Tell us about your PDF experience.

Overview of Oracle Applications and solutions on Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs

In this article, you learn about running Oracle solutions using the Azure infrastructure.

> ⓘ **Important**
>
> Oracle RAC and Oracle RAC OneNode are not supported in Azure Bare Metal Infrastructure.

Oracle databases on Azure infrastructure

Oracle supports running its Database 12.1 and higher Standard and Enterprise editions in Azure on VM images based on Oracle Linux. You can run Oracle databases on Azure infrastructure using Oracle Database on Oracle Linux images available in the Azure Marketplace.

- Oracle Database 12.2, and 18.3 Enterprise Edition
- Oracle Database 12.2, and 18.3 Standard Edition
- Oracle Database 19.3
  You can also take one of the following approaches:
- Set up Oracle Database on a non-Oracle Linux image available in Azure.
- Build a solution on a custom image you create from scratch in Azure.
- Upload a custom image from your on-premises environment.

You can also choose to configure your solution with multiple attached disks. You can improve database performance by installing Oracle Automated Storage Management (ASM). For the best performance for production workloads of Oracle Database on Azure, be sure to properly size the VM image and select the right storage options based on throughput, IOPS & latency. For instructions on how to quickly get an Oracle Database up and running in Azure using the Oracle published VM image, see Create an Oracle Database in an Azure VM.

Deploy Oracle VM images on Microsoft Azure

This section covers information about Oracle solutions based on virtual machine (VM) images published by Oracle in the Azure Marketplace. To get a list of currently available Oracle images, run the following command using Azure CLI or Azure Cloud Shell

```
az vm image list --publisher oracle --output table –all
```

The images are bring-your-own-license. You're charged only for the costs of compute, storage, and networking incurred running a VM. You can also choose to build your solutions on a custom image that you create from scratch in Azure or upload a custom image from your on-premises environment.

> **Important**
>
> You require a proper license to use Oracle software and a current support agreement with Oracle. Oracle has guaranteed license mobility from on-premises to Azure. For more information about license mobility, see the Oracle and Microsoft Strategic Partnership FAQ.

Applications on Oracle Linux and WebLogic server

Run enterprise applications on WebLogic server in Azure on supported Oracle Linux images. For more information, see the WebLogic documentation,Oracle WebLogic Server on Azure Solution Overview.

WebLogic Server with Azure service integrations

Oracle and Microsoft are collaborating to bring WebLogic Server to the Azure Marketplace in the form of Azure Application offering. For more information about these offers, see What are solutions for running Oracle WebLogic Server.

Oracle WebLogic Server VM images

Clustering is supported on Enterprise Edition only. You're licensed to use WebLogic clustering only when you use the Enterprise Edition of Oracle WebLogic Server. Don't use clustering with Oracle WebLogic Server Standard Edition. **UDP multicast isn't supported**. Azure supports UDP unicasting, but not multicasting or broadcasting. Oracle WebLogic Server can rely on Azure UDP unicast capabilities. For best results relying on

UDP unicast, we recommend that the WebLogic cluster size is kept static, or kept with no more than 10 managed servers. **Oracle WebLogic Server expects public and private ports to be the same for T3 access**. For example, when using Enterprise JavaBeans (EJB). Consider a multi-tier scenario where a service layer application is running on an Oracle WebLogic Server cluster consisting of two or more VMs, in a virtual network named SLWLS. The client tier is in a different subnet in the same virtual network, running a simple Java program trying to call EJB in the service layer. Because you must load balance the service layer, a public load-balanced endpoint needs to be created for the VMs in the Oracle WebLogic Server cluster. If the private port specified is different from the public port an error occurs. For example, if you use `7006:7008`, the following error occurs because for any remote T3 access, Oracle WebLogic Server expects the load balancer port and the WebLogic managed server port to be the same.

```
[java] javax.naming.CommunicationException [Root exception is
java.net.ConnectException: t3://example.cloudapp.net:7006:

Bootstrap to: example.cloudapp.net/138.91.142.178:7006' over: 't3' got an error or
timed out]
```

In the preceding case, the client is accessing port 7006, which is the load balancer port, and the managed server is listening on 7008, which is the private port. This restriction is applicable only for T3 access, not HTTP.

To avoid this issue, use one of the following workarounds:

- Use the same private and public port numbers for load balanced endpoints dedicated to T3 access.
- Include the following JVM parameter when starting Oracle WebLogic Server: configCopy `Dweblogic.rjvm.enableprotocolswitch=true`
- Dynamic clustering and load balancing limitations. Suppose you want to use a dynamic cluster in Oracle WebLogic Server and expose it through a single, public load-balanced endpoint in Azure. This approach can be done as long as you use a fixed port number for each of the managed servers, not dynamically assigned from a range, and don't start more managed servers than there are machines the administrator is tracking. There should be no more than one managed server per VM. If your configuration results in more Oracle WebLogic Servers being started than there are VMs, it isn't possible for more than one of those instances of Oracle WebLogic Servers to bind to a given port number. That is, if multiple Oracle WebLogic Server instances share the same virtual machine, the others on that VM fail. If you configure the admin server to automatically assign unique port numbers to its managed servers, then load balancing isn't possible because Azure doesn't

support mapping from a single public port to multiple private ports, as would be required for this configuration.

- Multiple instances of Oracle WebLogic Server on a VM. Depending on your deployment requirements, you might consider running multiple instances of Oracle WebLogic Server on the same VM, if the VM is large enough. For example, on a midsize VM, which contains two cores, you could choose to run two instances of Oracle WebLogic Server. However, we still recommend that you avoid introducing single points of failure into your architecture. Running multiple instances of Oracle WebLogic Server on just one VM would be such a single point.

Using at least two VMs could be a better approach. Each VM can run multiple instances of Oracle WebLogic Server. Each instance of Oracle WebLogic Server could still be part of the same cluster. However, it's currently not possible to use Azure to load-balance endpoints that are exposed by such Oracle WebLogic Server deployments within the same VM. Azure Load Balancer requires the load-balanced servers to be distributed among unique VMs.

High availability and disaster recovery options

When using Oracle solutions in Azure, you're responsible for implementing a high availability and disaster recovery solution to avoid any downtime. You can also implement high availability and disaster recovery for Oracle Database Enterprise Edition by using Data Guard, Active Data Guard, or Oracle GoldenGate. The approach requires two databases on two separate VMs, which should be in the same virtual network to ensure they can access each other over the private persistent IP address.

We recommend placing the VMs in the same availability set to allow Azure to place them into separate fault domains and upgrade domains. If you want to have geo-redundancy, set up the two databases to replicate between two different regions and connect the two instances with a VPN Gateway. To walk through the basic setup procedure on Azure, see Implement Oracle Data Guard on an Azure Linux virtual machine.

With Oracle Active Data Guard, you can achieve high availability with a primary database in one VM, a secondary (standby) database in another VM, and one-way replication set up between them. The result is read access to the copy of the database. With Oracle GoldenGate, you can configure bi-directional replication between the two databases. To learn how to set up a high-availability solution for your databases using these tools, see Active Data Guard and GoldenGate. If you need read-write access to the copy of the database, you can use Oracle Active Data Guard.

To walk through the basic setup procedure on Azure, see Implement Oracle Golden Gate on an Azure Linux VM.

You can effectively achieve high availability for your Oracle databases by using Azure NetApp Files availability zone volume placement in combination with Oracle Data Guard for an HA architecture across zones. Alternatively, to eliminate the cost of Data Guard licenses and running VMs in the secondary zone, you can use the storage-based replication functionality of Azure NetApp Files. Azure NetApp Files volumes can be placed in the availability zone of your choice the same way and can then be replicated between zones within the region by using cross-zone replication (or to another region using cross-region replication).

In addition to having a high availability and disaster recovery solution architected in Azure, you should have a backup strategy in place to restore your database.

Backup Oracle workloads

Different backup strategies are available for Oracle on Azure VMs, the following backups are other options:

- Using Azure files
- Using Azure NetApp Files
- Using Azure backup
- Using Oracle RMAN Streaming data backup

Deploy Oracle applications on Azure

Use Terraform templates, AZ CLI, or the Azure Portal to set up Azure infrastructure and install Oracle applications. You also use Ansible to configure DB inside the VM. For more information, see Terraform on Azure.

Oracle has certified the following applications to run in Azure when connecting to an Oracle database by using the Azure with Oracle Cloud interconnect solution:

- E-Business Suite
- JD Edwards EnterpriseOne
- PeopleSoft
- Oracle Retail applications
- Oracle Hyperion Financial Management

You can deploy custom applications in Azure that connect with OCI and other Azure services.

Support for JD Edwards

According to Oracle Support, JD Edwards EnterpriseOne versions 9.2 and above are supported on any public cloud offering that meets their specific Minimum Technical Requirements (MTR). You need to create custom images that meet their MTR specifications for operating system and software application compatibility. For more information, see Doc ID 2178595.1.

Licensing

Deployment of Oracle solutions in Azure is based on a bring-your-own-license model. This model assumes that you have licenses to use Oracle software and that you have a current support agreement in place with Oracle. Microsoft Azure is an authorized cloud environment for running Oracle Database. The Oracle Core Factor table isn't applicable when licensing Oracle databases in the cloud. For more information, see Oracle Processor Core Factor Table. Instead, when using VMs with Hyper-Threading Technology enabled for Enterprise Edition databases, count two vCPUs as equivalent to one Oracle Processor license if hyperthreading is enabled, as stated in the policy document. The policy details can be found at Licensing Oracle Software in the Cloud Computing Environment.

Oracle databases generally require higher memory and I/O. For this reason, we recommend Memory Optimized VMs for these workloads. To optimize your workloads further, we recommend Constrained Core vCPUs for Oracle Database workloads that require high memory, storage, and I/O bandwidth, but not a high core count. When you migrate Oracle software and workloads from on-premises to Microsoft Azure, Oracle provides license mobility as stated in Oracle and Microsoft Strategic Partnership FAQ.

Next steps

You now have an overview of current Oracle databases and solutions based on VM images in Microsoft Azure. Your next step is to deploy your first Oracle database on Azure.

- Create an Oracle database on Azure

Feedback

Was this page helpful? Yes No

Create an Oracle Database in an Azure VM

Article • 08/22/2024

Applies to: ✔ Linux VMs

This article describes how to use the Azure CLI to deploy an Azure virtual machine (VM) from the Oracle marketplace gallery image to create an Oracle Database 19c database. After you deploy the server, you connect the server via SSH to configure the Oracle database.

Prerequisites

- If you don't have an Azure subscription, create an Azure free account before you begin.
- Azure Cloud Shell or the Azure CLI.

  You can run the Azure CLI commands in this quickstart interactively in Azure Cloud Shell. To run the commands in Cloud Shell, select **Open Cloudshell** at the upper-right corner of a code block. Select **Copy** to copy the code, and paste it into Cloud Shell to run it. You can also run Cloud Shell from within the Azure portal. Cloud Shell always uses the latest version of the Azure CLI.

  Alternatively, you can install Azure CLI locally to run the commands. The steps in this article require the Azure CLI version 2.0.4 or later. Run az version to see your installed version and dependent libraries, and run az upgrade to upgrade. If you use a local installation, sign in to Azure by using the az login command.

Create resource group

Create a resource group with the az group create command. An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named **rg-oracle** in the **eastus** location.

Azure CLI

```
az group create --name rg-oracle --location eastus
```

> **Note**
>
> This quickstart creates a Standard_DS2_v2 SKU VM in the East US region. To view the list of supported SKUs by region, use the az vm list-skus command.

Create virtual machine

Create a virtual machine (VM) with the az vm create command.

The following example creates a VM named **vmoracle19c**. It also creates SSH keys, if they don't already exist in a default key location. To use a specific set of keys, you can use the `--ssh-key-value` option with the command.

Azure CLI

```
az vm create \
    --name vmoracle19c \
    --resource-group rg-oracle \
    --image Oracle:oracle-database-19-3:oracle-database-19-0904:latest \
    --size Standard_DS2_v2 \
    --admin-username azureuser \
    --generate-ssh-keys \
    --public-ip-address-allocation static \
    --public-ip-address-dns-name vmoracle19c
```

After you create the VM, Azure CLI displays information similar to the following example. Note the value for the `publicIpAddress` property. You use this IP address to access the VM.

Output

```
{
  "fqdns": "",
  "id": "/subscriptions/{snip}/resourceGroups/rg-
oracle/providers/Microsoft.Compute/virtualMachines/vmoracle19c",
  "location": "eastus",
  "macAddress": "00-0D-3A-36-2F-56",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "13.64.104.241",
  "resourceGroup": "rg-oracle"
}
```

Create disk for Oracle data files

Create and attach a new disk for Oracle data files and a fast recovery area (FRA) with the az vm disk attach command.

The following example creates a disk named **oradata01**.

Azure CLI

```
az vm disk attach \
    --name oradata01 --new \
    --resource-group rg-oracle \
    --size-gb 64 --sku StandardSSD_LRS \
    --vm-name vmoracle19c
```

Open ports for connectivity

In this task, you must configure some external endpoints for the database listener to use by setting up the Azure network security group (NSG) that protects the VM.

1. Create the NSG for the VM with the az network nsg create command. This command creates the **vmoracle19cNSG** NSG for rules to control access to the VM:

   Azure CLI

   ```
   az network nsg create --resource-group rg-oracle --name vmoracle19cNSG
   ```

2. Create an NSG rule with the az network nsg rule create command. This command creates the **allow-oracle** NSG rule to open the endpoint for remote access to the Oracle database:

   Azure CLI

   ```
   az network nsg rule create \
       --resource-group rg-oracle \
       --nsg-name vmoracle19cNSG \
       --name allow-oracle \
       --protocol tcp \
       --priority 1001 \
       --destination-port-range 1521
   ```

3. Create a second NSG rule to open the endpoint for remote access to Oracle. This command creates the **allow-oracle-EM** NSG rule:

   Azure CLI

```
az network nsg rule create \
    --resource-group rg-oracle \
    --nsg-name vmoracle19cNSG \
    --name allow-oracle-EM \
    --protocol tcp \
    --priority 1002 \
    --destination-port-range 5502
```

4. As needed, use the az network public-ip show command to get the public IP address of your VM:

Azure CLI

```
az network public-ip show \
    --resource-group rg-oracle \
    --name vmoracle19cPublicIP \
    --query "ipAddress" \
    --output tsv
```

Prepare VM environment

1. Create an SSH session with the VM. Replace the `<publicIPAddress>` portion with the public IP address value for your VM, such as `10.200.300.4`:

Bash

```
ssh azureuser@<publicIPAddress>
```

2. Switch to the root user:

Bash

```
sudo su -
```

3. Locate the most recently created disk device that you want to format to hold Oracle data files:

Bash

```
ls -alt /dev/sd*|head -1
```

The output is similar to this example:

Output

```
brw-rw----. 1 root disk 8, 16 Dec  8 22:57 /dev/sdc
```

4. As the root user, use the `parted` command to format the device.

   a. First, create a disk label:

   Bash

   ```bash
   parted /dev/sdc mklabel gpt
   ```

   b. Next, create a primary partition that spans the entire disk:

   Bash

   ```bash
   parted -a optimal /dev/sdc mkpart primary 0GB 64GB
   ```

   c. Finally, check the device details by printing its metadata:

   Bash

   ```bash
   parted /dev/sdc print
   ```

   The output is similar to this example:

   Bash

   ```
   Model: Msft Virtual Disk (scsi)
   Disk /dev/sdc: 68.7GB
   Sector size (logical/physical): 512B/4096B
   Partition Table: gpt
   Disk Flags:
   Number   Start    End      Size     File system   Name     Flags
   1        1049kB   64.0GB   64.0GB   ext4          primary
   ```

5. Create a filesystem on the device partition:

   Bash

   ```bash
   mkfs -t ext4 /dev/sdc1
   ```

   The output is similar to this example:

   Bash

```
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
3907584 inodes, 15624704 blocks
781235 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2164260864
477 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
     32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
2654208,
     4096000, 7962624, 11239424
 Allocating group tables: done
 Writing inode tables: done
 Creating journal (32768 blocks): done
 Writing superblocks and filesystem accounting information: done
```

6. Create a mount point:

```bash
mkdir /u02
```

7. Mount the disk:

```bash
mount /dev/sdc1 /u02
```

8. Change permissions on the mount point:

```bash
chmod 777 /u02
```

9. Add the mount to the **/etc/fstab** file:

```bash
echo "/dev/sdc1                /u02                    ext4    defaults
0 0" >> /etc/fstab
```

> **Important**
>
> This command mounts the /etc/fstab file without a specific UUID, which can prevent successful reboot of the disk. Before you attempt to reboot the disk, update the /etc/fstab entry to include a UUID for the mount point.

10. Update the **/etc/hosts** file with the public IP address and address hostname. Change the `<Public IP>` and two `<VMname>` portions to reflect your actual values:

    ```bash
    echo "<Public IP> <VMname>.eastus.cloudapp.azure.com <VMname>" >>
    /etc/hosts
    ```

11. Add the domain name of the VM to the **/etc/hostname** file. The following command assumes the resource group and VM are created in the **eastus** region:

    ```bash
    sed -i 's/$/\.eastus\.cloudapp\.azure\.com &/' /etc/hostname
    ```

12. Open firewall ports.

    Because SELinux is enabled by default on the Marketplace image, we need to open the firewall to traffic for the database listening port 1521, and Enterprise Manager Express port 5502. Run the following commands as root user:

    ```bash
    firewall-cmd --zone=public --add-port=1521/tcp --permanent
    firewall-cmd --zone=public --add-port=5502/tcp --permanent
    firewall-cmd --reload
    ```

Create the database

The Oracle software is already installed on the Marketplace image. Create a sample database as follows.

1. Switch to the **oracle** user:

    ```bash
    ```

```
sudo su - oracle
```

2. Start the database listener:

```bash
lsnrctl start
```

The output is similar to the following example:

```output
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 20-OCT-2020
01:58:18

Copyright (c) 1991, 2019, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/19.0.0/dbhome_1/bin/tnslsnr: please
wait...

TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Log messages written to
/u01/app/oracle/diag/tnslsnr/vmoracle19c/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=vmoracle19c.eastus.cloudapp.azure.com)(PORT=1521)))

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 -
Production
Start Date                20-OCT-2020 01:58:18
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Log File
/u01/app/oracle/diag/tnslsnr/vmoracle19c/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=vmoracle19c.eastus.cloudapp.azure.com)(PORT=1521)))
The listener supports no services
The command completed successfully
```

3. Create a data directory for the Oracle data files:

```bash
```

```
mkdir /u02/oradata
```

4. Run the Database Creation Assistant:

Bash

```
dbca -silent \
    -createDatabase \
    -templateName General_Purpose.dbc \
    -gdbname oratest1 \
    -sid oratest1 \
    -responseFile NO_VALUE \
    -characterSet AL32UTF8 \
    -sysPassword OraPasswd1 \
    -systemPassword OraPasswd1 \
    -createAsContainerDatabase false \
    -databaseType MULTIPURPOSE \
    -automaticMemoryManagement false \
    -storageType FS \
    -datafileDestination "/u02/oradata/" \
    -ignorePreReqs
```

It takes a few minutes to create the database.

The output is similar to the following example:

Output

```
     Prepare for db operation
    10% complete
    Copying database files
    40% complete
    Creating and starting Oracle instance
    42% complete
    46% complete
    50% complete
    54% complete
    60% complete
    Completing Database Creation
    66% complete
    69% complete
    70% complete
    Executing Post Configuration Actions
    100% complete
    Database creation complete. For details check the logfiles at:
/u01/app/oracle/cfgtoollogs/dbca/oratest1.
    Database Information:
    Global Database Name:oratest1
    System Identifier(SID):oratest1
    Look at the log file
```

```
"/u01/app/oracle/cfgtoollogs/dbca/oratest1/oratest1.log" for further
details.
```

5. Set Oracle variables:

   Before you connect, you need to set the environment variable `ORACLE_SID`:

   ```bash
   export ORACLE_SID=oratest1
   ```

   You should also add the `ORACLE_SID` variable to the `oracle` users **.bashrc** file for future sign-ins by using the following command:

   ```bash
   echo "export ORACLE_SID=oratest1" >> ~oracle/.bashrc
   ```

Automate database startup and shutdown

The Oracle database by default doesn't automatically start when you restart the VM. To set up the Oracle database to start automatically, first sign in as root. Then, create and update some system files.

1. Sign on as the root user:

   ```bash
   sudo su -
   ```

2. Change the automated startup flag from `N` to `Y` in the /etc/oratab file:

   ```bash
   sed -i 's/:N/:Y/' /etc/oratab
   ```

3. Create a file named **/etc/init.d/dbora** and add the following bash command to the file:

   ```bash
   #!/bin/sh
   # chkconfig: 345 99 10
   ```

```
# Description: Oracle auto start-stop script.
#
# Set ORA_HOME to be equivalent to $ORACLE_HOME.
ORA_HOME=/u01/app/oracle/product/19.0.0/dbhome_1
ORA_OWNER=oracle

case "$1" in
'start')
    # Start the Oracle databases:
    # The following command assumes that the Oracle sign-in
    # will not prompt the user for any values.
    # Remove "&" if you don't want startup as a background process.
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart $ORA_HOME" &
    touch /var/lock/subsys/dbora
    ;;

'stop')
    # Stop the Oracle databases:
    # The following command assumes that the Oracle sign-in
    # will not prompt the user for any values.
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut $ORA_HOME" &
    rm -f /var/lock/subsys/dbora
    ;;
esac
```

4. Change permissions on files with the `chmod` command:

```bash
chgrp dba /etc/init.d/dbora
chmod 750 /etc/init.d/dbora
```

5. Create symbolic links for startup and shutdown:

```bash
ln -s /etc/init.d/dbora /etc/rc.d/rc0.d/K01dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc3.d/S99dbora
ln -s /etc/init.d/dbora /etc/rc.d/rc5.d/S99dbora
```

6. To test your changes, restart the VM:

```bash
reboot
```

Clean up resources

After you finish exploring your first Oracle database on Azure and the VM is no longer needed, you can use the az group delete command to remove the resource group, VM, and all related resources.

Azure CLI

```
az group delete --name rg-oracle
```

Next steps

- Protect your database in Azure with Oracle backup strategies
- Explore Oracle solutions on Azure
- Install and configure Oracle Automated Storage Management

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What are solutions for running Oracle WebLogic Server on Azure Virtual Machines?

Article • 08/22/2024

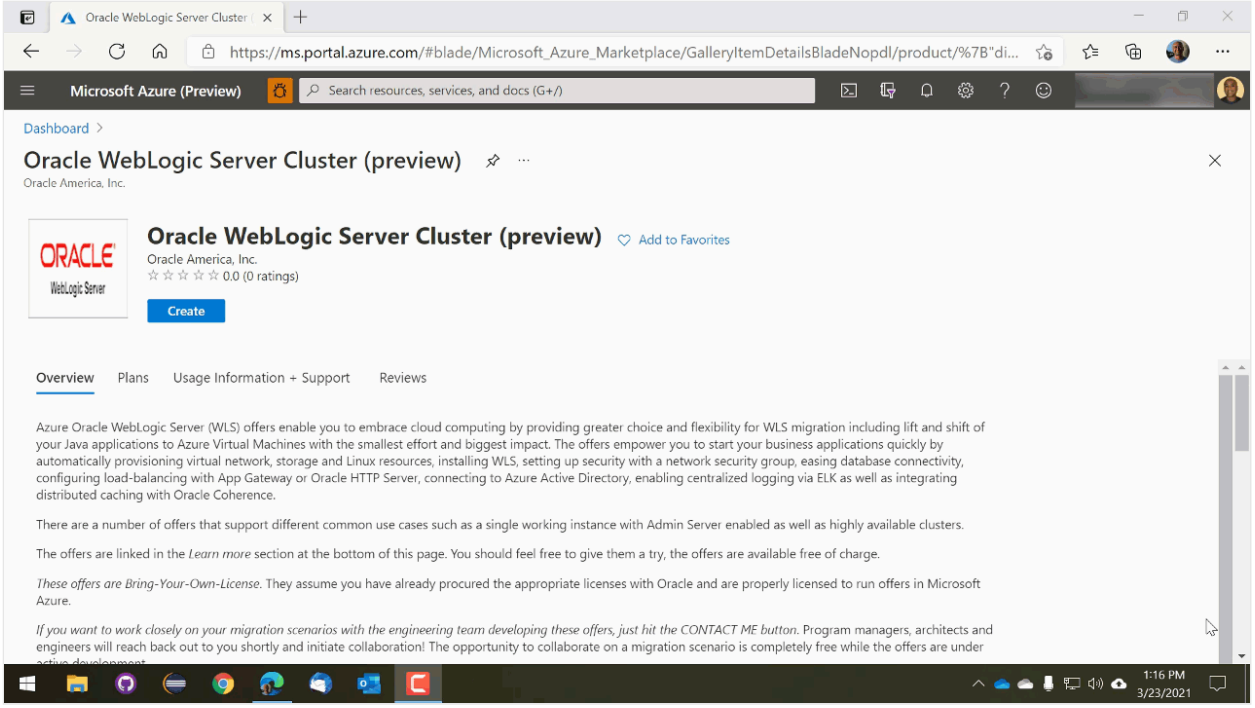
Applies to: ✔ Linux VMs

This page describes the solutions for running Oracle WebLogic Server (WLS) on Azure Virtual Machines. These solutions are jointly developed and supported by Oracle and Microsoft.

You can also run WebLogic Server on the Azure Kubernetes Service. The solutions to do so are described in the article Running Oracle WebLogic Server on the Azure Kubernetes Service.

WebLogic Server is a leading Java application server running some of the most mission-critical enterprise Java applications across the globe. WebLogic Server forms the middleware foundation for the Oracle software suite. Oracle and Microsoft are committed to empowering WebLogic customers with choice and flexibility to run workloads on Azure as a leading cloud platform.

There are several offers that target use cases such as single instance with Administration server enabled and WebLogic cluster including both static cluster and dynamic clusters. These offers are solution templates that aim to expedite your Java application deployment to Azure Virtual Machines. They automatically provision virtual network, storage, Java, WebLogic, and Linux resources required for most common cloud provisioning scenarios. After initial provisioning is done, you're completely free to customize deployments further.



The offers are Bring-Your-Own-License. They assume you have already got the appropriate licenses with Oracle and are properly licensed to run offers in Azure.

The solution templates support a range of operating system, Java, and WebLogic Server versions through base images (such as WebLogic Server 14 and Java 11 on Red Hat Enterprise Linux 8). These base images are also available on Azure Marketplace on their own. The base images are suitable for customers that require complex, customized Azure deployments.

If you're interested in providing feedback or working closely on your migration scenarios with the engineering team developing WeLogic on Azure solutions, fill out this short survey on WebLogic migration and include your contact information. The team of program managers, architects, and engineers will promptly get in touch with you to initiate close collaboration.

WebLogic Server single instance with admin console on Azure VM

The solution template WebLogic Server single instance with admin console on Azure VM provisions a single virtual machine and installs WebLogic Server on it. It creates a domain and starts up an administration server. After the solution template performs most boilerplate resource provisioning and configuration, you can manage the domain and get started with Java application deployments right away.

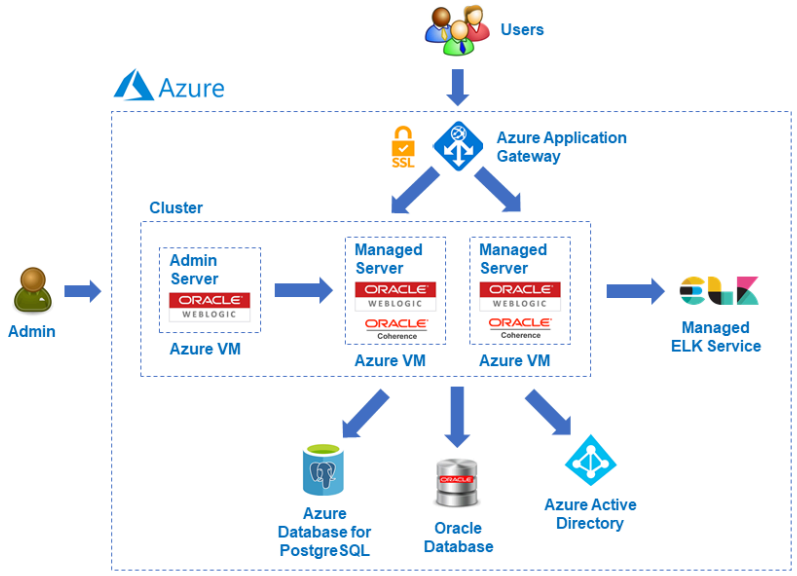
For the getting started guidance, see Quickstart: Deploy WebLogic Server on Azure Virtual Machine using the Azure portal. For deployment guidance, see Using Oracle

WebLogic Server on Microsoft Azure IaaS .

WebLogic Server configured cluster on Azure VMs

The solution template WebLogic Server configured cluster on Azure VMs creates a highly available configured cluster of WebLogic Server on Azure Virtual Machines. The administration server and all managed servers are started by default. After the solution template performs most boilerplate resource provisioning and configuration, you can manage the cluster and get started with highly available applications right away. For deployment guidance, see Using Oracle WebLogic Server on Microsoft Azure IaaS .

The solution enables a wide range of production-ready deployment architectures with relative ease. You can meet most migration cases in the most productive way possible by allowing a focus on business application development.



After resources are automatically provisioned by the solutions, you have complete flexibility to customize your deployments further. It's likely that, on top of deploying applications, you'll integrate further Azure resources with your deployments. You're encouraged to connect with the development team and provide feedback on further improving WebLogic on Azure solutions.

If you prefer step-by-step guidance for going from zero to a WebLogic Server cluster without any solution templates or base images, see Install Oracle WebLogic Server on Azure Virtual Machines manually.

WebLogic Server dynamic cluster on Azure VMs

The solution template WebLogic Server dynamic cluster on Azure VMs creates a dynamic cluster of WebLogic Server on Azure Virtual Machines. Administration Server is on one of the Azure Virtual Machines. For deployment guidance, see Using Oracle WebLogic Server on Microsoft Azure IaaS.

WebLogic Server on Azure VM Base Images

The solution templates introduced above support a range of operating system, Java, and WebLogic Server versions through base images - such as WebLogic Server 14 and Java 11 on Red Hat Enterprise Linux 8. These base images are also available on Azure Marketplace on their own. The base images are suitable for customers that require complex, customized Azure deployments. You can use the keywords *oracle weblogic base image* to search for the current set of base images of WebLogic Server on Azure VM available in Azure Marketplace.

Next steps

The following articles provide more information on getting started with these technologies.

- Quickstart: Deploy WebLogic Server on Azure Virtual Machine using the Azure portal
- Manually install Oracle WebLogic Server on Azure Virtual Machines
- What are solutions for running Oracle WebLogic Server on the Azure Kubernetes Service?

For more information about the Oracle WebLogic offers at Azure Marketplace, see Oracle WebLogic Server on Azure. These offers are all *Bring-Your-Own-License*. They assume that you already have the appropriate licenses with Oracle and are properly licensed to run offers in Azure.

You're encouraged to connect with the development team and provide feedback on further improving WebLogic on Azure solutions.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Quickstart: Deploy WebLogic Server on Azure Virtual Machines (VMs)

Article • 09/06/2024

This article shows you how to quickly deploy WebLogic Server (WLS) on an Azure Virtual Machine (VM) with the simplest possible set of configuration choices using the Azure portal. In this quickstart, you learn how to:

- Deploy WebLogic Server with Administration Server enabled on a VM using the Azure portal.
- Deploy a sample Java application with the WebLogic Server Administration Console.
- Connect to the VM running WebLogic using SSH.

This article uses the Azure Marketplace Offer for WebLogic Server to accelerate your journey to Azure. The offer automatically provisions several Azure resources, including the following resources:

- Azure Virtual Machine pre-installed with your choice of OS version, Java JDK version, and WebLogic Server Version.
- TLS/SSL configuration, including the easy generation of a self-signed certificate.
- Virtual network (VNET), including the ability to provision into an existing VNET.
- Database connection to a pre-existing database.

If you prefer a less automated step-by-step approach, see Manually install WebLogic Server on Azure Virtual Machines (VMs).

If you're interested in providing feedback or working closely on your migration scenarios with the engineering team developing WebLogic on Azure solutions, fill out this short survey on WebLogic migration and include your contact information. The team of program managers, architects, and engineers will promptly get in touch with you to initiate close collaboration.

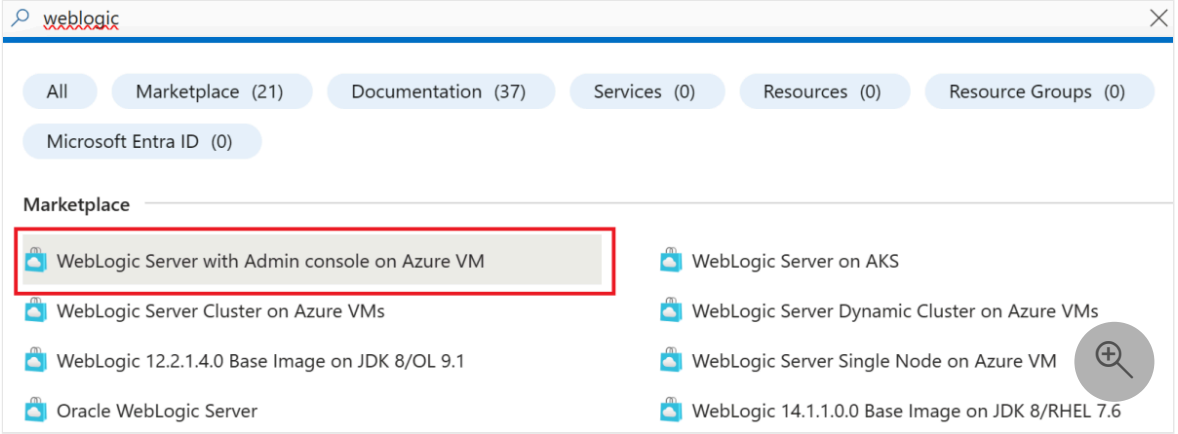
Prerequisites

- If you don't have an Azure subscription, create an Azure free account before you begin.

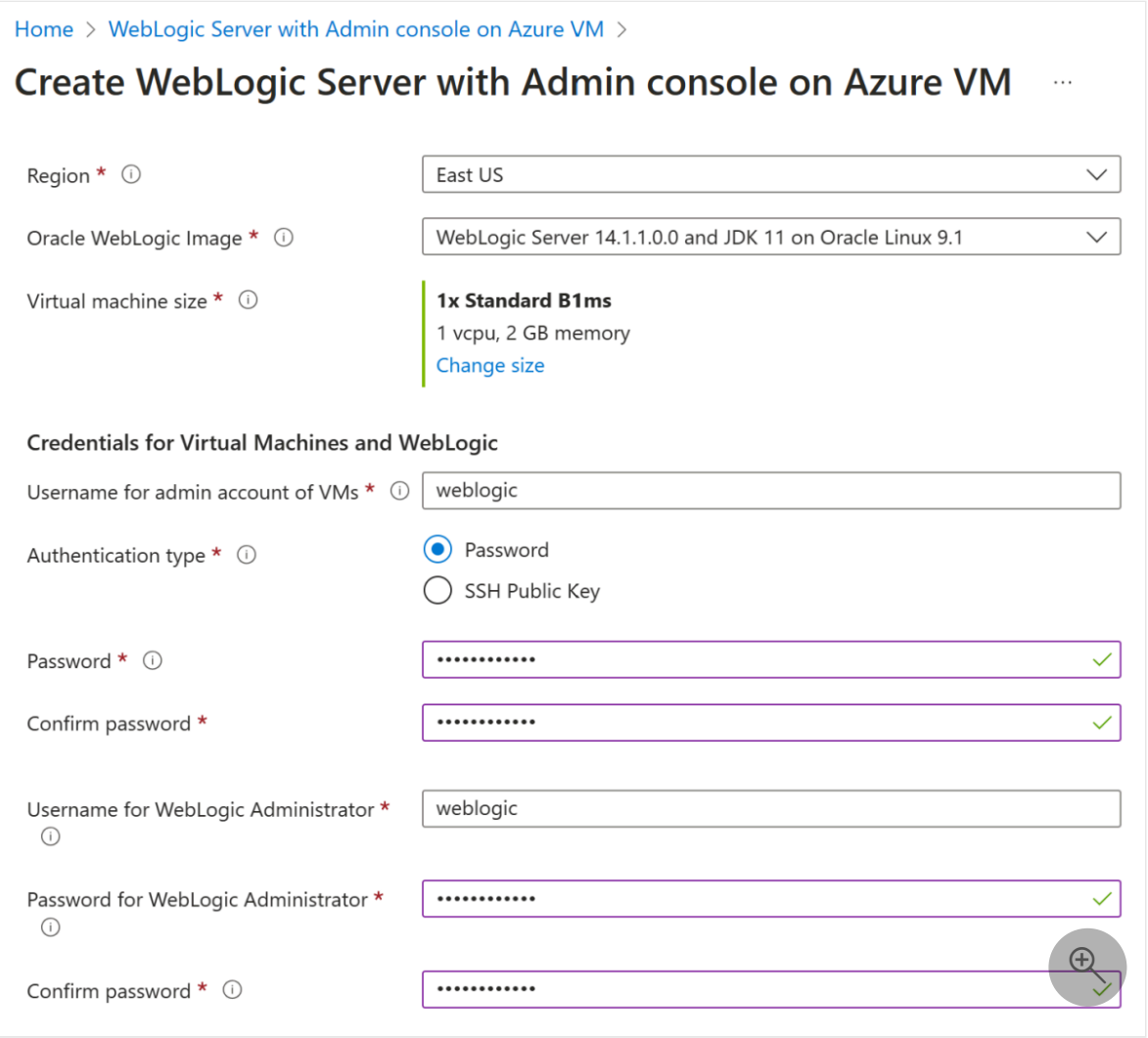
Deploy WebLogic Server with Administration Server on a VM

The following steps show you how to deploy WebLogic Server on a VM using the single instance with an admin server offer on the Azure portal. There are other offers that meet different scenarios such as WebLogic cluster on multiple VMs.

1. In the search bar at the top of the portal, enter *weblogic*. In the autosuggested search results, in the **Marketplace** section, select **WebLogic Server with Admin Console on VM**. You can also go directly to the offer using the portal link.



2. On the offer page, select **Create**. You then see the **Basics** pane.



3. On the **Basics** pane, ensure the value shown in the **Subscription** field is the same one that you used to sign in to the Azure portal.

4. The offer must be deployed in an empty resource group. In the **Resource group** field, select **Create new** and fill in a value for the resource group. Because resource groups must be unique within a subscription, pick a unique name. An easy way to have unique names is to use a combination of your initials, today's date, and some identifier. For example, *ejb0802wls*.

5. Under **Instance details**, select the region for the deployment.

6. Accept the default value in **Oracle WebLogic Image**.

7. Accept the default value in **Virtual machine size**.

   If the default size isn't available in your region, choose an available size by selecting **Change size**, then select one of the listed sizes.

8. Under **Credentials for Virtual Machines and WebLogic**, leave the default value for **Username for admin account of VMs**.

9. Next to **Authentication type**, select **Password**. This article uses a user name and password pair for the authentication. If you want to use SSH, see Create and use an SSH public-private key pair for Linux VMs in Azure. Fill in *wlsVmCluster2022* for **Password**. Use the same value for the confirmation.

10. Leave the default value for **Username for WebLogic Administrator**.

11. Fill in *wlsVmCluster2022* for the **Password for WebLogic Administrator**. Use the same value for the confirmation.

12. Select **Review + create**.

13. Ensure the green **Validation Passed** message appears at the top. If it doesn't, fix any validation problems and select **Review + create** again.

14. Select **Create**.

15. Track the progress of the deployment in the **Deployment is in progress** page.

Depending on network conditions and other activity in your selected region, the deployment might take up to 30 minutes to complete.

Examine the deployment output

The steps in this section show you how to verify the deployment successfully completed.

If you navigated away from the **Deployment is in progress** page, the following steps show you how to get back to that page. If you're still on the page that shows **Your deployment is complete**, you can skip to step 5, after the screenshot.

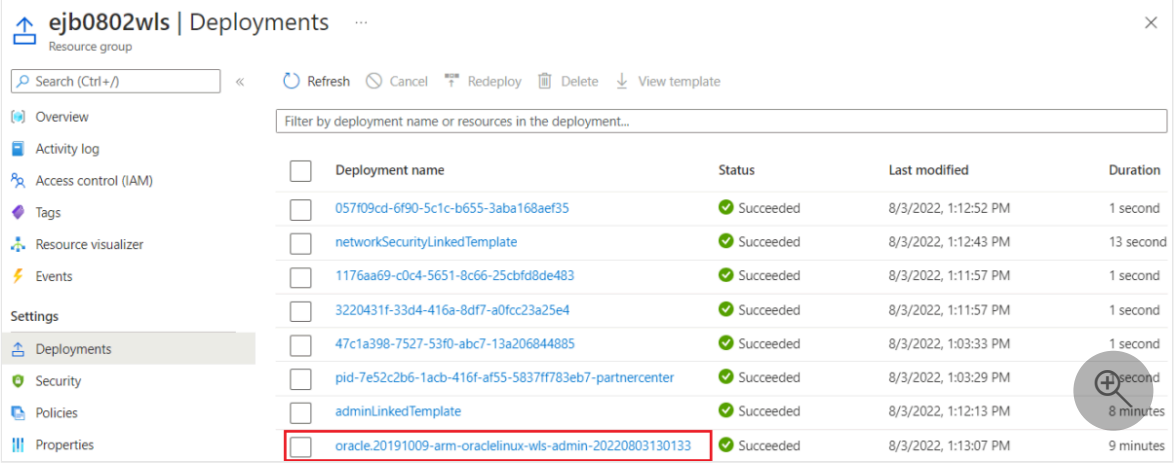
1. In the corner of any portal page, select the hamburger menu and select **Resource groups**.

2. In the box with the text **Filter for any field**, enter the first few characters of the resource group you created previously. If you followed the recommended convention, enter your initials, then select the appropriate resource group.

3. In the left navigation pane, in the **Settings** section, select **Deployments**. You can see an ordered list of the deployments to this resource group, with the most recent one first.

4. Scroll to the oldest entry in this list. This entry corresponds to the deployment you started in the preceding section. Select the oldest deployment, as shown in the

following screenshot: x



5. In the left panel, select **Outputs**. This list shows useful output values from the deployment.

6. The **sshCommand** value is the fully qualified SSH command to connect to the VM that runs WebLogic Server. Select the copy icon next to the field value to copy the link to your clipboard. Save this value aside for later.

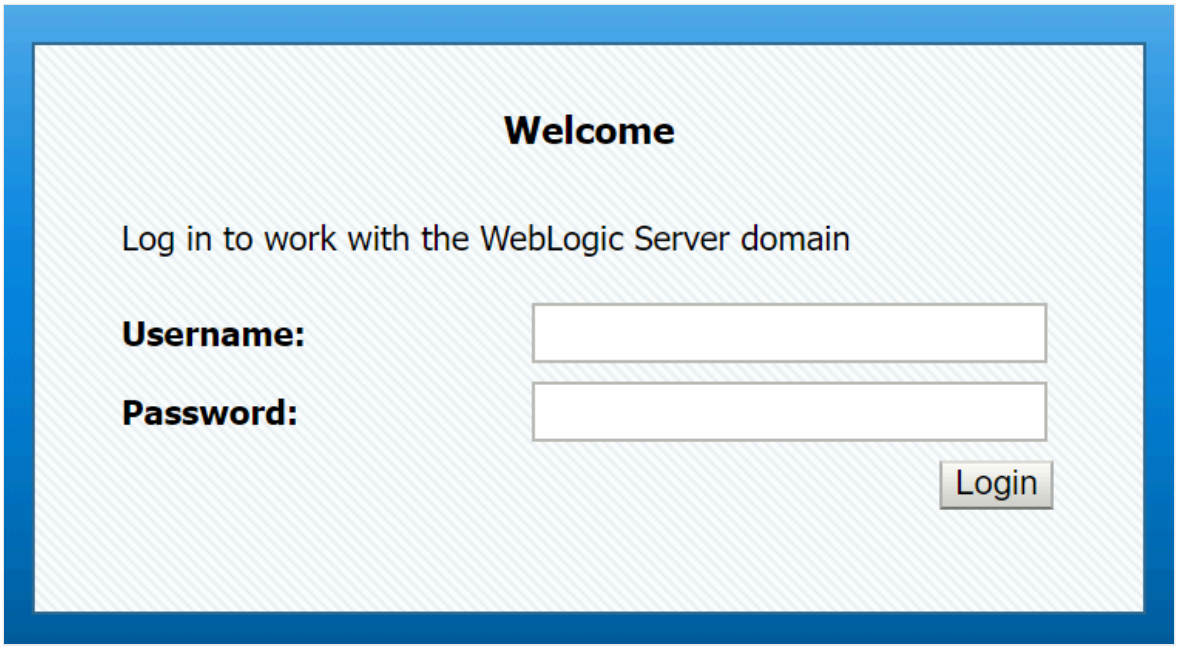
7. The **adminConsoleURL** value is the fully qualified public internet visible link to the WebLogic Server admin console. Select the copy icon next to the field value to copy the link to your clipboard. Save this value aside for later.

Deploy a Java application from Administration Console

Use the following steps to run a sample application on the WebLogic Server:

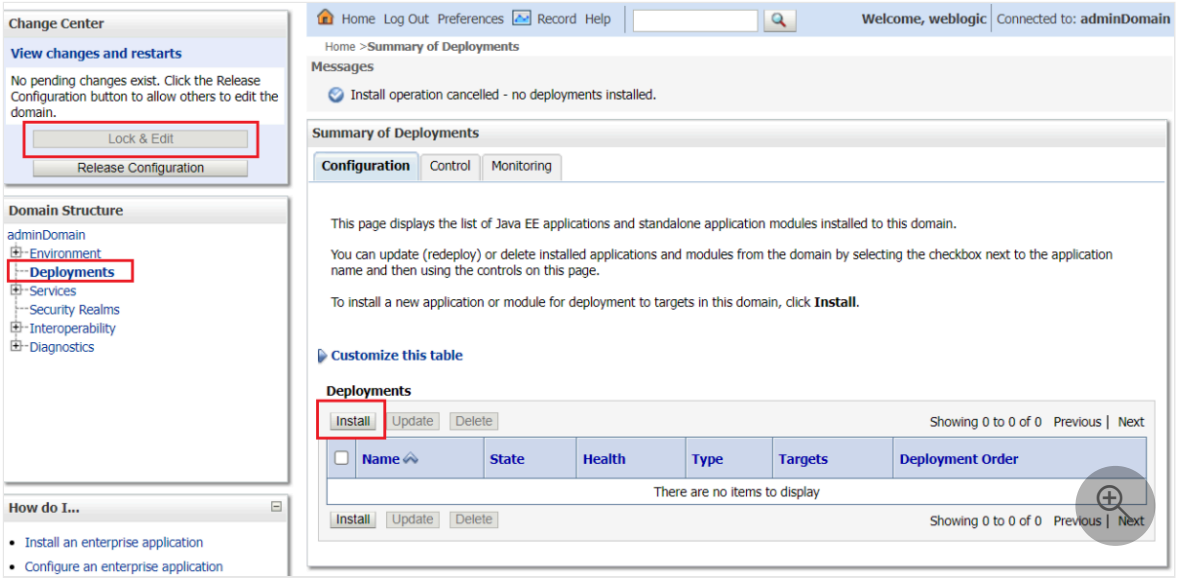
1. Download a sample application as a *.war* or *.ear* file. The sample app should be self contained and not have any database, messaging, or other external connection requirements. The sample app from the WebLogic Kubernetes Operator documentation is a good choice. You can download it from Oracle. Save the file to your local filesystem.

2. Paste the value of **adminConsoleURL** in an internet-connected web browser. You should see the familiar WebLogic Server admin console login screen as shown in the following screenshot:



3. Log in with user name *weblogic* and your password (this article uses *wlsVmCluster2022*). You can see the WebLogic Server Administration Console overview page.

4. Under **Change Center** on the top left corner, select **Lock & Edit**, as shown in the following screenshot:



5. Under **Domain Structure** on the left side, select **Deployments**.

6. Under **Configuration**, select **Install**. There's an **Install Application Assistant** to guide you to finish the installation.
   a. Under **Locate deployment to install and prepare for deployment**, select **Upload your file(s)**.
   b. Under **Upload a deployment to the Administration Server**, select **Choose File** and upload your sample application. Select **Next**.
   c. Accept the defaults in the next few screens and select **Finish**.

d. On the application configuration screen, select **Save**.

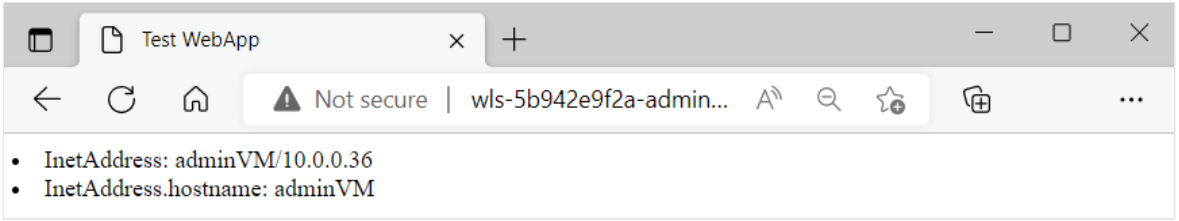
7. Under **Change Center** on the top left corner, select **Activate Changes**. You can see the message **All changes have been activated. No restarts are necessary**.

8. Under **Summary of Deployments**, select **Control**. Select the checkbox near the application name to select the application. Select **Start** and then select **Servicing all requests**.

9. Under **Start Application Assistant**, select **Yes**. If no error happens, you can see the message **Start requests have been sent to the selected deployments.**

10. Construct a fully qualified URL for the sample app, such as `http://<vm-host-name>:<port>/<your-app-path>`. You can get the host name and port from **adminConsoleURL** by removing `/console/`. If you're using the recommended sample app, the URL should be `http://<vm-host-name>:<port>/testwebapp/`, which should be similar to `http://wls-5b942e9f2a-admindomain.westus.cloudapp.azure.com:7001/testwebapp/`.

11. Paste the fully qualified URL in an internet-connected web browser. If you deployed the recommended sample app, you should see a page that looks similar to the following screenshot:



Connect to the virtual machine

If you want to manage the VM, you can connect to it with SSH command. Before accessing the machine, make sure you enabled port 22 for the SSH agent.

Use the following steps to enable port 22:

1. Navigate back to your working resource group in the Azure portal. In the overview page, you can find a network security group named **wls-nsg**. Select **wls-nsg**.

2. In the left panel, select **Settings**, then **Inbound security rules**. If there's a rule to allow port `22`, then you can jump to step 4.

3. In the top of the page, select **Add**.
   a. Under **Destination port ranges**, fill in the value *22*.

b. Fill in the rule name *Port_SSH* for **Name**.
c. Leave the default value for the other fields.
d. Select **Add**.

After the deployment completes, you can SSH to the VM.

4. Connect to the VM with the value of **sshCommand** and your password (this article uses *wlsVmCluster2022*).

Clean up resources

If you're not going to continue to use the WebLogic Server, navigate back to your working resource group in the Azure portal. At the top of the page, under the text **Resource group**, select **Delete resource group**.

Next steps

Continue to explore options to run WebLogic Server on Azure.

- WebLogic Server on virtual machines
- WebLogic Server on Azure Kubernetes Service

For more information about the Oracle WebLogic offers at Azure Marketplace, see Oracle WebLogic Server on Azure. These offers are all *Bring-Your-Own-License*. They assume that you already have the appropriate licenses with Oracle and are properly licensed to run offers in Azure.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

What are solutions for running Oracle WebLogic Server on the Azure Kubernetes Service?

Article • 08/22/2024

Applies to: ✔ Linux VMs

This page describes the solutions for running Oracle WebLogic Server (WLS) on the Azure Kubernetes Service (AKS). These solutions are jointly developed and supported by Oracle and Microsoft.

It's also possible to run WebLogic Server on Azure Virtual Machines. The solutions to do so are described in the article Running Oracle WebLogic Server on the Azure Virtual Machines.

WebLogic Server is a leading Java application server running some of the most mission-critical enterprise Java applications across the globe. WebLogic Server forms the middleware foundation for the Oracle software suite. Oracle and Microsoft are committed to empowering WebLogic Server customers with choice and flexibility to run workloads on Azure as a leading cloud platform.

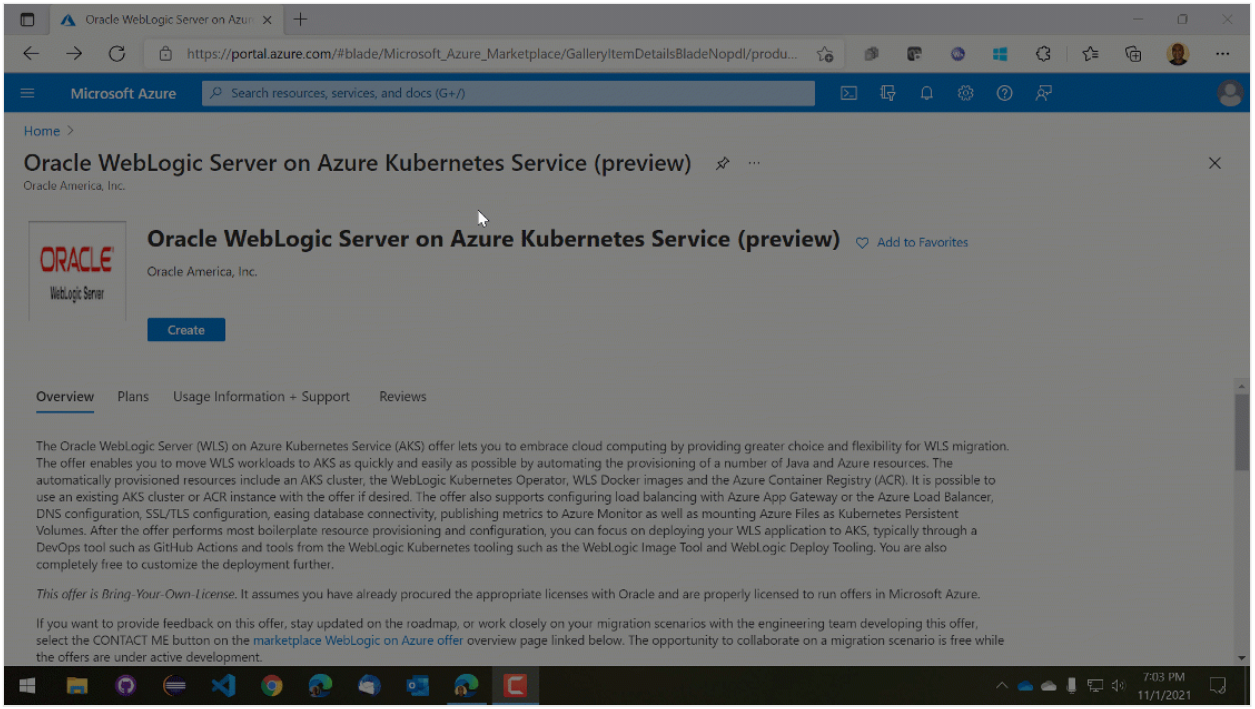
WebLogic Server on AKS certified and supported

WebLogic Server is certified by Oracle and Microsoft to run well on AKS. The WebLogic Server on AKS solutions are aimed at making it as easy as possible to run your containerized and orchestrated Java applications on Kubernetes. The solutions are focused on reliability, scalability, manageability, and enterprise support.

WebLogic Server clusters are fully enabled to run on Kubernetes via the WebLogic Kubernetes Operator (referred to simply as the 'Operator' from here onward). The Operator follows the standard Kubernetes Operator pattern. It simplifies the management and operation of WebLogic domains on Kubernetes by automating otherwise manual tasks and adding extra operational reliability features. The Operator supports Oracle WebLogic Server 12c, Oracle Fusion Middleware Infrastructure 12c and beyond. For details on the Operator, refer to the official documentation from Oracle.

WebLogic Server on AKS solution template

Beyond certifying WebLogic Server on AKS, Oracle and Microsoft jointly provide the Azure Marketplace offer of Oracle WebLogic Server on AKS with the goal of making it as quick and easy as possible to migrate WebLogic Server workloads to AKS. The offer does so by automating the provisioning of a number of Java and Azure resources. The automatically provisioned resources include an AKS cluster, the WebLogic Kubernetes Operator, WebLogic Server Docker images, and the Azure Container Registry (ACR). It's possible to use an existing AKS cluster or ACR instance with the offer. The offer also supports configuring load balancing with Azure App Gateway or the Azure Load Balancer, easing database connectivity, publishing metrics to Azure Monitor and mounting Azure Files as Kubernetes Persistent Volumes. The currently supported database integrations include Azure PostgreSQL, Azure MySQL, Azure SQL, and the Oracle Database on the Oracle Cloud or Azure.



After the solution template performs most boilerplate resource provisioning and configuration, you can focus on deploying your application to AKS, typically through a DevOps tool such as GitHub Actions and tools from WebLogic Kubernetes tooling such as the WebLogic Image Tool and WebLogic Deploy Tooling. You're completely free to customize the deployment further.

If you're interested in providing feedback or working closely on your migration scenarios with the engineering team developing WebLogic on AKS solutions, fill out this short survey on WebLogic migration and include your contact information. The team of program managers, architects, and engineers will promptly get in touch with you to initiate close collaboration.

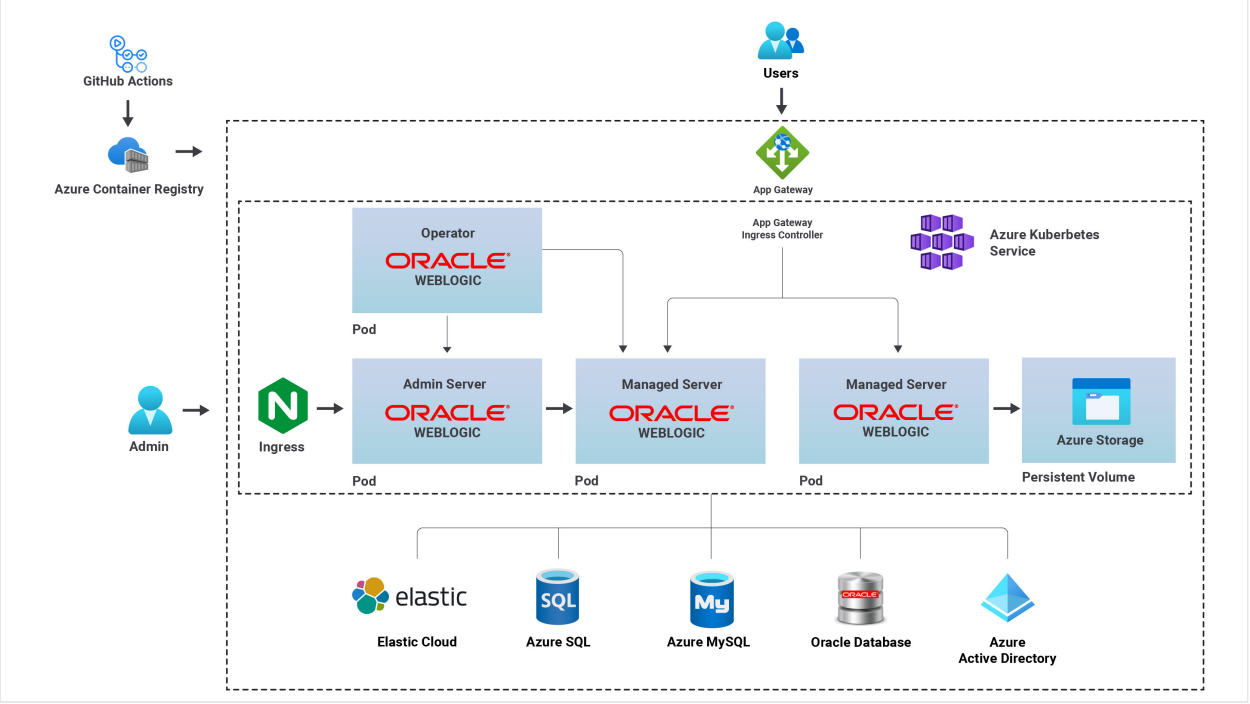
Manual guidance, scripts, and samples for WebLogic Server on AKS

Oracle and Microsoft also provide basic step-by-step guidance, scripts, and samples for running WebLogic Server on AKS. The guidance is suitable for customers that wish to remain as close as possible to a native Kubernetes manual deployment experience as an alternative to using a solution template. The guidance is incorporated into the Azure Kubernetes Service sample section of the Operator documentation. The guidance allows a very high degree of configuration and customization.

The guidance supports two ways of deploying WebLogic Server domains to AKS. Domains can be deployed directly to Kubernetes Persistent Volumes. This deployment option is good if you want to migrate to AKS but still want to administer WebLogic Server using the Admin Console or the WebLogic Scripting Tool (WLST). The option also allows you to move to AKS without adopting Docker development. The more Kubernetes native way of deploying WebLogic Server domains to AKS is to build custom container images based on official WebLogic Server images from the Oracle Container Registry, publish the custom images to ACR and deploy the domain to AKS using the Operator.

Deployment architectures

The solutions for running Oracle WebLogic Server on the Azure Kubernetes Service enable a wide range of production-ready deployment architectures with relative ease.



Beyond what the solutions provide you have complete flexibility to customize your deployments further. It's likely on top of deploying applications you'll integrate further Azure resources with your deployments or tune the deployments to your specific applications. You're encouraged to provide feedback in the survey on further improving WebLogic on AKS solutions.

Next steps

The following articles provide more information on getting started with these technologies.

- Deploy a Java application with WebLogic Server on an Azure Kubernetes Service (AKS) cluster
- What are solutions for running Oracle WebLogic Server on Azure Virtual Machines?

For more information about the Oracle WebLogic offers at Azure Marketplace, see Oracle WebLogic Server on Azure. These offers are all *Bring-Your-Own-License*. They assume that you already have the appropriate licenses with Oracle and are properly licensed to run offers in Azure.

You're encouraged to connect with the development team and provide feedback on further improving WebLogic on AKS solutions.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Deploy a Java application with WebLogic Server on an Azure Kubernetes Service (AKS) cluster

Article • 08/01/2024

This article demonstrates how to:

- Run your Java application on Oracle WebLogic Server (WLS).
- Stand up a WebLogic Server cluster on AKS using an Azure Marketplace offer.
- Build an application Docker image that includes WebLogic Deploy Tooling (WDT) models.
- Deploy the containerized application to the WebLogic Server cluster on AKS with connection to Microsoft Azure SQL.

This article uses the Azure Marketplace offer for WebLogic Server to accelerate your journey to AKS. The offer automatically provisions several Azure resources, including the following resources:

- An Azure Container Registry instance
- An AKS cluster
- An Azure App Gateway Ingress Controller (AGIC) instance
- The WebLogic Operator
- A container image including the WebLogic runtime
- A WebLogic Server cluster without an application

Then, the article introduces building an image to update the WebLogic Server cluster. The image provides the application and WDT models.

If you prefer a less automated approach to deploying WebLogic on AKS, see the step-by-step guidance included in the official documentation from Oracle for Azure Kubernetes Service.

If you're interested in providing feedback or working closely on your migration scenarios with the engineering team developing WebLogic on AKS solutions, fill out this short survey on WebLogic migration and include your contact information. The team of program managers, architects, and engineers will promptly get in touch with you to initiate close collaboration.

Prerequisites

- If you don't have an Azure subscription, create an Azure free account before you begin.
- Ensure the Azure identity you use to sign in and complete this article has either the Owner role in the current subscription or the Contributor and User Access Administrator roles in the current subscription. For an overview of Azure roles, see What is Azure role-based access control (Azure RBAC)? For details on the specific roles required by WLS on AKS, see Azure built-in roles.
- Have the credentials for an Oracle single sign-on (SSO) account. To create one, see Create Your Oracle Account.
- Accept the license terms for WebLogic Server.

- Visit the Oracle Container Registry and sign in.
- If you have a support entitlement, select **Middleware**, then search for and select **weblogic_cpu**.
- If you don't have a support entitlement from Oracle, select **Middleware**, then search for and select **weblogic**.
- Accept the license agreement.

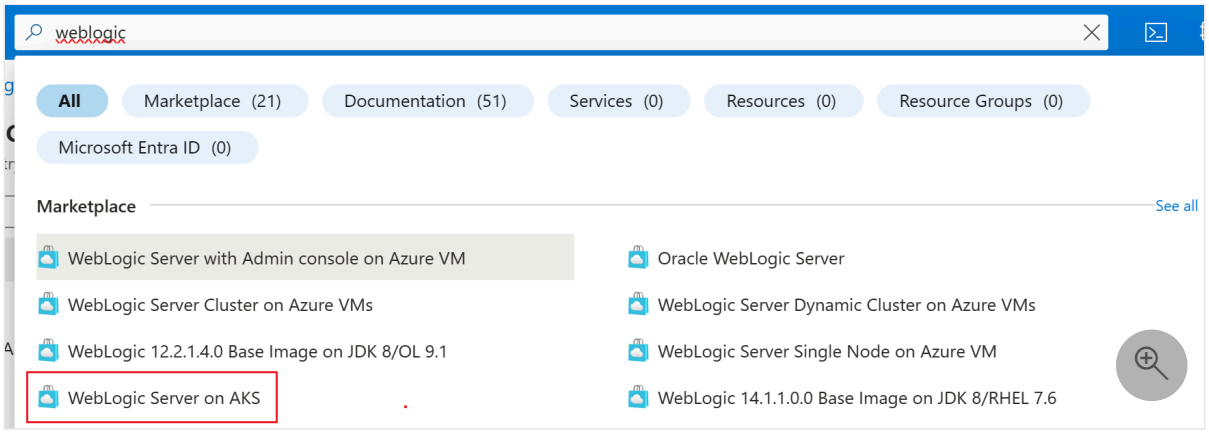
> **Note**
>
> Get a support entitlement from Oracle before going to production. Failure to do so results in running insecure images that are not patched for critical security flaws. For more information on Oracle's critical patch updates, see Critical Patch Updates, Security Alerts and Bulletins from Oracle.

- Prepare a local machine with Unix-like operating system installed - for example, Ubuntu, Azure Linux, macOS, Windows Subsystem for Linux.
  - Azure CLI. Use `az --version` to test whether az works. This document was tested with version 2.55.1.
  - kubectl. Use `kubectl version` to test whether kubectl works. This document was tested with version v1.21.2.
  - A Java Development Kit (JDK) compatible with the version of WebLogic Server you intend to run. The article directs you to install a version of WebLogic Server that uses JDK 11. Ensure that your `JAVA_HOME` environment variable is set correctly in the shells in which you run the commands.
  - Maven 3.5.0 or higher.
  - Ensure that you have the zip/unzip utility installed. Use `zip/unzip -v` to test whether `zip/unzip` works.

Deploy WebLogic Server on AKS

The following steps show you how to find the WebLogic Server on AKS offer and fill out the **Basics** pane.

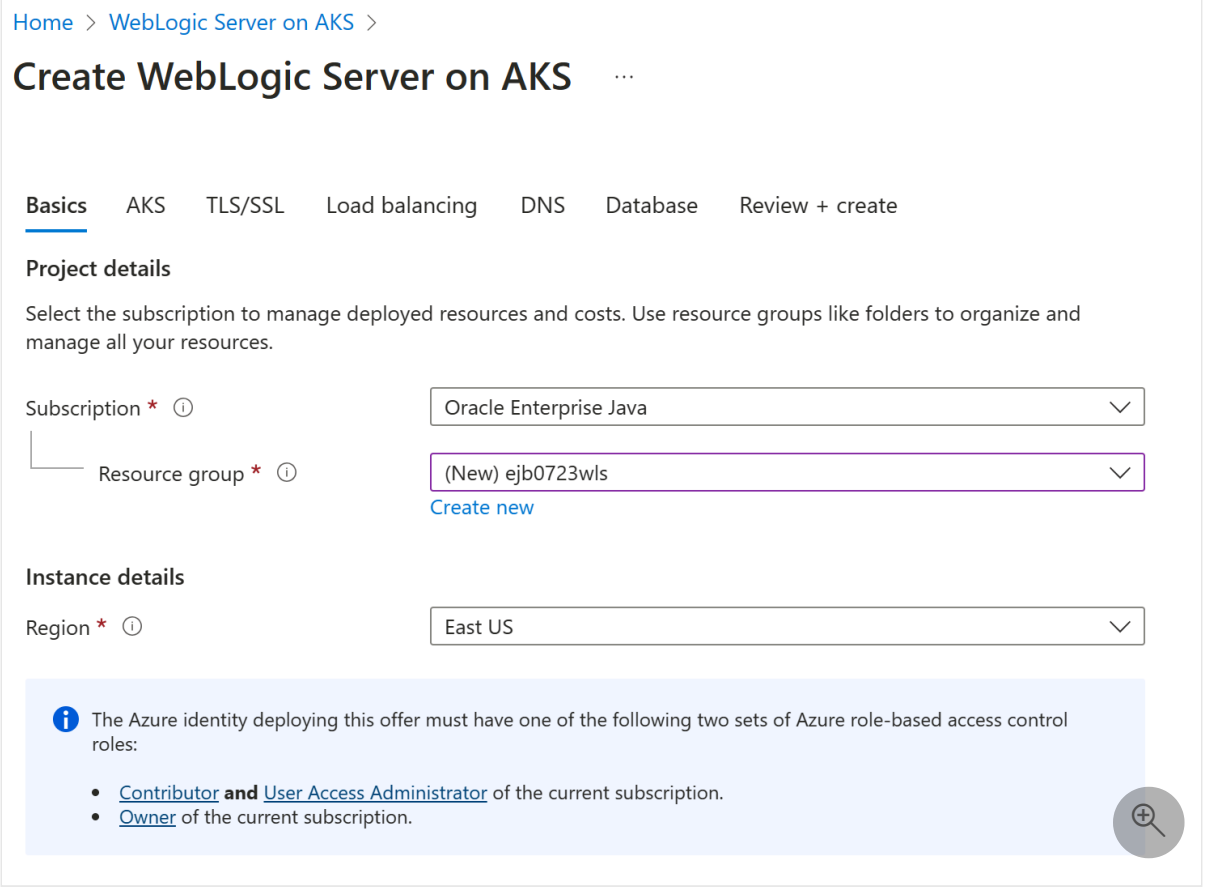
1. In the search bar at the top of the Azure portal, enter *weblogic*. In the autosuggested search results, in the **Marketplace** section, select **WebLogic Server on AKS**.



You can also go directly to the WebLogic Server on AKS offer.

2. On the offer page, select **Create**.

3. On the **Basics** pane, ensure the value shown in the **Subscription** field is the same one that you logged into in Azure. Make sure you have the roles listed in the prerequisites section for the subscription.



4. You must deploy the offer in an empty resource group. In the **Resource group** field, select **Create new** and then fill in a value for the resource group. Because resource groups must be unique within a subscription, pick a unique name. An easy way to have unique names is to use a combination of your initials, today's date, and some identifier - for example, `ejb0723wls`.

5. Under **Instance details**, select the region for the deployment. For a list of Azure regions where AKS is available, see AKS region availability.

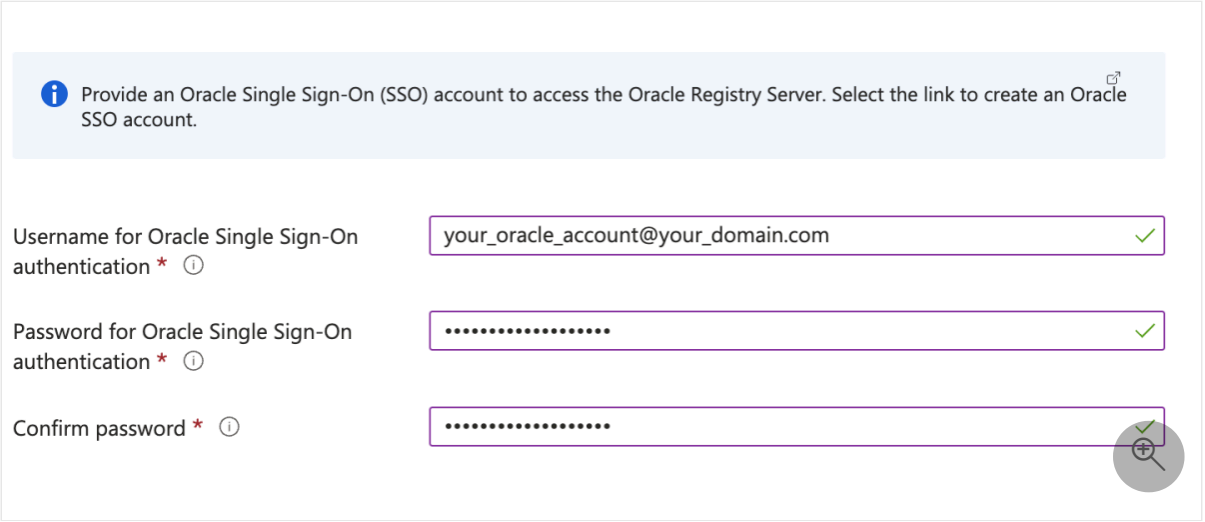
6. Under **Credentials for WebLogic**, leave the default value for **Username for WebLogic Administrator**.

7. Fill in `wlsAksCluster2022` for the **Password for WebLogic Administrator**. Use the same value for the confirmation and **Password for WebLogic Model encryption** fields.

8. Select **Next**.

The following steps show you how to start the deployment process.

1. Scroll to the section labeled **Provide an Oracle Single Sign-On (SSO) account**. Fill in your Oracle SSO credentials from the preconditions.



2. Make sure you note the steps in the info box starting with **Before moving forward, you must accept the Oracle Standard Terms and Restrictions.**

3. Depending on whether or not the Oracle SSO account has an Oracle support entitlement, select the appropriate option for **Select the type of WebLogic Server Images**. If the account has a support entitlement, select **Patched WebLogic Server Images**. Otherwise, select **General WebLogic Server Images**.

4. Leave the value in **Select desired combination of WebLogic Server...** at its default value. You have a broad range of choices for WebLogic Server, JDK, and OS version.

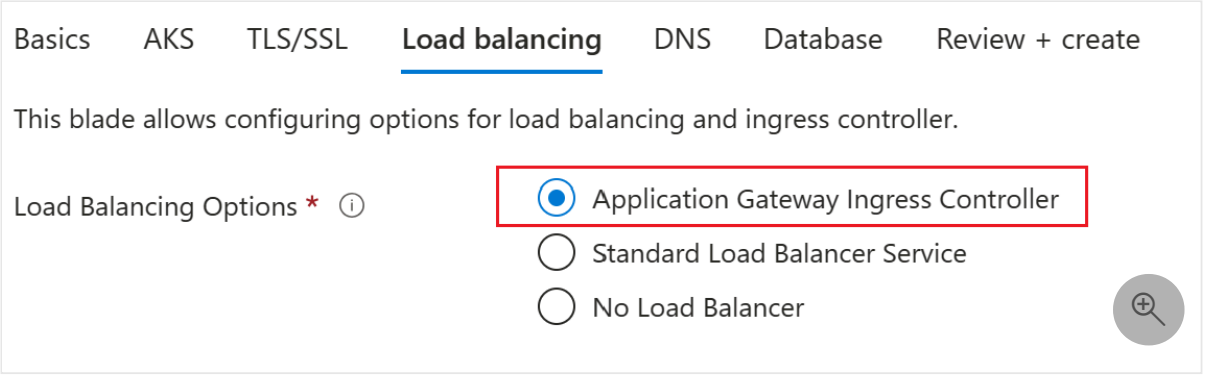
5. In the **Application** section, next to **Deploy an application?**, select **No**.

The following steps make it so the WebLogic Server admin console and the sample app are exposed to the public Internet with a built-in Application Gateway ingress add-on. For a more information, see What is Application Gateway Ingress Controller?

1. Select **Next** to see the **TLS/SSL** pane.

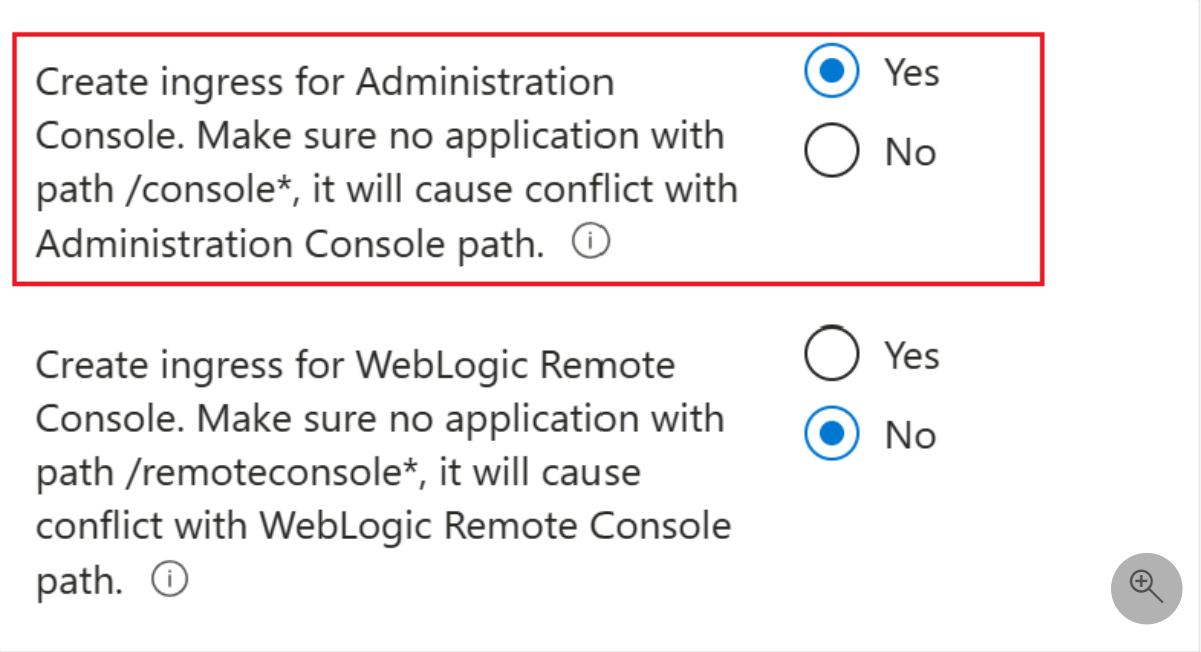
2. Select **Next** to see the **Load balancing** pane.

3. Next to **Load Balancing Options**, select **Application Gateway Ingress Controller**.



4. Under the **Application Gateway Ingress Controller**, you should see all fields prepopulated with the defaults for **Virtual network** and **Subnet**. Leave the default values.

5. For **Create ingress for Administration Console**, select **Yes**.



6. Leave the default values for other fields.
7. Select **Review + create**. Ensure the validation doesn't fail. If it fails, fix any validation problems, then select **Review + create** again.
8. Select **Create**.
9. Track the progress of the deployment on the **Deployment is in progress** page.

Depending on network conditions and other activity in your selected region, the deployment might take up to 50 minutes to complete.

You can perform the steps in the section Create an Azure SQL Database while you wait. Return to this section when you finish creating the database.

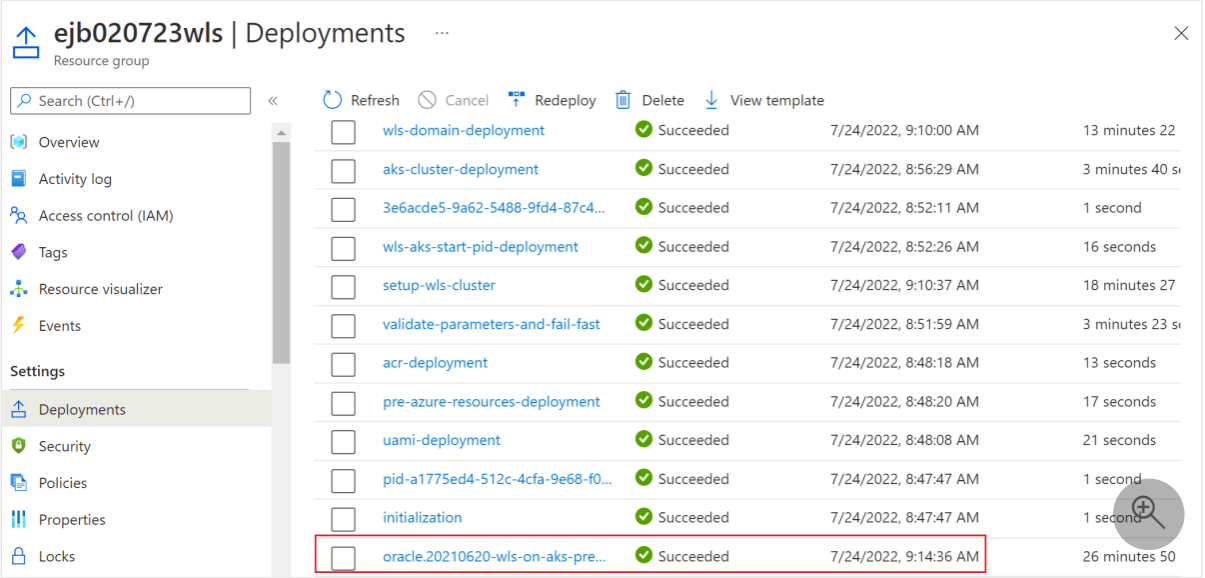
Examine the deployment output

Use the steps in this section to verify that the deployment was successful.

If you navigated away from the **Deployment is in progress** page, the following steps show you how to get back to that page. If you're still on the page that shows **Your deployment is complete**, you can skip to step 5 after the next screenshot.

1. In the corner of any Azure portal page, select the hamburger menu and select **Resource groups**.
2. In the box with the text **Filter for any field**, enter the first few characters of the resource group you created previously. If you followed the recommended convention, enter your initials, then select the appropriate resource group.
3. In the navigation pane, in the **Settings** section, select **Deployments**. You see an ordered list of the deployments to this resource group, with the most recent one first.

4. Scroll to the oldest entry in this list. This entry corresponds to the deployment you started in the preceding section. Select the oldest deployment, as shown in the following screenshot.



5. In the navigation pane, select **Outputs**. This list shows the output values from the deployment. Useful information is included in the outputs.

6. The **adminConsoleExternalUrl** value is the fully qualified, public Internet visible link to the WebLogic Server admin console for this AKS cluster. Select the copy icon next to the field value to copy the link to your clipboard. Save this value aside for later.

7. The **clusterExternalUrl** value is the fully qualified, public Internet visible link to the sample app deployed in WebLogic Server on this AKS cluster. Select the copy icon next to the field value to copy the link to your clipboard. Save this value aside for later.

8. The **shellCmdtoOutputWlsImageModelYaml** value is the base64 string of the WDT model that is used to build the container image. Save this value aside for later.

9. The **shellCmdtoOutputWlsImageProperties** value is the base64 string of the WDT model properties that is used to build the container image. Save this value aside for later.

10. The **shellCmdtoConnectAks** value is the Azure CLI command to connect to this specific AKS cluster.

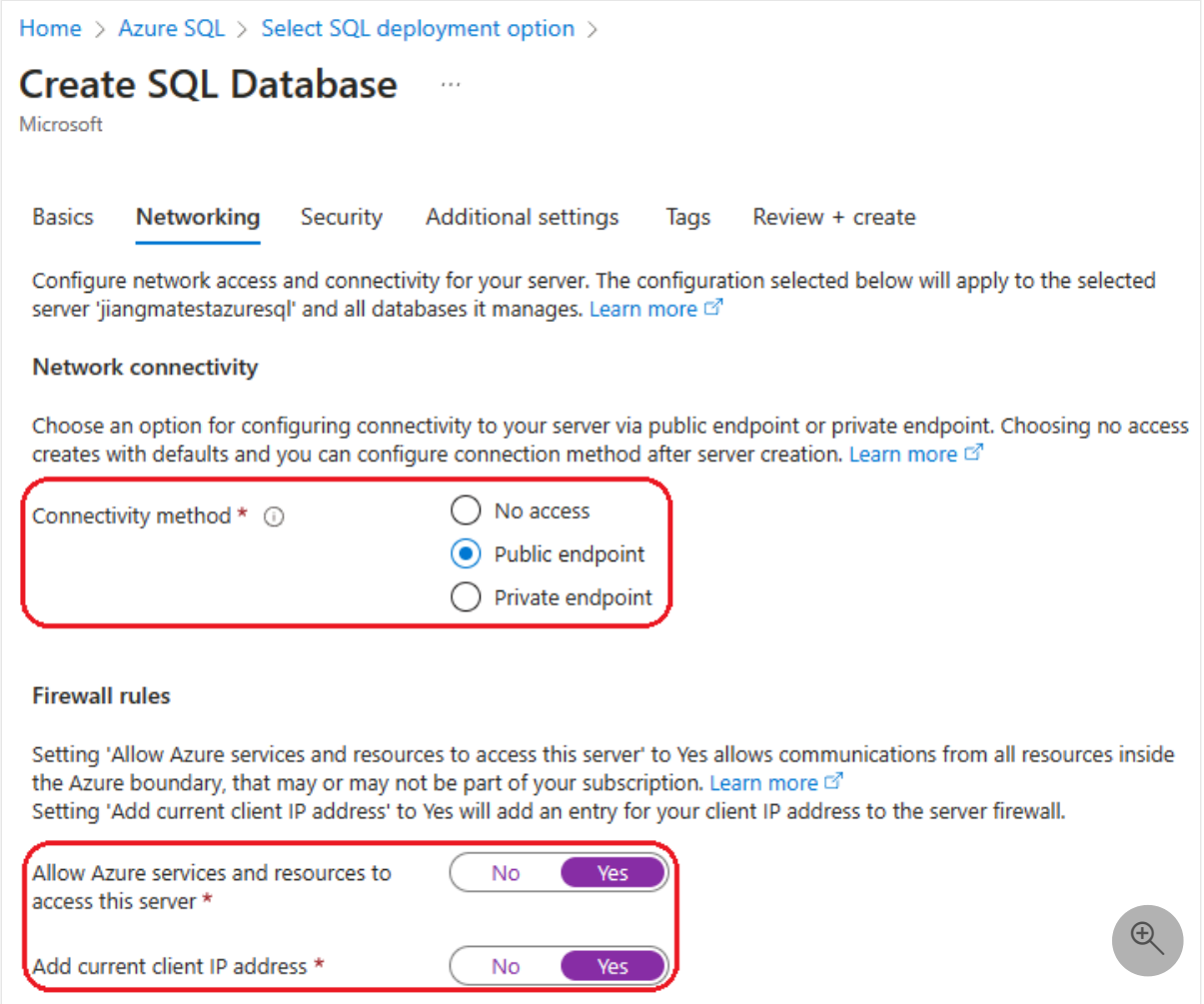
The other values in the outputs are beyond the scope of this article, but are explained in detail in the WebLogic on AKS user guide.

Create an Azure SQL Database

To create an Azure SQL Database single database for use with your app, follow the steps in Quickstart: Create a single database in Azure SQL Database. Carefully note the following differences:

- At the **Basics** step, write down the values for **Resource group**, **Database name**, ***<server-name>*.database.windows.net**, **Server admin login**, and **Password**. This article refers to the database **Resource group** value as `<db-resource-group>`.

- At the **Networking** step, set **Connectivity method** to **Public endpoint**, set **Allow Azure services and resources to access this server** to **Yes**, and set **Add current client IP address** to **Yes**.



> **Note**
>
> The serverless compute tier that you selected for this database saves money by putting the database to sleep during periods of inactivity. The sample app will fail if the database is asleep when the app starts up.
>
> To force the database to wake up, you can run a query by using the query editor. Follow the steps in Query the database. Here's an example query: `SELECT * FROM COFFEE;`.

Then, create a schema for the sample application by using the following steps:

1. Open the **Query editor** pane by following the steps in the Query the database section of Quickstart: Create a single database - Azure SQL Database.

2. Enter and run the following query:

```SQL
CREATE TABLE COFFEE (ID NUMERIC(19) NOT NULL, NAME VARCHAR(255) NULL, PRICE
FLOAT(32) NULL, PRIMARY KEY (ID));
CREATE TABLE SEQUENCE (SEQ_NAME VARCHAR(50) NOT NULL, SEQ_COUNT NUMERIC(28) NULL,
```

```
PRIMARY KEY (SEQ_NAME));
INSERT INTO SEQUENCE VALUES ('SEQ_GEN',0);
```

After a successful run, you should see the message **Query succeeded: Affected rows: 1**. If you don't see this message, troubleshoot and resolve the problem before proceeding.

The database, tables, AKS cluster, and WebLogic Server cluster are created. If you want, you can explore the admin console by opening a browser and navigating to the address of **adminConsoleExternalUrl**. Sign in with the values you entered during the WebLogic Server on AKS deployment.

You can proceed to preparing AKS to host your WebLogic application.

Configure and deploy the sample application

The offer provisions the WebLogic Server cluster via model in image. Currently, the WebLogic Server cluster has no application deployed.

This section updates the WebLogic Server cluster by deploying a sample application using auxiliary image.

Check out the application

In this section, you clone the sample code for this guide. The sample is on GitHub in the weblogic-on-azure repository in the *javaee/weblogic-cafe/* folder. Here's the file structure of the application.

text

```
weblogic-cafe
├── pom.xml
└── src
    └── main
        ├── java
        │   └── cafe
        │       ├── model
        │       │   ├── CafeRepository.java
        │       │   └── entity
        │       │       └── Coffee.java
        │       └── web
        │           ├── rest
        │           │   └── CafeResource.java
        │           └── view
        │               └── Cafe.java
        ├── resources
        │   ├── META-INF
        │   │   └── persistence.xml
        │   └── cafe
        │       └── web
        │           ├── messages.properties
        │           └── messages_es.properties
        └── webapp
            ├── WEB-INF
            │   ├── beans.xml
```

```
            │   ├── faces-config.xml
            │   └── web.xml
            ├── index.xhtml
            └── resources
                └── components
                    └── inputPrice.xhtml
```

Use the following commands to clone the repository:

```bash
cd <parent-directory-to-check-out-sample-code>
export BASE_DIR=$PWD
git clone --single-branch https://github.com/microsoft/weblogic-on-azure.git --branch
20240201 $BASE_DIR/weblogic-on-azure
```

If you see a message about being in "detached HEAD" state, this message is safe to ignore. It just means you checked out a tag.

Use the following command to build *javaee/weblogic-cafe/*:

```bash
mvn clean package --file $BASE_DIR/weblogic-on-azure/javaee/weblogic-cafe/pom.xml
```

The package should be successfully generated and located at *$BASE_DIR/weblogic-on-azure/javaee/weblogic-cafe/target/weblogic-cafe.war*. If you don't see the package, you must troubleshoot and resolve the issue before you continue.

Use ACR to create an auxiliary image

The steps in this section show you how to build an auxiliary image. This image includes the following components:

- The *Model in Image* model files
- Your application
- The Java Database Connectivity (JDBC) driver archive file
- The WebLogic Deploy Tooling installation

An *auxiliary image* is a Docker container image containing your app and configuration. The WebLogic Kubernetes Operator combines your auxiliary image with the `domain.spec.image` in the AKS cluster that contains the WebLogic Server, JDK, and operating system. For more information about auxiliary images, see Auxiliary images in the Oracle documentation.

This section requires a Linux terminal with Azure CLI and kubectl installed.

Use the following steps to build the image:

1. Use the following commands to create a directory to stage the models and application:

```bash
mkdir -p ${BASE_DIR}/mystaging/models
cd ${BASE_DIR}/mystaging/models
```

2. Copy the **shellCmdtoOutputWlsImageModelYaml** value that you saved from the deployment outputs, paste it into the Bash window, and run the command. The command should look similar to the following example:

```bash
echo -e IyBDb3B5cmlna...Cgo= | base64 -d > model.yaml
```

This command produces a *${BASE_DIR}/mystaging/models/model.yaml* file with contents similar to the following example:

```yaml
# Copyright (c) 2020, 2021, Oracle and/or its affiliates.
# Licensed under the Universal Permissive License v 1.0 as shown at
https://oss.oracle.com/licenses/upl.

# Based on ./kubernetes/samples/scripts/create-weblogic-domain/model-in-
image/model-images/model-in-image__WLS-v1/model.10.yaml
# in https://github.com/oracle/weblogic-kubernetes-operator.

domainInfo:
  AdminUserName: "@@SECRET:__weblogic-credentials__:username@@"
  AdminPassword: "@@SECRET:__weblogic-credentials__:password@@"
  ServerStartMode: "prod"

topology:
  Name: "@@ENV:CUSTOM_DOMAIN_NAME@@"
  ProductionModeEnabled: true
  AdminServerName: "admin-server"
  Cluster:
    "cluster-1":
      DynamicServers:
        ServerTemplate: "cluster-1-template"
        ServerNamePrefix: "@@ENV:MANAGED_SERVER_PREFIX@@"
        DynamicClusterSize: "@@PROP:CLUSTER_SIZE@@"
        MaxDynamicClusterSize: "@@PROP:CLUSTER_SIZE@@"
        MinDynamicClusterSize: "0"
        CalculatedListenPorts: false
  Server:
    "admin-server":
      ListenPort: 7001
  ServerTemplate:
    "cluster-1-template":
      Cluster: "cluster-1"
      ListenPort: 8001
  SecurityConfiguration:
    NodeManagerUsername: "@@SECRET:__weblogic-credentials__:username@@"
    NodeManagerPasswordEncrypted: "@@SECRET:__weblogic-credentials__:password@@"

resources:
  SelfTuning:
```

```
    MinThreadsConstraint:
      SampleMinThreads:
        Target: "cluster-1"
        Count: 1
    MaxThreadsConstraint:
      SampleMaxThreads:
        Target: "cluster-1"
        Count: 10
    WorkManager:
      SampleWM:
        Target: "cluster-1"
        MinThreadsConstraint: "SampleMinThreads"
        MaxThreadsConstraint: "SampleMaxThreads"
```

3. In a similar way, copy the **shellCmdtoOutputWlsImageProperties** value, paste it into the Bash window, and run the command. The command should look similar to the following example:

```bash
echo -e IyBDb3B5cml...pFPTUK | base64 -d > model.properties
```

This command produces a *${BASE_DIR}/mystaging/models/model.properties* file with contents similar to the following example:

```properties
# Copyright (c) 2021, Oracle Corporation and/or its affiliates.
# Licensed under the Universal Permissive License v 1.0 as shown at
https://oss.oracle.com/licenses/upl.

# Based on ./kubernetes/samples/scripts/create-weblogic-domain/model-in-
image/model-images/model-in-image__WLS-v1/model.10.properties
# in https://github.com/oracle/weblogic-kubernetes-operator.

CLUSTER_SIZE=5
```

4. Use the following steps to create the application model file.

   a. Use the following commands to copy *weblogic-cafe.war* and save it to *wlsdeploy/applications*:

   ```bash
   mkdir -p ${BASE_DIR}/mystaging/models/wlsdeploy/applications
   cp $BASE_DIR/weblogic-on-azure/javaee/weblogic-cafe/target/weblogic-cafe.war
   ${BASE_DIR}/mystaging/models/wlsdeploy/applications/weblogic-cafe.war
   ```

   b. Use the following commands to create the application model file with the contents shown. Save the model file to *${BASE_DIR}/mystaging/models/appmodel.yaml*.

   ```bash
   cat <<EOF >appmodel.yaml
   appDeployments:
   ```

```
  Application:
    weblogic-cafe:
      SourcePath: 'wlsdeploy/applications/weblogic-cafe.war'
      ModuleType: ear
      Target: 'cluster-1'
EOF
```

5. Use the following commands to download and install Microsoft SQL Server JDBC driver to *wlsdeploy/externalJDBCLibraries*:

```bash
export DRIVER_VERSION="10.2.1.jre8"
export
MSSQL_DRIVER_URL="https://repo.maven.apache.org/maven2/com/microsoft/sqlserver/mssql-jdbc/${DRIVER_VERSION}/mssql-jdbc-${DRIVER_VERSION}.jar"

mkdir ${BASE_DIR}/mystaging/models/wlsdeploy/externalJDBCLibraries
curl -m 120 -fL ${MSSQL_DRIVER_URL} -o
${BASE_DIR}/mystaging/models/wlsdeploy/externalJDBCLibraries/mssql-
jdbc-${DRIVER_VERSION}.jar
```

6. Next, use the following commands to create the database connection model file with the contents shown. Save the model file to *${BASE_DIR}/mystaging/models/dbmodel.yaml*. The model uses placeholders (secret `sqlserver-secret`) for database username, password, and URL. Make sure the following fields are set correctly. The following model names the resource with `jdbc/WebLogicCafeDB`.

Expand table

Item Name	Field	Value
JNDI name	`resources.JDBCSystemResource.<resource-name>.JdbcResource.JDBCDataSourceParams.JNDIName`	`jdbc/WebLogicCafeDB`
Driver name	`resources.JDBCSystemResource.<resource-name>.JDBCDriverParams.DriverName`	`com.microsoft.sqlserver.jdbc.SQLServerDriver`
Database Url	`resources.JDBCSystemResource.<resource-name>.JDBCDriverParams.URL`	`@@SECRET:sqlserver-secret:url@@`
Database password	`resources.JDBCSystemResource.<resource-name>.JDBCDriverParams.PasswordEncrypted`	`@@SECRET:sqlserver-secret:password@@`
Database username	`resources.JDBCSystemResource.<resource-name>.JDBCDriverParams.Properties.user.Value`	`'@@SECRET:sqlserver-secret:user@@'`

```bash
cat <<EOF >dbmodel.yaml
resources:
  JDBCSystemResource:
    jdbc/WebLogicCafeDB:
```

```
      Target: 'cluster-1'
      JdbcResource:
        JDBCDataSourceParams:
          JNDIName: [
            jdbc/WebLogicCafeDB
          ]
          GlobalTransactionsProtocol: None
        JDBCDriverParams:
          DriverName: com.microsoft.sqlserver.jdbc.SQLServerDriver
          URL: '@@SECRET:sqlserver-secret:url@@'
          PasswordEncrypted: '@@SECRET:sqlserver-secret:password@@'
          Properties:
            user:
              Value: '@@SECRET:sqlserver-secret:user@@'
        JDBCConnectionPoolParams:
          TestTableName: SQL SELECT 1
          TestConnectionsOnReserve: true
EOF
```

7. Use the following commands to create an application archive file and then remove the *wlsdeploy* folder, which you don't need anymore:

```bash
cd ${BASE_DIR}/mystaging/models
zip -r archive.zip wlsdeploy

rm -f -r wlsdeploy
```

8. Use the following commands to download and install WebLogic Deploy Tooling (WDT) in the staging directory and remove its *weblogic-deploy/bin/*.cmd* files, which aren't used in UNIX environments:

```bash
cd ${BASE_DIR}/mystaging
curl -m 120 -fL https://github.com/oracle/weblogic-deploy-
tooling/releases/latest/download/weblogic-deploy.zip -o weblogic-deploy.zip

unzip weblogic-deploy.zip -d .
rm ./weblogic-deploy/bin/*.cmd
```

9. Use the following command to remove the WDT installer:

```bash
rm weblogic-deploy.zip
```

10. Use the following commands to create a docker file:

```bash
cd ${BASE_DIR}/mystaging
cat <<EOF >Dockerfile
```

```
FROM busybox
ARG AUXILIARY_IMAGE_PATH=/auxiliary
ARG USER=oracle
ARG USERID=1000
ARG GROUP=root
ENV AUXILIARY_IMAGE_PATH=\${AUXILIARY_IMAGE_PATH}
RUN adduser -D -u \${USERID} -G \$GROUP \$USER
COPY --chown=\$USER:\$GROUP ./ \${AUXILIARY_IMAGE_PATH}/
USER \$USER
EOF
```

11. Run the `az acr build` command using *${BASE_DIR}/mystaging/Dockerfile*, as shown in the following example:

Bash

```bash
export ACR_NAME=<value-from-clipboard>
export IMAGE="wlsaks-auxiliary-image:1.0"
```

12. Use the following commands to double-check the staging files:

Bash

```bash
cd ${BASE_DIR}/mystaging
find -maxdepth 2 -type f -print
```

These commands produce output similar to the following example:

Output

```
./models/model.properties
./models/model.yaml
./models/archive.zip
./Dockerfile
./weblogic-deploy/VERSION.txt
./weblogic-deploy/LICENSE.txt
```

13. Build the image with `az acr build`, as shown in the following example:

Bash

```bash
az acr build -t ${IMAGE} --build-arg AUXILIARY_IMAGE_PATH=/auxiliary -r
${ACR_NAME} --platform linux/amd64 .
```

When you build the image successfully, the output looks similar to the following example:

Output

```
...
Step 1/9 : FROM busybox
latest: Pulling from library/busybox
Digest: sha256:9ae97d36d26566ff84e8893c64a6dc4fe8ca6d1144bf5b87b2b85a32def253c7
Status: Image is up to date for busybox:latest
```

```
---> 65ad0d468eb1
Step 2/9 : ARG AUXILIARY_IMAGE_PATH=/auxiliary
---> Running in 1f8f4e82ccb6
Removing intermediate container 1f8f4e82ccb6
---> 947fde618be9
Step 3/9 : ARG USER=oracle
---> Running in dda021591e41
Removing intermediate container dda021591e41
---> f43d84be4517
Step 4/9 : ARG USERID=1000
---> Running in cac4df6dfd13
Removing intermediate container cac4df6dfd13
---> e5513f426c74
Step 5/9 : ARG GROUP=root
---> Running in 8fec1763270c
Removing intermediate container 8fec1763270c
---> 9ef233dbe279
Step 6/9 : ENV AUXILIARY_IMAGE_PATH=${AUXILIARY_IMAGE_PATH}
---> Running in b7754f58157a
Removing intermediate container b7754f58157a
---> 4a26a97eb572
Step 7/9 : RUN adduser -D -u ${USERID} -G $GROUP $USER
---> Running in b6c1f1a81af1
Removing intermediate container b6c1f1a81af1
---> 97d3e5ad7540
Step 8/9 : COPY --chown=$USER:$GROUP ./ ${AUXILIARY_IMAGE_PATH}/
---> 21088171876f
Step 9/9 : USER $USER
---> Running in 825e0abc9f6a
Removing intermediate container 825e0abc9f6a
---> b81d6430fcda
Successfully built b81d6430fcda
Successfully tagged wlsaksacru6jyly7kztoqu.azurecr.io/wlsaks-auxiliary-image:1.0
2024/08/28 03:06:19 Successfully executed container: build
2024/08/28 03:06:19 Executing step ID: push. Timeout(sec): 3600, Working
directory: '', Network: ''
2024/08/28 03:06:19 Pushing image: wlsaksacru6jyly7kztoqu.azurecr.io/wlsaks-
auxiliary-image:1.0, attempt 1
The push refers to repository [wlsaksacru6jyly7kztoqu.azurecr.io/wlsaks-auxiliary-
image]
ee589b3cda86: Preparing
c1fd1adab3b9: Preparing
d51af96cf93e: Preparing
c1fd1adab3b9: Pushed
d51af96cf93e: Pushed
ee589b3cda86: Pushed
1.0: digest:
sha256:c813eb75576eb07a179c3cb4e70106ca7dd280f933ab33a2f6858de673b12eac size: 946
2024/08/28 03:06:21 Successfully pushed image:
wlsaksacru6jyly7kztoqu.azurecr.io/wlsaks-auxiliary-image:1.0
2024/08/28 03:06:21 Step ID: build marked as successful (elapsed time in seconds:
8.780235)
2024/08/28 03:06:21 Populating digests for step ID: build...
2024/08/28 03:06:22 Successfully populated digests for step ID: build
2024/08/28 03:06:22 Step ID: push marked as successful (elapsed time in seconds:
1.980158)
2024/08/28 03:06:22 The following dependencies were found:
2024/08/28 03:06:22
- image:
   registry: wlsaksacru6jyly7kztoqu.azurecr.io
   repository: wlsaks-auxiliary-image
```

```
   tag: "1.0"
   digest: sha256:c813eb75576eb07a179c3cb4e70106ca7dd280f933ab33a2f6858de673b12eac
runtime-dependency:
   registry: registry.hub.docker.com
   repository: library/busybox
   tag: latest
   digest: sha256:9ae97d36d26566ff84e8893c64a6dc4fe8ca6d1144bf5b87b2b85a32def253c7
git: {}

Run ID: ca1 was successful after 14s
```

The image is pushed to ACR after a success build.

14. You can run `az acr repository show` to test whether the image is pushed to the remote repository successfully, as shown in the following example:

    Bash

    ```
    az acr repository show --name ${ACR_NAME} --image ${IMAGE}
    ```

    This command should produce output similar to the following example:

    Output

    ```
    {
       "changeableAttributes": {
          "deleteEnabled": true,
          "listEnabled": true,
          "readEnabled": true,
          "writeEnabled": true
       },
       "createdTime": "2024-01-24T06:14:19.4546321Z",
       "digest":
    "sha256:a1befbefd0181a06c6fe00848e76f1743c1fecba2b42a975e9504ba2aaae51ea",
       "lastUpdateTime": "2024-01-24T06:14:19.4546321Z",
       "name": "1.0",
       "quarantineState": "Passed",
       "signed": false
    }
    ```

Apply the auxiliary image

In the previous steps, you created the auxiliary image including models and WDT. Before you apply the auxiliary image to the WebLogic Server cluster, use the following steps to create the secret for the datasource URL, username, and password. The secret is used as part of the placeholder in *dbmodel.yaml*.

1. Connect to the AKS cluster by copying the **shellCmdtoConnectAks** value that you saved aside previously, pasting it into the Bash window, then running the command. The command should look similar to the following example:

   Bash

```
az account set --subscription <subscription>;
az aks get-credentials \
    --resource-group <resource-group> \
    --name <name>
```

You should see output similar to the following example. If you don't see this output, troubleshoot and resolve the problem before continuing.

Output

```
Merged "<name>" as current context in /Users/<username>/.kube/config
```

2. Use the following steps to get values for the variables shown in the following table. You use these values to create the secret for the datasource connection.

Expand table

Variable	Description	Example
`DB_CONNECTION_STRING`	The connection string of SQL server.	`jdbc:sqlserver://server-name.database.windows.net:1433;database=wlsaksquickstart0125`
`DB_USER`	The username to sign in to the SQL server.	`welogic@sqlserverforwlsaks`
`DB_PASSWORD`	The password to sign in to the sQL server.	`Secret123456`

a. Visit the SQL database resource in the Azure portal.

b. In the navigation pane, under **Settings**, select **Connection strings**.

c. Select the **JDBC** tab.

d. Select the copy icon to copy the connection string to the clipboard.

e. For `DB_CONNECTION_STRING`, use the entire connection string, but replace the placeholder `{your_password_here}` with your database password.

f. For `DB_USER`, use the portion of the connection string from `azureuser` up to but not including `;password={your_password_here}`.

g. For `DB_PASSWORD`, use the value you entered when you created the database.

3. Use the following commands to create the Kubernetes Secret. This article uses the secret name `sqlserver-secret` for the secret of the datasource connection. If you use a different name, make sure the value is the same as the one in *dbmodel.yaml*.

   In the following commands, be sure to set the variables `DB_CONNECTION_STRING`, `DB_USER`, and `DB_PASSWORD` correctly by replacing the placeholder examples with the values described in the previous steps. To prevent the shell from interfering with them, enclose the value of the `DB_` variables in single quotes.

   Bash

   ```bash
   export DB_CONNECTION_STRING='<example-jdbc:sqlserver://server-name.database.windows.net:1433;database=wlsaksquickstart0125>'
   export DB_USER='<example-welogic@sqlserverforwlsaks>'
   export DB_PASSWORD='<example-Secret123456>'
   export WLS_DOMAIN_NS=sample-domain1-ns
   export WLS_DOMAIN_UID=sample-domain1
   export SECRET_NAME=sqlserver-secret
   export ACR_LOGIN_SERVER=$(az acr show --name ${ACR_NAME} --query "loginServer" --output tsv)

   kubectl -n ${WLS_DOMAIN_NS} create secret generic \
       ${SECRET_NAME} \
       --from-literal=password="${DB_PASSWORD}" \
       --from-literal=url="${DB_CONNECTION_STRING}" \
       --from-literal=user="${DB_USER}"

   kubectl -n ${WLS_DOMAIN_NS} label secret \
       ${SECRET_NAME} \
       weblogic.domainUID=${WLS_DOMAIN_UID}
   ```

   You must see the following output before you continue. If you don't see this output, troubleshoot and resolve the problem before you continue.

   Output

   ```
   secret/sqlserver-secret created
   secret/sqlserver-secret labeled
   ```

4. Apply the auxiliary image by patching the domain custom resource definition (CRD) using the `kubectl patch` command.

   The auxiliary image is defined in `spec.configuration.model.auxiliaryImages`, as shown in the following example.

   YAML

   ```yaml
   spec:
     clusters:
     - name: sample-domain1-cluster-1
     configuration:
       model:
         auxiliaryImages:
         - image: wlsaksacrafvzeyyswhxek.azurecr.io/wlsaks-auxiliary-image:1.0
   ```

```
        imagePullPolicy: IfNotPresent
        sourceModelHome: /auxiliary/models
        sourceWDTInstallHome: /auxiliary/weblogic-deploy
```

Use the following commands to increase the `restartVersion` value and use `kubectl patch` to apply the auxiliary image to the domain CRD using the definition shown:

Bash

```bash
export VERSION=$(kubectl -n ${WLS_DOMAIN_NS} get domain ${WLS_DOMAIN_UID} -
o=jsonpath='{.spec.restartVersion}' | tr -d "\"")
export VERSION=$((VERSION+1))

cat <<EOF >patch-file.json
[
  {
    "op": "replace",
    "path": "/spec/restartVersion",
    "value": "${VERSION}"
  },
  {
    "op": "add",
    "path": "/spec/configuration/model/auxiliaryImages",
    "value": [{"image": "$ACR_LOGIN_SERVER/$IMAGE", "imagePullPolicy":
"IfNotPresent", "sourceModelHome": "/auxiliary/models", "sourceWDTInstallHome":
"/auxiliary/weblogic-deploy"}]
  },
  {
    "op": "add",
    "path": "/spec/configuration/secrets",
    "value": ["${SECRET_NAME}"]
  }
]
EOF

kubectl -n ${WLS_DOMAIN_NS} patch domain ${WLS_DOMAIN_UID} \
    --type=json \
    --patch-file patch-file.json

kubectl get pod -n ${WLS_DOMAIN_NS} -w
```

5. Wait until the admin server and managed servers show the values in the following output block before you proceed:

Output

```
NAME                              READY   STATUS    RESTARTS   AGE
sample-domain1-admin-server       1/1     Running   0          20m
sample-domain1-managed-server1    1/1     Running   0          19m
sample-domain1-managed-server2    1/1     Running   0          18m
```

It might take 5-10 minutes for the system to reach this state. The following list provides an overview of what's happening while you wait:

- You should see the `sample-domain1-introspector` running first. This software looks for changes to the domain custom resource so it can take the necessary actions on the Kubernetes cluster.
- When changes are detected, the domain introspector kills and starts new pods to roll out the changes.
- Next, you should see the `sample-domain1-admin-server` pod terminate and restart.
- Then, you should see the two managed servers terminate and restart.
- Only when all three pods show the `1/1 Running` state, is it ok to proceed.

Verify the functionality of the deployment

Use the following steps to verify the functionality of the deployment by viewing the WebLogic Server admin console and the sample app:

1. Paste the **adminConsoleExternalUrl** value into the address bar of an Internet-connected web browser. You should see the familiar WebLogic Server admin console sign-in screen.

2. Sign in with the username `weblogic` and the password you entered when deploying WebLogic Server from the Azure portal. Recall that this value is `wlsAksCluster2022`.

3. In the **Domain Structure** box, select **Services**.

4. Under the **Services**, select **Data Sources**.

5. In the **Summary of JDBC Data Sources** panel, select **Monitoring**. Your screen should look similar to the following example. You find the state of data source is running on managed servers.

JDBC DataSource Runtime Statistics (Filtered - More Columns Exist)

Name	Type	Resource	Scope	Server	State	Active Connections Average Count
jdbc/WebLogicCafeDB	Generic	System	Global	managed-server2	Running	0
jdbc/WebLogicCafeDB	Generic	System	Global	managed-server1	Running	0

6. In the **Domain Structure** box, select **Deployments**.

7. In the **Deployments** table, there should be one row. The name should be the same value as the `Application` value in your *appmodel.yaml* file. Select the name.

8. Select the **Testing** tab.

9. Select **weblogic-cafe**.

10. In the **Settings for weblogic-cafe** panel, select the **Testing** tab.

11. Expand the + icon next to **weblogic-cafe**. Your screen should look similar to the following example. In particular, you should see values similar to `http://sample-domain1-managed-server1:8001/weblogic-cafe/index.xhtml` in the **Test Point** column.

Showing 1 to 1 of 1 Previous | Next

Name	Test Point	Comments
weblogic-cafe		
/weblogic-cafe/rest	/application.wadl	WADL page on server managed-server2
/weblogic-cafe/rest	/application.wadl	WADL page on server managed-server1
default	http://sample-domain1-managed-server2:8001/weblogic-cafe	Default url on server managed-server2
default	http://sample-domain1-managed-server1:8001/weblogic-cafe	Default url on server managed-server1
index.xhtml	http://sample-domain1-managed-server2:8001/weblogic-cafe/index.xhtml	Welcome file index.xhtml on server managed-server2
index.xhtml	http://sample-domain1-managed-server1:8001/weblogic-cafe/index.xhtml	Welcome file index.xhtml on server managed-server1

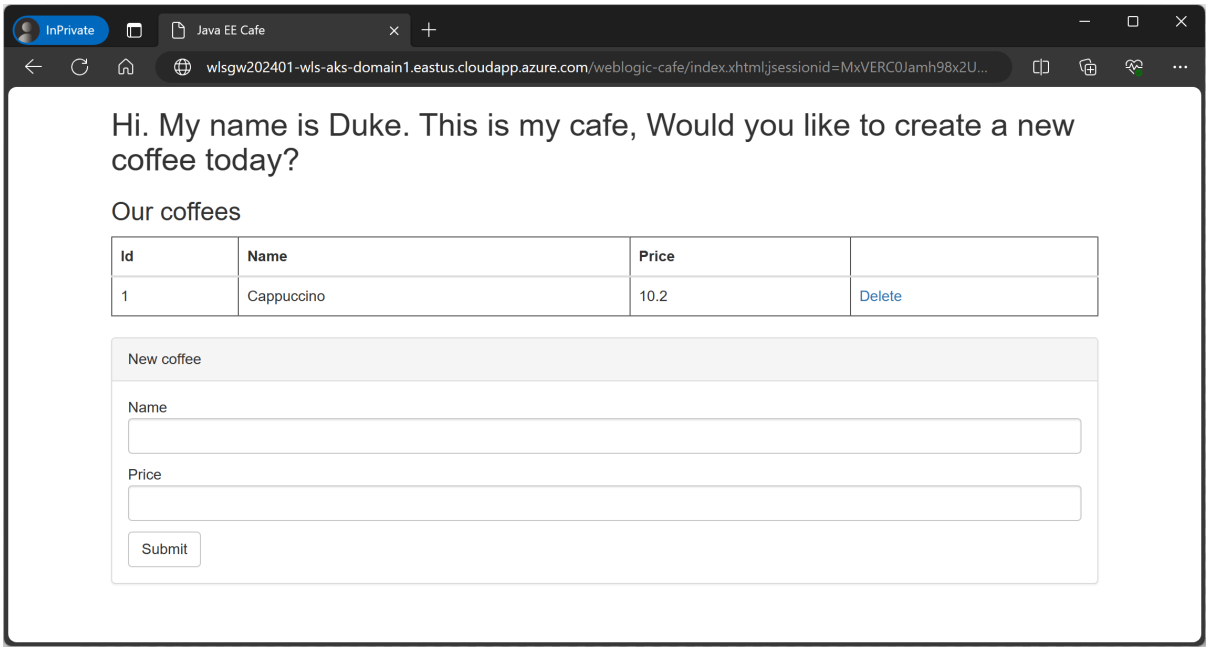
Showing 1 to 1 of 1 Previous | Next

> **Note**
>
> The hyperlinks in the **Test Point** column are not selectable because we did not configure the admin console with the external URL on which it is running. This article shows the WebLogic Server admin console merely by way of demonstration. Don't use the WebLogic Server admin console for any durable configuration changes when running WebLogic Server on AKS. The cloud-native design of WebLogic Server on AKS requires that any durable configuration must be represented in the initial docker images or applied to the running AKS cluster using CI/CD techniques such as updating the model, as described in the Oracle documentation.

12. Understand the `context-path` value of the sample app you deployed. If you deployed the recommended sample app, the `context-path` is `weblogic-cafe`.

13. Construct a fully qualified URL for the sample app by appending the `context-path` to the **clusterExternalUrl** value. If you deployed the recommended sample app, the fully qualified URL should be something like `http://wlsgw202401-wls-aks-domain1.eastus.cloudapp.azure.com/weblogic-cafe/`.

14. Paste the fully qualified URL in an Internet-connected web browser. If you deployed the recommended sample app, you should see results similar to the following screenshot:



Clean up resources

To avoid Azure charges, you should clean up unnecessary resources. When you no longer need the cluster, use the az group delete command. The following command removes the resource group, container service, container registry, database, and all related resources:

Azure CLI

```
az group delete --name <resource-group-name> --yes --no-wait
az group delete --name <db-resource-group-name> --yes --no-wait
```

Next steps

Learn more about running WebLogic Server on AKS or virtual machines by following these links:

- WebLogic Server on AKS
- WebLogic Server on virtual machines

For more information about the Oracle WebLogic offers at Azure Marketplace, see Oracle WebLogic Server on Azure. These offers are all *Bring-Your-Own-License*. They assume that you already have the appropriate licenses with Oracle and are properly licensed to run offers in Azure.

Set up Oracle ASM on an Azure Linux virtual machine

Article • 08/22/2024

Applies to: ✔ Linux VMs

Azure virtual machines provide a fully configurable and flexible computing environment. This tutorial covers basic Azure virtual machine deployment combined with the installation and configuration of Oracle Automatic Storage Management (ASM). You learn how to:

- ✔ Create and connect to an Oracle Database VM
- ✔ Install and configure Oracle Automatic Storage Management
- ✔ Install and configure Oracle Grid infrastructure
- ✔ Initialize an Oracle ASM installation
- ✔ Create an Oracle DB managed by ASM

For an overview of the value proposition of ASM, see the documentation at Oracle.

If you choose to install and use the CLI locally, this tutorial requires that you are running the Azure CLI version 2.0.4 or later. Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI.

Prepare the environment

This lab uses two VMs on Azure: **asmXServer** runs X Windows server used to run grid setup and **asmVM** hosts the Oracle Database and ASM installation. The Marketplace images used to create these virtual machines are

- asmVM: **Oracle:oracle-database-19-3:oracle-database-19-0904:19.3.1**
- asmXServer: **MicrosoftWindowsDesktop:Windows-10:win10-22h2-pro-g2:19045.2604.230207**

You also need to be familiar with Unix editor **vi** and have a basic understanding of X Server.

Sign in to Azure

1. Open your preferred shell on Windows, Linux or Azure Shell.

2. Sign in to your Azure subscription with the az login command. Then follow the on-screen directions.

Azure CLI

```
$ az login
```

3. Ensure you are connected to the correct subscription by verifying subscription name and/or ID.

Azure CLI

```
$ az account show
```

Output

```
{
  "environmentName": "XXXXX",
  "homeTenantId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "id": "<SUBSCRIPTION_ID>",
  "isDefault": true,
  "managedByTenants": [],
  "name": "<SUBSCRIPTION_NAME>",
  "state": "Enabled",
  "tenantId": XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "user": {
    "cloudShellID": true,
    "name": "aaaaa@bbbbb.com",
    "type": "user"
  }
}
```

Generate authentication keys

We use keyfile based authentication with ssh to connect to the Oracle Database VM. Ensure that you have your private (named `id_rsa`) and public (named `id_rsa.pub`) key files are created on your shell.

Location of key files depends on your source system.

Windows: %USERPROFILE%.ssh Linux: ~/.ssh

If they don't exist you can create a new keyfile pair.

Bash

```
$ ssh-keygen -m PEM -t rsa -b 4096
```

The .ssh directory and key files are created. For more information, refer to Create and manage SSH keys for authentication to a Linux VM in Azure

Create a resource group

To create a resource group, use the az group create command. An Azure resource group is a logical container in which Azure resources are deployed and managed.

Azure CLI

```
$ az group create --name ASMOnAzureLab --location westus
```

Create and configure network

Create virtual network

Use following command to create the virtual network that hosts resources we create in this lab.

Azure CLI

```
$ az network vnet create \
  --name asmVnet \
  --resource-group ASMOnAzureLab \
  --address-prefixes "10.0.0.0/16" \
  --subnet-name asmSubnet1 \
  --subnet-prefixes "10.0.0.0/24"
```

Create a Network Security Group (NSG)

1. Create network security group (NSG) to lock down your virtual network.

   Azure CLI

   ```
   $ az network nsg create \
     --resource-group ASMOnAzureLab \
     --name asmVnetNSG
   ```

2. Create NSG rule to allow communication within virtual network.

Azure CLI

```
$ az network nsg rule create  --resource-group ASMOnAzureLab --nsg-name
asmVnetNSG \
    --name asmAllowVnet \
    --protocol '*' --direction inbound --priority 3400 \
    --source-address-prefix 'VirtualNetwork' --source-port-range '*' \
    --destination-address-prefix 'VirtualNetwork' --destination-port-
range '*' --access allow
```

3. Create NSG rule to deny all inbound connections

Azure CLI

```
$ az network nsg rule create \
  --resource-group ASMOnAzureLab \
  --nsg-name asmVnetNSG \
  --name asmDenyAllInBound \
  --protocol '*' --direction inbound --priority 3500 \
  --source-address-prefix '*' --source-port-range '*' \
  --destination-address-prefix '*' --destination-port-range '*' --
access deny
```

4. Assign NSG to Subnet where we host our servers.

Azure CLI

```
$ az network vnet subnet update --resource-group ASMOnAzureLab --vnet-
name asmVNet --name asmSubnet1 --network-security-group asmVnetNSG
```

Create Bastion Network

1. Create Bastion subnet. Name of the subnet must be **AzureBastionSubnet**

Azure CLI

```
$ az network vnet subnet create  \
    --resource-group ASMOnAzureLab \
    --name AzureBastionSubnet \
    --vnet-name asmVnet \
    --address-prefixes 10.0.1.0/24
```

2. Create public IP for Bastion

Azure CLI

```
$ az network public-ip create \
    --resource-group ASMOnAzureLab \
    --name asmBastionIP \
    --sku Standard
```

3. Create Azure Bastion resource. It takes about 10 minutes for the resource to deploy.

Azure CLI

```
$ az network bastion create \
    --resource-group ASMOnAzureLab \
    --name asmBastion \
    --public-ip-address asmBastionIP \
    --vnet-name asmVnet \
    --sku Standard \
    --enable-tunneling \
    --enable-ip-connect true
```

Create X Server VM (asmXServer)

Replace your password and run following command to create a Windows workstation VM where we deploy X Server.

Azure CLI

```
$ az vm create \
    --resource-group ASMOnAzureLab \
    --name asmXServer \
    --image MicrosoftWindowsDesktop:Windows-10:win10-22h2-pro-
g2:19045.2604.230207 \
    --size Standard_DS1_v2  \
    --vnet-name asmVnet \
    --subnet asmSubnet1 \
    --public-ip-sku Standard \
    --nsg "" \
    --data-disk-delete-option Delete \
    --os-disk-delete-option Delete \
    --nic-delete-option Delete \
    --admin-username azureuser \
    --admin-password <ENTER_YOUR_PASSWORD_HERE>
```

Connect to asmXServer

Connect to **asmXServer** using Bastion.

1. Navigate to **asmXServer** from Azure portal.
2. Go to **Overview** in the left blade
3. Select **Connect** > **Bastion** on the menu at the top
4. Select Bastion tab
5. Click **Use Bastion**

Prepare asmXServer to run X Server

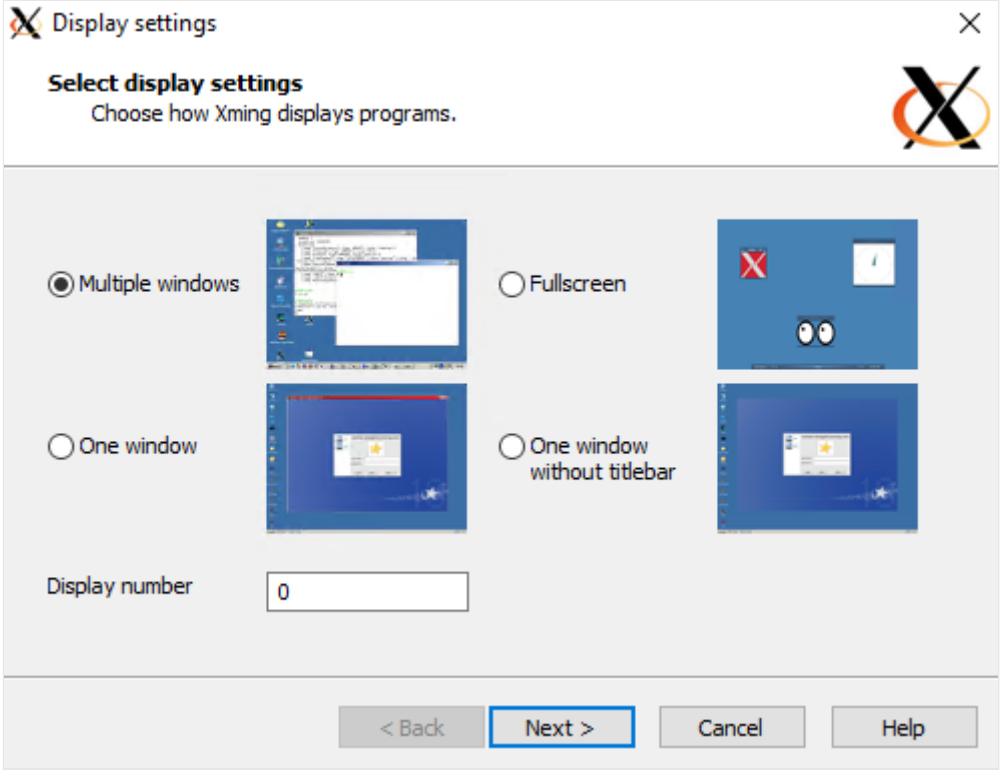
X Server is required for later steps of this lab. Perform following steps to install and start X Server.

1. Download Xming X Server for Windows to **ggXServer** and install with all default options.

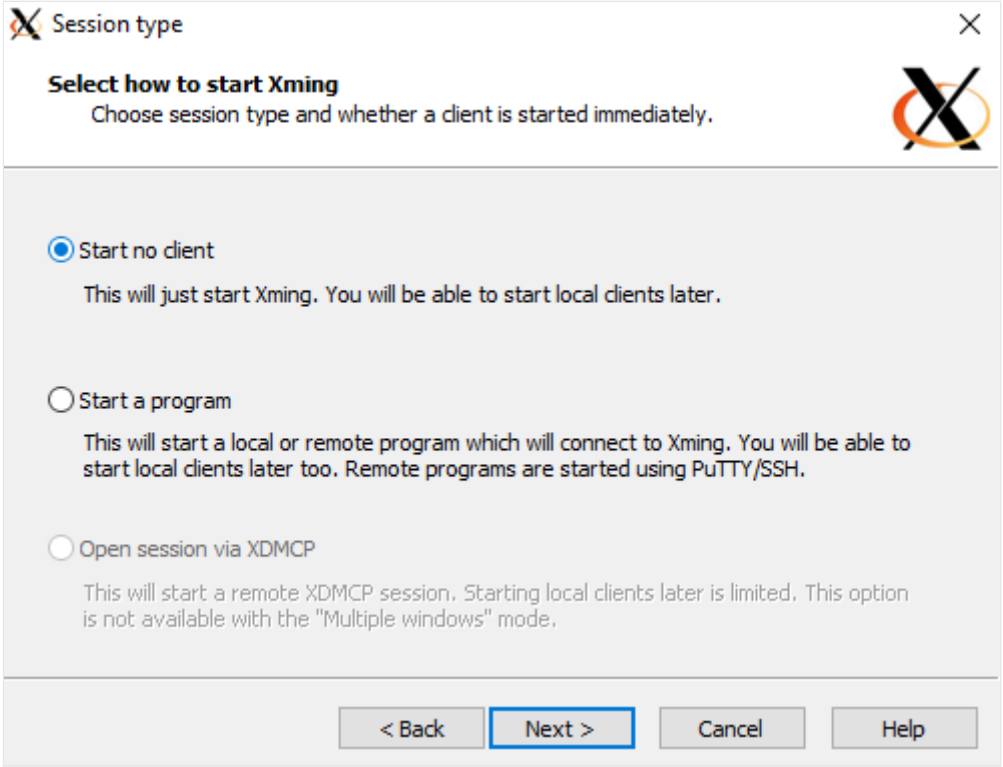
2. Ensure that you did not select **Launch** at the end of installation

3. Launch "XLAUNCH" application from start menu.

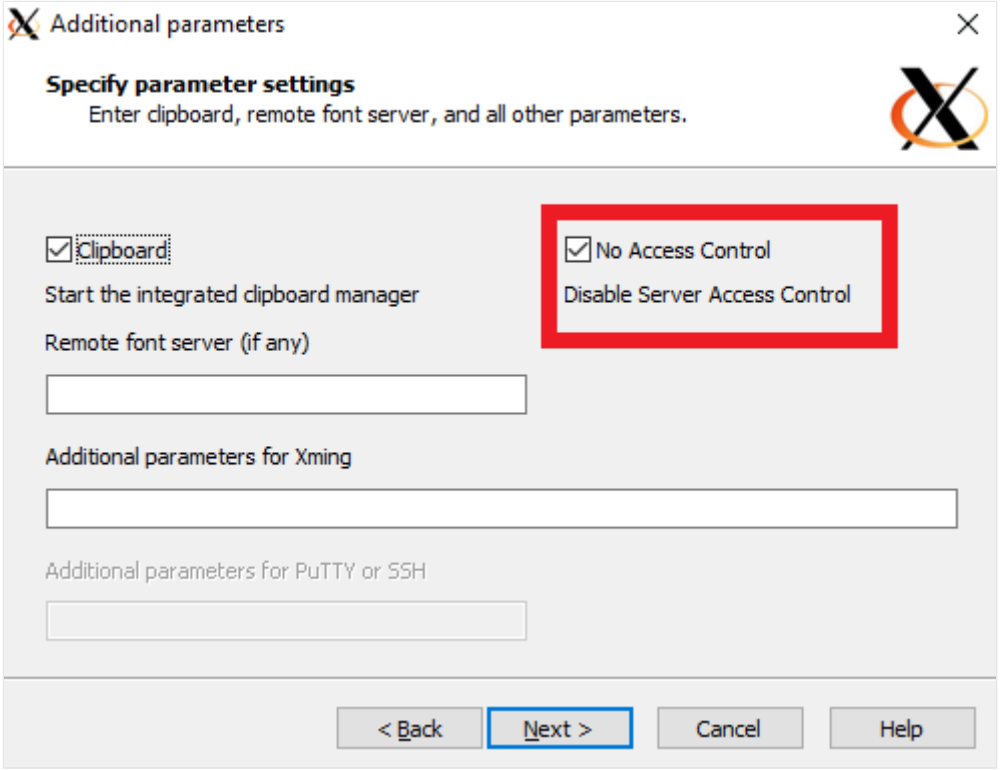
4. Select **Multiple Windows**



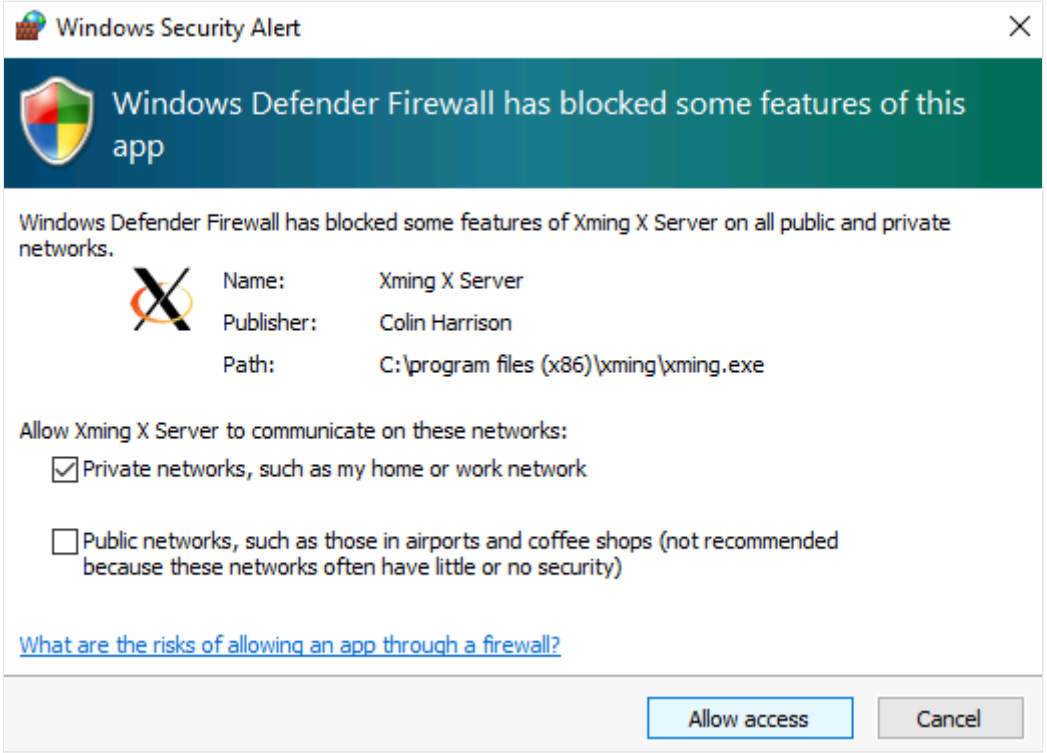
5. Select **Start no client**



6. Select **No access control**



7. Select **Allow Access** to allow X Server through Windows Firewall



If you restart your **asmXServer** VM, follow steps 2-6 above to restart X Server application.

Create Oracle Database VM

For this lab, we create a virtual machine `asmVM` from Oracle Database 19c image. Run following to create **asmVM** with multiple data disks attached. If they do not already exist in the default key location, this command also creates SSH keys. To use a specific set of keys, use the `--ssh-key-value` option. If you have already created your SSH keys in Generate authentication keys section, those keys will be used.

When creating a new virtual machine `size` parameter indicates the size and type of virtual machine created. Depending on the Azure region you selected to create virtual machine and your subscription settings, some virtual machine sizes and types may not be available for you to use. Following command uses minimum required size for this lab `Standard_D4_v5`. If you want to change specs of virtual machine, select one of the available sizes from Azure VM Sizes. For test purposes, you may choose from General Purpose (D-Series) virtual machine types. For production or pilot deployments, Memory Optimized (E-Series and M-Series) are more suitable.

Azure CLI

```
az vm create --resource-group ASMOnAzureLab \
   --name asmVM \
   --image Oracle:oracle-database-19-3:oracle-database-19-0904:19.3.1 \
   --size Standard_D4_v5 \
   --generate-ssh-keys \
```

```
    --os-disk-size-gb 30 \
    --data-disk-sizes-gb 20 40 40 \
    --admin-username azureuser \
    --vnet-name asmVnet \
    --subnet asmSubnet1 \
    --public-ip-sku Basic \
    --nsg ""
```

Connect to asmVM

Connect to **asmVM** using Bastion.

1. Navigate to **asmVM** from Azure portal.
2. Go to **Overview** in the left blade
3. Select **Connect** > **Bastion** on the menu at the top
4. Select Bastion tab
5. Click **Use Bastion**

Create swap file

This lab requires a swap file on the lab virtual machine. Complete following steps to create the swap file.

Prepare disk and mount point

1. When we created the virtual machine (asmVM) earlier, we included a 20GB data disk to place swap file. Run following command to find out the name for this 20GB disk. It is **/dev/sdb** most of the time but in case it comes up different make sure you note the name for 20G disk and use if for following steps. Similarly we use the names of 40G disks (which are named **/dev/sdc** and **/dev/sdd** in the following output) later on.

   Bash

   ```
   $ sudo su -
   $ lsblk
   ```

   Output

   ```
   NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
   sdd       8:48   0   40G  0 disk             ====> Data disk 2 (40GB)
   sdb       8:16   0   20G  0 disk             ====> Swap file disk
   (20GB)
   sr0      11:0    1  628K  0 rom
   ```

```
fd0       2:0    1    4K  0 disk
sdc       8:32   0   40G  0 disk             ====> Data disk 1 (40GB)
sda       8:0    0   30G  0 disk
├─sda2    8:2    0   29G  0 part /
├─sda14   8:14   0    4M  0 part
├─sda15   8:15   0  495M  0 part /boot/efi
└─sda1    8:1    0  500M  0 part /boot
```

2. Run following command to create the partition on the swap file disk, modify disk name (/dev/sdb) if necessary.

Bash

```
$ parted /dev/sdb --script mklabel gpt mkpart xfspart xfs 0% 100%
```

3. Check the name of the partition created. Below it is created as **sdb1**

Bash

```
$ lsblk
```

Output

```
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sdd       8:48   0   40G  0 disk
sdb       8:16   0   20G  0 disk
└─sdb1    8:17   0   20G  0 part             ====> Newly created
partition
sr0      11:0    1  628K  0 rom
fd0       2:0    1    4K  0 disk
sdc       8:32   0   40G  0 disk
sda       8:0    0   30G  0 disk
├─sda2    8:2    0   29G  0 part /
├─sda14   8:14   0    4M  0 part
├─sda15   8:15   0  495M  0 part /boot/efi
└─sda1    8:1    0  500M  0 part /boot
```

4. Run following commands to initialize file system (xfs) and mount the drive as **/swap**

Bash

```
$ mkfs.xfs /dev/sdb1
$ partprobe /dev/sdb1
$ mkdir /swap
$ mount /dev/sdb1 /swap
```

5. Run following command

```bash
$ blkid
```

In the output, you see a line for swap disk partition **/dev/sdb1**, note down the **UUID**.

```output
/dev/sdb1: UUID="00000000-0000-0000-0000-000000000000" TYPE="xfs"
PARTLABEL="xfspart" PARTUUID="...."
```

6. Paste UUID from previous step into the following command and run it. This command ensures proper mounting of drive every time system reboots.

```bash
$ echo "UUID=00000000-0000-0000-0000-000000000000   /swap   xfs
defaults,nofail   1   2" >> /etc/fstab
```

Configure swap file

1. Create and allocate the swap file (16GB). This command takes a couple of minutes to run.

```bash
$ dd if=/dev/zero of=/swap/swapfile bs=1M count=16384
```

2. Modify permissions and assign the swap file

```bash
$ chmod 600 /swap/swapfile
$ mkswap /swap/swapfile
$ swapon /swap/swapfile
```

3. Verify swap file is created

```bash
```

```
$ cat /proc/swaps
```

Output

```
Filename        Type    Size        Used    Priority
/swap/swapfile  file    16777212    0        -2
```

4. Ensure swap file setting is retained across reboots

Bash

```bash
$ echo "/swap/swapfile   none  swap  sw  0 0" >> /etc/fstab
```

Install Oracle ASM

To install Oracle ASM, complete the following steps.

For more information about installing Oracle ASM, see Oracle ASMLib Downloads for Oracle Linux 7.

1. You need to login as root in order to continue with ASM installation, if you have not already done so

Bash

```bash
$ sudo su -
```

2. Run these additional commands to install Oracle ASM components:

Bash

```bash
$ yum list | grep oracleasm
```

Output of the command looks like

Output

```
kmod-oracleasm.x86_64                    2.0.8-28.0.1.el7
ol7_latest
oracleasm-support.x86_64                 2.1.11-2.el7
ol7_latest
```

Continue installation by running following commands

```bash
$ yum -y install kmod-oracleasm.x86_64
$ yum -y install oracleasm-support.x86_64
$ wget https://download.oracle.com/otn_software/asmlib/oracleasmlib-
2.0.15-1.el7.x86_64.rpm
$ yum -y install oracleasmlib-2.0.15-1.el7.x86_64.rpm
$ rm -f oracleasmlib-2.0.15-1.el7.x86_64.rpm
```

3. Verify that Oracle ASM is installed:

```bash
$ rpm -qa |grep oracleasm
```

The output of this command should list the following components:

```output
oracleasm-support-2.1.11-2.el7.x86_64
oracleasmlib-2.0.15-1.el7.x86_64
kmod-oracleasm-2.0.8-28.0.1.el7.x86_64
```

4. ASM requires specific users and roles in order to function correctly. The following commands create the pre-requisite user accounts and groups.

```bash
$ groupadd -g 54345 asmadmin
$ groupadd -g 54346 asmdba
$ groupadd -g 54347 asmoper
$ usermod -a -g oinstall -G oinstall,dba,asmdba,asmadmin,asmoper oracle
```

5. Verify users and groups were created correctly.

```bash
$ grep oracle /etc/group
```

The output of this command should list the following users and groups.

```output
```

```
oinstall:x:54321:oracle
dba:x:54322:oracle
oper:x:54323:oracle
backupdba:x:54324:oracle
dgdba:x:54325:oracle
kmdba:x:54326:oracle
racdba:x:54330:oracle
asmadmin:x:54345:oracle
asmdba:x:54346:oracle
asmoper:x:54347:oracle
```

6. Create the app folder change the owner.

Bash

```bash
$ mkdir /u01/app/grid
$ chown oracle:oinstall /u01/app/grid
```

Set up Oracle ASM

For this tutorial, the default user is **oracle** and the default group is **asmadmin**. Ensure that the **oracle** user is part of the **asmadmin** group.

Bash

```bash
$ groups oracle
```

The output of command should look like

Output

```
oracle : oinstall dba oper backupdba dgdba kmdba racdba asmadmin asmdba
asmoper
```

To set up Oracle ASM, complete the following steps:

1. Set up the Oracle ASM library driver using the following command and providing following answers for prompts.

Bash

```bash
$ /usr/sbin/oracleasm configure -i
```

The output of this command should look similar to the following, stopping with prompts to be answered.

Output

```
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM library
driver. The following questions will determine whether the driver is
loaded on boot and what permissions it will have. The current values
will be shown in brackets ('[]'). Hitting <ENTER> without typing an
answer will keep that current value. Ctrl-C will abort.

Default user to own the driver interface []: oracle
Default group to own the driver interface []: asmadmin
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done
```

Note

The `/usr/sbin/oracleasm configure -i` command asks for the user and group that default to owning the ASM driver access point. The database will be running as the `oracle` user and the `asmadmin` group. By selecting **Start Oracle ASM library driver on boot = 'y'**, the system will always load the module and mount the filesystem on boot. By selecting **Scan for Oracle ASM disks on boot = 'y'**, the system will always scan the Oracle ASM disks on boot. The last two configurations are very important, otherwise, you will run into disk reboot problems.

2. View the disk configuration:

Bash

```
$ cat /proc/partitions
```

The output of this command should look similar to the following listing of available disks

Output

```
major minor  #blocks  name
   8       16   20971520 sdb
   8       17   20969472 sdb1
   8       32   41943040 sdc
```

```
   8       48   41943040 sdd
   8        0   31457280 sda
   8        1     512000 sda1
   8        2   30431232 sda2
   8       14       4096 sda14
   8       15     506880 sda15
  11        0        628 sr0
   2        0          4 fd0
```

3. Format disk **/dev/sdc** by running the following command and answering the prompts with:
   a. **n** for new partition
   b. **p** for primary partition
   c. **1** to select the first partition
   d. press **enter** for the default first sector
   e. press **enter** for the default last sector
   f. press **w** to write the changes to the partition table

Bash

```
$ fdisk /dev/sdc
```

The output for the `fdisk` command should look like the following output:

Output

```
 Welcome to fdisk (util-linux 2.23.2).

 Changes will remain in memory only, until you decide to write them.
 Be careful before using the write command.

 Device does not contain a recognized partition table
 Building a new DOS disklabel with disk identifier 0x947f0a91.

 The device presents a logical sector size that is smaller than
 the physical sector size. Aligning to a physical sector (or optimal
 I/O) size boundary is recommended, or performance may be impacted.

 Command (m for help): n
 Partition type:
    p   primary (0 primary, 0 extended, 4 free)
    e   extended
 Select (default p): p
 Partition number (1-4, default 1): 1
 First sector (2048-104857599, default 2048):
 Using default value 2048
 Last sector, +sectors or +size{K,M,G} (2048-104857599, default
104857599):
 Using default value 104857599
```

```
Partition 1 of type Linux and of size 50 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

4. Repeat the preceding `fdisk` command for `/dev/sdd`.

Bash

```
$ fdisk /dev/sdd
```

5. Check the disk configuration:

Bash

```
$ cat /proc/partitions
```

The output of the command should look like the following output:

Output

```
major minor  #blocks  name
    8       16   20971520 sdb
    8       17   20969472 sdb1
    8       32   41943040 sdc
    8       33   41942016 sdc1
    8       48   41943040 sdd
    8       49   41942016 sdd1
    8        0   31457280 sda
    8        1     512000 sda1
    8        2   30431232 sda2
    8       14       4096 sda14
    8       15     506880 sda15
   11        0        628 sr0
    2        0          4 fd0
```

Note

Note that, in the following configuration, please use the exact commands as this document shows.

6. Check the Oracle ASM service status and start the Oracle ASM service:

Bash

```
$ oracleasm status
```

Output

```
Checking if ASM is loaded: no
Checking if /dev/oracleasm is mounted: no
```

Bash

```
$ oracleasm init
```

Output

```
Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
```

7. Create Oracle ASM disks

   a. Create first disk

   Bash

   ```
   $ oracleasm createdisk VOL1 /dev/sdc1
   ```

   b. The output of command should look like

   Output

   ```
   Writing disk header: done
   Instantiating disk: done
   ```

   c. Create remaining disks

   Bash

   ```
   $ oracleasm createdisk VOL2  /dev/sdd1
   ```

ⓘ **Note**

Disks are marked for ASMLib using a process described in ASMLib Installation. ASMLib learns what disk are marked during a process called disk scanning. ASMLib runs this scan every time it starts up. The system administrator can also force a scan via the `oracleasm scandisks` command. ASMLib examines each disk in the system. It checks if the disk has been marked for ASMLib. Any disk that has been marked will be made available to ASMLib. You can visit documents Configuring Storage Device Path Persistence Using Oracle ASMLIB and Configuring Oracle ASMLib on Multipath Disks for more information.

8. List Oracle ASM disks

Bash

```
$ oracleasm scandisks
$ oracleasm listdisks
```

The output of the command should list off the following Oracle ASM disks:

Output

```
 VOL1
 VOL2
```

9. Change passwords for the root and oracle users. **Make note of these new passwords** as you are using them later during the installation.

Bash

```
$ passwd oracle
$ passwd root
```

10. Change folder permissions

Bash

```
$ chmod -R 775 /opt
$ chown oracle:oinstall /opt
$ chown oracle:oinstall /dev/sdc1
$ chown oracle:oinstall /dev/sdd1
$ chmod 600 /dev/sdc1
$ chmod 600 /dev/sdd1
```

Download and Prepare Oracle Grid Infrastructure

To download and prepare the Oracle Grid Infrastructure software, complete the following steps:

1. Download Oracle Grid Infrastructure from the Oracle ASM download page. Your download location should have Azure CLI installed because we copy these files to asmVM using Bastion. Because it uses a tunnel, this step will not work over Azure Cloud Shell, and it needs to be run on a workstation.

   Under the download titled **Oracle Database 19c Grid Infrastructure (19.3) for Linux x86-64**, download the .zip file.

2. After you download the .zip file to your client computer, you can use Secure Copy Protocol (SCP) to copy the files to your VM. Make sure that `scp` command points to correct path of .zip file.

   a. Login and ensure you are using the correct subscription as necessary as described in Sign in to Azure

   b. Open the tunnel to your target VM using the following PowerShell command

   PowerShell

   ```
   $asmVMid=$(az vm show --resource-group ASMOnAzureLab --name asmVM --query 'id' --output tsv)

   az network bastion tunnel --name asmBastion --resource-group ASMOnAzureLab --target-resource-id $asmVMid --resource-port 22 --port 57500
   ```

   c. Leave the first command prompt running and open a second command prompt to connect to your target VM through the tunnel. In this second command prompt window, you can upload files from your local machine to your target VM using the following command. Note that the correct `id_rsa` keyfile to access asmVM must reside in `.ssh` directory or you can point to a different key file using `-i` parameter to `scp` command.

   PowerShell

   ```
   scp -P 57500 "LINUX.X64_193000_grid_home.zip"  azureuser@127.0.0.1:.
   ```

3. When upload is complete SSH back into your **asmVM** in Azure using Bastion in order to move the .zip files into the **/opt** folder and change the owner of the file.

```bash
$ sudo su -
$ mv /home/azureuser/*.zip /opt
$ cd /opt
$ chown oracle:oinstall LINUX.X64_193000_grid_home.zip
```

4. Unzip the files. (Install the Linux unzip tool if it's not already installed.)

```bash
$ yum install unzip
$ unzip LINUX.X64_193000_grid_home.zip -d grid
```

5. Change permission

```bash
$ chown -R oracle:oinstall /opt/grid
```

6. Cleanup

```bash
$ rm -f LINUX.X64_193000_grid_home.zip
```

7. Exit **root**

```bash
$ exit
```

Install Oracle Grid Infrastructure

To install Oracle Grid Infrastructure, complete the following steps:

1. Sign in as **oracle**. (You should be able to sign in without being prompted for a password.)

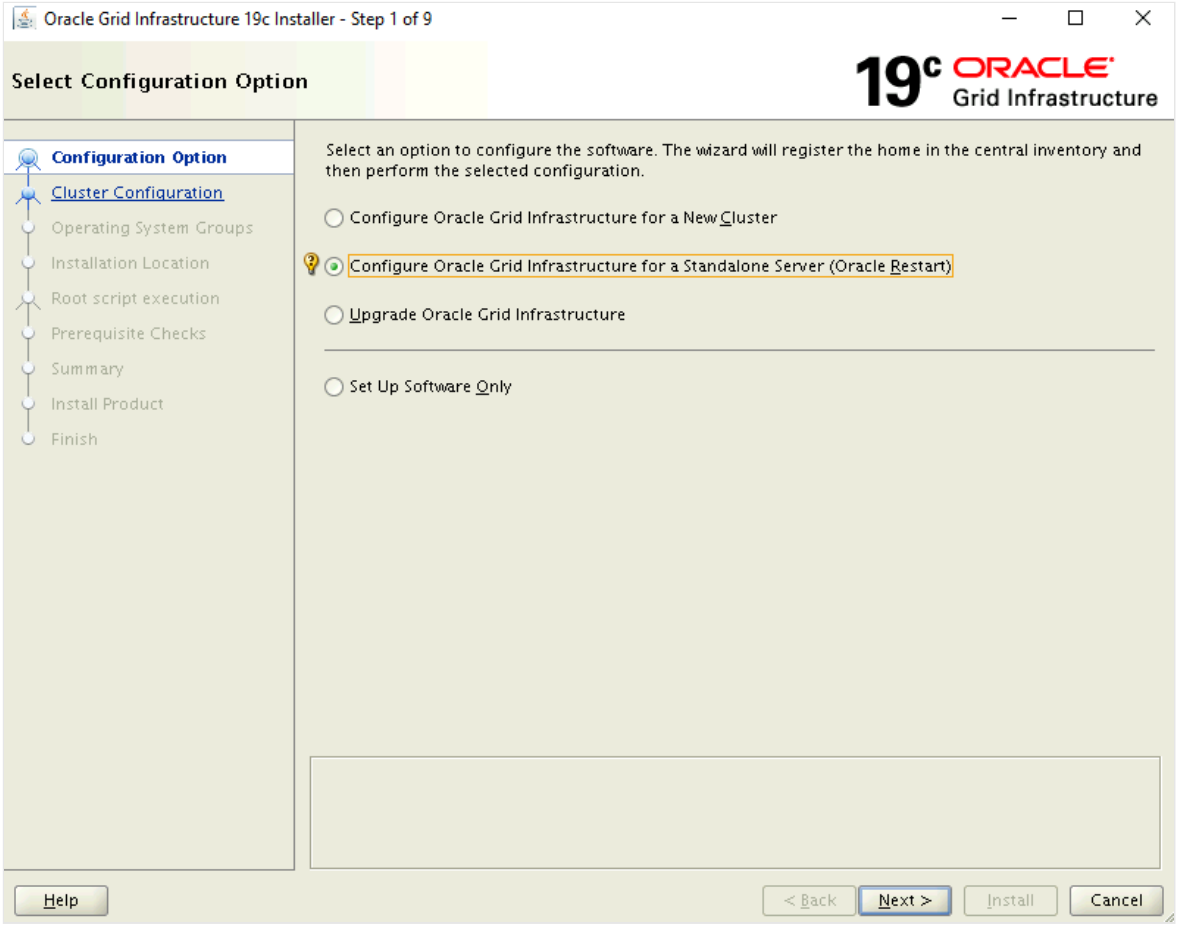
> ⓘ **Note**

Make sure you have started X Server before you begin the installation.

```bash
$ sudo su - oracle
$ export DISPLAY=10.0.0.4:0.0
$ cd /opt/grid
$ ./gridSetup.sh
```

Oracle Grid Infrastructure 19c Installer opens on **asmXServer** VM. (It might take a few minutes for the installer to start.)

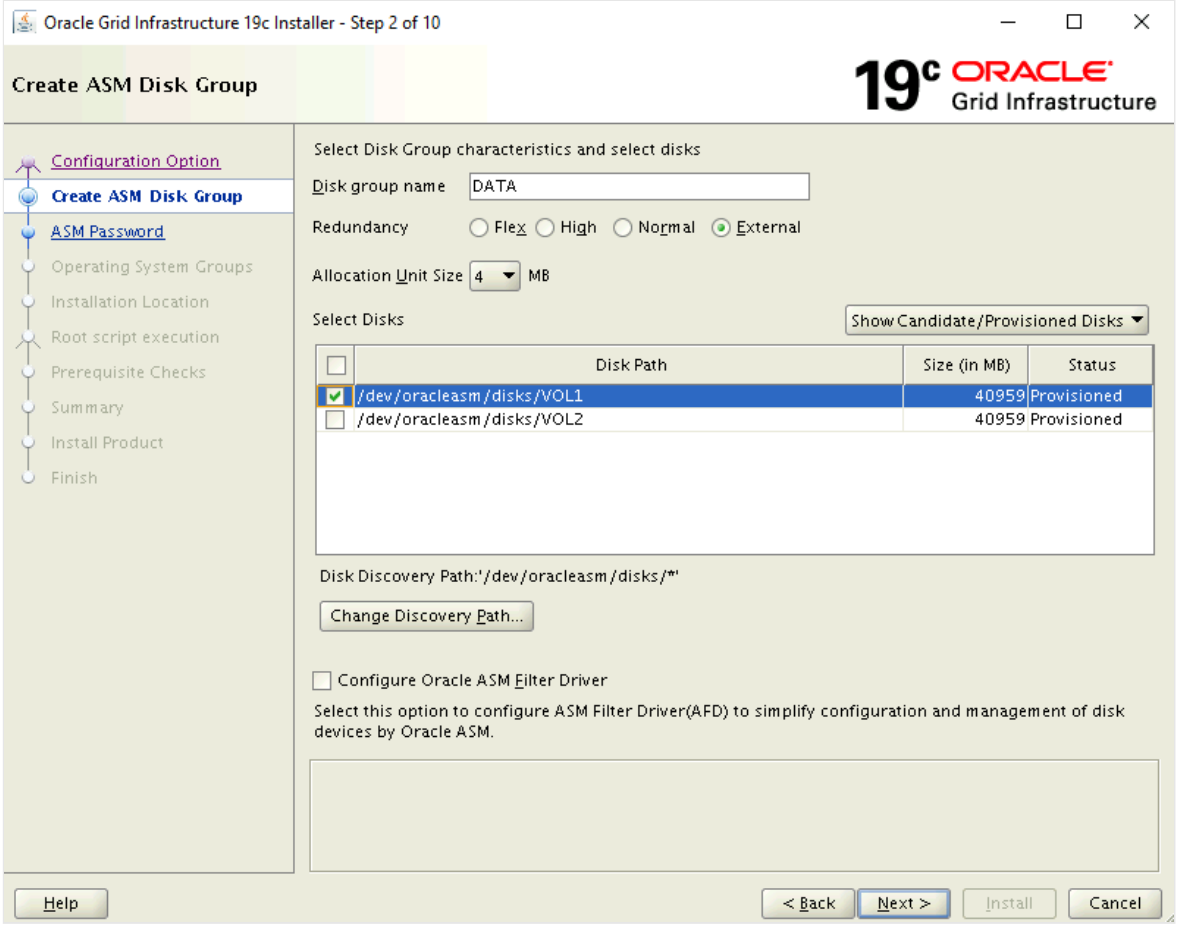
2. On the **Select Configuration Option** page, select **Configure Oracle Grid Infrastructure for a Standalone Server (Oracle Restart)**.



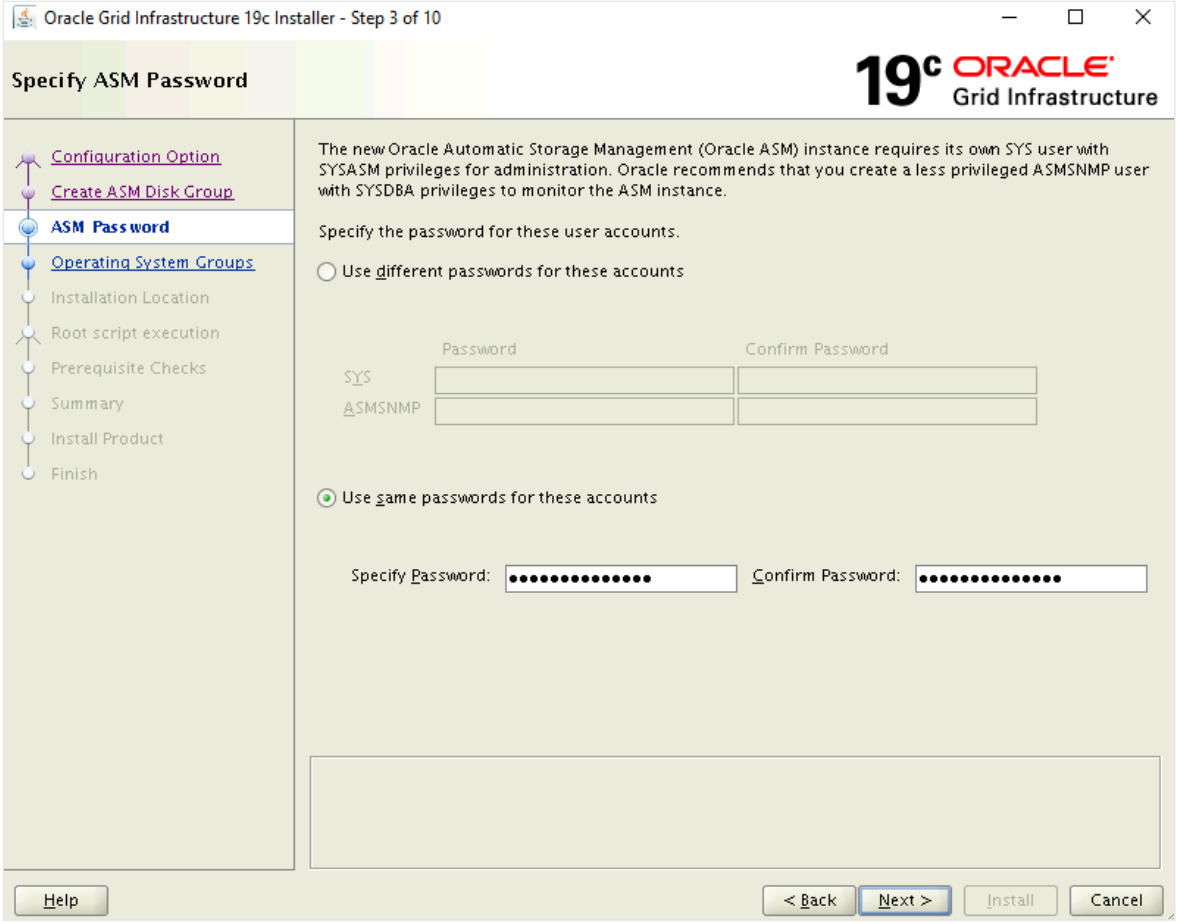
3. On the **Create ASM Disk Group** page:

- Click on **Change Discovery Path**
- Update the discovery path to be **/dev/oracleasm/disks/***
- Enter a name for the disk group **DATA**
- Under **Redundancy**, select **External**.
- Under **Allocation Unit Size**, select **4**.
- Under **Select Disks**, select **/dev/oracleasm/disks/VOL1**.

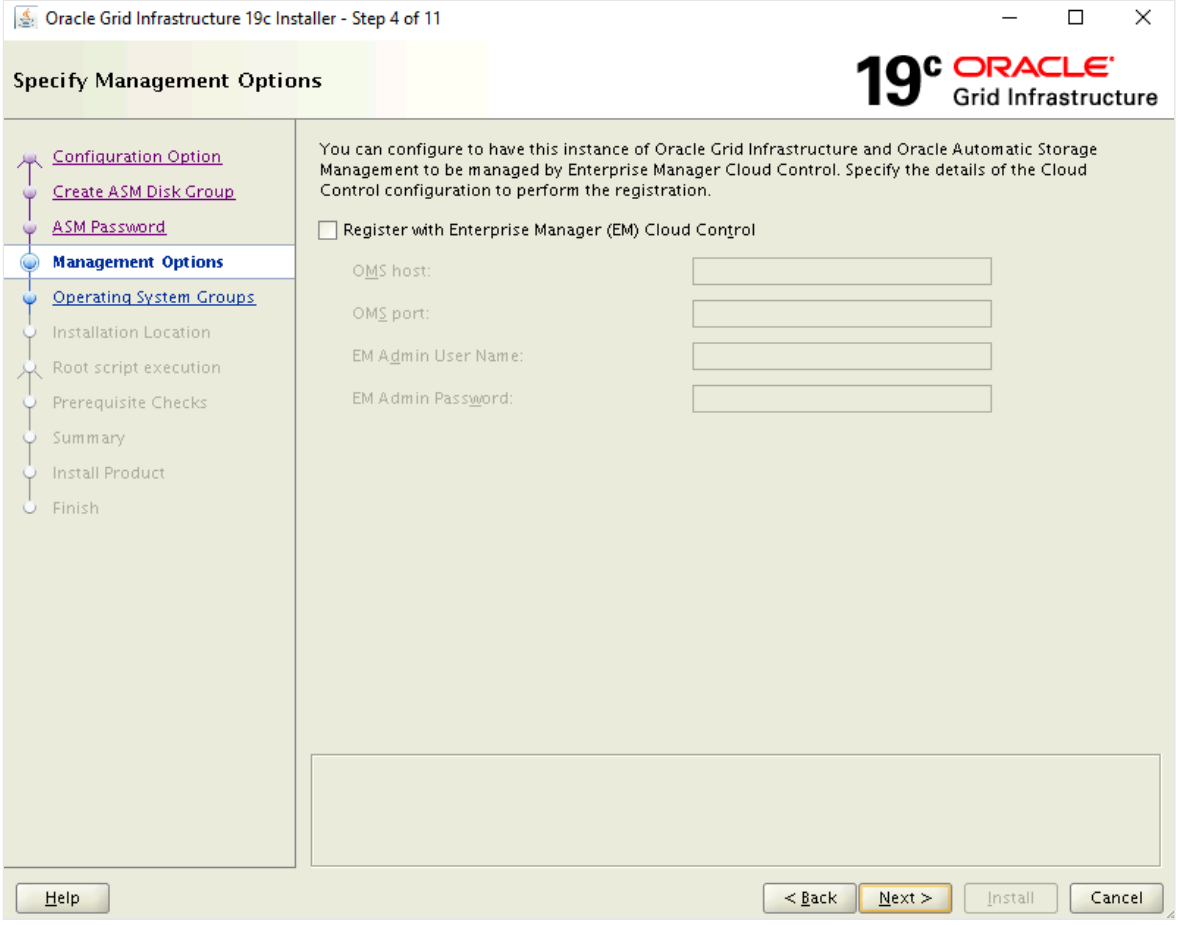
- Click **Next**.



4. On the **Specify ASM Password** page, select the **Use same passwords for these accounts** option, and enter a password.



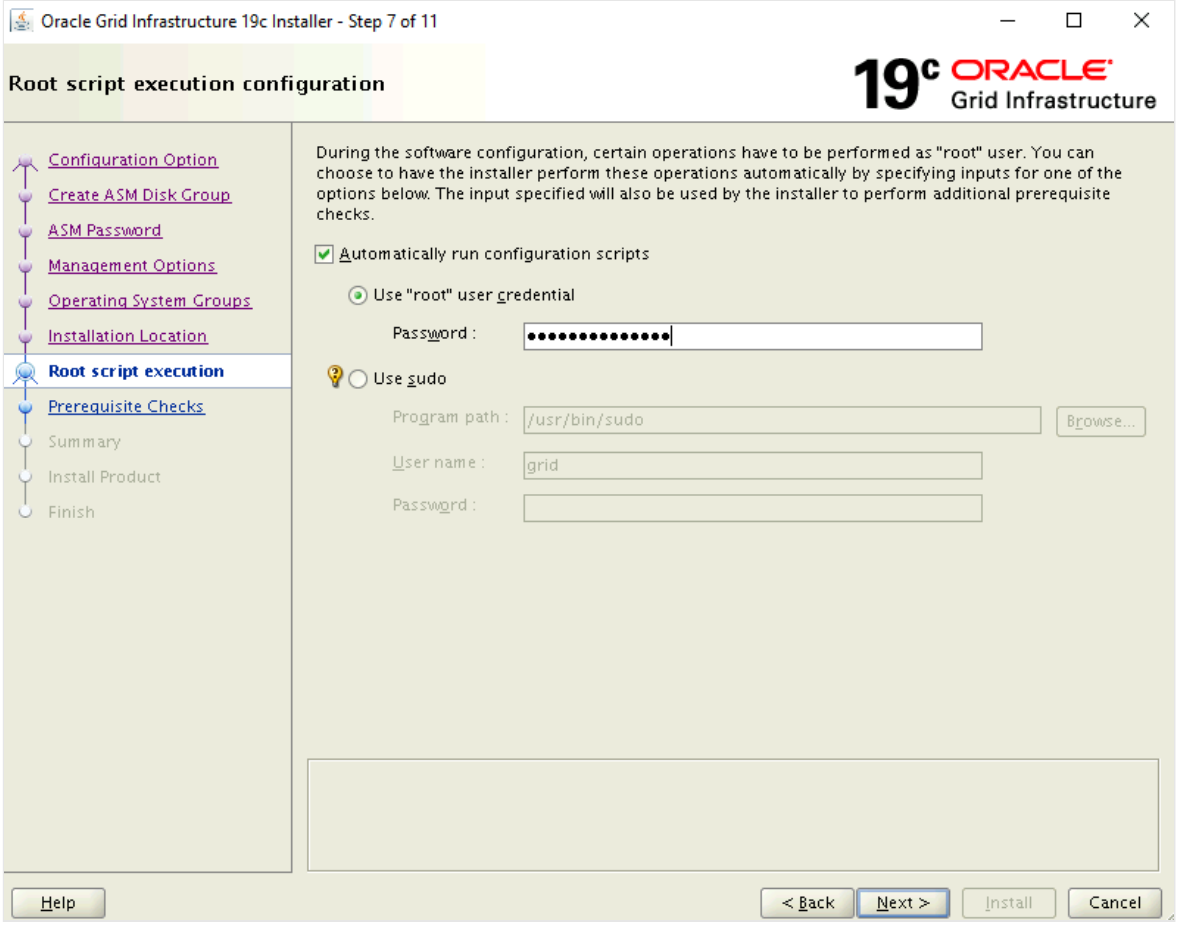
5. On the **Specify Management Options** page, make sure the option to configure EM Cloud Control is unselected. Click **Next** to continue.



6. On the **Privileged Operating System Groups** page, use the default settings. Click **Next** to continue.

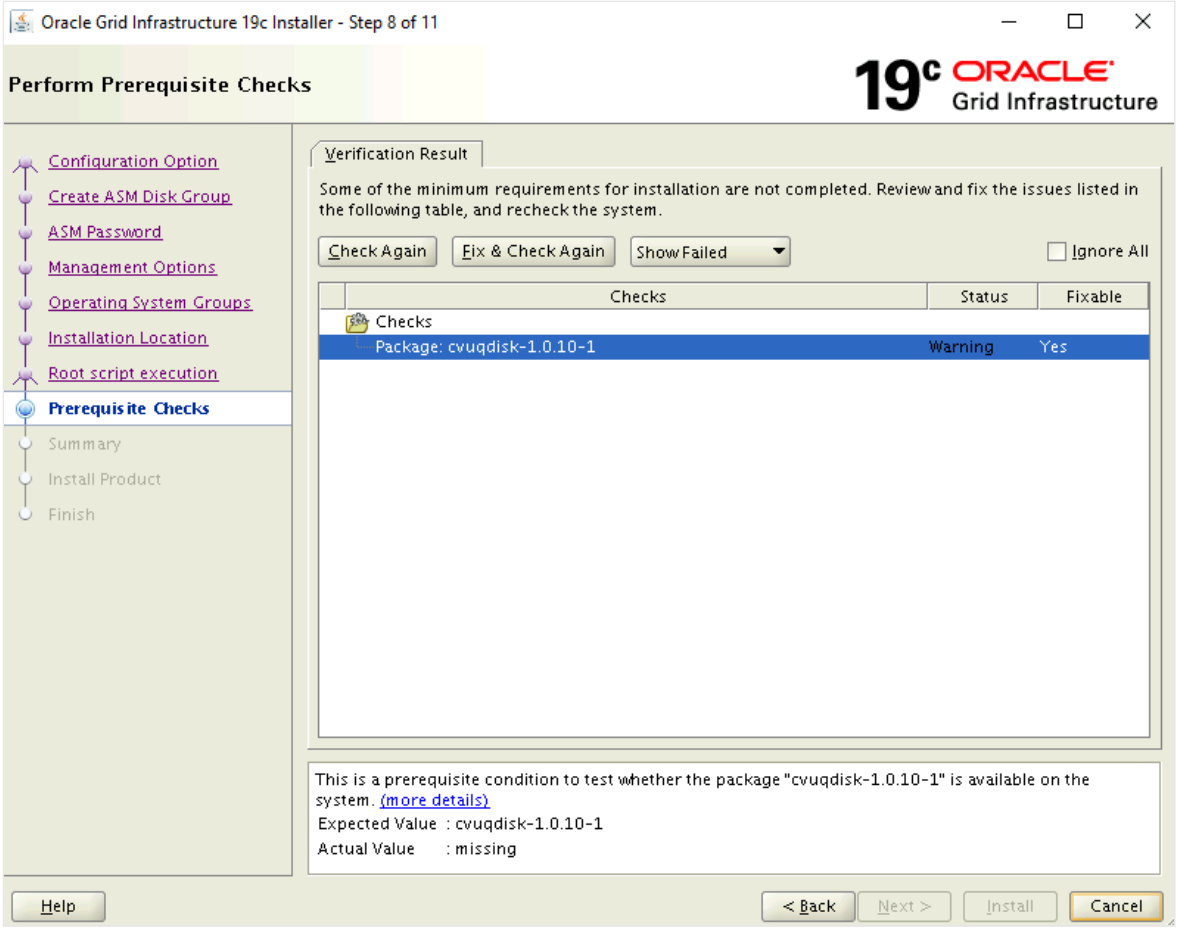
7. On the **Specify Installation Location** page, use the default settings. Click **Next** to continue.

8. On the **Root script execution configuration** page, select the **Automatically run configuration scripts** check box. Then, select the **Use "root" user credential** option, and enter the root user password.

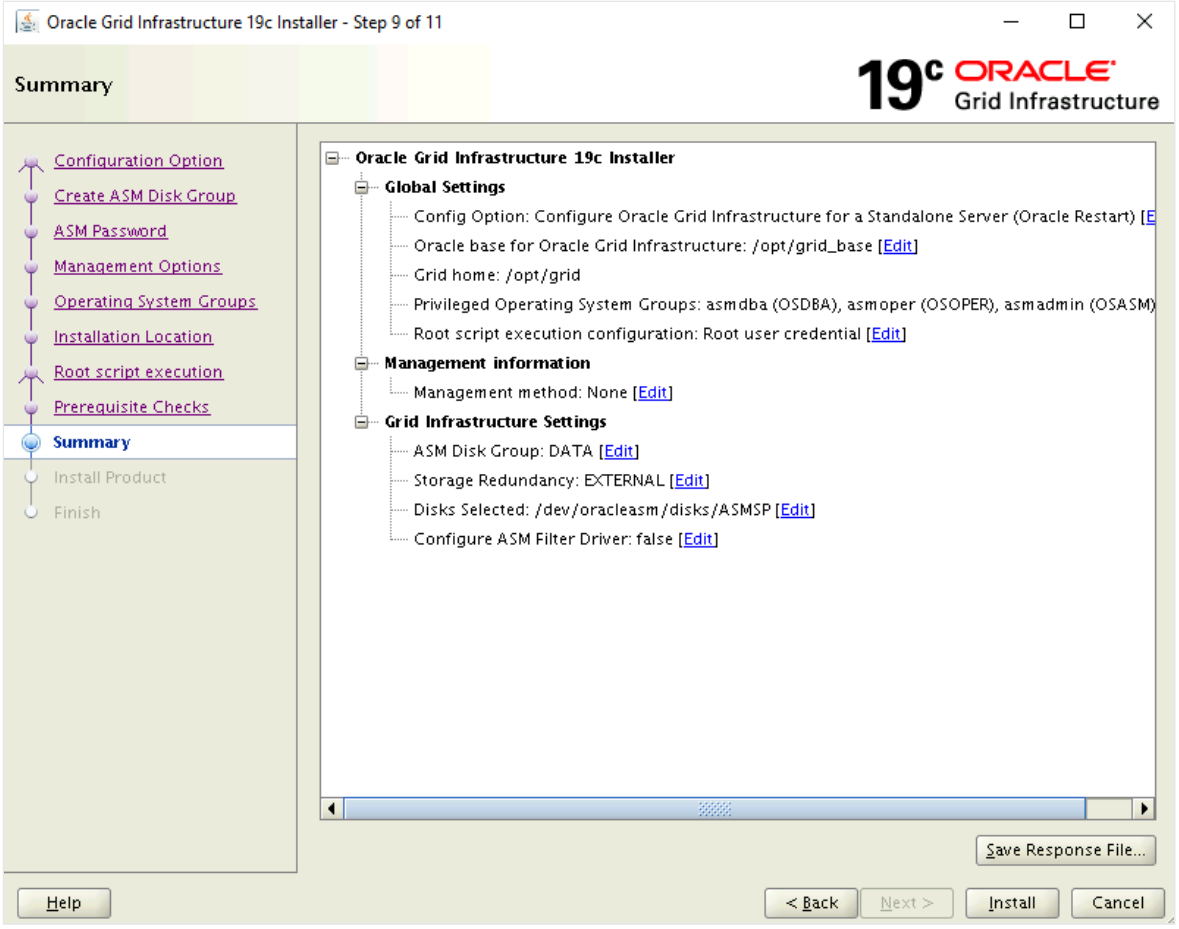


9. On the **Perform Prerequisite Checks** page, the current setup fails with errors. Select **Fix & Check Again**.

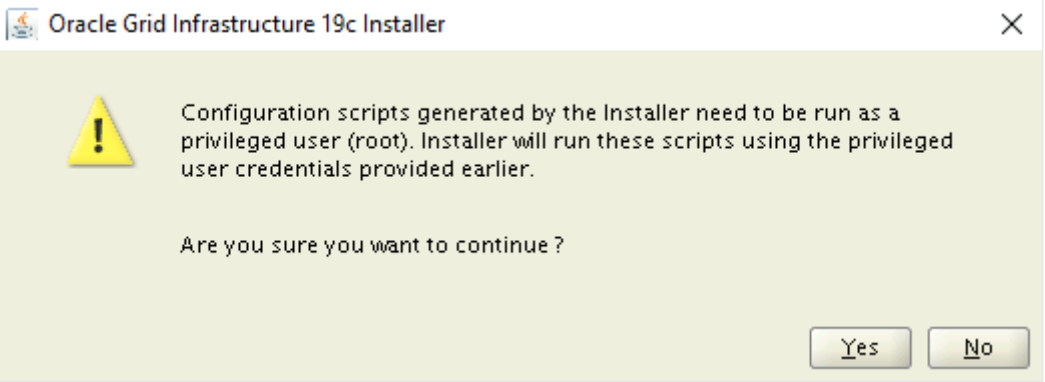
10. In the **Fixup Script** dialog box, click **OK**.



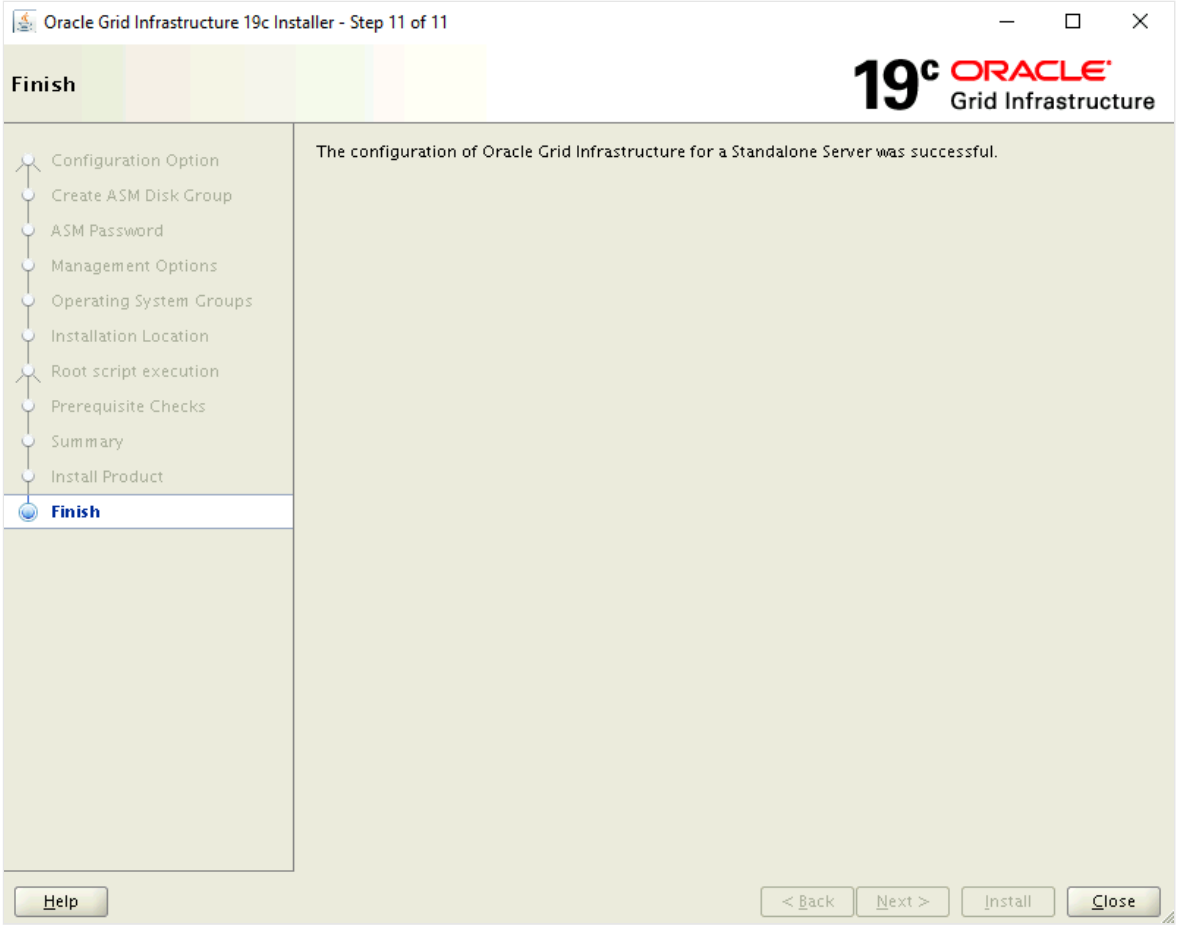
11. On the **Summary** page, review your selected settings, and then click `Install`.



12. A warning dialog box appears informing you configuration scripts need to be run as a privileged user. Click **Yes** to continue.



13. On the **Finish** page, click **Close** to finish the installation.



Setup Oracle ASM

Complete following steps to setup Oracle ASM.

1. Ensure you are still signed in as **oracle**, to asmVM from Bastion ssh session.

   Run following to set context. If you still have the shell open from previous command, you may skip this step.

   Bash

   ```
   $ sudo su - oracle
   $ export DISPLAY=10.0.0.4:0.0
   ```

   Launch the Oracle Automatic Storage Management Configuration Assistant

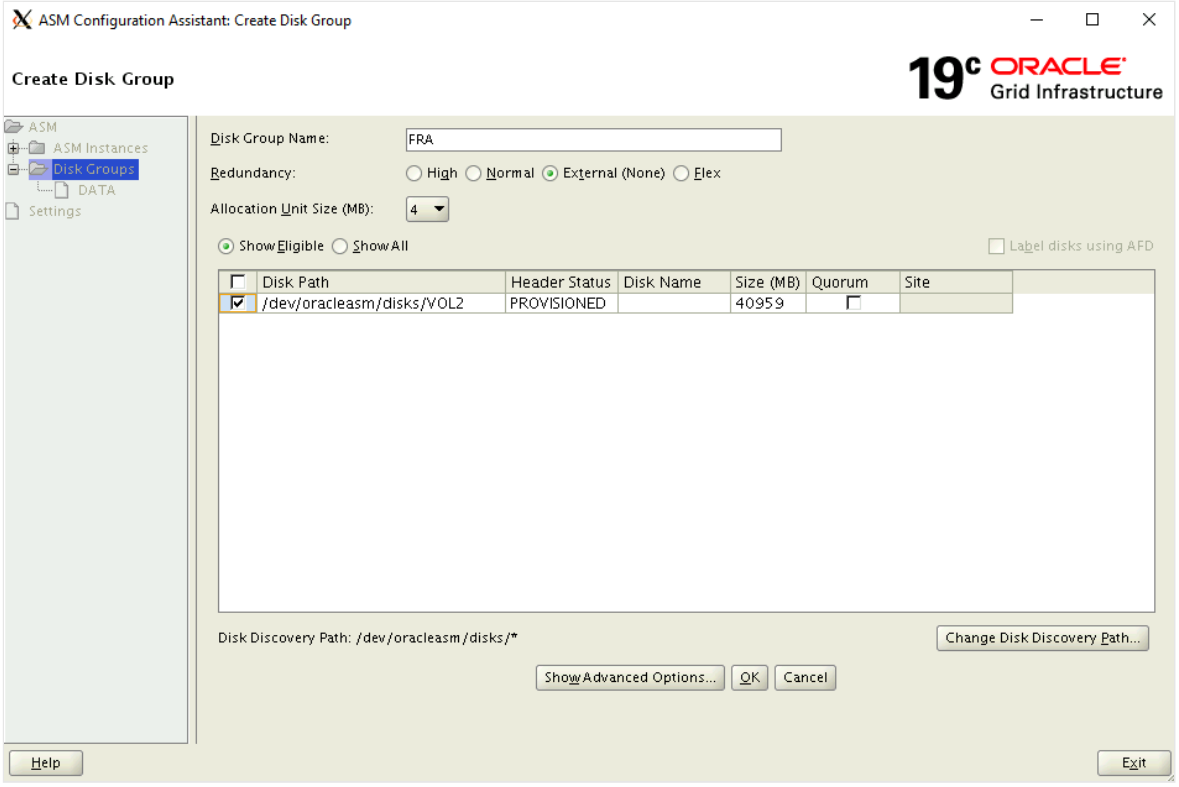
   Bash

   ```
   $ cd /opt/grid/bin
   $ ./asmca
   ```

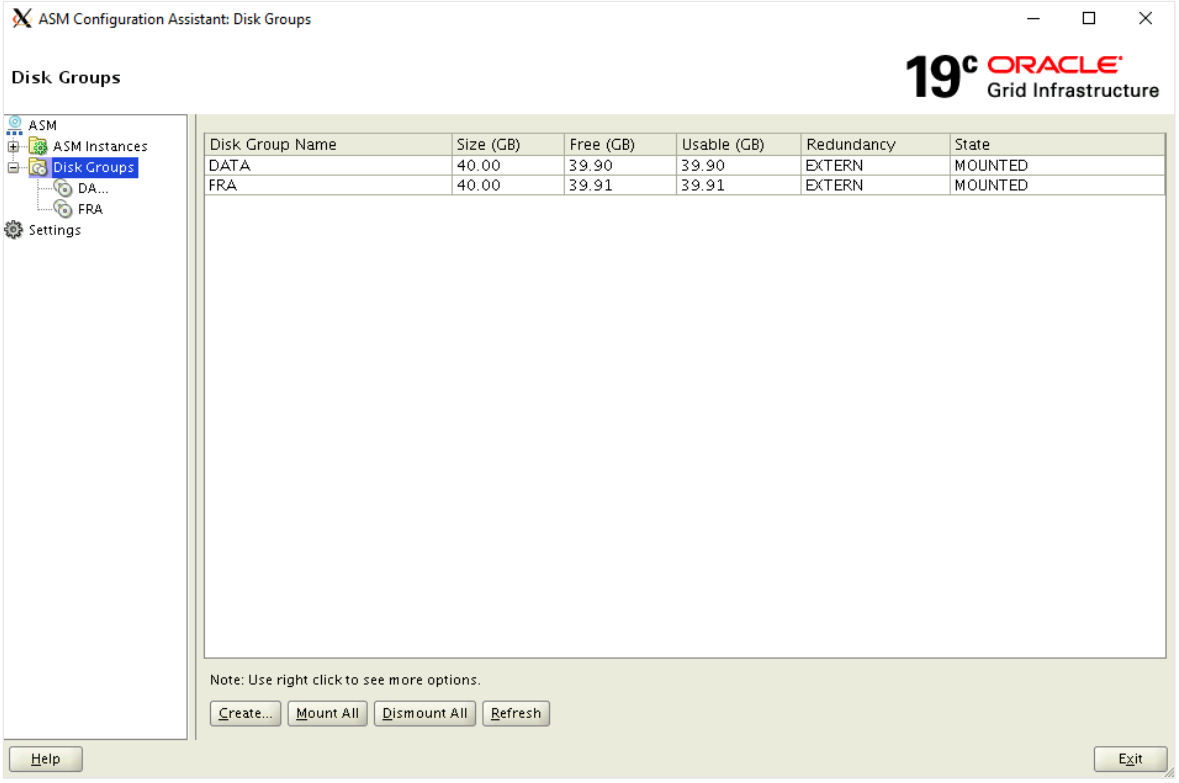
   In a few minutes, Oracle ASM Configuration Assistant window opens on **asmXServer** VM.

2. Select **DATA** under **Disk Groups** in the tree and click the **Create** button at the bottom.

3. In the **Create Disk Group** dialog box:
   a. Enter the disk group name **FRA**.
   b. For Redundancy option, select External (None).
   c. Under **Select Member Disks**, select **/dev/oracleasm/disks/VOL2**
   d. Under **Allocation Unit Size**, select **4**.
   e. Click **ok** to create the disk group.
   f. Click **ok** to close the confirmation window.



4. Select **Exit** to close ASM Configuration Assistant.



Create the database

The Oracle database software is already installed on the Azure Marketplace image. To create a database, complete the following steps:

1. Ensure the context is set to **oracle** user

- Run following to set context. If you still have the shell open from previous command, this may not be necessary.

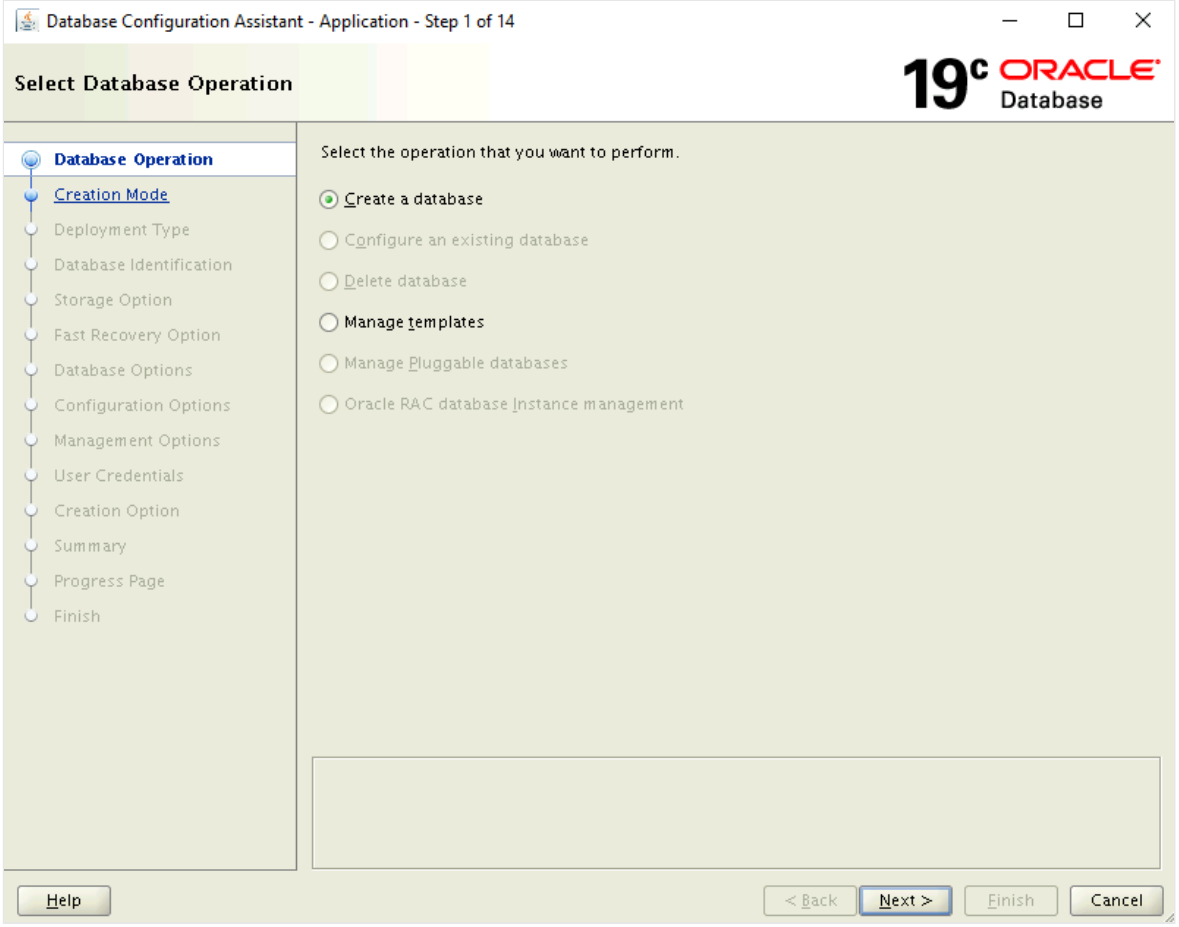
```bash
$ sudo su - oracle
$ export DISPLAY=10.0.0.4:0.0
```

Run Database Configuration Assistant

```bash
$ cd /u01/app/oracle/product/19.0.0/dbhome_1/bin
$ ./dbca
```

In a few seconds, Database Configuration Assistant window opens on **asmXServer** VM.

2. On the **Database Operation** page, click **Create Database**.



3. On the **Creation Mode** page:
   a. Ensure **Typical configuration** is selected.
   b. Enter a name for the database: **asmdb**
   c. For **Storage Type**, ensure **Automatic Storage Management (ASM)** is selected.

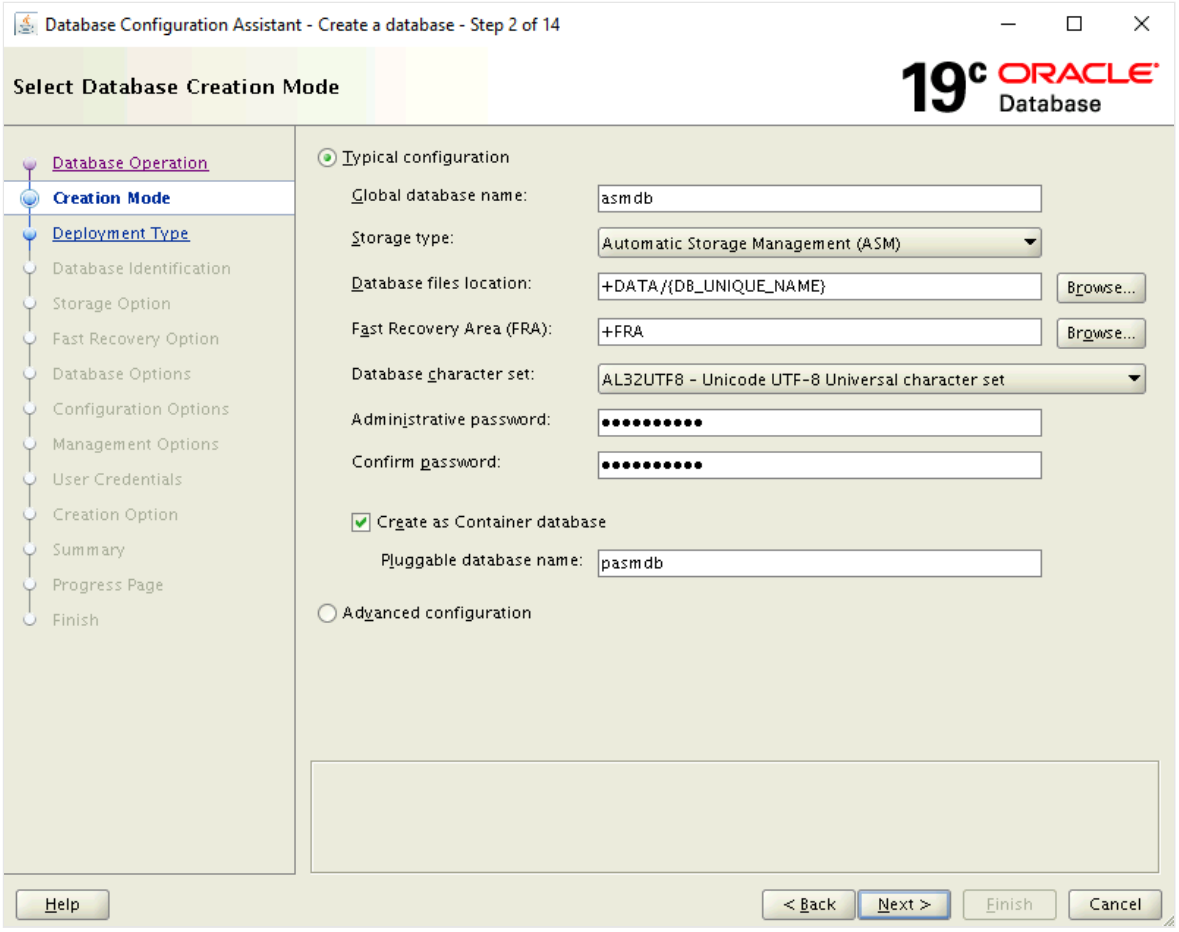
d. For **Database Files Location**, browse and select **DATA** location.

e. For **Fast Recovery Area**, browse and select **FRA** location.

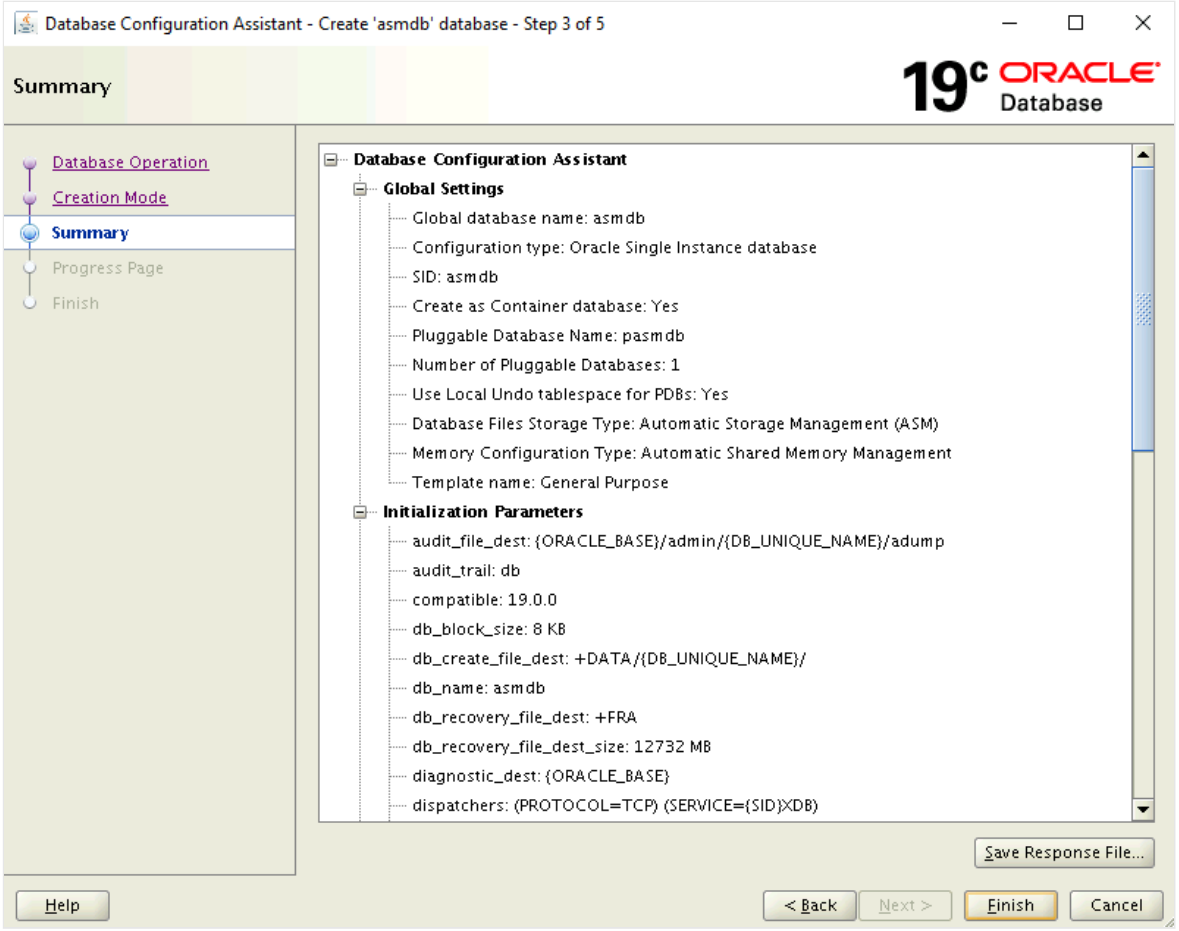
f. Type in an **Administrative Password** and **confirm password**.

g. Ensure **create as container database** is selected.

h. Type in a **pluggable database name** value: **pasmdb**



4. On the **Summary** page, review your selected settings, and then click **Finish** to create the database. Database creation may take more than 10 minutes.



5. The Database has been created. On the **Finish** page, you may opt to unlock additional accounts to use this database and change the passwords. If you wish to do so, select **Password Management** - otherwise click on **Close**.

Delete the asmXServer VM

asmXServer VM is only used during setup. You can safely delete it after completing this lab document but keep your ASM on Azure lab setup intact.

Azure CLI

```
$ az vm delete --resource-group ASMOnAzureLab --name asmXServer --force-deletion yes

$ az network public-ip delete --resource-group ASMOnAzureLab --name asmXServerPublicIP
```

Delete ASM On Azure Lab Setup

You have successfully configured Oracle Automatic Storage Management on the Oracle DB image from the Azure Marketplace. When you no longer need this environment, you can use the following command to remove the resource group and all related resources:

Azure CLI

```
$ az group delete --name ASMOnAzureLab
```

Next steps

Tutorial: Configure Oracle DataGuard

Tutorial: Configure Oracle GoldenGate

Review Architect an Oracle DB

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Implement Oracle Data Guard on a Linux-based Azure virtual machine

Article • 08/22/2024

Applies to: ✔ Linux VMs

You can use the Azure CLI to create and manage Azure resources from the command line or in scripts. This article describes how to use the Azure CLI to deploy an Oracle Database 19c Release 3 database from the Azure Marketplace image. This article then shows you, step by step, how to install and configure Data Guard on an Azure virtual machine (VM). To help secure the environment, no ports will be publicly accessible and an Azure Bastion instance will provide access to the Oracle VMs.

Before you start, make sure that the Azure CLI is installed. For more information, see the Azure CLI installation guide.

Prepare the environment

Assumptions

To install Oracle Data Guard, you need to create two Azure VMs on the same availability set:

- The primary VM (`OracleVM1`) has a running Oracle instance.
- The standby VM (`OracleVM2`) has the Oracle software installed only.

The Azure Marketplace image that you use to create the VMs is `Oracle:oracle-database-19-3:oracle-database-19-0904:latest`.

> **Note**
>
> Be aware of versions that have reached end of life (EOL) and are no longer supported by Red Hat. Uploaded images that are at or beyond EOL will be supported on a basis of reasonable business effort. For more information, see the Red Hat webpage for product lifecycles.

Sign in to Azure

Sign in to your Azure subscription by using the az login command and follow the on-screen directions.

Azure CLI

```
az login
```

Set environment variables

Adjust the `LOCATION` variable for your environment:

Azure CLI

```
LOCATION=eastus
RESOURCE_GROUP="Oracle-Lab"
VM_USERNAME="azureuser"
VM_PASSWORD="OracleLab123"
VNET_NAME="${RESOURCE_GROUP}VNet"
```

Enable the Azure Bastion extension

Include the Azure Bastion extension in the Azure CLI:

Azure CLI

```
az extension add \
  --name bastion
```

Create a resource group

Create a resource group by using the az group create command. An Azure resource group is a logical container in which Azure resources are deployed and managed.

Azure CLI

```
az group create \
  --name $RESOURCE_GROUP \
  --location $LOCATION
```

Create a virtual network with two subnets

Create a virtual network where you'll connect all compute services. One subnet will host Azure Bastion, an Azure service that helps protect your databases from public access. The second subnet will host the two Oracle database VMs.

Also, create a network security group that all services will reference to determine what ports are publicly exposed. Only port 443 will be exposed. The Azure Bastion service will open this port automatically when you create that service instance.

Azure CLI

```
az network vnet create \
  --resource-group $RESOURCE_GROUP \
  --location $LOCATION \
  --name $VNET_NAME \
  --address-prefix "10.0.0.0/16"
az network vnet subnet create \
  --resource-group $RESOURCE_GROUP \
  --name AzureBastionSubnet \
  --vnet-name $VNET_NAME \
  --address-prefixes 10.0.0.0/24
az network vnet subnet create \
  --resource-group $RESOURCE_GROUP \
  --name OracleSubnet \
  --vnet-name $VNET_NAME \
  --address-prefixes 10.0.1.0/24
az network nsg create \
  --name OracleVM-NSG \
  --resource-group $RESOURCE_GROUP \
  --location $LOCATION
```

Create an availability set

Creating an availability set is optional, but we recommend it. For more information, see the guidelines for Azure availability sets.

Azure CLI

```
az vm availability-set create \
  --resource-group $RESOURCE_GROUP \
  --name OracleVMAvailabilitySet \
  --platform-fault-domain-count 2 \
  --platform-update-domain-count 2
```

Create two virtual machines

Create two VMs by using the az vm create command. The following example creates VMs named `OracleVM1` and `OracleVM2`.

Create `OracleVM1` (primary):

Azure CLI

```
az vm create \
  --resource-group $RESOURCE_GROUP \
  --name OracleVM1 \
  --availability-set OracleVMAvailabilitySet \
  --image Oracle:oracle-database-19-3:oracle-database-19-0904:latest \
  --size Standard_DS1_v2 \
  --authentication-type password \
  --admin-username $VM_USERNAME \
  --admin-password $VM_PASSWORD \
  --vnet-name $VNET_NAME \
  --subnet OracleSubnet \
  --nsg OracleVM-NSG \
  --os-disk-size-gb 32
```

Create `OracleVM2` (standby):

Azure CLI

```
az vm create \
  --resource-group $RESOURCE_GROUP \
  --name OracleVM2 \
  --availability-set OracleVMAvailabilitySet \
  --image Oracle:oracle-database-19-3:oracle-database-19-0904:latest \
  --size Standard_DS1_v2 \
  --authentication-type password \
  --admin-username $VM_USERNAME \
  --admin-password $VM_PASSWORD \
  --vnet-name $VNET_NAME \
  --subnet OracleSubnet \
  --nsg OracleVM-NSG \
  --os-disk-size-gb 32
```

Create the Azure Bastion service instance

Azure Bastion provides a secure tunnel to all services hosted within the virtual network. It serves as a jump box to eliminate direct access to your Oracle databases.

Create a public IP address to access the Azure Bastion service:

Azure CLI

```
az network public-ip create \
  --resource-group $RESOURCE_GROUP \
  --name OracleLabBastionPublicIP \
  --sku Standard
```

Create the Azure Bastion service instance:

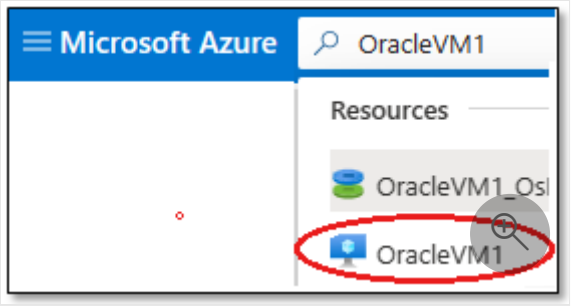
Azure CLI

```
az network bastion create \
  --location $LOCATION \
  --name OracleLabBastion \
  --public-ip-address OracleLabBastionPublicIP \
  --resource-group $RESOURCE_GROUP \
  --vnet-name $VNET_NAME \
  --sku basic
```

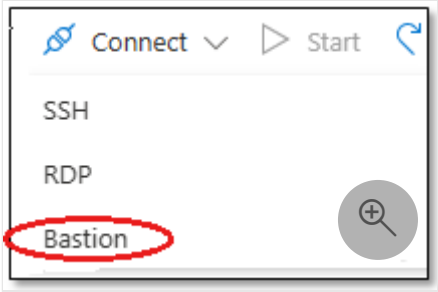
Connect to the virtual machine

Now, you'll access `OracleVM1` by using the Azure Bastion service from the Azure portal.

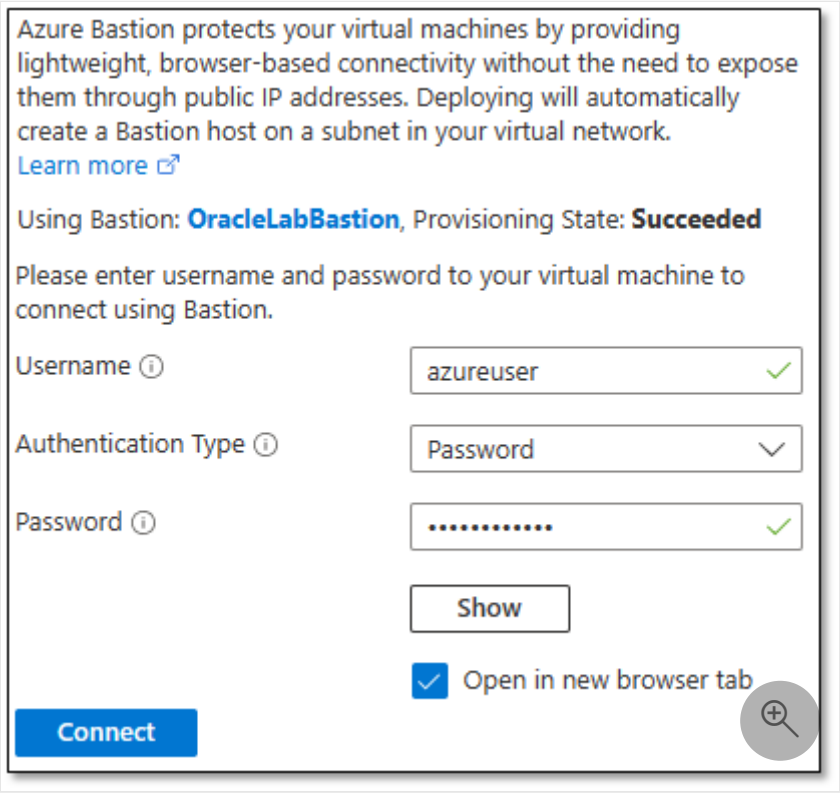
In the search box at the top of the window, enter **OracleVM1**. In the list of results, select **OracleVM1** to open that VM.



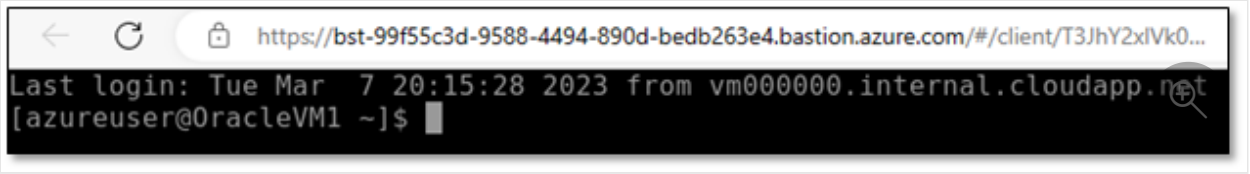
At the top of the pane, select **Connect** > **Bastion**.



Enter the username and password, and then select the **Connect** button.



A new tab opens with a secure connection to your virtual machine, where the Oracle software is already installed from an Azure Marketplace image.



Configure OracleVM1 (primary)

Disable the firewall:

```bash
sudo systemctl stop firewalld
sudo systemctl disable firewalld
```

Set the `oracle` user password:

```bash
sudo passwd oracle
```

Enter the `azureuser` password: `OracleLab123`.

Change the `oracle` user password to `OracleLab123`. (Enter it again to verify.)

Create the database on OracleVM1 (primary)

The Oracle software is already installed on the Azure Marketplace image, so the next step is to install the database.

Switch to the Oracle superuser:

Bash

```
sudo su - oracle
```

Create the database:

Bash

```
dbca -silent \
   -createDatabase \
   -datafileDestination /u01/app/oracle/cdb1 \
   -templateName General_Purpose.dbc \
   -gdbname cdb1 \
   -sid cdb1 \
   -responseFile NO_VALUE \
   -characterSet AL32UTF8 \
   -sysPassword OracleLab123 \
   -systemPassword OracleLab123 \
   -createAsContainerDatabase true \
   -numberOfPDBs 1 \
   -pdbName pdb1 \
   -pdbAdminPassword OracleLab123 \
   -databaseType MULTIPURPOSE \
   -automaticMemoryManagement false \
   -storageType FS
```

Outputs should look similar to the following response:

Output

```
Copying database files
1% complete
2% complete
8% complete
13% complete
19% complete
27% complete
Creating and starting Oracle instance
29% complete
32% complete
33% complete
34% complete
38% complete
```

```
42% complete
43% complete
45% complete
Completing Database Creation
48% complete
51% complete
53% complete
62% complete
70% complete
72% complete
Creating Pluggable Databases
78% complete
100% complete
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/cdb1/cdb1.log" for
further details.
```

Set the `ORACLE_SID` and `ORACLE_HOME` variables:

```bash
$ ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1; export ORACLE_HOME
$ ORACLE_SID=cdb1; export ORACLE_SID
```

Optionally, you can add `ORACLE_HOME` and `ORACLE_SID` to the */home/oracle/.bashrc* file, so that these settings are saved for future logins:

```bash
# add oracle home
export ORACLE_HOME= /u01/app/oracle/product/19.0.0/dbhome_1
# add oracle sid
export ORACLE_SID=cdb1
```

Configure Data Guard

Enable archive log mode on myVM1 (primary)

Enable log mode:

```bash
sqlplus / as sysdba
SQL> SELECT log_mode FROM v$database;
LOG_MODE
------------
NOARCHIVELOG
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```

Enable force logging, and make sure that at least one log file is present:

Bash

```
SQL> ALTER DATABASE FORCE LOGGING;
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

Create standby redo logs, setting the same size and quantity as the primary database redo logs:

Bash

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE
('/u01/app/oracle/cdb1/standby_redo01.log') SIZE 200M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE
('/u01/app/oracle/cdb1/standby_redo02.log') SIZE 200M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE
('/u01/app/oracle/cdb1/standby_redo03.log') SIZE 200M;
SQL> ALTER DATABASE ADD STANDBY LOGFILE
('/u01/app/oracle/cdb1/standby_redo04.log') SIZE 200M;
```

Turn on Flashback (which makes recovery a lot easier) and set `STANDBY_FILE_MANAGEMENT` to `AUTO`. Exit SQL*Plus after that.

Bash

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size=50G scope=both sid='*';
SQL> ALTER SYSTEM SET db_recovery_file_dest='/u01/app/oracle/cdb1'
scope=both sid='*';
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO SCOPE=BOTH;
SQL> ALTER SYSTEM SET STANDBY_FILE_MANAGEMENT=AUTO;
SQL> EXIT;
```

Set up the service on OracleVM1 (primary)

Edit or create the *tnsnames.ora* file, which is in the *$ORACLE_HOME/network/admin* folder.

Add the following entries:

Bash

```
cdb1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = cdb1)
    )
  )
cdb1_stby =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = cdb1)
    )
  )
```

Edit or create the *listener.ora* file, which is in the *$ORACLE_HOME/network/admin* folder.

Add the following entries:

Bash

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = cdb1_DGMGRL)
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0/dbhome_1)
      (SID_NAME = cdb1)
    )
  )
ADR_BASE_LISTENER = /u01/app/oracle
```

Enable Data Guard Broker:

Bash

```
sqlplus / as sysdba
SQL> ALTER SYSTEM SET dg_broker_start=true;
SQL> CREATE pfile FROM spfile;
SQL> EXIT;
```

Copy the parameter file to the standby server:

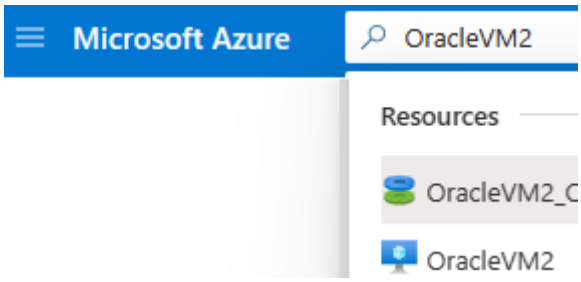
```bash
scp -p $ORACLE_HOME/dbs/initcdb1.ora oracle@OracleVM2:$ORACLE_HOME/dbs/
```

Start the listener:

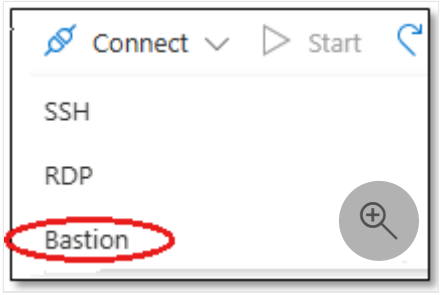
```bash
lsnrctl stop
lsnrctl start
```

Set up the service on OracleVM2 (standby)

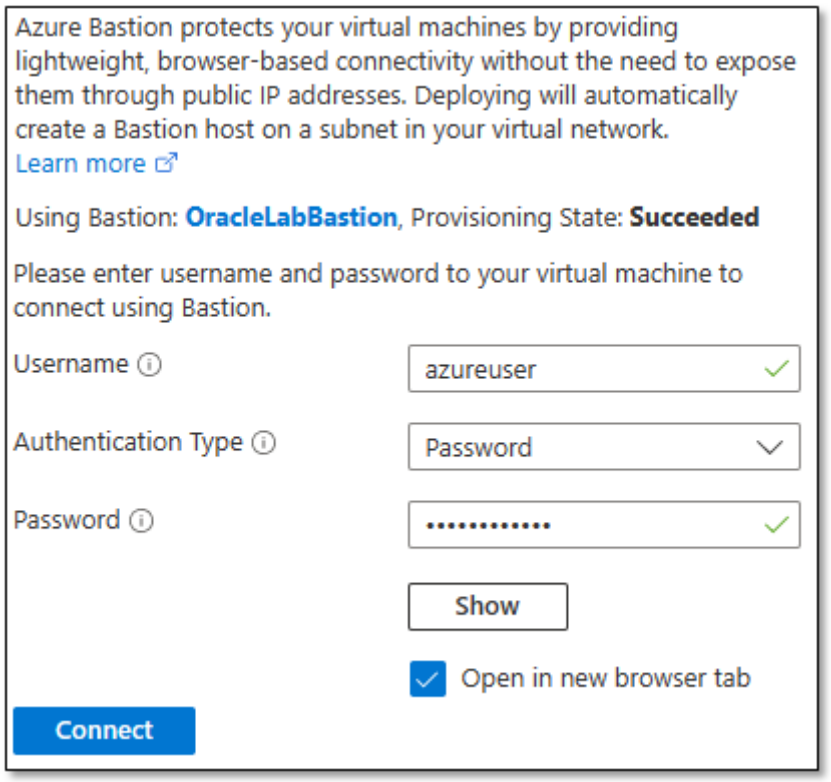
Return to the Azure portal. Search for **OracleVM2**, and then select the resource in the list of results.



At the top of the pane, select **Connect** > **Bastion**.



Enter the username and password, and then select the **Connect** button.



Disable the firewall on OracleVM2 (standby)

```bash
sudo systemctl stop firewalld
sudo systemctl disable firewalld
```

Configure the environment for OracleVM1

Set the `oracle` user password:

```bash
sudo passwd oracle
```

Enter the `azureuser` password: `OracleLab123`.

Change the `oracle` user password to `OracleLab123`. (Enter it again to verify.)

Switch to the `oracle` superuser:

```bash
$ sudo su - oracle
```

Set the `ORACLE_SID` and `ORACLE_HOME` variables:

```bash
ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1; export ORACLE_HOME
ORACLE_SID=cdb1; export ORACLE_SID
```

Optionally, you can add `ORACLE_HOME` and `ORACLE_SID` to the */home/oracle/.bashrc* file, so that these settings are saved for future logins:

```bash
# add oracle home
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/dbhome_1
# add oracle sid
export ORACLE_SID=cdb1
```

Edit or create the *tnsnames.ora* file, which is in the *$ORACLE_HOME/network/admin* folder.

Add the following entries:

```bash
cdb1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM1)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = cdb1)
    )
  )
cdb1_stby =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM2)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = cdb1)
    )
  )
```

Edit or create the *listener.ora* file, which is in the *$ORACLE_HOME/network/admin* folder.

Add the following entries:

```bash
```

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = OracleVM2)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = cdb1_DGMGRL)
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0/dbhome_1)
      (SID_NAME = cdb1)
    )
  )
ADR_BASE_LISTENER = /u01/app/oracle
```

Start the listener:

Bash

```
lsnrctl stop
lsnrctl start
```

Restore the database to OracleVM2 (standby)

Create the parameter file */tmp/initcdb1_stby.ora* with the following contents:

Bash

```
*.db_name='cdb1'
```

Create folders:

Bash

```
$ mkdir -p /u01/app/oracle/cdb1
$ mkdir -p /u01/app/oracle/oradata/cdb1/pdbseed
$ mkdir -p /u01/app/oracle/oradata/cdb1/pdb1
$ mkdir -p /u01/app/oracle/fast_recovery_area/cdb1
$ mkdir -p /u01/app/oracle/admin/cdb1/adump
```

Create a password file:

Bash

```
$ orapwd file=/u01/app/oracle/product/19.0.0/dbhome_1/dbs/orapwcdb1
password=OracleLab123 entries=10 force=y
```

Start the database on `OracleVM2`:

Bash

```
 export ORACLE_SID=cdb1
 sqlplus / as sysdba
SQL> CREATE spfile from pfile;
SQL> STARTUP NOMOUNT PFILE='/tmp/initcdb1_stby.ora';
SQL> EXIT;
```

Restore the database by using the Oracle Recovery Manager (RMAN) tool:

Bash

```
$ rman TARGET sys/OracleLab123@cdb1 AUXILIARY sys/OracleLab123@cdb1_stby
```

Run the following commands in RMAN:

Bash

```
DUPLICATE TARGET DATABASE
  FOR STANDBY
  FROM ACTIVE DATABASE
  DORECOVER
  SPFILE
    SET db_unique_name='CDB1_STBY' COMMENT 'Is standby'
  NOFILENAMECHECK;
```

Messages similar to the following ones appear when the commands are completed:

Output

```
media recovery complete, elapsed time: 00:00:00
Finished recover at 29-JUN-22
Finished Duplicate Db at 29-JUN-22
```

Exit RMAN:

Bash

```
RMAN> EXIT;
```

Enable Data Guard Broker:

```bash
sqlplus / as sysdba
SQL> ALTER SYSTEM SET dg_broker_start=true;
SQL> EXIT;
```

Configure Data Guard Broker on OracleVM1 (primary)

Start Data Guard Manager and log in by using SYS and a password. (Don't use OS authentication.)

```bash
$ dgmgrl sys/OracleLab123@cdb1
DGMGRL for Linux: Version 19.0.0.0 - 64bit Production
Copyright (c) 2000, 2013, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
Connected as SYSDBA.
DGMGRL> CREATE CONFIGURATION my_dg_config AS PRIMARY DATABASE IS cdb1
CONNECT IDENTIFIER IS cdb1;
Configuration "my_dg_config" created with primary database "cdb1"
DGMGRL> ADD DATABASE cdb1_stby AS CONNECT IDENTIFIER IS cdb1_stby MAINTAINED
AS PHYSICAL;
Database "cdb1_stby" added
DGMGRL> ENABLE CONFIGURATION;
Enabled.
```

Review the configuration:

```bash
DGMGRL> SHOW CONFIGURATION;
Configuration - my_dg_config
  Protection Mode: MaxPerformance
  Members:
  cdb1      - Primary database
  cdb1_stby - Physical standby database
Fast-Start Failover: DISABLED
Configuration Status:
SUCCESS   (status updated 26 seconds ago)
```

You've completed the Oracle Data Guard setup. The next section shows you how to test the connectivity and switch over.

Connect the database from the client machine

Update the *tnsnames.ora* file on your client machine. This file is usually in *$ORACLE_HOME/network/admin*.

Bash

```
cdb1=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (HOST=OracleVM1)
      (PORT=1521)
    )
    (CONNECT_DATA=
      (SERVER=dedicated)
      (SERVICE_NAME=cdb1)
    )
  )
cdb1_stby=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP)
      (HOST=OracleVM2)
      (PORT=1521)
    )
    (CONNECT_DATA=
      (SERVER=dedicated)
      (SERVICE_NAME=cdb1_stby)
    )
  )
```

Start SQL*Plus:

Bash

```
$ sqlplus sys/OraPasswd1@cdb1
SQL*Plus: Release 19.0.0.0 Production on Wed May 10 14:18:31 2022
Copyright (c) 1982, 2016, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 19.0.0.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
SQL>
```

Test the Data Guard configuration

Switch over the database on OracleVM1 (primary)

Switch from primary to standby (`cdb1` to `cdb1_stby`):

Bash

```
$ dgmgrl sys/OracleLab123@cdb1
DGMGRL for Linux: Version 19.0.0.0 - 64bit Production
Copyright (c) 2000, 2013, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
Connected as SYSDBA.
DGMGRL> SWITCHOVER TO cdb1_stby;
Performing switchover NOW, please wait...
Operation requires a connection to instance "cdb1" on database "cdb1_stby"
Connecting to instance "cdb1"...
Connected as SYSDBA.
New primary database "cdb1_stby" is opening...
Operation requires start up of instance "cdb1" on database "cdb1"
Starting instance "cdb1"...
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "cdb1_stby"
DGMGRL>
```

You can now connect to the standby database. Start SQL*Plus:

Bash

```
$ sqlplus sys/OracleLab123@cdb1_stby
SQL*Plus: Release 19.0.0.0 Production on Wed May 10 14:18:31 2022
Copyright (c) 1982, 2016, Oracle.  All rights reserved.
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
SQL>
```

Switch over the database on OracleVM2 (standby)

To switch over, run the following code on `OracleVM2`:

Bash

```
$ dgmgrl sys/OracleLab123@cdb1_stby
DGMGRL for Linux: Version 190.0.0.0 - 64bit Production
Copyright (c) 2000, 2013, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
Connected as SYSDBA.
DGMGRL> SWITCHOVER TO cdb1;
Performing switchover NOW, please wait...
Operation requires a connection to instance "cdb1" on database "cdb1"
Connecting to instance "cdb1"...
Connected as SYSDBA.
New primary database "cdb1" is opening...
```

```
Operation requires start up of instance "cdb1" on database "cdb1_stby"
Starting instance "cdb1"...
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "cdb1"
```

Once again, you should now be able to connect to the primary database. Start SQL*Plus:

Bash

```
$ sqlplus sys/OracleLab123@cdb1
SQL*Plus: Release 19.0.0.0 Production on Wed May 10 14:18:31 2022
Copyright (c) 1982, 2016, Oracle.  All rights reserved.
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
SQL>
```

You've finished the installation and configuration of Data Guard on Oracle Linux.

Delete the virtual machine

When you no longer need the VMs, you can use the following command to remove the resource group, the VMs, and all related resources:

Azure CLI

```
az group delete --name $RESOURCE_GROUP
```

Next steps

- Tutorial: Create highly available virtual machines
- Explore Azure CLI samples for VM deployment

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Implement Oracle Golden Gate on an Azure Linux virtual machine

Article • 08/22/2024

Applies to: ✔ Linux VMs

The Azure CLI is used to create and manage Azure resources from the command line or in scripts. This guide details how to use the Azure CLI to deploy an Oracle 19c database from the Azure Marketplace gallery image.

This document shows you step-by-step how to create, install, and configure Oracle Golden Gate on an Azure VM. In this tutorial, two virtual machines are set up in an availability set in a single region. The same tutorial can be used to set up OracleGolden Gate for VMs in different Availability Zones in a single Azure region or for VMs set up in two different regions.

Before you start, make sure that the Azure CLI has been installed. For more information, see Azure CLI installation guide.

Introduction

GoldenGate is a logical replication software that enables real-time replication, filtering, and transformation of data from a source database to a target database. This feature ensures that changes in the source database are replicated in real-time, making it possible for the target database to be up-to-date with the latest data.

Use GoldenGate mainly for heterogeneous replication cases, such as replicating data from different source databases to a single database. For example, a data warehouse. You can also use it for cross-platform migrations, such as from SPARC and AIX to Linux x86 environments, and advanced high availability and scalability scenarios.

Additionally, GoldenGate is also suitable for near-zero downtime migrations since it supports online migrations with minimal disruption to the source systems.

While GoldenGate facilitates bi-directional replication, the application must be configured accordingly. Additionally, the software permits filtering and transformation of data to meet specific business requirements, enabling users to make use of the replicated data for various purposes.

Prepare the environment

To perform the Oracle Golden Gate installation, you need to create two Azure VMs on the same availability set. The Marketplace image you use to create the VMs is **Oracle:oracle-database-19-3:oracle-database-19-0904:19.3.1**.

You also need to be familiar with Unix editor vi and have a basic understanding of X Server.

The following table is a summary of the environment configuration:

Expand table

	Primary site	Replicate site
Oracle release	Oracle Database 19.3.1.0	Oracle Database 19.3.1.0
Machine name	ggVM1	ggVM2
Operating system	Oracle Linux 7.x	Oracle Linux 7.x
Oracle SID	CDB1	CDB1
Replication schema	TEST	TEST
Golden Gate owner/replicate	C##GGADMIN	REPUSER
Golden Gate process	EXTORA	REPORA

Sign in to Azure

1. Open your preferred shell on Windows, Linux or Azure Shell.

2. Sign in to your Azure subscription with the az login command. Then follow the on-screen directions.

   Azure CLI

   ```
   $ az login
   ```

3. Ensure you're connected to the correct subscription by verifying subscription name and/or ID.

   Azure CLI

   ```
   $ az account show
   ```

   Output

```
{
  "environmentName": "XXXXX",
  "homeTenantId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "id": "<SUBSCRIPTION_ID>",
  "isDefault": true,
  "managedByTenants": [],
  "name": "<SUBSCRIPTION_NAME>",
  "state": "Enabled",
  "tenantId": XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
  "user": {
    "cloudShellID": true,
    "name": "aaaaa@bbbbb.com",
    "type": "user"
  }
}
```

Generate authentication keys

We use key file based authentication with ssh to connect to the Oracle Database VM. Ensure that you have your private (named `id_rsa`) and public (named `id_rsa.pub`) key files are created on your shell.

Location of key files depends on your source system.

Windows: %USERPROFILE%.ssh Linux: ~/.ssh

If they don't exist you can create a new keyfile pair.

ssh-keygen -m PEM -t rsa -b 4096

The .ssh directory and key files are created. For more information, refer to Create and manage SSH keys for authentication to a Linux VM in Azure

Create a resource group

To create a resource group, use the az group create command. An Azure resource group is a logical container in which Azure resources are deployed and managed.

Azure CLI

```
$ az group create --name GoldenGateOnAzureLab --location westus
```

Create and configure network

Create virtual network

Use following command to create the virtual network that hosts resources we create in this lab.

Azure CLI

```
$ az network vnet create \
  --name ggVnet \
  --resource-group GoldenGateOnAzureLab \
  --address-prefixes "10.0.0.0/16" \
  --subnet-name ggSubnet1 \
  --subnet-prefixes "10.0.0.0/24"
```

Create a Network Security Group (NSG)

1. Create network security group (NSG) to lock down your virtual network.

   Azure CLI

   ```
   $ az network nsg create \
     --resource-group GoldenGateOnAzureLab \
     --name ggVnetNSG
   ```

2. Create NSG rule to allow communication within virtual network.

   Azure CLI

   ```
   $ az network nsg rule create  --resource-group GoldenGateOnAzureLab --
   nsg-name ggVnetNSG \
       --name ggAllowVnet \
       --protocol '*' --direction inbound --priority 3400 \
       --source-address-prefix 'VirtualNetwork' --source-port-range '*' \
       --destination-address-prefix 'VirtualNetwork' --destination-port-
   range '*' --access allow
   ```

3. Create NSG rule to deny all inbound connections

   Azure CLI

   ```
   $ az network nsg rule create \
     --resource-group GoldenGateOnAzureLab \
     --nsg-name ggVnetNSG \
     --name ggDenyAllInBound \
     --protocol '*' --direction inbound --priority 3500 \
     --source-address-prefix '*' --source-port-range '*' \
   ```

```
  --destination-address-prefix '*' --destination-port-range '*' --
access deny
```

4. Assign NSG to Subnet where we host our servers.

Azure CLI

```
$ az network vnet subnet update --resource-group GoldenGateOnAzureLab -
-vnet-name ggVNet --name ggSubnet1 --network-security-group ggVnetNSG
```

Create Bastion Network

1. Create Bastion subnet. Name of the subnet must be **AzureBastionSubnet**

Azure CLI

```
$ az network vnet subnet create  \
    --resource-group GoldenGateOnAzureLab \
    --name AzureBastionSubnet \
    --vnet-name ggVnet \
    --address-prefixes 10.0.1.0/24
```

2. Create public IP for Bastion

Azure CLI

```
$ az network public-ip create \
    --resource-group GoldenGateOnAzureLab \
    --name ggBastionIP \
    --sku Standard
```

3. Create Azure Bastion resource. It takes about 10 minutes for the resource to deploy.

Azure CLI

```
$ az network bastion create \
    --resource-group GoldenGateOnAzureLab \
    --name ggBastion \
    --public-ip-address ggBastionIP \
    --vnet-name ggVnet \
    --sku Standard \
    --enable-tunneling \
    --enable-ip-connect true
```

Create X Server VM (ggXServer)

Replace your password and run following command to create a Windows workstation VM where we deploy X Server.

Azure CLI

```
$ az vm create \
     --resource-group GoldenGateOnAzureLab \
     --name ggXServer \
     --image MicrosoftWindowsDesktop:Windows-10:win10-22h2-pro-
g2:19045.2604.230207 \
     --size Standard_DS1_v2  \
     --vnet-name ggVnet \
     --subnet ggSubnet1 \
     --public-ip-sku Standard \
     --nsg "" \
     --data-disk-delete-option Delete \
     --os-disk-delete-option Delete \
     --nic-delete-option Delete \
     --admin-username azureuser \
     --admin-password <ENTER_YOUR_PASSWORD_HERE>
```

Connect to ggXServer through Bastion

Connect to **ggXServer** using Bastion.

1. Navigate to **ggXServer** from Azure portal.
2. Go to **Overview** in the left blade
3. Select **Connect** > **Bastion** on the menu at the top
4. Select Bastion tab
5. Click **Use Bastion**

Prepare ggXServer to run X Server

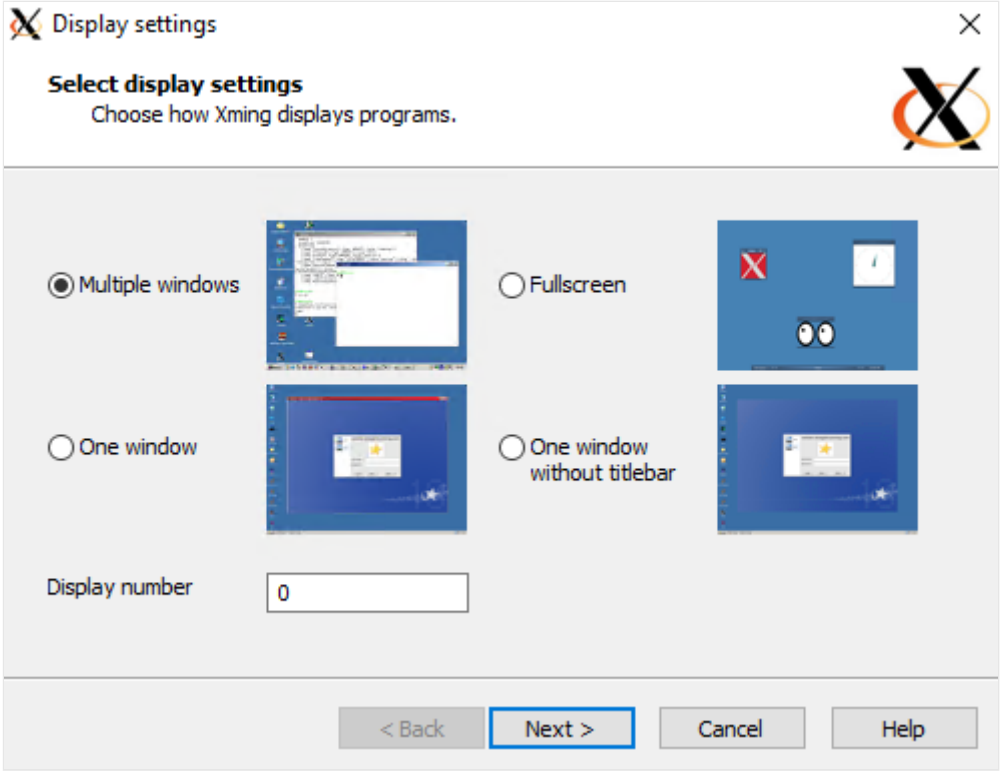
X Server is required for later steps of this lab. Perform following steps to install and start X Server.

1. Download Xming X Server for Windows to **ggXServer** and install with all default options.

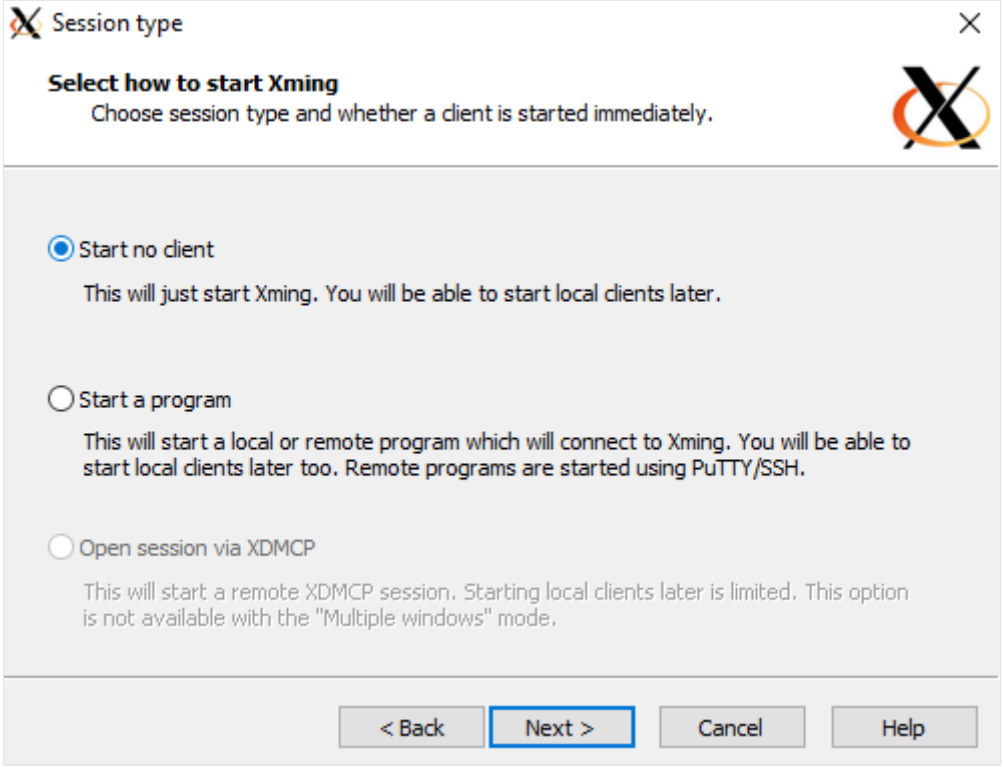
2. Ensure that you didn't select **Launch** at the end of installation

3. Launch "XLAUNCH" application from start menu.

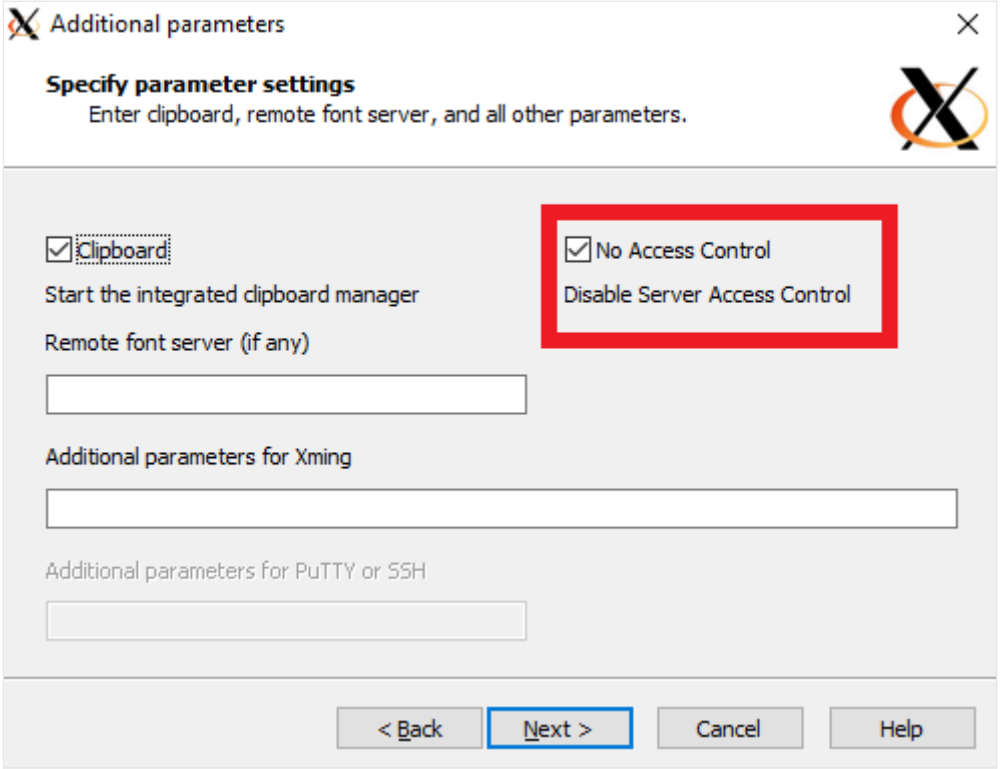
4. Select **Multiple Windows**



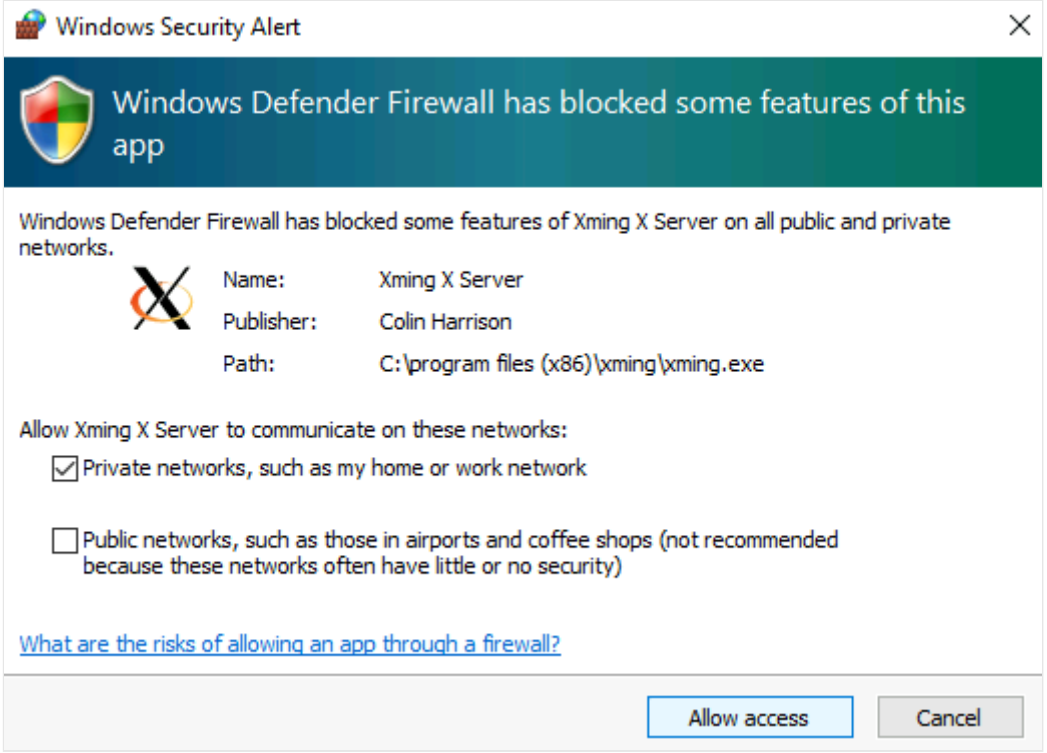
5. Select **Start no client**



6. Select **No access control**



7. Select **Allow Access** to allow X Server through Windows Firewall



If you restart your **ggXServer** VM, follow steps 2-6 above to restart X Server application.

Create Oracle database virtual machines

For this lab, we create virtual machines `ggVM1` and `ggVM2` from Oracle Database 19c image. If they don't already exist in the default key location, this command also creates SSH keys. To use a specific set of keys, use the `--ssh-key-value` option. If you have

already created your SSH keys in Generate authentication keys section, those keys will be used.

When creating a new virtual machine `size` parameter indicates the size and type of virtual machine created. Depending on the Azure region you selected to create virtual machine and your subscription settings, some virtual machine sizes and types may not be available for you to use. The following example uses minimum required size for this lab `Standard_DS1_v2`. If you want to change specs of virtual machine, select one of the available sizes from Azure VM Sizes. For test purposes, you may choose from General Purpose (D-Series) virtual machine types. For production or pilot deployments, Memory Optimized (E-Series and M-Series) are more suitable.

Create ggVM1 (primary)

Azure CLI

```
$ az vm create \
     --resource-group GoldenGateOnAzureLab \
     --name ggVM1 \
     --image Oracle:oracle-database-19-3:oracle-database-19-0904:19.3.1 \
     --size Standard_DS1_v2 \
     --generate-ssh-keys \
    --admin-username azureuser \
    --vnet-name ggVnet \
    --subnet ggSubnet1 \
    --public-ip-address "" \
    --nsg "" \
    --zone 1
```

Create ggVM2 (replicate)

Azure CLI

```
$ az vm create \
    --resource-group GoldenGateOnAzureLab \
    --name ggVM2 \
    --image Oracle:oracle-database-19-3:oracle-database-19-0904:19.3.1 \
    --size Standard_DS1_v2  \
    --generate-ssh-keys \
    --admin-username azureuser \
    --vnet-name ggVnet \
    --subnet ggSubnet1 \
    --public-ip-address "" \
    --nsg "" \
    --zone 2
```

Connect to ggVM1 (primary)

Connect to **ggVM1** using Bastion.

1. Navigate to **ggVM1** from Azure portal.
2. Go to **Overview** in the left pane.
3. Select **Connect** > **Bastion** on the menu at the top
4. Select Bastion tab
5. Click **Use Bastion**

Create the database on ggVM1 (primary)

The Oracle software is already installed on the Marketplace image, so the next step is to create the database.

1. Run the software as the `oracle` user.

```bash
$ sudo su - oracle
```

2. Create the database using following command. This command may take 30-40 minutes to complete.

```bash
$ dbca -silent \
   -createDatabase \
   -templateName General_Purpose.dbc \
   -gdbname cdb1 \
   -sid cdb1 \
   -responseFile NO_VALUE \
   -characterSet AL32UTF8 \
   -sysPassword OraPasswd1 \
   -systemPassword OraPasswd1 \
   -createAsContainerDatabase true \
   -numberOfPDBs 1 \
   -pdbName pdb1 \
   -pdbAdminPassword OraPasswd1 \
   -databaseType MULTIPURPOSE \
   -automaticMemoryManagement false \
   -storageType FS \
   -datafileDestination "/u01/app/oracle/oradata/" \
   -ignorePreReqs
```

Output should look similar to the following response.

Output

```
Copying database files
1% complete
2% complete
8% complete
13% complete
19% complete
27% complete
Creating and starting Oracle instance
29% complete
32% complete
33% complete
34% complete
38% complete
42% complete
43% complete
45% complete
Completing Database Creation
48% complete
51% complete
53% complete
62% complete
70% complete
72% complete
Creating Pluggable Databases
78% complete
100% complete
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/cdb1/cdb1.log" for
more details.
```

3. Set the ORACLE_SID and LD_LIBRARY_PATH variables.

Bash

```bash
$ export ORACLE_SID=cdb1
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

4. Run following to add ORACLE_SID and LD_LIBRARY_PATH to the .bashrc file, so that these settings are saved for future sign-ins. ORACLE_HOME variable should already be set in .bashrc file.

Bash

```bash
$ sed -i '$ a export ORACLE_SID=cdb1' .bashrc
$ sed -i '$ a export LD_LIBRARY_PATH=$ORACLE_HOME/lib' .bashrc
```

5. Start Oracle listener

```bash
$ lsnrctl start
```

Connect to ggVM2 (replicate)

Connect to **ggVM2** using Bastion.

1. Navigate to **ggVM2** from Azure portal.
2. Go to **Overview** in the left pane.
3. Select **Connect** > **Bastion** on the menu at the top
4. Select Bastion tab
5. Click **Use Bastion**

Open firewall ports for ggVM1

Configure firewall to allow connections from ggVM1. Following command is run on ggVM2.

```bash
$ sudo su -
$ firewall-cmd --permanent --zone=trusted --add-source=10.0.0.5
$ firewall-cmd --reload
$ exit
```

Create the database on ggVM2 (replicate)

1. Switch to `oracle` user if necessary

```bash
$ sudo su - oracle
```

2. Create the database

```bash
$ dbca -silent \
   -createDatabase \
   -templateName General_Purpose.dbc \
   -gdbname cdb1 \
   -sid cdb1 \
   -responseFile NO_VALUE \
```

```
    -characterSet AL32UTF8 \
    -sysPassword OraPasswd1 \
    -systemPassword OraPasswd1 \
    -createAsContainerDatabase true \
    -numberOfPDBs 1 \
    -pdbName pdb1 \
    -pdbAdminPassword OraPasswd1 \
    -databaseType MULTIPURPOSE \
    -automaticMemoryManagement false \
    -storageType FS \
    -datafileDestination "/u01/app/oracle/oradata/" \
    -ignorePreReqs
```

3. Set the ORACLE_SID and ORACLE_HOME variables.

Bash

```
$ export ORACLE_SID=cdb1
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

4. Run following to add ORACLE_SID and LD_LIBRARY_PATH to the .bashrc file, so that these settings are saved for future sign-ins. ORACLE_HOME variable should already be set in .bashrc file.

Bash

```
$ sed -i '$ a export ORACLE_SID=cdb1' .bashrc
$ sed -i '$ a export LD_LIBRARY_PATH=$ORACLE_HOME/lib' .bashrc
```

5. Start Oracle listener

Bash

```
$ lsnrctl start
```

Configure Golden Gate on ggVM1

Follow the steps in this section to install and configure Golden Gate.

Enable archive log mode on ggVM1 (primary)

1. Connect to `sqlplus`

Bash

```
$ sqlplus / as sysdba
```

2. Enable archive log

```PL/SQL
SQL> SELECT log_mode FROM v$database;

LOG_MODE
------------
NOARCHIVELOG

SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```

3. Enable force logging, and make sure at least one log file is present.

```PL/SQL
SQL> ALTER DATABASE FORCE LOGGING;
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> ALTER SYSTEM set enable_goldengate_replication=true;
SQL> ALTER PLUGGABLE DATABASE PDB1 OPEN;
SQL> ALTER SESSION SET CONTAINER=CDB$ROOT;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
SQL> ALTER SESSION SET CONTAINER=PDB1;
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
SQL> EXIT;
```

Download Golden Gate software

1. Download the **213000_fbo_ggs_Linux_x64_Oracle_shiphome.zip** file from the Oracle Golden Gate download page. Under the download title **Oracle GoldenGate 21.3.0.0.0 for Oracle on Linux x86-64**.

2. After you download the .zip file to your client computer, you can use Secure Copy Protocol (SCP) to copy the files to your virtual machines (ggVM1 and ggVM2). Make sure that `scp` command points to correct path of .zip file.

3. Copy to ggVM1

   a. Login and ensure you're using the correct subscription as necessary as described in Sign in to Azure

b. Open the tunnel to your target VM using the following PowerShell command

```PowerShell
$ggVM1id=$(az vm show --resource-group GoldenGateOnAzureLab --name ggVM1 --query 'id' --output tsv)
az network bastion tunnel --name ggBastion --resource-group GoldenGateOnAzureLab --target-resource-id $ggVM1id --resource-port 22 --port 57500
```

c. Leave the first command prompt running and open a second command prompt to connect to your target VM through the tunnel. In this second command prompt window, you can upload files from your local machine to your target VM using the following command. The correct `id_rsa` keyfile to access virtual machine must reside in `.ssh` directory or you can point to a different key file using `-i` parameter to `scp` command.

```PowerShell
 scp -P 57500 "213000_fbo_ggs_Linux_x64_Oracle_shiphome.zip" azureuser@127.0.0.1:.
```

4. Copy to ggVM2

a. Open the tunnel to your target VM using the following PowerShell command

```PowerShell
$ggVM2id=$(az vm show --resource-group GoldenGateOnAzureLab --name ggVM2 --query 'id' --output tsv)
az network bastion tunnel --name ggBastion --resource-group GoldenGateOnAzureLab --target-resource-id $ggVM2id --resource-port 22 --port 57501
```

b. Leave the first command prompt running and open a second command prompt to connect to your target VM through the tunnel. In this second command prompt window, you can upload files from your local machine to your target VM using the following command. The correct `id_rsa` keyfile to access virtual machine must reside in `.ssh` directory or you can point to a different key file using `-i` parameter to `scp` command.

```PowerShell
scp -P 57501 "213000_fbo_ggs_Linux_x64_Oracle_shiphome.zip"
```

```
azureuser@127.0.0.1:.
```

Install Golden Gate software

1. Connect to **ggVM1** using Bastion.
   a. Navigate to **ggVM1** from Azure portal.
   b. Go to **Overview** in the left panel
   c. Select **Connect** > **Bastion** on the menu at the top
   d. Select Bastion tab
   e. Click **Use Bastion**

2. Move the .zip file to the **/opt** folder, then change the owner

```bash
$ sudo su -
$ mv /home/azureuser/*.zip /opt
```

3. Unzip the files (install unzip utility if it's not already installed)

```bash
$ yum install unzip
$ cd /opt
$ unzip 213000_fbo_ggs_Linux_x64_Oracle_shiphome.zip
```

4. Change permission

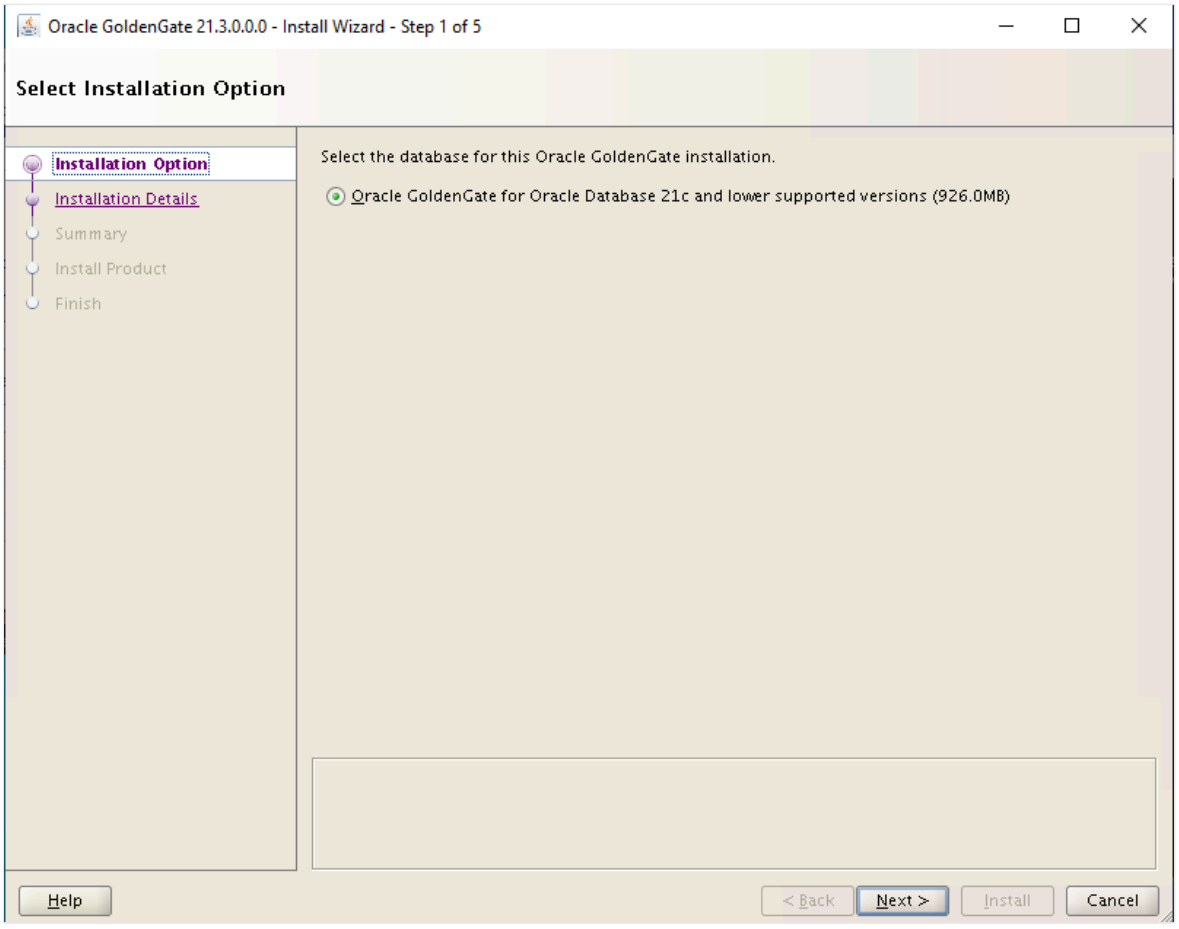
```bash
$ chown -R oracle:oinstall /opt/fbo_ggs_Linux_x64_Oracle_shiphome
$ exit
```

5. Start Golden Gate installer

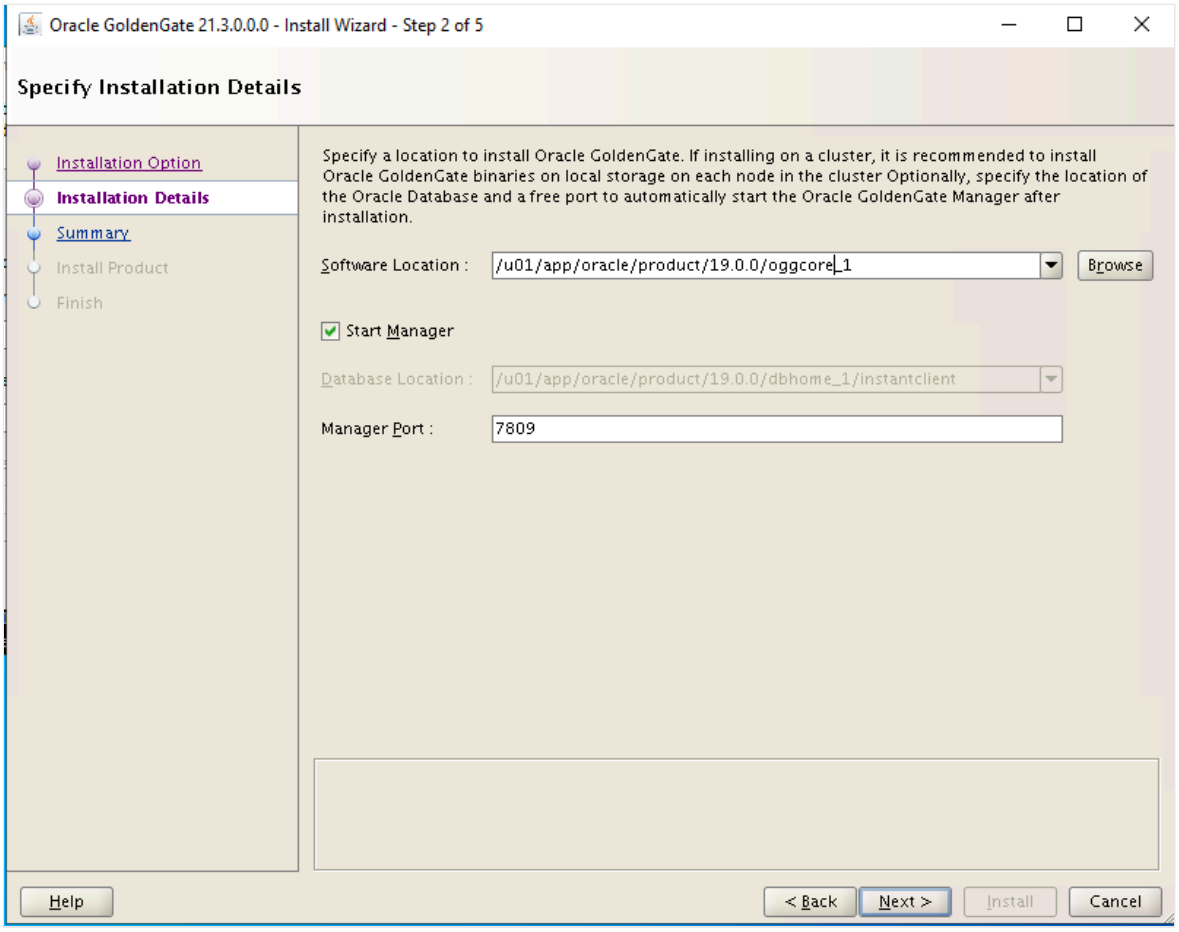
```bash
$ sudo su - oracle
$ export DISPLAY=10.0.0.4:0.0
$ cd /opt/fbo_ggs_Linux_x64_Oracle_shiphome/Disk1
$ ./runInstaller
```

The installer opens on **ggXServer**.

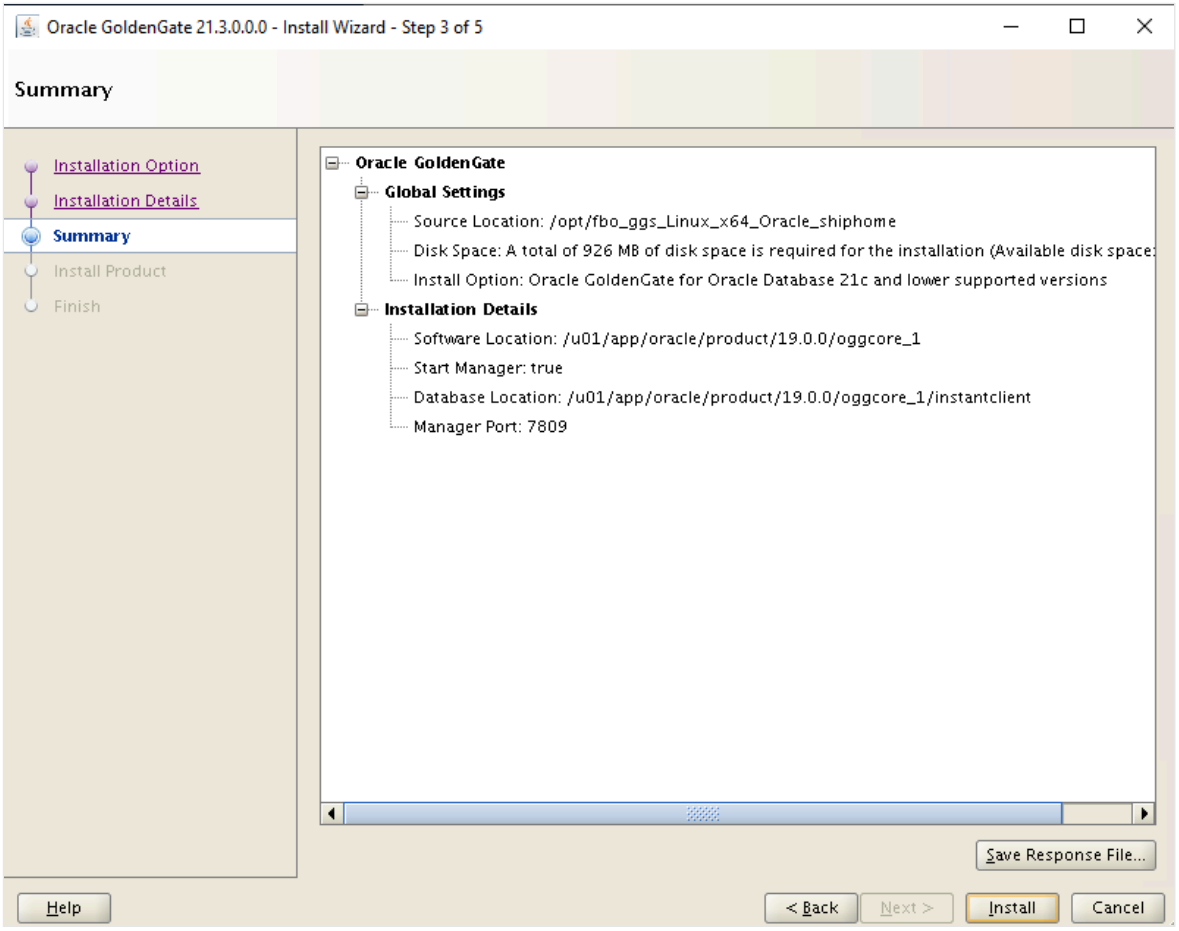
6. Select 'Oracle GoldenGate for Oracle Database 21c'. Then select **Next** to continue.



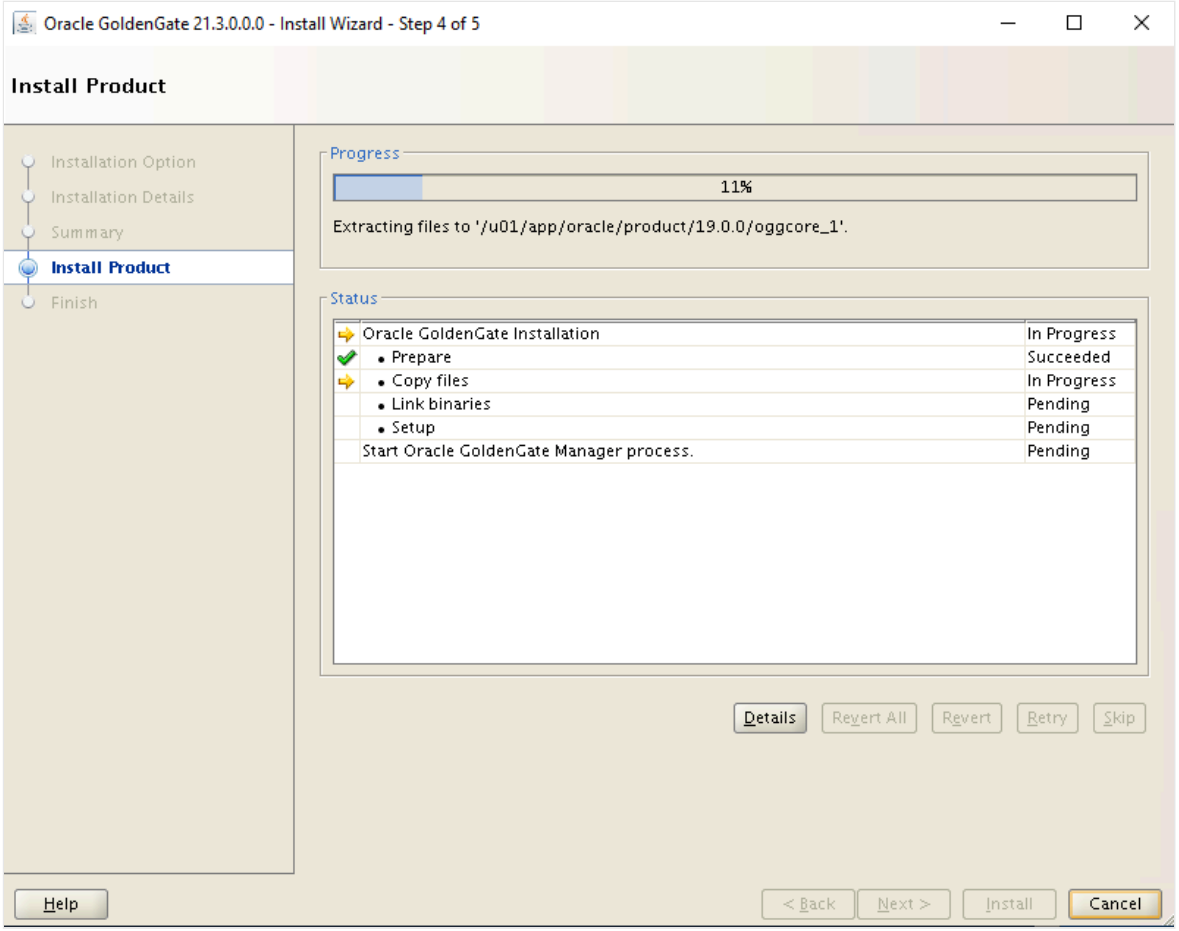
7. Set the software location to **/u01/app/oracle/product/19.0.0/oggcore_1**, make sure **Start Manager** box is selected and select **Next** to continue.



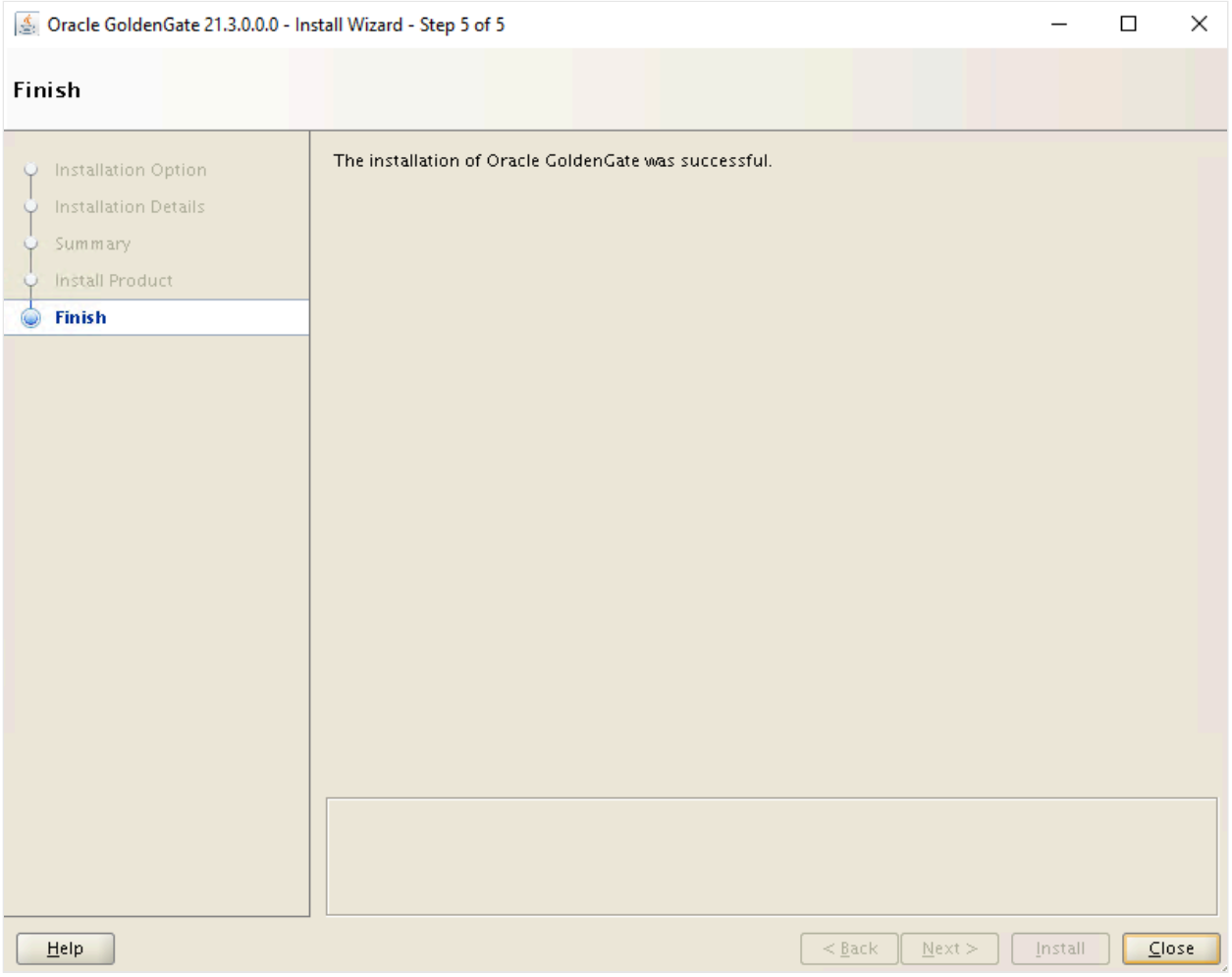
8. On the **Summary** screen, select **Install** to continue.



9. Wait for installation to finish.



10. Select **Close** to continue.



11. Connect to **ggVM2** using Bastion.
    a. Navigate to **ggVM2** from Azure portal.
    b. Go to **Overview** in the left pane
    c. Select **Connect** > **Bastion** on the menu at the top
    d. Select Bastion tab
    e. Click **Use Bastion**

12. Repeat steps 2-10 above on **ggVM2**

Set up service on ggVM1 (primary)

1. Create or update the tnsnames.ora file.

```bash
$ sudo su - oracle
$ cd $ORACLE_HOME/network/admin
$ vi tnsnames.ora
```

When vi editor opens you have to press `i` to switch to insert mode, then copy and paste file contents and press `Esc` key, `:wq!` to save file.

```text
cdb1=
 (DESCRIPTION=
   (ADDRESS=
     (PROTOCOL=TCP)
     (HOST=localhost)
     (PORT=1521)
   )
   (CONNECT_DATA=
     (SERVER=dedicated)
     (SERVICE_NAME=cdb1)
   )
 )

pdb1=
 (DESCRIPTION=
   (ADDRESS=
     (PROTOCOL=TCP)
     (HOST=localhost)
     (PORT=1521)
   )
   (CONNECT_DATA=
     (SERVER=dedicated)
     (SERVICE_NAME=pdb1)
```

```
    )
  )
```

2. Create the Golden Gate owner and user accounts.

> ⓘ **Note**
>
> The owner account must have C## prefix.

Bash

```
$ sqlplus / as sysdba
```

PL/SQL

```
 SQL> CREATE USER C##GGADMIN identified by ggadmin;
 SQL> EXEC
dbms_goldengate_auth.grant_admin_privilege('C##GGADMIN',container=>'ALL
');
 SQL> GRANT DBA to C##GGADMIN container=all;
 SQL> connect C##GGADMIN/ggadmin
 SQL> ALTER SESSION SET CONTAINER=PDB1;
 SQL> EXIT;
```

3. Create the Golden Gate test user account

Bash

```
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ sqlplus system/OraPasswd1@pdb1
```

PL/SQL

```
SQL> CREATE USER test identified by test DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP;
SQL> GRANT connect, resource, dba TO test;
SQL> ALTER USER test QUOTA 100M on USERS;
SQL> connect test/test@pdb1
SQL> @demo_ora_create
SQL> @demo_ora_insert
SQL> EXIT;
```

4. Configure the extract parameter file.

   Start the Golden gate command-line interface (ggsci):

Bash

```bash
$ sudo su - oracle
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ ./ggsci
```

PL/SQL

```
GGSCI> DBLOGIN USERID test@pdb1, PASSWORD test
Successfully logged into database  pdb1
GGSCI>  ADD SCHEMATRANDATA pdb1.test
2017-05-23 15:44:25  INFO    OGG-01788  SCHEMATRANDATA has been added
on schema test.
2017-05-23 15:44:25  INFO    OGG-01976  SCHEMATRANDATA for scheduling
columns has been added on schema test.

GGSCI> EDIT PARAMS EXTORA
```

5. Press `i` key to switch to insert mode and copy and paste (right click on the SSH window) following to the EXTRACT parameter file. Press Esc key, `:wq!` to save file.

PL/SQL

```
EXTRACT EXTORA
USERID C##GGADMIN@cdb1, PASSWORD ggadmin
RMTHOST 10.0.0.5, MGRPORT 7809
RMTTRAIL ./dirdat/rt
DDL INCLUDE MAPPED
DDLOPTIONS REPORT
LOGALLSUPCOLS
UPDATERECORDFORMAT COMPACT
TABLE pdb1.test.TCUSTMER;
TABLE pdb1.test.TCUSTORD;
```

6. Register extract--integrated extract.

PL/SQL

```
GGSCI> dblogin userid C##GGADMIN@cdb1, password ggadmin
Successfully logged into database CDB$ROOT.

GGSCI> REGISTER EXTRACT EXTORA DATABASE CONTAINER(pdb1)

2023-03-03 19:37:01  INFO    OGG-02003  Extract group EXTORA
successfully registered with database at SCN 2142014.

GGSCI> exit
```

7. Set up extract checkpoints and start real-time extract

Bash

```
$ ./ggsci
```

PL/SQL

```
GGSCI>  ADD EXTRACT EXTORA, INTEGRATED TRANLOG, BEGIN NOW
EXTRACT (Integrated) added.

GGSCI>  ADD RMTTRAIL ./dirdat/rt, EXTRACT EXTORA, MEGABYTES 10
RMTTRAIL added.

GGSCI>  START EXTRACT EXTORA

Sending START request to MANAGER ...
EXTRACT EXTORA starting

GGSCI > INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EXTORA      00:00:11      00:00:04

GGSCI > EXIT
```

In this step, you find the starting SCN, which is used later, in a different section.

Bash

```
$ sqlplus / as sysdba
```

PL/SQL

```
SQL> alter session set container = pdb1;
SQL> SELECT current_scn from v$database;
CURRENT_SCN
-----------
   2172191
SQL> EXIT;
```

Bash

```
$ ./ggsci
GGSCI> EDIT PARAMS INITEXT
```

When vi editor opens you have to press `i` to switch to insert mode, then copy and paste file contents and press `Esc` key, `:wq!` to save file.

```bash
EXTRACT INITEXT
USERID C##GGADMIN@cdb1, PASSWORD ggadmin
RMTHOST 10.0.0.6, MGRPORT 7809
RMTTASK REPLICAT, GROUP INITREP
TABLE pdb1.test.*, SQLPREDICATE 'AS OF SCN 2172191';
```

```bash
GGSCI> ADD EXTRACT INITEXT, SOURCEISTABLE
Extract added.
GGSCI> EXIT
```

Set up service on ggVM2 (replicate)

1. Create or update the tnsnames.ora file.

```bash
$ cd $ORACLE_HOME/network/admin
$ vi tnsnames.ora
```

When vi editor opens you have to press `i` to switch to insert mode, then copy and paste file contents and press `Esc` key, `:wq!` to save file.

```PL/SQL
cdb1=
 (DESCRIPTION=
   (ADDRESS=
     (PROTOCOL=TCP)
     (HOST=localhost)
     (PORT=1521)
   )
   (CONNECT_DATA=
     (SERVER=dedicated)
     (SERVICE_NAME=cdb1)
   )
 )

pdb1=
 (DESCRIPTION=
   (ADDRESS=
```

```
      (PROTOCOL=TCP)
      (HOST=localhost)
      (PORT=1521)
    )
    (CONNECT_DATA=
      (SERVER=dedicated)
      (SERVICE_NAME=pdb1)
    )
  )
```

2. Create a replicate account.

Bash

```
$ sqlplus / as sysdba
```

PL/SQL

```
SQL> ALTER SYSTEM set enable_goldengate_replication=true;
SQL> ALTER SESSION SET CONTAINER = pdb1;
SQL> CREATE USER REPUSER IDENTIFIED BY REP_PASS CONTAINER=CURRENT;
SQL> GRANT DBA TO REPUSER;
SQL> EXEC
DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('REPUSER',CONTAINER=>'PDB1')
;
SQL> CONNECT REPUSER/REP_PASS@PDB1
SQL> EXIT;
```

3. Create a Golden Gate test user account.

Bash

```
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ sqlplus system/OraPasswd1@pdb1
```

PL/SQL

```
SQL> CREATE USER test identified by test DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP;
SQL> GRANT connect, resource, dba TO test;
SQL> ALTER USER test QUOTA 100M on USERS;
SQL> CONNECT test/test@pdb1
SQL> @demo_ora_create
SQL> EXIT;
```

4. REPLICAT parameter file to replicate changes.

```bash
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ ./ggsci
```

```
GGSCI> EDIT PARAMS REPORA
```

When vi editor opens you have to press `i` to switch to insert mode, then copy and paste file contents and press `Esc` key, `:wq!` to save file.

```bash
REPLICAT REPORA
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/repora.dsc, PURGE, MEGABYTES 100
DDL INCLUDE MAPPED
DDLOPTIONS REPORT
DBOPTIONS INTEGRATEDPARAMS(parallelism 6)
USERID repuser@pdb1, PASSWORD REP_PASS
MAP pdb1.test.*, TARGET pdb1.test.*;
```

5. Set up a replicate checkpoint.

```bash
GGSCI> ADD REPLICAT REPORA, INTEGRATED, EXTTRAIL ./dirdat/rt
GGSCI> EDIT PARAMS INITREP
```

When vi editor opens you have to press `i` to switch to insert mode, then copy and paste file contents and press `Esc` key, `:wq!` to save file.

```bash
REPLICAT INITREP
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/tcustmer.dsc, APPEND
USERID repuser@pdb1, PASSWORD REP_PASS
MAP pdb1.test.*, TARGET pdb1.test.*;
```

```bash
GGSCI> ADD REPLICAT INITREP, SPECIALRUN
GGSCI> EXIT
```

Set up replication (ggVM1 and ggVM2)

1. Set up the replication on ggVM2 (replicate)

Logon to `ggsci`.

```bash
$ cd /u01/app/oracle/product/12.1.0/oggcore_1
$ ./ggsci
```

```bash
GGSCI> EDIT PARAMS MGR
```

Update the file with the following.

```bash
PORT 7809
ACCESSRULE, PROG *, IPADDR *, ALLOW
```

Restart the Manager service.

```bash
GGSCI> STOP MGR
GGSCI> START MGR
GGSCI> EXIT
```

2. Set up the replication on ggVM1 (primary)

Start the initial load and check for errors.

```bash
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ ./ggsci
```

```bash
GGSCI> START EXTRACT INITEXT
```

```
GGSCI> VIEW REPORT INITEXT
```

3. Set up the replication on ggVM2 (replicate)

Change the SCN number with the number you obtained before.

```bash
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ ./ggsci
```

```bash
START REPLICAT REPORA, AFTERCSN 2172191
```

The replication has begun, and you can test it by inserting new records to TEST tables.

View job status and troubleshooting

View reports

- To view reports on **ggVM1**, run the following commands.

  ```bash
  GGSCI> VIEW REPORT EXTORA
  ```

- To view reports on **ggVM2**, run the following commands.

  ```bash
  GGSCI> VIEW REPORT REPORA
  ```

View status and history

- To view status and history on **ggVM1**, run the following commands.

  ```bash
  GGSCI> DBLOGIN USERID C##GGADMIN@CDB1, PASSWORD ggadmin
  GGSCI> INFO EXTRACT EXTORA, DETAIL
  ```

- To view status and history on **ggVM2**, run the following commands.

```bash
GGSCI> DBLOGIN USERID REPUSER@PDB1 PASSWORD REP_PASS
GGSCI> INFO REP REPORA, DETAIL
```

```bash
$ sqlplus test/test@pdb1
```

```PL/SQL
SQL> select * from TCUSTMER;
SQL> select * from TCUSTORD;
```

Observe Data Replication

1. Connect to primary database on **ggVM1**

```bash
$ sqlplus test/test@pdb1
```

2. Select existing records from table

```PL/SQL
SQL> select * from TCUSTORD;
```

3. Create a test record

```PL/SQL
SQL> INSERT INTO TCUSTORD VALUES ('OLLJ',TO_DATE('11-APR-
99'),'CYCLE',400,16000,1,102);
SQL> COMMIT;
SQL> EXIT;
```

4. Observe transaction picked up by Golden Gate (Note **Total inserts** value)

```bash
```

```
$ cd /u01/app/oracle/product/19.0.0/oggcore_1
$ ./ggsci
```

Bash

```
GGSCI> STATS EXTORA
```

Output

```
  Sending STATS request to Extract group EXTORA ...

  Start of statistics at 2023-03-24 19:41:54.

  DDL replication statistics (for all trails):

  *** Total statistics since extract started     ***
  Operations                            0.00
  Mapped operations                     0.00
  Unmapped operations                     0.00
  Other operations                    0.00
  Excluded operations                     0.00

  Output to ./dirdat/rt:

  Extracting from PDB1.TEST.TCUSTORD to PDB1.TEST.TCUSTORD:

  *** Total statistics since 2023-03-24 19:41:34 ***
     Total inserts                                1.00
     Total updates                                0.00
     Total deletes                                0.00
     Total upserts                                0.00
     Total discards                               0.00
     Total operations                             1.00

  *** Daily statistics since 2023-03-24 19:41:34 ***
     Total inserts                                1.00
     Total updates                                0.00
     Total deletes                                0.00
     Total upserts                                0.00
     Total discards                               0.00
     Total operations                             1.00

  *** Hourly statistics since 2023-03-24 19:41:34 ***
     Total inserts                                1.00
     Total updates                                0.00
     Total deletes                                0.00
     Total upserts                                0.00
     Total discards                               0.00
     Total operations                             1.00

  *** Latest statistics since 2023-03-24 19:41:34 ***
```

```
    Total inserts                                 1.00
    Total updates                                 0.00
    Total deletes                                 0.00
    Total upserts                                 0.00
    Total discards                                0.00
    Total operations                              1.00

  End of statistics.
```

5. Connect to primary database on **ggVM2**

```
Bash

$ sqlplus test/test@pdb1
```

6. Verify new record is replicated

```
PL/SQL

SQL> select * from TCUSTORD;
```

```
PL/SQL

 CUST ORDER_DAT PRODUCT_   ORDER_ID PRODUCT_PRICE PRODUCT_AMOUNT
TRANSACTION_ID
 ---- --------- -------- ---------- ------------- --------------- ------
--------
 OLLJ 11-APR-99 CYCLE       400   16000      1    102
 WILL 30-SEP-94 CAR       144   17520      3    100
 JANE 11-NOV-95 PLANE       256  133300      1    100
```

The installation and configuration of Golden Gate on Oracle linux is now complete.

Delete the ggXServer VM

ggXServer VM is only used during setup. You can safely delete it after completing this lab document but keep your GoldenGate on Azure lab setup intact.

```
Azure CLI

$ az vm delete --resource-group GoldenGateOnAzureLab --name ggXServer --
force-deletion yes

$ az network public-ip delete --resource-group GoldenGateOnAzureLab --name
ggXServerPublicIP
```

Delete Golden Gate On Azure Lab Setup

When it's no longer needed, the following command can be used to remove the resource group, VM, and all related resources for this lab.

Azure CLI

```
$ az group delete --name GoldenGateOnAzureLab
```

Next steps

Create highly available virtual machines tutorial

Explore VM deployment CLI samples

Feedback

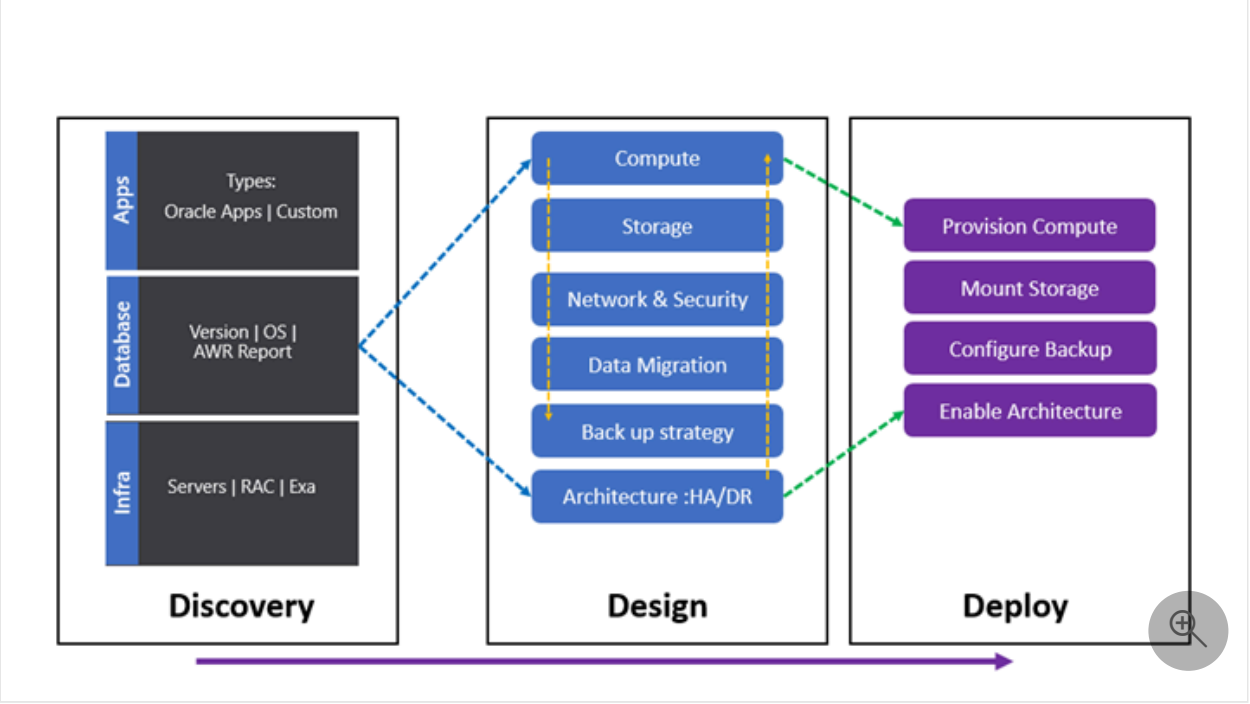
Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Migrate Oracle workloads to Azure VMs

Article • 08/22/2024

This article shows how to move your Oracle workload from your on-premises environment to the Azure virtual machines (VMs) landing zone. It uses the Landing zone for Oracle Database at Azure, which offers design advice and best practices for Oracle migration on Azure IaaS. A proven discovery, design, and deployment approach are recommended for the overall migration strategy, followed by data migration, and cut over.



Discovery

Migration begins with a detailed assessment of the Oracle product portfolio. The current infrastructure that supports Oracle database and apps, database versions, and types of applications that use Oracle database are: Oracle (EBS, Siebel, People Soft, JDE, and others) and non-Microsoft partner offerings like SAP or custom applications. The existing Oracle database can operate on servers, Oracle Real Application Clusters (RAC), or non-Microsoft partner RAC. For applications, we need to discover size of infrastructure that can be done easily by using Azure Migrate based discovery. For database, the approach is to get allowed with restrictions Automatic Workload repository (AWR) reports on peak load to move on to design steps.

Design

For Applications, Azure Migrate do lift and shift infrastructure and applications to Azure IaaS based on discovery. For Oracle first party applications, refer to the architecture requirements before deciding on Azure Migrate based migration. Database design begins with generated AWR reports on peak load. Once AWRs are in place, run Azure Oracle Migration Assistance Tool (OMAT) with AWR reports as input. The OMAT tool recommends the correct VM size and storage options required for your Oracle Database on Azure IaaS. The solution must have high reliability and resilience in the occurrence of disasters, as determined by the parameters of Recovery Point Objective (RPO) and Recovery Time Objective (RTO). Oracle landing zone offers architecture guidance to choose the best solution architecture based on RPO and RTO requirements. The RPO and RTO approach is applicable for separating RAC infrastructure into high availability (HA) and disaster recovery (DR) architecture using Oracle data guard.

Deployment

The OMAT tool analyzes the AWR report to provide you with information on the required infrastructure; correct VM size and recommendations on storage with capacity. Based on that information, select the suitable HA and DR (RPO/RTO) requirement to provide resilient architecture that provides Business Continuity and Disaster Recovery (BCDR) using Oracle on Azure landing zone. Use Ansible to describe the infrastructure and architecture as infrastructure as code (IaC) and launch the landing zone with either Terraform or Bicep. Use the GitHub actions available to automate the deployment.

Types for data migration

The data migration process has two types, online and offline. Online transfers data from source to destination as it happens. Offline extracts data from source and transfers it to destination afterwards. Both methods are essential. Offline is suitable for transferring large data between source and destination, while online can transfer incremental data before shifting from source to destination database. Both types of approach used together can provide an efficient solution for successful data migration.

Data migration approach

After you set up Oracle on Azure infrastructure, install Oracle database, and migrate related applications; the next step is to transfer data from on premise Oracle database to the new Oracle database on Azure. See the following Oracle tools:

- Recovery Manager (RMAN)
- Data Pump

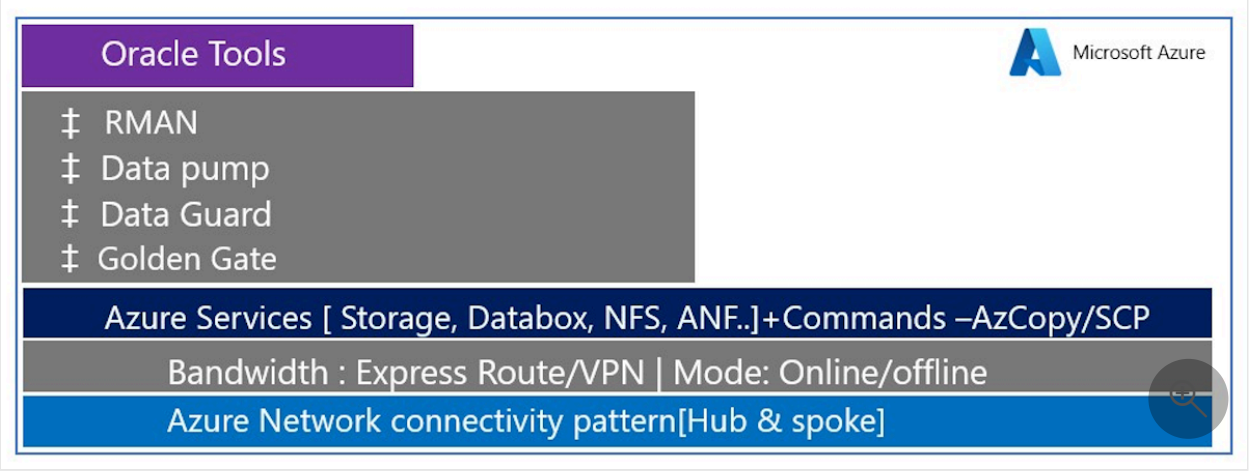
- Data Guard
- GoldenGate

Azure enhances the Oracle tools with the right network connectivity, bandwidth, and commands that are powered by the following Azure capabilities for data migration.

- VPN Connectivity
- Express Route
- AzCopy
- Data Box

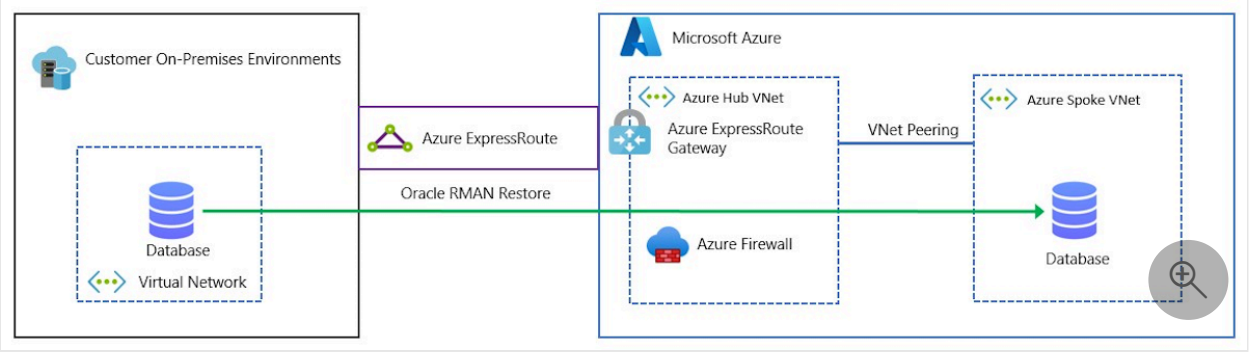
Oracle tools for data migration

The following diagram is a pictographic representation of the overall migration portfolio.

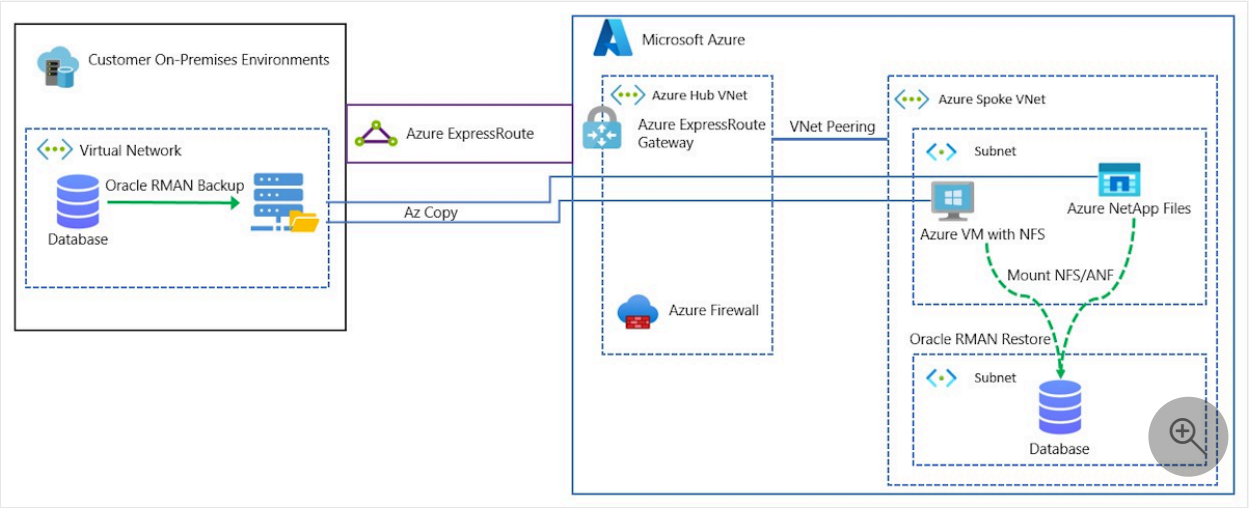


You need one of the Oracle Tools plus Azure infrastructures to deploy the correct solution architecture to migrate data. See the following reference solution scenarios:

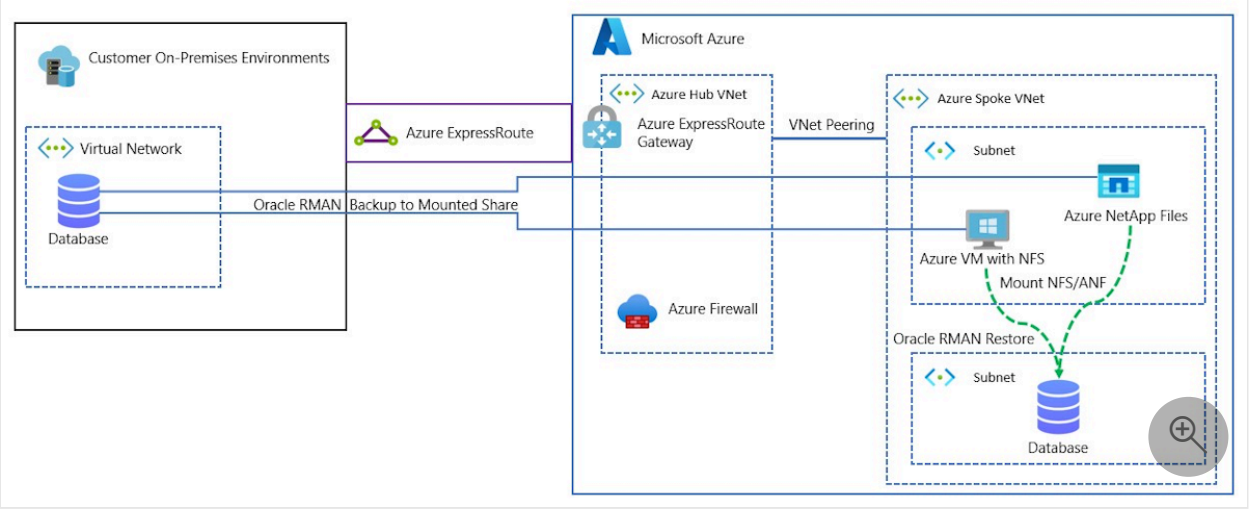
Scenario-1: RMAN: Use RMAN backup and restore with Azure features, the setup for RMAN based recovery. The main thing is the network between on-premises and Azure.



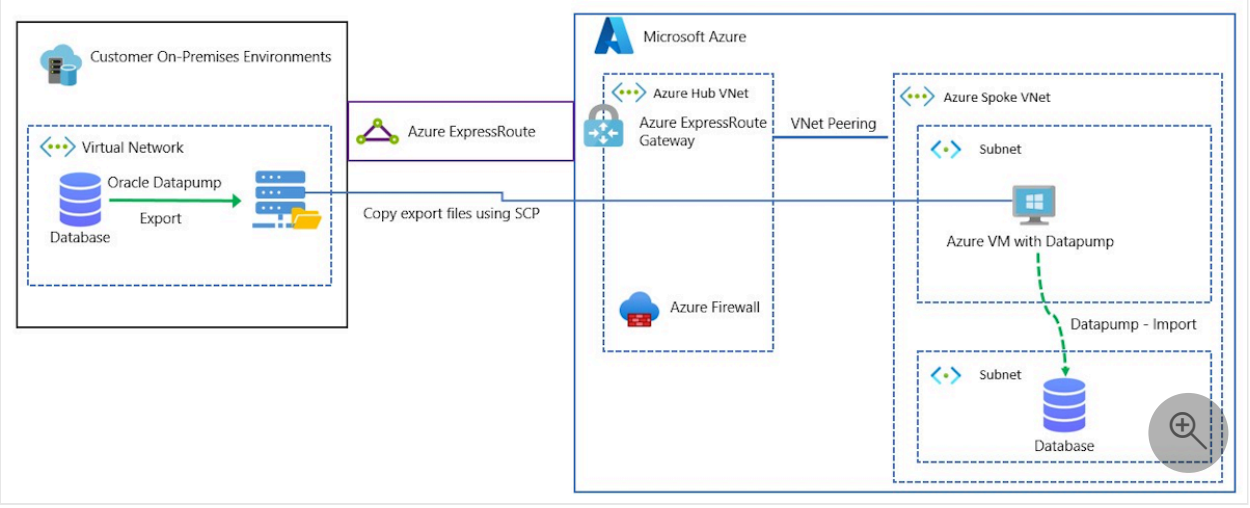
Scenario-2: RMAN Backup Approach



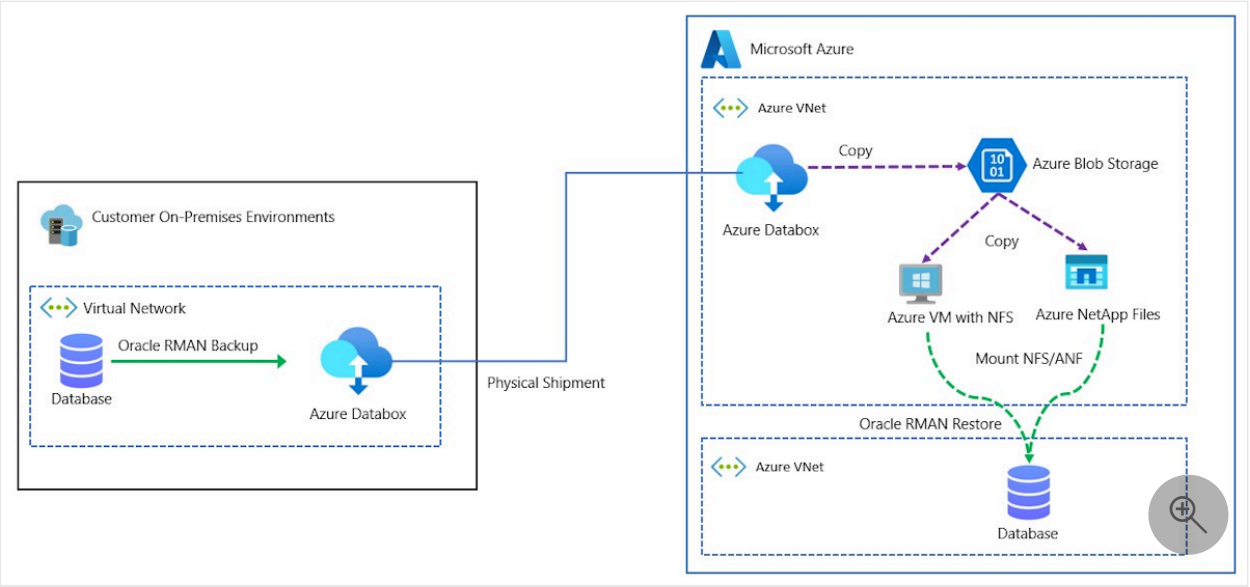
Scenario-3: Alternatively, setup can be modified in multiple different ways as depicted in the following scenario.



Scenario-4: Data PumpàAzCopy - easy and straight forward approach using Data Pump backup and restore using Azure capabilities.



Scenario-5: Data Box - a unique scenario in which data is moved between the locations using a storage device and physical shipment.



Cutover

Now your data is migrated and Oracle database servers and applications are up and running. In this section, use the following steps to transition business operations running on premise over to newfound Oracle workload and applications on Azure IaaS.

1. Schedule a maintenance window to minimize disruption to users.
2. Stop database activity on the source Oracle database.
3. Perform a final data synchronization to verify all changes are captured.
4. Update DNS configurations to point to the new Azure VM.
5. Start the Oracle database on the Azure VM and verify connectivity.
6. Monitor the system closely for any issues during the cutover process.

Post migration tasks

After the cutover, verify all business applications are functioning as expected to deliver business operations in tandem with on premise.

- Perform validation checks to verify data consistency and application functionality.
- Update documentation, including: network diagrams, configuration details, and disaster recovery plans.
- Implement ongoing monitoring and maintenance processes for Azure VM hosting the Oracle database.

Throughout the migration process, it's essential to communicate effectively with stakeholders, including application owners, IT operations teams, and end-users, to manage expectations and minimize disruption. Additionally, consider engaging with

experienced professionals or consulting services specializing in Oracle-to-Azure migrations to ensure a smooth and successful transition.

Next steps

Storage options for Oracle on Azure VMs

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Performance best practices for Oracle on Azure VMs

Article • 08/22/2024

This article describes how the right VM size and storage options you choose affects your Oracle workload performance - input/output operations (IOPS) and throughput - dependability, and cost. There's a trade-off between optimizing for costs and for performance. This performance best practices series is focused on getting the best performance for the Oracle workload on Azure VMs. If your workload is less demanding, you might not require every optimization recommended. It's critical in the planning phase, to assess the performance requirements of your Oracle workloads and right size the compute and storage as needed.

When considering to run Oracle workloads on Azure VMs, for a cost-effective configuration start by selecting a virtual machine that supports the necessary IOPS and throughput with the appropriate memory-to-vCore ratio and then add your storage requirement.

VM sizing recommendations

The following three VM series are the recommended to run Oracle database workloads on Azure.

E-series (Eds v5 and Ebds V5)

The E-series is designed for memory-intensive workloads. These VMs provide high memory-to-core ratios, making them suitable for Oracle databases. Also offer a range of CPU options to match the performance requirements of your Oracle database workload.

The new Ebdsv5-series provides the highest I/O throughput-to-vCore ratio in Azure along with a memory-to-vCore ratio of 8. This series offers the best price-performance for Oracle workloads on Azure VMs. Consider this series first for most Oracle database workloads.

M-series

The M-series is built for large databases, that is, up to 12-TB RAM and 416vCPUs. The M series VMs offer the highest memory-to-vCore ratio in Azure. Consider these VMs for

large and large mission critical Oracle database workloads or if you would need to consolidate databases into fewer VMs.

D-series

The D-series is built for general purpose VMs with smaller memory-to-vCore ratios with the General-Purpose virtual machines. It's important to carefully monitor memory-based performance counters to ensure Oracle workload can get the IOPS & through put. The Ddsv5-series offers a fair combination of vCPU, memory, and temporary disk but with smaller memory-to-vCore support. D-series doesn't have the memory-to-vCore ratio of 8 that is recommended for Oracle workloads. As such, consider using these virtual machines for small to medium databases or for dev/test environment for lower TCO.

Storage recommendations

This section provides storage best practices and guidelines to optimize performance for your Oracle workload on Azure Virtual Machines (VM). Consider your performance needs, costs, and workload patterns as you evaluate these recommendations. Let us take a quick look at the options:

- Disk Types: Premium SSD, Premium SSD V2 & Ultra disks are recommended disk types for Oracle workload. Refer to disk type comparison to understand maximum disk size, maximum throughput and maximum IOPS to choose right disk type for Azure VM to meet your Oracle workload performance. Generally, Premium SSD v2 is the best price per performance disk option that you could consider.

- Premium SSD V2 offers higher performance than Premium SSDs while also generally being less costly. You can individually tweak the performance (capacity, throughput, and IOPS) of Premium SSD v2 disks at any time, allowing workloads to be cost efficient while meeting shifting performance needs. For example, a transaction-intensive database needs a large amount of IOPS at a small size, or a gaming application can require a large amount of IOPS but only during peak hours. Because you can individually tweak the performance, for most general-purpose workloads, Premium SSD v2 can provide the best price performance.

- Premium SSDs are suitable for mission-critical production workloads. They deliver high-performance and low-latency disk support for virtual machines (VMs) with input/output (IO)-intensive workloads.

- Ultra disks are the highest-performing storage option for Azure virtual machines (VMs). They're suitable for data-intensive and transaction-heavy workloads. They

provide low sub millisecond latencies and feature a flexible performance configuration model that allows you to independently configure IOPS and throughput, before and after you provision the disk.

Azure Elastic SAN delivers a massively scalable, cost-effective, highly performant, and reliable block storage solution that connects to various Azure compute services over iSCSI protocol. Elastic SAN enables a seamless transition from an existing SAN storage estate to the cloud without having to refactor customer application architecture. This solution can achieve massive scale - up to millions of IOPS, double-digit GB/s of throughput, and low single-digit millisecond latencies with built-in resiliency to minimize downtime. This makes it a great fit for customers looking to consolidate storage, customers working with multiple compute services, or those who have workloads that require high throughput levels achieved by driving storage over network bandwidth.

> **Note**
>
> VM sizing with Elastic SAN should accommodate production (VM to VM) network throughput requirements along with storage throughput.

Consider placing Oracle workloads on Elastic SAN for better cost efficiency for the following reasons.

- **Storage consolidation and dynamic performance sharing**: Normally for Oracle workload on Azure VM, disk type storage is provisioned on a per VM basis based on customer's capacity and peak performance requirements for that VM. This overprovisioned performance is available when needed but the unused performance can't be shared with workloads on other VMs. Elastic SAN, like on-premises SAN, allows consolidating storage needs of multiple Oracle workloads to achieve better cost efficiency, with the ability to dynamically share provisioned performance across the volumes provisioned to these different workloads based on IO demands. For example, in East US, if you have 10 workloads that require 2-TiB capacity and 10K IOPS each, but collectively they don't need more than 60 K IOPS at any point in time. You can configure an Elastic SAN with 12 base units (one base unit = $0.08 per GiB/month) that will give you 12 TiB capacity and the needed 60K IOPS, and 8 capacity-only units (1 capacity-only unit = $0.06 per GiB/month) that will give you the remaining 8-TiB capacity at a cheaper price. This optimal storage configuration provides better cost efficiency while providing the necessary performance (10K IOPS) to each of these workloads. For more information on Elastic SAN base and capacity-only provisioning units, see Planning for an Azure Elastic SAN and for pricing, see Azure Elastic SAN - Pricing .

- **To drive higher storage throughput**: Oracle Workload on Azure VM deployments occasionally require overprovisioning a VM due disk throughput limit for that VM. You can avoid this with Elastic SAN, since you drive higher storage throughput over compute network bandwidth with the iSCSI protocol. For example, a Standard_E32bds_v5 (SCSI) VM is capped at 88,000 IOPS and 2,500 MBps for disk/storage throughput, but it can achieve up to a maximum of 16,000-MBps network throughput. If the storage throughput requirement for your workload is greater than 2,500 MBps, you won't have to upgrade the VM a higher SKU since it can now support up to 16,000 MBps by using Elastic SAN.

Additionally, the following are some inputs can help you to derive further value from Elastic SAN.

Expand table

Other parameters	description
Provisioning Model	Flexible model at TiB granularity
BCDR	Incremental snapshot for fast restore; Snapshot export for hardening.
Redundancy & Scale Targets	Refer redundancy capabilities of Azure Elastic SAN in redundancy requirements.
Encryption	Encryption at rest is supported.

Azure NetApp Files is an Azure native, first-party, enterprise-class, high-performance file storage service suitable for storing Oracle database files. It provides Volumes as a service for which you can create NetApp accounts, capacity pools, and volumes. You can also select service and performance levels and manage data protection. By using the same protocols and tools that you know and trust, and enterprise applications that depend on on-premises, you can build and maintain file shares that are fast, reliable, and scalable.

The following are key attributes of Azure NetApp files:

- Performance, cost optimization, and scale.
- Simplicity and availability.
- Data management and security.
- SLA 99.99%

Azure NetApp Files volumes are highly available by design and provide flexibility for scaling volumes up and down in capacity and performance without service interruption. For other availability across zones and regions volumes can be replicated using cross-zone and cross-region replication.

For hosting extremely demanding Oracle database files, redo logs and archive logs that scale well into multiple gigabytes per second throughput and multiple tens of terabytes capacity, you can utilize single or multiple volumes, depending on capacity and performance requirements. Volumes can be protected using snapshots for fast primary data protection and recoverability, and can be backed up using RMAN, AzAcSnap, Azure NetApp Files backup or other preferred backup methods or applications.

It's highly recommended to use Oracle direct NFS (dNFS) with Azure NetApp Files for enhanced performance. The combination of Oracle dNFS with Azure NetApp Files provides great advantage to your workloads. Oracle dNFS makes it possible to drive higher performance than the operating system's kernel NFS. The article explains the technology and provides a performance comparison between dNFS and the kernel NFS client.
Azure VMs are throttled for network traffic at higher speeds than direct attached storage such as SSD. As a result, the Oracle deployment performs better using Azure NetApp Files volumes at the same VM SKU, or you can choose a smaller VM SKU for the same performance and save on Oracle license cost.

Snapshots can be cloned to provide read/write access to current data for test and development purposes without interacting with the live data.

Expand table

Item	Description
Other parameter	Available in three performance service levels (Ultra, Premium, Standard) with dynamic interruption-free up- and down scaling of performance and capacity to balance changing requirements and cost
Provisioning model	Single volume for medium to large databases Multiple volumes for extremely large and high throughput Provisioning through Azure portal with online dynamic up-and downsizing Dynamic online performance scaling through dynamic service level changes and QoS adjustments
BDR	Snapshot-based independent data access for BC/DR and test/dev purposes Vaulting of snapshots with Azure NetApp Files backup Storage-based cross-region replication Storage-based cross-zone replication Integration with Oracle Data Guard for high availability and disaster recovery
Redundancy & scale targets	Demonstrated capability to support largest and highest performing Oracle databases over 100TiB in size and multiple gigabytes per second throughput while maintaining near-instantaneous snapshot-based primary data protection and recoverability
Encryption	Single or double encryption at rest with platform- or customer-managed keys

Automate VMs and storage selection

Consider using Community tool Oracle Migration Assistant Tool (OMAT) to get the right VM SKUs with recommended storage options including disk types, Elastic SAN & ANF with indicative cost based on list price. You can provide AWR report of the Oracle database as a input and run the OMT tool script to get an output of the recommended VM SKUs and storage options that aligns with the performance requirements of the database and is cost effective.

Next steps

- Migrate Oracle workload to Azure VMs (IaaS)
- Partner storage offerings for Oracle on Azure VMs

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Partner storage offerings for Oracle on Azure VMs

Article • 08/22/2024

This article describes Partner storage offerings for high performance - input/output operations (IOPS) and throughput - Oracle workloads on Azure virtual machines (VMs). While Microsoft first-party storage offerings for migrating Oracle workloads to Azure VMs are effective, there are use cases that require performance beyond the capacity of the first-party storage offering for Oracle on Azure VMs. These trusted third-party storage solutions are ideal for high performance use cases.

Oracle as a DBaaS on Azure

Administering Oracle as a DBaaS on Azure requires Azure cloud skills outside the traditional database administration functions. Managing infrastructure as a service can interfere with defined database operations. In such scenarios, a better option is to use the Oracle Database as a service on Azure (DBaaS). The DBaaS provides access to a database without requirements to deploy and manage the underlying infrastructure.

DBaaS is delivered as a managed database service, which means that the provider takes care of patching, upgrading, and backing up the database. Tessell primarily provides Oracle database as service – PaaS, also called as 'DBaaS – Database as service on Azure. Tessell's DBaaS platform is available for coselling with Microsoft, delivering the full power of Tessell on Azure. Joint Tessell-Microsoft customers can apply Tessell's advanced cloud-based database-as-a-service (DBaaS) platform with the expertise and support of Microsoft's sales and technical teams. Tessell's DBaaS is Azure-native service with the following benefits:

- Oracle self-service, DevOps integration, and production operations without having to deploy and manage the underlying infrastructure.
- Running on Azure high-performance-compute (HPC, LSV3 series), the most demanding production Oracle workloads can be brought to Azure.
- Support for all Oracle database management packs.

Lightbits: performance for Oracle on Azure VMs

The Lightbits Cloud Data Platform provides scalable and cost-efficient high-performance storage that is easy to consume on Azure. It removes the bottlenecks associated with native storage on the public cloud, such as scalable performance and consistently low latency. Removing these bottlenecks offers rich data services and resiliency that enterprises rely on. It can deliver up to 1 million IOPS/volume and up to 3 million IOPs per VM. Lightbits cluster can scale vertically and horizontally. Lightbits support different sizes of Lsv3 and Lasv3 VMs for their clusters.

For other options, see L32sv3/L32asv3: 7.68 TB, L48sv3/L48asv3: 11.52 TB, L64sv3/L64asv3: 15.36 TB, L80sv3/L80asv3: 19.20 TB. In real-world workload test scenarios, Lightbits delivers more than 10X more IOPS than the best available Azure native storage (Ultra disk).

The Lightbits Cloud Data Platform also provides synchronous replication across multiple availability zones, so you can have a dormant Oracle instance without starting it on a different zone, and if the zone fails, activate the database using the same Lightbits volumes that you used in the different zone without waiting for any log transfer.

The Lightbits Cloud Data Platform supports Oracle ASM and also supports shared raw block devices to use with Oracle RAC.

The following table provides other inputs to help you to determine the appropriate disk type.

Expand table

Parameter	Description
Other	Flexible model at TiB granularity
Provisioning Model	Incremental snapshot for fast restore; Snapshot export for hardening.
BCDR	See redundancy capabilities of Lightbits in redundancy requirements.
Redundancy & Scale Targets	Built-in multi availability zone support via synchronous replication, and automatic cluster expansion based on storage usage.
Encryption	Encryption at rest is supported.

Tessel: Performance best practices for Oracle on Azure VMs

Tessell primarily provides Oracle database as service – PaaS, which is also called as “DBaaS’ – Database as service. Tessell's DBaaS platform is available for coselling with

Microsoft, delivering the full power of Tessell on Azure. Joint Tessell-Microsoft customers can use Tessell's advanced cloud-based database-as-a-service (DBaaS) platform and the extensive expertise and support of Microsoft's sales and technical teams. Tessell's DBaaS as Azure-native solution provides the following benefits:

- Oracle self-service, DevOps integration, and production operations without having to deploy and manage the underlying infrastructure.
- Run on Azure high-performance-compute (HPC, LSV3 series), the most demanding production Oracle workloads can be brought to Azure.
- Support for all Oracle database management packs.

Apart from providing Oracle as DBaaS on Azure, Tessell provides NVMe uses Non-Volatile Memory Express (NVMe) to provide high IOPS and throughput required to run Oracle database on Azure VMs. Use NVMe storage mount on L series VMs to reach IOPS & throughput up to 3,800,000 & 20,000 MB/s. For more information, see Tessell's Oracle SLOB benchmark details on Azure.

The following table provides other inputs to help you to determine the appropriate disk type.

Expand table

Other parameters	DBaaS – A Managed service options for Oracle on Azure.
Provisioning Model	Upfront Provisioning
BCDR	Azure Snapshot, Backups, HA/DR
Redundancy & Scale Targets	Out-of-the-box Multi-Availability Zone (AZ) HA and cross-region DR services
Encryption	Azure Key Vault based & bring your own encryption

Silk: Performance best practices for Oracle on Azure VMs

Silk focuses on providing performance (IOPS & throughput) 50 times more than Azure Native storage recommended for Oracle on Azure IaaS. With the storage 1Gib-128TiB per volume, you can get IOPS & Throughput respectively 2,000,000 & 20,000 MB/sec.

The following table provides other inputs to help you to determine the appropriate disk type.

Expand table

Other parameters	SaaS offering
Provisioning Model	Per GB granularity, online resize & scale-up or out, thin provisioned, compressed, optional deduped
BCDR	One-to-Many Multi-Zone and Multi-Region Replication, Instant zero footprint Snapshot, Clone, Revert, and Extract for AI / BI, Testing, or Back up
Redundancy & Scale Targets	One-to-Many Multi-Zone and Multi-Region Replication
Encryption	Azure Key Vault based & bring your own encryption

Next steps

- Migrate Oracle workload to Azure VMs (IaaS)
- Performance best practices for Oracle on Azure VMs

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Backup strategies for Oracle Database on an Azure Linux VM

Article • 08/22/2024

Applies to: ✔ Linux VMs

Database backups help protect the database against data loss that's due to storage component failure and datacenter failure. They can also be a means of recovery from human error and a way to clone a database for development or testing purposes.

In Azure, all storage is highly redundant. The loss of one or more disks won't lead to a database outage. Backups are most often used to protect against human error, to facilitate cloning operations, or to preserve data for regulatory purposes.

Backups also help protect against regional outages when a disaster recovery technology like DataGuard is not in use. In this case, the backups must be stored in different Azure regions via geo-redundant replication, so they're available outside the primary database region.

Azure Storage

The Azure Storage services are Microsoft's cloud solution for modern data-storage scenarios. Azure Storage offers services that you can use to mount external storage to an Azure Linux virtual machine (VM), which is suitable as backup media for Oracle Database instances. A backup tool such as Oracle Recovery Manager (RMAN) is required to initiate a backup or restore operation, and to copy the backup to or from Azure Storage.

Azure Storage services offer the following benefits:

- **Durable and highly available**. Redundancy helps keep data safe during transient hardware failures. All storage is triple mirrored by default. You can also opt to replicate data across datacenters or geographical regions for more protection from local catastrophes or natural disasters. Data replicated in this way remains highly available in the event of an unexpected outage.
- **Secure**: Azure Storage encrypts all data written to a storage account. Azure Storage gives you detailed control over who has access to your data.
- **Scalable**: Azure Storage is massively scalable to meet the data storage and performance needs of today's applications.

- **Managed**: Azure handles hardware maintenance, updates, and critical issues for you.
- **Accessible**: Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides client libraries for Azure Storage in a variety of languages, including .NET, Java, Node.js, Python, PHP, Ruby, and Go. Microsoft also provides a mature REST API.

  Azure Storage supports scripting in Azure PowerShell or the Azure CLI. The Azure portal and Azure Storage Explorer offer visual solutions for working with your data.

The Azure Storage platform includes the following data services that are suitable to use as backup media for Oracle Database:

- **Azure Blob Storage**: An object store for text and binary data. It also includes support for big data analytics through Azure Data Lake Storage Gen2.
- **Azure NetApp Files**: Complete networked storage solution including advanced data management capabilities for taking snapshots, cloning, and replicating database volumes.
- **Azure Files**: Managed file shares for cloud or on-premises deployments.
- **Azure Disk Storage**: Block-level storage volumes for Azure VMs.

Cross-regional storage mounting

The ability to access backup storage across regions is an important aspect of business continuity and disaster recovery (BCDR). It's also useful for cloning databases from backups into different geographical regions. Azure cloud storage provides five levels of redundancy:

- Locally redundant storage (LRS): Your data is replicated three times within a single physical location in the primary region.
- Zone-redundant storage (ZRS): Your data is replicated synchronously across three availability zones in the primary region. LRS helps protect your data in the primary region and helps protect each availability zone.
- Geo-redundant storage (GRS): Your data is replicated asynchronously to a secondary region. LRS helps protect your data in the primary and secondary regions.
- Geo-zone-redundant storage (GZRS): Your data is copied synchronously across three Azure availability zones in the primary region via ZRS. Your data is then

copied asynchronously to a single physical location in the secondary region. In all locations, LRS helps protect the data.

- Read-access geo-redundant storage (RA-GRS) and read-access geo-zone-redundant storage (RA-GZRS): You have read-only access to data replicated to the secondary region at all times.

Blob and file storage

When you're using Azure Files with either the Server Message Block (SMB) protocol or the Network File System (NFS) 4.1 protocol to mount as backup storage, Azure Files doesn't support RA-GRS or RA-GZRS.

Azure Files backup storage can scale up to 100 (TiB), with support for LRS, GRS, and GZRS redundancy options.

Azure Blob Storage mounted via the NFS 3.0 protocol currently supports only LRS and ZRS redundancy. Azure Blob Storage configured with any redundancy option can be mounted via Blobfuse.

Recovery Services vault

A Recovery Services vault is a management entity that stores recovery points created over time. It provides an interface to perform backup-related operations. These operations include taking on-demand backups, performing restores, and creating backup policies.

Azure Backup automatically handles storage for the vault. You do need to specify how that storage is replicated at the time of creation. You can't change replication after items are protected in the vault. For regional redundancy, choose the geo-redundant setting.

If you intend to restore to a secondary Azure paired region, enable the Cross Region Restore feature. When you enable Cross Region Restore, the backup storage is moved from GRS to RA-GRS.

Azure Blob Storage

Azure Blob Storage is a cloud-based service for storing large amounts of unstructured data and is suitable for Oracle Database backups. You can mount Azure Blob Storage to Azure Linux VMs by using Blobfuse (Linux FUSE) or the NFS v3.0 protocol.

Blobfuse

Blobfuse is an open-source project that provides a virtual file system backed by Azure Blob Storage. It uses the libfuse open-source library to communicate with the Linux FUSE kernel module. It implements file-system operations by using the Azure Blob Storage REST APIs.

Blobfuse is currently available for Ubuntu and RedHat distributions. It's also available for Kubernetes via the CSI driver.

Blobfuse is ubiquitous across Azure regions and works with all storage account types, including general-purpose v1/v2 and Azure Data Lake Storage Gen2. But it doesn't perform as well as alternative protocols. For suitability as the database backup medium, we recommend using the SMB or NFS protocol to mount Azure Blob Storage.

NFS v3.0

Azure support for the NFS v3.0 protocol is available. NFS support enables Windows and Linux clients to mount an Azure Blob Storage container to an Azure VM.

To ensure network security, the storage account that you use for NFS mounting must be contained within a virtual network. Microsoft Entra security and access control lists (ACLs) are not yet supported in accounts that have NFS 3.0 protocol support enabled on them.

Azure Files

Azure Files is a cloud-based, fully managed distributed file system. You can mount it to on-premises or cloud-based Windows, Linux, or macOS clients.

Azure Files offers fully managed cross-platform file shares in the cloud that are accessible via the SMB and NFS protocols. Azure Files doesn't currently support multiple-protocol access, so a share can only be either an NFS share or an SMB share. We recommend determining which protocol best suits your needs before you create Azure file shares.

You can also help protect Azure file shares by using Azure Backup for a Recovery Services vault. This approach provides another layer of protection to the Oracle RMAN backups.

Azure Files with NFS v4.1

You can mount Azure file shares in Linux distributions by using the NFS v4.1 protocol. There are limitations to supported features. For more information, see Support for Azure

Storage features.

Azure NFS file shares are supported in all the same regions that support premium file storage.

For the most up-to-date list, see the **Premium Files Storage** entry on the page for Azure products available by region.

Azure Files with SMB 3.0

You can mount Azure file shares in Linux distributions by using the SMB kernel client. The Common Internet File System (CIFS) protocol, available on Linux distributions, is a dialect of SMB. When you mount an Azure file share on Linux VMs by using SMB, it's mounted as a CIFS-type file system, and the CIFS package must be installed.

The ability to mount Azure file shares via SMB is generally available in all Azure regions. It shows the same performance characteristics as NFS v3.0 and v4.1 protocols, so we currently recommend it as the method to provide backup storage media to Azure Linux VMs.

Two supported versions of SMB are available: SMB 2.1 and SMB 3.0. We recommend SMB 3.0, because it supports encryption in transit. However, Linux kernel versions have differing support for SMB 2.1 and 3.0. To ensure that your application supports SMB 3.0, see Mount an SMB Azure file share on Linux.

Because Azure Files is a multiuser file-share service, you should tune certain characteristics to make it more suitable as backup storage media. We recommend turning off caching and setting the user and group IDs for created files.

Azure NetApp Files

The Azure NetApp Files service is a complete storage solution for Oracle Database in Azure VMs. Built on metered file storage, it supports any workload type and is highly available by default. Together with the Oracle Direct NFS driver, Azure NetApp Files provides a highly optimized storage layer for Oracle Database.

Azure NetApp Files provides efficient storage-based snapshots on the underlying storage system that uses a redirect-on-write mechanism. Although snapshots are fast to take and restore, they serve only as a first line of defense. They can account for most of the required restore operations of any organization, which are often part of recovery from human error.

However, snapshots are not a complete backup. To cover all backup and restore requirements, you must create external snapshot replicas or other backup vaults in a remote geography to help protect against regional outages. Read more about how Azure NetApp Files snapshots work.

To ensure the creation of database-consistent snapshots, the backup process must be orchestrated between the database and the storage. The Azure Application Consistent Snapshot (AzAcSnap) command-line tool enables data protection for third-party databases. It handles all the orchestration required to put those databases into an application-consistent state before taking a storage snapshot. After that, it returns the databases to an operational state. Oracle Database instances are supported with AzAcSnap since version 5.1.

To learn more about using Azure NetApp Files for Oracle Database on Azure, see Solution architectures using Azure NetApp Files.

Azure Backup service

Azure Backup is a fully managed platform as a service (PaaS) solution for backing up your data and recovering it from the Microsoft Azure cloud. Azure Backup can back up and restore on-premises clients, Azure VMs, and Azure file shares. It can also back up SQL Server, Oracle, MySQL, PostgreSQL, and SAP HANA databases on Azure VMs.

Azure Backup provides independent and isolated backups to guard against accidental destruction of original data. Backups are stored in a Recovery Services vault with built-in management of recovery points.

Azure Backup uses the Azure cloud to deliver high availability with no maintenance or monitoring overhead. It doesn't limit the amount of inbound or outbound data that you transfer, and it doesn't charge for the data that you transfer. Data is secured in transit and at rest.

Azure Backup offers multiple types of replication to keep your backup data highly available:

- LRS replicates your data three times (that is, it creates three copies of your data) in a storage scale unit in a datacenter.
- GRS is the default and recommended replication option. GRS replicates your data to a secondary region, hundreds of miles away from the primary location of the source data.

A vault created with GRS redundancy includes the option to configure the Cross Region Restore feature. You can use this feature to restore data in a secondary Azure paired

region.

The Azure Backup service provides a framework to achieve application consistency during backups of Windows and Linux VMs for various applications like Oracle, MySQL, Mongo DB, SAP HANA, and PostgreSQL: application-consistent snapshots. This framework involves invoking pre-scripts (to quiesce the applications) before taking a snapshot of disks. It calls post-scripts (commands to unfreeze the applications) after the snapshot is completed, to return the applications to the normal mode.

Although you can find sample pre-scripts and post-scripts on GitHub, you're responsible for creating and maintaining these scripts. In the case of Oracle, the database must be in archive log mode to allow online backups. You must create and maintain the appropriate database beginning and ending backup commands that are run in the pre-scripts and post-scripts.

Azure Backup provides an enhanced pre-script and post-script framework in which it provides packaged pre-scripts and post-scripts for selected applications. You just name the application, and then Azure Backup automatically invokes the relevant pre-scripts and post-scripts. Microsoft manages the packaged pre-scripts and post-scripts, so you can be assured of the support, ownership, and validity of these scripts.

Currently, the supported applications for the enhanced framework are Oracle 12.1 or later and MySQL. The snapshot is a full copy of the storage and not an incremental or copy-on-write snapshot, so it's an effective medium to restore your database from.

VLDB considerations

Backup strategies for very large databases (VLDBs) require careful consideration because of their size. Using RMAN to back up to Azure Blob Storage or Azure Files might not provide the required throughput to back up a VLDB in the target time frame.

You can use RMAN incremental backup to reduce backup sizes. This approach might allow Azure Storage to be used as the backup medium for VLDBs. However, it might not be effective for VLDBs that have high volumes of changes.

We recommend using Azure services that provide snapshot capabilities, such as Azure Backup or Azure NetApp Files, for VLDBs. Application-consistent snapshots, where the databases are automatically placed in and out of backup mode, take only seconds to create regardless of the size of the database.

Your backup strategy might be also tied to the overall storage solution that the organization uses for Oracle Database. Database workloads that have extreme I/O throughput often use Azure NetApp Files or third-party Azure Marketplace solutions

such as Silk to underpin the database storage throughput and IOPS requirements. These solutions also provide application-consistent snapshots for fast database backup and restore operations.

Next steps

- Create an Oracle Database instance on an Azure VM
- Back up Oracle Database to Azure Files
- Back up Oracle Database by using the Azure Backup service

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Back up and recover Oracle Database on an Azure Linux VM by using Azure Files

Article • 08/22/2024

Applies to: ✔ Linux VMs

This article demonstrates the use of Azure Files as a medium to back up and restore an Oracle database running on an Azure virtual machine (VM). The steps in this article have been tested against Oracle 12.1 and later.

In this article, you use Oracle Recovery Manager (RMAN) to back up the database to an Azure file share mounted to a VM via the Server Message Block (SMB) protocol. Using Azure Files for backup media is cost-effective and performant. However, for very large databases, Azure Backup provides a better solution.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see Sign in with the Azure CLI.
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.
- To perform the backup and recovery process, you must first create a Linux VM that has an installed instance of Oracle Database. We recommend using Oracle 12.x or later.
- Create an Oracle Database instance by following the steps in Create an Oracle Database instance in an Azure VM.

Prepare the database environment

1. To create a Secure Shell (SSH) session with the VM, use the following command. Replace `<publicIpAddress>` with the public address value for your VM.

   ```bash
   ssh azureuser@<publicIpAddress>
   ```

2. Switch to the root user:

   ```bash
   sudo su -
   ```

3. Add the `oracle` user to the */etc/sudoers* file:

   ```bash
   echo "oracle   ALL=(ALL)      NOPASSWD: ALL" >> /etc/sudoers
   ```

4. This step assumes that you have an Oracle instance (test) that's running on a VM named *vmoracle19c*.

   Switch to the `oracle` user:

   ```bash
   sudo su - oracle
   ```

5. Before you connect, set the environment variable `ORACLE_SID`:

   ```bash
   export ORACLE_SID=test;
   ```

   You should also add the `ORACLE_SID` variable to the `oracle` user's *.bashrc* file for future sign-ins by using the following command:

   ```bash
   echo "export ORACLE_SID=test" >> ~oracle/.bashrc
   ```

6. Start the Oracle listener if it isn't already running:

```bash
lsnrctl start
```

The output should look similar to the following example:

```bash
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 18-SEP-2020
03:23:49

Copyright (c) 1991, 2019, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/19.0.0/dbhome_1/bin/tnslsnr: please
wait...

TNSLSNR for Linux: Version 19.0.0.0.0 - Production
System parameter file is
/u01/app/oracle/product/19.0.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/vmoracle19c/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=vmoracle19c.eastus.cloudapp.azure.com)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=vmoracle19c.eastus.cloudapp.azure.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 -
Production
Start Date                18-SEP-2020 03:23:49
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/19.0.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/vmoracle19c/listener/alert/log.xml
Listening Endpoints Summary...
 (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=vmoracle19c.eastus.cloudapp.azure.com)(PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
The listener supports no services
The command completed successfully
```

7. Create the location for the fast recovery area:

```bash
mkdir /u02/fast_recovery_area
```

8. Connect to the database:

```bash
sqlplus / as sysdba
```

9. Start the database if it isn't already running:

```bash
SQL> startup
```

10. Set database environment variables for the fast recovery area:

```bash
SQL> alter system set db_recovery_file_dest_size=4096M scope=both;
SQL> alter system set db_recovery_file_dest='/u02/fast_recovery_area'
scope=both;
```

11. Make sure the database is in `ARCHIVELOG` mode to enable online backups.

    Check the log archive status:

```bash
SQL> SELECT log_mode FROM v$database;

LOG_MODE
------------
NOARCHIVELOG
```

    If the log archive is in `NOARCHIVELOG` mode, run the following commands in SQL Plus:

```bash
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
```

```
SQL> ALTER DATABASE OPEN;
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

12. Create a table to test the backup and restore operations:

```bash
SQL> create user scott identified by tiger quota 100M on users;
SQL> grant create session, create table to scott;
SQL> connect scott/tiger
SQL> create table scott_table(col1 number, col2 varchar2(50));
SQL> insert into scott_table VALUES(1,'Line 1');
SQL> commit;
SQL> quit
```

Back up to Azure Files

To back up to Azure Files, complete these steps:

1. Set up Azure Files.
2. Mount the Azure file share to your VM.
3. Back up the database.
4. Restore and recover the database.

Set up Azure Files

In this section, you back up the Oracle database to Azure Files by using Oracle RMAN. Azure file shares are fully managed file shares that stay in the cloud. You can access them by using either the SMB protocol or the Network File System (NFS) protocol.

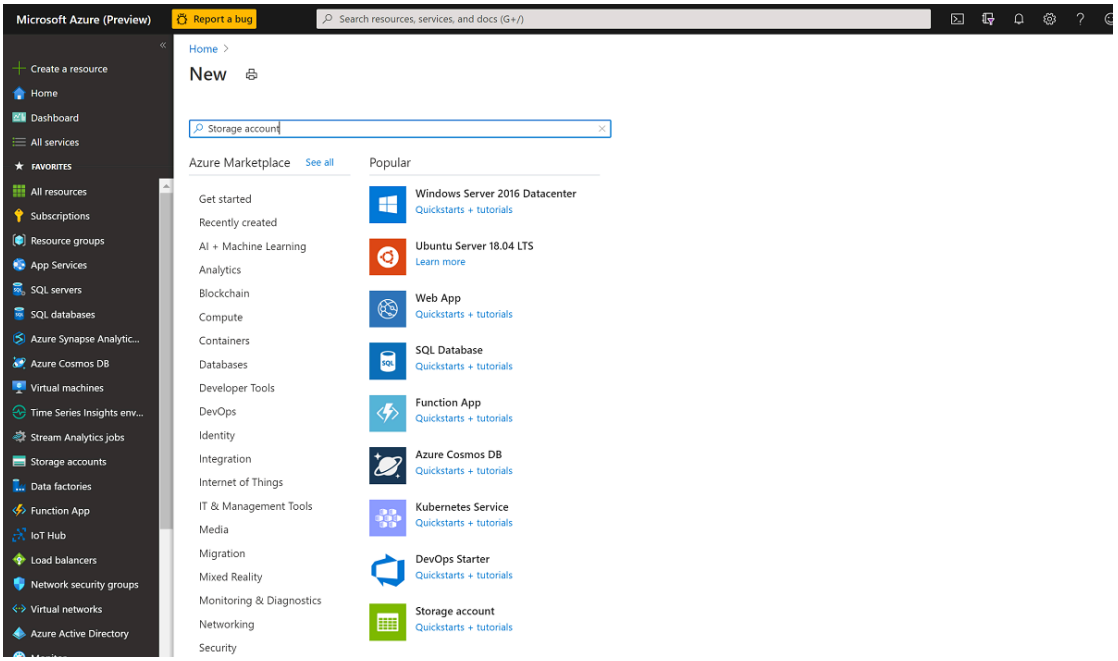
The following procedures cover creating a file share that uses the SMB protocol to mount to your VM. For information about how to mount by using NFS, see Create an NFS share.

When you're mounting the Azure file share, use the `cache=none` option to disable caching of file share data. To ensure that the `oracle` user owns the files created in the share, set the `uid=oracle` and `gid=oinstall` options.

Portal

Set up your storage account:

1. In the Azure portal, select + **Create a resource**, and then search for and select **Storage Account**.



2. On the **Create storage account** pane:
   a. For **Resource group**, select your existing resource group, **rg-oracle**.
   b. For **Storage account name**, enter **oracbkup1**.
   c. Ensure that **Location** is set to the same region as all your other resources in the resource group.
   d. Set **Performance** to **Standard**.
   e. For **Account kind**, select **StorageV2 (general purpose v2)**.
   f. For **Replication**, select **Locally-redundant storage (LRS)**.

Dashboard > Resource groups > rg-oracle > New >

Create storage account

Basics | Networking | Data protection | Advanced | Tags | Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.
Learn more about Azure storage accounts

Project details

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

Subscription *

Resource group * rg-oracle
Create new

Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Storage account name * orabkup1

Location * (US) East US

Performance Standard Premium

Account kind StorageV2 (general purpose v2)

Replication Locally-redundant storage (LRS)

Review + create | < Previous | Next : Networking >

3. Select the **Advanced** tab. Under **Azure Files**, set **Large file shares** to **Enabled**. Select **Review + Create**, and then select **Create**.

Dashboard > Resource groups > rg-oracle > New > Storage account >

Create storage account

Basics | Networking | Data protection | Advanced | Tags | Review + create

Security

Secure transfer required Disabled Enabled

Minimum TLS version Version 1.2

Infrastructure encryption Disabled Enabled

Sign up is currently required to enable infrastructure encryption on a per-subscription basis. Sign up for infrastructure encryption

Blob storage

Allow Blob public access Disabled Enabled

Blob access tier (default) Cool Hot

NFS v3 Disabled Enabled

The current combination of storage account kind, performance, replication, and location does not support the NFS v3 feature. Learn more about NFS v3

Data Lake Storage Gen2

Hierarchical namespace Disabled Enabled

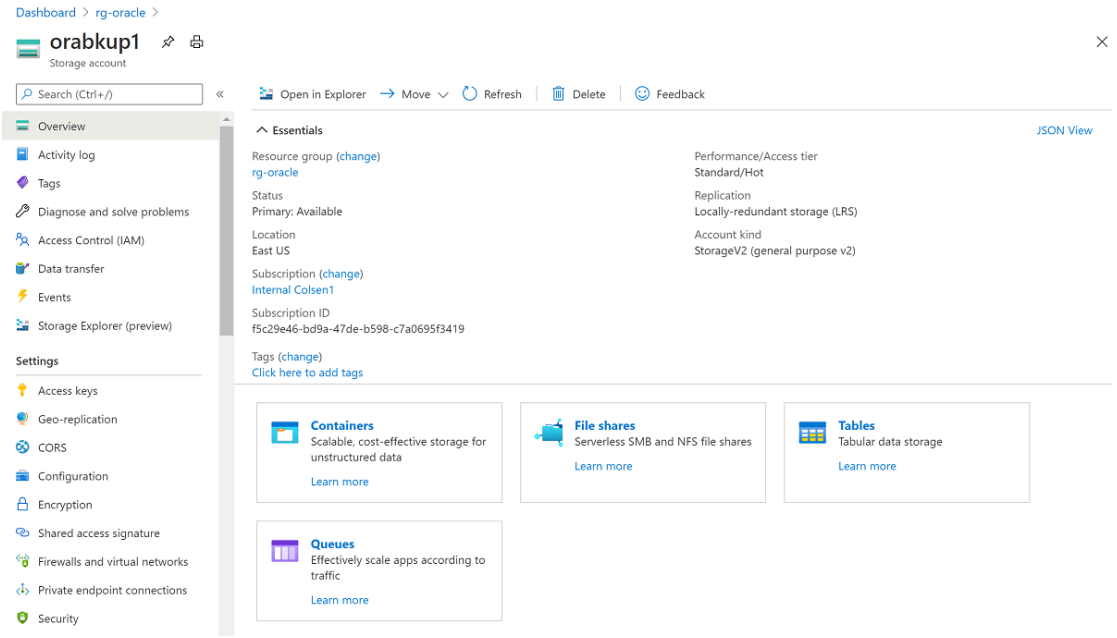
Azure Files

Large file shares Disabled Enabled

Large file share storage accounts do not have the ability to convert to geo-redundant storage offerings and upgrade is permanent.

Review + create | < Previous | Next : Tags >

4. When the storage account is created, go to the resource and select **File shares**.



5. Select + **File share**, and then on the **New file share** panel:

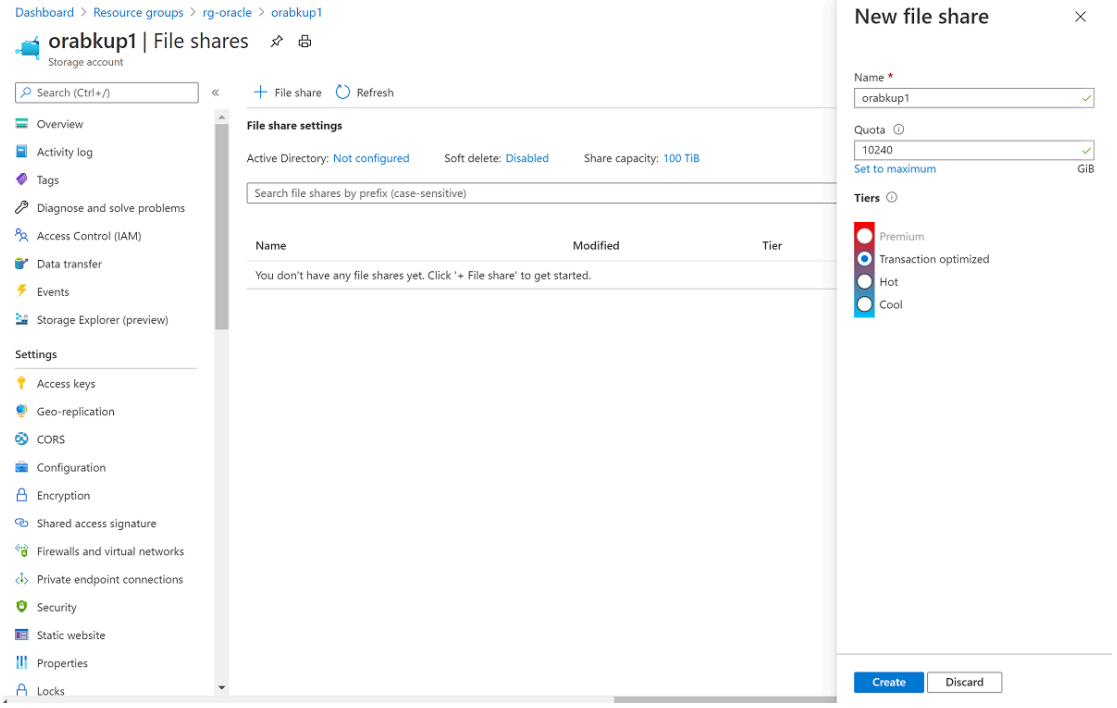
   a. For **Name**, enter **orabkup1**.

   b. Set **Quota** to **10240** gibibytes (GiB).

      The quota reflects an upper boundary that the file share can grow to. Because you're using standard storage in this example, resources are pay-as-you-go and not provisioned. So, setting the quota to 10 tebibytes (TiB) won't incur costs beyond what you use. If your backup strategy requires more storage, set the quota to an appropriate level to hold all backups.

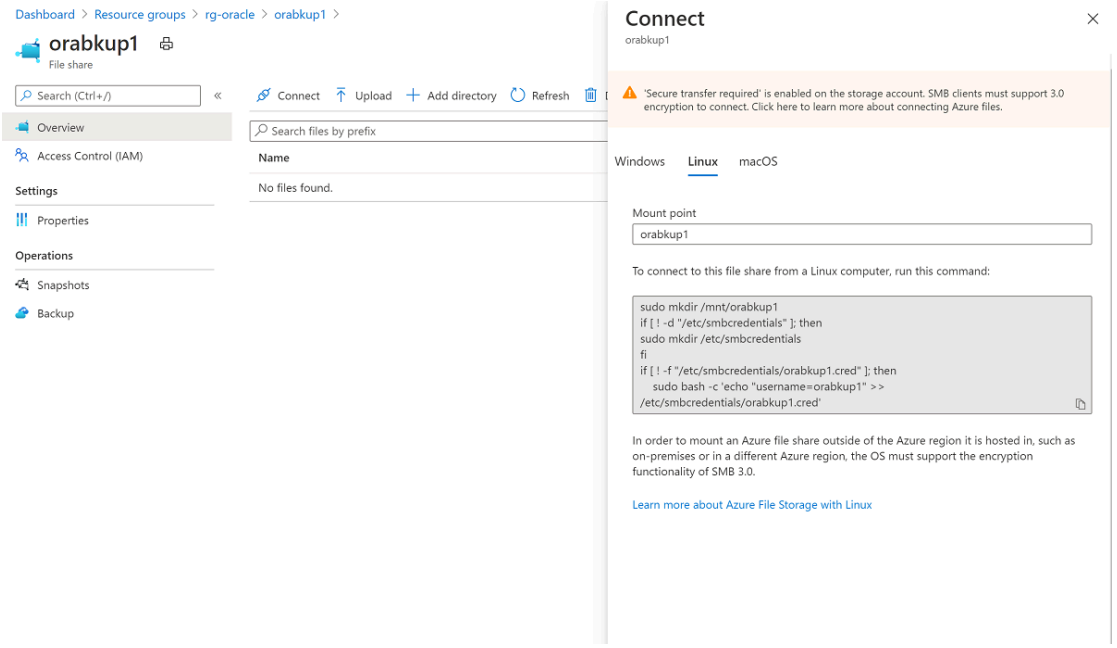
   c. Under **Tiers**, select **Transaction optimized**.

   d. Select **Create**.



6. When the file share is created, select **orabkup1** on the **File share settings** pane.

7. Select the **Connect** tab to open the **Connect** panel, and then select the **Linux** tab. Copy the provided commands to mount the file share by using the SMB protocol.



Mount the Azure file share to your VM

1. Create the mount point:

```
Bash
```

```
sudo mkdir /mnt/orabackup
```

2. Set up credentials:

```Bash
if [ ! -d "/etc/smbcredentials" ]; then
 sudo mkdir /etc/smbcredentials
fi
```

3. Run the following command. Substitute `<Your Storage Account Key1>` with the storage account key that you retrieved earlier.

```Bash
if [ ! -f "/etc/smbcredentials/orabackup1.cred" ]; then
  sudo bash -c 'echo "username=orabackup1" >>
/etc/smbcredentials/orabackup1.cred'
  sudo bash -c 'echo "password=<Your Storage Account Key1>" >>
/etc/smbcredentials/orabackup1.cred'
fi
```

4. Change permissions on the credentials file:

```Bash
sudo chmod 600 /etc/smbcredentials/orabackup1.cred
```

5. Add the mount to the */etc/fstab* file:

```Bash
sudo bash -c 'echo "//orabackup1.file.core.windows.net/orabackup
/mnt/orabackup cifs
nofail,vers=3.0,credentials=/etc/smbcredentials/orabackup1.cred,dir_mod
e=0777,file_mode=0777,serverino,cache=none,uid=oracle,gid=oinstall" >>
/etc/fstab'
```

6. Run the commands to mount the Azure file share by using the SMB protocol:

```Bash
sudo mount -t cifs //orabackup1.file.core.windows.net/orabackup
/mnt/orabackup -o
```

```
vers=3.0,credentials=/etc/smbcredentials/orabackup1.cred,dir_mode=0777,
file_mode=0777,serverino,cache=none,uid=oracle,gid=oinstall
```

If you get an error similar to the following example, the Common Internet File System (CIFS) package might not be installed on your Linux host:

Output

```
mount: wrong fs type, bad option, bad superblock on
//orabackup1.file.core.windows.net/orabackup
```

To check if the CIFS package is installed, run the following command:

Bash

```
sudo rpm -qa|grep cifs-utils
```

If the command returns no output, install the CIFS package by using the following command. Then rerun the `mount` command to mount the Azure file share.

Bash

```
sudo yum install cifs-utils
```

7. Check that the file share is mounted properly by using the following command:

Bash

```
df -h
```

The output should look similar to this example:

Output

```
$ df -h
Filesystem                                      Size  Used Avail Use%
Mounted on
devtmpfs                                        3.3G     0  3.3G   0%
/dev
tmpfs                                           3.3G     0  3.3G   0%
/dev/shm
tmpfs                                           3.3G   17M  3.3G   1%
/run
tmpfs                                           3.3G     0  3.3G   0%
/sys/fs/cgroup
/dev/sda2                                        30G  9.1G   19G  34% /
```

```
/dev/sdc1                                       59G  2.7G   53G   5%
/u02
/dev/sda1                                      497M  199M  298M  41%
/boot
/dev/sda15                                     495M  9.7M  486M   2%
/boot/efi
tmpfs                                          671M     0  671M   0%
/run/user/54321
/dev/sdb1                                       14G  2.1G   11G  16%
/mnt/resource
tmpfs                                          671M     0  671M   0%
/run/user/54322
//orabackup1.file.core.windows.net/orabackup   10T     0   10T   0%
/mnt/orabackup
```

Back up the database

In this section, you use Oracle RMAN to take a full backup of the database and archive logs. You then write the backup as a backup set to the Azure file share that you mounted earlier.

1. Configure RMAN to back up to the Azure Files mount point:

```bash
rman target /
RMAN> configure snapshot controlfile name to
'/mnt/orabkup/snapcf_ev.f';
RMAN> configure channel 1 device type disk format
'/mnt/orabkup/%d/Full_%d_%U_%T_%s';
RMAN> configure channel 2 device type disk format
'/mnt/orabkup/%d/Full_%d_%U_%T_%s';
```

2. In this example, you're limiting the size of RMAN backup pieces to 4 GiB. However, the RMAN backup `maxpiecesize` value can go up to 4 TiB, which is the file size limit for Azure standard file shares and premium file shares. For more information, see Azure Files scalability and performance targets.

```bash
RMAN> configure channel device type disk maxpiecesize 4000G;
```

3. Confirm the configuration change details:

```bash
```

```
RMAN> show all;
```

4. Run the backup. The following command takes a full database backup, including archive log files, as a backup set in compressed format:

```Bash
RMAN> backup as compressed backupset database plus archivelog;
```

You've now backed up the database online by using Oracle RMAN, with the backup located in Azure Files. Because you're storing the backups in Azure Files, you can access them from other VMs if you need to clone the database.

Although using RMAN and Azure Files for database backup has many advantages, backup and restore time is linked to the size of the database. For very large databases, these operations can take considerable time.

An alternative is to use application-consistent VM backups through Azure Backup. This mechanism uses snapshot technology to perform fast backups irrespective of database size. Integration with a Recovery Services vault provides cloud storage of your Oracle Database backups, so you can access them from other VMs and other Azure regions.

Restore and recover the database

1. Shut down the Oracle instance:

```Bash
sqlplus / as sysdba
SQL> shutdown abort
ORACLE instance shut down.
```

2. Remove the database datafiles:

```Bash
cd /u02/oradata/TEST
rm -f *.dbf
```

3. The following commands use RMAN to restore the missing datafiles and recover the database:

```Bash
```

```
rman target /
RMAN> startup mount;
RMAN> restore database;
RMAN> recover database;
RMAN> alter database open;
```

4. Check that the database content is fully recovered:

```bash
RMAN> SELECT * FROM scott.scott_table;
```

The backup and recovery of the Oracle Database 19c database on an Azure Linux VM are now finished.

Delete the VM

When you no longer need the VM, you can use the following command to remove the resource group, the VM, and all related resources:

```azure cli
az group delete --name rg-oracle
```

Next steps

Create highly available VMs

Explore Azure CLI samples for VM deployment

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Back up and recover Oracle Database on an Azure Linux VM by using Azure Backup

Article • 08/22/2024

Applies to: ✔ Linux VMs

This article demonstrates the use of Azure Backup to take disk snapshots of virtual machine (VM) disks, which include the Oracle Database files and the Oracle fast recovery area. By using Azure Backup, you can take full disk snapshots that are suitable as backups and are stored in a Recovery Services vault.

Azure Backup also provides application-consistent backups, which ensure that more fixes aren't required to restore the data. Application-consistent backups work with both file system and Oracle Automatic Storage Management (ASM) databases.

Restoring application-consistent data reduces restoration time, so you can quickly return to a running state. Oracle Database recovery is still necessary after restore. You facilitate the recovery by using Oracle archived redo log files that are captured and stored in a separate Azure file share.

This article walks you through the following tasks:

- ✔ Back up the database with application-consistent backup.
- ✔ Restore and recover the database from a recovery point.
- ✔ Restore the VM from a recovery point.

Prerequisites

- Use the Bash environment in Azure Cloud Shell. For more information, see Quickstart for Bash in Azure Cloud Shell.

  Launch Cloud Shell

- If you prefer to run CLI reference commands locally, install the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see How to run the Azure CLI in a Docker container.
  - If you're using a local installation, sign in to the Azure CLI by using the az login command. To finish the authentication process, follow the steps displayed in

your terminal. For other sign-in options, see Sign in with the Azure CLI.

  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see Use extensions with the Azure CLI.
  - Run az version to find the version and dependent libraries that are installed. To upgrade to the latest version, run az upgrade.

- To perform the backup and recovery process, you must first create a Linux VM that has an installed instance of Oracle Database 12.1 or later.
- Create an Oracle Database instance by following the steps in Create an Oracle Database instance in an Azure VM.

Prepare the environment

To prepare the environment, complete these steps:

1. Connect to the VM.
2. Set up Azure Files storage.
3. Prepare the databases.

Connect to the VM

1. To create a Secure Shell (SSH) session with the VM, use the following command. Replace `<publicIpAddress>` with the public address value for your VM.

```bash
ssh azureuser@<publicIpAddress>
```

2. Switch to the root user:

```bash
sudo su -
```

3. Add the `oracle` user to the */etc/sudoers* file:

```bash
echo "oracle   ALL=(ALL)      NOPASSWD: ALL" >> /etc/sudoers
```

Set up Azure Files storage for the Oracle archived redo log files

The Oracle Database instance's archived redo log files play a crucial role in database recovery. They store the committed transactions needed to roll forward from a database snapshot taken in the past.

When the database is in `ARCHIVELOG` mode, it archives the contents of online redo log files when they become full and switch. Together with a backup, they're required to achieve point-in-time recovery when the database is lost.

Oracle provides the capability to archive redo log files to different locations. The industry best practice is that at least one of those destinations should be on remote storage, so it's separate from the host storage and protected with independent snapshots. Azure Files meets those requirements.

An Azure file share is storage that you an attach to a Linux or Windows VM as a regular file-system component, by using the Server Message Block (SMB) or Network File System (NFS) protocol. To set up an Azure file share on Linux (by using the SMB 3.0 protocol) for use as archive log storage, see Mount an SMB Azure file share on Linux. When you complete the setup, return to this guide and complete all remaining steps.

Prepare the databases

This part of the process assumes that you followed Create an Oracle Database instance in an Azure VM. As a result:

- You have an Oracle instance named `oratest1` that's running on a VM named `vmoracle19c`.
- You're using the standard Oracle `oraenv` script with its dependency on the standard Oracle configuration file */etc/oratab* to set up environment variables in a shell session.

Perform the following steps for each database on the VM:

1. Switch to the `oracle` user:

```bash
sudo su - oracle
```

2. Set the environment variable `ORACLE_SID` by running the `oraenv` script. It will prompt you to enter the `ORACLE_SID` name.

```bash
. oraenv
```

3. Add the Azure file share as another destination for database archive log files.

   This step assumes that you configured and mounted an Azure file share on the Linux VM. For each database installed on the VM, make a subdirectory that's named after your database security identifier (SID).

   In this example, the mount point name is `/backup` and the SID is `oratest1`. So you create the subdirectory `/backup/oratest1` and change ownership to the `oracle` user. Substitute `/backup/SID` for your mount point name and database SID.

```bash
sudo mkdir /backup/oratest1
sudo chown oracle:oinstall /backup/oratest1
```

4. Connect to the database:

```bash
sqlplus / as sysdba
```

5. Start the database if it's not already running:

```bash
SQL> startup
```

6. Set the first archive log destination of the database to the file-share directory that you created earlier:

```bash
SQL> alter system set log_archive_dest_1='LOCATION=/backup/oratest1'
scope=both;
```

7. Define the recovery point objective (RPO) for the database.

To achieve a consistent RPO, consider the frequency at which the online redo log files will be archived. These factors control the frequency:

- The size of the online redo log files. As an online log file becomes full, it's switched and archived. The larger the online log file, the longer it takes to fill up. The added time decreases the frequency of archive generation.
- The setting of the `ARCHIVE_LAG_TARGET` parameter controls the maximum number of seconds permitted before the current online log file must be switched and archived.

To minimize the frequency of switching and archiving, along with the accompanying checkpoint operation, Oracle online redo log files generally have a large size (for example, 1,024M, 4,096M, or 8,192M). In a busy database environment, logs are still likely to switch and archive every few seconds or minutes. In a less active database, they might go hours or days before the most recent transactions are archived, which would dramatically decrease archival frequency.

We recommend that you set `ARCHIVE_LAG_TARGET` to ensure a consistent RPO. A setting of 5 minutes (300 seconds) is a prudent value for `ARCHIVE_LAG_TARGET`. It ensures that any database recovery operation can recover to within 5 minutes of the time of failure.

To set `ARCHIVE_LAG_TARGET`, run this command:

```bash
SQL> alter system set archive_lag_target=300 scope=both;
```

To better understand how to deploy highly available Oracle Database instances in Azure with zero RPO, see Reference architectures for Oracle Database.

8. Make sure the database is in archive log mode to enable online backups.

   Check the log archive status first:

```bash
SQL> SELECT log_mode FROM v$database;

LOG_MODE
------------
NOARCHIVELOG
```

If it's in `NOARCHIVELOG` mode, run the following commands:

```bash
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

9. Create a table to test the backup and restore operations:

```bash
SQL> create user scott identified by tiger quota 100M on users;
SQL> grant create session, create table to scott;
SQL> connect scott/tiger
SQL> create table scott_table(col1 number, col2 varchar2(50));
SQL> insert into scott_table VALUES(1,'Line 1');
SQL> commit;
SQL> quit
```

Back up your data by using Azure Backup

The Azure Backup service provides solutions to back up your data and recover it from the Microsoft Azure cloud. Azure Backup provides independent and isolated backups to guard against accidental destruction of original data. Backups are stored in a Recovery Services vault with built-in management of recovery points, so you can restore as needed.

In this section, you use Azure Backup to take application-consistent snapshots of your running VM and Oracle Database instances. The databases are placed into backup mode, which allows a transactionally consistent online backup to occur while Azure Backup takes a snapshot of the VM disks. The snapshot is a full copy of the storage and not an incremental or copy-on-write snapshot. It's an effective medium to restore your database from.

The advantage of using Azure Backup application-consistent snapshots is that they're fast to take, no matter how large your database is. You can use a snapshot for restore operations as soon as you take it, without having to wait for it to be transferred to the Recovery Services vault.

To use Azure Backup to back up the database, complete these steps:

1. Understand the Azure Backup framework.

2. Prepare the environment for an application-consistent backup.
3. Set up application-consistent backups.
4. Trigger an application-consistent backup of the VM.

Understand the Azure Backup framework

The Azure Backup service provides a framework to achieve application consistency during backups of Windows and Linux VMs for various applications. This framework involves invoking a pre-script to quiesce the applications before taking a snapshot of disks. It calls a post-script to unfreeze the applications after the snapshot is completed.

Microsoft has enhanced the framework so that the Azure Backup service provides packaged pre-scripts and post-scripts for selected applications. These pre-scripts and post-scripts are already loaded on the Linux image, so there's nothing for you to install. You just name the application, and then Azure Backup automatically invokes the relevant scripts. Microsoft manages the packaged pre-scripts and post-scripts, so you can be assured of the support, ownership, and validity of them.

Currently, the supported applications for the enhanced framework are Oracle 12.x or later and MySQL. For details, see Support matrix for managed Azure VM backups.

You can author your own scripts for Azure Backup to use with pre-12.x databases. Example scripts are available on GitHub.

Each time you do a backup, the enhanced framework runs the pre-scripts and post-scripts on all Oracle Database instances installed on the VM. The `configuration_path` parameter in the *workload.conf* file points to the location of the Oracle */etc/oratab* file (or a user-defined file that follows the oratab syntax). For details, see Set up application-consistent backups.

Azure Backup runs the pre-scripts and post-scripts for each database listed in the file that `configuration_path` points to. Exceptions are lines that begin with `#` (treated as comment) or `+ASM` (an Oracle ASM instance).

The Azure Backup enhanced framework takes online backups of Oracle Database instances that operate in `ARCHIVELOG` mode. The pre-scripts and post-scripts use the `ALTER DATABASE BEGIN` and `END BACKUP` commands to achieve application consistency.

For the database backup to be consistent, databases in `NOARCHIVELOG` mode must be shut down cleanly before the snapshot starts.

Prepare the environment for an application-consistent backup

Oracle Database employs job role separation to provide separation of duties by using least privilege. It associates separate operating system (OS) groups with separate database administrative roles. Users can then have different database privileges granted to them, depending on their membership in OS groups.

The `SYSBACKUP` database role (generic name `OSBACKUPDBA`) provides limited privileges to perform backup operations in the database. Azure Backup requires it.

During Oracle installation, we recommend that you use `backupdba` as the OS group name to associate with the `SYSBACKUP` role. But you can use any name, so you need to determine the name of the OS group that represents the Oracle `SYSBACKUP` role first.

1. Switch to the `oracle` user:

   Bash
   ```bash
   sudo su - oracle
   ```

2. Set the Oracle environment:

   Bash
   ```bash
   export ORACLE_SID=oratest1
   export ORAENV_ASK=NO
   . oraenv
   ```

3. Determine the name of the OS group that represents the Oracle `SYSBACKUP` role:

   Bash
   ```bash
   grep "define SS_BKP" $ORACLE_HOME/rdbms/lib/config.c
   ```

   The output looks similar to the following example:

   Output
   ```
   #define SS_BKP_GRP "backupdba"
   ```

   In the output, the value enclosed within double quotation marks is the name of the Linux OS group to which the Oracle `SYSBACKUP` role is externally authenticated. In

this example, it's `backupdba`. Note down the actual value.

4. Verify that the OS group exists by running the following command. Substitute `<group name>` with the value that the previous command returned (without the quotation marks).

   Bash

   ```
   grep <group name> /etc/group
   ```

   The output looks similar to the following example:

   Output

   ```
   backupdba:x:54324:oracle
   ```

   > **Important**
   >
   > If the output doesn't match the Oracle OS group value that you retrieved in step 3, use the following command to create the OS group that represents the Oracle `SYSBACKUP` role. Substitute `<group name>` with the group name that you retrieved in step 3.
   >
   > Bash
   >
   > ```
   > sudo groupadd <group name>
   > ```

5. Create a new backup user named `azbackup` that belongs to the OS group that you verified or created in the previous steps. Substitute `<group name>` with the name of the verified group. The user is also added to the `oinstall` group to enable it to open ASM disks.

   Bash

   ```
   sudo useradd -g <group name> -G oinstall azbackup
   ```

6. Set up external authentication for the new backup user.

   The backup user `azbackup` needs to be able to access the database by using external authentication, so it isn't challenged by a password. To enable this access,

you must create a database user that authenticates externally through `azbackup`. The database uses a prefix for the user name, which you need to find.

Perform the following steps for each database installed on the VM:

a. Log in to the database by using SQL Plus, and check the default settings for external authentication:

```bash
sqlplus / as sysdba
SQL> show parameter os_authent_prefix
SQL> show parameter remote_os_authent
```

The output should look like this example, which shows `ops$` as the database username prefix:

```output
NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------------
-----------
os_authent_prefix                    string      ops$
remote_os_authent                    boolean     FALSE
```

b. Create a database user named `ops$azbackup` for external authentication to the `azbackup` user, and grant `SYSBACKUP` privileges:

```bash
SQL> CREATE USER ops$azbackup IDENTIFIED EXTERNALLY;
SQL> GRANT CREATE SESSION, ALTER SESSION, SYSBACKUP TO ops$azbackup;
```

7. If you receive the error `ORA-46953: The password file is not in the 12.2 format` when you run the `GRANT` statement, follow these steps to migrate the *orapwd* file to 12.2 format. Perform these steps for every Oracle Database instance on the VM.

a. Exit SQL Plus.

b. Move the password file with the old format to a new name.

c. Migrate the password file.

d. Remove the old file.

e. Run the following commands:

```bash
mv $ORACLE_HOME/dbs/orapworatest1 $ORACLE_HOME/dbs/orapworatest1.tmp
orapwd file=$ORACLE_HOME/dbs/orapworatest1
input_file=$ORACLE_HOME/dbs/orapworatest1.tmp
rm $ORACLE_HOME/dbs/orapworatest1.tmp
```

f. Rerun the `GRANT` operation in SQL Plus.

8. Create a stored procedure to log backup messages to the database alert log. Use the following code for each database installed on the VM:

```bash
sqlplus / as sysdba
SQL> GRANT EXECUTE ON DBMS_SYSTEM TO SYSBACKUP;
SQL> CREATE PROCEDURE sysbackup.azmessage(in_msg IN VARCHAR2)
AS
  v_timestamp     VARCHAR2(32);
BEGIN
  SELECT TO_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS')
  INTO v_timestamp FROM DUAL;
  DBMS_OUTPUT.PUT_LINE(v_timestamp || ' - ' || in_msg);
  SYS.DBMS_SYSTEM.KSDWRT(SYS.DBMS_SYSTEM.ALERT_FILE, in_msg);
END azmessage;
/
SQL> SHOW ERRORS
SQL> QUIT
```

Set up application-consistent backups

1. Switch to the root user:

```bash
sudo su -
```

2. Check for the */etc/azure* folder. If it isn't present, create the working directory for the application-consistent backup:

```bash
if [ ! -d "/etc/azure" ]; then
   mkdir /etc/azure
fi
```

3. Check for the *workload.conf* file within the folder. If it isn't present, create it in the */etc/azure* directory and give it the following contents. The comments must begin with `[workload]`. If the file is already present, just edit the fields so that they match the following contents. Otherwise, the following command creates the file and populates the contents:

   Bash

   ```bash
   echo "[workload]
   workload_name = oracle
   configuration_path = /etc/oratab
   timeout = 90
   linux_user = azbackup" > /etc/azure/workload.conf
   ```

   The *workload.conf* file uses the following format:

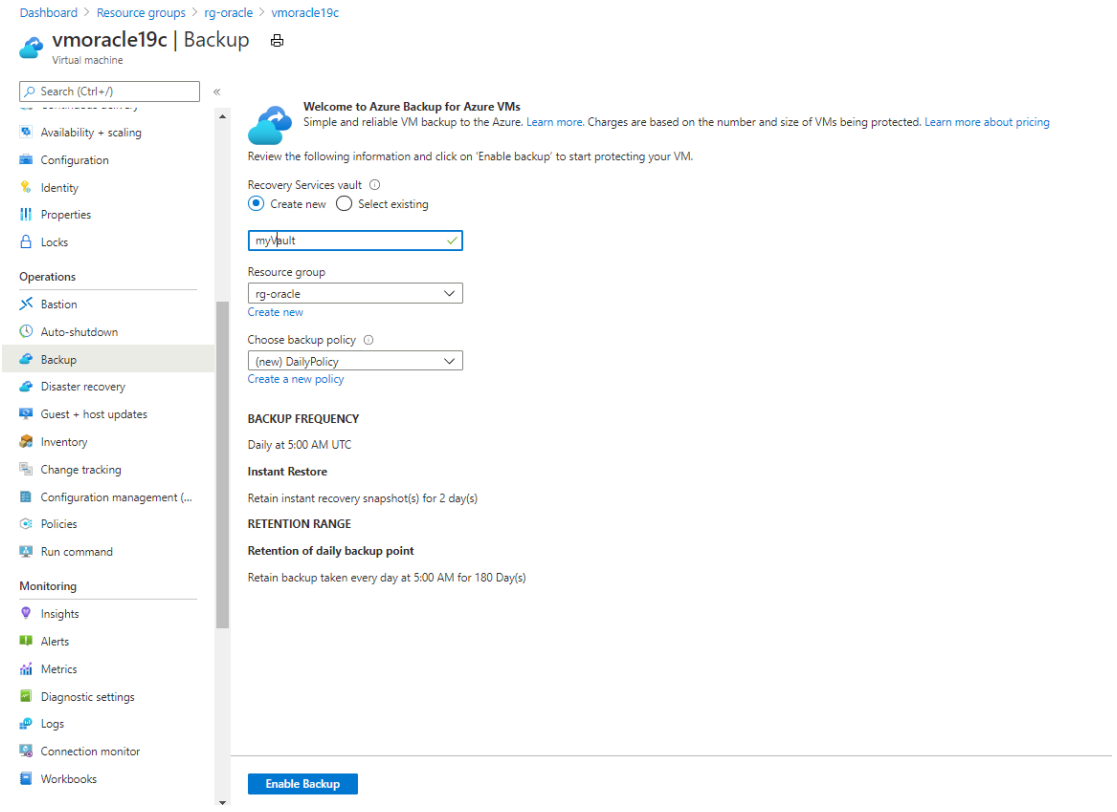
   - The `workload_name` parameter indicates the database workload type. In this case, setting the parameter to `Oracle` allows Azure Backup to run the correct pre-scripts and post-scripts (consistency commands) for Oracle Database instances.
   - The `timeout` parameter indicates the maximum time, in seconds, that each database must complete storage snapshots.
   - The `linux_user` parameter indicates the Linux user account that Azure Backup uses to run database quiesce operations. You created this user, `azbackup`, previously.
   - The `configuration_path` parameter indicates the absolute path name for a text file on the VM. Each line lists a database instance running on the VM. This is typically the */etc/oratab* file that Oracle generates during database installation, but it can be any file with any name that you choose. It must follow these format rules:
     - The file is a text file. Each field is delimited with the colon character (`:`).
     - The first field in each line is the name for an `ORACLE_SID` instance.
     - The second field in each line is the absolute path name for `ORACLE_HOME` for that `ORACLE_SID` instance.
     - All text after the first two fields is ignored.
     - If the line starts with a pound sign (`#`), the entire line is ignored as a comment.
     - If the first field has the value `+ASM`, denoting an Oracle ASM instance, it's ignored.

Trigger an application-consistent backup of the VM

Portal

1. In the Azure portal, go to your **rg-oracle** resource group and select your **vmoracle19c** virtual machine.

2. On the **Backup** pane:
   a. Under **Recovery Services vault**, select **Create new**.
   b. For the name of the vault, use **myVault**.
   c. For **Resource group**, select **rg-oracle**.
   d. For **Choose backup policy**, use **(new) DailyPolicy**. If you want to change the backup frequency or retention range, select **Create a new policy** instead.

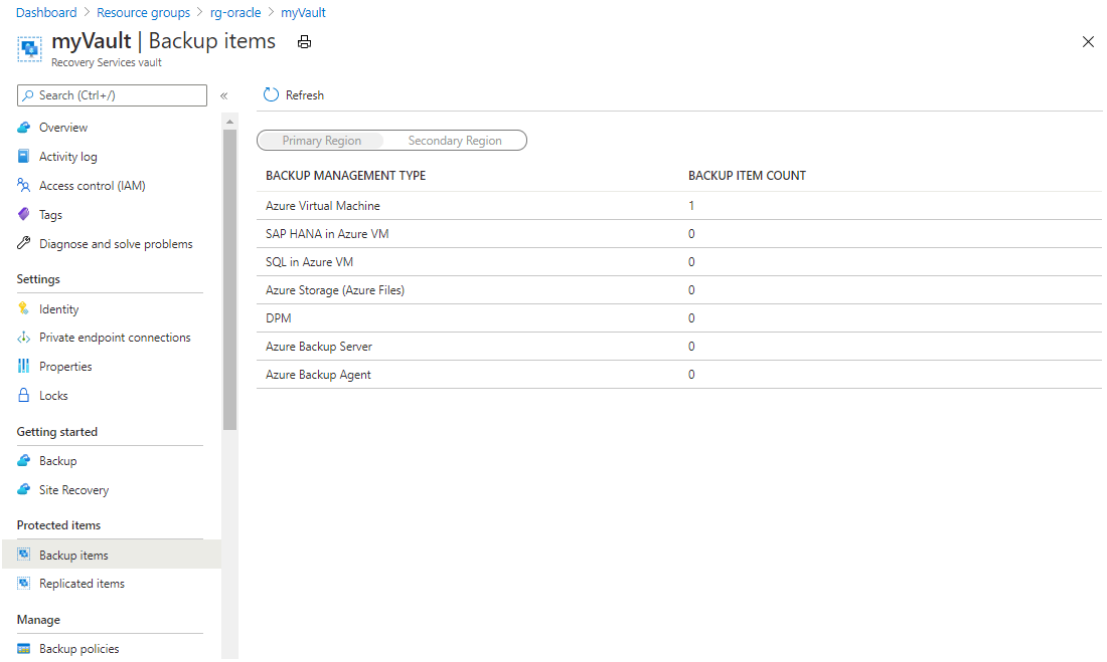


3. Select **Enable Backup**.

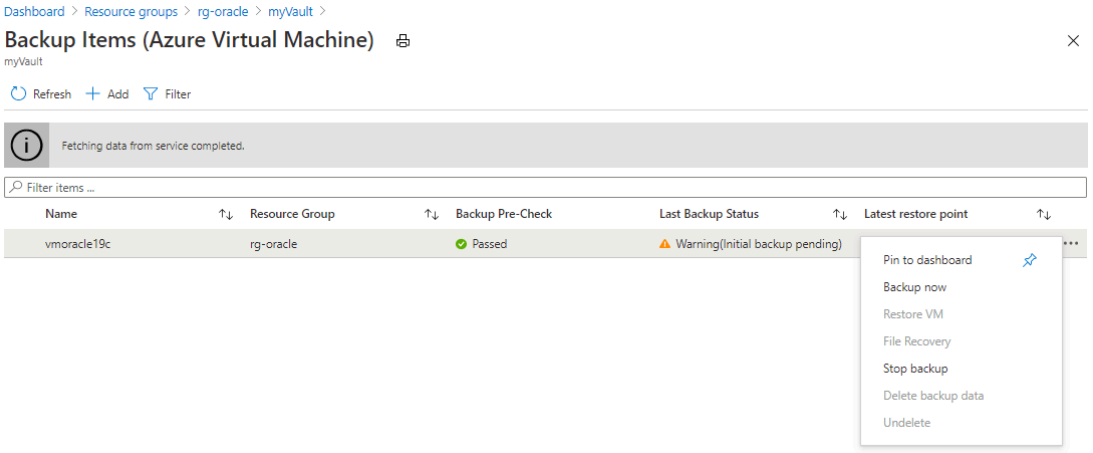
   The backup process doesn't start until the scheduled time expires. To set up an immediate backup, complete the next step.

4. From the resource group pane, select your newly created Recovery Services vault named **myVault**. You might need to refresh the page to see it.

5. On the **myVault - Backup items** pane, under **BACKUP ITEM COUNT**, select the backup item count.

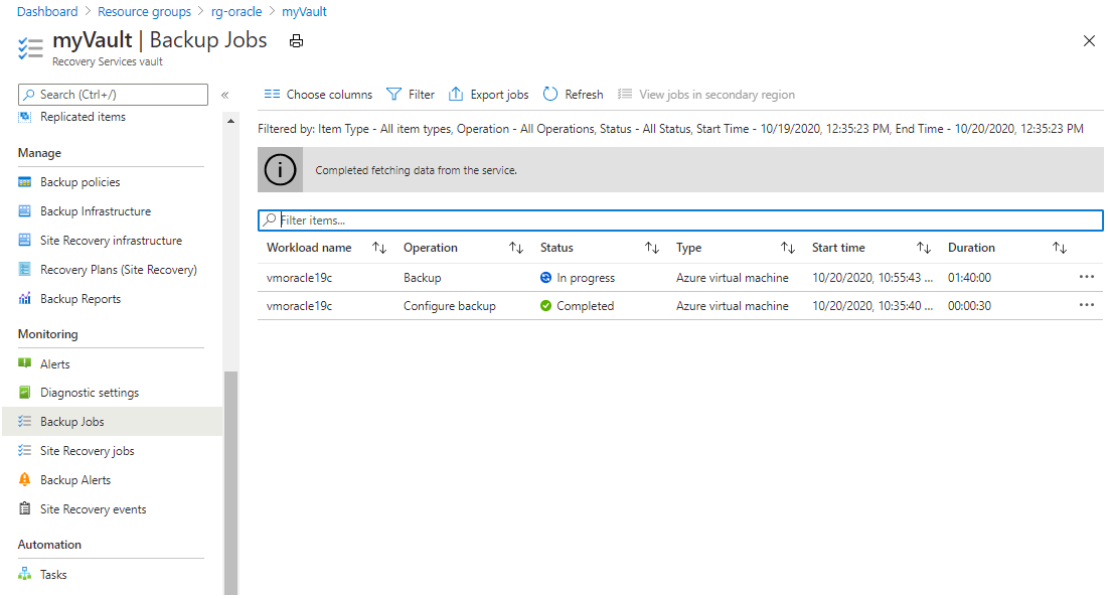


6. On the **Backup Items (Azure Virtual Machine)** pane, select the ellipsis (...) button, and then select **Backup now**.

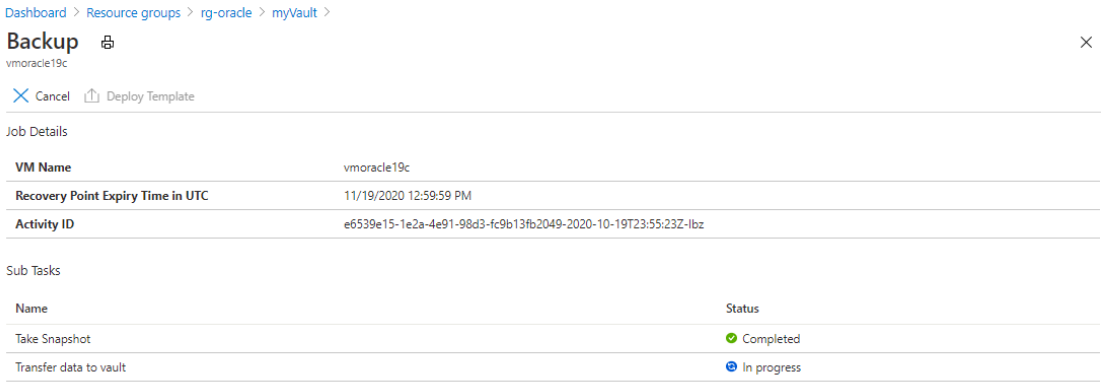


7. Accept the default **Retain Backup Till** value, and then select **OK**. Wait for the backup process to finish.

8. To view the status of the backup job, select **Backup Jobs**.



Select the backup job to see details about its status.



Although it takes only seconds to execute the snapshot, it can take some time to transfer it to the vault. The backup job is not completed until the transfer is finished.

9. For an application-consistent backup, address any errors in the log file at */var/log/azure/Microsoft.Azure.RecoveryServices.VMSnapshotLinux/extension.log*.

Restore the VM

Restoring an entire VM means that you restore the VM and its attached disks to a new VM from a selected restore point. This action also restores all databases that run on the VM. Afterward, you need to recover each database.

To restore an entire VM, complete these steps:

1. Stop and delete the VM.
2. Recover the VM.
3. Set the public IP address.
4. Recover the database.

There are two main choices when you're restoring a VM:

- Restore the VM from which the backups were originally taken.
- Restore (clone) a new VM without affecting the VM from which the backups were originally taken.

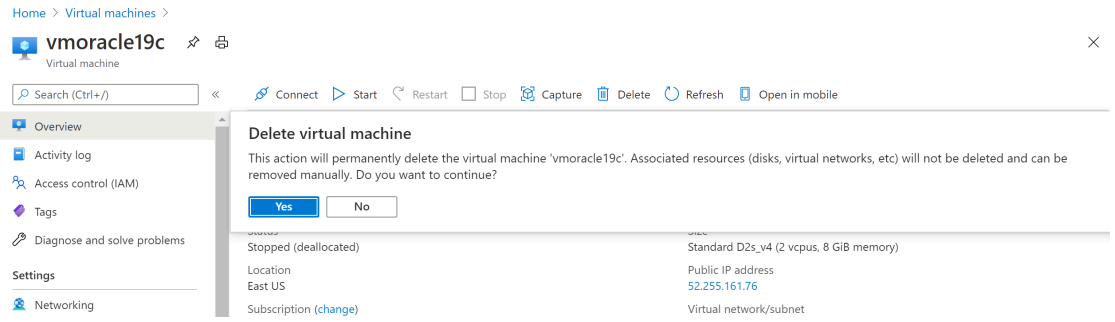
The first steps in this exercise (stopping, deleting, and then recovering the VM) simulate the first use case.

Stop and delete the VM

Portal

1. In the Azure portal, go to the **vmoracle19c** virtual machine, and then select **Stop**.

2. When the virtual machine is no longer running, select **Delete** and then **Yes**.

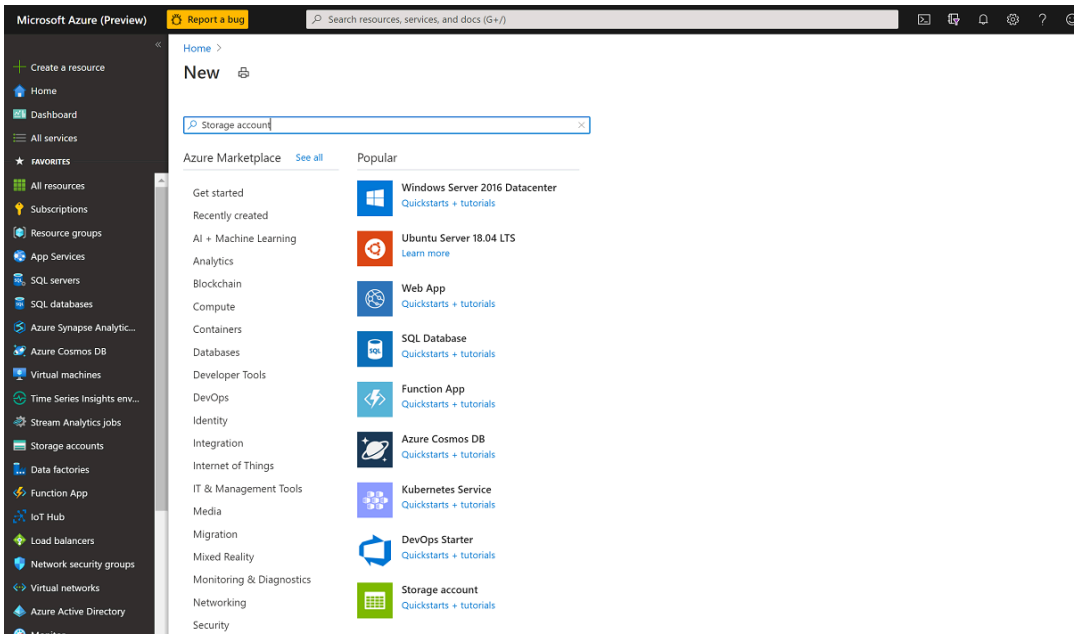


Recover the VM

Portal

1. Create a storage account for staging in the Azure portal:

   a. In the Azure portal, select **+ Create a resource**, and then search for and select **Storage account**.



b. On the **Create storage account** pane:

i. For **Resource group**, select your existing resource group, **rg-oracle**.

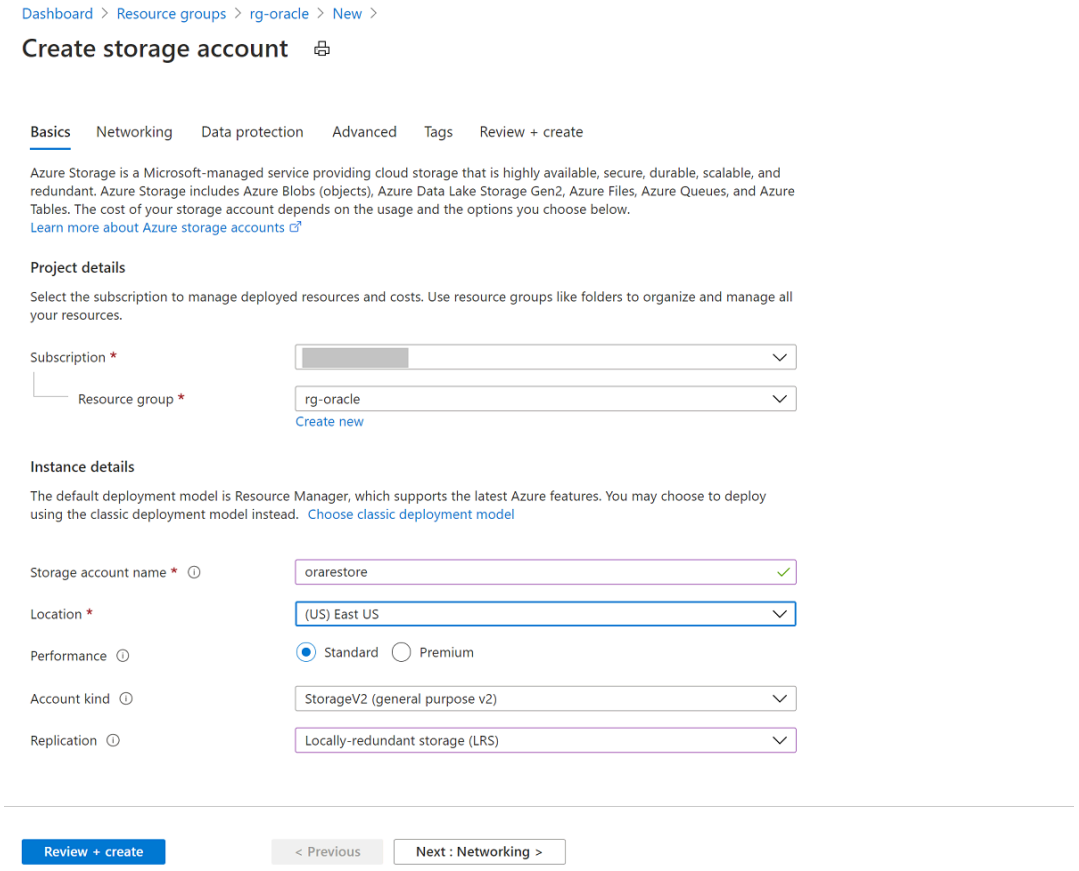
ii. For **Storage account name**, enter **oracrestore**.

iii. Ensure that **Location** is set to the same region as all your other resources in the resource group.

iv. Set **Performance** to **Standard**.

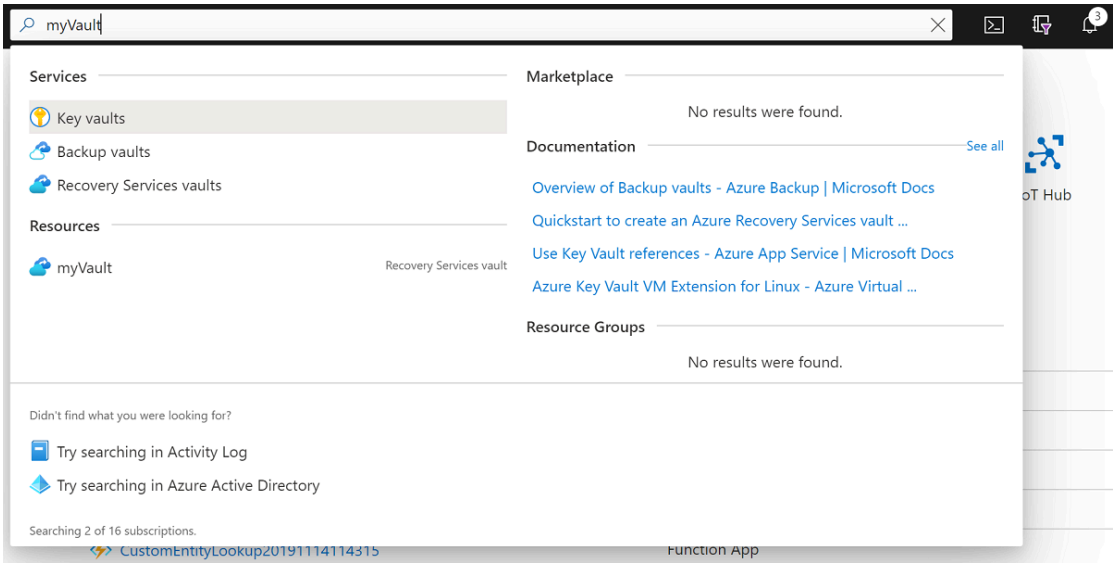
v. For **Account kind**, select **StorageV2 (general purpose v2)**.

vi. For **Replication**, select **Locally-redundant storage (LRS)**.

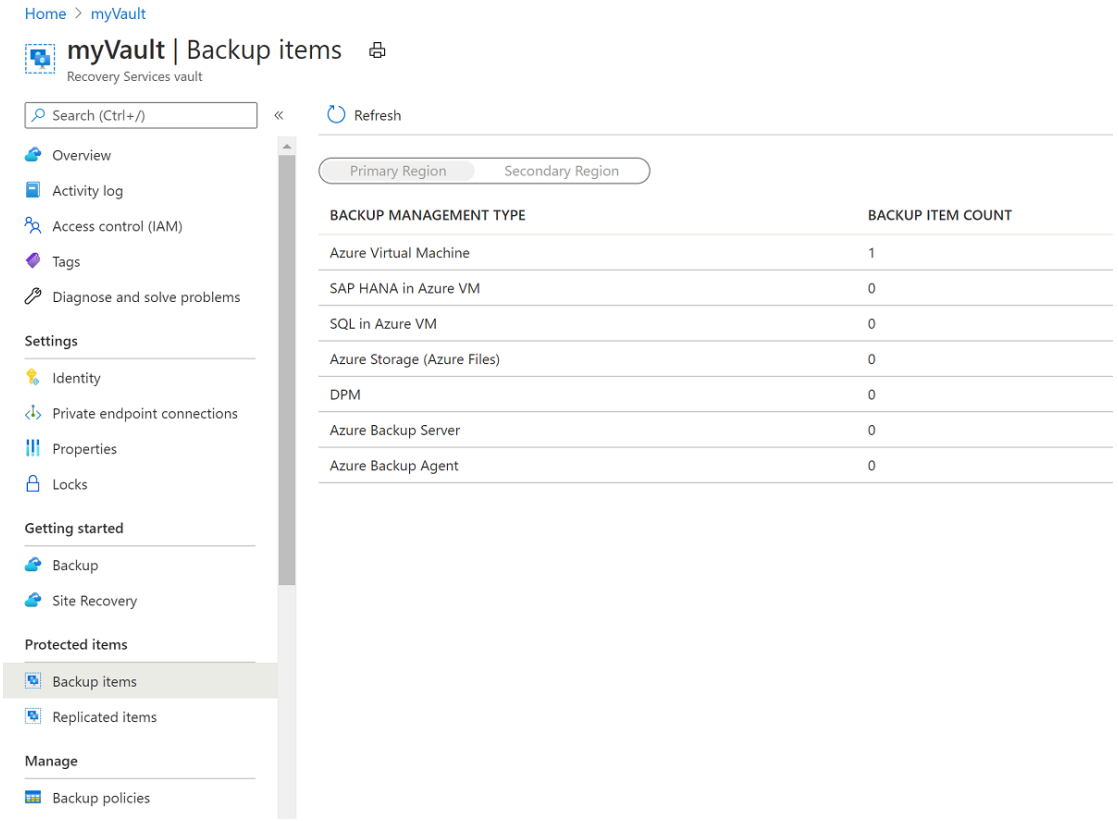


c. Select **Review + Create**, and then select **Create**.

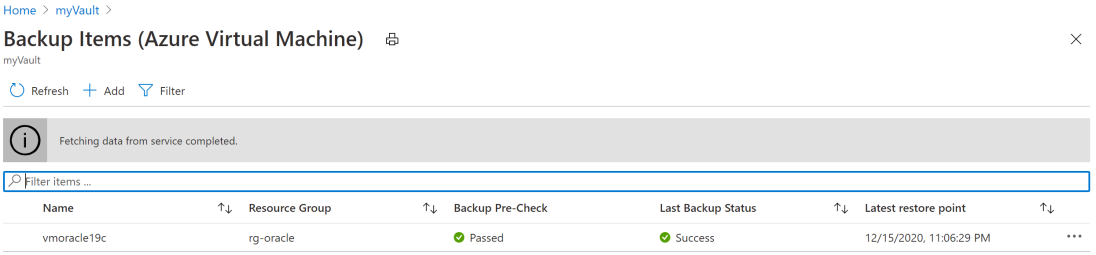
2. In the Azure portal, search for the **myVault** Recovery Services vault and select it.



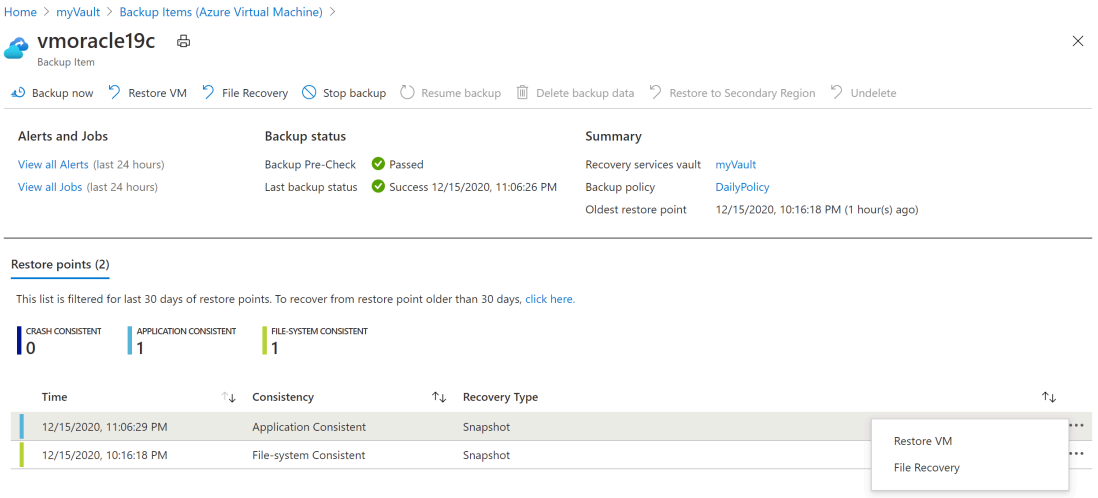
3. On the **Overview** pane, select **Backup items**. Then select **Azure Virtual Machine**, which should have a nonzero number for **BACKUP ITEM COUNT**.



4. On the **Backups Items (Azure Virtual Machine)** pane, select the **vmoracle19c** VM.



5. On the **vmoracle19c** pane, choose a restore point that has a consistency type of **Application Consistent**. Select the ellipsis (...), and then select **Restore VM**.



6. On the **Restore Virtual Machine** pane:

   a. Select **Create New**.

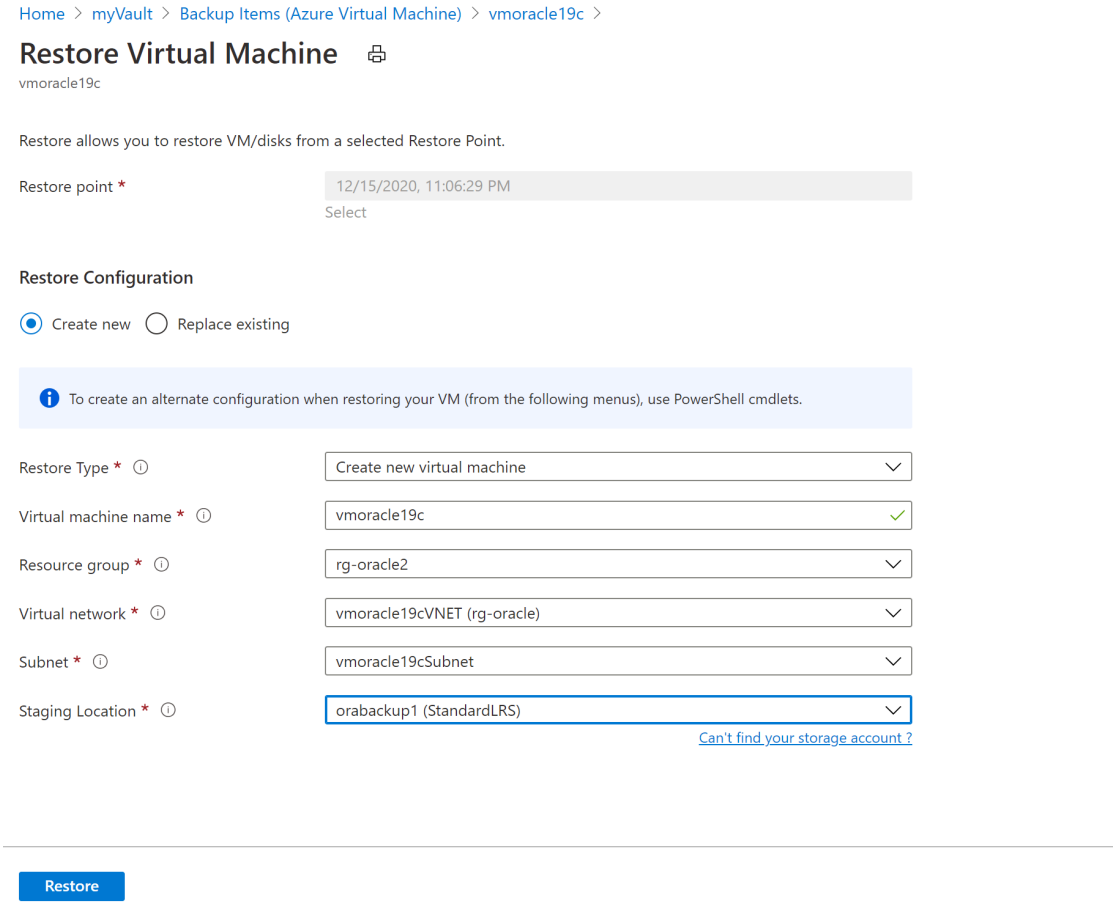
   b. For **Restore Type**, select **Create new virtual machine**.

   c. For **Virtual machine name**, enter **vmoracle19c**.

   d. For **Virtual network**, select **vmoracle19cVNET**.

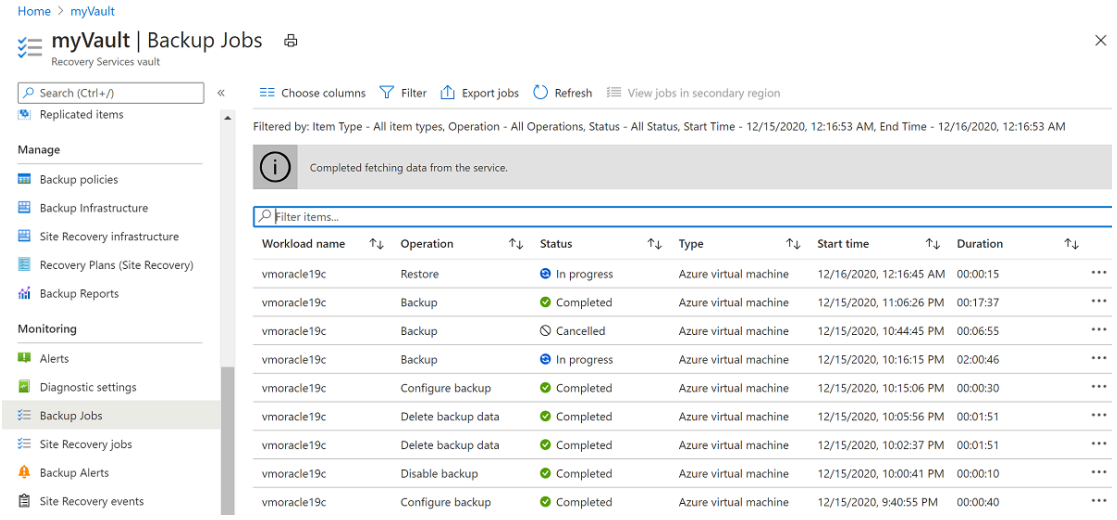
      The subnet is automatically populated based on your selection for the virtual network.

   e. For **Staging Location**, the process of restoring a VM requires an Azure storage account in the same resource group and region. You can choose a storage account or a restore task that you set up earlier.

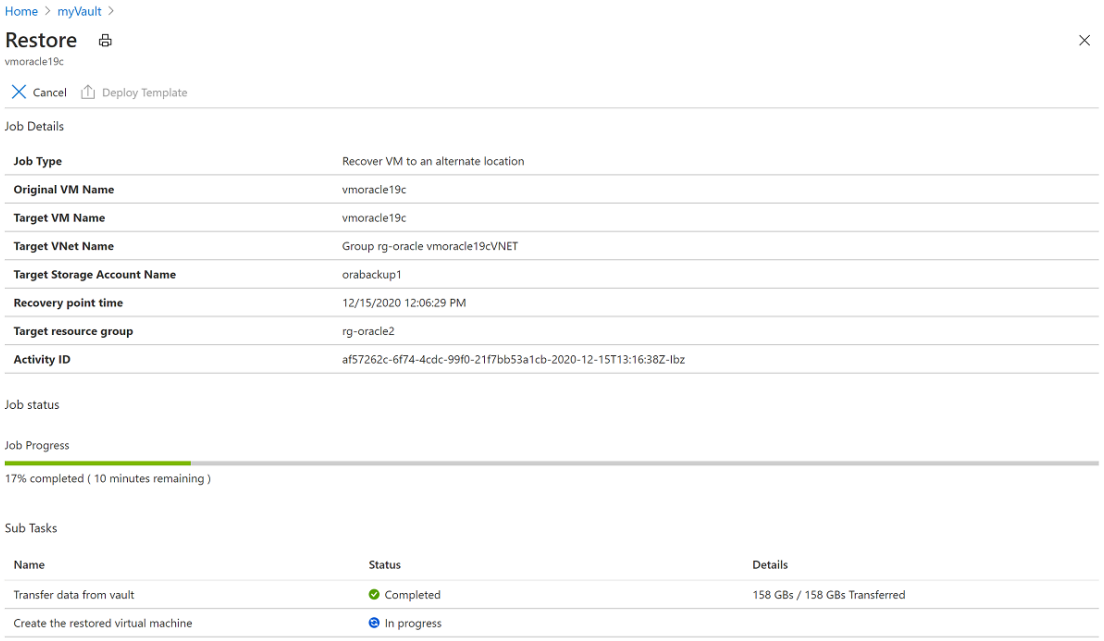


7. To restore the VM, select the **Restore** button.

8. To view the status of the restore process, select **Jobs**, and then select **Backup Jobs**.



Select the **In Progress** restore operation to show details about the status of the restore process.

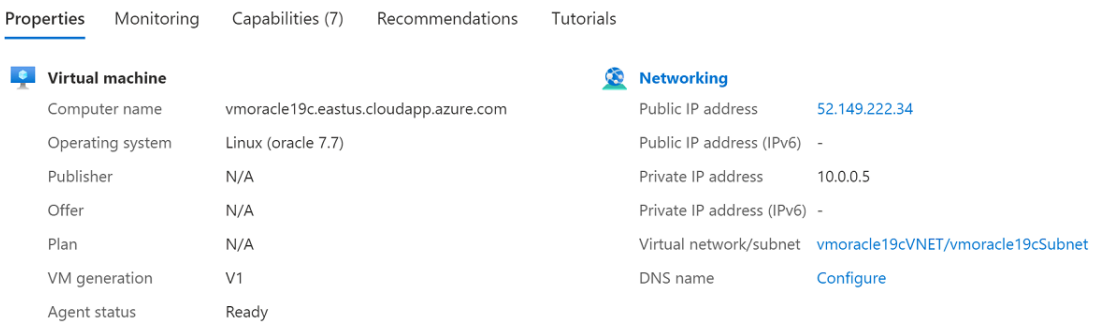


Set the public IP address

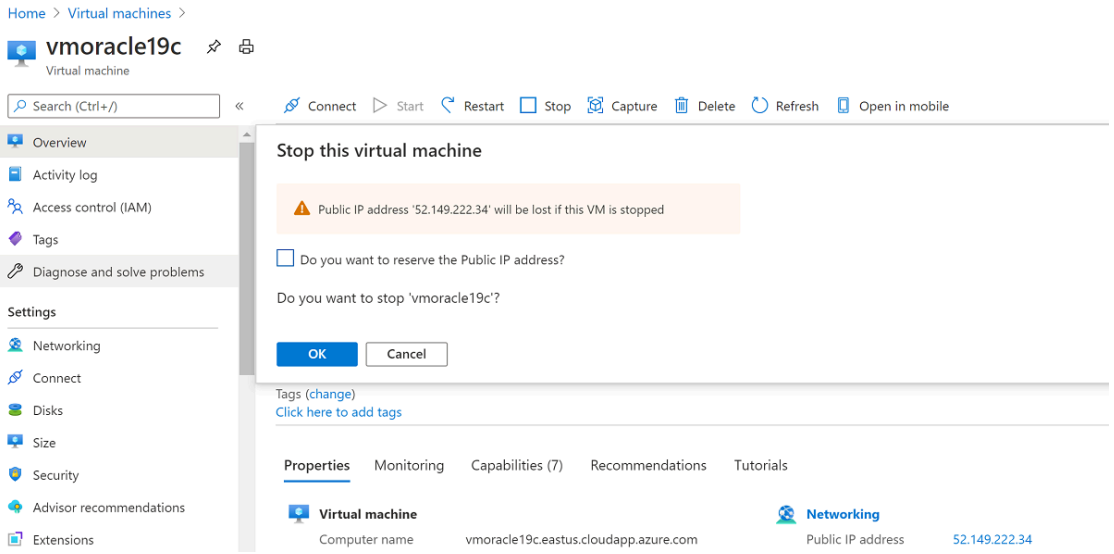
After the VM is restored, you should reassign the original IP address to the new VM.

Portal

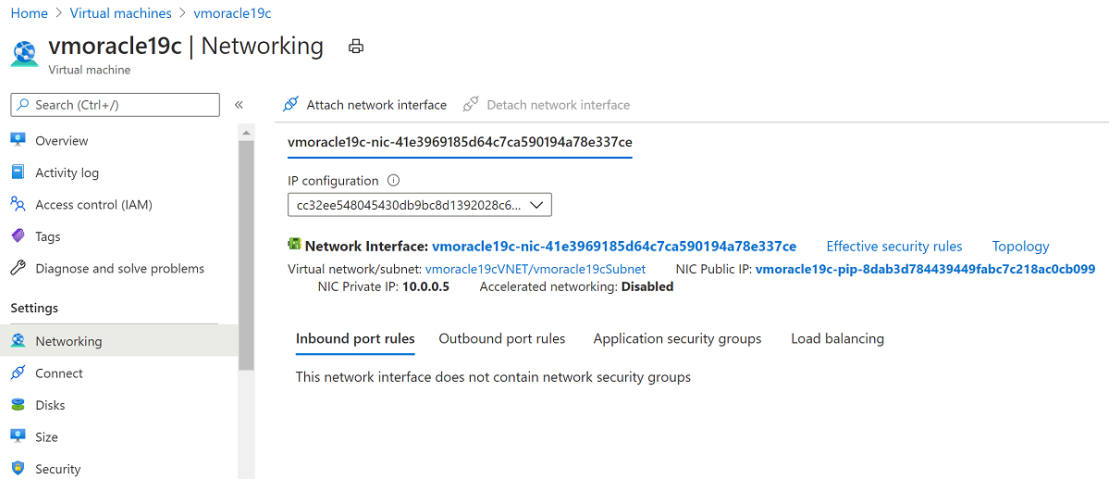
1. In the Azure portal, go to your virtual machine named **vmoracle19c**. It has been assigned a new public IP and NIC similar to **vmoracle19c-nic-XXXXXXXXXXXX**, but it doesn't have a DNS address. When the original VM was deleted, its public IP and NIC were retained. The next steps reattach them to the new VM.



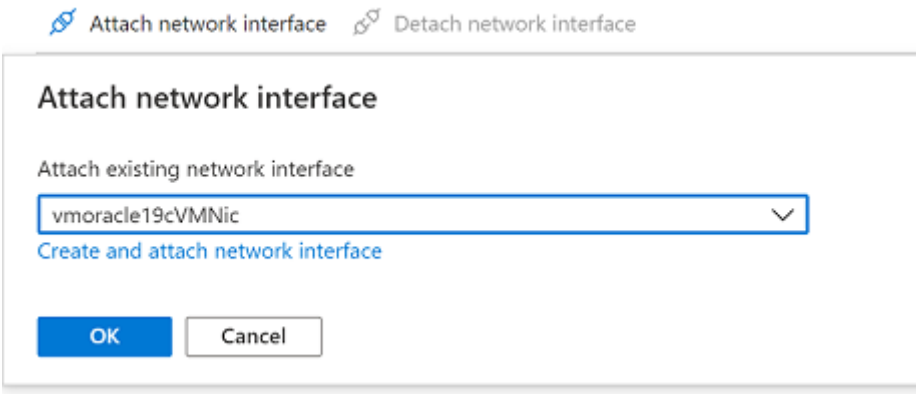
2. Stop the VM.



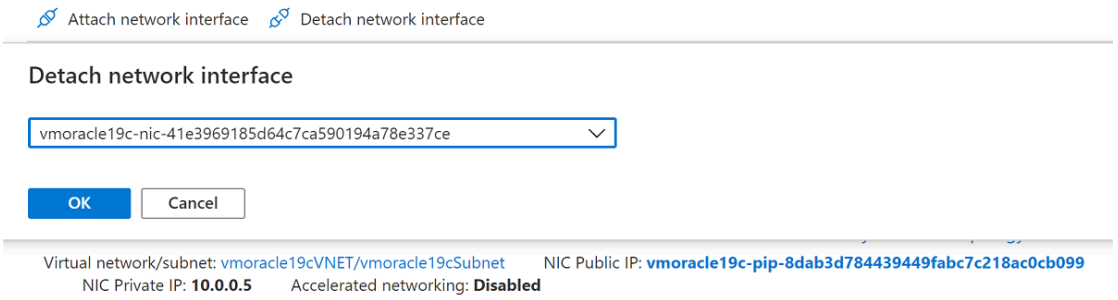
3. Go to **Networking**.



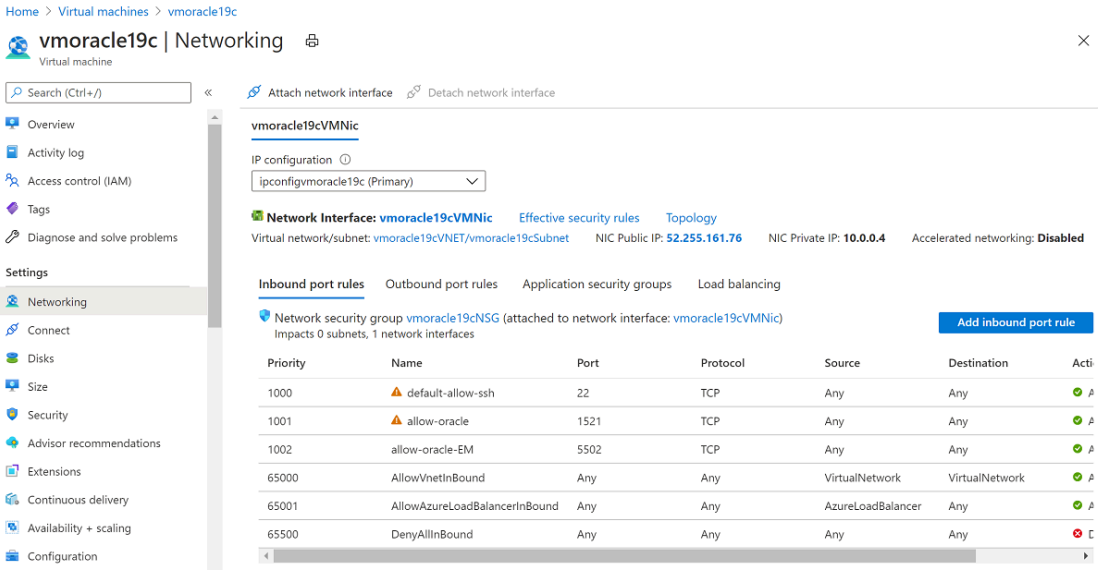
4. Select **Attach network interface**. Select the original NIC **vmoracle19cVMNic**, which the original public IP address is still associated with. Then select **OK**.



5. Detach the NIC that you created with the VM restore operation, because it's configured as the primary interface. Select **Detach network interface**, select the NIC that's similar to **vmoracle19c-nic-XXXXXXXXXXXX**, and then select **OK**.



Your re-created VM now has the original NIC, which is associated with the original IP address and network security group rules.



6. Go back to the **Overview** pane and select **Start**.

Recover the database

To recover a database after a complete VM restore:

1. Reconnect to the VM:

```
Bash

ssh azureuser@<publicIpAddress>
```

When the whole VM has been restored, it's important to recover the databases on the VM by performing the following steps on each database.

2. You might find that the instance is running, because the autostart attempted to start the database on VM startup. However, the database requires recovery and is

likely to be at the mount stage only. Run a preparatory shutdown before starting the mount stage:

```bash
sudo su - oracle
sqlplus / as sysdba
SQL> shutdown immediate
SQL> startup mount
```

3. Perform database recovery.

   It's important to specify the `USING BACKUP CONTROLFILE` syntax to inform the `RECOVER AUTOMATIC DATABASE` command that recovery should not stop at the Oracle system change number (SCN) recorded in the restored database control file.

   The restored database control file was a snapshot, along with the rest of the database. The SCN stored within it is from the point in time of the snapshot. There might be transactions recorded after this point, and you want to recover to the point of the last transaction committed to the database.

```bash
SQL> recover automatic database using backup controlfile until cancel;
```

4. When the last available archive log file has been applied, enter `CANCEL` to end recovery.

   When recovery finishes successfully, the message `Media recovery complete` appears.

   However, when you're using the `BACKUP CONTROLFILE` clause, the recover command ignores online log files. It's possible that changes in the current online redo log are required to complete point-in-time recovery. In this situation, you might see messages similar to these examples:

```output
SQL> recover automatic database until cancel using backup controlfile;
ORA-00279: change 2172930 generated at 04/08/2021 12:27:06 needed for
thread 1
ORA-00289: suggestion :
/u02/fast_recovery_area/ORATEST1/archivelog/2021_04_08/o1_mf_1_13_%u_.a
rc
ORA-00280: change 2172930 for thread 1 is in sequence #13
ORA-00278: log file
```

```
'/u02/fast_recovery_area/ORATEST1/archivelog/2021_04_08/o1_mf_1_13_%u_.
arc' no
longer needed for this recovery
ORA-00308: cannot open archived log
'/u02/fast_recovery_area/ORATEST1/archivelog/2021_04_08/o1_mf_1_13_%u_.
arc'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 7

Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

> **Important**
>
> If the current online redo log is lost or corrupted and you can't use it, you might cancel recovery at this point.

To correct this situation, you can identify which online log wasn't archived and supply the fully qualified file name to the prompt.

5. Open the database.

   The `RESETLOGS` option is required when the `RECOVER` command uses the `USING BACKUP CONTROLFILE` option. `RESETLOGS` creates a new incarnation of the database by resetting the redo history back to the beginning, because there's no way to determine how much of the previous database incarnation was skipped in the recovery.

   Bash
   ```
   SQL> alter database open resetlogs;
   ```

6. Check that the database content was recovered:

   Bash
   ```
   SQL> select * from scott.scott_table;
   ```

The backup and recovery of Oracle Database on an Azure Linux VM are now finished.

You can find more information about Oracle commands and concepts in the Oracle documentation, including:

- Performing Oracle user-managed backups of the entire database
- Performing complete user-managed database recovery

- Oracle STARTUP command
- Oracle RECOVER command
- Oracle ALTER DATABASE command
- Oracle LOG_ARCHIVE_DEST_n parameter
- Oracle ARCHIVE_LAG_TARGET parameter

Delete the VM

When you no longer need the VM, you can use the following commands to remove the resource group, the VM, and all related resources:

1. Disable soft delete of backups in the vault:

   Azure CLI

   ```
   az backup vault backup-properties set --name myVault --resource-group
   rg-oracle --soft-delete-feature-state disable
   ```

2. Stop protection for the VM and delete backups:

   Azure CLI

   ```
   az backup protection disable --resource-group rg-oracle --vault-name
   myVault --container-name vmoracle19c --item-name vmoracle19c --delete-
   backup-data true --yes
   ```

3. Remove the resource group, including all resources:

   Azure CLI

   ```
   az group delete --name rg-oracle
   ```

Next steps

Create highly available VMs

Explore Azure CLI samples for VM deployment

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Stream database backups using Oracle Recovery Manager

Article • 08/22/2024

In this article, you learn about how Azure VMs support streaming database backups with Oracle Recovery Manager (RMAN). The streaming process uses either the destination of a virtual tape library package, or writes those backups directly to a local or remote filesystem. This article describes how various virtual tape library packages are integrated with Oracle RMAN. For a few of the packages, you see links to the Azure Marketplace.

The backup and restore utility Oracle RMAN (Recovery MANager) can be configured to stream and capture back up images of Oracle databases then stream and send those back-up images to two different types of destinations.

Device type SBT

The serial backup tape (SBT) type of destination was originally designed for interacting with tape drives, though not directly. To simplify the interaction with multiple tape devices available when RMAN was created, Oracle developed an application programming interface (API) to interact with software packages to manage tape devices.

The device type SBT sends commands to software packages through its defined API. The software package vendors create corresponding “plug-ins” that interact according to the specifications of the API to translate the RMAN commands for the software package. Oracle doesn't charge more for this functionality, but various software vendors may charge licensing and support fees for their “plug-ins” to connect to the API for RMAN published by Oracle.

To use Device type SBT, the corresponding media management vendor (MMV) software must be previously installed onto the OS platform on which the Oracle database is available. Backups to the SBT installation aren't available “out of the box” following an Oracle database installation. Where there's no limit to the number of MMV packages that can be connected to an Oracle database instance, but it's exceedingly rare for there to be more than one in use at any time.

Many of these software packages, originally available for on-premises installation, are also available in the Azure Marketplace.

- CommVault
- Veritas NetBackup

- Dell PowerProtect DD Virtual Edition (DDVE)
- Veeam Backup & Replication

Other software packages can be found by searching the Azure Marketplace...

Device type disk

A more universal configuration option for Oracle RMAN is device type disk. For this option, streamed database backup images are written to OS filesystem directories directly addressable from the OS image on which the Oracle database runs. The storage used for backups is either directly mounted on the OS platform, or remotely mounted as a fileshare.

There's no extra licensing or support charges for this option because the DISK adapter for Oracle RMAN is entirely contained within the Oracle RDBMS software.

There are six storage options for Oracle RMAN backups within an Azure VM, of which five are Azure fileshares.

- Locally attached managed disk
- Azure blob over NFS
- Azure blobfuse 2.0
- Azure Files standard over CIFS/SMB
- Azure Files premium over NFS
- Azure NetApp Files

Each of these options has advantages or disadvantages in the areas of capacity, pricing, performance, durability. The following table is provided to allow easy comparison of features and prices.

Expand table

Type	Tier	Docs	Mount protocol for VM	Support model	Prices	Notes
Managed disk	Standard HDD	Introduction to Azure managed disks	SCSI	Microsoft	Managed disks pricing	1
Managed disk	Standard SSD	Introduction to Azure managed disks	SCSI	Microsoft	Managed Disks pricing	1

Type	Tier	Docs	Mount protocol for VM	Support model	Prices	Notes
Managed disk	Premium SSD	Introduction to Azure managed disks	SCSI	Microsoft	Managed Disks pricing	1
Managed disk	Premium SSD v2	Introduction to Azure managed disks	SCSI	Microsoft	Managed Disks pricing	1
Managed disk	UltraDisk	Introduction to Azure managed disks	SCSI	Microsoft	Managed Disks pricing	1
Azure blob	Block blobs	Mount Blob Storage by using the Network File System (NFS) 3.0 protocol	NFS v3.0	Microsoft	Azure Blob Storage pricing	2
Azure blobfuse	v1	How to mount Azure Blob Storage as a file system with BlobFuse v1	Fuse	Open source/GitHub	n/a	3, 5, 6
Azure blobfuse	v2	What is BlobFuse? - BlobFuse2	Fuse	Open source/GitHub	n/a	3, 5, 6
Azure Files	Standard	What is Azure Files?	SMB/CIFS	Microsoft	Azure Files pricing	4, 6
Azure Files	Premium	What is Azure Files?	SMB/CIFS, NFS v4.1	Microsoft	Azure Files pricing	4, 7
Azure NetApp Files	Standard	Azure NetApp Files	SMB/CIFS, NFS v3.0, NFS v4.1	Microsoft/NetApp	Azure NetApp Files pricing	4, 8, 11
Azure NetApp Files	Premium	Azure NetApp Files	SMB/CIFS, NFS v3.0, NFS v4.1	Microsoft/NetApp	Azure NetApp Files pricing	4, 9, 11

Type	Tier	Docs	Mount protocol for VM	Support model	Prices	Notes
Azure NetApp Files	Ultra	Azure NetApp Files	SMB/CIFS, NFS v3.0, NFS v4.1	Microsoft/NetApp	Azure NetApp Files pricing	4, 10, 11

Legend:

[1] Restricted by device-level and cumulative VM-level I/O limits on IOPS and I/O throughput.

- device limits are specified in the pricing documentation.
- cumulative limits for VM sizes are specified in the documentation Sizes for virtual machines in Azure

[2] Choose *hierarchical storage* in 1st drop-down, then *blob only* in the 2nd drop-down.

[3] Choose *flat storage* in 1st drop-down, then *blob only* in the 2nd drop-down.

[4] Uses CIFS protocol for which later versions of RHEL/OEL Linux are recommended.

- don't use lower Linux versions (that is, RHEL7/OEL7 below 7.5) for CIFS
- consider using mount option `cache=none` for Oracle archived redo log files use-case with CIFS mounts.

[5] supported on GitHub by the Azure Storage product group within Microsoft as an open source project in GitHub.

[6] *hot* usage tier recommended.

[7] *premium* usage tier recommended.

[8] I/O throughput of 16 MiB/s per TiB allocated.

[9] I/O throughput of 64 MiB/s per TiB allocated.

[10] I/O throughput of 128 MiB/s per TiB allocated.

[11] Azure NetApp Files Performance Calculator is useful for quick pricing calculations.

Next steps

Oracle performance best practices for Azure VMs

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Disaster recovery for an Oracle Database 12c database in an Azure environment

Article • 08/22/2024

Applies to: ✔ Linux VMs

Assumptions

- You have an understanding of Oracle Data Guard design and Azure environments.

Goals

- Design the topology and configuration that meet your disaster recovery (DR) requirements.

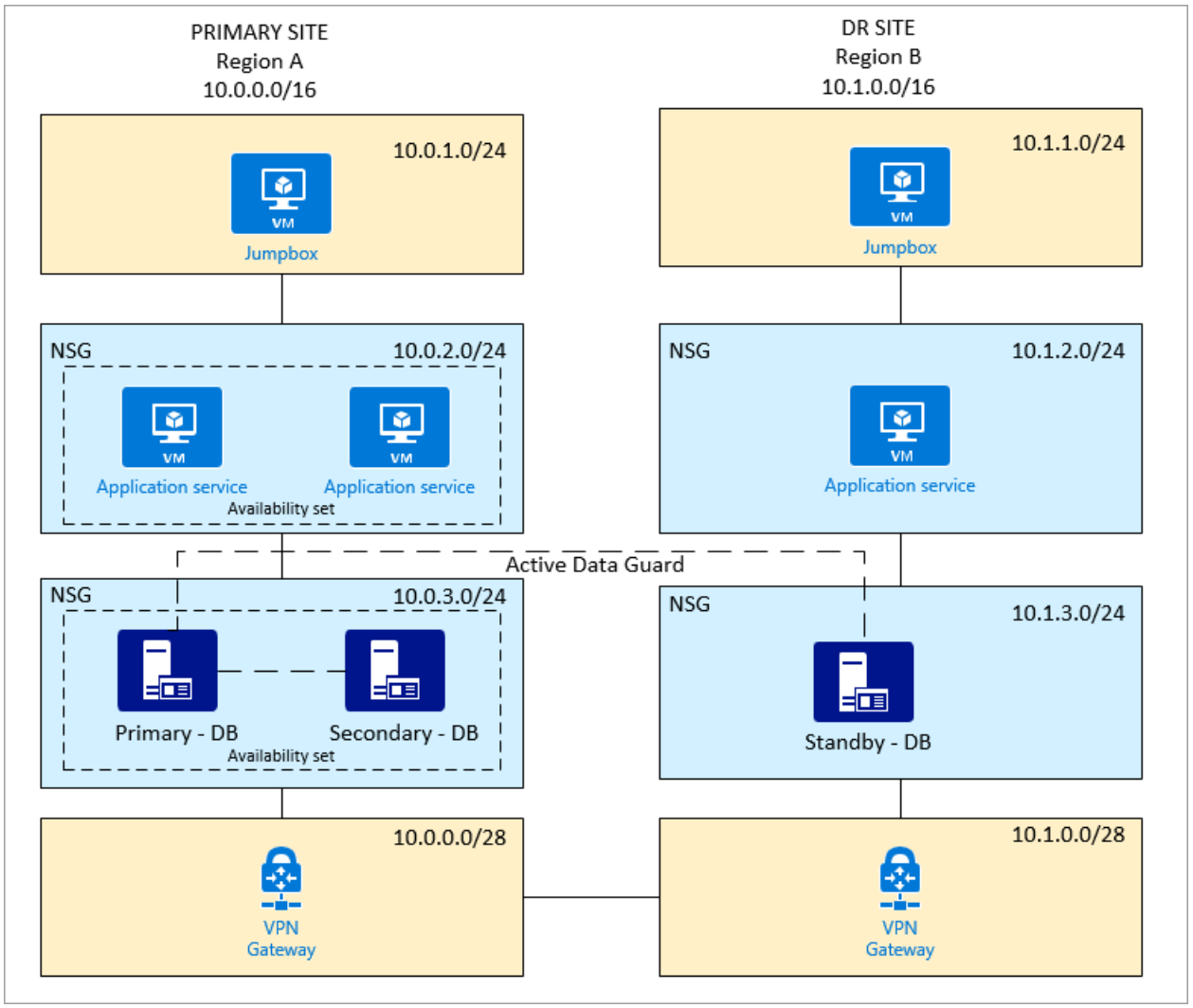
Scenario 1: Primary and DR sites on Azure

A customer has an Oracle database set up on the primary site. A DR site is in a different region. The customer uses Oracle Data Guard for quick recovery between these sites. The primary site also has a secondary database for reporting and other uses.

Topology

Here is a summary of the Azure setup:

- Two sites (a primary site and a DR site)
- Two virtual networks
- Two Oracle databases with Data Guard (primary and standby)
- Two Oracle databases with Golden Gate or Data Guard (primary site only)
- Two application services, one primary and one on the DR site
- An *availability set,* which is used for database and application service on the primary site
- One jumpbox on each site, which restricts access to the private network and only allows sign-in by an administrator
- A jumpbox, application service, database, and VPN gateway on separate subnets
- NSG enforced on application and database subnets



Scenario 2: Primary site on-premises and DR site on Azure

A customer has an on-premises Oracle database setup (primary site). A DR site is on Azure. Oracle Data Guard is used for quick recovery between these sites. The primary site also has a secondary database for reporting and other uses.

There are two approaches for this setup.

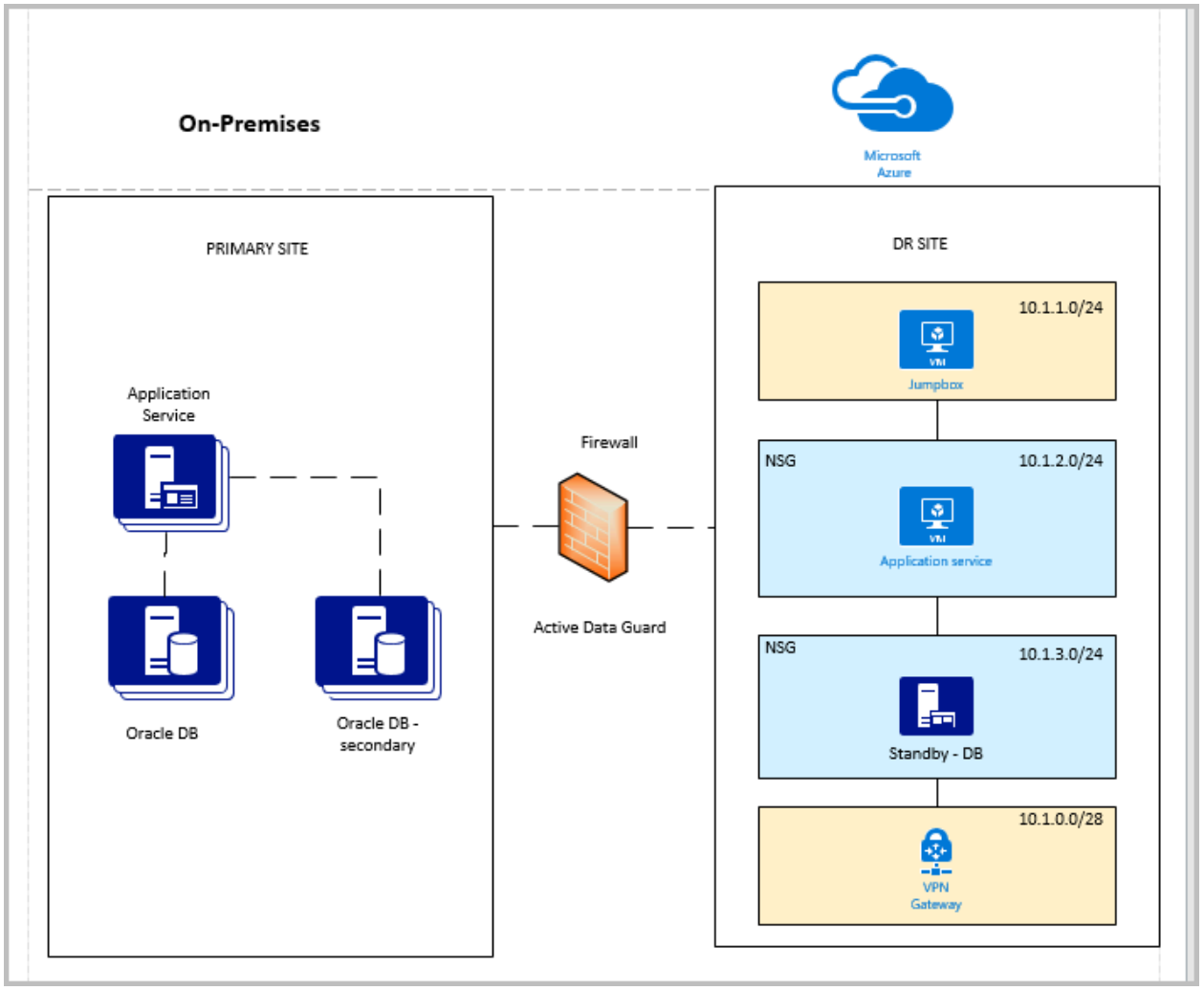
Approach 1: Direct connections between on-premises and Azure, requiring open TCP ports on the firewall

We don't recommend direct connections because they expose the TCP ports to the outside world.

Topology

Following is a summary of the Azure setup:

- One DR site
- One virtual network
- One Oracle database with Data Guard (active)
- One application service on the DR site
- One jumpbox, which restricts access to the private network and only allows sign-in by an administrator
- A jumpbox, application service, database, and VPN gateway on separate subnets
- NSG enforced on application and database subnets
- An NSG policy/rule to allow inbound TCP port 1521 (or a user-defined port)
- An NSG policy/rule to restrict only the IP address/addresses on-premises (DB or application) to access the virtual network



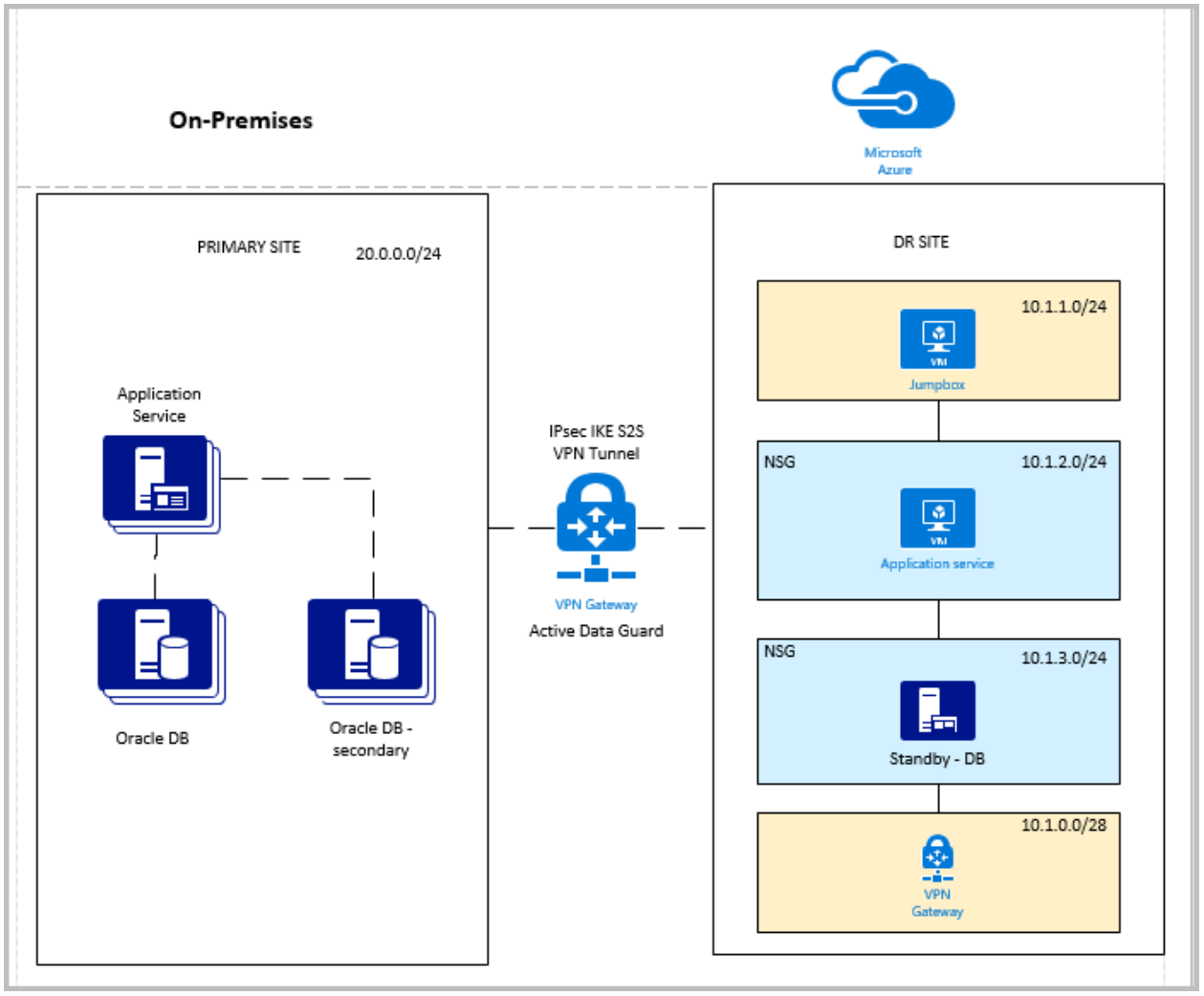
Approach 2: Site-to-site VPN

Site-to-site VPN is a better approach. For more information about setting up a VPN, see Create a virtual network with a Site-to-Site VPN connection using CLI.

Topology

Following is a summary of the Azure setup:

- One DR site
- One virtual network
- One Oracle database with Data Guard (active)
- One application service on the DR site
- One jumpbox, which restricts access to the private network and only allows sign-in by an administrator
- A jumpbox, application service, database, and VPN gateway are on separate subnets
- NSG enforced on application and database subnets
- Site-to-site VPN connection between on-premises and Azure



Additional reading

- Design and implement an Oracle database on Azure
- Configure Oracle Data Guard
- Configure Oracle Golden Gate
- Oracle backup and recovery

Next steps

- Tutorial: Create highly available VMs
- Explore VM deployment Azure CLI samples

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Architectures for Oracle applications with database on Azure Virtual Machines

Article • 08/22/2024

This article provides reference architecture to deploy Oracle application on Azure IaaS where the Oracle database also resides or is colocated.

Oracle workloads comprise not only Oracle databases, but also of Oracle first-party applications such as Siebel, PeopleSoft, JD Edwards, E-Business Suite, or customized WebLogic server applications. Deploying Oracle applications on Azure Infrastructure as a Service (IaaS) is a common scenario for organizations looking to use the cloud for their Oracle workloads along with Oracle database. Microsoft offers reference architectures and best practices to ease this process.

General application migration guidelines

As Oracle applications move on Azure IaaS, there are common design considerations, which must be followed irrespective of type of applications. Some considerations are specific to applications. In this section, we're listing common design considerations of all applications, and any application specific considerations are covered under each application.

Network and security

The provided network settings for Oracle Applications on Azure cover various aspects of network and security considerations. Here's a breakdown of the recommended network settings:

- Single sign-on (SSO) with Microsoft Entra ID and SAML: Use Microsoft Entra ID for single sign-on (SSO) using the Security Assertions Markup Language (SAML) protocol. This SSO allows users to authenticate once and access multiple services seamlessly.
- Microsoft Entra application proxy: Consider using Microsoft Entra application proxy, especially for remote users. This proxy allows you to securely access on-premises applications from outside your network.

- Routing Internal Users through ExpressRoute: For internal users, route traffic through Azure ExpressRoute for a dedicated, private connection to Azure services, ensuring low-latency and secure communication.

- Azure Firewall: If necessary, you can configure Azure Firewall in front of your application for added security. Azure Firewall helps protect your resources from unauthorized access and threats.

- Application Gateway for External Users: When external users need to access your application, consider using Azure Application Gateway. It supplies Web Application Firewall (WAF) capabilities for protecting your web applications and Layer 7 load balancing to distribute traffic.

- Network Security Groups (NSG): Secure your subnets by using Network Security Groups (NSG). NSGs allow you to control inbound and outbound traffic to network interfaces, Virtual Machines, and subnets by defining security rules.

- Role-Based Access Control (RBAC): To grant access to specific individuals or roles, use Azure Role-Based Access Control (RBAC). RBAC provides fine-grained access control to Azure resources based on roles and permissions.

- Bastion Host for SSH Access: Use a Bastion host as a jump box to enhance security for SSH access. A Bastion host acts as a secure gateway for administrators to access Virtual Machines in the virtual network. This host provides an added layer of security.

- More considerations:
  - Data Encryption: Ensure that data at rest and in transit is encrypted. Azure provides tools like Azure Disk Encryption and SSL/TLS for this purpose.
  - Patch Management: Regularly update and patch your EBS environment to protect against known vulnerabilities.
  - Monitoring and Logging: Implement Azure Monitor and Azure Defender for security to continuously check your environment for security threats and anomalies. Set up logging for auditing and forensic analysis.

- In summary, these network and security settings aim to provide a robust and secure environment for hosting Oracle applications on Azure IaaS. They incorporate best practices for authentication, access control, and network security, both for internal and external users. They also consider the need for SSH access to Application servers. These recommendations can help you set up a mature security posture for your Oracle applications deployment on Azure IaaS.

Web Tier: The web tier load balances the requests and sends the requests accordingly to the application tier, database tier and/or backup.

Application tier: The application tier typically involves application servers and shared file systems.

For autoscaling, Virtual Machine Scale Sets can be a great choice for scale-out multiple Virtual Machines based on demand with custom scaling rules to adapt to your workload.

Collaborate with Azure Subject Matter Experts (SMEs) to perform a thorough assessment of your architecture. They can help you determine the most suitable Azure services based on your specific requirements, including performance, availability, and scalability. Remember to consider factors like cost, data security, compliance, and disaster recovery when designing your architecture.

It's also essential to check and optimize your Azure resources continuously to ensure efficiency and cost-effectiveness.

Load Balancing and Throughput: It's important to evaluate the workload characteristics of application servers. Some servers handle more tasks and create higher throughput than others. This information is crucial when designing your Azure Virtual machine Scale Sets and load balancing configuration to ensure that resources are allocated effectively

Database Tier: HA architectures are recommended with Oracle Data Guard for Oracle on Azure IaaS. Applications require specific type of HA setup and are listed under each application.

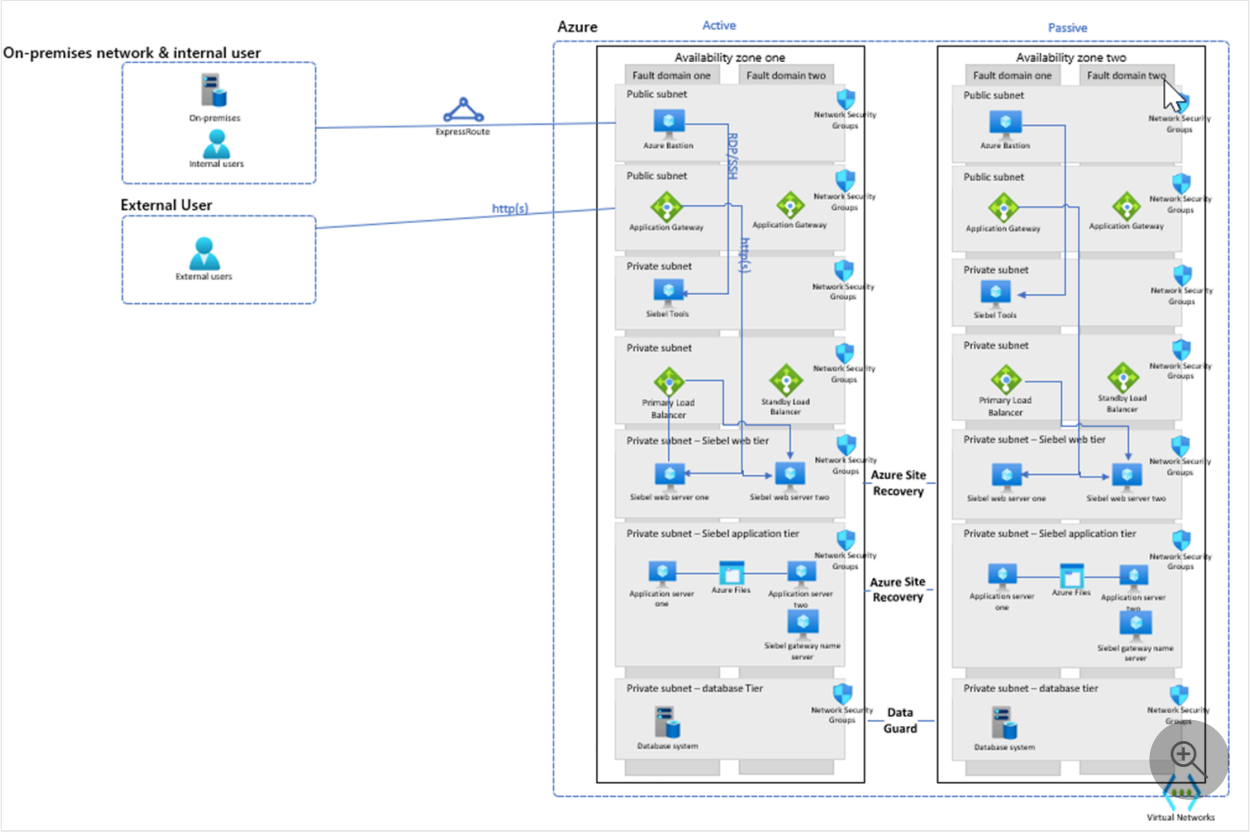
Backup - Backups are sent from the application tier and the database tier. It's just one of many reasons why those two tiers shouldn't be separated into two different vendors. Backups of the database are performed by Azure Backup Volume Snapshot on Premium Files to the secondary region.

Disaster Recovery - There are different solutions you can choose from. It very much depends on your requirements. The architecture is built to be highly available. For replicating the application tier, you can use Azure Site Recovery. Another solution you can choose is Redundancy options for managed disks. Both solutions replicate your data. Redundancy options for managed disks are a solution that can simplify the architecture but also comes with a few limitations.

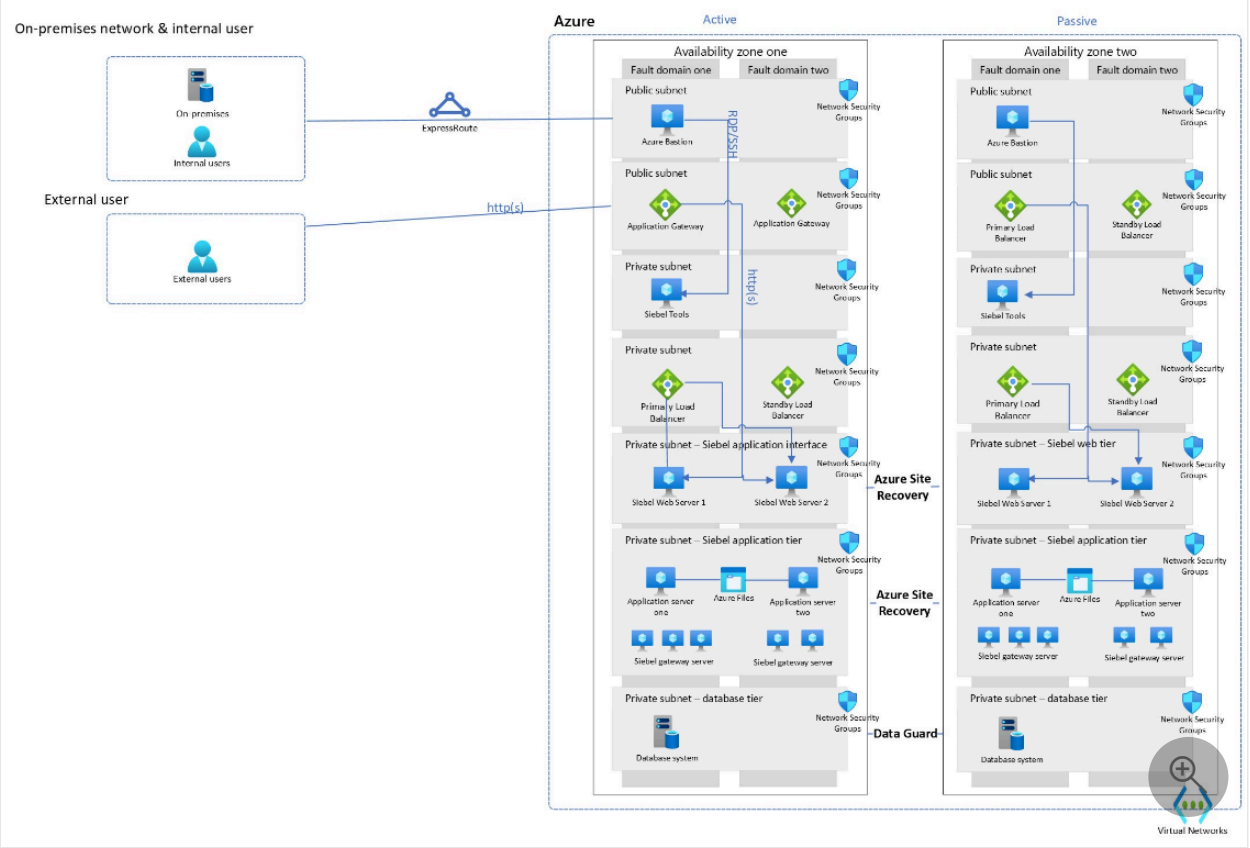
Siebel on Azure

Oracle Siebel CRM continues to be a preferred enterprise grade CRM solution by many enterprises. It's one of the most complex applications in Oracle's portfolio delivering a combination of transactional, analytical, and engagements features to manage customer facing operations.

Here's the recommended architecture of a Siebel application deployment on Azure Virtual Machines for Innovation Pack 16 and earlier:



The following diagram is architecture of a Siebel application deployment on Azure Virtual Machines for Innovation Pack 17 and earlier:



Oracle Siebel design considerations

- Network & Security: The network settings for Oracle Siebel on Azure required to follow the general network & security considerations, additionally.
- Migration must be done using Siebel Tool subnet.

Application Tier

- Version 17 or higher – configurations of certain server and utilities on the application and database is required.

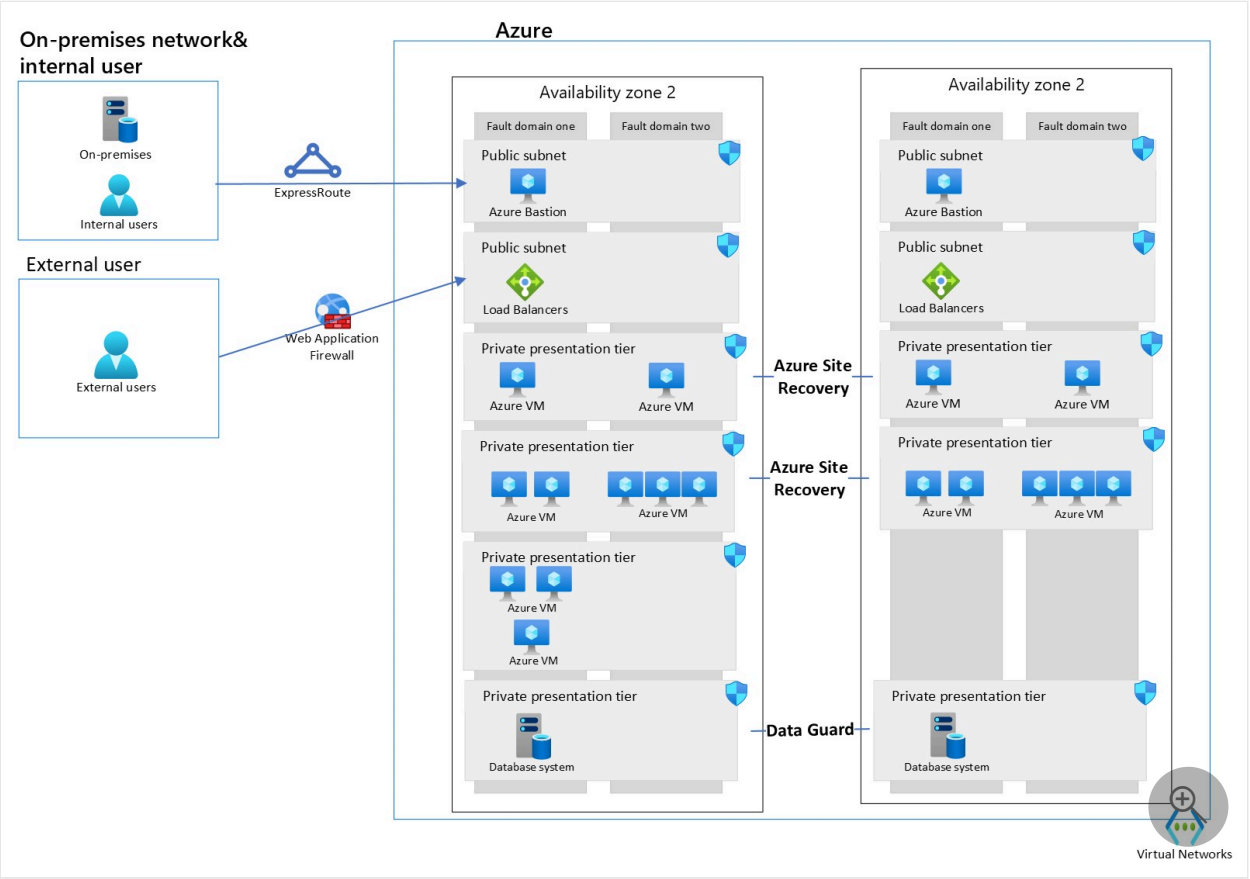
Database Tier

- Ensure Database and Siebel version match.
- Primary and replicated to a secondary using Data Guard based recommended Oracle reference architecture.

E-Business suite on Azure

Oracle E-Business Suite (EBS) is a suite of applications including Supply Chain Management (SCM) and Customer Relationship Management (CRM). As EBS is an SCM and CRM system, it usually has many interfaces to third-party systems. The below architecture is built to be highly available within one region.

We assume that external users don't cross the corporate network in the following diagram.



Oracle EBS design considerations

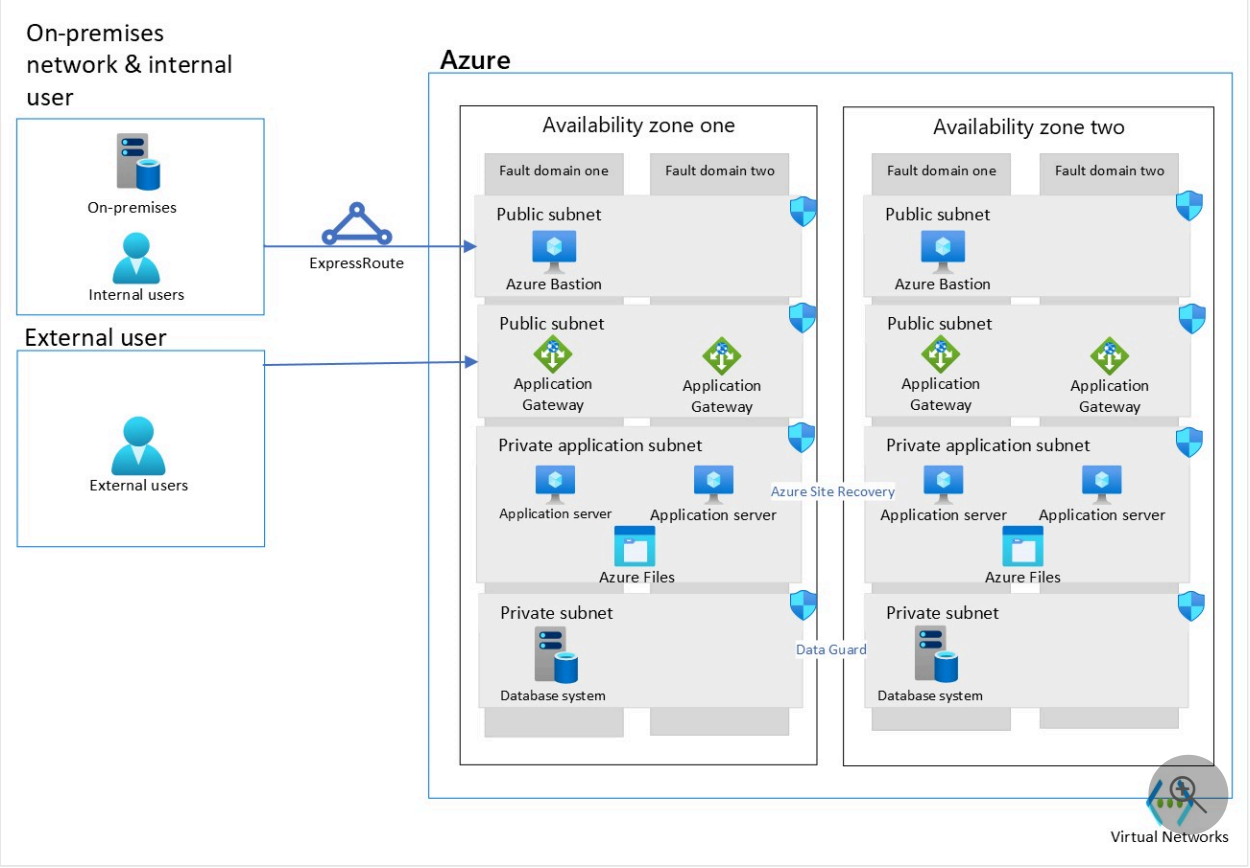
Database Tier - Primary & secondary database should be within one datacenter, the synchronous configuration should be used. If you install your application across datacenters, you should configure Data Guard in Asynchronous mode.

JD Edwards on Azure

Oracle's JD Edwards is an integrated applications suite of comprehensive enterprise resource planning software. We have seen JDE used in Supply chain, Warehouse Management, Logistics, Manufacturing resource planning and more. Because of the use of the application, we see that interfaces to other systems are important as well.

The following architecture is built to being highly available. We assumed that external users aren't accessing over the corporate network. If an external user accesses the application using corporate network, the architecture can be simplified on networking as

follows.



JD Edwards design considerations

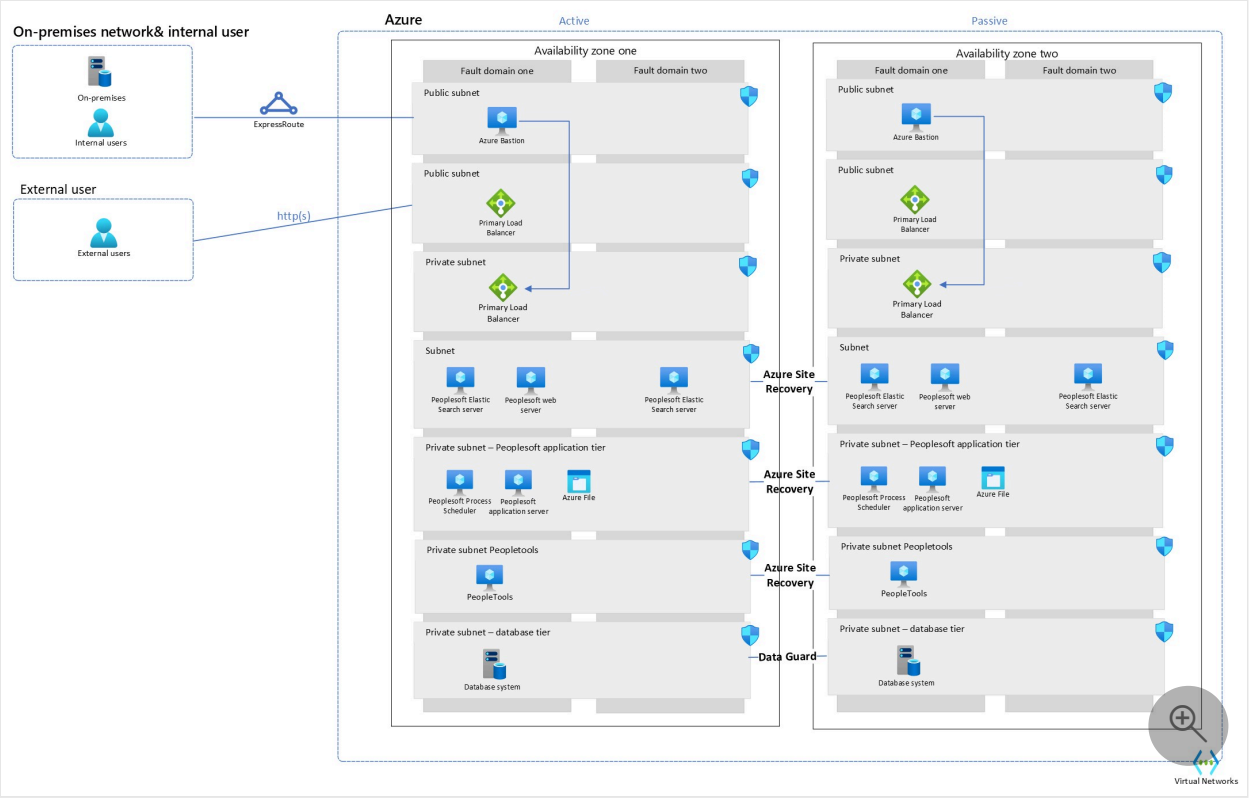
Web Tier: The application web tier typically consists of multiple application servers. In JD Edwards, rules are often saved on these application web servers.

- Presentation Tier: Each instance in the presentation tier is associated with storage. Cutting dependencies between instances can lead to high latencies, so it's crucial to assess them carefully.
- Server Performance Variation: Some servers can handle more tasks and create higher throughput than others. During the design phase, it's essential to evaluate this throughput variation to ensure that your infrastructure can handle peak workloads efficiently.
- Rearchitecture: Using Azure Virtual machine Scale Sets for autoscaling doesn't require a rearchitecture of your JD Edwards setup. It's a scalable solution that can be implemented without significant changes to your application's architecture.

Database Tier - Primary and secondary stay within one datacenter, the synchronous configuration should be used. If you install your application across datacenters, you should configure Data Guard in Asynchronous mode. Data from the database tier are sent directly to an Azure Storage. The Storage is dependent on your current architecture setup.

PeopleSoft on Azure

Oracle's PeopleSoft application suite contains software for human resources and financial management. The application suite is multi-tiered, and the applications include human resource management systems (HRMS), customer relationship management (CRM), financials and supply chain management (FSCM), and enterprise performance management (EPM).



PeopleSoft design considerations

Application Tier: The application tier contains several tasks and servers. It runs the business logic and processes but also maintains the connection to the database. As soon as this dependency is cut, it causes latencies.

- Dependency between Application and Database Tiers: It's important to minimize latency between the application and database tiers. By placing the application and database-tier in the same cloud provider (Azure, in this case), you reduce network latency. Azure provides various networking options and services like Virtual Network (VNet) peering or ExpressRoute to ensure low-latency connections between tiers.

- Operating System Considerations: If the Process Scheduler specifically requires Windows operating systems, you can still run it on Azure Virtual Machines. Azure supports various Windows Server versions, allowing you to choose the one that meets your application's requirements.

- Architecture Evaluation: Carefully evaluate your architecture requirements, including scalability, availability, and performance. Consider setting up multiple application server instances in a load-balanced configuration to ensure high availability and scalability.

Database Tier - The primary and replicated to a secondary should stay within one datacenter, the synchronous configuration should be used. If you install your application across datacenters, you should configure Data Guard in Asynchronous mode.

Next steps

Reference architectures for Oracle Database

Migrate Oracle workload to Azure Virtual Machines

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Architectures for Oracle applications with Azure Virtual Machines with database on OCI

Article • 08/23/2024

Applies to: ✔ Linux VMs

Microsoft and Oracle have worked together to enable customers to deploy Oracle applications such as Oracle E-Business Suite, JD Edwards EnterpriseOne, and PeopleSoft in the cloud. With the introduction of the preview private network interconnectivity between Microsoft Azure and Oracle Cloud Infrastructure (OCI), Oracle applications can now be deployed on Azure with their back-end databases in Azure or OCI. Oracle applications can also be integrated with Microsoft Entra ID, allowing you to set up single sign-on so that users can sign into the Oracle application using their Microsoft Entra credentials.

OCI offers multiple Oracle database options for Oracle applications, including DBaaS, Exadata Cloud Service, Oracle RAC, and Infrastructure-as-a-Service (IaaS). Currently, Autonomous Database isn't a supported back-end for Oracle applications.

There are multiple options for deploying Oracle applications in Azure, including in a highly available and secure manner. Azure also offers Oracle database VM images that you can deploy if you choose to run your Oracle applications entirely on Azure.

The following sections outline architecture recommendations by both Microsoft and Oracle to deploy Oracle E-Business Suite, JD Edwards EnterpriseOne, and PeopleSoft in a cross-cloud configuration or entirely in Azure. Microsoft and Oracle have tested these applications and confirmed that the performance meets standards set by Oracle for these applications.

Architecture considerations

Oracle applications are made up of multiple services, which can be hosted on the same or multiple virtual machines in Azure and optionally in OCI.

Application instances can be set up with private or public endpoints. Microsoft and Oracle recommend setting up a *bastion host VM* with a public IP address in a separate subnet for management of the application. Then, assign only private IP addresses to the other machines, including the database tier.

When setting up an application in a cross-cloud architecture, planning is required to ensure that the IP address space in the Azure virtual network doesn't overlap the private IP address space in the OCI virtual cloud network.

For added security, set up network security groups at a subnet level to ensure only traffic on specific ports and IP addresses is permitted. For example, machines in the middle tier should only receive traffic from within the virtual network. No external traffic should reach the middle tier machines directly.

For high availability, you can set up redundant instances of the different servers in the same availability set or different availability zones. Availability zones allow you to achieve a 99.99% uptime SLA, while availability sets allow you to achieve a 99.95% uptime SLA in-region. Sample architectures shown in this article are deployed across two availability zones.

When deploying an application using the cross-cloud interconnect, you may continue to use an existing ExpressRoute circuit to connect your Azure environment to your on-premises network. However, you need a separate ExpressRoute circuit for the interconnect to OCI than the one connecting to your on-premises network.

E-Business Suite

Oracle E-Business Suite (EBS) is a suite of applications including Supply Chain Management (SCM) and Customer Relationship Management (CRM). To take advantage of OCI’s managed database portfolio, EBS can be deployed using the cross-cloud interconnect between Microsoft Azure and OCI. In this configuration, the presentation and application tiers run in Azure and the database tier in OCI, as illustrated in the following architecture diagram (Figure 1).

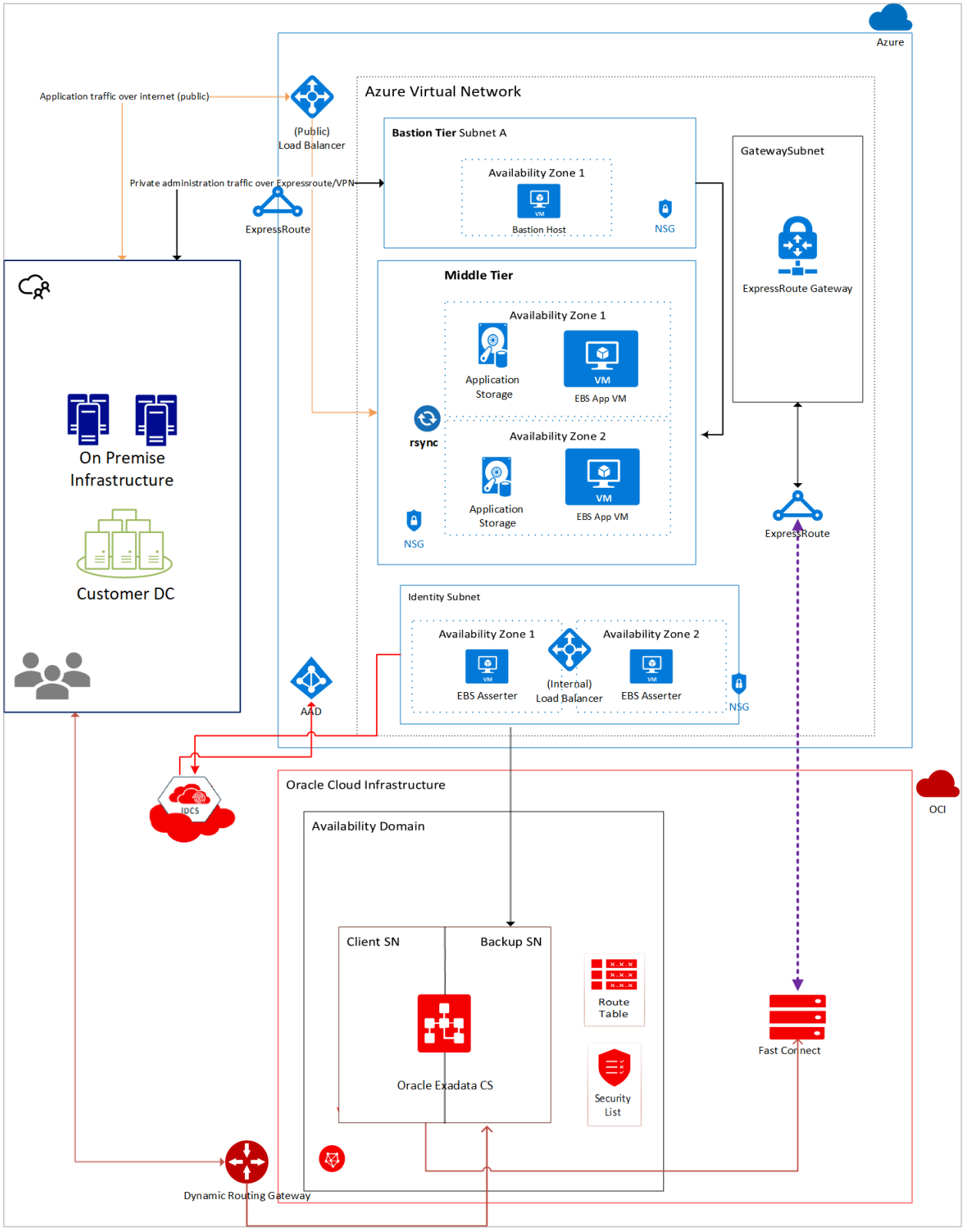


Figure 1: E-Business Suite cross-cloud architecture

In this architecture, the virtual network in Azure is connected to a virtual cloud network in OCI using the cross-cloud interconnect. The application tier is set up in Azure, whereas the database is set up in OCI. It's recommended to deploy each component to its own subnet with network security groups to allow traffic only from specific subnets on specific ports.

The architecture can also be adapted for deployment entirely on Azure with highly available Oracle databases configured using Oracle Data Guard in two availability zones in a region. The following diagram (Figure 2) is an example of this architectural pattern:

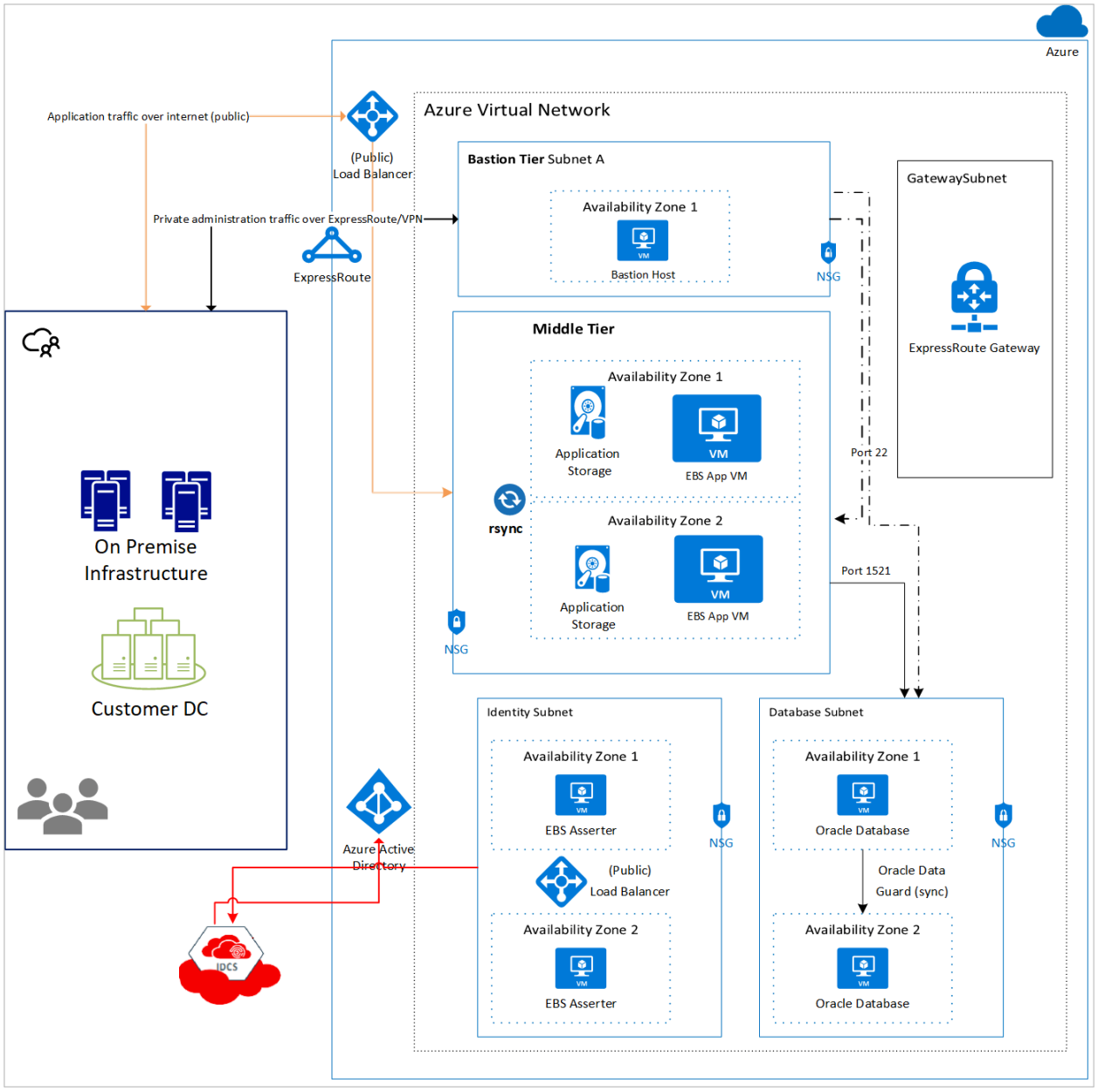


Figure 2: E-Business Suite Azure-only architecture

The following sections describe the different components at a high level.

Bastion tier

The bastion host is an optional component that you can use as a jump server to access the application and database instances. The bastion host VM can have a public IP address assigned to it, although the recommendation is to set up an ExpressRoute connection or site-to-site VPN with your on-premises network for secure access. Additionally, only SSH (port 22, Linux) or RDP (port 3389, Windows Server) should be

opened for incoming traffic. For high availability, deploy a bastion host in two availability zones or in a single availability set.

You may also enable SSH agent forwarding on your VMs, which allows you to access other VMs in the virtual network by forwarding the credentials from your bastion host. Or, use SSH tunneling to access other instances.

Here's an example of agent forwarding:

```
ssh -A -t user@BASTION_SERVER_IP ssh -A root@TARGET_SERVER_IP`
```

This command connects to the bastion and then immediately runs `ssh` again, so you get a terminal on the target instance. You may need to specify a user other than root on the target instance if your cluster is configured differently. The `-A` argument forwards the agent connection so your private key on your local machine is used automatically. Note that agent forwarding is a chain, so the second `ssh` command also includes `-A` so that any subsequent SSH connections initiated from the target instance also use your local private key.

Application (middle) tier

The application tier is isolated in its own subnet. There are multiple virtual machines set up for fault tolerance and easy patch management. These VMs can be backed by shared storage, which is offered by Azure NetApp Files and Ultra SSDs. This configuration allows for easier deployment of patches without downtime. The machines in the application tier should be fronted by a public load balancer so that requests to the EBS application tier are processed even if one machine in the tier is offline due to a fault.

Load balancer

An Azure load balancer allows you to distribute traffic across multiple instances of your workload to ensure high availability. In this case, a public load balancer is set up, because users are allowed to access the EBS application over the web. The load balancer distributes the load to both machines in the middle tier. For added security, allow traffic only from users accessing the system from your corporate network using a site-to-site VPN or ExpressRoute and network security groups.

Database tier

This tier hosts the Oracle database and is separated into its own subnet. It's recommended to add network security groups that only permit traffic from the application tier to the database tier on the Oracle-specific database port 1521.

Microsoft and Oracle recommend a high availability setup. High availability in Azure can be achieved by setting up two Oracle databases in two availability zones with Oracle Data Guard, or by using Oracle Database Exadata Cloud Service in OCI. When using Oracle Database Exadata Cloud Service, your database is deployed in two subnets. You may also set up Oracle Database in VMs in OCI in two availability domains with Oracle Data Guard.

Identity tier

The identity tier contains the EBS Asserter VM. EBS Asserter allows you to synchronize identities from Oracle Identity Cloud Service (IDCS) and Microsoft Entra ID. The EBS Asserter is needed because EBS doesn't support single sign-on protocols like SAML 2.0 or OpenID Connect. The EBS Asserter consumes the OpenID connect token (generated by IDCS), validates it, and then creates a session for the user in EBS.

While this architecture shows IDCS integration, Microsoft Entra ID unified access and single sign-on also can be enabled with Oracle Access Manager with Oracle Internet Directory or Oracle Unified Directory. For more information, see the whitepapers on Deploying Oracle EBS with IDCS Integration or Deploying Oracle EBS with OAM Integration.

For high availability, it's recommended that you deploy redundant servers of the EBS Asserter across multiple availability zones with a load balancer in front of it.

Once your infrastructure is set up, E-Business Suite can be installed by following the installation guide provided by Oracle.

JD Edwards EnterpriseOne

Oracle's JD Edwards EnterpriseOne is an integrated applications suite of comprehensive enterprise resource planning software. It's a multi-tiered application that can be set up with either an Oracle or SQL Server database backend. This section discusses details on deploying JD Edwards EnterpriseOne with an Oracle database back-end either in OCI or in Azure.

In the following recommended architecture (Figure 3), the administration, presentation, and middle tiers are deployed to the virtual network in Azure. The database is deployed in a virtual cloud network in OCI.

As with E-Business Suite, you can set up an optional bastion tier for secure administrative purposes. Use the bastion VM host as a jump server to access the application and database instances.

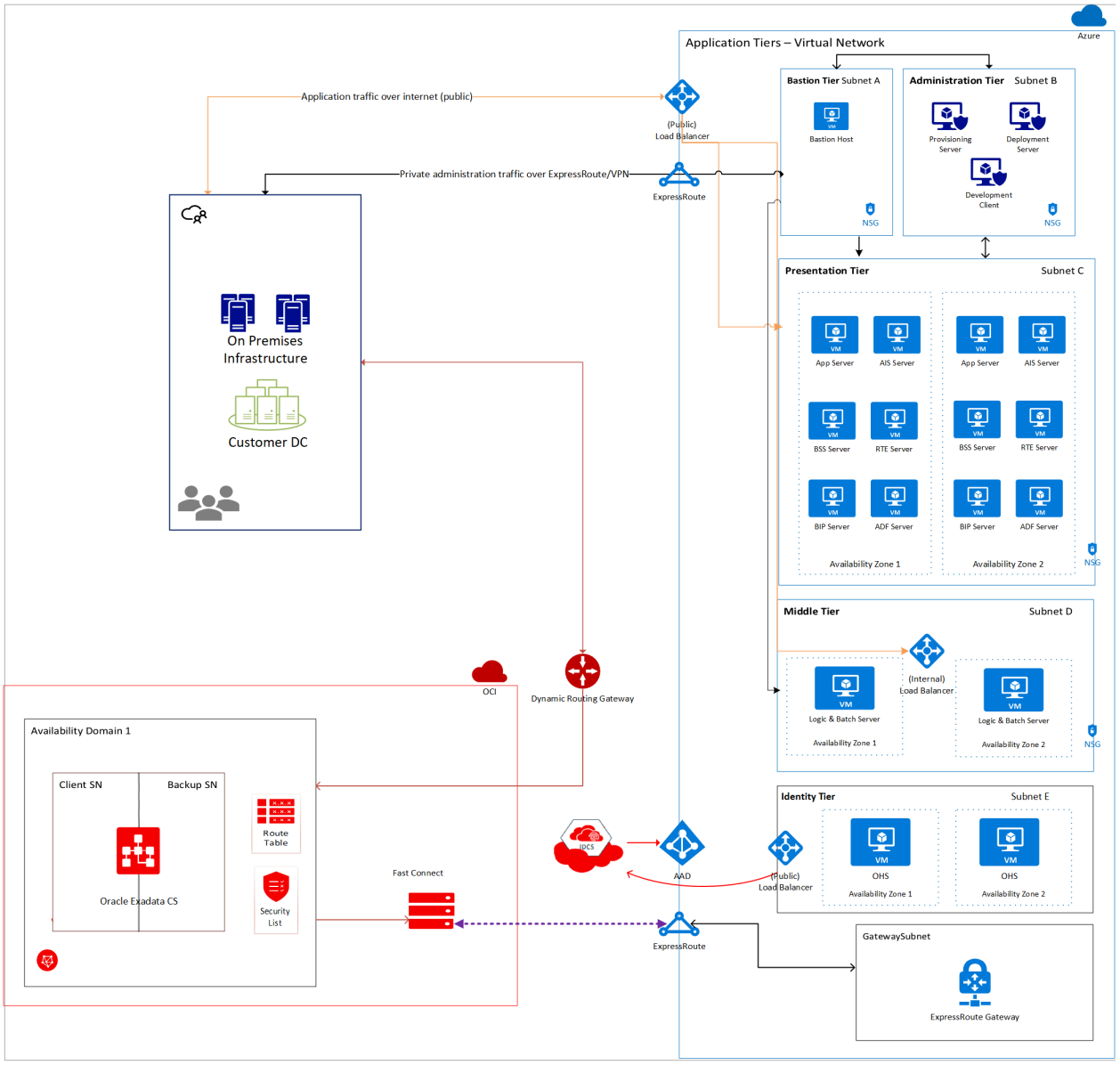


Figure 3: JD Edwards EnterpriseOne cross-cloud architecture

In this architecture, the virtual network in Azure is connected to the virtual cloud network in OCI using the cross-cloud interconnect. The application tier is set up in Azure, whereas the database is set up in OCI. It's recommended to deploy each component to its own subnet with network security groups to allow traffic only from specific subnets on specific ports.

The architecture can also be adapted for deployment entirely on Azure with highly available Oracle databases configured using Oracle Data Guard in two availability zones in a region. The following diagram (Figure 4) is an example of this architectural pattern:

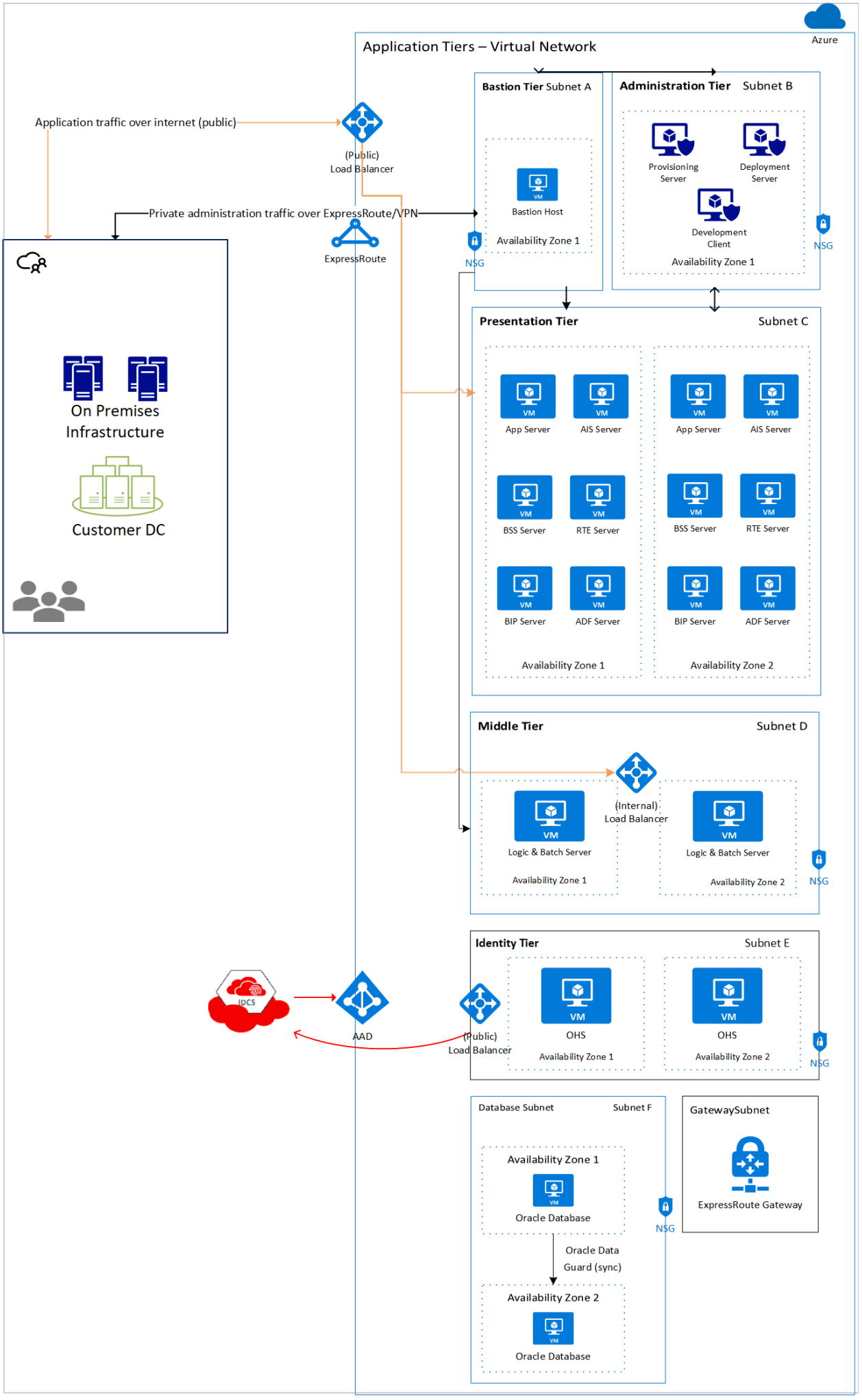

Azure
Application Tiers – Virtual Network
Application traffic over internet (public)
(Public)
Load Balancer
Private administration traffic over ExpressRoute/VPN
ExpressRoute
Bastion Tier Subnet A
Bastion Host
Availability Zone 1
NSG
Administration Tier Subnet B
Provisioning Server
Deployment Server
Development Client
Availability Zone 1
NSG
On Premises Infrastructure
Customer DC
Presentation Tier
Subnet C
App Server
AIS Server
BSS Server
RTE Server
BIP Server
ADF Server
Availability Zone 1
Availability Zone 2
Middle Tier
Subnet D
(Internal)
Load Balancer
Logic & Batch Server
Availability Zone 1
Availability Zone 2
NSG
Identity Tier
Subnet E
IDCS
AAD
(Public)
Load Balancer
OHS
Availability Zone 1
Availability Zone 2
NSG
Database Subnet
Subnet F
Availability Zone 1
Oracle Database
Oracle Data Guard (sync)
Availability Zone 2
Oracle Database
NSG
GatewaySubnet
ExpressRoute Gateway

Figure 4: JD Edwards EnterpriseOne Azure-only architecture

The following sections describe the different components at a high level.

Bastion tier

The bastion host is an optional component that you can use as a jump server to access the application and database instances. The bastion host VM can have a public IP address assigned to it, although the recommendation is to set up an ExpressRoute connection or site-to-site VPN with your on-premises network for secure access. Additionally, only SSH (port 22, Linux) or RDP (port 3389, Windows Server) should be opened for incoming traffic. For high availability, deploy a bastion host in two availability zones or in a single availability set.

You may also enable SSH agent forwarding on your VMs, which allows you to access other VMs in the virtual network by forwarding the credentials from your bastion host. Or, use SSH tunneling to access other instances.

Here's an example of agent forwarding:

```
ssh -A -t user@BASTION_SERVER_IP ssh -A root@TARGET_SERVER_IP`
```

This command connects to the bastion and then immediately runs `ssh` again, so you get a terminal on the target instance. You may need to specify a user other than root on the target instance if your cluster is configured differently. The `-A` argument forwards the agent connection so your private key on your local machine is used automatically. Note that agent forwarding is a chain, so the second `ssh` command also includes `-A` so that any subsequent SSH connections initiated from the target instance also use your local private key.

Administrative tier

As the name suggests, this tier is used for administrative tasks. You can carve out a separate subnet for the administrative tier. The services and servers in this tier are primarily used for installation and administration of the application. Hence, single instances of these servers are sufficient. Redundant instances aren't required for the high availability of your application.

The components of this tier are as follows:

- **Provisioning server** - This server is used for end-to-end deployment of the different components of the application. It communicates with the instances in the other tiers, including the instances in the database tier, over port 22. It hosts the Server Manager Console for JD Edwards EnterpriseOne.
- **Deployment server** - This server is primarily required for the installation of JD Edwards EnterpriseOne. During the installation process, this server acts as the central repository for required files and installation packages. The software is distributed or deployed to other servers and clients from this server.
- **Development client** - This server contains components that run in a web browser and native applications.

Presentation tier

This tier contains various components such as Application Interface Services (AIS), Application Development Framework (ADF), and Java Application Servers (JAS). The servers in this tier communicate with the servers in the middle tier. They're fronted by a load balancer that routes traffic to the necessary server based on the port number and URL that the traffic is received on. It's recommended that you deploy multiple instances of each server type for high availability.

The following are the components in this tier:

- **Application Interface Services (AIS)** - The AIS server provides the communication interface between JD Edwards EnterpriseOne mobile enterprise applications and JD Edwards EnterpriseOne.
- **Java Application Server (JAS)** - The JAS receives requests from the load balancer and passes it to the middle tier to execute complicated tasks. The JAS has the ability to execute simple business logic.
- **BI Publisher Server (BIP)** - This server presents reports based on the data collected by the JD Edwards EnterpriseOne application. You can design and control how the report presents the data based on different templates.
- **Business Services Server (BSS)** - The BSS enables information exchange and interoperability with other Oracle applications.
- **Real-Time Events Server (RTE)** - The RTE Server allows you to set up notifications to external systems about transactions occurring in the JDE EnterpriseOne system. It uses a subscriber model and allows third-party systems to subscribe to events. To load balance requests to both RTE servers, ensure that the servers are in a cluster.
- **Application Development Framework (ADF) Server** - The ADF server is used to run JD Edwards EnterpriseOne applications developed with Oracle ADF. This is deployed on an Oracle WebLogic server with ADF runtime.

Middle tier

The middle tier contains the logic server and batch server. In this case, both servers are installed on the same virtual machine. However, for production scenarios, it's recommended that you deploy logic server and batch server on separate servers. Multiple servers are deployed in the middle tier across two availability zones for higher availability. An Azure load balancer should be created and these servers should be placed in its backend pool to ensure that both servers are active and processing requests.

The servers in the middle tier receive requests from the servers in the presentation tier and the public load balancer only. Network security group rules must be set up to deny traffic from any address other than the presentation tier subnet and the load balancer. An NSG rule can also be set up to allow traffic on port 22 from the bastion host for management purposes. You may be able to use the public load balancer to load balance requests between the VMs in the middle tier.

The following two components are in the middle tier:

- **Logic server** - Contain the business logic or business functions.
- **Batch server** - Used for batch processing

Database tier

The Database tier contains the database instances for the application. The database can be either an Oracle DB, Oracle RAC, or Oracle Exadata Database system.

If the choice is to use Oracle DB, the database instance may be deployed on Azure via the Oracle DB images available on the Azure Marketplace. Alternatively, you may use the interconnect between Azure and OCI to deploy the Oracle DB in a PaaS model on OCI.

For Oracle RAC, you can use OCI in PaaS model. It is recommended that you use a two-node RAC system. While it is possible to deploy Oracle RAC on Azure CloudSimple in IaaS model, it is not a supported configuration by Oracle. Refer to Oracle programs eligible for Authorized Cloud Environments.

Finally, for Exadata systems, use the OCI interconnect and deploy the Exadata system in OCI. The preceding architecture diagram above shows an Exadata system deployed in OCI across two subnets.

For production scenarios, deploy multiple instances of the database across two availability zones (if deploying in Azure) or two availability domains (in OCI). Use Oracle Active Data Guard to synchronize the primary and standby databases.

The database tier only receives requests from the middle tier. It is recommended that you set up a network security group (security list if deploying the database in OCI) to only allow requests on port 1521 from the middle tier and port 22 from the bastion server for administrative reasons.

For databases deployed in OCI, a separate virtual cloud network must be set up with a dynamic routing gateway (DRG) that is connected to your FastConnect circuit.

Identity tier

The Microsoft-Oracle partnership enables you to set up a unified identity across Azure, OCI, and your Oracle application. For JD Edwards EnterpriseOne or PeopleSoft application suite, an instance of the Oracle HTTP Server (OHS) is needed to set up single sign-on between Microsoft Entra ID and Oracle IDCS.

OHS acts as a reverse proxy to the application tier, which means that all the requests to the end applications go through it. Oracle Access Manager WebGate is an OHS web server plugin that intercepts every request going to the end application. If a resource being accessed is protected (requires an authenticated session), the WebGate initiates OIDC authentication flow with Identity Cloud Service through the user's browser. For more information about the flows supported by the OpenID Connect WebGate, see the Oracle Access Manager documentation.

With this setup, a user already logged in to Microsoft Entra ID can navigate to the JD Edwards EnterpriseOne or PeopleSoft application without logging in again, through Oracle Identity Cloud Service. Customers that deploy this solution gain the benefits of single sign-on, including a single set of credentials, an improved sign-on experience, improved security, and reduced help-desk cost.

To learn more about setting up single sign-on for JD Edwards EnterpriseOne or PeopleSoft with Microsoft Entra ID, see the associated Oracle whitepaper.

PeopleSoft

Oracle's PeopleSoft application suite contains software for human resources and financial management. The application suite is multi-tiered and applications include human resource management systems (HRMS), customer relationship management (CRM), financials and supply chain management (FSCM), and enterprise performance management (EPM).

It's recommended that each tier of the software suite be deployed in its own subnet. An Oracle database or Microsoft SQL Server is required as the backend database for the

application. This section discusses details on deploying PeopleSoft with an Oracle database backend.

The following is a canonical architecture for deploying the PeopleSoft application suite in a cross-cloud architecture (Figure 5).

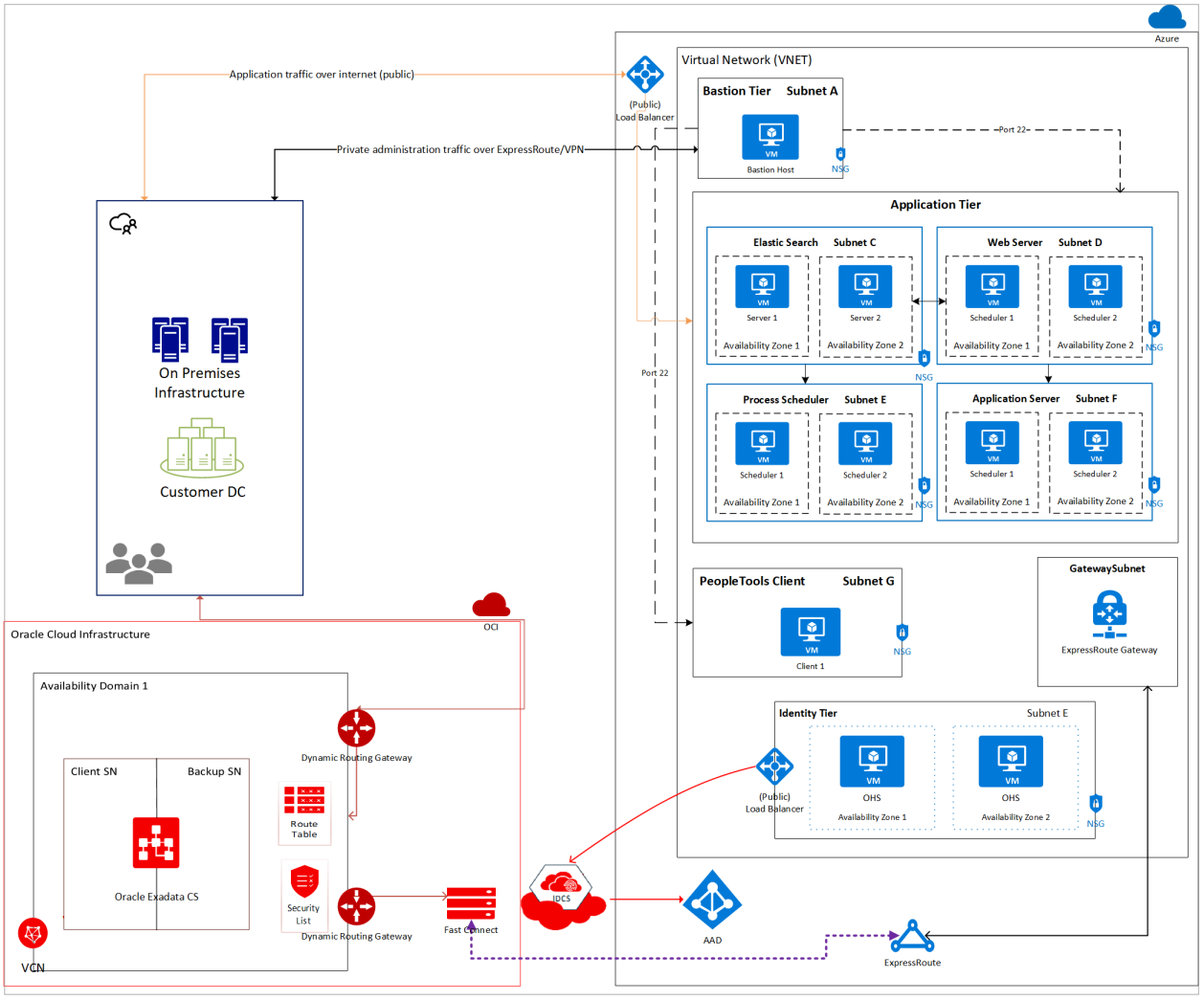


Figure 5: PeopleSoft cross-cloud architecture

In this sample architecture, the virtual network in Azure is connected to the virtual cloud network in OCI using the cross-cloud interconnect. The application tier is set up in Azure, whereas the database is set up in OCI. It's recommended to deploy each component to its own subnet with network security groups to allow traffic only from specific subnets on specific ports.

The architecture can also be adapted for deployment entirely on Azure with highly available Oracle databases configured using Oracle Data Guard in two availability zones in a region. The following diagram (Figure 6) is an example of this architectural pattern:

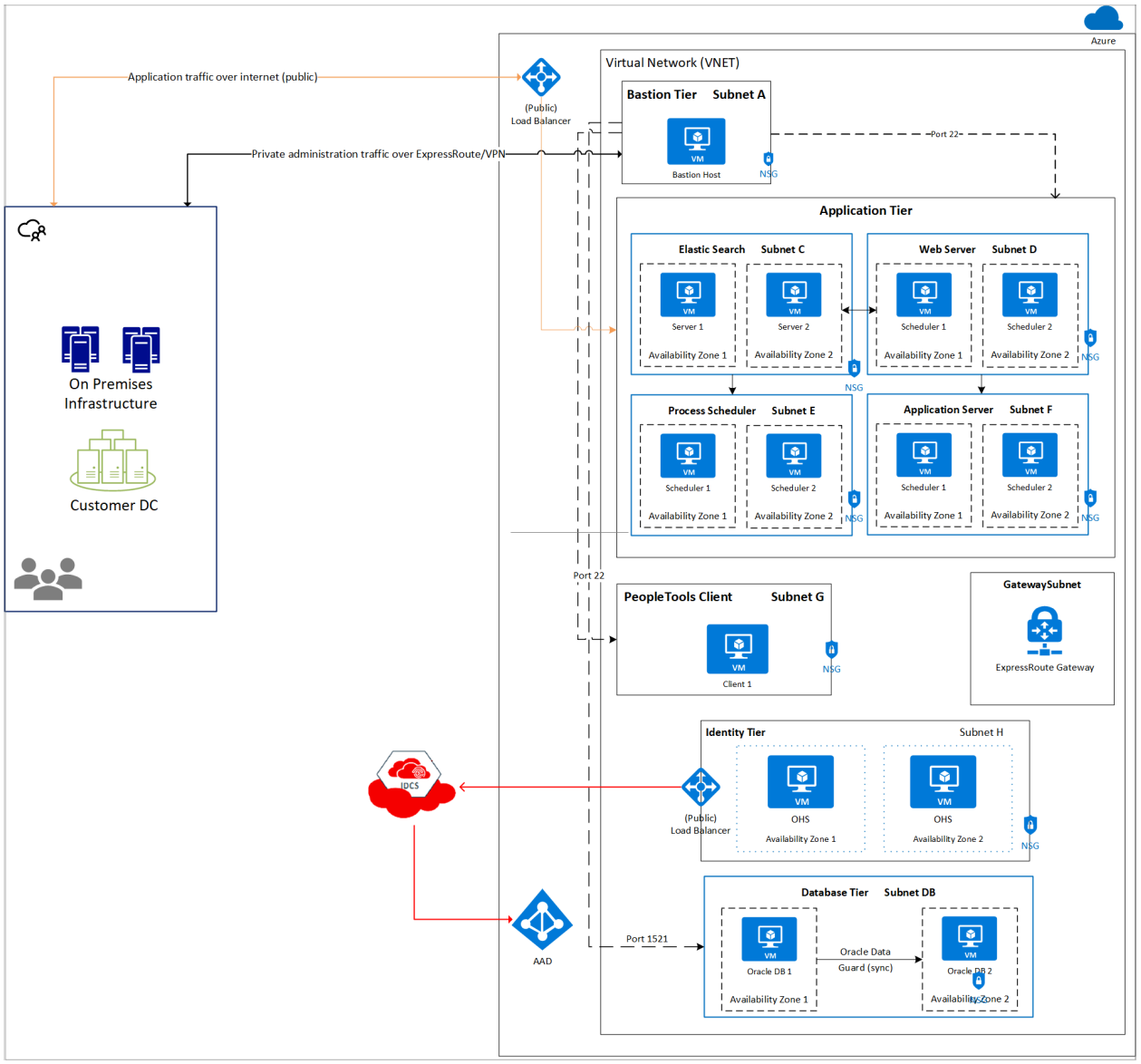


Figure 6: PeopleSoft Azure-only architecture

The following sections describe the different components at a high level.

Bastion tier

The bastion host is an optional component that you can use as a jump server to access the application and database instances. The bastion host VM can have a public IP address assigned to it, although the recommendation is to set up an ExpressRoute connection or site-to-site VPN with your on-premises network for secure access. Additionally, only SSH (port 22, Linux) or RDP (port 3389, Windows Server) should be opened for incoming traffic. For high availability, deploy a bastion host in two availability zones or in a single availability set.

You may also enable SSH agent forwarding on your VMs, which allows you to access other VMs in the virtual network by forwarding the credentials from your bastion host. Or, use SSH tunneling to access other instances.

Here's an example of agent forwarding:

```
ssh -A -t user@BASTION_SERVER_IP ssh -A root@TARGET_SERVER_IP`
```

This command connects to the bastion and then immediately runs `ssh` again, so you get a terminal on the target instance. You may need to specify a user other than root on the target instance if your cluster is configured differently. The `-A` argument forwards the agent connection so your private key on your local machine is used automatically. Note that agent forwarding is a chain, so the second `ssh` command also includes `-A` so that any subsequent SSH connections initiated from the target instance also use your local private key.

Application tier

The application tier contains instances of the PeopleSoft application servers, PeopleSoft web servers, elastic search, and PeopleSoft Process Scheduler. An Azure load balancer is set up to accept requests from users, which are routed to the appropriate server in the application tier.

For high availability, consider setting up redundant instances of each server in the application tier across different availability zones. The Azure load balancer can be set up with multiple back-end pools to direct each request to the right server.

PeopleTools Client

The PeopleTools Client is used to perform administration activities, such as development, migration, and upgrade. Because the PeopleTools Client isn't required for achieving high availability of your application, redundant servers of PeopleTools Client aren't needed.

Database tier

The Database tier contains the database instances for the application. The database can be either an Oracle DB, Oracle RAC, or Oracle Exadata Database system.

If the choice is to use Oracle DB, the database instance may be deployed on Azure via the Oracle DB images available on the Azure Marketplace. Alternatively, you may use the interconnect between Azure and OCI to deploy the Oracle DB in a PaaS model on OCI.

For Oracle RAC, you can use OCI in PaaS model. It is recommended that you use a two-node RAC system. While it is possible to deploy Oracle RAC on Azure CloudSimple in IaaS model, it is not a supported configuration by Oracle. Refer to Oracle programs eligible for Authorized Cloud Environments.

Finally, for Exadata systems, use the OCI interconnect and deploy the Exadata system in OCI. The preceding architecture diagram above shows an Exadata system deployed in OCI across two subnets.

For production scenarios, deploy multiple instances of the database across two availability zones (if deploying in Azure) or two availability domains (in OCI). Use Oracle Active Data Guard to synchronize the primary and standby databases.

The database tier only receives requests from the middle tier. It is recommended that you set up a network security group (security list if deploying the database in OCI) to only allow requests on port 1521 from the middle tier and port 22 from the bastion server for administrative reasons.

For databases deployed in OCI, a separate virtual cloud network must be set up with a dynamic routing gateway (DRG) that is connected to your FastConnect circuit.

Identity tier

The Microsoft-Oracle partnership enables you to set up a unified identity across Azure, OCI, and your Oracle application. For JD Edwards EnterpriseOne or PeopleSoft application suite, an instance of the Oracle HTTP Server (OHS) is needed to set up single sign-on between Microsoft Entra ID and Oracle IDCS.

OHS acts as a reverse proxy to the application tier, which means that all the requests to the end applications go through it. Oracle Access Manager WebGate is an OHS web server plugin that intercepts every request going to the end application. If a resource being accessed is protected (requires an authenticated session), the WebGate initiates OIDC authentication flow with Identity Cloud Service through the user's browser. For more information about the flows supported by the OpenID Connect WebGate, see the Oracle Access Manager documentation.

With this setup, a user already logged in to Microsoft Entra ID can navigate to the JD Edwards EnterpriseOne or PeopleSoft application without logging in again, through Oracle Identity Cloud Service. Customers that deploy this solution gain the benefits of single sign-on, including a single set of credentials, an improved sign-on experience, improved security, and reduced help-desk cost.

To learn more about setting up single sign-on for JD Edwards EnterpriseOne or PeopleSoft with Microsoft Entra ID, see the associated Oracle whitepaper.

Next steps

Use Terraform scripts to set up Oracle apps in Azure and establish cross-cloud connectivity with OCI.

For more information and whitepapers about OCI, see the Oracle Cloud documentation.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Architectures for Oracle database on Azure Virtual Machines

Article • 08/22/2024

Applies to: ✔ Linux VMs

This article includes information on deploying a highly available Oracle database on Azure. In addition, this guide dives into disaster recovery considerations. These architectures have been created based on customer deployments. This guide only applies to Oracle Database Enterprise Edition.

If you're interested in learning more about maximizing the performance of your Oracle database, see Design and implement an Oracle database in Azure.

Prerequisites

- An understanding of the different concepts of Azure such as availability zones
- Oracle Database Enterprise Edition 12c or later
- Awareness of the licensing implications when using the solutions in this article

High availability for Oracle databases

Achieving high availability in the cloud is an important part of every organization's planning and design. Azure offers availability zones and *availability sets* to be used in regions where availability zones are unavailable. For more information about how to design for the cloud, see Availability options for Azure Virtual Machines.

In addition to cloud-native tools and offerings, Oracle provides solutions for high availability that can be set up on Azure:

- Oracle Data Guard
- Data Guard with FSFO
- Sharding
- GoldenGate

This guide covers reference architectures for each of these solutions.

When you migrate or create applications for the cloud, we recommend using cloud-native patterns such as retry pattern and circuit breaker pattern. For other patterns that could help make your application more resilient, see Cloud Design Patterns guide.

Oracle RAC in the cloud

Oracle Real Application Cluster (RAC) is a solution by Oracle to help customers achieve high throughputs by having many instances accessing one database storage. This pattern is a shared-all architecture. While Oracle RAC can be used for high availability on-premises, Oracle RAC alone can't be used for high availability in the cloud. Oracle RAC only protects against instance level failures and not against rack-level or datacenter-level failures. For this reason, Oracle recommends using Oracle Data Guard with your database, whether single instance or RAC, for high availability.

Customers generally require a high SLA to run mission critical applications. Oracle doesn't currently certify or support Oracle RAC on Azure. However, Azure offers features such as availability zones and planned maintenance windows to help protect against instance-level failures. In addition to these offerings, you can use Oracle Data Guard, Oracle GoldenGate, and Oracle Sharding for high performance and resiliency. These technologies can help protect your databases from rack-level, datacenter-level, and geo-political failures.

When you run Oracle Databases on multiple availability zones with Oracle Data Guard or GoldenGate, you can get an uptime SLA of 99.99%. In Azure regions where availability zones aren't yet present, you can use availability sets and achieve an uptime SLA of 99.95%.

> **Note**
>
> You can have a uptime target that is much higher than the uptime SLA provided by Microsoft.

Disaster recovery for Oracle databases

When hosting your mission-critical applications in the cloud, it's important to design for high availability and disaster recovery.

For Oracle Database Enterprise Edition, Oracle Data Guard is a useful feature for disaster recovery. You can set up a standby database instance in a paired Azure region and set up Data Guard failover for disaster recovery. For zero data loss, we recommend that you deploy an Oracle Data Guard Far Sync instance in addition to Active Data Guard.

If your application permits the latency, consider setting up the Data Guard Far Sync instance in a different availability zone than your Oracle primary database. Test the configuration thoroughly. Use a *Maximum Availability* mode to set up synchronous

transport of your redo files to the Far Sync instance. These files are then transferred asynchronously to the standby database.

Your application might not allow for the performance loss when setting up Far Sync instance in another availability zone in *Maximum Availability* mode (synchronous). If not, you might set up a Far Sync instance in the same availability zone as your primary database. For added availability, consider setting up multiple Far Sync instances close to your primary database and at least one instance close to your standby database, if the role transitions.

When you use Oracle Standard Edition databases, there are ISV solutions that allow you to set up high availability and disaster recovery, such as DBVisit Standby.

Reference architectures

Oracle Data Guard

Oracle Data Guard ensures high availability, data protection, and disaster recovery for enterprise data. Data Guard maintains standby databases as transactionally consistent copies of the primary database. Depending on the distance between the primary and secondary databases and the application tolerance for latency, you can set up synchronous or asynchronous replication. If the primary database is unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the primary role, minimizing downtime.

When using Oracle Data Guard, you might also open your secondary database for read-only purposes. This configuration is called Active Data Guard. Oracle Database 12c introduced a feature called Data Guard Far Sync Instance. This instance allows you to set up a zero data loss configuration of your Oracle database without having to compromise on performance.

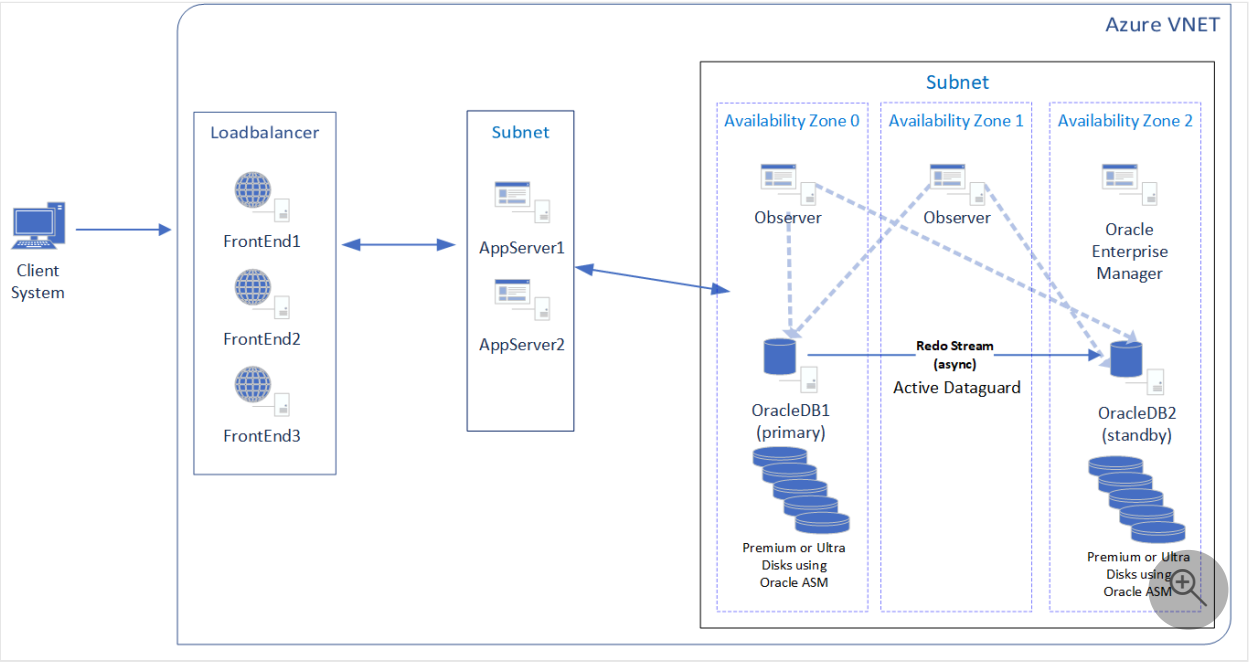
> **Note**
>
> Active Data Guard requires additional licensing. This license is also required to use the Far Sync feature. Contact with your Oracle representative to discuss the licensing implications.

Oracle Data Guard with Fast-Start Failover

Oracle Data Guard with Fast-Start Failover (FSFO) can provide more resiliency by setting up the broker on a separate machine. The Data Guard broker and the secondary database both run the observer and observe the primary database for downtime. This approach allows for redundancy in your Data Guard observer setup as well.

With Oracle Database version 12.2 and above, it's also possible to configure multiple observers with a single Oracle Data Guard broker configuration. This setup provides extra availability, in case one observer and the secondary database experience downtime. Data Guard Broker is lightweight and can be hosted on a relatively small virtual machine. For more information about Data Guard Broker and its advantages, see Oracle Data Guard Broker Concepts.

The following diagram is a recommended architecture for using Oracle Data Guard on Azure with availability zones. This architecture allows you to get a VM uptime SLA of 99.99%.



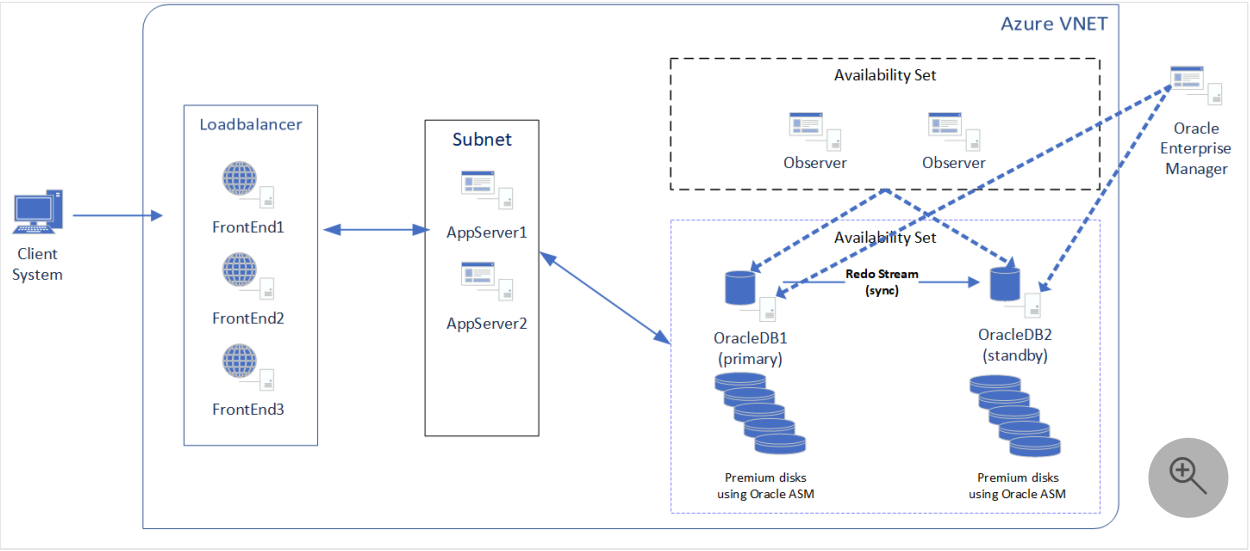
In the preceding diagram, the client system accesses a custom application with Oracle backend by using the web. The web frontend is configured in a load balancer. The web frontend makes a call to the appropriate application server to handle the work. The application server queries the primary Oracle database. The Oracle database has been configured using a hyperthreaded memory optimized virtual machine with constrained core vCPUs to save on licensing costs and maximize performance. Multiple premium or ultra disks (Managed Disks) are used for performance and high availability.

The Oracle databases are placed in multiple availability zones for high availability. Each zone is made up of one or more data centers equipped with independent power, cooling, and networking. To ensure resiliency, a minimum of three separate zones are set up in all enabled regions. The physical separation of availability zones within a region

protects the data from data center failures. Additionally, two FSFO observers are set up across two availability zones to initiate and fail over the database to the secondary when an outage occurs.

You might set up other observers or standby databases in a different availability zone, AZ 1, in this case, than the zone shown in the preceding architecture. Finally, Oracle Enterprise Manager (OEM) monitors Oracle databases for uptime and performance. OEM also allows you to generate various performance and usage reports.

In regions where availability zones aren't supported, you might use availability sets to deploy your Oracle Database in a highly available manner. Availability sets allow you to achieve a VM uptime of 99.95%. The following diagram is a reference architecture of this use:



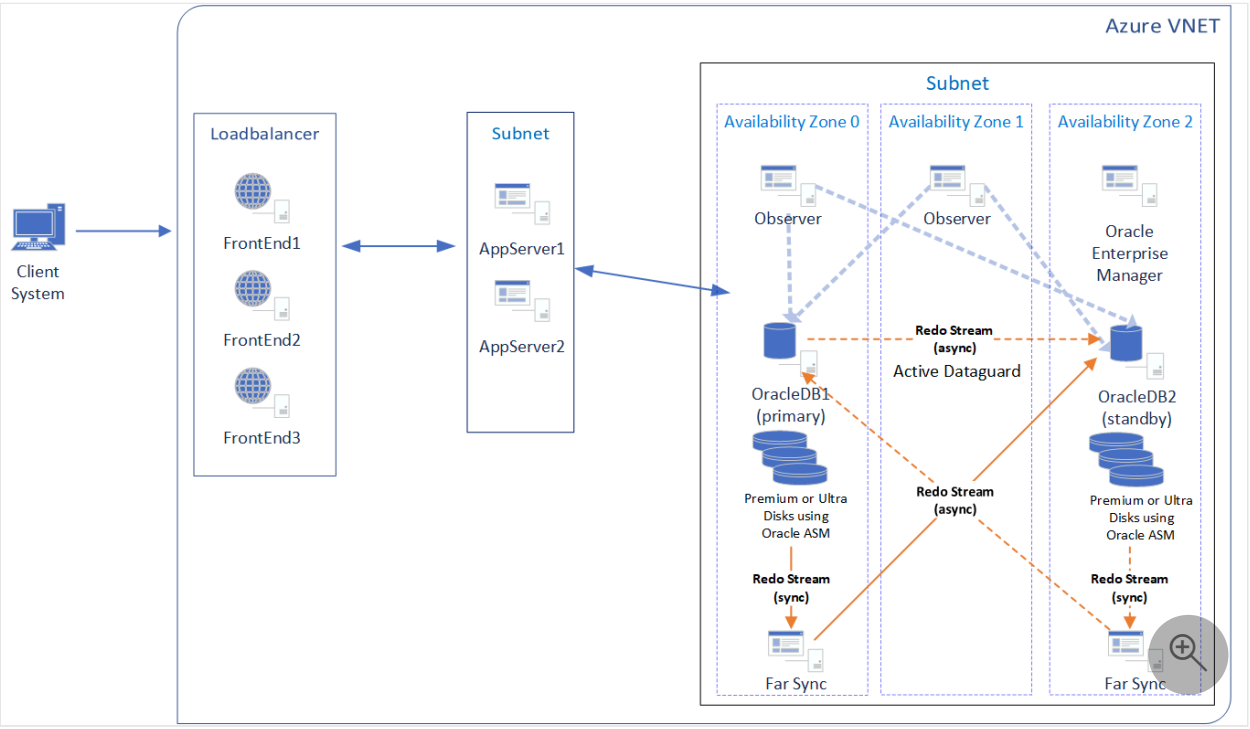
> **Note**
>
> - Because there is only one instance of OEM being deployed, you don't have to place the Oracle Enterprise Manager VM in an availability set.
> - Ultra disks aren't currently supported in an availability set configuration.

Oracle Data Guard Far Sync

Oracle Data Guard Far Sync provides zero data loss protection capability for Oracle Databases. This capability allows you to protect against data loss if your database machine fails. Oracle Data Guard Far Sync needs to be installed on a separate VM. Far Sync is a lightweight Oracle instance that only has a control file, password file, spfile, and standby logs. There are no data files or redo log files.

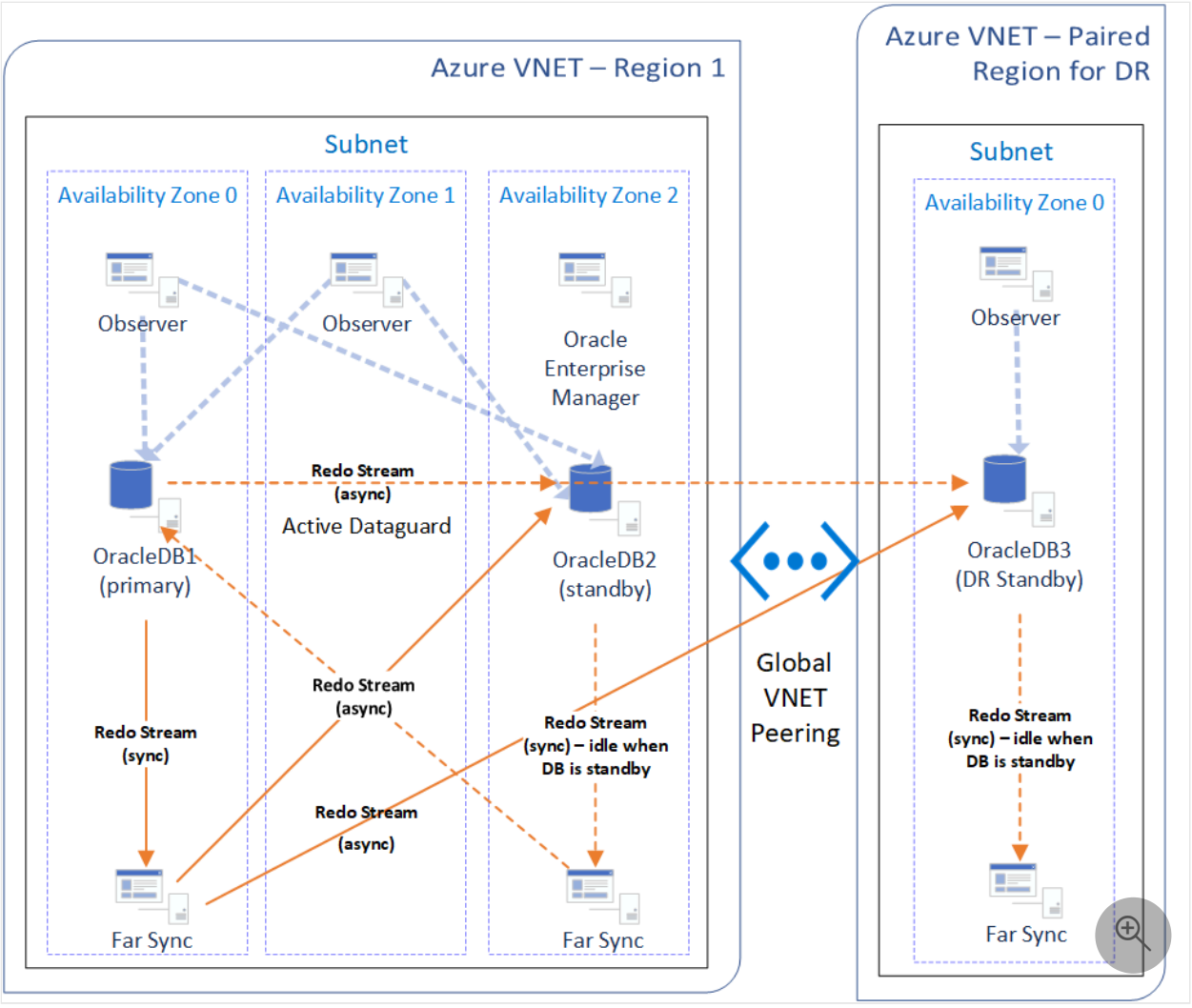
For zero data loss protection, there must be synchronous communication between your primary database and the Far Sync instance. The Far Sync instance receives redo from the primary in a synchronous manner and forwards it immediately to all the standby databases in an asynchronous manner. This setup also reduces the overhead on the primary database, because it only has to send the redo to the Far Sync instance rather than all the standby databases. If a Far Sync instance fails, Data Guard automatically uses asynchronous transport to the secondary database from the primary database to maintain near-zero data loss protection. For added resiliency, customers might deploy multiple Far Sync instances per each database instance, including primary and secondaries.

The following diagram is a high availability architecture using Oracle Data Guard Far Sync:



In the preceding architecture, there's a Far Sync instance deployed in a different availability zone as the database instance to ensure zero data loss and automatic failover in case of availability zone failure. In cases where the application is latency sensitive, consider deploying your database and Far Sync instance or instances in the same availability zone in a proximity placement group.

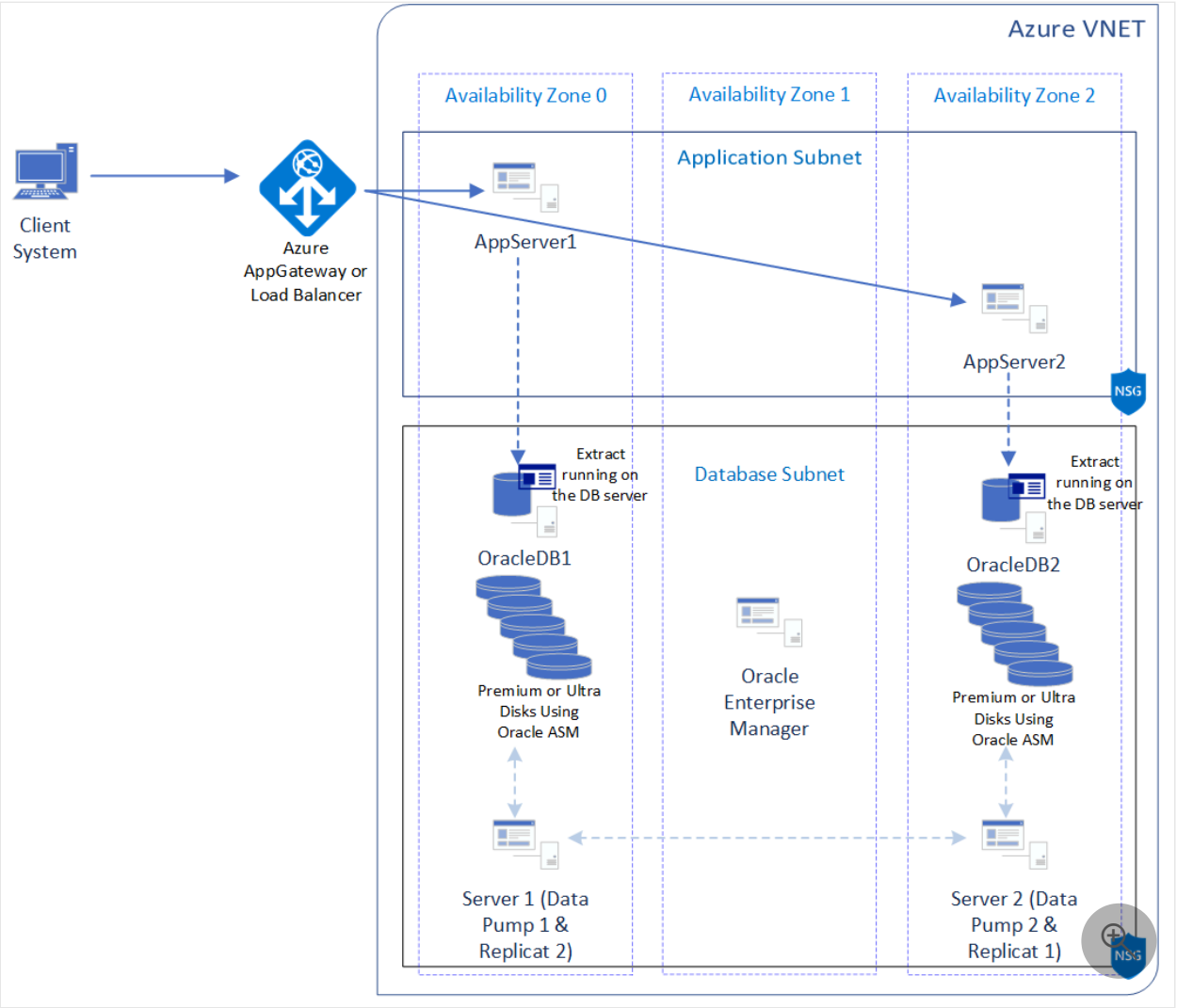
The following diagram is an architecture that uses Oracle Data Guard FSFO and Far Sync to achieve high availability and disaster recovery:



Oracle GoldenGate

GoldenGate enables the exchange and manipulation of data at the transaction level among multiple, heterogeneous platforms across the enterprise. It moves committed transactions with transaction integrity and minimal overhead on your existing infrastructure. Its modular architecture gives you the flexibility to extract and replicate selected data records, transactional changes, and changes to data definition language (DDL) across various topologies.

Oracle GoldenGate allows you to configure your database for high availability by providing bidirectional replication. This approach allows you to set up a *multi-master* or *active-active configuration*. The following diagram is a recommended architecture for Oracle GoldenGate active-active setup on Azure. In the following architecture, the Oracle database has been configured using a hyperthreaded memory optimized virtual machine with constrained core vCPUs to save on licensing costs and maximize performance. The architecture uses multiple premium or ultra disks (managed disks) for performance and availability.



> **Note**
>
> A similar architecture can be set up using availability sets in regions where availability zones aren't currently available.

Oracle GoldenGate has processes such as *Extract*, *Pump*, and *Replicat* that help you asynchronously replicate your data from one Oracle database server to another. These processes allow you to set up a bidirectional replication to ensure high availability of your database if there's availability zone-level downtime.

In the preceding diagram, the Extract process runs on the same server as your Oracle database. The Data Pump and Replicat processes run on a separate server in the same availability zone. The Replicat process is used to receive data from the database in the other availability zone and commit the data to the Oracle database in its availability zone. Similarly, the Data Pump process sends data that the Extract process extracts to the Replicat process in the other availability zone.

While the preceding architecture diagram shows the Data Pump and Replicat processes configured on a separate server, you might set up all the Oracle GoldenGate processes

on the same server, based on the capacity and usage of your server. Always consult your AWR report and the metrics in Azure to understand the usage pattern of your server.

When setting up Oracle GoldenGate bidirectional replication in different availability zones or different regions, it's important to ensure that the latency between the different components is acceptable for your application. The latency between availability zones and regions can vary. Latency depends on multiple factors. We recommend that you set up performance tests between your application tier and your database tier in different availability zones or regions. The tests can confirm that the configuration meets your application performance requirements.

The application tier can be set up in its own subnet and the database tier can be separated into its own subnet. When possible, consider using Azure Application Gateway to load-balance traffic between your application servers. Application Gateway is a robust web traffic load balancer. It provides cookie-based session affinity that keeps a user session on the same server, minimizing the conflicts on the database. Alternatives to Application Gateway are Azure Load Balancer and Azure Traffic Manager.

Oracle Sharding

Sharding is a data tier pattern that was introduced in Oracle 12.2. It allows you to horizontally partition and scale your data across independent databases. It's a share-nothing architecture where each database is hosted on a dedicated virtual machine. This pattern enables high read and write throughput in addition to resiliency and increased availability.

This pattern eliminates single points of failure, provides fault isolation, and enables rolling upgrades without downtime. The downtime of one shard or a data center-level failure doesn't affect the performance or availability of the other shards in other data centers.

Sharding is suitable for high throughput OLTP applications that can't afford any downtime. All rows with the same sharding key are always guaranteed to be on the same shard. This fact increases performance, providing high consistency. Applications that use sharding must have a well-defined data model and data distribution strategy, such as consistent hash, range, list, or composite. The strategy primarily accesses data using a sharding key, for example, *customerId* or *accountNum*. Sharding also allows you to store particular sets of data closer to the end customers, thus helping meet your performance and compliance requirements.

We recommend that you replicate your shards for high availability and disaster recovery. This setup can be done using Oracle technologies such as Oracle Data Guard or Oracle

GoldenGate. A unit of replication can be a shard, a part of a shard, or a group of shards. An outage or slowdown of one or more shards doesn't affect the availability of a sharded database.

For high availability, the standby shards can be placed in the same availability zone where the primary shards are placed. For disaster recovery, the standby shards can be located in another region. You might also deploy shards in multiple regions to serve traffic in those regions. To learn more about configuring high availability and replication of your sharded database, see Shard-Level High Availability.

Oracle Sharding primarily consists of the following components. For more information, see Oracle Sharding Overview:

- **Shard catalog**. Special-purpose Oracle database that is a persistent store for all Shard database configuration data. All configuration changes such as adding or removing shards, mapping of the data, and DDLs in a shard database are initiated on the shard catalog. The shard catalog also contains the master copy of all duplicated tables in an SDB.

  The shard catalog uses materialized views to automatically replicate changes to duplicated tables in all shards. The shard catalog database also acts as a query coordinator used to process multi-shard queries and queries that don't specify a sharding key.

  We recommend using Oracle Data Guard with availability zones or availability sets for shard catalog high availability as a best practice. The availability of the shard catalog has no effect on the availability of the sharded database. A downtime in the shard catalog only affects maintenance operations and multishard queries during the brief period that the Data Guard failover completes. The SDB continues to route and run online transactions. A catalog outage doesn't affect them.

- **Shard directors**. Lightweight services that need to be deployed in each region/availability zone that your shards reside in. Shard Directors are Global Service Managers deployed in the context of Oracle Sharding. For high availability, we recommend that you deploy at least one shard director in each availability zone that your shards exist in.

  When connecting to the database initially, the shard director sets up the routing information and caches the information for subsequent requests, which bypass the shard director. Once the session is established with a shard, all SQL queries and DMLs are supported and executed in the scope of the given shard. This routing is fast and is used for all OLTP workloads that perform intra-shard transactions. We recommend that you to use direct routing for all OLTP workloads that require the

highest performance and availability. The routing cache automatically refreshes when a shard becomes unavailable or changes occur to the sharding topology.

For high-performance, data-dependent routing, Oracle recommends using a connection pool when accessing data in the sharded database. Oracle connection pools, language-specific libraries, and drivers support Oracle Sharding. For more information, see Oracle Sharding Overview.

- **Global service**. Global service is similar to the regular database service. In addition to all the properties of a database service, a global service has properties for sharded databases. These properties include region affinity between clients and shard and replication lag tolerance. Only one global service needs to be created to read/write data to and from a sharded database. When using Active Data Guard and setting up read-only replicas of the shards, you can create another global service for read-only workloads. The client can use these global services to connect to the database.

- **Shard databases**. Shard databases are your Oracle databases. Each database is replicated using Oracle Data Guard in a Broker configuration with FSFO enabled. You don't need to set up Data Guard failover and replication on each shard. This aspect is automatically configured and deployed when the shared database is created. If a particular shard fails, Oracle Sharing fails over database connections from the primary to the standby.

You can deploy and manage Oracle sharded databases with two interfaces: Oracle Enterprise Manager Cloud Control GUI and the `GDSCTL` command-line utility. You can even monitor the different shards for availability and performance using Cloud control. The `GDSCTL DEPLOY` command automatically creates the shards and their respective listeners. In addition, this command automatically deploys the replication configuration used for shard-level high availability specified by the administrator.

There are different ways to shard a database:

- System-managed sharding: Automatically distributes across shards using partitioning
- User-defined sharding: Allows you to specify the mapping of the data to the shards, which works well when there are regulatory or data-localization requirements
- Composite sharding: A combination of system-managed and user-defined sharding for different *shardspaces*
- Table subpartitions: Similar to a regular partitioned table

For more information, see Sharding Methods.

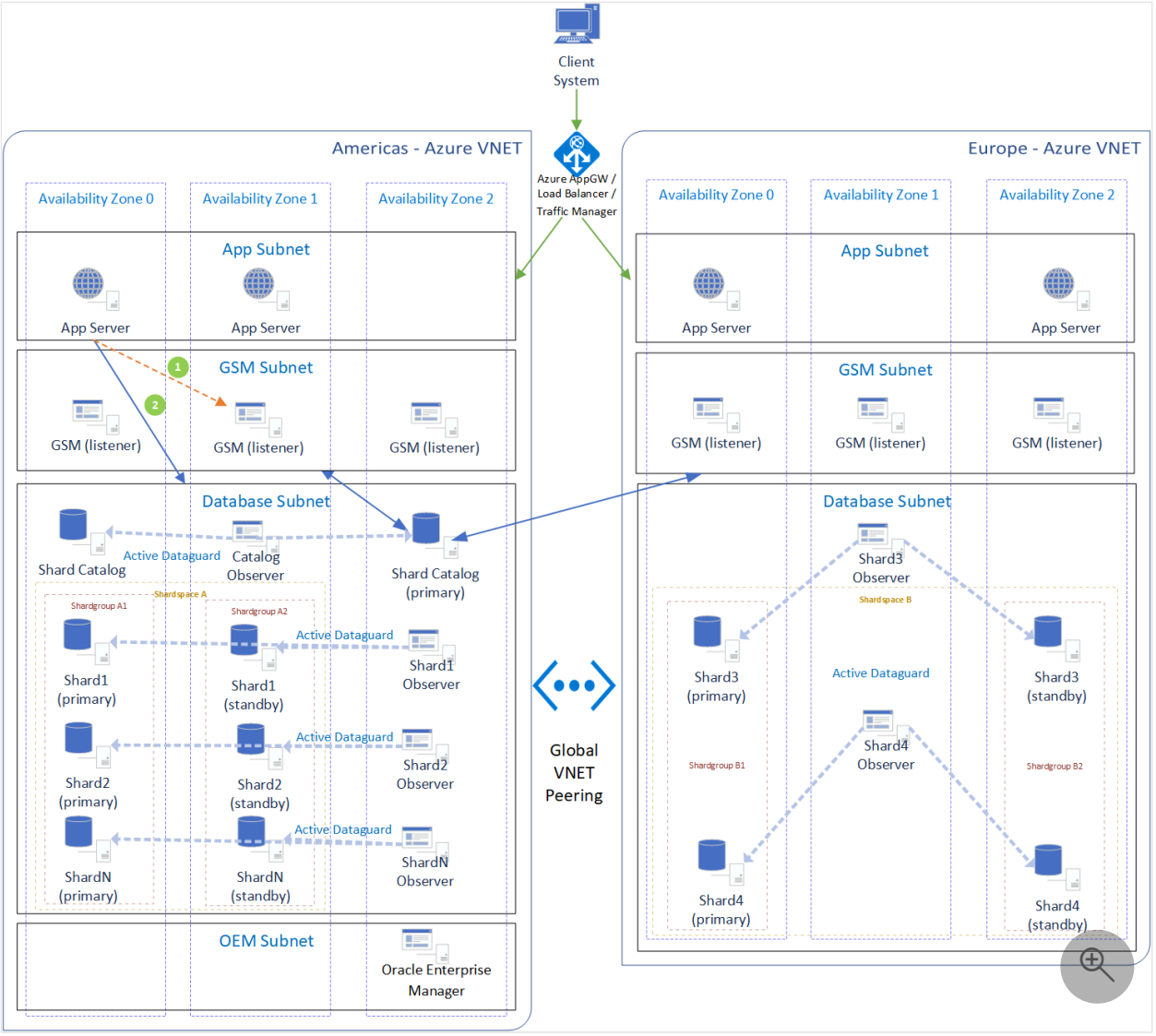
A sharded database looks like a single database to applications and developers. When you migrate to a sharded database, plan carefully to understand which tables are duplicated versus sharded.

Duplicated tables are stored on all shards, whereas sharded tables are distributed across different shards. We recommend that you duplicate small and dimensional tables and distribute/shard the fact tables. Data can be loaded into your sharded database using either the shard catalog as the central coordinator or by running Data Pump on each shard. For more information, see Migrating Data to a Sharded Database.

Oracle Sharding with Data Guard

Oracle Data Guard can be used for sharding with system-managed, user-defined, and composite sharding methods.

The following diagram is a reference architecture for Oracle Sharding with Oracle Data Guard used for high availability of each shard. The architecture diagram shows a *composite sharding method*. The architecture diagram likely differs for applications with different requirements for data locality, load balancing, high availability, and disaster recovery. Applications might use different method for sharding. Oracle Sharding allows you to meet these requirements and scale horizontally and efficiently by providing these options. A similar architecture can even be deployed using Oracle GoldenGate.



System-managed sharding is the easiest to configure and manage. User-defined sharding or composite sharding is well suited for scenarios where your data and application are geo-distributed or in scenarios where you need to have control over the replication of each shard.

In the preceding architecture, composite sharding is used to geodistribute the data and horizontally scale out your application tiers. Composite sharding is a combination of system-managed and user-defined sharding and thus provides the benefit of both methods. In the preceding scenario, data is first sharded across multiple shardspaces separated by region. Then, the data is further partitioned by using consistent hash across multiple shards in the shardspace.

Each shardspace contains multiple shardgroups. Each shardgroup has multiple shards and is a unit of replication. Each shardgroup contains all the data in the shardspace. Shardgroups A1 and B1 are primary shardgroups, while shardgroups A2 and B2 are standbys. You might choose to have individual shards be the unit of replication, rather than a shardgroup.

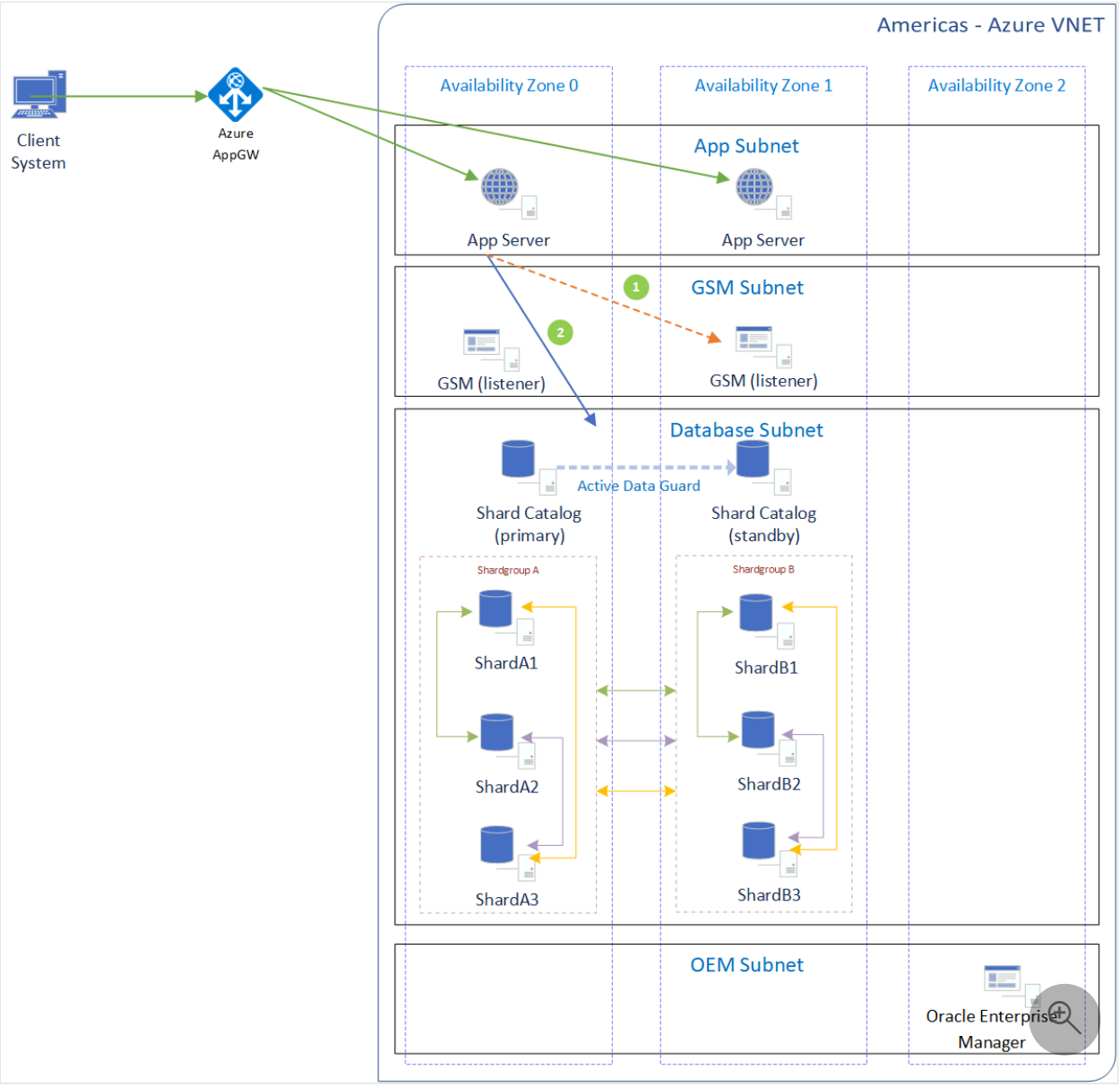
In the preceding architecture, a Global Service Manager (GSM)/shard director is deployed in every availability zone for high availability. We recommend that you deploy at least one GSM/shard director per data center/region. Additionally, an instance of the application server is deployed in every availability zone that contains a shardgroup. This setup allows the application to keep the latency between the application server and the database/shardgroup low. If a database fails, the application server in the same zone as the standby database can handle requests once the database role transition happens. Azure Application Gateway and the shard director keep track of the request and response latency and route requests accordingly.

From an application standpoint, the client system makes a request to Azure Application Gateway or other load-balancing technologies in Azure, which redirects the request to the region closest to the client. Azure Application Gateway also supports sticky sessions, so any requests coming from the same client are routed to the same application server. The application server uses connection pooling in data access drivers. This feature is available in drivers such as JDBC, ODP.NET, and OCI. The drivers can recognize sharding keys specified as part of the request. Oracle Universal Connection Pool (UCP) for JDBC clients can enable non-Oracle application clients such as Apache Tomcat and IIS to work with Oracle Sharding. For more information, see Overview of UCP Shared Pool for Database Sharding.

During the initial request, the application server connects to the shard director in its region to get routing information for the shard that the request needs to be routed to. Based on the sharding key passed, the director routes the application server to the respective shard. The application server caches this information by building a map, and for subsequent requests, bypasses the shard director and routes requests straight to the shard.

Oracle Sharding with GoldenGate

The following diagram is a reference architecture for Oracle Sharding with Oracle GoldenGate for in-region high availability of each shard. As opposed to the preceding architecture, this architecture only portrays high availability within a single Azure region, with multiple availability zones. You can deploy a multi-region high availability sharded database, similar to the preceding example, by using Oracle GoldenGate.



The preceding reference architecture uses the *system-managed* sharding method to shard the data. Since Oracle GoldenGate replication is done at a chunk level, half the data replicated to one shard can be replicated to another shard. The other half can be replicated to a different shard.

The way the data gets replicated depends on the replication factor. With a replication factor of two, you have two copies of each chunk of data across your three shards in the shardgroup. Similarly, with a replication factor of three and three shards in your shardgroup, all the data in each shard is replicated to every other shard in the shardgroup. Each shard in the shardgroup can have a different replication factor. This setup helps you define your high availability and disaster recovery design efficiently within a shardgroup and across multiple shardgroups.

In the preceding architecture, shardgroup A and shardgroup B both contain the same data but reside in different availability zones. If both shardgroup A and shardgroup B have the same replication factor of three, each row/chunk of your sharded table is replicated six times across the two shardgroups. If shardgroup A has a replication factor

of three and shardgroup B has a replication factor of two, each row/chunk is replicated five times across the two shardgroups.

This setup prevents data loss if an instance-level or availability zone-level failure occurs. The application layer is able to read from and write to each shard. To minimize conflicts, Oracle Sharding designates a *master chunk* for each range of hash values. This feature ensures that write requests for a particular chunk are directed to the corresponding chunk. In addition, Oracle GoldenGate provides automatic conflict detection and resolution to handle any conflicts that might arise. For more information and limitations of implementing GoldenGate with Oracle Sharding, see Using Oracle GoldenGate with a Sharded Database.

In the preceding architecture, a GSM/shard director is deployed in every availability zone for high availability. We recommend that you deploy at least one GSM/shard director per data center or region. An instance of the application server is deployed in every availability zone that contains a shardgroup. This setup allows the application to keep the latency between the application server and the database/shardgroup low. If a database fails, the application server in the same zone as the standby database can handle requests once the database role transitions. Azure Application Gateway and the shard director keep track of the request and response latency and route requests accordingly.

From an application standpoint, the client system makes a request to Azure Application Gateway or other load-balancing technologies in Azure, which redirects the request to the region closest to the client. Azure Application Gateway also supports sticky sessions, so any requests coming from the same client are routed to the same application server. The application server uses connection pooling in data access drivers. This feature is available in drivers such as JDBC, ODP.NET, and OCI. The drivers can recognize sharding keys specified as part of the request. Oracle Universal Connection Pool (UCP) for JDBC clients can enable non-Oracle application clients such as Apache Tomcat and IIS to work with Oracle Sharding.

During the initial request, the application server connects to the shard director in its region to get routing information for the shard that the request needs to be routed to. Based on the sharding key passed, the director routes the application server to the respective shard. The application server caches this information by building a map, and for subsequent requests, bypasses the shard director and routes requests straight to the shard.

Patching and maintenance

When you deploy your Oracle workloads to Azure, Microsoft takes care of all host operating system level patching. Microsoft communicates any planned operating system level maintenance to customers in advance. Two servers from two different availability zones are never patched simultaneously. For more information on VM maintenance and patching, see Availability options for Azure Virtual Machines.

Patching your virtual machine operating system can be automated using Azure Automation Update Management. Patching and maintaining your Oracle database can be automated and scheduled using Azure Pipelines or Azure Automation Update Management to minimize downtime. For more information about continuous delivery and blue/green deployments, see Progressive exposure techniques.

Architecture and design considerations

- Consider using hyperthreaded memory optimized virtual machine with constrained core vCPUs for your Oracle Database VM to save on licensing costs and maximize performance. Use multiple premium or ultra disks (managed disks) for performance and availability.
- When you use managed disks, the disk/device name might change on restart. We recommend that you use the device UUID instead of the name to ensure your mounts persist in sprite of restarting. For more information, see Add the new file system to /etc/fstab.
- Use availability zones to achieve high availability in-region.
- Consider using ultra disks when available or premium disks for your Oracle database.
- Consider setting up a standby Oracle database in another Azure region using Oracle Data Guard.
- Consider using proximity placement groups to reduce the latency between your application and database tier.
- Azure VMs maximum network bandwidth is (typically) higher than the maximum disk throughput on the same SKU. You can achieve higher throughput on the same VM SKU or use a smaller VM SKU by using high-performance, low-latency networked storage such as Azure NetApp Files. for the database.
- Set up Oracle Enterprise Manager for management, monitoring, and logging.
- Consider using Oracle Automatic Storage Management for streamlined storage management for your database.
- Use Azure Pipelines to manage patching and updates to your database without any downtime.
- Tweak your application code to add cloud-native patterns that might help your application be more resilient. Consider patterns such as retry pattern, circuit breaker pattern, and others defined in the Cloud Design Patterns guide.

Next steps

Review the following Oracle reference articles that apply to your scenario.

- Introduction to Oracle Data Guard
- Oracle Data Guard Broker Concepts
- Configuring Oracle GoldenGate for Active-Active High Availability
- Oracle Sharding Overview

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Architectures for Oracle Database Enterprise Edition on Azure

Article • 08/22/2024

Applies to: ✔ Linux VMs

Azure is home for all Oracle workloads, including workloads that need to continue to run optimally in Azure with Oracle. If you have the Oracle Diagnostic Pack or the Automatic Workload Repository (AWR), you can gather data about your workloads. Use this data to assess the Oracle workload, size the resource needs, and migrate the workload to Azure. The various metrics provided by Oracle in these reports can provide an understanding of application performance and platform usage.

This article helps you to prepare an Oracle workload to run in Azure and explore the best architecture solutions to provide optimal cloud performance. The data provided by Oracle in the Statspack and even more so in its descendent, the AWR, assists you in developing clear expectations. These expectations include the limits of physical tuning through architecture, the advantages of logical tuning of database code, and the overall database design.

Differences between the two environments

When you're migrating on-premises applications to Azure, keep in mind a few important differences between the two environments.

One important difference is that in an Azure implementation, resources such as VMs, disks, and virtual networks are shared among other clients. In addition, resources can be throttled based on the requirements. Instead of focusing on avoiding failing, Azure focuses more on surviving the failure. The first approach tries to increase *mean time between failures (MTBF)* and the second tries to decrease *mean time to recovery (MTTR)*.

The following table lists some of the differences between an on-premises implementation and an Azure implementation of an Oracle database.

Expand table

	On-premises implementation	Azure implementation
Networking	LAN/WAN	Software-defined networking (SDN)

	On-premises implementation	Azure implementation
Security group	IP/port restriction tools	Network security group (NSG)
Resilience	MTBF	MTTR
Planned maintenance	Patching/upgrades	Availability sets with patching/upgrades managed by Azure
Resource	Dedicated	Shared with other clients
Regions	Datacenters	Region pairs
Storage	SAN/physical disks	Azure-managed storage
Scale	Vertical scale	Horizontal scale

Requirements

Consider the following requirements before you start your migration:

- Determine the real CPU usage. Oracle licenses by core, which means that sizing your vCPU needs can be essential to help you reduce costs.
- Determine the database size, backup storage, and growth rate.
- Determine the I/O requirements, which you can estimate based on Oracle Statspack and the AWR reports. You can also estimate the requirements from storage monitoring tools available from the operating system.

Configuration options

It's a good idea to generate an AWR report and obtain some metrics from it to help you make decisions about configuration. Then, there are four potential areas that you can tune to improve performance in an Azure environment:

- Virtual machine size
- Network throughput
- Disk types and configurations
- Disk cache settings

Generate an AWR report

If you have an existing an Oracle Enterprise Edition database and are planning to migrate to Azure, you have several options. If you have the Diagnostics Pack for your

Oracle instances, you can run the Oracle AWR report to get the metrics, such as IOPS, Mbps, and GiBs. For those databases without the Diagnostics Pack license, or for an Oracle Standard Edition database, you can collect the same important metrics with a Statspack report after you collect manual snapshots. The main differences between these two reporting methods are that AWR is automatically collected, and that it provides more information about the database than does Statspack.

Consider running your AWR report during both regular and peak workloads, so you can compare. To collect the more accurate workload, consider an extended window report of one week, as opposed to one day. AWR provides averages as part of its calculations in the report. By default, the AWR repository retains eight days of data and takes snapshots at hourly intervals.

For a datacenter migration, you should gather reports for sizing on the production systems. Estimate remaining database copies used for user testing, test, and development by percentages. For example, estimate 50 percent of production sizing.

To run an AWR report from the command line, use the following command:

```bash
sqlplus / as sysdba
@$ORACLE_HOME/rdbms/admin/awrrpt.sql;
```

Key metrics

The report prompts you for the following information:

- Report type: HTML or TEXT. The HTML type provides more information.
- The number of days of snapshots to display. For example, for one-hour intervals, a one-week report produces 168 snapshot IDs.
- The beginning `SnapshotID` for the report window.
- The ending `SnapshotID` for the report window.
- The name of the report that the AWR script creates.

If you're running the AWR report on a Real Application Cluster (RAC), the command-line report is the *awrgrpt.sql* file, instead of *awrrpt.sql*. The `g` report creates a report for all nodes in the RAC database in a single report. This report eliminates the need to run one report on each RAC node.

You can obtain the following metrics from the AWR report:

- Database name, instance name, and host name

- Database version for supportability by Oracle
- CPU/Cores
- SGA/PGA, and advisors to let you know if undersized
- Total memory in GB
- CPU percentage busy
- DB CPUs
- IOPs (read/write)
- MBPs (read/write)
- Network throughput
- Network latency rate (low/high)
- Top wait events
- Parameter settings for database
- Whether the database is RAC, Exadata, or using advanced features or configurations

Virtual machine size

Here are some steps you can take to configure virtual machine size for optimal performance.

Estimate VM size based on CPU, memory, and I/O usage from the AWR report

Look at the top five timed foreground events that indicate where the system bottlenecks are. For example, in the following diagram, the log file sync is at the top. It indicates the number of waits that is required before the log writer writes the log buffer to the redo log file. These results indicate that better performing storage or disks are required. In addition, the diagram also shows the number of CPU cores and the amount of memory.

Top 5 Timed Foreground Events

Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
log file sync	16,997,881	163,293	10	72.04	Commit
buffer busy waits	4,014,884	18,767	5	8.28	Concurrency
DB CPU		15,413		6.80	
enq: TX - index contention	2,001,818	11,531	6	5.09	Concurrency
library cache: mutex X	2,024,372	6,986	3	3.08	Concurrency

Host CPU (CPUs: 16 Cores: 16 Sockets: 2)

Load Average Begin	Load Average End	%User	%System	%WIO	%Idle
4.65	11.03	25.8	5.1	6.4	68.4

Instance CPU

%Total CPU	%Busy CPU	%DB time waiting for CPU (Resource Manager)
31.7	100.3	0.0

Memory Statistics

	Begin	End
Host Mem (MB):	112,804.7	112,804.7
SGA use (MB):	61,440.0	61,440.0
PGA use (MB):	1,051.6	1,559.6
% Host Mem used for SGA+PGA:	55.40	55.85

The following diagram shows the total I/O of read and write. There were 59 GB read and 247.3 GB written during the time of the report.

IOStat by Function/Filetype summary

- 'Data' columns suffixed with M,G,T,P are in multiples of 1024 other columns suffixed with K,M,G,T,P are in multiples of 1000
- Ordered by (Data Read + Write) desc for each function

Function/File Name	Reads: Data	Reqs per sec	Data per sec	Writes: Data	Reqs per sec	Data per sec	Waits: Count	Avg Tm(ms)
Others	42G	25.51	13.1724	87.5G	63.98	27.4202	171.1K	6.55
Others (Flashback Log)	32M	0.03	.009795	47.5G	46.92	14.8805	153.4K	7.10
Others (Log File)	41.7G	20.11	13.0716	0M	0.00	0M	127	13.66
Others (Archive Log)	0M	0.00	0M	39.8G	12.47	12.4677	0	
Others (Control File)	297M	5.37	.090918	234M	4.58	.071632	17.6K	1.70
Others (Data File)	0M	0.00	0M	1M	0.00	.000306	10	1.80
DBWR	0M	0.00	0M	120.8G	802.32	37.8629	6	1.67
DBWR (Data File)	0M	0.00	0M	120.8G	802.32	37.8629	0	
DBWR (Control File)	0M	0.00	0M	0M	0.00	0M	6	1.67
LGWR	2M	0.05	.000612	39G	587.91	12.2329	180	1.93
LGWR (Log File)	0M	0.01	0M	39G	587.83	12.2316	36	1.92
LGWR (Control File)	2M	0.04	.000612	4M	0.08	.001224	144	1.94
Buffer Cache Reads	17.1G	638.76	5.36845	0M	0.00	0M	2086.6K	1.96
Buffer Cache Reads (Data File)	17.1G	638.76	5.36845	0M	0.00	0M	2086.6K	1.96
Direct Writes	0M	0.00	0M	1M	0.02	.000306	0	
Direct Writes (Data File)	0M	0.00	0M	1M	0.02	.000306	0	
Direct Reads	1M	0.02	.000306	0M	0.00	0M	0	
Direct Reads (Data File)	1M	0.02	.000306	0M	0.00	0M	0	
Streams AQ	0M	0.00	0M	0M	0.00	0M	1	2.00
Streams AQ (Data File)	0M	0.00	0M	0M	0.00	0M	1	2.00
TOTAL:	59.2G	664.34	18.5417	247.3G	1454.23	77.5164	2257.8K	2.31

Choose a VM

Based on the information that you collected from the AWR report, the next step is to choose a VM of a similar size that meets your requirements. For more information about available VMs, see Memory optimized virtual machine sizes.

Fine-tune the VM sizing with a similar VM series based on the ACU

After you choose the VM, pay attention to the Azure compute unit (ACU) for the VM. You might choose a different VM based on the ACU value that better suits your requirements. For more information, see Azure compute unit.

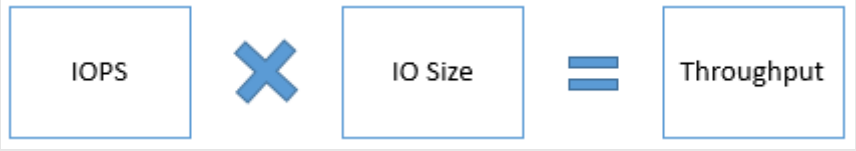
M-series*

ACU: 160-180

Size	vCPU	Memory: GiB	Local SSD: GiB	Max data disks	Max cached and local disk throughput: IOPS / MBps (cache size in GiB)	Max uncached disk throughput: IOPS / MBps	Max NICs / Network bandwidth
Standard_M64ms	64	1792	2048	32	80,000 / 800 (6348)	40,000 / 1,000	32 / extremely high
Standard_M128s**	128	2048	4096	64	160,000 / 1,600 (12,696)	80,000 / 2,000	32 / extremely high

Network throughput

The following diagram shows the relation between throughput and IOPS:



The total network throughput is estimated based on the following information:

- SQL*Net traffic
- MBps times the number of servers (outbound stream, such as Oracle Data Guard)
- Other factors, such as application replication

background timeouts	20,862	6.39	0.00
branch node splits	614	0.19	0.00
buffer is not pinned count	238,355,647	72,965.81	14.03
buffer is pinned count	353,113,420	108,095.64	20.78
bytes received via SQL*Net from client	1,758,217,872	538,228.42	103.49
bytes sent via SQL*Net to client	1,582,672,212	484,490.11	93.16
calls to get snapshot scn: kcmgss	30,773,323	30,236.05	5.81
calls to kcmgas	17,354,528	5,312.60	1.02
calls to kcmgcs	3,293,854	1,008.32	0.19
cell physical IO interconnect bytes	329,064,153,600	100,733,636.76	19,369.14
change write time	112,450	34.42	0.01
cleanout - number of ktugct calls	4,186,965	1,281.72	0.25
cleanouts and rollbacks - consistent read gets	4,231	1.30	0.00

Based on your network bandwidth requirements, there are various gateway types for you to choose from. These types include basic, VpnGw, and Azure ExpressRoute. For more information, see VPN Gateway pricing.

Recommendations

- Network latency is higher compared to an on-premises deployment. Reducing network round trips can greatly improve performance.
- To reduce round-trips, consolidate applications that have high transactions or *chatty* apps on the same virtual machine.
- Use virtual machines with accelerated networking for better network performance.
- For certain Linux distributions, consider enabling TRIM/UNMAP support.
- Install Oracle Enterprise Manager on a separate virtual machine.
- Huge pages aren't enabled on Linux by default. Consider enabling huge pages, and set `use_large_pages = ONLY` on the Oracle DB. This approach might help increase performance. For more information, see USE_LARGE_PAGES.

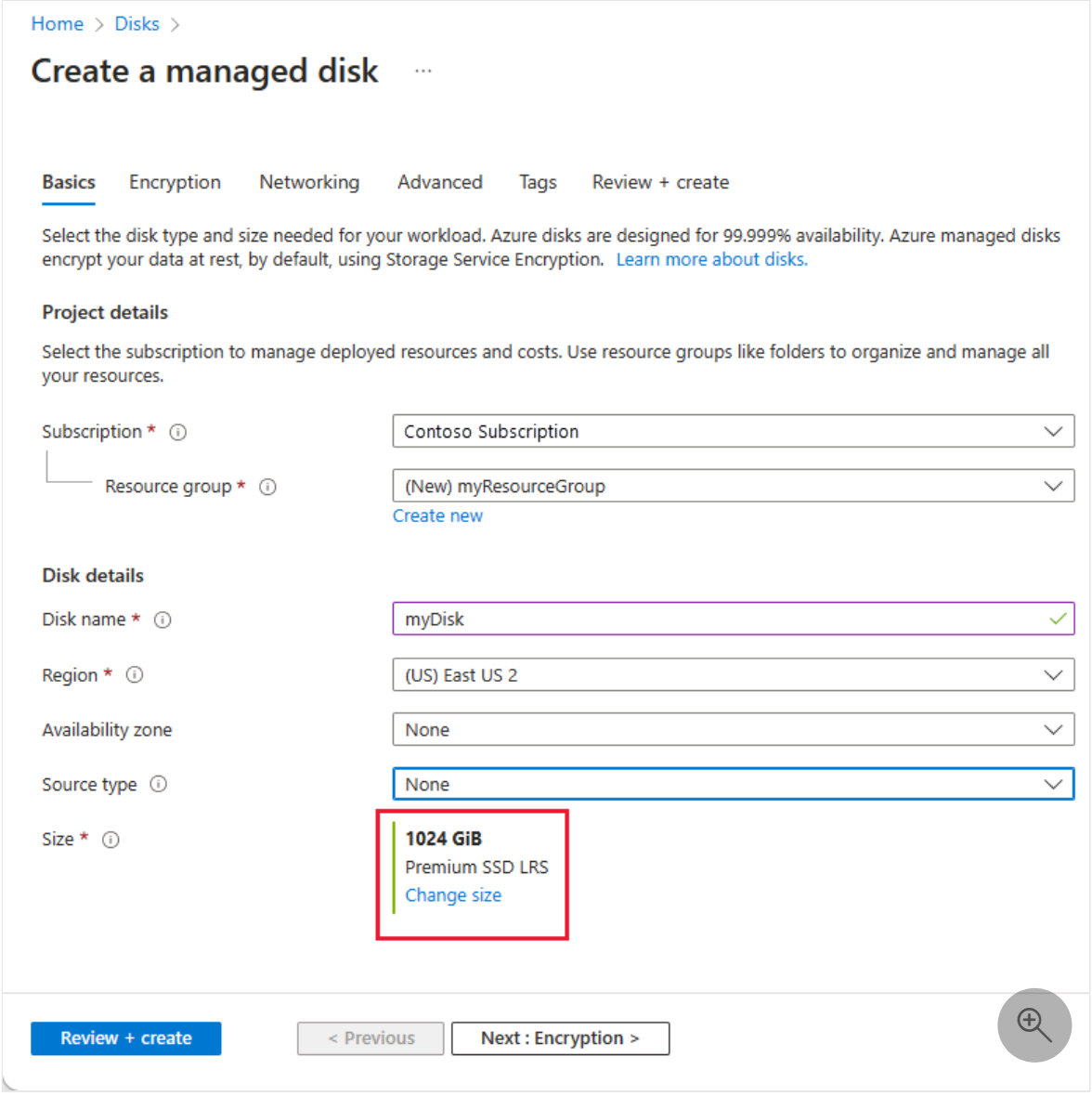
Disk types and configurations

Here are some tips as you consider disks.

- **Default OS disks:** These disk types offer persistent data and caching. They're optimized for operating system access at startup, and aren't designed for either transactional or data warehouse (analytical) workloads.

- **Managed disks:** Azure manages the storage accounts that you use for your VM disks. You specify the disk type and the size of the disk that you need. The type is most often Premium (SSD) for Oracle workloads. Azure creates and manages the disk for you. A premium SSD-managed disk is only available for memory-optimized and designed VM series. After you choose a particular VM size, the

menu shows only the available premium storage SKUs that are based on that VM size.



After you configure your storage on a VM, you might want to load test the disks before you create a database. Knowing the I/O rate in terms of both latency and throughput can help you determine if the VMs support the expected throughput with latency targets. There are several tools for application load testing, such as Oracle Orion, Sysbench, SLOB, and Fio.

Run the load test again after you deploy an Oracle database. Start your regular and peak workloads, and the results show you the baseline of your environment. Be realistic in the workload test. It doesn't make sense to run a workload that is nothing like what you run on the VM in reality.

Because Oracle can be an I/O intensive database, it's important to size the storage based on the IOPS rate rather than the storage size. For example, if the required IOPS value is 5,000, but you only need 200 GB, you might still get the P30 class premium disk even though it comes with more than 200 GB of storage.

You can get the IOPS rate from the AWR report. The redo log, physical reads, and writes rate determine the IOPS rate. Always verify that the VM series you choose has the ability to handle the I/O demand of the workload. If the VM has a lower I/O limit than the storage, the VM sets the limit maximum.

Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	69.4	0.0	0.00	0.01
DB CPU(s):	4.7	0.0	0.00	0.00
Redo size:	12,261,192.2	2,357.6		
Logical reads:	198,807.9	38.2		
Block changes:	76,974.9	14.8		
Physical reads:	687.4	0.1		
Physical writes:	4,846.9	0.9		
User calls:	10,406.6	2.0		
Parses:	5.5	0.0		
Hard parses:	0.1	0.0		
W/A MB processed:	0.1	0.0		
Logons:	0.2	0.0		
Executes:	38,500.2	7.4		
Rollbacks:	0.0	0.0		
Transactions:	5,200.7			

For example, the redo size is 12,200,000 bytes per second, which is equal to 11.63 MBPs. The IOPS value is 12,200,000 / 2,358 = 5,174.

After you have a clear picture of the I/O requirements, you can choose a combination of drives that are best suited to meet those requirements.

Disk type recommendations

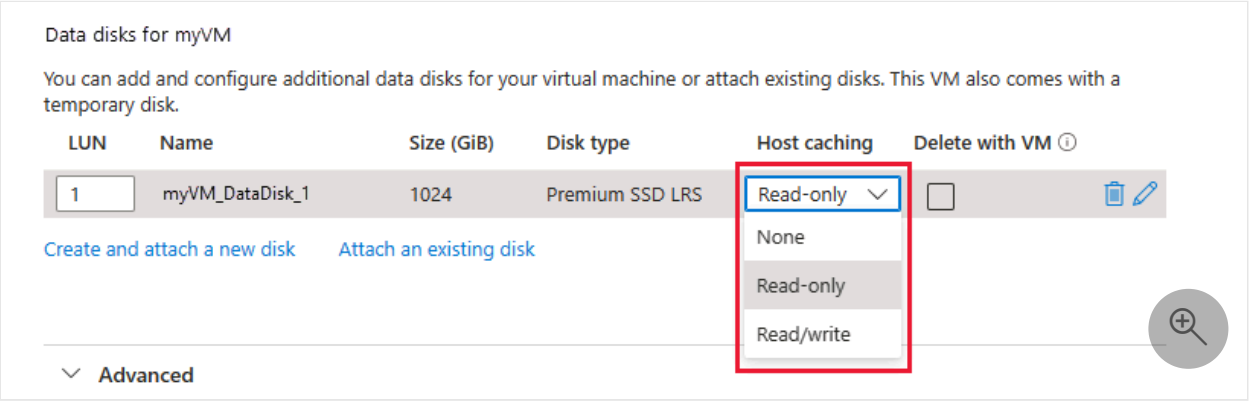
- For data tablespace, spread the I/O workload across several disks by using managed storage or Oracle Automatic Storage Management (ASM).
- Use Oracle advanced compression to reduce I/O for both data and indexes.
- Separate redo logs, temp, and undo tablespaces on separate data disks.
- Don't put any application files on default operating system disks. These disks aren't optimized for fast VM boot times, and they might not provide good performance for your application.
- When you're using M-Series VMs on premium storage, enable write accelerator on the redo logs disk.
- Consider moving redo logs with high latency to the ultra disk.

Disk cache settings

Although you have three options for host caching, only read-only caching is recommended for a database workload on an Oracle database. Read/write can introduce significant vulnerabilities to a data file, because the goal of a database write is to record it to the data file, not to cache the information. With read-only, all requests are cached for future reads. All writes continue to be written to disk.

Disk cache recommendations

To maximize throughput, start with read-only for host caching whenever possible. For premium storage, keep in mind that you must disable the barriers when you mount the file system with the read-only options. Update the */etc/fstab* file with the universally unique identifier to the disks.



- For operating system disks, use premium SSD with read-write host caching.
- For data disks that contain the following, use premium SSD with read-only host caching: Oracle data files, temp files, control files, block change tracking files, BFILEs, files for external tables, and flashback logs.
- For data disks that contain Oracle online redo log files, use premium SSD or UltraDisk with no host caching, the **None** option. Oracle redo log files that are archived and Oracle Recovery Manager backup sets, can also reside with the online redo log files. Host caching is limited to 4095 GiB, so don't allocate a premium SSD larger than P50 with host caching. If you need more than 4 TiB of storage, stripe several premium SSDs with RAID-0. Use Linux LVM2 or Oracle Automatic Storage Management.

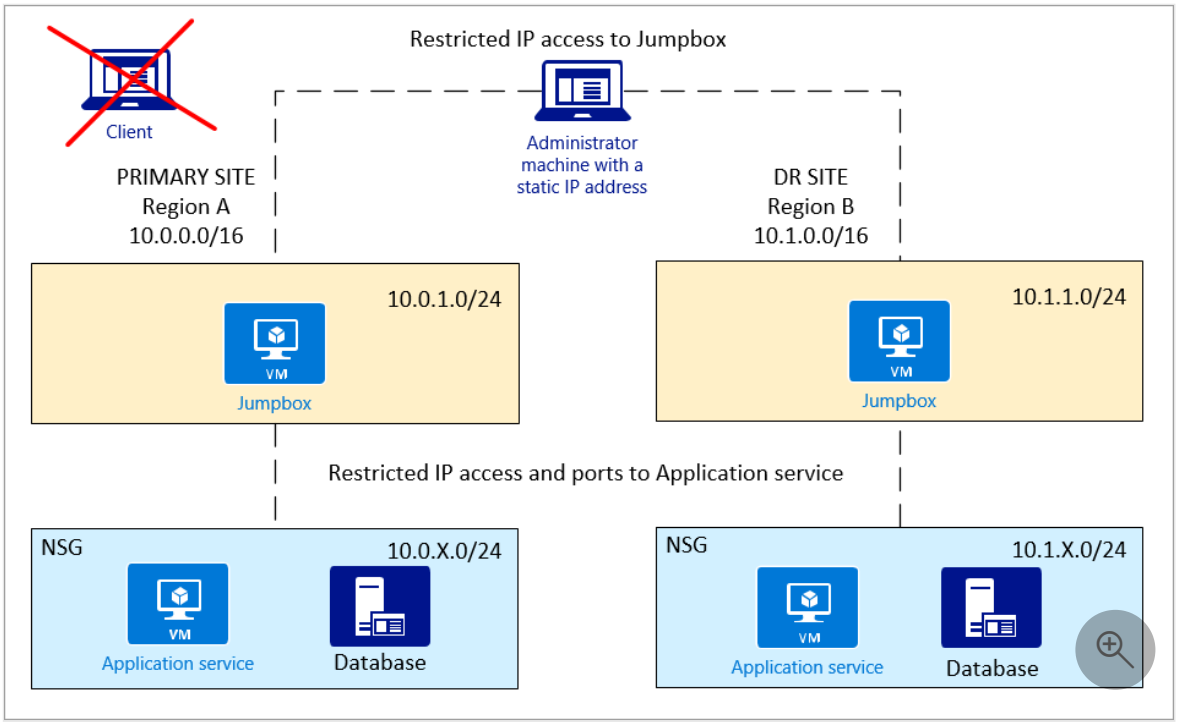
If workloads vary greatly between the day and evening, and the I/O workload can support it, P1-P20 premium SSD with bursting might provide the performance required during night-time batch loads or limited I/O demands.

Security

After you set up and configure your Azure environment, you need to secure your network. Here are some recommendations:

- **NSG policy:** You can define your NSG by a subnet or a network interface card. It's simpler to control access at the subnet level, both for security and for force-routing application firewalls.

- **Jumpbox:** For more secure access, administrators shouldn't directly connect to the application service or database. Use a jumpbox between the administrator machine and Azure resources.



  The administrator machine should only offer IP-restricted access to the jumpbox. The jumpbox should have access to the application and database.

- **Private network (subnets):** It's a good idea to have the application service and database on separate subnets, so that NSG policy can set better control.

Resources

- Configure Oracle ASM
- Configure Oracle Data Guard
- Configure Oracle GoldenGate
- Oracle backup and recovery

Next steps

- Create a complete Linux virtual machine with the Azure CLI
- Explore VM deployment Azure CLI samples

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Overview - Oracle Database@Azure

Article • 07/11/2024

Oracle Database@Azure is an Oracle database service running on Oracle Cloud Infrastructure (OCI), colocated in Microsoft data centers. This ensures that the Oracle Database@Azure service has the fastest possible access to Azure resources and applications.

Oracle Database@Azure allows you to subscribe to the Oracle Database Service inside your Azure environment. All infrastructure for your Oracle Database Service is located in Azure's physical data centers, giving your critical database workloads the high-performance and low-latency they require. Like other Azure services, Oracle Database@Azure uses an Azure Virtual Network for networking, managed within the Azure environment. The service uses the Azure tenancy's identity management and authorization, which can be either the Azure native identity service or a federated identity provider. The service allows you to monitor database metrics, audit logs, events, logging data, and telemetry natively in Azure.

Oracle Database@Azure runs on infrastructure managed by Oracle's expert Cloud Infrastructure operations team. The operations team performs software patching, infrastructure updates, and other operations through a connection to OCI. While the service requires that customers have an OCI tenancy, most service activities take place in the Azure environment.

Oracle Database@Azure interfaces

You can provision Oracle Database@Azure using the Azure portal and Azure APIs, SDKs and Terraform. Management of Oracle database system infrastructure and VM cluster resources takes place in the Azure portal as well.

For Oracle Container Databases (CDB) and Oracle Pluggable Databases (PDB), some management tasks are completed using the OCI console.

Database and application developers work in the Azure portal or use Azure tools (Azure API, SDK, Terraform) to interact with Oracle Database@Azure databases.

Purchase Oracle Database@Azure

To purchase Oracle Database@Azure, contact Oracle's sales team or your Oracle sales representative for a sale offer. Oracle Sales team creates an Azure Private Offer in the

Azure Marketplace for your service. After an offer has been created for your organization, you can accept the offer and complete the purchase in the Azure portal's Marketplace service. For more information on Azure private offers, see Overview of the commercial marketplace and enterprise procurement.

Billing and payment for the service is done through Azure. Payment for Oracle Database@Azure counts toward your Microsoft Azure Consumption Commitment (MACC). Existing Oracle Database software customers can use the Bring Your Own License (BYOL) option or Unlimited License Agreements (ULAs). On your regular Microsoft Azure invoices, you can see charges for Oracle Database@Azure alongside charges for your other Azure Marketplace services.

Compliance

Oracle Database@Azure is an Oracle Cloud database service that runs Oracle Database workloads in a customer's Azure environment. Oracle Database@Azure offers various Oracle Database Services through customer's Microsoft Azure environment. This service allows customers to monitor database metrics, audit logs, events, logging data, and telemetry natively in Azure. It runs on infrastructure managed by Oracle's Cloud Infrastructure operations team who performs software patching, infrastructure updates, and other operations through a connection to Oracle Cloud.
All infrastructure for Oracle Database@Azure is co-located in Azure's physical data centers and uses Azure Virtual Network for networking, managed within the Azure environment. Federated identity and access management for Oracle Database@Azure is provided by Microsoft Entra ID.

For detailed information on the compliance certifications please visit Microsoft Services Trust Portal and Oracle compliance website. If you have further questions about OracleDB@Azure compliance please reach out to your account team and/or get information through Oracle and Microsoft support for Oracle Database@Azure.

Available regions

Oracle Database@Azure is available in the following locations. Oracle Database@Azure infrastructure resources must be provisioned in the Azure regions listed.

Expand table

Azure region	Oracle Exadata Database@Azure	Oracle Autonomous Database@Azure
East US	✓	✓
Germany West Central	✓	✓
France Central	✓	✓
UK South	✓	✓
Canada Central	✓	✓
Australia East	✓	✓

Azure Support scope and contact information

See Contact Microsoft Azure Support in the Azure documentation for information on Azure support. For SLA information about the service offering, please refer to the Oracle PaaS and IaaS Public Cloud Services Pillar Document

Next steps

- Onboard with Oracle Database@Azure
- Provision and manage Oracle Database@Azure
- Oracle Database@Azure support information
- Network planning for Oracle Database@Azure
- Groups and roles for Oracle Database@Azure

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A

Introduction to Oracle on Azure adoption scenarios

Article • 08/01/2024

This article describes how to set up and manage Oracle workloads within your Azure landing zone. It also describes specific architectural strategies and provides reference implementations for Oracle database systems on Azure. The guidance assumes that you have an Azure landing zone. For more information, see What is an Azure landing zone? and Landing zone implementation options.

Define Oracle on Azure

Oracle on Azure adoption scenarios provide two principal technology platform options:

- **Oracle on Azure Virtual Machines**: Run Oracle databases and enterprise applications, such as Siebel, PeopleSoft, JD Edwards, E-Business Suite, or customized WebLogic Server applications on Azure infrastructure. You can use an Oracle Linux image, Red Hat Enterprise Linux (RHEL), or another endorsed operating system. There are multiple VMs and storage options available.
- **Oracle Database@Azure**: You can use Oracle Database@Azure to run Oracle Exadata infrastructure in Azure. Oracle Exadata is a high-performance database platform. Oracle Database@Azure supports tools, such as Oracle Real Application Clusters (RAC) and Oracle Data Guard. Oracle enterprise applications such as Siebel, PeopleSoft, JD Edwards, E-Business Suite, or customized WebLogic Server applications run on Azure VMs and can connect to Oracle Database@Azure.

For more information, see Oracle on Azure overview.

Oracle on Azure architectures

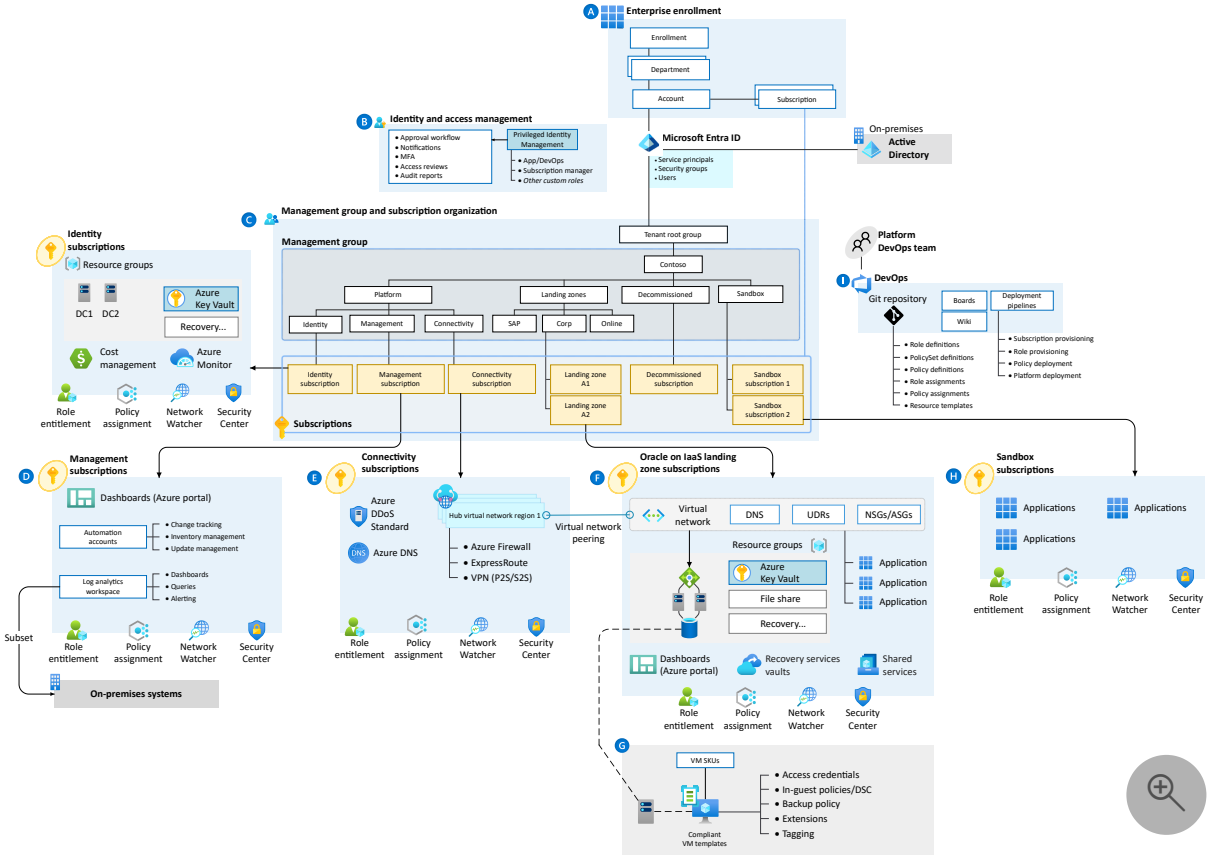
You can use Oracle on Azure to integrate database services for Oracle with Azure cloud capabilities. Apply data analytics and generative AI to your Oracle data. Monitor your apps and Oracle database service with a single view in Azure.

The following diagrams provide a conceptual reference architecture for Oracle on Azure Virtual Machines and Oracle Database@Azure that you can use to accelerate cloud adoption. The diagrams show critical design areas for Azure landing zones. They also present a network layout that demonstrates architectural principles, but they don't detail an entire enterprise network.

Use the reference architectures as a starting point. Modify the reference architectures to fit your specific business and technical requirements when you plan to implement your landing zone.

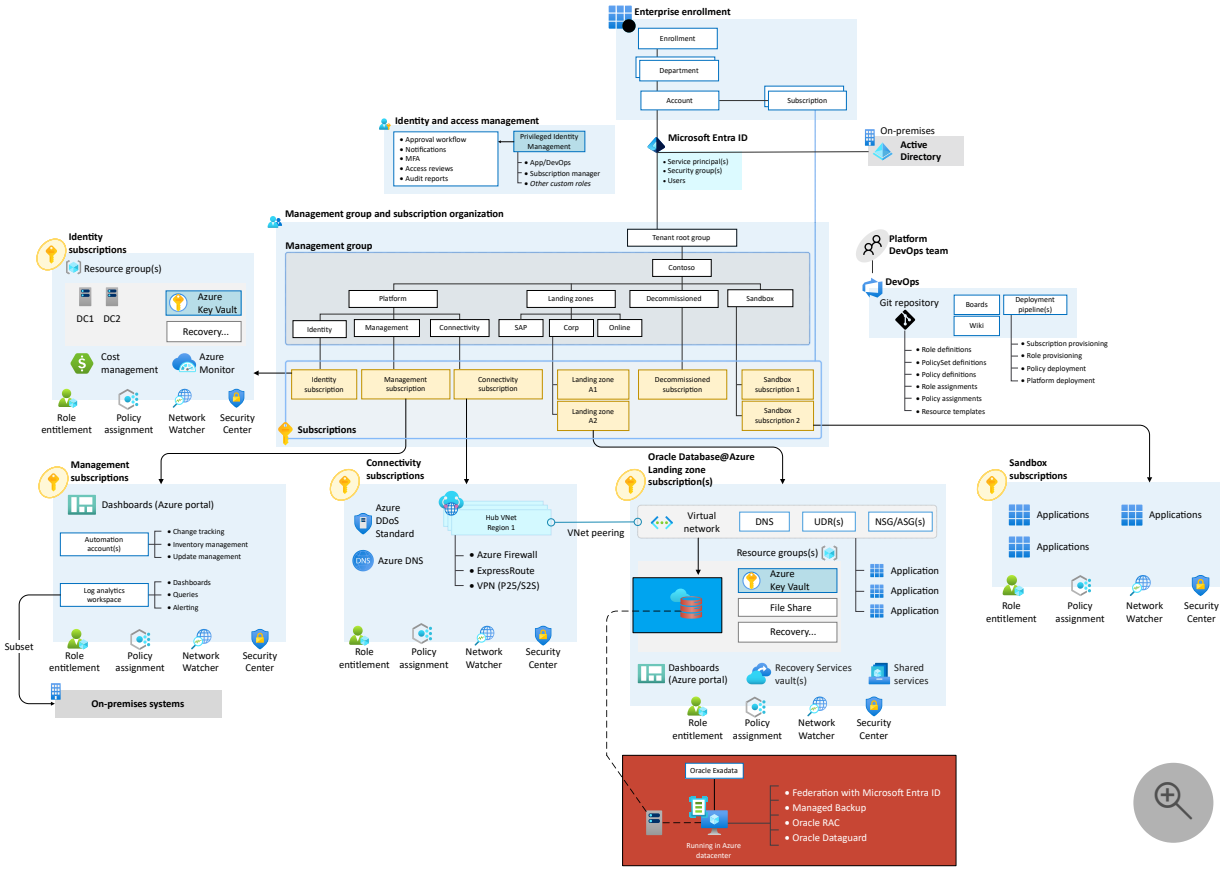
Landing zone architecture for Oracle on Azure Virtual Machines

The following diagram shows Oracle on Azure Virtual Machines deployed to an application landing zone. It also shows the critical design areas that support this deployment. The Oracle databases run on VMs. You can change the number and size of VMs to accommodate your needs.



Landing zone architecture for Oracle Database@Azure

The following diagram shows Oracle Database@Azure deployed to an application landing zone. It also shows the critical design areas that support this deployment. The Oracle databases run on Oracle Exadata VM clusters in Azure. You can change the number and size of VM clusters to accommodate your needs.



Next step

Strategic impact of Oracle on Azure

Feedback

Was this page helpful? Yes No

Azure Architecture Center

Guidance for architecting solutions on Azure using established patterns and practices

ARCHITECTU...
Browse Azure architectures

CONCEPT
Explore cloud best practices

WHAT'S NEW
See what's new

HOW-TO...
Well-Architected Framework...

HOW-TO...
Cloud Adoption Framework...

Architecting applications on Azure

Best practices and patterns for building applications on Microsoft Azure:

Design for the cloud

Principles of a well-designed application

Best practices in cloud applications

Responsible engineering

Application design patterns

Architect multitenant solutions on Azure

Build microservices on Azure

Optimize your workload

Guiding tenets for your architecture

Examine your workload

Performance tuning

Performance antipatterns

Secure your workload

Choose the right technology

Essential scenarios

Choose a compute service
Choose a Kubernetes at the edge option
Choose a data store
Choose a load-balancing service
Choose a messaging service

Architecture for startups
Azure and Power Platform solutions
Azure and Microsoft 365 solutions
AWS to Azure services comparison
Google Cloud services comparison

Technology areas

Explore architectures and guides for different technologies.

Popular articles

- Baseline OpenAI end-to-end chat architecture
- AWS to Azure services comparison
- AKS production baseline
- Google Cloud to Azure services comparison
- Cloud design patterns
- CQRS design pattern
- Best practices in cloud applications
- Web API design
- Choose your Azure compute service
- Application architecture fundamentals
- Hub-spoke network topology
- Architect multitenant solutions on Azure

AI & Machine Learning

- Baseline OpenAI end-to-end chat architecture
- AI architecture design
- Conversational bot
- Machine learning options
- Natural language processing
- Azure AI services options
- Team Data Science Process

Analytics

- Analytics architecture design
- Choose an analytical data store in Azure
- Choose a data analytics technology in Azure
- Analytics end-to-end with Azure Synapse
- Automated enterprise BI with Azure Data Factory
- Stream processing with Azure Databricks
- Databricks monitoring
- Real-time fraud detection
- Predict the length of stay in hospitals

Databases

- Databases architecture design
- Big Data architectures
- Build a scalable system for massive data
- Choose a data store
- Extract, transform, and load (ETL)
- Online analytical processing (OLAP)
- Online transaction processing (OLTP)
- Data lakes

DevOps

- Advanced ARM templates
- DevOps with containers

Enterprise integration

- Basic enterprise integration
- Enterprise BI with SQL Data Warehouse
- Enterprise integration with queues and events

High-performance computing (HPC)

- Computer-aided engineering
- HPC video rendering
- Image modeling

Introduction to HPC on Azure >

Identity

- Identity in multitenant applications
- Choose an Active Directory integration architecture
- Integrate on-premises AD with Microsoft Entra ID
- Extend AD DS to Azure
- Create an AD DS forest in Azure
- Extend AD FS to Azure

Internet of Things (IoT)

- IoT architecture
- Automotive IoT data

Microservices

- Domain analysis
- Tactical DDD
- Identify microservice boundaries
- Design a microservices architecture
- Monitor microservices in Azure Kubernetes Service (AKS)
- CI/CD for microservices
- CI/CD for microservices on Kubernetes
- Azure Kubernetes Service (AKS)
- Azure Service Fabric

Introduction to microservices on Azure >

Networking

- Choose a hybrid network architecture
- ExpressRoute
- ExpressRoute with VPN failover
- Troubleshoot a hybrid VPN connection

Serverless applications

- Code walkthrough
- Serverless event processing
- Serverless web application

Introduction to serverless applications on Azure >

- Hub-spoke topology
- Implement a secure hybrid network
- Highly available network virtual appliances
- Segment virtual networks

Virtual machine (VM) workloads

- Linux VM deployment
- Windows VM deployment
- N-tier application with Cassandra (Linux)
- Multi-region N-tier application
- WordPress on virtual machines

SAP

- Overview
- SAP HANA on Azure (Large Instances)
- SAP HANA scale-up on Linux
- SAP NetWeaver in Windows on Azure
- SAP S/4HANA in Linux on Azure
- SAP BW/4HANA in Linux on Azure
- SAP deployment using an Oracle database

Web apps

- Basic web application
- Baseline zone-redundant web application
- Multi-region deployment
- E-commerce API management
- E-commerce product search
- Publish internal APIs externally
- Securely managed web application
- Highly available web application

Build your skills with Microsoft Learn training

Build great solutions with the Microsoft Azure Well-Architected Framework

Introduction to the Well-Architected Framework

Azure fundamentals - Cloud concepts

Architect infrastructure operations in Azure

Tour the N-tier architecture style

Azure fundamentals - Azure architecture and services

Azure DevOps Roadmap

Article • 08/21/2024

| What's New | Developer Community | DevOps Blog | Documentation |

Product roadmap

This feature list is a peek into our roadmap. It identifies some of the significant features we are currently working on and a rough timeframe for when you can expect to see them. It is not comprehensive but is intended to provide some visibility into key investments. At the top you will find a list of our large multi-quarter initiatives and the features that they break down into. Further down you will find the full list of significant features we have planned.

Each feature is linked to an article where you can learn more about a particular item. These features and dates are the current plans and are subject to change. The Timeframe columns reflect when we expect the feature to be available.

Initiatives

GitHub Advanced Security for Azure DevOps

GitHub Advanced Security (GHAS) for Azure DevOps is now generally available. Any project collection administrator can now enable Advanced Security for their organization, projects and repos from the Project Settings or Organization Settings. You can learn more about how to configure GitHub Advanced Security for Azure DevOps in our documentation.

New capabilities we expect to deliver include:

Expand table

Feature	Area	Quarter
Display contextual comments to pull requests containing newly introduced Advanced Security findings	GitHub Advanced Security for Azure DevOps	2024 Q3
Determine detected partner secrets validity	GitHub Advanced Security	2024

Feature	Area	Quarter
	for Azure DevOps	Q4
Automatically fix detected dependency scanning vulnerabilities with Dependabot security updates	GitHub Advanced Security for Azure DevOps	Future

Minimizing the risks associated with credential theft

Azure DevOps supports many different authentication mechanisms, including basic authentication, personal access tokens (PATs), SSH, and Microsoft Entra ID (formerly Azure Active Directory) access tokens. These mechanisms are not created equally from a security perspective, especially when it comes to the potential for credential theft. For example, unintended leakage of credentials like PATs can let malicious actors into Azure DevOps organizations where they can gain access to critical assets like source code, pivot toward supply chain attacks, or even pivot toward compromising production infrastructure. To minimize the risks of credential theft, we will focus our efforts in the upcoming quarters in the following areas:

- Enable administrators to improve authentication security through control plane policies.
- Reducing the need for PATs and other stealable secrets by adding support for more secure alternatives.
- Deepening Azure DevOps' integration with Microsoft Entra ID to better support its various security features.
- Avoiding the need to store production secrets in Azure Pipelines service connections.

Expand table

Feature	Area	Quarter
PAT lifecycle APIs	General	✓ 2022 Q4
Control plane for personal access tokens (PAT)	General	✓ 2022 Q4
Managed Identity and Service Principal support (preview)	General	✓ 2023 Q1
Workload identity federation for Azure Deployments (preview)	Pipelines	✓ 2023 Q3
Granular scopes for Azure Active Directory OAuth	General	✓ 2023 Q3
Managed Identity and Service Principal support (GA)	General	✓ 2023 Q3

Feature	Area	Quarter
Workload identity federation for Azure service connection (GA)	Pipelines	✓ 2024 Q1
Workload identity federation for Docker service connection	Pipelines	2024 H1
Full web support for Conditional Access Policies	General	2024 H1
Policies to disable authentication methods	General	Future

Improved Boards + GitHub Integration

The existing Azure Boards + GitHub integration has been in place for several years now. The integration is a great starting point, but it does not offer the level of traceability that our customers have grown accustomed to. Based on customer feedback, we have put together set of investments to enhance this integration. Our goal is to improve upon it so that Azure Boards customers who choose to use GitHub repositories can maintain an equivalent level of traceability to having repositories in Azure DevOps.

These investments include:

Expand table

Feature	Area	Quarter
Add link to GitHub commit or pull request from work item	Boards	✓ 2024 Q1
Show more details about a GitHub pull request	Boards	✓ 2024 Q1
Improve scalability when searching and linking GitHub repos to an Azure DevOps project	Boards	✓ 2024 Q2
AB# links on GitHub pull request (preview)	Boards	✓ 2024 Q2
Create branch on GitHub repository from work item	Boards	✓ 2024 Q3
GH! mentions support	Boards	2024 Q4

YAML and release pipelines feature parity

For the past several years, all our pipelines investments have been in the area of YAML pipelines. Furthermore, all our security improvements have been for YAML pipelines. For example, with YAML pipelines, the control over protected resources (e.g., repositories, service connections, etc.) is in the hands of the resource owners as opposed to pipeline authors. The job access tokens that are used in YAML pipelines are scoped to specific repositories that are specified in the YAML file. These are just two examples of security

features that are available for YAML pipelines. For these reasons, we recommend using YAML pipelines over classic. Adoption of YAML over classic has been significant for builds (CI). However, many customers have continued to use classic release management pipelines over YAML for releases (CD). The primary reason for this is the lack of parity in various CD features between the two solutions. Over the past year, we addressed several gaps in this area, notably in **Checks**. Checks are the primary mechanism in YAML pipelines to gate promotion of a build from one stage to another. We will continue to address gaps in other areas over the next year. Our focus will be on user experiences, traceability, and environments.

Expand table

Feature	Area	Quarter
Auditing for checks	Pipelines	✓ 2022 Q4
Custom variables in checks	Pipelines	✓ 2023 Q1
Checks scalability	Pipelines	✓ 2023 Q2
Bypass approvals and checks	Pipelines	✓ 2023 Q4
Sequencing approvals and other checks	Pipelines	✓ 2024 Q1
Deferred approvals	Pipelines	✓ 2024 Q1
Rerun single stage	Pipelines	✓ 2024 Q1
Stage-level concurrency	Pipelines	2024 Q3
Manual queuing of stages	Pipelines	2024 H2
Stage-level traceability	Pipelines	2024 H2
Service connections in checks	Pipelines	Future
Checks extensibility	Pipelines	Future

All features

Azure DevOps Services

Expand table

Timeframe	Feature	Area
2024 Q3	Full web support for Conditional Access Policies	General
	Permanently delete work item attachment	Boards
	Create branch on GitHub repository from work item	Boards
	REST API suppport for connecting GitHub repositories (general availability)	Boards
	Enable area and iteration level fields	Boards
	Stage-level concurrency	Pipelines
	Manual queuing of stages	Pipelines
	Removal of TFVC from new projects	Repos
	Restore deleted test plans and test suites using REST API	Test Plans
	Display contextual comments to pull requests containing newly introduced Advanced Security findings	GitHub Advanced Security for Azure DevOps
	Display contextual comments to pull requests containing newly introduced dependencies	GitHub Advanced Security for Azure DevOps
2024 Q4	Secret Validity Checking	GitHub Advanced Security for Azure DevOps
	New Boards Hub on by default	Boards
	Improve cut and paste of images into work item comments	Boards
	GH! mentions support	Boards
Future	Auditing GA	General
	Policies to disable authentication methods	General
	Access events for PAT, SSH will be available in the Auditing Log	General
	Commits search	General
	PR search	General
	Additional fields on page filters	Boards

Timeframe	Feature	Area
	Show build status when using a YAML build pipeline with GitHub repository	Boards
	Show status of GitHub Actions work flow run	Boards
	Track repo cloning	Repos
	Ability to run tasks on next available Node version, if targeted version is not available	Pipelines
	Support Pipelines App with GitHub Enterprise	Pipelines
	Service connections in checks	Pipelines
	Checks extensibility	Pipelines
	Stage-level traceability	Pipelines
	Package promote task in Azure Pipelines	Artifacts
	Deprecate old Azure Artifacts tasks in Azure Pipelines and default to new, auth-only tasks	Artifacts
	Test Plan task in Pipelines	Test Plans
	JUnit / Java support in Test Plans	Test Plans
	New Test Runs experience	Test Plans
	Improve traceability of in-progress test executions	Test Plans
	Dependabot Security Update Support	GitHub Advanced Security for Azure DevOps

Azure DevOps Server

Expand table

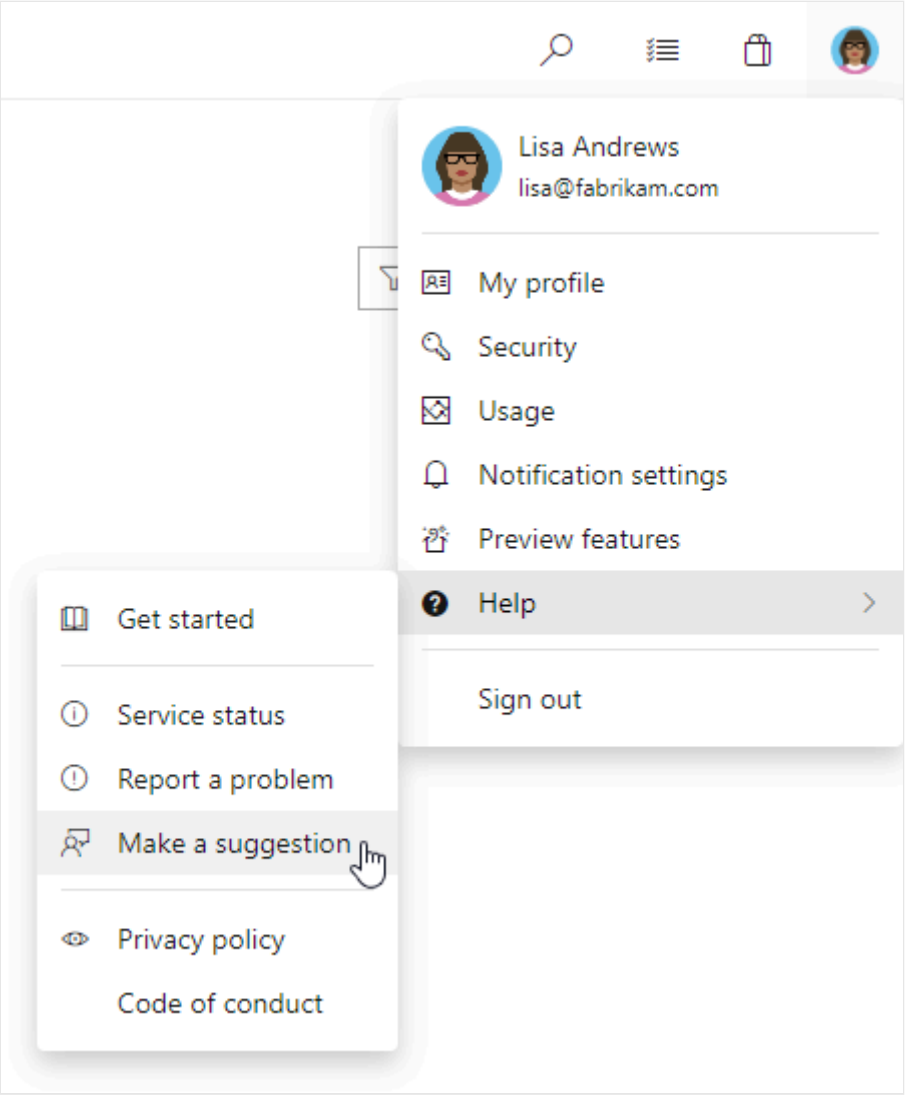
Timeframe	Feature	Area
2025 H1	Automated team work item rules	Boards
	Improved GitHub app and AB#{ID} validation	Boards
	Add link to GitHub commit or pull request from work item (preview)	Boards

Timeframe	Feature	Area
	Show more details about a GitHub pull request (preview)	Boards
	Markdown editor for work item comments (preview)	Boards
	All in-the-box tasks run on Node 16	Pipelines
	Ability to download and install old runners on self-hosted agents	Pipelines
	Ability to run tasks on next available Node version, if targeted version is not available	Pipelines
	Removal of Node 6 and 10 from Microsoft hosted pools	Pipelines
	Sequencing approvals and other checks	Pipelines
	In-product recommendations for secure settings	Pipelines
	Manual queuing of stages	Pipelines
	Rerun single stage	Pipelines
	Stage-level concurrency	Pipelines
	Stage-level traceability	Pipelines
2025 H2	Reduce time to filter test plans and test suites in copy and import operations	Test Plans
	Code coverage policy at a folder level	Test Plans
	New Boards Hub on by default	Boards
	Create branch on GitHub repository from work item	Boards
	Improve scalability when searching and linking GitHub repos to an Azure DevOps project	Boards
Future	Policies to disable authentication methods	General
	Substring search	General
	Commits search	General
	Additional fields on page filters	Boards
	Delivery plans improvements to filtering by parent	Boards
	Markdown editor for work item multi-line fields	Boards
	Track repo cloning	Repos

Timeframe	Feature	Area
	Stop shipping Node 6 and Node 10 runners with the agent	Pipelines
	Ship a Node 16 only agent in addition to the one that has all three versions (6, 10, 16)	Pipelines
	Support Pipelines App with GitHub Enterprise	Pipelines
	Service connections in checks	Pipelines
	Checks extensibility	Pipelines
	Package promote task in Azure Pipelines	Artifacts
	Deprecate old Azure Artifacts tasks in Azure Pipelines and default to new, auth-only tasks	Artifacts
	Dashboard Global Parameter	Reporting
	Dashboard Template	Reporting
	Sort test plans by any column	Test Plans
	Test Plan task in Pipelines	Test Plans
	JUnit / Java support in Test Plans	Test Plans
	Improved co-relation between test run and test case versions	Test Plans
	Restore deleted test plans and test suites using REST API	Test Plans
	Improve traceability of in-progress test executions	Test Plans
	Sort test suites in alphabetical order	Test Plans

How to provide feedback

We would love to hear what you think about these features. Report any problems or suggest a feature through Developer Community.



You can also get advice and your questions answered by the community on Stack Overflow.

FAQs - Oracle on Azure VMs

Article • 08/22/2024

What are the recommended Azure VM types for Oracle on Azure?

- **M Series**: high memory requirements, CPU performance, Lower IO limits, host level caching is low, accelerated networking options.
- **E series** - Availability in all regions, allows for Premium SSD for OS Disk, ephemeral storage to be used for swap.

What is the role of an Oracle Data Guard on Azure?
Data Guard is more focused on disaster recovery (DR) in an on-premises Oracle solution in Azure. It's central to high availability and disaster recovery. It applies mainly to Fast-Start Failover and the DG Broker & Observer. Data Guard provides a high-availability-based Architecture.

Does having an Oracle Data Guard setup on Azure VM between Availability Set/Zones or regions subject to ingress/egress cost?
Yes. There's US$0.02/GB charge for the Data Guard redo transport for a remote standby database in another region. There's no cost for the Data Guard redo transport to a local standby database in another availability zone in the same region.

What are the different design options for Oracle migration to Azure?

- **Good & Fast**: You can choose a solution involving Data Guard or Golden Gate, but that's not cost effective.
- **Good & Cost effective**: You can choose this option with non-Oracle solutions like Azure VM backup cross-region restore, or the Azure NetApp Files cross-region replication but it won't be fast, and you need to give some slacks in their RPO/RTO requirements. Both the Azure NetApp Files cross-region replication has cross-region transport of data included in the product cost.

What is the simple bare minimal Oracle reference architecture on Azure?
Two (2) Azure availability Zone architecture with VM.

Feedback

Was this page helpful? Yes No

Provide product feedback | Get help at Microsoft Q&A