ORACLE

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

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

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.

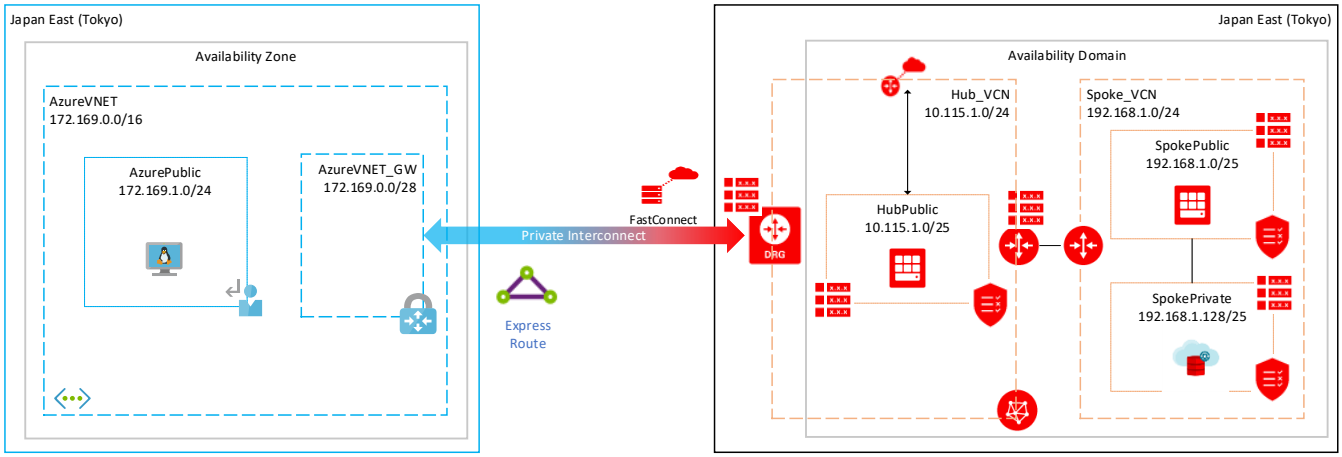

Figure 1: Example Hub-and-Spoke Topology Between Oracle Cloud Infrastructure and Microsoft Azure

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.

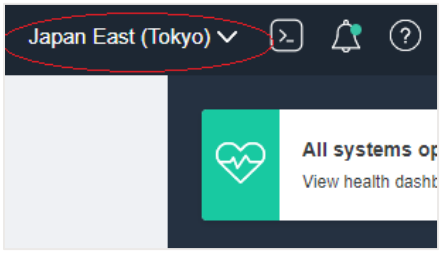

Figure 2: Select a Region for the Interconnect

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.

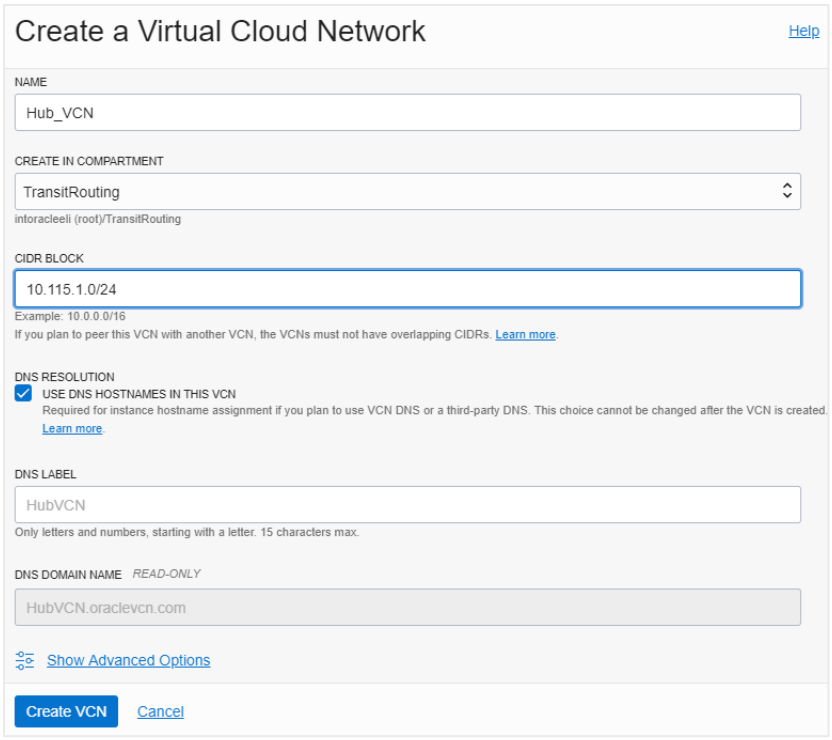

Figure 3: Create the Hub VCN

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.

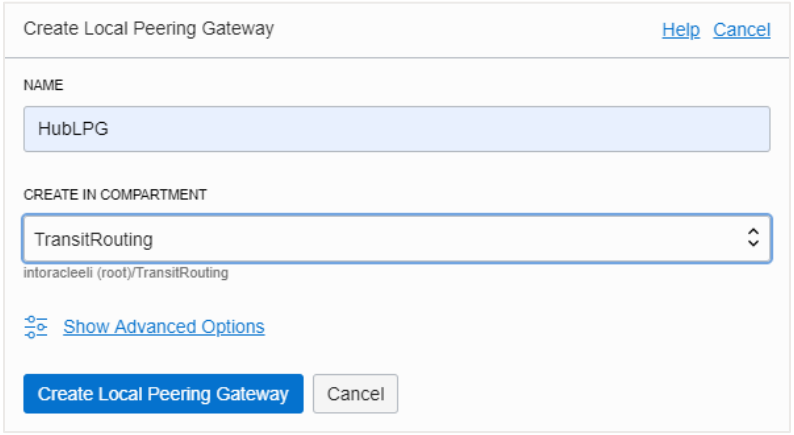

Figure 4: Create the Local Peering Gateway for the Hub VCN

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.

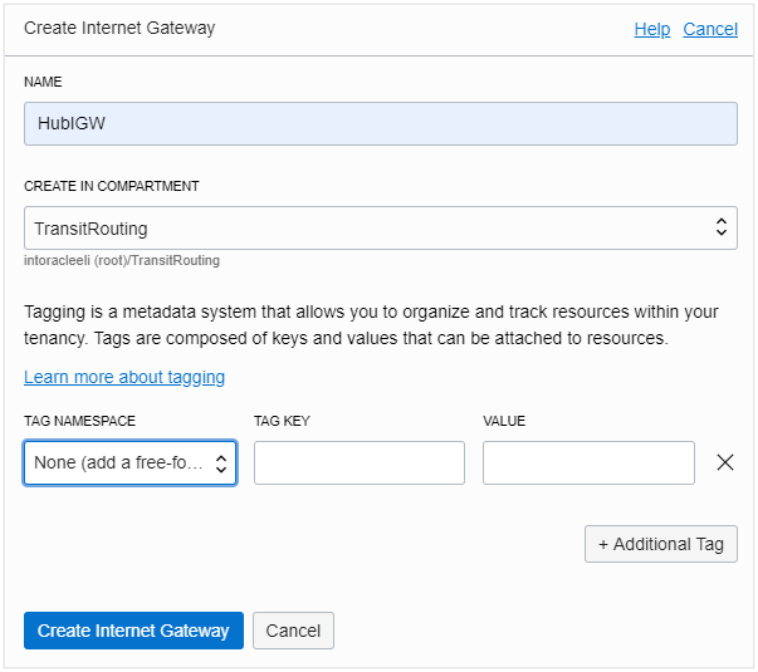

Figure 5: Create an Internet Gateway for the Hub VCN

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.

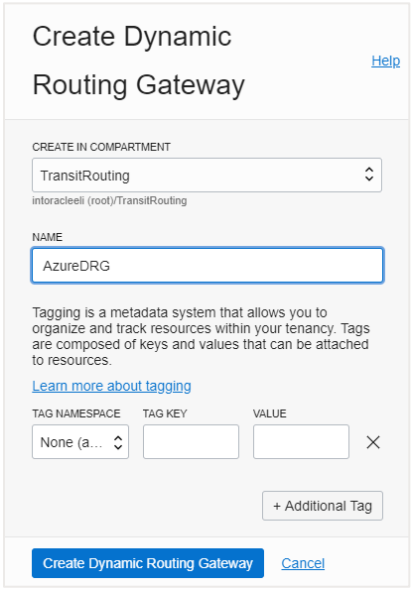

Figure 6: Create a DRG for the Hub VCN

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.

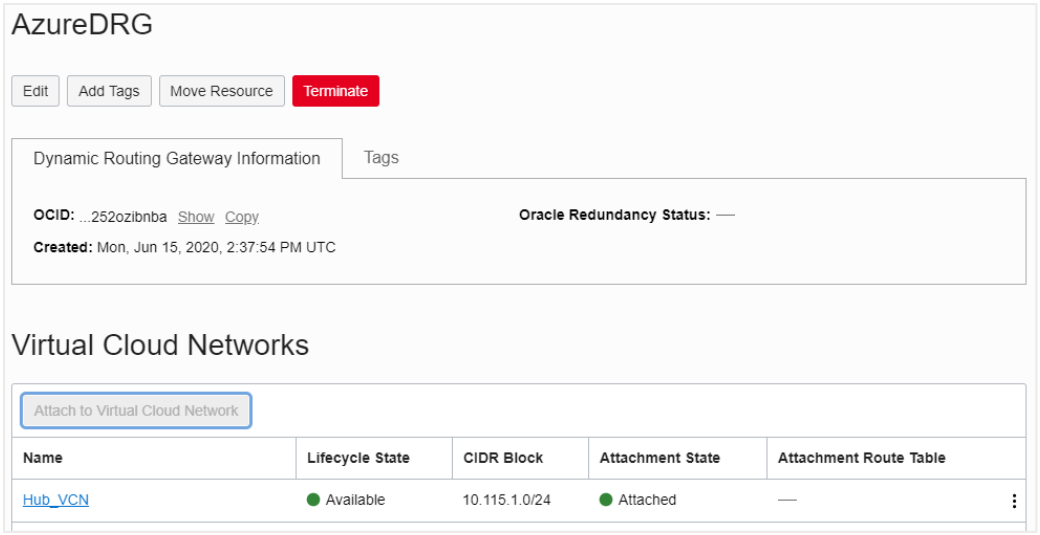

Figure 7: Attach the DRG to the Hub VCN

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.

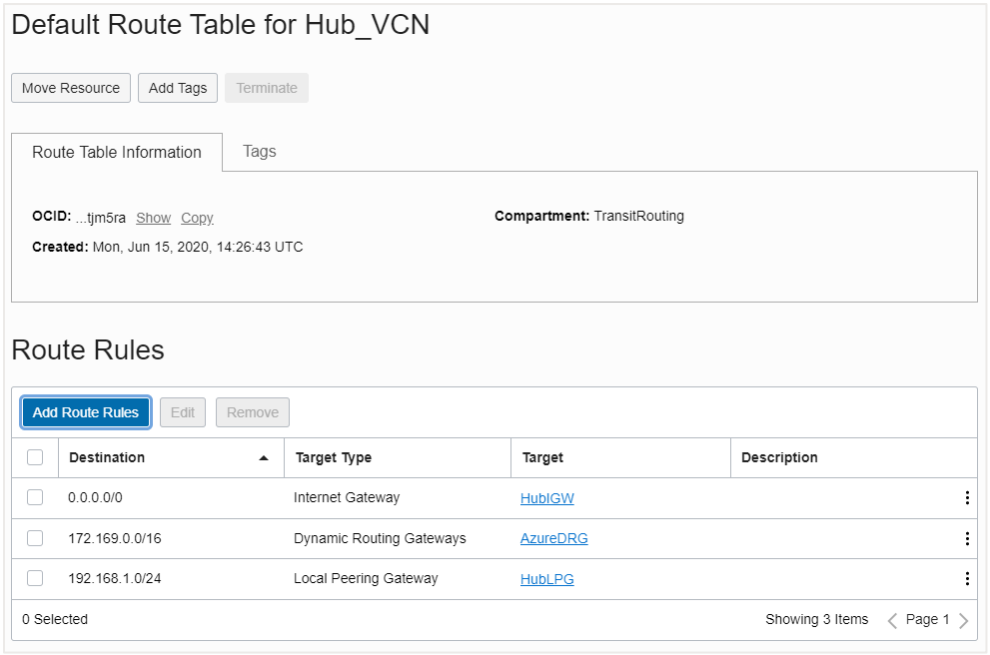

Figure 8: Create Route Rules for the Hub VCN's Default Route Table

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.

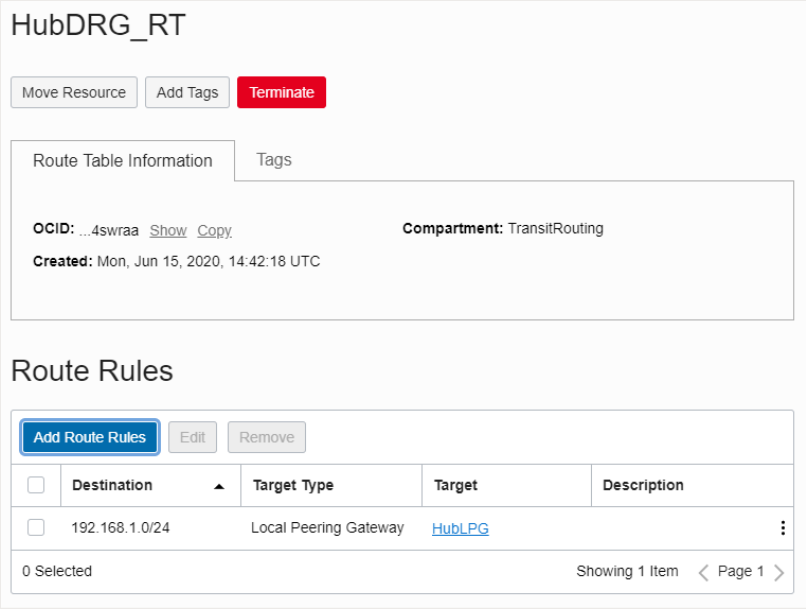

Figure 9: Create a Route Table for the DRG

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.

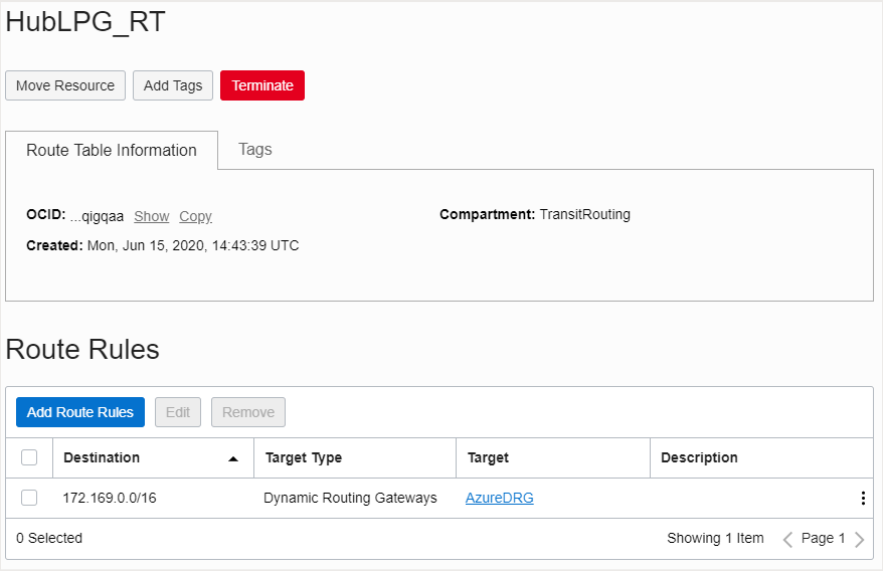

Figure 10: Create a Route Table for the Local Peering Gateway

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.

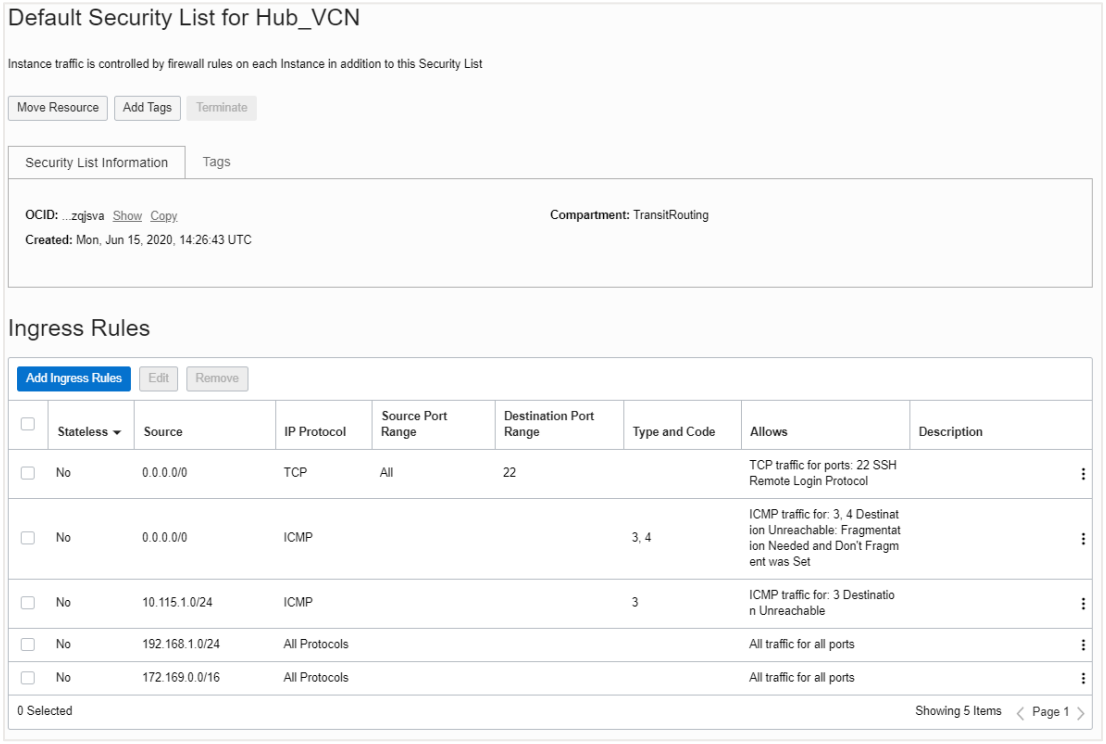

Figure 11: Create Security List Ingress Rules for the Hub VCN

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.

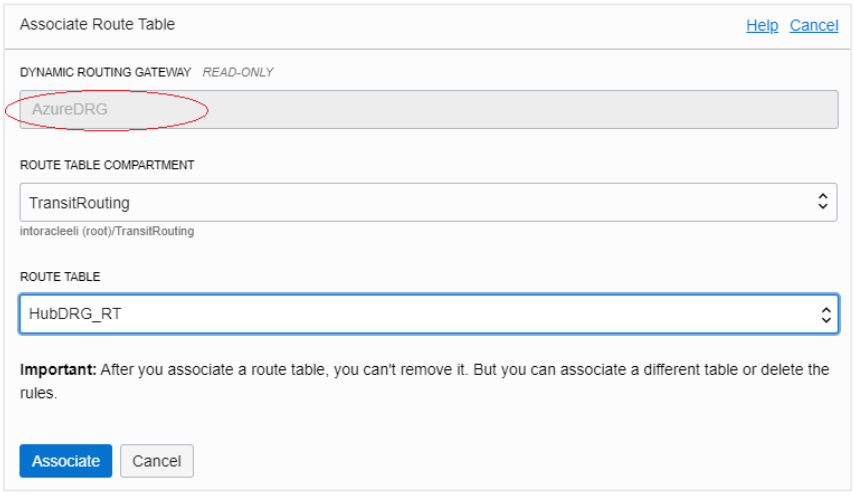

Figure 12: Associate a Route Table with the DRG

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

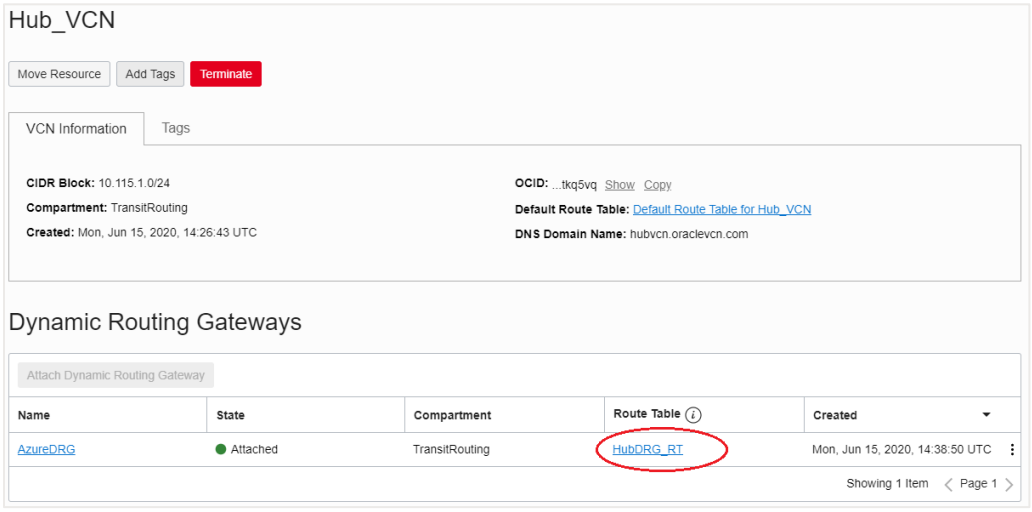

Figure 13: Verify the Route Table Attachment to the DRG

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.

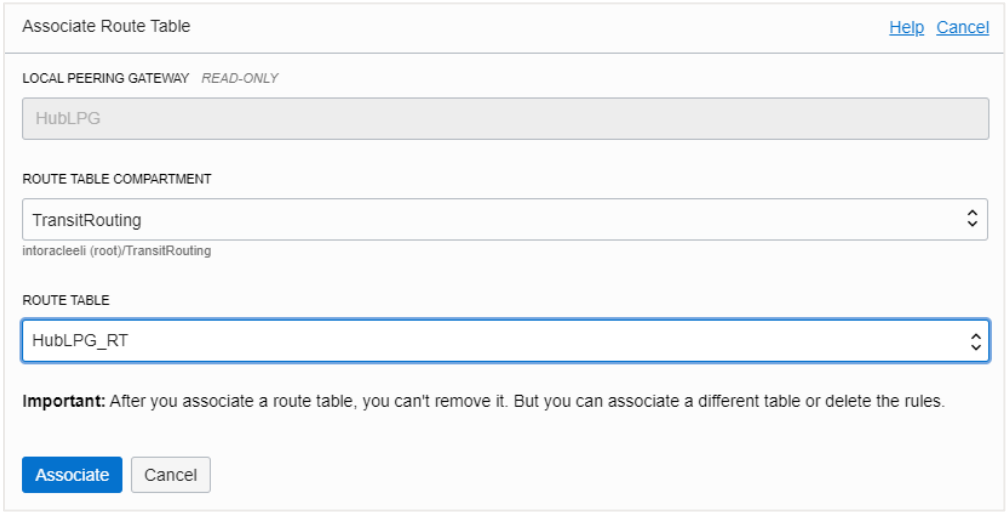

Figure 14: Associate a Route Table with the Local Peering Gateway

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

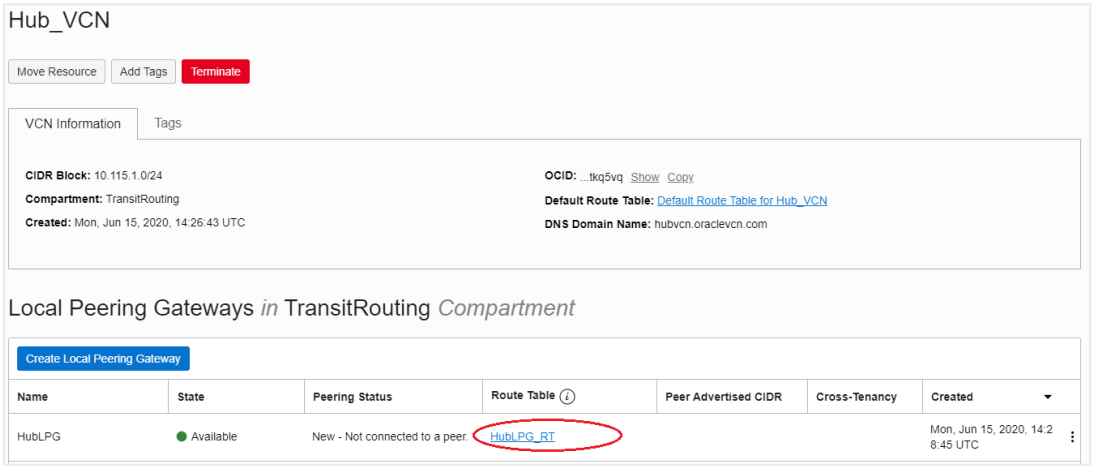

Figure 15: Verify the Route Table Attachment to the Local Peering Gateway

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.

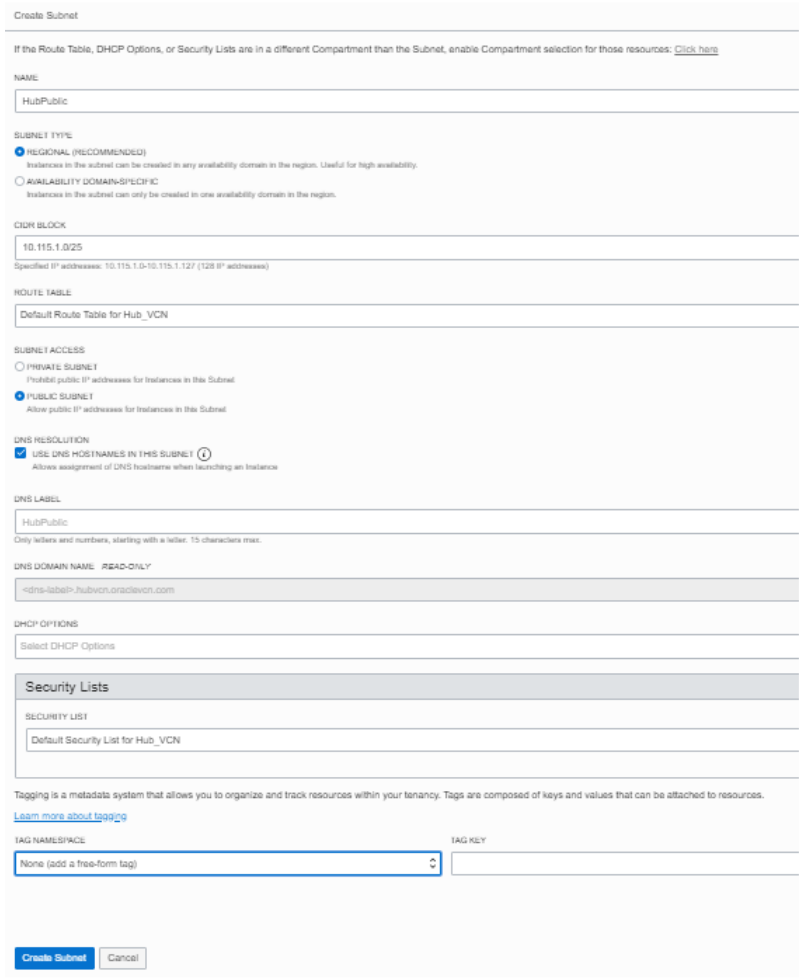

Figure 16: Create a Subnet in the Hub VCN

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.

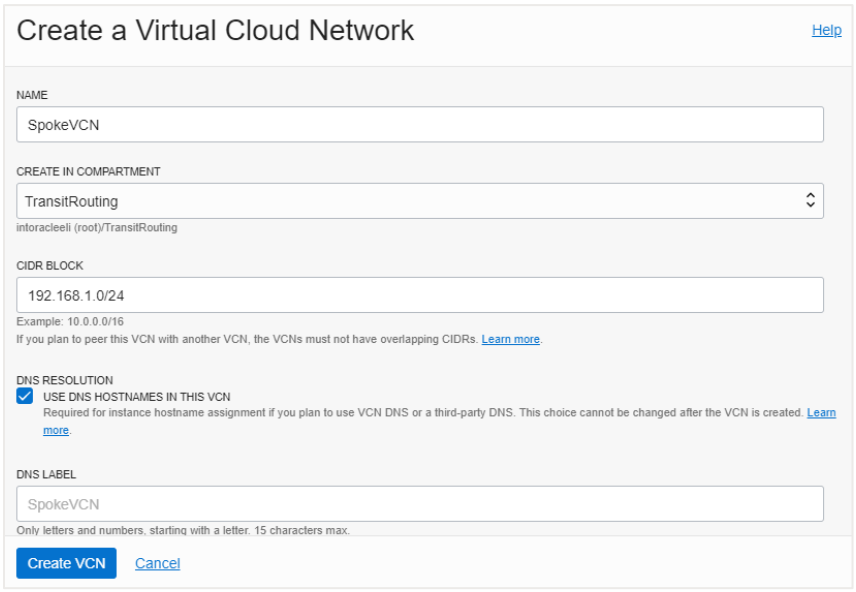

Figure 17: Create the Spoke VCN

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

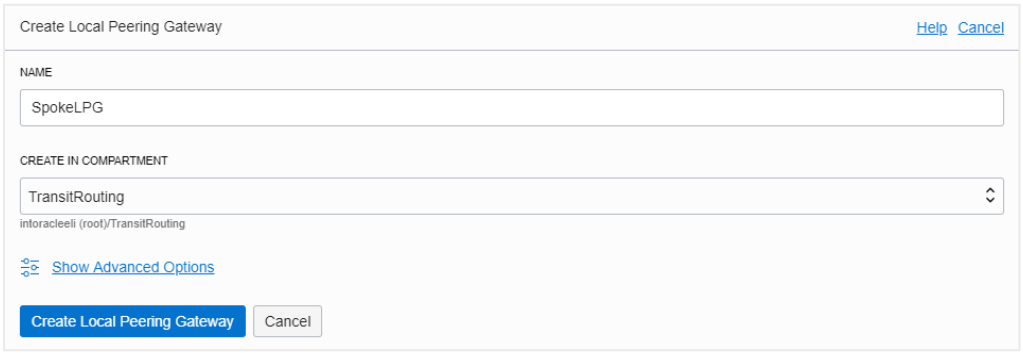

Figure 18: Create the Local Peering Gateway for the Spoke VCN

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

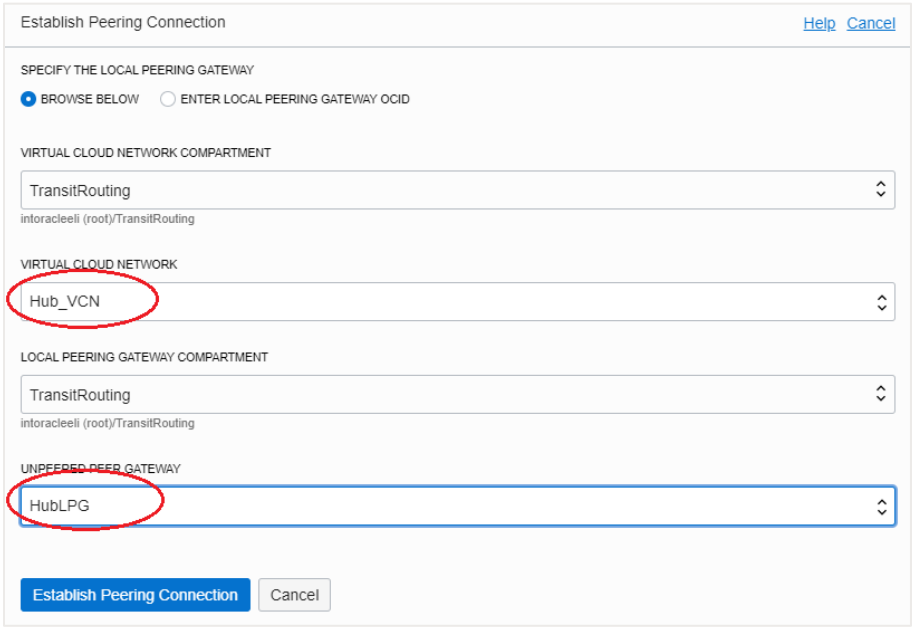

Figure 19: Create Local Peering from the Spoke VCN to the Hub VCN

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

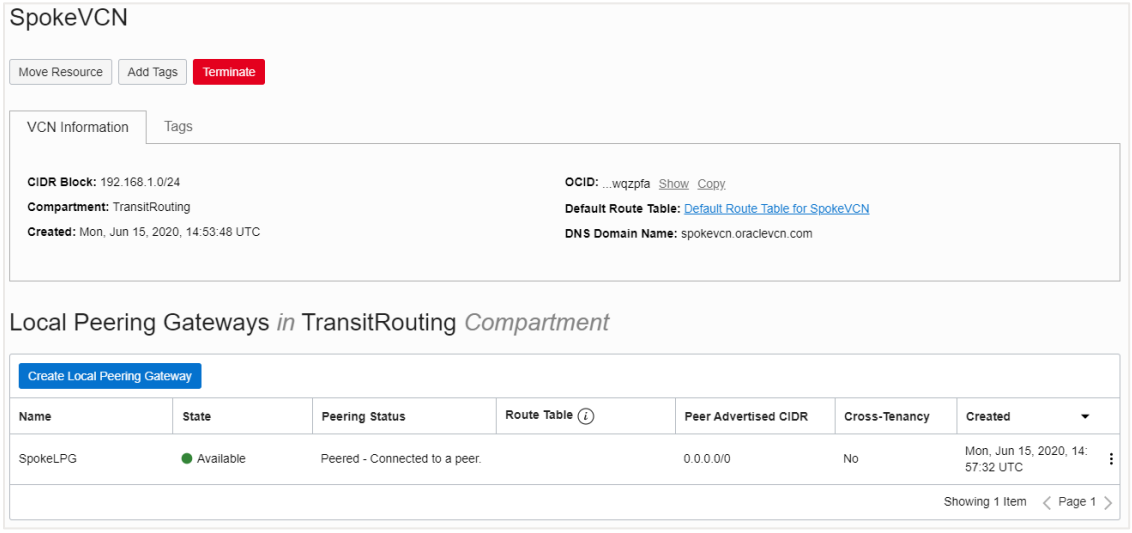

Figure 20: Verify Local Peering of Spoke and Hub VCNs

5. 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.

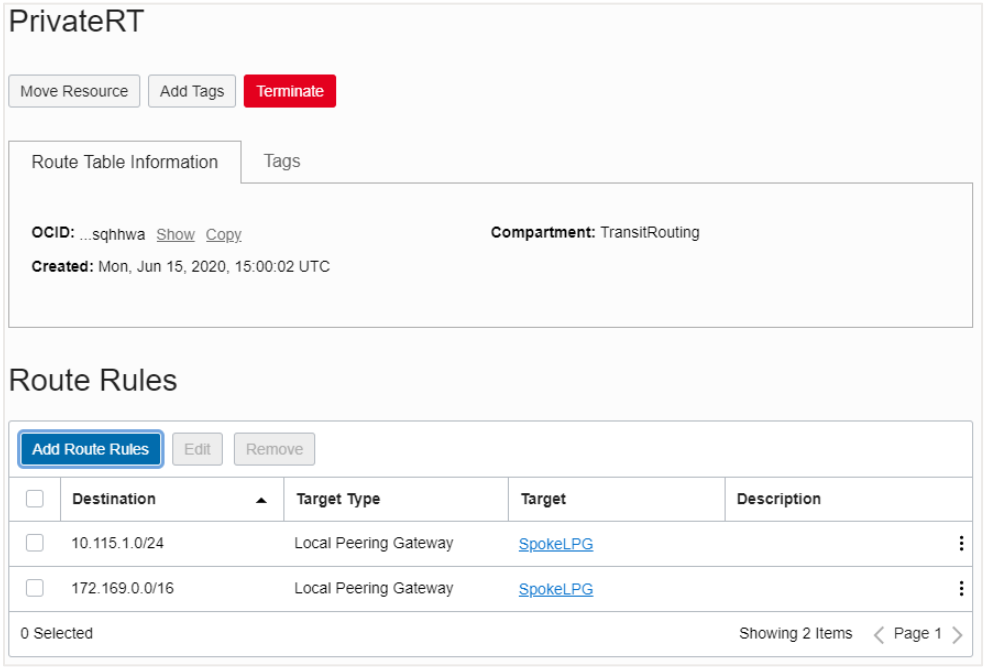

Figure 21: Create Route Table for Private Subnet

6. 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.

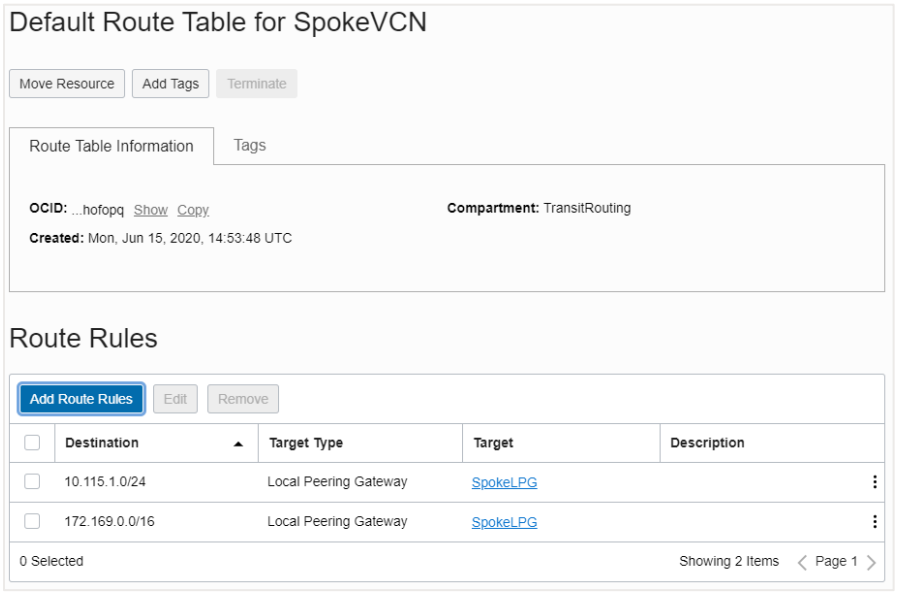

Figure 22: Create Route Rules for the Spoke VCN Default Route Table

7. 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.

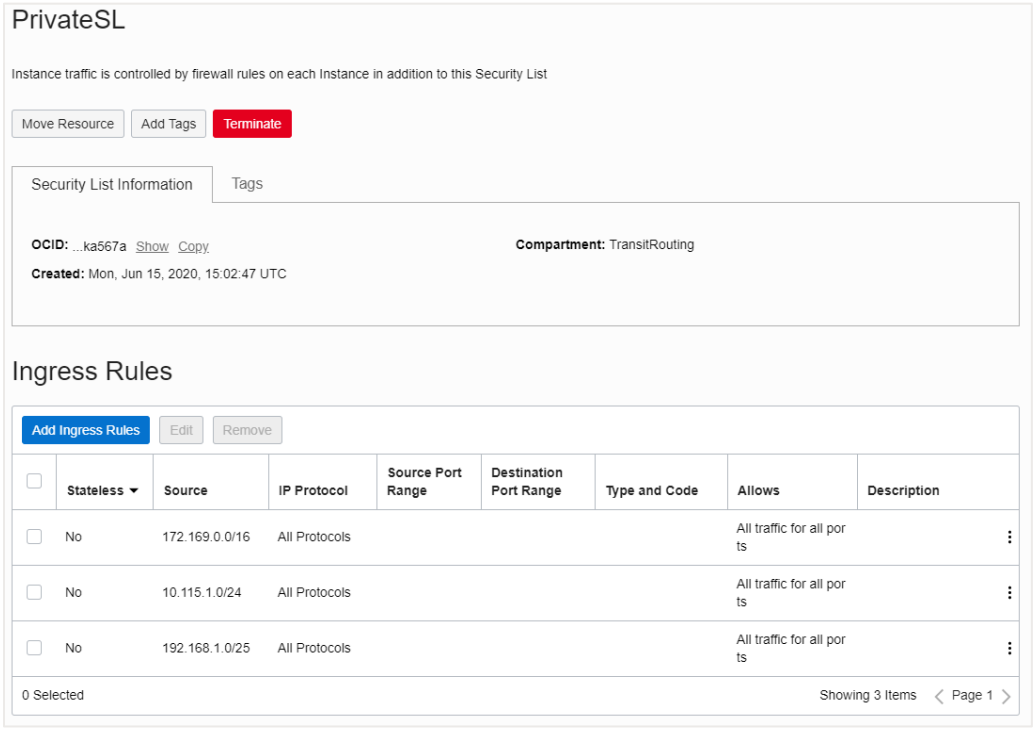

Figure 23: Create Ingress Rules for the Security List

8. 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.

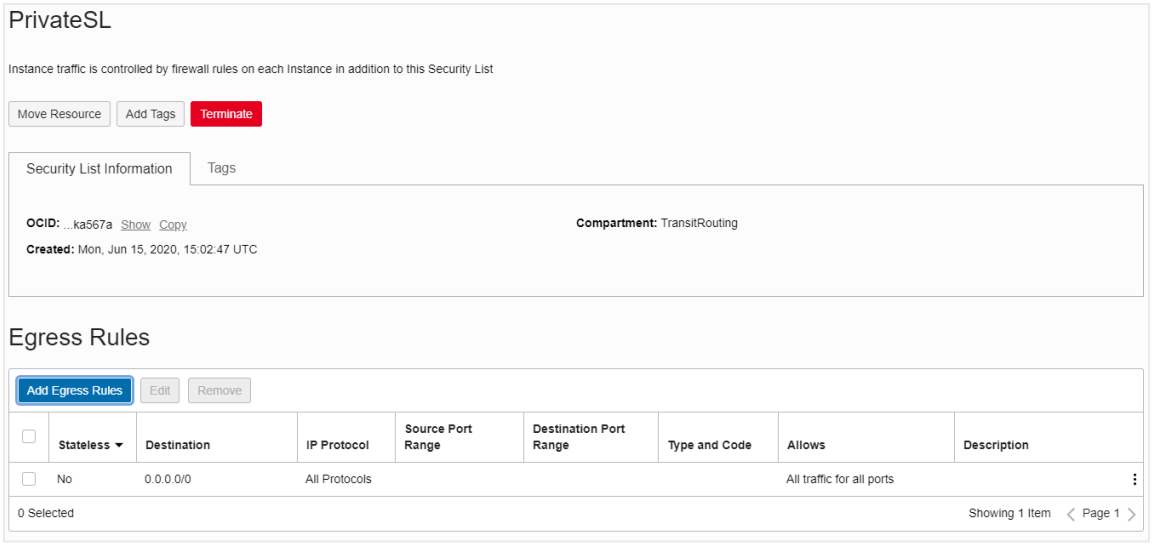

Figure 24: Create Egress Rules for the Security List

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.

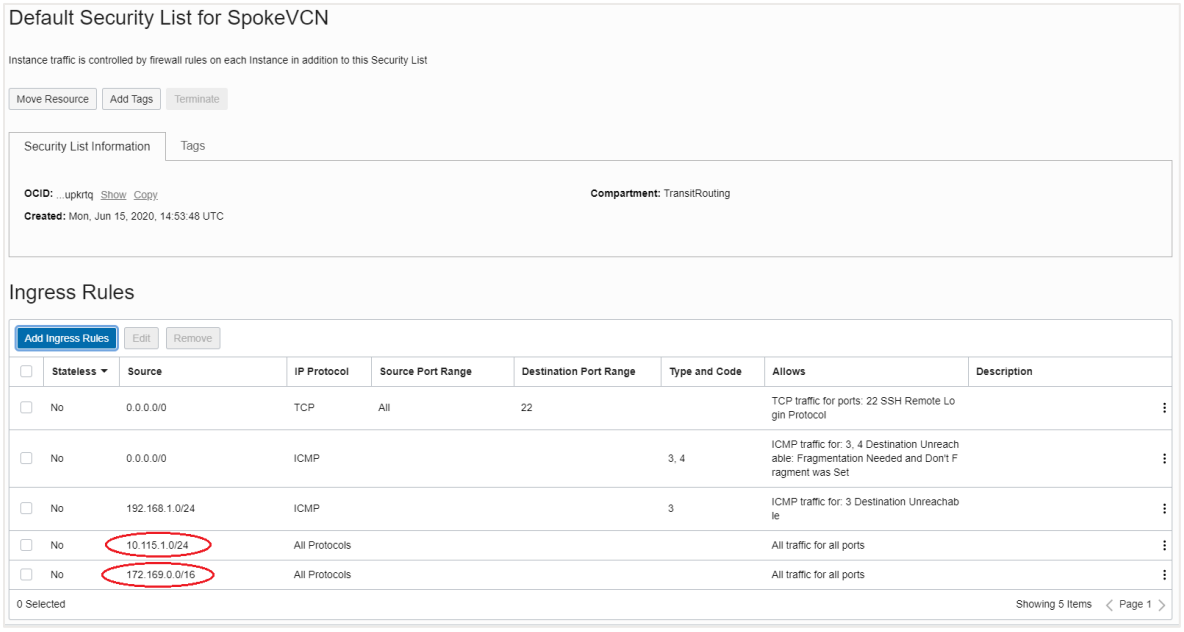

Figure 25: Create Ingress Rules for the Default Security List for the Spoke VCN

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

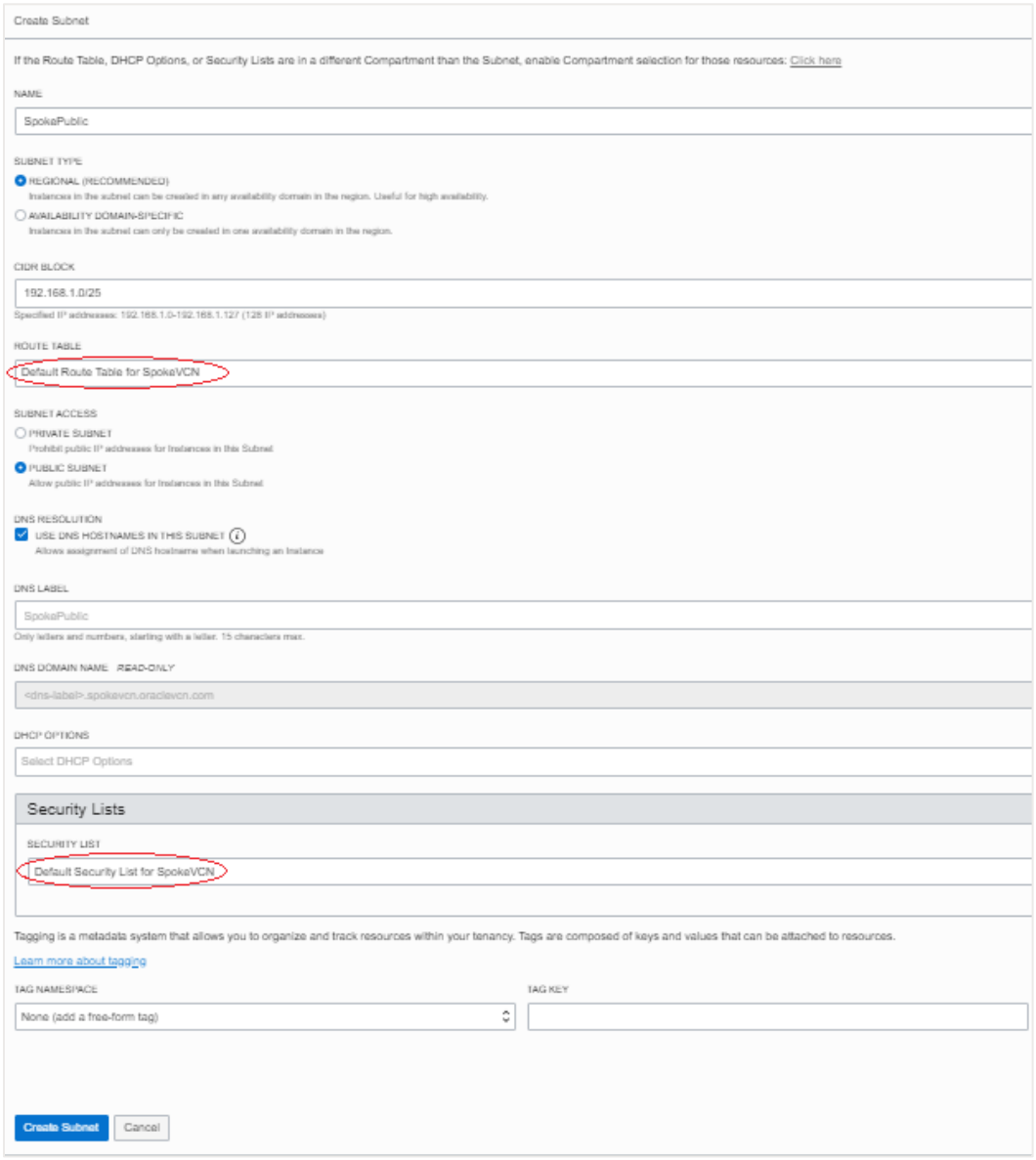

Figure 26: Create a Public Subnet for the Spoke VCN

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).

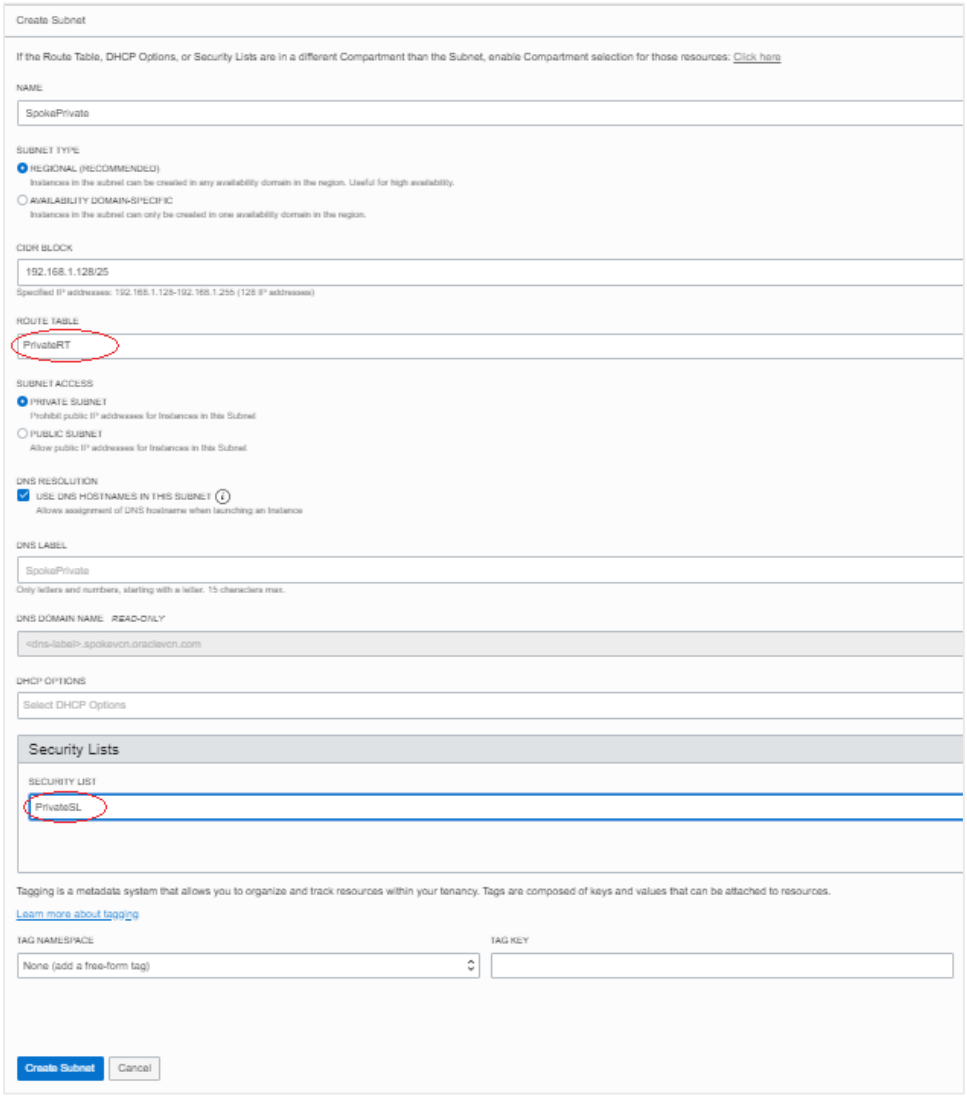

Figure 27: Create a Private Subnet for the Spoke VCN

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.

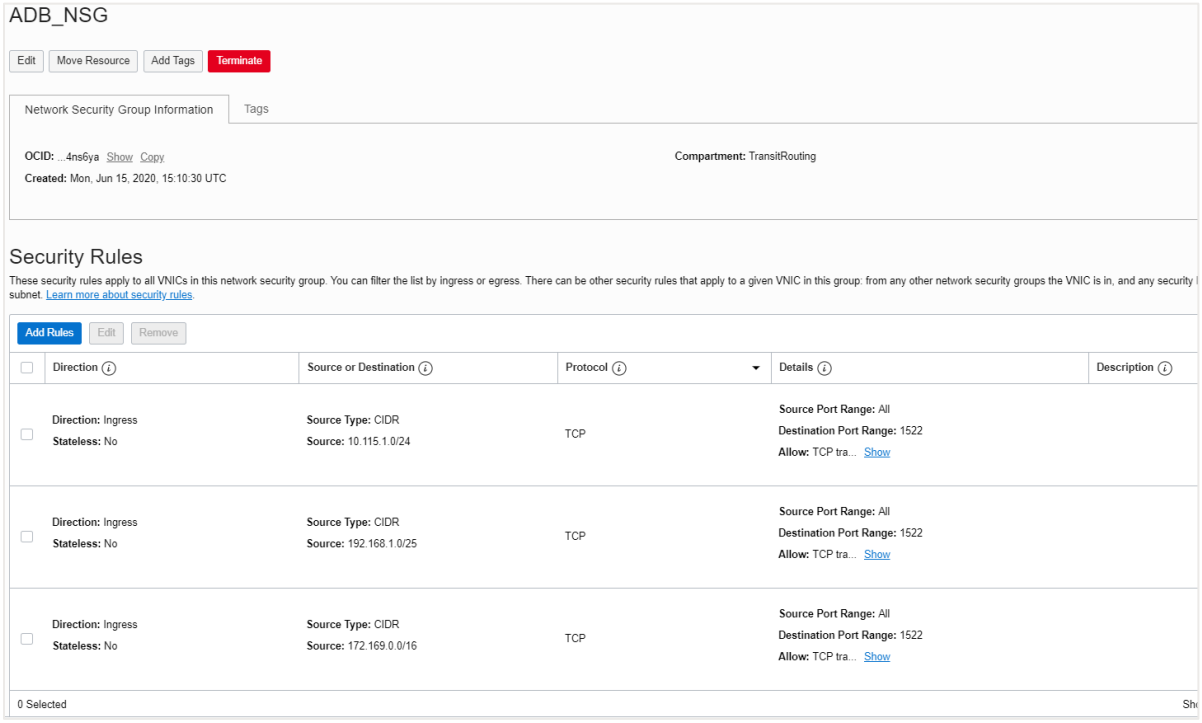

Figure 28: Create a Network Security Group for the Spoke VCN

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.

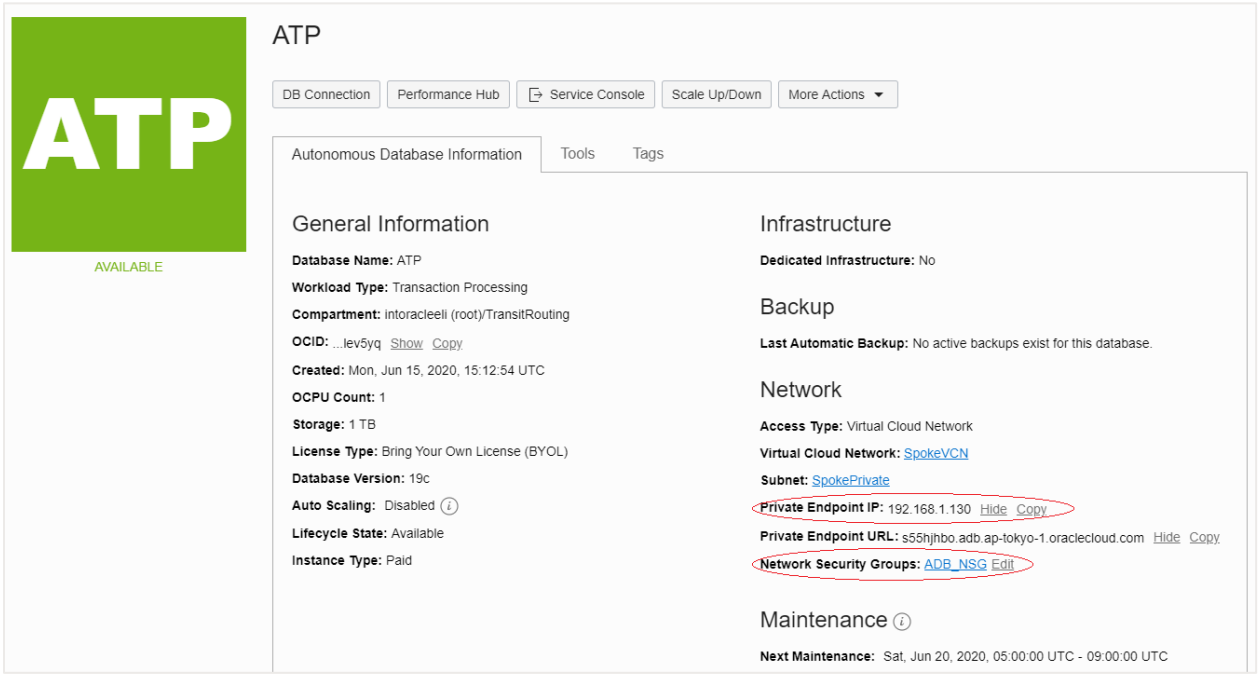

Figure 29: Details Page for an Autonomous Database

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).

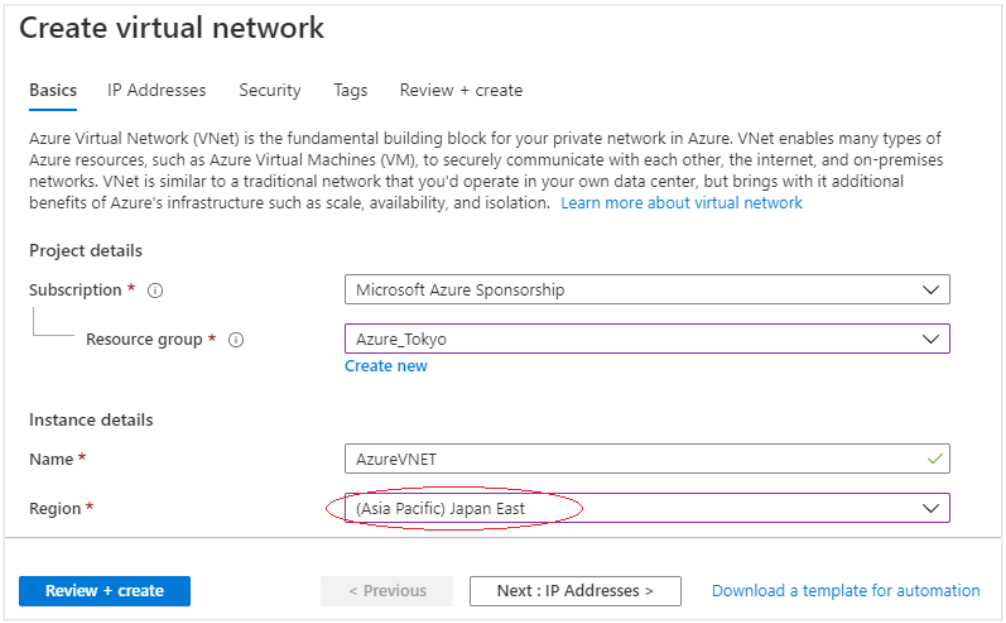

Figure 30: Create VNet in Azure, Specify Region

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

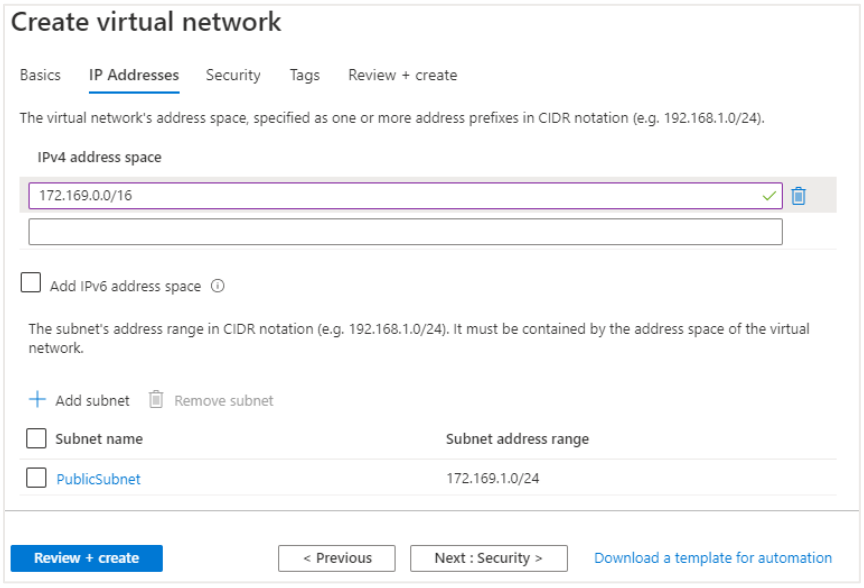

Figure 31: Create VNet in Azure, Specify CIDR Address Block

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

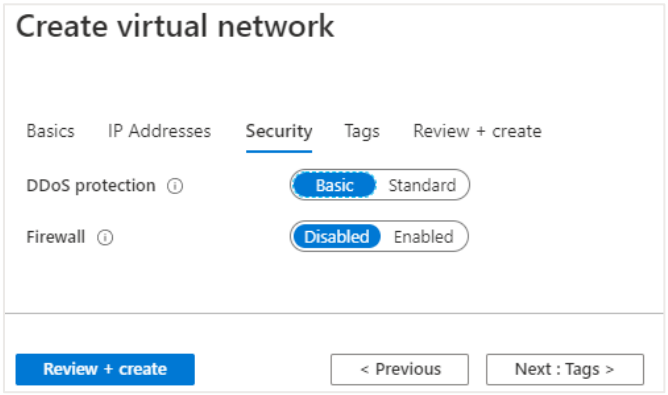

Figure 32: Create VNet in Azure, Specify Security Settings

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

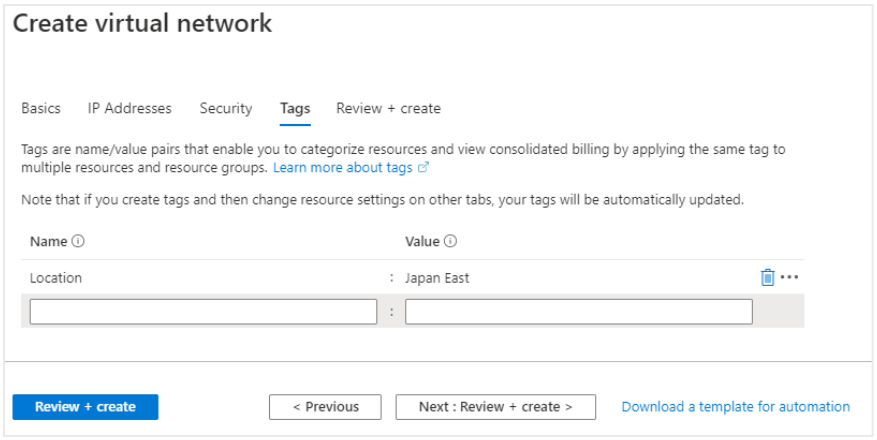

Figure 33: Create VNet in Azure, Specify Tags

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

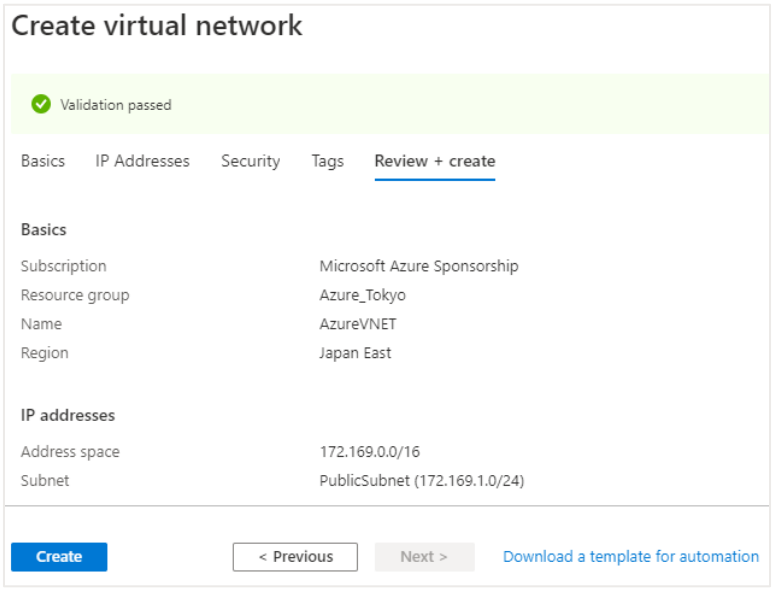

Figure 34: Create VNet in Azure, Review Information

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.

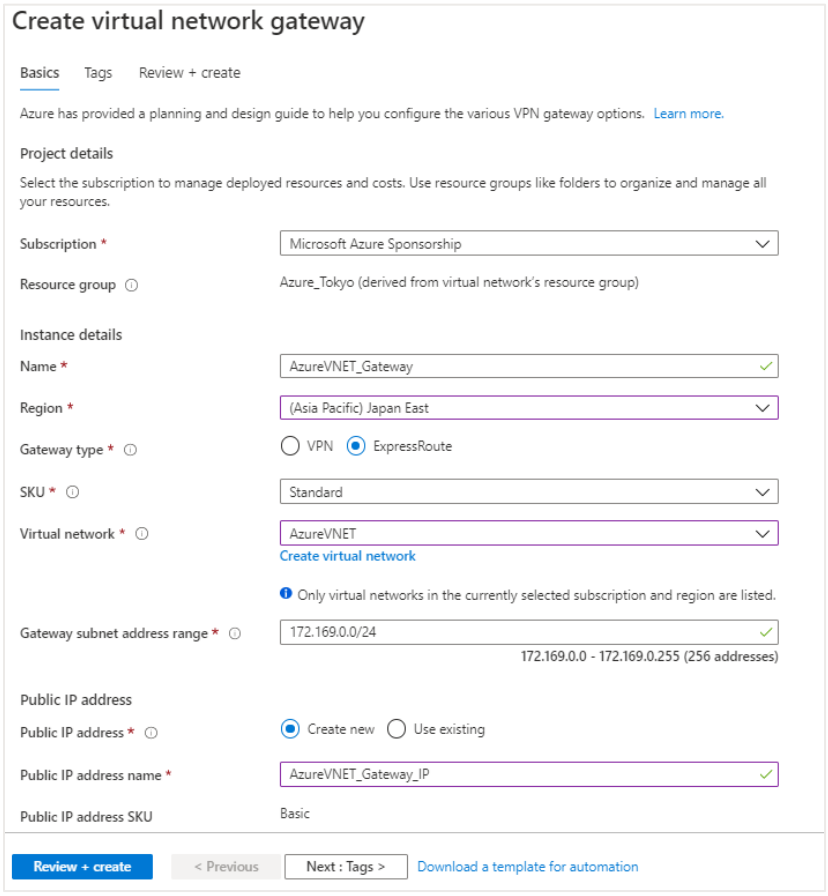

Figure 35: Create VNet Gateway

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.

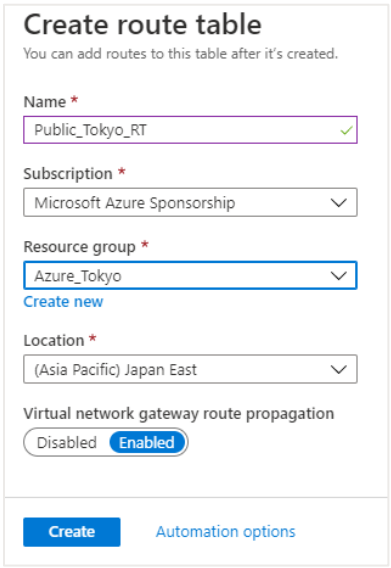

Figure 36: Create Route Table

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

   You add individual rules in a later step.

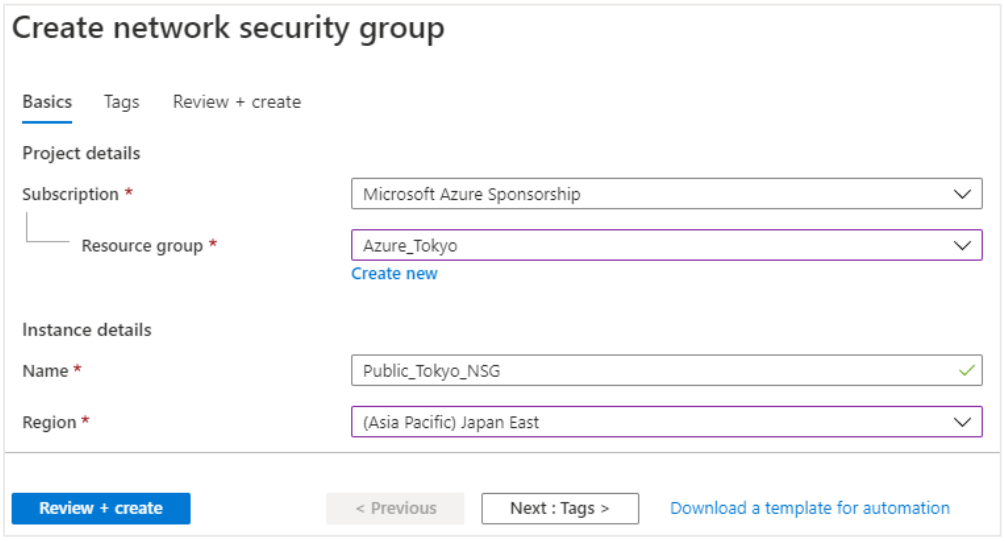

Figure 37: Create Network Security Group

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.

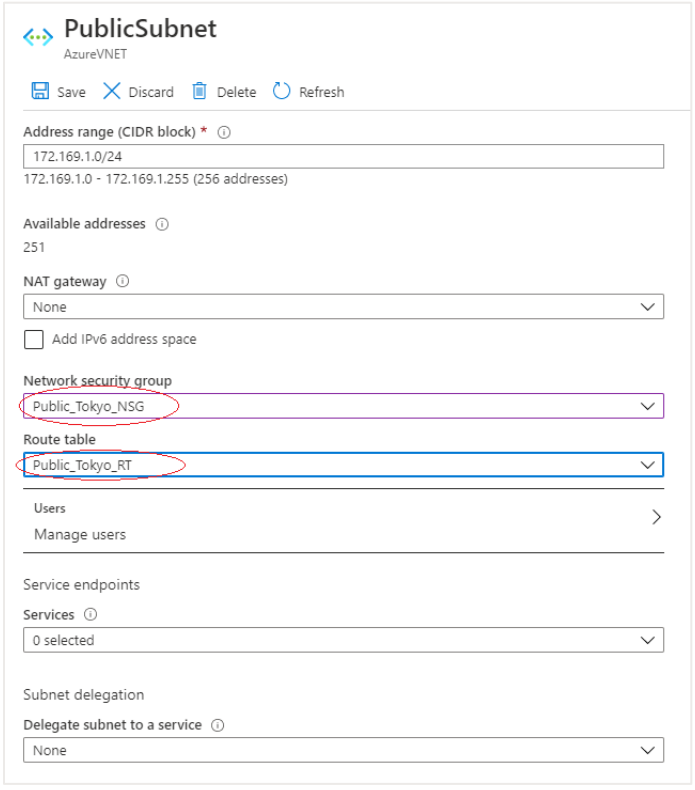

Figure 38: Associate Route Table and Network Security Group with Subnet

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

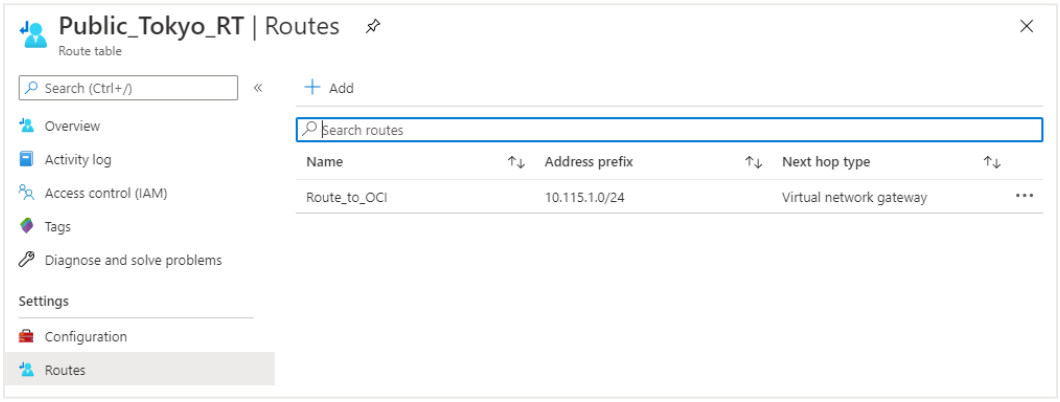

Figure 39: Modify Route Table with CIDR Address Block of the Hub VCN

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

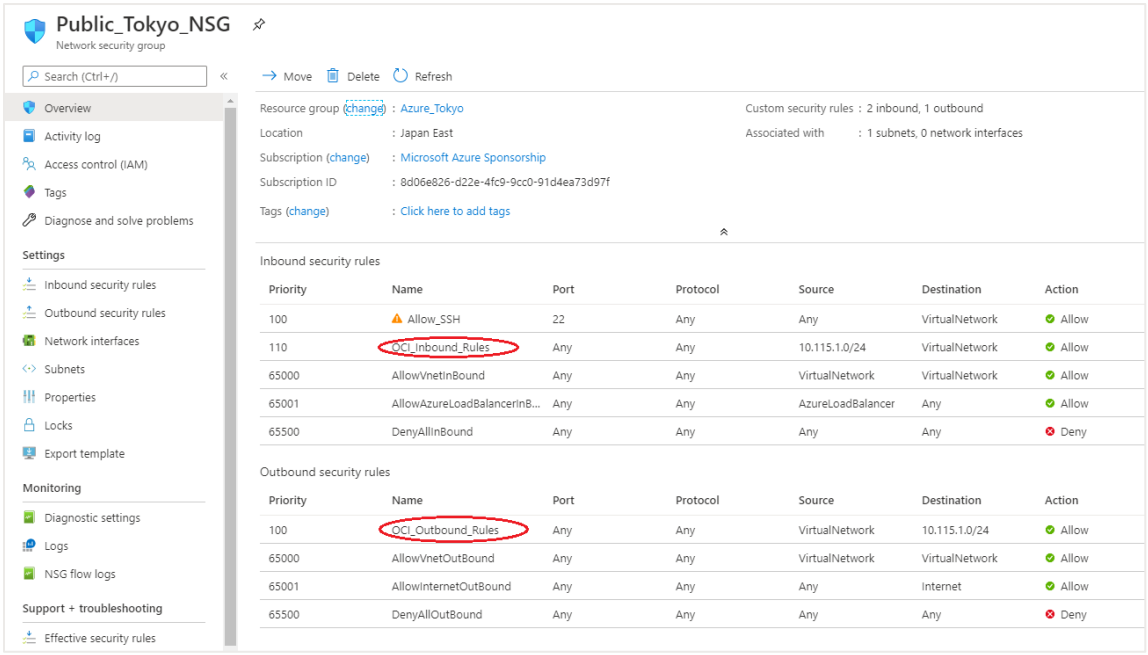

Figure 40: Modify Network Security Group with Traffic Rules to the Hub VCN

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.

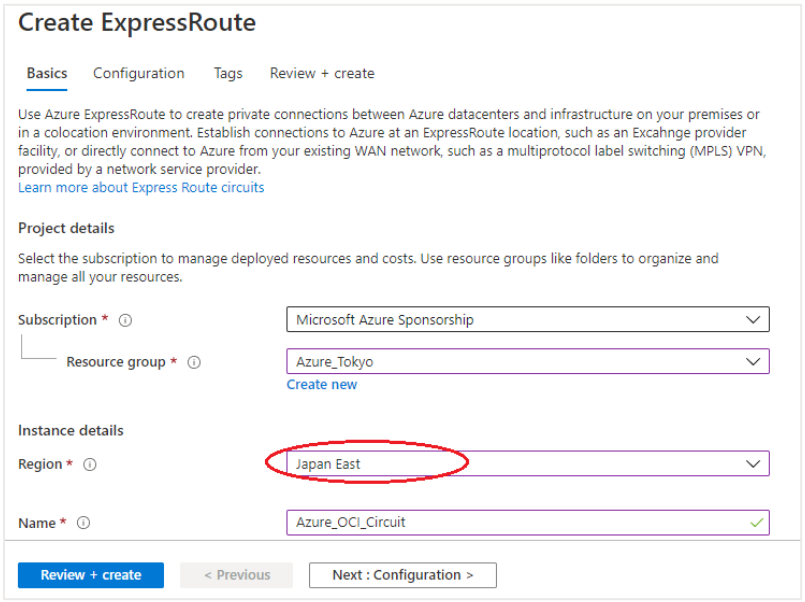

Figure 41: Create an ExpressRoute Connection in Azure

   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.

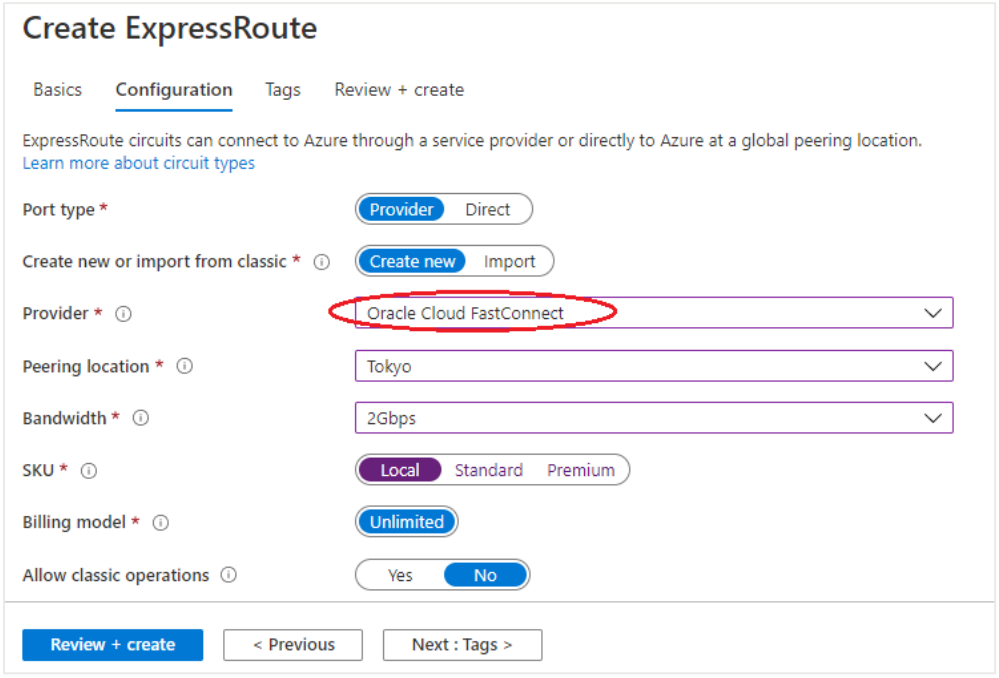

Figure 42: Configure the ExpressRoute Circuit

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

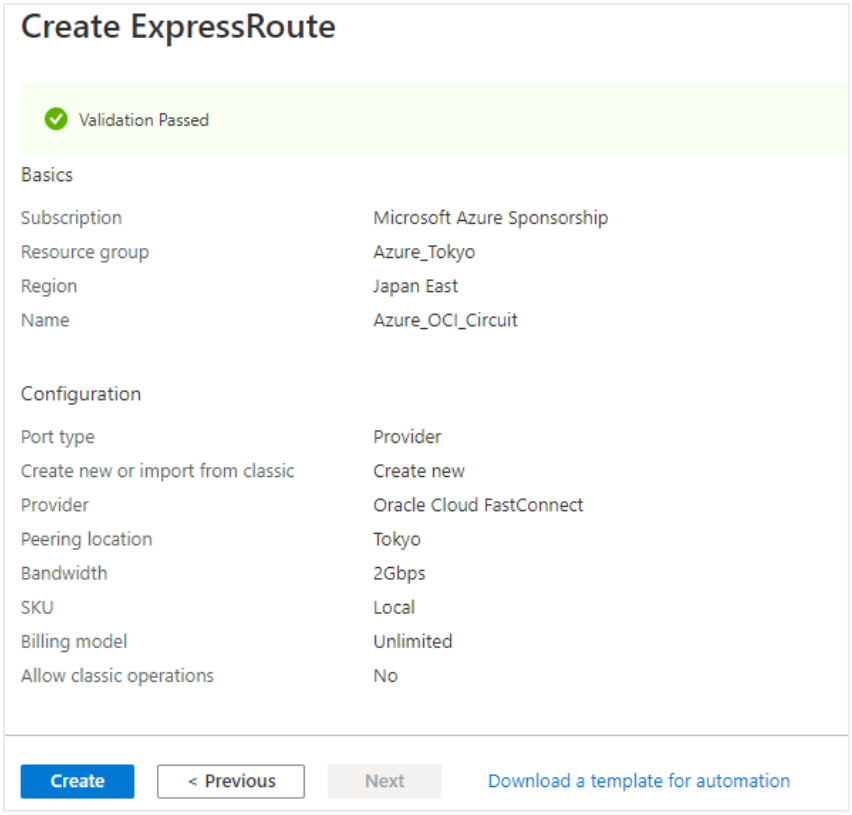

Figure 43: Verify the ExpressRoute Connection and Create

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.

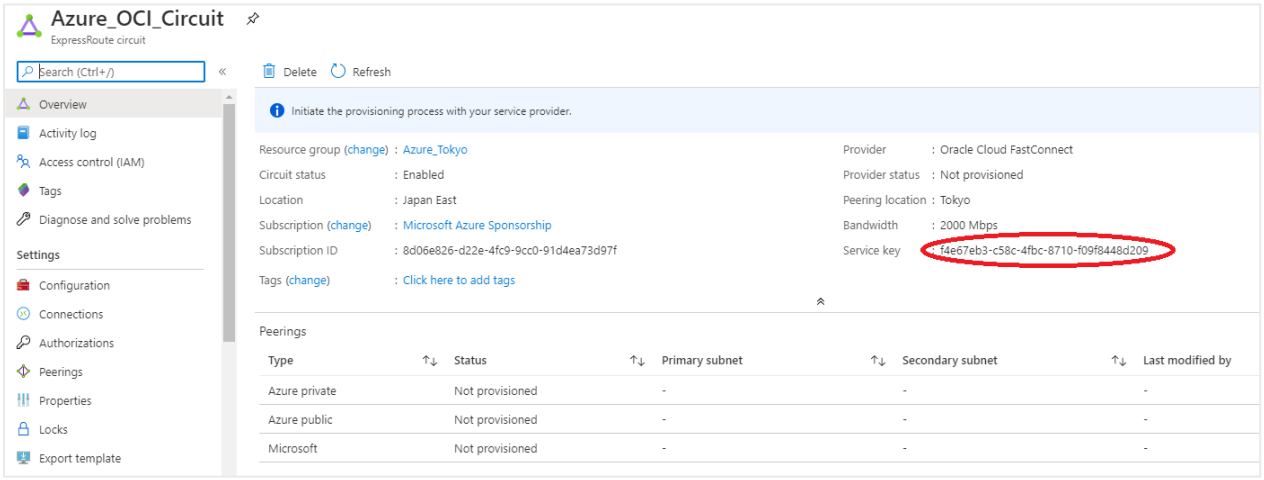

Figure 44: ExpressRoute Details Page Showing the Service Key

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.

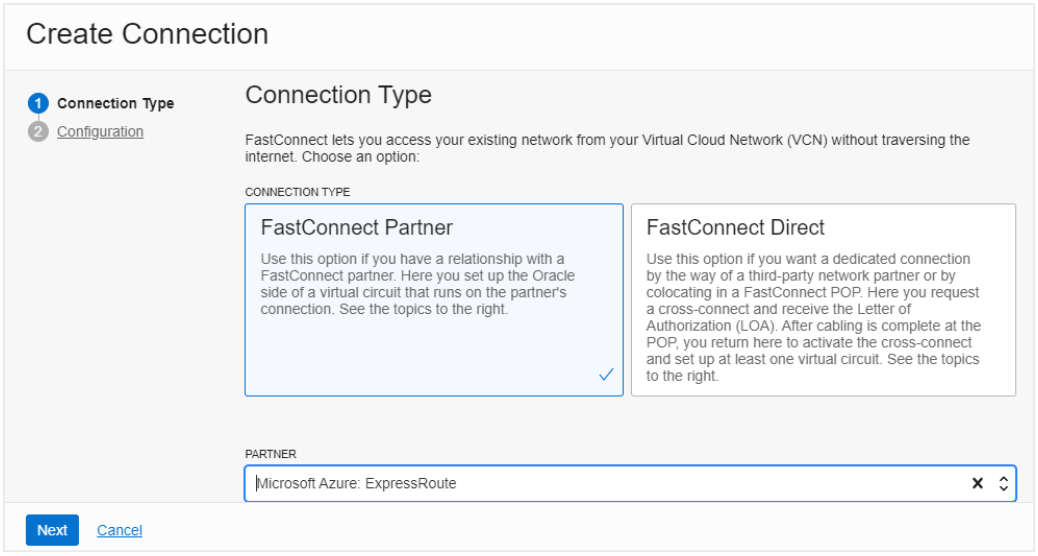

Figure 45: Create FastConnect Connection to Azure, Specify Connection Type

   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).

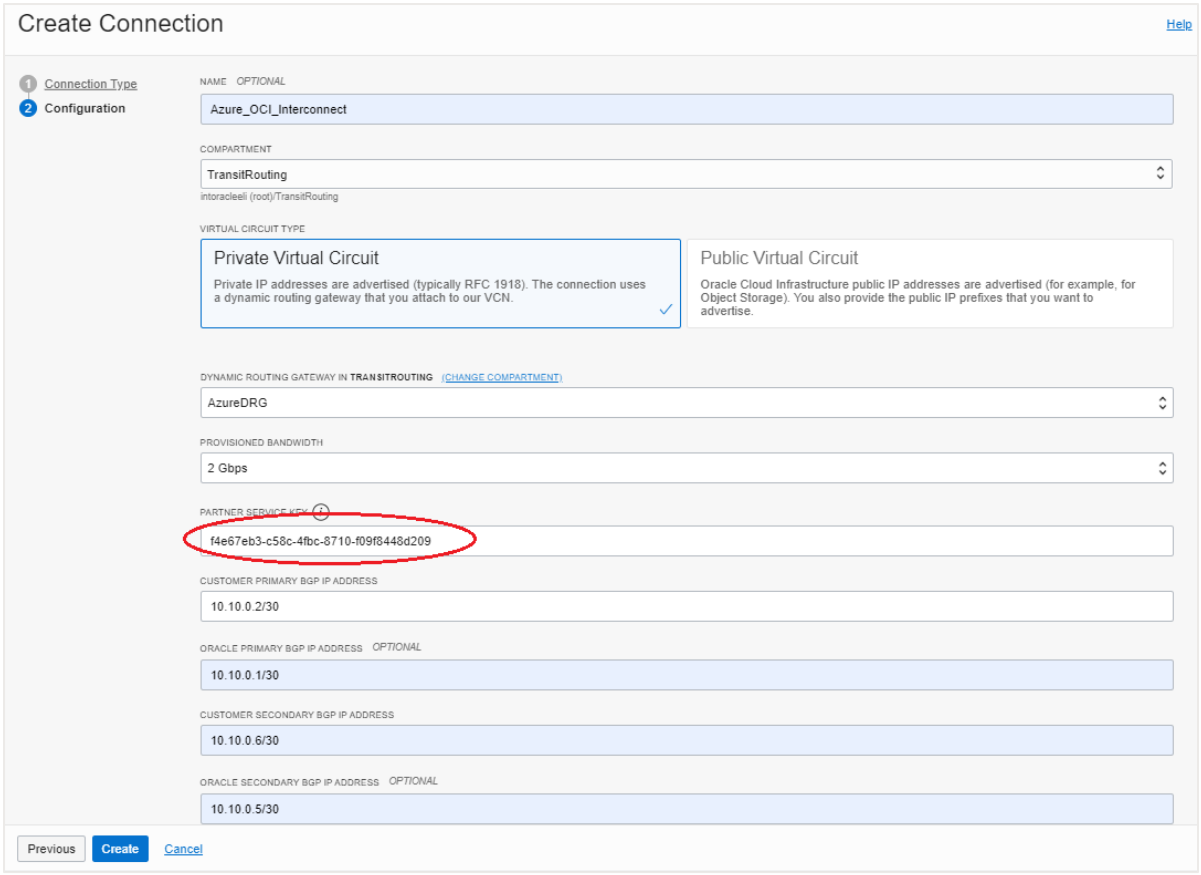

Figure 46: Create FastConnect Connection to Azure, Specify DRG and Service Key

C. Click **Create**.

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

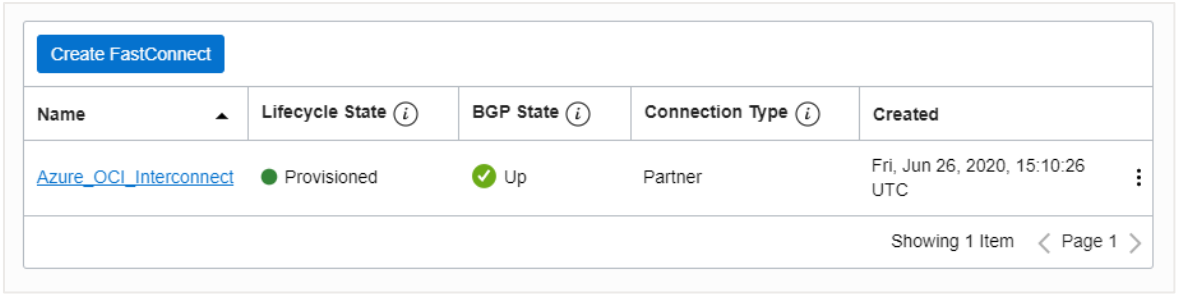

Figure 47: FastConnect Connection Provisioned and Up

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

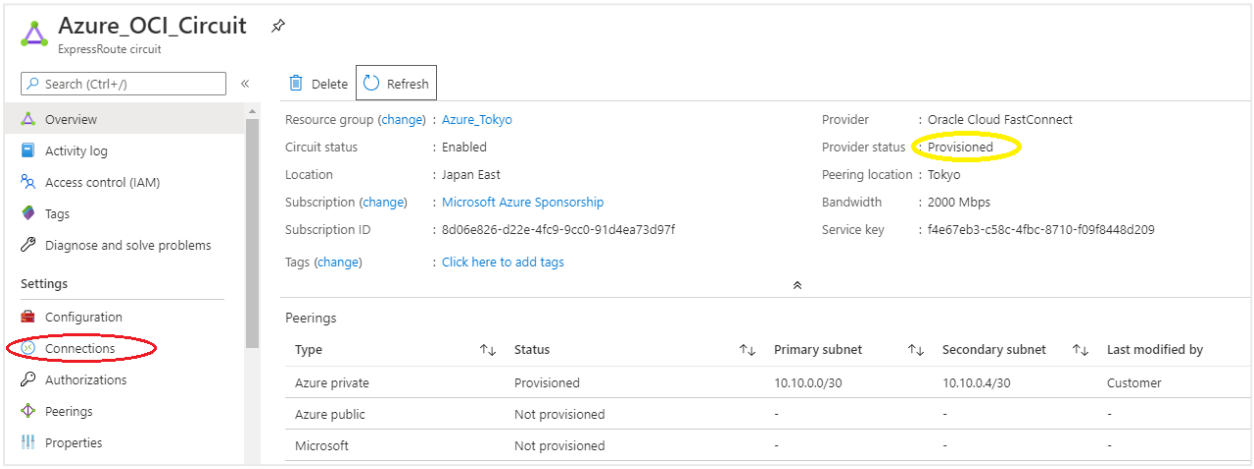

Figure 48: Verify Status of Provider on ExpressRoute Details Page

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.

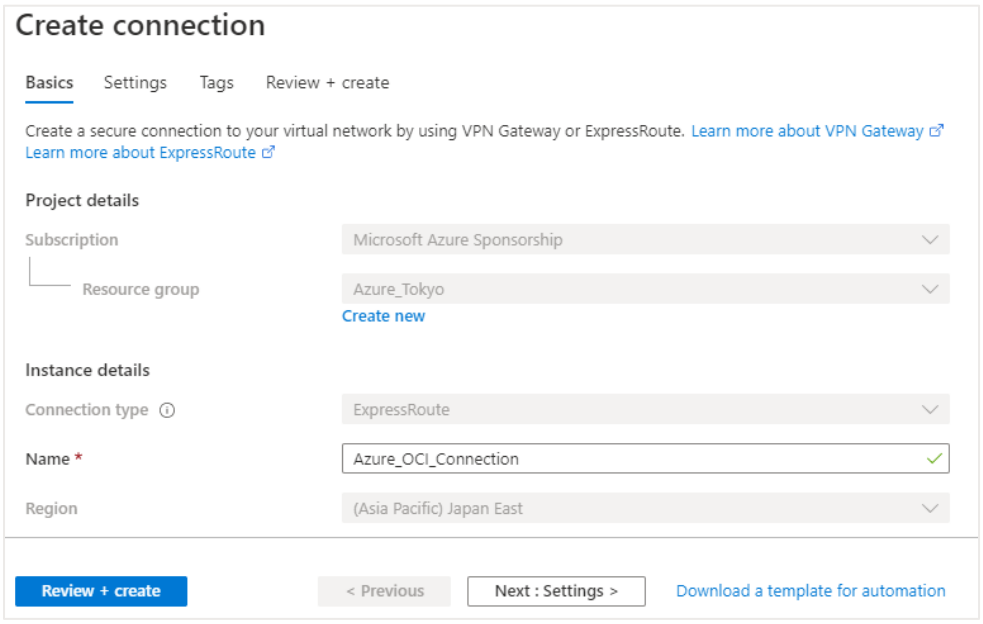

Figure 49: Create Connection to Attach ExpressRoute to VNet Gateway

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

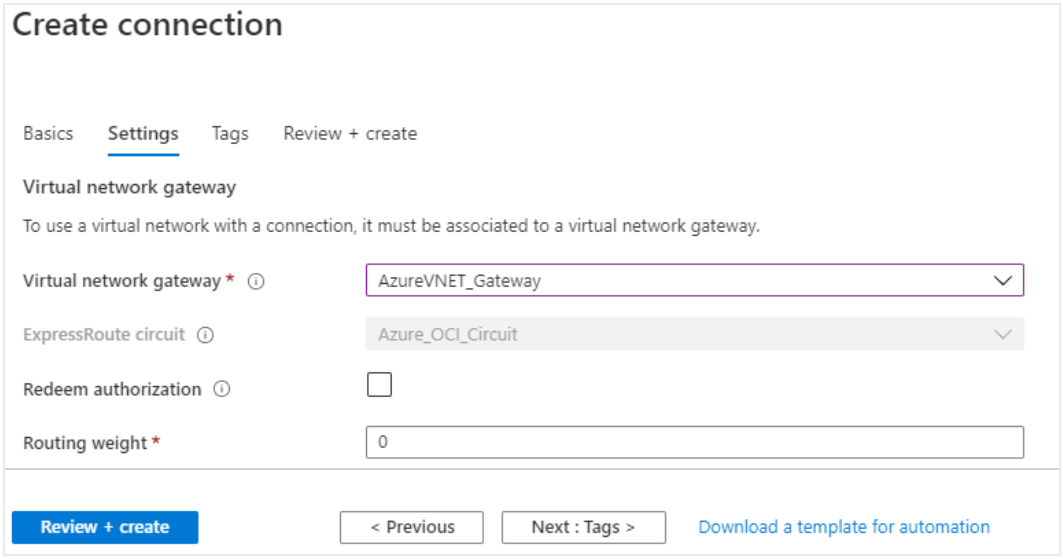

Figure 50: Create Connection to Attach ExpressRoute to VNet Gateway, Select VNet Gateway

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

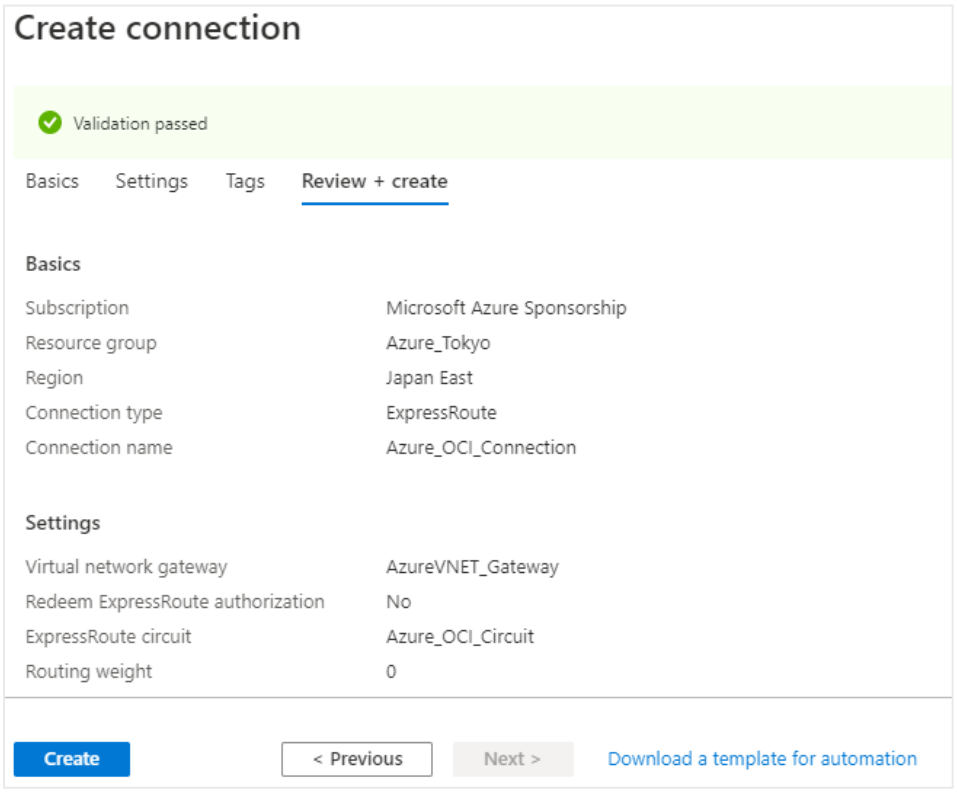

Figure 51: Create Connection to Attach ExpressRoute to VNet Gateway, Verify Options

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

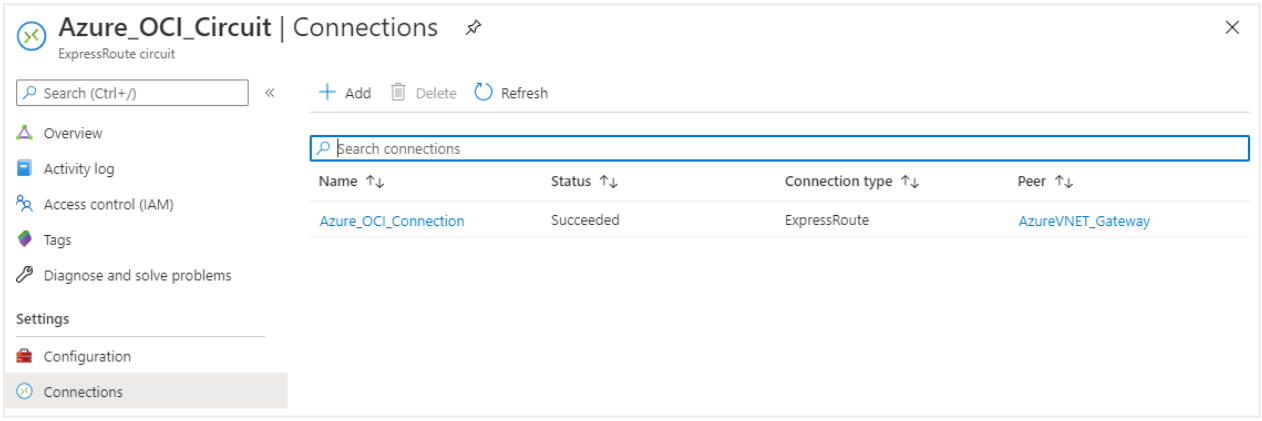

Figure 52: Verify Connection Is Added

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.

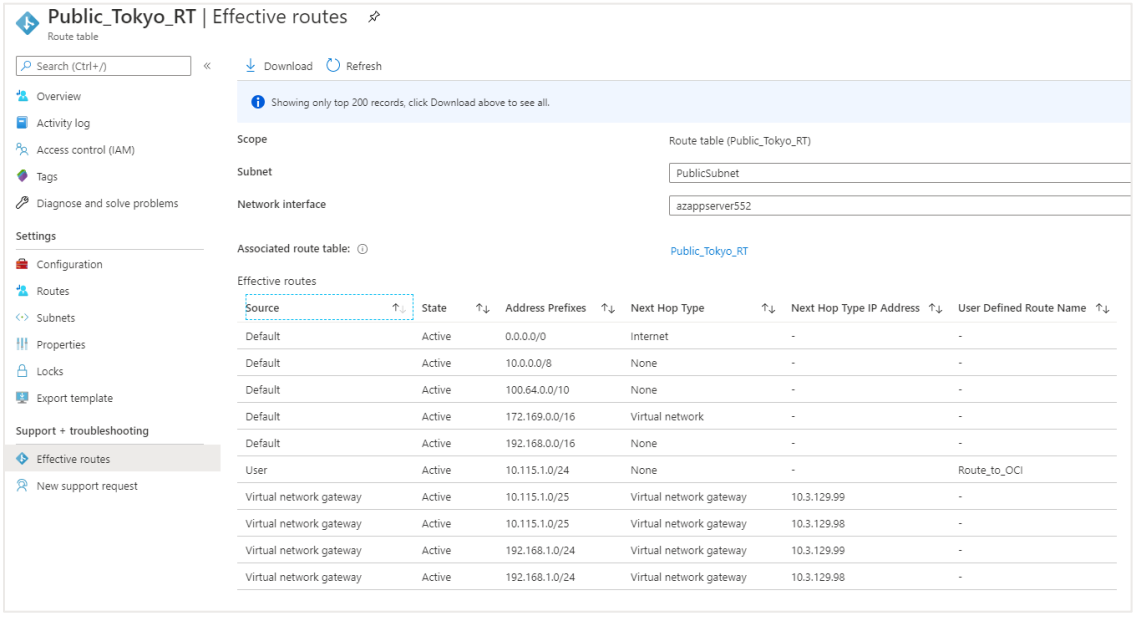

Figure 53: Effective Routes Between Azure and Oracle Cloud

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:fecd: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.

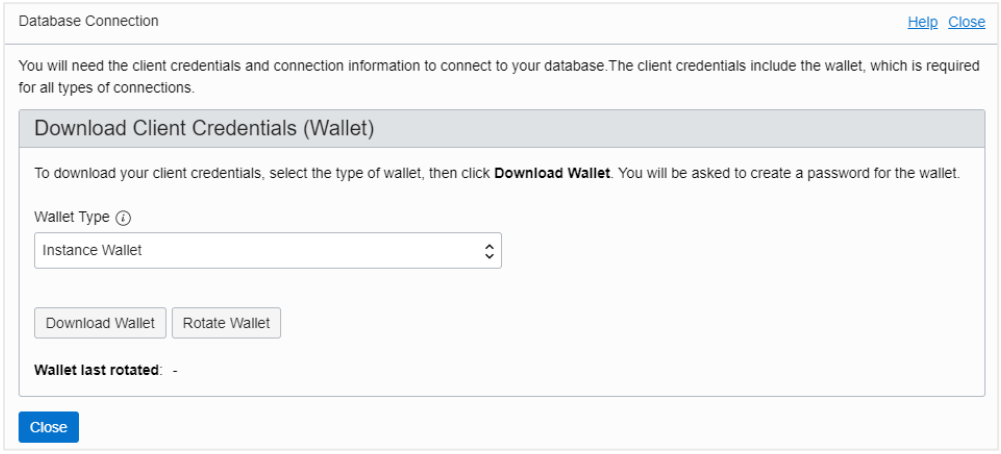

Figure 55: Select the Oracle Wallet Type for the Autonomous Database

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

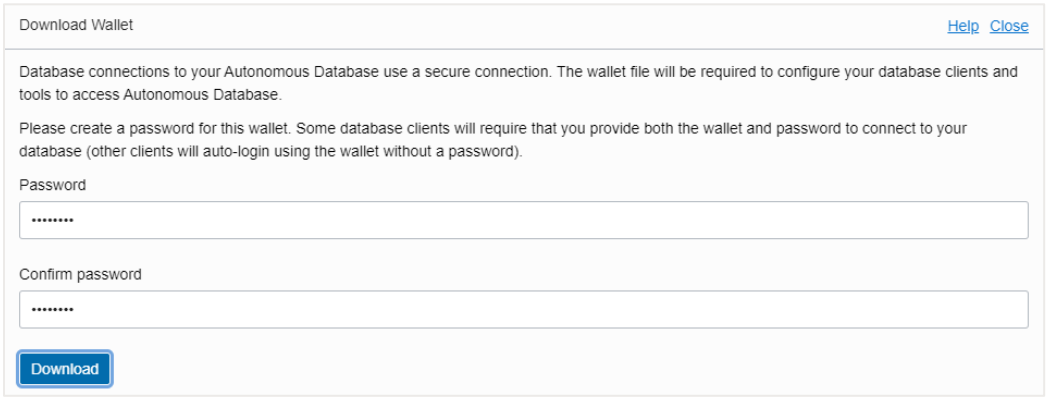

Figure 56: Download Oracle Wallet for the Autonomous Database

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` ping 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=tcps)(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