Next Generation CI

Building Eclipse projects on the new CBI build farm

Frederic Gurr
Eclipse Foundation

EclipseCon Europe - October 25, 2018
History & Status quo
History

**2009**
- Single Hudson instance
  - Maintained by the community

**2010**
- 1 Master, 2 Slaves
  - Maintained by Eclipse IT team

**2011-2012**
- More slaves
  - Dedicated master for performance tests

**2013**
- Hudson instance per project
  - Part of the CBI initiative
  - Shared instance with slaves for Windows and MacOS
History

2016

Hudson ecosystem not catching up

Started to evaluate Jenkins

2017

Mass migration to Jenkins

Migration tool https://github.com/eclipse-cbi/hipp2jipp

Custom scripts

2018

Jenkins instance per project

No more Hudson
Today’s infrastructure

• 220+ Jenkins instances

• Beefy machines (total ~250 cores, ~1.5TB RAM)

• A few dedicated slaves (e.g. for Windows and Mac)
Today’s infrastructure

• Managing 10s of Jenkins is “easy”, managing 100s of them is time consuming
• We want to spend our time on areas that add value:
  • Harden digital signature and packaging services
  • Ease publication/promotion (Maven central, Nexus, ...)
  • Better support of static analysis tools (SonarQube, ...)
  • Better integration with GitHub
Today’s infrastructure

• We also want to offer better performance
  • Pool our hardware to use it more efficiently
  • QoS - better isolation and control over resources usage
  • Be able to add additional resources more easily
    • Either cloud-based or on-premise
Next generation CI
A duet solution

RED HAT OPENSSHIFT
Container Platform

cloudbees

(kubernetes)
CloudBees Core

Operation Center

Kubernetes/OpenShift Cluster

JIPPS

Jobs run on temporary, dynamically provisioned agents
CloudBees Core

Operation Center

Dedicated agents

JIPPS

Kubernetes/OpenShift Cluster

Jobs run on temporary, dynamically provisioned agents
CloudBees Core

Shared agents

Operation Center

Dedicated agents

JIPPS

Kubernetes/OpenShift Cluster

Jobs run on temporary, dynamically provisioned agents
Next generation status quo

Already running on new infra:
• Every new provisioned JIPP
• Early adaptors like Xtext
• ~40 Jakarta EE projects
CBC and cluster problems

- We had some stability issues in the beginning
- Responsive and excellent support by CloudBees
- Cluster is now much more stable
Migrating OSS Jenkins instances to CBC

• How?
  • No built-in solution for mass migration of jobs
  • => Custom solution
• What needs to be moved?
  • Job configurations
  • Views
  • Credentials
  • Maven settings
Roadmap

**Setup**

OpenShift/CJE running on Foundation infrastructure

**Early adopters**

New JIPPs are created on CJE and a couple of volunteer projects are migrated

**JakartaEE**

All JakartaEE projects run on CJE

**Migration**

All JIPPs are migrated to CJE
FAQ & Best practices
FAQ

- Why does my build take so long and how can I make it faster?
FAQ

• Why does my build take so long and how can I make it faster?
• What are Jenkins pipelines and why/when should they be used?
Jenkins Pipelines

- Store your job config in your repo
- Config linked to code
- Config history
- Scripted Pipeline vs Declarative Pipeline
- Recommended for docker based builds
FAQ

• Why does my build take so long and how can I make it faster?
• What are Jenkins pipelines and why/when should they be used?
• Can I finally run docker based builds?
FAQ

• Why does my build take so long and how can I make it faster?
• What are Jenkins pipelines and why/when should they be used?
• Can I finally run docker based builds?
• What do I need to do for the migration?
Best practices

• Modular builds and tests => Parallelization
• Play fair and don’t be a resource hog
• Build configuration in Code => Pipelines
• Docker

More:  http://wiki.eclipse.org/CI_best_practices
       https://wiki.eclipse.org/Jenkins#FAQ
Questions?
Takeaways

• Less administration => more tools and services
• New infra => better utilization and scaling
• Stateless build agents will require adaptations
Thank you!

Please evaluate the talks on eclipsecon.org!