Web application hosting with Openshift, and Docker images

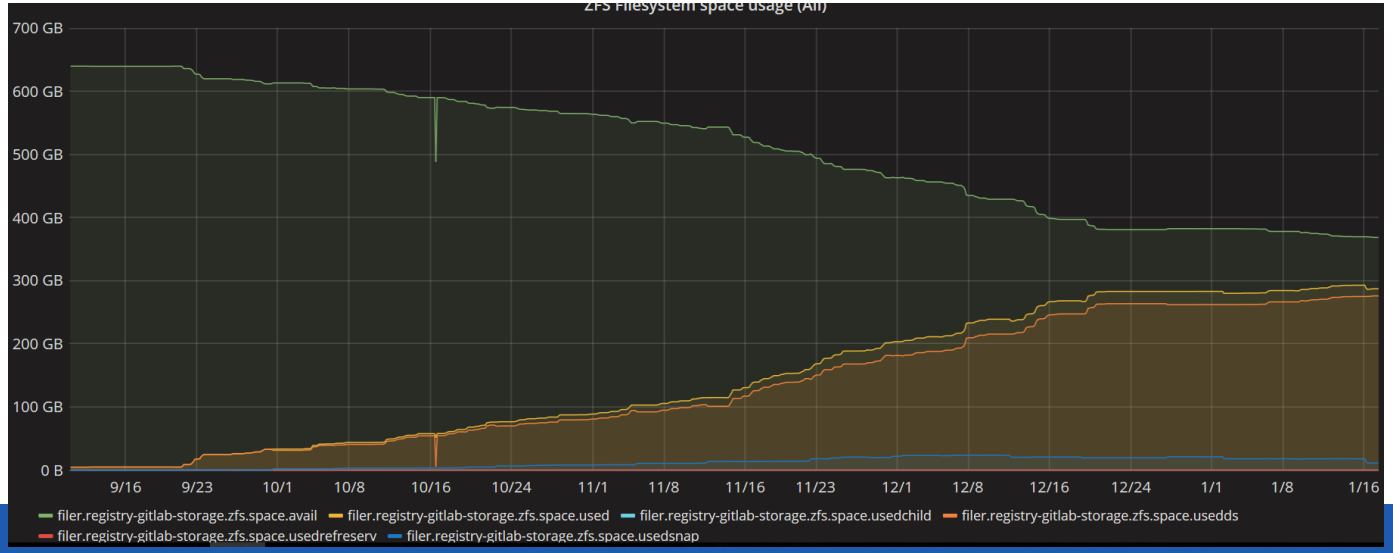
Alex Lossent – IT-CDA-WF

Outline

- Docker images
  - Image build
  - Registry
- Web Application Hosting with Openshift
  - Web hosting overview
  - Use cases
    - Central service hosting
    - Application templates
    - Custom user applications
  - Architecture

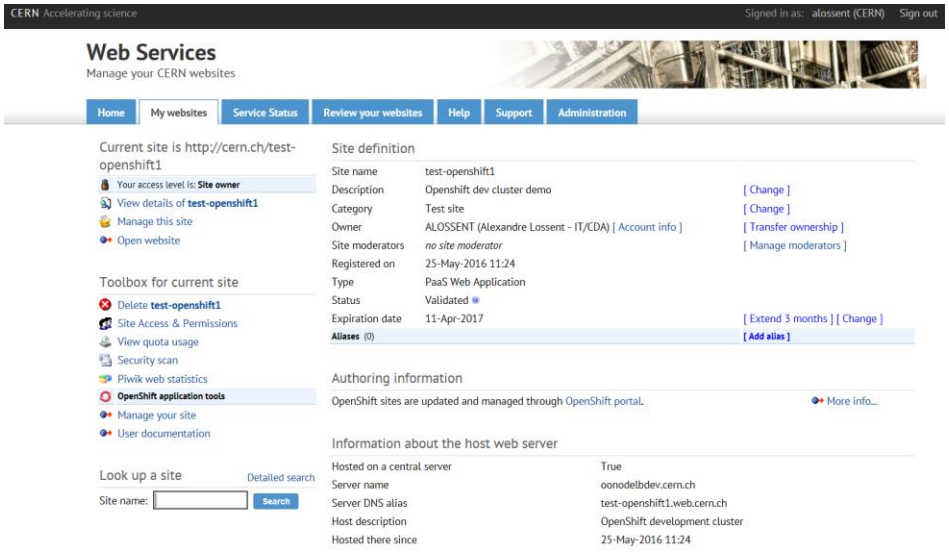
Docker images

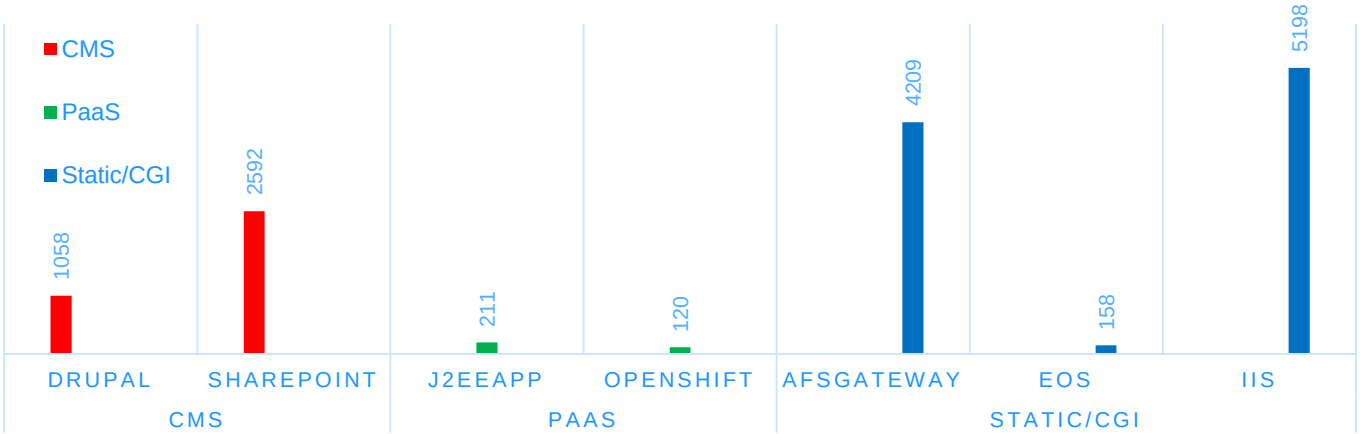
- Image build automation with GitLab-CI
  - Dedicated *runner* tag
  - Custom Docker image with access to Docker daemon to run ‘docker build’
    - User-provided script ignored
- Docker registry: GitLab Container Registry
  - Common hierarchy and permissions with GitLab projects

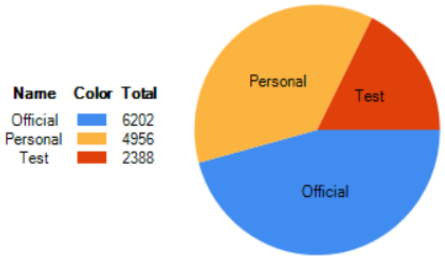


Web Hosting: overview

- CERN Web Services
  - Hosting 13000 “central” sites







Web Hosting

- Common features for all “central” sites
  - Ownership/lifecycle management (FIM)
  - SSO, visibility (Intranet/Internet)
  - Review & (un)block by security team
- Hundreds of independent web servers
  - Need for a library/framework version not provided in “central” sites
  - Large web applications
  - How to reduce the need for such servers?

Openshift: use cases

- Initial motivation: PaaS for Jenkins instances
  - Evaluated Openshift v2, then v3 end 2014
- Expanded scope
  - Platform to host “central” services (from CDA)

  - Provide CERN users with self-service application templates

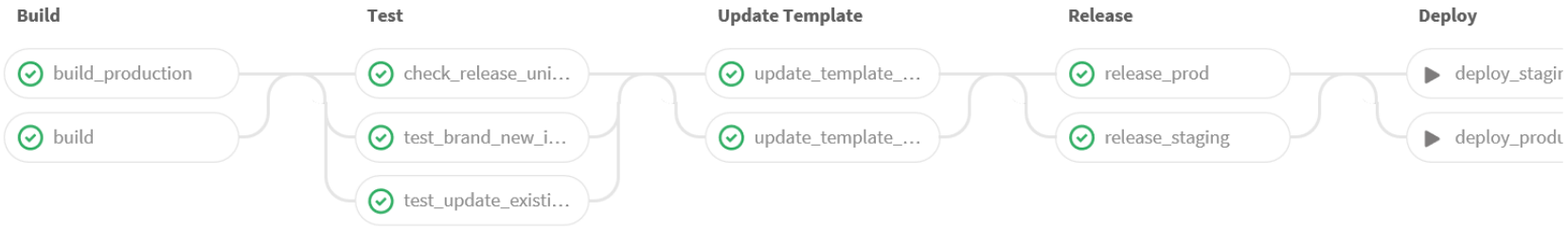


  - Increase flexibility of central web hosting and support custom web apps without dedicated VMs

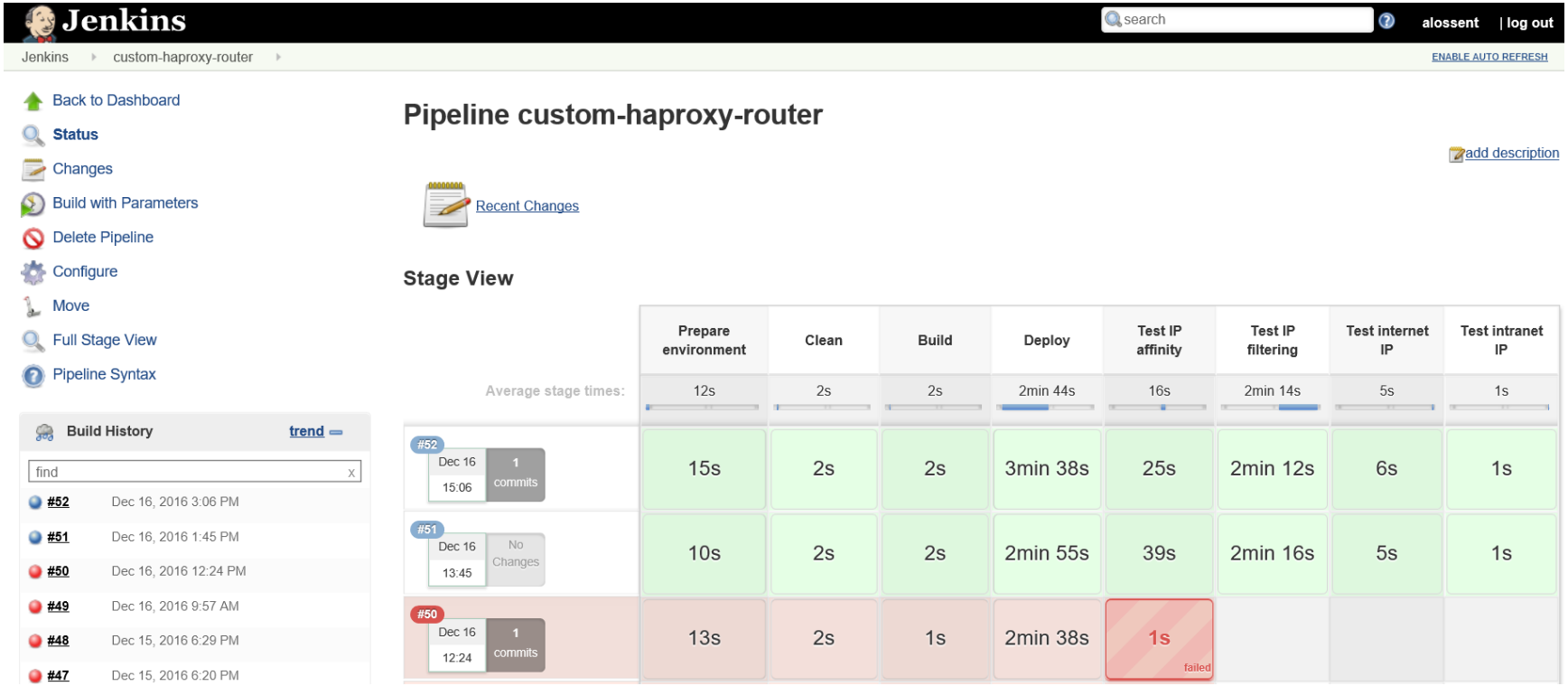


Openshift as a tool for developers

- Facilitate deployment and operation of web applications:
  - Getting started with a web application/prototype
  - Automate application deployment, rollback changes
  - No need to maintain a VM and its OS
  - Switch hosting platform (container portability)
- Good integration with code hosting (GitLab)
  - CI/CD pipelines (GitLab/Jenkins)
  - GitLab Review apps



Openshift CI example



Openshift: architecture

- Openshift Origin 1.3
  - Puppet-managed VMs + BYO Ansible playbook
- HA setup (masters and routers)
- Prod cluster: 5 large worker node VMs
  - sufficient for 33 Jenkins instances + CDA apps
- Integrated with CERN environment
  - Web Services/FIM: project lifecycle, DNS management
  - Authentication: SSO (SAML), LDAP, Kerberos
  - Storage: NFS, EOS, CVMFS

Openshift: integration

- Internal Python app to:
  - Implement Web Services API for project lifecycle and security (visibility, blocking…)
  - Customize autogeneration of DNS names on routes
  - Automate SSO (SAML) registration
  - Provision NFS volumes from the NFS Filer service
- Less customization needed as Openshift evolves
  - E.g. Volume classes in 1.4

Openshift: integration

- SDN: use IPSec to protect internal network traffic
- Kubernetes Flexvolume drivers for EOS, CVMFS (from cloud team)
- HAProxy routers:
  - IP affinity, IP filtering
  - Integration with DNS Load Balancing

Outlook

- Strategy: centralize web hosting on Openshift
  - Route all traffic for “central” web hosting via Openshift HAProxy routers
  - Move static/CGI web servers to containers
    - Including IIS when Windows containers possible
  - Containerize applications currently on VMs
    - Within CDA, and enable it for all CERN users

Questions?