

Deploy ICN Bluval

How to deploy Bluval for ICN in private Jenkins instance

Integrated Cloud Native

The following instructions assume the user executes all commands as root, to facilitate development and reduce the length of these instructions. For a production deployment, adaptations should be made to reduce the need for privilege escalation, as well as evaluate security as top-priority.

Requirements:

- Access to the Internet (proxy considerations are ignored in this documentation).
- Ubuntu 18.04 as the operating system (only one tested).
- SSH is configured in all machines part of the cluster.
- Login as root.

Furthermore, this guide assumes that:

- There are a total of two machines.
- The first machine includes Jenkins, the Kubernetes master node and the first worker node.
- The second machine only includes the second worker node.

Jenkins and Bluval

Start by installing basic dependencies:

```
apt-get remove -y python-pip
apt-get install -y python python-bashate build-essential
wget https://bootstrap.pypa.io/get-pip.py
python2 get-pip.py
```

Get ICN repository:

```
cd ~
git clone "https://gerrit.akraino.org/r/icn"
cd icn/ci
```

(optional) Update Jenkins to the latest version available:

```
sed -i "s/2.192/2.241/" vars.yaml
```

Install Jenkins using Ansible playbook:

```
./install_ansible.sh
pip install -U ansible
ansible-playbook site_jenkins.yaml --extra-vars "@vars.yaml" -vvv
```

Basic Jenkins Job Builder (JJB) configuration using admin/admin credentials:

```
mkdir -p ~/.config/jenkins_jobs
cat << EOF | tee ~/.config/jenkins_jobs/jenkins_jobs.ini
[job_builder]
ignore_cache=True
keep_descriptions=False
recursive=True
retain_anchors=True
update=jobs

[jenkins]
user=admin
password=admin
url=http://localhost:8080
EOF
```

Access the web UI and add the `jenkins-ssh` credentials to communicate with Gerrit.
Quick link: http://localhost:8080/credentials/store/system/domain/_/newCredentials.

There, create a new credential of Kind **SSH Username with private key**.
Set the following fields as such:

- **Kind:** "Global (...)"
- **ID:** "jenkins-ssh"
- **Username:** USERNAME"
- **Private key:** here, paste the private key respective to the public key that has been uploaded for USERNAME at Gerrit.

Since this documentation is for ICN, ICN Jenkins devs/maintainers should contact the ICN team to get the current private key.

A second private/public keypair is needed. This one is for accessing the nodes in the Kubernetes cluster.
The private key should be placed where it can be accessed by Jenkins. Ideally a fresh keypair should be created at this point.
The following creates a new/fresh keypair for the root user:

```
ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
```

Info: this is the key that will later be assigned to `CLUSTER_SSH_KEY`, i.e `CLUSTER_SSH_KEY=/var/lib/jenkins/jenkins-rsa`

Copy the private key (`~/.ssh/id_rsa`) of the keypair just created to/as `/var/lib/jenkins/jenkins-rsa`. Additionally make Jenkins the owner of this key:

```
cp ~/.ssh/id_rsa /var/lib/jenkins/jenkins-rsa
chown jenkins:jenkins /var/lib/jenkins/jenkins-rsa
chmod 600 /var/lib/jenkins/.netrc
```

This keypair is also going to be used for accessing the Akraino and ONAP repositories as non-privileged accounts (for basic operations such as cloning a repository).

As such, the public key (`~/.ssh/id_rsa.pub`) just generated should be uploaded to the Akraino and ONAP accounts to be used.

Set Nexus login credentials in order to upload Bluval logs (replace USERNAME/PASSWORD)

```
echo "machine nexus.akraino.org login USERNAME password PASSWORD" | tee /var/lib/jenkins/.netrc
chown jenkins:jenkins /var/lib/jenkins/.netrc
```

Also needed to upload Bluval logs is the `lftools` python3 package, install it:

```
pip3 install lftools
```

The Bluval job depends on templates and scripts from the ci-management repository:

```
cd ~
git clone --recursive "https://gerrit.akraino.org/r/ci-management"
```

Let's finally get Jenkins to recognize the Bluval job (install JJB):

```
pip install jenkins-job-builder
python2 -m jenkins_jobs test ci-management/jjb:icn/ci/jjb icn-bluval-daily-master
python2 -m jenkins_jobs update ci-management/jjb:icn/ci/jjb icn-bluval-daily-master
```

Recommendation: install the Rebuilder plugin to easily rebuild a job with the same/similar parameters:
Go to: <http://localhost:8080/pluginManager/available> and install "Rebuilder".

Since Jenkins will be running a job that calls Docker, it needs to have permissions to run Docker, so add jenkins user to the docker group:

```
usermod -aG docker jenkins
```

Restart Jenkins to apply new permissions (necessary) and finalize Rebuilder plugin installation (necessary):

```
systemctl restart jenkins
```

KUD and Kubernetes

Before running any job, the ICN/EMCO flavor of Kubernetes needs to be installed. Here is the current recommended procedure.

Again, this guide assumed that:

- There are a total of two machines.
- The first machine includes Jenkins, the Kubernetes master nodes and the first worker node.
- The second machine only includes the second worker node.

The first thing to do is have master node's SSH trust itself, root@localhost.

SSH to localhost and accept the connection to persist the fingerprint in ~/.ssh/known_hosts:

```
ssh root@localhost
```

Likewise, the master node should also trust root @ the worker nodes (only 1 other worker node in this guide). SSH to them and accept the connection to persist the fingerprint in in ~/.ssh/known_hosts. This trust will be needed for Ansible to install the Kubernetes cluster (powered by KUD).

```
ssh root@WORKER_NODE_IPADDR
```

At the master node (where Jenkins is already installed at this point), download KUD:

```
cd ~
apt-get install -y git-review
git clone "https://gerrit.onap.org/r/multicloud/k8s"
cd k8s
```

Replace all localhost references with \$HOSTNAME in KUD's aio.sh:

```
sed -i 's/localhost/$HOSTNAME/' kud/hosting_providers/baremetal/aio.sh
```

Remove [ovn-central], [ovn-controller], [virtlet] and [cmk] groups (and contents) from the aio.sh file below:

```
vim kud/hosting_providers/baremetal/aio.sh
```

Configure KUD for multi-node by also modifying aio.sh:

```
vim kud/hosting_providers/baremetal/aio.sh
```

... specifically, the only change for this guide's dual-node deployment is to add the worker node details to the [all] and [kube-node] groups, as follows:

In [all], add line:

```
WORKER_NODE_HOSTNAME ansible_ssh_host=WORKER_NODE_IPADDR ansible_ssh_port=22
```

In [kube-node], add line:

```
WORKER_NODE_HOSTNAME
```

In installer.sh, disable KUD addons and plugins:

```
vim kud/hosting_providers/vagrant/installer.sh
```

... specifically, the following lines (near the end of the file) can be commented, as such:

```
# install_addons
# if ${KUD_PLUGIN_ENABLED:-false}; then
#   install_plugin
# fi
```

Add `ansible_user=root` at the end of each host line in `aio.sh` [all], it should look like this (this is required when jenkins attempts to install KUD):

```
[all]
$HOSTNAME ansible_ssh_host=${OVN_CENTRAL_IP_ADDRESS} ansible_ssh_port=22 ansible_user=root
WORKER_NODE_HOSTNAME ansible_ssh_host=WORKER_NODE_IPADDR ansible_ssh_port=22 ansible_user=root
```

(optional) Finally install Kubernetes with KUD (ansible will automatically install it in the worker node too):

```
kud/hosting_providers/baremetal/aio.sh
```

The above step is optional because the ICN Jenkins Bluval job is now capable of installing and uninstalling KUD automatically. This is done before and after running the Bluval suite, respectively. However, what's **mandatory** is copying both `aio.sh` and `installer.sh` files above into `/var/lib/jenkins/`:

```
cp kud/hosting_providers/baremetal/aio.sh /var/lib/jenkins/
cp kud/hosting_providers/vagrant/installer.sh /var/lib/jenkins/
chown jenkins:jenkins /var/lib/jenkins/aio.sh
chown jenkins:jenkins /var/lib/jenkins/installer.sh
```

Also necessary, for the time being, is copying the `/var/lib/jenkins/jenkins-rsa` private key into jenkins's own `.ssh`:

```
cd /var/lib/jenkins/.ssh
rm id_rsa*
cp ../jenkins-rsa id_rsa
chown jenkins:jenkins id_rsa
```

Remove libvirt and the virtual bridges it creates (this will be fixed in the future), as they create a security vulnerability in `os/lynis`:

```
apt-get purge -y $(apt-cache depends libvirt-bin qemu-kvm | awk '{ print $2 }' | tr '\n' ' ')
apt-get autoremove --purge -y
ip link delete dev virbr0
ip link delete dev virbr0-nic
```

(optional - Jenkins will take care of this too) A few fixes have to be applied to Kubernetes to address kube-hunter security vulnerabilities. Execute the following commands:

```
kubectl replace -f - << EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: pod-reader
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "false"
  labels:
    kubernetes.io/bootstrapping: rbac-defaults
  name: system:public-info-viewer
rules:
- nonResourceURLs:
  - /livez
  - /readyz
  - /healthz
  verbs:
  - get
EOF
```

```
kubectl replace -f - << EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: default
automountServiceAccountToken: false
EOF
```

At this point, everything is ready. Jump over to <http://localhost:8080>, log-in using admin/admin credentials and create a new build for **icn-bluval-daily-master**.

For the build, here are the **recommended parameters** to set according to the deployment herein outlined as well as to conform to upstream Bluval logging requirements:

```
CLUSTER_MASTER_IP: localhost
CLUSTER_SSH_USER: root
CLUSTER_SSH_PASSWORD: <empty>
CLUSTER_SSH_KEY: /var/lib/jenkins/jenkins-rsa
BLUEPRINT: icn
LAYER: <empty>
VERSION: master
OPTIONAL: false
PULL: true
LAB_SILO: intel
```

And pull the trigger.

Total time to run should be anywhere **from 1.5 to 3 hours** on an average server-grade dual-node with good Internet connection (~90% of the time will be spent running the k8s layer conformance testing [sonobuoy]).

The easiest way to check what logs have been uploaded to the Nexus is by loading the following URL:
https://logs.akraino.org/intel/bluval_results/icn/master/

Troubleshooting

Task download mitogen release failed

```
TASK [download mitogen release] *****
task path: /opt/kubespray-2.12.6/mitogen.yaml:17
Thursday 29 October 2020 18:51:30 +0000 (0:00:00.385) 0:00:00.491 *****
An exception occurred during task execution. To see the full traceback, use -vvv. The error was:
TypeError: load_file_common_arguments() got an unexpected keyword argument 'path'
fatal: [localhost]: FAILED! => {"changed": false, "module_stderr": "Traceback (most recent call
last):\n File \"<stdin>\", line 113, in <module>\n File \"<stdin>\", line 105, in
_ansiballz_main\n File \"<stdin>\", line 48, in invoke_module\n File \"/tmp
/ansible_get_url_payload_40xmhT/__main__.py\", line 650, in <module>\n File \"/tmp
/ansible_get_url_payload_40xmhT/__main__.py\", line 633, in main\nTypeError:
load_file_common_arguments() got an unexpected keyword argument 'path'\n", "module_stdout": "",
"msg": "MODULE FAILURE\nSee stdout/stderr for the exact error", "rc": 1}
to retry, use: --limit @/opt/kubespray-2.12.6/mitogen.retry
```

This error occurs in the get_url module of ansible. Purging the system of ansible resolved it. Note that simply uninstalling ansible is insufficient, ansible-base must be uninstalled also.

```
pip uninstall ansible-base
pip uninstall --yes ansible
```