

ASICLab IT Upgrades

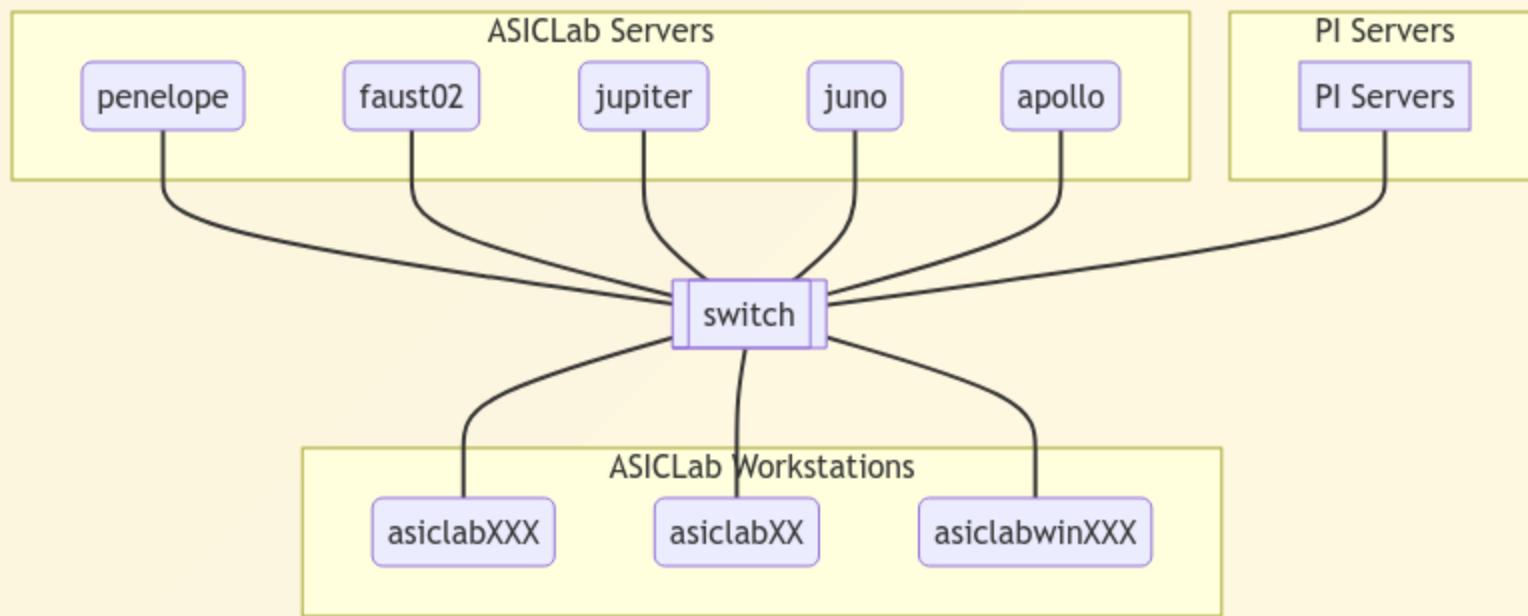


Kennedy Caisley

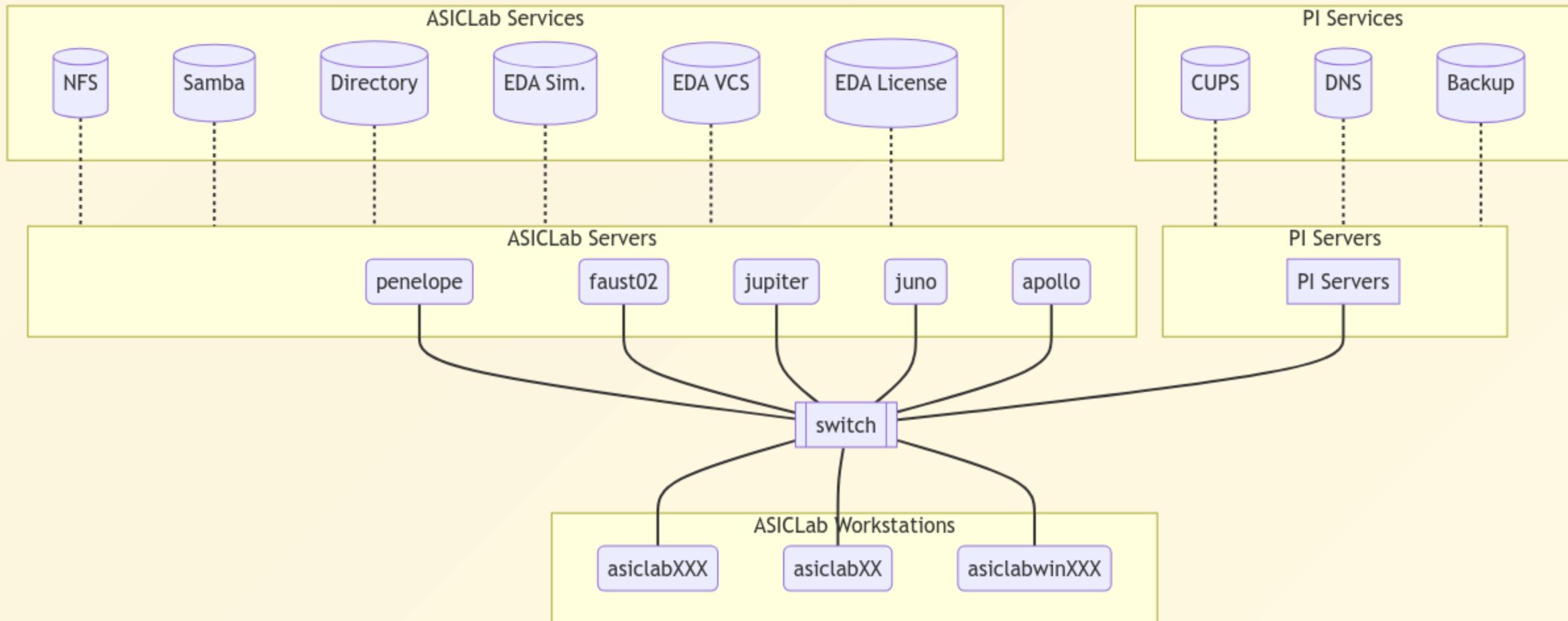
Marco Vogt

26 July 2023

The physical network...



...with services running on top



Motivations

- ✗ CentOS 7 reaching EOL with no upgrade path
- ✗ Drive failures & low storage in file server
- ✗ Workstations failing to boot & softwares outdated
- ✗ Config management opaque and slow
- ✗ Docs not maintain-ed | able

Project #1: An OS Upgrade

	Distribution	Design	Services	Desktop	Pricing	Future
RHEL						
Rocky/Alma						
SUSE						
OpenSUSE						
Ubuntu						
Fedora						

fedora, I choose you!

- ✓ Installed across 14 workstations and 2 servers
- OK Some EDA tools run; work-around for the rest
- ✓ Enterprise services are well-supported & documented (RHEL)
- ✓ Desktop apps mostly available (`zoom`, `slack`, `code`)
- ✓ Automatic driver & firmware upgrades
- ✓ Successful version updates since: **36 -> 37 -> 38 -> 39 (soon)**

Project #2: Workstation Setup

`clonezilla` isn't the right tool for *Configuration Management*:

“ The art of setting and maintaining a machine in the desired state. ”

`ansible` provides a simple, *idempotent* approach

1. `dnf install ansible` on a machine not being configured
2. `ssh-copy-id` to all target machines
3. List target machines in `inventory.yaml`:

```
workstations:  
hosts:  
    asiclab001.physik.uni-bonn.de:  
        mac: 54:BF:64:98:25:D4  
    asiclab002.physik.uni-bonn.de:  
        mac: 54:BF:64:98:25:CC  
    asiclab003.physik.uni-bonn.de:  
        mac: 54:BF:64:98:25:BAs
```

4. List desired state in `playbook.yaml`:

```
- name: Send a wake-on-LAN magic packet
  community.general.wakeonlan:
    mac: '{{ mac }}'
- name: Ensure client.conf exists & contains CUPS hostname
  ansible.builtin.lineinfile:
    path: /etc/cups/client.conf
    line: ServerName cups.physik.uni-bonn.de
    create: yes
- name: Check development tools are installed
  ansible.builtin.apt:
    name:
      - gcc
      - tmux
      - git-lfs
    state: latest
```

5. Run playbook on target inventory:

```
[asiclab@penelope ~]$ ansible -K playbook.yaml -i inventory.yaml
```

Some additional useful arguments:

- `--limit` to specific subset of machines
- Only run playbook tasks with certain `--tag`
- Be more `--verbose`

Ansible output log:

```
PLAY [Workstation Configuration] ****
TASK [Send a wake-on-LAN magic packet] ****
ok: [asiclab001.physik.uni-bonn.de]
ok: [asiclab002.physik.uni-bonn.de]
ok: [asiclab003.physik.uni-bonn.de]

TASK [Ensure client.conf exists & contains CUPS hostname] ****
ok: [asiclab001.physik.uni-bonn.de]
ok: [asiclab002.physik.uni-bonn.de]
changed: [asiclab003.physik.uni-bonn.de]

TASK [Check development tools are installed] ****
ok: [asiclab001.physik.uni-bonn.de]
changed: [asiclab002.physik.uni-bonn.de]
changed: [asiclab003.physik.uni-bonn.de]
```

Project #3: Fixing the File Server

1. Copied `/tools` and `/users` (~3 days)
2. Built `raid6` array with **5** new 16 TB drives:
 - Capacity: 48 TB
 - Speed gain: 3x read, but no write
 - Fault tolerance: 2-drive failure (double parity)
3. Copied back data (another ~3 days)

Raid array details, seen from penelope server

```
[asiclab@penelope ~]$ sudo mdadm --detail /dev/md127  
  
/dev/md127:  
      Creation Time : Sat Jan 14 14:46:26 2023  
      Raid Level   : raid6  
      Array Size   : 46877242368 (43.66 TiB 48.00 TB)  
Used Dev Size : 15625747456 (14.55 TiB 16.00 TB)  
      Raid Devices : 5  
      Chunk Size  : 512K  
  
Number  Major  Minor  RaidDevice State  
     0      8      16          0  active sync  /dev/sdb  
     1      8      32          1  active sync  /dev/sdc  
     2      8      48          2  active sync  /dev/sdd  
     3      8      64          3  active sync  /dev/sde  
     4      8      80          4  active sync  /dev/sdf
```

4. Enable new incremental backups with HRZ using IBM `dsmc`
5. Enable automatic array checks with `raid-check.timer`
6. Enable `nfs4` server for `/users` and `/tools`.

A sanity check from `asiclab001` workstation:

```
[asiclab@asiclab008 ~]$ showmount -e penelope.physik.uni-bonn.de
```

Export list for penelope.physik.uni-bonn.de:

```
/export/disk/tools    asiclab*, juno.physik.uni-bonn.de, noyce.physik.uni-bonn.de,  
jupiter.physik.uni-bonn.de, faust02.physik.uni-bonn.de, apollo.physik.uni-bonn.de
```

```
/export/disk/users    asiclab*, juno.physik.uni-bonn.de, noyce.physik.uni-bonn.de,  
jupiter.physik.uni-bonn.de, faust02.physik.uni-bonn.de, apollo.physik.uni-bonn.de
```

7. Use `ansible` to auto-mount NFS shares on all workstations:

```
- name: Create mount points and mount /users
  ansible.posix.mount:
    src: penelope.physik.uni-bonn.de:/export/disk/users
    path: /users
    opts: rw
    state: mounted
    fstype: nfs4
- name: Create mount points and mount /tools
  ansible.posix.mount:
    src: penelope.physik.uni-bonn.de:/export/disk/tools
    path: /tools
    opts: ro
    state: mounted
    fstype: nfs4
```

Project #4: Identity Management

[Under construction]

- User data on `rw` NFS share
- Ported old LDAP data to modern FreeIPA distribution (LDAP, SSSD, NSS)
- Data protected using groups `base`, `icdesign`, `tsmc65`, etc
- Downside: No IdM or NFS = no login or crashing

Don't just manually add users

```
freeipa command to list users
```

[Under construction]



Then on client client

```
SHOULD PUT THE ANSIBLE VERSION  
sudo realm join penelope.physik.uni-bonn.de
```

```
$ ls /home  
asiclab  
  
$ ls /users  
dschuechter  kcaisley  krueger  mvogt  skahn  szhang  ydieter  ...
```

Project #5: EDA Tools

[Under construction]

```
$ ls /tools
cadence  clio  containers  designs  kits  mentor  synopsys  xilinx ...
```

1. Create a `.def` file, for target application. Add the following:
2. Select a starting OS base image (Docker compatible!)

```
Bootstrap: docker
From: centos:7
```

3. Create mount points for external locations; `$HOME` is already done

```
%setup
mkdir ${APPTAINER_ROOTFS}/tools
mkdir ${APPTAINER_ROOTFS}/users
```

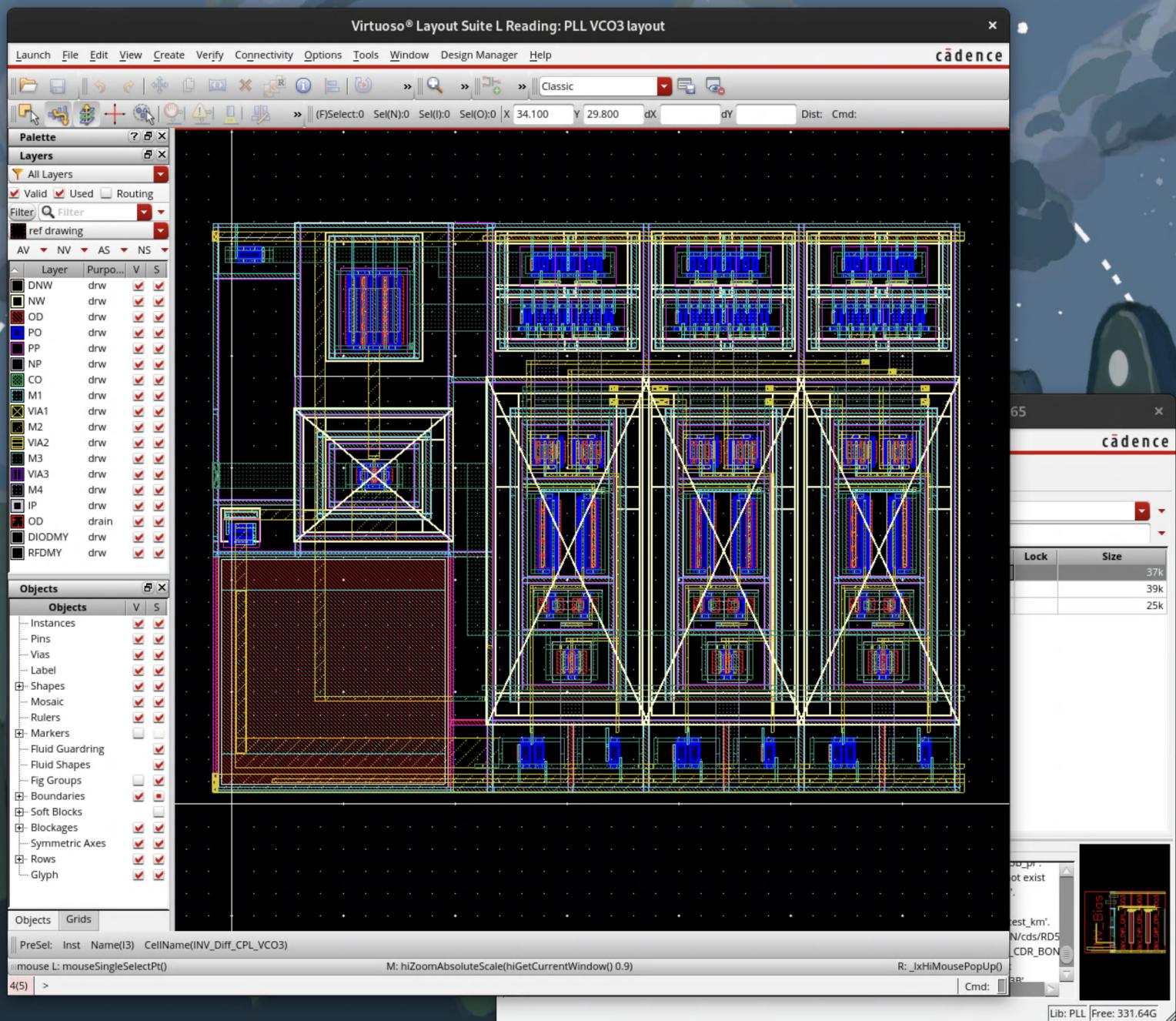
4. List packages to install on top of OS base image

```
%post
yum -y update && yum clean all
yum install -y csh tcsh glibc gdb
```

5. Compile the container image:

```
[kcaisley@asiclab008]$ sudo apptainer build demo.img demo.def

INFO: Starting build...
Getting image source signatures
INFO: Running setup scriptlet
INFO: Running post scriptlet
INFO: Adding environment to container
INFO: Creating SIF file...
Done!
```



Project #6: Write the docs

Decision making framework:

`confluence.atlassian.com` is 

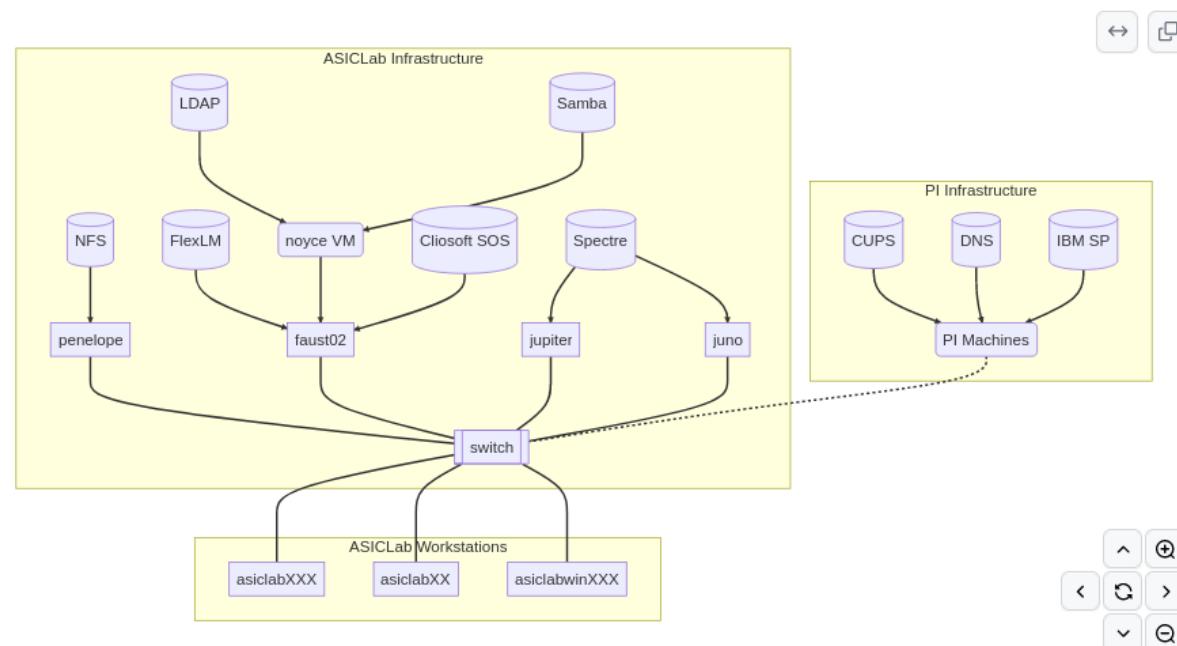
`.md + github.com` is 

Overview

The documentation covers ASICLab software, servers, workstations, and other infrastructure. The scope of this document is information that a user with basic familiarity with Linux systems would need to fix problems with existing systems or add new systems to the network. Therefore, extensive documentation of general Linux knowledge should be avoided to avoid diluting the usefulness of information.

Network Architecture

The follow diagram shows the various workstations and servers (i.e. infrastructure used in the lab, and the major services that run on the servers like NFS and LDAP. Additionally, some useful services like DNS and CUPS are inherited from the larger PI network, as seen on the right.



List of Machines and their Purpose

asiclab### : User Workstations

Configuration

1. Navigation in installation directory

```
cd /opt/tivoli/tsm/client/ba/bin
```

2. Create the file "dsm.opt" and add the configurations

```
sudo vi dsm.opt
```

Insert the following lines (Ctrl + Shift + V)

```
Servername tsm3.rhrz.uni-bonn.de
Domain all-local
Subdir yes
```

3. Create the file "dsm.sys" and insert the following configuration

```
sudo vi dsm.sys
```

Insert the following lines (Str + Shift + V) but replace [nodename] with the name assigned by HRZ

```
Servername tsm3.rhrz.uni-bonn.de
CommMethod tcpip
TCPPort 1500
TCPClientPort 1501
WEBPorts 1501,0
NODEname [nodename]
TCPServeraddress tsm2.rhrz.uni-bonn.de
PASSWORDAccess generate
INCLEXCL /opt/tivoli/tsm/client/ba/bin/dsm.excl_incl
SCHEDLOGNAME /var/log/tsm/dsmsched.log
ERRORLOGNAME /var/log/tsm/dsmerror.log
SCHEDLOGRETENTION 7 S
ERRORLOGRETENTION 7 S
schedmode prompted
managedservices schedule
```

4. Create Include/Exclude file (leave empty, or read more here)

```
sudo touch /opt/tivoli/tsm/client/ba/bin/dsm.excl_incl
```

5. Running the command line programme

```
sudo dsmc
```

6. Change password

```
sudo dsmc set password [old password] [new password]
```

Discussion

Summary

[Under construction]