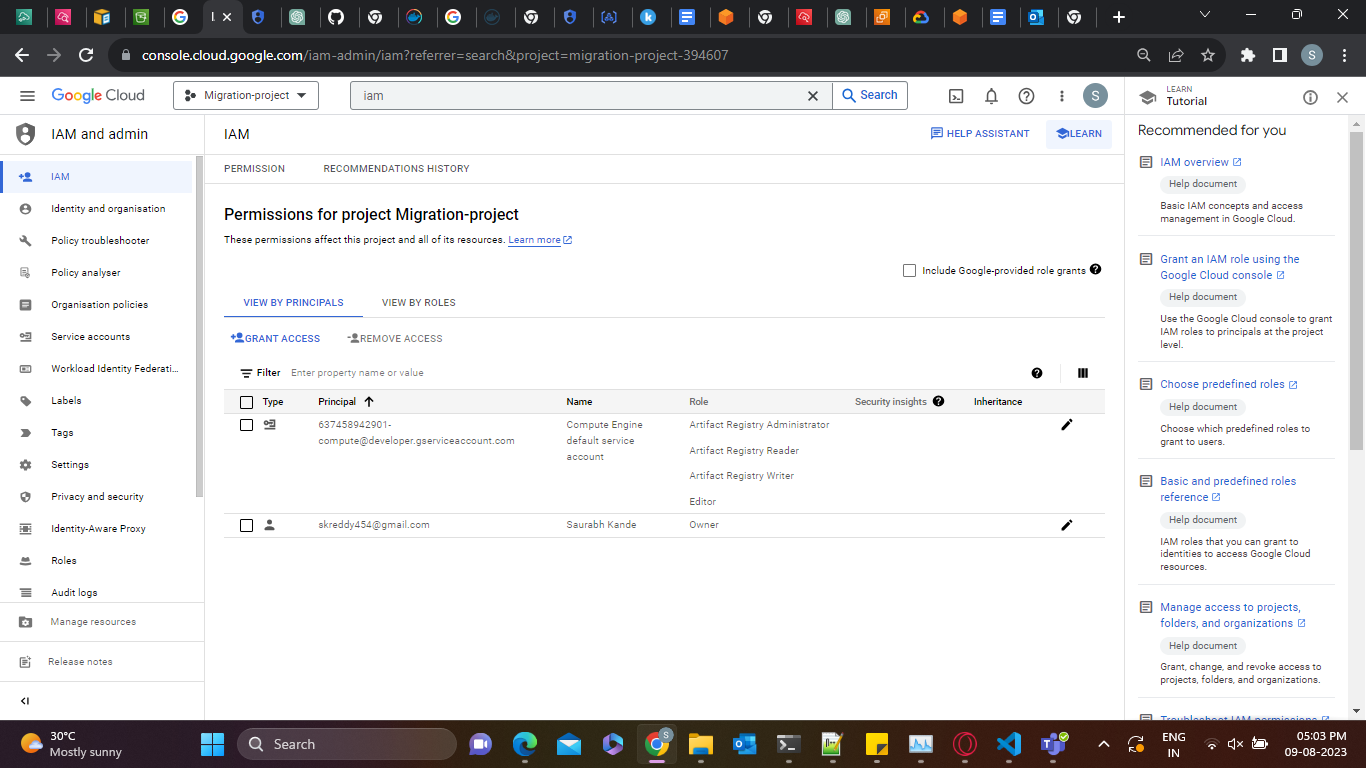
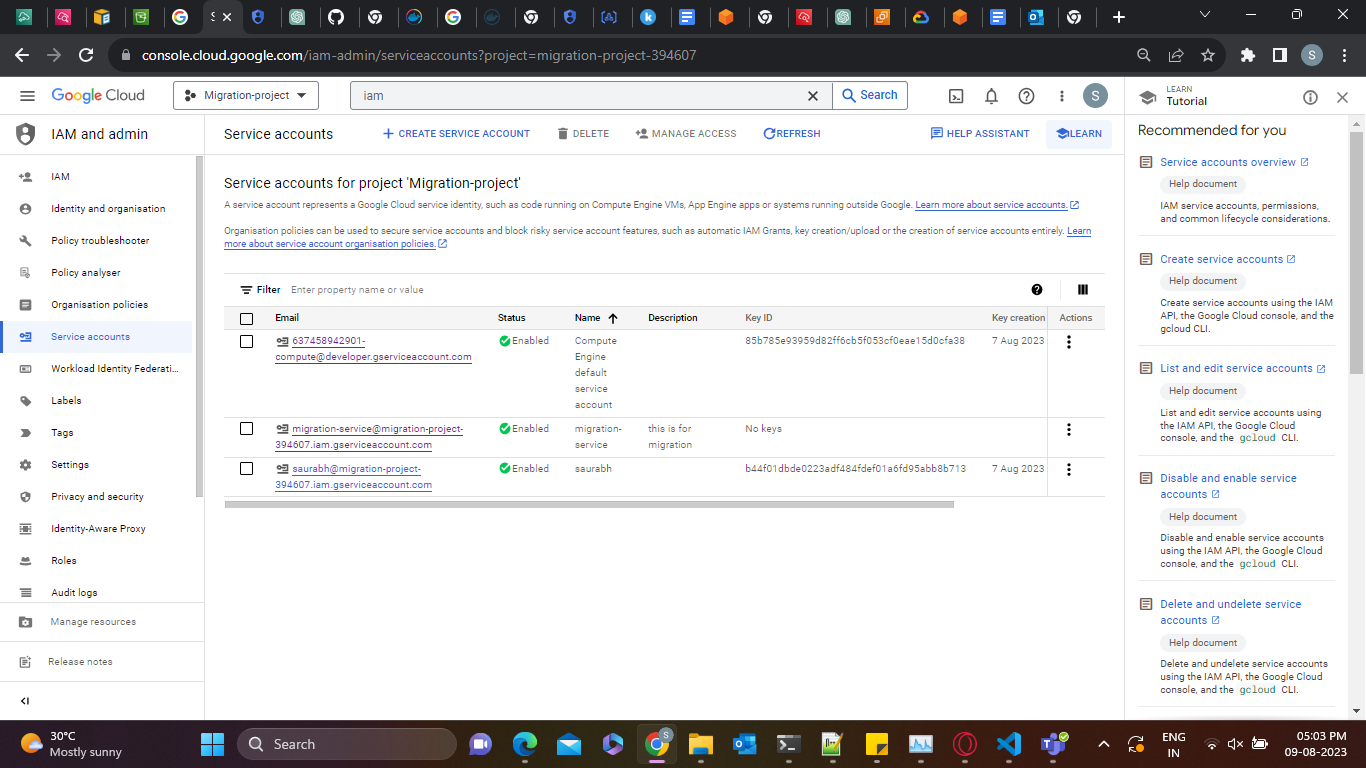
**AWS Datasync Transfer Data from GCS to S3**

* Go to the GCP and create a new service account go to the search bar and type IAM and go to the IAM Dashboard



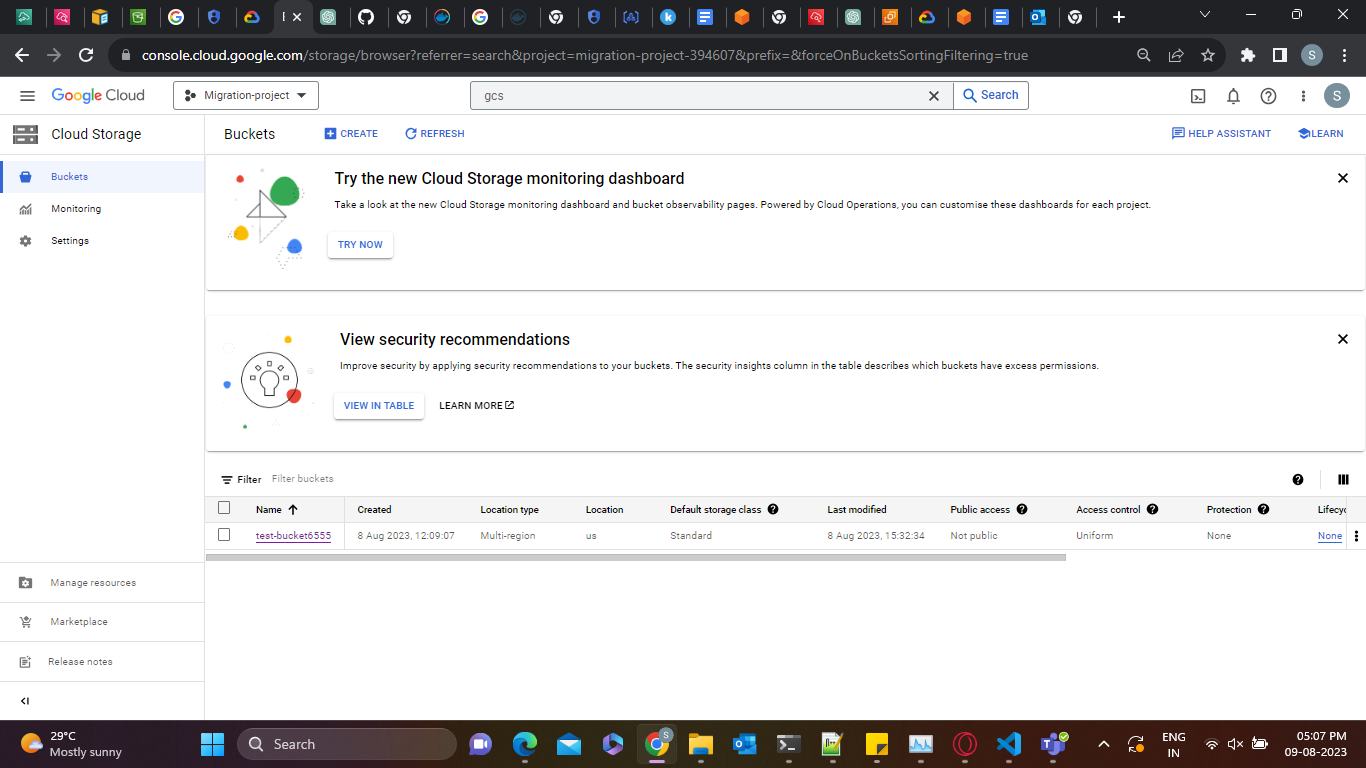
* On the left side, you can see the service account click on it and create a service account



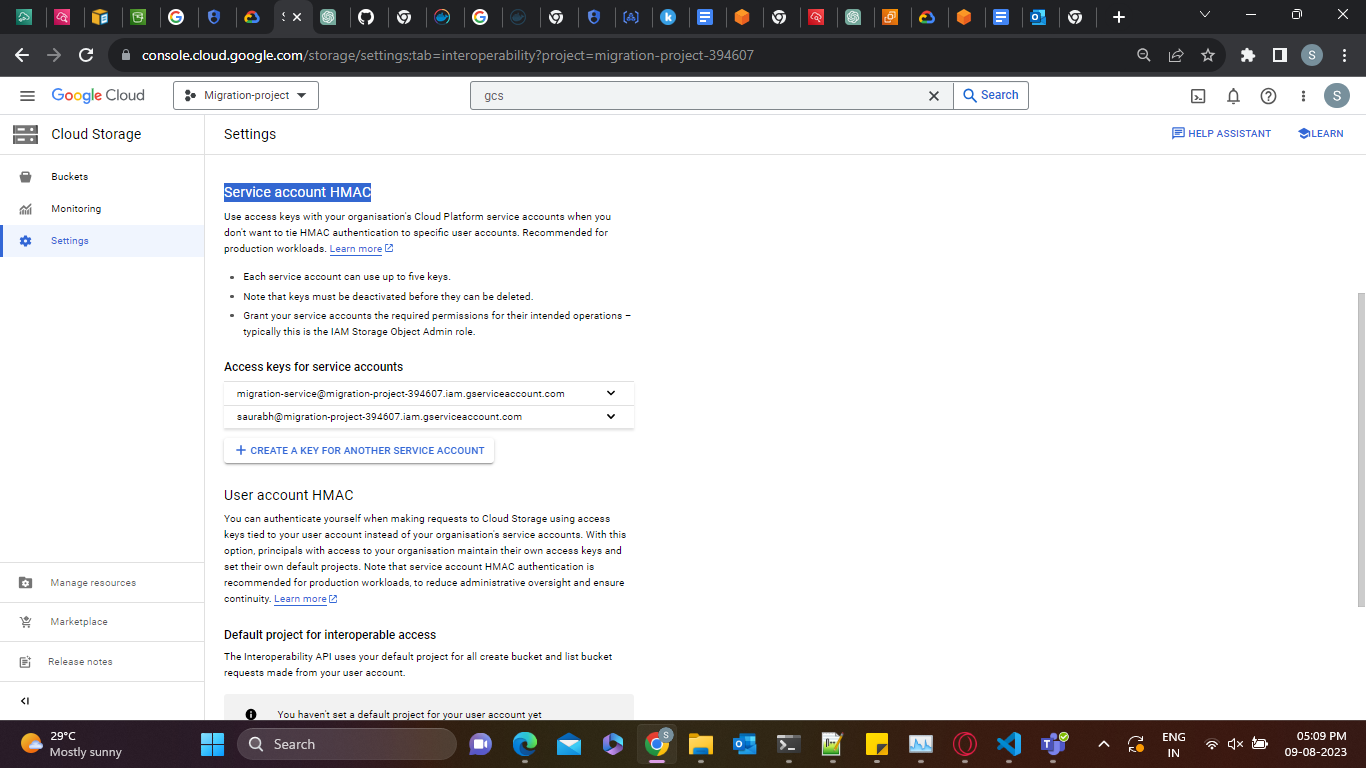
* When creating a service account give a name and unique id to identify the service account and click on create



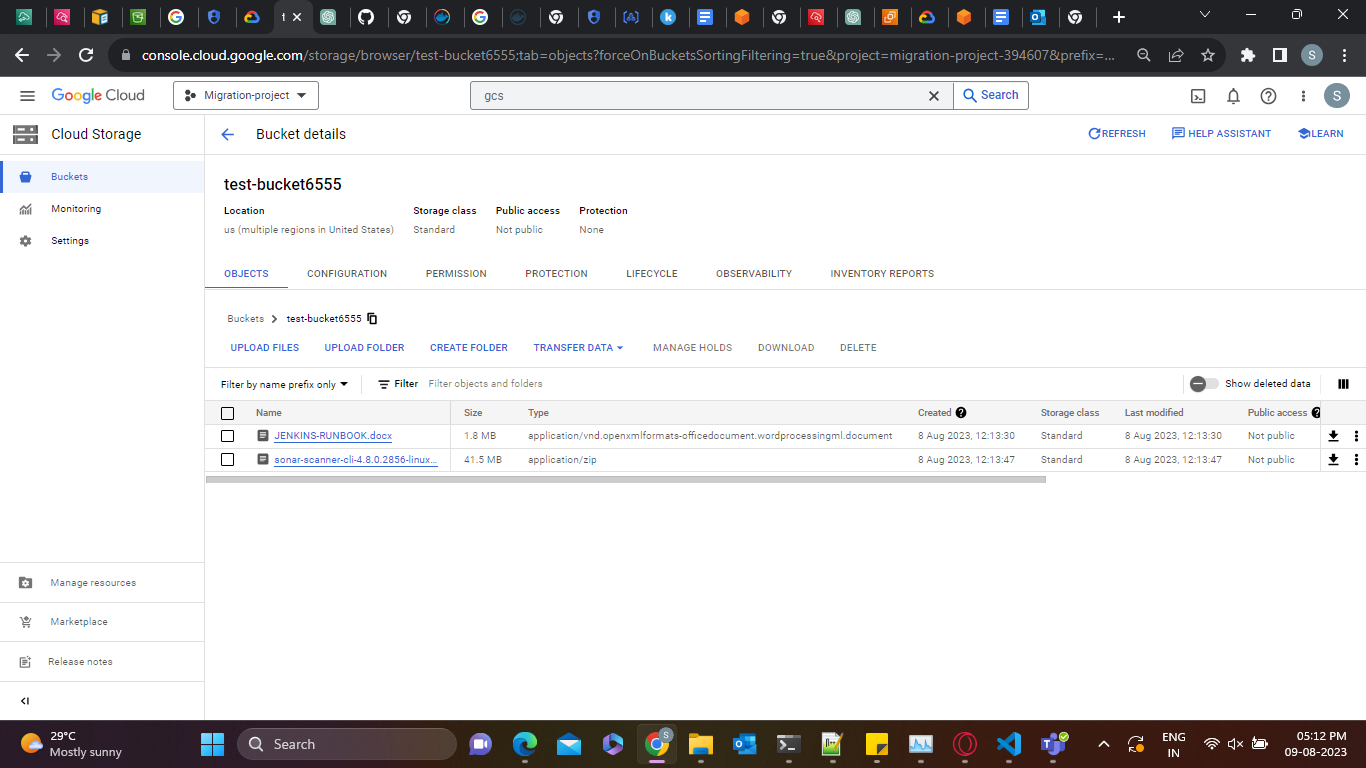
* keep the rest as default and click continue and create a service account
* After you have created the service account go to the Cloud Storage service



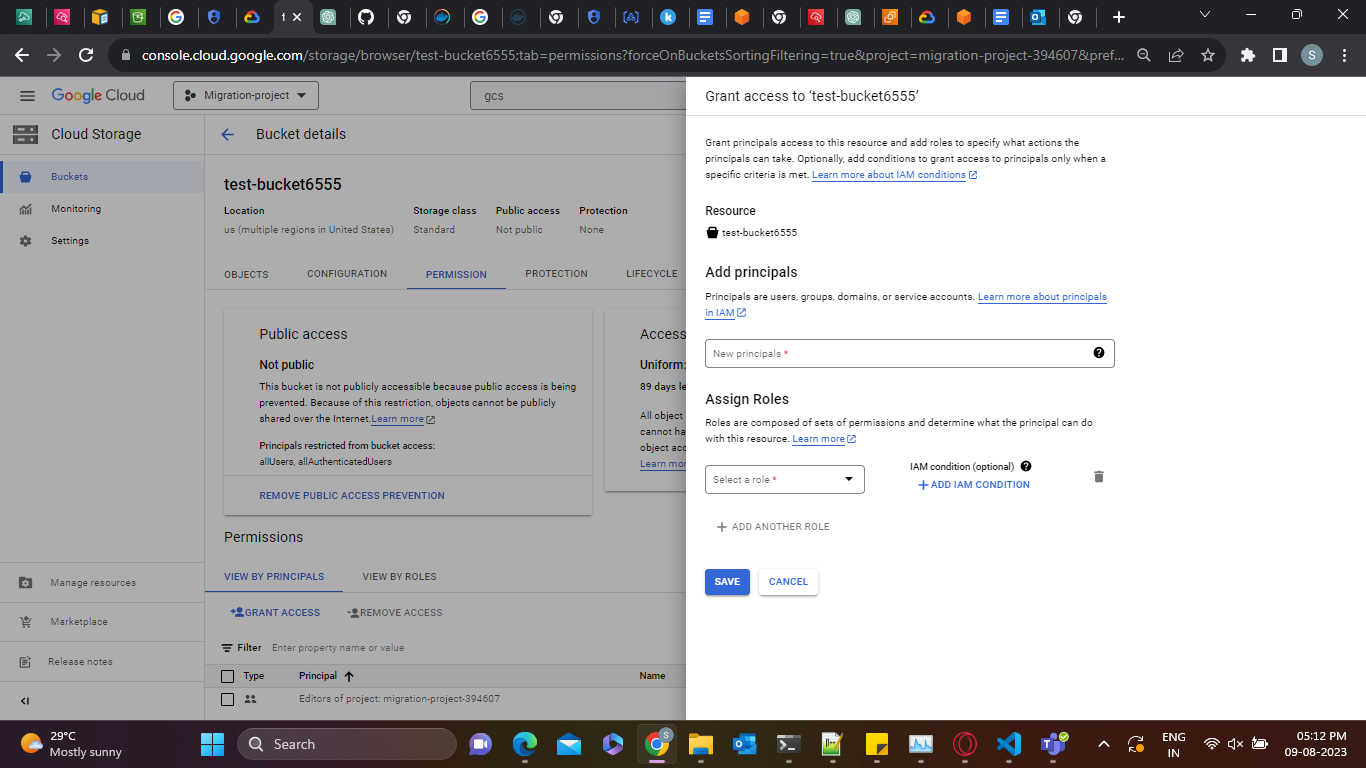
* Now go to settings > INTEROPERABILITY and below there is an option for Service account HMAC
* Click on Create key for another service account



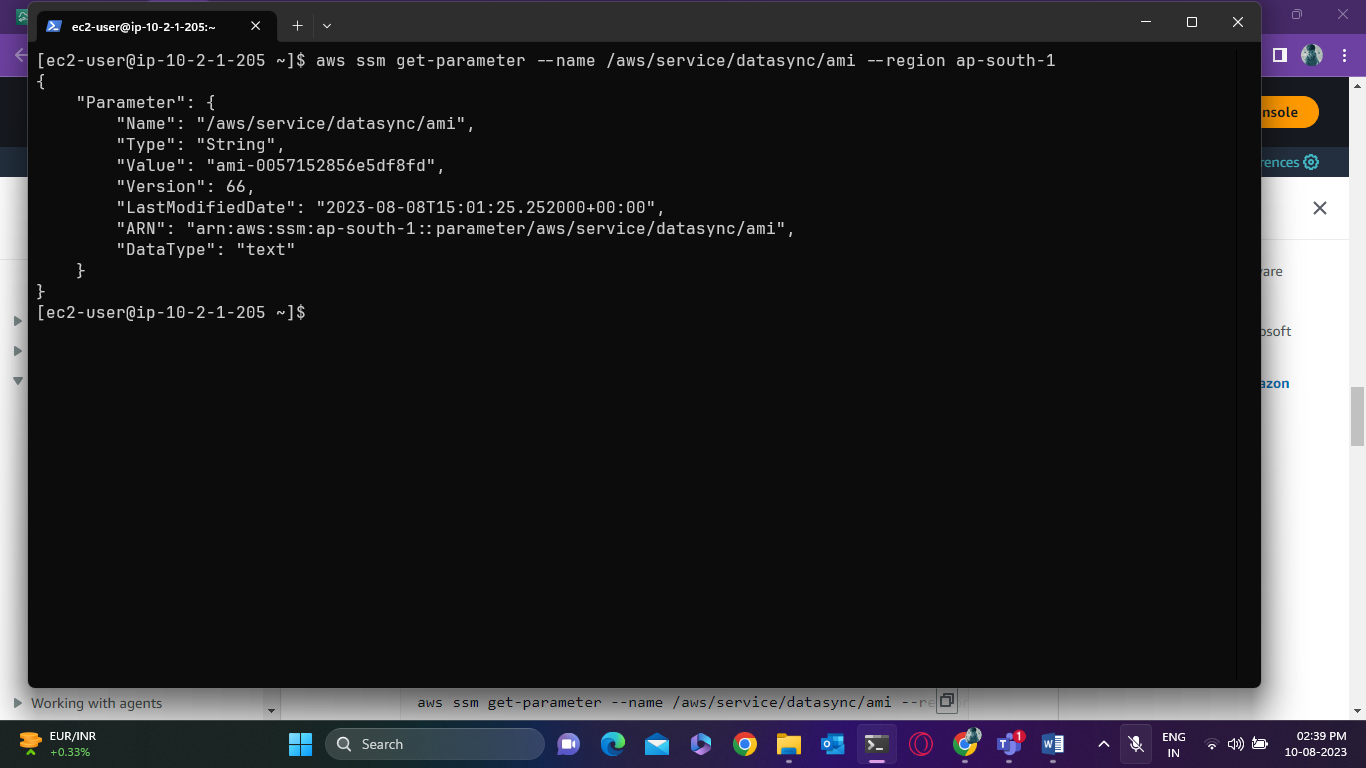
* Select the service account you have just created and click on create key after the key is created you will get an access key and a secret key save this key securely somewhere and it is required to obtain permission while migrating the data
* After you have created the HMAC key go back to the bucket and select which you want to migrate



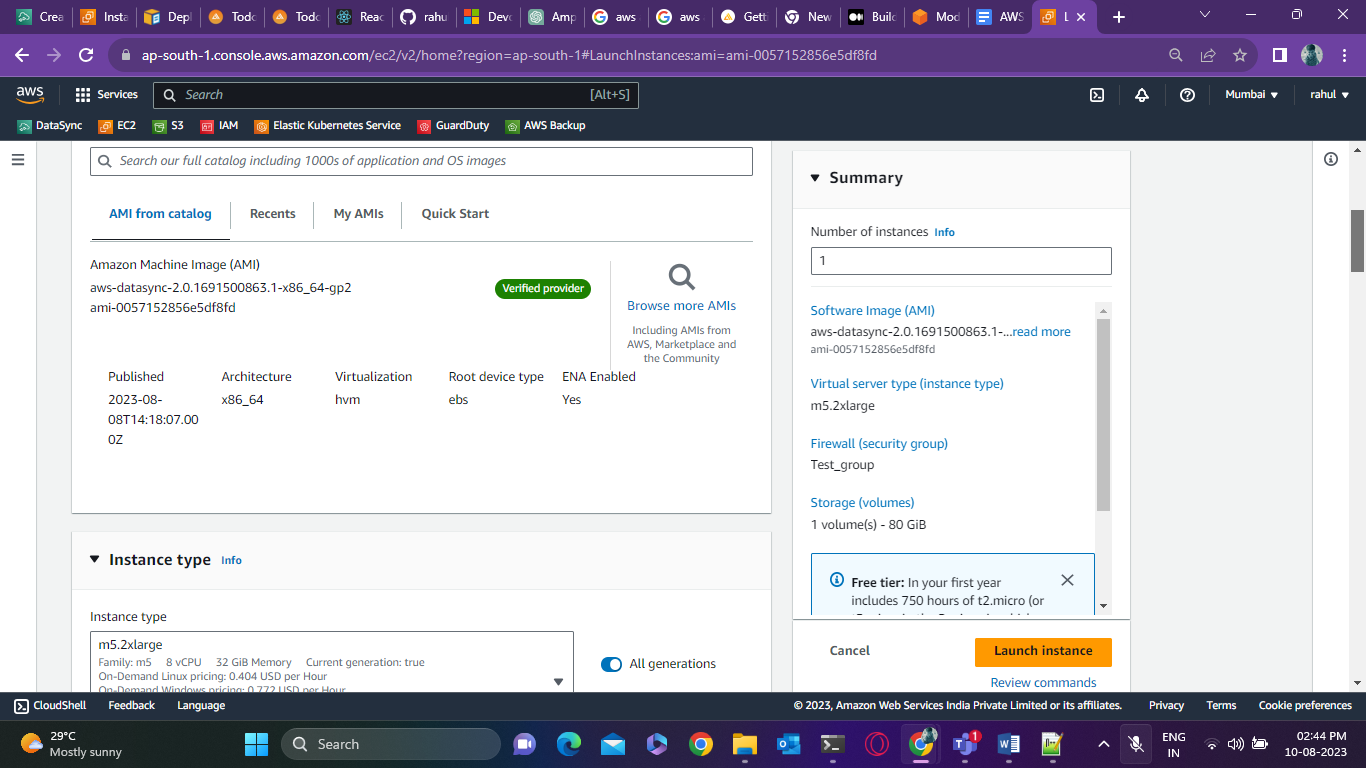
* Now go to the permission tab and in the permission below click on grant access



* In the add principal session give the service account name in the principal box which you have created the HMAC key for and in the assign role you have to select this policy **Storage Object Viewer**  and click on save
* Now we set up the data sync agent and give a proper role for it to access the s3 bucket
* Go to the data sync service tab and go to an agent click on Create agent
* To deploy the agent you need to find the AMI id of the Datasync agent ami to get that you need to go to a instance in a region you want to deploy the agent and use this command to get the ami id
* aws ssm get-parameter --name /aws/service/datasync/ami --region ap-south-1



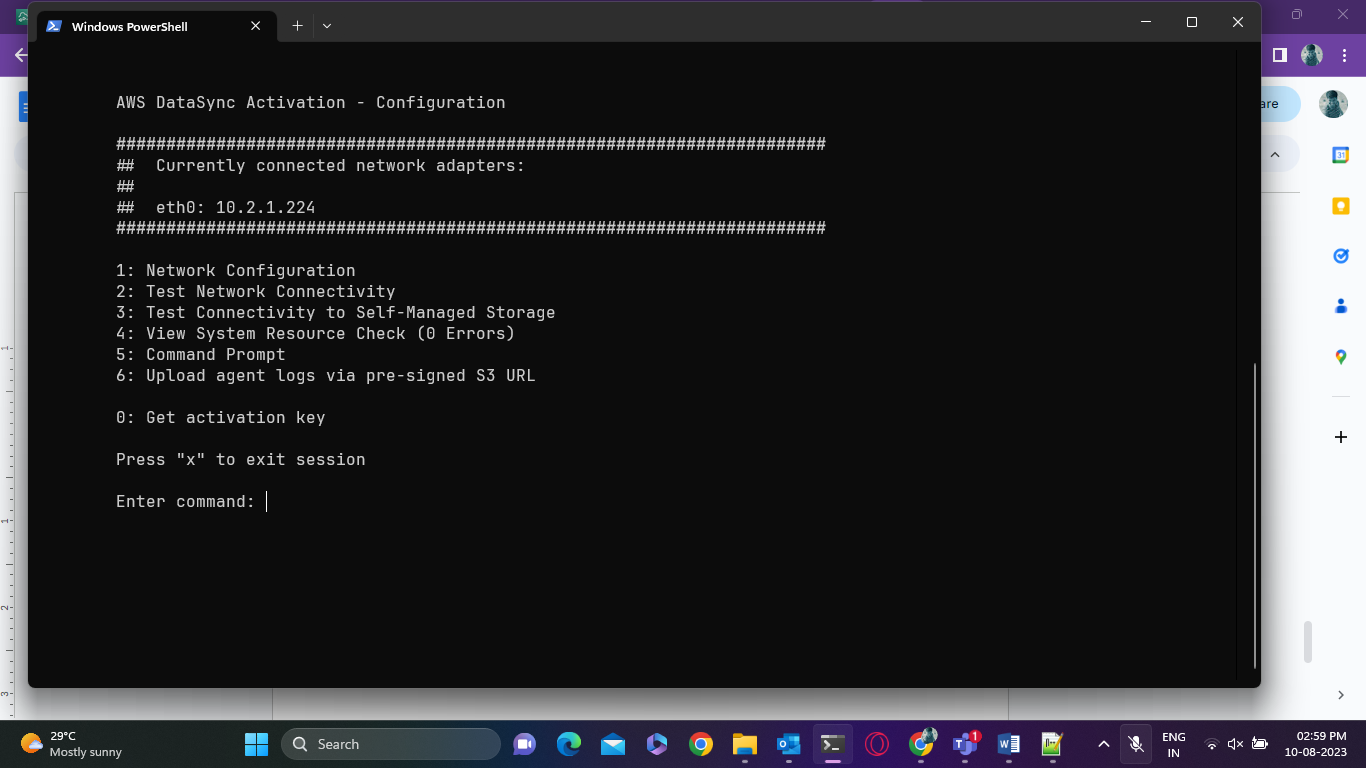
* Copy the AMI id from the terminal now from the aws console create an instance and copy this URL with the region you given and ami id of the data sync agent
* <https://console.aws.amazon.com/ec2/v2/home?region=source-file-system-region#LaunchInstanceWizard:ami=ami-id>



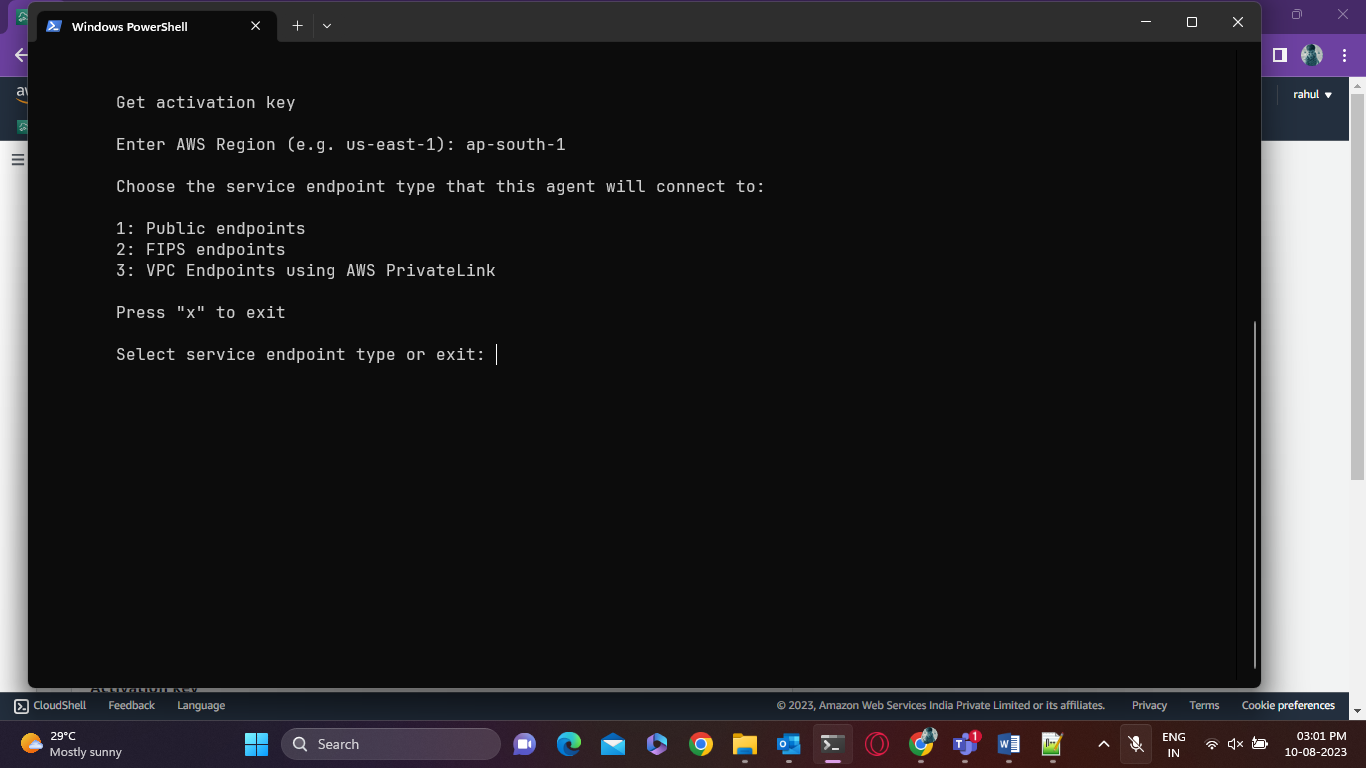
* Select the appropriate instances type and launch the instances if you have a lot of transfers to be done it is recommended to use m5.2xlarge
* Use this command as an example change the pem key and IP address of the instances login as admin

**ssh -i "istio-serv.pem" -oKexAlgorithms=+diffie-hellman-group14-sha1** [**admin@ec2-13-126-17-231.ap-south-1.compute.amazonaws.com**](mailto:admin@ec2-13-126-17-231.ap-south-1.compute.amazonaws.com)

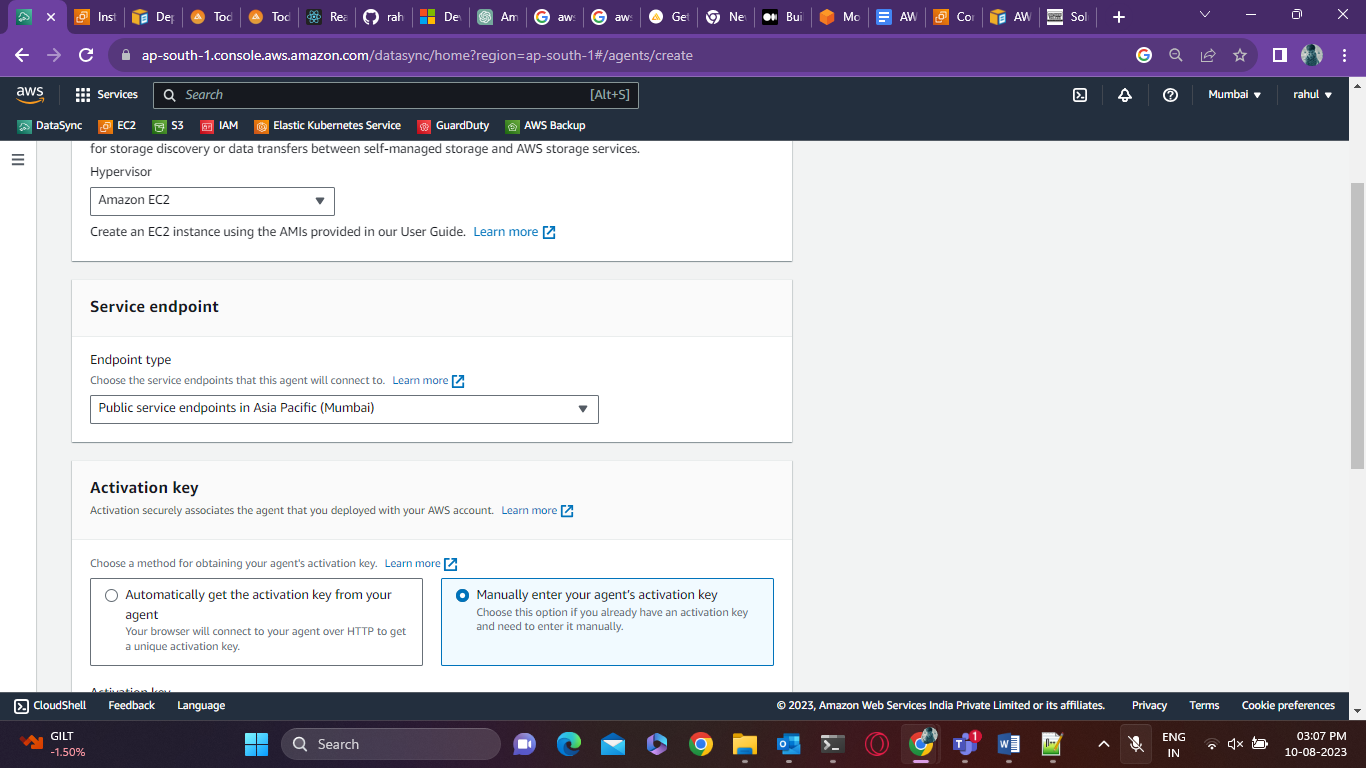
* After login select 0 to get the activation key

****

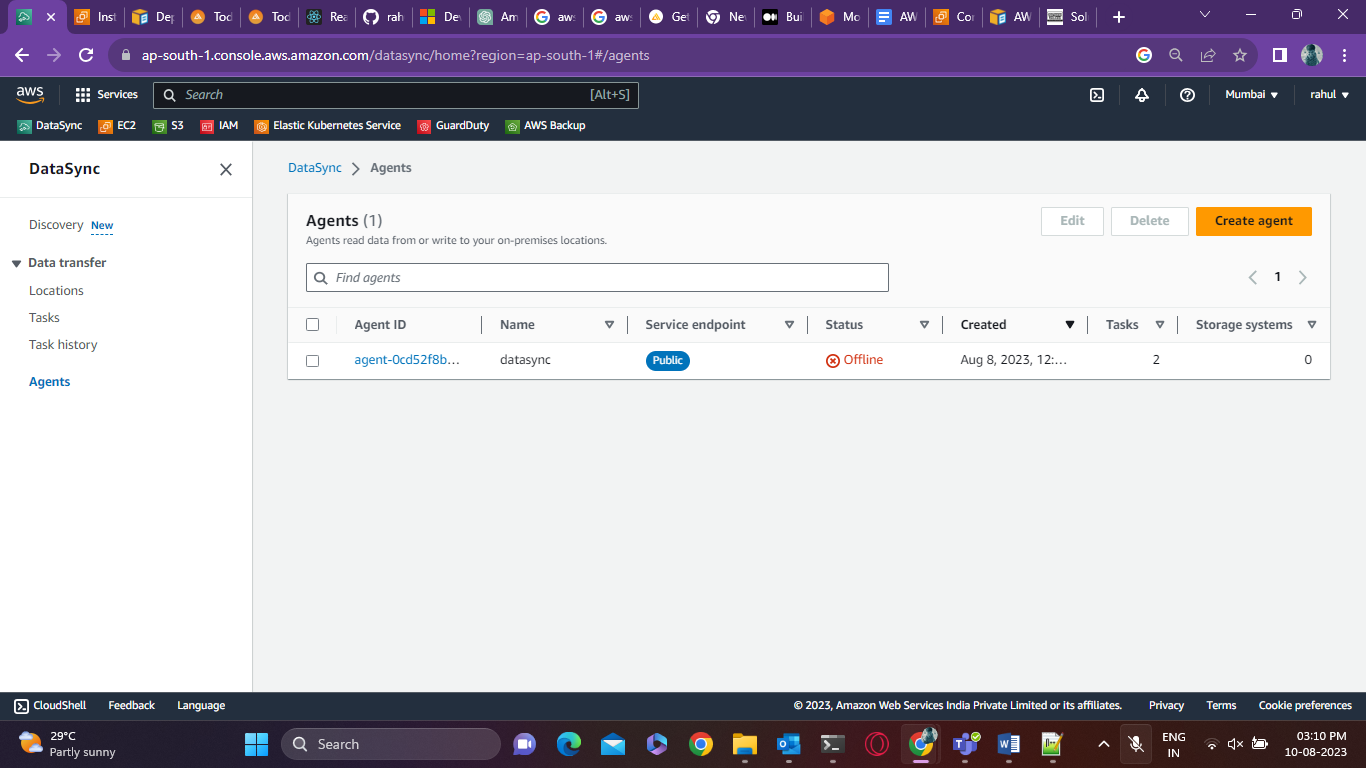
* Select the region and enter now select public endpoint as for this demo we are using the public endpoint
* After that, you will get an activation code copy this action code and exit

****

* Now go back to the data sync agent and create an agent choose where to deploy the argent we have taken ec2 instances and select the endpoint type we are migrating using the public endpoint in this demo



* Click on Enter the activation key manually and give the key which you have got from the data sync agent and give the agent a name now click on Create



* Go to locations and create a location select the type as object storage select the agent you have created
* Give server name as: **storage.googleapis.com**
* Bucket name of the GCS and folder which you want to copy if you want everything to be transferred just add “/”
* Below there is an option for authentication for the access key and secret key we need to add the HMAC key which you have taken from the GCP service account and add it here
* After you have added it click on Create a location
* We also need to create a location for the destination but before that, we need to create an IAM role for the data sync agent to be able to access our s3 bucket to write the files into it
* Go to IAM and create a new policy

{

"Version": "2012-10-17",

"Statement": [

{

"Action": [

"s3:GetBucketLocation",

"s3:ListBucket",

"s3:ListBucketMultipartUploads"

],

"Effect": "Allow",

"Resource": "**YourS3BucketArn**"

},

{

"Action": [

"s3:AbortMultipartUpload",

"s3:DeleteObject",

"s3:GetObject",

"s3:ListMultipartUploadParts",

"s3:GetObjectTagging",

"s3:PutObjectTagging",

"s3:PutObject"

],

"Effect": "Allow",

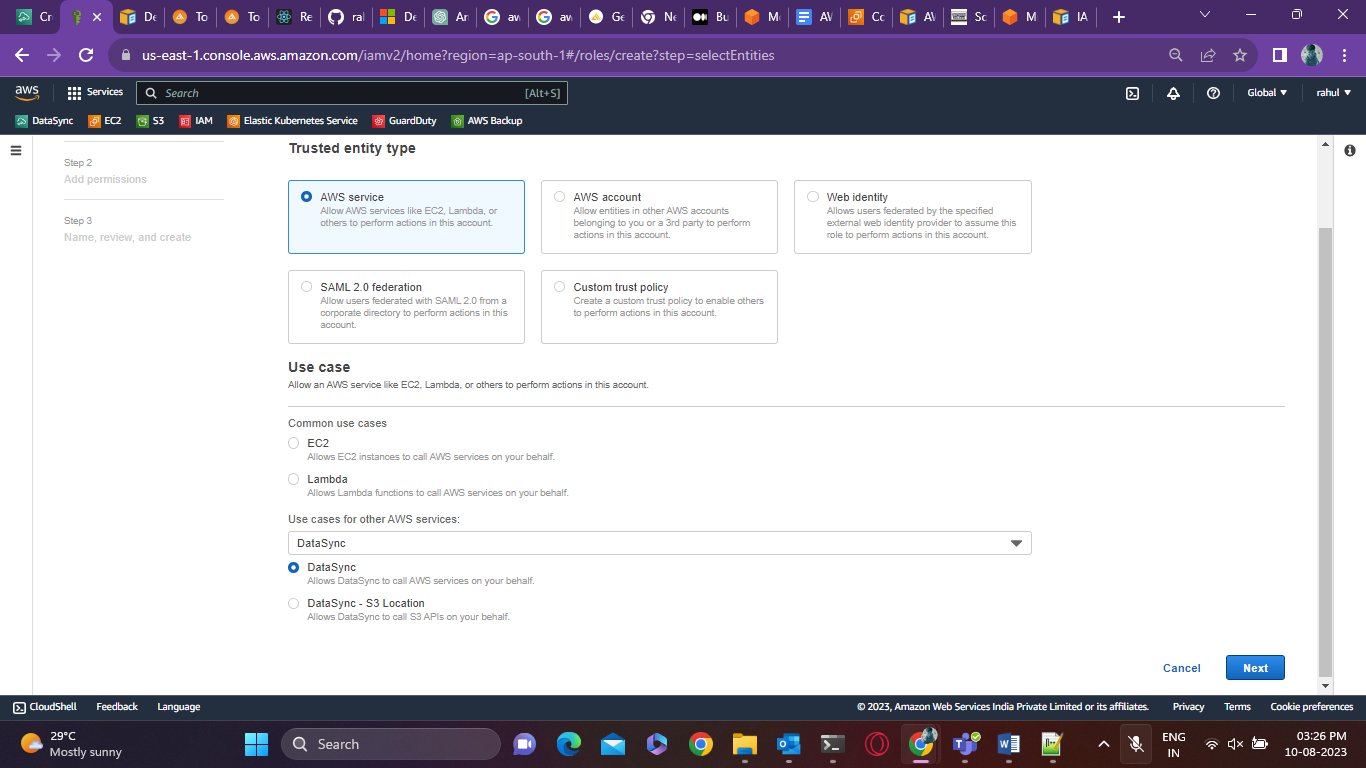
"Resource": "**YourS3BucketArn/\***"

}

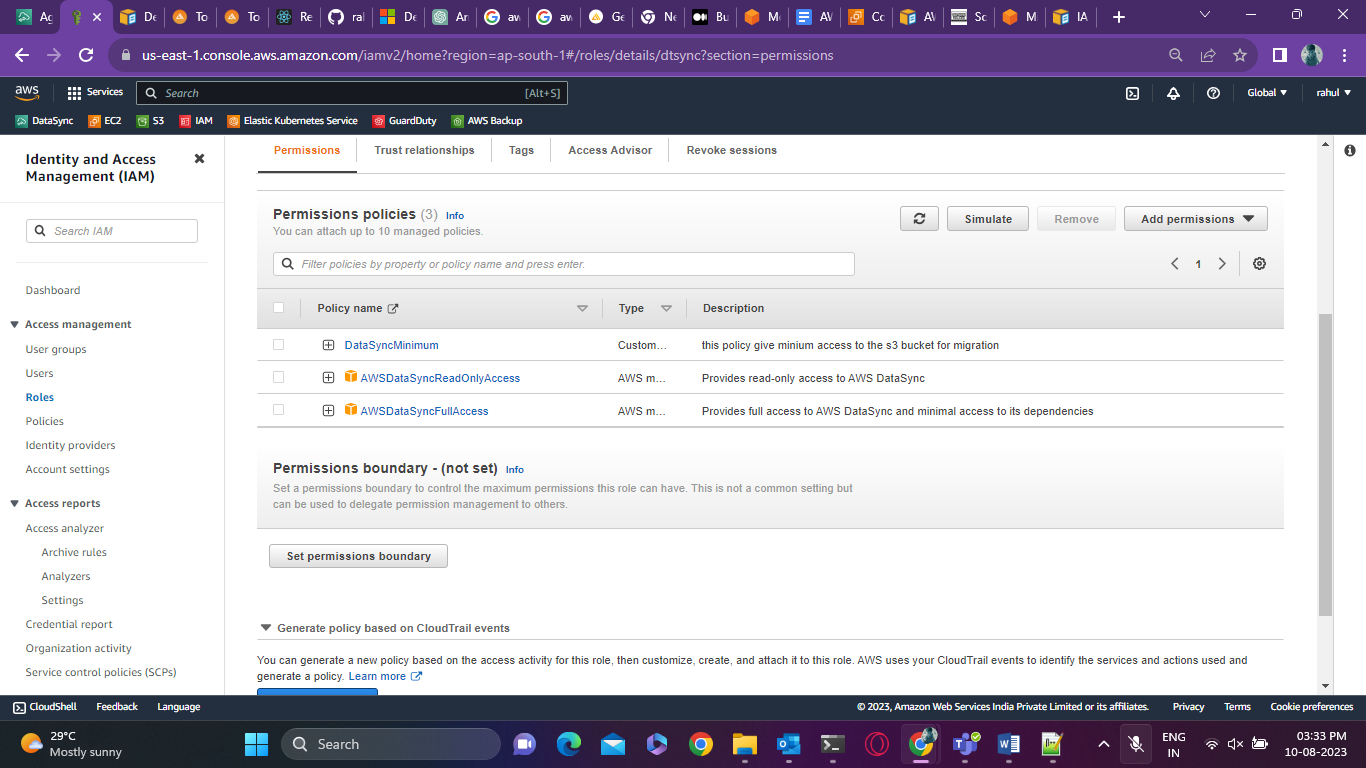
]

}

* Change the s3 bucket arn and create a policy after it is created we need to create a new role go-to the role and create a new role



* Create the role and after the role is created attach the policy you have created to this role



* After you have added the policy to the role go to the trust-relationship tab and replace it with this JSON code below

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "datasync.amazonaws.com"

},

"Action": "sts:AssumeRole",

"Condition": {

"StringEquals": {

"aws:SourceAccount": "123456789012"

},

"StringLike": {

"aws:SourceArn": "arn:aws:datasync:us-east-2:123456789012:\*"

}

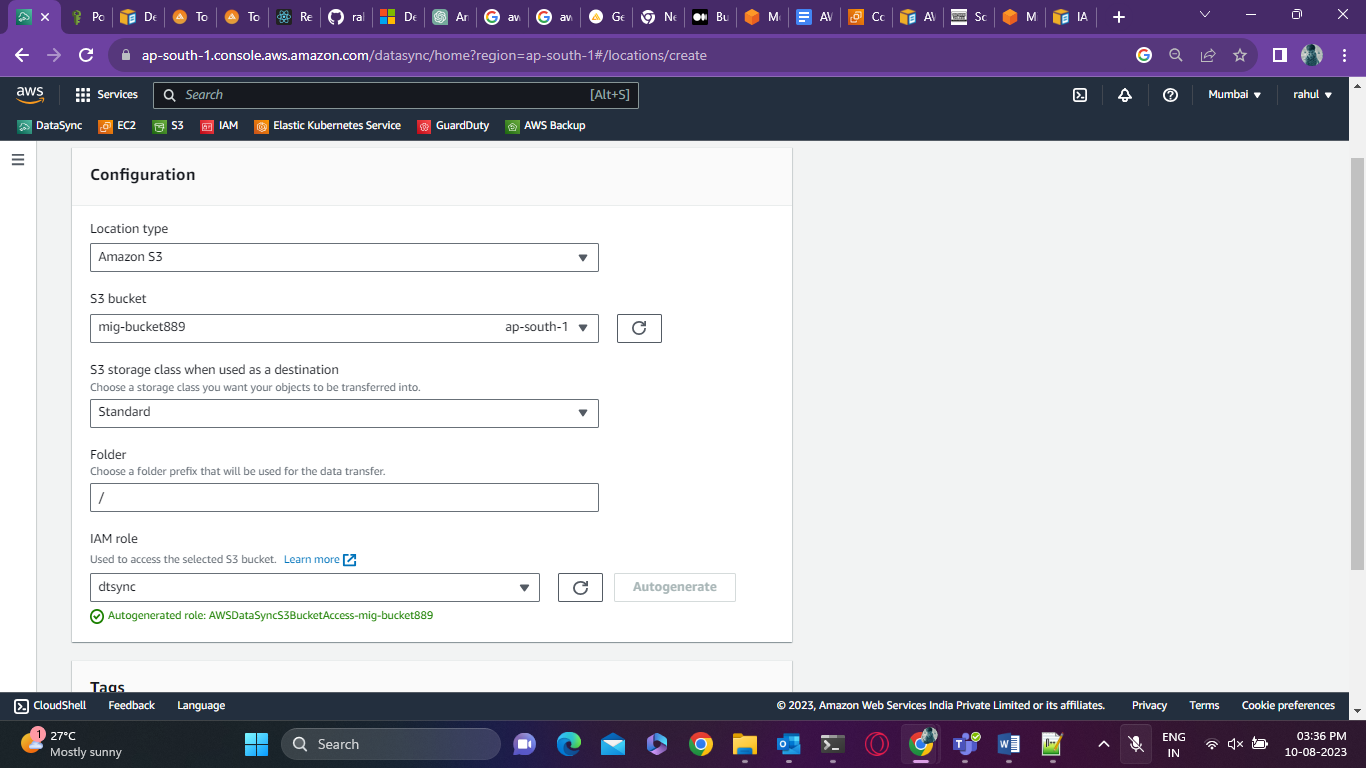
}

}

]

}

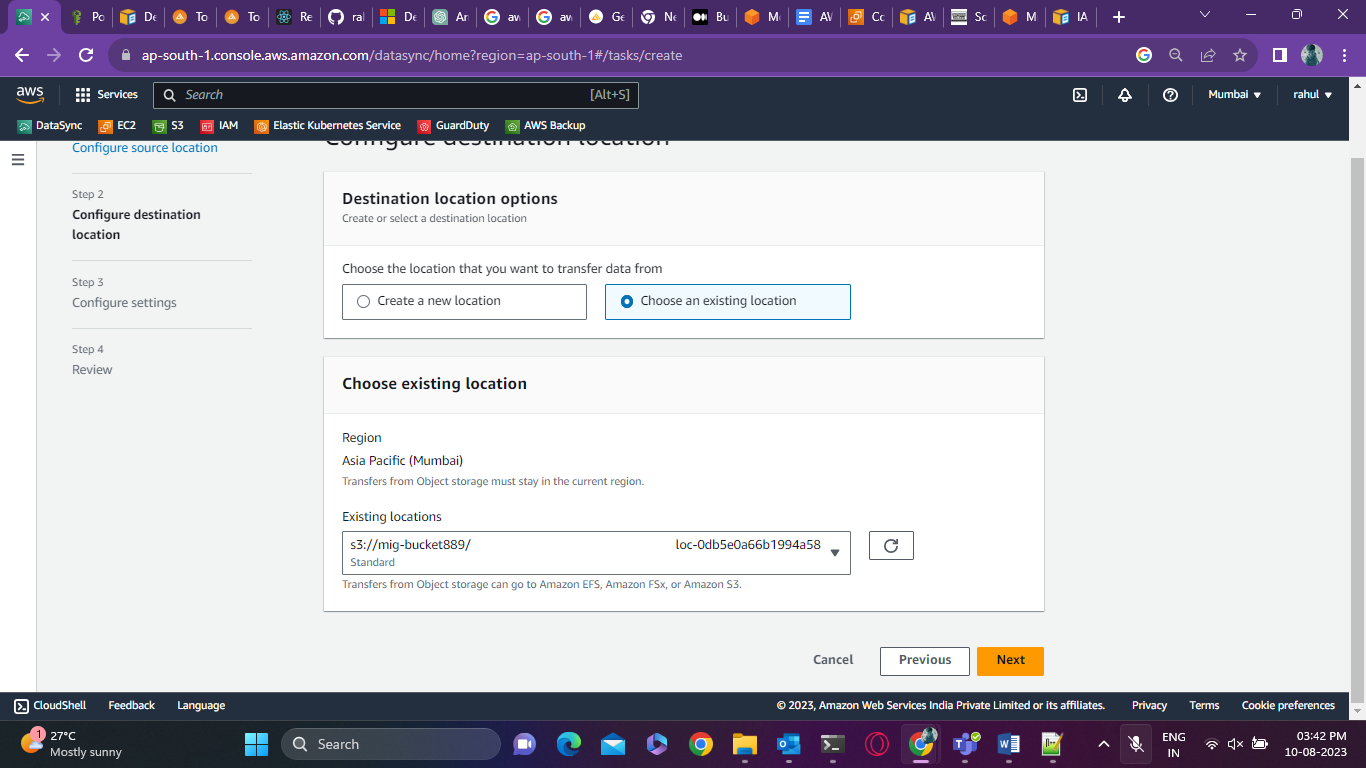
* Replace the account number and data sync arn number with your aws account number and save
* Now create a destination location from the data sync



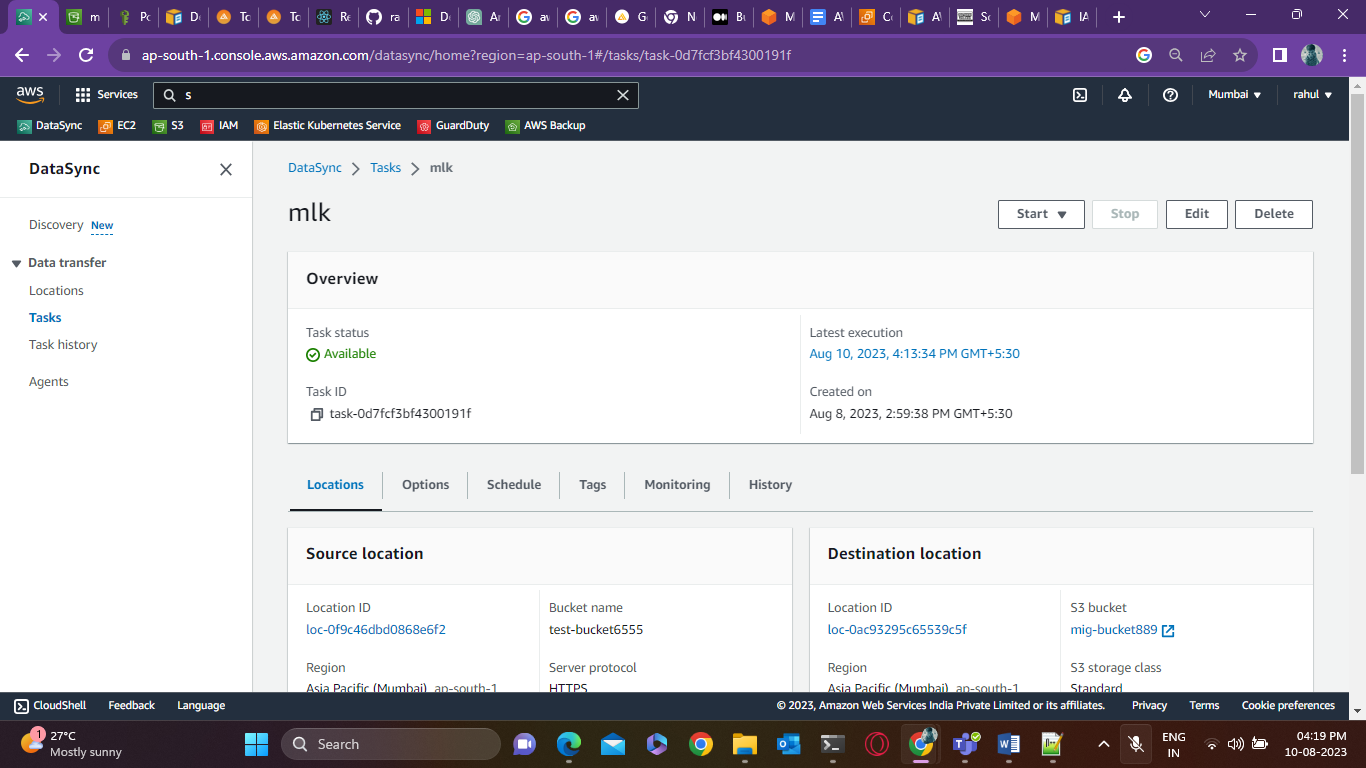
* Select s3 and select the correct role which you have just created and click on create
* Now go to tasks and click on Create tasks



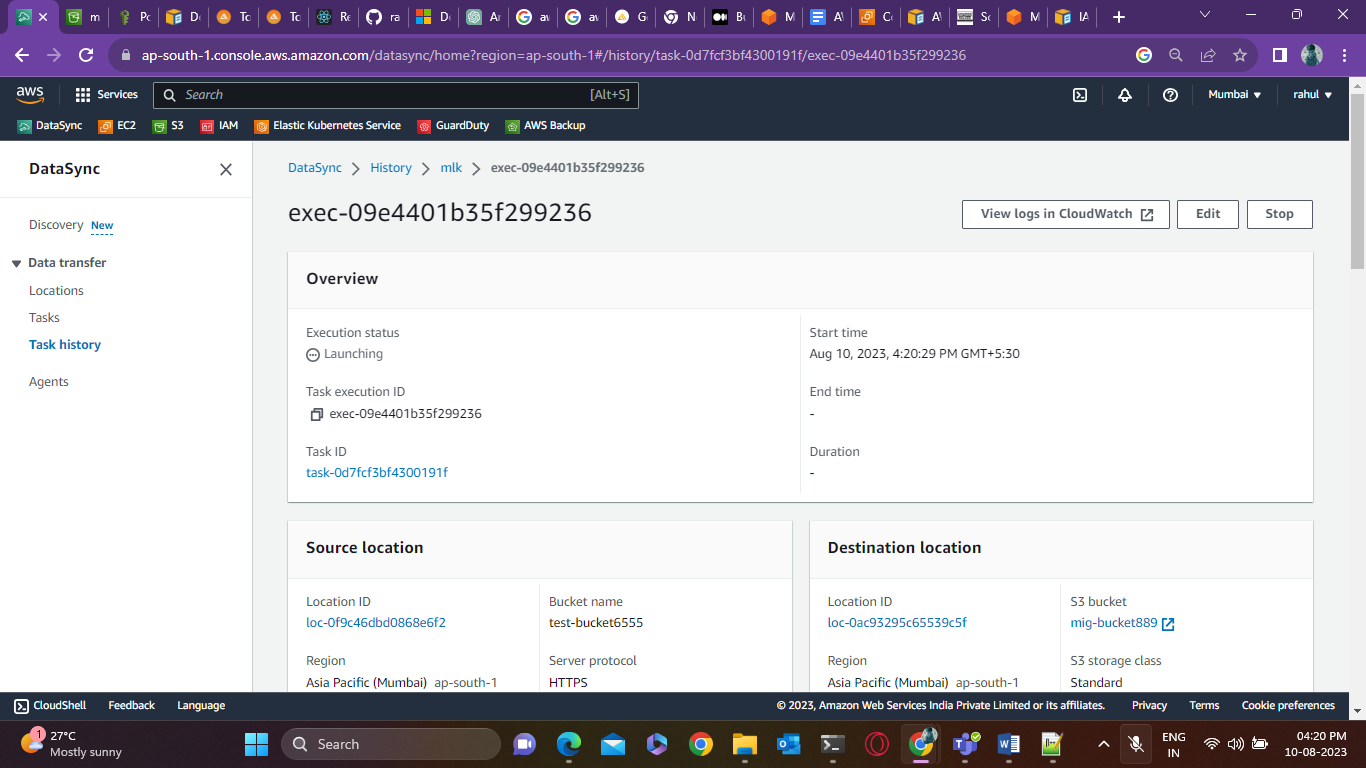
* Select the source from the existing location and go next



* Select the destination place and click next in the configuration settings give a task name and below transfer data select transfer all data and keep the rest as default if not required and click next
* After the task is created you need to go inside Start and on top click on start

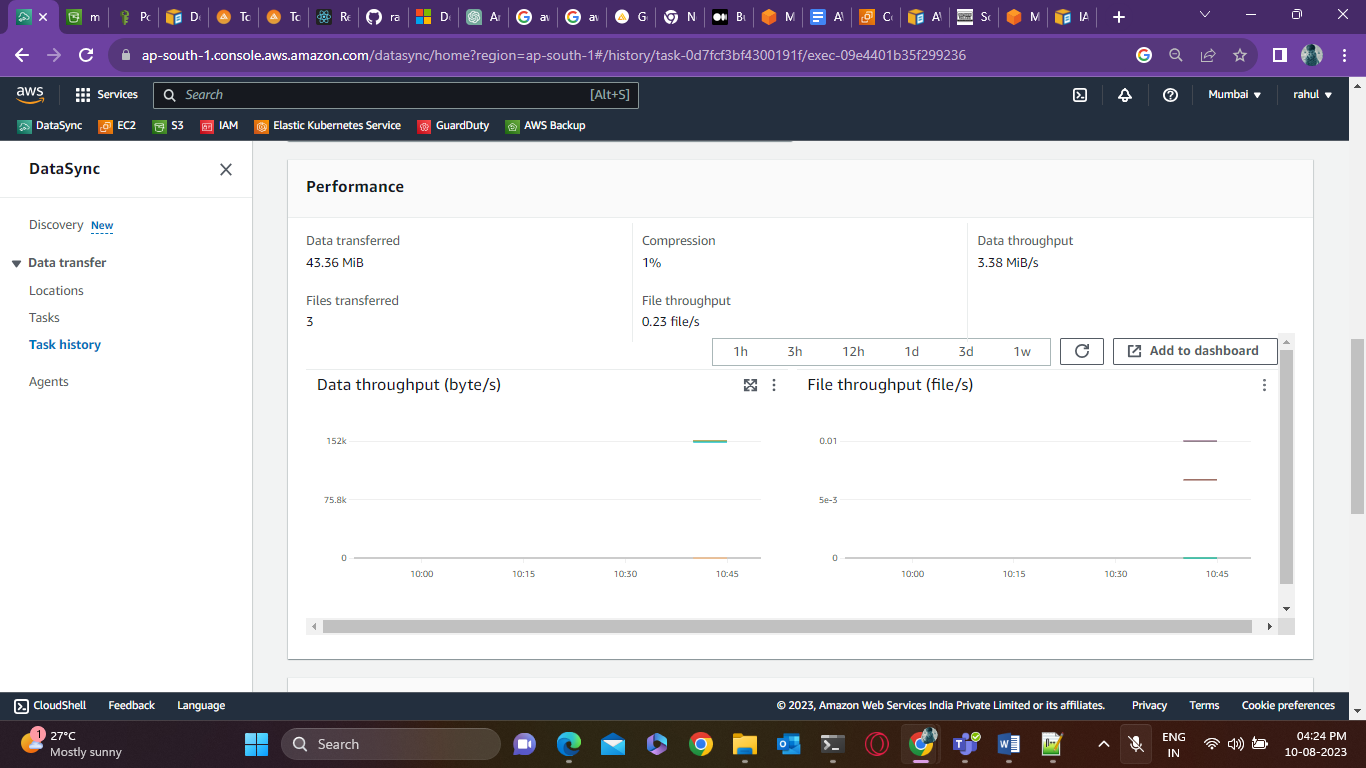


* The task will launch and will start the migration of everything is configured correct

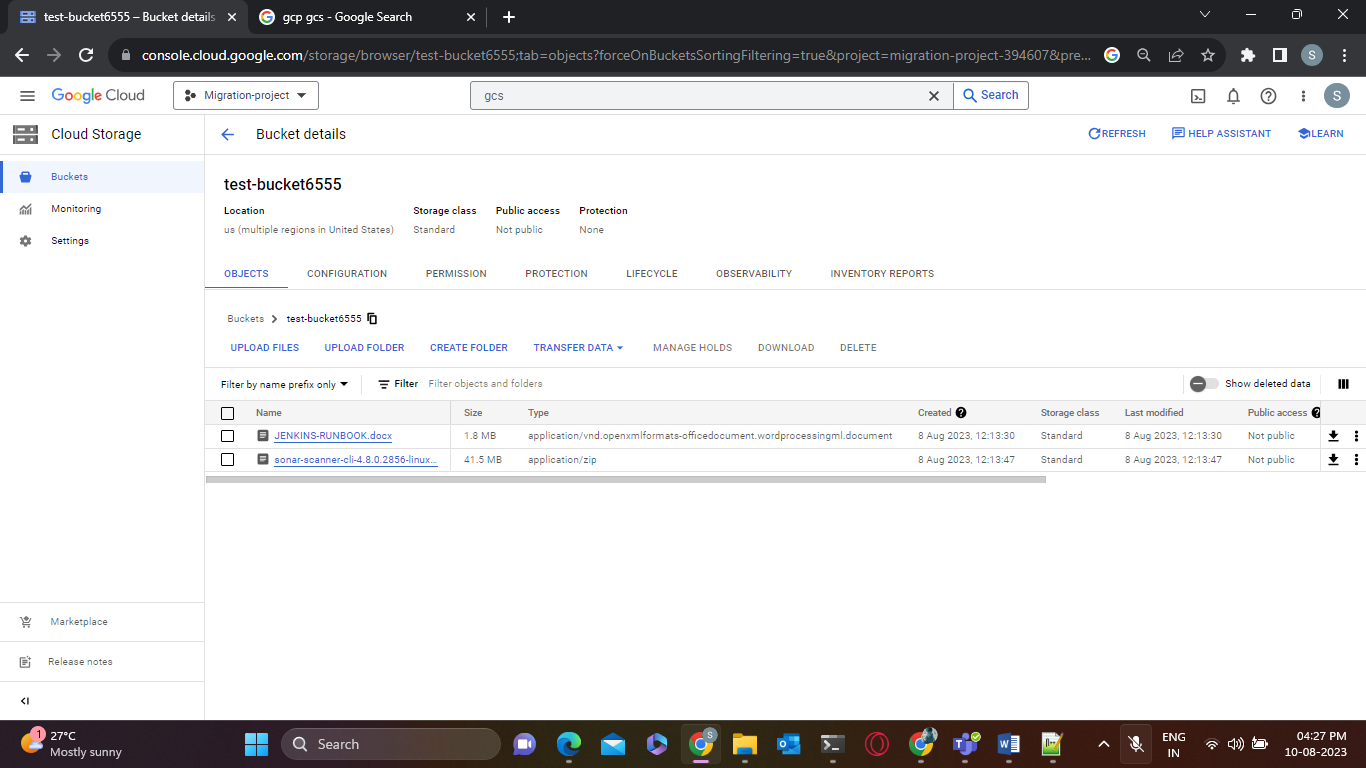




* The files have been transferred successfully



* Files in Google Cloud storage which is the source bucket



* S3 bucket where all the files have been migrated which is the destination or target bucket

