

## AWS Migration Assignment - 1

---

You have been asked to:

1. Create anUbuntu server VM on your local machine by using Oracle Virtual Box

Oracle Virtual Box - <https://www.virtualbox.org/wiki/Downloads>

Ubuntu server ISO file - <https://releases.ubuntu.com/18.04.5/ubuntu-18.04.5-live-server-amd64.iso>

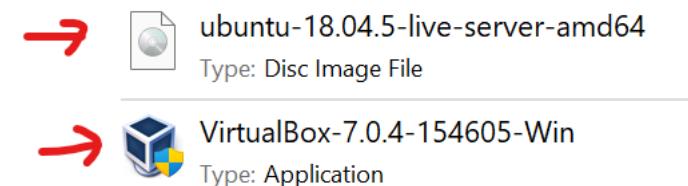
---

**Step-1:** Download Oracle Virtual box from  
<https://www.virtualbox.org/wiki/Downloads>

**Step-2:** Download Ubuntu Server ISO file from  
<https://releases.ubuntu.com/18.04.5/ubuntu-18.04.5-live-server-amd64.iso>



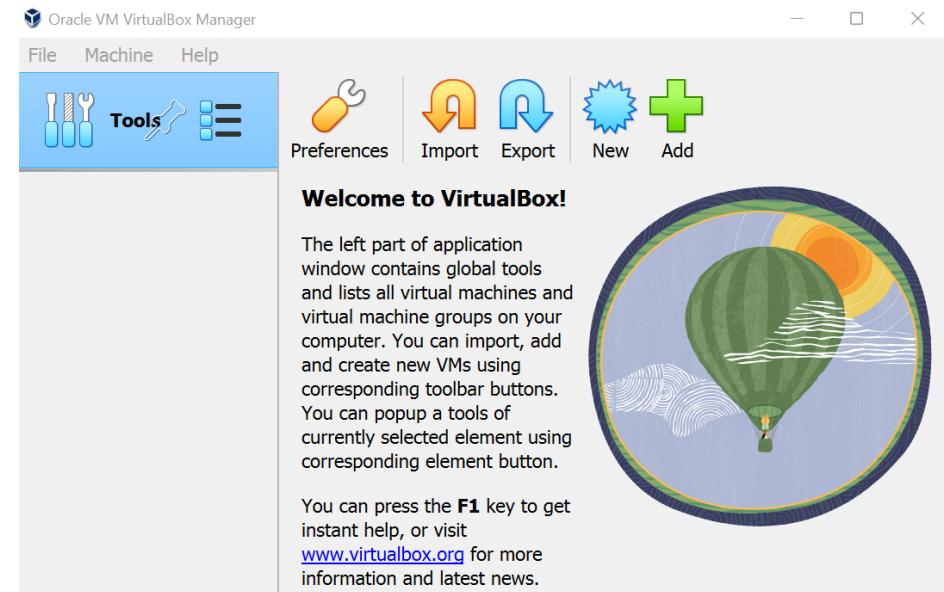
The screenshot shows a web browser window with the URL <https://www.virtualbox.org/wiki/Downloads>. The page title is "VirtualBox Download VirtualBox". It features a large blue "VirtualBox" logo. On the left, there's a sidebar with links: About, Screenshots, Downloads, Documentation, End-user docs, Technical docs, Contribute, and Community. The "Downloads" link is highlighted. The main content area has a heading "VirtualBox binaries" followed by a paragraph about terms and conditions. Below that is a section for "VirtualBox 7.0.4 platform packages" with a bulleted list including "Windows hosts" (which is highlighted in yellow), "macOS / Intel hosts", "Developer preview for macOS / Arm64 (M1/M2) hosts", "Linux distributions", "Solaris hosts", and "Solaris 11 IPS hosts".



### Step-3: Install Oracle VirtualBox.

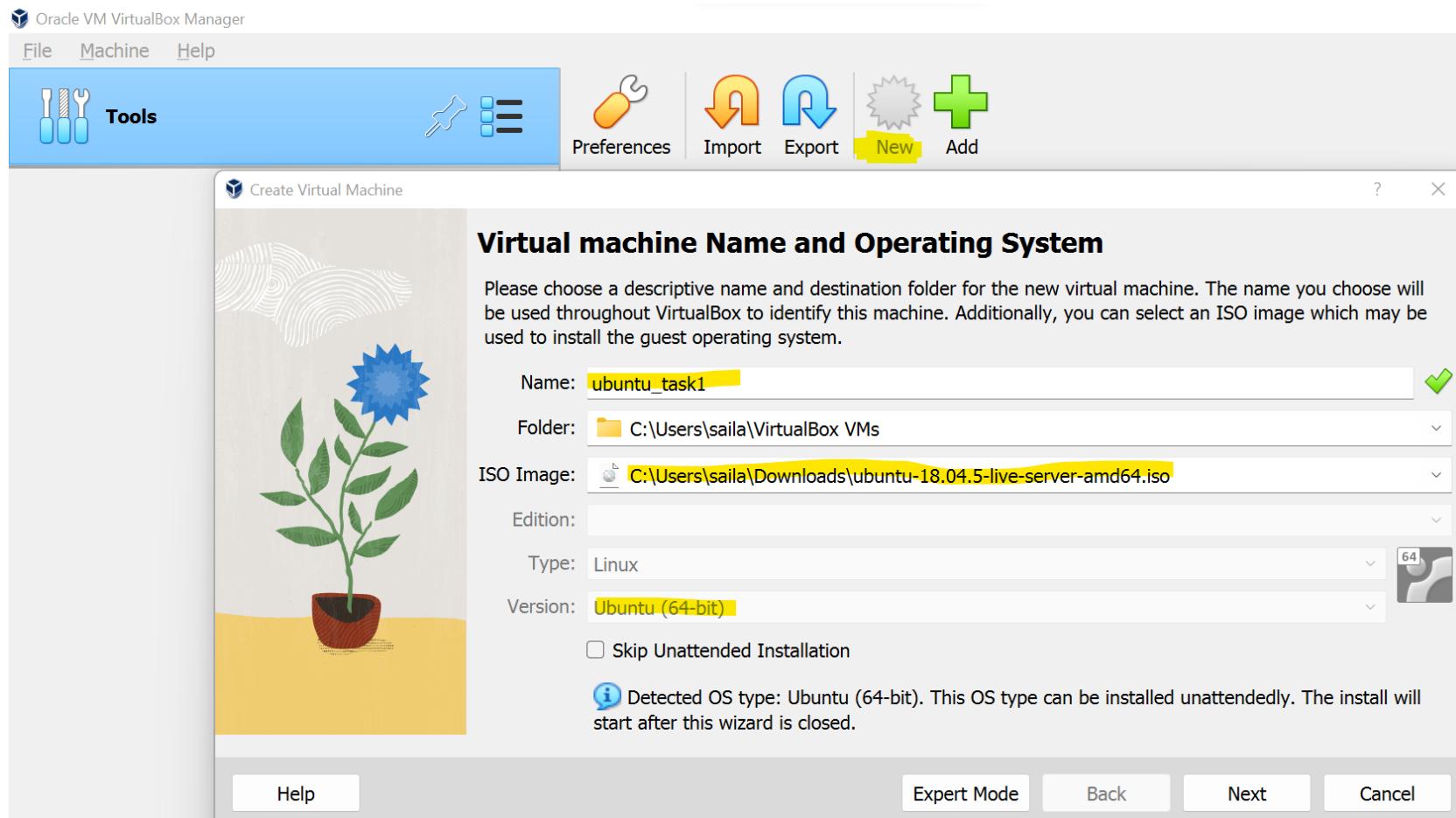


**Step-4:** Once installation of Oracle VirtualBox is complete, start Oracle VirtualBox to configure Ubuntu OS.

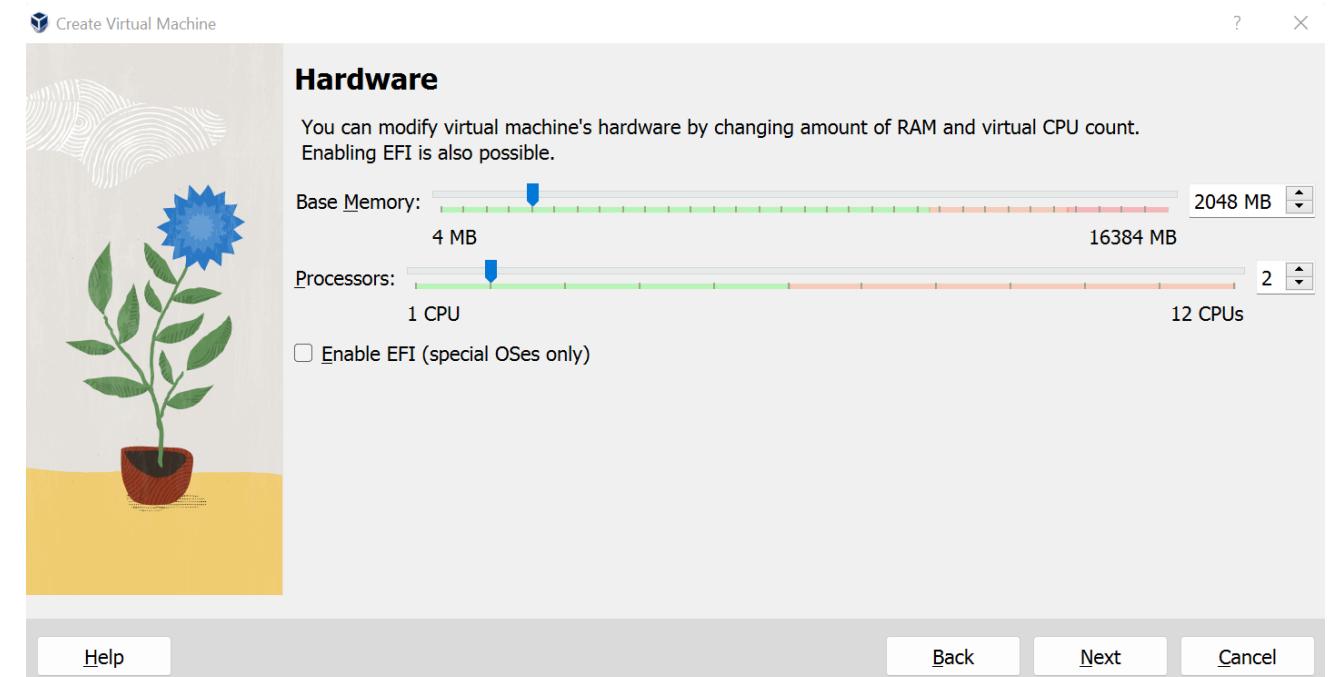
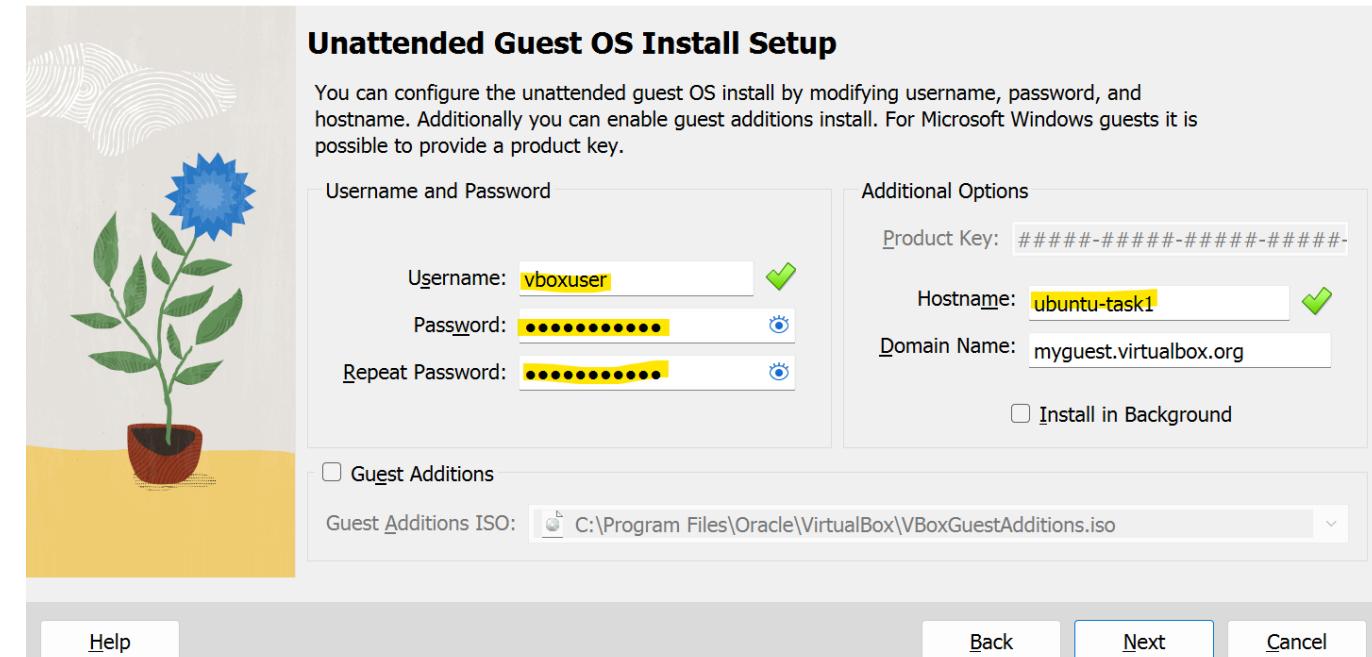


## Step-5: Provide “Name” and ISO image downloaded location as highlighted.

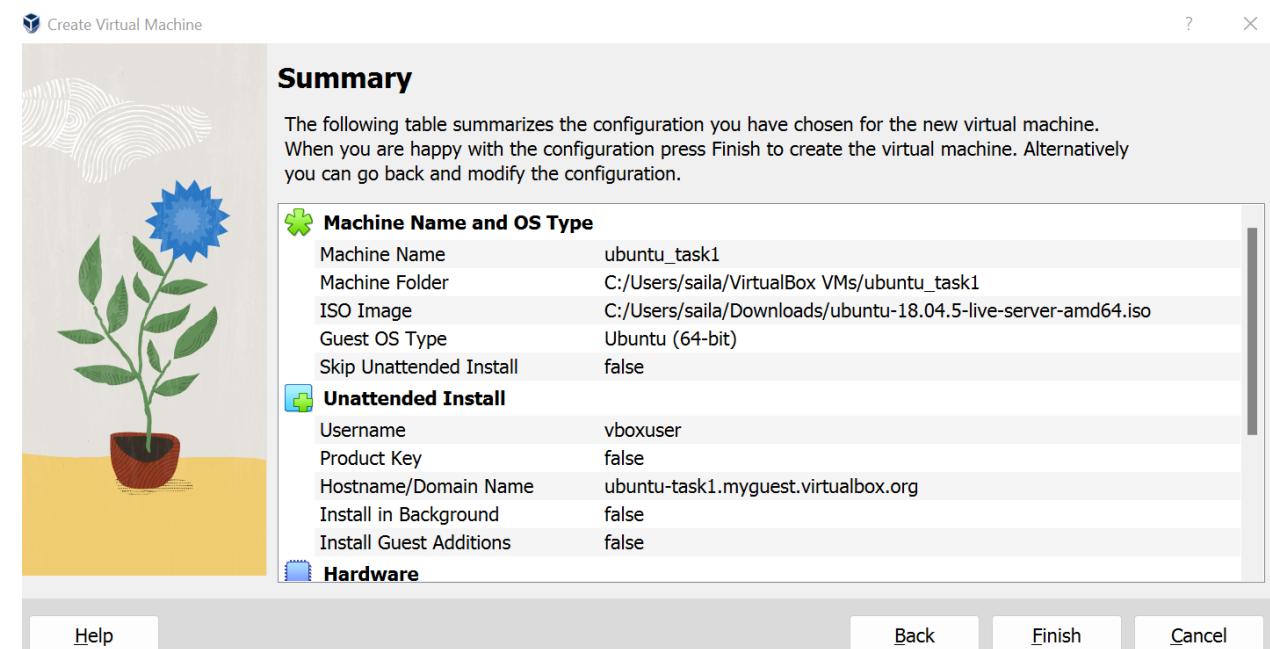
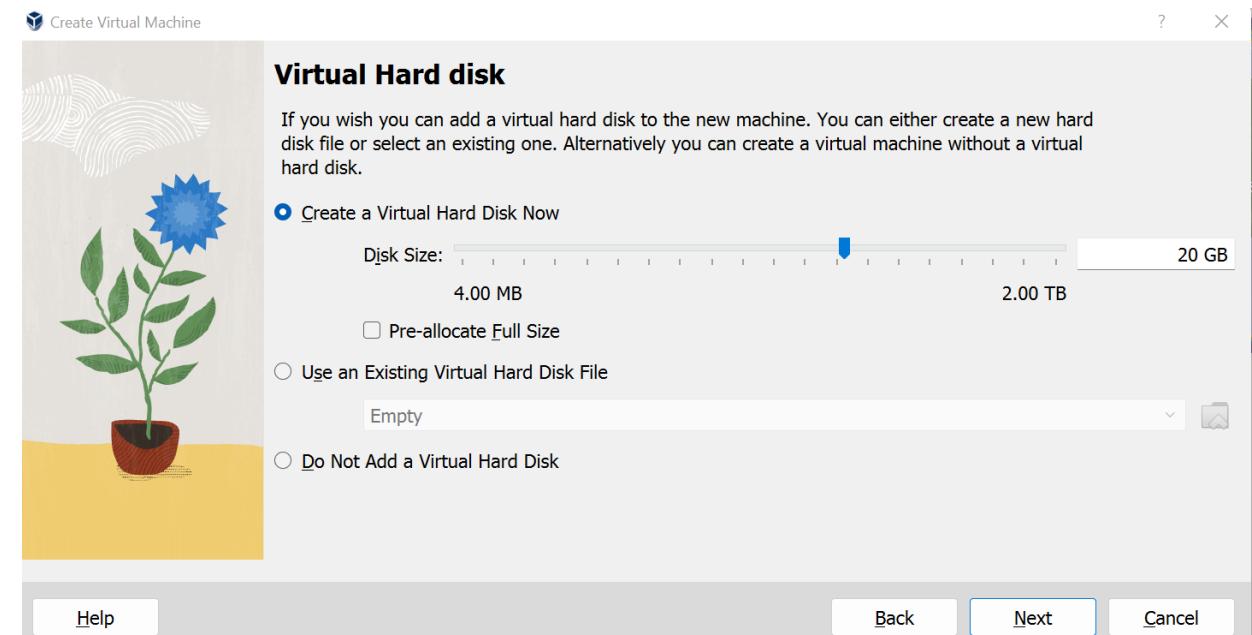
**Note:** While specifying the “name”, based on few characters, it automatically identifies the OS and selects. For example, here I have entered name as “ubuntu-task1” and the “Type” is selected as “Linux” and Version as “Ubuntu (64-bit)”



**Step-6:** Enter required details as highlighted.

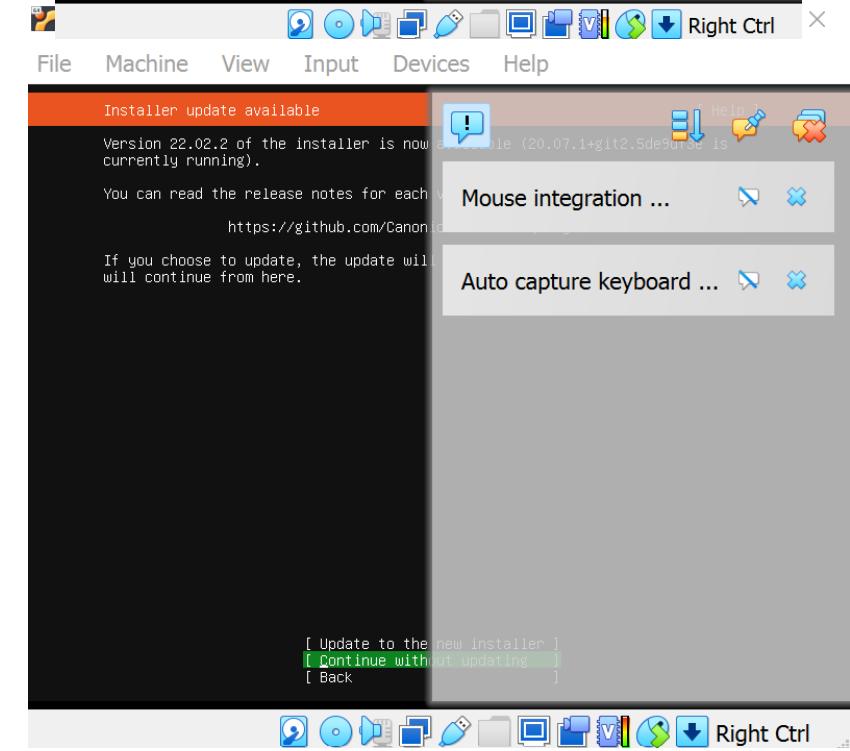
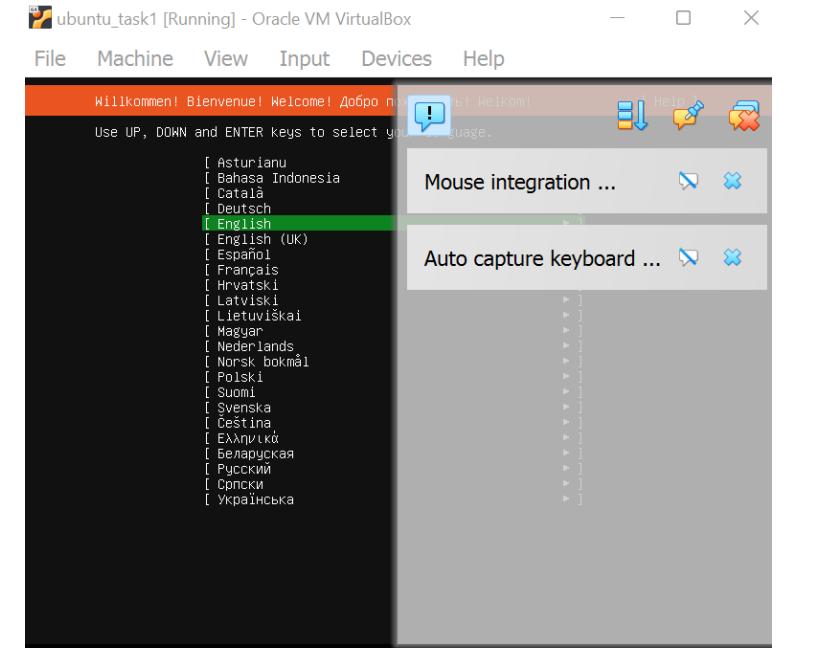


## Step-8: Specify the Virtual Hard disk specifications based on the resources availability on your machine and requirement.



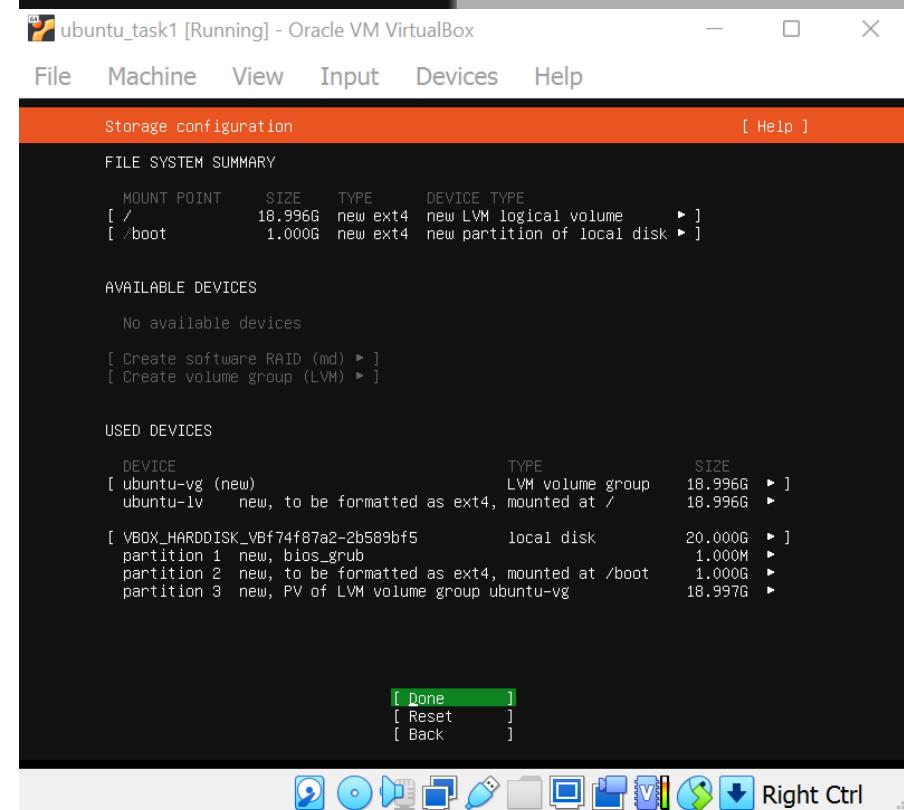
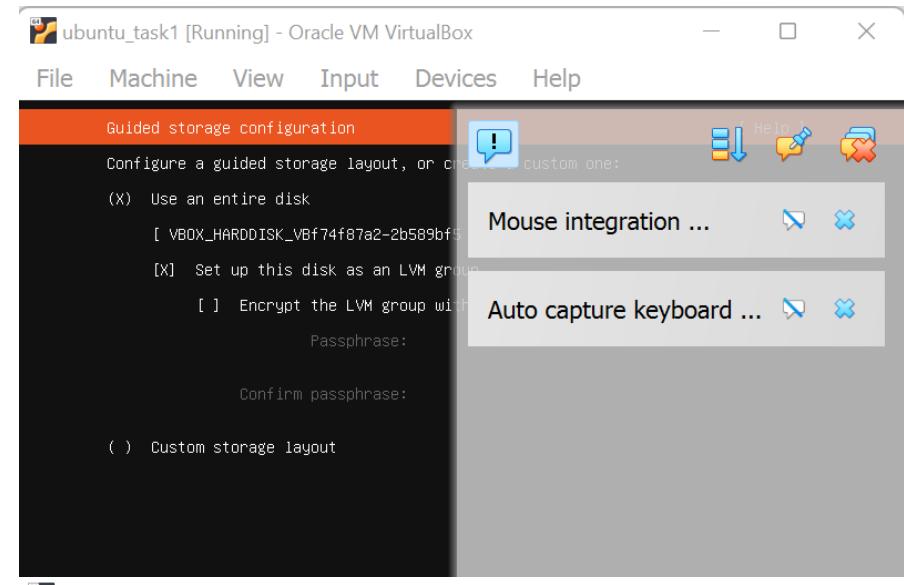
**Step-9:** Once the VirtualBox configuration is complete, it will start with the Linux VM creation process.

Choose the required parameters as selected in the images.



## Step-10: Specify the storage configuration for VM.

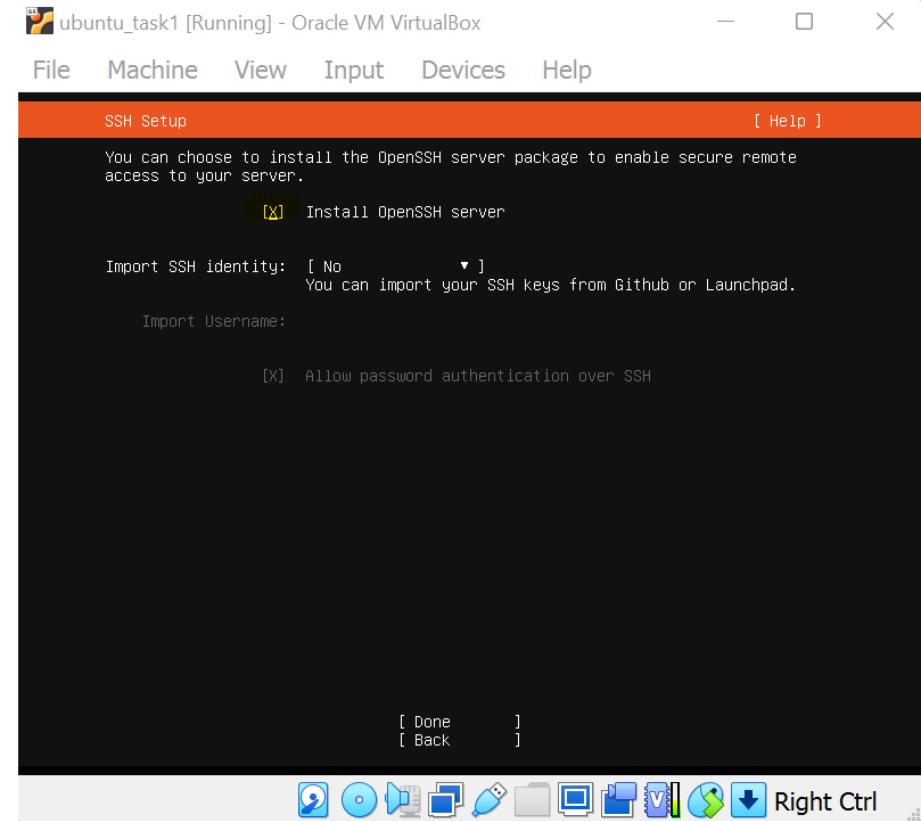
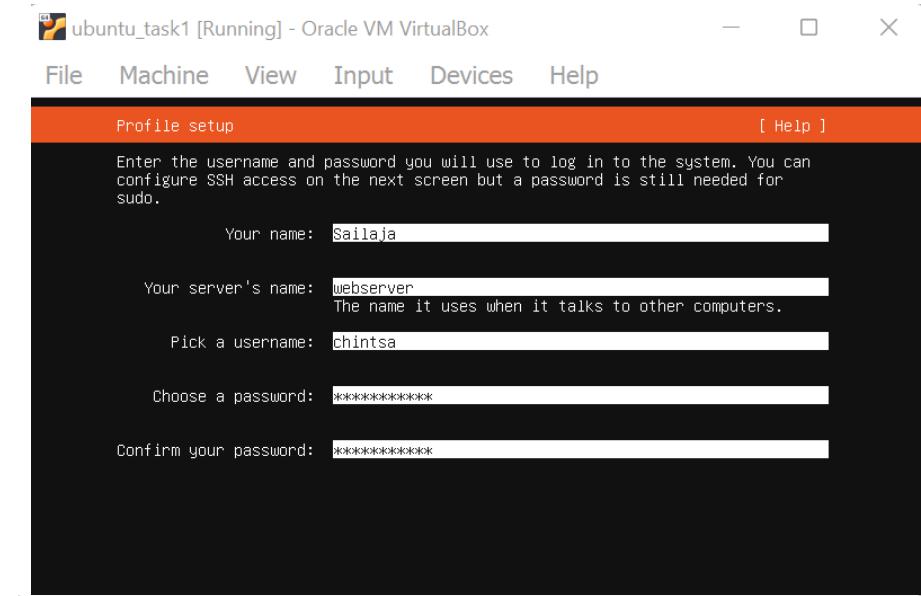
Choose the required parameters as selected in the images.



**Step-11:** Enter your **Profile details to login to the Linux VM** created.

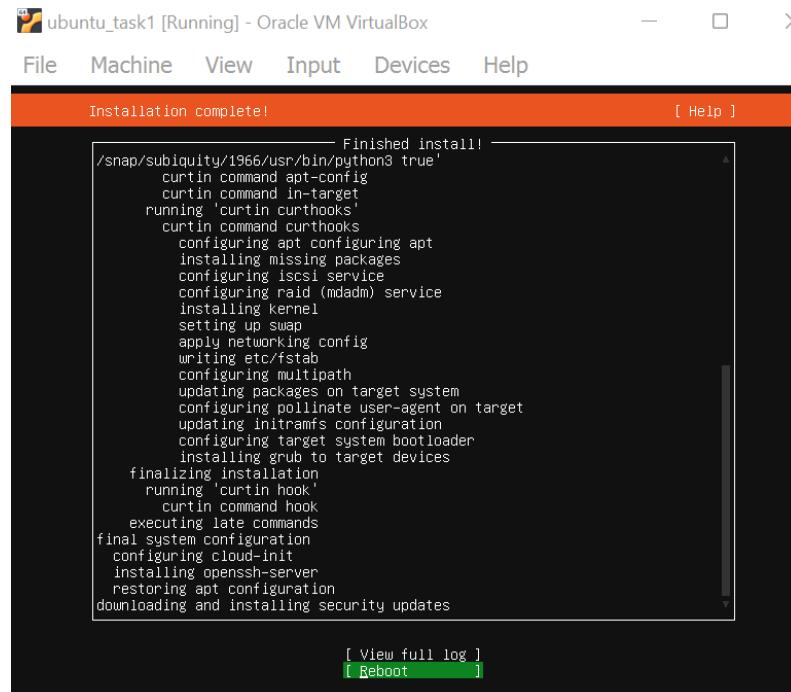
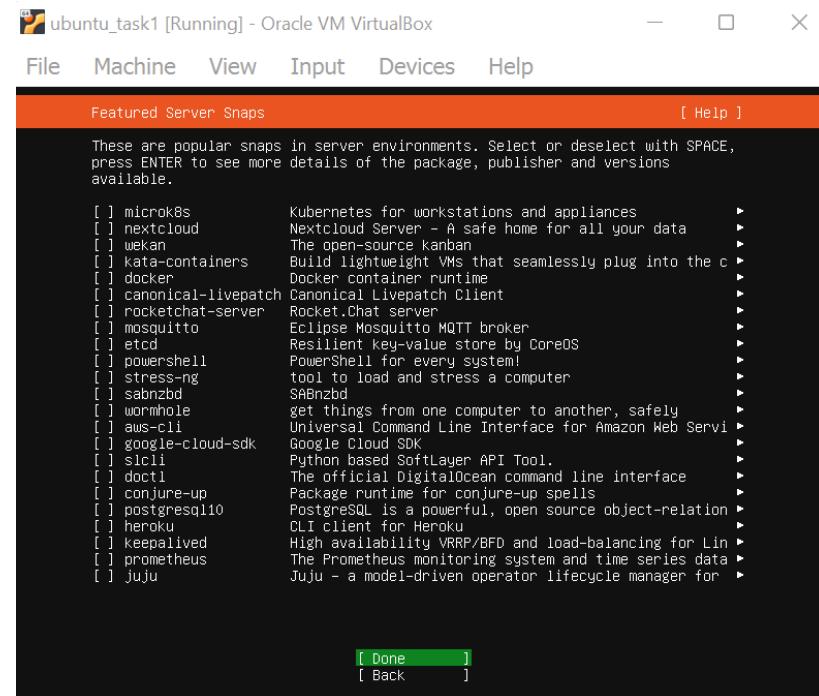
Provide the required parameters as given in the image.

**Step-12:** For SSH setup, select “**Install OpenSSH Server**” option and click on “Done”



**Step-13:** Do not select any packages here as we do not use them and select as “Done”.

**Step-14:** Once the VM installation is complete, click on “Reboot”.



**Step-15:** After Reboot is successful, enter login details of Ubuntu Virtual Machine as per the image shown.

```
ubuntu 18.04.5 LTS webserver tty1
webserver login: vboxuser
Password:
Login incorrect
webserver login: chintsa
Password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-197-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Tue Nov 22 21:00:12 UTC 2022

 System load:  0.19      Processes:          102
 Usage of /:   20.9% of 18.53GB  Users logged in:     0
 Memory usage: 7%
 Swap usage:   0%

65 packages can be updated.
1 update is a security update.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

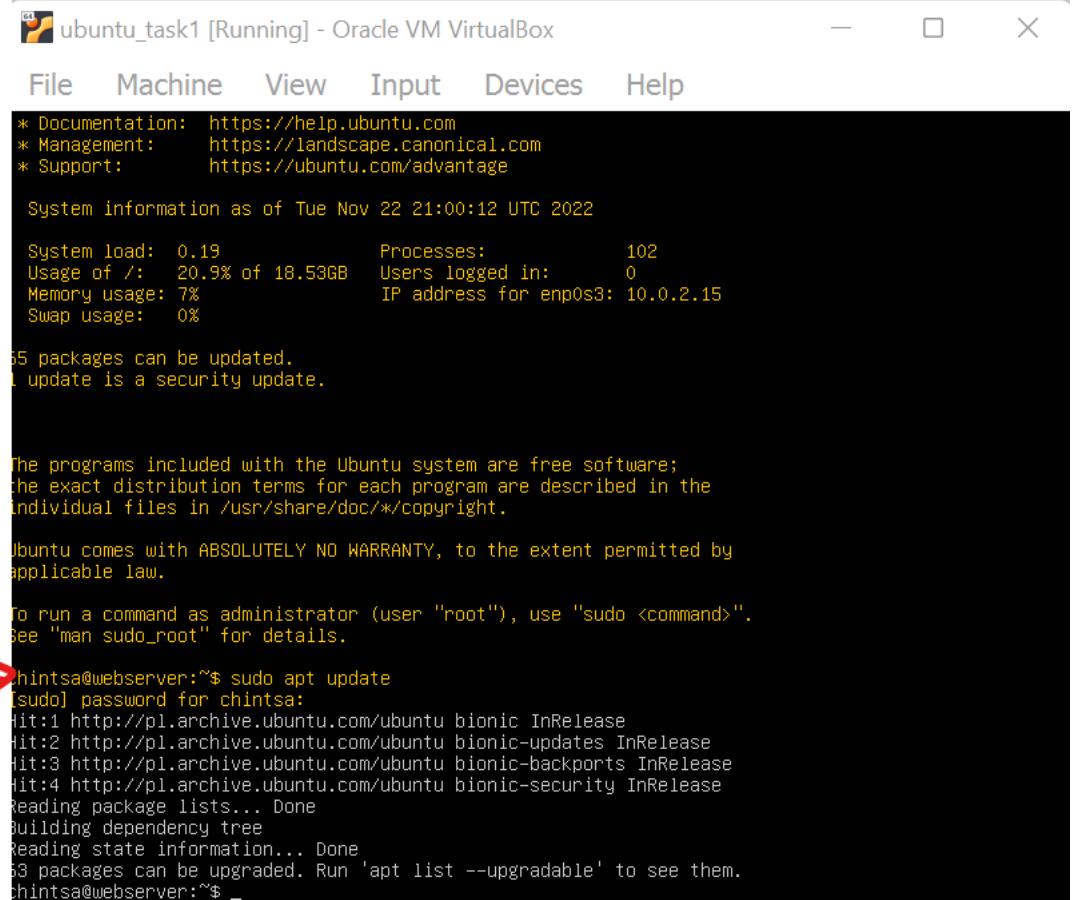
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

chintsa@webserver:~$
```

**Step-16:** Once login to VM is successful, run the update command.

***Sudo apt update***



The screenshot shows a terminal window titled "ubuntu\_task1 [Running] - Oracle VM VirtualBox". The window contains the following text:

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Tue Nov 22 21:00:12 UTC 2022

System load: 0.19          Processes: 102
Usage of /: 20.9% of 18.53GB Users logged in: 0
Memory usage: 7%           IP address for enp0s3: 10.0.2.15
Swap usage: 0%

55 packages can be updated.
1 update is a security update.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

chintsa@webserver:~$ sudo apt update
[sudo] password for chintsa:
Hit:1 http://pl.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://pl.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://pl.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://pl.archive.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
53 packages can be upgraded. Run 'apt list --upgradable' to see them.
chintsa@webserver:~$ _
```

A red arrow points to the line "chintsa@webserver:~\$ sudo apt update".

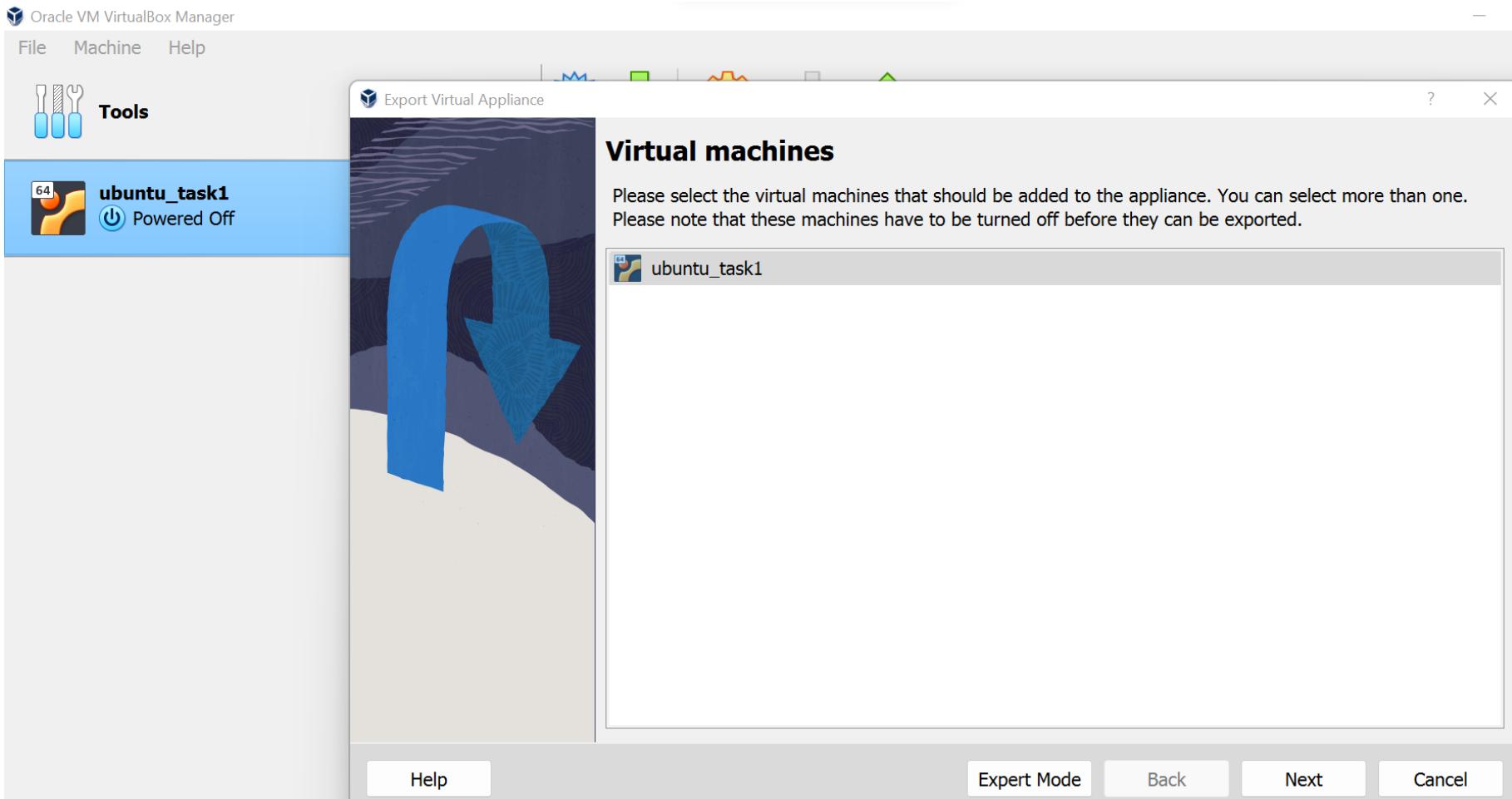
## AWS Migration Assignment - 2

---

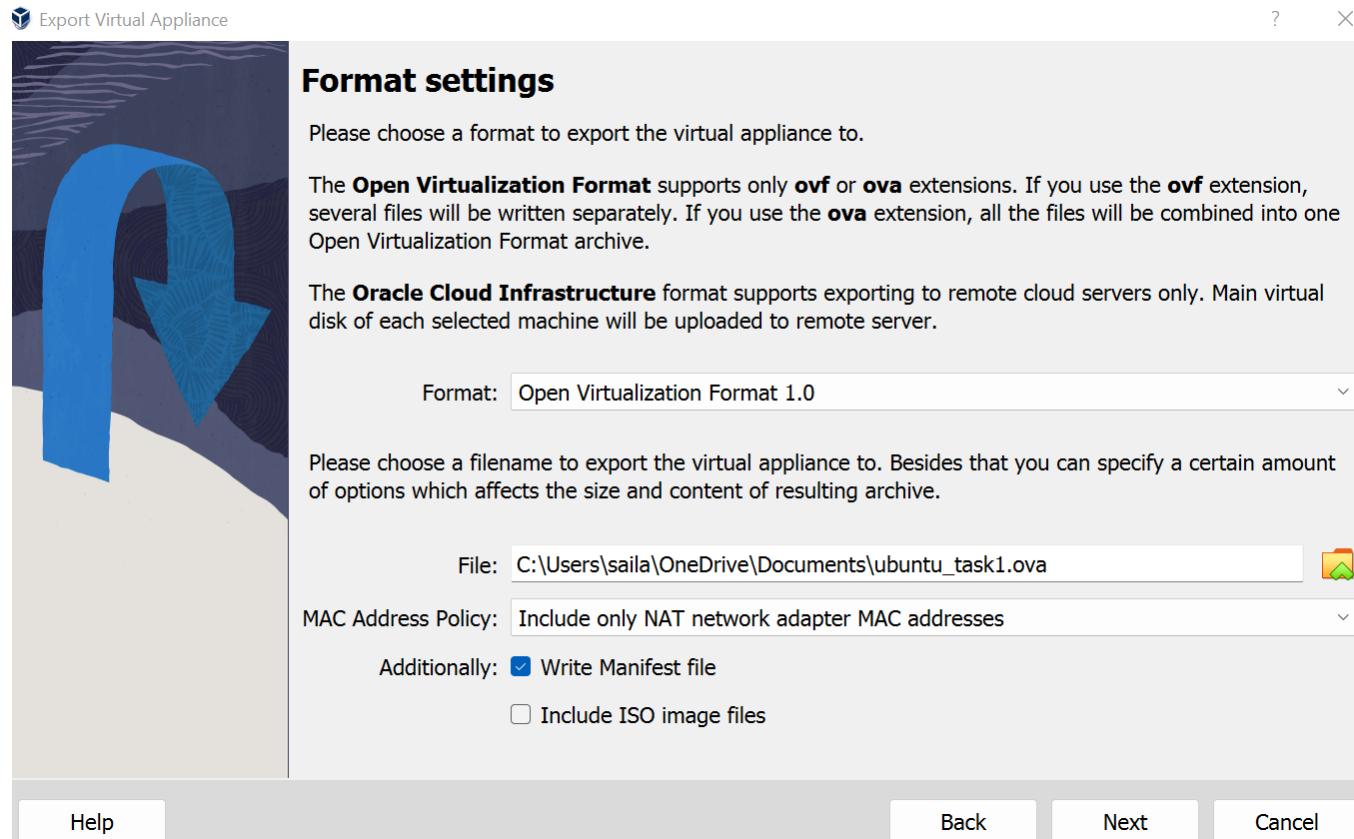
You have been asked to:

1. Export the Ubuntu server VM created in task 1
  2. Upload it to a bucket
  3. Use that VM to create a AMI and create an EC2 instance with it
-

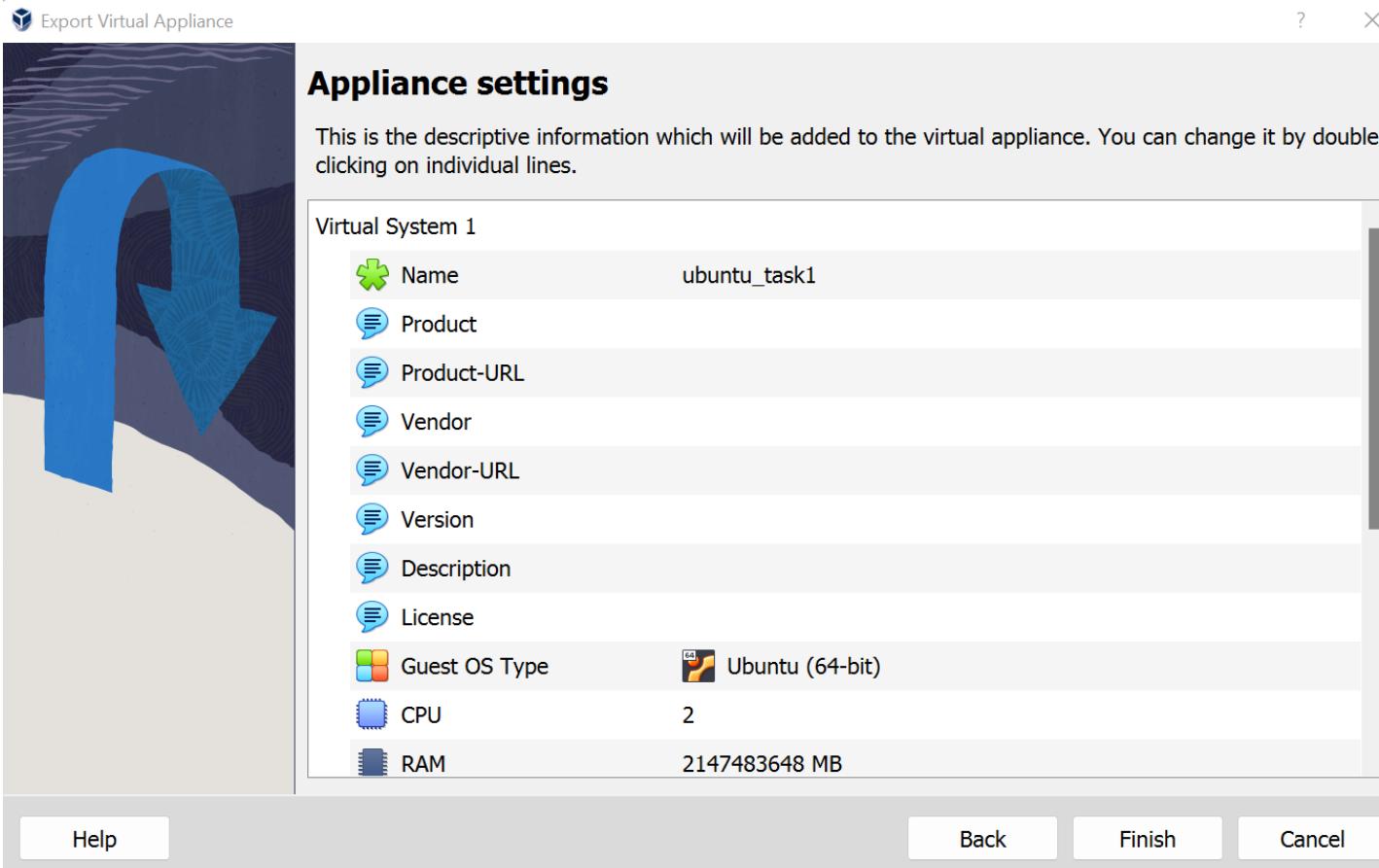
**Step-1: Open Oracle VirtualBox Manager >> Select the Ubuntu VM >> Export Virtual Appliance >> Next**



## Step-2: Specify the location for the file to be saved.



## Step-3: Validate the appliance settings >> Finish



**Step-4:** Now, create an S3 bucket from AWS console.

Specify the S3 bucket configuration as shown here.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. It consists of two main sections: 'General configuration' and 'Object Ownership'.

**General configuration:**

- Bucket name:** s.awsmigration-task2
- AWS Region:** US East (N. Virginia) us-east-1
- Copy settings from existing bucket - optional:** A link to choose a bucket for copying settings.

**Object Ownership:**

- ACLs disabled (recommended):** This option is selected. A note states: "All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies."
- ACLs enabled:** An alternative option where objects can be owned by other accounts.

## Step-5: Uncheck “Block all public access” and “acknowledge the settings”.

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

→  **Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- **Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- **Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- **Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- **Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

→ **⚠ Turning off block all public access might result in this bucket and the objects within becoming public**  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

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

## Step-6: Enable the Bucket versioning and create the bucket.

The screenshot shows the 'Bucket Versioning' configuration step in the AWS S3 Bucket creation wizard. A red arrow points to the 'Enable' radio button, which is selected. Below this section are fields for 'Tags (0) - optional' and 'Default encryption'. The 'Default encryption' section also has a red arrow pointing to the 'Disable' radio button, which is selected. At the bottom of the screen, a red arrow points to the 'Create bucket' button in the top right corner of the main content area.

**Bucket Versioning**  
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning  
 Disable  
 Enable

**Tags (0) - optional**  
You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.  
[Add tag](#)

**Default encryption**  
Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption  
 Disable  
 Enable

Amazon S3 > Buckets

**Account snapshot**  
Storage lens provides visibility into storage usage and activity trends. [Learn more](#) [View Storage Lens dashboard](#)

**Buckets (1) [Info](#)**  
Buckets are containers for data stored in S3. [Learn more](#)

[C](#) [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Name	AWS Region	Access	Creation date
s.awsmigration-task2	US East (N. Virginia) us-east-1	Objects can be public	November 22, 2022, 23:08:53 (UTC+01:00)

## Step-7: Upload the “OVA” file saved on your local machine to the S3 bucket.

Upload succeeded  
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

**Summary**

Destination	Succeeded	Failed
s3://s.awsmigration-task2	1 file, 1.5 GB (100.00%)	0 files, 0 B (0%)

**Files and folders** (1 Total, 1.5 GB)

Name	Folder	Type	Size	Status	Error
ubuntu_task1.ova	-	application/x-virtualbox-ova	1.5 GB	Succeeded	-

Amazon S3 > Buckets > s.awsmigration-task2 > Upload

**Upload** [Info](#)

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](#)

Drag and drop files and folders you want to upload here, or choose **Add files**, or **Add folders**.

**Files and folders** (1 Total, 1.5 GB)

All files and folders in this table will be uploaded.

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	ubuntu_task1.ova	-	application/x-virtualbox-ova	1.5 GB

**Destination**

Destination  
[s3://s.awsmigration-task2](#)

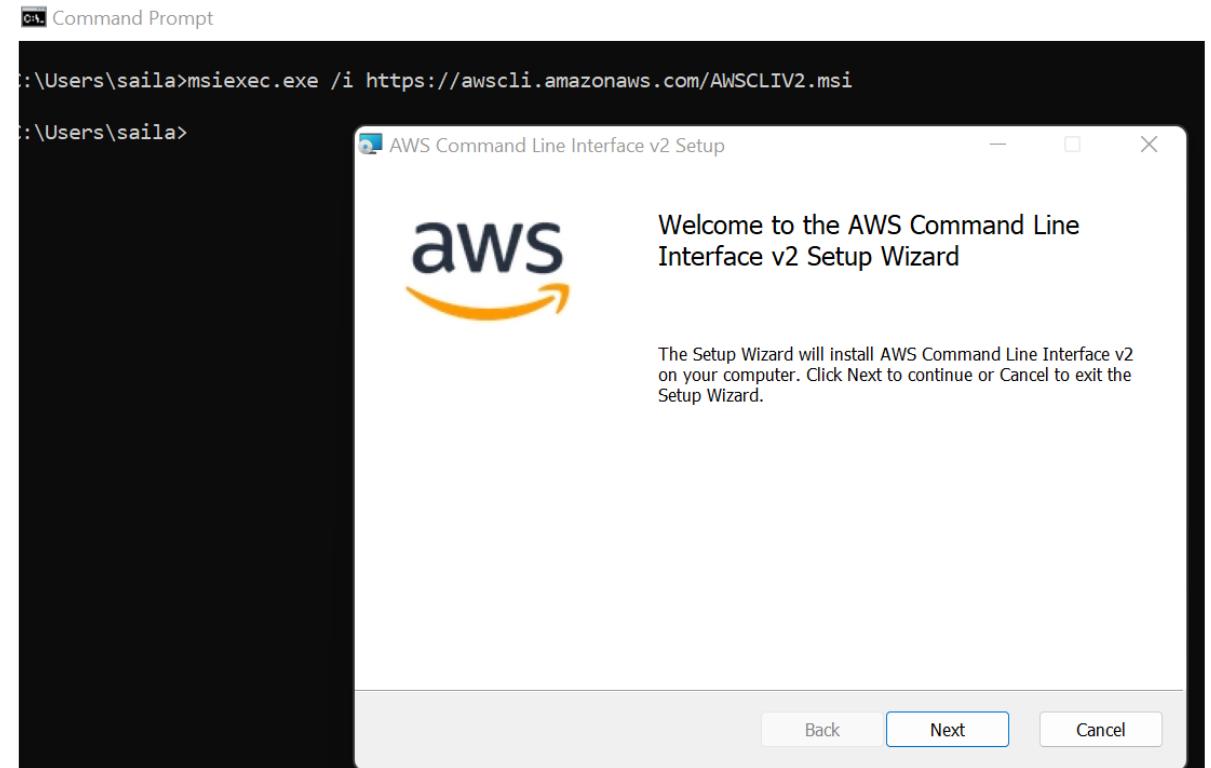
**Destination details**

Bucket settings that impact new objects stored in the specified destination.

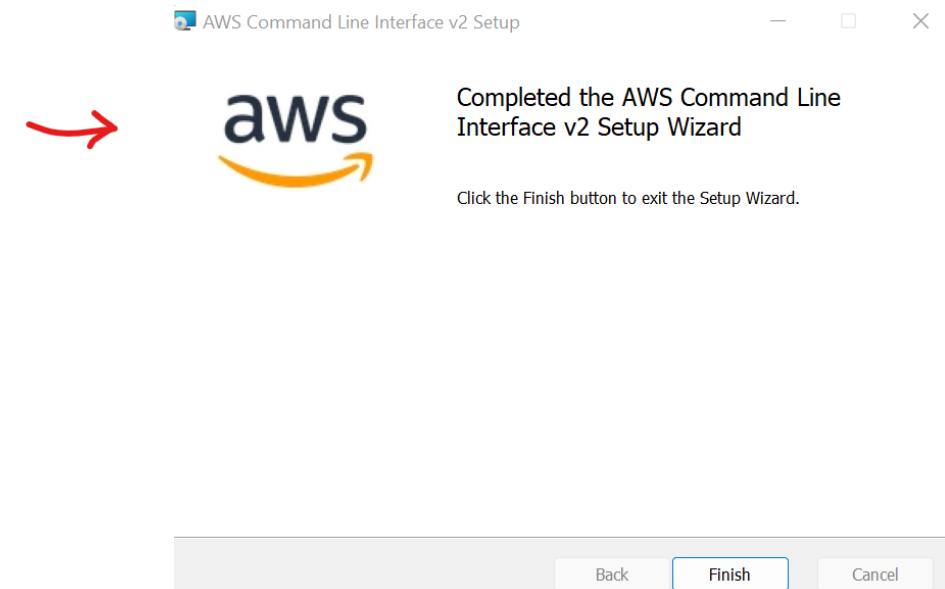
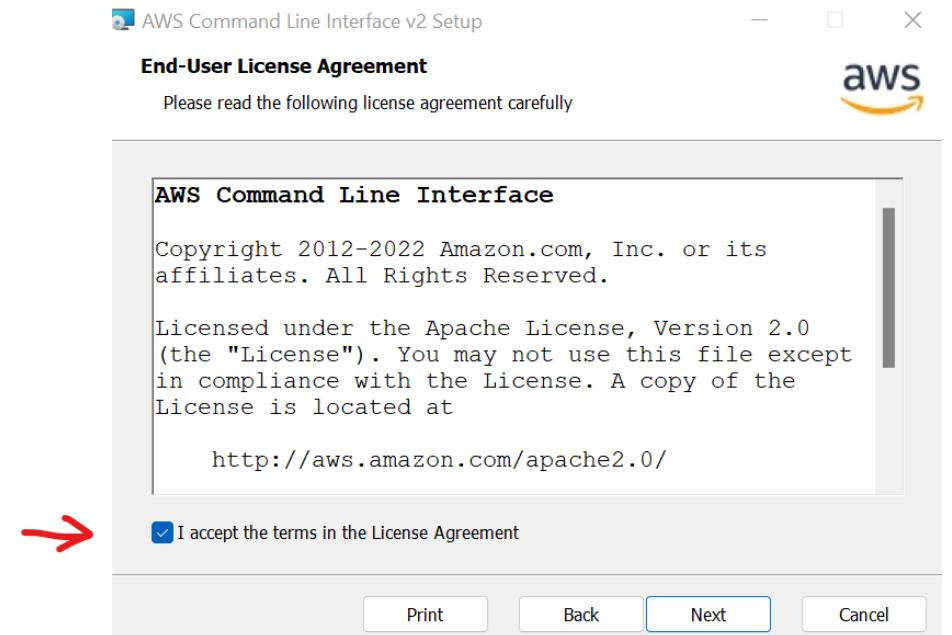
**Step-8:** Now install “AWS CLI” on your laptop using the below commands.

Open CMD >>

msiexec.exe /i <https://awscli.amazonaws.com/AWSCLIV2.msi>



## Step-9: Accept to the EULA and complete the installation of AWS CLI.



## Summary

User ARN arn:aws:iam::537907379462:user/ovatask2 

Path /

Creation time 2022-11-23 00:00 UTC+0100

Permissions Groups (1) Tags Security credentials Access Advisor

▼ Permissions policies (1 policy applied)

Add permissions

Policy name ▾

Attached from group

▶  AdministratorAccess

## Step-10: Create an IAM role to access the image from S3 bucket on your local machine.

### Admins

#### Summary

User group name

Admins

Creation time

November 22, 2022, 23:56 (UTC+01:00)

[Users](#)

[Permissions](#)

[Access Advisor](#)

#### Users in this group (1)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

 Search

User name 

ovatask2

### Add user

1 2 3 4 5



#### Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://537907379462.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key	Email login instructions
	ovatask2	AKIA2POHHUDMAMNKDGE	***** Show	<a href="#">Send email</a>

**Step-11:** Open CMD >> aws configure >> enter aws access key id >> aws secret access key >> default region name >>output format as shown in image.

Region is “us-east-1” based on S3 location created

**Note:** Get AWS access key id and key from IAM user created from Step-10

**Step-12:** Create “**trust-policy.json**” file on your local computer where AWS CLI is installed using below code.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": { "Service": "vmie.amazonaws.com" },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals":{
                    "sts:ExternalId": "vmimport"
                }
            }
        }
    ]
}
```

```
C:\Users\saila>aws configure
AWS Access Key ID [*****MWPE]: AKIAJ2POHHUDMAMNKDGE
AWS Secret Access Key [*****JzY8]: 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
Default region name [us-east-2]: us-east-1
Default output format [json]: json
```

```
C:\Users\saila>aws s3 ls
2022-11-22 23:08:53 s.awsmigration-task2
```

**C:\Users\saila>aws iam create-role --role-name vmimport --assume-role-policy-document file://C:\Users\saila\OneDrive\Documents\trust-policy.json**

```
C:\Users\saila>aws iam create-role --role-name vmimport --assume-role-policy-document file://C:\Users\saila\OneDrive\Documents\trust-policy.json
{
    "Role": {
        "Path": "/",
        "RoleName": "vmimport",
        "RoleId": "AROAX2POHHUDAT5XNEQYP",
        "Arn": "arn:aws:iam::537907379462:role/vmimport",
        "CreateDate": "2022-11-22T23:18:20+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "vmie.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole",
                    "Condition": {
                        "StringEquals": {
                            "sts:ExternalId": "vmimport"
                        }
                    }
                }
            ]
        }
    }
}
```

**Step-13:** Create “role-policy.json” using the below code by updating with S3 name as highlighted in red and execute on CMD.

```
C:\Users\saila>aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://C:\Users\saila\OneDrive\Documents\role-policy.json
```

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucket",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": [  
                "arn:aws:s3:::s.awsmigration-task2"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::s.awsmigration-task2/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:ModifySnapshotAttribute",  
                "ec2:CopySnapshot",  
                "ec2:RegisterImage",  
                "ec2:Describe"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

**C:\Users\saila>aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://C:\Users\saila\OneDrive\Documents\role-policy.json**

**Step-14:** Create “containers.json” file using the below code by updating Description, S3 bucket name and S3 key

```
C:\Users\saila>aws ec2 import-image --description "ubuntu_task1" --license-type BYOL --disk-containers file://C:\Users\saila\OneDrive\Documents\containers.json  
[  
  {  
    "Description": "ubuntu_task1 OVA",  
    "Format": "ova",  
    "UserBucket": {  
      "S3Bucket": "s.awsmigration-task2",  
      "S3Key": "ubuntu_task1.ova"  
    }  
  }]
```

```
C:\Users\saila>aws ec2 import-image --description "ubuntu_task1" --license-type BYOL --disk-containers file://C:\Users\saila\OneDrive\Documents\containers.json  
{  
  "Description": "ubuntu_task1",  
  "ImportTaskId": "import-ami-0fff00189c07cee3f",  
  "LicenseType": "BYOL",  
  "Progress": "1",  
  "SnapshotDetails": [  
    {  
      "Description": "ubuntu_task1 OVA",  
      "DiskImageSize": 0.0,  
      "Format": "OVA",  
      "UserBucket": {  
        "S3Bucket": "s.awsmigration-task2",  
        "S3Key": "ubuntu_task1.ova"  
      }  
    }  
  ],  
  "Status": "active",  
  "StatusMessage": "pending"  
}
```

**Step-15:** Check the status of import image by using the below command.

**C:\Users\saila>aws ec2 describe-import-image-tasks --import-task-ids import-ami-0fff00189c07cee3f**

```
C:\Users\saila>aws ec2 describe-import-image-tasks --import-task-ids import-ami-0fff00189c07cee3f
{
    "ImportImageTasks": [
        {
            "Architecture": "x86_64",
            "Description": "ubuntu_task1",
            "ImageId": "ami-07a06184c17d5a014",
            "ImportTaskId": "import-ami-0fff00189c07cee3f",
            "LicenseType": "BYOL",
            "Platform": "Linux",
            "SnapshotDetails": [
                {
                    "DeviceName": "/dev/sda1",
                    "DiskImageSize": 1649886208.0,
                    "Format": "VMDK",
                    "SnapshotId": "snap-019261ed5c7113e4e",
                    "Status": "completed",
                    "UserBucket": {
                        "S3Bucket": "s.awsmigration-task2",
                        "S3Key": "ubuntu_task1.ova"
                    }
                }
            ],
            "Status": "completed",
            "Tags": [],
            "BootMode": "legacy_bios"
        }
    ]
}
```

## Step-16: Now from AWS console >> EC2 >> AMIs >> Check the imported AMI listed as shown below

The screenshot shows the AWS EC2 service interface. On the left, a sidebar displays 'New EC2 Experience' and links to 'EC2 Dashboard', 'EC2 Global View', and 'Events'. The main navigation bar includes 'Services' (selected), 'Search', and tabs for 'IAM', 'EFS', 'EC2' (selected), 'VPC', 'Route 53', 'RDS', 'Lambda', 'S3', 'Simple Queue Service', 'Elastic Beanstalk', and 'OpsWorks'. The top status bar shows the URL 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Images:visibility=owned-by-me' and various browser icons.

The central area displays the 'Amazon Machine Images (AMIs)' list. A table shows one item:

Name	AMI ID	AMI name	Source	Owner	Visibility	Status	Creation date
-	ami-07a06184c17d5a014	import-ami-0fff00189c07cee3f	537907379462/import-ami-0fff00189c...	537907379462	Private	Available	2022/11/23 00:46 GM

Below the table, a breadcrumb trail reads 'EC2 > Instances > Launch an instance'. The 'Launch an instance' wizard is open, showing the 'Name and tags' step. A text input field contains 'ubuntuvm-task2', and a link 'Add additional tags' is visible.

The 'Application and OS Images (Amazon Machine Image)' section is expanded, showing a search bar and a catalog of AMIs. One item is listed:

Amazon Machine Image (AMI)  
import-ami-0fff00189c07cee3f  
ami-07a06184c17d5a014

Filter options at the bottom include 'Published', 'Architecture', 'Virtualization', 'Root device type', and 'ENI Enabled'.

## Step-17: Check the instance in EC2 which is created based on AMI.

Instances (1) <a href="#">Info</a>											
		<a href="#">C</a> <a href="#">Connect</a> <a href="#">Instance state ▾</a> <a href="#">Actions ▾</a> <a href="#">Launch instances</a> ▾									
		<a href="#"> Find instance by attribute or tag (case-sensitive)</a> < 1 > <a href="#"></a>									
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP		
ubuntuvm-task2	<a href="#">i-0331f3b90655256ce</a>	Running		t2.micro	-	No alarms	us-east-1d	ec2-3-91-225-174.com...	3.91.225.174	-	

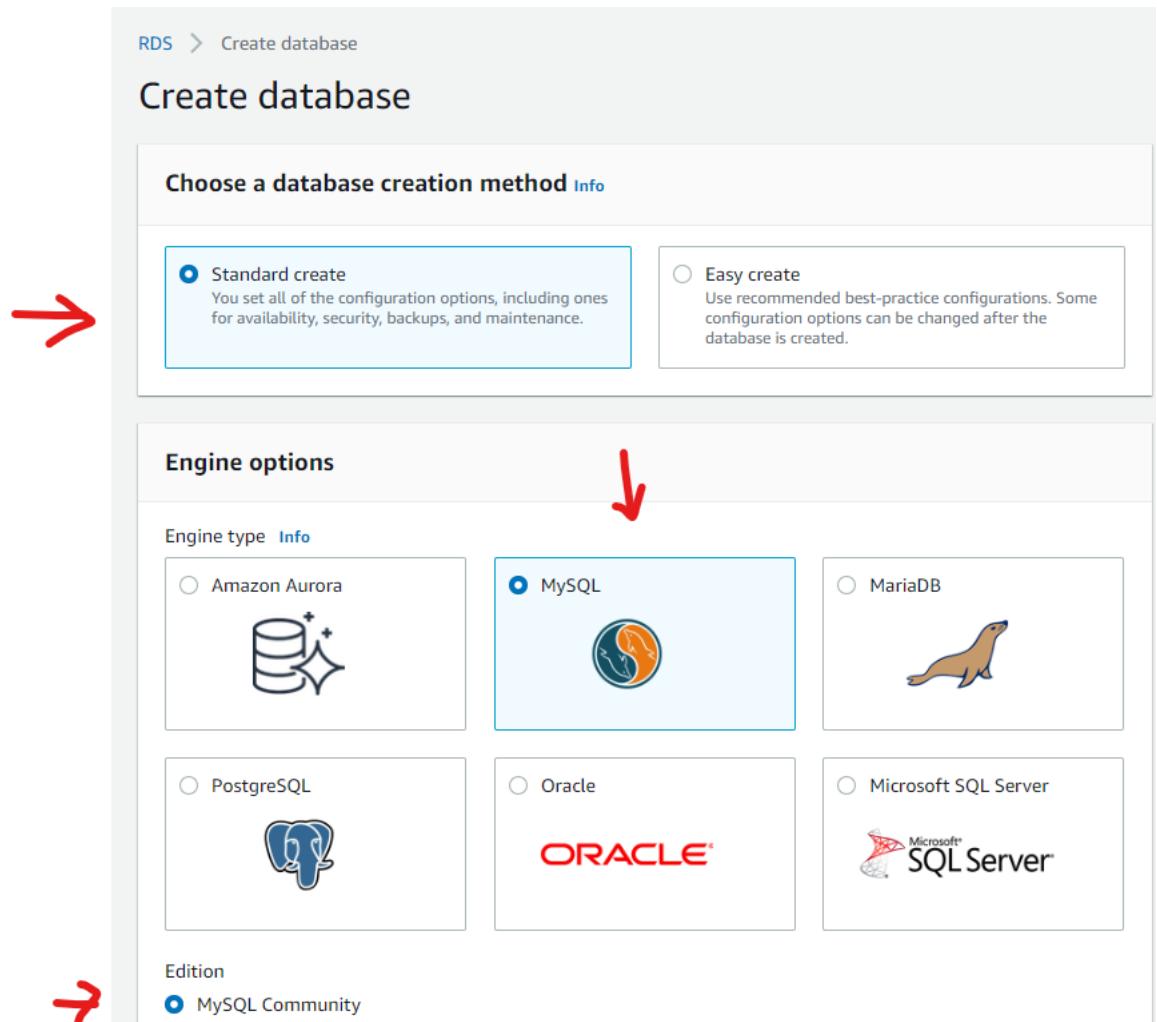
## AWS Migration Assignment - 3

---

You have been asked to:

1. Launch an RDS MySQL database. Login and insert some data into it.
  2. Use Database Migration System to migrate the MySQL database into an RDS PostgreSQL database
-

## Step-1: Create “MySQL” RDS DB.



## Step-2: Choose “Free tier” and the default engine version.

The screenshot shows the 'Engine Version' dropdown set to 'MySQL 8.0.28'. A red arrow points to this dropdown. Below it, the 'Templates' section is shown with three options: 'Production', 'Dev/Test', and 'Free tier'. The 'Free tier' option is selected, indicated by a blue circle and a red arrow pointing to its 'Info' link. The 'Availability and durability' section follows, featuring 'Deployment options' with 'Multi-AZ DB Cluster - new' selected. Other options include 'Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)' and 'Single DB instance (not supported for Multi-AZ DB cluster snapshot)'. Each deployment option has a corresponding 'Info' link.

Engine Version

MySQL 8.0.28

**Templates**  
Choose a sample template to meet your use case.

Production  
Use defaults for high availability and fast, consistent performance.

Dev/Test  
This instance is intended for development use outside of a production environment.

Free tier  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.  
[Info](#)

**Availability and durability**

Deployment options [Info](#)  
The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB Cluster - *new*  
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a single DB instance with no standby DB instances.

## Step-3: Specify the DB credentials as shown.

**Settings**

**DB instance identifier** [Info](#)  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.  


The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

**Credentials Settings**

**Master username** [Info](#)  
Type a login ID for the master user of your DB instance.  


1 to 16 alphanumeric characters. First character must be a letter.  
 **Auto generate a password**  
Amazon RDS can generate a password for you, or you can specify your own password.

**Master password** [Info](#)  
   
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

**Confirm master password** [Info](#)

**Step-4:** Choose the “**Burstable classes**” as DB instance class and with the Storage settings as shown

**Instance configuration**  
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)**

→ db.t3.micro  
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

**Storage**

Storage type [Info](#)

→ General Purpose SSD (gp2)  
Baseline performance determined by volume size

→ Allocated storage  
20 GiB  
The minimum value is 20 GiB and the maximum value is 6,144 GiB

Storage autoscaling [Info](#)  
Provides dynamic scaling support for your database's storage based on your application's needs.

→  Enable storage autoscaling  
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

## Step-5: Choose the Connectivity and VPC settings as shown.

**Connectivity** [Info](#) C

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) [Info](#)  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-02d757d78577ca754) ▼

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change its VPC.

DB Subnet group [Info](#)  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-02d757d78577ca754 ▼

Public access [Info](#)

Yes  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) [Info](#)  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  Create new

## Step-6: Specify the VPC and Security groups settings as shown in the image.

VPC security group (firewall) [Info](#)  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

Choose existing  
Choose existing VPC security groups

Create new  
Create new VPC security group

Existing VPC security groups  
[Choose one or more options](#) ▾

default [X](#)

Availability Zone [Info](#)  
[No preference](#) ▾

RDS Proxy  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy [Info](#)  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

► Additional configuration

**Database authentication**

Database authentication options [Info](#)

Password authentication  
Authenticates using database passwords.

Password and IAM database authentication  
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication  
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

## Step-7: Specify the VPC and Security groups settings as shown in the image.

**MySQL RDS DB** is created successfully as shown below.

▼ Additional configuration  
Database options, encryption turned off, backup turned off, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name [Info](#) → rdsmysql1db  
If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#) default.mysql8.0

Option group [Info](#) default:mysql-8-0

Backup

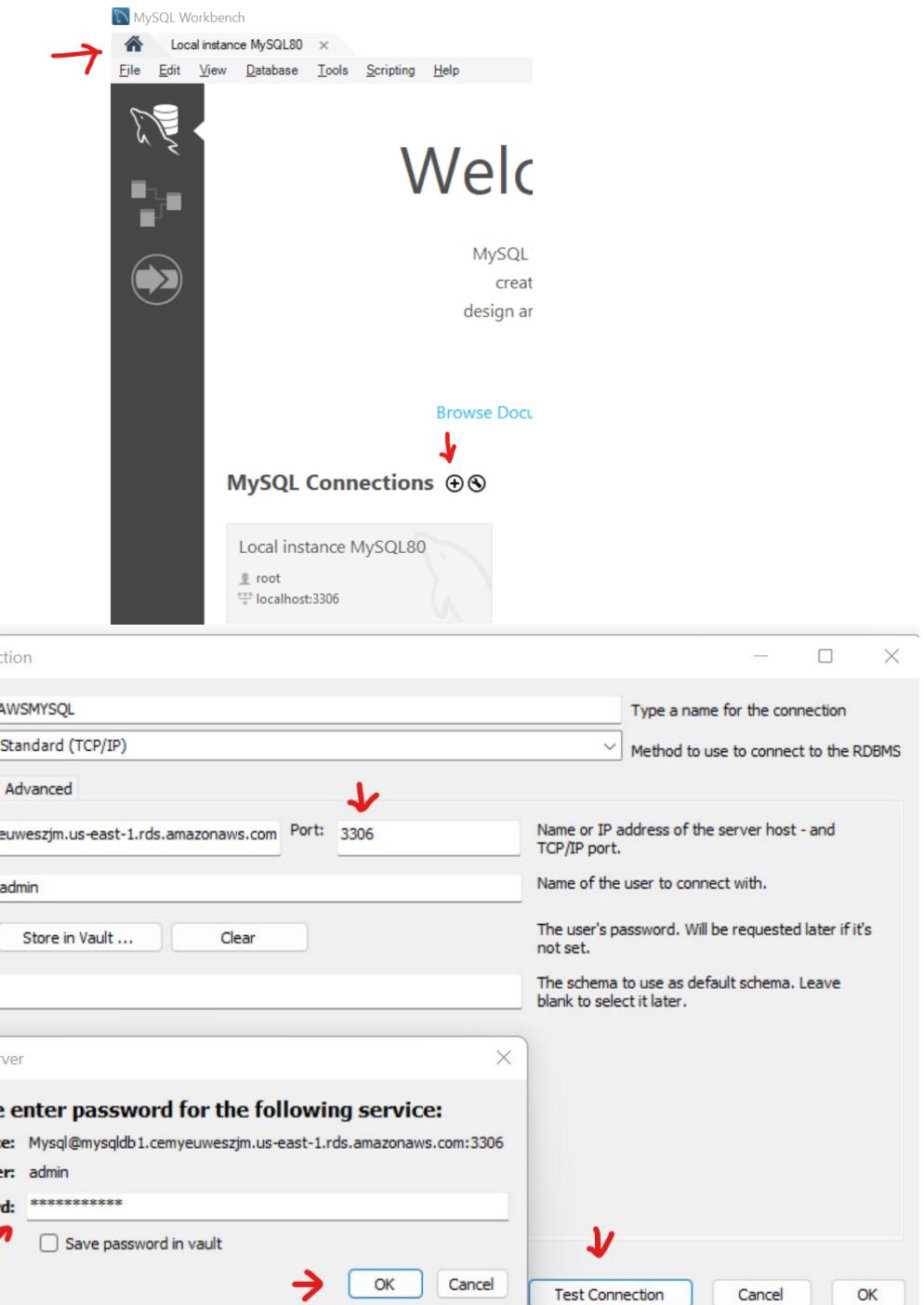
Enable automated backups  
Creates a point-in-time snapshot of your database

Encryption

Enable encryption  
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. [Info](#)

The screenshot shows the AWS RDS Databases page. At the top, there's a navigation bar with 'RDS' and 'Databases'. Below it is a search bar labeled 'Filter by databases'. A red arrow points to the 'Create database' button, which is orange with white text. To its right is a table with columns: DB identifier, Role, Engine, Region & AZ, Size, Status, CPU, Current activity, Maintenance, and VPC. A red arrow points to the 'mysql1db' entry in the DB identifier column. The table also shows the instance type as MySQL Community, region as us-east-1d, size as db.t3.micro, status as Available, and VPC as vpc-02d757d78. There are also buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3', and a back/forward navigation bar.

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC
mysql1db	Instance	MySQL Community	us-east-1d	db.t3.micro	Available	-	none		vpc-02d757d78



**Step-8:** To connect to MySQL RDS DB from local MySQL Workbench, open **MySQL workbench >> Create new connection**.

Specify the New connection details to allow connectivity to RDS MySQL DB from local Workbench and test the connection.

**Step-9: Add a new inbound rule to the security group used for the MySQL RDS by selecting type “MySQL/Aurora” and Source as anywhere.**

Once updated the security group, test the connection on local MySQL workbench.

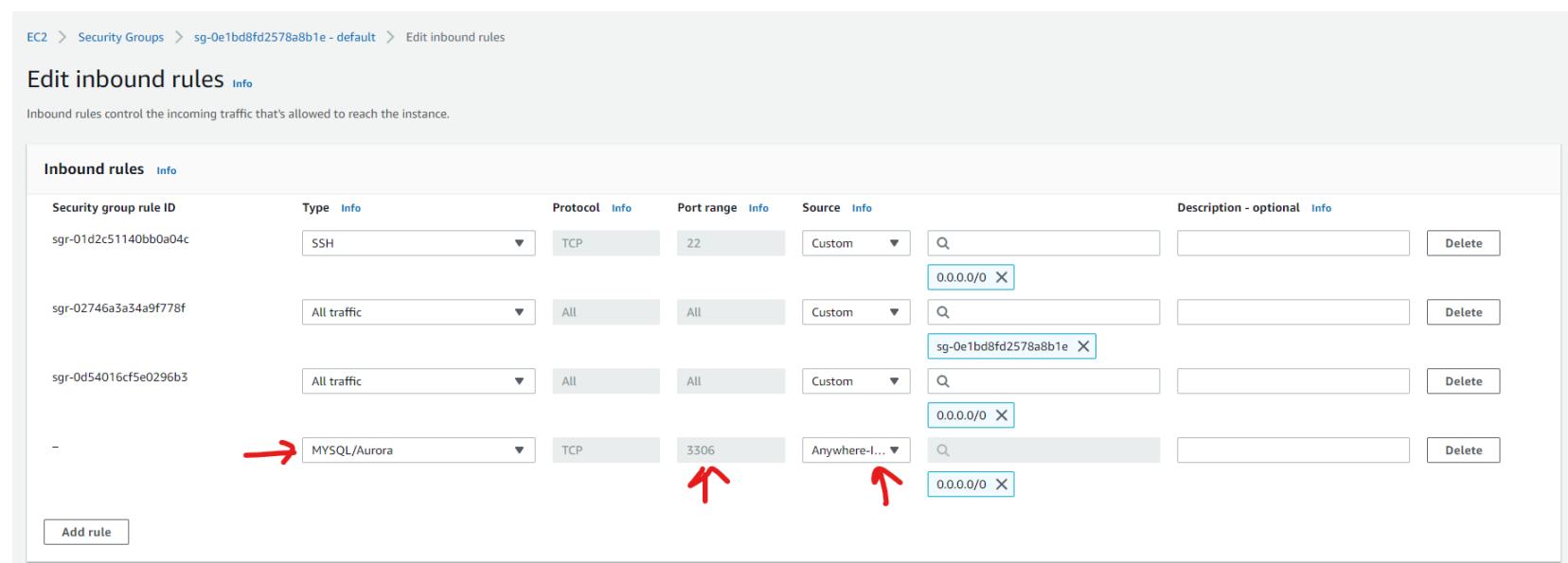
EC2 > Security Groups > sg-0e1bd8fd2578a8b1e - default > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules <a href="#">Info</a>	Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>	Action
sgr-01d2c51140bb0a04c	SSH	TCP	22	Custom	0.0.0.0/0	<a href="#">Delete</a>
sgr-02746a3a34a9f778f	All traffic	All	All	Custom	sg-0e1bd8fd2578a8b1e	<a href="#">Delete</a>
sgr-0d54016cf5e0296b3	All traffic	All	All	Custom	0.0.0.0/0	<a href="#">Delete</a>
-	MYSQL/Aurora	TCP	3306	Anywhere-IP	0.0.0.0/0	<a href="#">Delete</a>

[Add rule](#)



Setup New Connection

Connection Name: AWSMySQL Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: euwesjm.us-east-1.rds.amazonaws.com Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: admin Name of the user to connect with.

Password: User's password. Will be requested later if it's not set.

Default Schema: Schema to use as default schema. Leave blank to select it later.

**MySQL Workbench**

**i Successfully made the MySQL connection**

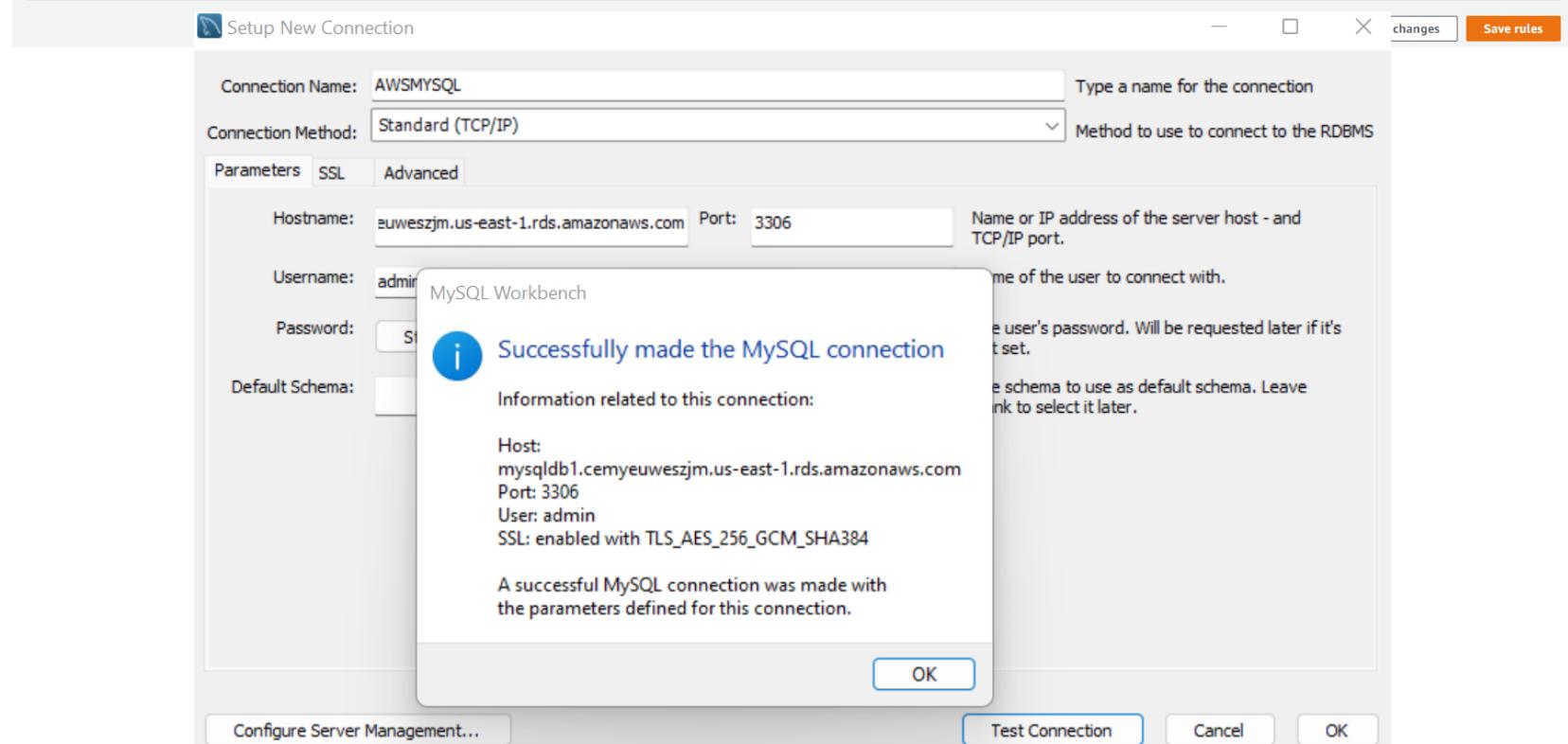
Information related to this connection:

Host: mysqldb1.cemyeuwesjm.us-east-1.rds.amazonaws.com  
Port: 3306  
User: admin  
SSL: enabled with TLS\_AES\_256\_GCM\_SHA384

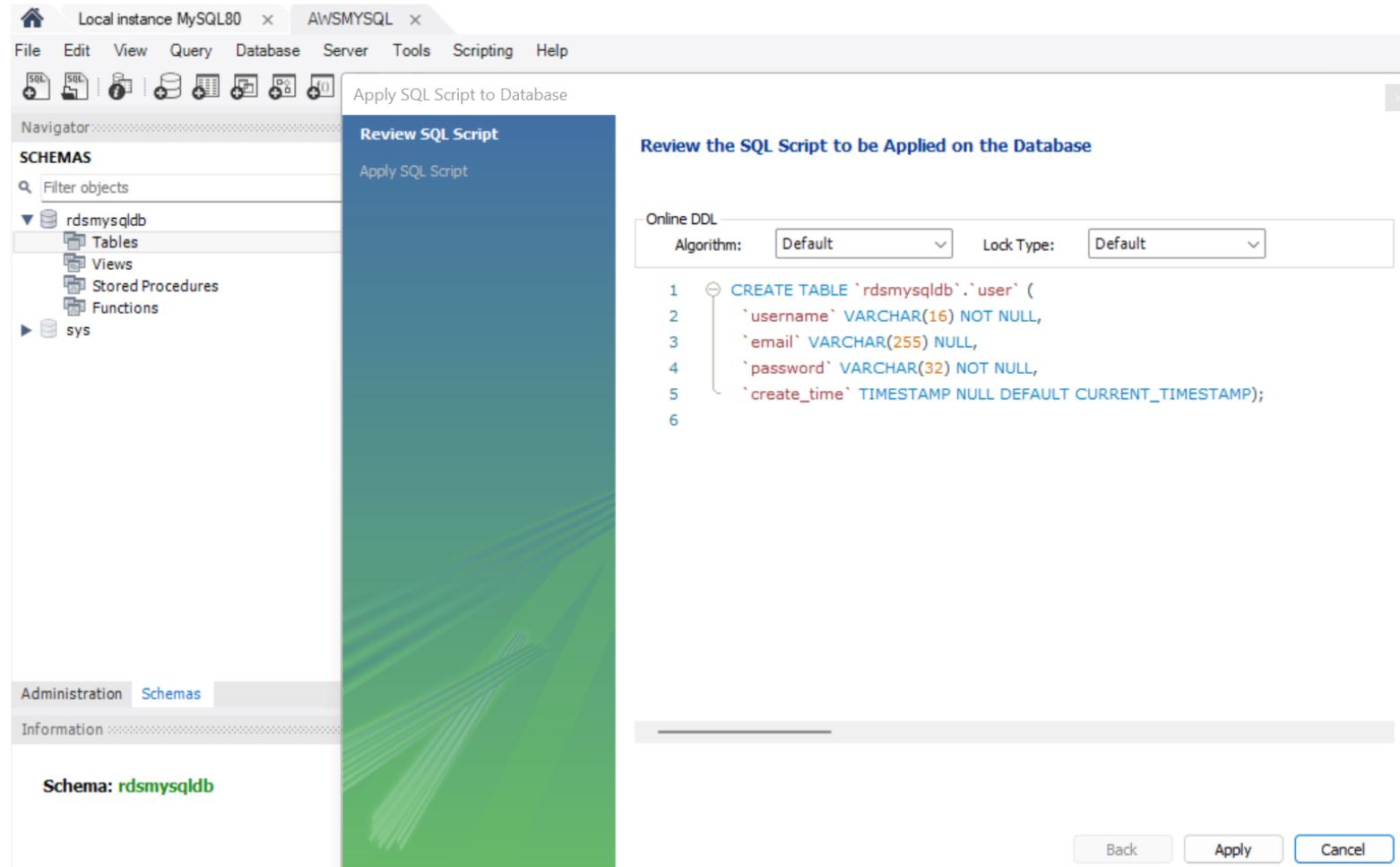
A successful MySQL connection was made with the parameters defined for this connection.

OK

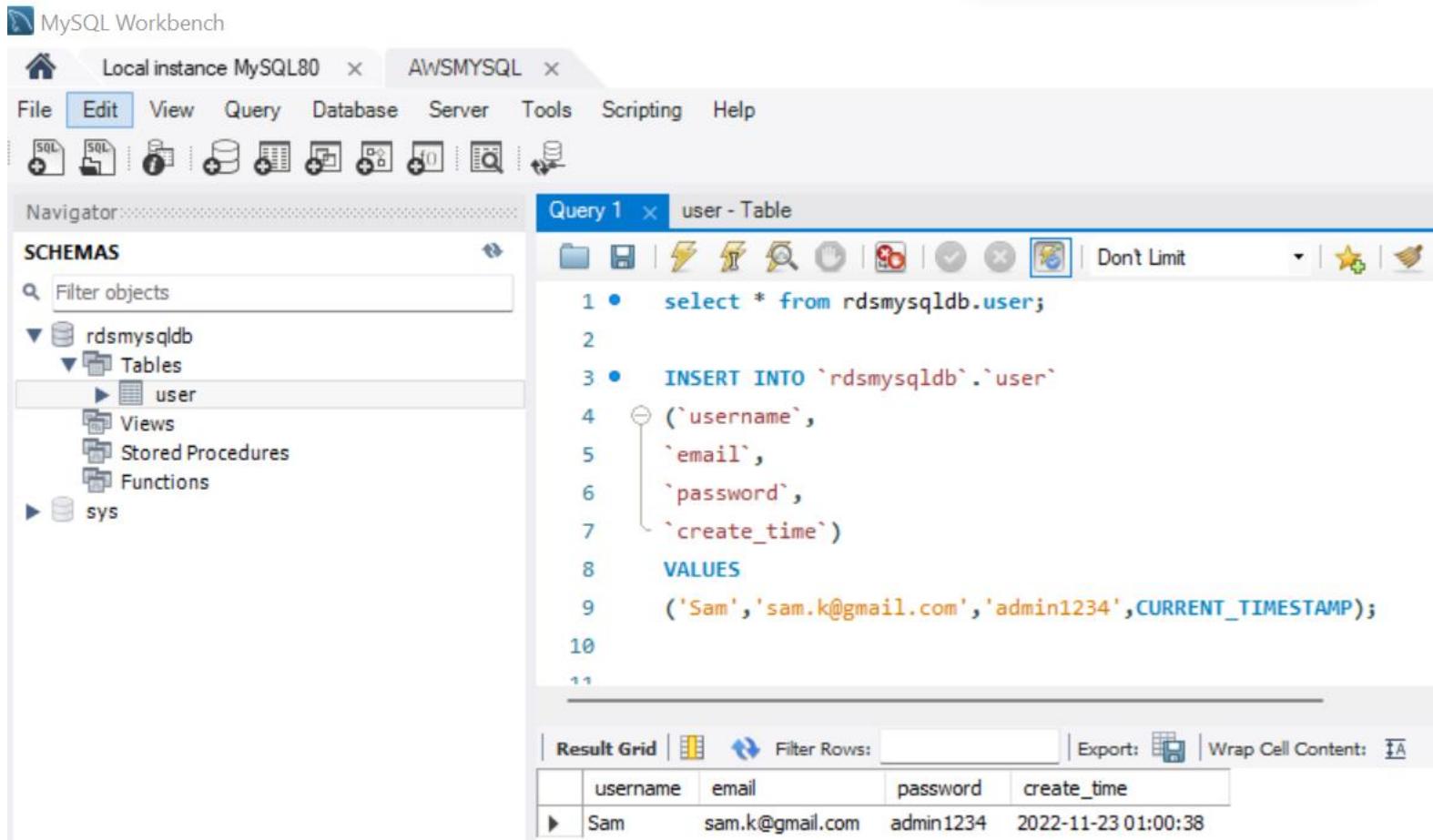
Configure Server Management... Test Connection Cancel OK



## Step-10: Connect to the RDS MySQL DB and Create a new table using the wizard on MySQL Workbench inside this RDS DB.



## Step-11: Insert values in the table created as shown below.



The screenshot shows the MySQL Workbench interface. The title bar says "MySQL Workbench". There are two tabs open: "Local instance MySQL80" and "AWSMySQL". The "Edit" tab is selected. The toolbar has various icons for database management. The "Navigator" pane on the left shows the schema "rdsmysqladb" with a table named "user". The main area is a "Query 1" window titled "user - Table". It contains the following SQL code:

```
1 •   select * from rdsmysqladb.user;
2
3 •   INSERT INTO `rdsmysqladb`.`user`
4     (`username`,
5      `email`,
6      `password`,
7      `create_time`)
8   VALUES
9     ('Sam','sam.k@gmail.com','admin1234',CURRENT_TIMESTAMP);
10
11
```

Below the code, there is a "Result Grid" showing the inserted data:

	username	email	password	create_time
▶	Sam	sam.k@gmail.com	admin1234	2022-11-23 01:00:38

**Step-12:** Now to migrate the RDS MySQL DB to RDS Postgresql DB, navigate to DMS service and create a “replication instance”

The screenshot shows the AWS Database Migration Service (DMS) console. A red arrow points from the top-left towards the 'AWS DMS' logo in the top-left corner of the main window. Another red arrow points from the bottom-left towards the 'Create replication instance' button at the bottom of the left sidebar. A third red arrow points from the bottom-left towards the 'mysqltopostgres' input field in the 'Name' section of the main form.

**AWS DMS**

Welcome to Database Migration Service

## AWS Database Migration Service

Migrate your databases to AWS with minimal downtime

Getting started

To start using AWS Database Migration Service, you need a replication instance which will be used to run your migration tasks.

Create replication instance

DMS > Replication instances > Create replication instance

### Create replication instance

#### Settings

**Name**  
The name must be unique among all of your replication instances in the current AWS region.  
 →

Replication instance name must not start with a numeric value

**Descriptive Amazon Resource Name (ARN) - optional**  
A friendly name to override the default DMS ARN. You cannot modify it after creation.

**Description - optional**  
Replication instance for migrating from MySQL RDS to Postgresql  
 →

The description must only have unicode letters, digits, whitespace, or one of these symbols: \_:/=-+. 1000 maximum character.

## Step-13: Specify the instance configuration as shown for DMS.

**Instance configuration**

**Instance class**  
Choose an availability zone (AZ) where you want your instance to run. The default is "No preference", meaning that AWS DMS will determine which AZ to use. [DMS pricing](#)

dms.t3.medium  
2 vCPUs 4 GiB Memory

Include previous-generation instance classes

**Engine version**  
Choose an AWS DMS version to run on your replication instance. [DMS versions](#)

3.4.7

Include Beta DMS versions

**Upgrades to versions 3.4.7 and higher**  
Upgrades to AWS DMS versions 3.4.7 and higher require that you configure AWS DMS to use VPC endpoints or use public routes. This requirement applies to source and target endpoints for S3, Kinesis, Secrets Manager, DynamoDB, Amazon Redshift, and OpenSearch Service. [Learn more](#)

[View endpoints](#)

**Multi AZ**  
The Multi-AZ option deploys a primary replication instance in one Availability Zone (AZ) and a standby in another AZ. The Single-AZ option deploys a single replication instance in one AZ. Billing is based on DMS pricing.

Dev or test workload (Single-AZ)



## Step-14: Choose the VPC and subnet group for DMS.

**Storage**

Allocated storage (GiB)  
Choose the amount of storage space you want for your replication instance. AWS DMS uses this storage for log files and cached transactions while replication tasks are in progress.

50

**Connectivity and security**

Network type - new | [Info](#)  
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

**IPv4**  
Replication instance with an IPv4 network type that supports IPv4 addressing.

**Dual-stack mode**  
Replication instance with a dual network type that supports both IPv4 and IPv6 addressing.

Virtual private cloud (VPC) for IPv4 | [Info](#)  
Choose the VPC where you want your replication instances to run. It includes VPCs in IPv4 and dual-stack mode.

→ Default VPC (vpc-02d757d78577ca754)

Create a new VPC [\[x\]](#)

Replication subnet group  
Choose a subnet group for your replication instance. The subnet group defines the IP ranges and subnets that your replication instance can use within the VPC you've chosen.

→ default-vpc-02d757d78577ca754

**Public accessible**  
If you choose this option, AWS DMS will assign a public IP address to your replication instance, and you'll be able to connect to databases outside of your VPC.

## Step-15: Choose the VPC security groups and AWS KMS key.

### ▼ Advanced settings

#### Availability zone

Choose an availability zone (AZ) where you want your replication instance to run. The default is "No preference", meaning that will determine which AZ to use.

No Preference

#### VPC security groups

Choose one or more security groups for your replication instances. The security groups specify inbound and outbound rules to network access to your instance.

Choose VPC to select associated VPC security group

default X  
default VPC security group

#### AWS KMS key | [Info](#)

aws/dms

**Step-16:** To create the **dms-vpc-role** IAM role for use with the **AWS CLI**. Create a JSON file with the following IAM policy. Name the JSON file as **dmsAssumeRolePolicyDocument.json**

*aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://dmsAssumeRolePolicyDocument.json*

```
C:\Users\saila>aws iam create-role --role-name dms-vpc-role --assume-role-policy-document file://dmsAssumeRolePolicyDocument.json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "dms.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

**Step-17:** Attach the **AmazonDMSVPCManagementRole** policy to **dms-vpc-role** using the below command.

*aws iam attach-role-policy --role-name dms-vpc-role --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole*

```
C:\Users\saila>aws iam attach-role-policy --role-name dms-vpc-role --policy-arn arn:aws:iam::aws:policy/service-role/AmazonDMSVPCManagementRole
```

**Step-18:** To create the **dms-cloudwatch-logs-role** IAM role for use with the **AWS CLI**. Create a JSON file with the following IAM policy. Name the JSON file as **dmsAssumeRolePolicyDocument2.json**

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "dms.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole"  
        }  
    ]  
}
```

• Create the **IAM role** using the AWS CLI using the below command.

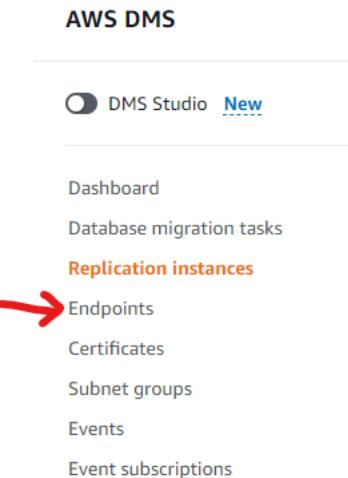
```
aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-document file://dmsAssumeRolePolicyDocument2.json
```

```
C:\Users\saila>aws iam create-role --role-name dms-cloudwatch-logs-role --assume-role-policy-document file://C:\Users\saila\OneDrive\Documents\dmsAssumeRolePolicyDocument2.json  
{  
    "Role": {  
        "Path": "/",  
        "RoleName": "dms-cloudwatch-logs-role",  
        "RoleId": "AROAX2POHHUDCITXTV04V",  
        "Arn": "arn:aws:iam::537907379462:role/dms-cloudwatch-logs-role",  
        "CreateDate": "2022-11-23T01:26:26+00:00",  
        "AssumeRolePolicyDocument": {  
            "Version": "2012-10-17",  
            "Statement": [  
                {  
                    "Effect": "Allow",  
                    "Principal": {  
                        "Service": "dms.amazonaws.com"  
                    },  
                    "Action": "sts:AssumeRole"  
                }  
            ]  
        }  
    }  
}
```

**Step-19:** The DMS Replication instance is created as shown below.

Replication instances (1)										
<input type="text"/> Find replication instance										
	Name	Status	VPC	Class	Engine version	Availability zone	Network type	Public	Public IP address	Private IP address
<input type="checkbox"/>	mysqltopostgres	<span>Available</span>	vpc-02d757d78577ca754	dms.t3.medium	3.4.7	us-east-1c	IPv4	Yes	54.210.29.53	172.31.38.115

**Step-20:** Now on the side panel, select “Endpoints” for AWS DMS migration.



**Step-21:** Choose the **endpoint type** as “**Source**” and enable “**RDS DB instance**” to select the RDS instance as “MySQL RDS” DB created earlier.

The image shows two screenshots of the AWS DMS console. The top screenshot is titled 'Endpoints' and shows a table with columns: Name, Type, Status, Engine, Server name, Port, Migration Hub Mapping, ARN, and Certificate ARN. A red arrow points to the 'Create endpoint' button at the top right. The bottom screenshot is titled 'Create endpoint' and shows the 'Endpoint type' section. It has two options: 'Source endpoint' (selected) and 'Target endpoint'. Below this is a checkbox 'Select RDS DB instance' which is checked. A red arrow points to the 'Select RDS DB instance' checkbox. Another red arrow points to the dropdown menu where 'mysqldb1' is selected under 'RDS Instance'.

DMS > Endpoints

Endpoints (0)

Find endpoint

Name Type Status Engine Server name Port Migration Hub Mapping ARN Certificate ARN

Create endpoint

DMS > Endpoints > Create endpoint

Create endpoint

Endpoint type Info

Source endpoint  
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

Target endpoint  
A target endpoint allows AWS DMS to write data to a database, or to other data source.

Select RDS DB instance

RDS Instance  
Instances available only for current user and region

mysqldb1

## Step-22: Choose the “Access to endpoint database” as “Provide access information manually” and specify the password of the RDS MySQL DB.

Endpoint configuration

Endpoint identifier | [Info](#)  
A label for the endpoint to help you identify it.  
 

Descriptive Amazon Resource Name (ARN) - optional  
A friendly name to override the default DMS ARN. You cannot modify it after creation.

Source engine  
The type of database engine this endpoint is connected to. [Learn more](#) 



Access to endpoint database  
 AWS Secrets Manager  
 Provide access information manually 

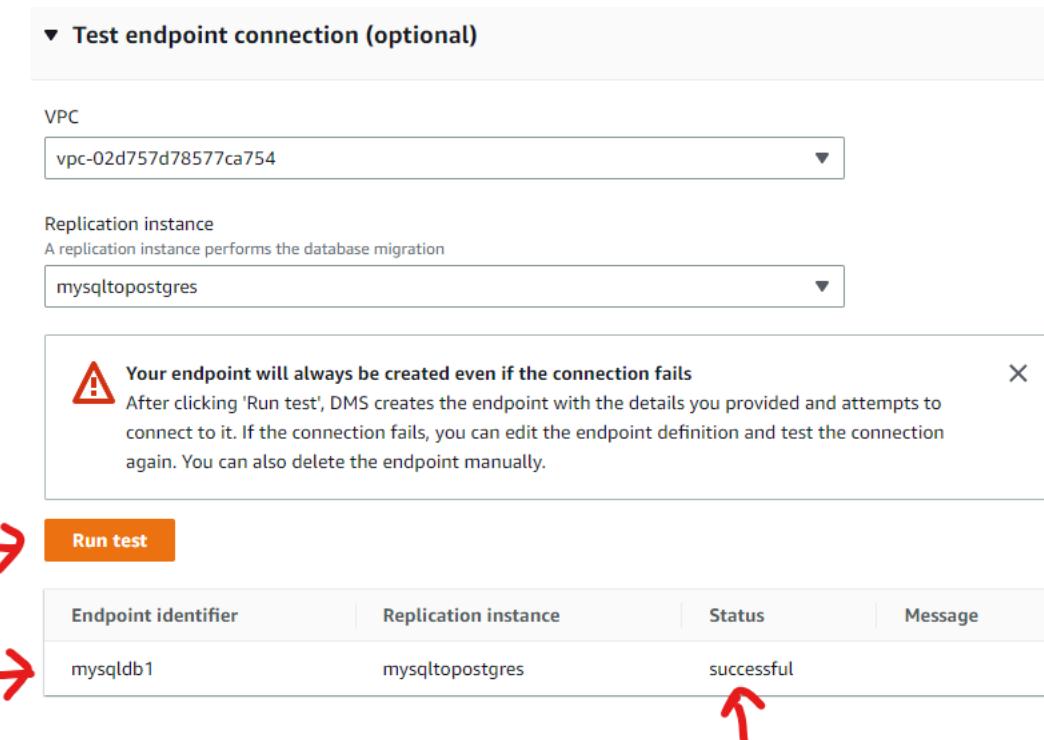
Server name  
The name of the data server for the data provider.

Port  
The port the database runs on for this endpoint.

User name | [Info](#)      Password | [Info](#)  
       

**Step-23:** Test the source endpoint connection to create the endpoint once successful.

The **source endpoint for DMS** is complete.



The screenshot shows the 'Endpoints' list in the AWS DMS console. It displays one endpoint named 'mysqldb1' with the status 'Active'. A red arrow points to the endpoint row.

Name	Type	Status	Engine	Server name	Port	Migration Hub Mapping	ARN
mysqldb1	Source	Active	MySQL	mysqldb1.cemyeuwesjm.us-east-1.rds.amazonaws.com	3306		arn:aws:dms:us-east-1:537907379462:endpoint:KSH2VHIXWO

**Step-24:** Now create another RDS DB for Postgresql engine with required configuration settings as shown.

The screenshot shows the 'Create database' wizard in the AWS RDS console. The first step, 'Choose a database creation method', has 'Standard create' selected. The second step, 'Engine options', shows various database engines: Amazon Aurora, MySQL, MariaDB, PostgreSQL (selected), Oracle, and Microsoft SQL Server. The third step, 'Settings', requires entering a 'DB instance identifier'. A red arrow points to the 'postgresrlds' input field, which is highlighted with a red border. Below it, the 'Master username' is set to 'postgres' and the 'Master password' is masked with dots. A red arrow also points to the 'Master password' input field. The 'Confirm master password' field below it contains the same masked password.

RDS > Create database

## Create database

Choose a database creation method [Info](#)

Standard create  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Amazon Aurora

MySQL

MariaDB

PostgreSQL

Oracle

Microsoft SQL Server

Settings

DB instance identifier [Info](#)  
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the Region.

postgresrlds

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)  
Type a login ID for the master user of your DB instance.

postgres

1 to 16 alphanumeric characters. First character must be a letter.

Auto generate a password  
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)  
.....

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote), @ (at sign).

Confirm master password [Info](#)  
.....

**Step-25:** Choose the storage and instance configuration for Postgresql RDS DB with required configuration settings as shown.

**Instance configuration**  
The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

→ **db.t3.micro**  
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

**Storage**

Storage type [Info](#)

→ **General Purpose SSD (gp2)**  
Baseline performance determined by volume size

Allocated storage  
→ **20** GiB  
The minimum value is 20 GiB and the maximum value is 6,144 GiB

Storage autoscaling [Info](#)  
Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling  
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

## Step-26: Choose the VPC and subnet group settings as needed.

Connectivity [Info](#) C

Compute resource  
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) [Info](#)  
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-02d757d78577ca754) ▼

Only VPCs with a corresponding DB subnet group are listed.

i After a database is created, you can't change its VPC.

DB Subnet group [Info](#)  
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

default-vpc-02d757d78577ca754 ▼

Public access [Info](#)

Yes  
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

No  
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

## Step-27: Choose the VPC security group and verify the port number of the DB.

VPC security group (firewall) [Info](#)  
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

**Choose existing**  
Choose existing VPC security groups

**Create new**  
Create new VPC security group

Existing VPC security groups  
[Choose one or more options](#) ▾

default [X](#)

Availability Zone [Info](#)  
No preference ▾

RDS Proxy  
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

**Create an RDS Proxy** [Info](#)  
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

▼ Additional configuration

Database port [Info](#)  
TCP/IP port that the database will use for application connections.

5432



**Step-28:** Specify a name for database and create.

**RDS DB for Postgresql** is now created successfully.

▼ Additional configuration  
Database options, encryption turned off, backup turned off, backtrack turned off, maintenance, CloudWatch Log protection turned off.

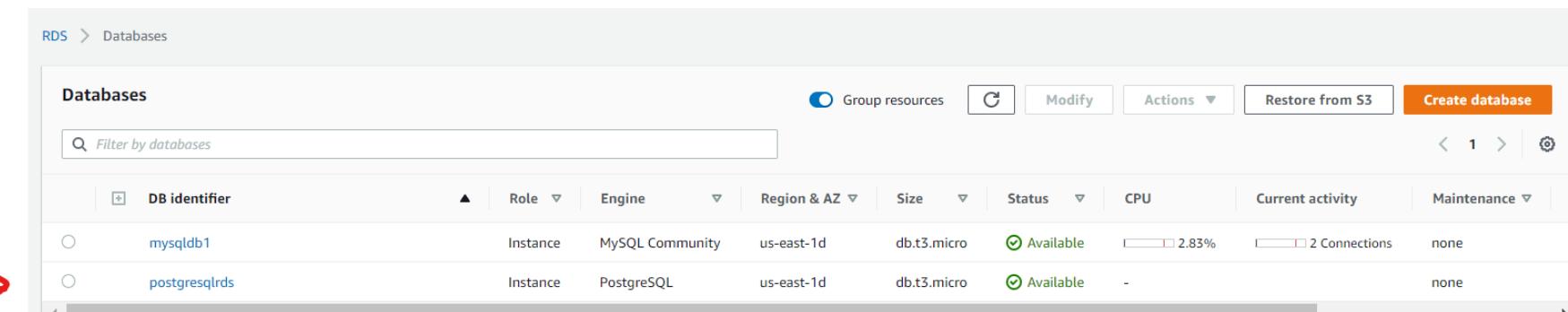
Database options

Initial database name [Info](#)  
 →

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)  
 →

Option group [Info](#)



The screenshot shows the AWS RDS Databases page. At the top, there's a navigation bar with 'RDS > Databases'. Below it is a search bar labeled 'Filter by databases'. A red arrow points to the second database entry in the list.

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance
mysqlDb1	Instance	MySQL Community	us-east-1d	db.t3.micro	Available	2.83%	2 Connections	none
postgreslrds	Instance	PostgreSQL	us-east-1d	db.t3.micro	Available	-	-	none

**Step-29: Now create “target endpoint” for “Postgresql RDS DB” on DMS >> Endpoints.**

Select the Postgresql RDS instance and update the password for the DB.

DMS > Endpoints > Create endpoint

### Create endpoint

Endpoint type [Info](#)

Source endpoint  
A source endpoint allows AWS DMS to read data from a database (on-premises or in the cloud), or from other data source such as Amazon S3.

Target endpoint  
A target endpoint allows AWS DMS to write data to a database, or to other data source.

Select RDS DB instance

RDS Instance  
Instances available only for current user and region

→ postgresqlrds

Endpoint configuration

Endpoint identifier [Info](#)  
A label for the endpoint to help you identify it.  
→ postgresqlrds

Descriptive Amazon Resource Name (ARN) - optional  
A friendly name to override the default DMS ARN. You cannot modify it after creation.  
Friendly-ARN-name

Target engine  
The type of database engine this endpoint is connected to. [Learn more](#)

PostgreSQL

Access to endpoint database

AWS Secrets Manager

Provide access information manually

Server name  
The name of the data server for the data provider.  
postgreslrdscemyeuwesjm.us-east-1.rds.amazonaws.com

Port  
The port the database runs on for this endpoint.  
5432

User name [Info](#)  
postgres

Password [Info](#)  
\*\*\*\*\* ←

**Step-30:** Choose the VPC and test the endpoint connection. Once the test is shown as successful, create the endpoint for target DB.

The screenshot shows the 'Test endpoint connection (optional)' section of the DMS console. It includes fields for 'VPC' (set to 'vpc-02d757d78577ca754') and 'Replication instance' (set to 'mysqltopostgres'). A warning message states: 'Your endpoint will always be created even if the connection fails. After clicking 'Run test', DMS creates the endpoint with the details you provided and attempts to connect to it. If the connection fails, you can edit the endpoint definition and test the connection again. You can also delete the endpoint manually.' Below this is a table with one row, showing the endpoint identifier 'postgresqlrds', replication instance 'mysqltopostgres', status 'successful', and a message column. At the bottom are 'Cancel' and 'Create endpoint' buttons, with a red arrow pointing to the 'Create endpoint' button.

Endpoint identifier	Replication instance	Status	Message
postgresqlrds	mysqltopostgres	successful	

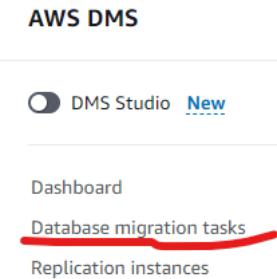
Cancel Create endpoint

**Step-31:** Now we have two endpoints created successfully for both the source and target endpoints of DMS.

The screenshot shows the AWS DMS Endpoints page. A red arrow points to the 'Endpoints (2)' heading. Another red arrow points to the search bar labeled 'Find endpoint'. A third red arrow points to the first endpoint listed, 'mysqlDb1'. The table lists two endpoints with the following details:

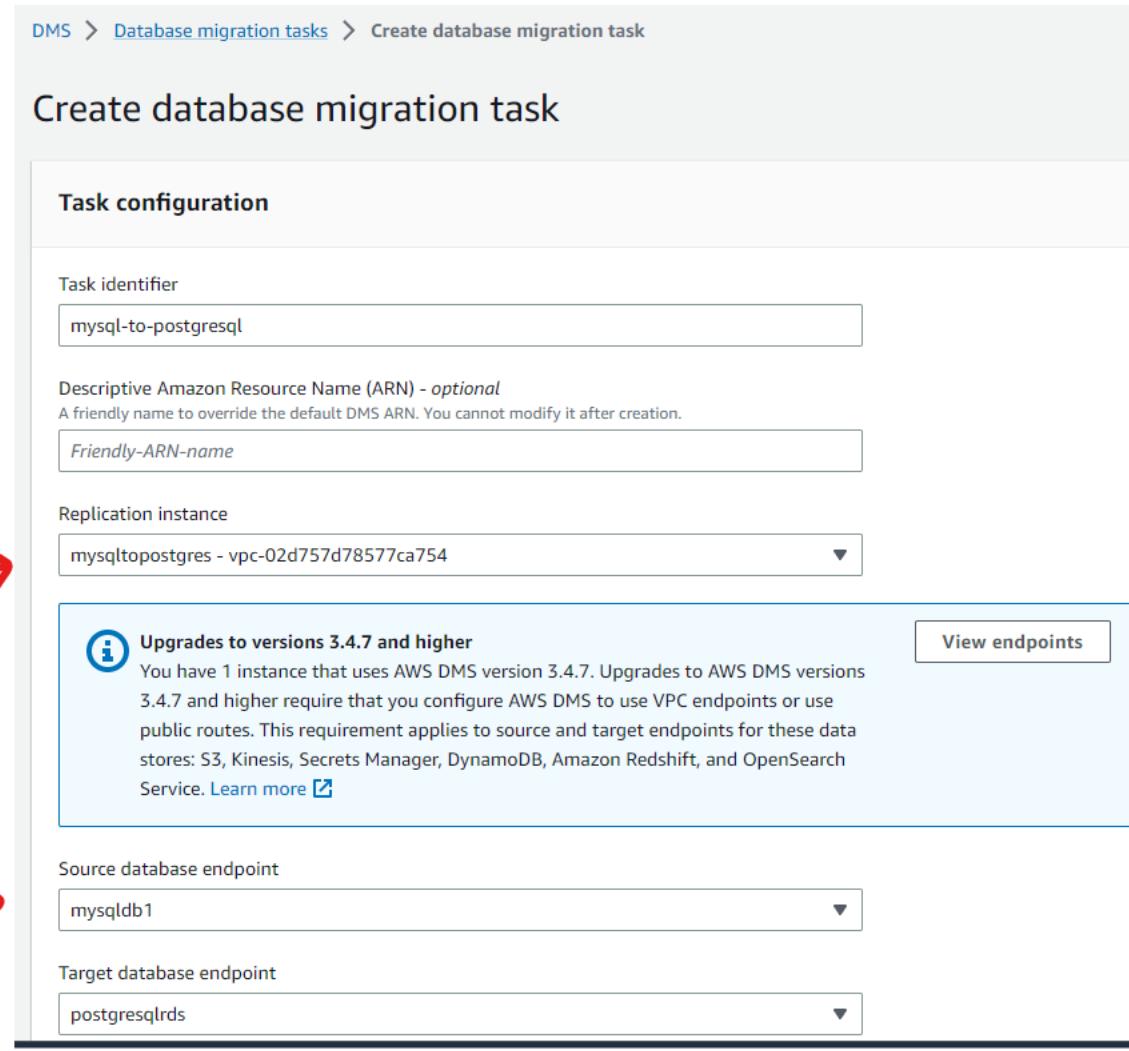
	Name	Type	Status	Engine	Server name	Port	Migration Hub Mapping	ARN
<input type="checkbox"/>	mysqlDb1	Source	Active	MySQL	mysqlDb1.cemyeuwesjm.us-east-1.rds.amazonaws.com	3306		arn:aws:dms:us-east-1:537907379462:endpoint:KSH2
<input type="checkbox"/>	postgresqlrds	Target	Active	PostgreSQL	postgresqlrds.cemyeuwesjm.us-east-1.rds.amazonaws.com	5432		arn:aws:dms:us-east-1:537907379462:endpoint:FFZZ

**Step-32:** Now from AWS DMS left panel >>  
**Database migration tasks** >> **create task**

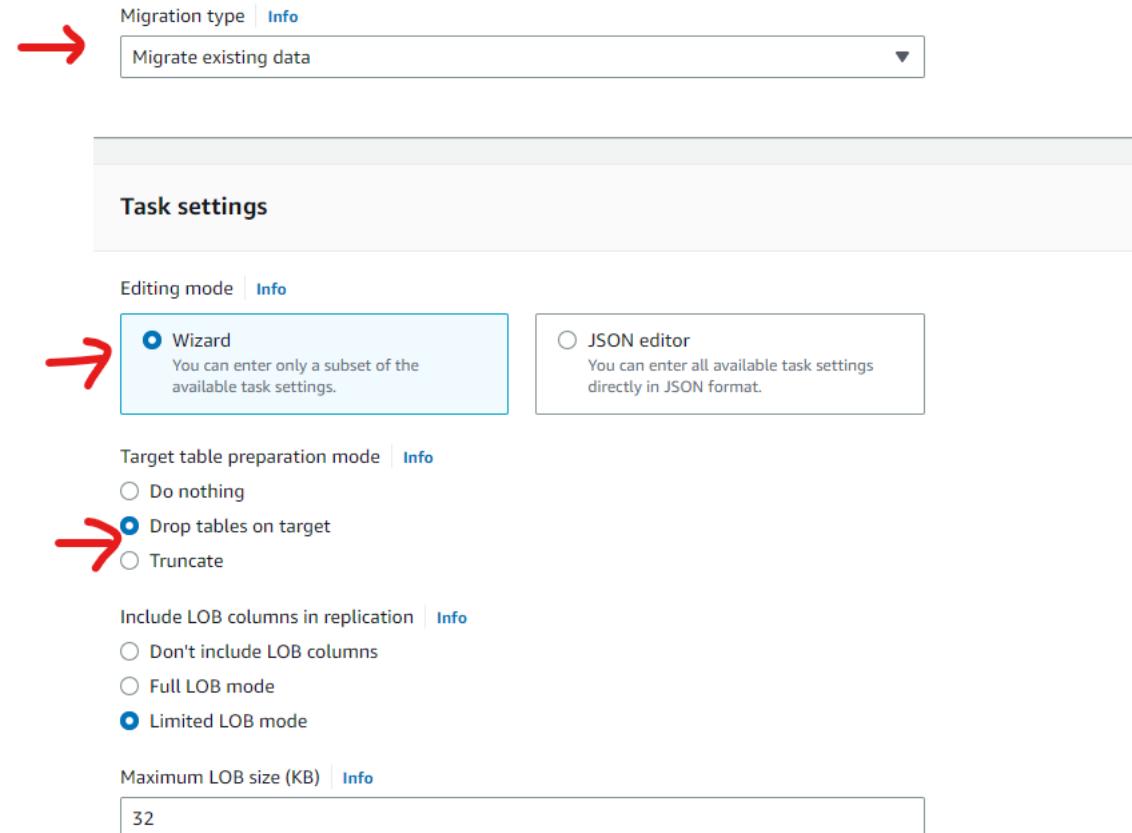


A screenshot of the AWS DMS Database migration tasks page. The URL in the browser is 'DMS &gt; Database migration tasks'. The page shows a table header for 'Database migration tasks (0)'. On the right, there are buttons for 'Actions', 'Quick view and compare', and a prominent orange 'Create task' button. A red arrow points to the 'Create task' button, and another red arrow points to the 'Database migration tasks' link in the navigation bar above.

## Step-33: In the database migration task >> choose the source and target database endpoints as :mysql and postgresql db



**Step-34:** Choose the “Migration type” as “migrate existing data” and from the task settings >> wizard >> drop tables on target.



**Step-35:** In the “selection rules” >> source name : “%” >> source table name >> “%” and “Action” >> “include”.

Table mappings

Editing mode | Info

Wizard  
You can enter only a subset of the available table mappings.

JSON editor  
You can enter all available table mappings directly in JSON format.

Specify at least one selection rule with an include action. After you do this, you can add one or more transformation rules.

▼ Selection rules

Choose the schema and/or tables you want to include with, or exclude from, your migration task. | Info

Add new selection rule

▼ where schema name is like '%' and Source table name is like '%', include

Schema

Enter a schema

Source name

Use the % character as a wildcard

%

Source table name

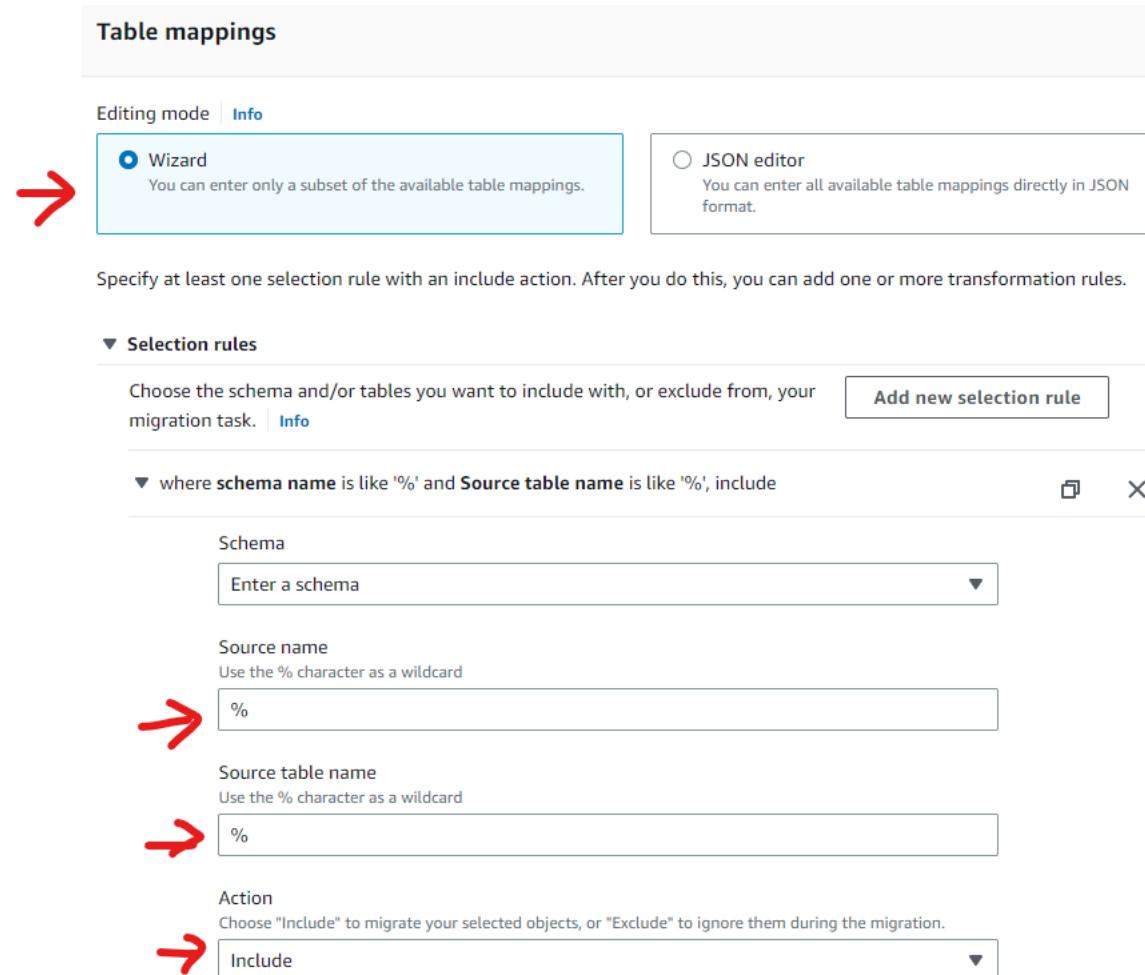
Use the % character as a wildcard

%

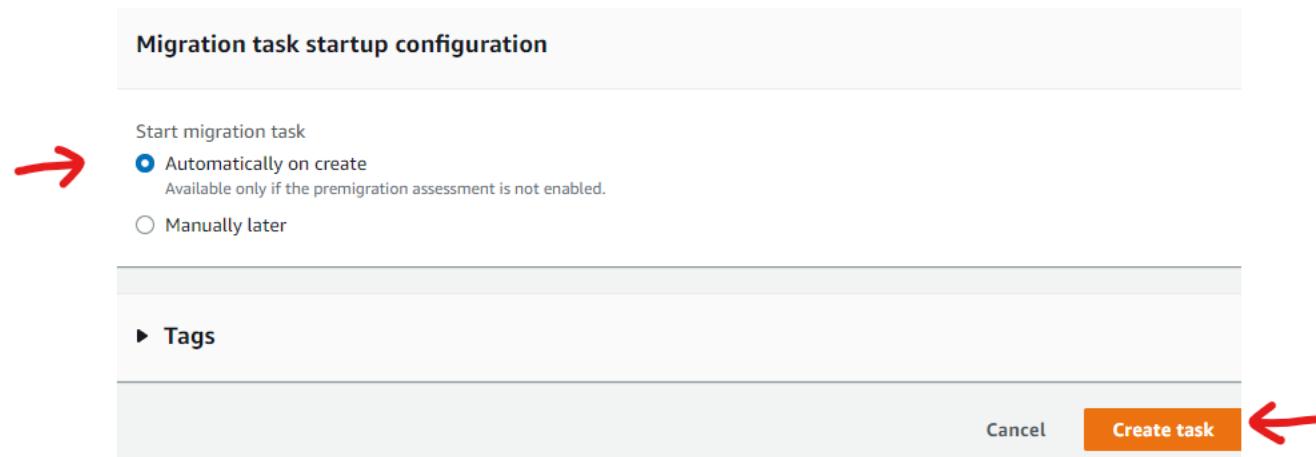
Action

Choose "Include" to migrate your selected objects, or "Exclude" to ignore them during the migration.

Include



**Step-36: Choose the migration task configuration >> Automatically on create >> create task.**



Database migration tasks (1)											Actions	Quick view and compare	Create task
	Find database migration tasks									<	1	>	⚙️
	Identifier	Status	Progress	Type	Source	Target	Replication instance	Started	Stopped				
	mysql-to-postgresql	Load complete	100%	Full load	mysqldb1	postgresqlrds	mysqltopostgres	November 23, 2022 at 03:30:45 (UTC+01:00)	November 23, 2022 at 03:31:40				

**Step-37:** Once the migration is successful, we can verify the table statistics for the migration activity as shown below.

DMS > Database migration tasks > mysql-to-postgresql

### mysql-to-postgresql

Status: Load complete | Type: Full load | Source: mysqlldb1 | Target: postgresqlrds

Actions ▾ Quick view and compare

Summary

Overview details | **Table statistics** | CloudWatch metrics | Mapping rules | Premigration assessments | Tags

Table statistics (157) →

Total rows include loaded source table rows from Inserts, Deletes, Updates, DDLs, and Full load rows.

Find schema

Export to CSV | Validate again | Reload table data

Schema name	Table	Load state	Elapsed load time	Inserts	Deletes	Updates	DDLs	Applied inserts
mysql	server_cost	Table completed	1 s	0	0	0	0	0
mysql	tables_priv	Table completed	1 s	0	0	0	0	0
mysql	gtid_executed	Table completed	< 1 s	0	0	0	0	0
mysql	replication_asynchronous_connection_failover	Table completed	< 1 s	0	0	0	0	0
mysql	replication_asynchronous_connection_failover_managed	Table completed	< 1 s	0	0	0	0	0
mysql	replication_group_configuration_version	Table completed	1 s	0	0	0	0	0