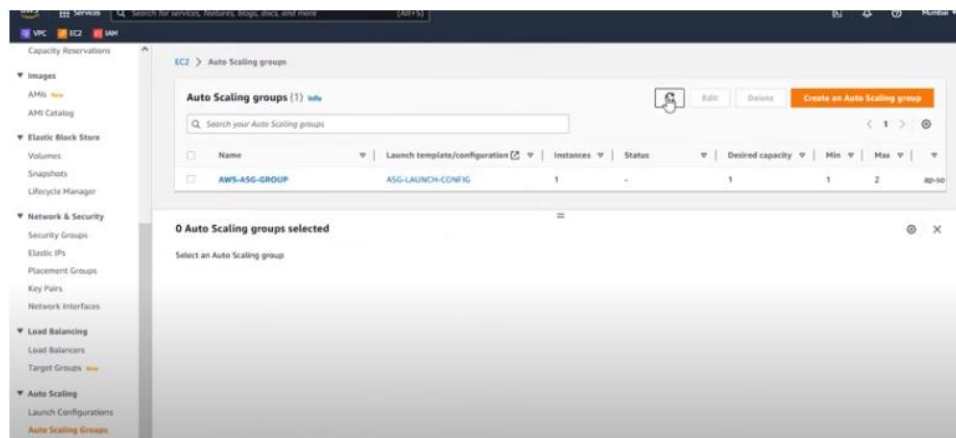
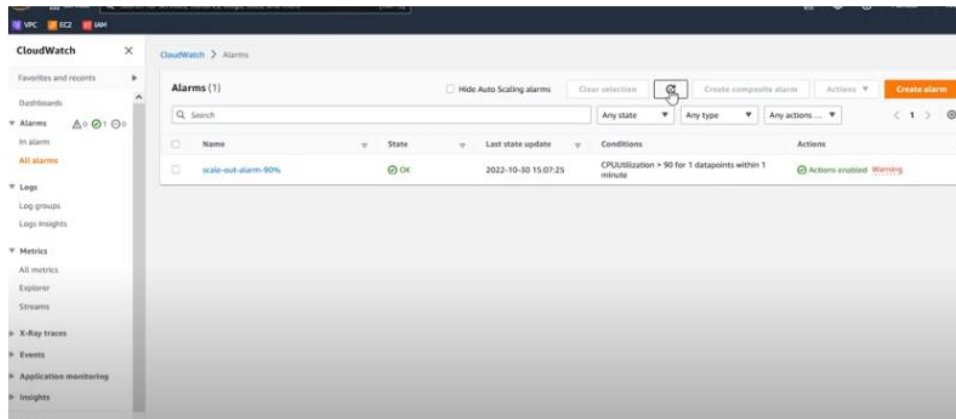
Dependencies Resolved

Package Arch Version Repository
Installing:
stress x86_64 1.0.4-16.el7 epel

Transaction Summary
Install 1 Package

Total download size: 39 k
Installed size: 84 k
Is this ok [y/d/N]: y
Downloading packages!
warning: /var/cache/yum/x86_64/epel/packages/stress-1.0.4-16.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID 352064e5: NOKEY
While key for stress-1.0.4-16.el7.x86_64.rpm is not installed
stress-1.0.4-16.el7.x86_64.rpm
Extracting key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-7 | 39 kB 00:00:00
Importing GPG key 0x352064E5:
Unread 1 "Fedora EPEL (7) epel@fedoraproject.org"

```



EC2 > Auto Scaling groups > AWS-ASG-GROUP

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

Activity notifications (0)

Filter notifications

Send to | On instance action

No notifications are currently specified

Create notification

Activity history (2)

Filter activity history

Status	Description	Cause	Start time	End time
PreInService	Launching a new EC2 instance i-0F520964c413e9f	At 2022-10-30T09:47:00Z a monitor alarm scale-out-alarm-90% in state ALARM triggered policy scale-out policy changing the desired capacity from 1 to 2. At 2022-10-30T09:47:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2022 October 30, 09:17:08 PM +05:30	
InService	Terminating a new EC2 instance i-0A555-5LW40W-11-162P	A monitor alarm scale-in-alarm-10% in state ALARM triggered policy scale-in policy changing the desired capacity from 2 to 1. At 2022-10-30T09:47:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2022 October 30, 09:17:08 PM +05:30	

CloudWatch > Alarms > Create alarm

Step 1: Specify metric and conditions

Step 2: Configure actions

Step 3: Add name and description

Step 4: Preview and create

Specify metric and conditions

Metric

Graph

Preview of the metric or metric expression and the alarm threshold.

Select metric

Cancel Next

