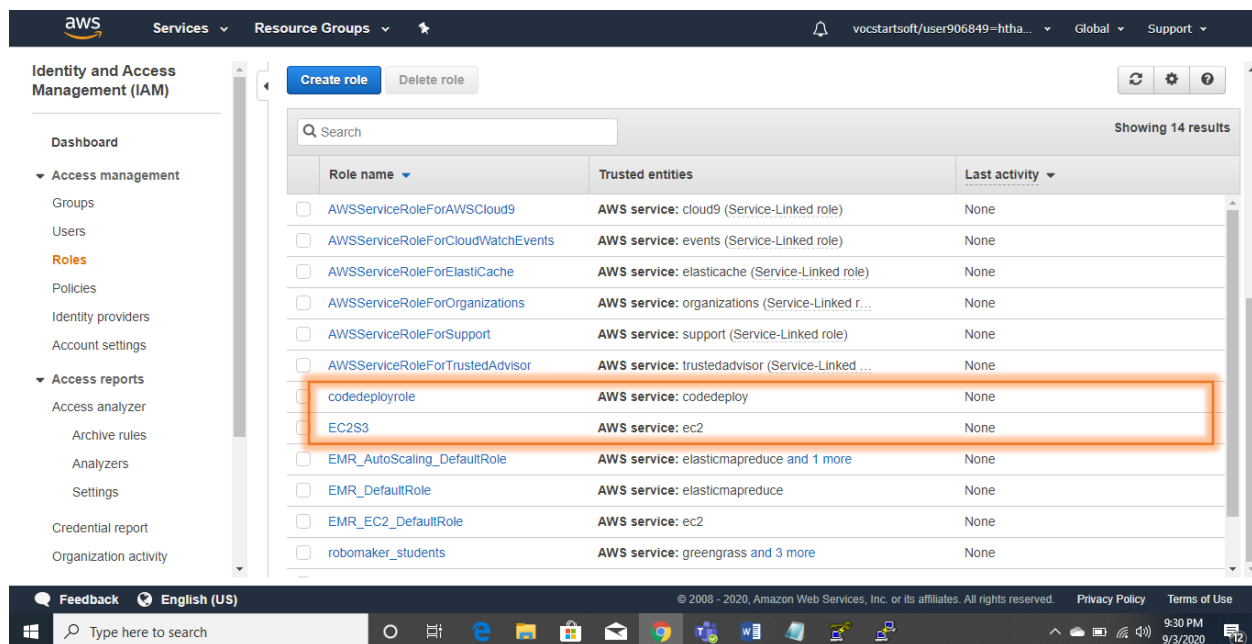


Created 2 IAM (Identity Access Management) roles

Codedeployrole and EC2S3 role

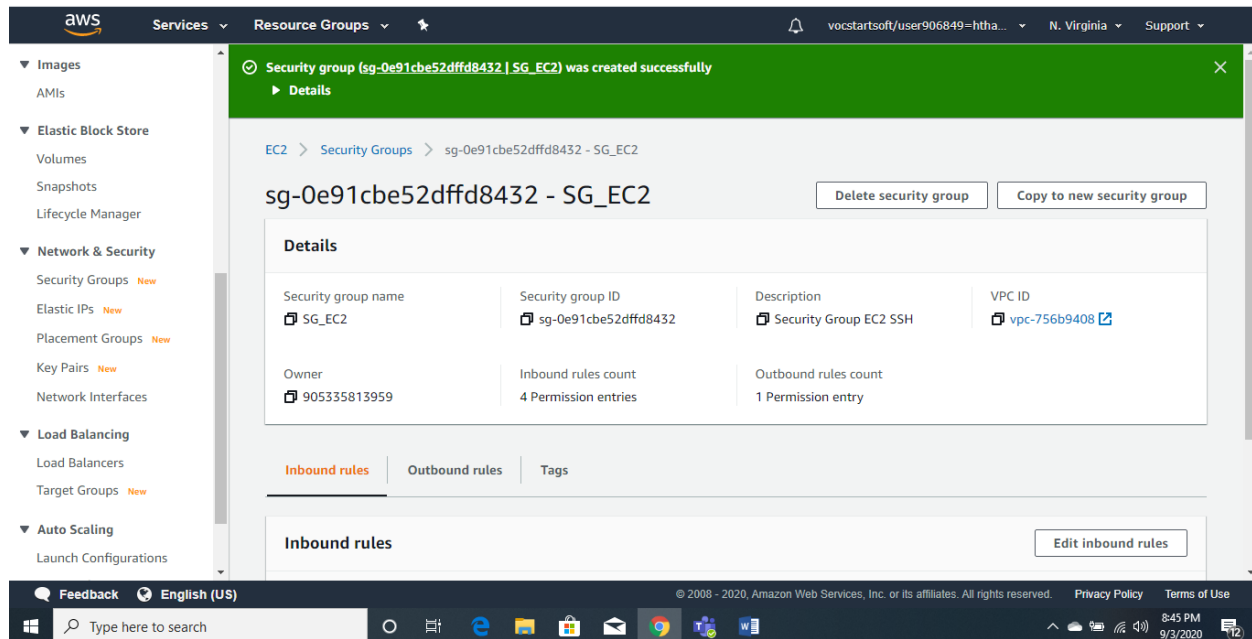
Codedeployrole was created for using it to deploy the Github code webpage to EC2 instance

EC2S3 for creating an EC2 instance using this role

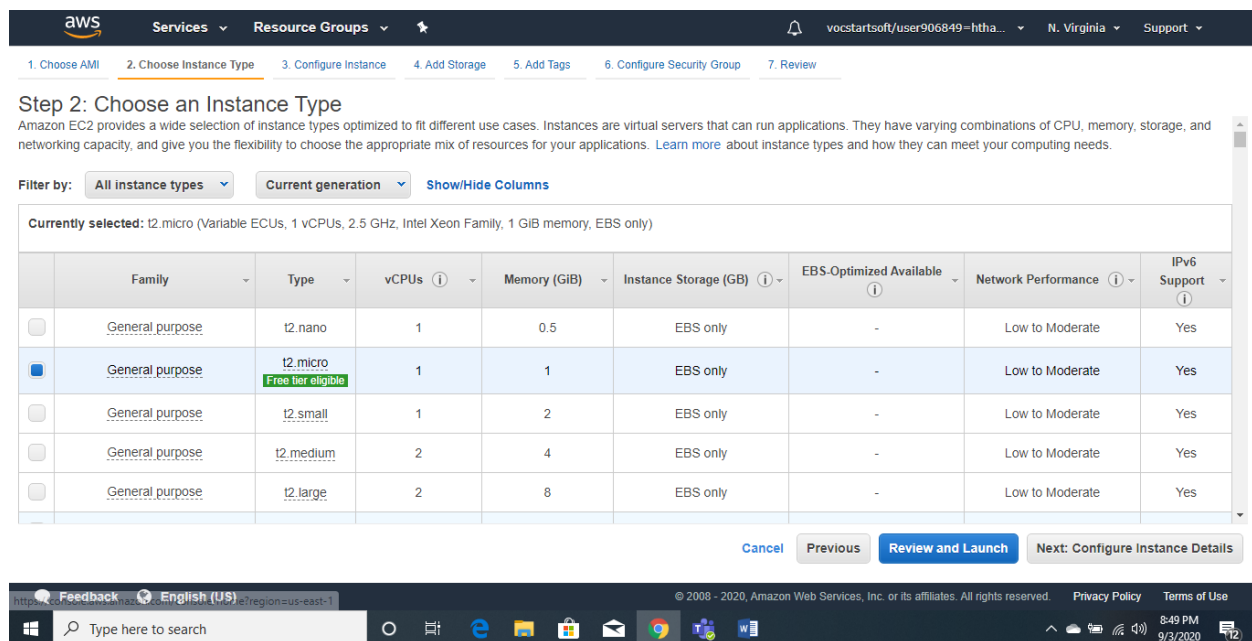


Security Group Creation

Created SG_EC2 Security group with 2 inbound rules for "SSH" and "HTTP" with "Anywhere" option as the source



Launching the EC2 Instance



Launching the EC2 instance with the following configuration settings: "EC2S3" for IAM role for EC2 instance to read the data from the S3 bucket, Amazon Linux AMI 2018.03.0 for Amazon Machine Image (AMI), script in advance details to install codeploy agent in EC2 instance, tag with "Key" as "Name" and "Value" as "Web Server" along with selecting the Security Group as "SG_EC2".

The screenshot shows the 'Step 6: Configure Security Group' page in the AWS Management Console. It includes a progress bar at the top with steps 1 through 7. The main heading is 'Step 6: Configure Security Group'. Below it, a paragraph explains that a security group is a set of firewall rules that control traffic for an instance. It provides instructions on how to add rules for HTTP and HTTPS ports. There are two radio buttons under 'Assign a security group': 'Create a new security group' (unselected) and 'Select an existing security group' (selected). Below this is a table of existing security groups. The table has columns: Security Group ID, Name, Description, and Actions. Two groups are listed: 'sg-851f25b8' (default VPC security group) and 'sg-0e91cbe52dff8432' (SG_EC2, Security Group EC2 SSH). The second group is selected. Below the table, there is a section for 'Inbound rules for sg-0e91cbe52dff8432 (Selected security groups: sg-0e91cbe52dff8432)'. It shows a table with columns: Type, Protocol, Port Range, Source, and Description. Two rules are listed: HTTP (TCP, Port 80, Source 0.0.0.0/0) and HTTP (TCP, Port 80, Source ::/0). At the bottom right of the rules table are buttons: 'Cancel', 'Previous', and 'Review and Launch'. The footer of the console shows the user's name, region (N. Virginia), and support link. The Windows taskbar at the bottom shows the time as 8:54 PM on 9/3/2020.

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☐ Create a new security group
☒ Select an existing security group

Security Group ID	Name	Description	Actions
sg-851f25b8	default	default VPC security group	Copy to new
sg-0e91cbe52dff8432	SG_EC2	Security Group EC2 SSH	Copy to new

Inbound rules for sg-0e91cbe52dff8432 (Selected security groups: sg-0e91cbe52dff8432)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTP	TCP	80	::/0	

[Cancel](#) [Previous](#) [Review and Launch](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Type here to search

Launched Instance EC2 is running successfully.

The screenshot shows the 'Instances' page in the AWS Management Console. The left sidebar contains navigation links: 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances' (expanded), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Scheduled Instances', 'Capacity Reservations', 'Images', and 'AMIs'. The main content area shows a table of instances. The table has columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). One instance is listed: 'Web Server' with Instance ID 'i-0d475f11b01a7e06d', Instance Type 't2.micro', Availability Zone 'us-east-1a', Instance State 'running', Status Checks 'Initializing', Alarm Status 'None', and Public DNS 'ec2-52-207-220-80.compute-1.amazonaws.com'. Below the table, there is a detailed view for the selected instance 'i-0d475f11b01a7e06d (Web Server)'. It shows the Public DNS as 'ec2-52-207-220-80.compute-1.amazonaws.com'. The 'Description' tab is selected, showing the Instance ID 'i-0d475f11b01a7e06d', Instance state 'running', and IPv4 Public IP '52.207.220.80'. The footer of the console shows the user's name, region (N. Virginia), and support link. The Windows taskbar at the bottom shows the time as 8:59 PM on 9/3/2020.

aws Services Resource Groups

New EC2 Experience

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Web Server	i-0d475f11b01a7e06d	t2.micro	us-east-1a	running	Initializing	None	ec2-52-207-220-80.co...

Instance: i-0d475f11b01a7e06d (Web Server) Public DNS: ec2-52-207-220-80.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID i-0d475f11b01a7e06d Public DNS (IPv4) ec2-52-207-220-80.compute-1.amazonaws.com

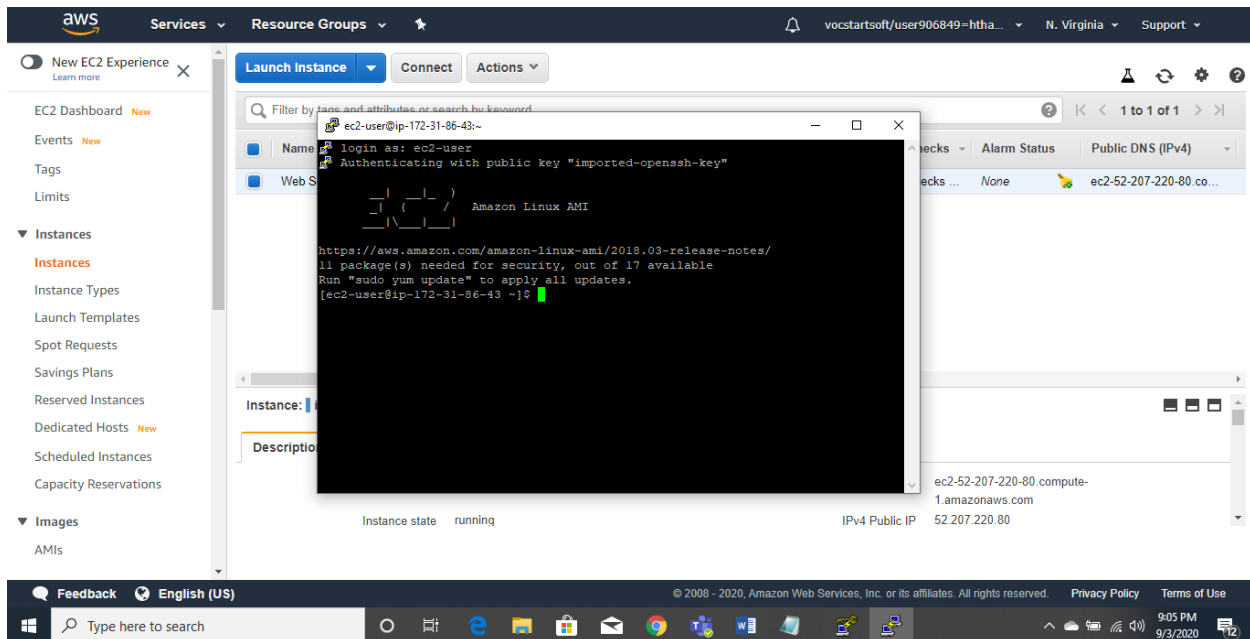
Instance state running IPv4 Public IP 52.207.220.80

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

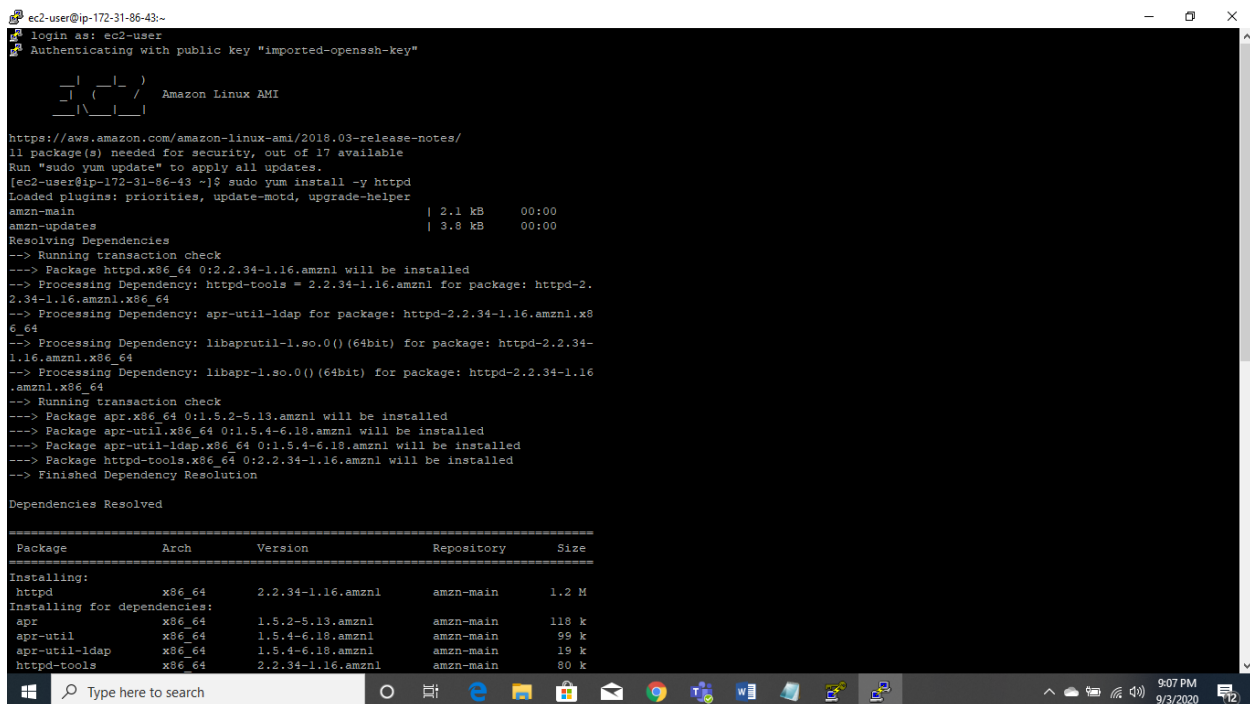
Type here to search

Connecting to the EC2 Instance

Connecting to the EC2 instance using SSH to ensure the privacy by generating the Key pairs for authenticating the login for the permitted users.



Setting up the EC2 instance to host the web page



```
ec2-user@ip-172-31-86-43:~$ sudo yum install httpd
apr-util      x86_64      1.5.4-6.18.amzn1      amzn-main      96 k
apr-util-ldap x86_64      1.5.4-6.18.amzn1      amzn-main      19 k
httpd-tools   x86_64      2.2.34-1.16.amzn1      amzn-main      80 k

Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 3.6 M
Downloading packages:
(1/5): apr-util-ldap-1.5.4-6.18.amzn1.x86_64.rpm      | 19 kB  00:00
(2/5): apr-util-1.5.4-6.18.amzn1.x86_64.rpm          | 99 kB  00:00
(3/5): apr-1.5.2-5.13.amzn1.x86_64.rpm               | 118 kB  00:00
(4/5): httpd-tools-2.2.34-1.16.amzn1.x86_64.rpm      | 80 kB  00:00
(5/5): httpd-2.2.34-1.16.amzn1.x86_64.rpm            | 1.2 MB  00:00
-----
Total                                          2.2 MB/s | 1.5 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.5.2-5.13.amzn1.x86_64            1/5
  Installing : apr-util-1.5.4-6.18.amzn1.x86_64       2/5
  Installing : httpd-tools-2.2.34-1.16.amzn1.x86_64   3/5
  Installing : apr-util-ldap-1.5.4-6.18.amzn1.x86_64  4/5
  Installing : httpd-2.2.34-1.16.amzn1.x86_64        5/5
  Verifying   : httpd-tools-2.2.34-1.16.amzn1.x86_64  1/5
  Verifying   : apr-util-1.5.4-6.18.amzn1.x86_64     2/5
  Verifying   : httpd-2.2.34-1.16.amzn1.x86_64      3/5
  Verifying   : apr-1.5.2-5.13.amzn1.x86_64         4/5
  Verifying   : apr-util-ldap-1.5.4-6.18.amzn1.x86_64 5/5

Installed:
  httpd.x86_64 0:2.2.34-1.16.amzn1

Dependency Installed:
  apr.x86_64 0:1.5.2-5.13.amzn1
  apr-util.x86_64 0:1.5.4-6.18.amzn1
  apr-util-ldap.x86_64 0:1.5.4-6.18.amzn1
  httpd-tools.x86_64 0:2.2.34-1.16.amzn1

Complete!
ec2-user@ip-172-31-86-43 ~$
```

```
ec2-user@ip-172-31-86-43:~$ sudo yum install httpd
apr-util      x86_64      1.5.4-6.18.amzn1      amzn-main      96 k
apr-util-ldap x86_64      1.5.4-6.18.amzn1      amzn-main      19 k
httpd-tools   x86_64      2.2.34-1.16.amzn1      amzn-main      80 k

Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 3.6 M
Downloading packages:
(1/5): apr-util-ldap-1.5.4-6.18.amzn1.x86_64.rpm      | 19 kB  00:00
(2/5): apr-util-1.5.4-6.18.amzn1.x86_64.rpm          | 99 kB  00:00
(3/5): apr-1.5.2-5.13.amzn1.x86_64.rpm               | 118 kB  00:00
(4/5): httpd-tools-2.2.34-1.16.amzn1.x86_64.rpm      | 80 kB  00:00
(5/5): httpd-2.2.34-1.16.amzn1.x86_64.rpm            | 1.2 MB  00:00
-----
Total                                          2.2 MB/s | 1.5 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.5.2-5.13.amzn1.x86_64            1/5
  Installing : apr-util-1.5.4-6.18.amzn1.x86_64       2/5
  Installing : httpd-tools-2.2.34-1.16.amzn1.x86_64   3/5
  Installing : apr-util-ldap-1.5.4-6.18.amzn1.x86_64  4/5
  Installing : httpd-2.2.34-1.16.amzn1.x86_64        5/5
  Verifying   : httpd-tools-2.2.34-1.16.amzn1.x86_64  1/5
  Verifying   : apr-util-1.5.4-6.18.amzn1.x86_64     2/5
  Verifying   : httpd-2.2.34-1.16.amzn1.x86_64      3/5
  Verifying   : apr-1.5.2-5.13.amzn1.x86_64         4/5
  Verifying   : apr-util-ldap-1.5.4-6.18.amzn1.x86_64 5/5

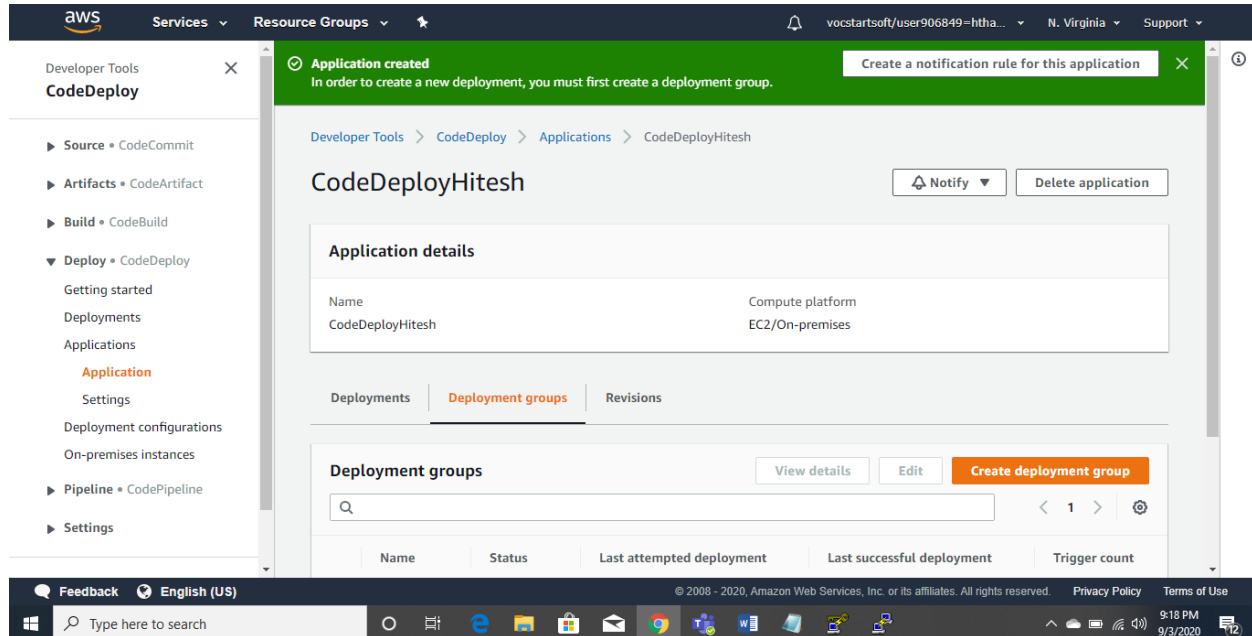
Installed:
  httpd.x86_64 0:2.2.34-1.16.amzn1

Dependency Installed:
  apr.x86_64 0:1.5.2-5.13.amzn1
  apr-util.x86_64 0:1.5.4-6.18.amzn1
  apr-util-ldap.x86_64 0:1.5.4-6.18.amzn1
  httpd-tools.x86_64 0:2.2.34-1.16.amzn1

Complete!
ec2-user@ip-172-31-86-43 ~$ sudo service httpd start
Starting httpd: [ OK ]
ec2-user@ip-172-31-86-43 ~$ sudo service httpd restart
Stopping httpd: [ OK ]
Starting httpd: [ OK ]
ec2-user@ip-172-31-86-43 ~$
```

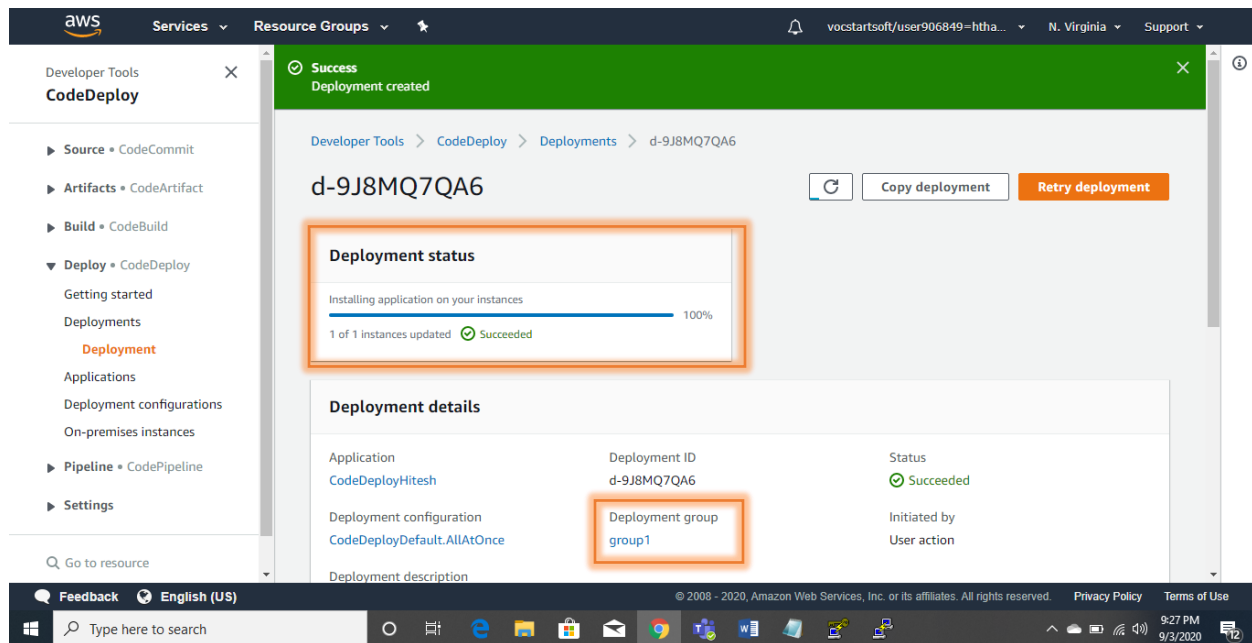
Setting up CodeDeploy to create a new application

Created a New Application CodeDeployHitesh on EC2/on-premise cloud



Deployment was successful. Created Code Deployment Group “group1” by selecting Create deployment group with Service Role as codedeployrole created at the 1st step and Deployment type as “In-place”. Configuring this Code Deployment Group with “Key” as “Name” and “Value” as “Web Server” for facilitating the EC2 instance for code deployment with deployment setting set to CodeDeployDefault.AllAtOnce along with Disabled Load Balancer.

Deployment created setting the Revision type as “My application is stored in Github” and inputting the Github token name type as “mlabouardy” along with specifying the Repository name and commit ID for the instance to know the exact Github Repository.



Using the IPV4 Public IP of the EC2 instance in the browser to access the application deployed in the AWS cloud. The deployment is successful so I was able to access the application.

