Creating a Certified Oracle Database Environment on an Azure VM

Contents

[In This Guide 2](#_Toc36737654)

[Beginning with Oracle on Azure 3](#_Toc36737655)

[Create a resource group 4](#_Toc36737656)

[Once created, you will receive a successful confirmation. 4](#_Toc36737657)

[Create virtual machine 4](#_Toc36737658)

[Connect to the VM 5](#_Toc36737659)

[Add and Prepare Disk for ASM 6](#_Toc36737660)

[Best practices on storage for Oracle 6](#_Toc36737661)

[Check Resources Deployed 7](#_Toc36737662)

[Set up the Environment 7](#_Toc36737663)

[Configure Added Disk 8](#_Toc36737664)

[File System Management 8](#_Toc36737665)

[Installing and Configuring ASM on an Oracle VM 10](#_Toc36737666)

[Create the database 10](#_Toc36737667)

[Set Oracle variables 13](#_Toc36737668)

[Oracle EM Express connectivity 14](#_Toc36737669)

[Automate database startup and shutdown 15](#_Toc36737670)

[Open ports for connectivity 17](#_Toc36737671)

[Next steps 18](#_Toc36737672)

[Secure your Linux host and Oracle database. 18](#_Toc36737673)

[Monitoring and Managing Oracle databases via Oracle Cloud Control 19](#_Toc36737674)

[Creating a Disaster Recovery Solution for Oracle on Azure 19](#_Toc36737675)

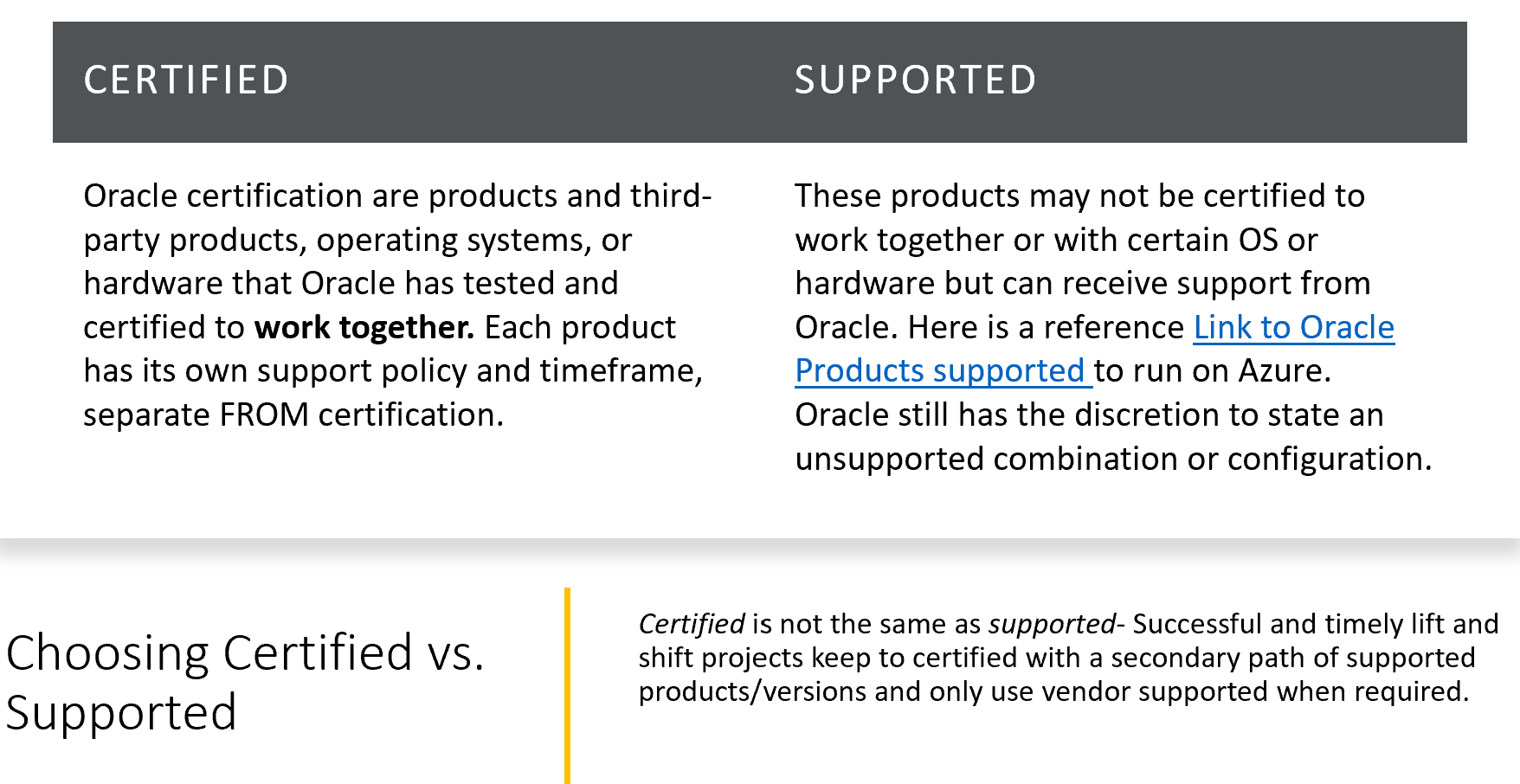
[Creating an Azure VM image template from your Oracle VM 19](#_Toc36737676)

[Cleaning Up Resources in Azure 19](#_Toc36737677)

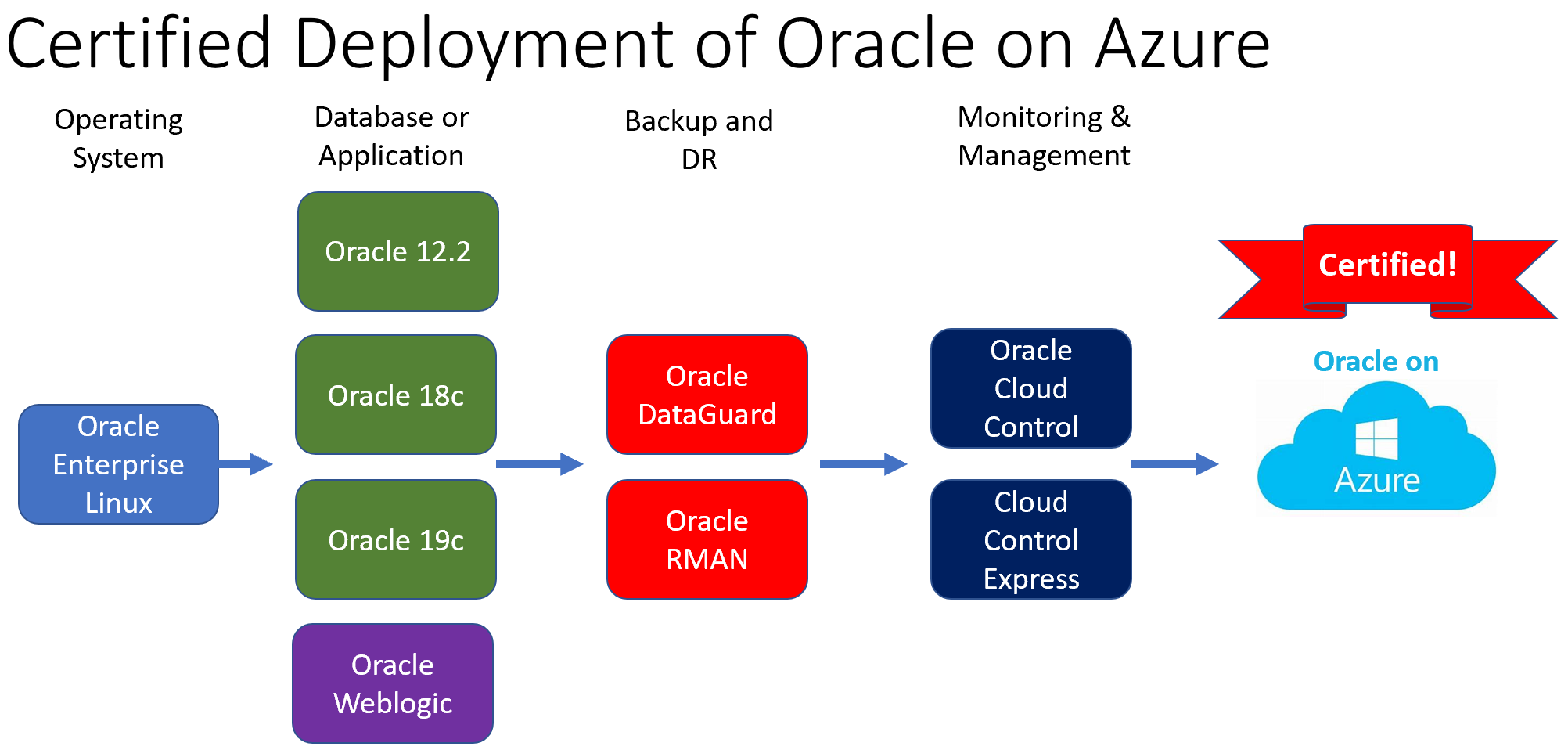
[Removing a Resource Group from Azure 20](#_Toc36737678)

# In This Guide

This guide details using the Azure CLI to deploy an Oracle certified version of Oracle Enterprise Linux virtual machine from the [Oracle marketplace gallery image](https://azuremarketplace.microsoft.com/marketplace/apps/Oracle.OracleDatabase12102EnterpriseEdition?tab=Overview) in order to create a host to install a fully certified Oracle database server from.



[Certified Oracle deployments](https://docs.oracle.com/cd/E25290_01/doc.60/e25224/certifications.htm) on Azure ensure long-term satisfaction with your cloud investment and guaranteed vendor support from Oracle.



Along with the Oracle products listed above, there are significant choices to supplement with high availability, high performance and other services from Azure.

For a list of all supported products on Azure from Oracle, refer to the following information.

# Beginning with Oracle on Azure

If you don't have an Azure subscription, create a [free account](https://azure.microsoft.com/free/?WT.mc_id=A261C142F) before you begin.

Once the server is deployed, you will connect via SSH in order to install and configure the Oracle database. If you choose to install and use the CLI locally, this quickstart 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](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli).

If you choose to use the Azure Cloud Shell, no installation is required locally and Azure Cloud storage connection will allow you to store scripts in the cloud and deploy repeatedly if an ARM template is created at a later date.

# Create a resource group

An Azure resource group is a logical container into which Azure resources are deployed and managed. If a resource group doesn’t already exist to house the Oracle Enterprise Linux VM, one will need to be created. To create a resource group, you’ll need to use the [az group create](https://docs.microsoft.com/en-us/cli/azure/group) command. Consider the name of the resource group to identify the resources residing in the group’s usage and the location decides on which region the resources will be created.

The following example creates a resource group named *myResourceGroup* in the *eastus* location.

az group create --name <Group Name> --location <region>

Once created, you will receive a successful confirmation.

# Create virtual machine

To create a virtual machine (VM), use the [az vm create](https://docs.microsoft.com/en-us/cli/azure/vm) command.

The following example creates a VM named myVM. It also creates SSH keys, if they do not already exist in a default key location. To use a specific set of keys, use the --ssh-key-value option.

First, know how to locate the image that you wish to install for Oracle Enterprise Linux:

az vm image list \

--offer Oracle-Linux \

--all \

--publisher Oracle \

--output table

From the table provided in the output, we’ll use the 7.6 image to create our VM:

az vm create \

--resource-group myResourceGroup \

--name myVM \

--image Oracle:Oracle-Linux:7.6:7.6.3:latest \

--size Standard\_DS2\_v2 \

--admin-username azureuser \

--generate-ssh-keys

If a different release of Oracle Enterprise Linux is desired, simply replace the URN and by adding the “:latest” ensures the most up to date image is used.

Azure CLI will respond to show the VM is running to complete the deployment similar to the following example. Note the value for publicIpAddress. You use this address to access the VM.

"fqdns": "",

"id": "/subscriptions/73aa270e-fffd-411a-b368-b44263f61deb/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",

"location": "eastus",

"macAddress": "00-0D-3A-1B-7B-A9",

"powerState": "VM running",

"privateIpAddress": "10.0.0.4",

"publicIpAddress": "50.171.268.227",

"resourceGroup": "myResourceGroup",

"zones": ""

}

# Add and Prepare Disk for ASM

Storage is separate from the main VM and will need to be allocated. How much disk would you like to add and what type?

**Types of disk:**

Premium LRS: Premium\_LRS

Standard SSD: StandardSSD\_LRS

Standard Disk: Standard\_LRS

Ultra SSD: UltraSSD\_LRS

Depending on the Size, IOPs and throughput that is required for your Oracle VM, see specific documentation on options for [Azure managed disk](https://docs.microsoft.com/en-us/cli/azure/vm/disk?view=azure-cli-latest).

For our example, we’ll create three disks-

-ASM

-Data

-Redo

az vm disk attach --resource-group myResourceGroup \

--vm-name myVM \

--caching ReadWrite \

--name ASMdsk1 \

--sku StandardSSD\_LRS \

--size-gb 128 \

--new

az vm disk attach --resource-group myResourceGroup \

--vm-name myVM \

--caching ReadOnly \

--name DataDsk1 \

--sku StandardSSD\_LRS \

--size-gb 1000 \

--new

az vm disk attach --resource-group myResourceGroup \

--vm-name myVM \

--caching ReadWrite \

--name RedoDsk1 \

--sku StandardSSD\_LRS \

--size-gb 264 \

--new

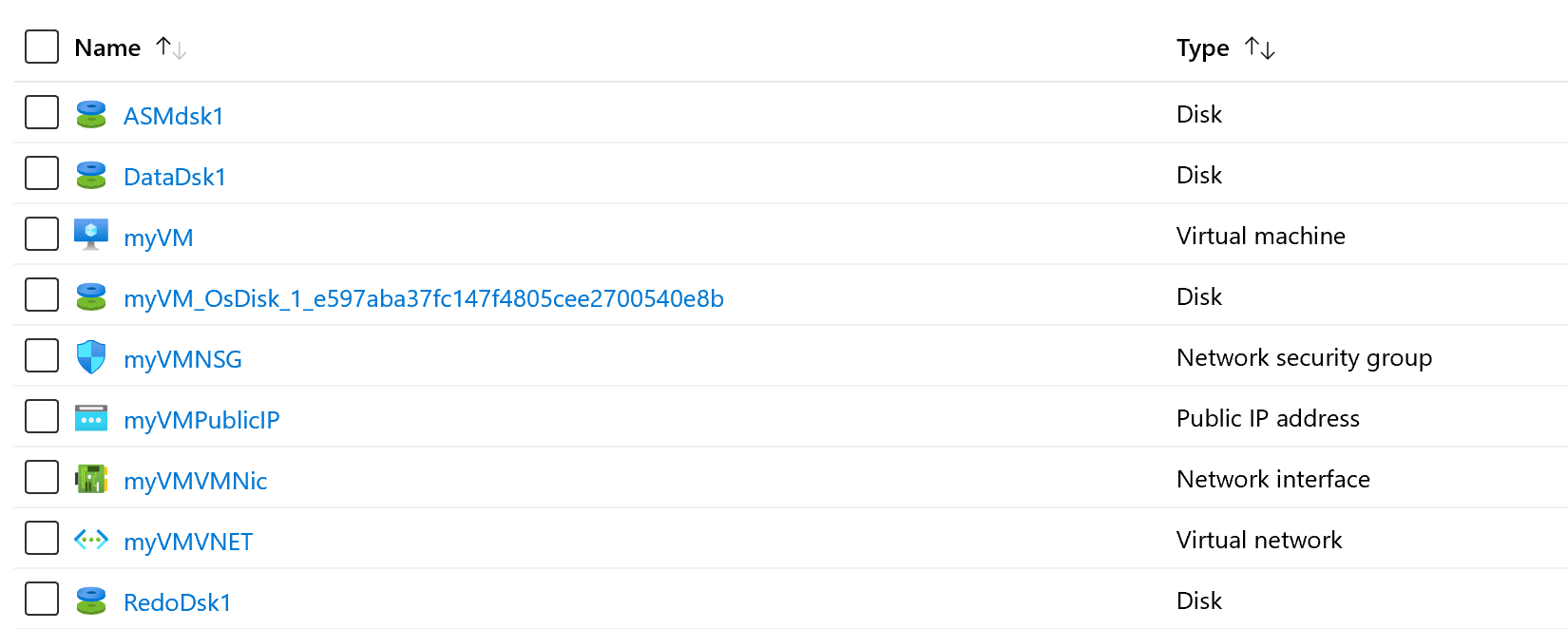
If the database will be using flashback, this should be placed on a separate disk, requiring another disk to be created, using similar setting to the RedoDsk1.

# Best practices on storage for Oracle

* Separate data from redo and place on separate disks, not just partitions. ASM has its own management disk as a requirement, too.
* Turn on [read caching](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/premium-storage-performance#disk-caching) for data disks
* Use Ultra disk only when required and be aware of current limitations for [ultra disk](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/disks-types#ultra-disk) use.
* Don’t use disk sizes over [[1]](#footnote-2)4095G in size, which is the current limit for read cache enablement.
* [Local redundancy](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy), (LRS) offers better performance than spread across multiple regions.
* For more information, see documentation on [Azure VM storage.](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Fazure.microsoft.com%2Fen-us%2Fpricing%2Fdetails%2Fmanaged-disks%2F&data=02%7C01%7CKellyn.Gorman%40microsoft.com%7C95b450b2352e4a30779408d7d141c4cf%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637207952711999508&sdata=oiAO1XOjIzYaxkLyP9%2BvMLwUimc1yFxD05b4SjJMC58%3D&reserved=0)

# Check Resources Deployed

At this point in the installation, when viewed from the Azure Portal, MyResourceGroup should display resources similar to the following for a default Oracle Enterprise Linux installation with the three additional disks:

****

# Connect to the VM

To create an SSH session with the VM, use the following command. Replace the IP address with the publicIpAddress value for your VM.

ssh azureuser@<publicIpAddress>

SSH Sessions can be initiated via Azure Cloud Shell, Putty or another terminal emulator. If connecting from another workstation or PC with a different client IP address, a firewall rule may need to be created for access to the resource.

# Set up the Environment

To set up the Oracle database environment, we need to prepare for ASM, grid control and database software installation.

First step is to log into the VM using SSH from Azure Cloud Shell and become Root user:

ssh azureuser@<ip address>

sudo su -

Create the Oracle, Grid Control users and secondary groups:

useradd -u 501 oracle

groupadd -g 54321 oinstall

groupadd -g 54322 dba

groupadd -g 54329 asmadmin

groupadd -g 54327 asmdba

groupadd -g 54328 asmoper

useradd -u 3000 -g oinstall -G dba,asmadmin,asmdba,asmoper grid

usermod -g oinstall -G dba,asmdba,asmadmin oracle

Note: Passwords will be set for the users created at a later point in the instructions. No need to create them now.

Ensure, even though you requested the latest image, all latest packages you may need and utilities that are essential to the successful installation are updated and the installation of wget, zip and unzip are completed:

yum update -y

yum install wget zip unzip -y

If you receive a message stating “Nothing to do”, simply disregard, it means the image being used already has the latest updates.

# Configure Added Disk

\*\*If you’re going to use ASM, the following step isn’t required. If you are using a volume manager or storage management solution other than Oracle Automatic Storage Manager, (ASM) then the following will be required to set up the disk for use by the host. If you’re using ASM, proceed to the Installing and Configuring ASM on an Oracle VM documentation.

## File System Management

If you added a disk and plan on using the local file system for datafiles vs. ASM, then you will need to prepare it to be used by ASM/Linux Host. To do this, first check the partitions on the server:

cat /proc/partitions

In our example VM host, we’ve created another 1TB disk, sdc, that we can use for data:

11 0 628 sr0

8 16 14680064 sdb

8 17 14678976 sdb1

8 0 31457280 sda

8 1 512000 sda1

8 2 30944256 sda2

2 0 4 fd0

8 32 1072693248 sdc

Create the file system from the partition:

mkfs.ext4 -L /u02 /dev/sdc

Add a second, appropriately sized disk using the same steps if redo latency is a concern to separate redo from data. For any database that has extensive transactional or batch processing, placing redo logs on a separate disk is essential to removing latency issues.

Create the directory:

mkdir /u02

Connect the directory name to the volume you’ve created:

mount -t ext4 /dev/sdc /u02

Perform this for each disk that must be mounted to use with the Oracle database. The disk will need to be added to the /etc/fstab file by editing the file as the root user.Modify it to reflect your new filesystem layout. In our example, we will make a backup of the fstab file to retain before editing and then add the appropriate information using an editor of choice:

cp /etc/fstab /etc/fstab\_bk

vi /etc/fstab

Add the following line to ensure our sdc partition is mounted as part of any reboot:

/dev/sdc /mnt/sdc ext4 defaults 0 0

The entry above translates to:

the path for the hard drive

the destination for the mounted drive

the format type

and defaults settings- 0 and 0.

Save the changes and after rebooting and testing the mount is present after the reboot, remove the backup file:

rm /etc/fstab\_bk

Create any directories that will be required for file system datafiles:

mkdir /u02/<directory path>

Perform this command for all directories that are required for data, redo, etc. that Oracle will use. Update permissions so that Oracle and groups that manage Oracle are the only ones that have permission to these directories:

chown oracle:oinstall /u02/<directory path>

chmod 770 /u02/<directory path>

Perform these two commands for any directories and disks that were added to the server that Oracle will use. If you’re using ASM, follow the directions in the Installing and Configuring ASM on an Oracle VM. If you mounted any drives with this section of the instructions, you will need to unmount them, as ASM will fail, (only one file system manager can mount a drive.)

## Installing and Configuring ASM on an Oracle VM

[Install and Configuring Oracle Automated Storage Management](https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/oracle/configure-oracle-asm), (skip this step if you’ve already configured a volume manager at the Linux disk layer.) Either a volume manager or ASM should be used, not both.

For additional support, refer to [Oracle’s latest ASM administration documentation](https://docs.oracle.com/en/database/oracle/oracle-database/18/ostmg/index.html).

# Create the database

To create a certified Oracle environment, Oracle will need to be installed on the Oracle Enterprise Linux image. The software installation should be performed by the Oracle user.

Software can be downloaded from the following site, depending on the version. If using Oracle ASM, ensure you use the same version or lower from the ASM instance.

* [Oracle 12c](https://www.oracle.com/database/technologies/oracle12c-linux-12201-downloads.html)
* [Oracle 18c](https://www.oracle.com/database/technologies/oracle18c-linux-180000-downloads.html)
* [Oracle 19c](https://www.oracle.com/database/technologies/oracle19c-linux-downloads.html)

After you download the .zip files to your client computer, you can use Secure Copy Protocol (SCP) to copy the files to your VM. There are two choices for Oracle Enterprise Linux, (also true for Redhat) of the rpm or a zip file:

scp oracle-database-ee-19c-1.0.x86\_64.rpm <publicIpAddress>:.

SSH back into your Oracle VM in Azure in order to move the .rpm file into the /opt folder. Then, change the owner of the files:

ssh <publicIPAddress>

sudo mv ./<file name>.rpm /u01/app

cd /u01/app

chown -R oracle.oinstall /u01

sudo chown oracle:oinstall oracle-database-ee-19c-1.0.x86\_64.rpm

Switch over to the Oracle user:

sudo su - oracle

If the zip files were chosen, Unzip the files.

sudo yum install unzip

sudo unzip linux.x64\_19300\_grid\_home.zip

Change permission:

sudo chown -R oracle:oinstall /u01/app

Perform the installation of the database software in Silent Mode

Create a database using the following steps:

Switch user to *oracle*, then initialize the listener for logging using the Listener Control utility (lsnrctl):

sudo su - oracle

lsnrctl start

If you receive an error that the lsnrctl isn’t found, it may be due to Linux being unable to locate the Oracle home files. You can resolve this by:

1. Running the command from the Oracle Home directory
2. Creating a variable and adding to the $PATH

export ORACLE\_HOME=/u01/app/oracle/product/<version>/dbhome\_1

cd $ORACLE\_HOME/bin/lsnrctl

The output is similar to the following:

Copyright (c) 1991, 2014, Oracle. All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome\_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Log messages written to /u01/app/oracle/diag/tnslsnr/myVM/listener/alert/log.xml

Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=myVM.twltkue3xvsujaz1bvlrhfuiwf.dx.internal.cloudapp.net)(PORT=1521)))

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))

STATUS of the LISTENER

------------------------

Alias LISTENER

Version TNSLSNR for Linux: Version 12.1.0.2.0 - Production

Start Date 23-MAR-2017 15:32:08

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/myVM/listener/alert/log.xml

Listening Endpoints Summary...

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=myVM.twltkue3xvsujaz1bvlrhfuiwf.dx.internal.cloudapp.net)(PORT=1521)))

The listener supports no services

The command completed successfully

Create an database using the Oracle Database Assistant, (DBCA). The DBCA includes a command line tool to create a database without the need of a user interface. The following is an example of a common dbca creation script:

dbca -silent \

-createDatabase \

-templateName General\_Purpose.dbc \

-gdbname cdb1 \

-sid cdb1 \

-responseFile NO\_VALUE \ #RF location for advanced config

-characterSet AL32UTF8 \

-sysPassword OraPasswd1 \ # set password wanted

-systemPassword OraPasswd1 \ # set password wanted

-createAsContainerDatabase true \

-numberOfPDBs 1 \ #can be more if the needed

-pdbName pdb1 \

-pdbAdminPassword OraPasswd1 \ # set password wanted

-databaseType MULTIPURPOSE \ #default

-automaticMemoryManagement false \

-datafileDestination /u01/data \ # or ASM

-filesystem </u01/directory for datafiles> \ #Remove if ASM

-ignorePreReqs

*Note: It can take around 20+ minutes to create the database.*

## Set Oracle variables

Before you connect, you need to set two environment variables: *ORACLE\_HOME* and *ORACLE\_SID*.

export ORACLE\_HOME=/u01/app/oracle/product/12.1.0/dbhome\_1

export ORACLE\_SID=cdb1

You also can add ORACLE\_HOME and ORACLE\_SID variables to the .bashrc file. This would save the environment variables for future sign-ins. Confirm the following statements have been added to the ~/.bashrc file using editor of your choice.

# Add ORACLE\_HOME.

export ORACLE\_HOME=/u01/app/oracle/product/18.3.0/dbhome\_1

# Add ORACLE\_SID.

export ORACLE\_SID=cdb1

# Oracle EM Express connectivity

For a GUI management tool that you can use to explore the database, set up Oracle EM Express. For multiple databases and need for a singular management tool, consider deploying a separate VM in Azure and installing <Oracle Cloud Control>.

To connect to Oracle EM Express, you must first set up the port in Oracle.

Connect to your database using sqlplus:

sqlplus / as sysdba

Once connected, set the port 5502 for EM Express

exec DBMS\_XDB\_CONFIG.SETHTTPSPORT(5502);

Open the container PDB1 if not already opened, but first check the status:

select con\_id, name, open\_mode from v$pdbs;

The output is similar to the following:

CON\_ID NAME OPEN\_MODE

----------- ------------------------- ----------

2 PDB$SEED READ ONLY

3 PDB1 MOUNT

If the OPEN\_MODE for PDB1 is not READ WRITE, then run the followings commands to open PDB1:

alter session set container=pdb1;

alter database open;

You need to type quit to end the sqlplus session and type exit to logout of the oracle user.

# 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.

Sign on as root

sudo su -

Using your favorite editor, edit the file /etc/oratab and change the default N to Y:

cdb1:/u01/app/oracle/product/12.1.0/dbhome\_1:Y

Create a file named /etc/init.d/dbora and paste the following contents:

#!/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/12.1.0/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

Change permissions on files with *chmod* as follows:

chgrp dba /etc/init.d/dbora

chmod 750 /etc/init.d/dbora

Create symbolic links for startup and shutdown as follows:

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

To test your changes, restart the VM:

reboot

Verify the database restarted with the reboot of the VM as part of the initialization steps:

ps -ef | grep pmon

In the results should be seen the pmon, (process monitor) for the databases running on the VM that were placed in the dbora file.

# Open ports for connectivity

The final task is to configure some external endpoints. To set up the Azure Network Security Group that protects the VM, first exit your SSH session in the VM (should have been kicked out of SSH when rebooting in previous step).

To open the endpoint that you use to access the Oracle database remotely, create a Network Security Group rule with [az network nsg rule create](https://docs.microsoft.com/en-us/cli/azure/network/nsg/rule) as follows:

az network nsg rule create \

--resource-group myResourceGroup\

--nsg-name myVmNSG \

--name allow-oracle \

--protocol tcp \

--priority 1001 \

--destination-port-range 1521

To open the endpoint that you use to access Oracle EM Express remotely, create a Network Security Group rule with [az network nsg rule create](https://docs.microsoft.com/en-us/cli/azure/network/nsg/rule) as follows:

az network nsg rule create \

--resource-group myResourceGroup \

--nsg-name myVmNSG \

--name allow-oracle-EM \

--protocol tcp \

--priority 1002 \

--destination-port-range 5502

If needed, obtain the public IP address of your VM again with [az network public-ip show](https://docs.microsoft.com/en-us/cli/azure/network/public-ip) as follows:

az network public-ip show \

--resource-group myResourceGroup \

--name myVMPublicIP \

--query [ipAddress] \

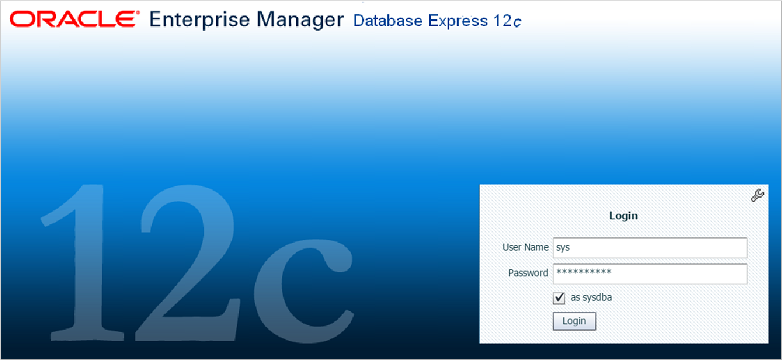
--output tsv

Connect EM Express from your browser. Make sure your browser is compatible with EM Express (Flash install is required):

https://<VM ip address or hostname>:5502/em

You can log in by using the **SYSTEM** account, and check the **as sysdba** checkbox. Use the password for the SYSTEM user for this step.

Replace\*\*\*\*\*\*\*\*\*\*\*we don’t use SYS!!



# Next steps

Learn about other [Oracle solutions on Azure](https://docs.microsoft.com/en-us/azure/virtual-machines/workloads/oracle/oracle-considerations).

# Secure your Linux host and Oracle database.

<To be added>

# Monitoring and Managing Oracle databases via Oracle Cloud Control

<to be added>

# Creating a Disaster Recovery Solution for Oracle on Azure

<to be added>

## Creating an Azure VM image template from your Oracle VM

Creating templates offers the opportunity to build once and reuse multiple times, as well as automate deployments in Azure.

[Working with ARM Templates](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-tutorial-create-first-template?tabs=azure-powershell) will provide guidance on how to work with the JSON files involved with ARM templates to be used with Terraform and other automation scripting tools.

Use an ARM Template that’s been [exported from an existing resource group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-tutorial-export-template?tabs=azure-powershell).

To create an Image from an existing VM already created, use the [following instructions](https://docs.microsoft.com/en-us/azure/virtual-machines/linux/capture-image) to create VM images to reuse.

# Cleaning Up Resources in Azure

Unlike an on-prem solution, cloud resources are simpler to remove when no longer necessary. Outside of dependent pipeline or snapshots, a resource or resource group can simple be removed once no longer necessary through the Azure portal or from the command line using terminal access like Azure Cloud Shell.

## Removing a Resource Group from Azure

Removing an Oracle database on an Azure VM, along with the supporting resources in the same resource group are no longer needed, you can use the [az group delete](https://docs.microsoft.com/en-us/cli/azure/group) command to remove all related resources along with the resource group. If a single resource needs to be removed, you can do this from the Azure portal or with the az resource delete command.

To remove a resource group via the Azure Command Shell:

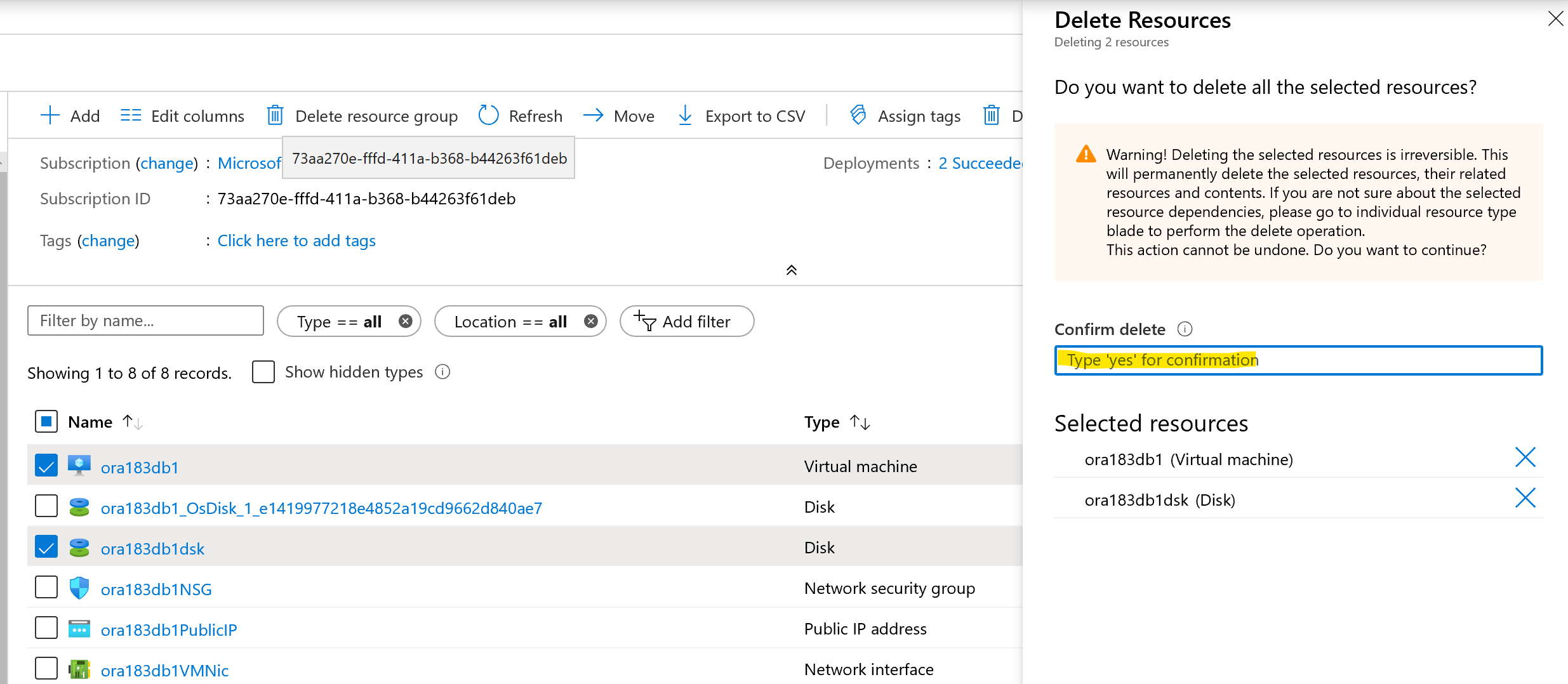
az group delete --name <Resource Group>

This command will remove ALL resources in the MyResourceGroup and the group that you created in your first step. If there is a desire to retain the resource group, but simple delete the VM, then this can be accomplished by executing the following command:

az resource delete -g <Resource Group> -n <resource name>

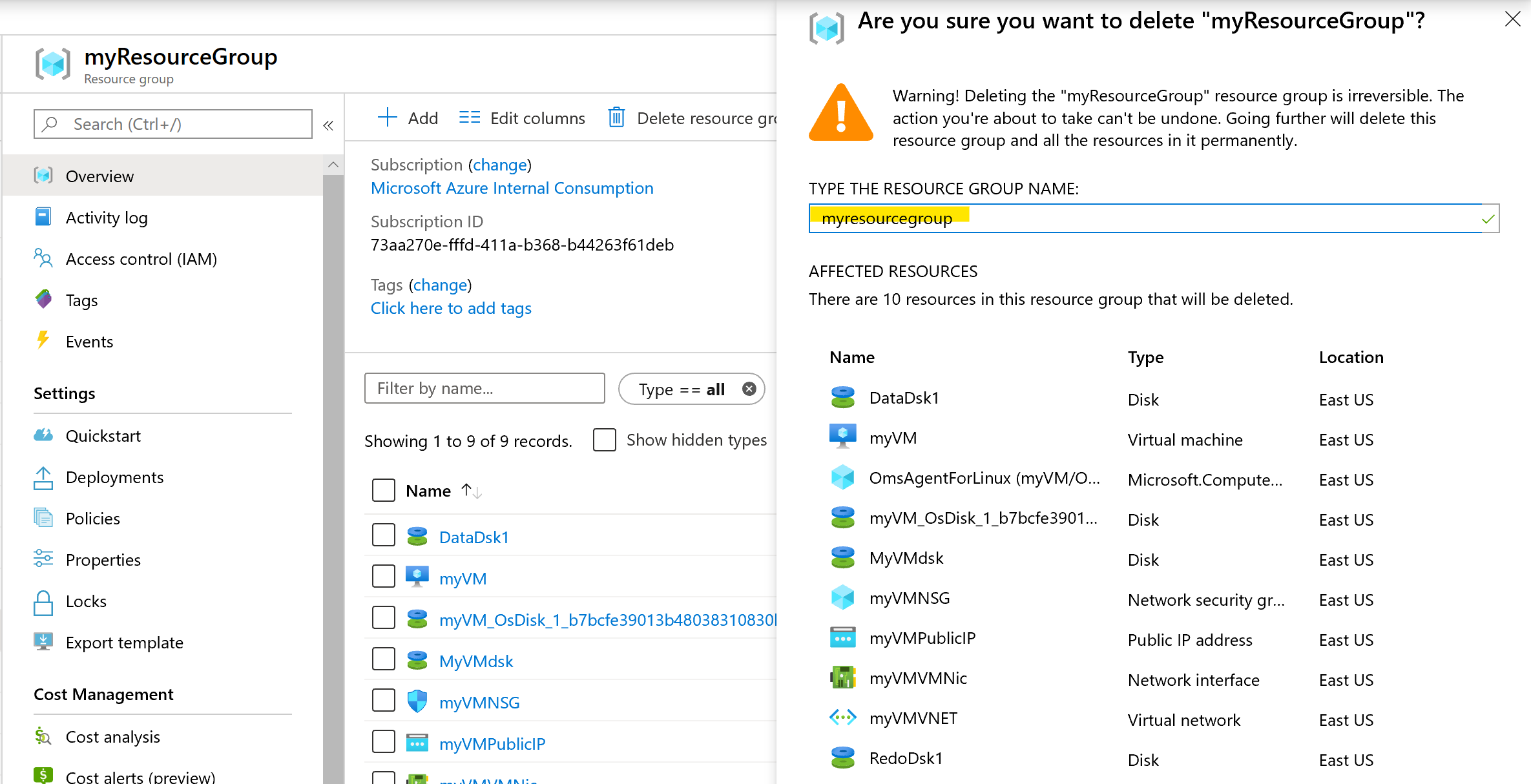
The Azure Portal can be used to remove any resource or resource group, too.

Log into the Azure Portal and go to Resource Groups and find the Resource Group you wish to work with. To remove individual resources from the group, check mark the ones to be removed and click Delete from the upper ribbon:

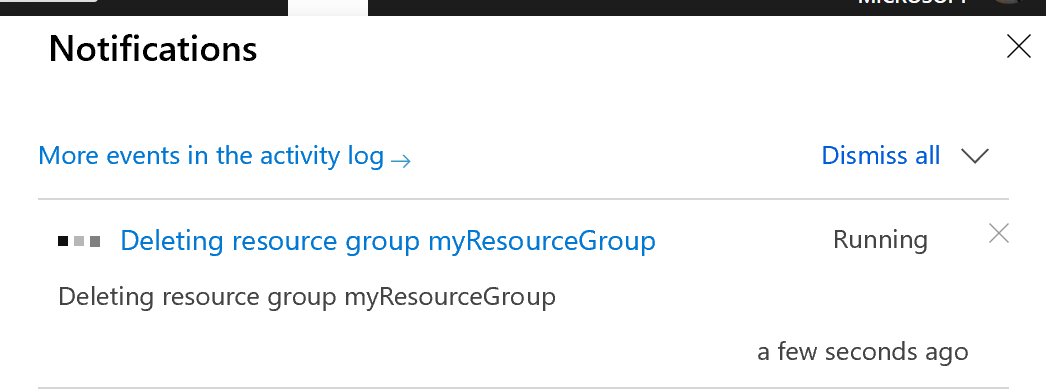


A pane will slide into view asking you to confirm the deletion by typing in “yes” and once you type in your response, click the Delete button at the bottom to complete the request. Depending on the resource, the process to delete the resource(s) will occur and status can be monitored at the top right of the Azure Portal.

To remove an entire resource group in the Azure Portal, go to Resource Groups and choose the resource group you wish to remove. Click on the Delete button from the ribbon at top and confirm by typing in the name of the resource group just as it is displayed and then click the Delete button at the bottom right.



As with a single resource deletion, depending on the amount of resources and type of resources, time to remove from the portal will vary and the status will be displayed at the top right of the portal.



1. This may not be an option for very large databases, those over 4TB. [↑](#footnote-ref-2)