

AWS Solutions Architect Assignment

SUBMITTED BY:

NAME : Avinash Komarabathini

Date : 22 December 2022

Enrolled course: Cloud and DevOps job guarantee Program

Module-3: ELB Assignment – 1

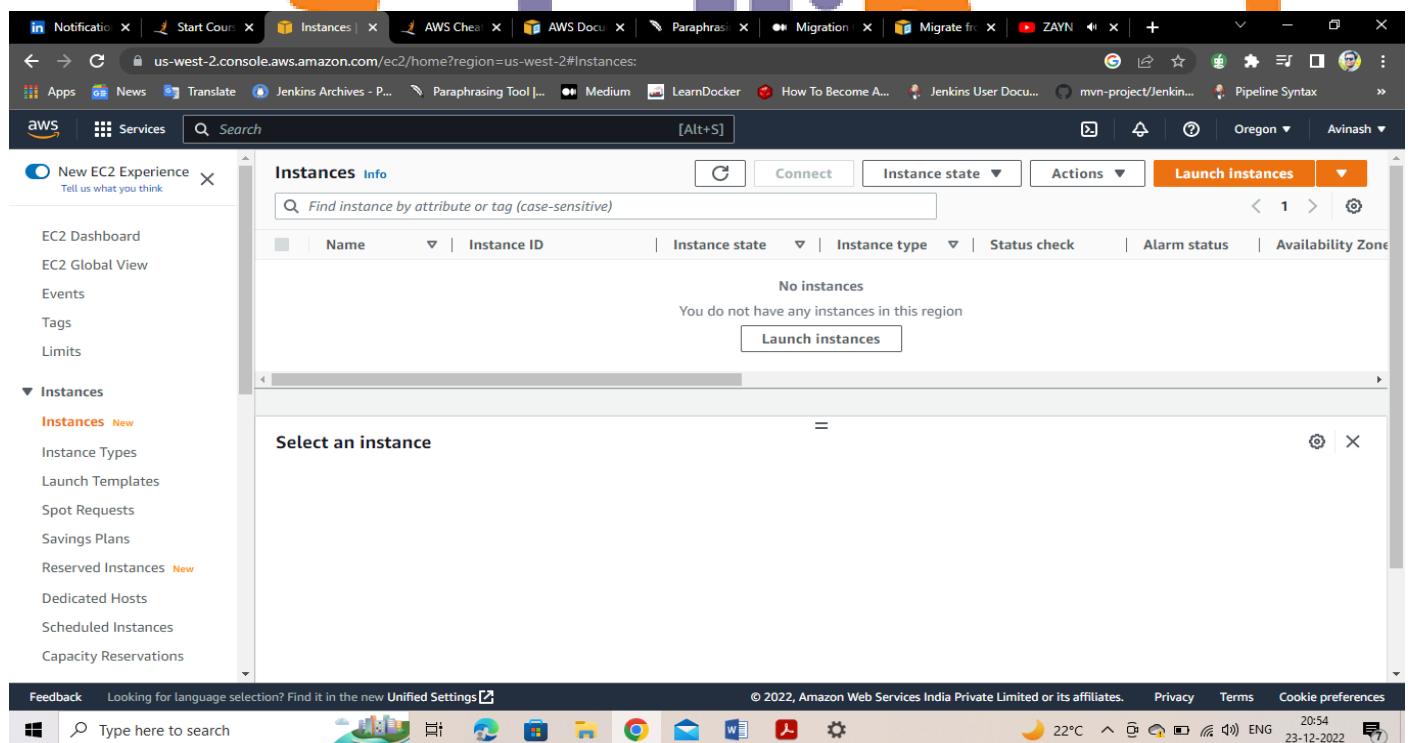
You have been asked to:

1. Create a Classic Load Balancer and register 3 EC2 instances with different web pages running in them
2. Migrate the Classic Load Balancer into an Application Load Balancer

1. Create a Classic Load Balancer and register 3 EC2 instances with different web pages running in them

I will first create three EC2 instances, each with a unique web page and title that corresponds to the web servers.

Go to the AWS console > EC2 dashboard > instance.



The screenshot shows the AWS EC2 Instances page. The top navigation bar includes links for Notifications, Start Course, Instances, AWS Cheat Sheet, AWS Documentation, Paraphrase, Migration, Migrate from, ZAYN, and more. The main content area is titled "Instances Info" and displays a search bar with placeholder text "Find instance by attribute or tag (case-sensitive)". Below the search bar is a table header with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. A message below the table states "No instances" and "You do not have any instances in this region". At the bottom of the table is a "Launch instances" button. On the left side, there is a sidebar with sections for EC2 Dashboard, EC2 Global View, Events, Tags, Limits, and Instances. Under the Instances section, there are links for Instances (selected), New, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances (New), Dedicated Hosts, Scheduled Instances, and Capacity Reservations. The bottom of the page features a feedback link, a search bar, and standard footer links for Privacy, Terms, and Cookie preferences, along with system status icons.

Give the required details and click on launch

The screenshot shows the AWS EC2 Launch Instances page. A search bar at the top has "Search" and "[Alt+S]" placeholder text. Below it, a navigation bar includes "AWS" and "Services". A sidebar on the left lists AMIs: Amazon Linux, macOS, Ubuntu (selected), Windows, Red Hat, and Suse. A search bar for "Browse more AMIs" is present. The main area shows "Ubuntu Server 22.04 LTS (HVM), SSD Volume Type" selected, with "Free tier eligible" status. Below this, "Description" and "Architecture" (64-bit x86) are listed. To the right, a summary panel shows "Number of instances" set to 3, with a note about using EC2 Auto Scaling for more than one instance. It also lists the "Software Image (AMI)" as Canonical, Ubuntu, 22.04 LTS, and specifies the instance type as t2.micro, which is part of a security group and uses storage volumes. At the bottom right are "Cancel" and "Launch Instance" buttons.



The screenshot shows the AWS EC2 Instances page. A sidebar on the left includes "New EC2 Experience" (with a feedback link), "EC2 Dashboard", "EC2 Global View", "Events", "Tags", "Limits", and "Instances" (selected). Under "Instances", there are links for "Instances", "Instance Types", "Launch Templates", "Spot Requests", "Savings Plans", "Reserved Instances", "Dedicated Hosts", "Scheduled Instances", and "Capacity Reservations". The main area displays a table titled "Instances (3) Info" with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Three instances are listed: "webserver1" (i-015ba99ddef8858c3, Running, t2.micro, Initializing, No alarms, us-east-1b), "webserver2" (i-0caa67b8bcb6e06e3, Running, t2.micro, Initializing, No alarms, us-east-1b), and "webserver3" (i-03b22db2eb3ec5931, Running, t2.micro, Initializing, No alarms, us-east-1b). At the bottom, a modal window titled "Select an instance" is open, showing the same list of instances. The status bar at the bottom indicates "Feedback", "Type here to search", "© 2022, Amazon Web Services India Private Limited or its affiliates.", "Privacy", "Terms", "Cookie preferences", and a timestamp "23-12-2022 20:58".

Now that the instances have been installed, we need to connect using the putty app and change the html page to use the new names.

- Webserver 1
- Webserver 2
- Webserver 3

For webserver 1

```
ubuntu@ip-172-31-94-23: ~
Last login: Fri Dec 23 15:39:49 UTC 2022 from 172.31.94.23
[1]:~# 
[1]:~# login as: ununtu
[1]:~# Server refused our key
[1]:~# 
[1]:~# Authenticating with public key "Ec2demo"
[1]:~# Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Fri Dec 23 15:39:49 UTC 2022

System load: 0.0 Processes: 97
Usage of /: 19.8% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.94.23
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-94-23: ~
```



For Webserver 2

```
ubuntu@ip-172-31-94-23: ~
Last login: Fri Dec 23 15:39:49 UTC 2022 from 172.31.94.23
[1]:~# 
[1]:~# login as: ununtu
[1]:~# Server refused our key
[1]:~# 
[1]:~# Authenticating with public key "Ec2demo"
[1]:~# Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Fri Dec 23 15:39:49 UTC 2022

System load: 0.0 Processes: 97
Usage of /: 19.8% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.94.23
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (use "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-94-23: ~
```



For Webserver 3

```
ubuntu@ip-172-31-94-23:~$ 
Last login: Fri Dec 23 15:39:49 UTC 2022
Server refused our key
Authenticating with public key "Ec2demo"
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-1026-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Dec 23 15:39:49 UTC 2022

System load: 0.09 0.09 0.07
Usage of /: 0.95% of 7.57GB  Users logged in: 0
Memory usage: 0%  IPv4 address for eth0: 172.31.94.23
Swap usage: 0% 

0 updates can be applied immediately.

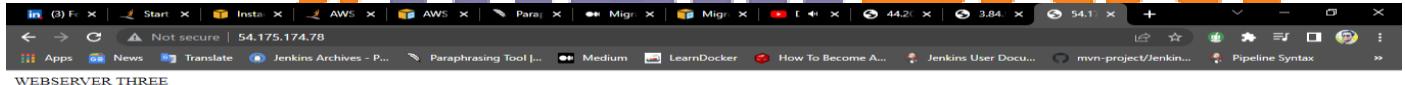
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Run "sudo -s" as administrator (use "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-94-23:~$
```



I have successfully created the 3 EC2 instances with different title names and now I will create the classic load balancer and register this 3 EC2 instances.

For creating the classic load balancer, we need to go the Load Balancing > load balance > create load balancer > create classic load balancer.

The screenshot shows the AWS Management Console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers>. The sidebar on the left has 'Load Balancing' expanded, with 'Load Balancers' selected. The main content area is titled 'Load balancers' and contains a message: 'Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.' Below this is a table with columns: Name, DNS name, State, VPC ID, and Availability Zones. A search bar and filter button are above the table. At the bottom right of the table area is a large orange 'Create load balancer' button.

Click on create.

The screenshot shows the 'Create' step of the 'Classic Load Balancer - previous generation' wizard. The title bar says 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SelectCreateELBWizard'. The main content area shows a diagram of a load balancer (CLB) connected to four EC2 instances. The instances are labeled with protocols: HTTP, HTTPS, TCP, and SSL. A note on the right says 'AWS will be retiring the EC2-Classic network on August 15, 2022.' There is also a 'Create' button.

Give the required details and click on next.

The screenshot shows 'Step 1: Define Load Balancer' of the 'Create ELB Wizard'. The title bar says 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateELBWizard'. The main content area has tabs: 1. Define Load Balancer, 2. Assign Security Groups, 3. Configure Security Settings, 4. Configure Health Check, 5. Add EC2 Instances, 6. Add Tags, 7. Review. The 'Define Load Balancer' tab is active. It includes sections for 'Basic Configuration' (warning about retiring EC2-Classic), 'Load Balancer name' (set to 'EL_for_classic'), 'Create LB Inside' (set to 'My Default VPC (172.31.0.0/16)'), 'Create an internal load balancer' (checkbox unchecked), 'Enable advanced VPC configuration' (checkbox unchecked), and 'Listener Configuration' (Protocol: HTTP, Port: 80, Instance Protocol: HTTP, Instance Port: 80). At the bottom are 'Cancel' and 'Next: Assign Security Groups' buttons.

Select the security group and click on next

Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-00fd8fc489cda7	default	default VPC security group	Copy to new
sg-04c247bf0e54f6789	launch-wizard-1	launch-wizard-1 created 2022-12-20T05:48:33.867Z	Copy to new

Cancel Previous Next: Configure Security Settings

Click on next.

Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

Ping Protocol: HTTP
Ping Port: 80
Ping Path: /index.html

Advanced Details

Response Timeout	5	seconds
Interval	30	seconds
Unhealthy threshold	2	
Healthy threshold	10	

Cancel Previous Next: Add EC2 Instances

Add ec2 instances and click on next.

Step 5: Add EC2 Instances

VPC: vpc-0ce126fdf8b90001 (172.31.0.0/16)

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-015ba99ddef8858c3	webserver1	running	launch-wizard-1, default	us-east-1b	subnet-0eff8547...	172.31.80.0/20
i-0caa67b8bc6e06e3	webserver2	running	launch-wizard-1, default	us-east-1b	subnet-0eff8547...	172.31.80.0/20
i-03b22db2eb3ec5931	webserver3	running	launch-wizard-1, default	us-east-1b	subnet-0eff8547...	172.31.80.0/20

Availability Zone Distribution: 3 instances in us-east-1b

Enable Cross-Zone Load Balancing
Enable Connection Draining (300 seconds)

Cancel Previous Next: Add Tags

Click on review and create.

The screenshot shows the AWS Lambda Step 6: Add Tags page. At the top, there is a navigation bar with tabs: 1. Define Load Balancer, 2. Assign Security Groups, 3. Configure Security Settings, 4. Configure Health Check, 5. Add EC2 Instances, 6. Add Tags, and 7. Review. The '6. Add Tags' tab is currently selected. Below the tabs, a section titled 'Step 6: Add Tags' contains the instruction 'Apply tags to your resources to help organize and identify them.' A note below states: 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.' There is a table with two columns: 'Key' and 'Value'. A 'Create Tag' button is located at the bottom left of the table area. At the bottom right, there are 'Cancel', 'Previous', and 'Review and Create' buttons.

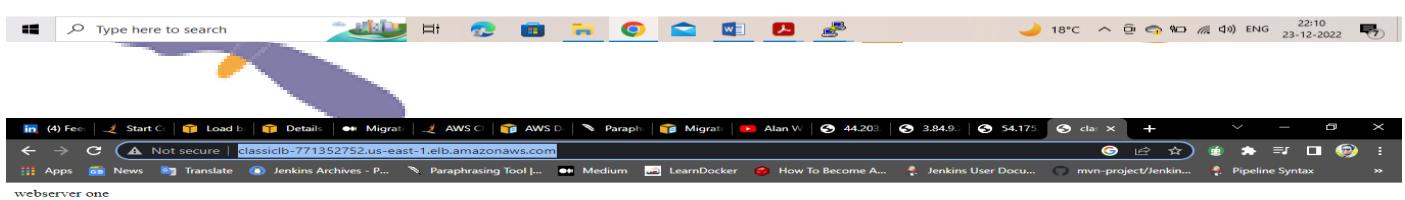
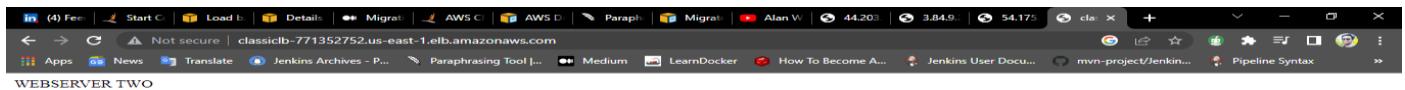
Click on Create

This screenshot is identical to the one above, showing the 'Add Tags' step of the Lambda creation wizard. The 'Create Tag' button is highlighted with a red box, indicating where the user should click to proceed.

Load Balancer Creation Status

The screenshot shows the 'Load Balancer Creation Status' page. It displays a green success message box with a checkmark icon. The message reads: 'Successfully created load balancer' and 'Load balancer classicLB was successfully created.' Below this, a note says: 'Note: It may take a few minutes for your instances to become active in the new load balancer.' At the bottom right of the message box is a 'Close' button.

The screenshot shows the Windows taskbar at the bottom of the screen. The status bar displays the date and time as '21:50 23-12-2022' and the network connection status as 'ENG'. Other icons for various applications like File Explorer, Task View, and Start are visible.



2. Migrate the Classic Load Balancer into an Application Load Balancer

For migrating the classic LB into ALB, we need to select the CLB which we have created > click on classic load balancer, it will redirect to the page there you see the migration option.

The screenshot shows the AWS EC2 Load Balancers page. On the left, a sidebar lists various EC2 services like Dashboard, Global View, Instances, and more. The main area displays a table titled 'Load balancers (1 / 1)'. A single row is present with the following details:

Name	State	VPC ID	Availability Zones
classicLB-641664123.us-east-2.amazonaws.com	-	vpc-0ce126fdfe8b90001	6 Availability Zones

An 'Actions' dropdown menu is open over the first row, showing options: Create load balancer, Edit IP address, Edit subnets, Edit instances, Edit health check settings, Edit listener, Edit security groups, Edit load balancer attributes, Manage tags, and Delete load balancer.

Click on migration > launch ALB migration wizard

The screenshot shows the AWS EC2 Load Balancers page for a specific CLB named 'classicLB'. The sidebar on the left is identical to the previous screenshot. The main area has a tab bar with 'Description', 'Instances', 'Health check', 'Listeners', 'Monitoring', 'Tags', and 'Migration'. The 'Migration' tab is currently selected. Below the tabs, a message reads: 'Migrate this Classic Load Balancer to a next generation load balancer. See [Comparison of Elastic Load Balancing Products](#)'. A prominent blue button labeled 'Launch ALB Migration Wizard' is centered at the bottom of this section.

Click on create.

The screenshot shows the AWS ELB Migration Wizard Step 6: Review page. The URL is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ELBMigrationWizard?type=application&clbName=classicLB>. The page displays the following details:

- Load balancer:** classicLB, Scheme: internet-facing, Listeners: Port:80 - Protocol:HTTP, IP address type: ipv4, VPC: vpc-0ce126fdfe8b90001, Subnets: subnet-013c86b2a0dd5c481, subnet-04bf1dda9224ba529, subnet-08e8474e12b4fb589, subnet-0d8cf62dcccb1bedd, subnet-0eff8547986aadc11, subnet-0f80b9b8988b69daf, Tags
- Security groups:** sg-00fd8fcfcc489cda7, sg-04c247bf0e54f6789
- Routing:** Edit

The page also includes a note: "The highlighted fields below indicate the new values (green) and the original values (grey)."

Below the wizard, the Windows taskbar shows the intelliPaat logo, indicating the browser is being used on a Windows machine.

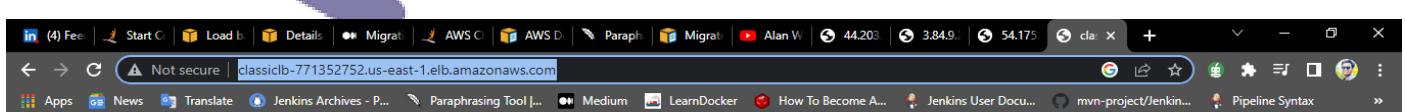
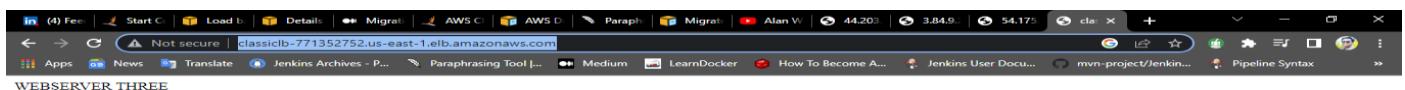
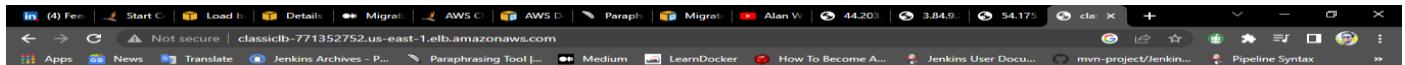
We can see in the down that we have successfully migrated the CLB to ALB.

The screenshot shows the AWS EC2 Load Balancers page. The URL is <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers>. The page displays the following table of load balancers:

Name	State	VPC ID	Availability Zones	Type	Created At
CLB-641664123.us-east-amazonaws.com	-	vpc-0ce126fdfe8b90001	6 Availability Zones	classic	December 23, 2022, 21:50 (UTC+05:30)
CLB-771352752.us-east-amazonaws.com	Active	vpc-0ce126fdfe8b90001	6 Availability Zones	application	December 23, 2022, 22:00 (UTC+05:30)

The left sidebar shows the EC2 dashboard, global view, events, tags, limits, instances, and other EC2 services.

At the bottom, the Windows taskbar shows the intelliPaat logo.



Module-2: Auto Scaling Assignment - 2

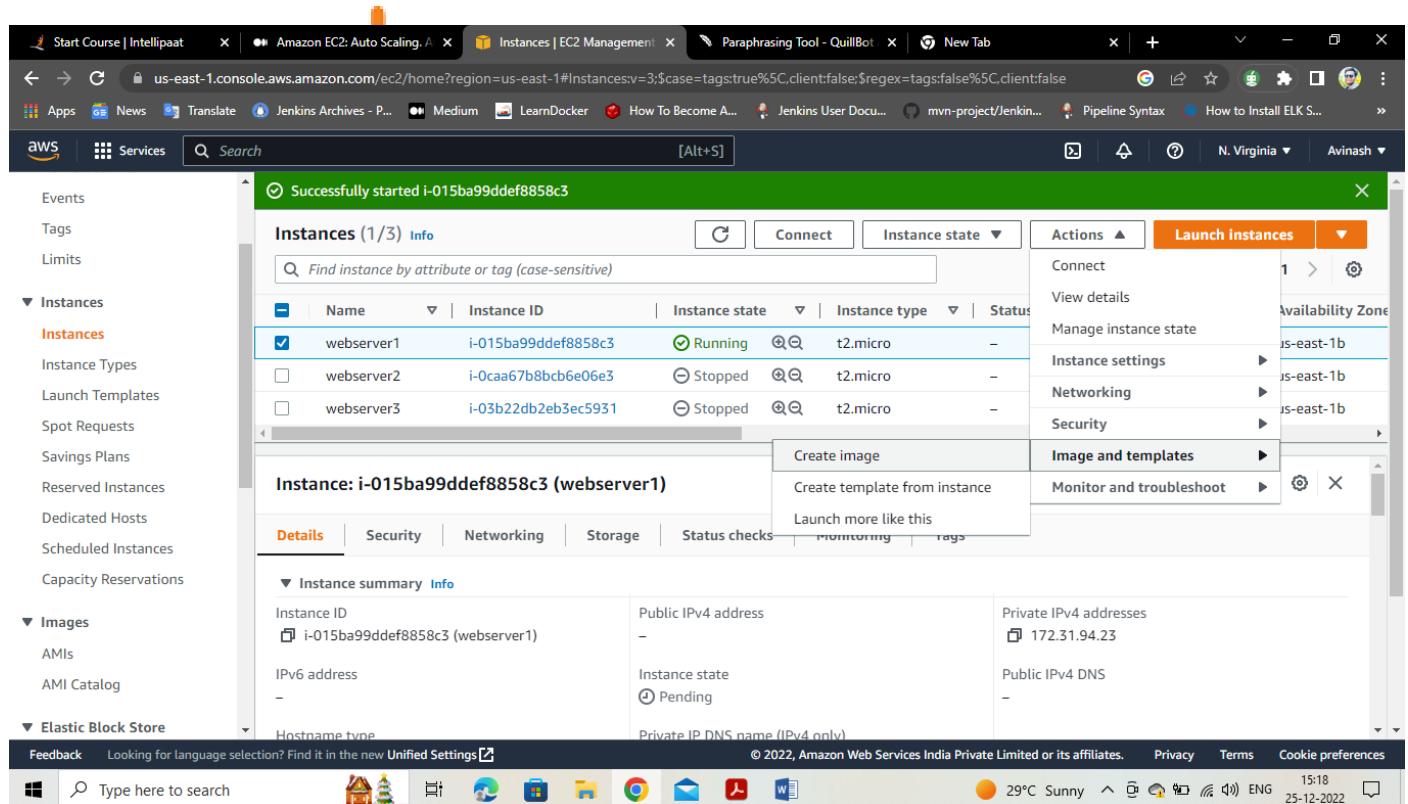
You have been asked to:

1. Create a Web Server AMI with Apache 2 server running in it
2. Create a Launch Configuration with this AMI
3. Use this Launch Configuration to create an Auto Scaling group with 1 minimum and 3 maximum instances

1. Create a Web Server AMI with Apache 2 server running in it

I already have a web server; I just need to integrate it with the AMI.

So, go to the amazon portal > EC2 dashboard > select the web server > click on action >image and templates >create image.



The screenshot shows the AWS EC2 Instances page. There are three instances listed:

Name	Instance ID	Instance state	Instance type	Status
webserver1	i-015ba99ddef8858c3	Running	t2.micro	-
webserver2	i-0caa67b8bcbb6e06e3	Stopped	t2.micro	-
webserver3	i-03b22db2eb3ec5931	Stopped	t2.micro	-

The 'Actions' dropdown for the selected instance 'webserver1' is open, showing the following options:

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Create image
- Image and templates
- Monitor and troubleshoot

A green banner at the top of the page says "Successfully started i-015ba99ddef8858c3".

Give the required details and click on create image.

The screenshot shows the 'Create image' step in the AWS EC2 console. It includes fields for the instance ID (i-015ba99ddef8858c3), image name (AMiforalaunchconfig), and image description (optional). Under 'Instance volumes', a single EBS volume is configured with a size of 8 GiB, IOPS of 100, and throughput of 100. The 'Delete on termination' and 'Encrypted' checkboxes are checked. A note at the bottom states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.'

The screenshot shows the 'Create image' step in the AWS EC2 console. It includes fields for the instance ID (i-015ba99ddef8858c3), image name (AMiforalaunchconfig), and image description (optional). Under 'Instance volumes', a single EBS volume is configured with a size of 8 GiB, IOPS of 100, and throughput of 100. The 'Delete on termination' and 'Encrypted' checkboxes are checked. A note at the bottom states: 'During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.' Below the volume configuration, there are two sections for tags: 'Tag image and snapshots together' (selected) and 'Tag image and snapshots separately'. A note says: 'If you assign a tag 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.'

We have successfully created the AMI.

The screenshot shows the AWS EC2 Home page. On the left, a navigation menu includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', and 'Images'. Under 'Images', 'AMIs' is selected. The main content area displays the 'Amazon Machine Images (AMIs) (1/1)' section. It shows a table with one row for the newly created AMI, identified by its ID (ami-08a438a4227d2e9b6), name (AMiforalaunchconfig), source (477385823196/AMiforalaunchconfig), and owner (477385823196). The table has columns for 'Name', 'AMI ID', 'AMI name', 'Source', and 'Owner'. Below the table, a detailed view for the AMI ID ami-08a438a4227d2e9b6 is shown, with tabs for 'Details', 'Permissions', 'Storage', and 'Tags'. The 'Details' tab displays information such as AMI ID, Image type (machine), Platform details (Linux/UNIX), Root device type (EBS), AMI name, Owner account ID, Architecture (x86_64), Usage operation (RunInstances), Root device name (/dev/sda1), Status (Available), Source, and Virtualization type (hvm).

2. Create a Launch Configuration with this AMI

To create a launch configuration with this AMI, go the auto scaling > launch configuration >create a **Launch Configuration**.

The screenshot shows the AWS EC2 Management Console with the 'Launch configurations' page. A prominent message at the top reads: 'Recommendation to not use launch configurations. Amazon EC2 Auto Scaling no longer adds support for new EC2 features to launch configurations and will stop supporting new EC2 instance types after December 31, 2022. We recommend that customers using launch configurations migrate to launch templates. For more information, see the documentation.' Below this, a table titled 'Launch configurations (0)' shows columns for Name, AMI ID, Instance type, Spot price, and Creation time. A large button labeled 'Create launch configuration' is located at the bottom right of the table area.

Give the required details click on launch

The screenshot shows the 'Create launch configuration' wizard. It includes fields for 'Launch configuration name' (with 'Name' set to 'ICami'), 'Amazon machine image (AMI)' (with 'AMI' set to 'AMIforalaunchconfig'), and 'Instance type' (set to 't2.micro'). A note at the top of the wizard page advises using launch templates instead of launch configurations. A 'Create launch template' button is also visible.

Choose AMI which you have created

The screenshot shows the 'Choose instance type' dialog box. It lists several instance types: t2.2xlarge, t2.large, t2.xlarge, t2.medium, and t2.small. The 't2.micro' row is highlighted with a blue border, indicating it is selected. Other rows show details like vCPUs, Memory, Storage, EBS optimized status, and Network performance.

Click on the create launch configuration.

The screenshot shows the 'Create launch configuration' wizard. In the 'Rules' section, a single rule is defined: 'All traffic' on 'All' protocol and port range '0 - 65535' from 'Anywhere' source. A warning message 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.' In the 'Key pair (login)' section, the 'Choose an existing key pair' dropdown is set to 'Ec2demo'. A checkbox is checked, acknowledging the user's access to the private key file. At the bottom right are 'Cancel' and 'Create launch configuration' buttons.

The screenshot shows the 'Launch configurations' page with one entry: 'ICami'. The configuration details are as follows:

AMI ID	Instance type	IAM instance profile
ami-08a438a4227d2e9b6	t2.micro	-
Kernel ID	Key name	Monitoring
-	Ec2demo	false
EBS optimized	Security groups	Spot price
false	sg-0285d062c7f7f29aa	-
Create time	RAM disk ID	IP address type
Sun Dec 25 2022 15:59:33 GMT+0530 (India Standard Time)	-	Default

A blue banner at the top left provides a recommendation: 'Recommendation to not use launch configurations' stating that EC2 Auto Scaling no longer adds support for new EC2 features to launch configurations and will stop supporting new EC2 instance types after December 31, 2022. It suggests migrating to launch templates. A green banner at the top right indicates the configuration was successfully created.

3. Use this Launch Configuration to create an Auto Scaling group with 1 minimum and 3 maximum instances

Go to the Auto scaling groups > create auto scaling group.

The screenshot shows the AWS EC2 Auto Scaling landing page. On the left, there's a navigation sidebar with options like Capacity Reservations, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. Under Auto Scaling, 'Launch Configurations' and 'Auto Scaling Groups' are listed. The main content area features a large heading 'Amazon EC2 Auto Scaling helps maintain the availability of your applications'. Below it, a section titled 'How it works' contains a diagram showing an 'Auto Scaling group' with two solid boxes labeled 'Minimum size' and two dashed boxes labeled 'Scale out as needed'. To the right, there are sections for 'Pricing' and 'Getting started'.

Give the required details and click on next.

The screenshot shows the 'Step 3 (optional)' configuration screen for creating an Auto Scaling group. On the left, there are optional steps: Step 4 (optional) to configure group size and scaling policies, Step 5 (optional) to add notifications, Step 6 (optional) to add tags, and Step 7 Review. The main form has a field 'Auto Scaling group name' with the value 'AGLC'. A note below it says: 'Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, see the documentation.' Below this, there's a 'Launch configuration' section with a dropdown set to 'ICami', and detailed fields for AMI ID, Instance type, Security groups, Date created, and Key pair name.

Select the network as default and click on next.

The screenshot shows the 'Step 4 (optional)' configuration screen for creating an Auto Scaling group. On the left, there are optional steps: Step 4 (optional) to configure group size and scaling policies, Step 5 (optional) to add notifications, Step 6 (optional) to add tags, and Step 7 Review. The main form has a 'VPC' section where 'vpc-0ce126fdf8b90001' is selected from a dropdown. Below it, there's a 'Create a VPC' link. The 'Availability Zones and subnets' section shows a list of subnets: us-east-1a | subnet-013c86b2a0dd5c481, us-east-1b | subnet-0eff8547986aadc11, us-east-1c | subnet-0d8cf62dcccb1bedd, us-east-1d | subnet-04bf1dda9224ba529, us-east-1e | subnet-08e8474e12b4fb589, and us-east-1f | subnet-0f80b9b8988b69daf. Each subnet entry includes its IP range and a 'Default' label.

I have not taken load balancer, click on next.

This screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 6 (optional). The left sidebar lists various EC2 services like Instances, Images, and Elastic Block Store. The main panel has two sections: 'Health checks - optional' and 'Additional settings - optional'. In 'Health checks', there's a checkbox for 'EC2' (unchecked) and 'ELB' (unchecked). A note says EC2 Auto Scaling automatically replaces instances that fail health checks. In 'Additional settings', there's a checkbox for 'Enable group metrics collection within CloudWatch' (unchecked) and a note about CloudWatch metrics for new instances. At the bottom are 'Cancel', 'Previous', 'Skip to review' (disabled), and 'Next' buttons.

As task suggested I have taken 1 minimum and 3 maximum instances

This screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 4 (optional). The left sidebar lists various EC2 services. The main panel has six steps: Step 1 (Choose launch template or configuration), Step 2 (Choose instance launch options), Step 3 (optional) Configure advanced options, Step 4 (optional) Configure group size and scaling policies (selected), Step 5 (optional) Add notifications, and Step 6 (optional) Add tags. Step 4 is expanded to show 'Group size - optional' with fields for Desired capacity (2), Minimum capacity (1), and Maximum capacity (3). It also includes a section for 'Scaling policies - optional' with a note about dynamically resizing the group. At the bottom are 'Cancel', 'Previous', 'Skip to review' (disabled), and 'Next' buttons.

Click on next..

This screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 5 (optional). The left sidebar lists various EC2 services. The main panel has six steps: Step 5 (optional) Add notifications, Step 6 (optional) Add tags, Step 7 Review. Step 5 is expanded to show 'Scaling policies - optional' with a note about dynamically resizing the group. It includes two options: 'Target tracking scaling policy' (radio button unchecked) and 'None' (radio button checked). It also includes an 'Instance scale-in protection - optional' section with a note about protecting newly launched instances and a checkbox for 'Enable instance scale-in protection' (unchecked). At the bottom are 'Cancel', 'Previous', 'Skip to review' (disabled), and 'Next' buttons.

Click on next

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 5: Add notifications. The left sidebar includes links for EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area has a heading "Add notifications" with a link to "Info". A note says "Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group." A "Add notification" button is present. Below it, there are sections for Step 1 through Step 7, each with a "Configure" link. Step 7 is labeled "Review". At the bottom right are "Cancel", "Previous", "Skip to review", and "Next" buttons.

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 6: Add tags. The left sidebar is identical to the previous screenshot. The main content area has a heading "Add tags" with a link to "Info". A note says "Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched." A callout box contains the text: "You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group." A "Tags (0)" section shows an "Add tag" button and a note "50 remaining". At the bottom right are "Cancel", "Previous", and "Next" buttons.

Click on create auto scaling group

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 7: Review. The left sidebar is identical. The main content area displays the summary of the configuration: "Instance scale-in protection" (with an "Edit" button), "Step 5: Add notifications" (with an "Edit" button and a "Notifications" section showing "No notifications"), and "Step 6: Add tags" (with an "Edit" button and a "Tags (0)" section showing "No tags"). At the bottom right are "Cancel" and "Create Auto Scaling group" buttons.

See down in the below I have successfully created the Auto scaling group

The screenshot shows the AWS EC2 Auto Scaling groups page. At the top, there are three informational banners: one about predictive scaling policy, one about a new allocation strategy, and one confirming the creation of an Auto Scaling group named 'AGLC'. Below these, the main table lists the single Auto Scaling group 'AGLC' with its details: Desired capacity is 2, and the Auto Scaling group name is AGLC. The 'Details' tab is selected. At the bottom of the page, there are links for Feedback, Unified Settings, Privacy, Terms, and Cookie preferences.

You can see the 2 desired instances are created by auto scaling group.

The screenshot shows the AWS EC2 Instances page. The table lists five instances, with the first two, 'webserver1' and 'webserver2', being the ones created by the Auto Scaling group. Both are in the 'Running' state. The 'Monitoring' section below shows four charts for CPU utilization and status checks, all of which show 'No data available' at the current time.

If copy the public IP you can see the same web page which we have used for creating AMI.

Instance 1

The screenshot shows a browser window with the URL '44.211.151.159'. The page content is minimal, displaying the text 'webserver one'.



Instances 2



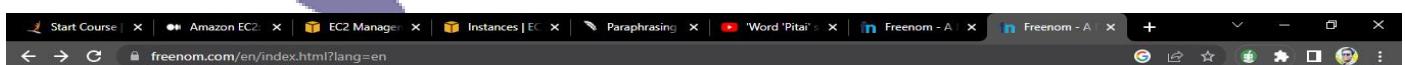
AWS Solutions Architect Training

Module-3: Route 53 Assignment - 3

You have been asked to:

1. Use the Route 53 Hosted Zone created in the Assignment
 2. Route the traffic to an EC2 instance with an Apache web server running in it using its IP address
1. Use the Route 53 Hosted Zone created in the Assignment
For this task we required the domain name, go to the freenom domain sit check the availability of the domain name.

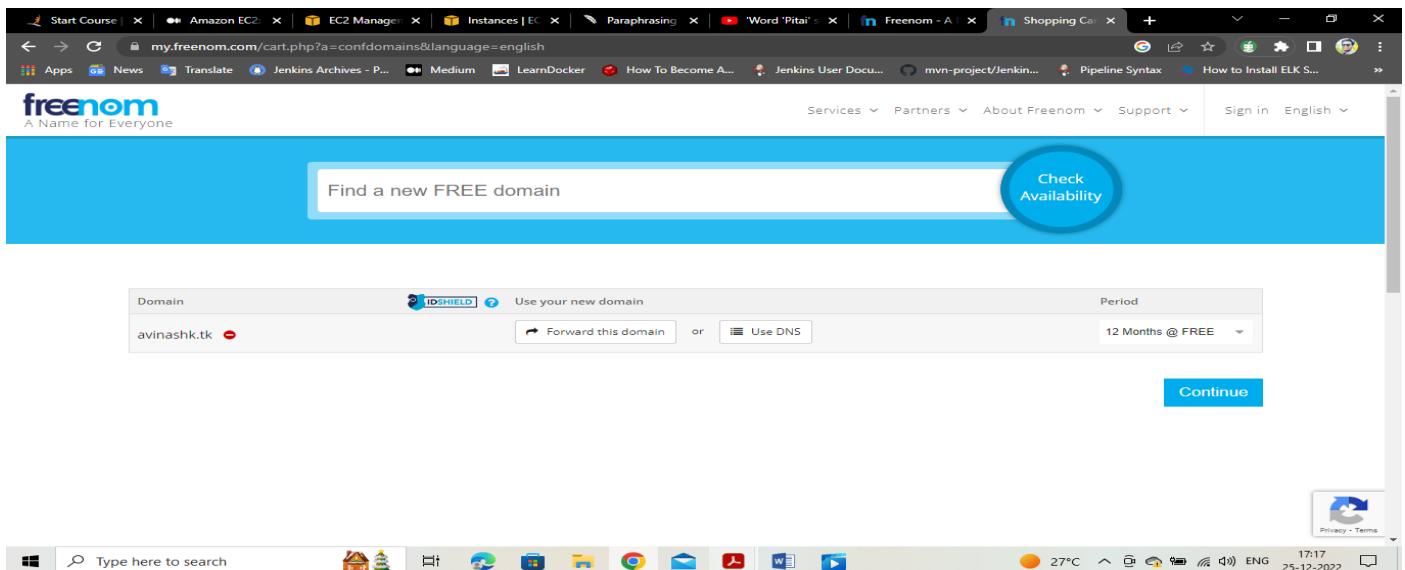
And click on checkout



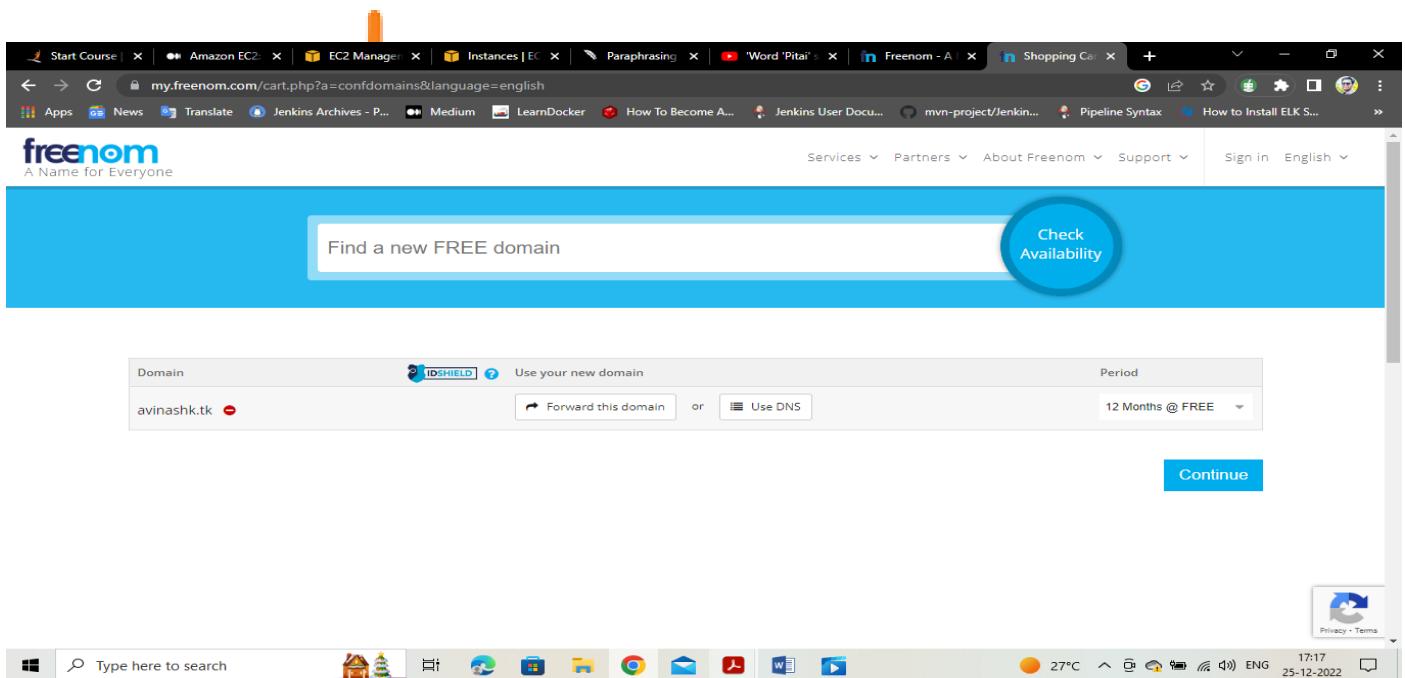
The screenshot shows the freenom domain checker results for "avinashk.tk". The domain is listed as available and free. The price is shown as "USD 0.00" and there is a green "Selected" button. Below the main result, there is a section titled "Get one of these domains. They are free!" which lists ".ml" and ".tk" domains.



Click on continue.



Click on continue.



And after sign in to the account >services >my domain

The screenshot shows a web browser window with the URL my.freenom.com/clientarea.php?action=domains. The page title is "My Domains" with the subtitle "View & manage all the domains you have registered with us from here...". There is a search bar labeled "Enter Domain to Find" and a blue "Filter" button. Below is a table with columns: Domain, Registration Date, Expiry date, Status, and Type. A single row is shown for "avinashk.tk", which is registered on 2022-12-25 and expires on 2023-12-25, marked as ACTIVE and Free. A "Manage Domain" button is available for this entry. At the bottom, it says "Results Per Page: 10" and "1 Records Found. Page 1 of 1". The browser's address bar shows the same URL. The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and other system icons.

Click on manage domain

The screenshot shows a web browser window with the URL my.freenom.com/clientarea.php?action=domaindetails&id=1146063394. The page title is "Managing avinashk.tk". It has tabs for Information, Upgrade, Management Tools, and Manage Freenom DNS. A blue banner at the top says "An unexpected error occurred". The "Information" tab is selected, displaying domain details: Domain: avinashk.tk (ACTIVE), Registration Date: 25/12/2022, and Expiry date: 25/12/2023. A "Back to Domains List" button is visible. The browser's address bar shows the same URL. The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and other system icons.

Now go to the amazon portal > search rout53 >hosted Zone

The screenshot shows a web browser window with the URL <https://us-east-1.console.aws.amazon.com/route53/home?region=us-east-1>. The page title is "Route 53". On the left, there is a sidebar with "Services (4)" and "Instances". Under "Services", "Route 53" is highlighted. The main content area shows "Search results for 'route 53'" with several items: "Route 53" (Scalable DNS and Domain Name Registration), "Route 53 Resolver" (Resolve DNS queries in your Amazon VPC and on-premises network), "Route 53 Application Recovery Controller" (Monitor application recovery readiness and manage failovers), and "Amazon Location Service" (Securely and easily add location data to applications). To the right, there is a "Launch Instances" section with a table for "Alarm status" and "Availability Zone", listing five entries for "No alarms" across "us-east-1b" and "us-east-1a". The browser's address bar shows the same URL. The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and other system icons.

Click on hosted zone.

The screenshot shows the AWS Route 53 Dashboard. On the left, there's a sidebar with navigation links for Hosted zones, Health checks, IP-based routing, Traffic flow, Domains, and Resolver. The main area has four sections: 'DNS management' (Create hosted zone), 'Traffic management' (Create policy), 'Availability monitoring' (Create health check), and 'Domain registration' (Register domain). At the bottom, there's a search bar and a status bar indicating the URL is https://us-east-1.console.aws.amazon.com/route53/v2/hostedzones/Createhostedzone? and the date/time as 25-12-2022 17:36.

Copy the domain name which we have created and past over the name of the domain

The screenshot shows the 'Create hosted zone' configuration page. It includes fields for 'Domain name' (avinashk.tk), 'Description - optional' (The hosted zone is used for...), and 'Type' (Public hosted zone selected). There are also sections for 'Tags' and 'Feedback'. The status bar at the bottom indicates the URL is https://us-east-1.console.aws.amazon.com/route53/v2/hostedzones/Createhostedzone? and the date/time as 25-12-2022 17:36.

Click on create hosted zone

The screenshot shows the final step of creating a hosted zone, where the user is prompted to 'Create hosted zone'. The status bar at the bottom indicates the URL is https://us-east-1.console.aws.amazon.com/route53/v2/hostedzones/Createhostedzone? and the date/time as 25-12-2022 17:36.

We can see down we have successfully created the hosted zone.

The screenshot shows the AWS Route 53 console. On the left, a sidebar lists various services: Dashboard, Hosted zones (which is selected), Health checks, IP-based routing, Traffic flow, Domains, Resolver, and DNS Firewall. The main pane displays a success message: "avinashk.tk was successfully created. Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain." Below this, it shows the "Hosted zone details" for "Public avinashk.tk". Under the "Records (2)" tab, there are two entries:

Record name	Type	Routing...	Differ...	Value/Route traffic to
avinashk.tk	NS	Simple	-	ns-766.awsdns-31.net; ns-348.awsdns-43.com; ns-1751.awsdns-26.co.uk; ns-1192.awsdns-21.org
avinashk.tk	SOA	Simple	-	ns-766.awsdns-31.net. awsdns-hostmaster.amazon.c...

Now we have to copy the routing traffic rout links to the name server in freenom domain portal.

The screenshot shows the Freenom domain management portal. The URL is "my.freenom.com/clientarea.php?action=domaindetails&id=1146063394". The page title is "Managing avinashk.tk". On the left, there's a sidebar with "Information" and "Upgrade" tabs, and a "Management Tools" dropdown menu with options like "Nameservers", "Register glue records", and "URL Forwarding". The main content area shows the domain details for "avinashk.tk" (status: ACTIVE, creation date: 02-12-2022, expiration date: 29-12-2023). A message "An unexpected error occurred" is displayed. At the bottom, there's a "Back to Domains List" button.

Fill the name server value and click on change name server.

The screenshot shows the "Nameservers" configuration page for the domain "avinashk.tk". The URL is "my.freenom.com/clientarea.php?action=domaindetails&id=1146063394". The page title is "Nameservers". It asks if you want to "Use default nameservers (Freenom Nameservers)" or "Use custom nameservers (enter below)". The "Use custom nameservers" option is selected. The "Nameserver 1" field contains "ns-766.awsdns-31.net". There are five empty fields for "Nameserver 2" through "Nameserver 5". At the bottom, there's a "Change Nameservers" button.

We have successfully changed the nameservers.

The screenshot shows the freenom client area interface. At the top, there are tabs for Start Count, Amazon E, avinashk.tk, Client Area, Instances, Paragraphs, Word Pits, WhatsApp, AMCAT Test, and others. Below the tabs, the freenom logo is visible with the tagline "A Name for Everyone". The main title is "Managing avinashk.tk". Below the title, there are tabs for Information, Upgrade, Management Tools, and Manage Freenom DNS. A blue banner at the bottom says "Changes Saved Successfully!". On the left, there's a sidebar titled "Information" with a sub-section for "Domain: avinashk.tk ACTIVE". It shows the registration date as 25/12/2022 and the expiry date. There's a link to "Back to Domains List".

Now we have to integrate the IP address to route to instance.

Click create record

The screenshot shows the AWS Route 53 console. The left sidebar includes "Route 53", "Hosted zones" (selected), "Health checks", "IP-based routing", "Traffic flow", "Domains", and "Resolver". The main content area shows a green success message: "avinashk.tk was successfully created. Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain." Below this, there are tabs for "Records (2)", "DNSSEC signing", and "Hosted zone tags (0)". The "Records (2)" tab displays a table with two entries: "avinashk.tk" (NS, Simple) and "avinashk.tk" (SOA, Simple). The table has columns for Record name, Type, Routing policy, Differ..., and Value/Route traffic to. The "Value/Route traffic to" column lists IP addresses and domain names. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

Select the simple routing > next

The screenshot shows the "Choose routing policy" step of a wizard. The left sidebar shows "Step 1 Choose routing policy" and "Step 2 Configure records". The main content area is titled "Choose routing policy" with a sub-link "Info". It says "The routing policy determines how Amazon Route 53 responds to queries." Below this, there are five options: "Simple routing" (selected), "Weighted", "Geolocation", "Latency", "Failover", "Weighted" (selected), "Geolocation", "Latency", "Failover", and "Multivalue answer". Each option has a brief description and a small icon. A "Switch to quick create" link is located in the top right of the content area. The bottom of the screen shows the Windows taskbar and system tray.

Click on define simple record

The screenshot shows the AWS Route 53 'Configure records' page. The left sidebar shows 'Step 1 Choose routing policy' and 'Step 2 Configure records'. The main area is titled 'Simple routing records to add to avinashk.tk' with a sub-section 'Record name'. A table has one row defined: 'Record name' is 'avinashk.tk', 'Type' is 'A', 'Value/Route traffic to' is '3.86.25.167', and 'TTL (seconds)' is '300'. A large orange 'Create records' button is at the bottom right.

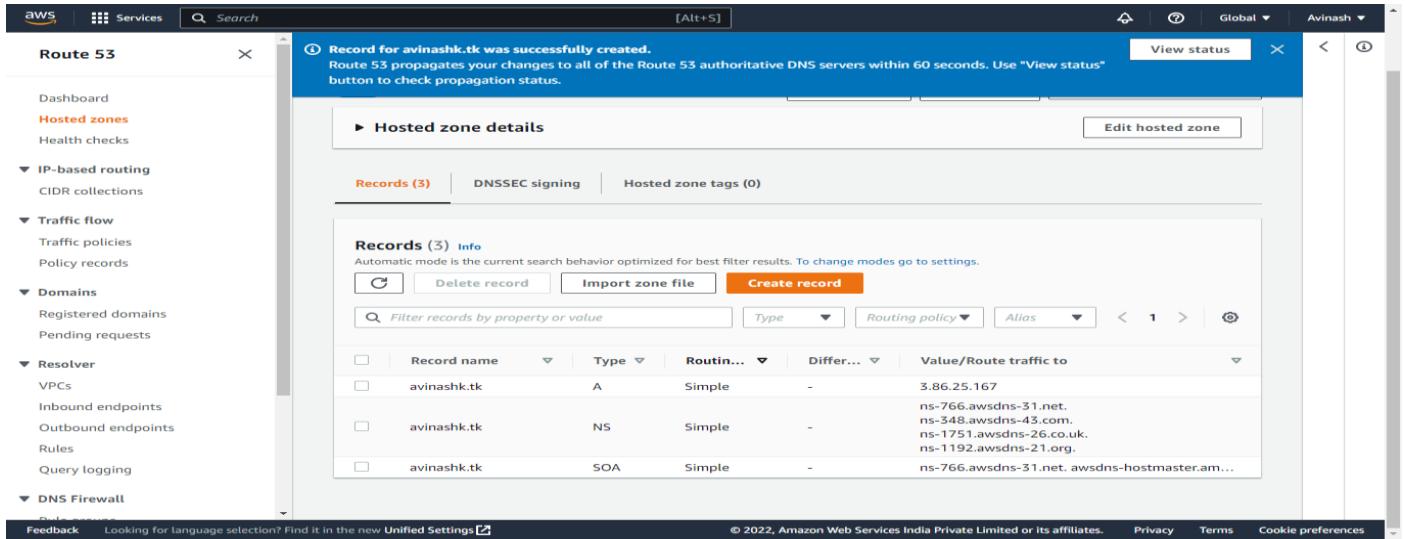
Given the required information > define simple record.

The screenshot shows the 'Define simple record' dialog box. It has fields for 'Record name' (subdomain: 'avinashk.tk'), 'Record type' (selected: 'A - Routes traffic to an IPv4 address and some AWS resources'), 'Value/Route traffic to' (selected: 'IP address or another value, depending on the record type', with value '3.86.25.167'), and 'TTL (seconds)' (value '300'). A large orange 'Create records' button is at the bottom right.

We have successfully created the define simple record

The screenshot shows the 'Configure records' page again. The table now has two rows: the first row is 'avinashk.tk' (Type: A, Value: 3.86.25.167, TTL: 300), and the second row is 'Existing records' (empty). The orange 'Create records' button is still present at the bottom right.

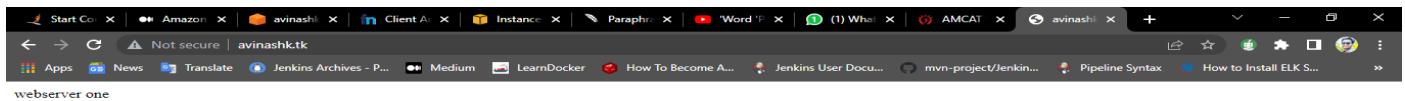
Click on create record.



The screenshot shows the AWS Route 53 service interface. On the left, a sidebar lists various services like Dashboard, Hosted zones, IP-based routing, Traffic flow, Domains, Resolver, and DNS Firewall. The 'Hosted zones' section is selected. In the main pane, a success message states 'Record for avinashk.tk was successfully created.' Below this, the 'Hosted zone details' section is shown with tabs for 'Records (3)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (3)' tab is active. It displays three records:

Record name	Type	Routing policy	Differentiator	Value/Route traffic to
avinashk.tk	A	Simple	-	3.86.25.167
avinashk.tk	NS	Simple	-	ns-766.awsdns-31.net. ns-348.awsdns-43.com. ns-1751.awsdns-26.co.uk. ns-1192.awsdns-21.org.
avinashk.tk	SOA	Simple	-	ns-766.awsdns-31.net. awsdns-hostmaster.am...

Copy domain name and check whether the domain name which we have created is redirecting to the webserver or not.



The screenshot shows a browser window with several tabs open. One tab is active and shows the URL 'avinashk.tk'. The browser's address bar also displays 'avinashk.tk'. The taskbar at the bottom of the screen shows icons for various applications, including File Explorer, Google Chrome, Mail, Word, and others.

We can see the above that we are able to see the webpage with domain name.