PAUL SCHERRER INSTITUT

Leonardo Sala :: AWI :: Paul Scherrer Institut

Computing and deployment

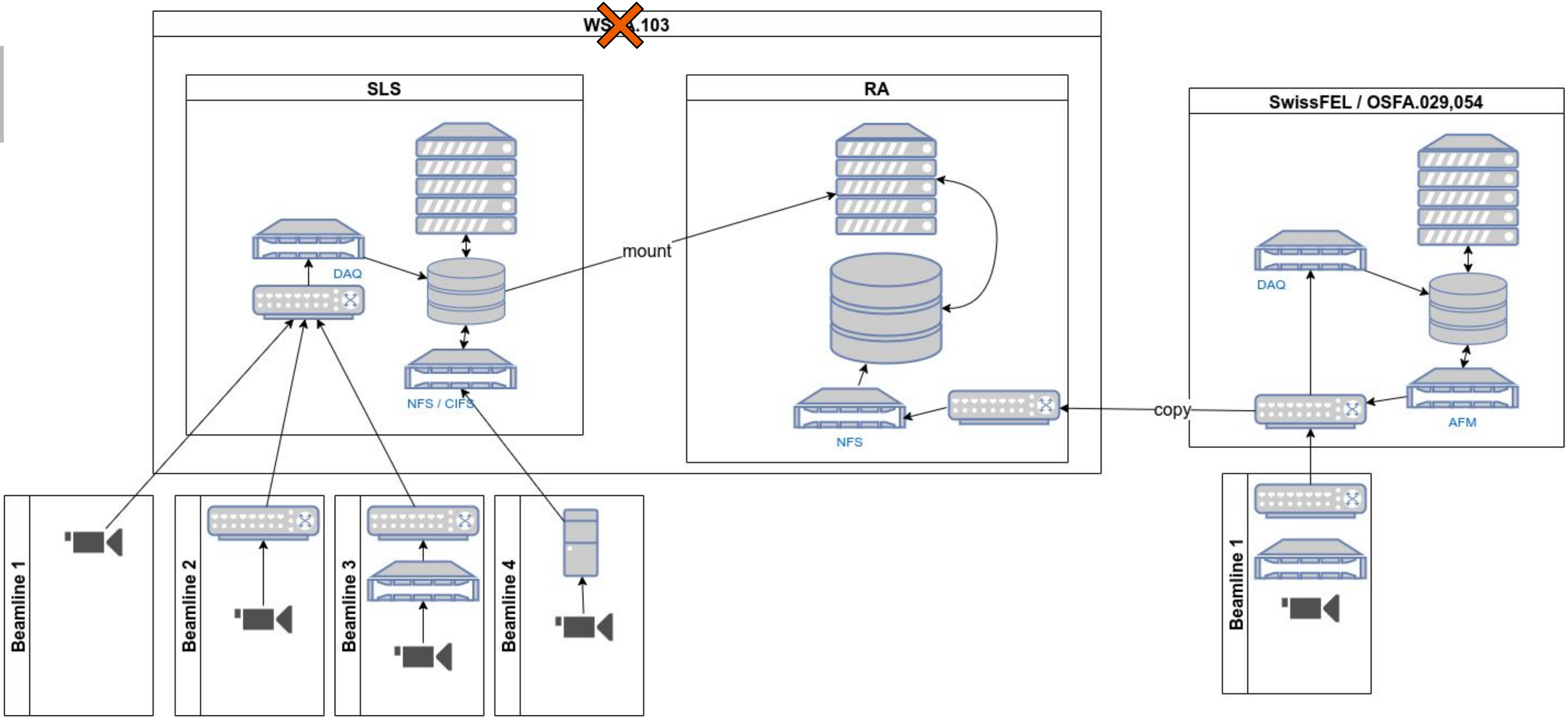
Meeting with SOLEIL 2023.09.19 / PSI

SLS / SwissFEL overview

On the Photons side, we do manage three clusters:

- **SLS:** dedicated compute / storage for the SLS beamlines
- **SwissFEL:** dedicated compute / storage for the SwissFEL beamlines
- **RA:** shared photonics data analysis facility

SLS / SwissFEL overview



SLS / SwissFEL overview

- IBM Storage Scale as high-performance storage
- Infiniband (EDR) as main storage fabric
- 100G Ethernet as main DAQ fabric
- standard Intel / AMD processors
  - limited amount of GPUs (~10)
- RHEL 7 (transitioning to 8)

Some numbers

- **~9000 cores, ~20 PiB, ~70 TB ram, ~30 managed switches (Infiniband, Ethernet)**
- managed by **Puppet and ansible**
  - infrastructure as code backed up by Gitlab
  - Puppet for basic / standard OS installation
  - Ansible for special setups, pipelines and operations
- monitored by **Icinga and InfluxDB / Grafana**
  - looking into ELK (Central IT)
  - soon migration to Icinga2
- We even have a test Openshift k8s cluster
  - used for gitlab runners and tests

Current DC situation

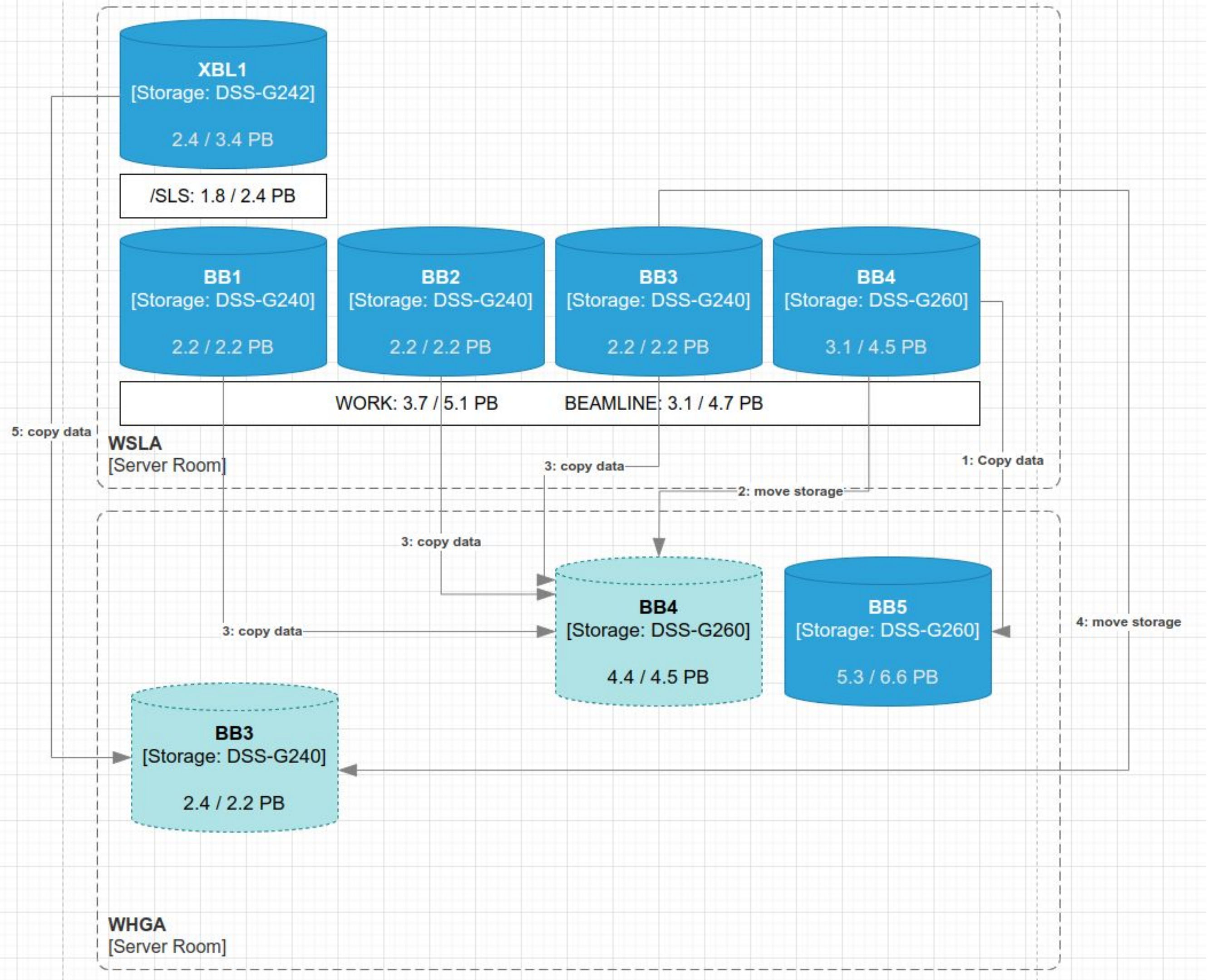
PSI has various Server rooms:

- Main West: 300 kW
- Main East: 80 kW
- Main SLS: 90 kW (decommissioning)
- SwissFEL: 60 + 30 kW

Current mid-term strategy is to consolidate, but there are limits in power that will be problematic

Due to SLS shutdown, we need to migrate all our compute to Main West server room - 2 years project, gradual and transparent migration with minimal downtimes

- extended IB fabric over 600m link
- gradual compute migration
- background data copy


XBL1
[Storage: DSS-G242]
2.4 / 3.4 PB
/SLS: 1.8 / 2.4 PB
BB1
[Storage: DSS-G240]
2.2 / 2.2 PB
BB2
[Storage: DSS-G240]
2.2 / 2.2 PB
BB3
[Storage: DSS-G240]
2.2 / 2.2 PB
BB4
[Storage: DSS-G260]
3.1 / 4.5 PB
WORK: 3.7 / 5.1 PB
BEAMLINE: 3.1 / 4.7 PB
5: copy data
WSLA
[Server Room]
3: copy data
1: Copy data
2: move storage
3: copy data
BB4
[Storage: DSS-G260]
4.4 / 4.5 PB
BB5
[Storage: DSS-G260]
5.3 / 6.6 PB
3: copy data
4: move storage
BB3
[Storage: DSS-G240]
2.4 / 2.2 PB
WHGA
[Server Room]

Storage details

We do heavily rely on IBM Storage Scale since years

Key technologies:

- AFM for data migration between SwissFEL and RA
- storage vs compute clusters for data access policies
- GNR for distributed RAID and disk hospitals
- NFS / CIFS High Availability exports using Protocol Nodes
- Policy scans for deletion and reporting
- Future: back to tiering with new SSD storage

For long term storage we rely on Tapes provided by CSCS

SciCat and tape

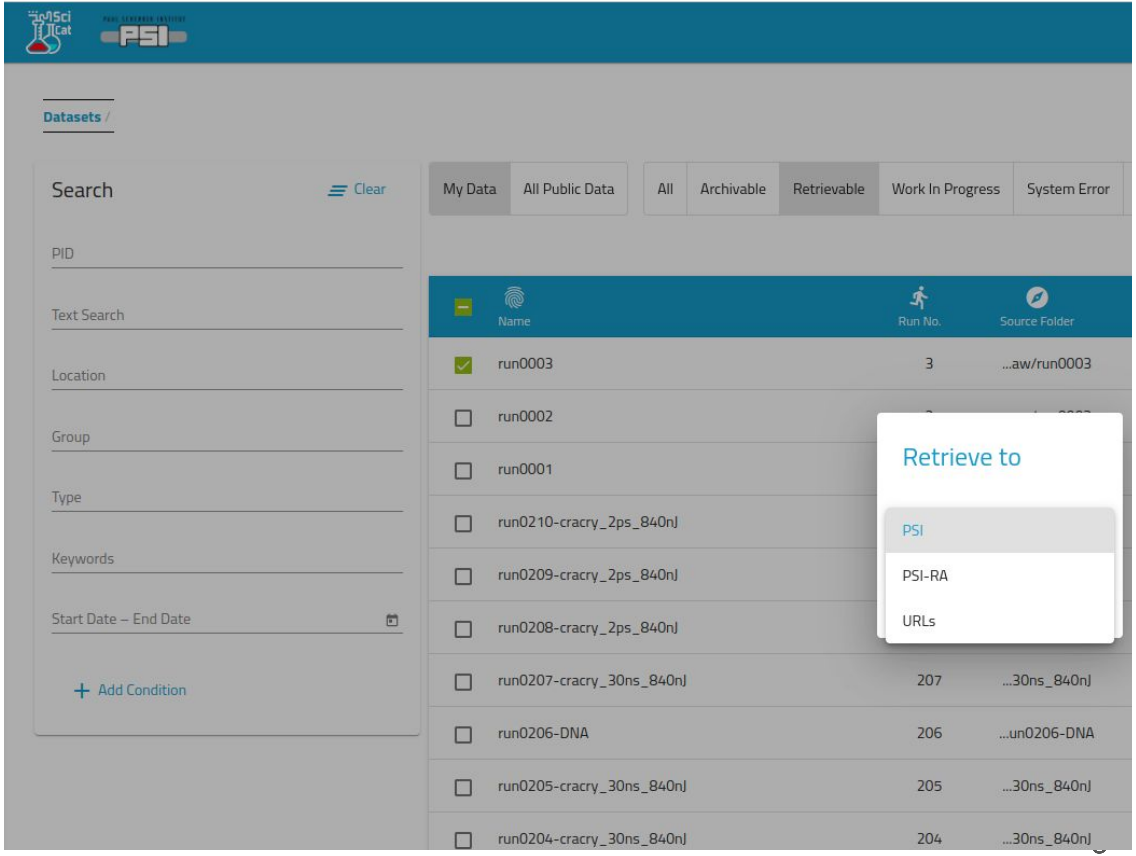
Data management functionality provided by SciCat

Copy to tape:

- Automatic for SwissFEL
- Managed by Beamlines in SLS

Retrieve from tape:

- to general PSI storage
- to RA cluster
- to CSCS object storage



−	Name	Run No.	Source Folder
☑	run0003	3	...aw/run0003
□	run0002		
□	run0001		
□	run0210-cracry_2ps_840nJ		
□	run0209-cracry_2ps_840nJ		
□	run0208-cracry_2ps_840nJ		
□	run0207-cracry_30ns_840nJ	207	...30ns_840nJ
□	run0206-DNA	206	...un0206-DNA
□	run0205-cracry_30ns_840nJ	205	...30ns_840nJ
□	run0204-cracry_30ns_840nJ	204	...30ns_840nJ

Various phases of deployment

We use different flavours of automation

- Linux Group provides standard RHEL installation over puppet
- Ansible playbooks for operations, including
  - server installation
  - updates
  - special services deployment
- Gitlab runners + playbooks for user-driven deployments

Linux installations

Central puppet modules maintained by Linux group and experts

- both own-developed and from external libraries
- standard development cycle in GitLab:
  - feature branch -> preprod ->prod
  - weekly merge meetings

Servers are collected into **groups**

- every expert group is responsible of their infrastructure
- separate hiera GIT repos
- linux inventory based on own-developed lightweight system (sysdb)

Puppet dashboard also available

Central Linux infrastructure **fully reproducible** from scratch

Example: server installation

Our server installation is mostly automated:

- physical server installation
- register default admin credentials and required variables
- run playbook that:
  - configure BIOS based on profiles
  - register system in linux inventory
  - configure RAID, boot device, …
  - boot up server
- based in industry-standard Redfish API

Next steps:

- automatic filling of our Data Center management system (opendcim)
- this is possible now as we recently installed a version with a RESTful API

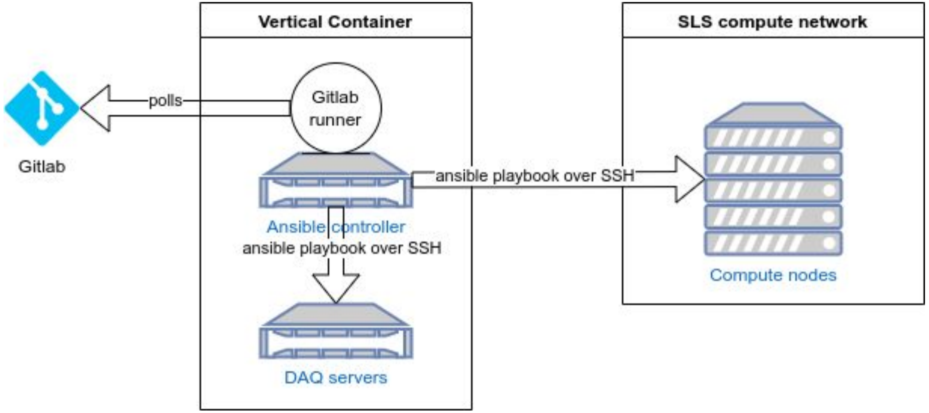
Gitlab pipelines - architecture

Solution:

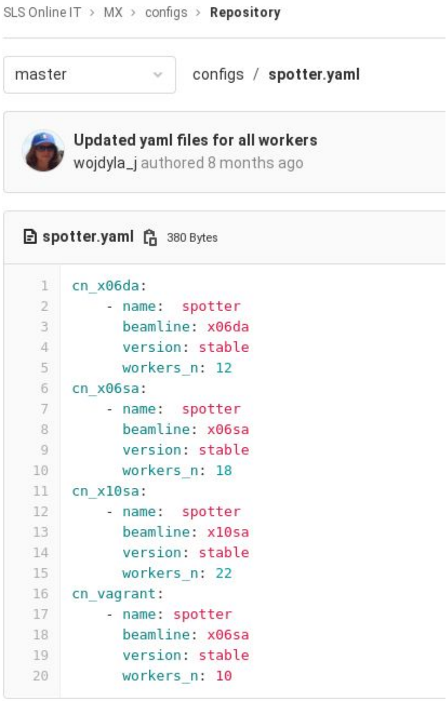
- IaaC with Ansible playbooks
- Gitlab as interface, deployment jobs through pipelines
- access control based on repositories

Advantages:

- fine grained control
- reproducible and versionable
- web interface to output



Pipelines



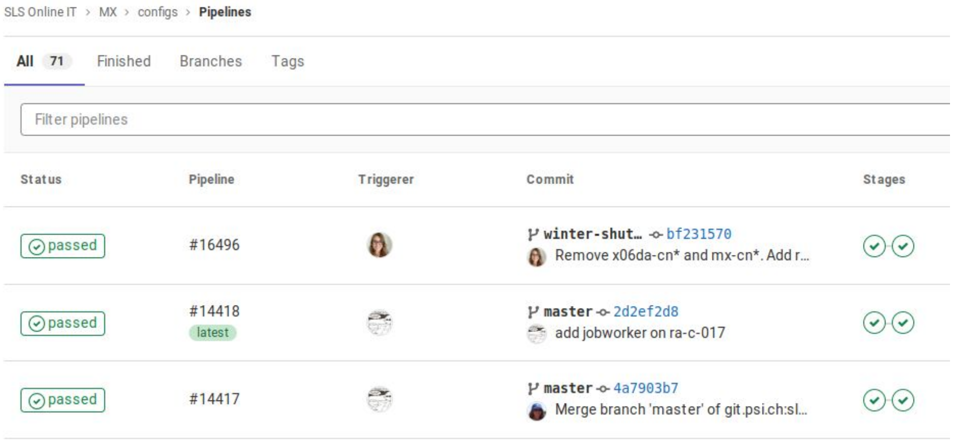
SLS Online IT › MX › configs › **Repository**

master configs / **spotter.yaml**

Updated yaml files for all workers
wojdyla_j authored 8 months ago

spotter.yaml 380 Bytes

```
cn_x06da:
    - name:  spotter
      beamline: x06da
      version: stable
      workers_n: 12
cn_x06sa:
    - name:  spotter
      beamline: x06sa
      version: stable
      workers_n: 18
cn_x10sa:
    - name:  spotter
      beamline: x10sa
      version: stable
      workers_n: 22
cn_vagrant:
    - name: spotter
      beamline: x06sa
      version: stable
      workers_n: 10
```

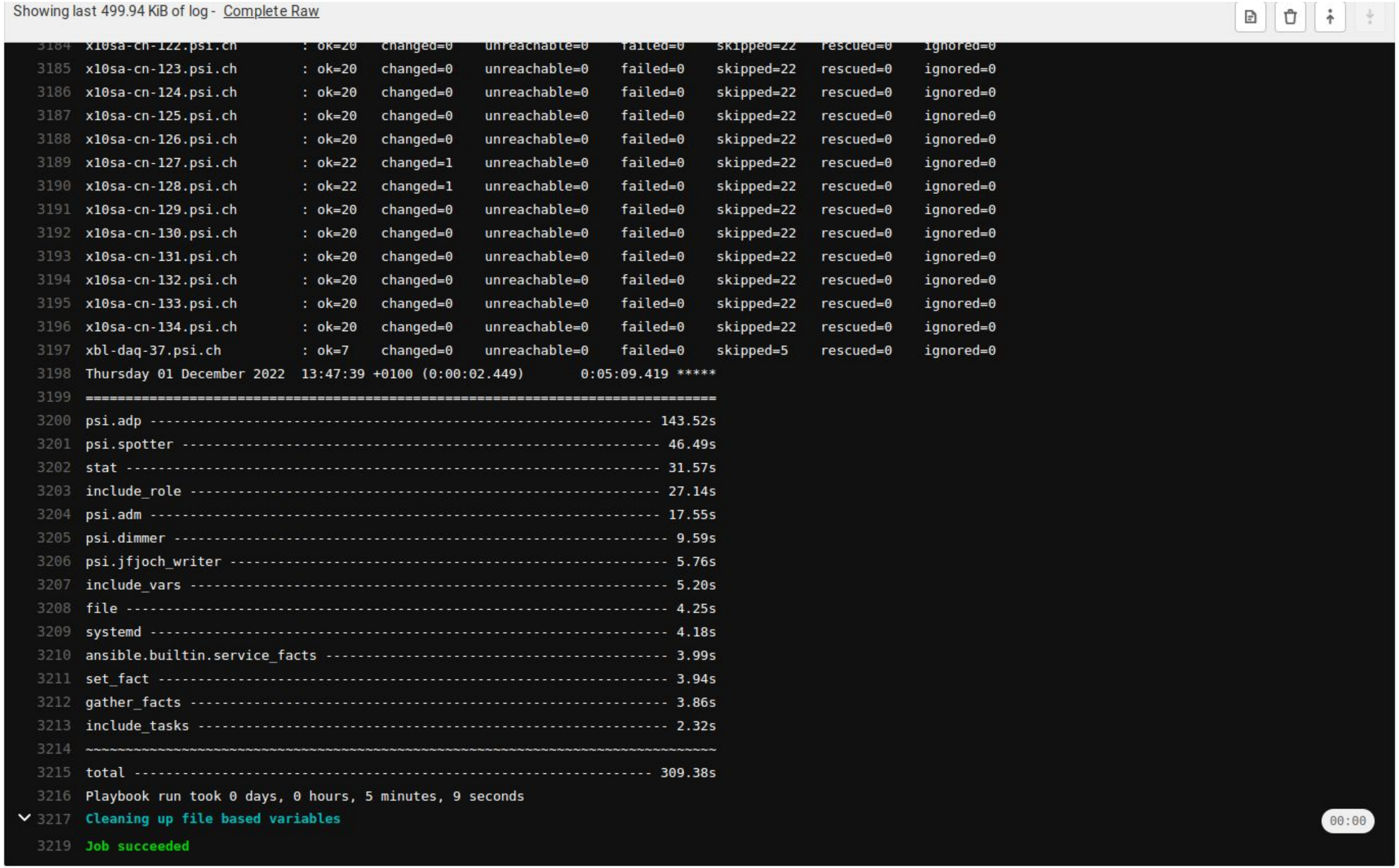


SLS Online IT › MX › configs › **Pipelines**

All 71 Finished Branches Tags

Filter pipelines

Status	Pipeline	Triggerer	Commit	Stages
passed	#16496		**winter-shut…** bf231570 Remove x06da-cn* and mx-cn*. Add r...	✓ ✓
passed	#14418 latest		**master** 2d2ef2d8 add jobworker on ra-c-017	✓ ✓
passed	#14417		**master** 4a7903b7 Merge branch 'master' of git.psi.ch:sl...	✓ ✓



Showing last 499.94 KiB of log - Complete Raw

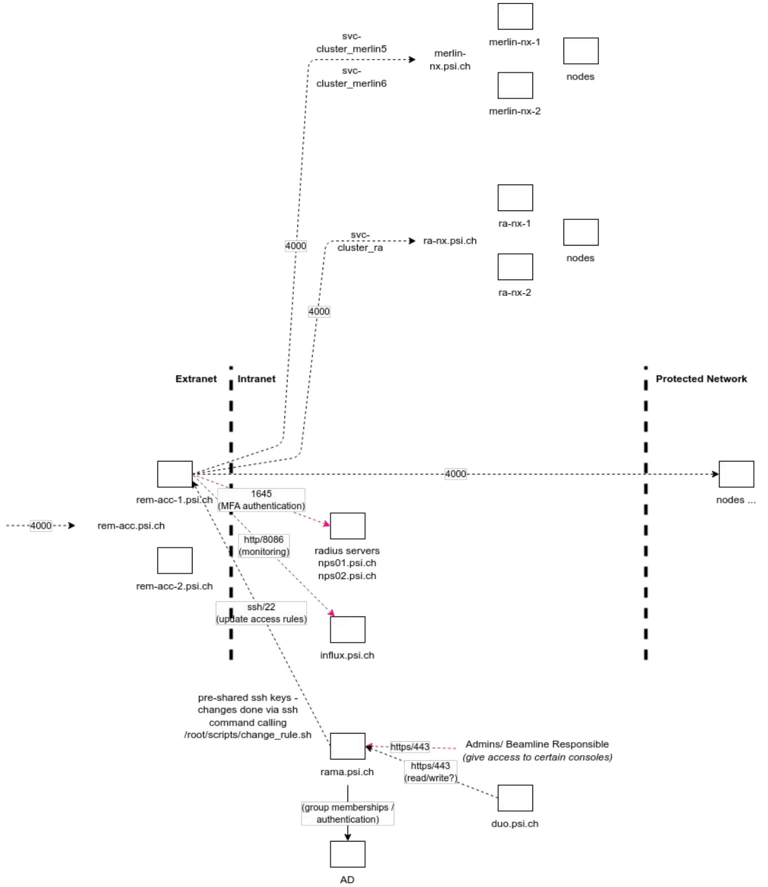
```
x10sa-cn-122.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-123.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-124.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-125.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-126.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-127.psi.ch        : ok=22   changed=1    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-128.psi.ch        : ok=22   changed=1    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-129.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-130.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-131.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-132.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-133.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
x10sa-cn-134.psi.ch        : ok=20   changed=0    unreachable=0    failed=0    skipped=22   rescued=0    ignored=0
xbl-daq-37.psi.ch          : ok=7    changed=0    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0
Thursday 01 December 2022  13:47:39 +0100 (0:00:02.449)       0:05:09.419 *****
===============================================================================
psi.adp --------------------------------------------------------------- 143.52s
psi.spotter ------------------------------------------------------------ 46.49s
stat ------------------------------------------------------------------- 31.57s
include_role ----------------------------------------------------------- 27.14s
psi.adm ---------------------------------------------------------------- 17.55s
psi.dimmer ------------------------------------------------------------- 9.59s
psi.jfjoch_writer ------------------------------------------------------ 5.76s
include_vars ----------------------------------------------------------- 5.20s
file ------------------------------------------------------------------- 4.25s
systemd ---------------------------------------------------------------- 4.18s
ansible.builtin.service_facts ------------------------------------------ 3.99s
set_fact --------------------------------------------------------------- 3.94s
gather_facts ----------------------------------------------------------- 3.86s
include_tasks ---------------------------------------------------------- 2.32s
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
total ----------------------------------------------------------------- 309.38s
Playbook run took 0 days, 0 hours, 5 minutes, 9 seconds
Cleaning up file based variables                                                                      00:00
Job succeeded
```

Remote access

We do rely on NoMachine for remote graphical access

Access to protected network secured with further level of filtering, managed by beamline staff

MFA enabled on every external-accessible system



Cloud strategy

PSI has a strategic partnership with CSCS and its ALPS cluster

- Use CSCS to cloud-burst SLS2 special cases
- see Alun's presentation

We do use commercial cloud providers for some external services

- Hetzner cloud for SciCat and SciLog
- we are exploring Azure for similar uses, as it could be supported by Central IT

Virtualization and containers

We extensively use VMs for services

- based on VmWare ESX cluster, managed by central IT

Containers are used mostly for services, some tests for user analysis (sarus / apptainer)

- we do lack a central container registry
- support is not still not well organized by Central IT

K8s is still a test technology at PSI

- we range from test vanilla k8s
- to test OKD clusters (mostly used for runners)
- to soon to production vanilla k8s for the ALPS project
- no central IT infrastructure nor support