

IAM Policies

Problem Statement:

You work for XYZ Corporation. To maintain the security of the AWS account and the resources you have been asked to implement a solution that can help easily recognize and monitor the different users.

Tasks To Be Performed:

1. Create policy number 1 which lets the users to:
 - a. Access S3 completely
 - b. Only create EC2 instances
 - c. Full access to RDS
2. Create a policy number 2 which allows the users to:
 - a. Access CloudWatch and billing completely
 - b. Can only list EC2 and S3 resources.
3. Attach policy number 1 to the Dev Team from task 1
4. Attach policy number 2 to Ops Team from Task 1.

Dev1 user

<https://219449977354.signin.aws.amazon.com/console>

Password for dev1- Dev1awsgurus

Dev2 user

<https://219449977354.signin.aws.amazon.com/console>

password for dev2- Dev2awsgurus

Test1 user

<https://219449977354.signin.aws.amazon.com/console>

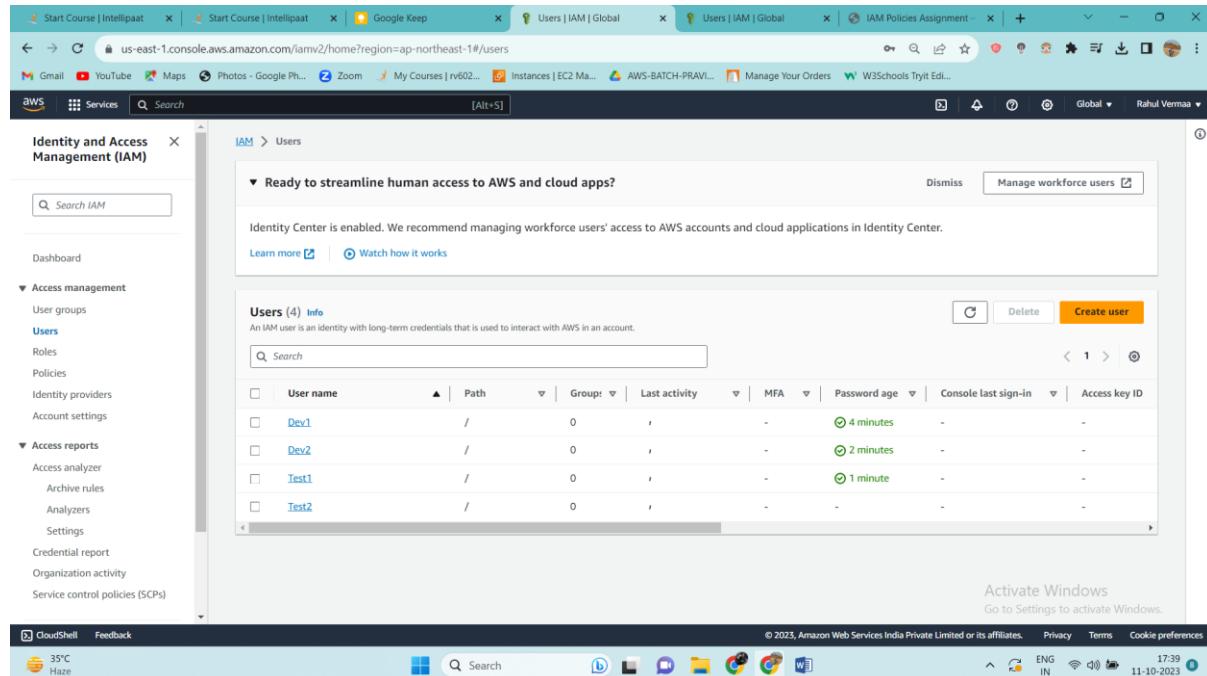
password for test1- Test1awsgurus

Test2 user

<https://219449977354.signin.aws.amazon.com/console>

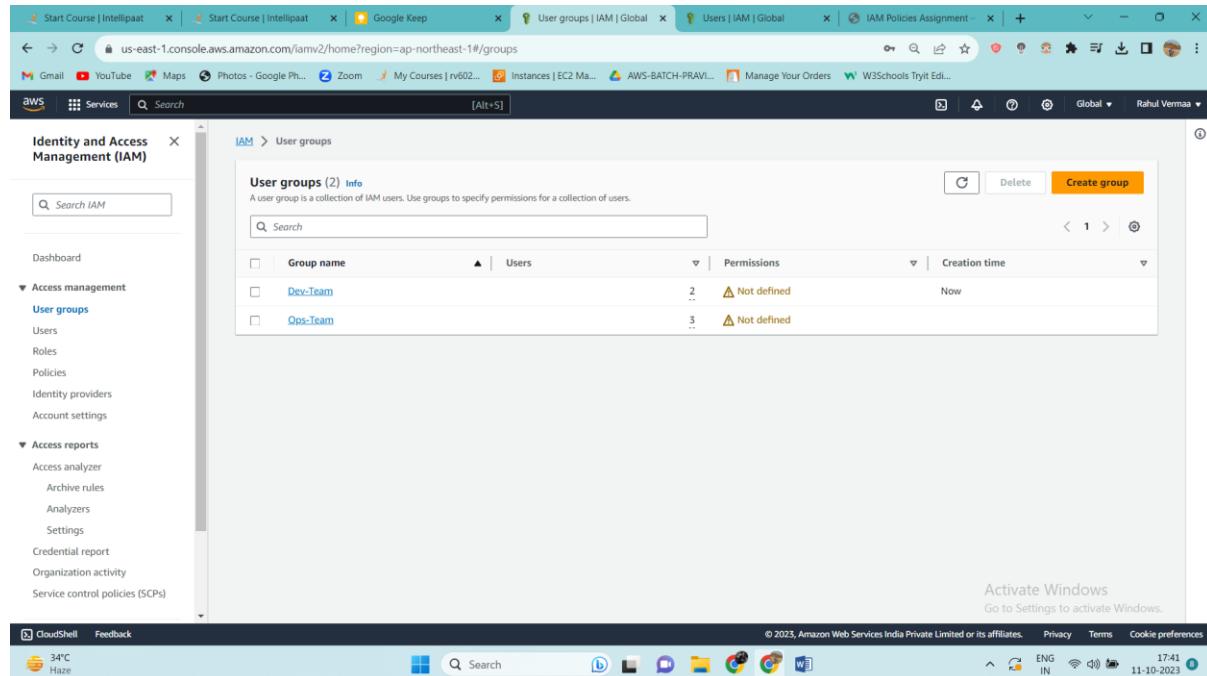
password for test2- Test2awsgurus

We have created these users in our IAM users PDF



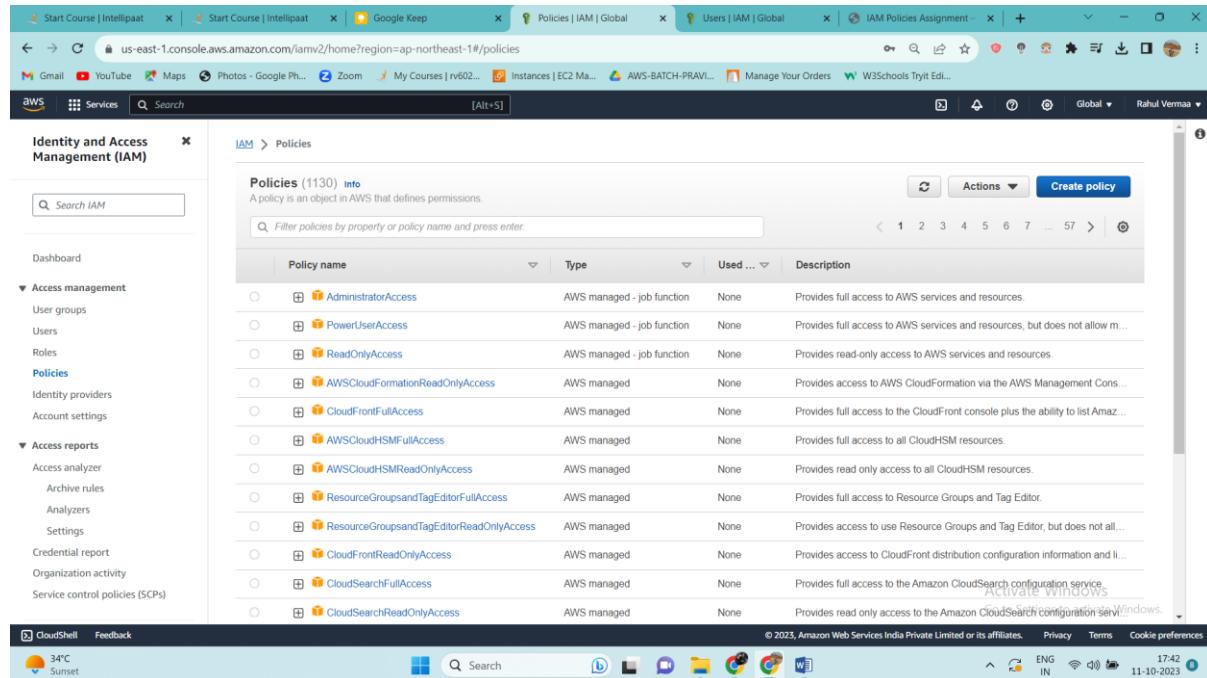
User name	Path	Group	Last activity	MFA	Password age	Console last sign-in	Access key ID
Dev1	/	0	-	-	4 minutes	-	-
Dev2	/	0	-	-	2 minutes	-	-
Test1	/	0	-	-	1 minute	-	-
Test2	/	0	-	-	-	-	-

And in our groups we havn't defined any ploicies aur permissions



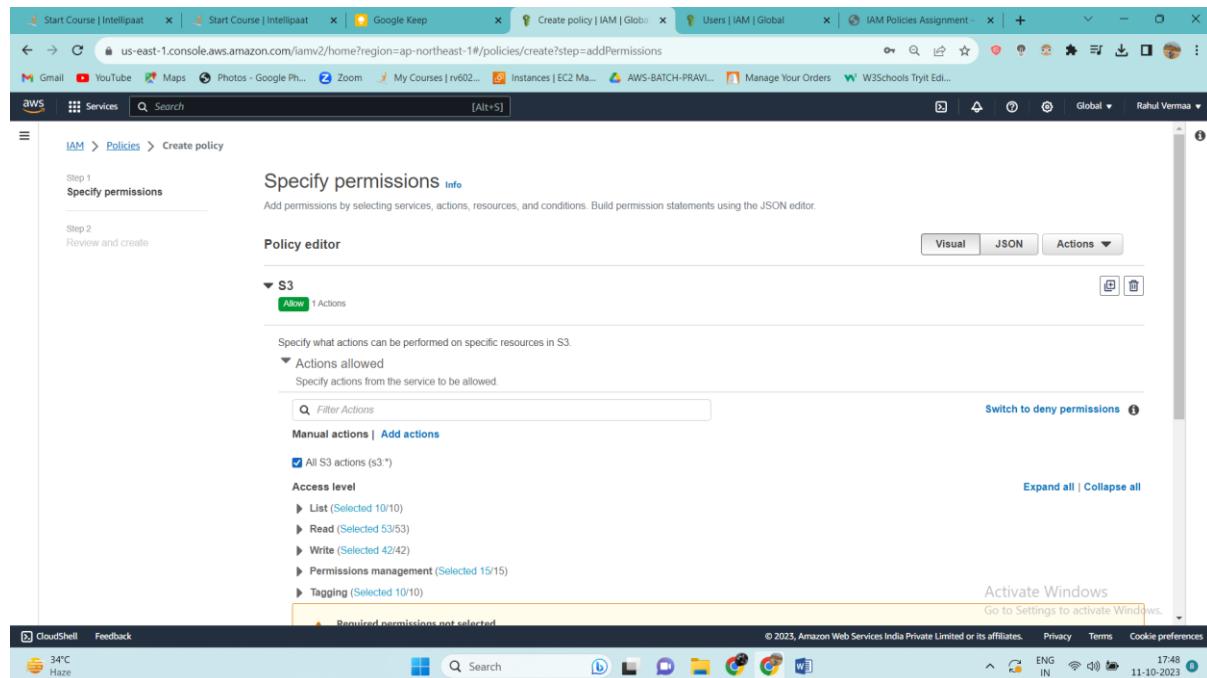
Group name	Users	Permissions	Creation time
Dev-Team	2	Not defined	Now
Ops-Team	3	Not defined	-

Now let's create policies



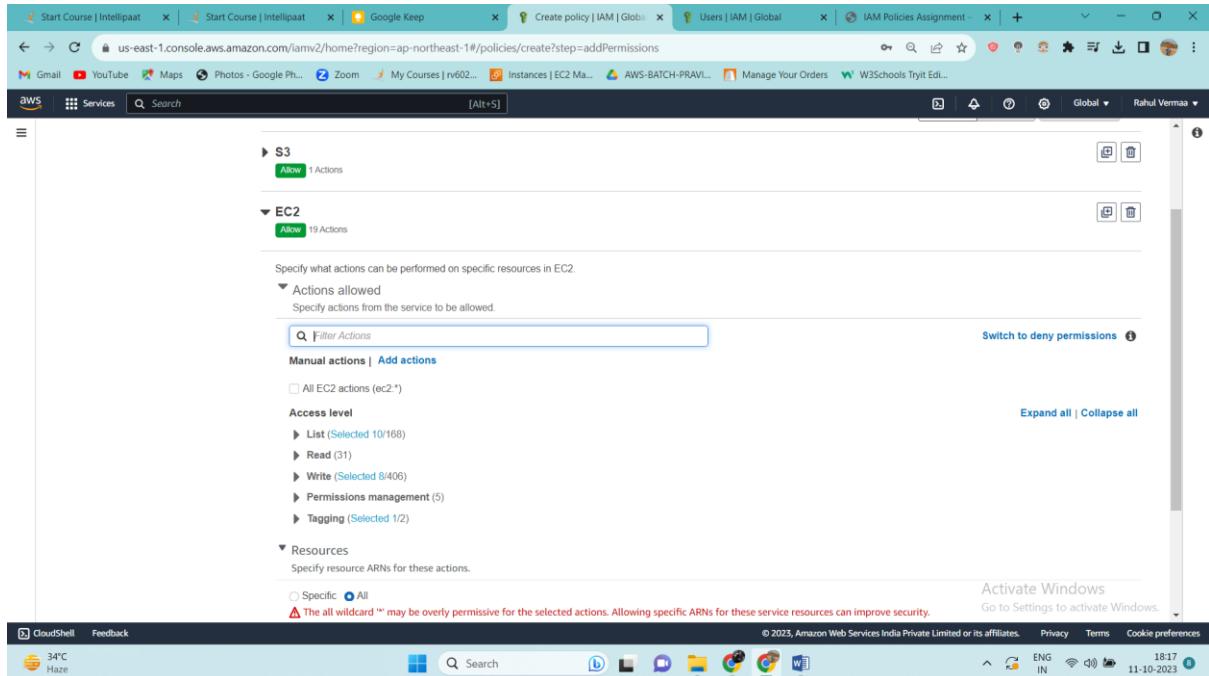
The screenshot shows the AWS IAM Policies page. The left sidebar is collapsed, and the main content area displays a table of 1130 policies. The table columns are: Policy name, Type, Used ..., and Description. The policies listed are: AdministratorAccess, PowerUserAccess, ReadOnlyAccess, AWSCloudFormationReadOnlyAccess, CloudFrontFullAccess, AWSCloudHSMFullAccess, AWSCloudHSMReadOnlyAccess, ResourceGroupsandTagEditorFullAccess, ResourceGroupsandTagEditorReadOnlyAccess, CloudFrontReadOnlyAccess, CloudSearchFullAccess, and CloudSearchReadOnlyAccess. The 'Description' column for each policy provides a brief summary of its permissions.

We have to allow all resources for s3 (full access)



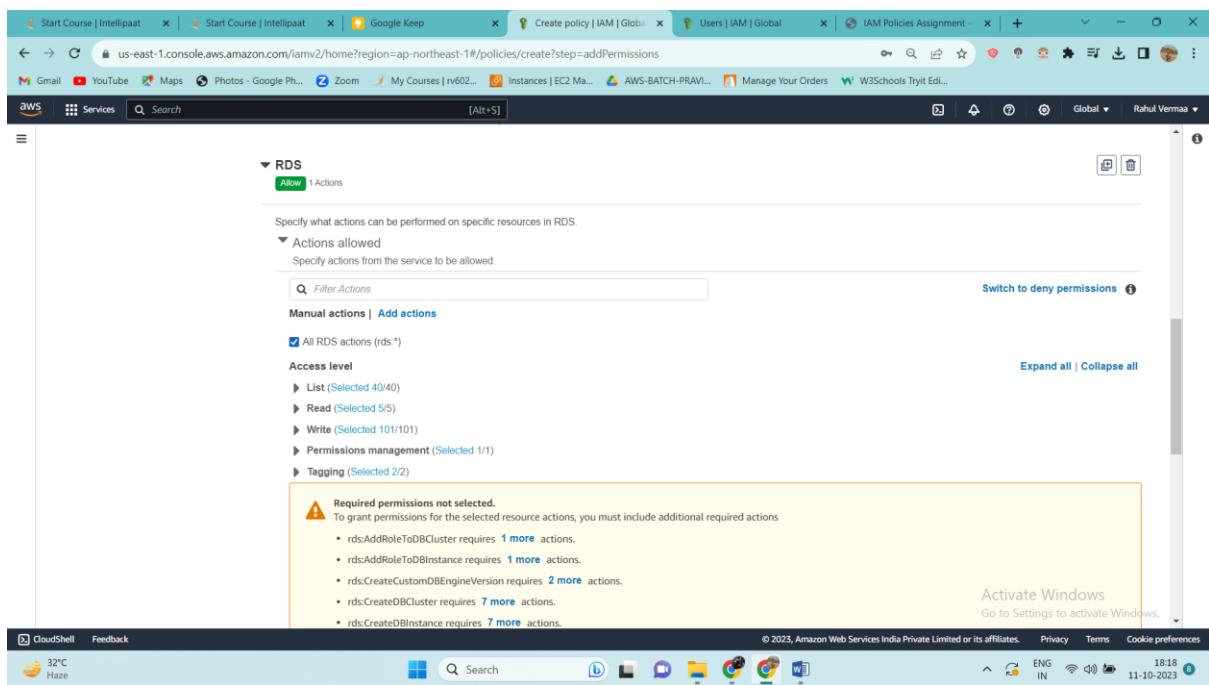
The screenshot shows the 'Create policy' wizard, Step 1: Specify permissions. The left sidebar shows 'Step 1: Specify permissions' and 'Step 2: Review and create'. The main content area is titled 'Specify permissions' and includes a note: 'Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.' Below this is a 'Policy editor' section with tabs for 'Visual' and 'JSON'. A 'S3' section is expanded, showing an 'Actions' list with 'Allow' and '1 Actions'. The 'Actions allowed' section lists 'Actions allowed' and 'Specify actions from the service to be allowed'. A 'Filter Actions' input field is present. The 'Manual actions' section has a 'Add actions' button and a checked checkbox for 'All S3 actions (s3:*)'. The 'Access level' section lists actions: List (Selected 10/10), Read (Selected 53/53), Write (Selected 42/42), Permissions management (Selected 15/15), and Tagging (Selected 10/10). A 'Required permissions not selected' message is at the bottom. The bottom of the screen shows the Windows taskbar with various icons and the date/time: 11-10-2023, 17:42, ENG IN.

Now we have defined for ec2(should be able to create instance only)



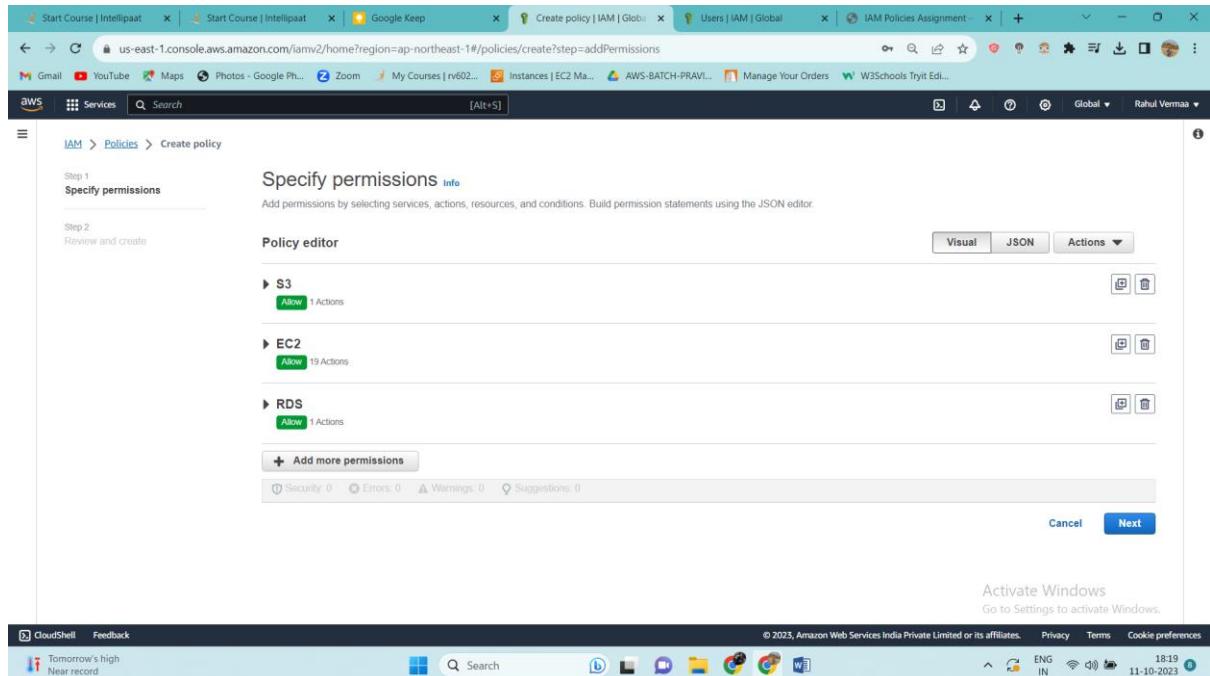
The screenshot shows the AWS IAM Policy creation interface. The policy is titled 'Create policy [IAM | Global]'. The 'EC2' section is expanded, showing 19 actions allowed. The 'Actions allowed' section is expanded, showing 'List' (10/168 selected), 'Read' (31), 'Write' (8/406 selected), 'Permissions management' (5), and 'Tagging' (1/2 selected). The 'Resources' section is collapsed. The status bar at the bottom shows 'CloudShell Feedback' and the date '11-10-2023'.

Now we will allow RDS full access



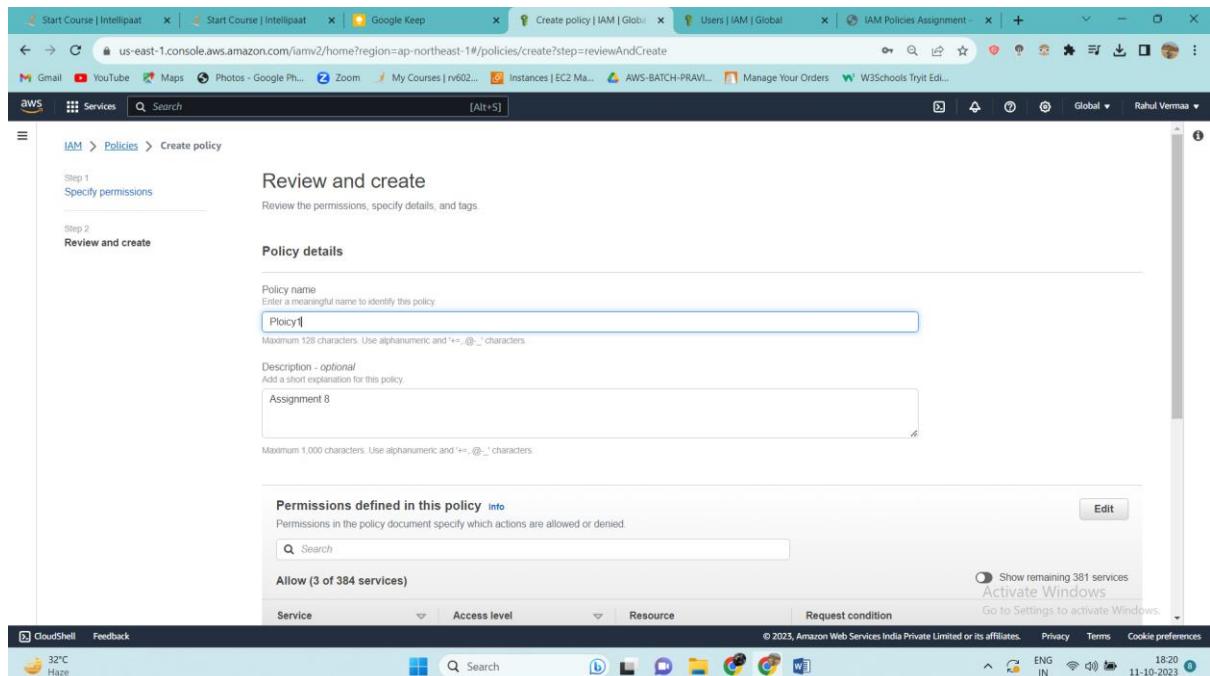
The screenshot shows the AWS IAM Policy creation interface. The policy is titled 'Create policy [IAM | Global]'. The 'RDS' section is expanded, showing 1 action allowed. The 'Actions allowed' section is expanded, showing 'List' (40/40 selected), 'Read' (5/5), 'Write' (101/101 selected), 'Permissions management' (1/1 selected), and 'Tagging' (2/2 selected). A warning message in a yellow box states: 'Required permissions not selected. To grant permissions for the selected resource actions, you must include additional required actions.' It lists several actions required: 'rds:AddRoleToDBCluster' (1 more), 'rds:AddRoleToDBInstance' (1 more), 'rds:CreateCustomDBEngineVersion' (2 more), 'rds:CreateDBCluster' (7 more), and 'rds:CreateDBInstance' (7 more). The status bar at the bottom shows 'CloudShell Feedback' and the date '11-10-2023'.

We have defined permissions for our policy



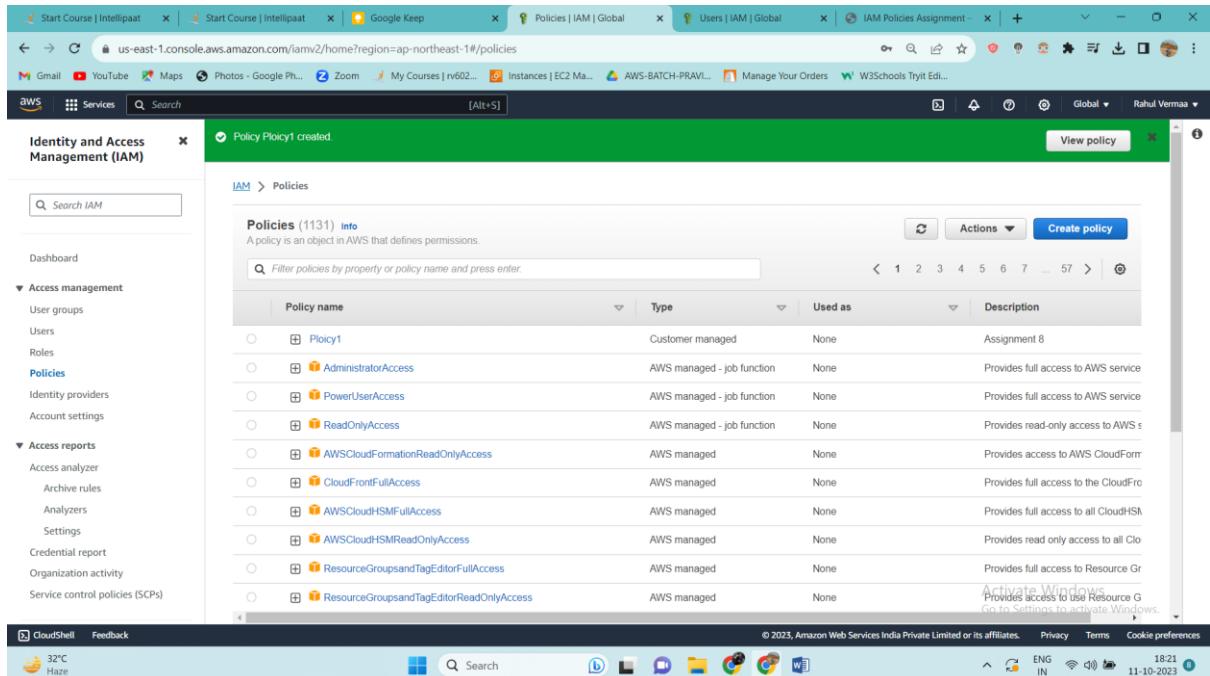
The screenshot shows the 'Specify permissions' step of the IAM policy creation wizard. It lists three service-based permissions: S3, EC2, and RDS, each with specific actions selected. The S3 section shows 1 action, EC2 shows 19 actions, and RDS shows 1 action. A 'Next' button is visible at the bottom.

Define policy name



The screenshot shows the 'Review and create' step of the IAM policy creation wizard. It displays the policy name 'Policy1' and a description 'Assignment 8'. The 'Permissions defined in this policy' section shows 'Allow (3 of 384 services)'. A table lists the services, access levels, resources, and request conditions for the selected permissions. A 'Next' button is visible at the bottom.

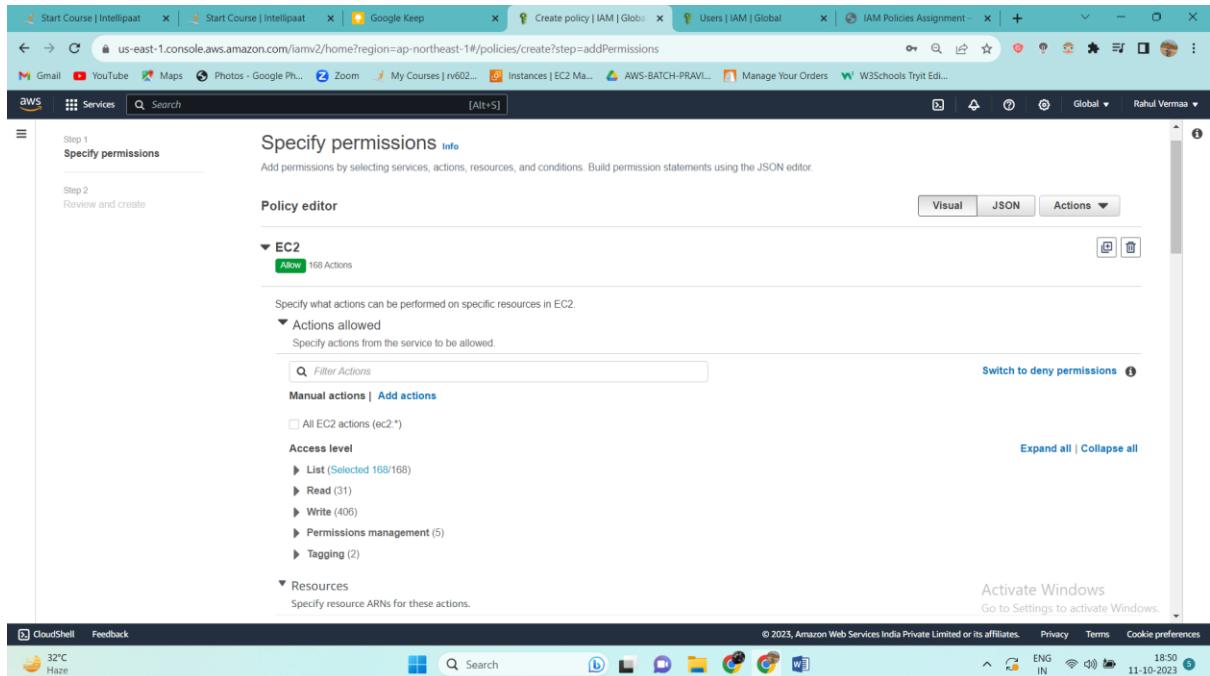
Our policy1 is created successfully



The screenshot shows the AWS IAM Policies page. A green banner at the top indicates 'Policy Policy1 created.' A table below lists 1131 policies. The first policy, 'Policy1', is highlighted. The table columns are 'Policy name', 'Type', 'Used as', and 'Description'. The 'Description' column for 'Policy1' states 'Assignment 8'.

Policy name	Type	Used as	Description
Policy1	Customer managed	None	Assignment 8
AdministratorAccess	AWS managed - job function	None	Provides full access to AWS service
PowerUserAccess	AWS managed - job function	None	Provides full access to AWS service
ReadOnlyAccess	AWS managed - job function	None	Provides read-only access to AWS
AWSCloudFormationReadOnlyAccess	AWS managed	None	Provides access to AWS CloudForm
CloudFrontFullAccess	AWS managed	None	Provides full access to the CloudFront
AWSCloudHSMFullAccess	AWS managed	None	Provides full access to all CloudHSM
AWSCloudHSMReadOnlyAccess	AWS managed	None	Provides read only access to all Clo
ResourceGroupsTagEditorFullAccess	AWS managed	None	Provides full access to Resource Gr
ResourceGroupsTagEditorReadOnlyAccess	AWS managed	None	Provides access to use Resource G

Now we will create policy2. In that we have to give permission for ec2 and s3 be able to list resources only



The screenshot shows the 'Create policy' wizard, Step 1: Specify permissions. It lists actions for the EC2 service, including 'List' and 'Read' under 'Actions allowed'. The 'Resources' section is empty. A 'Visual' tab is selected.

Step 1: Specify permissions

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

EC2 (Allow 168 Actions)

Actions allowed

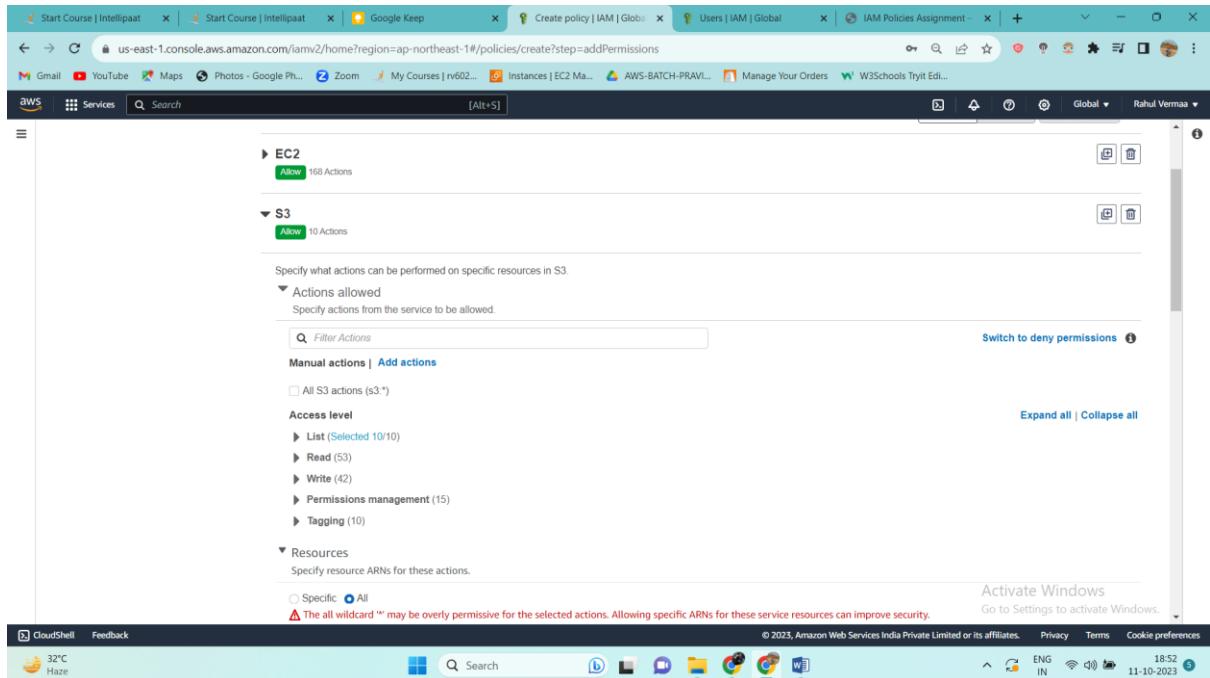
Specify actions from the service to be allowed.

Access level

Resources

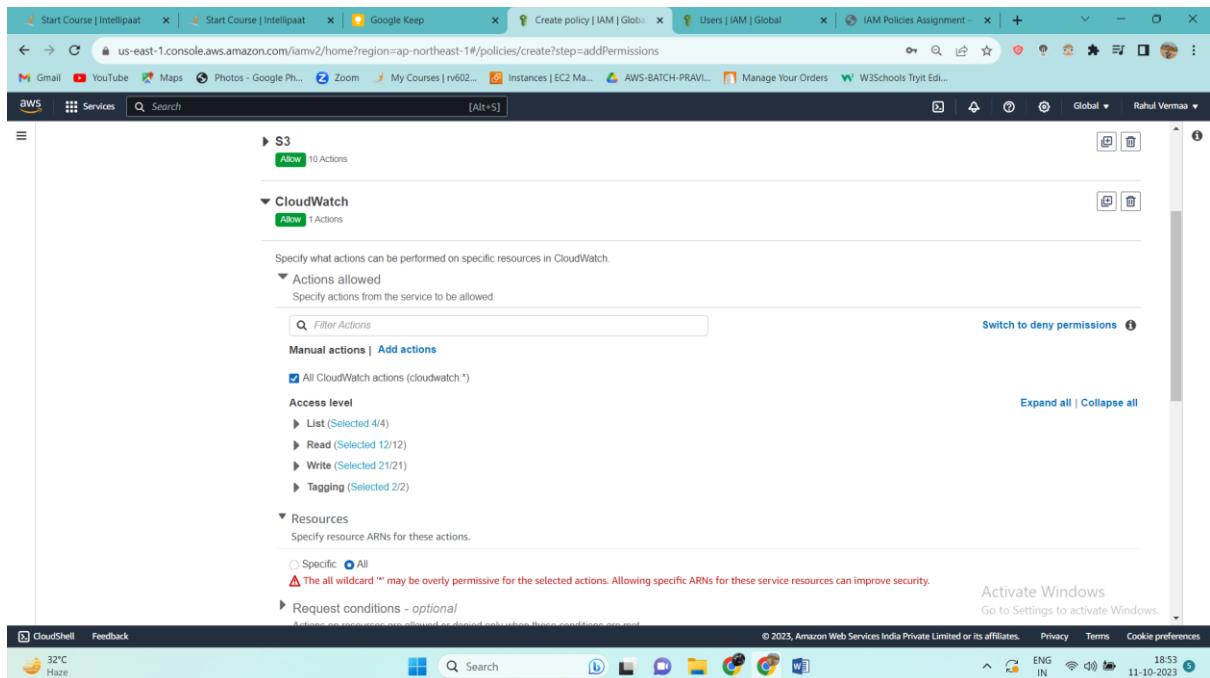
Specify resource ARNs for these actions.

This for s3 (list only)



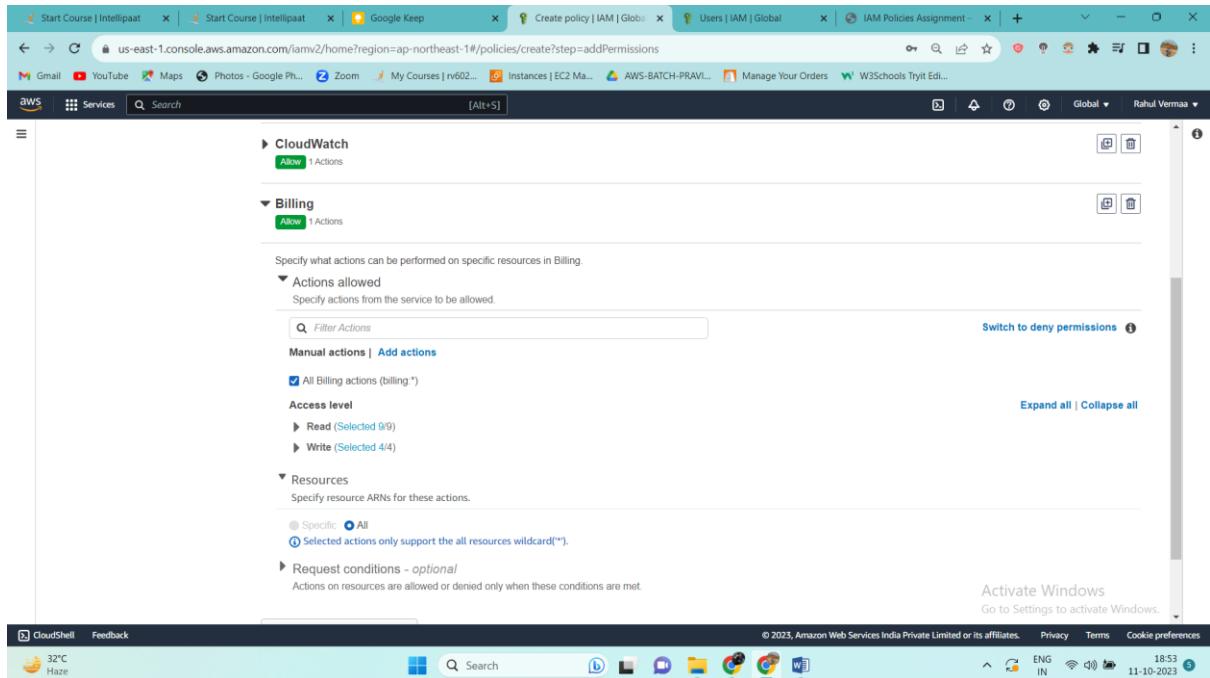
The screenshot shows the AWS IAM Policy creation interface. The policy is titled 'EC2' and has an 'Allow' section for S3 actions. Under 'Actions allowed', 'List' is selected. Under 'Access level', 'List' is also selected. The 'Resources' section is set to 'All'. A note at the bottom states: 'The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.'

Now for cloud watch all resources



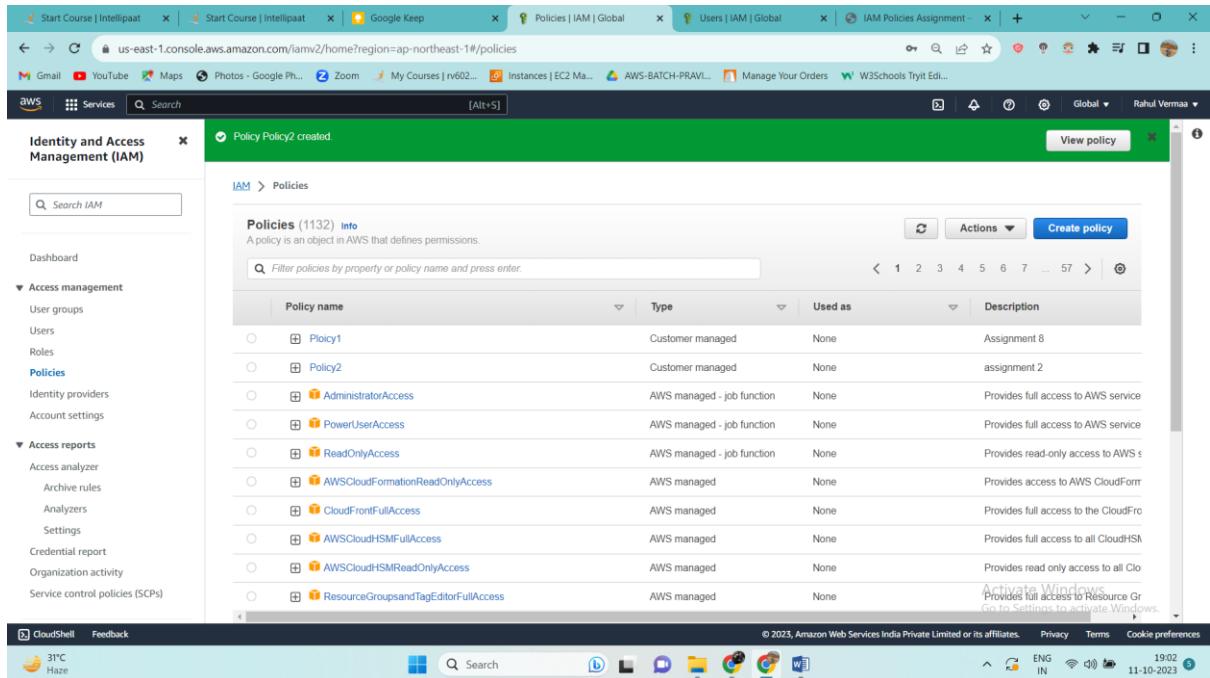
The screenshot shows the AWS IAM Policy creation interface. The policy is titled 'S3' and has an 'Allow' section for CloudWatch actions. Under 'Actions allowed', 'All CloudWatch actions (cloudwatch*)' is selected. Under 'Access level', 'List' is selected. The 'Resources' section is set to 'All'. A note at the bottom states: 'The all wildcard '*' may be overly permissive for the selected actions. Allowing specific ARNs for these service resources can improve security.'

And for billing also all resources



The screenshot shows the AWS IAM Policy creation interface for CloudWatch Billing. The 'Actions allowed' section is expanded, showing 'All Billing actions (billing*)' selected. The 'Access level' section shows 'Read' and 'Write' selected. The 'Resources' section has 'All' selected. The 'Request conditions - optional' section is collapsed.

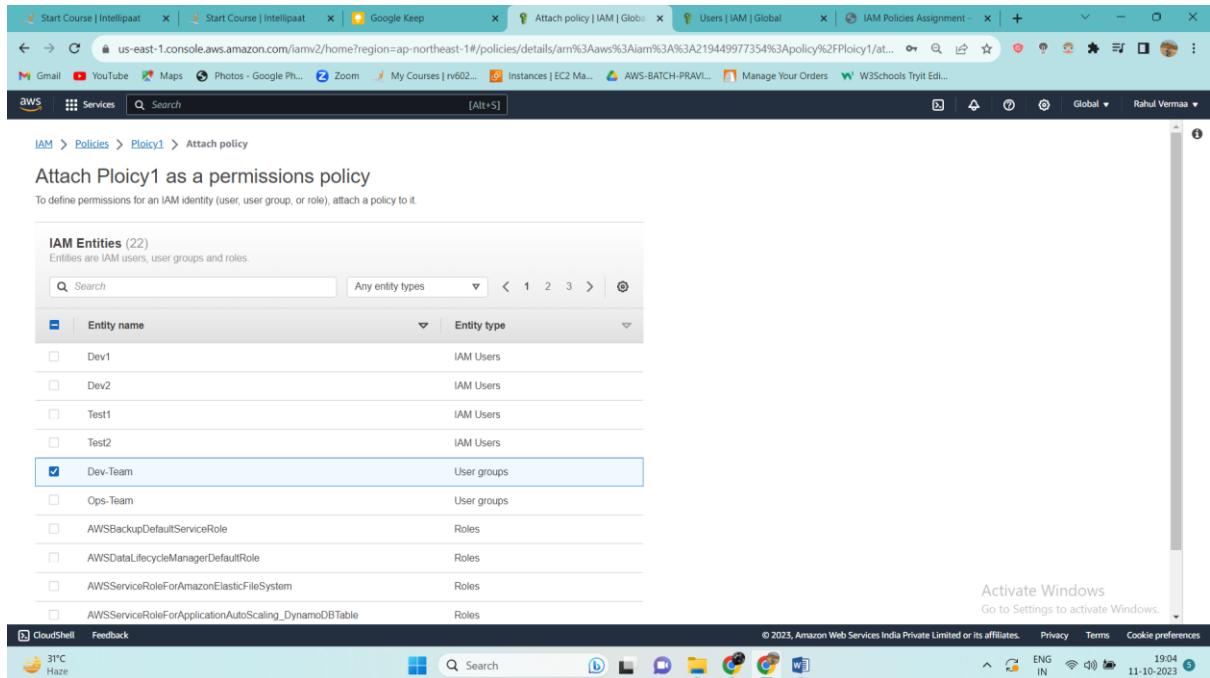
And our policy2 is also created



The screenshot shows the AWS IAM Policies list interface. A green banner at the top indicates 'Policy Policy2 created.' The table lists various policies, including Policy1, Policy2, and several AWS-managed policies like AdministratorAccess and PowerUserAccess.

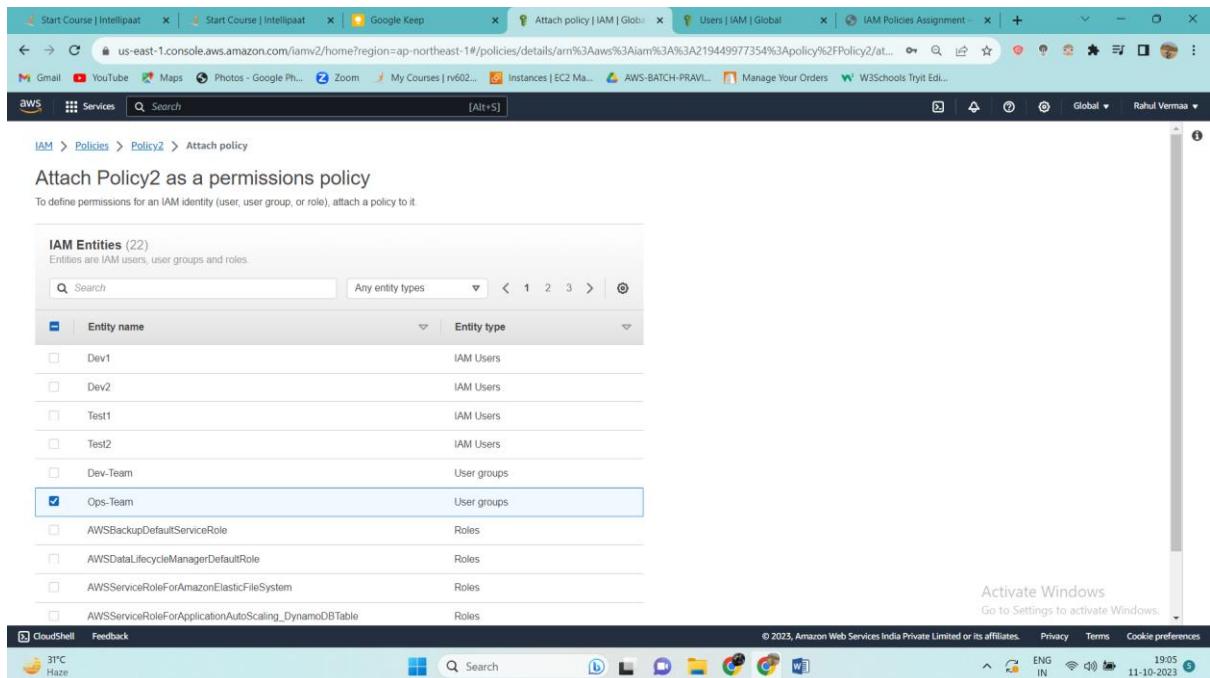
Policy name	Type	Used as	Description
Policy1	Customer managed	None	Assignment 8
Policy2	Customer managed	None	assignment 2
AdministratorAccess	AWS managed - job function	None	Provides full access to AWS service
PowerUserAccess	AWS managed - job function	None	Provides full access to AWS service
ReadOnlyAccess	AWS managed - job function	None	Provides read-only access to AWS service
AWSCloudFormationReadOnlyAccess	AWS managed	None	Provides access to AWS CloudFormation
CloudFrontFullAccess	AWS managed	None	Provides full access to the CloudFront service
AWSCloudHSMFullAccess	AWS managed	None	Provides full access to all CloudHSM services
AWSCloudHSMReadOnlyAccess	AWS managed	None	Provides read only access to all CloudHSM services
ResourceGroupsandTagEditorFullAccess	AWS managed	None	Provides full access to Resource Groups and Tag Editor

Now let's add policy1 to Dev-Team



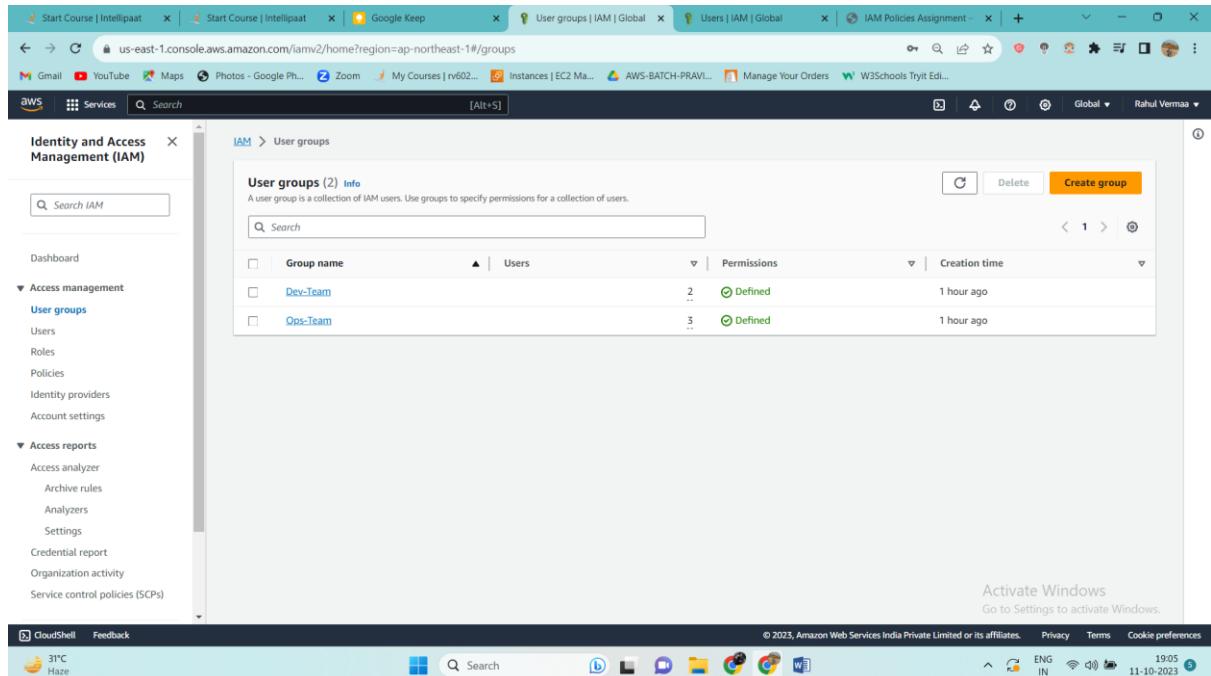
The screenshot shows the AWS IAM console with the URL us-east-1.console.aws.amazon.com/iamv2/home?region=ap-northeast-1#/policies/details/arm%3Aaws%3Aiam%3A219449977354%3Apolicy%2Fpolicy1/at.... The page title is "Attach Policy1 as a permissions policy". The "IAM Entities (22)" section lists entities: Dev1, Dev2, Test1, Test2, Dev-Team (selected), Ops-Team, AWSBackupDefaultServiceRole, AWSDataLifecycleManagerDefaultRole, AWSServiceRoleForAmazonElasticFileSystem, and AWSServiceRoleForApplicationAutoScaling_DynamoDBTable. The "Entity type" dropdown is set to "User groups". The status bar at the bottom shows "CloudShell Feedback" and the date "11-10-2023".

And now let's attach policy2 to Ops-Team



The screenshot shows the AWS IAM console with the URL us-east-1.console.aws.amazon.com/iamv2/home?region=ap-northeast-1#/policies/details/arm%3Aaws%3Aiam%3A219449977354%3Apolicy%2Fpolicy2/at.... The page title is "Attach Policy2 as a permissions policy". The "IAM Entities (22)" section lists entities: Dev1, Dev2, Test1, Test2, Dev-Team, and Ops-Team (selected). The "Entity type" dropdown is set to "User groups". The status bar at the bottom shows "CloudShell Feedback" and the date "11-10-2023".

And now in both groups policies are defined

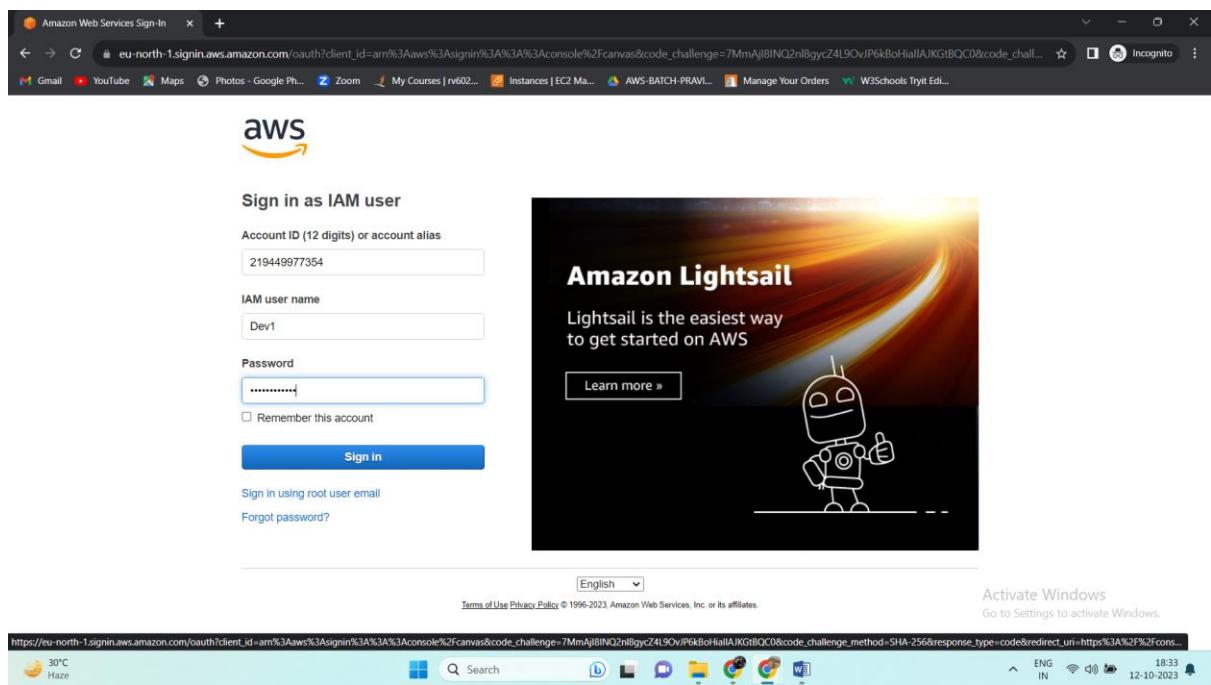


The screenshot shows the AWS IAM User Groups page. The left sidebar is collapsed. The main content area shows a table of user groups:

Group name	Users	Permissions	Creation time
Dev-Team	2	Defined	1 hour ago
Ops-Team	3	Defined	1 hour ago

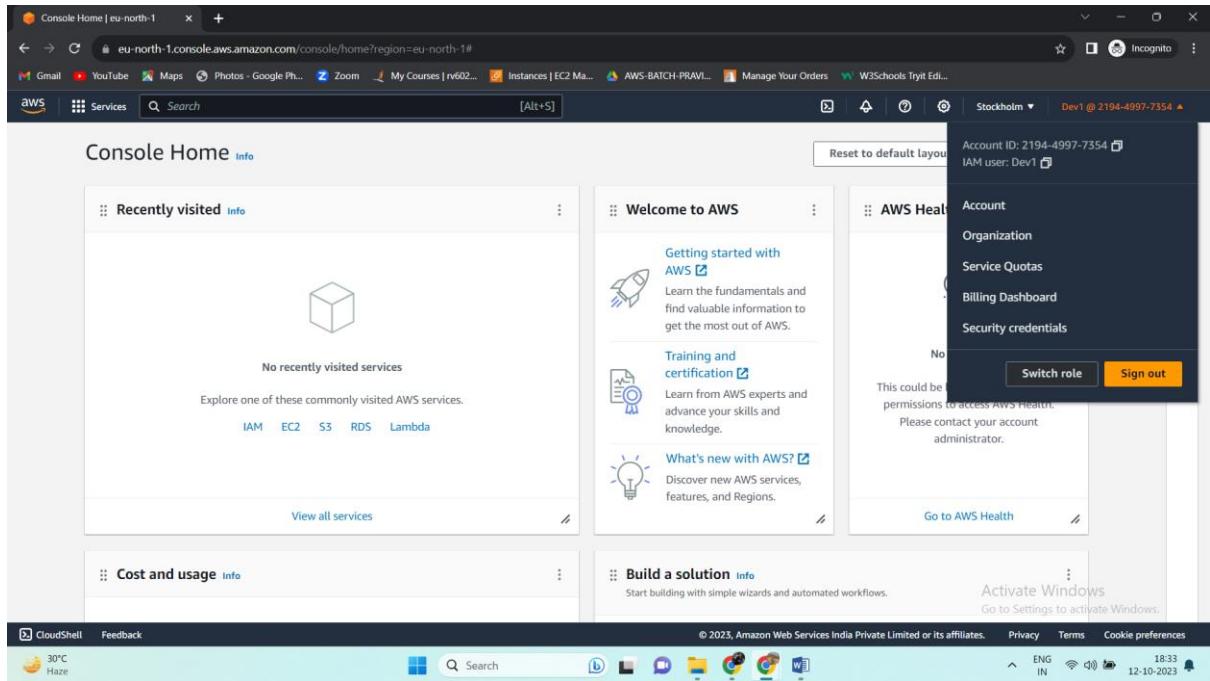
At the bottom right of the main area, there is a message: "Activate Windows Go to Settings to activate Windows." The bottom of the screen shows the Windows taskbar with various icons and a system tray showing the date and time as 11-10-2023.

Now let's test our policies are working or not so we will login to Dev1 from Dev-teams and Ops-Team



The screenshot shows the AWS Sign In page for the user Dev1. The left side has a "Sign in as IAM user" form with fields for Account ID, IAM user name, Password, and a Remember this account checkbox. Below the form are links for "Sign in using root user email" and "Forgot password?". The right side features a large advertisement for "Amazon Lightsail" with the tagline "Lightsail is the easiest way to get started on AWS" and a "Learn more" button. The bottom of the screen shows the Windows taskbar with various icons and a system tray showing the date and time as 12-10-2023.

And we have logged in to Dev1 user



Now Dev1 user should be able to do following things

Policy1 (Dev-team) :

- a. Access S3 completely
- b. Only create EC2 instances
- c. Full access to RDS

Policy2 (Ops-team):

- a. Access CloudWatch and billing completely
- b. Can only list EC2 and s3 resources.

A) S3 full access

The screenshot shows the AWS S3 console with the following details:

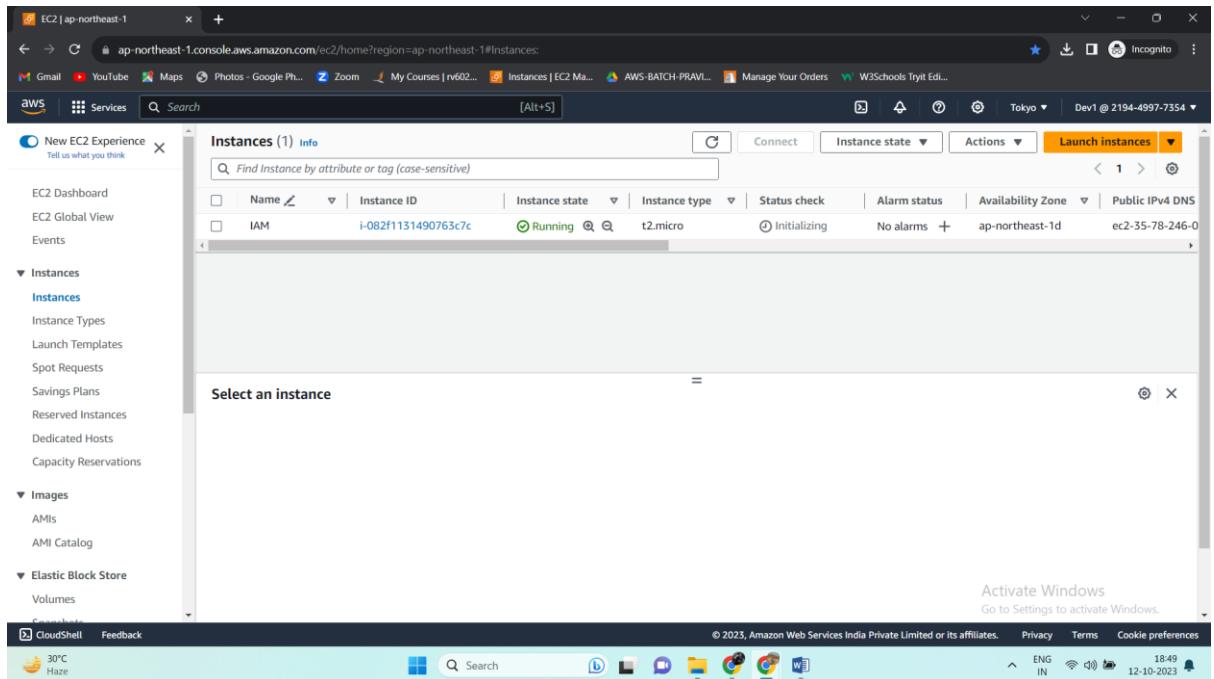
- Bucket:** cf-templates-ru6urdery9gd-ap-northeast-1
- Objects:** 2
 - 20232845QM-designer/ (Folder)
 - 2023284qTw-new:templatebs78j8ply7 (File, 161.0 B, Standard storage class)

B) Only create ec2 instance

The screenshot shows the AWS EC2 'Launch an instance' wizard with the following configuration:

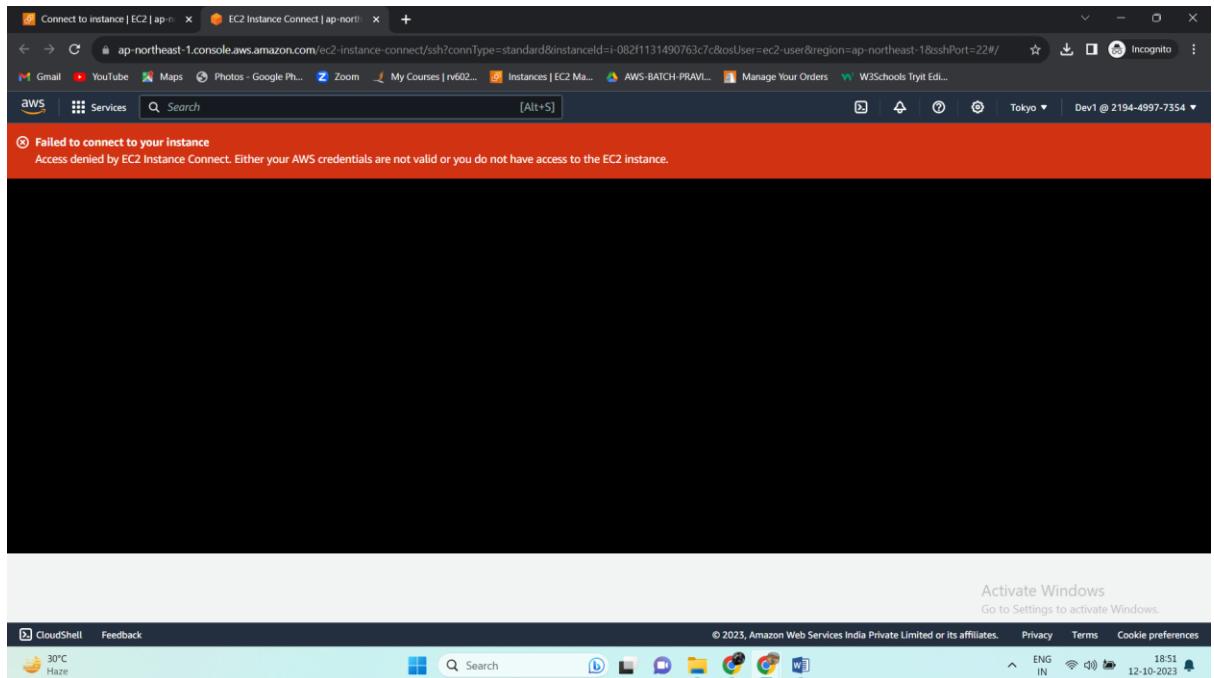
- Name and tags:** IAM
- Summary:**
 - Number of instances: 1
 - Software Image (AMI): Amazon Linux 2023 AMI 2023.2.2... (read more)
 - Virtual server type (instance type): t3.micro
 - Firewall (security group): New security group
 - Storage (volumes): 1 volume(s) - 8 GiB

Instance is created



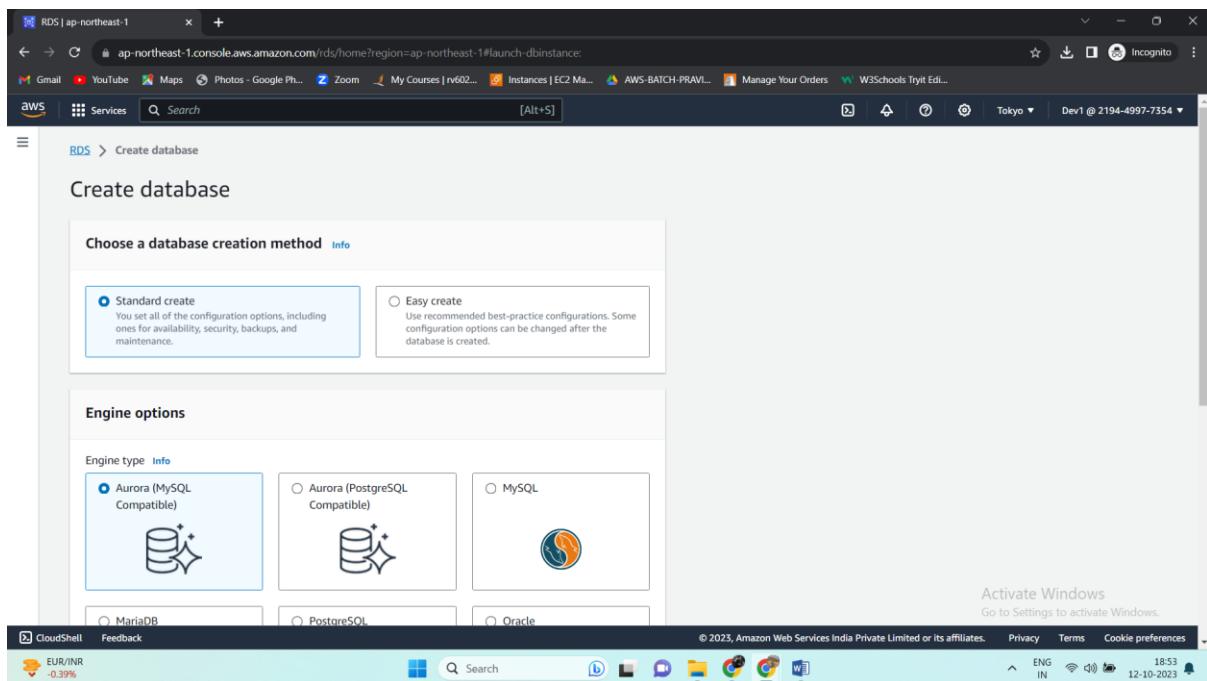
The screenshot shows the AWS EC2 Instances page. On the left, a sidebar menu is open with the 'Instances' section selected. The main table displays one instance: 'IAM' (Instance ID: i-082f1131490763c7c, State: Running, Type: t2.micro). The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Below the table, a modal window titled 'Select an instance' is open, showing the same single instance entry. The browser's address bar shows the URL: ap-northeast-1.console.aws.amazon.com/ec2/home?region=ap-northeast-1#Instances. The status bar at the bottom indicates the date and time as 12-10-2023.

Not able to connect

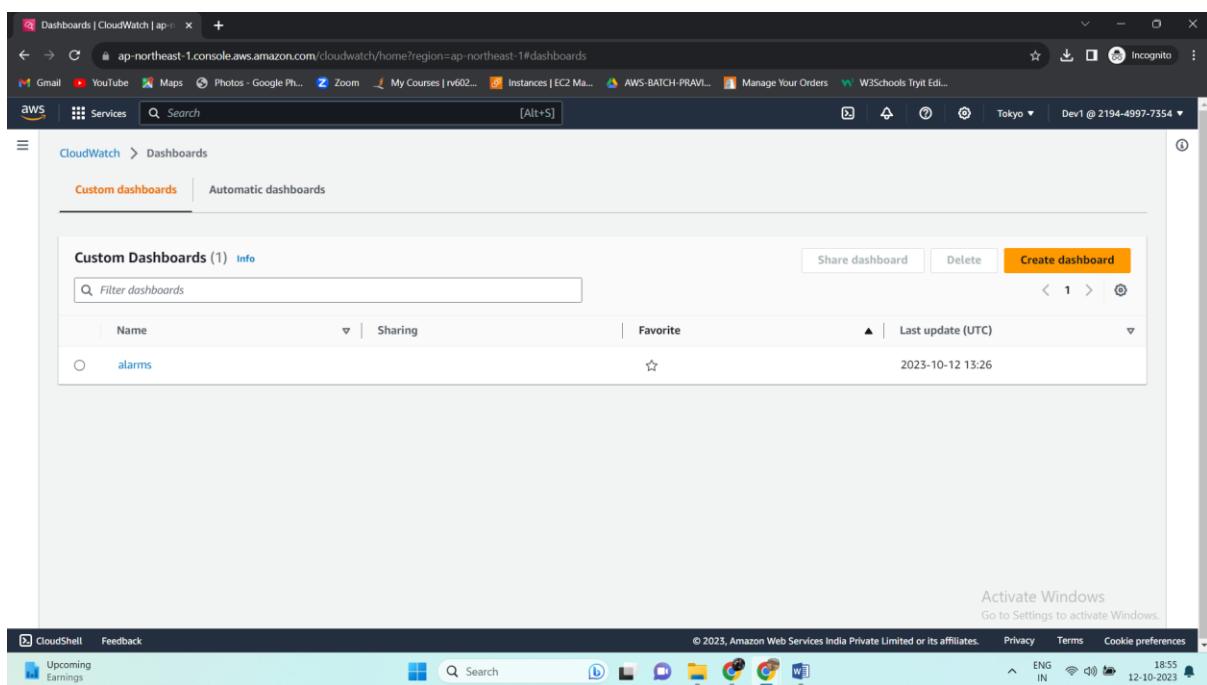


The screenshot shows the AWS EC2 Instance Connect page. The browser address bar shows the URL: ap-northeast-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-082f1131490763c7c&osUser=ec2-user®ion=ap-northeast-1&sshPort=22#. The main content area displays an error message: 'Failed to connect to your instance. Access denied by EC2 Instance Connect. Either your AWS credentials are not valid or you do not have access to the EC2 instance.' The status bar at the bottom indicates the date and time as 12-10-2023.

C) Full access to RDS



Now cloud watch



So this is how IAM policies works. **DONE!**