

Deployment script for production

Prior steps

This guide is aimed to deploy the coda-site-node for the sites that are part of the coda project.

You must have received a communication by the CODA team in which the id of your hospital and the vault secret are given to your team.

A virtual machine using rocky linux 9

(https://download.rockylinux.org/pub/rocky/9/isos/x86_64/Rocky-9.2-x86_64-minimal.iso)

with the following specs must be created :

8 cores/vCPUs and 16 GB RAM

GPU: RTX 2080 or better NVIDIA-compatible GPU

Disks: OS - 100 GB, data - 512 GB

Make sure all the packages and the os is up to date by running the following command inside the virtual machine :

```
$sudo yum update -y
```

```
$sudo reboot
```

The following steps are run inside the updated Rocky linux machine

deploy-scripts

Deployment scripts for the different CODA components.

Bootstrap

All below steps must be run as «root» by using

```
$sudo su -
```

Install package dependencies

Ensure `curl` is installed:

```
$yum install -y curl
```

Export proxy

If current host doesn't have direct internet access and needs to use a proxy, simply adjust PROXY variable and export:

```
$export PROXY=http://<hostname>:<port>
```

```
$export HTTP_PROXY=${PROXY}
$export HTTPS_PROXY=${PROXY}
$export http_proxy=${PROXY}
$export https_proxy=${PROXY}
```

Also, ensure that the following URLs are whitelisted in your proxy server:

URL	Purpose
http://mirror.centos.org http://centos.mirror.iweb.ca/ http://dl.fedoraproject.org/	Installing software packages (yum)
https://github.com https://github-releases.githubusercontent.com	Pulling deployment code. Pulling various artifacts (ie: node_exporter)
https://license.aidbox.io	Validating Aidbox Licence
https://dl.min.io	Downloading Min.IO artifacts.
https://hub.docker.com https://production.cloudflare.docker.com https://registry-1.docker.io https://auth.docker.io	Pulling Docker images for Aidbox software
https://heartbeat.hub.coda19.com	Sending heartbeat to control plane (hub)

Bootstrapping

Download the bootstrap script and execute it:

```
$curl -s https://raw.githubusercontent.com/coda-platform/site-deployer-ansible/main/scripts/bootstrap.sh \
-o bootstrap.sh
```

```
$bash bootstrap.sh
```

At the end, you'll be asked a few questions necessary to complete the bootstrap process:

- Your SITE ID, provided by CODA
- Your VAULT PASSWORD, provided by CODA
- Proxy server, only if needed for bootstrapping process. (press enter to skip)

The bootstrap script instantiates a cronjob for the root user which executes a ansible-pull every 10 minutes.

```

Enter your site PROXY, if required, for internet access (format: http://<host>:<port>) []:
PLAY [BOOTSTRAP PLAYBOOK - PLAY #1 - ASK INFORMATIONS AND STORE IT IN LOCAL FACTS] *****
TASK [FILE | Ensure /etc/ansible/facts.d directory is present] *****
ok: [localhost]
TASK [INI_FILE | Add entries to /etc/ansible/facts.d/coda.fact] *****
changed: [localhost] => (item={ 'section': 'site', 'option': 'id', 'value': '114'})
changed: [localhost] => (item={ 'section': 'site', 'option': 'proxy', 'value': ''})
TASK [COPY | Create vault.pass file] *****
changed: [localhost]
PLAY [BOOTSTRAP PLAYBOOK - PLAY #2 - SETUP ANSIBLE-PULL] *****
TASK [Gathering Facts] *****
ok: [localhost]
TASK [FILE | Ensure log directory exists] *****
changed: [localhost]
TASK [COPY | Create /usr/local/bin/execute-ansible-pull.sh] *****
changed: [localhost]
TASK [CRON | Create ansible-pull cron job] *****
changed: [localhost]
TASK [FILE | Ensure /etc/coda folder exists] *****
changed: [localhost]
TASK [FILE | Add file /etc/coda/bootstrapping] *****
changed: [localhost]
PLAY RECAP *****
localhost : ok=9  changed=7  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

```

The Ansible pull that instantiate the Node runs each 10 minutes. To see the logs, if it ran correctly, run this command:

```
$cd /var/log/ansible
```

```
$ls -ll
```

```
$cat ansible-pull-YYYYMMDD-HHMMSS.log
```

```

Starting Ansible Pull at 2023-11-17 13:30:01
/opt/coda/venv-ansible/bin/ansible-pull --url https://github.com/coda-platform/site-deployer-ansible.git
[WARNING]: Could not match supplied host pattern, ignoring: cnmtlscodanim
localhost | CHANGED => {
  "after": "7bc3798ae9d60230bfd3ce1caa29cd28987c8623",
  "before": null,
  "changed": true
}
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'
[WARNING]: Could not match supplied host pattern, ignoring: cnmtlscodanim

PLAY [Deploy the current box based on local facts (coda.host.role)] *****

TASK [Gathering Facts] *****
ok: [localhost]

TASK [DEBUG | Show local facts] *****
ok: [localhost] => {
  "ansible_local": {
    "coda": {
      "site": {
        "id": "114",
        "proxy": ""
      }
    }
  }
}

TASK [SET_FACT | Set coda_* facts from local ones] *****
ok: [localhost]

TASK [INCLUDE_VARS | Node's specific vars] *****
ok: [localhost] => (item=/opt/coda/deploy-scripts-pull/playbooks/../../group_vars/114)
ok: [localhost] => (item=/opt/coda/deploy-scripts-pull/playbooks/../../vaults/vault.114)

TASK [INCLUDE_ROLE | CODA_ONEOFFS] *****

TASK [coda_oneoffs : INCLUDE_TASKS | oneoff-s116.yml] *****
included: /opt/coda/deploy-scripts-pull/roles/coda_oneoffs/tasks/oneoff-s116.yml for localhost

```

**** WHEN NEEDING TO RESTART BOOTSTRAP PROCESS****

```

$rm -rf /opt/coda/ \
$ /etc/ansible/facts.d/ \
$ /etc/profile.d/ansible.sh \
$ /etc/profile.d/env-ansible.sh \
$ /usr/local/bin/env-ansible.sh

```

Final Ouput

```
PLAY RECAP *****
localhost : ok=217  changed=19   unreachable=0    failed=0    skipped=119  rescued=0    ignored=3
Jeffrey Li (CHUM)
Finished at: 2023-11-17T15:14:03-05:00
[root@cnmtiscodanim ansible]#
```