IT Innovation Series:   
ATTENDEE GUIDE

“What’s new in Azure Infrastructure: Using Open Source and Other Technologies”

## Table of Contents

[Table of Contents 1](#_Toc452062032)

[Attendee guide 3](#_Toc452062033)

[What this guide is for 3](#_Toc452062034)

[Today’s goals 3](#_Toc452062035)

[Event format 3](#_Toc452062036)

[Agenda 3](#_Toc452062037)

[Setting up an Azure subscription for this lab 5](#_Toc452062038)

[Create a new Azure Trial Account 5](#_Toc452062039)

[Activating a Microsoft Azure Pass (no Creditcard required) 5](#_Toc452062040)

[Notes about Monetary Credit 6](#_Toc452062041)

[Install and get to know the Cross Platform Command-Line Interface 7](#_Toc452062042)

[Use an installer for the Cross Platform CLI 7](#_Toc452062043)

[Connect to an Azure subscription from the Azure Command-Line Interface (Azure CLI) 7](#_Toc452062044)

[Use azure login to authenticate interactively 8](#_Toc452062045)

[Use azure login with a username and password 9](#_Toc452062046)

[Multiple subscriptions 10](#_Toc452062047)

[CLI command modes 10](#_Toc452062048)

[Storage of CLI settings 11](#_Toc452062049)

[Logging out 11](#_Toc452062050)

[Using Azure CLI commands in Resource Manager mode 12](#_Toc452062051)

[azure group: Commands to manage your resource groups 14](#_Toc452062052)

[azure location: Commands to get the available locations for all resource types 15](#_Toc452062053)

[azure network: Commands to manage network resources 15](#_Toc452062054)

[azure provider: Commands to manage resource provider registrations 16](#_Toc452062055)

[azure storage: Commands to manage your Storage objects 18](#_Toc452062056)

[azure vm: Commands to manage your Azure Virtual Machines 19](#_Toc452062057)

[Using Azure CLI Commands to create a Linux VM 21](#_Toc452062058)

[Installing and using GIT as a repository 29](#_Toc452062059)

[Using ARM templates to further automate VM Deployments 33](#_Toc452062060)

[Use the Azure CLI for Mac, Linux, and Windows with Azure Resource Manager 33](#_Toc452062061)

[Authentication 34](#_Toc452062062)

[Set the Resource Manager mode 34](#_Toc452062063)

[Create a resource group 34](#_Toc452062064)

[Use resource group templates 35](#_Toc452062065)

[Export a resource template 37](#_Toc452062066)

[At home: Resources to continue learning 40](#_Toc452062067)

# Attendee guide

### How to use this guide

* Now: Review [Attendee guide](#_Overview) to understand event goals, format and agenda
* Today: Follow the [Step-by-step guidance](#_Step-by-step_Guidance) to complete the technical exercises
* At home: Review [Next Steps](#_At_home:_Resources) for continued learning

## What this guide is for

Thank you for joining us for today’s IT Innovation Series event, focused on “What’s new in Cloud Infrastructure: Improving Datacenter Flexibility with Microsoft, Open Source and Other Technologies”. This document will guide you through today’s instructor-led and individual activities, and provide you with the key resources and information necessary to continue building your skills to be able to implement in your environment.

## Today’s goals

By the time you leave today, you should walk away with:

* A technical understanding of the latest Azure Infrastructure Services and how to use the cross platform CLI
* The knowledge to configure and upload a custom Linux VM
* An understanding of using Docker containers and Microsoft Azure

## Event format

Today’s event will include a combination of presentation, short activities and hands-on work, organized in a step-by-step approach to lead you through overview to implementation. You will use your personal laptop and a free Azure Pass we’ll provide to complete today’s activities directly in Azure.

## Agenda

|  |  |  |
| --- | --- | --- |
| 10:00 – 11:00 |  | Microsoft en Open Source : overzicht en lab set-up |
| 11:00 – 12:30 |  | OSS Deployment met ARM templates hands-on lab |
| 12:30 - 13:30 |  | LUNCH |
| 13:30 – 15:00 |  | Custom Suse Linux image maken en uitrollen met Suse Studio en Azure |
| 15:00 - 15:15 |  | PAUZE |
| 15:15 – 16:50 |  | Docker on Linux Hands on Lab |
| 16:50 – 17:00 |  | Wrap-Up |
| 17:00 |  | Netwerkborrel |

Today: Step-by-step guidance

Following the introduction and overview, you’ll have the chance to try out real-world scenarios directly in our preconfigured labs, aimed to help give you the skills necessary to deploy in your own environment.

#### Use of Own System

You may complete lab instructions using your own workstation, (Windows, Linux or OSX) providing you download the files used for the lab from GitHub and have the following software installed.

#### GitHub repository for Lab Files

If you are not using the hosted virtual machine and are using your own workstation, any custom files the lab instruction call out can be found in a GitHub repository. The repository is located here: <https://github.com/robert-bakker/AccelerateLab> .

#### Required Software

| Description | Steps |
| --- | --- |
| Required software will be called out throughout the lab. Detailed instructions are in the lab steps.  You can choose to install them now or when you reach these steps. Any additional software that you require will be called out in the lab. The following software may be useful when working with Azure in general. | 1. Visual Studio Code - <https://code.visualstudio.com/> 2. GitHub Desktop - <https://desktop.github.com/> 3. Microsoft Azure Cross-platform Command Line Tools (for Windows, OSX or Linux) - <https://azure.microsoft.com/en-us/downloads/> 4. Install the JQ JSON parser - <https://stedolan.github.io/jq/download/> 5. Install the latest Node JS for your platform (v6.2.0 at the time of writing) from <https://nodejs.org/en/> 6. Chocolatey - <https://chocolatey.org/install> or |

# 

# Setting up an Azure subscription for this lab

This IT Innovation Series event lab requires a valid Azure subscription. While you may use an existing subscription such as a subscription associated MSDN account or existing corporate account, it would preferable to use an Azure Trial subscription for this event. By using a trial subscription, you will avoid any charges against your MSDN or corporate subscription that would result from doing the activities in this camp.

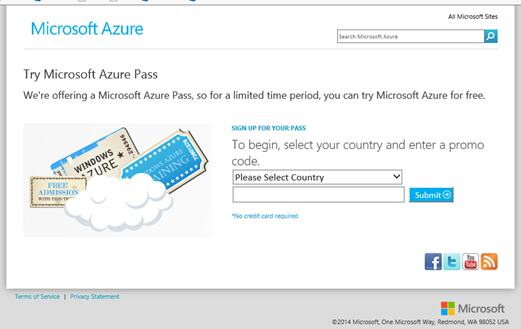
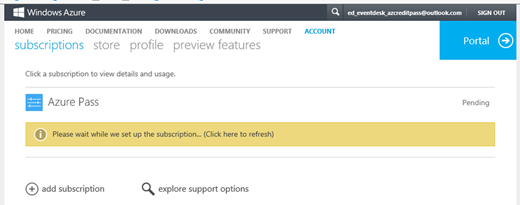
Your instructor may be able to provide you with an Azure Pass. Or, you may use a CLEAN and UNUSED Azure Trial account - details on how to set one up are detailed below.

## Create a new Azure Trial Account

| Description | Steps |
| --- | --- |
| Create a new Azure trial account perform the following steps | Open an InPrivate window in your browser  Navigate to [www.live.com](http://www.live.com) and click **Sign up now**.  Follow the on-screen instructions to create a new Microsoft Account.  Navigate to [www.azure.com](http://www.azure.com) and click **Free Trial**.  Follow the on-screen instructions to activate a new Windows Azure Trial.  Navigate to Manage.windowsazure.com and sign in.  In Microsoft Azure portal, in the upper left, click your user name, and then click **View my bill**.  Click your current trial subscription, and then click **Edit subscription details.**  Type a name you will recognize in SUBSCRIPTION NAME, such as ITCamps, and then click the **Done** icon. |

## Activating a Microsoft Azure Pass (no Creditcard required)

To activate a Microsoft Azure Pass, follow the instructions provided with your azure pass code. To be sure, here are the steps to follow:

1. Go to <http://microsoftazurepass.com>. Select your country, type in the provided promo code and click “Submit.”  
   
2. Sign into a Microsoft account to continue. If you have an account password saved or you automatically sign into other Microsoft services that Microsoft account will be used to redeem the promo code.
3. The subscription will take 3-10 minutes to activate.  
   

## Notes about Monetary Credit

* Accounts are activated within minutes of the promo code being redeemed
* Users can view the cost of account usage
* User can enable public preview features
* Accounts cannot be extended - the duration and monetary credit amount cannot be changed
* Only one promo code can be redeemed per the life of a Microsoft account (LiveID)
* Users who have used an Azure Trial can redeem a promo code. Users are able to redeem one trial and one promo code per account.
* If a user has already redeemed a promo code, he/she will need to create a new Microsoft account.

For more detailed instructions on how to activate your Azure monetary credit, please visit <https://www.microsoftazurepass.com/howto>

# Install and get to know the Cross Platform Command-Line Interface

You will start by familiarizing yourself with the Microsoft Azure Cross Platform Command Line Interface (xplat-cli) and configuring your Azure subscription.

Quickly install the Azure Command-Line Interface (Azure CLI) to use a set of open-source shell-based commands for creating and managing resources in Microsoft Azure. You have several installation choices: use one of the provided installer packages for different operating systems, install from an npm package, or install the Azure CLI as a container in a Docker host. For more options and background, see the project repository on GitHub <https://github.com/azure/azure-xplat-cli> .

Once the Azure CLI has been installed, you will be able to connect it with your Azure subscription and run the **azure** commands from your command-line interface (Bash, Terminal, Command prompt, and so on) to work with your Azure resources.

## Use an installer for the Cross Platform CLI

The following installer packages are available on <https://azure.microsoft.com/en-us/downloads/>

[Windows installer](https://www.microsoft.com/web/handlers/webpi.ashx?command=getinstallerredirect&appid=windowsazurexplatcli&mode=new) (<https://www.microsoft.com/web/handlers/webpi.ashx?command=getinstallerredirect&appid=windowsazurexplatcli&mode=new>)

[OS X installer](OS%20X%20installer) (http://aka.ms/mac-azure-cli)

[Linux installer](http://aka.ms/linux-azure-cli) (http://aka.ms/linux-azure-cli)

## Connect to an Azure subscription from the Azure Command-Line Interface (Azure CLI)

The Azure CLI is a set of open-source, cross-platform commands for working with the Azure platform. This article describes ways to provide your Azure account credentials to connect the Azure CLI to your Azure subscription. If you haven't already installed the CLI, see Install the Azure CLI above. If you don't have an Azure subscription, you can create a [free account](http://azure.microsoft.com/free/) in just a couple of minutes (credit card required) or use a provided Microsoft Azure Pass.

* **Log in to Azure using a work or school account or a Microsoft account identity** - Use the azure login command in CLI version 0.9.10 and above with either type of account identity to authenticate through Azure Active Directory. The CLI (version 0.9.9 and above) also supports interactive authentication through a web portal for accounts that have enabled multi-factor authentication. Also use the azure login command to authenticate a service principal for an Azure Active Directory application, which is useful for running automated services. After logging in with a supported account identity, you can use either Azure Resource Manager mode or Azure Service Management mode commands.
* If you are using a version of the Azure CLI that is prior to version 0.9.10, you can use the azure login command only with a work or school account; Microsoft account identities do not work. However, if you want, you can [create a work or school ID from your Microsoft account ID](https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-windows-create-aad-work-id/). https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-windows-create-aad-work-id/

For background about different account identities and Azure subscriptions, see [How Azure subscriptions are associated with Azure Active Directory](https://azure.microsoft.com/en-us/documentation/articles/active-directory-how-subscriptions-associated-directory/). https://azure.microsoft.com/en-us/documentation/articles/active-directory-how-subscriptions-associated-directory/

## Use azure login to authenticate interactively

Use the azure login command -- without any arguments -- to authenticate interactively with either:

a work or school account identity (also called an organizational account) that requires multi-factor authentication, or

a Microsoft account identity when you want to access Resource Manager mode commands

* In both cases, authentication and authorization are performed using Azure Active Directory. If you use a Microsoft account identity, the log in process accesses your Azure Active Directory default domain. (If you signed up for a free Azure account, you might not be aware that Azure Active Directory created a default domain for your account.)

Interactively logging in is easy: type azure login and follow the prompts as shown below:

azure login

info: Executing command login

info: To sign in, use a web browser to open the page http://aka.ms/devicelogin. Enter the code XXXXXXXXX to authenticate. If you're signing in as an Azure AD application, use the --username and --password parameters.

Copy the code offered to you, above, and open a browser to http://aka.ms/devicelogin. Enter the code, and then you are prompted to enter the username and password for the identity you want to use. When that process completes, the command shell completes the log in process. It might look something like:

info: Added subscription Visual Studio Ultimate with MSDN

info: Added subscription Azure Free Trial

info: Setting subscription "Visual Studio Ultimate with MSDN" as default

+

info: login command OK

## Use azure login with a username and password

Use the azure login command with a username parameter or with both a username and a password to authenticate when you want to use a work or school account that doesn't require multi-factor authentication. The following example passes the username of an organizational account:

azure login -u robakk@nlazurepro.onmicrosoft.com

info: Executing command login

Password: \*\*\*\*\*\*\*\*\*

/info: Added subscription Internal

+

info: login command OK

Enter your password when prompted.

If this is your first time logging in with these credentials, you are asked to verify that you wish to cache an authentication token. This prompt also occurs if you have previously used the azure logout command (described later in the article). To bypass this prompt for automation scenarios, run azure login with the -q parameter.

## Multiple subscriptions

If you have multiple Azure subscriptions, connecting to Azure will grant access to all subscriptions associated with your credentials. One subscription is selected as the default, and used by the Azure CLI when performing operations. You can view the subscriptions, as well as which one is the default, using the azure account list command. This command returns information similar to the following:

info: Executing command account list

data: Name Id Current

data: ---------------- ------------------------------------ -------

data: Azure-sub-1 #################################### true

data: Azure-sub-2 #################################### false

In the above list, the **Current** column indicates the current default subscription as Azure-sub-1. To change the default subscription, use the azure account set command, and specify the subscription that you wish to be the default. For example:

azure account set Azure-sub-2

This changes the default subscription to Azure-sub-2.

* Changing the default subscription takes effect immediately, and is a global change; new Azure CLI commands, whether you run them from the same command-line instance or a different instance, use the new default subscription.

If you wish to use a non-default subscription with the Azure CLI, but don't want to change the current default, you can use the --subscription option for the command and provide the name of the subscription you wish to use for the operation.

Once you are connected to your Azure subscription, you can start using the Azure CLI commands to work with Azure resources.

## CLI command modes

The Azure CLI provides two command modes for working with Azure resources, with different command sets:

* **Resource Manager mode** - for working with Azure resources in the Resource Manager deployment model. To set this mode, run azure config mode arm.
* **Service Management mode** - for working with Azure resources in the classic deployment model. To set this mode, run azure config mode asm.

When first installed, the CLI is in Service Management mode.

* The Resource Manager mode and Service Management mode are mutually exclusive. That is, resources created in one mode cannot be managed from the other mode.

## Storage of CLI settings

Whether you log in with the azure login command or import publish settings, your CLI profile and logs are stored in a .azure directory located in your user directory. Your user directory is protected by your operating system; however, it is recommended that you take additional steps to encrypt your user directory. You can do so in the following ways:

* On Windows, modify the directory properties or use BitLocker.
* On Mac, turn on FileVault for the directory.
* On Ubuntu, use the Encrypted Home directory feature. Other Linux distributions offer similar features.

## Logging out

To log out, use the following command:

azure logout -u <username>

If the subscriptions associated with the account were only authenticated with Active Directory, logging out deletes the subscription information from the local profile. However, if a publish settings file had also been imported for the subscriptions, logging out only deletes Active Directory related information from the local profile.

* If you prefer using the login method, use the following command: azure login -u username -p password
* Now that you have configured your Azure subscription in the command line you will proceed to use Azure CLI commands in Resource Manager Mode

# Using Azure CLI commands in Resource Manager mode

This module provides syntax and options for Azure command line interface (CLI) commands you'd commonly use to create and manage Azure resources in the Azure Resource Manager deployment model. You access these commands by running the CLI in Resource Manager (arm) mode.

* This is not a complete reference, and your CLI version may show slightly different commands or parameters.

To get started, first install the Azure CLI and connect to your Azure subscription by using a work or school account or a Microsoft account identity.

For current command syntax and options at the command line in Resource Manager mode, type azure help or, to display help for a specific command, azure help [command]. You'll also find CLI examples in the documentation for creating and managing specific Azure services.

Optional parameters are shown in square brackets (for example, [parameter]). All other parameters are required.

In addition to command-specific optional parameters documented here, there are three optional parameters that can be used to display detailed output such as request options and status codes. The -v parameter provides verbose output, and the -vv parameter provides even more detailed verbose output. The --json option will output the result in raw json format.

### Setting the Resource Manager mode

Use the following command to enable Azure CLI Resource Manager commands.

azure config mode arm

### Imperative and declarative approaches

The Resource Manager mode of the Azure CLI gives you commands that create resources imperatively on the command line. For example, if you type azure group create <groupname> <location> you are asking Azure to create a resource group, and with azure group deployment create <resourcegroup> <deploymentname> you are instructing Azure to create a deployment of any number of items and place them in a group. Because each type of resource has imperative commands, you can chain them together to create fairly complex deployments.

However, using resource group templates that describe a resource group is a declarative approach that is far more powerful, allowing you to automate complex deployments of (almost) any number of resources for (almost) any purpose. When using templates, the only imperative command is to deploy one. For a general overview of templates, resources, and resource groups, see [Azure Resource Group Overview](https://github.com/Azure/azure-content/blob/master/articles/resource-group-overview.md). https://github.com/Azure/azure-content/blob/master/articles/resource-group-overview.md

**Manage your account information with “azure account”**

**List the imported subscriptions**

account list [options]

**Show details about a subscription**

account show [options] [subscriptionNameOrId]

**Set the current subscription**

account set [options] <subscriptionNameOrId>

**Commands to manage your account environment**

account env list [options]

account env show [options] [environment]

**Manage your local settings**

With the config command, you can make configurations settings to the azure cli.

Azure CLI configuration settings

config list [options]

Delete a config setting

config delete [options] <name>

Update a config setting

config set <name> <value>

Set the Azure CLI working mode to either arm or asm

config mode [options] <modename>

## azure group: Commands to manage your resource groups

**An important concept to understand is the use of resource groups. A resource group is a container that holds related resources for an application. The resource group could include all of the resources for an application, or only those resources that are logically grouped together. You can decide how you want to allocate resources to resource groups based on what makes the most sense for your organization.**

**There are some important factors to consider when defining your resource group:**

**All of the resources in your group should share the same lifecycle. You will deploy, update and delete them together. If one resource, such as a database server, needs to exist on a different deployment cycle it should be in another resource group.**

1. **Each resource can only exist in one resource group.**
2. **You can add or remove a resource to a resource group at any time.**
3. **You can move a resource from one resource group to another group. For more information, see Move resources to new resource group or subscription.**
4. **A resource group can contain resources that reside in different regions.**
5. **A resource group can be used to scope access control for administrative actions.**
6. **A resource can be linked to a resource in another resource group when the two resources must interact with each other but they do not share the same lifecycle (for example, multiple apps connecting to a database).**

Create a new resource group

group create [options] <name> <location>

Set tags to a resource group

group set [options] <name> <tags>

Delete a resource group

group delete [options] <name>

Lists the resource groups for your subscription

group list [options]

Shows a resource group for your subscription

group show [options] <name>

Commands to manage resource group logs

group log show [options] [name]

Commands to manage your deployment in a resource group

group deployment create [options] [resource-group] [name]

group deployment list [options] <resource-group> [state]

group deployment show [options] <resource-group> [deployment-name]

group deployment stop [options] <resource-group> [deployment-name]

## azure location: Commands to get the available locations for all resource types

List the available locations

location list [options]

## azure network: Commands to manage network resources

**Commands to manage virtual networks**

To see all commands : azure network -h

azure network vnet create [options] <resource-group> <name> <location>

Allows to create a new virtual network. In the following example we create a virtual network named newvnet for resource group myresourcegroup in the West US region.

azure network vnet create myresourcegroup newvnet "west us"

info: Executing command network vnet create

+ Looking up virtual network "newvnet"

+ Creating virtual network "newvnet"

Loading virtual network state

data: Id: /subscriptions/###############################/resourceGroups/myresourcegroup/providers/Microsoft.Network/virtualNetworks/newvnet

data: Name: newvnet

data: Type: Microsoft.Network/virtualNetworks

data: Location: westus

data: Tags:

data: Provisioning state: Succeeded

data: Address prefixes:

data: 10.0.0.0/8

data: DNS servers:

data: Subnets:

data:

info: network vnet create command OK

To update a virtual network configuration within a resource group.

azure network vnet set myresourcegroup newvnet

info: Executing command network vnet set

+ Looking up virtual network "newvnet"

+ Updating virtual network "newvnet"

+ Loading virtual network state

data: Id: /subscriptions/###############################/resourceGroups/myresourcegroup/providers/Microsoft.Network/virtualNetworks/newvnet

data: Name: newvnet

data: Type: Microsoft.Network/virtualNetworks

data: Location: westus

data: Tags:

data: Provisioning state: Succeeded

data: Address prefixes:

data: 10.0.0.0/8

data: DNS servers:

data: Subnets:

data:

info: network vnet set command OK

network vnet list [options] <resource-group>

The command allows to list all virtual networks in a resource group.

C:\>azure network vnet list myresourcegroup

info: Executing command network vnet list

+ Listing virtual networks

data: ID

Name Location Address prefixes DNS servers

data: -------------------------------------------------------------------

------ -------- -------- ---------------- -----------

data: /subscriptions/###############################/resourceGroups/

wvnet newvnet westus 10.0.0.0/8

info: network vnet list command OK

Parameter options:

-h, --help output usage information

-v, --verbose use verbose output

--json use json output

-g, --resource-group <resource-group> the name of the resource group

-s, --subscription <subscription> the subscription identifier

The command removes a virtual network.

azure network vnet delete myresourcegroup newvnet

Commands to create a virtual network subnet

Azure network vnet subnet create [options] <resource-group> <vnet-name> <name>

This command allows to add another subnet to an existing virtual network.

azure network vnet subnet create -g myresourcegroup --vnet-name newvnet -n subnet --address-prefix 10.0.1.0/24

## azure provider: Commands to manage resource provider registrations

A resource provider is a service that supplies the resources you can deploy and manage through Resource Manager. Each resource provider offers REST API operations for working with the resources. For example, if you want to deploy an Azure Key Vault for storing keys and secrets, you will work with the **Microsoft.KeyVault** resource provider.

To deploy and manage your infrastructure, you will need to know details about the resource providers; such as, what resource types it offers, the version numbers of the REST API operations, the operations it supports, and the schema to use when setting the values of the resource type to create.

List currently registered providers in ARM

provider list [options]

Show details about the requested provider namespace

provider show [options] <namespace>

Register provider with the subscription

provider register [options] <namespace>

Unregister provider with the subscription

provider unregister [options] <namespace>

## azure storage: Commands to manage your Storage objects

Many operations in Azure concern storage

Commands to manage your Storage accounts

storage account list [options]

storage account show [options] <name>

storage account create [options] <name>

Commands to manage your Storage account keys

storage account keys list [options] <name>

storage account keys renew [options] <name>

Commands to show your Storage connection string

storage account connectionstring show [options] <name>

Commands to manage your Storage blobs

storage blob list [options] [container] [prefix]

storage blob show [options] [container] [blob]

storage blob delete [options] [container] [blob]

## azure vm: Commands to manage your Azure Virtual Machines

Create a VM

vm create [options] <resource-group> <name> <location> <os-type>

Create a VM with default resources

vm quick-create [options] <resource-group> <name> <location> <os-type> <image-urn> <admin-username> <admin-password

* Starting with CLI version 0.10, you can provide a short alias such as "OpenSuse" or "Win2012R2Datacenter" as the image-urn for some popular Marketplace images. Run azure help vm quick-create for options. Additionally, starting with version 0.10, azure vm quick-create uses premium storage by default if it's available in the selected region.

List the virtual machines within an account

vm list [options]

Get one virtual machine within a resource group

vm show [options] <resource-group> <name>

Delete one virtual machine within a resource group

vm delete [options] <resource-group> <name>

Shutdown one virtual machine within a resource group

vm stop [options] <resource-group> <name>

Restart one virtual machine within a resource group

vm restart [options] <resource-group> <name>

Start one virtual machine within a resource group

vm start [options] <resource-group> <name>

Shutdown one virtual machine within a resource group and releases the compute resources

vm deallocate [options] <resource-group> <name>

List available virtual machine sizes

vm sizes [options]

Capture the VM as OS Image or VM Image

vm capture [options] <resource-group> <name> <vhd-name-prefix>

Enable you to reset Remote Desktop Access or SSH settings on a Virtual Machine and to reset the password for the account that has administrator or sudo authority

vm reset-access [options] <resource-group> <name>

Commands to manage your Virtual Machine data disks

vm disk attach-new [options] <resource-group> <vm-name> <size-in-gb> [vhd-name]

Commands to manage VM resource extensions

vm extension set [options] <resource-group> <vm-name> <name> <publisher-name> <version>

vm extension get [options] <resource-group> <vm-name>

Commands to manage VM images

vm image list-publishers [options] <location>

vm image list-offers [options] <location> <publisher>

vm image list-skus [options] <location> <publisher> <offer>

vm image list [options] <location> <publisher> [offer] [sku]

## Using Azure CLI Commands to create a Linux VM

There are multiple ways to create environmens in Microsoft Azure. In this exercise we will look at two different ones; creating a vm using quick-create and after that using the create command for more granularity and control.

### Method 1: Using the Azure CLI command Azure VM Create:

Using the azure vm create gives you more control on parameters than using quick-create. With this command you can specify networks, subnets, storage accounts etc at the time of deployment.

* First delete the resource groups created in the previous exercises to start with a new resource group. Important: all resources you previously created will be lost.

View all resource groups in your subscription

azure group list

Then delete the specific resource group:

azure group delete [resource group name]

Create a new storage group using the last 4 digits of your mobile phone number as a unique identifier

azure group create accelerate[uniqueid] -l westeurope

Verify the RG using the JSON parser

azure group show accelerate[unique] --json | jq '.'

Create the Storage Account

azure storage account create -g Accelerate[UNIQUEID] -l westeurope --type GRS accelerate[UNIQUEID]storage

Verify the storage using the JSON parser

azure storage account show -g accelerate[UNIQUEID] accelerate[UNIQUEID]storage --json | jq '.'

Create the Virtual Network

azure network vnet create -g accelerate[UNIQUEID] -n acceleratevnet -a 192.168.0.0/16 -l westeurope

Create the Subnet

azure network vnet subnet create -g accelerate[UNIQUEID] -e acceleratevnet -n FrontEnd -a 192.168.1.0/24

Verify the VNet and Subnet using the JSON parser

azure network vnet show accelerate[UNIQUEID] acceleratevnet --json | jq '.'

Create a public IP

azure network public-ip create -g accelerate[UNIQUEID] -n acceleratelbpip -l westeurope -d acceleratelb -a static -i 4

Create the load balancer

azure network lb create -g accelerate[UNIQUEID] -n acceleratelb -l westeurope

Create a front-end IP pool for the load balancer, and associate our public IP

azure network lb frontend-ip create -g accelerate[UNIQUEID] -l acceleratelb -n acceleratefrontendpool -i acceleratelbpib

Create our back-end IP pool for the load balancer

azure network lb address-pool create -g accelerate[UNIQUEID] -l Acceleratelb -n Acceleratefrontendpool

Create SSH inbound NAT rules for the load balancer

azure network lb inbound-nat-rule create -g Accelerate[UNIQUEID] -l Acceleratelb -n VM1-SSH -p tcp -f 4222 -b 22

azure network lb inbound-nat-rule create -g Accelerate[UNIQUEID] -l Acceleratelb -n VM2-SSH -p tcp -f 4223 -b 22

Create our web inbound NAT rules for the load balancer

azure network lb rule create -g Accelerate[UNIQUEID] -l Acceleratelb -n WebRule -p tcp -f 80 -b 80 -t Acceleratefrontendpool -o Acceleratefrontendpool

Create our load balancer health probe

azure network lb probe create -g Accelerate[UNIQUEID] -l Acceleratelb -n HealthProbe -p "http" -f healthprobe.aspx -i 15 -c 4

Verify the load balancer, IP pools, and NAT rules using the JSON parser

azure network lb show -g Accelerate[UNIQUEID] -n Acceleratelb --json | jq '.'

Create the first NIC

azure network nic create -g Accelerate[UNIQUEID] -n LB-NIC1 -l westeurope --subnet-vnet-name Acceleratevnet --subnet-name FrontEnd

Create the second NIC

azure network nic create -g Accelerate[UNIQUEID] -n LB-NIC2 -l westeurope --subnet-vnet-name Acceleratevnet --subnet-name FrontEnd

Verify the NICs using the JSON parser

azure network nic show Accelerate[UNIQUEID] LB-NIC1 --json | jq '.'

azure network nic show Accelerate[UNIQUEID] LB-NIC2 --json | jq '.'

Create the NSG

azure network nsg create -g Accelerate[UNIQUEID] -n AccelerateNSG -l westeurope

Add the inbound rules for the NSG

azure network nsg rule create --protocol tcp --direction inbound --priority 1000 --destination-port-range 22 --access allow -g Accelerate[UNIQUEID] -a AccelerateNSG -n SSHRule

azure network nsg rule create --protocol tcp --direction inbound --priority 1001 --destination-port-range 80 --access allow -g Accelerate[UNIQUEID] -a AccelerateNSG -n HTTPRule

Verify the NSG and inbound rules using the JSON parser

azure network nsg show -g Accelerate[UNIQUEID] -n AccelerateNSG --json | jq '.'

Bind the NSG to the NICs

azure network nic set -g Accelerate[UNIQUEID] -n LB-NIC1 -o AccelerateNSG

azure network nic set -g Accelerate[UNIQUEID] -n LB-NIC2 -o AccelerateNSG

Create the availability set

azure availset create -g Accelerate[UNIQUEID] -n AccelerateAvailSet -l westeurope

Create the first Linux VM (type as one command)

azure vm create \

--resource-group Accelerate[UNIQUEID] \

--name LinuxVM1 \

--location westeurope \

--os-type linux \

--availset-name AccelerateAvailSet \

--nic-name LB-NIC1 \

--vnet-name Acceleratevnet \

--vnet-subnet-name FrontEnd \

--storage-account-name Accelerate[UNIQUEID]storage \

--image-urn suse:openSUSE:13.2:2015.05.12 \

--admin-username ops

* To shorten the command, you can also use the options listed when you type in azure help vm create.

Create the second Linux VM

azure vm create \

--resource-group Accelerate[UNIQUEID] \

--name LinuxVM2 \

--location westeurope \

--os-type linux \

--availset-name AccelerateAvailSet \

--nic-name LB-NIC1 \

--vnet-name Acceleratevnet \

--vnet-subnet-name FrontEnd \

--storage-account-name Accelerate[UNIQUEID]storage \

--image-urn suse:openSUSE:13.2:2015.05.12 \

--admin-username ops

Verify everything built using the JSON parser

azure vm show -g Accelerate[UNIQUEID] -n LinuxVM1 --json | jq '.'

azure vm show -g Accelerate[UNIQUEID] -n LinuxVM2 --json | jq '.'

Method 2: Using the Azure CLI command Azure VM Quick-Create:

As you noticed, manually typing all commands to create a working solution, while fully customizable, takes quite some time and effort. For quicker deployments, you can use the azure quick-create command.

First delete the resource group created in the previous exercises. Important: all resources you previously created will be lost.

View all resource groups in your subscription

azure group list

Then delete the specific resource group:

azure group delete [resource group name]

Create a new storage group using the last 4 digits of your mobile phone number as a unique identifier

azure group create accelerate[uniqueid] -l westeurope

From a terminal or command prompt, run the following command to list all the available locations from which you can choose to create a virtual machine. Take note of one of them (e.g.: *WestEurope*); as you will use it in the following step.

azure location list

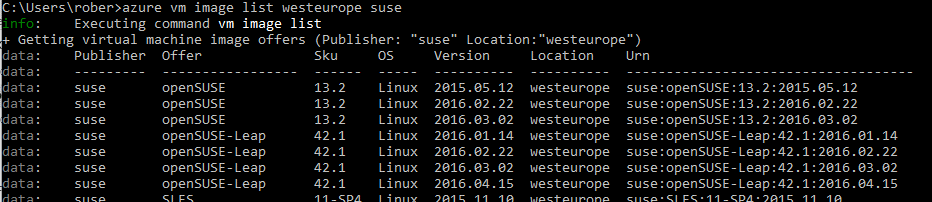
Select the Linux image you would like to deploy.

* In this example we choose OpenSuse 13.2

Get a list of all available images :

azure vm image list westeurope suse

Note the Image URN



* For a select number of images in the Azure Marketplace, aliases are available. So you can use CentOS, Debian, OpenSuse, RHEL for example as the URN

To create a new virtual machine based on this image run the following command. Replace the values in [ ] with your specific values for the virtual machine name and administrator user.

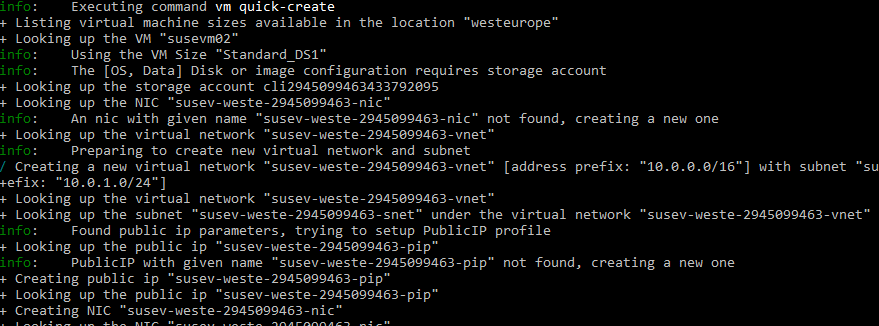
As the command executes it will prompt you to enter the password for the admin user.

azure vm quick-create -g [RESOURCEGROUP] -n [VMNAME] -l [LOCATION] -y [OSTYPE] -Q [URN] [USER]

example: azure vm quick-create -g accelerate1574 -n susevm02 -l westeurope -y linux -Q opensuse robakk

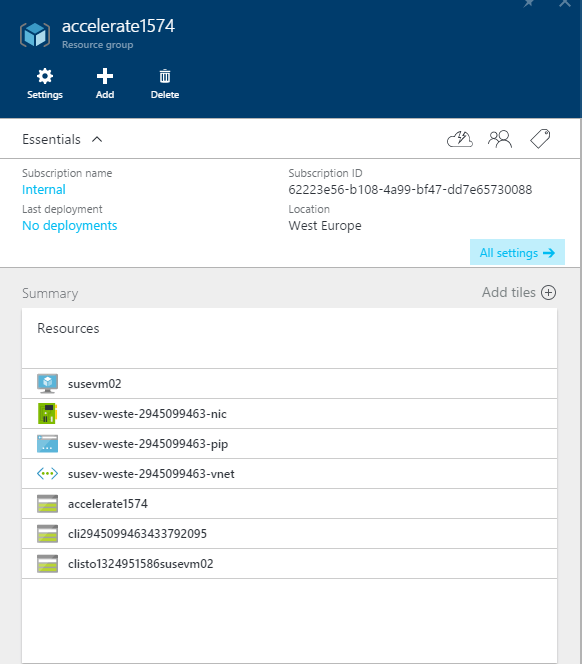
For more detailed help, type azure help vm quick-create

You should see output like:



Note that the storage account, virtual networks etc. are all automatically populated.

When you go to <https://portal.azure.com> and check the resource group you just created, you should see something similar like this image:



Installing and using GIT as a repository

In this next Module, we are going to install and configure GIT as a repository for ARM templates.

### **Install NODE.JS**

Download install the node.JS package from [https://nodejs.org](https://nodejs.org/).

When you install Node.js, you’ll want to ensure your PATH variable includes your install path so you can call Node from anywhere. Node comes with npm installed so you should have a version of npm.

### **Testing Your Install**

Create a new directory named hello-world, add a new app.js file with the following content:

/\* app.js \*/

console.log('Hello World!');

In the command prompt (or terminal on Mac) , run

$ node app.js

If you get any error regarding Node is not found, open a new command prompt (or terminal) to reflect the new environment variable for Node.

### **Installing and Setting up Git**

Chances are, you already have git running on your computer or you at least have heard of Git before. For this module, you will need to setup git locally and know few basics about Git.

**Getting Git for Mac**

There are several ways to install Git on a Mac. You can choose from one of the following ways:

1) If you want a more up to date version, you can also install it via a binary installer. An OSX Git installer is maintained and available for download at the Git website, at <http://git-scm.com/download/mac>.

2) If you already have Homebrew, you can install Git by executing:

$ brew install git

Make sure you update your $PATH environment variable to include the latest install path of Git. For example:

$ echo 'export PATH="/usr/local/bin:/usr/local/sbin:~/bin:$PATH"' >> ~/.bash\_profile

**Getting Git for Windows**

If you’re already using Chocolatey or Windows 10’s package manager to install software, you can simply run the following command from an elevated Powershell or Bash console (right click, select ‘Run as Administrator’):

$ cinst git.install

$ cinst poshgit

# Restart PowerShell / CMDer before moving on - or run

$ env:Path = [System.Environment]::GetEnvironmentVariable("Path","Machine") + ";" + [System.Environment]::GetEnvironmentVariable("Path","User")

$ cinst git-credential-winstore

$ cinst github

Verifying Git Installation

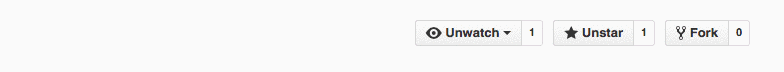
Now that you have Git installed, open up PowerShell on Windows or terminal on Mac. If everything worked correctly, you should be able to rungit --version.

Signing up for a Free GitHub account

Before we can get started, you need to register with GitHub for a free account. Either create or login into your account on [www.github.com](http://www.github.com).

Forking a Repository to your Account

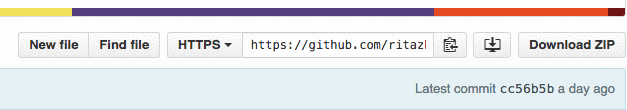
Now you are ready to do more with Git. Let’s start with a sample project. Head over to the Accelerate Lab repository on <https://github.com/robert-bakker/AccelerateLab> and click the little ‘fork’ button in the upper right.



This will create a copy of the repository as it exists in the original account into your own account.

Cloning the Repository to Your Machine

Visit your fork (**which should be at github.com/{your\_github\_username}/acceleratelab)** and copy the “HTTPS Clone URL”. Using this URL, you’re able to clone the repository, which downloads the whole repository, including its history and information about its origin locally. From PowerShell on Windows or terminal on Mac, change into the directory where you would like to clone your repo.



* Copy your repo URL

Clone the code to your local machine.

$ git clone https://github.com/{your\_github\_username}/acceleratelab

This should generate output that looks roughly like this:

$ git clone https://github.com/{your\_github\_username}/acceleratelab

Cloning into 'acceleratelab'...

...

...

Checking connectivity... done.

You can run explorer . from PowerShell on Windows or open . from Terminal on Mac to open up the folder in Explorer or Finder respectively. All the files are there - including the history of the whole repository. The connection to your fork ({your\_github\_username}/acceleratelab) is still there.

Creating a new Branch for your repo

In modern Git development, every single change that you want to make to the code base will be made in a “branch”. Like a tree branch, the branch is “based” on a different branch, and unlike other SCM systems, Git branches are very lightweight. The default branch name for GitHub repositories is master. In order to create a new branch, you can always run:

# This makes sure that your new branch is based on master

# When the default branch of a repo is "master", you should "git checkout master"

# The below command creates a new branch

$ git checkout -b my-new-branch

You can now go ahead and make your changes - adding files, writing code, fixing bugs. Keep in mind that a branch should host isolated changes. For example, you should create one branch that fixes a bug, then another branch for to develop a new feature you want to implement.

Staging your Changes for a Commit

Now that you made your changes, you can “stage” them for a commit. Whenever you stage a file for a commit, you make a snapshot of the file at the time you’re staging it for a commit. If you change a file after you staged it, you will have to stage it again. To stage a file, simply run:

$ git add ./path-to/my-file.md

If you just want to stage all files in your current repository (including deletions), run:

$ git add --all

Committing your Changes

Now that your changes have been staged, we’re ready to commit them. You can either pass the commit command a title for your commit - or omit the parameter, in which case Git will open up the default text editor for you to create a commit message.

To commit the quick way:

$ git commit -m "Add new feature: Git is Awesome"

To commit the long way, allowing you to define both title and message of your commit:

$ git commit

Pushing your new Branch to Your Fork on GitHub

Let’s say you have implemented a new feature, made some changes, committed the changes - now we have to make sure that your changes also end up on GitHub. To do so, we have to push your local branch to your fork on GitHub. Run the command below, using the name of the branch you want to push to

$ git checkout NAME\_OF\_YOUR\_NEW\_BRANCH

$ git push -u origin NAME\_OF\_YOUR\_NEW\_BRANCH

# Using ARM templates to further automate VM Deployments

## Use the Azure CLI for Mac, Linux, and Windows with Azure Resource Manager

### Azure resources

Use the Azure Resource Manager to create and manage a group of resources (user-managed entities such as a virtual machine, database server, database, or website) as a single logical unit, or resource group.

One advantage of the Azure Resource Manager is that you can create your Azure resources in a declarative way: you describe the structure and relationships of a deployable group of resources in JSON templates. The template identifies parameters that can be filled in either inline when running a command or stored in a separate JavaScript Object Notation (JSON) parameters file.

This allows you to easily create new resources using the same template by simply providing different parameters. For example, a template that creates a website will have parameters for the site name, the region the website will be located in, and other common settings.

When a template is used to modify or create a group, a deployment is created, which is then applied to the group.

After you create a deployment, you can manage the individual resources imperatively on the command line, just like you do in the classic deployment model.

For example, use CLI commands in Resource Manager mode to start, stop, or delete resources such as [Azure Resource Manager virtual machines](https://github.com/Azure/azure-content/blob/master/articles/virtual-machines/virtual-machines-linux-cli-deploy-templates.md). https://github.com/Azure/azure-content/blob/master/articles/virtual-machines/virtual-machines-linux-cli-deploy-templates.md

## Authentication

Working with the Azure Resource Manager through the Azure CLI currently requires you to authenticate to Microsoft Azure by using the azure login command and then specifying an account managed by Azure Active Directory - either a work or school account (an organizational account) or a Microsoft account.

## Set the Resource Manager mode

Because the CLI is not in Resource Manager mode by default, use the following command to enable Azure CLI Resource Manager commands.

azure config mode arm

## Create a resource group

A resource group is a logical grouping of resources such as network, storage, and compute resources. Almost all commands in the Resource Manager mode need a resource group.

First delete the resource group created in the previous exercises.

* Important: all resources you previously created will be lost.

View all resource groups in your subscription

azure group list

Then delete the specific resource group:

azure group delete [resource group name]

Create a new resource group using the last 4 digits of your mobile phone number as a unique identifier

azure group create -n accelerate[uniqueid] -l westeurope

* You will deploy to this accelerate[ID] resource group later when you use a template to launch a Linux VM. Once you have created a resource group, you can add resources like virtual machines and networks or storage.

## Use resource group templates

When working with templates, you can either [create your own](https://github.com/Azure/azure-content/blob/master/articles/resource-group-authoring-templates.md) (https://github.com/Azure/azure-content/blob/master/articles/resource-group-authoring-templates.md), or use one of the community-contributed [Quick Start templates](https://azure.microsoft.com/documentation/templates/),( https://azure.microsoft.com/documentation/templates/) which are also available on [GitHub](https://github.com/Azure/azure-quickstart-templates) (https://github.com/Azure/azure-quickstart-templates).

### Creating a multi node VMM scale set running an Apache/PHP website.

To create this more advanced scenario (VM Scale Set, loadbalancer, Apache & PHP), using an azure resource template from GitHub we will use the [201-vmss-lapstack-autoscale](https://github.com/Azure/azure-quickstart-templates/tree/master/201-vmss-lapstack-autoscale) template created by Madhan Arumugam Ramakrishnan, available on the Azure QuickStart templates repository on GitHub.

<https://github.com/Azure/azure-quickstart-templates/blob/master/201-vmss-lapstack-autoscale/>

This template creates a Simple self-contained Ubuntu/Apache/PHP (LAP stack) with auto scaling & load balancing. The VM Scale Set scales up when avg CPU across all VMs is larger than 60%, and it automatically scales down again when the average CPU is less than 50%. This template will deploy the following:

* Deploy the VM Scale Set with an instance count of 1
* After it is deployed look at the resource group public IP address resource (in portal or resources explorer). Get the IP or domain name.
* Browse to the website (port 80), which shows the current backend VM name.
* Hit the "Do work" button with an iteration count of say 300 (represents seconds of max CPU).
* After a few minutes the VM Scale Set capacity will increase, and refreshing the browser and going to the home page a few times will show additional backend VM name(s).

You can increase the work by connecting to more backend websites, or decrease by letting the iterations time-out, in which case the VM Scale Set will scale down - hence after about 10 minutes the capacity should be back down to 1.

### Deploy this template to your Azure resource group

You can use this template directly from [GitHub](https://github.com/Azure/azure-quickstart-templates), instead of downloading one to your computer. To do this, pass the URL to the azuredeploy.json file for the template in your command by using the **--**template**-uri** option. To get the URL, open azuredeploy.json on GitHub in raw mode, and copy the URL that appears in the browser's address bar. You can then use this URL directly to create a deployment by using a command similar to the following.

* (the new resource group needs to be created first using: azure group create -n accelerate[uniqueid] -l westeurope)

azure group deployment create [Resource Group Name] [Resource Group Deployment Name] --template-uri [RAW Github link to azuredeploy.json]

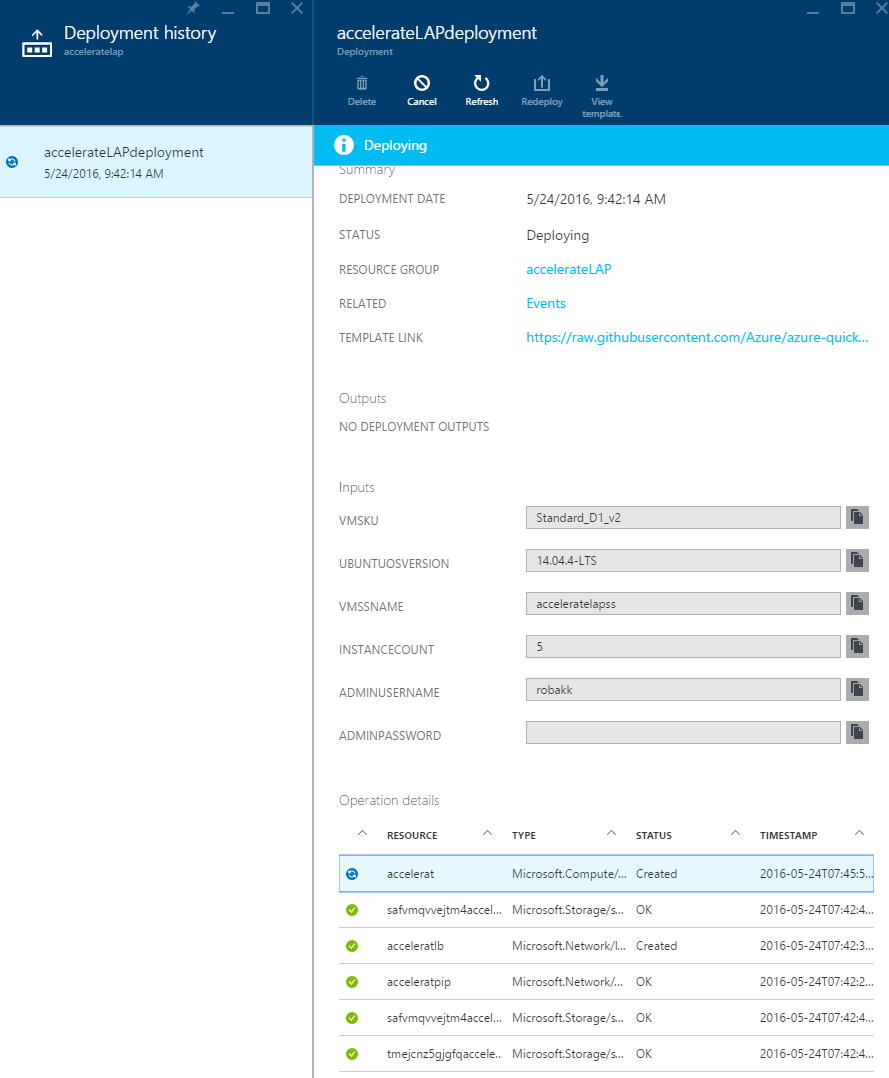
azure group deployment create accelerateLAP accelerateLAPdeployment https://raw.githubusercontent.com/robert-bakker/AccelerateLab/master/LabFiles/201-vmss-lapstack-autoscale/azuredeploy.json

You are then prompted to enter the necessary template parameters.

* vmssName : the name of the VM Scale Set
* instancecount : how many VMs need to be in the scale set
* adminusername : user name for the administrator
* Password: the admin password

This deployment will take approximately 5 minutes. You can watch the progress of the deployment on portal.azure.com in the resource group specified:

Now you should be able to use a web browser to connect to [scalesetname].westeurope.cloudapp.azure.com and enter a number of seconds to



## Export a resource template

For an existing resource group, you can view the Resource Manager template for the resource group. Exporting the template offers two benefits:

1. You can easily automate future deployments of the solution because all of the infrastructure is defined in the template.
2. You can become familiar with template syntax by looking at the JSON that represents your solution.

Using the Azure CLI, you can either export a template that represents the current state of your resource group, or download the template that was used for a particular deployment.

* **Export the template for a resource group** - This is helpful when you have made changes to a resource group, and need to retrieve the JSON representation of its current state. However, the generated template contains only a minimal number of parameters and no variables. Most of the values in the template are hard-coded. Before deploying the generated template, you may wish to convert more of the values into parameters so you can customize the deployment for different environments.

To export the template for a resource group to a local directory, run the azure group export command as shown in the following example. (Substitute a local directory appropriate for your operating system environment.)

azure group export accelerate[uniqueid] ~/azure/templates/

* **Download the template for a particular deployment** -- This is helpful when you need to view the actual template that was used to deploy resources. The template will include all of the parameters and variables defined for the original deployment. However, if someone in your organization has made changes to the resource group outside of what is defined in the template, this template will not represent the current state of the resource group.

To download the template used for a particular deployment to a local directory, run the azure group deployment template download command.

azure group deployment template download TestRG testRGDeploy ~/azure/templates/downloads/

Template export is in preview, and not all resource types currently support exporting a template. When attempting to export a template, you may see an error that states some resources were not exported. If needed, you can manually define these resources in your template after downloading it.

To view and edit these JSON templates, Visual Studio Code or Visual Studio are recommended. Install visual studio code for Windows, OSX or Linux from here : <https://code.visualstudio.com/>

# At home: Resources to continue learning

By participating in today’s event you will have exclusive access to download the presentation, step-by-step setup guidance, and latest online training and demos at [aka.ms/itinnovationresources](http://www.aka.ms/itinnovationresources).

|  |  |
| --- | --- |
| Resources | What you’ll learn |
| Event presentation. | Today’s presentation on “What’s new in Azure Infrastructure: Improving Datacenter Flexibility with Microsoft, Open Source and Other Technologies.” |
| Step-by-step guidance (attendee guide) | This document guides you through today’s instructor-led and individual activities, and provide you with the key resources and information necessary to continue building your skills to be able to implement in your environment. |
| Try Azure | Sign up for a free one-month trial of Azure. |
| Access Azure videos and online training | The most recent Enterprise Mobility, On Premises, In Cloud (Azure) and DevOps news and resources, from Microsoft and in the industry. |
| Blog: Regular IT Guy | Stay up to date on the latest in Azure. |

This document supports a preliminary release of a software product that may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright 2014 © Microsoft Corporation. All rights reserved.

Microsoft Active Directory, Azure Active Directory, Azure, Hyper-V, Windows, and Windows Server 2012 are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.