# Deployment and Configuration of LAMP (Linux, Apache, MySQL, PHP) in Azure Introduction

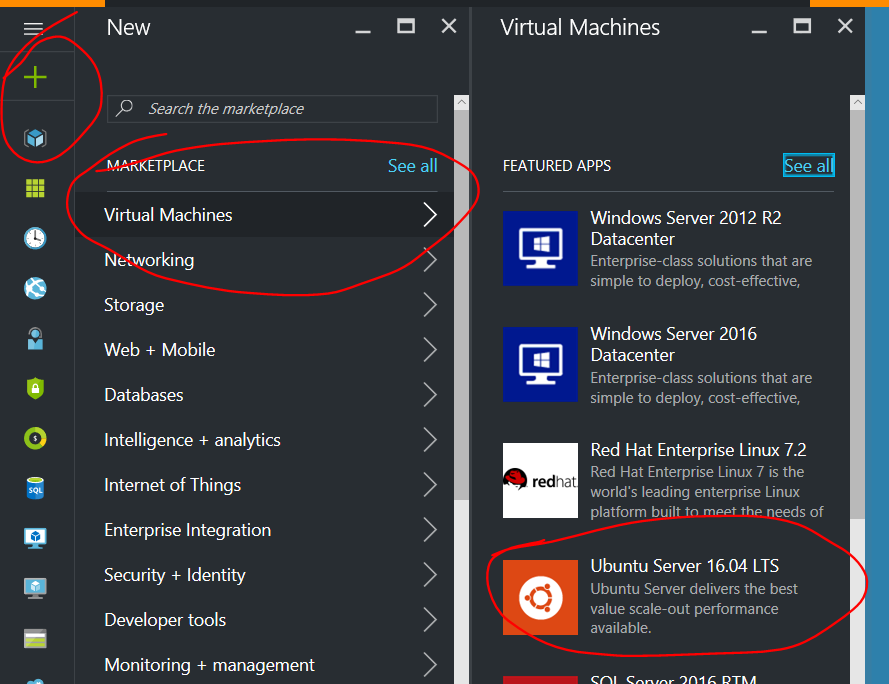
The following document is designed to help quickly and efficiently deploy and configure a LAMP stack in Microsoft Azure. The LAMP stack is a web service solution, comprised of four main open source components. For the purpose of this document the LAMP stack consists of the four original components: A Linux based operating system, Apache web server, MySQL relational database, and the PHP programming language. Although a LAMP stack could be comprised of other components, they will not be covered in this document. This document will cover deploying a Linux machine on Microsoft Azure, Installing the LAMP stack on either Ubuntu and CentOS, Configuring the LAMP stack on Ubuntu and CentOS, and finally, provide a bash script to automate the installation and configuration of the LAMP stack on both Ubuntu and CentOS.

## 2.1 Deploying Linux VM Through Azure Portal

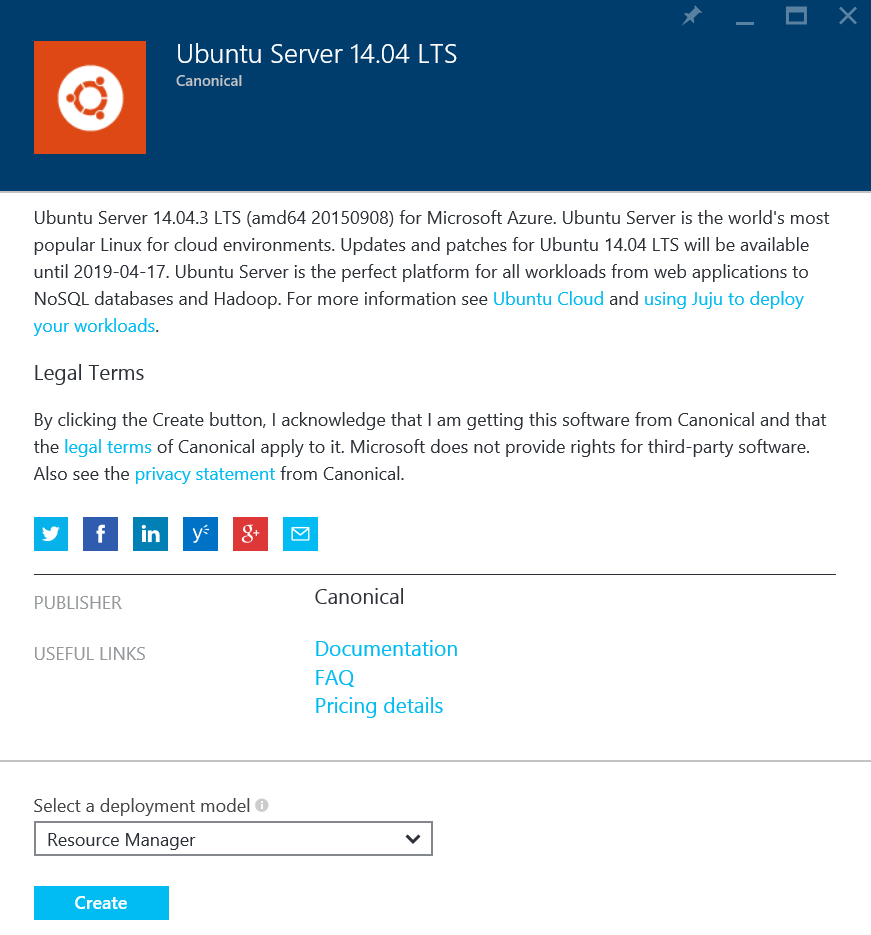
Use the following general procedures to deploy an Ubuntu Virtual Machine in Microsoft Azure

**Complete these steps by connecting to your Azure subscription**

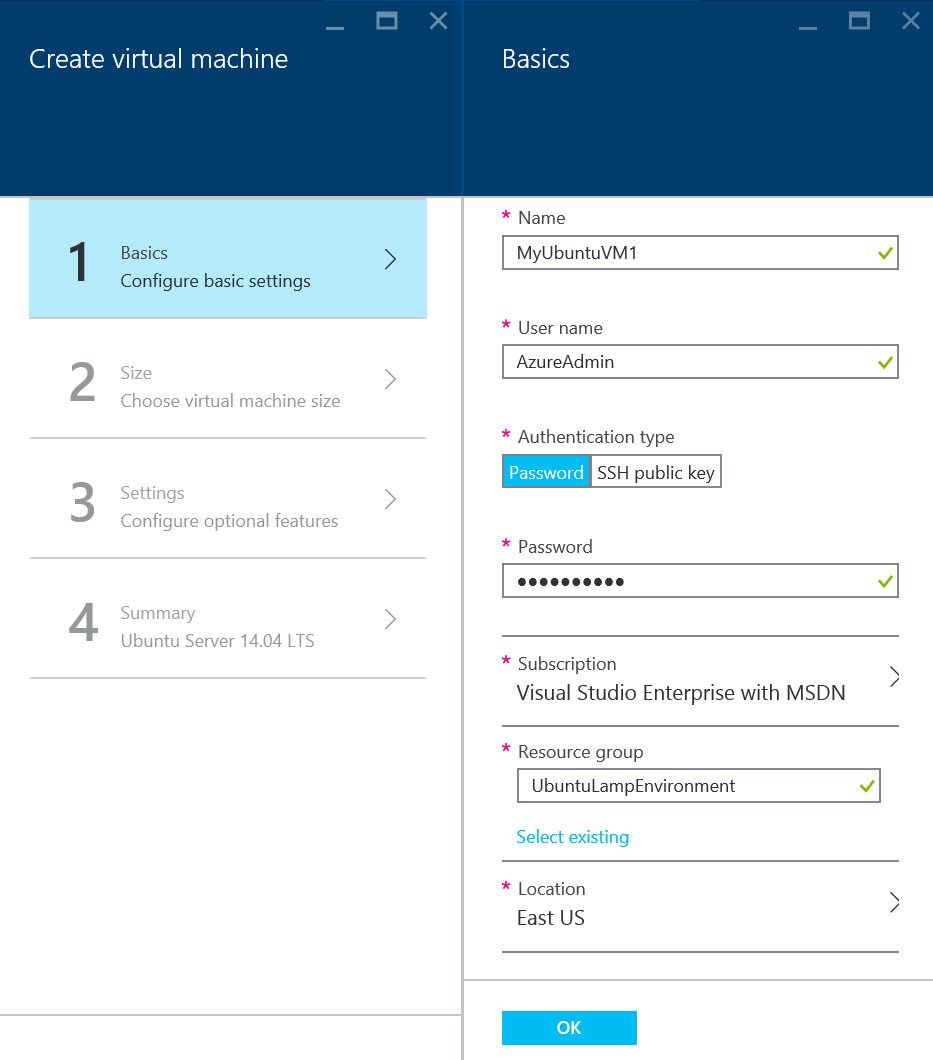
1. From your system, open Internet Explorer in In Private browsing mode
2. Navigate to <http://portal.azure.com>
3. When prompted, enter your Microsoft Account credentials



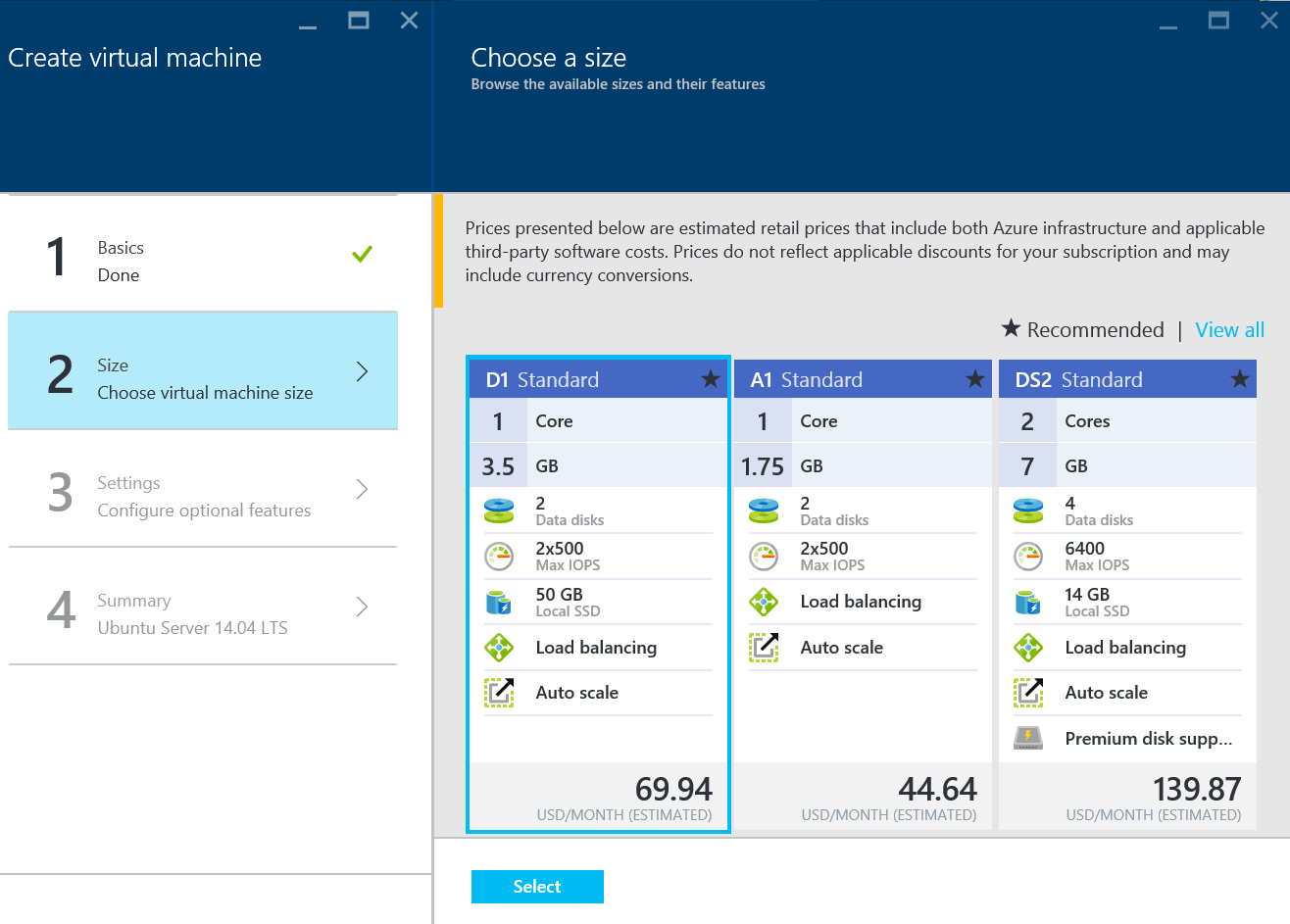
1. On the Hub menu, click New (+) > Virtual Machines > Ubuntu Server 16.04 LTS
2. On the **Ubuntu Server 14.04 LTS** page, under **Select a deployment model**, select **Resource Manager**. Click Create



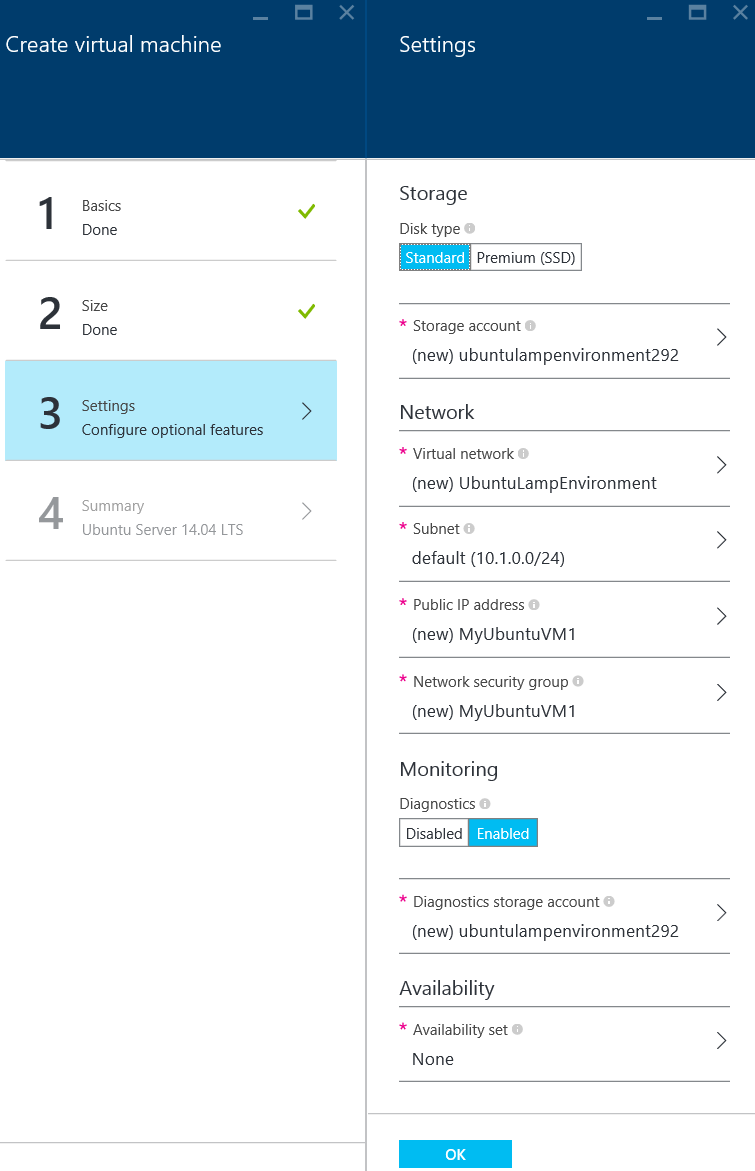
1. On the **Create virtual machine** blade, click **Basics**. Enter a **Name** you want for the virtual machine, the administrative **User name**, and a **SSH public key** file (in ssh-rsa format). While a SSH public key is the preferred method of authentication, you can use a strong **Password** if you would like. If you have more than one subscription, specify the one for the new virtual machine, as well as a new or existing **Resource group** and an Azure datacenter **Location.**



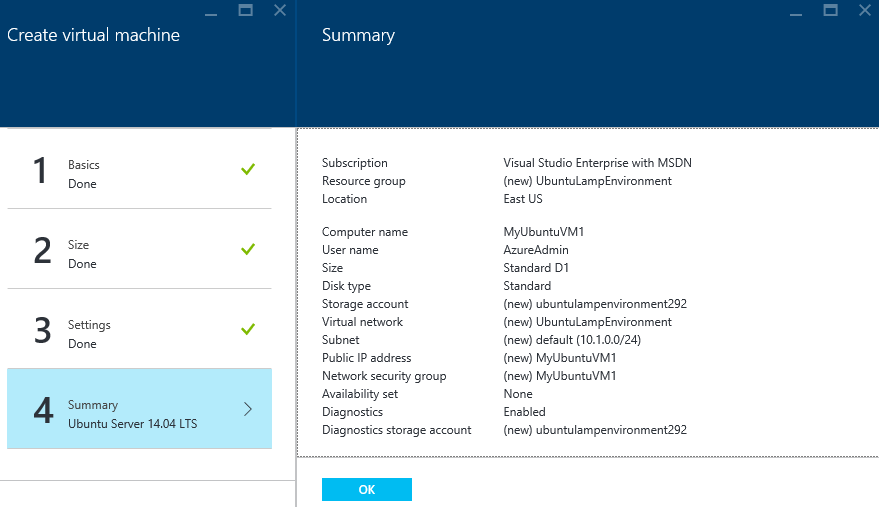
1. Click **Size** and select an appropriate virtual machine size for your needs. Each size specifies the number of compute cores, memory, and other features, such as support for Premium Storage, which will affect the price. Azure recommends certain sizes automatically depending on the image you choose. To view all size options available, select the View All option.



1. Click **Settings** to see storage and networking settings for the new virtual machine. For a first virtual machine you can generally accept the default settings. If you selected a virtual machine size that supports it, you can try out Premium Storage by selecting Premium (SSD) under Disk type. Azure provides defaults to many values based on whether an existing resource group was chosen or a new one was created, and the default values can normally be accepted. However, on this tab you can also choose to manually set options for naming the **Storage Account**, and choosing the storage account type, as well assetting the **Virtual Network** preferences including **Subnet** IP addresses, a **Public IP Address**, and the **Network Security Group**. There are similarly options for **Monitoring** and **Availability Sets**.



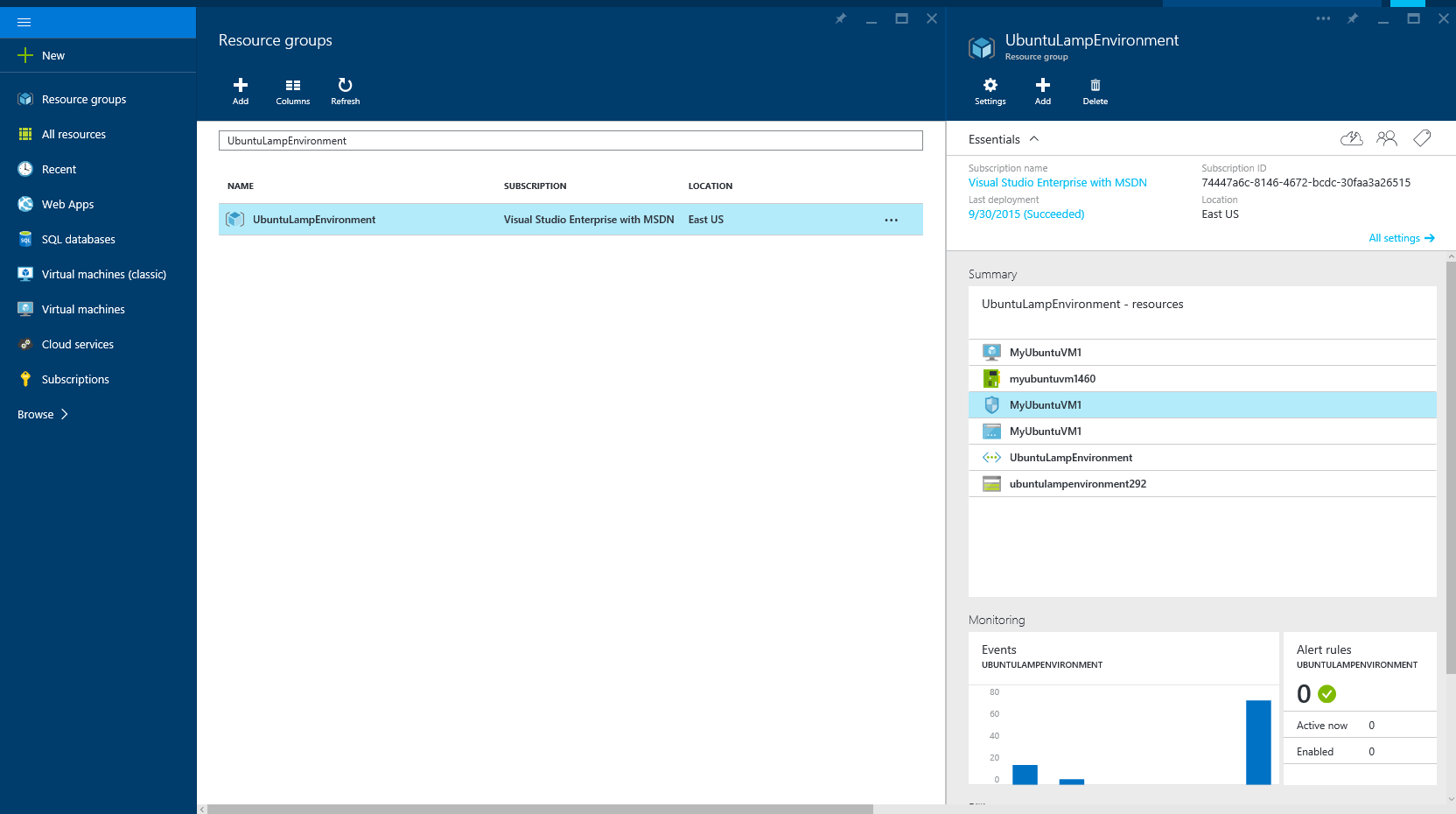
1. Click **Summary** to review your configuration choices. When you're done reviewing or updating the settings, click **OK**



### 2.1.1 Create HTTP Endpoint for Access to LAMP Stack

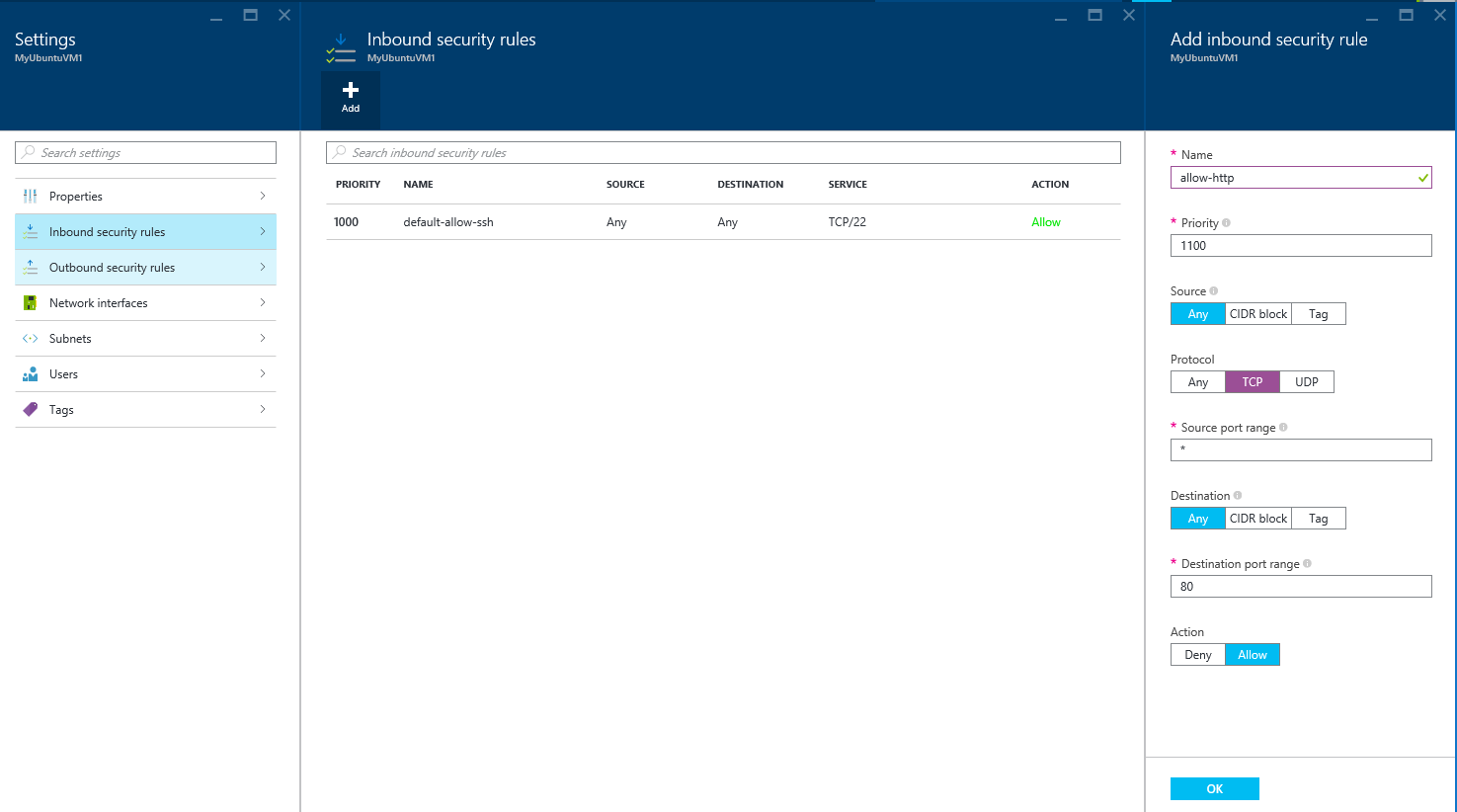
In order to view your LAMP stack on your local machine through a web browser you must open an HTTP endpoint in the azure portal for your newly created VM. This allows you to reach the VM using the HTTP protocol on a public IP address. In Azure, the HTTP endpoint is not created by default on Linux machines and must be manually enabled. This endpoint can be used while deploying the Apache, PHP, and MySQL to test and verify that the software has been installed correctly.

1. Navigate to the portal, and choose Resource **Groups** on the main menu, Choose the name of the **Resource Group** you just created with your VM.



Choose the **VM**, then select the **Network Interface**, then choose the **Network Security Group** that your VM belongs to.

1. Select **Inbound Security Rules** and Choose **Add** to add a new rule.
2. Choose a name for your security rule, and select protocol as TCP. We will set the **Source Port Range** to \*. By Default, Apache will listen on port 80 (The default HTTP port), so we will specify this as the **Destination Port Range.** Make sure allow is selected, so that the server will accept the HTTP request.



1. After clicking ok, the network security group will update and the new rule will be listed under the **Inbound Security Rules**.
2. For more information on Network Security Groups go to:

<https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/>

## 2.2 Deploying Linux VM through PowerShell

The following content describes the necessary steps to deploy and configure an Ubuntu Virtual Machine on Microsoft Azure under the **Resource Manager** framework. The following steps require Windows PowerShell version 3.0 or higher, and Azure PowerShell version 0.9.0 or higher.   
To verify you have the correct version of Windows PowerShell, run the following command:

$PSVersionTable

To verify you have the correct version of Azure PowerShell, run the following command:

Get-Module azure | format-table version

If you do not have the correct versions, they can be downloaded here: <https://azure.microsoft.com/en-us/downloads/>

Getting PowerShell Ready

1. Open an Azure PowerShell command prompt and log on to Azure with this command.

Add-AzureRMAccount

### 2.2.1 Deploying Default Linux VM In Resource Manager Using JSON Template

1. In an opened PowerShell command prompt set to Azure Resource Manager, run the following script, replacing *resource group name* with the name of the new resource group and *Azure location* with the Azure datacenter location where you want the resource to be located, in this case, “West US”

New-AzureRMResourceGroup -Name "resource group name" -Location "Azure location"

1. substituting *Deployment name* with the name that you want to use for the deployment and *resource group name* with the name of the existing resource group.

New-AzureRMResourceGroupDeployment -Name "deployment name" -ResourceGroupName "resource group name" -TemplateUri https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vm-simple-linux/azuredeploy.json

1. You're prompted to supply the values of parameters in the parameters section of the JSON file, including, storage account name, username, password, and DNS Public IP name. This JSON file defaults to use Ubuntu Server as the Linux operating system. This can be substituted for another Linux distribution such as CentOS by changing the **Image Publisher** and **Image Offer**  variables within the JSON template.
2. Once the script is finished running, you have an Ubuntu VM Started and running in a resource group.

# 3 Installing LAMP Stack on Ubuntu

There are many different ways to install a LAMP stack on an Ubuntu machine running in Microsoft Azure. This section examines different approaches, from using a default Ubuntu apt-get install, as well as individually installing and configuring each component.

## 3.1 Default Install Script

On Ubuntu versions 10.04 and higher, a default version of the LAMP stack can be installed with a single command. This will deploy Apache2, MySQL, and PHP5 in their default configurations. You must then configure them after they are installed. \*Note: The script will stop and prompt the user for a root password for the MySQL instance. This cannot be passed as an argument using the single script. To use the single script install, use the following commands

1. Update the package index

$ sudo apt-get update

1. Install the LAMP stack (\*Note: The caret (^) at the end IS part of the command)

$ sudo apt-get install tasksel

$ sudo tasksel

The tasksel command should show a GUI, go down in the Menu options and select “LAMP Server” , press “SPACE” to select and “Tab” to move to “OK” button.

1. Once the script is finished running, the machine will have the LAMP stack installed and running on the machine. \*Note: All of the components (Apache, PHP, & MySQL) will be configured in their respective default settings & ports

## Installing Components Individually (Optionally)

The components of the LAMP stack can also be installed and configured individually. Many packages can be installed using any method, however this document provides the installs using the apt-get method. After installing the packages individually, you can test that each module is working by navigating to your Azure VM in your web browser. In order to do this, you must create an HTTP end point for your VM, to do this, refer to the setting up endpoints section

### Installing Apache2

1. To install Apache2 use: \*Note: The –y silences the install confirmation

$ sudo apt-get -y install apache2

1. The apache server must be restarted after install to run. Either of the following commands can be used:

$ sudo /etc/init.d/apache2 restart

$ sudo service apache2 restart

### Installing PHP 5

1. Install PHP5 using any method \*Note: The –y silences the install confirmation

$ sudo apt-get -y install php5 libapache2-mod-php5

### Installing MySQL

1. Install My SQL with PHP5 using any method. The command will stop during runtime to prompt for a root password. \*Note: The –y silences the install confirmation

$ sudo apt-get -y install mysql-server libapache2-mod-auth-mysql php5-mysql

1. Once the script is finished running, MySQL will be installed. It is reccomended to restart the MySQL server and the Apache server. This can be done with the following commands

$ /etc/init.d/mysql restart

$ /etc/init.d/apache2 restart

## Configuring Installed LAMP Stack

Once the components of the LAMP stack have been installed, it may be necessary to configure MySQL and Apache.

### Configuring Apache2

1. When deploying Apache2 in Microsoft Azure, Apache can often not initially determine the servers fully qualified domain name, giving the error “apache2: Could not determine the server's fully qualified domain name, using 127.0.0.1 for ServerName”. To fix this run the following commands:
2. For Ubuntu Version 13.10 and newer:

$ echo "ServerName localhost" | sudo tee /etc/apache2/conf-available/fqdn.conf && sudo a2enconf fqdn

1. For Ubuntu Versions 13.04 and older:

$ echo "ServerName localhost" | sudo tee /etc/apache2/conf.d/fqdn

1. Restart Apache2 server for the changes to take effect

$ sudo service apache2 restart

1. Make sure that Apache installed correctly. Use a web browser to go to the public IP address of your Ubuntu VM in Azure, and you should see a page with the headline “It Works!”

Setting the Apache HTTP Listening Port

1. Apache will default to listening on port 80. This can be changed by editing the ports.conf file located in /etc/apache2/ports.conf and replacing the line that reads “Listen 80” with “Listen *NewPort*” where “*NewPort”* is the port of your choosing

$ sudo nano /etc/apache2/ports.conf

1. You will also need to edit the file 000-default.conf file located in /etc/apache2/sites-available/000-default.conf. Replace “VirtualHost \*:80” with “VirtualHost \*:*NewPort”* where “*NewPort”* is the port of your choosing

$ sudo nano /etc/apache2/sites-available/000-default.conf

Rebooting Apache

1. After making any configuration changes, it will be necessary to restart the Apache service for the changes to take effect. To restart Apache, run the following command

/etc/init.d/apache2 restart

### 3.3.3 Configuring PHP 5

1. Enable the PHP 5 module \*Note this creates a symbolic link /etc/apache2/mods-enabled/php5 pointing to /etc/apache2/mods-availble/php5

$ sudo a2emod php5

1. Apache2 will need to be restarted to accept the new PHP 5 module

$ sudo service apache2 restart

1. To check the PHP install, create a file in the /var/www/html directory called phpinfo.php and add the code   
   "<?php phpinfo(); ?>" to the file.

$ echo \<\?php phpinfo\(\)\; \?\> > /var/www/html/phpinfo.php

1. In your web browser, go to your virtual machines public IP address/phpinfo.php (10.1.1.10/phpinfo.php). You should see a page listing the PHP configuration for your machine. This verifies that PHP is working with Apache

### 3.3.2 Configuring MySQL

Set the MySQL Bind address.

1. Edit the my.cnf file located in /etc/mysql/my.cnf and replace the bind-address to 0.0.0.0

$ sudo nano /etc/mysql/my.cnf

bind-address = 0.0.0.0

1. This can be done with a single command as well using the sed command. \*Note: The .’s in the IP address must be escaped using the “\”

$ sed -i 's/127\.0\.0\.1/0\.0\.0\.0/g' /etc/mysql/my.cnf

Change the MySQL Port

1. MySQL will default to port 3306. This can be changed by editing the my.cnf file located in /etc/mysql/my.cnf and replacing 2 lines that specify the port as 3306 to the port of your choosing

$ sudo nano /etc/mysql/my.cnf

Port = 9999

1. This can be done with a single command as well using the sed command. In This case, the new port would be 9999

$ sed -i "s/3306/9999/g" /etc/mysql/my.cnf

Grant Full Privileges to the Root Account

1. It may also be necessary to grant full privileges to the root account in MySQL. To do this, use the following command substituting your specified root MySQL password in place of “*MYSQLPASSSWORD”*

$ mysql -h 127.0.0.1 -uroot –p*MYSQLPASSSWORD* -P *MYSQLPASSSWORD* -e 'USE mysql; UPDATE `user` SET `Host`="%" WHERE `User`="root" AND `Host`="localhost"; DELETE FROM `user` WHERE `Host` != "%" AND `User`="root"; FLUSH PRIVILEGES;'

Restarting MySQL

1. After making any configuration changes, it will be necessary to reboot the MySQL service for the changes to take effect. To reboot MySQL, run the following command

$ /etc/init.d/mysql restart

# 4 Installing LAMP Stack on CentOS

## 4.1 Installing with Specified Configurations

Unlike Ubuntu there is no packages that will install the lamp stack, as a result we will have the install all the packages individually. (Please note: A nested command can also be used)

### 4.1.1 Installing Apache2

1. To install Apache2 use: \*Note: The –y silences the install confirmation

$ sudo yum -y install httpd httpd-devel

1. The apache server must be restarted after install to run. Either of the following commands can be used:

$ sudo /etc/init.d/httpd restart

$ sudo service httpd restart

### Installing PHP 5

1. Install PHP5 using any method \*Note: The –y silences the install confirmation

$ sudo yum -y install php php-mysql php-common php-gd php-mbstring php-mcrypt php-devel php-xml

### Installing MySQL

1. Install My SQL with PHP5 using any method. The command will stop during runtime to prompt for a root password. \*Note: The –y silences the install confirmation

$ sudo yum –y install mysql mysql-server mysql-devel

1. Once the script is finished running, MySQL will be installed. It is recommended to restart the MySQL server and the Apache server. This can be done with the following commands

$ /etc/init.d/mysqld restart

$ /etc/init.d/httpd restart

## Configuring Installed LAMP Stack

### 4.2.1 Configuring Apache

1. When deploying Apache2 in Microsoft Azure, Apache can often not initially determine the servers fully qualified domain name, giving the error “apache2: Could not determine the server's fully qualified domain name, using 127.0.0.1 for ServerName”. To fix this run the following commands:

$ sed -i "s/#ServerName\ www.example.com:80/ServerName\ localhost/g" /etc/httpd/conf/httpd.conf

1. Restart Apache2 server for the changes to take effect

$ sudo service httpd restart

1. Make sure that Apache installed correctly. Use a web browser to go to the public IP address of your CentOS VM in Azure, and you should see a page with the headline “It Works!”

Setting the Apache HTTP Listening Port

1. Apache will default to listening on port 80. This can be changed by editing the httpd.conf file located in /etc/httpd/conf/httpd.conf and replacing the line that reads “Listen 80” with “Listen *NewPort*” where “*NewPort”* is the port of your choosing

$ sudo nano /etc/httpd/conf/httpd.conf

1. You will also need to find the line “Listen 80” and you simply change the port from 80 to whatever port you desire or simply execute the command below:

$ sed -i "s/Listen\ 80/Listen 81/g" /etc/httpd/conf/httpd.conf

Change 81 to whatever port you intend to use.

### 4.2.2 Configuring MySQL

Set the MySQL Bind address.

1. Edit the my.cnf file located in /etc/my.cnf and replace the bind-address to 0.0.0.0

$ sudo nano /etc/my.cnf

bind-address = 0.0.0.0

1. This can be done with a single command as well using the sed command. \*Note: The .’s in the IP address must be escaped using the “\”

$ sed -i 's/127\.0\.0\.1/0\.0\.0\.0/g' /etc/my.cnf

Change the MySQL Port

1. MySQL will default to port 3306. This can be changed by editing the my.cnf file located in /etc/mysql/my.cnf and replacing 2 lines that specify the port as 3306 to the port of your choosing

$ sudo nano /etc/my.cnf

Port = 9999

1. This can be done with a single command as well using the sed command. In This case, the new port would be 9999

$ sed -i "s/3306/9999/g" /etc/my.cnf

Grant Full Privileges to the Root Account

1. It may also be necessary to grant full privileges to the root account in MySQL. To do this, use the following command substituting your specified root MySQL password in place of “*MYSQLPASSSWORD”*

$ mysql -h 127.0.0.1 -uroot –p*MYSQLPASSSWORD* -P *MYSQLPASSSWORD* -e 'USE mysql; UPDATE `user` SET `Host`="%" WHERE `User`="root" AND `Host`="localhost"; DELETE FROM `user` WHERE `Host` != "%" AND `User`="root"; FLUSH PRIVILEGES;'

Restarting MySQL

1. After making any configuration changes, it will be necessary to reboot the MySQL service for the changes to take effect. To reboot MySQL, run the following command

$ /etc/init.d/mysqld restart

# 5 Automation Script for Full Configured Install

The LAMP stack can be installed and configured with the following bash script. The script takes the inputs of a MySQL root password, MySQL port, and an Apache Port as command line parameters. MySQL password can be specified by using -mysqlap or --mysql\_admin\_password; the MySQL port can be specified by using -mysqlp or --mysql\_port; and the Apache port can be specified by using -httpp or --http\_port. If no Apache & MySQL ports are specified, both will be deployed with their default ports of 80 and 3306 respectively. If no MySQL password is specified, the script will pause during execution and prompt the user for a root password.

#!/bin/bash

mysqlap=""

mysqlp=""

httpp=""

count=0

# Simply dump your arguments, used for debugging purposes

echo "1: $1"

echo "2: $2"

echo "3: $3"

echo "4: $4"

echo "5: $5"

echo "6: $6"

if [ $# -eq 0 ]; then

echo "No argument supplied"

else

for args; do

if [ -n "$1" ];then

case "$1" in

-mysqlap | --mysql\_admin\_password )

if [ -z "$2" ]; then

echo "Argument -mysqlap or --mysql\_admin\_password cannot be null"

break

else

mysqlap="$2"

shift 2

fi

;;

-mysqlp | --mysql\_port )

if [ -z "$2" ]; then

echo "Argument -mysqlp or --mysql\_port cannot be null"

break

else

mysqlp="$2"

shift 2

fi

;;

-httpp | --http\_port )

if [ -z "$2" ]; then

echo "Argument -httpp or --http\_port cannot be null"

break

else

httpp="$2"

shift 2

fi

;;

\*) # No more options

break

;;

esac

fi

done

fi

echo "MySQLPassword: $mysqlap"

echo "MySQLPort: $mysqlp"

echo "HTTPPort: $httpp"

# Update package dependancies

sudo apt-get -y update

#Install Apache 2

sudo apt-get -y install apache2

#Check Version & Update DNS

OS=$(lsb\_release -sr)

Version=13.10

if (( $(echo $OS '>=' $Version | bc -l) )); then

echo "ServerName localhost" | sudo tee /etc/apache2/conf-available/fqdn.conf && sudo a2enconf fqdn

else

echo "ServerName localhost" | sudo tee /etc/apache2/conf.d/fqdn

fi

#Install PHP5

sudo apt-get -y install php5 libapache2-mod-php5

#Activate PHP5 Apache Mod

sudo a2emod php5

#Restart Apache To Make Changes Take Effect

sudo service apache2 restart

#Set MySQL Admin Password

if [ -n "$mysqlap" ]; then

sudo apt-get -y install debconf-utils

sudo debconf-set-selections <<< "mysql-server mysql-server/root\_password password $mysqlap"

sudo debconf-set-selections <<< "mysql-server mysql-server/root\_password\_again password $mysqlap"

sudo debconf-set-selections <<< 'mysql-server-5.1 mysql-server/start\_on\_boot boolean true'

fi

#Install MySQL wiht PHP5

sudo apt-get -y install mysql-server libapache2-mod-auth-mysql php5-mysql

# Modify MySQL bind-address

sed -i 's/127\.0\.0\.1/0\.0\.0\.0/g' /etc/mysql/my.cnf

# Modify MySQL Port

if [ -n "$mysqlp" ]; then

sed -i "s/3306/$mysqlp/g" /etc/mysql/my.cnf

fi

#Modify Apache Listening Address

if [ -n "$httpp" ]; then

port\_to\_use="Listen $httpp"

sed -i "s/Listen 80/$port\_to\_use/g" /etc/apache2/ports.conf

fi

#Modify Apache Listening Address

if [ -n "$httpp" ]; then

port\_to\_use="VirtualHost \*:$httpp"

sed -i "s/VirtualHost \\*:80/$port\_to\_use/g" /etc/apache2/sites-available/000-default.conf

fi

# write some PHP

echo \<center\>\<h1\>My Demo App\</h1\>\<br/\>\</center\> > /var/www/html/phpinfo.php

echo \<\?php phpinfo\(\)\; \?\> >> /var/www/html/phpinfo.php

# Restart MySQL and Apache

/etc/init.d/mysql restart

/etc/init.d/apache2 restart

# Grant Full Privileges to the root account

mysql -h 127.0.0.1 -uroot -p$mysqlap -P $mysqlp -e 'USE mysql; UPDATE `user` SET `Host`="%" WHERE `User`="root" AND `Host`="localhost"; DELETE FROM `user` WHERE `Host` != "%" AND `User`="root"; FLUSH PRIVILEGES;'

# 6 Resources

* Deploying VMs Through Azure Portal
  + <https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-tutorial-portal-rm/>
* Deploying LAMP Stack on Ubuntu
  + <https://help.ubuntu.com/community/ApacheMySQLPHP>
* Deploying LAMP Stack on CentOS
  + <https://www.howtoforge.com/quick-n-easy-lamp-server-centos-rhel>