

Part 1: EC2 with ELB and ASG

Objective: Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

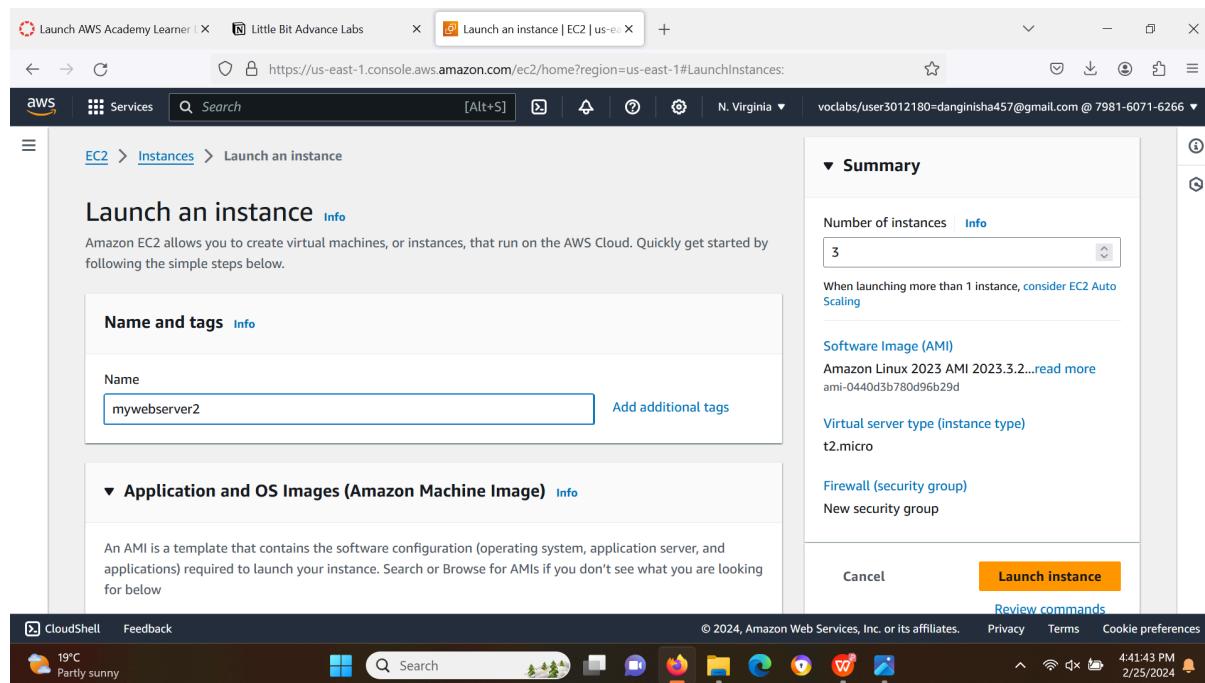
Approach:

- 1. Launch EC2 Instances:** Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).
- 2. Configure Load Balancer:** Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.
- 3. Set Up Auto Scaling Group (ASG):** Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.
- 4. Test Your Setup:** Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.
- 5. Verify Website Functionality:** Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations.

Goal: By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

Solution:

We start by launching a new instance.

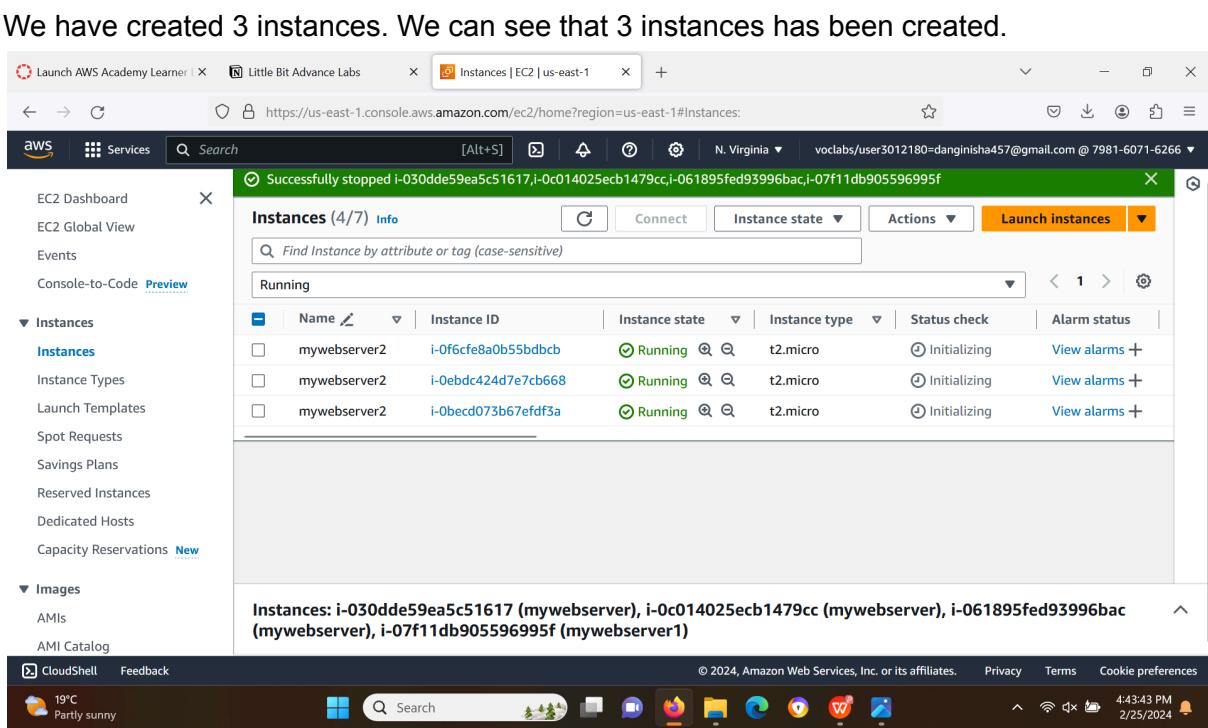
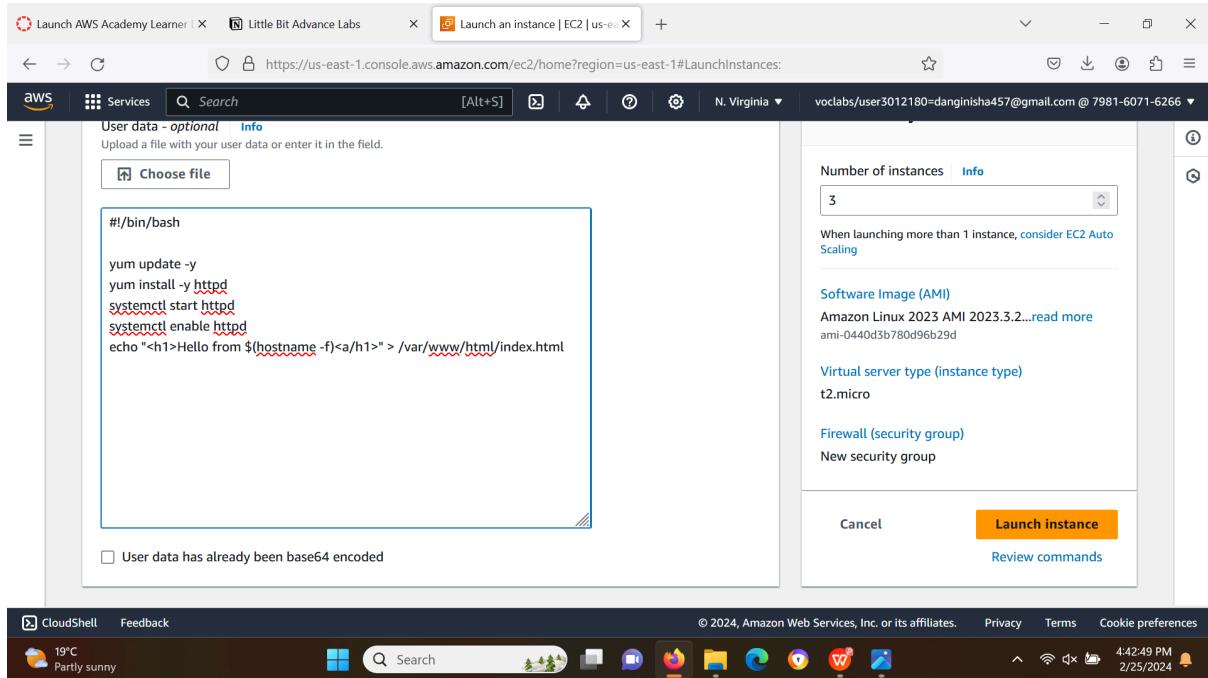


Assign a keypair.

The screenshot shows the AWS EC2 Launch Instances wizard. In the top left, there are three tabs: "Launch AWS Academy Learner", "Little Bit Advance Labs", and "Launch an instance | EC2 | us-east-1". The current tab is "Launch an instance | EC2 | us-east-1". The URL in the address bar is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances>. The main content area shows the "Instance type" section for the t2.micro instance type. It includes details like "Family: t2", "1 vCPU", "1 GiB Memory", "Current generation: true", and pricing information for On-Demand Windows, SUSE, RHEL, and Linux. Below this, a note says "Additional costs apply for AMIs with pre-installed software". To the right, a sidebar displays files: "mykp.pem" (Completed — 1.6 KB), "newkp.pem" (Completed — 1.6 KB), "Advanced Lab-1.pdf" (Completed — 44.5 MB), and "Advanced Lab.pdf" (Completed — 44.5 MB). At the bottom right of the main area is a large orange "Launch instance" button.

Select security groups.

The screenshot shows the AWS EC2 Launch Instances wizard at the "Configure security groups" step. The top navigation bar shows "Launch AWS Academy Learner", "Little Bit Advance Labs", and "EC2 | us-east-1". The URL is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances>. The main content area has a heading "Auto-assign public IP" with an "Info" link. Below it, there's a section for "Firewall (security groups)" with an "Info" link. It says "A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance." There are two radio buttons: "Create security group" (selected) and "Select existing security group". A note below says "We'll create a new security group called 'launch-wizard-2' with the following rules:". Under this, there are three checkboxes: "Allow SSH traffic from Anywhere" (selected), "Allow HTTPS traffic from the internet" (unchecked), and "Allow HTTP traffic from the internet" (selected). A warning message in a yellow box says: "⚠ 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." To the right, a sidebar shows "Virtual server type (instance type)" as t2.micro, "Firewall (security group)" as "New security group", and "Storage (volumes)" as 1 volume(s) - 8 GiB. At the bottom right is an orange "Launch instance" button.



Then, we create a target group.

The screenshot shows the AWS Lambda console with the title "Step 1 Create target group | EC2". The "Target group name" field contains "mytargetgroups". The "Protocol : Port" section shows "HTTP" selected as the protocol and "80" as the port. The "IP address type" section shows "IPv4" selected. A note below it states: "Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target." The "Include as pending" button is visible at the bottom.

Include as pending

The screenshot shows the AWS Lambda console with the title "Step 2 Create target group | EC2". The "Available instances (3/3)" table lists three instances: "mywebserver2" (Instance ID: i-0f6cfe8a0b55bdbcb), "mywebserver2" (Instance ID: i-0ebdc424d7e7cb668), and "mywebserver2" (Instance ID: i-0becd073b67efdf3a). All three instances are selected, as indicated by the checked checkboxes in the first column. Below the table, the "Ports for the selected instances" section shows "80" as the port. The "Include as pending below" button is visible at the bottom.

Review targets

Targets (3)

Instance ID	Name	Port	State	Security groups	Zone
i-0f6cfe8a0b55bdbcb	mywebserver2	80	Running	launch-wizard-3	us-east-1c
i-0ebdc424d7e7cb668	mywebserver2	80	Running	launch-wizard-3	us-east-1c
i-0becd073b67efdf3a	mywebserver2	80	Running	launch-wizard-3	us-east-1c

3 pending

Create target group

Then, we create a new load balancer. We select three addresses

Mappings Info

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

us-east-1a (use1-az2)

Subnet

subnet-0131cb674688d50ec

IPv4 address

Assigned by AWS

us-east-1b (use1-az4)

Subnet

subnet-01fe709618e3ec3bf

IPv4 address

Assigned by AWS

us-east-1c (use1-az6)

Subnet

subnet-09bed1161453896df

IPv4 address

Create load balancer

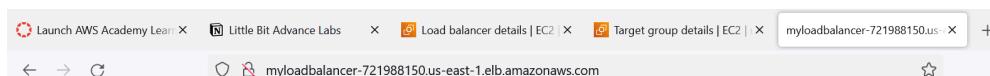
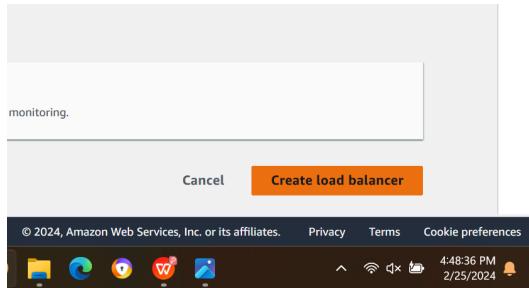
Assigning security groups.

The screenshot shows the AWS Cloud Console interface for creating an Application Load Balancer. The top navigation bar includes tabs for 'Launch AWS Academy Learner', 'Little Bit Advance Labs', and 'Create application load balance'. The main search bar shows 'load balancers'. The region is set to 'N. Virginia'. The user's email is listed as 'voclabs/user3012180=danginisha457@gmail.com @ 7981-6071-6266'. The current step is 'Security groups'. A note states: '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](#)'. Below this, a 'Security groups' section lists 'Select up to 5 security groups' with two items: 'default' (sg-0fcf966a02e573e08) and 'launch-wizard-3' (sg-0ac0cf1b13a6aafe). The 'Listeners and routing' step is partially visible below, showing a 'Listener HTTP:80' configuration with 'Protocol: HTTP' and 'Port: 80'. The 'Default action' dropdown is set to 'Forward to' and 'Select a target group'. The bottom of the screen shows the AWS navigation bar with 'CloudShell', 'Feedback', and various icons.

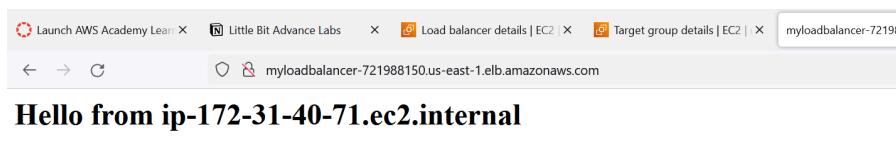
Assigning a target group that we created earlier.

The screenshot shows the 'Target group details' step of the Application Load Balancer wizard. The top navigation bar and user information are identical to the previous screenshot. The current step is 'Listeners and routing'. A note states: 'A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.' Below this, a 'Listener HTTP:80' configuration is shown with 'Protocol: HTTP' and 'Port: 80'. The 'Default action' dropdown is set to 'Forward to' and 'Select a target group'. A 'Create target' button is visible. A dropdown menu is open, showing a search bar with 'Q |' and a list item 'mytargetgroups' with the subtext 'Target type: Instance, IPv4'. At the bottom of the screen, there is an 'Add listener' button. The bottom of the screen shows the AWS navigation bar with 'CloudShell', 'Feedback', and various icons.

Then we finally create a load balancer.

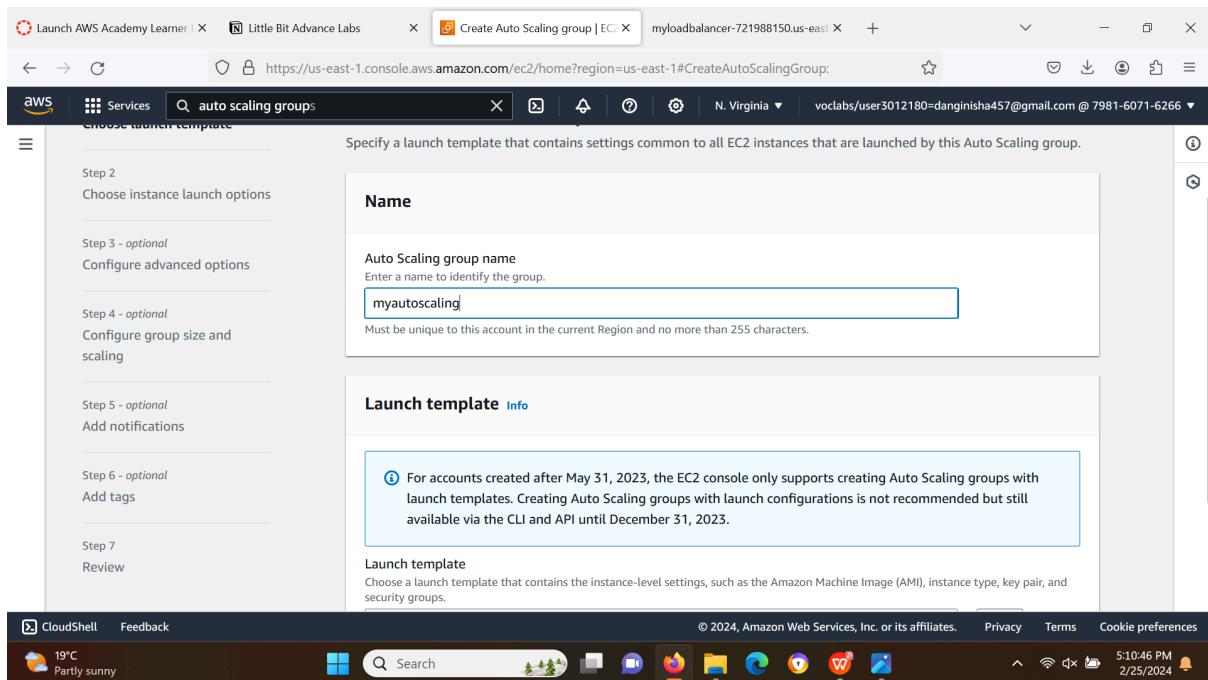


When we refresh, we can see a new IP.



Our load balancer is working fine.

Now, we create an auto scaling group.



For that we need to create a launch 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 'mytemplate1'. A note below it says: 'Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\"', '@''. The 'Template version description' field contains 'A prod webserver for MyApp'. A note below it says: 'Max 255 chars'. The 'Auto Scaling guidance' section has a link to 'Info' and a checkbox 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling'. The right panel shows sections for 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A tooltip for the 'Free tier' is visible: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch instances)'. At the bottom are 'Cancel' and 'Create launch template' buttons.

The screenshot shows the 'Create launch template' wizard on the AWS Management Console. The current step is 'Instance type'. The 'Instance type' dropdown is set to 't2.nano'. Below it, details show: Family: t2, 1 vCPU, 0.5 GiB Memory, Current generation: true. Pricing: On-Demand Linux base pricing: 0.0058 USD per Hour, On-Demand SUSE base pricing: 0.0058 USD per Hour, On-Demand Windows base pricing: 0.0081 USD per Hour. A note says: 'Additional costs apply for AMIs with pre-installed software'. The right panel shows sections for 'Software Image (AMI)', 'Virtual server type (instance type)', 'Firewall (security group)', and 'Storage (volumes)'. A tooltip for the 'Free tier' is visible: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch instances)'. At the bottom are 'Cancel' and 'Create launch template' buttons.

Enter a name to identify the group.
myautoscaling

Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

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.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

mytemplatee

Create a launch template Info

Version

Default (1)

Create a launch template version Info

CloudShell Feedback 19°C Partly sunny Search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 5:13:25 PM 2/25/2024

Select availability zones.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0340b4a91474b1f2e
172.31.0.0/16 Default

Create a VPC Info

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

Select Availability Zones and subnets

us-east-1c | subnet-09bed1161453896df
172.31.32.0/20 Default

us-east-1a | subnet-0131cb674688d50ec
172.31.80.0/20 Default

us-east-1b | subnet-01fe709618e3ec3bf
172.31.16.0/20 Default

Create a subnet Info

Cancel Skip to review Previous Next

CloudShell Feedback 17°C Mostly cloudy Search © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 5:13:59 PM 2/25/2024

Attach existing load balancer, choose load balancer from target groups and select target groups.

The screenshot shows the AWS Cloud Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup>. The page is titled "Create Auto Scaling group | EC2". The left sidebar shows steps 4 through 7. Step 4 is expanded, showing options for attaching to a load balancer: "No load balancer" (radio button), "Attach to an existing load balancer" (selected radio button), and "Attach to a new load balancer". The "Attach to an existing load balancer" section includes a sub-section "Attach to an existing load balancer" with a dropdown menu "Select target groups" containing "mytargetgroups | HTTP" and "Application Load Balancer: myloadbalancer". The status bar at the bottom shows "CloudShell Feedback", the date "2/25/2024", and the time "5:14:58 PM".

Select scaling

The screenshot shows the AWS Cloud Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup>. The page is titled "Create Auto Scaling group | EC2". The left sidebar shows steps 4 through 7. Step 5 is expanded, showing "Configure group size and scaling". The "Desired capacity type" section includes a dropdown menu "Units (number of instances)" and a "Desired capacity" input field set to "1". The "Scaling" section includes "Scaling limits" with "Min desired capacity" set to "1" and "Max desired capacity" set to "4". The status bar at the bottom shows "CloudShell Feedback", the date "2/25/2024", and the time "5:15:16 PM".

Click next and then finally we can create auto scaling groups.

AWS Lambda | X Little Bit Advance Labs | X Create Auto Scaling group | EC2 | X Create launch template | EC2 | X +

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:

Services auto scaling groups N. Virginia voclabs/user3012180=danginisha457@gmail.com @ 7981-6071-6266

Step 5: Add notifications

Notifications

No notifications

Step 6: Add tags

Tags (0)

Key Value Tag new instances

No tags

Cancel Previous Create Auto Scaling group

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

17°C Sunset coming

Search

5:15:40 PM 2/25/2024

Key	Value	Tag new instances
No tags		