

Deployment Details

InvenioRDM
Max Moser, Guillaume Viger

The possibilities are endless

You can host InvenioRDM productively...

- On a single machine
- Spanning multiple machines
- In a Kubernetes cluster
- ...

Hard to give out recommendations, because the “correct” answer depends on your use case

Initial Considerations

- Migrating from an existing repository or starting from scratch?
- What are **your** available resources?
 - How many full-time equivalents?
 - What are their skill sets?
- What are **your** known requirements?
 - Load / Intensity of usage?
 - Features?
- Don't forget about non-software considerations
 - University Policies, User engagement, Documentation ...
- Then consider InvenioRDM's [system requirements](#)

System requirements

Invenio can run in Docker, on virtual machines, or on physical machines. Invenio can run on a single machine or a cluster of 100s of machines.

It all depends on exactly how much data you are handling and your performance requirements.

Small installation:

- Web/app/background servers and Redis: 1 node
- Database: 1 node
- Elasticsearch: 1 node

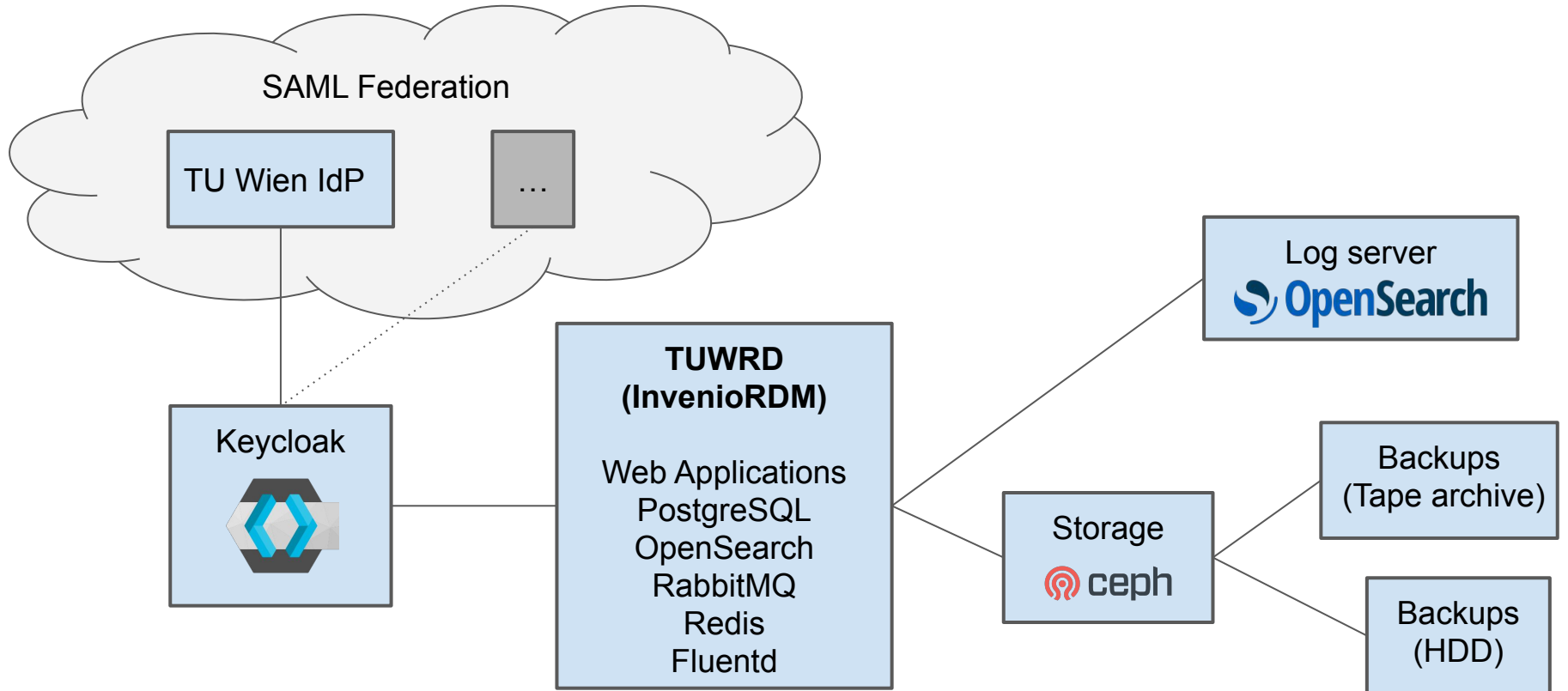
Medium installation:

- Load balancer: 1 node
- Web/app servers and background workers: 2 nodes
- Database: 1 node
- Opensearch: 3 nodes
- Redis/RabbitMQ: 1 node

Large installation:

- Load balancer: 2 node (with DNS load balancing)
- Web/app servers: 3+ nodes
- Background workers: 3+ nodes
- Database: 2 nodes (primary/replica)
- Opensearch: 5 nodes (3 data, 2 clients)
- Redis: 3 nodes (HA setup)
- RabbitMQ: 2 nodes (HA setup)

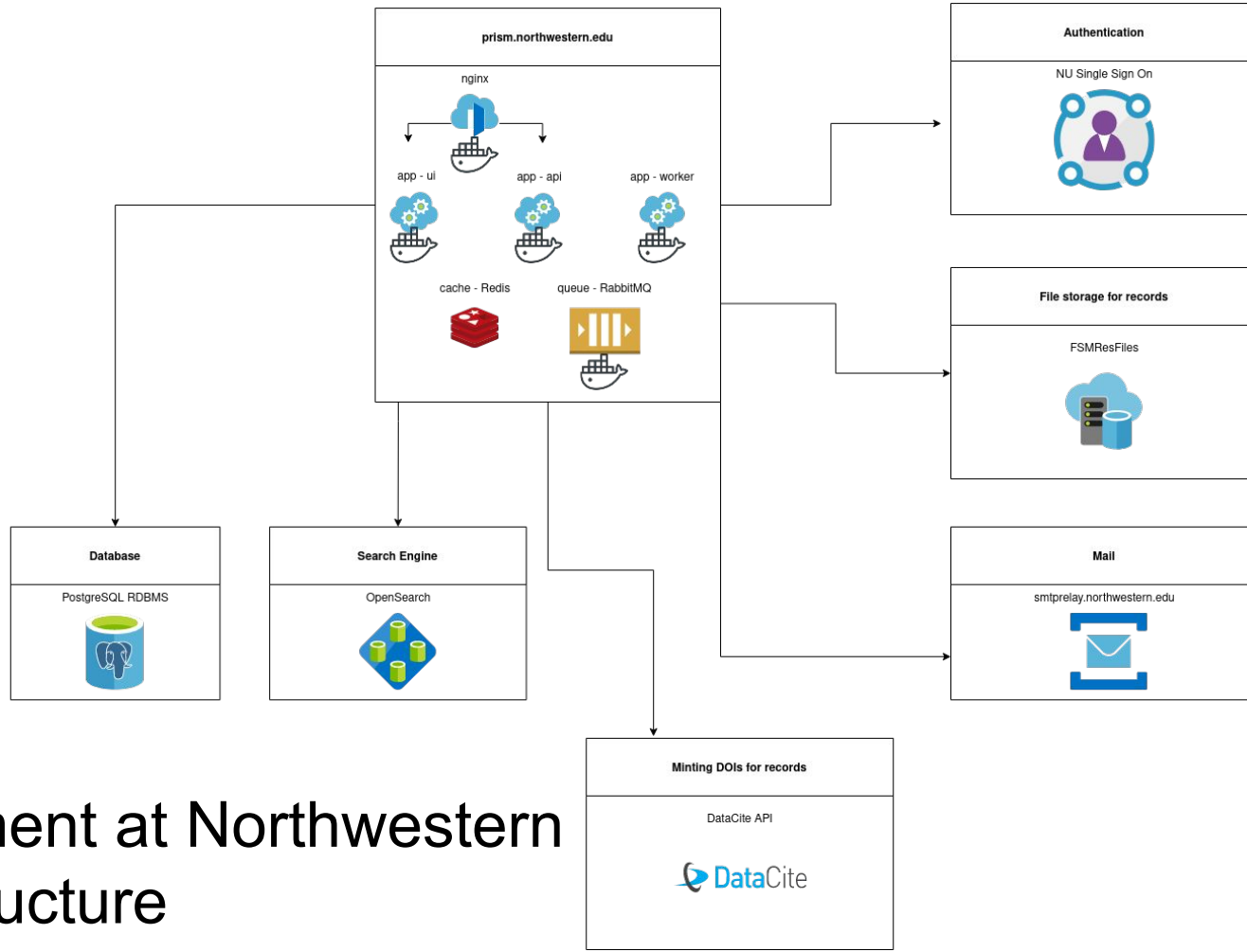
Deployment Overview at TU Wien



Deployment Pipeline at TU Wien

- New tag on our deployment sources triggers CI/CD pipeline
 - Container build (automatic trigger)
 - Our GitLab runner builds the container images
 - Pushes them to our internal container registry
 - Deploy staging (automatic trigger)
 - CI/CD variables are dumped to file and transferred to the host
 - SSH into the target machine
 - Check out the deployment sources & pull tagged images
 - Execute deployment script
 - Deploy production (manual trigger)
 - Same as above

Prism Production



Deployment at Northwestern
- infrastructure

Deployment at Northwestern - Ansible

```
playbooks > ! deploy-playbook.yml
1  ---|         You, 3 years ago * deployment: lay out deployment framework + copy s...
2  - hosts: "prism-app"
3
4  --vars_files:
5  --- vars.yml
6
7  --roles:
8  > --- - role: geerlingguy.docker ...
19
20 --tasks:
21
22 > --- - import_tasks: roles/git/main.yml ...
25
26 ---#
27 ---# Build application image
28 ---#
29 > --- - name: Build application docker image on prism-foundry ...
54
55 ---#
56 ---# Build nginx image
57 ---#
58 ---# We have to check out the src code on target machine too in order to build the nginx image
59 > --- - name: Prism branch "{{ deploy_branch | default('master') }}" is checked out on prism-app ...
68
69 ---# The .crt and .key files are already present on prism-app and updated by IT
70 > --- - name: nginx's docker image is built ...
80
81 ---#
82 ---# Transfer stage app image to production server
83 ---# Production specific
84 ---#
85 > --- - name: Transfer prism-foundry image to prism-app if not same machine ...
111
112 ---#
```

Future

At TU Wien

- Explore Ansible
- Keep existing connections alive during redeploy

At NU

- New Relic to SolarWinds
- GitHub actions for image building