# SDN e Ansible per il sistema di acquisizione di KM3NeT

Lorenzo Chiarelli, Tommaso Chiarusi, Emidio Giorgio, Stefano Zani

Workshop di CCR INFN

Rimini 15 Giugno 2018

# The KM3NeT collaboration

- Collaboration of 51 institutions in 15 countries
- 2 submarine detectors : **ARCA** (Portopalo), **ORCA**(Toulon)
  - Building Blocks (2 ARCA, 1 ORCA)
    - 115 Strings (aka lines)
      - 18 DOM + 1 Base Module
        - 31 PMT
- **All-data to shore** : Detectors are connected via deep-sea electro-optical cable (EOC) to onshore station, where raw data is reconstructed and filtered

# (some of ) DAQ actors

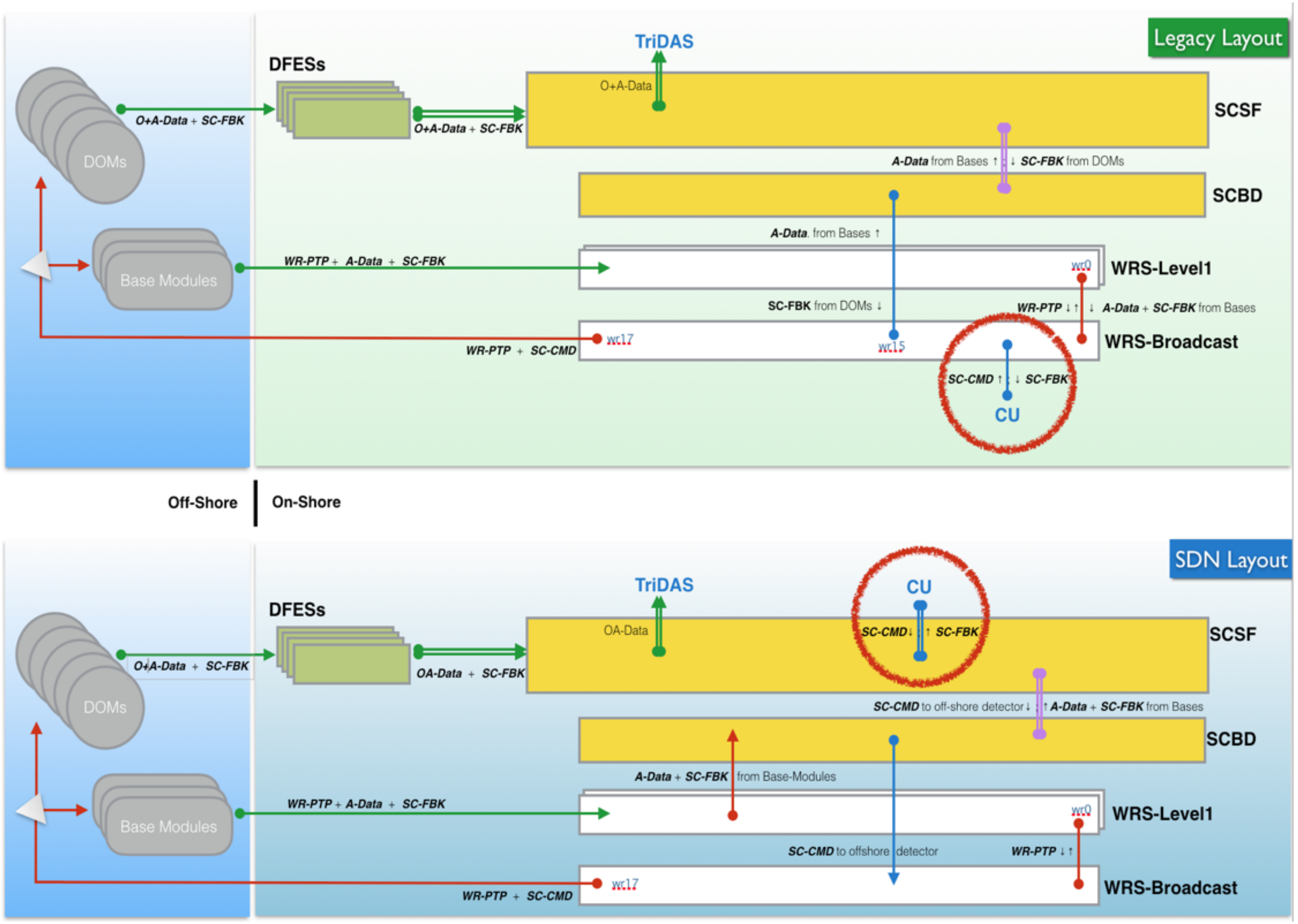
- DOMs (and DU bases) bitstream is processed by **TriDAS**
  - DataQueues reassambles ethernet frames sent by DOMs
  - DataFilter applies online trigger and selects ‘’good’’ events
  - **CU** is TriDAS process which orchestrates on/offshore resources through SlowControl commands (**SC**)
- Type of network streams:
  - SC-CMD, SC-FBK: the slow control commands and feedbacks exchanged between the CU and the detector;
  - O + A-Data: the optical and acoustic data, respectively (the Base-modules don’t produce any optical data)

# RAW Data LAN

- **RAW network** is the LAN connecting TriDAS with DOMs
  - Asymmetric: in order to optimize fibers in EOC, information (on->off) shore is embedded in a single stream of data. All DOMS share the same information, but only the real receiver process it. ~100 Kb/s with 2BB
  - Hybrid switching layout :
    - White Rabbit Switch fabric : Customization of WR protocol, instead of p2p connection from WS switch to devices, there is an intermediate layer (WRS-layer1), feed by **WRS Broadcast**
      - that allows also to scale up to 18^2 Strings
    - Standard Switch fabric: DOM Front End Switch (DFES), SlowControl and Base Data Switch (SDBC)

# Why SDN ?

- Without SDN, CU had to be connected to WRS broadcast
  - Loops followed otherwise
- Same for SC-FBK and A-Data
  - As the detector scales up, **WRS performance degrade**
- SDN it is used to define routes for specific data flows
- Openflow 1.3
  - OpenDaylight Nitrogen



# SDN Flows

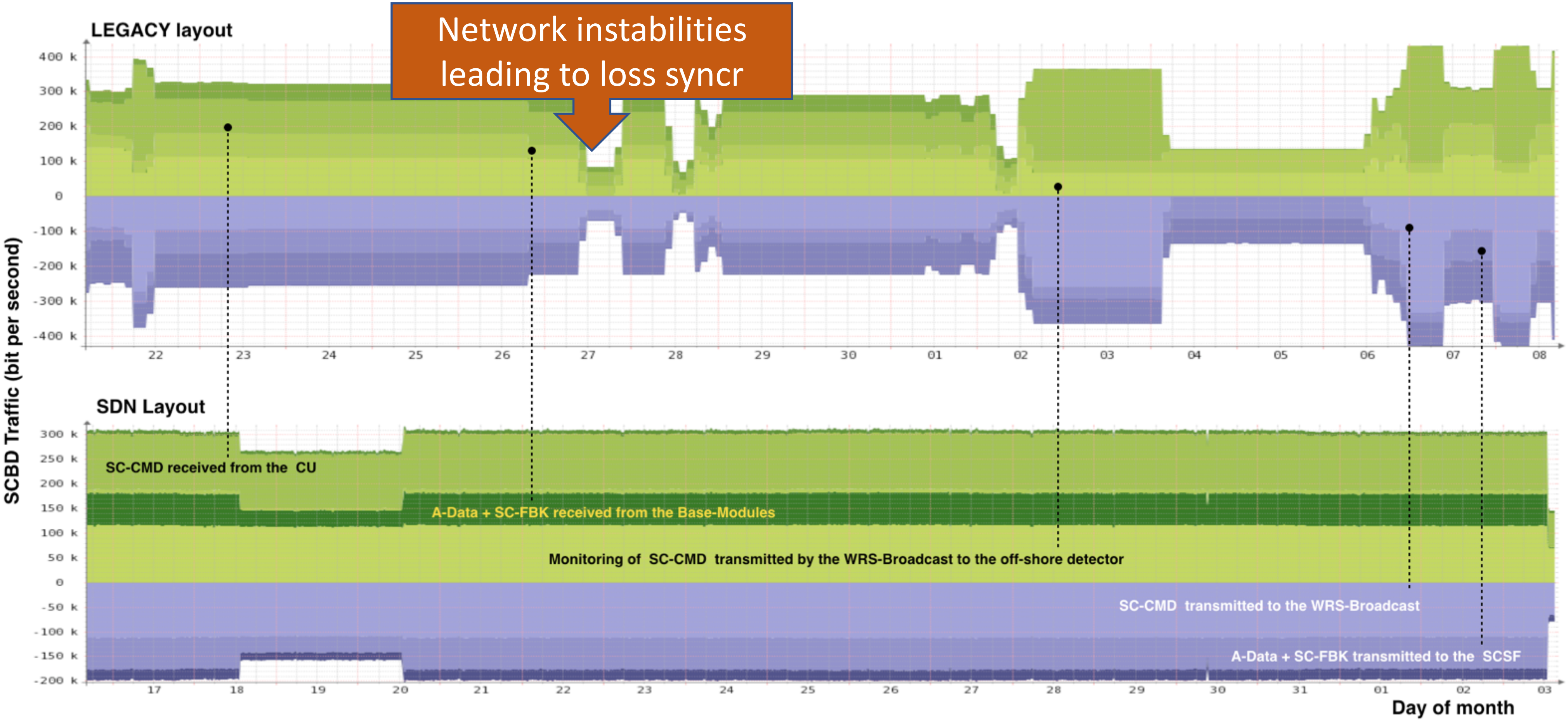
| #ID | Source | Destination | Action |
|---|---|---|---|
| SCSF-1 | any | ff:ff:ff:ff:ff:ff (broadcast) | To raw-dhcp-server |
| SCSF-2 | 08:00:30:00:00:00/ff:ff:ff:00:00:00 | Control-unit | SC-FBK to CU |
| SCSF-3 | 08:00:30:00:00:00/ff:ff:ff:00:00:00 | TriDAS Front-end (DAQ server) | O+A Data to TriDAS |
| SCSF-4 | CU | 08:00:30:00:00:00/ff:ff:ff:00:00:00 | SC-CMD to SCBD |

| #ID | Source | Destination | Action |
|---|---|---|---|
| SCBD-1 | 08:00:30:00:00:00/ff:ff:ff:00:00:00 | any | (SC-FBK,A-Data) to SCSF |
| SCBD-2 | Everything from SCSF uplink | 08:00:30:00:00:00/ff:ff:ff:00:00:00 | SC-CMD to WRS-B/cast |
| SCBD-3 | 08:00:30:00:00:00/ff:ff:ff:00:00:00 | ff:ff:ff:ff:ff:ff | To SCSF |

# Rule example

```
[ {  "flow": [   {
      "id": "3",
      "match": {
        "ethernet-match": {
          "ethernet-source":
            {"address": "08:00:30:00:00:00",
            "mask": "ff:ff:ff:00:00:00"},
          "ethernet-destination": {
            "address": "00:26:18:2c:73:91"
          }}},
      "instructions": {
        "instruction": [
          {  "order": "0",
                  "apply-actions": {
                    "action": [
                      {   "output-action": {
                            "output-node-connector":
              "openflow:303570285128704:86",
                            "max-length": "60"
                          },
                          "order": "0"  } ]}}]},
                "flow-name": "SCSF_DOMtoDQ",
                "installHw": "true",
                "idle-timeout": "0",
                "cookie": "3",
                "table_id": "1"}]}
```

# Performances

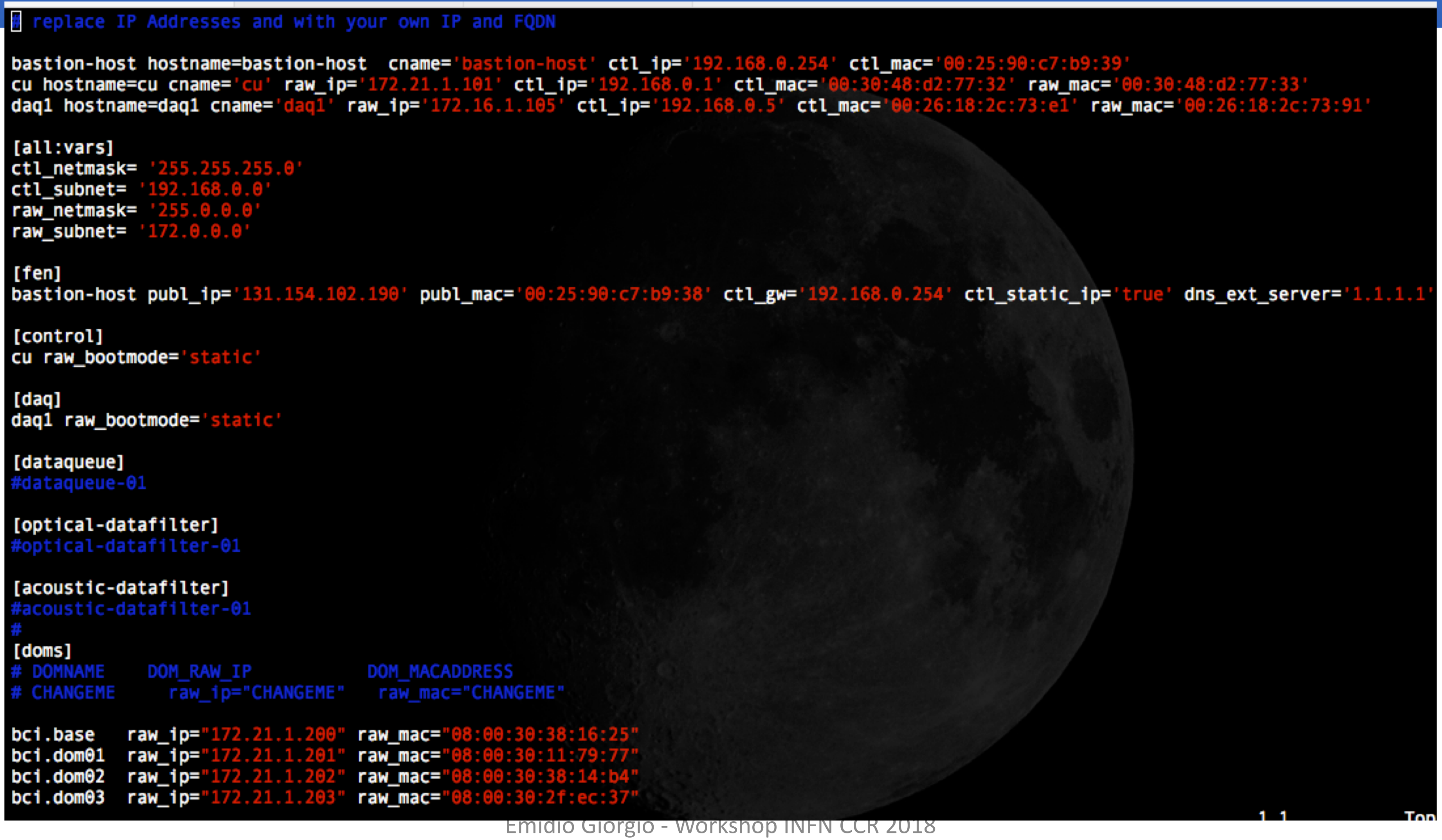


Network instabilities leading to loss syncr

# Why Ansible ?

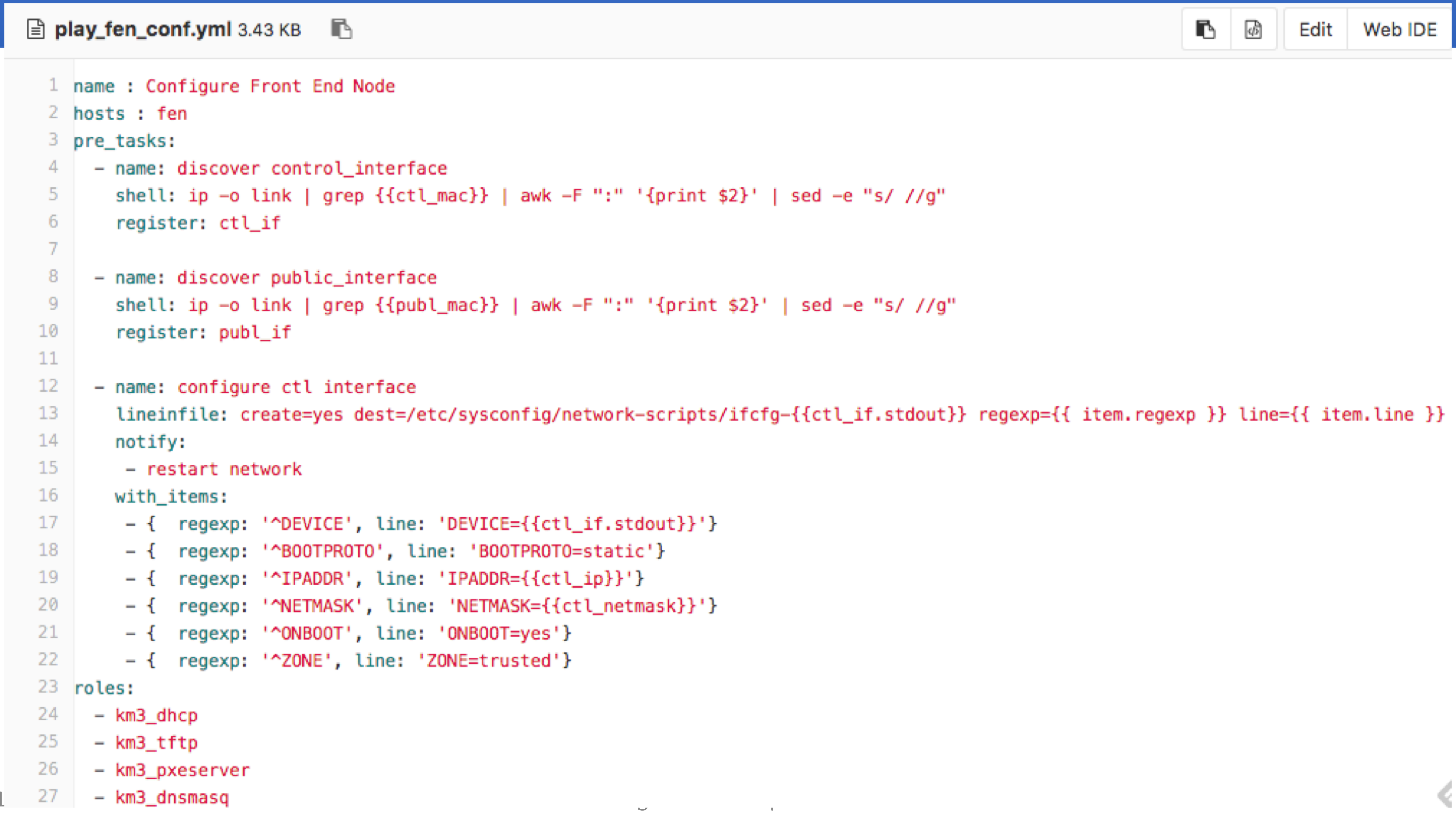
- Aside with shore stations datacenters, KM3NeT has several integration and test stations
  - Non trivial configuration : two private network, external connectivity....
  - Need to uniform and simplify system and exp software setup
  - Reduce request for IT support
- OS Installation with Foreman or Kickstart
- Single configuration file for test site
- Roles for FrontEnd Node, CU and DAQ server
  - Fen : pxeserver, nat, CTL net dhcp
- Repo : [http://git.km3net.de/egiorgio/DAQAutomatedSetup](http://git.km3net.de/egiorgio/DAQAutomatedSetup)

# Example : Inventory



```
# replace IP Addresses and with your own IP and FQDN

bastion-host hostname=bastion-host  cname='bastion-host' ctl_ip='192.168.0.254' ctl_mac='00:25:90:c7:b9:39'
cu hostname=cu cname='cu' raw_ip='172.21.1.101' ctl_ip='192.168.0.1' ctl_mac='00:30:48:d2:77:32' raw_mac='00:30:48:d2:77:33'
daq1 hostname=daq1 cname='daq1' raw_ip='172.16.1.105' ctl_ip='192.168.0.5' ctl_mac='00:26:18:2c:73:e1' raw_mac='00:26:18:2c:73:91'

[all:vars]
ctl_netmask= '255.255.255.0'
ctl_subnet= '192.168.0.0'
raw_netmask= '255.0.0.0'
raw_subnet= '172.0.0.0'

[fen]
bastion-host publ_ip='131.154.102.190' publ_mac='00:25:90:c7:b9:38' ctl_gw='192.168.0.254' ctl_static_ip='true' dns_ext_server='1.1.1.1'

[control]
cu raw_bootmode='static'

[daq]
daq1 raw_bootmode='static'

[dataqueue]
#dataqueue-01

[optical-datafilter]
#optical-datafilter-01

[acoustic-datafilter]
#acoustic-datafilter-01
#
[doms]
# DOMNAME     DOM_RAW_IP            DOM_MACADDRESS
# CHANGEME      raw_ip="CHANGEME"    raw_mac="CHANGEME"

bci.base   raw_ip="172.21.1.200" raw_mac="08:00:30:38:16:25"
bci.dom01  raw_ip="172.21.1.201" raw_mac="08:00:30:11:79:77"
bci.dom02  raw_ip="172.21.1.202" raw_mac="08:00:30:38:14:b4"
bci.dom03  raw_ip="172.21.1.203" raw_mac="08:00:30:2f:ec:37"
                                                                 1,1           Top
```

# FEN playbook



```
name : Configure Front End Node
hosts : fen
pre_tasks:
  - name: discover control_interface
    shell: ip -o link | grep {{ctl_mac}} | awk -F ":" '{print $2}' | sed -e "s/ //g"
    register: ctl_if

  - name: discover public_interface
    shell: ip -o link | grep {{publ_mac}} | awk -F ":" '{print $2}' | sed -e "s/ //g"
    register: publ_if

  - name: configure ctl interface
    lineinfile: create=yes dest=/etc/sysconfig/network-scripts/ifcfg-{{ctl_if.stdout}} regexp={{ item.regexp }} line={{ item.line }}
    notify:
     - restart network
    with_items:
     - {  regexp: '^DEVICE', line: 'DEVICE={{ctl_if.stdout}}'}
     - {  regexp: '^BOOTPROTO', line: 'BOOTPROTO=static'}
     - {  regexp: '^IPADDR', line: 'IPADDR={{ctl_ip}}'}
     - {  regexp: '^NETMASK', line: 'NETMASK={{ctl_netmask}}'}
     - {  regexp: '^ONBOOT', line: 'ONBOOT=yes'}
     - {  regexp: '^ZONE', line: 'ZONE=trusted'}
roles:
  - km3_dhcp
  - km3_tftp
  - km3_pxeserver
  - km3_dnsmasq
```

# Next steps

- SDN Rules
  - Anti flooding
  - Easy port mirroring
- Complete integration of DAQ roles
  - SDN controller
  - CU configuration
  - Data Filter and Data Queues

# Questions ?