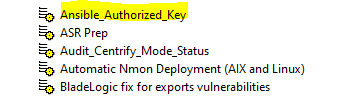
|  |
| --- |
| HCSC |
| On-Premises Linux VM OS Upgrade – RHEL 7 to RHEL 8 |
| **STANDARD OPERATING PROCEDURE** |

|  |
| --- |
| * **Paperwork** * Open the share point link to download ‘Linux RHEL Upgrades for HCL to process’ spreadsheet:   [**https://myfyi.sharepoint.com/:x:/r/teams/DistrComp/\_layouts/15/Doc.aspx?sourcedoc=%7B2B6CE346-A3A1-44D0-874C-8BCC7E9B3593%7D&file=Linux%20RHEL%20Upgrades%20for%20HCL%20to%20process.xlsx&action=default&mobileredirect=true**](https://myfyi.sharepoint.com/:x:/r/teams/DistrComp/_layouts/15/Doc.aspx?sourcedoc=%7B2B6CE346-A3A1-44D0-874C-8BCC7E9B3593%7D&file=Linux%20RHEL%20Upgrades%20for%20HCL%20to%20process.xlsx&action=default&mobileredirect=true)     * All RHEL 7 server approved for OS Upgrade is placed in sheet with tab named ‘HCL Master - To Be Upgraded’. |
| 1. In vSphere on the summary page of the VM, verify that Compatibility      1. If the Compatibility version is lower that ESXi 7.0 U2 and later (VM version 19), then power off the VM and upgrade it to ESXi 7.0 U2 and later (VM version 19). 2. Upgrade VM Compatibility option is greyed out while the VM is in running state.      1. Power Off VM to Upgrade the VM Compatibility.          1. Once the Compatibility Version is upgraded to ESXi 7.0 U2 and later (VM version 19), power on the VM. |

**USING LEAPP UPGRADE ANSIBLE AUTOMATION FROM EXECUTION NODE :**

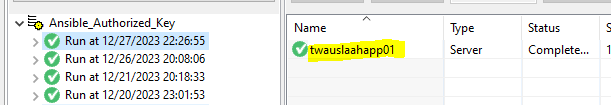
1. Run the Ansible\_Authorized\_Key job in Blade Logic to ensure the ansible ssh key from pwauslifapp01 is in the authorized\_keys file.



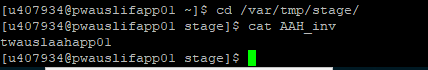
1. Location of this job in Bladelogic Server.

[/BladeLogic/Jobs/ECS OpenSystems/Linux/Ansible\_Authorized\_Key]

1. Once job completed check if its successful or not. On BL GUI tool used here for password less authentication



1. Now login on pwauslifapp01.app.hcscint.net with your LAN ID.
2. Create an inventory file with your list of target servers (using shortname) in the /var/tmp/stage directory.



1. Execute the script /usr/local/ecsos/bin/leapp\_upgrade.bash as the anssvc user using sudo:

Cat /usr/local/ecsos/bin/leapp\_upgrade.bash --- leap.upgrade.sh ka script on ansible server yah hai ..

[root@pwauslifapp01 ~]# cat /usr/local/ecsos/bin/leapp\_upgrade.bash

#!/bin/bash

export ANSIBLE\_VAULT\_PASSWORD\_FILE=/home/ansible/ansible-automation/Leapp-Upgrade/.vault\_pass.txt

if [[ -z "$1" ]]

then

echo "Usage: leapp\_upgrade.bash /var/tmp/stage/[inventory\_file\_name]"

exit 1

fi

if [ ! -f /var/tmp/stage/$1 ]

then

echo "Inventory file $1 does not exist in /var/tmp/stage"

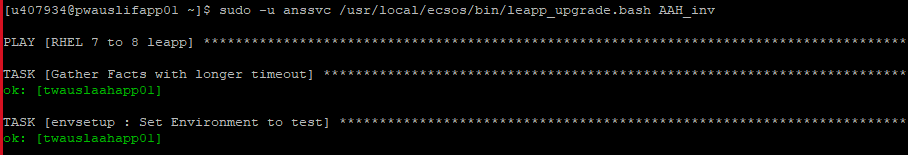
exit 1

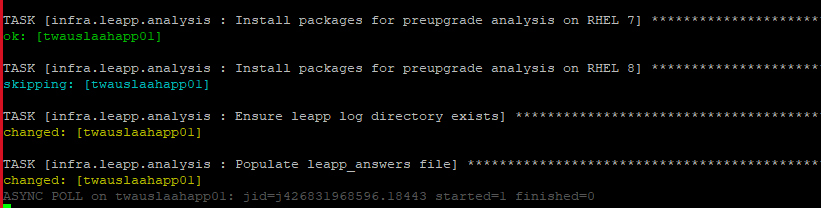
fi

cd /home/ansible/ansible-automation/Leapp-Upgrade; ansible-navigator run leapp\_upgrade\_8.yml --inventory /var/tmp/stage/$1 -e 'ansible\_user=ansible' | tee /var/tmp/leapp\_out\_$$

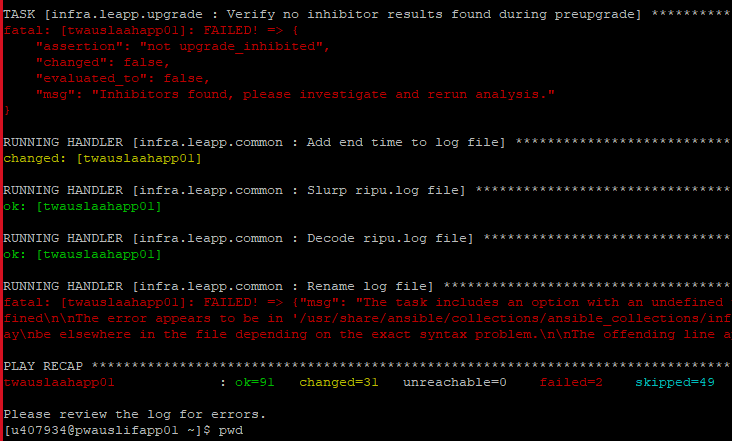
[root@pwauslifapp01 ~]#

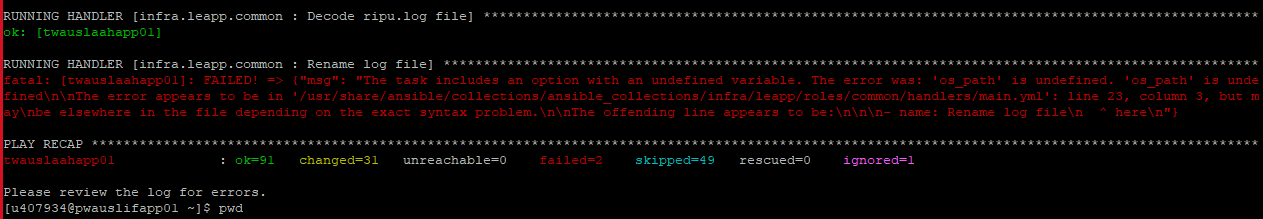
1. sudo -u anssvc /usr/local/ecsos/bin/leapp\_upgrade.bash inventory\_file



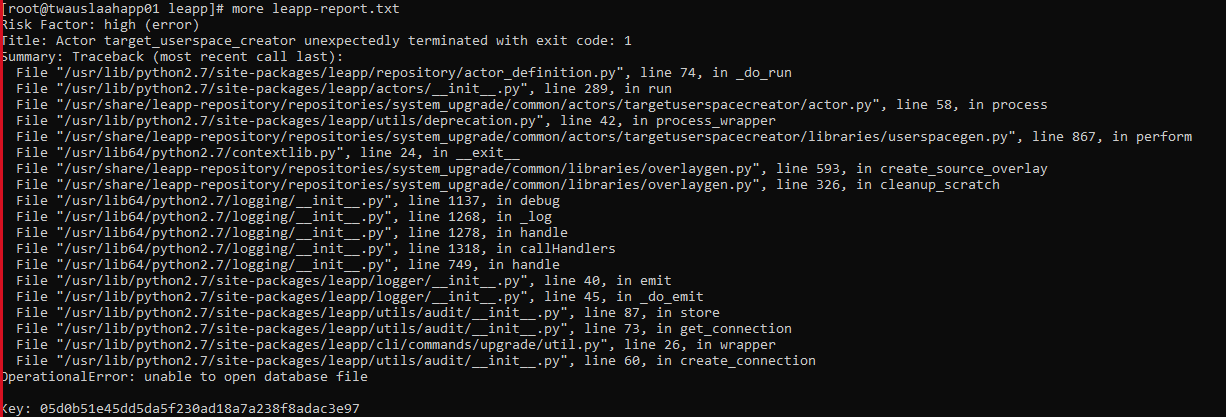


1. If there is any inhibitor which will not remediated the script will through an error like below.

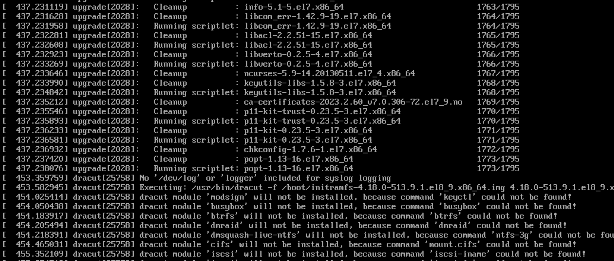


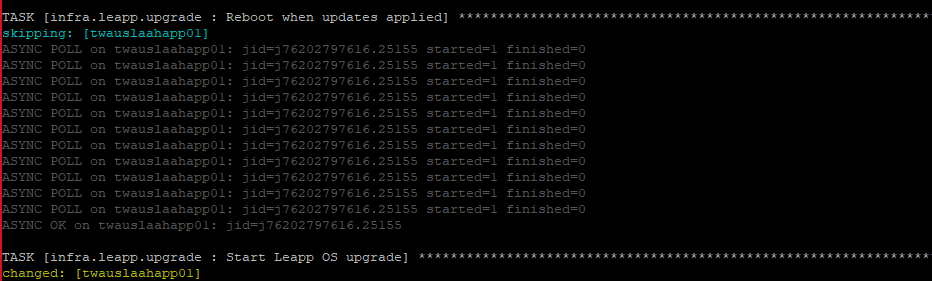


1. Check leap log files in target server /var/log/leap/leap-report.txt.

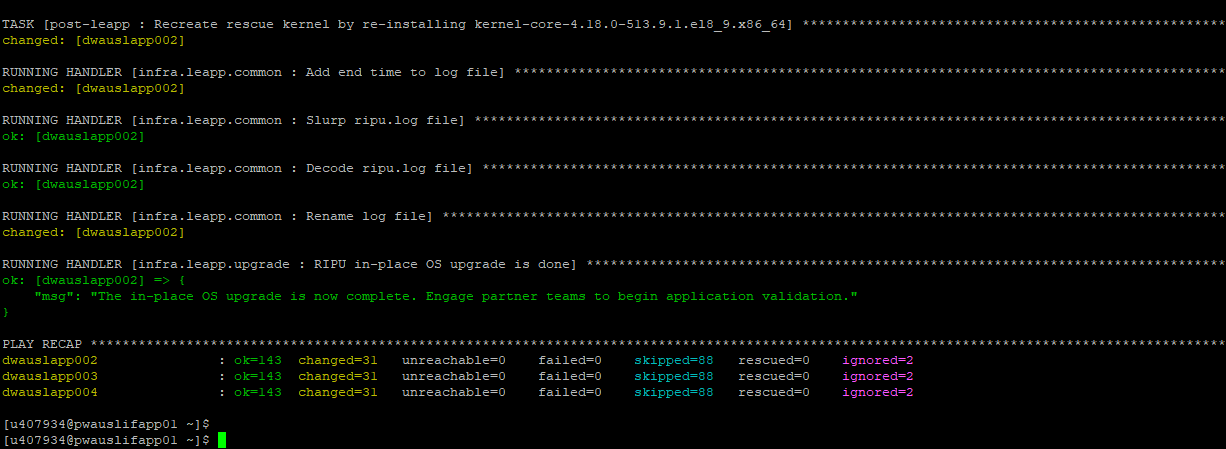


1. Once all inhibitors fixed the migration/upgrade will start and it is visible in console.

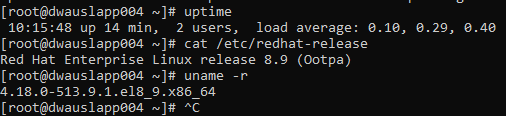


/

1. Once upgrade/migration completed the script will show completed status.



1. We can login on the Server and check the status of upgrade.



=================================on ansible server par full sript jo hcsc mai bana hai -----===

[root@pwauslifapp01 Leapp-Upgrade]# cat /usr/local/ecsos/bin/leapp\_upgrade.bash

#!/bin/bash

export ANSIBLE\_VAULT\_PASSWORD\_FILE=/home/ansible/ansible-automation/Leapp-Upgrade/.vault\_pass.txt

if [[ -z "$1" ]]

then

echo "Usage: leapp\_upgrade.bash /var/tmp/stage/[inventory\_file\_name]"

exit 1

fi

if [ ! -f /var/tmp/stage/$1 ]

then

echo "Inventory file $1 does not exist in /var/tmp/stage"

exit 1

fi

cd /home/ansible/ansible-automation/Leapp-Upgrade; ansible-navigator run leapp\_upgrade\_8.yml --inventory /var/tmp/stage/$1 -e 'ansible\_user=ansible' | tee /var/tmp/leapp\_out\_$$

[root@pwauslifapp01 Leapp-Upgrade]# cat /home/ansible/ansible-automation/Leapp-Upgrade/.vault\_pass.txt

!XXL92mm

[root@pwauslifapp01 Leapp-Upgrade]#

[root@pwauslifapp01 Leapp-Upgrade]# pwd

/home/ansible/ansible-automation/Leapp-Upgrade

[root@pwauslifapp01 Leapp-Upgrade]# ls -rlth

total 148K

-rw-rw-r-- 1 anssvc anssvc 15 Feb 14 15:48 README.md

-rw-rw-r-- 1 anssvc anssvc 2.2K Feb 14 15:48 fstab\_remove\_duplines.yml

-rw-rw-r-- 1 anssvc anssvc 456 Feb 14 15:48 env\_dict\_demo.yml

-rw-rw-r-- 1 anssvc anssvc 201 Feb 14 15:48 disable\_yum\_repos.yml

-rw-rw-r-- 1 anssvc anssvc 2.3K Feb 14 15:48 check-for-old-packages.yml

-rw-rw-r-- 1 anssvc anssvc 227 Feb 14 15:48 ansible-navigator.yml

drwxrwxr-x 3 anssvc anssvc 19 Feb 14 15:48 leapp-upgrade

-rw-rw-r-- 1 anssvc anssvc 3.4K Feb 14 15:48 leapp-post-upgrade.yml

-rw-rw-r-- 1 anssvc anssvc 379 Feb 14 15:48 leapp-postupgrade-only.yml

-rw-rw-r-- 1 anssvc anssvc 136 Feb 14 15:48 hello\_world.yml

drwxrwxr-x 2 anssvc anssvc 33 Feb 14 15:48 hcsc-powerpath

-rw-rw-r-- 1 anssvc anssvc 314 Feb 14 15:48 regex-test.yml

-rw-rw-r-- 1 anssvc anssvc 332 Feb 14 15:48 recovery.yml

-rw-rw-r-- 1 anssvc anssvc 138 Feb 14 15:48 printfacts.yml

-rw-rw-r-- 1 anssvc anssvc 268 Feb 14 15:48 nics\_rename\_only.yml

-rwxrwxr-x 1 anssvc anssvc 275 Feb 14 15:48 lint.bash

-rw-rw-r-- 1 anssvc anssvc 61 Feb 14 15:48 leapp\_vars.yml

-rw-rw-r-- 1 anssvc anssvc 5.1K Feb 14 15:48 leapp\_upgrade\_9.yml

-rw-rw-r-- 1 anssvc anssvc 5.4K Feb 14 15:48 **leapp\_upgrade\_8.yml**

-rw-rw-r-- 1 anssvc anssvc 5.7K Feb 14 15:48 leapp\_upgrade\_8\_mulesoft.yml

-rw-rw-r-- 1 anssvc anssvc 5.7K Feb 14 15:48 leapp\_upgrade\_8\_mq.yml

-rw-rw-r-- 1 anssvc anssvc 5.7K Feb 14 15:48 leapp\_upgrade\_8\_middleware.yml

-rw-rw-r-- 1 anssvc anssvc 322 Feb 14 15:48 leapp\_powerpath.yml

-rw-rw-r-- 1 anssvc anssvc 408 Feb 14 15:48 vminfo.yml

-rw-rw-r-- 1 anssvc anssvc 3.7K Feb 14 15:48 vminfo\_debug.yml

-rw-rw-r-- 1 anssvc anssvc 172 Feb 14 15:48 test\_ulimit\_files\_chk.yml

-rw-rw-r-- 1 anssvc anssvc 304 Feb 14 15:48 test\_ssh.yml

-rw-rw-r-- 1 anssvc anssvc 173 Feb 14 15:48 test\_post\_ulimit\_files\_chk.yml

-rw-rw-r-- 1 anssvc anssvc 419 Feb 14 15:48 test\_play.yml

-rw-rw-r-- 1 anssvc anssvc 268 Feb 14 15:48 test\_middleware\_start.yml

-rw-rw-r-- 1 anssvc anssvc 263 Feb 14 15:48 test\_fstab.yml

drwxrwxr-x 27 anssvc anssvc 4.0K Feb 14 15:48 roles

-rw-r--r-- 1 anssvc anssvc 1.8K Feb 14 15:49 ansible.cfg

-rw-r--r-- 1 anssvc anssvc 1.4K Feb 14 16:18 ansible-navigator.log

[root@pwauslifapp01 Leapp-Upgrade]#

[root@pwauslifapp01 Leapp-Upgrade]# cat leapp\_upgrade\_8.yml

---

- name: RHEL 7 to 8 leapp

hosts: all

strategy: free

gather\_facts: false

become: true

force\_handlers: true

environment:

http\_proxy: ""

https\_proxy: ""

HTTPS\_PROXY: ""

HTTP\_PROXY: ""

vars\_files:

- leapp\_vars.yml

tasks:

- name: Force Reboot

ansible.builtin.reboot:

- name: Wait for server to come up

ansible.builtin.wait\_for\_connection:

timeout: 1200

- name: Import Pre Ulimits role

ansible.builtin.import\_role:

name: pre\_ulimit\_files\_chk

tags:

- pre

- capsule

- name: Gather Facts with longer timeout

ansible.builtin.setup:

gather\_timeout: 720

tags:

- pre

- capsule

- name: Import envsetup role - setup variable based on target host env

ansible.builtin.import\_role:

name: envsetup

tags:

- pre

- capsule

# - name: check vars

# ansible.builtin.debug:

# var: hcsc\_runtime[hcsc\_env\_key]

# - name: fail here

# ansible.builtin.fail:

# msg: "debugging"

- name: Import VMWare role

ansible.builtin.import\_role:

name: pre-leapp-vmware

when: ansible\_facts.virtualization\_type == "VMware"

tags:

- pre

- capsule

- name: call check disk space role

ansible.builtin.include\_role:

name: chkdskspc

loop: "{{ chkdskspc | dict2items }}"

loop\_control:

loop\_var: chkdirs

tags:

- pre

- name: print chkdskspc dict

ansible.builtin.debug:

var: chkdskspc\_outdict

tags:

- pre

- name: call resize volume role

ansible.builtin.include\_role:

name: resizelvmfs

loop: "{{ chkdskspc\_outdict | dict2items }}"

loop\_control:

loop\_var: chkdskspc\_out

tags:

- pre

- name: Import pre-leapp steps

ansible.builtin.import\_role:

name: pre-leapp

tags:

- pre

- capsule

- name: Import Pre Leapp NFS

ansible.builtin.import\_role:

name: pre\_leapp\_nfs

tags:

- pre

- capsule

- name: Import Pre Leapp CIFS

ansible.builtin.import\_role:

name: pre\_leapp\_cifs

tags:

- pre

- capsule

- name: Import Pre Leapp automount

ansible.builtin.import\_role:

name: pre\_leapp\_automount

tags:

- pre

- capsule

- name: Import Pre Leapp powerpath

ansible.builtin.import\_role:

name: pre\_leapp\_powerpath

tags:

- pre

- capsule

- name: Recursively remove directory

ansible.builtin.file:

path: /var/log/ripu/ripu.log

state: absent

tags:

- pre

- capsule

- name: Import analysis-leapp steps - infra-leapp

ansible.builtin.import\_role:

name: infra.leapp.analysis

vars:

satellite\_organization: "{{ hcsc\_runtime[hcsc\_env\_key].satellite\_organization }}"

satellite\_activation\_key\_pre\_leapp: "{{ hcsc\_runtime[hcsc\_env\_key].satellite\_activation\_key\_pre\_leapp }}"

satellite\_activation\_key\_leapp: "{{ hcsc\_runtime[hcsc\_env\_key].satellite\_activation\_key\_leapp }}"

leapp\_preupg\_opts: '--target 8.8'

crypto\_policy: DEFAULT

leapp\_answerfile: |

[remove\_pam\_pkcs11\_module\_check]

confirm = True

tags:

- analysis

- name: RHEL 7 to 8 leapp upgrade

hosts: all

strategy: free

gather\_facts: false

become: true

force\_handlers: true

environment:

http\_proxy: ""

https\_proxy: ""

HTTPS\_PROXY: ""

HTTP\_PROXY: ""

tasks:

- name: Gather Facts with longer timeout

ansible.builtin.setup:

gather\_timeout: 720

tags:

- replay

- post

- fstab

- finish

- name: Include envsetup role - setup variable based on target host env

ansible.builtin.import\_role:

name: envsetup

tags:

- upgrade

- replay

- post

- fstab

- finish

- name: Import upgrade-leapp steps - infra-leapp

ansible.builtin.import\_role:

name: infra.leapp.upgrade

vars:

satellite\_organization: "{{ hcsc\_runtime[hcsc\_env\_key].satellite\_organization }}"

satellite\_activation\_key\_post\_leapp: "{{ hcsc\_runtime[hcsc\_env\_key].satellite\_activation\_key\_post\_leapp }}"

satellite\_activation\_key\_leapp: "{{ hcsc\_runtime[hcsc\_env\_key].satellite\_activation\_key\_leapp }}"

leapp\_preupg\_opts: '--target 8.8'

#update\_grub\_to\_grub\_2: true

selinux\_mode: disabled

crypto\_policy: DEFAULT

tags:

- upgrade

- replay

- name: Import Post Leapp powerpath

ansible.builtin.import\_role:

name: post\_leapp\_powerpath

tags:

- post

- replay

- name: Import Post Leapp Fstab

ansible.builtin.import\_role:

name: post\_leapp\_fstab

tags:

- post

- replay

- fstab

- name: Import post upgrade tasks

ansible.builtin.import\_role:

name: post-leapp

tags:

- post

- replay

- finish

- name: Import Post Ulimits role

ansible.builtin.import\_role:

name: post\_ulimit\_files\_chk

tags:

- post

- replay

- finish

- name: Force Reboot

ansible.builtin.reboot:

- name: Wait for server to come up

ansible.builtin.wait\_for\_connection:

timeout: 1200

...

[root@pwauslifapp01 Leapp-Upgrade]#

On BL GUI .. authorized key script as below for passwordless authotication

Paste this cmd on BL on GUI for passwordless authenciton to run upgrade job

#!/bin/nsh

#

# determine OS and env

#

DEST="`echo ${1} | sed -e 's/\..\*$//'`"

FQDN=`host ${DEST} | awk '{print tolower($1)}'`

. //pwausxpbrmgt01/usr/local/bl\_depot/personal/Paul/envsetup.nsh

OS=$(GetOS)

ENV=$(GetEnv)

DEST\_IP=${DESTIP}

#

# end of determine OS and env

#

#

# end of determine OS and env

#

#

# Matt Camp Request - BL Script to Setup Ansible User

#

#check for ansible user add if its not there

ANSIBLEID=`nexec $DEST "if id ansible 2&>1 /dev/null; then echo "true"; else echo "false"; fi"`

ANSSVCID=`nexec $DEST "if id anssvc 2&>1 /dev/null; then echo "true"; else echo "false"; fi"`

ANSIBLEGRP=`nexec $DEST "if grep -q ansible /etc/group; then echo "true"; else echo "false"; fi"`

echo $ANSIBLEGRP

if [ $ANSSVCID = "true" ]

then

nexec $DEST "usermod -l ansible anssvc"

fi

if [ $ANSIBLEID = "false" ]

then

if [ $ANSIBLEGRP = "true" ]

then

echo "ansible group exists"

nexec $DEST "useradd -u 444333 -g ansible -m -d /home/ansible ansible"

else

echo "creating ansible user group"

nexec $DEST "useradd -u 444333 -m -d /home/ansible ansible"

fi

nexec $DEST "echo \'\!XXL92mm\' | passwd --stdin ansible"

ANSIBLEUID=`awk -F: '$1 == "ansible" { print $3 }' //$DEST/etc/passwd`

ANSIBLEGID=`awk -F: '$1 == "ansible" { print $4 }' //$DEST/etc/passwd`

nexec $DEST "chown -R ${ANSIBLEUID}:${ANSIBLEGID} /home/ansible"

echo 'ansible ALL=(ALL) NOPASSWD: ALL' > //$DEST/etc/sudoers.d/ansible

fi

#add ansible user key

ONPREM\_ANSIBLE\_KEY='ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDomdvfW/HuvjZIpVniuPjc/kfkvplmXRZPPZscvCMqej0y2Kaws45UugHEYoJLYq5pjfr6IB///k4GghhTTZTKveQIekbFEY4dzUVTSDCZwzpT3IsAkXMSqdPMzAPnxEpQ4YPK6dqW9JzFFbe4MkiRalronUUUKZrkvvCgsgTG8aYqkxrIcnvmvEti3mikv56tcstAdN94T1FcuqfDQS/Uwri87bf4dCxIEV1lm3wJOCIGN9X/Pjl6UOFdo0S404Xzjbup1YlM5ZKwu9lyfcyG/fv4Hu6VpUZyP1kOtmQ+E/Ye6zW2B48OX6QEKXDNNwhuK/4uHBMYkJiaPeEY2eOt anssvc@twauslpdans0009'

#RSA\_KEY='ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDFp/QqYSO663qBwybizjRoKLn2VDHYDcwTLV4TVDdBJFiPcnhMoxLIfSyLuf1n5yD62/8VrYAz2eJk1U5Bu7sThHzBSZX4/9qhUm8TtH4ErU4GxgFBSdAFkZnzpwKutOhRZ49YXJ+9f1CWB6twUtQuvib0DBAO1tIRB9fLGt3VYaeIj338N7zF29sxjLu4g+y8XekSpkpsr8Kw1gkeDlB+jmF10mUJTkP0zvTHWoQQrfbmKSvIuFOJZnwE/hwy64OOcTwMKDFWQQwYQYcMbJ71hWdEINAex0GaBDiU+yQTHTlV/5gHRh3GWBl/Tj1s9yhIyAw3+aQEa/bHXgYIuCkf ansible@pwauslifbh09'

#RSA\_KEY1='ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDomdvfW/HuvjZIpVniuPjc/kfkvplmXRZPPZscvCMqej0y2Kaws45UugHEYoJLYq5pjfr6IB///k4GghhTTZTKveQIekbFEY4dzUVTSDCZwzpT3IsAkXMSqdPMzAPnxEpQ4YPK6dqW9JzFFbe4MkiRalronUUUKZrkvvCgsgTG8aYqkxrIcnvmvEti3mikv56tcstAdN94T1FcuqfDQS/Uwri87bf4dCxIEV1lm3wJOCIGN9X/Pjl6UOFdo0S404Xzjbup1YlM5ZKwu9lyfcyG/fv4Hu6VpUZyP1kOtmQ+E/Ye6zW2B48OX6QEKXDNNwhuK/4uHBMYkJiaPeEY2eOt anssvc@twauslpdans0009'

#RSA\_KEY2='ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDW9F+qopv65GaIlWsTcQQ9FssXAP1i3VaO64ucgatu9Rx7wuluHod3CMG2APGjAQ4MrJmaT2LWdiZ6Az8L6f1mwPmSpMYrFYQ1ZfpfG2xq2muAbqC3PCLeNzoVNneuTSd/+vHzARXglSnEvXsbp1HgPemWKrGTjKpVZDLRSqhUwm1CEPS+XIUTkjN3f8uv8o864RykCgqq9kGQnHfnoaJY/ObNLmr3I0pPZUdH/0Nvc+khhUH/iBqdaX4yHutvjr/cVte4zAr4Jl+3Rp8vOyVOHidoUVYT22tLEUL1HJeaeWmLe+PTVsAmJERl3ri2M0aRU/Ez1SG8IAiz7GG7efLv awx@pwausr21hans001.app.hcscint.net'

if [ ! -d //$DEST/home/ansible/.ssh ]

then

mkdir -p //$DEST/home/ansible/.ssh

chown -R 444333:1000 //$DEST/home/ansible

chmod -R 750 //$DEST/home/ansible

chmod -R 700 //$DEST/home/ansible/.ssh

else

echo "Home dir and .ssh directory already exist for ansible user on $DEST."

fi

#set perms on .ssh just in case it exists and is wrong

chmod -R 700 //$DEST/home/ansible/.ssh

#echo "key is: \"${ONPREM\_ANSIBLE\_KEY}\""

#nexec $DEST echo key is: \"${ONPREM\_ANSIBLE\_KEY}\"

#load key locally if it doesnt exist

if grep -q "$ONPREM\_ANSIBLE\_KEY" //$DEST/home/ansible/.ssh/authorized\_keys

then

echo "Key already exists in authorized\_keys on $DEST."

else

echo $ONPREM\_ANSIBLE\_KEY >> //$DEST/home/ansible/.ssh/authorized\_keys

#echo $RSA\_KEY >> //$DEST/home/ansible/.ssh/authorized\_keys

#echo $RSA\_KEY1 >> //$DEST/home/ansible/.ssh/authorized\_keys

#echo $RSA\_KEY2 >> //$DEST/home/ansible/.ssh/authorized\_keys

#echo $RSA\_KEY2 >> //$DEST/root/.ssh/authorized\_keys

fi