Automation



Ansible

```
# On testing node
python3 -m venv ~/venv/ansible
source  ~/venv/ansible/bin/activate

# Clone lab-openstack git repository
http_proxy=http://proxy.lab.nfgd.net:3128  \
https_proxy=http://proxy.lab.nfgd.net:3128 \
git clone https://github.com/infraguide-org/lab-openstack.git

# Install dependencies for Ansible
cd  lab-openstack/modules/automation/ansible
pip install --proxy=http://proxy.lab.nfgd.net:3128 -U pip
pip install --proxy=http://proxy.lab.nfgd.net:3128 -r requirements.txt

# Install dependencies via ansible-galaxy
http_proxy=http://proxy.lab.nfgd.net:3128  \
https_proxy=http://proxy.lab.nfgd.net:3128 \
ansible-galaxy collection install -r requirements.yml --force
```



Credentials

```
cat > ~/automation-openrc.sh<<"EOF"

# Clear any old environment that may conflict.
for key in $( set | awk '{FS="="}  /^OS_/ {print $1}' ); do unset $key ; done

export OS_ENDPOINT_TYPE=internalURL
export OS_REGION_NAME=RegionOne
export OS_INTERFACE=internal
export OS_AUTH_PLUGIN=password
export OS_AUTH_URL=http://int.os.lab.nfgd.net:5000
export OS_PROJECT_DOMAIN_NAME=Default
export OS_TENANT_NAME=automation
export OS_USERNAME=automation
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=automation
export OS_PASSWORD=automation
export OS_IDENTITY_API_VERSION=3
EOF
```



Credentials

```
cat > clouds.yml <<EOF
---
clouds:
  lab:
    auth:
      auth_url:            "http://int.os.lab.nfgd.net:5000/v3"
      username:            "automation"
      password:            "automation"
      project_name:        "automation"
      user_domain_name:    "Default"
      project_domain_name: "Default"
    region_name: "RegionOne"
    interface: "internal"
    identity_api_version: 3
EOF
```



Run

```
# Copy admin-openrc.sh file from deployment to testing node
scp ~/openstack/custom-config/lab/admin-openrc.sh testing.mgmt:~/

# Create automation project
source ~/admin-openrc.sh
ansible-playbook playbooks/create-project.yml

# Create test infra in automation project
source ~/automation-openrc.sh
ansible-playbook playbooks/create-infra.yml

# Destroy test infra
ansible-playbook playbooks/destroy-infra.yml
```



Terraform

```
# Install Terraform on testing node
KEYRING_PATH=/usr/share/keyrings/hashicorp-archive-keyring.gpg

curl -fsSL https://apt.releases.hashicorp.com/gpg \
  --proxy http://proxy.lab.nfgd.net:3128          \
  | sudo gpg --dearmor -o $KEYRING_PATH

echo "deb [signed-by=$KEYRING_PATH] https://apt.releases.hashicorp.com $(lsb_release -cs) main" \
  | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt-get update && sudo apt-get install terraform

cd ../terraform
terraform --version

# Download providers
http_proxy=http://proxy.lab.nfgd.net:3128  \
https_proxy=http://proxy.lab.nfgd.net:3128 \
terraform init

ls -la
cat .terraform.lock.hcl
terraform providers

# Run
TF_VAR_os_password=$OS_PASSWORD terraform plan
TF_VAR_os_password=$OS_PASSWORD terraform apply -auto-approve
```



Testinfra

```
cd ../testinfra/
py.test  --hosts=ssh://ubuntu@<SERVER_IP> server.py -v
```

