Microsoft Azure for Linux and Mac Users

This article is an introduction to Microsoft Azure, Microsoft's application for the public cloud. The article is aimed at the research and academic users who are familiar with Linux or the Mac OS. The paper shows how you can take advantage of some of the capabilities Microsoft Azure offers, such as using virtual machines (VMs), persistent storage and deploying applications from a source code repository.

Note: It's also possible to run Microsoft Windows on your Mac by using VM hosting software such as VMWare Fusion 6 and Parallels Desktop 9 for Mac. You can then take advantage of integrated development environments such as Microsoft Visual Studio.

What is Microsoft Azure

Microsoft Azure is an open and flexible cloud-computing platform that you can use in many ways. For example, you can create VMs, create and deploy web sites and applications, store data, and run big data and high performance computing (HPC) workloads. For a detailed introduction to Microsoft Azure, read Intro to Microsoft Azure.

Microsoft Azure supports a diverse set of development tools. You can create applications in any language or tool, with a variety of languages supported by cross-platform SDKs. These languages are Java, PHP, Node.js, Ruby, Python and those in the .NET Framework family (C#, F# and VB.NET). In this article we introduce you to some Microsoft Azure fundamentals and show you how you can use Linux or Mac OS to:

- Set up a Microsoft Azure account.
- Download the correct SDK for your preferred development language.
- Use the Microsoft Azure Management Portal
- Use the command line interface (CLI) to perform basic VM operations.
- Upload data to Microsoft Azure Blob Storage
- Deploy to Microsoft Azure from a source repository.

Setting up a Microsoft Azure Account

If you don't already have a Microsoft Azure account, you'll need to create one. You can sign up for a free trial.

- 1. Go to the Microsoft Azure home page.
- 2. Scroll down the page and click **Start your free trial**.
- 3. On the Microsoft Azure pricing page, click **Try it now**.
- 4. Enter your Microsoft account credentials. If you don't have a Microsoft account, you can create one from this page. Click **Sign In**.
- 5. Fill out the Microsoft Azure Sign up form. Click **Sign up**.

Downloading an SDK

Now that you have an account, choose the language you want to use and download the corresponding SDK. Go to the <u>Microsoft Azure downloads</u> page, and select Node.js, PHP, Java, Ruby or Python.

Managing Resources

Microsoft Azure provides three options for managing resources such as web sites, virtual machines and databases.

- Web-based management portal
- Command line interface (CLI)
- Programmatic interface (API).

We cover the first two options in this article. Direct use of the API is not described here but to find out more, go to the <u>API and Schema References for Microsoft Azure</u> page. Both the portal and the CLI use the API behind the scenes.

Using the Management Portal

Once you have an account, you can use the Management Portal to perform many common tasks. Later in this article we show you how use the CLI and the Management Portal together, but if you are only using a few instances, the Management Portal may be your most convenient option. Here is an example of what the Management Portal looks like.

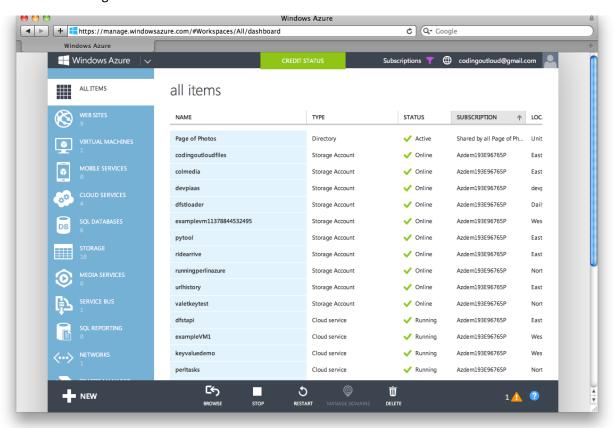


Figure 1

The Management Portal allows you to manage all the resources in subscriptions associated with your Microsoft Azure account. (A subscription is a group of services that shares a common billing account.) For example, you can view your current resources, add new resources, delete resources, start and stop services such as VMs, view dashboards, and view or modify configuration settings. You can also use the Management Portal to review bills and manage spending limits.

Installing and Using the CLI to Manage VMs

This section shows you how to install the CLI tools for the Mac OS and for Linux and how to use them to manage VMs.

Installing the CLI on a Mac

If you are using a Mac, download the <u>Microsoft Azure SDK Installer</u>. Open the .pkg file and follow the installation instructions.

Installing on Linux

If you are using Linux, install the latest version of Node.js. For more information, see <u>Installing Node.js</u> via package manager, on GitHub.

After you have installed Node.js, run the following shell command:

```
npm install azure-cli -g
```

If you need to use elevated privileges, use the sudo command:

```
sudo npm install azure-cli -g
```

Testing the Installation

To ensure that the tools installed correctly, type azure at the command prompt. You should see a list of all the available commands.

Using the CLI Tools

This section shows you how to use the CLI to perform common VM management tasks. Each CLI command has a help feature at each level to show supported commands. Simply type the command and append --help (or –h for short). Here is an example.

```
Terminal — bash — 84×18
                                                                                     mac osx> azure vm image -h
help:
        Commands to manage your Virtual Machine images
help:
        Show details about a VM image
help:
help:
          vm image show [options] <name>
help:
        List VM images
help:
          vm image list [options]
help:
help:
help:
        Delete a VM image from a personal repository
help:
          vm image delete [options] <name>
help:
help:
        Upload and register a VM image
          vm image create [options] <name> [source-path]
help:
help:
help:
        Options:
help:
          -h, --help output usage information
mac osx>
```

Listing the VM Images Available in the Image Gallery

The Microsoft Azure Image Gallery stores a variety of virtual machine images. One way to create a VM from an image in the gallery is to use the Management Portal. To learn how to do this, read <u>Create a Virtual Machine Running Linux</u>.

You can also do it programmatically. Here is how to list the available Windows and Linux images that are included in the gallery.

```
azure vm image list
```

Each image has a name, a category, and an OS. The following image, which is an Ubuntu distribution from Canonical, will be used in the subsequent examples in this article.

- Name: b39f27a8b8c64d52b05eac6a62ebad85__Ubuntu-13_04-amd64-server-20130824-en-us-30GB
- Category: Canonical
- OS: Linux

Creating a VM Using the Image Gallery

You will remotely manage your Linux VM by using OpenSSL. The recommended approach is to first create a digital certificate and then provide its public key when you create the VM. For a useful overview, see <u>Creating secure Linux VMs in Azure with SSH key pairs</u>.

OpenSSL is already installed by default on Mac OS and Linux desktop operating systems. The following shell command creates a certificate named example1.key.

```
openssl req -x509 -nodes -days 365 \
-newkey rsa:2048 \
-keyout example1.key \
```

```
-out example1.pem
```

The following screenshot shows an example of creating a certificate.

```
Terminal - bash - 84×26
                                                                                 \equiv
mac osx> openssl req -x509 -nodes -days 365 \
> -newkey rsa:2048 \
> -keyout example1.key \
> -out example1.pem
Generating a 2048 bit RSA private key
......+++
writing new private key to 'example1.key'
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
mac osx> ls example1.*
example1.key
              example1.pem
mac osx>
```

Figure 3

Note that OpenSSL prompts for some additional data, but you do not need to provide it (just press Enter). The ssh connection will work just as well without it.

The example 1. key file is a security credential, so you must change its permissions. One way to secure the file is to use the chmod command.

```
chmod 600 example1.key
```

If you do not secure the key file, OpenSSL will refuse to accept it.

Now that you have a digital certificate, you can create your VM, named example VM1, which is based on the Ubuntu image mentioned above. Here is the command.

```
azure vm create --location "West US" --ssh 22 --ssh-cert ./example1.pem --no-ssh-password exampleVM1 b39f27a8b8c64d52b05eac6a62ebad85__Ubuntu-13_04-amd64-server-20130824-en-us-30GB exampleUserName
```

The -ssh 22 parameter opens port 22 for SSH access. Here is the output when we run the command.

```
exampleUserName@exampleVM1: ~ — bash — 84×12

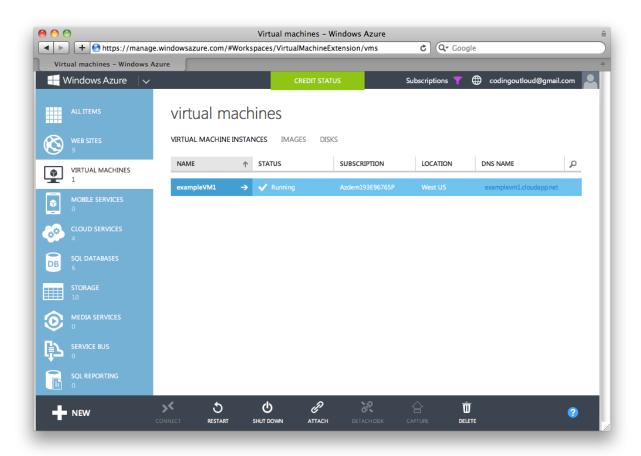
mac osx> azure vm create --location "West US" --ssh 22 --ssh-cert ./example1.pem --n o-ssh-password exampleVM1 b39f27a8b8c64d52b05eac6a62ebad85__Ubuntu-13_04-amd64-serve r-20130824-en-us-30GB exampleUserName info: Executing command vm create + Looking up image + Looking up cloud service + Creating cloud service + Retrieving storage accounts + Configuring certificate + Creating VM info: vm create command OK mac osx>
```

The command takes a short time to complete. Once the VM is created, type the following command. azure vm list

You will see all the VMs in the currently selected subscription, which now includes the VM you just created. Here is the output.

Figure 5

If you use the management portal, and select the Virtual Machines tab, you will see the same thing.



You can also use the Management Portal to select **exampleVM1** and see its dashboard. The dashboard gives you a summary of important information about the VM. Here is an example. (The two screenshots show the complete dashboard as you scroll down.)

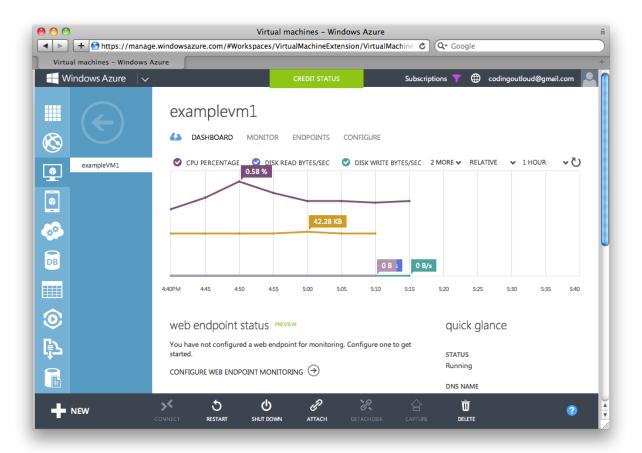
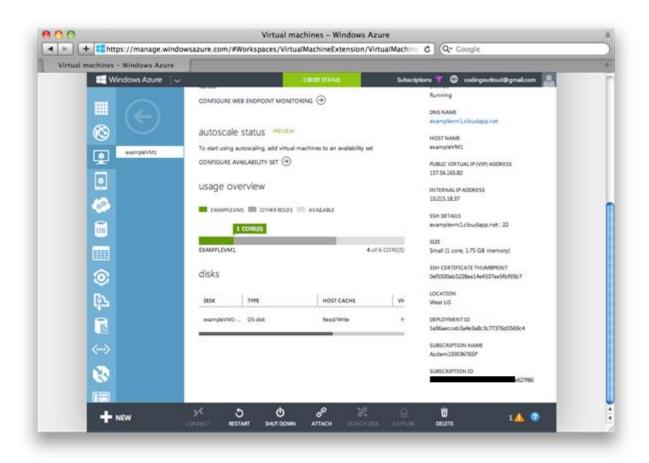


Figure 7

In the following screenshot on the right, under **SSH DETAILS**, you can see that SSH is enabled through port 22.



Note: In this example we created a Linux VM from a standard distribution of the operating system. If you want your VM to interact with other Microsoft Azure services, you will need to use SSH to install the CLI. You could also instantiate an image from the Image Gallery that already contains the CLI.

Logging on to a VM

Now that you have a key and have created the VM, you are ready to log on.

ssh -i ./example1.key exampleUserName@exampleVM1.cloudapp.net -p22

The 22 in the p22 parameter is the SSH port. The ./example1.key value is the path to the certificate.

The first time you log on to a newly created VM, OpenSSL will issue a warning, such as the following.

The authenticity of host 'examplevm1.cloudapp.net (138.91.78.126)' can't be established.

RSA key fingerprint is 2b:31:83:7f:3d:74:a2:3f:59:92:d4:14:ad:2e:86:ad.

Are you sure you want to continue connecting (yes/no)?

If you are comfortable proceeding, respond with **yes**. You will not be asked again. Here is an example of logging onto a VM for the first time.

```
exampleUserName@exampleVM1: ~ — bash — 84×7

mac osx> ssh -i ./example1.key exampleUserName@exampleVM1.cloudapp.net -p22
The authenticity of host 'examplevm1.cloudapp.net (138.91.78.126)' can't be establis hed.

RSA key fingerprint is 2b:31:83:7f:3d:74:a2:3f:59:92:d4:14:ad:2e:86:ad.

Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'examplevm1.cloudapp.net,138.91.78.126' (RSA) to the list of known hosts.
```

After you log on for the first time, the same OpenSSL command will log you onto the VM without a warning and without requiring a password. Here is an example.

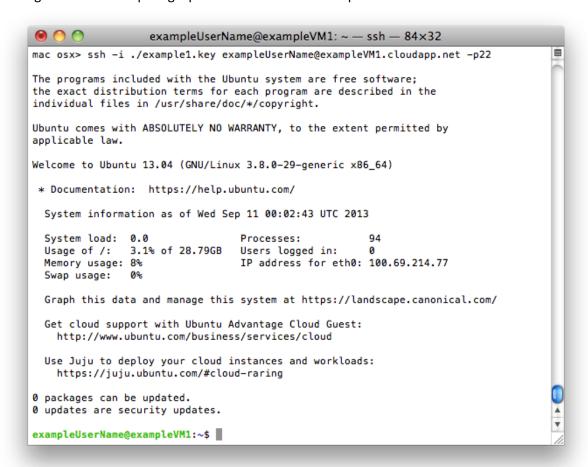


Figure 10

Opening a Port

Ports other than 22 are closed by default on Microsoft Azure VMs that run Linux.

In Microsoft Azure, all load balancer ports are closed by default. You need to explicitly open any ports you want to use. Enabling the port on the load balancer allows packets to be delivered to the VM. In this example, the VM does not have its own firewall. If it did, that would have to be configured as well.

You may want to open additional ports on your VM, depending on the server's intended role. For example, the following command opens port 80 so that a web server can accept traffic.

azure vm endpoint create-multiple exampleVM1 80 80

On the command line, the first **80** specifies the public port on the Microsoft Azure Load Balancer, and the second **80** specifies the private port used on VM that receives packets forwarded by the load balancer. The public and private port numbers don't have to be the same. Here is the output for this example.



Figure 11

Shutting Down a VM Instance

Here's an example of shutting down a VM instance. This stops the VM temporarily.

azure vm shutdown exampleVM1

Here is the output.

```
Terminal — bash — 84×6

mac osx> azure vm shutdown exampleVM1
info: Executing command vm shutdown
+ Getting virtual machines
+ Shutting down VM
info: vm shutdown command OK
mac osx>
```

Figure 12

There are no VM charges for VM instances that are stopped. The VM's operating system drive and other drives you've attached (not shown in this document, but see "azure vm disk -h") are persisted into Microsoft Azure Blob Storage for safe keeping until you need them again.

You can also shutdown (stop) the VM from the Management Portal. Here is an example.

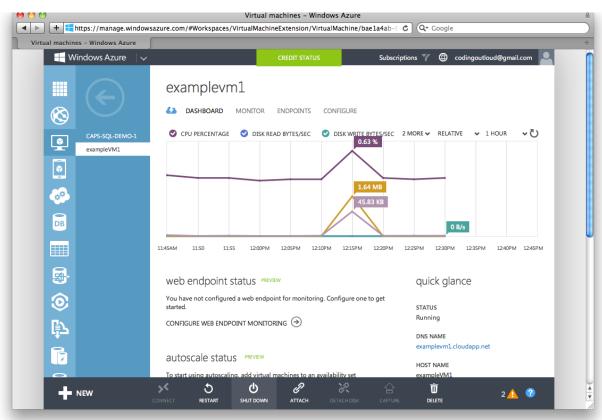


Figure 13

You are warned that you will lose your IP address (another will be assigned when you start the VM again).

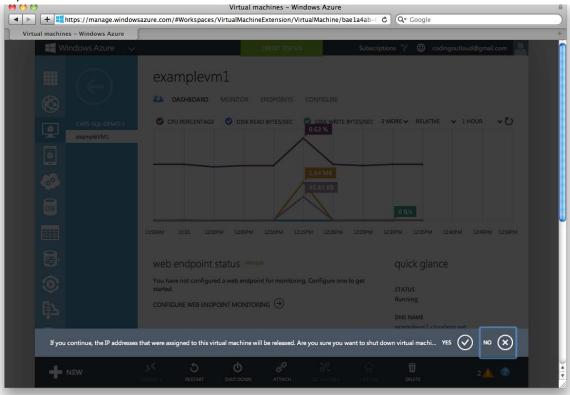


Figure 14

You can see the VM's status by selecting the **Check Status** tab. Here is an example. You can see that the VM is now stopped. Note that STATUS reads **Stopped (Deallocated)** in the dashboard.

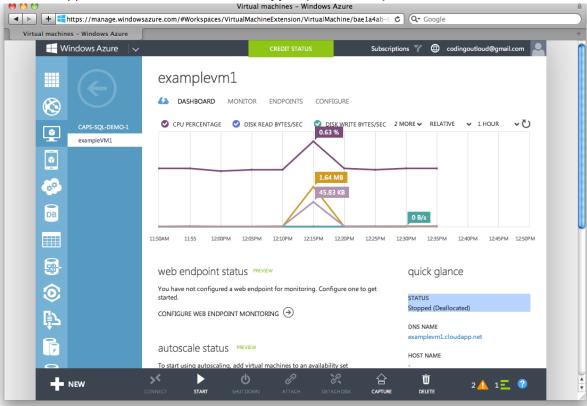


Figure 15

Starting a VM Instance

Here is an example of starting an existing VM instance.

azure vm start exampleVM1

Here is the output.

```
Terminal — bash — 84×6

mac osx> azure vm start exampleVM1
info: Executing command vm start
+ Getting virtual machines
+ Starting VM
info: vm start command OK
mac osx>
```

Figure 16

Using Python to Upload Text and Binary Data to Microsoft Azure Blob Storage

Microsoft Azure Blob storage is a service for storing large amounts of data that can be accessed from anywhere in the world via HTTP or HTTPS. A single blob can be hundreds of gigabytes in size, and a single storage account can contain up to 200TB of blobs. Common uses of Blob storage include serving images or documents directly to a browser and streaming video and audio. It is also well suited for storing and sharing scientific data sets. For more information, see Storage. Select the language you want to use. The following example uses a mixture of Python and the CLI. The code uploads a text file and a binary file.

```
# This example creates a storage account called "pytool" in the East US data center
# Creating storage account using the CLI:
      azure storage account create pytool --location 'East US'
# Viewing your available storage accounts using the CLI:
     azure storage account list
# Viewing the security account key for the "pytool" storage account using the CLI:
     azure storage account keys list pytool
from azure.storage import *
# You will need to substitute your own account name and account key
# (determined by using the CLI commands above, or the Microsoft Azure Portal)
blob_service = BlobService(account_name='pytool', account_key='0eiY2nZG...8M3q4P9s1r7A==')
storage_container_name = 'pyfiles'
blob_service.create_container(storage_container_name)
# Make any files publicly visible (default is private and private access requires readers
to have either the account_key or a Shared Access Signature URL)
blob_service.set_container_acl(storage_container_name,
                               x_ms_blob_public_access='container')
# Upload a text file and set appropriate content type
myblob = open(r'foo.txt', 'r').read()
blob_name = 'hello.txt'
blob service.put blob(storage container name, blob name, myblob,
                      x ms blob type='BlockBlob')
blob service.set blob properties(storage container name, blob name,
                                 x ms blob content type='text/plain')
# Upload a photo and set appropriate content type
myblob = open(r'clouds.jpeg', 'r').read()
blob name = 'clouds.jpeg'
blob service.put blob(storage container name, blob name, myblob,
                      x ms blob type='BlockBlob')
blob service.set blob properties(storage container name, blob name,
                                 x ms blob content type='image/jpeg')
# Show a blob listing which now includes the blobs just uploaded
blobs = blob service.list blobs(storage container name)
for blob in blobs:
    print(blob.name)
   print(blob.url)
# format for blobs is: <account>.blob.core.windows.net/<container>/<file>
# example blob for us: pytool.blob.core.windows.net/pyfiles/clouds.jpeg
```

Here is an example that shows how to read one of the blobs that was uploaded. It uses wget from a Linux workstation.

Figure 17

If you want to upload very large files, such as a VHD, you may want to use third-party tools such as <u>Cloud Data Sync</u> (CDS) from <u>Elastacloud</u>. You may also want to read <u>Creating and Uploading a Virtual</u> Hard Disk that Contains the Linux Operating System.

If you want to upload very large files, such as a VHD, and are using Windows, then the AzCopy utility is the recommended choice. For more information, see AzCopy - Uploading/Downloading files for Microsoft Azure Blobs.

There is a PowerShell command that you can use to upload VHDs to Microsoft Azure.

Add-AzureVhd

For more information about how to use this command, see <u>Uploading and Downloading VHDs to</u> Microsoft Azure.

Here is the CLI equivalent.

azure storage blob upload [options] [file] [container] [blob]

Deploying to Microsoft Azure from a Source Repository

This example shows how to automatically deploy to Microsoft Azure from GitHub.

The Python SDK for Microsoft Azure can be used with any Python application, including a web site. In addition, Microsoft Azure web sites allow you to easily deploy to them from many source repositories such as GitHub and Dropbox.

Start with a Python web application such as Django or Flask that is already available. You can retrieve a simple example from the GitHub repository codingoutloud/flasky. You can then associate the application with a Microsoft Azure web site so that it is auto-deployed whenever you commit your files to GitHub. Here is an example of a command that ties a specific Microsoft Azure web site to a specific repo in a GitHub account.

azure site create SomeWebSite --github --githubusername codingoutloud --githubpassword myG1tHuBSecretP@ssword --githubrepository codingoutloud/somerepo Here is the output.

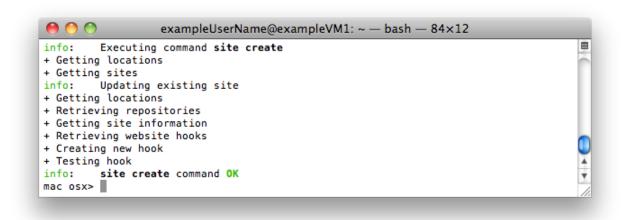


Figure 18

You can also see the result of this command in the Management Portal. Here is an example.

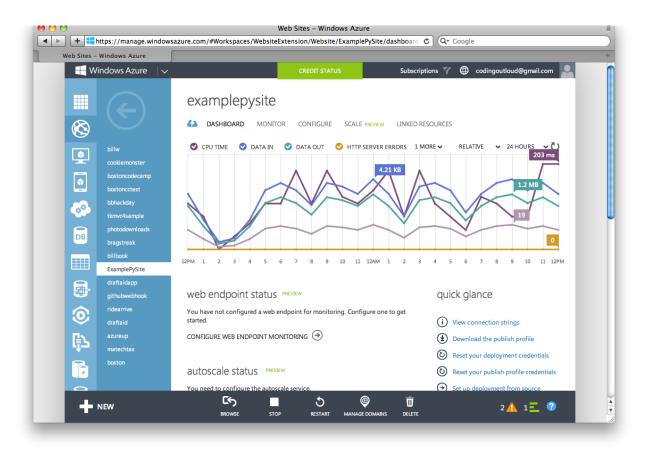


Figure 19

Summary of Web Sites

This page lists the web sites mentioned in this article.

For an overview of the Microsoft Azure platform, read <u>Intro to Microsoft Azure</u>. (http://azure.microsoft.com/en-us/documentation/articles/fundamentals-introduction-to-azure/)

To learn more about using Linux in a Microsoft Azure VM, go to the <u>Microsoft Azure Virtual Machines</u> documentation. (http://azure.microsoft.com/en-us/documentation/services/virtual-machines/)

To learn how to create a Linux VM with the Azure Management Portal, see <u>Create a Virtual Machine Running Linux</u>. (http://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-tutorial/)

To learn more about uploading very large files, such as a VHD, see <u>Creating and Uploading a Virtual Hard Disk that Contains the Linux Operating System</u>. (http://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-create-upload-vhd/)

To learn more about using PowerShell to upload very large files, see <u>Uploading and Downloading VHDs</u> <u>to Microsoft Azure</u>. (http://michaelwasham.com/windows-azure-powershell-reference-guide/uploading-and-downloading-vhds-to-windows-azure/)

For more information about using the Microsoft Azure CLI tools for Mac and Linux, read <u>Install and Configure the Azure Cross-Platform Command-Line Interface</u>. (http://azure.microsoft.com/en-us/documentation/articles/xplat-cli/)

For more information about Microsoft Azure storage, see Storage. (http://azure.microsoft.com/en-us/documentation/services/storage/)

For more information on how to create Microsoft Azure web sites, read <u>Web Sites</u>. (http://azure.microsoft.com/en-us/documentation/services/web-sites/)

If you use Java, go the <u>Microsoft Azure Java Developer Center</u> for resources and tutorials. (http://azure.microsoft.com/en-us/develop/java/)

If you use Node.js, go to the <u>Microsoft Azure Node.js Developer Center</u> for resources and tutorials. (http://azure.microsoft.com/en-us/develop/nodejs/)

If you use PHP, go to the <u>Microsoft Azure PHP Developer Center</u> for resources and tutorials. (http://azure.microsoft.com/en-us/develop/php/)

If you use Ruby, go to the <u>Microsoft Azure Ruby Developer Center</u> for resources and tutorials. (http://azure.microsoft.com/en-us/develop/ruby/)

If you use Python, go to the <u>Microsoft Azure Python Developer Center</u> for resources and tutorials. (http://azure.microsoft.com/en-us/develop/python/)

For an overview of using key pairs to create Linux VMs, see <u>Creating secure Linux VMs in Azure with SSH key pairs</u>. (http://www.jeff.wilcox.name/2013/06/secure-linux-vms-with-ssh-certificates/)

Third-Party Tools

This section lists the third-party tools mentioned in this article.

<u>VMware Fusion</u> lets you run more than 200 operating systems as virtual machines including Windows XP through Windows 8. (http://www.vmware.com/products/fusion/)

<u>Parallels Desktop 9</u> lets you run both Windows and Mac OS X applications side-by-side without rebooting. You can drag and drop files between Windows and Mac applications and launch Windows applications from your Mac dock. (http://www.parallels.com/products/desktop/)

<u>Elastacloud CDS</u> supports CDS supports migrations from a variety of data sources, such as Amazon S3 storage, into an Azure blob storage account. (http://www.elastacloud.com/Product/Details/3)

A set of cross-platform tools for Microsoft Azure is available on GitHub at https://github.com/Azure/azure-sdk-tools-xplat.

A popular cross-platform editor is Sublime. (http://www.sublimetext.com/)

© 2013 Microsoft Corporation. All rights reserved. Except where otherwise noted, content on this site is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.