

Steps for STS > Docker > AWS Code Commit > EC2 > RUN

```
#pom.xml : Add <finalName>demo</finalName>

<build>

    <plugins>

        <plugin>

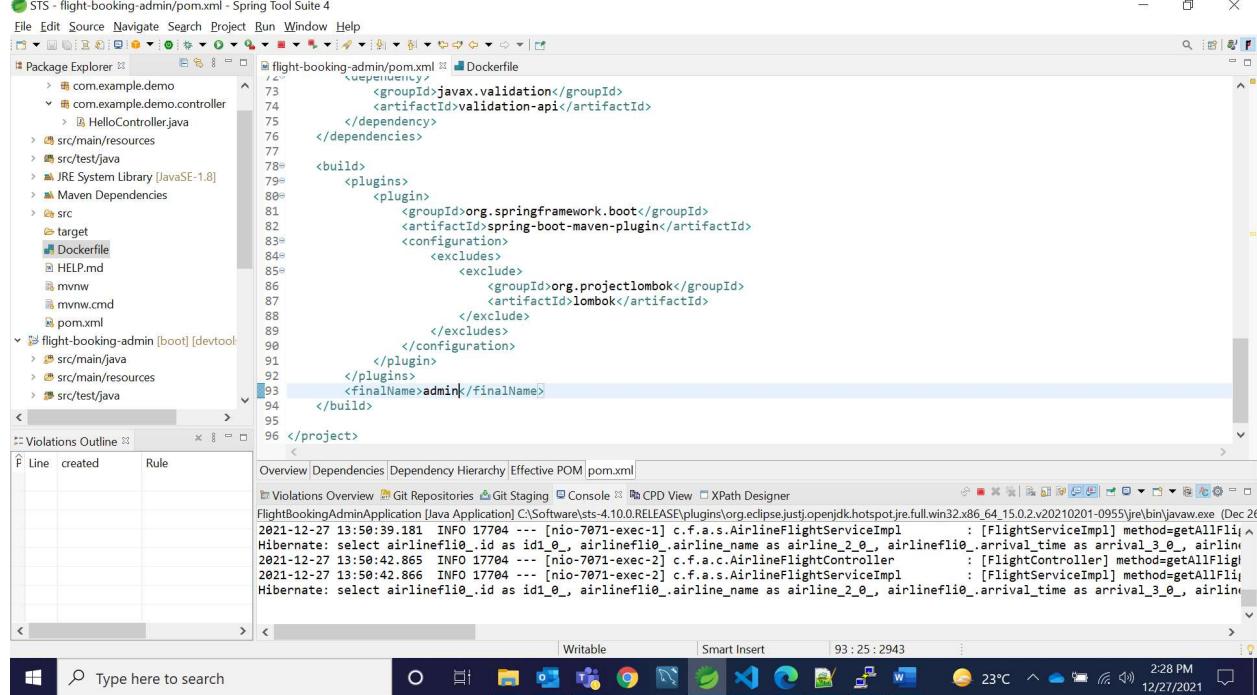
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>

        </plugin>

    </plugins>

    <finalName>demo</finalName>

</build>
```



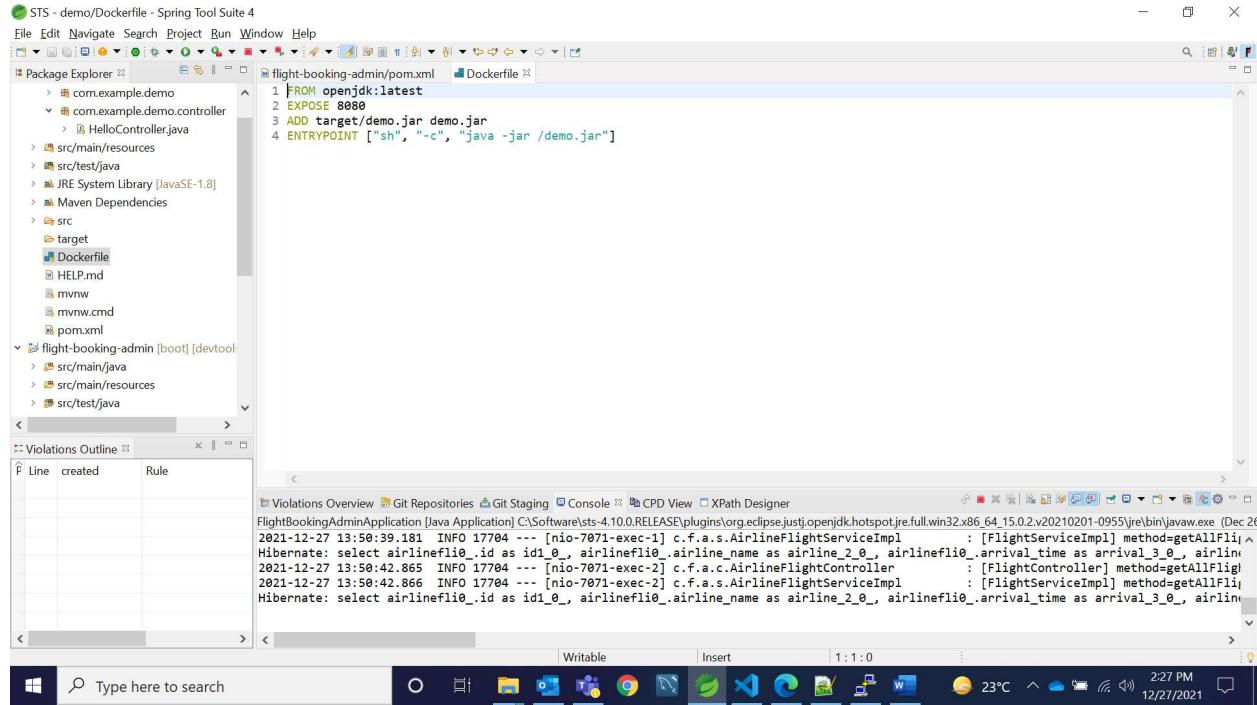
#Dockerfile : Add Dockerfile in source of Application Microservice

```
FROM openjdk:latest
```

```
EXPOSE 8080
```

```
ADD target/admin.jar admin.jar
```

```
ENTRYPOINT ["sh", "-c", "java -jar /admin.jar"]
```



#Create EC2 Instances in AWS

Security Group SSH = All & All Traffic : Anywhere

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 name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	Anywhere	e.g. SSH for Admin Desktop

Add Rule

Warning

Cancel Previous Review and Launch

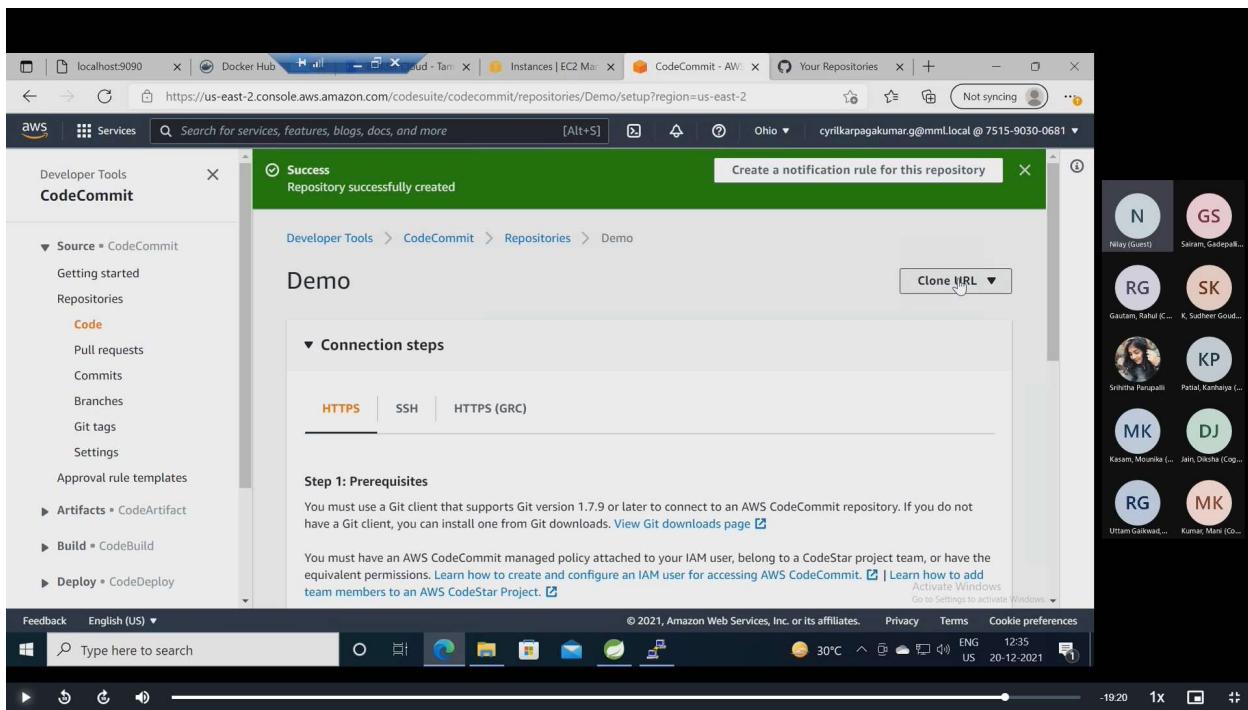
And Generate Private Key Pair & Download it

AWS Code Commit:

Create a Repository

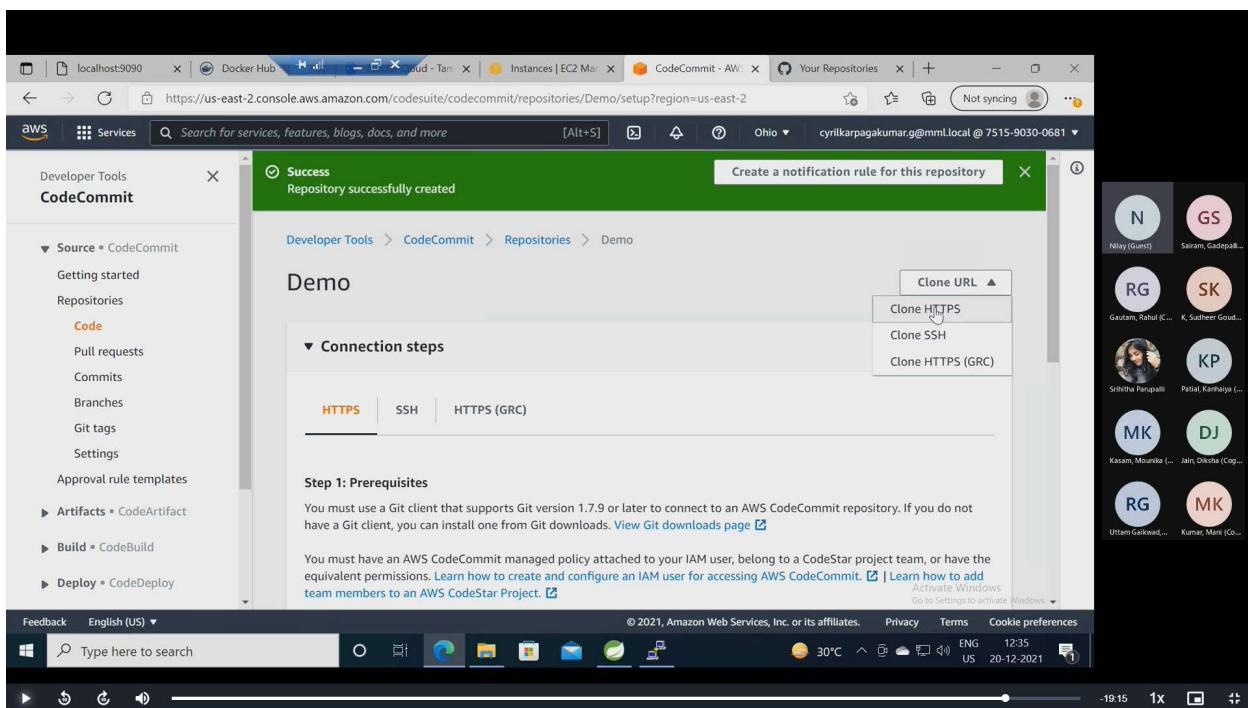
The screenshot shows the AWS CodeCommit interface. On the left, a sidebar titled 'Developer Tools' has 'CodeCommit' selected. Under 'Source', 'Repositories' is also selected. The main area displays a table with columns: Name, Description, Last modified, and Clone URL. A large orange 'Create repository' button is prominently displayed at the top right of the table area. To the right of the table, there is a sidebar showing user profiles with initials and names.

The screenshot shows the 'Create repository' dialog box. It has fields for 'Repository name' (set to 'Demo'), 'Description - optional' (empty), 'Tags' (empty), and a checkbox for 'Enable Amazon CodeGuru Reviewer for Java and Python - optional'. At the bottom right of the dialog is a 'Create' button. The background shows the same AWS CodeCommit interface as the previous screenshot.



Clone URL

Clone HTTPS



Now IAM for Permission

Click on User

The screenshot shows the AWS IAM Dashboard. On the left, the navigation menu is open, with 'Users' selected under 'Access management'. The main area displays the 'IAM dashboard' with sections for 'Security recommendations', 'IAM resources', and 'What's new'. A prominent red warning box at the top right says 'Add MFA for root user'. The 'Users' section shows 3 users listed: 'CloudOpsUbsAdmin', 'cyrilkarpagakumar.g@mml.local', and 'iamprepaidinc+2126'. The 'IAM resources' section shows 2 user groups, 3 users, 29 roles, 12 policies, and 0 identity providers.

Choose AWS user name

The screenshot shows the 'Users' list page in the AWS IAM Dashboard. The navigation bar shows 'IAM > Users'. The main table lists three users: 'CloudOpsUbsAdmin', 'cyrilkarpagakumar.g@mml.local', and 'iamprepaidinc+2126'. The 'cyrilkarpagakumar.g@mml.local' user is highlighted in blue, indicating it is selected. The table includes columns for 'User name', 'Groups', 'Last activity', 'MFA', and 'Password a...'. A search bar at the top of the table allows filtering by username or access key.

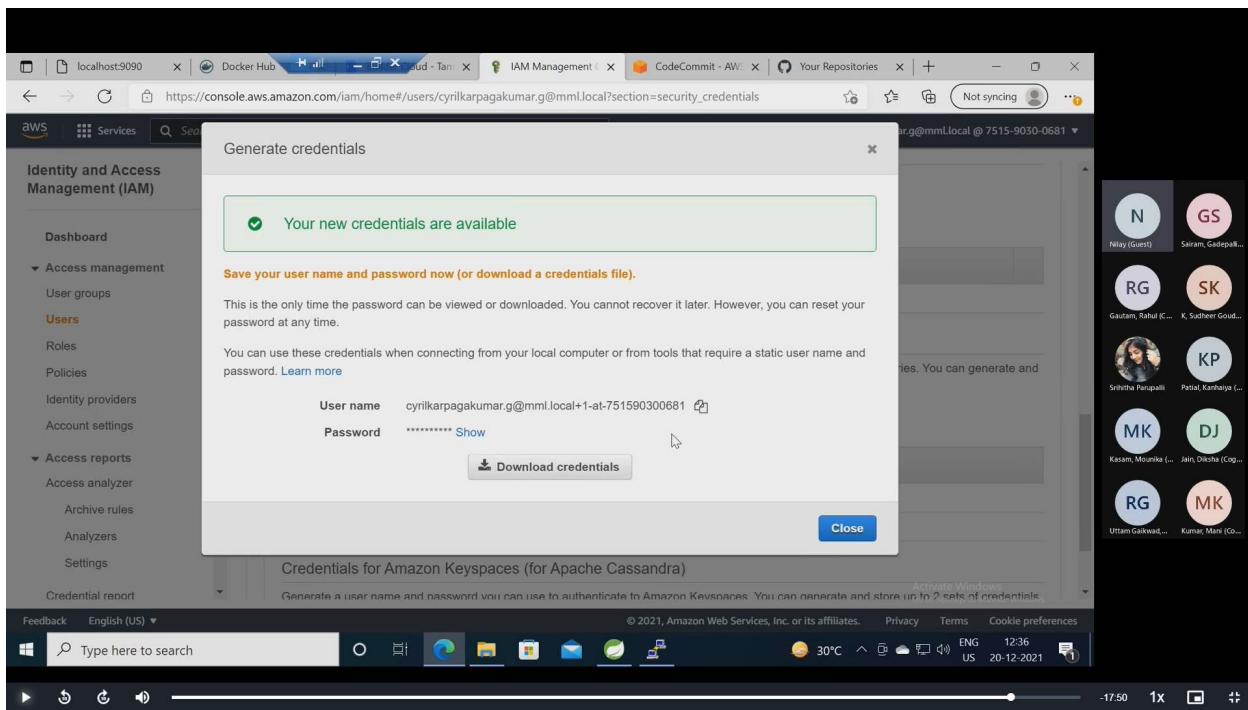
Security Credentials

The screenshot shows the AWS IAM Management console for a user named 'cyrilkarpagakumar.g@mml.local'. The left sidebar shows navigation options like Dashboard, Access management, and Access reports. The main area displays user details: User ARN (arn:aws:iam::751590300681:user/cyrilkarpagakumar.g@mml.local), Path (/), and Creation time (2021-10-29 09:49 UTC+0530). Below this, tabs for Permissions, Groups (1), Tags, Security credentials, and Access Advisor are visible. Under Permissions, it shows 'Permissions policies (3 policies applied)'. One policy listed is 'AWSCompromisedKeyQuarantineV2' attached directly. A 'Show 2 more' link is present. A 'Permissions boundary (not set)' section follows. On the right, a grid of user icons and names is shown, including Nilay (Guest), Sairam, Gadepalli, Gautam, Rahul, K. Sudheer Goud, Sritha Pangali, Patal, Karthika, Kasam, Mounika, Jain, Dilchha, Utam Gaikwad, and Kumar, Mani. The bottom of the screen shows a Windows taskbar with various pinned icons.

Generate Credentials

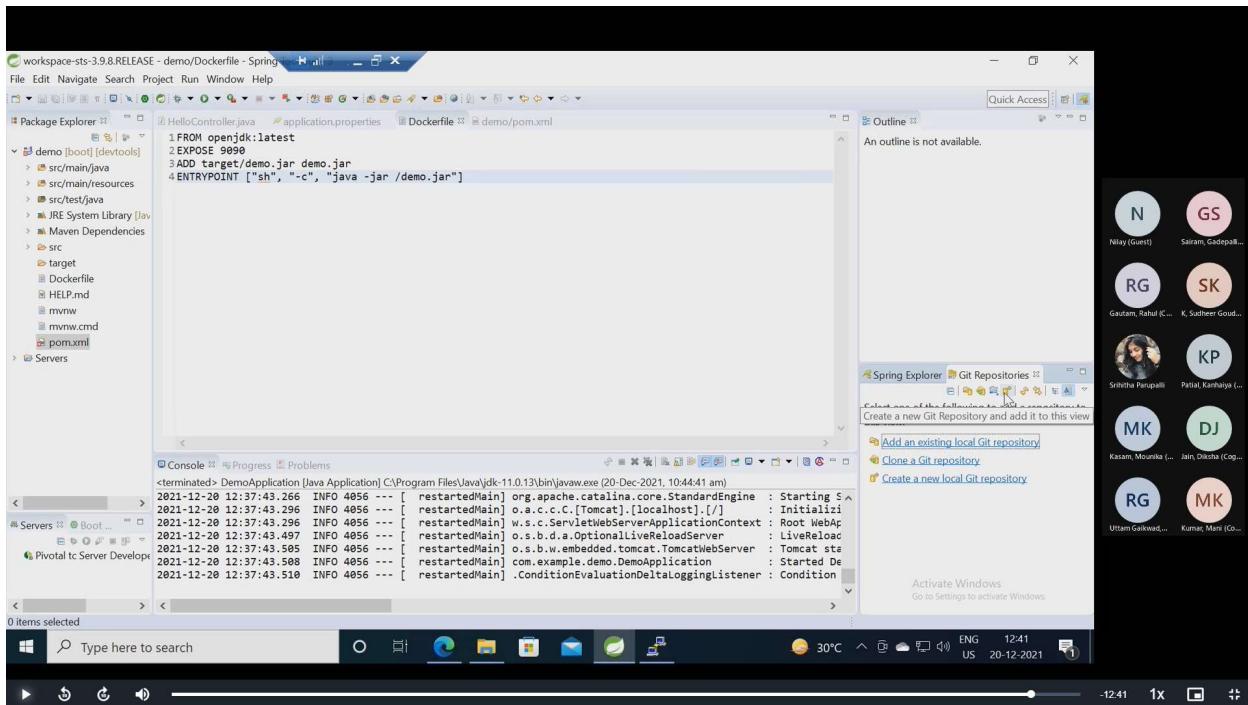
The screenshot shows the AWS IAM Management console under the 'Generate Credentials' section. It includes instructions for using SSH public keys to authenticate access to AWS CodeCommit repositories. Below this, there's a table for 'SSH key ID' with columns for 'Uploaded' and 'Status', showing 'No results'. The next section, 'HTTPS Git credentials for AWS CodeCommit', provides instructions for generating user names and passwords for HTTPS connections. It includes a 'General' dropdown and an 'Actions' button. The final section, 'Credentials for Amazon Keyspaces (for Apache Cassandra)', also provides instructions for generating credentials for Amazon Keyspaces. A note states 'No credentials have been generated.' The interface is consistent with the previous screenshot, featuring the same sidebar and Windows taskbar at the bottom.

Download



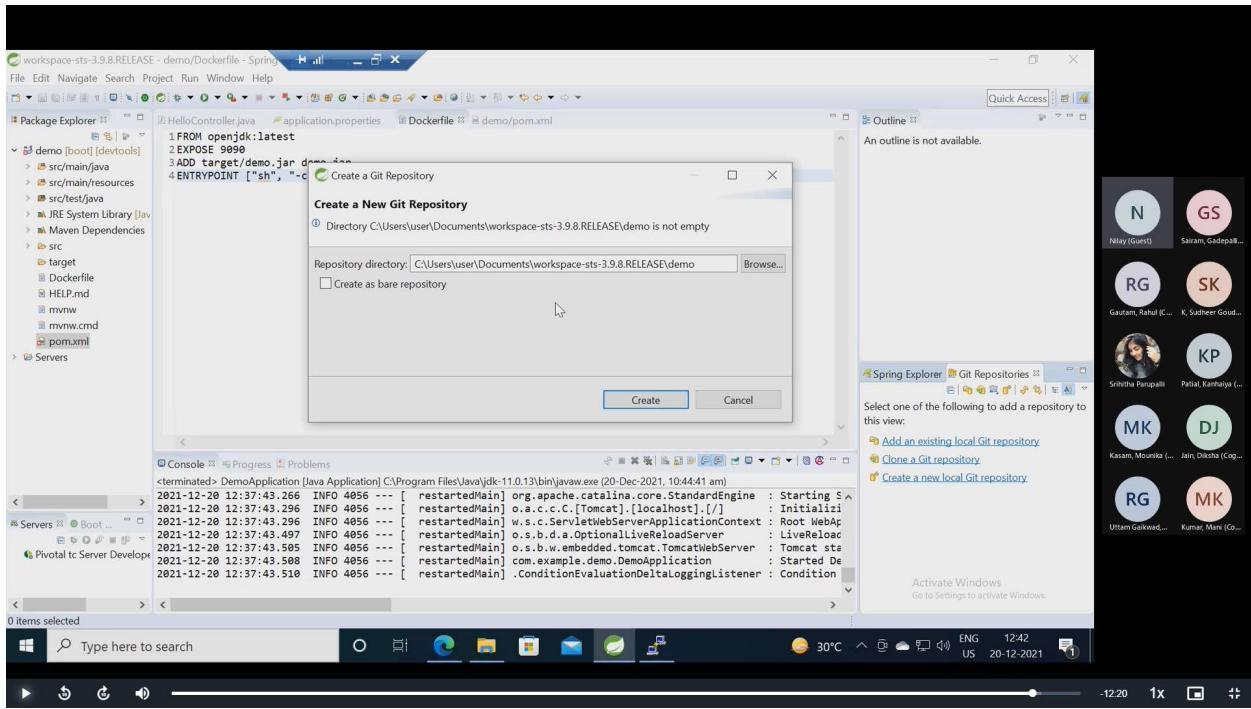
NOW in STS

Window > Show view > git

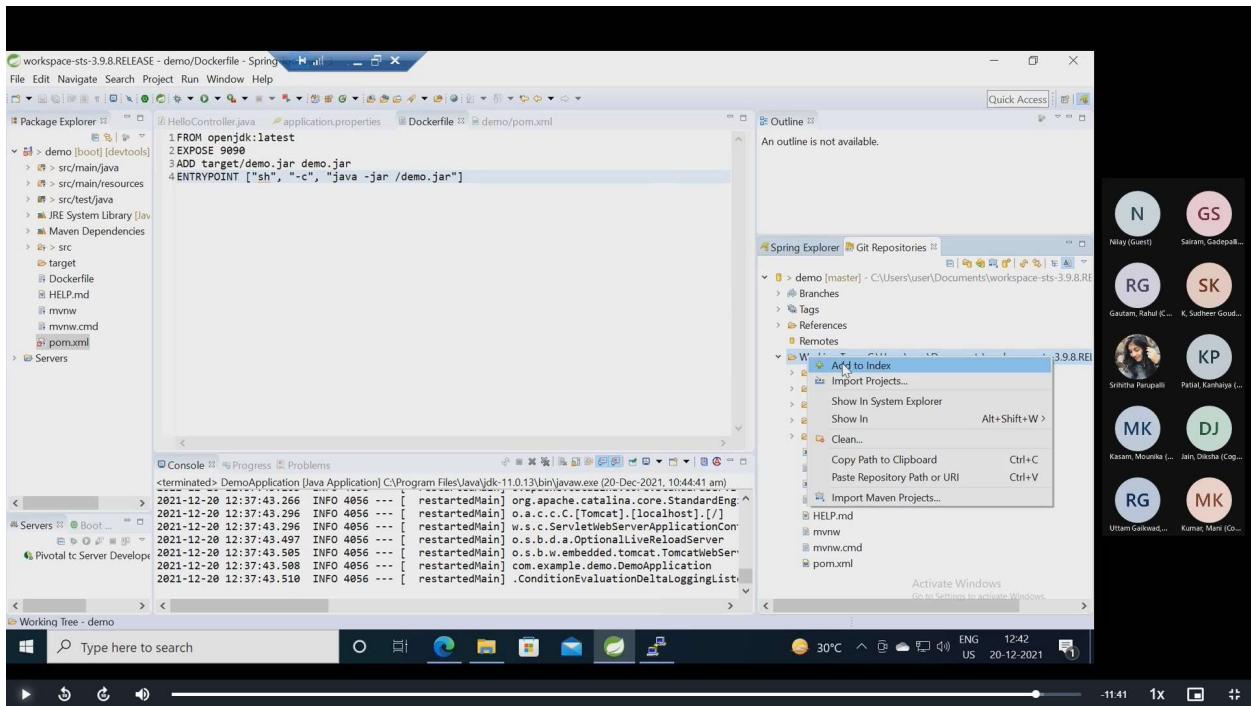


Click on Create a new Git Repository and add icon

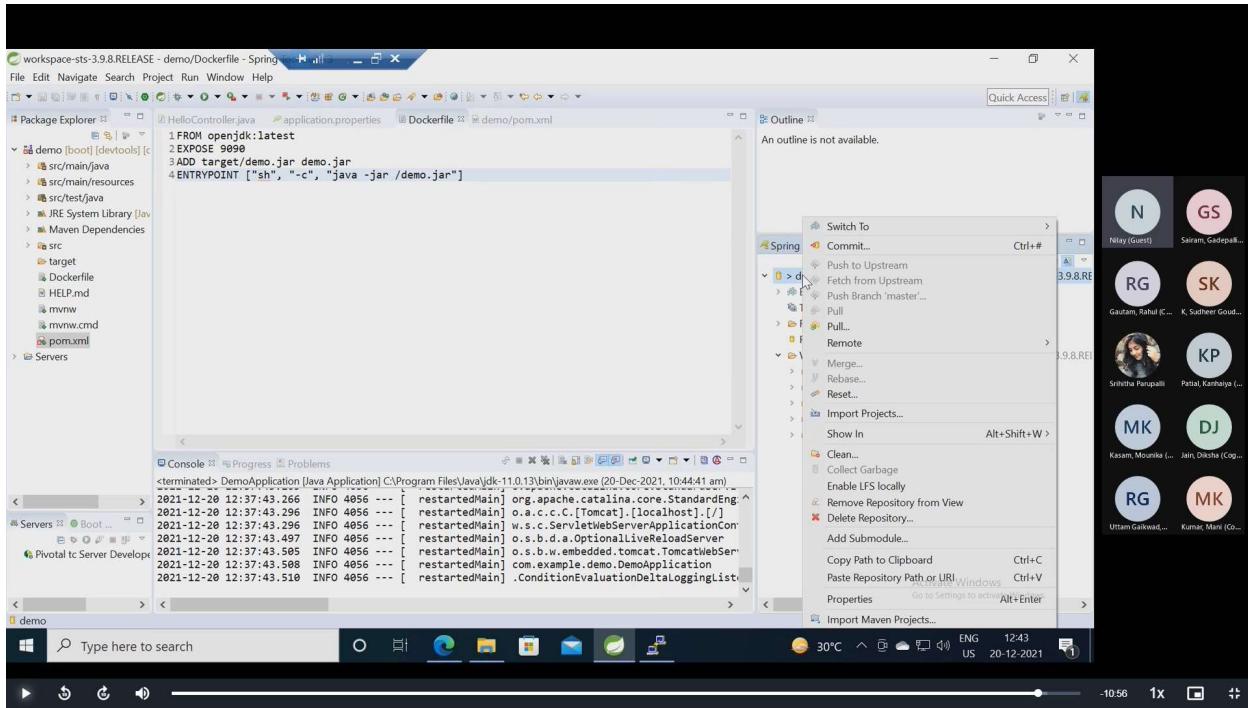
Browse and choose where Application Microservice is located



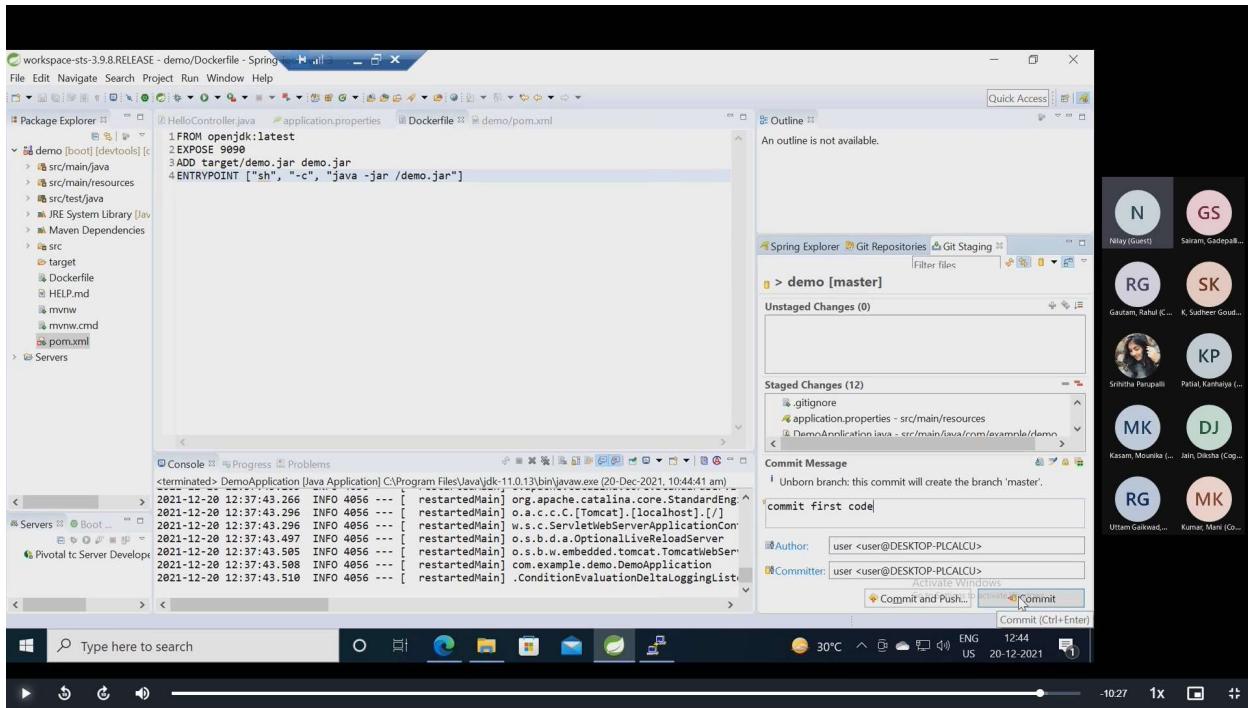
Right Click on working tree > add to index



Right click on App name (Ex : Demo) > Commit



Commit msg and click on commit



Then Again right click on Demo > Push Branch Master

Give the URL of AWS Code Commit ()

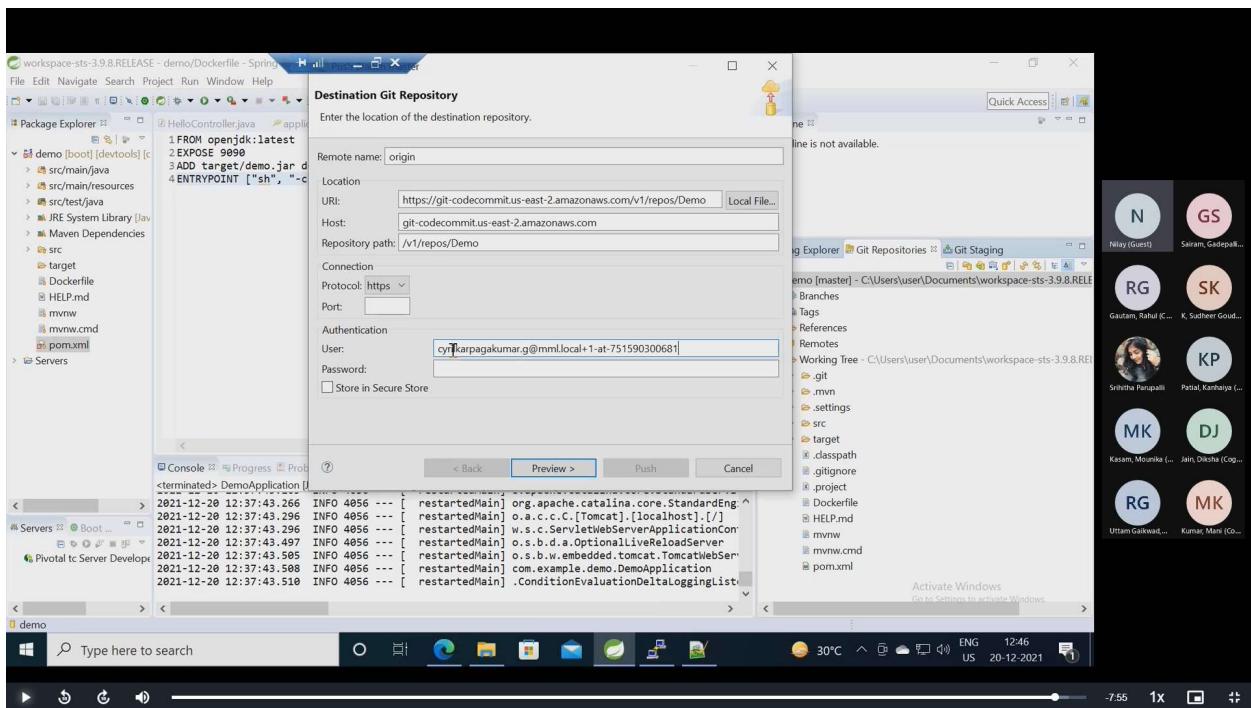
User id n Password which is downloaded by AWS IAM excel file

The screenshot shows the AWS CodeCommit console. A modal window is open with two notifications: "Success" (Repository successfully created) and "Copied" (the repository URL). The URL is <https://git-codecommit.us-east-2.amazonaws.com/v1/repos/Demo>. Below the URL, there's a "Copy" button. To the right, a sidebar shows user profiles: Nilay (Guest), Sairam, Gadepalli, Gautam, Rahul, K. Sudheer Goud, Srilatha Pangali, Patali, Kanthaja, Kasam, Mounika, Jain, Dilchha, Uttam Gaikwad, and Kumar, Mani (Owner). On the left, a sidebar menu is visible with sections like Source, Code, Pull requests, Commits, Branches, Git tags, Settings, Approval rule templates, Artifacts, Build, and Deploy.

The screenshot shows an Excel spreadsheet titled "cyrilkarpagakumar.g@mml.local_codecommit_credentials". The spreadsheet contains two rows of data:

	User Name	Password
1	cyrilkarpagakumar.g@mml.local+1-at-4515958888888888	nCuVcYoBss4tpd15xPvfTIVVe31dsv+ge5kVcE2BdQ

The status bar at the bottom indicates "Ready" and shows system information: 30°C, 3:17 PM, 12/27/2021, 23°C, and 100%.



Check Code Commit in AWS

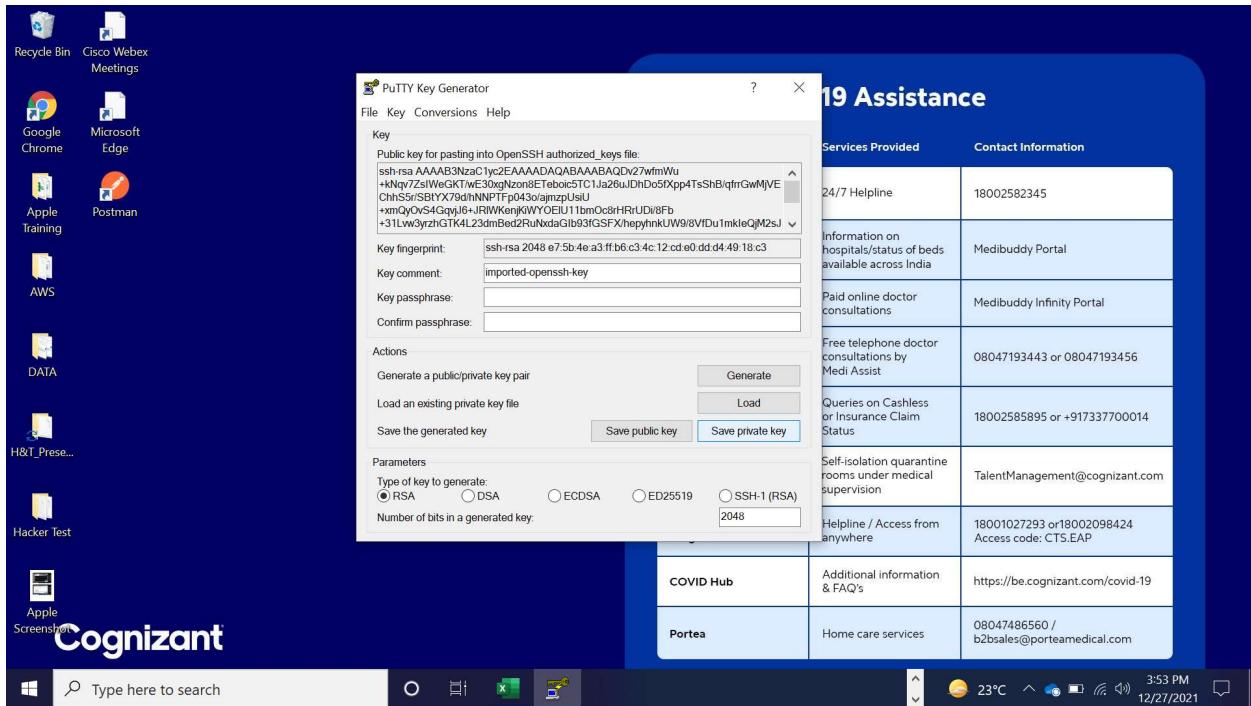
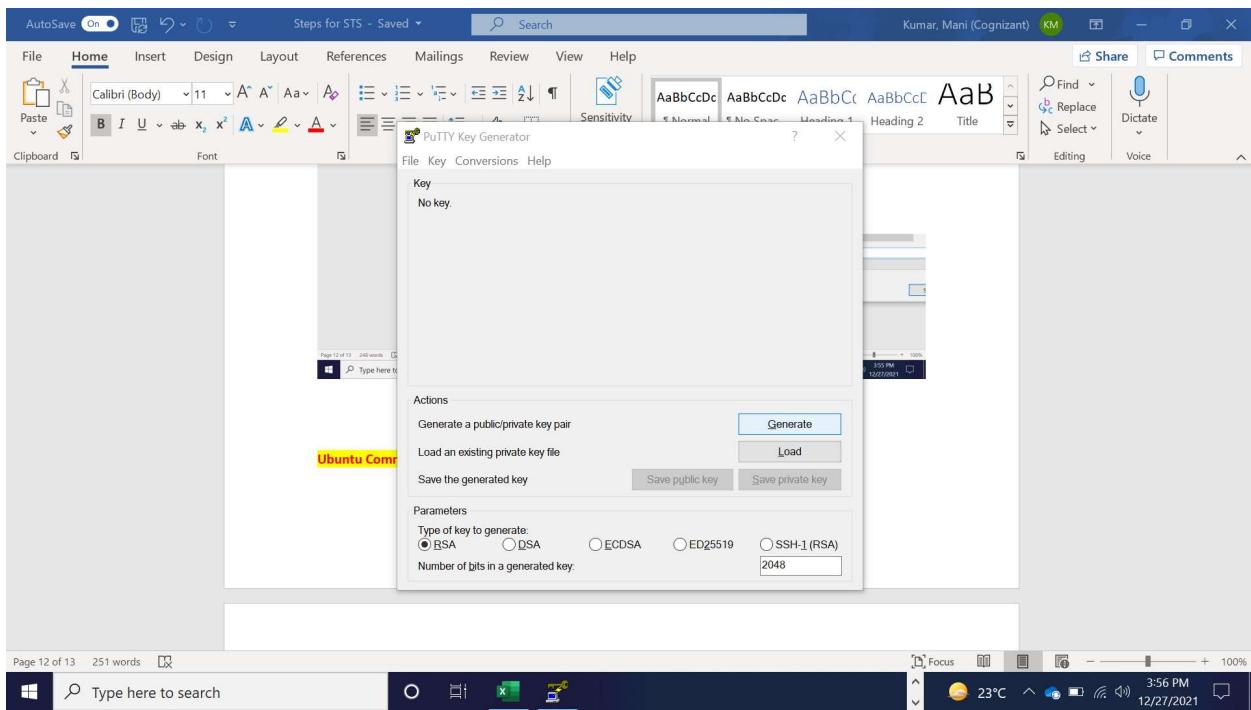
The screenshot shows the AWS CodeCommit console. The left sidebar navigation includes 'Developer Tools' and 'CodeCommit'. Under 'CodeCommit', the 'Repositories' section is selected. The main content area displays the 'Repositories' page with the following details:

Name	Description	Last modified	Clone URL
Demo	-	5 days ago	HTTPS SSH HTTPS (GRC)

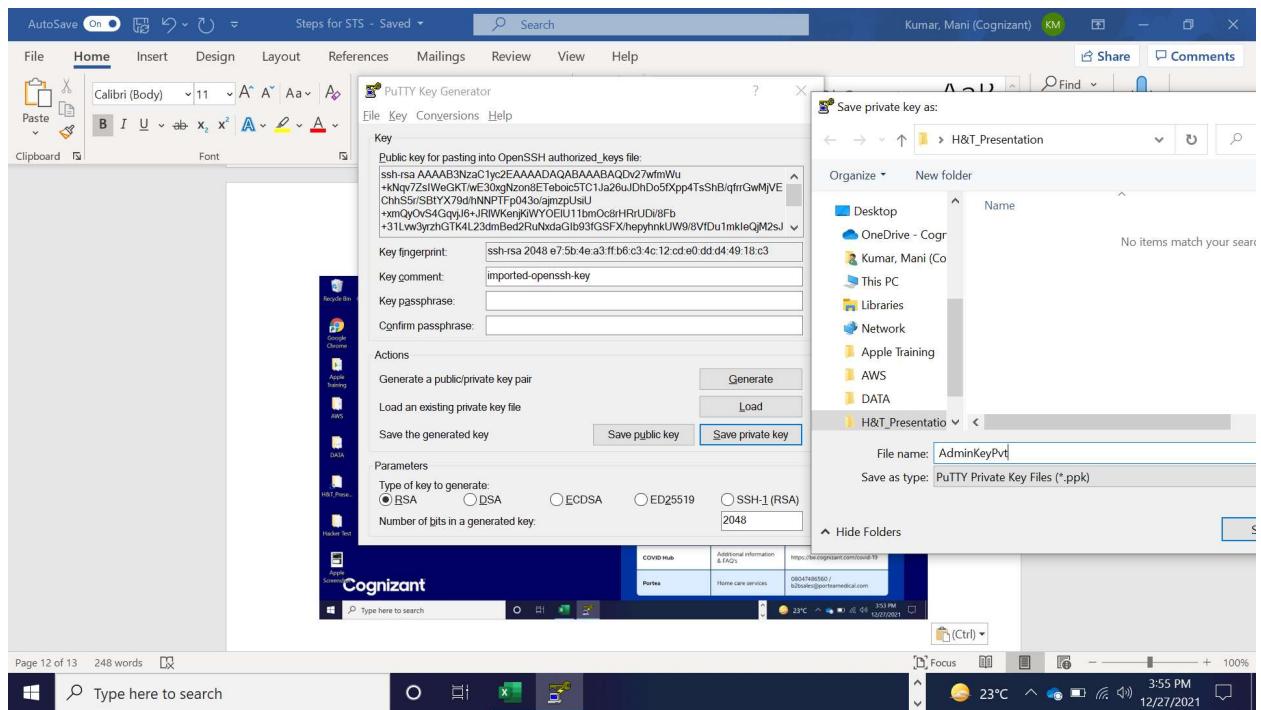
The bottom of the screen shows the Windows taskbar with various pinned icons.

Putty Gen

Generate Private Key

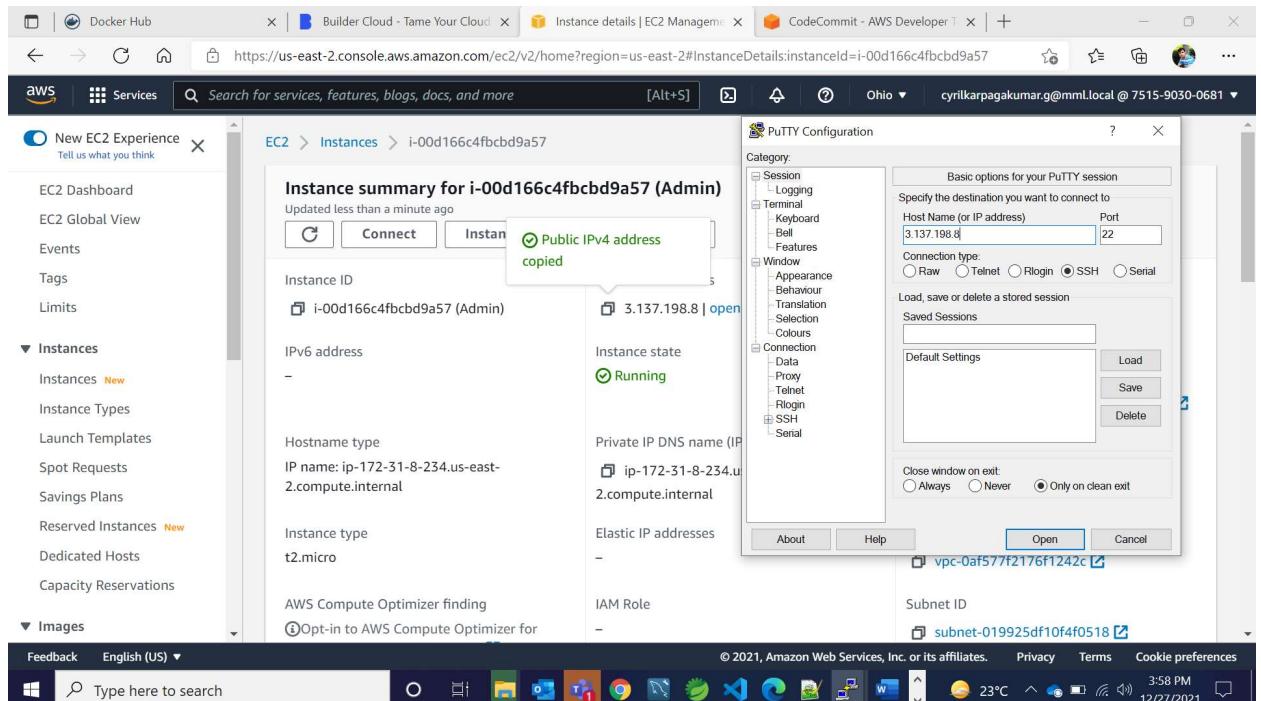


Save Private Key

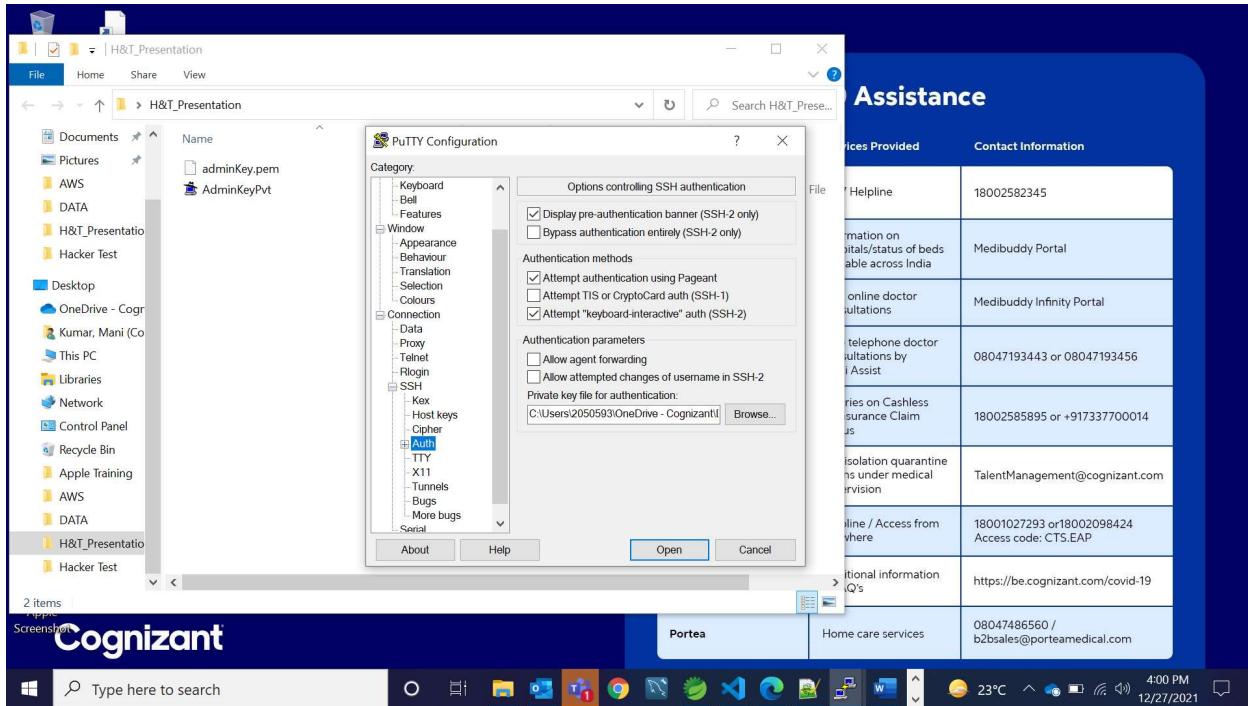


Putty

Copy AWS HostName IP and paste in Putty



Then Click SSH>Auth and Browse path of Private Key



Login As ubuntu

```
ubuntu@ip-172-31-8-234:~$ 
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-1060-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support:      https://ubuntu.com/advantage

System information as of Mon Dec 27 10:31:35 UTC 2021

System load: 0.0          Processes:         93
Usage of /: 15.2% of 7.69GB   Users logged in: 0
Memory usage: 19%           IP address for eth0: 172.31.8.234
Swap usage: 0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

0 updates can be applied immediately.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-8-234:~$ 
```

Ubuntu Command

Login as ubuntu

sudo su

apt update -y

apt install docker.io

y

copy the code link from code commit https

git clone and link

enter username and password AWS code commit

cd reponame (Ex: cd Admin)

apt install maven

mvn clean package -DskipTests=true

docker build -t dockerid name/servicename . (this dot should be in cmd)

Ex: docker build -t maniranjangupta/demoservice .

docker build -t maniranjangupta/adminservice .

docker images

docker login (No need, if we want to run from EC2 container)

docker container run -p8080:8080 imagename:latest

Ex : docker container run -p8080:8080 maniranjangupta/adminservice:latest

and run the task

<http://18.118.247.226:8080/hello/show>