Azure Hand-Lab – Infrastructure as Code
Student Version

Date : 2017, October

Authors:
- Maxime Coquerel (Microsoft Azure MVP) http://zigmax.net
- Olivier Puech
- Tidjani Belmansour.

Requirements:
- An active Microsoft Azure subscription
    - You can create a free Azure Account (200$) at : https://azure.microsoft.com/en-us/free/

Acknowledges:
- Microsoft Quebec City (Host)
- And for all contributors to Quebec Azure Meetup.

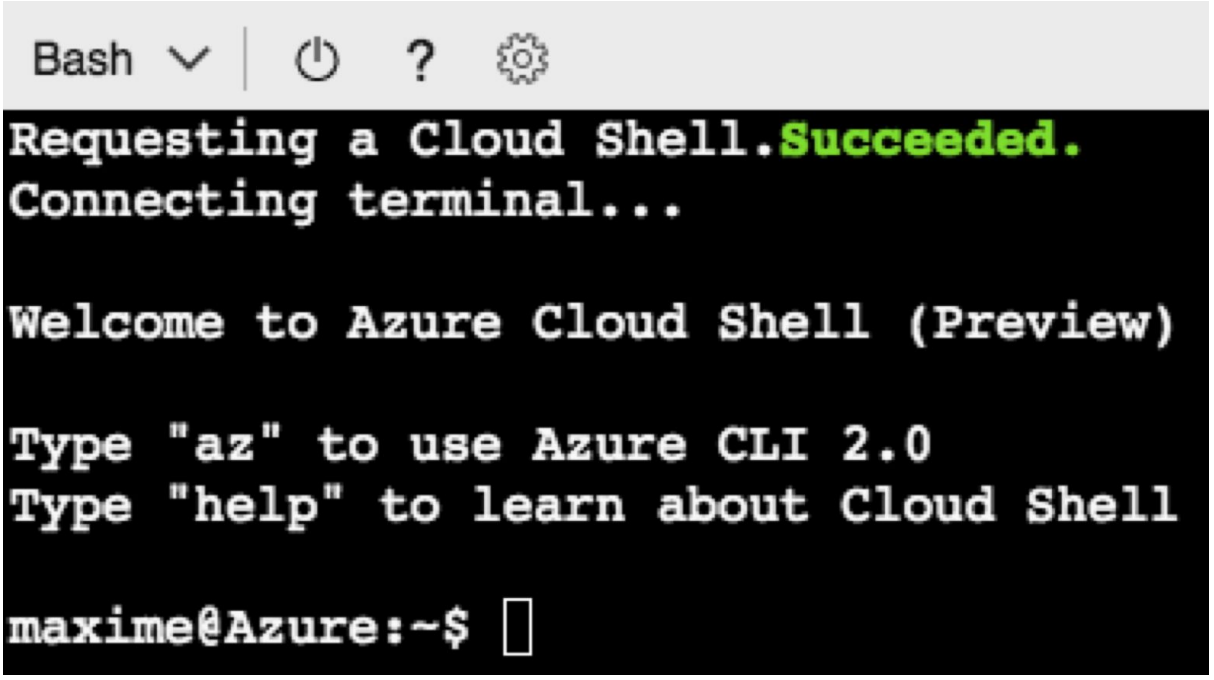
Lab 1: Create a VM with Azure CLI
Level: 100
Tested: Max | Tidjani (15 minutes) | Olivier (15 minutes)

In this lab, we create a single Linux (Ubuntu) virtual machine with Azure CLI.

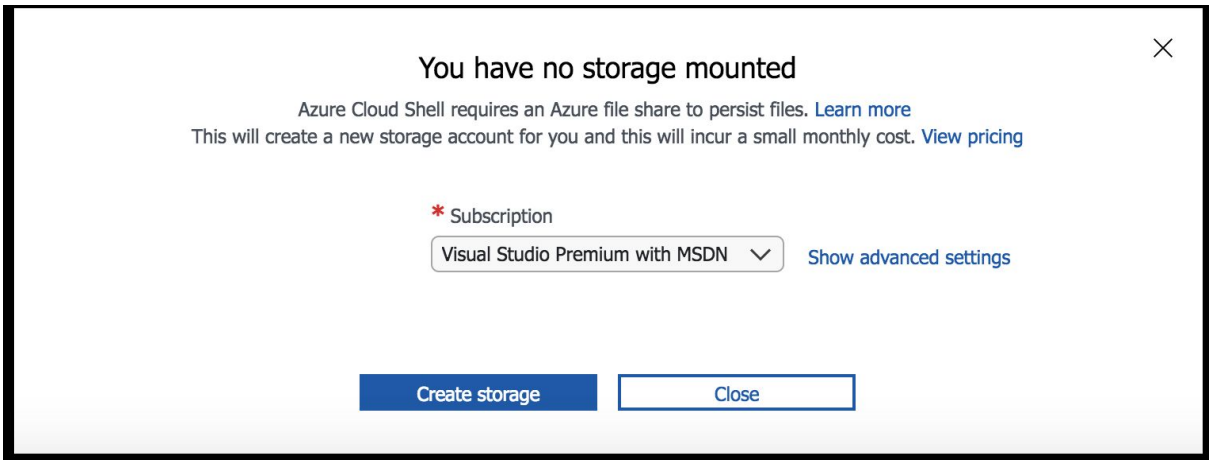
Requirements :

- Azure Subscription
- Azure Cloud Shell

Step 1: Launch Azure Cloud Shell (Bash)



If you're launching Cloud Shell for the first time, you'll be probably facing the following screen. If so, simply click "Create storage" and wait for few seconds:



Step 2: Create a resource groupe

```
az group create --name myResourceGroup --location eastus
```

```
maxime@Azure:~$ az group create --name myResourceGroup --location eastus
{
  "id": "/subscriptions/7(                                    /resourceGroups/myResourceGroup",
  "location": "eastus",
  "managedBy": null,
  "name": "myResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

Step 3 : Create virtual machine

```
az vm create --resource-group myResourceGroup --name myVM --image
UbuntuLTS --generate-ssh-keys
```

```
maxime@Azure:~$ az vm create --resource-group myResourceGroup --name myVM --image UbuntuLTS --generate-ssh-keys
SSH key files '/home/maxime/.ssh/id_rsa' and '/home/maxime/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM. If
back up your keys to a safe location.
{
  "fqdns": "",
  "id": "/subscriptions/7                    /resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/myVM",
  "location": "eastus",
  "macAddress": "00-0D-3A-1C-27-54",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "40.117.187.193",
  "resourceGroup": "myResourceGroup",
  "zones": ""
}
```

Step 4 : Open port 80 for web traffic
By default only SSH connections are allowed into Linux virtual machines deployed in Azure

```
az vm open-port --port 80 --resource-group myResourceGroup --name myVM
```

```
maxime@Azure:~$ az vm open-port --port 80 --resource-group myResourceGroup --name myVM
{
  "defaultSecurityRules": [
    {
      "access": "Allow",
      "description": "Allow inbound traffic from all VMs in VNET",
      "destinationAddressPrefix": "VirtualNetwork",
      "destinationAddressPrefixes": [],
      "destinationApplicationSecurityGroups": null,
      "destinationPortRange": "*",
      "destinationPortRanges": [],
      "direction": "Inbound",
      "etag": "W/\"70060b3f-6986-4574-a4d6-6769e0f02087\"",
      "id": "/subscriptions/7(                                    /resourceGroups/myResou
nBound",
      "name": "AllowVnetInBound",
      "priority": 65000,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "VirtualNetwork",
      "sourceAddressPrefixes": [],
      "sourceApplicationSecurityGroups": null,
      "sourcePortRange": "*",
      "sourcePortRanges": []
    },
```

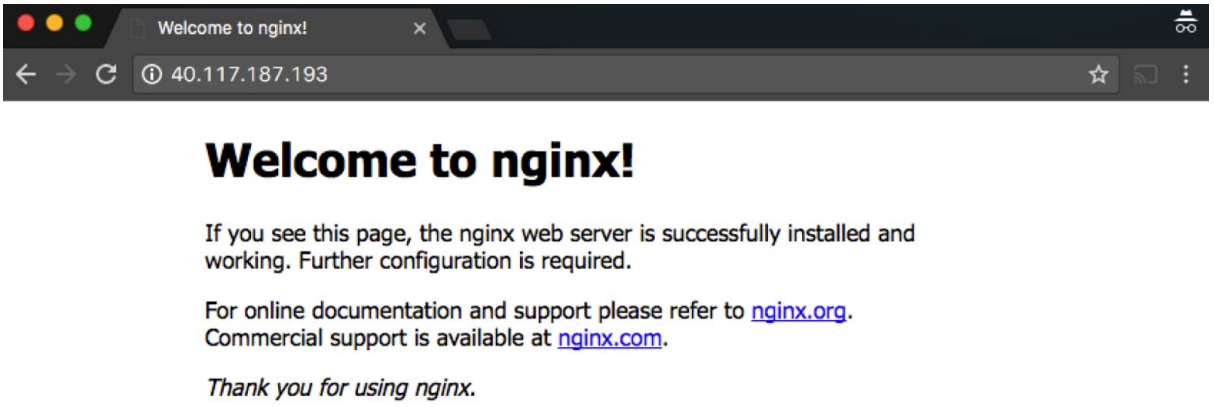
Step 5 : Install web server (nginx)

```
ssh <publicIpAddress>
```

```
maxime@Azure:~$ ssh 40.117.187.193
The authenticity of host '40.117.187.193 (40.117.187.193)' can't be established.
ECDSA key fingerprint is SHA256:OA2yz/6byYSvMQTWIooUoJQBFpmt48b+TDkErNIo7RE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.117.187.193' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.11.0-1013-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.
```

```
# update package source
sudo apt-get update

# install NGINX
sudo apt-get install nginx
```

To confirm that nginx is effectively up and running, open a new browser tab and navigate to the public IP address of your VM. You should see something like this:

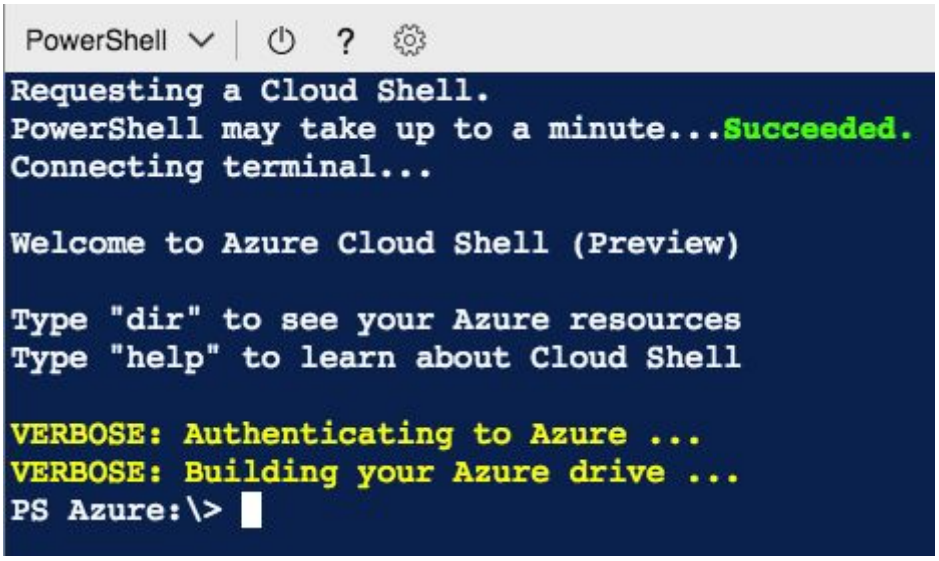


Lab 2 : Create a custom image in Azure
Level: 200
Tested: Max | Tidjani (45 minutes) | Olivier (45 minutes, galère avec VIM)

Requirements :

- Azure Subscription
- Azure Cloud Shell
- Basic knowledge of PowerShell

Step 1 : Azure Cloud Shell (PowerShell)



Step 2 : Stop the machine

```
# stop VM
Stop-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM -Force
```

```
PS Azure:\>
>> Stop-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM -Force


OperationId :
Status      : Succeeded
StartTime   : 10/17/2017 12:26:19 AM
EndTime     : 10/17/2017 12:28:01 AM
Error       :
```

Optional if it's Linux Machine:

```
# Remove all your personal account information
Set-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM -Generalized
```

```
PS Azure:\>
>> Set-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM -Generalized


OperationId :
Status      :
StartTime   :
EndTime     :
Error       :
```

Generalization removes all your personal account information, among other things, and prepares the machine to be used as an image.

Step 3 : Get a reference to the virtual image

```
# Get a reference to the virtual image
$vm = Get-AzureRmVM -ResourceGroupName myResourceGroup -Name myVM
```

```
PS Azure:\>
>> $vm = Get-AzureRmVM -Name myVM -ResourceGroupName myResourceGroup
PS Azure:\>
```

Step 4 : Create the image configuration
This command creates a configurable image object.

```
# Create the image configuration
$image = New-AzureRmImageConfig -Location EastUS -SourceVirtualMachineId
$vm.ID
```

```
PS Azure:\>
>> $image = New-AzureRmImageConfig -Location EastUS -SourceVirtualMachineId $vm.ID
PS Azure:\>
```

Step 5 : Create the image

```
# Create the image
New-AzureRmImage -Image $image -ImageName myImage -ResourceGroupName
myResourceGroup
```

```
PS Azure:\>
>> New-AzureRmImage -Image $image -ImageName myImage -ResourceGroupName myResourceGroup


ResourceGroupName     : myResourceGroup
SourceVirtualMachine  : Microsoft.Azure.Management.Compute.Models.SubResource
StorageProfile        : Microsoft.Azure.Management.Compute.Models.ImageStorageProfile
ProvisioningState     : Succeeded
Id                    : /subscriptions/7[redacted]/resourceGroups/
Name                  : myImage
Type                  : Microsoft.Compute/images
Location              : eastus
Tags                  : {}
```

Step 6 : Create VM from the image

```
cd C:\Users\ContainerAdministrator\CloudDrive
dir
git clone https://github.com/zigmax/azureqc17-iac-lab2/

vim createvmfromimg.ps1 (optional)
```

```
PS Azure:\Microsoft Azure Sponsorship> cd C:\Users\ContainerAdministrator\CloudDrive\
PS C:\Users\ContainerAdministrator\CloudDrive> dir


    Directory: C:\Users\ContainerAdministrator\CloudDrive


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/15/2017   9:10 PM                .cloudconsole
d-----        10/15/2017   9:08 PM                .pscloudshell


PS C:\Users\ContainerAdministrator\CloudDrive>
```

```
PS C:\Users\ContainerAdministrator\CloudDrive> vim createvmfromimg.ps1
PS C:\Users\ContainerAdministrator\CloudDrive>
```

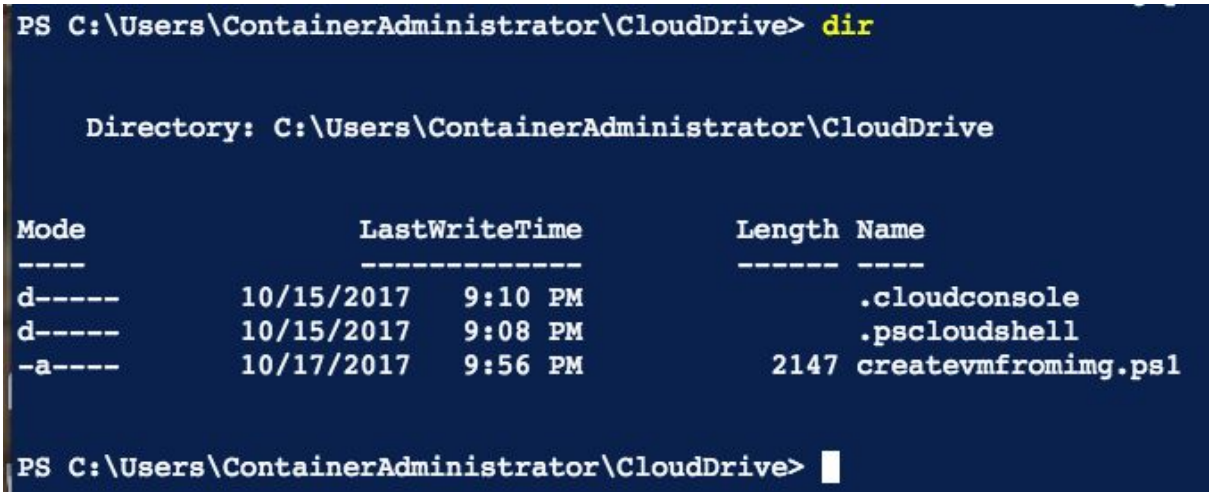
```
$cred = Get-Credential -Message "Enter a username and password
for the virtual machine."

New-AzureRmResourceGroup -Name myResourceGroupFromImage -Location
EastUS

$subnetConfig = New-AzureRmVirtualNetworkSubnetConfig `
   -Name mySubnet `
   -AddressPrefix 192.168.1.0/24

$vnet = New-AzureRmVirtualNetwork `
   -ResourceGroupName myResourceGroupFromImage `
   -Location EastUS `
   -Name MYvNET `
   -AddressPrefix 192.168.0.0/16 `
   -Subnet $subnetConfig

$pip = New-AzureRmPublicIpAddress `
   -ResourceGroupName myResourceGroupFromImage `
   -Location EastUS `
   -Name "mypublicdns$(Get-Random)" `
   -AllocationMethod Static `
   -IdleTimeoutInMinutes 4

 $nsgRuleWeb = New-AzureRmNetworkSecurityRuleConfig `
   -Name myNetworkSecurityGroupRuleWeb `
   -Protocol Tcp `
   -Direction Inbound `
   -Priority 1000 `
   -SourceAddressPrefix * `
   -SourcePortRange * `
   -DestinationAddressPrefix * `
```

```
   -DestinationPortRange 80 `
   -Access Allow

 $nsg = New-AzureRmNetworkSecurityGroup `
   -ResourceGroupName myResourceGroupFromImage `
   -Location EastUS `
   -Name myNetworkSecurityGroup `
   -SecurityRules $nsgRuleWeb

$nic = New-AzureRmNetworkInterface `
   -Name myNic `
   -ResourceGroupName myResourceGroupFromImage `
   -Location EastUS `
   -SubnetId $vnet.Subnets[0].Id `
   -PublicIpAddressId $pip.Id `
   -NetworkSecurityGroupId $nsg.Id

$vmConfig = New-AzureRmVMConfig `
   -VMName myVMfromImage `
   -VMSize Standard_D1 | Set-AzureRmVMOperatingSystem -Linux `
       -ComputerName myComputer `
       -Credential $cred

# Here is where we create a variable to store information about
the image
$image = Get-AzureRmImage `
   -ImageName myImage `
   -ResourceGroupName myResourceGroup

# Here is where we specify that we want to create the VM from and
image and provide the image ID
$vmConfig = Set-AzureRmVMSourceImage -VM $vmConfig -Id $image.Id

$vmConfig = Add-AzureRmVMNetworkInterface -VM $vmConfig -Id
$nic.Id
```

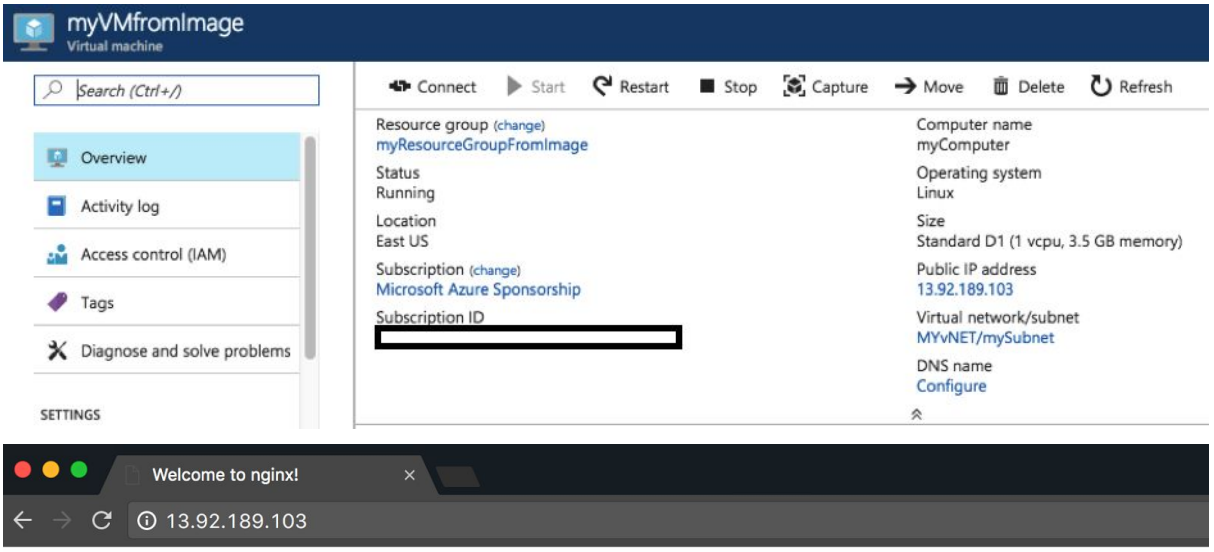
```
New-AzureRmVM `
   -ResourceGroupName myResourceGroupFromImage `
   -Location EastUS `
   -VM $vmConfig
```

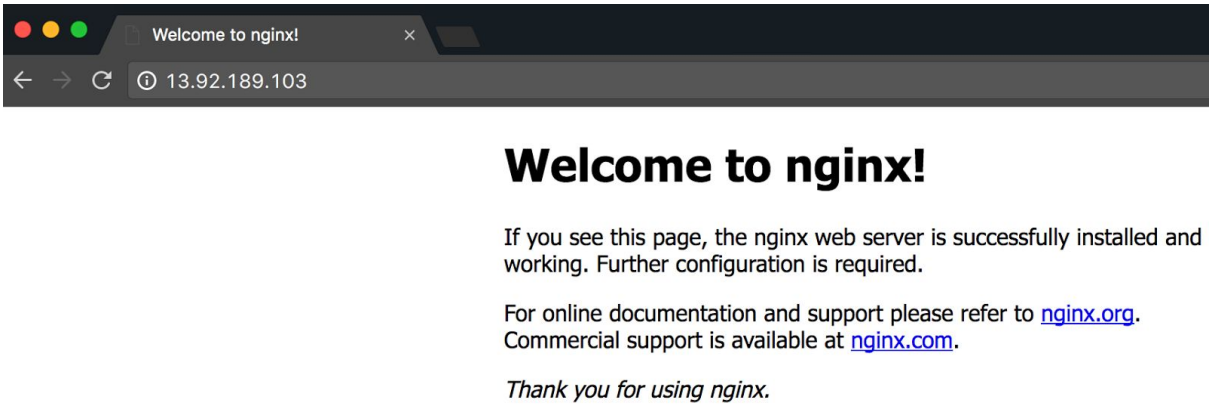
[Press ESC]:wq! pour sauvegarder puis sortir
pour re-modifier en cas d'erreur :

```
vim createvmfromimg.ps1
:edit!
[modif]
:wq!
```



```
ResourceGroupName : myResourceGroupFromImage
Location          : eastus
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/7d[redacted]/resourceGroup

WARNING: The output object type of this cmdlet will be modified in a future release.
WARNING: The output object type of this cmdlet will be modified in a future release.
WARNING: The output object type of this cmdlet will be modified in a future release.
WARNING: The output object type of this cmdlet will be modified in a future release.
WARNING: Since the VM is created using premium storage, existing standard storage acc

RequestId           :
IsSuccessStatusCode : True
StatusCode          : OK
ReasonPhrase        : OK
```





Step 7 : List all images by name

```
# Display image name
$images = Find-AzureRMResource -ResourceType Microsoft.Compute/images
$images.name
```

```
PS C:\Users>
>> $images = Find-AzureRMResource -ResourceType Microsoft.Compute/images
PS C:\Users>
>> $images.name
myImage
PS C:\Users>
```

Step 8 : Delete an image

```
Remove-AzureRmImage `
   -ImageName myImage `
   -ResourceGroupName myResourceGroup
```

```
PS Azure:\> Remove-AzureRmImage `
>> -ImageName myImage `
>> -ResourceGroupName myResourceGroup

Remove-AzureRmImage operation
This cmdlet will remove the specified resource.  Do you want to continue?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y


Name      : af16740a-d16a-4285-aec1-da22a76915fb
Status    : Succeeded
StartTime : 10/17/2017 10:47:41 PM
EndTime   : 10/17/2017 10:47:46 PM



PS Azure:\>
```

Lab 3: Create a VM with Azure ARM Template

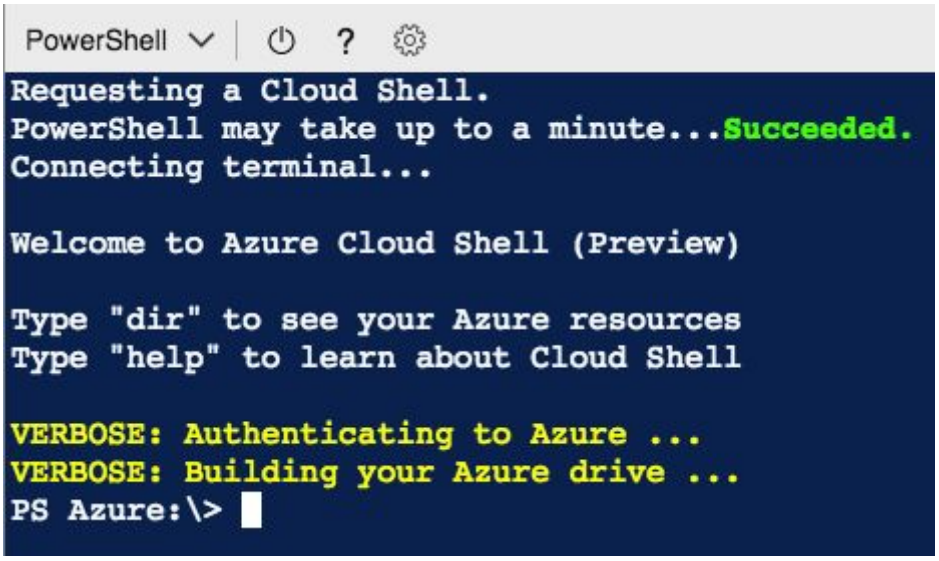
Level: 200
Tested: Max | Olivier (30 minutes, galère avec conflit demomax) | Tidjani (20 minutes, merci à Olivier pour l'astuce relative au conflit demomax)

Requirements :

- Azure Subscription
- Azure Cloud Shell
- Github: https://github.com/zigmax/azureqc17-iac-lab3

Goal of this Lab :
Deploy Windows Server 2016 Datacenter Machine based on Azure ARM Template.

Step 1 : Azure Cloud Shell (PowerShell)



Step 2 : Get your SubscriptionId

```
PS Azure:\> dir


    Directory: Azure:


Mode SubscriptionName                 SubscriptionId
---- ----------------                 --------------
+    Microsoft Azure Sponsorship      7db5e03c-f3c2-
+    Visual Studio Ultimate avec MSDN d310c4a5-81d3-
```

Step 3:

```
PS Azure:\> cd C:\Users\ContainerAdministrator\CloudDrive\
PS C:\Users\ContainerAdministrator\CloudDrive>
```

```
cd C:\Users\ContainerAdministrator\CloudDrive\
```

Step 4:

```
PS C:\Users\ContainerAdministrator\CloudDrive> git clone https://github.com/zigmax/azureqc17-iac-lab3/
Cloning into 'azureqc17-iac-lab3'...
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.
PS C:\Users\ContainerAdministrator\CloudDrive>
```

```
git clone https://github.com/zigmax/azureqc17-iac-lab3
```

Step 5:

```
PS C:\Users\ContainerAdministrator\CloudDrive> cd .\azureqc17-iac-lab3\
PS C:\Users\ContainerAdministrator\CloudDrive\azureqc17-iac-lab3> ls


    Directory: C:\Users\ContainerAdministrator\CloudDrive\azureqc17-iac-lab3


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/18/2017  12:30 AM           7002 azuredeploy.json
-a----        10/18/2017  12:30 AM            420 azuredeploy.parameters.json
-a----        10/18/2017  12:30 AM            171 README.md
-a----        10/18/2017  12:30 AM            657 VM.ps1
```

Step 6: Explore azuredeploy.json, azuredeploy.parameters.json and VM.ps1 files

https://github.com/zigmax/azureqc17-iac-lab3/blob/master/azuredeploy.json
https://github.com/zigmax/azureqc17-iac-lab3/blob/master/azuredeploy.parameters.json
https://github.com/zigmax/azureqc17-iac-lab3/blob/master/VM.ps1

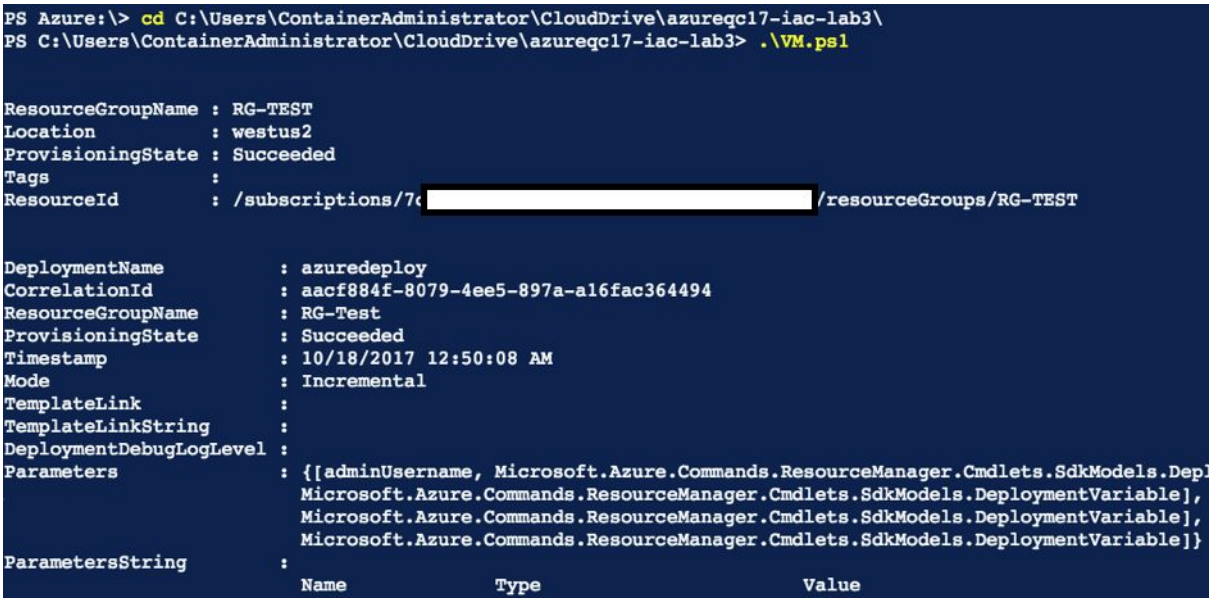
Or:

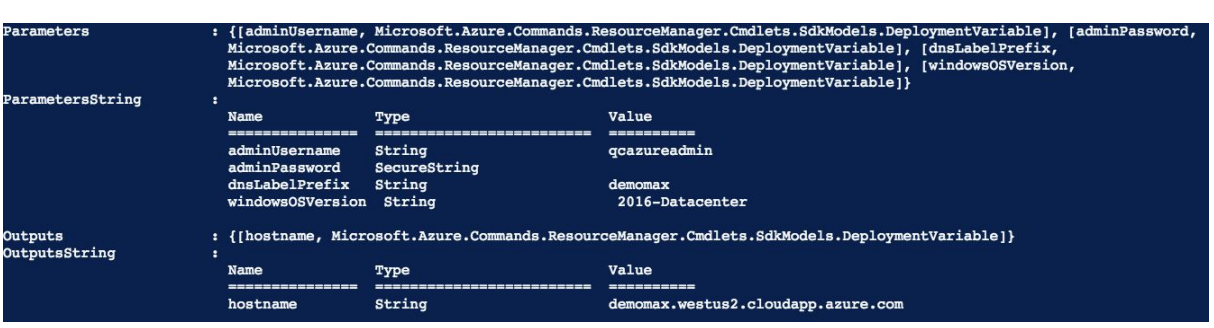
```
type VM.ps1
type azuredeploy.json
type azuredeploy.parameters.json
```

!!Warning!! There's a potential conflict with the "dnsLabelPrefix" value. You need to set your own before running the VM.ps1 command. Here's how:

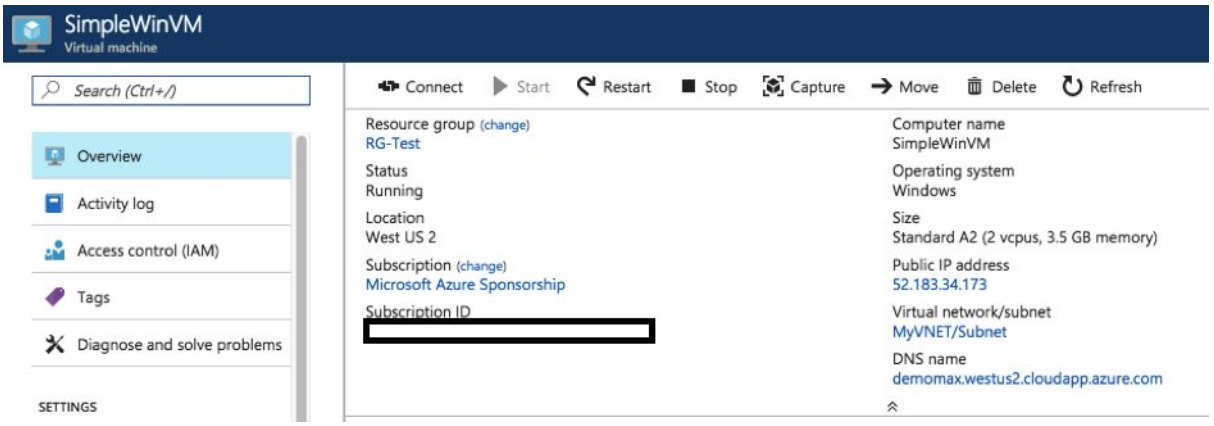
```
vim azuredeploy.parameters.json
:edit!
[change the value of dnsLabelPrefix so it is unique (demomax is already in use)]
[ESC]:wq!
```

Step 9: Run VM.ps1





You can confirm the creation of the VM either from the portal:



Or directly from Cloud Shell (we see it on the 2nd line):

```
PS C:\Users\ContainerAdministrator\CloudDrive\azureqc17-iac-lab3> Get-AzureRmVM

ResourceGroupName        Name   Location          VmSize  OsType              NIC ProvisioningState Zone
-----------------        ----   --------          ------  ------              --- ----------------- ----
TIDJANIDEMORG     tidjLinuxVM CanadaEast Standard_DS1_v2   Linux tidjLinuxVMVMNic         Succeeded
RG-TEST           SimpleWinVM    westus2     Standard_A2 Windows          myVMNic         Succeeded
```

Lab 4: Create an Azure VM with HashiCorp Terraform
Level : 300
Tested : Max | Olivier (20 minutes, chemin rapide) | Tidjani (20 minutes)

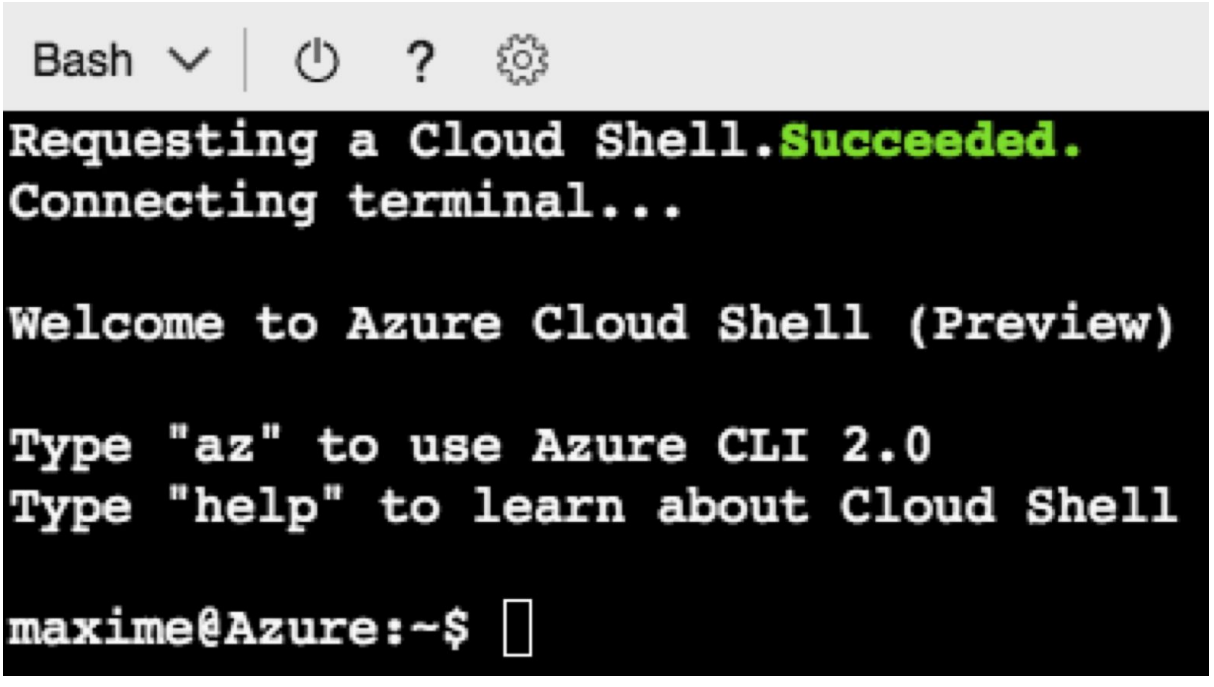
Requirements :

- Azure Subscription
- Azure Cloud Shell

Step 0: Need to read before started

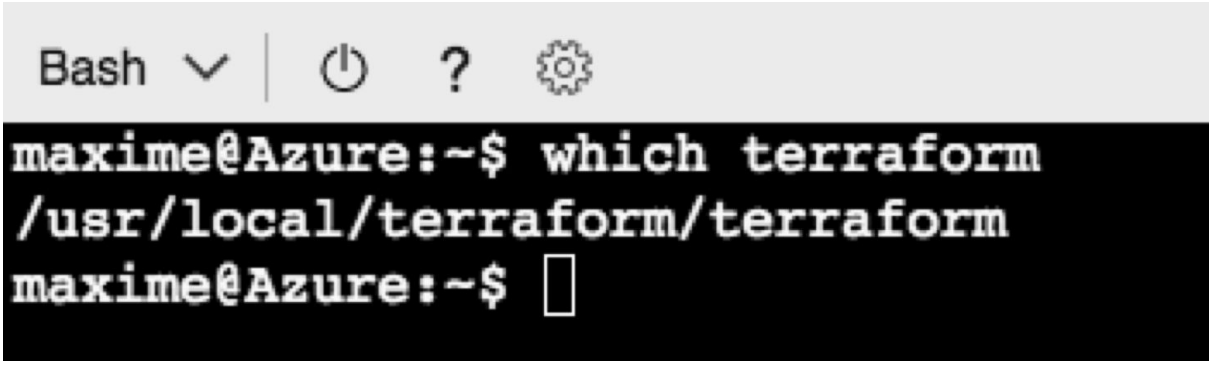
- Introduction to Terraform : https://www.terraform.io/intro/index.html
- Introduction à Terraform (FR) avec Azure : http://zigmax.net/azure-avec-terraform/

Step 1: Launch Azure Cloud Shell (Bash)



Step 2: Where is Terraform ?

```
which terraform
```



Step 3 : Create a Terraform module : main.tf

Accelerated method:

```
git clone https://github.com/zigmax/azureqc17-iac-lab4/

Jump to step 8
```

Standard method:

Tip : In production, we recommend to use lot of small modules.

```
vi main.tf
```

Step 4 : Create a resource group (need be included in main.tf)

```
resource "azurerm_resource_group" "test" {
  name     = "acctestrg"
  location = "West US 2"
}
```

Step 5 : Create a virtual network with a public IP (need be included in main.tf)

```
resource "azurerm_virtual_network" "test" {
  name                = "acctvn"
  address_space       = ["10.0.0.0/16"]
  location            = "${azurerm_resource_group.test.location}"
  resource_group_name = "${azurerm_resource_group.test.name}"
}
```

```
resource "azurerm_subnet" "test" {
  name                 = "acctsub"
  resource_group_name  = "${azurerm_resource_group.test.name}"
  virtual_network_name = "${azurerm_virtual_network.test.name}"
  address_prefix       = "10.0.2.0/24"
}
```

```
resource "azurerm_public_ip" "test" {
  name                         = "pubip"
  location                     = "${azurerm_resource_group.test.location}"
  resource_group_name          = "${azurerm_resource_group.test.name}"
  public_ip_address_allocation = "Dynamic"
  idle_timeout_in_minutes      = 30

  tags {
    environment = "test"
  }
}
```

```
resource "azurerm_network_interface" "test" {
  name                = "acctni"
  location            = "West US 2"
  resource_group_name = "${azurerm_resource_group.test.name}"

  ip_configuration {
    name                          = "testconfiguration1"
    subnet_id                     = "${azurerm_subnet.test.id}"
    private_ip_address_allocation = "static"
    private_ip_address = "10.0.2.5"
    public_ip_address_id = "${azurerm_public_ip.test.id}"
  }
}
```

Step 6 : Create Managed Disk

```
resource "azurerm_managed_disk" "test" {
  name                 = "datadisk_existing"
  location             = "West US 2"
```

```
  resource_group_name  = "${azurerm_resource_group.test.name}"
  storage_account_type = "Standard_LRS"
  create_option        = "Empty"
  disk_size_gb         = "1023"
}
```

Step 7 : Create Ubuntu VM

```
resource "azurerm_virtual_machine" "test" {
  name                = "acctvm"
  location            = "West US 2"
  resource_group_name   = "${azurerm_resource_group.test.name}"
  network_interface_ids = ["${azurerm_network_interface.test.id}"]
  vm_size             = "Standard_DS1_v2"

storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }

storage_os_disk {
    name              = "myosdisk1"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
# Optional data disks
  storage_data_disk {
    name              = "datadisk_new"
    managed_disk_type = "Standard_LRS"
    create_option     = "Empty"
    lun               = 0
    disk_size_gb      = "1023"
  }

  storage_data_disk {
    name          = "${azurerm_managed_disk.test.name}"
    managed_disk_id = "${azurerm_managed_disk.test.id}"
```

```
  create_option   = "Attach"
  lun             = 1
  disk_size_gb    = "${azurerm_managed_disk.test.disk_size_gb}"
 }

 os_profile {
  computer_name  = "hostname"
  admin_username = "qcazureadmin"
  admin_password = "QuebecMeetupAzure!"
 }

 os_profile_linux_config {
  disable_password_authentication = false
 }

 tags {
  environment = "demomeetupazure"
 }
}
data "azurerm_public_ip" "test" {
 name                = "${azurerm_public_ip.test.name}"
 resource_group_name = "${azurerm_resource_group.test.name}"
 depends_on          = ["azurerm_virtual_machine.test"]
}

output "ip_address" {
 value = "${data.azurerm_public_ip.test.ip_address}"
}
```

Step 8: Terraform “init”

```
Bash
maxime@Azure:~$ terraform init

Initializing provider plugins...
- Checking for available provider plugins on https://releases.hashicorp.com...
- Downloading plugin for provider "azurerm" (0.2.2)...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.azurerm: version = "~> 0.2"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
maxime@Azure:~$
```

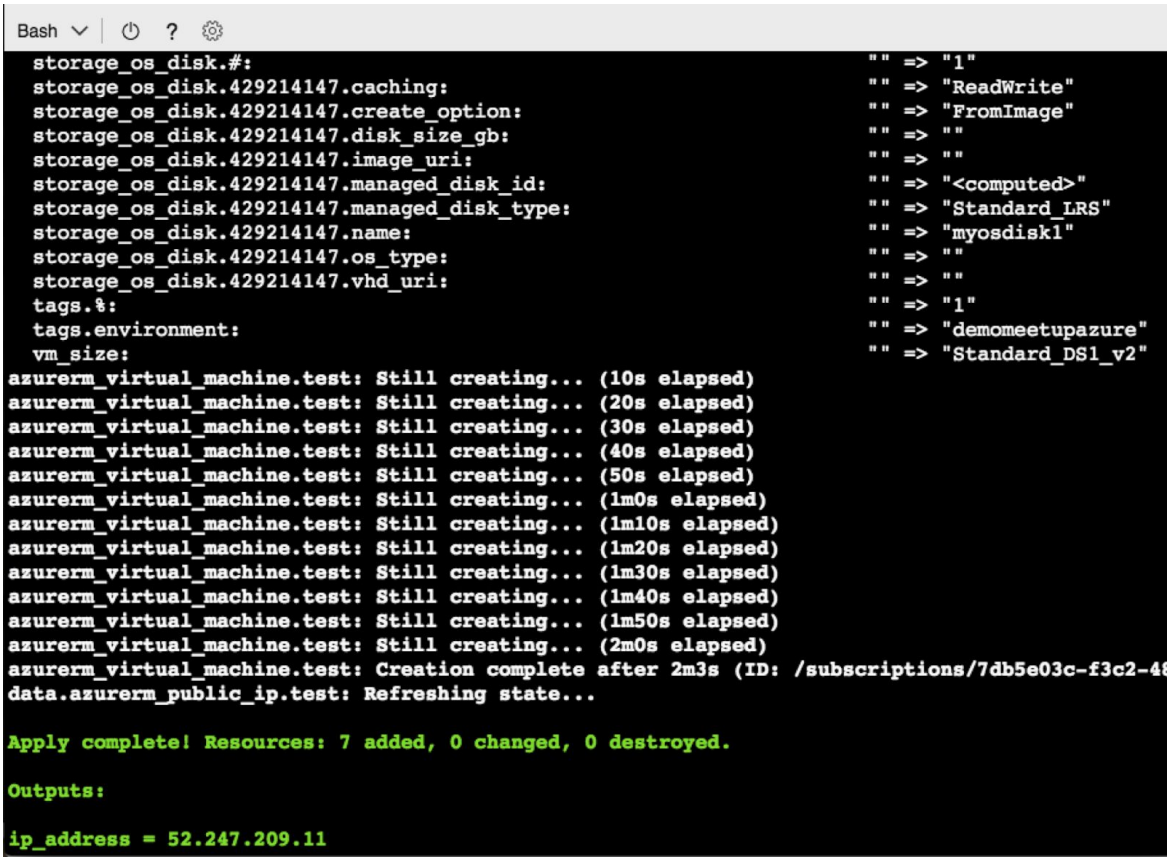
Step 9 : Terraform plan

```
Bash
maxime@Azure:~$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


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

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

 <= data.azurerm_public_ip.test
      id:                                                   <computed>
      domain_name_label:                                    <computed>
      fqdn:                                                 <computed>
      idle_timeout_in_minutes:                              <computed>
      ip_address:                                           <computed>
      name:                                                 "pubip"
      resource_group_name:                                  "acctestrg"
      tags.%:                                               <computed>

  + azurerm_managed_disk.test
      id:                                                   <computed>
      create_option:                                        "Empty"
      disk_size_gb:                                         "1023"
      location:                                             "westus2"
      name:                                                 "datadisk_existing"
      resource_group_name:                                  "acctestrg"
      source_uri:                                           <computed>
```

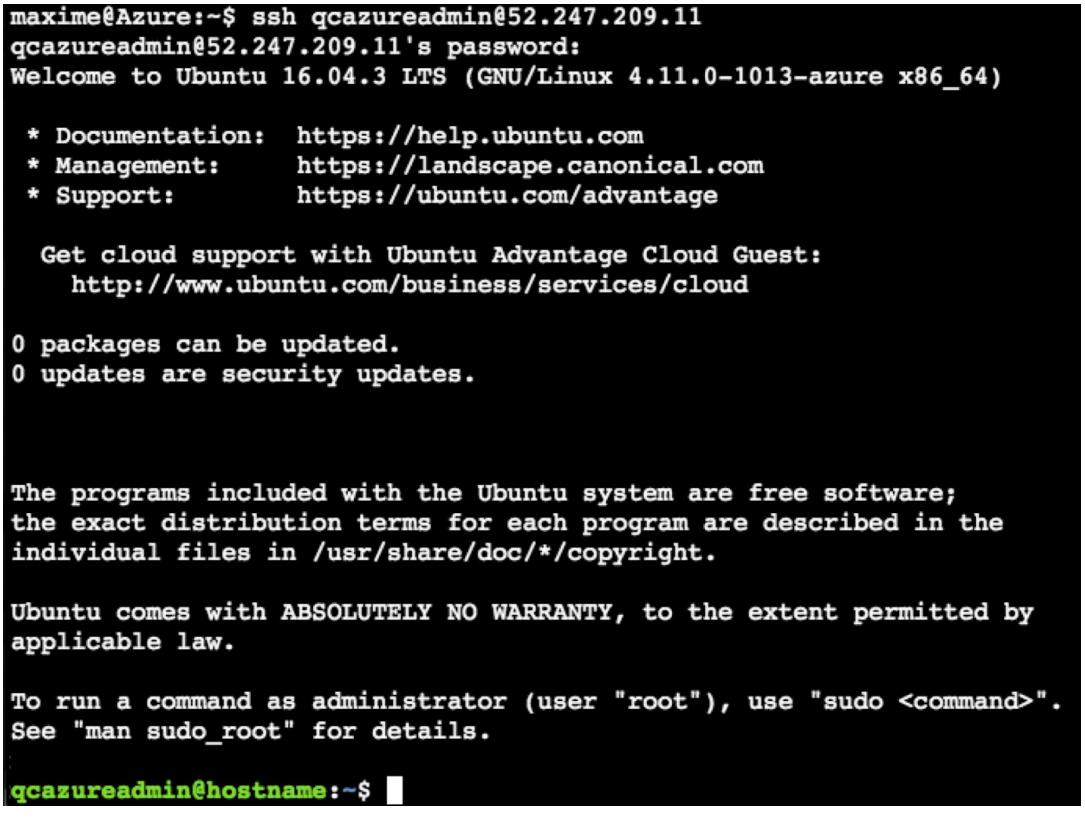
```
Bash
      storage_os_disk.#:                                              "1"
      storage_os_disk.429214147.caching:                              "ReadWrite"
      storage_os_disk.429214147.create_option:                        "FromImage"
      storage_os_disk.429214147.disk_size_gb:                         ""
      storage_os_disk.429214147.image_uri:                            ""
      storage_os_disk.429214147.managed_disk_id:                      <computed>
      storage_os_disk.429214147.managed_disk_type:                    "Standard_LRS"
      storage_os_disk.429214147.name:                                 "myosdisk1"
      storage_os_disk.429214147.os_type:                              ""
      storage_os_disk.429214147.vhd_uri:                              ""
      tags.%:                                                         "1"
      tags.environment:                                               "demomeetupazure"
      vm_size:                                                        "Standard_DS1_v2"

  + azurerm_virtual_network.test
      id:                                                             <computed>
      address_space.#:                                                "1"
      address_space.0:                                                "10.0.0.0/16"
      location:                                                       "westus2"
      name:                                                           "acctvn"
      resource_group_name:                                            "acctestrg"
      subnet.#:                                                       <computed>
      tags.%:                                                         <computed>


Plan: 7 to add, 0 to change, 0 to destroy.

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

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

Step 10 : Terrafom apply

```
Bash
maxime@Azure:~$ terraform apply
azurerm_resource_group.test: Creating...
  location: "" => "westus2"
  name:     "" => "acctestrg"
  tags.%:   "" => "<computed>"
azurerm_resource_group.test: Creation complete after 1s (ID: /subscriptions/7db5e03c-f3c
azurerm_public_ip.test: Creating...
  fqdn:                         "" => "<computed>"
  idle_timeout_in_minutes:      "" => "30"
  ip_address:                   "" => "<computed>"
  location:                     "" => "westus2"
  name:                         "" => "pubip"
  public_ip_address_allocation: "" => "dynamic"
  resource_group_name:          "" => "acctestrg"
  tags.%:                       "" => "1"
  tags.environment:             "" => "test"
azurerm_managed_disk.test: Creating...
  create_option:        "" => "Empty"
  disk_size_gb:         "" => "1023"
  location:             "" => "westus2"
  name:                 "" => "datadisk_existing"
  resource_group_name:  "" => "acctestrg"
  source_uri:           "" => "<computed>"
  storage_account_type: "" => "Standard_LRS"
  tags.%:               "" => "<computed>"
azurerm_virtual_network.test: Creating...
  address_space.#:     "" => "1"
  address_space.0:     "" => "10.0.0.0/16"
  location:            "" => "westus2"
  name:                "" => "acctvn"
  resource_group_name: "" => "acctestrg"
  subnet.#:            "" => "<computed>"
```



Step 11 : SSH with Public IP



Ressources:

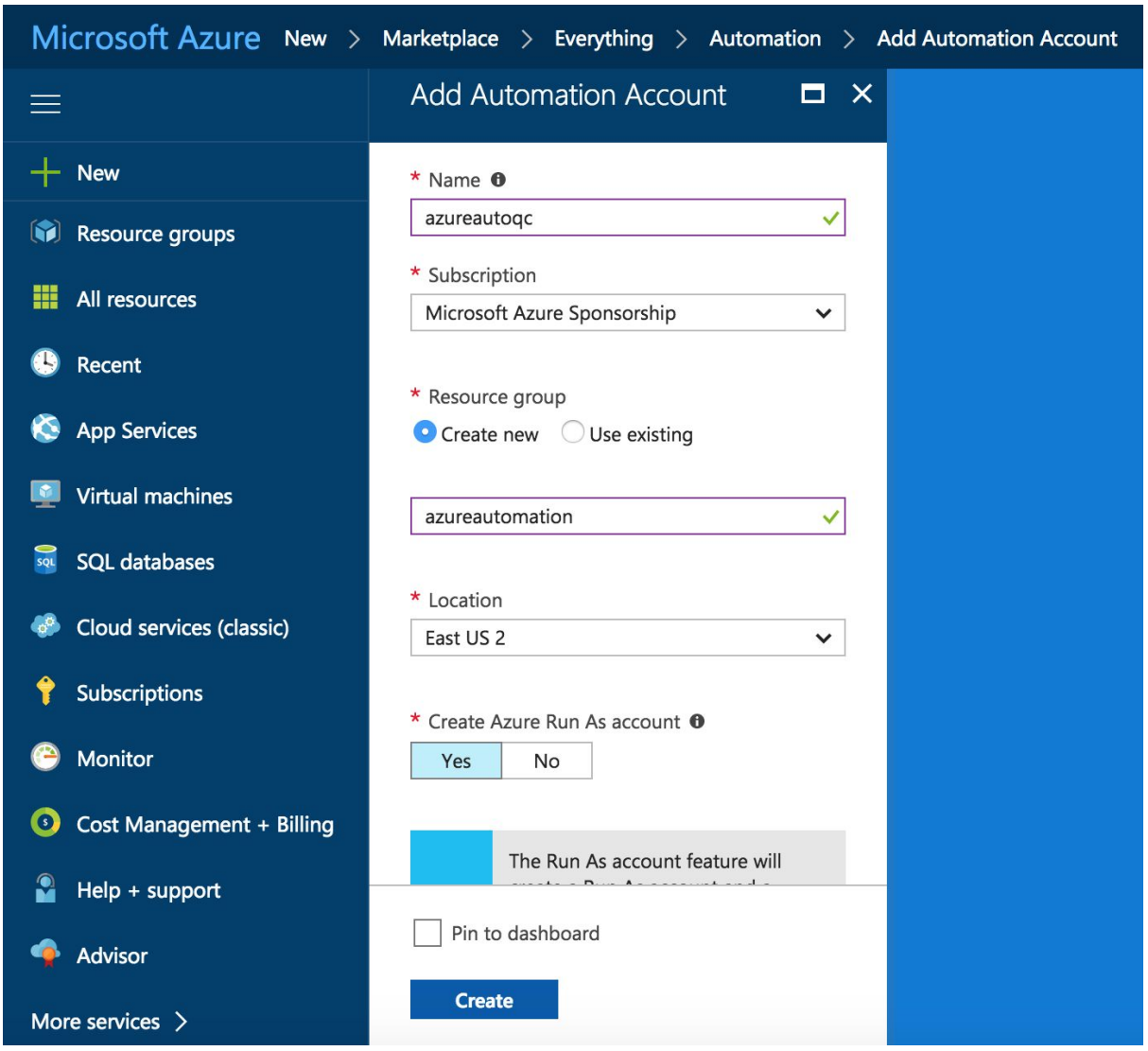
- Automatiser votre infrastructure Azure avec Terraform : http://zigmax.net/azure-avec-terraform/
- Terraform Azure ARM documentation : https://www.terraform.io/docs/providers/azurerm/r/virtual_machine.html
- HashiCorp Terraform : https://www.terraform.io/

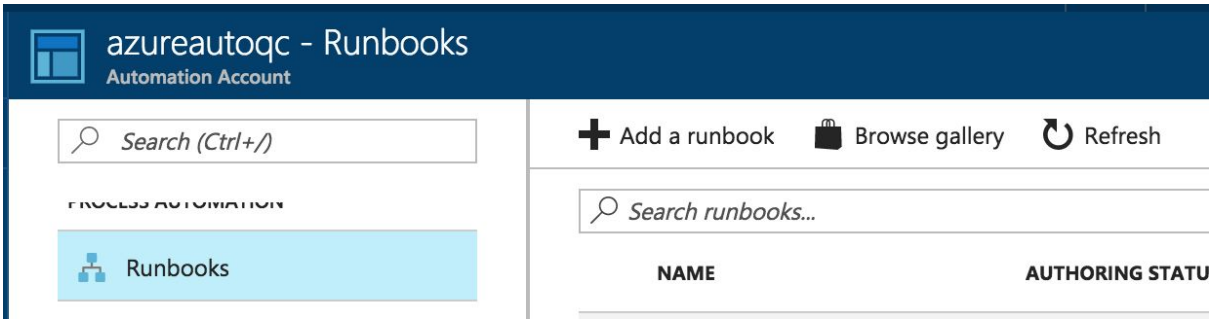
Lab 5: Azure Automation
Level: 100
Tested: Max | Olivier (30 minutes) | Tidjani (15 minutes)

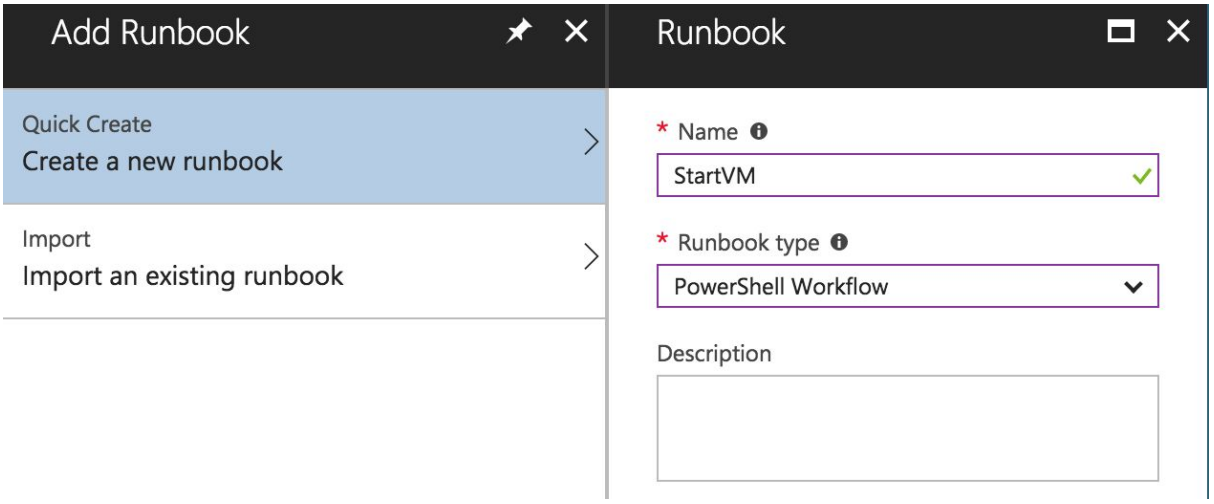
Requirements :

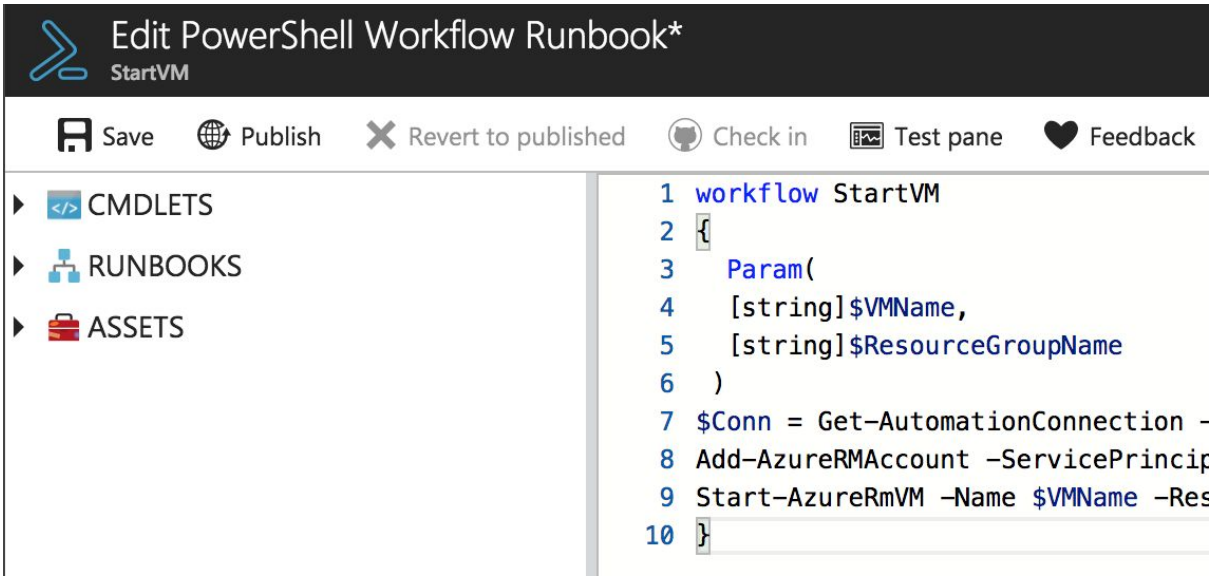
- Azure Subscription
- Basic knowledge of PowerShell

Requirement: Read this :
https://docs.microsoft.com/en-us/azure/automation/automation-intro



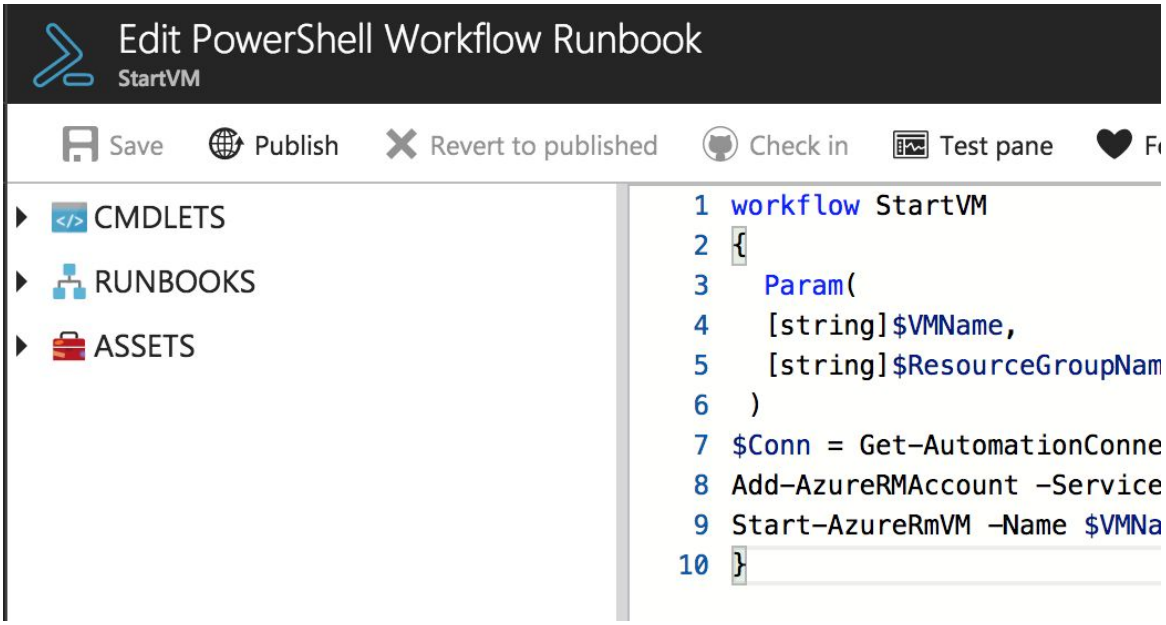




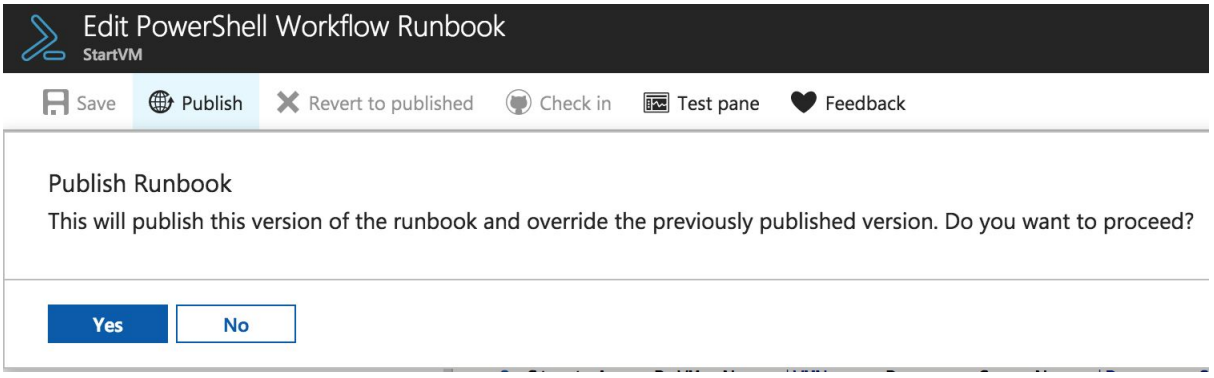


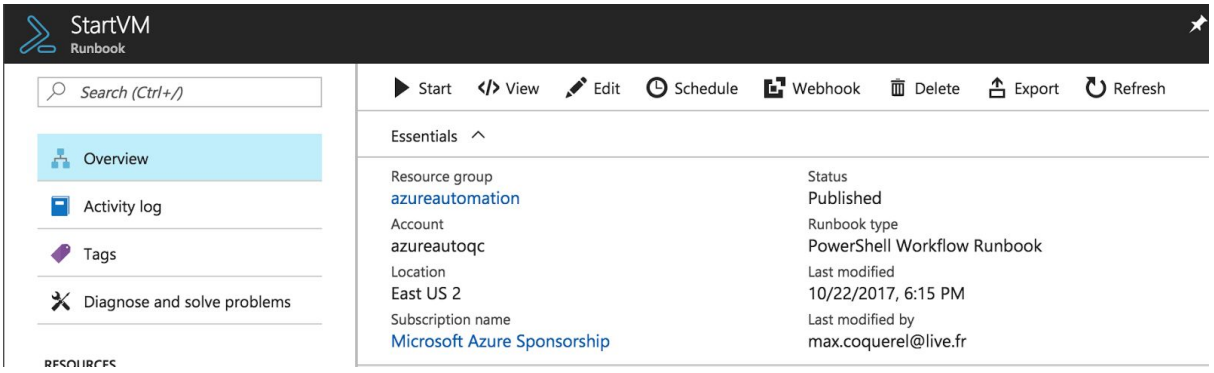

```
workflow StartVM
{
  Param(
  [string]$VMName,
  [string]$ResourceGroupName
 )
$Conn = Get-AutomationConnection -Name AzureRunAsConnection
Add-AzureRMAccount -ServicePrincipal -Tenant $Conn.TenantID -ApplicationId
```

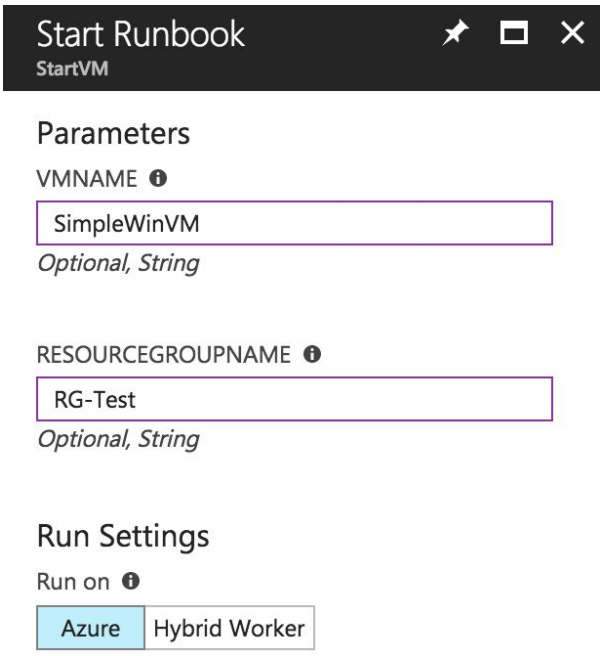
```
$Conn.ApplicationID -CertificateThumbprint $Conn.CertificateThumbprint
Start-AzureRmVM -Name $VMName -ResourceGroupName $ResourceGroupName
}
```



Click “Save” then click “Publish”.

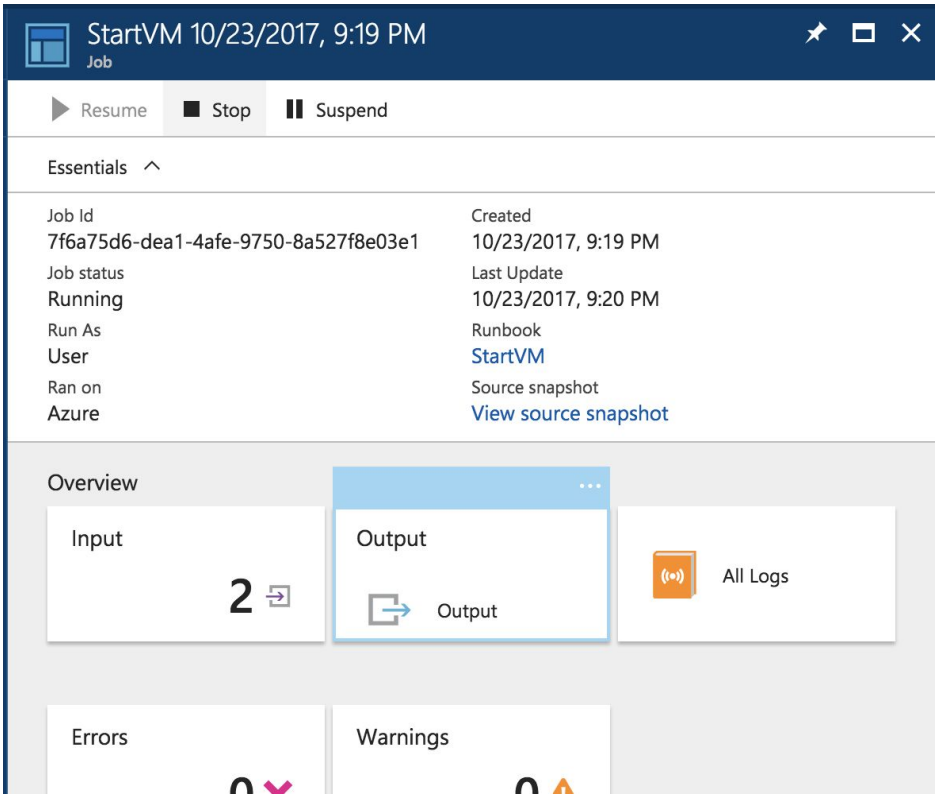


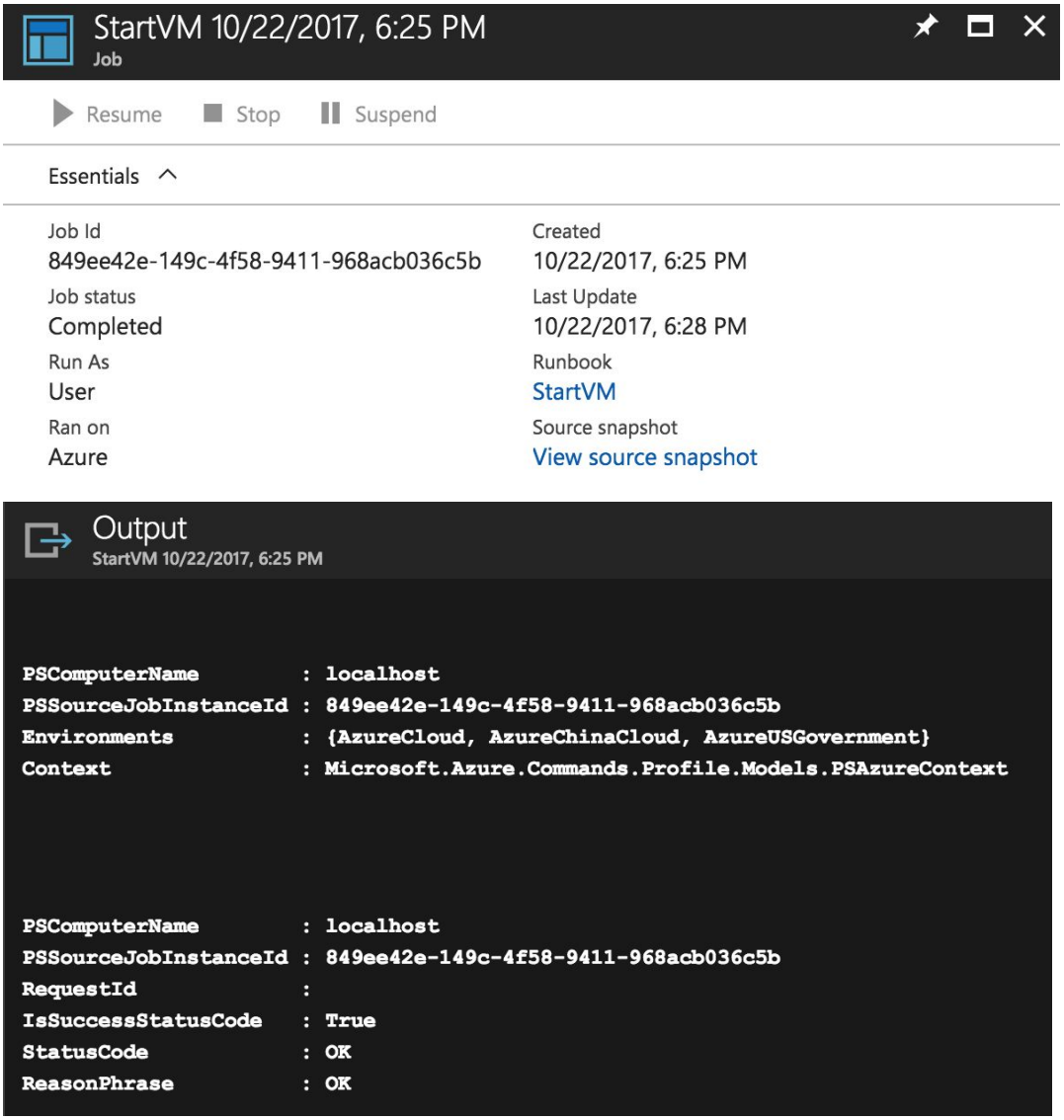




Click “OK” to start the runbook.

Click the “Output” button to view the output log.




StartVM 10/22/2017, 6:25 PM
Job
Resume
Stop
Suspend
Essentials
Job Id
849ee42e-149c-4f58-9411-968acb036c5b
Created
10/22/2017, 6:25 PM
Job status
Completed
Last Update
10/22/2017, 6:28 PM
Run As
User
Runbook
StartVM
Ran on
Azure
Source snapshot
View source snapshot
Output
StartVM 10/22/2017, 6:25 PM
PSComputerName : localhost
PSSourceJobInstanceId : 849ee42e-149c-4f58-9411-968acb036c5b
Environments : {AzureCloud, AzureChinaCloud, AzureUSGovernment}
Context : Microsoft.Azure.Commands.Profile.Models.PSAzureContext
PSComputerName : localhost
PSSourceJobInstanceId : 849ee42e-149c-4f58-9411-968acb036c5b
RequestId :
IsSuccessStatusCode : True
StatusCode : OK
ReasonPhrase : OK

Lab 6: Continuous deployment in Azure with Jenkins CI
Level: 300
Tested: Max | Olivier (jusqu'à Step 6, 45 minutes) | Tidjani (xx minutes)

Requirements:

- Azure Subscription
- Azure Cloud Shell (PowerShell)
- Github

Step 1 : Init Config file (Jenkins Deployment)

```
maxime@Azure:/usr/maxime/clouddrive$ vim cloud-init-jenkins.txt
```

```
cd c:\users\containeradministrator\CloudDrive\

vim cloud-init-jenkins.txt
```

```
#cloud-config
package_upgrade: true
write_files:
  - path: /etc/systemd/system/docker.service.d/docker.conf
    content: |
      [Service]
        ExecStart=
        ExecStart=/usr/bin/dockerd
  - path: /etc/docker/daemon.json
    content: |
      {
        "hosts": ["fd://","tcp://127.0.0.1:2375"]
      }
runcmd:
  - wget -q -O - https://jenkins-ci.org/debian/jenkins-ci.org.key | apt-key add -
  - sh -c 'echo deb http://pkg.jenkins-ci.org/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'
  - apt-get update && apt-get install jenkins -y
  - curl -sSL https://get.docker.com/ | sh
  - usermod -aG docker azureuser
  - usermod -aG docker jenkins
  - service jenkins restart
```

Step 2 : Jenkins Deployment

```
az group create --name myResourceGroupJenkins --location eastus
```

```
maxime@Azure:/usr/maxime/clouddrive$ az group create --name myResourceGroupJenkins --location eastus
{
  "id": "/subscriptions/7c                                    /resourceGroups/myResourceGroupJenkins",
  "location": "eastus",
  "managedBy": null,
  "name": "myResourceGroupJenkins",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null
}
```

az vm create --resource-group myResourceGroupJenkins --name myVM --image UbuntuLTS --admin-username azureuser --generate-ssh-keys --custom-data cloud-init-jenkins.txt

```
maxime@Azure:/usr/maxime/clouddrive$ az vm create --resource-group myResourceGroupJenkins --name myVM --image UbuntuLTS
--admin-username azureuser --generate-ssh-keys --custom-data cloud-init-jenkins.txt
{
  "fqdns": "",
  "id": "/subscriptions/7c                                /resourceGroups/myResourceGroupJenkins/providers/Microsoft.C
ompute/virtualMachines/myVM",
  "location": "eastus",
  "macAddress": "00-0D-3A-1A-7C-CA",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "40.71.251.76",
  "resourceGroup": "myResourceGroupJenkins",
  "zones": ""
}
```

az vm open-port --resource-group myResourceGroupJenkins --name myVM --port 8080 --priority 1001

az vm open-port --resource-group myResourceGroupJenkins --name myVM --port 1337 --priority 1002

```
maxime@Azure:/usr/maxime/clouddrive$ az vm open-port --resource-group myResourceGroupJenkins --name myVM --port 8080 --p
riority 1001
{
  "defaultSecurityRules": [
    {
```

```
maxime@Azure:/usr/maxime/clouddrive$ az vm open-port --resource-group myResourceGroupJenkins --name myVM --port 8080 --p
riority 1001
{
  "defaultSecurityRules": [
```

az vm show --resource-group myResourceGroupJenkins --name myVM -d --query [publicIps] --o tsv

```
maxime@Azure:/usr/maxime/clouddrive$ az vm show --resource-group myResourceGroupJenkins --name myVM -d --query [publicIp
s] --o tsv
40.71.251.76
```

ssh azureuser@<publicIps>

```
maxime@Azure:/usr/maxime/clouddrive$ ssh azureuser@40.71.251.76
The authenticity of host '40.71.251.76 (40.71.251.76)' can't be established.
ECDSA key fingerprint is SHA256:evNteoBXORLBd0drAGTag/+FII0VOuPIj59iTvympFA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.71.251.76' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.11.0-1013-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.
```

Step 3: Install Azure CLI in Jenkins Server

```
echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ wheezy main" |
sudo tee /etc/apt/sources.list.d/azure-cli.list
```

```
azureuser@myVM:~$ echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ wheezy main" | \
>       sudo tee /etc/apt/sources.list.d/azure-cli.list
deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ wheezy main
azureuser@myVM:~$
```

```
sudo apt-key adv --keyserver packages.microsoft.com --recv-keys 417A0893
```

```
azureuser@myVM:~$ sudo apt-key adv --keyserver packages.microsoft.com --recv-keys 417A0893
Executing: /tmp/tmp.W1HWvOdk5E/gpg.1.sh --keyserver
packages.microsoft.com
--recv-keys
417A0893
gpg: requesting key 417A0893 from hkp server packages.microsoft.com
gpg: key 417A0893: public key "MS Open Tech <interop@microsoft.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

```
sudo apt-get install apt-transport-https
```

```
azureuser@myVM:~$ sudo apt-get install apt-transport-https
Reading package lists... Done
Building dependency tree
Reading state information... Done
apt-transport-https is already the newest version (1.2.24).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
sudo apt-get update && sudo apt-get install azure-cli
```

```
azureuser@myVM:~$ sudo apt-get update && sudo apt-get install azure-cli
Ign:1 http://pkg.jenkins-ci.org/debian-stable binary/ InRelease
Hit:2 http://pkg.jenkins-ci.org/debian-stable binary/ Release
Hit:4 https://download.docker.com/linux/ubuntu xenial InRelease
Get:5 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Hit:6 http://azure.archive.ubuntu.com/ubuntu xenial InRelease
```

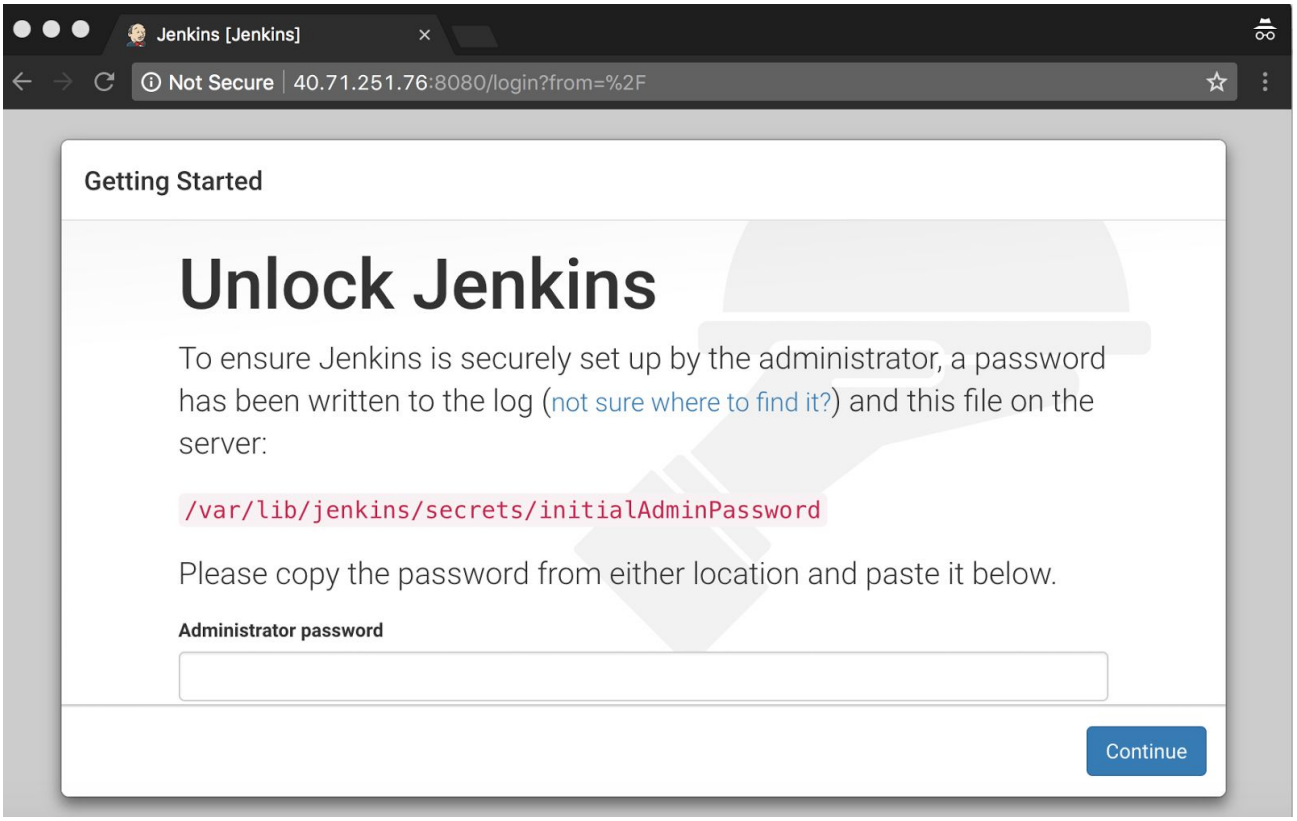
```
Preparing to unpack .../azure-cli_2.0.19-1_all.deb ...
Unpacking azure-cli (2.0.19-1) ...
Setting up azure-cli (2.0.19-1) ...
azureuser@myVM:~$
```

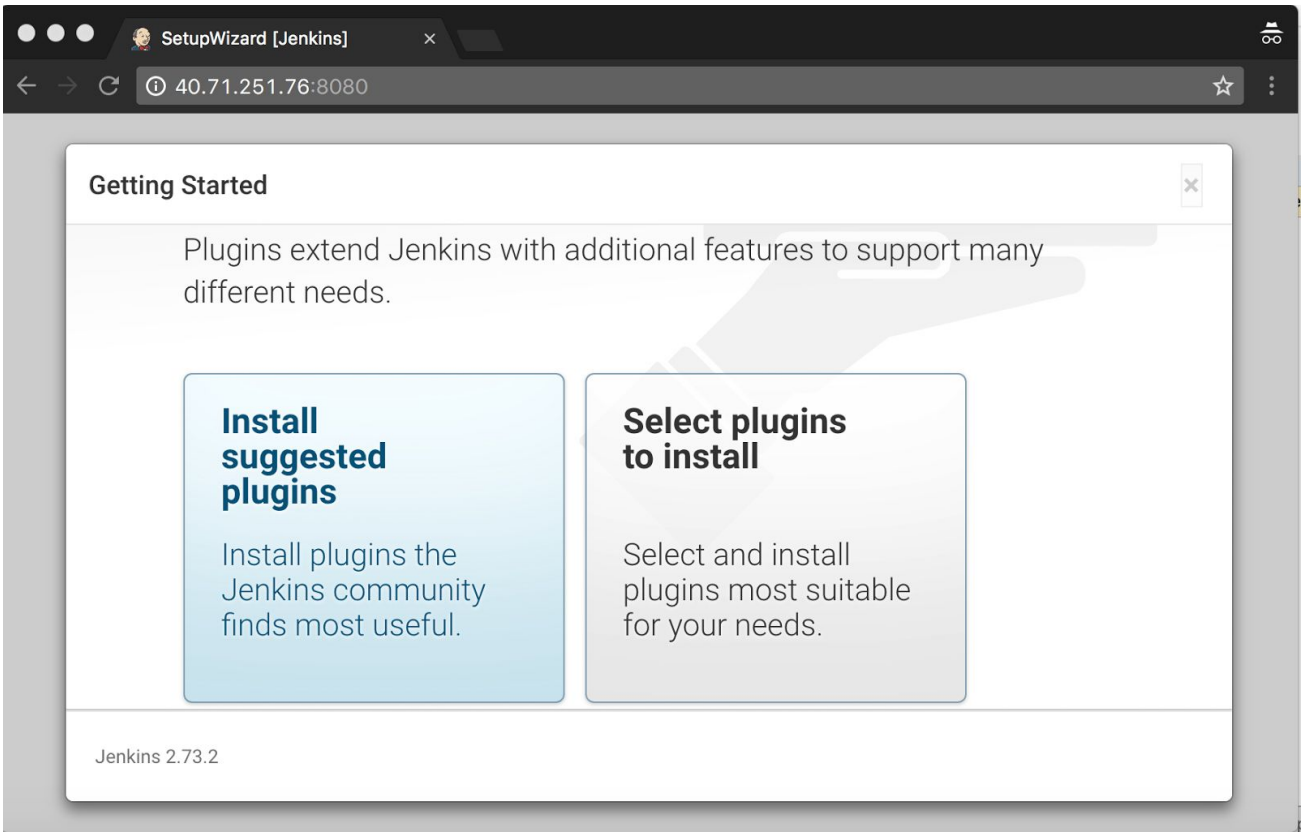
Step 4: Unlock Jenkins

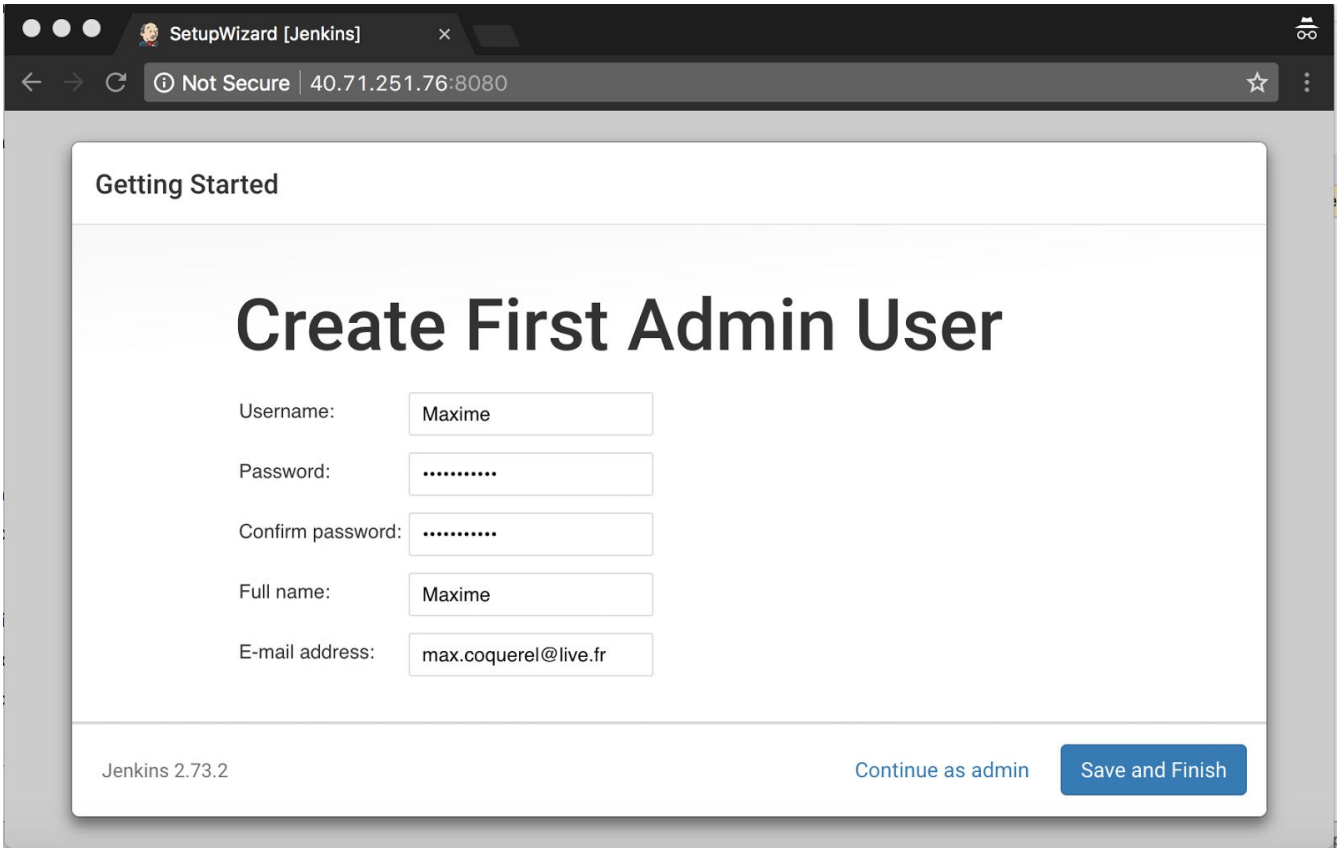
```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

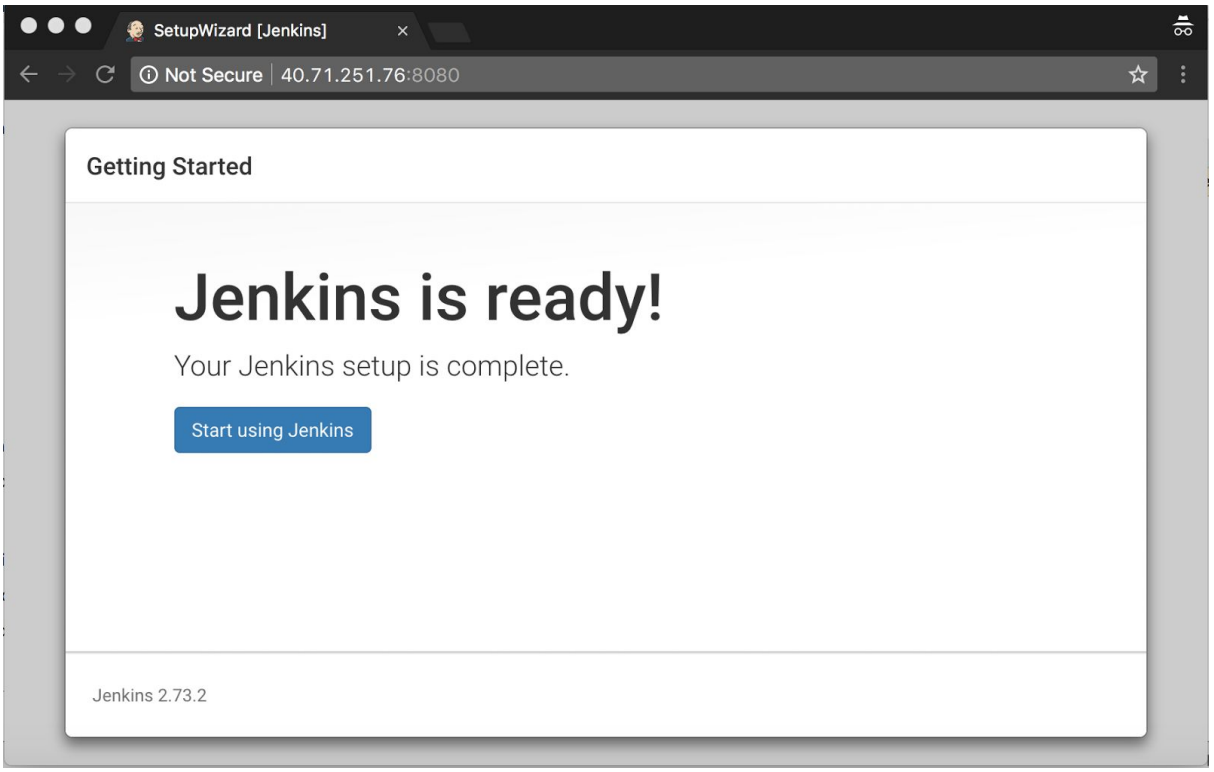
```
azureuser@myVM:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
efcbb4a9ba8444aba2d2cc9ceabc12f2
```

```
http://<publicIps>:8080
```




SetupWizard [Jenkins]
40.71.251.76:8080
Getting Started
Plugins extend Jenkins with additional features to support many different needs.
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.
Jenkins 2.73.2


SetupWizard [Jenkins]
Not Secure | 40.71.251.76:8080
Getting Started
Create First Admin User
Username: Maxime
Password:
Confirm password:
Full name: Maxime
E-mail address: max.coquerel@live.fr
Jenkins 2.73.2
Continue as admin
Save and Finish



Step 5: Create Azure Service Principal

Read this : https://docs.microsoft.com/en-us/azure/jenkins/jenkins-azure-vm-agents

```
az ad sp create-for-rbac --name jenkins_sp --password jenkins_sp
```

```
{
  "appId": "BBBBBBBB-BBBB-BBBB-BBBB-BBBBBBBBBBB",
  "displayName": "jenkins_sp",
  "name": "http://jenkins_sp",
  "password": "secure_password",
  "tenant": "CCCCCCCC-CCCC-CCCC-CCCCCCCCCCC"
}
```

```
az account list
```

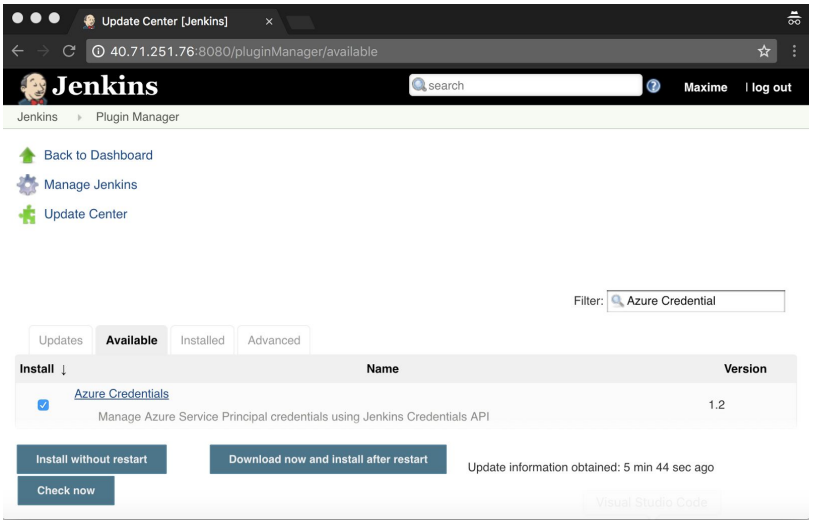
```
{
    "cloudName": "AzureCloud",
    "id": "AAAAAAAA-AAAA-AAAA-AAAA-AAAAAAAAAAAA",
    "isDefault": true,
    "name": "Visual Studio Enterprise",
```

```
"state": "Enabled",
"tenantId": "CCCCCCCC-CCCC-CCCC-CCCC-CCCCCCCCCCCC",
"user": {
"name": "max@fabrikam.com",
"type": "user"
}
```

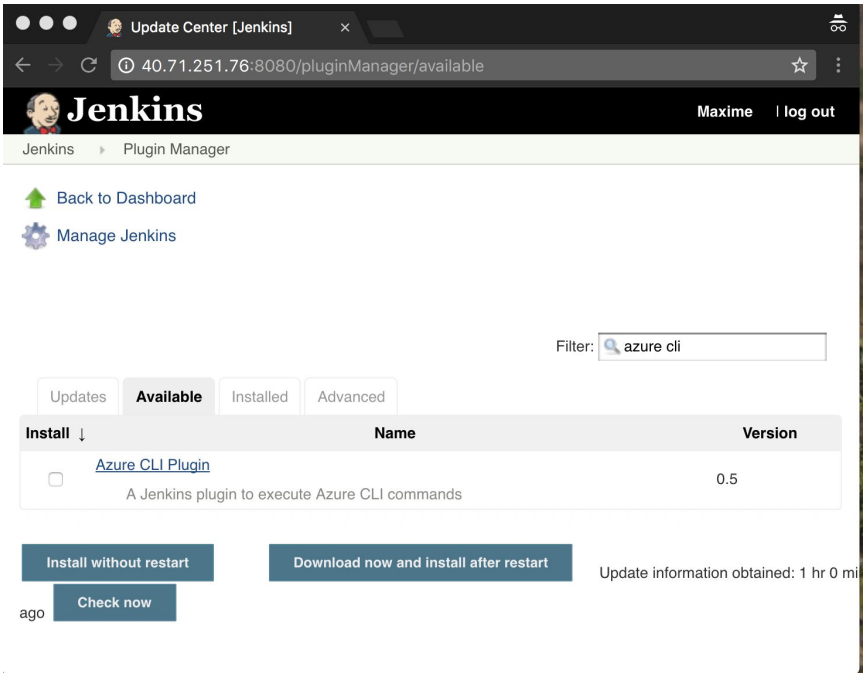
https://docs.microsoft.com/en-us/cli/azure/create-an-azure-service-principal-azure-cli?toc=%2Fazure%2Fazure-resource-manager%2Ftoc.json&view=azure-cli-latest

Step 6: Configure Jenkins Plugin

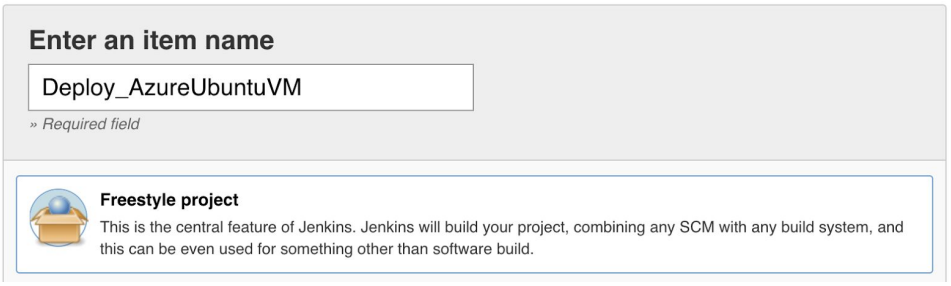
Install Azure Credentials

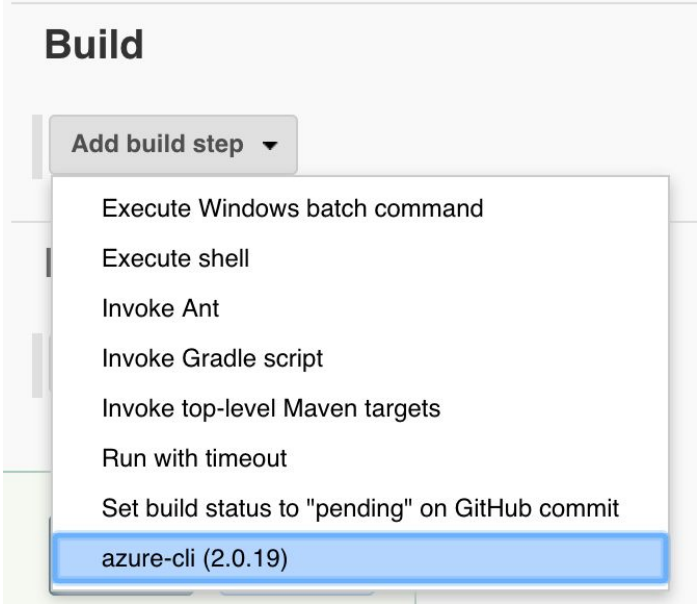


Install Azure CLI

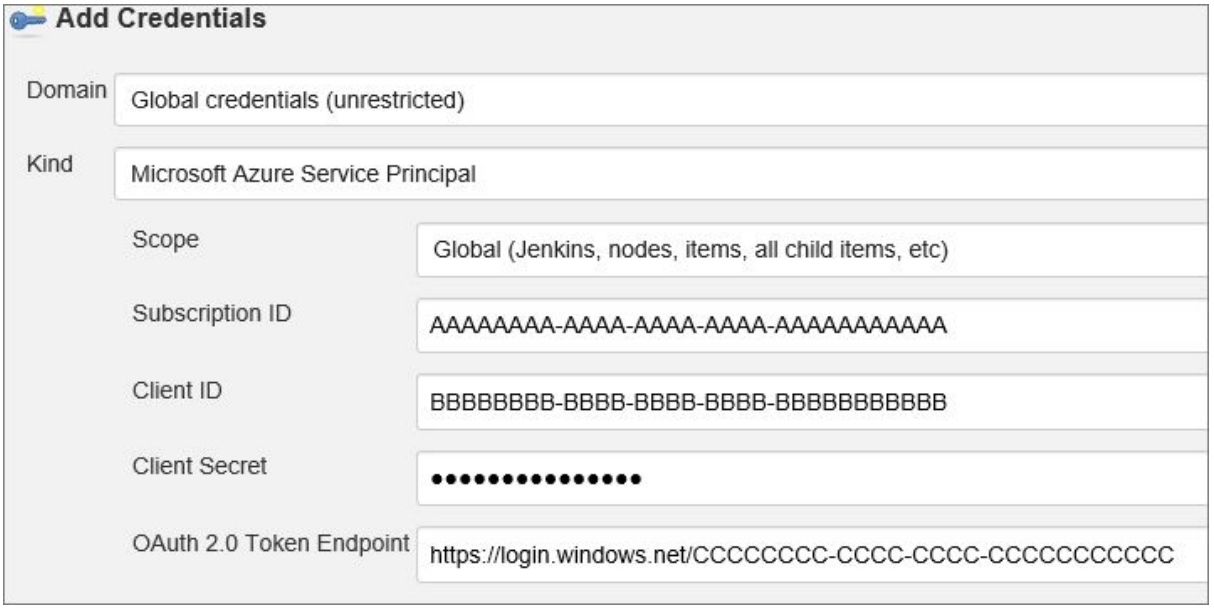


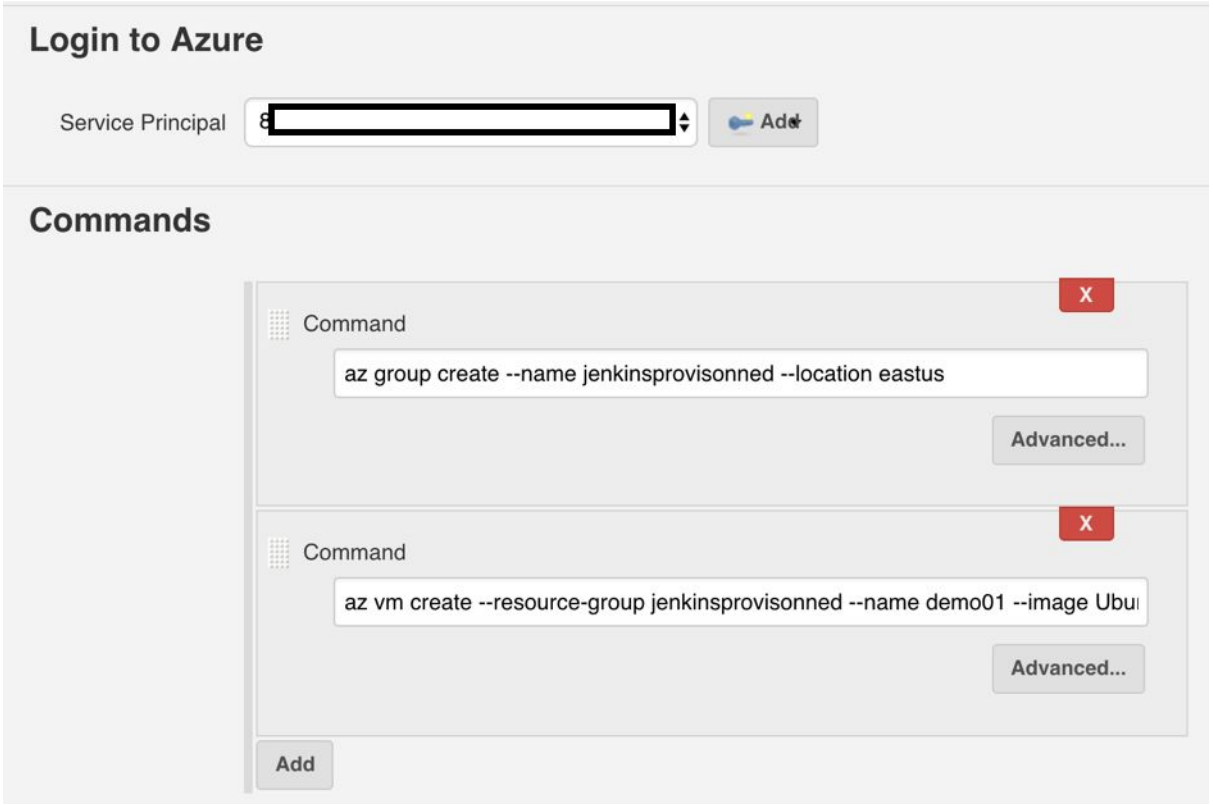
Step 7: Configure Jenkins Job - “Deploy Ubuntu VM from Jenkins”





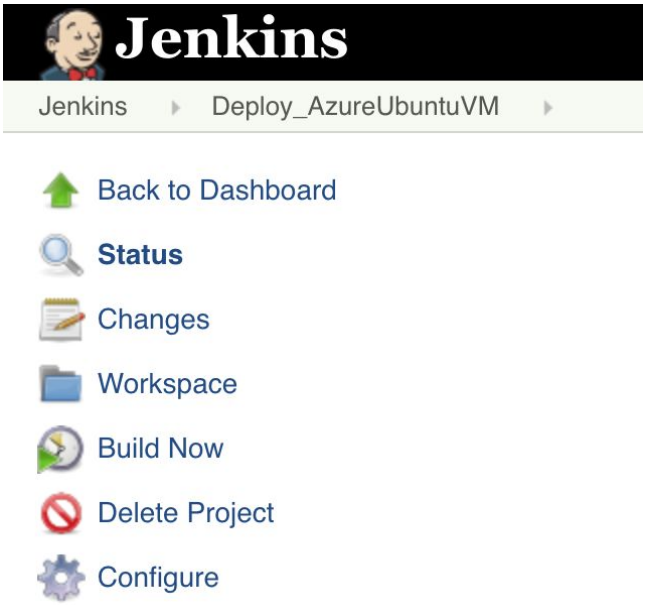
Please, refer to “Step 3: Create Azure Service Principal”

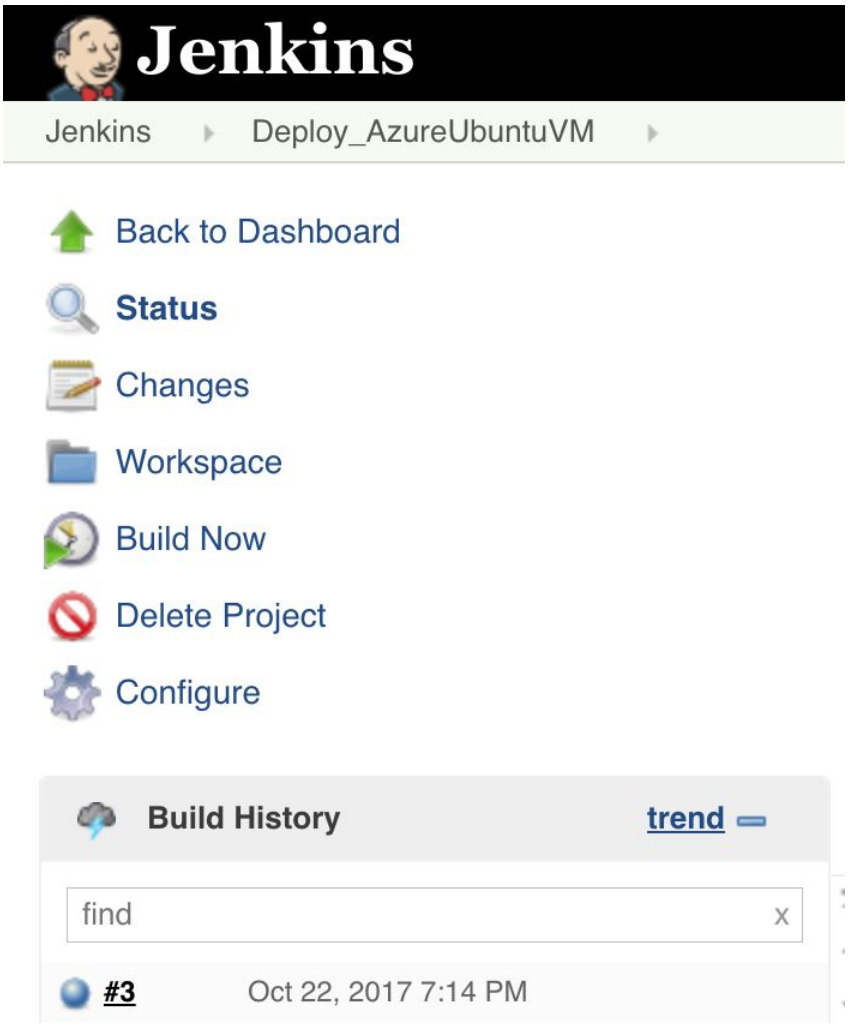


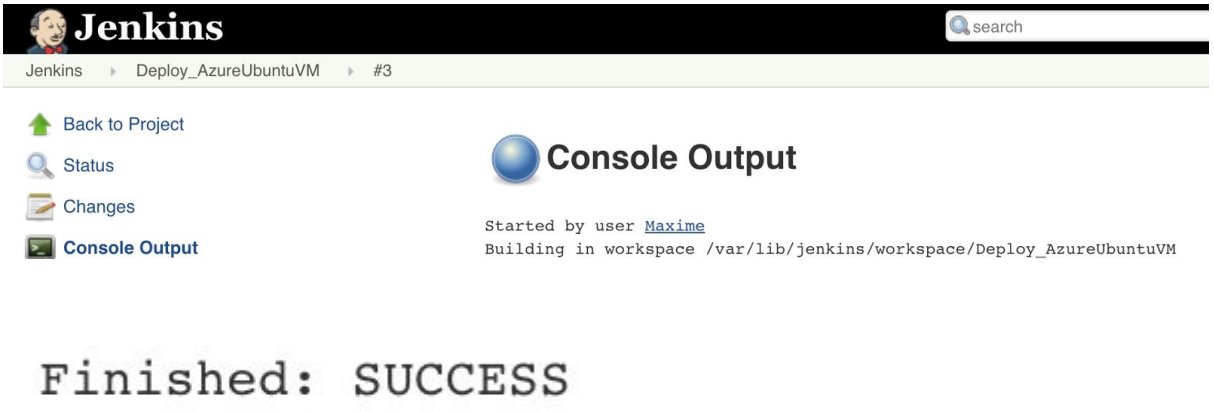



```
az group create --name jenkinsprovisonned --location eastus
```

```
az vm create --resource-group jenkinsprovisonned --name demo01 --image UbuntuLTS
--admin-username maxime --admin-password IloveAzureMeetupQC17!
```



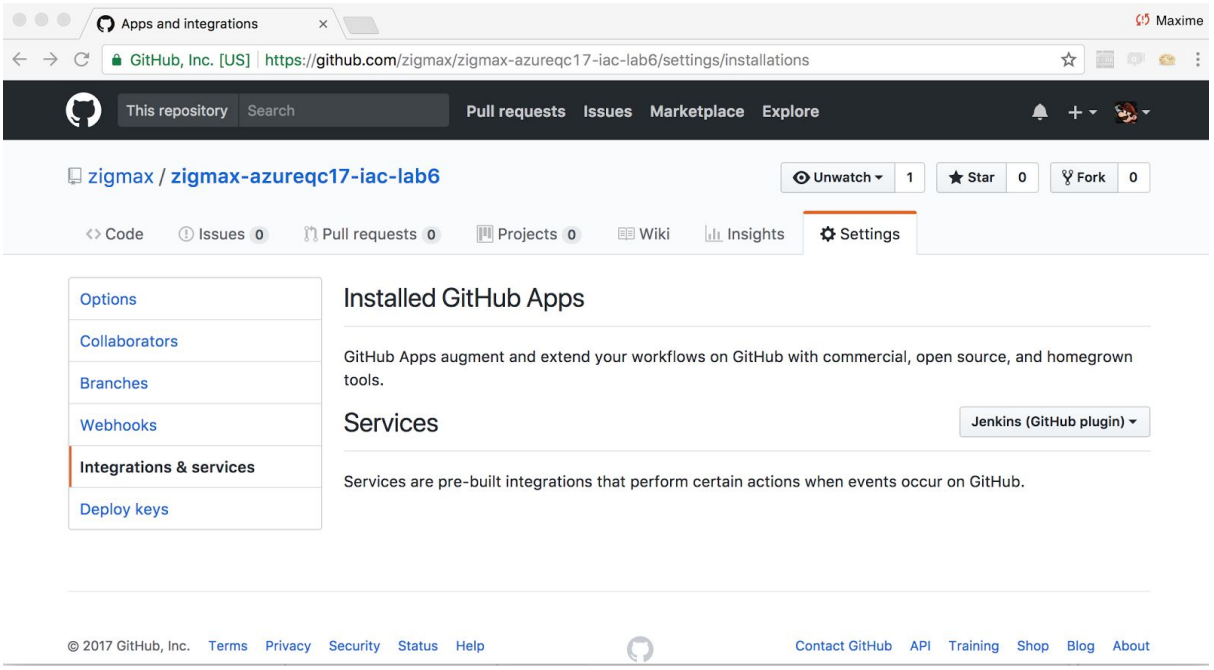


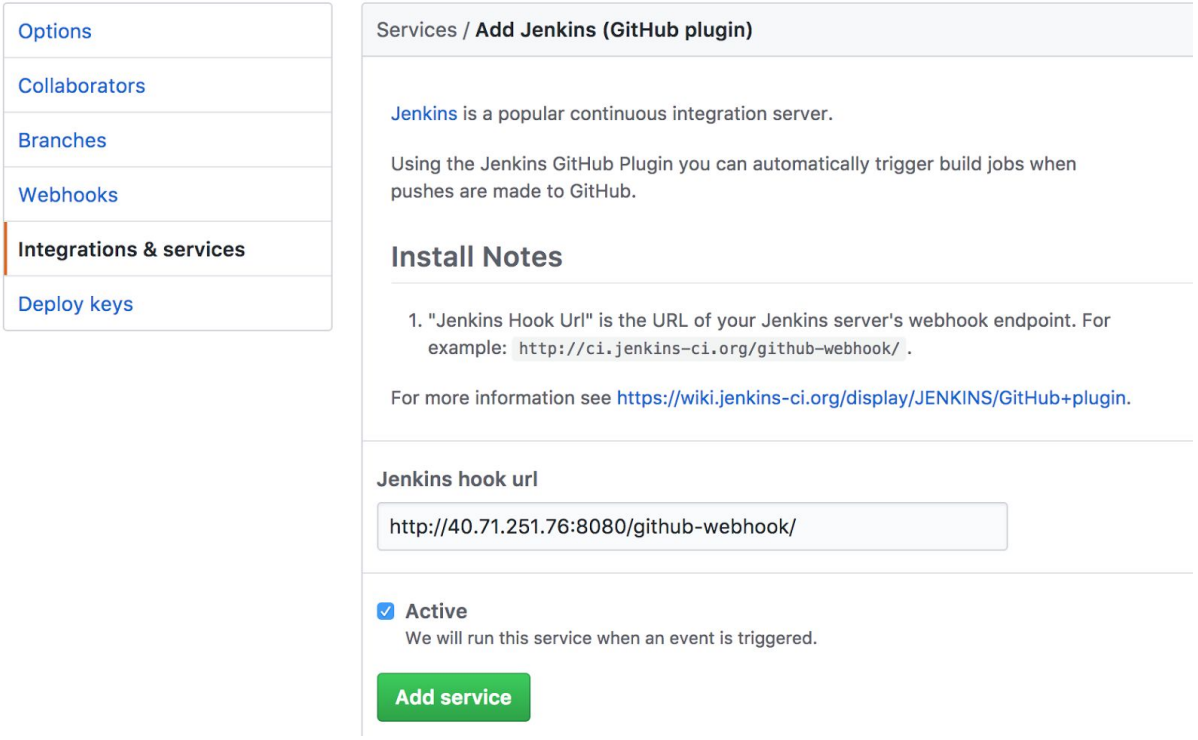


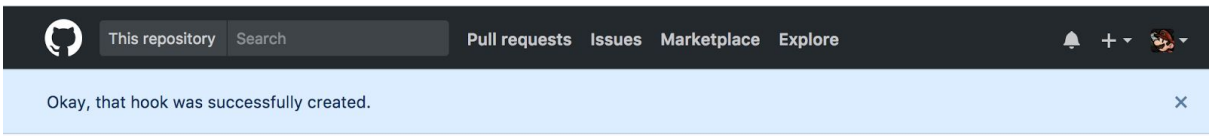
Step 8: Configure Jenkins Job - “Deploy in Azure from Github hook”

Requirement: Personal Github Account: https://github.com/

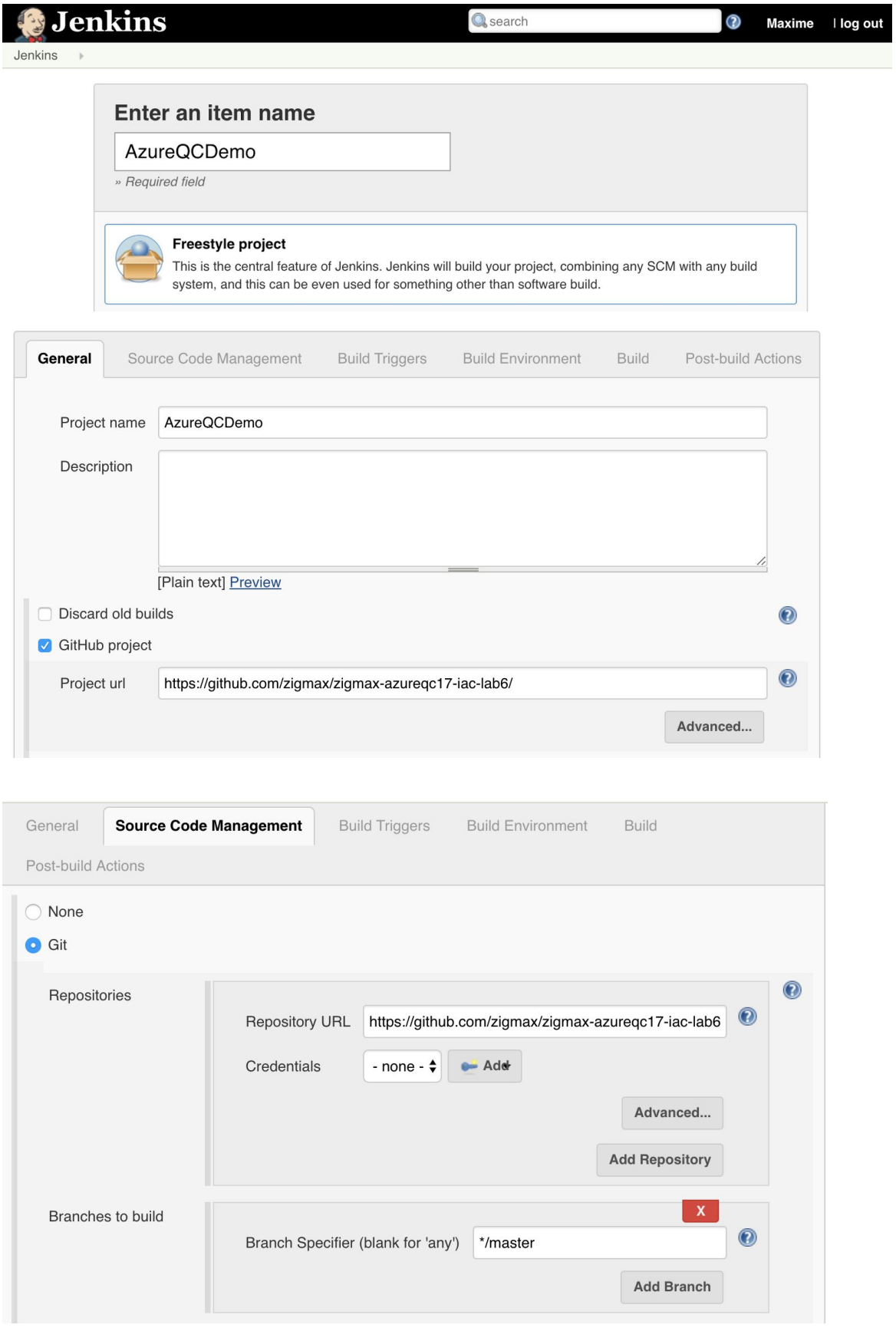
Configure Jenkins url Hook

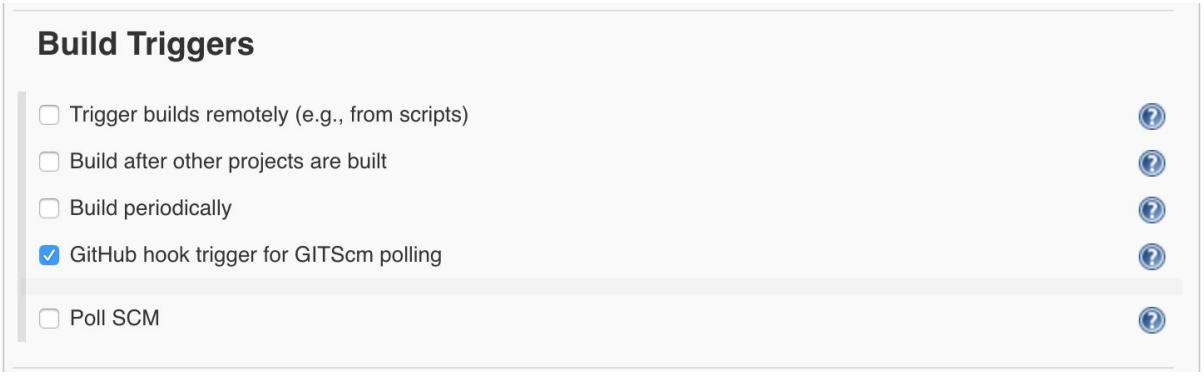


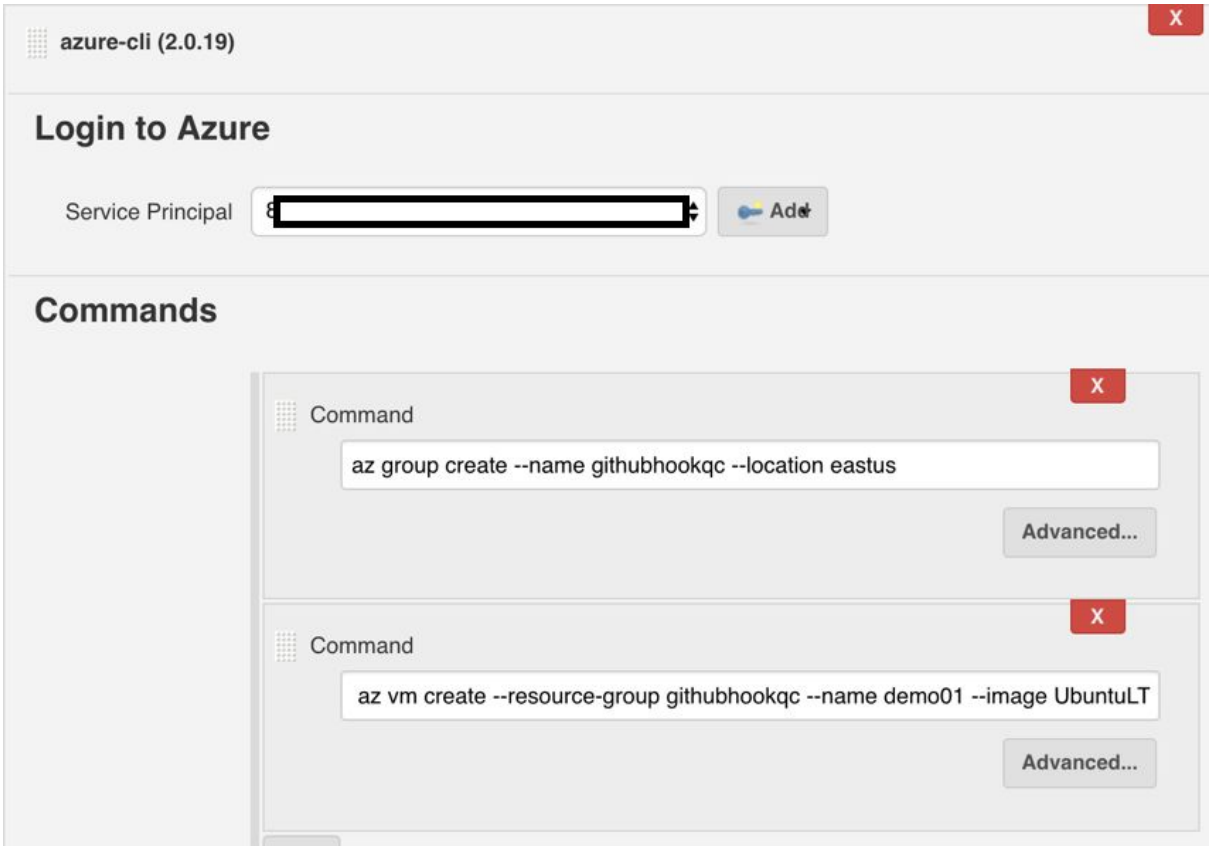




Create a new Jenkins : Freestyle project

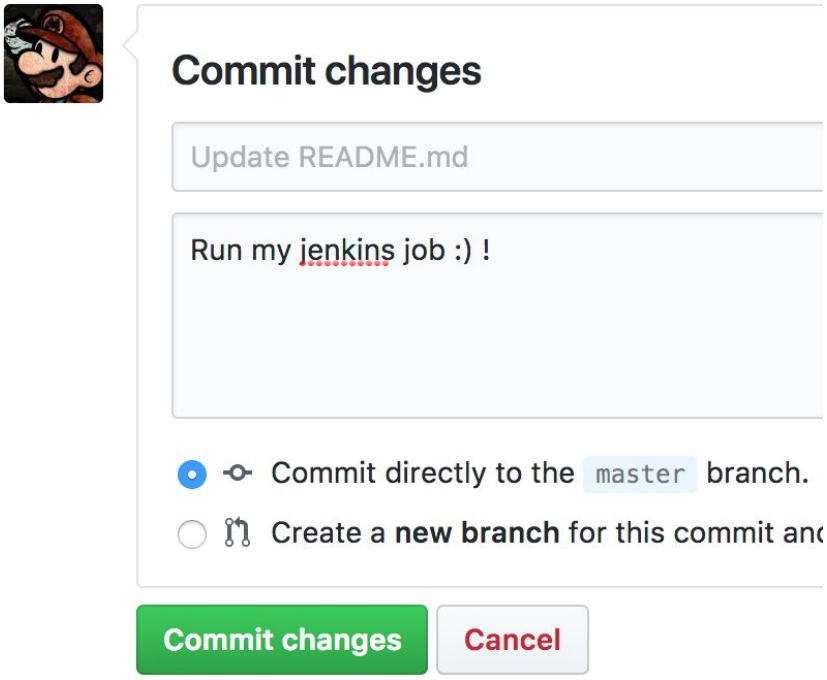

Jenkins
search
Maxime
log out
Jenkins
Enter an item name
AzureQCDemo
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
Project name
AzureQCDemo
Description
[Plain text] Preview
Discard old builds
GitHub project
Project url
https://github.com/zigmax/zigmax-azureqc17-iac-lab6/
Advanced...
General
Source Code Management
Build Triggers
Build Environment
Build
Post-build Actions
None
Git
Repositories
Repository URL
https://github.com/zigmax/zigmax-azureqc17-iac-lab6
Credentials
- none -
Add
Advanced...
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch

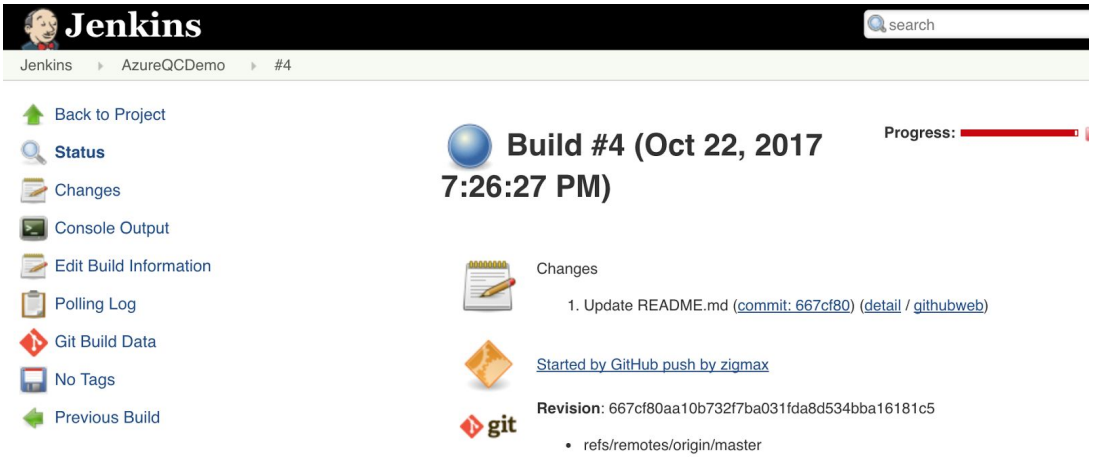




Run execution : Commit, what you want in your repo :)


zigmax / zigmax-azureqc17-iac-lab6
Code
Issues 0
Pull requests 0
Projects 0
Wiki
zigmax-azureqc17-iac-lab6 / README.md
or cancel
Edit file
Preview changes
1 # zigmax-azureqc17-iac-lab6
2
3 Run my Jenkins Job :) !


Commit changes
Update README.md
Run my jenkins job :) !
Commit directly to the master branch.
Create a new branch for this commit anc
Commit changes
Cancel


Jenkins
search
Jenkins AzureQCDemo #4
Back to Project
Status
Changes
Console Output
Edit Build Information
Polling Log
Git Build Data
No Tags
Previous Build
Build #4 (Oct 22, 2017 7:26:27 PM)
Progress:
Changes
1. Update README.md (commit: 667cf80) (detail / githubweb)
Started by GitHub push by zigmax
Revision: 667cf80aa10b732f7ba031fda8d534bba16181c5
refs/remotes/origin/master

```
  "location": "eastus",
  "macAddress": "00-0D-3A-1A-EA-6A",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.168.7.165",
  "resourceGroup": "githubhookqc",
  "zones": ""
}

Finished: SUCCESS
```

Annexes:

Lab 4: Create an Azure VM with HashiCorp Terraform

Output : terraform plan

```
maxime@Azure:~$ terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


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

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)

Terraform will perform the following actions:

 <= data.azurerm_public_ip.test
          id:                                 <computed>
          domain_name_label:                        <computed>
          fqdn:                               <computed>
      idle_timeout_in_minutes:                    <computed>
          ip_address:                           <computed>
          name:                                "pubip"
      resource_group_name:                          "acctestrg"
      tags.%:                                <computed>

  + azurerm_managed_disk.test
          id:                                 <computed>
      create_option:                            "Empty"
      disk_size_gb:                            "1023"
          location:                             "westus2"
          name:                                "datadisk_existing"
      resource_group_name:                          "acctestrg"
          source_uri:                           <computed>
      storage_account_type:                         "Standard_LRS"
          tags.%:                               <computed>

  + azurerm_network_interface.test
          id:                                 <computed>
          applied_dns_servers.#:                    <computed>
      dns_servers.#:                          <computed>
      enable_ip_forwarding:                       "false"
      internal_dns_name_label:                      <computed>
      internal_fqdn:                           <computed>
```

```
    ip_configuration.#:                                   "1"
    ip_configuration.0.load_balancer_backend_address_pools_ids.#:       <computed>
    ip_configuration.0.load_balancer_inbound_nat_rules_ids.#:           <computed>
    ip_configuration.0.name:                              "testconfiguration1"
        ip_configuration.0.primary:                       <computed>
    ip_configuration.0.private_ip_address:                "10.0.2.5"
    ip_configuration.0.private_ip_address_allocation:     "static"
    ip_configuration.0.public_ip_address_id:
"${azurerm_public_ip.test.id}"
    ip_configuration.0.subnet_id:                         "${azurerm_subnet.test.id}"
        location:                                         "westus2"
    mac_address:                                          <computed>
        name:                                             "acctni"
    private_ip_address:                                   <computed>
    resource_group_name:                                  "acctestrg"
        tags.%:                                           <computed>
    virtual_machine_id:                                   <computed>

  + azurerm_public_ip.test
        id:                                               <computed>
        fqdn:                                             <computed>
    idle_timeout_in_minutes:                              "30"
        ip_address:                                       <computed>
        location:                                         "westus2"
        name:                                             "pubip"
    public_ip_address_allocation:                         "dynamic"
    resource_group_name:                                  "acctestrg"
        tags.%:                                           "1"
    tags.environment:                                     "test"

  + azurerm_resource_group.test
        id:                                               <computed>
        location:                                         "westus2"
        name:                                             "acctestrg"
        tags.%:                                           <computed>

  + azurerm_subnet.test
        id:                                               <computed>
    address_prefix:                                       "10.0.2.0/24"
    ip_configurations.#:                                  <computed>
        name:                                             "acctsub"
    network_security_group_id:                            <computed>
    resource_group_name:                                  "acctestrg"
    route_table_id:                                       <computed>
    virtual_network_name:                                 "acctvn"

  + azurerm_virtual_machine.test
        id:                                               <computed>
    availability_set_id:                                  <computed>
    delete_data_disks_on_termination:                     "false"
```

```
    delete_os_disk_on_termination:                    "false"
        location:                                     "westus2"
        name:                                         "acctvm"
    network_interface_ids.#:                          <computed>
    os_profile.#:                                     "1"
    os_profile.3971669894.admin_password:             <sensitive>
    os_profile.3971669894.admin_username:             "qcazureadmin"
    os_profile.3971669894.computer_name:              "hostname"
    os_profile.3971669894.custom_data:                <computed>
    os_profile_linux_config.#:                        "1"
    os_profile_linux_config.2972667452.disable_password_authentication: "false"
    os_profile_linux_config.2972667452.ssh_keys.#:    "0"
    resource_group_name:                              "acctestrg"
    storage_data_disk.#:                              "2"
    storage_data_disk.0.caching:                      <computed>
    storage_data_disk.0.create_option:                "Empty"
    storage_data_disk.0.disk_size_gb:                 "1023"
    storage_data_disk.0.lun:                          "0"
        storage_data_disk.0.managed_disk_id:          <computed>
    storage_data_disk.0.managed_disk_type:            "Standard_LRS"
    storage_data_disk.0.name:                         "datadisk_new"
        storage_data_disk.1.caching:                  <computed>
    storage_data_disk.1.create_option:                "Attach"
    storage_data_disk.1.disk_size_gb:                 "1023"
        storage_data_disk.1.lun:                      "1"
    storage_data_disk.1.managed_disk_id:
"${azurerm_managed_disk.test.id}"
    storage_data_disk.1.managed_disk_type:            <computed>
    storage_data_disk.1.name:                         "datadisk_existing"
    storage_image_reference.#:                        "1"
    storage_image_reference.1458860473.id:            ""
        storage_image_reference.1458860473.offer:     "UbuntuServer"
    storage_image_reference.1458860473.publisher:     "Canonical"
    storage_image_reference.1458860473.sku:           "16.04-LTS"
    storage_image_reference.1458860473.version:       "latest"
    storage_os_disk.#:                                "1"
    storage_os_disk.429214147.caching:                "ReadWrite"
        storage_os_disk.429214147.create_option:      "FromImage"
    storage_os_disk.429214147.disk_size_gb:           ""
    storage_os_disk.429214147.image_uri:              ""
    storage_os_disk.429214147.managed_disk_id:        <computed>
    storage_os_disk.429214147.managed_disk_type:      "Standard_LRS"
    storage_os_disk.429214147.name:                   "myosdisk1"
    storage_os_disk.429214147.os_type:                ""
    storage_os_disk.429214147.vhd_uri:                ""
        tags.%:                                       "1"
    tags.environment:                                 "demomeetupazure"
        vm_size:                                      "Standard_DS1_v2"

 + azurerm_virtual_network.test
```

```
      id:                                                    <computed>
  address_space.#:                                           "1"
  address_space.0:                                           "10.0.0.0/16"
      location:                                              "westus2"
      name:                                                  "acctvn"
  resource_group_name:                                       "acctestrg"
      subnet.#:                                              <computed>
      tags.%:                                                <computed>


Plan: 7 to add, 0 to change, 0 to destroy.

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

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

Output: Terraform Apply:

```
maxime@Azure:~$ terraform apply
azurerm_resource_group.test: Creating...
  location: "" => "westus2"
  name:           "" => "acctestrg"
  tags.%:   "" => "<computed>"
azurerm_resource_group.test: Creation complete after 1s (ID:
/subscriptions/7dxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/acctestrg)
azurerm_virtual_network.test: Creating...
  address_space.#:     "" => "1"
  address_space.0:     "" => "10.0.0.0/16"
  location:        "" => "westus2"
  name:                "" => "acctvn"
  resource_group_name: "" => "acctestrg"
  subnet.#:            "" => "<computed>"
  tags.%:              "" => "<computed>"
azurerm_managed_disk.test: Creating...
  create_option:          "" => "Empty"
  disk_size_gb:           "" => "1023"
  location:               "" => "westus2"
  name:                   "" => "datadisk_existing"
  resource_group_name:  "" => "acctestrg"
  source_uri:             "" => "<computed>"
  storage_account_type: "" => "Standard_LRS"
  tags.%:                 "" => "<computed>"
azurerm_virtual_network.test: Creation complete after 6s (ID:
/subscriptions/7xxxx-xxxx-xxxx-xxxx-...crosoft.Network/virtualNetworks/acctvn)
```

```
azurerm_subnet.test: Creating...
  address_prefix:        "" => "10.0.2.0/24"
  ip_configurations.#:       "" => "<computed>"
  name:                  "" => "acctsub"
  network_security_group_id: "" => "<computed>"
  resource_group_name:       "" => "acctestrg"
  route_table_id:        "" => "<computed>"
  virtual_network_name:      "" => "acctvn"
azurerm_subnet.test: Creation complete after 3s (ID:
/subscriptions/7dbxxxx-xxxx-xxxx-xxx-...virtualNetworks/acctvn/subnets/acctsub)
azurerm_network_interface.test: Creating...
  applied_dns_servers.#:                          "" => "<computed>"
  dns_servers.#:                                  "" => "<computed>"
  enable_ip_forwarding:                           "" => "false"
  internal_dns_name_label:                        "" => "<computed>"
  internal_fqdn:                                  "" => "<computed>"
  ip_configuration.#:                             "" => "1"
  ip_configuration.0.load_balancer_backend_address_pools_ids.#: "" => "<computed>"
  ip_configuration.0.load_balancer_inbound_nat_rules_ids.#:         "" => "<computed>"
  ip_configuration.0.name:                        "" => "testconfiguration1"
  ip_configuration.0.primary:                     "" => "<computed>"
  ip_configuration.0.private_ip_address:                  "" => "<computed>"
  ip_configuration.0.private_ip_address_allocation:       "" => "dynamic"
  ip_configuration.0.public_ip_address_id:                "" => "<computed>"
  ip_configuration.0.subnet_id:                   "" =>
"/subscriptions/7dxxxxxx-xxxx-xxx-xxx-xxxxxxxxxx/resourceGroups/acctestrg/providers/Mi
crosoft.Network/virtualNetworks/acctvn/subnets/acctsub"
  location:                                       "" => "westus2"
  mac_address:                                    "" => "<computed>"
  name:                                           "" => "acctni"
  private_ip_address:                             "" => "<computed>"
  resource_group_name:                            "" => "acctestrg"
  tags.%:                                         "" => "<computed>"
  virtual_machine_id:                             "" => "<computed>"
azurerm_managed_disk.test: Still creating... (10s elapsed)
azurerm_network_interface.test: Creation complete after 2s (ID:
/subscriptions/7dxxxxxx-xxxxx-xxxx-xxxx-...osoft.Network/networkInterfaces/acctni)
azurerm_managed_disk.test: Still creating... (20s elapsed)
azurerm_managed_disk.test: Still creating... (30s elapsed)
azurerm_managed_disk.test: Still creating... (40s elapsed)
azurerm_managed_disk.test: Still creating... (50s elapsed)
azurerm_managed_disk.test: Still creating... (1m0s elapsed)
azurerm_managed_disk.test: Creation complete after 1m2s (ID:
/subscriptions/7dxxxxxx-xxxx-xxx-xxx-...rosoft.Compute/disks/datadisk_existing)
azurerm_virtual_machine.test: Creating...
  availability_set_id:                            "" => "<computed>"
  delete_data_disks_on_termination:                       "" => "false"
  delete_os_disk_on_termination:                          "" => "false"
  location:                                       "" => "westus2"
  name:                                           "" => "acctvm"
```

```
  network_interface_ids.#:                                  "" => "1"
  network_interface_ids.476834197:                              "" =>
"/subscriptions/7dxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/acctestrg/providers/
Microsoft.Network/networkInterfaces/acctni"
  os_profile.#:                                         "" => "1"
  os_profile.3971669894.admin_password:                          "<sensitive>" =>
"<sensitive>"
  os_profile.3971669894.admin_username:                         "" => "qcazureadmin"
  os_profile.3971669894.computer_name:                              "" => "hostname"
  os_profile.3971669894.custom_data:                            "" => "<computed>"
  os_profile_linux_config.#:                                "" => "1"
  os_profile_linux_config.2972667452.disable_password_authentication: "" => "false"
  os_profile_linux_config.2972667452.ssh_keys.#:                            "" => "0"
  resource_group_name:                                    "" => "acctestrg"
  storage_data_disk.#:                                    "" => "2"
  storage_data_disk.0.caching:                              "" => "<computed>"
  storage_data_disk.0.create_option:                            "" => "Empty"
  storage_data_disk.0.disk_size_gb:                             "" => "1023"
  storage_data_disk.0.lun:                                  "" => "0"
  storage_data_disk.0.managed_disk_id:                          "" => "<computed>"
  storage_data_disk.0.managed_disk_type:                                "" => "Standard_LRS"
  storage_data_disk.0.name:                                 "" => "datadisk_new"
  storage_data_disk.1.caching:                              "" => "<computed>"
  storage_data_disk.1.create_option:                            "" => "Attach"
  storage_data_disk.1.disk_size_gb:                             "" => "1023"
  storage_data_disk.1.lun:                                  "" => "1"
  storage_data_disk.1.managed_disk_id:                          "" =>
"/subscriptions/7dxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx/resourceGroups/acctestrg/providers/
Microsoft.Compute/disks/datadisk_existing"
  storage_data_disk.1.managed_disk_type:                                "" => "<computed>"
  storage_data_disk.1.name:                                 "" => "datadisk_existing"
  storage_image_reference.#:                                "" => "1"
  storage_image_reference.1458860473.id:                                "" => ""
  storage_image_reference.1458860473.offer:                             "" => "UbuntuServer"
  storage_image_reference.1458860473.publisher:                         "" => "Canonical"
  storage_image_reference.1458860473.sku:                               "" => "16.04-LTS"
  storage_image_reference.1458860473.version:                           "" => "latest"
  storage_os_disk.#:                                        "" => "1"
  storage_os_disk.429214147.caching:                            "" => "ReadWrite"
  storage_os_disk.429214147.create_option:                              "" => "FromImage"
  storage_os_disk.429214147.disk_size_gb:                               "" => ""
  storage_os_disk.429214147.image_uri:                          "" => ""
  storage_os_disk.429214147.managed_disk_id:                            "" => "<computed>"
  storage_os_disk.429214147.managed_disk_type:                          "" => "Standard_LRS"
  storage_os_disk.429214147.name:                               "" => "myosdisk1"
  storage_os_disk.429214147.os_type:                            "" => ""
  storage_os_disk.429214147.vhd_uri:                            "" => ""
  tags.%:                                               "" => "1"
  tags.environment:                                         "" => "demomeetupazure"
  vm_size:                                              "" => "Standard_DS1_v2"
```

```
azurerm_virtual_machine.test: Still creating... (10s elapsed)
azurerm_virtual_machine.test: Still creating... (20s elapsed)
azurerm_virtual_machine.test: Still creating... (30s elapsed)
azurerm_virtual_machine.test: Still creating... (40s elapsed)
azurerm_virtual_machine.test: Still creating... (50s elapsed)
azurerm_virtual_machine.test: Still creating... (1m0s elapsed)
azurerm_virtual_machine.test: Still creating... (1m10s elapsed)
azurerm_virtual_machine.test: Still creating... (1m20s elapsed)
azurerm_virtual_machine.test: Still creating... (1m30s elapsed)
azurerm_virtual_machine.test: Still creating... (1m40s elapsed)
azurerm_virtual_machine.test: Still creating... (1m50s elapsed)
azurerm_virtual_machine.test: Still creating... (2m0s elapsed)
azurerm_virtual_machine.test: Creation complete after 2m3s (ID:
/subscriptions/7dxxxxx-xxxx-xxxx-xxxx-...crosoft.Compute/virtualMachines/acctvm)

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
```