Exercise - Configure network settings

- 10 minutes

Let's apply a network security group to our network, so that we only allow HTTP traffic through our server.

Login to the Azure Portal

1. Open the Azure portal in a browser.

2. Sign into Azure using the Microsoft account email address and password you created for this session.

Create a network security group

Azure should have created a security group for us because we indicated we wanted SSH access. But let's create a new security group, so you can walk through the entire process. This is particularly important if you decide to create your virtual network *before* your VMs. As mentioned earlier, security groups are *optional* and not necessarily created with the network.

1. In the Azure portal, click the **Create a resource** button in the left-corner sidebar to start a new resource creation.

2. Type **Network security group** into the filter box and select the matching item in the list.

3. Make sure the **Resource Manager** deployment model is selected and click **Create**.

4. Provide a **Name** for your security group. Again, naming conventions are a good idea here. Let's use **test-web-nsg1** for **Test Web Network Security Group #1**.

5. Select the proper **Subscription** and use your existing **Resource group** named **labvms**.

6. Finally, select the same **Location** as the VM / virtual network. This is important; you won't be able to apply this resource if it's in a different location.

7. Click **Create** to create the group.

Add new inbound rules to our network security group

Deployment should complete quickly. When it's finished, we can add new rules to our security group:

1. Find the new security group resource and select it in the Azure portal.

2. On the overview page, you'll find that it has some default rules created to lock down the network.

   On the inbound side:

   - All incoming traffic from one VNet to another is allowed. This lets resources on the VNet talk to each other.
   - Azure Load Balancer **probe** requests to ensure the VM is alive.
   - All other inbound traffic is denied.

On the outbound side:

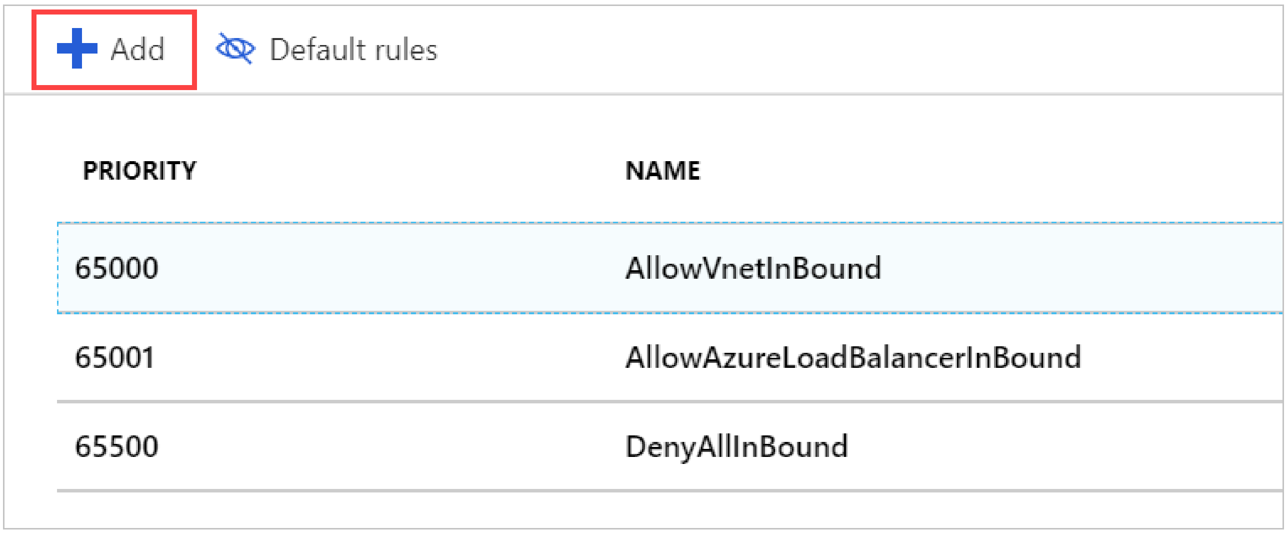
- All in-network traffic on the VNet is allowed.
- All outbound traffic to the internet is permitted.
- All other outbound traffic is denied.

Note These default rules are set with high-priority values, which means that they get evaluated *last*. They cannot be changed or deleted, but you can *override* them by creating more specific rules to match your traffic with a lower priority value.

3. Click the **Inbound security rules** section in the **Settings** panel for the security group.

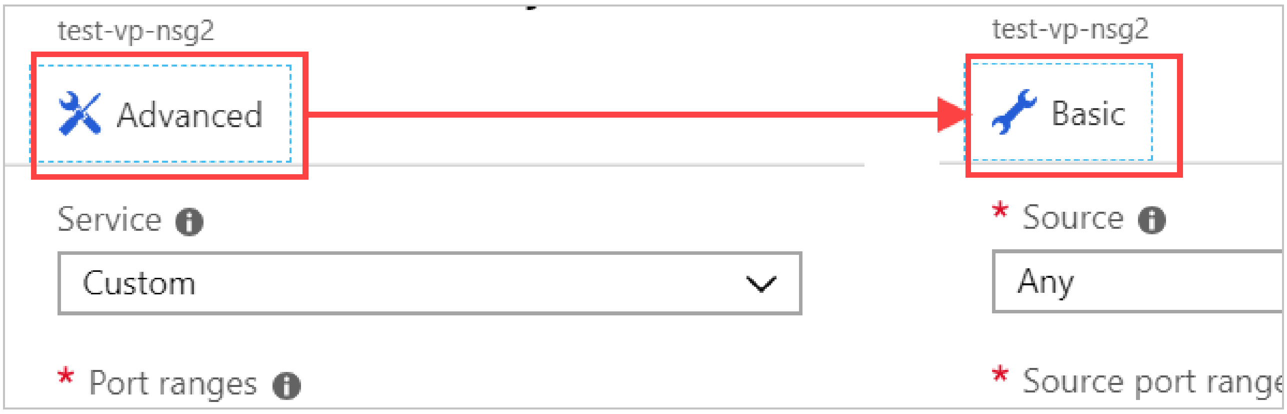
Add http inbound rules to our network security group

1. Click **+ Add** to add a new security rule.



PRIORITY	NAME
65000	AllowVnetInBound
65001	AllowAzureLoadBalancerInBound
65500	DenyAllInBound

There are two ways to enter the information necessary for a security rule: basic and advanced. You can switch between them by clicking the button at the top of the **add** panel.



Service: Custom

* Port ranges

* Source: Any

* Source port range

The advanced mode provides the ability to customize the rule completely. However, if you need to configure a known protocol, the basic mode is a bit easier to work with.

2. Switch to the **Advanced** mode.

3. Add the information for our HTTP rule:

   - Set the **Service** to be HTTP. This will set your port range up for you.

- Set the **Priority** to **1000**. It has to be a lower number than the default **Deny** rule. You can start the range at any value, but it's recommended that you give yourself some space in case an exception needs to be created later.
- Give the rule a name; we'll use **allow-http-traffic**.
- Give the rule a description.

4. Switch between the **Advanced** and **Basic** modes. Notice that our settings are still present. We can use these panels to create more fine-grained settings. In particular, we would likely restrict the **Source** to be a specific IP address or range of IP addresses specific to the cameras. Leave the setting as **Any**, so you can test the rule.

5. Click **Add** to create the rule. This will update the list of inbound rules - notice they are in priority order, which is how they will be examined.

Add ssh inbound rules to our network security group

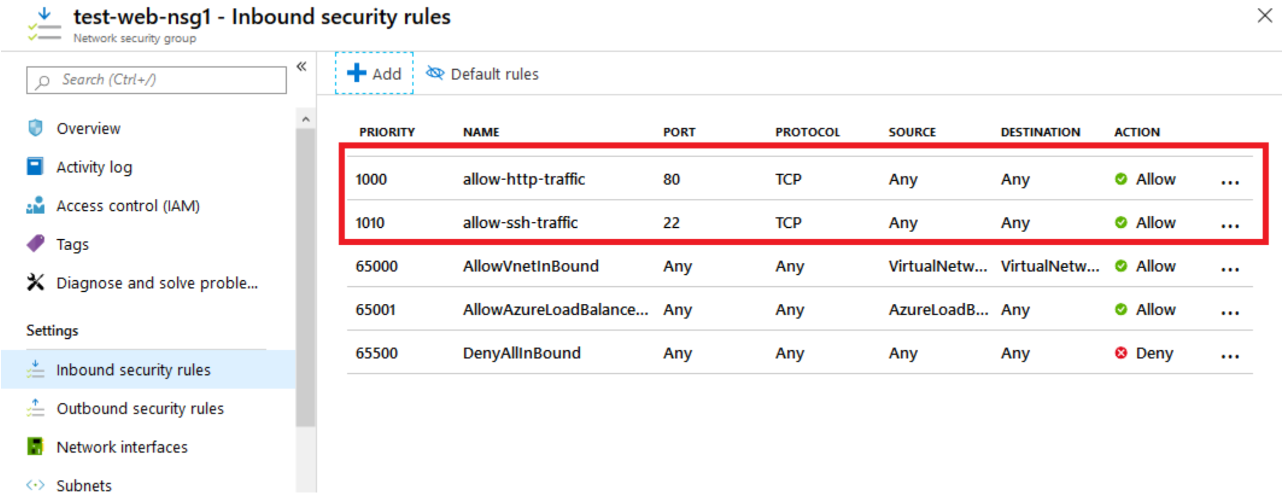
1. Click **+ Add** to add a new security rule.

2. Switch to the **Advanced** mode.

3. Add the information for our SSH rule:

   - Set the **Service** to be SSH. This will set your port range up for you.
   - Set the **Priority** to **1010**.
   - Give the rule a name; we'll use **allow-ssh-traffic**.
   - Give the rule a description.

4. Click **Add** to create the rule. This will update the list of inbound rules - notice they are in priority order, which is how they will be examined.



PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
1000	allow-http-traffic	80	TCP	Any	Any	Allow
1010	allow-ssh-traffic	22	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetw...	VirtualNetw...	Allow
65001	AllowAzureLoadBalance...	Any	Any	AzureLoadB...	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Apply the security group

Recall that we can apply the security group to a network interface to guard a single VM or to a subnet where it would apply to any resources on that subnet. The latter approach tends to be the most common, but we are going to replace the NSG assigned to the VM. We could get to this resource in Azure through either the virtual network resource or indirectly through the VM that is using the virtual network.

1. Navigate to the **Overview** panel for the virtual machine named *test-linux-vm1*. You can find the VM under **All Resources**.

2. Select the **Networking** item in the **Settings** section.

3. In the networking properties, you will find information about the networking applied to the VM, including the **Network Interface**. This is a clickable link to get to the resource. Click it to open the network interface.

4. In the **Settings** section, select the **Network security group** item, and click **Edit**.

5. Click the **Network security group** entry.

6. Select your new security group: **test-web-nsg1**. There should be another group here as well that was created with the VM.

7. Click **Save** to save the change. It will take a minute to apply to the network.

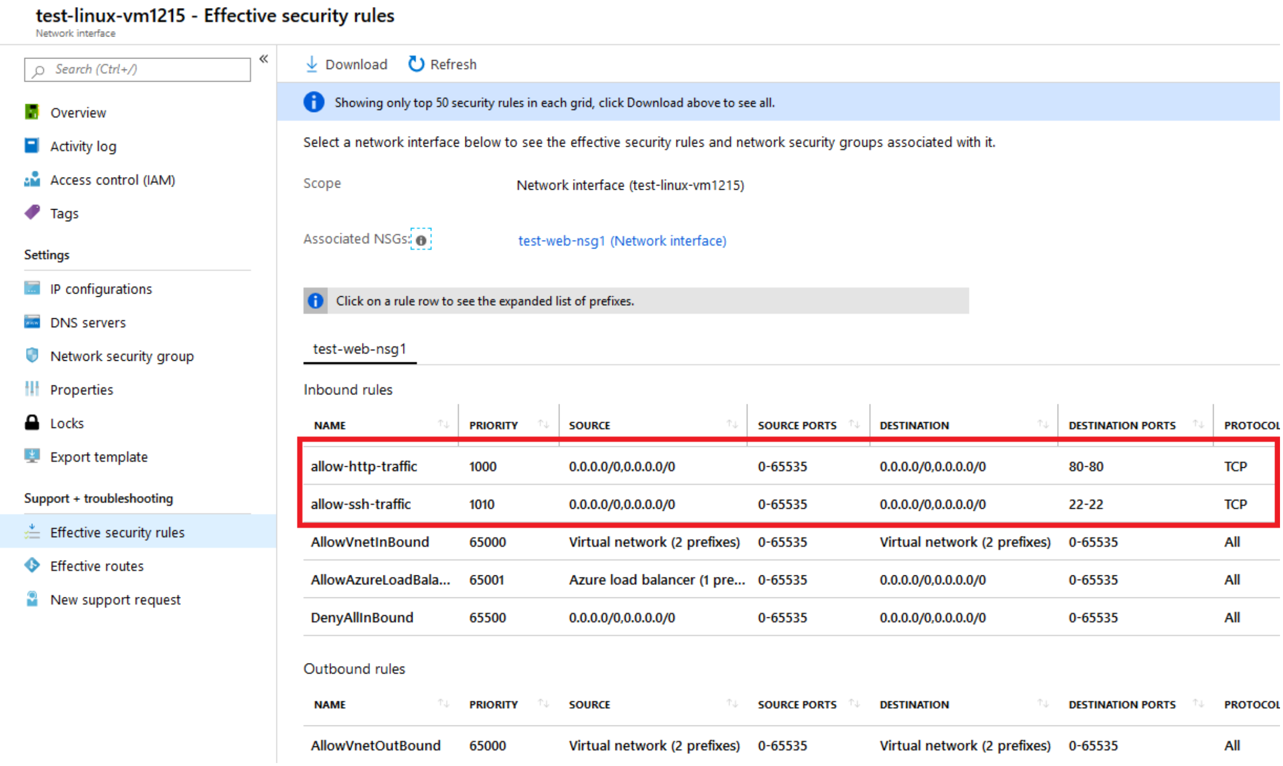
Verify the rules

Let's validate the change:

1. Switch back to the **Overview** panel for the virtual machine. You can find the VM under **All Resources**.

2. Select the **Networking** item in the **Settings** section.

3. In the network interface details, there is a link for **Effective security rules** that will quickly show you how rules are going to be evaluated. Click the link to open the analysis and make sure you see your new rules.



test-linux-vm1215 - Effective security rules

Network interface

Showing only top 50 security rules in each grid, click Download above to see all.

Select a network interface below to see the effective security rules and network security groups associated with it.

Scope Network interface (test-linux-vm1215)

Associated NSGs: test-web-nsg1 (Network interface)

Click on a rule row to see the expanded list of prefixes.

test-web-nsg1

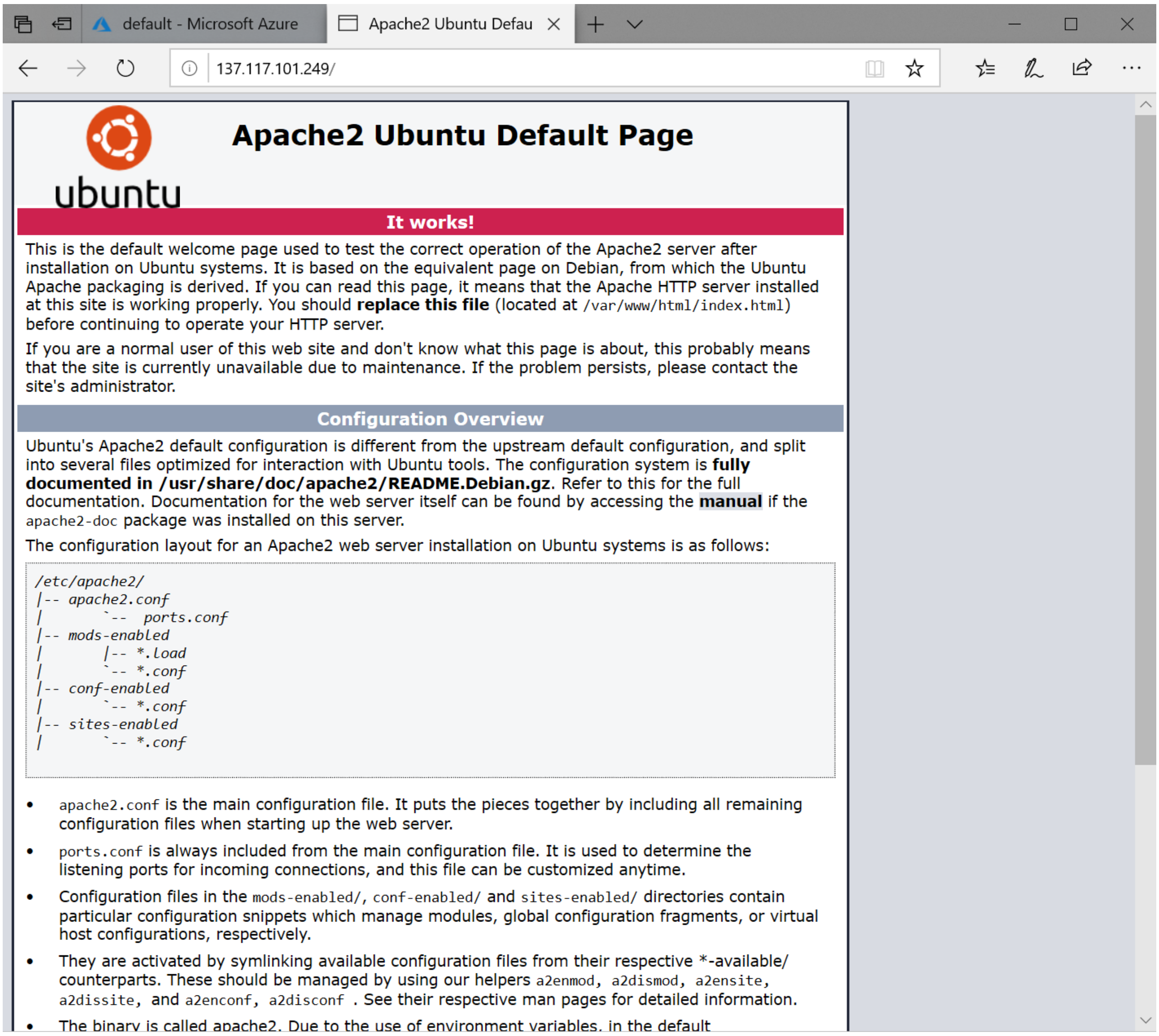
Inbound rules

NAME	PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL
allow-http-traffic	1000	0.0.0.0/0,0.0.0.0/0	0-65535	0.0.0.0/0,0.0.0.0/0	80-80	TCP
allow-ssh-traffic	1010	0.0.0.0/0,0.0.0.0/0	0-65535	0.0.0.0/0,0.0.0.0/0	22-22	TCP
AllowVnetInBound	65000	Virtual network (2 prefixes)	0-65535	Virtual network (2 prefixes)	0-65535	All
AllowAzureLoadBala...	65001	Azure load balancer (1 pre...	0-65535	0.0.0.0/0,0.0.0.0/0	0-65535	All
DenyAllInBound	65500	0.0.0.0/0,0.0.0.0/0	0-65535	0.0.0.0/0,0.0.0.0/0	0-65535	All

Outbound rules

NAME	PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL
AllowVnetOutBound	65000	Virtual network (2 prefixes)	0-65535	Virtual network (2 prefixes)	0-65535	All

4. Of course, the best way to validate it's all working is to hit our server with an HTTP request. It should now work.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default

Warning - Always make sure to lock down ports used for administrative access. An even better approach is to create a VPN to link the virtual network to your private network and only allow RDP or SSH requests from that address range. You can also change the port used by SSH to be something other than the default. Keep in mind that changing ports is not sufficient to stop attacks. It simply makes it a little harder to discover.

Congratulations! With a few steps, you deployed a network security group to a VM that runs Linux.