

Migration Guide: On-premises MySQL 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 on-premises MySQL 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 on-premises MySQL v5.7 and above. This can be a MySQL Community Edition, MySQL Standard Edition, MySQL Enterprise Edition, or Percona Server.
- 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 on-premises MySQL instance, use an admin/root user.
- You do not need to make any configuration changes to your on-premises MySQL for this migration.
- If you have MySQL replication configured in your current on-premises 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 MySQL 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 MySQL 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 MySQL 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 on-premises MySQL 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 on-premises MySQL 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 8.1 (or above) on an on-premises instance that can connect to your on-premises MySQL.

[MySQL Shell will be used to copy DDL and data from on-premises MySQL to HeatWave MySQL on OCI. You must download MySQL Shell 8.1 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 on-premises MySQL using MySQL Shell. Afterwards, execute the MySQL Shell `util.copyInstance()` utility to export all schemas (including users, indexes, routines, triggers) from on-premises MySQL 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/8.1/en/mysql-shell-utils-copy.html>

VI) (Optional) On OCI, use the Cloud Shell to verify whether the data was migrated successfully from on-premises MySQL 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>

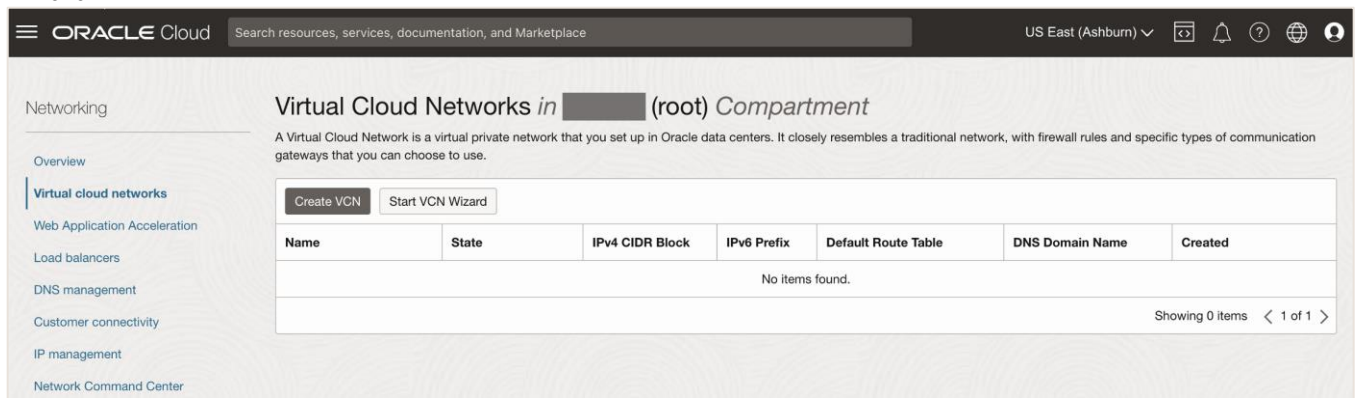
II) Set up a VPN connection from OCI to on-premises.

Note: this guide uses OpenVPN Access Server which lets you connect your on-premises MySQL with OCI HeatWave MySQL. 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 on-premises MySQL instance version and [the sample database \("world"\)](#) that will be migrated for this guide. The sample world database consists of 3 tables.

```
MySQL localhost:33060+ ssl SQL > SELECT @@VERSION;
+-----+
| @@VERSION |
+-----+
| 8.0.33    |
+-----+
1 row in set (0.0015 sec)
MySQL localhost:33060+ ssl SQL > SHOW SCHEMAS;
+-----+
| Database |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
| world          |
+-----+
5 rows in set (0.0036 sec)
MySQL localhost:33060+ ssl SQL > SHOW TABLES IN world;
+-----+
| Tables_in_world |
+-----+
| city             |
| country          |
| countrylanguage |
+-----+
3 rows in set (0.0038 sec)
MySQL localhost:33060+ ssl SQL >
```

2. Log in to [OCI](#) and create a VCN. Open the navigation menu, click **Networking**, and click **Virtual cloud networks**.
3. Ensure you are in your desired compartment - we have chosen the `root` compartment. Click **Start VCN Wizard**.



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

Start VCN Wizard

Create VCN with Internet Connectivity

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

Creates a VCN with a public subnet that can be reached from the internet. Also creates a private subnet that can connect to the internet through a NAT gateway, and also privately connect to the Oracle Services Network.

Includes: VCN, public subnet, private subnet, internet gateway (IG), NAT gateway (NAT), service gateway (SG).

Start VCN Wizard Cancel

5. 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

1 Configuration

2 Review and create

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

Next Cancel

VCN with internet connectivity

Includes:

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

6 Migration Guide: On-premises MySQL to HeatWave MySQL on Oracle Cloud Infrastructure (OCI)

6. 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:** A checkbox labeled "Use DNS hostnames in this VCN" is checked. Below it is a note: "Required for instance hostname assignment if you plan to use VCN DNS or a third-party DNS. This choice cannot be changed after the VCN is created. [Learn more.](#)"
- Configure public subnet:** A dropdown menu is set to "IPv4 CIDR block". The input field contains "10.0.0/24". Below the input is an example: "Example: 172.16.0.0/16." and a button "+ Another IP address type".
- Configure private subnet:** A dropdown menu is set to "IPv4 CIDR block". The input field contains "10.0.1.0/24". Below the input is an example: "Example: 172.16.0.0/16." and a button "+ Another IP address type".

At the bottom left, there are "Next" and "Cancel" buttons. At the bottom right, there is a "Show tagging options" link and a mobile app icon. The footer contains "Terms of Use and Privacy", "Cookie Preferences", and "Copyright © 2023, Oracle and/or its affiliates. All rights reserved."

7. 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 a 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). The main content area is divided into sections:

- Resource availability checked successfully:** A notification box with an information icon and a "Close" button.
- Oracle VCN:** A summary box containing:
 - Name: MySQL-VCN
 - Compartment: (root)
 - Tags: VCN: VCN-2023-05-15T14:57:35
 - IPv4 CIDR block: 10.0.0.0/16
 - DNS label: MySQLVCN
 - DNS domain name: MySQLVCN.oraclevcn.com
- Subnets:** A summary box containing:
 - Public subnet
 - Subnet name: public subnet-MySQL-VCN
 - IPv4 CIDR block: 10.0.0.0/24
 - Security list name: default security list for MySQL-VCN
 - Route table name: default route table for MySQL-VCN

At the bottom left, there are "Previous", "Create", and "Cancel" buttons. At the bottom right, there is a mobile app icon. The footer contains "Terms of Use and Privacy", "Cookie Preferences", and "Copyright © 2023, Oracle and/or its affiliates. All rights reserved."

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

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

Create a VCN with internet connectivity

1 Configuration
2 Review and create

Created VCN

Creating resources

- VCN creation complete
- Create VCN (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
- Update private subnet (1 resolved) Done

View VCN

Terms of Use and Privacy Cookie Preferences Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

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.

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

Networking > Customer connectivity > Site-to-Site VPN

Customer connectivity

Overview
Site-to-Site VPN
FastConnect
Dynamic routing gateway
Customer-premises equipment

Site-to-Site VPN in (root) Compartment

Site-to-Site VPN securely connects your on-premises corporate network to Oracle Cloud Infrastructure, using your existing internet connection.

If your users have client devices that need offsite access to Oracle Cloud resources, you can also create an OpenVPN access server. See their [marketplace solution](#).

Create IPsec connection Start VPN wizard

Name	Lifecycle state	Customer-premises equipment	Dynamic routing gateway	Created
No items found.				

Showing 0 items < 1 of 1 >

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**.

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

Marketplace > OpenVPN Access Server

OpenVPN Access Server

VPN solution for Virtual Cloud Network (VCN). Two connections for FREE. Buy license for more

OpenVPN Access Server delivers the enterprise VPN your business has been looking for. Protect your data communications, secure IoT resources, and provide encrypted remote access to on-premise, hybrid, and public cloud resources.

Categories: Networking, Security

Type: Stack

Version: AS 2.8.3 Stack Gov (...)

Compartment: (root)

Software price per OCPU: BYOL (Bring your own license)

There are additional fees for the infrastructure usage.

I have reviewed and accept the [Oracle terms of use](#) and the [Partner terms and conditions](#).

Launch Stack

Reminder: Patch the instance once installed.

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' stack. 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'. There are two optional fields: 'Name' with the value 'OpenVPN Access Server-20230515143705' and 'Description' with the text '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 are 'Next' and 'Cancel' buttons.

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 divided into two sections: 'Compute Configuration' and 'Application Configuration'. The 'Compute Configuration' section has a text field for 'OpenVPN Access Server Name' with the value 'openvpn_access_server' and a dropdown for 'Compute Shape' with the value 'VM.Standard2.2'. The 'Application Configuration' section has a text field for 'Administrator Username' with the value 'root', a text field for 'Administrator Password' with masked characters, and a text field for 'Activation Key' with the value 'Optional'. At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons.

14. 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**.

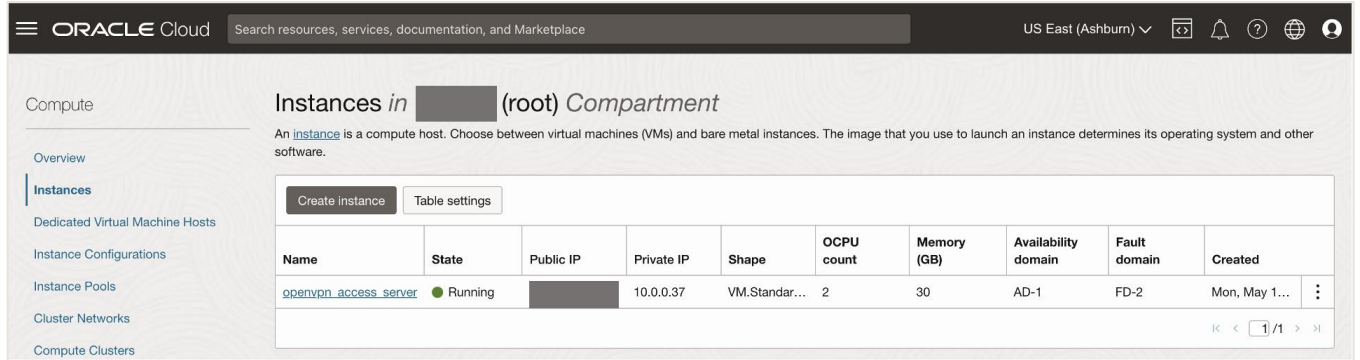
The screenshot shows the 'Create stack' page in the Oracle Cloud console, specifically the 'Network Configuration' step. The page has a dark header with the Oracle Cloud logo, a search bar, and the region 'US East (Ashburn)'. On the left, a progress indicator shows three steps: 'Stack information', 'Configure variables', and 'Review'. The main content area is divided into two sections: 'Network Configuration' and 'Additional Configuration'. In the 'Network Configuration' section, 'Use Existing VCN' is selected in the 'Network Strategy' dropdown. Below it, 'MySQL-VCN' is selected in the 'Existing Network' dropdown, and 'public subnet-MySQL-VCN (Regional)' is selected in the 'Existing Subnet' dropdown. The 'Additional Configuration' section shows 'Compartment' set to '(root)'. There is a text input field for 'Public SSH Key string' with the label 'Optional'. At the bottom, there are 'Previous', 'Next', and 'Cancel' buttons. The footer contains 'Terms of Use and Privacy', 'Cookie Preferences', and 'Copyright © 2023, Oracle and/or its affiliates. All rights reserved.'

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

The screenshot shows the 'Create stack' page in the Oracle Cloud console, specifically the 'Review' step. The page has the same dark header as the previous screenshot. The progress indicator on the left now highlights the 'Review' step. The main content area contains a summary of the configuration. At the top, it says 'Verify your configuration variables, and then create your stack. The apply job will automatically run to create resources specified in the configuration. Due to limited space, we show only variables without default values or that you edited.' Below this, there are three sections: 'Stack information', 'Compute Configuration', and 'Application Configuration'. 'Stack information' shows: Name: OpenVPN Access Server-20230515174018, Description: ...erver. Show Copy, Compartment: ...qedpia Show Copy, Terraform version: 0.14.x. 'Compute Configuration' shows: Compute Shape: VM.Standard2.2. 'Application Configuration' shows: Administrator Username: root, Administrator Password: At the bottom, there are 'Previous', 'Create', and 'Cancel' buttons. The footer contains 'Terms of Use and Privacy', 'Cookie Preferences', and 'Copyright © 2023, Oracle and/or its affiliates. All rights reserved.'

16. 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.

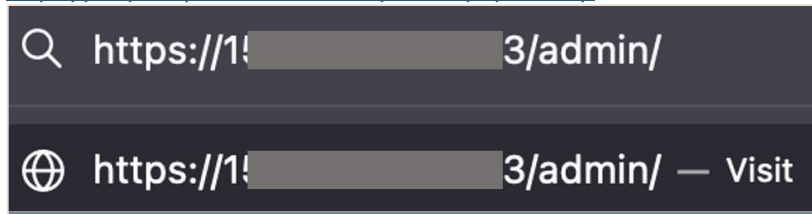
- Copy and save the Public and the Private IP of the `openvpn_access_server`.



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

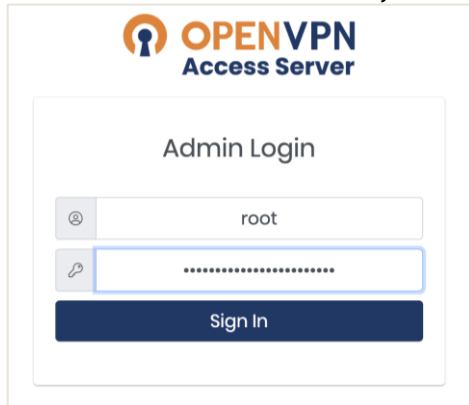
- Open a web browser and enter the following in the search bar.

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

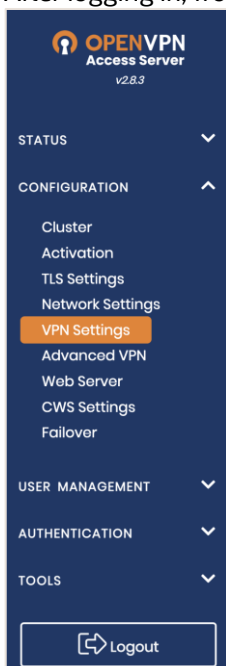


Note: in the web browser when prompted, click **Advanced** and click **Proceed to <openvpn-access-server-public-ip> (unsafe)** or **Accept the Risk and Continue**.

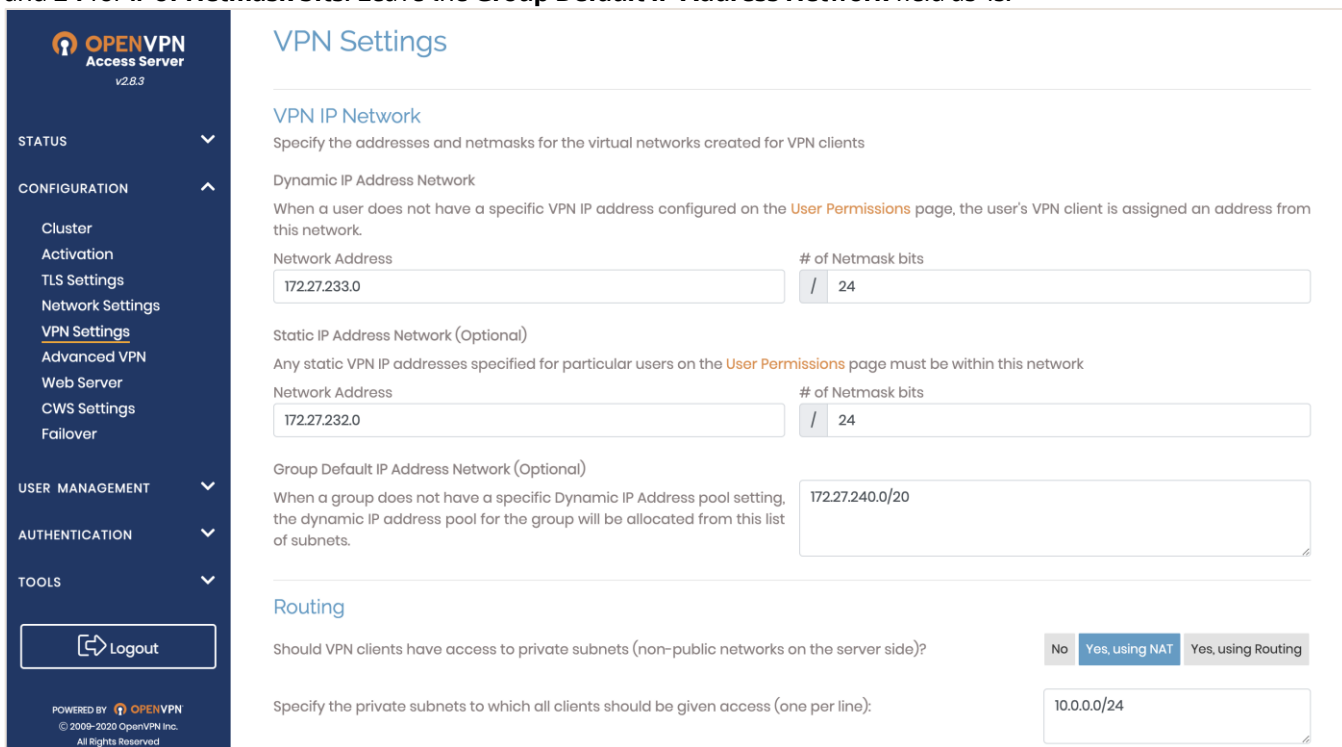
- Enter the admin credentials that you configured earlier in step 13 to log in.



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
10.0.1.0/24

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

Save Settings

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

Logout

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.

The screenshot shows the 'User Permissions' interface. On the left is a navigation sidebar with categories like STATUS, CONFIGURATION, USER MANAGEMENT (highlighted), AUTHENTICATION, and TOOLS. Under USER MANAGEMENT, 'User Permissions' is selected. The main area has a search bar and a table of users. The table has columns: Username, Group, More Settings, Admin, Allow Auto-login, Deny Access, and Delete. There are three rows: 'openvpn', 'root', and 'New Username'. The 'New Username' field contains 'openvpnuser'. The 'More Settings' icon (pencil) is highlighted for the 'openvpnuser' row. Below the table is a checkbox for 'Require user permissions record for VPN access' set to 'No'. A 'Save Settings' button is at the bottom right.

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**.

The screenshot shows the configuration page for the 'openvpnuser' user. The page is divided into several sections: Local Password, IP Addressing, Access Control, VPN Gateway, and DMZ settings. Under 'Local Password', there is a password field and radio buttons for 'Allow password change from CWS' and 'Enable password strength checking in CWS'. Under 'IP Addressing', 'Use Static' is selected, and '172.27.232.25' is entered in the 'VPN Static IP Address' field. Under 'Access Control', 'Use Routing' is selected, and '10.0.0/24' and '10.0.10/24' are entered in the 'Allow Access To these Networks' field. Under 'Allow Access From', the 'all server-side private subnets' checkbox is checked. The 'VPN Gateway' section has 'No' selected for 'Configure VPN Gateway'. The 'DMZ settings' section has 'No' selected for 'Configure DMZ IP address'.

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

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

Require user permissions record for VPN access

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>/>

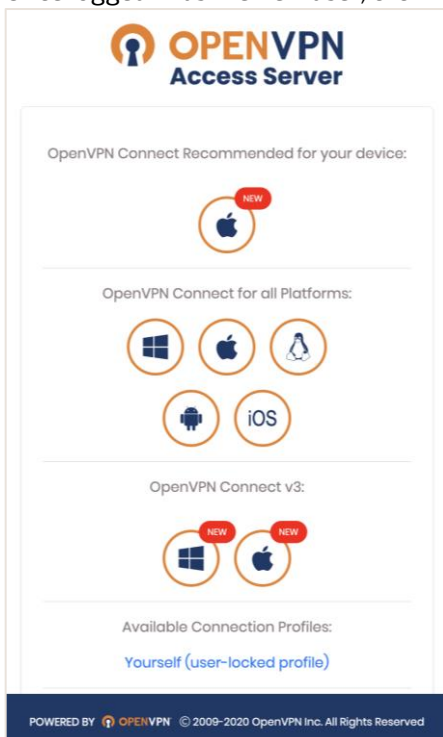
OPENVPN Access Server

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.



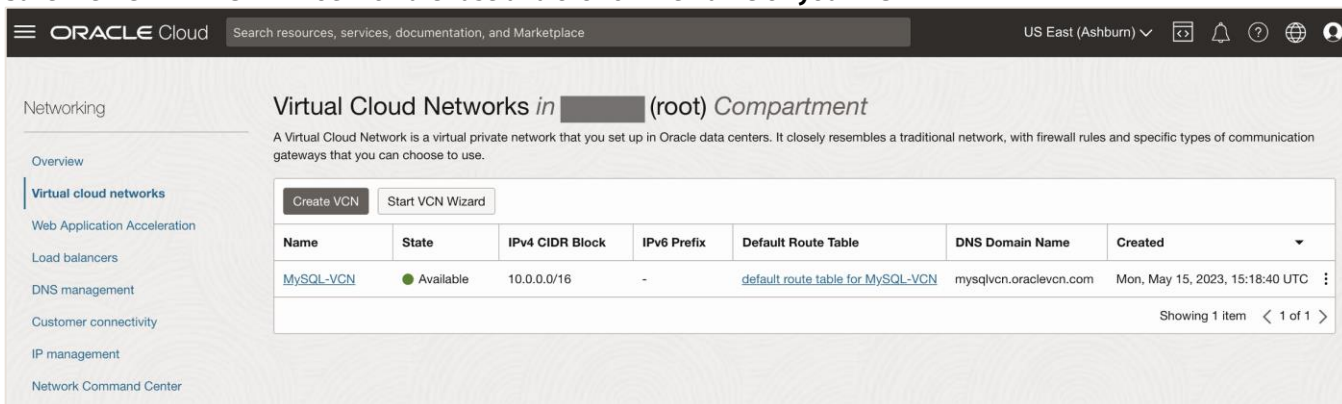
30. Click the appropriate platform icon depending on the Operating System (OS) you are running to download the OpenVPN client. For this guide, we are using macOS. After downloading the client, install it. For more information see, [Installation guide for macOS](#), [Installation guide for Windows](#), 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, Created date, IPv4 CIDR Block, and IPv6 Prefix. 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".

Name	State	Number of Rules	Created
route table for private subnet-MySQL-VCN	Available	2	Mon, May 15, 2023, 15:18:41 UTC
default route table for MySQL-VCN	Available	1	Mon, May 15, 2023, 15:18:40 UTC

36. Click **Add Route Rules**.

The screenshot shows the Oracle Cloud console interface for the "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 and Created date. 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".

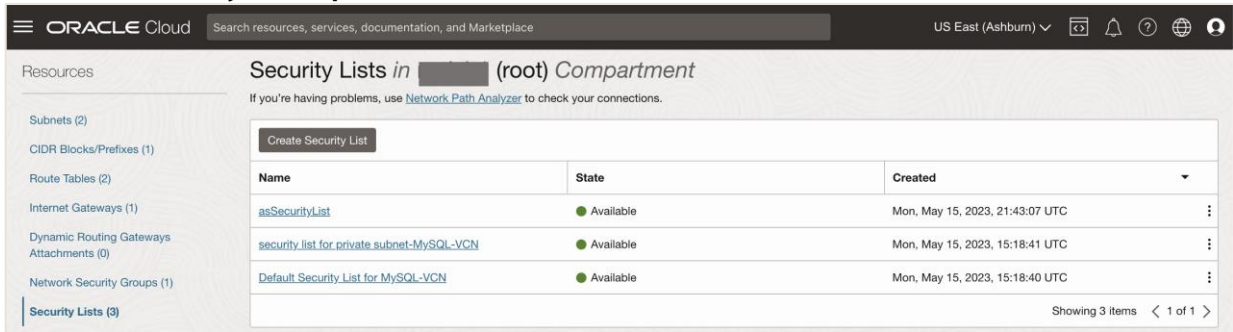
Destination	Target Type	Target	Route Type	Description
<input type="checkbox"/> 0.0.0.0/0	NAT Gateway	NAT.gateway-MySQL-VCN	Static	
<input type="checkbox"/> All IAD Services In Oracle Services Network	Service Gateway	Service.gateway-MySQL-VCN	Static	

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**.

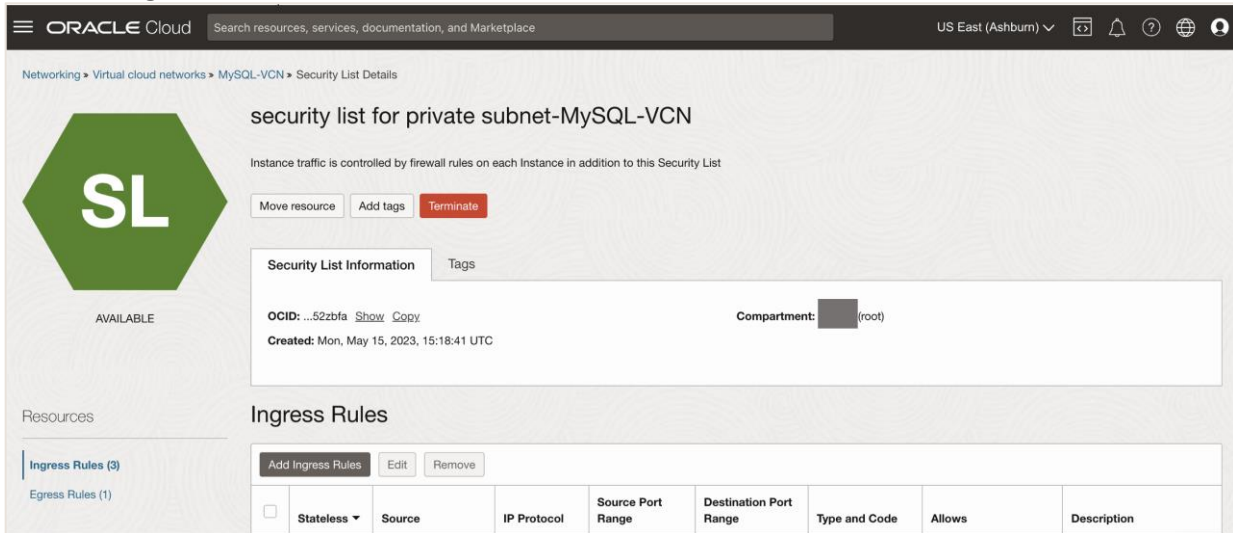
38. Go back to the Virtual Cloud Network Details page of your VCN and click **Security Lists**.

Name	State	Created
asSecurityList	Available	Mon, May 15, 2023, 21:43:07 UTC
security_list_for_private_subnet-MYSQL-VCN	Available	Mon, May 15, 2023, 15:18:41 UTC
Default_Security_List_for_MYSQL-VCN	Available	Mon, May 15, 2023, 15:18:40 UTC

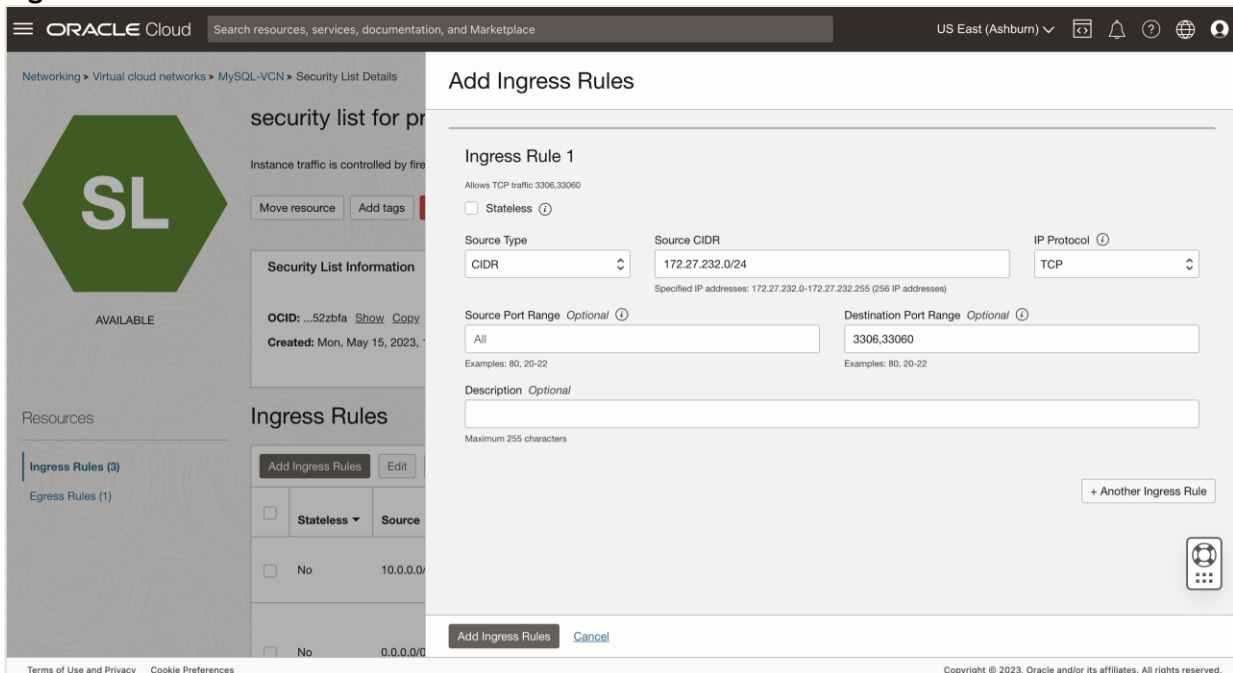
39. Click on the **security list for private subnet-<vcn-name>**.



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 'security list for private subnet-MYSQL-VCN' page. The page displays the security list name, OCID, creation date, and compartment. The 'Ingress Rules' section shows a table with columns: Stateless, Source, IP Protocol, Source Port Range, Destination Port Range, Type and Code, Allows, and Description. The table contains two rules:

Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
<input type="checkbox"/>	No	10.0.0.0/16	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

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 dialog is titled 'Add Ingress Rules' and shows 'Ingress Rule 1' configuration. It includes fields for 'Source Type' (set to CIDR), 'Source CIDR' (10.0.0.0/16), 'IP Protocol' (TCP), 'Source Port Range' (All), and 'Destination Port Range' (3306,33060). There is also a 'Description' field and a '+ Another Ingress Rule' button.

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 sidebar, a main header, and a section for Security Lists.

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
asSecurityList	Available	Mon, May 15, 2023, 21:43:07 UTC
security_list_for_private_subnet-MySQL-VCN	Available	Mon, May 15, 2023, 15:18:41 UTC
Default Security List for MySQL-VCN	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 Security Lists in the (root) compartment. The 'Default Security List for MySQL-VCN' is highlighted.

Security Lists in (root) Compartment

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

Create Security List

Name	State	Created
asSecurityList	Available	Mon, May 15, 2023, 21:43:07 UTC
security_list_for_private_subnet-MySQL-VCN	Available	Mon, May 15, 2023, 15:18:41 UTC
Default Security List for MySQL-VCN	Available	Mon, May 15, 2023, 15:18:40 UTC

Showing 3 items < 1 of 1 >

46. Click **Add Ingress Rules**.

Oracle Cloud console showing the 'Default Security List for MySQL-VCN' page. The page includes a green hexagonal icon with 'SL' and 'AVAILABLE' text. Below it, there are buttons for 'Move resource', 'Add tags', and 'Terminate'. The 'Security List Information' section shows OCID: ...cw33fa and Compartment: (root). The 'Ingress Rules' section has a table with one rule: Stateless, Source: 0.0.0.0/0, IP Protocol: TCP, Source Port Range: All, Destination Port Range: 22, Type and Code: SSH Remote Login Protocol, Allows: TCP traffic for ports: 22 SSH Remote Login Protocol.

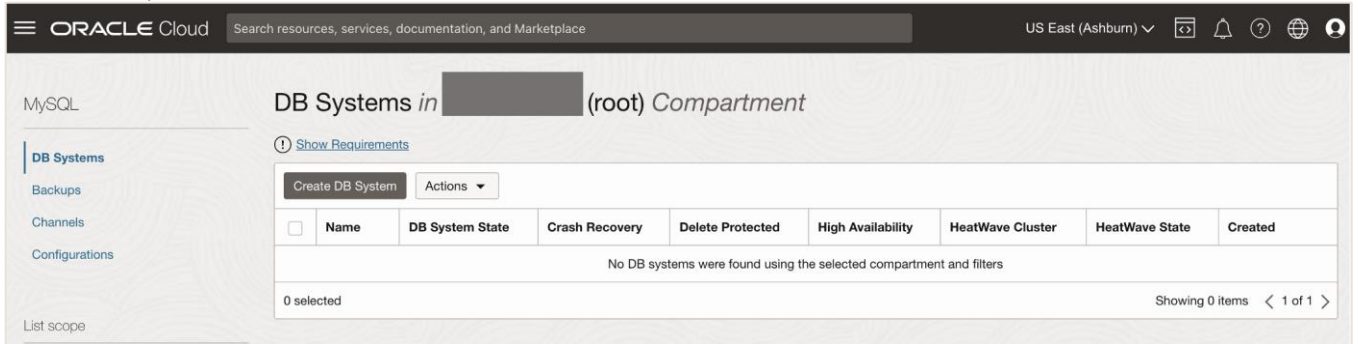
47. For **Source CIDR**, enter the **IPv4 CIDR Block** of your OCI VCN from step 34. For **Destination Port Range**, specify **3306,33060**. Leave everything as-is and click **Add Ingress Rules**.

Oracle Cloud console showing the 'Add Ingress Rules' dialog box. The dialog is titled 'Ingress Rule 1' and allows TCP traffic 3306,33060. It has a 'Stateless' checkbox. The 'Source Type' is set to 'CIDR' with a 'Source CIDR' of '10.0.0.0/16'. The 'IP Protocol' is set to 'TCP'. The 'Source Port Range' is 'All' and the 'Destination Port Range' is '3306,33060'. There is a 'Description' field and a '+ Another Ingress Rule' button.

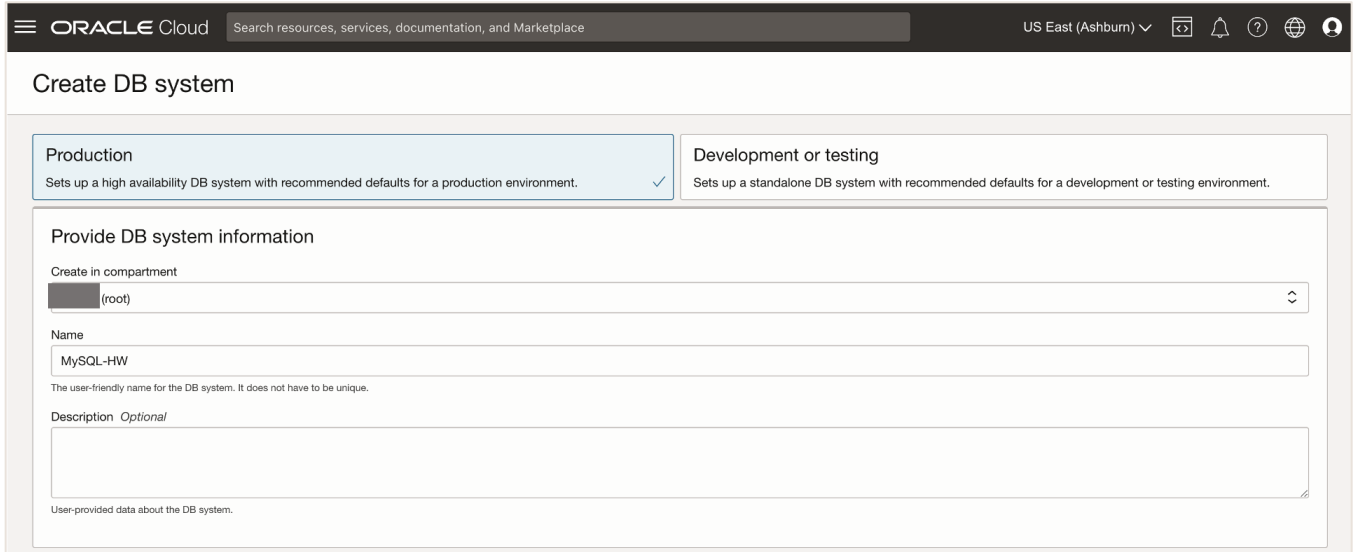
48. The VPN connection from on-premises to OCI is now set up. Make sure the OpenVPN client is started/running. We are now ready to perform the migration.

III) On OCI, create a HeatWave MySQL instance.

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



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 is a navigation bar with 'ORACLE Cloud', 'Cloud Classic >', a search bar, and the region 'US East (Ashburn)'. The main heading is 'Create DB System'. Below this, there are two tabs: 'Standalone' (selected, with a checkmark) and 'High availability'. The 'Standalone' tab is described as a 'Single-instance DB system'. Below the tabs, there is a section for 'Configure MySQL HeatWave' with a toggle switch turned 'ON'. The next section is 'Create administrator credentials', which includes three input fields: 'Username' (containing 'admin'), 'Password', and 'Confirm password'. The final section is 'Configure networking', which is currently collapsed. At the bottom, there are three buttons: 'Create', 'Save as stack', and 'Cancel'.

- 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.

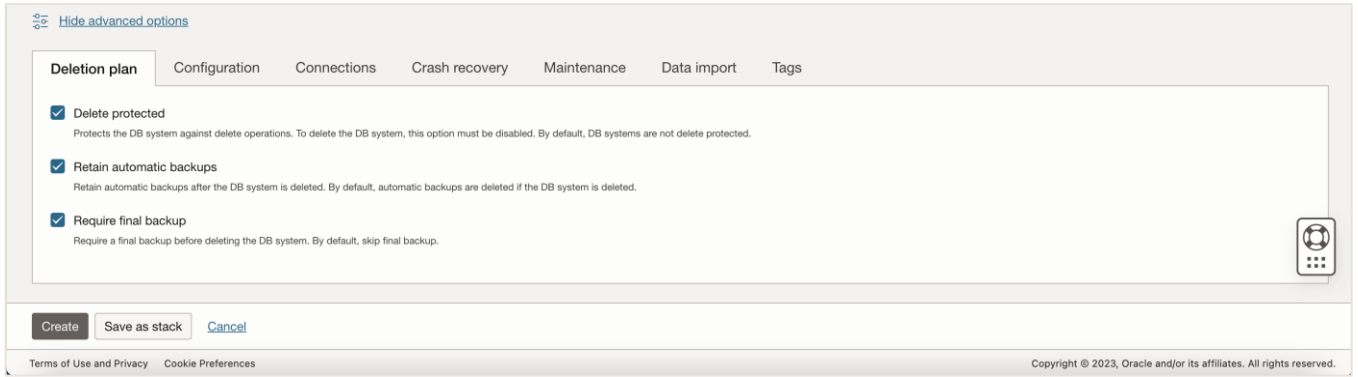
The screenshot shows the 'Create DB system' page in the Oracle Cloud console, focusing on the 'Configure networking' and 'Configure placement' sections. The 'Configure networking' section is expanded and shows the 'Virtual cloud network in' dropdown set to 'MySQL-VCN' and the 'Subnet in' dropdown set to 'private subnet-MySQL-VCN (Regional)'. The 'Configure placement' section is also expanded and shows three availability domain options: 'AD-1' (selected with a checkmark), 'AD-2', and 'AD-3'. Below these options, there is a checkbox for 'Choose a fault domain' which is currently unchecked. At the bottom, there are three buttons: 'Create', 'Save as stack', and 'Cancel'. The footer contains 'Terms of Use and Privacy', 'Cookie Preferences', and 'Copyright © 2023, Oracle and/or its affiliates. All rights reserved.'

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.

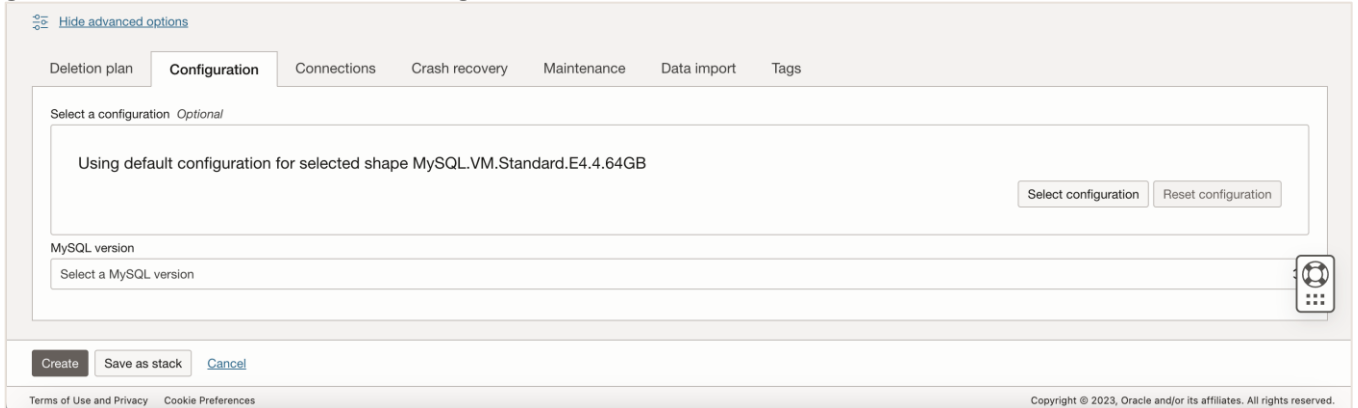
The screenshot shows the 'Create DB system' page in the Oracle Cloud console. The 'Configure hardware' section is active, showing the selected shape 'MySQL.HeatWave.VM.Standard' with 16 CPU core count, 512 GB memory size, and 16 Gbps max network bandwidth. A 'Change shape' button is visible. Below this, the 'Data storage size (GB)' is set to 1024. A note states: 'Storage allocated for data and log files. Storage size impacts IOPS and throughput. Data storage size must be an integer between 50 and 131,072.' The total IOPS is 76800 and total throughput is 600 MB. At the bottom of the section 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 'Create DB System' page in the Oracle Cloud console, with the 'Configure backup plan' section expanded. The 'Data storage size (GB)' is 1024, with a note: 'Storage allocated for data and log files. Storage size impacts IOPS and throughput. Data storage size must be an integer between 50 and 131,072.' The total IOPS is 76800 and total throughput is 600 MB. In the 'Configure backup plan' section, 'Enable automatic backups' is checked, with a note: 'Enables automatic backups. You must also specify a retention period, and select a backup window.' The 'Backup retention period' is set to 7 days, with a note: 'The retention period defines how long to store the backups, in days.' 'Enable point in time restore' is also checked, with a note: 'Enables you to restore from a DB system at a point in time.' 'Select backup window' is unchecked, with a note: 'The backup window start time defines the start of the time period during which your DB system is backed up.' At the bottom of the section is a 'Show advanced options' link. At the bottom of the page 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.



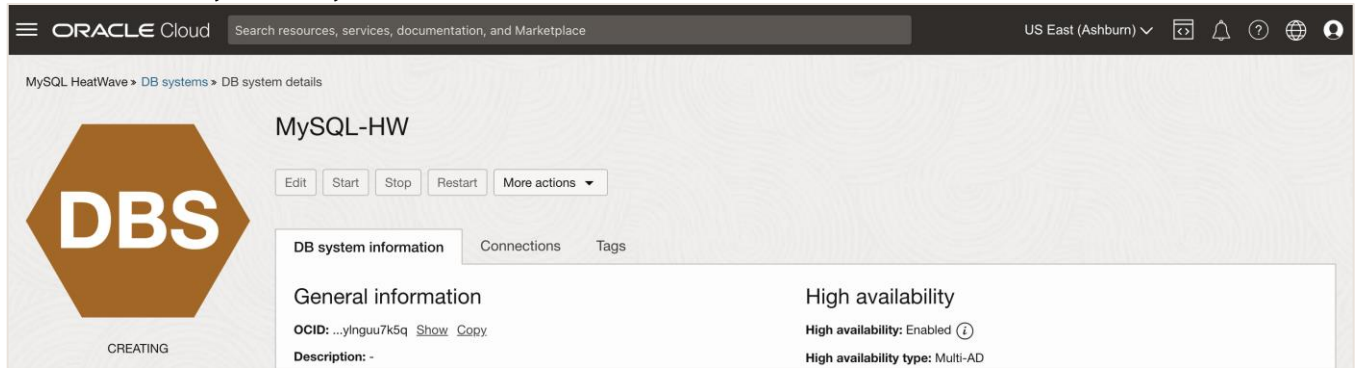
56. For **MySQL version**, choose either **Innovation** or **Bug fix**. With the new MySQL versioning model, you have the flexibility to select an innovation or a bug fix release. Both the releases are production-grade quality. MySQL innovation releases allow you to access the latest features and improvements. 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's 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 the HeatWave MySQL DB system creation process. The 'MySQL version' dropdown menu is open, displaying a list of versions. The version '8.0.34 - Bug fix' is selected and highlighted. Other versions listed include '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 configuration area.

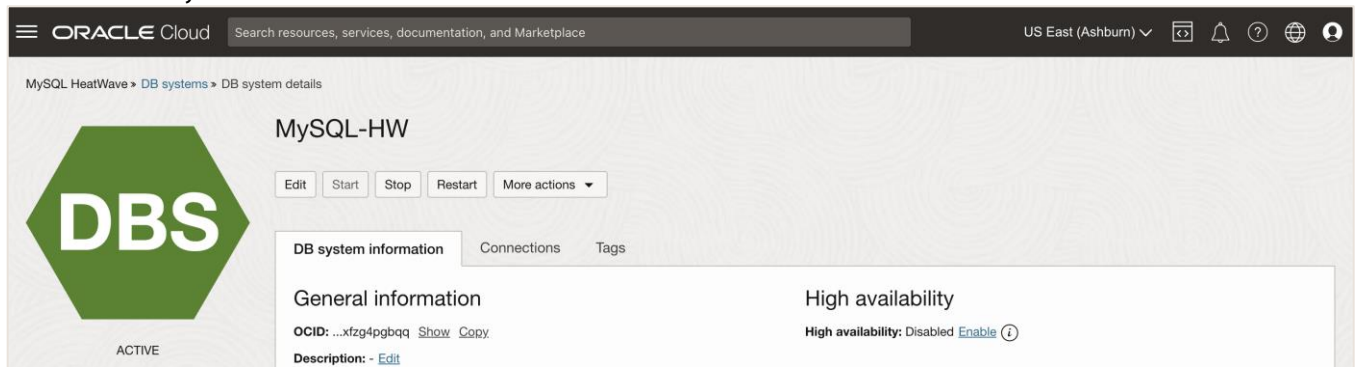
57. Click **Create** to finish the HeatWave MySQL DB system creation process.

The screenshot shows the 'Configuration' tab of the HeatWave MySQL DB system creation process. The 'MySQL version' dropdown menu is closed, and the selected option '8.0.34 - Bug fix' is visible. The 'Create' button is visible at the bottom left of the configuration area.

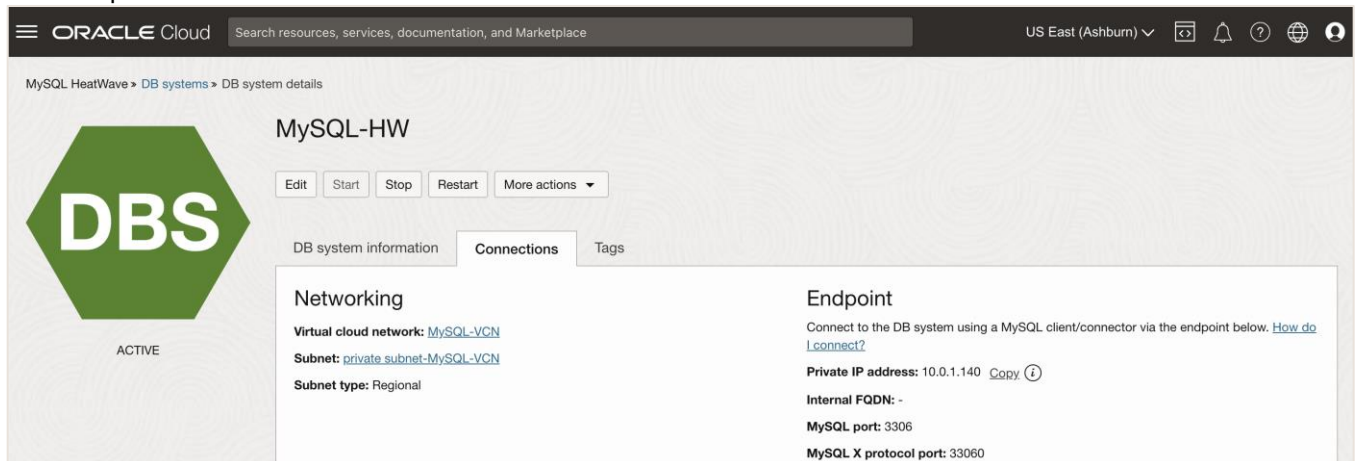
58. Your HeatWave MySQL DB system will start **CREATING**.



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 8.1 (or above) on an on-premises instance that can connect to your on-premises MySQL.

61. Have an on-premises instance that can connect to your on-premises MySQL. Go to the below website and download MySQL Shell 8.1 on your on-premises instance. For this guide, we have deployed our on-premises MySQL on a Linux instance. From the MySQL Shell download page, ensure **8.1.x Innovation** is selected under **Select Version**. MySQL Shell 8.1 is fully compatible with MySQL 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/>

MySQL Community Downloads

< MySQL Shell

General Availability (GA) Releases Archives ↓

MySQL Shell 8.1.1 Innovation

Select Version:
8.1.1 Innovation

Select Operating System:
Red Hat Enterprise Linux / Oracle Linux

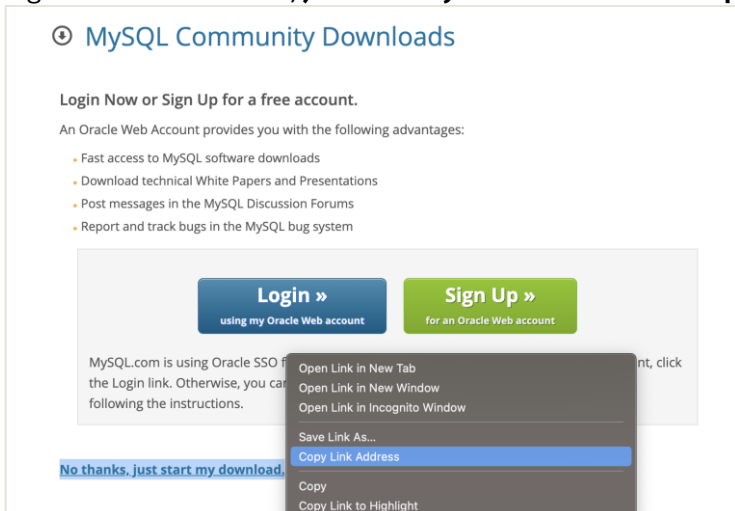
Select OS Version:
Red Hat Enterprise Linux 8 / Oracle Linux 8 (x86, 64-bit)

RPM Package <small>(mysql-shell-8.1.1-1.el8.x86_64.rpm)</small>	8.1.1	27.5M	Download
RPM Package, Debug Information <small>(mysql-shell-debuginfo-8.1.1-1.el8.x86_64.rpm)</small>	8.1.1	494.1M	Download

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

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 MySQL and execute the below command to download MySQL Shell:

```
$ wget <MySQL-Shell-Download-Link>
```

Replace the link with what you have.

```
$ wget https://dev.mysql.com/get/Downloads/MySQL-Shell/mysql-shell-8.1.1-1.e18.x86_64.rpm
```

```
[lopc@linux-8 ~]$ wget https://dev.mysql.com/get/Downloads/MySQL-Shell/mysql-shell-8.1.1-1.e18.x86_64.rpm
--2023-08-15 19:29:50-- https://dev.mysql.com/get/Downloads/MySQL-Shell/mysql-shell-8.1.1-1.e18.x86_64.rpm
Resolving dev.mysql.com (dev.mysql.com)... 23.49.176.249, 2600:1408:c400:188c::2e31, 2600:1408:c400:1881::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.49.176.249|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://cdn.mysql.com//Downloads/MySQL-Shell/mysql-shell-8.1.1-1.e18.x86_64.rpm [following]
--2023-08-15 19:29:51-- https://cdn.mysql.com//Downloads/MySQL-Shell/mysql-shell-8.1.1-1.e18.x86_64.rpm
Resolving cdn.mysql.com (cdn.mysql.com)... 23.219.8.226, 2600:1408:c400:1884::1d68, 2600:1408:c400:188d::1d68
Connecting to cdn.mysql.com (cdn.mysql.com)|23.219.8.226|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 28857020 (28M) [application/x-redhat-package-manager]
Saving to: 'mysql-shell-8.1.1-1.e18.x86_64.rpm'

mysql-shell-8.1.1-1.e 100%[=====>] 27.52M 72.7MB/s in 0.4s

2023-08-15 19:29:51 (72.7 MB/s) - 'mysql-shell-8.1.1-1.e18.x86_64.rpm' saved [28857020/28857020]
```

64. After downloading the MySQL Shell rpm, install MySQL Shell:

```
$ sudo yum localinstall mysql-shell*
```

```
[opc@linux-8 ~]$ sudo yum localinstall mysql-shell*
Last metadata expiration check: 0:00:45 ago on Tue 15 Aug 2023 07:27:54 PM GMT.
Dependencies resolved.
=====
Package                Arch    Version                                Repository    Size
=====
Installing:
mysql-shell            x86_64  8.1.1-1.el8                            @commandline 28 M
Installing dependencies:
python39-libs          x86_64  3.9.16-1.module+el8.8.0+21116+ee8c18cf.1 ol8_appstream 8.2 M
python39-pip-wheel     noarch  20.2.4-7.module+el8.6.0+20625+ee813db2 ol8_appstream 1.1 M
python39-setuptools-wheel
noarch  50.3.2-4.module+el8.5.0+20364+c7fe1181 ol8_appstream 497 k
Installing weak dependencies:
python39               x86_64  3.9.16-1.module+el8.8.0+21116+ee8c18cf.1 ol8_appstream 33 k
python39-pip           noarch  20.2.4-7.module+el8.6.0+20625+ee813db2 ol8_appstream 1.9 M
python39-setuptools    noarch  50.3.2-4.module+el8.5.0+20364+c7fe1181 ol8_appstream 871 k
Enabling module streams:
python39                3.9
=====
```

65. 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 8.1.1 for Linux on x86_64 - for MySQL 8.1.0 (MySQL Community Server (GPL))
[opc@linux-8 ~]$
```

66. To login to your on-premises MySQL using MySQL Shell, use the below commands:

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

-OR-

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

```
[opc@linux-8 ~]$ mysqlsh root@localhost:3306
Please provide the password for 'root@localhost:3306': *****
Save password for 'root@localhost:3306'? [Y]es/[N]o/Ne[v]er (default No):
MySQL Shell 8.1.1

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 session to 'root@localhost:3306'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 12
Server version: 8.0.33 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL localhost:3306 ssl 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 on-premises MySQL using MySQL Shell. Afterwards, execute the MySQL Shell's `util.copyInstance()` utility to export all schemas (including users, indexes, routines, triggers) from on-premises MySQL to the HeatWave MySQL on OCI.

67. Before connecting to on-premises MySQL 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 &`

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

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

-OR-

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

```
[opc@linux-8 ~]$ tmux
[opc@linux-8 ~]$ mysqlsh root@localhost:3306
Please provide the password for 'root@localhost:3306': *****
Save password for 'root@localhost:3306'? [Y]es/[N]o/[Ne[v]er (default No):
MySQL Shell 8.1.1

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 session to 'root@localhost:3306'
Fetching schema names for auto-completion... Press ^C to stop.
Your MySQL connection id is 12
Server version: 8.0.33 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL localhost:3306 ssl JS >
```

69. Change to the JavaScript mode of MySQL Shell and run the `util.copyInstance()` utility to export all on-premises MySQL data into OCI HeatWave MySQL.

```
MySQL JS> \js
MySQL JS> util.copyInstance('mysql://admin@10.0.1.140', {"compatibility":
["force_innodb", "skip_invalid_accounts", "strip_definers",
"strip_restricted_grants", "strip_tablespaces", "ignore_wildcard_grants",
"strip_invalid_grants", "create_invisible_pks"], users: "true", threads: 4,
dryRun:"true"})
```

Note: replace the username (`admin`) and IP address (`10.0.1.140`) with your HeatWave MySQL username and IP address (not the on-premises MySQL username and IP address).

```
MySQL localhost:33060+ ssl JS > util.copyInstance('mysql://admin@10.0.1.140', {"compatibility": ["force_innodb", "skip_invalid_accounts", "strip_definers", "strip_restricted_grants", "strip_tablespaces", "ignore_wildcard_grants", "strip_invalid_grants", "create_invisible_pks"], users: "true", threads: 4, dryRun:"true"})
Copying DDL, Data and Users from in-memory FS, source: linux-8:3306, target: wxchfcbv4fxym5b6:3306.
SRC: dryRun enabled, no locks will be acquired and no files will be created.
SRC: Acquiring global read lock
SRC: Global read lock acquired
Initializing - done
SRC: 1 out of 5 schemas will be dumped and within them 3 tables, 0 views.
SRC: 1 out of 4 users will be dumped.
Gathering information - done
SRC: All transactions have been started
SRC: Locking instance for backup
SRC: Global read lock has been released
SRC: Checking for compatibility with MySQL Database Service 8.1.1
```

[... output truncated]

```
TGT: Executing view DDL...
TGT: Executing view DDL - done
TGT: Loading data...
TGT: Recreating indexes...
TGT: Starting data load
TGT: Waiting for more data to become available...
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 accounts were loaded
TGT: 0 warnings were reported during the load.
---
Dump_metadata:
  Binlog_file: binlog.000001
  Binlog_position: 735682
  Executed_GTID_set: ''
MySQL localhost:33060+ ssl JS >
```


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 the 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_tablespace`: 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 which 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.

70. Once you have run the command in step 69 and did not see any errors in the output (warnings are okay), run the same step 69 command but this time change the `dryRun` option to `false`.

```
MySQL JS> util.copyInstance('mysql://admin@10.0.1.140', {"compatibility":
["force_innodb", "skip_invalid_accounts", "strip_definers",
"strip_restricted_grants", "strip_tablespaces", "ignore_wildcard_grants",
"strip_invalid_grants", "create_invisible_pks"], users: "true", threads: 4,
dryRun:"false"})
```

Note: replace the username (`admin`) and IP address (`10.0.1.140`) with your HeatWave MySQL username and IP address (not the on-premises MySQL username and IP address).

```
MySQL localhost:33060+ ssl JS > util.copyInstance('mysql://admin@10.0.1.140', {"compa
tibility": ["force_innodb", "skip_invalid_accounts", "strip_definers", "strip_restricted
_grants", "strip_tablespaces", "ignore_wildcard_grants", "strip_invalid_grants", "create
_invisible_pks"], users: "true", threads: 4, dryRun:"false"})
Copying DDL, Data and Users from in-memory FS, source: linux-8:3306, target: wxchfcbv4fx
ym5b6:3306.
SRC: Acquiring global read lock
SRC: Global read lock acquired
Initializing - done
SRC: 1 out of 5 schemas will be dumped and within them 3 tables, 0 views.
SRC: 1 out of 4 users will be dumped.
Gathering information - done
SRC: All transactions have been started
SRC: Locking instance for backup
SRC: Global read lock has been released
SRC: Checking for compatibility with MySQL Database Service 8.1.1
```

[... 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: 1
SRC: Tables dumped: 3
SRC: Data size: 194.62 KB
SRC: Rows written: 5302
SRC: Bytes written: 194.62 KB
SRC: Average throughput: 194.62 KB/s
TGT: Executing common postamble SQL
100% (194.62 KB / 194.62 KB), 38.91 KB/s, 3 / 3 tables done
Recreating indexes - done
TGT: 3 chunks (5.30K rows, 194.62 KB) for 3 tables in 1 schemas were loaded in 5 sec (av
g throughput 38.45 KB/s)
TGT: 1 accounts were loaded
TGT: 0 warnings were reported during the load.

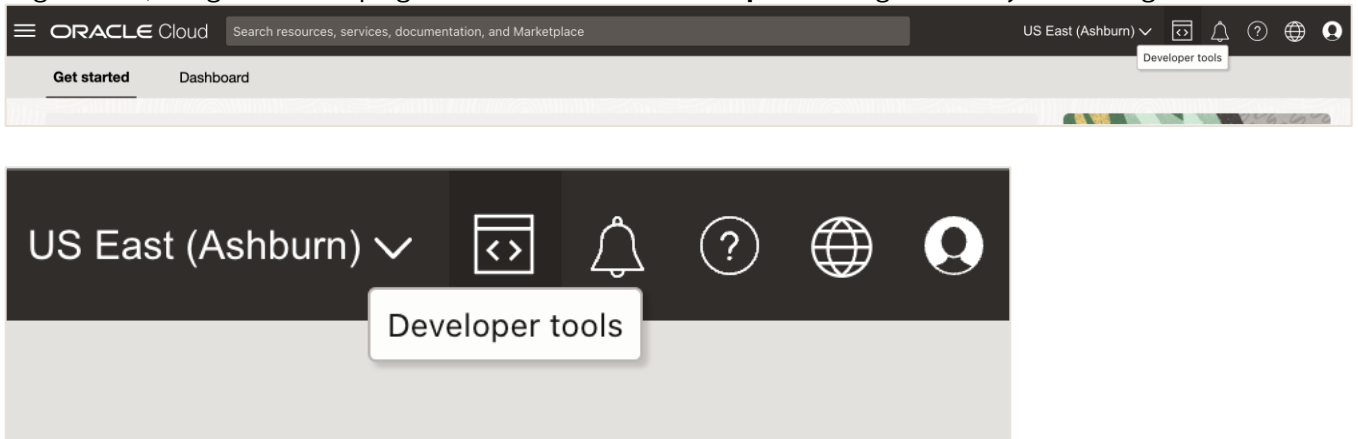
---
Dump_metadata:
  Binlog_file: binlog.000001
  Binlog_position: 735682
  Executed_GTID_set: ''

MySQL localhost:33060+ ssl JS >
```

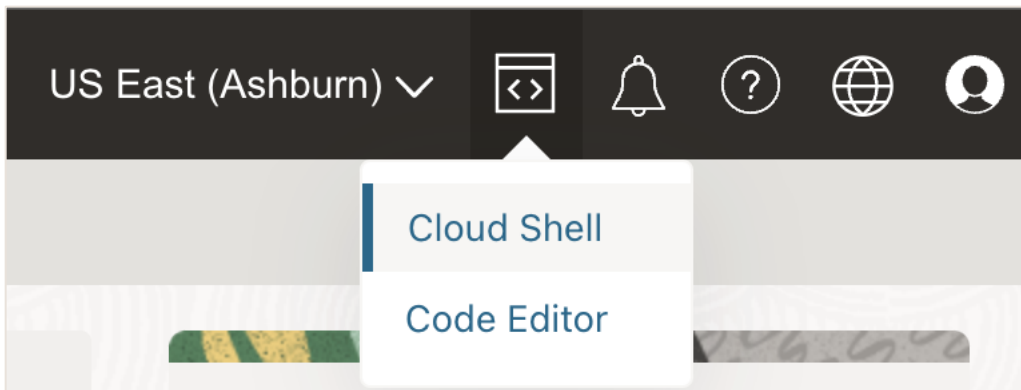
Note: once the MySQL Shell copy utility finishes, all your data will be copied over from on-premises MySQL to OCI HeatWave MySQL. This completes the migration process. You can end your tmux session.

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

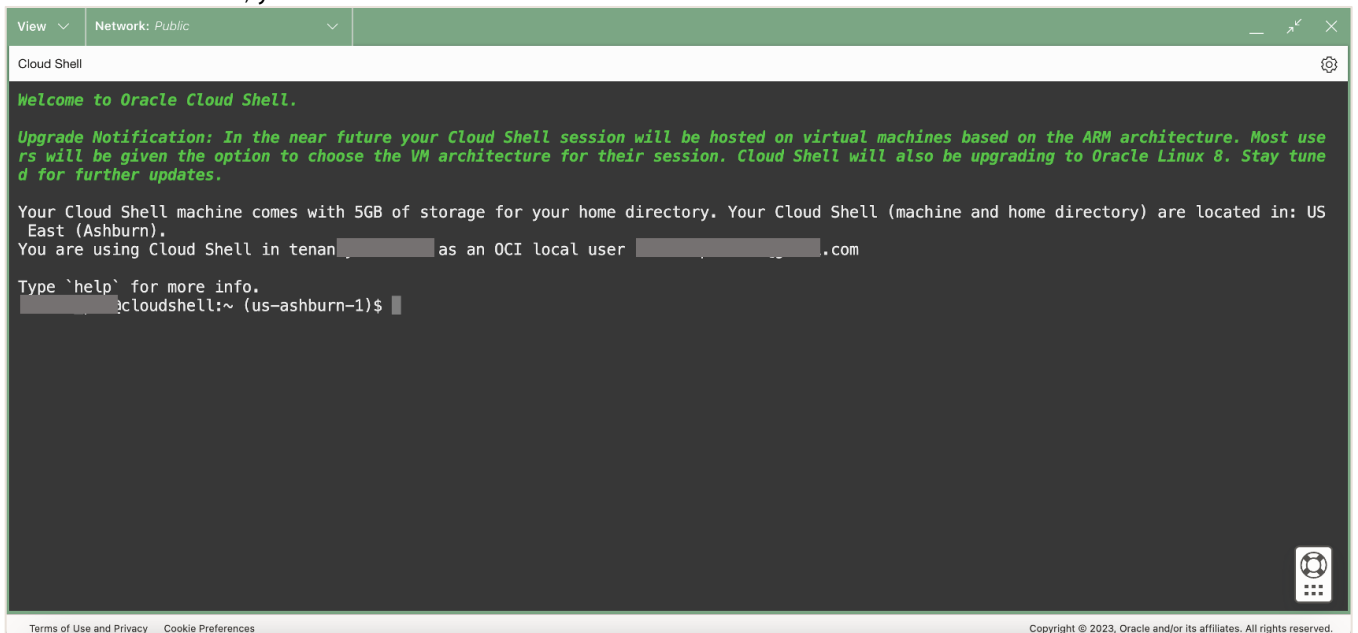
71. Login to [OCI](#), navigate to the top right corner and click on **Developer tools** right next to your OCI Region.



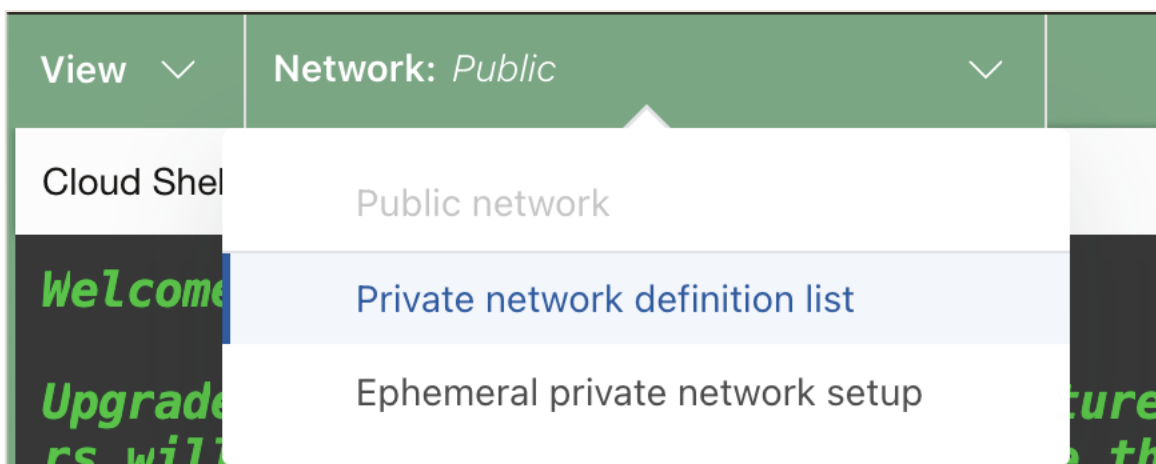
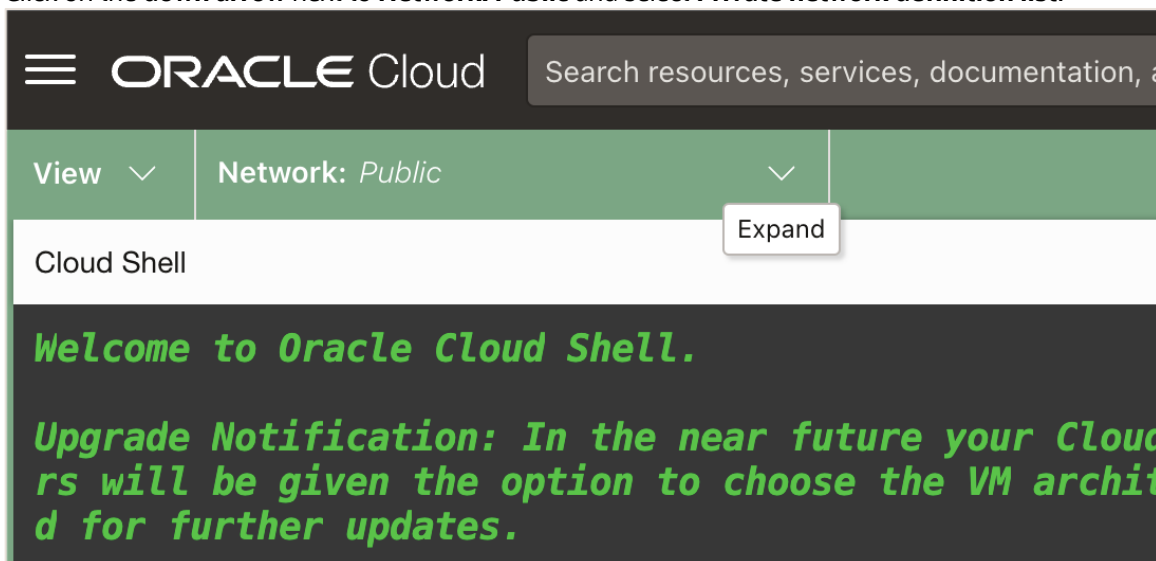
72. Click **Cloud Shell**.



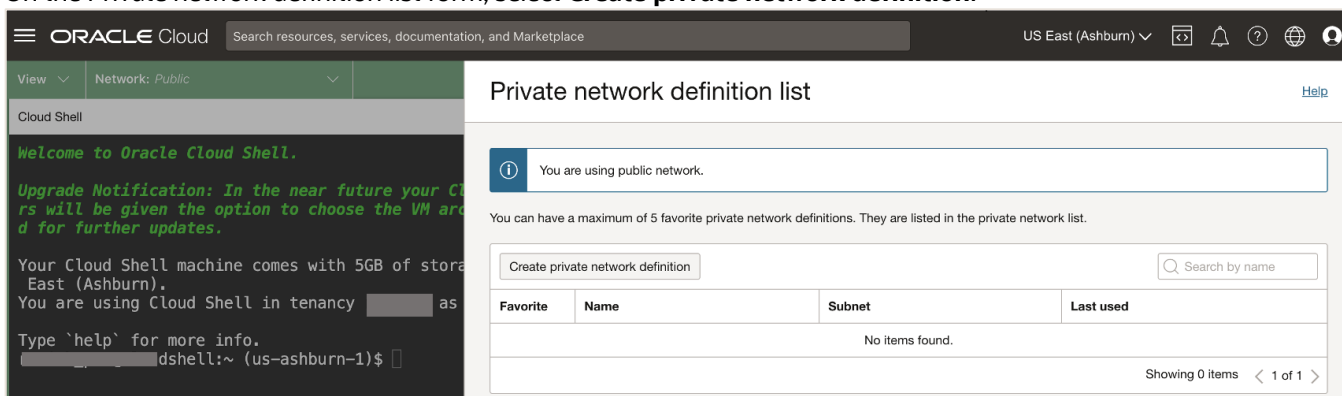
73. Within a few minutes, you will be connected to the OCI Cloud Shell like below:



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



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



76. 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

Use as active network

[Cancel](#)

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

77. 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.
(root)@cloudshell:~ (us-ashburn-1)$
```

Private network definition list [Help](#)

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.

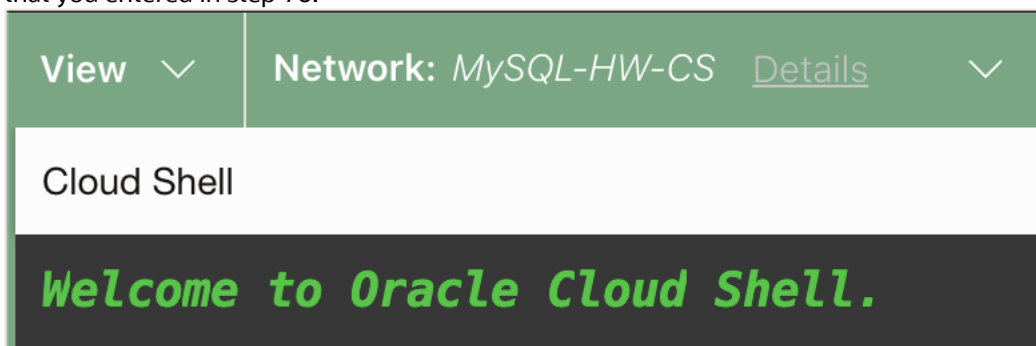
Favorite	Name	Subnet	Last used
<input checked="" type="checkbox"/>	MySQL-HW-CS	...yciry4ma Show Copy	-

Showing 1 item < 1 of 1 >

Default network
Select default network description
Public

Terms of Use and Privacy Cookie Preferences Copyright © 2023, Oracle and/or its affiliates. All rights reserved.

78. 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.



79. 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@10.0.1.140
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 >
```

80. 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 10.0.1.140:33060+ 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 > SHOW SCHEMAS;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| world |
+-----+
5 rows in set (0.0014 sec)
MySQL 10.0.1.140:33060+ ssl SQL > SHOW TABLES IN world;
+-----+
| Tables_in_world |
+-----+
| city |
| country |
| countrylanguage |
+-----+
3 rows in set (0.0021 sec)
MySQL 10.0.1.140:33060+ ssl SQL >
```

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

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

82. Here is our row count comparison for on-premises MySQL and OCI HeatWave MySQL:

On-premises MySQL 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 >
```

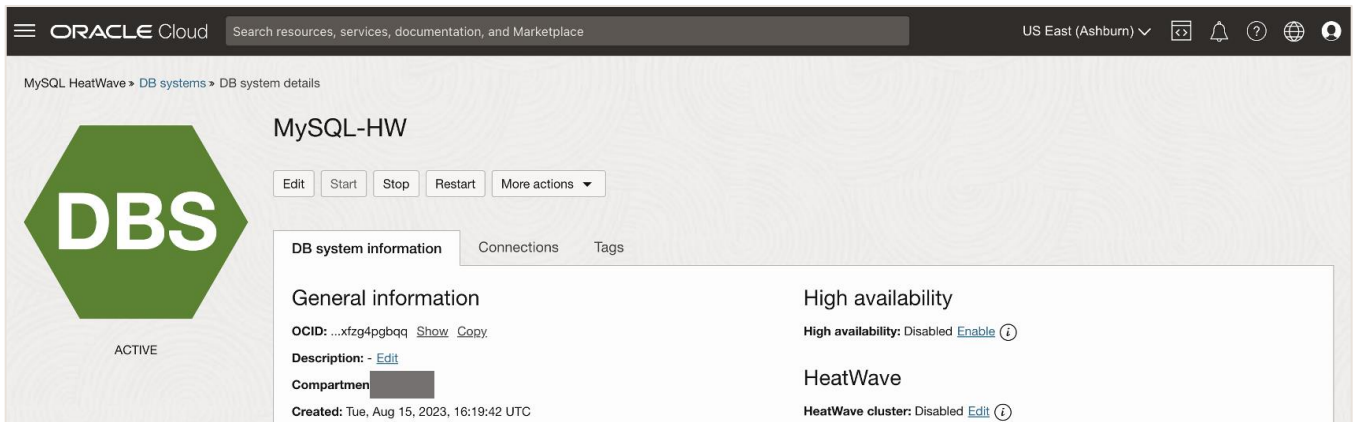
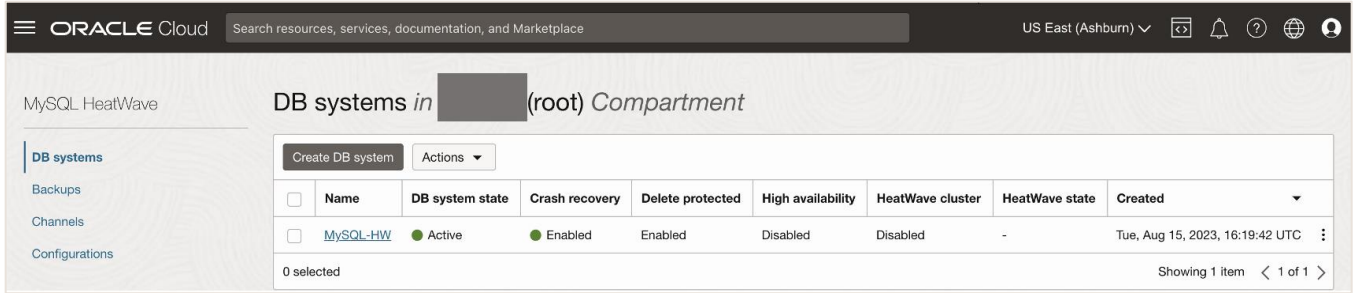
OCI HeatWave MySQL row count:

```
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 >
```

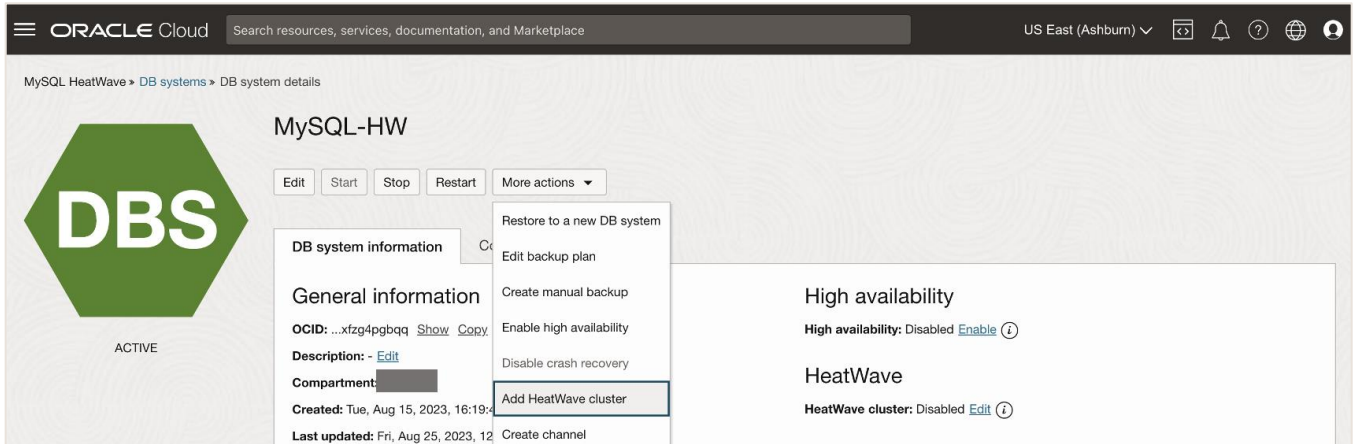
83. After validating, you can have your application/s point to the new OCI HeatWave MySQL instance.

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.

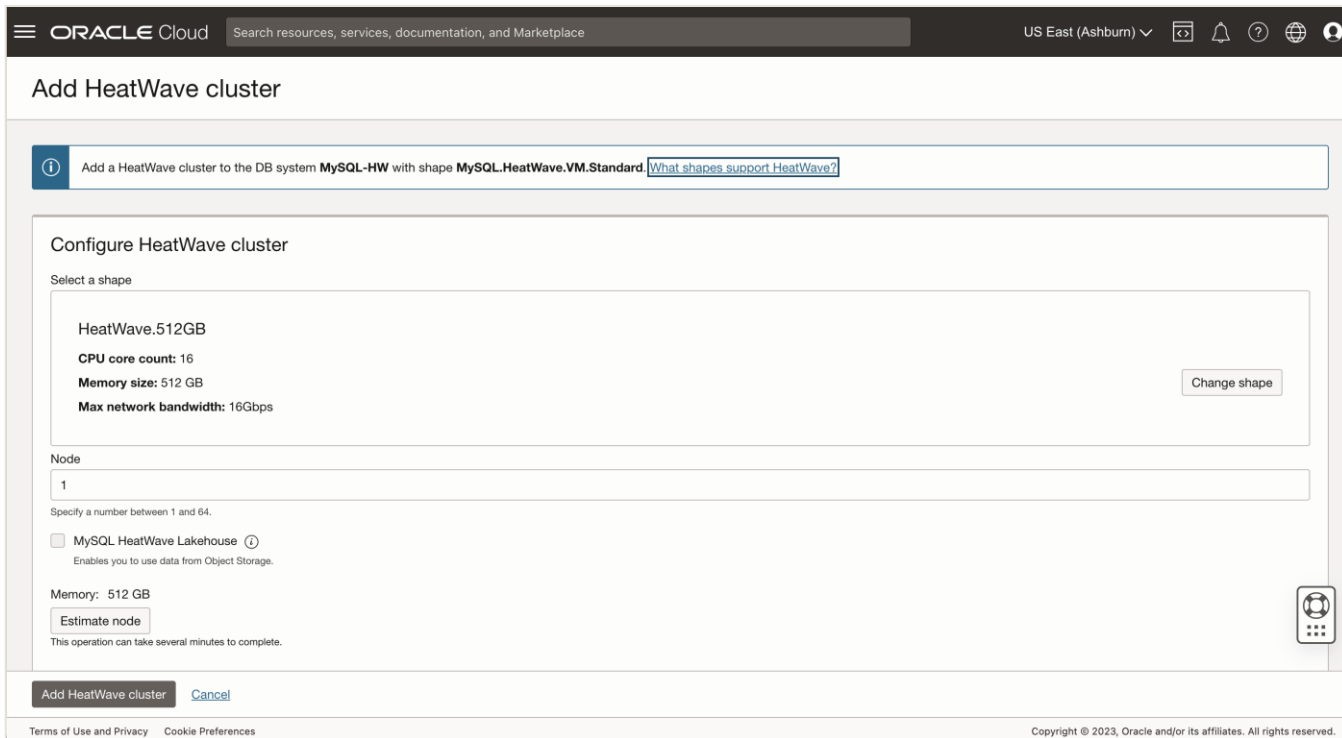
- 84. Login to [OCI](#). Click on the navigation menu, go to **Databases**, and click **HeatWave MySQL**.
- 85. Click on the name of your HeatWave MySQL instance to go to the **DB System Details** page.



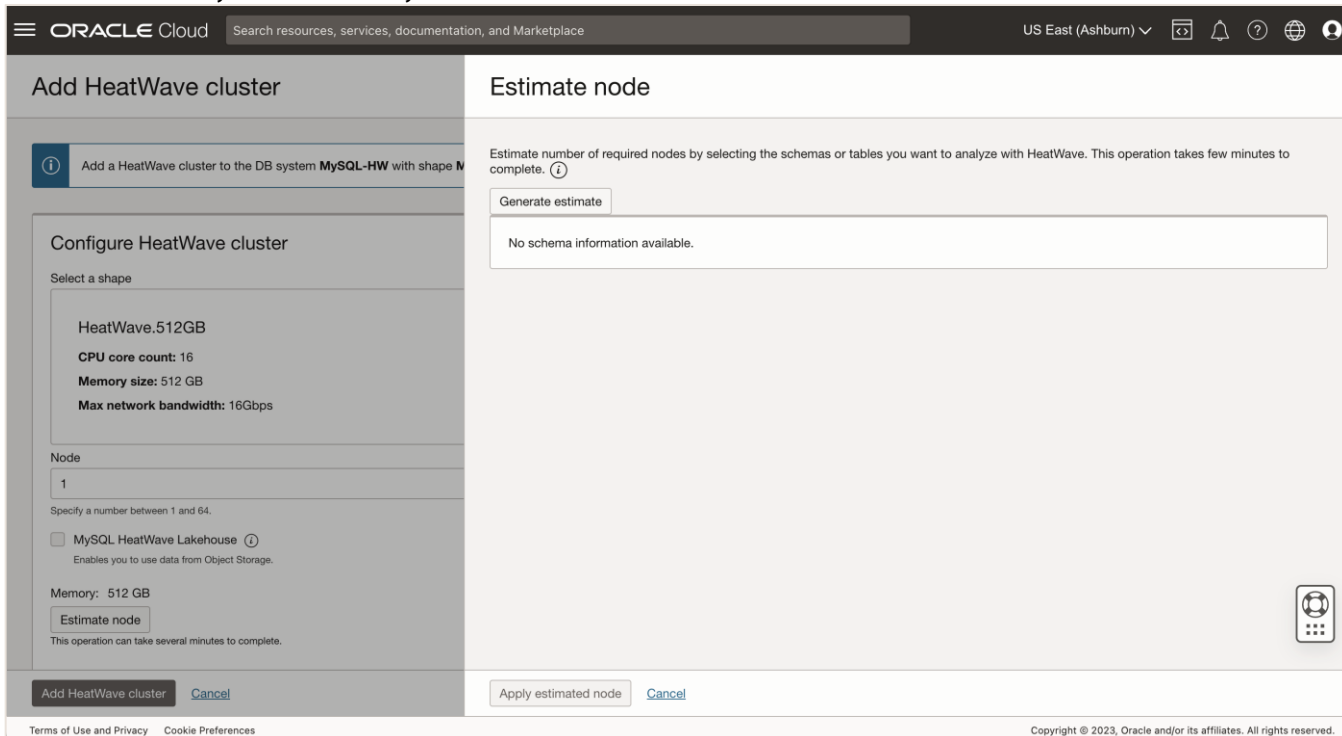
- 86. Click **More actions** and click **Add HeatWave cluster**.



87. Click **Estimate node**.



88. 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.



89. 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 for configuring a HeatWave cluster. The left pane is titled 'Add HeatWave cluster' and shows configuration options for a HeatWave.512GB shape with 16 CPU cores and 512 GB of memory. The right pane is titled 'Estimate node' and contains a table with the following data:

Name	Memory estimate	Information
<input type="checkbox"/> world	9 MB	Number of tables: 3

Below the table, the 'Total memory selected' is 0 Bytes. The summary section states: 'No schema or table selected. Select the schemas and tables to use for the node estimate.'

90. 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**.

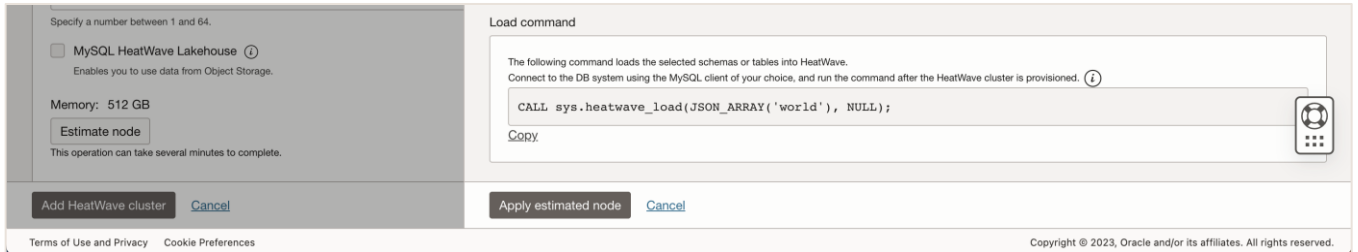
The screenshot shows the Oracle Cloud console interface for configuring a HeatWave cluster. The left pane is the same as in the previous screenshot. The right pane is titled 'Estimate node' and shows the 'world' schema selected in the table:

Name	Memory estimate	Information
<input checked="" type="checkbox"/> world	9 MB	Number of tables: 3

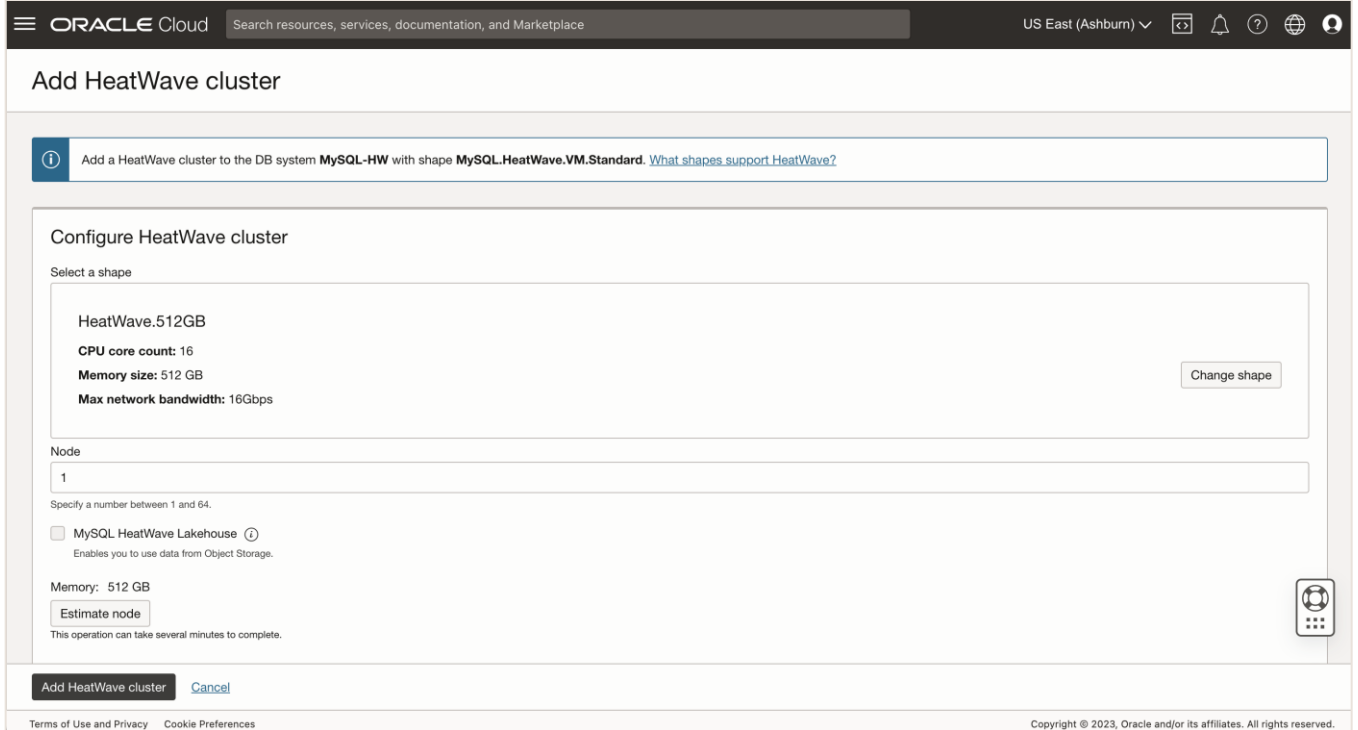
The 'Total memory selected' is now 9 MB. The summary section shows the following details:

- 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

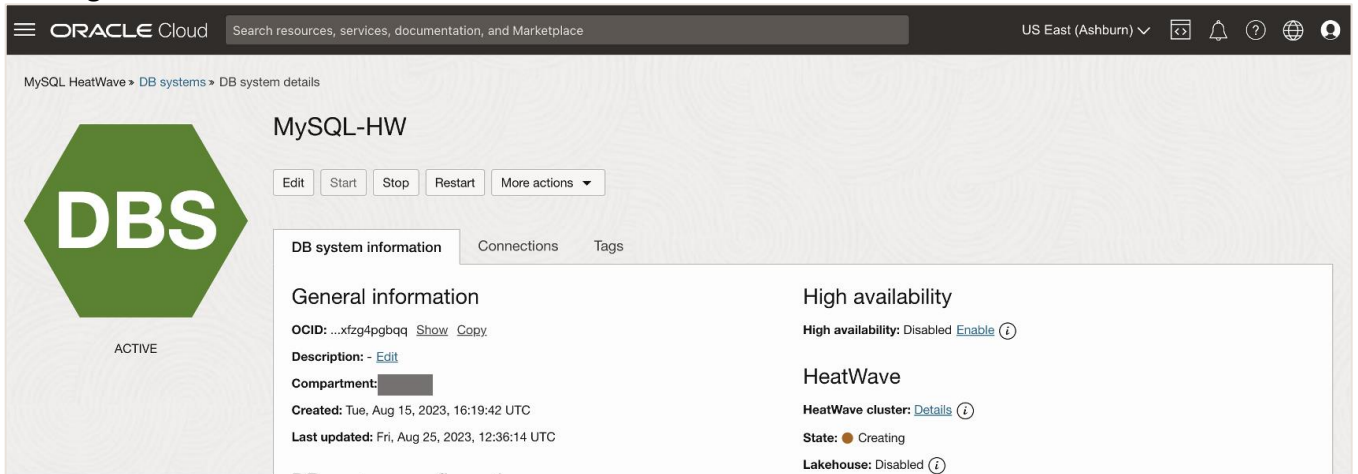
At the bottom of the right pane, a 'Load command' section is visible, and the 'Apply estimated node' button is highlighted.



91. 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.



92. 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: ...xfzg4pgbqq 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

93. 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 >

```

94. 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 >
```

95. 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

 facebook.com/oracle

 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.