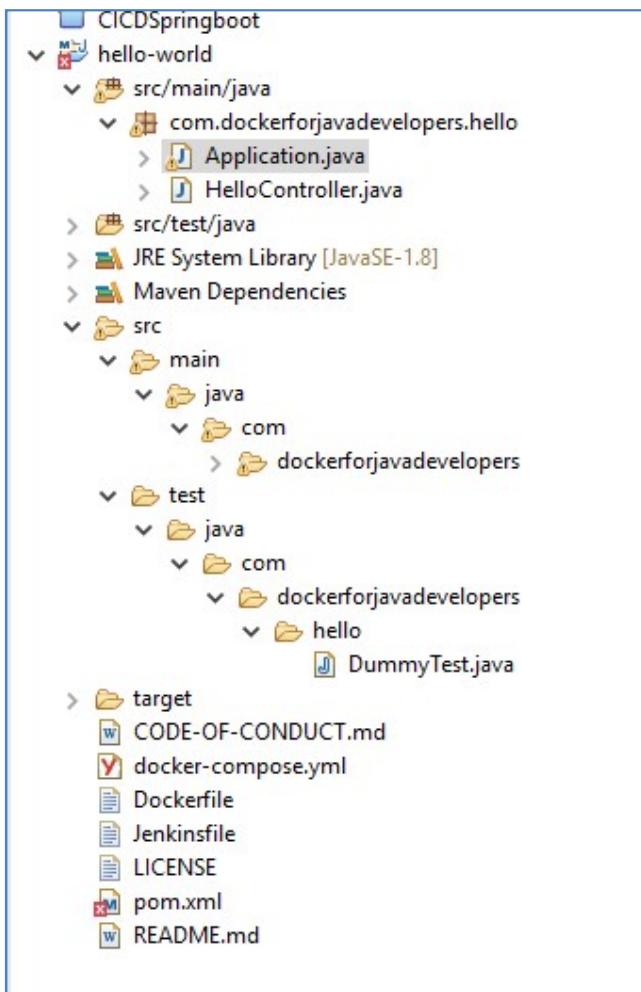


Phase-5
Testing in a DevOps Lifecycle

Project
**CI/CD Deployment for
spring boot Application**

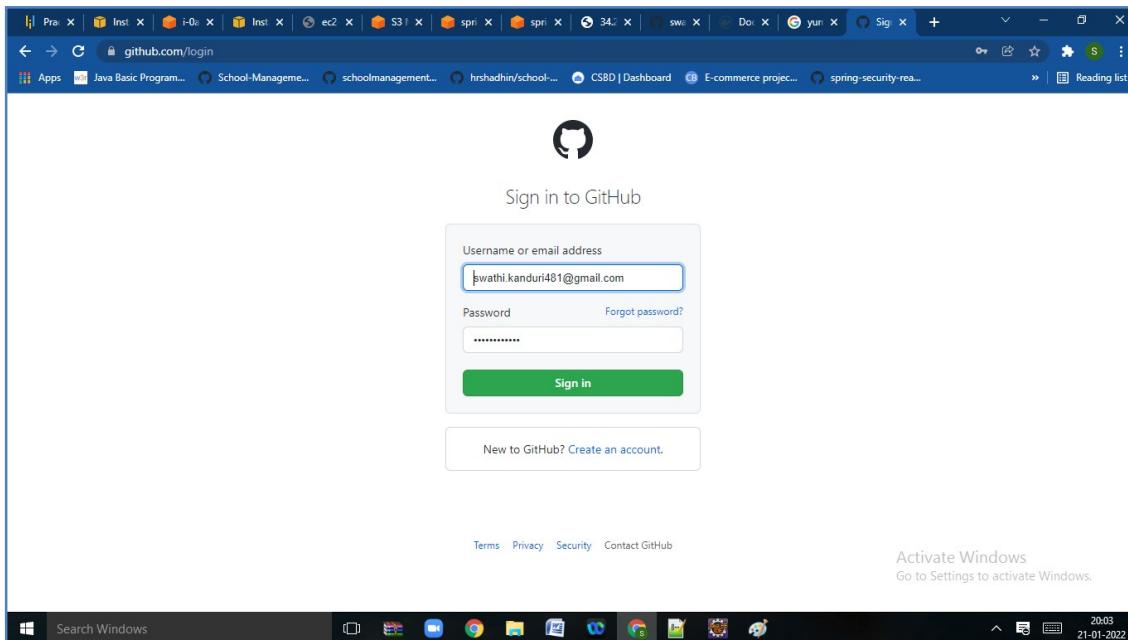
Application Screenshots

Spring boot application project explorer

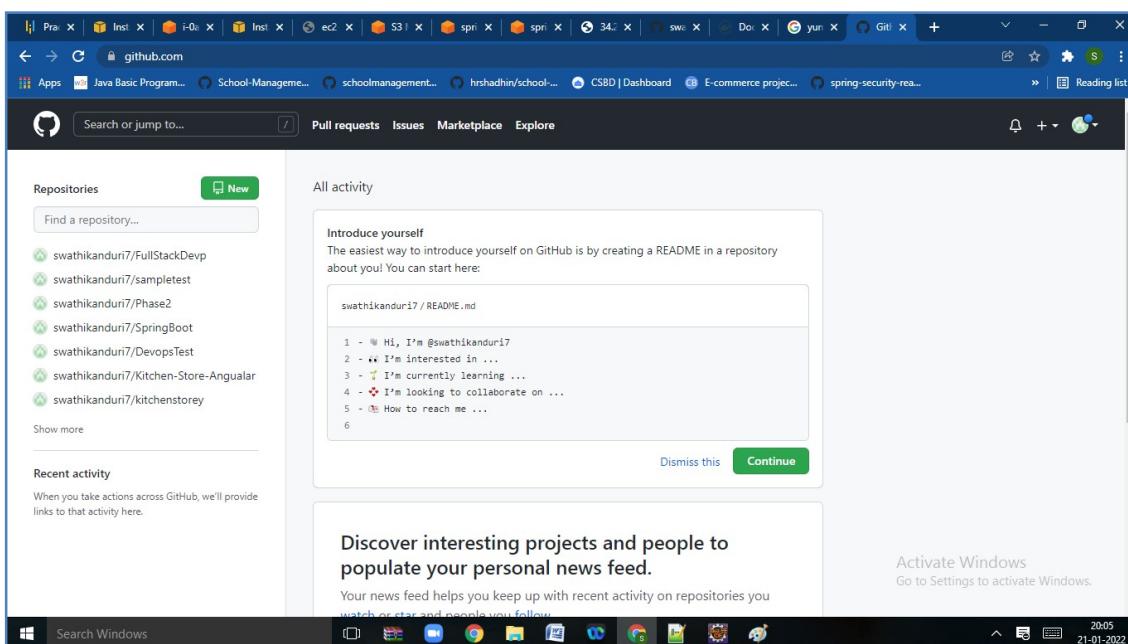


Git hub:-

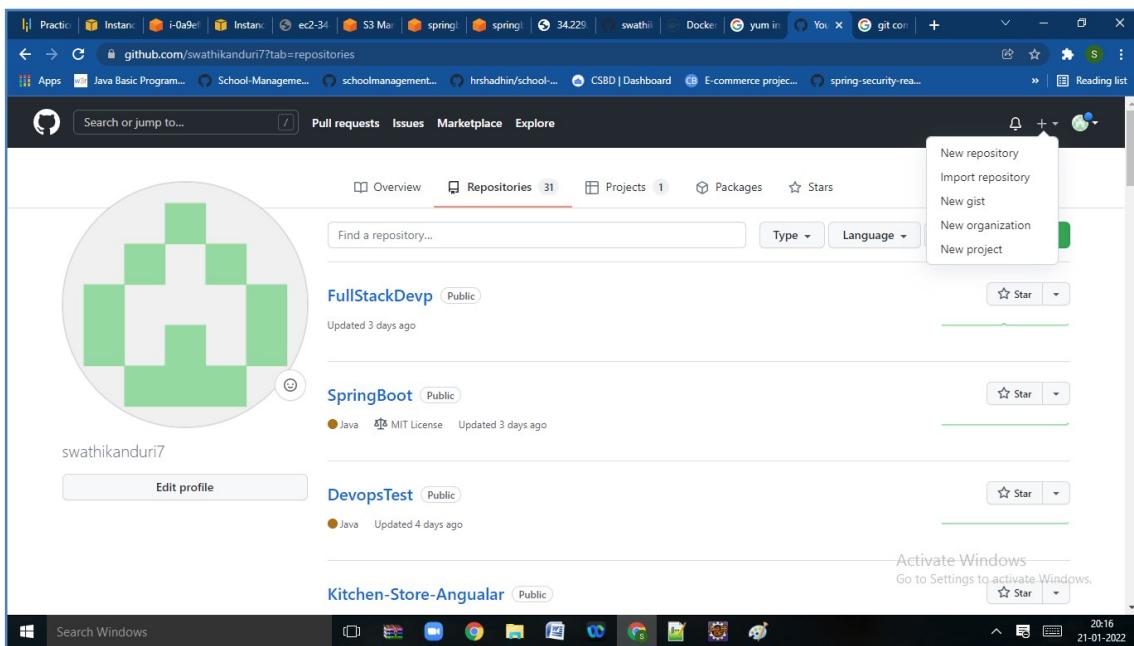
Login to github by providing the email address and password



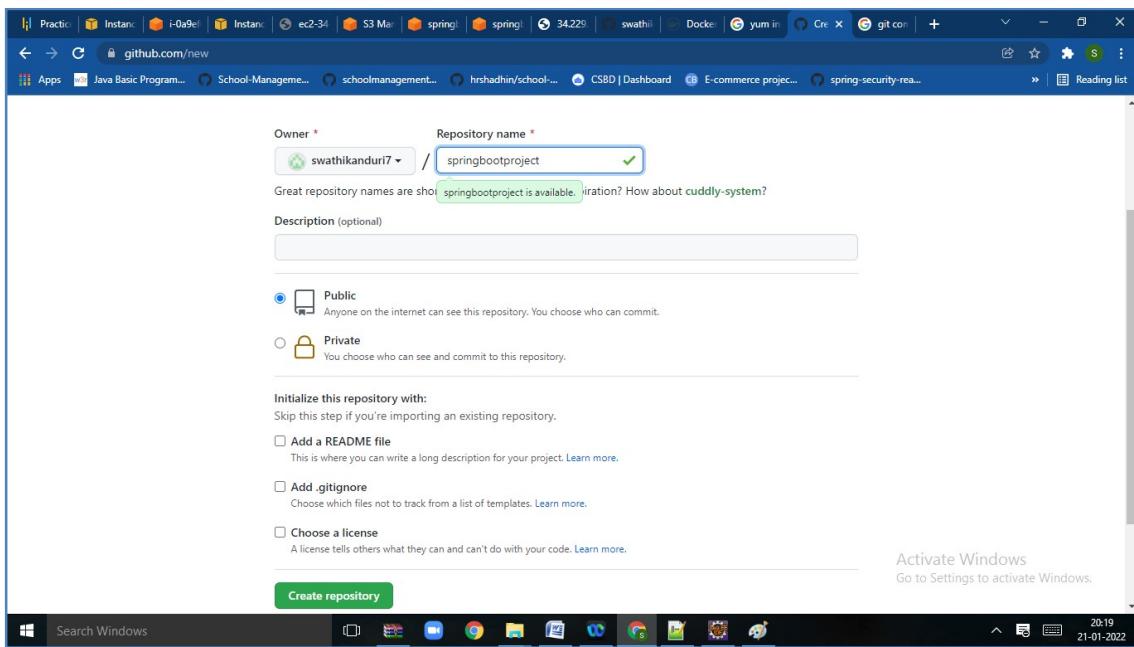
Click on sign in option to open the page



Click on new repository to create new repository



Give the repository name and select create repository



Pushing the code to Git Hub Repository

- Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

- Initialize repository using the following command

git init

- Add all the files to your git repository using the following command

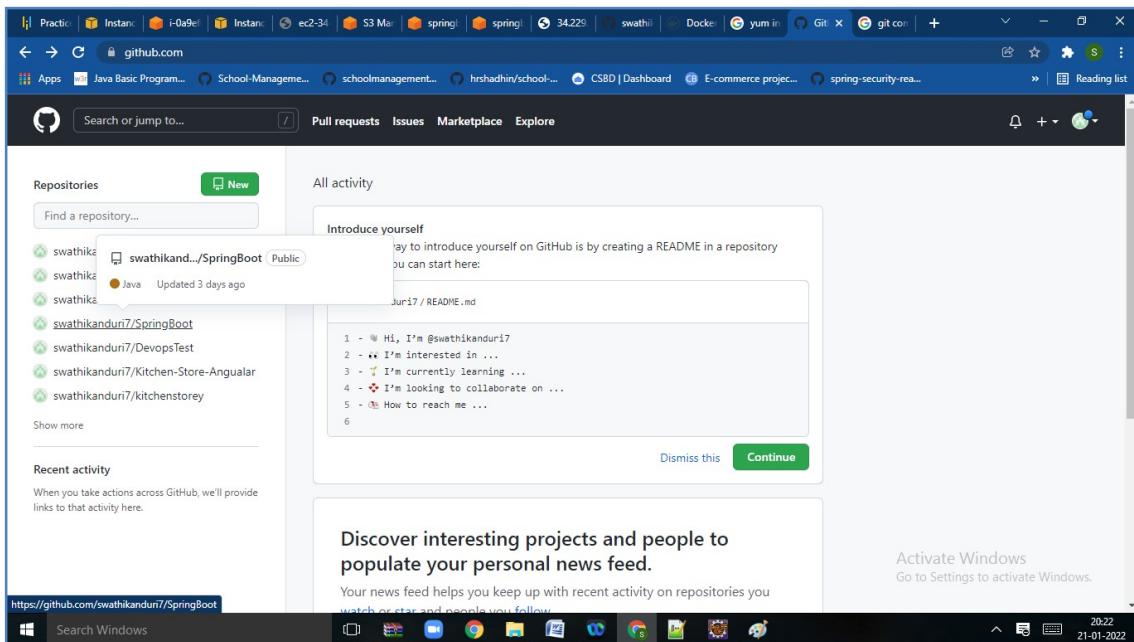
git add .

- Commit the changes using the following command

git commit . -m<commit message>

- Push the files to the folder you initially created using the following command

git push -u origin master

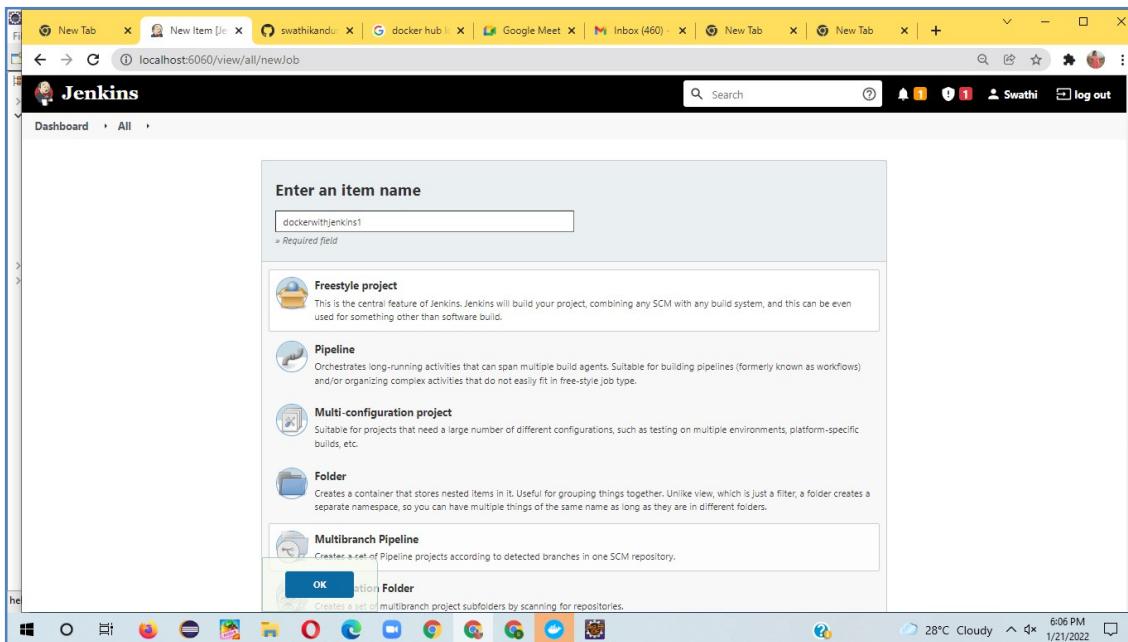


Code pushes to the git hub

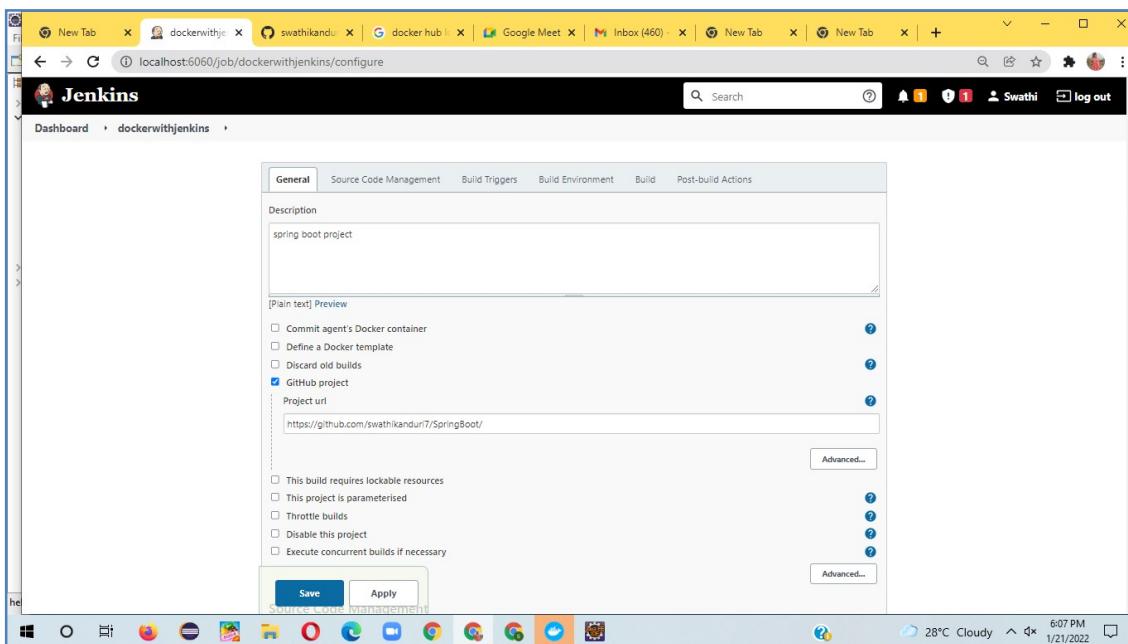
The screenshot shows a GitHub repository page for a user named 'swathikanduri7'. The repository is titled 'SpringBoot' and is public. The 'Code' tab is selected, displaying a single commit from 'sathvi21' with the message 'first commit'. The commit was made 3 days ago at 8526b12. The commit history also lists other files like '.settings', 'src', '.project', 'CODE-OF-CONDUCT.md', 'Dockerfile', 'Jenkinsfile', 'LICENSE', 'README.md', 'docker-compose.yml', and 'pom.xml', all with their respective commit times. To the right of the code area, there's an 'About' section with a note: 'No description, website, or topics provided.' It includes links for 'Readme', 'MIT License', 'Code of conduct', '0 stars', '1 watching', and '0 forks'. Below that is a 'Releases' section with a note: 'No releases published' and a link to 'Create a new release'. At the bottom, there's a 'Packages' section with a note: 'Activate Windows' and 'No packages published'. The browser taskbar at the bottom shows various open tabs and the date '21-01-2022'.

Jenkins Job

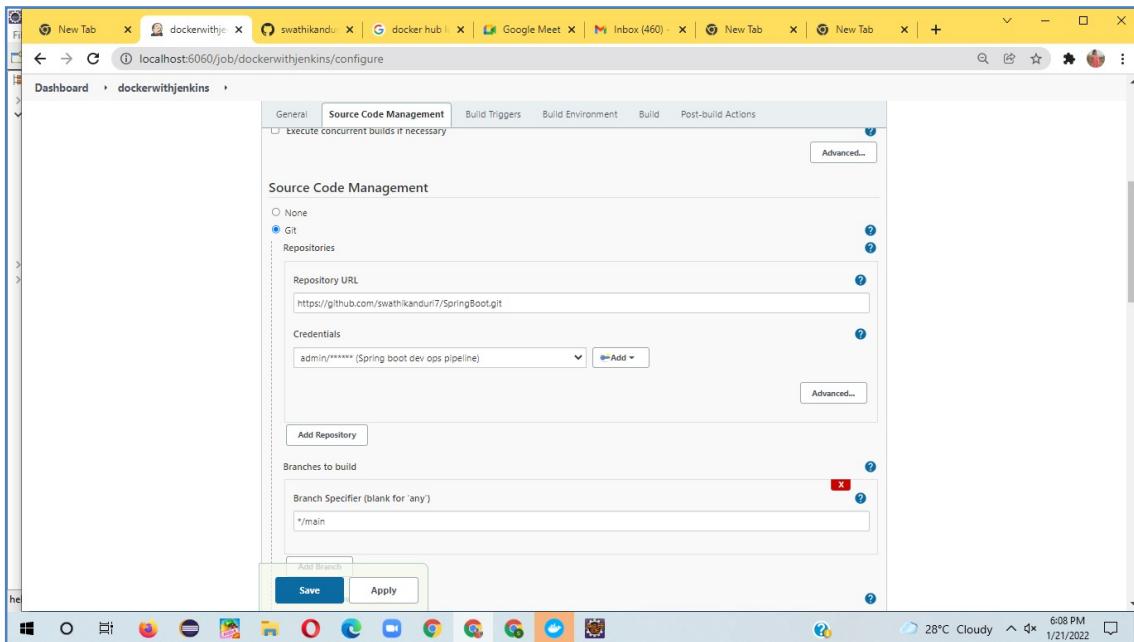
step1:- New item
give the name of the job
select free project



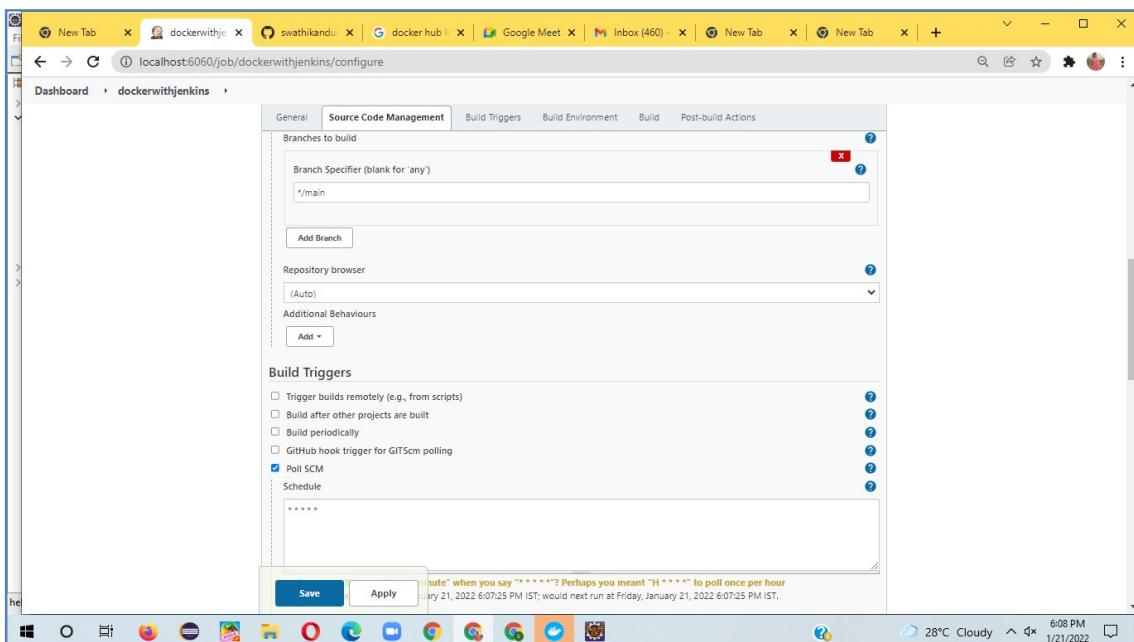
config git url:-



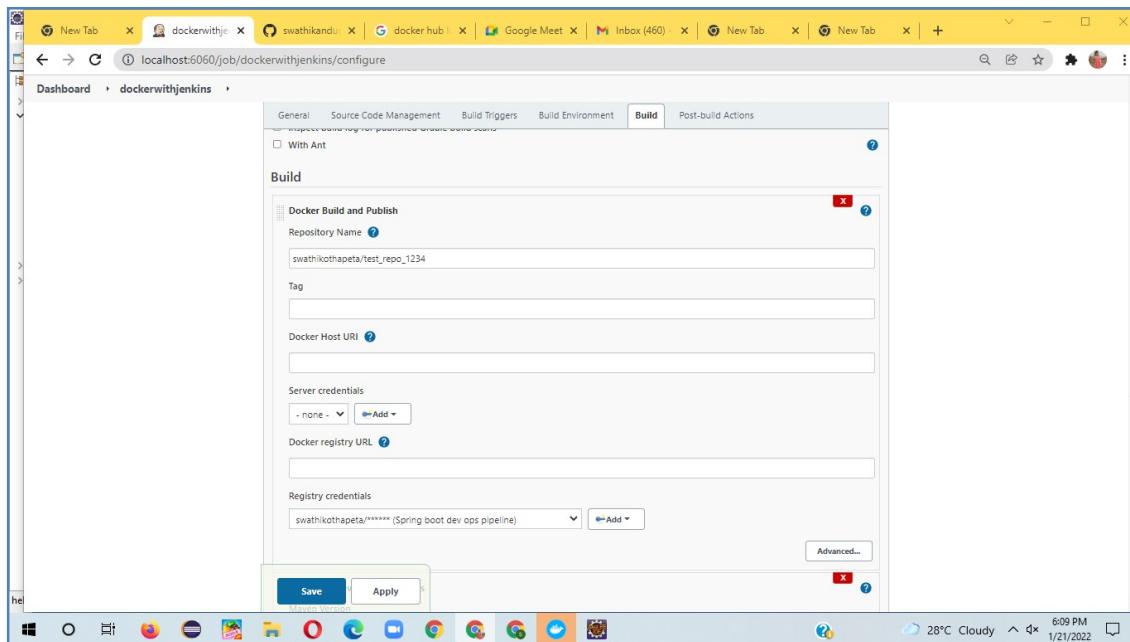
config scm



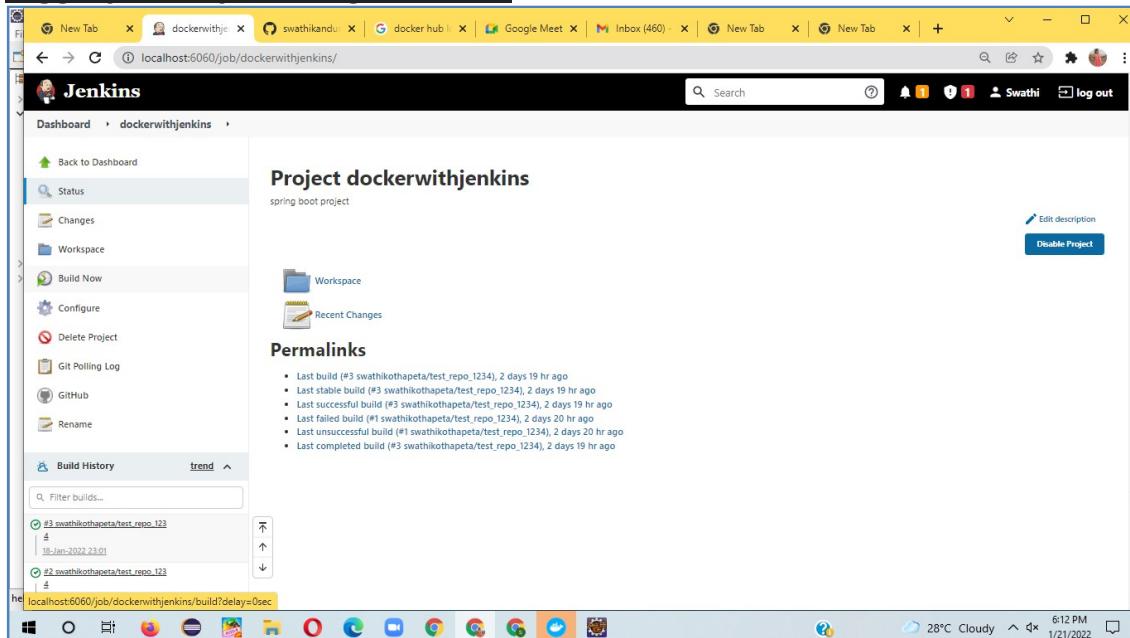
config build trigger rule, branch name



configure docker build and push and click on save to update the details



trigger jenkins job, using build now



to view the build status go to view history and open the console logs

Console Output

```

Started by user Swathi
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\workspace\dockerwithjenkins
The recommended git tool is: NONE
using credential devopspipeline
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\workspace\dockerwithjenkins\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/swathikothapeta7/SpringBoot.git # timeout=10
Fetching upstream changes from https://github.com/swathikothapeta7/SpringBoot.git
> C:\Program Files\Git\bin\git.exe -version # timeout=10
> git --version # git version 2.13.0.windows.1'
using GIT_ASKPASS to set credentials Spring boot dev ops pipeline
> C:\Program Files\Git\bin\git.exe fetch --tags --progress -- https://github.com/swathikothapeta7/SpringBoot.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/main{commit}" # timeout=10
Checking out Revision 852b012e957fbac8c58e6062e071778ad9efc3 (refs/remotes/origin/main)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f 852b012e957fbac8c58e6062e071778ad9efc3 # timeout=10
Commit message: "First commit"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk 852b012e957fc858e6062e071778ad9efc3 # timeout=10
[dockerwithjenkins] $ docker build -t swathikothapeta/test_repo_1234 --pull=true C:\ProgramData\Jenkins\workspace\dockerwithjenkins
WARNING: Support for the legacy .dockercfg configuration file and file-format is deprecated and will be removed in an upcoming release
WARNING: Support for the legacy .dockercfg configuration file and file-format is deprecated and will be removed in an upcoming release
WARNING: Support for the legacy .dockercfg configuration file and file-format is deprecated and will be removed in an upcoming release
#1 [internal] load build context from Dockerfile
#1 sha256:3231ca7fed1f874b76fd8e4358983c17bd3cc23f1b2bff71e474c980ce00e
#1 DONE 0.3s

#2 [internal] load .dockercfg
#2 sha256:292aa900cc1d1a424256bf2464aa89510177085c28f799048c569ce90e68acc
#2 transferring context:
#2 transferred: 2B 1.0s done

```

Docker image pushed to docker hub

```

[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\ProgramData\Jenkins\workspace\dockerwithjenkins\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ hello-world ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.dockerforjavadevelopers.hello.DummyTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Time elapsed: 0:141 s - in com.dockerforjavadevelopers.hello.DummyTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.1.2:jar (default-jar) @ hello-world ---
[INFO] building jar: C:\ProgramData\Jenkins\workspace\dockerwithjenkins\target\hello-world-0.1.0.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.2.1.RELEASE:repackage (repackage) @ hello-world ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ hello-world ---
[INFO] Installing C:\ProgramData\Jenkins\workspace\dockerwithjenkins\target\hello-world-0.1.0.jar to E:\Raju\Softwares\apache-maven-3.6.3\repo\com\dockerforjavadevelopers\hello-world\0.1.0\hello-world-0.1.0.jar
[INFO] Installing C:\ProgramData\Jenkins\workspace\dockerwithjenkins\pom.xml to E:\Raju\Softwares\apache-maven-3.6.3\repo\com\dockerforjavadevelopers\hello-world\0.1.0\hello-world-0.1.0.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:32 min
[INFO] Finished at: 2022-01-18T23:06:24+05:30
[INFO]
Finished: SUCCESS

```

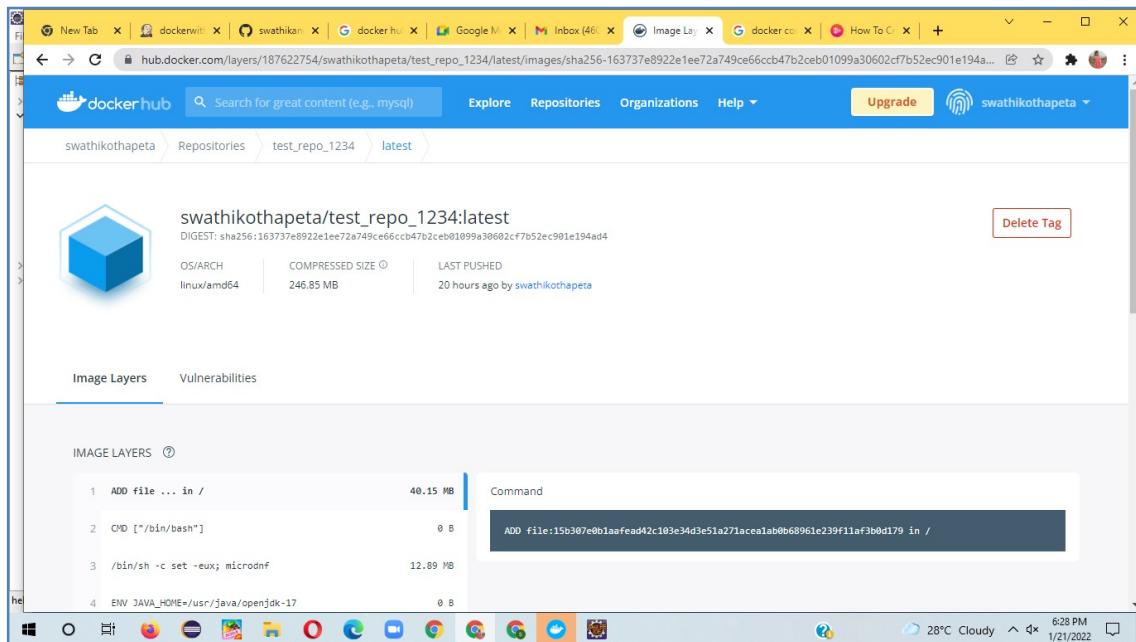
pull the image from the docker in EC2(docker pull
swathikothapeta/test_repo_1234:latest)

Docker Hub

to create the repository From the Docker Hub dashboard, click Create Repository. Fill out the repository details (be sure to set the Visibility drop-down to either public or private), then click Create

The screenshot shows the Docker Hub dashboard. In the top navigation bar, there are tabs for 'Explore', 'Repositories', 'Organizations', 'Help', and a 'Create Repository' button. A search bar at the top has 'swathikothapeta' selected. Below the search bar, a repository card for 'swathikothapeta / test_repo_1234' is displayed, showing it was updated 20 hours ago. It has 46 downloads and is set to 'Public'. A tip message suggests switching namespaces via the dropdown menu. The bottom of the screen shows a Windows taskbar with various icons.

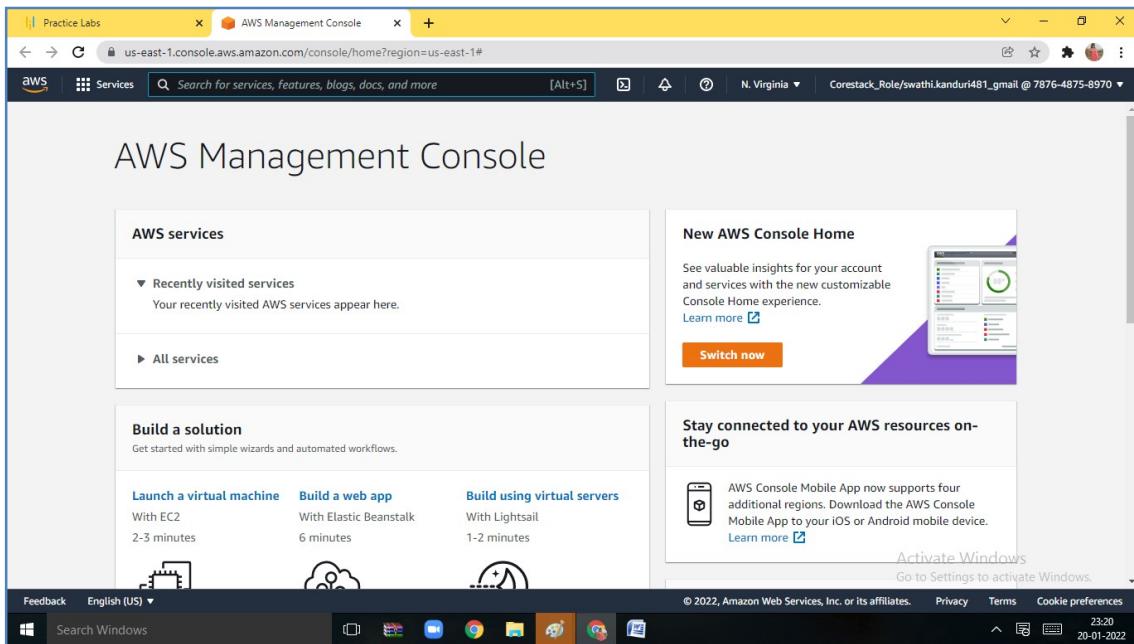
The screenshot shows the detailed view of the 'test_repo_1234' repository. At the top, there's an 'Advanced Image Management' section with a note about viewing preview. Below that, the repository name 'swathikothapeta / test_repo_1234' is shown, along with a 'testing' tag and a note about the last push being 20 hours ago. To the right, there are 'Docker commands' (including a 'docker push' command) and a 'Public View' button. The 'Tags and Scans' section lists two tags: 'latest' and 'springsample1', both pulled 2 hours ago and pushed 20 hours ago. The 'VULNERABILITY SCANNING - DISABLED' status is shown with an 'Enable' link. On the right side, there's an 'Automated Builds' section with a note about manually pushing images and connecting to GitHub or Bitbucket, and a 'Learn more' link. The bottom of the screen shows a Windows taskbar.



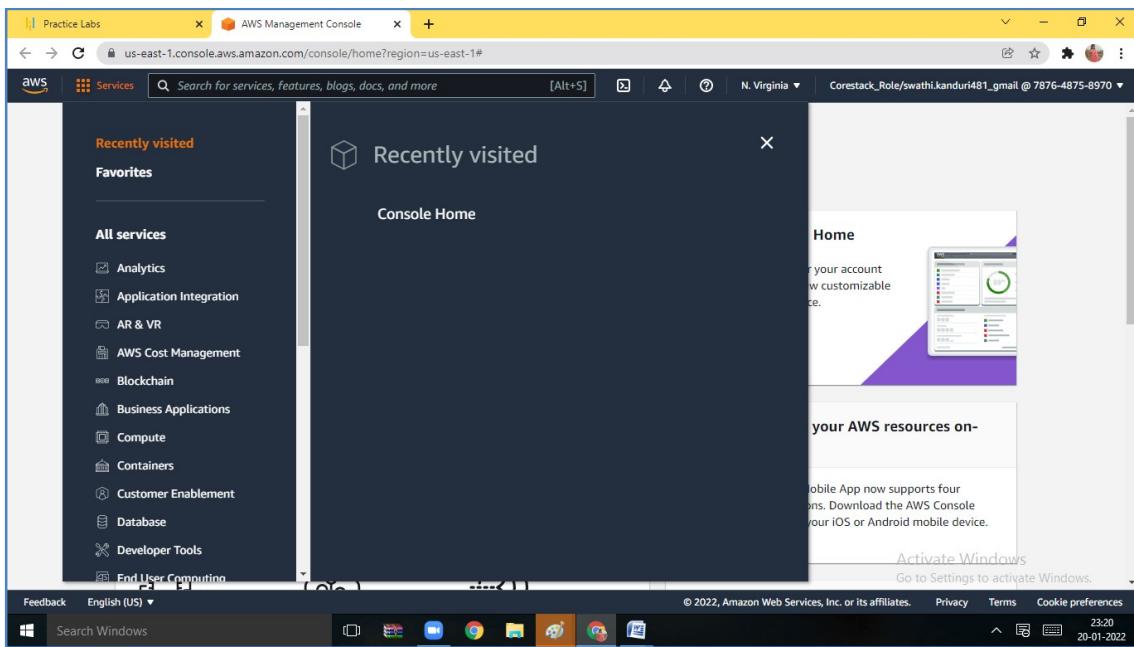
AWS page

A screenshot of a web browser showing an AWS Practice Labs interface. The URL is lms.simplilearn.com/courses/3865/Testing-in-a-DevOps-Lifecycle/practice-labs. The page title is 'Testing in a DevOps Lifecycle'. It shows '1 Class completed | 86% Self-Learning Videos Watched | 0/2 Projects Done'. On the left sidebar, there are icons for 'SELF LEARNING', 'LIVE CLASSES', 'PRACTICE LABS', 'ASSESSMENT', and 'CERTIFICATE'. The main content area is titled 'Current Lab : AWS Certification - Dedicated Account'. It has tabs for 'Access Information', 'Lab Details', 'Components', 'Log Details', and 'Usage Details'. Under 'Access Information', there are links for 'AWS Web Console' and 'AWS API Access', both with 'Open in New Tab' buttons. An 'Auth Url' field contains the value 'https://signin.aws.amazon.c'. Below it, a timer says 'Session Expires in: 7h 33m 48s' with a 'Refresh Link' button. To the right, there is a box for 'AWS Certification - Dedicated Account' with details: Category: Cloud Computing, Start Date: 2022-01-19 17:02, End Date: 2022-01-21 07:17, and Code: SLAWS. A descriptive text block about AWS follows. At the bottom, there is an 'Activate Windows' message: 'Go to Settings to activate Windows.' The browser status bar shows the URL https://signin.aws.amazon.com/federation?Action=login&Issuer=class.corestack.io&Destination=https%3A%2F%2Fus-east-1.console.aws.amazon.com%2Fconsole%2Fhome%3Fregion%3Dus-east-1%23&SignInToken=JDBGlctyU7REu215rEtytRAH7G... and the date 20-01-2022.

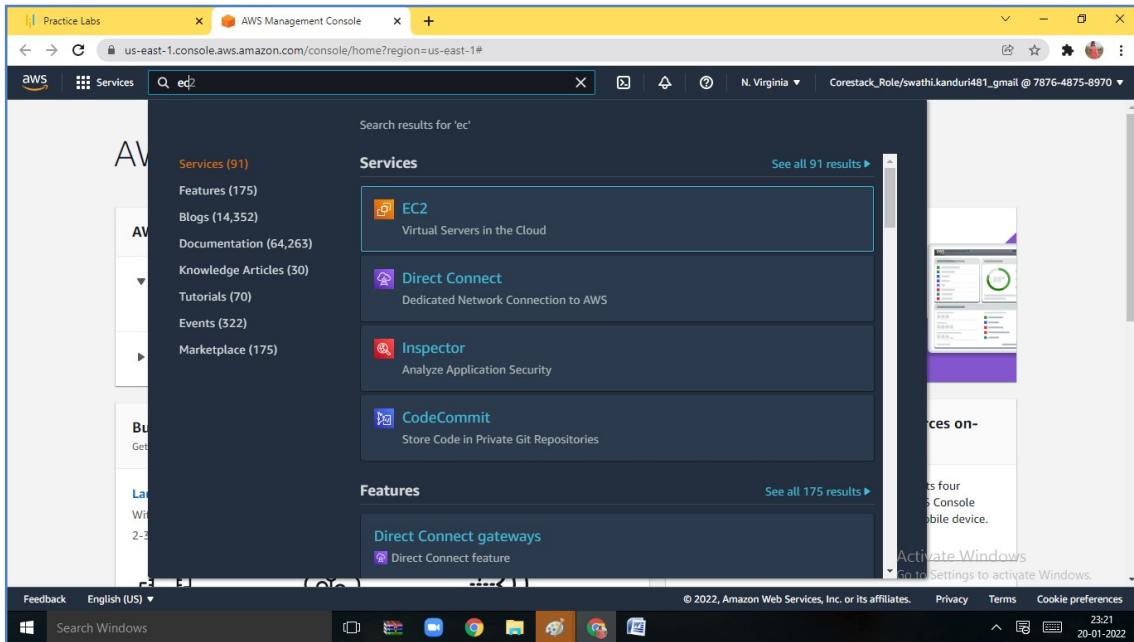
AWS Home Page



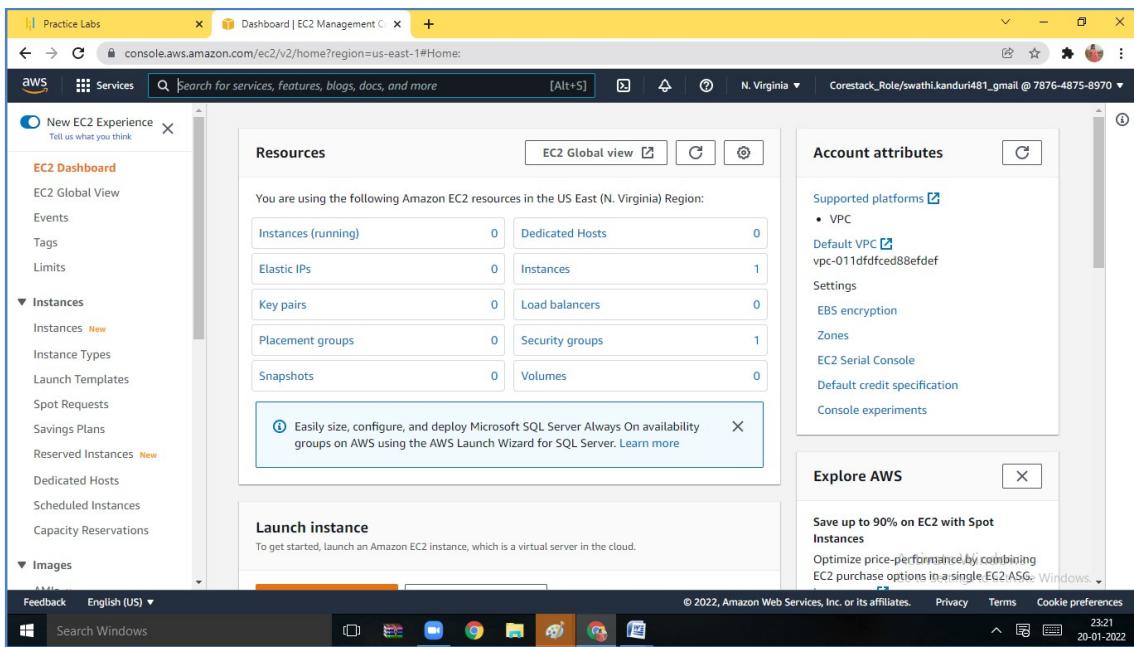
Click on Services to select EC2



Click on EC2



Click on instances launch an instance



The screenshot shows the AWS EC2 Management Console in the Instances section. A single instance, `i-0a0d1c4380179a15b`, is listed as `Terminated`. The instance type is `t2.micro`, and it has no alarms. It is located in the `us-east-1b` availability zone. The status check is marked with a question mark icon.

**Click on launch instance and select the Amazon Linux 2(HVM)-kernel 4.14
4.14, SSD Volume Type - ami-083602cee93914c0c (64-bit x86) / ami-**

The screenshot shows the AWS Launch Instance Wizard at Step 1: Choose an Amazon Machine Image (AMI). The user is invited to try an early beta iteration of the wizard. The `Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-083602cee93914c0c (64-bit x86)` is selected. The `64-bit (x86)` option is chosen for the virtualization type. The `Select` button is highlighted in blue.

Select the choose an instance type

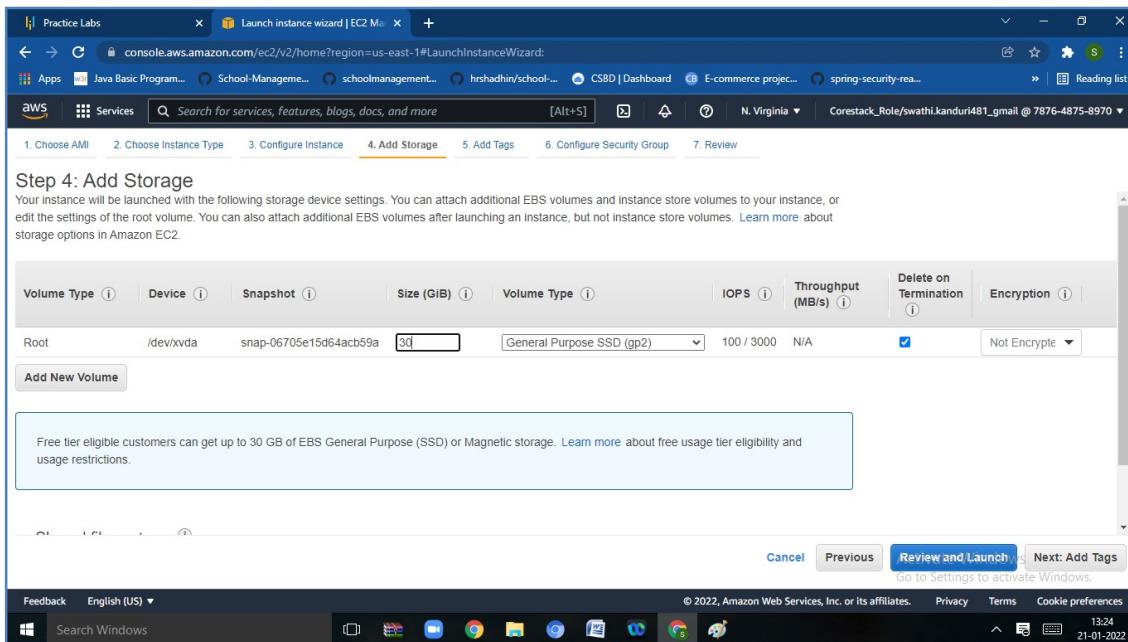
The screenshot shows the AWS Launch instance wizard at Step 2: Choose an Instance Type. The user has selected the t2.micro instance, which is highlighted with a blue border and labeled "Free tier eligible". The table lists various t2 instances with their details: Family, Type, vCPUs, Memory (GiB), Instance Storage (GB), EBS-Optimized Available, Network Performance, and IPv6 Support. The t2.micro row is the first one in the list.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes

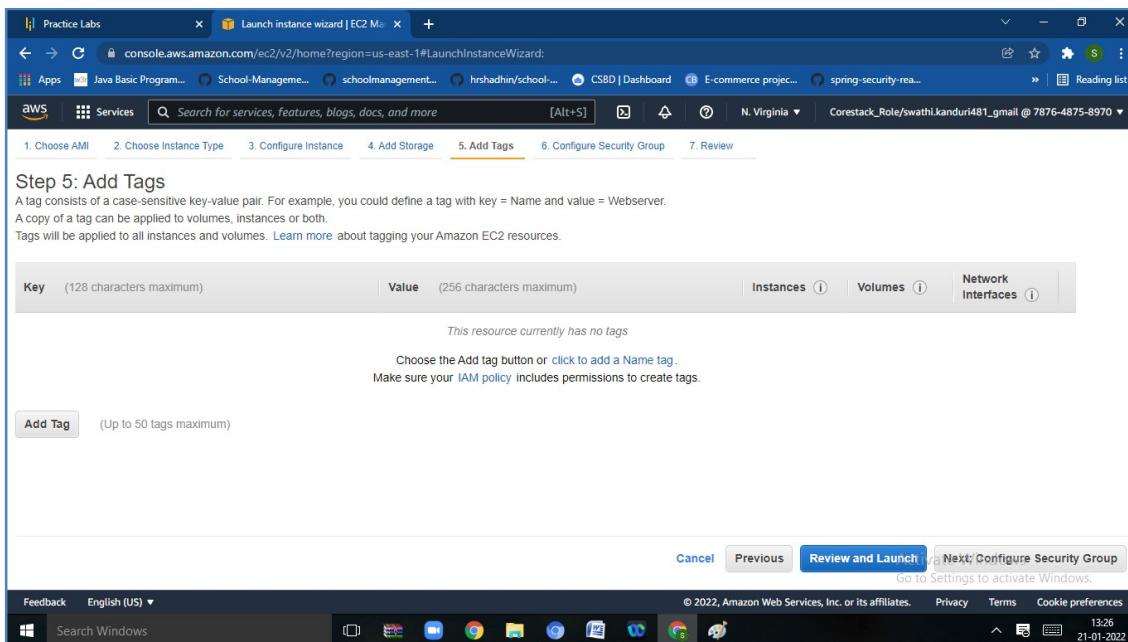
Click on next: configure instance details

The screenshot shows the AWS Launch instance wizard at Step 3: Configure Instance Details. The "Request Spot Instances" checkbox is currently unchecked. Other configuration options include the number of instances (set to 1), network settings (using default VPC and subnet), and hostname types (using subnet setting). The "Review and Launch" button is visible at the bottom right.

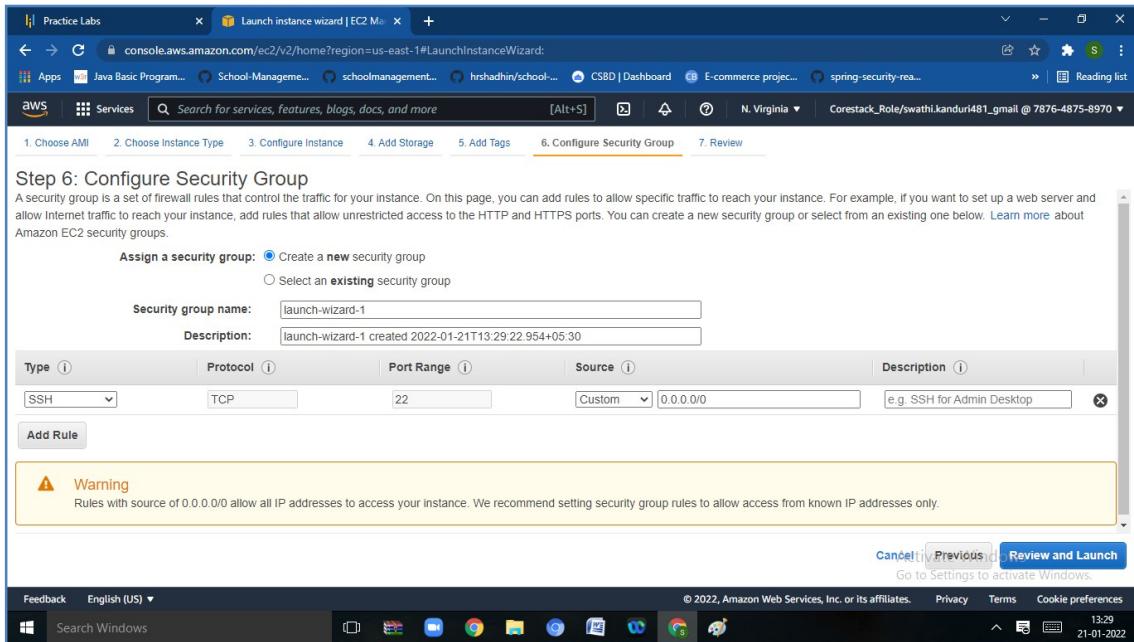
Click on Next: add storage and change the value to 30(GB)



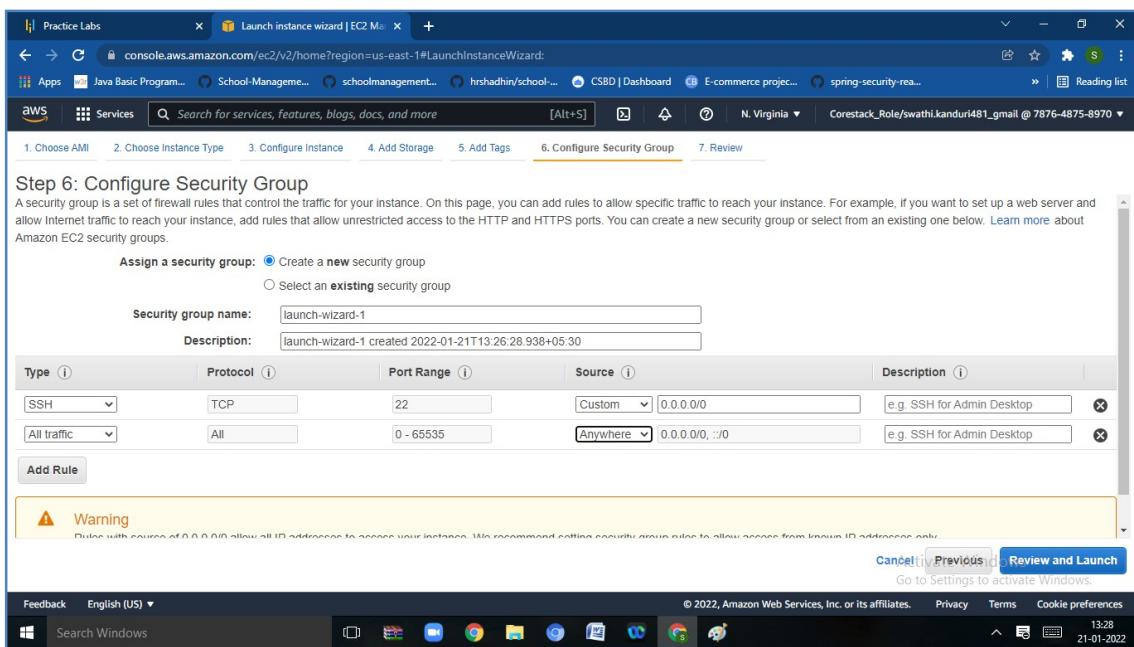
Click on next: Add Tags



Click on Next:Configure Security Group



Click on Next:Configure Security Group and add the Rule as type is AllTraffic and source is Anywhere



Click on Review and launch

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-083602cee93914c0c

Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is...

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

Feedback English (US) **Cancel** **Previous** **Launch** Go to Settings to activate Windows.

Step 7: Review Instance Launch

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

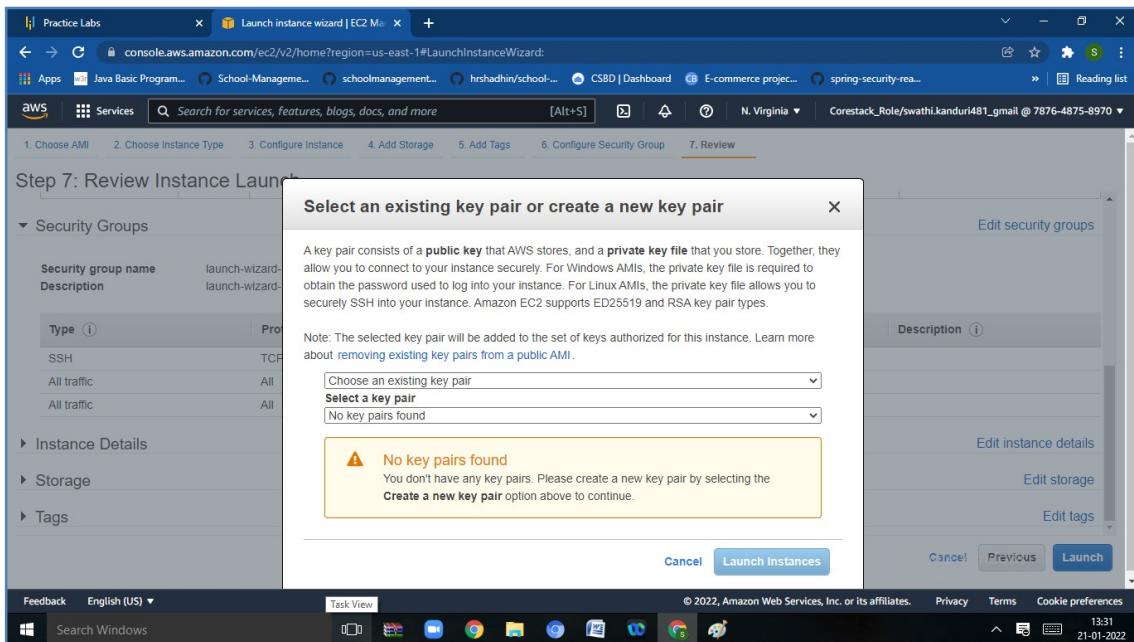
Security Groups

Security group name	Description
launch-wizard-1	launch-wizard-1 created 2022-01-21T13:29:22.954+05:30

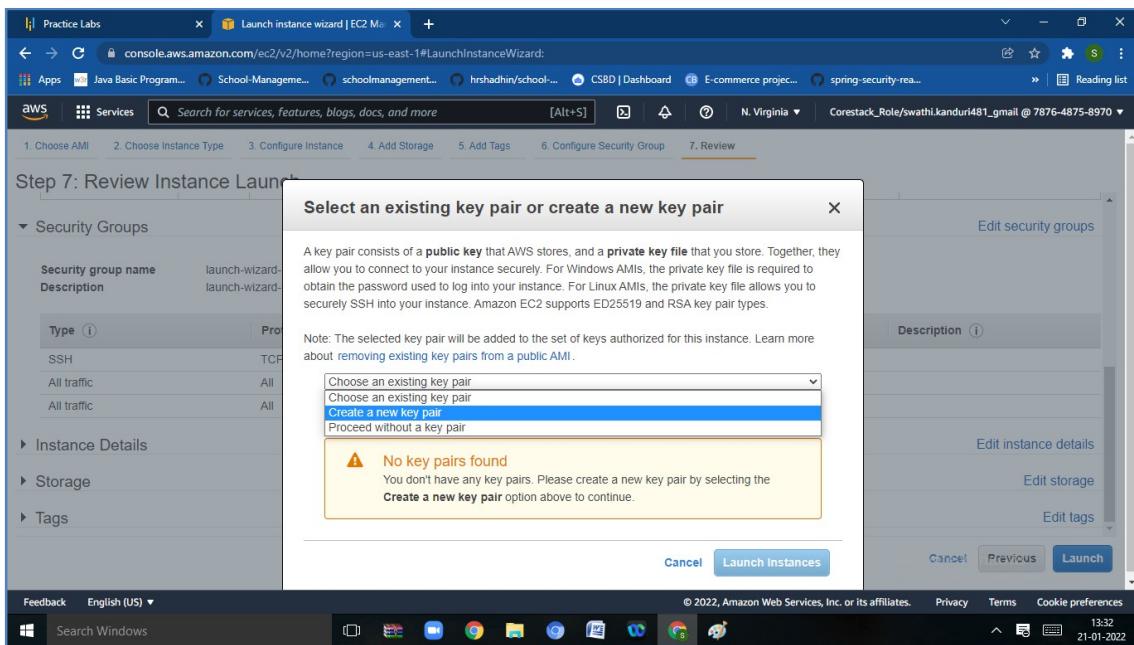
Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	
All traffic	All	All	0.0.0.0/0	
All traffic	All	All	::/0	

Feedback English (US) **Cancel** **Previous** **Launch** Go to Settings to activate Windows.

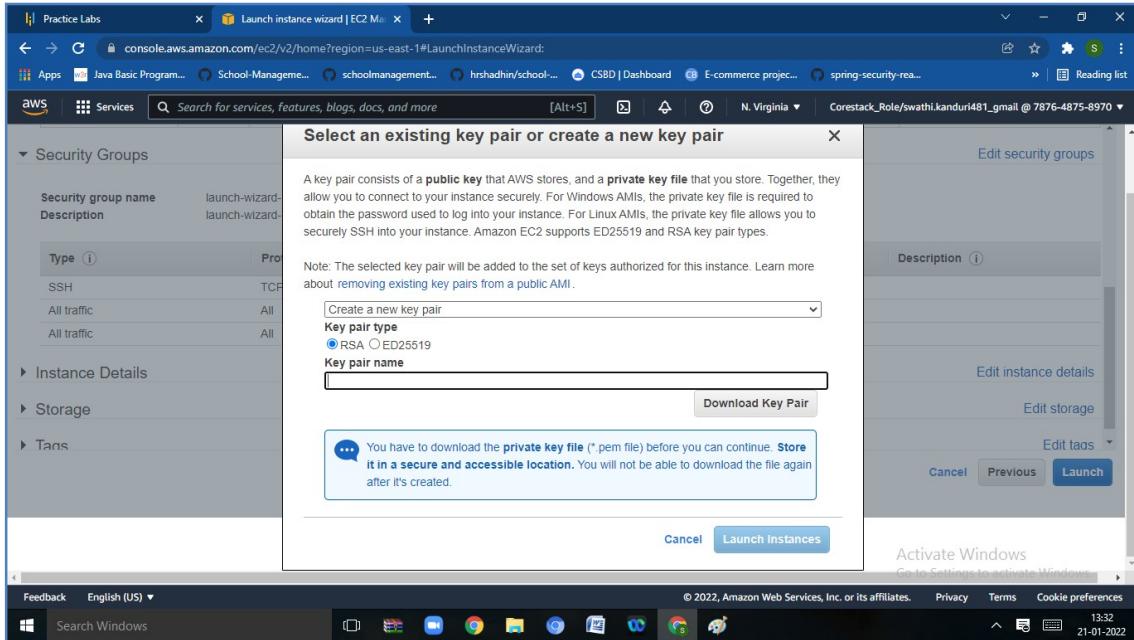
Click on launch



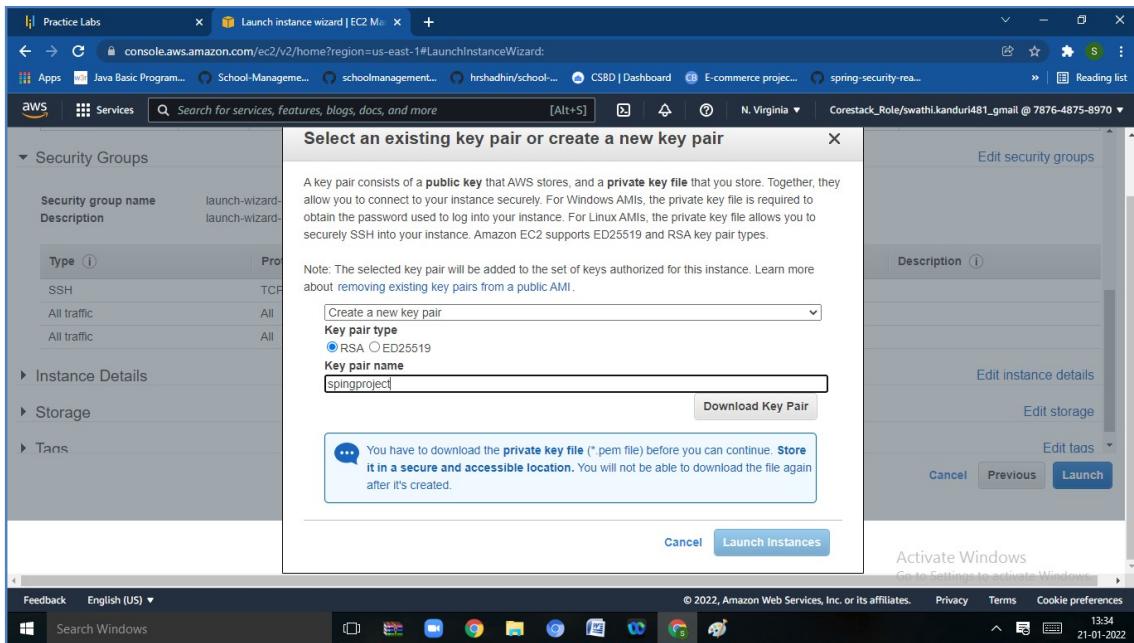
Click on launch and select existing key pair



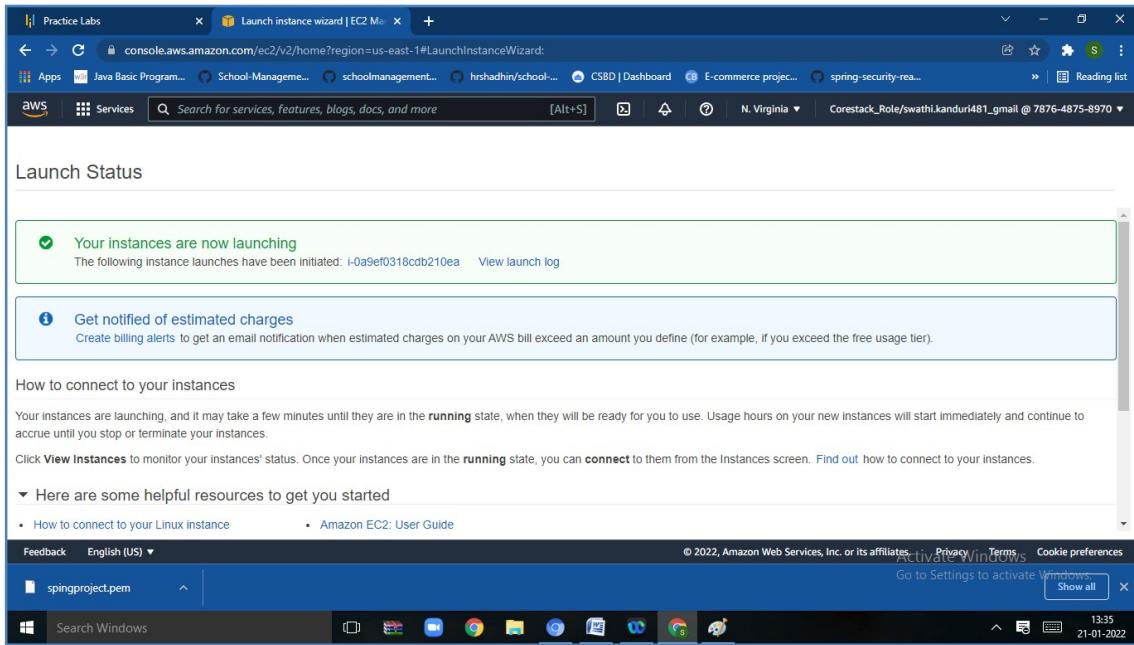
Select the key pair type as RSA



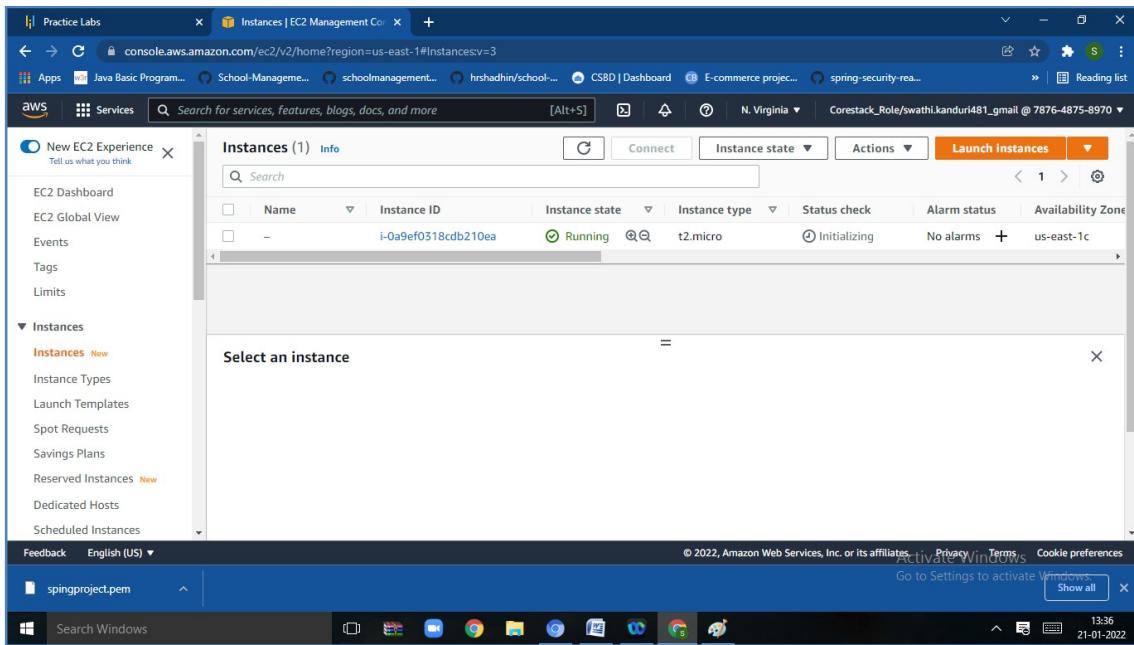
Give the name as spring project and download it



Click on Launch Instances



Click on view instance



Click on check box and select the instance and click on connect

The screenshot shows the AWS EC2 Management Console. The left sidebar has a 'Instances' section with 'Instances New' selected. The main area displays a table of instances with one row selected. The 'Actions' dropdown menu is open, and the 'Connect' option is highlighted.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
-	i-0a9ef0318cdb210ea	Running	t2.micro	Initializing	No alarms	us-east-1c

Savings Plans	Hostname type IP name: ip-172-31-23-34.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-23-34.ec2.internal	Answer private resource DNS name IPv4 (A)
Reserved Instances New	Instance type	Elastic IP addresses	VPC ID

Instances New	Instance: i-0a9ef0318cdb210ea	Platform Amazon Linux (Inferred)	AMI ID ami-083602cee93914c0c	Monitoring disabled
Instance Types	Platform details Linux/UNIX	AMI name amzn2-ami-hvm-2.0.20211223.0-x86_64-gp2	Termination protection Disabled	
Launch Templates	Launch time Fri Jan 21 2022 13:35:27 GMT+0530 (India)	AMI location amazon/amzn2-ami-hvm-2.0.20211223.0-	Lifecycle normal	

Instances New	Instance: i-0a9ef0318cdb210ea	Owner 787648758970	ClassicLink -	Enclaves Support -
Instance Types	Boot mode -	Allow tags in instance metadata Enabled	Use RBN as guest OS hostname Disabled	

Instance types	Enabled	Affinity	Placement group
Launch Templates	Host and placement group	Tenancy	Partition number

Launch Templates	Host resource group name -	Tenancy default	Partition number -
Spot Requests	Virtualization type hvm	Reservation r-0bc2ce59496fc99b	Number of vCPUs 1

Spot Requests	Capacity reservation Info
Savings Plans	Capacity Reservation ID
Reserved Instances New	Capacity Reservation setting
Dedicated Hosts	open
Scheduled Instances	

After clicking on connect -> connect to instance

Connect to instance [Info](#)
Connect to your instance i-0a9ef0318cdb210ea using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 Serial Console

Instance ID: i-0a9ef0318cdb210ea
Public IP address: 34.229.162.133
User name: ec2-user

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

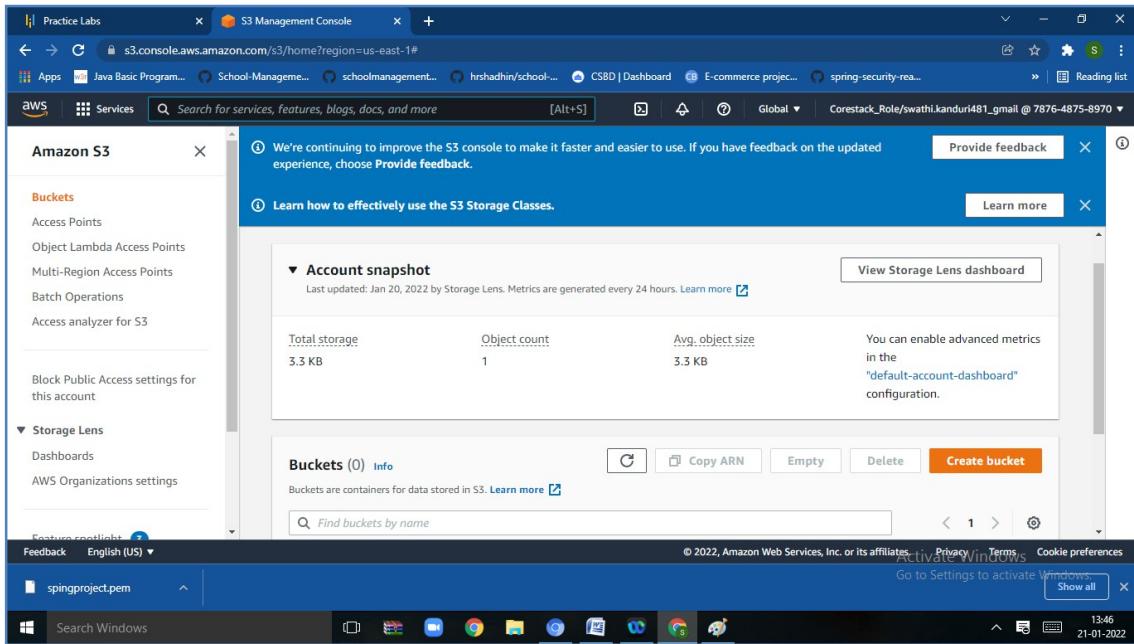
Select S3 bucket from services

Search results for 's3'

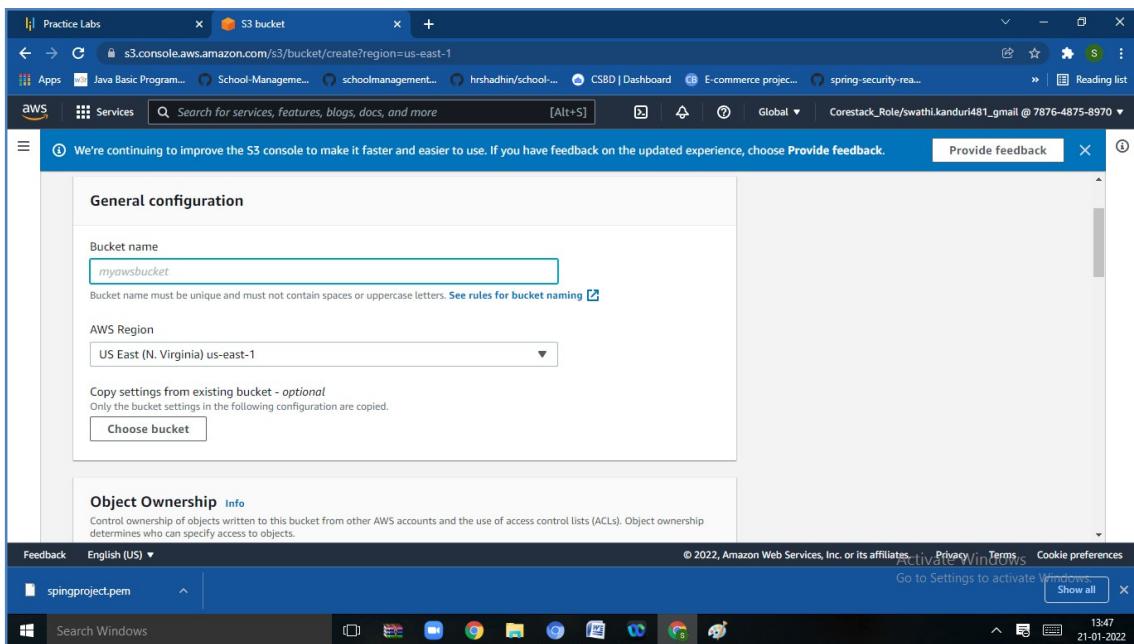
Services (7)

- S3** Scalable Storage in the Cloud
- S3 Glacier** Archive Storage in the Cloud
- Athena** Query Data in S3 using SQL
- AWS Snow Family** Large Scale Data Transport

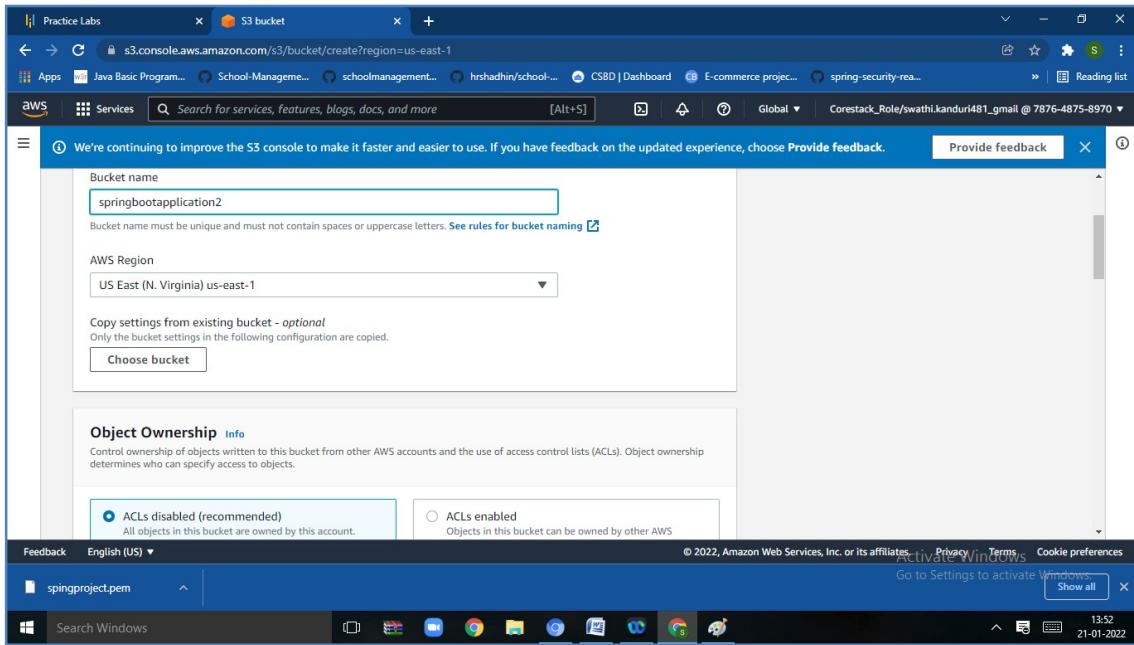
See all 7 results ▾



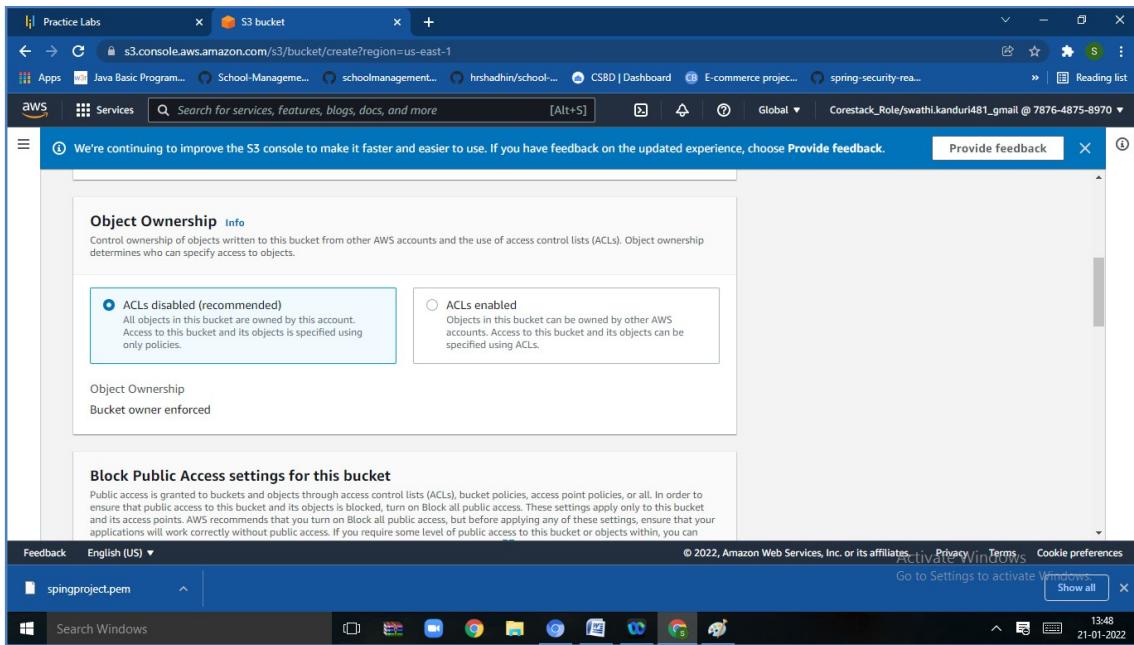
Click on create bucket

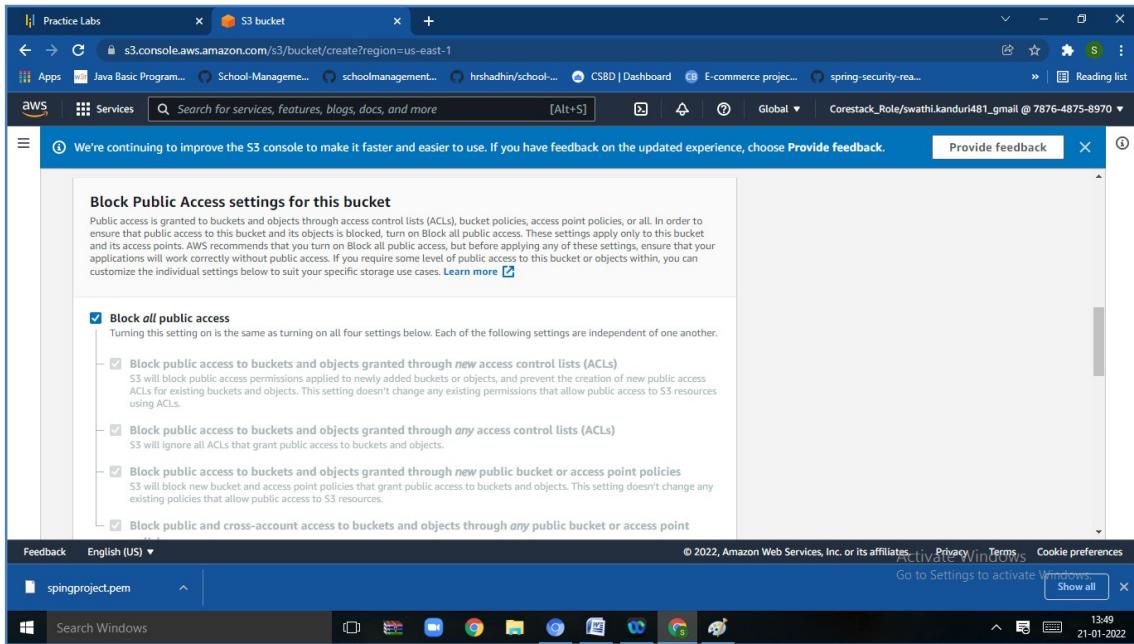


Name given for bucket

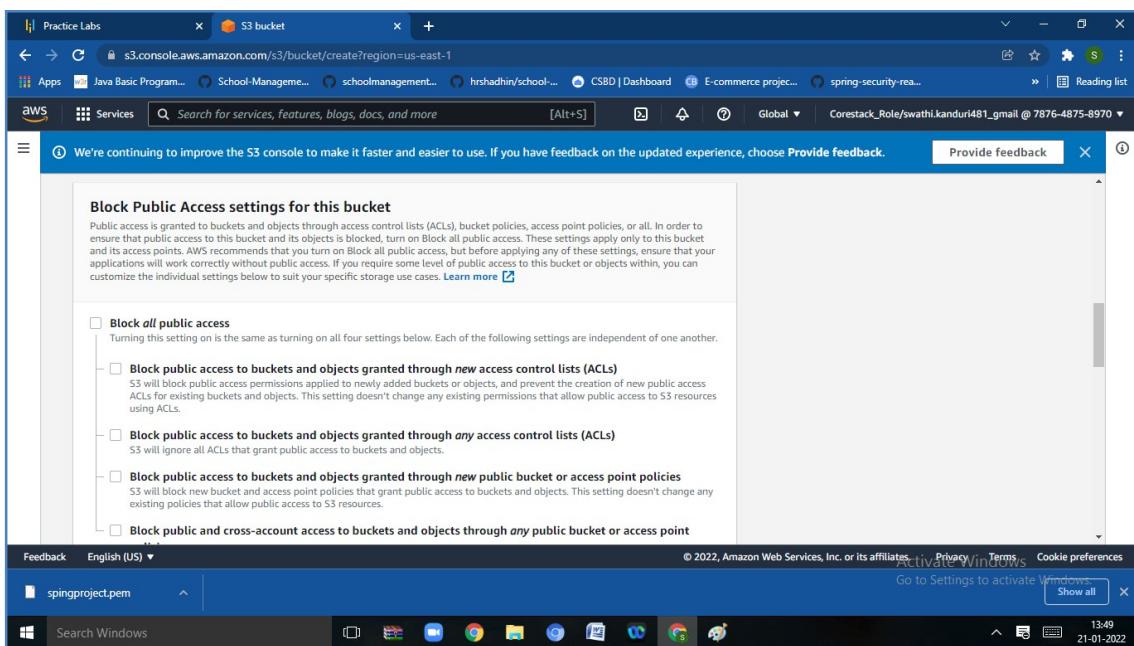


Select ACLs disabled in Object Ownership

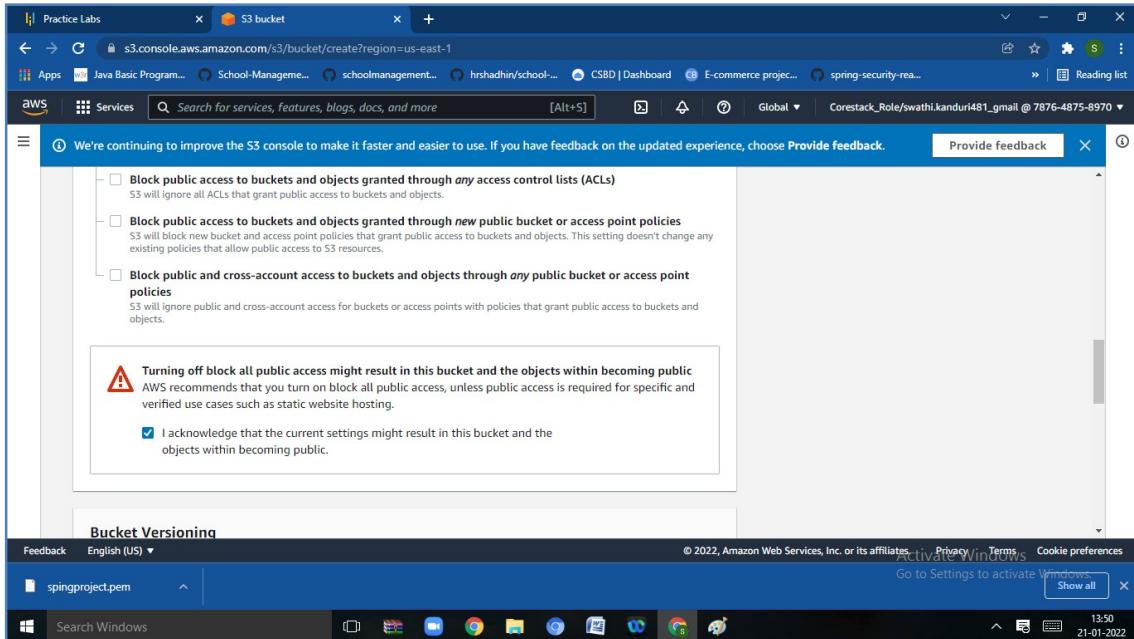




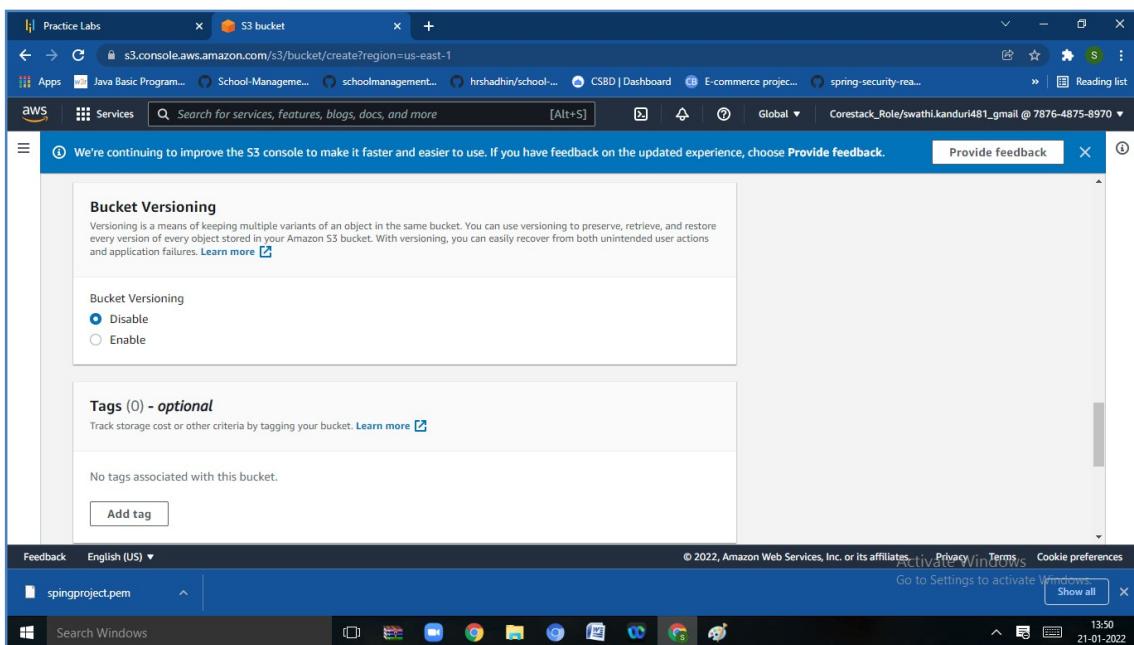
Deselect on block all public access



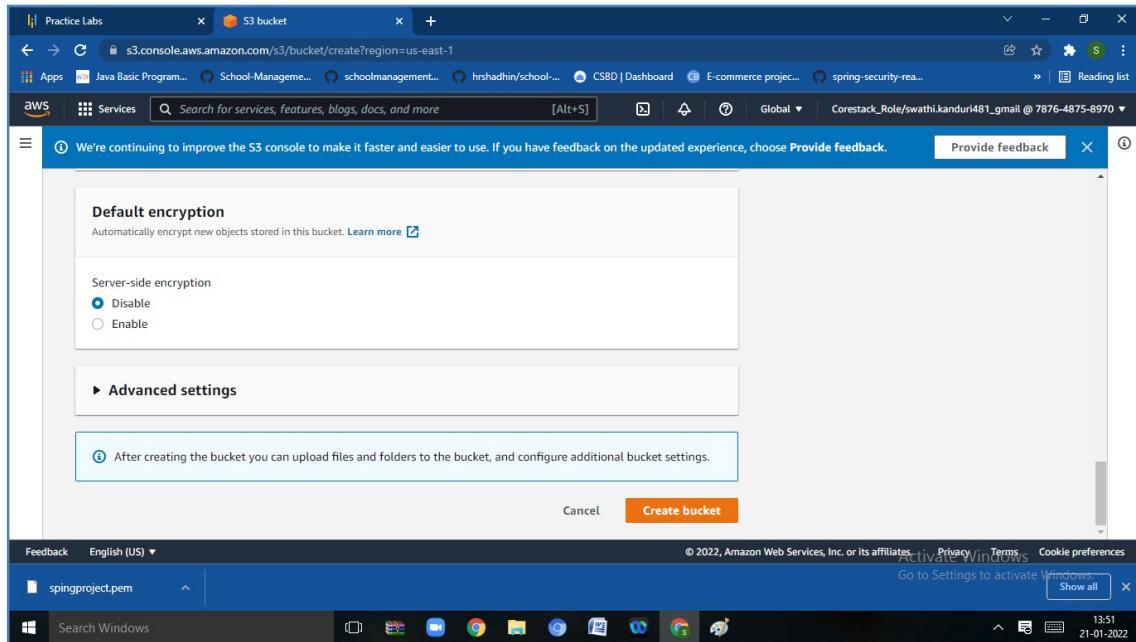
Click on I acknowledge that the current settings might result in this bucket and the objects within becoming public.



Bucket versioning

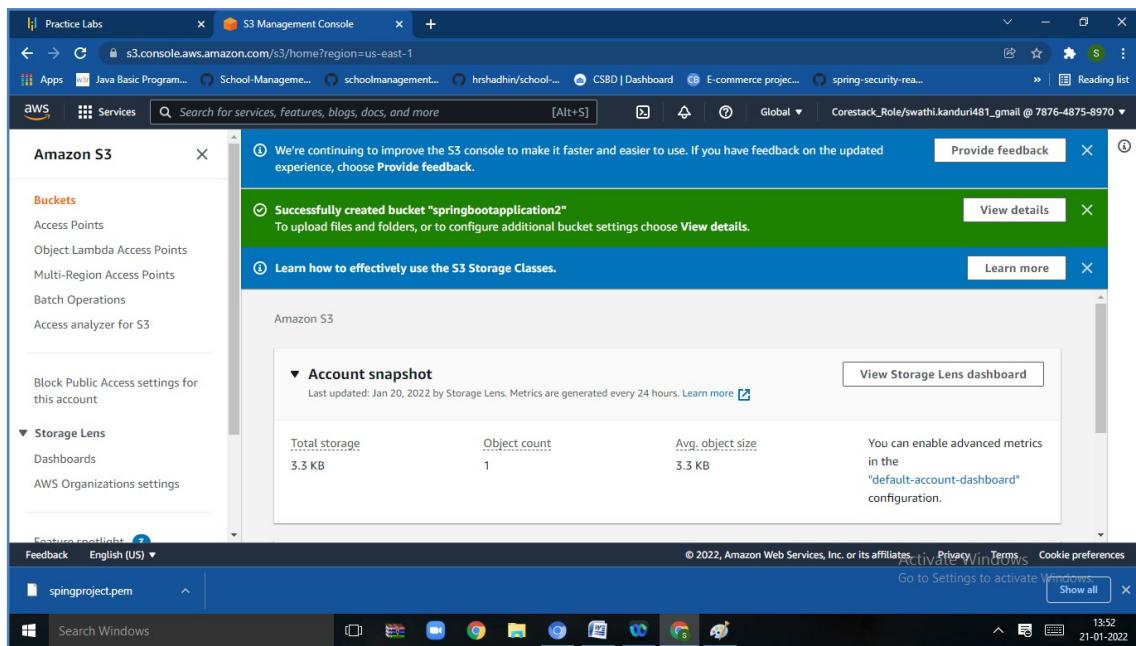


Default encryption



After clicking on create bucket it display the message as

Successfully created bucket "springbootapplication2"
To upload files and folders, or to configure additional bucket settings choose **View details**



Click on the type

The screenshot shows the AWS S3 console interface. The left sidebar is titled 'Amazon S3' and includes sections for 'Buckets', 'Access Points', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', and 'Access analyzer for S3'. Below these are 'Block Public Access settings for this account' and a 'Storage Lens' section with 'Dashboards' and 'AWS Organizations settings'. A 'Feedback' link is at the bottom of the sidebar. The main content area shows the 'springbootapplication2' bucket. At the top, there's a message: 'We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose Provide feedback.' Below this is the breadcrumb navigation 'Amazon S3 > springbootapplication2'. The main panel is titled 'springbootapplication2' with an 'Info' link. It has tabs for 'Objects' (selected), 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Objects' tab displays a message: 'Objects (0)' and 'Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' Below this are buttons for 'Create folder' (disabled) and 'Upload' (highlighted in orange). There's also a search bar with the placeholder 'Find objects by prefix'.

The screenshot shows the AWS S3 console interface, similar to the previous one but with a different tab selected. The left sidebar is identical. The main content area shows the 'springbootapplication2' bucket. The breadcrumb navigation now includes 'Upload'. The main panel is titled 'Upload' with an 'Info' link. It has instructions: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more' and a link to 'Drag and drop files and folders you want to upload here, or choose Add files, or Add folders'. Below this is a table titled 'Files and folders (0)' with a note: 'All files and folders in this table will be uploaded.' It includes a search bar 'Find by name' and columns for Name, Folder, Type, and Size. At the bottom, there are 'Remove', 'Add files', and 'Add folder' buttons. The footer includes standard AWS links like 'Privacy', 'Terms', and 'Cookie preferences', and a Windows activation notice.

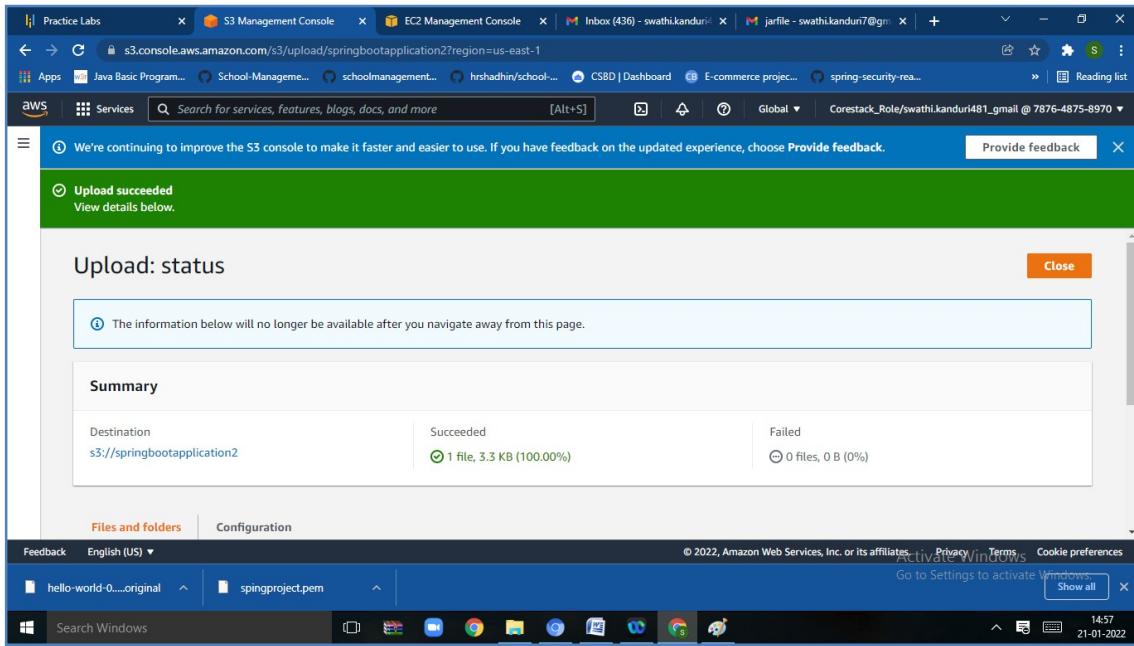
Upload the file by using add files and click on upload

The screenshot shows the AWS S3 Management Console 'Upload' interface. A single file, 'hello-world-0.1.0.jar.original', is listed in the 'Files and folders' section. The file is 3.3 KB in size. Below the file list is a search bar labeled 'Find by name'. At the bottom of the page, there are buttons for 'Feedback', 'English (US)', 'Privacy', 'Terms', and 'Cookie preferences', along with a link to 'Activate Windows'.

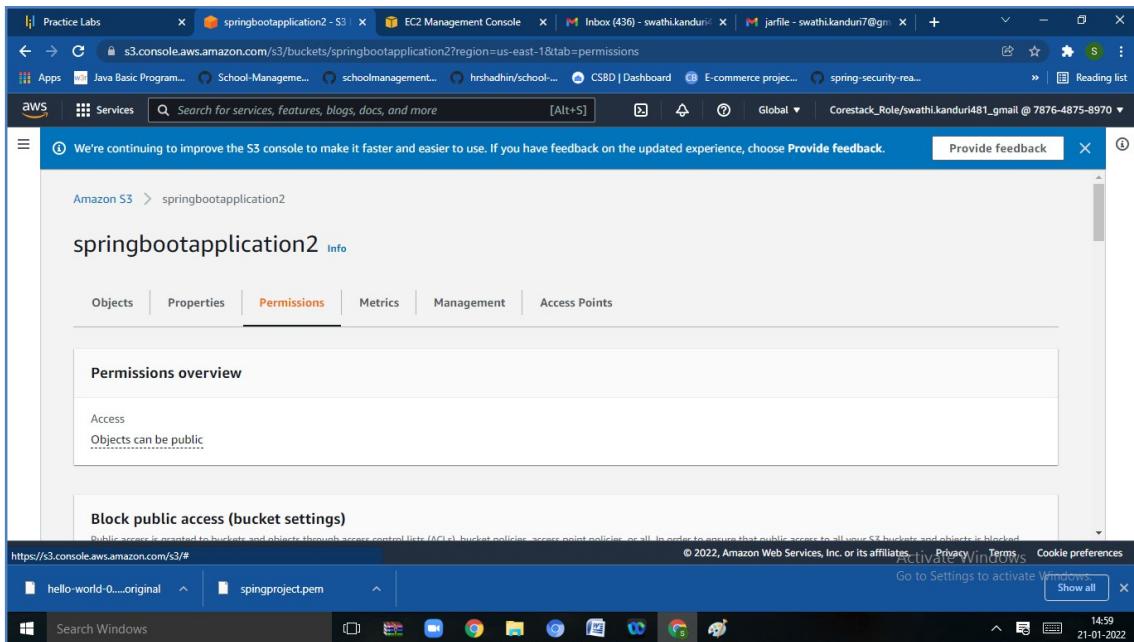
The screenshot shows the 'Destination' configuration step of the upload process. The destination is set to 's3://springbootapplication2'. Below this, there are sections for 'Destination details' (Bucket settings that impact new objects stored in the specified destination), 'Permissions' (Grant public access and access to other AWS accounts), and 'Properties' (Specify storage class, encryption settings, tags, and more). At the bottom right, there are 'Cancel' and 'Upload' buttons. The status bar at the bottom indicates '14:56 21-01-2022'.

Upload is success and it shows the message as

Upload succeeded
View details below.



Click on permissions to set the permissions



Click on object ownership and click on edit

The screenshot shows the AWS S3 console with the 'Object Ownership' tab selected. A note at the top states: 'This bucket has the bucket owner enforced setting applied for Object Ownership. When bucket owner enforced is applied, use bucket policies to control access.' Below this, there is an 'Edit' button.

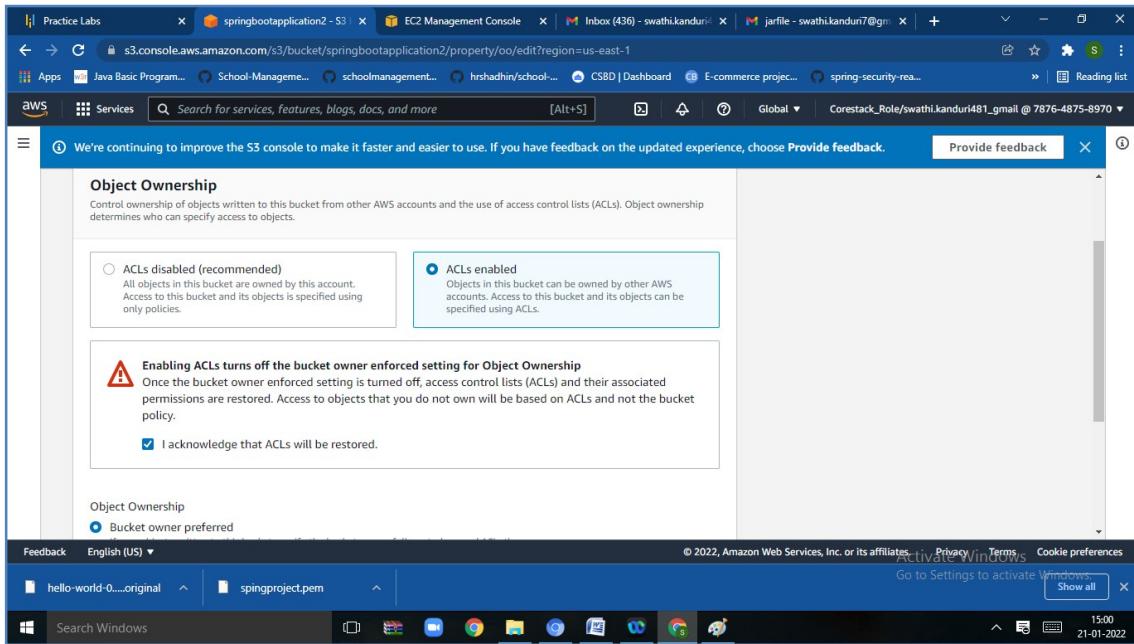
The screenshot shows the 'Edit Object Ownership' page. It lists two options: 'ACLs disabled (recommended)' and 'ACLs enabled'. The 'ACLs disabled' option is selected and highlighted with a blue border. A note below it says: 'All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.' The 'ACLs enabled' option is described as: 'Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.' Below this, there is an 'Edit' button.

Click on ACLs enabled

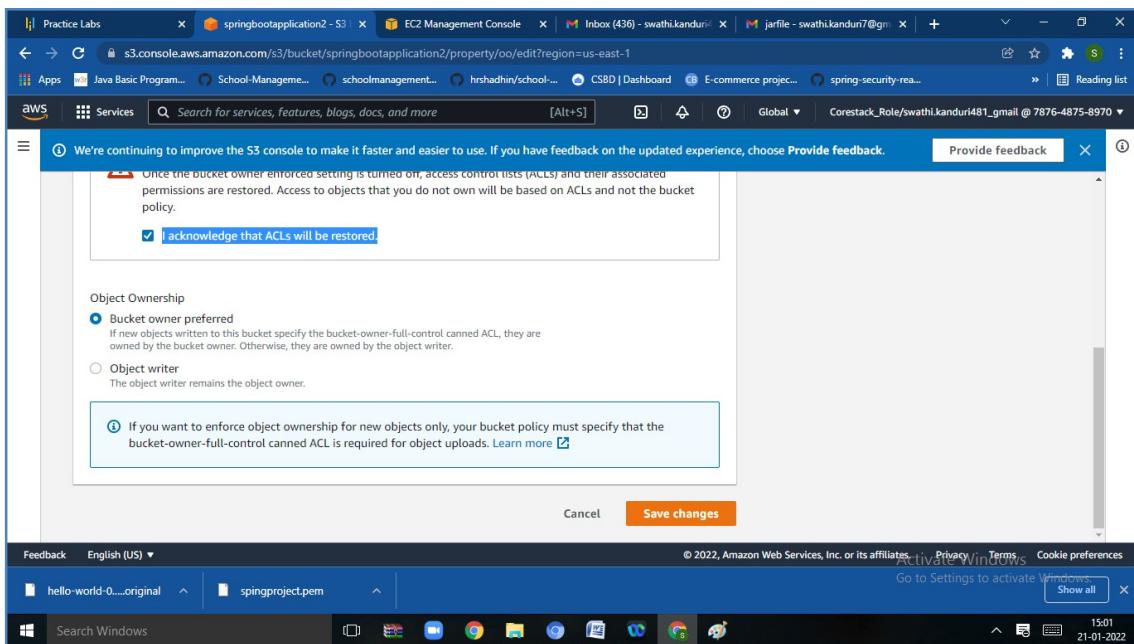
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

And

Click on I acknowledge that ACLs will be restored.



Click on save changes



Successfully edited object ownership as shown in below picture

The screenshot shows the AWS S3 console for the bucket 'springbootapplication2'. A green success message at the top states 'Successfully edited Object Ownership.' Below it, the 'Permissions' tab is selected under the 'Info' section. Under 'Access', it says 'Objects can be public'. The bottom status bar shows the date and time as 21-01-2022 15:02.

Object overview

The screenshot shows the AWS S3 console for the object 'hello-world-0....original' in the 'springbootapplication2' bucket. The 'Object overview' section displays various metadata fields: Owner (claasklabs+5f0353c12d1de5048e47094), AWS Region (US East (N. Virginia) us-east-1), Last modified (January 21, 2022, 14:57:39 (UTC+05:30)), Size (3.3 KB), Type (original), Key (empty), S3 URI (s3://springbootapplication2/hello-world-0.1.0.jar.original), Amazon Resource Name (ARN) (arn:aws:s3:::springbootapplication2/hello-world-0.1.0.jar.original), Entity tag (Etag) (c9a1ce57715b4a075f3ad14ce16ff77a), and Object URL (<https://springbootapplication2.s3.amazonaws.com/hello-world-0.1.0.jar.original>). The bottom status bar shows the date and time as 21-01-2022 15:09.

Click on Access Control List(ACL)

The screenshot shows the AWS S3 console with the URL <https://s3.console.aws.amazon.com/s3/buckets/springbootapplication2?region=us-east-1&tab=permissions>. A green success message at the top says "Successfully edited Object Ownership." Below it, the "Access control list (ACL)" section shows grants for "Bucket owner (your AWS account)" and "Everyone (public access)". The "Edit" button is visible in the top right corner.

Click on Access control list(ACL) and select edit option to make changes for public access

The screenshot shows the AWS S3 console with the URL <https://s3.console.aws.amazon.com/s3/bucket/springbootapplication2/property/ac/edit?region=us-east-1>. The "Edit access control list (ACL)" page is displayed, showing grants for "Bucket owner (your AWS account)" and "Everyone (public access)". The "Edit" button is visible in the top right corner.

Select list and Read for public access

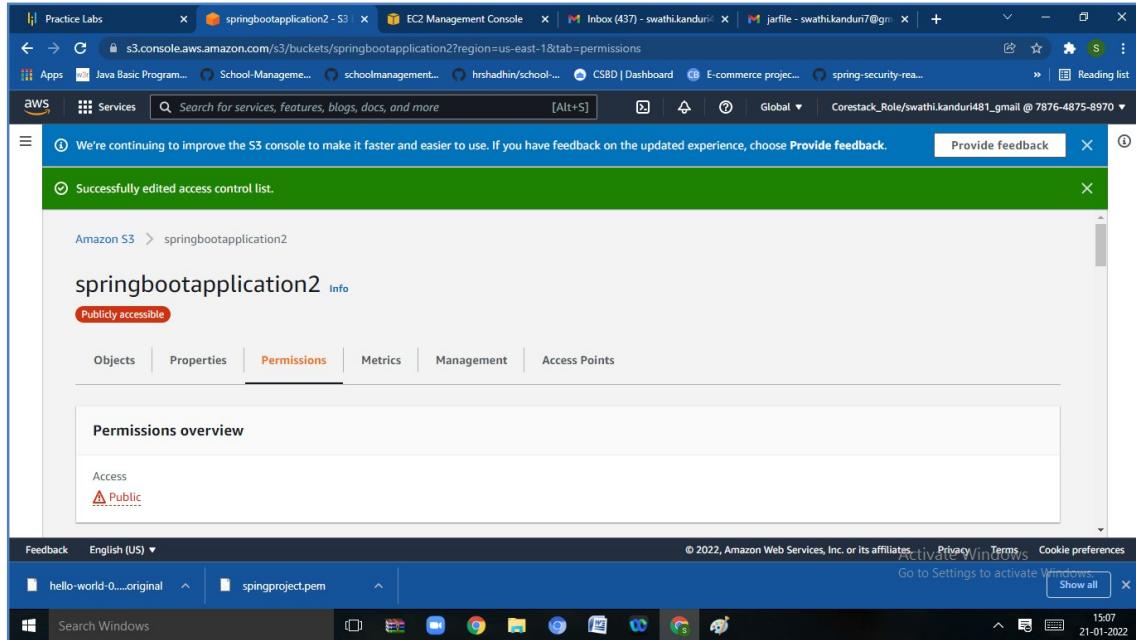
The screenshot shows the 'Edit access control list (ACL)' page for a bucket named 'springbootapplication2'. Under the 'Bucket ACL' section, the 'Bucket owner (your AWS account)' has 'List', 'Read', and 'Write' checked. The 'Everyone (public access)' group has 'List' and 'Read' checked, while 'Write' is unchecked. A note at the bottom states: 'When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access the objects in this bucket.' A checkbox for 'I understand the effects of these changes on my objects and buckets.' is checked.

Click on I understand the effects of these changes on my objects and buckets.and save changes

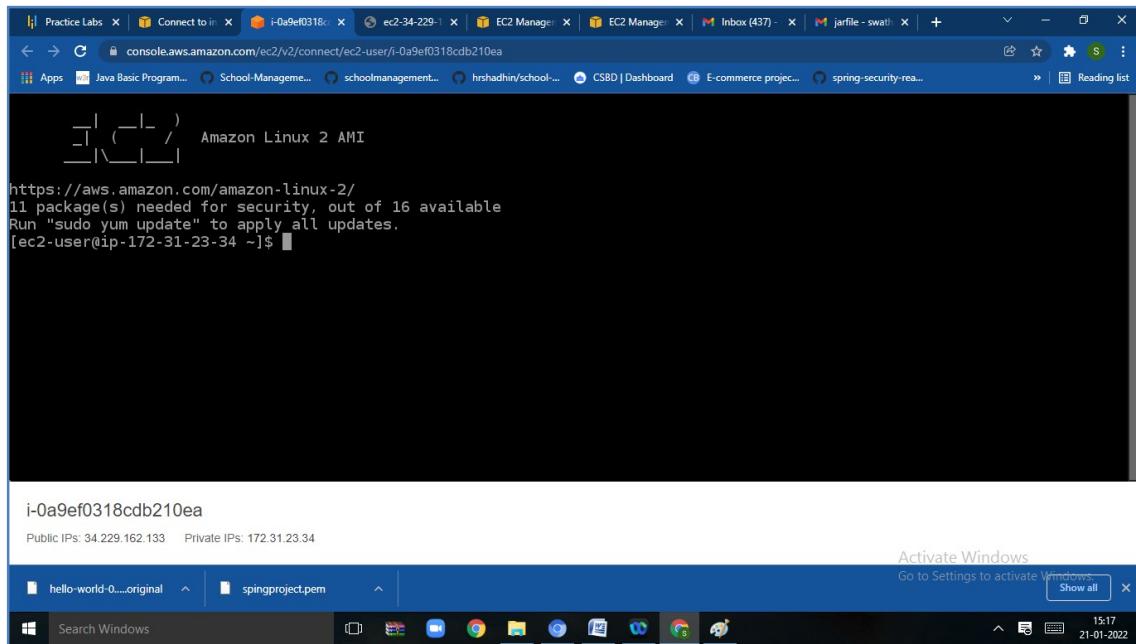
The screenshot shows the same 'Edit access control list (ACL)' page after saving changes. The 'Bucket ACL' section remains the same. The 'Everyone (public access)' group still has 'List' and 'Read' checked. The note about granting access to the Everyone group is present. The 'I understand the effects of these changes on my objects and buckets.' checkbox is checked. At the bottom right, there are 'Cancel' and 'Save changes' buttons, with 'Save changes' being the active button.

After clicking on save changes it display the message as

Successfully edited access control list.



Console after connecting



Type sudo -i

The screenshot shows a terminal window titled 'Amazon Linux 2 AMI' with the URL <https://aws.amazon.com/amazon-linux-2/>. The terminal displays the following text:

```
11 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-23-34 ~]$ sudo -i
```

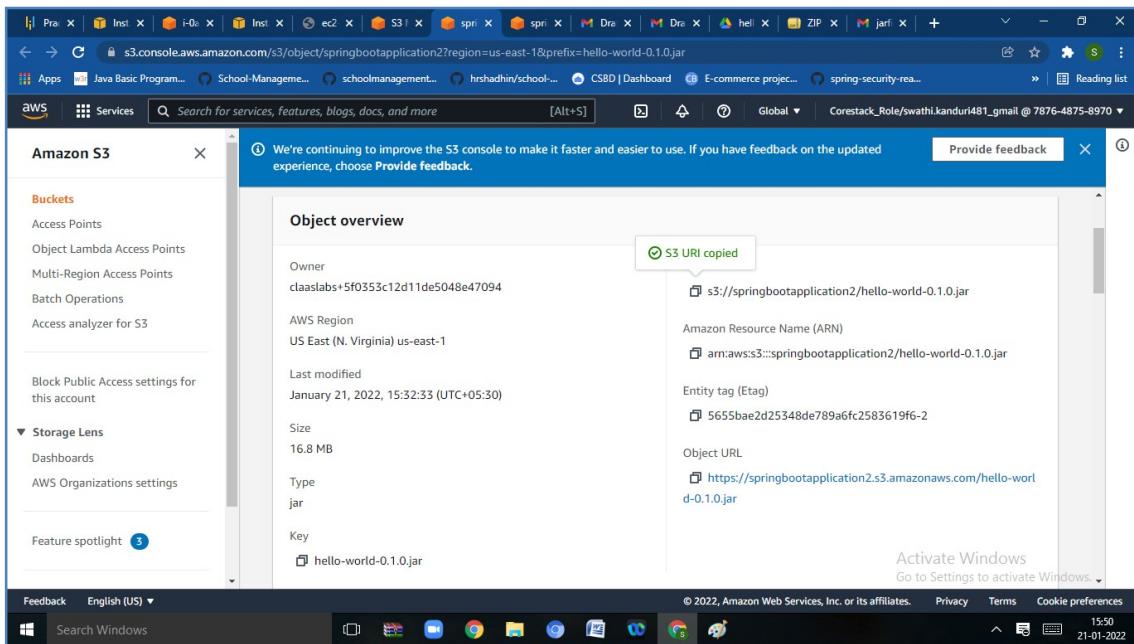
Below the terminal, the AWS CloudWatch interface is visible, showing the instance ID 'i-0a9ef0318cdb210ea' and its public and private IP addresses. A Windows taskbar at the bottom indicates the session is running on a Windows host.

The screenshot shows a terminal window titled 'Amazon Linux 2 AMI' with the URL <https://aws.amazon.com/amazon-linux-2/>. The terminal displays the following text:

```
11 package(s) needed for security, out of 16 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-23-34 ~]$ sudo -i
[root@ip-172-31-23-34 ~]#
```

Below the terminal, the AWS CloudWatch interface is visible, showing the instance ID 'i-0a9ef0318cdb210ea' and its public and private IP addresses. A Windows taskbar at the bottom indicates the session is running on a Windows host.

Copy s3 uri from object overview and paste



The file is downloaded

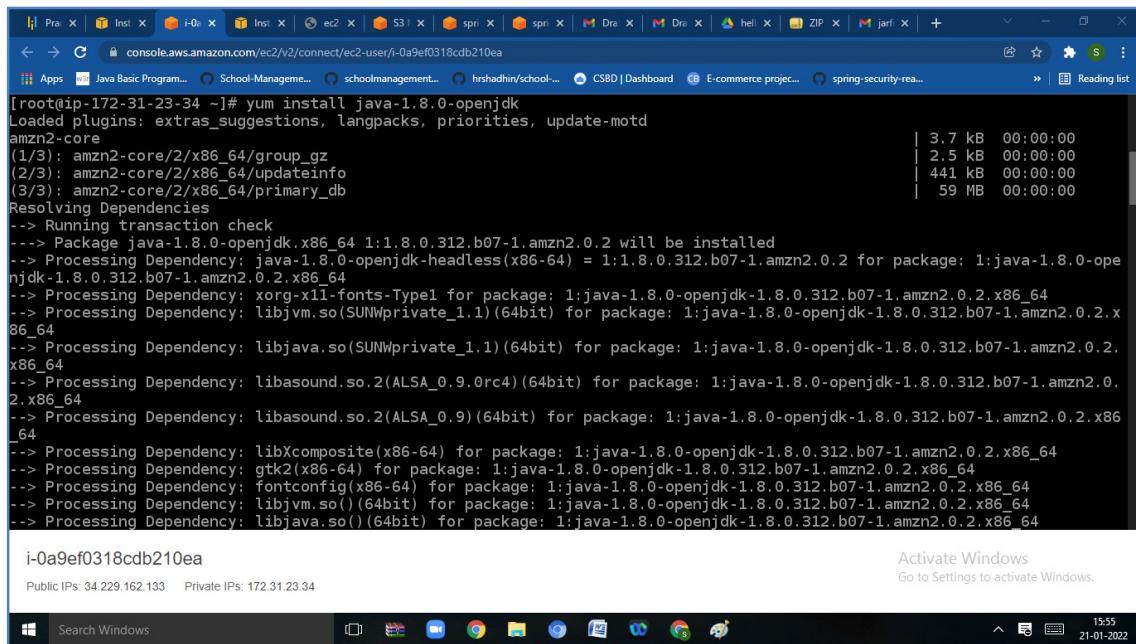
```
^C
[1]+ Done wget https://s3.console.aws.amazon.com/s3/object/springbootapplication2?region=us-east-1
[root@ip-172-31-23-34 ~]# wget https://s3.console.aws.amazon.com/s3/object/springbootapplication2?region=us-east-1&pr
e
[1] 1306
-bash: pre: command not found
[root@ip-172-31-23-34 ~]# fix=hello-world-0.1.0.jar--2022-01-21 10:18:53-- http://wget/
Resolving wget (wget)... failed: Name or service not known.
wget: unable to resolve host address 'wget'
--2022-01-21 10:18:53-- https://s3.console.aws.amazon.com/s3/object/springbootapplication2?region=us-east-1
Resolving s3.console.aws.amazon.com (s3.console.aws.amazon.com) ... 3.3.9.1, 3.3.8.1
Connecting to s3.console.aws.amazon.com (s3.console.aws.amazon.com)|3.3.9.1|:443... connected.
HTTP request sent, awaiting response... 200
Length: 1023 [text/html]
Saving to: 'springbootapplication2?region=us-east-1.2'

100%[=====] 1,023      --.-K/s   in 0s

2022-01-21 10:18:53 (46.0 MB/s) - 'springbootapplication2?region=us-east-1.2' saved [1023/1023]

FINISHED --2022-01-21 10:18:53--
Total wall clock time: 0:1s
Downloaded: 1 files, 1023 in 0s (46.0 MB/s)
```

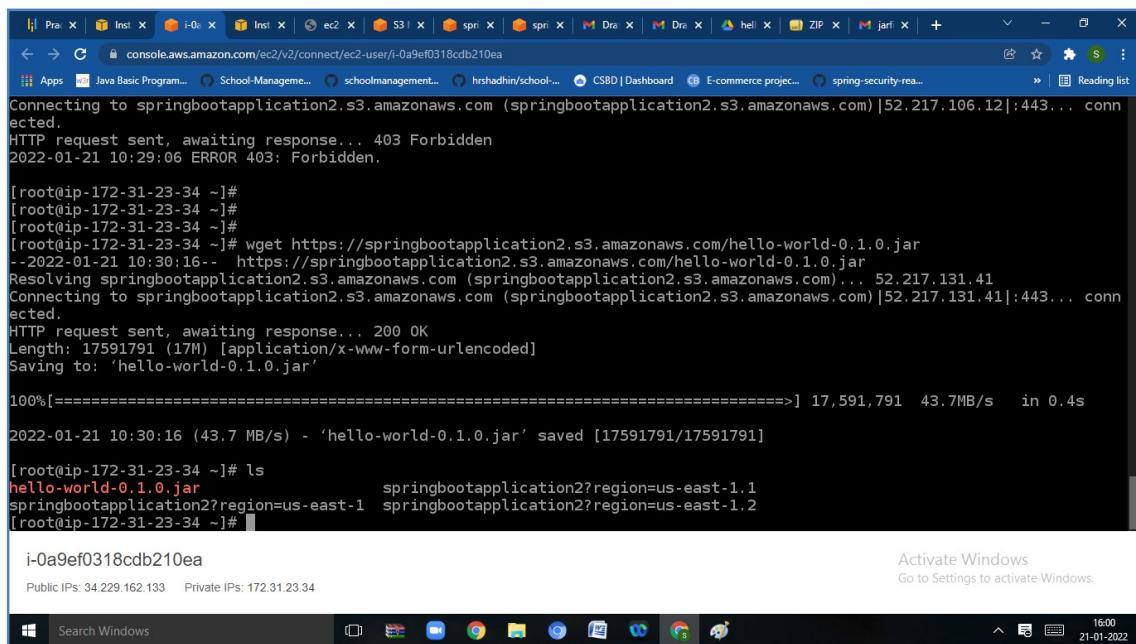
To install java use the command as yum install java



```
[root@ip-172-31-23-34 ~]# yum install java-1.8.0-openjdk
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
(1/3): amzn2-core/2/x86_64/group_gz | 3.7 kB 00:00:00
(2/3): amzn2-core/2/x86_64/updateinfo | 2.5 kB 00:00:00
(3/3): amzn2-core/2/x86_64/primary_db | 441 kB 00:00:00
| 59 MB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package java-1.8.0-openjdk.x86_64 1:1.8.0.312.b07-1.amzn2.0.2 will be installed
--> Processing Dependency: java-1.8.0-openjdk-headless(x86-64) = 1:1.8.0.312.b07-1.amzn2.0.2 for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2
--> Processing Dependency: xorg-x11-fonts-Type1 for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libjvm.so(SUNWprivate_1.1)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libjava.so(SUNWprivate_1.1)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libasound.so.2(ALSA_0.9.0rc4)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libasound.so.2(ALSA_0.9)(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libXcomposite(x86-64) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: gtk2(x86-64) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: fontconfig(x86-64) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libjvm.so()(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
--> Processing Dependency: libjava.so()(64bit) for package: 1:java-1.8.0-openjdk-1.8.0.312.b07-1.amzn2.0.2.x86_64
i-0a9ef0318cdb210ea
Activate Windows
Go to Settings to activate Windows.
Public IPs: 34.229.162.133 Private IPs: 172.31.23.34
15:55
21-01-2022
```

Case 1: through maven jar

wget <https://springbootapplication2.s3.amazonaws.com/hello-world-0.1.0.jar>



```
Connecting to springbootapplication2.s3.amazonaws.com (springbootapplication2.s3.amazonaws.com) |52.217.106.12|:443... connected.
HTTP request sent, awaiting response... 403 Forbidden
2022-01-21 10:29:06 ERROR 403: Forbidden.

[root@ip-172-31-23-34 ~]#
[root@ip-172-31-23-34 ~]#
[root@ip-172-31-23-34 ~]#
[root@ip-172-31-23-34 ~]# wget https://springbootapplication2.s3.amazonaws.com/hello-world-0.1.0.jar
--2022-01-21 10:30:16-- https://springbootapplication2.s3.amazonaws.com/hello-world-0.1.0.jar
Resolving springbootapplication2.s3.amazonaws.com (springbootapplication2.s3.amazonaws.com)... 52.217.131.41
Connecting to springbootapplication2.s3.amazonaws.com (springbootapplication2.s3.amazonaws.com) |52.217.131.41|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 17591791 (17M) [application/x-www-form-urlencoded]
Saving to: 'hello-world-0.1.0.jar'

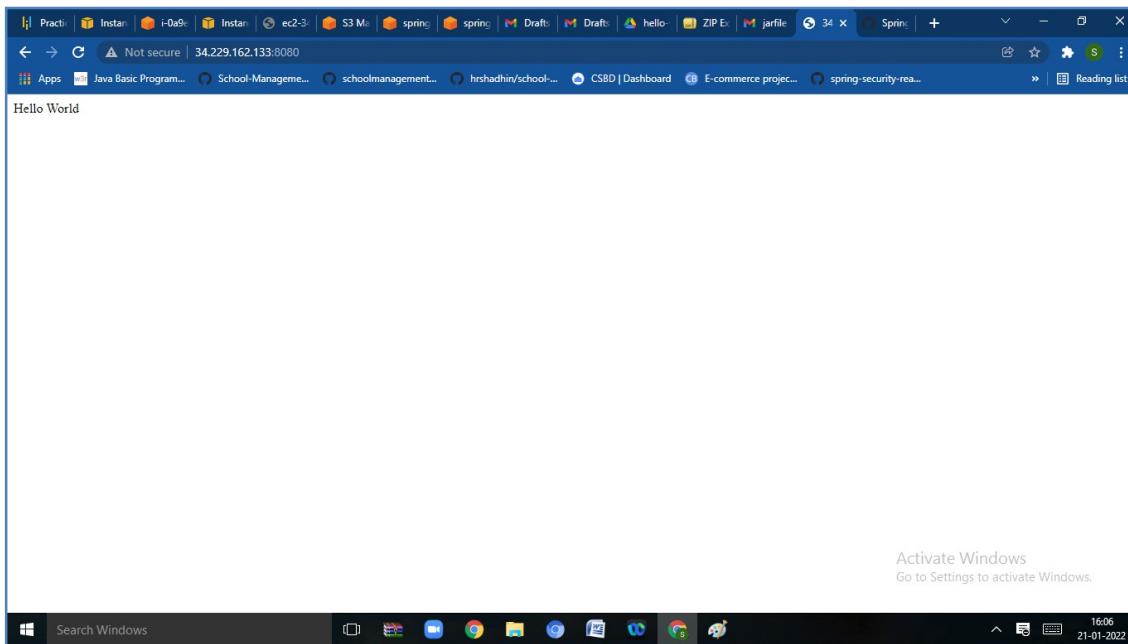
100%[=====] 17,591,791 43.7MB/s in 0.4s

2022-01-21 10:30:16 (43.7 MB/s) - 'hello-world-0.1.0.jar' saved [17591791/17591791]

[root@ip-172-31-23-34 ~]# ls
hello-world-0.1.0.jar      springbootapplication2?region=us-east-1.1
springbootapplication2?region=us-east-1  springbootapplication2?region=us-east-1.2
[root@ip-172-31-23-34 ~]#
i-0a9ef0318cdb210ea
Activate Windows
Go to Settings to activate Windows.
Public IPs: 34.229.162.133 Private IPs: 172.31.23.34
16:00
21-01-2022
```

```
Prj X Inst X i-0e X Inst X ec2 X s3 X spri X spri X Dra X Dre X hell X ZIP X jarf X +  
← → C console.aws.amazon.com/ec2/v2/connect/ec2-user/i-0a9ef0318cdb210ea  
Apps Java Basic Program... School-Manageme... schoolmanagement... hrshadhin/school... CSBD | Dashboard E-commerce projec... spring-security-re...  
:: Spring Boot :: (v2.2.1.RELEASE)  
  
2022-01-21 10:32:08.895 INFO 4646 --- [           main] c.d.hello.Application      : Starting Application v  
0.1.0 on ip-172-31-23-34.ec2.internal with PID 4646 (/root/hello-world-0.1.0.jar started by root in /root)  
2022-01-21 10:32:08.907 INFO 4646 --- [           main] c.d.hello.Application      : No active profile set,  
falling back to default profiles: default  
2022-01-21 10:32:11.404 INFO 4646 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized wit  
h port(s): 8080 (http)  
2022-01-21 10:32:11.431 INFO 4646 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomc  
at]  
2022-01-21 10:32:11.432 INFO 4646 --- [           main] o.apache.catalina.core.StandardEngine : Starting Servlet engin  
e: [Apache Tomcat/9.0.27]  
2022-01-21 10:32:11.570 INFO 4646 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]      : Initializing Spring em  
beddedWebApplicationContext  
2022-01-21 10:32:11.574 INFO 4646 --- [           main] o.s.web.context.ContextLoader       : Root WebApplicationCon  
text: initialization completed in 2550 ms  
2022-01-21 10:32:12.941 INFO 4646 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorS  
ervice 'applicationTaskExecutor'  
2022-01-21 10:32:13.367 INFO 4646 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port  
(s): 8080 (http) with context path ''  
2022-01-21 10:32:13.370 INFO 4646 --- [           main] c.d.hello.Application      : Started Application in  
5.563 seconds (JVM running for 6.433)  
  
i-0a9ef0318cdb210ea  
Activate Windows  
Public IPs: 34.229.162.133 Private IPs: 172.31.23.34 Go to Settings to activate Windows.  
Windows Search Windows 1603 21-01-2022
```

Execute the response in the Browser with public IP and port



Case 2: through Docker Image/Docker Hub

Step1: install java in ec2:-

Commands :- yum install java -1.8.0-openjdk

Step2: install Jenkins:-

Yum install Jenkins

Step3:-install Maven:-

Yum install maven

Step4:-install docker:-

Yum install docker

Configure Jenkins with all necessary plugins

Install docker related plugins in Jenkins using manage Jenkins

Configure Jenkins job for spring boot project using scm repository

Docker login

```
https://aws.amazon.com/amazon-linux-2/
11 package(s) needed for security, out of 15 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-23-34 ~]$ sudo -i
[root@ip-172-31-23-34 ~]# docker login
Authenticating with existing credentials...
Login did not succeed, error: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username (swathikothapeta): y
Password:
Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
[root@ip-172-31-23-34 ~]# docker login
Authenticating with existing credentials...
Login did not succeed, error: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username (swathikothapeta): swathikothapeta
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

i-0a9ef0318cdb210ea
```

Activate Windows
Go to Settings to activate Windows.

Public IPs: 34.229.162.133 Private IPs: 172.31.23.34

Windows Search bar, Taskbar with various icons, Date: 21-01-2022

Restart the Docker and pull the image

```
Redirecting to /bin/systemctl stop docker.service
[root@ip-172-31-23-34 ~]# docker status
docker: 'status' is not a docker command.
See 'docker --help'.
[root@ip-172-31-23-34 ~]# docker --version
Docker version 20.10.7, build f0df350
[root@ip-172-31-23-34 ~]# docker images
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
[root@ip-172-31-23-34 ~]# sudo usermod -aG docker $USER
[root@ip-172-31-23-34 ~]# systemctl restart docker
[root@ip-172-31-23-34 ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
[root@ip-172-31-23-34 ~]# docker pull swathikothapeta/test_repo_1234:latest
latest: Pulling from swathikothapeta/test_repo_1234
155aced26663: Pull complete
ac5901c58ecb: Pull complete
5fcacf14fb826: Pull complete
5eba28ea4ae9: Pull complete
Digest: sha256:163737e8922e1ee72a749ce66ccb47b2ceb01099a30602cf7b52ec901e194ad4
Status: Downloaded newer image for swathikothapeta/test_repo_1234:latest
docker.io/swathikothapeta/test_repo_1234:latest
[root@ip-172-31-23-34 ~]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
swathikothapeta/test_repo_1234 latest cb2269f3d869 19 hours ago 489MB
[root@ip-172-31-23-34 ~]# docker run -p 7080:8080 cb2269f3d869

i-0a9ef0318cdb210ea
```

Activate Windows
Go to Settings to activate Windows.

Public IPs: 34.229.162.133 Private IPs: 172.31.23.34

Windows Search bar, Taskbar with various icons, Date: 21-01-2022

Docker run -p 7080:8080 and cb2269f3d869(imageid)

The screenshot shows a terminal window with the following content:

```
[root@ip-172-31-23-34 ~]# docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
swathikothapeta/test_repo_1234    latest   cb2269f3d869  19 hours ago  489MB
[root@ip-172-31-23-34 ~]# docker run -p 7080:8080 cb2269f3d869

.
.
.

:: Spring Boot ::      (v2.2.1.RELEASE)

2022-01-21 11:22:49.824  INFO 1 --- [           main] c.d.hello.Application      : Starting Application v0.1
2022-01-21 11:22:49.832  INFO 1 --- [           main] c.d.hello.Application      : No active profile set, falling back to default profiles: default
2022-01-21 11:22:52.157 INFO 1 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-01-21 11:22:52.187 INFO 1 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-01-21 11:22:52.194 INFO 1 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.27]
2022-01-21 11:22:52.328 INFO 1 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]      : Initializing Spring embedded WebApplicationContext
2022-01-21 11:22:52.334 INFO 1 --- [           main] o.s.web.context.ContextLoader      : Root WebApplicationContext
```

At the bottom of the terminal, it shows:

i-0a9ef0318cdb210ea
Activate Windows
Go to Settings to activate Windows.
Public IPs: 34.229.162.133 Private IPs: 172.31.23.34

The taskbar at the bottom shows various icons and the date/time: 17:10 21-01-2022.

Execution in browser

