List Available VM Sizes and SKUs

In this lab, you will learn about Azure Virtual Machines (VMs) and list available VM types/sizes in a region.

Azure Virtual Machine (VMs) are Virtual Machines hosted in the Azure cloud. You can run any workload that you normally run on a VM, on these Virtual Machines too. Both Linux and Windows are supported.

Log into the Azure CLI
Understand Azure Virtual Machines (VMs)
List available VM sizes (SKUs) for an Azure region

Understanding Azure Virtual Machines
So what is an Azure Virtual Machine (VM)?

Azure Virtual Machine (VMs) are Azure-hosted Virtual Machines. You can run any workload that you normally run on a VM on these Virtual Machines too. Both Linux and Windows operating systems are supported.

VMs are great if you need to run infrastructure-as-a-service (IaaS) workloads in Azure. When using IaaS, you are responsible to manage your solutions VM, operating system, networking, patching, and more. Microsoft recommends using platform-as-a-service (PaaS) services whenever you can. However, VMs have their use cases, such as when you need to have full control on your platform or when you have a legacy application that you are going to move to Azure.

Before working with the VMs, let's talk about VM sizes.

Azure Virtual Machines, similar to any other computer, have limited resources. They have fixed amount of memory, CPU, and disk space. The VM can have a magnetic HDD disk or a faster, solid-state-drive, SSD. From the processing point of view, a VM can have a single-core CPU or a multi-core CPU. In Azure, you can choose how "big" you need your VM to be by choosing the desired VM size (or SKU) at the time of provisioning. Not all sizes are available in all Azure regions.

Note: SKU stands for Stock Keeping Unit, representing what resources (in our case VMs) are available in Azure for sale.

In the next step, we will run a CLI command, which shows us which VM sizes we have available to us in the East US region.

List All Available VMs in Your Region
Let's see which VM sizes are available in the East US (eastus) region.

Use the following command to get the list of VM sizes (SKUs).

az vm list-sizes --location eastus

In the command output, you will get a list of JSON items. Each JSON item gives you the following information:

name: The VM SKU name. You need to use this name when creating the resource; for example, Standard_ND96amsr_A100_v4.
maxDataDiskCount: Maximum number of disks which can be attached to this VM SKU; for example, 16.
memoryInMb: The memory (RAM) for this VM type in megabytes; for example, 1,970,176.
numberOfCores: Number of CPU cores; for example, 96.
osDiskSizeInMb: The size of the OS (primary) disk for this VM type; for example, 1,047,552.
resourceDiskSizeInMb: The size of the default data disk attached to this VM type; for example, 2,969,600.

You can also get a detailed list of VM SKUs, by running the following command:

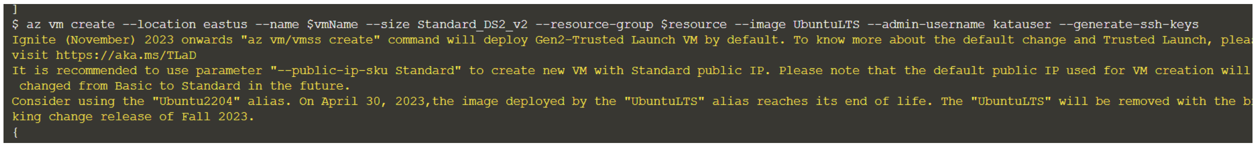
az vm list-skus --location eastus

Provision a New Azure Virtual Machine

Provision a New Azure Virtual Machine
In this lab, we're going to provision a new Azure VM in the East US region.

Use the following CLI command to deploy this VM:

az vm create --location eastus --name $vmName --size Standard_DS2_v2 --resource-group $resource --image UbuntuLTS --admin-username katauser --generate-ssh-keys



```
]
$ az vm create --location eastus --name $vmName --size Standard_DS2_v2 --resource-group $resource --image UbuntuLTS --admin-username katauser --generate-ssh-keys
Ignite (November) 2023 onwards "az vm/vmss create" command will deploy Gen2-Trusted Launch VM by default. To know more about the default change and Trusted Launch, plea
visit https://aka.ms/TLaD
It is recommended to use parameter "--public-ip-sku Standard" to create new VM with Standard public IP. Please note that the default public IP used for VM creation will
 changed from Basic to Standard in the future.
Consider using the "Ubuntu2204" alias. On April 30, 2023,the image deployed by the "UbuntuLTS" alias reaches its end of life. The "UbuntuLTS" will be removed with the b
king change release of Fall 2023.
{
```

Let's take a look at the command parameters:

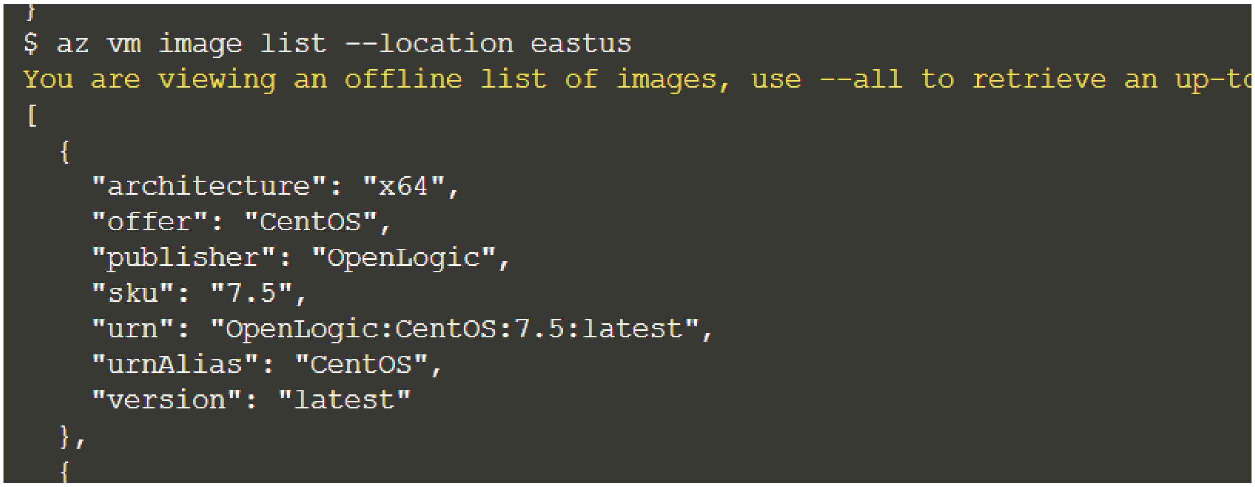
--location: The new VM location/region.
--name: The name for the new VM.
--resource-group: The resource group to create the VM in.
--size: The new VM size.

--image: The image name to be installed on the VM. This determines which OS (operating system) the VM will run.
--admin-username: The VM administrator username. This along with with VM password, or SSH key will be used to log into the new VM.
--generate-ssh-keys: You need an SSH key to log into a Linux machine. By passing this parameter, we instruct CLI ro generate this key for us.

Run the following command to get a list of available VM images. Images determine which OS (operating system) will be installed on your new VM.

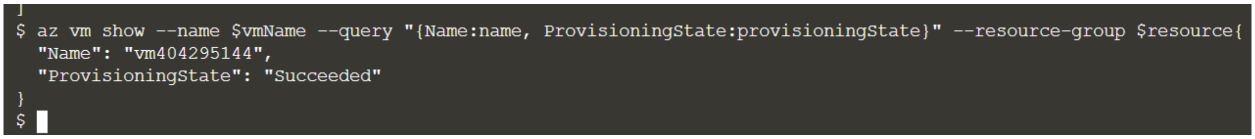
az vm image list --location eastus

In the next step, we will check our new VM both in CLI and the Azure portal.

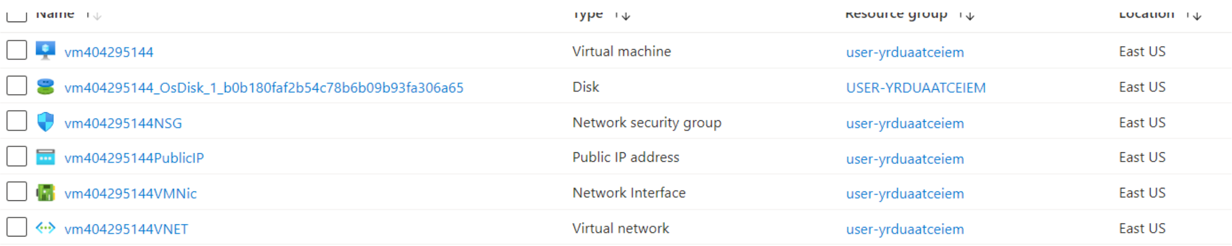


```
}
$ az vm image list --location eastus
You are viewing an offline list of images, use --all to retrieve an up-to
[
  {
    "architecture": "x64",
    "offer": "CentOS",
    "publisher": "OpenLogic",
    "sku": "7.5",
    "urn": "OpenLogic:CentOS:7.5:latest",
    "urnAlias": "CentOS",
    "version": "latest"
  },
  {
```

Check our Azure Virtual Machines in the Azure portal
First, let's use the Azure CLI to confirm that our new VM has been provisioned:



```
]
$ az vm show --name $vmName --query "{Name:name, ProvisioningState:provisioningState}" --resource-group $resource{
  "Name": "vm404295144",
  "ProvisioningState": "Succeeded"
}
$
```



□ Name	Type	Resource group	Location
□ vm404295144	Virtual machine	user-yrduaatceiem	East US
□ vm404295144_OsDisk_1_b0b180faf2b54c78b6b09b93fa306a65	Disk	USER-YRDUAATCEIEM	East US
□ vm404295144NSG	Network security group	user-yrduaatceiem	East US
□ vm404295144PublicIP	Public IP address	user-yrduaatceiem	East US
□ vm404295144VMNic	Network Interface	user-yrduaatceiem	East US
□ vm404295144VNET	Virtual network	user-yrduaatceiem	East US

Use the following command to clean up the new VM from your allocated resource group:

az vm delete --resource-group $resource --name $vmName

Use the following command to list all VMs in your allocated resource group:

az vm list --resource-group $resource

Stop and Start an Azure Virtual Machine

In this lab, you will stop, and start an Azure Virtual Machine using the CLI.

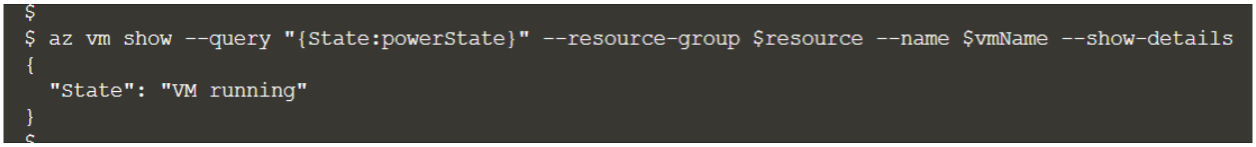
Stop an Azure Virtual Machine
Start an Azure Virtual Machine

When started, Azure VMs will run 24x7. This means you will pay for them all the time. This is acceptable for a production system, but you might be able to save some cost with your development and staging VMs by simply stopping them when when they're not being used. For instance, you might want to stop your development VMs at 5:00 PM and restart them at 9:00 AM the following morning.

With Azure VMs, you pay for both Azure Virtual Machine storage (disks) and processing. When stopped, you only pay for storage, saving money on processing.

First, let's confirm our VM exists, and is indeed running:

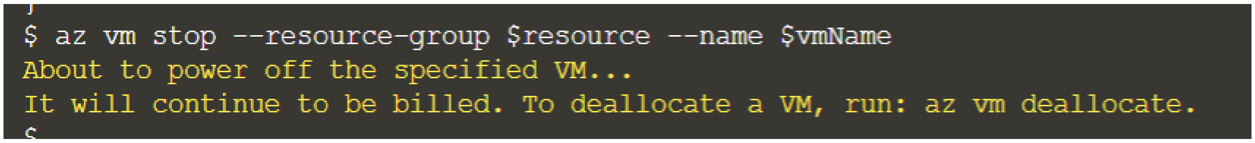
az vm show --query "{State:powerState}" --resource-group $resource --name $vmName --show-details



```
$
$ az vm show --query "{State:powerState}" --resource-group $resource --name $vmName --show-details
{
  "State": "VM running"
}
```

Now, let's stop the VM by running the following CLI command:

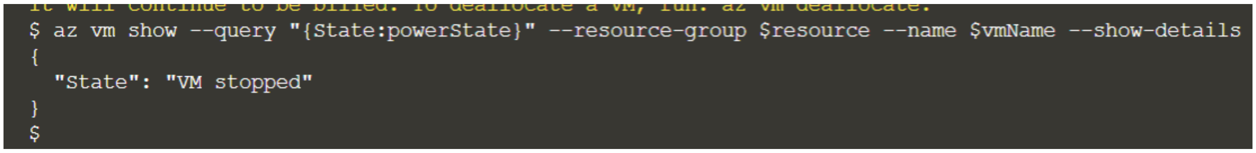
az vm stop --resource-group $resource --name $vmName



```
$ az vm stop --resource-group $resource --name $vmName
About to power off the specified VM...
It will continue to be billed. To deallocate a VM, run: az vm deallocate.
```

Now, let's confirm our VM state again by running this command:

az vm show --query "{State:powerState}" --resource-group $resource --name $vmName --show-details



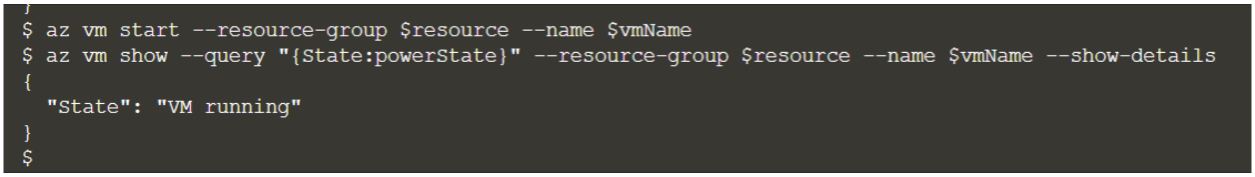
```
It will continue to be billed. To deallocate a VM, run: az vm deallocate.
$ az vm show --query "{State:powerState}" --resource-group $resource --name $vmName --show-details
{
  "State": "VM stopped"
}
$
```

Start an Azure Virtual Machine
Time to start our VM again by running the following CLI command:

az vm start --resource-group $resource --name $vmName

Now, let's confirm our VM state again by running this command:

az vm show --query "{State:powerState}" --resource-group $resource --name $vmName --show-details



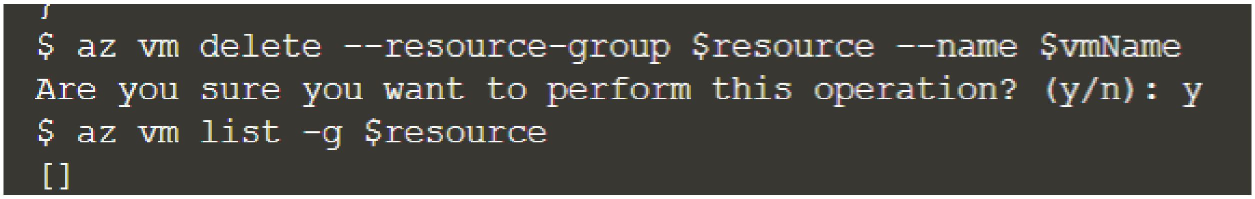
```
}
$ az vm start --resource-group $resource --name $vmName
$ az vm show --query "{State:powerState}" --resource-group $resource --name $vmName --show-details
{
  "State": "VM running"
}
$
```

Clean Up
Use the following command to clean up the new VM from your allocated resource group:

az vm delete --resource-group $resource --name $vmName

Use the following command to list all VMs in your allocated resource group:

az vm list -g $resource



```
 }
$ az vm delete --resource-group $resource --name $vmName
Are you sure you want to perform this operation? (y/n): y
$ az vm list -g $resource
[]
```

Manage Auto-Shutdown for the Azure Virtual Machine

In this lab, you will configure auto-shutdown for an existing Azure Virtual Machine, using the CLI.

Log into the Azure CLI

Enable auto-shutdown for an Azure Virtual Machine
Disable auto-shutdown for an Azure Virtual Machine
Clean up

Use the following credentials to log into the Azure CLI:

az login -u $username -p $password

We preconfigured this lab so that a new Azure Virtual Machine is created for you.

In the next step, we will configure auto-shutdown for an Azure VM, so at a specific time, your VM will shutdown. This time could be the end of a working day at 5:00 PM.

Enabling Auto-shutdown for an Azure Virtual Machine
When started, Azure VMs will run 24x7. This means you will pay for them all the time. This is acceptable for a production system, but you might be able to save some cost with your development and staging VMs by simply stopping them when when they're not being used. For instance, you might want to stop your development VMs at 5:00 PM.

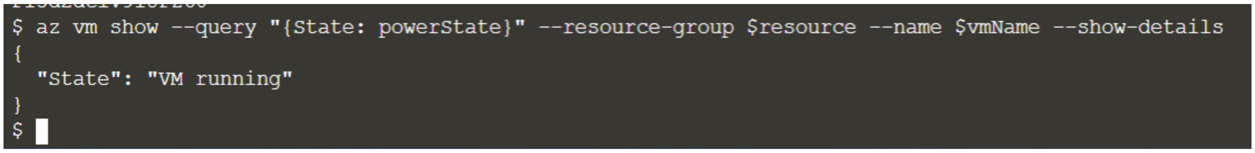
With Azure VMs, you pay for both Azure Virtual Machine storage (disks) and processing. When stopped, you only pay for storage, saving money on processing.

In the previous lab, we saw how you can save money by stopping a nonproduction Azure VM. We used the Azure CLI to manually stop and start an Azure VM.

The shutting down process can be automated using Azure portal or using tools such as the Azure CLI.

First, let's confirm our VM exists and is indeed running:

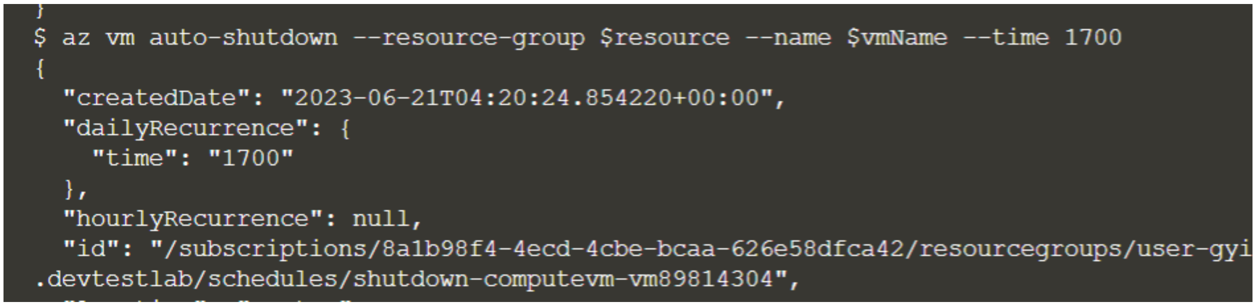
az vm show --query "{State: powerState}" --resource-group $resource --name $vmName --show-details



```
$ az vm show --query "{State: powerState}" --resource-group $resource --name $vmName --show-details
{
  "State": "VM running"
}
$ 
```

Now, let's configure our VM to auto-shutdown everyday at 5:00 PM:

az vm auto-shutdown --resource-group $resource --name $vmName --time 1700



```
}
$ az vm auto-shutdown --resource-group $resource --name $vmName --time 1700
{
  "createdDate": "2023-06-21T04:20:24.854220+00:00",
  "dailyRecurrence": {
    "time": "1700"
  },
  "hourlyRecurrence": null,
  "id": "/subscriptions/8a1b98f4-4ecd-4cbe-bcaa-626e58dfca42/resourcegroups/user-gyi
.devtestlab/schedules/shutdown-computevm-vm89814304",
```

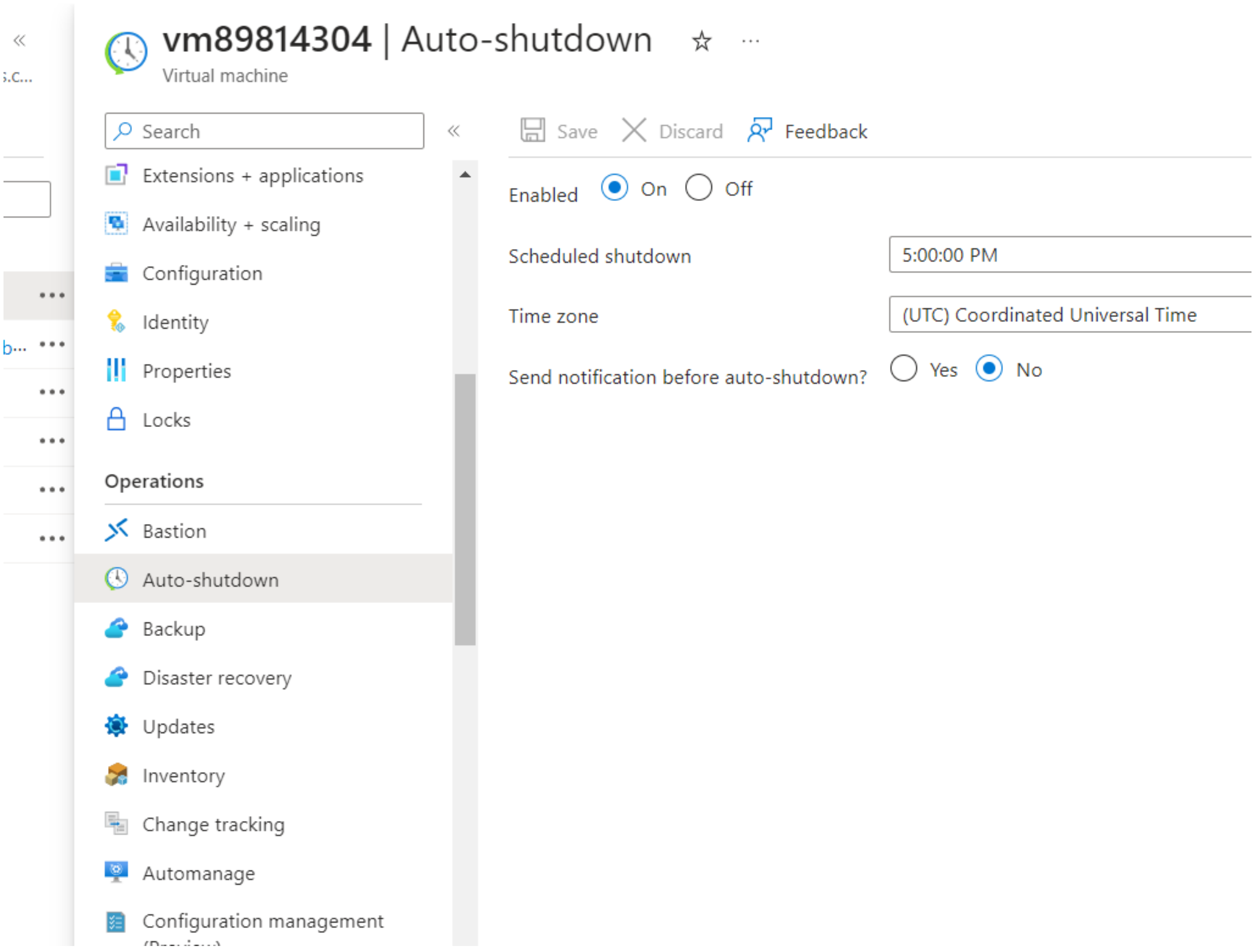
Let's take a look at the command parameters:

--time: The auto-shutdown time in UTC.
--name: The name of the VM.
--resource-group: The parent Resource Group of the VM.

You can also optionally pass the following parameters to the CLI command:

--email: The email recipient to send shutdown notifications to.
--webhook: A webhook URL (HTTPS) to call when the VM is shutting down. This can be used to call an HTTP API so you can implement custom shutdown notifications.

In the next step, we will confirm the auto-shutdown settings in the Azure portal.



vm89814304 | Auto-shutdown

Virtual machine

Enabled ● On ○ Off

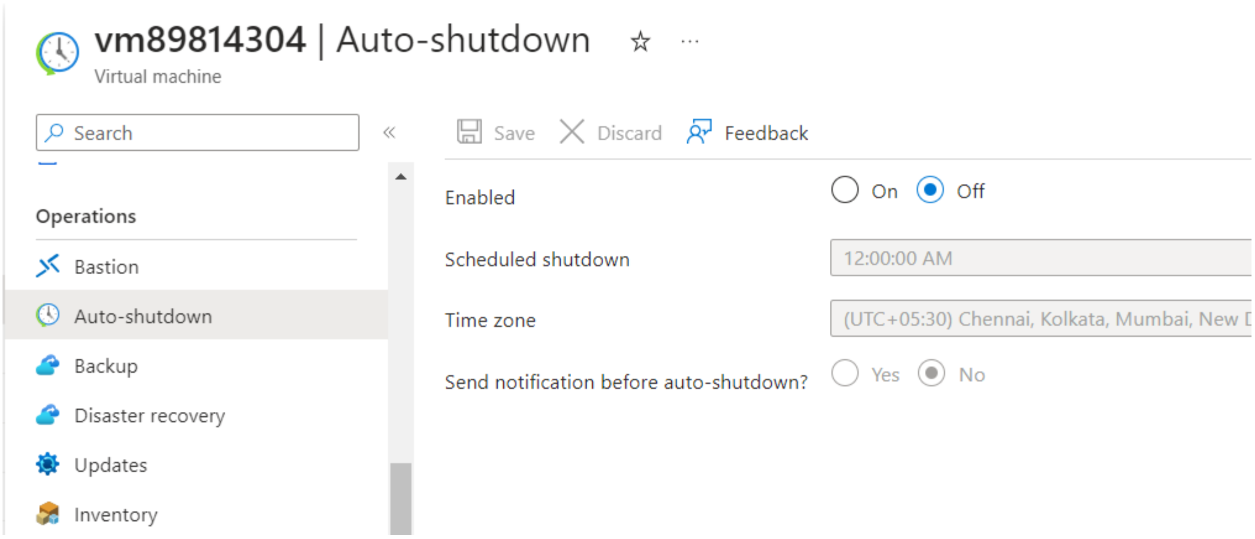
Scheduled shutdown: 5:00:00 PM

Time zone: (UTC) Coordinated Universal Time

Send notification before auto-shutdown? ○ Yes ● No

Disabling Auto-shutdown for an Azure Virtual Machine
Now, let's disable auto-shutdown for our VM using the Azure CLI:

```
az vm auto-shutdown --resource-group $resource --name $vmName --off
```



vm89814304 | Auto-shutdown ☆ ···

Virtual machine

Enabled ○ On ● Off

Scheduled shutdown 12:00:00 AM

Time zone (UTC+05:30) Chennai, Kolkata, Mumbai, New D

Send notification before auto-shutdown? ○ Yes ● No

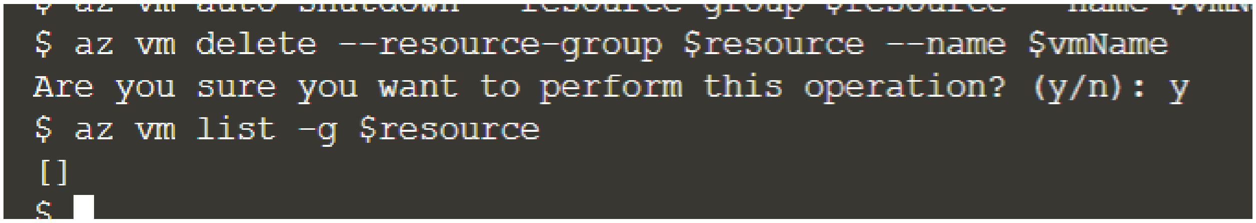
Clean Up

Use the following command to clean up the new VM from your allocated resource group:

az vm delete --resource-group $resource --name $vmName

Use the following command to list all VMs in your allocated resource group:

az vm list -g $resource



```
$ az vm auto-shutdown --resource-group $resource --name $vmN
$ az vm delete --resource-group $resource --name $vmName
Are you sure you want to perform this operation? (y/n): y
$ az vm list -g $resource
[]
$ 
```

Manage Open Ports on the Azure Virtual Machine

Understand Azure VM networking and communication ports
Configure open ports for an existing VM
Check VM port settings in the Azure portal

In this lab, you will configure communication ports for an existing Azure Virtual Machine using the CLI.

In the next step, we will talk about Azure Virtual Machine communication via ports and how to open and block them.

Understanding Azure VM Port Communication

We now have a running Azure Virtual Machine. The next step is to connect to this Virtual Machine so we can install and configure our custom workloads. Generally speaking, you will connect to a Windows Virtual Machine via RDP port 3389. In case of Linux Azure VMs, you will use the SSH port 22.

When provisioning an Azure VM, all the communication ports are blocked for security reasons. You need to specifically open or allow the ports you need. This can be done both in the Azure portal as well as by using tools such as the Azure CLI. Other ports might need to be allowed based on the functionality of your VM. For instance, if your VM is a web server, you need to allow both communication ports 80 (HTTP), and 443 (HTTPS/SSL).

each VM is always a port of an Azure Virtual Network (VNet) and can have two IP addresses. A public IP address, which is routable from the public Internet, and a private IP address, which can be accessed from other resources in the same Azure Virtual Network.

Important: Allowing RDP (3389), and SSH (22) ports are not recommended for a Production VM with a public IP address. It is recommended that you assign a private IP address to a VM and access it using technologies such an VPN or Azure Bastion.

In this introduction, let's go ahead and open ports 80 and 443 for our existing Azure Virtual Machine so it can be used as a web server.

Configuring Open Ports for an Azure VM

Let's prepare our Azure VM for a web server role. Web traffic uses ports 80 and 443, so we need to open them for this VM.

Use the following command to allow both web ports 80 and 443 for your VM:

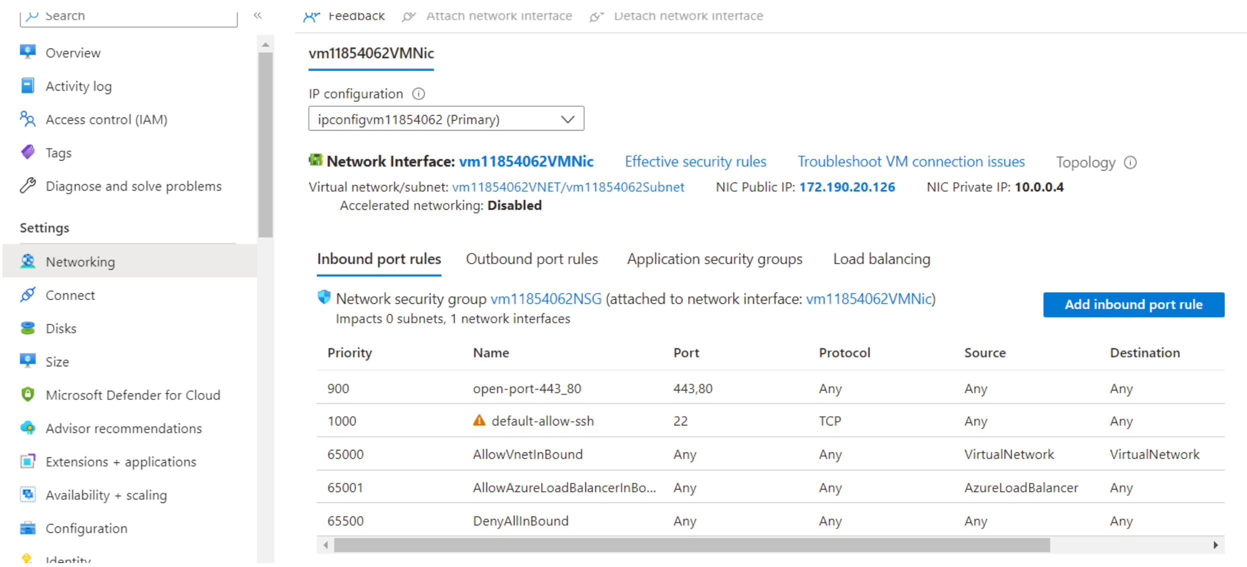
```
az vm open-port --resource-group $resource --name $vmName --port 443,80
```

simply check that they are open in the Azure portal.

Let's take a look at the command parameters:

--name: The name of the VM.
--resource-group: The parent Resource Group of the VM.
--port: The inbound port or port range (e.g. 100-200) to open. You can use * to allow/open incoming traffic to all ports. Use comma-separated values to specify more than one port (e.g., 80,443).

In the next step, we will check (confirm) the port settings for this VM in the Azure portal.



IP configuration: ipconfigvm11854062 (Primary)

Network Interface: vm11854062VMNic

Virtual network/subnet: vm11854062VNET/vm11854062Subnet NIC Public IP: 172.190.20.126 NIC Private IP: 10.0.0.4

Accelerated networking: Disabled

Network security group vm11854062NSG (attached to network interface: vm11854062VMNic)
Impacts 0 subnets, 1 network interfaces

Priority	Name	Port	Protocol	Source	Destination
900	open-port-443_80	443,80	Any	Any	Any
1000	⚠ default-allow-ssh	22	TCP	Any	Any
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork
65001	AllowAzureLoadBalancerInBo...	Any	Any	AzureLoadBalancer	Any
65500	DenyAllInBound	Any	Any	Any	Any

Configure Azure VM Access to Azure Storage Account

In this lab, you will grant storage account access for an Azure Virtual Machine using the CLI.

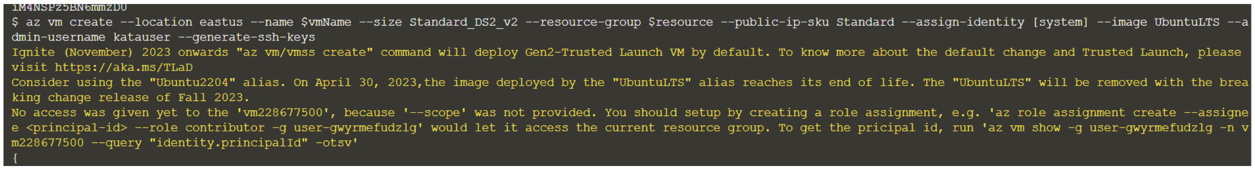
Azure storage accounts offer multiple authentication options. You can use storage account keys or SAS tokens but the issue with these options is their security. There is always a chance that a SAS token or storage key is accidentally hardcoded and checked into a source control repository. The second option is to use the combination of Azure Active Directory authentication and RBAC roles to grant access to the storage account.

Log into the Azure CLI
Understand Azure VM identity and RBAC access
Configure storage account access for Azure VM
Check storage account accesses
Clean up

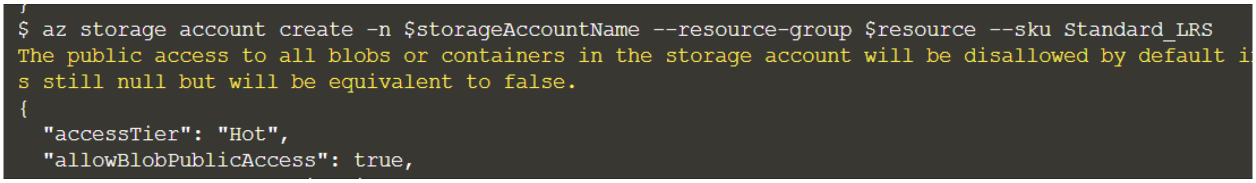
To create a new Azure VM with a system-assigned identity: az vm create --location eastus --name $vmName --size Standard_DS2_v2 --resource-group $resource --public-ip-sku Standard --assign-identity [system] --image UbuntuLTS --admin-username katauser --generate-ssh-keys

To create a new Azure storage account: az storage account create -n $storageAccountName --resource-group $resource --sku Standard_LRS

In the next step, we will give storage account access to the Azure Virtual Machine, so any application running on this VM can talk to the storage account without needing any storage key or SAS token.



```
1M4NSPz5BN6mmzD0
$ az vm create --location eastus --name $vmName --size Standard_DS2_v2 --resource-group $resource --public-ip-sku Standard --assign-identity [system] --image UbuntuLTS --a
dmin-username katauser --generate-ssh-keys
Ignite (November) 2023 onwards "az vm/vmss create" command will deploy Gen2-Trusted Launch VM by default. To know more about the default change and Trusted Launch, please
visit https://aka.ms/TLaD
Consider using the "Ubuntu2204" alias. On April 30, 2023,the image deployed by the "UbuntuLTS" alias reaches its end of life. The "UbuntuLTS" will be removed with the brea
king change release of Fall 2023.
No access was given yet to the 'vm228677500', because '--scope' was not provided. You should setup by creating a role assignment, e.g. 'az role assignment create --assigne
e <principal-id> --role contributor -g user-gwyrmefudzlg' would let it access the current resource group. To get the pricipal id, run 'az vm show -g user-gwyrmefudzlg -n v
m228677500 --query "identity.principalId" -otsv'
{
```



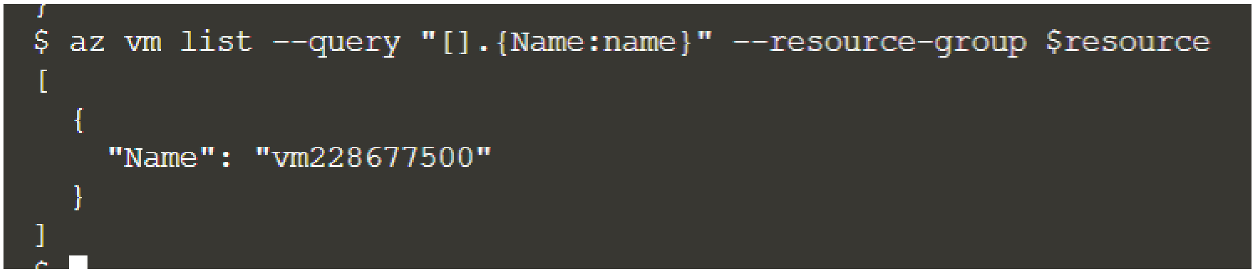
```
}
$ az storage account create -n $storageAccountName --resource-group $resource --sku Standard_LRS
The public access to all blobs or containers in the storage account will be disallowed by default i
s still null but will be equivalent to false.
{
  "accessTier": "Hot",
  "allowBlobPublicAccess": true,
```

Understand Azure VM Identity and RBAC

Azure storage accounts offer multiple authentication options. You can use storage account keys or SAS tokens but the issue with these options is their security. There is always a chance that a SAS token or Storage key is accidentally hardcoded and checked into a source control repository.

There is a better way to authenticate clients to Azure storage accounts, and any other resource type supporting Azure Active Directory authentication. You can create a managed identity for your client resources (Azure Virtual Machine in this lab) and configure the Azure storage account to allow access to this identity using RBAC roles. Both built-in and custom roles can be used.

We preconfigured this lab so that a new Azure Virtual Machine was created for you. Use the following command to confirm the app service exists:



```
}
$ az vm list --query "[].{Name:name}" --resource-group $resource
[
  {
    "Name": "vm228677500"
  }
]
$
```

Let's configure storage account access for this Azure VM in the next step.

Configure Storage Account Access for an Azure VM

Use the following command to store the storage account resource ID in a variable:

STORAGE_RESOURCE_ID=$(az storage account show --name $storageAccountName --resource-group $resource --query id --output tsv)

Use this command to see the value of the previous variable:

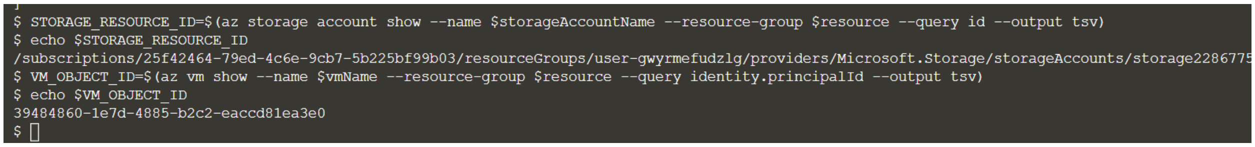
echo $STORAGE_RESOURCE_ID

We also need to grab the identity ID for the Virtual Machine and store it in a variable:

VM_OBJECT_ID=$(az vm show --name $vmName --resource-group $resource --query identity.principalId --output tsv)

You can use this command to see the value of the Azure VM ID:

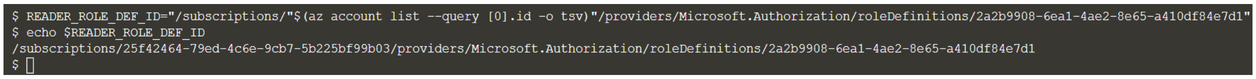
echo $VM_OBJECT_ID



```
$ STORAGE_RESOURCE_ID=$(az storage account show --name $storageAccountName --resource-group $resource --query id --output tsv)
$ echo $STORAGE_RESOURCE_ID
/subscriptions/25f42464-79ed-4c6e-9cb7-5b225bf99b03/resourceGroups/user-gwyrmefudzlg/providers/Microsoft.Storage/storageAccounts/storage2286775
$ VM_OBJECT_ID=$(az vm show --name $vmName --resource-group $resource --query identity.principalId --output tsv)
$ echo $VM_OBJECT_ID
39484860-1e7d-4885-b2c2-eaccd81ea3e0
$ 
```

Now it is time to choose which access level we need to grant to our Virtual Machine. There are several built-in storage account RBAC roles that we can choose from. We need our Virtual Machine to only read files from the storage account, so we will go with the Storage Blob Data Reader role. Let's store this role ID in a variable. This ID can be grabbed from the Azure documentation or using the Azure CLI:

READER_ROLE_DEF_ID="/subscriptions/"$(az account list --query [0].id -o tsv)"/providers/Microsoft.Authorization/roleDefinitions/2a2b9908-6ea1-4ae2-8e65-a410df84e7d1"



```
$ READER_ROLE_DEF_ID="/subscriptions/"$(az account list --query [0].id -o tsv)"/providers/Microsoft.Authorization/roleDefinitions/2a2b9908-6ea1-4ae2-8e65-a410df84e7d1"
$ echo $READER_ROLE_DEF_ID
/subscriptions/25f42464-79ed-4c6e-9cb7-5b225bf99b03/providers/Microsoft.Authorization/roleDefinitions/2a2b9908-6ea1-4ae2-8e65-a410df84e7d1
$ 
```

Important: the following CLI command can be used to assign the role to our Virtual Machine. However, for security reasons, our platform does not allow users to assign RBAC roles to identities. You can take note of this command and run it in your own platforms.

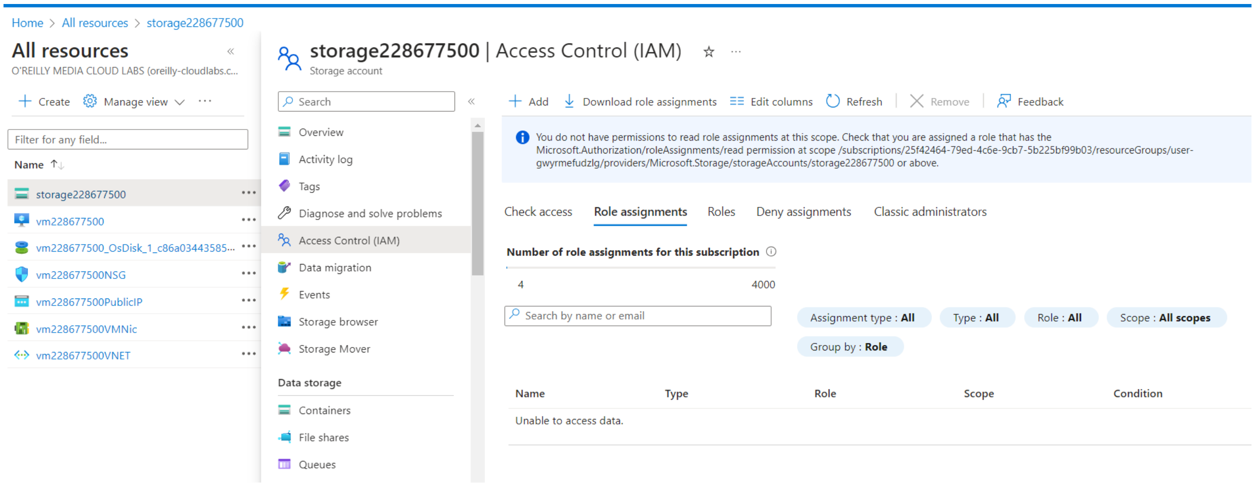
az role assignment create --assignee $VM_OBJECT_ID --role $READER_ROLE_DEF_ID --scope $STORAGE_RESOURCE_ID

In the next step, we will use the Azure CLI command to see role assignments.

Check the Storage Access
Use the following command to list all the assigned RBAC roles to your Azure VM identity:

az role assignment list --resource-group $resource

You can also assign or see RBAC roles assigned to your storage account in the Azure portal. Let's take a look at that in the next step.



You do not have permissions to read role assignments at this scope. Check that you are assigned a role that has the Microsoft.Authorization/roleAssignments/read permission at scope /subscriptions/25f42464-79ed-4c6e-9cb7-5b225bf99b03/resourceGroups/user-gwyrmefudzlg/providers/Microsoft.Storage/storageAccounts/storage228677500 or above.

Number of role assignments for this subscription

4 4000

Name	Type	Role	Scope	Condition
Unable to access data.				

Use the following command to remove the role assignment from your Azure VM:

```
az role assignment delete --resource-group $resource --assignee $VM_OBJECT_ID --role
"Storage Blob Data Reader"
```