

MySQL Enterprise Edition - Implementation Essentials Bootcamp | Introduction

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 4 of the MySQL Security Based Enterprise Features. We will go through how to setup and run Enterprise Audit, Enterprise Transparent Data Encryption, Enterprise FireWall and Enterprise Data Masking.

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.

Prerequisite

For working on this workshop you will need a clean computer to run on. We should have provided you with a link on setting up a free account on Oracle Cloud Infrastructure (OCI).

If you have not set it up, then please go through the following steps:

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 4 of the MySQL Security Based Enterprise Features. We will go through how to setup and run Enterprise Audit, Enterprise Transparent Data Encryption, Enterprise FireWall and Enterprise Data Masking.

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.

Prerequisite

For working on this workshop you will need a clean computer to run on. We should have provided you with a link on setting up a free account on Oracle Cloud Infrastructure (OCI).

If you have not set it up, then please go through the following steps:

- [OCI Free Tier Setup](#)

Objectives

In this workshop, you will learn how to work with

- MySQL Enterprise Edition
- MySQL Shell

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 1: Create Virtual Cloud Network and Related Components

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

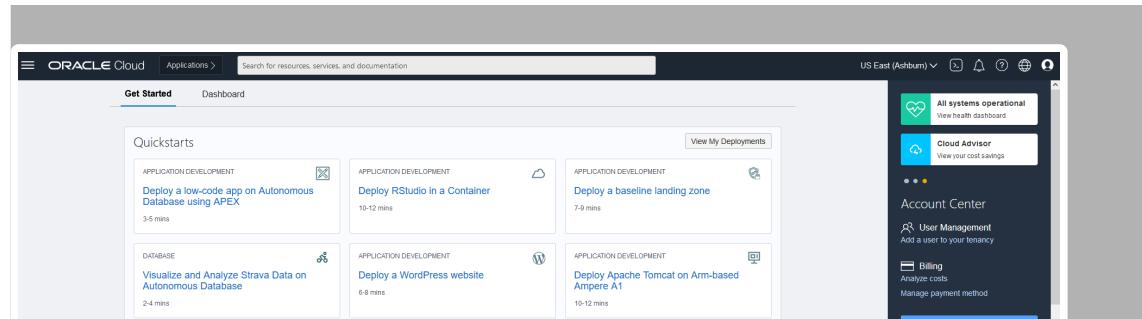
Objectives

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)



Launch Resources

- COMPUTE**
Create a VM Instance
2-4 mins
- AUTONOMOUS TRANSACTION PROCESSING**
Create an ATP database
3-5 mins
- AUTONOMOUS DATA WAREHOUSE**
Create an ADW database
3-5 mins
- NETWORKING**
Set up a network with a wizard
- RESOURCE MANAGER**
Create a stack
- OBJECT STORAGE**
Store data

View All Resources

Oracle Live: Running OCI workloads your way

Learn to run any application faster, with increased flexibility, improved security and optimized cost. Watch 10, 2022.

Register now

What's New

Financial Services for GPU conda environment is introduced Mar 25, 2022

Database Migration is now available in the Oracle Cloud

Copyright © 2022, Oracle and/or its affiliates. All rights reserved.

Task 1: Create Virtual Cloud Network

1. Click Navigation Menu

Select Networking

Select Virtual Cloud Networks

Search for resources, services, and documentation

US East (Ashburn) ▾

Networking

Overview

- Virtual Cloud Networks**
- Load Balancers
- Network Visualizer
- Inter-Region Latency

DNS Management

- Overview
- Zones
- Traffic Management
- Steering Policies
- Private Vnws
- HTTP Redirects
- TSL Keys

Customer Connectivity

- Site-to-Site VPN (IPsec)
- FastConnect
- Dynamic Routing Gateway
- Customer Premises Equipment

IP Management

- Reserved IPs
- BYOP
- Pools

Related Services

- Cloud Networks
- Cloud Guard
- Web Application Firewall
- VMware Solution

Help

- Networking Overview
- Virtual Cloud Networks
- VPN Connect
- FastConnect
- Traffic Steering
- DNS Management

Terms of Use and Privacy | Cookie Preferences

2. Click Start VCN Wizard

Networking

Virtual Cloud Networks

Name	State	IPv4 CIDR Block	IPv6 CIDR Block	Default Route Table	DNS Domain Name	Created

Create VCN Start VCN Wizard

Virtual Cloud Networks are virtual, private networks that you set up in Oracle data centers. It closely resembles a traditional network, with firewall rules and specific types of communication gateways that you can choose to use.

Name: persidefoster91

State: Active

IPv4 CIDR Block: 10.0.0.0/16

IPv6 CIDR Block: 2001:db8::/32

Default Route Table: Default

DNS Domain Name: persidefoster91.oci.com

Created: Jul 17, 2021 16:18:54 UTC

Tags:

- comartment: persidefoster91
- STATE: Any State

Filters

no log filters applied

Terms of Use and Privacy | Cookie Preferences

3. Select 'Create VCN with Internet Connectivity'

Click 'Start VCN Wizard'

Networking

Virtual Cloud Networks

Create VCN Start VCN Wizard

Start VCN Wizard

Step 1: Create VCN with Internet Connectivity

Create VCN with Internet Connectivity

Add Internet Connectivity and Site-to-Site VPN to a VCN

Help | Cancel

Virtual Cloud Networks are virtual, private networks that you set up in Oracle data centers. It closely resembles a traditional network, with firewall rules and specific types of communication gateways that you can choose to use.

Name: persidefoster91

Created: Jul 17, 2021 16:18:54 UTC

persidefoster91.oci.com



4. Create a VCN with Internet Connectivity

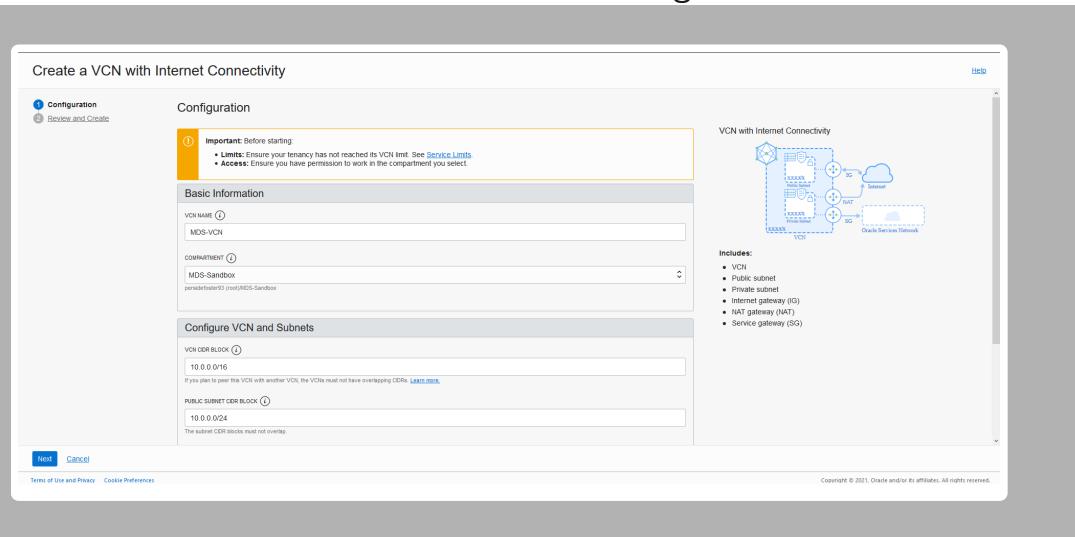
On Basic Information, complete the following fields:

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

paradevteam01 (root/MDS-Sandbox)

Configure VCN and Subnets

VCN CDR BLOCK

If you plan to peer this VCN with another VCN, the VCNs must not have overlapping CDRs. [Learn more](#).

PUBLIC SUBNET CDR BLOCK

The subnet CDR blocks must not overlap.

VCN with Internet Connectivity



Includes:

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

Showing 1 item < 1 of 1 >

Copyright © 2021, Oracle and/or its affiliates. All rights reserved.

10.0.0.0/16
If you plan to peer this VCN with another VCN, the VCNs must not have overlapping CIDR blocks. [Learn more.](#)

PUBLIC SUBNET CIDR BLOCK 10.0.0/24
The subnet CIDR block must not overlap.

[Next](#) [Cancel](#)

[Terms of Use and Privacy](#) [Cookie Preferences](#)

Copyright © 2021, Oracle and/or its affiliates. All rights reserved.

7. The Virtual Cloud Network creation is completing

ORACLE Cloud Search for resources, services, and documentation US East (Ashburn) Help

Create a VCN with Internet Connectivity

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)

[View Virtual Cloud Network](#)

<https://cloud.oracle.com/links/> [Cookie Preferences](#)

Copyright © 2021, Oracle and/or its affiliates. All rights reserved.

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

ORACLE Cloud Search for resources, services, and documentation US East (Ashburn) Help

Networking > Virtual Cloud Networks > Virtual Cloud Network Details

MDS-VCN

Move Resource Add Tags [Terminate](#)

VCN Information

Compartment: MDS-Sandbox
Created: Wed, May 12, 2021, 19:04:15 UTC
IPv4 CIDR Block: 10.0.0/16
IPv6 CIDR Block: No Value

OCID: ...dbr3a Show Copy
DNS Resolver: MDS-VCN
Default Route Table: Default Route Table for MDS-VCN
DNS Domain Name: mdsvcn.oraclevcn.com

Resources

Subnets in MDS-Sandbox Compartment

Name	State	IPv4 CIDR Block	Subnet Access	Created
Private_Subnet-MDS-VCN	Available	10.0.1/24	Private (Regional)	Wed, May 12, 2021, 19:04:15 UTC
Public_Subnet-MDS-VCN	Available	10.0.0/24	Public (Regional)	Wed, May 12, 2021, 19:04:15 UTC

Show 2 items < 1 of 1 >

[Create Subnet](#)

Subnets (2)
CIDR Blocks (1)
Route Tables (2)
Internet Gateways (1)
Dynamic Routing Gateways (0)
Network Security Groups (0)
Security Lists (2)
DHCP Options (1)

[Local Databases \(Database.jsf\)](#)

[Terms of Use and Privacy](#) [Cookie Preferences](#)

Copyright © 2021, Oracle and/or its affiliates. All rights reserved.

Task 2: Configure security list to allow MySQL incoming connections

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

ORACLE Cloud Cloud Classic > Search resources, services, and documentation US East (Ashburn) Help

Networking > Virtual Cloud Networks > myvcn > Subnet Details

Public Subnet-myvcn

Edit Move Resource Add Tags [Terminate](#)

Subnet Information

OCID: ...ivakna Show Copy
IPv4 CIDR Block: 10.0.0/24
Virtual Router Mac Address: 00:01:17:E6:B5:8E
Subnet Type: Regional

Compartment: DaleDasker-Sandbox
DNS Domain Name: sub04112049310... Show Copy
Subnet Access: Public Subnet
DHCP Options: Default DHCP Options for myvcn
Route Table: Default Route Table for myvcn

[Edit](#) [Move Resource](#) [Add Tags](#) [Terminate](#)

[Terms of Use and Privacy](#) [Cookie Preferences](#)

Resources

Security Lists

Add Security List			
Name	State	Compartment	Created
Default Security List for myvcn	Available	DaleDasker-Sandbox	Mon, Apr 11, 2022, 20:49:42 UTC

Showing 1 Item < 1 of 1 >

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

ORACLE Cloud Cloud Classic > Search resources, services, and documentation US East (Ashburn) ▾ 🔍 🌐 🌐

Networking > Virtual Cloud Networks > myvcn > Security List Details

Default Security List for myvcn

Instance traffic is controlled by firewall rules on each Instance in addition to this Security List

Move Resource Add Tags Terminate

Security List Information Tags

OCID: ...wxc2da Show Copy Compartment: DaleDasker-Sandbox

Created: Mon, Apr 11, 2022, 20:49:42 UTC

AVAILABLE

Resources Ingress Rules

Add Ingress Rules Edit Remove

Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol	
No	0.0.0.0/0	ICMP		3, 4		ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set	
No	10.0.0.0/16	ICMP		3		ICMP traffic for: 3 Destination Unreachable	

0 Selected Showing 3 Items < 1 of 1 >

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

ORACLE Cloud Search for resources, services, and documentation US East (Ashburn) ▾ 🔍 🌐 🌐

Networking > Virtual Cloud Networks > MDS-VCN > Security List Details

Security List for Public Subnet-myvcn

Instance traffic is controlled by firewall rules on each Instance in addition to this Security List

Move Resource Add Tags Terminate

Security List Information

OCID: ...amhhas Show Created: Wed, May 12, 2022, 19:45:22 UTC

AVAILABLE

Resources Ingress Rules

Add Ingress Rules

Ingress Rule 1

Allows TCP traffic for ports: all

STATELESS

SOURCE CIDR: 10.0.0.0/16

IP PROTOCOL: TCP

SOURCE PORT RANGE: All

DESTINATION PORT RANGE: All

DESCRIPTION: Maximum 255 characters

+ Another Ingress Rule

Ports: 22 SSH Remote Login Protocol

ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set

ICMP traffic for: 3 Destination Unreachable

Showing 3 Items < 1 of 1 >

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'

The screenshot shows the Oracle Cloud interface for managing security lists. A modal window titled 'Add Ingress Rules' is open over a list of existing rules. The rule being added is for 'MySQL Port Access' with source CIDR '0.0.0.0/0' and destination port '3306'. The 'IP Protocol' is set to 'TCP'. The 'Description' field contains 'MySQL Port Access'. The main table lists other rules, including ICMP and TCP rules for ports 22 and 3306.

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

The screenshot shows the 'Default Security List for myvcn' page. The 'Ingress Rules' section now includes the newly added rule for MySQL Port Access, along with the previous ICMP and TCP rules. The table shows five items total.

Selected	Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	22			TCP traffic for ports: 22 SSH Remote Login Protocol
<input type="checkbox"/>	No	0.0.0.0/0	ICMP		3, 4			ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set
<input type="checkbox"/>	No	10.0.0.0/16	ICMP		3			ICMP traffic for: 3 Destination Unreachable
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	3306			TCP traffic for ports: 3306
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	33060			TCP traffic for ports: 33060

You may now proceed to the next lab

Acknowledgements

MySQL Enterprise Edition - Implementation Essentials Bootcamp | Lab 2: Create Linux Compute Instance

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

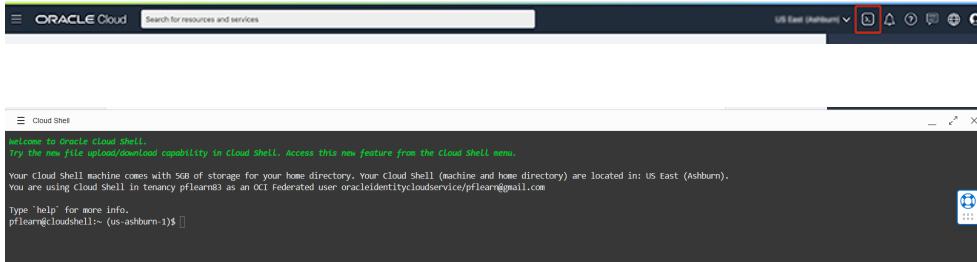
Prerequisites

- 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.

A screenshot of the Oracle Cloud Cloud Shell terminal showing the execution of the ssh-keygen command. The command "ssh-keygen -t rsa" is run, followed by prompts for saving the key pair and entering a passphrase. The output shows the keys being generated and saved to the .ssh directory. A large portion of the output is a long string of characters representing the public key fingerprint.

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
```

```
pFlearing@cloudshell:~ (us-ashburn-1)$ cd .ssh  
pFlearing@cloudshell:~:ssh (us-ashburn-1)$ ls  
id_rsa id_rsa.pub  
pFlearing@cloudshell:~:ssh (us-ashburn-1)$ []
```

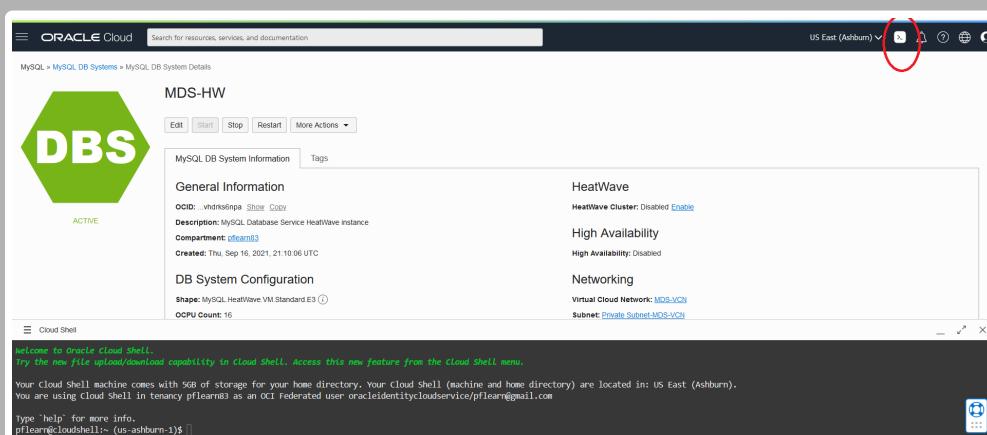
Note: in the output there are two files, a *private key*: `id_rsa` and a *public key*: `id_rsa.pub`. Keep the private key safe and don't share its content with anyone. The public key will be needed for various activities and can be uploaded to certain systems as well as copied and pasted to facilitate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

Task 2: Create Compute instance

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

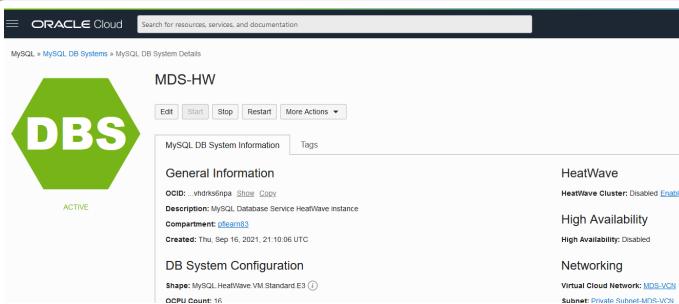
1. Before creating the Compute instance open a notepad
2. Do the followings steps to copy the public SSH key to the notepad

Open the Cloud shell



Enter the following command

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



```

Try the new file upload/download capability in Cloud Shell. Access this new feature from the Cloud Shell menu.
Your Cloud Shell machine comes with 5GB of storage for your home directory. Your Cloud Shell (machine and home directory) are located in: US East (Ashburn).
You are using Cloud Shell in tenancy pflearn03 as an OCI Federated user oracleidentitycloudservice/pfLearn@gmail.com

Type 'help' for more info.
pflearn@cloudshell:~$ cat .ssh/id_rsa.pub
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.2.20 (Darwin)
Comment: https://www.gnupg.org/ - https://gnupg.org/pgpkey/
-----END PGP PUBLIC KEY BLOCK-----
pflearn@cloudshell:~$ 

```

3. Copy the id_rsa.pub content the notepad

Your notepad should look like this

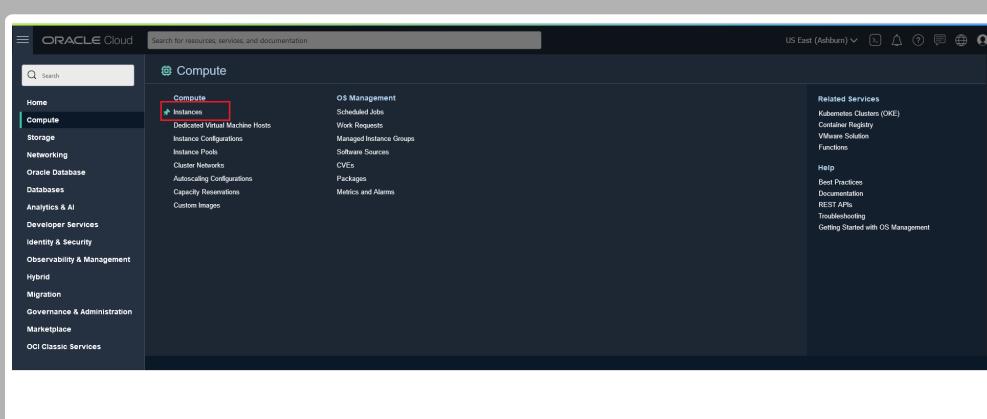
id-rsa.pub

```

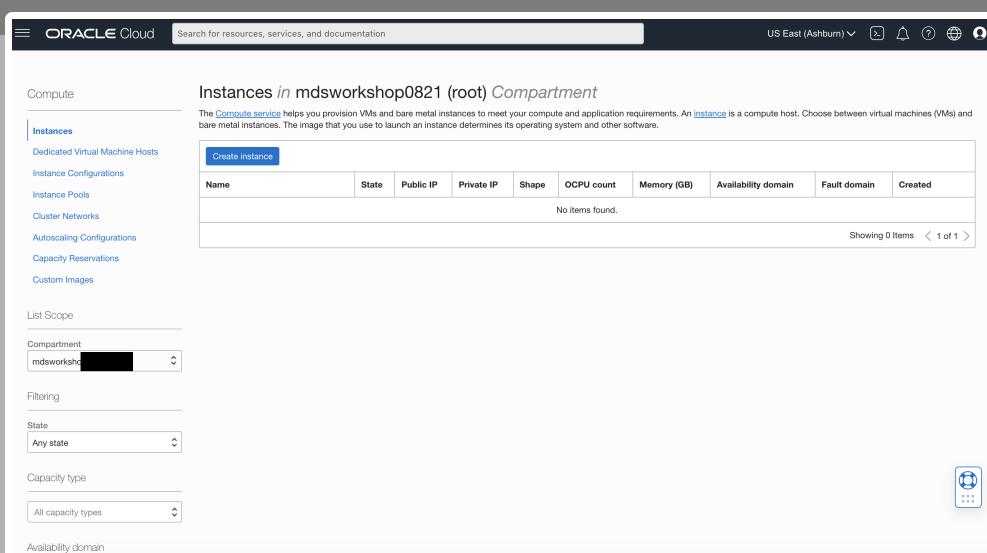
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDdBwp1v2CYDDZyQ1F+V3Fu5zyZY9augGBLbmT6+Mbt0bn+wgwgA+Gotn
+uVUGn3Z5AiXREi1h5ILdS8/6Ya46vvu66C0gRErgED2nIF737G8/zsM7P6hVMnRCQywvHzW9+yRcGN/XW
+Fx0Jgfmx0VV1WB/K6s81FD6TRohIhQgnUSxZr1ZhrgQAkjvsZfvU6ZrIGT70fy0+1gav0wu6bZy1vx94E5y6LigmPKiNB5b3rpCA6x
5a6uPa1FncdNTUFbBvY/XoBRQ3amPA8TTG8ZsYa3wZjQQxEu6bsu7udiQWZ3n9xNOxv0bDa4ay7DZ8r+1ewQC/+351crwlwxVj9
pflearn@6264b69ff0ec

```

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
priscilagalvao40 (root)/Test_1

Placement

Availability domain: AD-3 Always Free-eligible Capacity type: On-demand capacity

Fault domain: Let Oracle choose the best fault domain

Image and shape

Image: Oracle Linux 8 Image build: 2022.01.24-0

Shape: VM.Standard.E2.1.Micro Always Free-eligible
OCPU count: 1 Memory (GB): 1 Network bandwidth (Gbps): 0.48

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

- 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 creating an instance with Resource Manager.

Name: MDS-Client

Create in compartment: Test_1
priscilagalvao40 (root)/Test_1

Placement

Availability domain: AD-3 Always Free-eligible 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 allocated to a newly created instance.

Image: Oracle Linux 8 Image build: 2022.01.24-0

Shape:

AMD: VM.Standard.E2.1.Micro Always Free-eligible
Virtual machine, 1 core OCPU, 1 GB memory, 0.48 Gbps network bandwidth

Intel: VM.Standard.E2.2 Always Free-eligible
Virtual machine, 2 cores OCPUs, 16 GB memory, 1.4 Gbps network bandwidth

Ampere: VM.Standard1.1 Always Free-eligible
Arm-based processor, 1 core OCPU, 7 GB memory, 0.6 Gbps network bandwidth

Bare metal machine

Bare metal machine: VM.Standard1.2 Always Free-eligible
A bare metal compute instance gives you dedicated physical server access for highest performance and strong isolation.

Specialty and previous generation

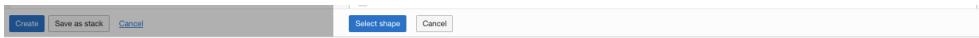
Specialty and previous generation: VM.Standard1.2 Always Free-eligible
Earlier generation AMD and Intel standard shapes. Always Free, Dense IO, GPU, and HPC shapes.

Image: Oracle Linux 8

Shape name	OCPU	Memory (GB)	Network bandwidth (Gbps)	Max. total VNICs
VM.Standard.E2.1.Micro	1	1	0.48	1
VM.Standard.E2.2	2	16	1.4	2
VM.Standard1.1	1	7	0.6	2
VM.Standard1.2	2	14	1.2	2
VM.Standard1.4	4	28	1.2	4

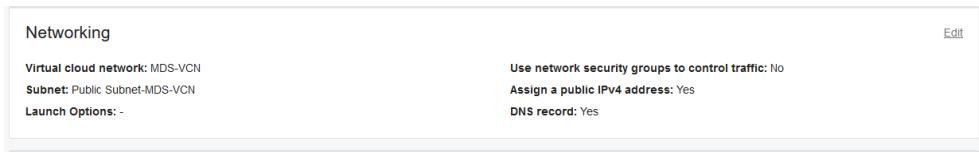
Networking

Virtual cloud network: MDS-VCN Subnet: Public Subnet-MDS-VCN Launch options: -

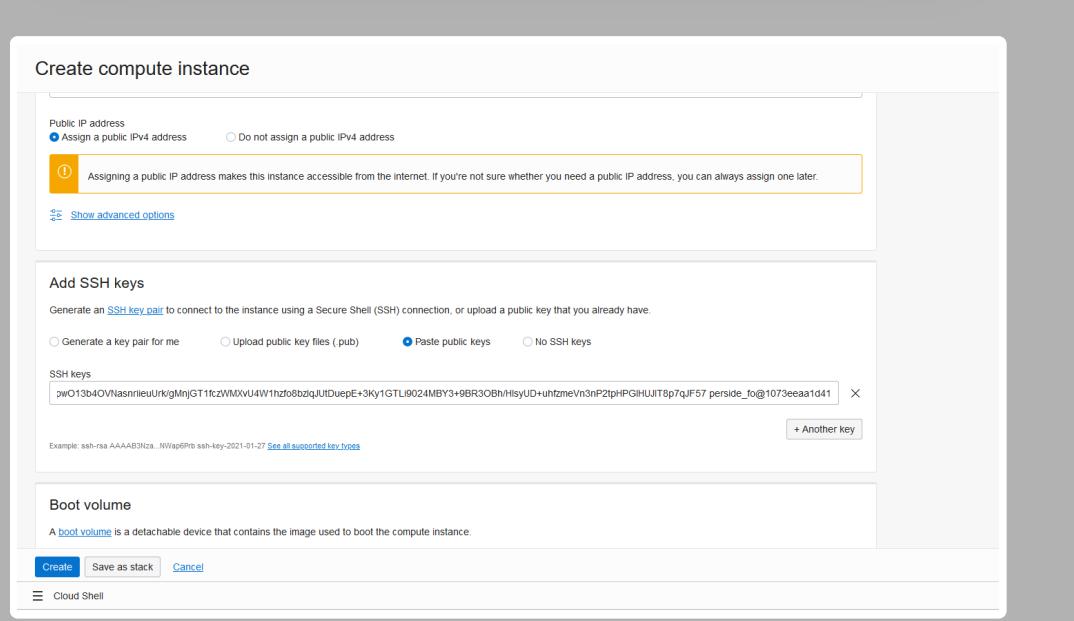


10. On Networking, make sure 'myvcn' is selected

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

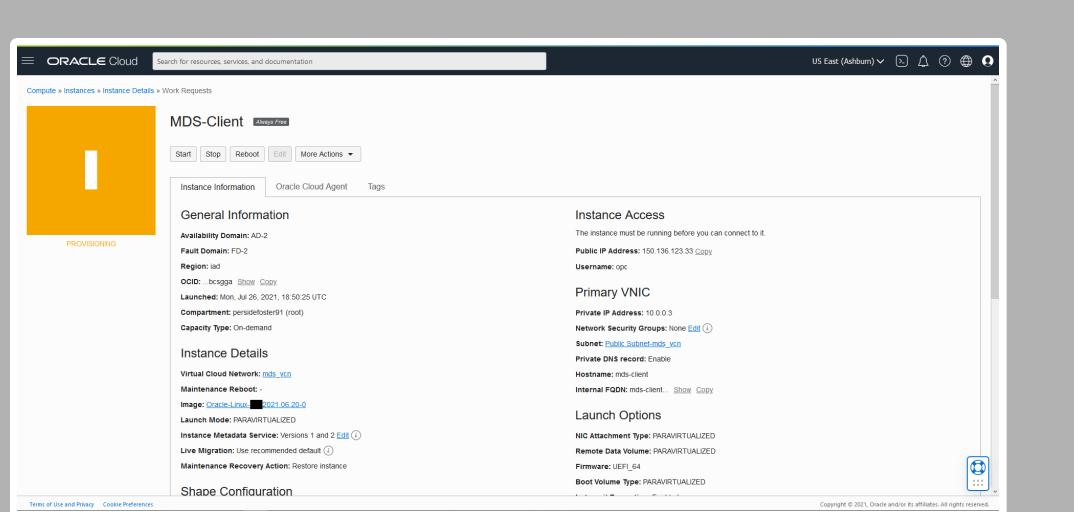


11. On Add SSH keys, paste the public key from the notepad.



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

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



14. 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 **myclient** Compute Instance link

c. Copy **myclient** plus the Public IP Address to the notepad

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

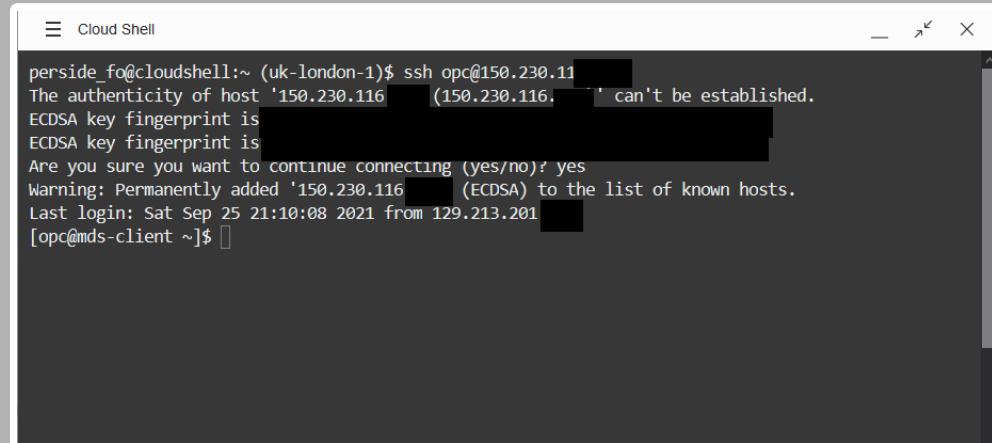
Enter the username **opc** and the Public IP Address.

Note: The **myclient** 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>
```



The screenshot shows a terminal window titled "Cloud Shell". The command entered is "ssh -i ~/.ssh/id_rsa opc@<your_compute_instance_ip>". The output shows the following:

```
perside_fo@cloudshell:~ (uk-london-1)$ ssh opc@150.230.116
The authenticity of host '150.230.116' can't be established.
ECDSA key fingerprint is [REDACTED]
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 [REDACTED]
[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

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 3: Setup

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. lecture

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

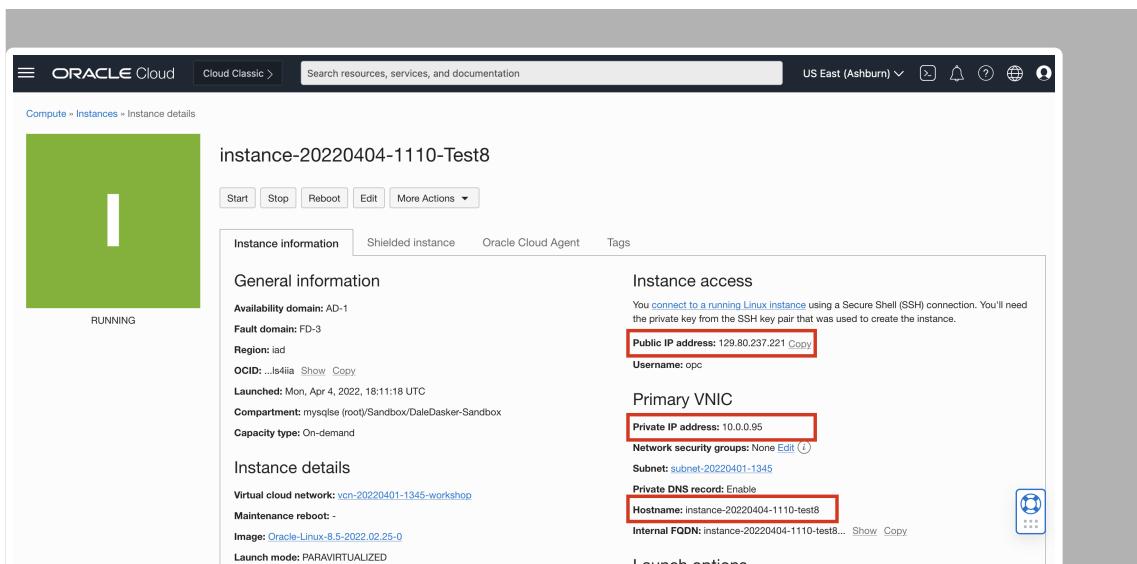
4. If you have not yet installed an SSH client on your laptop, please install one e.g. (windows)
<https://www.putty.org/>

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 shows the Oracle Cloud Instance details page for 'instance-20220404-1110-Test8'. The instance is listed as 'RUNNING'. The 'Instance information' tab is selected, displaying the following details:

- Availability domain: AD-1
- Fault domain: FD-3
- Region:iad
- OCID: ...ls4ia [Show Copy](#)
- Launched: Mon, Apr 4, 2022, 18:11:18 UTC
- Compartment: mysqse (rooty/Sandbox/DaleDasker-Sandbox)
- Capacity type: On-demand

The 'Instance access' section shows the Public IP address: 129.80.237.221 ([Copy](#)). The 'Primary VNIC' section shows the Private IP address: 10.0.0.95, Private DNS record: Enable, and Hostname: instance-20220404-1110-test8. The 'Launch options' section includes a 'Launch' button.

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

- 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

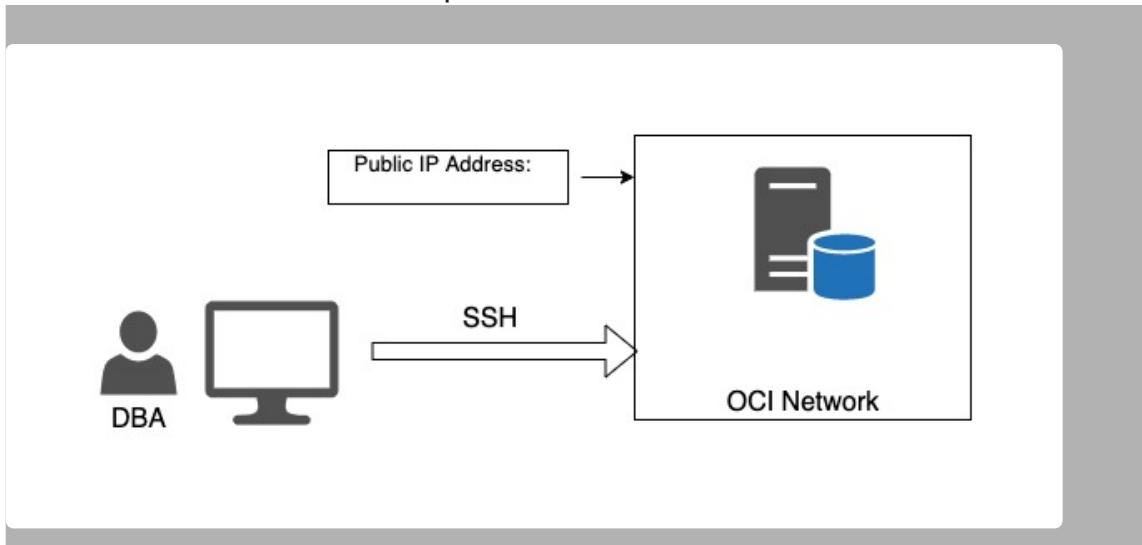
5. Linux **opc** user has limited privileges. To work with administrative privileges, use "sudo" like
shell> sudo su - root

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

- o a. SSH connection
- o b. SSH key file named "id_rsa" or "
- o c. username "opc"
- o d. no password
- o e. Public IP address of your assigned Linux VM (serverA, serverB)

4. Examples of connections:

Linux: use "id_rsa" key file

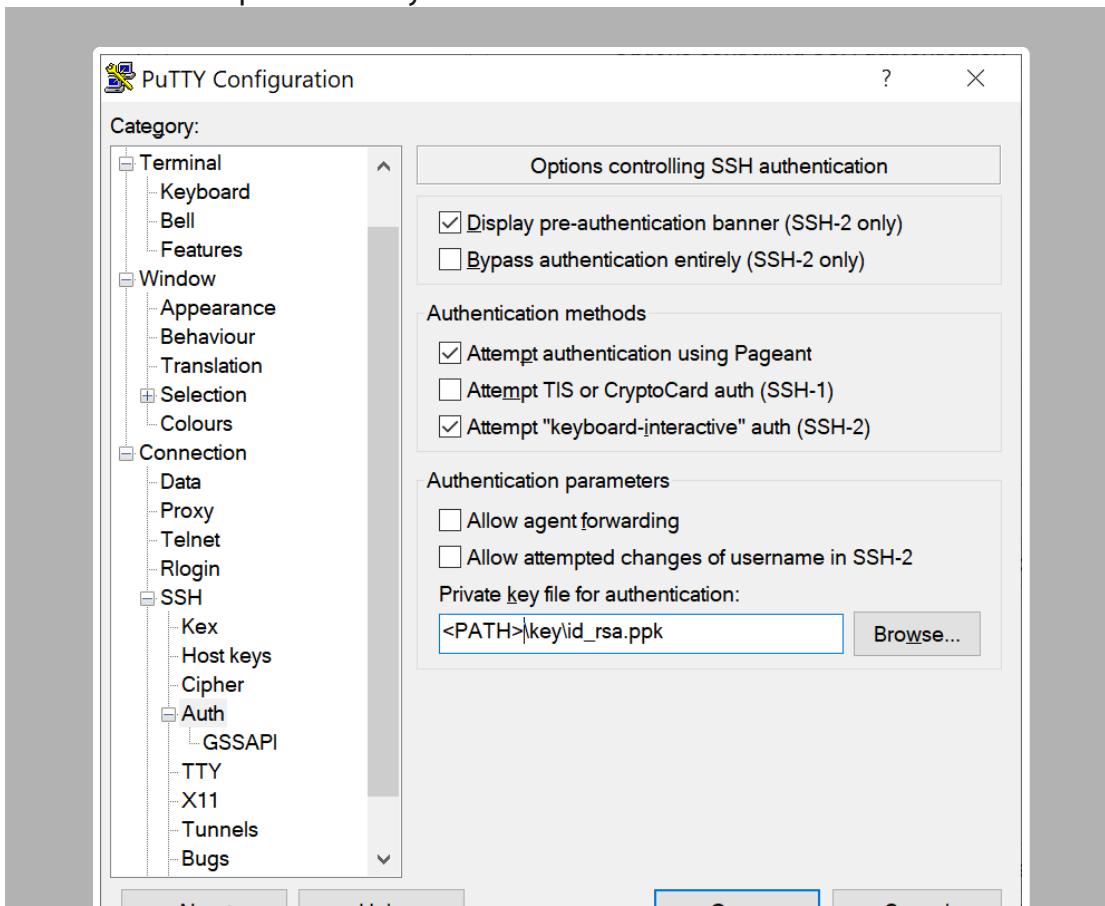
shell>

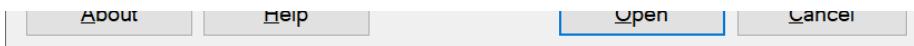
```
ssh -i id_rsa opc@public_ip
```

Windows: use "id_rsa.ppk" key file

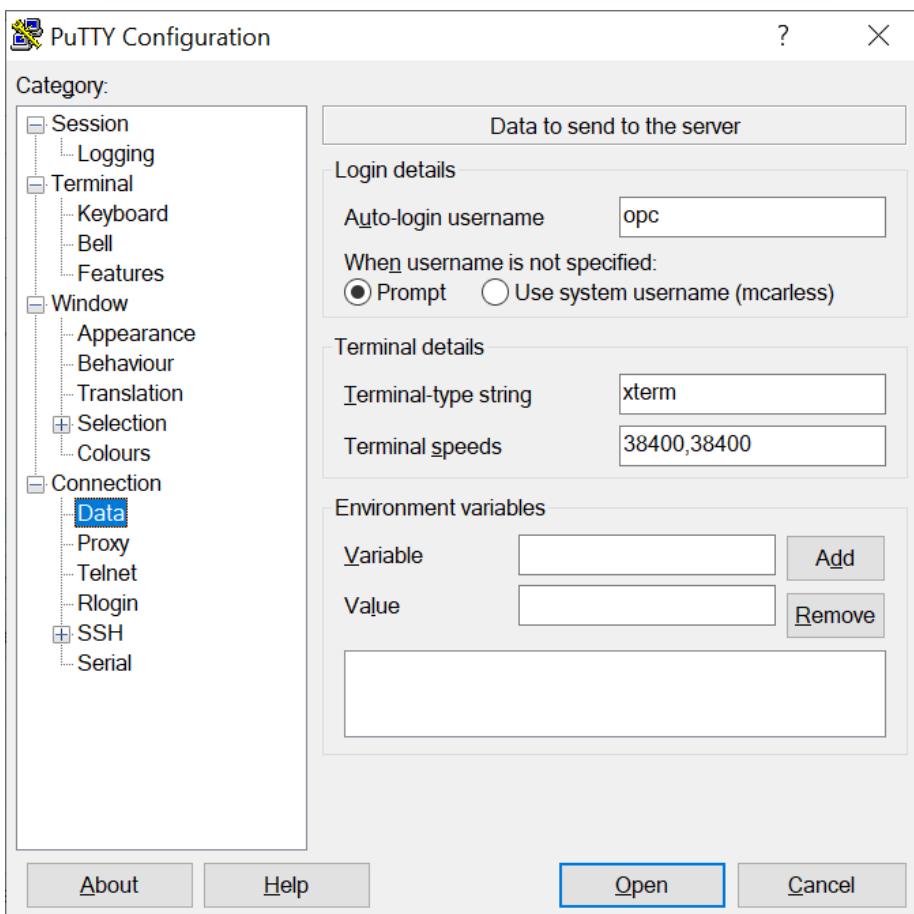
1. Open putty

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

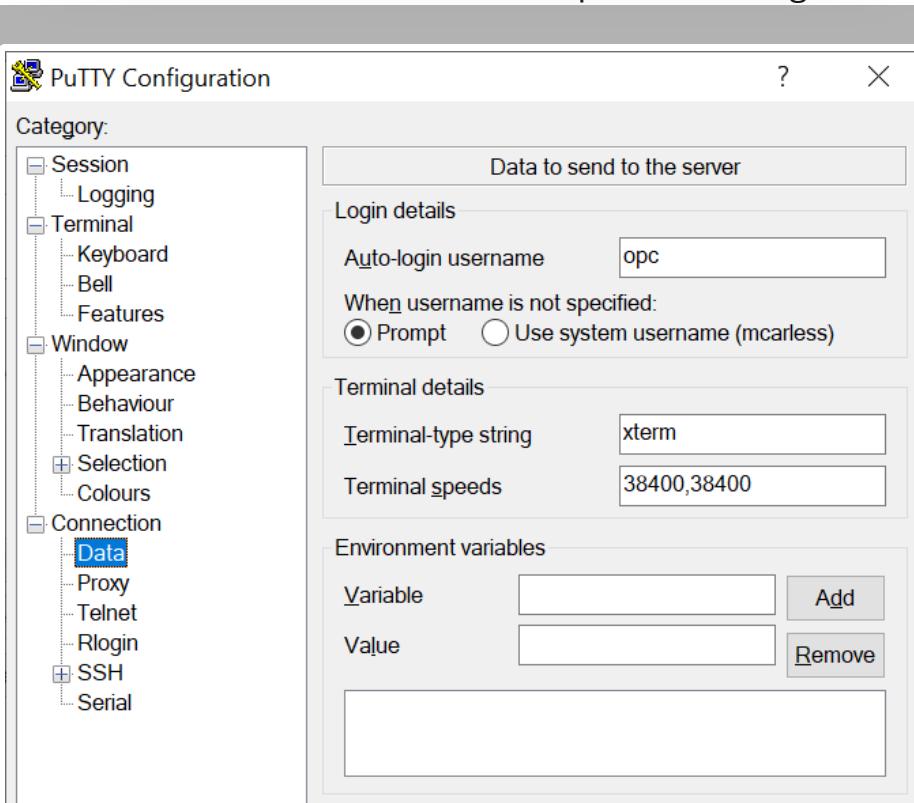


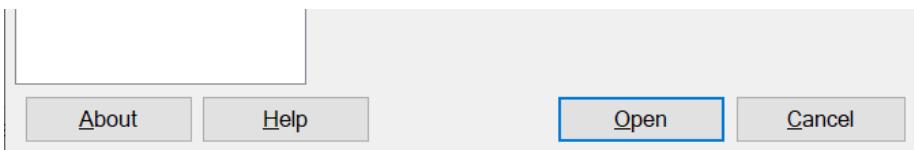


3. Choose "Connection SSH Auth" and provide the id_rsa.ppk path

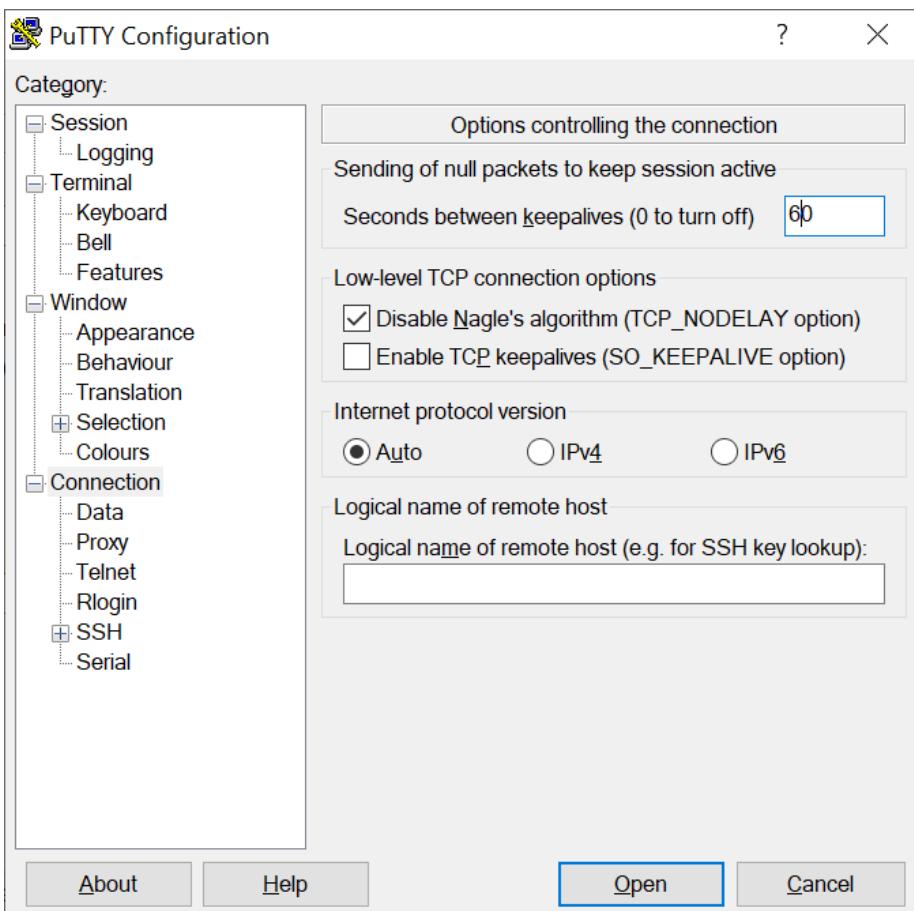


4. Select "Connection Data" and insert "opc" in "Auto-login username"

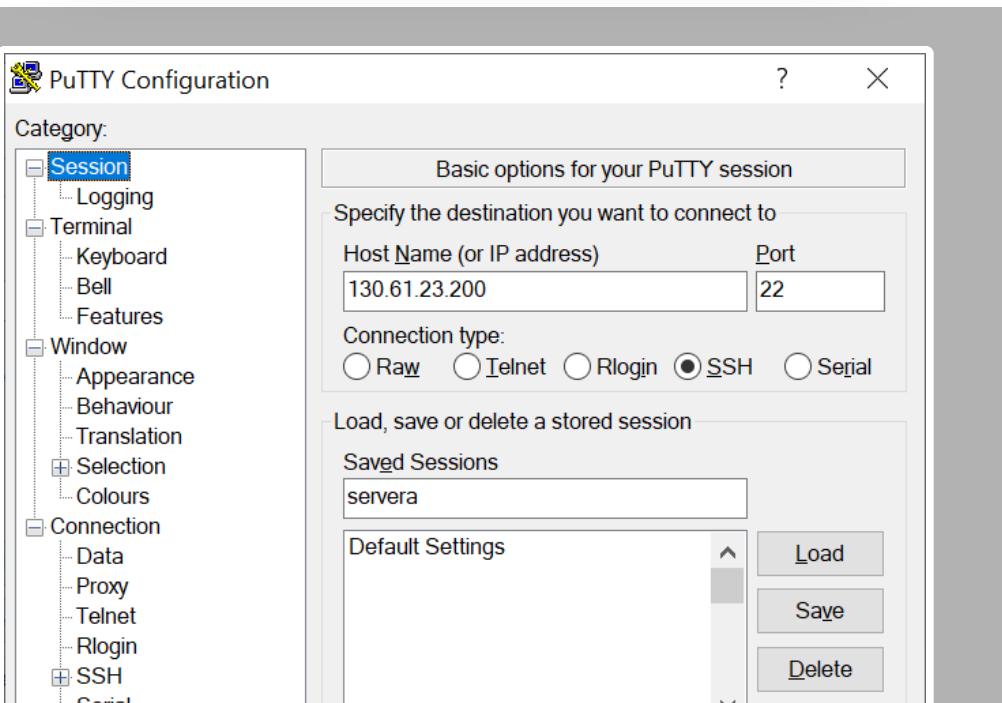


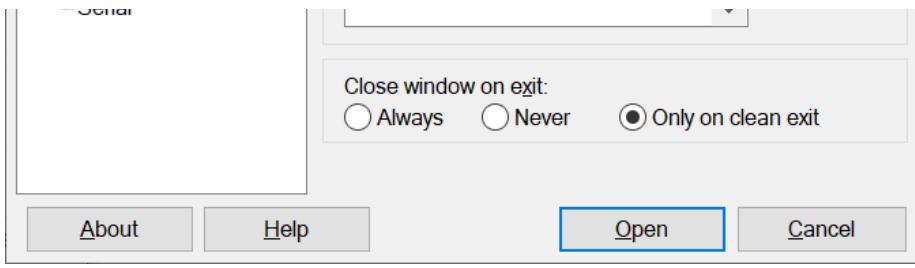


5. e) Choose Connection and insert "60" in "Seconds between keepalives"



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 wget https://objectstorage.us-ashburn-1.oraclecloud.com/p/votWQhR-wMX-ro12lv5DZ9XVHvGaNu5FcMEumkeKQ5gSOeEhjkXVKyWHdorYUyIL/n/idazzjlcjqzj/b/bucket-20230109-1614-Security_Workshop_01252023/o/workshop.tar.gz
```

4. Extract workshop files

shell>

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

Learn More

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

Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, January 2023

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 4: Install - MySQL Enterprise Edition

INSTALL - MYSQL ENTERPRISE EDITION

Introduction

Detailed Installation of MySQL Enterprise Edition 8.0 and MySQL Shell on Linux

Objective: RPM Installation of MySQL 8 Enterprise on Linux

RPM 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:

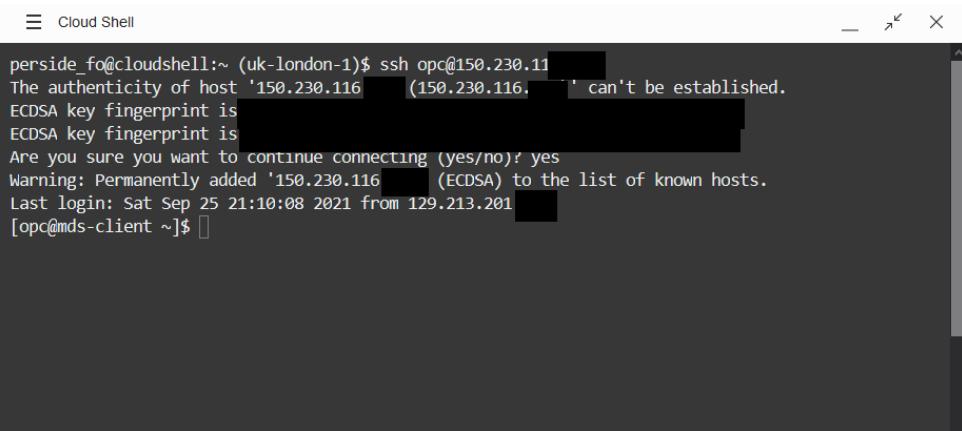
- An Oracle account
- All previous labs successfully completed
- Lab standard
 - shell> the command must be executed in the Operating System shell
 - mysql> the command must be executed in a client like MySQL, MySQL Workbench
 - mysqlsh> the command must be executed in MySQL shell

Task 1: Install MySQL Enterprise Edition using Linux RPM's

Note: If not already connected with SSH

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

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



A screenshot of a Cloud Shell terminal window. The title bar says "Cloud Shell". The terminal output shows:

```
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 [REDACTED]  
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 [REDACTED]  
[opc@mds-client ~]$
```

1. Install the RPM's

shell>

```
cd /workshop
```

shell>

```
sudo yum -y install *.rpm
```

Task 2: Start and test MySQL Enterprise Edition Install

1. Start your new mysql instance

shell>

```
sudo systemctl start mysqld
```

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>

```
sudo grep -i ready /var/log/mysqld.log
```

4. Retrieve root password for first login:

shell>

```
sudo grep -i 'temporary password'  
/var/log/mysqld.log
```

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

shell>

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

6. Create New Password for MySQL Root

mysqlsh>

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

```
mysqlsh>
```

```
status
```

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

```
mysqlsh>
```

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

```
mysqlsh>
```

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

```
mysqlsh>
```

```
quit
```

Learn More

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

Acknowledgements

- **Author** - Dale Dasker, MySQL Solution Engineering
- **Last Updated By/Date** - <Dale Dasker, January 2023

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 5: Install - Verify MySQL Enterprise Edition

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: -- 10 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
```

shell>

```
mysql -uroot -pWelcome1! -P3306 -h 127.0.0.1 <
./employees.sql
```

Task 4: Learn Useful SQL Statements

1. **shell>**

```
mysql -uroot -pWelcome1! -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 VARIABLESG
```

mysql>

```
SHOW GLOBAL STATUSUG
```

mysql>

```
SHOW FULL PROCESSLIST;
```

```
mysql>
```

```
SHOW ENGINE INNODB STATUSG
```

```
exit
```

Learn More

- [MySQL Tutorial](#)

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 6: Security - MySQL Users

🌐 Web Clip

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 -pWelcome1! -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: 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';
```

Task 3: 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 -pWelcome1! -h 127.0.0.1 -P  
3306
```

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

mysql>

```
USE employees;
```

mysql>

```
SELECT * FROM employees;
```

Task 4: Use appuser1 connection - **** OPTIONAL ****

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

mysql>

```
exit
```

shell>

```
mysql -u appuser1 -pWelcome1! -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 5: Restore user privileges ** OPTIONAL ******

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';
```

Learn More

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 7: Security - MySQL Enterprise Audit

🌐 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. Load Audit functions. If running in a replicated environment, load the plugin no each of the Replicas first and then modify the SQL script to only load the functions. **shell>**

```
mysql -uroot -pWelcome1! -h 127.0.0.1 -P 3306 <
/usr/share/mysql-
8.0/audit_log_filter_linux_install.sql
```

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

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

- c. Add the following lines to the bottom of the file. These lines will make sure that the audit plugin can't be unloaded and that the file is automatically rotated at 20 MB and format of data is JSON.

shell>

```
plugin-load=audit_log.so
audit_log=FORCE_PLUS_PERMANENT
audit_log_rotate_on_size=20971520
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>

```
sudo service mysqld restart
```

3. Connect to your mysql-enterprise with administrative user

shell>

```
mysql -uroot -pWelcome1! -h 127.0.0.1 -P 3306
```

a. Using the Administrative Connection, create a Audit Filter for all activity and all users. Privileges required are AUDIT_ADMIN and SUPER

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>

```
sudo tail -f /var/lib/mysql/audit.log
```

Task 2: Use Audit

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

a. **shell>**

```
mysql -u appuser1 -pWelcome1! -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 -pWelcome1! -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_events  
'');
```

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 -pWelcome1! -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;
```

e. **mysql>**

```
quit;
```

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

a. **shell>**

```
mysql -u appuser2 -pWelcome1! -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 table accesss. Using the Administrative Connection, create a Audit Filter for tables

a. Remove previous filter:

mysql>

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

mysql>

```
SELECT audit_log_filter_flush();
```

b. **mysql>**

```
SELECT
audit_log_filter_set_filter('log_table_access_events
', '{ "filter": { "class": { "name": "table_access"
} } }');
```

c. **mysql>**

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

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

```
mysql -u appuser1 -pWelcome1! -h127.0.0.1 -P
3306
```

e. **mysql>**

```
USE employees;
```

f. **mysql>**

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

g. **mysql>**

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

8. 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');
```

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

a. **shell>**

```
mysql -u appuser1 -pWelcome1! -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;
```

10. 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_filterG
```

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

```
SELECT * FROM mysql.audit_log_userG
```

c. Reading from Audit Log within MySQL Client **mysql>**

```
SELECT JSON_PRETTY(CONVERT(audit_log_read(audit_log_read_bookmark()) using  
utf8mb4))G
```

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

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

e. Check what Audit Functions are available **mysql>**

```
SELECT * FROM mysql.func;
```

f. Check that the Audit plugin loaded **mysql>**

```
SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS WHERE  
PLUGIN_NAME LIKE 'audit%';
```

11. 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, January 2023

MySQL Enterprise Edition - Implementation Essentials

Bootcamp | Lab 8: Security - MySQL Enterprise

Transparent Data Encryption (TDE)

🌐 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:

- InnoDB Data At Rest

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 -pWelcome1! -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 /etc/my.cnf

shell>

```
sudo nano /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=/var/lib/mysql-
keyring/keyring-encrypted
keyring_encrypted_file_password=V&rySec4eT
```

- c. Restart MySQL

shell>

```
sudo service mysqld restart
```

3. "Spy" on employees.employees table

a. **shell>**

```
sudo strings  
"/var/lib/mysql/employees/employees.ibd" | head -  
n50
```

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

a. **shell>**

```
mysql -u root -pWelcome1! -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>**

```
sudo strings  
"/var/lib/mysql/employees/employees.ibd" | head -  
n50
```

6. Administrative commands

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

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

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>**

```
sudo strings "/var/lib/mysql/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>**

```
sudo strings "/var/lib/mysql/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'G
```

Learn More

MySQL Enterprise Edition - Implementation Essentials Bootcamp | Lab 9: Security - MySQL Data Masking

🌐 Web Clip

SECURITY - DATA MASKING

Introduction

Data Masking and de-identification MySQL Enterprise Masking and De-identification provides an easy to use, built-in database solution to help organizations protect sensitive data from unauthorized uses by hiding and replacing real values with substitutes. Objective: Install and use data masking functionalities

Estimated Lab Time: -- 12 minutes

Objectives

In this lab, you will:

- Create sample data with random generation utilites which are part of Enterprise Masking
- Test Masking of Sensitive Data
- Create a View and user which only sees masked data

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:

- Data masking has more functions than what we test in the lab. The full list of functions is here
- <https://dev.mysql.com/doc/refman/8.0/en/data-masking-usage.html>

Task 1: Install masking plugin

1. To install the data masking plugin, execute with statements

a. **shell>**

```
mysql -uroot -pWelcome1! -h 127.0.0.1 -P 3306
```

b. **mysql>**

```
INSTALL PLUGIN data_masking SONAME  
'data_masking.so';
```

c. **mysql>**

```
SHOW PLUGINS;
```

2. Look for data_masking and check the status? Is it active?

Task 2: Use masking functions

1. Install masking functions

a. **mysql>**

```
CREATE FUNCTION gen_range RETURNS INTEGER SONAME  
'data_masking.so';  
CREATE FUNCTION gen_rnd_email RETURNS STRING SONAME  
'data_masking.so';  
CREATE FUNCTION gen_rnd_us_phone RETURNS STRING  
SONAME 'data_masking.so';  
CREATE FUNCTION gen_rnd_ssn RETURNS STRING SONAME
```

```
'data_masking.so';
CREATE FUNCTION mask_inner RETURNS STRING SONAME
'data_masking.so';
CREATE FUNCTION mask_outer RETURNS STRING SONAME
'data_masking.so';
CREATE FUNCTION mask_ssn RETURNS STRING SONAME
'data_masking.so';
```

2. Use data masking functions

a. **mysql>**

```
SELECT mask_inner(last_name, 2,1) FROM
employees.employees limit 10;
```

b. **mysql>**

```
SELECT mask_outer(last_name, 2,1) FROM
employees.employees limit 10;
```

Task 3: Discussion and use Masking functions and random generators

1. Create Table to generate and add masking data

mysql>

```
CREATE TABLE employees_mask LIKE employees;
```

2. Add data to newly created table

mysql>

```
INSERT INTO employees_mask SELECT * FROM
employees;
```

3. Create new column for SSN's

mysql>

mysql>

```
ALTER TABLE employees_mask ADD COLUMN ssn  
varchar(11);
```

4. Create new column for emails's

mysql>

```
ALTER TABLE employees_mask ADD COLUMN email  
varchar(40);
```

5. Use Functions to generate sample SSN data

mysql>

```
UPDATE employees_mask SET ssn = genRndSSN() WHERE  
1;
```

6. Use Functions to generate sample Email data

mysql>

```
UPDATE employees_mask SET email = genRndEmail()  
WHERE 1;
```

7. Let's look at the data that we just created

mysql>

```
SELECT * FROM employees_mask LIMIT 5;
```

8. Let's mask the SSN

mysql>

```
SELECT  
emp_no,first_name,last_name,maskSSN(CONVERT(ssn  
USING latin1) AS ssn FROM employees_mask T TMTT
```

```
USING latin1), AND SSN FROM employees_mask LIMIT  
5;
```

9. Let's create a view which only shows the masked data

mysql>

```
CREATE VIEW masked_customer AS SELECT  
emp_no,first_name,last_name,mask_ssn(CONVERT(ssn  
USING latin1)) AS ssn FROM employees_mask;
```

10. Let's create a user who only has access to the view with the masked data

mysql>

```
CREATE USER 'accounting'@'%' IDENTIFIED BY  
'Pa33word!';
```

mysql>

```
GRANT SELECT ON employees.masked_customer TO  
'accounting'@'%';
```

11. Log in with new user account and run queries

mysql>

```
quit;
```

shell>

```
mysql -uaccounting -pPa33word! -h 127.0.0.1 -P  
3306
```

mysql>

```
SELECT * FROM employees.masked_customer LIMIT 5;
```

12. Try accessing table that is not masked

mysql>

```
SELECT * FROM employees.employees_mask LIMIT 5;
```

Task 4: * OPTIONAL *** Discussion and use Masking functions and random generators**

1. Discuss differences between mask_inner and mask_outer

mysql>

```
SELECT mask_inner(last_name, 1, 1, '&') FROM  
employees.employees limit 1;
```

2. Use data masking random generators to these statements several times

a. **mysql>**

```
SELECT gen_range(1, 200);
```

b. **mysql>**

```
SELECT genRndUsPhone();
```