#!/bin/env bash

SATELLITE\_HOST="hlsatellite02.hobbylobby.corp"

LOG\_FILE="/var/log/$0.log"

SWITCHES="""$@"""

if [[ $SWITCHES = \*" -h"\* || $SWITCHES = \*" --help"\* || $SWITCHES == "-h" || $SWITCHES == "--help" ]]

then

echo -e """$0

--host <hostname>

--ip <ip\_address>

--org <satellite\_org>

--key <satellite\_activationkey>

--swap-disk <swap\_drive>

--app-disk <app\_disk>

--app-dir <app\_dir> (default to "/mnt/app01" if --app-disk specified

--ou <ad\_ou>

--sub-ou <ad\_sub\_ou>

--ad-env <ad\_env>

--reboot

--update"""

exit 1

fi

while [ "$SWITCHES" != "" ]

do

case $1 in

--host|--name|--hostname)

NEW\_HOSTNAME=$2

shift

;;

--ip|--ip-address)

IP\_ADDRESS=$2

shift

;;

--org|--satorg|--satellite-org)

ORG=$2

shift

;;

--key|--activationkey)

ACTIVATION\_KEY=$2

shift

;;

--swap-drive|--swap|--swap-disk)

SWAP\_DISK=$2

shift

;;

--app-drive|--app|--app-disk)

APP\_DISK=$2

shift

;;

--add-dir|--app-folder)

APP\_DIR=$2

shift

;;

--ou|--ad-ou)

OU=$2

shift

;;

--sub-ou|--ad-sub-ou)

SUB\_OU=$2

shift

;;

--environment|--env)

ENV=$2

shift

;;

--reboot)

REBOOT=TRUE

;;

--update)

UPDATE=TRUE

;;

esac

shift

SWITCHES="""$@"""

done

if [ "z$NEW\_HOSTNAME" != "z" ]

then

echo "Setting Hostname to ${NEW\_HOSTNAME}"

hostnamectl set-hostname ${NEW\_HOSTNAME} > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Failed to change Hostname to ${NEW\_HOSTNAME}" >&2

exit 1

else

sed -i "s/template/${NEW\_HOSTNAME}/g" /etc/hosts

fi

fi

if [ "z${IP\_ADDRESS}" != "z" ]

then

INTERFACES=`ls -1 /sys/class/net/`

INTERFACES="""${INTERFACES/$'\n'lo/}"""

if [ `echo """${INTERFACES}""" | wc -l` -eq 1 ]

then

sed -i "s/10.100.37.98/${IP\_ADDRESS}/g" /etc/sysconfig/network-scripts/ifcfg-${INTERFACES}

sed -i "s/10.100.37.98/${IP\_ADDRESS}/g" /etc/hosts

else

echo "Unable to determine network interface"

fi

fi

if [ "$ORG" != "" ]

then

echo "Configuring Subscription-manager to attach to ${SATELLITE\_HOST}"

yum list katello-ca-consumer-hlsatellite02.hobbylobby.corp -q >$LOG\_FILE 2>&1

if [ $? -eq 1 ]

then

echo "Installing Katello CA"

# Get and Install Katello CA from Satellite server

curl -O http://${SATELLITE\_HOST}/pub/katello-ca-consumer-latest.noarch.rpm > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Unable to retireve Katello CA RPM" >&2

exit 1

fi

rpm -ivh katello-ca-consumer-latest.noarch.rpm > $LOG\_FILE 2>&1

rm -f katello-ca-consumer-latest.noarch.rpm > $LOG\_FILE 2>&1

fi

# Register with satellite server

echo "Registering server with satellite"

subscription-manager register --org="${ORG}" --activationkey="${ACTIVATION\_KEY}" --force > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

subscription-manager attach --auto > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Failed to register with Satellite server" >&2

exit 1

fi

fi

echo "Enabling standard repos"

subscription-manager repos \

--enable rhel-7-server-optional-rpms \

--enable rhel-7-server-rh-common-rpms \

--enable rhel-7-server-rpms \

--enable rhel-7-server-extras-rpms \

--enable rhel-7-server-supplementary-rpms >$LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Failed to enable satellite repos" >&2

fi

echo "Add satellite ssh public key"

# Get Satellite SSH Public key

if [ ! -d "~/.ssh" ]

then

mkdir -p ~/.ssh

fi

curl -s https://${SATELLITE\_HOST}:9090/ssh/pubkey --insecure >> ~/.ssh/authorized\_keys

if [ $? -ne 0 ]

then

echo "Failed to get SSH Key from satellite server" >&2

else

chmod 700 ~/.ssh

chmod 600 ~/.ssh/authorized\_keys

fi

# Install Katello agent, upgrade packages, and install puppet agent

echo "Installing Katello agent"

yum install katello-agent -y > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Failed to install Katello agent" >&2

fi

fi

# Create swap partition and format if swap disk specified in cli switches

if [ -n "${SWAP\_DISK}" ]

then

if [ ! -b ${SWAP\_DISK}[0-9]\* ]

then

echo "Formating ${SWAP\_DISK} as swap"

mkswap ${SWAP\_DISK}

swapon ${SWAP\_DISK} >$LOG\_FILE 2>&1

UUID=`blkid ${SWAP\_DISK} | awk '{print $2}'`

if [[ "$(</etc/fstab)" != \*${UUID}\* ]]

then

echo -e "${UUID}\tswap\tswap\tdefaults\t0\t0" >> /etc/fstab

fi

else

echo "${SWAP\_DISK} already has a partition skipping"

fi

mount -a

if [ $? -ne 0 ]

then

echo "Failed to mount swap"

exit 1

fi

fi

# Create app partition and format to xfs if app disk specified in cli switches

if [ -n "${APP\_DISK}" ]

then

if [ ! -b ${APP\_DISK}1 ]

then

echo "Formatting ${APP\_DISK} as xfs"

# Setup LVM and format lv

pvcreate ${APP\_DISK} > $LOG\_FILE 2>&1

vgcreate app\_vol\_grp ${APP\_DISK} > $LOG\_FILE 2>&1

lvcreate -n lv\_data1 app\_vol\_grp -l 100%FREE > $LOG\_FILE 2>&1

mkfs.xfs /dev/app\_vol\_grp/lv\_data1 > $LOG\_FILE 2>&1

# Create App directory and add mount entry to fstab

APP\_DIR=${APP\_DIR:-"/mnt/app01"}

if [ ! -d ${APP\_DIR} ]

then

mkdir -p ${APP\_DIR}

fi

if [[ "$(</etc/fstab)" != \*"/dev/mapper/app\_vol\_grp-lv\_data1"\* ]]

then

echo -e "/dev/mapper/app\_vol\_grp-lv\_data1\t${APP\_DIR}\txfs\tdefaults\t0\t0" >> /etc/fstab

fi

else

echo "${APP\_DISK} already has a partition skipping"

fi

mount -a

if [ $? -ne 0 ]

then

echo "Failed to mount application volume"

exit 1

fi

fi

if [ "$UPDATE" == "TRUE" ]

then

# Update all packages

echo "Updating packages"

yum update -y > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Error updating packages" >&2

fi

fi

# Install open-vm-tools if VMWare virtual machine

if [ "$(</sys/class/dmi/id/product\_name)" == "VMware Virtual Platform" ]

then

echo "Installing open-vm-tools"

yum install open-vm-tools -y > $LOG\_FILE 2>&1

fi

if [ "$OU" != "" ]

then

# Install and configure SSSD

echo "Setting up SSSD"

yum install sssd realmd oddjob oddjob-mkhomedir samba-common-tools nfs-utils cifs-utils -y > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Failed to install required packages" >&2

exit 1

fi

systemctl is-active realmd > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

systemctl restart dbus realmd >$LOG\_FILE 2>&1

fi

realm discover HobbyLobby.corp > $LOG\_FILE 2>&1

if [ $? -ne 0 ]

then

echo "Failed to discover domain" >&2

exit 1

fi

until [ "${OU}" == "AD Management" -o "${OU}" == "Development" -o "${OU}" == "Operations" -o "${OU}" == "Security" -o "${OU}" == "Apps" ]

do

read -p """Active Directory OU [AD Management, Apps, Development, Operations, Security]: """ OU

done

if [ "${OU}" == "Apps" ]

then

until [ "${SUB\_OU}" == "Customer" -o "${SUB\_OU}" == "Store" -o "${SUB\_OU}" == "Web" ]

do

read -p """Apps sub OU [Customer, Store, Web]: """ SUB\_OU

done

until [ "${ENV}" == "Dev" -o "${ENV}" == "Perf" -o "${ENV}" == "Prod" -o "${ENV}" == "QA" ]

do

read -p """Enviroment [Dev, QA, Perf, Prod]: """ ENV

done

fi

# Join domain

read -p "Username: " AD\_USER

if [ -z "${SUB\_OU}" ]

then

realm join hobbylobby.corp -U ${AD\_USER} --computer-ou "OU=${OU},OU=Tier 1 RHEL,DC=HobbyLobby,DC=corp"

else

realm join hobbylobby.corp -U ${AD\_USER} --computer-ou "OU=${ENV},OU=${SUB\_OU},OU=${OU},OU=Tier 1 RHEL,DC=HobbyLobby,DC=corp"

fi

if [ $? -ne 0 ]

then

echo "Failed to join domain" >&2

exit 1

fi

if [[ "$(</etc/sssd/sssd.conf)" != \*default\_domain\* ]]

then

sed -i "s/\[sssd\]/[sssd]\ndefault\_domain\_suffix = HobbyLobby.corp/" /etc/sssd/sssd.conf

fi

if [[ "$(</etc/sssd/sssd.conf)" != \*ad\_hostname\* ]]

then

sed -i "s/\[domain\/HobbyLobby.corp\]/[domain\/HobbyLobby.corp]\nad\_hostname = `hostname -f`/" /etc/sssd/sssd.conf

fi

if [[ "$(</etc/sssd/sssd.conf)" != \*dyndns\* ]]

then

sed -i "s/\[domain\/HobbyLobby.corp\]/[domain\/HobbyLobby.corp]\ndyndns\_update = true\ndyndns\_refresh\_interval = 43200\ndyndns\_update\_ptr = true\ndyndns\_ttl = 3600/" /etc/sssd/sssd.conf

fi

if [[ "$(</etc/fstab)" != \*\/home\* ]]

then

echo -e """hlisi01nfs:/NFS\_Users\t/home\tnfs\tnfsvers=3\t0\t0""" >> /etc/fstab

mount -a

if [ $? -ne 0 ]

then

echo "Failed to mount nfs home directory" >&2

exit 1

fi

fi

if [ -z "$(grep -r '^[^\#].\*sg\_rhel\_admins' /etc/sudoers\*)" ]

then

echo -e "%sg\_rhel\_admins@HobbyLobby.corp\tALL=(ALL)\tALL" > /etc/sudoers.d/sg\_rhel\_admins

fi

fi

echo "System configuration complete"