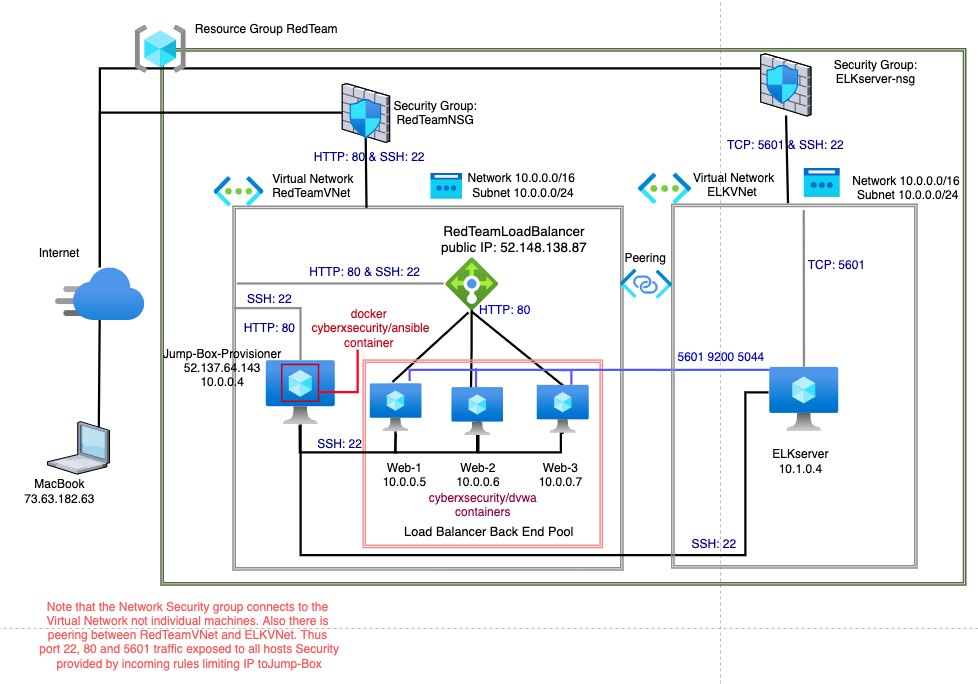
Azure Setup

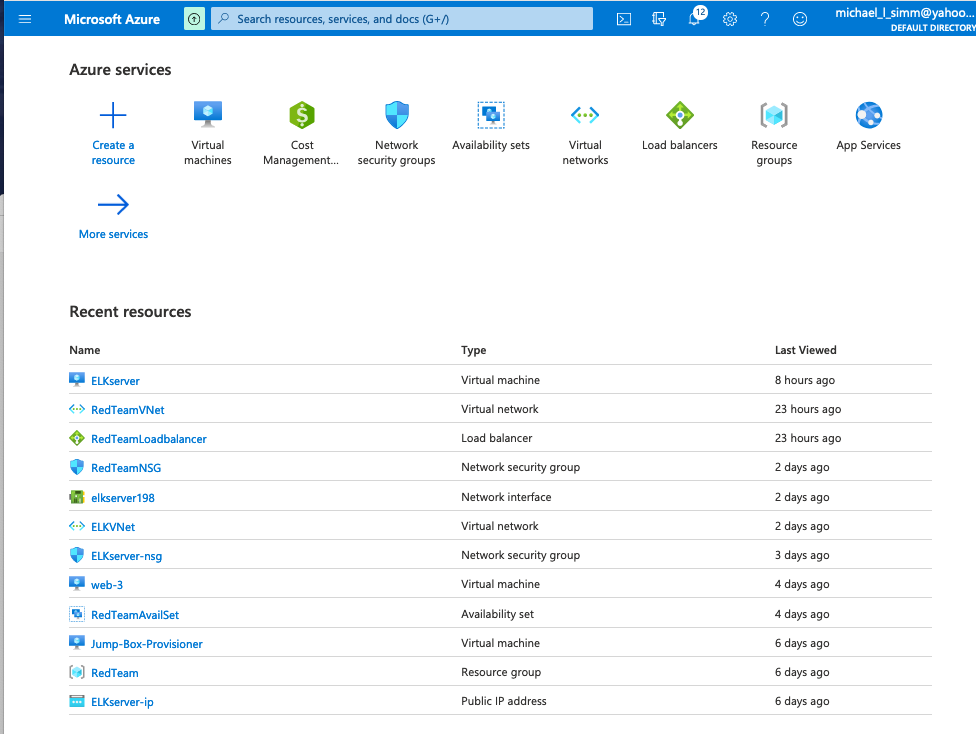
This sections documents specifics steps to create the Azure instance diagrammed below

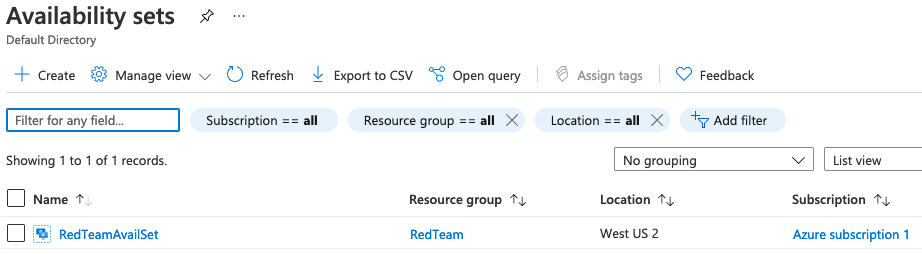


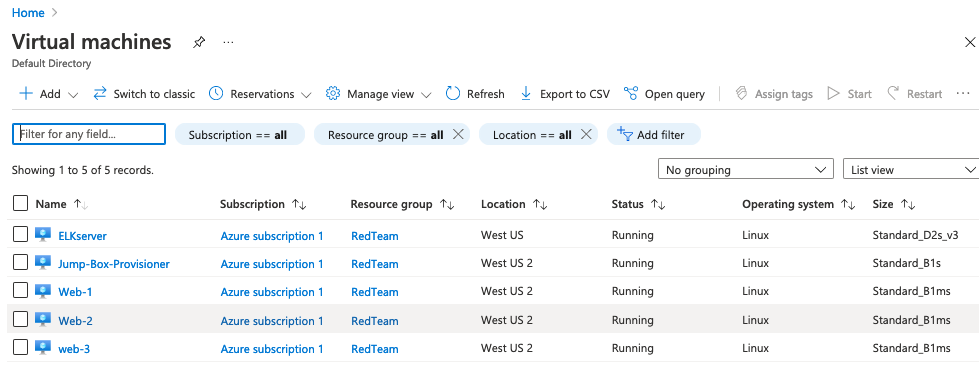
This used the Azure free $200/30 day option. This choice imposes certain constraints, and complicates the diagram. There is a max of 4 cpus per region. Consequently the elk server had to be put into a different region. That in turn required another Virtual network. The two Virtual Networks had to be peered. Building in the same region simplifies the architecture.

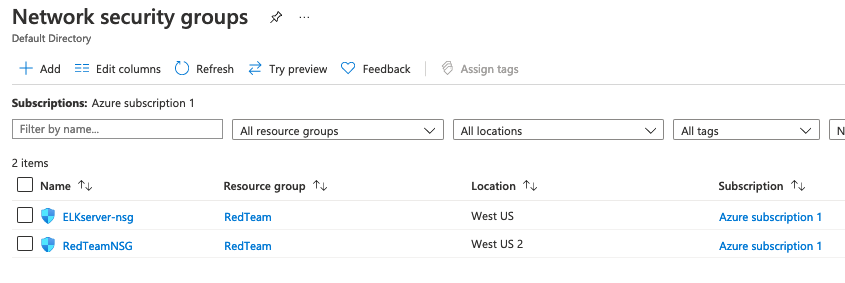
A free account on [portal.azure.com](http://portal.azure.com) was created with my personal email and credit card. All Azure info below from my [portal.azure.com](http://portal.azure.com) instance.

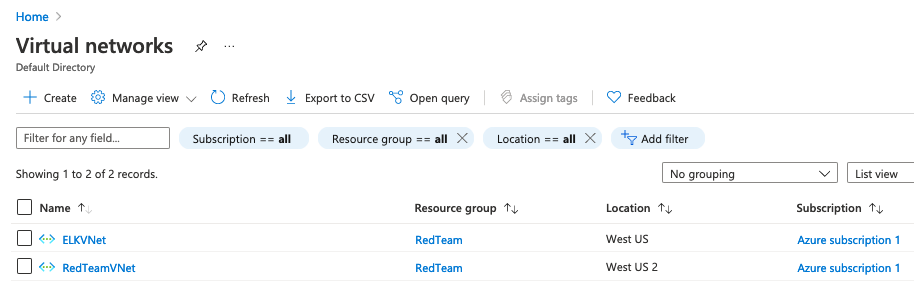
Azure Components: all components listed below at top level. More details and how to build them follows.

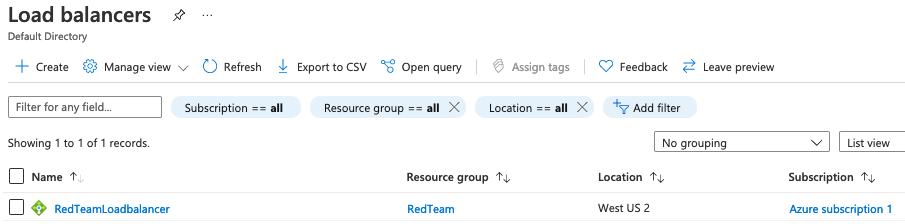
**All components in the RedTeam Resource Group**.

**Availability Sets:**

**Virtual Machines:**

**Network Security Groups**:

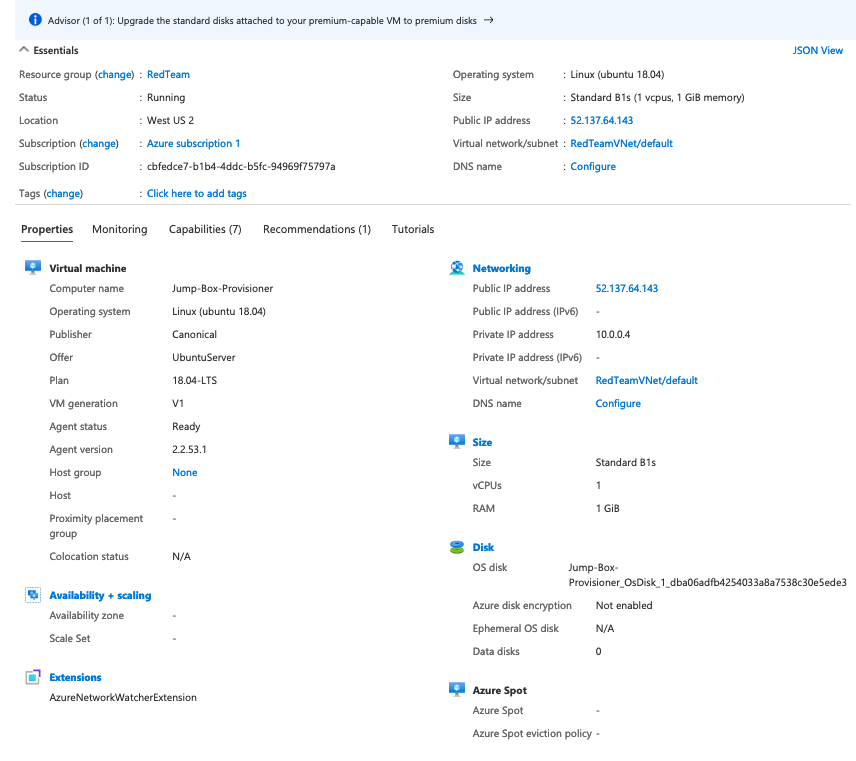
**Virtual Networks**:

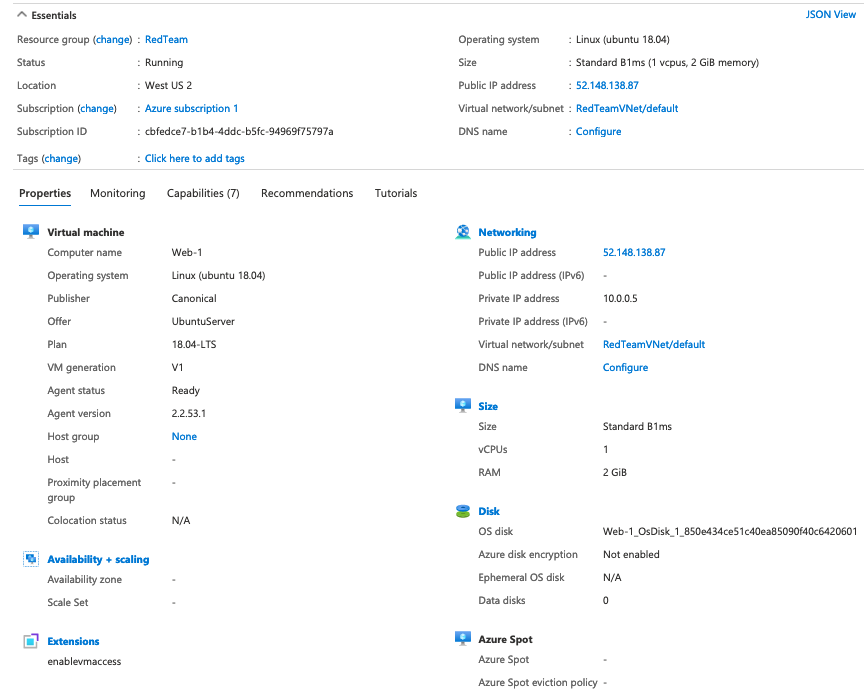
**Load Balancer**:

**Azure Components more detailed view**

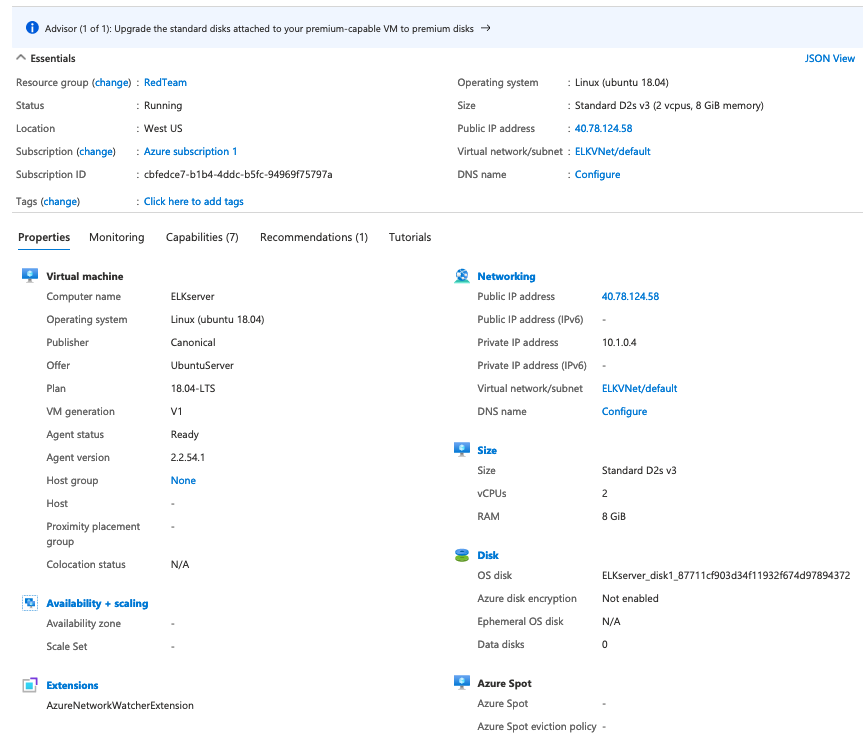
**Virtual Machines:**

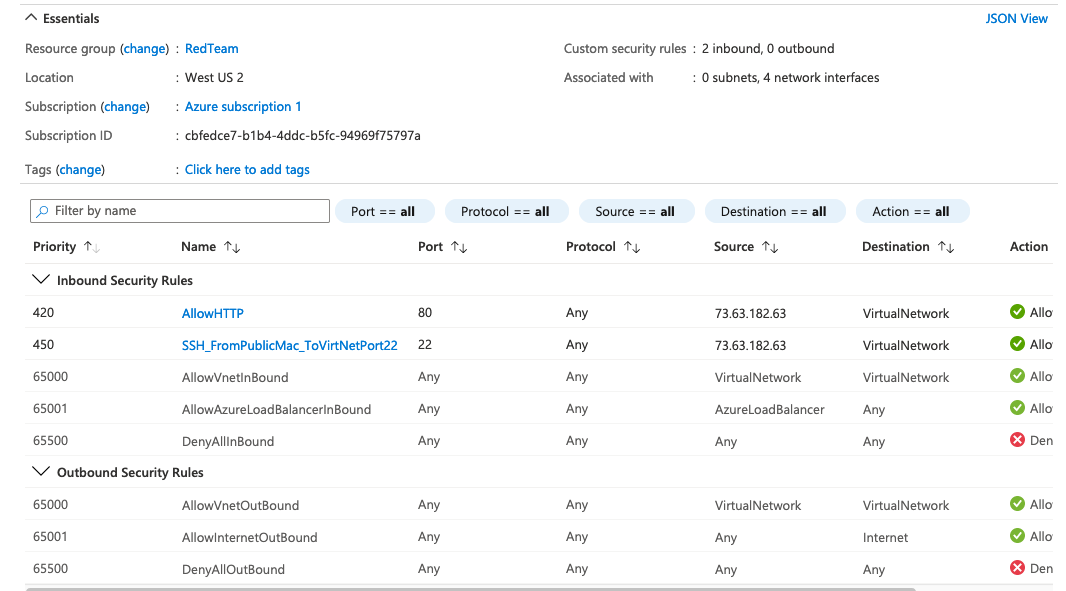
**Jump-Box-Provisioner**

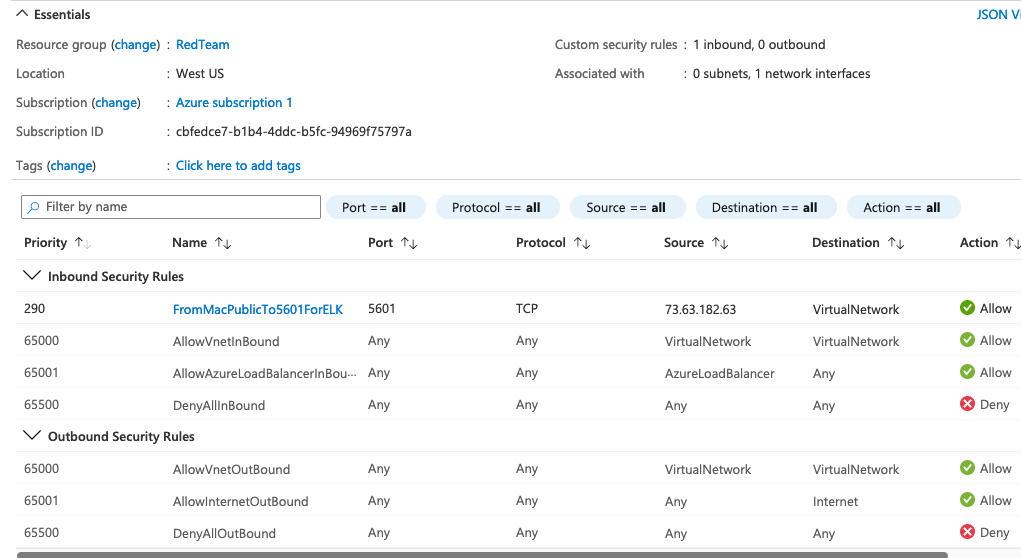
**Virtual Machines:**

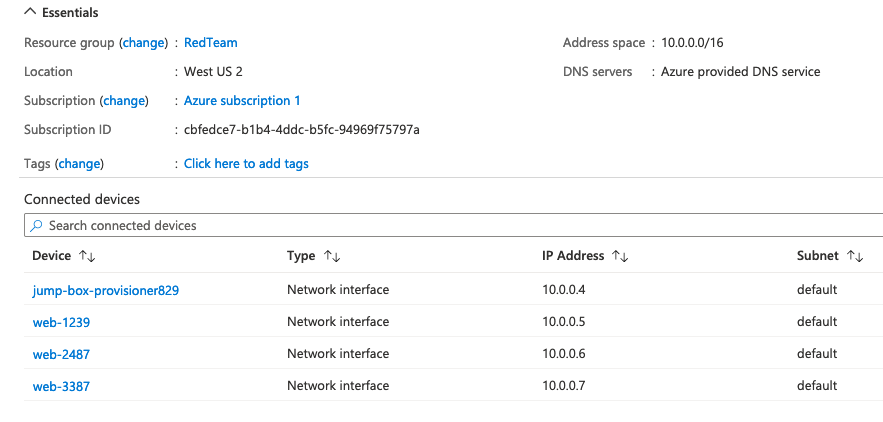
**Web-servers**. All three are the same except for name so only one shown here

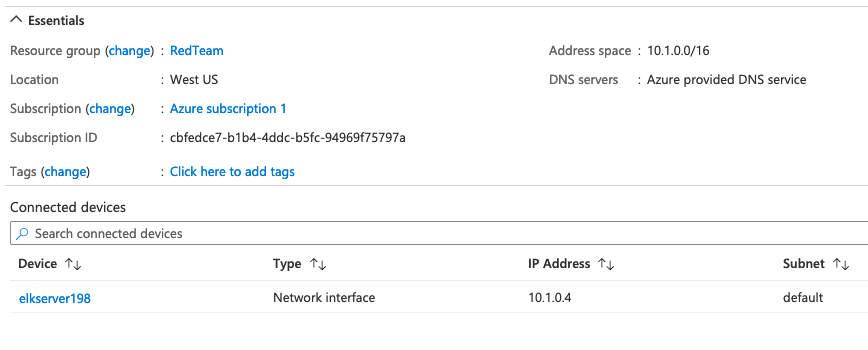
**Virtual Machines:**

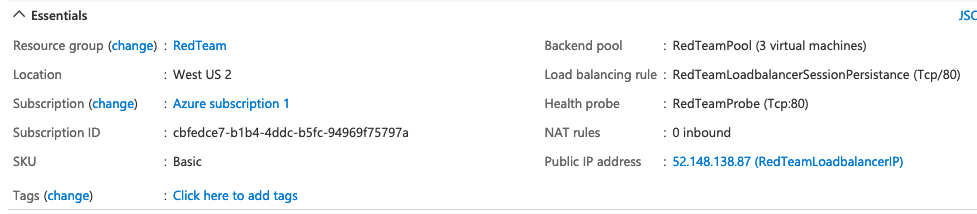
**ELK Server**

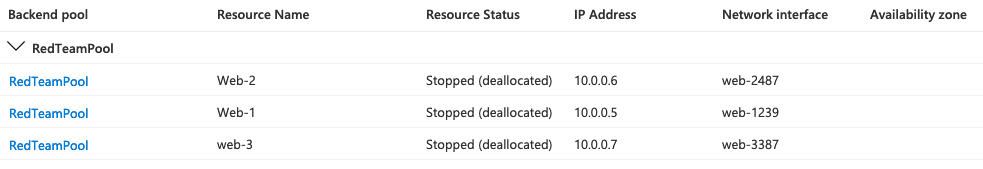
**Network Security Groups**: **RedTeamNSG**

**Network Security Groups: ELKserver-nsg**

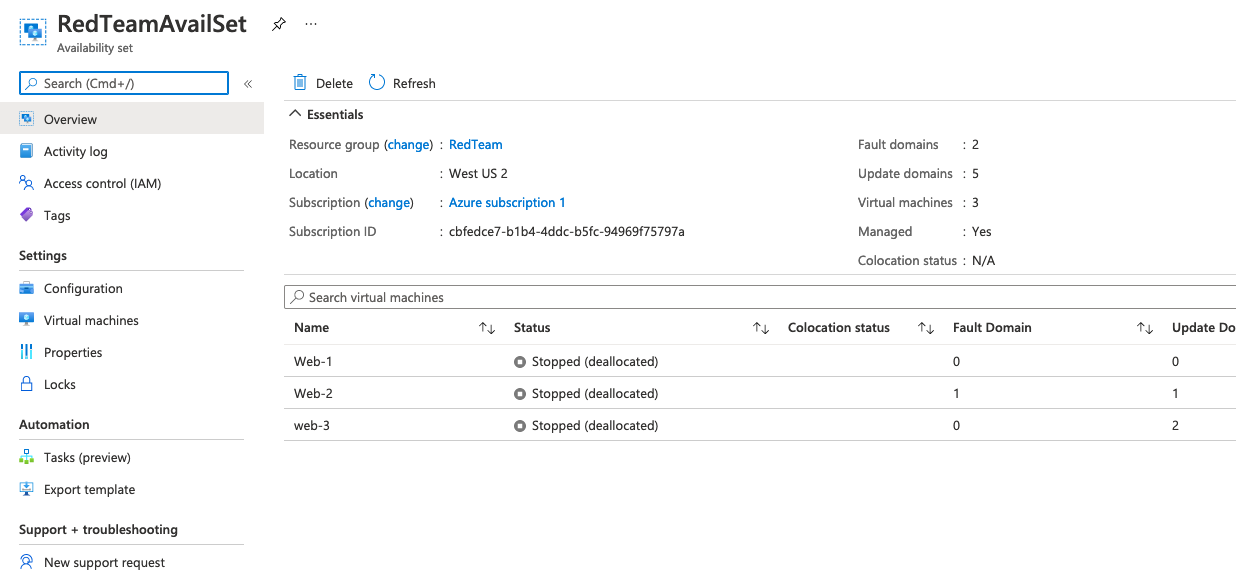
**Virtual Networks: RedTeamVNet**

**Virtual Networks: ELKVNet**

**Load Balancer: RedTeamLoadbalancer**

**Backend Pool:**

**Availability Set:**

****

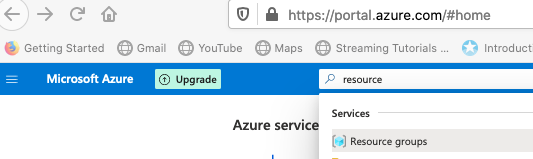
**Building Components:**

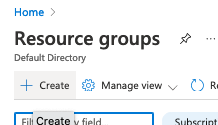
Step by step instructions for components. **Build in the order shown**

**Building the Resource Group.**

First thing that’s needed is creating a resource. From Home type resource in search box and click Resource Groups

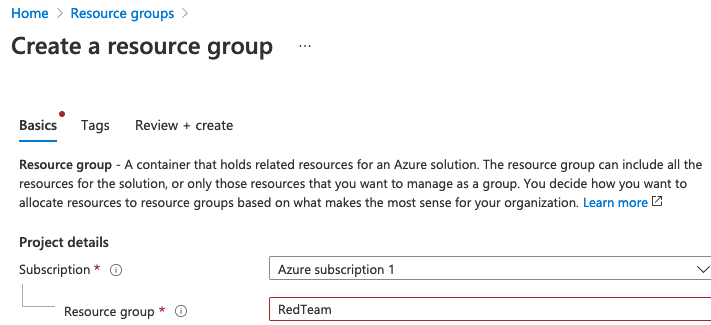
Click on “+ Create” icon.

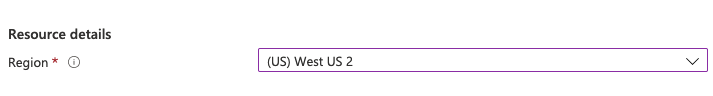




**Subscription**: Azure Subscription 1

**Resource Group**: RedTeam

**Region**: Use drop-down to set (US) West US 2



Then click blue Review + create button

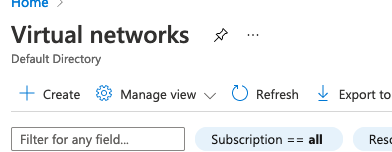
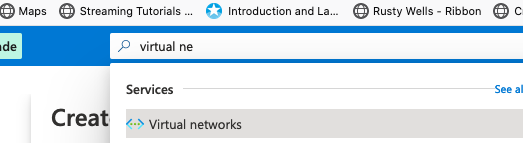
Review the info and if all looks correct click blue Create button

**Building the Virtual Networks**

Two are required due to the limitation of 4 cpus per region. First one is for web servers and jump box. 2nd one for ELK server.

**First VNet**

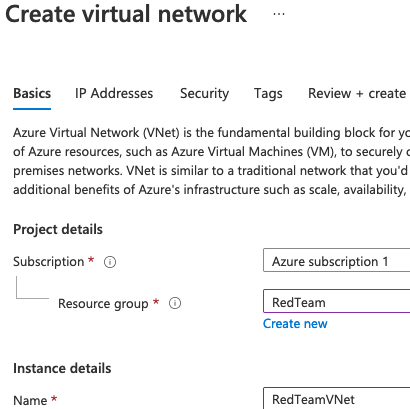
Type virtual net in search area and then click on Virtual Networks

Click on + Create icon

**Subscription**: Azure subscription 1

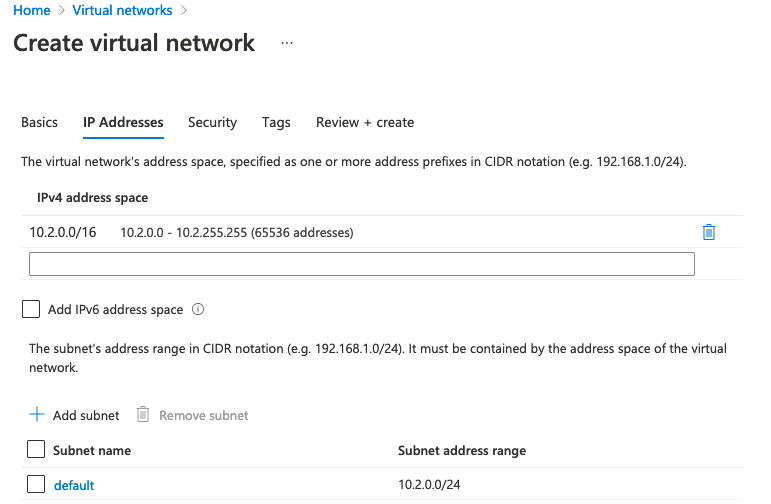
**Resource Group**: RedTeam

**Name**: RedTeamVNet (

**Region**: (US) West US 2 



Click Next: IP Addresses and use the defaults. Note screen shot below just for example. The first time I ran it to actually create the VNet the IP addresses were different.

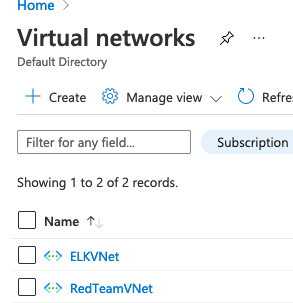


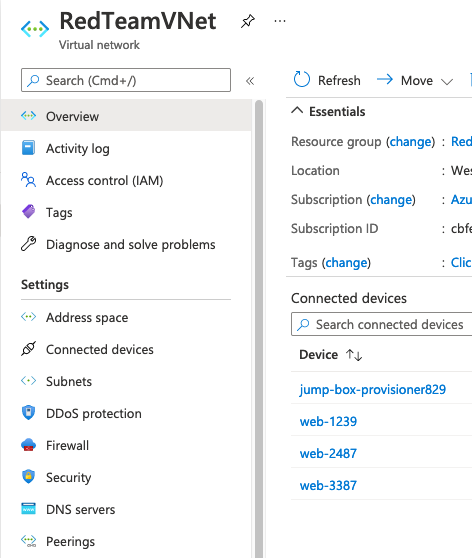
Security and Tags use the defaults so Click Review + create

Look over what appears. If any corrections are needed click < Previous. If all looks good click Create.

**2nd Virtual Network**. Repeat steps above except choose West US (not US 2) as the region and use the name ELKVNet

**Peering**: This allows communication between VM’s in both Virtual Networks.

From Home click on Virtual Networks. Then click on RedTeamVNet

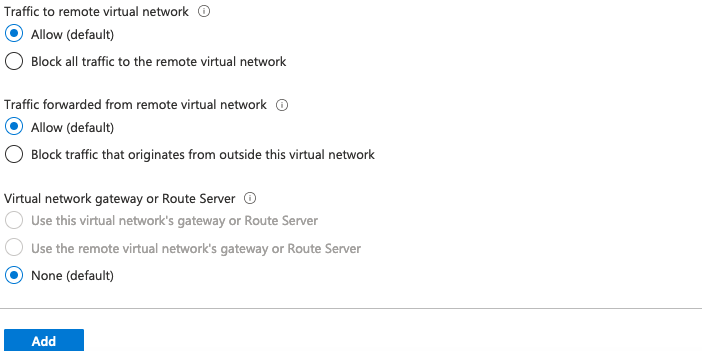
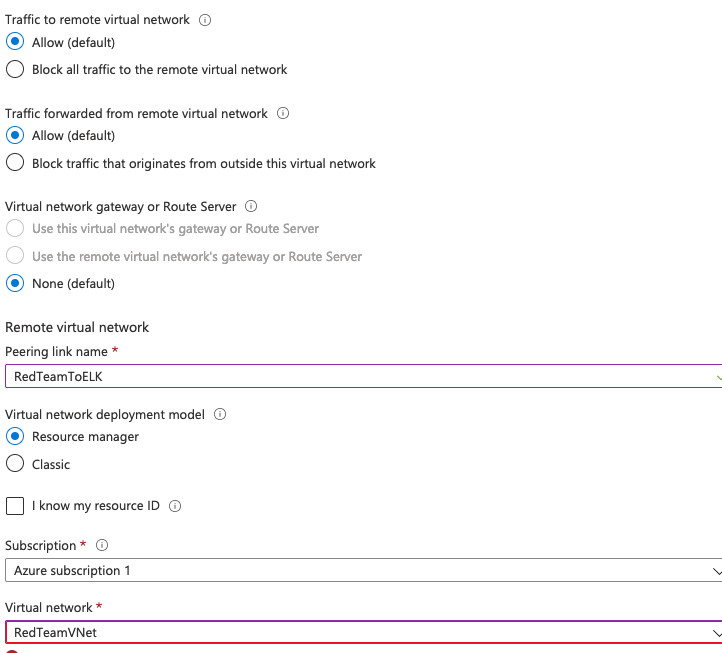
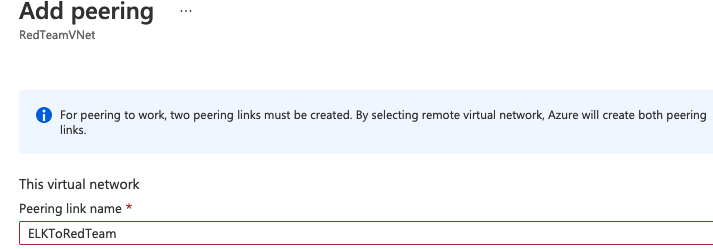
From there click on Peerings

From Peerings click + Add

This Virtual Network Peering link name: ELKToRedTeam

Remote Virtual Network Peering link name: RedTeamToELK

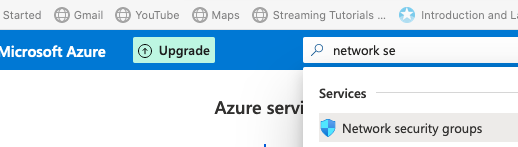
Then click Add to create the peering.



**Building the Network Security Groups (NSG).**

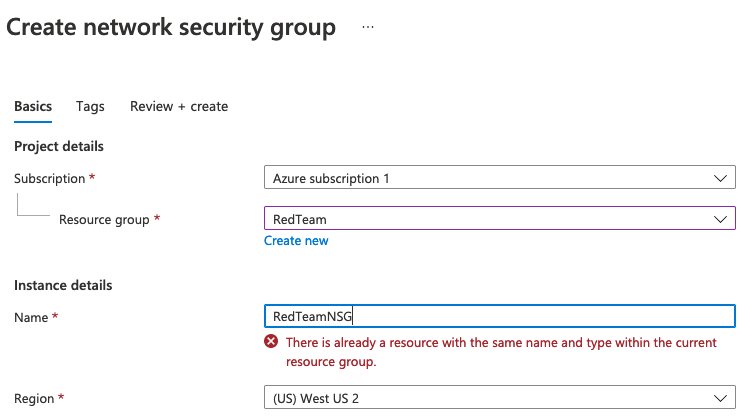
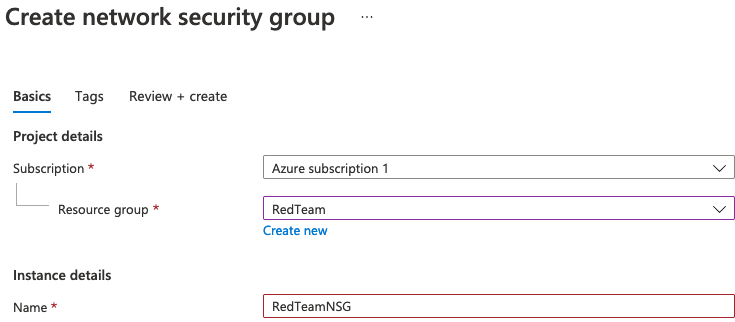
Here again two are required due to two regions.

**First NSG**

Type network se in search bar and click on Network Security Groups then click + Add iconSubscription: Azure subscription 1

**Resource Group**: RedTeam

**Name**: RedTeamNSG

**Region**: (US) West US 2

No tags are being used so click on Review + create

If validation passes and all looks correct click Create. Otherwise make any needed corrections.

**2nd NSG**

Repeat steps above with the following changes:

**Region** (US) West US

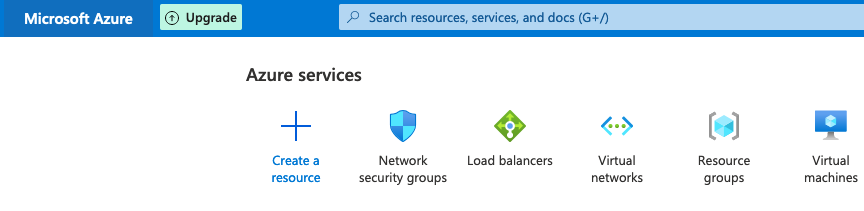
**Name**: ELKserver-nsg

The purpose of NSGs is to be able to create rules so the following need to be created with the documented steps.

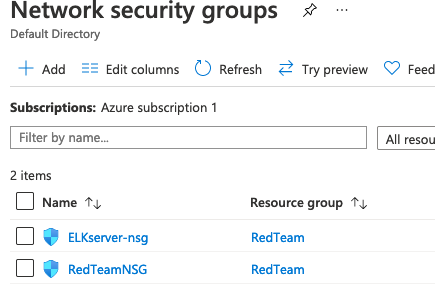
**RedTeamNSG Rules**

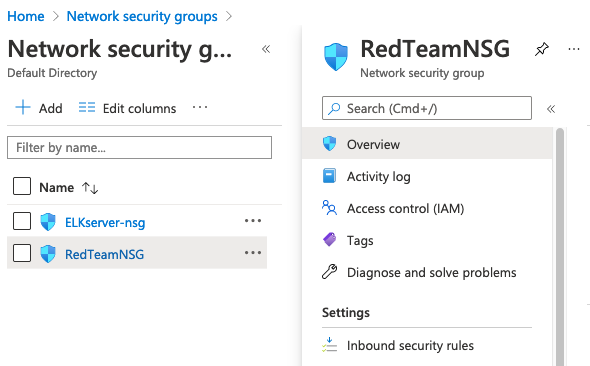
2 Inbound rules need to be added. SSH port 22 and HTTP port 80

Click Microsoft Azure to go to home page then click on Network security groups icon



Then click on RedTeamNSG link



Then click on Inbound Security Rules

Then click the + Add icon to add a rule.

Here are the steps to add SSH access through Port 22

**Source**: IP Addresses

**Source: IP addresses/CIDR ranges**: 73.63.182.63

**Source port ranges**: \*

**Destination**: VirtualNetwork

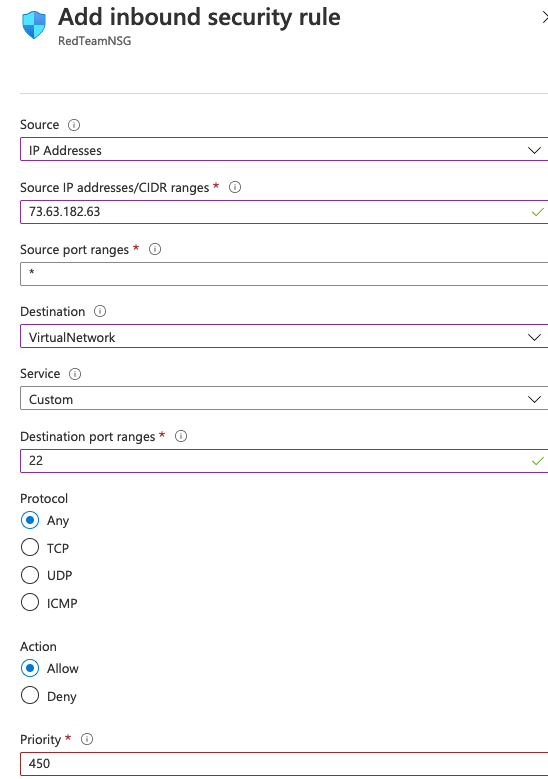
**Service**: Custom

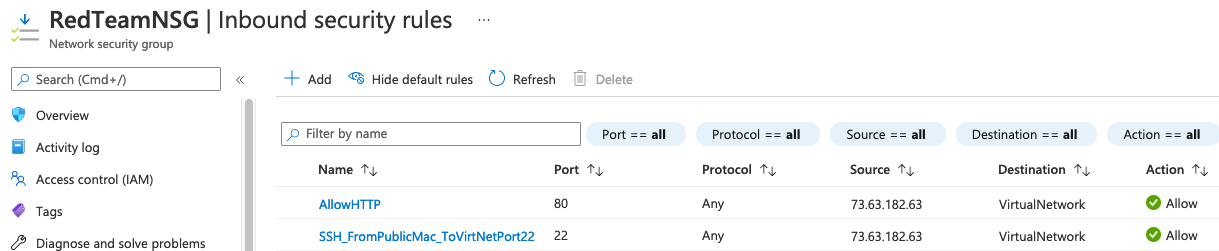
**Destination port ranges**: 22

**Protocol**: Any

**Action**: Allow

**Priority**: 450

**Name**: SSH\_From\_PublicMac\_ToVNetPort22

Then click Add button to add the rule. It should appear in the list of Inbound rules:

**RedTeamNSG 2nd rule**

Follow same procedure above with the below settings to add 2nd rule.

**Source**: IP Addresses

**Source IP addresses/CIDR ranges**: 73.63.182.63

**Source port ranges**: \*

**Destination**: VirtualNetwork

**Service**: Custom

**Destination port ranges**: 80

**Protocol**: Any

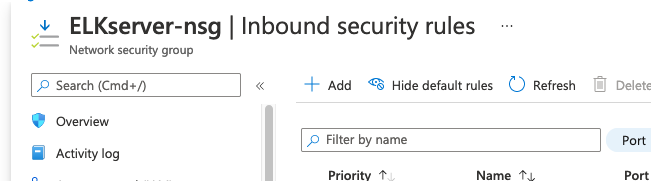
**Action**: Allow

**Priority**: 420

**Name**: AllowHTTP

**ELKserver-nsg rule**

From Network security group page click on ELKserver-nsg to bring up that NSG

That will bring you to the ELKserver-nsg Inbound Security rules page

Follow same process to add one rule with the following settings:

**Source**: IP Addresses

**Source IP addresses/CIDR ranges**: 73.63.182.63

**Source port ranges**: \*

**Destination**: VirtualNetwork

**Service**: Custom

**Destination port ranges:** 5601

**Protocol**: TCP

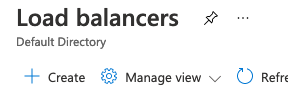
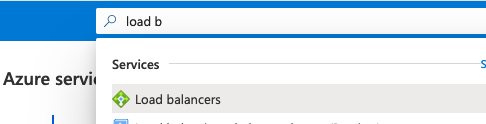
**Action**: Allow

**Priority**: 290

**Name**: FromMacPublicTO5601ForELK

**Building the Load Balancer**

Type load bal into search box and click on Load Balancers then click on + Creat icon



Subscription: Azure subscription 1

Resource Group: RedTeam

Name: RedTeamLoadbalancer

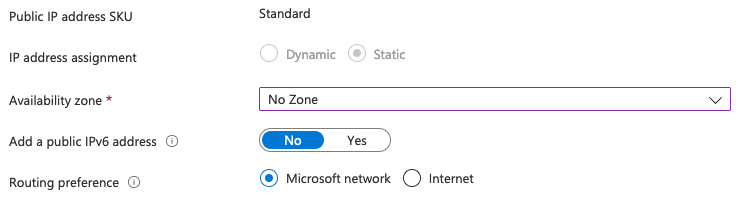
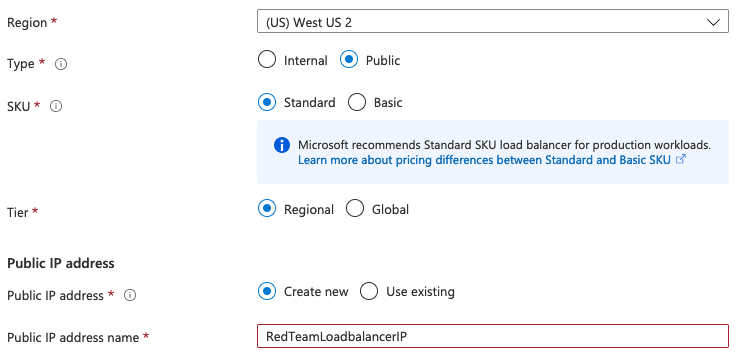
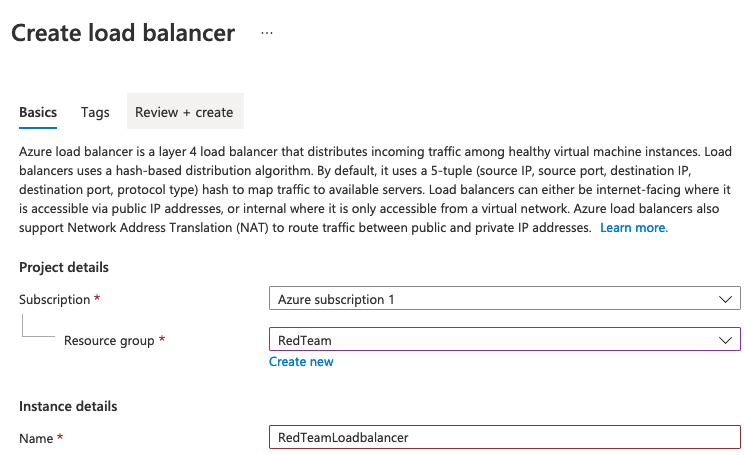
Region: (US) West US 2

**Type**: Public

**SKU**: Internal

**Availability zone:** No zone

**Public IP address:** Create new

**Public IP address name:** RedTeamLoadbalancerIP

Click Review + create

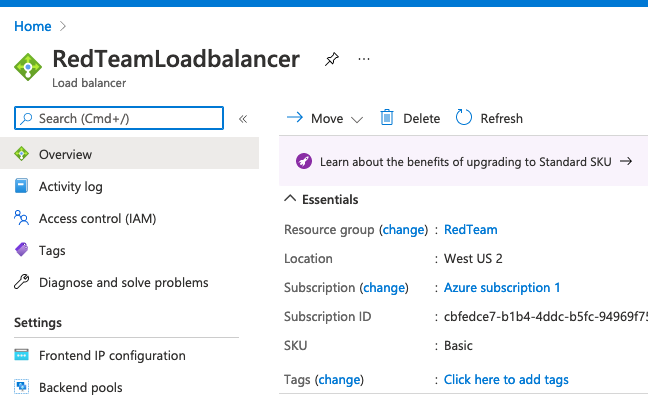
If all looks good click Create.

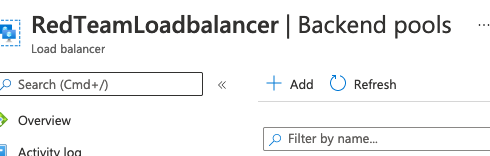
**Backend Pool**

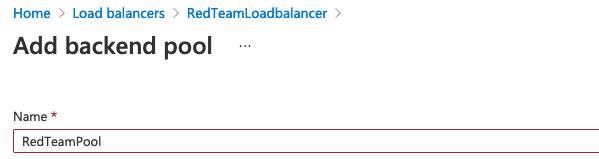
This is how to add the Virtual Machines to the backend pool. **Cannot be done until the Virtual Machines are created**! Make sure to do this at that point!

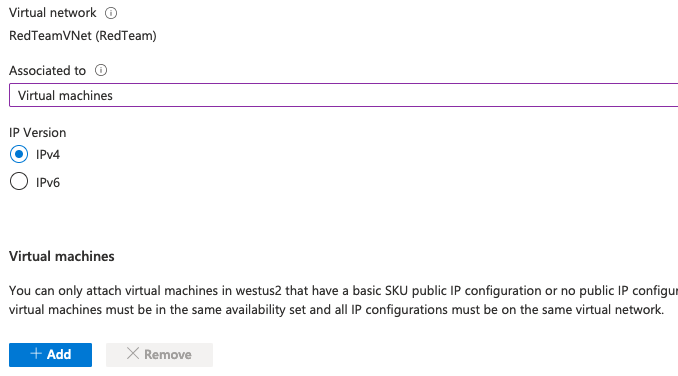
**Name**: ReadTeamPool

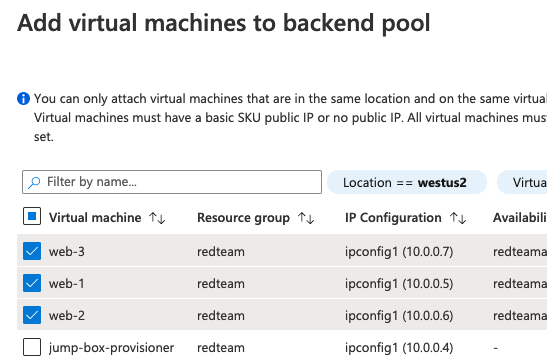
**Associated to:** Virtual machines

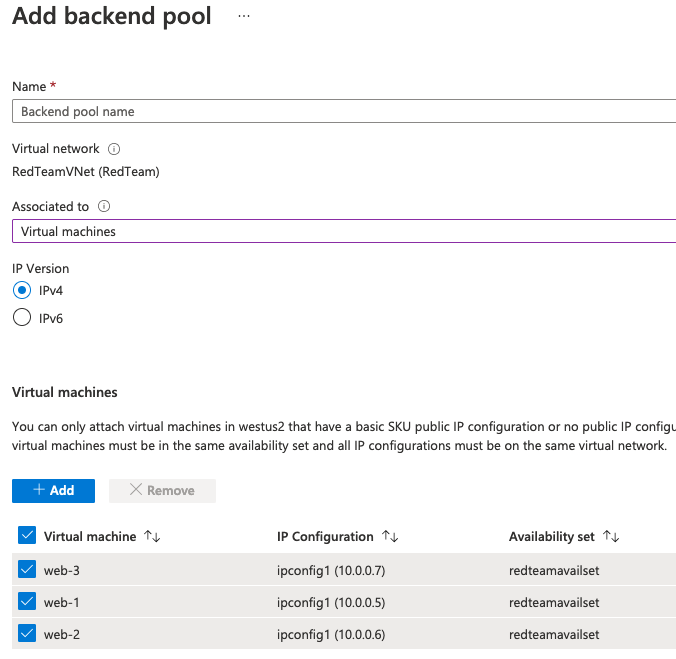
Click on Backend Pools. Then + Add Icon



From below click Add to choose which VMs to add



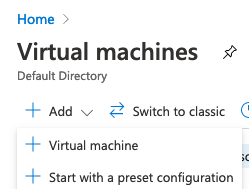
Select the 3 web servers and click Add

Then select 3 web servers again and click Add

**Building Virtual Machines—Only do this after all other components setup**

Some of the Azure building needs to wait until container work is done on the Jump-Box. So build the Jump-Box here and then go to the docker/ansible section to generate SSH keys for the web and ELK servers

**Building Jump-Box**

On Virtual machines click Add to bring up Add window. Click + Virtual Machine

This brings up the Create a virtual machine page.

**Settings Basics:**

**Subscription**: Azure subscription 1

**Virtual Machine Name:** Jump-Box-Provisioner

**Region**: (US) West US 2

**Availability Options**: No infrastructure redundancy required

**Image**: Ubuntu Server 18.04

**Size**: Standard\_B1s - 1 vcpu, 1 GiB memory

**Authentication Type**: SSH public key

**Username**: RedTeamAdmin

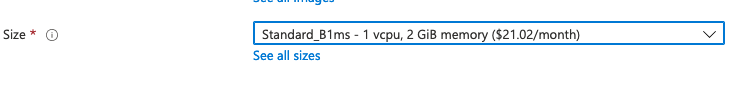
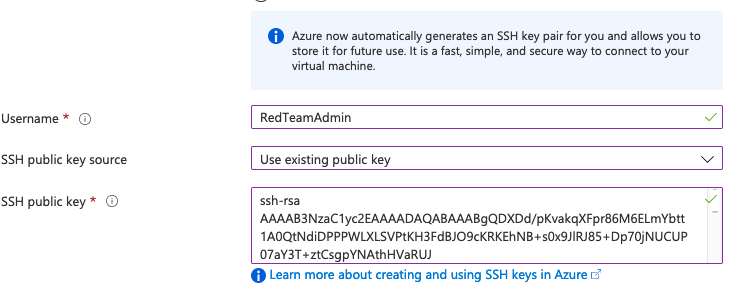
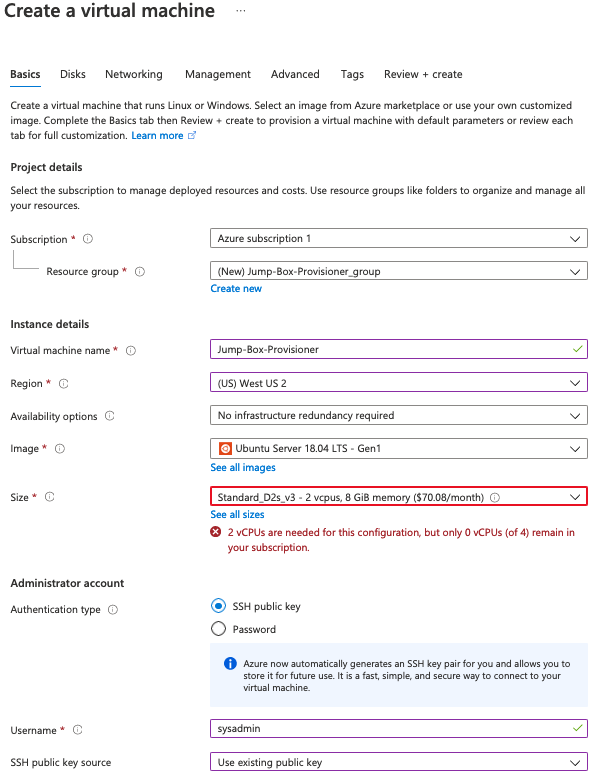
**SSH public key source**: Use existing public key

**SSH public key**: Need to paste in the key created per below

**Public inbound ports**: Don’t make any changes. Will be over-written when security group chosen.

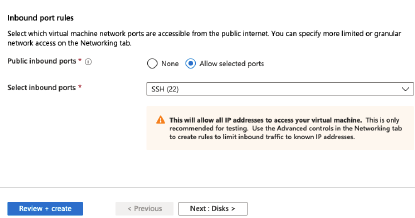
**SSH public key process:** Do as below on the local PC which accesses the Jump-Box. In this case my Mac. Copy and paste the public key into SSH public key text box as part of the Jump-Box build below .

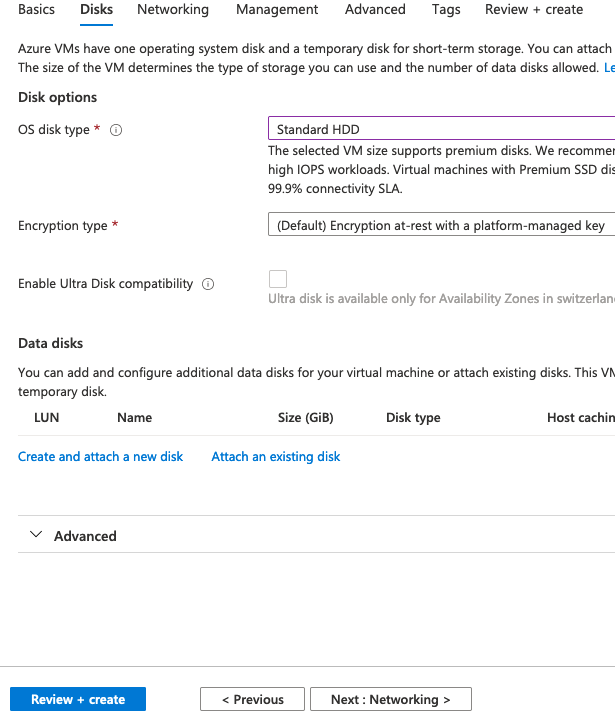
****



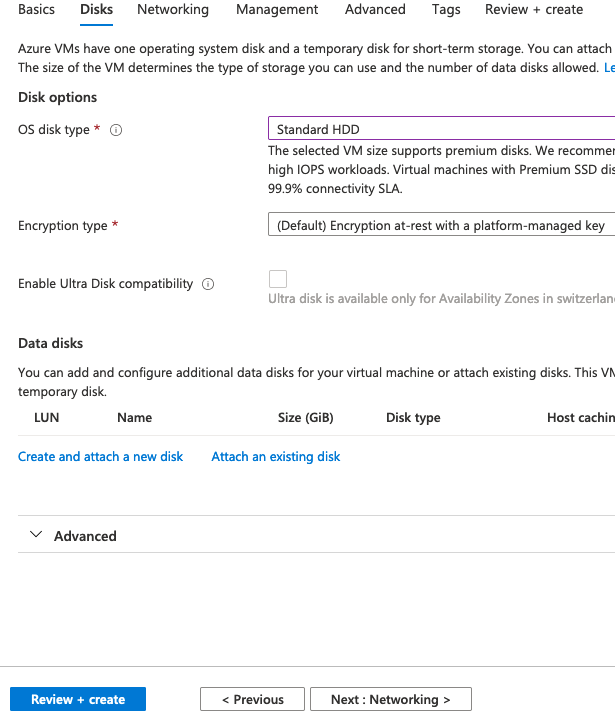
**Settings** **Disks**:

Click Next:Disks > button at bottom to setup disks

Choose Standard HDD and click Next: Networking at bottom



**Settings**



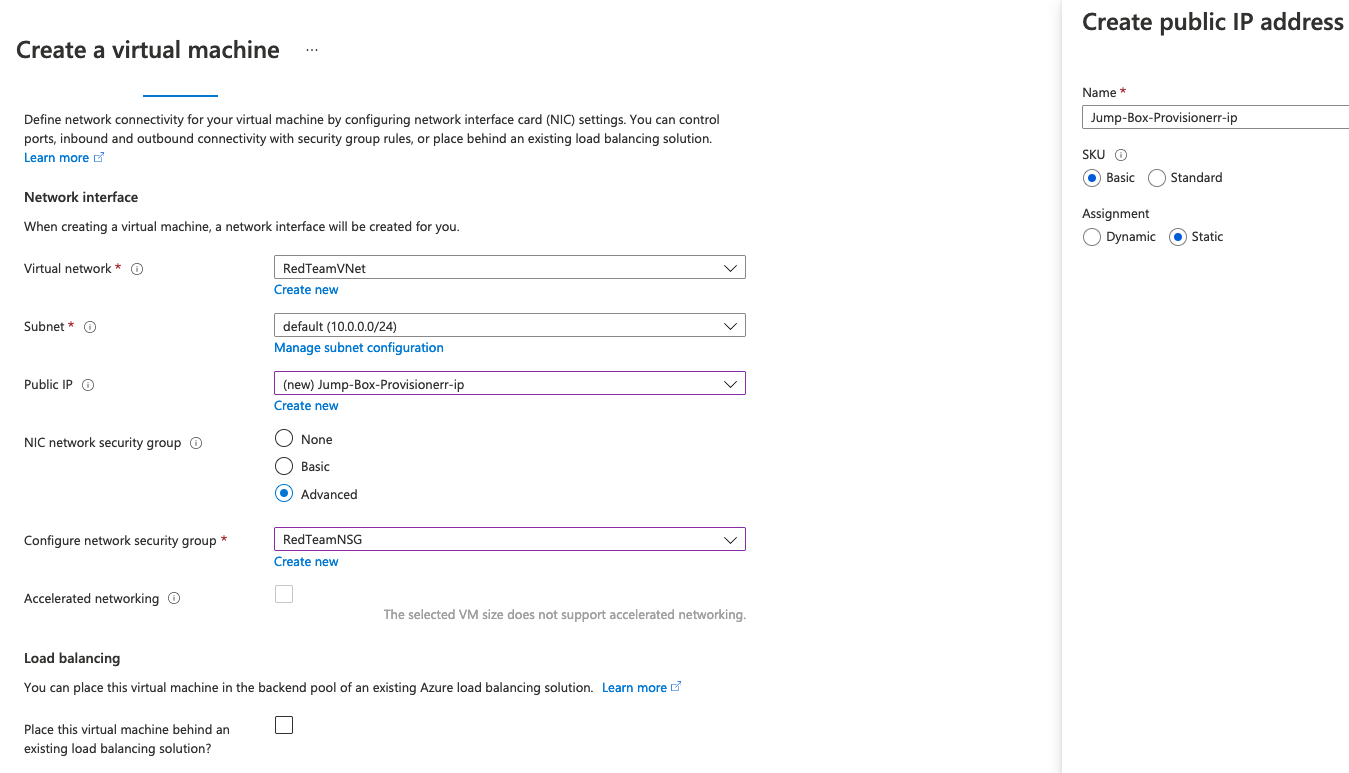
Click Next: Networking

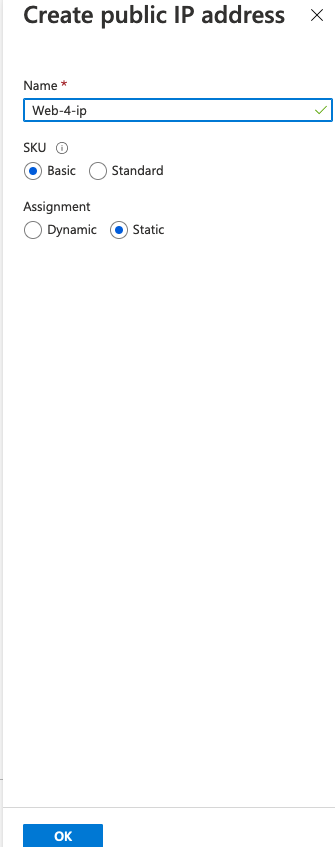
**Virtual network:** RedTeamVnet

**Subnet: default** (10.0.0.0/24)

**Public IP**: (new Jump-Box-Provisioner-ip—**Click create new and choose** Static in Create Public IP address Panel that appears

**NIC network security group:** Advanced

**Configure network security group:** RedTeamNSG



Click on Review + create.

Look over the configuration. If it looks good click Create. If anything Neds to be modified, make the necessary changes. Then when all looks good click Create

Look over the results. If OK click Create. Otherwise make any needed fixes.

***Before building the VM’s for Web Server make sure the public SSH key generation procession in docker/ansible section has been completed. That section may completed after the Jump-Box VM has been created as above.***

**Building Web Server VMs**

Three will be built but the entire process for building each is the same except for the name. Thus it’s only documented once.

Follow the same steps as creating the Jump-Box VM using the following settings:

**Settings Basics:**

**Subscription:** Azure subscription 1

**Virtual Machine Name:** Web-1/2/3

**Region**: (US) West US 2

**Availability Options:** Availability set per below

**Image**: Ubuntu Server 18.04

**Size**: Standard\_B1ms - 1 vcpu, 2 GiB memory

**Authentication Type**: SSH public key

**Username**: sysadmin

**SSH public key source**: Use existing public key

SSH public key: Need to paste in the key created in docker container. **This should not be the key from the local machine. It needs to be from the Jump-Box container.**

**Public inbound ports**: Don’t make any changes. Will be over-written when security group chosen.

**Settings Disks:**

OD disk type: Standard HDD

**Settings Networking:**

**Virtual network:** RedTeamVnet

**Subnet: default** (10.0.0.0/24)

**Public IP**: None

**NIC network security group:** Advanced

**Configure network security group:** RedTeamNSG

Review and Create this VM if all settings correct.

**Building ELK Server VM**

Due to limitations of free subscription’s limit of 4 cpus per region this server has some different settings.

Follow the same steps as creating the other VMs using the following settings:

**Settings Basics:**

**Subscription:** Azure subscription 1

**Virtual Machine Name:** ELKserver

**Region**: (US) West US

**Availability Options:** Availability set per below

**Image**: Ubuntu Server 18.04

**Size**: D2s\_V3 - 2 vcpus, 8 GiB memory

**Authentication Type**: SSH public key

**Username**: sysadmin

**SSH public key source**: Use existing public key

SSH public key: Need to paste in the key created in docker container. **This should not be the key from the local machine. It needs to be from the Jump-Box container.**

**Public inbound ports**: Don’t make any changes. Will be over-written when security group chosen.

**Settings Disks:**

OD disk type: Standard HDD

**Settings Networking:**

**Virtual network:** ELKVnet

**Subnet: default** (10.1.0.4/24)

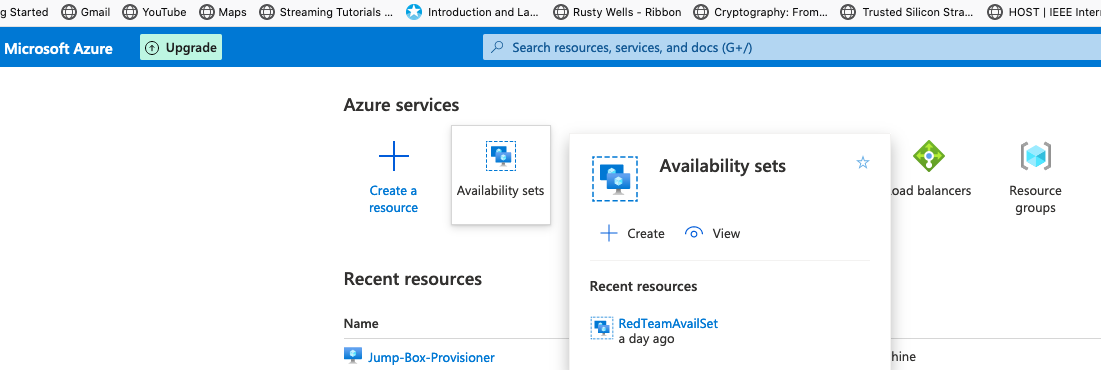
**Public IP**: None

**NIC network security group:** Advanced

**Configure network security group:** RedTeamNSG

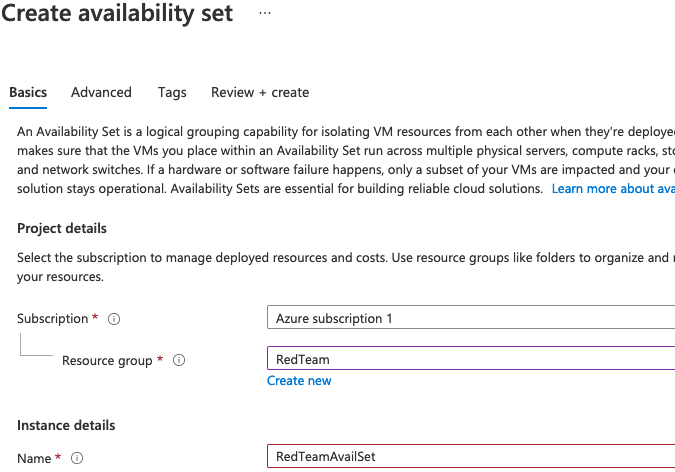
**Availability Set**

Build as follows

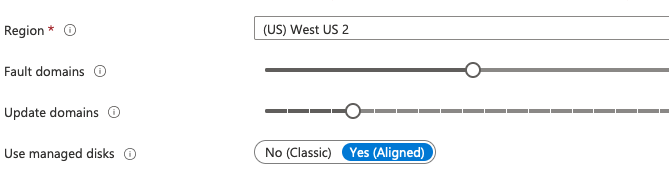
From Home click on Availability sets -> + Create 

**Resource Group**: RedTeam

**Name**: RedTeamAvailSet

Region: (US) West US 2

Click Review + create. If all looks good click Create

**Adding VMs**

Go to the availability set just created and add the three web servers to it.