

Puppet-Conjur Integration Demo

7 commits

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

jeepapichet	Create README.md	Latest commit e58fa07 a day ago
agentnode	First commit	2 days ago
conjurcli	First commit	2 days ago
puppet-enterprise	Update PP image script	2 days ago
puppet	modify docker-compose and run scripts	2 days ago
.gitignore	First commit	2 days ago
README.md	Create README.md	a day ago
build-agentnode-image.sh	First commit	2 days ago
build-cli-image.sh	add cli build	2 days ago
build-puppetmaster-image.sh	Update PP image script	2 days ago
conjur-init.sh	First commit	2 days ago
conjur.conf	First commit	2 days ago
docker-compose.yml	added puppet fqdn alias in docker-compose	2 days ago
docker-compose.yml.bk1	First commit	2 days ago
puppetdemo-policy.yml	First commit	2 days ago
restart-demowebapp.sh	Update PP image script	2 days ago
start-demo.sh	Update PP image script	2 days ago
test-puppet-image.sh	Update PP image script	2 days ago

README.md

Puppet-Conjur Integration Demo

This is a demo integration of Puppet and Conjur using Conjur module from puppet forge. The demo includes sample webapp nodes that use hardcoded secrets vs secrets retrieved from Conjur. The demo also shows how Puppet module uses HF token to automatically bootstrap machine identity for puppet node.

Demo Requirements

- Linux host with Docker daemon and Docker Compose installed
- Conjur Enterprise 4.x image - Load the image to docker using `docker load -i conjur-appliance-4.9.3.0.tar`. Edit `docker-compose.yml` if using a different version.
- Puppet Enterprise image. - This can be downloaded from <https://puppet.com/download-puppet-enterprise>
The script was tested with `puppet-enterprise-2017.3.1-ubuntu-14.04-amd64.tar.gz`.
Edit `PE_VERSION` parameter in `build-puppetmaster-image.sh`, `puppet-enterprise/install-puppet-in-docker.sh` and `docker-compose.yml` if using a different version.

Setting Up The Demo Environment

The build process may take 10-15 minutes and require Internet connection.

```
./build-cli-image.sh
./build-agentnode-image.sh
./build-puppetmaster-image.sh
```

After finish, execute `docker images` to check the new images. There should be three new images in local docker repo.

Starting The Demo

To start the demo, execute `./start-demo.sh` This will bring up following services

- puppet - Puppet Enterprise server
- conjur - Conjur Server
- conjurcli - Conjur cli to load policy and secret
- dev-webapp
- pro-webapp

Conjur service is exposed on port 443 and puppet service is exposed on port 1443.
Web console credential is admin/Cyberark1. OS credential is root/Cyberark1.

Running The Demo

Conjur policy for the demo is already loaded. In this demo, there are sample manifest for two nodes. The dev-webapp show a sample manifest using hardcoded credential. The prod-webapp demonstrates how Conjur can be used to establish machine identity and fetch secret.

1. Review manifest file in `puppet/manifests/site.pp`.
2. SSH to dev-webapp then run `puppet agent -t` to apply configuration. The puppet simply dumps the hardcoded password in a file at `/etc/mysecretkey`.
3. Login to Conjur UI and review the puppetdemo policy. Create new hostfactory under puppetdemo/webapp layer. Copy this host factory and paste it to hostfactory parameter in `puppet/manifests/site.pp` file.
4. SSH to prod-webapp, and run `puppet agent -t` to apply configuration. Check
5. Review new host that is created in Conjur UI as well as audit activities.

The dev-webapp and prod-webapp can be restarted with `./restart-demos.sh`. This will remove and restart containers as well as purge those from Puppet Enterprise.

To restart all services, reexecute `./start-demo.sh` again.

To stop and remove all containers, `docker-compose down`

Acknowledgments

Puppet Enterprise build script is based on <https://github.com/jefferyb/puppet-enterprise-in-docker>