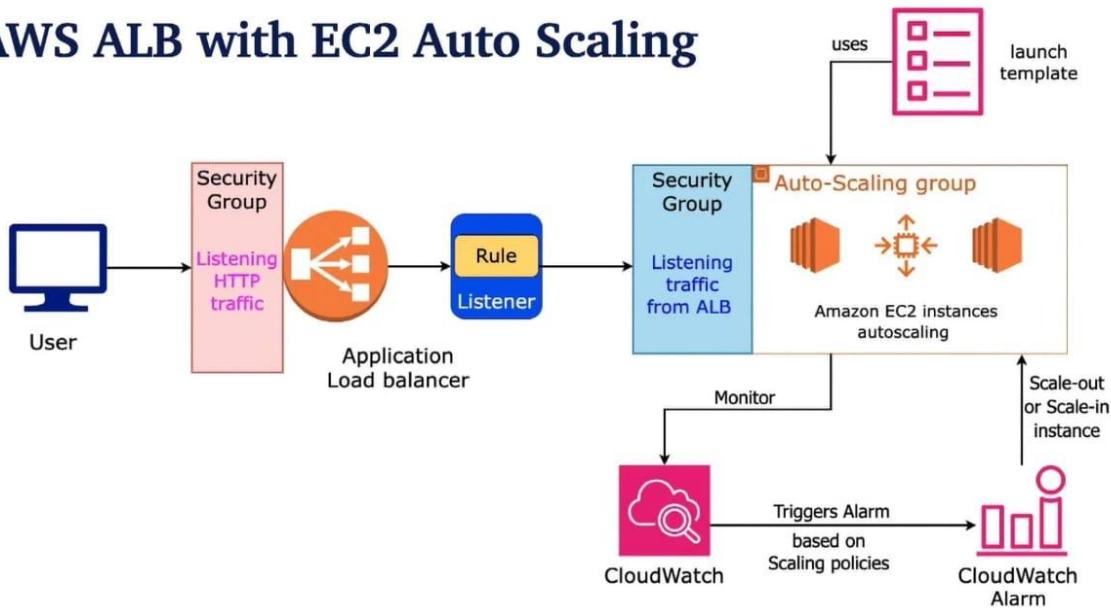


# Project

## AWS ALB with EC2 Auto Scaling



## Auto-Scaling and Load-Balancing in AWS EC2 Using Templates

22.11.2023

Himanshu Nimje  
Nagpur-440017

## Overview

Auto-Scaling and Load-Balancing in AWS EC2 using templates involve setting up a scalable and resilient infrastructure to handle varying levels of traffic. This is achieved through the use of Amazon EC2 instances, Auto Scaling groups, and an Elastic Load Balancer (ELB). Additionally, AWS CloudWatch is used to automate the deployment and configuration of these resources.

## Benefits

1. Scalability:- Auto-scaling:- Templates allow you to define auto-scaling groups that automatically adjust the number of EC2 instances based on demand. This ensures that your application can handle varying workloads efficiently.
2. High Availability:- Load balancing:- By using load balancers, incoming traffic is distributed across multiple instances. In the event of an instance failure, the load balancer redirects traffic to healthy instances, minimizing downtime and enhancing the overall availability of your application.
3. Cost Optimization:- Auto-scaling:- With auto-scaling, you can dynamically adjust the number of instances based on demand. This means you only pay for the resources you actually need, helping to optimize costs and avoid over-provisioning.
4. Fault Tolerance:- Auto-scaling and Load balancing:- Both mechanisms contribute to fault tolerance. Auto-scaling ensures that new instances are launched if existing ones fail, and load balancing redirects traffic away from unhealthy instances, preventing them from affecting the overall application performance.
5. Improved Performance:- Load balancing:- Distributing incoming traffic across multiple instances improves the overall performance of your application by preventing individual instances from being overwhelmed.
6. Improved resource utilization, cost-effectiveness, scalability, high availability, and easy management.

## Service Used

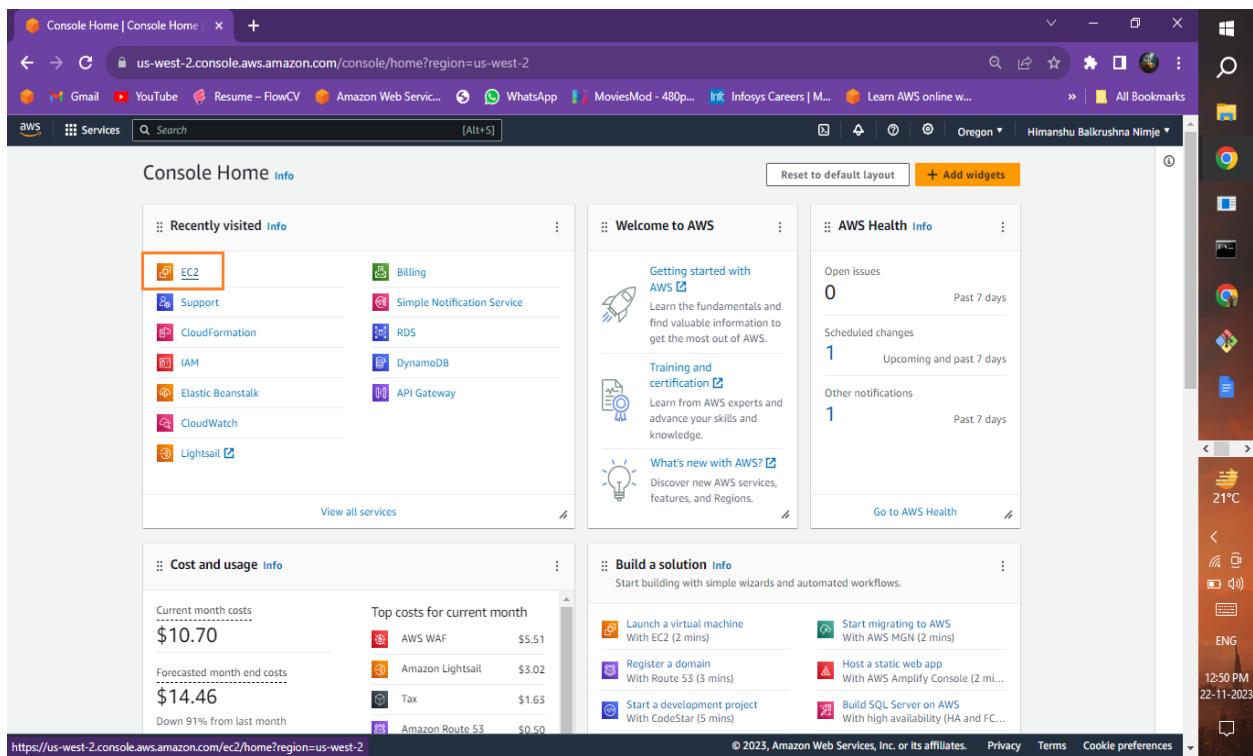
1. Auto-Scaling,
2. Load- Balancing
3. EC2 Templates
4. AWS CloudWatch
5. AWS Cloudshell

## Steps

1. Create a Security Groups
2. Create an Auto Scaling Group
3. Create a launch template
4. Define Launch Configuration:- Specify the launch configuration for instances
5. Configure Load Balancing (Add Elastic Load Balancers)
6. Set Up Auto Scaling Policy
7. Create A cloudWatch Alarm
8. Cleaning

## Implementation↓

Go to AWS Console And Select EC2



## Create a Security Group For Load-Balancer and Auto-Scaling,

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar is collapsed, and the main area displays the 'Security Groups (1) Info' table. The table has one row with the following data:

| Name | Security group ID   | Security group name | VPC ID                | Description                |
|------|---------------------|---------------------|-----------------------|----------------------------|
| -    | sg-024ea13fdb4995b3 | default             | vpc-0cfdc8f994cb5dabe | default VPC security group |

At the top right of the table, there is a 'Create security group' button, which is highlighted with a yellow box. The status bar at the bottom right shows the date and time as 22-11-2023 12:51 PM.

Naming the security group for Load-Balancer

Select VPC

Add Inbound Rule → Select (HTTP) and Click on Create

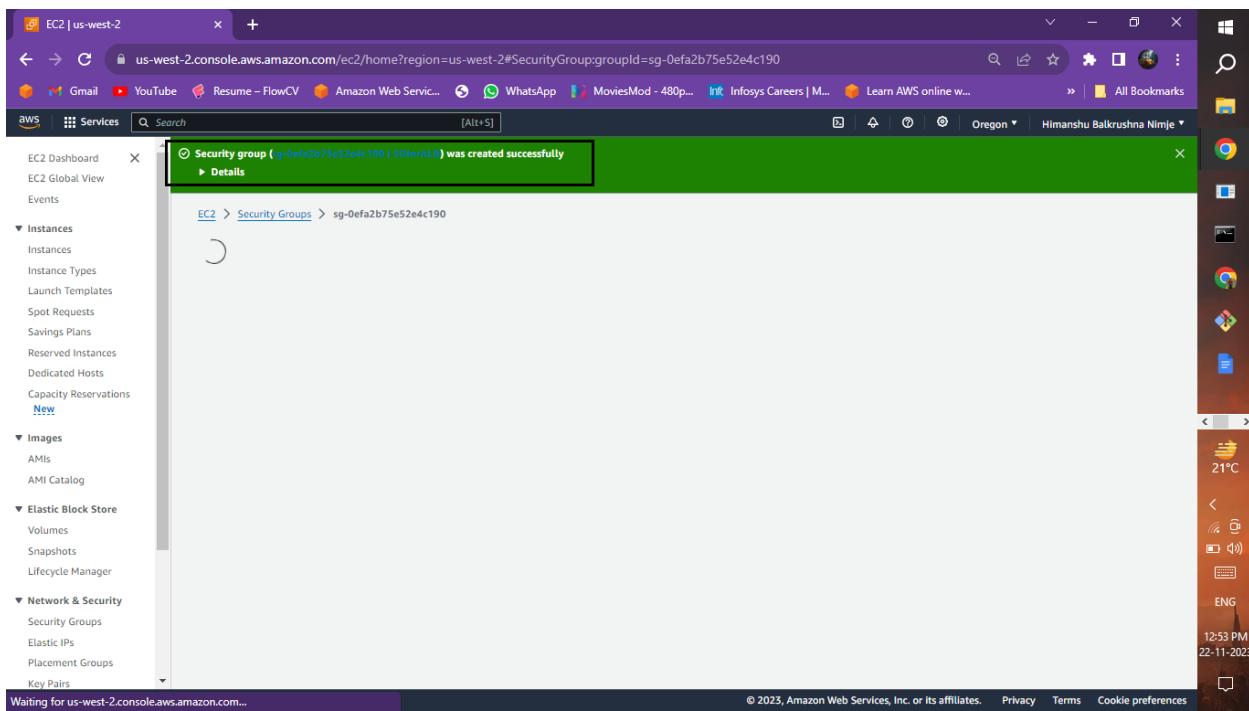
The screenshot shows the 'Create security group' wizard. The first step, 'Basic details', is completed with the following information:

- Security group name info:** SGforALB
- Description info:** SGforALB
- VPC info:** vpc-0cfdc8f994cb5dabe

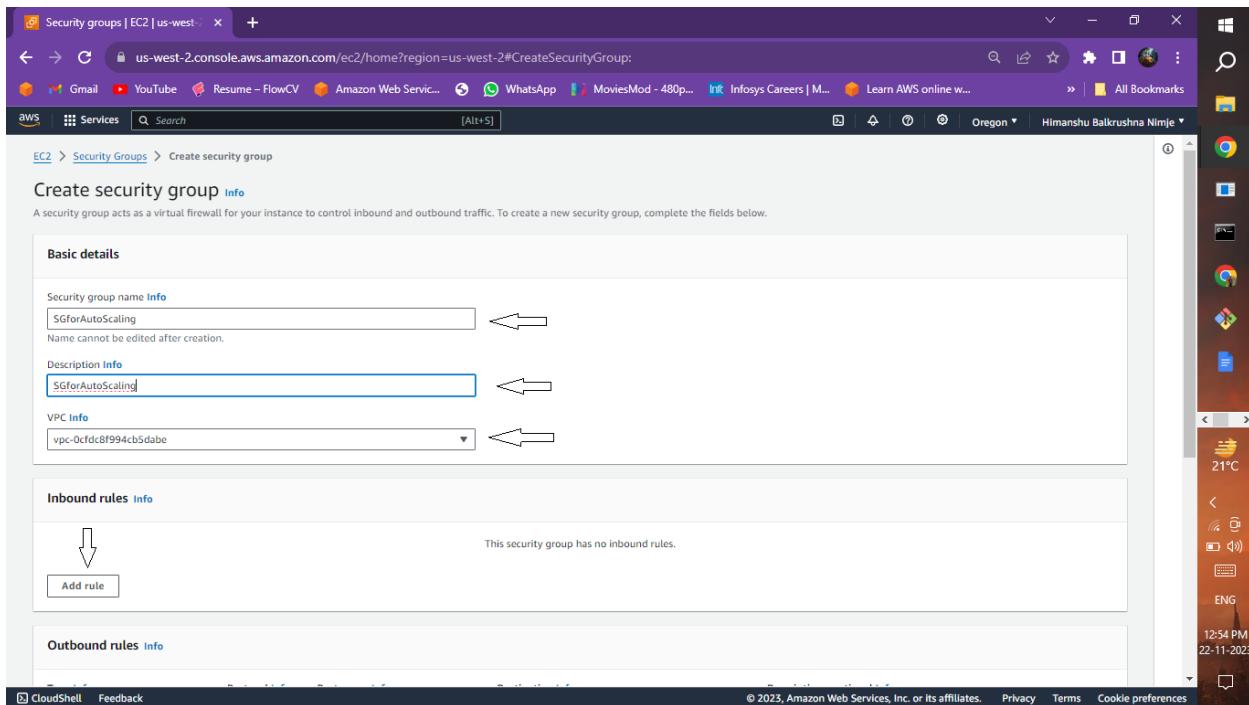
The second step, 'Inbound rules info', is partially completed with the following settings:

| Type Info | Protocol Info | Port range Info | Source Info | Description - optional Info |
|-----------|---------------|-----------------|-------------|-----------------------------|
| HTTP      | TCP           | 80              | Custom      |                             |

A 'Delete' button is visible next to the port range entry. Below the table, there is an 'Add rule' button.



## Create security group for Auto-Scaling, Naming a security group name



Add Inbound Rule → select (All TCP) and (SSH)

The screenshot shows the AWS EC2 Security Groups creation interface. In the 'Inbound rules' section, there are two entries:

- Type Info:** All TCP → TCP → Port range info: 0 - 65535 → Source Info: Custom → sg-0efa2b75e52e4c190 → Description: (empty) → Delete
- Type Info:** SSH → TCP → Port range info: 22 → Source Info: Anywhere... → Description: (empty) → Delete

At the bottom right of the window, there is a status bar showing 'CloudShell' and 'Feedback'.

And Click on the Create

The screenshot shows the AWS EC2 Security Groups creation interface. At the bottom right, there is a large orange button labeled 'Create security group'.

Other visible sections include:

- Outbound rules:** Type Info: All traffic → Protocol Info: All → Port range Info: All → Destination Info: Custom → 0.0.0.0/0 → Description: (empty) → Delete
- Tags - optional:** A note states: 'A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.' Below it, it says 'No tags associated with the resource.' and 'Add new tag'.

## Creating Auto-Scaling Group, Click on Create

The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a sidebar with various AWS services like Spot Requests, Savings Plans, Reserved Instances, etc. The 'Auto Scaling Groups' option is selected and highlighted with an orange box. The main content area has a title 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. Below it, a diagram titled 'How it works' illustrates an 'Auto Scaling group' with four squares representing instances. Labels indicate 'Minimum size' (two solid squares), 'Desired capacity' (three solid squares), 'Scale out as needed' (dashed square), and 'Maximum size' (four solid squares). To the right, there are sections for 'Pricing' and 'Getting started'. The 'Create Auto Scaling group' button at the top right of the main content area is also highlighted with an orange box.

## Assign a Name (HNDemoAutoScaling), click on the create a launch Template

The screenshot shows the 'Choose launch template' step of the 'Create Auto Scaling group' wizard. On the left, a sidebar lists steps from 'Step 1: Choose launch template' to 'Step 7: Review'. The 'Step 1' section is active. The main form has a 'Name' field where 'HNDemoAutoScaling' is entered, and a 'Launch template' dropdown menu where 'Select a launch template' is chosen. Both the 'Name' field and the 'Launch template' dropdown are highlighted with orange arrows. At the bottom right, there are 'Cancel' and 'Next' buttons, with 'Next' being highlighted with an orange box.

## Assign a name (Test Template)

The screenshot shows the 'Create launch template' wizard on the AWS Management Console. The current step is 'Launch template name and description'. The 'Launch template name - required' field contains 'TestTemplate'. A tooltip indicates that the name must be unique to this account and can't contain spaces or special characters like '&', '\*', '@'. The 'Template version description' field contains 'A prod webserver for MyApp'. Under 'Auto Scaling guidance', the 'Info' checkbox is checked. The 'Free tier' information is displayed in a callout box: '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, 30 GiB of EBS storage, 2 million IOPS, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' The 'Create launch template' button is at the bottom right.

## Select OS(Amazon Linux)

The screenshot shows the 'Create launch template' wizard on the AWS Management Console. The current step is 'Application and OS Images (Amazon Machine Image) - required'. The 'Quick Start' section highlights 'Amazon Linux' (which is also highlighted with a red box). Other options shown include macOS, Ubuntu, Windows, Red Hat, and SUSE. A tooltip for 'Amazon Machine Image (AMI)' states: 'Amazon Linux 2023 AMI ami-093467ec28ae4fe03 (64-bit (x86)) / ami-058a0afa5f1acc577 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs'. The 'Free tier eligible' status is indicated. The 'Summary' section on the right shows the selected 'Software Image (AMI)': 'Amazon Linux 2023.2.2...read more ami-093467ec28ae4fe03'. The 'Free tier' information is identical to the previous screenshot. The 'Create launch template' button is at the bottom right.

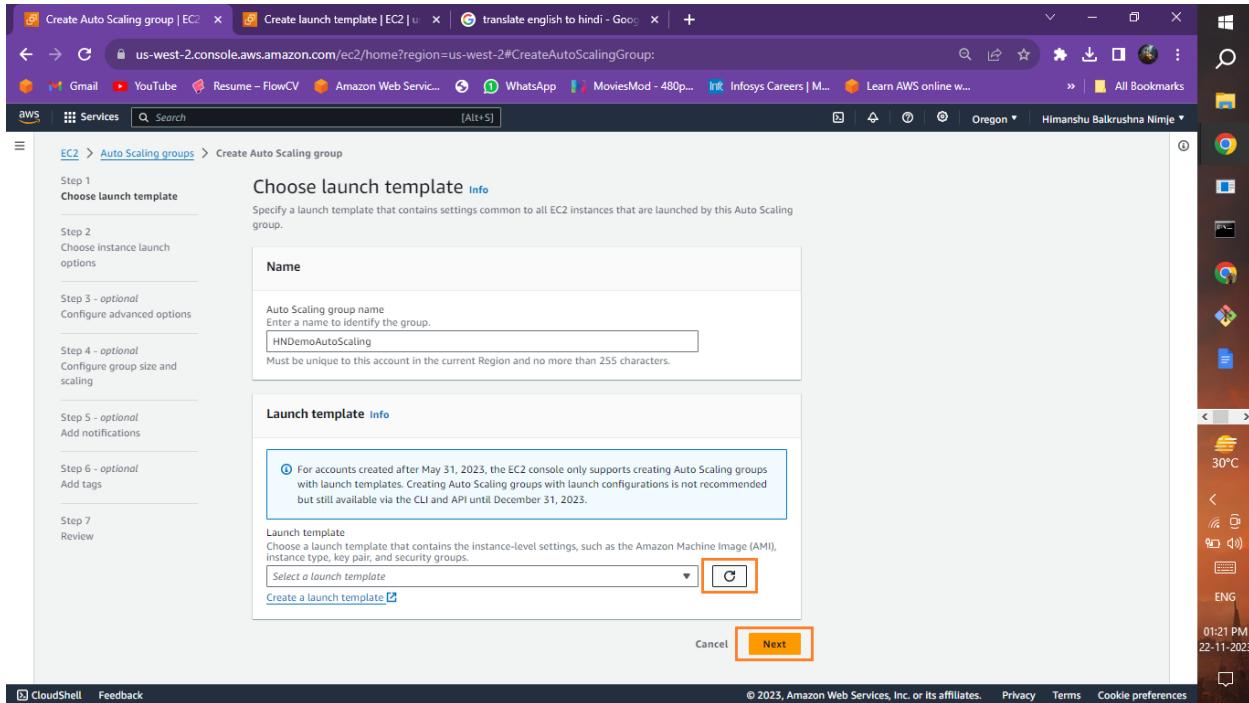
## Create New Key-Pair, Select Security Group

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The current step is 'Network settings'. In the 'Subnet Info' section, there is a dropdown menu labeled 'Select security groups'. An orange arrow points to this dropdown. A callout box is overlaid on the screen, providing details about the '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, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.'

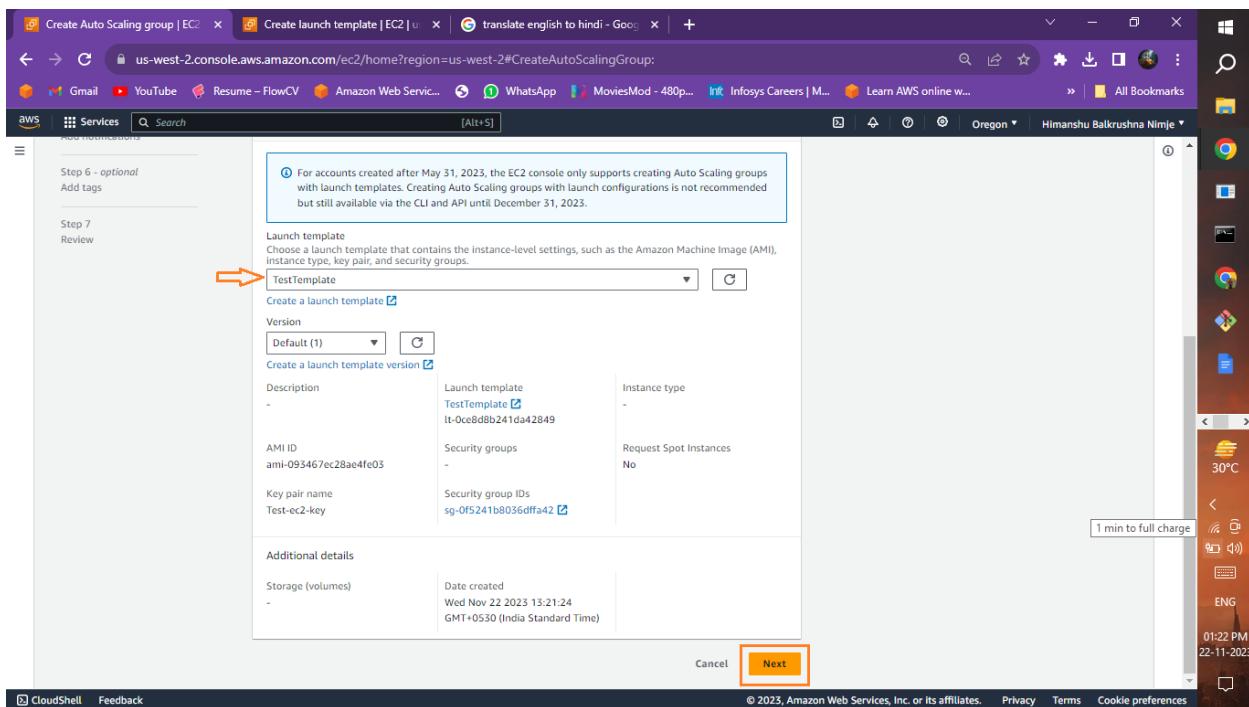
## Click on the create launch template

The screenshot shows the 'Create launch template' wizard completed. A green success bar at the top states 'Successfully created TestTemplate(t-0ce8db241da42849)'. An orange arrow points to this message. Below the success bar, there is a 'Next Steps' section with several options: 'Launch an instance', 'Create an Auto Scaling group from your template', 'Create Auto Scaling group', and 'Create Spot Fleet'. At the bottom right of the page, there is a 'View launch templates' button.

## Click on the Refresh Option



## (TestTemplate) are Show, Click on the Next option



## Select VPC

**Allocation strategies** Info

On-Demand allocation strategy  
Choose the allocation strategy to apply to your On-Demand Instances when they are launched.

Prioritized Request On-Demand Instances based on the priority order of instance types that you set above. This strategy can't be used with attribute-based instance type selection.

Lowest price Request On-Demand Instances from the lowest priced pools within an Availability Zone.

**Network** Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC  
Choose the VPC that defines the virtual network for your Auto Scaling group.  
 172.31.0.0/16 Default

Create a VPC [?]

Availability Zones and subnets  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Create a subnet [?]

Cancel Skip to review Previous Next

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Select subnets and Click on the Next

**Allocation strategies** Info

On-Demand allocation strategy  
Choose the allocation strategy to apply to your On-Demand Instances when they are launched.

Prioritized Request On-Demand Instances based on the priority order of instance types that you set above. This strategy can't be used with attribute-based instance type selection.

Lowest price Request On-Demand Instances from the lowest priced pools within an Availability Zone.

**Network** Info

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC  
Choose the VPC that defines the virtual network for your Auto Scaling group.  
 172.31.0.0/16 Default

Create a VPC [?]

Availability Zones and subnets  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

us-west-2a | subnet-0973ad2afeafb10ec   
172.31.16.0/20 Default

us-west-2b | subnet-05cffc38508f3f9c5   
172.31.32.0/20 Default

us-west-2c | subnet-01bbfc4034b2d2f86   
172.31.0.0/20 Default

Create a subnet [?]

Cancel Skip to review Previous Next

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Select (Attach to a New Load Balancer) select Application Load Balancer

**Configure advanced options - optional**

Integrate your Auto Scaling group with other services to distribute network traffic across multiple servers using a load balancer or to establish service-to-service communications using VPC Lattice. You can also set options that give you more control over health check replacements and monitoring.

**Load balancing**

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.

HNDemoAutoScaling-1

## Select Internet Facing option

**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.

HNDemoLoadBalancer

**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-0cfdc8f994cb5dabe

**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.

|            |                          |
|------------|--------------------------|
| us-west-2b | subnet-05cffc38508f3f9c5 |
| us-west-2a | subnet-0973ad2afeafb10ec |
| us-west-2c | subnet-01bbfc4034b2d2f86 |

## Select target group

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-0cfdc8f994cb5dabe

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-west-2b | subnet-05cffc38508f3f9c5 |
| <input checked="" type="checkbox"/> us-west-2a | subnet-0973ad2afeaf810ec |
| <input checked="" type="checkbox"/> us-west-2c | subnet-01bbfc4034b2d2f86 |
| <input type="checkbox"/> us-west-2d            | 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   | <input type="button" value="Create a target group"/> |

New target group name  
An instance target group with default settings will be created.

HNDemoLoadBalancer

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

Add tag

50 remaining

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on The health checks option and click on the Next

EC2 health checks  
 Always enabled

Additional health check types - optional [Info](#)

Turn on Elastic Load Balancing health checks [Recommended](#)  
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

Turn on VPC Lattice health checks [Info](#)  
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

Health check grace period [Info](#)  
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

Additional settings

Monitoring [Info](#)  
 Enable group metrics collection within CloudWatch

Default instance warmup [Info](#)  
The amount of time CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

Enable default instance warmup

Cancel Skip to review Previous **Next**

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Select Desired Capacity 1, Minimum desired capacity 1, And Maximum desired capacity 2

Group size info  
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type  
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity  
Specify your group size.  
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  
Max desired capacity  
1 2  
Equal or less than desired capacity Equal or greater than desired capacity

Automatic scaling - optional  
Choose whether to use a target tracking policy. 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.

Click on the Next

Control availability and cost during replacement events  
An instance maintenance policy determines how much availability your application has when EC2 Auto Scaling replaces instances. It also establishes guardrails that limit the amount of capacity that can be added or removed when replacing instances.

Choose a replacement behavior depending on your availability requirements

Mixed behavior  
For rebalancing events, new instances will launch before terminating others. For all other events, instances terminate and launch at the same time.

Prioritize availability  
Launch before terminating  
Launch new instances and wait for them to be ready before terminating others. This allows you to go above your desired capacity by a given percentage and may temporarily increase costs.

Control costs  
Terminate and launch  
Terminate and launch instances at the same time. This allows you to go below your desired capacity by a given percentage and may temporarily reduce availability.

Flexible  
Custom behavior  
Set custom values for the minimum and maximum amount of available capacity. This gives you flexibility in setting how far below and over your desired capacity EC2 Auto Scaling goes when replacing instances.

Instance scale-in protection  
Scale-in protection prevents newly launched instances from being terminated by scaling activities. Make sure to remove scale-in protection for the group or individual instances when instances are ready to be terminated.

Enable instance scale-in protection

Cancel Skip to review Previous **Next**

Click on the Create Auto Scaling group

The screenshot shows the final step of creating an Auto Scaling group. It includes sections for replacement behavior, instance scale-in protection, notifications, and tags. At the bottom right is a prominent orange 'Create Auto Scaling group' button.

Auto Scaling group has been Created

The screenshot shows the 'Auto Scaling groups' page with one entry listed:

| Name              | Launch template/configuratio... | Instan... | Status              | Desired capa... | M... | M... | Availability Zones                 |
|-------------------|---------------------------------|-----------|---------------------|-----------------|------|------|------------------------------------|
| HNDemoAutoScaling | TestTemplate   Version Default  | 0         | Updating capacity.. | 1               | 1    | 2    | us-west-2a, us-west-2b, us-west-2c |

At the bottom left, it says '0 Auto Scaling groups selected'.

Go to inside Auto Scaling group, go to security, click on the Edit

The screenshot shows the AWS Management Console for the HNDemoLoadBalancer. On the left, there's a navigation sidebar with various services like Spot Requests, Savings Plans, Reserved Instances, etc. The main area displays the 'Details' section for the load balancer, including its type (Application), status (Provisioning), VPC (vpc-0cfcd8f994cb5dabe), and IP address type (IPv4). Below this, the 'Listeners and rules' and 'Network mapping' tabs are shown, followed by the 'Security' tab, which is highlighted with an orange box. Under the 'Security' tab, there's a table for 'Security groups (1)'. The table has columns for 'Security Group ID', 'Name', and 'Description'. It lists one entry: sg-0f5241b8036dfffa42, named SGforAutoScal..., with the description SGforAutoScaling. An orange box highlights the 'Edit' button next to the table.

Select security group and click on save changes

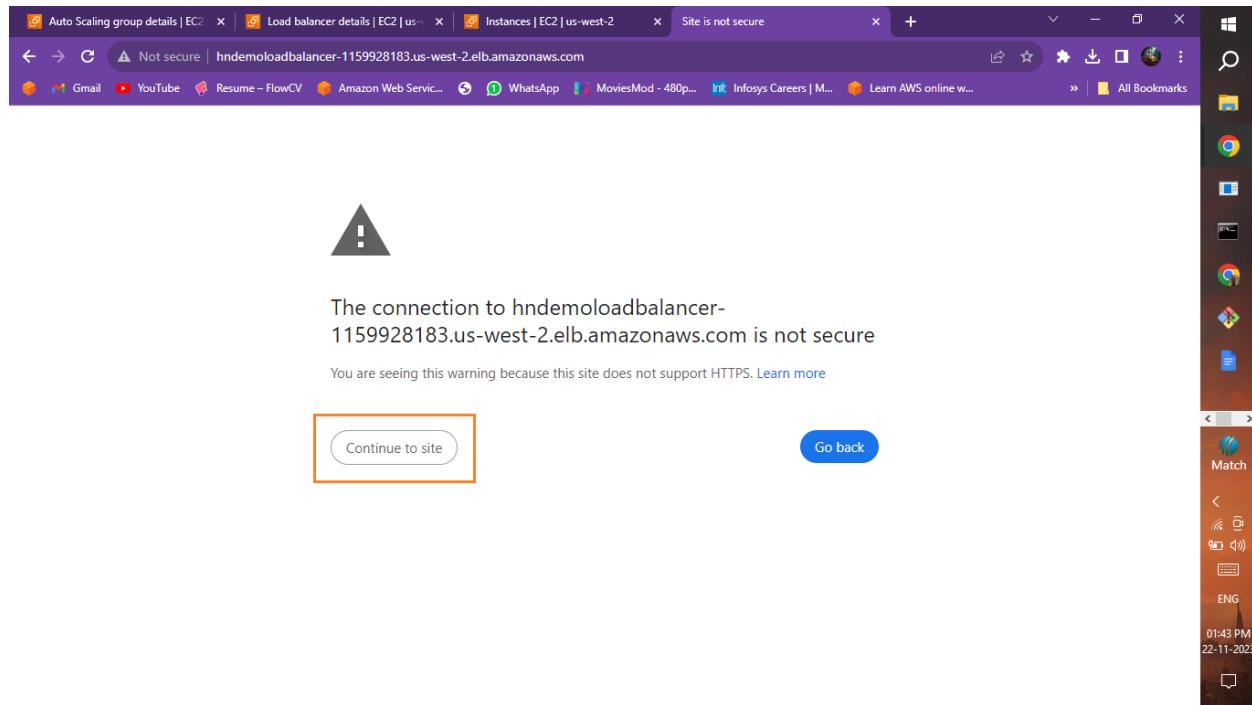
The screenshot shows the 'Edit security groups' dialog for the HNDemoLoadBalancer. At the top, it says 'Edit security groups' and 'Load balancer details: HNDemoLoadBalancer'. Below this, there's a 'Security groups' section with a note: 'A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group'. A dropdown menu titled 'Select up to 5 security groups' shows three options: 'SGforALB' (selected with a checked checkbox), 'SGforAutoScaling' (unchecked), and 'default' (unchecked). An arrow points to the checked checkbox for 'SGforALB'. At the bottom right of the dialog, there's a 'Save changes' button, which is highlighted with an orange box.

The screenshot shows the AWS Management Console with the EC2 service selected. Under the 'Auto Scaling Groups' section, there is one item listed: 'HNDemoAutoScaling'. A black arrow points to the group name 'HNDemoAutoScaling'.

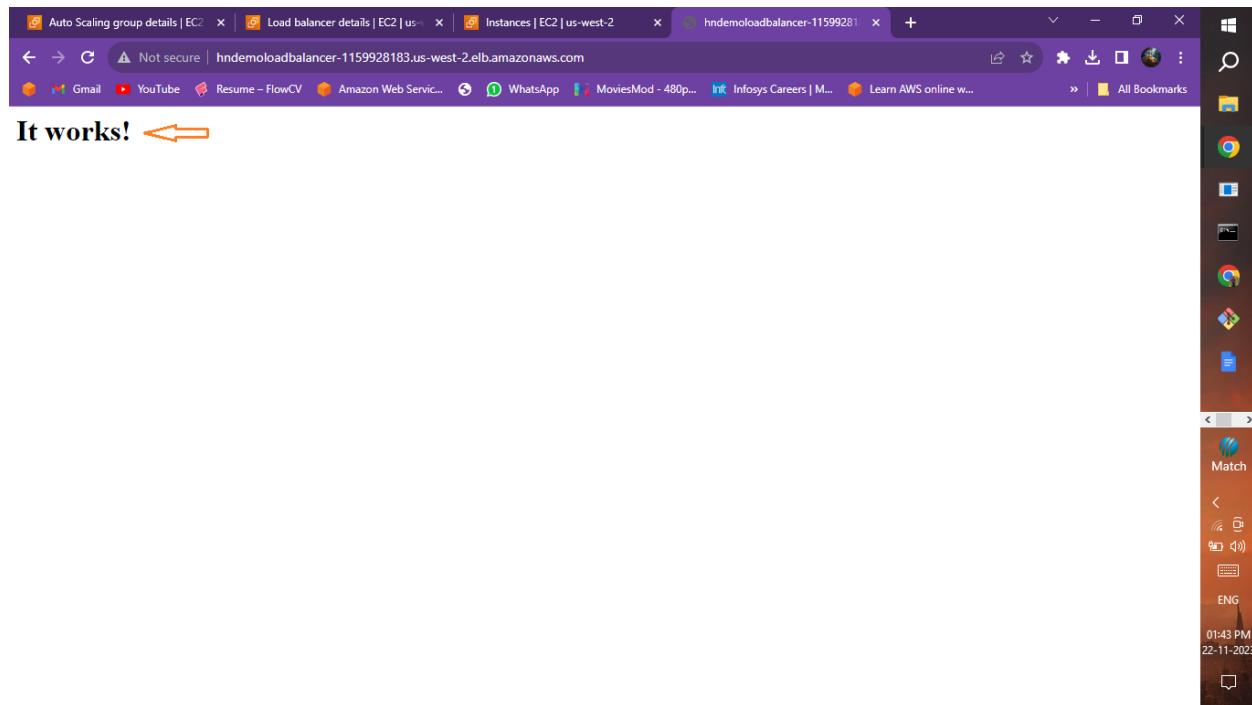
Copy DNS name and click on

The screenshot shows the AWS Management Console with the EC2 service selected. Under the 'Load balancers' section, there is one item listed: 'HNDemoLoadBalancer'. The 'Details' tab is selected, showing the VPC information: 'vpc-0cfdc8ff994cb5dabe'. Below the VPC information, a message box says 'Successfully modified security groups.' and 'DNS name copied'. The copied DNS name is 'HNDemoLoadBalancer-1159928183.us-west-2.elb.amazonaws.com (A Record)'. This message box is highlighted with an orange border.

continue to site



Result is showing



## Creating Auto Scaling policy

Select dynamic scaling policy

Select policy type→(Target Tracking Policy) and click on the create

The screenshot shows the 'Create dynamic scaling policy' wizard. The 'Policy type' dropdown is set to 'Target tracking scaling'. The 'Scaling policy name' input field contains 'Target Tracking Policy'. The 'Target value' input field contains '30'. The 'Create' button at the bottom right is highlighted with an orange box.

The screenshot shows the 'Auto Scaling group details' page for 'HNDemoAutoScaling'. A green banner at the top says 'Dynamic scaling policy created or edited successfully.' The 'Automatic scaling' tab is selected. It shows one policy named 'Target Tracking Policy' with the following details: 'Target tracking scaling', 'Enabled', 'As required to maintain Average CPU utilization at 30', 'Add or remove capacity units as required', and '300 seconds to warm up before including in metric'. The 'Create dynamic scaling policy' button is located at the top of the policy list.

Go to the cloudwatch  
(cloudwatch alarm to invoke an Amazon EC2 Auto Scaling policy)

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services like EC2, Support, CloudFormation, IAM, Elastic Beanstalk, CloudWatch (which is highlighted with a red box), and Lightsail. Below that is a 'Cost and usage' section showing current month costs of \$10.70 and forecasted month end costs of \$14.46. The main area has sections for 'Welcome to AWS', 'AWS Health', and 'Build a solution'. A status bar at the bottom indicates the URL is <https://us-west-2.console.aws.amazon.com/cloudwatch/home?region=us-west-2>.

Click on the Configure Application Insights

The screenshot shows the CloudWatch Overview page. On the left, there's a sidebar with 'CloudWatch' selected, followed by 'Favorites and recent', 'Alarms' (0), 'Logs', 'Metrics New', 'X-Ray traces', 'Events', 'Application monitoring', and 'Insights'. Under 'Insights', there are links for 'Settings New', 'Getting Started', and 'What's new'. The main area has sections for 'Get started with CloudWatch' (with options to 'Create alarms', 'Create a default dashboard', 'View logs', and 'View events') and 'Get started with Application Insights' (with a 'Configure Application Insights' button highlighted with a red box). At the bottom, there's a 'Cross service dashboard' section for 'Application ELB'.

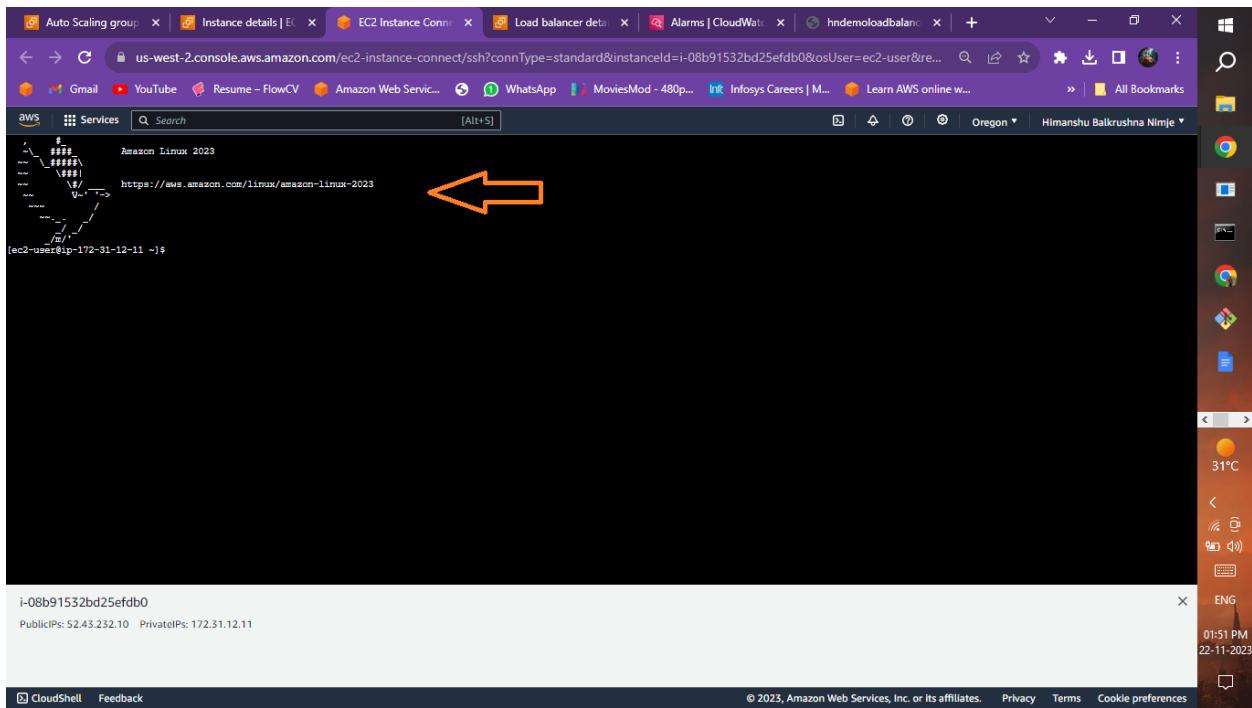
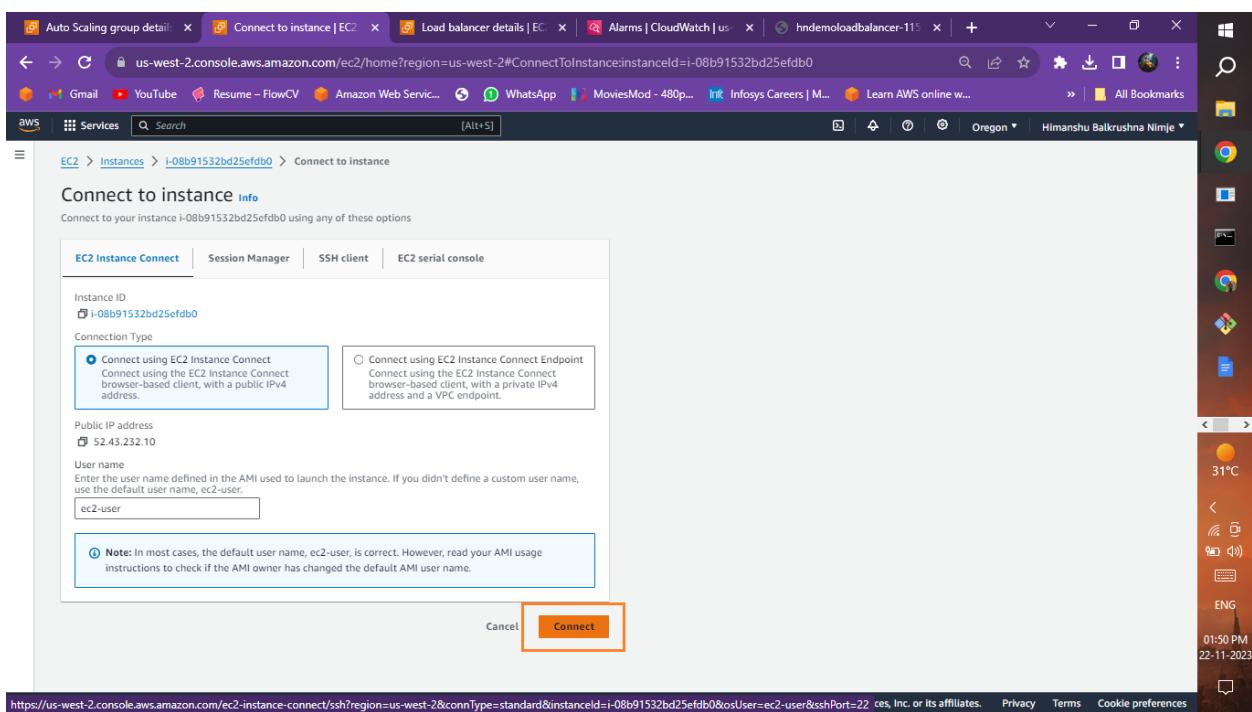
The screenshot shows the AWS CloudWatch Alarms page. On the left, there's a sidebar with navigation links like 'Alarms', 'Logs', 'Metrics', etc. The main area displays two alarms:

- TargtTracking-HNDemoAutoScaling-AlarmHigh**: State: OK, Last state update: 2023-11-22 08:18:33. Condition: CPUUtilization > 30 for 3 datapoints within 3 minutes. Actions: Actions enabled.
- TargtTracking-HNDemoAutoScaling-AlarmLow**: State: Insufficient data, Last state update: 2023-11-22 08:17:45. Condition: CPUUtilization < 21 for 15 datapoints within 15 minutes. Actions: Actions enabled.

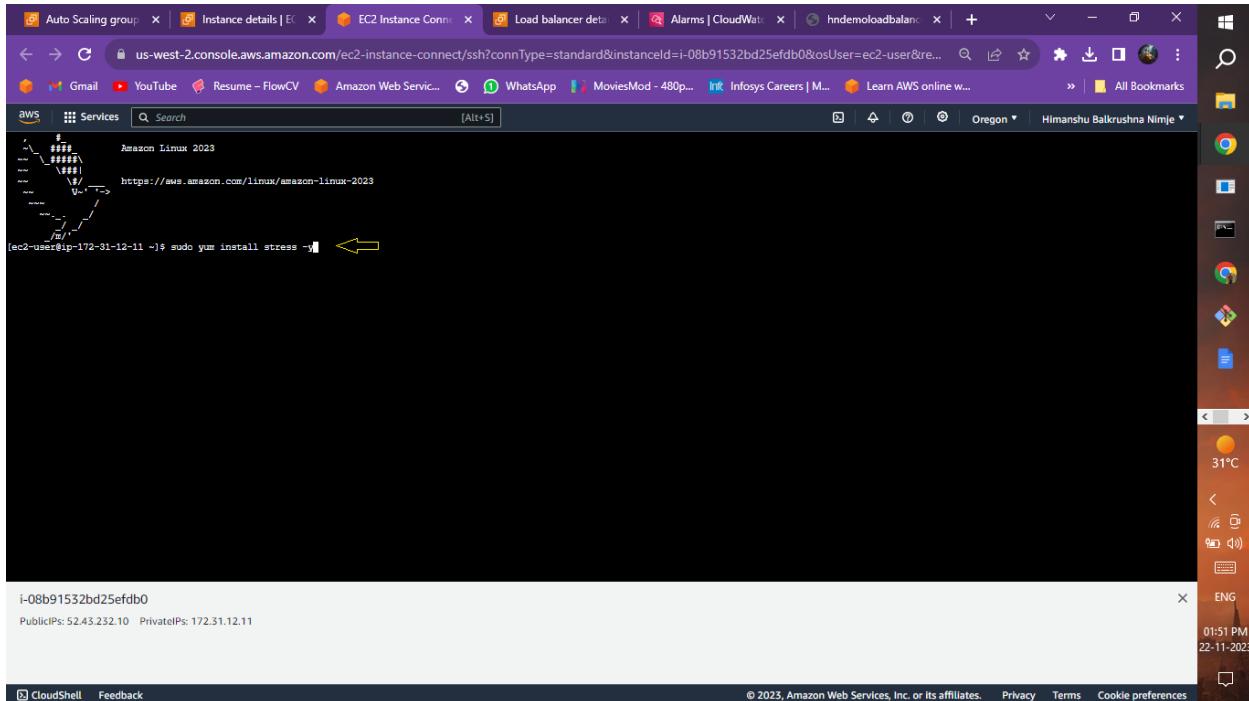
Click on the connect

The screenshot shows the AWS EC2 Instances page for instance **i-08b91532bd25efdb0**. The instance is currently running. The 'Connect' button in the top right corner is highlighted with a red box. The instance details include:

- Instance ID:** i-08b91532bd25efdb0
- Public IPv4 address:** 52.43.232.10 [open address]
- Private IP address:** 172.31.12.11
- Public IPv4 DNS:** ec2-52-43-232-10.us-west-2.compute.amazonaws.com [open address]
- Hostname type:** ip-172-31-12-11.us-west-2.compute.internal
- IP name:** ip-172-31-12-11.us-west-2.compute.internal
- Answer private resource DNS name:** -
- Auto-assigned IP address:** 52.43.232.10 [Public IP]
- VPC ID:** vpc-0cfdc8f994cb5dabe
- Subnet ID:** subnet-01bbfc4034b2d2f86
- IAM Role:** -
- IMDSv2 Required:** -
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. [Learn more]
- Auto Scaling Group name:** HNDemoAutoScaling

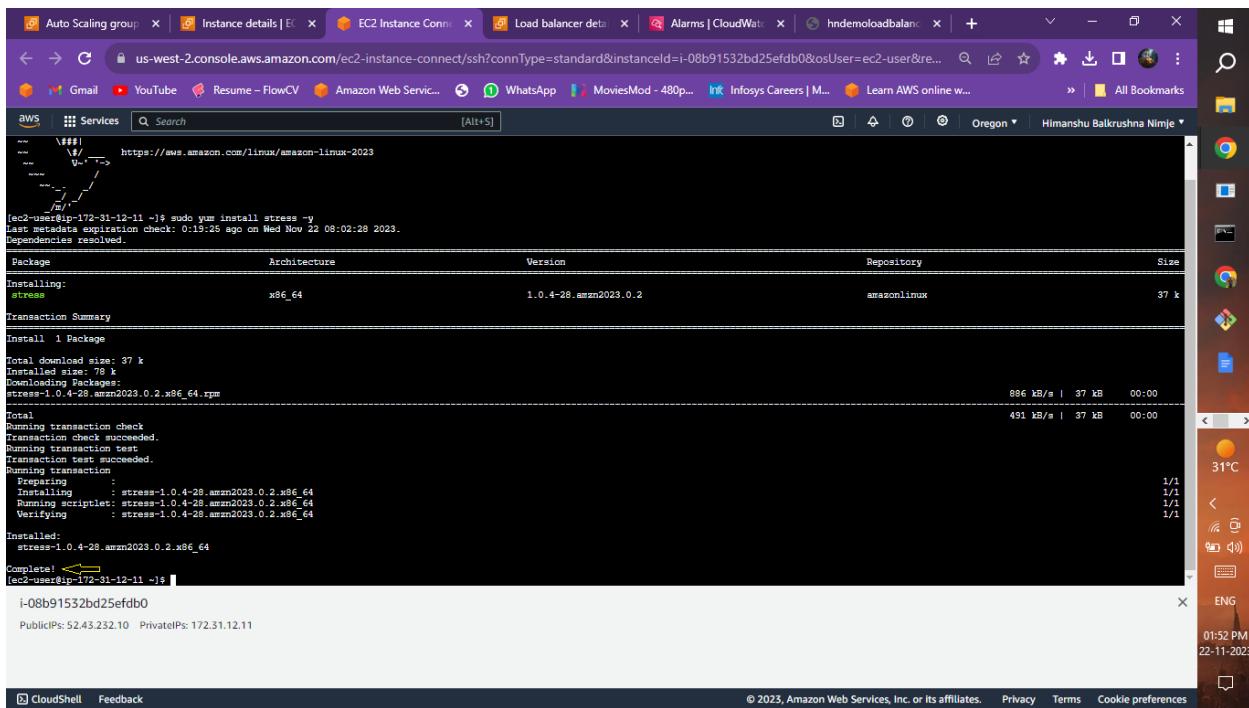


Enter the Command→sudo yum install stress -y



```
(ec2-user@ip-172-31-12-11 ~)$ sudo yum install stress -y
```

package is install



```
[ec2-user@ip-172-31-12-11 ~]$ sudo yum install stress -y
Last metadata expiration check: 0:19:25 ago on Wed Nov 22 08:02:28 2023.
Dependencies resolved.
Package           Architecture Version      Repository   Size
Installing:
stress            x86_64       1.0.4-28.amzn2023.0.2   amazonlinux 37 k
Transaction Summary
Install 1 Package
Total download size: 37 k
Installed size: 78 k
Downloading Packages:
stress-1.0.4-28.amzn2023.0.2.x86_64.rpm          886 kB/s | 37 kB  00:00
stress-1.0.4-28.amzn2023.0.2.x86_64               491 kB/s | 37 kB  00:00
Total
Running transaction check
  Running transaction test
    Transaction test succeeded.
  Running transaction
    Preparing: [ 1/1]
    Installing : stress-1.0.4-28.amzn2023.0.2.x86_64 1/1
    Running scriptlet: stress-1.0.4-28.amzn2023.0.2.x86_64 1/1
    Verifying   : stress-1.0.4-28.amzn2023.0.2.x86_64 1/1
Installed:
  stress-1.0.4-28.amzn2023.0.2.x86_64
Complete!
```

Enter Command → sudo stress -rpm 12 -timeout 240s  
For Trigger the Auto Scaling Policy

```
[ec2-user@ip-172-31-12-11 ~]$ sudo stress --cpu 12 --timeout 240s
```

Auto-Scaling Dashboard, the policy will get activated and Automatically Create New Instance

| Status  | Description  | Cause  | Start time                           | End time                             |
|---------|--|--|--------------------------------------|--------------------------------------|
| Pending | Waiting for instance warmup  | At 2023-11-22T08:29:33Z a monitor alarm TargetTracking-HNDemoAutoScaling triggered policy Target Tracking Policy increased the desired capacity from 1 to 2. At 2023-11-22T08:29:41Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2. | 2023 November 22, 01:59:44 PM +05:30 |                                      |
| Success | Launching a new EC2 instance i-08b91532bd25efdb0   | At 2023-11-22T08:02:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.  | 2023 November 22, 01:32:09 PM +05:30 | 2023 November 22, 01:32:15 PM +05:30 |
| Success | Updating load balancers/target groups/resource health  | At 2023-11-22T08:02:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.  | 2023 November 22, 01:32:07 PM +05:30 | 2023 November 22, 01:32:07 PM +05:30 |
| Failed  | Launched a new EC2 instance, but Resource could not launch On-Demand Instances. Unable to fulfill capacity. Failed to fulfill capacity. Please review. | At 2023-11-22T08:01:52Z a user-requested created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2023-11-22T08:01:52Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.  | 2023 November 22, 01:31:57 PM +05:30 | 2023 November 22, 01:31:57 PM +05:30 |

**HNDemoAutoScaling**

**Activity notifications (0)**

**Activity history (4)**

| Status                      | Description                                       | Cause   | Start time                    | End    |
|-----------------------------|---|---|-------------------------------|--------|
| Waiting for instance warmup | Launching a new EC2 instance; i-070e7008239b52821 | At 2023-11-22T08:29:33Z a monitor alarm TargetTracking-HNDemoAutoScaling-AlarmHigh-92d80850-20f0-42d9-9042-19b0609fd813 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2023-11-22T08:29:41Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2. | 2023 November 22, 01:59:44 PM | +05:30 |

**HNDemoAutoScaling**

**Instances (2)**

| Instance ID         | Lifecycle | Instance ... | Weighted ... | Launch t...  | Available... | Health st... | Protect... |
|---------------------|-----------|--------------|--------------|--------------|--------------|--------------|------------|
| i-070e7008239b52821 | InService | c7a.medium   | -            | TestTemplate | us-west-2a   | Healthy      |            |
| i-08b91532bd25efdb0 | InService | c7a.medium   | -            | TestTemplate | us-west-2c   | Healthy      |            |

**Lifecycle hooks (0) Info**

No lifecycle hooks are currently configured.

Lifecycle hooks help you perform custom actions on instances as they launch and before they terminate.

**Create lifecycle hook**

Here is new instance is Initializing, thus the policy is working successfully

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like Auto Scaling group, Instances, Images, and Elastic Block Store. The main area displays a table of instances. One instance, with the ID i-08b91532bd25efdb0, has its 'Status check' column highlighted with a red box, indicating it is currently 'Initializing'. Another instance, i-070e7008239b52821, is listed as 'Running'. The status bar at the bottom right of the browser window shows the date as 22-11-2023.

## Delete (Target Tracking Policy)

The screenshot shows the AWS Auto Scaling Groups page. The 'HNDemoAutoScaling' group is selected. A modal dialog is open, prompting the user to confirm the deletion of a scaling policy. The dialog asks, 'Are you sure you want to delete this resource? Target Tracking Policy'. The 'Delete' button in the dialog is highlighted with a red box. The status bar at the bottom right of the browser window shows the date as 22-11-2023.

Create new policy, i.e. (Step Scaling policy) and Click on the create

The screenshot shows the 'Create dynamic scaling policy' wizard in the AWS EC2 console. The 'Policy type' dropdown is set to 'Step scaling'. The 'Scaling policy name' input field contains 'Avg-cpu-policy'. The 'Create' button at the bottom right is highlighted with a box.

Go to cloudwatch for create new alarm, click on the select metric

The screenshot shows the 'Specify metric and conditions' step of the 'Create alarm' wizard in the AWS CloudWatch console. The 'Select metric' button is highlighted with a box and has an arrow pointing to it.

Select EC2 and Go inside

The screenshot shows the AWS CloudWatch Metrics Selection interface. The top navigation bar has tabs for 'Dynamic scaling', 'CloudWatch', 'Instances | EC2', 'EC2 Instance C...', 'Load balancer', 'Alarms | Cloud', 'hndemoloadb', and '+'. Below the navigation is a toolbar with icons for Gmail, YouTube, FlowCV, Amazon Web Services, WhatsApp, MoviesMod, Infosys Careers, Learn AWS online, and All Bookmarks. The main area is titled 'Step 1 Specify condition' and shows a graph titled 'Untitled graph'. The graph displays a single data series from 05:45 to 08:30, with values ranging from 0 to 1. A message says 'Your CloudWatch graph is empty. Select some metrics to appear here.' Below the graph is a search bar and a 'Metrics (230)' section. The 'Metrics' tab is selected, showing a list of metrics grouped by service. The 'EC2' group is highlighted with a red box, containing 18 metrics. Other groups include ApplicationELB (64), EBS (44), and Events (1). The bottom right corner shows system status: 31°C, ENG, 02:06 PM, 22-11-2023.

Select Auto Scaling Group

The screenshot shows the AWS CloudWatch Metrics Selection interface. The top navigation bar has tabs for 'Dynamic scaling', 'CloudWatch', 'Instances | EC2', 'EC2 Instance C...', 'Load balancer', 'Alarms | Cloud', 'hndemoloadb', and '+'. Below the navigation is a toolbar with icons for Gmail, YouTube, FlowCV, Amazon Web Services, WhatsApp, MoviesMod, Infosys Careers, Learn AWS online, and All Bookmarks. The main area is titled 'Step 1 Specify condition' and shows a graph titled 'Untitled graph'. The graph displays a single data series from 05:45 to 08:30, with values ranging from 0 to 1. A message says 'Your CloudWatch graph is empty. Select some metrics to appear here.' Below the graph is a search bar and a 'Metrics (44)' section. The 'Metrics' tab is selected, showing a list of metrics grouped by service. The 'By Auto Scaling Group' group is highlighted with a red box, containing 16 metrics. Other groups include All (28) and Per-Instance Metrics (28). The bottom right corner shows system status: 31°C, ENG, 02:06 PM, 22-11-2023.

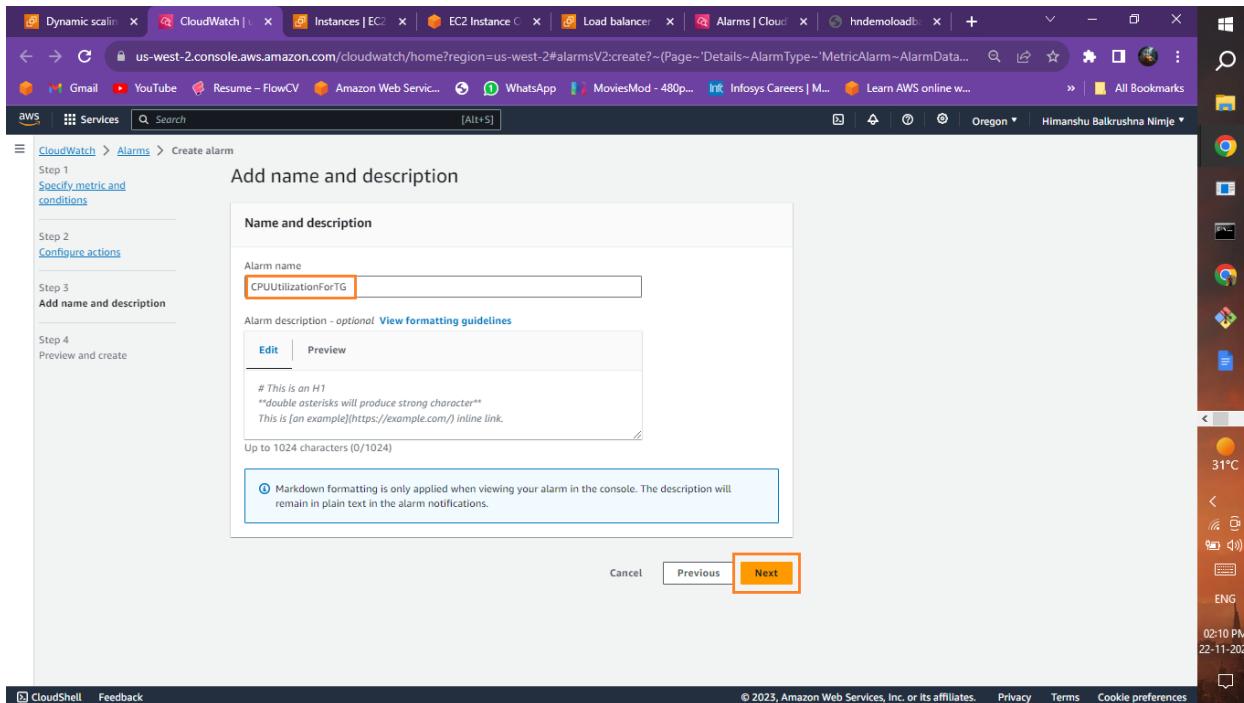
## select CPUUtilization

The screenshot shows the 'Select metric' dialog box from the AWS CloudWatch Metrics Selection interface. The 'Metrics' tab is selected, showing a list of metrics for the 'HNDemoAutoScaling' Auto Scaling group in the 'Oregon' region. The 'CPUUtilization' metric is selected and highlighted with a blue border. An orange arrow points to this selection. At the bottom right of the dialog is a large orange button labeled 'Select metric'.

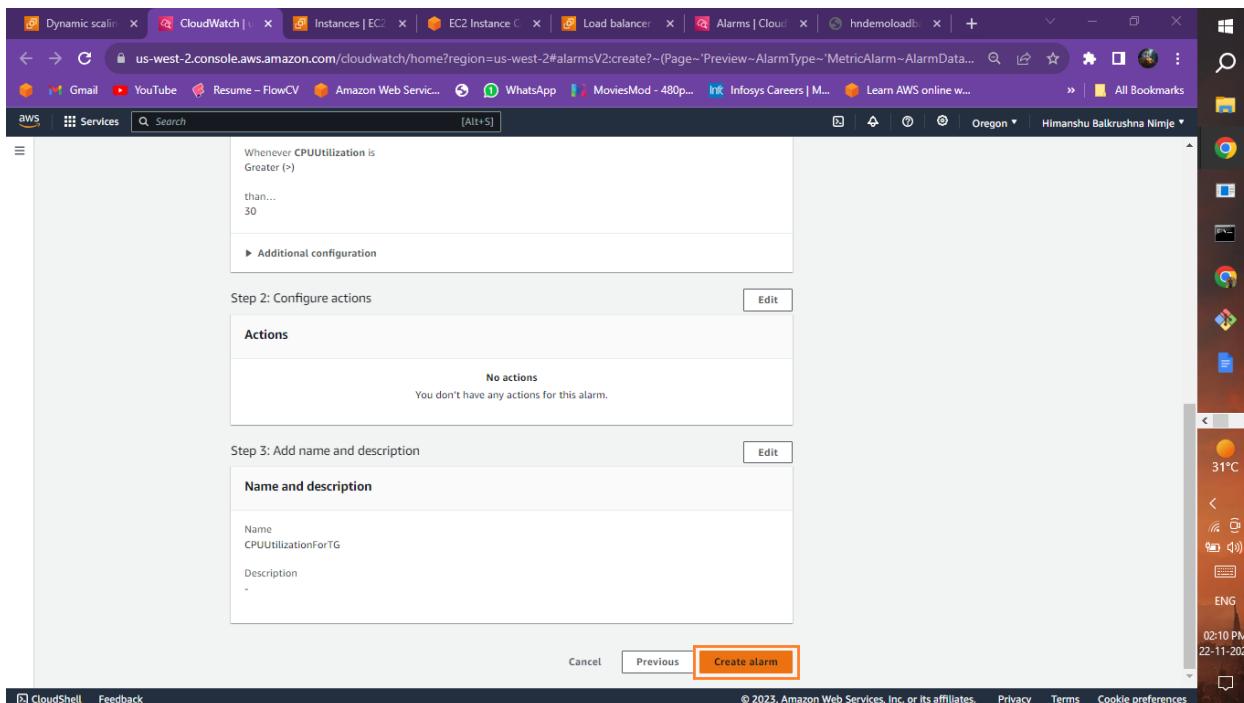
after select metric, click on Next

The screenshot shows the 'Specify metric and conditions' step of the 'Create alarm' wizard. In the 'Metric' section, the 'CPUUtilization' metric is selected for the 'HNDemoAutoScaling' Auto Scaling group. In the 'Conditions' section, the 'Threshold type' is set to 'Static' (using a value as a threshold). The 'Greater than threshold' option is selected, and the threshold value is set to '30'. An orange arrow points to the 'Next Step' button at the bottom right of the page.

## Assign the Alarm name



And click on the Create alarm



Alarm is successfully created

The screenshot shows the AWS CloudWatch Alarms page. At the top, a green banner says "Successfully created alarm CPUUtilizationForTG.". Below it, the "Alarms (1)" section lists one item:

| Name                | State             | Last state update   | Conditions  | Actions    |
|---------------------|-------------------|---------------------|---|------------|
| CPUUtilizationForTG | Insufficient data | 2023-11-22 08:40:57 | CPUUtilization > 30 for 1 datapoints within 5 minutes | No actions |

A red arrow points to the "CPUUtilizationForTG" row.

Refresh. Alarm is present in cloudwatch alarm, select Alarm and click on create

The screenshot shows the "Create dynamic scaling policy" page. The "Scaling policy name" field is filled with "Avg-cpu-policy". The "CloudWatch alarm" dropdown is set to "CPUUtilizationForTG". The "Create" button at the bottom right is highlighted with a red box.

The screenshot shows the AWS EC2 Auto Scaling Groups page for the group 'HNDemoAutoScaling'. A green success message box at the top states: 'Dynamic scaling policy created or edited successfully.' An arrow points from the text to the message box. The page includes tabs for Details, Activity, Automatic scaling (which is selected), Instance management, Monitoring, and Instance refresh. Below these tabs, there are sections for 'Dynamic scaling policies' and 'Predictive scaling policies', both currently showing zero items.

## change Auto Scaling Instance Capacity

Click on the Edit

The screenshot shows the AWS EC2 Auto Scaling Groups page for the group 'HNDemoAutoScaling'. The 'Automatic scaling' tab is selected. In the 'Group details' section, the 'Desired capacity' field is highlighted with a red box, and the 'Edit' button next to it is also highlighted. Other fields shown include 'Auto Scaling group name: HNDemoAutoScaling', 'Date created: Wed Nov 22 2023 13:31:51 GMT+0530 (India Standard Time)', 'Minimum capacity: 1', and 'Maximum capacity: 2'. The 'Launch template' section shows a launch template named 'TestTemplate' with AMI ID 'ami-093467cc28ae4fe03' and other configuration details.

## Change Desired capacity type

### And click on the Update

The screenshot shows the AWS Auto Scaling Groups page. In the center, a modal dialog box titled "Group size" is open. Inside, the "Desired capacity" section shows a dropdown menu with "1" selected. A red arrow points to this selection. At the bottom right of the dialog, the "Update" button is highlighted with a red box. The background shows the main Auto Scaling group details, including the group name "HNDemoAutoScaling" and its launch template.

created policy work successfully The EC2 Instance is Terminating

The screenshot shows the AWS Activity history page. The table lists five entries. The first entry, which is highlighted with a red arrow, shows an "Terminating EC2 instance" status with a "Cause" description: "At 2023-11-22T08:45:42Z a user request update of AutoScalingGroup constraints to min: 1, max: 2, desired: 1 changing the desired capacity from 2 to 1. At 2023-11-22T08:45:49Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2023-11-22T08:45:50Z instance i-08b91532bd25efdb0 was selected for termination." The "Start time" and "End time" for this entry are both listed as "2023 November 22, 02:15:50 PM +05:30".

The screenshot shows the AWS CloudWatch Activity history page for the HNDemoAutoScaling group. The left sidebar shows various AWS services like EC2, S3, Lambda, etc. The main area displays activity notifications and a detailed history of actions taken on the Auto Scaling group.

**Activity notifications (0)**

**Activity history (5)**

| Status                   | Description  | Cause   | Start time                    | End time                         |
|--------------------------|--|---|-------------------------------|----------------------------------|
| Terminating EC2 instance | i-08b91532bd25efdb0 - Waiting For ELB Connection Draining. | At 2023-11-22T08:45:42Z a user request update of AutoScalingGroup constraints to min: 1, max: 2, desired: 1 changing the desired capacity from 2 to 1. At 2023-11-22T08:45:49Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2023-11-22T08:45:50Z instance i-08b91532bd25efdb0 was selected for termination. | 2023 November 22, 02:15:50 PM | 40:50                            |
| Successful               | Launching a new EC2 instance                               | At 2023-11-22T08:29:33Z a monitor alarm TargetTracking-HNDemoAutoScaling-AlarmHigh-92d80850-20f0-42d9-9042-19b0609fdb13 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2023-11-22T08:29:41Z an instance was launched.   | 2023 November 22, 01:49:44 PM | 2023 November 22, 02:07:45:50 PM |

## Open cloudshell

Enter the command and cloudwatch Alarm is trigger

The screenshot shows the AWS CloudShell terminal window. The command `aws cloudwatch set-alarm-state --alarm-name CPUUtilizationForTG --state-value ALARM --state-reason "test"` is being typed into the terminal. A yellow arrow points to the end of the command line.

```
[cloudshell-user@ip-10-130-32-48 ~]$ aws cloudwatch set-alarm-state --alarm-name CPUUtilizationForTG --state-value ALARM --state-reason "test"
```

A message at the bottom of the terminal indicates: "CloudShell will migrate from Amazon Linux 2 to Amazon Linux 2023 starting December 4, 2023. For more information, see the User Guide." There is also a "Don't show this again" button.

## IN ALARM

The screenshot shows the AWS CloudWatch Alarms interface. On the left, a sidebar lists various monitoring services like CloudWatch Metrics, X-Ray traces, and Application monitoring. The main area displays a table of alarms. One alarm, named 'CPUUtilizationForTG', is highlighted with a red border and labeled 'In alarm'. The condition for this alarm is 'CPUUtilization > 30 for 1 datapoints within 5 minutes'. The status update was at 2023-11-22 09:02:31. A message indicates 'Actions enabled'.

Here is the new Instance is Initializing

The screenshot shows the AWS EC2 Instances interface. The left sidebar includes options like EC2 Dashboard, Instances, Instance Types, and Network & Security. The main area shows a table of instances. One instance, with the ID 'i-070e7008239b52821', is highlighted with a red arrow pointing to it. Its status is listed as 'Running' with a green circle icon. Another instance, 'i-028ab0de899b3e11c', is shown with a status of 'Initializing' and a yellow circle icon. The table also includes columns for Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 IP.

Here Both policies are activated, the first is automatically Initializing the instance and the second is automatically terminating the instance

The screenshot shows the AWS CloudWatch Activity history page. The left sidebar lists various AWS services like EC2 Dashboard, Instances, and Network & Security. The main area displays activity history with the following details:

| Status     | Description  | Cause   | Start time                           | End time                             |
|------------|--|---|--------------------------------------|--------------------------------------|
| Successful | Launching a new EC2 instance: i-028ab0de899b3e11c                        | At 2023-11-22T09:01:53Z a user request update of AutoScalingGroup constraints to min: 1, max: 2, desired: 2 changing the desired capacity from 1 to 2. At 2023-11-22T09:02:02Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.  | 2023 November 22, 02:32:05 PM +05:30 | 2023 November 22, 02:32:11 PM +05:30 |
| Successful | Terminating EC2 instance: i-08091532bd25efdb0                            | At 2023-11-22T08:45:42Z a user request update of AutoScalingGroup constraints to min: 1, max: 2, desired: 1 changing the desired capacity from 2 to 1. At 2023-11-22T08:45:49Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2023-11-22T08:45:50Z instance i-08091532bd25efdb0 was selected for termination. | 2023 November 22, 02:15:50 PM +05:30 | 2023 November 22, 02:21:55 PM +05:30 |
| Successful | Launching a new EC2 instance: i-070e7008239b52821                        | At 2023-11-22T08:29:33Z a monitor alarm TargetTracking-HNDemoAutoScaling-AlarmHigh-92d80850-20f0-42d9-9042-19b0609fd813 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 1 to 2. At 2023-11-22T08:29:41Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.                             | 2023 November 22, 01:59:44 PM +05:30 | 2023 November 22, 02:04:50 PM +05:30 |
| Successful | Launching a new EC2 instance: i-08091532bd25efdb0                        | At 2023-11-22T08:02:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.   | 2023 November 22, 01:32:09 PM +05:30 | 2023 November 22, 01:32:15 PM +05:30 |
|            | Updating load balancers/target groups: Successful. Status Reason: Arithm |   |                                      |                                      |

## What I Learned

By completing the project on Auto-Scaling and Load-Balancing in AWS EC2 using templates I gained a variety of skills and knowledge related to cloud computing

1. Understanding of Auto-Scaling Concept:- Learn how to set up and configure auto-scaling groups to automatically adjust the number of EC2 instances based on demand.
2. Load-Balancing Configuration:- Understand the significance of load balancing in distributing incoming traffic across multiple instances to ensure even resource utilization and improve fault tolerance.
3. AWS EC2 Instances Management:- Acquire proficiency in launching and managing EC2 instances, including configuring security groups, key pairs, and other instance-related settings.
4. Monitoring and Alarming:- Understand how to use AWS CloudWatch or other monitoring solutions to gather metrics and trigger alerts based on predefined thresholds.
5. Cost Optimization:- Understand the cost implications of different EC2 instance types and how to choose the right instance type for your application.