CCT College Dublin

Assessment Cover Page

Module Title:	Cloud Services
Assessment Title:	Implementing a proof-of-concept cloud architecture
Lecturer Name:	Michael Weiss
Student Full Name:	Muhammad Sajjad Haider
Student Number:	2021384
Assessment Due Date:	: Sunday 17th November – 23:59
Date of Submission:	: Sunday 16th November
GitHub Link:	https://github.com/HaiderAk007/AWS_Architect-Project1

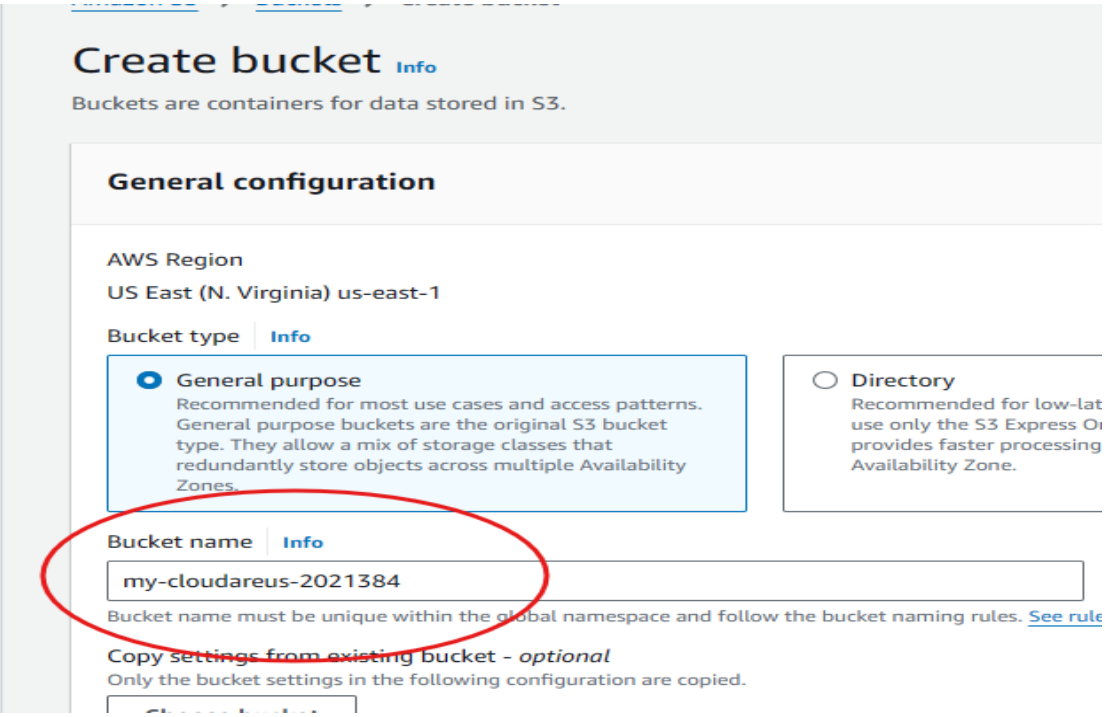
Table of Contents

Q:1(a)Clouds-Are-Us: Setup a static website 3
Q:1(b):Research Task: Comparison of S3 & EC2 with EBS 8
Type of Storage: 9
Data Structure: 9
Patterns of Access: 9
Availability and Durability: 9
Costs: 10
Similarities: 10
Security: 10
Applications Use Cases Amazon EBS 10
Q2(a): Application Load Balancer (ALB) with 5 Linux and Custom VPC 11
Creation of Custom VPC: 11
Creation of 5 Linux-servers: 12

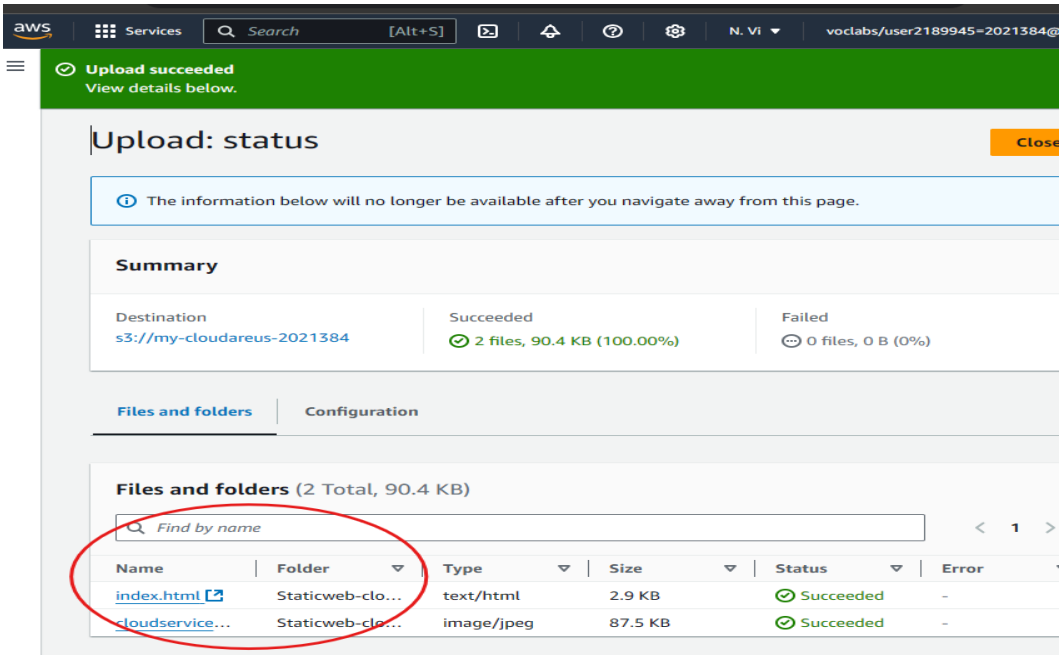
Creation Of Target Group 17
Creation of Application Load Balancer (ALB) 20
Q2(b): Challenge Task: Incorporate the DigiTech Website with ALB 24
Q3(a): Challenge Task: Proof Of concept, Auto Scaling Group with ALB 28
Creation AMI image: 28
Target group for ASG: 29
Creation of Launch Template: 31
Creation of ASG with attached ALB: 33
Modify Launch Template and Edit ASG: 38
Q3(b): Research Task: Principles of ASG with ALB 40
The Principle of Operation of DigiTech ALB 40
Principles for combining ASG with ALB 40
References: 41

Q:1(a)Clouds-Are-Us: Setup a static website

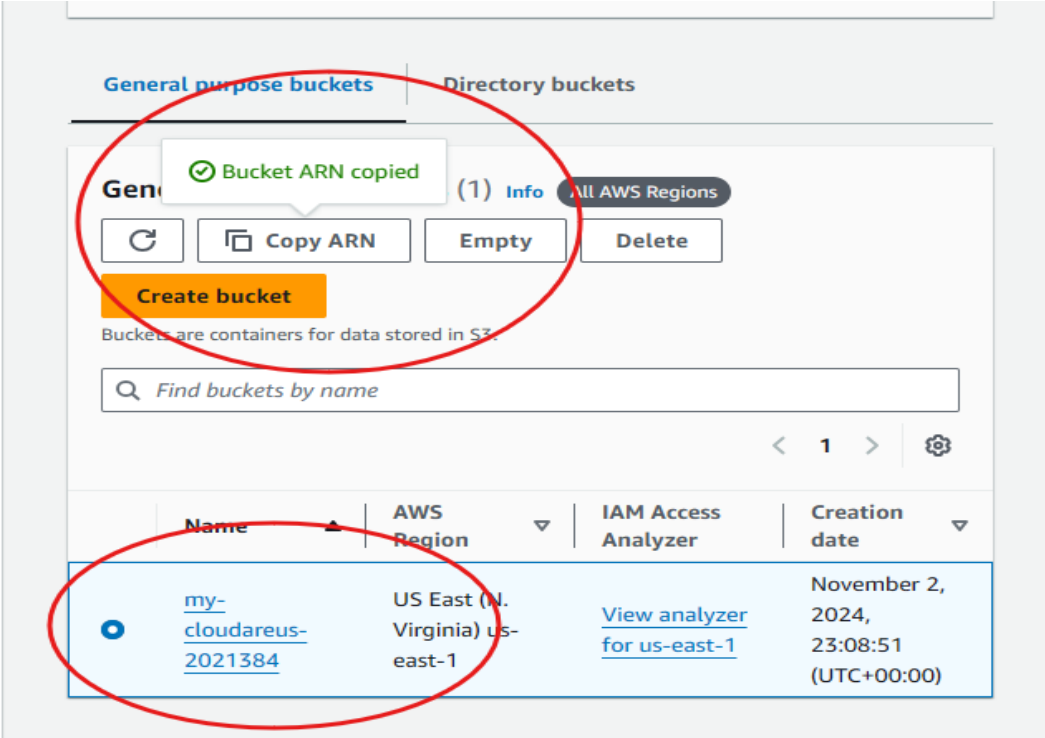
- In Services, navigate S3, select Buckets and hit Create bucket.
- I gave bucket name as my-cloudareus-2021384.



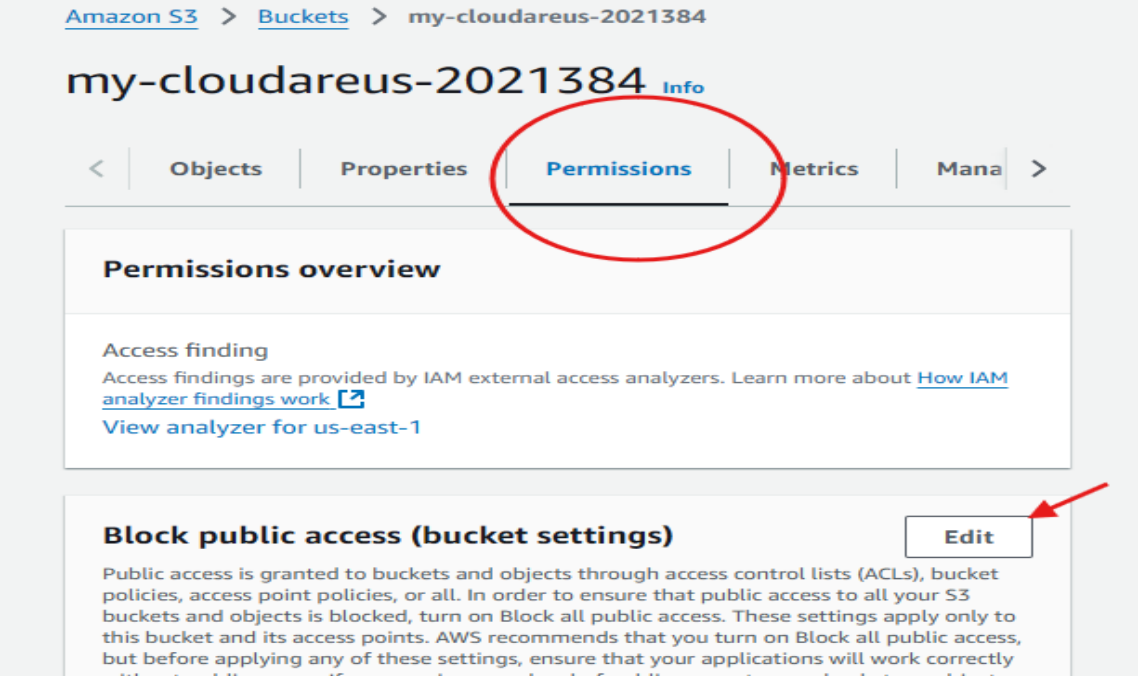
- Inside the bucket, I uploaded my custom made index.html page of cloud-are-us website and image folder.



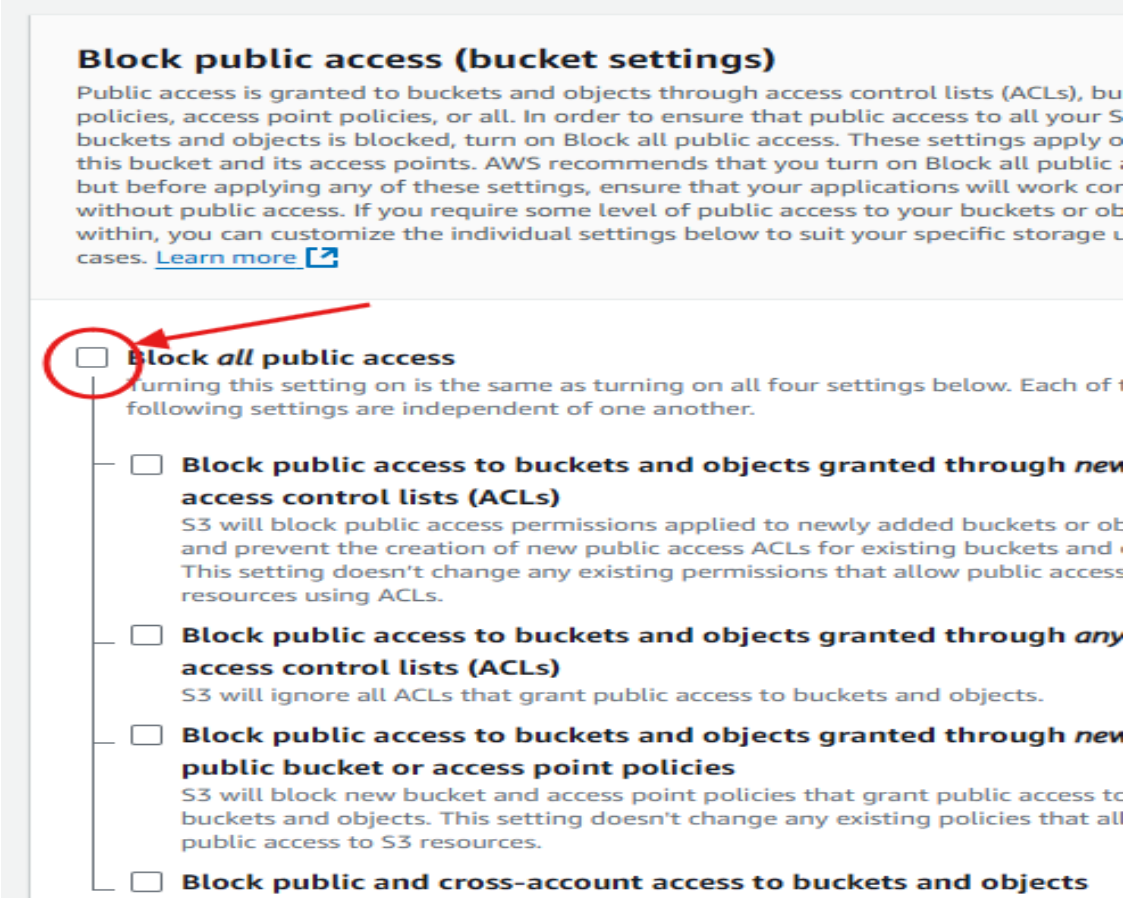
- Copy the ARN of my-cloudareus-2021384 bucket for modification in bucket policy script.



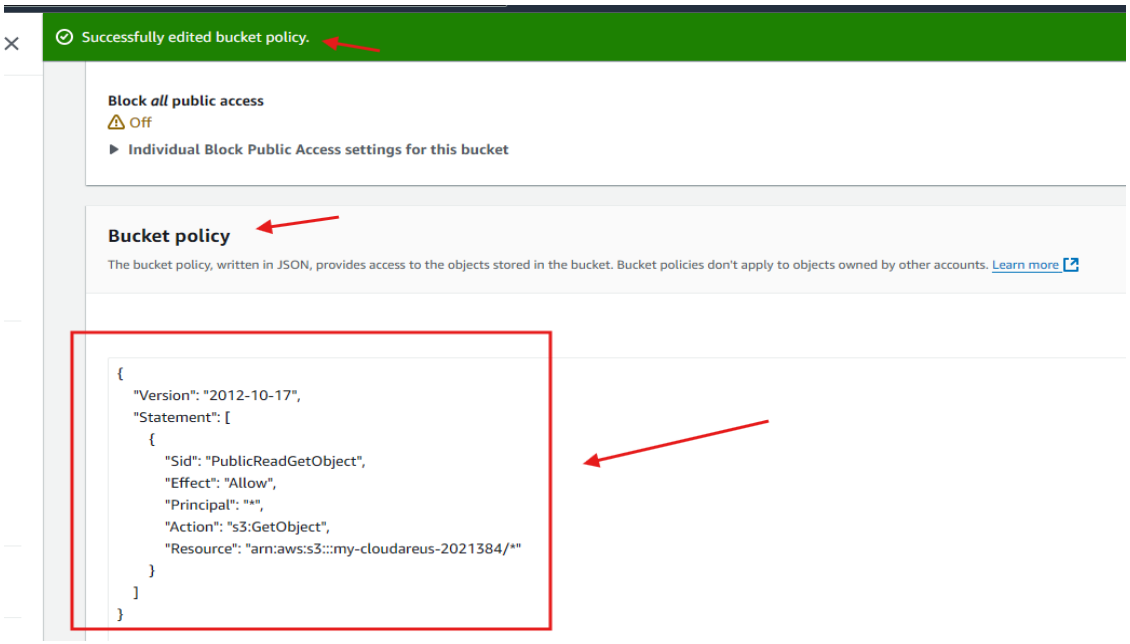
- Go to Permissions, inside the bucket.
- Edit Bucket public access setting.



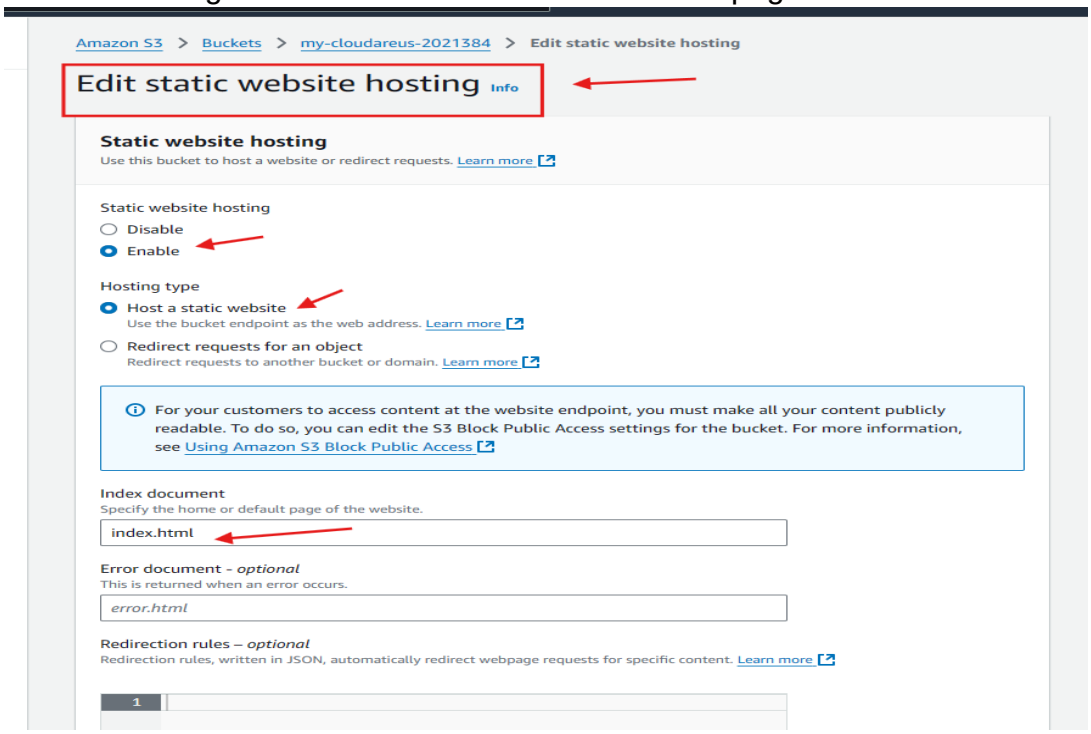
- Untick all block access section publicly.



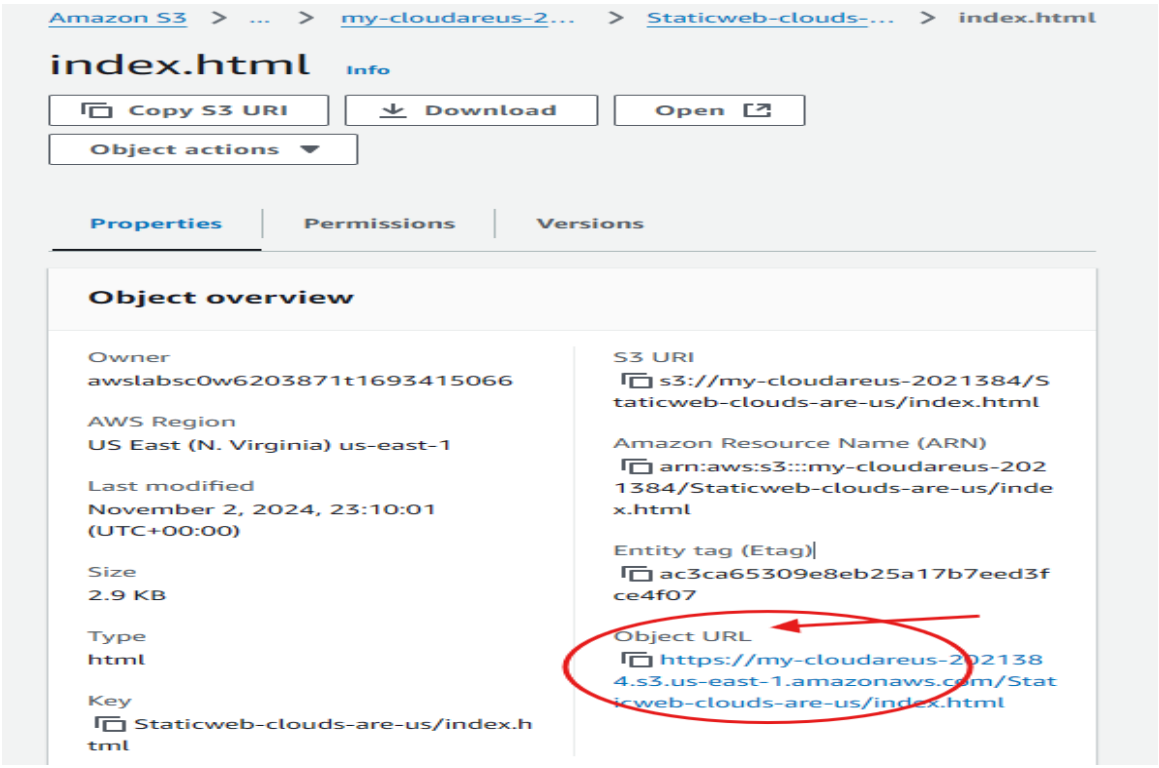
- After that, Edited Bucket Policy by adding the script inside the policy with bucket ARN.



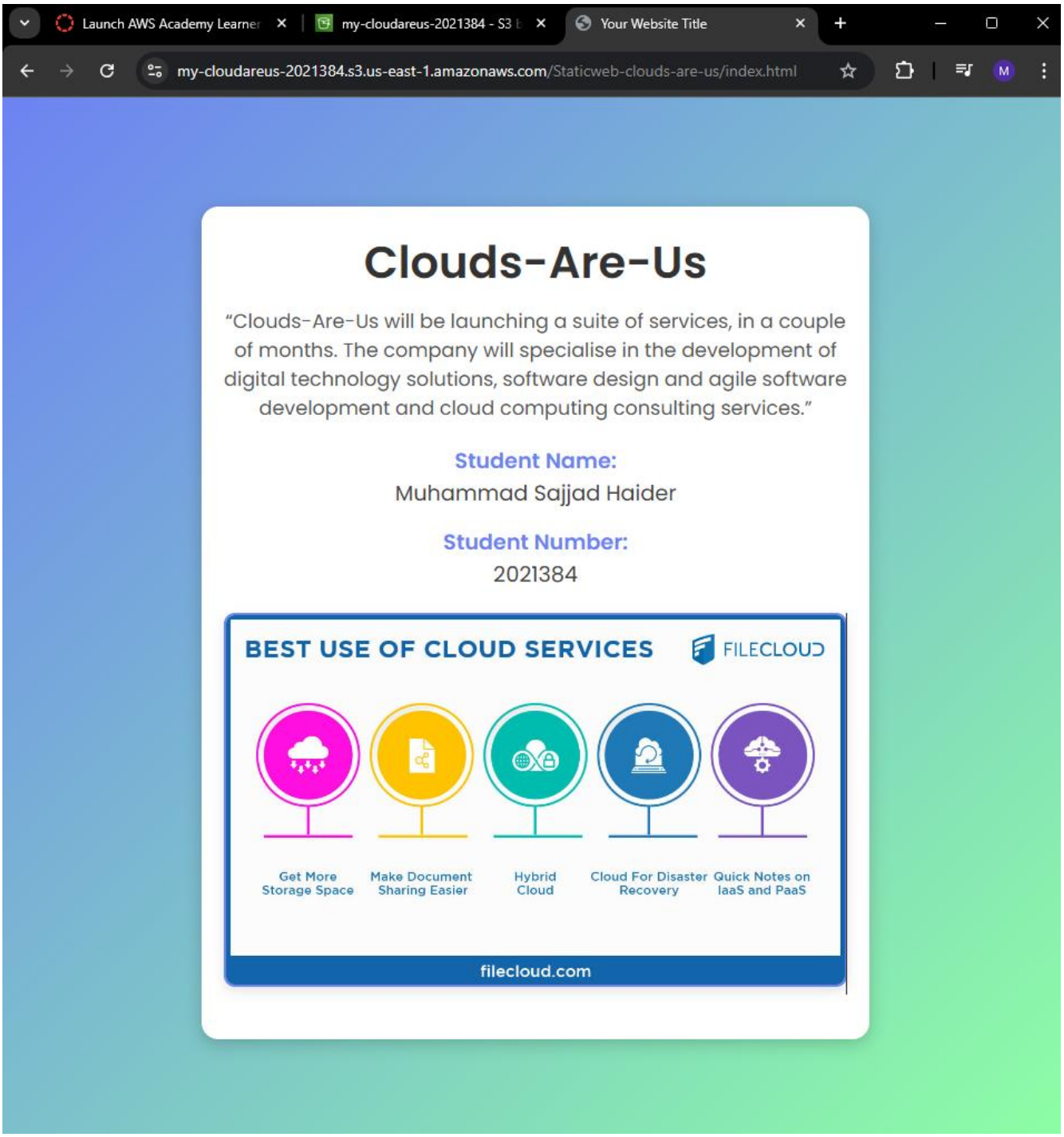
- Now, Go to properties of bucket.
- Edited the static website hosting and enable it.
- Set on hosting a static website and set the default page as index.html.



- Copy the URL of index.html file inside the bucket and search it on browser.



- It can be seen that Cloud-Are-Us website is successfully working with my name and student number in shot below:



Q:1(b):Research Task: Comparison of S3 & EC2 with EBS

The type of data that can be stored and the utility of the two Amazon services, S3 and EC2 coupled with EBS differ as well. These two kinds of storage options are compared

in detail in this representation: how they are alike, how they differ, and where they are suitable for use.

Type of Storage:

S3: Object storage designed to be available, scalable and most importantly built to be durable. It stores data as objects using buckets.

For objects that want to hook up directly to an EC2 instance and works like a regular hard drive, there is EBS stands for Elastic Block Storage with storage volumes at block level available for provision.

Data Structure:

S3: In this model, the data is stored in objects which contain a reference number, some type of information about the piece of data, and the actual data. Depending on the size it is possible that everything up to and including 5TB.

EBS: Blocks of data are arranged. Volumes are attached to EC2 instances by direct connection, and they can be scaled up to a 64TB throughput.

Patterns of Access:

For the large amounts of structured, semi-structured, and unstructured data in:

S3: It is suitable for occasional or unpredictable use, distribution of print media, such as newsletters, and any form of content that is not frequently updated.

EBS: Meant for workloads such as databases or file systems that regularly write and read data out on their storage device. Specifically, it is designed for low latency access.

Availability and Durability:

S3: offers the availability and the durability of 99.999999999% (11 nines) in many geographical regions. Database replication occurs and it just happens in the background.

EBS: It provides 99.999% durability; however, it comes with the provision of availability zone only. For redundancy, they can take the so-called snapshot.

Costs:

S3: Bill is based on traffic associated with operations such as requests and amount of data transferred, stored, and used. So, it can be economically reasonable for big data sets.

EBS: Account cost is derived from IOPS (input/output operations per second) and volume size. However, the level of high performance may only be achieved at a higher cost.

Similarities:

S3 and EBS both have the provision of scalability. They achieved scalability in quite opposite manner. S3 is bucket can be manually resized while EBS automatically scales itself in response to usage.

Security:

Both have some of the analyzed features such as encryption data in transport and in storage, access through IAM policies, and compliance with several standards.

Applications Use Cases for Amazon S3

Data Lakes:

Storing massive amounts of raw data for AI processing and use for machine learning and big data computing.

Static Website Hosting:

Deliver, HTML based, CSS based, image-based static content.

Backup and Restore:

Coping with data, databases and on-premise applications back up.

Media Storage and Streaming:

Availing apps with the use of audio files, videos and other media.

Applications Use Cases Amazon EBS

Database Storage:

where you are executing low latency relational databases like PostgreSQL and MySQL.

File Systems:

Extending hosting for file systems which have to be stable and performant synchronously.

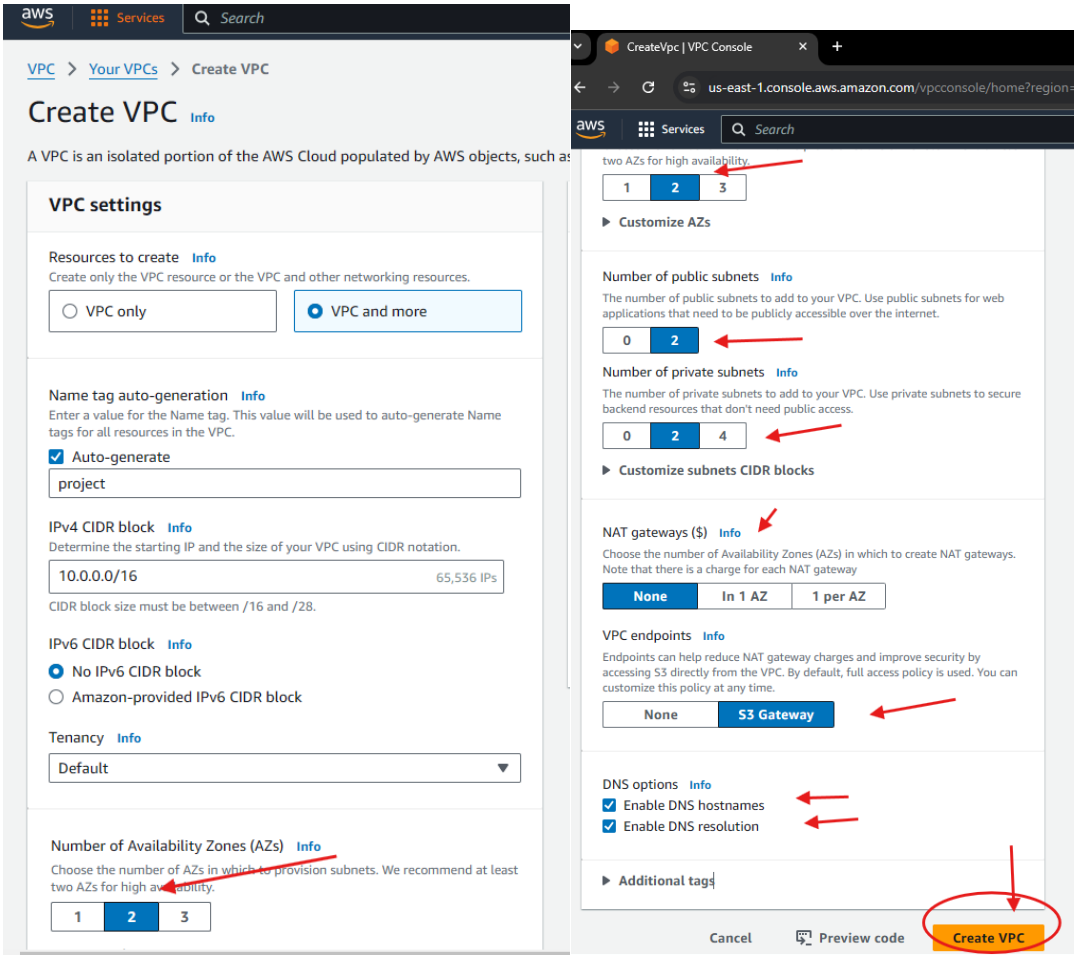
Development and Testing:

Charging programmers with the task of establishing scenarios in which they should launch and terminate instances as quickly as possible.

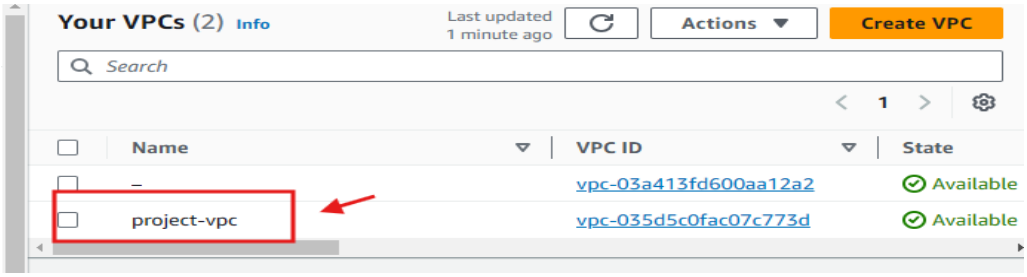
Q2(a): Application Load Balancer (ALB) with 5 Linux and Custom VPC

Creation of Custom VPC:

- Navigate VPC, from services and click create VPC.
- Selected VPC and more and gave name project.
- Select 2 Availability Zones (AZs) with two public subnets and two private subnets and stick with S3 Gateway.
- Stick with Enable hostnames and resolution and create VPC.

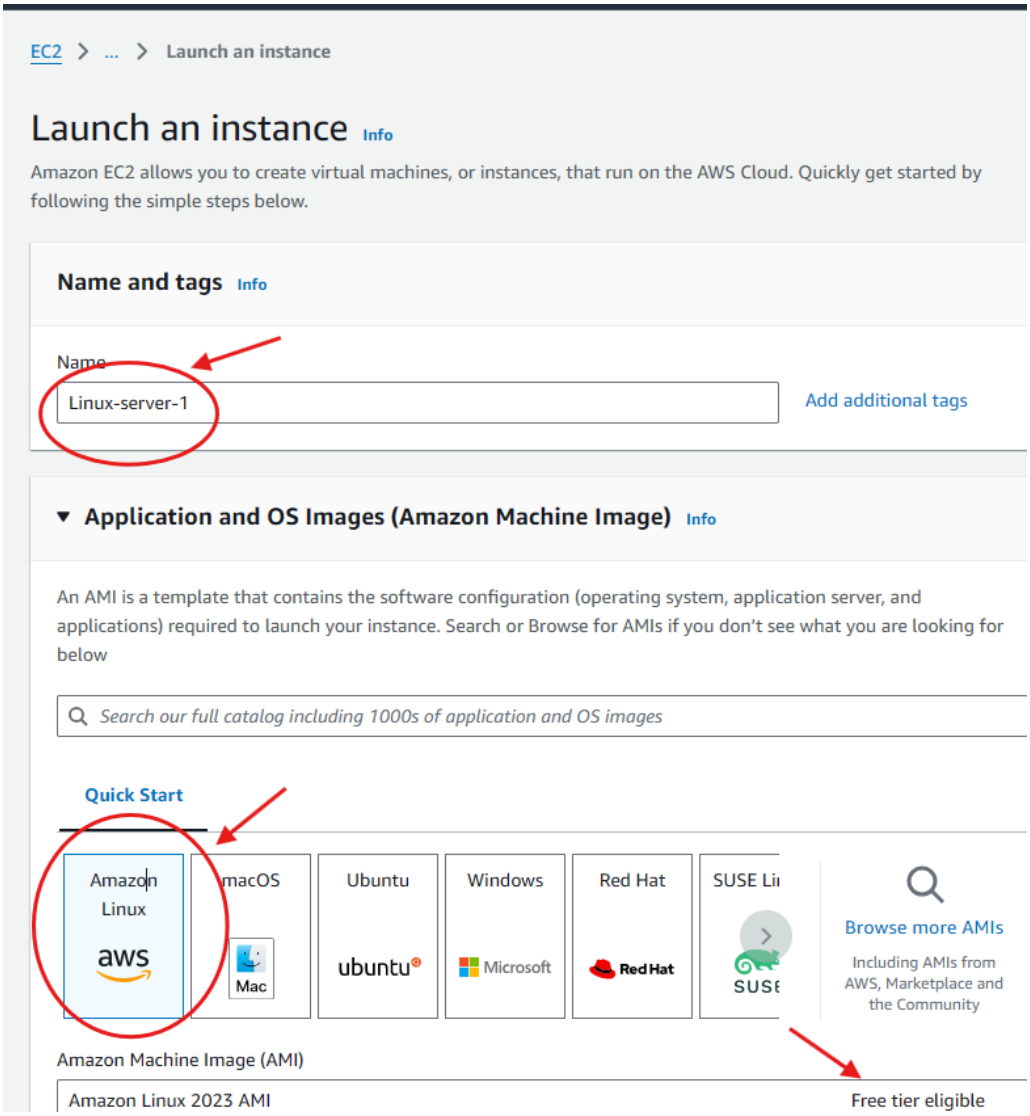


- The new project-vpc has created successfully.

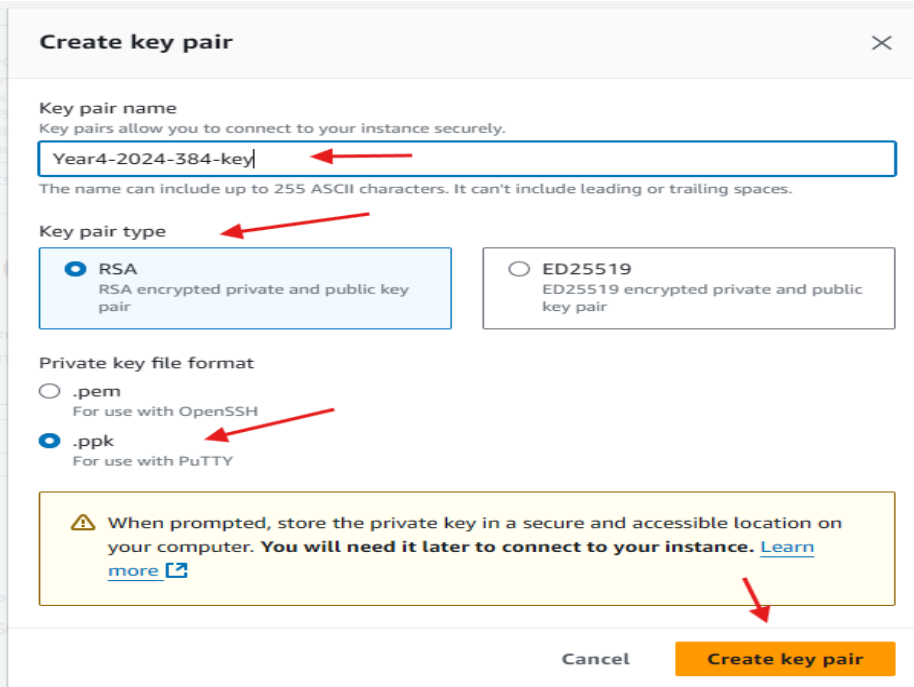


Creation of 5 Linux-servers:

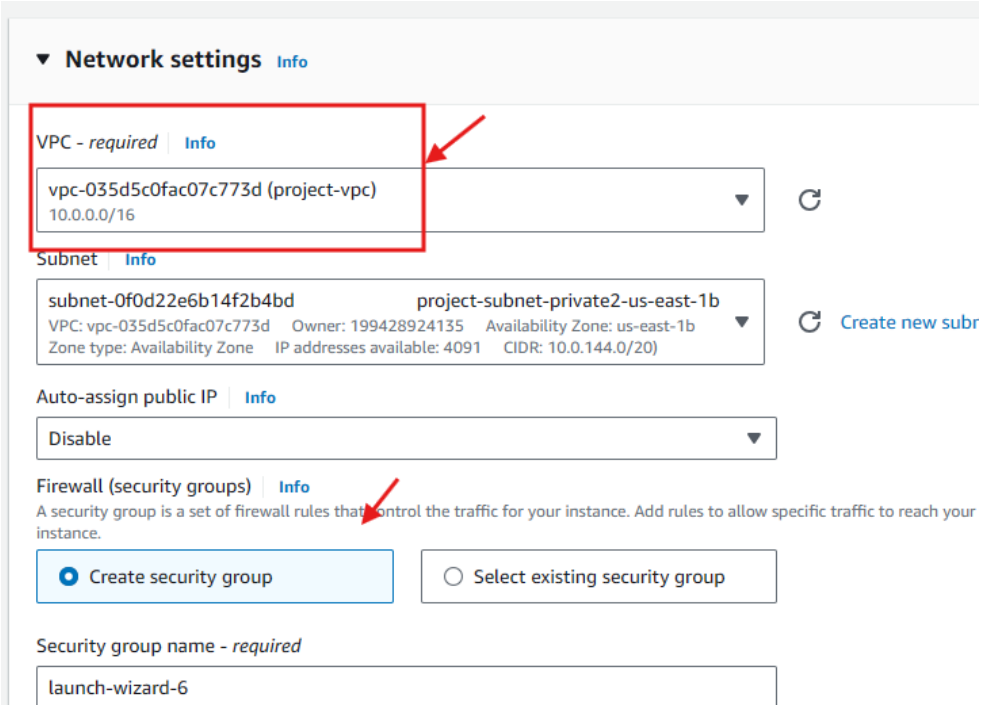
- Navigate EC2, from services and click Launch instance.
- Gave names of all 5 instances as Linux-server-1, Linux-server-2, Linux-server-3, Linux-server-4 and Linux-server-5.
- Selected Amazon Linux as Amazon Machine Image (AMI).



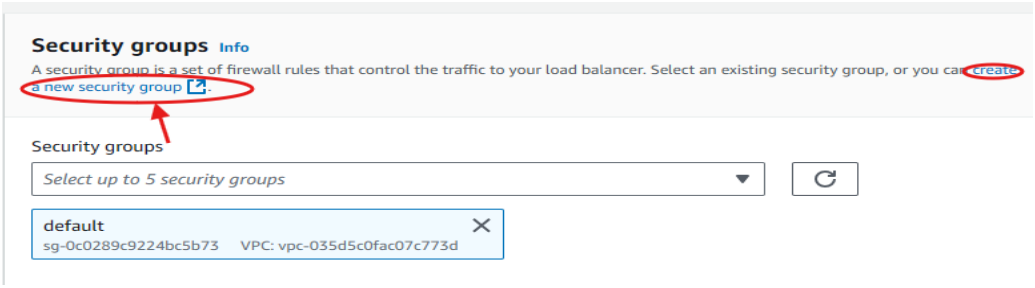
- I create new pair with name Year4-2024-384-key with the type RSA and file format .ppk.



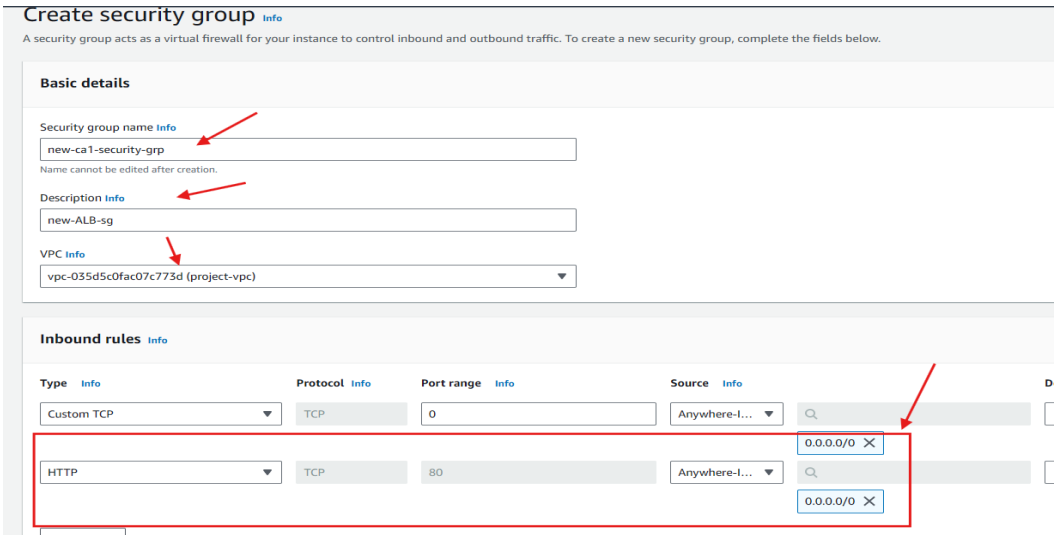
- Selected Custom created project-vpc with the Public subnet.



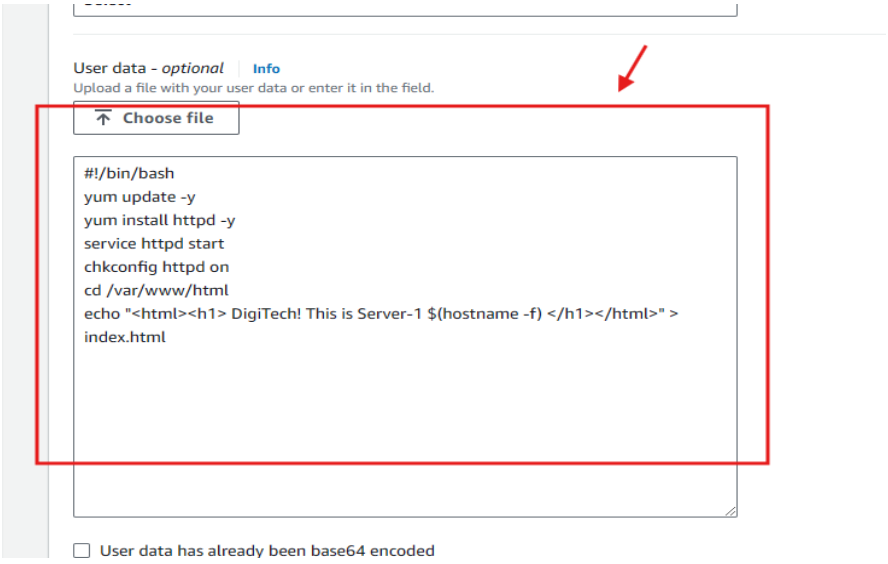
- Create new security group for all Linux servers.



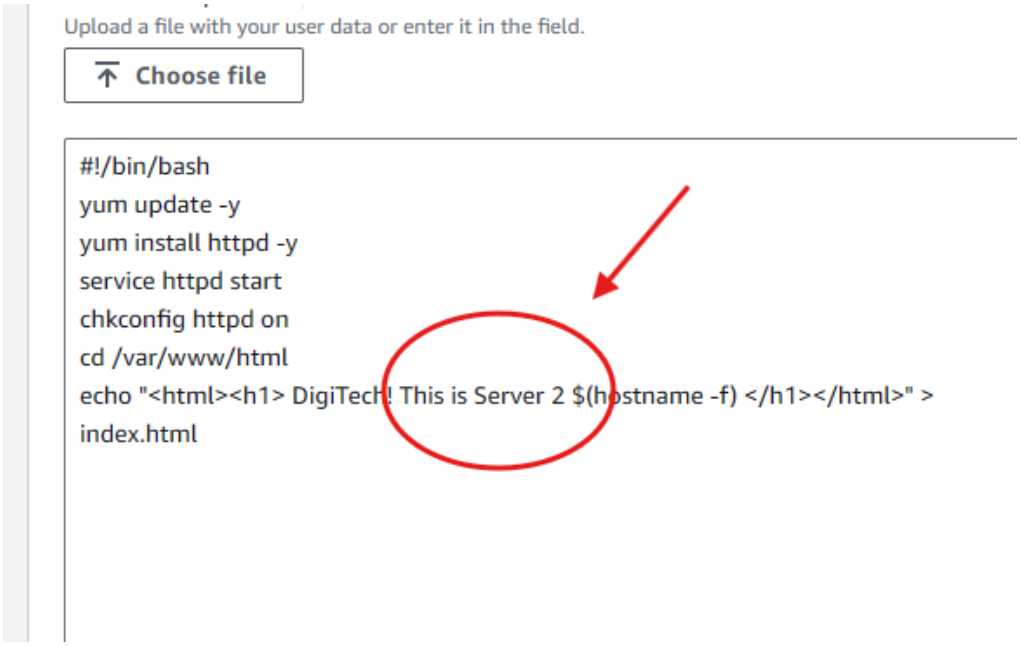
- I gave name with new-ca1-secuirty-grp and description as new-ALB-sg.
- Selected with custom created project-vpc.
- Added inbound rule HTTP.



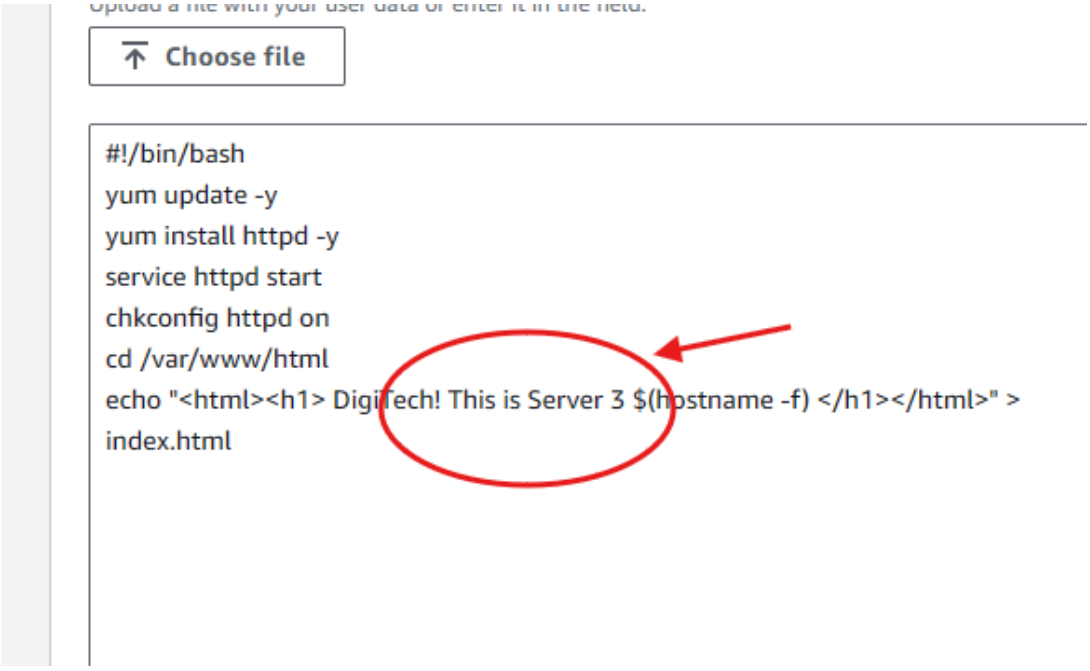
- Gave Linux-startup script in advanced tab for 1st Linux-server and click Launch instance.



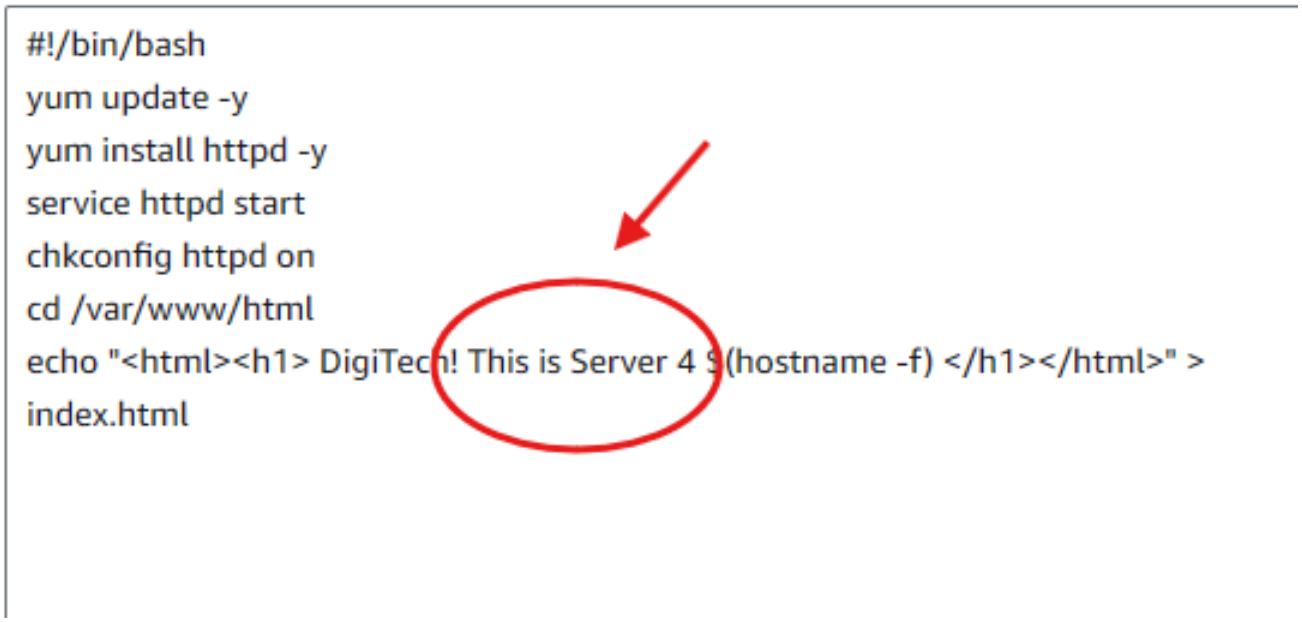
- I have created all 5 Linux-servers in same way.
- Gave Linux-startup script in advanced tab for 2nd Linux-server and click Launch instance.



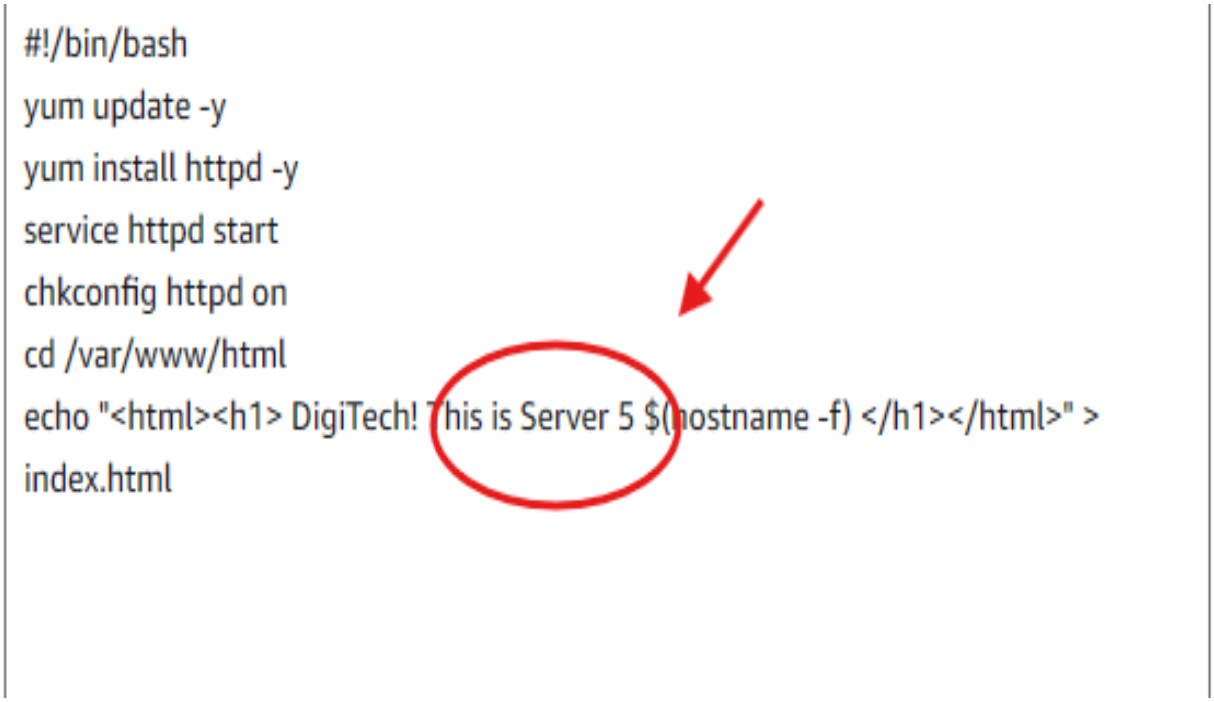
- Gave Linux-startup script in advanced tab for 3rd Linux-server and click Launch instance.



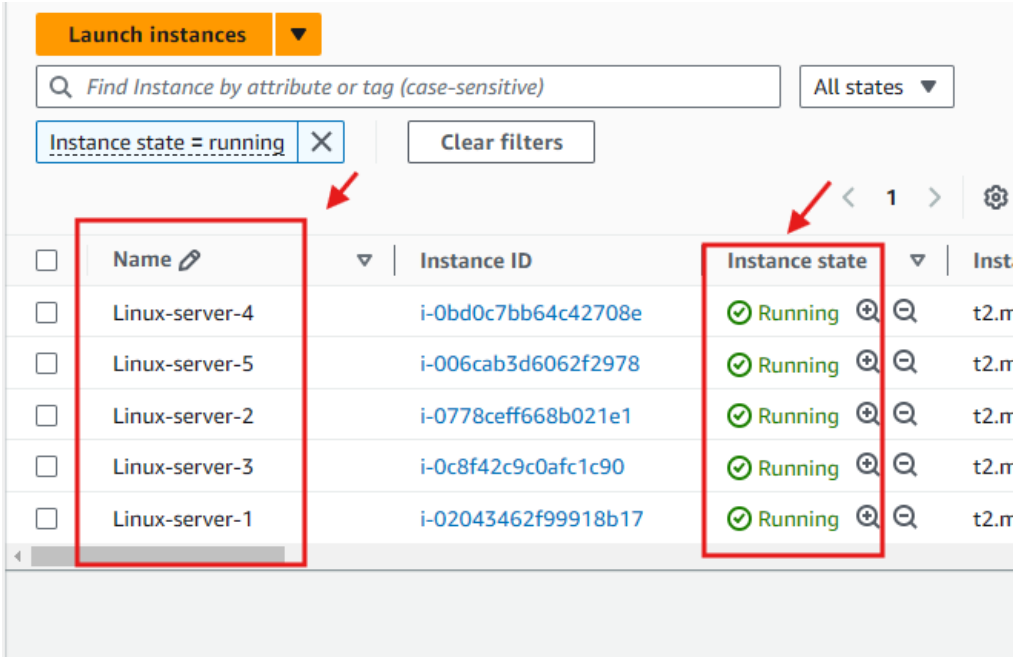
- Gave Linux-startup script in advanced tab for 4th Linux-server and click Launch instance.



- Gave Linux-startup script in advanced tab for 5th Linux-server and click Launch instance.

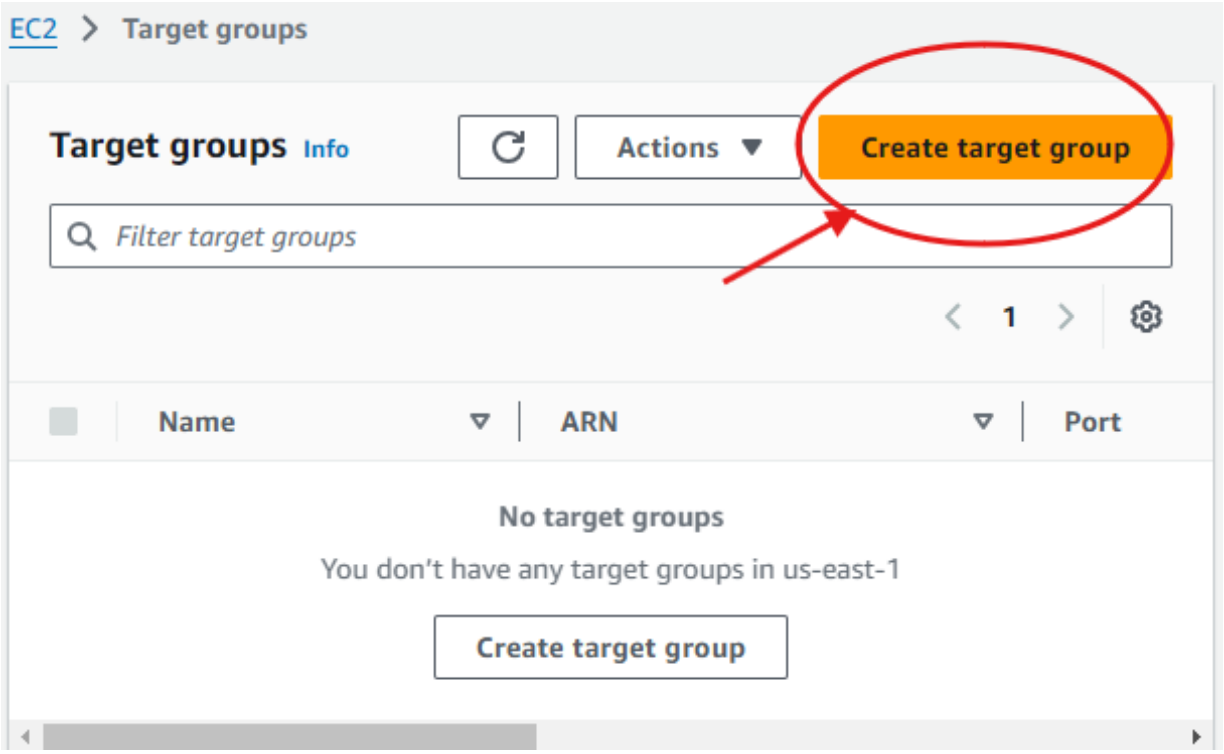


- All 5 Linux-servers has created successfully and can be seen in shot below:

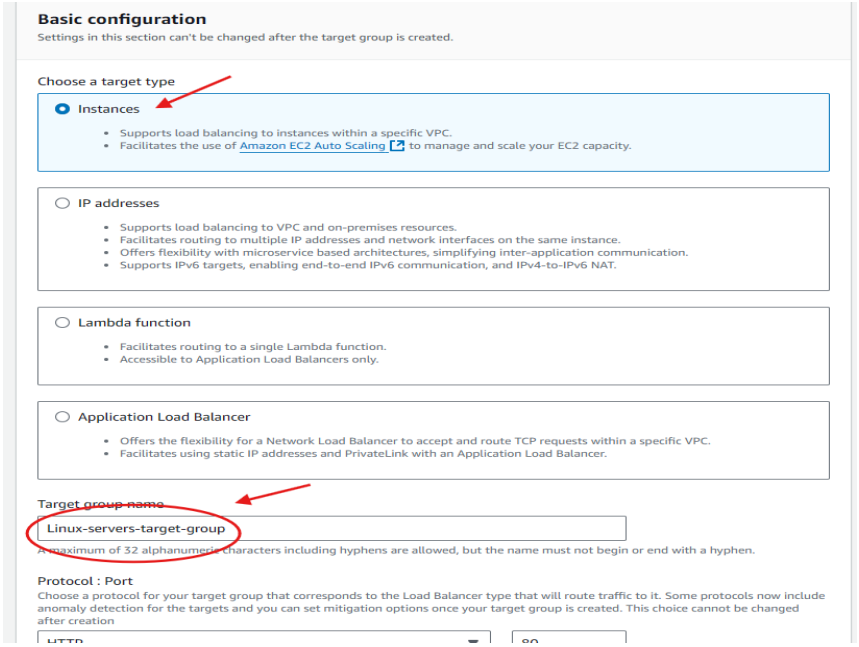


Creation Of Target Group

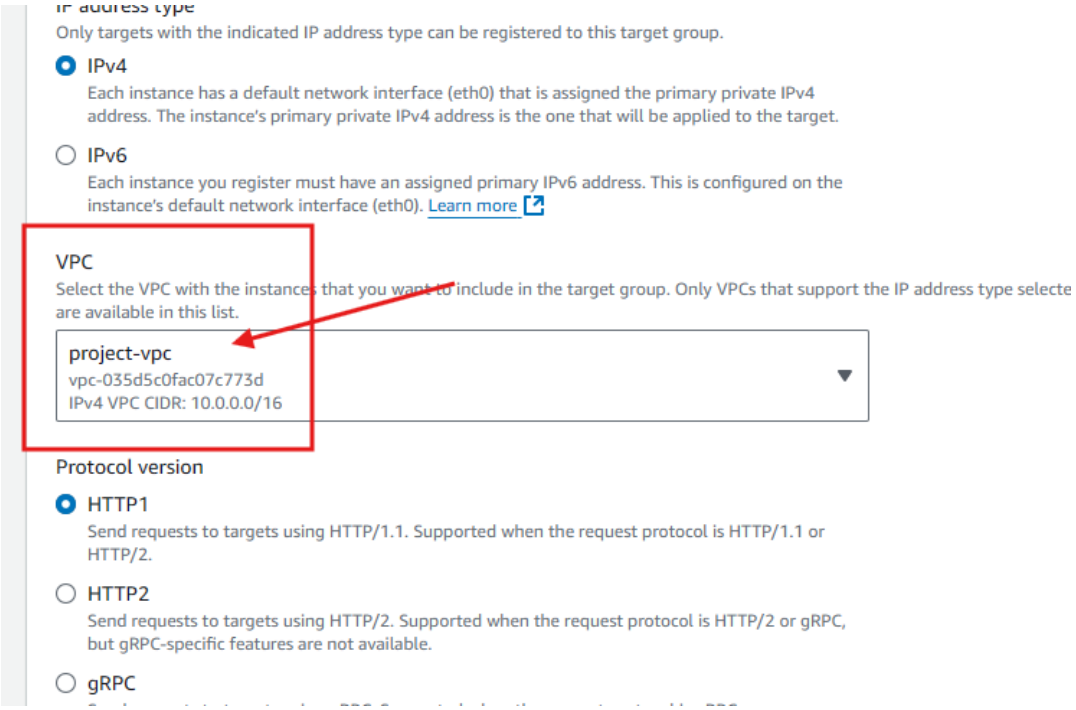
- Click create target group navigate from left menu in EC2.



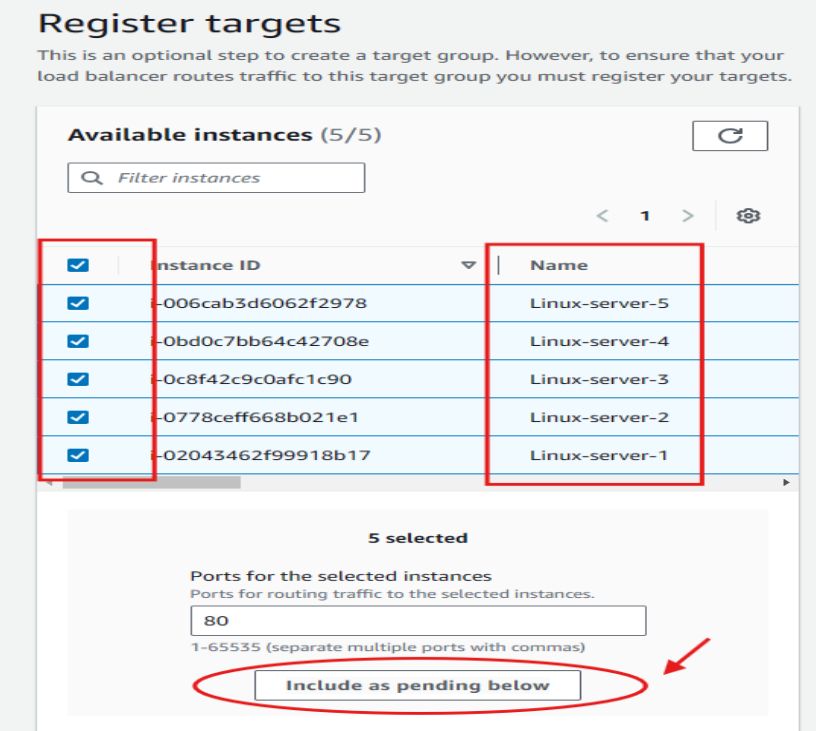
- Stick with Instances as target type and gave target group name as Linux-servers-target-group.



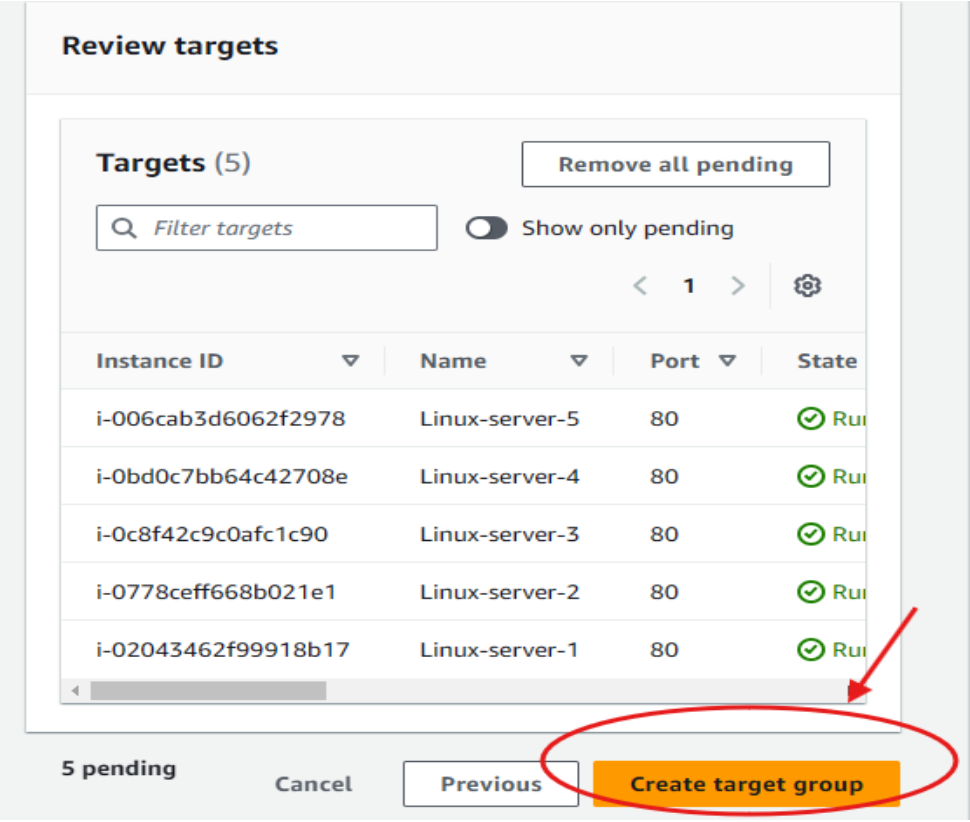
- Selected custom created project-vpc and Click Next.



- Add all 5 Linux-servers in Register targets and click option include pending.

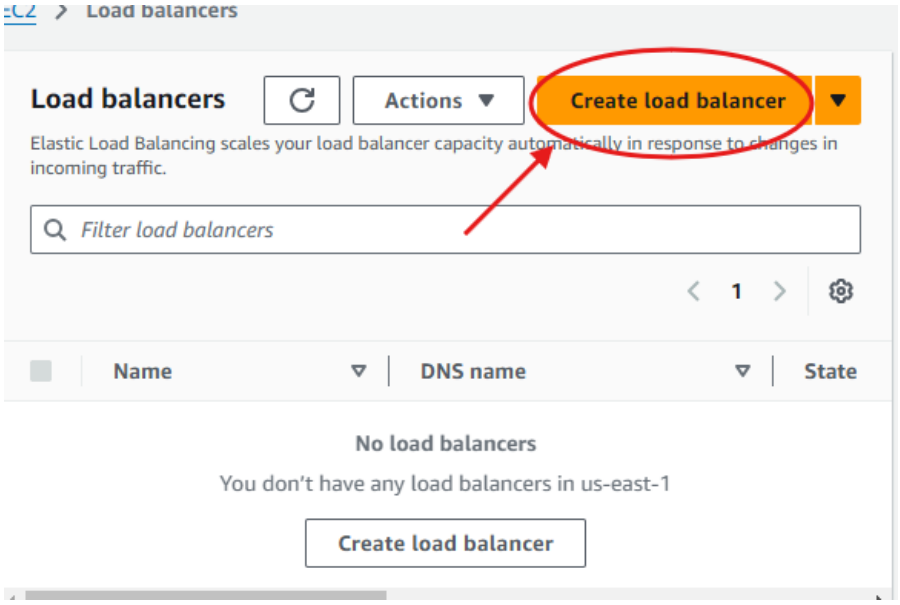


- All Linux-servers has added and click Create target group.

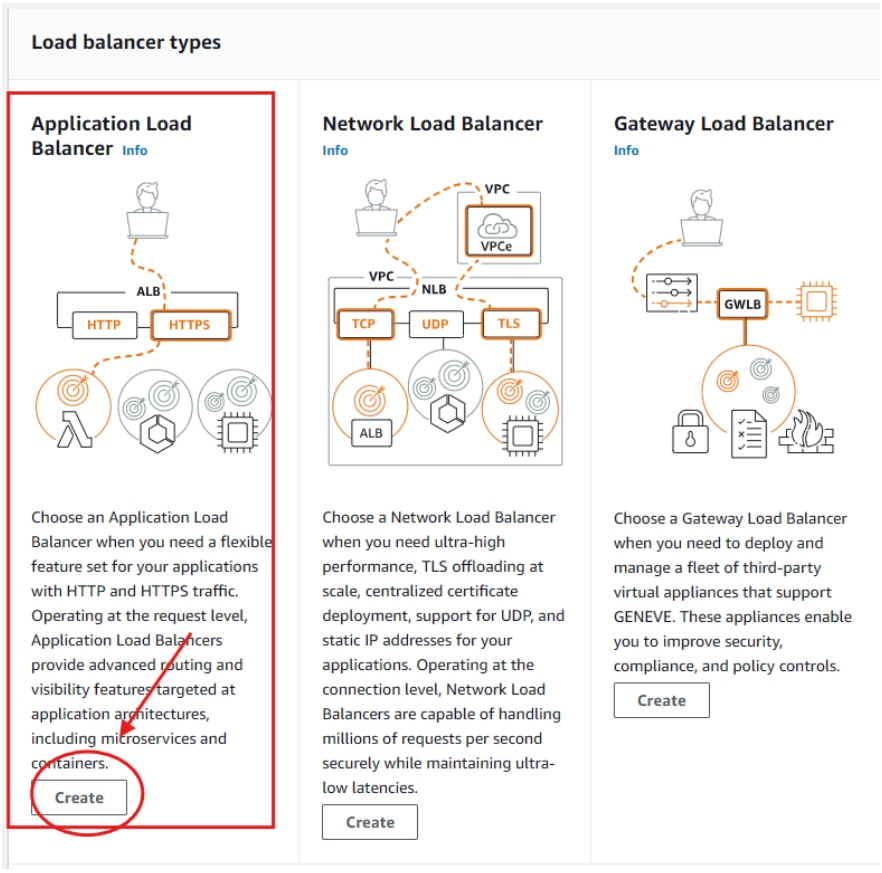


Creation of Application Load Balancer (ALB)

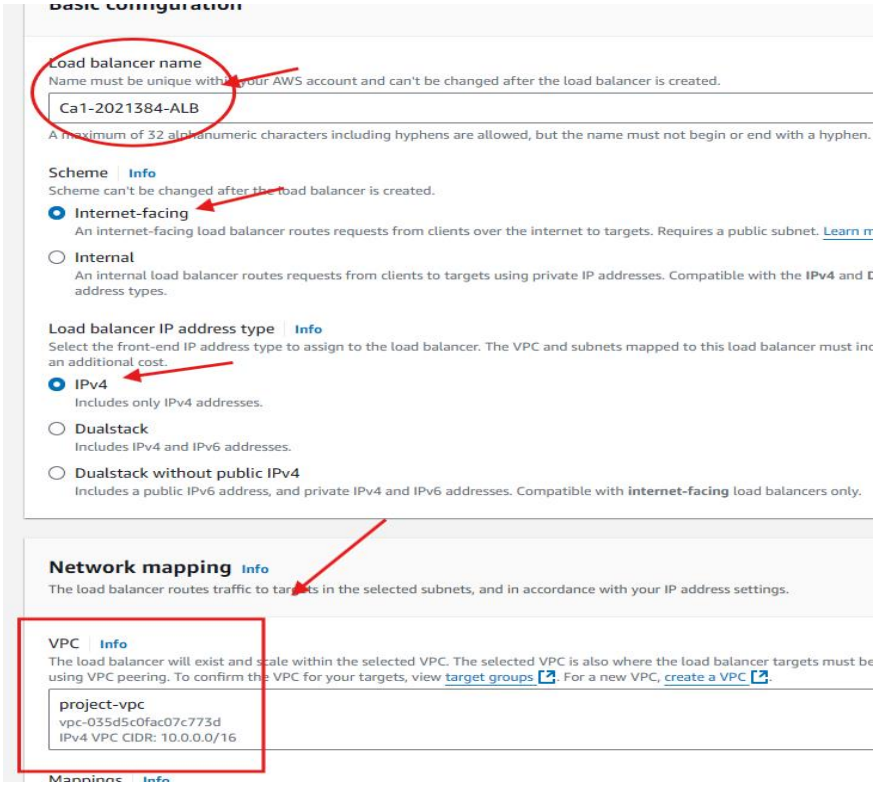
- Click create load balancer by navigating from left menu in EC2.



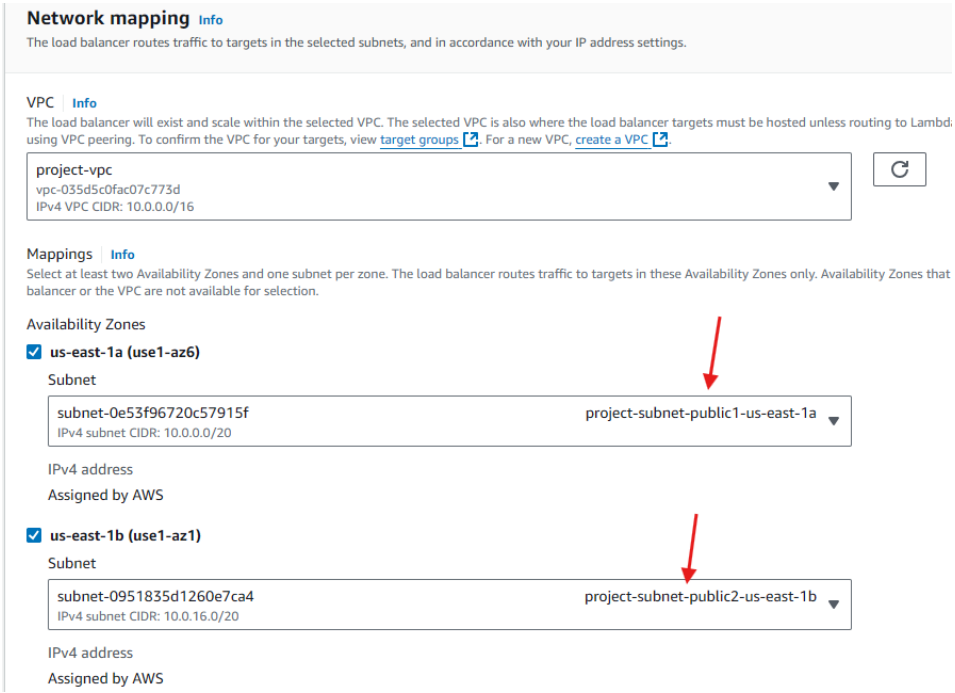
- Selected Application Load Balancer (ALB) and click create.



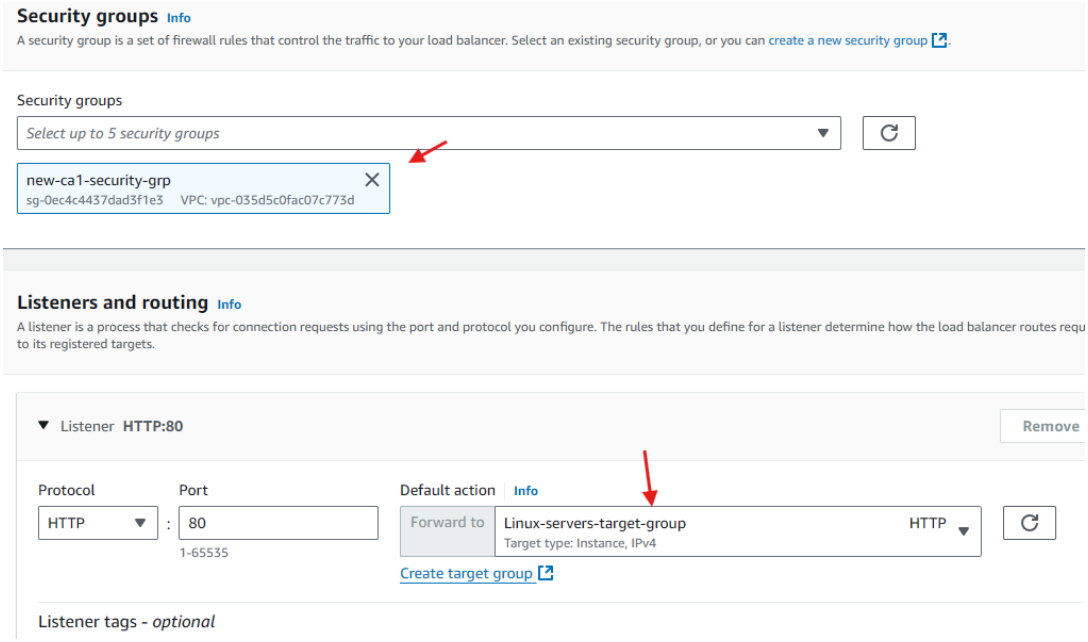
- Gave name as ca1-2021384-ALB.
- Set internet-facing as Scheme with IPv4 IP address.
- Select custom project-vpc in Network mapping.



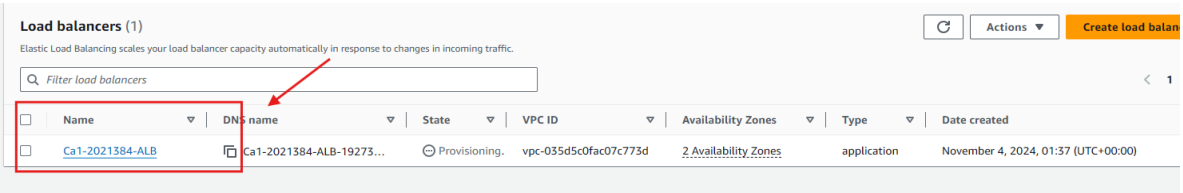
- Selected both public subnets in both Availability Zones.



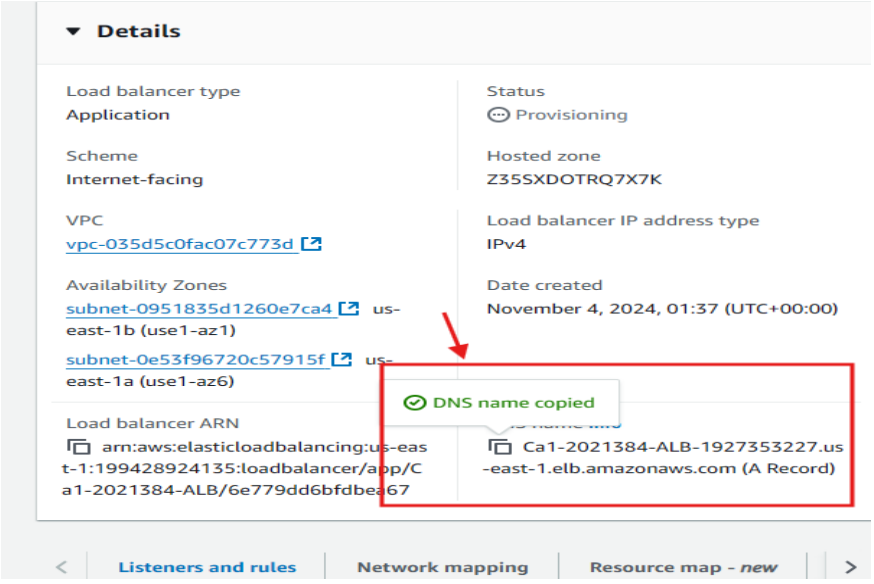
- Selected custom created new-ca1-security-grp for security group and newly created target group Linux-servers-target-group for listeners from HTTP port 80.



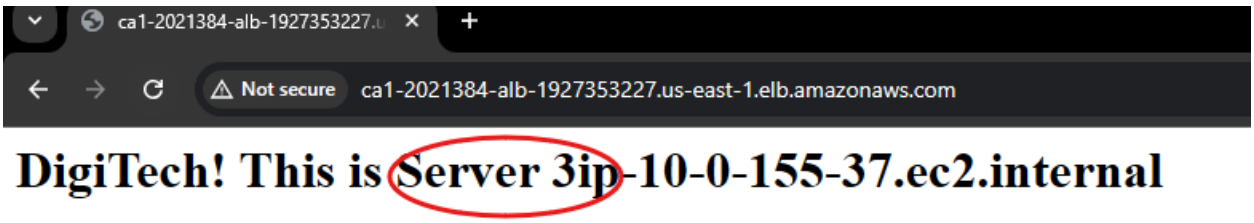
- Ca1-2021384-ALB application load balancer has created successfully.

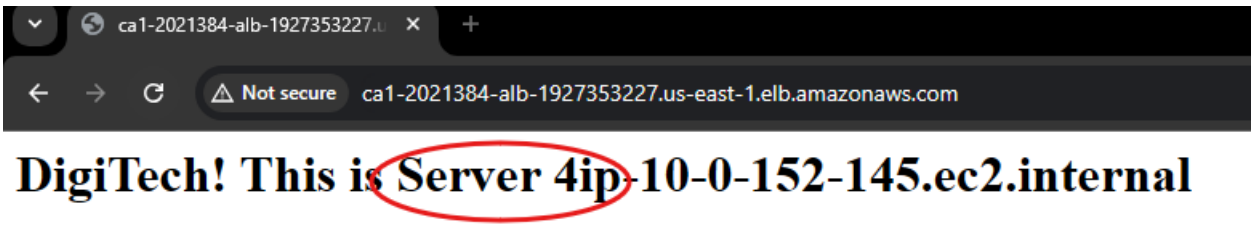


- Now, Go inside the ALB and copy the DNS name of Ca1-2021384-ALB and search in browser.



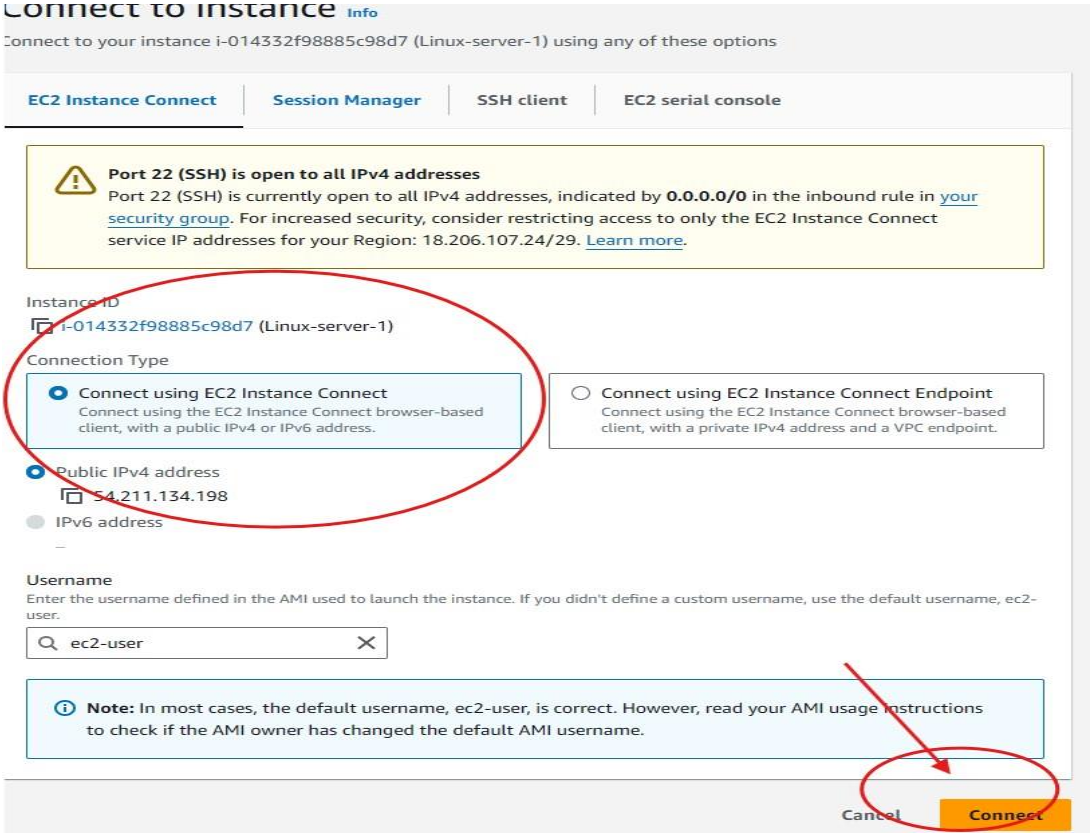
- It can be seen in the screen shots below the Application Load Balancer (ALB) is working properly by reloading the page again and again it showing different servers with the same DNS name URL in browser.



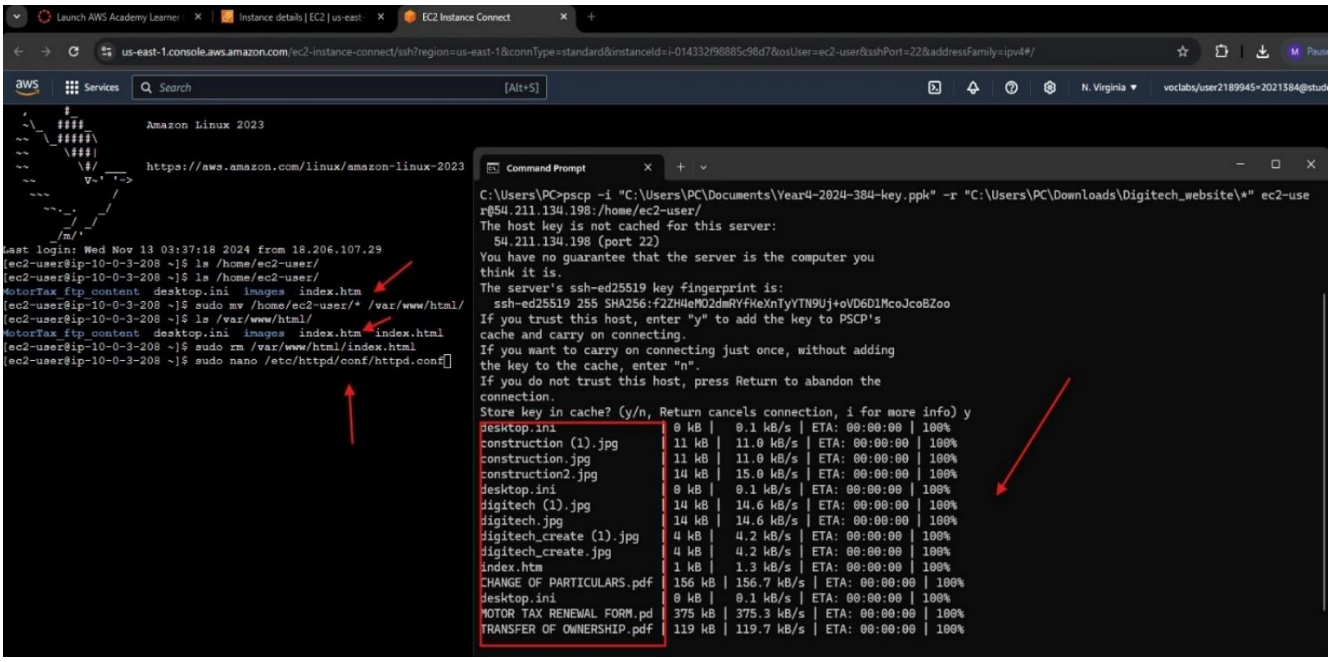


Q2(b): Challenge Task: Incorporate the DigiTech Website with ALB

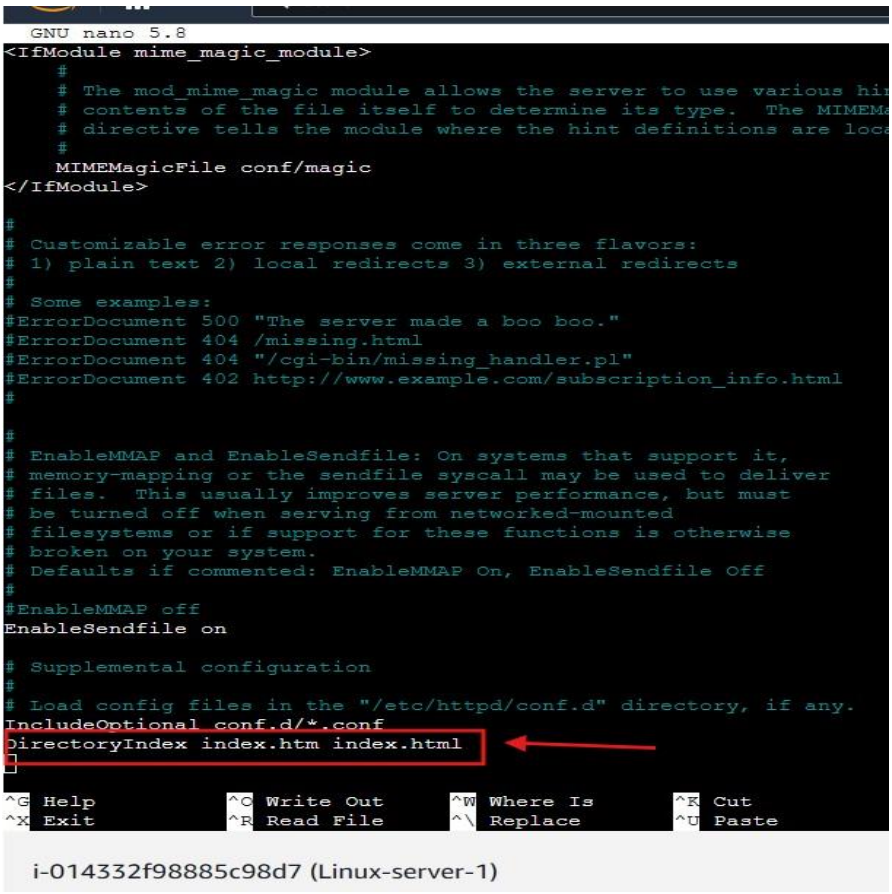
- Go inside each server one by one and connect them using EC2 instance connect for uploading the DigiTech website and configuration.



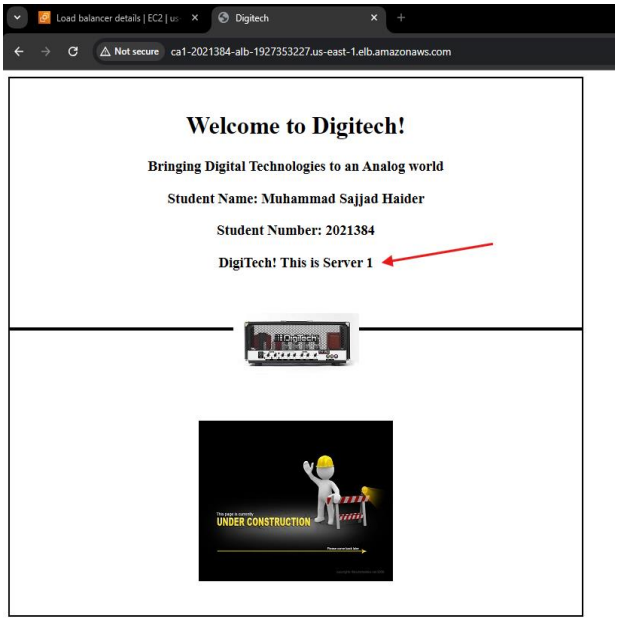
- Uploaded Digitech web content inside each server from local machine by using pscp command in command prompt with the .ppk key which used for each server.
- By using this method I uploaded Digitech website in each machine's home by using /home/ec2-user/ at the end of pscp command.
- Then, move all files from /home/ec2-user/ directory to /var/www/html/ directory by using sudo mv command.
- Removed default Apache html page by using sudo rm command from /var/www/html/ directory.
- Configure the httpd page with sudo nano.

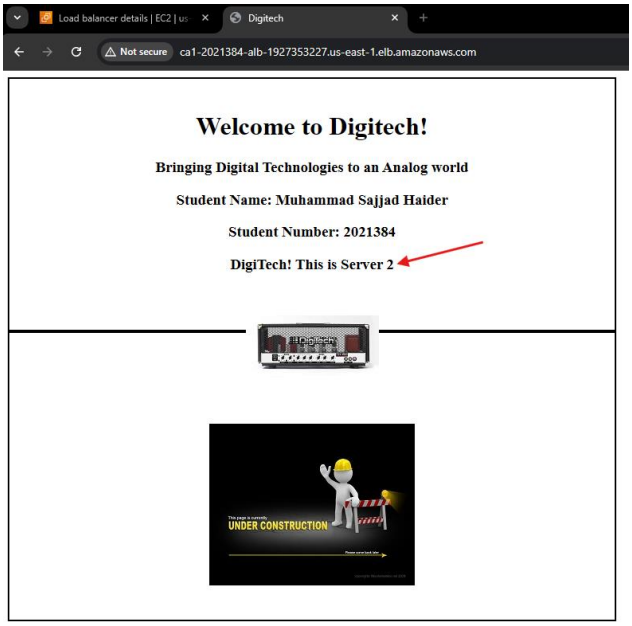


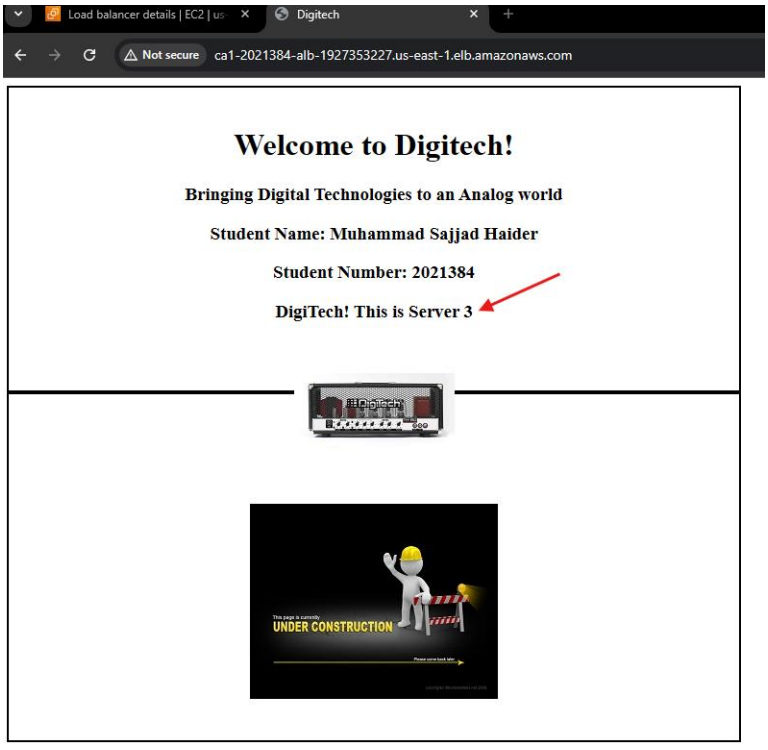
- Set DirectoryIndex inside the httpd file for giving priority to Digitech html file first while displaying and save file by ctrl x.
- Run command sudo systemctl restart httpd for restarting.

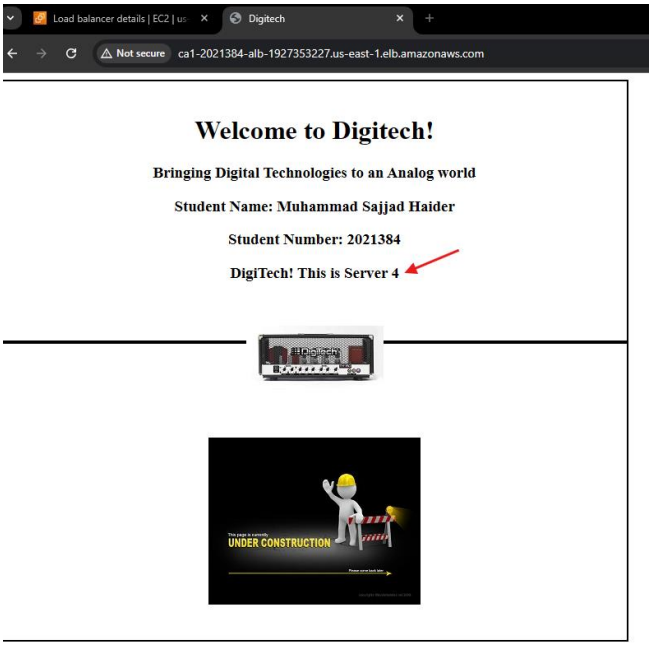


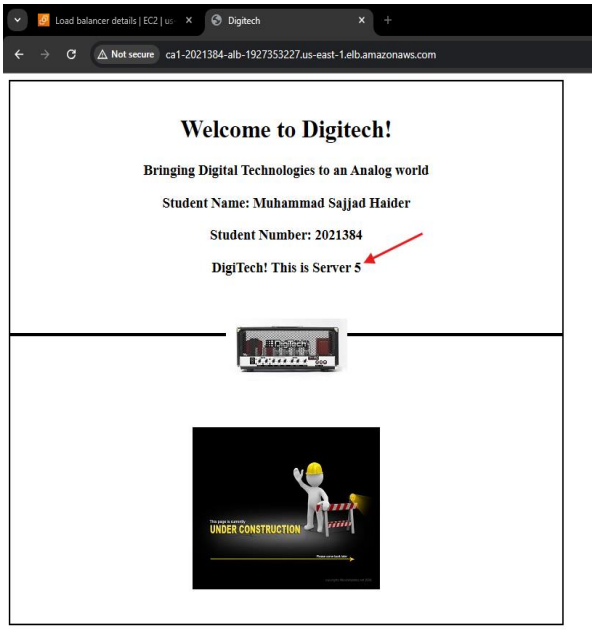
- After configuring each Linux-server instance.
- Go inside Application Load Balancer (ALB).
- Copy DNS name and search in browser.
- It can be seen the Digitech webpage is displaying with different server names by applying ALB.







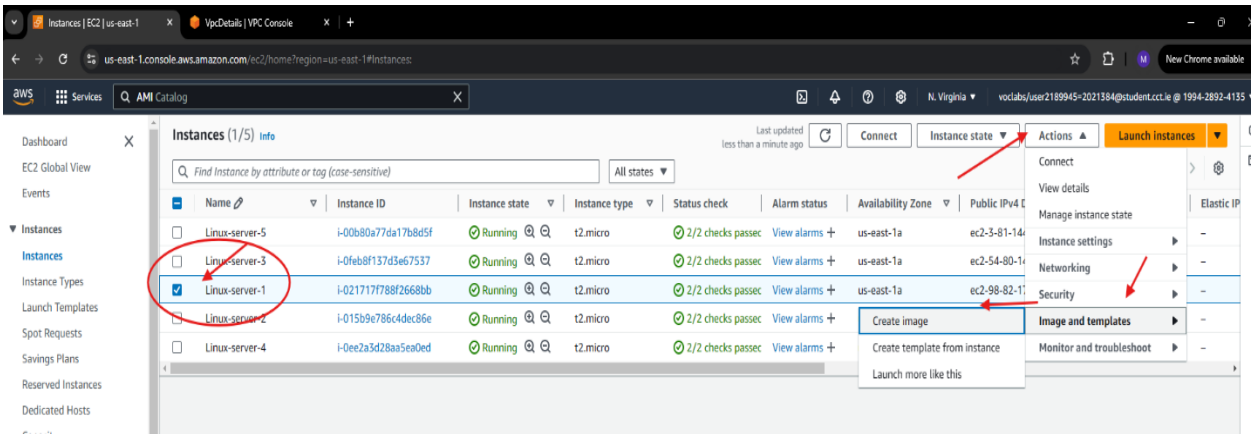

Load balancer details | EC2 | us
Digitech
Not secure
ca1-2021384-alb-1927353227.us-east-1.elb.amazonaws.com
Welcome to Digitech!
Bringing Digital Technologies to an Analog world
Student Name: Muhammad Sajjad Haider
Student Number: 2021384
DigiTech! This is Server 4
UNDER CONSTRUCTION


Load balancer details | EC2 | us
Digitech
Not secure
ca1-2021384-alb-1927353227.us-east-1.elb.amazonaws.com
Welcome to Digitech!
Bringing Digital Technologies to an Analog world
Student Name: Muhammad Sajjad Haider
Student Number: 2021384
DigiTech! This is Server 5
UNDER CONSTRUCTION

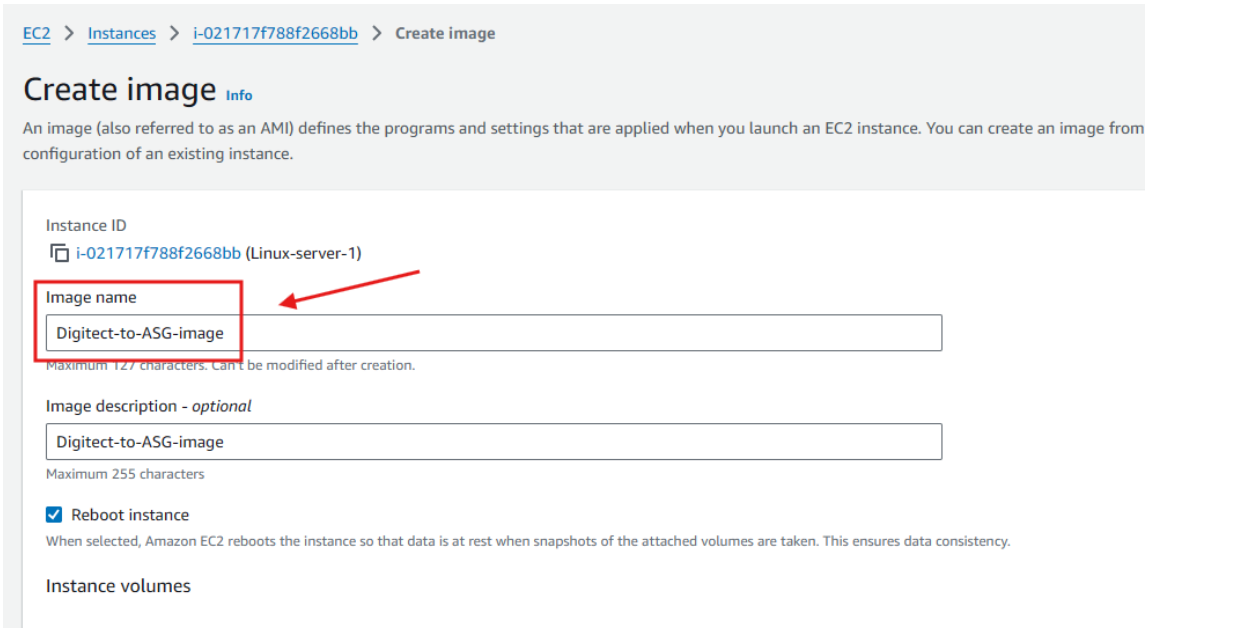
Q3(a): Challenge Task: Proof Of concept, Auto Scaling Group with ALB

Creation AMI image:

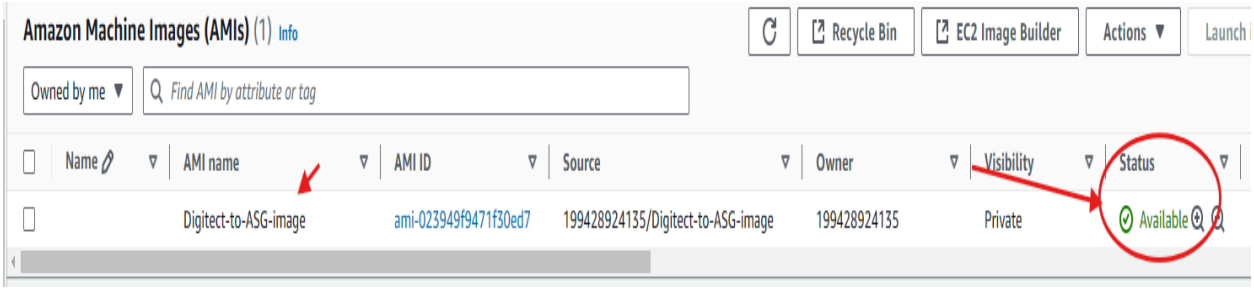
- Firstly, I created AMI image from Digitech website hosting Linux-server-1 for corporation with Clouds-Are-Us.
- Select Linux-server-1, click Actions.
- Go to Image and templates and click Create Image.



- Set Image name as Digitect-to-ASG-image.
- Click Create.

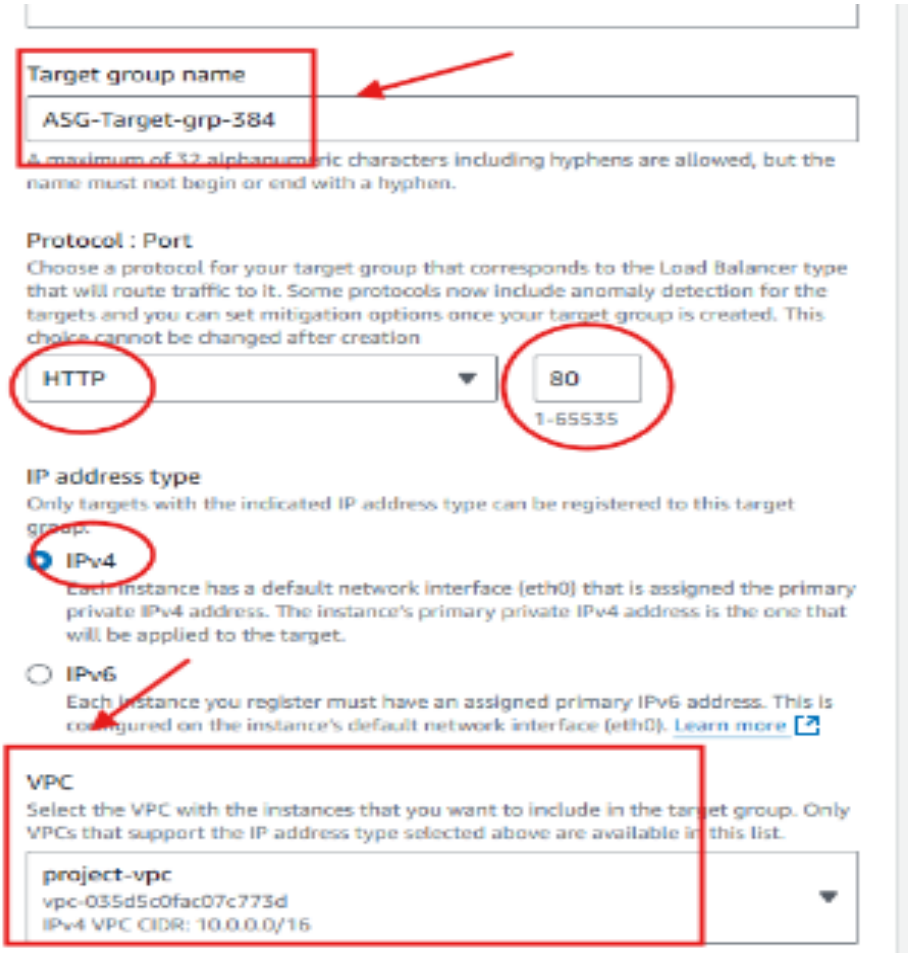


- Navigate AMIs in EC2, it can be seen the Digitect-to-ASG-image has created.

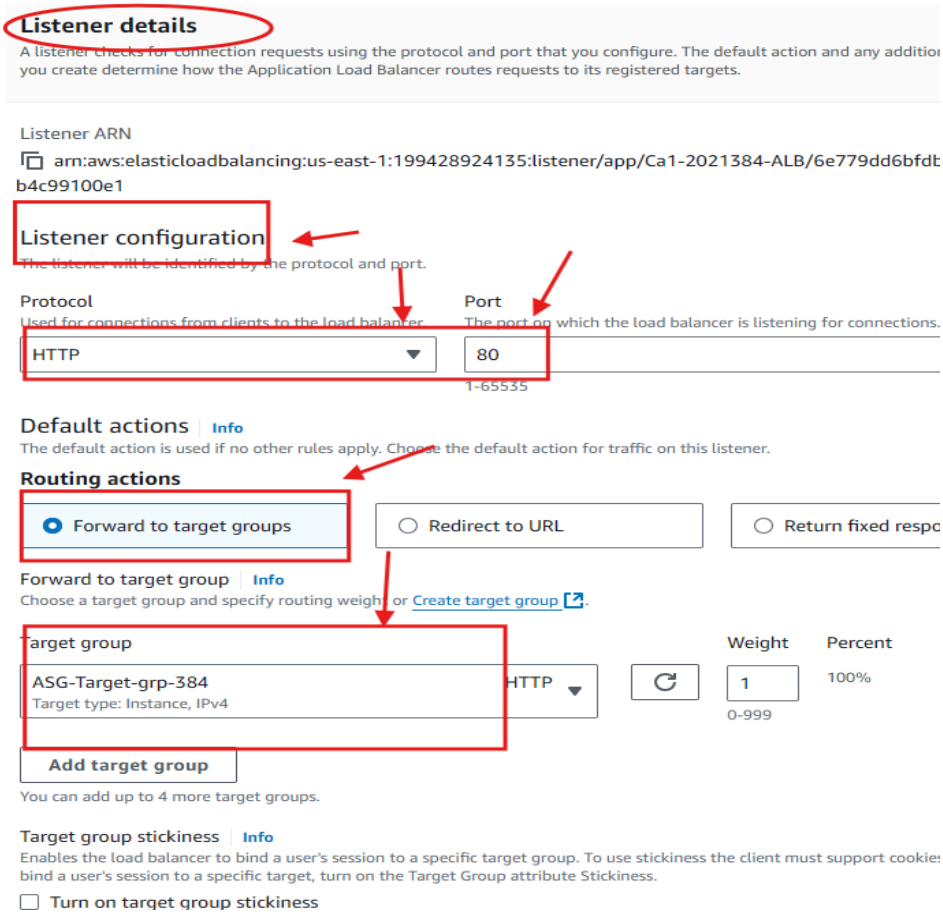


Target group for ASG:

- Set Target group name as ASG-Target-grp-384.
- HTTP protocol with port 80 and IPv4 as IP address type.
- Attached custom created VPC project-vpc.
- Click Create.

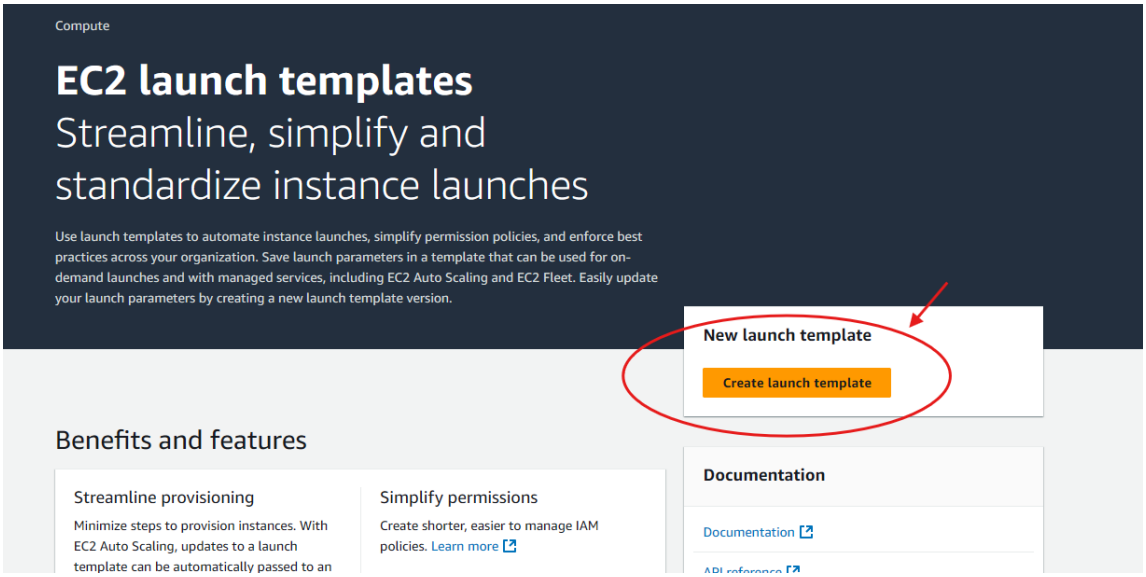


- Updated ALB, Attach ASG-Target-grp-384 with forward routing actions at HTTP port 80 in Listener details.

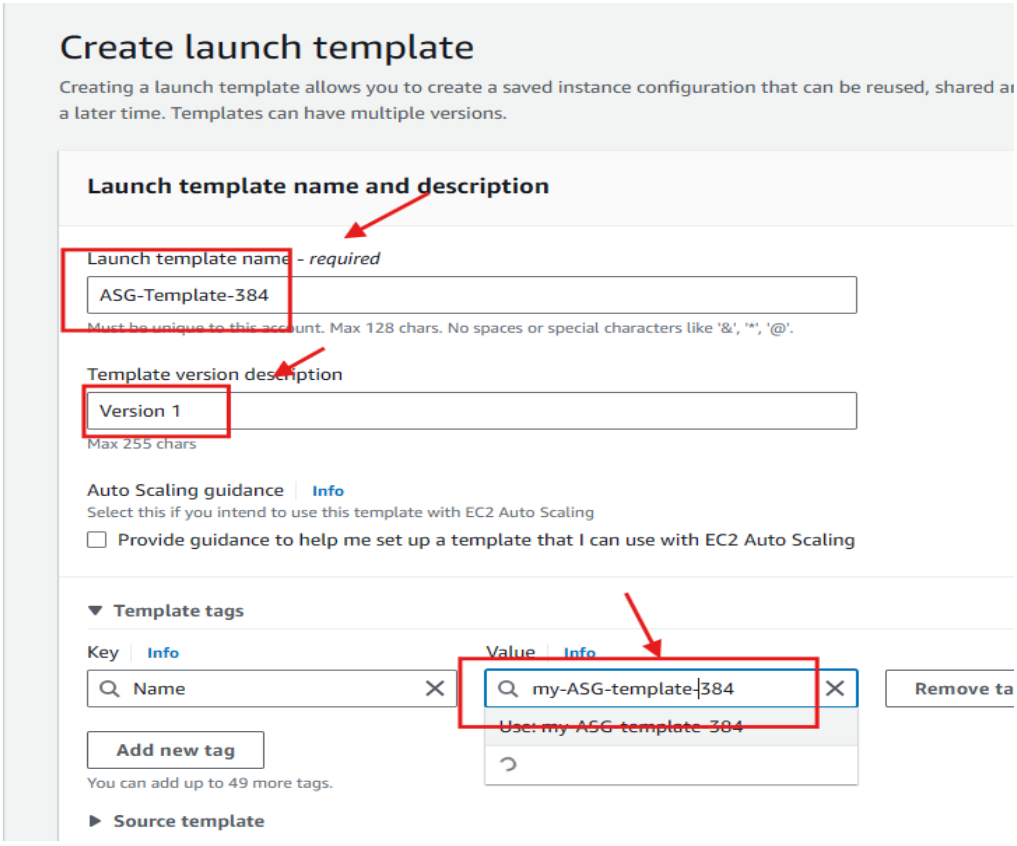


Creation of Launch Template:

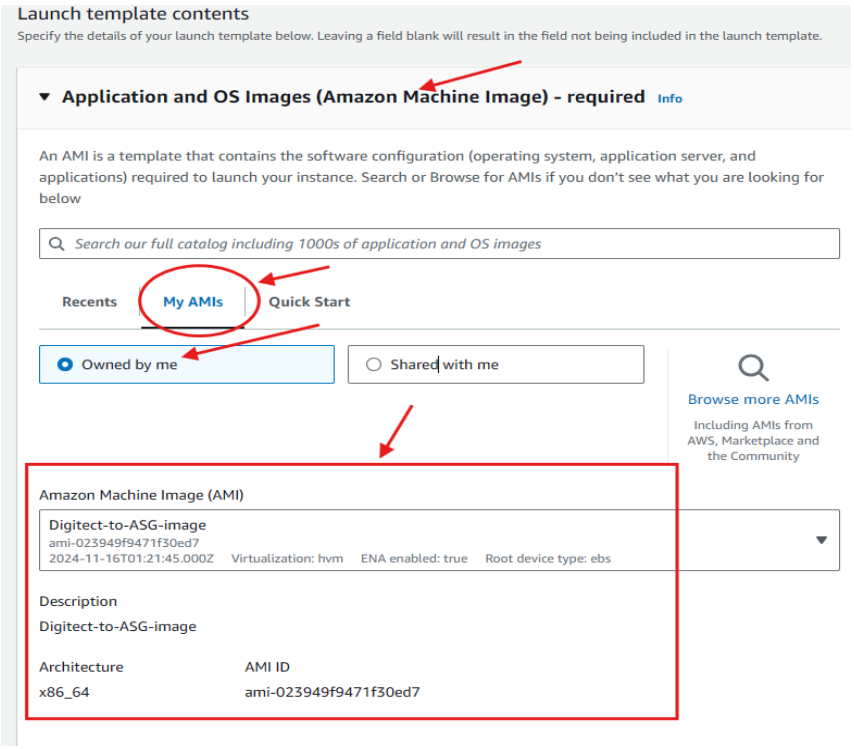
- Navigate Launch Template inside EC2.
- Create Launch Template for Auto scaling group (ASG).



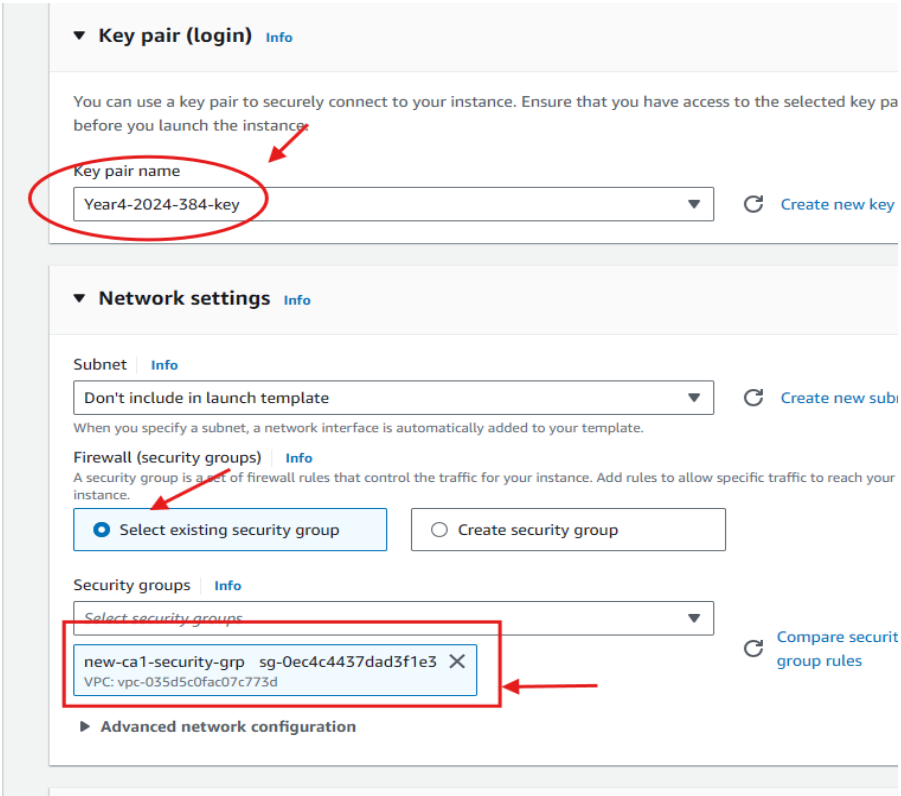
- Gave template name as ASG-Template-384.
- Set version 1 and set tag value as my-ASG-template-384.



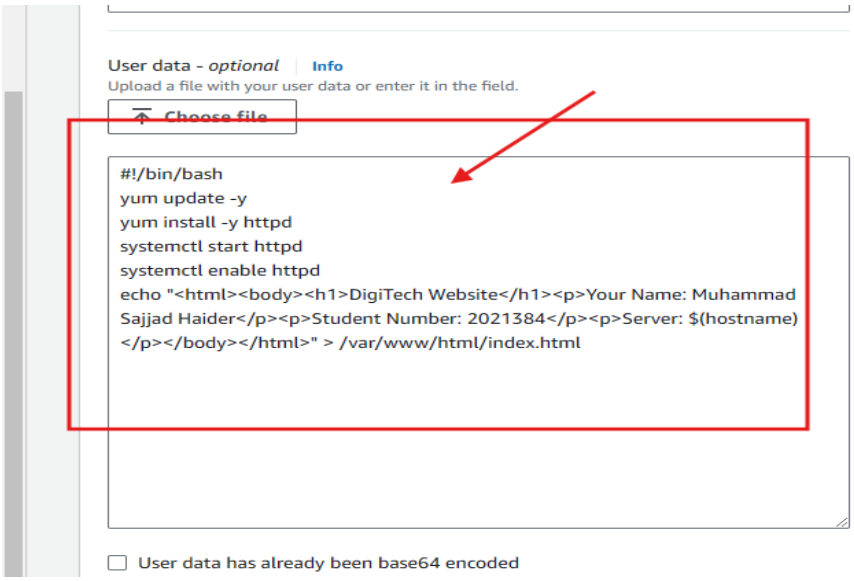
- Attached Digitect-to-ASG-image as Amazon Machine Image (AMI) , which is created for corporation with Clouds-Are-Us.



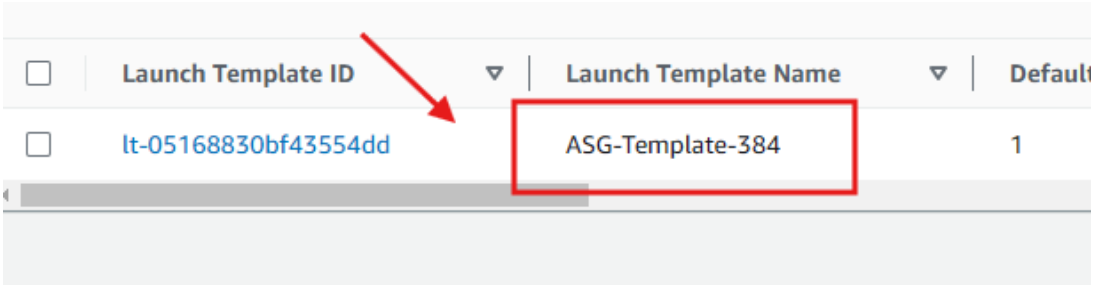
- Stick with Year4-2024-384-key and existing new-ca1-security-grp.



- Simple User data script for Launch Template.

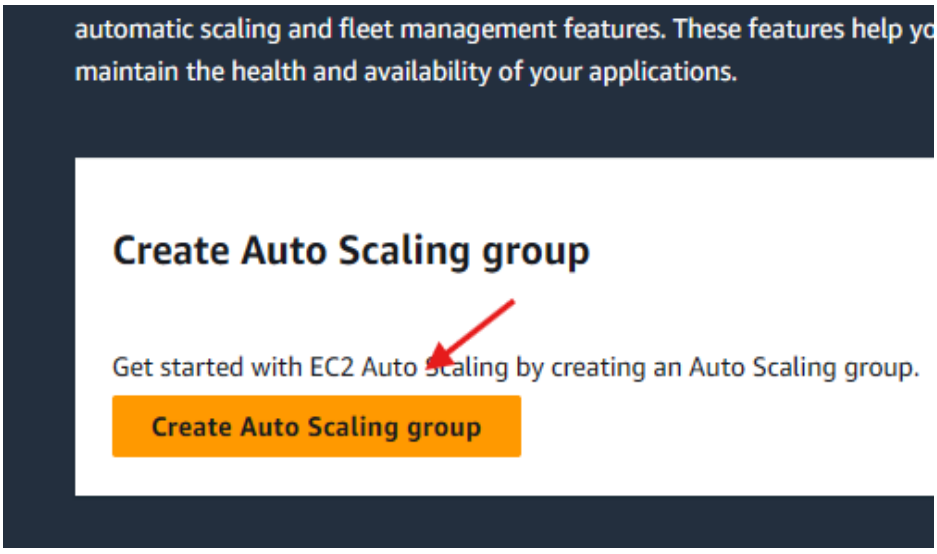


- ASG-Template-384 has successfully created.

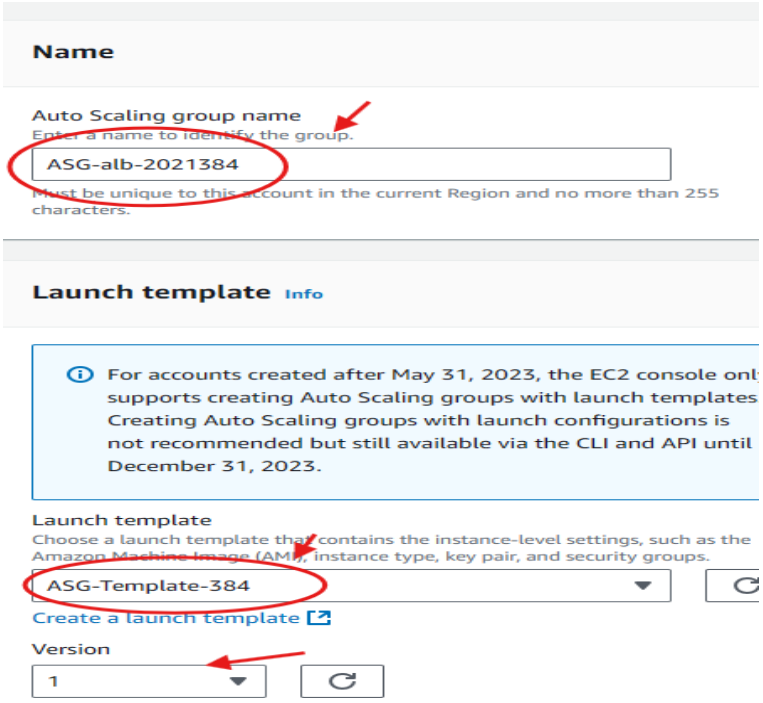


Creation of ASG with attached ALB:

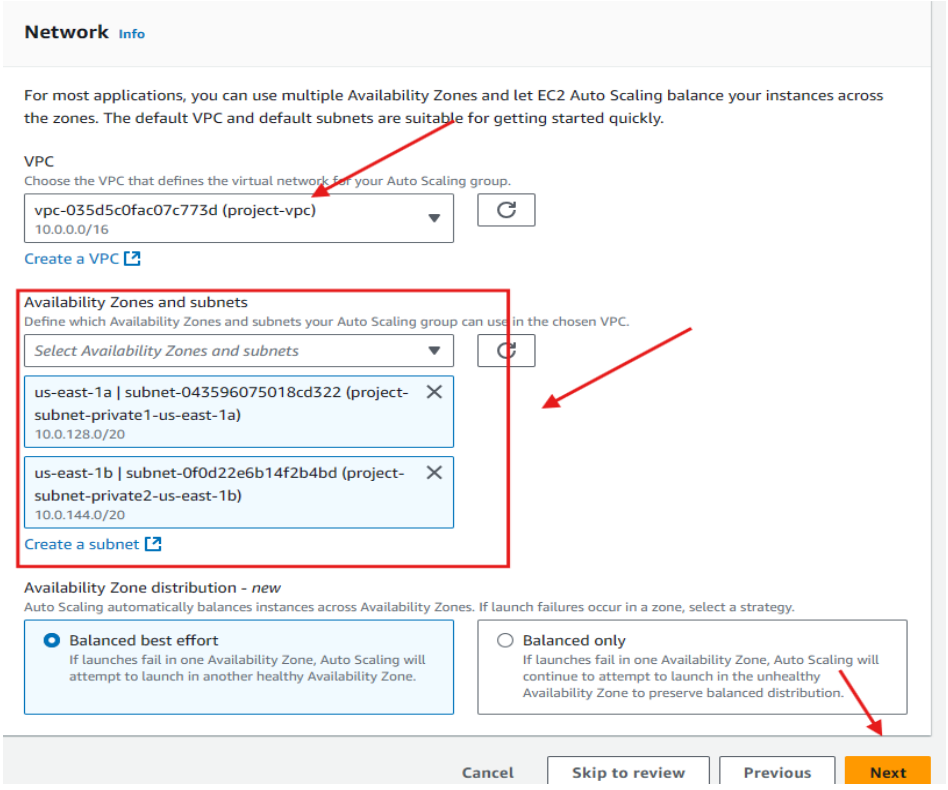
- Navigate Auto scaling group (ASG) in EC2 and Click create Auto Scaling Group.



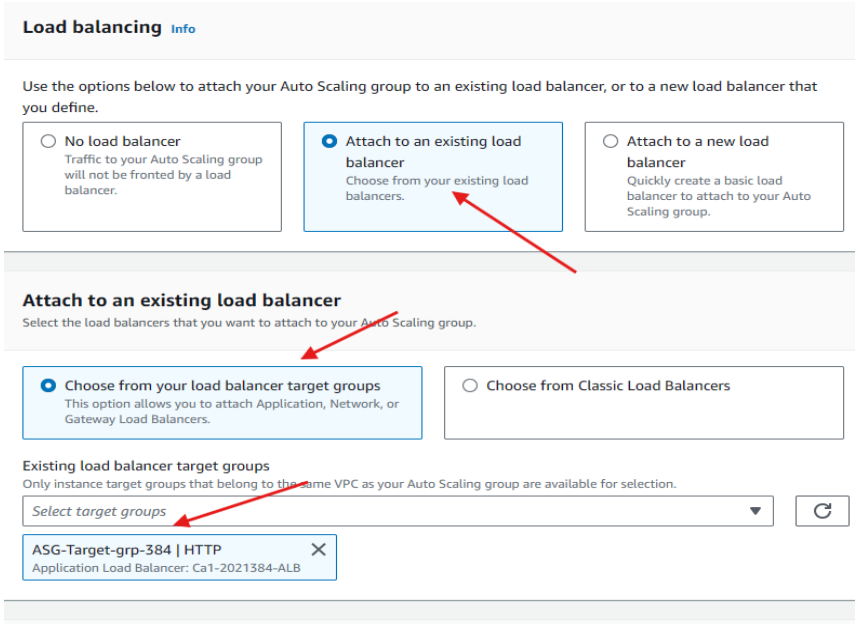
- Set name as ASG-alb-2021384 and attached ASG-Template-384 custom created with version 1.



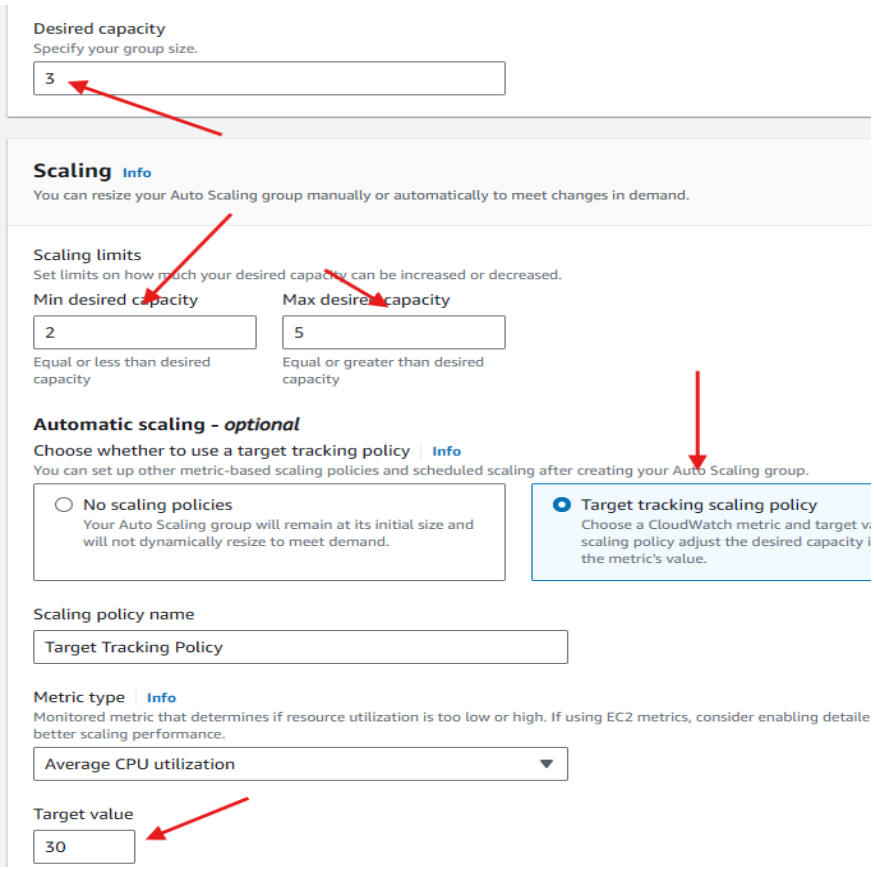
- In Network, Attached custom created project-vpc.
- Attached private subnets only of both Availability Zones and hit Next.



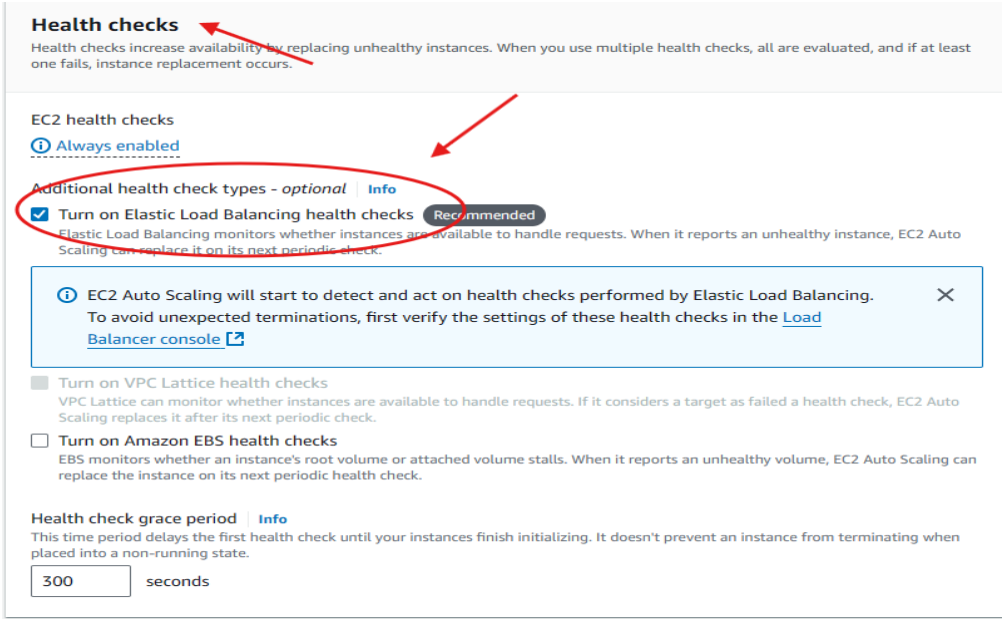
- Attach Existed Application Load Balancer (ALB) from ASG-Target-grp-384 of load balancer which configured above.



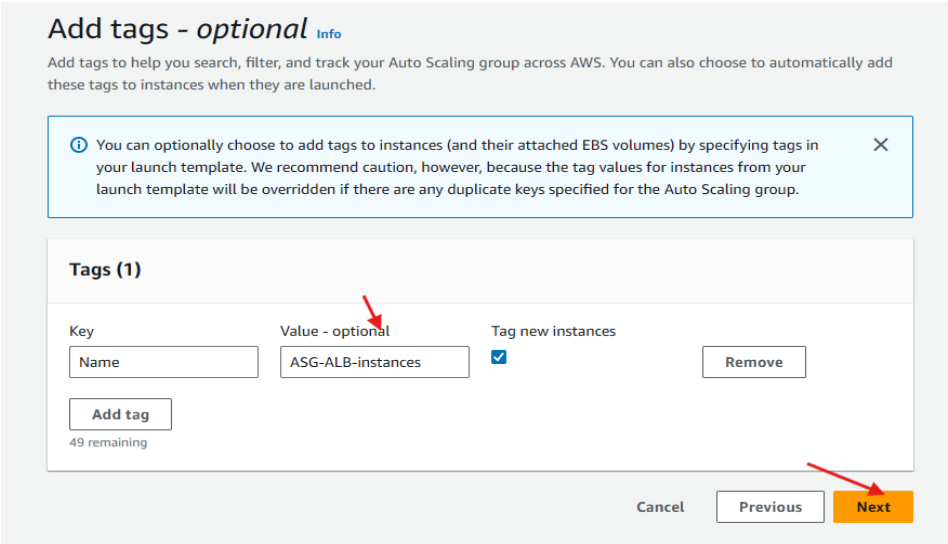
- Set desired capacity at 3 and scaling limits min max desired capacities at 2 and 5, respectively.



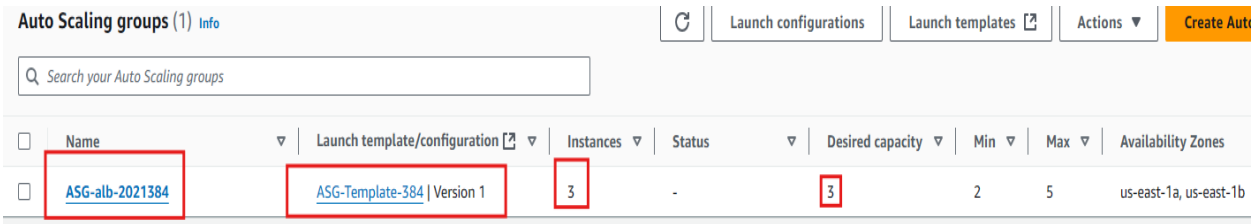
- Set Health Checks with Elastic Load Balancing.



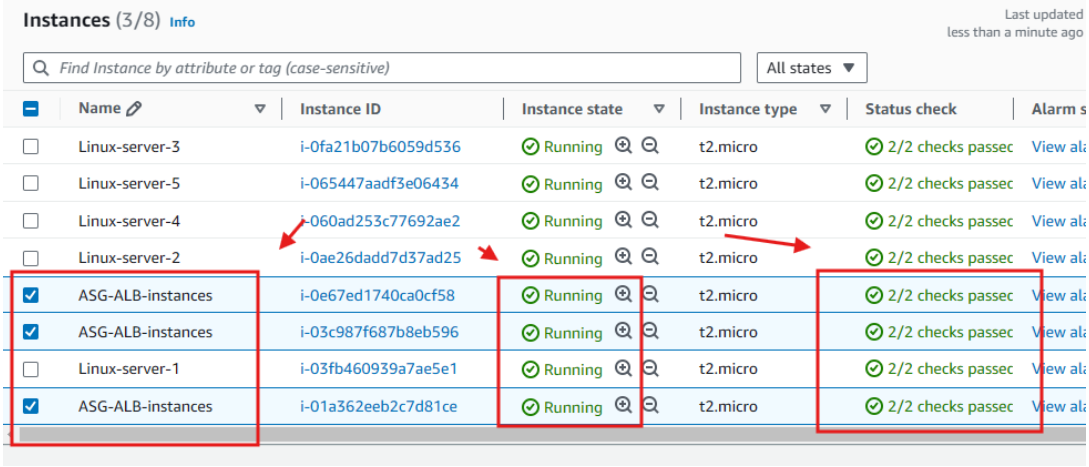
- Add Tags Value as ASG-ALB-instances.



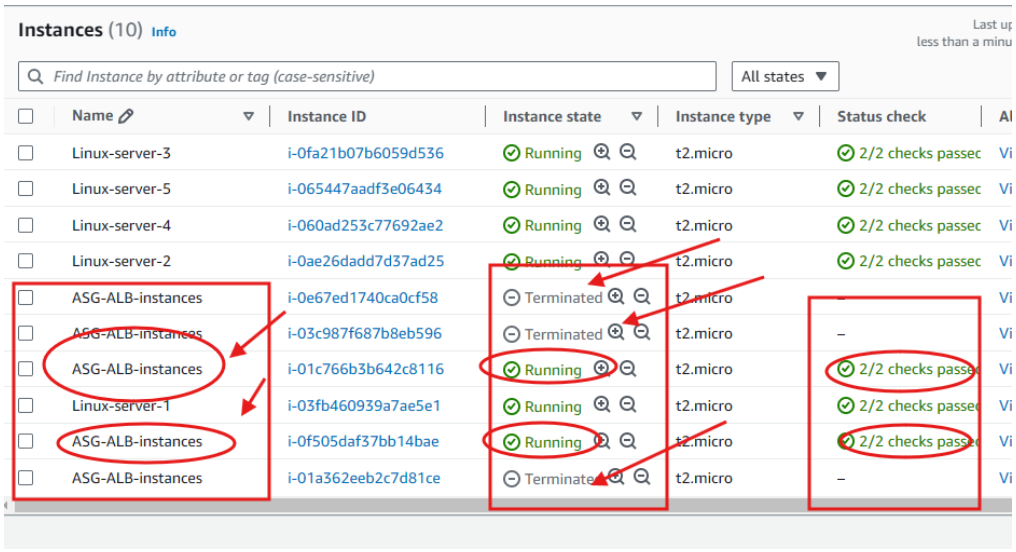
- ASG-alb-2021384 auto scaling group has created successfully.



- ASG-ALB-instances are appeared and running successfully.

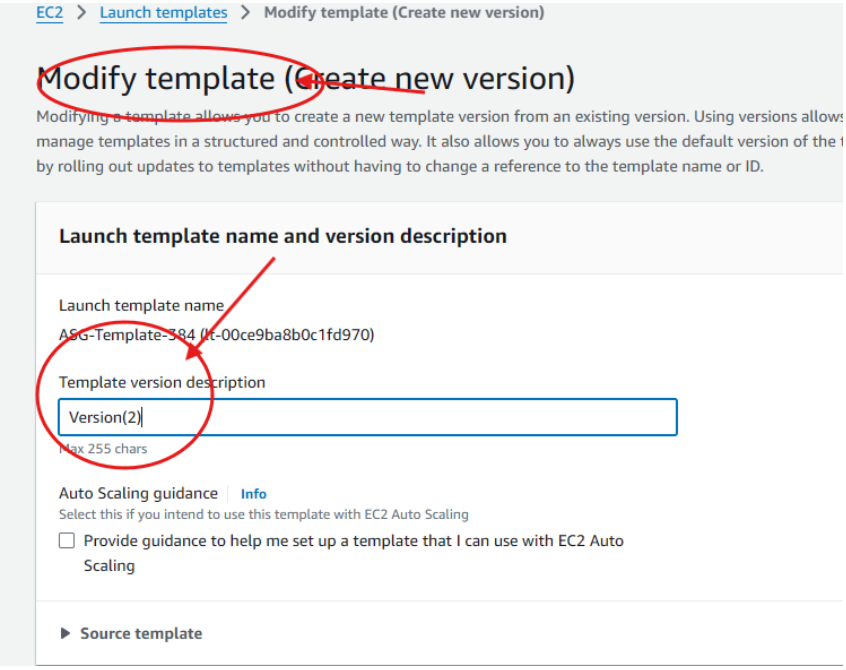


- It can be seen that ASG-ALB-instances are coming again and again after termination.

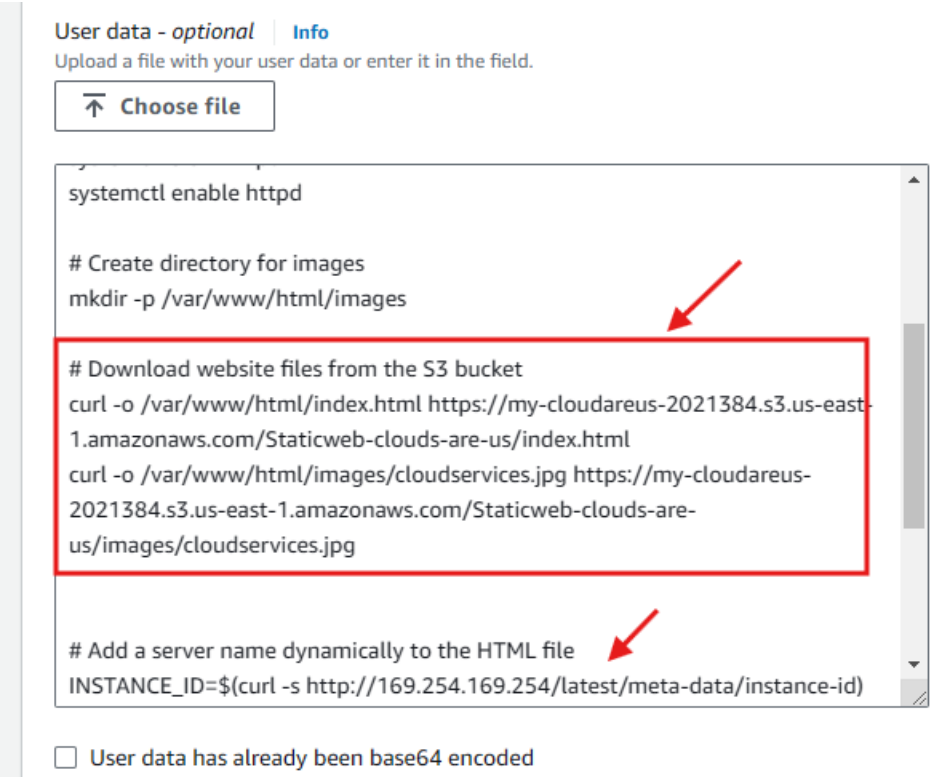


Modify Launch Template and Edit ASG:

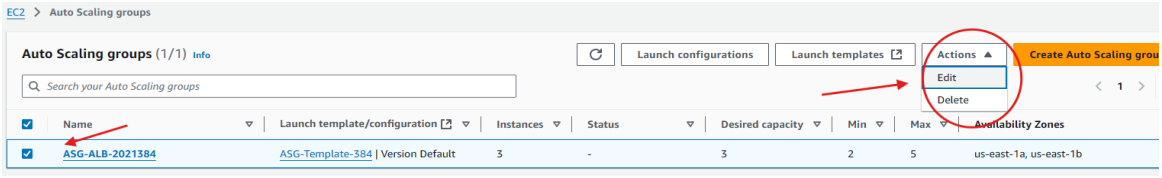
- Go back to Launch template and click edit.
- Set Version (2).



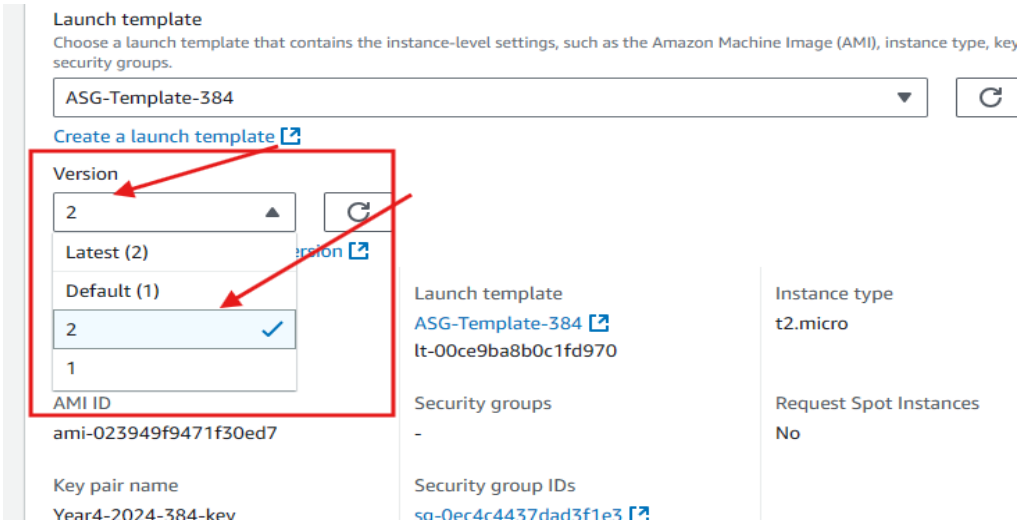
- Edited User data, Add curl commands with public URLs of index.html and image objects from S3 static website bucket which configures in Task 1(a).



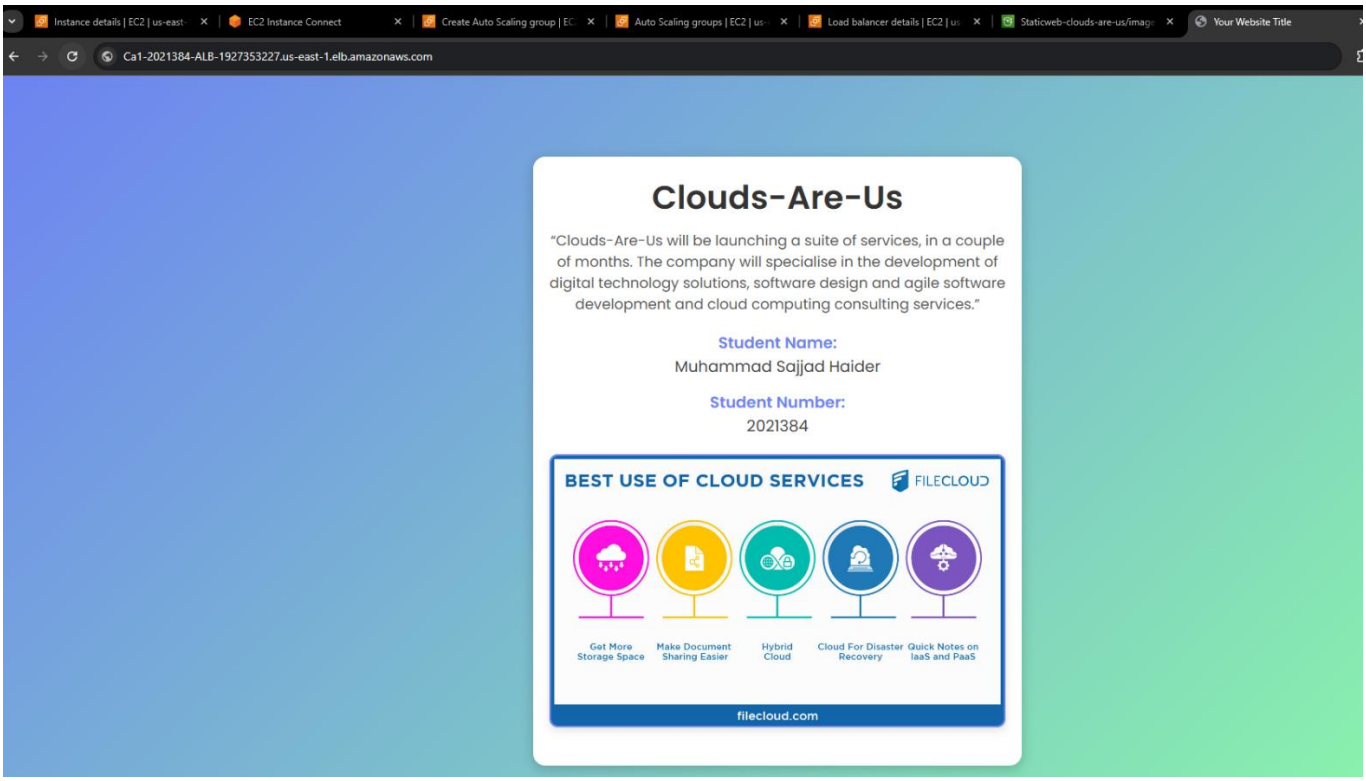
- Edit Auto scaling group for modified Launch template.



- Attached Modified ASG-Template-384 Version 2.



- Copy the DNS name of ALB and search in browser and Clouds-Are-Us web page hosted Successfully.



Q3(b): Research Task: Principles of ASG with ALB

The use of the proposed DigiTech Application Load Balancer (ALB) is to duly spread the operation business across targets similar as IP addresses and EC2 cases, or holders. The load balancing maximizes the vacuity and trustability of DigiTech's apps by icing that one case does not get overwhelmed with business. These are the effects it provides, how it works, and that would be employed in this case the architectural design generalities according to AWS Well-Architected Framework.

The Principle of Operation of DigiTech ALB

Business Distribution:

It uses content, paths, and HTTP heads as parameter to route incoming requests to the applicable target destination depending on set of conditions. This also ensures the stylish but possible utilization of the coffers and response time.

Health Checks:

The ALB always monitors the health of its targets at any one time. The ALB routing intelligent directs business to healthy targets while avoiding cases that have come unhealthy or do not respond.

Subcaste 7 Routing:

In its operation, ALB works at the operation subcaste (Subcaste 7) I/ E it's able of making routing opinions grounded on specific operation- subcaste data. This includes features that can give better and further extents control of operation business to colorful operation waiters, including host grounded or path grounded routing.

Principles for combining ASG with ALB

The operations that are being run by DigiTech have some assurance of being suitable to continue running in the event that commodity is wrong or if there's too important business thanks to the ALB performing business distribution and also health checks.

Scalability: Using the integration of ALB and ASG, DigiTech can have a feature of automatic resource scaling depending on the demand. Improvements in the user experience will occur due to their capacity for traffic handling without owners' interferences.

Cost Efficiency: DigiTech can curb costs by only using what they have to and maintain efficiency by changes its resources based on traffic needs.

Enhanced Security: For enhanced protection against established common web strikes, an ALB can communicate with AWS web application firewall (WAF).

Sources of Information The principles of architectural design will be derived from both primary and secondary research.

Scalability: Scalability is one of the key concepts of architectural principles of the AWS architecture that confer a promise of the application's ability to elastically grow or shrink based on the load requirements. It is possible to add or delete instances of DigiTech based on traffic intensity with the help of association with an ASG of the ALB.

Fault Tolerance: Another concept is the design for fault tolerance. By frequent checking of the health of targets and the directing of traffic towards healthy instances, the ALB also significantly contributes to the formation of an extremely reliable architecture with minimal or no downtime. While using ASG, instances can be horizontally expanded or rebuilt to ensure reliability in the execution of the offered services by DigiTech with no impact on the availability of the application.

To sum up, with regard to its interaction with Auto Scaling Groups, traffic handling capabilities and advocacy of scalability and reliability, the DigiTech ALB enhances the general architecture.

References:

- Eliot, S. and Valverde, L. (2018). *The 6 Pillars of the AWS Well-Architected Framework*. [online] Amazon Web Services. Available at: https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/.

- Jain, A. (2023). *Amazon AWS 5 Pillars of Well-Architected Framework*. [online] Cloud Training Program. Available at: https://k21academy.com/amazon-web-services/aws-solutions-architect/6-pillars-of-aws-well-architected-framework/.

- www.stormit.cloud. (n.d.). *6 pillars of AWS Well Architected Framework | StormIT*. [online] Available at: https://www.stormit.cloud/blog/6-pillars-of-aws-well-architected-framework/.

- GeeksforGeeks. (2021). *Difference Between AWS S3 and AWS EBS*. [online] Available at: https://www.geeksforgeeks.org/difference-between-aws-s3-and-aws-ebs/.

- Stack Overflow. (n.d.). *AWS EFS vs EBS vs S3 (differences & when to use?)*. [online] Available at: https://stackoverflow.com/questions/29575877/aws-efs-vs-ebs-vs-s3-differences-when-to-use.

- Hallett, N. (2020). *EBS vs EFS vs S3 - when to use AWS' three storage solutions*. [online] Just After Midnight. Available at: https://www.justaftermidnight247.com/insights/ebs-efs-and-s3-when-to-use-awss-three-storage-solutions/.

- AWS Official (2017). *How do I utilize user data to automatically run a script with every restart of my Amazon EC2 Linux instance?* [online] Amazon Web Services, Inc. Available at: https://repost.aws/knowledge-center/execute-user-data-ec2 [Accessed 16 Nov. 2024].

- Amazonaws.cn. (2024). Create launch templates with Amazon CloudFormation - Amazon CloudFormation. [online] Available at: https://docs.amazonaws.cn/en_us/AWSCloudFormation/latest/UserGuide/quickref-ec2-launch-templates.html [Accessed 16 Nov. 2024].

- docs.aws.amazon.com. (n.d.). Tutorial: Set up a scaled and load-balanced application - Amazon EC2 Auto Scaling. [online] Available at: https://docs.aws.amazon.com/autoscaling/ec2/userguide/tutorial-ec2-auto-scaling-load-balancer.html.