

Zero-touch provisioning of access switch

Using CLI

1. Log on to norpan-nms.cnaas.io using ssh.
2. Start cli:

```
root@norpan-nms: ~ # ./cli.sh
```

3. List all devices (make sure you know the MAC address of the switch):

```
CNaas NMS (norpan-nms)# show devices
  id | hostname | description | management_ip | serial | vendor | model | os_version | synchronized | state
  device_type | last_seen |
-----
-----
  6 | c1 | None | 10.101.2.0 | None | None | None | None | True | MANAGED | CORE | 2020-02-21 11:28:
55.440027 |
  7 | c2 | None | 10.101.2.1 | None | None | None | None | True | MANAGED | CORE | 2020-02-21 11:29:
04.248114 |
  8 | d1 | None | 10.101.2.2 | None | None | None | None | True | MANAGED | DIST | 2020-02-21 11:29:
24.371996 |
  9 | d2 | None | 10.101.2.3 | None | None | None | None | True | MANAGED | DIST | 2020-02-21 11:29:
38.845510 |
 10 | d3 | None | 10.101.2.4 | None | None | None | None | True | MANAGED | DIST | 2020-02-21 11:29:
55.278407 |
 11 | d4 | None | 10.101.2.5 | None | None | None | None | True | MANAGED | DIST | 2020-02-21 11:38:
37.139920 |
 55 | mac-B8C253EA6871 | None | None | None | None | None | None | False | DHCP_BOOT | UNKNOWN | 2020-03-30
14:17:00.107229 |
 68 | a1 | None | 10.101.3.7 | JPE19260468 | Arista | CCS-720XP-24ZY4-F | 4.23.0F-13614065.4230F | False |
UNMANAGED | ACCESS | 2020-04-01 11:26:22.358777 |
 69 | a2 | None | 10.101.3.6 | JPE19433851 | Arista | CCS-720XP-48Y6-F | 4.22.1F-13916651.helsinki |
True | MANAGED | ACCESS | 2020-04-01 11:36:10.528371 |
 72 | a3 | None | 10.101.4.6 | JPE19313445 | Arista | CCS-720XP-24ZY4-F | 4.23.0F-13614065.4230F | True |
MANAGED | ACCESS | 2020-04-17 10:29:43.167417 |
 73 | mac-FCBD67E94A21 | None | None | None | None | None | None | False | DHCP_BOOT | UNKNOWN | 2020-04-17
10:58:42.457652 |
```

4. In this case it is MAC: FCBD67E94A21 and we can see that it's gotten id 73 from the NMS.
5. Initialize the device:

```
CNaas NMS (norpan-nms)# device_init device_type access hostname a4 id 73
Scheduled job to initialize device_id 73
Job ID: 94389
```

6. Show the job to see when it is finished:

```
CNaas NMS (norpan-nms)# show job id 94389
  id | status | start_time | finish_time | function_name | scheduled_by | exception |
-----
-----
94389 RUNNING 2020-04-17T11:08:45.970322 None init_access_device_step1 admin None
```

7. Synchronize the config to the switch:

```
CNaas NMS (norpan-nms)# sync hostname a4 dry_run false force true
Scheduled job to synchronize a4
Job ID: 94433
```

8. Check that the job is finished.
9. Done!

Using command line:

First, make sure distribution switches are added as devices. If they are not, add them. Can be done using the API:

```
curl https://localhost/api/v1.0/device -X POST -H "Content-Type: application/json" -d '{"hostname": "dist0",
"management_ip": "192.168.0.10", "platform": "junos", "state": "UNKNOWN", "device_type": "DIST"}
```

Once the distribution switches are added, we can create a management domain and assign the distribution switches to it. Of course the IPv4 range, device IDs and VLAN should be replaced with whatever values you want.

```
curl https://localhost/api/v1.0/mgmtdomains -X POST -H "Content-Type: application/json" -d '{"ipv4_gw": "192.168.0.0/24", "device_a": "cn-dist-juniper1", "device_b": "cn-dist-juniper2", "vlan": 100}'
```

Then we can power on the access switch, wait for it to boot using DHCP. List new devices that has booted using CNaas startup config:

```
curl https://localhost/api/v1.0/devices?filter[state]=DISCOVERED
```

If the device serial/MAC matches with a device you want to provision, call the API to initialize the device with a specified hostname and device type:

```
curl https://localhost/api/v1.0/device_init/20 -X POST -H "Content-Type: application/json" -d '{"hostname": "ex2300-top", "device_type": "ACCESS"}'
```

Check job status to see progress, there should be two jobs running after each other, step1 and step2.

```
curl https://localhost/api/v1.0/jobs?limit=2
```