

Deploy the Oracle Cloud and Microsoft Azure Interconnect Using Hub-and-Spoke Topology

Access Oracle Cloud Infrastructure Resources from
Microsoft Azure over a Private Interconnect

February 2022, version 1.1
Copyright © 2022, Oracle and/or its affiliates
Public

Purpose Statement

This document provides an overview of the capabilities enabled by the cross-cloud direct interconnection between Oracle Cloud Infrastructure and Microsoft Azure Interconnect. It's solely intended to help you assess the business benefits of migrating workloads to Oracle Cloud Infrastructure and plan your IT projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document since its initial publication.

| DATE | REVISION |
|----------------|------------------------------------|
| February 2022 | Updated template, copy-edited text |
| September 2020 | Initial publication |

Table of Contents

| | |
|--|-----------|
| Introduction | 4 |
| Solution Summary | 4 |
| Prerequisites and Considerations | 5 |
| Deploying the Topology | 5 |
| Step 1: Set Up the Hub VCN | 5 |
| Step 2: Set Up the Spoke VCN | 13 |
| Step 3: Set Up the Azure Environment | 21 |
| Step 4: Connect Oracle Cloud Infrastructure and Azure Using the Private Interconnect | 26 |
| Step 5: Access Autonomous Database from Azure Through the Private Interconnect | 33 |
| Conclusion | 35 |
| References | 35 |

Introduction

The partnership between Oracle and Microsoft provides cloud interoperability that enables customers to distribute mission-critical enterprise workloads between the two cloud providers. This interoperability provides many opportunities for customers to deploy their applications across Oracle Cloud Infrastructure (OCI) and Microsoft Azure simultaneously. A highly redundant, low-latency, high-bandwidth private connection provides a cross-cloud direct interconnection between OCI and Azure in various regions across the globe. This interconnection is achieved by connecting OCI FastConnect directly with Azure ExpressRoute without an intermediary network provider.

In a *hub-and-spoke* model, the *hub* is a virtual network that serves as a central location for managing external connectivity either to on-premises resources or other cloud vendors. The *spokes* are virtual networks that host applications and provide connectivity through a central hub by way of local peering gateways. Traffic passing through the hub can be routed, inspected, and centrally managed according to rules and processes.

Companies are increasingly using the concept of *transit routing* to isolate and compartmentalize their resources into dedicated spoke virtual networks. This configuration makes it easy to comply with governance and security guidelines because each spoke might have access to different levels of services and access controls.

A dynamic routing gateway (DRG) on OCI has a one-to-one relationship with a virtual network gateway (VNet gateway) on Azure. If a customer is managing multiple virtual cloud networks (VCNs) in OCI, then the only way to connect all the VCNs to Azure is to deploy multiple interconnects to establish the application connectivity between Azure and OCI. This setup soon becomes expensive and hard to manage. With transit routing, however, instead of multiple VCNs being managed separately, you can architect them in a hub-and-spoke topology. This topology provides a single source of connection to Azure.

This document describes how to distribute a typical application workload in OCI in a hub-and-spoke topology and how to access it from Azure through the private interconnect.

Solution Summary

The example in this document uses an Oracle Autonomous Database that runs on shared Oracle Exadata infrastructure. The database resides in the private subnet of a spoke VCN and is accessed from Azure over a private interconnect using the private IP endpoint of the database. The connectivity to the spoke VCN is established by enabling local peering between the hub and spoke VCN.

The hub VCN in OCI is connected to Azure through a private interconnect that uses ExpressRoute and FastConnect. The traffic between OCI and Azure is directed by a DRG on OCI and a VNet gateway on Azure.

You can provision a bastion server in the public subnet of the hub VCN to connect to other sources that are deployed in the spoke VCN. Also, you can deploy more application servers in the public subnet of a spoke VCN that need connectivity to the database.

The setup is highly scalable. You can create more spoke VCNs later and connect to Azure through the hub VCN, using the existing private interconnect. Figure 1 illustrates this hub-and-spoke setup.

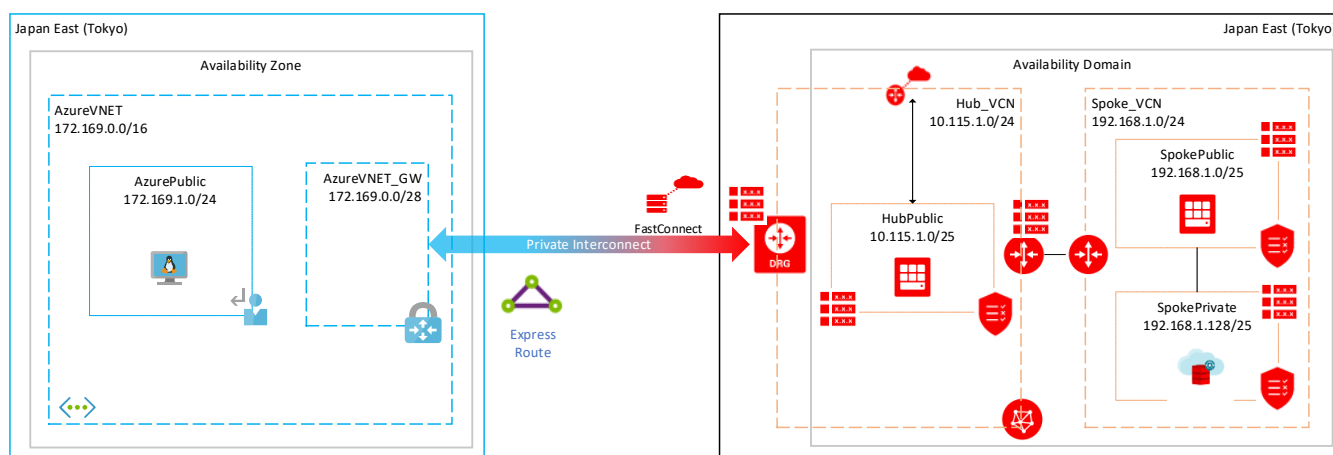


Figure 1: Example Hub-and-Spoke Topology Between Oracle Cloud Infrastructure and Microsoft Azure

Prerequisites and Considerations

Before you start to create resources, note the following prerequisites and considerations:

- Select the region for the setup carefully. OCI and Azure provide the private interconnect facility for various regions across the globe, and the list is always growing. Select a region that offers the option of connecting OCI and Azure through the private interconnect. For a complete list of regions that support this connection, see [Access to Microsoft Azure](#).
- Carefully plan your network and always use nonoverlapping CIDR blocks across OCI and Azure.
- Identify a pair of /30 CIDR blocks for the border gateway protocol (BGP) IP addresses to use for the two redundant BGP sessions between Oracle and Azure.
- Because you can add spoke VCNs after the initial setup, when the demand grows to host more resources, the configuration can become complex to manage and troubleshoot. Plan for future growth.

Deploying the Topology

This section provides the detailed steps for setting up a hub-and-spoke topology in OCI and accessing it from Azure through the private interconnect.

Step 1: Set Up the Hub VCN

The first step is to create and configure a hub VCN. The hub VCN is connected to a spoke VCN in OCI and to Azure through the private interconnect.

Select a region where the interconnect is offered. This example uses the Japan East region. For a complete list of connected regions, see [Access to Microsoft Azure](#).

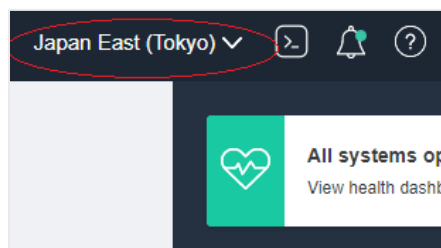


Figure 2: Select a Region for the Interconnect

1. In the Oracle Cloud Console navigation menu, select **Networking** and then click **Virtual Cloud Networks**.
2. Click **Create VCN**.
3. Enter a name for the VCN, such as Hub_VCN, and specify the compartment to create it in, such as TransitRouting.
4. Specify a CIDR address block for the hub VCN, ensuring that it doesn't overlap with the spoke VCN or the Azure VNet.

The screenshot shows the 'Create a Virtual Cloud Network' form. At the top, there's a title 'Create a Virtual Cloud Network' and a 'Help' link. The form has several sections: 'NAME' with a text input containing 'Hub_VCN'; 'CREATE IN COMPARTMENT' with a dropdown menu showing 'TransitRouting' and a subtext 'intoracleeli (root)/TransitRouting'; 'CIDR BLOCK' with a text input containing '10.115.1.0/24', an example '10.0.0.0/16', and a note about overlapping CIDRs with a 'Learn more' link; 'DNS RESOLUTION' with a checked checkbox 'USE DNS HOSTNAMES IN THIS VCN' and a note about instance hostname assignment with a 'Learn more' link; 'DNS LABEL' with a text input containing 'HubVCN' and a note 'Only letters and numbers, starting with a letter. 15 characters max.'; and 'DNS DOMAIN NAME' with a read-only text input containing 'HubVCN.oraclevcn.com'. At the bottom, there are 'Show Advanced Options' and 'Create VCN' (highlighted in blue) and 'Cancel' buttons.

Figure 3: Create the Hub VCN

5. Click **Create VCN**.
6. Create a local peering gateway, such as HubLPG, for the hub VCN.

This gateway connects the hub VCN with the spoke VCN. For instructions, see [To create a local peering gateway](#).

The screenshot shows the 'Create Local Peering Gateway' form. At the top, there's a title 'Create Local Peering Gateway' and 'Help' and 'Cancel' links. The form has two main sections: 'NAME' with a text input containing 'HubLPG'; and 'CREATE IN COMPARTMENT' with a dropdown menu showing 'TransitRouting' and a subtext 'intoracleeli (root)/TransitRouting'. At the bottom, there are 'Show Advanced Options' and 'Create Local Peering Gateway' (highlighted in blue) and 'Cancel' buttons.

Figure 4: Create the Local Peering Gateway for the Hub VCN

7. Create an internet gateway, such as HubIGW, for the hub VCN.

An internet gateway is required only if you're planning to use a bastion server that needs accessibility from the internet. For instructions, see [To set up an internet gateway](#).

Create Internet Gateway [Help](#) [Cancel](#)

NAME

HubIGW

CREATE IN COMPARTMENT

TransitRouting

inforacleeli (root)/TransitRouting

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.

[Learn more about tagging](#)

| TAG NAMESPACE | TAG KEY | VALUE |
|-------------------------|---------|-------|
| None (add a free-fo...) | | |

+ Additional Tag

Create Internet Gateway Cancel

Figure 5: Create an Internet Gateway for the Hub VCN

8. Create a DRG, such as AzureDRG, for the hub VCN.

In a later step, you attach it to FastConnect to provide connectivity to Azure. For instructions, see [Creating a DRG](#).

Create Dynamic Routing Gateway [Help](#)

CREATE IN COMPARTMENT

TransitRouting

inforacleeli (root)/TransitRouting

NAME

AzureDRG

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.

[Learn more about tagging](#)

| TAG NAMESPACE | TAG KEY | VALUE |
|---------------|---------|-------|
| None (a...) | | |

+ Additional Tag

Create Dynamic Routing Gateway Cancel

Figure 6: Create a DRG for the Hub VCN

9. Attach the DRG to the hub VCN.

In general, a VCN can be attached to only one DRG at a time, and a DRG can be attached to only one VCN at a time. For instructions, see [Attaching a VCN to a DRG](#).

AzureDRG

Edit Add Tags Move Resource **Terminate**

Dynamic Routing Gateway Information Tags

OCID: ...252ozibnba [Show](#) [Copy](#) Oracle Redundancy Status: —

Created: Mon, Jun 15, 2020, 2:37:54 PM UTC

Virtual Cloud Networks

[Attach to Virtual Cloud Network](#)

| Name | Lifecycle State | CIDR Block | Attachment State | Attachment Route Table |
|-------------------------|-----------------|---------------|------------------|------------------------|
| Hub_VCN | Available | 10.115.1.0/24 | Attached | — |

Figure 7: Attach the DRG to the Hub VCN

10. Modify the default route table for the hub VCN by creating the following route rules. For instructions, see [To update rules in an existing route table](#).

- Create a route rule of the target type DRGs that points the DRG that you created, which allows the route to the Azure CIDR.
- Create a route rule of the target type local peering gateway (LPG) that points to the local peering gateway that you created, which allows the traffic route to the spoke VCN.
- If you created an internet gateway, create a route rule of the target type internet gateway that points to it, which allows traffic to the public internet.

Default Route Table for Hub_VCN

Move Resource Add Tags Terminate

Route Table Information Tags

OCID: ...tjm5ra [Show](#) [Copy](#) Compartment: TransitRouting

Created: Mon, Jun 15, 2020, 14:26:43 UTC

Route Rules

Add Route Rules Edit Remove

| <input type="checkbox"/> | Destination | Target Type | Target | Description |
|--------------------------|----------------|--------------------------|--------------------------|-------------|
| <input type="checkbox"/> | 0.0.0.0/0 | Internet Gateway | HubIGW | ⋮ |
| <input type="checkbox"/> | 172.169.0.0/16 | Dynamic Routing Gateways | AzureDRG | ⋮ |
| <input type="checkbox"/> | 192.168.1.0/24 | Local Peering Gateway | HubLPG | ⋮ |

0 Selected Showing 3 Items < Page 1 >

Figure 8: Create Route Rules for the Hub VCN's Default Route Table

11. Create a route table, such as HubDRG_RT, that you will attach to the DRG in a later step.

The specified route allows the flow of traffic from Azure to the spoke VCN through the LPG that you created. For instructions, see [To create a route table](#).

The screenshot displays the 'HubDRG_RT' route table configuration in the Oracle Cloud console. At the top, there are three buttons: 'Move Resource', 'Add Tags', and 'Terminate'. Below these are two tabs: 'Route Table Information' and 'Tags'. The 'Route Table Information' tab is active, showing the OCID as '...4swraa', a 'Show' link, a 'Copy' link, and the compartment as 'TransitRouting'. The creation time is listed as 'Mon, Jun 15, 2020, 14:42:18 UTC'. Below this is the 'Route Rules' section, which includes 'Add Route Rules', 'Edit', and 'Remove' buttons. A table lists one route rule with the following details:

| <input type="checkbox"/> | Destination | Target Type | Target | Description |
|--------------------------|----------------|-----------------------|------------------------|-------------|
| <input type="checkbox"/> | 192.168.1.0/24 | Local Peering Gateway | HubLPG | |

At the bottom of the table, it indicates '0 Selected' and 'Showing 1 Item < Page 1 >'.

Figure 9: Create a Route Table for the DRG

12. Create another route table, such as HubLPG_RT, that you will assign to the LPG in a later step.

This rule allows the flow of traffic from the local peering gateway to the Azure CIDR.

The screenshot displays the 'HubLPG_RT' route table configuration in the Oracle Cloud console. At the top, there are three buttons: 'Move Resource', 'Add Tags', and 'Terminate'. Below these are two tabs: 'Route Table Information' and 'Tags'. The 'Route Table Information' tab is active, showing the OCID as '...qigqaa', a 'Show' link, a 'Copy' link, and the compartment as 'TransitRouting'. The creation time is listed as 'Mon, Jun 15, 2020, 14:43:39 UTC'. Below this is the 'Route Rules' section, which includes 'Add Route Rules', 'Edit', and 'Remove' buttons. A table lists one route rule with the following details:

| <input type="checkbox"/> | Destination | Target Type | Target | Description |
|--------------------------|----------------|--------------------------|--------------------------|-------------|
| <input type="checkbox"/> | 172.169.0.0/16 | Dynamic Routing Gateways | AzureDRG | |

At the bottom of the table, it indicates '0 Selected' and 'Showing 1 Item < Page 1 >'.

Figure 10: Create a Route Table for the Local Peering Gateway

13. Modify the default security list for the hub VCN to allow traffic to flow between Azure and the spoke VCN through the hub VCN.

For this example, traffic is allowed on all protocols, but you can apply more restrictive port and traffic-specific rules. For instructions, see [To update rules in an existing security list](#).

Default Security List for Hub_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: ...zqjsva [Show](#) [Copy](#) Compartment: TransitRouting

Created: Mon, Jun 15, 2020, 14:26:43 UTC

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 | | TCP traffic for ports: 22 SSH Remote Login Protocol | ⋮ |
| <input type="checkbox"/> | No | 0.0.0.0/0 | ICMP | | | 3, 4 | ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set | ⋮ |
| <input type="checkbox"/> | No | 10.115.1.0/24 | ICMP | | | 3 | ICMP traffic for: 3 Destination Unreachable | ⋮ |
| <input type="checkbox"/> | No | 192.168.1.0/24 | All Protocols | | | | All traffic for all ports | ⋮ |
| <input type="checkbox"/> | No | 172.169.0.0/16 | All Protocols | | | | All traffic for all ports | ⋮ |

0 Selected Showing 5 Items < Page 1 >

Figure 11: Create Security List Ingress Rules for the Hub VCN

14. Associate the route table that you created in step 11 (in our example, HubDRG_RT) with the DRG. For instructions, see [To associate a VCN route table with an existing DRG](#).

Associate Route Table [Help](#) [Cancel](#)

DYNAMIC ROUTING GATEWAY *READ-ONLY*

AzureDRG

ROUTE TABLE COMPARTMENT

TransitRouting

intoracleeli (root)/TransitRouting

ROUTE TABLE

HubDRG_RT

Important: After you associate a route table, you can't remove it. But you can associate a different table or delete the rules.

[Associate](#) [Cancel](#)

Figure 12: Associate a Route Table with the DRG

15. On the details page for the hub VCN, verify that the route table is attached successfully to the DRG.

Hub_VCN

Move Resource Add Tags Terminate

VCN Information Tags

CIDR Block: 10.115.1.0/24
Compartment: TransitRouting
Created: Mon, Jun 15, 2020, 14:26:43 UTC

OCID: ...tkq5vq Show Copy
Default Route Table: [Default Route Table for Hub_VCN](#)
DNS Domain Name: hubvcn.oraclevcn.com

Dynamic Routing Gateways

Attach Dynamic Routing Gateway

| Name | State | Compartment | Route Table ⓘ | Created |
|--------------------------|------------|----------------|---------------------------|---------------------------------|
| AzureDRG | ● Attached | TransitRouting | HubDRG_RT | Mon, Jun 15, 2020, 14:38:50 UTC |

Showing 1 Item < Page 1 >

Figure 13: Verify the Route Table Attachment to the DRG

16. Associate the route table that you created in step 12 (in our example, HubLPG_RT) with the LPG of the hub VCN. For instructions, see [To associate a route table with an existing local peering gateway](#).

Associate Route Table [Help](#) [Cancel](#)

LOCAL PEERING GATEWAY READ-ONLY

HubLPG

ROUTE TABLE COMPARTMENT

TransitRouting

inforacleeli (root)/TransitRouting

ROUTE TABLE

HubLPG_RT

Important: After you associate a route table, you can't remove it. But you can associate a different table or delete the rules.

[Associate](#) [Cancel](#)

Figure 14: Associate a Route Table with the Local Peering Gateway

17. On the details page for the hub VCN, verify that the route table is attached successfully to the LPG.

Hub_VCN

Move Resource Add Tags Terminate

VCN Information Tags

CIDR Block: 10.115.1.0/24 **OCID:** ...tkq5vq [Show](#) [Copy](#)
Compartment: TransitRouting **Default Route Table:** [Default Route Table for Hub_VCN](#)
Created: Mon, Jun 15, 2020, 14:26:43 UTC **DNS Domain Name:** hubvcn.oraclevcn.com

Local Peering Gateways in TransitRouting Compartment

Create Local Peering Gateway

| Name | State | Peering Status | Route Table ⓘ | Peer Advertised CIDR | Cross-Tenancy | Created |
|--------|-----------|-------------------------------|---------------------------|----------------------|---------------|---------------------------------|
| HubLPG | Available | New - Not connected to a peer | HubLPG_RT | | | Mon, Jun 15, 2020, 14:28:45 UTC |

Figure 15: Verify the Route Table Attachment to the Local Peering Gateway

18. Create a subnet, such as HubPublic, for the hub VCN by using the default route table and default security list that you modified earlier. For instructions, see [To create a subnet](#).

Create Subnet

If the Route Table, DHCP Options, or Security Lists are in a different Compartment than the Subnet, enable Compartment selection for those resources: [Click here](#)

NAME
HubPublic

SUBNET TYPE
☒ REGIONAL (RECOMMENDED)
Instances in the subnet can be created in any availability domain in the region. Useful for high availability.
☐ AVAILABILITY DOMAIN-SPECIFIC
Instances in the subnet can only be created in one availability domain in the region.

CIDR BLOCK
10.115.1.0/25
Specified IP addresses: 10.115.1.0-10.115.1.127 (128 IP addresses)

ROUTE TABLE
Default Route Table for Hub_VCN

SUBNET ACCESS
☐ PRIVATE SUBNET
Prohibit public IP addresses for instances in this Subnet.
☒ PUBLIC SUBNET
Allow public IP addresses for instances in this Subnet.

DNS RESOLUTION
☒ USE DNS HOSTNAMES IN THIS SUBNET ⓘ
Allow assignment of DNS hostnames when launching an instance.

DNS LABEL
HubPublic
Only letters and numbers, starting with a letter. 15 characters max.

DNS DOMAIN NAME READ-ONLY
<dns-label>-hubvcn.oraclevcn.com

DHCP OPTIONS
Select DHCP Options

Security Lists
SECURITY LIST
Default Security List for Hub_VCN

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.
[Learn more about tagging](#)

TAG NAME/SPACE
None (add a free-form tag)

TAG KEY

Create Subnet Cancel

Figure 16: Create a Subnet in the Hub VCN

19. If required, create a bastion server in the hub VCN.

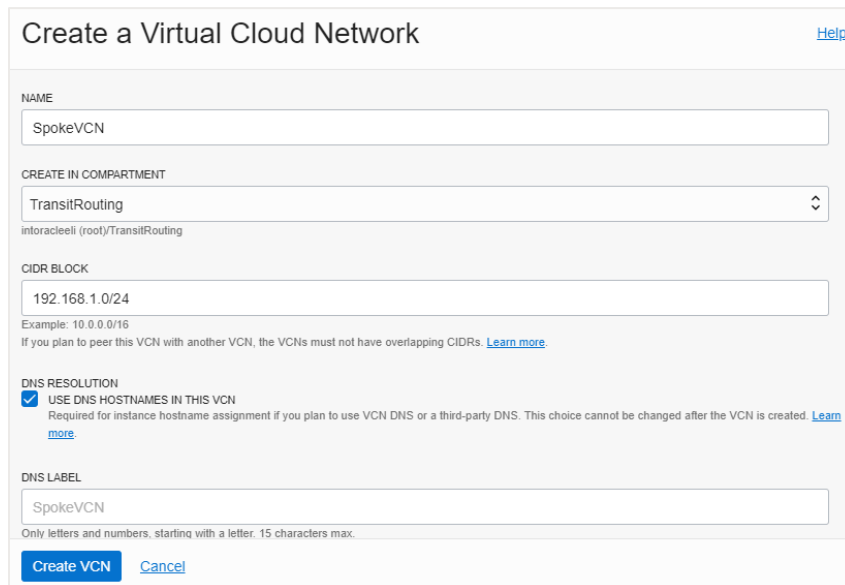
Step 2: Set Up the Spoke VCN

As a part of the spoke VCN setup, create distinct private and public subnets. The private subnet hosts a shared autonomous database that's accessed from Azure through the hub VCN in OCI over the private interconnect.

The public subnet can be used to host any other internet-facing applications. In this example, it's configured but not used.

For details about how to create the OCI resources, see the instructions referred in the steps in the preceding section.

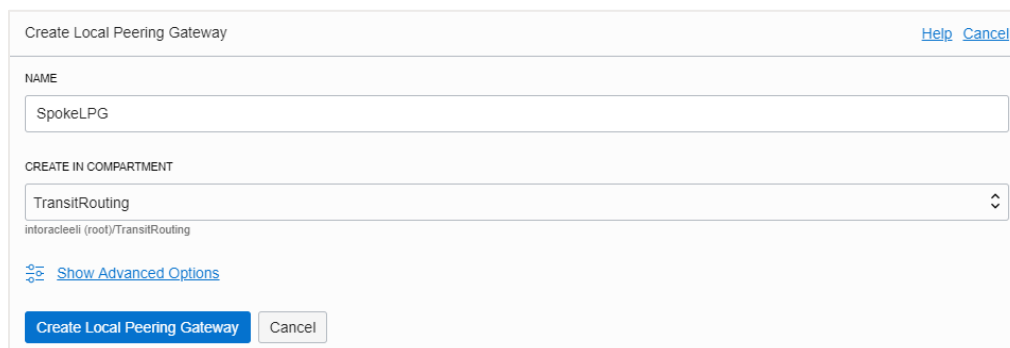
1. Create a spoke VCN, such as SpokeVCN. Specify a CIDR address block that doesn't overlap with the hub VCN or the Azure VNet.



The screenshot shows the 'Create a Virtual Cloud Network' form. The title is 'Create a Virtual Cloud Network' with a 'Help' link. The form fields are: NAME (SpokeVCN), CREATE IN COMPARTMENT (TransitRouting, with a dropdown arrow and the path 'inforacleeli (root)/TransitRouting' below it), CIDR BLOCK (192.168.1.0/24, with an example '10.0.0.0/16' and a note about overlapping CIDRs with a 'Learn more' link), DNS RESOLUTION (checked 'USE DNS HOSTNAMES IN THIS VCN' with a note about instance hostname assignment and a 'Learn more' link), and DNS LABEL (SpokeVCN, with a note 'Only letters and numbers, starting with a letter. 15 characters max.'). At the bottom are 'Create VCN' and 'Cancel' buttons.

Figure 17: Create the Spoke VCN

2. Create a local peering gateway, such as SpokeLPG, for the spoke VCN.



The screenshot shows the 'Create Local Peering Gateway' form. The title is 'Create Local Peering Gateway' with 'Help' and 'Cancel' links. The form fields are: NAME (SpokeLPG), CREATE IN COMPARTMENT (TransitRouting, with a dropdown arrow and the path 'inforacleeli (root)/TransitRouting' below it), and a 'Show Advanced Options' link. At the bottom are 'Create Local Peering Gateway' and 'Cancel' buttons.

Figure 18: Create the Local Peering Gateway for the Spoke VCN

3. Connect the hub VCN and the spoke VCN by using their local peering gateways. For instructions, see [Setting Up a Local Peering](#).

Establish Peering Connection [Help](#) [Cancel](#)

SPECIFY THE LOCAL PEERING GATEWAY

☒ BROWSE BELOW ☐ ENTER LOCAL PEERING GATEWAY OCID

VIRTUAL CLOUD NETWORK COMPARTMENT

TransitRouting
inforacleeli (root)/TransitRouting

VIRTUAL CLOUD NETWORK

Hub_VCN

LOCAL PEERING GATEWAY COMPARTMENT

TransitRouting
inforacleeli (root)/TransitRouting

UNPEERED PEER GATEWAY

HubLPG

[Establish Peering Connection](#) [Cancel](#)

Figure 19: Create Local Peering from the Spoke VCN to the Hub VCN

4. On the details page of the spoke VCN, verify that the peering is successful and connected.

SpokeVCN

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

VCN Information [Tags](#)

CIDR Block: 192.168.1.0/24
Compartment: TransitRouting
Created: Mon, Jun 15, 2020, 14:53:48 UTC

OCID: ...wqzpf [Show](#) [Copy](#)
Default Route Table: [Default Route Table for SpokeVCN](#)
DNS Domain Name: spokevcn.oraclevcn.com

Local Peering Gateways in TransitRouting Compartment

[Create Local Peering Gateway](#)

| Name | State | Peering Status | Route Table ⓘ | Peer Advertised CIDR | Cross-Tenancy | Created |
|----------|-----------|-------------------------------|---------------|----------------------|---------------|---------------------------------|
| SpokeLPG | Available | Peered - Connected to a peer. | | 0.0.0.0/0 | No | Mon, Jun 15, 2020, 14:57:32 UTC |

Showing 1 Item < Page 1 >

Figure 20: Verify Local Peering of Spoke and Hub VCNs

- Create a route table, such as PrivateRT, that will be attached to create a private subnet in the spoke VCN. Specify rules that allow the flow of traffic to the hub VCN and to Azure through an LPG.

PrivateRT

Move Resource
Add Tags
Terminate

Route Table Information

Tags

OCID: ...sqhhwa [Show](#) [Copy](#)
Compartment: TransitRouting

Created: Mon, Jun 15, 2020, 15:00:02 UTC

Route Rules

Add Route Rules
Edit
Remove

| <input type="checkbox"/> | Destination | Target Type | Target | Description |
|--------------------------|----------------|-----------------------|--------------------------|-------------|
| <input type="checkbox"/> | 10.115.1.0/24 | Local Peering Gateway | SpokeLPG | ⋮ |
| <input type="checkbox"/> | 172.169.0.0/16 | Local Peering Gateway | SpokeLPG | ⋮ |

0 Selected
Showing 2 Items
Page 1

Figure 21: Create Route Table for Private Subnet

- If resources in the public subnet also require access from Azure, the hub VCN, or both, add the same rules to the default route table for the spoke VCN.

Default Route Table for SpokeVCN

Move Resource
Add Tags
Terminate

Route Table Information

Tags

OCID: ...hofopq [Show](#) [Copy](#)
Compartment: TransitRouting

Created: Mon, Jun 15, 2020, 14:53:48 UTC

Route Rules

Add Route Rules
Edit
Remove

| <input type="checkbox"/> | Destination | Target Type | Target | Description |
|--------------------------|----------------|-----------------------|--------------------------|-------------|
| <input type="checkbox"/> | 10.115.1.0/24 | Local Peering Gateway | SpokeLPG | ⋮ |
| <input type="checkbox"/> | 172.169.0.0/16 | Local Peering Gateway | SpokeLPG | ⋮ |

0 Selected
Showing 2 Items
Page 1

Figure 22: Create Route Rules for the Spoke VCN Default Route Table

- Create a security list, such as Private SL, for the private subnet, and specify ingress traffic rules from Azure, the hub VCN, and the public subnet of the spoke VCN.

You can specify more restrictive rules to limit the flow of traffic on specified ports.

The rules specified in the security list of a private subnet don't govern the access to the autonomous database. A network security group that you create later controls and manages access to the database.

PrivateSL

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: ...ka567a
Show
Copy
Compartment: TransitRouting
Created: Mon, Jun 15, 2020, 15:02:47 UTC

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 | 172.169.0.0/16 | All Protocols | | | | All traffic for all ports | ⋮ |
| <input type="checkbox"/> | No | 10.115.1.0/24 | All Protocols | | | | All traffic for all ports | ⋮ |
| <input type="checkbox"/> | No | 192.168.1.0/25 | All Protocols | | | | All traffic for all ports | ⋮ |

0 Selected
Showing 3 Items
Page 1

Figure 23: Create Ingress Rules for the Security List

- Add an egress traffic rule to allow all the outgoing traffic.

By default, user-created, nondefault security lists don't have any default egress rules. You can also specify more restrictive rules.

PrivateSL

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: ...ka567a
Show
Copy
Compartment: TransitRouting
Created: Mon, Jun 15, 2020, 15:02:47 UTC

Egress Rules

Add Egress Rules
Edit
Remove

| <input type="checkbox"/> | Stateless ▾ | Destination | IP Protocol | Source Port Range | Destination Port Range | Type and Code | Allows | Description |
|--------------------------|-------------|-------------|---------------|-------------------|------------------------|---------------|---------------------------|-------------|
| <input type="checkbox"/> | No | 0.0.0.0/0 | All Protocols | | | | All traffic for all ports | ⋮ |

0 Selected
Showing 1 Item
Page 1

Figure 24: Create Egress Rules for the Security List

9. Modify the default security list for the spoke VCN and add ingress rules to allow the incoming traffic from the hub VCN and Azure VNet.

The default egress rules are sufficient for this example.

Default Security List for SpokeVCN

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: ...upkrtq [Show](#) [Copy](#)

Compartment: TransitRouting

Created: Mon, Jun 15, 2020, 14:53:48 UTC

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 | | TCP traffic for ports: 22 SSH Remote Login Protocol | ⋮ |
| <input type="checkbox"/> | No | 0.0.0.0/0 | ICMP | | | 3, 4 | ICMP traffic for: 3, 4 Destination Unreachable: Fragmentation Needed and Don't Fragment was Set | ⋮ |
| <input type="checkbox"/> | No | 192.168.1.0/24 | ICMP | | | 3 | ICMP traffic for: 3 Destination Unreachable | ⋮ |
| <input type="checkbox"/> | No | 10.115.1.0/24 | All Protocols | | | | All traffic for all ports | ⋮ |
| <input type="checkbox"/> | No | 172.169.0.0/16 | All Protocols | | | | All traffic for all ports | ⋮ |

0 Selected Showing 5 items Page 1

Figure 25: Create Ingress Rules for the Default Security List for the Spoke VCN

10. Create a public subnet, such as SpokePublic, that uses the default route table and the default security list.

Create Subnet

If the Route Table, DHCP Options, or Security Lists are in a different Compartment than the Subnet, enable Compartment selection for those resources: [Click here](#)

NAME

SpokePublic

SUBNET TYPE

☒ REGIONAL (RECOMMENDED)
Instances in the subnet can be created in any availability domain in the region. Useful for high availability.

☐ AVAILABILITY DOMAIN-SPECIFIC
Instances in the subnet can only be created in one availability domain in the region.

CIDR BLOCK

192.168.1.0/25

Specified IP addresses: 192.168.1.0-192.168.1.127 (128 IP addresses)

ROUTE TABLE

Default Route Table for SpokeVCN

SUBNET ACCESS

☐ PRIVATE SUBNET
Prohibit public IP addresses for instances in this Subnet.

☒ PUBLIC SUBNET
Allow public IP addresses for instances in this Subnet.

DNS RESOLUTION

☒ USE DNS HOSTNAMES IN THIS SUBNET ⓘ
Allows assignment of DNS hostname when launching an instance.

DNS LABEL

SpokePublic

Only letters and numbers, starting with a letter. 15 characters max.

DNS DOMAIN NAME READ-ONLY

<dns-label>.spokevcn.oraclevcn.com

DHCP OPTIONS

Select DHCP Options

Security Lists

SECURITY LIST

Default Security List for SpokeVCN

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.
[Learn more about tagging](#)

TAG NAMESPACE TAG KEY

None (add a free-form tag)

Create Subnet Cancel

Figure 26: Create a Public Subnet for the Spoke VCN

11. Create a private subnet, such as Spoke Private, using the route table and security list that you created (in this example, PrivateRT and PrivateSL).

Create Subnet

If the Route Table, DHCP Options, or Security Lists are in a different Compartment than the Subnet, enable Compartment selection for those resources. [Click here](#)

NAME
SpokePrivate

SUBNET TYPE
☒ REGIONAL (RECOMMENDED)
Instances in the subnet can be created in any availability domain in the region. Useful for high availability.
☐ AVAILABILITY DOMAIN-SPECIFIC
Instances in the subnet can only be created in one availability domain in the region.

CIDR BLOCK
192.168.1.128/25
Specified IP addresses: 192.168.1.128-192.168.1.255 (128 IP addresses)

ROUTE TABLE
PrivateRT

SUBNET ACCESS
☒ PRIVATE SUBNET
Prohibit public IP addresses for instances in this Subnet.
☐ PUBLIC SUBNET
Allow public IP addresses for instances in this Subnet.

DNS RESOLUTION
☒ USE DNS HOSTNAMES IN THIS SUBNET ⓘ
Allows assignment of DNS hostnames when launching an instance.

DNS LABEL
SpokePrivate
Only letters and numbers, starting with a letter. 15 characters max.

DNS DOMAIN NAME: `spoke-private-vcn`
<dns-label>-spoke-private-vcn.oraclecloud.com

DHCP OPTIONS
Select DHCP Options

Security Lists

SECURITY LIST
PrivateSL

Tagging is a metadata system that allows you to organize and track resources within your tenancy. Tags are composed of keys and values that can be attached to resources.
[Learn more about tagging](#)

TAG NAMESPACE
None (add a free-form tag)

TAG KEY

Create Subnet Cancel

Figure 27: Create a Private Subnet for the Spoke VCN

12. Create a network security group (NSG) (in our example, ADB_NSG) used by the autonomous database that controls access to the database on the specified port. For instructions, see [To create an NSG](#).

The screenshot shows the 'ADB_NSG' Network Security Group details page. At the top, there are buttons for 'Edit', 'Move Resource', 'Add Tags', and 'Terminate'. Below this is the 'Network Security Group Information' tab, which displays the OCID as '4ns6ya', the compartment as 'TransitRouting', and the creation time as 'Mon, Jun 15, 2020, 15:10:30 UTC'. The 'Security Rules' tab is active, showing a table of three ingress rules. Each rule is for TCP traffic on destination port 1522, with different source CIDR ranges. The table has columns for checkboxes, direction, stateless status, source/destination, protocol, details, and description. At the bottom, it indicates '0 Selected'.

| <input type="checkbox"/> | Direction | Stateless | Source or Destination | Protocol | Details | Description |
|--------------------------|--------------------|---------------|---|----------|--|-------------|
| <input type="checkbox"/> | Direction: Ingress | Stateless: No | Source Type: CIDR Source: 10.115.1.0/24 | TCP | Source Port Range: All Destination Port Range: 1522 Allow: TCP tra... Show | |
| <input type="checkbox"/> | Direction: Ingress | Stateless: No | Source Type: CIDR Source: 192.168.1.0/25 | TCP | Source Port Range: All Destination Port Range: 1522 Allow: TCP tra... Show | |
| <input type="checkbox"/> | Direction: Ingress | Stateless: No | Source Type: CIDR Source: 172.169.0.0/16 | TCP | Source Port Range: All Destination Port Range: 1522 Allow: TCP tra... Show | |

Figure 28: Create a Network Security Group for the Spoke VCN

13. Create a shared autonomous database in the private subnet and use the NSG that you created to control the access to the database. For instructions, see [Creating an Autonomous Database on Shared Exadata Infrastructure](#).

Remember the private endpoint IP address, which is used to access the database from Azure.

The screenshot shows the 'ATP' details page. On the left is a green 'ATP' logo with the status 'AVAILABLE'. The main content area has tabs for 'DB Connection', 'Performance Hub', 'Service Console', 'Scale Up/Down', and 'More Actions'. The 'Autonomous Database Information' tab is active, showing 'General Information' and 'Infrastructure'. The 'General Information' section includes details like Database Name (ATP), Workload Type (Transaction Processing), Compartment (Intoracleeeli (root)/TransitRouting), OCID, creation time, OCPU Count (1), Storage (1 TB), License Type (Bring Your Own License (BYOL)), Database Version (19c), Auto Scaling (Disabled), Lifecycle State (Available), and Instance Type (Paid). The 'Infrastructure' section shows 'Dedicated Infrastructure: No', 'Backup' status, 'Network' details (Access Type: Virtual Cloud Network, Virtual Cloud Network: SpokeVCN, Subnet: SpokePrivate, Private Endpoint IP: 192.168.1.130, Private Endpoint URL: s55nhbo.adb.ap-tokyo-1.oraclecloud.com, Network Security Groups: ADB_NSG), and 'Maintenance' information (Next Maintenance: Sat, Jun 20, 2020, 05:00:00 UTC - 09:00:00 UTC).

Figure 29: Details Page for an Autonomous Database

Step 3: Set Up the Azure Environment

1. In the Azure Portal, create a virtual network (VNet).
 - A. On the **Basics** page, enter a name for the VNet, such as AzureVNET, and select the same region that you're using for OCI (in our example, Japan East).

Create virtual network

Basics IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

Project details

Subscription * ⓘ Microsoft Azure Sponsorship

Resource group * ⓘ Azure_Tokyo
[Create new](#)

Instance details

Name * AzureVNET ✓

Region * (Asia Pacific) Japan East

[Review + create](#) < Previous Next : IP Addresses > [Download a template for automation](#)

Figure 30: Create VNet in Azure, Specify Region

- B. On the **IP Addresses** page, specify the CIDR address block and create subnets, as required. In this example, the subnet is called PublicSubnet.

Create virtual network

Basics IP Addresses Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space

172.169.0.0/16 ✓

☐ Add IPv6 address space ⓘ

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

+ Add subnet - Remove subnet

| Subnet name | Subnet address range |
|--------------|----------------------|
| PublicSubnet | 172.169.1.0/24 |

[Review + create](#) < Previous Next : Security > [Download a template for automation](#)

Figure 31: Create VNet in Azure, Specify CIDR Address Block

- C. On the **Security** page, modify the security settings for the VNet, as required.

Create virtual network

Basics IP Addresses **Security** Tags Review + create

DDoS protection ⓘ **Basic** Standard

Firewall ⓘ **Disabled** Enabled

Review + create < Previous Next : Tags >

Figure 32: Create VNet in Azure, Specify Security Settings

- D. On the **Tags** page, specify any optional tags, as required.

Create virtual network

Basics IP Addresses Security **Tags** Review + create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. [Learn more about tags](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

| Name ⓘ | Value ⓘ |
|----------------------|------------------------|
| Location | : Japan East |
| <input type="text"/> | : <input type="text"/> |

Review + create < Previous Next : Review + create > [Download a template for automation](#)

Figure 33: Create VNet in Azure, Specify Tags

- E. On the **Review + create** page, review the information, and then click **Create**.

Create virtual network

✓ Validation passed

Basics IP Addresses Security Tags **Review + create**

Basics

| | |
|----------------|-----------------------------|
| Subscription | Microsoft Azure Sponsorship |
| Resource group | Azure_Tokyo |
| Name | AzureVNET |
| Region | Japan East |

IP addresses

| | |
|---------------|-------------------------------|
| Address space | 172.169.0.0/16 |
| Subnet | PublicSubnet (172.169.1.0/24) |

Create < Previous Next > [Download a template for automation](#)

Figure 34: Create VNet in Azure, Review Information

2. Create a VNet gateway, such as AzureVNET_Gateway, and select ExpressRoute as the gateway type.

The gateway requires a dedicated subnet (the minimum subnet requirement is /28) and a public IP address.

Create virtual network gateway

Basics Tags Review + create

Azure has provided a planning and design guide to help you configure the various VPN gateway options. [Learn more.](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Microsoft Azure Sponsorship

Resource group Azure_Tokyo (derived from virtual network's resource group)

Instance details

Name * AzureVNET_Gateway

Region * (Asia Pacific) Japan East

Gateway type * VPN ExpressRoute

SKU * Standard

Virtual network * AzureVNET

Only virtual networks in the currently selected subscription and region are listed.

Gateway subnet address range * 172.169.0.0/24

172.169.0.0 - 172.169.0.255 (256 addresses)

Public IP address

Public IP address * Create new Use existing

Public IP address name * AzureVNET_Gateway_IP

Public IP address SKU Basic

Review + create < Previous Next : Tags > Download a template for automation

Figure 35: Create VNet Gateway

3. Create a route table, such as Public_Tokyo_RT, for the public subnet that you created earlier.

You add specific routes in a later step.

Create route table

You can add routes to this table after it's created.

Name * Public_Tokyo_RT

Subscription * Microsoft Azure Sponsorship

Resource group * Azure_Tokyo

Location * (Asia Pacific) Japan East

Virtual network gateway route propagation Disabled Enabled

Create Automation options

Figure 36: Create Route Table

4. Create an NSG, such as Public_Tokyo_NSG, to control the incoming and outgoing traffic.

You add individual rules in a later step.

Create network security group

Basics Tags Review + create

Project details

Subscription * Microsoft Azure Sponsorship

Resource group * Azure_Tokyo

[Create new](#)

Instance details

Name * Public_Tokyo_NSG

Region * (Asia Pacific) Japan East

[Review + create](#) [< Previous](#) [Next : Tags >](#) [Download a template for automation](#)

Figure 37: Create Network Security Group

5. Associate the route table and the network security group that you created with the subnet, which accesses the autonomous database that you created in OCI in the preceding section.

PublicSubnet
AzureVNET

[Save](#) [Discard](#) [Delete](#) [Refresh](#)

Address range (CIDR block) * 172.169.1.0/24
172.169.1.0 - 172.169.1.255 (256 addresses)

Available addresses 251

NAT gateway None

☐ Add IPv6 address space

Network security group Public_Tokyo_NSG

Route table Public_Tokyo_RT

Users Manage users

Service endpoints Services 0 selected

Subnet delegation Delegate subnet to a service None

Figure 38: Associate Route Table and Network Security Group with Subnet

6. Modify the route table and specify the CIDR address block of the hub VCN in OCI.

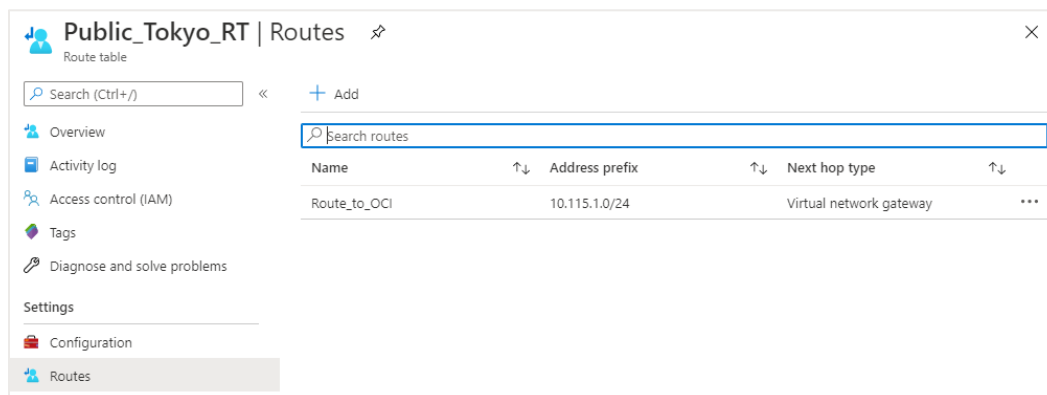


Figure 39: Modify Route Table with CIDR Address Block of the Hub VCN

7. Modify the network security group and specify the incoming and outgoing traffic rules to the hub VCN in OCI.

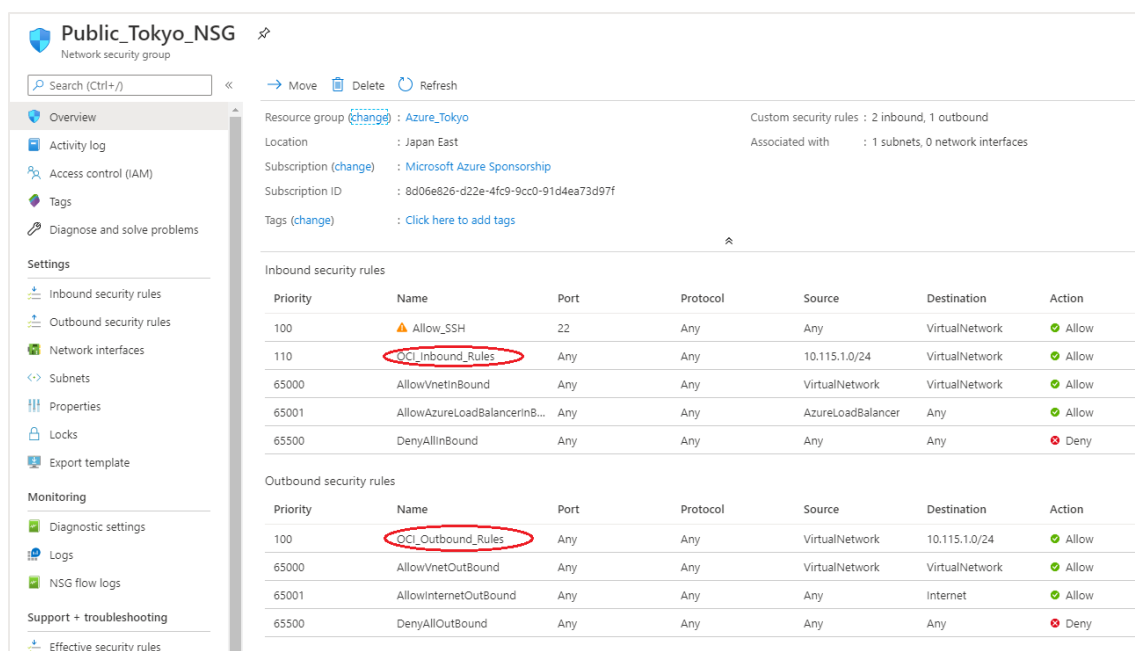


Figure 40: Modify Network Security Group with Traffic Rules to the Hub VCN

Step 4: Connect Oracle Cloud Infrastructure and Azure Using the Private Interconnect

1. In Azure Portal, create an ExpressRoute connection.
 - A. On the **Basics** page, enter a name, such as Azure_OCI_Circuit, and select the appropriate region and resource group that you specified in the preceding section.

Create ExpressRoute

Basics Configuration Tags Review + create

Use Azure ExpressRoute to create private connections between Azure datacenters and infrastructure on your premises or in a colocation environment. Establish connections to Azure at an ExpressRoute location, such as an Exchange provider facility, or directly connect to Azure from your existing WAN network, such as a multiprotocol label switching (MPLS) VPN, provided by a network service provider.
[Learn more about Express Route circuits](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Microsoft Azure Sponsorship

Resource group * ⓘ Azure_Tokyo
[Create new](#)

Instance details

Region * ⓘ Japan East

Name * ⓘ Azure_OCI_Circuit

[Review + create](#) < Previous Next: Configuration >

Figure 41: Create an ExpressRoute Connection in Azure

- B. On the **Configuration** page, select Oracle Cloud FastConnect as the provider.

The peering location shows only the regions that are currently supported for the private interconnect with OCI. Select the SKU option carefully; the price can vary drastically among the options. The Local SKU option has no separate ingress or egress charges.

Create ExpressRoute

Basics Configuration Tags Review + create

ExpressRoute circuits can connect to Azure through a service provider or directly to Azure at a global peering location.
[Learn more about circuit types](#)

Port type * Provider Direct

Create new or import from classic * ⓘ Create new Import

Provider * ⓘ Oracle Cloud FastConnect

Peering location * ⓘ Tokyo

Bandwidth * ⓘ 2Gbps

SKU * ⓘ Local Standard Premium

Billing model * ⓘ Unlimited

Allow classic operations ⓘ Yes No

[Review + create](#) < Previous Next: Tags >

Figure 42: Configure the ExpressRoute Circuit

- C. Verify the options that you selected and click **Create**.

Create ExpressRoute

✓ Validation Passed

Basics

| | |
|----------------|-----------------------------|
| Subscription | Microsoft Azure Sponsorship |
| Resource group | Azure_Tokyo |
| Region | Japan East |
| Name | Azure_OCI_Circuit |

Configuration

| | |
|-----------------------------------|--------------------------|
| Port type | Provider |
| Create new or import from classic | Create new |
| Provider | Oracle Cloud FastConnect |
| Peering location | Tokyo |
| Bandwidth | 2Gbps |
| SKU | Local |
| Billing model | Unlimited |
| Allow classic operations | No |

[Create](#) [< Previous](#) [Next](#) [Download a template for automation](#)

Figure 43: Verify the ExpressRoute Connection and Create

2. After the ExpressRoute circuit is created, copy the service key shown on the details page.

You use this key in the next step to provision a FastConnect connection and establish dynamic routing between Azure and OCI.

Azure_OCI_Circuit

ExpressRoute circuit

Search (Ctrl+/) Delete Refresh

Initiate the provisioning process with your service provider.

| | | | |
|-------------------------|--|------------------|--|
| Resource group (change) | : Azure_Tokyo | Provider | : Oracle Cloud FastConnect |
| Circuit status | : Enabled | Provider status | : Not provisioned |
| Location | : Japan East | Peering location | : Tokyo |
| Subscription (change) | : Microsoft Azure Sponsorship | Bandwidth | : 2000 Mbps |
| Subscription ID | : 8d05e826-d22e-4fc9-9cc0-91d4ea73d97f | Service key | : f4e67eb3-c58c-4fbc-8710-f09f8448d209 |
| Tags (change) | : Click here to add tags | | |

Peerings

| Type | ↑↓ Status | ↑↓ Primary subnet | ↑↓ Secondary subnet | ↑↓ Last modified by |
|---------------|-----------------|-------------------|---------------------|---------------------|
| Azure private | Not provisioned | - | - | - |
| Azure public | Not provisioned | - | - | - |
| Microsoft | Not provisioned | - | - | - |

Figure 44: ExpressRoute Details Page Showing the Service Key

3. In the Oracle Cloud Console, create a FastConnect connection.
 - A. On the **Connection Type** page, select **Microsoft Azure: ExpressRoute** from the list of FastConnect partners. For instructions, see [FastConnect: With an Oracle Partner](#).

Create Connection

1 Connection Type
2 Configuration

Connection Type

FastConnect lets you access your existing network from your Virtual Cloud Network (VCN) without traversing the internet. Choose an option:

CONNECTION TYPE

FastConnect Partner

Use this option if you have a relationship with a FastConnect partner. Here you set up the Oracle side of a virtual circuit that runs on the partner's connection. See the topics to the right.

✓

FastConnect Direct

Use this option if you want a dedicated connection by the way of a third-party network partner or by colocating in a FastConnect POP. Here you request a cross-connect and receive the Letter of Authorization (LOA). After cabling is complete at the POP, you return here to activate the cross-connect and set up at least one virtual circuit. See the topics to the right.

PARTNER

Microsoft Azure: ExpressRoute

Next [Cancel](#)

Figure 45: Create FastConnect Connection to Azure, Specify Connection Type

- B. On the **Configuration** page, specify the DRG that you created when you set up the hub VCN. Also, specify the service key that was provided when you set up the ExpressRoute circuit in Azure.

Provide the nonoverlapping BGP IP addresses to use for the two redundant BGP sessions between Oracle and Azure. For each pair, you must provide a separate /30 block of addresses (each /30 has four IP addresses).

Create Connection [Help](#)

1 Connection Type
2 Configuration

NAME OPTIONAL
Azure_OCI_Interconnect

COMPARTMENT
TransitRouting
inforaclell (root)/TransitRouting

VIRTUAL CIRCUIT TYPE

Private Virtual Circuit

Private IP addresses are advertised (typically RFC 1918). The connection uses a dynamic routing gateway that you attach to our VCN. ✓

Public Virtual Circuit

Oracle Cloud Infrastructure public IP addresses are advertised (for example, for Object Storage). You also provide the public IP prefixes that you want to advertise.

DYNAMIC ROUTING GATEWAY IN TRANSITROUTING [\[CHANGE COMPARTMENT\]](#)
AzureDRG

PROVISIONED BANDWIDTH
2 Gbps

PARTNER SERVICE KEY ⓘ
f4e67eb3-c58c-4fbc-8710-f09f8448d209

CUSTOMER PRIMARY BGP IP ADDRESS
10.10.0.2/30

ORACLE PRIMARY BGP IP ADDRESS OPTIONAL
10.10.0.1/30

CUSTOMER SECONDARY BGP IP ADDRESS
10.10.0.6/30

ORACLE SECONDARY BGP IP ADDRESS OPTIONAL
10.10.0.5/30

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

Figure 46: Create FastConnect Connection to Azure, Specify DRG and Service Key

C. Click **Create**.

After a few minutes, the FastConnect connection is provisioned, and dynamic routing is established between Azure and OCI.

[Create FastConnect](#)

| Name ▲ | Lifecycle State ⓘ | BGP State ⓘ | Connection Type ⓘ | Created |
|--|-------------------|-------------|-------------------|---------------------------------|
| Azure_OCI_Interconnect | ● Provisioned | ✓ Up | Partner | Fri, Jun 26, 2020, 15:10:26 UTC |

Showing 1 Item < Page 1 >

Figure 47: FastConnect Connection Provisioned and Up

4. In the Azure Portal, verify that the provider status of the ExpressRoute circuit has changed to **Provisioned**.

The screenshot shows the 'Azure_OCI_Circuit' page in the Azure Portal. The left-hand navigation pane has a 'Connections' tab highlighted with a red circle. The main content area displays the circuit's details:

- Resource group (change): [Azure_Tokyo](#)
- Circuit status: Enabled
- Location: Japan East
- Subscription (change): [Microsoft Azure Sponsorship](#)
- Subscription ID: 8d06e826-d22e-4fc9-9cc0-91d4ea73d97f
- Tags (change): [Click here to add tags](#)
- Provider: Oracle Cloud FastConnect
- Provider status: **Provisioned** (highlighted with a yellow circle)
- Peering location: Tokyo
- Bandwidth: 2000 Mbps
- Service key: f4e67eb3-c58c-4fbc-8710-f09f8448d209

Below the details is a 'Peering' table:

| Type | ↑↓ | Status | ↑↓ | Primary subnet | ↑↓ | Secondary subnet | ↑↓ | Last modified by |
|---------------|----|-----------------|----|----------------|----|------------------|----|------------------|
| Azure private | | Provisioned | | 10.10.0.0/30 | | 10.10.0.4/30 | | Customer |
| Azure public | | Not provisioned | | - | | - | | - |
| Microsoft | | Not provisioned | | - | | - | | - |

Figure 48: Verify Status of Provider on ExpressRoute Details Page

5. In the navigation pane, under **Settings**, click **Connections**.
6. Create a connection to attach the ExpressRoute circuit with the VNet gateway.
 - A. On the **Basics** page, enter a name for the connection, such as `Azure_OCI_Connection`.

The screenshot shows the 'Create connection' form in the Azure Portal. The 'Basics' tab is selected. The form contains the following fields:

- Project details:**
 - Subscription: Microsoft Azure Sponsorship
 - Resource group: Azure_Tokyo
- Instance details:**
 - Connection type: ExpressRoute
 - Name: Azure_OCI_Connection (with a green checkmark)
 - Region: (Asia Pacific) Japan East

At the bottom, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next : Settings >'. A link 'Download a template for automation' is also present.

Figure 49: Create Connection to Attach ExpressRoute to VNet Gateway

- B. On the **Settings** page, select the virtual network gateway from the list (in our example, AzureVNET_Gateway).

Create connection

Basics Settings Tags Review + create

Virtual network gateway

To use a virtual network with a connection, it must be associated to a virtual network gateway.

Virtual network gateway * ⓘ AzureVNET_Gateway ▼

ExpressRoute circuit ⓘ Azure_OCI_Circuit ▼

Redeem authorization ⓘ ☐

Routing weight * 0

[Review + create](#) [< Previous](#) [Next : Tags >](#) [Download a template for automation](#)

Figure 50: Create Connection to Attach ExpressRoute to VNet Gateway, Select VNet Gateway

- C. On the **Review + create** page, confirm the options, and click **Create**.

Create connection

✓ Validation passed

Basics Settings Tags Review + create

Basics

| | |
|-----------------|-----------------------------|
| Subscription | Microsoft Azure Sponsorship |
| Resource group | Azure_Tokyo |
| Region | Japan East |
| Connection type | ExpressRoute |
| Connection name | Azure_OCI_Connection |

Settings

| | |
|-----------------------------------|-------------------|
| Virtual network gateway | AzureVNET_Gateway |
| Redeem ExpressRoute authorization | No |
| ExpressRoute circuit | Azure_OCI_Circuit |
| Routing weight | 0 |

[Create](#) [< Previous](#) [Next >](#) [Download a template for automation](#)

Figure 51: Create Connection to Attach ExpressRoute to VNet Gateway, Verify Options

7. Verify that the connection is successfully added to the ExpressRoute circuit.

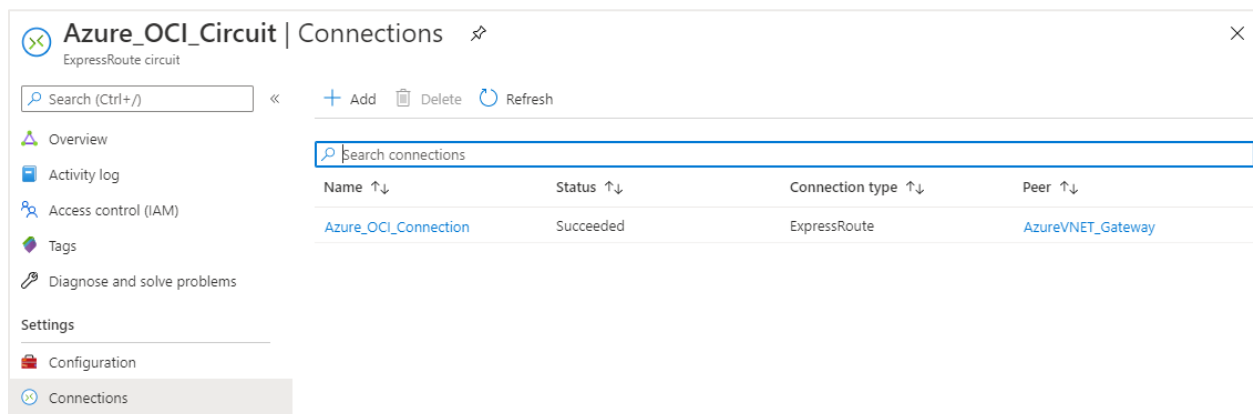


Figure 52: Verify Connection Is Added

The effective routes are now visible in the route table associated with the subnet (for example, Public_Tokyo_RT). If successful, the routes to the hub and spoke VCNs in OCI are visible through the virtual network gateway.

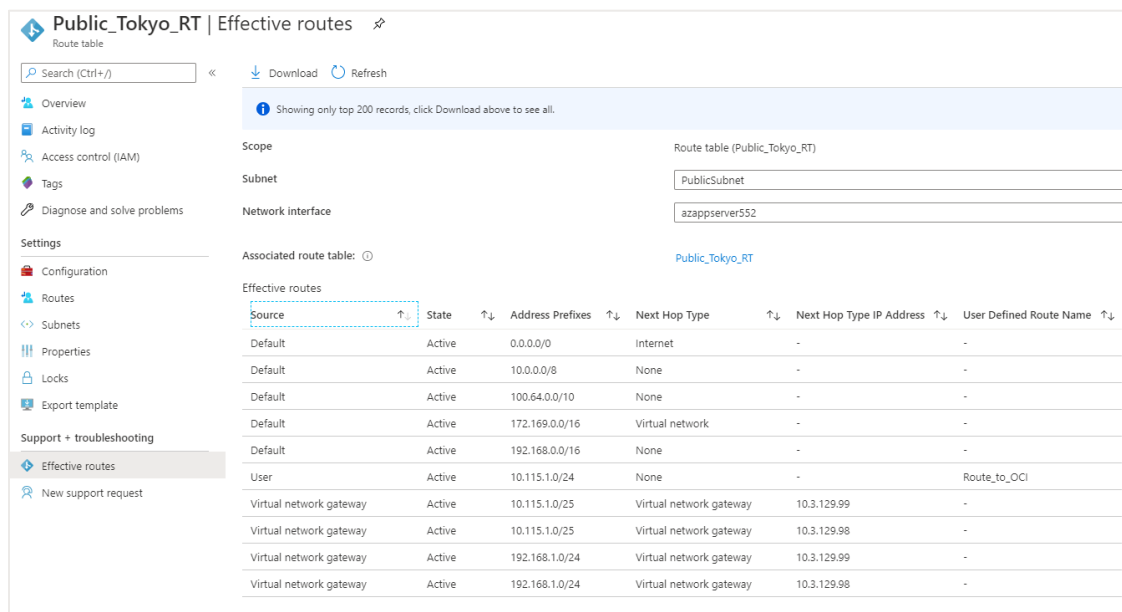


Figure 53: Effective Routes Between Azure and Oracle Cloud

Step 5: Access Autonomous Database from Azure Through the Private Interconnect

1. Create a virtual machine in the Azure public subnet and install Oracle Client to be able to access an Oracle database in OCI.

```
[oracle@AzAppServer ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.169.1.4 netmask 255.255.255.0 broadcast 172.169.1.255
    inet6 fe80::20d:3aff:fedc:4c92 prefixlen 64 scopeid 0x20<link>
    ether 00:0d:3a:cd:4c:92 txqueuelen 1000 (Ethernet)
    RX packets 873387 bytes 1278932209 (1.1 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 353034 bytes 34754225 (33.1 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

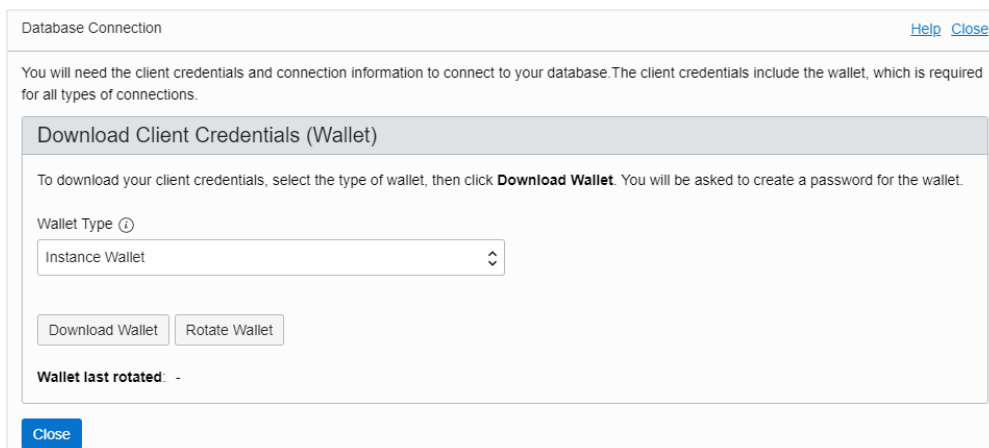
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 10437 bytes 13305824 (12.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10437 bytes 13305824 (12.6 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[oracle@AzAppServer ~]$
```

Figure 54: IP Address of an Azure VM

2. In the Oracle Cloud Console, go to the details page of the autonomous database, click **DB Connection**, and download the instance wallet.

The wallet contains the connection string and the wallet files for database traffic encryption between the database and the clients.



Database Connection [Help](#) [Close](#)

You will need the client credentials and connection information to connect to your database. The client credentials include the wallet, which is required for all types of connections.

Download Client Credentials (Wallet)

To download your client credentials, select the type of wallet, then click **Download Wallet**. You will be asked to create a password for the wallet.

Wallet Type ⓘ

Instance Wallet

Download Wallet Rotate Wallet

Wallet last rotated: -

Close

Figure 55: Select the Oracle Wallet Type for the Autonomous Database

3. Specify the password for the wallet and download it to your machine.

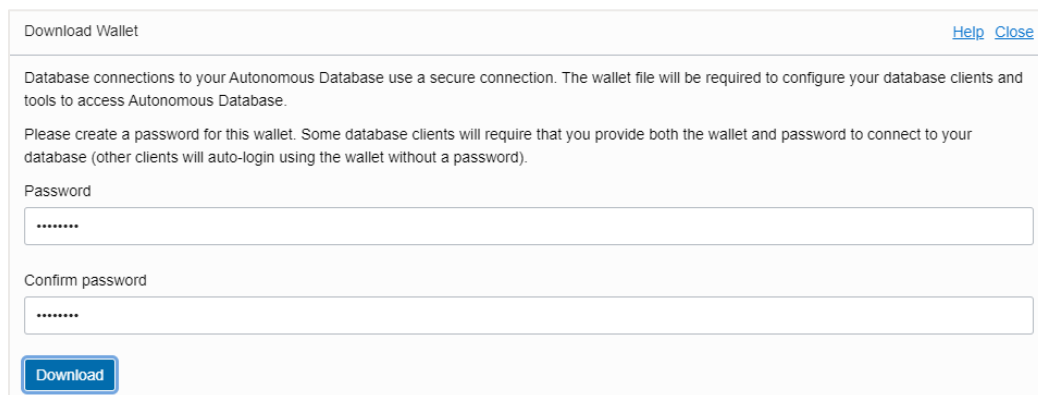


Figure 56: Download Oracle Wallet for the Autonomous Database

4. Transfer the wallet file to the Azure client server and unzip the contents in the `$ORACLE_HOME/network/admin` folder, overwriting any existing content that might exist.
5. On the database client machine, locate the `tnsnames.ora` file in `$ORACLE_HOME/network/admin` and replace the hostname with the private endpoint IP address of the autonomous database.
6. Perform a `tnsping` test to ensure that the listener of the autonomous database is reachable.

```
[oracle@AzAppServer ~]$ tnsping atp_high
TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 15-JUN-2020 16:56:46
Copyright (c) 1997, 2019, Oracle. All rights reserved.

Used parameter files:
/u01/app/oracle/product/19.0.0/client_1/network/admin/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (description= (retry_count=20) (retry_delay=3) (address=(protocol=tcp)(port=1522)(host=192.168.1.130))(connect_data=(servi
h.atp.oraclecloud.com)) (security=(ssl_server_cert_dn=CN=adb.ap-tokyo-1.oraclecloud.com, OU=Oracle ADB TOKYO, O=Oracle Corporation, L=Redwood City,
OK (70 msec)
[oracle@AzAppServer ~]$
```

Figure 57: Result of a Ping Test

The connection to the database is now successfully established.

```
[oracle@AzAppServer ~]$ sqlplus admin@atp_high

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Jun 15 17:03:12 2020
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter password:
Last Successful login time: Mon Jun 15 2020 16:57:34 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.5.0.0.0

SQL> Select open_mode from v$database;

OPEN_MODE
-----
READ WRITE

SQL>
```

Figure 58: Successful Database Connection

Conclusion

The implementation of hub-and-spoke configured transit routing offered in OCI can extend to applications that span between Microsoft Azure and OCI. A Database service deployed in a spoke VCN in OCI can be securely accessed over the private interconnect with Azure without the use of service gateway. Traffic originating from Azure can be centrally controlled and directed from a hub VCN in Oracle Cloud Infrastructure to the appropriate spoke VCN.

References

- [Transit Routing: Access to Multiple VCNs in the Same Region](#)
- [Set up a hub-and-spoke network topology](#)
- [Set up a direct interconnection between Azure and Oracle Cloud Infrastructure](#)
- [Create and modify an ExpressRoute circuit using PowerShell](#)
- [Access to Microsoft Azure](#)

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 © 2022, 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120