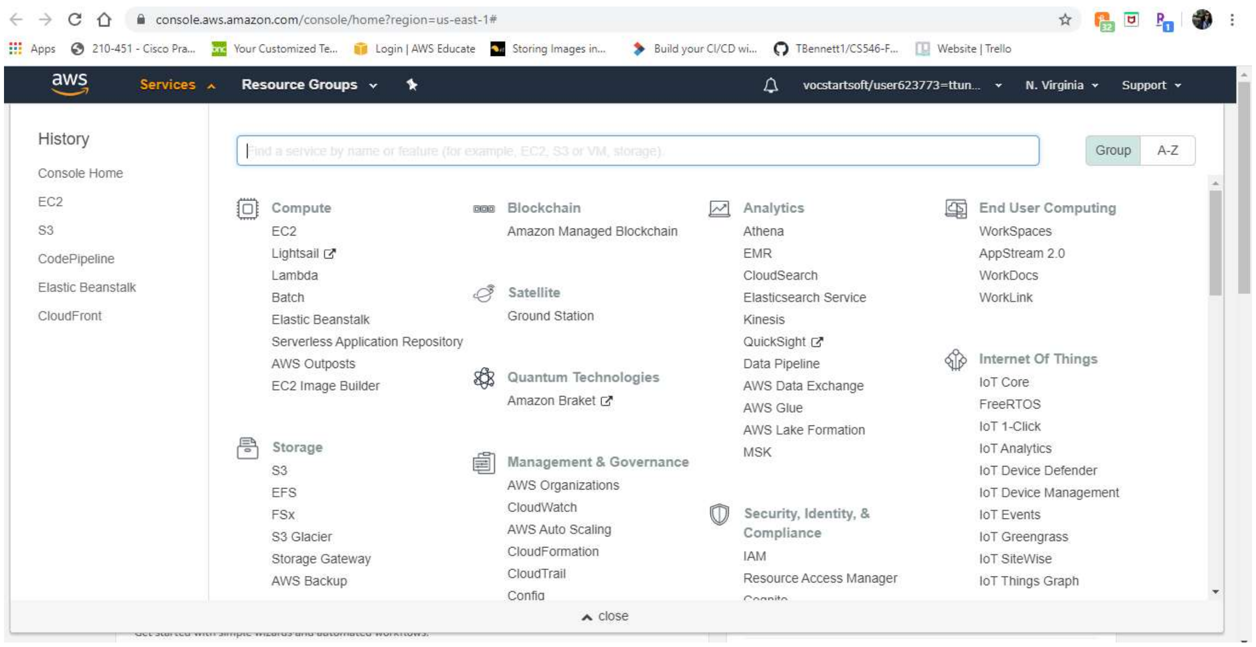
CS 524 Lab # 4

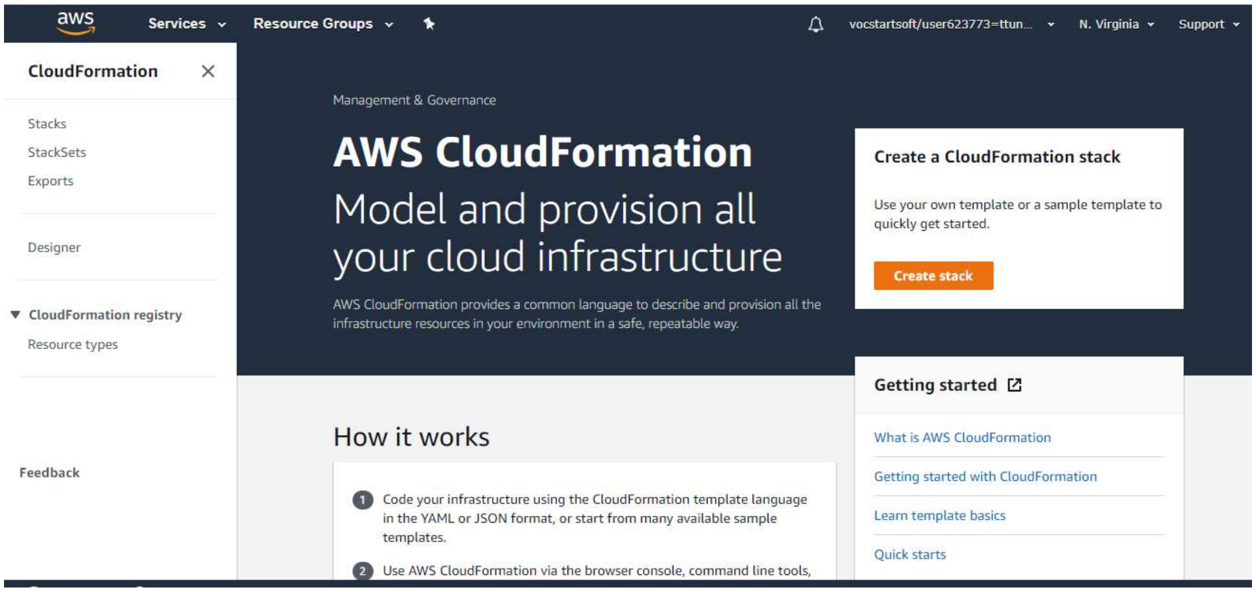
1) Creating a Stack on AWS CloudFormation:

Step 1:

Login to Amazon account and go to Services dropdown and select CloudFormation.



Select CloudFormation and you will be redirected to the page shown below:



AWS CloudFormation

Model and provision all your cloud infrastructure

AWS CloudFormation provides a common language to describe and provision all the infrastructure resources in your environment in a safe, repeatable way.

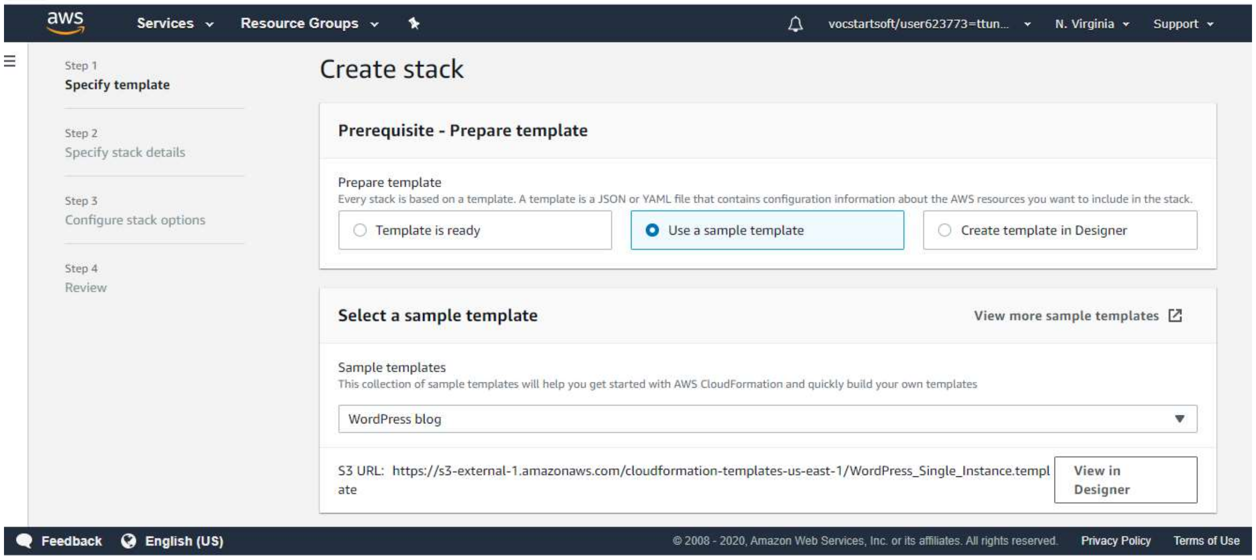
Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

How it works

1. Code your infrastructure using the CloudFormation template language in the YAML or JSON format, or start from many available sample templates.
2. Use AWS CloudFormation via the browser console, command line tools,

Click on Create stack button:


Create stack

Step 1
Specify template

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

Select a sample template

View more sample templates

Sample templates
This collection of sample templates will help you get started with AWS CloudFormation and quickly build your own templates

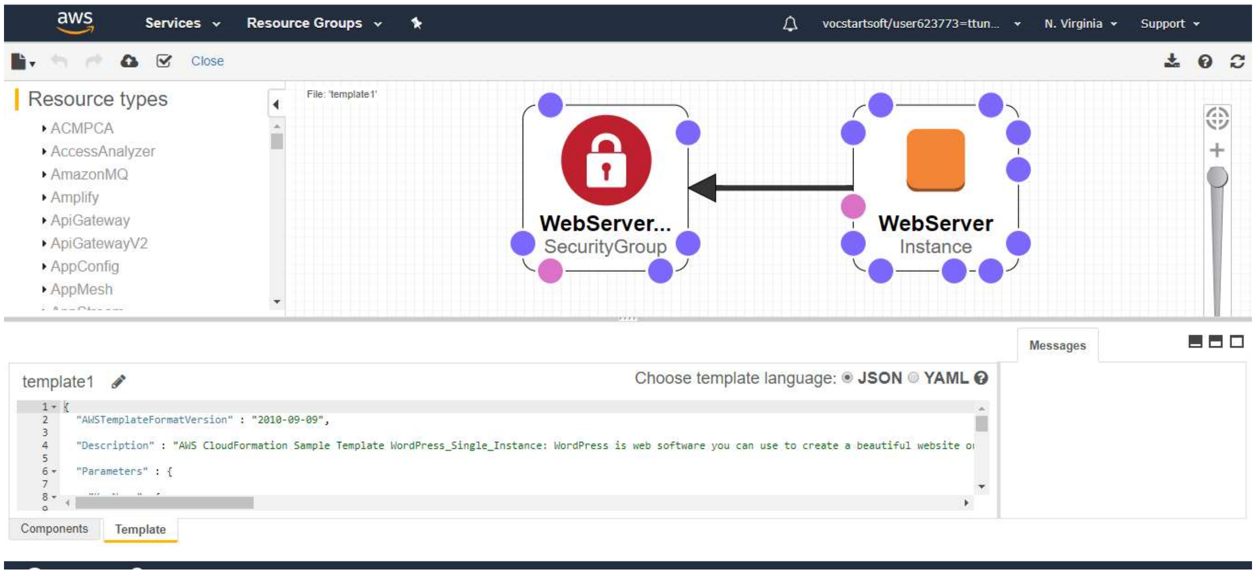
WordPress blog

S3 URL: https://s3-external-1.amazonaws.com/cloudformation-templates-us-east-1/WordPress_Single_Instance.template

View in Designer

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Select Use a sample template and then I selected a WordPress blog. If you click on View In designer button, the following page is displayed.



template1

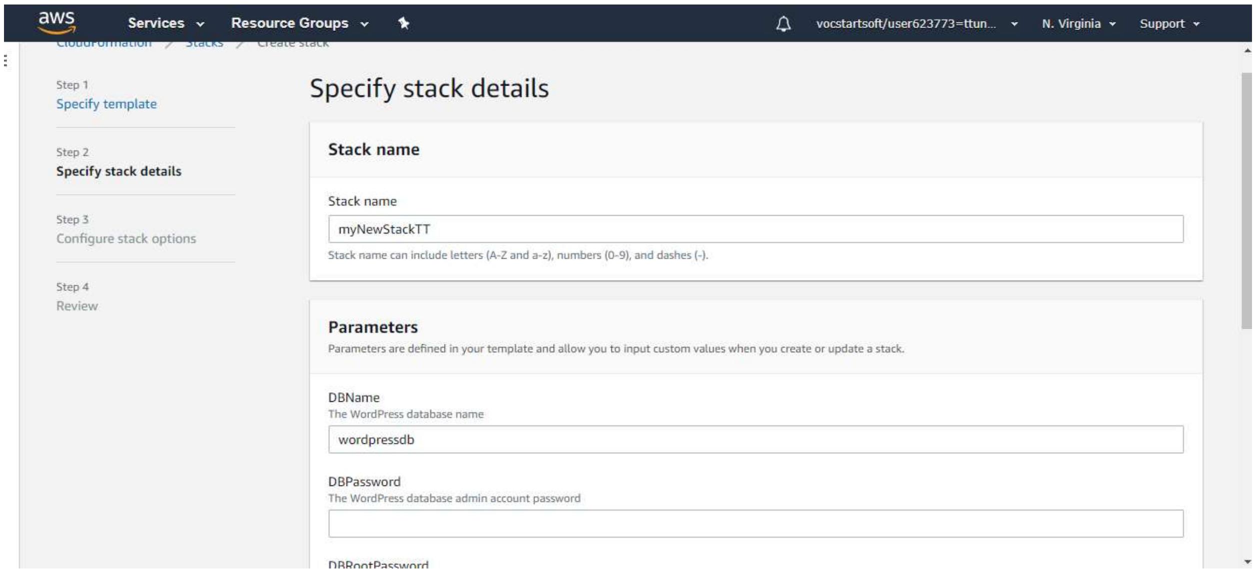
Choose template language: [✓ JSON] YAML

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website o

  "Parameters" : {
```

Go back to the creating stack page and click Next.

You will be asked for: stack name.

The stack name is an identifier that helps you find a particular stack from a list of stacks.



Specify stack details

Stack name

Stack name: myNewStackTT

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.

DBName
The WordPress database name
wordpressdb

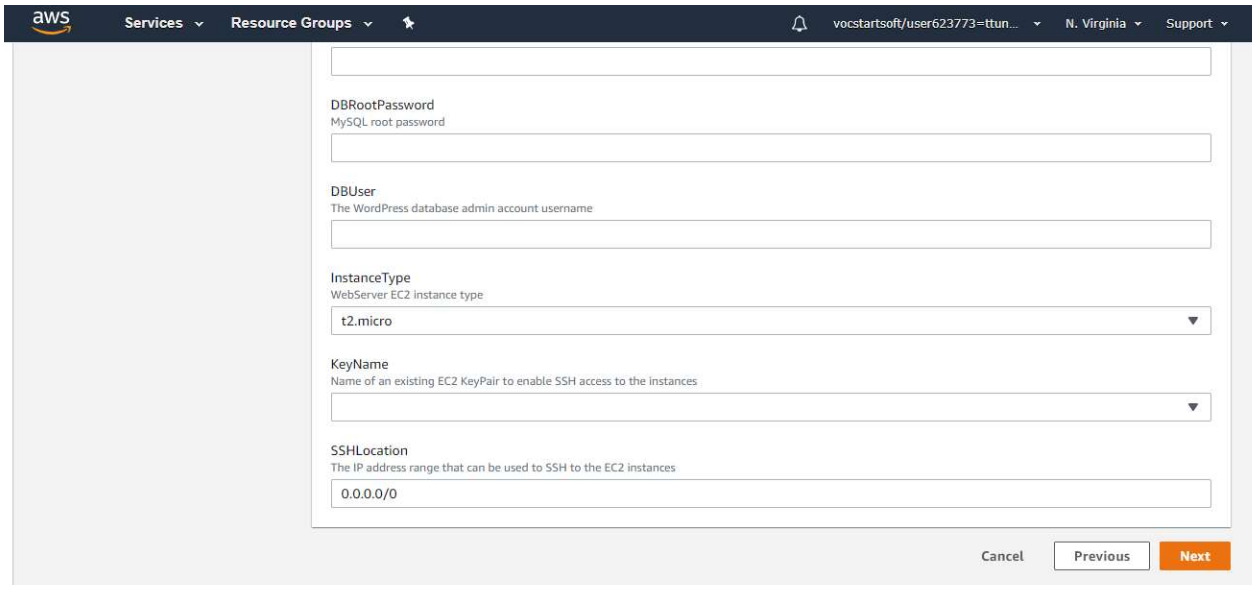
DBPassword
The WordPress database admin account password

DBRootPassword

So I have named my stack.

I did not give the database any password.

I selected t2.micro as type of instance.



DBRootPassword
MySQL root password

DBUser
The WordPress database admin account username

InstanceType
WebServer EC2 instance type
t2.micro

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

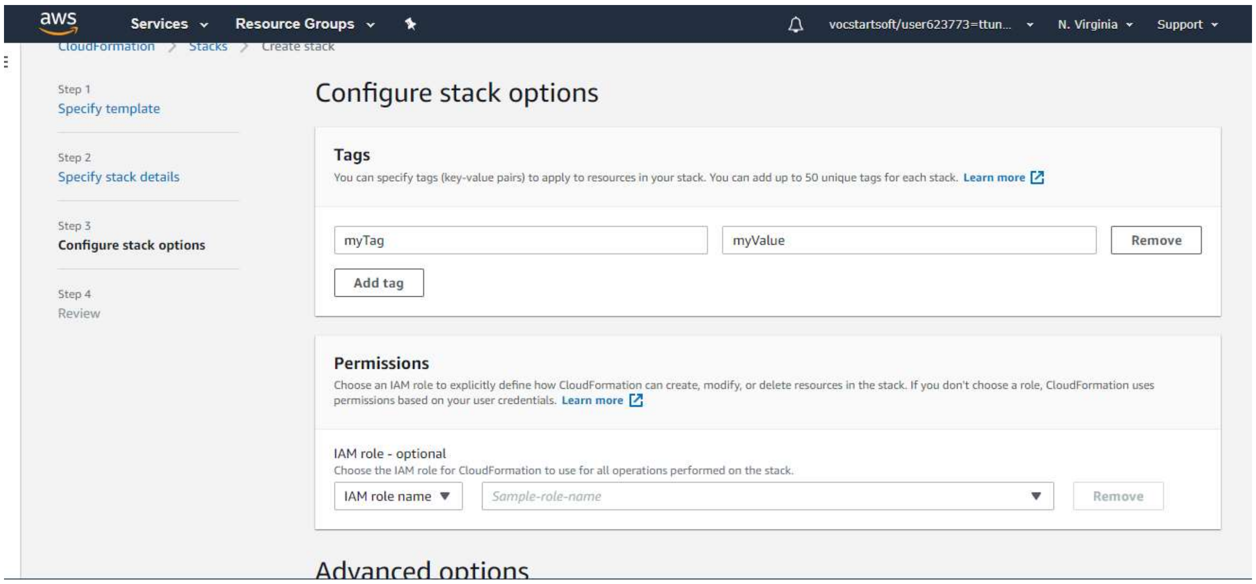
SSHLocation
The IP address range that can be used to SSH to the EC2 instances
0.0.0.0/0

Click Next.

You will be asked to specify tags, if any.

Tags are arbitrary key-value pairs that can be used to identify your stack for purposes such as cost allocation.

Permissions can also be given. An existing AWS Identity and Access Management (IAM) service role that AWS CloudFormation can assume. I will leave it as is.



Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. Learn more

myTag | myValue | Remove

Add tag

Permissions

Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. Learn more

IAM role - optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

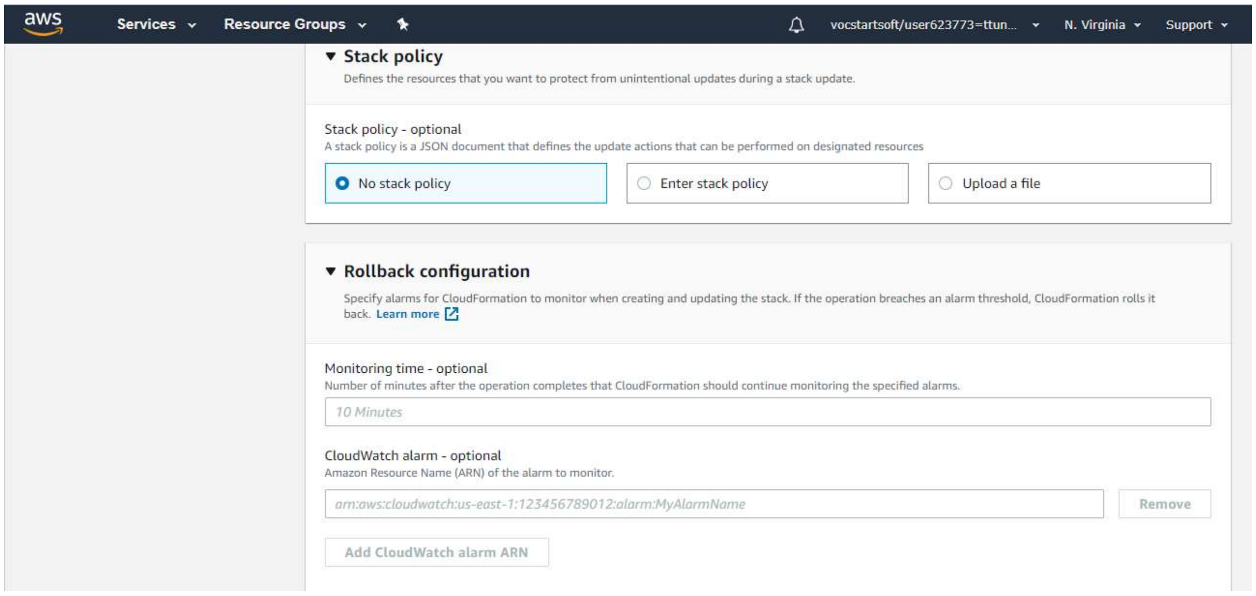
IAM role name | Sample-role-name | Remove

Advanced options

Scroll down and you will see Advanced Options.

Stack Policy: Defines the resources that you want to protect from unintentional updates during a stack update. By default, all resources can be updated during a stack update.

Rollback Configuration: Enables you to have AWS CloudFormation monitor the state of your stack during stack creation and updating, and to roll back that operation if the stack breaches the threshold of any of the alarms you've specified.



Stack policy

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

Stack policy - optional

A stack policy is a JSON document that defines the update actions that can be performed on designated resources

[✓ No stack policy] Enter stack policy Upload a file

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

Monitoring time - optional

Number of minutes after the operation completes that CloudFormation should continue monitoring the specified alarms.

10 Minutes

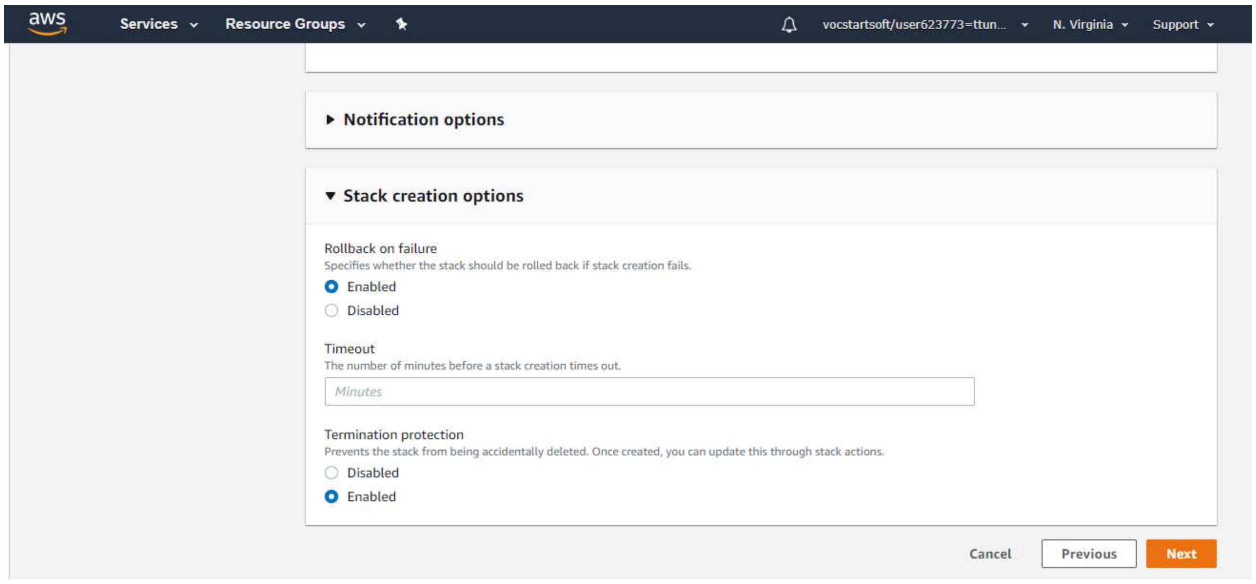
CloudWatch alarm - optional

Amazon Resource Name (ARN) of the alarm to monitor.

arn:aws:cloudwatch:us-east-1:123456789012:alarm:MyAlarmName | Remove

Add CloudWatch alarm ARN

I have kept these to default.



▸ Notification options

▾ Stack creation options

Rollback on failure
Specifies whether the stack should be rolled back if stack creation fails.
● Enabled
○ Disabled

Timeout
The number of minutes before a stack creation times out.
Minutes

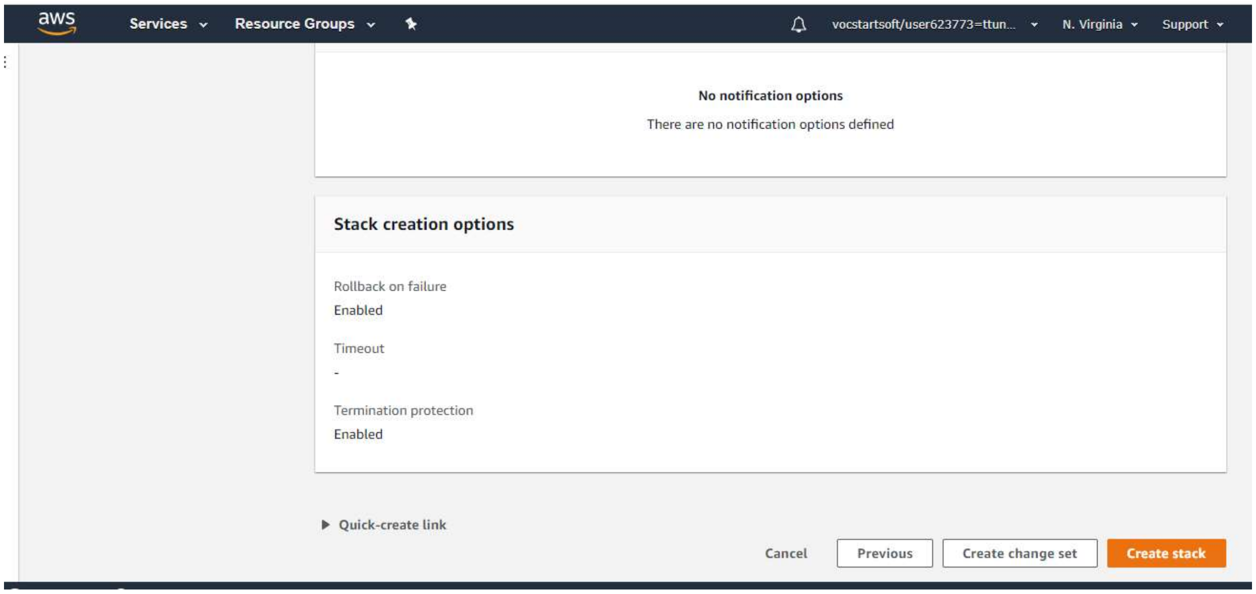
Termination protection
Prevents the stack from being accidentally deleted. Once created, you can update this through stack actions.
○ Disabled
● Enabled

Rollback on failure: Specifies whether the stack should be rolled back if stack creation fails. Typically, you want to accept the default value of Enabled. Select Disabled if you want the stack's state retained even if creation fails, such as when you are debugging a stack template.

Termination protection: Prevents a stack from being accidently deleted. By default, it is disabled. I have set it to Enabled.

Click Next.

Review all the setting and parameters and click on Create Stack.



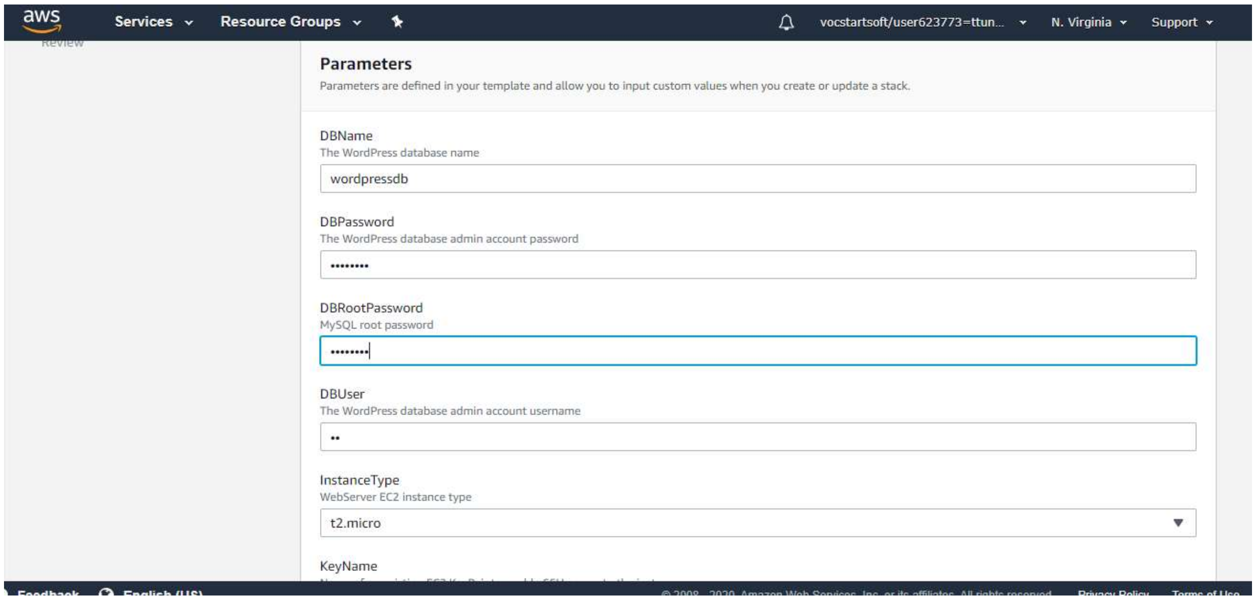
No notification options
There are no notification options defined

Stack creation options

Rollback on failure
Enabled

Timeout
-

Termination protection
Enabled



Parameters

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

DBName
The WordPress database name
wordpressdb

DBPassword
The WordPress database admin account password
••••••••

DBRootPassword
MySQL root password
••••••••

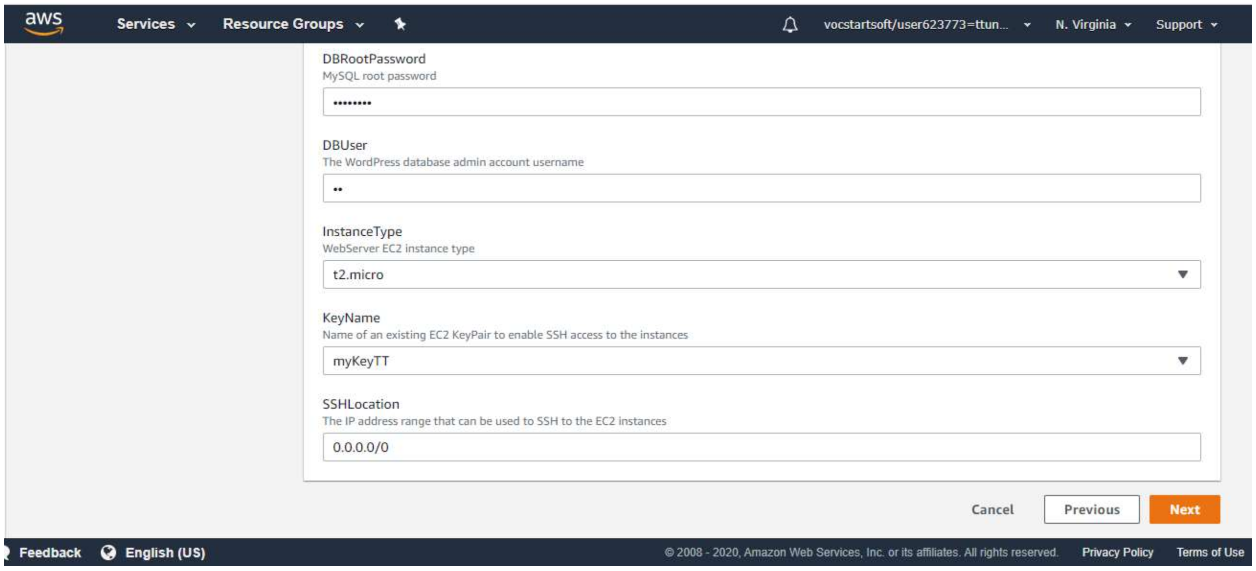
DBUser
The WordPress database admin account username
••

InstanceType
WebServer EC2 instance type
t2.micro

KeyName

So my password is abcdefgh and dbuser is tt. You have to set the dbpassword to be atleast 8 characters ; same applies to dbrootpassword. You can check these contraints are listed in the JSON file of wordpress that you created in the previous step. I had already a key named myKeyTT so I used it.

Click next and click on Create stack.


aws
Services
Resource Groups
vocstartsoft/user623773=ttun...
N. Virginia
Support
Cancel
Previous
Next
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

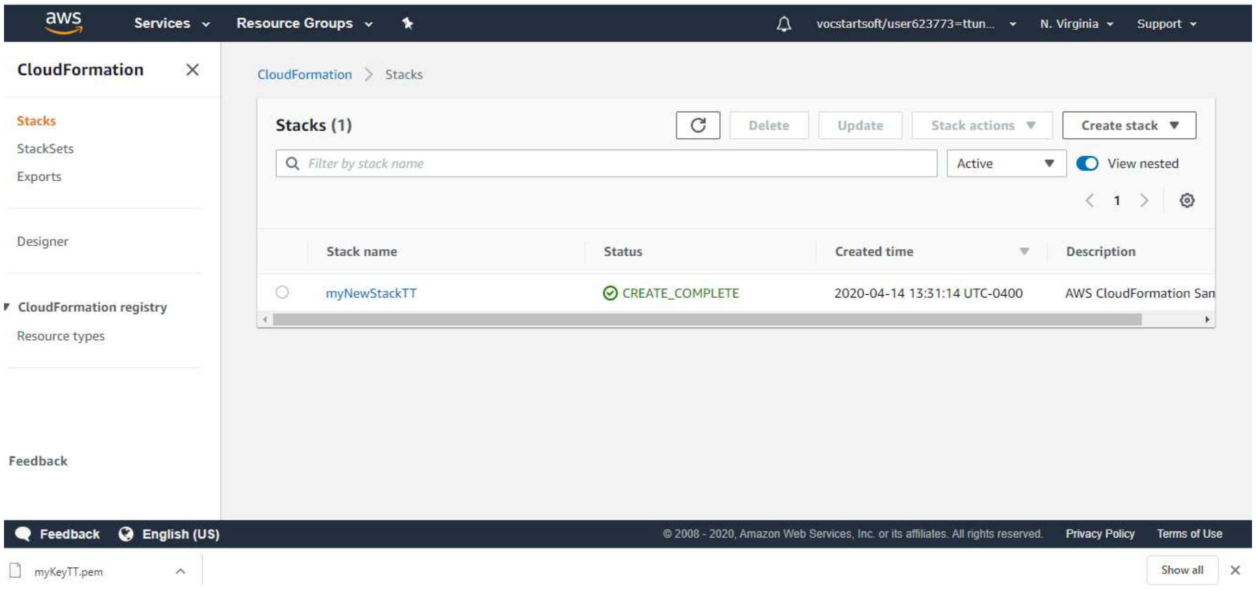
DBRootPassword
MySQL root password
••••••••

DBUser
The WordPress database admin account username
••

InstanceType
WebServer EC2 instance type
t2.micro

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

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



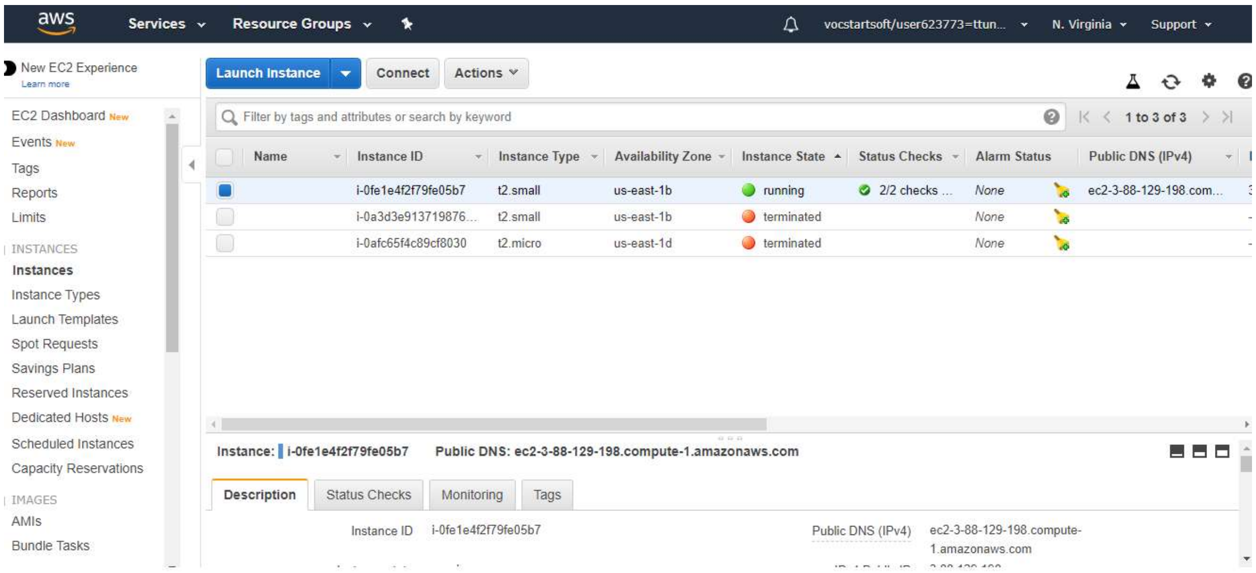
Stacks (1)

Stack name	Status	Created time	Description
myNewStackTT	CREATE_COMPLETE	2020-04-14 13:31:14 UTC-0400	AWS CloudFormation San

The stack has been created successfully!

In the EC2 dashboard, an instance has automatically been created by my new stack.

Go to EC2 Dashboard and under Instances:



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-0fe1e4f2f79fe05b7	t2.small	us-east-1b	running	2/2 checks ...	None	ec2-3-88-129-198.com...
	i-0a3d3e913719876...	t2.small	us-east-1b	terminated		None	
	i-0afc65f4c89cf8030	t2.micro	us-east-1d	terminated		None	

Instance: i-0fe1e4f2f79fe05b7 Public DNS: ec2-3-88-129-198.compute-1.amazonaws.com

Instance ID i-0fe1e4f2f79fe05b7

Public DNS (IPv4) ec2-3-88-129-198.compute-1.amazonaws.com

So I had changed my instance to t2.small. Also, my server has Amazon Linux running on it instead of Amazon Linux 2. The WordPress website uses PHP version 5.6 and higher and my server had PHP version 5.3. So I uninstalled it using $ sudo yum remove -y httpd24 php56 mysql55-server php56-mysqlnd perl-DBD-MySQL56

Then I had to install the current version of PHP using:

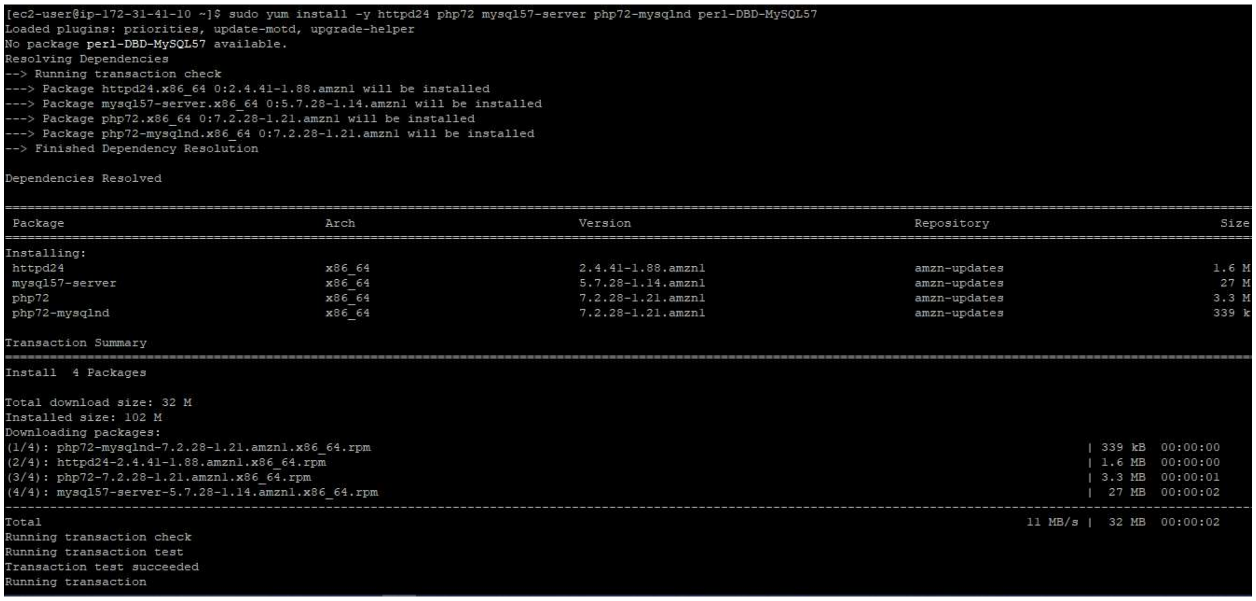
$ sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd perl-DBD-MySQL57

Then after it is successfully installed, I had to restart my httpd and mysqld services using commands:

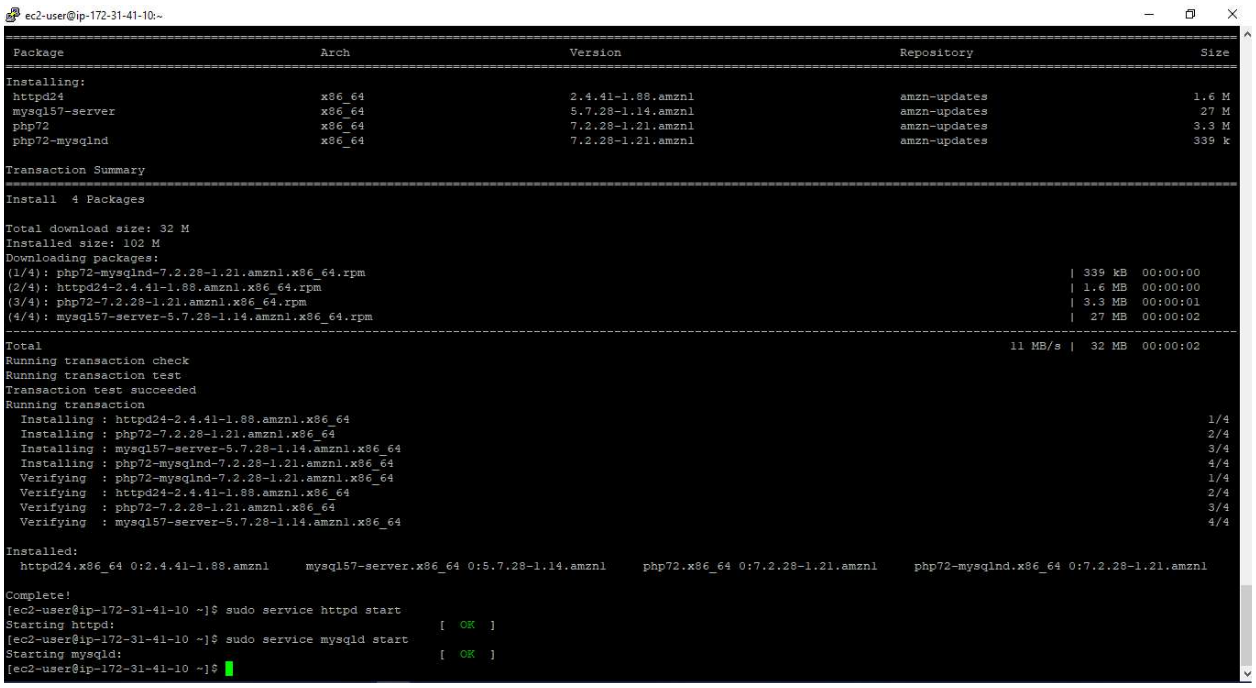
$ sudo service httpd start

$ sudo service mysqld start

After running these commands, I got the desired WordPress website running.



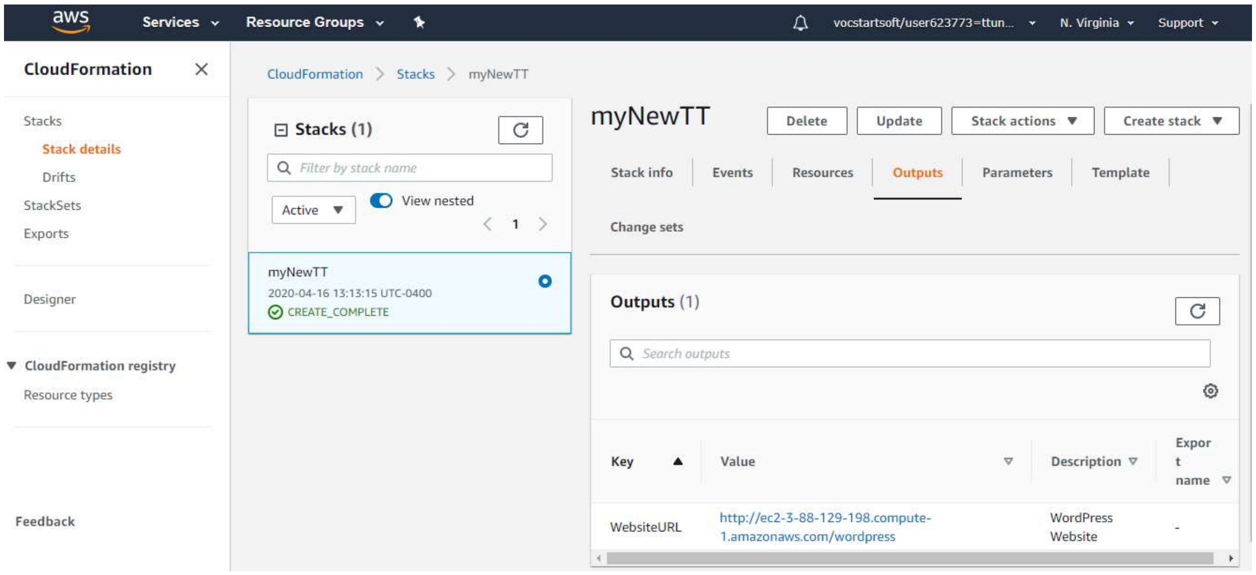
```
[ec2-user@ip-172-31-41-10 ~]$ sudo yum install -y httpd24 php72 mysql57-server php72-mysqlnd perl-DBD-MySQL57
Loaded plugins: priorities, update-motd, upgrade-helper
No package perl-DBD-MySQL57 available.
Resolving Dependencies
--> Running transaction check
---> Package httpd24.x86_64 0:2.4.41-1.88.amzn1 will be installed
---> Package mysql57-server.x86_64 0:5.7.28-1.14.amzn1 will be installed
---> Package php72.x86_64 0:7.2.28-1.21.amzn1 will be installed
---> Package php72-mysqlnd.x86_64 0:7.2.28-1.21.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                     Arch                      Version                          Repository                         Size
================================================================================================================================
Installing:
 httpd24                     x86_64                    2.4.41-1.88.amzn1                amzn-updates                       1.6 M
 mysql57-server              x86_64                    5.7.28-1.14.amzn1                amzn-updates                        27 M
 php72                       x86_64                    7.2.28-1.21.amzn1                amzn-updates                       3.3 M
 php72-mysqlnd               x86_64                    7.2.28-1.21.amzn1                amzn-updates                       339 k

Transaction Summary
================================================================================================================================
Install  4 Packages

Total download size: 32 M
Installed size: 102 M
Downloading packages:
(1/4): php72-mysqlnd-7.2.28-1.21.amzn1.x86_64.rpm                                                    | 339 kB  00:00:00
(2/4): httpd24-2.4.41-1.88.amzn1.x86_64.rpm                                                          | 1.6 MB  00:00:00
(3/4): php72-7.2.28-1.21.amzn1.x86_64.rpm                                                            | 3.3 MB  00:00:01
(4/4): mysql57-server-5.7.28-1.14.amzn1.x86_64.rpm                                                   |  27 MB  00:00:02
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                        11 MB/s |  32 MB  00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```



ec2-user@ip-172-31-41-10:~

```
================================================================================================================================
 Package                     Arch                      Version                          Repository                         Size
================================================================================================================================
Installing:
 httpd24                     x86_64                    2.4.41-1.88.amzn1                amzn-updates                       1.6 M
 mysql57-server              x86_64                    5.7.28-1.14.amzn1                amzn-updates                        27 M
 php72                       x86_64                    7.2.28-1.21.amzn1                amzn-updates                       3.3 M
 php72-mysqlnd               x86_64                    7.2.28-1.21.amzn1                amzn-updates                       339 k

Transaction Summary
================================================================================================================================
Install  4 Packages

Total download size: 32 M
Installed size: 102 M
Downloading packages:
(1/4): php72-mysqlnd-7.2.28-1.21.amzn1.x86_64.rpm                                                    | 339 kB  00:00:00
(2/4): httpd24-2.4.41-1.88.amzn1.x86_64.rpm                                                          | 1.6 MB  00:00:00
(3/4): php72-7.2.28-1.21.amzn1.x86_64.rpm                                                            | 3.3 MB  00:00:01
(4/4): mysql57-server-5.7.28-1.14.amzn1.x86_64.rpm                                                   |  27 MB  00:00:02
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                                        11 MB/s |  32 MB  00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : httpd24-2.4.41-1.88.amzn1.x86_64                                                                             1/4
  Installing : php72-7.2.28-1.21.amzn1.x86_64                                                                               2/4
  Installing : mysql57-server-5.7.28-1.14.amzn1.x86_64                                                                      3/4
  Installing : php72-mysqlnd-7.2.28-1.21.amzn1.x86_64                                                                       4/4
  Verifying  : php72-mysqlnd-7.2.28-1.21.amzn1.x86_64                                                                       1/4
  Verifying  : httpd24-2.4.41-1.88.amzn1.x86_64                                                                             2/4
  Verifying  : php72-7.2.28-1.21.amzn1.x86_64                                                                               3/4
  Verifying  : mysql57-server-5.7.28-1.14.amzn1.x86_64                                                                      4/4

Installed:
  httpd24.x86_64 0:2.4.41-1.88.amzn1     mysql57-server.x86_64 0:5.7.28-1.14.amzn1     php72.x86_64 0:7.2.28-1.21.amzn1     php72-mysqlnd.x86_64 0:7.2.28-1.21.amzn1

Complete!
[ec2-user@ip-172-31-41-10 ~]$ sudo service httpd start
Starting httpd:                                            [  OK  ]
[ec2-user@ip-172-31-41-10 ~]$ sudo service mysqld start
Starting mysqld:                                           [  OK  ]
[ec2-user@ip-172-31-41-10 ~]$ 
```

Now accessing the WordPress website from Stacks>Click on the name of your stack and navigate to Outputs tab:

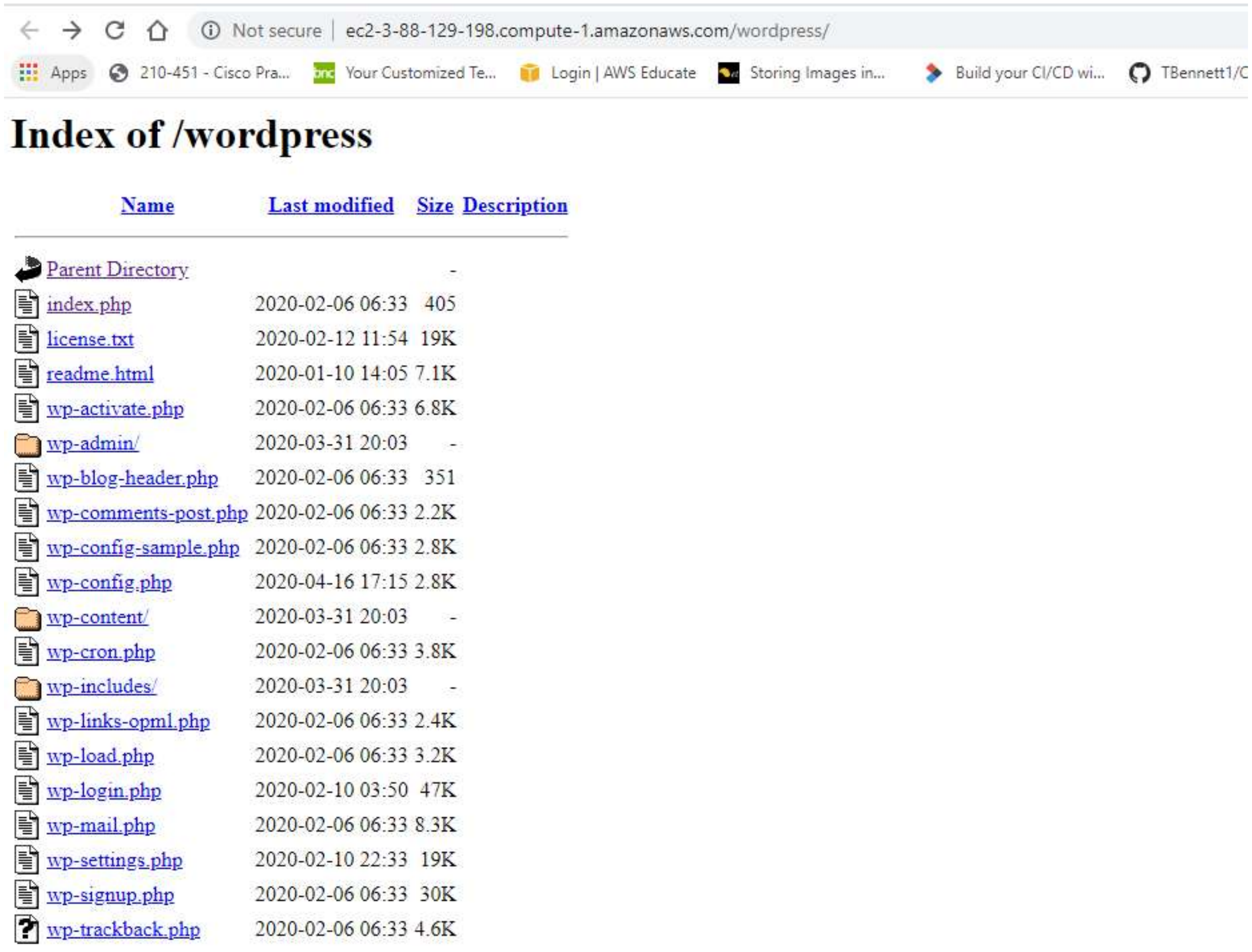


myNewTT
2020-04-16 13:13:15 UTC-0400
CREATE_COMPLETE

Outputs (1)

Key	Value	Description	Export name
WebsiteURL	http://ec2-3-88-129-198.compute-1.amazonaws.com/wordpress	WordPress Website	-

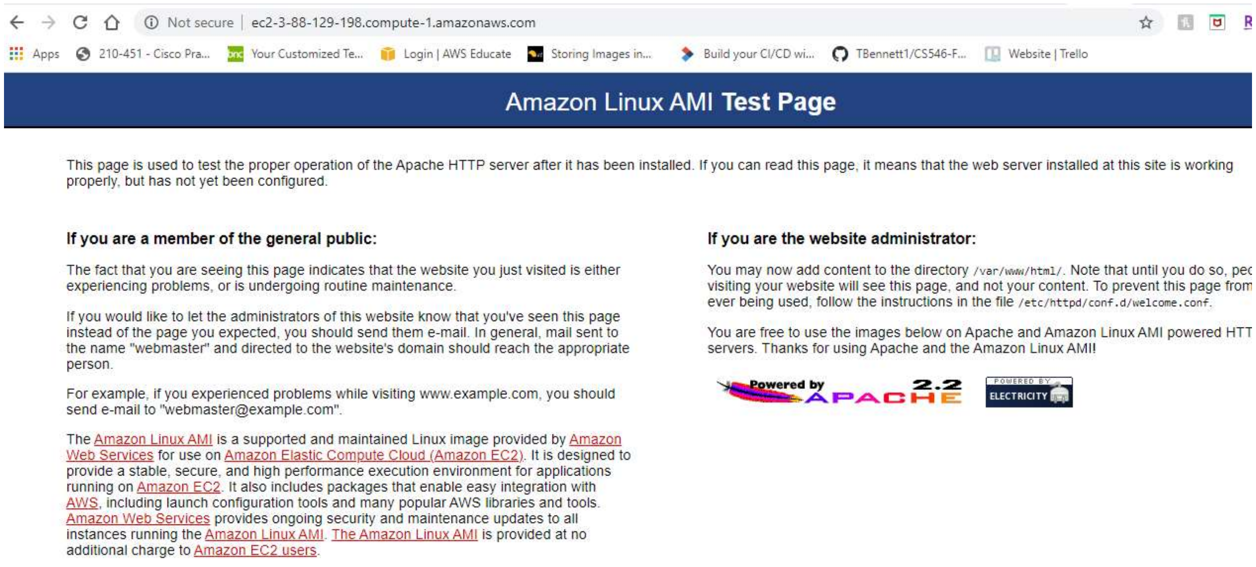
Copy the Website URL and input it in the browser:



Index of /wordpress

Name	Last modified	Size	Description
Parent Directory		-	
index.php	2020-02-06 06:33	405	
license.txt	2020-02-12 11:54	19K	
readme.html	2020-01-10 14:05	7.1K	
wp-activate.php	2020-02-06 06:33	6.8K	
wp-admin/	2020-03-31 20:03	-	
wp-blog-header.php	2020-02-06 06:33	351	
wp-comments-post.php	2020-02-06 06:33	2.2K	
wp-config-sample.php	2020-02-06 06:33	2.8K	
wp-config.php	2020-04-16 17:15	2.8K	
wp-content/	2020-03-31 20:03	-	
wp-cron.php	2020-02-06 06:33	3.8K	
wp-includes/	2020-03-31 20:03	-	
wp-links-opml.php	2020-02-06 06:33	2.4K	
wp-load.php	2020-02-06 06:33	3.2K	
wp-login.php	2020-02-10 03:50	47K	
wp-mail.php	2020-02-06 06:33	8.3K	
wp-settings.php	2020-02-10 22:33	19K	
wp-signup.php	2020-02-06 06:33	30K	
wp-trackback.php	2020-02-06 06:33	4.6K	

In your instances, if you put the public DNS in the browser, you should get the following output:


Not secure | ec2-3-88-129-198.compute-1.amazonaws.com

Apps 210-451 - Cisco Pra... Your Customized Te... Login | AWS Educate Storing Images in... Build your CI/CD wi... TBennett1/CS546-F... Website | Trello

Amazon Linux AMI **Test Page**

This page is used to test the proper operation of the Apache 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, but has not yet been configured.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

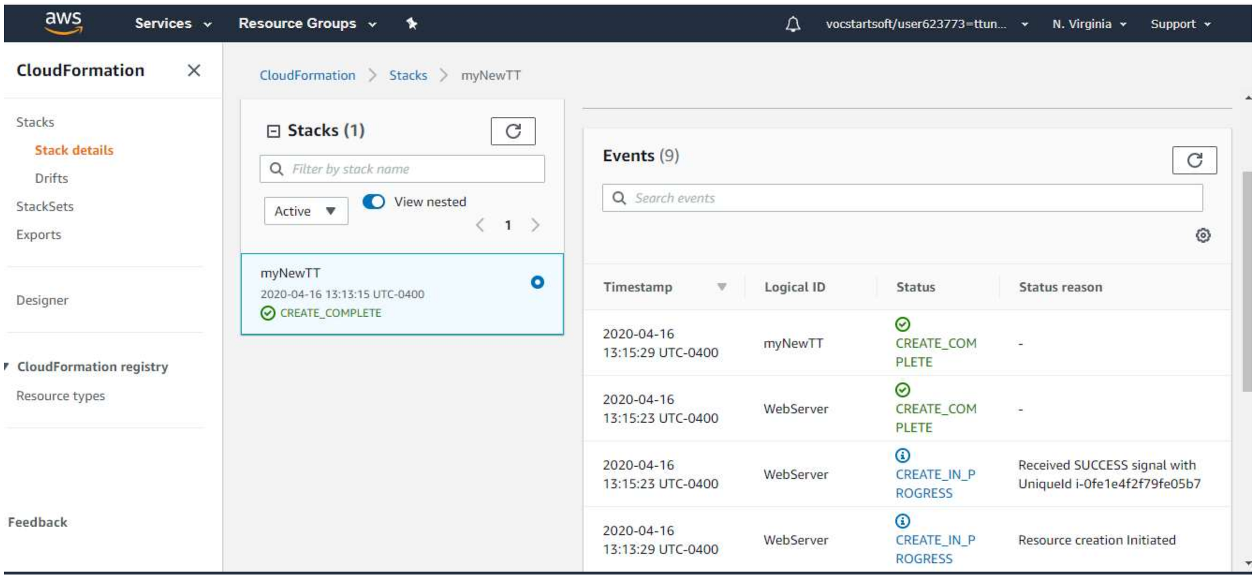
The Amazon Linux AMI is a supported and maintained Linux image provided by Amazon Web Services for use on Amazon Elastic Compute Cloud (Amazon EC2). It is designed to provide a stable, secure, and high performance execution environment for applications running on Amazon EC2. It also includes packages that enable easy integration with AWS, including launch configuration tools and many popular AWS libraries and tools. Amazon Web Services provides ongoing security and maintenance updates to all instances running the Amazon Linux AMI. The Amazon Linux AMI is provided at no additional charge to Amazon EC2 users.

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, peo visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the images below on Apache and Amazon Linux AMI powered HTT servers. Thanks for using Apache and the Amazon Linux AMI!

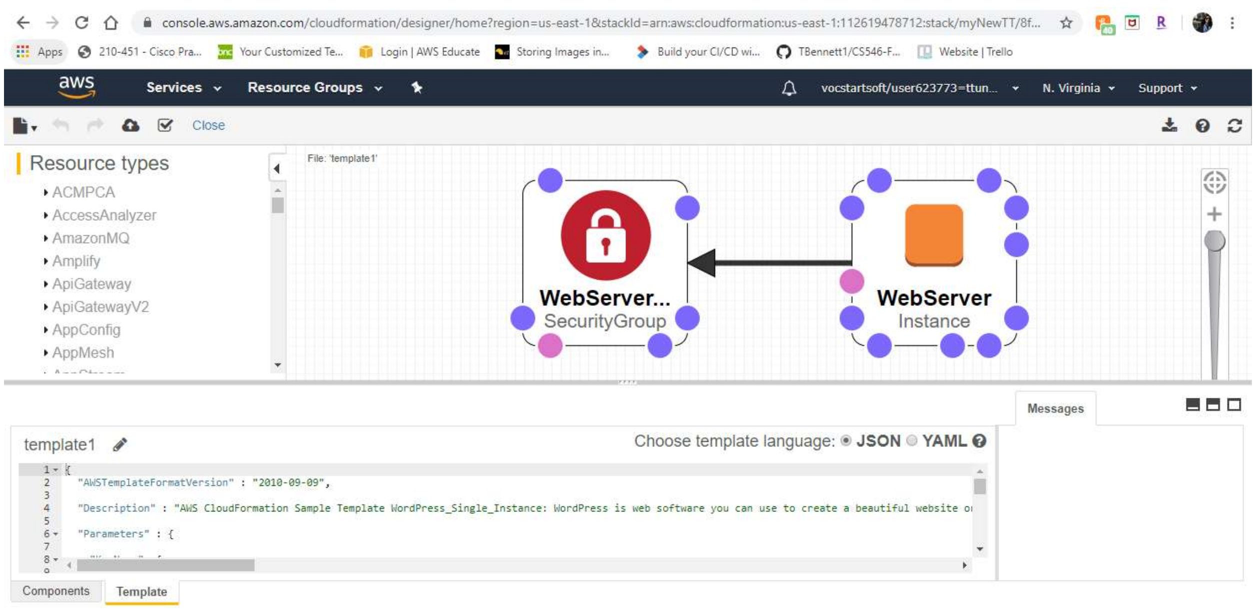
You will not get this page if httpd service is not running. Also, if the mysqld service is not running, your website will give you an error: Error connecting to database. So make sure to start these two services.



myNewTT
2020-04-16 13:13:15 UTC-0400
CREATE_COMPLETE

Timestamp	Logical ID	Status	Status reason
2020-04-16 13:15:29 UTC-0400	myNewTT	CREATE_COMPLETE	-
2020-04-16 13:15:23 UTC-0400	WebServer	CREATE_COMPLETE	-
2020-04-16 13:15:23 UTC-0400	WebServer	CREATE_IN_PROGRESS	Received SUCCESS signal with UniqueId i-0fe1e4f2f79fe05b7
2020-04-16 13:13:29 UTC-0400	WebServer	CREATE_IN_PROGRESS	Resource creation Initiated

If you go in Designer View you will see the template as:



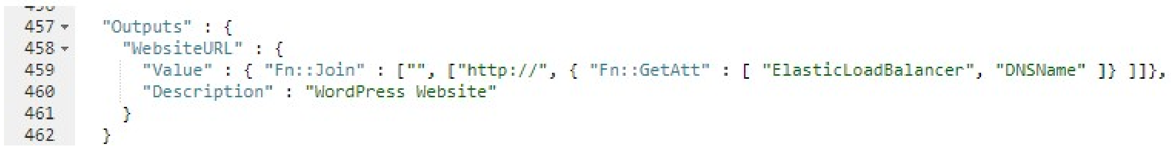
```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website o

  "Parameters" : {
```

The template has to be modified:

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website or blog. This
template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. **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.",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
    },

    "InstanceType" : {
      "Description" : "WebServer EC2 instance type",
      "Type" : "String",
      "Default" : "t2.small",
      "AllowedValues" : [ "t1.micro", "t2.nano", "t2.micro", "t2.small", "t2.medium", "t2.large", "m1.small", "m1.medium", "m1.large", "m1.xlarge", "m2.xlarge",
"m2.2xlarge", "m2.4xlarge", "m3.medium", "m3.large", "m3.xlarge", "m3.2xlarge", "m4.large", "m4.xlarge", "m4.2xlarge", "m4.4xlarge", "m4.10xlarge", "c1.medium",
"c1.xlarge", "c3.large", "c3.xlarge", "c3.2xlarge", "c3.4xlarge", "c3.8xlarge", "c4.large", "c4.xlarge", "c4.2xlarge", "c4.4xlarge", "c4.8xlarge", "g2.2xlarge",
"g2.8xlarge", "r3.large", "r3.xlarge", "r3.2xlarge", "r3.4xlarge", "r3.8xlarge", "i2.xlarge", "i2.2xlarge", "i2.4xlarge", "i2.8xlarge", "d2.xlarge", "d2.2xlarge",
"d2.4xlarge", "d2.8xlarge", "hi1.4xlarge", "hs1.8xlarge", "cr1.8xlarge", "cc2.8xlarge", "cg1.4xlarge"]
,
      "ConstraintDescription" : "must be a valid EC2 instance type."
    },

    "SSHLocation": {
      "Description": "The IP address range that can be used to SSH to the EC2 instances",
      "Type": "String",
      "MinLength": "9",
      "MaxLength": "18",
      "Default": "0.0.0.0/0",
      "AllowedPattern": "(\\d{1,3})\\.(\\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/x."
    },
```

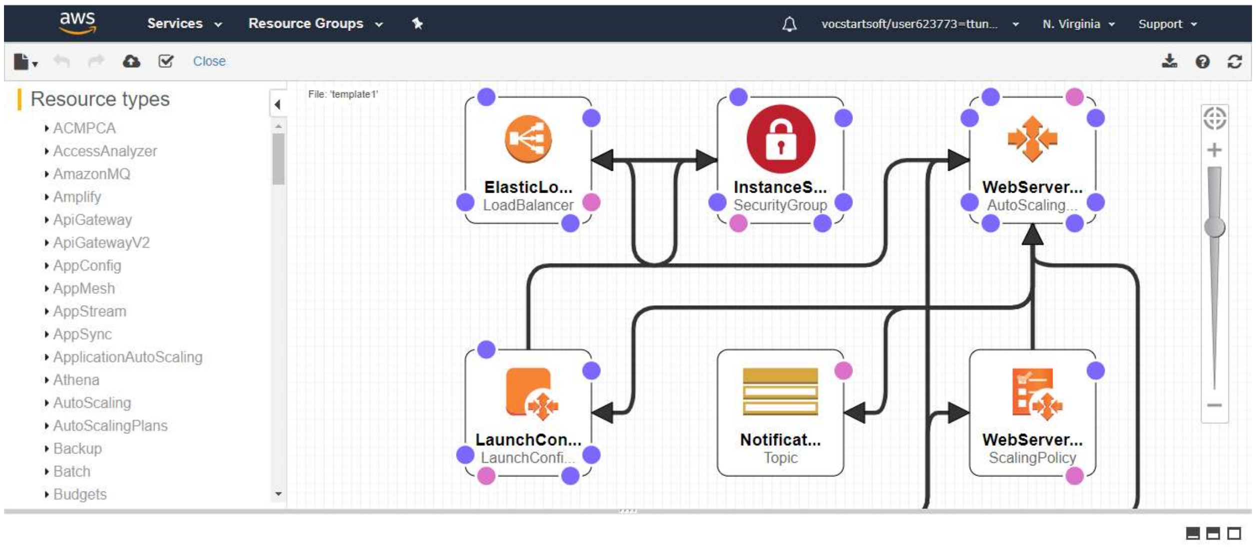
I added the following code in this template:

```
"WebServerGroup" : {
      "Type" : "AWS::AutoScaling::AutoScalingGroup",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : ""},
        "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
        "MinSize" : "1",
        "MaxSize" : "3",
        "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ],
        "NotificationConfiguration" : {
          "TopicARN" : { "Ref" : "NotificationTopic" },
          "NotificationTypes" : [ "autoscaling:EC2_INSTANCE_LAUNCH",
                                  "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
                                  "autoscaling:EC2_INSTANCE_TERMINATE",
                                  "autoscaling:EC2_INSTANCE_TERMINATE_ERROR"]
        }
      },


    "LaunchConfig" : {
      "Type" : "AWS::AutoScaling::LaunchConfiguration",
      "Metadata" : {
        "Comment" : "Install a simple application",
        "AWS::CloudFormation::Init" : {
          "config" : {
            "packages" : {
              "yum" : {
                "httpd" : []
              }
            },

            "files" : {
              "/var/www/html/index.html" : {
                "content" : { "Fn::Join" : ["\n", [
                  "<img src=\"", {"Fn::FindInMap" : ["Region2Examples",
{"Ref" : "AWS::Region"}, "Examples"]}, "/cloudformation_graphic.png\"
alt=\"AWS CloudFormation Logo\"/>",
                  "<h1>Congratulations, you have successfully launched the
AWS CloudFormation sample.</h1>"
                ]]},
                "mode"    : "000644",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/cfn-hup.conf" : {
                "content" : { "Fn::Join" : ["", [
                  "[main]\n",
                  "stack=", { "Ref" : "AWS::StackId" }, "\n",
                  "region=", { "Ref" : "AWS::Region" }, "\n"
                ]]},
                "mode"    : "000400",
                "owner"   : "root",
                "group"   : "root"
              },

              "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
                "content": { "Fn::Join" : ["", [
                  "[cfn-auto-reloader-hook]\n",
                  "triggers=post.update\n",
"path=Resources.LaunchConfig.Metadata.AWS::CloudFormation::Init\n",
                  "action=/opt/aws/bin/cfn-init -v ",
                  "         --stack ", { "Ref" : "AWS::StackName" },
                  "         --resource LaunchConfig ",
                  "         --region ", { "Ref" : "AWS::Region" }, "\n",
                  "runas=root\n"
                ]]}
              }
            },

            "services" : {
              "sysvinit" : {
                "httpd"    : { "enabled" : "true", "ensureRunning" : "true"
},
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                              "files" : ["/etc/cfn/cfn-hup.conf",
"/etc/cfn/hooks.d/cfn-auto-reloader.conf"]}
              }
            }
          }
        }
      },



"ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "CrossZone" : "true",
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : "80",
          "Protocol" : "HTTP"
        } ],
        "HealthCheck" : {
          "Target" : "HTTP:80/",
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
      }
    },
```

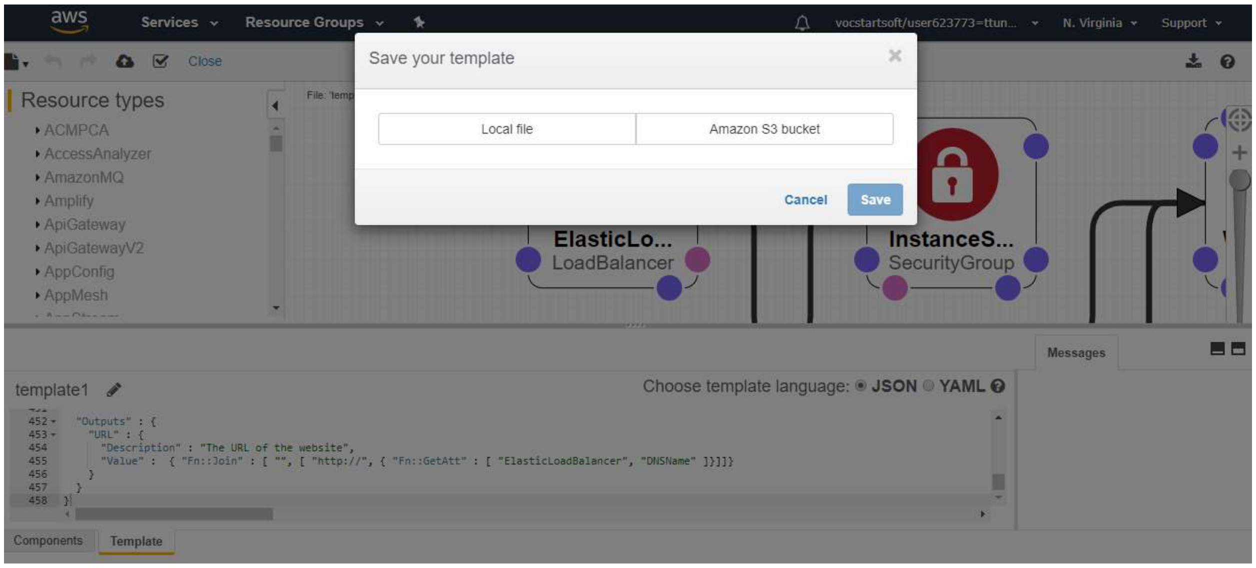


```
"Outputs" : {
  "WebsiteURL" : {
    "Value" : { "Fn::Join" : ["", ["http://", { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]} ]]},
    "Description" : "WordPress Website"
  }
}
```

The design looks like this:



Save it on S3 bucket or as a local .json file.

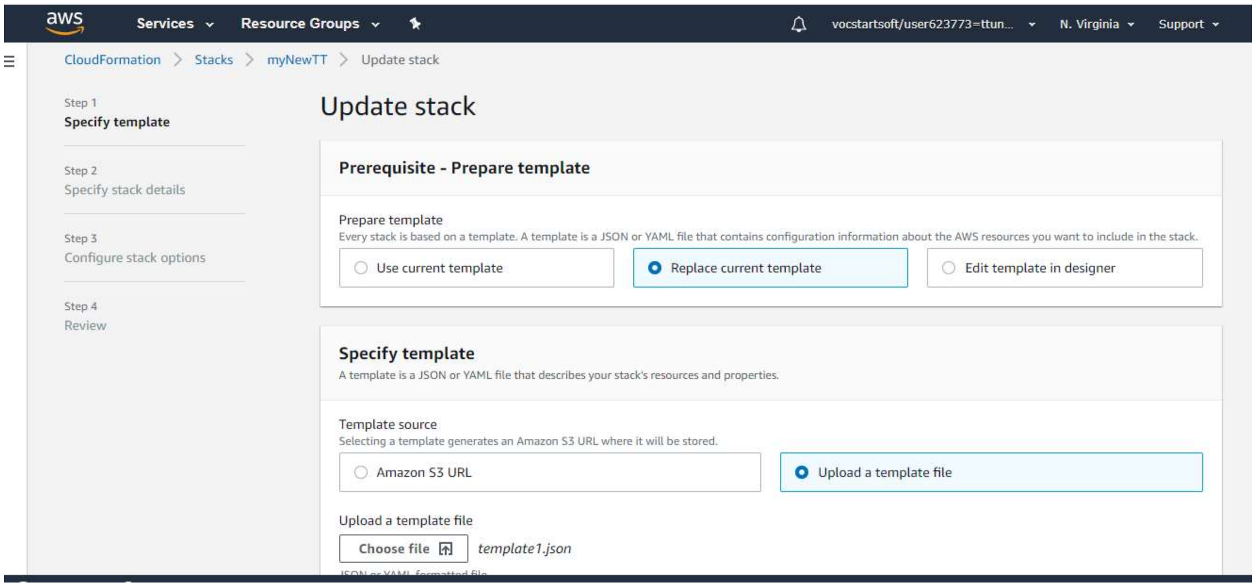


Choose template language: [✓ JSON] YAML

```
"Outputs" : {
  "URL" : {
    "Description" : "The URL of the website",
    "Value" : { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [ "ElasticLoadBalancer", "DNSName" ]}]]}
  }
}
}
```

Now go to Services> CloudFormation> Stacks and click on the stack you created.

Click on Update button and replace the template by uploading the newly created template:



Update stack

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

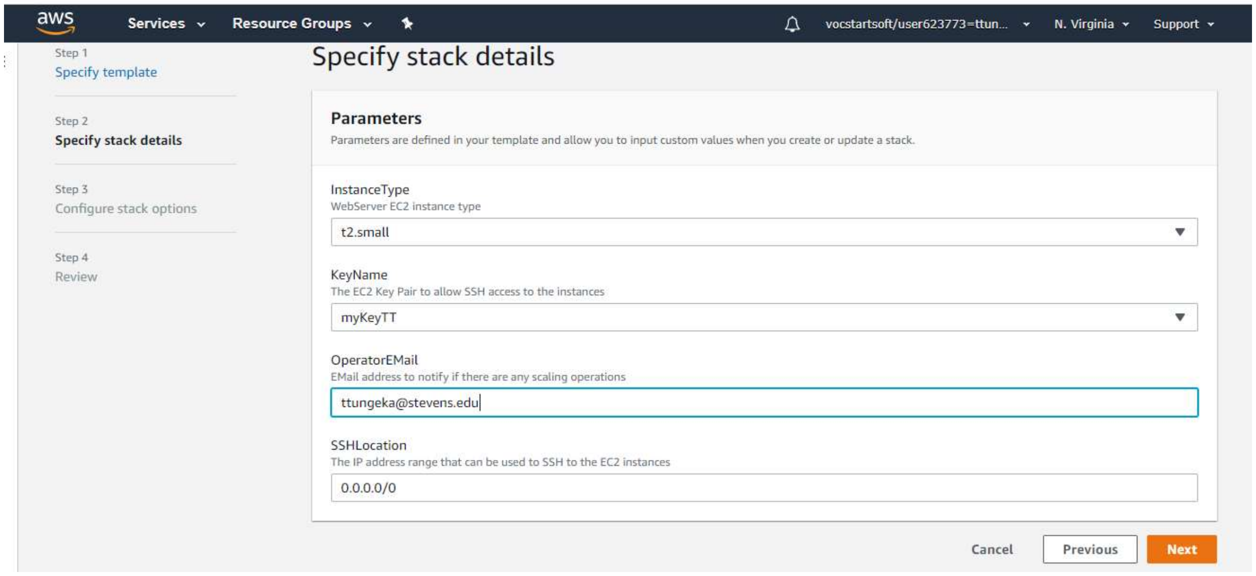
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL [✓ Upload a template file]

Upload a template file

Choose file *template1.json*

You need to provide an email id after clicking next:



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.small

KeyName

The EC2 Key Pair to allow SSH access to the instances

myKeyTT

OperatorEMail

EMail address to notify if there are any scaling operations

ttungeka@stevens.edu

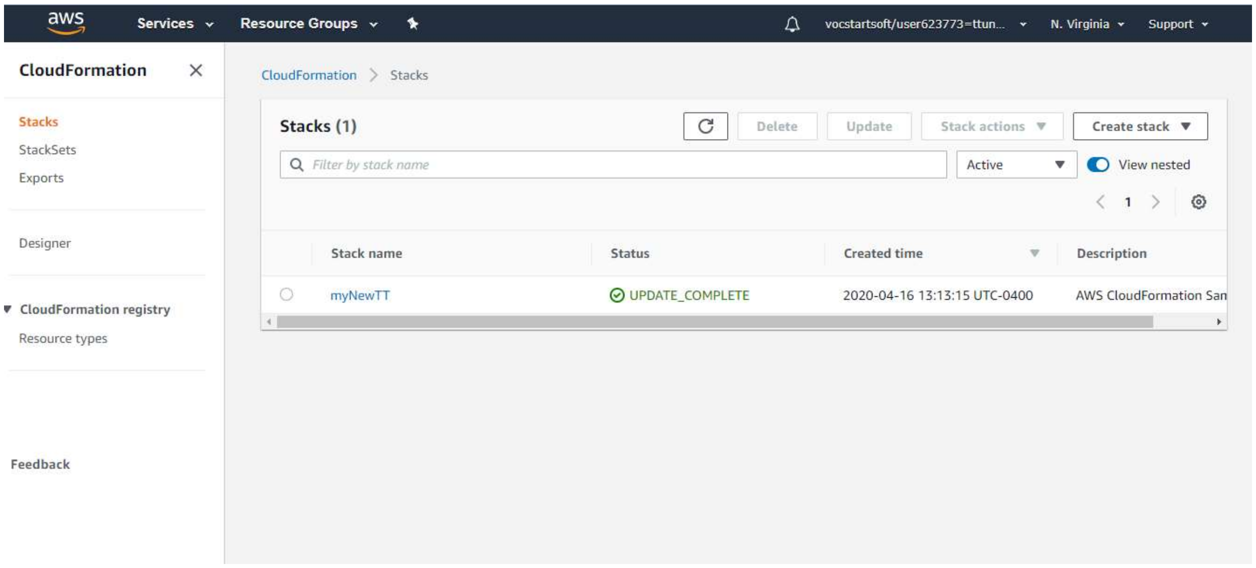
SSHLocation

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

0.0.0.0/0

Click on Next and then select Update Stack.

Since you have provided your email address as the operator email address, you will receive a confirmation when the stack is autoscaled.


Services | Resource Groups | vocstartsoft/user623773=ttun... | N. Virginia | Support

CloudFormation > Stacks

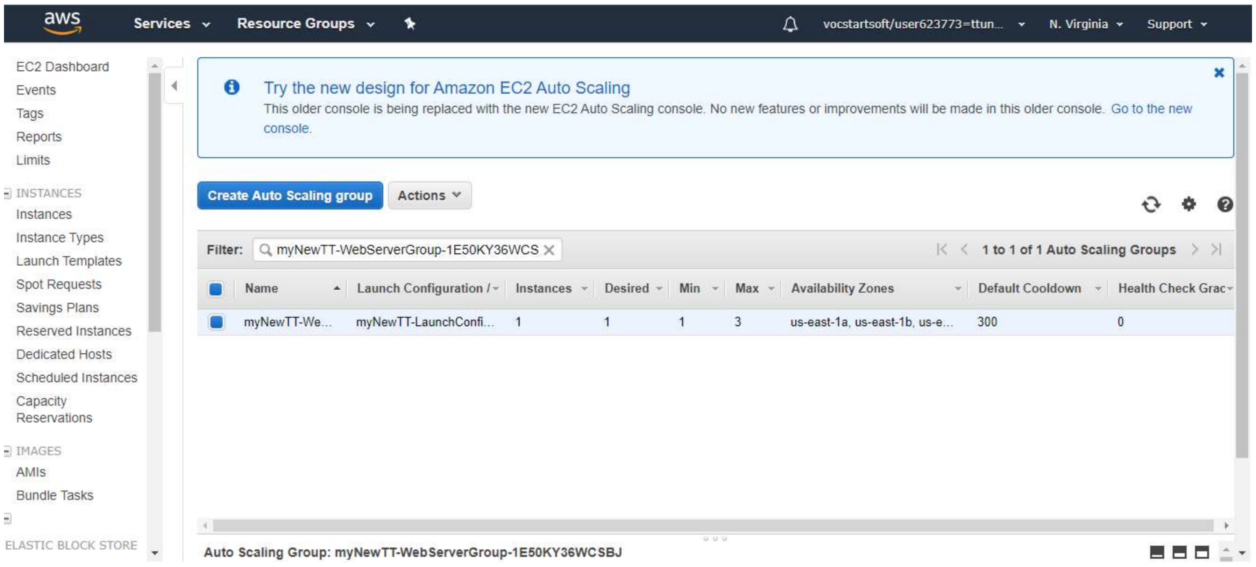
Stacks (1)

Filter by stack name | Active | View nested

Stack name	Status	Created time	Description
myNewTT	UPDATE_COMPLETE	2020-04-16 13:13:15 UTC-0400	AWS CloudFormation San

The stack has been updated successfully!

The Auto-Scaling group looks like this:


Try the new design for Amazon EC2 Auto Scaling

This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. Go to the new console.

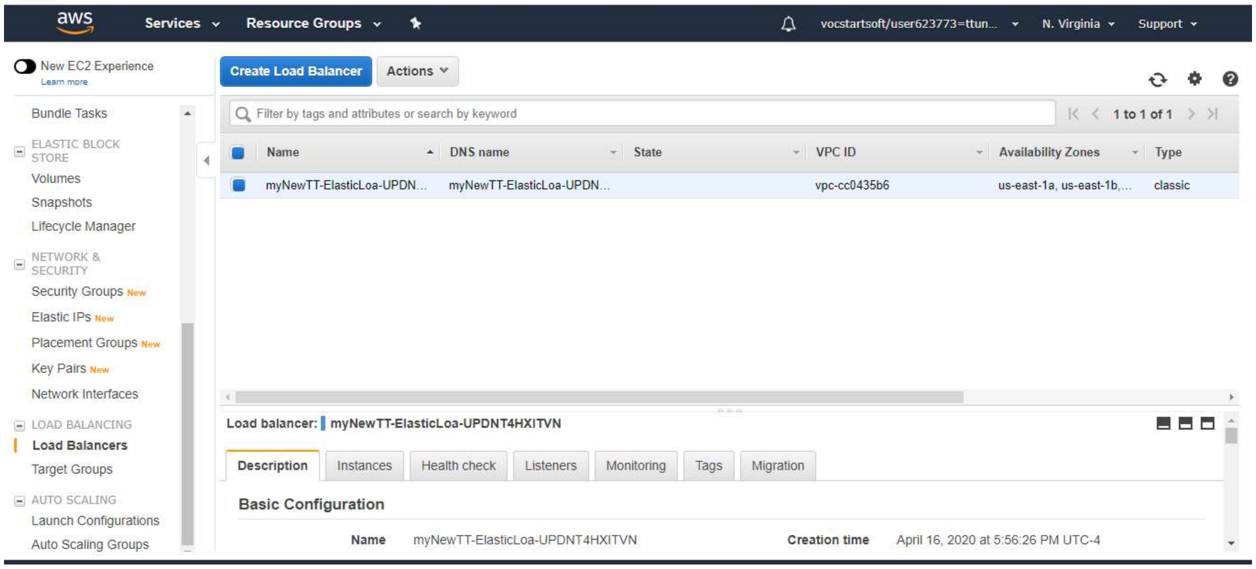
Filter: myNewTT-WebServerGroup-1E50KY36WCS

1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grac
myNewTT-We...	myNewTT-LaunchConfi...	1	1	1	3	us-east-1a, us-east-1b, us-e...	300	0

Auto Scaling Group: myNewTT-WebServerGroup-1E50KY36WCSBJ

MinSize and MaxSize set the minimum and maximum number of EC2 instances in the Auto Scaling group.



Name	DNS name	State	VPC ID	Availability Zones	Type
myNewTT-ElasticLoa-UPDN...	myNewTT-ElasticLoa-UPDN...		vpc-cc0435b6	us-east-1a, us-east-1b,...	classic

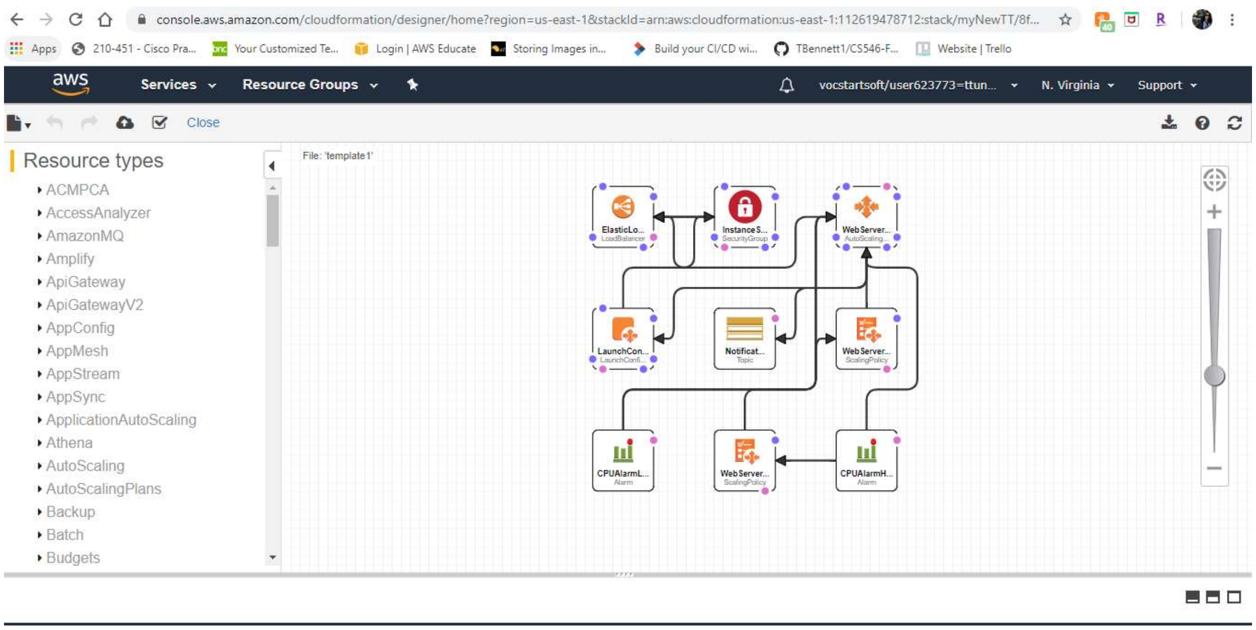
Load balancer: myNewTT-ElasticLoa-UPDNT4HXITVN

Basic Configuration

Name myNewTT-ElasticLoa-UPDNT4HXITVN   Creation time April 16, 2020 at 5:56:26 PM UTC-4

The template of load balancer

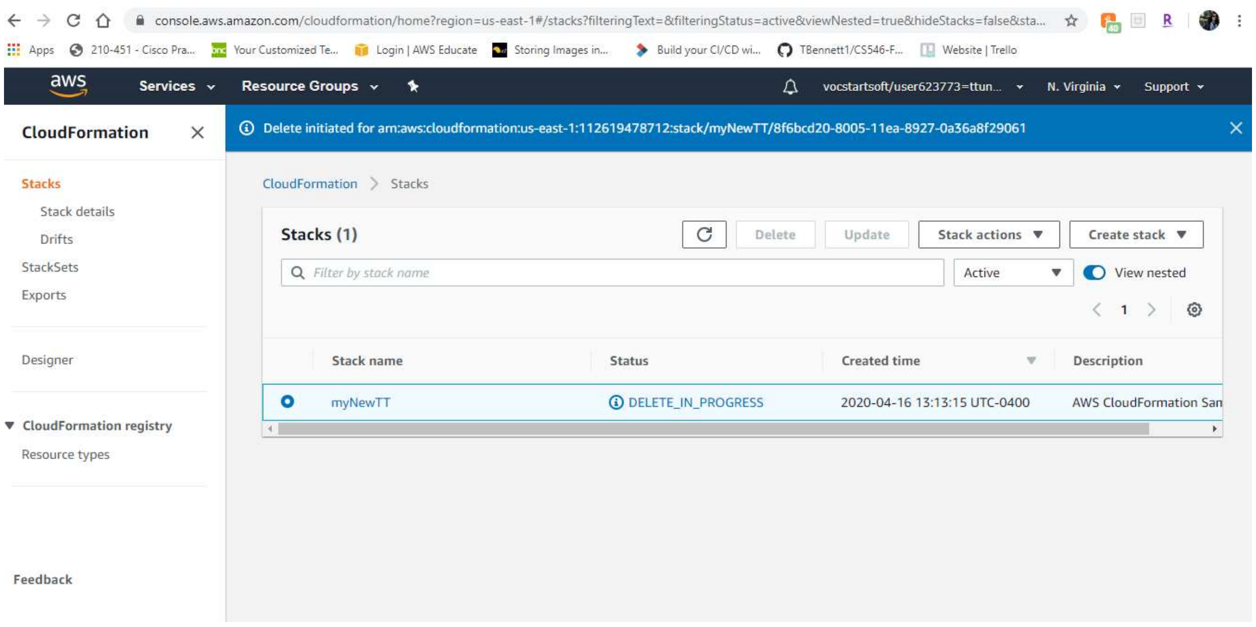
:



console.aws.amazon.com/cloudformation/designer/home?region=us-east-1&stackId=arn:aws:cloudformation:us-east-1:112619478712:stack/myNewTT/8f...

File: 'template1'

Delete Stack after these steps:



console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks?filteringText=&filteringStatus=active&viewNested=true&hideStacks=false&sta...

Delete initiated for arn:aws:cloudformation:us-east-1:112619478712:stack/myNewTT/8f6bcd20-8005-11ea-8927-0a36a8f29061

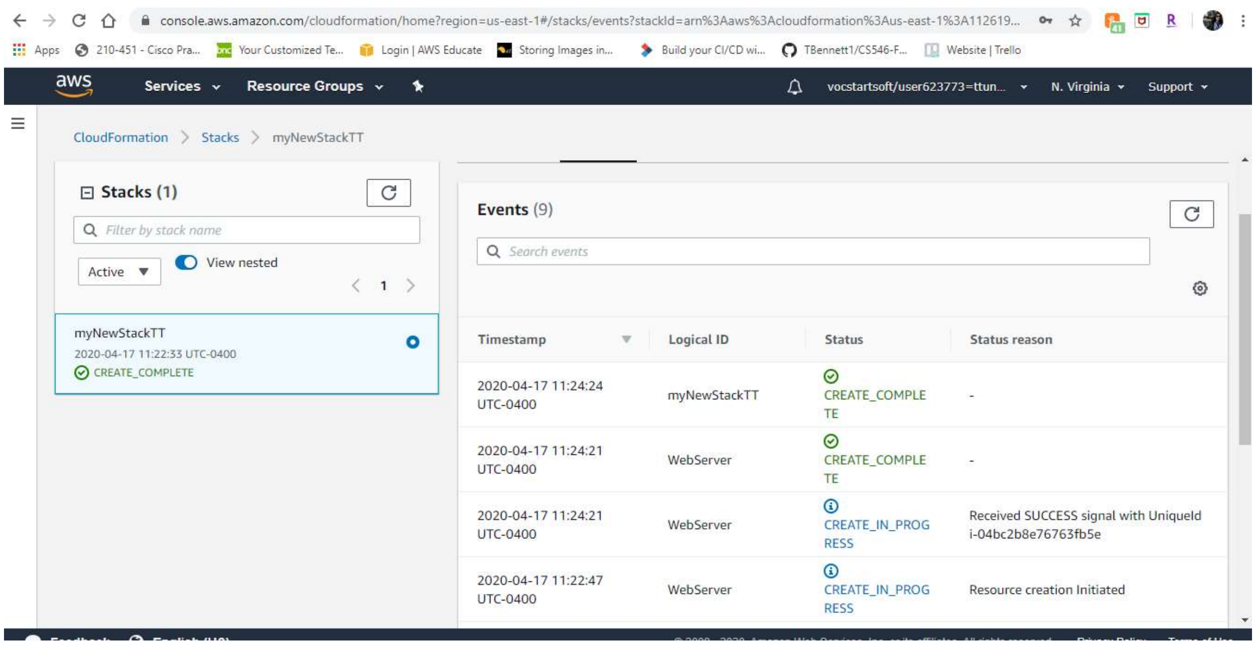
CloudFormation > Stacks

Stacks (1)

	Stack name	Status	Created time	Description
●	myNewTT	DELETE_IN_PROGRESS	2020-04-16 13:13:15 UTC-0400	AWS CloudFormation San

Since I deployed a LoadBalancer, my initial WordPress website has been overridden.

So I created a new stack using same steps described above:



console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/events?stackId=arn%3Aaws%3Acloudformation%3Aus-east-1%3A112619...

CloudFormation > Stacks > myNewStackTT

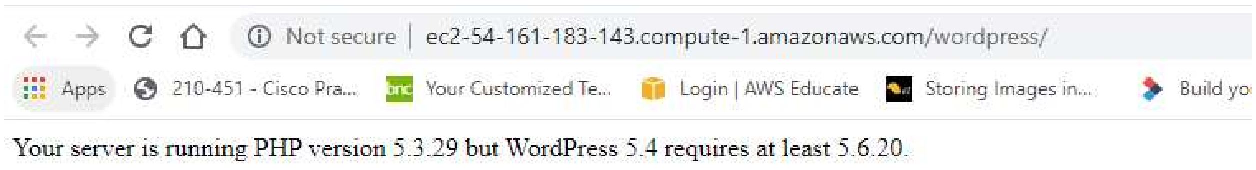
Stacks (1)

myNewStackTT
2020-04-17 11:22:33 UTC-0400
CREATE_COMPLETE

Events (9)

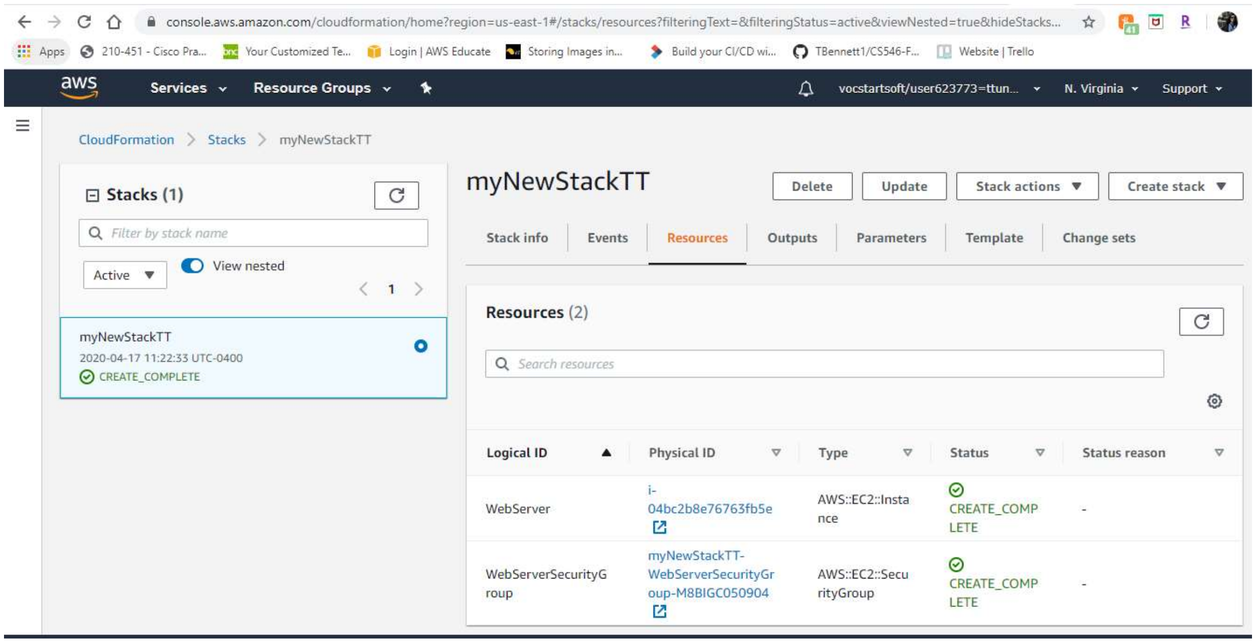
Timestamp	Logical ID	Status	Status reason
2020-04-17 11:24:24 UTC-0400	myNewStackTT	CREATE_COMPLETE	-
2020-04-17 11:24:21 UTC-0400	WebServer	CREATE_COMPLETE	-
2020-04-17 11:24:21 UTC-0400	WebServer	CREATE_IN_PROGRESS	Received SUCCESS signal with UniqueId i-04bc2b8e76763fb5e
2020-04-17 11:22:47 UTC-0400	WebServer	CREATE_IN_PROGRESS	Resource creation Initiated

The worpress URL can be found in the Outputs tab of the Stack:



Your server is running PHP version 5.3.29 but WordPress 5.4 requires at least 5.6.20.

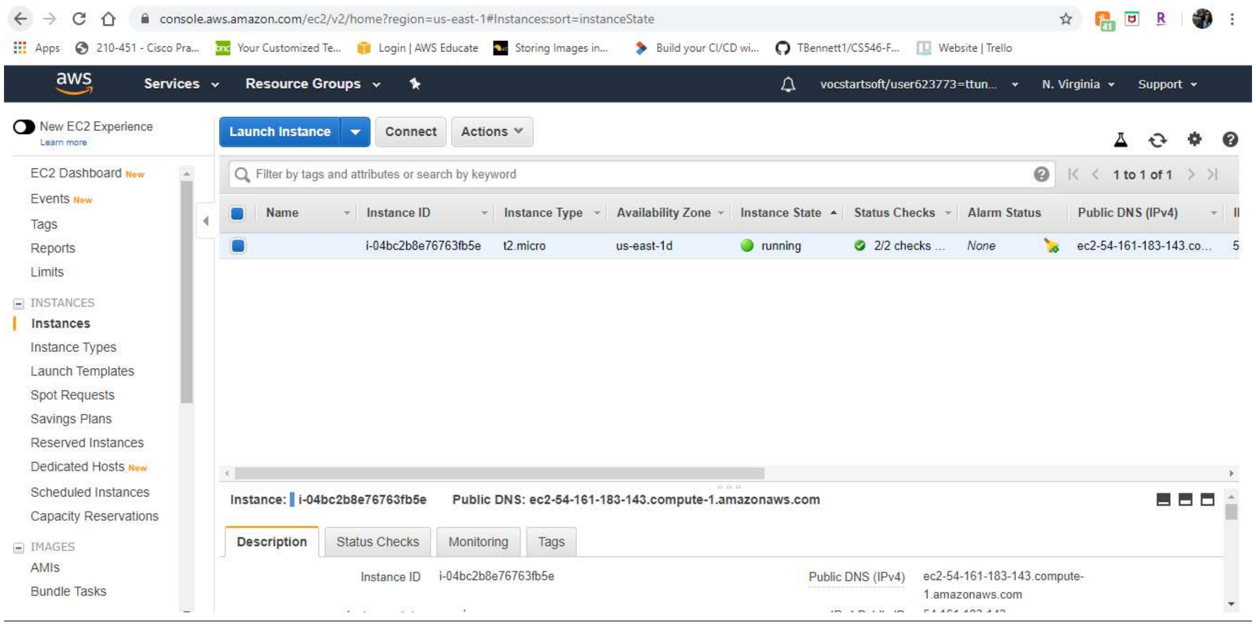
So the instance created in the backend is again Linux AMI ☹ instead of Linux 2. So I will have to install the new PHP version by connecting to the machine through Putty.



myNewStackTT

Logical ID	Physical ID	Type	Status	Status reason
WebServer	i-04bc2b8e76763fb5e	AWS::EC2::Instance	CREATE_COMPLETE	-
WebServerSecurityGroup	myNewStackTT-WebServerSecurityGroup-M8BIGC050904	AWS::EC2::SecurityGroup	CREATE_COMPLETE	-

The WebServer is the instance created in EC2.



console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceState

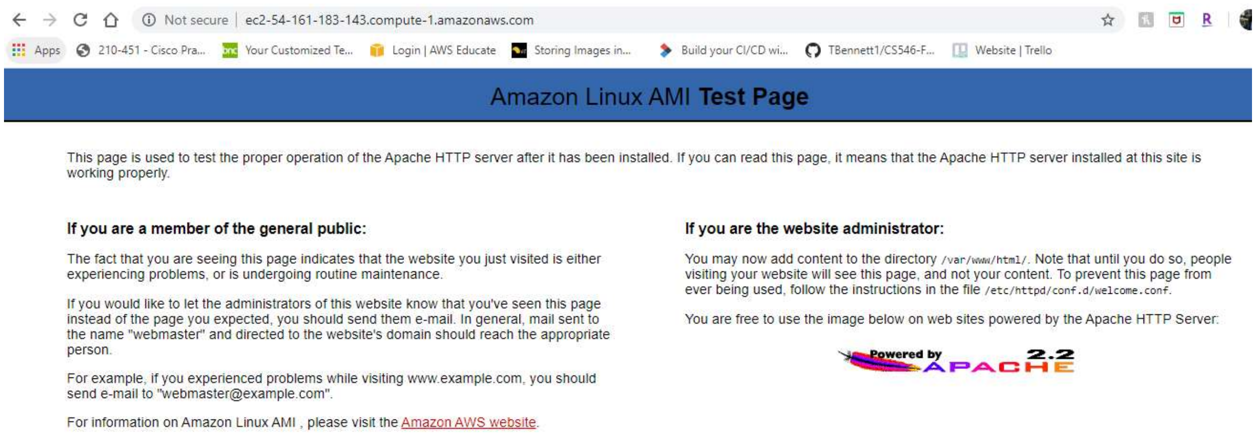
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-04bc2b8e76763fb5e	t2.micro	us-east-1d	running	2/2 checks ...	None	ec2-54-161-183-143.co...

Instance: i-04bc2b8e76763fb5e Public DNS: ec2-54-161-183-143.compute-1.amazonaws.com

Instance ID i-04bc2b8e76763fb5e Public DNS (IPv4) ec2-54-161-183-143.compute-1.amazonaws.com

This instance was created by the CloudFormation service.

I already have myKeyTT to connect to this machine using putty.



ec2-54-161-183-143.compute-1.amazonaws.com

Amazon Linux AMI **Test Page**

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

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

For information on Amazon Linux AMI , please visit the Amazon AWS website.

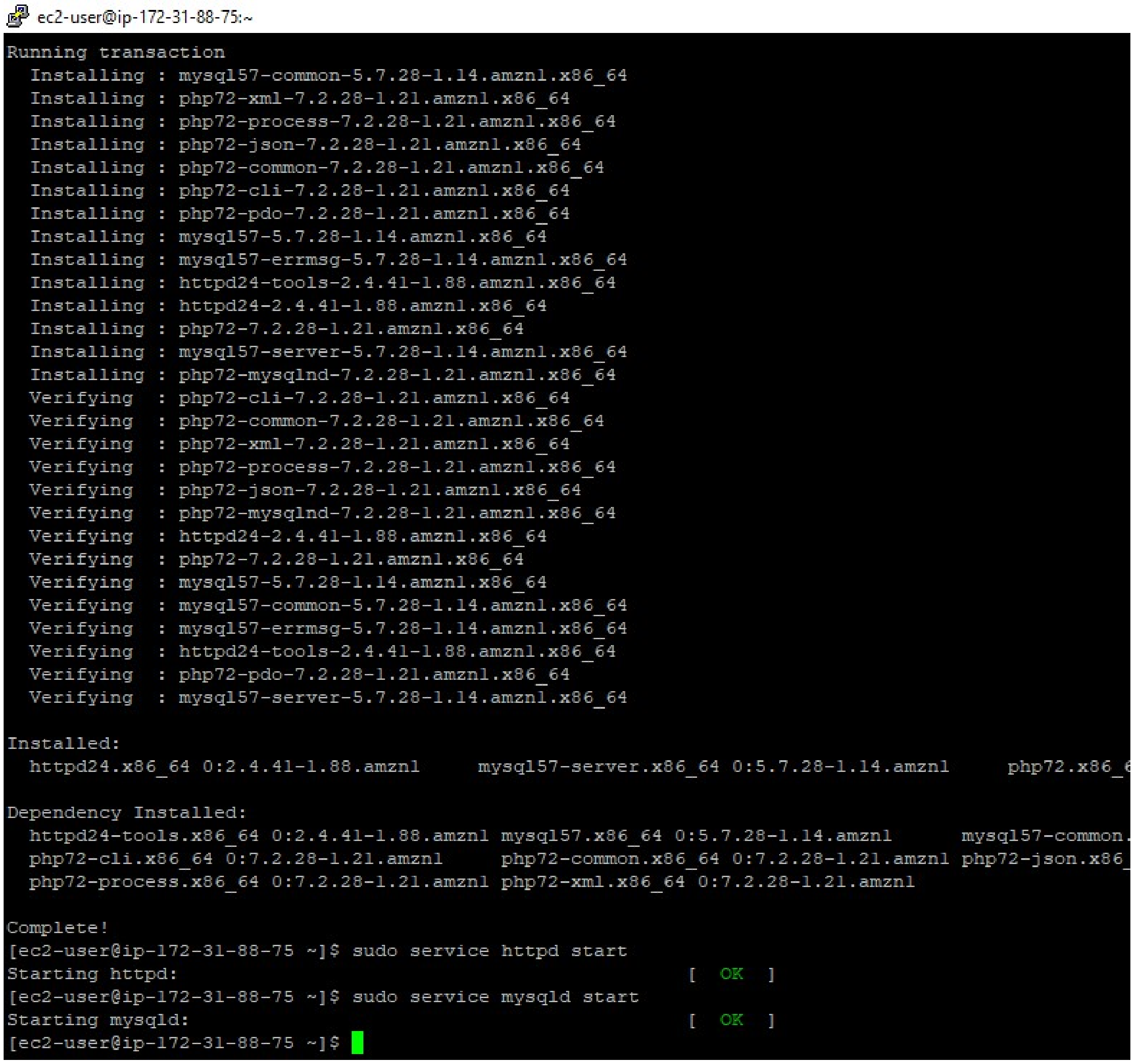
If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

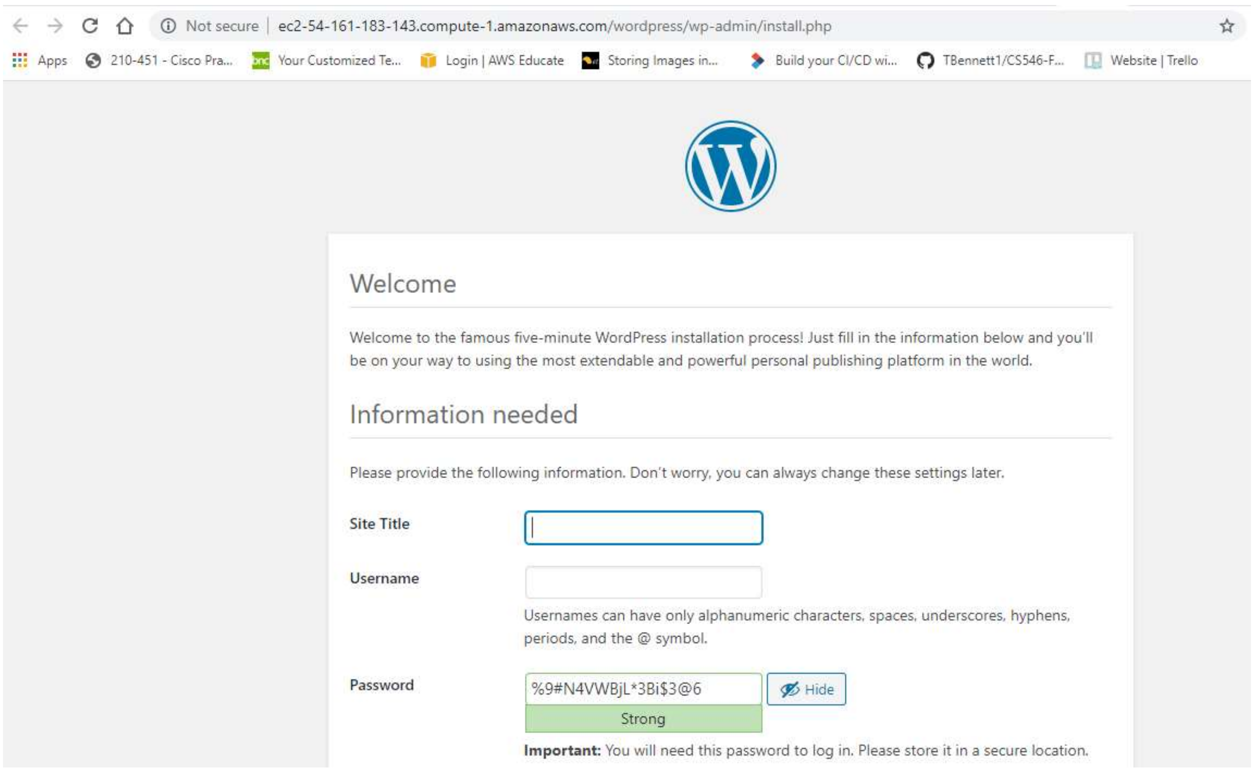
Powered by APACHE 2.2

After installing php 7.2 and restarting httpd and mysqld services:



```
ec2-user@ip-172-31-88-75:~
Running transaction
  Installing : mysql57-common-5.7.28-1.14.amzn1.x86_64
  Installing : php72-xml-7.2.28-1.21.amzn1.x86_64
  Installing : php72-process-7.2.28-1.21.amzn1.x86_64
  Installing : php72-json-7.2.28-1.21.amzn1.x86_64
  Installing : php72-common-7.2.28-1.21.amzn1.x86_64
  Installing : php72-cli-7.2.28-1.21.amzn1.x86_64
  Installing : php72-pdo-7.2.28-1.21.amzn1.x86_64
  Installing : mysql57-5.7.28-1.14.amzn1.x86_64
  Installing : mysql57-errmsg-5.7.28-1.14.amzn1.x86_64
  Installing : httpd24-tools-2.4.41-1.88.amzn1.x86_64
  Installing : httpd24-2.4.41-1.88.amzn1.x86_64
  Installing : php72-7.2.28-1.21.amzn1.x86_64
  Installing : mysql57-server-5.7.28-1.14.amzn1.x86_64
  Installing : php72-mysqlnd-7.2.28-1.21.amzn1.x86_64
  Verifying  : php72-cli-7.2.28-1.21.amzn1.x86_64
  Verifying  : php72-common-7.2.28-1.21.amzn1.x86_64
  Verifying  : php72-xml-7.2.28-1.21.amzn1.x86_64
  Verifying  : php72-process-7.2.28-1.21.amzn1.x86_64
  Verifying  : php72-json-7.2.28-1.21.amzn1.x86_64
  Verifying  : php72-mysqlnd-7.2.28-1.21.amzn1.x86_64
  Verifying  : httpd24-2.4.41-1.88.amzn1.x86_64
  Verifying  : php72-7.2.28-1.21.amzn1.x86_64
  Verifying  : mysql57-5.7.28-1.14.amzn1.x86_64
  Verifying  : mysql57-common-5.7.28-1.14.amzn1.x86_64
  Verifying  : mysql57-errmsg-5.7.28-1.14.amzn1.x86_64
  Verifying  : httpd24-tools-2.4.41-1.88.amzn1.x86_64
  Verifying  : php72-pdo-7.2.28-1.21.amzn1.x86_64
  Verifying  : mysql57-server-5.7.28-1.14.amzn1.x86_64

Installed:
  httpd24.x86_64 0:2.4.41-1.88.amzn1     mysql57-server.x86_64 0:5.7.28-1.14.amzn1     php72.x86_6

Dependency Installed:
  httpd24-tools.x86_64 0:2.4.41-1.88.amzn1 mysql57.x86_64 0:5.7.28-1.14.amzn1      mysql57-common.
  php72-cli.x86_64 0:7.2.28-1.21.amzn1     php72-common.x86_64 0:7.2.28-1.21.amzn1 php72-json.x86_
  php72-process.x86_64 0:7.2.28-1.21.amzn1 php72-xml.x86_64 0:7.2.28-1.21.amzn1

Complete!
[ec2-user@ip-172-31-88-75 ~]$ sudo service httpd start
Starting httpd:                                            [  OK  ]
[ec2-user@ip-172-31-88-75 ~]$ sudo service mysqld start
Starting mysqld:                                           [  OK  ]
[ec2-user@ip-172-31-88-75 ~]$
```

Copy the URL of WordPress website and paste it in the browser:


Not secure | ec2-54-161-183-143.compute-1.amazonaws.com/wordpress/wp-admin/install.php

Apps | 210-451 - Cisco Pra... | Your Customized Te... | Login | AWS Educate | Storing Images in... | Build your CI/CD wi... | TBennett1/CS546-F... | Website | Trello

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title:

Username:

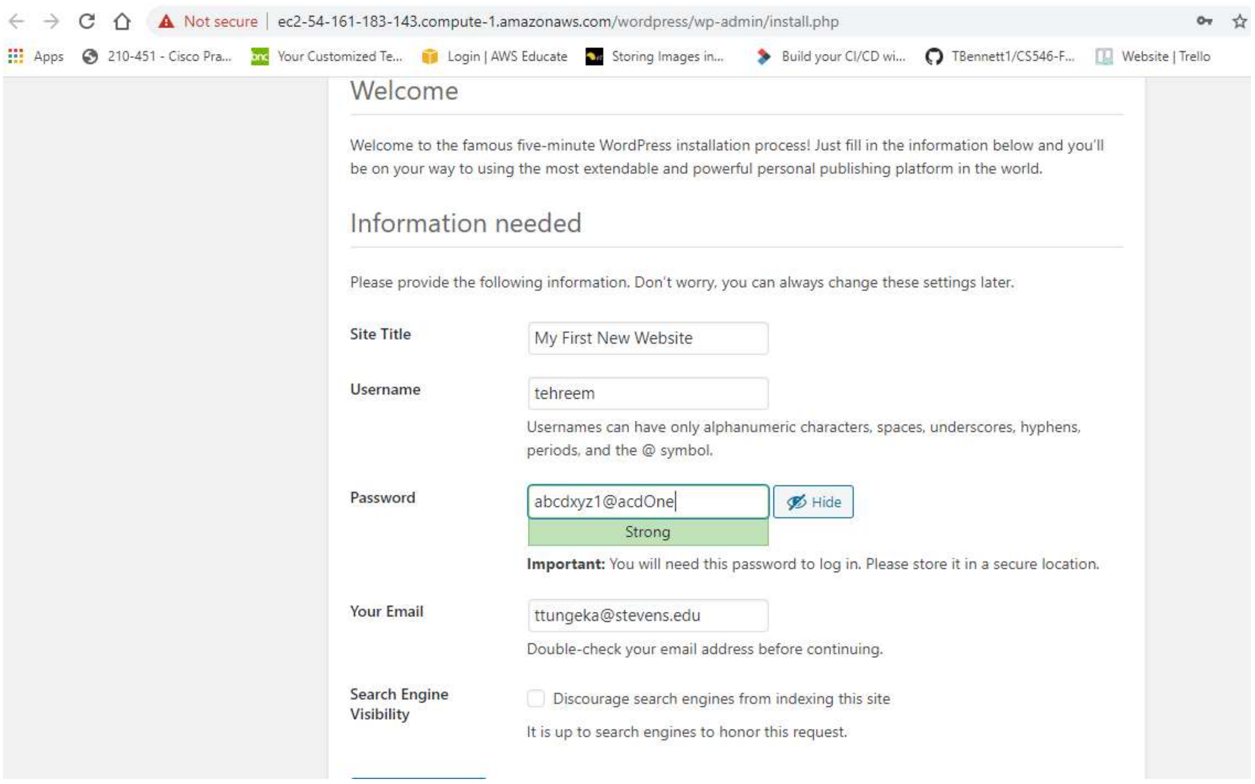
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: %9#N4VWBjL*3Bi$3@6 — Hide — Strong

Important: You will need this password to log in. Please store it in a secure location.

So the website is running successfully!

WordPress lets you create blogs and it is widely used on the web.


Not secure | ec2-54-161-183-143.compute-1.amazonaws.com/wordpress/wp-admin/install.php

Apps | 210-451 - Cisco Pra... | Your Customized Te... | Login | AWS Educate | Storing Images in... | Build your CI/CD wi... | TBennett1/CS546-F... | Website | Trello

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title: My First New Website

Username: tehreem

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password: abcdxyz1@acdOne — Hide — Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email: ttungeka@stevens.edu

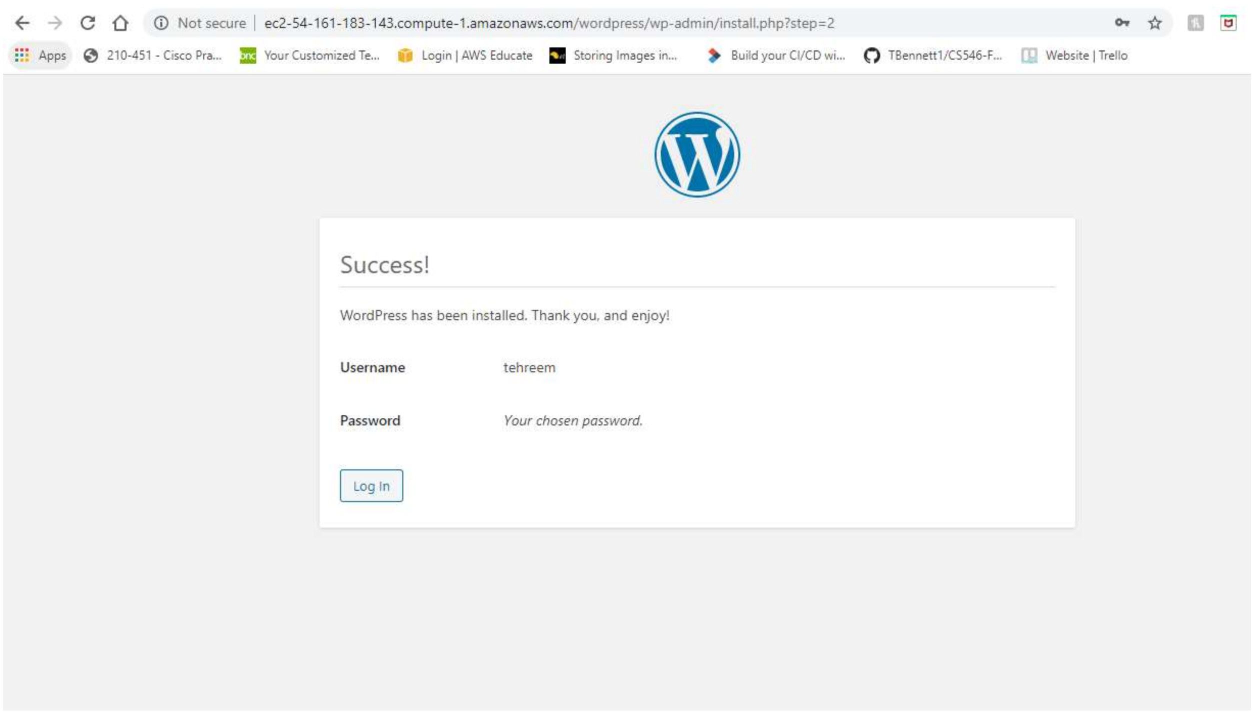
Double-check your email address before continuing.

Search Engine Visibility: ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Don’t check the search engine visibility checkbox else traffic to your website will be lost.

Click on Install WordPress.



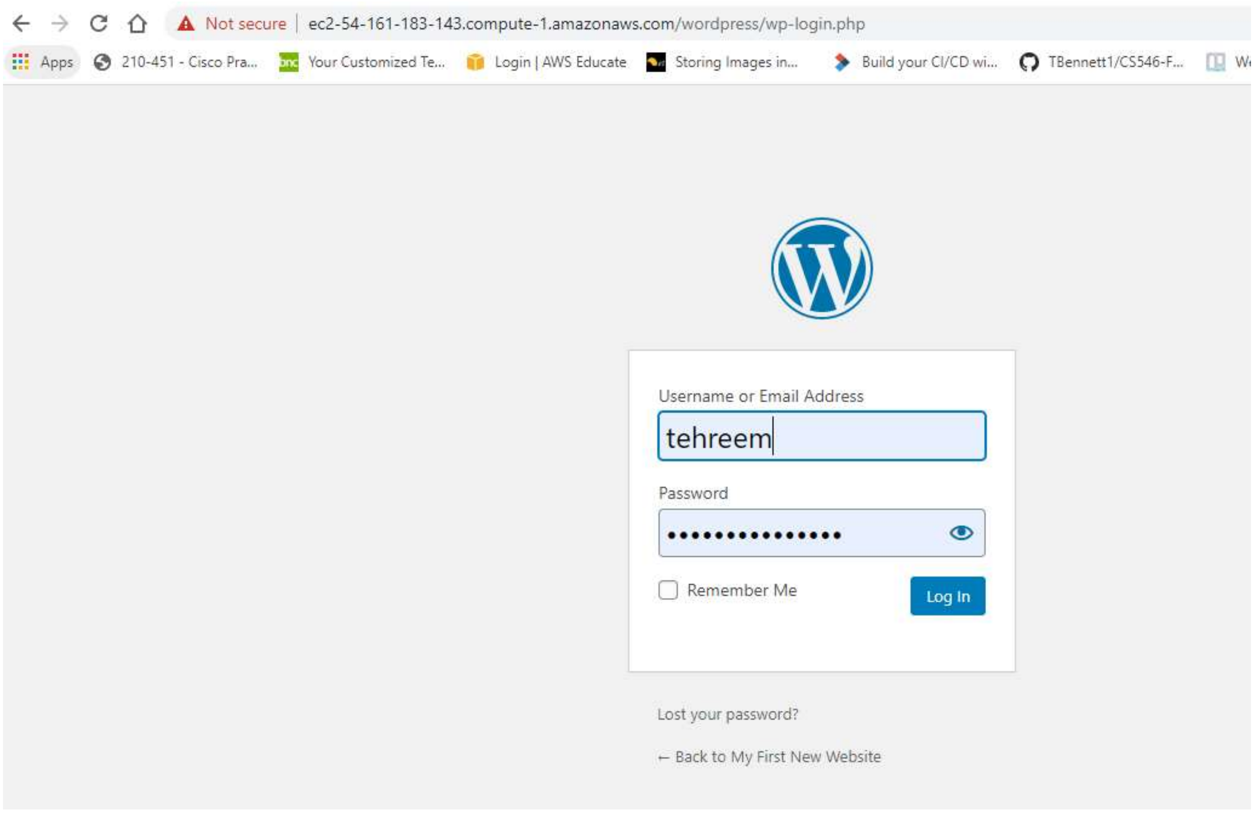
Not secure | ec2-54-161-183-143.compute-1.amazonaws.com/wordpress/wp-admin/install.php?step=2

Success!

WordPress has been installed. Thank you, and enjoy!

Username	tehreem
Password	*Your chosen password.*

Click on Log in.

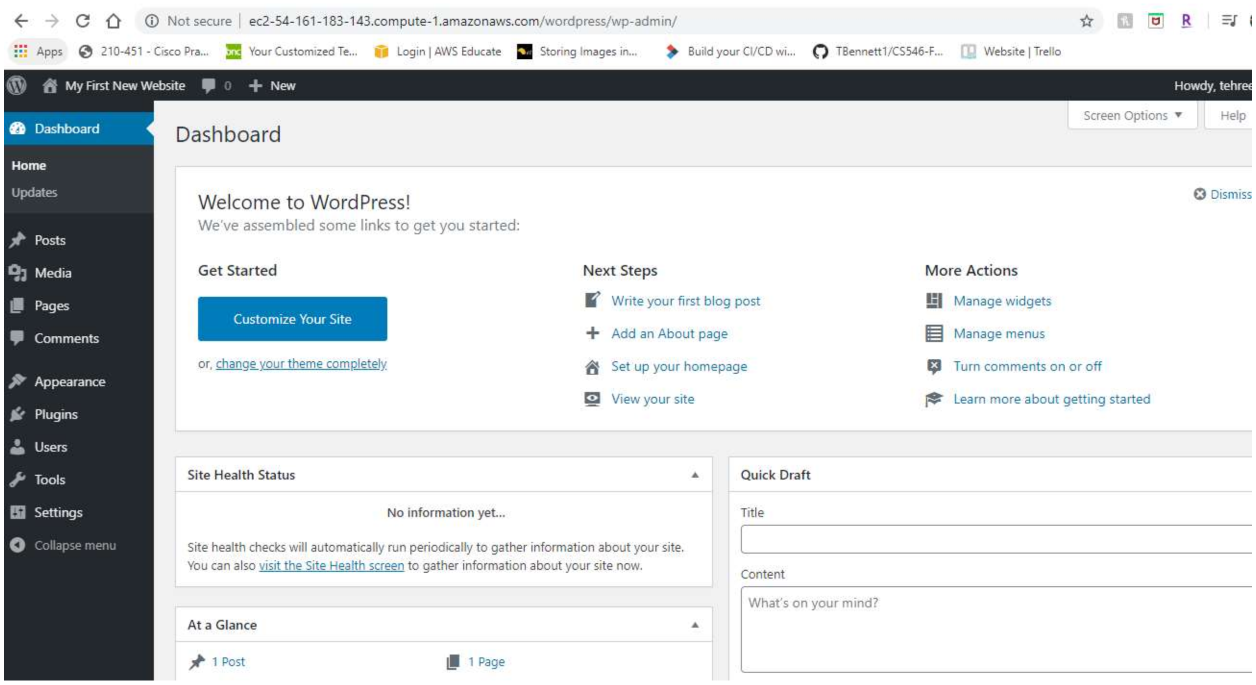


Not secure | ec2-54-161-183-143.compute-1.amazonaws.com/wordpress/wp-login.php

Username or Email Address: tehreem

Password: •••••••••••••••

Click on Log in after entering your username and password. You will be redirected to WordPress dashboard.



Dashboard

Welcome to WordPress!
We've assembled some links to get you started:

Get Started
Customize Your Site
or, change your theme completely

Next Steps
Write your first blog post
Add an About page
Set up your homepage
View your site

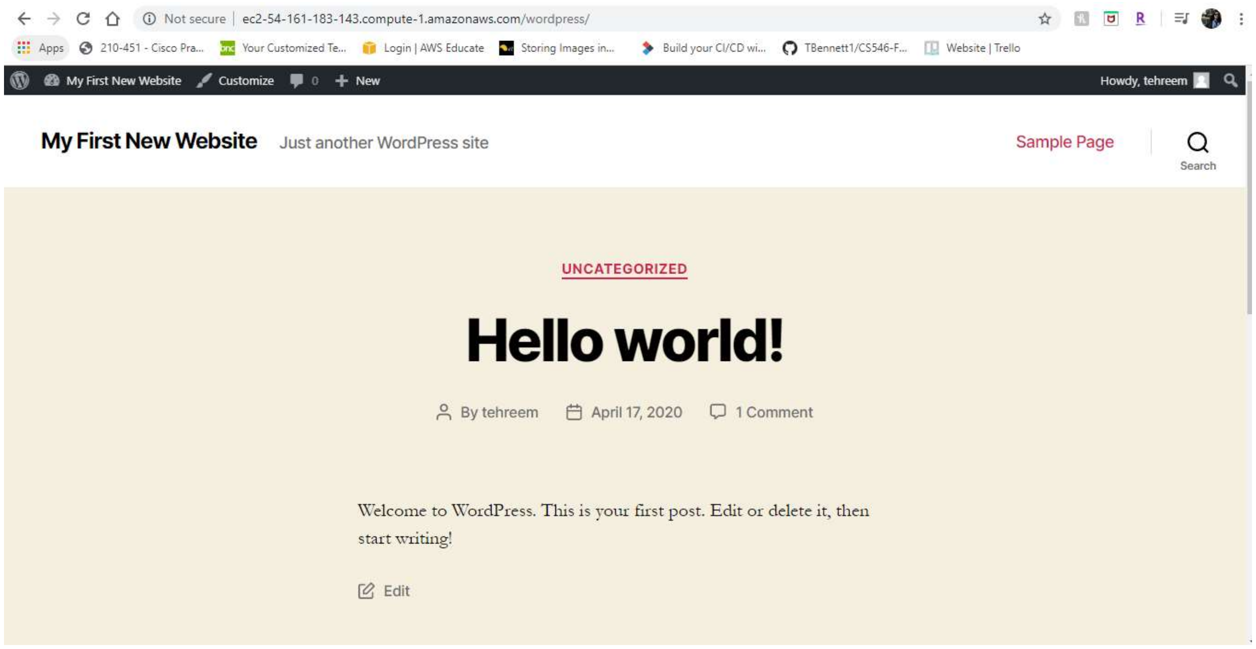
More Actions
Manage widgets
Manage menus
Turn comments on or off
Learn more about getting started

Site Health Status
No information yet...
Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.

At a Glance
1 Post
1 Page

Quick Draft
Title
Content
What's on your mind?

Click on the Home button on the top-left corner – My first Website and Click on Visit.


Not secure | ec2-54-161-183-143.compute-1.amazonaws.com/wordpress/
My First New Website
Customize
New
Howdy, tehreem
Sample Page
Search

My First New Website Just another WordPress site

UNCATEGORIZED

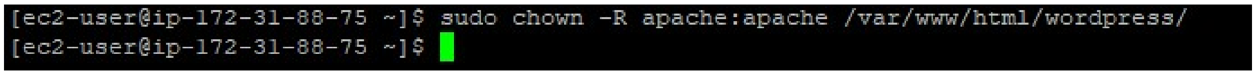
Hello world!

By tehreem April 17, 2020 1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

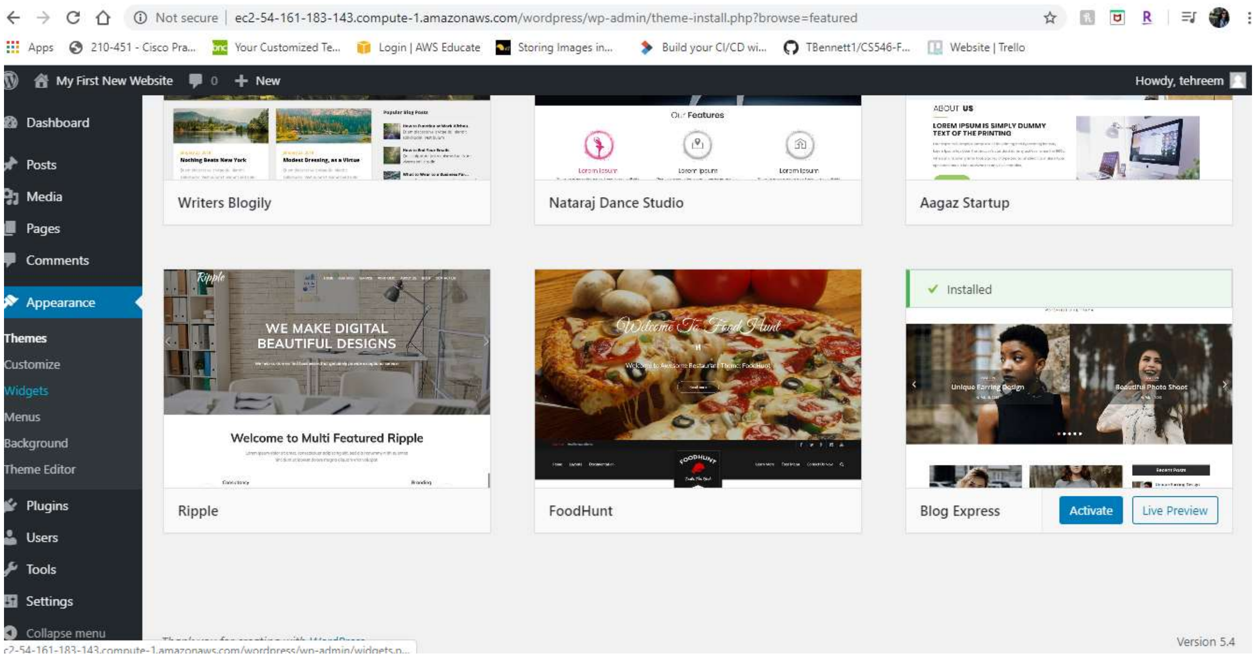
Edit

In the terminal, put this command so that you can install other themes without giving any credentials for FTP.



```
[ec2-user@ip-172-31-88-75 ~]$ sudo chown -R apache:apache /var/www/html/wordpress/
[ec2-user@ip-172-31-88-75 ~]$
```

Now install any theme of your choice:

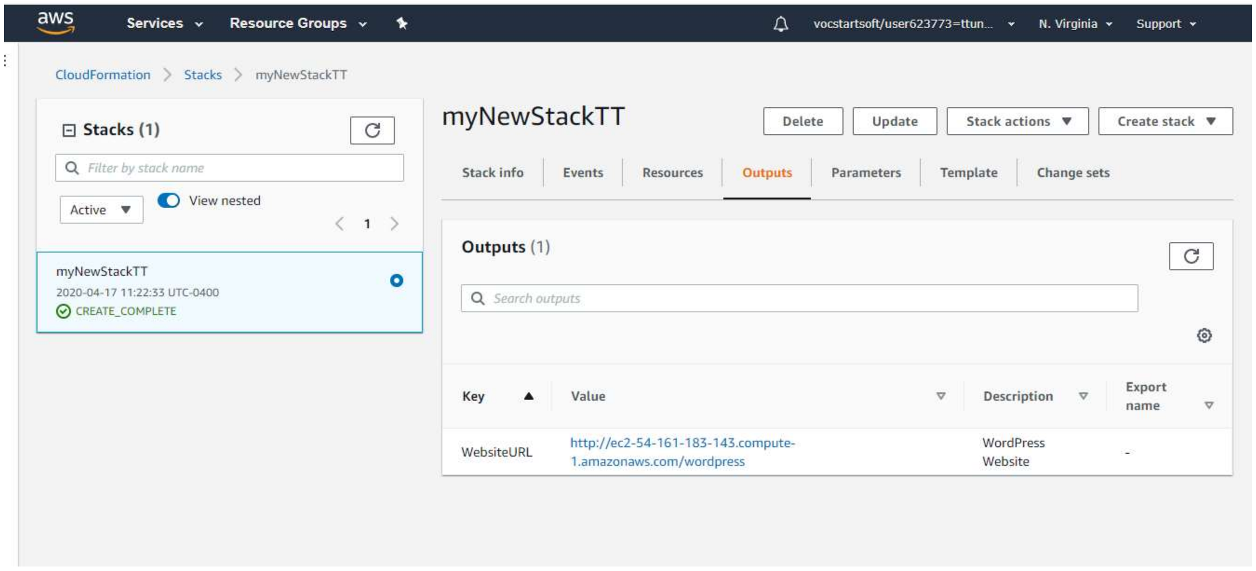


I have installed Blog Express Theme.

Click on Activate.

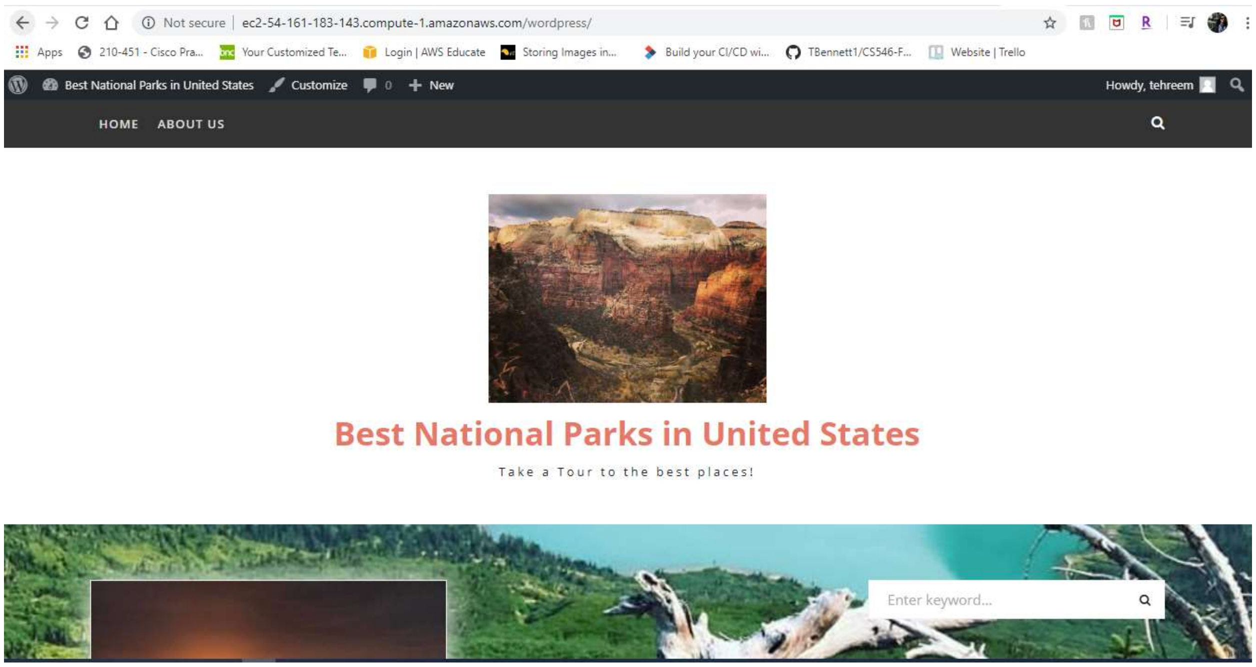
So I have edited and customized my website.

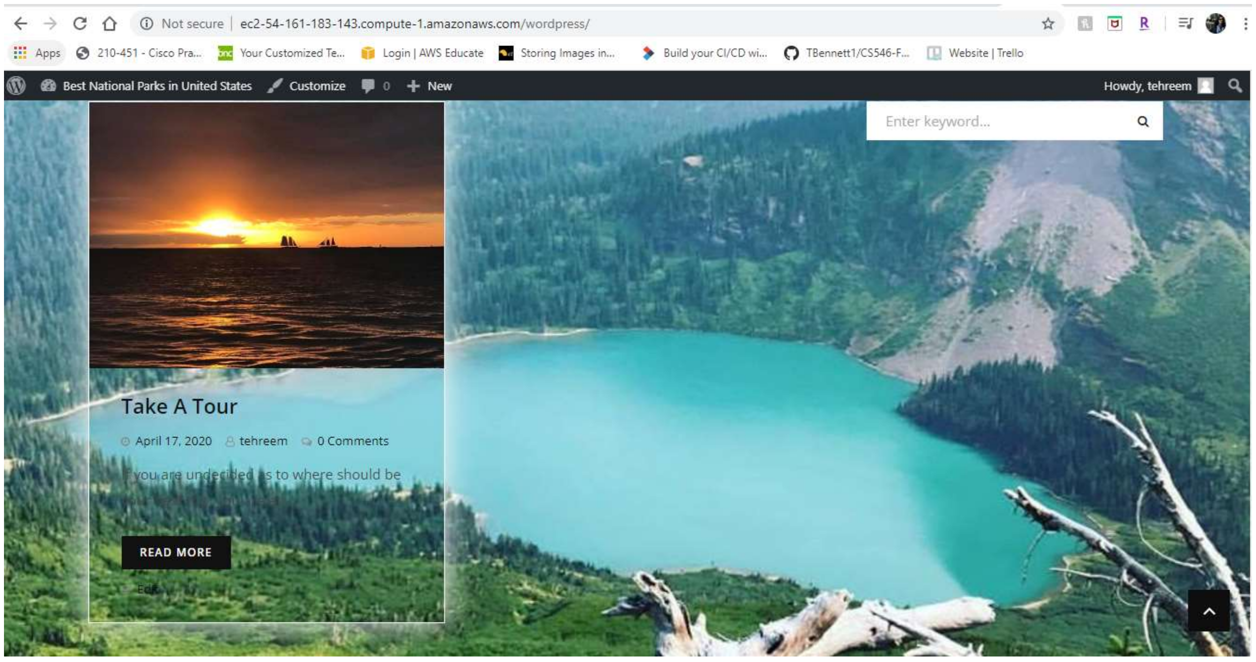
Now go to CloudFormation from Services:



Key	Value	Description	Export name
WebsiteURL	http://ec2-54-161-183-143.compute-1.amazonaws.com/wordpress	WordPress Website	-

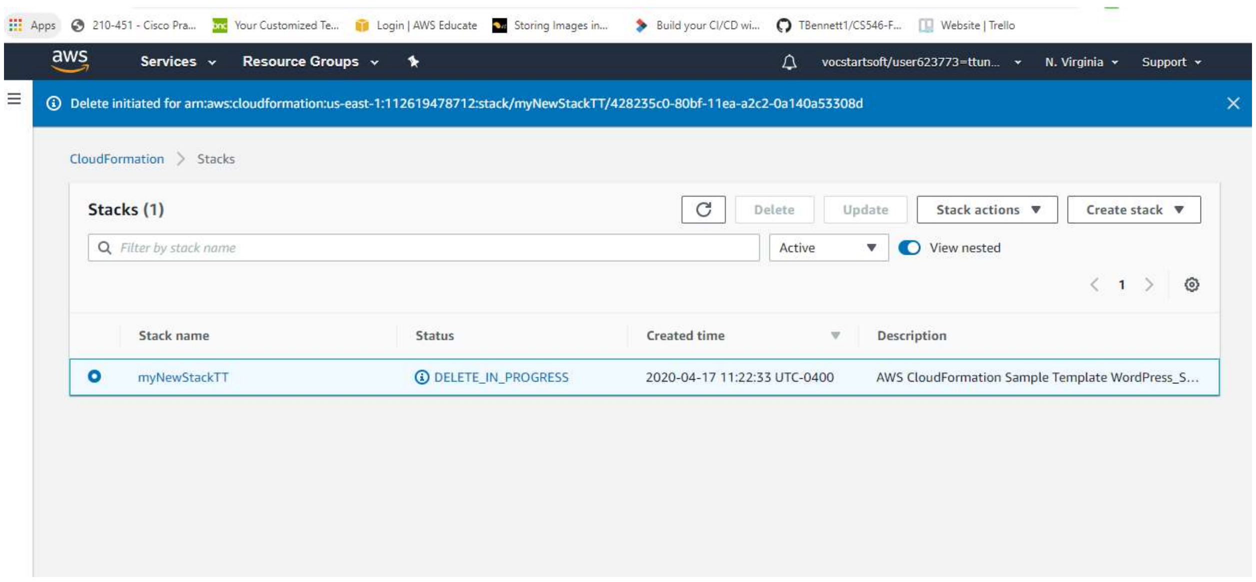
Click on the website URL in the Outputs tab:




Not secure | ec2-54-161-183-143.compute-1.amazonaws.com/wordpress/
Best National Parks in United States
Customize
0
New
Howdy, tehreem
Enter keyword...
Take A Tour
April 17, 2020
tehreem
0 Comments
If you are undecided as to where should be
READ MORE

The WordPress website has been successfully deployed.

Now go to Stacks and Delete the stack. The instance associated with the stack will be deleted automatically.



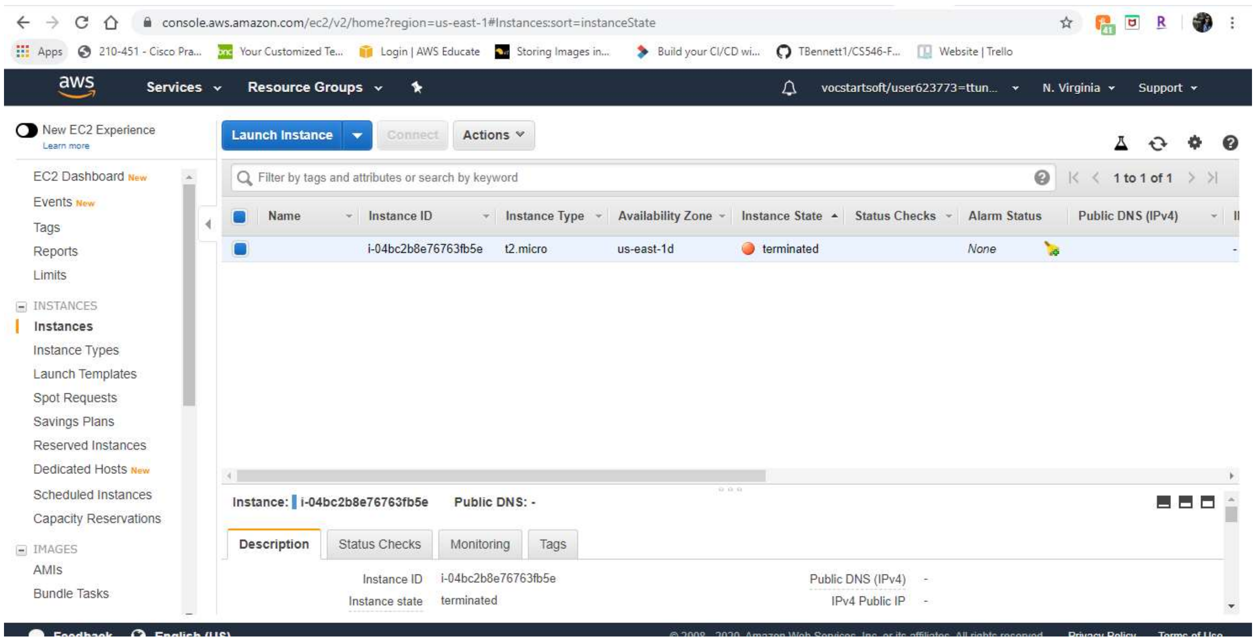
Delete initiated for arn:aws:cloudformation:us-east-1:112619478712:stack/myNewStackTT/428235c0-80bf-11ea-a2c2-0a140a53308d

CloudFormation > Stacks

Stacks (1)

Stack name	Status	Created time	Description
myNewStackTT	DELETE_IN_PROGRESS	2020-04-17 11:22:33 UTC-0400	AWS CloudFormation Sample Template WordPress_S...

The EC2 Instance has been terminated in the backend:



console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceState

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-04bc2b8e76763fb5e	t2.micro	us-east-1d	terminated		None	-

Instance: i-04bc2b8e76763fb5e Public DNS: -

Instance ID	i-04bc2b8e76763fb5e	Public DNS (IPv4)	-
Instance state	terminated	IPv4 Public IP	-