

# Migration Guide: MariaDB to HeatWave MySQL on Oracle Cloud Infrastructure (OCI)

---

## Before you start:

- You must have an account on Oracle Cloud Infrastructure (OCI).
- Some OCI knowledge is preferred.
- This migration document only covers how to migrate your database from MariaDB to HeatWave MySQL on OCI. Before performing the migration, you should have considered downtime (the length of the downtime will mostly depend on the size of your database and checks you may want to perform before bringing your database back online), application compatibility, current database metrics (CPU, storage size, RAM, max number of concurrent users, backups, binary logs expiration, number of replicas if any, etc.), desired database metrics, networking, security, user testing, etc.
- The migration method shown in this guide works for MariaDB 5.5 and MariaDB 10.3 to 10.11. It could work with more recent versions but has not been tested.
- When following the guide, you should always execute the commands/steps shown as an admin/root user wherever applicable.
  - On OCI you must have the ability to create and manage resources.
  - For your MariaDB instance, use an admin/root user.
- You do not need to make any configuration changes to your on-premises MariaDB for this migration unless it's explicitly stipulated.
- If you have replication configured in your current MariaDB environment, you can perform the migration steps shown in this guide from either your source or replica instance.
- The Overview section of this migration guide contains all the steps that are needed to complete the database migration from on-premises MariaDB to HeatWave MySQL on OCI.
- In the Walkthrough section of this guide, we will apply the information provided in the Overview section and give you a simple step-by-step guide. In this step-by-step guide, we will have an on-premises MariaDB instance with some sample data pre-loaded and will migrate it over to HeatWave MySQL on OCI. This will help you follow and better visualize the process/information provided in the Overview section.
- You can use the Walkthrough section's step-by-step guide as a reference for your migration from on-premises MariaDB to HeatWave MySQL. When following the guide, make changes along the way to your on-premises and OCI environment accordingly or as required. Since each user following the step-by-step guide will have their environments configured differently, we cannot provide an ideal example that works for everyone.

## Overview:

Following are the required steps to migrate data from MariaDB to HeatWave MySQL on OCI:

### I) Have an Oracle Cloud Infrastructure (OCI) account.

OCI Sign in/Sign up page: <https://cloud.oracle.com>

### II) Set up a VPN connection from OCI to on-premises.

[A VPN connection will allow you to bridge your on-premises network with the OCI VCN. The VPN connection will allow your MariaDB to connect to HeatWave MySQL on OCI and it also ensures that your data in transit is encrypted while it is being migrated.]

VPN Connection to on-premises: <https://docs.public.oneportal.content.oci.oraclecloud.com/en-us/iaas/mysql-database/doc/vpn-connection.html>

### III) On OCI, create a HeatWave MySQL instance.

[You can create either a Standalone or High Availability HeatWave MySQL instance. Both options are fully-managed.]

Provision OCI HeatWave MySQL: <https://docs.oracle.com/en-us/iaas/mysql-database/doc/creating-db-system1.html>

### IV) Install MySQL Shell 8.2.1 (or above) on an on-premises instance that can connect to your MariaDB.

[MySQL Shell will be used to copy DDL and data from MariaDB to HeatWave MySQL on OCI. You must download MySQL Shell 8.2.1 or above.]

Download MySQL Shell: <https://dev.mysql.com/downloads/shell/>

Install MySQL Shell: <https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install.html>

### V) Connect to the MariaDB using MySQL Shell. And perform some compatibility checks.

MariaDB is no longer a drop-in replacement for MySQL and some checks are mandatory before the migration.

### VI) Afterwards, execute the MySQL Shell `util.copyInstance()` utility to export all schemas (including users, indexes, routines, triggers) from MariaDB to the HeatWave MySQL on OCI.

[The dump created by MySQL Shell's instance copy utility comprises DDL files specifying the schema structure, and tab-separated `.tsv` files containing the data.]

MySQL Shell Copy Utilities: <https://dev.mysql.com/doc/mysql-shell/8.2/en/mysql-shell-utils-copy.html>

### VI) (Optional) On OCI, use the Cloud Shell to verify whether the data was migrated successfully from MariaDB to HeatWave MySQL on OCI.

[Cloud Shell is a web browser-based terminal accessible from the Oracle Cloud Console.]

OCI Cloud Shell: <https://docs.oracle.com/en-us/iaas/Content/API/Concepts/cloudshellintro.htm>

### VII) (Optional) On OCI, if the HeatWave option was enabled during HeatWave MySQL DB creation, add the HW Cluster and load data from MySQL InnoDB storage into the HW Cluster using automation.

[Attaching the HeatWave in-memory Cluster combines transactions, analytics, and machine learning services into one MySQL Database.]

Add a HeatWave Cluster: <https://docs.oracle.com/en-us/iaas/mysql-database/doc/adding-heatwave->

[cluster.html#GUID-2335AC1F-FB01-4701-9EFD-810A3489A850](#)

Load Data into HeatWave: <https://dev.mysql.com/doc/heatwave/en/mys-hw-auto-parallel-load.html>

## Walkthrough:

### I) Have an Oracle Cloud Infrastructure (OCI) account.

OCI Sign in/Sign up page: <https://cloud.oracle.com>

### II) Set up a VPN connection from OCI to on-premises.

Note: this guide uses OpenVPN Access Server which lets you connect your MariaDB with HeatWave MySQL on OCI. You cannot use OpenVPN Access Server to connect entire sites or networks to an Oracle VCN; in that scenario, it is recommended to use [Site-to-site VPN](#) or [FastConnect](#).

1. Below is the MariaDB instance version and [the sample database \("world"\)](#) that will be migrated for this guide. The sample world database consists of 3 tables.

```
SQL select @@version;
+-----+
| @@version |
+-----+
| 11.1.2-MariaDB |
+-----+
1 row in set (0.0002 sec)
MySQL 127.0.0.1:3306 2023-10-29 21:57:01
SQL show schemas;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test |
| world |
+-----+
6 rows in set (0.0010 sec)
MySQL 127.0.0.1:3306 2023-10-29 21:57:11
SQL show tables in world;
+-----+
| Tables_in_world |
+-----+
| city |
| country |
| countrylanguage |
+-----+
3 rows in set (0.0003 sec)
```

2. Log in to [OCI](#) and create a VCN. Open the navigation menu, click **Networking**, and click **Virtual cloud networks**.

- Ensure you are in your desired compartment - we have chosen the `root` compartment. Click **Start VCN Wizard**.

- Select **Create VCN with Internet Connectivity** and click **Start VCN Wizard**.

- Enter a **VCN name** and **configure your VCN's IPv4 CIDR block - including the public and the private subnet**. The guide uses the default values for all. Make sure that the OCI VCN IPv4 CIDR block does not overlap with your on-premises network.

## Create a VCN with internet connectivity

Help

- 1 Configuration
- 2 Review and create

### Configuration

Resource availability checked successfully. [Close]

#### Basic information

VCN name ⓘ  
MySQL-VCN

Compartment ⓘ  
[root]

#### Configure VCN

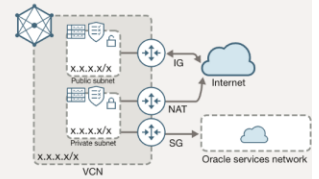
VCN IPv4 CIDR block ⓘ  
10.0.0.0/16

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

IPv6 prefixes *Optional*  
 Enable IPv6 in this VCN

DNS resolution  
 Enable DNS resolution for this VCN

#### VCN with internet connectivity



- Includes:**
- Virtual cloud network (VCN)
  - Public subnet
  - Private subnet
  - Internet gateway (IG)
  - NAT gateway (NAT)
  - Service gateway (SG)

[Next] [Cancel]

- Click **Next** after the configuration for your VCN is completed.

The screenshot shows the Oracle Cloud console interface for creating a VCN. The page title is "Create a VCN with internet connectivity". The navigation pane on the left shows two steps: "1 Configuration" (active) and "2 Review and create". The main content area is divided into sections: "DNS resolution" with a checked option "Use DNS hostnames in this VCN", "Configure public subnet" with an IP address type dropdown set to "IPv4 CIDR block" and a value of "10.0.0/24", and "Configure private subnet" with an IP address type dropdown set to "IPv4 CIDR block" and a value of "10.0.1.0/24". At the bottom, there are "Next" and "Cancel" buttons.

- On the Review and create page, validate the information for your VCN and click **Create**.

The screenshot shows the Oracle Cloud console interface for reviewing and creating the VCN. The page title is "Create a VCN with internet connectivity". The navigation pane on the left shows two steps: "1 Configuration" and "2 Review and create" (active). A success message "Resource availability checked successfully." is displayed at the top. The main content area shows the "Oracle VCN" details: Name: MySQL-VCN, Compartment: (root), Tags: VCN: VCN-2023-05-15T14:57:35, IPv4 CIDR block: 10.0.0.0/16, DNS label: MySQLVCN, and DNS domain name: MySQLVCN.oraclevcn.com. Below this, the "Subnets" section shows the "Public subnet" details: Subnet name: public subnet-MySQL-VCN, IPv4 CIDR block: 10.0.0.0/24, Security list name: default security list for MySQL-VCN, and Route table name: default route table for MySQL-VCN. At the bottom, there are "Previous", "Create", and "Cancel" buttons.



8. Click **View VCN** after your VCN creation has been completed.

9. From the OCI navigation menu, click **Networking** and click **Site-to-Site VPN**.

10. Click **marketplace solution** on the right side of the page.

11. On the OpenVPN Access Server page, from the dropdown, **select the compartment where your VCN resides**. Check the **terms of use and conditions** checkbox and click **Launch Stack**.

12. On the **Stack information** page of **Create stack**, leave everything as-is and click **Next**.

The screenshot shows the 'Create stack' page in Oracle Cloud. The page title is 'Create stack' and the breadcrumb is 'Stack information'. The page is divided into three steps: 1. Stack information (selected), 2. Configure variables, and 3. Review. The main content area is titled 'Stack information' and contains a diagram of an 'OpenVPN Access Server' (represented by three server icons). Below the diagram is a section for 'Custom providers' with a checkbox for 'Use custom Terraform providers' and a link to 'Store custom Terraform providers in a bucket'. The 'Name' field is optional and contains 'OpenVPN Access Server-20230515143705'. The 'Description' field is optional and contains 'Installs Access Server and configures the needed Security Lists, Network Security Groups, and any other needed resources. Assigns a reserved public IP address to the Access Server.' At the bottom, there is a 'Create in compartment' section and a 'Next' button.

13. On the **Configure variables** page, under **Compute Shape** select either **VM.Standard2.2** or **VM.StandardE2.2**. For **Application Configuration**, create an admin username and password. Make a note of the admin credentials.

The screenshot shows the 'Create stack' page in Oracle Cloud, Step 2: Configure variables. The page title is 'Create stack' and the breadcrumb is 'Configure variables'. The page is divided into three steps: 1. Stack information, 2. Configure variables (selected), and 3. Review. The main content area is titled 'Compute Configuration' and contains a form for 'OpenVPN Access Server Name' with the value 'openvpn\_access\_server'. Below this is a 'Compute Shape' dropdown menu with 'VM.Standard2.2' selected. The 'Application Configuration' section contains a form for 'Administrator Username' with the value 'root', 'Administrator Password' (masked with dots), and 'Activation Key' (optional). At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons.

- For **Network Configuration**, under **Network Strategy**, select **Use Existing VCN** and select the **VCN that we created earlier** from the **Existing Network** dropdown. For the **Existing Subnet**, select the **Public Subnet** of your VCN. Under **Additional Configuration**, ensure the compartment is where your VCN resides. Click **Next**.

The screenshot shows the 'Create stack' page in Oracle Cloud, specifically the 'Network Configuration' step. The page has a dark header with the Oracle Cloud logo, a search bar, and the region 'US East (Ashburn)'. On the left, there are three steps: 'Stack information', 'Configure variables', and 'Review'. The 'Network Configuration' section includes a 'Network Strategy' dropdown set to 'Use Existing VCN', an 'Existing Network' dropdown set to 'MySQL-VCN', and an 'Existing Subnet' dropdown set to 'public subnet-MySQL-VCN (Regional)'. Below this is the 'Additional Configuration' section, where the 'Compartment' dropdown is set to '(root)'. There is also a field for 'Public SSH Key string' with the label 'Optional'. At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons.

- On the Review page of Create stack, click **Create**.

The screenshot shows the 'Create stack' page in Oracle Cloud, specifically the 'Review' step. The page has the same dark header as the previous screenshot. On the left, the 'Review' step is highlighted. The main content area shows a summary of the configuration. It includes a 'Stack information' table with the following details:

Name	OpenVPN Access Server-20230515174018
Description	...erver. <a href="#">Show</a> <a href="#">Copy</a>
Compartment	...qedpia <a href="#">Show</a> <a href="#">Copy</a>
Terraform version	0.14.x

Below this is a 'Compute Configuration' table:

Compute Shape	VM.Standard2.2
---------------	----------------

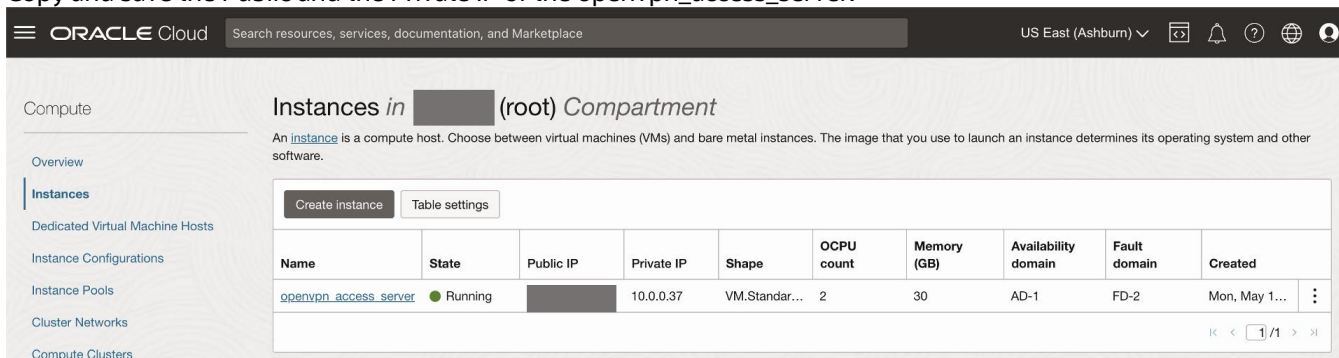
And finally, an 'Application Configuration' table:

Administrator Username	root
Administrator Password	*****

At the bottom, there are 'Previous', 'Create', and 'Cancel' buttons.

- Finishing the previous step will provision a compute instance for the VPN. From the OCI navigation menu, click **Compute** and click **Instances**. It may take a few minutes for your compute host to be ready.

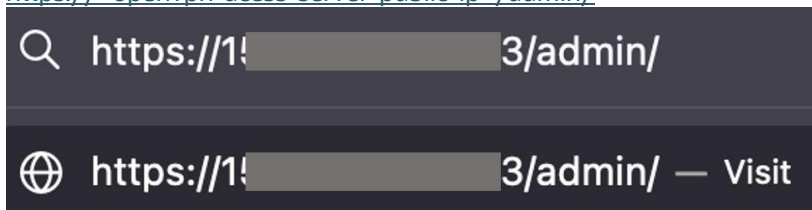
17. Copy and save the Public and the Private IP of the `openvpn_access_server`.



Name	State	Public IP	Private IP	Shape	OCPU count	Memory (GB)	Availability domain	Fault domain	Created
<a href="#">openvpn_access_server</a>	Running	[REDACTED]	10.0.0.37	VM.Standard...	2	30	AD-1	FD-2	Mon, May 1...

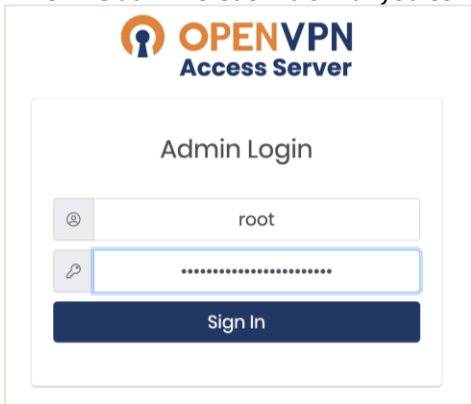
18. Open a web browser and enter the following in the search bar.

<https://<openvpn-access-server-public-ip>/admin/>

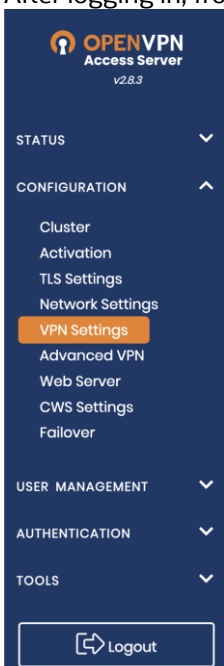


Note: in the web browser when prompted, click **Advanced** and click **Proceed to <openvpn-access-server-public-ip> (unsafe)** or **Accept the Risk and Continue**.

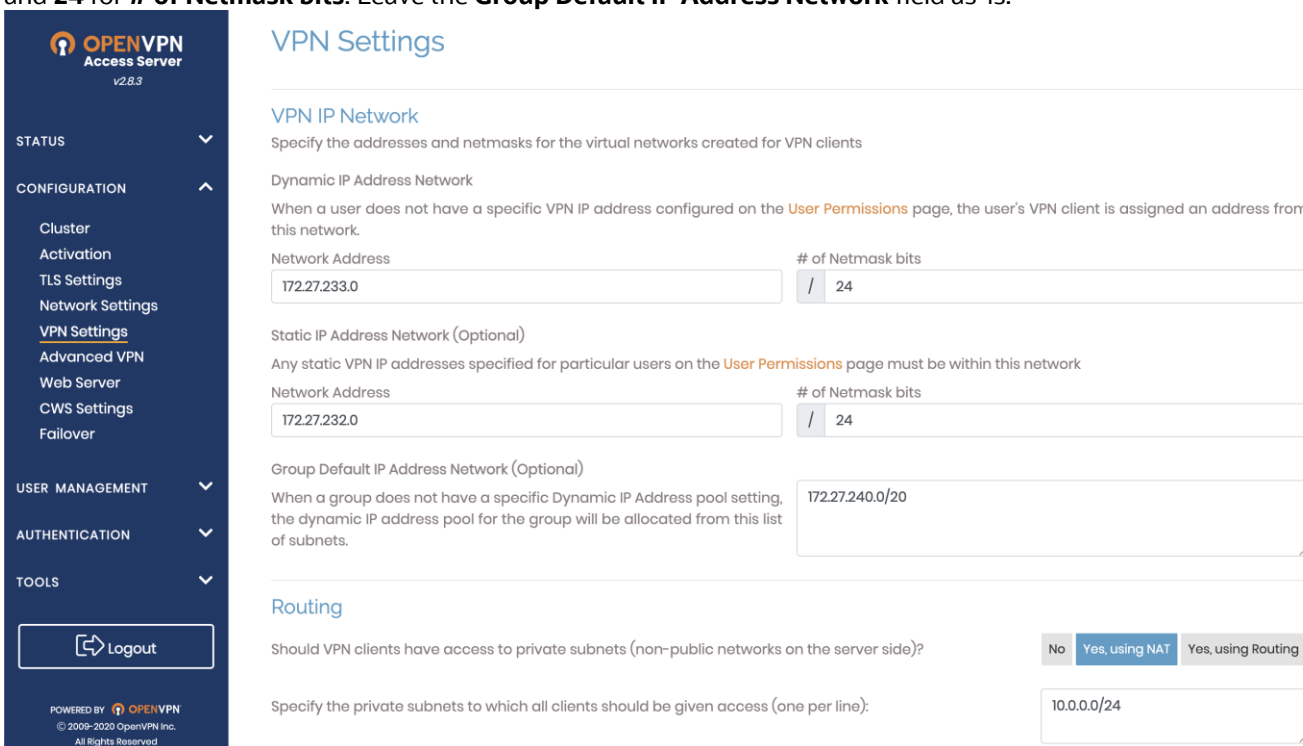
19. Enter the admin credentials that you configured earlier in step 13 to log in.



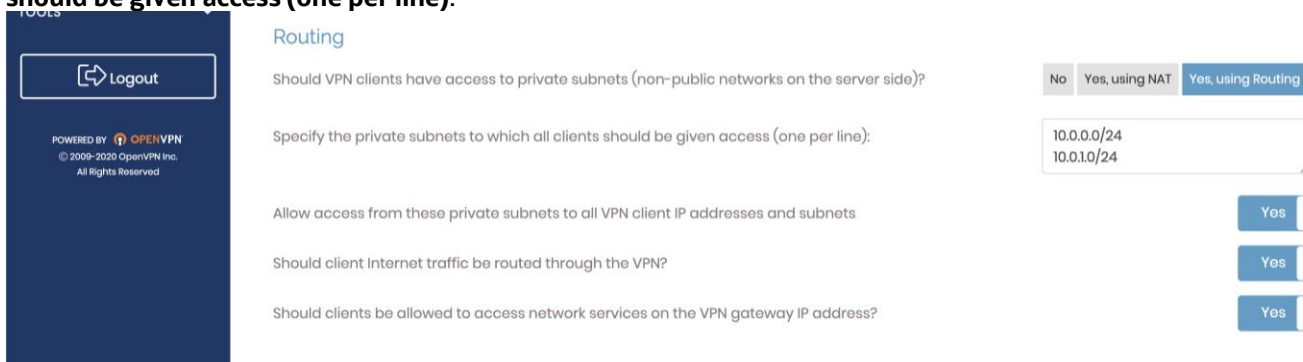
20. After logging in, from the left-hand side menu, select **Configuration** and click **VPN Settings**.



21. On the VPN Settings page, under **Dynamic IP Address Network** - input **172.27.233.0** for **Network Address** and **24** for **# of Netmask bits**. Under **Static IP Address Network**, input **172.27.232.0** for **Network Address** and **24** for **# of Netmask bits**. Leave the **Group Default IP Address Network** field as-is.



22. While on the VPN Settings page, scroll down to **Routing**. Select **Yes, using Routing**, and specify your OCI VCN public and private subnets IPv4 CIDR blocks next to **Specify the private subnets to which all clients should be given access (one per line)**.



Routing

Should VPN clients have access to private subnets (non-public networks on the server side)?  No  Yes, using NAT  Yes, using Routing

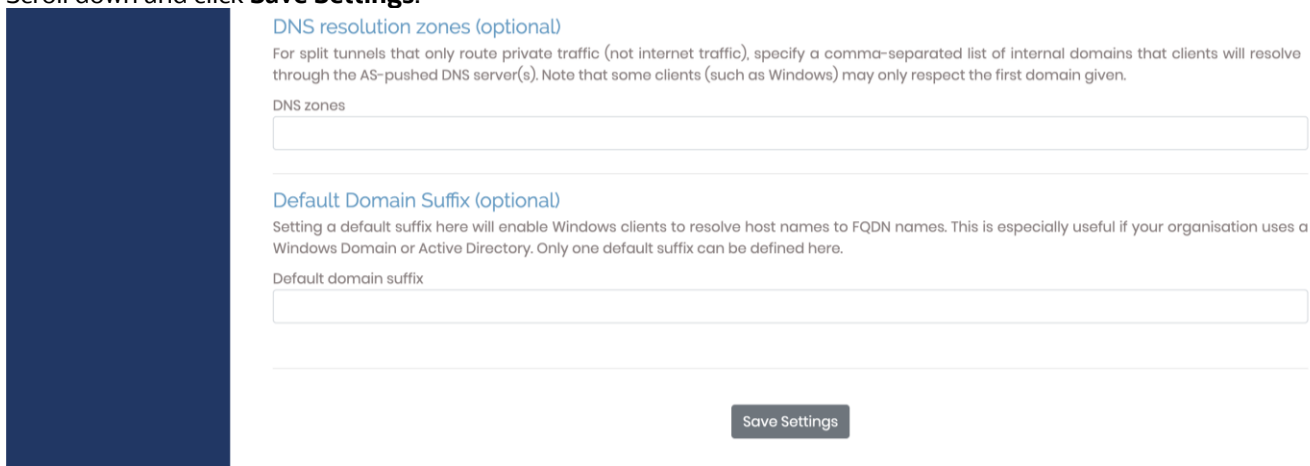
Specify the private subnets to which all clients should be given access (one per line):  
10.0.0.0/24  
10.0.10.0/24

Allow access from these private subnets to all VPN client IP addresses and subnets  Yes

Should client Internet traffic be routed through the VPN?  Yes

Should clients be allowed to access network services on the VPN gateway IP address?  Yes

23. Scroll down and click **Save Settings**.



DNS resolution zones (optional)

For split tunnels that only route private traffic (not internet traffic), specify a comma-separated list of internal domains that clients will resolve through the AS-pushed DNS server(s). Note that some clients (such as Windows) may only respect the first domain given.

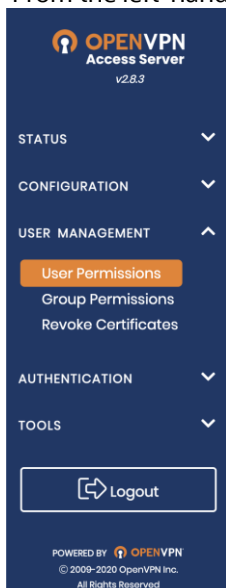
DNS zones

Default Domain Suffix (optional)

Setting a default suffix here will enable Windows clients to resolve host names to FQDN names. This is especially useful if your organisation uses a Windows Domain or Active Directory. Only one default suffix can be defined here.

Default domain suffix

24. From the left-hand OpenVPN Access Server menu, select **USER MANAGEMENT** and click **User Permissions**.



OPENVPN Access Server v2.8.3

STATUS ▾

CONFIGURATION ▾

USER MANAGEMENT ▸

User Permissions

Group Permissions

Revoke Certificates

AUTHENTICATION ▾

TOOLS ▾

POWERED BY OPENVPN  
© 2009-2020 OpenVPN Inc.  
All Rights Reserved

25. Enter a username in the **New Username** field and click the **More Settings** icon in the adjacent column.

**OpenVPN Access Server v2.8.3**

STATUS  
CONFIGURATION  
USER MANAGEMENT  
  **User Permissions**  
  Group Permissions  
  Revoke Certificates  
AUTHENTICATION  
TOOLS  
Logout

POWERED BY OPENVPN  
© 2009-2020 OpenVPN Inc.  
All Rights Reserved

### User Permissions

Search By Username/Group (use '%' as wildcard)

No Default Group

Username	Group	More Settings	Admin	Allow Auto-login	Deny Access	Delete
openvpn	No Default Group		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
root	No Default Group		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="New Username"/>	No Default Group		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Require user permissions record for VPN access

26. Enter a **Password** for the user you created in the previous step. For **Select IP Addressing**, click **Use Static** and specify the IP address to assign to the new user in the **VPN Static IP Address** field. This IP address must be in the range defined in the **Static IP Address Network** field of the VPN Configuration, see step 21. For this guide, we have chosen **172.27.232.25**. Select **Use Routing** for **Select addressing method** and specify your OCI VCN public and private subnets IPv4 CIDR blocks in the **Allow Access To these Networks** field. For **Allow Access From**, select **all server-side private subnets**. Click **Save Settings**.

Group Permissions  
Revoke Certificates  
AUTHENTICATION  
TOOLS  
Logout

POWERED BY OPENVPN  
© 2009-2020 OpenVPN Inc.  
All Rights Reserved

root No Default Group

No Default Group

#### Local Password

Password:

Allow password change from CWS:  Default  Yes  No

Enable password strength checking in CWS:  Default  Yes  No

#### IP Addressing

Select IP Addressing:  Use Dynamic  Use Static

VPN Static IP Address:

#### Access Control

Select addressing method:  Use NAT  Use Routing

Allow Access To these Networks:

Allow Access From:  all server-side private subnets  
 all other VPN clients

#### VPN Gateway

Configure VPN Gateway:  No  Yes

#### DMZ settings

Configure DMZ IP address:  No  Yes

27. After saving the completed previous step, click **Update Running Server**.

OPENVPN Access Server v2.8.3

STATUS

CONFIGURATION

USER MANAGEMENT

User Permissions

Group Permissions

Revoke Certificates

AUTHENTICATION

TOOLS

Logout

POWERED BY OPENVPN © 2009-2020 OpenVPN Inc. All Rights Reserved

User Permissions Changed

User 'openvpnuser' added.

Default permissions changed (default set to Allow access).

Press the button below to propagate the changes to the running server.

Update Running Server

### User Permissions

Search By Username/Group (use '%' as wildcard)

No Default Group Search/Refresh

Username	Group	More Settings	Admin	Allow Auto-login	Deny Access	Delete
New Username	No Default Group		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Require user permissions record for VPN access  No

28. Log out and log in using the new user credentials that you created in step 26. Remove the `/admin` from the URL when logging in if you did not assign the new user to be an admin.  
<https://<openvpn-access-server-public-ip>/>

OPENVPN Access Server

### Admin Login

Sign In

POWERED BY OPENVPN © 2009-2020 OpenVPN Inc. All Rights Reserved



29. Once logged in as the new user, click **Yourself (user-locked profile)** to download `client.ovpn` profile.



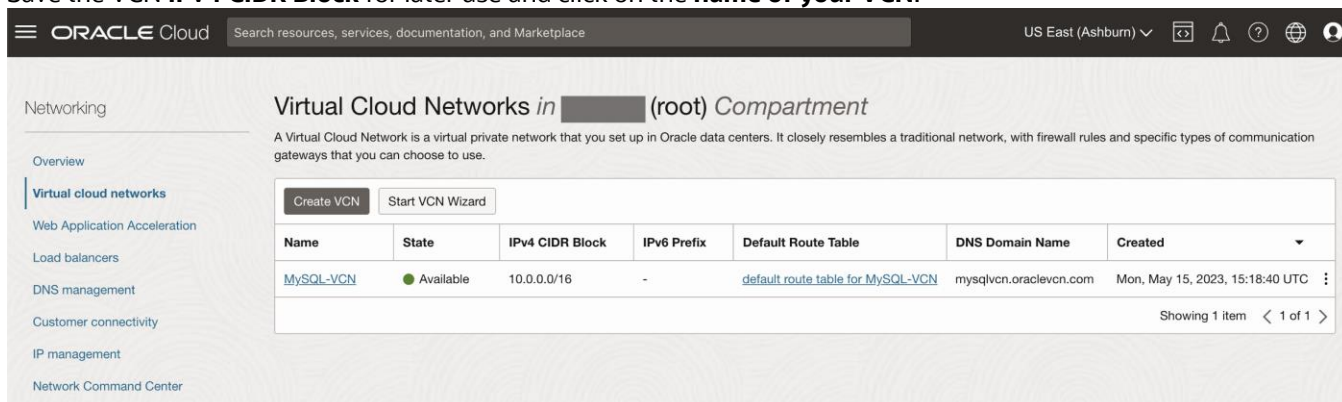
30. Click the appropriate platform icon depending on the Operating System (OS) you are running to download the OpenVPN client. For this guide, we are using macOS. After downloading the client, install it. For more information see, [Installation guide for macOS](#), [Installation guide for Windows](#), and [Connecting to Access Server with Linux](#).

31. After installing the OpenVPN client on your OS, import the `client.ovpn` profile. For more information see, [Import a Profile](#).

32. Once the profile has been imported, **start the OpenVPN Client**. It is now time to configure the OCI VCN to enable communications from the OpenVPN Access Server.

33. Login to [OCI](#) and open the navigation menu. Select **Networking** and click **Virtual Cloud Networks**.

34. Save the VCN **IPv4 CIDR Block** for later use and click on the **name of your VCN**.



35. On the Virtual Cloud Network Details page, click **Route Tables** and click **route table for private subnet-  
<vcn-name>**.

The screenshot shows the Oracle Cloud console interface for a Virtual Cloud Network (VCN) named 'MySQL-VCN'. The page title is 'MySQL-VCN' and it is in an 'AVAILABLE' state. The 'VCN Information' tab is active, displaying details such as Compartment, OCID, Created time, IPv4 CIDR Block, IPv6 Prefix, DNS Resolver, Default Route Table, and DNS Domain Name. Below this, the 'Route Tables in (root) Compartment' section shows a table with two route tables: 'route table for private subnet-MySQL-VCN' and 'default route table for MySQL-VCN'. The left sidebar lists various resources including Subnets, CIDR Blocks, Route Tables, Internet Gateways, Dynamic Routing Gateways, Attachments, and Network Security Groups.

36. Click **Add Route Rules**.

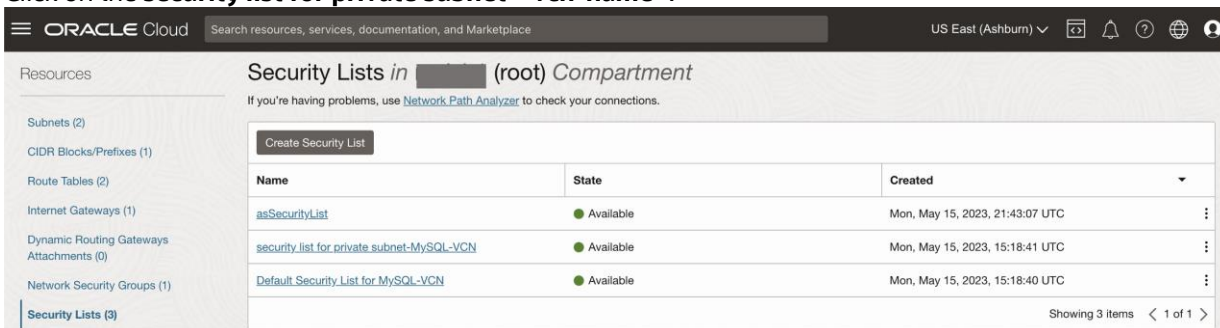
The screenshot shows the Oracle Cloud console interface for a specific route table named 'route table for private subnet-MySQL-VCN'. The page title is 'route table for private subnet-MySQL-VCN' and it is in an 'AVAILABLE' state. The 'Route Table Information' tab is active, displaying details such as OCID, Compartment, and Created time. Below this, the 'Route Rules' section shows a table with two route rules: '0.0.0.0/0' and 'All IAD Services In Oracle Services Network'. The 'Add Route Rules' button is highlighted, indicating the next step in the process. The left sidebar lists various resources including Route Rules.

37. For **Target Type** select **Private IP**. Make sure **CIDR Block** is selected under **Destination Type**. For **Destination CIDR Block**, input the **Static IP Address Network CIDR Block** from step 21 - in our case, it is **172.27.232.0/24**. Under **Target Selection**, enter the **Private IP** of the OpenVPN access server from step 17. Click **Add Route Rules**.

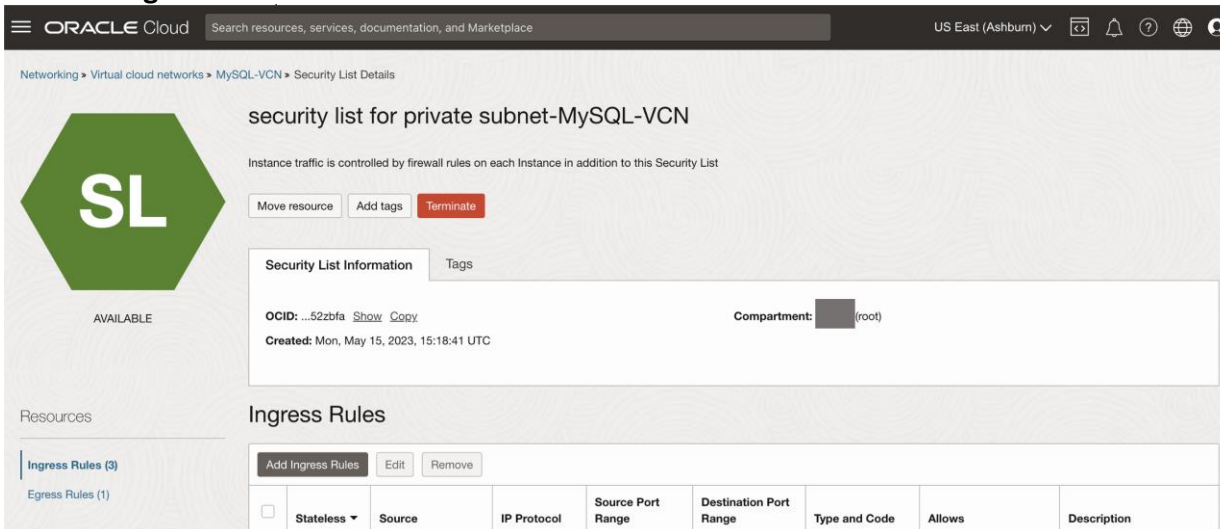
38. Go back to the Virtual Cloud Network Details page of your VCN and click **Security Lists**.

Name	State	Created
<a href="#">asSecurityList</a>	Available	Mon, May 15, 2023, 21:43:07 UTC
<a href="#">security_list_for_private_subnet-MySQL-VCN</a>	Available	Mon, May 15, 2023, 15:18:41 UTC
<a href="#">Default Security List for MySQL-VCN</a>	Available	Mon, May 15, 2023, 15:18:40 UTC

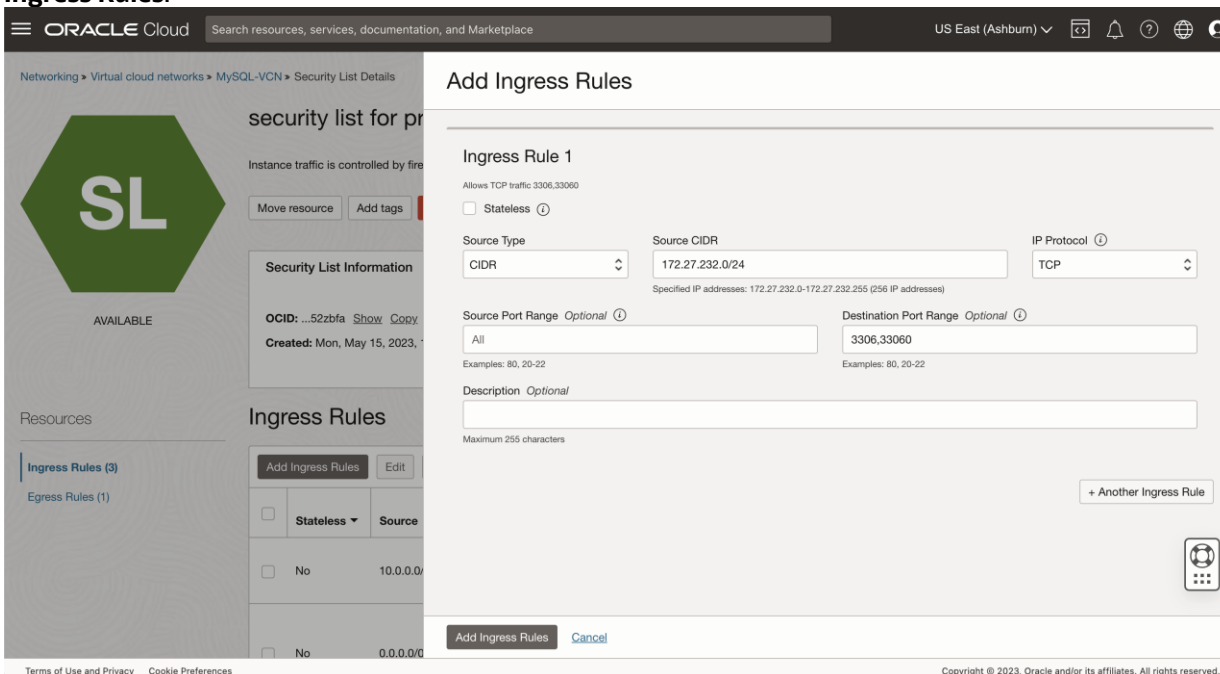
39. Click on the **security list for private subnet-<vcn-name>**.



40. Click **Add Ingress Rules**.



41. For **Source CIDR**, input the **Static IP Address Network CIDR Block** from step 21 - in our case, it is **172.27.232.0/24**. For **Destination Port Range**, specify **3306,33060**. Leave everything as-is and click **Add Ingress Rules**.



42. Stay on the same security list for private subnet-<vcn-name> page and click **Add Ingress Rules** again.

Oracle Cloud console showing the details of a security list named "security list for private subnet-MySQL-VCN". The page includes a sidebar with "Resources" showing "Ingress Rules (3)" and "Egress Rules (1)". The main content area shows "Security List Information" with OCID: ...52zbf and creation date: Mon, May 15, 2023, 15:18:41 UTC. Below this is a table of "Ingress Rules" with columns for checkboxes, Stateless status, Source, IP Protocol, Source Port Range, Destination Port Range, Type and Code, Allows, and Description. The table lists two rules: one for TCP traffic on port 22 and one for ICMP traffic on ports 3 and 4. The "Add Ingress Rules" button is highlighted in the table header.

43. For **Source CIDR**, enter the **IPv4 CIDR Block** of your OCI VCN from step 34. For **Destination Port Range**, specify **3306,33060**. Leave everything as-is and click **Add Ingress Rules**.

Oracle Cloud console showing the "Add Ingress Rules" dialog box. The "Ingress Rule 1" form is filled out with the following details:
 

- Allows TCP traffic 3306,33060
- Stateless
- Source Type: CIDR
- Source CIDR: 10.0.0/16
- IP Protocol: TCP
- Specified IP addresses: 10.0.0.0-10.0.255.255 (65,536 IP addresses)
- Source Port Range: All
- Destination Port Range: 3306,33060
- Description: (empty)

 The "Add Ingress Rules" button is highlighted at the bottom of the dialog.

44. Go back to the **Virtual Cloud Network Details** page of your VCN and click **Security Lists**.

Oracle Cloud console showing the details for a Virtual Cloud Network (VCN) named MySQL-VCN. The page includes a navigation breadcrumb: Networking > Virtual cloud networks > Virtual Cloud Network Details > Security Lists. A green VCN icon is shown with the status 'AVAILABLE'. Below the icon are buttons for 'Move resource', 'Add tags', and 'Delete'. The 'Security Lists' tab is active, displaying a table of security lists in the root compartment.

**VCN Information**

- Compartment: (root)
- Created: Mon, May 15, 2023, 15:18:40 UTC
- IPV4 CIDR Block: 10.0.0.0/16
- IPV6 Prefix: No value
- OCID: ...vux3zq [Show Copy](#)
- DNS Resolver: [MySQL-VCN](#)
- Default Route Table: [default route table for MySQL-VCN](#)
- DNS Domain Name: mysqlvcn.oraclevcn.com

**Security Lists in (root) Compartment**

If you're having problems, use [Network Path Analyzer](#) to check your connections.

Create Security List

Name	State	Created
<a href="#">asSecurityList</a>	Available	Mon, May 15, 2023, 21:43:07 UTC
<a href="#">security list for private subnet-MySQL-VCN</a>	Available	Mon, May 15, 2023, 15:18:41 UTC
<a href="#">Default Security List for MySQL-VCN</a>	Available	Mon, May 15, 2023, 15:18:40 UTC

Showing 3 items < 1 of 1 >

45. Click on **Default Security List for <vcn-name>**.

Oracle Cloud console showing the details for the 'Default Security List for MySQL-VCN'. The page includes a navigation breadcrumb: Security Lists in (root) Compartment. A table of security lists is displayed, showing the selected list.

**Security Lists in (root) Compartment**

If you're having problems, use [Network Path Analyzer](#) to check your connections.

Create Security List

Name	State	Created
<a href="#">asSecurityList</a>	Available	Mon, May 15, 2023, 21:43:07 UTC
<a href="#">security list for private subnet-MySQL-VCN</a>	Available	Mon, May 15, 2023, 15:18:41 UTC
<a href="#">Default Security List for MySQL-VCN</a>	Available	Mon, May 15, 2023, 15:18:40 UTC

Showing 3 items < 1 of 1 >

46. Click **Add Ingress Rules**.

Networking > Virtual cloud networks > MySQL-VCN > Security List Details

### Default Security List for MySQL-VCN

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: ...cw33fa Show Copy **Compartment:** (root)

Created: Mon, May 15, 2023, 15:18:40 UTC

Resources

**Ingress Rules**

Add Ingress Rules Edit Remove

<input type="checkbox"/>	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	SSH Remote Login Protocol	TCP traffic for ports: 22	SSH Remote Login Protocol

47. For **Source CIDR**, enter the **IPv4 CIDR Block** of your OCI VCN from step 34. For **Destination Port Range**, specify **3306,33060**. Leave everything as-is and click **Add Ingress Rules**.

Networking > Virtual cloud networks > MySQL-VCN > Security List Details

### Default Security List

Instance traffic is controlled by fire

Move resource Add tags

**Security List Information**

OCID: ...cw33fa Show Copy

Created: Mon, May 15, 2023,

Resources

**Ingress Rules**

Add Ingress Rules Edit

### Add Ingress Rules

**Ingress Rule 1**

Allows TCP traffic 3306,33060

Stateless

Source Type: CIDR Source CIDR: 10.0.0.0/16 IP Protocol: TCP

Specified IP addresses: 10.0.0.0-10.0.255.255 (65,536 IP addresses)

Source Port Range: All Destination Port Range: 3306,33060

Examples: 80, 20-22 Examples: 80, 20-22

Description: Optional

Maximum 255 characters

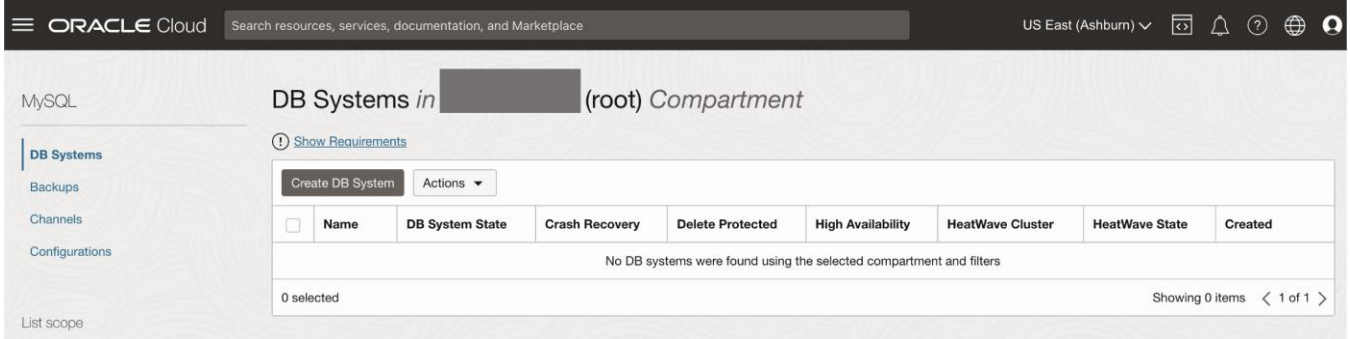
+ Another Ingress Rule

Add Ingress Rules Cancel

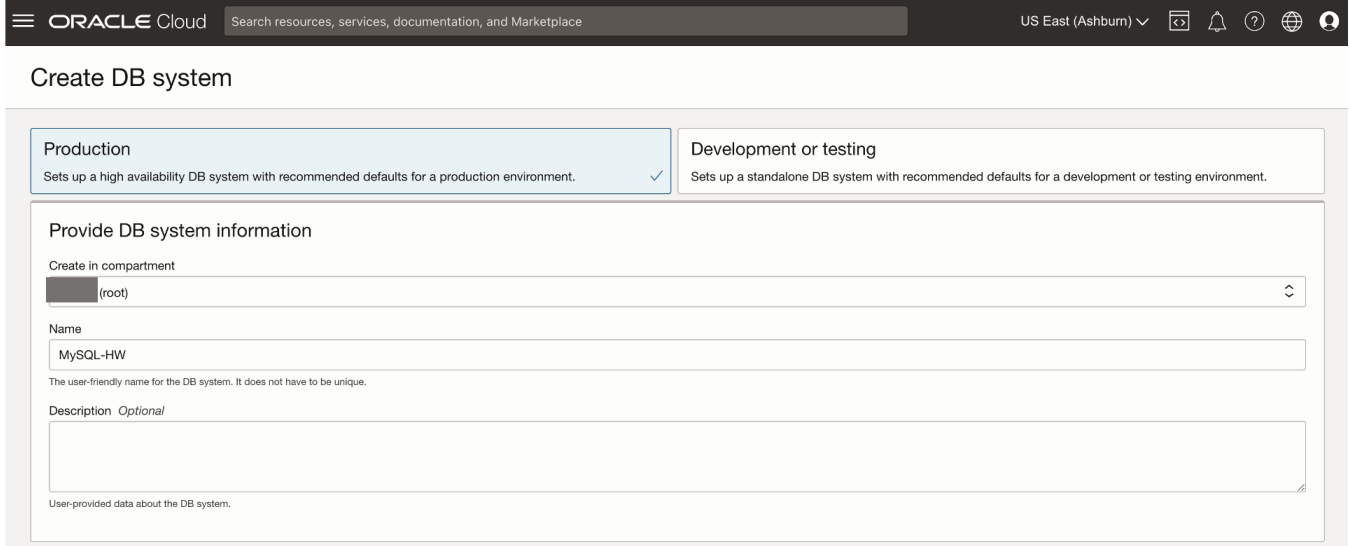
48. The VPN connection from on-premises to OCI is now set up. Make sure the OpenVPN client is started/running. We are now ready to perform the migration.

### III) On OCI, create a HeatWave MySQL instance.

49. From the OCI Console, click on the navigation menu, click **Databases**, and click **HeatWave MySQL**.



50. Pick **Production** or **Development or testing** and enter a **MySQL DB system name**.





- Select **Standalone** or **High Availability**. Turn **ON** the button for HeatWave MySQL - if you want to run OLTP, OLAP, and ML workloads. Afterwards, create your **Administrator credentials** that will be used to manage the HeatWave MySQL database.

The screenshot shows the 'Create DB System' page in the Oracle Cloud console. At the top, there are navigation elements including the Oracle Cloud logo, a search bar, and the region 'US East (Ashburn)'. The main content area is titled 'Create DB System' and contains several sections:

- Standalone**: A selected option for a 'Single-instance DB system' with a checkmark.
- High availability**: An option for a 'DB system with 3 MySQL instances providing automatic failover and zero data loss'.
- Configure MySQL HeatWave**: A section with a toggle switch for 'MySQL HeatWave' which is turned on. Below it, text reads: 'Show shapes and configurations that support HeatWave for accelerated query processing, which is suitable for running both OLTP and OLAP workloads. The default data storage size is 1,024 GB.'
- Create administrator credentials**: A section with three input fields: 'Username' (containing 'admin'), 'Password' (masked with dots), and 'Confirm password' (masked with dots).
- Configure networking**: A section that is partially visible at the bottom of the screenshot.

At the bottom of the page, there are buttons for 'Create', 'Save as stack', and 'Cancel', along with a 'Collapse' link for the networking section.

- For **Configuring Networking** - choose the earlier created VCN and make sure the **Private Subnet** is selected under **Subnet in <compartment-name>**. For **Configure Placement** leave it as-is.

The screenshot shows the 'Create DB system' page in the Oracle Cloud console, focusing on the 'Configure networking' and 'Configure placement' sections.

- Configure networking**: This section explains that the VCN and subnet where the DB system endpoint will be attached. It provides a link for 'How do I connect to a DB system?' and a link to 'create a VCN'. The 'Virtual cloud network in' dropdown is set to 'MySQL-VCN'. The 'Subnet in' dropdown is set to 'private subnet-MySQL-VCN (Regional)'. There is a 'Collapse' link on the right.
- Configure placement**: This section explains that the availability domain/fault domain in which the DB system endpoint will be physically placed. It is recommended to allow Oracle to choose the best placement for the fault domain. There are three availability domain options:
  - AD-1**: QDIL:US-ASHBURN-AD-1 (Selected with a checkmark)
  - AD-2**: QDIL:US-ASHBURN-AD-2
  - AD-3**: QDIL:US-ASHBURN-AD-3
 Below these options, there is a checkbox for 'Choose a fault domain' which is unchecked. A note states: 'If you do not select a fault domain, Oracle will choose the best placement for you.' There is a 'Collapse' link on the right.

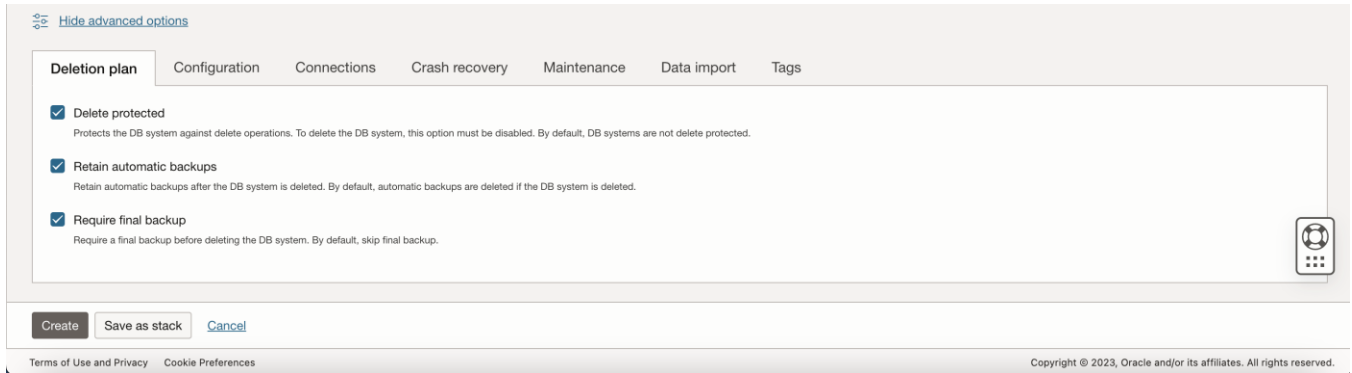
At the bottom of the page, there are buttons for 'Create', 'Save as stack', and 'Cancel', along with a 'Collapse' link for the placement section.

53. **Configure hardware** (OCPU and Memory) for MySQL by choosing an appropriate DB Shape. For this guide, we will use the default HeatWave shape. For the **Data Storage Size** be sure to make the size large enough for future growth.

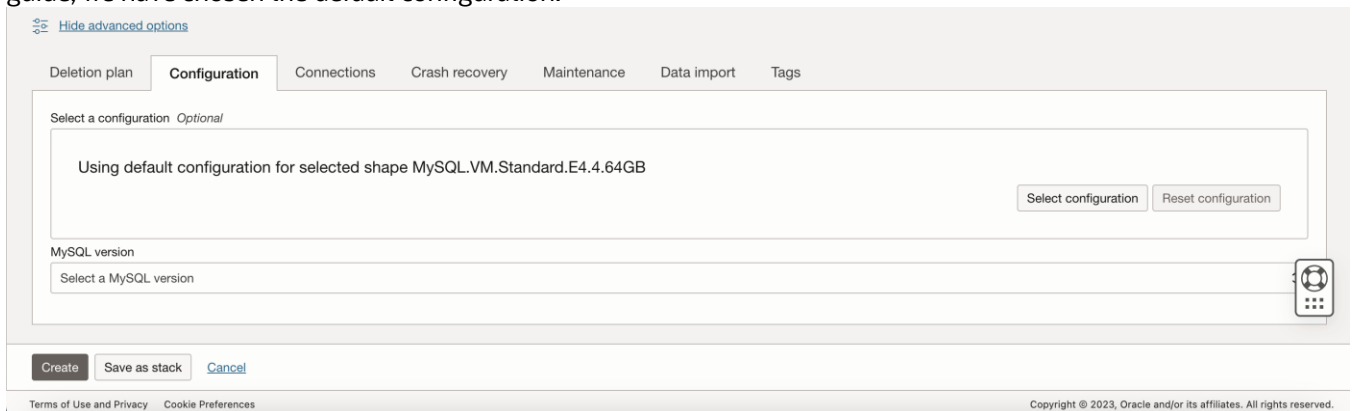
The screenshot shows the Oracle Cloud console interface for creating a database system. The page title is "Create DB system". The "Configure hardware" section is expanded, showing the selected shape: "MySQL.HeatWave.VM.Standard". The shape details are: CPU core count: 16, Memory size: 512 GB, and Max network bandwidth: 16Gbps. Below this, the "Data storage size (GB)" is set to 1024. The resulting performance metrics are: Total IOPS: 76800 and Total throughput: 600 MB. A "Change shape" button is visible on the right. At the bottom of the section, there are "Create", "Save as stack", and "Cancel" buttons. The footer includes "Terms of Use and Privacy", "Cookie Preferences", and "Copyright © 2023, Oracle and/or its affiliates. All rights reserved."

54. **Configure a backup plan** according to what suits your needs. Lastly, scroll down until you see **Show advanced options**. Click on it to expand.

The screenshot shows the Oracle Cloud console interface for creating a database system, specifically the "Configure backup plan" section. The "Data storage size (GB)" is still set to 1024. The "Configure backup plan" section has the following settings: "Enable automatic backups" is checked, with a note that a retention period and backup window must also be specified. The "Backup retention period" is set to 7 days. "Enable point in time restore" is checked, with a note that it enables restoration from a point in time. "Select backup window" is unchecked. At the bottom of the section, there is a link for "Show advanced options". Below the backup plan section, there are "Create", "Save as stack", and "Cancel" buttons. The footer includes "Terms of Use and Privacy", "Cookie Preferences", and "Copyright © 2023, Oracle and/or its affiliates. All rights reserved."



55. From the advanced options screen, go to the **Configuration** tab. If you have a custom configuration that you would like to apply to your HeatWave MySQL instance - you can do so by clicking **Select configuration**. Custom configurations allow you to tweak MySQL variables (i.e., max connections, binary log expire seconds, etc.) rather than using the default values. You must create a custom configuration in advance before applying. For more information regarding custom configurations, see [Configuration of a DB System](#). For this guide, we have chosen the default configuration.



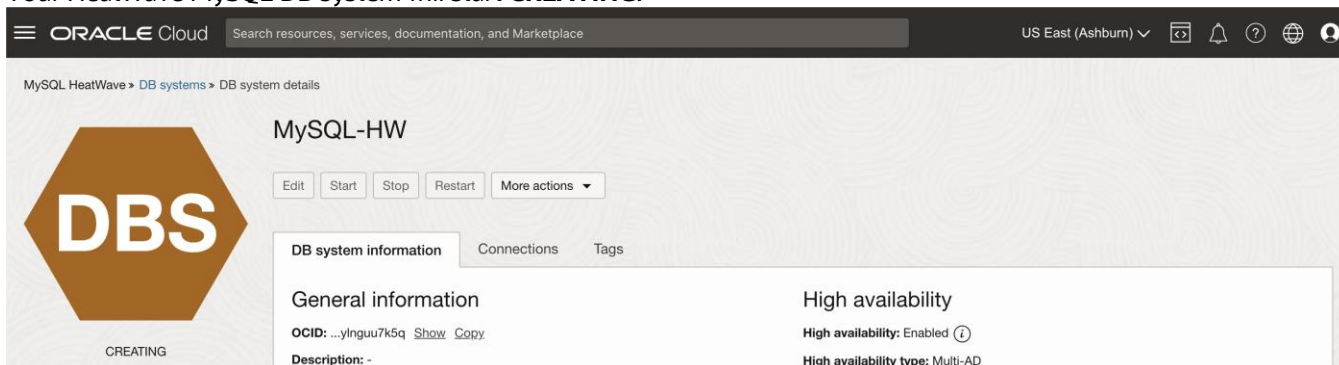
56. For **MySQL version**, choose either **Innovation** or **Bug fix**. With the new MySQL versioning model, you have the flexibility to select an innovation or a bug fix release. Both releases are production-grade quality. MySQL innovation releases allow you to access the latest features and improvements. Innovation releases are ideal for fast-paced development environments with high levels of automated tests and modern continuous integration techniques for faster upgrade cycles. MySQL bug fix releases (aka long-term support releases) allow you to reduce the risks associated with changes in the database software behavior, as these releases only contain necessary fixes (bugfix and security patches). For more information regarding MySQL innovation and bug fix releases, see [Introducing MySQL Innovation and Bug fix versions](#). For this guide, we have chosen **8.0.34 - Bug fix**.

The screenshot shows the 'Configuration' tab of a MySQL database creation wizard. The 'MySQL version' dropdown menu is open, displaying a list of versions: 8.0.34 - Bug fix (highlighted), 8.0.33 - Bug fix (Deprecated), 8.0.32 - Bug fix (Deprecated), 8.0.31 - Bug fix (Deprecated), 8.0.30 - Bug fix (Deprecated), and 8.1.0 - Innovation. The 'Create' button is visible at the bottom left of the wizard.

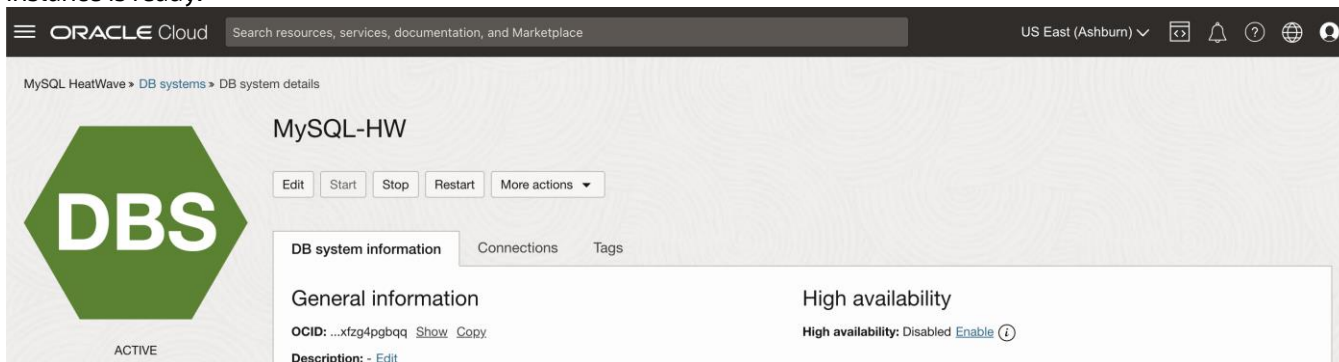
57. Click **Create** to finish the HeatWave MySQL DB system creation process.

This screenshot is identical to the previous one, but the 'Create' button at the bottom left of the wizard is highlighted in a darker shade, indicating it is the next step in the process.

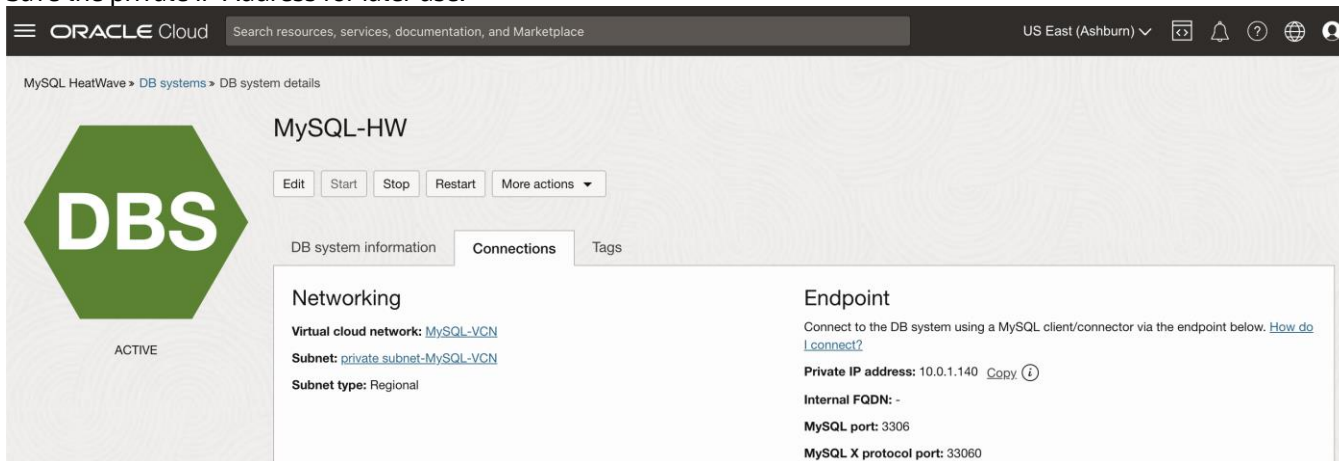
58. Your HeatWave MySQL DB system will start **CREATING**.



59. Within a few minutes, HeatWave MySQL DB system will change its state from CREATING to **ACTIVE** once the instance is ready.



60. On the same DB system details page, click **Connections** to grab the **private IP address** for HeatWave MySQL. Save the private IP Address for later use.



Note: you can navigate to the **DB System Details** page by going to the Navigation menu in OCI. Click **Databases** and click **HeatWave MySQL**. Click on the name of your MySQL DB System to open the **DB System Details** page.

#### IV) Install MySQL Shell 8.2.1 (or above) on an on-premises instance that can connect to your MariaDB.

### MySQL Community Downloads

MySQL Shell

The screenshot shows the MySQL Community Downloads page for MySQL Shell 8.2.1 Innovation. The page has three tabs: "General Availability (GA) Releases", "Archives", and a download icon. The "General Availability (GA) Releases" tab is selected. The main heading is "MySQL Shell 8.2.1 Innovation". Below the heading are three dropdown menus: "Select Version:" with "8.2.1 Innovation" selected, "Select Operating System:" with "Red Hat Enterprise Linux / Oracle Linux" selected, and "Select OS Version:" with "Red Hat Enterprise Linux 8 / Oracle Linux 8 (x86, 64-bit)" selected. Below the dropdowns is a table of RPM packages:

RPM Package	Version	Size	Download
(mysql-shell-8.2.1-1.el8.x86_64.rpm)	8.2.1	30.1M	<a href="#">Download</a>
(mysql-shell-debuginfo-8.2.1-1.el8.x86_64.rpm)	8.2.1	496.0M	<a href="#">Download</a>

MD5: d646ac389259f563c2a67d9365182360

MD5: 2131793913ee046f91d4e41926fe4b9e

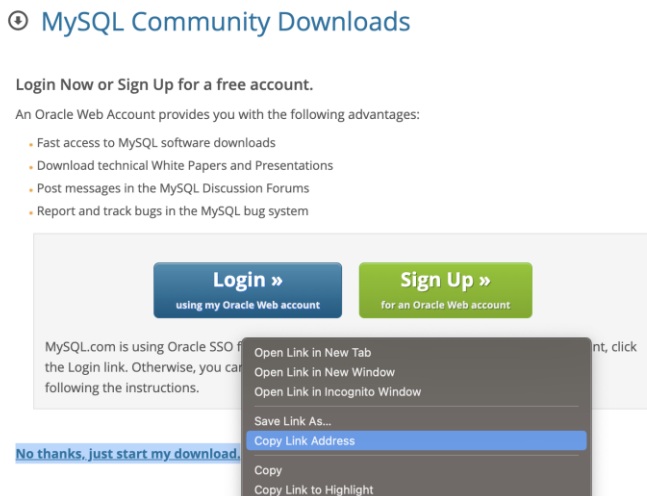
We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

61. Have an on-premises instance that can connect to your MariaDB. Go to the below website and download MySQL Shell **8.2.1** on your on-premises instance. For this guide, we have deployed MySQL Shell on a Linux instance. From the MySQL Shell download page, ensure **8.2.x Innovation** is selected under **Select Version**. MySQL Shell 8.2 is fully compatible with MySQL 8.2, 8.1, 8.0, and 5.7. For **Operating System** and **OS Version** - pick the appropriate option depending on the OS and the OS Version that you are running. Click **Download**.

<https://dev.mysql.com/downloads/shell/>

Note: for this guide, we will show you how to install MySQL Shell on a Linux environment. For other environments, see [Installing MySQL Shell on Windows](#), [Installing MySQL Shell on Linux](#), and [Installing MySQL Shell on macOS](#).

62. Right-click on **No thanks, just start my download** and click **Copy link address**.



63. Go back to the on-premises instance that can connect to your on-premises MariaDB and execute the below command to install MySQL Shell:

```
$ sudo yum install -y <MySQL-Shell-Download-Link>
```

Replace the link with what you have.

```
$ sudo yum install https://dev.mysql.com/get/Downloads/MySQL-Shell/mysql-shell-8.2.1-1.el8.x86\_64.rpm
```

```
[opc@linux-8 ~]$ sudo yum install https://dev.mysql.com/get/Downloads/MySQL-Shell/mysql-shell-8.2.1-1.el8.x86_64.rpm
Last metadata expiration check: 1:10:27 ago on Tue 21 Nov 2023 06:09:14 PM GMT.
mysql-shell-8.2.1-1.el8.x86_64.rpm                88 MB/s | 30 MB      00:00
Dependencies resolved.
=====
Package                Arch    Version                               Repository    Size
=====
Installing:
mysql-shell             x86_64  8.2.1-1.el8                           @commandline 30 M
Installing dependencies:
python39-libs          x86_64  3.9.18-1.module+el8.9.0+90071+8dc52a4f ol8_appstream 8.2 M
python39-pip-wheel     noarch  20.2.4-8.module+el8.9.0+90016+9c2d6573 ol8_appstream 1.1 M
python39-setuptools-wheel noarch  50.3.2-4.module+el8.9.0+90016+9c2d6573 ol8_appstream 497 k
Installing weak dependencies:
python39               x86_64  3.9.18-1.module+el8.9.0+90071+8dc52a4f ol8_appstream 33 k
python39-pip           noarch  20.2.4-8.module+el8.9.0+90016+9c2d6573 ol8_appstream 1.9 M
python39-setuptools    noarch  50.3.2-4.module+el8.9.0+90016+9c2d6573 ol8_appstream 871 k

Transaction Summary
=====
Install 7 Packages

Total size: 43 M
Total download size: 12 M
Installed size: 261 M
```

64. You can now verify if MySQL Shell has successfully installed on your on-premises instance by executing the below command:

```
$ mysqlsh --version
```

```
[opc@linux-8 ~]$ mysqlsh --version
mysqlsh Ver 8.2.1 for Linux on x86_64 - for MySQL 8.2.0 (MySQL Community Server (GPL))
[opc@linux-8 ~]$
```

65. To login to your MariaDB using MySQL Shell, use the below commands:

```
$ mysqlsh mysql://<user>@<hostname>[:<port-number>]
```

-OR-

```
$ mysqlsh -u <user> -p -h <hostname> [-P <port-number>]
```

```
[opc@linux-8 ~]$ mysqlsh mysql://root@127.0.0.1
Please provide the password for 'root@127.0.0.1': *****
Save password for 'root@127.0.0.1'? [Y]es/[N]o/[e]ver (default No): yes
MySQL Shell 8.2.1

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '? for help; '\quit' to exit.
Creating a Classic session to 'root@127.0.0.1'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 61
Server version: 11.1.2-MariaDB-log MariaDB Server
No default schema selected; type \use <schema> to set one.
MySQL 127.0.0.1:3306 JS >
```

Note: you can interact with MySQL Shell using JavaScript, Python, or SQL mode. The default is JavaScript. To switch between the different modes, execute `/js` for JavaScript, `/py` for Python, and `/sql` for SQL mode inside MySQL Shell. To exit out of MySQL Shell, execute `/q`.

## V) Connect to the MariaDB using MySQL Shell. And perform some compatibility checks.

### Storage Engines

66. MariaDB Community Edition contains various storage engines in alpha or beta stages. These engines are not included in MariaDB Enterprise Edition and are also not supported in MySQL. In MySQL, InnoDB is the main engine used for transaction processing. Before migration, you will need to convert data to InnoDB.

List all the Storage Engines used in your MariaDB instance:

```
MySQL> SELECT COUNT(*) as '# TABLES',
CONCAT(ROUND(sum(data_length) / ( 1024 * 1024 * 1024 ), 2), 'G') DATA,
CONCAT(ROUND(sum(index_length) / ( 1024 * 1024 * 1024 ), 2), 'G') INDEXES,
CONCAT(cast(sum(ROUND(( data_length + index_length ) / ( 1024 * 1024 * 1024 ), 2))
as decimal(5,3), 'G') 'TOTAL SIZE',
ENGINE FROM information_schema.TABLES
WHERE TABLE_SCHEMA
NOT IN ('mysql', 'information_schema', 'performance_schema', 'sys')
GROUP BY engine;
```



```

MySQL 127.0.0.1:3306 JS > \sql
Switching to SQL mode... Commands end with ;
MySQL 127.0.0.1:3306 SQL > SELECT COUNT(*) as '# TABLES',
-> CONCAT(ROUND(sum(data_length) / ( 1024 * 1024 * 1024 ), 2), 'G') DATA,
-> CONCAT(ROUND(sum(index_length) / ( 1024 * 1024 * 1024 ), 2), 'G') INDEXES,
-> CONCAT(cast(sum(ROUND(( data_length + index_length ) / ( 1024 * 1024 * 1024 ),
-> 2)) as decimal(5,2)), 'G') 'TOTAL SIZE',
-> ENGINE FROM information_schema.TABLES
-> WHERE TABLE_SCHEMA
-> NOT IN ('mysql', 'information_schema', 'performance_schema', 'sys') GROUP BY engine;
+-----+-----+-----+-----+-----+
| # TABLES | DATA | INDEXES | TOTAL SIZE | ENGINE |
+-----+-----+-----+-----+-----+
|          3 | 0.00G | 0.00G   | 0.00G      | InnoDB |
+-----+-----+-----+-----+-----+
1 row in set (0.0009 sec)

```

In this example, it's fine as only InnoDB tables are present. If there was another line with another engine, those tables would need to be converted to InnoDB.

### Functions

67. MariaDB also differs from MySQL by some different functions. For example, in MariaDB you have `JSON_DETAILED` which is called `JSON_PRETTY` in MySQL.

This is not a blocking factor unless those functions are present in the default value of a column. If the application uses some of these functions, it may be necessary to make some modifications to use the appropriate one in MySQL.

### Data Types

68. MySQL and MariaDB have some different data types.

For example, MariaDB supports `INET6` as a data type and in MySQL, IPv6 values are stored into `VARBINARY(16)`.

But on the other hand, MySQL supports `JSON` data type that in MariaDB are stored as `LONGTEXT`.

To list all data types used in your database, you can execute the following query:

```

MySQL> SELECT DATA_TYPE , count(*) TOT FROM information_schema.COLUMNS
        WHERE TABLE_SCHEMA NOT
        IN ('mysql', 'sys', 'information_schema', 'performance_schema')
        GROUP BY 1;

```

```

MySQL 127.0.0.1:3306 SQL > SELECT DATA_TYPE , count(*) TOT FROM information_schema.COLUMNS
-> WHERE TABLE_SCHEMA NOT
-> IN ('mysql', 'sys', 'information_schema', 'performance_schema') GROUP BY 1;
+-----+-----+
| DATA_TYPE | TOT |
+-----+-----+
| char       | 12 |
| decimal    | 5  |
| enum       | 2  |
| int        | 4  |
| smallint   | 1  |
+-----+-----+
5 rows in set (0.0019 sec)

```

If you find some data types that are not present in MySQL, you need to convert them. See: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

**V) Connect to MariaDB using MySQL Shell. Afterwards, execute the MySQL Shell `util.copyInstance()` utility to export all schemas (including users, indexes, routines, triggers) from the MariaDB instance to HeatWave MySQL on OCI.**

69. Before connecting to MariaDB using MySQL Shell and proceeding with the below steps, it is highly recommended that you use a command like **screen** or **tmux**. These commands will allow you to reconnect to a dropped session in case your connection drops in the middle of performing the MySQL Shell export using `util.copyInstance()`. For small databases, the screen or tmux may not be necessary. For this guide, we will use tmux. To learn more about tmux, see [A beginner's guide to tmux](#). Below are the basics of using the tmux command:

- Install tmux on Linux: `$ sudo yum install tmux`
- Start a new tmux session, from your terminal execute: `$ tmux`
- List all the active tmux sessions: `$ tmux ls`
- Detach from a tmux session and leave it running in the background: `$ Ctrl+B d`
- Attach a tmux session running in the background: `$ tmux attach`
- End a tmux session: `$ Ctrl+B &`

70. Start a tmux session and connect to your on-premises MariaDB using MySQL Shell.

```
$ tmux
$ mysqlsh mysql://<user>@<hostname>[:<port-number>]
-OR-
$ mysqlsh -u <user> -p -h <hostname> [-P <port-number>]
```

```
[opc@linux-8 ~]$ tmux
[opc@linux-8 ~]$ mysqlsh mysql://root@localhost
Please provide the password for 'root@localhost': *****
Save password for 'root@localhost'? [Y]es/[N]o/[e]ver (default No): yes
MySQL Shell 8.2.0

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '? for help; '\quit' to exit.
Creating a Classic session to 'root@localhost'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 23
Server version: 11.1.2-MariaDB MariaDB Server
No default schema selected; type \use <schema> to set one.
MySQL localhost JS >
```

71. Change to the JavaScript mode of MySQL Shell if needed, and run the `util.copyInstance()` utility to export all MariaDB data into OCI HeatWave MySQL.

```
MySQL JS> \js
MySQL JS> util.copyInstance('mysql://admin@10.0.1.105', {"compatibility":
["force_innodb", "skip_invalid_accounts", "strip_definers",
"strip_restricted_grants", "strip_tablespaces", "ignore_wildcard_grants",
"strip_invalid_grants", "create_invisible_pks"], users: "false", threads: 4,
ignoreVersion: "true", dryRun:"true"})
```

Note: replace the username (`admin`) and IP address (`10.0.1.105`) with your HeatWave MySQL username and IP address (not the on-premises MariaDB username and IP address).

Note: Migrating MariaDB users is not a supported operation.

```
MySQL localhost JS > util.copyInstance('mysql://admin@10.0.1.105', {"compatibility": ["force_inno
db", "skip_invalid_accounts", "strip_definers", "strip_restricted_grants", "strip_tablespaces", "ign
ore_wildcard_grants", "strip_invalid_grants", "create_invisible_pks"], users: "false", threads: 4, i
gnoreVersion: "true", dryRun:"true"})
Copying DDL and Data from in-memory FS, source: linux-8:3306, target: s1cbj26efjaq6cmx:3306.
SRC: dryRun enabled, no locks will be acquired and no files will be created.
NOTE: SRC: Backup lock is not supported in MySQL 5.6 and DDL changes will not be blocked. The dump m
ay fail with an error if schema changes are made while dumping.
SRC: Acquiring global read lock
SRC: Global read lock acquired
Initializing - done
WARNING: SRC: Failed to fetch value of @@GLOBAL.GTID_EXECUTED.
SRC: 2 out of 6 schemas will be dumped and within them 3 tables, 0 views.
Gathering information - done
SRC: All transactions have been started
SRC: Global read lock has been released
NOTE: SRC: When migrating to MySQL HeatWave Service, please always use the latest available version
of MySQL Shell.
SRC: Checking for compatibility with MySQL HeatWave Service 8.0.35
NOTE: SRC: MySQL Server 5.6 detected, please consider upgrading to 8.0 first.
SRC: Compatibility checks finished.ompatibility \ 0 / 5
Validating MySQL HeatWave Service compatibility - done
SRC: Writing global DDL files
```

[... output truncated]

```
Writing schema metadata - done
Writing DDL - done
Writing table metadata - done
SRC: Starting data dump
0% (0 rows / ~5.27K rows), 0.00 rows/s, 0.00 B/s
TGT: Executing common postamble SQL
?% (0 bytes / ?), 0.00 B/s, 3 / 3 tables done
Recreating indexes - done
TGT: No data loaded.
TGT: 0 warnings were reported during the load.
---
Dump_metadata:
  Binlog_file: mysql-bin.000001
  Binlog_position: 328
  Executed_GTID_set: ''
```

Note:

- `util.copyInstance(connectionData[, options])`: MySQL instance copy utility enables copying of an entire instance to another server. By default, this utility includes all schemas, users, indexes, routines, and triggers. See [Copy Utilities](#).
  - `connectionData`: Defines the connection details for the destination server you want to copy to.
- `compatibility`: Apply the specified requirements for compatibility with HeatWave MySQL for all tables in the dump output, altering the dump files as necessary.
  - `force_innodb`: Change `CREATE TABLE` statements to use the InnoDB storage engine for any tables that do not already use it.
  - `skip_invalid_accounts`: You cannot export a user that has no password defined. This option skips any such users.
  - `strip_definers`: Remove the `DEFINER` clause from views, routines, events, and triggers, so these objects are created with the default definer (the user invoking the schema), and change the `SQL SECURITY` clause for views and routines to specify `INVOKER` instead of `DEFINER`. HeatWave MySQL requires special privileges to create these objects with a definer other than the user loading the schema. If your security model requires that views and routines have more privileges than the account querying or calling them, you must manually modify the schema before loading it.
  - `strip_restricted_grants`: Certain privileges are restricted in HeatWave MySQL. Privileges such as `RELOAD`, `FILE`, `SUPER`, `BINLOG_ADMIN`, and `SET_USER_ID`. You cannot create users granting these privileges. This option strips these privileges from dumped `GRANT` statements.
  - `strip_tablespaces`: Tablespaces have some restrictions in HeatWave MySQL. If you need tables created in their default tablespaces, this option strips the `TABLESPACE=` option from `CREATE TABLE` statements.
  - `ignore_wildcard_grants`: If enabled, ignores errors from grants on schemas with wildcards, which are interpreted differently in systems where the `partial_revokes` system variable is enabled.
  - `strip_invalid_grants`: If enabled, strips grant statements which would fail when users are copied. Such as grants referring to a specific routine that does not exist.
  - `create_invisible_pks`: Primary keys are required by High Availability and HeatWave. If you intend to export data for use in a highly available DB system or a HeatWave DB system, add primary keys as they are not defined on the tables. This compatibility flag adds invisible primary keys to each table that requires them.
- `users`: Include (`true`) or exclude (`false`) users and their roles and grants in the dump.
- `threads`: (Optional) The number of parallel threads to use to copy chunks of data from the MySQL instance. Each thread has its own connection to the MySQL instance. The default is 4. The copy utilities require twice the number of threads, one thread to copy and one thread to write. If `threads` is set to `N`, `2N` threads are used.
- `dryRun`: Displays information about the copy with the specified set of options, and about the results of HeatWave MySQL Service compatibility checks, but does not proceed with the copy. Setting this option enables you to list out all of the compatibility issues before starting the copy.

72. Once you have run the command in step 71 and did not see any errors in the output (warnings are okay), run the same step 71 command but this time change the `dryRun` option to `false`.

```
MySQL JS> util.copyInstance('mysql://admin@10.0.1.105', {"compatibility":
["force_innodb", "skip_invalid_accounts", "strip_definers",
"strip_restricted_grants", "strip_tablespaces", "ignore_wildcard_grants",
"strip_invalid_grants", "create_invisible_pks"], users: "false", threads: 4,
ignoreVersion: "true", dryRun:"false"})
```

Note: replace the username (`admin`) and IP address (`10.0.1.105`) with your HeatWave MySQL username and IP address (not the on-premises MariaDB username and IP address).

```
MySQL localhost JS > util.copyInstance('mysql://admin@10.0.1.105', {"compatibility": ["for
ce_innodb","skip_invalid_accounts","strip_definers","strip_restricted_grants", "strip_tablesp
aces", "ignore_wildcard_grants", "strip_invalid_grants", "create_invisible_pks"], users: "fal
se", threads: 4, ignoreVersion: "true", dryRun: "false"})
Copying DDL and Data from in-memory FS, source: linux-8:3306, target: s1cbj26efjaq6cmx:3306.
NOTE: SRC: Backup lock is not supported in MySQL 5.6 and DDL changes will not be blocked. The
dump may fail with an error if schema changes are made while dumping.
SRC: Acquiring global read lock
SRC: Global read lock acquired
Initializing - done
WARNING: SRC: Failed to fetch value of @@GLOBAL.GTID_EXECUTED.
SRC: 2 out of 6 schemas will be dumped and within them 3 tables, 0 views.
Gathering information - done
SRC: All transactions have been started
SRC: Global read lock has been released
NOTE: SRC: When migrating to MySQL HeatWave Service, please always use the latest available v
ersion of MySQL Shell.
SRC: Checking for compatibility with MySQL HeatWave Service 8.0.35
NOTE: SRC: MySQL Server 5.6 detected, please consider upgrading to 8.0 first.
SRC: Compatibility checks finished.ompatibility \ 0 / 5
Validating MySQL HeatWave Service compatibility - done
SRC: Writing global DDL files
SRC: Running data dump using 4 threads.
```

[... output truncated]

```

SRC: Starting data dump
100% (5.30K rows / ~5.27K rows), 0.00 rows/s, 0.00 B/s
SRC: Dump duration: 00:00:00s
SRC: Total duration: 00:00:00s
SRC: Schemas dumped: 2
SRC: Tables dumped: 3
SRC: Data size: 194.61 KB
SRC: Rows written: 5302
SRC: Bytes written: 194.61 KB
SRC: Average throughput: 194.61 KB/s
TGT: Executing common postamble SQL
100% (194.61 KB / 194.61 KB), 38.91 KB/s, 3 / 3 tables done
Recreating indexes - done
TGT: 3 chunks (5.30K rows, 194.61 KB) for 3 tables in 2 schemas were loaded in 5 sec (avg throughput
38.45 KB/s)
TGT: 0 warnings were reported during the load.

---
Dump_metadata:
Binlog_file: mysql-bin.000001
Binlog_position: 328
Executed_GTID_set: ''

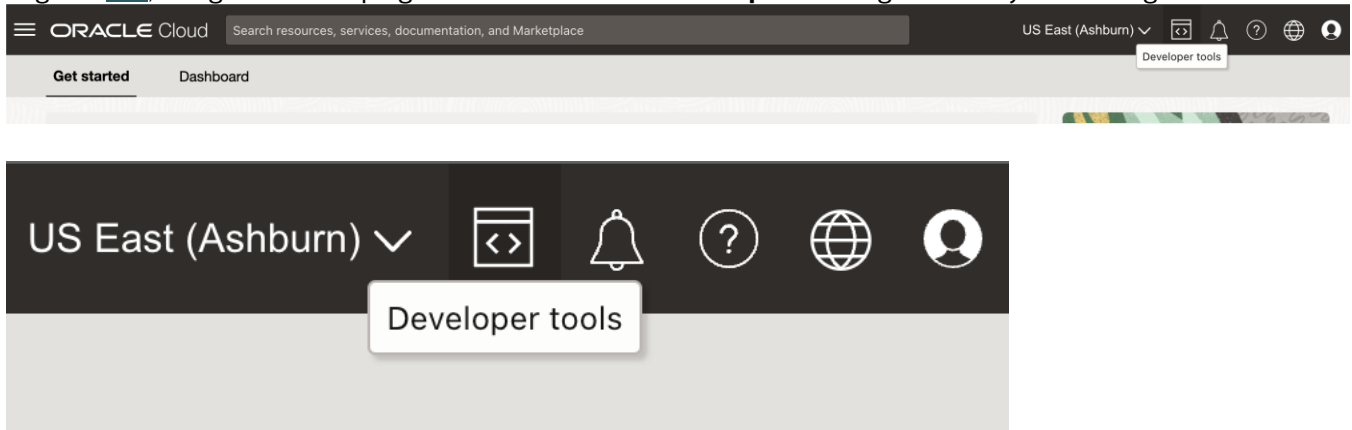
```

Note: once the MySQL Shell copy utility finishes, all your data will be copied over from MariaDB to HeatWave MySQL. This completes the migration process. You can end your tmux session.

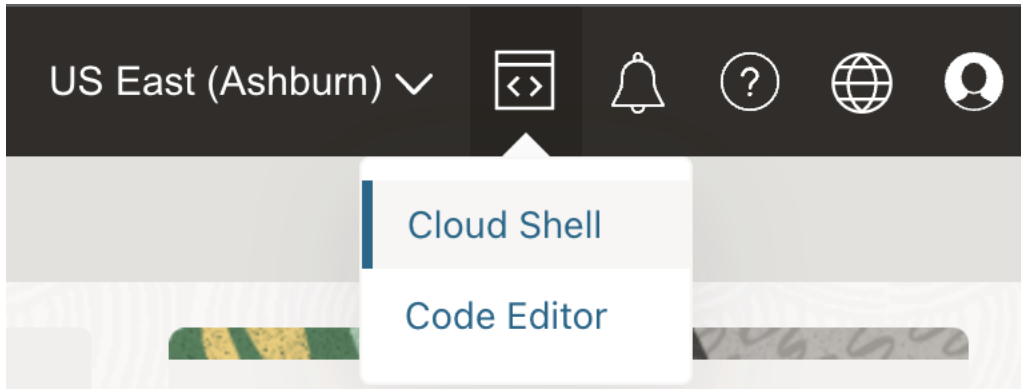
Note: you will have to manually recreate your MariaDB users in HeatWave MySQL.

**VI) (Optional) On OCI, use the Cloud Shell to verify whether the data was migrated successfully from MariaDB to HeatWave MySQL on OCI.**

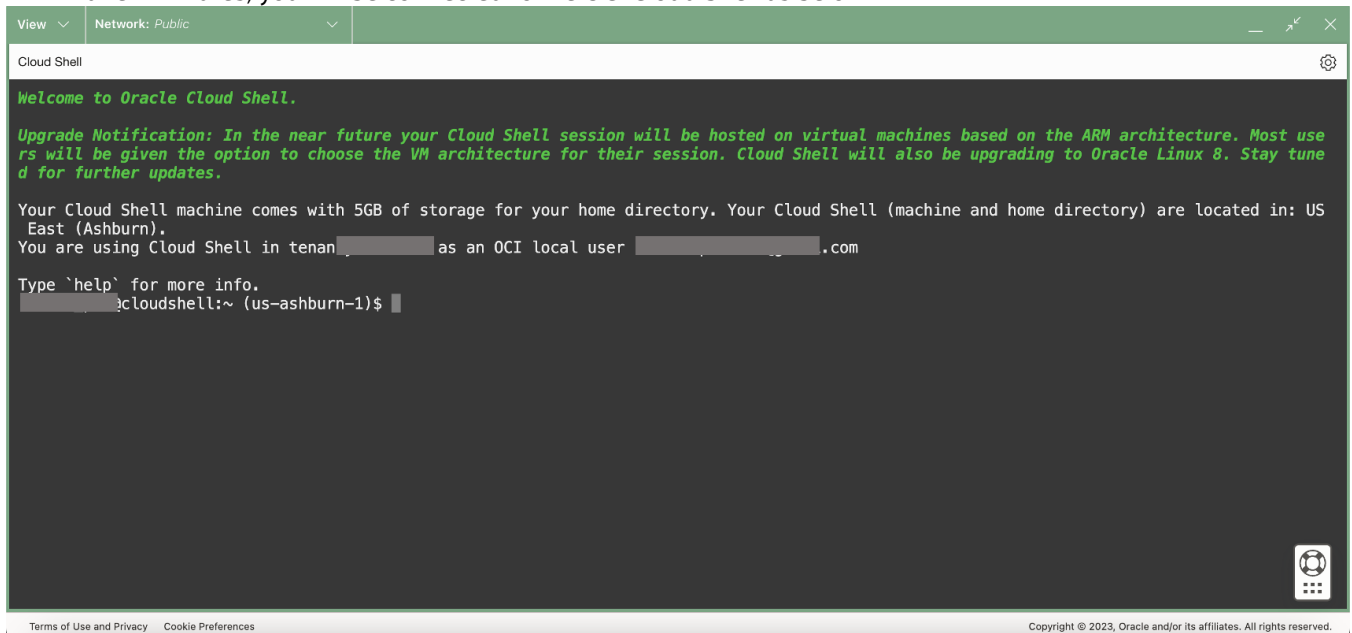
73. Login to [OCI](#), navigate to the top right corner and click on **Developer tools** right next to your OCI Region.



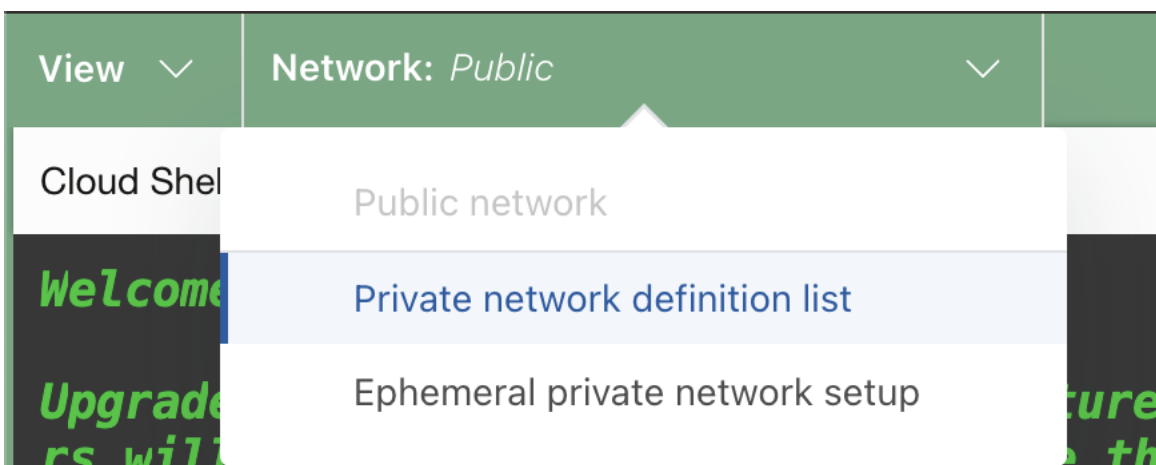
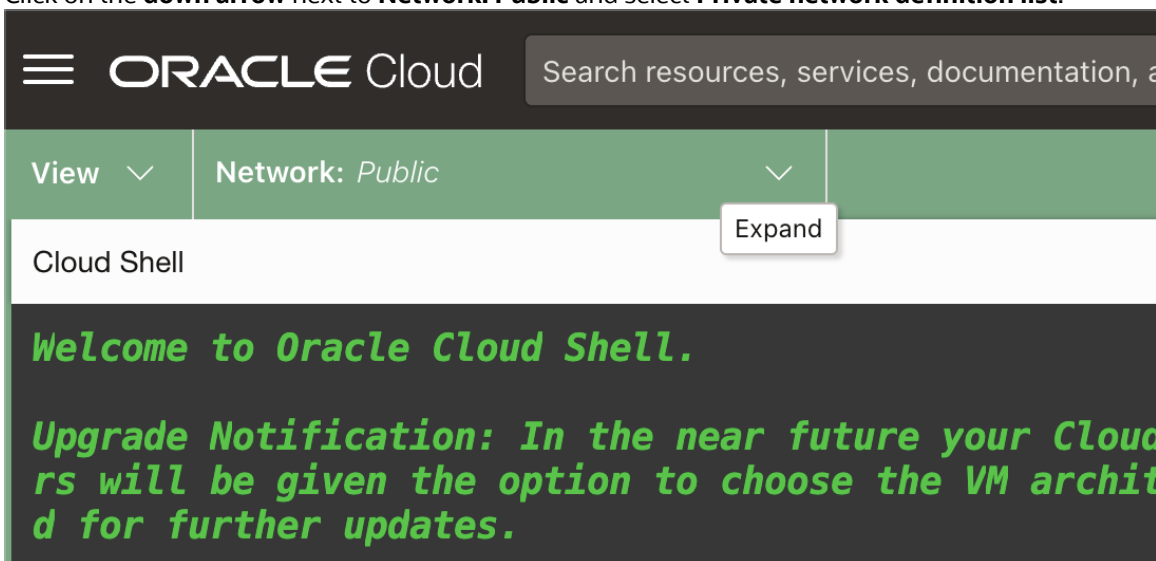
74. Click **Cloud Shell**.



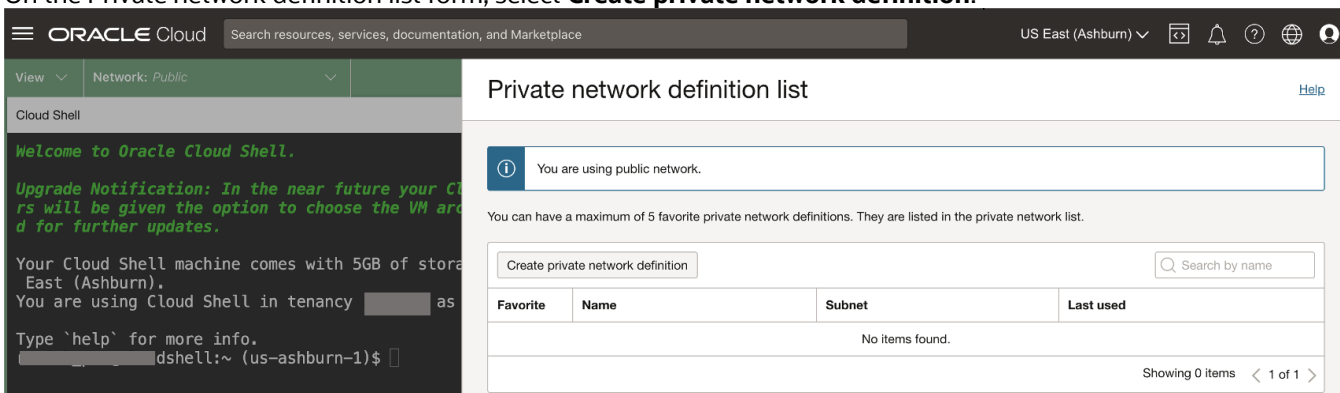
75. Within a few minutes, you will be connected to the OCI Cloud Shell as below:



76. Click on the **down arrow** next to **Network: Public** and select **Private network definition list**.



77. On the Private network definition list form, select **Create private network definition**.





78. Enter a **private network definition name**. From the **VCN in <compartment-name>** dropdown, select the **VCN associated with HeatWave MySQL**. For **Subnet in <compartment-name>** dropdown, select the **private subnet**. Leave the **Network security groups** as-is and **check the box** where it says **Use as active network**. Click **Create**.

Create private network definition [Help](#)

Name  
MySQL-HW-CS

VCN in **(root)** [\(Change compartment\)](#)  
MySQL-VCN

Subnet in **(root)** [\(Change compartment\)](#)  
private subnet-MySQL-VCN

Network security groups (Optional)  
Network security groups in **(root)** [\(Change compartment\)](#)  
Select a network security group  X  
[+ Another NSG](#)

Use as active network

[Create](#) [Cancel](#)

Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

79. Click **Close**.

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

View Network: MySQL-HW-CS (Connecting...) Details

Cloud Shell

```
Welcome to Oracle Cloud Shell.
Upgrade Notification: In the near future your Cloud Shell VMs will be given the option to choose the VM architecture for further updates.
Your Cloud Shell machine comes with 5GB of storage in the US East (Ashburn) region.
You are using Cloud Shell in tenancy (root) as (root).
Type 'help' for more info.
mysql@cloudshell:~ (us-ashburn-1)$
```

Private network definition list [Help](#)

**i** You are using private network "MySQL-HW-CS".

You can have a maximum of 5 favorite private network definitions. They are listed in the private network list.

Create private network definition  Search by name

Favorite	Name	Subnet	Last used
<input type="checkbox"/>	MySQL-HW-CS	...jcity4ma <a href="#">Show</a> <a href="#">Copy</a>	-

Showing 1 item < 1 of 1 >

Default network

Select default network description

Public

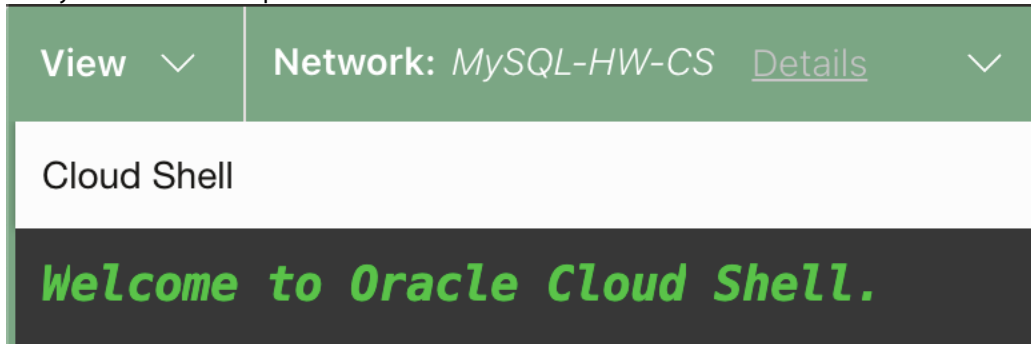
[Close](#)

Terms of Use and Privacy Cookie Preferences

Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

80. Within a few minutes, you will be able to access your private subnet (where HeatWave MySQL resides) from the Cloud Shell. You should see the **Network** change from Public to the **private network definition name**

that you entered in step 76.



81. From the Cloud Shell terminal, login to your HeatWave MySQL instance (by providing the username and private IP of HeatWave MySQL) using MySQL Shell to validate whether the migration was successful:

```
$ mysqlsh <user>@<hostname>:<port-number>
```

-OR-

```
$ mysqlsh -u <user> -p -h <hostname> -P <port-number>
```

```
@cloudshell:~ (us-ashburn-1)$ mysqlsh admin@10.0.1.105
Please provide the password for 'admin@10.0.1.105': *****
Save password for 'admin@10.0.1.105'? [Y]es/[N]o/[Ne]ver (default No): yes
MySQL Shell 8.0.34-commercial

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
Creating a session to 'admin@10.0.1.105'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 118 (X protocol)
Server version: 8.0.35-cloud MySQL Enterprise - Cloud
No default schema selected; type \use <schema> to set one.
MySQL 10.0.1.105:33060+ JS
```

82. Change to the SQL mode of MySQL Shell and run the below commands:

```
MySQL JS> \sql
MySQL SQL> SHOW SCHEMAS;
MySQL SQL> SHOW TABLES IN <schema-name>;
```

```
MySQL 10.0.1.105:33060+ JS \sql
Switching to SQL mode... Commands end with ;
Fetching global names for auto-completion... Press ^C to stop.
MySQL 10.0.1.105:33060+ SQL SHOW SCHEMAS;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| mysql_audit |
| performance_schema |
| sys |
| test |
| world |
+-----+
7 rows in set (0.0011 sec)
MySQL 10.0.1.105:33060+ SQL SHOW TABLES IN world;
+-----+
| Tables_in_world |
+-----+
| city |
| country |
| countrylanguage |
+-----+
3 rows in set (0.0013 sec)
```

83. You can run the below query on every table that you have for your on-premises MariaDB and HeatWave MySQL on OCI to ensure that the row count matches on both sides:

```
MySQL SQL> SELECT COUNT(*) FROM <schema-name>.<table-name>;
```

84. Here is our row count comparison for MariaDB and HeatWave MySQL:

MariaDB row count:

```
MySQL localhost:33060+ ssl SQL > USE world;
Default schema set to `world`.
Fetching global names, object names from `world` for auto-completion... Press ^C to stop
.
MySQL localhost:33060+ ssl world SQL > SELECT COUNT(*) FROM city;
+-----+
| COUNT(*) |
+-----+
|      4079 |
+-----+
1 row in set (0.0015 sec)
MySQL localhost:33060+ ssl world SQL > SELECT COUNT(*) FROM country;
+-----+
| COUNT(*) |
+-----+
|       239 |
+-----+
1 row in set (0.0008 sec)
MySQL localhost:33060+ ssl world SQL > SELECT COUNT(*) FROM countrylanguage;
+-----+
| COUNT(*) |
+-----+
|       984 |
+-----+
1 row in set (0.0009 sec)
MySQL localhost:33060+ ssl world SQL >
```

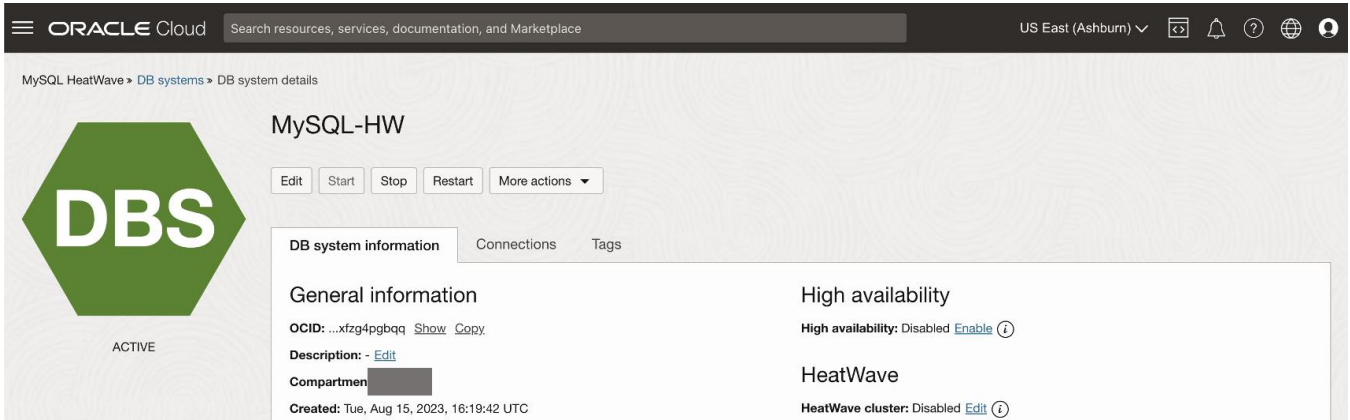
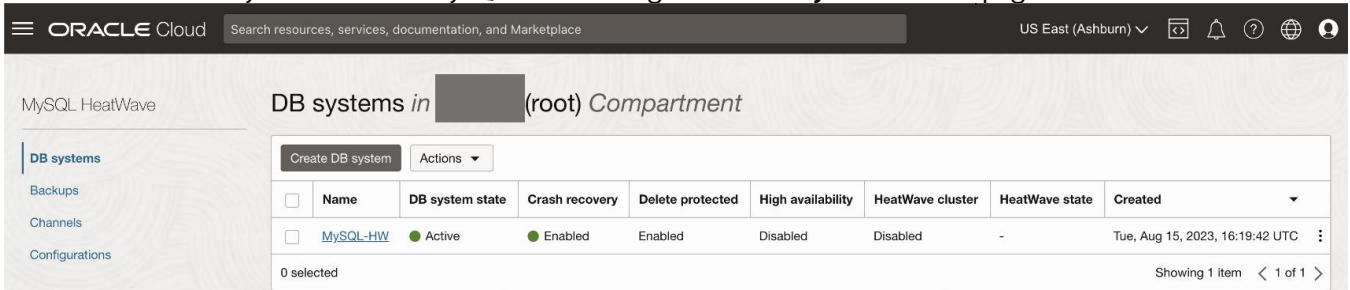
HeatWave MySQL row count:

```
MySQL 10.0.1.140:33060+ ssl SQL > USE world;
Default schema set to `world`.
Fetching global names, object names from `world` for auto-completion... Press ^C to stop.
MySQL 10.0.1.140:33060+ ssl world SQL > SELECT COUNT(*) FROM city;
+-----+
| COUNT(*) |
+-----+
|      4079 |
+-----+
1 row in set (0.0030 sec)
MySQL 10.0.1.140:33060+ ssl world SQL > SELECT COUNT(*) FROM country;
+-----+
| COUNT(*) |
+-----+
|       239 |
+-----+
1 row in set (0.0160 sec)
MySQL 10.0.1.140:33060+ ssl world SQL > SELECT COUNT(*) FROM countrylanguage;
+-----+
| COUNT(*) |
+-----+
|       984 |
+-----+
1 row in set (0.0166 sec)
MySQL 10.0.1.140:33060+ ssl world SQL >
```

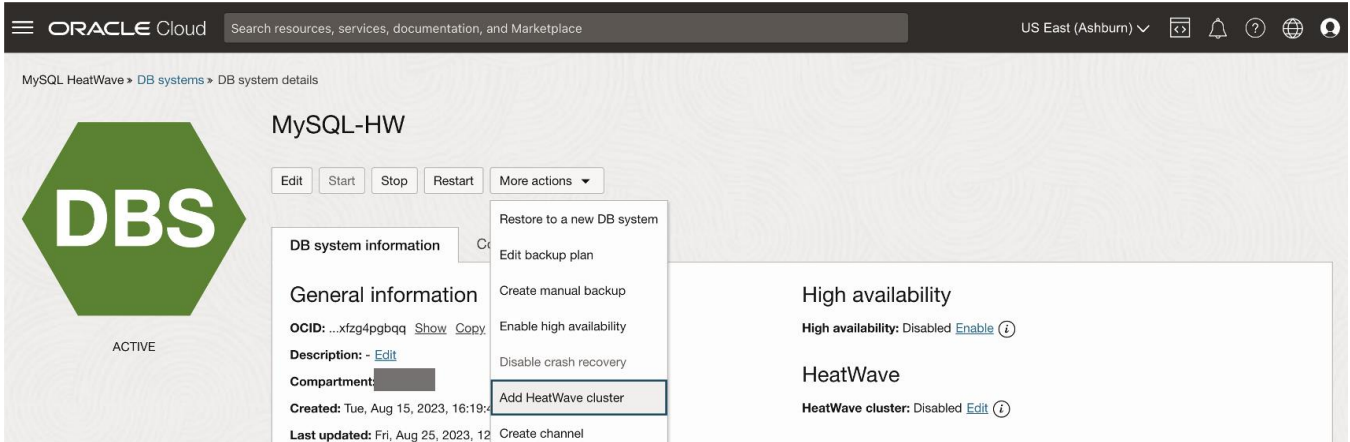
85. After validating, you can have your application/s point to the new HeatWave MySQL instance.

**VII) (Optional) On OCI, if the HeatWave option was enabled during HeatWave MySQL DB creation, add the HW Cluster and load data from MySQL InnoDB storage into the HW Cluster using automation.**

- 86. Login to [OCI](#). Click on the navigation menu, go to **Databases**, and click **HeatWave MySQL**.
- 87. Click on the name of your HeatWave MySQL instance to go to the **DB System Details** page.



- 88. Click **More actions** and click **Add HeatWave cluster**.



89. Click **Estimate node**.

The screenshot shows the Oracle Cloud console interface for adding a HeatWave cluster. The main heading is "Add HeatWave cluster". Below it, there is a message: "Add a HeatWave cluster to the DB system **MySQL-HW** with shape **MySQL.HeatWave.VM.Standard**. [What shapes support HeatWave?](#)".

The "Configure HeatWave cluster" section includes:

- Select a shape:** A box showing "HeatWave.512GB" with details: "CPU core count: 16", "Memory size: 512 GB", and "Max network bandwidth: 16Gbps". A "Change shape" button is on the right.
- Node:** A text input field containing "1". Below it, a note says "Specify a number between 1 and 64."
- MySQL HeatWave Lakehouse:** A checkbox that is currently unchecked. A tooltip indicates it "Enables you to use data from Object Storage."
- Memory:** A label "Memory: 512 GB" and an "Estimate node" button.
- A note at the bottom: "This operation can take several minutes to complete."

At the bottom of the configuration area, there are "Add HeatWave cluster" and "Cancel" buttons. The footer contains "Terms of Use and Privacy", "Cookie Preferences", and "Copyright © 2023, Oracle and/or its affiliates. All rights reserved."

90. Click **Generate estimate**. This step will estimate the number of HeatWave nodes required by selecting the schemas or tables you want to analyze with HeatWave.

The screenshot shows the same Oracle Cloud console interface, but with the "Estimate node" panel open on the right side. The "Add HeatWave cluster" panel is dimmed in the background.

The "Estimate node" panel contains:

- A heading "Estimate node".
- A message: "Estimate number of required nodes by selecting the schemas or tables you want to analyze with HeatWave. This operation takes few minutes to complete." with a help icon.
- A "Generate estimate" button.
- A text box containing "No schema information available."

At the bottom of the "Estimate node" panel, there are "Apply estimated node" and "Cancel" buttons. The footer remains the same as in the previous screenshot.

91. Within a few minutes, the list of your schemas that are in the MySQL InnoDB storage engine will be listed. **Check the box** next to the schema or table name that you wish to load in HeatWave for query acceleration and to run OLAP and ML workloads - alongside OLTP.

The screenshot shows the Oracle Cloud console interface. On the left, the 'Add HeatWave cluster' panel is visible, showing configuration options for a HeatWave.512GB cluster. On the right, the 'Estimate node' panel is active, displaying a table of schemas and tables to be analyzed. The table has the following data:

Name	Memory estimate	Information
world	9 MB	Number of tables: 3

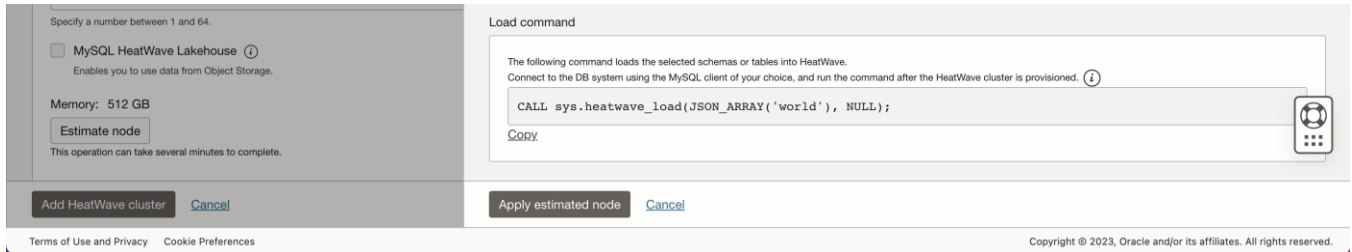
The 'Total memory selected' is currently 0 Bytes. The 'Summary' section indicates that no schema or table has been selected for the estimate.

92. After selecting the schemas or tables, scroll down on that page until you see the **Load command**. Copy the **CALL sys.heatwave\_load** command and save it. Click **Apply estimated node**.

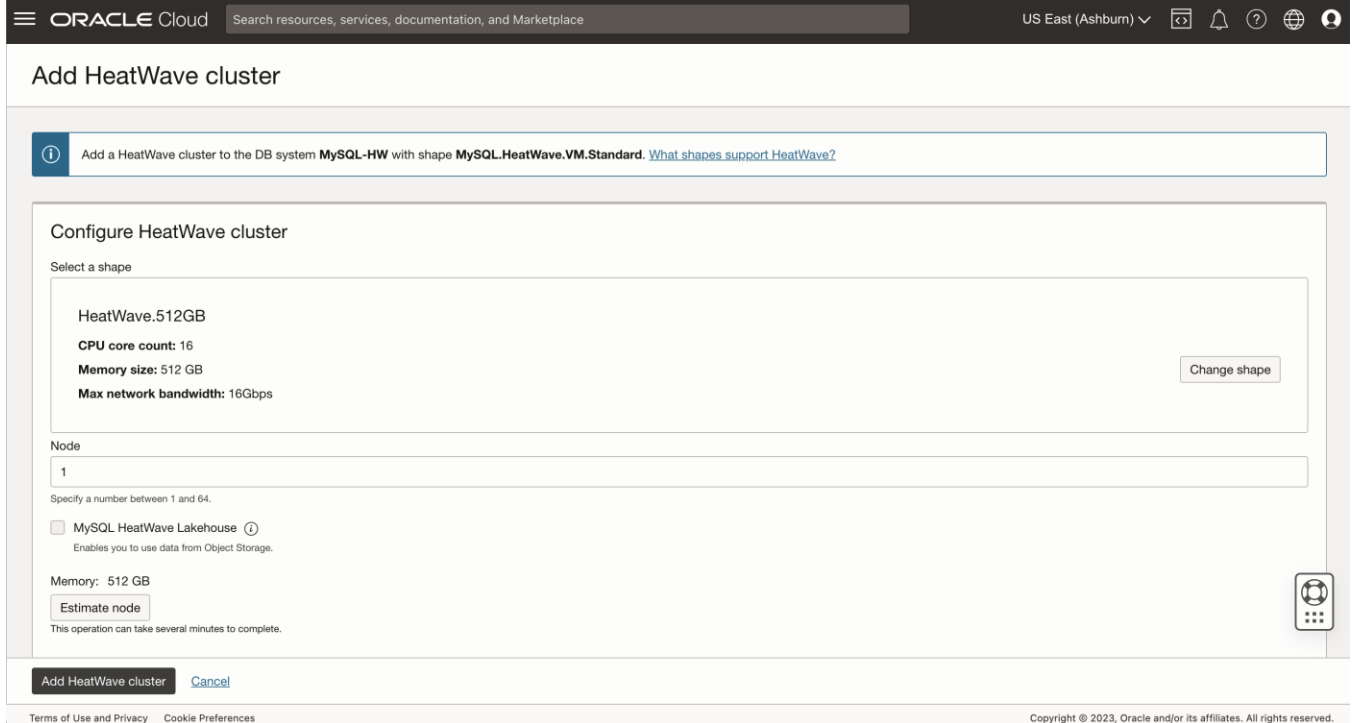
This screenshot shows the same Oracle Cloud console interface as the previous one, but with the 'world' schema selected in the 'Estimate node' panel. The 'Total memory selected' is now 9 MB. The 'Summary' section provides details for the selected node:

HeatWave.512GB  
 CPU core count: 16  
 Memory size: 512 GB  
 Max network bandwidth: 16Gbps  
 Node: 1  
 Total memory required: 9 MB  
 Total memory: 512 GB

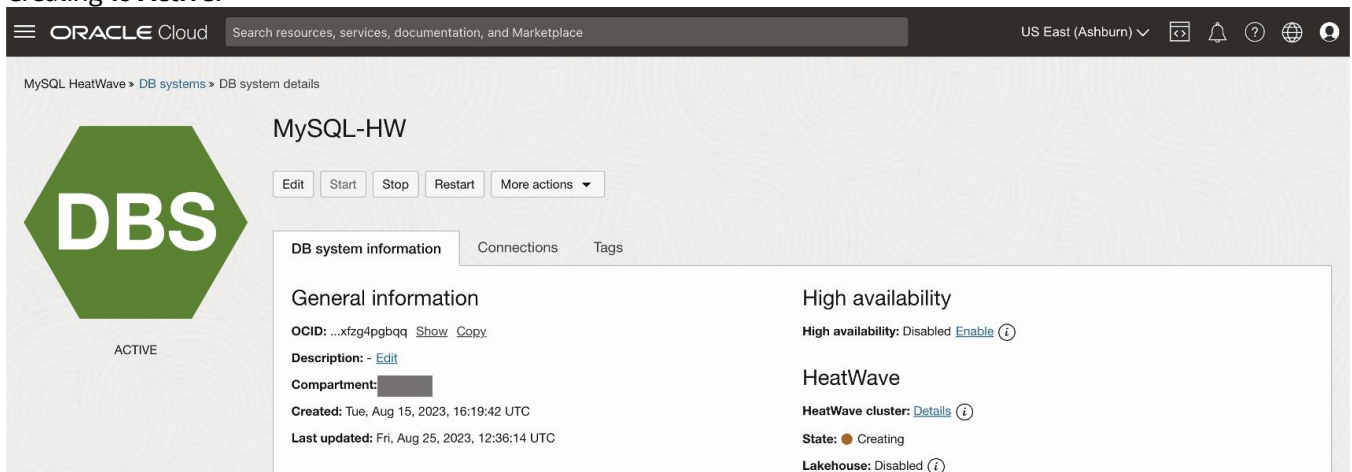
The 'Load command' section is visible at the bottom of the 'Estimate node' panel, and the 'Apply estimated node' button is highlighted.



93. Executing the previous step will change the HeatWave node count depending on the data you have selected to load into HeatWave's in-memory engine. Click **Add HeatWave cluster** to finish adding the HeatWave cluster creation process.



94. The HeatWave cluster will be ready within a few minutes. You should see the HeatWave state change from **Creating** to **Active**.





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

MySQL HeatWave > DB systems > DB system details

## MySQL-HW

ACTIVE

Edit Start Stop Restart More actions

DB system information Connections Tags

**General information**

OCID: ...xfzg4pgbqg [Show](#) [Copy](#)

Description: - [Edit](#)

Compartment: [REDACTED]

Created: Tue, Aug 15, 2023, 16:19:42 UTC

Last updated: Fri, Aug 25, 2023, 12:36:14 UTC

**High availability**

High availability: Disabled [Enable](#) ⓘ

**HeatWave**

HeatWave cluster: [Details](#) [Edit](#) ⓘ

State: ● Active

95. Connect to your HeatWave MySQL system using MySQL Shell via Cloud Shell.

```
$ mysqlsh <user>@<hostname>:<port-number>
```

-OR-

```
$ mysqlsh -u <user> -p -h <hostname> -P <port-number>
```

```
[REDACTED]@cloudshell:~ (us-ashburn-1)$ mysqlsh admin@10.0.1.140
Please provide the password for 'admin@10.0.1.140': *****
Save password for 'admin@10.0.1.140'? [Y]es/[N]o/Ne[v]er (default No): Y
MySQL Shell 8.0.34-commercial

Copyright (c) 2016, 2023, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '? ' for help; '\quit' to exit.
Creating a session to 'admin@10.0.1.140'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 2332 (X protocol)
Server version: 8.0.34-u1-cloud MySQL Enterprise - Cloud
No default schema selected; type \use <schema> to set one.
MySQL 10.0.1.140:33060+ ssl JS >
```

96. Switch to the SQL mode of MySQL Shell and execute the Load command that we had copied earlier to load data into HeatWave from the MySQL InnoDB storage.

```
MySQL JS> \sql
MySQL SQL> CALL sys.heatwave_load(JSON_ARRAY('world'), NULL);
```

Note: replace the `sys.heatwave_load` command with what you have.

```
MySQL 10.0.1.140:33060+ ssl JS > \sql
Switching to SQL mode... Commands end with ;
Fetching global names for auto-completion... Press ^C to stop.
MySQL 10.0.1.140:33060+ ssl SQL > CALL sys.heatwave_load(JSON_ARRAY('world'), NULL);
+-----+
| INITIALIZING HEATWAVE AUTO PARALLEL LOAD |
+-----+
| Version: 2.20 |
| Load Mode: normal |
| Load Policy: disable_unsupported_columns |
| Output Mode: normal |
+-----+
6 rows in set (1.4644 sec)
+-----+
| OFFLOAD ANALYSIS |
+-----+
| Verifying input schemas: 1 |
| User excluded items: 0 |
+-----+
| SCHEMA NAME | OFFLOADABLE TABLES | OFFLOADABLE COLUMNS | SUMMARY OF ISSUES |
+-----+
| `world` | 3 | 24 | |
+-----+
Total offloadable schemas: 1
```

[... output truncated]

```
+-----+
| LOADING TABLE |
+-----+
| TABLE (3 of 3): `world`.`countrylanguage` |
| Commands executed successfully: 3 of 3 |
| Warnings encountered: 0 |
| Table loaded successfully! |
| Total columns loaded: 4 |
| Table loaded using 1 thread(s) |
| Elapsed time: 402.27 ms |
+-----+
8 rows in set (1.4644 sec)
+-----+
| LOAD SUMMARY |
+-----+
| SCHEMA NAME | TABLES LOADED | TABLES FAILED | COLUMNS LOADED | LOAD DURATION |
+-----+
| `world` | 3 | 0 | 24 | 1.40 s |
+-----+
6 rows in set (1.4644 sec)
Query OK, 0 rows affected (1.4644 sec)
MySQL 10.0.1.140:33060+ ssl SQL >
```

97. You now have a complete HeatWave MySQL cluster.

To learn more about using HeatWave, please visit [our documentation](#).

---

## Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 [blogs.oracle.com](https://blogs.oracle.com)

 [facebook.com/oracle](https://facebook.com/oracle)

 [twitter.com/oracle](https://twitter.com/oracle)

---

Copyright © 2024, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.