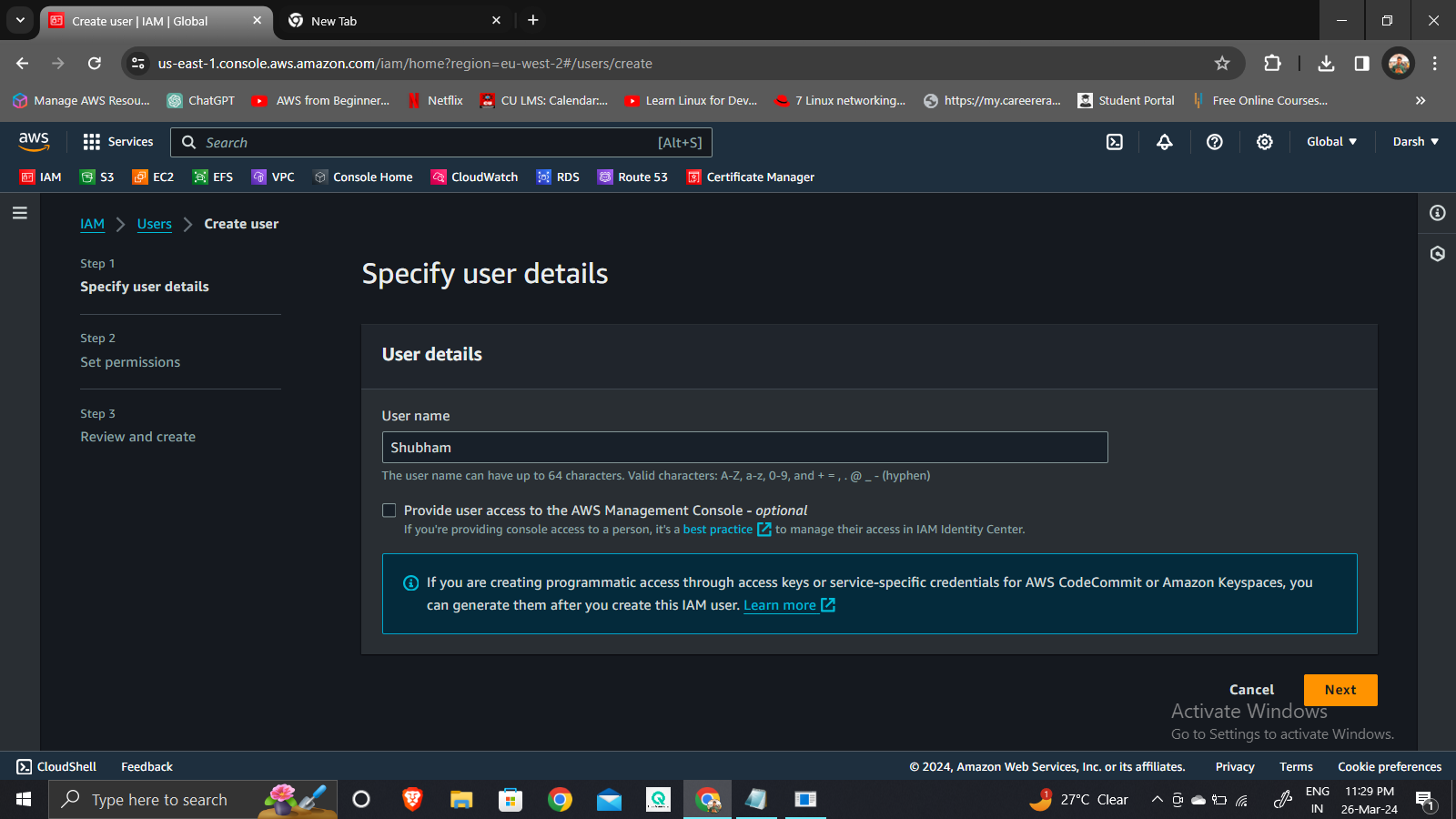
## Darshan Nikam Date: 26/03/2024

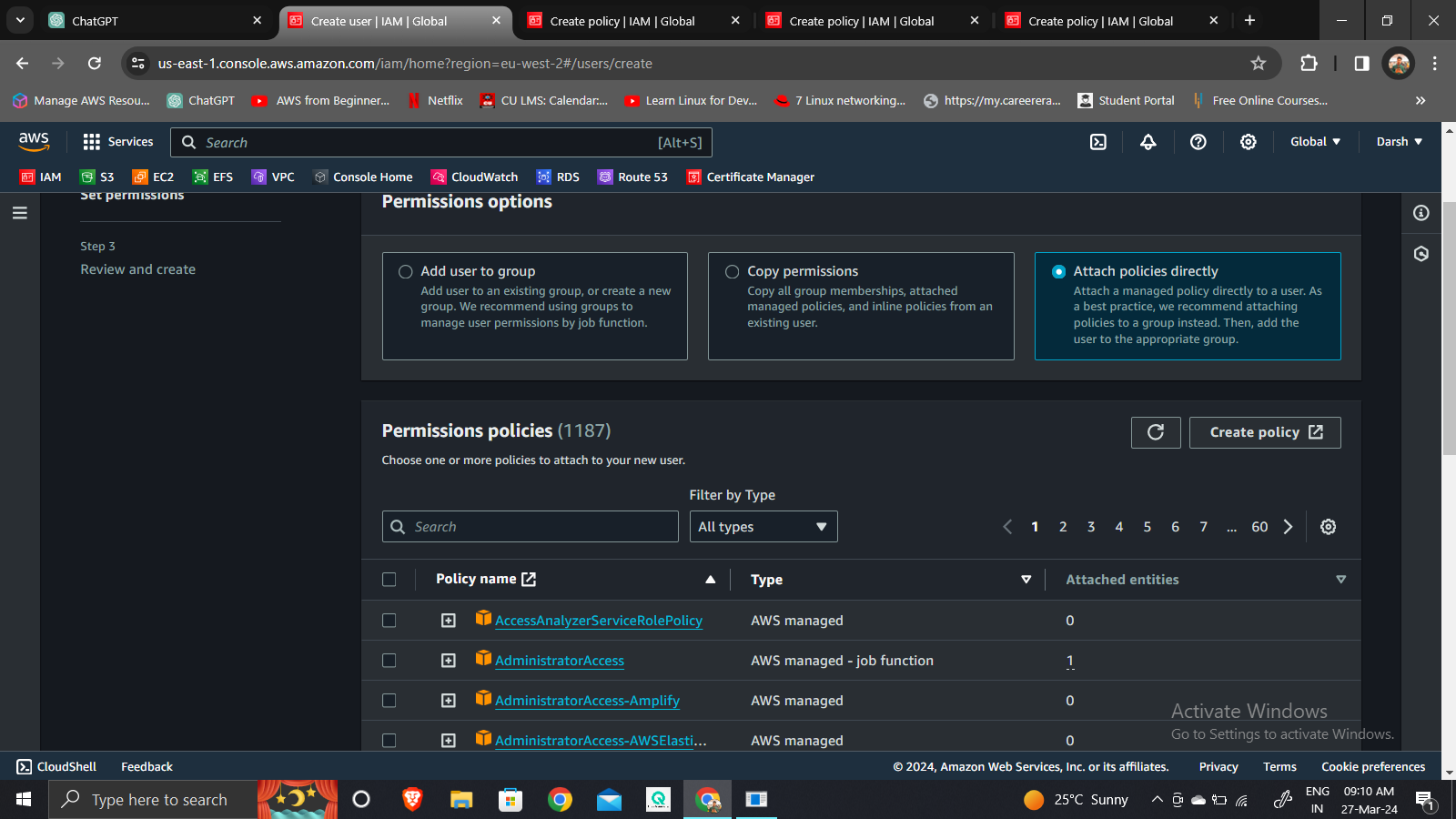
### **Practice Tasks**

1. Create user Shubham. Shubham can only launch ec2 instance in the Mumbai region.

* Go to IAM Service, Click on Create User, Enter user name, and click on Next.



* Select the Attach policies directly option, and click on Create Policy.



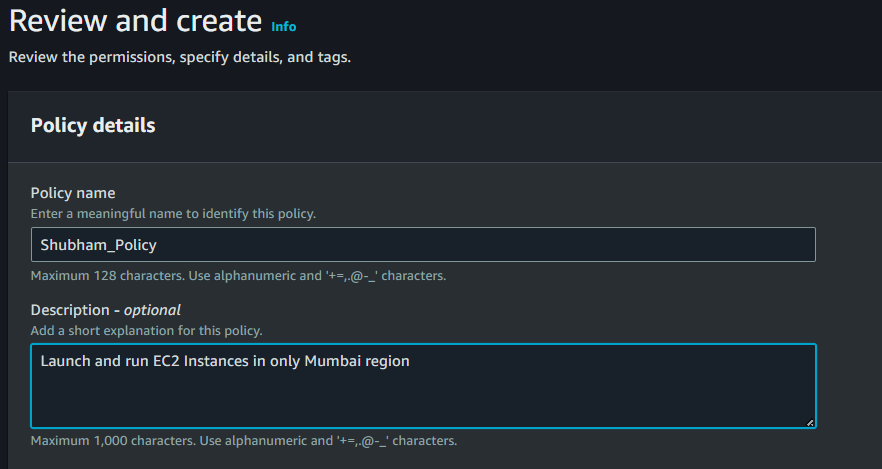
* The IAM policy we need to grant the Shubham user the following permissions to launch Ec2 Instance
* Launch EC2 instances (ec2:RunInstances) and describe instances (ec2:DescribeInstances).
* Create tags for EC2 instances (ec2:CreateTags).
* Describe available key pairs (ec2:DescribeKeyPairs), images (ec2:DescribeImages), instance types (ec2:DescribeInstanceTypes), VPCs (ec2:DescribeVpcs), and subnets (ec2:DescribeSubnets).
* Manage security groups, including creating (ec2:CreateSecurityGroup), deleting (ec2:DeleteSecurityGroup), and authorizing ingress rules (ec2:AuthorizeSecurityGroupIngress).
* Check the status of instances (ec2:DescribeInstanceStatus).
* Describe available availability zones (ec2:DescribeAvailabilityZones) and security groups (ec2:DescribeSecurityGroups).

These permissions are restricted to the Mumbai region (ap-south-1) due to the specified condition. With this policy, Shubham will be able to launch, manage, and monitor EC2 instances effectively within the Mumbai region.

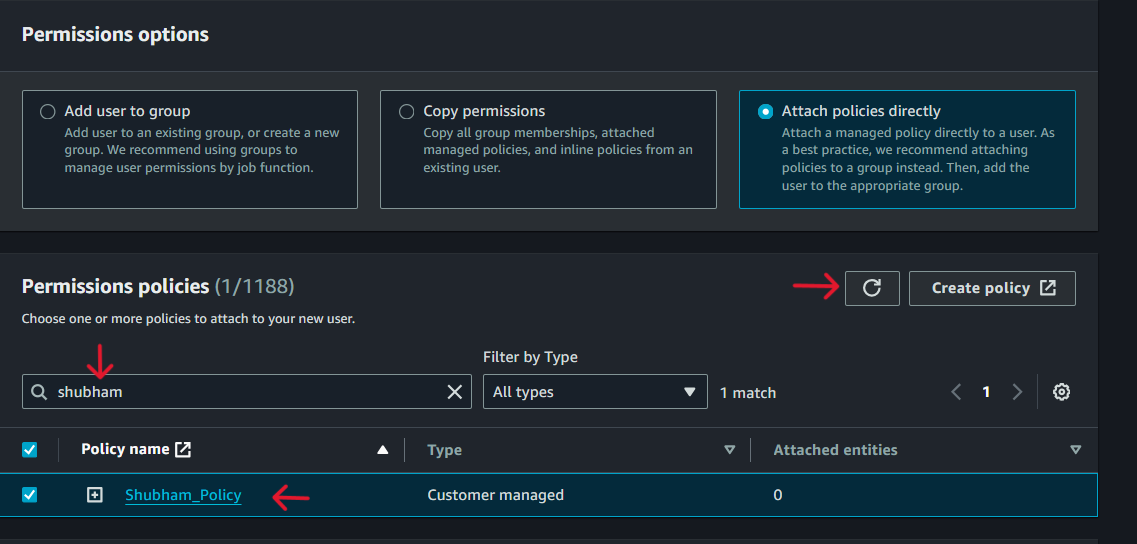
* Select JSON format and enter the below policy.

|  |
| --- |
| {  "Version": "2012-10-17",  "Statement": [  {  "Effect": "Allow",  "Action": [  "ec2:RunInstances",  "ec2:DescribeInstances",  "ec2:CreateTags",  "ec2:DescribeKeyPairs",  "ec2:DescribeImages",  "ec2:DescribeInstanceTypes",  "ec2:DescribeVpcs",  "ec2:DescribeSubnets",  "ec2:CreateSecurityGroup",  "ec2:DeleteSecurityGroup",  "ec2:AuthorizeSecurityGroupIngress",  "ec2:DescribeInstanceStatus"  ],  "Resource": "\*",  "Condition": {  "StringEquals": {  "ec2:Region": "ap-south-1"  }  }  },  {  "Effect": "Allow",  "Action": "ec2:DescribeAvailabilityZones",  "Resource": "\*"  },  {  "Effect": "Allow",  "Action": "ec2:DescribeSecurityGroups",  "Resource": "\*"  }  ]  } |

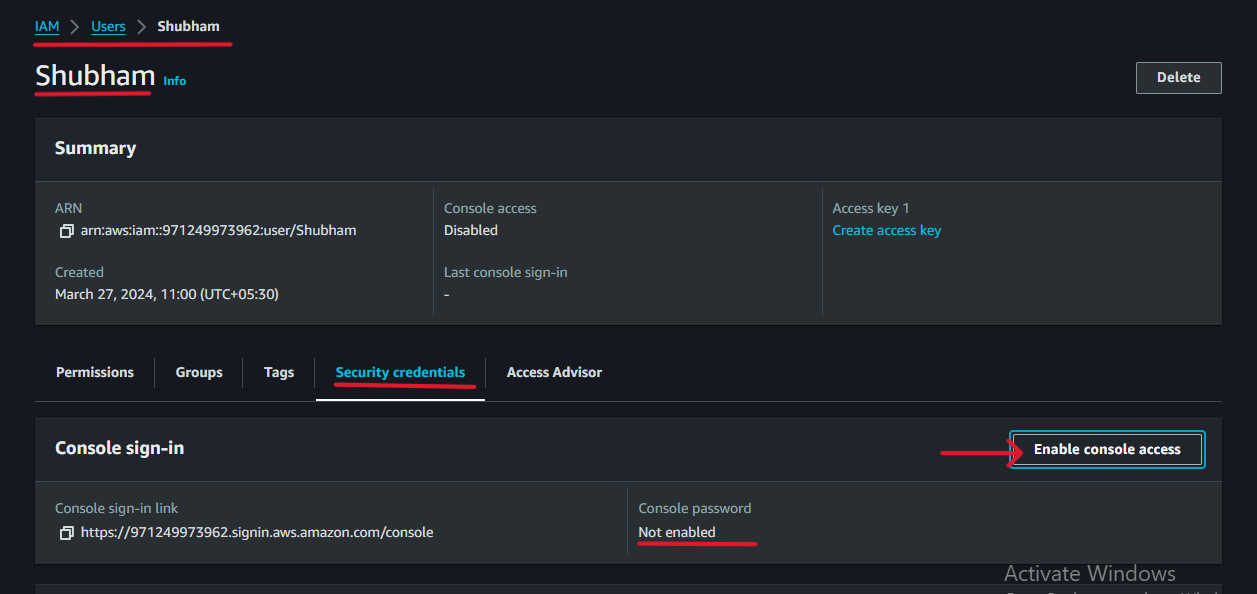
* Click on the Next button, name the policy, and on the Create policy button.



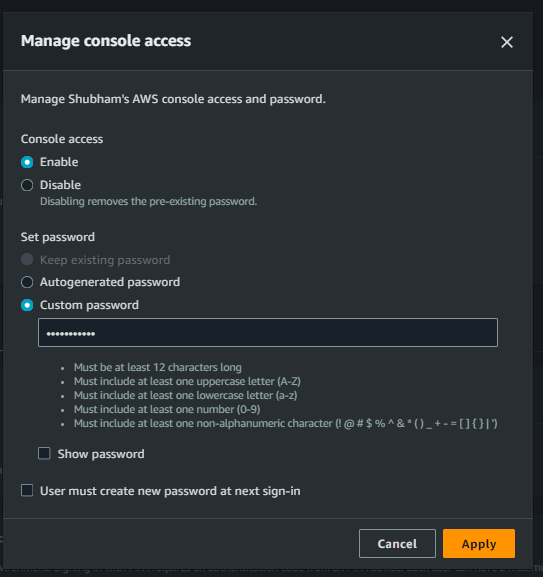
* Now Go back to the Create User tab, refresh the policy, search Shubham\_Policy that we created earlier in the search box, select the policy, and click on the next button.



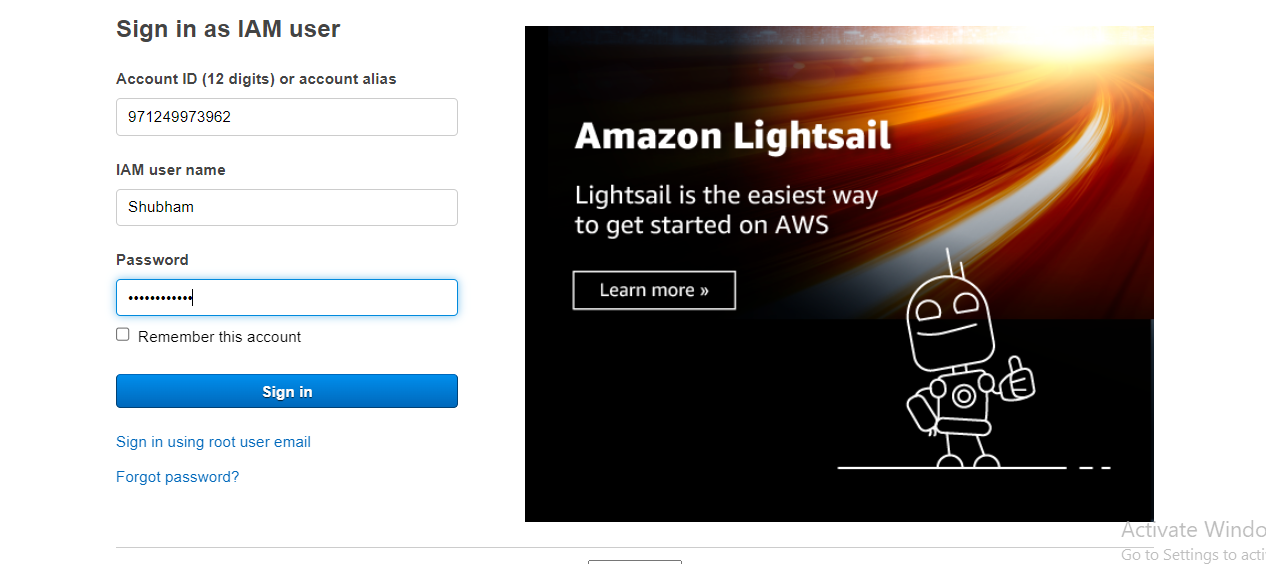
* Review the user details, and policy and click on Create User, the user will be created
* Now we need to Enable Console Access for user Shubham.
* Click on the user name, select the Security credential option, and click on the Enable console Access button.



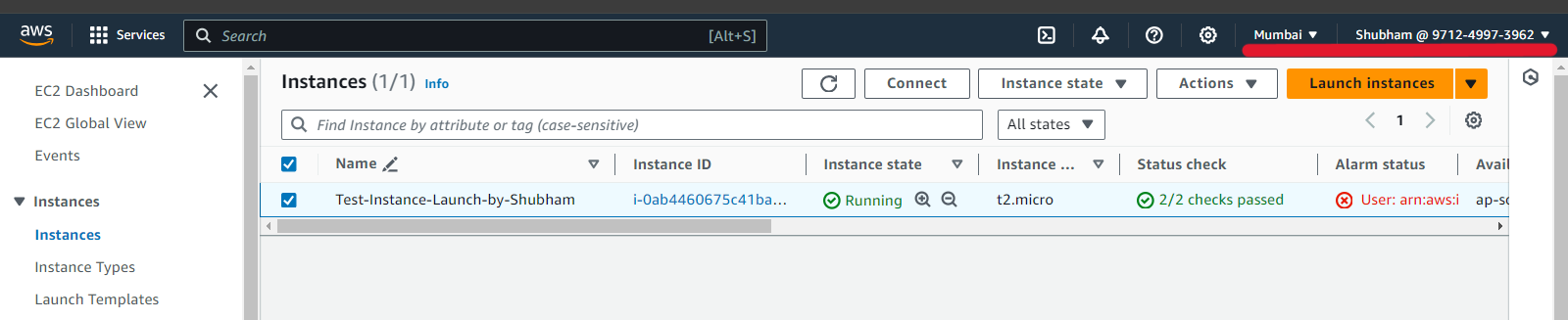
* Click on Enable, Select Custom Password, Enter password, and click on the Apply button.



* Now Copy the console sign link and open it in the new tab, enter the username and password, and sign in.

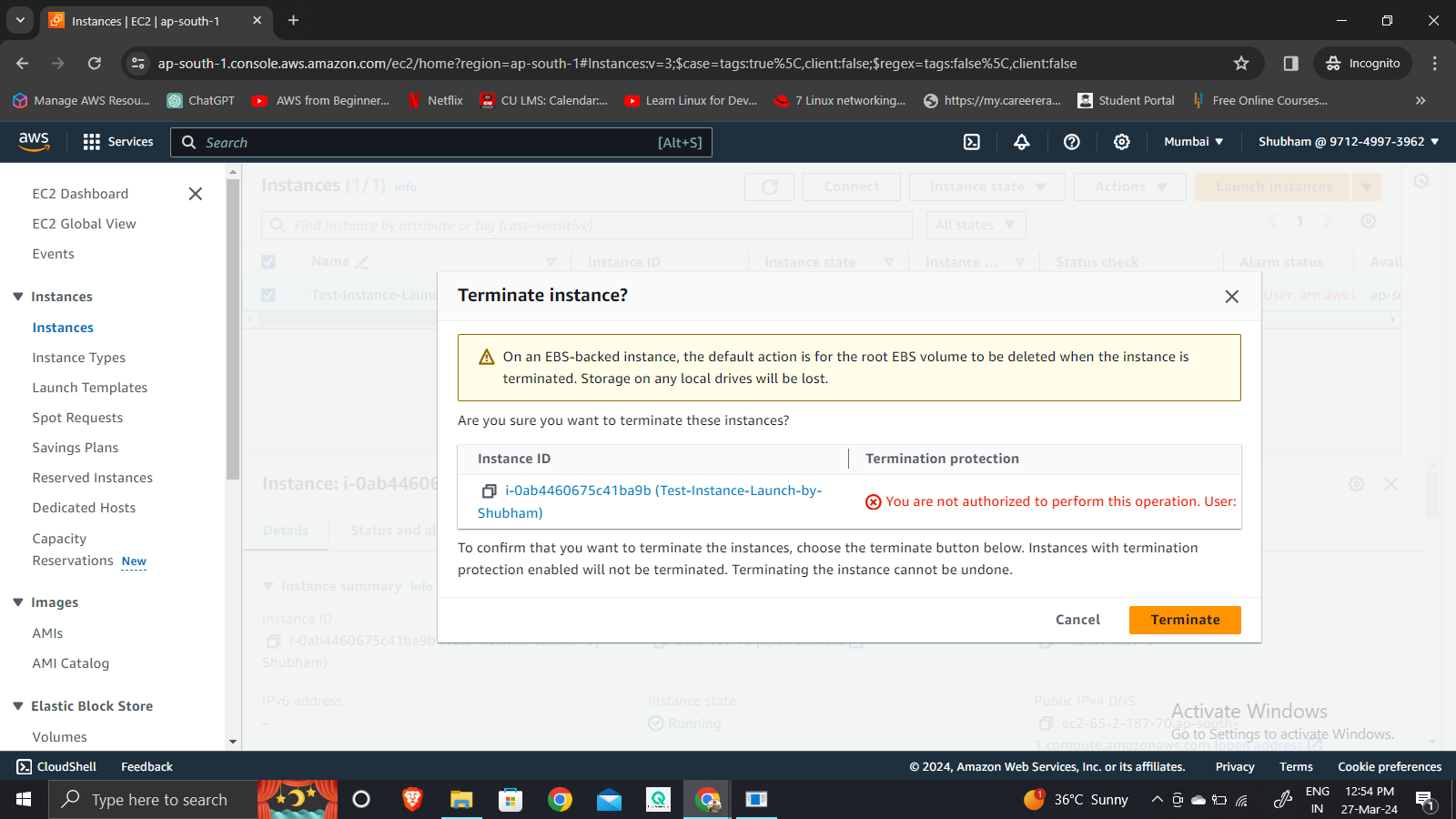


* Now launch the instance in the Mumbai region.



Here we can see that user Shubham has successfully launched the Ec2 instance in the Mumbai region.

* Now try to terminate the Instance.



User Shubham has no permission to terminate the Instance.

Summary of task

Creating a custom IAM policy for Shubham is like giving him a personalized access pass for specific tasks within AWS. This pass only allows him to do certain things, like launching EC2 instances in a particular region, managing security settings, and checking instance statuses. It's like a set of rules tailored just for him, ensuring he can do his job effectively without accessing anything unnecessary or risky.

This approach enhances security by limiting Shubham's access to only what needs, making sure he follows company policies and compliance requirements. And if Shubham's role changes or new team members join, we can easily update his Policy to fit their responsibilities.

In simple terms, custom IAM policies help Shubham work safely and efficiently within AWS, without giving him too much access or leaving anything important out.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Create an S3 bucket and share it with another AWS account.