Microsoft Cloud Demo

Open Source App Dev

February 2017

© 2017 Microsoft Corporation. All rights reserved. This document is confidential and proprietary to Microsoft. Internal use only. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

## Contents

Containers and DevOps Hackathon 1

Containers and DevOps Hackathon answers 23

# Open source Containers and DevOps Overview with Azure

## Setup of Environment

Setup of Client software @ Azure environment:

* MS Internal - Configure Internal MS Azure account - <https://microsoft.sharepoint.com/teams/AzAccess/Pages/AzureTC.aspx>
* Download MobaXterm - <http://mobaxterm.mobatek.net/download.html>

GITHUB Repository:

* <https://github.com/dansand71/OSSonAzure>

**Azure Setup**

* Download the latest Azure CLI v2 - <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>
* Create a directory /source on your setup machine

**mkdir /source**

**cd /source**

curl -L https://aka.ms/InstallAzureCli | bash

**# Install git and Clone the GitRepository:**

**sudo yum install git**

**sudo git clone** <https://github.com/dansand71/OSSonAzure>

**# Mark the scripts executable**

**sudo chmod +x /source/OSSonAzure/createAzureDemoEnvironment.sh /source/OSSonAzure/deleteAzureDemoEnvironment.sh**

* Create the following resource groups: **This script is available in /source/OSSonAzure/createAzureDemoEnvironment.sh**
  + **docker-demos** 🡪 this is used for demonstration of deploying containers across linux servers
  + **docker-linux-paas-demo** 🡪 this will show how to deploy containers in the new public preview of docker on Azure Linux Paas
  + **kubernetes-demo1** 🡪 sample of a kubernetes cluster to show how the container will be deployed across servers
  + **utility** 🡪 group that holds the build Server and utility box

**az login**

**az component update -add network**

**az account set** --subscription "Microsoft Azure Internal Consumption"

**az account set** --subscription "Visual Studio Enterprise"

**az group create** –name **ossdemo-docker** --location **eastus**

**az group create** --name **ossdemo-docker-linux-paas** --location **eastus**

**az group create** --name **ossdemo-kubernetes** --location **eastus**

**az group create** --name **ossdemo-utility** --location **eastus**

**# Create Azure Storage account**

**az storage account create** --resource-group **ossdemo-utility** --location **eastus** --sku **Premium\_LRS** --name <youridenitifer>demoutility

***#example:***

*#az storage account create --resource-group ossdemo-utility --location eastus --sku Premium\_LRS --name gbbossdemoutility*

**# Create network security groups**

az network nsg create --resource-group **ossdemo-utility** --name NSG-**ossdemo-**utility --location eastus

az network nsg create --resource-group **ossdemo-docker** --name NSG-**ossdemo-**docker --location eastus

az network nsg create --resource-group **ossdemo-kubernetes** --name NSG-**ossdemo-**k8s --location eastus

# **Create security rules to allow connectivity**

az network nsg rule create --resource-group **ossdemo-utility** --nsg-name NSG-**ossdemo-**utility --name rdp-rule --access Allow --protocol Tcp --direction Inbound --priority 100 --source-address-prefix Internet --source-port-range "\*" --destination-address-prefix "\*" --destination-port-range 3389

az network nsg rule create --resource-group **ossdemo-utility** --nsg-name NSG-**ossdemo-**utility --name ssh-rule --access Allow --protocol Tcp --direction Inbound --priority 110 --source-address-prefix Internet --source-port-range "\*" --destination-address-prefix "\*" --destination-port-range 22

az network nsg rule create --resource-group **ossdemo-docker** --nsg-name NSG-**ossdemo-**docker --name ssh-rule --access Allow --protocol Tcp --direction Inbound --priority 100 --source-address-prefix Internet --source-port-range "\*" --destination-address-prefix "\*" --destination-port-range 22

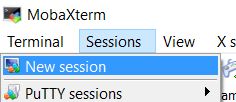
az network nsg rule create --resource-group **ossdemo-kubernetes** --nsg-name NSG-**ossdemo-**k8s --name ssh-rule --access Allow --protocol Tcp --direction Inbound --priority 100 --source-address-prefix Internet --source-port-range "\*" --destination-address-prefix "\*" --destination-port-range 22

**Create CENTOS Utility VM –** The Admin username is not critical here but will be referenced as we create additional machines and configure them via Ansible. For demo purposes it is best to keep this consistent. The SSH public key references a private key that can be retrieved by mailing [dansand@microsoft.com](mailto:dansand@microsoft.com)

az vm create -g **'ossdemo-utility'** -n **centos-utility** --public-ip-address-dns-name '**gbboss-centos-utility**' --**os-disk-name** **'centos-utility-disk**' **--image** "**OpenLogic:CentOS:7.2:latest**" **--os-type** **linux** **--nsg** 'NSG-utility' **--storage-sku** **'Premium\_LRS**' **--size** **Standard\_DS2\_v2** **--admin-username** **GBBOSSDemo** **--ssh-key-value** 'ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAz7ItfqCoqLGGbSdNT52SrZvIO2Fc26yUUyPxohN4IYxUcc1O9tmXzxHwah0jwMOw6ux+JbycOEiEpxoYPLOe9R98cKMyilnL9hGs6jCmVmRLuc/ny76euR2t8v0lhGT1yTrkLpwIlfkcaDqpufkIqQmqd20NlWbdHzsYA+s++e3jIgE5qJwO/InlMvv90nkPftR/PRYq7etWgImi00qQgX1VcD8NMZzm1qC4unzEQhYbIqYAgScCzeaj5U5NSOvDm6wgwceBCcdM8jSm7SYdetVm3J3cd+hO+SVKYgx8Zg1+kdh9RkaE2+ZRr0wtoUi/ClOXb53a4rtfYYzj85/W9w== rsa-key-20170222'

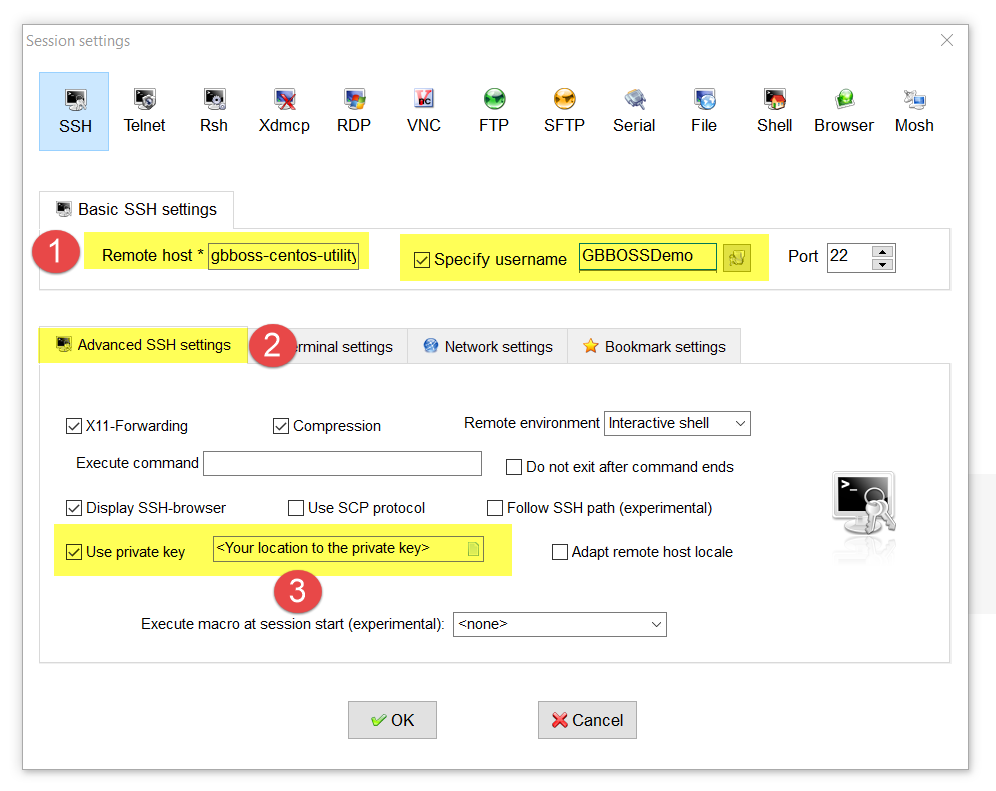
**Terminal into the new CENTOS machine –** There are several good tools for accessing the remote terminal. This demo uses MobaXTerm – available here – <http://mobaxterm.mobatek.net/download.html>

Create a new session:

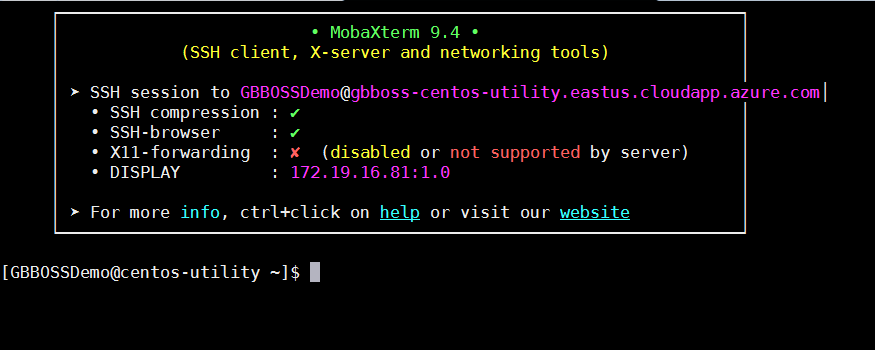


Setup the new session with the following parameters:

1. **Server name:** gbboss-centos-utility.eastus.cloudapp.azure.com **Login Name**: GBBOSSDemo (Or what you may have changed it to in the CENTOS creation script above)
2. **Location of the Private key that corresponds to the public key when the VM was created**



Upon successful login you should see:



**Set the Administrative Passwords**

**sudo passwd GBBOSSDemo**

**sudo passwd root**

**Install and Configure Ansible**

**wget** <http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-9.noarch.rpm>

**sudo rpm** -ivh epel-release-7-9.noarch.rpm

**sudo** yum -y install ansible

**#Install and configure GIT**

**sudo yum -y install git**

**sudo mkdir /source**

**cd /source**

**sudo git clone** <https://github.com/dansand71/OSSonAzure>

**sudo chown -R GBBOSSDemo /source/OSSonAzure/.**

**Run the Ansible configuration script**

Copy the private key “id\_rsa” to the ~/.ssh home directory of the GBBOSSDemo account & mark the id\_rsa as secure

**sudo chmod 600 ~/.ssh/id\_rsa**

**ansible-playbook -i /source/OSSonAzure/ansible/hosts /source/OSSonAzure/ansible/utility-server-configuration.yml -v**

**# Install Azure CLI (Cant get it to default to the right answers)**

**curl -L https://aka.ms/InstallAzureCli | bash**

# **Demo Material -**

Switch over to Demo 1

* Todo –
  + create a desktop link for the GBBOSSDemo user for VS Code
  + create app insights & update the tags and code as well as docker VM Images
  + access denied when creating k8s client
    - az acs kubernetes install-cli
  + create OMS workspace and setup discovery process

**az login**

**az account set --subscription "Visual Studio Enterprise"**

**# Create two new VM’s**

**sudo chmod +x ./source/OSSonAzure/azscripts/newVM.sh**

**./source/OSSonAzure/azscripts/newVM.sh**

**# Add the new vm’s you are creating to the Ansible Hosts file**

**sudo gedit ./source/OSSonAzure/ansible/hosts**

**# Setup the new hosts with Docker**

**cd /source/OSSonAzure/ansible**

**ansible-playbook docker-setup.yml -i hosts -v**