

Dharmi Gujarati

CWID: 20018001

Lab Assignment: 3

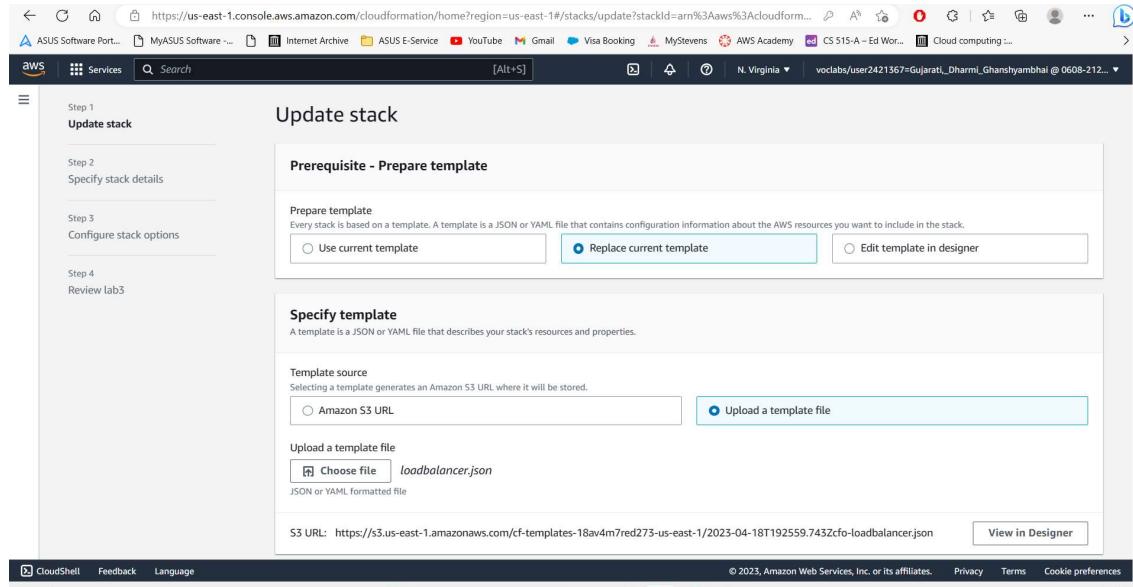
- Login to AWS academy.
- Open CloudFormation in services and click on create stack.

The screenshot shows the AWS CloudFormation home page. On the left, there's a sidebar with options like Stacks, StackSets, Exports, Designer, Registry (Public extensions, Activated extensions, Publisher), Spotlight, Feedback, CloudShell, Feedback, and Language. The main content area features the title "AWS CloudFormation" and the subtitle "Model and provision all your cloud infrastructure". Below this is a section titled "How it works" with a video player showing a thumbnail of a person working on a computer. To the right, there's a "Create a CloudFormation stack" box with a "Create stack" button, and a "Getting started" sidebar with links for "What is AWS CloudFormation", "Getting started with CloudFormation", "Learn template basics", and "Quick starts". At the bottom, there's a "More resources" section and a footer with copyright information and links for Privacy, Terms, and Cookie preferences.

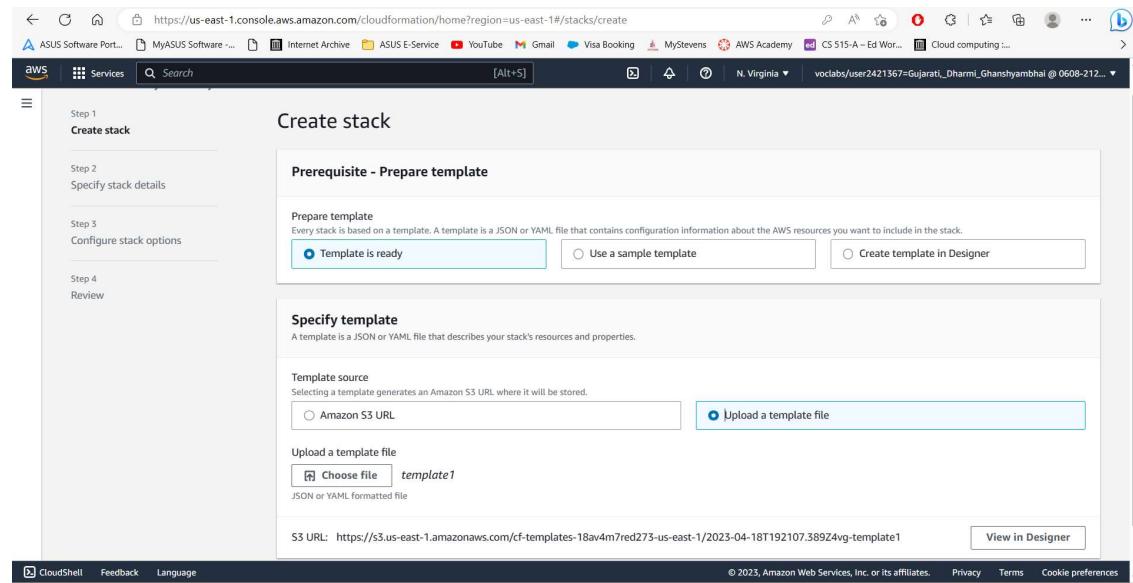
- Select the Use as a sample template option and then select view in designer.

The screenshot shows the "Create stack" wizard at Step 2: "Prerequisite - Prepare template". The left sidebar lists steps: Step 2 (Specify stack details), Step 3 (Configure stack options), Step 4 (Review). The main panel has a "Prepare template" section with a note that every stack is based on a template. It shows three radio button options: "Template is ready" (selected), "Use a sample template", and "Create template in Designer". Below this is a "Specify template" section with a note that a template is a JSON or YAML file. It has a "Template source" section for selecting an Amazon S3 URL or uploading a template file. A file input field shows "template1" selected. At the bottom, there's a "S3 URL" field with the value "https://s3.us-east-1.amazonaws.com/cf-templates-p6bff7jwpgye-us-east-1/2023-04-18T193107.742Zyes-template1" and a "View in Designer" button. At the very bottom, there are "Cancel" and "Next" buttons.

- Then, add Default as a key pair name that is Dharmi in file and save it as .json in my device.



- After went back and select upload a template file and upload that saved file here.



- Then click next and add stack name as lab3. Click next until submit it.

Step 2
Specify stack details

Stack name
 Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

InstanceType
WebServer EC2 Instance type

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instance

SSHLlocation
The IP address range that can be used to SSH to the EC2 instances

Cancel Previous Next

ROLL BACK ALL STACK RESOURCES
 Roll back the stack to the last known stable state.
 Preserve successfully provisioned resources
 Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.

Advanced options
 You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

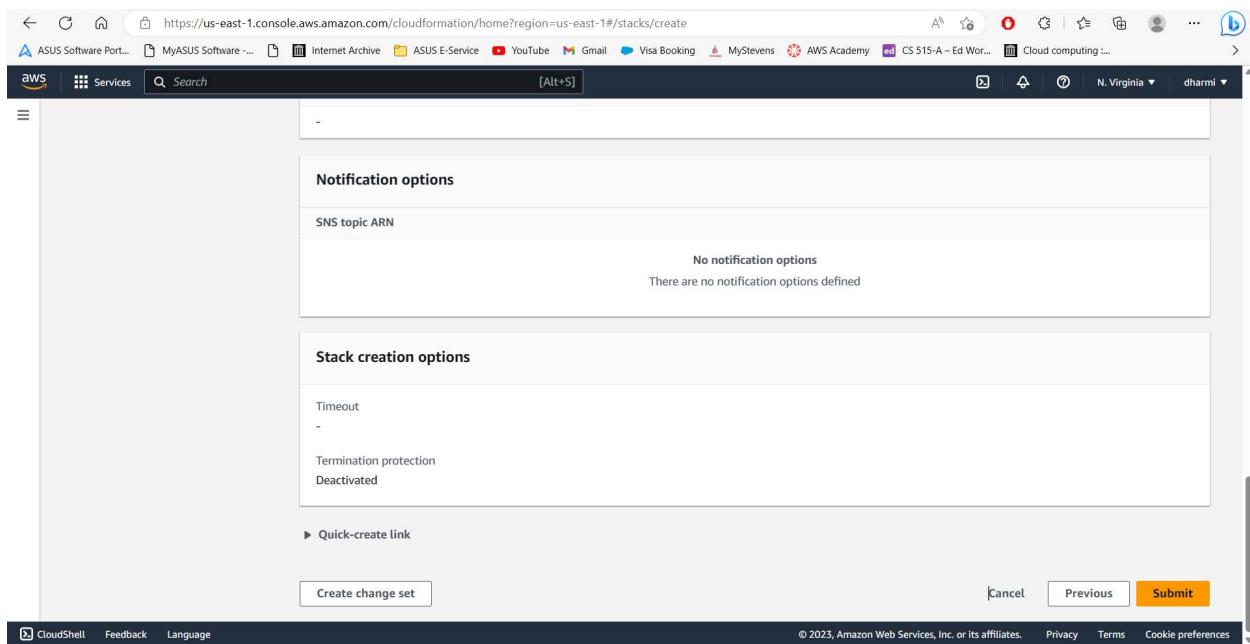
▶ Stack policy
Defines the resources that you want to protect from unintentional updates during a stack update.

▶ Rollback configuration
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back.

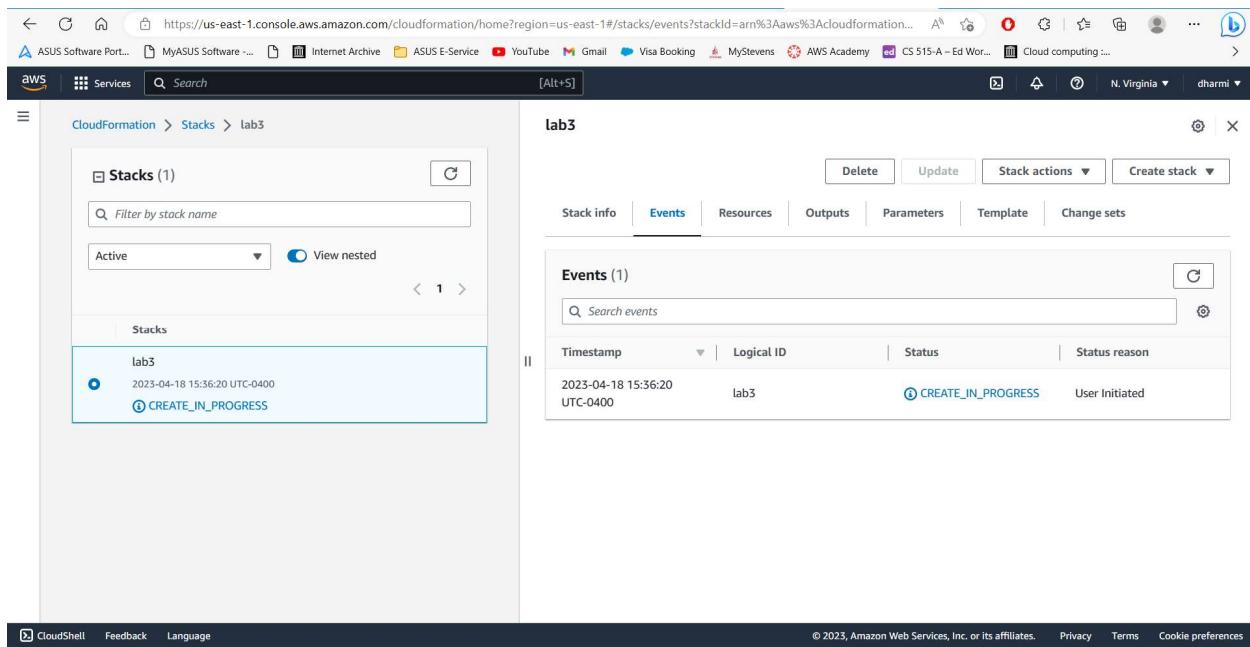
▶ Notification options

▶ Stack creation options

Cancel Previous Next



- After click on submit, the stack is created successfully.



The screenshot shows the AWS CloudFormation console with the URL <https://us-east-1.console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringText=&filteringStatus=active&viewNeste...>. The left sidebar includes sections for Stacks, StackSets, Exports, Designer, Registry (Public extensions, Activated extensions, Publisher), Spotlight, Feedback, CloudShell, Feedback, and Language. The main area displays a table titled 'Stacks (1)' with one entry: 'lab3' (Status: CREATE_COMPLETE, Created time: 2023-04-18 15:36:20 UTC-0400). A detailed description of the stack is provided, stating it's an AWS CloudFormation Sample Template for EC2InstanceWithSecurityGroupSample. The description notes that it creates an Amazon EC2 instance running the Amazon Linux AMI, chosen based on the region, and includes a security group for SSH access. It also mentions billing for AWS resources if a stack is created from this template.

- When the stack is successfully created, an instance is also created. Checking in ec2 that an instance is also created.

The screenshot shows the AWS EC2 Instances page with the URL [\\$case=true%5C;client=false;\\$regex=tags:false%5C...">https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances?v=3">\\$case=true%5C;client=false;\\$regex=tags:false%5C.... The left sidebar includes sections for New EC2 Experience \(Tell us what you think\), EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances \(Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations\), and Images \(AMIs, AMI Catalog\). The main area displays a table titled 'Instances \(6\) Info' with six entries: Server2, Load balancer, Server4, Server3, Server1, and another unnamed instance. All instances are listed as 'Running' in the 'Instance state' column, with instance IDs like i-07f35ed77f7586f8d, i-062bb33c0b1627306, etc. The table also includes columns for Name, Instance ID, Instance type \(t2.micro\), Status check, Alarm status, Availability Zone, and Public IPv4 DNS.](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances?v=3)

- Stack create complete.

The screenshot shows the AWS CloudFormation console with the 'lab3' stack selected. The left sidebar shows navigation options like 'Stack details', 'Drifts', 'StackSets', 'Exports', 'Designer', 'Registry', 'Spotlight', and 'Feedback'. The main area displays the 'Stacks (1)' section with a table showing the stack 'lab3' created on 2023-04-18 at 15:36:20 UTC-0400. The status is 'CREATE_COMPLETE'. The 'Overview' tab is selected, showing the stack ID (arn:aws:cloudformation:us-east-1:129104681719:stack/lab3/4b7286f0-de20-11ed-b95d-0a3eb8eb5335), description (AWS CloudFormation Sample Template EC2InstanceWithSecurityGroupSample: Create an Amazon EC2 instance running the Amazon Linux AMI. The AMI is chosen based on the region in which the stack is run. This example creates an EC2 security group for the instance to give you SSH access. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.), and other details like status, root stack, and created time.

- Now, editing the template file for load balancer. Edit keyname, instance type and set minsize and maxsize to 4. Save this file as .json

The screenshot shows the 'Prerequisite - Prepare template' step of a stack update. On the left, a sidebar lists steps: Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review lab3). The main area has two tabs: 'Prerequisite - Prepare template' and 'Specify template'. Under 'Prerequisite - Prepare template', it says 'Prepare template: Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.' There are three radio buttons: 'Use current template' (selected), 'Replace current template', and 'Edit template in designer'. Under 'Specify template', it says 'A template is a JSON or YAML file that describes your stack's resources and properties.' It has sections for 'Template source' (Amazon S3 URL selected) and 'Upload a template file' (choose file 'loadbalancer (1)'). Below that, it shows the S3 URL: https://s3.us-east-1.amazonaws.com/cf-templates-p6bff7wpgey-us-east-1/2023-04-18T193849.1262kmv-loadbalancer(1). There are 'View in Designer', 'Cancel', and 'Next' buttons at the bottom.

- Now, editing the template file for load balancer. Edit keyname, instance type and set minsize and maxsize to 4. Save this file as .json and saved as loabalancer.

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Use current template Replace current template Edit template in designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

Upload a template file

loadbalancer (1)

JSON or YAML formatted file

S3 URL: https://s3.us-east-1.amazonaws.com/cf-templates-p6bff7jwpgey-us-east-1/2023-04-18T193849.126Zkmv-loadbalancer(1)

Cancel

- Click next and select all subnets in parameters.

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

InstanceType

WebServer EC2 Instance type

t2.micro

KeyNames

Q

subnet-0c6daaba7ec86b296 (172.31.48.0/20)
 subnet-06b7f7b07958065fc (172.31.80.0/20)
 subnet-0ac560e7a1d49b8d2 (172.31.16.0/20)
 subnet-0291870d0b2506ab4 (172.31.32.0/20)
 subnet-04e79865e0730ee07 (172.31.64.0/20)
 subnet-011c426855187a662 (172.31.0.0/20)

Select List<AWS::EC2::Subnet::Id>

subnet-011c426855187a662 X subnet-04e79865e0730ee07 X subnet-0291870d0b2506ab4 X subnet-0ac560e7a1d49b8d2 X
subnet-06b7f7b07958065fc X subnet-0c6daaba7ec86b296 X

VpcId

VpcId of your existing Virtual Private Cloud (VPC)

Dharmi

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

- Go to the next.

Step 3
Configure stack options

InstanceType
WebServer EC2 instance type
t2.micro

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instances
Dharmi

SSHLocation
The IP address range that can be used to SSH to the EC2 instances
0.0.0.0/0

Subnets
The list of Subnetids in your Virtual Private Cloud (VPC)
Select List<AWS::EC2::Subnet::Id>

VpcId
VpcId of your existing Virtual Private Cloud (VPC)
vpc-08dfc40c1a3d7ad54

Cancel **Previous** **Next**

- Again go to next.

Notification options

SNS topic ARN

No notification options
There are no notification options defined

Change set preview

Changes (7)

Action	Logical ID	Physical ID	Resource type	Replacement	Module
Add	ALBListener	-	AWS::ElasticLoadBal...	-	-
Add	ALBTargetGroup	-	AWS::ElasticLoadBal...	-	-
Add	ApplicationLoadBalancer	-	AWS::ElasticLoadBal...	-	-
Remove	EC2Instance	i-0e3c0587581a6cd8b...	AWS::EC2::Instance	-	-

CloudShell **Feedback** **Language** © 2023, Amazon Web Services, Inc. or its affiliates. **Privacy** **Terms** **Cookie preferences**

- Now, the update of stack is completed.

Timestamp	Logical ID	Status	Status reason
2023-04-18 15:42:41 UTC-0400	lab3	UPDATE_COMPLETE	-
2023-04-18 15:42:41 UTC-0400	InstanceSecurityGroup	DELETE_COMPLETE	-
2023-04-18 15:42:40 UTC-0400	InstanceSecurityGroup	DELETE_IN_PROGRESS	-
2023-04-18 15:42:39 UTC-0400	EC2Instance	DELETE_COMPLETE	-
2023-04-18 15:42:09 UTC-0400	EC2Instance	DELETE_IN_PROGRESS	-
2023-04-18 15:42:08 UTC-0400	lab3	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	-
2023-04-18 15:42:06			

- All instances created successfully.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Server2	i-07f55ed77f7586f8d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-52-91-114-1
Load balancer	i-062bb53c0b1627306	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-54-224-224-
Server4	i-0d338b05be58401ee	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-44-203-91-7
Server3	i-086e4fc61ee0c87c3	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-52-70-113-1
Server1	i-011af6f321589120d	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-94-128-13
-	i-0e3c0587581a6cd8b	Running	t2.micro	Initializing	No alarms	us-east-1c	ec2-18-234-175-

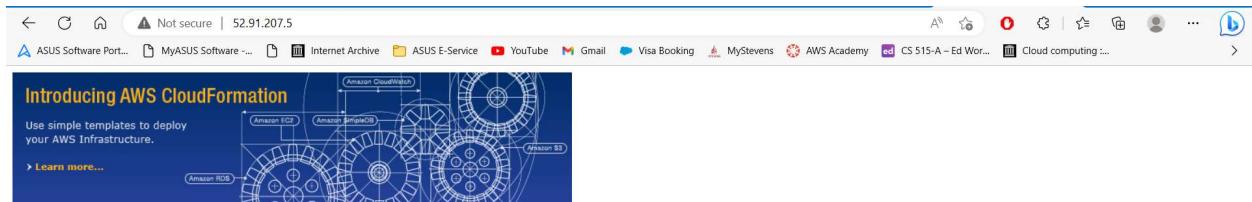
- Go to the stack and in outputs, access to the url value.

The screenshot shows the AWS CloudFormation Outputs page. On the left, there's a sidebar with 'Stacks (1)' and a list containing 'lab3'. The main area is titled 'lab3' and shows the 'Outputs (1)' tab selected. It displays one output named 'URL' with the value: <http://lab3-Applic-804H4L8GTVDI-1157606357.us-east-1.elb.amazonaws.com>. Other tabs like 'Stack info', 'Events', 'Resources', 'Parameters', 'Template', and 'Change sets' are also visible.

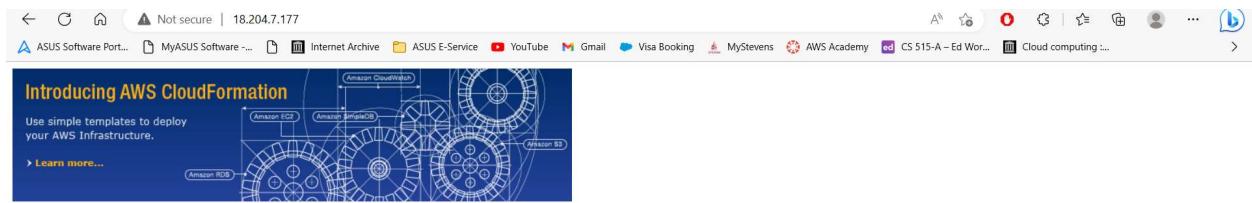
- Now, the update of stack is completed.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes 'New EC2 Experience' (selected), 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (selected), 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'Images', 'AMIs', and 'AMI Catalog'. The main area shows a table titled 'Instances (10) Info' with columns: Public IPv4, Elastic IP, IPv6 IPs, Monitoring, Security group name, Key name, and Launch time. The table lists 10 instances, all associated with 'lab3-InstanceSecurityGr...' security group, 'Dharmi' key pair, and launched on 2023/04/18 15:40 GMT-4. A modal window titled 'Select an instance' is open at the bottom.

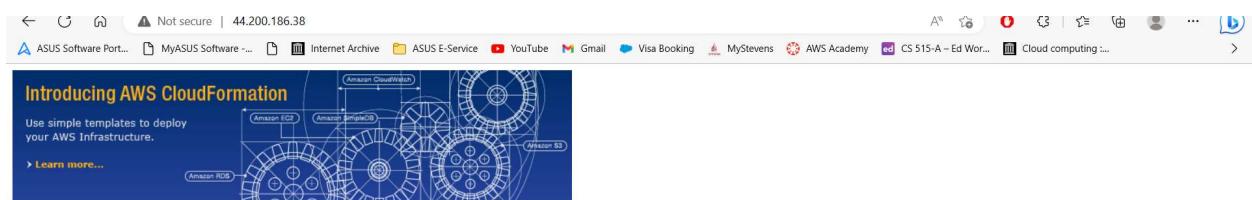
- Now, access the IP address of all 4 instances and got the output.



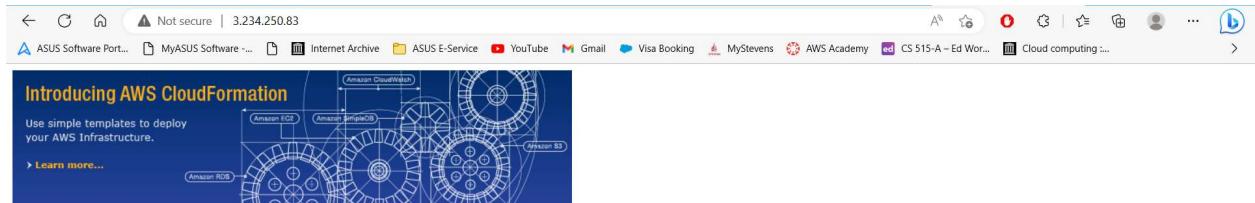
Congratulations, you have successfully launched the AWS CloudFormation sample.



Congratulations, you have successfully launched the AWS CloudFormation sample.



Congratulations, you have successfully launched the AWS CloudFormation sample.



Congratulations, you have successfully launched the AWS CloudFormation sample.

- Then, go to the load balancer and it finds server here as a load balancer.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for EC2 Dashboard, Global View, Events, Tags, Limits, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations), and Images (AMIs, AMI Catalog). The main content area displays a table titled "Instances (10) Info" with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
-	i-04372f94c30b87244	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-52-91-207-5
-	i-0f3f1bb62d7ee5c59	Running	t2.micro	2/2 checks passed	No alarms	us-east-1e	ec2-18-204-7-17
-	i-07a2cb70k0e090e9f	Running	t2.micro	2/2 checks passed	No alarms	us-east-1f	ec2-44-200-186-
-	i-00bd7f51d74cc1f14	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-3-234-250-8

EC2 > Load balancers

Load balancers (1)

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

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
lab3-Applic-8Q4H4L8GTVDI	lab3-Applic-8Q4H4L8GT... 357.us-east-1.elb.amazonaws.com (A Record)	Active	vpc-08dfc40c1a5d7ad34	6 Availability Zones	application	April 1 (UTC)

0 load balancers selected

Select a load balancer above.

- Then, access the DNS name to command prompt.

Load balancer: lab3-Applic-8Q4H4L8GTVDI

Details

arn:aws:elasticloadbalancing:us-east-1:129104681719:loadbalancer/app/lab3-Applic-8Q4H4L8GTVDI/93eb789723e64466

DNS name copied

Load balancer type	Status	VPC
Application	Active	vpc-08dfc40c1a5d7ad34
IP address type	Availability Zones	Hosted zone
IPv4	subnet-0c6daaba7ec86b296, us-east-1, us-east-1	Z35SXDOTRQ7X7K
Scheme		
Internet-facing		

- Do this process for 3 instances and follow these codes in all.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\skm18> cd Downloads
PS C:\Users\skm18\Downloads> icacls.exe Dharmi.pem
Dharmi.pem LAPTOP-S8RUQON8\admin:(F)

Successfully processed 1 files; Failed processing 0 files
PS C:\Users\skm18\Downloads> ssh -i "Dharmi.pem" ec2-user@ec2-52-91-207-5.compute-1.amazonaws.com
The authenticity of host 'ec2-52-91-207-5.compute-1.amazonaws.com (52.91.207.5)' can't be established.
ED25519 key fingerprint is SHA256:Psay4S5vwmfYaXr5oLn2dPZGXHBE0p7MRWZGylwmaAQM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-91-207-5.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

__| __|_
_| ( /   Amazon Linux AMI
___|\_\_|___|_

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
30 package(s) needed for security, out of 46 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-83-232 ~]$ sudo yum install nginx
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                         | 2.1 kB  00:00:00
amzn-updates                                     | 3.8 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package nginx.x86_64 1:1.18.0-1.44.amzn1 will be installed
--> Processing Dependency: libprofiler.so.0()(64bit) for package: 1:nginx-1.18.0-1.44.amzn1.x86_64
--> Running transaction check
--> Package gperftools-libs.x86_64 0:2.0-11.5.amzn1 will be installed
--> Processing Dependency: libunwind.so.8()(64bit) for package: gperftools-libs-2.0-11.5.amzn1.x86_64
--> Running transaction check
--> Package libunwind.x86_64 0:1.1-10.8.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

```

```

---> Package libunwind.x86_64 0:1.1-10.8.amzn1 will be installed
---> Finished Dependency Resolution

Dependencies Resolved

=====
Version          Repository      Size
=====
nginx           x86_64        1:1.18.0-1.44.amzn1    amzn-updates   604 k
Installing:
=====
nginx           x86_64        1:1.18.0-1.44.amzn1    amzn-updates   604 k
Installing for dependencies:
gperftools-libs x86_64        2.0-11.5.amzn1       amzn-main      570 k
libunwind        x86_64        1.1-10.8.amzn1       amzn-main      72 k

Transaction Summary
=====
Install  1 Package (+2 Dependent packages)

Total download size: 1.2 M
Installed size: 3.0 M
Is this ok [y/d/N]: y
Downloading packages:
(1/3): libunwind-1.1-10.8.amzn1.x86_64.rpm          | 72 kB  00:00:00
(2/3): nginx-1.18.0-1.44.amzn1.x86_64.rpm          | 604 kB  00:00:00
(3/3): gperftools-libs-2.0-11.5.amzn1.x86_64.rpm    | 570 kB  00:00:00
Total
-----Total
6.2 MB/s | 1.2 MB  00:00:00

Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libunwind-1.1-10.8.amzn1.x86_64          1/3
  Installing : gperftools-libs-2.0-11.5.amzn1.x86_64    2/3
  Installing : 1:nginx-1.18.0-1.44.amzn1.x86_64        3/3
  Verifying  : libunwind-1.1-10.8.amzn1.x86_64          1/3
  Verifying  : gperftools-libs-2.0-11.5.amzn1.x86_64    2/3
  Verifying  : 1:nginx-1.18.0-1.44.amzn1.x86_64        3/3

Installed:
  nginx.x86_64 1:1.18.0-1.44.amzn1

```

```
Verifying : libunwind-1.1-10.8.amzn1.x86_64 1/3
Verifying : gperftools-libs-2.0-11.5.amzn1.x86_64 2/3
Verifying : 1:nginx-1.18.0-1.44.amzn1.x86_64 3/3

Installed:
  nginx.x86_64 1:1.18.0-1.44.amzn1

Dependency Installed:
  gperftools-libs.x86_64 0:2.0-11.5.amzn1           libunwind.x86_64 0:1.1-10.8.amzn1

Complete!
[ec2-user@ip-172-31-83-232 ~]$ sudo service nginx start
Starting nginx: nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to [:1:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to [:1:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to [:1:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to [:1:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)
nginx: [emerg] bind() to [:1:80 failed (98: Address already in use)
nginx: [emerg] still could not bind()                                [FAILED]
[ec2-user@ip-172-31-83-232 ~]$ sudo fuser -k 80/tcp
80/tcp:                         9176  9179  9180  9181  9182  9183  9184  9185  9186  22401
[ec2-user@ip-172-31-83-232 ~]$ sudo service nginx start          [ OK ]
Starting nginx:
[ec2-user@ip-172-31-83-232 ~]$ cd ..
[ec2-user@ip-172-31-83-232 home]$ cd ..
[ec2-user@ip-172-31-83-232 /]$ cd usr/share/nginx/html
[ec2-user@ip-172-31-83-232 html]$ sudo nano index.html
[ec2-user@ip-172-31-83-232 html]$ client_loop: send disconnect: Connection reset
PS C:\Users\skm18\Downloads> |
```

- change the server name in particular index file.

```
GNU nano 2.5.3                                         File: index.html

</style>
</head>
<body>
<h1>Welcome to <strong>nginx</strong> on the Amazon Linux AMI! server 3</h1>

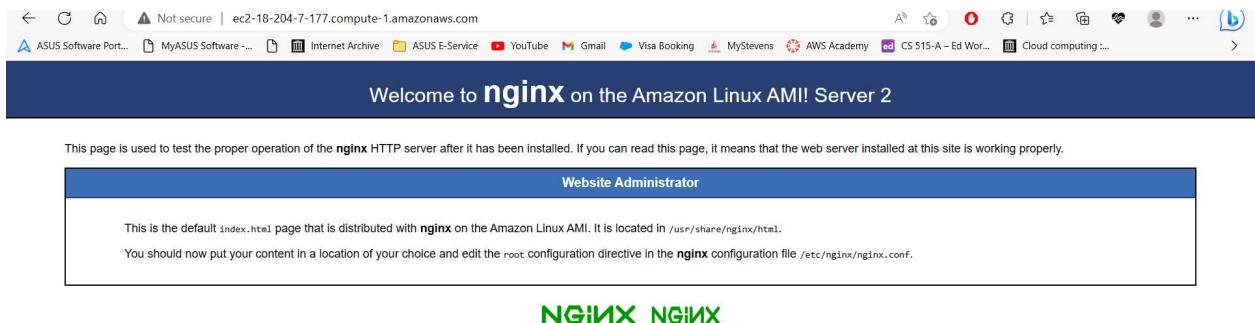
<div class="content">
  <p>This page is used to test the proper operation of the
  <strong>nginx</strong> HTTP server after it has been
  installed. If you can read this page, it means that the
  web server installed at this site is working
  properly.</p>

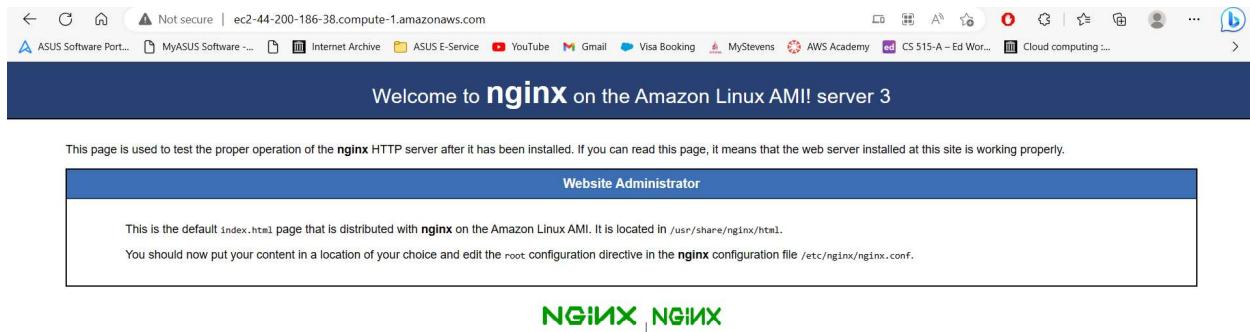
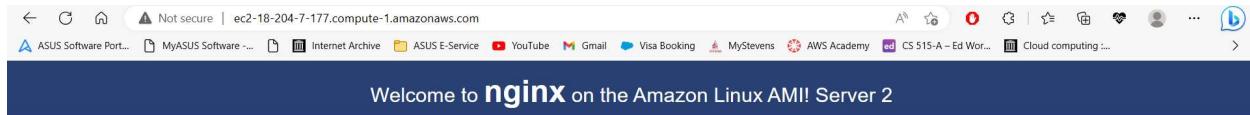
  <div class="alert">
    <h2>Website Administrator</h2>
    <div class="content">
      <p>This is the default <tt>index.html</tt> page that
      is distributed with <strong>nginx</strong> on
      the Amazon Linux AMI. It is located in
      <tt>/usr/share/nginx/html</tt>. </p>

      <p>You should now put your content in a location of
      your choice and edit the <tt>root</tt> configuration
      directive in the <strong>nginx</strong>
      configuration file
      <tt>/etc/nginx/nginx.conf</tt>.</p>
    </div>
  </div>

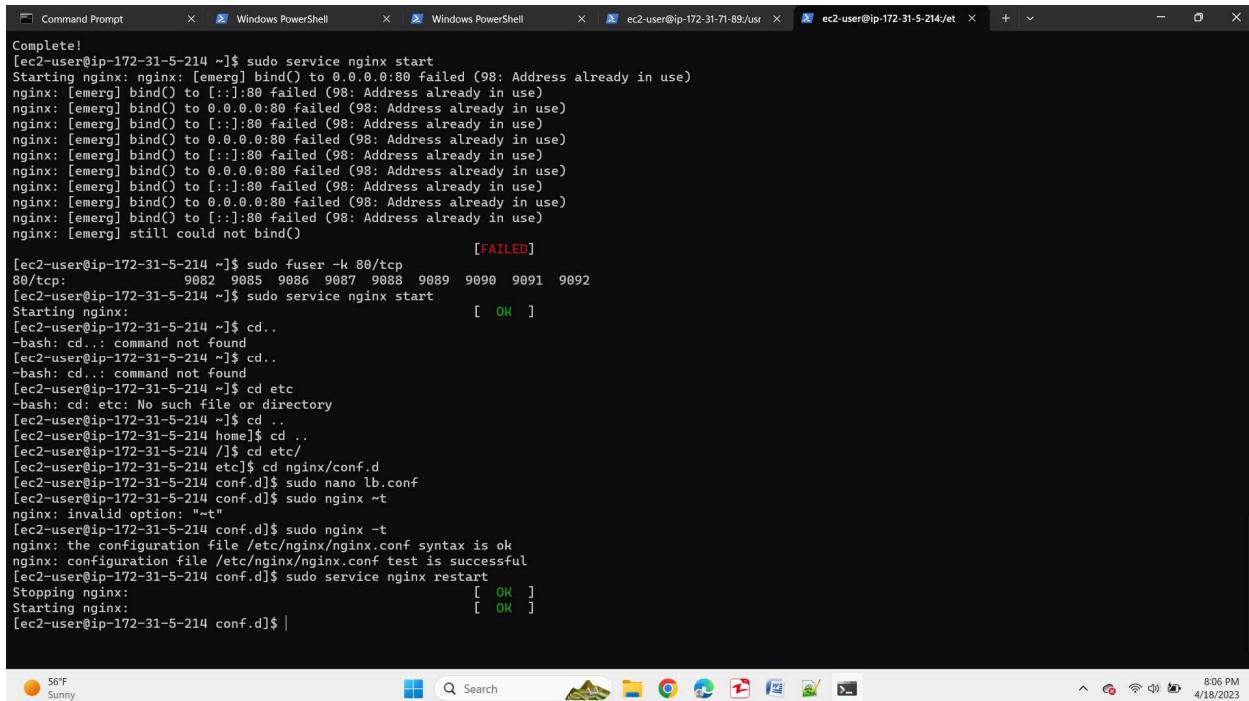
  <div class="logos">
    <a href="http://nginx.net/"></a>
  </div>
</div>
```

- Then, access the particular IP address to run every instance and it shows the output as it given here.





- Follow these steps in 4th instance that is load balancer.

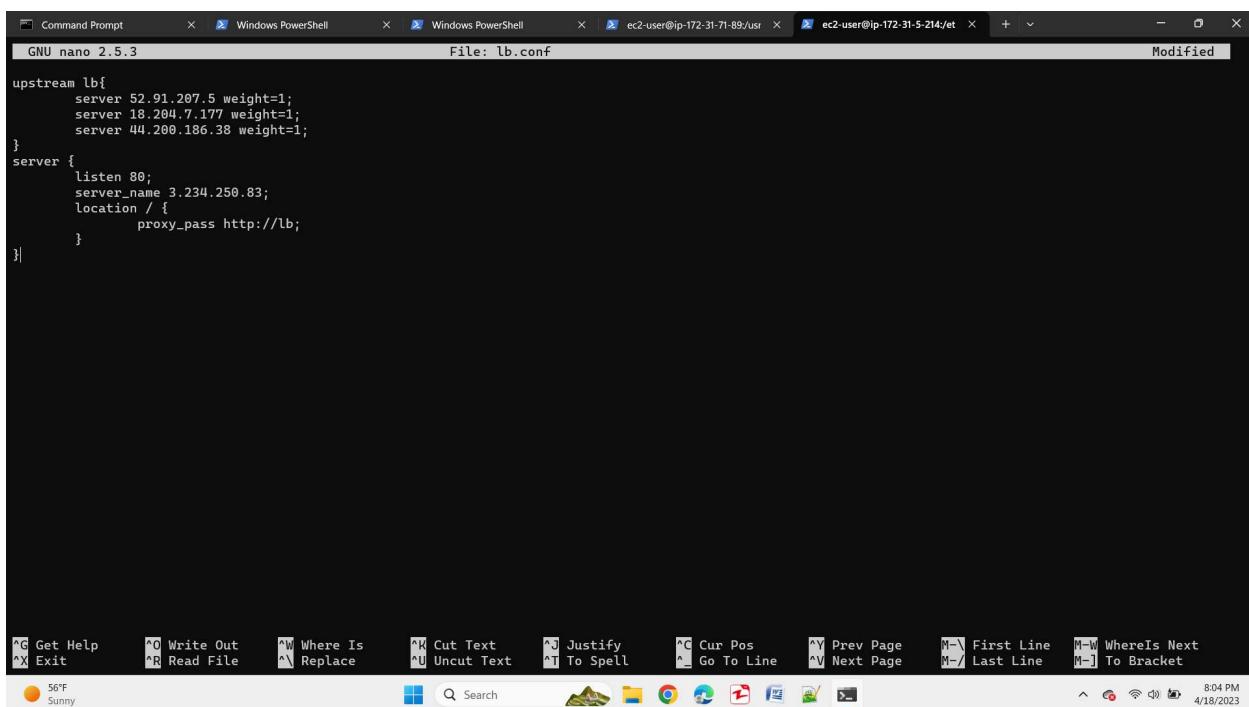


```

[ec2-user@ip-172-31-5-214 ~]$ sudo service nginx start
Starting nginx: [emerg] bind() to 0.0.0.80 failed (98: Address already in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.80 failed (98: Address already in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.80 failed (98: Address already in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.80 failed (98: Address already in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address already in use)
nginx: [emerg] bind() to 0.0.0.80 failed (98: Address already in use)
nginx: [emerg] bind() to [::]:80 failed (98: Address already in use)
nginx: [emerg] still could not bind() [FAILED]
[ec2-user@ip-172-31-5-214 ~]$ sudo fuser -k 80/tcp
80/tcp: 9082 9085 9086 9087 9088 9090 9091 9092
[ec2-user@ip-172-31-5-214 ~]$ sudo service nginx start
Starting nginx: [ OK ]
[ec2-user@ip-172-31-5-214 ~]$ cd ..
-bash: cd: command not found
[ec2-user@ip-172-31-5-214 ~]$ cd ..
-bash: cd: command not found
[ec2-user@ip-172-31-5-214 ~]$ cd etc
-bash: cd: etc: No such file or directory
[ec2-user@ip-172-31-5-214 ~]$ cd ..
[ec2-user@ip-172-31-5-214 ~]$ cd home
[ec2-user@ip-172-31-5-214 ~]$ cd etc/
[ec2-user@ip-172-31-5-214 ~]$ cd nginx/conf.d
[ec2-user@ip-172-31-5-214 conf.d]$ sudo nano lb.conf
[ec2-user@ip-172-31-5-214 conf.d]$ sudo nginx -t
nginx: invalid option: "-t"
[ec2-user@ip-172-31-5-214 conf.d]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-172-31-5-214 conf.d]$ sudo service nginx restart
Stopping nginx: [ OK ]
Starting nginx: [ OK ]
[ec2-user@ip-172-31-5-214 conf.d]$ 

```

- Add this code in config file.



```

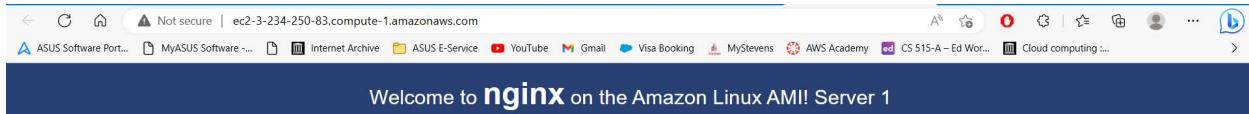
GNU nano 2.5.3
File: lb.conf
Modified

upstream lb{
    server 52.91.207.5 weight=1;
    server 18.204.7.177 weight=1;
    server 44.200.186.38 weight=1;
}
server {
    listen 80;
    server_name 3.234.250.83;
    location / {
        proxy_pass http://lb;
    }
}

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page M-\ First Line M-W WhereIs Next
 ^X Exit ^R Read File ^V Replace ^U Uncut Text ^T To Spell ^L Go To Line ^V Next Page M-/ Last Line M-J To Bracket

- It gives the output simultaneously of 3 instances as we load the address again. Here is output after reload the load balancer.



This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on the Amazon Linux AMI. It is located in `/usr/share/nginx/html`. You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX NGINX



This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on the Amazon Linux AMI. It is located in `/usr/share/nginx/html`. You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX NGINX



