**## VM\_Deploy Task:**

1. **Deploy operation faster notification VMs**
2. **Decommission operation realtime VMs**
3. **RHUI VM\_Disk expansion**
4. **Infra test for expansion of virtual server**
5. **Restructure for after changed from active to coldstandby**
6. **Restructure and create New VMs for Coldstandby**
7. **Backyard deploy server – 777**
8. **Deploy Baremetal**

# Declare Variable

REGION =

ENV =

**## Move to Git directory**

# cd git\_ansible

# git pull

# git submodule update –init –recursive

# git checkout master

# git branch

# Declare target hypervisor

Ex: Abvd-oo1hv

# Declare target vm

Ex:

Target\_VM=(

#bear-linux-kickstart-conductor

${ENV}nv-adwi1234a

echo ${TARGET\_VM[@]}

## Enable Filter:

## Notified operation team by email.

**##** Confirm there is no alert

## Before Check:

## check status

grep\_opt=""

for vm in ${TARGET\_VM[@]}; do

 grep\_opt="${grep\_opt} -e ${vm}"

done

for host in ${HV\_GROUP[@]}; do

 ssh -o GSSAPIAuthentication=no -o StrictHostKeyChecking=no -o KbdInteractiveAuthentication=no -o PasswordAuthentication=no -o ConnectTimeout=120 $host sudo bash -s <<EOF

(

virsh -q list --all | grep${grep\_opt}

) | while read line; do

echo "\$HOSTNAME: \$line"

done

EOF

done

## VMs are not exist

## Update pxeserver target definition

#pyenv activate venv

# ansible-playbook –I hosts/${Region} site.yml –l pxeserver –skip-tags (mention skip tag name)

## Deploy VM with new definition

## rsync material

ssh ${ENV}np-adhv0024n

sudo -i

rsync -Sauv 10.X.253.39:/data/libvirt/templates/baremetal\_windows2012r2\_201510230000.qcow2 /data/libvirt/templates/

bg

disown

exit

exit

ssh ${ENV}np-adhv0025n

sudo -i

rsync -Sauv 10.X.253.39:/data/libvirt/templates/baremetal\_windows2012r2\_201510230000.qcow2 /data/libvirt/templates/

bg

disown

exit

exit

ansible-playbook -i hosts/${REGION} site.yml -t virtualmachine

## check status

grep\_opt=""

for vm in ${TARGET\_VM[@]}; do

 grep\_opt="${grep\_opt} -e ${vm}"

done

for host in ${HV\_GROUP[@]}; do

 ssh -o GSSAPIAuthentication=no -o StrictHostKeyChecking=no -o KbdInteractiveAuthentication=no -o PasswordAuthentication=no -o ConnectTimeout=120 $host sudo bash -s <<EOF

(

virsh -q list --all | grep${grep\_opt}

) | while read line; do

echo "\$HOSTNAME: \$line"

done

EOF

done

## Do OS install

# Log in target hypervisor, open console of target VM (virsh console TARGET\_VM), and then install OS using pxeboot.

# ssh Target Hypervisor

# sudo -i

# virsh list --all

# virsh console Target VM

# Enter

# \*\*\*\*\*\*\*\*\*\*\*\*\*ff MAC address Enter

# install OS via Kickstart

# Ctrl + ]

## Notify the team on completion of task.

## Enable filter

1. **Deploy Keystone VM**
2. **Re-install OS**

## 11 Update your local ssh\_config reflecting deploy of new VMs

cp ssh\_configs/${REGION} /root/.ssh/config

# cp: overwrite `/root/.ssh/config'? yes

vi /root/.ssh/config

# Change USERNAME to Line 3 and Line 51

# Change YOURKEY to id\_rsa : Line 4

# copy&paste Line 4 to line 52

# :wq

1. ## Do OS re-install
2. # Log in target hypervisor, open console of target VM (virsh console TARGET\_VM), and then install OS using pxeboot.
3. # ssh Target Hypervisor
4. # sudo -i
5. # virsh list --all
6. # virsh destroy Target VM
7. # virsh start Target VM
8. # virsh console Target VM
9. # Enter
10. # ESC+@
11. # \*\*\*\*\*\*\*\*\*\*\*\*\*ff MAC address Enter
12. # install OS via Kickstart
13. # Ctrl + ]