Total time available: 75 minutes

Lab requirements:

* User has subscription available
* User is able to provision VMSS instances with Ubuntu and RHEL images
* User is able to use Azure Cloud Shell

# Part 1 - Deploy an Azure VM using the Azure Portal (10 minutes):

* + Click Create a resource in the portal:

Machine generated alternative text:
Create a 
Home 

* + Click Ubuntu 18.04 LTS:

Machine generated alternative text:
Ubuntu Server 18.04 LTS 
Learn more 

* + Fill in the parameters with reasonable defaults
    - Resource group name (create new)
      * Might have to explain what resource groups are
    - VM name
    - SSH key
      * In Cloud Shell, run `ssh-keygen -t rsa -b 2048`
        + Accept all defaults
      * Then run cat ~/.ssh/id\_rsa.pub
      * Copy that text into the SSH public key box
      * Add a reasonable username
    - Public inbound ports: Allow 22
  + Review + Create, then Deploy

# Part 2 - Using the Azure Cloud shell and troubleshooting tools (10 minutes):

* + SSH to VM from the Azure Cloud Shell to ensure it is up



* + - ssh <username>@<ip-address>
    - Play around, log out
  + The Reset password tool is a useful tool for resetting a forgotten SSH key or password
    - It can also be used to add a user to a VM
  + Use reset password tool to add a user to the VM you created
    - azlinux/P@ssword1234

Machine generated alternative text:
update X Discard 
This uses the VMAccessForLinux extension to reset the credentials of an existing user 
Mode O 
@ Reset password 
o 
Reset SSH public key 
o 
Reset configuration only 
username O 
azlinux 
Confirm password 

* + If you ever lose connectivity to your VM (SSH blocked, VM fails to boot), you can use the serial console feature to troubleshoot
  + Go to Serial Console (scroll down to Support + troubleshooting)



* + From within serial console, restart or reset VM to view the boot text

A screenshot of a cell phone

Description automatically generated

* + - This is something that Azure gives you, and something AWS does not have
    - Very useful for using GRUB
* More serial console documentation is available at <https://aka.ms/serialconsolelinux>

Now that you've deployed a VM and gotten comfortable with it, you're probably wondering how to scale a VM in a real-world situation. This is where VM scale sets come in handy, and this is what we will learn in the rest of this lab.

# Part 3: Deploy an Azure VM Scale Set (15 minutes)

* + A VM scale set (VMSS) is a way to deploy a number of similar VMs with a single command – you can use a scale set to deploy between 0 to 1000 VMs. Scale sets have built-in high availability and integrate with Azure auto-scale.
  + We have used the portal to create a VM, so let's use the Azure CLI to deploy a VMSS and an app to it
  + Open up Cloud Shell again and clone this repo: <https://github.com/asinn826/Ignite2019VMSS-HOL>

git clone https://github.com/asinn826/Ignite2019VMSS-HOL.git

* + cd into the repo

cd Ignite2019VMSS-HOL

* + Open up azuredeploy.parameters.json, and edit the azuredeploy.parameters.json file
* Change the vmssName value to a name of your choice
* Change the adminSshKey to the SSH public key value that you generated [earlier in the lab](#_Part_1_-)
  + Create a resource group for your VMSS deployment

az group create --name <resource group name> --location <location – you can use westus>

* + Now, create the deployment by using your new resource group!

az group deployment create -g <resource group name> -n <deployment name> --template-file azuredeploy.json –parameters @azuredeploy.parameters.json

* + Now, wait for the deployment to finish

## Deployment notes to read through while you wait (in other words, what did I just do?)

* + The files azuredeploy.json and azuredeploy.parameters.json are most relevant here
  + The command above just deployed a virtual machine scale set (<https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview>)
  + VM scale sets are great for scaling out applications – for example, if you are running a workload and you anticipate a spike in holiday-related traffic, the VMSS can scale out automatically to meet your compute needs, and scale back in when traffic subsides.
  + The template deployed the VMSS using a Red Hat Enterprise Linux (RHEL) image, one of the many available images in the Azure Marketplace
  + The template also used cloud-init to deploy a simple web application onto the VM. The cloud-init script runs once, at the beginning of deployment, to configure the VM to its desired end state
  + Note about cloud-init: the RHEL image we chose has cloud-init enabled, which means that cloud-init is the provisioning agent
* We are bringing cloud-init to Azure VM images, so this will slowly become the default option
* If you are familiar with cloud-init from other environments, this will function exactly the same
* Ask the lab proctors for more details if you’re curious

# Part 4: Familiarizing yourself with your VM scale set (5 minutes)

* + Go to the Azure Portal, navigate to your resource group, and click on your newly-created VMSS
  + Go to Instances
    - Note that you only have one – this was defined by the template

A close up of a sign

Description automatically generated

* + Go to individual instance
    - Here you can view details for your individual VMSS instance - you can restart, deallocate, reimage, or upgrade instance
  + Note: What is upgrading?
    - VM scale sets work on a model basis - there is a model that the scale set follows, and all VMs within the scale set have the same configuration as defined in the model
    - If the model changes, you will need to upgrade the instance(s) to the latest model - hence the Upgrade button in the UI (you can also configure scale sets to automatically upgrade when the model changes)
    - If you have more questions, ask a proctor
  + In the instance view, you can also access features like Azure Bastion, Serial console, and Boot diagnostics
    - Note that none of these were configured initially in your VM scale set, so you will need to upgrade the scale set model and then update individual instances to use them
    - There is a bonus section to this lab where you can try this for yourself

# Part 5: Use autoscale rules on your VM scale set (25 minutes)

* + Go to Scaling in the VMSS

Machine generated alternative text:
Scalmg 

* + Currently, the VMSS is set to automatic scaling

A screenshot of a cell phone screen with text

Description automatically generated

* + The VMSS will scale automatically based on load as measured by % CPU usage

## Autoscale via the deployed web application

* + Go back to the Overview section for your scale set

A close up of a screen

Description automatically generated

* + Copy the public IP address, and navigate to <ip-address>:9000 in your browser
  + You will see a landing page that looks like:

A screenshot of a social media post

Description automatically generated

* + To view the autoscale in action, simply click “Start work” on the page
  + Then, go back to the VM scale set in the Azure portal and watch its CPU rise
    - Once CPU > 60%, a new scale set instance will automatically be created

A screenshot of a cell phone

Description automatically generated

* + Go to Instances and watch VMs get created

A close up of a screen

Description automatically generated

* + You can also SSH into your individual instance by running:

ssh <adminusername>@<ip-address> -p 50000

## Autoscale manually in the Azure Portal based on date/time

* + Go back to the portal and go to Scaling, and let's try out creating a scheduled autoscale rule (<https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-overview#scheduled-autoscale>)
  + Click Custom autoscale, and note all the options you have

A screenshot of a computer screen

Description automatically generated

* + Leave the auto created scale condition with the default settings
  + Click on "Add a scale set condition"
    - Select "Sale to a specific instance count
    - Set Instance count to a random number >1 and <100
    - Select "Specify start/end dates"
    - Timezone: "(UTC-05:00) Eastern Time (US & Canada)"
    - Start date: today, November 15, 2019, and a time a minute or two in the future
    - End date: today, November 15, 2019, and a time several minutes in the future
  + Machine generated alternative text:
    Auto created scale condition 1 
    Scale mode O 
    Scale based on a metric 
    Instance count 20 
    Scale to a specific instance count 
    Repeat specific days 
    Schedule 
    Timezone 
    Start date 
    End date 
    SpeciW start/end dates 
    (UTC-05:OO) Eastern Time US & Canada) 
    11/05/2019 
    11/05/2019 
    3:1000 PM 
    PM 
    + Add scaLe condition 

## Autoscale documentation and information

* + See the docs here for more info on autoscale rules:
    - <https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-overview>
    - <https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-portal>
  + The following examples are scenarios that may benefit the use of schedule-based autoscale rules:
    - Automatically scale out the number of VM instances at the start of the work day when customer demand increases. At the end of the work day, automatically scale in the number of VM instances to minimize resource costs overnight when application use is low.
    - If a department uses an application heavily at certain parts of the month or fiscal cycle, automatically scale the number of VM instances to accommodate their additional demands.
    - When there is a marketing event, promotion, or holiday sale, you can automatically scale the number of VM instances ahead of anticipated customer demand.
  + The following scenarios may benefit from load-based autoscale:
    - Elastic loads with no set schedule – devops builds for an organization, a web server that can receive traffic from anywhere at any time

 Note: You can combine multiple scale-out and scale-in conditions.

**You are done the lab!**

# Bonus section (optional): configure your VMSS for serial console (10-15 extra minutes)

* + - Add a password to your VMSS

Machine generated alternative text:
Reset passwcyd 

* + azlinux/P@ssword1234 will work
  + Go to Instances and pick a random instance
  + Open up serial console in your VMSS instance and get the boot diagnostics error

Machine generated alternative text:
diagnostics is not configured for this virtual machine. 

Machine generated alternative text:
la nostics 
Boot 
R save 
X Discard 
Status 
(Doff @ 
Diagnostics storage account 
Create new 

* + Go back to the instance view and upgrade the instance to the latest model

Machine generated alternative text:
UF"e 

* + Once the upgrade completes, go back to serial console, and it should load
  + You can log into serial console - helpful if you need to debug VMSS connectivity