# Practice M2: Advanced Infrastructure Services

For this module we will need a PC running recent version of **Windows**, **Linux**, or **macOS**. It can be either physical or virtual machine

All resource related tasks can be executed either in the locally installed tools or in the **Azure Cloud Shell**

**Any type of subscription** should be suitable for doing the exercises

Azure CLI code blocks are not formatted for multi-line execution as the separator is different in Windows CMD and Linux Bash

## Part 1: Core Services

#### Portal: Two VMs in an Availability Set + Cloud Init + Load Balancer + Security Group

Navigate to <https://portal.azure.com>

Enter your credentials

##### Resource group

Go to **Resource groups**

Create new resource group **RG-Demo-P1-1** in the **West Europe** region

##### Virtual network

Enter the resource group and click on the **+ Add** button to add new resources

In the search field enter **Virtual Network** and hit **Enter**

Then click on the **Create** button

For name enter **p11vnet**

Ensure that the **Region** is set to **West Europe**

Click the **Next: IP Addresses** button

Leave the **address space** as it is. Make sure that it is **10.0.0.0/16** and for the **default** subnet is must be **10.0.0.0/24**

Click on the **Review + create** button

Finally, click on the **Create** button

##### Network security group

Return to the resource group and click on the **+ Add** button to add new resources

In the search field enter **Network security group** and hit **Enter**

Then click on the **Create** button

For **name** enter **p11sg**

Ensure that the **Region** is set to **West Europe**

Accept the proposed values for all other parameters (subscription and resource group)

Click on the **Review + create** button and then on **Create**

Once the deployment is done click on the **Go to resource** button

In the **Settings** section click on **Subnets** to associate the group with a network

Click on the **+ Associate** button

In **Virtual network** drop-down select **p11vnet** and then in the **Subnet** drop-down select **default**

Click on **OK**

In the **Settings** section click on the **Inbound security rules** to add two rules

##### Security rules

Click on the **+ Add** button

Change **Destination port ranges** to **22**

Change **Protocol** to **TCP**

In the **Name** field enter **Port\_22**

Click on the **Add** button

Repeat the procedure once again but change **22** to **80**

##### Virtual machines

Return to the resource group and click on the **+ Add** button to add new resources

Select **Ubuntu Server 18.04 LTS** from the list of popular resources

Leave **Subscription** and **Resource group** as they are

In the **Virtual machine name** enter **p11vm1**

Set the **Region** to **West Europe**

For **Availability options** set **Availability set**

Click on the **Create new** under the **Availability set**

For **Name** set **p11as**

Click **OK**

Change the **Size** to **Standard B1S**

Select **Password** as **Authentication type**

Enter **demouser** as **Username**

For **Password** set for example **DemoPassword-2021**

Of course, you can use a different pair of credentials

In the **Public inbound ports** section select **None**

Click on the **Next: Disks** button

Accept all default values and click on the **Next: Networking** button

Do not change proposed values in the **Virtual network**, **subnet**, and **Public IP** fields

In the **NIC network security group** select **Advanced**

Select **p11sg** from the **Configure network security group** drop-down

Click on the **Next: Management** button and then on the **Next: Advanced** button

In the **Cloud init** section paste (be sure the keep the formatting incl. empty spaces) the following text:

**#cloud-config**

**package\_upgrade: true**

**packages:**

**- apache2**

**- php**

**write\_files:**

**- path: /var/www/html/index.php**

**content: |**

**Hello from <b><?php echo gethostname(); ?></b>**

**runcmd:**

**- systemctl restart apache2**

**NOTE:** You can find this in the **cloud-init.yaml** file in the archive provided with the practice

Click on the **Review + create** button

Click on the **Create** button

Once the deployment is done, repeat the procedure but enter **p11vm2** for VM name

##### Load balancer

Now, if we test with the public IP address of either machine, we must see a simple web page

Let’s add a load balancer in from of the VMs

Return to the resource group and click on the **+ Add** button to add new resources

In the search field enter **Load Balancer** and hit **Enter**

Then click on the **Create** button

In the **Name** field enter **p11lb**

Set the **Region** to **West Europe**

Make sure that **Type** is set to **Public** and **SKU** is set to **Basic**

In the **Public IP address name** enter **p11lb-ip**

Accept all other values as they are and click on **Review + Create** and then on **Create**

Once the deployment is done click on the **Go to resource** button

In the **Settings** section click on the **Backend pools**

Then click on the **+ Add** button

For **Name** enter **p11lb-back-pool**

In the **Virtual network** drop-down list select the **p11vnet** created earlier

Change **Associated to** to **Virtual machines**

In the **Virtual machines** section select **p11vm1** and **ipconfig1** under **IP address**

In the **Virtual machines** section select **p11vm2** and **ipconfig1** under **IP address**

Finally, click **Add**

Next, go to **Health probes**

Click on the **+ Add** button

For **Name** enter **p11lb-health**, accept the default values and click on **Add**

Next, go to the **Load balancing rules** section in the menu

Click on the **+ Add** button

In the **Name** field enter **p11lb-rule**

Accept all other values and click on **OK**

Now, return to the **Overview** of the load balancer

Copy the **Public IP address** and paste in a browser tab

Refresh several times

You should see that the page is served by the two virtual machines

#### Azure CLI: Two VMs in an Availability Set + Cloud Init + Load Balancer + Security Group

*NOTE: If you went through the steps using the* ***Azure Portal*** *then at first, you must delete the* ***RG-Demo-P1-1*** *resource group. Once it is deleted, you can continue with the steps below. Alternatively, you can adjust the resource group’s name on the following lines*

If using the tool on-premise, issue:

**az login**

If using the **Azure Cloud Shell**, this is not necessary

##### Resource group

Before you continue, you should wither delete the resource group, created earlier, or adjust the resource group name used in the commands that follow

Creating a resource group is done with:

**az group create --name RG-Demo-P1-1 --location westeurope**

##### Network security group

First, create the network security group:

**az network nsg create --name p11sg --resource-group RG-Demo-P1-1**

##### Security rules

Then, add both inbound rules:

**az network nsg rule create --name Port\_22 --nsg-name p11sg --resource-group RG-Demo-P1-1 --access Allow --protocol tcp --direction inbound --priority 100 --destination-port-range 22**

**az network nsg rule create --name Port\_80 --nsg-name p11sg --resource-group RG-Demo-P1-1 --access Allow --protocol tcp --direction inbound --priority 110 --destination-port-range 80**

##### Virtual network

Now, let’s create the virtual network:

**az network vnet create --name p11vnet --resource-group RG-Demo-P1-1**

And now, the subnet:

**az network vnet subnet create --name default --vnet-name p11vnet --resource-group RG-Demo-P1-1 --address-prefix 10.0.0.0/24 --network-security-group p11sg**

Let’s create both virtual network adapters:

**az network nic create --name p11nic1 --resource-group RG-Demo-P1-1 --vnet-name p11vnet --subnet default**

**az network nic create --name p11nic2 --resource-group RG-Demo-P1-1 --vnet-name p11vnet --subnet default**

##### Virtual machines

First, we will create the availability set:

**az vm availability-set create --name p11as --resource-group RG-Demo-P1-1 --platform-fault-domain-count 2 --platform-update-domain-count 2**

Then, let’s prepare the **cloud init** file

Using an editor, open an empty file and store the code from the previous task in a file named **cloud-init.yaml**

Don’t forget to check if the first line contains **#cloud-config**

Now, we are ready to create the first machine:

**az vm create --name p11vm1 --resource-group RG-Demo-P1-1 --image UbuntuLTS --size Standard\_B1s --authentication-type password --admin-username demouser --admin-password DemoPassword-2021 --custom-data cloud-init.yaml --nic p11nic1 --availability-set p11as --verbose**

The **--verbose** at the end is to monitor the creation progress. It can be applied on other commands as well

And, then the second one:

**az vm create --name p11vm2 --resource-group RG-Demo-P1-1 --image UbuntuLTS --size Standard\_B1s --authentication-type password --admin-username demouser --admin-password DemoPassword-2021 --custom-data cloud-init.yaml --nic p11nic2 --availability-set p11as --verbose**

##### Load balancer

First, we will create a public IP address for our load balancer:

**az network public-ip create --name p11lb-ip --resource-group RG-Demo-P1-1 --allocation-method dynamic**

Then, the load balancer itself:

**az network lb create --name p11lb --resource-group RG-Demo-P1-1 --frontend-ip-name p11lb-fe --backend-pool-name p11lb-be --public-ip-address p11lb-ip**

Next step is to create a health probe:

**az network lb probe create --name p11lb-hp --lb-name p11lb --resource-group RG-Demo-P1-1 --protocol tcp --port 80**

We need a load balancing rule as well:

**az network lb rule create --name p11lb-rule --lb-name p11lb --resource-group RG-Demo-P1-1 --protocol tcp --frontend-port 80 --backend-port 80 --frontend-ip-name p11lb-fe --backend-pool-name p11lb-be --probe-name p11lb-hp**

As last step, especially if we first created the VMs and NICs, we must update IP configurations of both virtual network adapters. The first or default IP configuration is named **ipconfig1** for both adapters:

**az network nic ip-config update --name ipconfig1 --resource-group RG-Demo-P1-1 --nic-name p11nic1 --lb-name p11lb --lb-address-pools p11lb-be**

**az network nic ip-config update --name ipconfig1 --resource-group RG-Demo-P1-1 --nic-name p11nic2 --lb-name p11lb --lb-address-pools p11lb-be**

Let’s get the public IP address of the load balancer:

**az network public-ip show --name p11lb-ip --resource-group RG-Demo-P1-1 --query [ipAddress] --output tsv**

Open a browser tab and navigate to the address

Refresh a few times

You should see that the page is served by the two virtual machines

#### Azure PS: Two VMs in an Availability Set + Cloud Init + Load Balancer + Security Group

*NOTE: If you went through the steps using the* ***Azure Portal*** *or* ***Azure CLI*** *then at first, you must delete the* ***RG-Demo-P1-1*** *resource group. Once it is deleted, you can continue with the steps below. Alternatively, you can adjust the resource group’s name on the following lines*

If using the tool on-premise, issue:

**Connect-AzAccount**

If using the **Azure Cloud Shell**, this is not necessary

##### Resource group

Before you continue, you should wither delete the resource group, created earlier, or adjust the resource group name used in the commands that follow

Creating a resource group is done with:

**$LO = "westeurope"**

**$RG = "RG-Demo-P1-1"**

**New-AzResourceGroup -Name $RG -Location $LO**

##### Network security group

First, create the network security group rules:

**$rule1 = New-AzNetworkSecurityRuleConfig -Name Port\_22 -Access Allow -Protocol Tcp `**

**-Direction Inbound -Priority 100 -SourceAddressPrefix \* -SourcePortRange \* `**

**-DestinationAddressPrefix \* -DestinationPortRange 22**

**$rule2 = New-AzNetworkSecurityRuleConfig -Name Port\_80 -Access Allow -Protocol Tcp `**

**-Direction Inbound -Priority 110 -SourceAddressPrefix \* -SourcePortRange \* `**

**-DestinationAddressPrefix \* -DestinationPortRange 80**

And then the security group:

**$nsg = New-AzNetworkSecurityGroup -Name "p11sg" -ResourceGroupName $RG -Location $LO `**

**-SecurityRules $rule1,$rule2**

##### Virtual network

First, define the subnet:

**$S1 = New-AzVirtualNetworkSubnetConfig -Name 'default' -AddressPrefix '10.0.0.0/24' `**

**-NetworkSecurityGroup $nsg**

Ignore the warnings

Then, the network itself:

**$VN = New-AzVirtualNetwork -Name 'p11vnet' -ResourceGroupName $RG `**

**-AddressPrefix '10.0.0.0/16' -Location $LO -Subnet $S1**

Now, it is time to define the two network interface cards:

**$N1 = New-AzNetworkInterface -Name 'p11nic1' -ResourceGroupName $RG -Location $LO `**

**-Subnet $VN.Subnets[0]**

**$N2 = New-AzNetworkInterface -Name 'p11nic2' -ResourceGroupName $RG -Location $LO `**

**-Subnet $VN.Subnets[0]**

##### Virtual machines

Let’s create the availability set:

**$AS = New-AzAvailabilitySet -Name "p11as" -ResourceGroupName $RG -Location $LO -Sku Aligned `**

**-PlatformFaultDomainCount 2 -PlatformUpdateDomainCount 2**

Store the credentials for the two virtual machines, for example **demouser** / **DemoPassword-2021**

**$CR = Get-Credential**

Store the contents of the provision file in a variable:

**$CD = Get-Content -Raw cloud-init.yaml**

Create virtual machine configuration for the first machine:

**$VC = New-AzVMConfig -VMName 'p11vm1' -VMSize 'Standard\_B1s' -AvailabilitySetId $AS.Id | `**

**Set-AzVMOperatingSystem -Linux -ComputerName 'p11vm1' -Credential $CR `**

**-CustomData $CD | Set-AzVMSourceImage -PublisherName 'Canonical' `**

**-Offer 'UbuntuServer' -Skus '18.04-LTS' -Version latest | Add-AzVMNetworkInterface -Id $N1.Id**

Then, the machine itself:

**New-AzVM -ResourceGroupName $RG -Location $LO -VM $VC**

Same procedure for the second one – configuration:

**$VC = New-AzVMConfig -VMName 'p11vm2' -VMSize 'Standard\_B1s' -AvailabilitySetId $AS.Id | `**

**Set-AzVMOperatingSystem -Linux -ComputerName 'p11vm2' -Credential $CR `**

**-CustomData $CD | Set-AzVMSourceImage -PublisherName 'Canonical' `**

**-Offer 'UbuntuServer' -Skus '18.04-LTS' -Version latest | Add-AzVMNetworkInterface -Id $N2.Id**

And then the machine:

**New-AzVM -ResourceGroupName $RG -Location $LO -VM $VC**

##### Load balancer

First, we will define the public IP address

**$LBIP = New-AzPublicIpAddress -Name 'p11lb-ip' -ResourceGroupName $RG `**

**-Location $LO -AllocationMethod Dynamic -Sku Basic**

Ignore the warning. We are using SKU Basic and not the Standard, so no worries here

Then, the front-end part of the load balancer:

**$LBFE = New-AzLoadBalancerFrontendIpConfig -Name 'p11lb-fe' -PublicIpAddress $LBIP**

Ignore the warning. It is related to the public IP that we are using here

Now, we will define the back-end pool:

**$LBBE = New-AzLoadBalancerBackendAddressPoolConfig -Name 'p11lb-be'**

The health probe as well:

**$LBHP = New-AzLoadBalancerProbeConfig -Name 'p11lb-hp' -Protocol tcp -Port 80 `**

**-IntervalInSeconds 5 -ProbeCount 2**

The rule can be created with this command:

**$LBRL = New-AzLoadBalancerRuleConfig -Name 'p11lb-rule' -FrontendIpConfiguration $LBFE `**

**-BackendAddressPool $LBBE -Protocol Tcp -FrontendPort 80 -BackendPort 80 -Probe $LBHP**

And finally, the load balancer:

**$LB = New-AzLoadBalancer -Name 'p11lb' -ResourceGroupName $RG -Location $LO `**

**-FrontendIpConfiguration $LBFE -BackendAddressPool $LBBE -Probe $LBHP -LoadBalancingRule $LBRL**

One last step remains, we must assign the network adapters to the back-end pool:

**$N1.IpConfigurations[0].LoadBalancerBackendAddressPools = $LBBE**

**$N1 | Set-AzNetworkInterface**

**$N2.IpConfigurations[0].LoadBalancerBackendAddressPools = $LBBE**

**$N2 | Set-AzNetworkInterface**

Get the public IP address of the load balancer:

**Get-AzPublicIPAddress -Name 'p11lb-ip' -ResourceGroupName $RG | Select IpAddress**

Open a browser tab and navigate to the address

Refresh a few times

You should see that the page is served by the two virtual machines

#### Portal: Virtual Machine Scale Set (VMSS)

Navigate to <https://portal.azure.com>

Enter your credentials

##### Resource group

Go to **Resource groups**

Create new resource group **RG-Demo-P1-2** in the **West Europe** region

##### Virtual machines scale set

Enter the resource group and click on the **+ Add** button to add new resources

In the search field enter **Virtual machine scale set** and press **Enter**

Then click on the **Create** button

For the **Virtual machine scale set name** enter **vss**

Make sure that **West Europe** is selected in the **Region** field

From the list of **Operating system disk images** choose **Ubuntu Server 18.04 LTS**

Set **Instance size** to **Standard B1s**

In the **Username** field enter **demouser**

For password use **DemoPassword-2021**

Of course, you can use another set of credentials

Click on the **Next: Disks** button

Click again on the **Next: Networking** button

Edit the proposed network / nic settings by clicking the **pencil** icon

Change the **Public inbound ports** to **Allow selected ports**

Select port **HTTP (80)** and **SSH (22)** in the **Select inbound ports** drop-down list

Confirm with the **OK** button

Set the **Use a load balancer** option to **Yes**

Accept the default settings for the load balancer

Click on the **Next: Scaling** button

Leave **Instance count** to **2**

Accept the other scaling options defaults and click on the **Next: Management** button

Accept the default values and click on the **Next: Health** button

Click on the **Next: Advanced** button

In the **Custom data** field paste:

**#cloud-config**

**package\_upgrade: true**

**packages:**

**- apache2**

**- php**

**write\_files:**

**- path: /var/www/html/index.php**

**content: |**

**Hello from <b><?php echo gethostname(); ?></b>**

**runcmd:**

**- systemctl restart apache2**

**NOTE:** You can find this in the **cloud-init.yaml** file in the archive provided with the practice

Click on the **Review + create** button

Finally, click the **Create** button

Once the deployment is done, click on the **Go to resource** button

Get the public IP address and paste it into a browser window, a web site should be seen

#### Azure CLI: Virtual Machine Scale Set (VMSS)

*NOTE: If you went through the steps using the* ***Azure Portal*** *then at first, you must delete the* ***RG-Demo-P1-2*** *resource group. Once it is deleted, you can continue with the steps below. Alternatively, you can adjust the resource group’s name on the following lines*

If using the tool on-premise, issue:

**az login**

If using the Azure Cloud Shell, this is not necessary

##### Resource group

Let’s first create a resource group:

**az group create --name RG-Demo-P1-2 --location westeurope**

##### Virtual machine scale set

In order to create a virtual machine scale set, like the one created in the previous section, we must execute:

**az vmss create --resource-group RG-Demo-P1-2 --name vss --image UbuntuLTS --vm-sku Standard\_B1s --instance-count 2 --upgrade-policy-mode automatic --custom-data cloud-init.yaml --authentication-type password --admin-username demouser --admin-password DemoPassword-2021**

Knowing how the names of the components are structured, we can execute the following command in order to create a load balancer rule:

**az network lb rule create --resource-group RG-Demo-P1-2 --name vss-lb-rule --lb-name vssLB --backend-pool-name vssLBBEPool --backend-port 80 --frontend-ip-name loadBalancerFrontEnd --frontend-port 80 --protocol tcp**

Let’s get the public IP address of the load balancer:

**az network public-ip show --resource-group RG-Demo-P1-2 --name vssLBPublicIP --query [ipAddress] --output tsv**

And paste it into a browser window, a web site should be seen. Refresh several times to check if everything is working as expected

#### Azure PS: Virtual Machine Scale Set (VMSS)

*NOTE: If you went through the steps using the* ***Azure Portal*** *or* ***Azure CLI*** *then at first, you must delete the* ***RG-Demo-P1-2*** *resource group. Once it is deleted, you can continue with the steps below. Alternatively, you can adjust the resource group’s name on the following lines*

If using the tool on-premise, issue:

**Connect-AzAccount**

If using the Azure Cloud Shell, this is not necessary

##### Resource group

Create variables for the resource group and location

**$RG = 'RG-Demo-P1-2'**

**$LO = 'westeurope'**

**New-AzResourceGroup -ResourceGroupName $RG -Location $LO**

Next, we prepare the rules and the network security group:

**$rule1 = New-AzNetworkSecurityRuleConfig -Name Port\_22 -Access Allow -Protocol Tcp `**

**-Direction Inbound -Priority 100 -SourceAddressPrefix \* -SourcePortRange \* `**

**-DestinationAddressPrefix \* -DestinationPortRange 22**

**$rule2 = New-AzNetworkSecurityRuleConfig -Name Port\_80 -Access Allow -Protocol Tcp `**

**-Direction Inbound -Priority 110 -SourceAddressPrefix \* -SourcePortRange \* `**

**-DestinationAddressPrefix \* -DestinationPortRange 80**

**$NSG = New-AzNetworkSecurityGroup -Name "NSG" -ResourceGroupName $RG `**

**-Location $LO -SecurityRules $rule1,$rule2**

Then, we move forward to the network part. First, create the subnet:

**$S1 = New-AzVirtualNetworkSubnetConfig -Name VNET1-S1 `**

**-AddressPrefix 10.0.0.0/24 -NetworkSecurityGroup $NSG**

Ignore the warning

Now, create the network itself:

**$NET = New-AzVirtualNetwork -ResourceGroupName $RG -Location $LO -Name VNET1 `**

**-AddressPrefix 10.0.0.0/16 -Subnet $S1**

Now, it is time to create the load balancer. First, the IP address:

**$LBIP = New-AzPublicIpAddress -ResourceGroupName $RG -Location $LO `**

**-Name 'LB-IP' -AllocationMethod Dynamic**

Ignore the warning. We are creating a public IP address of Basic SKU (even if not explicitly specified)

Then, the front-end configuration:

**$LBFE = New-AzLoadBalancerFrontendIpConfig -Name 'LB-FE' -PublicIpAddress $LBIP**

Ignore the warning. It is because of the public IP address

Next, the back-end configuration:

**$LBBE = New-AzLoadBalancerBackendAddressPoolConfig -Name 'LB-BE'**

The health probe:

**$LBHP = New-AzLoadBalancerProbeConfig -Name 'LB-HP' -Protocol tcp -Port 80 `**

**-IntervalInSeconds 5 -ProbeCount 2**

The load balancing rule:

**$LBRL = New-AzLoadBalancerRuleConfig -Name 'LB-RULE' -FrontendIpConfiguration $LBFE `**

**-BackendAddressPool $LBBE -Protocol Tcp -FrontendPort 80 -BackendPort 80 -Probe $LBHP**

And the load balancer itself:

**$LB = New-AzLoadBalancer -Name 'LB' -ResourceGroupName $RG -Location $LO `**

**-FrontendIpConfiguration $LBFE -BackendAddressPool $LBBE -Probe $LBHP -LoadBalancingRule $LBRL**

We are ready to prepare our virtual machine scale set. First, we will create the IP configuration:

**$VSSIPC = New-AzVmssIpConfig -Name 'VSSIPCONF' -Primary $true -SubnetId $NET.Subnets[0].Id `**

**-LoadBalancerBackendAddressPoolsId $LBBE.Id**

Then, the scale set configuration:

**$VSSCONF = New-AzVmssConfig -Location $LO -SkuCapacity 2 -SkuName Standard\_B1s `**

**-UpgradePolicyMode Automatic**

Ignore the warning as we are passing one of the stated parameters

Next is the storage profile:

**Set-AzVmssStorageProfile $VSSCONF -ImageReferencePublisher Canonical `**

**-ImageReferenceOffer UbuntuServer -ImageReferenceSku '18.04-LTS' `**

**-ImageReferenceVersion latest -OsDiskCreateOption FromImage**

The OS profile of the instances:

**Set-AzVmssOsProfile $VSSCONF -AdminUsername 'demouser' -AdminPassword 'DemoPassword-2021' `**

**-ComputerNamePrefix 'VM'**

Last is the networking configuration:

**Add-AzVmssNetworkInterfaceConfiguration -VirtualMachineScaleSet $VSSCONF -Name 'VSSNETCONF' `**

**-Primary $true -IPConfiguration $VSSIPC**

And finally, we initiate the scale set creation:

**$vmss = New-AzVmss -ResourceGroupName $RG -Name 'VSS' -VirtualMachineScaleSet $VSSCONF**

Just, after we have our scale set up and running, we can continue with the provision process. For this to happen, we must prepare a custom configuration. We must upload somewhere the script:

**$customConfig = @{**

**"fileUris" = (,"http://tuionui.com/custom-script.sh");**

**"commandToExecute" = "bash custom-script.sh"**

**}**

And then, install an extension that will push the configuration:

**$vmss = Add-AzVmssExtension -VirtualMachineScaleSet $vmss -Name "customScript" `**

**-Publisher "Microsoft.Azure.Extensions" -Type "CustomScript" `**

**-TypeHandlerVersion 2.0 -Setting $customConfig**

In order to initiate the installation process, we must update the scale set:

**Update-AzVmss -ResourceGroupName $RG -Name "VSS" -VirtualMachineScaleSet $vmss**

After minute or so, both instances will be updated. You can monitor their status in the portal

Once updated, you can get the public IP address with:

**Get-AzPublicIPAddress -Name 'LB-IP' -ResourceGroupName $RG | Select IpAddress**

And navigate in a browser tab to this URL:

**http://<public ip>/index.php**

Refresh several times to see that the setup is working

#### Portal: BLOB + Files

Navigate to <https://portal.azure.com>

Enter your credentials

##### Resource group

Go to **Resource groups**

Create new resource group **RG-Demo-P1-3** in the **West Europe** region

Enter the resource group and click on the **+ Add** button to add new resources

In the search field enter **Storage account** and hit **Enter**

Then click on the **Create** button

For name enter something unique, for example **sadvzaze**

Set the **Location** to **West Europe**

Leave all other values as they are and click on the **Review + create** button

Then click on the **Create** button

##### BLOB

Enter (navigate to) the account you just created

Click on the **Containers** section

Click on the **+ Container** button to create new one

For **Name** enter **blobs** and click on **Create**

Click on the newly created container

Now, we can upload one or more files to it

Click on the **Upload** button

Click on the **Browse** button and select one or more files, then click on **Open**

Finally, click on **Upload**

Depending on the selected file’s size, after a while it will be uploaded

##### Files

Return to the storage account

Click on the **File shares** option

Click on the **+ File share** button to create one

For **Name** enter **share**

For **Quota** enter for example **2**

Click on **Create**

Click on the share and then on the **Upload** button to upload a file

You can upload the same file as you did with the BLOB

##### Storage Explorer

Download the **Storage Explorer** application from <https://azure.microsoft.com/en-us/features/storage-explorer/>

Install it and run it

Connect it to your subscription

Navigate to the storage account and explore its contents

## Part 2: Users. Roles. Policies

Login to the Azure portal if you are not

#### Users

Enter **Users** in the search field, then hit the **Enter** key

Click on the **+ New user** button

In the **User name** field enter something, for example **azeuser**

For **Name** enter **AzE User**

Select the **Let me create the password** option

Then for **Initial password** enter something (up to 16 symbols), for example **Password-2020**

Click on **Create**

Now, the new account which will be **azeuser@<domain>.onmicrosoft.com** can be used for login

Open new browser window in private mode and navigate to **portal.azure.com**

Enter the account name

Next you will be asked to change the password

Once you are in the portal, you will notice that the new user is not assigned to a subscription

#### RBAC

Return to the session with your main user

In the search field enter **Subscription** and hit **Enter**

Now, we will grant certain privileges to our new user on a subscription level

Go to **Access control (IAM)**

Switch to **Role assignments**

Click on the **+ Add** button

Select **Add role assignments**

In the **Role** field select **Virtual Machine Contributor**

Select the user from the users list

Click on **Save**

Now, if we return to the private session and refresh, the user must be able to create virtual machines

In a similar way, we can make him an owner of a resource group, for example

#### Policies

Go to **Home** view with you regular user

In the search bar enter **Policy** and hit **Enter**

Let’s go to **Definitions**

Filter the **Category** to **Compute** only

Click on the **Allowed virtual machine SKUs**

Explore its contents

Click on the **Assign** button

In scope choose both **subscription** and **resource group**. For example, the one we created earlier

Click **Select**

In the **Assignment name** append **only B1S**

Click on **Next**

In the **Allowed SKUs** select **Standard\_B1s**

Click **Review + create**

Finally, click on **Create**

Return to **Assignments** in the **Policies.** Our new policy assignment should be there

Now, let’s test it

Go to the resource group and try to create a VM of different size

The process should not complete successfully

In fact, instead at the **review + create** phase, it will stop during the actual **create** phase

#### Locks

Locks are an effective prevention mechanism against deleting resources by a mistake

Let’s navigate to a resource, for example one of the machines, created during the first part – **p11vm1**

In the **Overview** mode, scroll down to the **Settings** section

There is the **Locks** command. Click on it

Click on the **+ Add** button

For **Lock name** enter **vm-lock**

Set the **Lock type** to **Delete**

In the **Notes** field enter **Do not delete this VM**

Click on the **OK** button

Now, let’s try to delete the VM. We won’t succeed, even though we created the lock

In order to delete the resource, first we must delete the lock

## Part 3: Monitoring and Costs. Templates

Login to the **Azure Portal** if you are not

#### Cost Management

Go to the **Home** view

From the menu on your left choose **Cost Management + Billing**

It opens in **Overview** mode

Here we can see information about the past few periods and our currently accumulated amount

Click on the **Cost Management**

Then on **Cost analysis**

Here, interactively we can see what led to our current bill and an estimation

By clicking on **Accumulated costs** we can switch between different built-in views

We can change the time frame or add a filter

We can change the **Group by** clause as well

There are options to save and export our cost analysis

From this tool we can set budgets as well

Or ask the **Advisor** for recommendations

Return to **Cost Management + Billing**

Click on **Subscriptions** and select your subscription

Here we can see at a glance all **resource groups** and **resources**

If you visit the **Locks** section, you can also see all locks in the currently selected subscription

#### Limits

Being in the **Cost Management + Billing**

Another interesting option is the **Usage + quotas**. Let’s go there

Select **Microsoft.Compute** from the **All providers** drop-down

Now, we can see what our consumption is and how much room we have

Should we need bigger limits, we can hit the **Request Increase** button

#### Calculators

We can estimate how much will cost if we deploy a solution in Azure with:

<https://azure.microsoft.com/en-us/pricing/calculator/>

In a similar way, we can calculate what will be the savings if we migrate to Azure with:

<https://azure.microsoft.com/en-us/pricing/tco/calculator/>

#### Tags

As we saw earlier, we can break costs by different criteria

Tags, provide us with a way to break them further

We can add tags for example to resource groups and resources

Tags applied to a resource group are not inherited by its resources

We can add a few tags on the artifacts we created so far

Return to one of the resource groups and add tags to the VMs for example

#### Azure Resource Manager Templates

##### Obtain a template

We can download a template from an existing resource

Go to the resource, for example a VM

Click on the **Export template**

From here, we can add the template to our library, or download it on our PC

We can copy the template code and paste it into the **ARMVIZ** (http://armviz.io/) application in order to visualize it

By using this template, we may omit some related resources. A better option would be to use extract the template from a resource group

##### Quickstart templates

We can go one step further. There is a publicly available library with sample templates:

<https://github.com/Azure/azure-quickstart-templates>

Explore a few

You can either copy a definition from here and paste it into the ARMVIZ tool, or use a direct **Visualize** link if one provided

You can go to one of the resource groups, then navigate to **Deployments** or to **Automation > Export template** and copy and paste one of them into the visualization tool

##### Portal: ARM Templates #1

Go to the **Home** view

In the search bar enter **Deploy a custom template** and hit **Enter**

Here we can:

* Use our own template
* Use one of the proposed
* Use a quickstart template

Let’s go with the quickstart one

Type **wordpress** in the **Select a template (disclaimer)** drop-down

Select the **wordpress-single-vm-ubuntu** item

Click on the **Select template** button

Here, we can edit either the template or its parameters

Let’s fill-in the required fields:

* Click on **Create new** under the **Resource group** and for **Name** enter **RG-Template**
* Set **Vm Dns Name** to **azewp**
* In **Admin Username** enter **demouser**
* For **My Sql Password** set **DemoPassword-2021**
* Enter **Standard\_B1s** for **Vm Size**
* Set **Authentication Type** to **password**
* Enter **DemoPassword-2021** in the **Admin Password Or Key** field

Click on the **Review + create** button

Finally, click on the **Create** button to start the deployment process

Sit back and watch the deployment

After the deployment finishes, click on the **Go to resource group** button

Navigate to the virtual machine and copy either the **Public IP address**, or the **DNS name**

If all went according to the plan, we should see the welcome page of **Apache**

If we modify the URL to <http://azewp.westeurope.cloudapp.azure.com/wordpress>, we will see the welcome page of **WordPress**

##### Portal: ARM Templates #2

This time we will try something simpler – just a Linux (Ubuntu) VM

In the search bar enter **Deploy a custom template** and hit **Enter**

Click on **Create a Linux virtual machine** under **Common templates**

Check the values for **Subscription**, **Resource group** and **Location**

All parameters marked with a **red star** are mandatory

Change **VM Name** to **VM1**

Set **Admin Username** to **demouser**

Change **Authentication Type** to **Password**

In the **Admin Password Or Key** enter **DemoPassword-2021**

Change **VM Size** to **Standard\_B1s**

Click on the **Review + create** button

Finally, click on the **Create** button

Monitor the deployment process

Once the deployment is complete, you can navigate to the virtual machine

You can click on **Connect** in the **Overview** section of the VM in order to acquire information how to connect to it

##### Azure CLI: ARM Templates

Same template can be deployed on the command line as well

If using the tool on-premise, issue:

**az login**

If using the **Azure Cloud Shell**, this step is not necessary

You can execute the following to achieve the same result as the one done via the Azure Portal:

**az group deployment create --name "ARMDeployment" --resource-group "RG-Template" --template-uri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vm-simple-linux/azuredeploy.json --verbose**

Or instead of typing interactively all required parameters, you can specify them on the command line:

**az group deployment create --name "ARMDeployment" --resource-group "RG-Template" --template-uri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vm-simple-linux/azuredeploy.json --verbose --parameters vmName=VM2 dnsLabelPrefix=aze-vm2 VmSize=Standard\_B1s adminUsername=demouser authenticationType=password adminPasswordOrKey="DemoPassword-2021"**

Let’s use the second command

*Please note that the above commands are marked for deprecation and we should use* ***az deployment group*** *instead*

There is also an option to store them in a JSON file either local or remote

The same applies to the template itself – it can be either local or remote

No matter how you specified them, you can return to the portal

Navigate to the resource group

Click on **Deployments** under **Settings**

And then click on the deployment you created, for example **ARMDeployment**

Once, the deployment is complete, you can navigate to the VM and examine how you can connect to it

Try to execute the above command in a loop. This way you can create multiple similar machines

## A Reminder

Don’t forget to delete (if you have any locks, remove them first) resources that you do not need anymore

Also remember to stop VMs that you are not going to use in the coming hours or days but will need them later

There is an **Auto-shutdown** option in the **Operations** section of every VM check if it set and if not, set it

You can enable the automatic shutdown, for example at 19:00. This will definitely save you some money

Please note, that the machines are not automatically started