EST
المدرسة العليا للتكنولوجيا
جامعة الحسن الثاني الدار البيضاء
Ecole Supérieure de Te hnologie
Université Hassan II de Casablanca

Oracle Real Application Cluster

(Installation & Configuration)

Supervised By :

Mr.houssaine ziyati

Realized By :

El arrouchi fatima ezzahra

Table of Contents

Chapter I

➢ What is an Oracle RAC :

An Oracle Real Application Clusters (RAC) cluster is a high-availability and scalability solution for Oracle Databases. It enables multiple servers (nodes) to run Oracle Database instances concurrently, all accessing a single shared database. This configuration allows applications to connect to any node, providing continuous access to data even if one node fails. RAC is particularly beneficial for mission-critical applications requiring uninterrupted service and the ability to handle large volumes of transactions.

In a RAC environment, each node operates its own Oracle Database instance, but all instances share the same physical database stored on shared storage. This setup ensures that if one node experiences a failure, other nodes can continue to provide database services without interruption. The nodes communicate with each other through a high-speed interconnect, maintaining data consistency and synchronization across the cluster.

➢ Why a RAC is used :

A RAC offer several reasons to be adopted, the main ones are :

- **High Availability:** RAC ensures continuous database access by allowing multiple instances to run on different servers. If one node fails, others can take over, minimizing downtime.
- **Scalability:** RAC enables horizontal scaling by adding more nodes to the cluster, accommodating growing workloads without significant reconfiguration.
- **Load Balancing:** It distributes database requests across multiple instances, optimizing resource utilization and enhancing performance.
- **Cost Efficiency:** By utilizing commodity hardware and reducing the need for expensive, high-end servers, RAC can lower overall infrastructure costs.

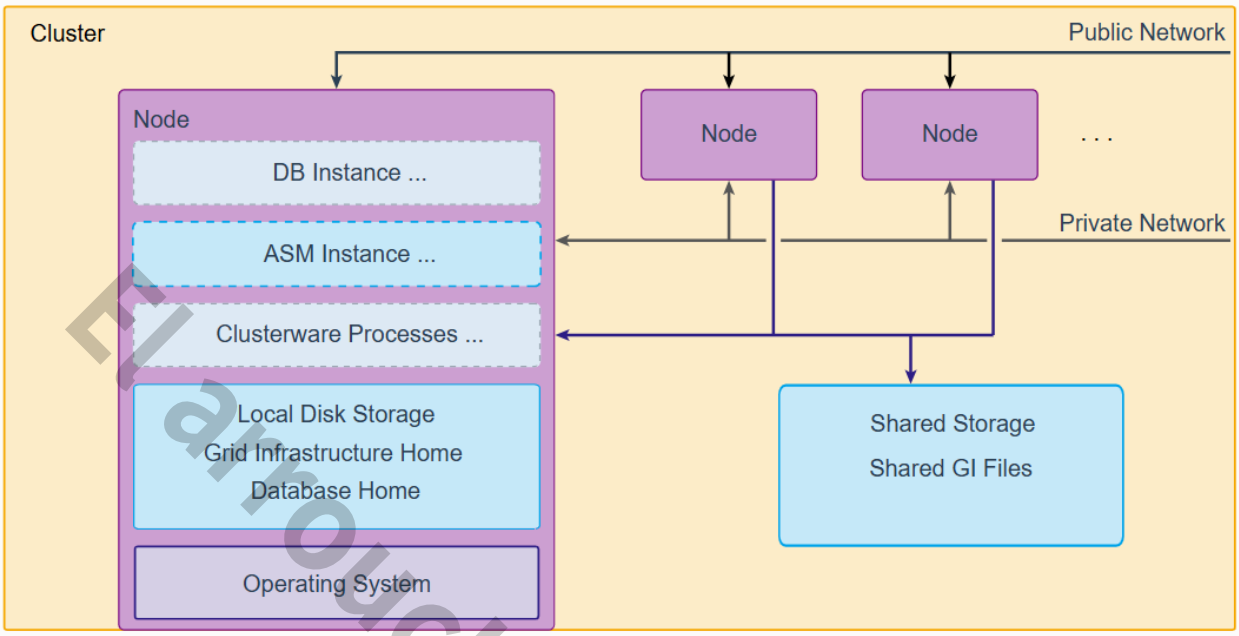
- **Simplified Maintenance:** RAC allows for rolling upgrades and maintenance, enabling system updates without significant downtime.

Many companies have adopted Oracle RAC for the reasons just mentioned, among them we find :

- **AmerisourceBergen:** A global pharmaceutical distribution company that leverages Oracle RAC to ensure high availability and scalability for its critical database applications.
- **NTT Docomo Systems, Inc.:** A leading telecommunications provider that utilizes Oracle RAC to support its large-scale database operations, ensuring uninterrupted service for millions of users.
- **Port of Fujairah:** The port authority employs Oracle RAC to manage its database systems, ensuring high availability and performance for its operations.
- **Qatalum:** A joint venture between Qatar Petroleum and Hydro Aluminium, Qatalum uses Oracle RAC to support its manufacturing operations, ensuring data availability and scalability.
- **Union Life:** A life insurance company that relies on Oracle RAC to maintain high availability and performance for its critical business applications.

➢ The technical architecture of a RAC:

A RAC environment is composed of several key components that collaborate to provide high availability, scalability, and performance for Oracle Databases. The primary components include :



A Clustered Node is a server that is a member or a planned member of a Cluster. There may be one or many nodes in a cluster.

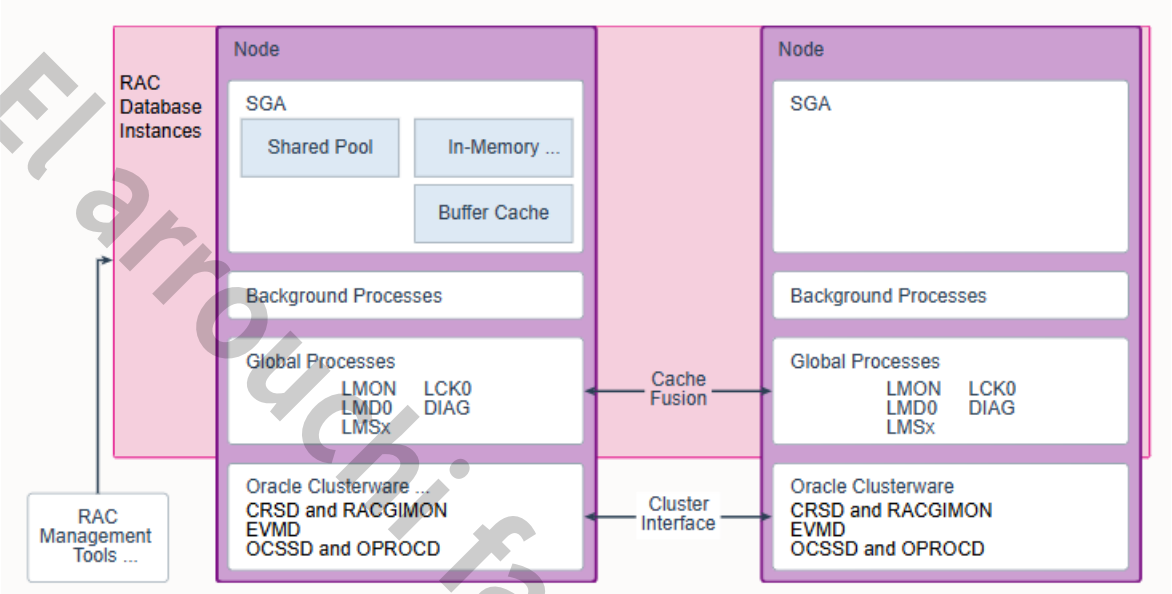
Each node has local storage. Local storage provides for the OS needs and the Grid Infrastructure software installation. Every clustered node has an instance of Clusterware running. The Automatic Diagnostics Repository (ADR) and a local copy for the Oracle Cluster Registry(OLR) are both stored in the node local storage.

Each node must have access to shared storage. Shared storage is available from all the nodes of the cluster.

A node may have one or more other instances running of a RAC database or an application. It is possible to install the software for these instances on shared storage, but it is recommended that these be installed on local storage to allow rolling upgrades, or relocation. However, using local storage also increases your software maintenance costs, because each Oracle home must be upgraded or patched separately. This cost can be mitigated by using Fleet Patching and Provisioning.

Each Node has a network connection to what is called the public network. This network provides the connection applications that use the database or clustered application. Each Node has a private network connection also called the interconnect that provides communication between the nodes of a cluster. Shared storage must be accessible from each node.

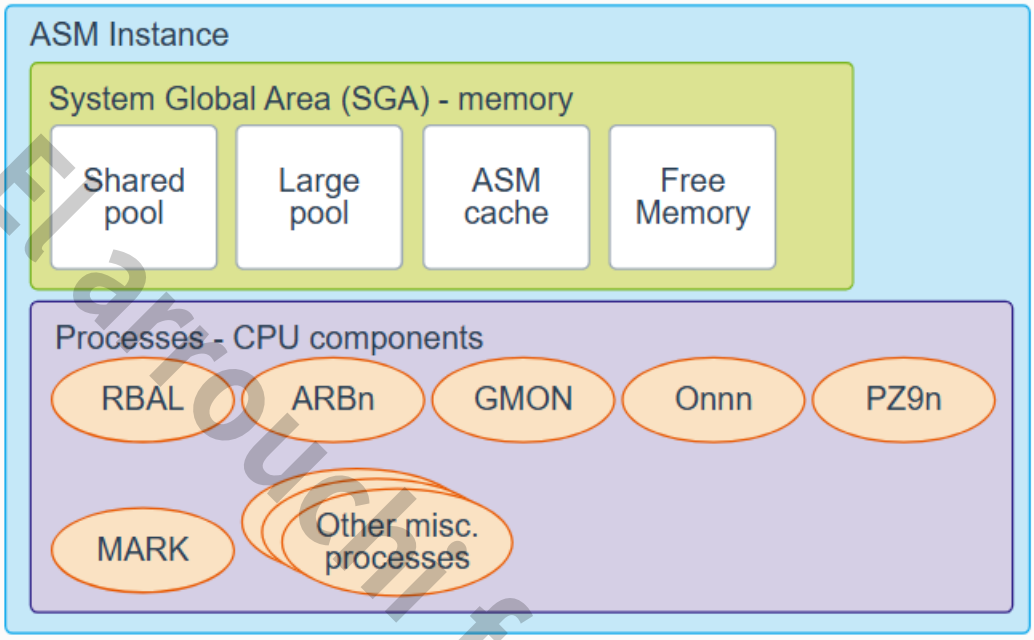
- **Database Instances:** Each node in the RAC cluster runs an Oracle Database instance. These instances operate independently but access a shared database, allowing multiple instances to process requests simultaneously. This parallel processing enhances performance and ensures high availability.



A RAC database is a set of files on shared storage and can span multiple nodes. A RAC database instance is a set of processes and memory that operates on the database files. A RAC database instance resides on one node. It communicates through the global processes and clusterware with other RAC database instances in the same cluster.A RAC database instance has the same memory and background processes as a single instance database, with the addition of Cache Fusion, and Global Processes. Cache Fusion is a diskless cache coherency mechanism in Oracle Real Application Clusters that provides copies of blocks directly from a holding instance's memory cache to a requesting instance's memory cache. This provides a logical merging of the SGA's across the DB instances in the cluster. Global Processes (not all are shown) provide the necessary services to maintain the cache coherency, and recovery in the event of a node failure.
Cache Fusion and Cluster Interface both use the Private Network (Interconnect) for communication between cluster nodes.

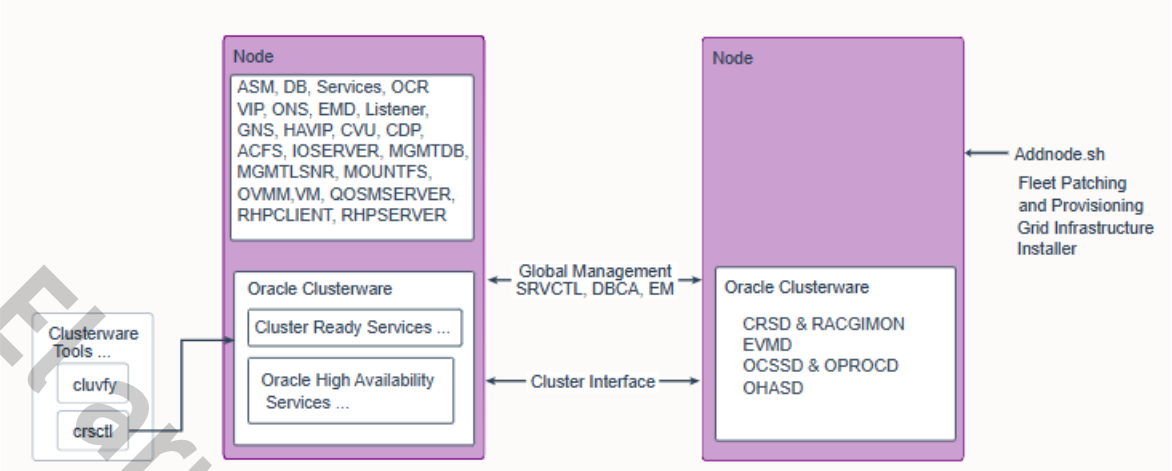
- **Oracle Automatic Storage Management (ASM):** ASM is an integrated, high-performance, and high-availability file system and volume manager for Oracle

Database files. It simplifies storage management by providing a single storage pool for all database files and automates tasks such as striping and mirroring.



Every time ASM or a database is started, a shared memory area called the System Global Area (SGA) is allocated and the ASM background processes are started. However, because ASM performs fewer tasks than a database, an ASM SGA is much smaller than a database SGA. The combination of backgroundprocesses and the SGA is called an Oracle ASM instance. The instance represents the CPU and RAM components of a running ASM environment.

- **Oracle Clusterware:** Oracle Clusterware is the infrastructure that enables the nodes to function as a single system. It manages the cluster's resources, including virtual IP addresses, databases, listeners, and services. Clusterware ensures that if a node fails, its resources are relocated to another node, maintaining continuous database availability.



Oracle Clusterware consists of two separate technology stacks: an upper technology stack anchored by the Cluster Ready Services (CRS) daemon (CRSD) and a lower technology stack anchored by the Oracle High Availability Services daemon (OHASD). The Clusterware components vary slightly based on the platform.
You can use the addnode.sh script to add nodes to an existing cluster. In Oracle 19c Grid Infrastructure, you can use the installer, gridSetup.sh or the Fleet Patching and Provisioning utility rhpctl to add a new node to an existing cluster.

- **Shared Storage:** The database files, including data files, control files, and redo logs, reside on shared storage accessible by all nodes in the cluster. This shared storage ensures that all instances have consistent access to the same data, maintaining data integrity and synchronization across the cluster.

- **Interconnect Network:** The interconnect network facilitates communication between the nodes in the cluster. It is used for data transfer, cache fusion, and synchronization of data blocks between instances. A high-speed, low-latency interconnect is crucial for optimal performance in a RAC environment.

- **Oracle Cluster Registry (OCR) and Voting Disk:** Oracle Clusterware uses the Oracle Cluster Registry (OCR) to store configuration information, including cluster and node membership details. The voting disk is used to determine the majority node in the event of a network partition, preventing split-brain scenarios and ensuring data consistency.

- **Nodes:** Each node in a RAC cluster is a server that runs an Oracle Database instance. These nodes work together to provide a unified database service, with each instance accessing the shared database stored on shared storage. The

nodes communicate with each other through a private interconnect network, enabling coordinated operations and data consistency.

- **Local Disk Storage:** Each node has its own local disk storage, which is used for the operating system (OS) and Oracle software installations. This local storage is essential for the OS's needs and for installing the Oracle Grid Infrastructure and Oracle RAC software. Additionally, node-specific diagnostics and configuration information are stored in the local storage.

- **Grid Infrastructure Home (Grid Home):** The Grid Home is the directory where Oracle Grid Infrastructure software is installed. This software includes Oracle Clusterware and Oracle ASM, which are fundamental for managing the cluster's resources and storage. The Grid Home is typically installed on the local disk storage of each node.

- **Database Home (DB Home):** The Database Home is the directory where the Oracle Database software is installed. It contains the Oracle Database binaries and configuration files. The DB Home is also installed on the local disk storage of each node and is used by the Oracle Database instances running on that node.

- **Operating System (OS):** The operating system provides the foundational environment for the Oracle software to run. It manages hardware resources, facilitates communication between nodes, and supports the execution of Oracle Clusterware, Oracle ASM, and Oracle Database software. The OS is installed on the local disk storage of each node.

➢ How does RAC components collaborate to function :

An Oracle Real Application Clusters (RAC) environment comprises several interconnected components that collaboratively ensure high availability, scalability, and performance. Each **node** in the cluster runs an **Oracle Database instance**, accessing **shared storage** to read and write data, thereby maintaining consistency across the cluster. **Oracle Clusterware** manages the cluster's

resources and monitors the health of nodes and instances, utilizing the **interconnect network** to facilitate communication between nodes and enabling coordinated operations and failover processes. **Oracle Automatic Storage Management (ASM)** manages the shared storage, organizing database files into disk groups and automating storage management tasks to ensure data is distributed evenly across disks and provides redundancy for data protection. The **Oracle Cluster Registry (OCR)** stores configuration information, while the **voting disk** helps in node arbitration during network partitions, enabling Oracle Clusterware to manage node membership and resource allocation effectively. Each node utilizes its **local disk storage** for the **operating system (OS)** and Oracle software installations, ensuring that each node operates independently with its own OS environment and Oracle software binaries. Both the **Grid Home** and **Database Home** are installed on the local disk storage of each node; the Grid Home contains the Oracle Clusterware and Oracle ASM software, which manage the cluster's resources and storage, while the Database Home contains the Oracle Database software, used by the Oracle Database instances running on the node. The **operating system** provides the necessary environment for the Oracle software to function, managing hardware resources, facilitating communication between nodes, and supporting the execution of Oracle Clusterware, Oracle ASM, and Oracle Database software.

Chapter 2

➢ Required Hardware and Resources :

- **Servers (Nodes):** Each node should have at least 8 GB of RAM for Oracle Grid Infrastructure installations.
- **Storage:** A minimum of 1 GB of disk space is required to install Java Runtime Environment (JRE) and Oracle Universal Installer (OUI) on the disk partition where the operating system is installed.
- **Network:** Each node must have at least two network interface cards (NICs): one for the public network and another for the private interconnect network.

➢ Configuration of Virtual Machines :

- **Processor:** Ensure that each virtual machine (VM) has a compatible processor architecture, such as x86-64.
- **Memory:** Allocate a minimum of 8 GB of RAM to each VM for Oracle Grid Infrastructure installations.
- **Storage:** Assign sufficient disk space to each VM to accommodate the operating system, Oracle software installations, and database files.
- **Network:** Configure at least two virtual NICs per VM: one for the public network and another for the private interconnect network.
-

➢ Operating Systems and Versions:

- **Oracle Linux:** Supported versions include Oracle Linux 8.2 with the Unbreakable Enterprise Kernel 6: 5.4.17-2011.1.2.el8uek.x86_64 or later, and Oracle Linux 7.6 with the Unbreakable Enterprise Kernel 5: 4.14.35-2025.404.1.el7uek.x86_64 or later.
- **Red Hat Enterprise Linux (RHEL):** Supported versions include RHEL 8.2: 4.18.0-193.19.1.el8_2.x86_64 or later.
- **SUSE Linux Enterprise Server (SLES):** Supported versions include SLES 15 SP1: 4.12.14-197.29-default or later.
- **Windows Server:** Supported versions include Windows Server 2022 x64 - Standard, Datacenter, and Essentials editions (Supported starting Oracle Database 19c Release Update 19.13 or later), Windows Server 2019 x64 - Standard, Datacenter, and Essentials editions, and Windows Server 2016 x64 - Standard, Datacenter, and Essentials editions.

Chapter 3

1. Setting Up Virtual Machines for Oracle RAC Configuration

Software Prerequisites

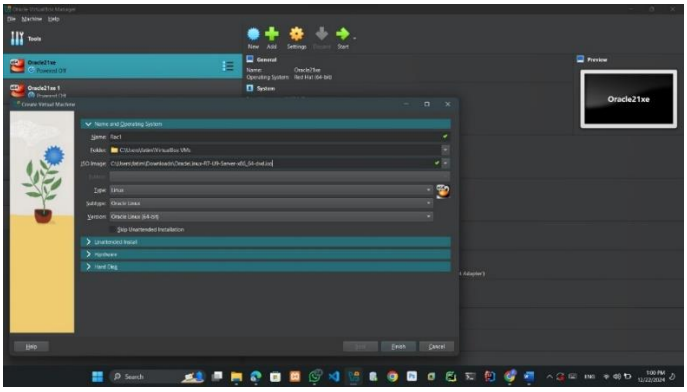
Step 1: Installing Oracle VirtualBox
We started by downloading Oracle VirtualBox version 7.1.0 from the official VirtualBox website. To ensure compatibility, we selected the version that matched our operating system. Once downloaded, we followed the installation instructions and successfully set up VirtualBox.

Step 2: Downloading Oracle Linux ISO
Next, we downloaded the Oracle Linux 7.7 ISO file from the Oracle Linux downloads page. This file served as the operating system for the virtual machines that we configured to host our Oracle RAC nodes.

Step1: Setting Up Your VM for Node 1

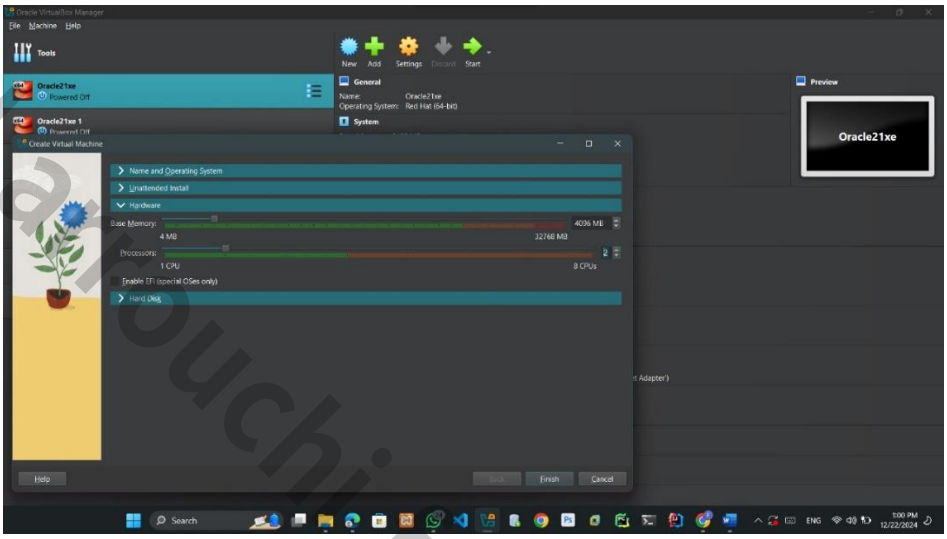
- **Name and Operating System**

We started by creating our first virtual machine, which we named **Rac1**. We selected a folder location to store the VM files and determined the amount of storage to allocate. For the **Type**, we chose the operating system we wanted to configure, such as Windows, Solaris, or Linux. Since we were setting up Linux, we specified the **Subtype** (e.g., Oracle Linux or RHEL) and ensured that we selected the correct **Version** (64-bit) to meet our requirements.



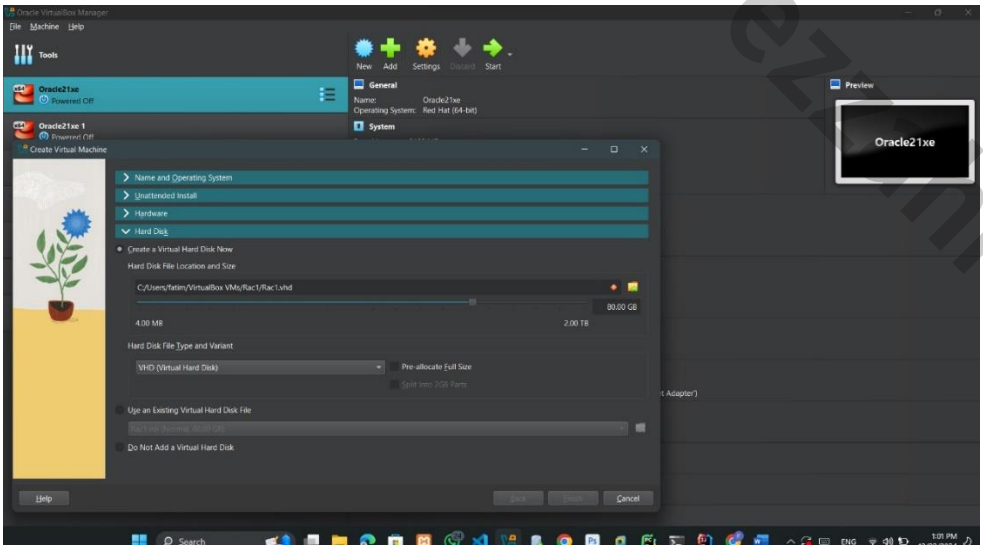
- **Hardware:**

We made sure to configure a minimum of 4 GB of RAM to ensure smooth performance for the virtual machine. Additionally, we allocated at least 2 CPU cores to handle the processing requirements effectively.



- **Hard Disk:**

We created a virtual hard disk for the virtual machine, allocating 60 GB of storage for one node. This value can vary depending on the requirements of your environment. For the hard disk type, we selected **VHD (Virtual Hard Disk)** as the variant. Once we finalized these selections, we clicked on **Finish** to complete the setup.

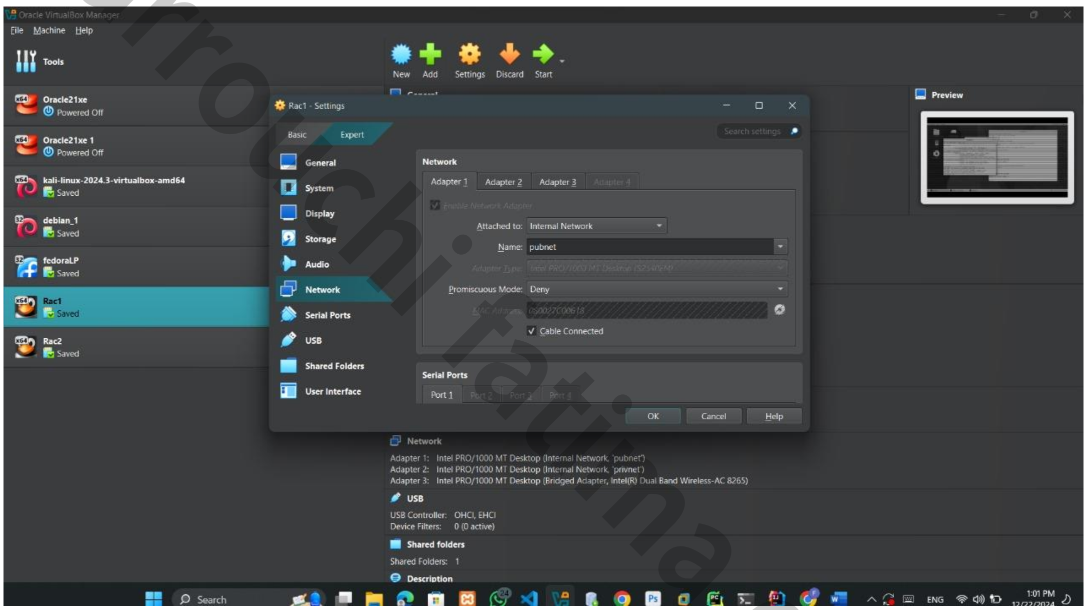


Step 2: Configuring Network Settings for Node 1

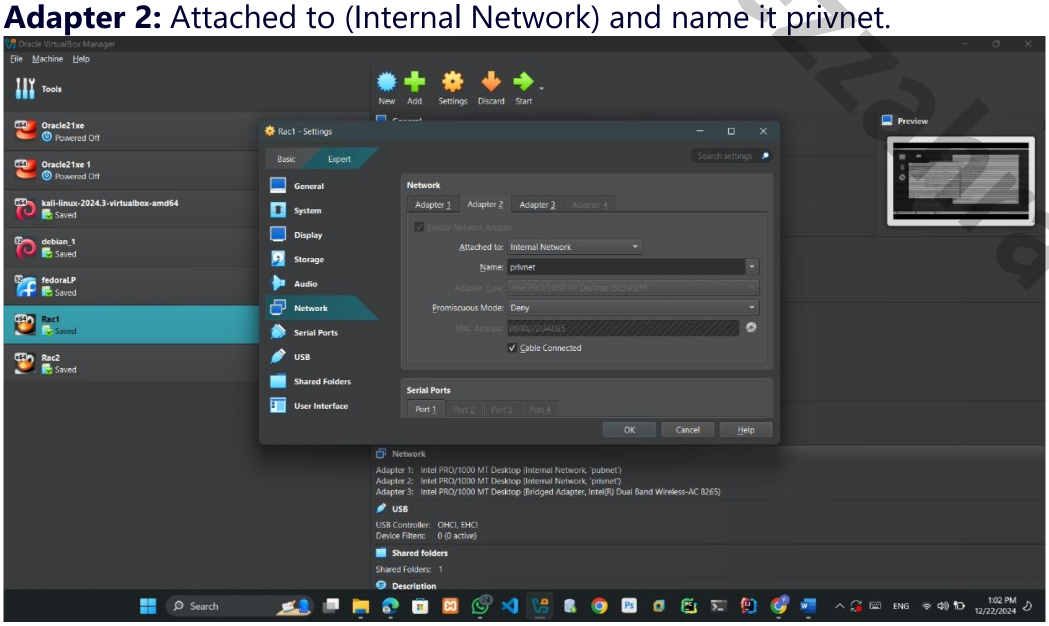
Now it's time to configure the network settings for our **Node 1** virtual machine. This step is crucial to ensure effective communication between our Oracle RAC nodes.

We began by enabling all three network adapters for Node 1:

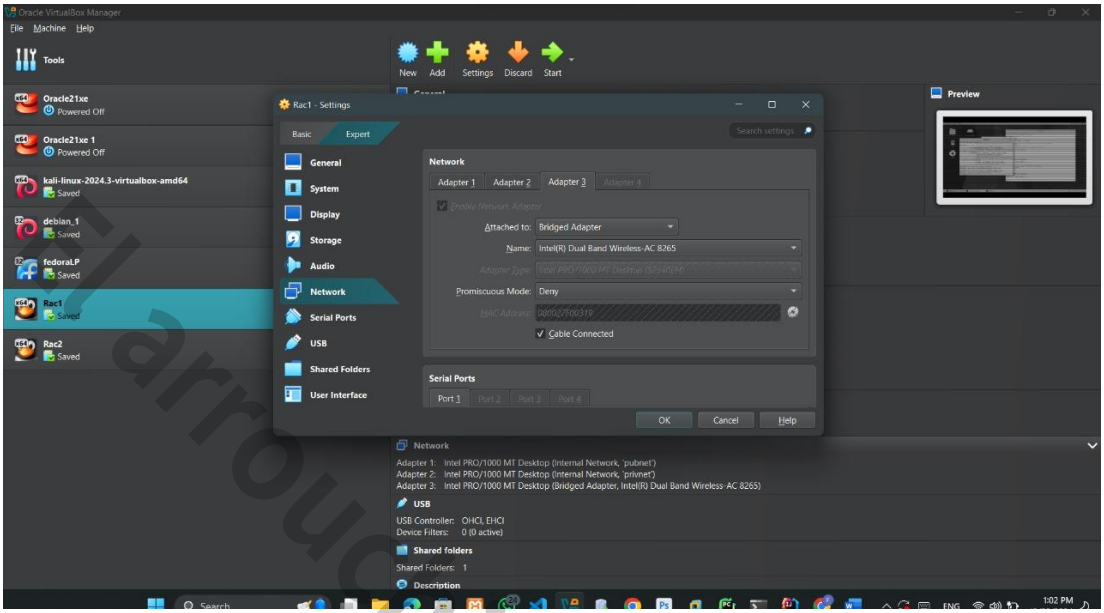
- **Adapter 1:** We attached this adapter to an **Internal Network** and named it **pubnet** to serve as the public network interface.



- **Adapter 2:** Attached to (Internal Network) and name it privnet.

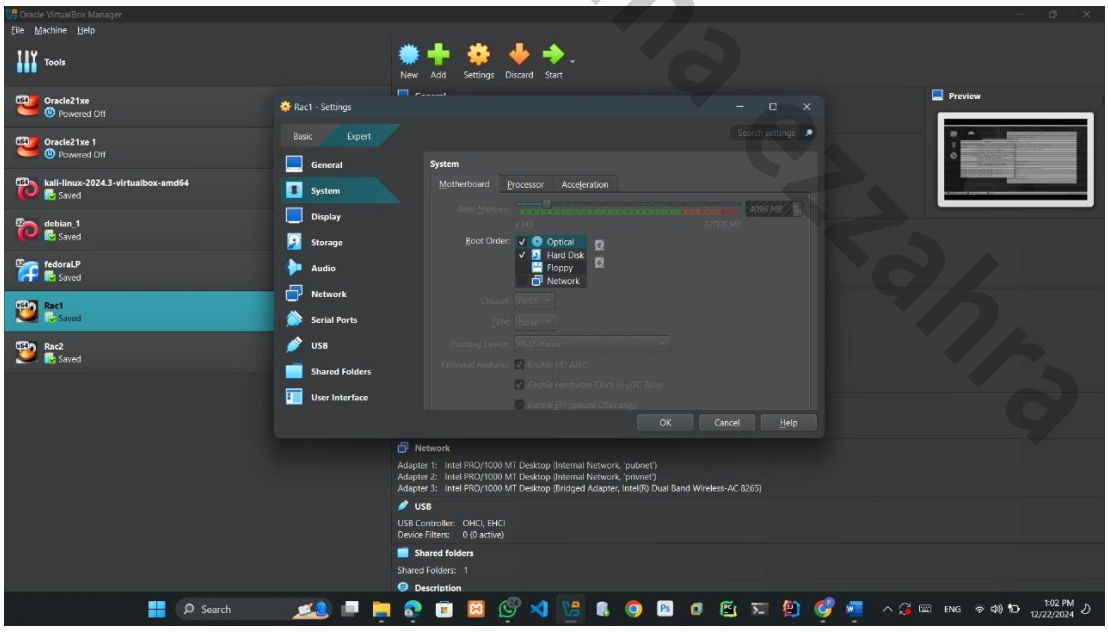


- **Adapter 3:** Attached to (Bridged Adapter)



Once we've configured the adapters, we clicked OK to save our settings.

We navigated to the **System** section of the virtual machine settings and removed the **Floppy disk** from the boot order to streamline the boot process. Once this change was made, we clicked on **OK** to save our settings.

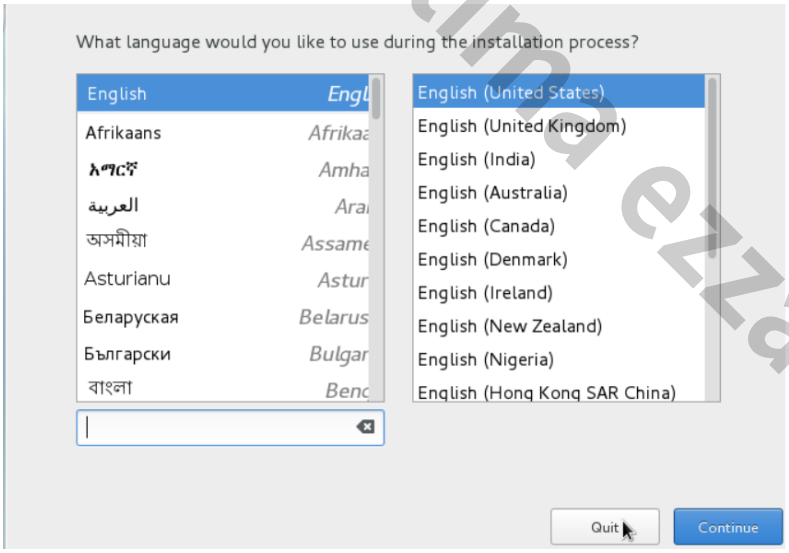


After that We clicked on the **Storage** section and selected **Controller: IDE Empty**. Next, we chose the optimal drive and attached the Oracle Linux ISO file. Once the ISO was selected, we clicked on **OK** to save the settings.

Step 3: Starting the Installation

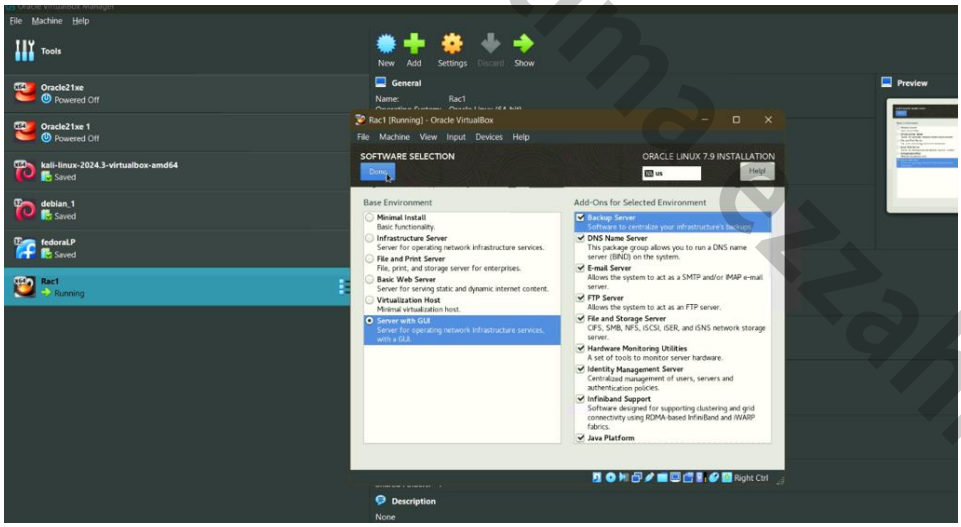
Once our configuration was complete, we clicked on the **Start** button to begin the installation process. We followed the on-screen prompts to guide us through the installation steps. After clicking **Start**, we used the **up-arrow** key to navigate through the menu options and selected **Install Oracle Linux 7.7** to proceed with the installation.

When prompted, we chose the default language settings. After selecting the language, we pressed **Continue** to move on to the next step of the installation process.

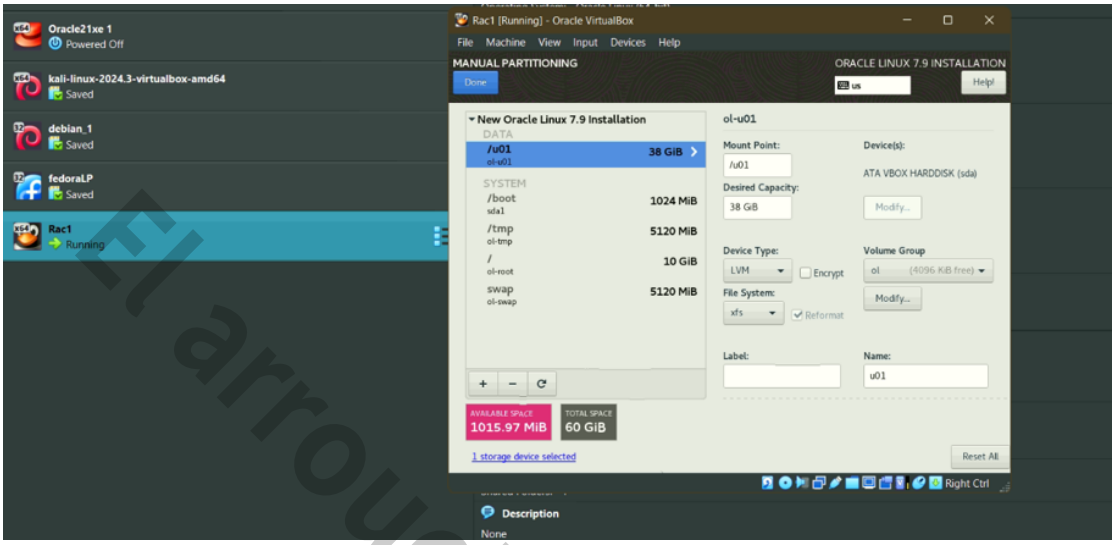


Next, we selected the **Date & Time** option to configure our time zone. After adjusting the settings, we clicked on **Done** to save our changes.

Next, we navigated to the **Software Selection** section. We selected **Server with GUI** and checked the box for **Add-Ons for Selected Environment**. To ensure a comprehensive installation, we made sure to select all available options. After making these selections, we clicked on **Done** to save our changes.

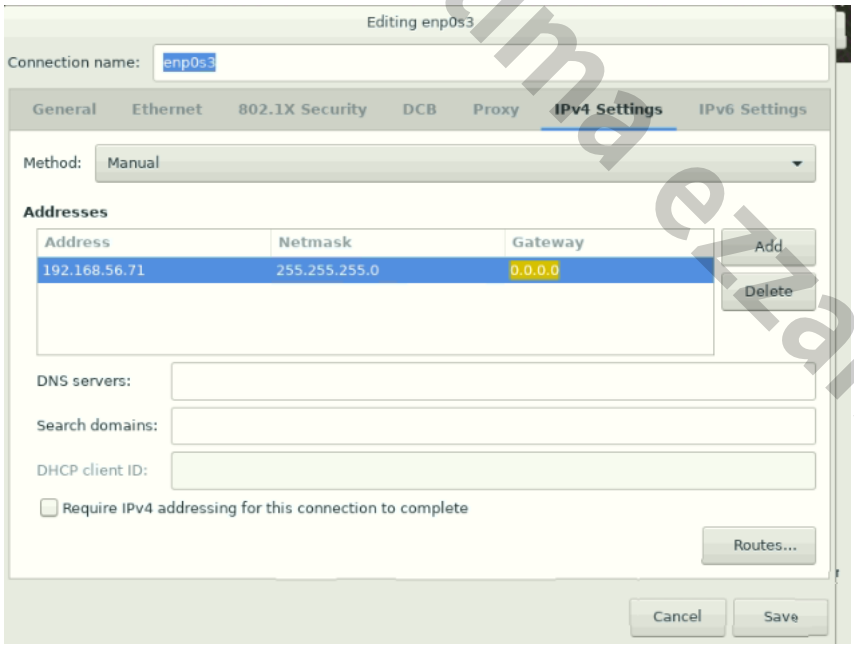


Next, we selected **Installation Destination** and chose the option **I will configure partitioning**. We customized the disk partitions according to our needs. After configuring the partitions, we clicked on **Done** to review our settings. Finally, we accepted the changes to proceed with the installation.

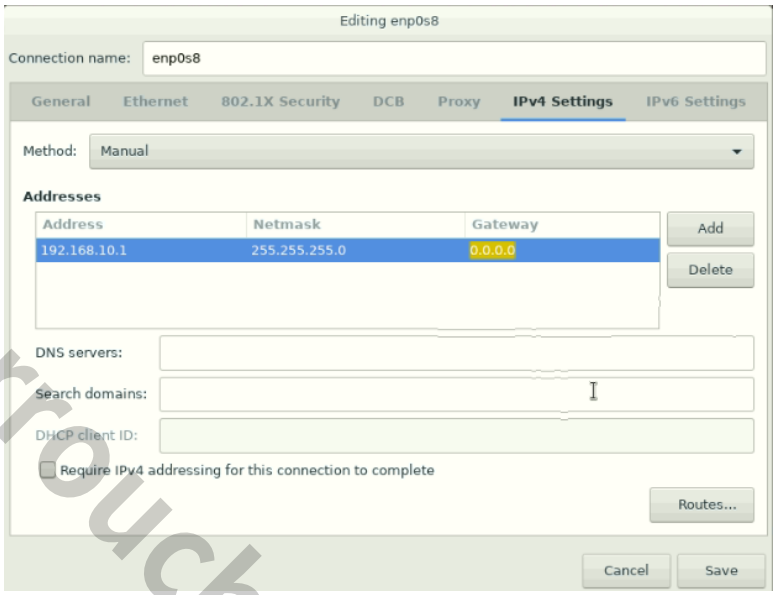


We selected **Network & Hostname** to change the hostname of the system. Then, we clicked on **Configure** to access the network settings. Under **IPv4**, we changed the method from **Automatic** to **Manual** and entered our desired IP address.

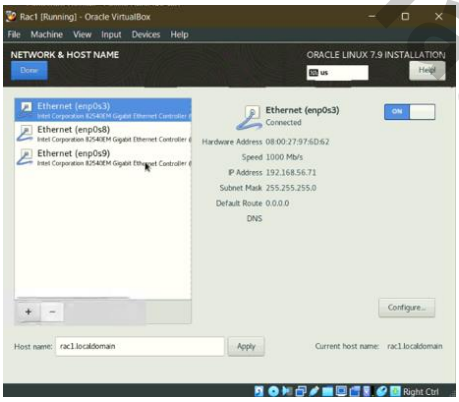
Configuring a Public IP Address



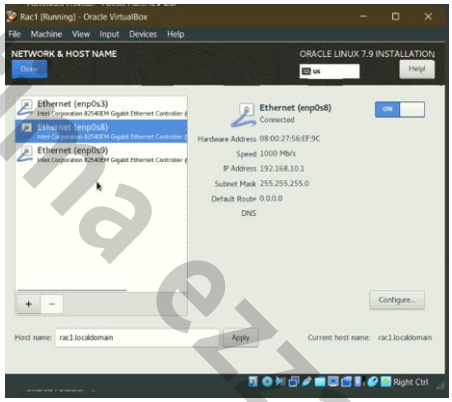
Configuring a Private IP Address



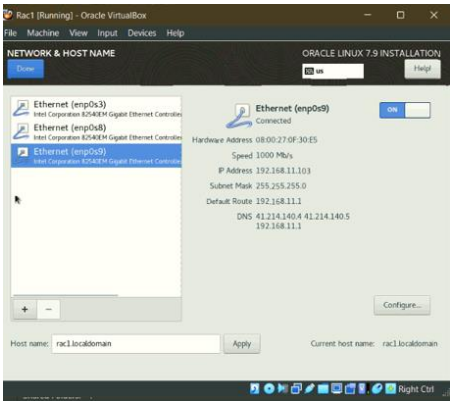
Enp0s3:



Enps0s8:

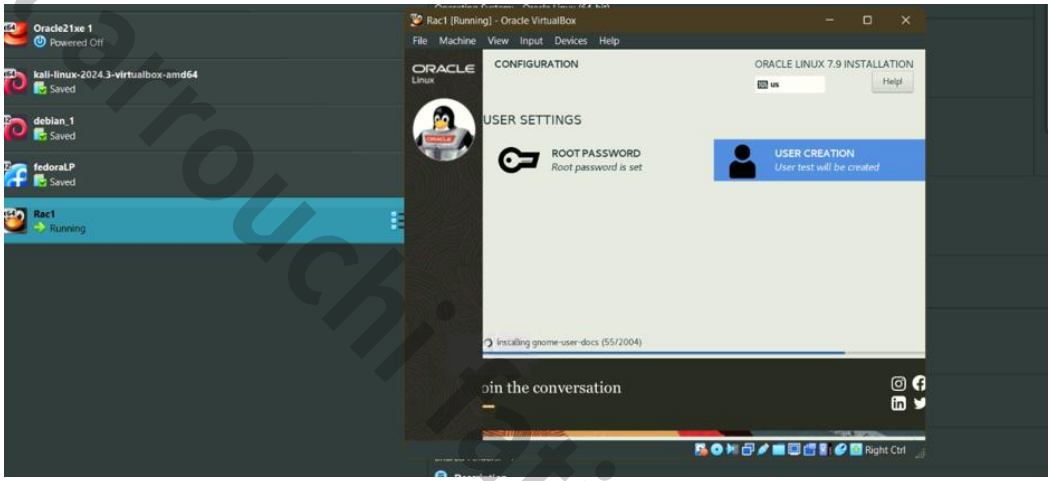


Enps0s9:



Once we had completed all configurations, we clicked on **Begin Installation** to start the installation process.

Final Step: Configuring Root Password and User Creation
In this final step, we configured the **Root Password** to secure our system. After setting the root password, we proceeded to create a new user account. We followed the prompts to enter the username and password for the new user.



2. Preparing the Environment for Oracle RAC Installation

In this step, we will prepare the environment for the Oracle RAC installation by installing the necessary packages, setting up user groups and directories, and configuring hostname resolution. We will also adjust the firewall settings

Install Required Packages

```
yum install -y oracle-database-preinstall-19c
yum install oracleasm-support
yum install -y bind*
```

Load System Parameters

```
sysctl -p
```

Create ASM Groups

```
groupadd -g 54327 asmdba
groupadd -g 54328 asmoper
groupadd -g 54329 asmadmin
```

Modify User Groups

```
usermod -u 54321 -g oinstall -G dba,oper,asmdba oracle
```

Create Grid User

```
useradd -u 54331 -g oinstall -G dba,asmdba,asmoper,asmadmin,racdba grid
```

Set Passwords

```
passwd oracle
passwd grid
```

Create Necessary Directories & Set Permissions

```
 mkdir -p /u01/app/oracle/product/19/db_home #Oracle Home
chown -R oracle:oinstall /u01
mkdir -p /u01/app/grid # Grid Base
mkdir -p /u01/app/19/grid # Grid Home
```

```
cd /u01/app
chown -R grid:oinstall 19
chown -R grid:oinstall grid
```

Update Hostname Resolution

```
[root@rac1 ~]#
[root@rac1 ~]#
[root@rac1 ~]#
[root@rac1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

[root@rac1 ~]# vi /etc/hosts
[root@rac1 ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.lc

# Public

192.168.56.71 rac1.localdomain rac1
192.168.56.72 rac2.localdomain rac2

# Private

192.168.10.1 rac1-priv.localdomain rac1-priv
```

Configure Bash Profile of Users

We Log in as the “oracle” user and we added the following lines at the end of the “/home/oracle/.bash_profile” file.

```
# Oracle Settings
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=/u01/app/19c/grid
export DB_HOME=$ORACLE_BASE/product/19c/db_1
export ORACLE_HOME=$DB_HOME
export ORACLE_SID=racdb1
export ORACLE_TERM=xterm
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HO
ME/rdbms/jlib

alias grid=’. /home/oracle/grid.env’
alias db=’. /home/oracle/db.env’
```

We Created a file called “/home/oracle/grid.env” with the following contents.

```
export ORACLE_SID=+ASM1
export ORACLE_HOME=$GRID_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HO
ME/rdbms/jlib
```

We Created a file called "/home/oracle/db.env" with the following contents.

```
export ORACLE_SID=racdb1
export ORACLE_HOME=$DB_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

Firewall Configuration

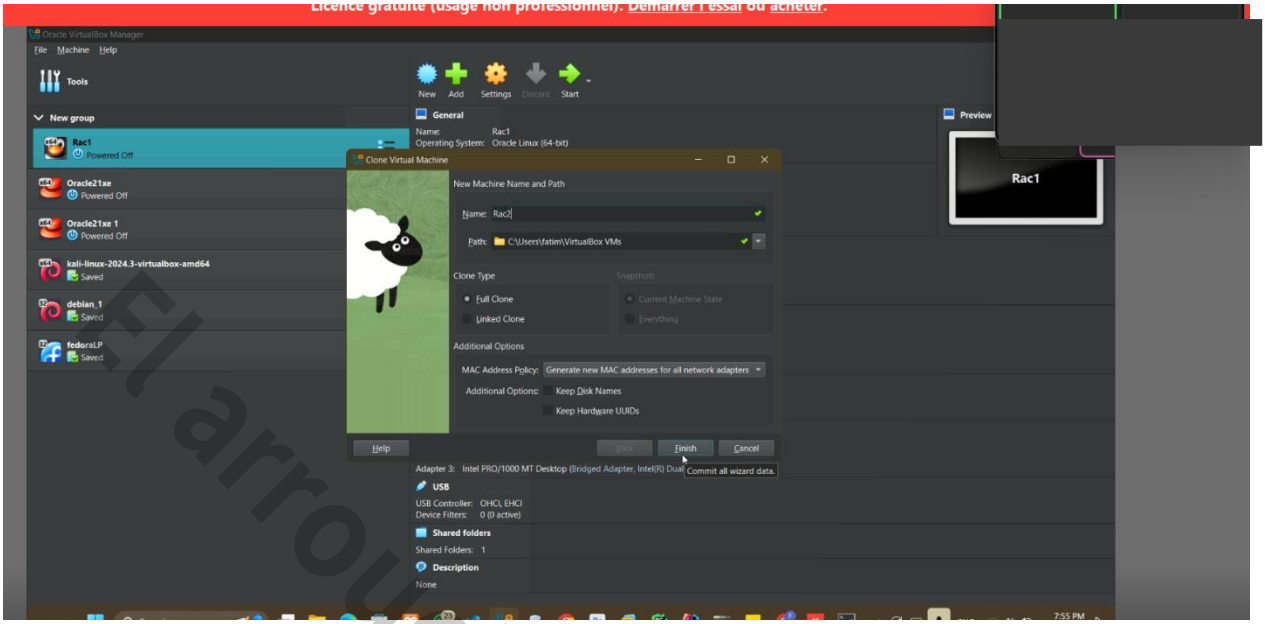
```
systemctl status firewalld
systemctl stop firewalld
systemctl disable firewalld
```

3. Backup VMbox for build 2nd Node and adding Diskgroup

In this step, we will clone a virtual machine, configure it with a unique name, and ensure it has distinct MAC addresses. We will also handle disk partitioning, attach disks, and convert them to shareable for our Oracle RAC setup.

Cloning the Virtual Machine

To clone the virtual machine, we first ensure that it is shut down. We right-click on the VM and select Clone from the context menu. We then follow the prompts to create a copy of our virtual machine.
We assign the name rac2 to the new virtual machine and select the appropriate Path where we want to store the clone. We choose Full as the Clone Type to ensure a complete copy of the VM is created.



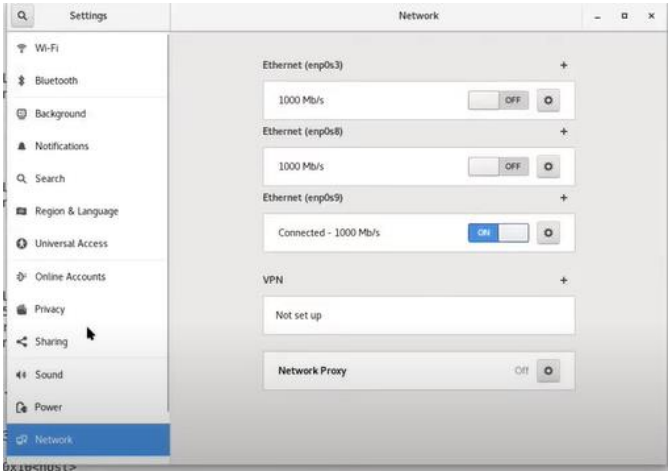
Once the cloning is complete, start both VMs. To change the hostname of the newly cloned VM, use the appropriate command.

```
Hostnamectl set-hostname rac2
```

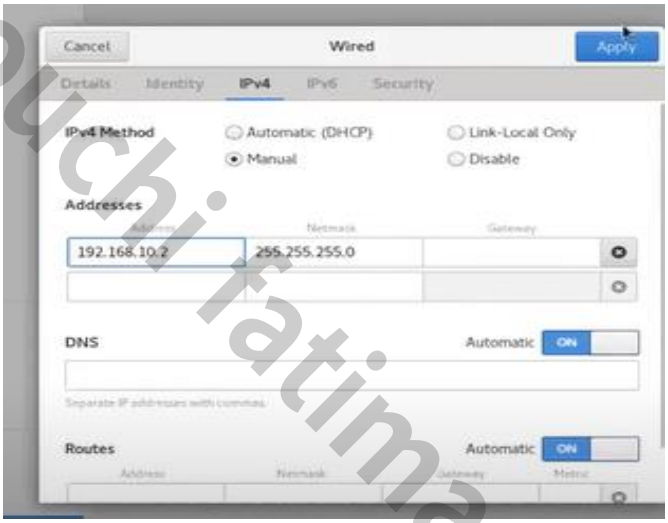
After updating the hostname, we enter **su** and press Enter to switch to the superuser, allowing us to perform system updates or make additional configuration changes as needed.

To change the IP address, we clicked on Settings and selected Wired settings. This allowed us to configure the network settings, including the IP address for the VM.

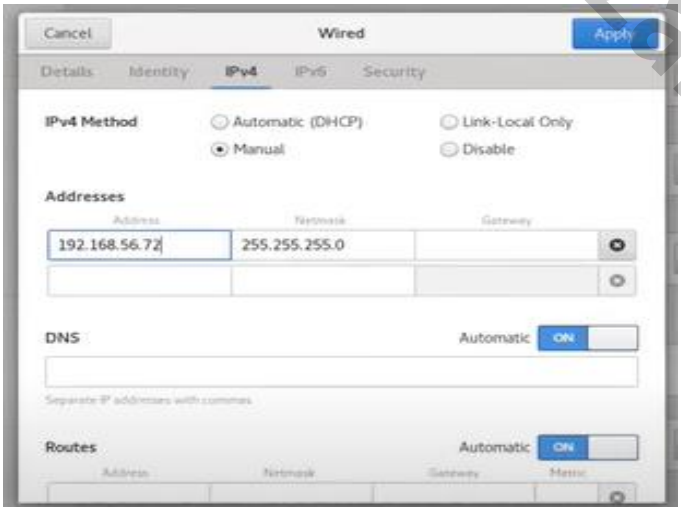
Click on **Settings**, then navigate to **IPv4**. Select the **IPv4 Method** as **Manual** and enter the desired IP address. This will configure your network settings accordingly.



Private IP Address



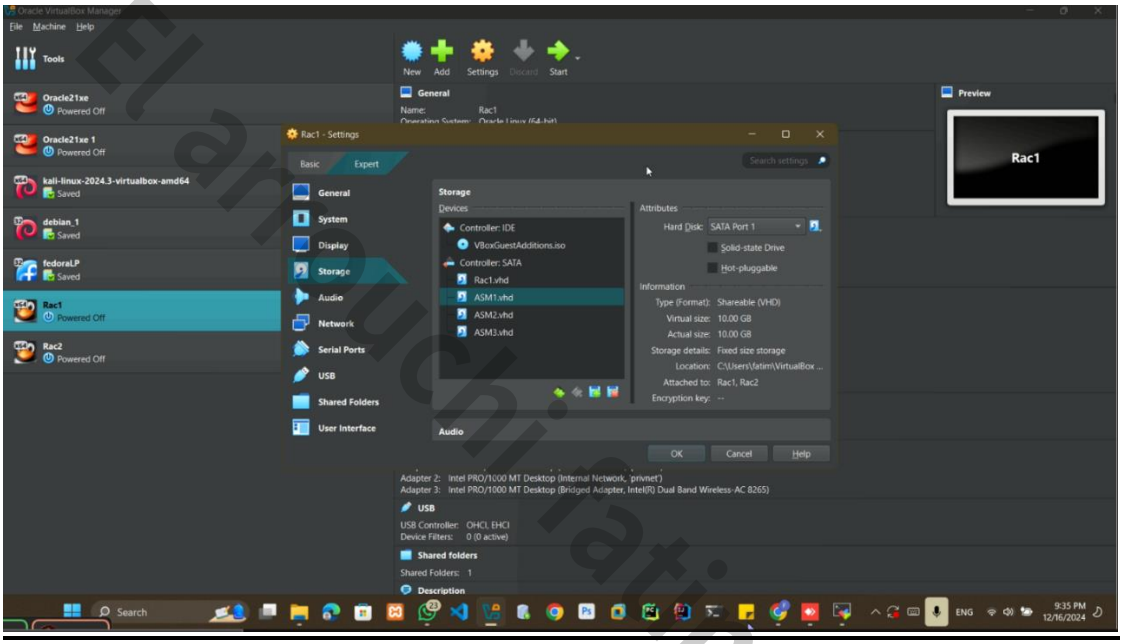
Public IP Address



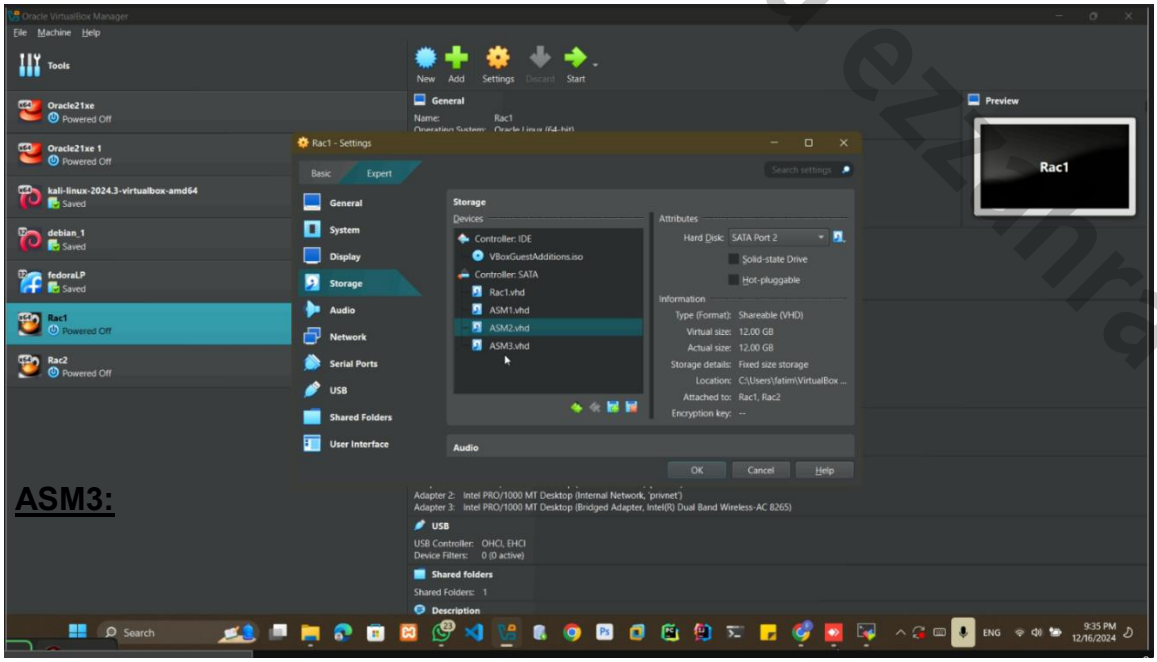
We clicked on **Apply** to save our changes. Then, to attach the shared disk, we shut down both VM 1 and VM 2 to ensure a safe configuration process.

We clicked on Storage, selected the controller, and then chose the option to add four new hard disks. We followed the prompts to configure the new hard disks for our VMs.

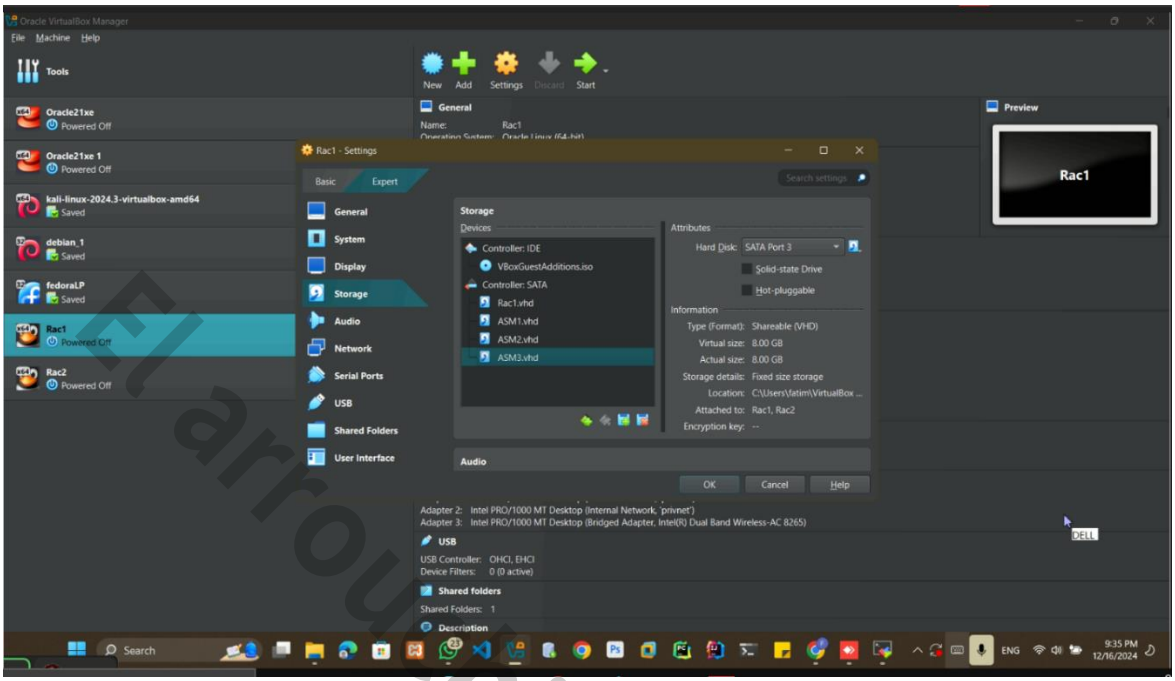
ASM1



ASM2:



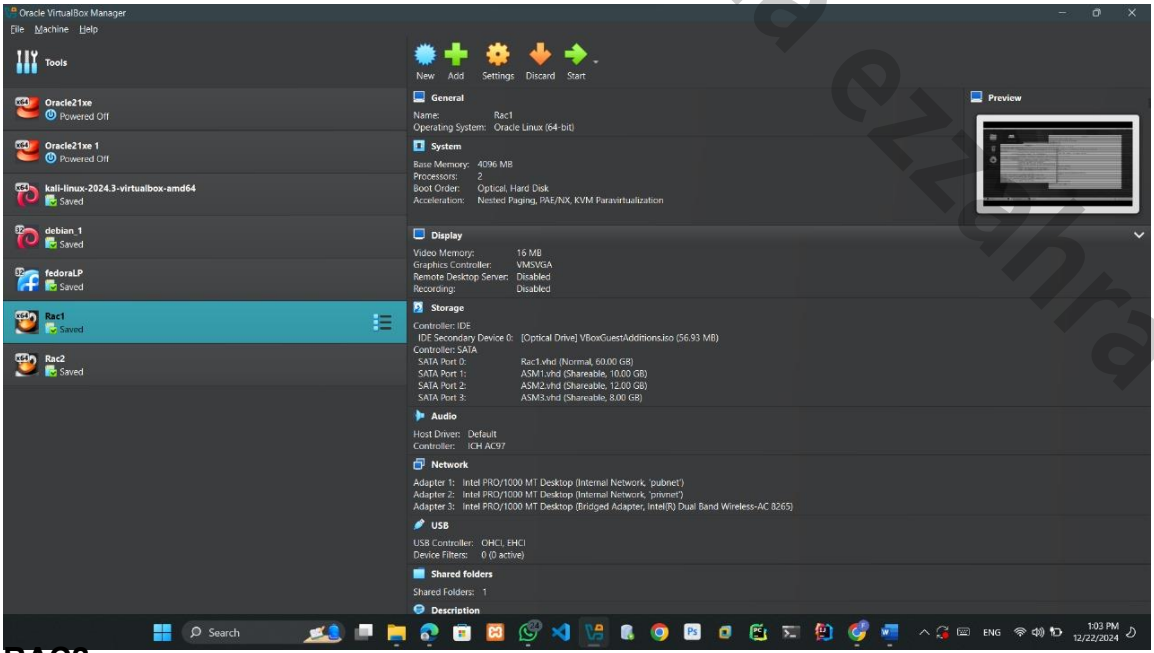
ASM3:



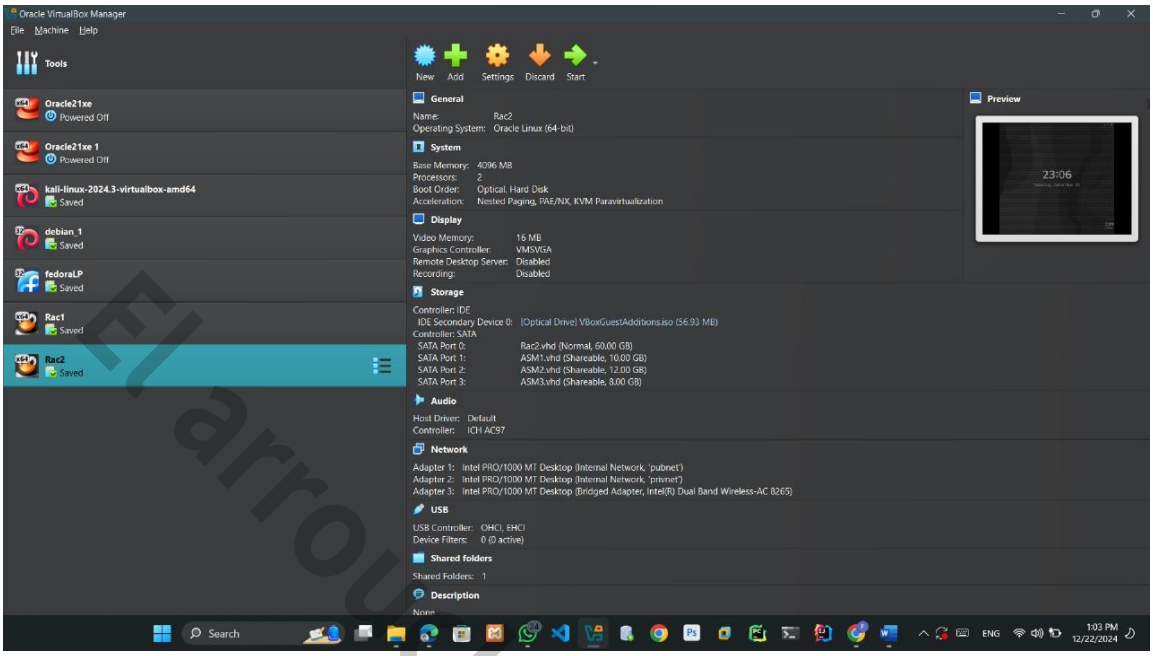
We repeated the process for each disk by selecting it and clicking on **Choose** to attach it. We made sure to add all disks to both VM 1 and VM 2 to ensure they had access to the shared storage.

Now both VMs display similarly as shown below.

RAC1



RAC2:



Once we reviewed the configurations of both VMs, we proceeded to start both virtual machines: RAC 1 and RAC 2.

Start both VMs to begin the disk partitioning process. Once they are running, you can check the disks using the command **lsblk**. You should see the disks listed, for example, as **sdb**, **sdc**, **sdd**, and **sde**. This will allow you to identify the disks you want to partition

```
fdisk /dev/sde
```

Welcome to fdisk (util-linux 2.23.2).
Changes will remain in memory only until you decide to write them.
Be cautious when using the write command.

Device does not contain a recognized partition table.
Building a new DOS disk label with disk identifier 0x50231075.

Command (m for help): n
Partition type:

p primary (0 primary, 0 extended, 4 free)

e extended

Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048): *(Using default value 2048)*

Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039): *(Using default value 41943039)*

Partition 1 of type Linux and of size 20 GiB has been created.

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read the partition table.
Syncing disks.

4. Configuration of the Oracle ASM Library Driver, Creation of ASM Disks, and Establishing SSH Connections for Seamless Node Communication

Initialize ASM Configuration

We started by running the following command to configure the Oracle ASM library driver:

```
oracleasm configure -i

Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM library
driver. The following questions will determine whether the driver is
loaded on boot and what permissions it will have. The current values
will be shown in brackets ('[]'). Hitting without typing an
answer will keep that current value. Ctrl-C will abort.

Default user to own the driver interface []: grid
Default group to own the driver interface []: asmadmin
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done
```

Initialize ASM:

Next, we initialized ASM by running the following command:

This command performed several tasks:

- Created the /dev/oracleasm mount point.
- Loaded the ASM module.
- Configured it to use the physical block size.
- Mounted the ASMlib driver filesystem at /dev/oracleasm.

```
oracleasm init

Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
```

Create ASM Disk

Now, we created a disk for ASM using the following command:

```
oracleasm createdisk DATA /dev/sdb1

Writing disk header: done
Instantiating disk: done
```

To confirm the disk creation, run:

```
oracleasm scandisks

Reloading disk partitions: done
Cleaning any stale ASM disks…
Scanning system for ASM disks…
```

We verified the created disk by running the following command:

```
oracleasm listdisks
DATA
```

Switch to Node 2:

We repeated the configuration process for the second node. We started by running the same initialization command.

```
oracleasm configure -i

Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM library
driver. The following questions will determine whether the driver is
loaded on boot and what permissions it will have. The current values
will be shown in brackets ('[]'). Hitting without typing an
answer will keep that current value. Ctrl-C will abort.

Default user to own the driver interface []: grid
Default group to own the driver interface []: asmadmin
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done
```

```
oracleasm init

Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
```

We confirmed the disk creation by running the following command:

```
oracleasm scandisks

Reloading disk partitions: done
Cleaning any stale ASM disks…
Scanning system for ASM disks…
```

We verified the created disk and ensured that we saw the same output as on Node 1 by running the following command:

```
oracleasm listdisks
DATA
```

Unzip the Grid Software :

As the grid user, we navigated to the directory where the Oracle Grid Infrastructure software was downloaded. We unzipped the software to the Grid home directory.

Setting Up SSH Connection

We set up SSH connections to facilitate communication between our RAC nodes. We executed the following command:

```
./sshUserSetup.sh -user oracle -hosts "rac1 rac2" -
noPromptPassphrase -confirm -advanced
```

Verifying RAC Configuration with Cluvfy

We verified the Real Application Clusters (RAC) configuration to ensure it met the necessary requirements. Cluvfy (Cluster Verify) was used as a valuable tool to check the cluster's configuration and readiness.

Running the Cluvfy Utility

We performed a pre-installation check for Oracle Clusterware by executing the following command:

```
./runcluvfy.sh stage -pre crsinst -n rac1,rac2 -verbose
```

```
Verifying Physical Memory ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          3.5908GB (3765196.0KB)    8GB (8388608.0KB)         failed
  rac1          3.5908GB (3765196.0KB)    8GB (8388608.0KB)         failed
Verifying Physical Memory ...FAILED (PRVF-7530)
Verifying Available Physical Memory ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          2.9132GB (3054712.0KB)    50MB (51200.0KB)          passed
  rac1          2.6486GB (2777304.0KB)    50MB (51200.0KB)          passed
Verifying Available Physical Memory ...PASSED
Verifying Swap Size ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          20GB (2.0971516E7KB)      3.5908GB (3765196.0KB)    passed
  rac1          20GB (2.0971516E7KB)      3.5908GB (3765196.0KB)    passed
Verifying Swap Size ...PASSED
Verifying Swap Size ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          20GB (2.0971516E7KB)      3.5908GB (3765196.0KB)    passed
  rac1          20GB (2.0971516E7KB)      3.5908GB (3765196.0KB)    passed
Verifying Swap Size ...PASSED
Verifying Free Space: rac2:/usr,rac2:/etc,rac2:/sbin ...
  Path              Node Name     Mount point   Available     Required      Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  /usr              rac2          /             130.4824GB    25MB          passed
  /etc              rac2          /             130.4824GB    25MB          passed
  /sbin             rac2          /             130.4824GB    10MB          passed
Verifying Free Space: rac2:/usr,rac2:/etc,rac2:/sbin ...PASSED
Verifying Free Space: rac2:/var ...
  Path              Node Name     Mount point   Available     Required      Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  /var              rac2          /var          6.9707GB      5MB           passed
Verifying Free Space: rac2:/var ...PASSED
Verifying Free Space: rac2:/tmp ...
```

```
  Path            Node Name     Mount point   Available     Required      Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  /tmp            rac2          /tmp          20.9131GB     1GB           passed
Verifying Free Space: rac2:/tmp ...PASSED
Verifying Free Space: rac1:/usr,rac1:/etc,rac1:/sbin ...
  Path            Node Name     Mount point   Available     Required      Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  /usr            rac1          /             118.4326GB    25MB          passed
  /etc            rac1          /             118.4326GB    25MB          passed
  /sbin           rac1          /             118.4326GB    10MB          passed
Verifying Free Space: rac1:/usr,rac1:/etc,rac1:/sbin ...PASSED
Verifying Free Space: rac1:/var ...
  Path            Node Name     Mount point   Available     Required      Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  /var            rac1          /var          7.9434GB      5MB           passed
Verifying Free Space: rac1:/var ...PASSED
Verifying Free Space: rac1:/tmp ...
  Path            Node Name     Mount point   Available     Required      Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  /tmp            rac1          /tmp          20.9111GB     1GB           passed
Verifying Free Space: rac1:/tmp ...PASSED
Verifying User Existence: grid ...
  Node Name     Status                    Comment
  ------------  ------------------------  ------------------------
  rac2          passed                    exists(54331)
  rac1          passed                    exists(54331)
  Verifying Users With Same UID: 54331 ...PASSED
Verifying User Existence: grid ...PASSED
Verifying Group Existence: asmadmin ...
  Node Name     Status                    Comment
  ------------  ------------------------  ------------------------
  rac2          passed                    exists
  rac1          passed                    exists
Verifying Group Existence: asmadmin ...PASSED
Verifying Group Existence: asmdba ...
```

```
  Node Name     Status                    Comment
  ------------  ------------------------  ----------------------
  rac2          passed                    exists
  rac1          passed                    exists
Verifying Group Existence: asmdba ...PASSED
Verifying Group Existence: oinstall ...
  Node Name     Status                    Comment
  ------------  ------------------------  ----------------------
  rac2          passed                    exists
  rac1          passed                    exists
Verifying Group Existence: oinstall ...PASSED
Verifying Group Membership: asmdba ...
  Node Name         User Exists   Group Exists  User in Group  Status
  ----------------  ------------  ------------  ------------  ----------------
  rac2              yes           yes           yes           passed
  rac1              yes           yes           yes           passed
Verifying Group Membership: asmdba ...PASSED
Verifying Group Membership: asmadmin ...
  rac2              yes           yes           yes           passed
  rac1              yes           yes           yes           passed
Verifying Group Membership: asmadmin ...PASSED
Verifying Group Membership: oinstall(Primary) ...
  Node Name         User Exists   Group Exists  User in Group  Primary       Status
  ----------------  ------------  ------------  ------------  ------------  ------------
  rac2              yes           yes           yes           yes           passed
  rac1              yes           yes           yes           yes           passed
Verifying Group Membership: oinstall(Primary) ...PASSED
Verifying Run Level ...
  Node Name     run level                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          5                         3,5                       passed
  rac1          5                         3,5                       passed
Verifying Run Level ...PASSED
Verifying Hard Limit: maximum open file descriptors ...
  Node Name         Type          Available     Required      Status
  ----------------  ------------  ------------  ------------  ----------------
  rac2              hard          4096          65536         failed
  rac1              hard          4096          65536         failed
Verifying Hard Limit: maximum open file descriptors ...FAILED (PRVG-0446)
Verifying Soft Limit: maximum open file descriptors ...
```

```
  rac2             250           250           250           passed
Verifying OS Kernel Parameter: semmsl ...PASSED
Verifying OS Kernel Parameter: semmns ...
  Node Name         Current       Configured    Required      Status        Comment
  ----------------  ------------  ------------  ------------  ------------  ------------
  rac1             32000         32000         32000         passed
  rac2             32000         32000         32000         passed
Verifying OS Kernel Parameter: semmns ...PASSED
Verifying OS Kernel Parameter: semopm ...
  Node Name         Current       Configured    Required      Status        Comment
  ----------------  ------------  ------------  ------------  ------------  ------------
  rac1             100           100           100           passed
  rac2             100           100           100           passed
Verifying OS Kernel Parameter: semopm ...PASSED
Verifying OS Kernel Parameter: semmni ...
  Node Name         Current       Configured    Required      Status        Comment
  ----------------  ------------  ------------  ------------  ------------  ------------
  rac1             128           128           128           passed
  rac2             128           128           128           passed
Verifying OS Kernel Parameter: semmni ...PASSED
Verifying OS Kernel Parameter: shmmax ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          kmod(x86_64)-20-25.0.1.el7  kmod(x86_64)-20-21        passed
  rac1          kmod(x86_64)-20-25.0.1.el7  kmod(x86_64)-20-21        passed
Verifying Package: kmod-20-21 (x86_64) ...PASSED
Verifying Package: kmod-libs-20-21 (x86_64) ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          kmod-libs(x86_64)-20-25.0.1.el7  kmod-libs(x86_64)-20-21   passed
  rac1          kmod-libs(x86_64)-20-25.0.1.el7  kmod-libs(x86_64)-20-21   passed
Verifying Package: kmod-libs-20-21 (x86_64) ...PASSED
Verifying Package: binutils-2.23.52.0.1 ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          binutils-2.27-41.base.0.1.el7  binutils-2.23.52.0.1      passed
  rac1          binutils-2.27-41.base.0.1.el7  binutils-2.23.52.0.1      passed
Verifying Package: binutils-2.23.52.0.1 ...PASSED
Verifying Package: compat-libcap1-1.10 ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          compat-libcap1-1.10-7.el7  compat-libcap1-1.10       passed
  rac1          compat-libcap1-1.10-7.el7  compat-libcap1-1.10       passed
Verifying Package: compat-libcap1-1.10 ...PASSED
```

```
  rac1          compat-libcap1-1.10-7.el7  compat-libcap1-1.10       passed
Verifying Package: compat-libcap1-1.10 ...PASSED
Verifying Package: libgcc-4.8.2 (x86_64) ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          libgcc(x86_64)-4.8.5-39.0.1.el7  libgcc(x86_64)-4.8.2      passed
  rac1          libgcc(x86_64)-4.8.5-39.0.1.el7  libgcc(x86_64)-4.8.2      passed
Verifying Package: libgcc-4.8.2 (x86_64) ...PASSED
Verifying Package: libstdc++-4.8.2 (x86_64) ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          libstdc++(x86_64)-4.8.5-39.0.1.el7  libstdc++(x86_64)-4.8.2   passed
  rac1          libstdc++(x86_64)-4.8.5-39.0.1.el7  libstdc++(x86_64)-4.8.2   passed
Verifying Package: libstdc++-4.8.2 (x86_64) ...PASSED
Verifying Package: libstdc++-devel-4.8.2 (x86_64) ...
  Node Name     Available                 Required                  Status
  ------------  ------------------------  ------------------------  ----------
  rac2          libstdc++-devel(x86_64)-4.8.5-39.0.1.el7  libstdc++-devel(x86_64)-4.8.2  pa
  rac1          libstdc++-devel(x86_64)-4.8.5-39.0.1.el7  libstdc++-devel(x86_64)-4.8.2  pa
Verifying Package: libstdc++-devel-4.8.2 (x86_64) ...PASSED
Verifying Package: sysstat-10.1.5 ...
Verifying Network Time Protocol (NTP) ...PASSED
Verifying Same core file name pattern ...PASSED
Verifying User Mask ...
  Node Name     Available                 Required                  Comment
  ------------  ------------------------  ------------------------  ----------
  rac2          0022                      0022                      passed
  rac1          0022                      0022                      passed
Verifying User Mask ...PASSED
Verifying User Not In Group "root": grid ...
  Node Name     Status                    Comment
  ------------  ------------------------  ------------------------
  rac2          passed                    does not exist
  rac1          passed                    does not exist
Verifying User Not In Group "root": grid ...PASSED
Verifying Time zone consistency ...PASSED
Verifying Time offset between nodes ...PASSED
Verifying resolv.conf Integrity ...
  Node Name                             Status
  ------------------------------------  ------------------------
  rac1                                  passed
  rac2                                  passed
checking response for name "rac1" from each of the name servers specified in
"/etc/resolv.conf"
```

```
      PRVG-11095 : The TCP system call "connect" failed with error "111" while
      executing exectask on node "rac1"
      Connection refused
Verifying Daemon "avahi-daemon" not configured and running ...FAILED
rac2: PRVG-1359 : Daemon process "avahi-daemon" is configured on node "rac2"
rac2: PRVG-1360 : Daemon process "avahi-daemon" is running on node "rac2"
rac1: PRVG-1359 : Daemon process "avahi-daemon" is configured on node "rac1"
rac1: PRVG-1360 : Daemon process "avahi-daemon" is running on node "rac1"
Verifying RPM Package Manager database ...INFORMATION
PRVG-11250 : The check "RPM Package Manager database" was not performed because
it needs 'root' user privileges.
Verifying /dev/shm mounted as temporary file system ...FAILED
rac2: PRVE-0426 : The size of in-memory file system mounted as /dev/shm is
      "1843" megabytes which is less than the required size of "2048" megabytes
      on node "discoveryrac2"
rac1: PRVE-0426 : The size of in-memory file system mounted as /dev/shm is
      "1843" megabytes which is less than the required size of "2048" megabytes
      on node "discoveryrac1"
CVU operation performed:      stage -pre crsinst
Date:                         Oct 6, 2024 7:39:07 AM
CVU home:                     /u01/app/19/grid/
User:                         grid
```

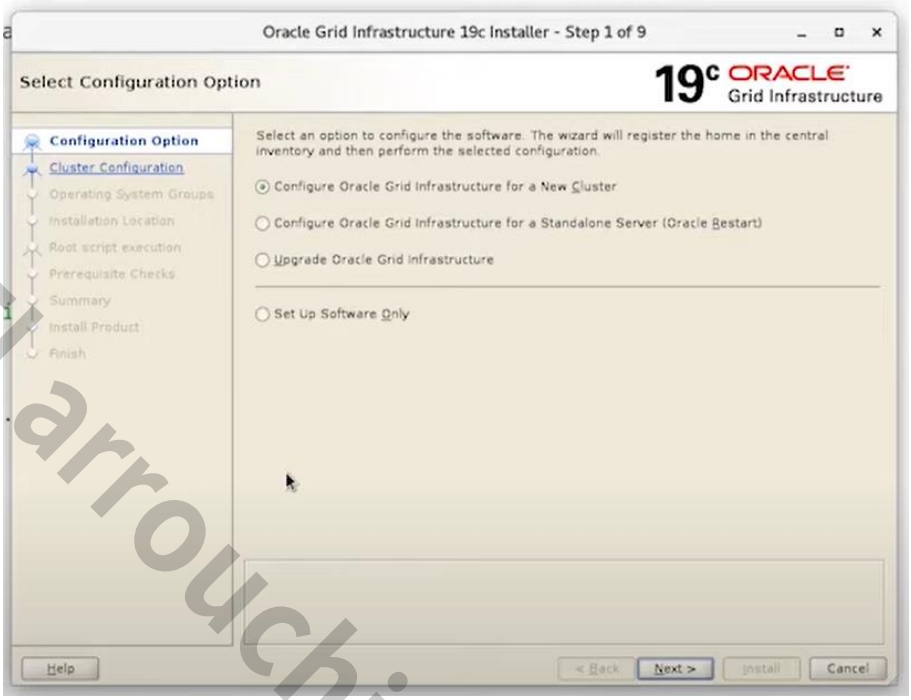
5. Grid Installation in GUI Mode:

Step1: Start the grid infrastructure installation

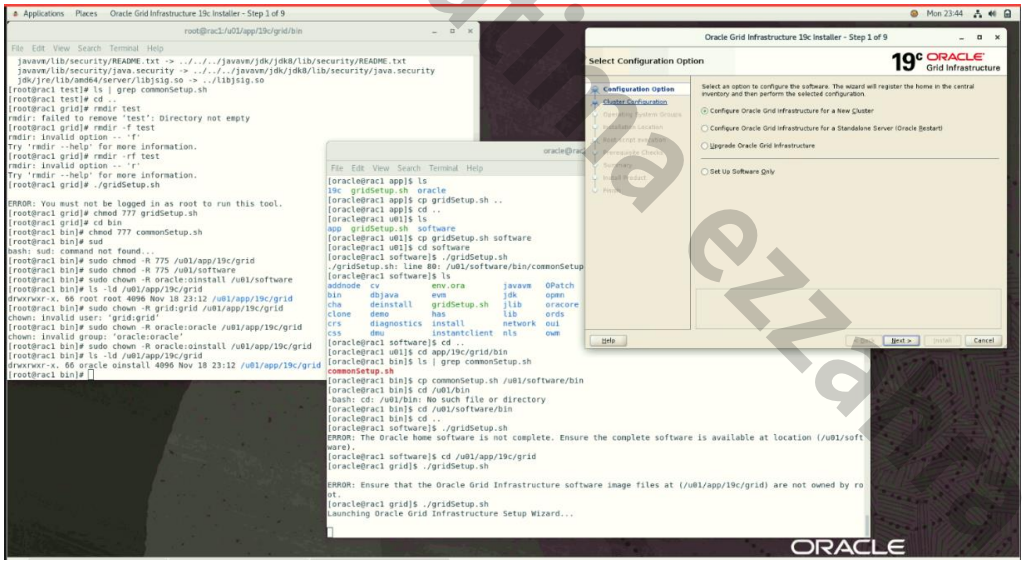
To start the installation process, we ran the following command:

```
./gridSetup.sh
```

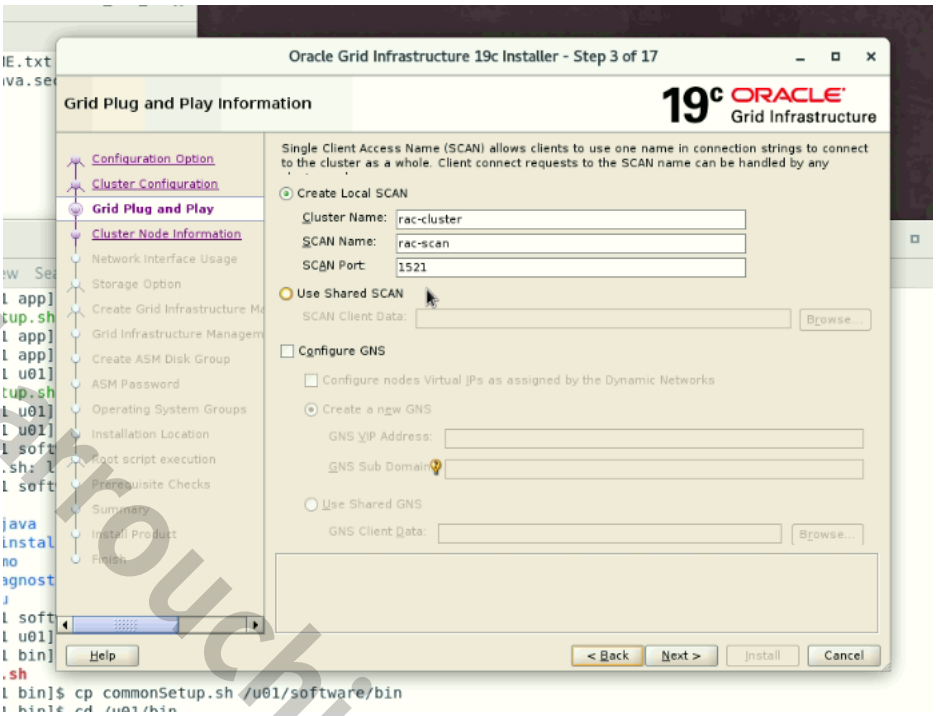
On the first screen, we selected **Configure Oracle Grid Infrastructure for a New Cluster** and click **Next** to proceed with the installation



We selected **Configure an Oracle Standalone Cluster** and clicked **Next** to continue with the configuration process

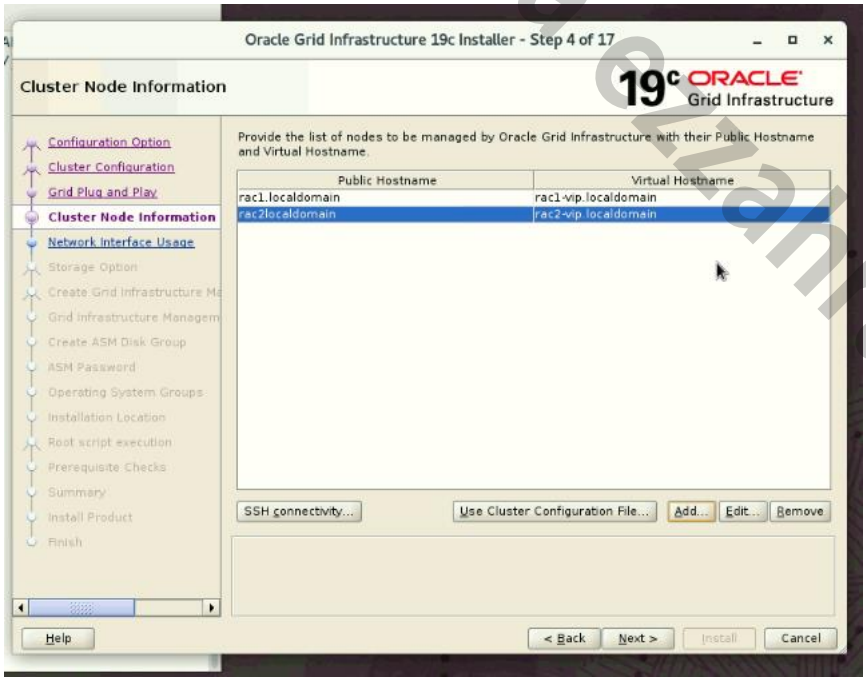


Then we changed the SCAN name and Cluster name to rac-scan and rac-cluster accordingly

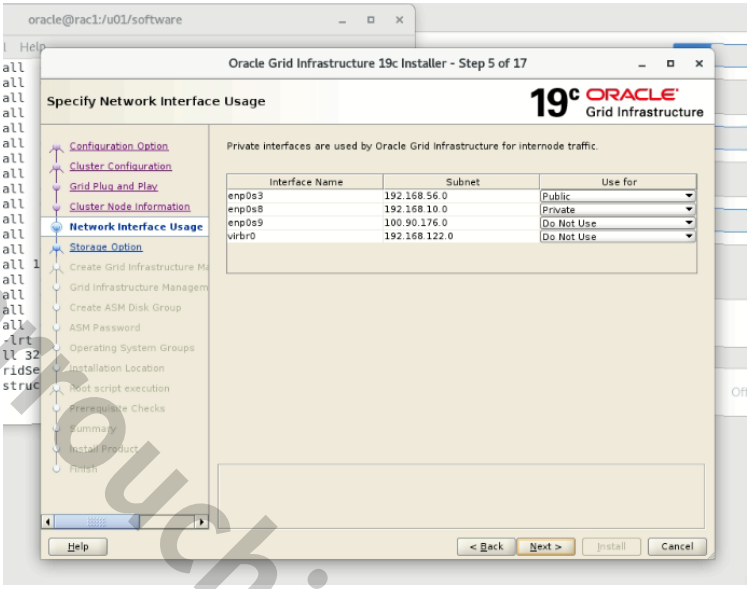


On the **Cluster Node Information** screen, we clicked the **Add** button, entered the details for the second node in the cluster, then clicked

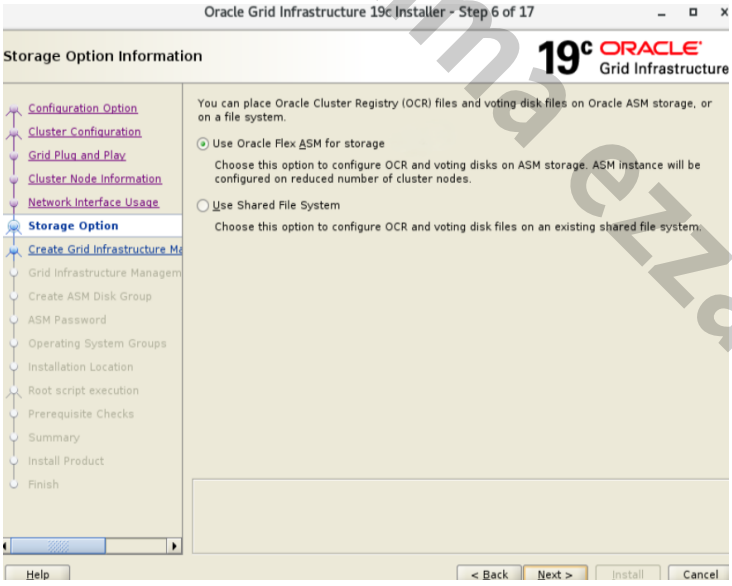
the **OK** button to save .



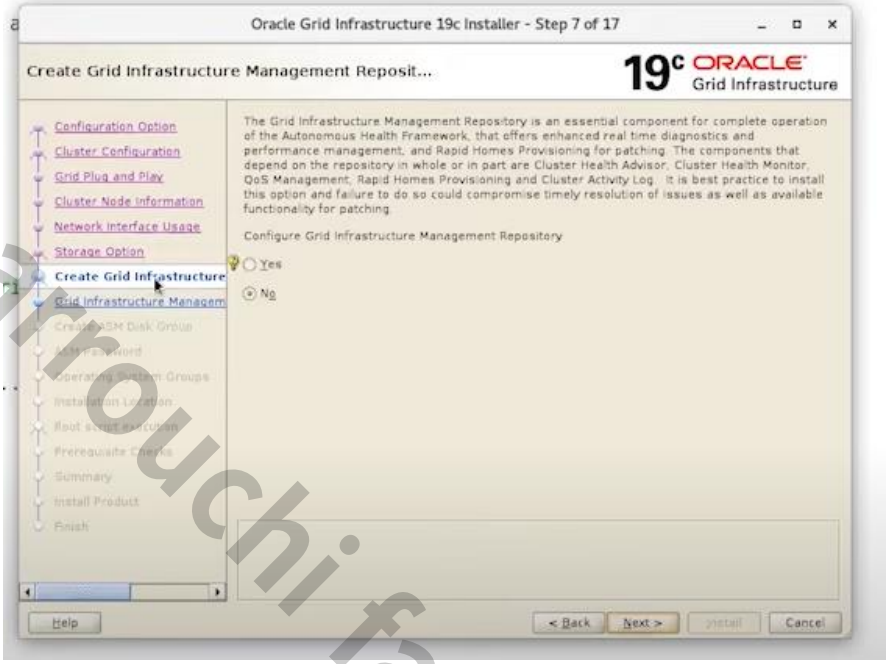
Then we specified **Network Interface Usage**. For the public network, we select the public option, and for the private network, we select ASM & Private.



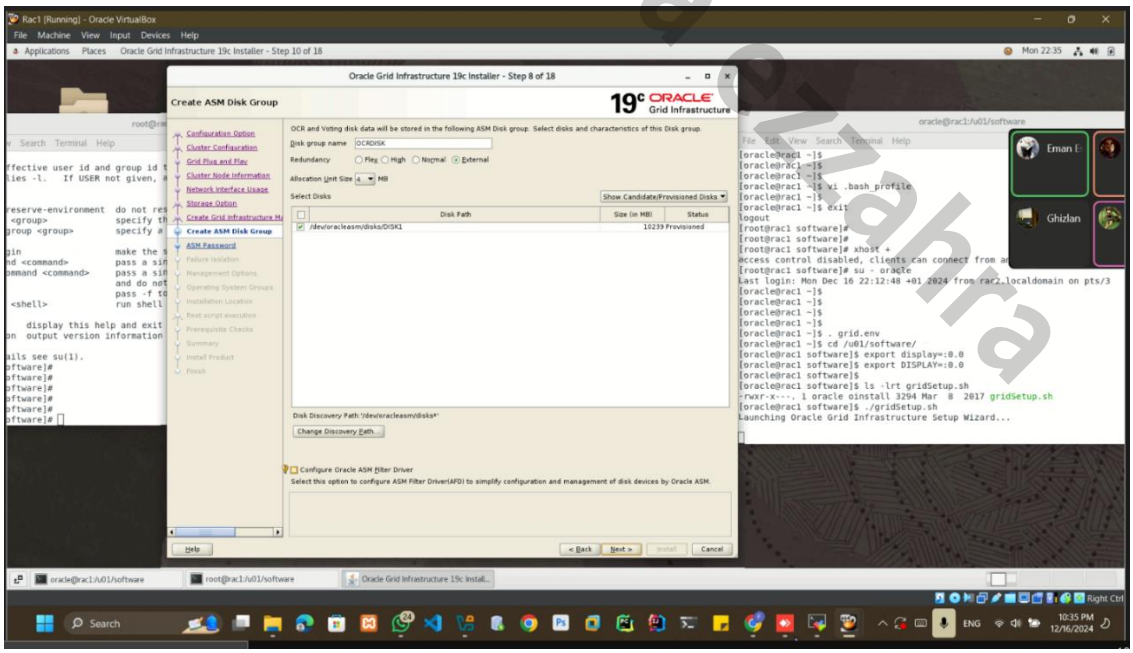
We selected the Oracle Flex ASM for the storage option and then click the Next button to proceed.



We select No to indicate that you prefer not to create a separate disk group for the Grid Infrastructure Management Repository

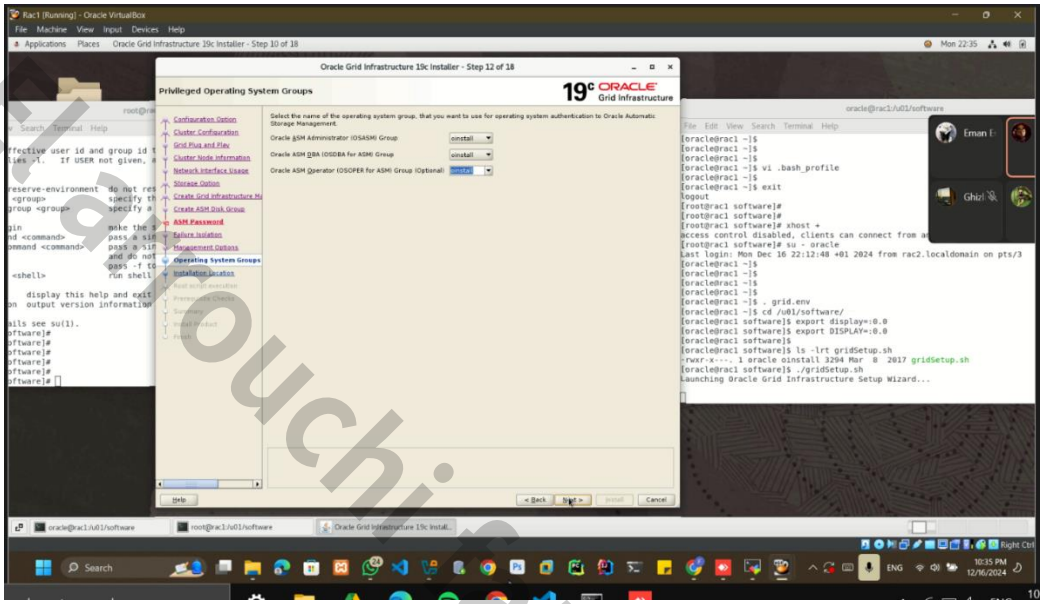


Then we create ASM Disk Group by selecting the Discovery Path and added the path where the disks are located.

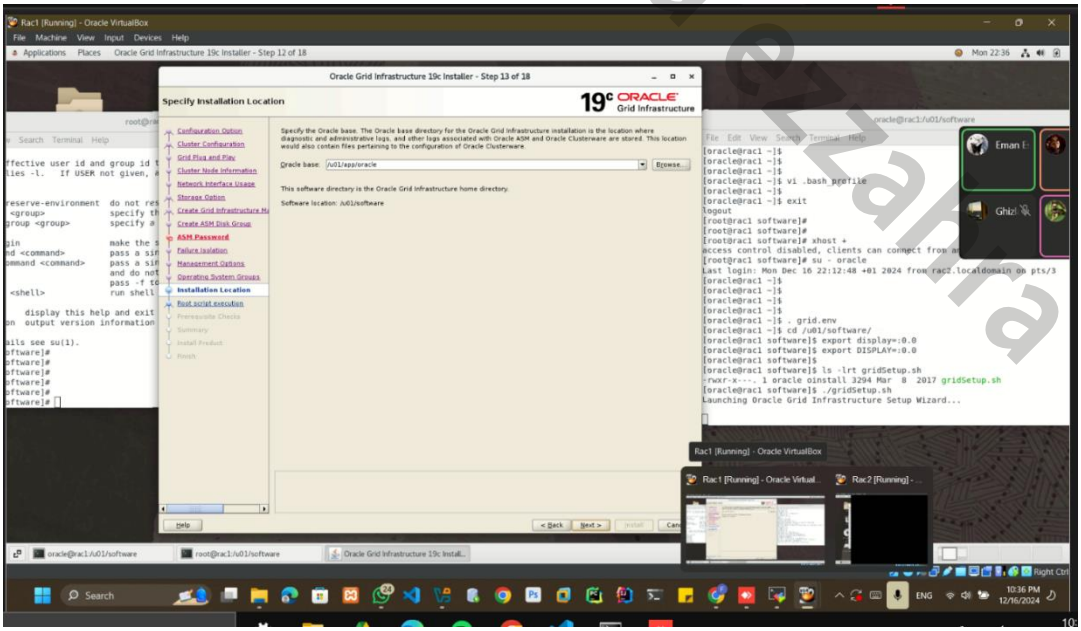


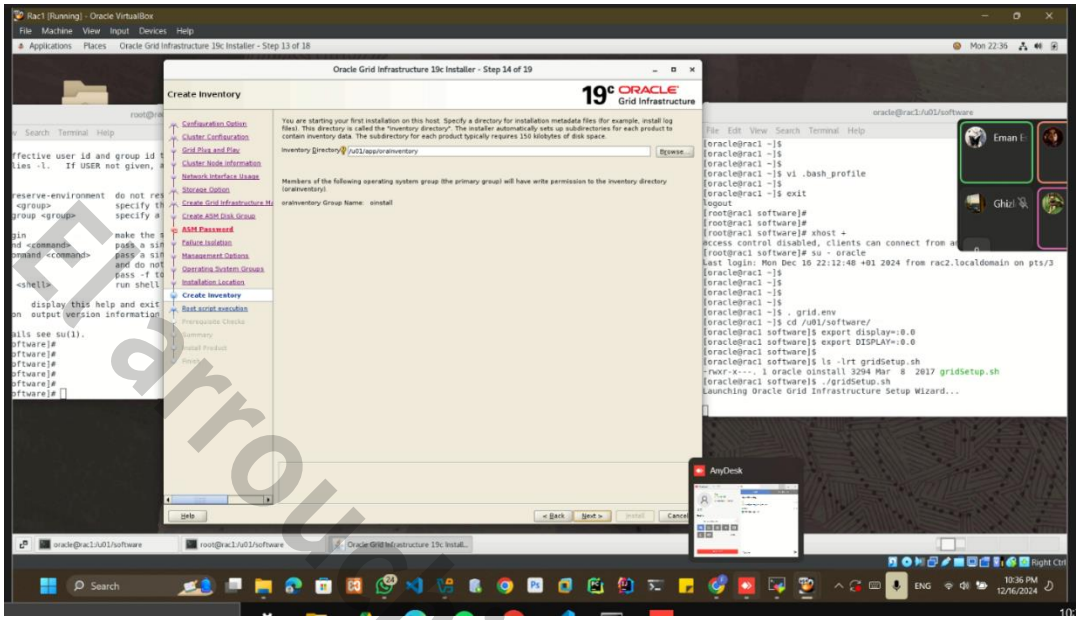
We selected the default option for failure isolation support then clicked Next button. Unchecked the Enterprise Manager (EM) option, and then click the Next button to proceed.

We used the default settings for the privileged operating system groups.

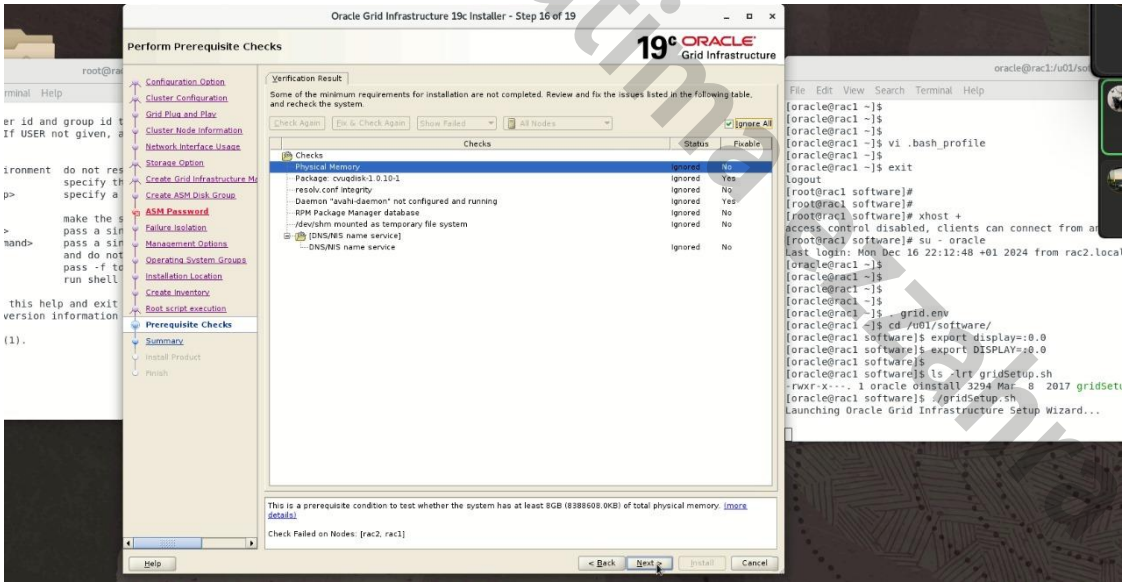


After that, we chose the installation and inventory path.

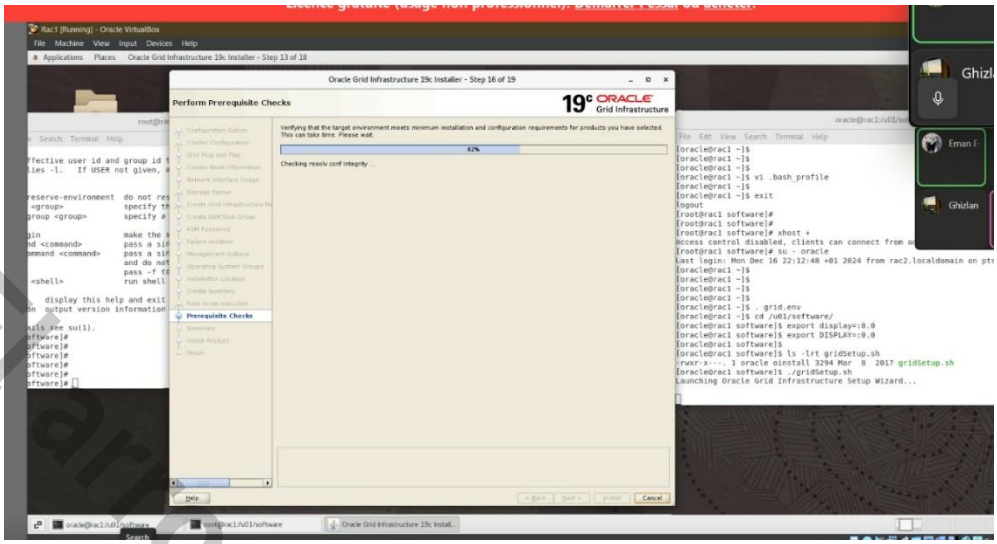




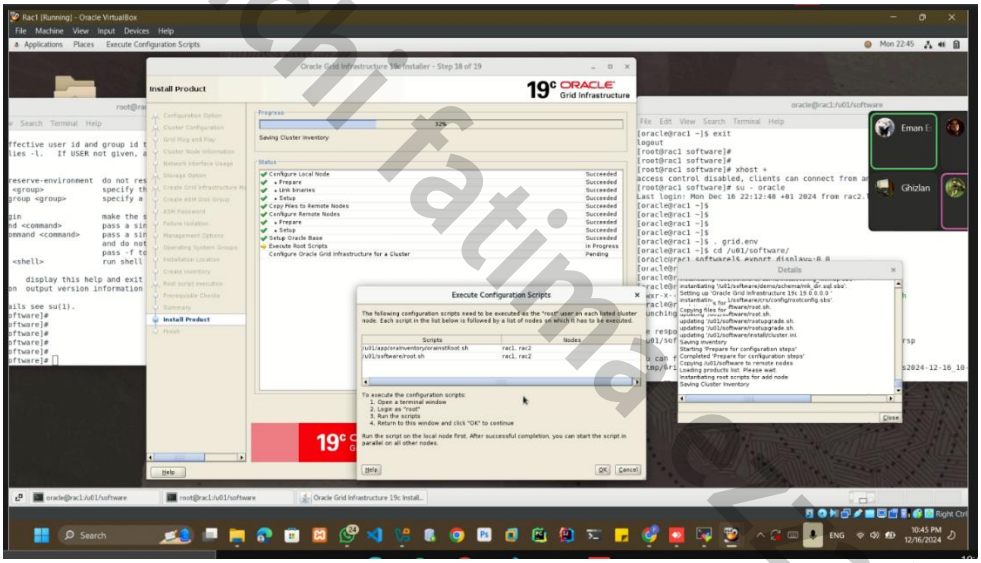
Pressed ignore on the checks and made sure everything was correct



And waited for the verification to be done successfully.



Then the installation was launched and done successfully.



This step ensures that my installation process is on track.

Node 1

[root@rac1 grid]# /u01/app/oraInventory/orainstRoot.sh

Node 2

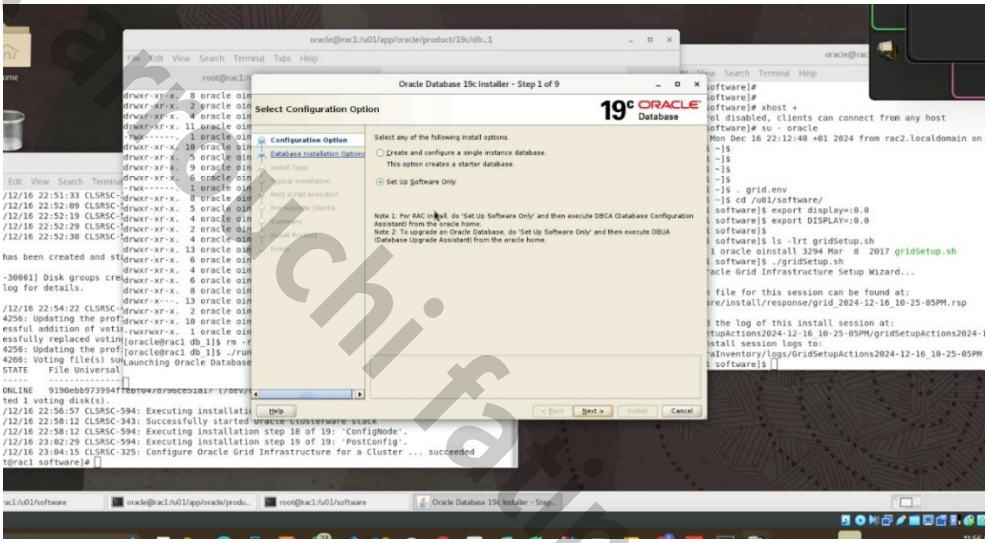
[root@rac2 grid]# /u01/app/oraInventory/orainstRoot.sh

Step 2: Installing Oracle Database 19c

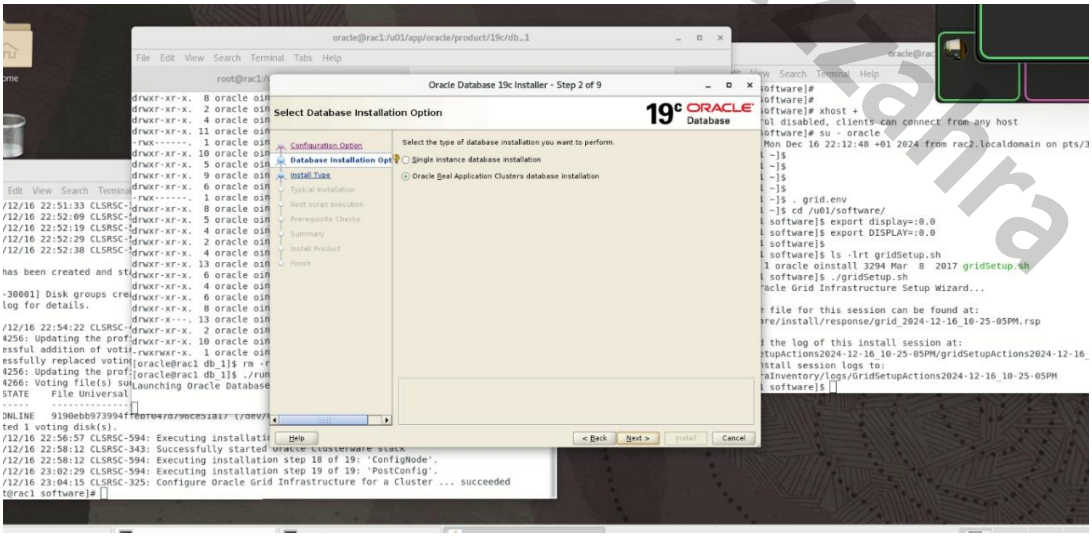
We start by navigating to the **Oracle Home Bin Directory and** executing the following command to start the Database Configuration Assistant:

```
./dbca
```

This will launch the DBCA, allowing us to configure the Oracle database.

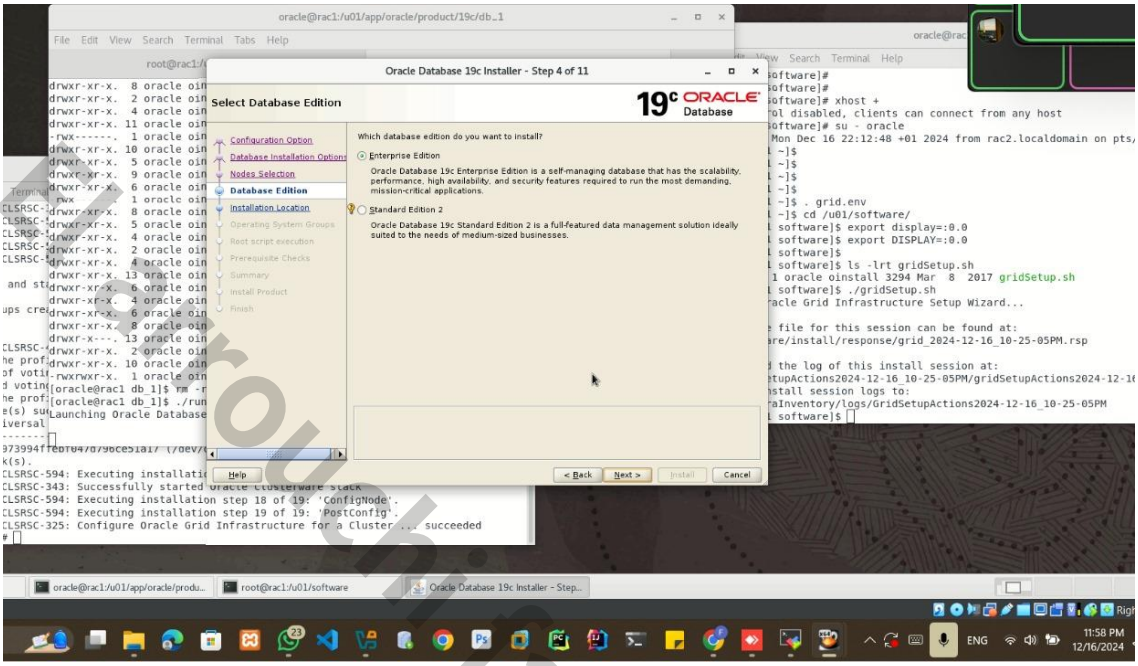


We check the **oracle RAC database installation** since the goal of this project is to have a clustered database

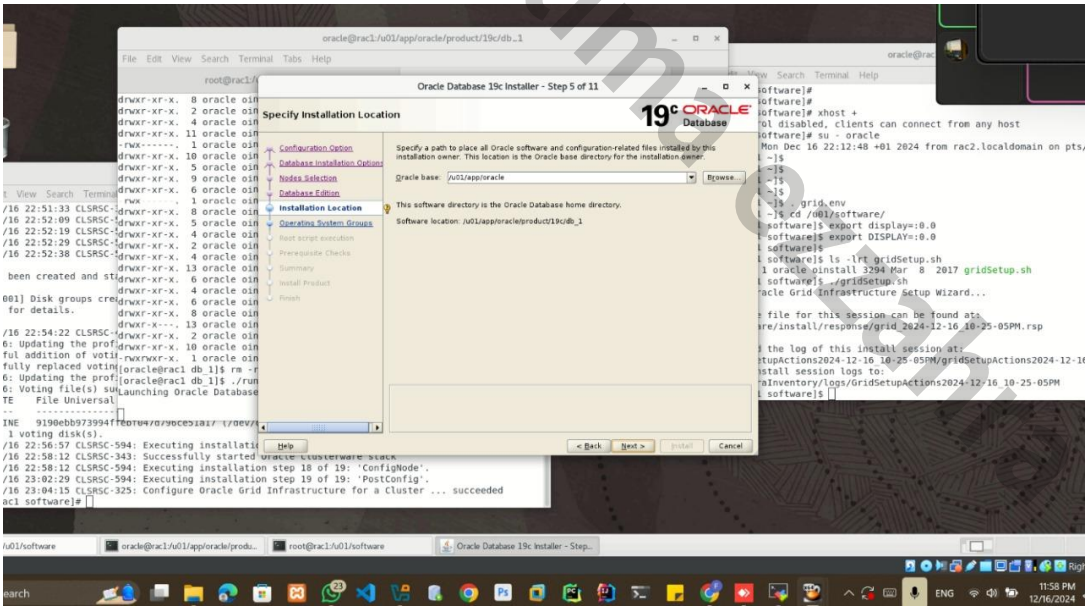


we specify the nodes and click Next.

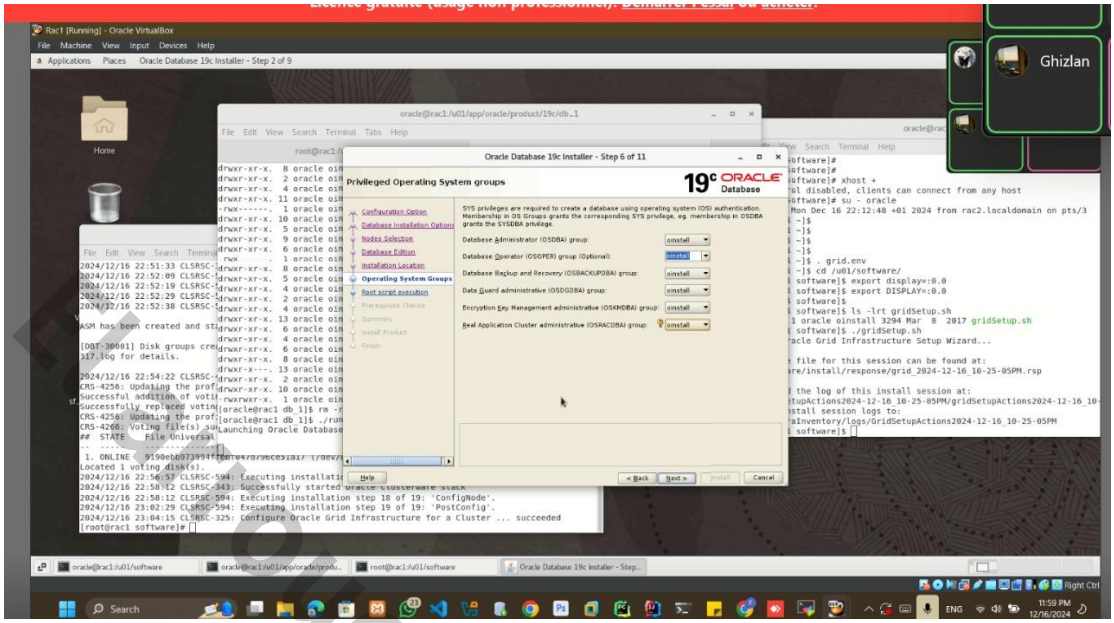
Then we specify what database edition we will need



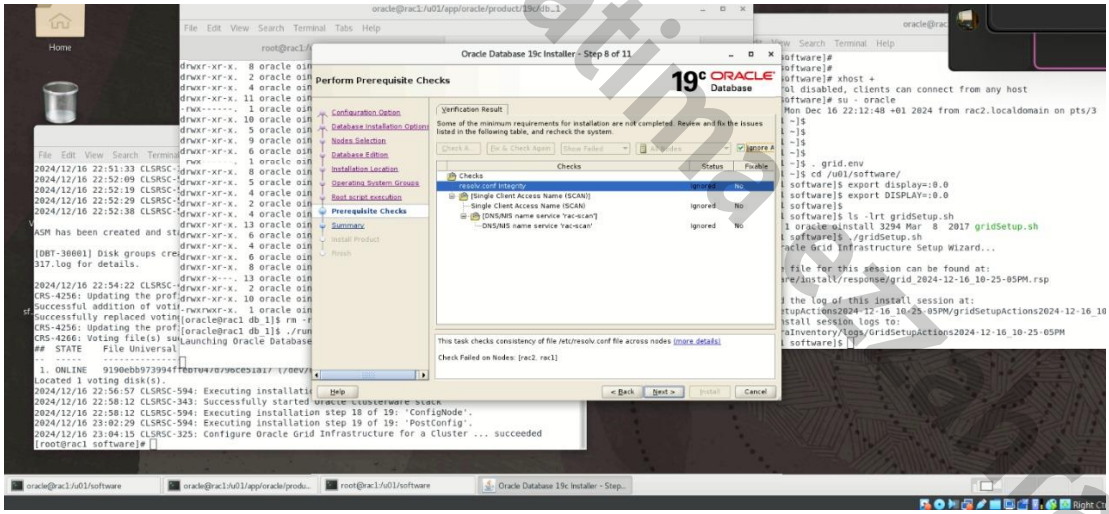
We specify the installation path and click Next.



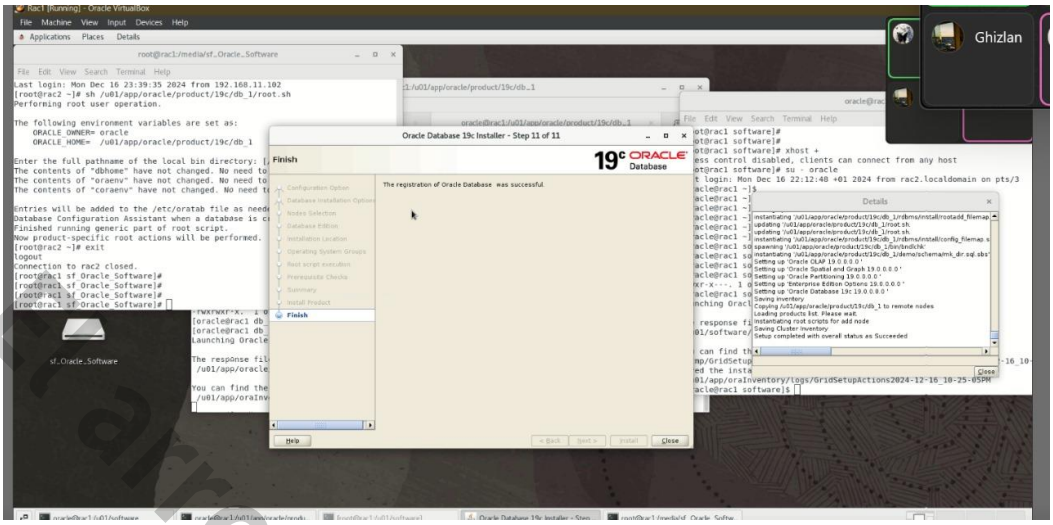
Then we choose the appropriate group according to the configuration and environment.



Since we haven't set up the SCAN and DNS configurations yet, we'll ignore this step for now.



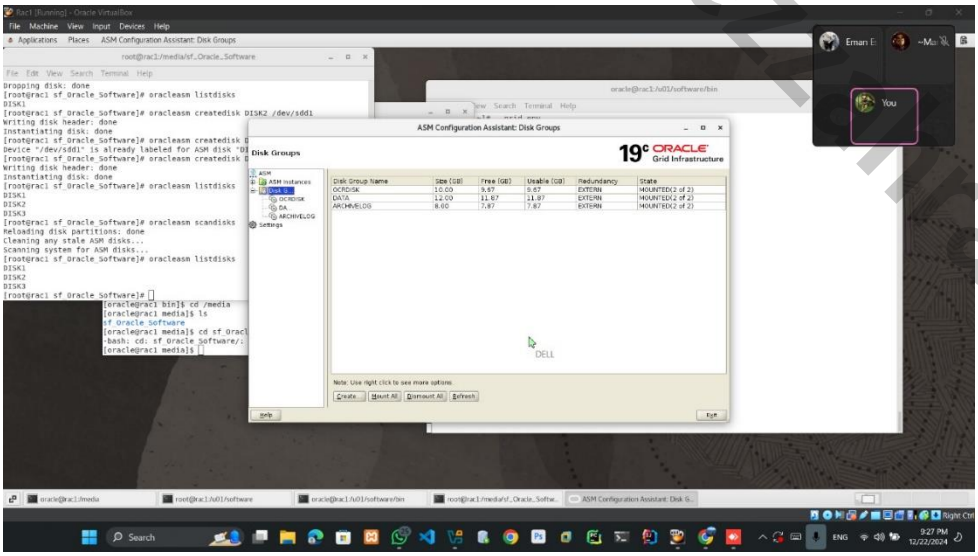
After checking the summary and making sure there are no issues we started the installation that was successful



Step3: Creating ASM Diskgroup using ASMCA

To create ASM diskgroupe we followed these steps:

- Click on **“Disk Groups”** in the left pane Once ASMCA opens.
- Click on **“Create”** to create a new disk group.
- Select the **PATH** of the ASM disk group.
- In the **Create Disk Group** window, enter a name for your disk group.
- Select the redundancy type as **“External.”**
- Click **“OK”** to proceed



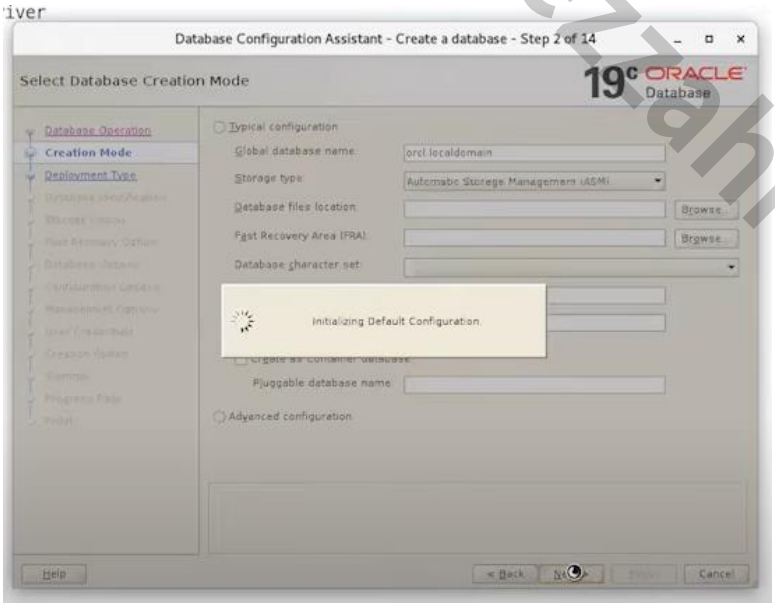
Step4: Installing Oracle Database 19c

We start by navigating to the **Oracle Home Bin Directory and** executing the following command to start the Database Configuration Assistant:

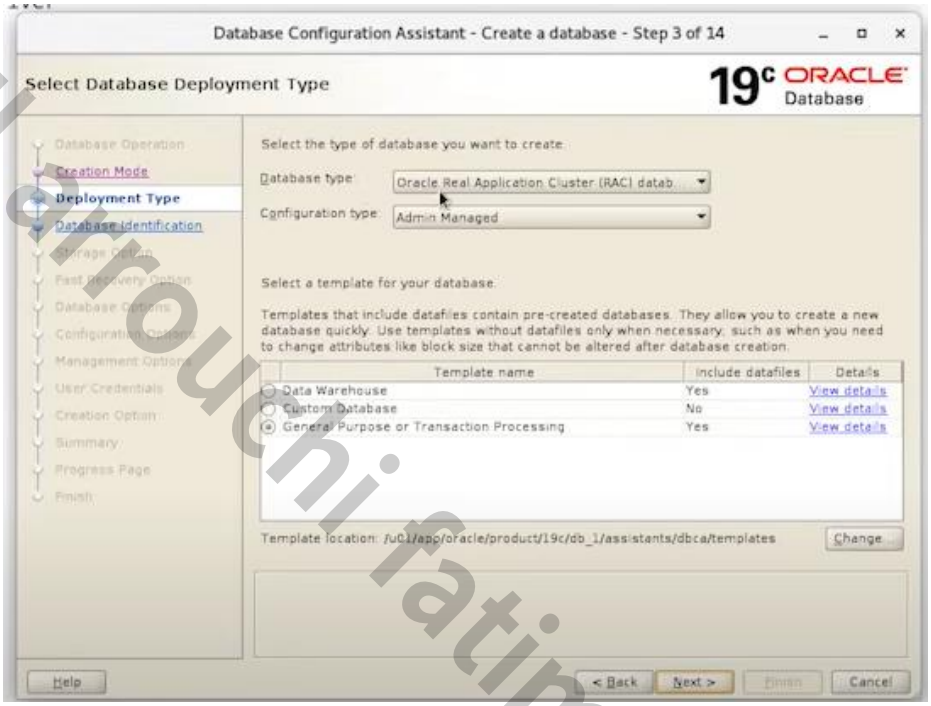
```
./dbca
```

In the DBCA welcome screen, we selected **“Create Database”** as the operation we want to perform, then click **“Next.”**

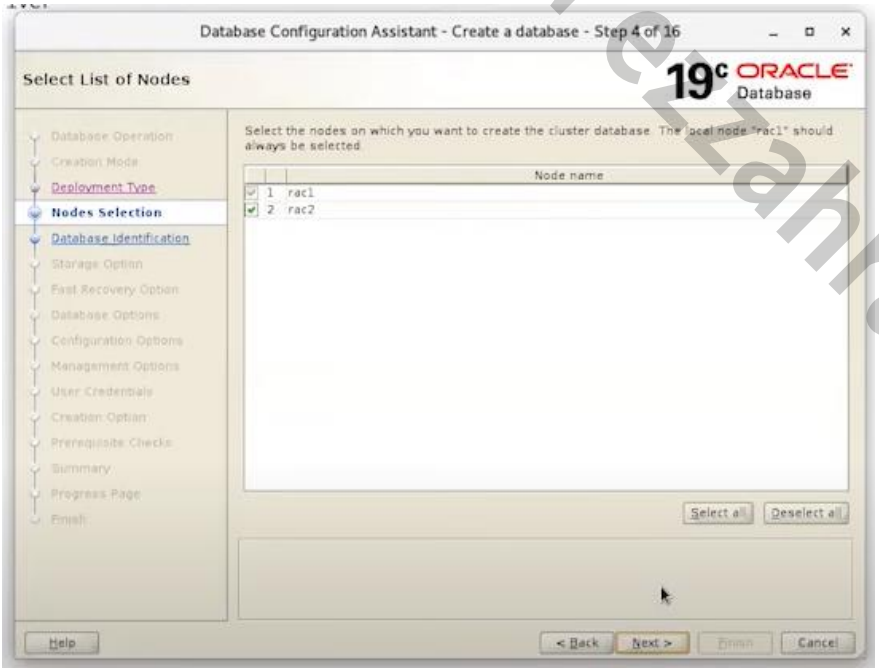
in the creation mode we chose advanced



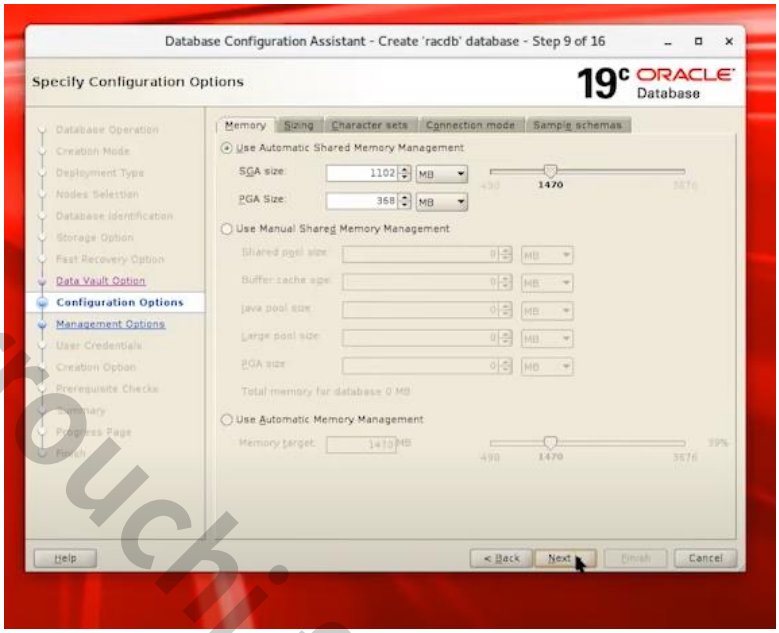
Select **“General Purpose”**, then click next



We select the node on which we want to create the cluster database.

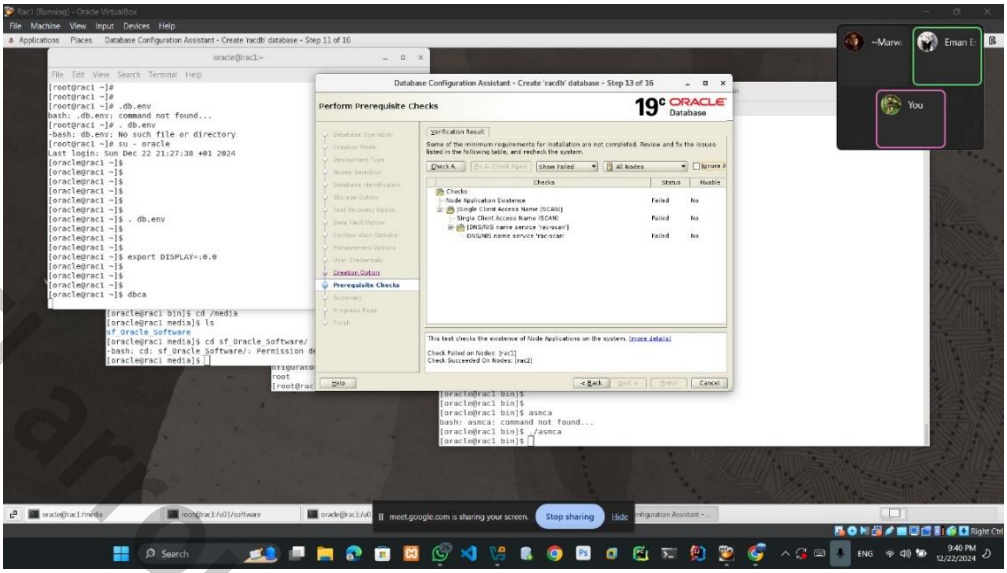


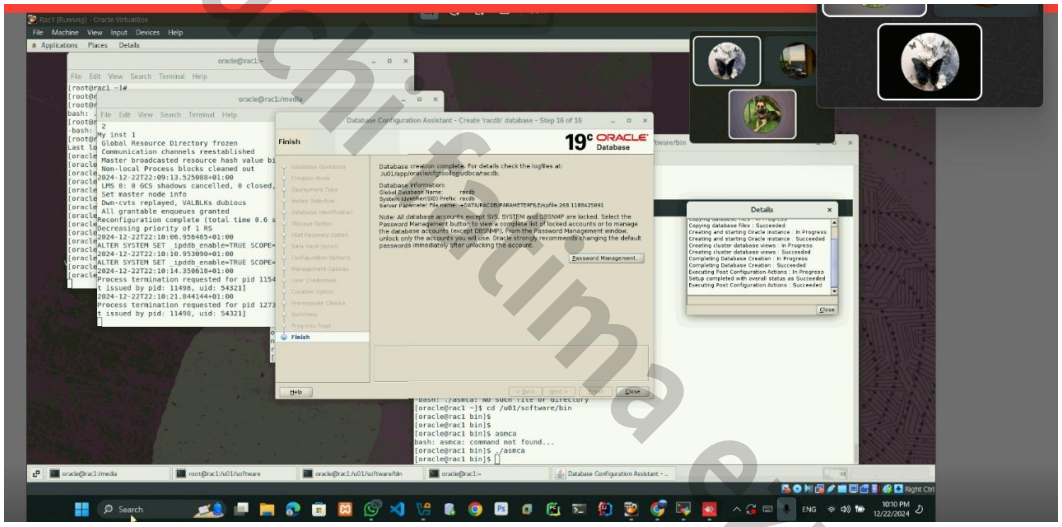
In this step we had to assign the size for the SGA (System Global Area) and PGA (Program Global Area) as needed.



for the Database Creation Options we chose the default settings .

the last steps were to check the prerequisites and finish the installation





This was the final check to make sure that both the instances we created are working successfully

```
[oracle@rac1 trace]$ srvctl status database -d racdb
Instance racdb1 is running on node rac1
Instance racdb2 is running on node rac2
[oracle@rac1 trace]$ ps -ef |grep pmon
oracle    3934     1  0 02:08 ?        00:00:03 asm_pmon_+ASM1
oracle   17536     1  0 04:42 ?        00:00:00 ora_pmon_racdb1
oracle   21220  1756  0 04:49 pts/0    00:00:00 grep --color=auto pmon
[oracle@rac1 trace]$
[oracle@rac1 trace]$ srvctl status database -d racdb
Instance racdb1 is running on node rac1
Instance racdb2 is running on node rac2
[oracle@rac1 trace]$ srvctl status database -d racdb -v
Instance racdb1 is running on node rac1. Instance status: Open.
Instance racdb2 is running on node rac2. Instance status: Open.
[oracle@rac1 trace]$
[oracle@rac1 trace]$
[oracle@rac1 trace]$
```

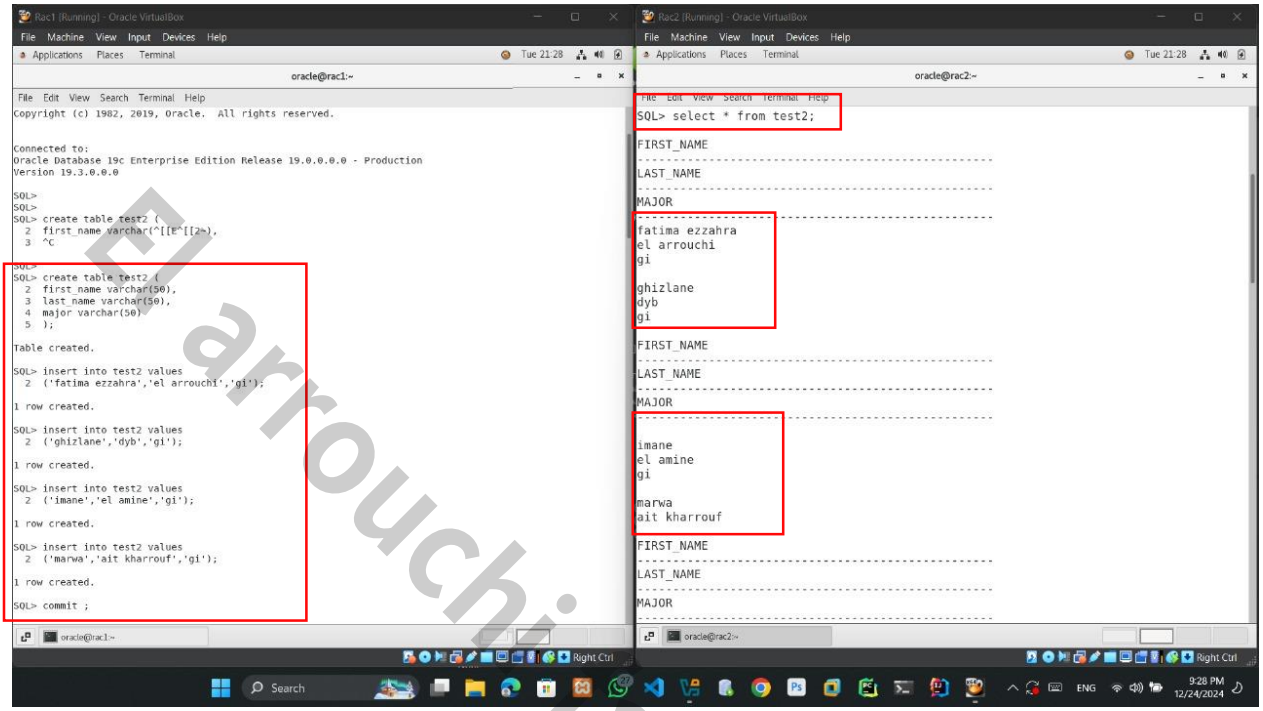
Step5: test

we can view the instances connected to the database using this command

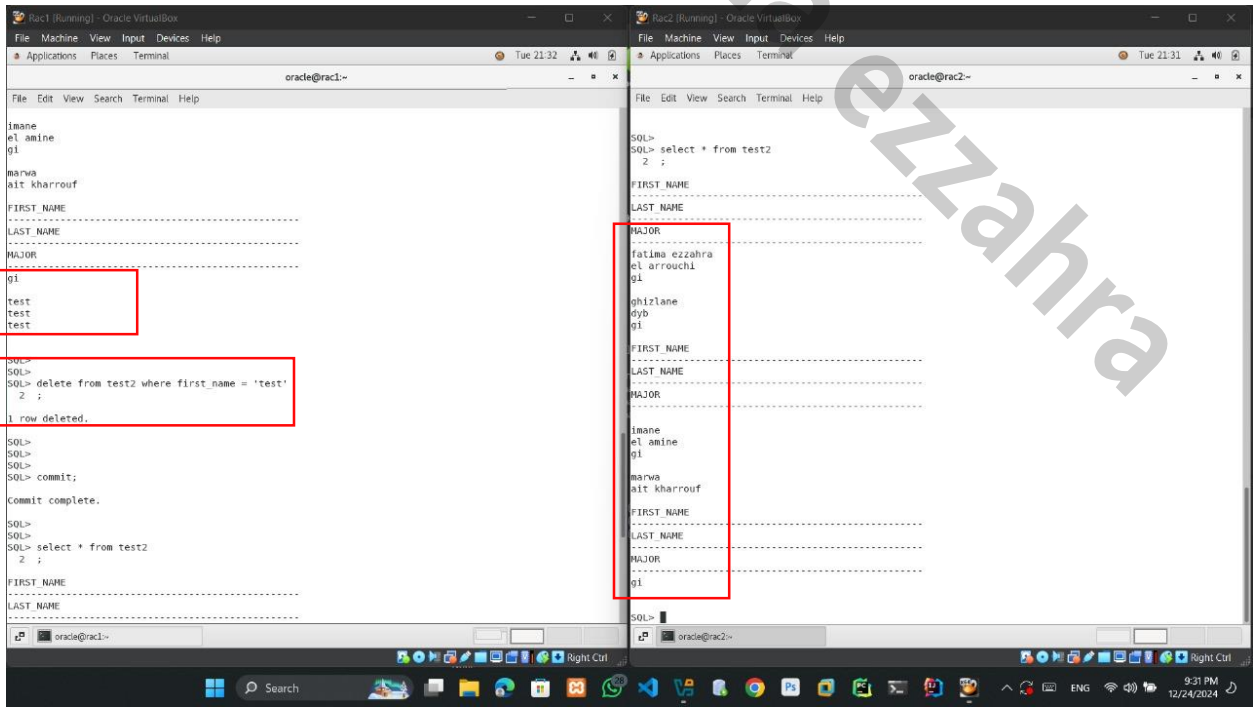
```
SQL>
SQL>
SQL> select inst_id, instance_name, status from gv$instance
  2  ;

   INST_ID INSTANCE_NAME    STATUS
---------- ---------------- ------------
         1 racdb1           OPEN
         2 racdb2           OPEN

SQL>
SQL>
```

After that we connected to the database and create a test table in the 1st instance and insert into it then we viewed the content of the test table in the 2nd instance.

And as we can see, the 2 instances are connected to the same database which is why we can see the table that was created .



We can also see the changes made in the database by one of the instances for example we have:



```
The user test is removed
```

6. Common Issues and Solutions During Oracle RAC Setup

1. Files Without Proper Permissions

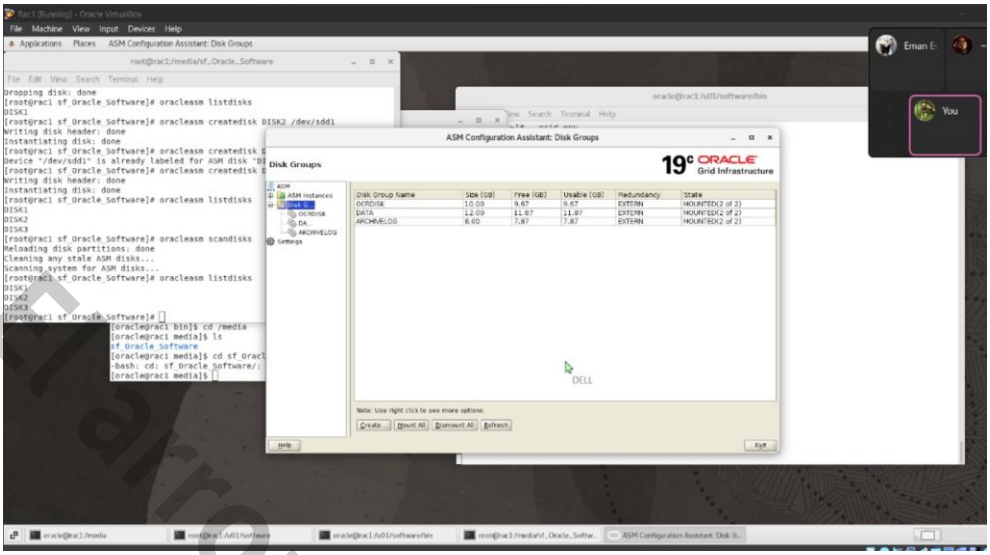
- **Issue**: During the configuration process, we discovered that some critical files did not have the necessary permissions, causing certain operations to fail.
- **Root Cause**: This often occurs when files are not set with the correct ownership or permission levels during setup.
- **Solution**:We used the `chmod` and `chown` commands to fix file permissions.

2. Installation Package Dependencies

- **Issue**: Missing or outdated packages can cause installation failures or errors during prerequisites checks.
- **Solution**:We ensured all required packages were installed using `yum install` or by running a pre-install script provided by Oracle.

3. Forgot to Execute `scandisks` and `listdisks` Commands on Both Nodes

- **Issue**: Initially, we only ran the `scandisks` and `listdisks` commands on one RAC node, which caused issues with shared disk detection.
- **Solution**:
  - We deleted the existing partitions, data, and archive logs to resolve the inconsistencies.
  - Afterward, we recreated the partitions, data, and archive logs, ensuring that the state was correctly displayed as **2 of 2** (fully mounted) instead of **1 of 2** (partially mounted).



Then, we executed the following commands on both nodes to re-scan and verify the shared disks:

- **Scandisks**
- **Listdisks**

Conclusion:

In this project, we dedicated significant time, effort, and resources to successfully implement a fully functional Oracle RAC cluster. This required a systematic approach that combined planning, research, and hands-on execution. From setting up the virtual environment to configuring the RAC nodes and resolving technical challenges, every step presented unique learning opportunities.

We began by carefully setting up the prerequisites, such as creating virtual machines, configuring shared storage, and establishing reliable network communication. Each configuration required precision to ensure compatibility with the Oracle RAC architecture. Along the way, we encountered challenges, such as file permission issues, misconfigured partitions, and network inconsistencies. Through persistent troubleshooting and collaboration, we resolved these issues by applying innovative solutions, such as recreating partition structures, reconfiguring IP settings, and verifying shared disk states across nodes.

The experience enhanced not only our technical understanding of Oracle RAC but also our ability to manage complex projects effectively. We gained practical insights into configuring high-availability systems, ensuring data integrity, and optimizing performance in multi-node environments. Furthermore, this project helped us develop critical skills in problem-solving, resource management, and teamwork, all of which are essential in professional IT environments.