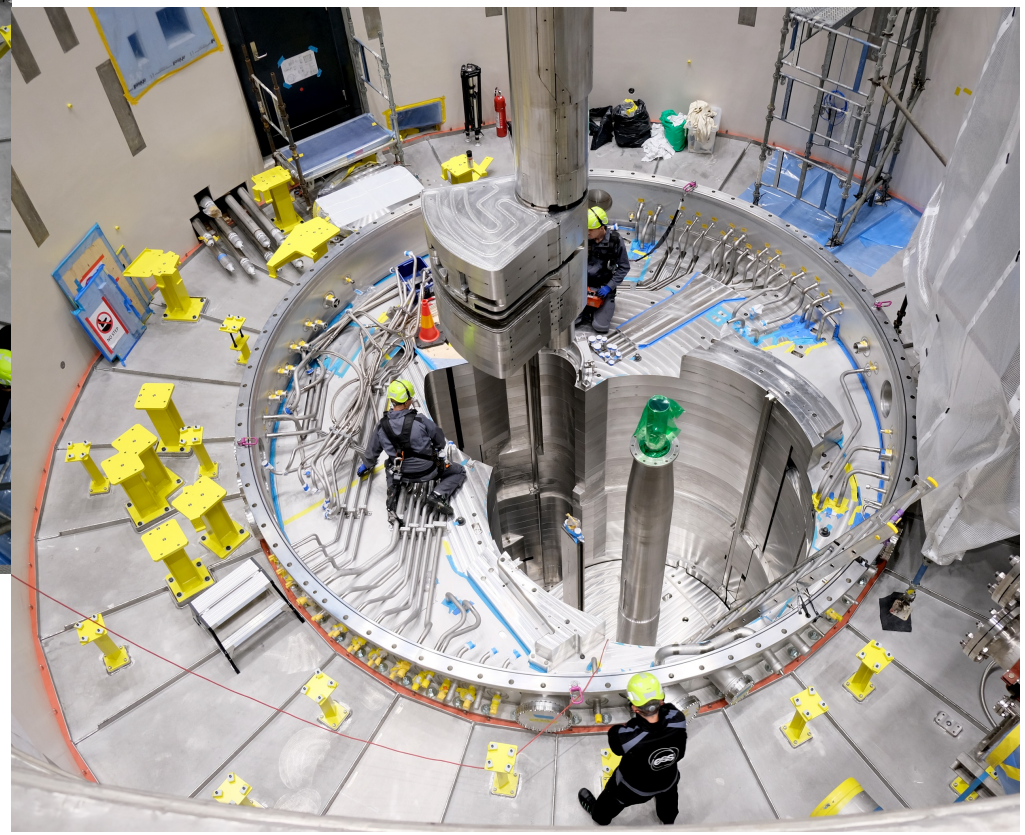
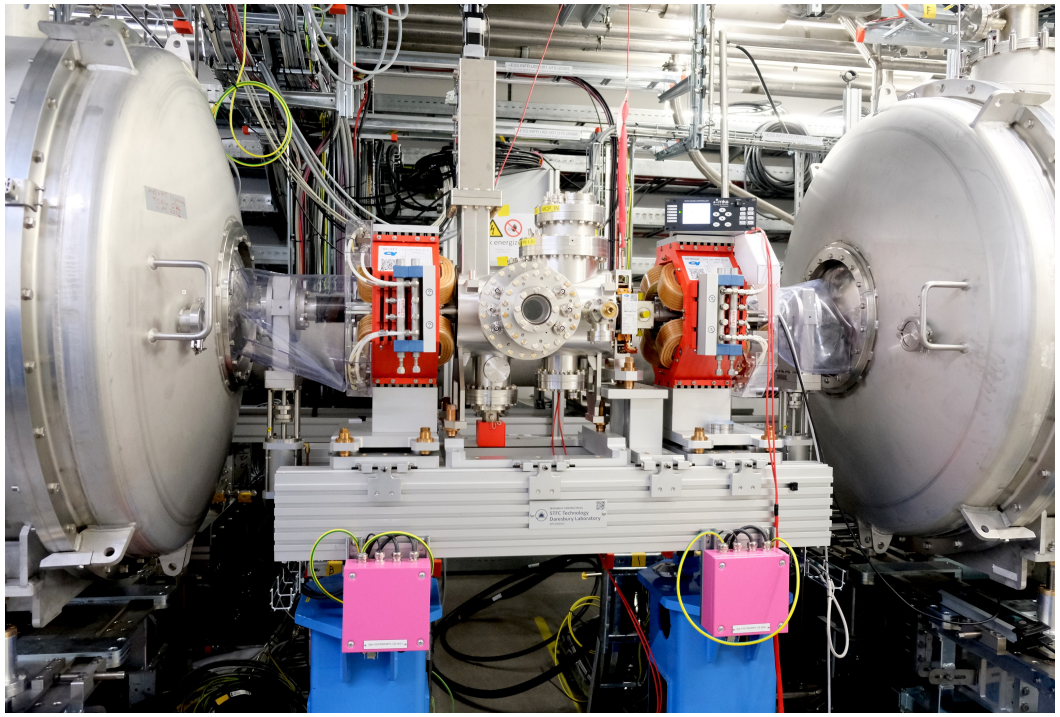
Development of Standard MicroTCA Deployment at ESS

PRESENTED BY FAY CHICKEN

2023-12-05

Quick Update on ESS Construction





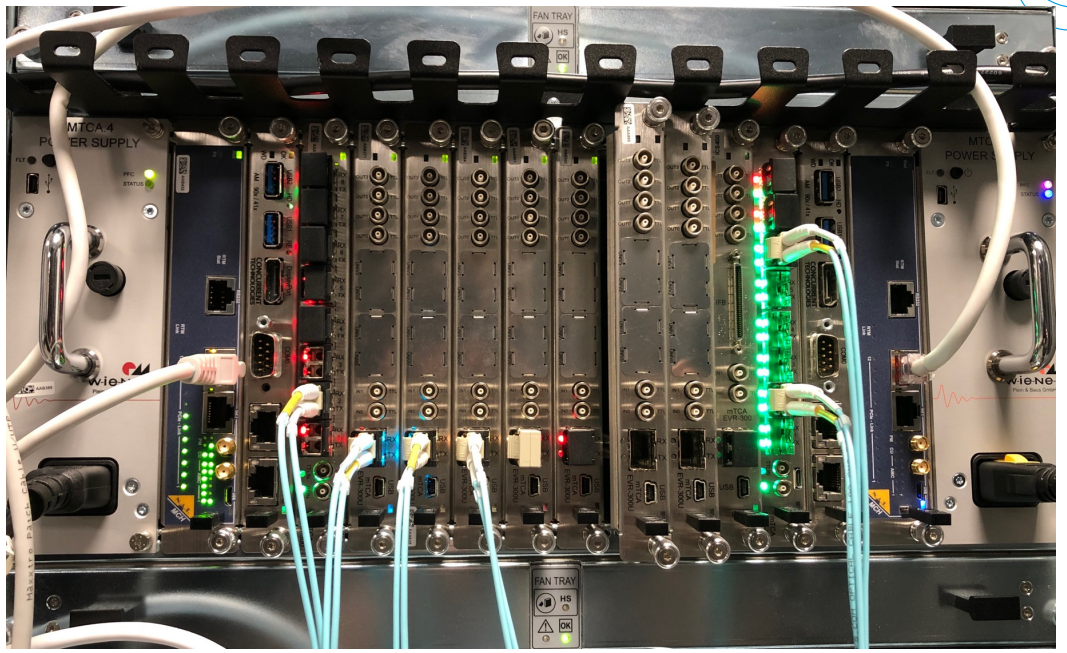
Drive for Standardisation

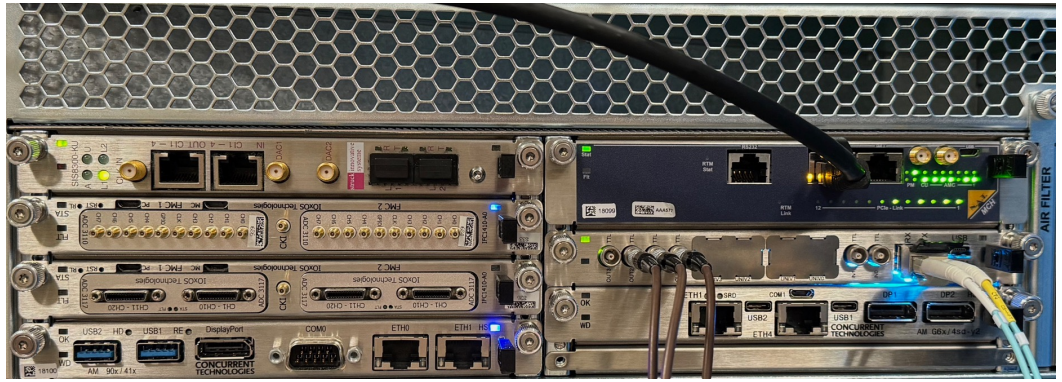
Over 300 MicroTCA systems will be deployed at the European Spallation Source

Early system assemblies were all individual setups

Need to have a basic “vanilla” system setup for easy maintenance and system reliability

Various AMCs, FMCs & RTM configurations creates enough complexity





Distribution of MTCA at ESS

ESS ~ 300			
RF	BI	TD	MP
175 x 9U	70 x 3U 15 x 9U	35 x 9U	10 x 3U

Deployed with combinations of:
6 x AMC
7 X RTM
12 x FMC
+ options of Power Supplies, Universal I/O modules for timing and CPU (older obsolete version also in operation)

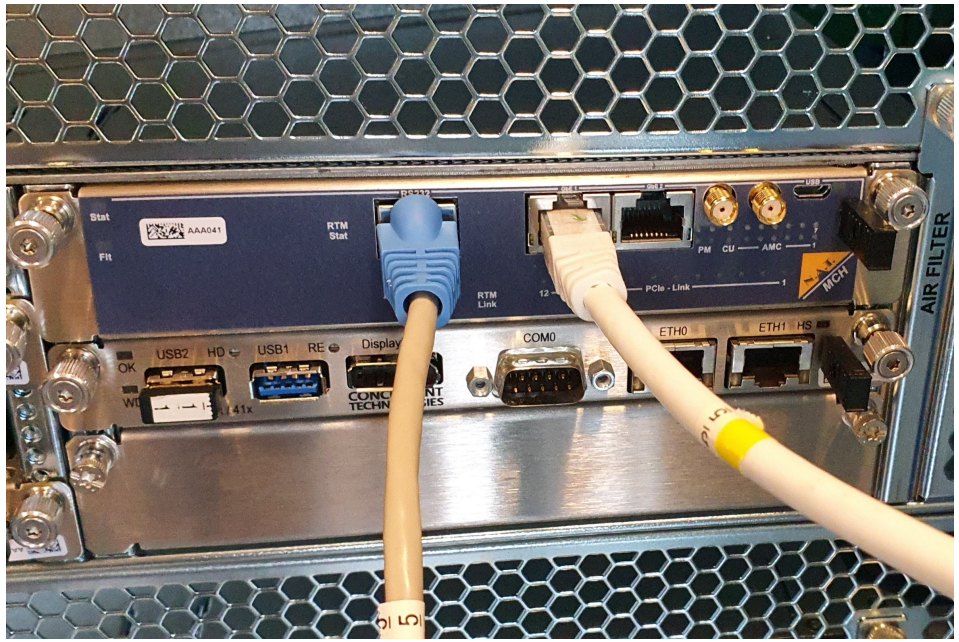
Only board standard to all MTCA is NAT-MCH-PHYS

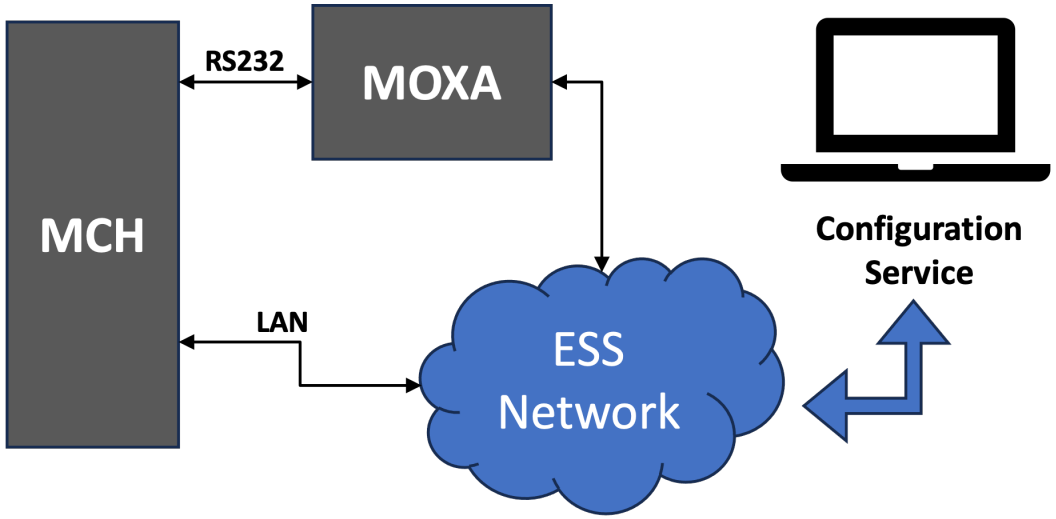
Micro-Carrier Hub Deployment

Use of RS232 through MOXA server for initial configuration using serial connection

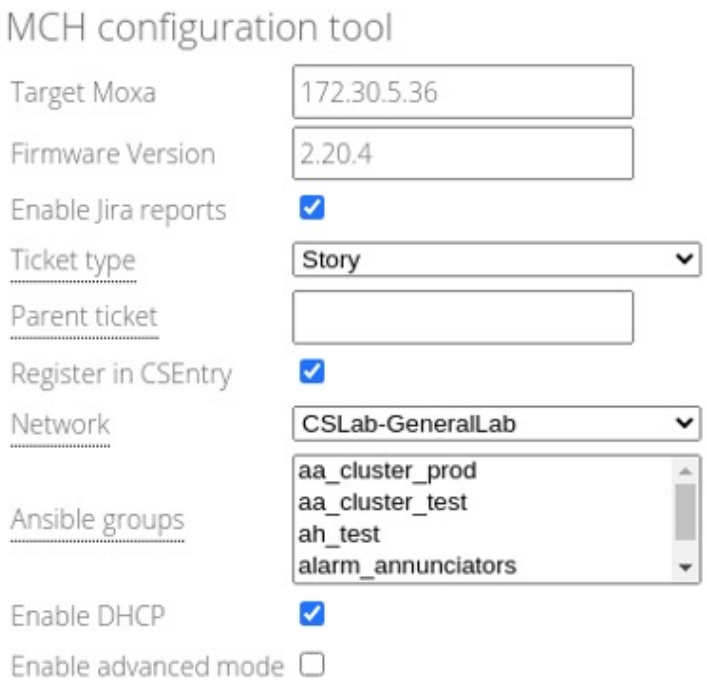
Register in Csentry (a custom in-house configuration management tool, REST API)

Switch to Telnet connection once IP is assigned on local network





First Deployment



The first version of this process was a command line tool in Bash with a simple HTML graphical user interface (GUI) which was just a wrapper for the command line. This gave a web-based interface that was simple for anybody needing to deploy an MCH.

Second Deployment

A second more advanced version used Jupyterhub (a multi-user web server for Jupyter notebooks) as the interface. The Bash was converted to Python programming language, which again could be run through the command line but this time with "python3" needed as a prerequisite. Otherwise, the Jupyterhub notebook provided a nice GUI, with easy changes to the MOXA IP address, port number, and backplane configuration possible.

Deployment of a new crate

Change the values for the following set of variables, then press the play button in the top bar.

```
# Port number in the MOXA Hub, no need to specify for deployed crates
moxa_mch_port=2
# Type of crate: 3U-> 3 or 9U -> 9
mch_backplane=9
```

The following variables are optional, if not changed, they'll take a default value:

```
# MOXA configuration ----------
# IP address of the MOXA hub
moxa_ip_addr="172.30.5.37"

# Jira configuration ----------
# Don't share this token with anyone!
jira_credential=""
jira_parent_ticket=""
jira_ticket_type="Story"

# CSEntry configuration -------
# VLAN for the CSEntry host
network_vlan="CSLab-GeneralLab"
# Ansible groups
ansible_groups=""
# Don't share this token with anyone!
csentry_token=
# URL to the CSEntry API
csentry_url="https://csentry.esss.lu.se/"
```

```
%%capture
pip install csentry-api==1.0.2
```

Current Deployment

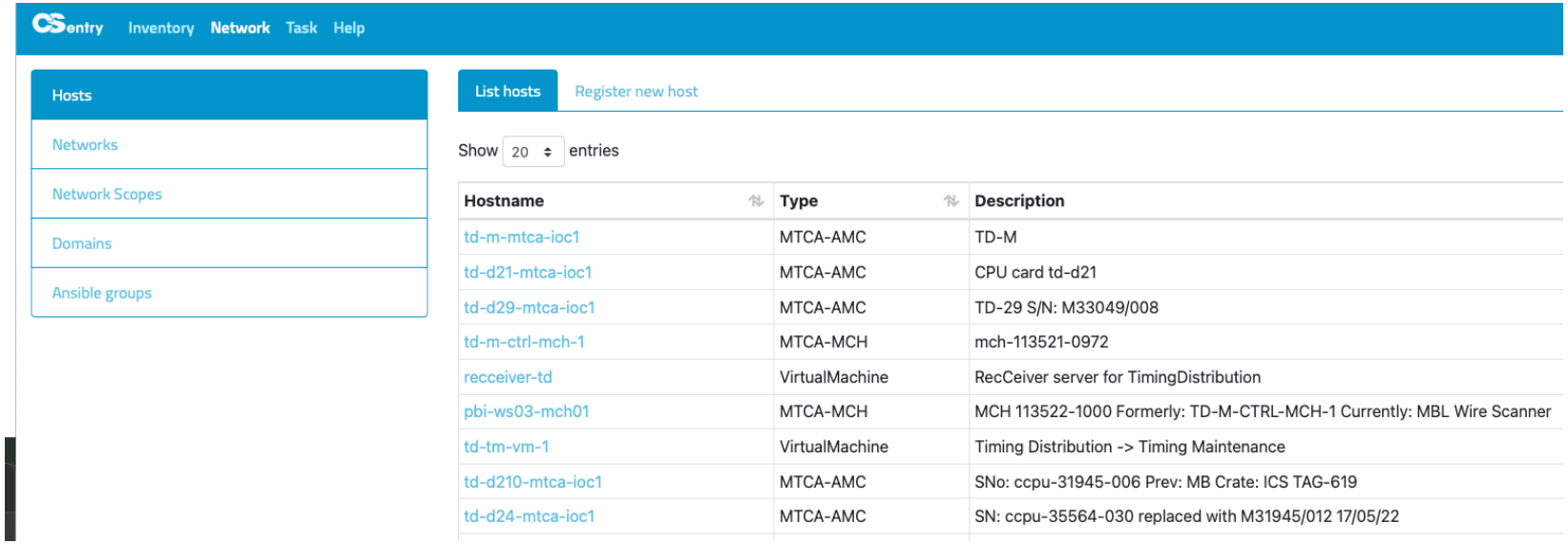
Now deployment of this script is either through a virtual environment to run the Python scripts on a lab-based workstation or directly using Gitlab-CI (DevOps platform with continuous integration (CI)). The fixed IP address for the MOXA and the port number, along with the backplane type are set via the script command line arguments

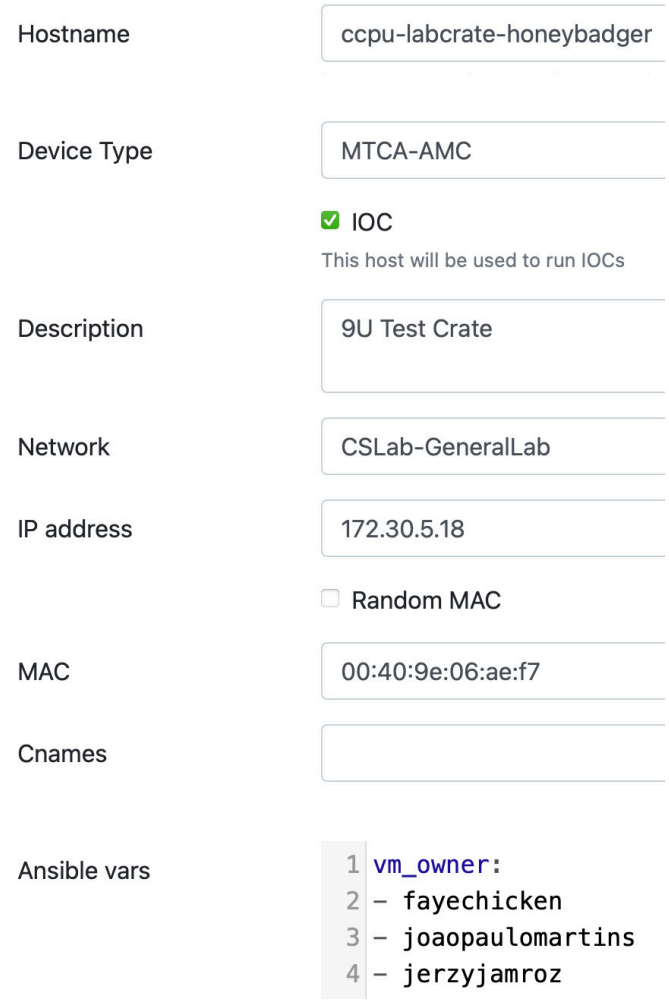
```
Running with gitlab-runner 15.10.0 (456e3482)
  on fat WBy1vi_Y, system ID: s_0c4c9508ef21
Resolving secrets                                                    00:00
Preparing the "shell" executor                                       00:00
Using Shell (bash) executor...
Preparing environment                                                00:00
Running on cslab-wp4-workstation.cslab.esss.lu.se...
Getting source from Git repository                                   00:01
Fetching changes with git depth set to 50...
Reinitialized existing Git repository in /var/lib/gitlab-runner/builds/WBy1vi_Y/0/hwcore/mtca/mch/.git/
Checking out 6ef772e5 as detached HEAD (ref is main)...
Removing __pycache__/
Removing logs/log_20230927_101454
Removing plugs/__pycache__/
Skipping Git submodules setup
Executing "step_script" stage of the job script
$ echo "/usr/bin/python3 mch.py --token *** $PARAMS"
/usr/bin/python3 mch.py --token *** --moxa_ip 172.30.5.36 --moxa_port 3 --mch_backplane 3 --update_fw 2.22.3 --configure_mch --reg_csentry
$ /usr/bin/python3 mch.py --token $CSEntryToken $PARAMS
2023-09-27 10:18:52,724 INFO mch.py:127 Log file: ./logs/log_20230927_101852
2023-09-27 10:18:52,724 INFO mch.py:135 Working on MCH
2023-09-27 10:18:52,724 INFO mchlib.py:73 Attempting to connect to the MCH using the MOXA backend
2023-09-27 10:18:52,724 INFO nat_mch.py:125 GenDev::Constructor - A new device has been registered
Device model: MCH
2023-09-27 10:18:52,724 INFO nat_mch_telnet.py:130 NATMCHTelnet - NAT MCH Telnet instance created.
MCH IP Address:
```

CSentry

Control System Entry

a custom in-house configuration management tool, REST API





CPU Deployment

Change to BIOS settings to allow PXE boot

Register in CSentry

Linux OS is installed using network boot installer

Post-install job in Ansible installs ESS EPICS Environment and all the standard libraries and kernel drivers

Back to Jobs | Details | Output

run-post-install-configuration — Plays 10 | Tasks 142 | Hosts 1 | Elapsed 00:02:17

```
0  Identity added: /tmp/awx_414100_26mdr46s/artifacts/414100/ssh_key_data (/tmp/awx_414100_26mdr46s/artifacts/414100/ssh_key_data)
1  Vault password:
2
3  PLAY [all] ********************************************************************** 12:28:00
4
5  TASK [Gathering Facts] ********************************************************** 12:28:00
6  ok: [ccpu-38533-045.cslab.esss.lu.se]
7
8  PLAY [all] ********************************************************************** 12:28:02
9
10 TASK [ics-ans-role-dns-client : Configure /etc/resolv.conf] ******************** 12:28:02
11 changed: [ccpu-38533-045.cslab.esss.lu.se]
12
13 TASK [ics-ans-role-dns-client : Remove DNS entries from centos ifcfg files] **** 12:28:04
14 included: /tmp/awx_414100_26mdr46s/requirements_roles/ics-ans-role-dns-client/tasks/centos.yml for ccpu-38533-045.cslab.esss.lu.se
15
16 TASK [ics-ans-role-dns-client : Find files that contains the DNS and SEARCH options in /etc/sysconfig/network-scripts] *** 12:28:04
```

ESS Linux Distribution

- YOCTO Project version Dunfell 23.0.21+ - Linux 4.14
- The objective is to provide a minimal system for running EPICS IOCS and in middleware.
- Supports:
  - Concurrent AMG6x/AM900 (Intel XEON/ Core-i7 64-bit)
  - IOxOS IFC14xx (NXP QorIQ PowerPC 64-bit)
  - Both systems are supported also with real time Linux.
- We also provide users with a small amount of tools for test and debugging natively.