

# MySQL Enterprise Edition - Implementation Essentials Bootcamp



## Introduction

### About this Workshop

Securing the data stored in your MySQL Server is key towards avoiding breaches and achieving Regulatory Compliance. This workshop covers the installation, configuration and testing of 2 of the MySQL Security Based Enterprise Features. We will go through how to setup and run Enterprise Audit and Transparent Data Encryption.

Estimated Workshop Time: 1 hours 30 minutes (This estimate is for the entire workshop - it is the sum of the estimates provided for each of the labs included in the workshop.)

Your Free Tier server should be accessible for a couple of days after this workshop so what you do not finish when following instruction will be able to be covered later at your own pace.

### Objectives

In this workshop, you will learn how to work with

- MySQL Enterprise Edition
- MySQL Shell
- MySQL Enterprise Audit
- MySQL Enterprise Transparent Data Encryption

### Learn More

- [MySQL Home Page](#)
- [MySQL Enterprise Audit](#)

- [MySQL Enterprise Transparent Data Encryption](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - Dale Dasker, April 2022

### Web Clip

# Create your Virtual Cloud Network and Related Components

## Introduction

### Create your VCN and subnets

Set up a Virtual Cloud Network (VCN) to connect your Linux instance to the internet. You will configure all the components needed to create your virtual network.

*Estimated Time:* 10 minutes

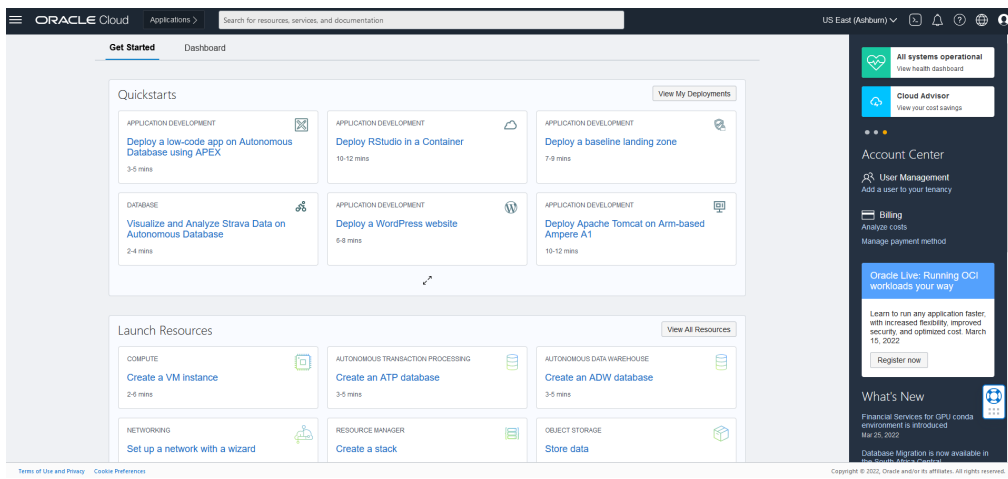
## Objectives

In this lab, you will be guided through the following tasks:

- Create Virtual Cloud Network
- Configure security list to allow MySQL incoming connections

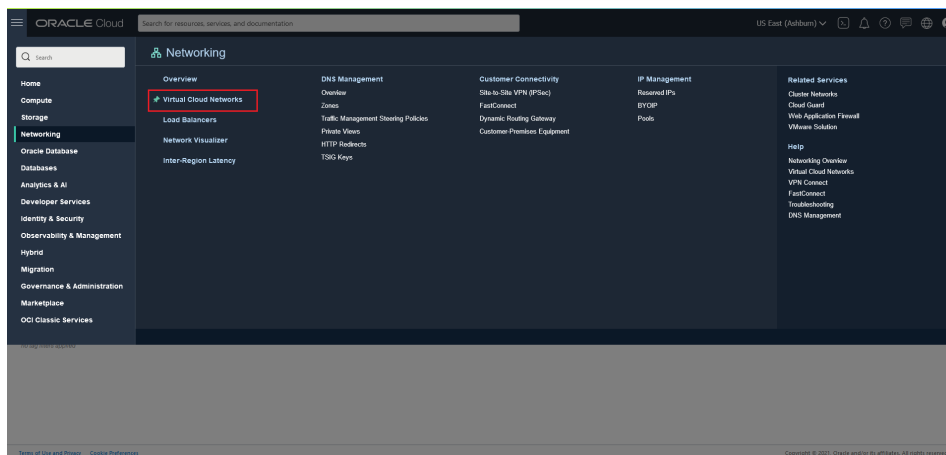
## Prerequisites

- An Oracle Free Tier or Paid Cloud Account
- A web browser
- Login to OCI to land on OCI Dashboard (This image shows a trial account)

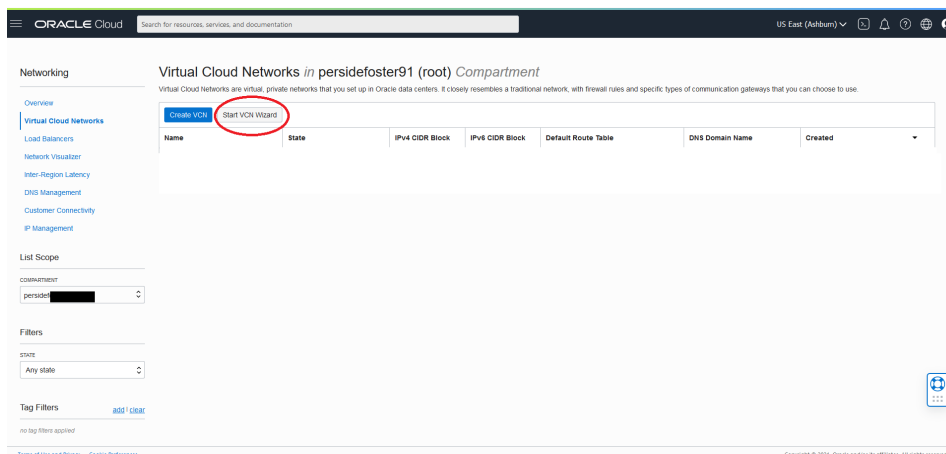


## Task 1: Create Virtual Cloud Network

1. Click Navigation Menu  
Select Networking  
Select Virtual Cloud Networks

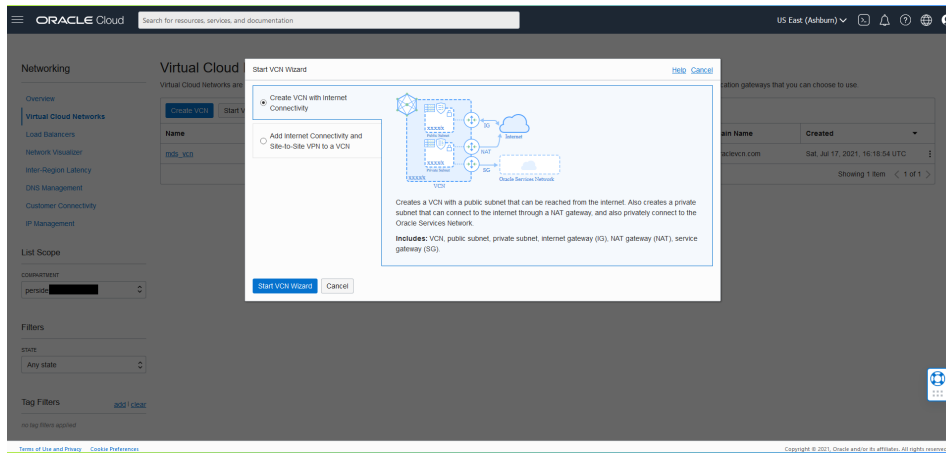


2. Click **Start VCN Wizard**



### 3. Select 'Create VCN with Internet Connectivity'

Click 'Start VCN Wizard'



### 4. Create a VCN with Internet Connectivity

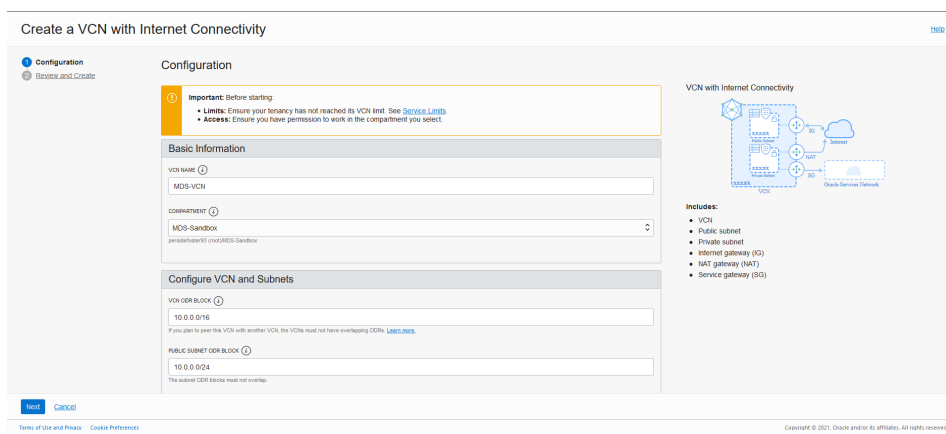
On Basic Information, complete the following fields:

VCN Name:

myvcn

Compartment: Select **(root)**

Your screen should look similar to the following



### 5. Click 'Next' at the bottom of the screen

### 6. Review Oracle Virtual Cloud Network (VCN), Subnets, and Gateways

Click 'Create' to create the VCN

Create a VCN with Internet Connectivity

Configuration

Review and Create

Configuration

Important: Before starting

- **Limits:** Ensure your tenancy has not reached its VCN limit. See [Service Limits](#).
- **Access:** Ensure you have permission to work in the compartment you select.

Basic Information

VCN Name

MDS-VCN

Compartment

MDS-Sandbox

Configure VCN and Subnets

VCN CIDR BLOCK

10.0.0.0/16

PUBLIC SUBNET CIDR BLOCK

10.0.0.0/24

VCN with Internet Connectivity

Includes:

- VCN
- Public subnet
- Private subnet
- Internet gateway (IG)
- NAT gateway (NAT)
- Service gateway (SG)

7. The Virtual Cloud Network creation is completing

ORACLE Cloud

Create a VCN with Internet Connectivity

Configuration

Review and Create

Creating Resources

Virtual Cloud Network creation complete

Create Virtual Cloud Network (1 resolved) Done

Create Subnets (2 resolved) Done

Create Internet Gateway (1 resolved) Done

Create NAT Gateway (1 resolved) Done

Create Service Gateway (1 resolved) Done

Create Route Table for Private Subnet (1 resolved) Done

Create Security List for Private Subnet (1 resolved) Done

Update Route Tables (2 resolved) Done

VCN with Internet Connectivity

Includes:

- VCN
- Public subnet
- Private subnet
- Internet gateway (IG)
- NAT gateway (NAT)
- Service gateway (SG)

8. Click 'View Virtual Cloud Network' to display the created VCN

ORACLE Cloud

Networking > Virtual Cloud Networks > Virtual Cloud Network Details

MDS-VCN

VCN Information

Compartment: MDS-Sandbox

Created: Wed, May 12, 2021, 19:04:13 UTC

IPv4 CIDR Block: 10.0.0.0/16

IPv6 CIDR Block: No value

OCID: ...

Default Route Table: Default Route Table for MDS-VCN

Default Domain Name: mdsvcn.oraclecloud.com

Resources

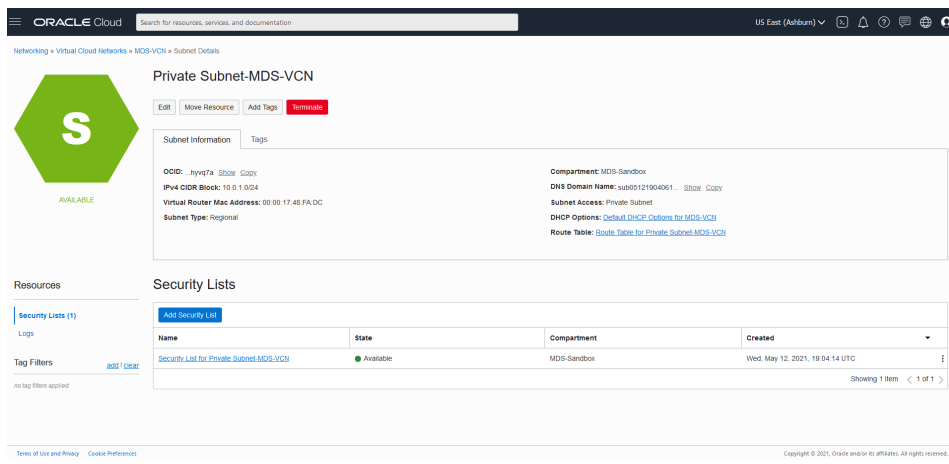
Subnets (2)

Subnets in MDS-Sandbox Compartment

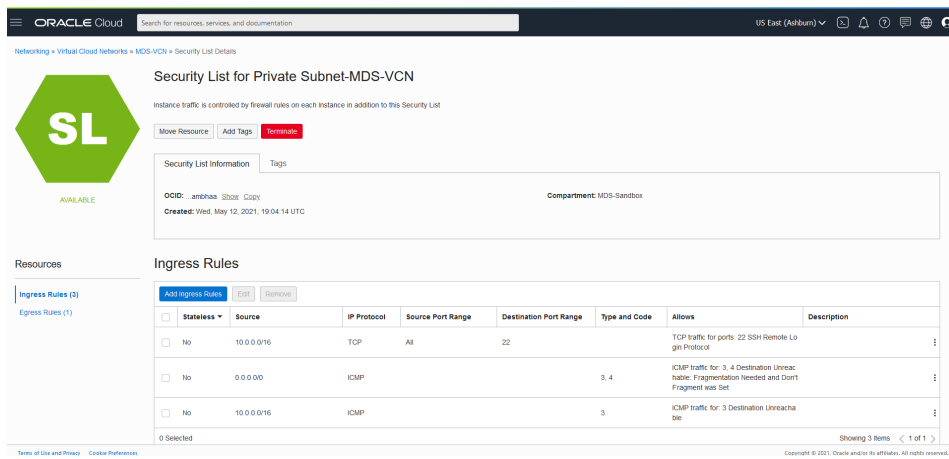
| Name                   | State     | IPv4 CIDR Block | Subnet Access      | Created                         |
|------------------------|-----------|-----------------|--------------------|---------------------------------|
| Private Subnet-MDS-VCN | Available | 10.0.1.0/24     | Private (Regional) | Wed, May 12, 2021, 19:04:15 UTC |
| Public Subnet-MDS-VCN  | Available | 10.0.0.0/24     | Public (Regional)  | Wed, May 12, 2021, 19:04:15 UTC |

## Task 2: Configure security list to allow MySQL incoming connections

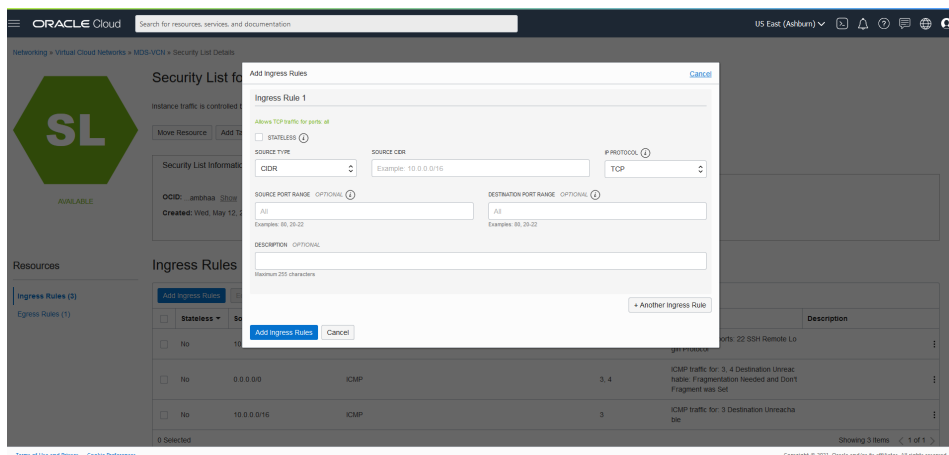
1. On myvcn page under 'Subnets in (root) Compartment', click 'Private Subnet-myvcn'



2. On Private Subnet-myvcn page under 'Security Lists', click '**Security List for Private Subnet-myvcn**'



3. On Security List for Private Subnet-myvcn page under 'Ingress Rules', click '**Add Ingress Rules**'



4. On Add Ingress Rules page under Ingress Rule 1

Add an Ingress Rule with Source CIDR

0.0.0.0/0

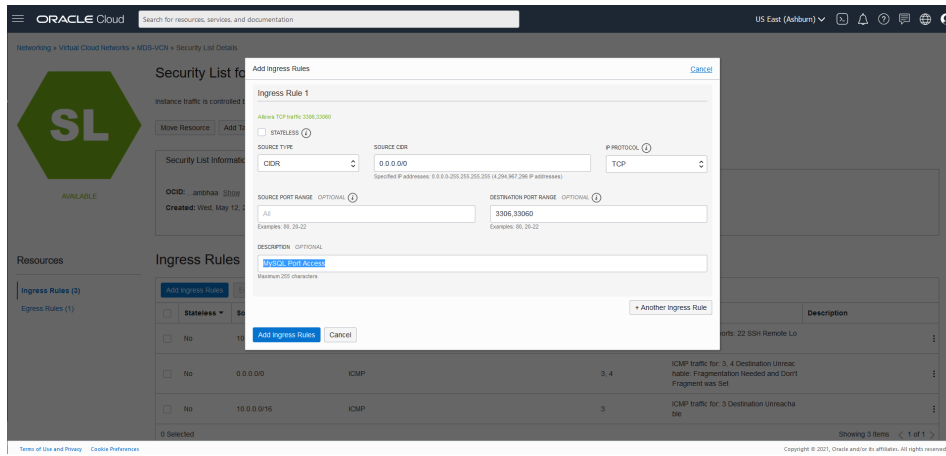
## Destination Port Range

3306,33060

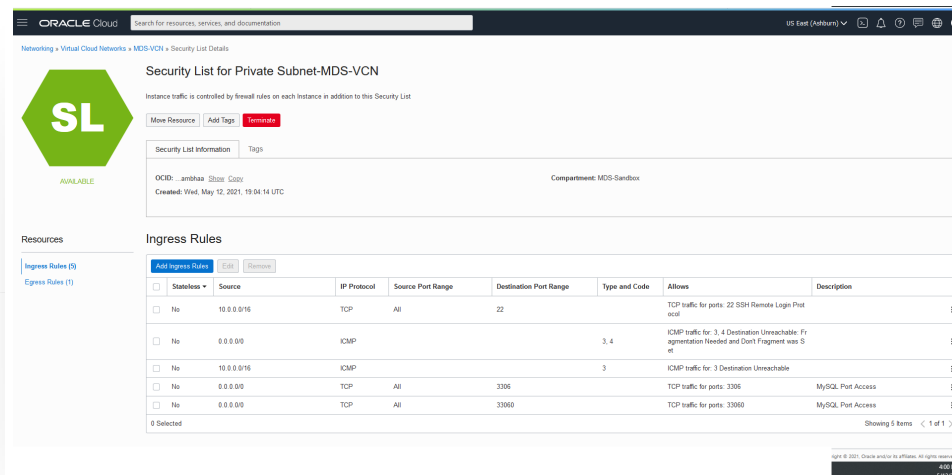
## Description

MySQL Port Access

## Click 'Add Ingress Rule'



5. On Security List for Private Subnet-mycn page, the new Ingress Rules will be shown under the Ingress Rules List



You may now proceed to the next lab

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, April 2022



# Create Linux Compute Instance

## Introduction

Oracle Cloud Infrastructure Compute lets you provision and manage compute hosts, known as instances . You can create instances as needed to meet your compute and application requirements. After you create an instance, you can access it securely from your computer or cloud shell.

## Create Linux Compute Instance

In this lab, you use Oracle Cloud Infrastructure to create an Oracle Linux instance.

*Estimated Time:* 10 minutes

## Objectives

In this lab, you will be guided through the following tasks:

- Create SSH Key on OCI Cloud
- Create Compute Instance

## Prerequisites

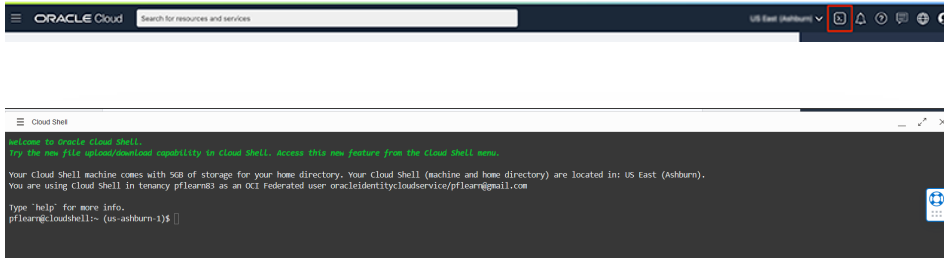
- An Oracle Free Tier or Paid Cloud Account
- A web browser
- Should have completed Lab 1

## Task 1: Create SSH Key on OCI Cloud Shell

The Cloud Shell machine is a small virtual machine running a Bash shell which you access through the Oracle Cloud Console (Homepage). You will start the Cloud Shell and generate a

SSH Key to use for the Bastion session.

1. To start the Oracle Cloud shell, go to your Cloud console and click the cloud shell icon at the top right of the page. This will open the Cloud Shell in the browser, the first time it takes some time to generate it.



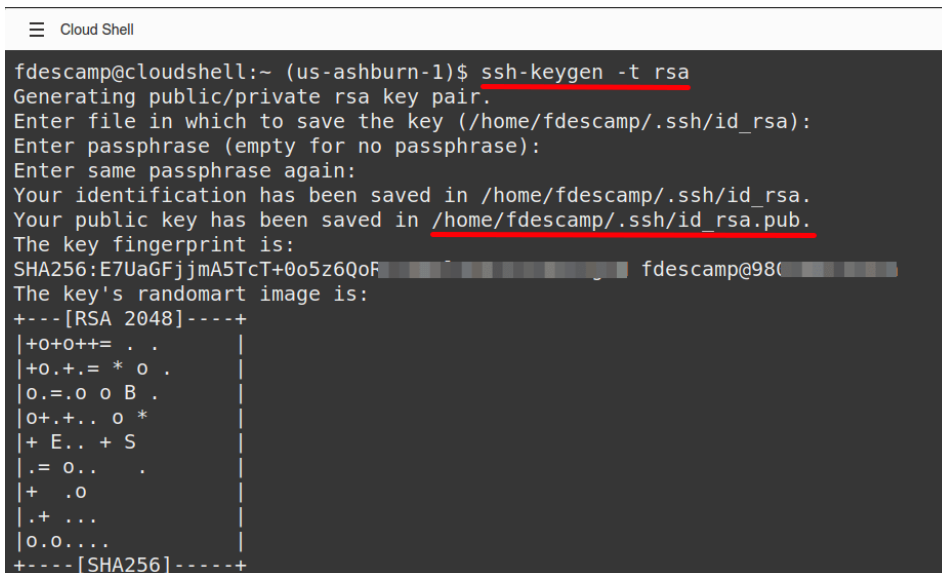
**Note:** You can use the icons in the upper right corner of the Cloud Shell window to minimize, maximize, restart, and close your Cloud Shell session.

2. Once the cloud shell has started, create the SSH Key using the following command:

```
ssh-keygen -t rsa
```

Press enter for each question.

Here is what it should look like.



3. The public and private SSH keys are stored in ~/.ssh/id\_rsa.pub.
4. Examine the two files that you just created.

```
cd .ssh
```

```
ls
```

```
pfilearn@cloudshell:~ (us-ashburn-1)$ cd .ssh
pfilearn@cloudshell:~/.ssh (us-ashburn-1)$ ls
id_rsa  id_rsa.pub
pfilearn@cloudshell:~/.ssh (us-ashburn-1)$
```

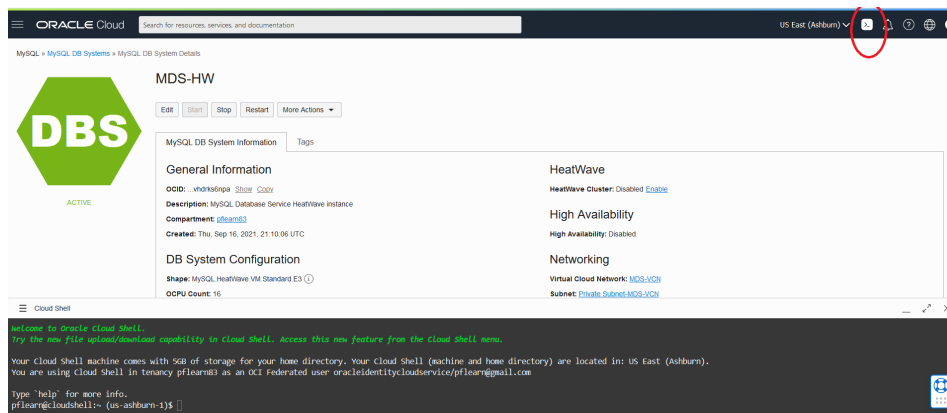
**Note:** in the output there are two files, a *private key*: id\_rsa and a *public key*: id\_rsa.pub. Keep the private key safe and don't share its content with anyone. The public key will be needed for various activities and can be uploaded to certain systems as well as copied and pasted to facilitate secure communications in the cloud.

## Task 2: Create Compute instance

You will need a compute Instance to connect to your brand new MySQL database.

1. Before creating the Compute instance open a notepad
2. Do the followings steps to copy the public SSH key to the notepad

Open the Cloud shell



Enter the following command

```
cat ~/.ssh/id_rsa.pub
```



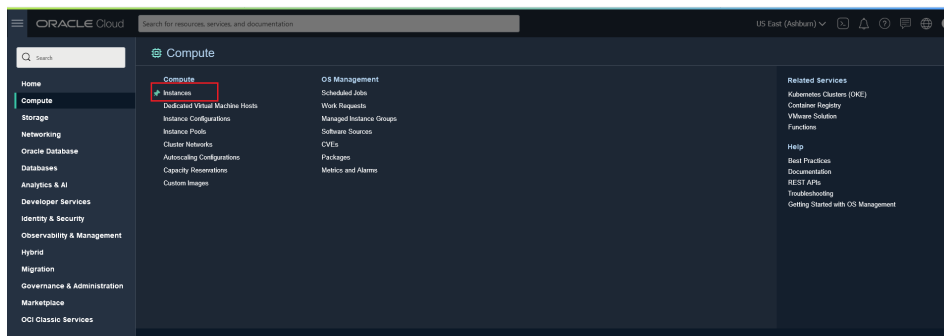
3. Copy the id\_rsa.pub content the notepad

Your notepad should look like this

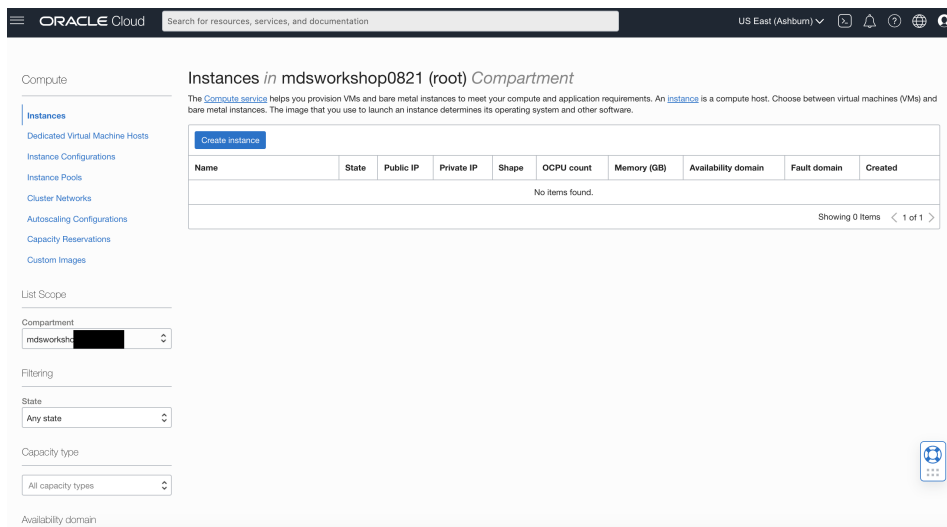
### id\_rsa.pub

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDDbWp1v2CYDDZyQ1F+V3Fu5zyZY9augGBLbmT6+Mbt0bn+WgwgA+Gotn
+uVUGn3Z5AiXRIeiih5ILdS8/6Ya46vvuD6C0gREngED2nIF737G8/zsM7P6hVMnRCQyvvHzW9+yRcGN/XW
+Fx0JgfmX0VV1wB/K6s81FD6TRohIhQgnUSxZr1ZhrfQAkjvsZfvU6ZrIGT70fy0+1gav0wu6bZYy1vx94E5y6LigmPWKiNB5b3rpCA6x
5a6uPa1FNcdTUFbBvY/XoBRQ3amPA8TTG8ZsYa3wZjQQXEu6bsu7ud1QWZ3n9xNOXvObDa4ay7DZ8r+1ewQC/+351crwWxVj9
pf1earn6264b69ff0ec
```

4. To launch a Linux Compute instance, go to Navigation Menu Compute Instances



5. On Instances in **(root)** Compartment, click **Create Instance**



## 6. On Create Compute Instance

Enter Name

myclient

## 7. Make sure **(root)** compartment is selected

## 8. On Placement, keep the selected Availability Domain

## 9. On Image and Shape click the **Edit** link

- On Image: Keep the selected Image, Oracle Linux 8

Create an instance to deploy and run applications, or save as a reusable Terraform stack for creating an instance with Resource Manager.

Name  
MDS-Client

Create in compartment  
Test\_1  
priscilla@vaio40 (root)/Test\_1

**Placement** [Edit](#)

Availability domain: AD-3 **Always Free-eligible** Capacity type: On-demand capacity  
Fault domain: Let Oracle choose the best fault domain

**Image and shape** [Edit](#)

Image: Oracle Linux 8 Shape: VM.Standard.E2.1.Micro **Always Free-eligible**  
Image build: 2022.01.24-0 OCPU count: 1  
Memory (GB): 1  
Network bandwidth (Gbps): 0.48

**Networking** [Edit](#)

Virtual cloud network: MDS-VCN Use network security groups to control traffic: No  
Subnet: Public Subnet-MDS-VCN Assign a public IPv4 address: Yes  
Launch options: - DNS record: Yes

10.

- On Shape - Click the **change shape** button

## ◦ Select Instance Shape: VM.Standard.E2.2

### Create compute instance

Create an instance to deploy and run applications, or save as a reusable Terraform stack for reuse.

Name: MDS-Client

Create in compartment: Test\_1

placementgroupid: pgrp/Test\_1

#### Placement

Availability domain: AD-3 [View availability domains](#)

Fault domain: Let Oracle choose the best fault domain

#### Image and shape

A [shape](#) is a template that determines the number of CPUs, amount of memory, and other resources of the shape.

Image: ORACLE Linux 8  
Image build: 2022.01.24-0

Shape: AMD VM.Standard.E2.1.Micro [View new shapes](#)  
Virtual machine, 1 core OCPU, 1 GB memory, 0.48 Gbps network bandwidth

[Show advanced options](#)

#### Networking

Virtual cloud network: MDS-VCN

Subnet: Public Subnet-MDS-VCN

Launch options: -

[Create](#) [Save as stack](#) [Cancel](#)

### Browse all shapes

A [shape](#) is a template that determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance.

Don't see the shape you want? To access all shapes, [upgrade](#). You'll pay only for what you use, no minimum terms and no prepayments.

[Upgrade](#)

#### Instance type

Virtual machine [View new shapes](#) ✓  
A virtual machine is an independent computing environment that runs on top of physical bare metal hardware.

Bare metal machine  
A bare metal compute instance gives you dedicated physical server access for highest performance and strong isolation.

#### Shape series

AMD [View new shapes](#)  
Flexible OCPU count. AMD processors.

Intel [View new shapes](#)  
Flexible OCPU count. Intel processors.

Ampere [View new shapes](#)  
Arm-based processor.

Specialty and previous generation  
Earlier generation AMD and Intel standard shapes. Always Free, Dense IO, GPU, and HPC shapes.

Image: Oracle Linux 8

| Shape name  | OCPU | Memory (GB) | Network bandwidth (Gbps) | Max. total VNICs |
|---|------|-------------|--------------------------|------------------|
| <input type="checkbox"/> VM.Standard.E2.1.Micro <a href="#">View new shapes</a> | 1    | 1           | 0.48                     | 1                |
| Local disk: Block storage only<br>Processor: 2.0 GHz AMD EPYC™ 7551 (Naples)    |      |             |                          |                  |
| <input type="checkbox"/> VM.Standard.E2.1                                       | 1    | 8           | 0.7                      | 2                |
| <input checked="" type="checkbox"/> VM.Standard.E2.2                            | 2    | 16          | 1.4                      | 2                |
| Local disk: Block storage only<br>Processor: 2.0 GHz AMD EPYC™ 7551 (Naples)    |      |             |                          |                  |
| <input type="checkbox"/> VM.Standard.E2.4                                       | 4    | 32          | 2.8                      | 4                |
| <input type="checkbox"/> VM.Standard.E2.8                                       | 8    | 64          | 5.6                      | 8                |
| <input type="checkbox"/> VM.Standard1.1   | 1    | 7           | 0.6                      | 2                |
| <input type="checkbox"/> VM.Standard1.2   | 2    | 14          | 1.2                      | 2                |
| <input type="checkbox"/> VM.Standard1.4   | 4    | 28          | 1.2                      | 4                |

[Select shape](#) [Cancel](#)

11.

12. On Networking, make sure '**myvcn**' is selected

'Assign a public IP address' should be set to Yes

### Networking

Virtual cloud network: MDS-VCN

Subnet: Public Subnet-MDS-VCN

Launch Options: -

Use network security groups to control traffic: No

Assign a public IPv4 address: Yes

DNS record: Yes

[Edit](#)

13. On Add SSH keys, paste the public key from the notepad.

### Create compute instance

Public IP address

☒ Assign a public IPv4 address ☐ Do not assign a public IPv4 address

[Show advanced options](#)

#### Add SSH keys

Generate an [SSH key pair](#) to connect to the instance using a Secure Shell (SSH) connection, or upload a public key that you already have.

☐ Generate a key pair for me ☐ Upload public key files (.pub) ☒ Paste public keys ☐ No SSH keys

SSH keys

2wO13b4OVNasnriueUrk/gMnjGT1fzVMxvU4W1zhfbzqJUDuepE+3Ky1GTL9024MBY3+9BR3OBhHsyUD+uhfzmeVn3nP2tPHPGIHUJIT8p7qJF57 perside\_fo@1073eeaa1d41

[Example: ssh-rsa AAAAB3Qza... NVApPb ssh-key-2021-01-27 \[See all supported key types\]\(#\)](#)

[+ Another key](#)

#### Boot volume

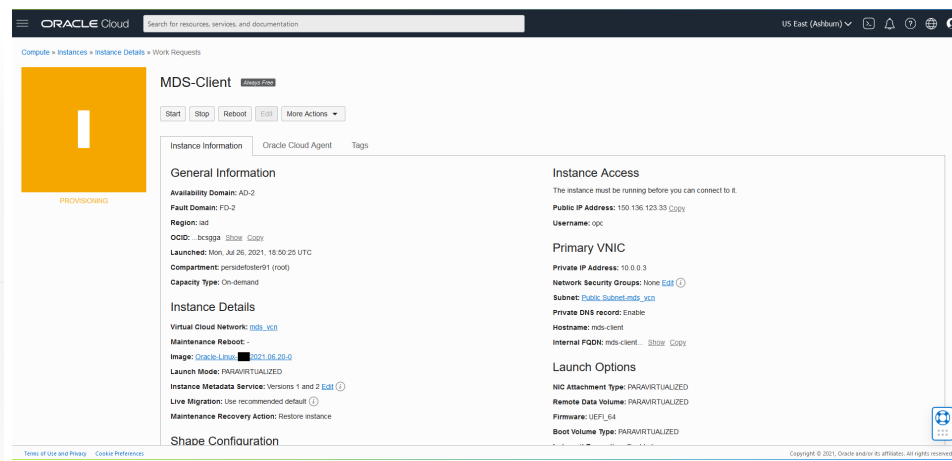
A [boot volume](#) is a detachable device that contains the image used to boot the compute instance.

[Create](#) [Save as stack](#) [Cancel](#)

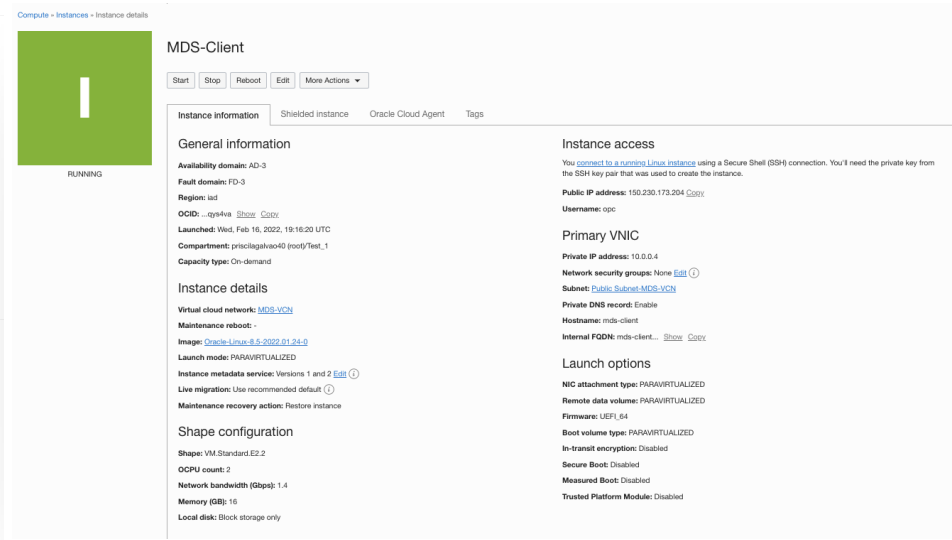
Cloud Shell

14. Click '**Create**' to finish creating your Compute Instance.

15. The New Virtual Machine will be ready to use after a few minutes. The state will be shown as 'Provisioning' during the creation



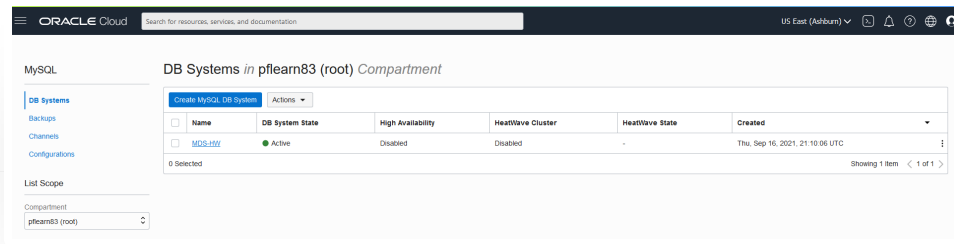
16. The state 'Running' indicates that the Virtual Machine is ready to use.



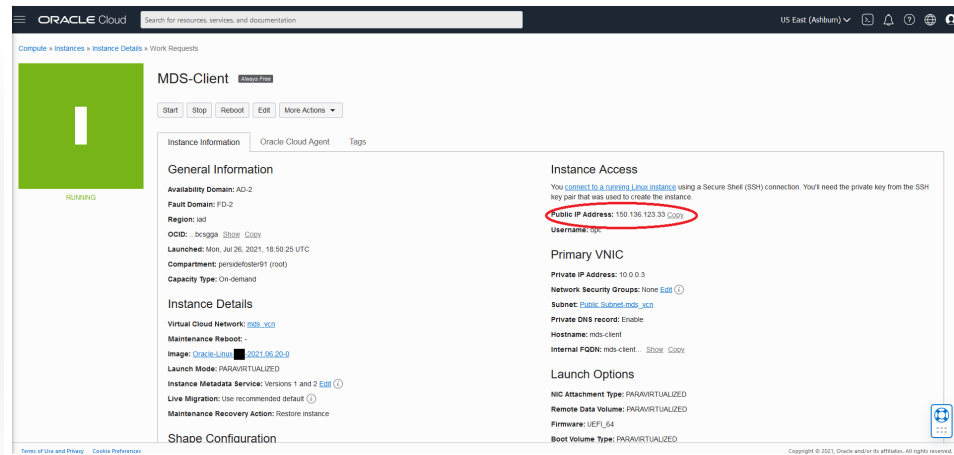
## Task 3: Connect to Compute Instance with SSH Key

To connect to **myclient** you will need to properly setup your SSH command. Do the following steps:

1. Copy the public IP address of the active Compute Instance to a notepad
  - a. Go to Navigation Menu Compute Instances



b. Click the mycClient Compute Instance link



c. Copy mycClient plus the Public IP Address to the notepad

2. Indicate the location of the private key you created earlier with **mycClient**.

Enter the username **opc** and the Public **IP Address**.

Note: The **mycClient** instance shows the Public IP Address as mentioned on TASK 5: #11

(Your SSH login command should look like this:

**ssh -i ~/.ssh/id\_rsa opc@132.145.170...**)

```
ssh -i ~/.ssh/id_rsa
opc@<your_compute_instance_ip>
```



```
Cloud Shell
perside_fo@cloudshell:~ (uk-london-1)$ ssh opc@150.230.116
The authenticity of host '150.230.116 (150.230.116)' can't be established.
ECDSA key fingerprint is
ECDSA key fingerprint is
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '150.230.116' (ECDSA) to the list of known hosts.
Last login: Sat Sep 25 21:10:08 2021 from 129.213.201
[opc@mds-client ~]$
```

**\*\* You are ready to install MySQL on the Compute Instance\*\***

**You may now proceed to the next lab**

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, April 2022

# SETUP

## Environment Setup

Objective: Connect Personal Computer to the Oracle Network and the Oracle Cloud Infrastructure (OCI)

In this lab you will Download lab materials, plus connect your Personal Computer to the Oracle Network and the Oracle Cloud Infrastructure (OCI)

Estimated Lab Time: -- 10 minutes

### Objectives

In this lab, you will:

- Download lab materials
- Setup SSH client
- Record Server information

### Prerequisites

*In compliance with Oracle security policies, I acknowledge I will not load actual confidential customer data or Personally Identifiable Information (PII) into my demo environment*

This lab assumes you have:

- An Oracle account
- All previous labs successfully completed

## Task 1: Download Lab Material and SSH client

1. lectures pdf
2. lab guide
3. SSH keys to connect labs (it's the same key in two different formats). These keys should have been created when you were creating your Compute Instance.
  - id\_rsa in native openssl format. Use it with Workbench
  - id\_rsa.ppk in putty format for windows. Use it only with putty
4. If you have not yet installed an SSH client on your laptop, please install one e.g. (windows) <https://www.putty.org/>

## Task 2: Record Lab Server info on Notepad

### student###-Server:

- Hostname:
- Hostname FQDN:
- Public IP: (e.g. 130.61.56.195)
- Private IP: (e.g. 10.0.11.18)

### Example:

The screenshot displays the Oracle Cloud console interface. At the top, there's a navigation bar with the Oracle Cloud logo, a search bar, and the region 'US East (Ashburn)'. Below this, the breadcrumb 'Compute > Instances > Instance details' is visible. The main content area shows the details for the instance 'instance-20220404-1110-Test8'. On the left, there's a green status icon with a white 'I' and the word 'RUNNING' below it. To the right of the icon are buttons for 'Start', 'Stop', 'Reboot', 'Edit', and 'More Actions'. The instance details are organized into several sections: 'Instance information' (with tabs for 'Shielded instance', 'Oracle Cloud Agent', and 'Tags'), 'General information' (listing availability domain, fault domain, region, OCID, launch time, compartment, and capacity type), 'Instance details' (listing virtual cloud network, maintenance reboot, image, and launch mode), 'Instance access' (providing instructions for SSH connection and listing public IP, username, and private IP), 'Primary VNIC' (listing private IP, network security groups, subnet, and private DNS record), and 'Launch options'.

**Instance information**

instance-20220404-1110-Test8

Start Stop Reboot Edit More Actions

**General information**

Availability domain: AD-1  
Fault domain: FD-3  
Region: iad  
OCID: ...ls4ila Show Copy  
Launched: Mon, Apr 4, 2022, 18:11:18 UTC  
Compartment: mysqlsls (root/Sandbox/DaleDasker-Sandbox)  
Capacity type: On-demand

**Instance details**

Virtual cloud network: vcn-20220401-1345-workshop  
Maintenance reboot: -  
Image: Oracle-Linux-8.5-2022.02.25-0  
Launch mode: PARAVIRTUALIZED

**Instance access**

You connect to a running Linux instance using a Secure Shell (SSH) connection. You'll need the private key from the SSH key pair that was used to create the instance.

Public IP address: 129.80.237.221 Copy  
Username: opc

**Primary VNIC**

Private IP address: 10.0.0.95  
Network security groups: None Edit  
Subnet: subnet-20220401-1345  
Private DNS record: Enable  
Hostname: instance-20220404-1110-test8  
Internal FQDN: instance-20220404-1110-test8... Show Copy




**Launch options**

## Task 3: Review Misc Lab Information

### 1. Document standard


- When in the manual you read **shell>** the command must be executed in the Operating System shell.
- When in the manual you read **mysql>** the command must be executed in a client like MySQL, MySQL Shell, MySQL Workbench, etc. We recommend students to use MySQL Shell to practice with it.
- When in the manual you read MySQL **mysqlsh>** the command must be executed in MySQL Shell.

### 2. Lab standard

-  shell> the command must be executed in the Operating System shell
-  mysql> the command must be executed in a client like MySQL, MySQL Workbench
-  mysqlsh> the command must be executed in MySQL shell

3. The software used for the labs is located on a local /workshop folder within each server.

4. Tip: set the keep alive for SSH connection to 60 seconds, to keep session open during lectures

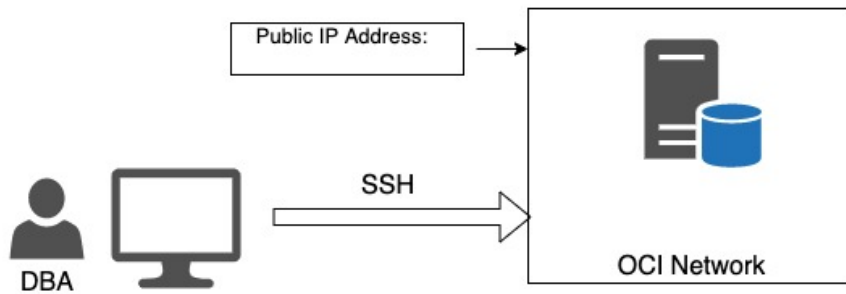
5. Linux **opc** user has limited privileges. To work with administrative privileges, use "sudo" like  shell> sudo su - root

## Task 4: Setup Lab Server and Connection

1. Server description **ServerA** will be used to run the full Workshop on. You will:

- Install MySQL Enterprise Edition 8.0.
- Install a MySQL Shell as a command line interface for MySQL Enterprise Edition.
- Install the Sample Employees Database

2. Sever Connections example:



3. Test the connection to your Linux machines from your laptop using these parameters

- a. SSH connection
- b. SSH key file named “id\_rsa” or “
- c. username “opc”
- d. no password
- e. Public IP address of your assigned Linux VM (serverA, serverB)

4. Examples of connections:

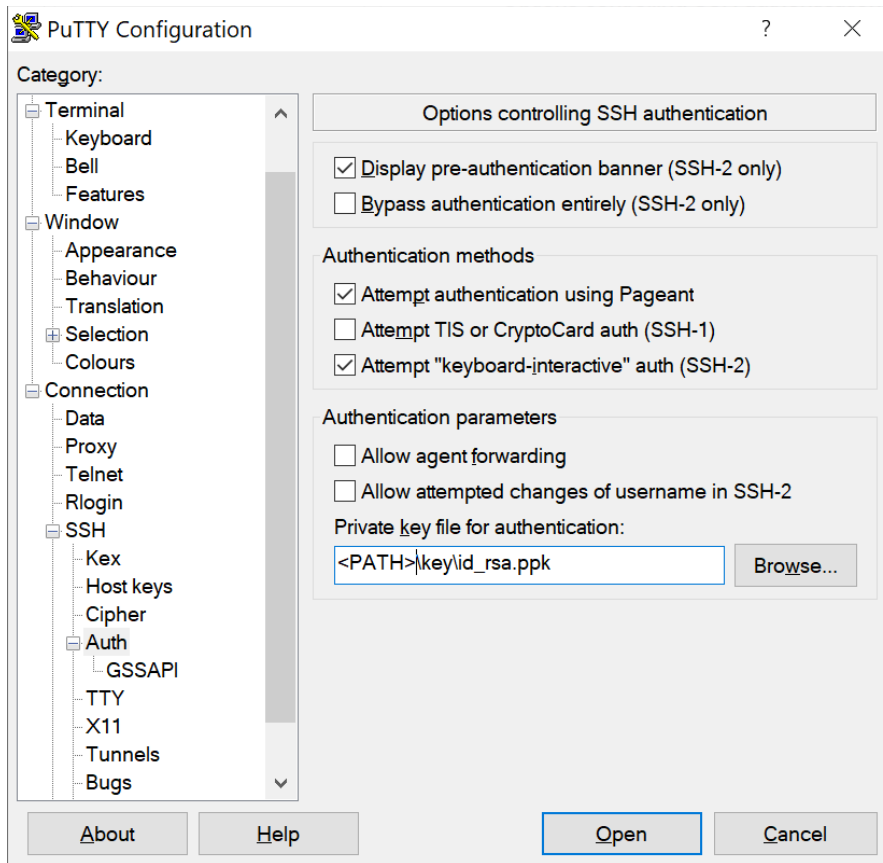
*Linux:* use “id\_rsa” key file

■ **shell>**

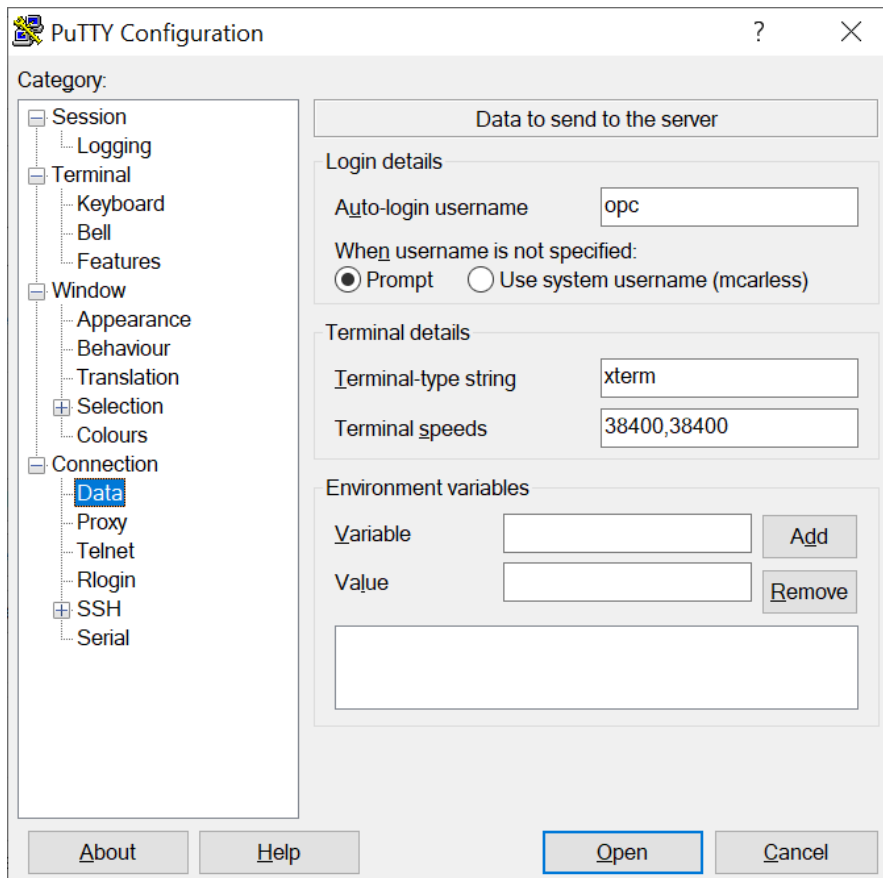
```
ssh -i id_rsa opc@public_ip
```

*Windows:* use “id\_rsa.ppk” key file

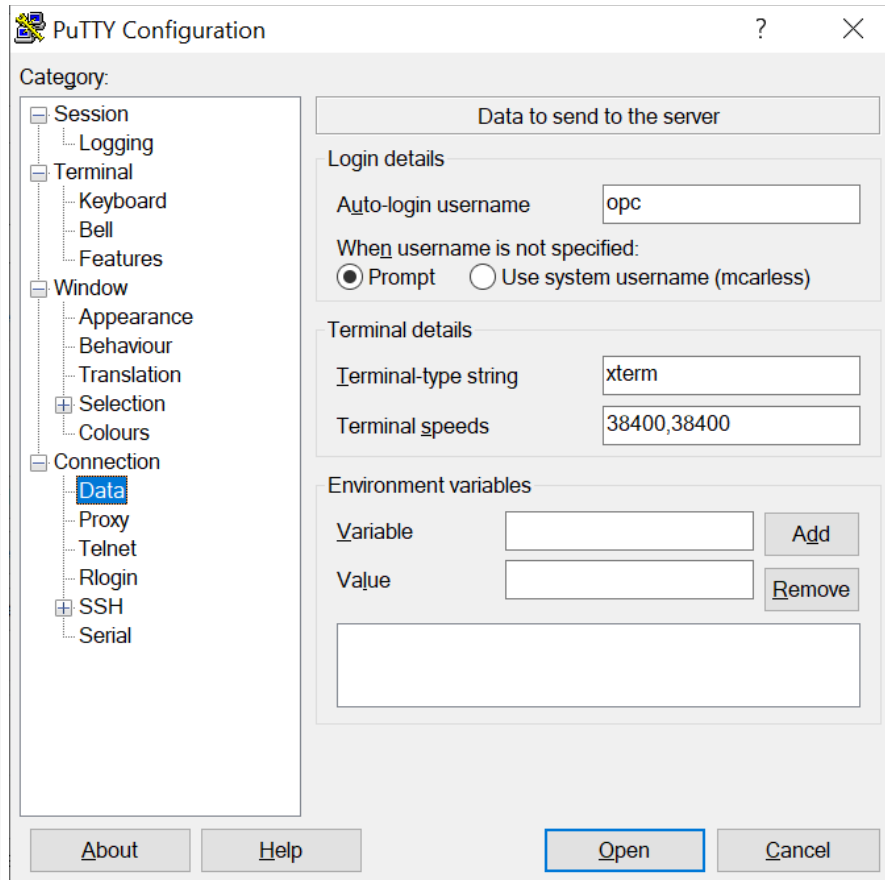
1. Open putty
2. Insert the public IP of your server and a mnemonic session name



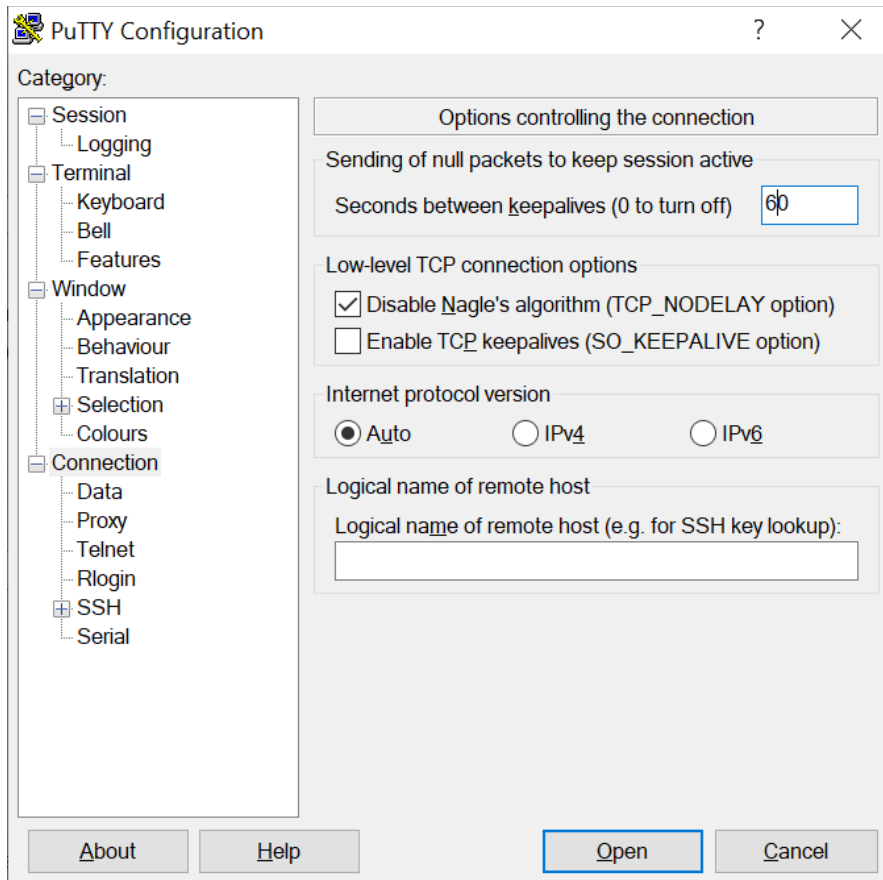
3. Choose "Connection SSH Auth" and provide the id\_rsa.ppk path



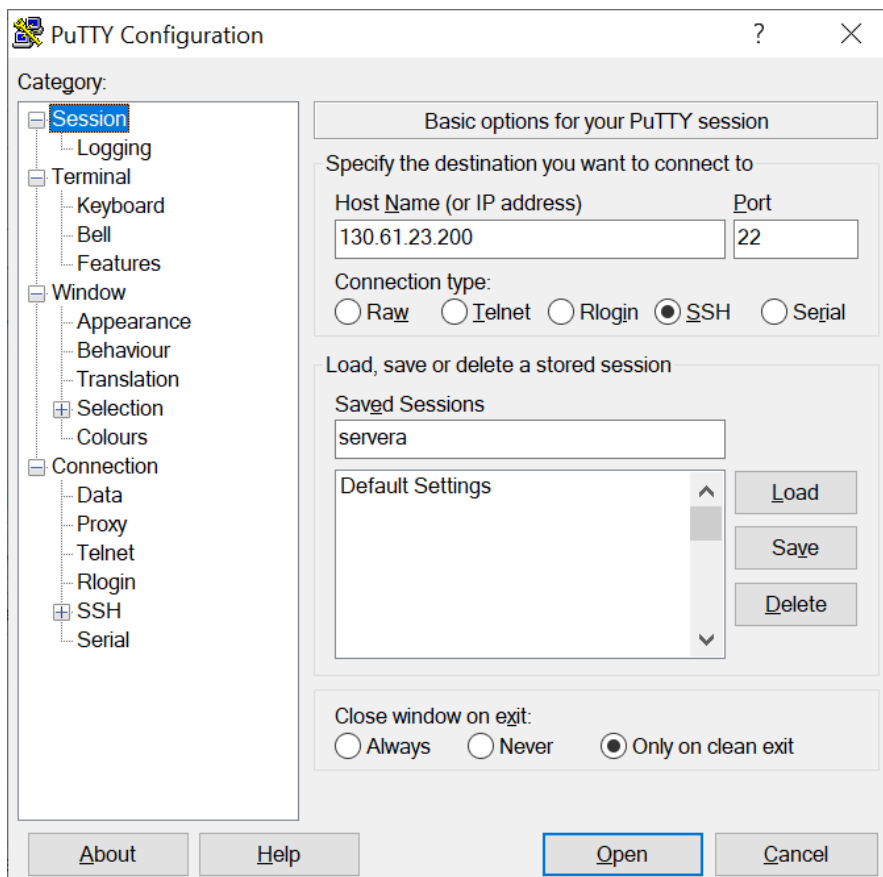
4. Select "Connection Data" and insert "opc" in "Auto-login username"



5. e) Choose Connection and insert "60" in "Seconds between keepalives"



6. Return to "Session" and click save





## Task 5: Setup workshop directory on Server

### 1. SSH to Server

■ shell>

```
ssh -i id_rsa opc@public_ip
```

### 2. Make /workshop Directory

■ shell>

```
sudo mkdir /workshop
```

### 3. FTP workshop files

■ shell>

```
cd /workshop
```

■ shell>

```
sudo curl -u vsftp_user:vsftp_Pa33w0rd -o  
workshop.tar.gz ftp://129.158.254.37  
/workshop.tar.gz
```

### 4. Extract workshop files

■ shell>

```
sudo tar xvf workshop.tar.gz
```

### 5. Setup for MySQL Client

■ shell>

```
sudo ln -s /usr/lib64/libtinfo.so.6.1  
/usr/lib64/libtinfo.so.5
```

## Learn More

- [Creating SSH Keys](#)
- [Compute SSH Connections](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, March 2022

### Web Clip

# INSTALL - MYSQL ENTERPRISE EDITION

## Introduction

Detailed Installation of MySQL Enterprise Edition 8.0 and MySQL Shell on Linux Objective:  
Tarball Installation of MySQL 8 Enterprise on Linux

Tarball Installation of MySQL Enterprise 8 on Linux

Estimated Time: 15 minutes

## Objectives




In this lab, you will:

- Install MySQL Enterprise Edition
- Start and test MySQL Enterprise Edition Install

- Install MySQL Shell and Connect to MySQL Enterprise

## Prerequisites

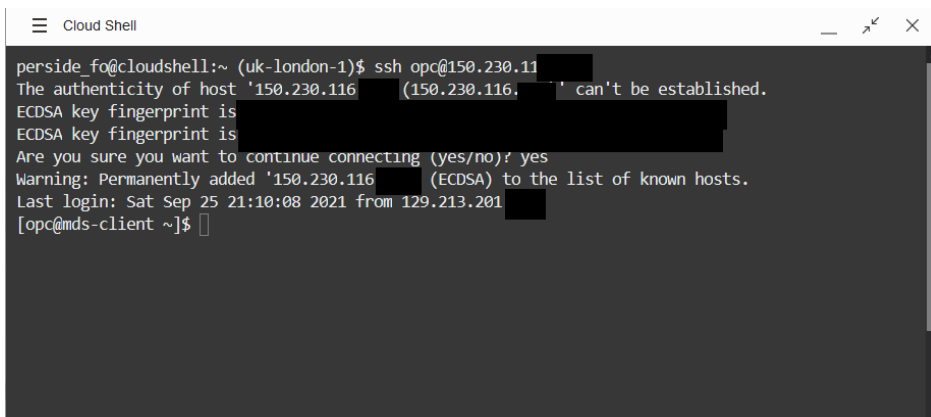
This lab assumes you have:

- An Oracle account
- All previous labs successfully completed
- Lab standard
  -  shell> the command must be executed in the Operating System shell
  -  mysql> the command must be executed in a client like MySQL, MySQL Workbench
  -  mysqlsh> the command must be executed in MySQL shell

## Task 1: Install MySQL Enterprise Edition

**Note:** If not already connected with SSH

- connect to **myclient** instance using Cloud Shell (**Example:** `ssh -i ~/.ssh/id_rsa opc@132.145.17....`) <copy>`ssh -i ~/.ssh/id_rsa`  
`opc@<your_compute_instance_ip></copy>`



```
perside_fo@cloudshell:~ (uk-london-1)$ ssh opc@150.230.116
The authenticity of host '150.230.116 (150.230.116)' can't be established.
ECDSA key fingerprint is
ECDSA key fingerprint is
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '150.230.116' (ECDSA) to the list of known hosts.
Last login: Sat Sep 25 21:10:08 2021 from 129.213.201
[opc@mds-client ~]$
```

1. Usually to run mysql the user “mysql” is used, but because it is already available we show here how create a new one.
2. Create a new user/group for your MySQL service (mysqluser/mysqlgrp) and add ‘mysqlgrp’ group to opc to help labs execution.

■ **shell>**

```
sudo groupadd mysqlgrp
```

■ **shell>**

```
sudo useradd -r -g mysqlgrp -s /bin/false  
mysqluser
```

■ **shell>**

```
sudo usermod -a -G mysqlgrp opc
```

3. Close and reopen shell session or use “newgrp” command as below

■ **shell>**

```
newgrp - mysqlgrp
```

4. Create new directory structure:

■ **shell>**

```
sudo mkdir /mysql/ /mysql/etc /mysql/data
```

■ **shell>**

```
sudo mkdir /mysql/log /mysql/temp /mysql  
/binlog
```

5. Extract the tarball in your /mysql folder

■ **shell>**

```
cd /mysql/
```

■ **shell>**

```
sudo tar xvf /workshop/mysql_8.0.28/mysql-  
commercial-8.0.28-linux-glibc2.12-  
x86_64.tar.xz
```

6. Create a symbolic link to mysql binary installation

■ **shell>**

```
sudo ln -s mysql-commercial-8.0.28-linux-  
glibc2.12-x86_64 mysql-latest
```

7. Create a new configuration file my.cnf inside /mysql/etc To help you we created one with some variables, please copy it

■ **shell>**

```
sudo cp /workshop/my.cnf.first /mysql  
/etc/my.cnf
```

8. For security reasons change ownership and permissions

■ **shell>**

```
sudo chown -R mysqluser:mysqlgrp /mysql
```

■ **shell>**

```
sudo chmod -R 755 /mysql
```

9. The following permission is for the Lab purpose so that opc account can make changes and copy files to overwrite the content

■ **shell>**

```
sudo chmod -R 770 /mysql/etc
```

## 10. initialize your database

■ **shell>**

```
sudo /mysql/mysql-latest/bin/mysqld  
--defaults-file=/mysql/etc/my.cnf --initialize  
--user=mysqluser
```

## Task 2: Start and test MySQL Enterprise Edition Install

### 1. Start your new mysql instance

■ **shell>**

```
sudo /mysql/mysql-latest/bin/mysqld  
--defaults-file=/mysql/etc/my.cnf  
--user=mysqluser &
```

### 2. Verify that process is running

■ **shell>**

```
ps -ef | grep mysqld
```

■ **shell>**

```
netstat -an | grep 3306
```

### 3. Another way is searching the message “ready for connections” in error log as one of the last

■ **shell>**

```
grep -i ready /mysql/log/err_log.log
```

#### 4. Install the MySQL Shell command line utility

 **shell>**

```
sudo yum -y install /workshop/shell/mysql-  
shell-commercial-8.0.28-1.1.el8.x86_64.rpm
```

#### 5. Retrieve root password for first login:

 **shell>**

```
grep -i 'temporary password' /mysql  
/log/err_log.log
```

#### 6. Login to the the mysql-enterprise installation and check the status (you will be asked to change password)

 **shell>**

```
mysqlsh --uri root@localhost:3306 --sql -p
```

#### 7. Create New Password for MySQL Root

 **mysqlsh>**

```
ALTER USER 'root'@'localhost' IDENTIFIED BY  
'Welcome1!';
```

 **mysqlsh>**

```
status
```

#### 8. Shutdown the service

■ mysqlsh>

```
quit
```

9. Create a new administrative user called 'admin' with remote access and full privileges

■ shell>

```
mysqlsh --sql --uri root@127.0.0.1:3306 -p
```

■ mysqlsh>

```
CREATE USER 'admin'@'%' IDENTIFIED BY  
'Welcome1!';
```

■ mysqlsh>

```
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%'  
WITH GRANT OPTION;
```

10. Add the mysql bin folder to the bash profile

■ mysqlsh>

```
quit
```

■ shell>

```
nano /home/opc/.bash_profile
```

11. After the value **# User specific environment and startup programs**. Add the following line:



```
PATH=$PATH:/mysql/mysql-latest/bin:$HOME  
/.local/bin:$HOME/bin
```

12. Save the changes, log out and log in again from the ssh for the changes to take effect on the user profile.

## Learn More

- [MySQL Linux Installation](#)
- [MySQL Shell Installation](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, April 2022

### Web Clip

# INSTALL - VERIFY MYSQL ENTERPRISE EDITION

## Introduction

Goal: Verify the new MySQL Installation on Linux and import test databases

Objectives:

- understand better how MySQL connection works
- install test databases for labs (world and employees)
- have a look on useful statements

Estimated Time: -- minutes




## Objectives

In this lab, you will:

- Discuss MySQL Connection
- Connect to Port 3306
- Import Sample Databases
- Learn Useful SQL Statements

## Prerequisites

This lab assumes you have:

- An Oracle account
- All previous labs successfully completed
- Lab standard
  -  shell> the command must be executed in the Operating System shell
  -  mysql> the command must be executed in a client like MySQL, MySQL Workbench
  -  mysqlsh> the command must be executed in MySQL shell

## Task 1: Discuss MySQL Connection

Please note that now you have an instance on the server on port 3306. To connect to MySQL, always use the IP address, otherwise you may connect to wrong instance. Here we practice connecting to the right one (port 3310 is intentionally wrong). To help you understand “why” these check lines (not all are always available...)

- Current user:
- Connection:
- UNIX socket:
- TCP port:

## Task 2: Connect to Port 3306

1.  shell>

```
mysql -u root -p --protocol=tcp
```

■ **mysql>**

```
status
```

■ **mysql>**

```
exit
```

2. Check a different port: ■ **mysql>**

```
mysql -uroot -p -h localhost -P3310  
--protocol=tcp
```

### Task 3: Import Sample Databases

1. Import the employees demo database that is in /workshop/databases folder.

■ **shell>**

```
cd /workshop/database/employees/
```

■ **shell>**

```
mysql -uroot -p -P3306 -h 127.0.0.1 <  
./employees.sql
```

### Task 4: Learn Useful SQL Statements

1. ■ **shell>**

```
mysql -uroot -p -h 127.0.0.1 -P 3306
```

2.  **mysql>**

```
SHOW VARIABLES LIKE "%version%";
```

3.  **mysql>**

```
SELECT table_name, engine FROM  
INFORMATION_SCHEMA.TABLES WHERE engine <>  
'InnoDB';
```

4.  **mysql>**

```
SELECT table_name, engine FROM  
INFORMATION_SCHEMA.TABLES WHERE engine =  
'InnoDB';
```

5.  **mysql>**

```
SELECT table_name, engine FROM  
INFORMATION_SCHEMA.TABLES where engine =  
'InnoDB' and table_schema not in  
('mysql','information_schema','sys');
```


6.  **mysql>**

```
SELECT ENGINE, COUNT(*), SUM(DATA_LENGTH)/  
1024 / 1024 AS 'Data MB',  
SUM(INDEX_LENGTH)/1024 / 1024 AS 'Index MB'  
FROM information_schema.TABLES group by  
engine;
```

7.  **mysql>**

```
SELECT table_schema AS 'Schema', SUM(  
data_length ) / 1024 / 1024 AS 'Data MB', SUM(  
index_length ) / 1024 / 1024 AS 'Index MB',
```

```
SUM( data_length + index_length ) / 1024 /  
1024 AS 'Sum' FROM information_schema.tables  
GROUP BY table_schema ;
```

8. The “G” is like “;” with a different way to show results  **mysql>**

```
SHOW GLOBAL VARIABLES
```

 **mysql>**

```
SHOW GLOBAL STATUS
```

 **mysql>**

```
SHOW FULL PROCESSLIST;
```

 **mysql>**

```
SHOW ENGINE INNODB STATUS
```

```
exit
```

## Learn More

- [MySQL Tutorial](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, April 2022

# SECURITY - MYSQL USERS

## Introduction

Users management Objective: explore user creation and privileges on a Server

\*This lab walks you through creating some users which will be used to Audit.

Estimated Time: 10 minutes




## Objectives

In this lab, you will do the followings:

- Connect to mysql-enterprise
- Create appuser

## Prerequisites

This lab assumes you have:

- An Oracle account
- All previous labs successfully completed
- Lab standard
  -  shell> the command must be executed in the Operating System shell
  -  mysql> the command must be executed in a client like MySQL, MySQL Workbench
  -  mysqlsh> the command must be executed in MySQL shell

## Notes:

- Open a notepad file and your linux Private IP on student###-serverA
- serverA PRIVATE ip: (client\_ip)

## Task 1: Connect to mysql-enterprise on Server

1. Connect to your mysql-enterprise with administrative user

■ **shell>**

```
mysql -uroot -p -h 127.0.0.1 -P 3306
```

2. Create a new user and restrict the user to your “Server” IP

a. ■ **mysql>**

```
CREATE USER 'appuser1'@'127.0.0.1' IDENTIFIED  
BY 'Welcome1!';
```

b. ■ **mysql>**

```
GRANT ALL PRIVILEGES ON employees.* TO  
'appuser1'@'127.0.0.1';
```

c. ■ **mysql>**

```
SHOW GRANTS FOR 'appuser1'@'127.0.0.1';
```

## Task 2: Connect to a second mysql-enterprise on Server

1. Open a new SSH connection on Server and from there connect to mysql-enterprise with appuser1

a. connect to mysql-enterprise with appuser1

■ **shell>**

```
mysql -u appuser1 -p -h 127.0.0.1 -P 3306
```

b. Run a select on the tables e.g.

■ **mysql>**

```
USE employees;
```

 **mysql>**

```
SELECT * FROM employees;
```

2. Switch to the administrative connection revoke privilege on city to appuser

 **mysql>**

```
REVOKE SELECT ON employees.* FROM  
'appuser1'@'127.0.0.1';
```

 **mysql>**

```
SHOW GRANTS FOR 'appuser1'@'127.0.0.1';
```

3. Repeat the select on appuser connection for the user. There is a difference?

 **mysql>**

```
SELECT * FROM employees;
```

### Task 3: Use appuser1 connection

1. Close and reopen the appuser1 connection for the user, then repeat above commands. There is a difference?

 **mysql>**

```
exit
```

 **shell>**



```
mysql -u appuser1 -p -h 127.0.0.1 -P 3306
```

 **mysql>**

```
USE employees;
```

 **mysql>**

```
SELECT * FROM employees;
```

2. Switch to the administrative connection revoke 'USAGE' privilege using and administrative connection and verify (tip: this privilege can't be revoked...)

 **mysql>**

```
REVOKE USAGE ON *.* FROM  
'appuser1'@'127.0.0.1';
```

 **mysql>**

```
SHOW GRANTS FOR 'appuser1'@'127.0.0.1';
```

3. Using the administrative connection revoke all privileges using and administrative connection and verify

 **mysql>**

```
REVOKE ALL PRIVILEGES ON *.* FROM  
'appuser1'@'127.0.0.1';
```

 **mysql>**

```
SHOW GRANTS FOR 'appuser1'@'127.0.0.1';
```

4. Close and reopen appuser session, do you see schemas?

## Task 4: Restore user privileges

1. Using the administrative connection restore user privileges to reuse it in next labs

■ **mysql>**

```
GRANT ALL PRIVILEGES ON employees.* TO  
'appuser1'@'127.0.0.1';
```

## Task 5: Add additional users

1. Using the Administrative Connection, create a new user and restrict the user to your “Server” IP

a. ■ **mysql>**

```
CREATE USER 'appuser2'@'127.0.0.1' IDENTIFIED  
BY 'Welcome1!';
```

b. ■ **mysql>**

```
GRANT ALL PRIVILEGES ON employees.* TO  
'appuser2'@'127.0.0.1';
```

2. Using the Administrative Connection, create another new user and restrict the user to your “Server” IP

a. ■ **mysql>**

```
CREATE USER 'appuser3'@'127.0.0.1' IDENTIFIED  
BY 'Welcome1!';
```

b. ■ **mysql>**

```
GRANT ALL PRIVILEGES ON employees.* TO  
'appuser3'@'127.0.0.1';
```

## Learn More

- [CREATE USER](#)
- [MySQL Access Control Lists](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering

### Web Clip

# SECURITY - MYSQL ENTERPRISE AUDIT

## Introduction

MySQL Enterprise Audit Objective: Auditing in action...

Estimated Lab Time: 20 minutes

## Objectives




In this lab, you will:

- Setup Audit Log
- Use Audit

## Prerequisites

This lab assumes you have:

- An Oracle account
- All previous labs successfully completed
- Lab standard

-  **shell>** the command must be executed in the Operating System shell
-  **mysql>** the command must be executed in a client like MySQL, MySQL Workbench
-  **mysqlsh>** the command must be executed in MySQL shell

### Notes:


- Audit can be activated and configured without stopping the instance. In the lab we edit my.cnf to see how to do it in this way

## Task 1: Setup Audit Log

1. If already connected to MySQL then exit  **mysql>**

```
exit
```


2. Enable Audit Log on mysql-enterprise (remember: you can't install on mysql-gpl). Audit is an Enterprise plugin.

- a. Edit the my.cnf setting in /mysql/etc/my.cnf  **shell>**

```
sudo nano /mysql/etc/my.cnf
```

- b. Change the line "plugin-load=thread\_pool.so" to load the plugin  **shell>**

```
plugin-load=thread_pool.so;audit_log.so
```

- c. below the previous add these lines to make sure that the audit plugin can't be unloaded and that the file is automatically rotated at 20 MB  **shell>**

```
audit_log=FORCE_PLUS_PERMANENT
```

 **shell>**

```
audit_log_rotate_on_size=20971520
```

■ **shell>**

```
audit_log_format=JSON
```

d. Restart MySQL (you can configure audit without restart the server, but here we show how to set the configuration file) ■ **shell>**

```
mysqladmin -uroot -p -h 127.0.0.1 -P3306  
shutdown
```

■ **shell>**

```
sudo /mysql/mysql-latest/bin/mysqld  
--defaults-file=/mysql/etc/my.cnf $MYSQLD_OPTS  
&
```

e. Load Audit functions ■ **shell>**

```
mysql -uroot -p -h 127.0.0.1 -P 3306 <  
/workshop/audit_log_filter_linux_install.sql
```

3. Connect to your mysql-enterprise with administrative user

■ **shell>**

```
mysql -uroot -p -h 127.0.0.1 -P 3306
```

a. Using the Administrative Connection, create a Audit Filter for all activity and all users ■ **mysql>**


```
SELECT audit_log_filter_set_filter('log_all',  
'{ "filter": { "log": true } }');
```

 **mysql>**

```
SELECT audit_log_filter_set_user('%',  
'log_all');
```

b.  **mysql>**

```
exit
```

c. Monitor the output of the audit.log file:  **shell>**

```
tail -f /mysql/data/audit.log
```

## Task 2: Use Audit

1. Login to mysql-enterprise with the user “appuser1”, then submit some commands

a.  **shell>**

```
mysql -u appuser1 -p -h 127.0.0.1 -P 3306
```

b.  **mysql>**

```
USE employees;
```

c.  **mysql>**

```
SELECT * FROM employees limit 25;
```

d.  **mysql>**

```
SELECT emp_no,salary FROM employees.salaries  
WHERE salary > 90000;
```

2. Let's setup Audit to only log connections. Using the Administrative Connection, create a Audit Filter for all connections

a.  **mysql>**

```
SET @f = '{ "filter": { "class": { "name":  
"connection" } } }';
```

b.  **mysql>**

```
SELECT  
audit_log_filter_set_filter('log_conn_events',  
@f);
```

c.  **mysql>**

```
SELECT audit_log_filter_set_user('%',  
'log_conn_events');
```

3. Login to mysql-enterprise with the user “appuser1”, then submit some commands

a.  **shell>**

```
mysql -u appuser1 -p -h 127.0.0.1 -P 3306
```

b.  **mysql>**

```
USE employees;
```

c.  **mysql>**

```
SELECT * FROM employees limit 25;
```

d.  **mysql>**

```
SELECT emp_no,salary FROM employees.salaries
WHERE salary > 90000;
```

4. Let's setup Audit to only log unique users. Using the Administrative Connection, create a Audit Filter for appuser1

a. Remove previous filter:

■ **mysql>**

```
SELECT
audit_log_filter_remove_filter('log_conn_event
s ');
```

■ **mysql>**

```
SELECT audit_log_filter_flush();
```

b. ■ **mysql>**

```
SELECT audit_log_filter_set_filter('log_all',
'{ "filter": { "log": true } }');
```

c. ■ **mysql>**

```
SELECT
audit_log_filter_set_user('appuser1@127.0.0.1'
, 'log_all');
```

d. ■ **mysql>**

```
SELECT audit_log_filter_flush();
```

5. Login to mysql-enterprise with the user “appuser1”, then submit some commands



a.  **shell>**

```
mysql -u appuser1 -p -h127.0.0.1 -P 3306
```

b.  **mysql>**

```
USE employees;
```

c.  **mysql>**

```
SELECT * FROM employees limit 25;
```

d.  **mysql>**

```
SELECT emp_no,salary FROM employees.salaries  
WHERE salary > 90000;
```

6. Login to mysql-enterprise with the user “appuser2”, then submit some commands

a.  **shell>**

```
mysql -u appuser2 -p -h127.0.0.1 -P 3306
```

b.  **mysql>**

```
USE employees;
```

c.  **mysql>**

```
SELECT * FROM employees limit 25;
```

d.  **mysql>**

```
SELECT emp_no,salary FROM employees.salaries
```

```
WHERE salary > 90000;
```

7. Let's setup Audit to only log access to salaries tables. Using the Administrative Connection, create a Audit Filter for salaries

a. Remove previous filter:  **mysql>**

```
SELECT audit_log_filter_remove_filter('log_all  
' );
```

 **mysql>**

```
SELECT audit_log_filter_flush();
```

b.  **mysql>**

```
SET @f='  
{  
  "filter": {  
    "class":  
      {  
        "name": "table_access",  
        "event":  
          {  
            "name": [ "insert", "update",  
"delete" ],  
            "log": { "field": { "name":  
"table_name.str", "value": "salaries" }}  
          }  
        }  
      }  
    }  
  }'  
;
```

c.  **mysql>**

```
SELECT  
audit_log_filter_set_filter('salary_insert',
```

```
@f);
```

d.  **mysql>**

```
SELECT audit_log_filter_set_user('%',  
'salary_insert');
```

8. Login as 'appuser1' and run a query against the salaries table;

a.  **shell>**

```
mysql -u appuser1 -p -h127.0.0.1 -P 3306
```

b.  **mysql>**


```
USE employees;
```

c.  **mysql>**

```
SELECT * FROM employees limit 25;
```

d. Run updates on salaries table  **mysql>**

```
UPDATE employees.salaries SET salary = 74234  
WHERE emp_no = 10001;
```

9. Some Administrative commands for checking Audit filters and users. Log in using the Administrative Connection,  **shell>**

```
mysql -uroot -p -h 127.0.0.1 -P 3306
```

a. Check existing filters:  **mysql>**

```
SELECT * FROM mysql.audit_log_filter;
```

b. Check Users being Audited:  **mysql>**

```
SELECT * FROM mysql.audit_log_user;
```

c. Global Audit log disable  **mysql>**

```
SET GLOBAL audit_log_disable = true;
```

10. You can check the documentation about other Log filters & policies

## Learn More

- [Writing Audit Filters](#)
- [Audit Filter Definitions](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, March 2022

### Web Clip

# SECURITY - MYSQL ENTERPRISE TRANSPARENT DATA ENCRYPTION

## Introduction

3c) MySQL Enterprise Transparent Data Encryption Objective: Data Encryption in action...

This lab will walk you through encrypting InnoDB Tablespace files at rest

Estimated Lab Time: 20 minutes




## Objectives

In this lab, you will:

- Install and encrypt Data Files

## Prerequisites (Optional)

This lab assumes you have:

- An Oracle account
- All previous labs successfully completed
- Lab standard
  -  **shell>** the command must be executed in the Operating System shell
  -  **mysql>** the command must be executed in a client like MySQL, MySQL Workbench
  -  **mysqlsh>** the command must be executed in MySQL shell

## Notes:

- References
- <https://dev.mysql.com/doc/refman/8.0/en/innodb-data-encryption.html>

## Task 1: Install and setup TDE

1. Install MySQL Enterprise Transparent Data Encryption on mysql-enterprise using Administrative MySQL client connections

 **shell>**

```
mysql -u root -p -P3306 -h127.0.0.1
```

2. Check to see if any keyring plugin is installed and load if not:

a.  **mysql>**

```
SELECT PLUGIN_NAME, PLUGIN_STATUS FROM  
INFORMATION_SCHEMA.PLUGINS WHERE PLUGIN_NAME
```

```
LIKE 'keyring%';
```

b. Edit the my.cnf setting in /mysql/etc/my.cnf

■ **shell>**

```
sudo nano /mysql/etc/my.cnf
```

b. Add the following lines to load the plugin and set the encrypted key file

■ **shell>**

```
early-plugin-load=keyring_encrypted_file.so
```

```
keyring_encrypted_file_data=/mysql/data/mysql-  
keyring/keyring-encrypted
```

```
keyring_encrypted_file_password=V&rySec4eT
```

c. Restart MySQL

■ **shell>**

```
mysqladmin -uroot -p -h 127.0.0.1 -P3306  
shutdown
```

■ **shell>**

```
sudo /mysql/mysql-latest/bin/mysqld  
--defaults-file=/mysql/etc/my.cnf $MYSQLD_OPTS  
&
```

3. "Spy" on employees.employees table

a. ■ **shell>**

```
strings "/mysql/data/employees/employees.ibd"  
| head -n50
```

4. Now we enable Encryption on the employees.employees table:

a.  **shell>**

```
mysql -u root -p -P3306 -h127.0.0.1
```

b.  **mysql>**

```
USE employees;
```

c.  **mysql>**

```
ALTER TABLE employees ENCRYPTION = 'Y';
```

5. "Spy" on employees.employees table again:

a.  **shell>**

```
strings "/mysql/data/employees/employees.ibd"  
| head -n50
```

6. Administrative commands

a. Get details on encrypted key file:  **mysql>**

```
SHOW VARIABLES LIKE  
'keyring_encrypted_file_data'
```

b. Set default for all tables to be encrypted when creating them:  **mysql>**

```
SET GLOBAL default_table_encryption=ON;
```

c. Peek on the mysql System Tables:  **mysql>**

```
strings "/mysql/data/mysql.ibd" | head -n70
```

d. Encrypt the mysql System Tables:  **mysql>**

```
ALTER TABLESPACE mysql ENCRYPTION = 'Y';
```

e. Validate encryption of the mysql System Tables:  **mysql>**

```
strings "/mysql/data/mysql.ibd" | head -n70
```

f. Show all the encrypted tables:  **mysql>**

```
SELECT SPACE, NAME, SPACE_TYPE, ENCRYPTION  
FROM INFORMATION_SCHEMA.INNODB_TABLESPACES  
WHERE ENCRYPTION='Y'
```

## Learn More

- [Keyring Plugins](#)
- [InnoDB Data At Rest](#)

## Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Contributors** -
- **Last Updated By/Date** - <Dale Dasker, March 2022