

I have created my template and for that I have added the below attributes.

Parameters

We can define the parameters which is required for the resources for reference.

KeyName – Providing the existing keypair which I have created in ec2 console and I will refer this keypair for all my instances

InstanceType – I am creating t2.micro instance because that comes under free tier.

SSHL defense – I have provide my ip address which I will refer for port 22.

```
  "AWSTemplateFormatVersion" : "2010-09-09",
  "Description" : "AWS CloudFormation sample template",
  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instance",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "Default": "lab3keypair",
      "ConstraintDescription" : "Can contain only ASCII characters."
    },
    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.micro",
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },
    "SSHL defense" : {
      "Description" : "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "173.70.217.190/32",
      "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,2})",
      "ConstraintDescription": "Must be a valid IP CIDR range of the form x.x.x.x"
    }
  },
}
```

Resources

I am creating 3 instances, one Load balancer and two server Server 1 and Server 2 and providing ImageID from aws console and for instance Id and keyName, I am using which I have defined in properties.

And two security groups I have created one for load balancer with inbound and outbound rules and one for all the servers with inbound rules

```

"Resources" : {

    "LoadBalancer": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId" : "ami-0742b4e673072066f",
            "InstanceType" : { "Ref" : "InstanceType" },
            "SecurityGroupIds" : [ { "Ref" : "LoadBalancerSecurityGroup" } ],
            "KeyName" : { "Ref" : "KeyName" }
        }
    },

    "Server1": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId" : "ami-0742b4e673072066f",
            "InstanceType" : { "Ref" : "InstanceType" },
            "SecurityGroupIds" : [ { "Ref" : "WebServerSecurityGroup" } ],
            "KeyName" : { "Ref" : "KeyName" }
        }
    },

    "Server2": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
            "ImageId" : "ami-0742b4e673072066f",
            "InstanceType" : { "Ref" : "InstanceType" },
            "SecurityGroupIds" : [ { "Ref" : "WebServerSecurityGroup" } ],
            "KeyName" : { "Ref" : "KeyName" }
        }
    }
},
}

```

Here I have defined two security groups with the Type AWS::EC2::SecurityGroup. And also building attribute for output where my website will be the public DNS name of the load balancer.

```

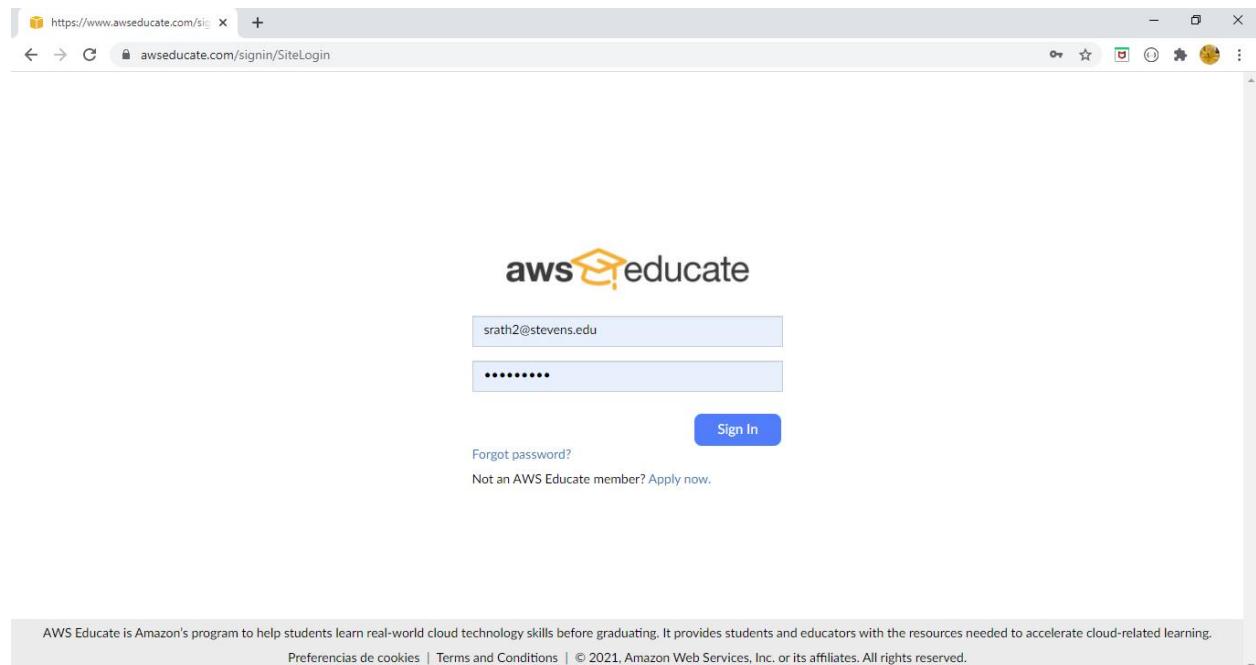
"LoadBalancerSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80",
        "SecurityGroupIngress" : [
            {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.0.0/0"},
            {"IpProtocol" : "tcp", "FromPort" : "443", "ToPort" : "443", "CidrIp" : "0.0.0.0/0"},
            {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : { "Ref" : "SSHLocation"}}
        ],
        "SecurityGroupEgress" : [
            {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.0.0/0"},
            {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "443", "CidrIp" : "0.0.0.0/0"}
        ],
        "VpcId" : "vpc-2e883e53"
    }
},

"WebServerSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Enable HTTP access via port 80",
        "SecurityGroupIngress" : [
            {"IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.0.0/0"},
            {"IpProtocol" : "tcp", "FromPort" : "443", "ToPort" : "443", "CidrIp" : "0.0.0.0/0"},
            {"IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : { "Ref" : "SSHLocation"}}
        ]
    }
},
}

"Outputs" : {
    "WebsiteURL" : {
        "Description" : "URL for newly created LAMP stack",
        "Value" : { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [ "LoadBalancer", "PublicDnsName" ] } ] ] }
    }
}

```

Login into aws management console to create the stack with the template.



A screenshot of a web browser showing the AWS Educate student dashboard. The URL in the address bar is https://awseducate.com/student/s/. The page has a header with the AWS Educate logo and navigation links for "My Classrooms", "Portfolio", "Career Pathways", "Badges", "Jobs", and "Logout". On the left, a sidebar displays a profile picture of "Smita Rath", her name, and stats: "Consecutive Days: 1", "Pathways Completed: 0", and "Badges Earned: 0". It also includes a message about the prevalence of cloud technology and a link to begin the journey. The main content area features a section titled "Search Thousands of Cloud Jobs and Internships on the AWS Educate Job Board" with a "Learn More" link. To the right, a "Suggested Jobs" section lists "Associate Consultant" at "Slalom" and "Deloitte Consulting Solutions Engineering Analyst" at "Deloitte Consulting", each with a "more about this opportunity" link. A "See More" button is at the bottom of this section. A vertical scrollbar is visible on the right side of the dashboard.

The screenshot shows the AWS Educate My Classrooms interface. At the top, there's a header with the AWS Educate logo, user profile (Smita Rath), and navigation links (My Classrooms, Portfolio, Career Pathways, Badges, Jobs, Logout). Below the header, a section titled "My Classrooms" displays a course invitation for "Introduction to Cloud Computing". The course details are: Description: "Full introduction to Cloud technologies and economics with the lab work on AWS.", Educator: Igor Faynberg, Course End Date: 05/19/2021, Credit Allocated Per Student: \$100. The status is "Accepted" with a "Go to classroom" button.

The screenshot shows the Vocareum Workbench interface. At the top, there's a header with the Vocareum logo, user profile (srath2@stevens.edu), and navigation links (My Classes, Help). Below the header, a section titled "Your AWS Account Status" displays account information: Active (full access), \$98.94 remaining credits (estimated), and 2:60 session time. There are "Account Details" and "AWS Console" buttons at the bottom. A note at the bottom right encourages responsible usage of the AWS Educate Account.

To create stack, I clicked on cloud formation service.

The screenshot shows the AWS Management Console homepage. The left sidebar lists recently visited services (CloudFormation, IAM, Simple Queue Service, EC2, Billing, VPC, Security Hub) and all services (Compute, EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application, Quantum Technologies, Amazon Braket, Management & Governance, AWS Organizations, Security, Identity, & Compliance, IAM, Resource Access Manager, Cognito, Secrets Manager). On the right, there are promotional boxes for the AWS Console Mobile App and Modernize Your APIs with GraphQL, along with a link to S3 Object Lambda. The URL in the address bar is https://console.aws.amazon.com/cloudformation/home?region=us-east-1#.

Under CloudFormation, click on create stack

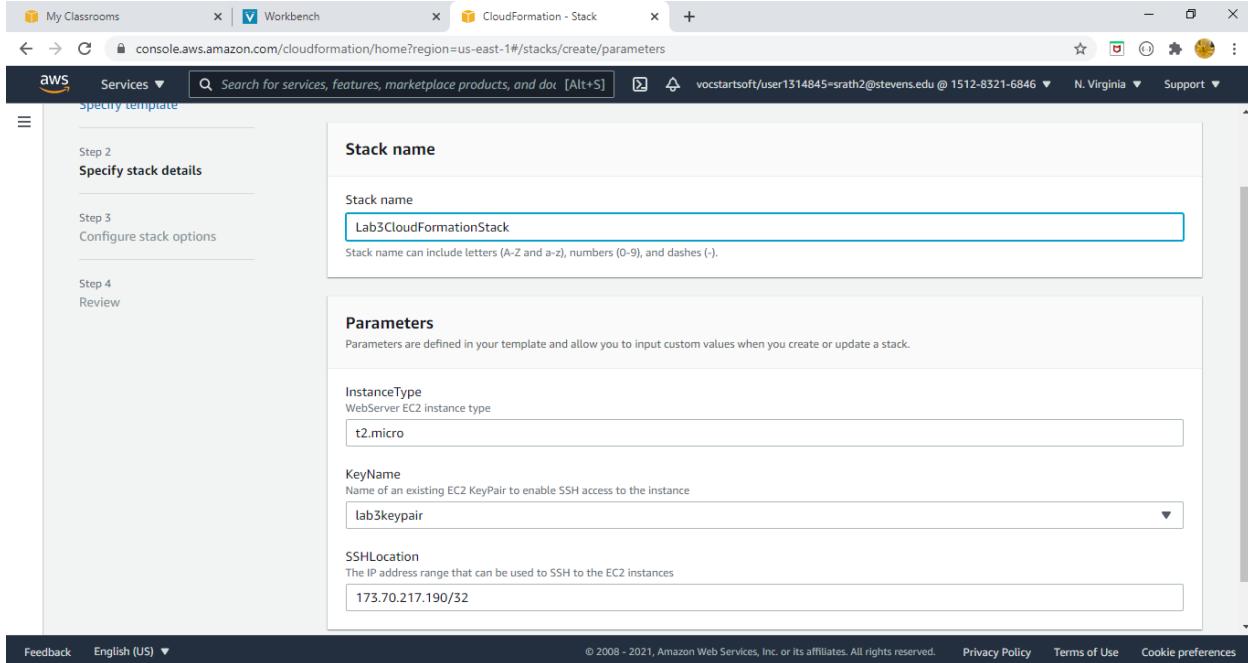
The screenshot shows the AWS CloudFormation homepage. It features a main banner with the text "Model and provision all your cloud infrastructure" and a sub-banner below it stating "Simplify Your Infrastructure Management Using AWS". A "Create a CloudFormation stack" button is prominently displayed. To the right, there is a "Getting started" sidebar with links to "What is AWS CloudFormation", "Getting started with CloudFormation", "Learn template basics", and "Quick starts". The URL in the address bar is https://console.aws.amazon.com/cloudformation/home?region=us-east-1#.

On this page either a sample template can be used, existing template can be uploaded, I am uploading my template which I have created.

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The left sidebar lists steps: Step 1 (Specify template, currently selected), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main panel is titled 'Prerequisite - Prepare template'. It contains a section 'Prepare template' with the note: '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.' Below this are three radio buttons: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The next section is 'Specify template', which explains that a template is a JSON or YAML file describing stack resources. It includes a 'Template source' section with 'Amazon S3 URL' selected and a text input field containing 'https://'. At the bottom right of the main panel are 'Cancel' and 'Next' buttons.

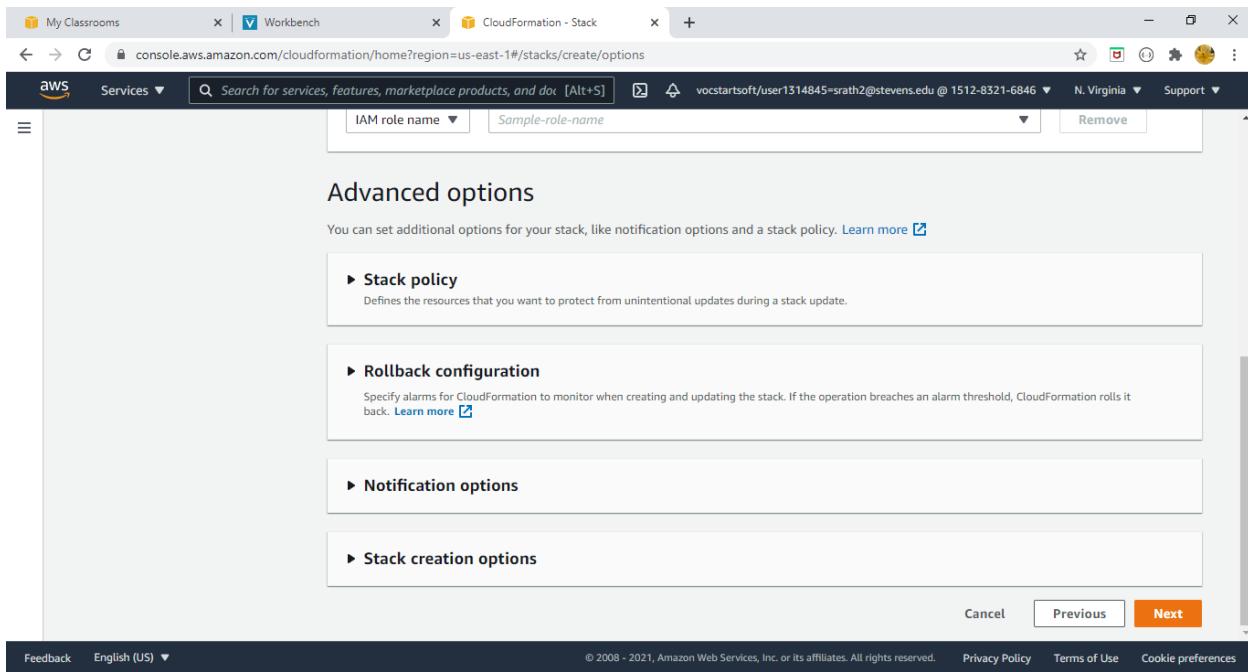
This screenshot shows the same 'Create stack' wizard as the previous one, but with a different configuration. In the 'Template source' section, the 'Upload a template file' radio button is selected instead of 'Amazon S3 URL'. Below it, there is a 'Choose file' button with the path 'lab3cloudFormation.json' and a note 'JSON or YAML formatted file'. At the bottom right of the main panel are 'Cancel' and 'Next' buttons.

On the next page, provide the stack name and it will display the parameters which are defined in the template.



The screenshot shows the AWS CloudFormation 'Specify stack details' step. The 'Stack name' field contains 'Lab3CloudFormationStack'. The 'Parameters' section includes 'InstanceType' set to 'WebServer EC2 instance type t2.micro', 'KeyName' set to 'lab3keypair', and 'SSHLocation' set to '173.70.217.190/32'.

Click on next



The screenshot shows the AWS CloudFormation 'Advanced options' step. It includes sections for 'Stack policy' (described as defining resources to protect from unintentional updates), 'Rollback configuration' (described as specifying alarms for monitoring stack creation and updating), 'Notification options' (described as setting up notifications for stack events), and 'Stack creation options'.

On the next page stack details can be reviewed here.

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The current step is 'Step 1: Specify template'. The left sidebar shows navigation steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main area displays the template configuration:

- Template**:
 - Template URL: <https://s3-external-1.amazonaws.com/cf-templates-13edpgj2uquad-us-east-1/2021103cyn-lab3cloudFormation.json>
 - Stack description: AWS CloudFormation sample template
 - [Estimate cost](#)

The screenshot shows the AWS CloudFormation 'Create stack' wizard. The current step is 'Step 2: Specify stack details'. The left sidebar shows navigation steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main area displays the stack details configuration:

- Parameters (3)**:

Key	Value
InstanceType	t2.micro
KeyName	lab3keypair
SSHLocation	173.70.217.190/32

For the final step to create the template, click on create stack.

The screenshot shows the 'CloudFormation - Stack' creation wizard. The top section displays a message: 'No notification options' with the subtext 'There are no notification options defined'. Below this is a 'Stack creation options' section containing the following settings:

- Rollback on failure: Enabled
- Timeout: -
- Termination protection: Disabled

At the bottom of the wizard are buttons for 'Quick-create link', 'Cancel', 'Previous', 'Create change set', and a prominent orange 'Create stack' button.

Once stack creation is un process, under events tab the different events are visible from creation to completion of all the resources which are defined in template.

The screenshot shows the 'CloudFormation - Stacks' page for the 'Lab3CloudFormationStack'. The left sidebar lists 'Stacks (1)'. The main area is titled 'Lab3CloudFormationStack' and shows the 'Events' tab selected. The 'Events' table contains one entry:

Timestamp	Logical ID	Status	Status reason
2021-04-12 22:38:09 UTC-0400	Lab3CloudFormationStack	CREATE_IN_PROGRESS	User Initiated

The screenshot shows the AWS CloudFormation console with the 'Events' tab selected for the 'Lab3CloudFormationStack'. The stack was created on 2021-04-12 at 22:38:09 UTC-0400 and is currently in the 'CREATE_IN_PROGRESS' state. The events listed are:

Timestamp	Logical ID	Status	Status reason
2021-04-12 22:38:14 UTC-0400	LoadBalancerSecurityGroup	CREATE_IN_PROGRESS	-
2021-04-12 22:38:14 UTC-0400	WebServerSecurityGroup	CREATE_IN_PROGRESS	-
2021-04-12 22:38:09 UTC-0400	Lab3CloudFormationStack	CREATE_IN_PROGRESS	User Initiated

The screenshot shows the AWS CloudFormation console with the 'Resources' tab selected for the 'Lab3CloudFormationStack'. The stack was created on 2021-04-12 at 22:52:29 UTC-0400 and is currently in the 'CREATE_IN_PROGRESS' state. The resources listed are:

Timestamp	Logical ID	Status	Status reason
2021-04-12 22:52:45 UTC-0400	LoadBalancer	CREATE_IN_PROGRESS	Resource creation initiated
2021-04-12 22:52:44 UTC-0400	Server1	CREATE_IN_PROGRESS	Resource creation initiated
2021-04-12 22:52:43 UTC-0400	Server2	CREATE_IN_PROGRESS	Resource creation initiated
2021-04-12 22:52:43 UTC-0400	LoadBalancer	CREATE_IN_PROGRESS	-
2021-04-12 22:52:42 UTC-0400	Server1	CREATE_IN_PROGRESS	-

Under Resources I can see two security groups and two servers and load balancer is created.

The screenshot shows the AWS CloudFormation console with the following details:

- Stacks (1)**: Lab3CloudFormationStack (Active, 1 resource)
- Resources (5)**:

Logical ID	Physical ID	Type	Status
LoadBalancer	i-03a113234385edb54	AWS::EC2::Instance	CREATE_COMPLETE
LoadBalancerSecurityGroup	sg-09cf812e1427a2f6b	AWS::EC2::SecurityGroup	CREATE_COMPLETE
Server1	i-0bd6e7be5119c3fd	AWS::EC2::Instance	CREATE_COMPLETE
Server2	i-007afcc13194bc009	AWS::EC2::Instance	CREATE_COMPLETE
WebServerSecurityGroup	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LSN	AWS::EC2::SecurityGroup	CREATE_COMPLETE

Now if I go and check the created security group I can check the inbound and outbound rules which I have defined in the template.

The screenshot shows the AWS EC2 Management Console with the following details:

- Details** for security group **sg-09cf812e1427a2f6b - Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX**
- Owner**: 151283216846
- VPC ID**: vpc-2e883e53
- Inbound rules** (selected):

Source IP Range	Port Range	Protocol	Action
0.0.0.0/0	80	TCP	Allow
0.0.0.0/0	443	TCP	Allow
0.0.0.0/0	22	TCP	Allow
- Outbound rules**: 2 permission entries

My Classrooms | Workbench | CloudFormation - Stack Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX | EC2 Management Console | + | - | X

console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-09cf812e1427a2f6b

aws Services ▾

Search for services, features, marketplace products, and documentation [Alt+S]

vocestartsoft/user1314845@srauth2@stevens.edu @ 1512-8321-6846 N. Virginia Support ▾

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups New

Elastic IPs New

Placement Groups

Key Pairs

Network Interfaces

Feedback English (US) ▾

Details

Security group name	sg-09cf812e1427a2f6b	Description	VPC ID
Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX	Enable HTTP access via port 80	vpc-2e883e53	
Owner	Inbound rules count	Outbound rules count	
151283216846	3 Permission entries	2 Permission entries	

Inbound rules Outbound rules Tags

Inbound rules (3)

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
SSH	TCP	22	173.70.217.190/32	-
HTTPS	TCP	443	0.0.0.0/0	-

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

My Classrooms | Workbench | CloudFormation - Stack Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX | EC2 Management Console | + | - | X

console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-09cf812e1427a2f6b

aws Services ▾

Search for services, features, marketplace products, and documentation [Alt+S]

vocestartsoft/user1314845@srauth2@stevens.edu @ 1512-8321-6846 N. Virginia Support ▾

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups New

Elastic IPs New

Placement Groups

Key Pairs

Network Interfaces

Feedback English (US) ▾

Details

Security group name	sg-09cf812e1427a2f6b	Description	VPC ID
Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX	Enable HTTP access via port 80	vpc-2e883e53	
Owner	Inbound rules count	Outbound rules count	
151283216846	3 Permission entries	2 Permission entries	

Inbound rules Outbound rules Tags

Outbound rules (2)

Type	Protocol	Port range	Destination	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
Custom TCP	TCP	80 - 443	0.0.0.0/0	-

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

Also the webserver security group details I am verifying here.

My Classrooms Workbench CloudFormation - Stack Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN EC2 Management Console

console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-0e82d1d3bb8f3eaf3

aws Services ▾

EC2 > Security Groups > sg-0e82d1d3bb8f3eaf3 - Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN

sg-0e82d1d3bb8f3eaf3 - Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN

Actions ▾

Details			
Security group name Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN	Security group ID sg-0e82d1d3bb8f3eaf3	Description Enable HTTP access via port 80	VPC ID vpc-2e883e53
Owner 151283216846	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

This screenshot shows the AWS EC2 Security Groups details page for a specific security group. The security group is named 'sg-0e82d1d3bb8f3eaf3' and is associated with the 'Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN' VPC. It has three inbound rules (HTTP, TCP port 80, 0.0.0.0/0) and one outbound rule (SSH, TCP port 22, 173.70.217.190/32). The 'Outbound rules' tab is selected.

My Classrooms Workbench CloudFormation - Stack Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN EC2 Management Console

console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-0e82d1d3bb8f3eaf3

aws Services ▾

sg-0e82d1d3bb8f3eaf3 - Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN

Details

Details			
Security group name Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN	Security group ID sg-0e82d1d3bb8f3eaf3	Description Enable HTTP access via port 80	VPC ID vpc-2e883e53
Owner 151283216846	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules Outbound rules Tags

Inbound rules (3)

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
SSH	TCP	22	173.70.217.190/32	-
HTTPS	TCP	443	0.0.0.0/0	-

Edit inbound rules

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

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

This screenshot shows the same security group details page as the first one, but the 'Inbound rules' tab is selected. It lists three inbound rules: one for HTTP (TCP port 80), one for SSH (TCP port 22), and one for HTTPS (TCP port 443). The 'Edit inbound rules' button is visible at the top right of the table.

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like New EC2 Experience, EC2 Dashboard, Instances (selected), and Images. The main content displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Server1	i-007afcc13194bc009	Running	t2.micro	2/2 checks passed	1/1 has +	us-east-1e
LoadBalancer	i-03a113234385edb54	Running	t2.micro	2/2 checks passed	1/1 has +	us-east-1e
Server2	i-0bd6e7be5119c3cf0	Running	t2.micro	Initializing	1/1 has +	us-east-1b

Below the table, the details for instance i-0bd6e7be5119c3cf0 (Server2) are shown:

Instance summary						
Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance ID i-0bd6e7be5119c3cf0 (Server2)	Public IPv4 address 3.88.116.196 open address	Private IPv4 addresses 172.31.83.101	Public IPv4 DNS	Private IPv4 DNS		
Instance state Running						
Instance type t2.micro						
AWS Compute Optimizer finding User: arn:aws:sts::151283216846:assumed-						

Load balancer instance under instance which got created with stack formation

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like New EC2 Experience, EC2 Dashboard, Instances (selected), and Images. The main content displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
LoadBalancer	i-03a113234385edb54	Running	t2.micro	2/2 checks passed	1/1 has +	us-east-1e

Below the table, the details for instance i-03a113234385edb54 (LoadBalancer) are shown:

Instance summary						
Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance ID i-03a113234385edb54 (LoadBalancer)	Public IPv4 address 34.202.159.151 open address	Private IPv4 addresses 172.31.57.255	Public IPv4 DNS ec2-34-202-159-151.compute-1.amazonaws.com open address	Private IPv4 DNS ip-172-31-57-255.ec2.internal	VPC ID vpc-2e883e53	Subnet ID subnet-2ac46d1b
Instance state Running						
Instance type t2.micro			Elastic IP addresses -			
AWS Compute Optimizer finding User: arn:aws:sts::151283216846:assumed-			IAM Role -			

The screenshot shows the AWS EC2 Instances page. A single instance, named "LoadBalancer" with Instance ID i-03a113234385edb54, is listed as "Running". It is a t2.micro type in us-east-1e Availability Zone. The instance has two security groups: Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX. Inbound rules allow traffic on ports 80, 22, and 443 from 0.0.0.0/0. There are no outbound rules listed.

Port range	Protocol	Source	Security groups
80	TCP	0.0.0.0/0	Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX
22	TCP	173.70.217.190/32	Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX
443	TCP	0.0.0.0/0	Lab3CloudFormationStack-LoadBalancerSecurityGroup-18G8FIMWCIERX

This screenshot is identical to the one above, showing the same instance configuration and network settings. The instance "LoadBalancer" is still running and associated with the same security groups and port ranges.

Server 1 instance details

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances with one row selected. The selected instance is named "Server1" with the ID "i-007afcc13194bc009". It is listed as "Running" with the instance type "t2.micro". The "Status check" column shows "2/2 checks passed" and the "Alarm status" column shows "1/1 has". The "Availability Zone" is "us-east-1e". Below the table, the "Instance summary" section provides detailed information about the instance's network settings, including its Public IPv4 address (100.25.28.54), Private IPv4 addresses (172.31.55.150), Public IPv4 DNS (ec2-100-25-28-54.compute-1.amazonaws.com), Private IPv4 DNS (ip-172-31-55-150.ec2.internal), VPC ID (vpc-2e883e53), and Subnet ID (subnet-2ac46d1b).

This screenshot is identical to the one above, but it also displays the security group configuration for the selected instance. The "Security groups" section shows the group "sg-0e82d1d3bb8f3eaf3" which is associated with the CloudFormation stack "Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN". The "Inbound rules" section lists three TCP rules: port 80 from 0.0.0.0/0, port 22 from 173.70.217.190/32, and port 443 from 0.0.0.0/0, all associated with the same security group.

Server 2 instance details

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances. One instance, named "Server2" with the ID "i-0bd6e7be5119c3cf", is listed and is currently "Running". The "Instance summary" section provides detailed information about the instance, including its Public IPv4 address (3.88.116.196), Public IPv4 DNS (ec2-3-88-116-196.compute-1.amazonaws.com), Instance type (t2.micro), and IAM Role (AWS Compute Optimizer finding). The instance is also associated with a VPC ID (vpc-2e883e53) and a subnet (subnet-c56000e4).

This screenshot is identical to the one above, but it highlights the security group information for the instance. It shows that the instance is associated with a security group named "sg-0e82d1d3bb8f3eaf3" (Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN). Below this, the "Inbound rules" section is expanded, showing three rules: port 80 (TCP) from 0.0.0.0/0, port 22 (TCP) from 173.70.217.190/32, and port 443 (TCP) from 0.0.0.0/0, all belonging to the same security group.

Connecting to load balancer instance to install nginx and configure nginx.conf file to distribute the weight of servers.

```
ec2-user@ip-172-31-57-255:~  
[mitali611@LAPTOP-08H7518L:/mnt/c/Users/smita$ ssh -i ~/ssh/lab3keypair.pem ec2-user@ec2-34-202-159-151.compute-1.amazonaws.com  
The authenticity of host 'ec2-34-202-159-151.compute-1.amazonaws.com (34.202.159.151)' can't be established.  
ECDSA key fingerprint is SHA256:sd2+oCjy1Oy1pdvN6ZQnuv/9QFPbc7Vm+KwKAMKDE.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-34-202-159-151.compute-1.amazonaws.com,34.202.159.151' (ECDSA) to the list of known hosts.  
|_(-_-)/ Amazon Linux 2 AMI  
__\_\_\_\_]  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-57-255 ~]$ -
```

```
ec2-user@ip-172-31-57-255:~  
[mitali611@LAPTOP-08H7518L:/mnt/c/Users/smita$ ssh -i ~/ssh/lab3keypair.pem ec2-user@ec2-34-202-159-151.compute-1.amazonaws.com  
The authenticity of host 'ec2-34-202-159-151.compute-1.amazonaws.com (34.202.159.151)' can't be established.  
ECDSA key fingerprint is SHA256:sd2+oCjy1Oy1pdvN6ZQnuv/9QFPbc7Vm+KwKAMKDE.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-34-202-159-151.compute-1.amazonaws.com,34.202.159.151' (ECDSA) to the list of known hosts.  
|_(-_-)/ Amazon Linux 2 AMI  
__\_\_\_\_]  
https://aws.amazon.com/amazon-linux-2/  
[ec2-user@ip-172-31-57-255 ~]$ sudo amazon-linux-extras install nginx1  
Installing nginx  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-nginx1  
12 metadata files removed  
4 sqlite files removed  
0 metadata files removed  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core  
amzn2extra-docker  
amzn2extra-nginx1  
(1/7): amzn2-core/2/x86_64/group_gz | 3.7 kB 00:00:00  
(2/7): amzn2-core/2/x86_64/updateInfo | 3.0 kB 00:00:00  
(3/7): amzn2extra-nginx1/2/x86_64-primary_db | 2.5 kB 00:00:00  
(4/7): amzn2extra-docker/2/x86_64/updateInfo | 362 kB 00:00:00  
(5/7): amzn2extra-nginx1/2/x86_64-updateInfo | 23 kB 00:00:00  
(6/7): amzn2extra-docker/2/x86_64-primary_db | 76 B 00:00:00  
(7/7): amzn2-core/2/x86_64-primary_db | 76 kB 00:00:00  
71% [=====] 18 MB/s | 37 MB 00:00:00 ETA
```

After successful installation of nginx changing index.html to distinguish between load balancer pages and server pages.

```
ec2-user@ip-172-31-57-255:/usr/share/nginx/html
[ec2-user@ip-172-31-57-255 html]$ cd /usr/share/nginx/html
[ec2-user@ip-172-31-57-255 html]$ ls
404.html 50x.html  icons index.html nginx-logo.png poweredby.png
[ec2-user@ip-172-31-57-255 html]$ sudo vim index.html
[ec2-user@ip-172-31-57-255 html]$ sudo systemctl reload nginx
[ec2-user@ip-172-31-57-255 html]$ sudo vim index.html
[ec2-user@ip-172-31-57-255 html]$
```

```
ec2-user@ip-172-31-57-255:/usr/share/nginx/html
.alert {
    border: 2px solid #000;
}
img {
    border: 2px solid #fff;
    padding: 2px;
    margin: 2px;
}
a:hover img {
    border: 2px solid #294172;
}
.logos {
    margin: 1em;
    text-align: center;
}
/*]]>*/
```

</style>

</head>

<body>

```
    <h1>Welcome to <strong>Loadbalancer Lab3 Cloud Formation Template</strong> on Amazon Linux!</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
                Amazon Linux. It is located in
                <tt>/usr/share/nginx/html</tt>.</p>
                <p>You should now put your content in a [ ] 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>

```

"index.html" 111L, 3557C

Now If I am going to output tab under stack, it gives me deployed website, upon clicking I can see the load balancer console because right now I have not configured nginx.conf file.

CloudFormation - Stack Lab3CloudFormationStack

Lab3CloudFormationStack

Outputs (1)

Key	Value	Description
WebsiteURL	http://ec2-34-202-159-151.compute-1.amazonaws.com	URL for newly created LAMP stack

Welcome to **Loadbalancer Lab3 Cloud Formation Template** on Amazon Linux!

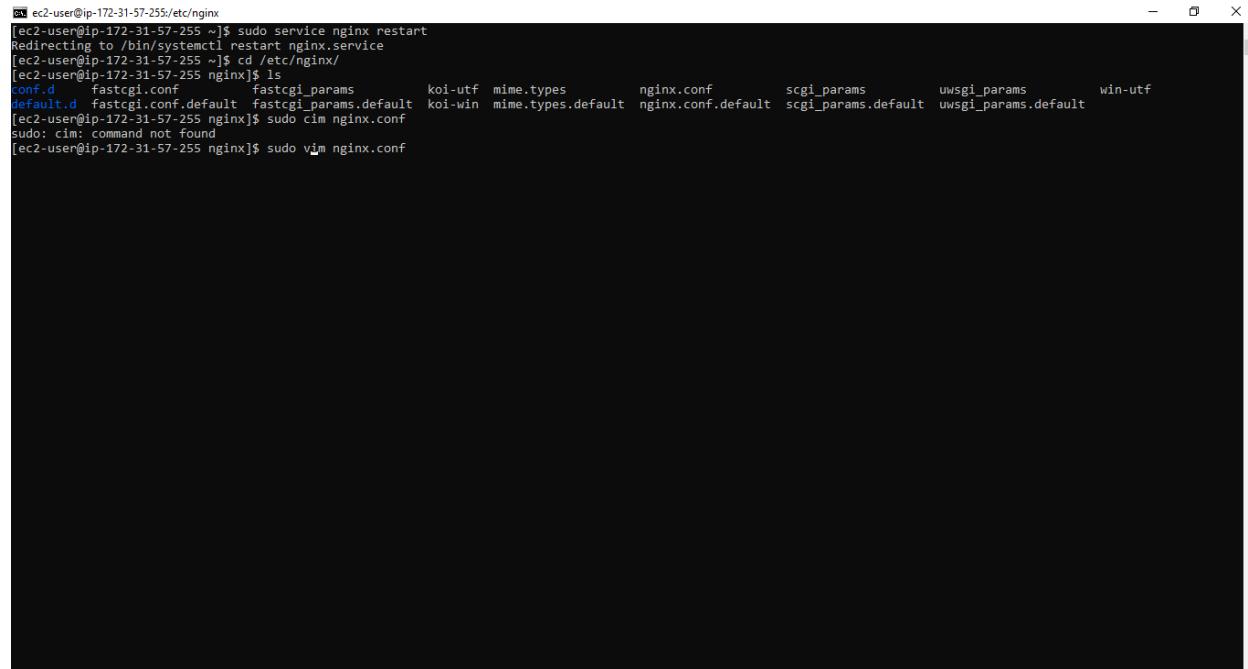
This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

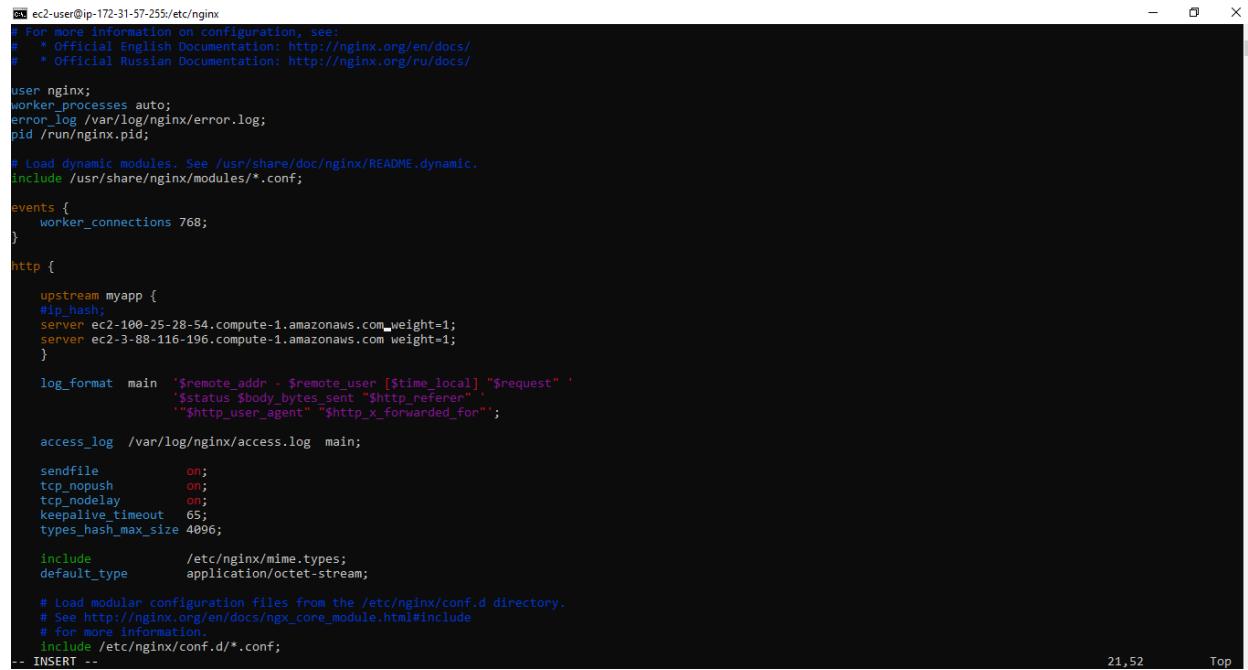
This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. It is located in `/usr/share/nginx/html`. You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

NGINX

Updating nginx.conf file in load balancer.



```
[ec2-user@ip-172-31-57-255 etc/nginx]$ sudo service nginx restart
Redirecting to /bin/systemctl restart nginx.service
[ec2-user@ip-172-31-57-255 ~]$ cd /etc/nginx/
[ec2-user@ip-172-31-57-255 nginx]$ ls
conf.d      fastcgi_params    koi-utf  mime.types      nginx.conf      scgi_params    uwsgi_params      win-utf
default.conf  fastcgi.conf.default  fastcgi_params.default  koi-win  mime.types.default  nginx.conf.default  scgi_params.default  uwsgi_params.default
[ec2-user@ip-172-31-57-255 nginx]$ sudo vim nginx.conf
sudo: vim: command not found
[ec2-user@ip-172-31-57-255 nginx]$ sudo vim nginx.conf
```



```
[ec2-user@ip-172-31-57-255 etc/nginx]
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 768;
}

http {
    upstream myapp {
        #ip_hash;
        server ec2-100-25-28-54.compute-1.amazonaws.com weight=1;
        server ec2-3-88-116-196.compute-1.amazonaws.com weight=1;
    }

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer"'
                    '"$http_user_agent" "$http_X_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile          on;
    tcp_nopush        on;
    tcp_nodelay       on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

    include           /etc/nginx/mime.types;
    default_type     application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;
-- INSERT --
```

```

ec2-user@ip-172-31-57-255:/etc/nginx
# See http://nginx.org/en/docs/ngx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    listen      80;
    listen      [::]:80;
    server_name myapp.com;
    root       /usr/share/nginx/html;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        proxy_pass http://myapp;
    }

    error_page 404 /404.html;
    location = /40x.html {

    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}

# Settings for a TLS enabled server.

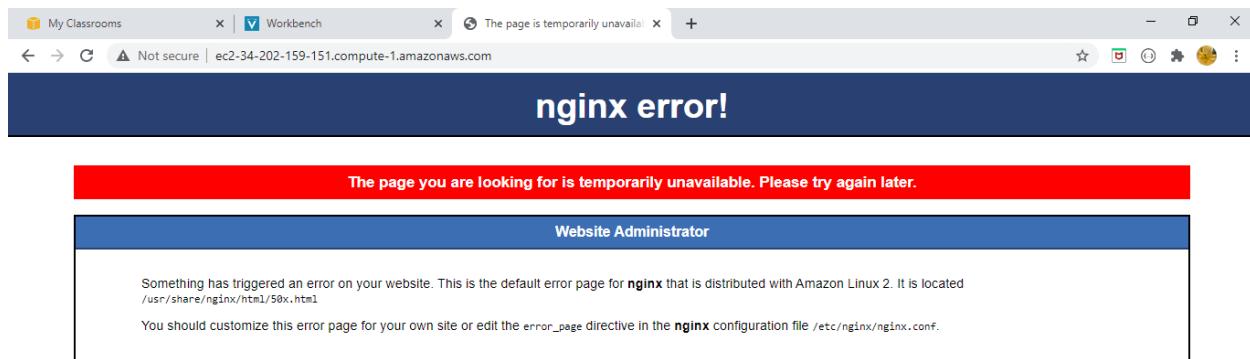
server {
    listen      443 ssl http2;
    listen      [::]:443 ssl http2;
    server_name .;
    root       /usr/share/nginx/html;

    ssl_certificate "/etc/pki/nginx/server.crt";
    ssl_certificate_key "/etc/pki/nginx/private/server.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROFILE+SYSTEM;
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;
}

```

Now If I go to website it is trying to redirect to servers but we have not installed nginx on other servers so website is not able to launch the page, but my configuration is working which I have set in nginx.conf.



Installing nginx on other servers and updating index.html file for server 1 and server2.

```
ec2-user@ip-172-31-55-150:~  
[unita011@LAPTOP-08H7518U:~/mnt/c/Users/smita]$ ssh -i ~/.ssh/lab3keypair.pem ec2-user@ec2-100-25-28-54.compute-1.amazonaws.com  
The authenticity of host 'ec2-100-25-28-54.compute-1.amazonaws.com (100.25.28.54)' can't be established.  
ECDSA key fingerprint is SHA256:GIRan1w09g9y5ZkXtsMlyI3ajKDM+uJVA864NKev6o.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'ec2-100-25-28-54.compute-1.amazonaws.com,100.25.28.54' (ECDSA) to the list of known hosts.  
_ | _ / Amazon Linux 2 AMI  
_ \_ |_  
  
https://aws.amazon.com/amazon-linux-2/  
No packages needed for security; 2 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-55-150 ~]$ sudo amazon-linux-extras install nginx1  
Installing nginx  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-nginx1  
12 metadata files removed  
4 sqlite files removed  
0 metadata files removed  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
amzn2-core  
amzn2extra-docker  
amzn2extra-nginx1  
(1/7): amzn2-core/2/x86_64/group_gz 3.7 kB 00:00:00  
(2/7): amzn2-core/2/x86_64/updateinfo 3.0 kB 00:00:00  
(3/7): amzn2extra-nginx1/2/x86_64/primary_db 3.0 kB 00:00:00  
(4/7): amzn2extra-docker/2/x86_64/updateinfo 2.5 kB 00:00:00  
(5/7): amzn2extra-nginx1/2/x86_64/updateinfo 362 kB 00:00:00  
(6/7): amzn2extra-docker/2/x86_64/primary_db 23 kB 00:00:00  
(7/7): amzn2-core/2/x86_64/primary_db 76 B 00:00:00  
52% [=====] 76 B 00:00:00  
] 0.0 B/s 27 MB --::-- ETA
```

```
ec2-user@ip-172-31-55-150:~$ cd /usr/share/nginx/html  
[ec2-user@ip-172-31-55-150 ~]$ ls  
[ec2-user@ip-172-31-55-150 html]$ ls  
404.html 50x.html  icons index.html  nginx-logo.png  poweredby.png  
[ec2-user@ip-172-31-55-150 html]$ sudo vim index.html  
[ec2-user@ip-172-31-55-150 html]$ sudo vim index.html
```

```

ec2-user@ip-172-31-55-150:/usr/share/nginx/html
        border-bottom: 2px solid #294172;
    }
    hr {
        display: none;
    }
    .content {
        padding: 1em 5em;
    }
    .alert {
        border: 2px solid #000;
    }
    img {
        border: 2px solid #fff;
        padding: 2px;
        margin: 2px;
    }
    a:hover img {
        border: 2px solid #294172;
    }
    .logos {
        margin: 1em;
        text-align: center;
    }
/*]]>/
</style>
</head>

<body>
<h1>Welcome to <strong>Server 1 Cloud Formation template</strong> on Amazon Linux!</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

```

nginx console for server1

The screenshot shows a web browser window with multiple tabs open. The active tab displays the Nginx default index.html page for 'Server 1 Cloud Formation template' on Amazon Linux. The page content includes a welcome message, a paragraph about testing the server's proper operation, and a section for a 'Website Administrator'. At the bottom, there is a large green 'NGINX' logo.

Welcome to **Server 1 Cloud Formation template** on Amazon Linux!

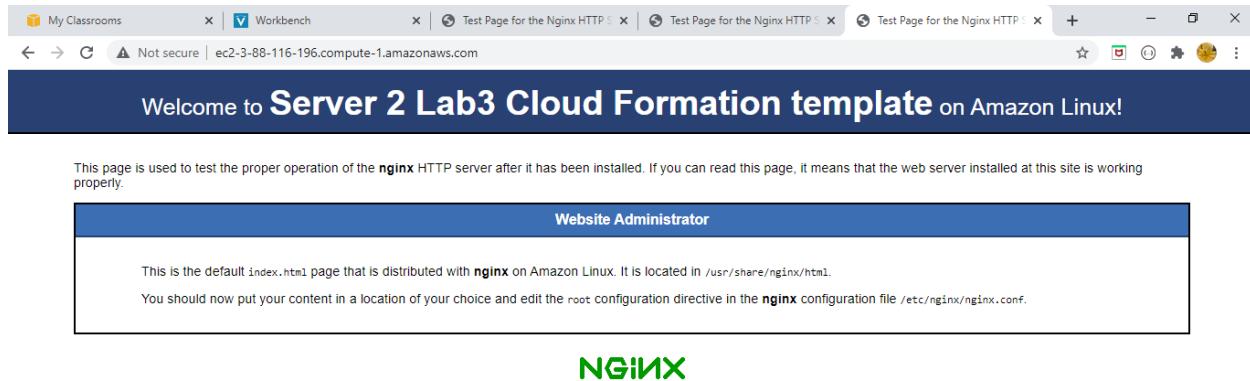
This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Amazon Linux. 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

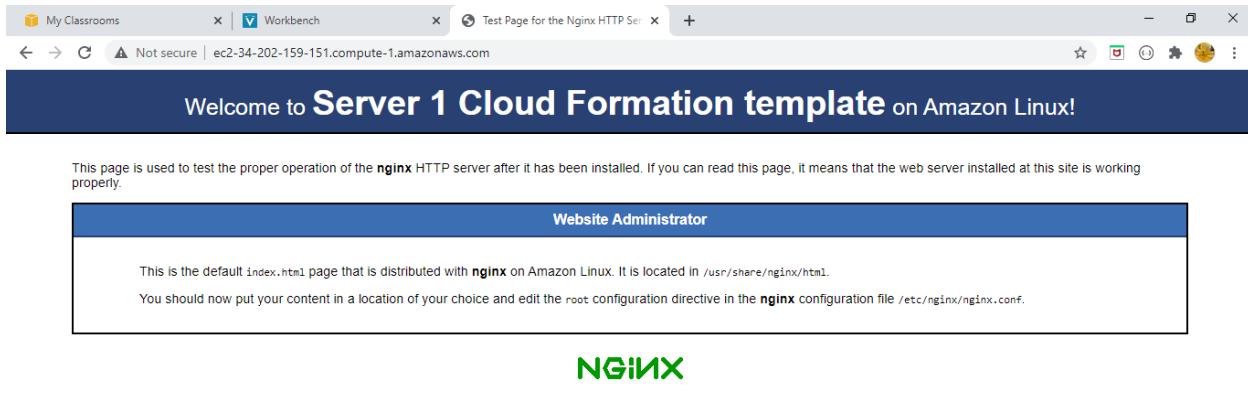
nginx console for server2



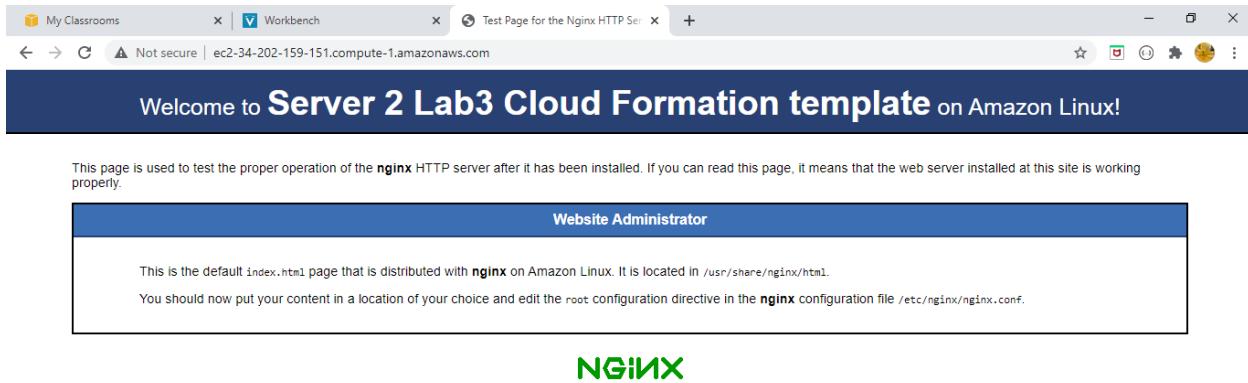
After nginx installed on servers if I go to output tab and click on the deployed website it redirects me to other servers one by one because the weight is equally distributed.

A screenshot of the AWS CloudFormation console. The top navigation bar shows 'My Classrooms', 'Workbench', and the current stack 'CloudFormation - Stack Lab3CloudFormationStack'. The main area shows the 'Stacks (1)' list with 'Lab3CloudFormationStack' selected. On the right, the 'Outputs' tab is active, displaying a table with one row. The table has columns for 'Key', 'Value', 'Description', and 'Export name'. The 'Key' column contains 'WebsiteURL', the 'Value' column contains 'http://ec2-34-202-159-151.compute-1.amazonaws.com', and the 'Description' column contains 'URL for newly created LAMP stack'. The 'Value' column is hyperlinked.

Redirected to Server 1



Redirected to Server 2



I am adding two more servers in template to update my stack, so now my stack will keep the old resources and will create the new resources.

```

"Server2": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId" : "ami-0742b4e673072066f",
    "InstanceType" : { "Ref" : "InstanceType" },
    "SecurityGroupIds" : [ {"Ref" : "WebServerSecurityGroup"} ],
    "KeyName" : { "Ref" : "KeyName" }
  }
},
"Server3": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId" : "ami-0742b4e673072066f",
    "InstanceType" : { "Ref" : "InstanceType" },
    "SecurityGroupIds" : [ {"Ref" : "WebServerSecurityGroup"} ],
    "KeyName" : { "Ref" : "KeyName" }
  }
},
"Server4": {
  "Type": "AWS::EC2::Instance",
  "Properties": {
    "ImageId" : "ami-0742b4e673072066f",
    "InstanceType" : { "Ref" : "InstanceType" },
    "SecurityGroupIds" : [ {"Ref" : "WebServerSecurityGroup"} ],
    "KeyName" : { "Ref" : "KeyName" }
  }
},

```

Under stack, select the created stack and click on update

Stack name	Status	Created time	Description
Lab3CloudFormationStack	✓ CREATE_COMPLETE	2021-04-12 22:52:29 UTC-0400	AWS CloudFormation sample template

On clicking update it gives option about how you want to edit your template, I selected replace template

The screenshot shows the AWS CloudFormation console interface. The top navigation bar includes tabs for 'My Classrooms', 'Workbench', and 'CloudFormation - Stack Lab3CloudFormation'. The main content area is titled 'Prerequisite - Prepare template'. On the left, a sidebar lists steps: Step 2 'Specify stack details', Step 3 'Configure stack options', Step 4 'Review', and 'Feedback English (US) ▾'. The main panel has two sections: 'Prepare template' and 'Specify template'. Under 'Prepare template', it says '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.' Below are three radio buttons: 'Use current template' (unchecked), 'Replace current template' (checked), and 'Edit template in designer' (unchecked). Under 'Specify template', it says 'A template is a JSON or YAML file that describes your stack's resources and properties.' It includes a 'Template source' section with 'Amazon S3 URL' selected (radio button checked) and a text input field containing 'https://'. There is also an 'Upload a template file' option (radio button unchecked). At the bottom right are 'Cancel' and 'Next' buttons.

This screenshot is identical to the one above, but the 'Amazon S3 URL' input field now contains the uploaded file name: 'lab3cloudFormation.json'. The rest of the interface, including the sidebar, template source options, and bottom buttons, remains the same.

After upload I checked my template in designer for errors and the template design

My Classrooms Workbench AWS CloudFormation Designer

Services ▾ Search for services, features, marketplace products, and doc [Alt+S] vocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia Support

File: 'template1'

Resource types

- ACMPCA
- AccessAnalyzer
- AmazonMQ
- Amplify
- ApiGateway
- ApiGatewayV2
- AppConfig
- AppFlow
- AppIntegrations

Choose template language: **JSON** **YAML**

```
template1
1 {
2   "AWSTemplateFormatVersion": "2010-09-09",
3   "Description": "AWS CloudFormation sample template",
4
5   "Parameters": {
6     "KeyName": {
7       "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instance",
8       "Type": "AWS::EC2::KeyPair::KeyName",
9       "Default": "lab3keypair"
10    }
11  }
12}
```

Components Template

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

My Classrooms Workbench CloudFormation - Stack Lab3CloudFormationStack

Services ▾ Search for services, features, marketplace products, and doc [Alt+S] vocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia Support

CloudFormation > Stacks > Lab3CloudFormationStack > Update stack

Step 1 Step 2 Step 3 Step 4

Specify stack details

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
KeyName Name of an existing EC2 KeyPair to enable SSH access to the instance lab3keypair
SSHLocation The IP address range that can be used to SSH to the EC2 instances 173.70.217.190/32

Cancel Previous Next

Feedback English (US) © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

The screenshot shows the AWS CloudFormation 'Update stack' interface. At the top, there's a header bar with tabs for 'My Classrooms', 'Workbench', and 'CloudFormation - Stack Lab3CloudFormationStack'. The main content area has a title 'IAM role - optional' with a note about choosing a role for CloudFormation operations. Below this is a section titled 'Advanced options' with three expandable sections: 'Stack policy during update', 'Rollback configuration', and 'Notification options'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

The screenshot shows the 'Review Lab3CloudFormationStack' step of the CloudFormation stack update process. On the left, a sidebar lists steps: 'Step 1 Specify template' (selected), 'Step 2 Specify stack details', 'Step 3 Configure stack options', and 'Step 4 Review'. The main panel shows the 'Step 1: Specify template' section with a 'Template' tab. It displays a 'Template URL' field containing the URL <https://s3-external-1.amazonaws.com/cf-templates-13edpgj2uquad-us-east-1/20211030AT-lab3cloudFormation.json>. A 'Stack description' field below it contains 'AWS CloudFormation sample template'. There are 'Edit' buttons for both the template URL and the stack description.

Step 2: Specify stack details

Parameters (3)

Key	Value
InstanceType	t2.micro
KeyName	lab3keypair
SSHLocation	173.70.217.190/32

Step 3: Configure stack options

Tags (0)

Key	Value

Under changes it will show the update changes which are defined in the template, if changes are correct click on update template.

Notification options

No notification options
There are no notification options defined

Change set preview

Changes (2)

Action	Logical ID	Physical ID	Resource type	Replacement	Module
Add	Server3	-	AWS::EC2::Instance	-	-
Add	Server4	-	AWS::EC2::Instance	-	-

Cancel Previous View change set Update stack

Under events, new events are generated for the creation of new servers.

Screenshot of the AWS CloudFormation console showing the 'Lab3CloudFormationStack' stack details.

The 'Events' tab is selected, displaying 18 events:

Timestamp	Logical ID	Status	Status reason
2021-04-12 23:49:01 UTC-0400	Lab3CloudFormationStack	CREATE_IN_PROGRESS	User Initiated
2021-04-12 22:53:19 UTC-0400	Lab3CloudFormationStack	CREATE_COMPLETE	-
2021-04-12 22:53:17 UTC-0400	LoadBalancer	CREATE_COMPLETE	-

Screenshot of the AWS CloudFormation console showing the 'Lab3CloudFormationStack' stack details.

The 'Events' tab is selected, displaying 22 events:

Timestamp	Logical ID	Status	Status reason
2021-04-12 23:49:09 UTC-0400	Server3	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-12 23:49:09 UTC-0400	Server4	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-12 23:49:07 UTC-0400	Server3	CREATE_IN_PROGRESS	-
2021-04-12 23:49:07 UTC-0400	Server4	CREATE_IN_PROGRESS	-

The screenshot shows the AWS CloudFormation Events page. On the left, the 'Stacks' section displays one stack named 'Lab3CloudFormationStack' with a status of 'UPDATE_COMPLETE'. On the right, the 'Events' tab is selected, showing a table of 26 events. The table includes columns for Timestamp, Logical ID, Status, and Status reason. Key entries include:

Timestamp	Logical ID	Status	Status reason
2021-04-12 23:49:45 UTC-0400	Lab3CloudFormationStack	UPDATE_COMPLETE	-
2021-04-12 23:49:44 UTC-0400	Lab3CloudFormationStack	UPDATE_COMPLETE	CL EANUP_IN_PROGRESS
2021-04-12 23:49:42 UTC-0400	Server4	CREATE_COMPLETE	-
2021-04-12 23:49:41 UTC-0400	Server3	CREATE_COMPLETE	-
2021-04-12 23:49:09 UTC-0400	Server3	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-12 23:49:09 UTC-0400	Server4	CREATE_IN_PROGRESS	Resource creation

Under resources I can see two new servers are created.

The screenshot shows the AWS CloudFormation Resources page. On the left, the 'Stacks' section displays one stack named 'Lab3CloudFormationStack' with a status of 'UPDATE_COMPLETE'. On the right, the 'Resources' tab is selected, showing a table of 7 resources. The table includes columns for Logical ID, Physical ID, Type, and Status. Key entries include:

Logical ID	Physical ID	Type	Status
LoadBalancer	i-03a113234385edb54	AWS::EC2::Instance	CREATE_COMPLETE
LoadBalancerSecurityGroup	sg-09cf812e1427a2f6b	AWS::EC2::SecurityGroup	CREATE_COMPLETE
Server1	i-0bd6e7be5119c3cf0	AWS::EC2::Instance	CREATE_COMPLETE
Server2	i-007afcc13194bc009	AWS::EC2::Instance	CREATE_COMPLETE
Server3	i-0b3c6e390c2a8a354	AWS::EC2::Instance	CREATE_COMPLETE
Server4	i-0940c7cf04ed06e8	AWS::EC2::Instance	CREATE_COMPLETE
WebServerSecurityGroup	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P7LTSN	AWS::EC2::SecurityGroup	CREATE_COMPLETE

Under instances I checked two new instances are running.

Server 3 details

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Server1	i-007afcc13194bc009	Running	t2.micro	2/2 checks passed	1 alarms	us-east-1e
LoadBalancer	i-03a113234385edb54	Running	t2.micro	2/2 checks passed	1 alarms	us-east-1e
Server3	i-0940c7cf04ed06e8	Running	t2.micro	Initializing	1/1 has i	us-east-1e
Server2	i-0bd6e7be5119c3cf0	Running	t2.micro	2/2 checks passed	1 alarms	us-east-1b
Server4	i-0b3c6e390c2a8a354	Running	t2.micro	Initializing	1/1 has i	us-east-1d

Below the table, the "Instance summary" section provides detailed information for the selected instance (Server3):

- Instance ID: i-0940c7cf04ed06e8 (Server3)
- Public IPv4 address: 100.26.156.147 | [open address](#)
- Private IPv4 addresses: 172.31.58.71
- Instance state: Running
- Public IPv4 DNS: ec2-100-26-156-147.compute-1.amazonaws.com | [open address](#)
- Private IPv4 DNS: ip-172-31-58-71.ec2.internal

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the security group configuration for the selected instance (Server3):

WebServerSecurityGroup-18GV62P77LTSN

Inbound rules

Port range	Protocol	Source	Security groups
80	TCP	0.0.0.0/0	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN
22	TCP	173.70.217.190/32	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN
443	TCP	0.0.0.0/0	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN

Outbound rules

Server 4 details

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Instances' section with 1/5 instances running. A single instance, 'Server4' (i-0b3c6e390c2a8a354), is selected. Below the table, the 'Instance summary' pane is expanded, showing detailed information for the selected instance.

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b3c6e390c2a8a354 (Server4)	3.81.51.206 open address	172.31.46.57

Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	ec2-3-81-51-206.compute-1.amazonaws.com open address	ip-172-31-46-57.ec2.internal

Instance type	Elastic IP addresses	VPC ID
t2.micro	-	vpc-2e883e53

AWS Compute Optimizer finding	IAM Role	Subnet ID
(User: arn:aws:s:151283216846:assumed-role:Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN)	-	-

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Instances' section with 1/5 instances running. A single instance, 'Server4' (i-0b3c6e390c2a8a354), is selected. Below the table, the 'WebServerSecurityGroup-18GV62P77LTSN' security group is selected, and its 'Inbound rules' are displayed.

Port range	Protocol	Source	Security groups
80	TCP	0.0.0.0/0	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN
22	TCP	173.70.217.190/32	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN
443	TCP	0.0.0.0/0	Lab3CloudFormationStack-WebServerSecurityGroup-18GV62P77LTSN

Installed nginx on server3 and server 4 and launched the console for both the servers.

Welcome to **Server 4 Lab3 Cloud Formation** on Amazon Linux!

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 Amazon Linux. 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

Welcome to **Server 3 Lab3 Cloud Formation Template** on Amazon Linux!

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 Amazon Linux. 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

Edited nginx.conf file in load balancer to distribute the weight between different servers.

```
ec2-user@ip-172-31-57-255:/etc/nginx
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 768;
}

http {
    upstream myapp {
        ip_hash;
        server ec2-3-88-116-196.compute-1.amazonaws.com weight=1;
        server ec2-100-25-28-54.compute-1.amazonaws.com weight=1;
        server ec2-100-26-156-147.compute-1.amazonaws.com weight=1;
        server ec2-3-81-51-206.compute-1.amazonaws.com weight=1;
    }

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

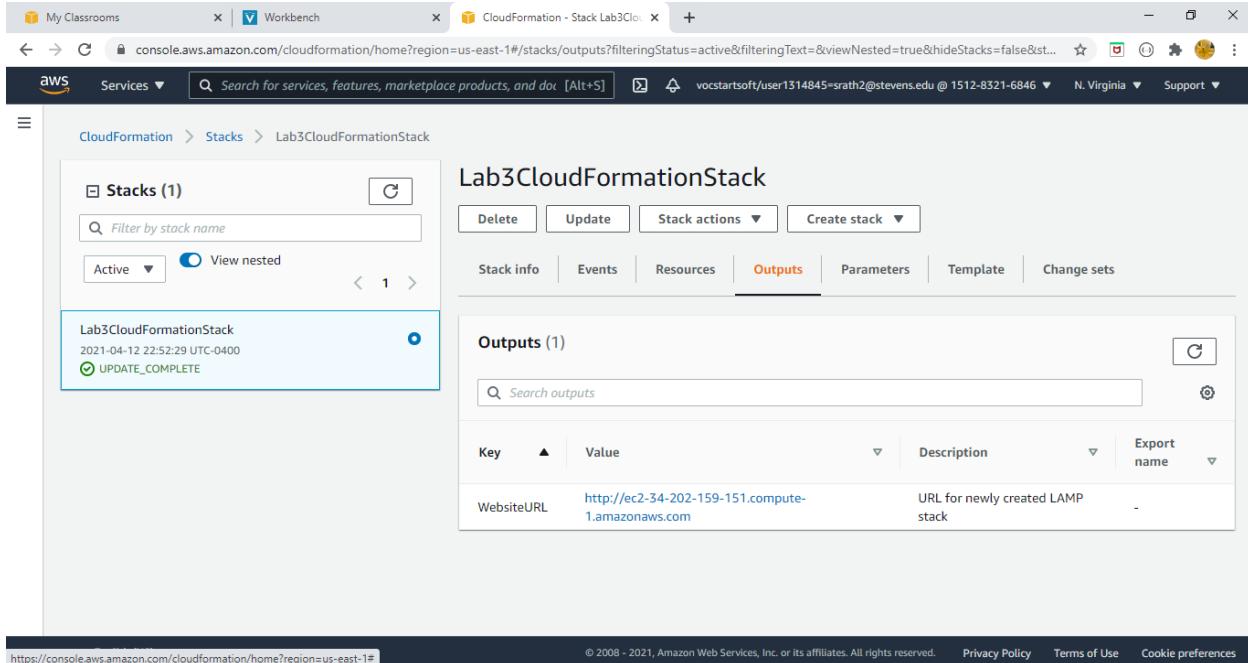
    sendfile      on;
    tcp_nopush    on;
    tcp_nodelay   on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs NGX_core_module.html#include
-- INSERT --
```

24,61 Top ▾

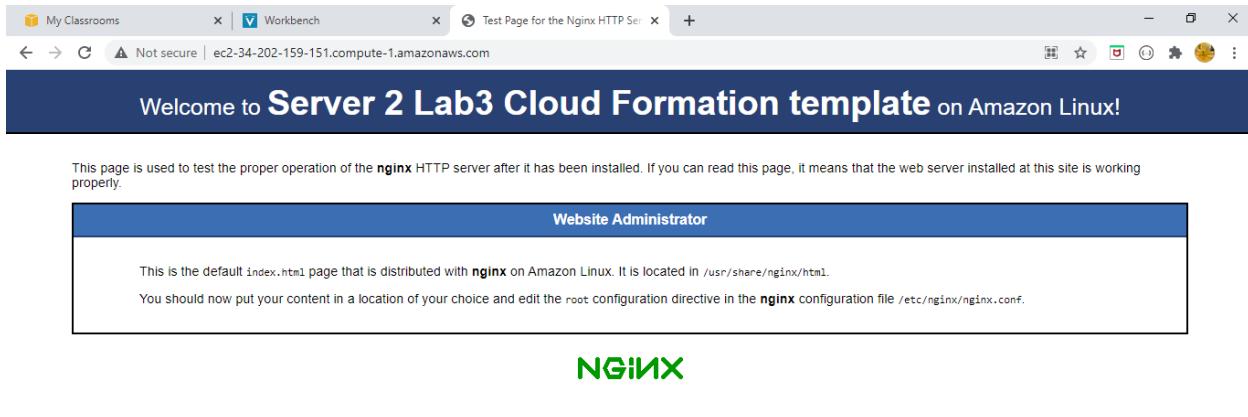
After changing nginx.conf file, I clicked on deployed website and it redirected to different servers one by one.



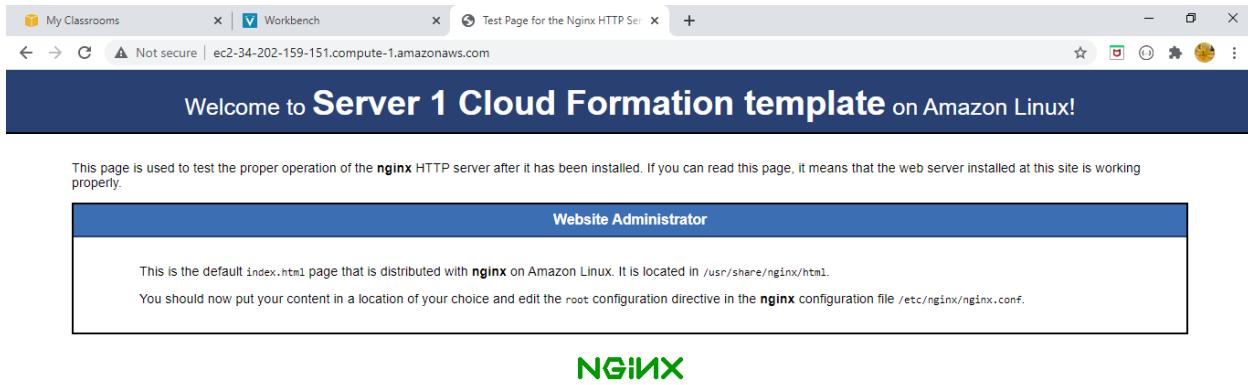
The screenshot shows the AWS CloudFormation console with the 'Outputs' tab selected. A single output named 'WebsiteURL' is listed, pointing to the URL <http://ec2-34-202-159-151.compute-1.amazonaws.com>. The output description is 'URL for newly created LAMP stack'. The status of the stack is shown as 'UPDATE_COMPLETE'.

Key	Value	Description	Export name
WebsiteURL	http://ec2-34-202-159-151.compute-1.amazonaws.com	URL for newly created LAMP stack	

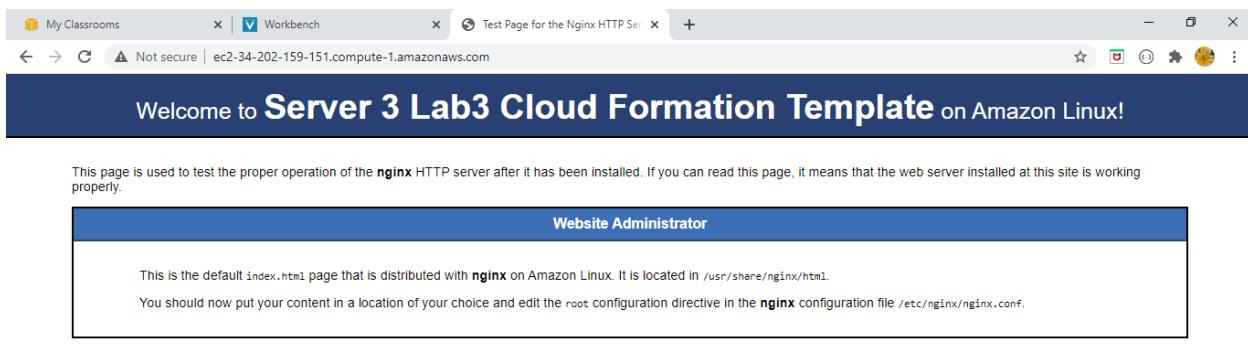
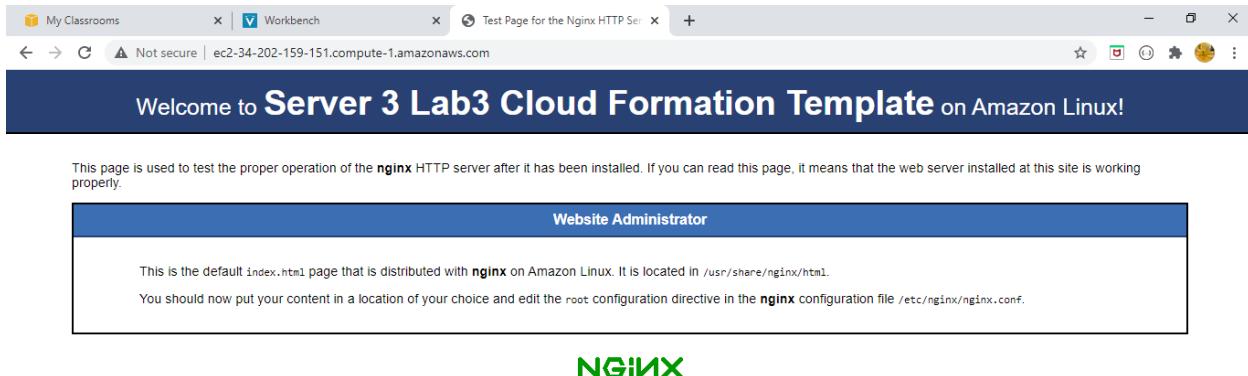
Redirected to Server 2



Redirected to Server 1



Redirected to Server 3



To verify again I ran ruby script in one of my servers to check the visits.

```
ec2-user@ip-172-31-46-57:~]$ ruby visit_server.rb -d ec2-34-202-159-151.compute-1.amazonaws.com
Starting to visit load balancing server
.....
```

Visit counts are same for all the servers.

```
ec2-user@ip-172-31-46-57:~]$ ruby visit_server.rb -d ec2-34-202-159-151.compute-1.amazonaws.com
Starting to visit load balancing server
.....
```

Summary

Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
[ec2-user@ip-172-31-46-57 ~]\$

For deleting the stack select the created stack and click on delete.

The screenshot shows the AWS CloudFormation console interface. At the top, there are three tabs: 'My Classrooms', 'Workbench', and 'CloudFormation - Stack Lab3CloudFormationStack'. The 'CloudFormation' tab is active. The main area is titled 'CloudFormation > Stacks'. A table lists one stack:

Stack name	Status	Created time	Description
Lab3CloudFormationStack	UPDATE_COMPLETE	2021-04-12 22:52:29 UTC-0400	AWS CloudFormation sample template

At the bottom of the page, there are links for 'Feedback', 'English (US)', and 'Cookie preferences', along with a copyright notice: '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

The screenshot shows the AWS CloudFormation console interface. The 'CloudFormation - Stack Lab3CloudFormationStack' tab is active. A modal dialog box is open, asking 'Delete Lab3CloudFormationStack?'. The dialog contains the following text: 'Deleting this stack will delete all stack resources. Resources will be deleted according to their DeletionPolicy. Learn more' with a link. There are two buttons at the bottom: 'Cancel' and a highlighted 'Delete stack' button.

The screenshot shows the AWS CloudFormation console with a single stack named "Lab3CloudFormationStack". The status of the stack is "DELETE_IN_PROGRESS", indicating it is being deleted. The stack was created on 2021-04-12 at 22:52:29 UTC-0400 and is described as an "AWS CloudFormation sample template".

Once deletion it will terminate all the instances as well, which were created during stack creation.

The screenshot shows the AWS EC2 Instances page with five instances listed. All instances are in a "Terminated" state. The instances are named Server1, LoadBalancer, Server3, Server2, and Server4. They are all of type t2.micro and located in the us-east-1 region. Each instance has 1 alarm associated with it.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Server1	i-007afcc13194bc009	Terminated	t2.micro	-	1 alarms	us-east-1e
LoadBalancer	i-03a113234385edb54	Terminated	t2.micro	-	1 alarms	us-east-1e
Server3	i-0940c7cf04ed06e8	Terminated	t2.micro	-	1 alarms	us-east-1e
Server2	i-0bd6e7be5119c3cf	Terminated	t2.micro	-	1 alarms	us-east-1b
Server4	i-0b3c6e390c2a8a354	Terminated	t2.micro	-	1 alarms	us-east-1d

Security groups will also be deleted with the stack deletion.

Name	Security group ID	Security group name	VPC ID	Description
cli-sg-lab2	sg-0074d8f6a644cada5	cli-sg-lab2	vpc-2e883e53	My security group
default	sg-0467ca0b	default	vpc-2e883e53	default VPC securi
default_elb_6ffaf4fd-d...	sg-0d490f3cdff9a3aed	default_elb_6ffaf4fd-d...	vpc-2e883e53	ELB created securi

<https://aws.amazon.com/cloudformation/>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/GettingStarted.html>

https://www.youtube.com/watch?v=_jqwVpO1w6A&t=158s

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/deploying.applications.html>

https://s3-external-1.amazonaws.com/cloudformation-templates-us-east-1/LAMP_Single_Instance.template

I have tried load balancer template as well to create a load balancer and a web server group which will create 4 instances and one load balancer, but I couldn't load the deployed website. I am putting screenshots for the template and steps which I have used for creation.

I have taken the reference from below two links to create the template.

https://docs.amazonaws.cn/en_us/AWSCloudFormation/latest/UserGuide/example-templates-autoscaling.html

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-elasticloadbalancingv2-loadbalancer.html>

Defined parameters keypair and instance types

```
{  
    "AWSTemplateFormatVersion": "2010-09-09",  
  
    "Description" : "Load Balancer Template";  
    "Parameters": {  
        "KeyName": {  
            "Description" : "The EC2 Key Pair to allow SSH access to the instances",  
            "Type" : "AWS::EC2::KeyPair::KeyName",  
            "Default": "lab3keypair",  
            "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."  
        },  
        "InstanceType" : {  
            "Description" : "WebServer EC2 instance type",  
            "Type" : "String",  
            "Default" : "t2.micro"  
        }  
    },  
},
```

Defined resources

Security group and inbound rules

```
"Resources": {  
    "EC2SecurityGroup": {  
        "Properties": {  
            "SecurityGroupIngress": [  
                {  
                    "FromPort": "22",  
                    "CidrIp": "173.70.217.190/32",  
                    "ToPort": "22",  
                    "IpProtocol": "tcp"  
                },  
                {  
                    "FromPort": "80",  
                    "CidrIp": "0.0.0.0/0",  
                    "ToPort": "80",  
                    "IpProtocol": "tcp"  
                }  
            ],  
            "GroupDescription": "HTTP and SSH access"  
        },  
        "Type": "AWS::EC2::SecurityGroup"  
    },
```

Defined load balancer it will create a load balancer and the type will be elastic load balancer.

```
"ElasticLoadBalancer": {
  "Properties": {
    "Listeners": [
      {
        "InstancePort": "80",
        "Protocol": "HTTP",
        "LoadBalancerPort": "80"
      }
    ],
    "HealthCheck": {
      "HealthyThreshold": "2",
      "Timeout": "5",
      "Interval": "10",
      "UnhealthyThreshold": "5",
      "Target": "HTTP:80/"
    },
    "AvailabilityZones": {
      "Fn::GetAZs": ""
    }
  },
  "Type": "AWS::ElasticLoadBalancing::LoadBalancer"
},
```

Defined autoscaling group which will create the number of instances given in min and it can extend the number of instances to max and it is referencing to ElasticLoadBalancer which I have defined earlier

```
"WebServerGroup": {
  "Properties": {
    "LoadBalancerNames": [
      {
        "Ref": "ElasticLoadBalancer"
      }
    ],
    "LaunchConfigurationName": {
      "Ref": "LaunchConfig"
    },
    "AvailabilityZones": {
      "Fn::GetAZs": ""
    },
    "MinSize": "4",
    "MaxSize": "4"
  },
  "Type": "AWS::AutoScaling::AutoScalingGroup"
},
```

In launchConfig I have defined my instance types, security groups, key pair and installing nginx which will create the 4 instances.

```
"LaunchConfig": {
    "Properties": {
        "SecurityGroups": [
            {
                "Ref": "EC2SecurityGroup"
            }
        ],
        "KeyName": { "Ref": "KeyName" },
        "InstanceType": { "Ref": "InstanceType" },
        "ImageId": { "Fn::FindInMap": [ "AWSRegionArch2AMI", { "Ref": "AWS::Region" },
            { "Fn::FindInMap": [ "AWSInstanceType2Arch", { "Ref": "InstanceType" }, "Arch" ] } ] },
        "UserData": { "Fn::Base64": { "Fn::Join": [ "", [
            "#!/bin/bash -xe\n",
            "yum install -y aws-cfn-bootstrap\n",

            "# Install the files and packages from the metadata\n",
            "/opt/aws/bin/cfn-init -v ",
            "  --stack ", { "Ref": "AWS::StackName" },
            "  --resource WebServerInstance ",
            "  --configsets Install ",
            "  --region ", { "Ref": "AWS::Region" }, "\n"
        ]]} },
        "Type": "AWS::AutoScaling::LaunchConfiguration",
        ...
    }
},

{
    "Type": "AWS::AutoScaling::LaunchConfiguration",
    "Metadata": {
        "Comment1": "Configure the bootstrap helpers to install the Apache Web Server and PHP",
        "Comment2": "Save website content to /var/www/html/index.php",
        ...
    },
    "AWS::CloudFormation::Init": {
        "configSets": {
            "Install": [ "Install" ]
        },
        "Install": {
            "packages": {
                "yum": {
                    "nginx": []
                }
            },
            "services": {
                "sysvinit": {
                    "httpd": { "enabled": "true", "ensureRunning": "true" }
                }
            }
        }
    }
},
```

In the output I am giving the url as load balancer public dns .

```
"Outputs": {
  "WebsiteURL": {
    "Value": {
      "Fn::Join": [
        "",
        [
          "http://",
          {
            "Fn::GetAtt": [
              "ElasticLoadBalancer",
              "DNSName"
            ]
          }
        ]
      ]
    }
  }
}
```

Creating stack with the template

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with tabs like 'CloudFormation', 'Services', and 'Management & Governance'. Below the navigation, the main heading is 'AWS CloudFormation' with the subtext 'Model and provision all your cloud infrastructure'. A paragraph explains that CloudFormation provides a common language to describe and provision infrastructure resources. To the right, a prominent call-to-action button says 'Create stack'. A sidebar on the right is titled 'Getting started' and lists links such as 'What is AWS CloudFormation', 'Getting started with CloudFormation', 'Learn template basics', and 'Quick starts'. At the bottom of the page, there are footer links for 'Feedback', 'English (US)', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

CloudFormation - Stack

console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/template

Services ▾

Search for services, features, marketplace products, and documentation [Alt+S]

vocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia Support

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review

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.

Template is ready Use a sample template Create 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

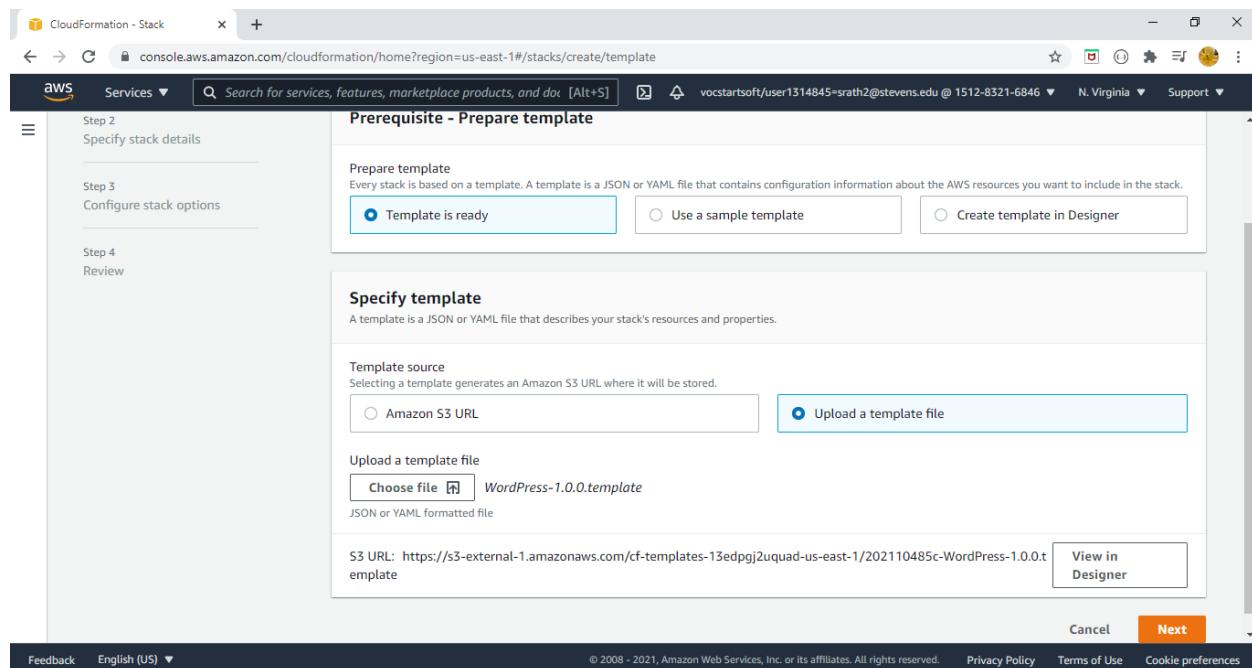
WordPress-1.0.0.template
JSON or YAML formatted file

S3 URL: <https://s3-external-1.amazonaws.com/cf-templates-13edpgj2uquad-us-east-1/202110485c-WordPress-1.0.0.template>

Cancel Next

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences



CloudFormation - Stack

console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/parameters

Services ▾

Search for services, features, marketplace products, and documentation [Alt+S]

vocstartsoft/user1314845=srath2@stevens.edu @ 1512-8321-6846 N. Virginia Support

Step 1 Specify template

Step 2 Specify stack details

Step 3 Configure stack options

Step 4 Review

Specify stack details

Stack name

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.

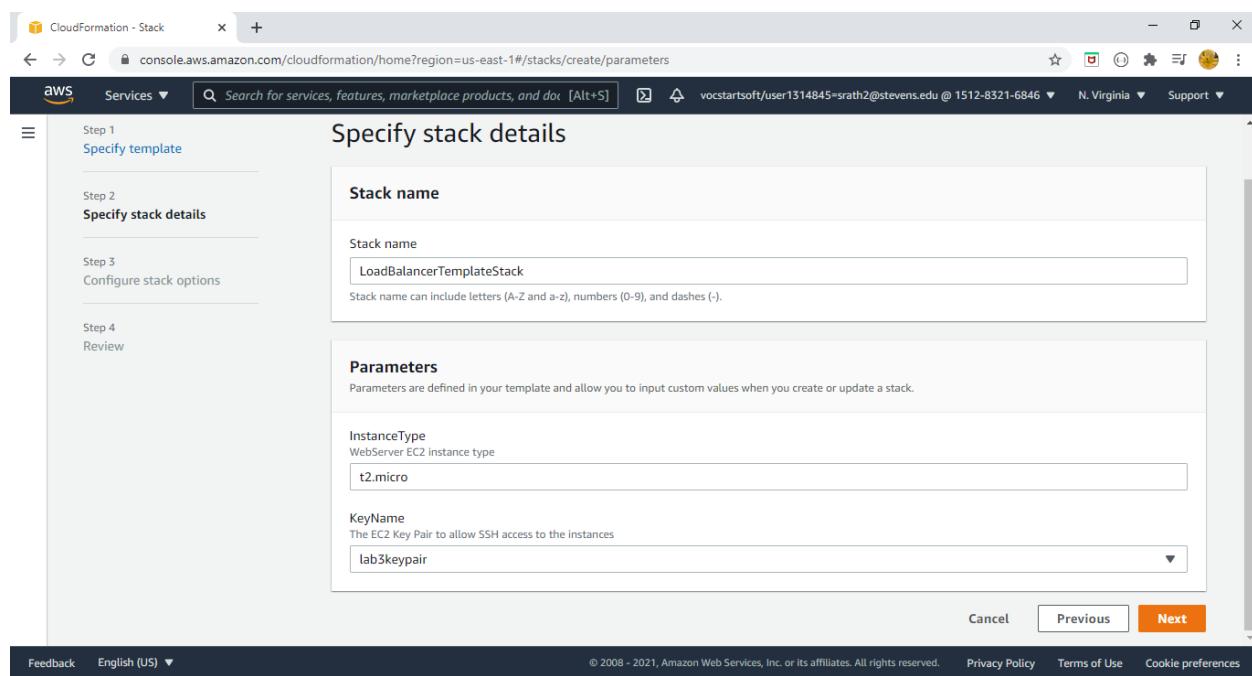
InstanceType
WebServer EC2 instance type

KeyName
The EC2 Key Pair to allow SSH access to the instances

Cancel Previous Next

Feedback English (US) ▾

© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences



Screenshot of the AWS CloudFormation 'Advanced options' configuration screen.

The top navigation bar shows the URL: console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/options.

The main section is titled "Advanced options" and contains four expandable sections:

- ▶ 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. [Learn more](#)
- ▶ Notification options**
- ▶ Stack creation options**

At the bottom right are buttons for "Cancel", "Previous", and "Next".

Screenshot of the AWS CloudFormation 'Review' step of the stack creation wizard.

The top navigation bar shows the URL: console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/summary.

The left sidebar shows the progress: Step 3 (Configure stack options) and Step 4 (Review).

The main area displays the following details:

- Template URL: <https://s3-external-1.amazonaws.com/cf-templates-13edpgj2uquad-us-east-1/2021104HG5-WordPress-1.0.0.template>
- Stack description: Load Balancer Template
- Estimate cost: [Estimate cost](#)

Below this is the "Step 2: Specify stack details" section, which includes a "Parameters (2)" table:

Key	Value
InstanceType	t2.micro
KeyName	lab3keypair

CloudFormation - Stack

console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/create/summary

aws Services Search for services, features, marketplace products, and doc [Alt+S]

No notification options
There are no notification options defined

Stack creation options

Rollback on failure
Enabled

Timeout
-

Termination protection
Disabled

Quick-create link

Cancel Previous Create change set Create stack

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

This screenshot shows the 'Create New Stack' wizard in the AWS CloudFormation console. It displays 'No notification options' and the 'Stack creation options' section. Under 'Rollback on failure', 'Enabled' is selected. There is a single timeout entry with a minus sign. Under 'Termination protection', 'Disabled' is selected. At the bottom, there are buttons for 'Quick-create link', 'Cancel', 'Previous', 'Create change set', and a prominent orange 'Create stack' button.

Stack creation is initialized

CloudFormation - Stack LoadBal... ▾ +

console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/events?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A151283216846...

aws Services Search for services, features, marketplace products, and doc [Alt+S]

CloudFormation > Stacks > LoadBalancerTemplateStack

Stacks (1)

LoadBalancerTemplateStack

2021-04-13 22:36:04 UTC-0400

CREATE_IN_PROGRESS

Events (1)

Timestamp Logical ID Status Status reason

2021-04-13 22:36:04 LoadBalancerTemplateStack CREATE_IN_PROG User Initiated

Feedback English (US) ▾ © 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

This screenshot shows the 'Events' tab for the 'LoadBalancerTemplateStack' in the AWS CloudFormation console. It lists one event: 'CREATE_IN_PROGRESS' at '2021-04-13 22:36:04 UTC-0400'. The event is associated with the logical ID 'LoadBalancerTemplateStack' and has a status of 'User Initiated'. The page also includes tabs for 'Stack info', 'Resources', 'Outputs', 'Parameters', 'Template', and 'Change sets'.

Screenshot of the AWS CloudFormation console showing the 'Events' tab for the 'LoadBalancerTemplateStack'. The stack status is 'CREATE_IN_PROGRESS'. The events table shows four entries:

Timestamp	Logical ID	Status	Status reason
2021-04-13 22:46:50 UTC-0400	WebServerGroup	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-13 22:46:50 UTC-0400	WebServerGroup	CREATE_IN_PROGRESS	-
2021-04-13 22:46:47 UTC-0400	LaunchConfig	CREATE_COMPLETE	-
2021-04-13 22:46:46 UTC-0400	LaunchConfig	CREATE_IN_PROGRESS	Resource creation Initiated

Screenshot of the AWS CloudFormation console showing the 'Events' tab for the 'LoadBalancerTemplateStack'. The stack status is 'CREATE_COMPLETE'. The events table shows eight entries:

Timestamp	Logical ID	Status	Status reason
2021-04-13 22:48:10 UTC-0400	LoadBalancerTemplateStack	CREATE_COMPLETE	-
2021-04-13 22:48:08 UTC-0400	WebServerGroup	CREATE_COMPLETE	-
2021-04-13 22:46:50 UTC-0400	WebServerGroup	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-13 22:46:50 UTC-0400	WebServerGroup	CREATE_IN_PROGRESS	-
2021-04-13 22:46:47 UTC-0400	LaunchConfig	CREATE_COMPLETE	-
2021-04-13 22:46:46 UTC-0400	LaunchConfig	CREATE_IN_PROGRESS	Resource creation Initiated
2021-04-13 22:46:46 UTC-0400	LaunchConfig	CREATE_IN_PROGRESS	-

Under resources tab I can see loadbalancer, security group, launch config and web server autoscaling is created.

The screenshot shows the AWS CloudFormation Resources page. On the left, there's a sidebar with 'Stacks (1)' containing one stack named 'LoadBalancerTemplateStack' with a status of 'CREATE_COMPLETE'. The main area is titled 'Resources (4)' and lists the following:

Logical ID	Type	Status
EC2SecurityGroup	AWS::EC2::SecurityGroup	CREATE_COMPLETE
ElasticLoadBalancer	AWS::ElasticLoadBalancing::LoadBalancer	CREATE_COMPLETE
LaunchConfig	AWS::AutoScaling::LaunchConfiguration	CREATE_COMPLETE
WebServerGroup	AWS::AutoScaling::AutoScalingGroup	CREATE_COMPLETE

Under output the deployed website is load balancer dns name

The screenshot shows the AWS CloudFormation Outputs page. On the left, there's a sidebar with 'Stacks (1)' containing one stack named 'LoadBalancerTemplateStack' with a status of 'CREATE_COMPLETE'. The main area is titled 'LoadBalancerTemplateStack' and has tabs for Stack info, Events, Resources, Outputs, Parameters, Template, and Change sets. The 'Outputs' tab is selected, showing one output:

Key	Value	Description	Export name
WebsiteURL	http://LoadBalan-ElasticL-1NDEQT7FUTL54-1891066965.us-east-1.elb.amazonaws.com	-	-

When I clicked on load balancer under resources it redirected me to load balancer where I can check the details of the newly generated load balancer.

Name	DNS name	State	VPC ID	Availability Zones	Type
LoadBal-ElasticL-1NDEQ7FUTL54	LoadBal-ElasticL-1NDEQ7FUTL54-1891066965.us-east-1.elb.amazonaws.com (A Record)	Active	vpc-2e883e53	us-east-1f, us-east-1e, ...	classic

Basic Configuration

Name	LoadBal-ElasticL-1NDEQ7FUTL54	Creation time	April 13, 2021 at 10:46:40 PM UTC-4
* DNS name	LoadBal-ElasticL-1NDEQ7FUTL54-1891066965.us-east-1.elb.amazonaws.com (A Record)	Hosted zone	Z35SXDOTRQ7X7K
Type	Classic (Migrate Now)	Status	0 of 4 instances in service
Scheme	internet-facing	VPC	vpc-2e883e53
Availability Zones	subnet-2ac46d1b - us-east-1e, subnet-73d5897d - us-east-1f.		

Auto scaling group and launch config details

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
LoadBalancerTemplateStack-WebServerGroup-P7M8ZD867HKQ	LoadBalancerTemplateStack-Laun...	4	-	4	4

Group details

Desired capacity	Auto Scaling group name LoadBalancerTemplateStack-WebServerGroup-P7M8ZD867HKQ
Minimum capacity	Date created Tue Apr 13 2021 22:46:50 GMT-0400 (Eastern Daylight Time)
Maximum capacity	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:1520710101:group:LoadBalancerTemplateStack-WebServerGroup-P7M8ZD867HKQ

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Auto Scaling groups' section. A table lists one auto scaling group:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
LoadBalancerTem...	LoadBalancerTemplateStack-Launc...	4	-	4	4

Below the table, the 'Launch configuration' details are shown:

Launch configuration LoadBalancerTemplateStack-LaunchConfig-1X18GBGJIOKSS	AMI ID ami-032930428bf1abbff	Security groups LoadBalancerTemplateStack-EC2SecurityGroup-Z16JNTRBFWOW
Instance type t2.micro	Key pair name lab3keypair	Create time Tue Apr 13 2021 22:46:46 GMT-0400 (Eastern Daylight Time)
Storage (volumes)		

At the bottom of the page, there are links for Feedback, English (US), and various AWS terms.

It created four instances and if I terminate one instance then it creates one more instance as we have defined in autoscaling group min is 4.

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed. The main area displays the 'Instances' section. A table lists four running instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
-	i-0ce55345dc2e22679	Running	t2.micro	2/2 checks passed	1 alarms +	us-east-1a
-	i-07a8c837c6a9e5285	Running	t2.micro	2/2 checks passed	1 alarms +	us-east-1e
-	i-05189dc9d34bbf5dc	Running	t2.micro	2/2 checks passed	1 alarms +	us-east-1d
-	i-02d17409590eb6aa9	Running	t2.micro	2/2 checks passed	1 alarms +	us-east-1b

At the bottom of the page, there are links for Feedback, English (US), and various AWS terms.

Screenshot of the AWS EC2 Instances page showing four running instances.

The page title is "Instances | EC2 Management Console". The URL is "console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running".

The left sidebar shows the "New EC2 Experience" and lists categories like EC2 Dashboard, Instances, and Images.

The main content area displays the "Instances (4) Info" table:

Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name
ec2-3-236-130-174.co...	3.236.130.174	-	-	enabled	LoadBalancerTemplateS...	lab3keypair
ec2-100-26-235-124.co...	100.26.235.124	-	-	enabled	LoadBalancerTemplateS...	lab3keypair
ec2-18-234-251-184.co...	18.234.251.184	-	-	enabled	LoadBalancerTemplateS...	lab3keypair
ec2-34-238-120-2.com...	34.238.120.2	-	-	enabled	LoadBalancerTemplateS...	lab3keypair

A message below the table says "Select an instance above".

At the bottom, there are links for Feedback, English (US), Privacy Policy, Terms of Use, and Cookie preferences.