

Rootless Docker in TSD

haatveit

University Center for Information Technology (USIT), University of Oslo

What?

Running containers without elevating privileges

Motivation

- Making containers more practical for
 - our **users**
 - us
- Improving security

Old TSD Docker setup & policy

- dockerd still running as root
- Daemon *default* setting runs containers in user namespace
- User namespaces not being a standardized range of non-production UIDs/GIDs
- User control over the daemon restricted by complex sudo ruleset for docker CLI

Rootless Docker in TSD

- `dockerd` running as non-root user(s) – usually a local system account
 - Allowed usage of user namespaces in high UID/GID ranges, which don't overlap with what's used outside of containers
 - Project users don't end up with more privileges or access than they are supposed to

Rootless Docker in TSD

- Users control dockerd via authenticated TCP
 - mTLS for daemon access
 - Non-restrictive usage of docker CLI
 - Tools like Compose just work™
- Granular access possible via client certificate access control

Non-root containers: challenges

- NFS doesn't understand user namespaces
 - Local VM storage required
- Namespaces have to be set up on a per-user basis
 - Hence the sharing of an unprivileged system user
- ESS not yet tested

Future: podman

Podman is a container engine from Red Hat.

- Supported on RHEL
- Actively targeting the non-root use case
- As of podman 3.0.0 from last week, now has Docker API compatibility