

TASK 5 – Load Balancer & Auto Scaling

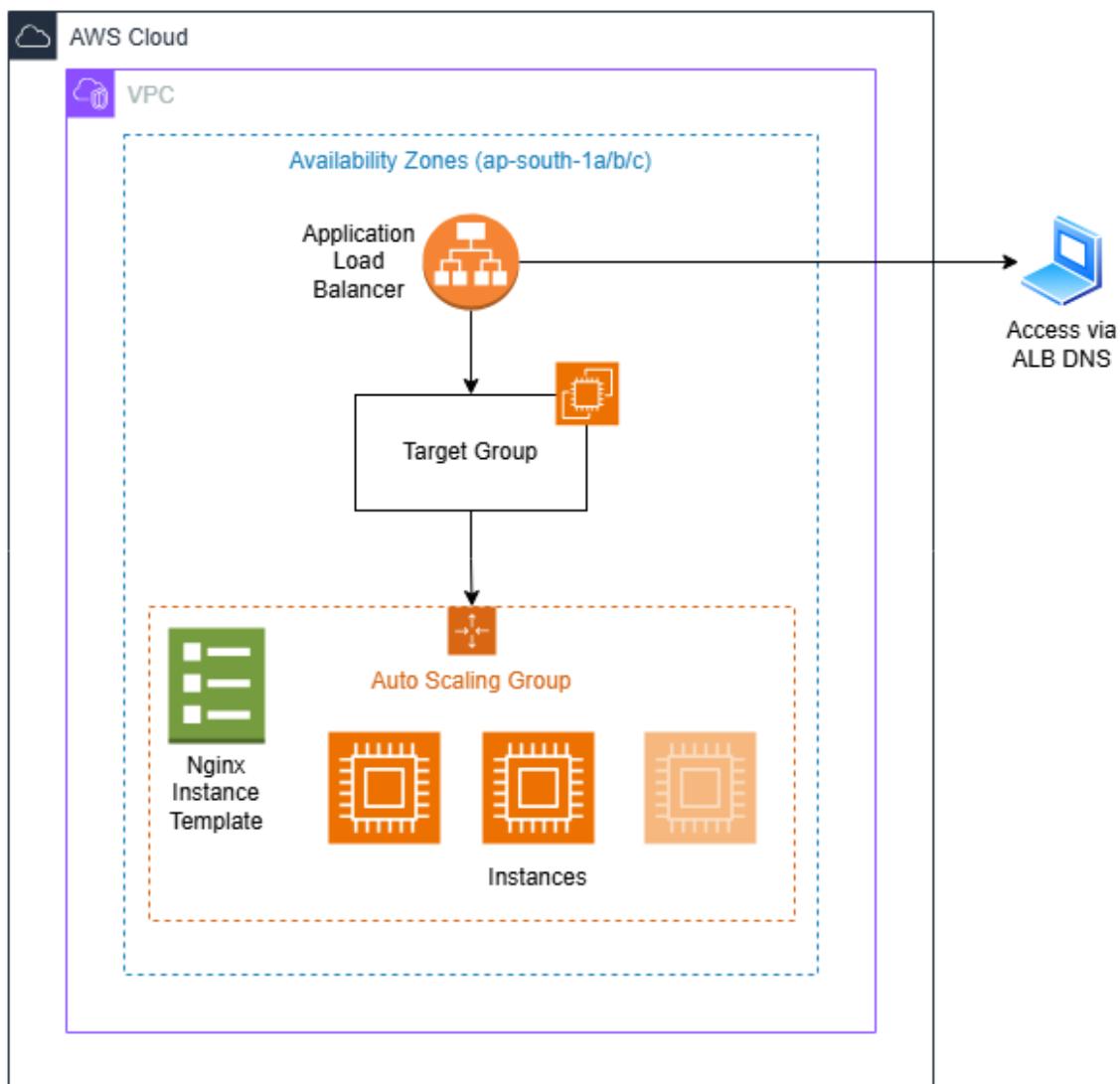
Requirements:

- Configure Application Load Balancer (ALB)
- Attach Auto Scaling Group (ASG)
- Scale based on CPU utilization

Prerequisites:

- An active AWS account with access to EC2, VPC, and Auto Scaling services.
- An existing or default VPC with public subnets across multiple Availability Zones.
- An EC2 key pair for secure instance access.

Architecture:



The architecture consists of an Application Load Balancer (ALB) deployed inside a VPC across multiple Availability Zones in the ap-south-1 (Mumbai) region. The ALB forwards incoming HTTP traffic to a Target Group associated with an Auto Scaling Group (ASG). The Auto Scaling Group launches EC2 instances using a Launch Template, where each instance automatically runs a NGINX web server inside a Docker container. This design ensures high availability, load distribution, automatic scaling, and self-healing of the application.

Step-by-step procedure:

I) EC2 Template Creation

1. Logged in to the AWS Management Console and selected the Asia Pacific (Mumbai – ap-south-1) region to deploy resources close to users.

The screenshot shows the AWS Management Console home page. On the left, there's a sidebar with 'Recently visited' services: IAM, VPC, Amazon EventBridge, Key Management Service, EC2, Lambda, Aurora and RDS, S3, and CloudWatch. Below this is a 'View all services' link. To the right, there's a 'Applications' section with a 'Create application' button and a note that says 'No applications'. It also includes a 'Find applications' search bar and a 'Go to myApplications' link. At the bottom of the page, there are tabs for 'Welcome to AWS', 'AWS Health', and 'Cost and usage', along with various status icons and a date/time stamp (07-02-2026).

2. Opened the EC2 service dashboard to manage compute resources required for the application.

The screenshot shows the EC2 service dashboard. On the left, there's a navigation menu with sections like EC2 (Dashboard, AWS Global View, Events), Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager), Images (AMIs, AMI Catalog), and Elastic Block Store. The main content area features a large heading 'Amazon Elastic Compute Cloud (EC2)' with the subtext 'Create, manage, and monitor virtual servers in the cloud.' Below this, there's a section titled 'Benefits and features' with a sub-section 'EC2 offers ultimate scalability and control'. It lists three bullet points: 'Highest level of control of the entire technology stack, allowing full integration with all AWS services', 'Wide variety of server size options', and 'Wide availability of operating systems to choose from including Linux, Windows, and macOS'. To the right, there are two call-to-action boxes: 'Launch a virtual server' (with 'Launch instance', 'View dashboard', and 'Get started walkthroughs' buttons) and 'Additional actions' (with 'View running instances' and 'Migrate a server' buttons). The bottom of the page has standard AWS footer links and a date/time stamp (07-02-2026).

3. Navigated to Launch Templates and clicked on Create launch template to define reusable EC2 configuration.

The screenshot shows the AWS EC2 Launch Templates page. On the left, there's a sidebar with 'EC2' selected. The main content area has a heading 'EC2 launch templates' and sub-headings 'Streamline, simplify and standardize instance launches'. Below this, a text box explains the purpose of launch templates. To the right, there's a call-to-action box with a 'New launch template' button and a 'Create launch template' button. At the bottom, there are sections for 'Benefits and features' (Streamline provisioning, Simplify permissions) and 'Documentation'.

4. Provided the launch template name and description and enabled Auto Scaling guidance to ensure compatibility.

The screenshot shows the 'Create launch template' wizard. The first step, 'Launch template name and description', is displayed. It includes fields for 'Launch template name - required' (containing 'myapp-temp'), 'Template version description' (containing 'A prod webserver for MyApp'), and 'Auto Scaling guidance' (with a checked checkbox for 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'). Below this, there are sections for 'Template tags' and 'Source template'. The right side of the screen shows the 'Summary' section, which includes 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. At the bottom right is a 'Create launch template' button.

5. Selected Amazon Linux AMI as the operating system for secure and optimized AWS performance.

The screenshot shows the 'Create launch template' wizard. In the 'Application and OS Images (Amazon Machine Image) - required' section, the 'Amazon Linux 2023 kernel-6.1 AMI' is selected. This AMI is described as 'Free tier eligible'. The 'Summary' panel on the right includes sections for Software Image (AMI), Virtual server type (instance type), Firewall (security group), and Storage (volumes). A large orange 'Create launch template' button is at the bottom right.

6. Chose t3.micro instance type to keep the deployment cost-effective and Free Tier eligible.

The screenshot shows the 'Create launch template' wizard. In the 'Instance type' section, the 't3.micro' instance type is selected. It is described as 'Free tier eligible' and has details about its family, vCPUs, memory, and current generation. The 'Summary' panel on the right is identical to the previous screenshot, showing the same configuration options. A large orange 'Create launch template' button is at the bottom right.

7. Selected an existing key pair to allow secure SSH access to EC2 instances.

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The current step is 'Key pair (login)'. A dropdown menu is open, showing several key pairs: 'Specify a custom value...', 'sample-2-ind', 'docker-mumbai-key' (which is selected and highlighted in blue), and 'Availability Zone'. Other options like 'Create new key pair' and 'Create new subnet' are also visible.

8. Created a security group allowing inbound SSH (22) and HTTP (80) traffic.

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. The current step is 'Firewall (security groups)'. A new security group named 'mysg' is being created. An inbound rule for TCP port 22 is listed under 'Inbound Security Group Rules'. The 'Create launch template' button is visible at the bottom right.

The screenshot shows the 'Create launch template' wizard in the AWS Management Console. On the left, under 'Inbound Security Group Rules', there are two security group rules defined:

- Security group rule 1 (TCP, 22, 0.0.0.0/0):** Type: ssh, Protocol: TCP, Port range: 22, Source type: Anywhere, Description: e.g. SSH for admin desktop.
- Security group rule 2 (TCP, 80, 0.0.0.0/0):** Type: HTTP, Protocol: TCP, Port range: 80, Source type: Anywhere, Description: e.g. SSH for admin desktop.

A warning message at the bottom left states: "⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." On the right, the 'Summary' section includes:

- Software Image (AMI):** Amazon Linux 2023 AMI 2023.10....read more
- Virtual server type (instance type):** t3.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

At the bottom right are 'Cancel' and 'Create launch template' buttons.

9. Added a user data script to install Docker and run a NGINX container automatically on instance launch.

The screenshot shows the 'Create launch template' wizard with the 'User data' section open. The user data script is a shell script that installs Docker and runs a NGINX container:#!/bin/bash
yum install -y docker
systemctl start docker
systemctl enable docker
docker run -d -p 80:80 --name my-web-server nginx

Below the script, there is a checkbox labeled "User data has already been base64 encoded". On the right, the 'Summary' section remains the same as in the previous step, including the selected AMI, instance type, security group, and storage.

10. Successfully created the EC2 launch template, which will be used by the Auto Scaling Group.

The screenshot shows the AWS Cloud Console interface for creating a launch template. At the top, a green success message states: "Successfully created myapp-temp(lt-016f7103550bbcfa2)". Below this, an "Actions log" section details the steps: "Initializing requests" (Succeeded), "Creating security groups" (Succeeded), "Creating security group rules" (Succeeded), and "Create Launch Template" (Succeeded). A "Next Steps" section includes links for "Launch an instance", "Create an Auto Scaling group from your template", and "Create Auto Scaling group". The bottom of the screen shows standard AWS navigation and status icons.

II) Target Group Creation

11. Navigated to Target Groups to define how traffic will be routed to EC2 instances.

The screenshot shows the AWS Cloud Console interface for managing target groups. The left sidebar is expanded to show categories like "Elastic Block Store", "Network & Security", "Load Balancing", and "Auto Scaling". Under "Load Balancing", the "Target Groups" option is selected. The main content area displays a table titled "Target groups" with columns for Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A message at the top states "No target groups" and "You don't have any target groups in ap-south-1". A prominent blue button labeled "Create target group" is located below the table. The bottom of the screen shows standard AWS navigation and status icons.

12. Created a new target group with target type set to Instance. Configured the target group to use HTTP protocol on port 80.

Create target group

A target group can be made up of one or more targets. Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Settings - immutable

Choose a target type and the load balancer and listener will route traffic to your target. These settings can't be modified after target group creation.

Target type

Indicate what resource type you want to target. Only the selected resource type can be registered to this target group.

- Instances

Supports load balancing to instances in a VPC. Integrate with Auto Scaling Groups or ECS services for automatic management.

Suitable for: ALB, NLB, GWLB
- IP addresses

Supports load balancing to VPC and on-premises resources. Facilitates routing to IP addresses and network interfaces on the same instance. Supports IPv6 targets.

Suitable for: ALB, NLB, GWLB
- Lambda function

Supports load balancing to a single Lambda function. ALB required as traffic source.

Suitable for: ALB
- Application Load Balancer

Allows use of static IP addresses and PrivateLink with an Application Load Balancer. NLB required as traffic source.

Suitable for: NLB

Target group name

Name must be unique per Region per AWS account.

my-tg

Accepts: a-z, A-Z, 0-9, and hyphen (-). Can't begin or end with hyphen. 1-32 total characters; Count: 5/32

13. Verified health check settings and created the target group successfully.

Review and create

Review your target group configuration before creating

Step 1: Target group details

Target group details

Name my-tg	Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1
VPC vpc-0387f1efb5c5f5cfef	IP address type IPv4		

Health check details

Health check protocol HTTP	Health check path /	Health check port traffic-port	Interval 30 seconds
Timeout 5 seconds	Healthy threshold 5	Unhealthy threshold 2	Success codes 200

Target group details | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#TargetGroup:targetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:673586849368:targetgroup/my-tg/650cecbc763f24a3...

Nilima (6735-8684-9368) Asia Pacific (Mumbai) Nilima

IAM VPC S3 EC2 Aurora and RDS Route 53 Certificate Manager Lambda Amazon EventBridge DynamoDB Simple Notification Service Key Management Service CloudFront CloudWatch

EC2 Target groups my-tg

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
- Trust Stores

Auto Scaling

- Auto Scaling Groups

Details

arn:aws:elasticloadbalancing:ap-south-1:673586849368:targetgroup/my-tg/650cecbc763f24a3

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-03871efb5c5f5fce

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
0	0	0	0	0	0

0 Anomalous

Targets Monitoring Health checks Attributes Tags

Registered targets (0)

Anomaly mitigation: Not applicable

Deregister Register targets

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 11:46 AM 07-02-2026

III) Load Balancer Creation

14. Navigated to Load Balancers and selected Create load balancer.

Load balancers | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LoadBalancers:

Nilima (6735-8684-9368) Asia Pacific (Mumbai) Nilima

IAM VPC S3 EC2 Aurora and RDS Route 53 Certificate Manager Lambda Amazon EventBridge DynamoDB Simple Notification Service Key Management Service CloudFront CloudWatch

EC2 Load balancers

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
- Trust Stores

Auto Scaling

- Auto Scaling Groups

Load balancers What's new?

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

Name	State	Type	Scheme	IP address type	VPC ID	Availability Zone
No load balancers	You don't have any load balancers in ap-south-1					

0 load balancers selected

CloudShell Feedback Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 11:46 AM 07-02-2026

15. Chose Application Load Balancer (ALB) to handle HTTP traffic at Layer 7.

The screenshot shows the AWS CloudFront console with the URL ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#SelectCreateELBWizard. The page title is "Compare and select load balancer type". It compares three types of load balancers:

- Application Load Balancer (ALB)**: Handles both HTTP and HTTPS traffic, routing it to Lambda functions, Application Load Balancers, or AWS API Gateways.
- Network Load Balancer (NLB)**: Handles TCP, UDP, and TLS traffic, routing it to Application Load Balancers, AWS Lambda functions, or AWS API Gateways.
- Gateway Load Balancer (GWLB)**: Handles traffic from VPC endpoints, routing it to AWS Lambda functions, AWS API Gateways, or AWS AppSync endpoints.

16. Configured the ALB as Internet-facing and selected the project VPC.

The screenshot shows the AWS CloudFront console with the URL ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateALBWizard. The page title is "Create application load balance". The "Basic configuration" section includes:

- Load balancer name**: "my-alb" (highlighted in blue).
- Scheme**: "Internet-facing" (selected, highlighted in blue). Description: "Serves internet-facing traffic, Has public IP addresses, DNS name resolves to public IPs, Requires a public subnet."
- Load balancer IP address type**: "IPv4" (selected, highlighted in blue). Description: "Includes only IPv4 addresses."
- Dualstack**: "Includes IPv4 and IPv6 addresses."
- Dualstack without public IPv4**: "Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with internet-facing load balancers only."

17. Selected multiple Availability Zones and subnets for high availability.

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS CloudFormation console. In the 'Network mapping' section, a VPC is selected. In the 'IP pools' section, the 'Use IPAM pool for public IPv4 addresses' checkbox is unchecked. Under 'Availability Zones and subnets', three subnets are selected: 'ap-south-1a (aps1-az1)', 'ap-south-1b (aps1-az3)', and 'ap-south-1c (aps1-az2)'. Each subnet is listed with its CIDR range: 172.31.0.0/20, 172.31.16.0/20, and 172.31.32.0/20 respectively.

18. Attached the previously created security group to the load balancer.

The screenshot shows the 'Create Application Load Balancer' wizard in the AWS CloudFormation console. In the 'Security groups' section, a security group named 'mysg' is selected. The security group details show it is associated with VPC 'vpc-0387f1efb5cf5f5f' and has an ID of 'sg-0f4ee808fa9fbbead'. The 'Select up to 5 security groups' dropdown also lists 'mysg'.

19. Configured the listener on HTTP port 80 and forwarded traffic to the target group.

The screenshot shows the 'Listeners and routing' configuration for a listener named 'HTTP:80'. The 'Protocol' is set to 'HTTP' and the 'Port' is '80'. Under 'Default action', the 'Forward to target groups' option is selected. A single target group, 'my-tg', is listed with a weight of 1 and a percent of 100%. The interface includes tabs for 'Info', 'Forward to target group', and 'Redirect to URL'.

20. Created the Application Load Balancer successfully.

The screenshot shows the 'my-alb' load balancer details page. A green success message states: 'Successfully created load balancer: my-alb. It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.' The 'Details' section shows the load balancer type as 'Application', status as 'Provisioning', scheme as 'Internet-facing', and VPC as 'vpc-0387f1efb5c5f5fce'. The 'Load balancer ARN' is listed as 'arn:aws:elasticloadbalancing:ap-south-1:673586849368:loadbalancer/app/my-alb/7209ba2...'. The 'DNS name Info' is 'my-alb-410556716.ap-south-1.elb.amazonaws.com (A Record)'. The left sidebar shows navigation links for EC2, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling.

IV) Auto Scaling Creation

21. Navigated to Auto Scaling Groups and clicked Create Auto Scaling group.

The screenshot shows the AWS EC2 Auto Scaling Groups page. On the left, there's a navigation sidebar with sections like Volumes, Snapshots, Lifecycle Manager, Network & Security, Load Balancing, and Auto Scaling. Under Auto Scaling, 'Auto Scaling Groups' is selected. The main content area features a large heading 'Amazon EC2 Auto Scaling' and a sub-headline 'helps maintain the availability of your applications'. Below this is a paragraph about Auto Scaling groups. To the right, there's a 'Create Auto Scaling group' button. At the bottom, there's a 'How it works' section with a diagram showing an 'Auto Scaling group' managing multiple EC2 instances, and a 'Pricing' section.

22. Provided the Auto Scaling group name and selected the previously created launch template.

The screenshot shows the 'Create Auto Scaling group' wizard at Step 1: 'Choose launch template'. On the left, a vertical navigation bar lists steps from Step 1 to Step 7. Step 1 is highlighted with a blue circle. The main content area has two sections: 'Name' and 'Launch template'. In the 'Name' section, the 'Auto Scaling group name' is set to 'my-asg'. In the 'Launch template' section, it says 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' A dropdown menu shows 'myapp-temp' selected. The bottom of the screen shows standard browser controls and a taskbar.

23. Selected the VPC and multiple subnets across different Availability Zones.

The screenshot shows the 'Create Auto Scaling group' wizard on the AWS Management Console. The current step is 'Step 2 - Choose instance launch options'. Under 'VPC', the VPC 'vpc-0387f1efb5c5fcfe' is selected. In the 'Availability Zones and subnets' section, three subnets are chosen: 'aps1-az1 (ap-south-1a) | subnet-04db1ff83b1caeba', 'aps1-az2 (ap-south-1c) | subnet-0feb3c3994f12cf2b', and 'aps1-az3 (ap-south-1b) | subnet-0375e18025a1586f2'. Under 'Availability Zone distribution', the 'Balanced best effort' option is selected. The bottom of the screen shows the AWS navigation bar and system status.

24. Attached the Auto Scaling Group to the existing Application Load Balancer target group.

The screenshot shows the 'Create Auto Scaling group' wizard on the AWS Management Console, currently at 'Step 3 - optional: Integrate with other services'. The 'Load balancing' section is active, showing the 'Attach to an existing load balancer' configuration. Under 'Select the load balancers to attach', the 'Choose from your load balancer target groups' option is selected, with 'my-tg | HTTP Application Load Balancer: my-alb' listed in the dropdown. The bottom of the screen shows the AWS navigation bar and system status.

25. Configured desired capacity as 2, minimum as 1, and maximum as 4 instances.

The screenshot shows the AWS Auto Scaling group creation wizard at Step 4: Configure group size and scaling. The 'Desired capacity' field is set to 2. The 'Min desired capacity' field is set to 1 and 'Max desired capacity' is set to 4. The 'Automatic scaling - optional' section is collapsed.

26. Enabled target tracking scaling policy based on average CPU utilization.

The screenshot shows the AWS Auto Scaling group creation wizard at Step 5: Automatic scaling - optional. The 'Target tracking scaling policy' option is selected, and the 'Metric type' is set to 'Average CPU utilization'. Other options like 'No scaling policies' and 'Scheduled scaling' are also shown.

27. Reviewed all configurations and created the Auto Scaling Group.

The screenshot shows the AWS Create Auto Scaling group wizard. It is on Step 5: Add notifications, which has a 'Notifications' section showing 'No notifications'. There is an 'Edit' button. Step 6: Add tags follows, with a 'Tags (0)' section showing 'No tags'. There is also an 'Edit' button. At the bottom, there are 'Preview code', 'Cancel', 'Previous', and 'Create Auto Scaling group' buttons. The top navigation bar shows the AWS logo, search bar, and various services like IAM, VPC, S3, EC2, Route 53, Certificate Manager, Lambda, Amazon EventBridge, DynamoDB, Simple Notification Service, Key Management Service, CloudFront, and CloudWatch. The top right corner shows the region as Asia Pacific (Mumbai) and a user profile named Nilima.

28. Verified that EC2 instances were automatically launched by the Auto Scaling Group.

The screenshot shows the AWS Instances page under the EC2 service. The left sidebar includes options like Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, and Elastic Block Store. The main area displays 'Instances (4) Info' with a table. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The data is as follows:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
docker	i-0280a02887f217165	Terminated	t3.micro	-	View alarms +	ap-south-1c	-
docker	i-0df7fb7191c6b6a42	Terminated	t3.micro	-	View alarms +	ap-south-1c	-
	i-07f02ac72d407101d	Running	t3.micro	Initializing	View alarms +	ap-south-1a	-
	i-0bb5c0e1e98b6b224	Running	t3.micro	Initializing	View alarms +	ap-south-1b	-

The bottom of the page includes standard AWS footer links and a timestamp of 11:53 AM on 07-02-2026.

V) Verify the created resources

29. Verified that the load balancer status became Active and copied its DNS name.

The screenshot shows the AWS EC2 Load Balancers console. On the left, there's a sidebar with 'EC2' selected, followed by 'Instances', 'Images', and 'Elastic Block Store'. The main area is titled 'my-alb' under 'Details'. It shows the load balancer type as 'Application', status as 'Active', scheme as 'Internet-facing', VPC as 'vpc-0387f1efb5c5f5cfe', and Availability Zones as 'subnet-0feb5c3994f12cf2b', 'subnet-04db1f5f83b1caeba', and 'subnet-0375e18025a1586f2'. The 'Listeners and rules' tab is selected, showing one rule. A tooltip 'DNS name copied' appears over the copied URL 'my-alb-410556716.ap-south-1.elb.amazonaws.com (A Record)'. The bottom navigation bar includes CloudShell, Feedback, and Console Mobile App.

30. Accessed the ALB DNS URL in a browser and confirmed the NGINX welcome page was displayed.

The screenshot shows a web browser window with the title 'Welcome to nginx!'. The address bar shows the URL 'my-alb-410556716.ap-south-1.elb.amazonaws.com'. The page content is the standard NGINX welcome message: 'Welcome to nginx!', 'If you see this page, the nginx web server is successfully installed and working. Further configuration is required.', 'For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.', and 'Thank you for using nginx.' The bottom navigation bar includes various icons for file operations and system status.

31. Manually stopped EC2 instances to test Auto Scaling behaviour.

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, 'Instances' is selected. In the main content area, two instances are listed:

Instance ID	Stop protection	Result
i-07f02ac72d407101d	Disabled	Can stop
i-0bb5c0e1e98b6b224	Disabled	Can stop

A yellow box highlights a note: "The following instances are attached to an Auto Scaling group:
• i-07f02ac72d407101d (my-asg)
• i-0bb5c0e1e98b6b224 (my-asg)" followed by a warning message: "When you stop an instance, Amazon EC2 Auto Scaling might terminate it and automatically launch a replacement. Termination can result in deletion of the EBS root volume. To prevent termination and replacement, before stopping the instance, detach it from the Auto Scaling group."

Below this, another note says: "Associated resources" with the sub-note "You will continue to incur charges for these resources while the instance is stopped". A yellow box highlights a warning: "You will be billed for associated resources" with the sub-note "After you stop the instance, you are no longer charged usage or data transfer fees for it. However, you will still be billed for associated Elastic IP addresses and EBS volumes."

At the bottom right, there are 'Cancel' and 'Stop' buttons. The status bar at the bottom right shows: ENG IN 12:14 PM 07-02-2026.

32. Observed that Auto Scaling automatically launched new instances to replace stopped ones.

The screenshot shows the AWS EC2 Instances page after stopping the instances from the previous step. A green banner at the top indicates: "Successfully initiated stopping of i-07f02ac72d407101d, i-0bb5c0e1e98b6b224".

The main content area displays the following table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 IP
docker	i-0df7fb7191c6b6a42	Terminated	t3.micro	-	View alarms +	ap-south-1c	-
	i-07f02ac72d407101d	Stopped	t3.micro	-	View alarms +	ap-south-1a	-
	i-02e0ee0ae4dca4285	Running	t3.micro	Initializing	View alarms +	ap-south-1b	-
	i-0bb5c0e1e98b6b224	Stopped	t3.micro	-	View alarms +	ap-south-1b	-

A yellow box highlights the newly launched instance: "i-02e0ee0ae4dca4285" which is "Running" and "Initializing".

At the bottom right, the status bar shows: ENG IN 12:16 PM 07-02-2026.

33. Checked Auto Scaling activity history, which confirmed instance termination and replacement events.

The screenshot shows the AWS Management Console for the Auto Scaling group 'my-asg'. The left sidebar has sections for Instances, Images, Elastic Block Store, and Network & Security. The main content area is titled 'Activity history (6)' and lists six entries:

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-066918252993713bb8	At 2026-02-07T06:47:50Z an instance was launched in response to an unhealthy instance needing to be replaced.	2026 February 07, 12:17:51 PM +05:30
Connection draining in progress	Terminating EC2 instance: i-07f02ac72d407101d - Waiting For ELB Connection Draining.	At 2026-02-07T06:47:49Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2026 February 07, 12:17:49 PM +05:30
Successful	Launching a new EC2 instance: i-02e0ee0ae4dca4285	At 2026-02-07T06:45:53Z an instance was launched in response to an unhealthy instance needing to be replaced.	2026 February 07, 12:15:55 PM +05:30
Connection	Terminating EC2 instance: i-		2026