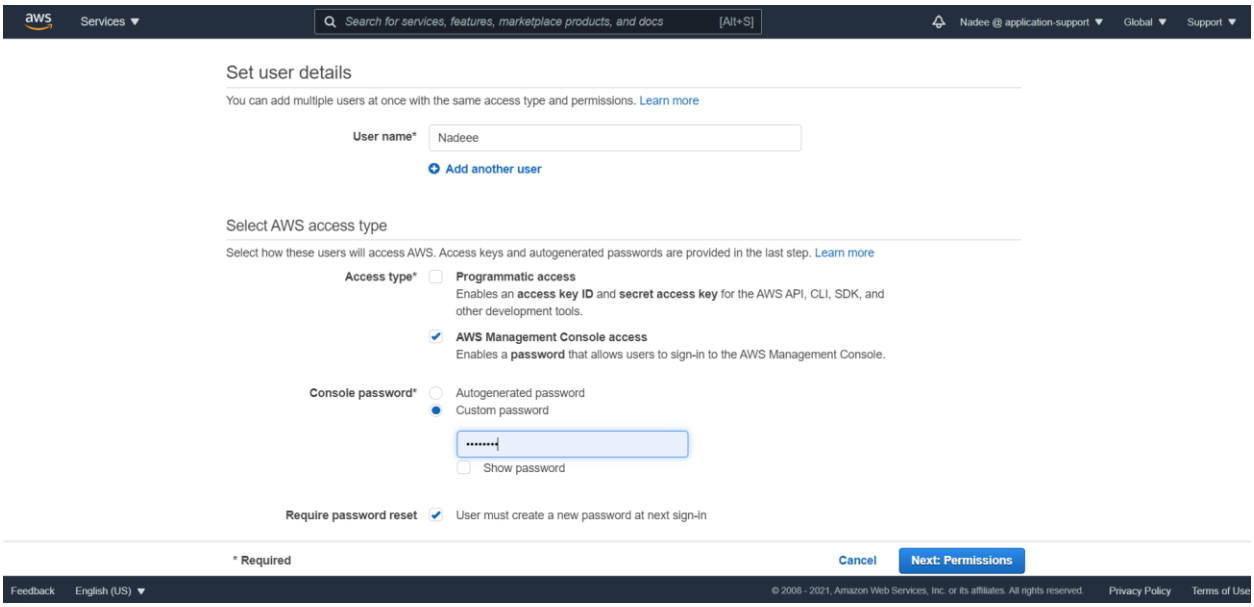
CAPITAL MARKETS APPLICATION SUPPORT ASSIGNMENT

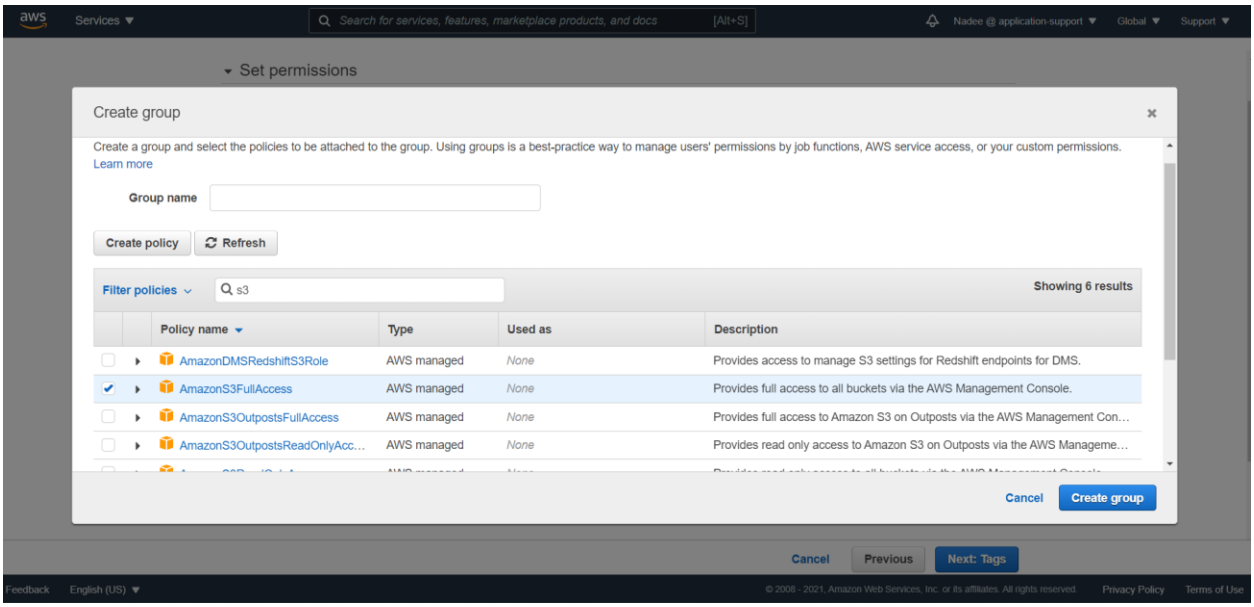
Nadee Ranasinghe

Contents

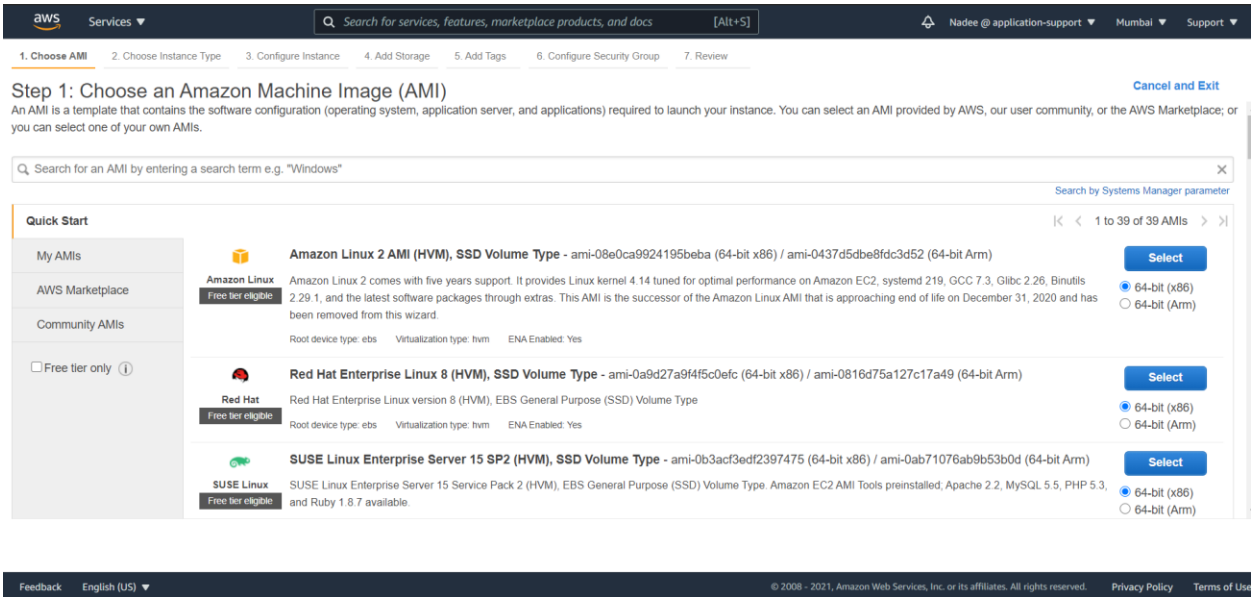
IAM User

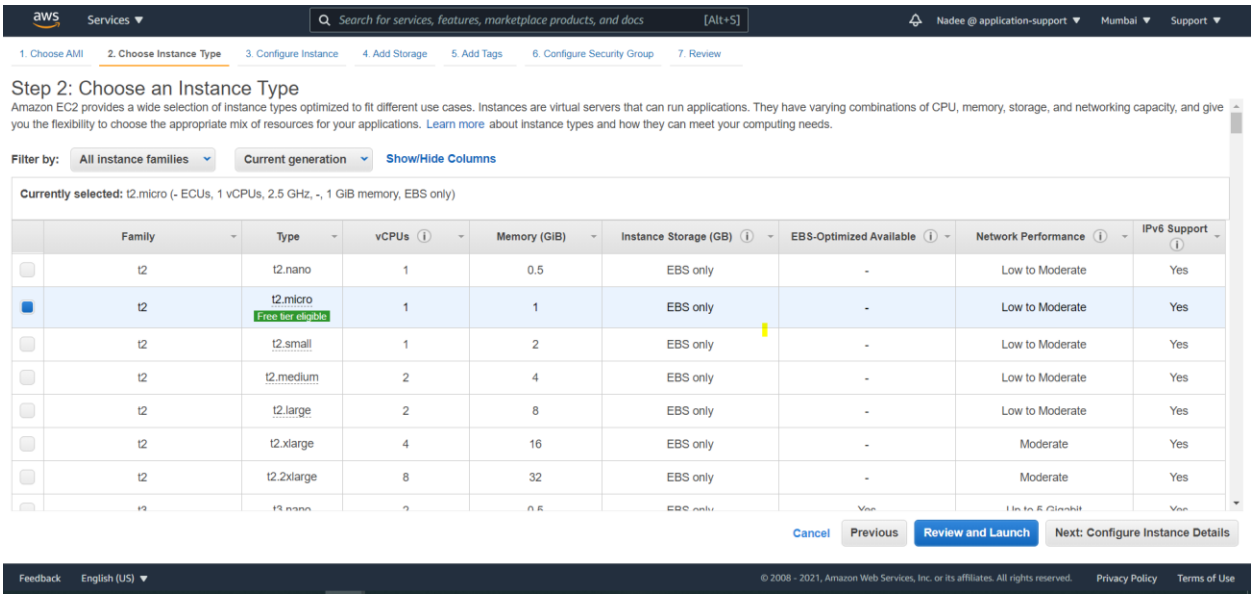


AmazonS3FullAcesss policy is attached to the newly created group.

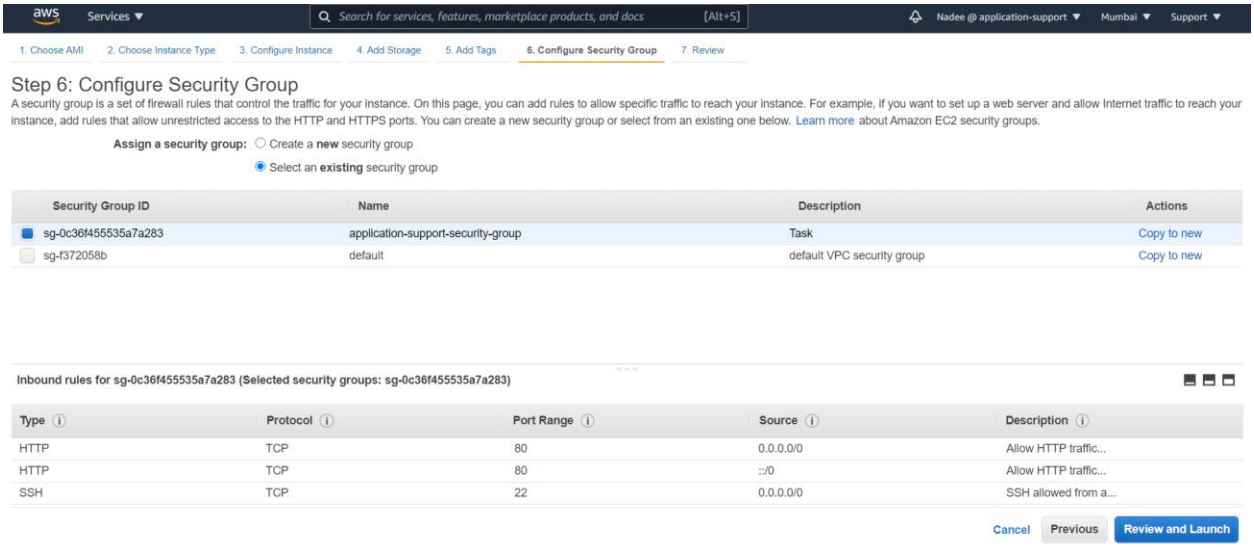


EC2 Instance



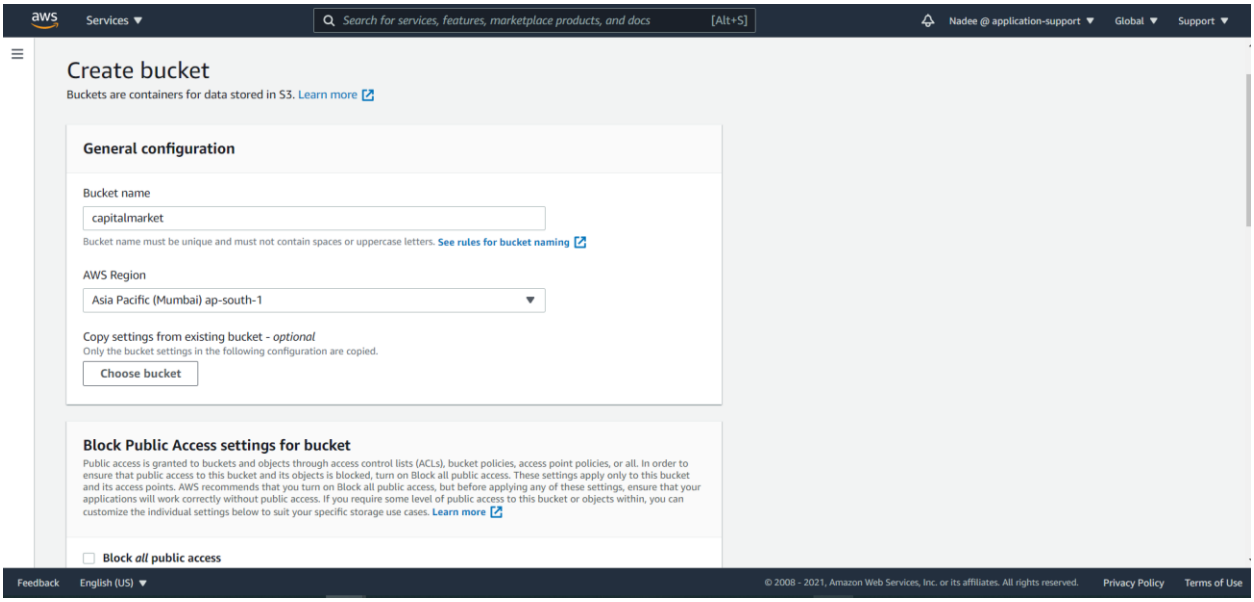


New security group is created with following inbound and outbound rules.



S3 Bucket

A S3 bucket was created to log the timestamps.

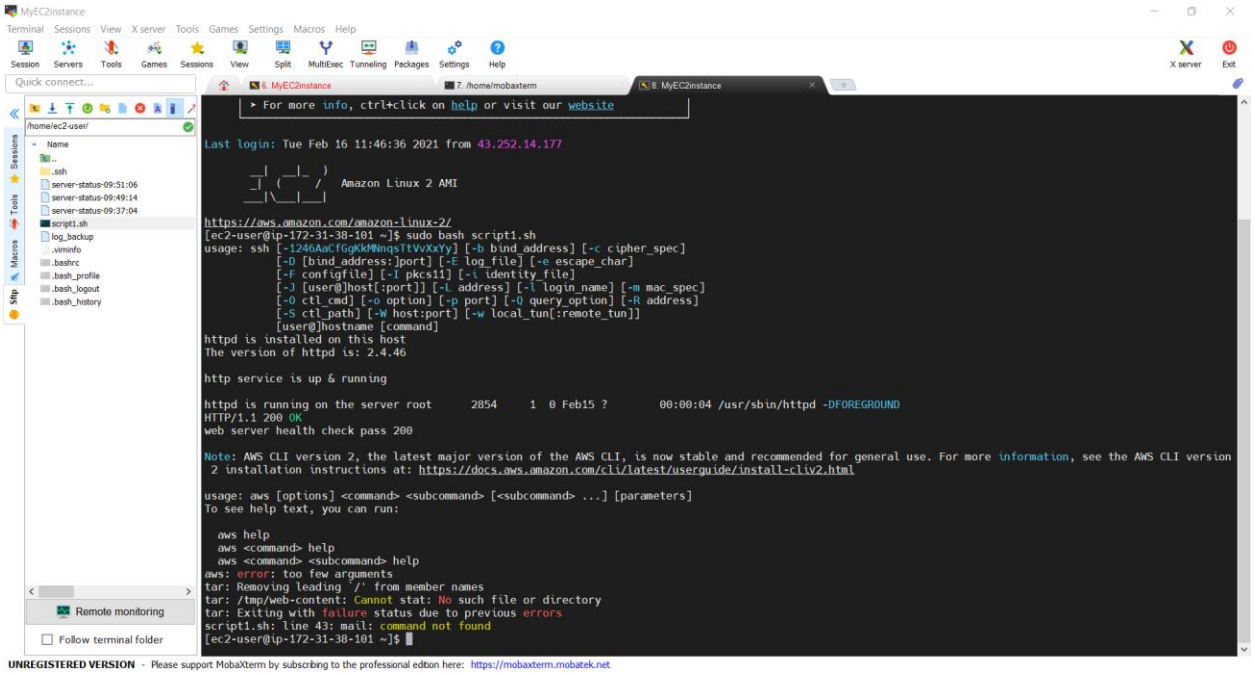


Script

```
#!/bin/bash
#shell script
status=$(echo $?)
server_health=$(curl -o /dev/null -s -w '%{http_code}'
http://localhost:80)
pid=$(ps -ef | grep httpd | grep -v grep | head -n 1)

ssh -t -o StrictHostKeyChecking=No $server_IP ;

######## check apache web server installed or not

if which httpd &>/dev/null
then
    echo "httpd is installed on this host"
    http_ver=$(httpd -v | awk -F '[ /]' '/version/ { print $4 }')
    echo "The version of httpd is: $http_ver"
else
   echo "httpd is not installed"

fi


#### add content to document root

echo "Hello Welcome to Capital Market Website !!!" >
/var/www/html/index.html

echo ""

#######check service status

systemctl status httpd &> /dev/null && echo "http service is up & running"
|| sudo systemctl start httpd &> /dev/null


###### Process ID of httpd process

echo ""
echo "httpd is running on the server $pid"

####Loading web server content using curl

if curl -I "http://localhost:80" 2>&1 | grep -w "200" ; then
    echo "web server health check pass $server_health" #### display health
with the status code
       curl -o server-status-$(date +%T) http://localhost:80 2> /dev/null

#### get web server content in to file with the timestamps
       echo ""
       aws s3 cp $filename s3://mybucket/ #### copy content to s3
```

```
else
    echo "web server is down"
       echo "web server health check falied" | mail -s "health check"
username@example.com ### email for app team
fi
```

```
#### please add IAM role for EC2 instance with S3 full access####

 ############ logs archive ################
logpath=/var/log/httpd/
tempdir=/tmp/web-content/
find $logpath/* -type -f -name "*.log" -mtime 1 -exec cp {} $tempdir/ \;
&> /dev/null
tar -cvzf log_backup -c $tempdir . > /dev/null
aws s3 cp log_backup $bucketname/web-content-$(date +%F).tar &>/dev/null
&& rm -rf web-content.tar || echo "Upload failed"  | mail -s "logs upload
failed" $emailid
rm -rf $tempdir
```



Checking the status of the web server and displaying the content with timestamps.

Supplementary requirements

Below is a cloud formation template that can be used to automate this. But this needs to be further developed.

```
Parameters:
  NameOfService:
    Description: "The name of the service this stack is to be used for."
    Type: String
Parameters:
  myKeyPair:
    Description: Amazon EC2 Key Pair
    Type: "AWS::EC2::KeyPair::KeyName"
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket'
    DeletionPolicy: Retain
    Properties:
      AccessControl: PublicRead
      BucketName: web-server-2021
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            yum:
              httpd: []
          services:
            sysvinit:
              httpd:
                enabled: true
                ensureRunning: true
```

```
    Properties:
      InstanceType: t2.micro
      ImageId: ami-7a11e213
      SecurityGroupIds:
        - !Ref MySecurityGroup
  MySecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Open Ports 22 and 80
      SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: '22'
        ToPort: '22'
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp: 0.0.0.0/0
Outputs:
  servername:
    Description: The Public DNS for the EC2 Instance
    Value: !Sub 'http://${EC2Instance.PublicDnsName}'
```