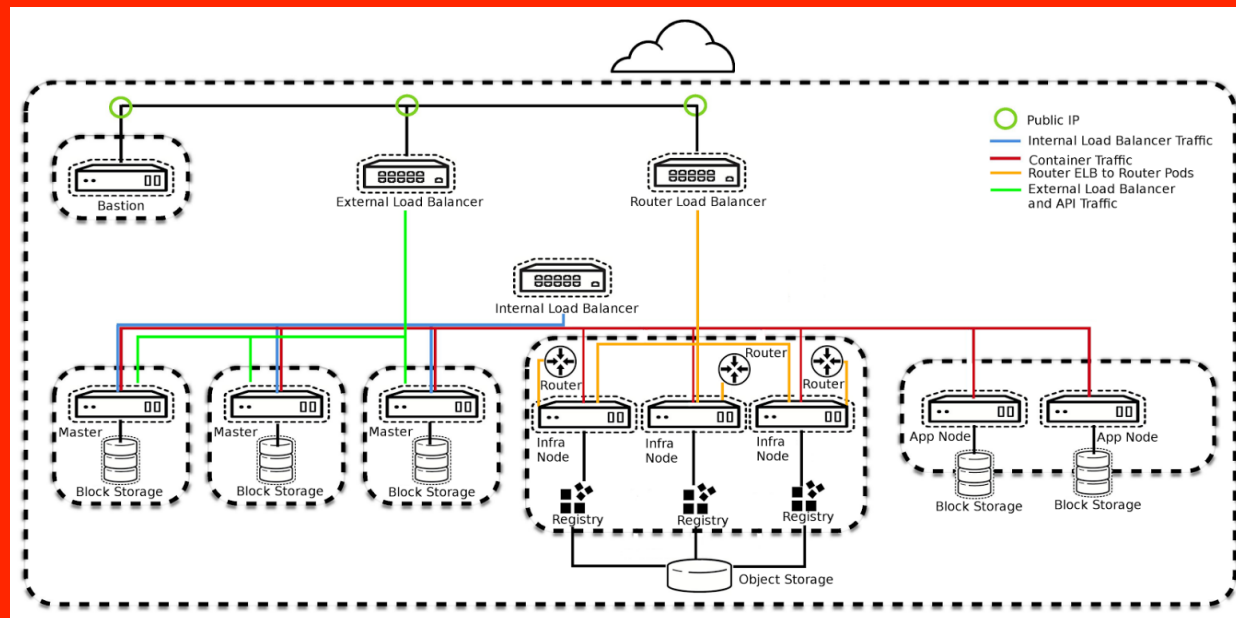# HowTo: OpenShift Hybrid Windows and Linux

Glenn West, Principle Engineer

How-To

Overview

- Azure ARM Templatre (Multi-Host)
- OVN Setup
- Windows Setup

Where To Find It

- Azure ARM Template
  - https://github.com/glennswest/hybrid-openshift-contrib/commits/master
- OVN and Windows Setup
  - https://github.com/glennswest/hybrid

Usage – 2 ways to use

- Use the hybrid-openshift-contrib
  - Provides the auto creation of a complete cluster and all the related infrasturure.
  - 3 Masters, 3 Infra, 1 or more compute nodes, and 1 or more windows nodes
  - Requires OpenShift Subscriptions/Rhel and/or Employee Subscription
  - Requires a Azure Subscription
- Hybrid Scripts
  - Self provision bare metal or Any cloud provider, provide a ansible host file, and use the ansible OVN and Windows scripts. Assume you have a bastion host.

What you get with ARM Template

- Bastion Host

- 3 Masters (3 Vm's)

- 3 Infra (3 Vm's)

- 1(or more) Linux Compute Nodes 1(or more VM's)

- 1(or more) Windows Nodes (1 or more VM's)

- OCP 3.9

- Automatic use of ovn and windows scripts (WIP)

How to Use the Arm Templates

- Install the ARM command line tools on your machine
    - https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

- Create the Parameter Files

- Login To Azure

- Create a Resource Group for the cluster

- Deploy It

```
~/azurelogin.sh
sed -i.bak '/gswmonb/d' ~/.ssh/known_hosts
azure group delete --name xgswmon
azure group create --name xgswmon --location "East US 2"
azure group deployment create \
    --name xgswmon \
    --resource-group xgswmon \
    --template-uri "https://raw.githubusercontent.com/glennswest/hybrid-openshift-contrib/master/reference-architecture/azure-ansible/3.9/azuredeploy.json" \
    -e "winactivation.parameters.json"
```

Logging Into Azure

- az login --service-principal -u <app-url> -p <password-or-cert> --tenant <tenant>

- Note that you may also login to a "user" account as well.
    - Must have the ability to create and delete a resource group
    - Must have quota enough to deploy. Note the current script creates several machines, so it needs appropriate core limits
    - Must be able to create storage accounts
    - Microsoft Trial Accounts are truly no point. Don't try. A min employee lasts no time with a trial account

Parameters File

- JSON File
- Provides all the parameter for deployment
  - RHN Username/Password
  - Default user login for cluster
  - Number of nodes
  - Azure Subscription Info
  - Settings
  - SSH Keys (Note you need to do a encode on the key)
- Detailed Walkthru on Parameters:
- https://access.redhat.com/documentation/en-us/reference_architectures/2017/html-single/deploying_red_hat_openshift_container_platform_3.6_on_microsoft_azure/index#parameters_required

Example - Kickoff


```
ovpn-117-190:~ gwest$ ./activatewiny-v39.sh
info:     Executing command login
info:      Added subscription Microsoft Azure Sponsorship

info:     login command OK
info:     Executing command group delete
Delete resource group ygswmon? [y/n]  y
+ Deleting resource group ygswmon
info:     group delete command OK
info:     Executing command group create
+ Getting resource group ygswmon
+ Creating resource group ygswmon
info:     Created resource group ygswmon
data:     Id:                /subscriptions/2586c64b-38b4-4527-a140-012d49dfc02c/resourceGroups/ygswmon
data:     Name:              ygswmon
data:     Location:          eastus2
data:     Provisioning State: Succeeded
data:     Tags: null
data:
info:     group create command OK
info:     Executing command group deployment create
+ Initializing template configurations and parameters
+ Creating a deployment
info:     Created template deployment "ygswmon"
+ Waiting for deployment to complete
+
info:     Resource 'saregygswmon' of type 'Microsoft.Storage/storageAccounts' provisioning status is Running
info:     Resource 'masteravailabilityset' of type 'Microsoft.Compute/availabilitySets' provisioning status is Succeeded
info:     Resource 'infranodeavailabilityset' of type 'Microsoft.Compute/availabilitySets' provisioning status is Succeeded
info:     Resource 'sanodygswmon' of type 'Microsoft.Storage/storageAccounts' provisioning status is Running
info:     Resource 'nodeavailabilityset' of type 'Microsoft.Compute/availabilitySets' provisioning status is Succeeded
```

Example – End of ARM Deploy

```
data:    TemplateLink        : https://raw.githubusercontent.com/glennswest/hybrid-openshift-contrib/master/reference-architecture/azure-a
data:    ContentVersion      : 1.0.0.0
data:    DeploymentParameters :
data:    Name                                Type            Value
data:    ----------------------------------- --------------- ------------------------------------------
data:    adminUsername                       String          glennswest
data:    adminPassword                       SecureString    undefined
data:    sshKeyData                          SecureString    undefined
data:    wildcardZone                        String          gswweby
data:    numberOfNodes                       Int             1
data:    numberOfWinNodes                    Int             2
data:    image                               String          rhel
data:    masterVMSize                        String          Standard_DS4_v2
data:    infranodeVMSize                     String          Standard_DS4_v2
data:    nodeVMSize                          String          Standard_DS4_v2
data:    winVMSize                           String          Standard_DS4_v2
data:    rhsmUsernamePasswordOrActivationKey String          activationkey
data:    rhnUserName                         String          6616363
data:    rhnPassword                         SecureString    undefined
data:    subscriptionPoolId                  String          8a85f98160da20de0160db1525ca38a6
data:    sshPrivateData                      SecureString    undefined
data:    aadClientId                         String          7f35aaec-4523-406d-9f8f-ed06e717fabd
data:    aadClientSecret                     SecureString    undefined
data:    openShiftSDN                        String          redhat/openshift-ovs-subnet
data:    metrics                             Bool            true
data:    logging                             Bool            true
data:    opslogging                          Bool            false
data:    Outputs             :
data:    Name                        Type      Value
data:    --------------------------- --------- --------------------------------------------------------
data:    openshift Webconsole        String    https://ygswmon.eastus2.cloudapp.azure.com:8443/console
data:    bastion ssh                 String    ssh -A ygswmonb.eastus2.cloudapp.azure.com
data:    openshift Router Public IP  String    40.70.189.100
info:    group deployment create command OK
```

Next Step (Note this is Temporary)

- Ssh to username@hostname
  - Username is the one supplied in the paramters file
  - Hostname is returned at the end of the deployment
    - Full hostname will vary by azure zone its deployed in
  - This connections you to the bastion host

Next Step (Note this is Temporary)

- Run:
    - Openshift-install.sh (As Root)
    - Git pull https://github.com/glennswest/hybrid
    - setup-clients.sh
    - As root: ansible-playbook ovn_presetup.yml
    - As root: ansible-playbook ovn_postsetup.yml
    - As root: ansible-playbook recreate-containers.yml
    - As root: recreate-console.sh
    - As root: windows.yml

Using Hybrid Scripts Independently

- OVN Setup
  - ovn_presetup/ovn_postsetup
    - Windows nodes requires ovn SDN solution. Currently its not integrated into ocp, The scripts builds upstream, installed it, and configures it, post openshift install. You must choose Redhat SDN disabled, and the cni plugin.
  - Ansible inventory file: Please console the hybrid_openshift_contrib bastion.sh which creates the inventory file for reference
  - Expects normal subscriptions to already be existing
  - Must be able to do yum installs

Windows Nodes

- Window Nodes are expected to be:

windows	Object	Object
publisher	String	MicrosoftWindowsServer
offer	String	WindowsServerSemiAnnual
sku	String	Datacenter-Core-1709-with-Containers-smalldisk
version	String	latest

- Notes this specification on Azure allows for the latest image to be used at time of install.

Installation using windows.yml

- Uses Ansible on Windows

- Must have a windows group in ansible inventory

- If using script separately, must do some setup before:

```
echo "Setup for windows nodes"
yum -y install --enablerepo="epel" python-devel krb5-devel krb5-libs krb5-workstation python-kerberos python-setuptools
yum -y install --enablerepo="epel" python-pip
pip install "pywinrm>=0.2.2"
pip install pywinrm[kerberos]
yum install -y python-dns
```

Setting up Windows group_vars

- Must create a group_vars to setup windows nodes
- The following is done automatically in bastion.sh in the ARM template

```
echo "Setup group_vars for windows machines"
mkdir /home/${AUSERNAME}/group_vars
cat <<EOF > /home/${AUSERNAME}/group_vars/windows
ansible_user: ${AUSERNAME}
ansible_password: ${PASSWORD}
ansible_port: 5986
ansible_connection: winrm
# The following is necessary for Python 2.7.9+ (or any older Python that has backported SSLContext, eg, Python 2.7.5 on RHEL7) when using default WinRM self-signed
ansible_winrm_server_cert_validation: ignore
EOF
```

Hybrid Script – Windows Node Setup

Current Issues and Problems

- OVN Network Stability Improvement – No Proxy option needs to be implemented
- Automatic Run of ovn/windows scripts during deployment
- More testing with windows
- See epic on trello