Managing build infrastructure at ALICE using Hashicorp Nomad

Computing in High Energy Physics 2023, Norfolk, VA

Timo Wilken[1] [2] Giulio Eulisse[2]

9 May 2023

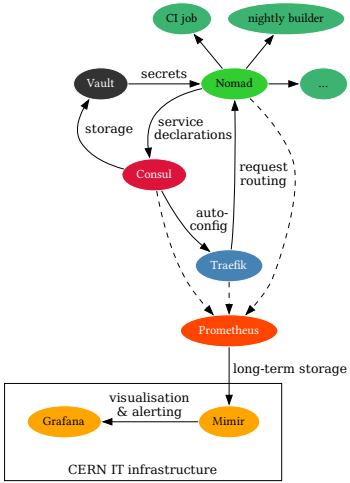
[1]E-mail: timo.wilken@cern.ch
[2]for the ALICE Collaboration

What do we build?

- O^2: software suite for data-taking, physics analysis, Monte-Carlo simulation, …
  - split across multiple repositories: O^2, O^2Physics, QualityControl and more
- Run 2 software still maintained for analysing old data
- alidist: pseudo-distribution of O^2 dependencies
  - designed to function on top of recent versions of CentOS, Alma, Ubuntu, MacOS, …
- nightly release builds, continuous integration (CI) compilation checks, unit and integration tests
- 1 non-trivial CI check completed every 2 minutes, on average
  - plus lots of fast rebuilds where nothing has changed
- …all on 5+ platforms (mostly) through containerization

Architecture overview

- resources: 600 CPU cores + 1.7 TiB memory
- Nomad, Consul, Vault from Hashicorp, designed to complement each other
- Nomad: allocates jobs to machines; resource accounting
  - long-running jobs: release builders, custom CI
  - web services: user account admin, tarball servers
  - scheduled jobs: repository maintenance and cleanup
- Consul: generic key/value store and DNS
  - job discovery: `*.service.consul` DNS
  - Traefik auto-config for web access
  - job monitoring: simple health checks
- Vault stores secrets, using Consul as backend
- metrics of the whole cluster stored and visualised



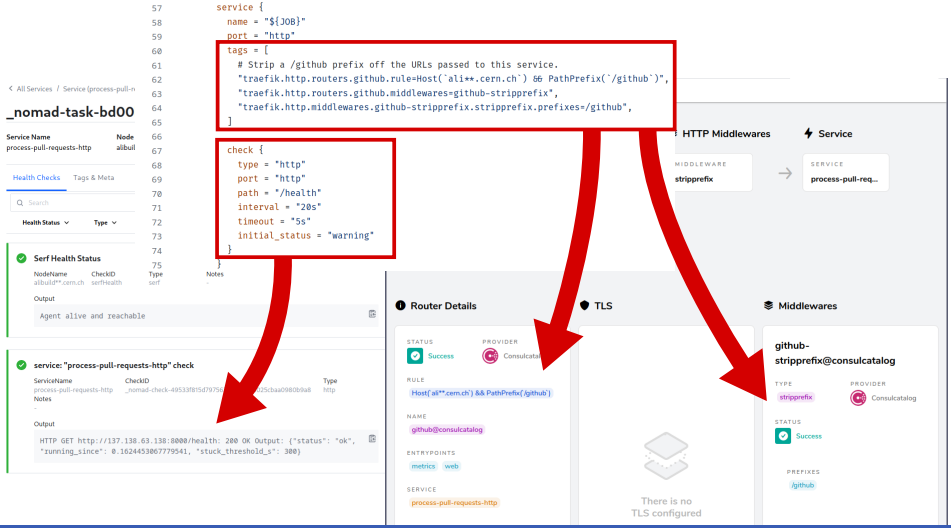
Reasons for switching away from Mesos and Aurora

- previous stack: Mesos + Marathon + Aurora
- Aurora not intensively developed any more
- requires Python 2 (EOL since 2020) on server and developers' machines
  - difficult to install, deploy and maintain
- some features difficult to integrate with or nonexistent
  - autoscaling (or even manual scaling without restarts of all jobs)
  - difficult to keep build caches "hot"
  - little monitoring and alerting integration

Improvements with Nomad + Consul + Vault

- simple deployment: static binary + systemd/launchd service + configuration = 3 files
- first-class support for web services: health checks, autoconfiguration
- more secure secrets management
- excellent monitoring & alerting support through Prometheus
  - resource use statistics (CPU, memory, disk)
  - alerts when build machines are unavailable or have problems
- …more features, for deeper future integration

Web services: health checks & Traefik autoconfiguration



Monitoring example: nightly build performance

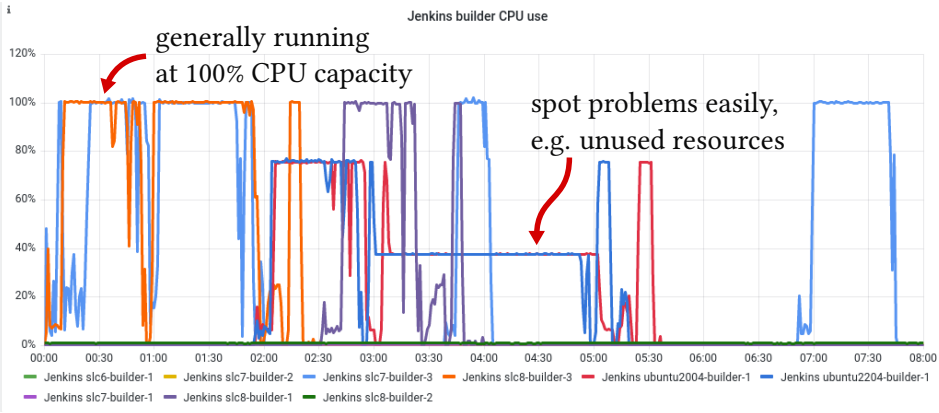


Figure: CPU use of a sequence of nightly builds as a fraction of total allocated CPU resources (usually the entire VM).

Monitoring example: nightly build performance

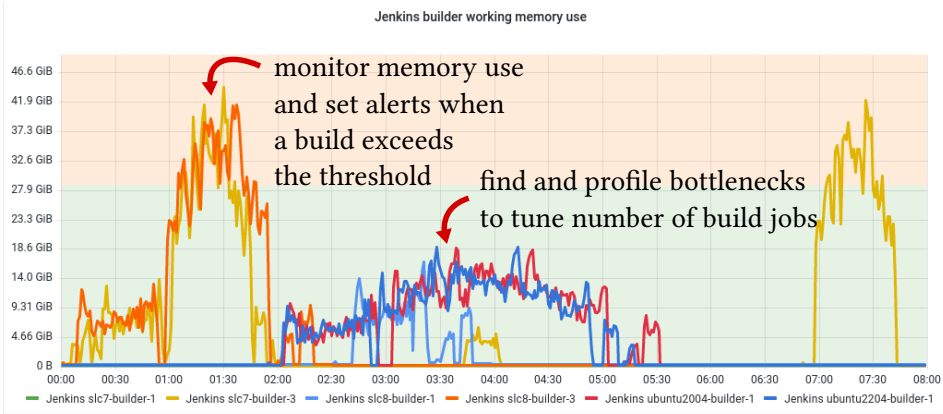


Figure: Working memory (RSS) use of a sequence of nightly builds. Total available memory on a typical build VM in green.

Rough edges

1. Nomad's handling of disk space allocation
   - restarting daemon with non-empty disk confuses Nomad's accounting
   - can cause scheduling issues much further down the line
   - must manually clean up the node and restart the Nomad agent process
2. integration with CERN single sign-on
   - by default: token authentication with Nomad/Consul/Vault
   - could integrate SSO with Vault, which would then issue Nomad/Consul tokens

Future work integrating build infra with Nomad

- "true" autoscaling, based on real-time demand
  - manual scaling already much smoother than previously: build caches are kept most of the time, existing builders uninterrupted
  - remaining challenge is cache invalidation: scaling often invalidates multiple gigabytes of cached builds
- temporary configuration (e.g. for testing software deployment) through Consul instead of text files
- get build secrets from Vault only when needed, instead of storing them in env variables and relying on sanitisation during build