

Networking Lab1

Create a Virtual Network

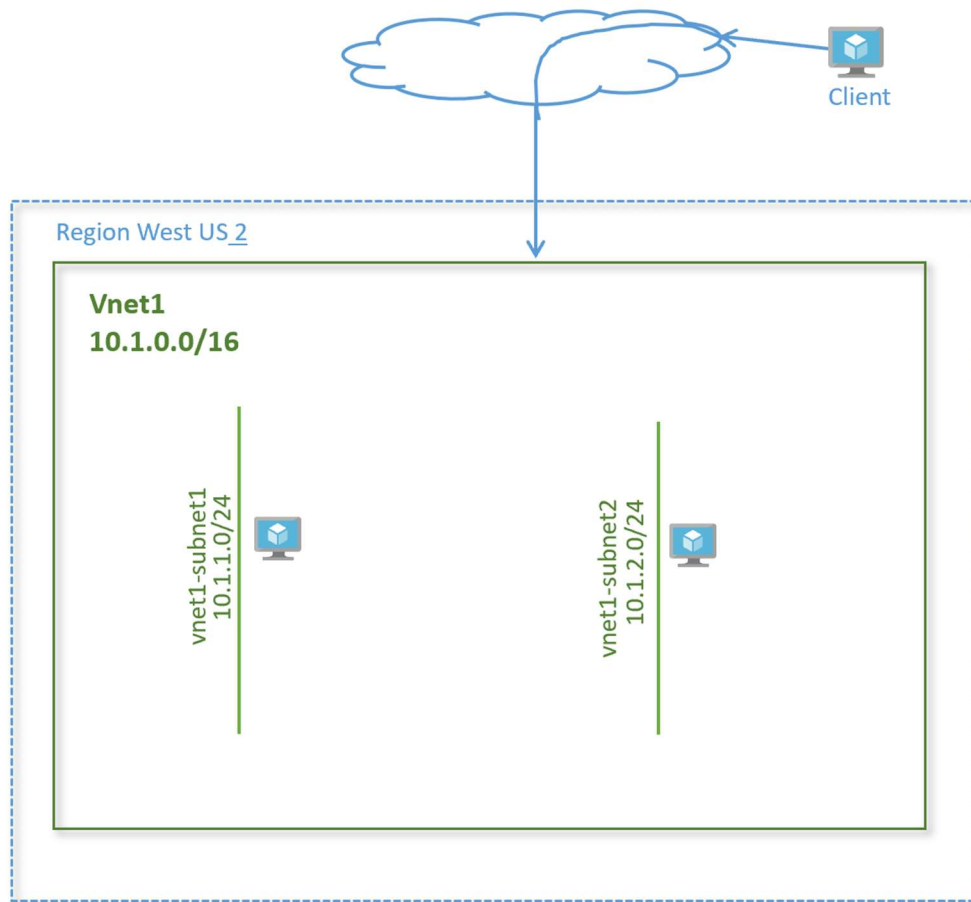
Author:
Binal Shah
Principal Cloud Solution Architect, Microsoft

Lab Overview

In this lab, we will learn how to get started with on Azure to deploy your IaaS resources. We will create a virtual network (vnet) in Azure. We will then add two subnets in the lab and add two virtual machines, one in each subnet.

It is expected you have access to Azure portal and have an account and subscription created on Azure.

Lab Diagram



Create a virtual network

1. To access the Azure portal, go to <http://portal.azure.com>
2. Click on **Create a resource** > **Networking** > **Virtual network**.
3. In **Create virtual network**, enter or select this information:

Setting	Value
Subscription	Select your subscription
Resource group	Select Create new , enter <i>rg-lab</i> , then select OK
Name	Enter <i>vnet1</i>
Region	Select (US) West US 2

3. Click **Next: IP Addresses**>.

IPv4 address space	Enter <i>10.1.0.0/16</i>
--------------------	--------------------------

4. Click **+Add subnet**.

Subnet-name	Enter <i>vnet1-subnet1</i>
Subnet - Address range	Enter <i>10.1.1.0/24</i>

5. Click **Add**.
6. Repeat steps 4 and 5 to add one more subnet as below:

Subnet-name	Enter <i>vnet1-subnet2</i>
Subnet - Address range	Enter <i>10.1.2.0/24</i>

7. Leave the rest as default and select **Review+ Create**. Review the values. Your output should look like this:

Home > New > Create virtual network

Create virtual network

✓ Validation passed

Basics IP Addresses Security Tags Review + create

Basics

Subscription	binal-sandbox
Resource group	(new) rg-lab
Name	vnet1
Region	(US) West US 2

IP addresses

Address space	10.1.0.0/16
Subnet	vnet1-subnet1 (10.1.1.0/24),vnet1-subnet2 (10.1.2.0/24),vnet1-subnet3 (10.1.3.0/24)

Tags

Name	None
------	------

Security

DDoS protection plan	Basic
Firewall	Disabled

[Create](#) [< Previous](#) [Next >](#) [Download a template for automation](#)

8.

9. Click **Create**.
10. Once the deployment is complete, go to the search bar at the top and type 'Virtual Networks'. Select **Virtual Networks** in the search results. You should see vnet1 show up in the list.

<input type="checkbox"/> < vnet1	rg-lab	West US 2	binal-sandbox
----------------------------------	--------	-----------	---------------

Create virtual machines

Create a virtual machine in the virtual network:

Create the first VM

1. On the upper-left side of the screen, select **Create a resource > Compute > Virtual Machine**.
2. In **Create a virtual machine - Basics**, enter or select this information:

Setting	Value
PROJECT DETAILS	
Subscription	Select your subscription.
Resource group	Select rg-lab . You created this in the previous section.
INSTANCE DETAILS	
Virtual machine name	Enter <i>vnet1-vm-mgmt1</i> .
Region	Select West US 2 .
Availability options	Leave the default No infrastructure redundancy required .
Image	Leave the default Ubuntu Server 18.04 LTS .
Size	Leave the default Standard DS2 v3 .
ADMINISTRATOR ACCOUNT	
Username	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements .
Confirm Password	Reenter password.
INBOUND PORT RULES	
Public inbound ports	None

3. Select **Next : Disks**.
4. In **Create a virtual machine - Disks**, leave the defaults and select **Next : Networking**.
5. In **Create a virtual machine - Networking**, select this information:

Setting	Value
Virtual network	Leave the default vnet1 .
Subnet	Leave the default vnet1-subnet1 (10.1.1.0/24) .
Public IP	Leave the default (new) vnet1-vm-mgmt1-ip .

Public inbound ports	Select Allow selected ports .
Select inbound ports	Select HTTP and SSH .

9. Select **Review + create**. You're taken to the **Review + create** page where Azure validates your configuration.
10. When you see the **Validation passed** message, select **Create**.
11. Once the deployment is complete, click **Go to resource**. This will take you to the VM overview page. Verify the VM status shows as **Running**.

Create a second virtual machine

Repeat the above steps to spin up a second virtual machine with the following configuration:

Instance name: **vnet1-vm-web1**

Subnet: **vnet1**

Subnet: **vnet1-subnet2**

Public Inbound ports: **SSH, HTTP**

Keep the rest of the parameters default and create the virtual machine.

Install web server on virtual machine vnet1-vm-web1

Connect to the virtual machine.

1. Search virtual machines in the **Search** bar in the portal.
2. Select the virtual machine **vnet1-vm-web1**.
3. Go to the **Overview** page.
4. Copy the public IP address of the VM.
5. From your laptop terminal, run command:


```
ssh <username>@<Public_IP_of_the_VM>
```
6. Install apache2 on the server.



```
sudo apt-get -y update
sudo apt-get -y install apache2
```
7. Verify the service is running on the server. You should see a status of active (running) in the output:


```
sudo service apache2 status
```
8. When done, type `exit` to leave the SSH session.

View the web server in action

Use a web browser of your choice to view the default welcome page. Type the public IP address of the VM as the web address. The public IP address can be found on the VM overview page or as part of the SSH connection string you used earlier. Verify the web page loads successfully.

← → ↻ ⓘ Not secure | 52.246.249.251



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 configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.