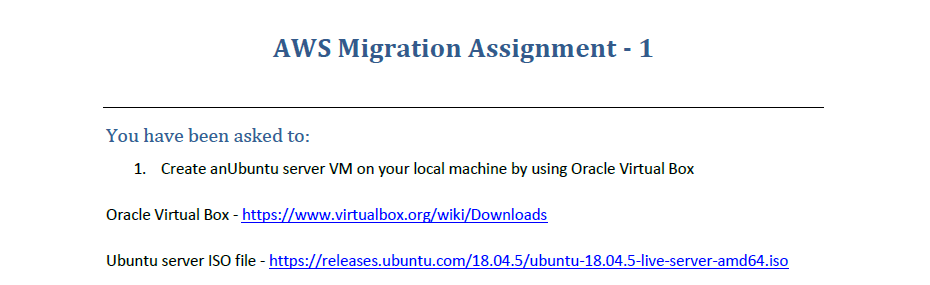
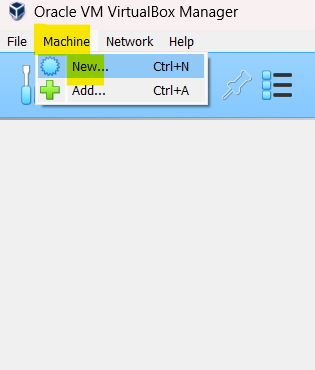
**Migration Assignment**

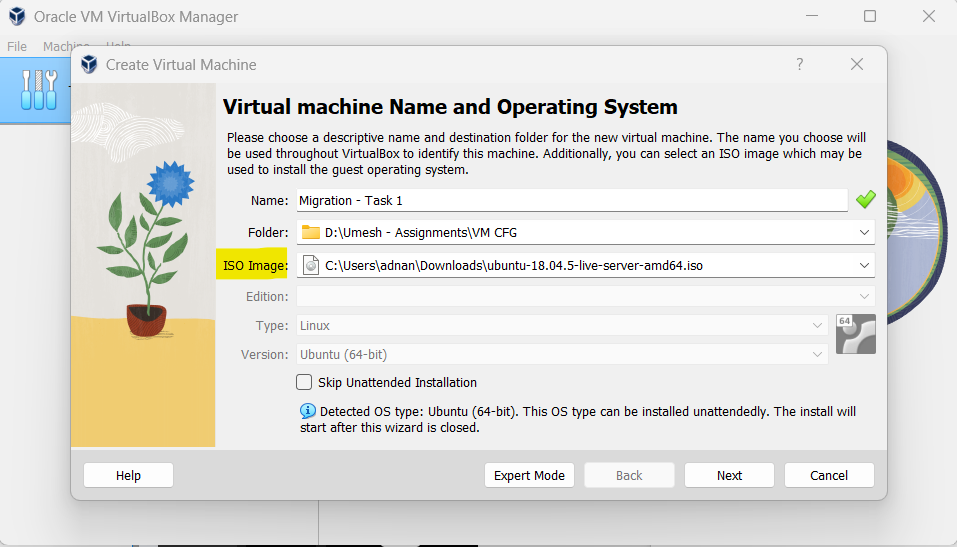
**Task 1**

****

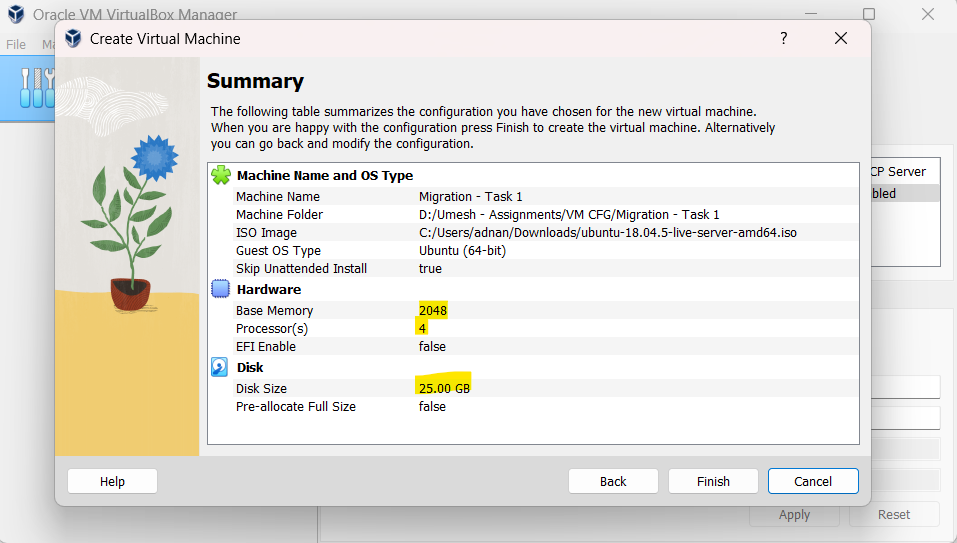
1. Open Virtual and Box and click on Machine > New to configure a new VM.



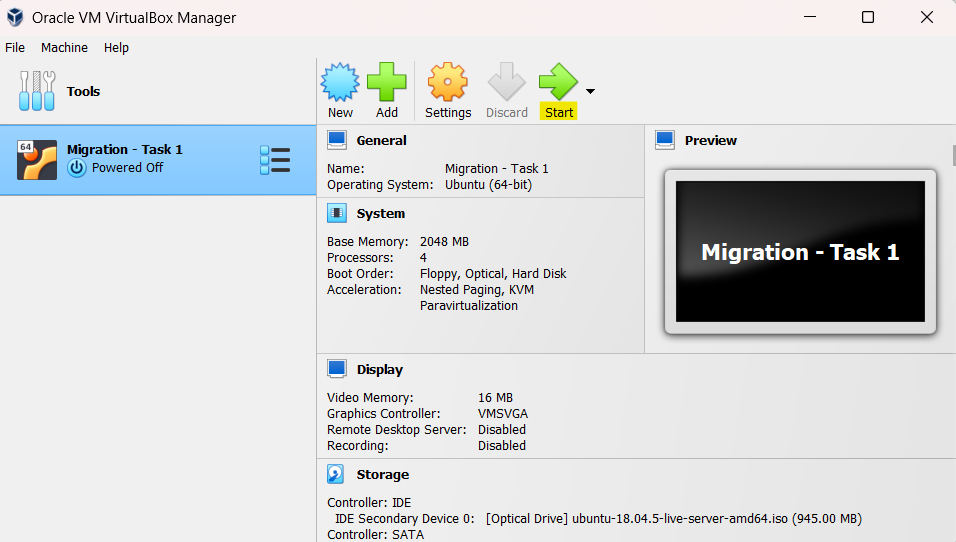
1. Make sure the ISO file is downloaded, provide the file path and proceed for installation.



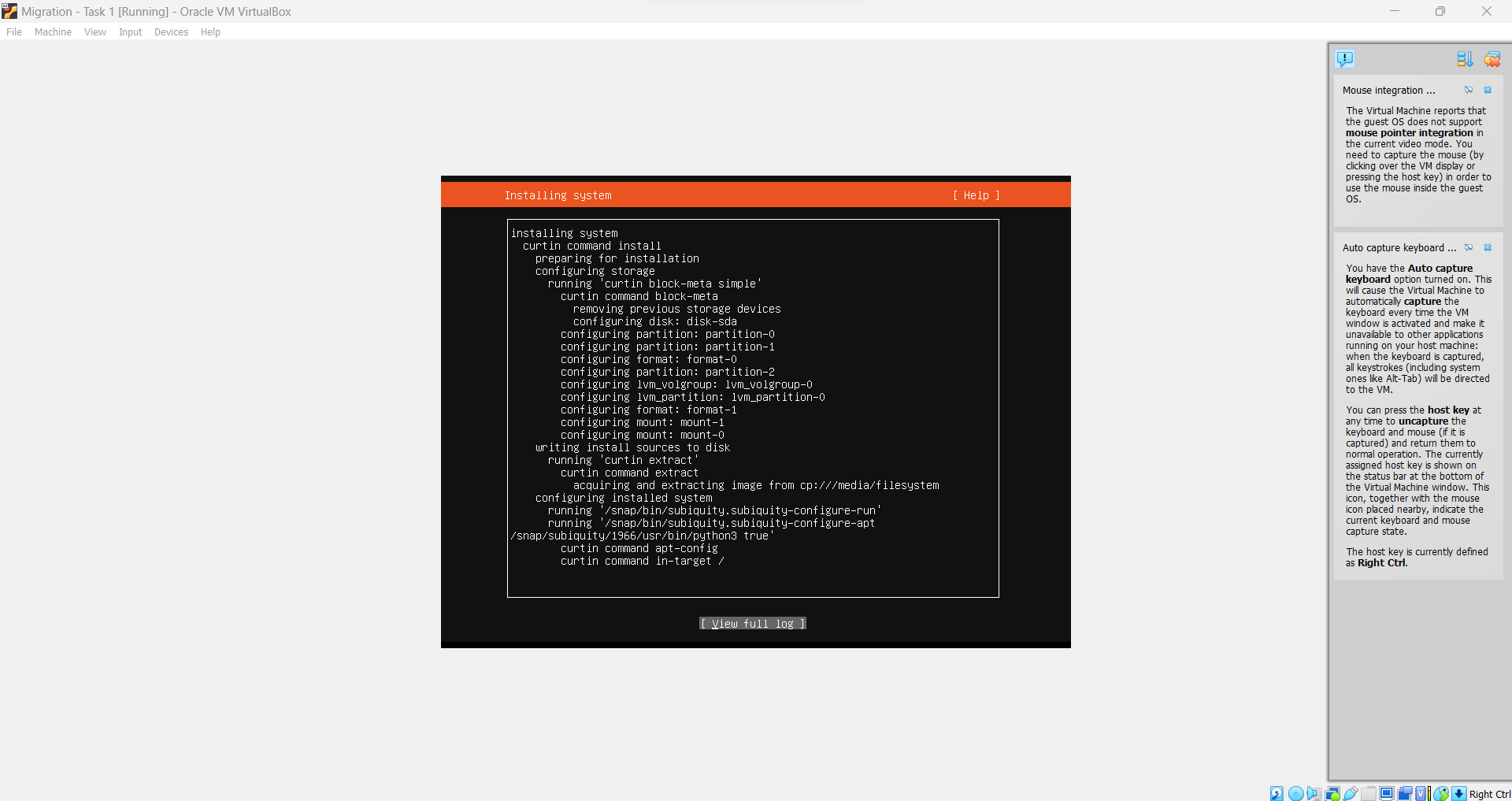
1. Specify the Hardware, Storage and other requirements as per your convenience. (I am going with default settings)
2. Cross Verify the Summary to check if everything is selected as expected.



1. Click Start to Start your VM.



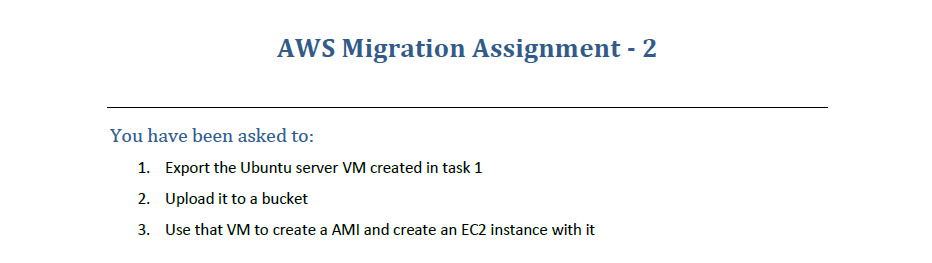
1. After starting the VM, select all the default values and proceed with installation.



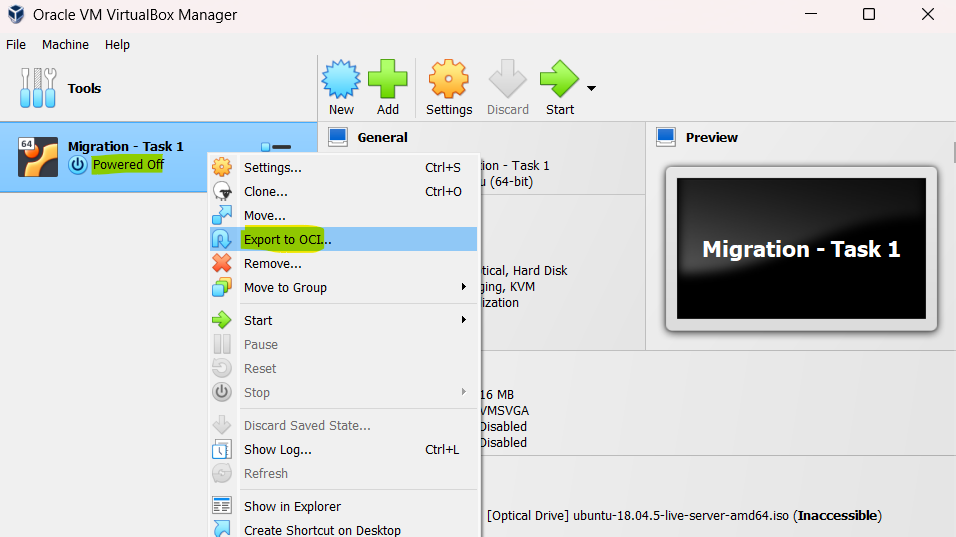
1. The Linux Server VM has been created successfully.

**Migration Assignment**

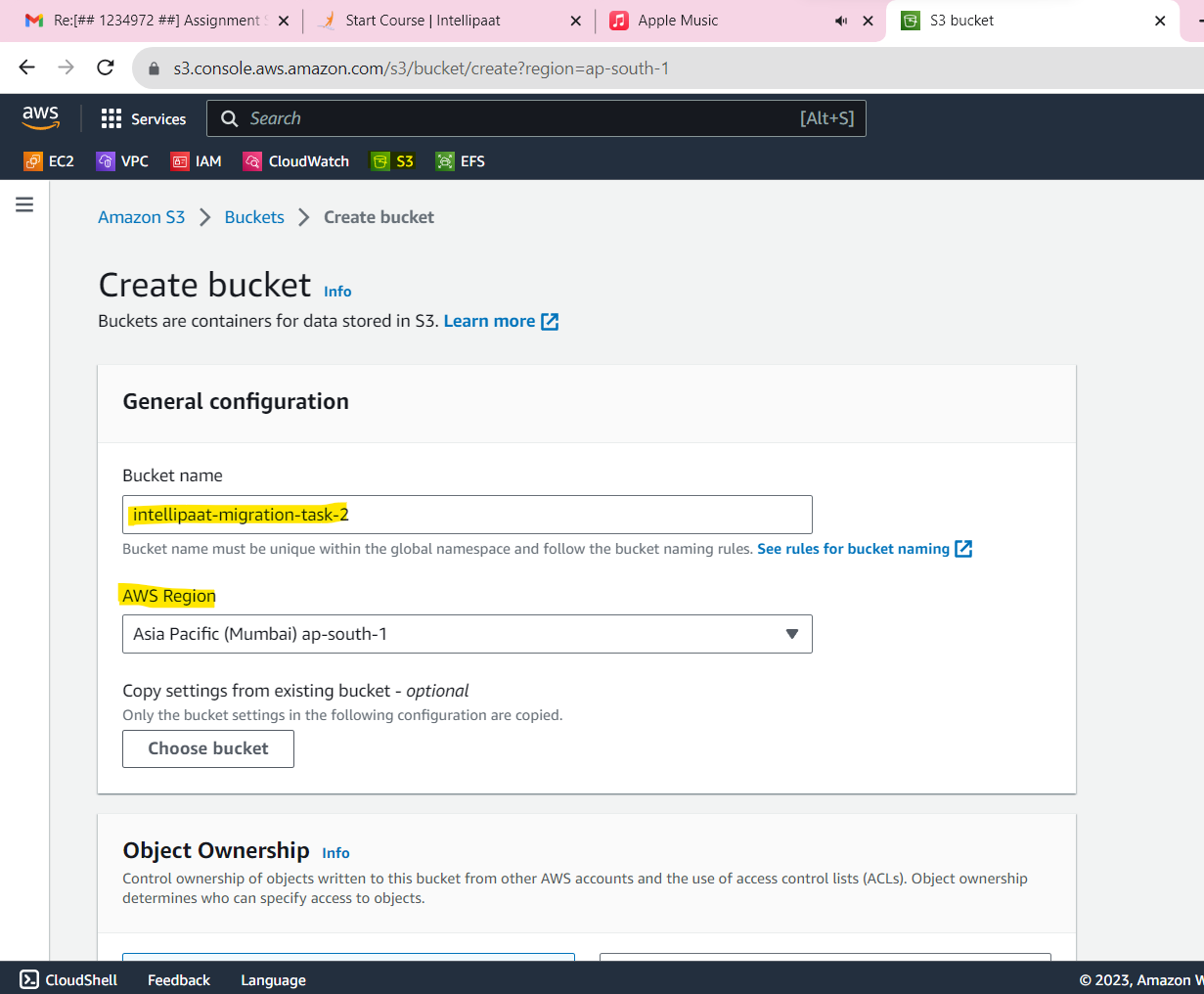
**Task 2**

****

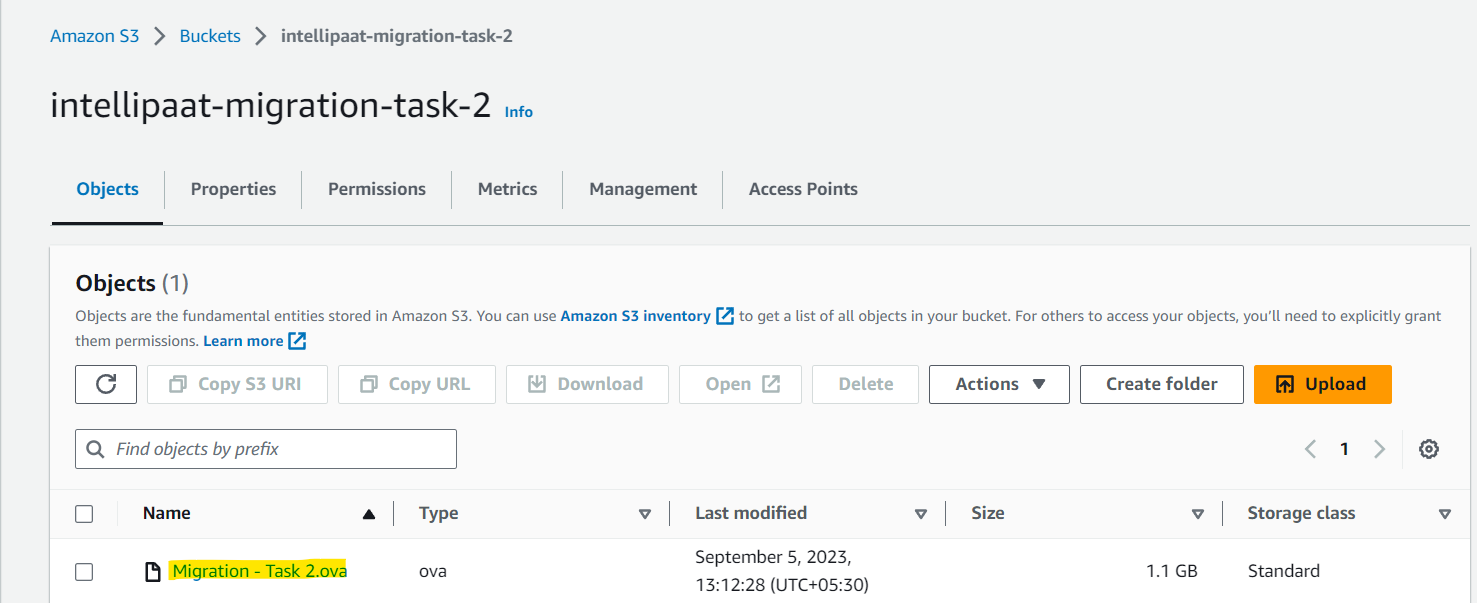
1. Power off the VM, and right click on the VM and click on Export to OCI.



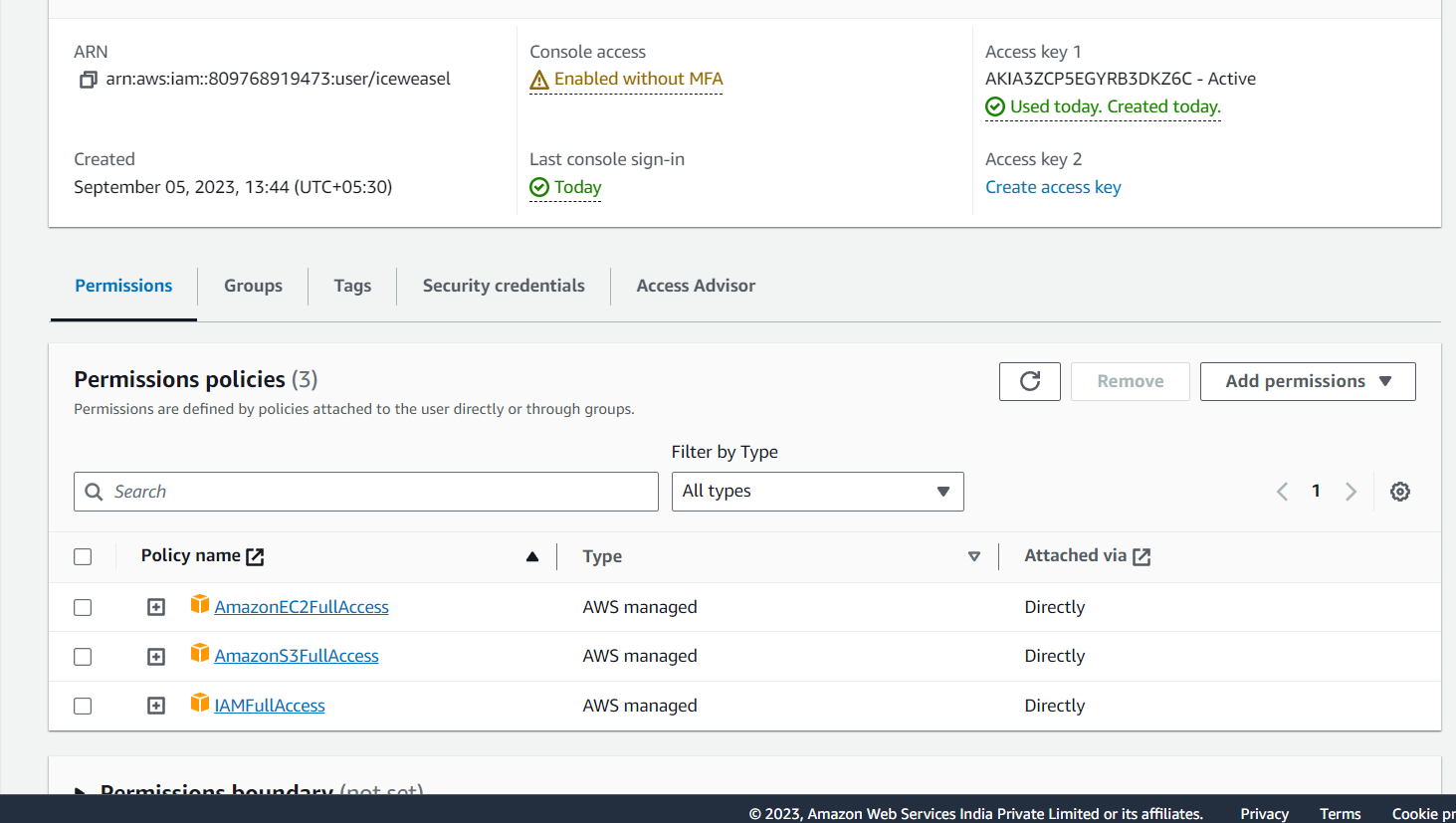
1. Create a S3 Bucket in the Region where you want the VM to be Imported and upload the exported OVA File to the S3 Bucket.



1. Upload the OVA to the Bucket.



1. Setup all the required configurations and create a role and policy to execute the below commands :
2. Create a EC2 User with EC2, IAM and S3 Full Access and generate the Access Key.



1. Configure the AWS CLI with the new User’s Access Key :

aws configure (Pass on the Access and Secret Key and the region should be the one where the bucket is created)

1. Create a Role with the below policy :

sudo nano trust-policy.json

{

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

"Statement": [

{

"Effect": "Allow",

"Principal": { "Service": "vmie.amazonaws.com" },

"Action": "sts:AssumeRole",

"Condition": {

"StringEquals":{

"sts:Externalid": "vmimport"

}

}

}

]

}

aws iam create-role --role-name vmimport --assume-role-policy-document "file:///home/ec2-user/trust-policy.json" ----> This command creates a role “vmimport” with the policy of “trust-policy,json” contents

1. Create an IAM Policy for the role which will grant access to the S3 Bucket where the ova file is uploaded.

sudo nano role-policy.json

{

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

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:GetBucketLocation",

"s3:GetObject",

"s3:ListBucket",

"s3:PutObject","s3:GetBucketAcl"

],

"Resource": [

"arn:aws:s3:::intellipaat-migration-task-2",

"arn:aws:s3:::intellipaat-migration-task-2/\*"

]

},

{

"Effect": "Allow",

"Action": [

"ec2:ModifySnapshotAttribute",

"ec2:CopySnapshot",

"ec2:RegisterImage",

"ec2:Describe\*"

],

"Resource": "\*"

}

]

}

aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document "file:///home/ec2-user/role-policy.json"

1. Create a file bucket.json and update it with your bucket name and object name.

[

{

"Description": "My Server OVA",

"Format": "ova",

"UserBucket": {

"S3Bucket": "intellipaat-migration-task-2",

"S3Key": "Migration - Task 2"

}

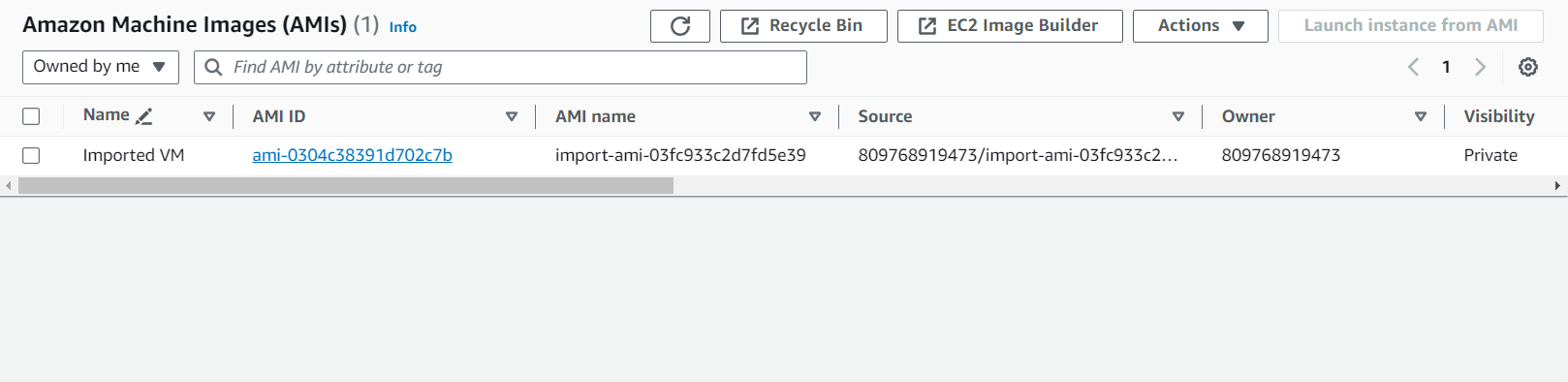
}

]

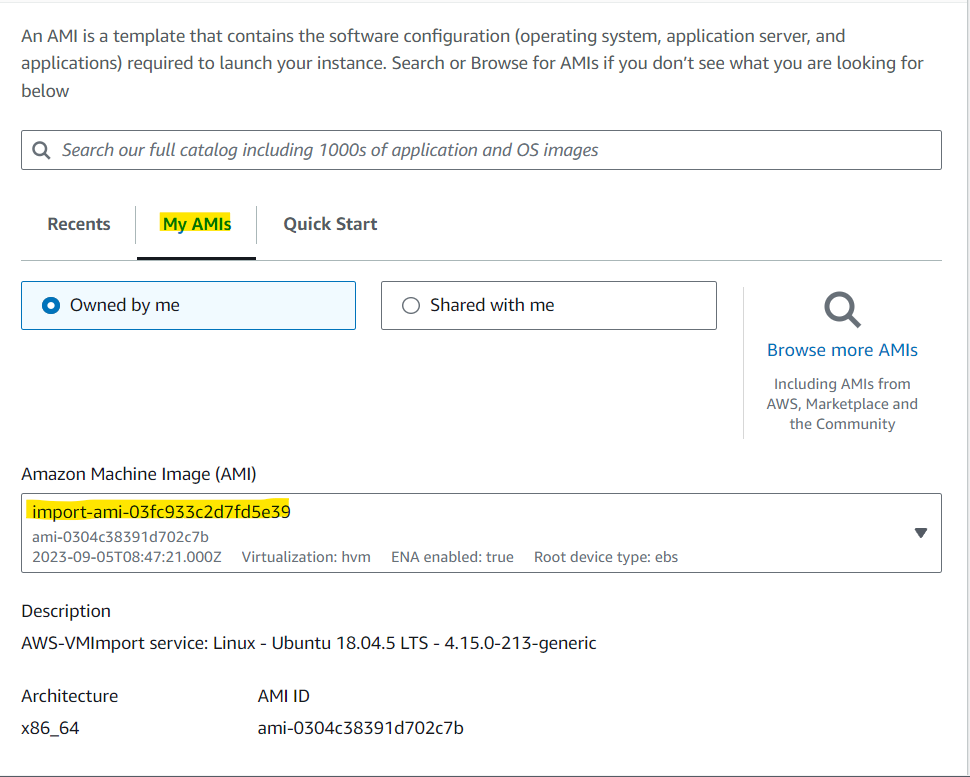
1. Finally after completing all the above steps, import the image which will create an AMI by below command :

aws ec2 import-image --description "Imported VM" --disk-containers "file:///home/ec2-user/bucket.json"

1. Go to EC2 and Check under AMIs, you will now see a new AMI which was imported via S3.

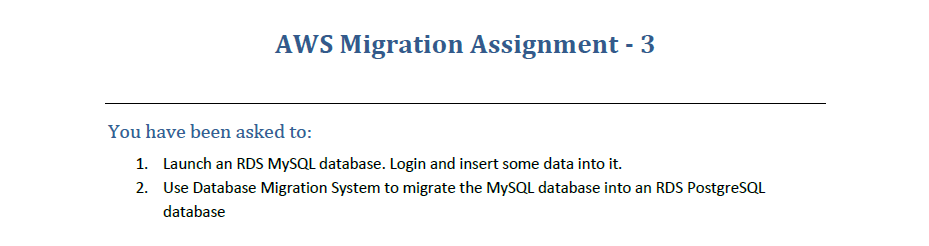


1. Now, you can launch the Instance with this Custom AMI.

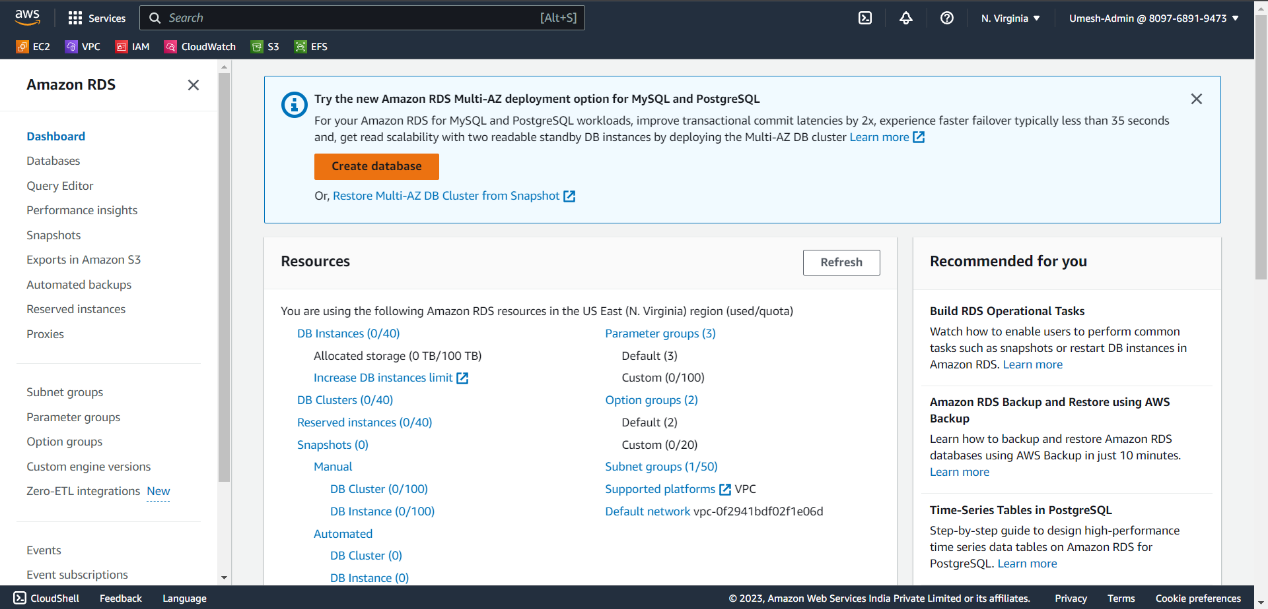


**Migration Assignment**

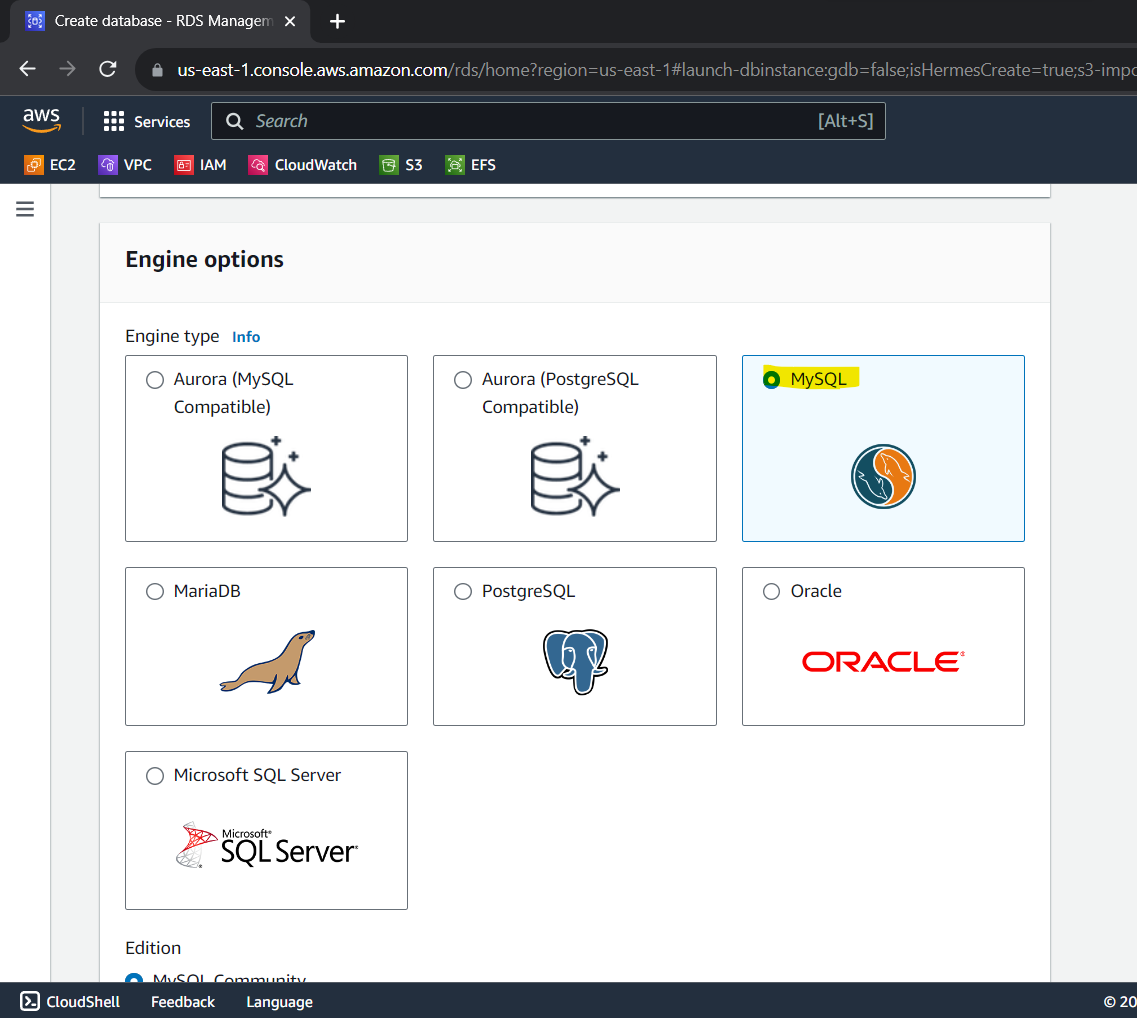
**Task 3**

****

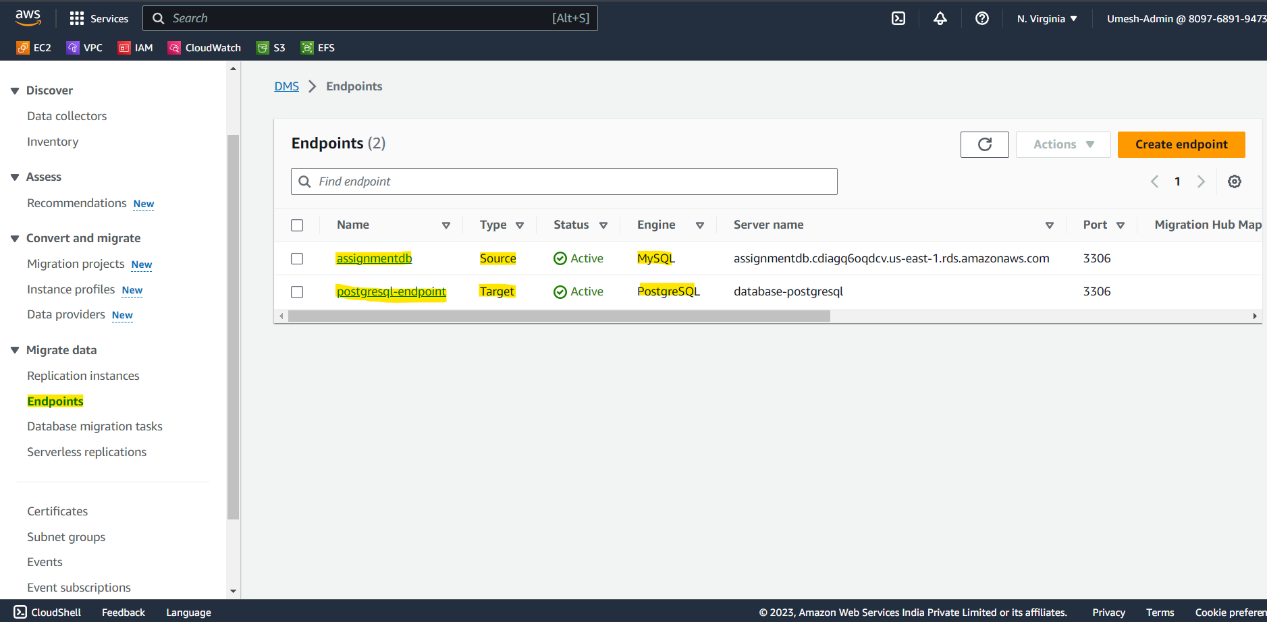
1. Go to AWS RDS Service and Click on Create a Database.



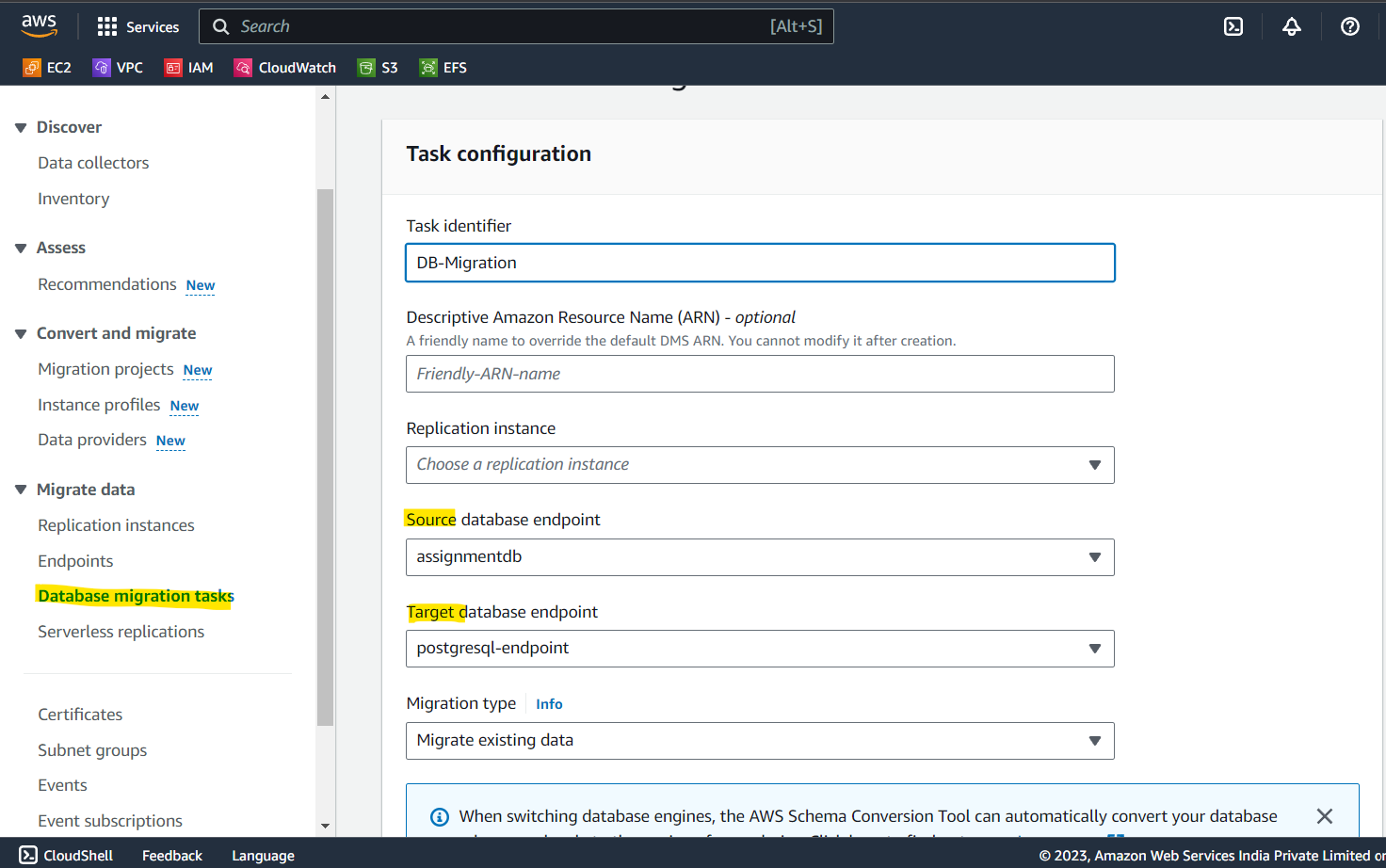
1. Choose MySQL as the Engine.



1. Set the DB Admin Credentials and keep the rest as default.
2. Connect to the DB via EC2 / Local and create Table and Add Values into the Table.
3. Go to AWS DMS Service. (Database Migration Service)
4. First, Create Replication Instance (all default settings) which will help us in migration process.
5. Create 2 Endpoints (1. Source i.e MySQL [already exists] 2. Target i.e PostgreSQL)



1. Create a Database Migration Task and input both the endpoint to migrate from MySQL to PostgreSQL.



1. The replication instance will take care of the rest! The DB will be Migrated from MySQL to PostgreSQL via AWS DMS (Database Migration Service)