Next Generation CI

Building Eclipse projects on the new CBI build farm

Frederic Gurr
Eclipse Foundation

EclipseCon Europe - October 25, 2018

History & Status quo

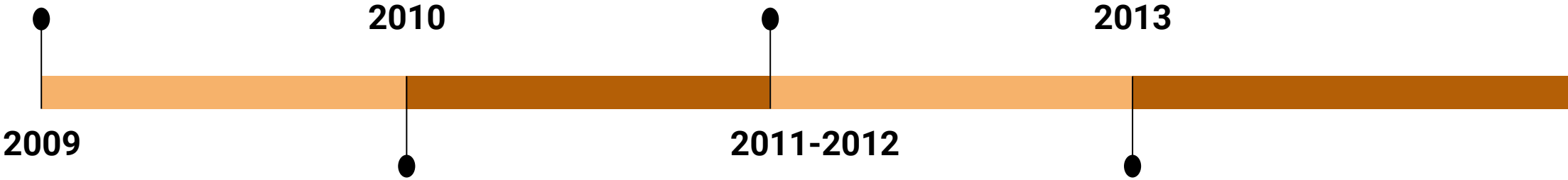
History

Single Hudson instance

Maintained by the community

More slaves

Dedicated master for performance tests



1 Master, 2 Slaves

Maintained by Eclipse IT team.

Hudson instance per project

Part of the CBI initiative

Shared instance with slaves for Windows and MacOS

History

Hudson ecosystem not catching up

Started to evaluate Jenkins

2016

2017

Mass migration to Jenkins

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

Custom scripts

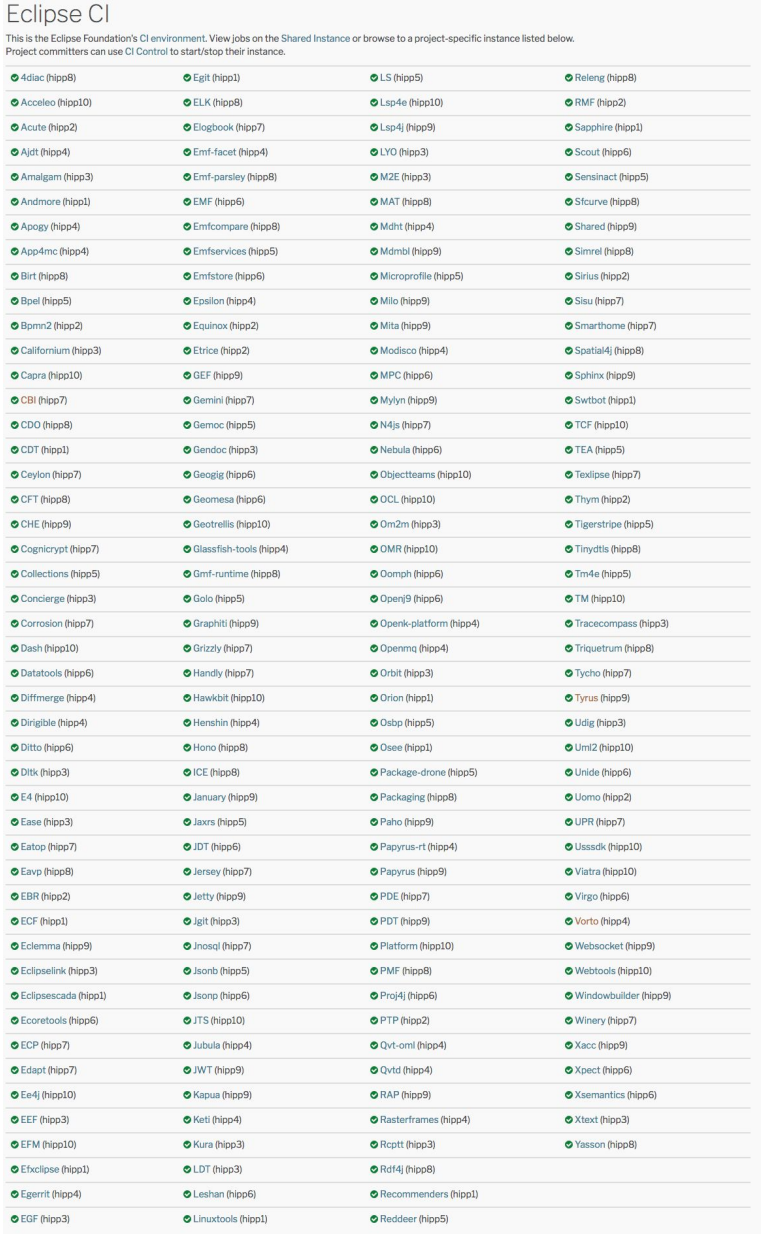
2018

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)



Eclipse CI

This is the Eclipse Foundation's CI environment. View jobs on the Shared Instance or browse to a project-specific instance listed below.
Project committers can use CI Control to start/stop their instance.

4diac (hipp8)	Egit (hipp1)	LS (hipp5)	Releng (hipp8)
Acceleo (hipp10)	ELK (hipp8)	Lsp4e (hipp10)	RMF (hipp2)
Acute (hipp2)	Elogbook (hipp7)	Lsp4j (hipp9)	Sapphire (hipp1)
Ajdt (hipp4)	Emf-facet (hipp4)	LYO (hipp3)	Scout (hipp6)
Amalgam (hipp3)	Emf-parsley (hipp8)	M2E (hipp3)	Sensinact (hipp5)
Andmore (hipp1)	EMF (hipp6)	MAT (hipp8)	Sfcurve (hipp8)
Apogy (hipp4)	Emfcompare (hipp8)	Mdht (hipp4)	Shared (hipp9)
App4mc (hipp4)	Emfservices (hipp5)	Mdmbl (hipp9)	Simrel (hipp8)
Birt (hipp8)	Emfstore (hipp6)	Microprofile (hipp5)	Sirius (hipp2)
Bpel (hipp5)	Epsilon (hipp4)	Milo (hipp9)	Sisu (hipp7)
Bpmn2 (hipp2)	Equinox (hipp2)	Mita (hipp9)	Smarthome (hipp7)
Californium (hipp3)	Etrice (hipp2)	Modisco (hipp4)	Spatial4j (hipp8)
Capra (hipp10)	GEF (hipp9)	MPC (hipp6)	Sphinx (hipp9)
CBI (hipp7)	Gemini (hipp7)	Mylyn (hipp9)	Swtbot (hipp1)
CDO (hipp8)	Gemoc (hipp5)	N4js (hipp7)	TCF (hipp10)
CDT (hipp1)	Gendoc (hipp3)	Nebula (hipp6)	TEA (hipp5)
Ceylon (hipp7)	Geogig (hipp6)	Objectteams (hipp10)	Texlipse (hipp7)
CFT (hipp8)	Geomesa (hipp6)	OCL (hipp10)	Thym (hipp2)
CHE (hipp9)	Geotrellis (hipp10)	Om2m (hipp3)	Tigerstripe (hipp5)
Cognicrypt (hipp7)	Glassfish-tools (hipp4)	OMR (hipp10)	Tinydtls (hipp8)
Collections (hipp5)	Gmf-runtime (hipp8)	Oomph (hipp6)	Tm4e (hipp5)
Concierge (hipp3)	Golo (hipp5)	Openj9 (hipp6)	TM (hipp10)
Corrosion (hipp7)	Graphiti (hipp9)	Openk-platform (hipp4)	Tracecompass (hipp3)
Dash (hipp10)	Grizzly (hipp7)	Openmq (hipp4)	Triquetrum (hipp8)
Datatools (hipp6)	Handly (hipp7)	Orbit (hipp3)	Tycho (hipp7)
Diffmerge (hipp4)	Hawkbit (hipp10)	Orion (hipp1)	Tyrus (hipp9)
Dirigible (hipp4)	Henshin (hipp4)	Osbp (hipp5)	Udig (hipp3)
Ditto (hipp6)	Hono (hipp8)	Osee (hipp1)	Uml2 (hipp10)
Ditk (hipp3)	ICE (hipp8)	Package-drone (hipp5)	Unide (hipp6)
E4 (hipp10)	January (hipp9)	Packaging (hipp8)	Uomo (hipp2)
Ease (hipp3)	Jaxrs (hipp5)	Paho (hipp9)	UPR (hipp7)
Eatop (hipp7)	JDT (hipp6)	Papyrus-rt (hipp4)	Usssdk (hipp10)
Eavp (hipp8)	Jersey (hipp7)	Papyrus (hipp9)	Viatra (hipp10)
EBR (hipp2)	Jetty (hipp9)	PDE (hipp7)	Virgo (hipp6)
ECF (hipp1)	Jgit (hipp3)	PDT (hipp9)	Vorto (hipp4)
Eclemma (hipp9)	Jnosql (hipp7)	Platform (hipp10)	Websocket (hipp9)
Eclipselink (hipp3)	Jsonb (hipp5)	PMF (hipp8)	Webtools (hipp10)
Eclipsescada (hipp1)	Jsonp (hipp6)	Proj4j (hipp6)	Windowbuilder (hipp9)
Ecoretools (hipp6)	JTS (hipp10)	PTP (hipp2)	Winery (hipp7)
ECP (hipp7)	Jubula (hipp4)	Qvt-oml (hipp4)	Xacc (hipp9)
Edapt (hipp7)	JWT (hipp9)	Qvtd (hipp4)	Xpect (hipp6)
Ee4j (hipp10)	Kapua (hipp9)	RAP (hipp9)	Xsemantics (hipp6)
EEF (hipp3)	Keti (hipp4)	Rasterframes (hipp4)	Xtext (hipp3)
EFM (hipp10)	Kura (hipp3)	Rcptt (hipp3)	Yasson (hipp8)
Efxclipse (hipp1)	LDT (hipp3)	Rdf4j (hipp8)	
Egerrit (hipp4)	Leshan (hipp6)	Recommenders (hipp1)	
EGF (hipp3)	Linuxtools (hipp1)	Reddeer (hipp5)	

https://ci.eclipse.org

Today's infrastructure

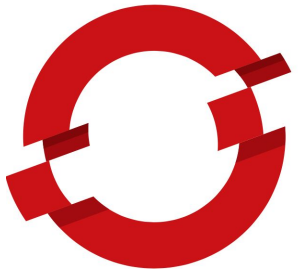
- Managing 10s of Jenkins is "easy", managing 100s of them is time consuming
- We want to spend our time on areas that add value:
  - Harden digital signature and packaging services
  - Ease publication/promotion (Maven central, Nexus, ...)
  - Better support of static analysis tools (SonarQube, ...)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

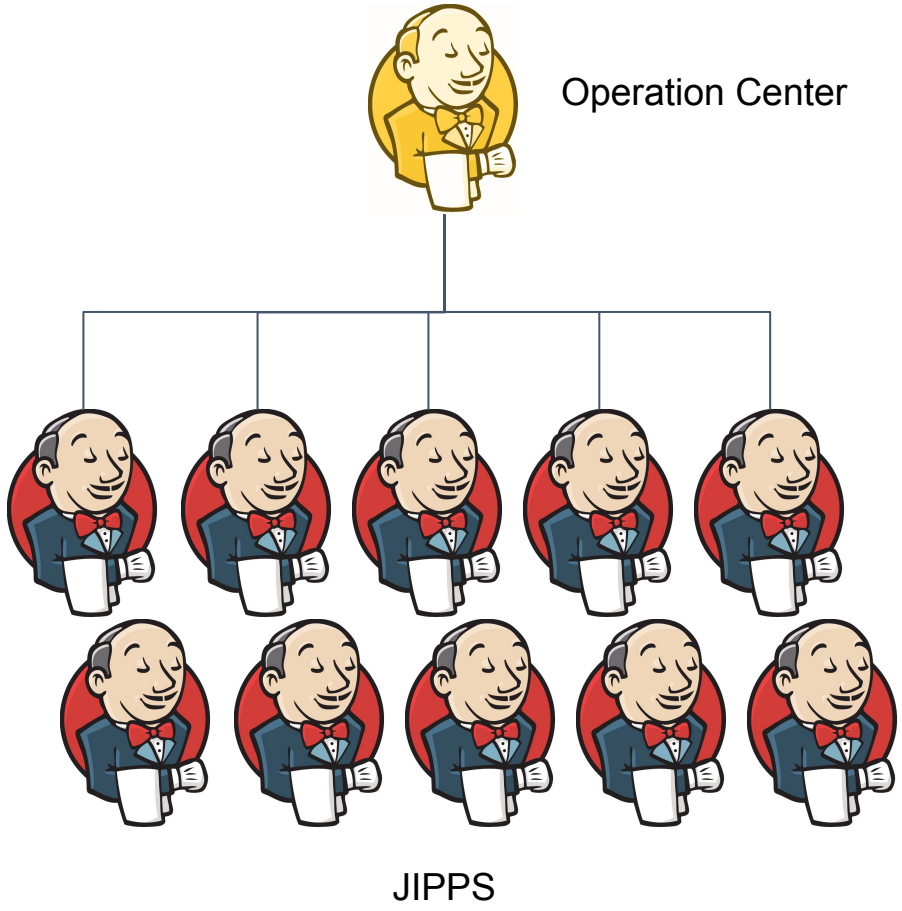


RED HAT®
OPENSHIFT
Container Platform

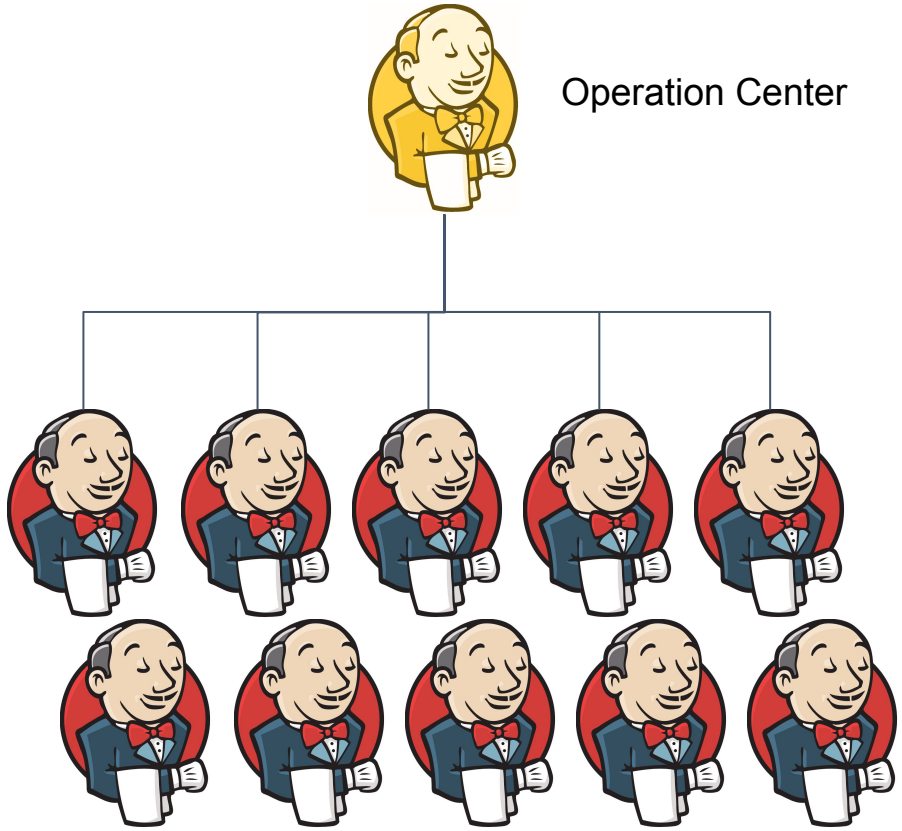
(kubernetes)

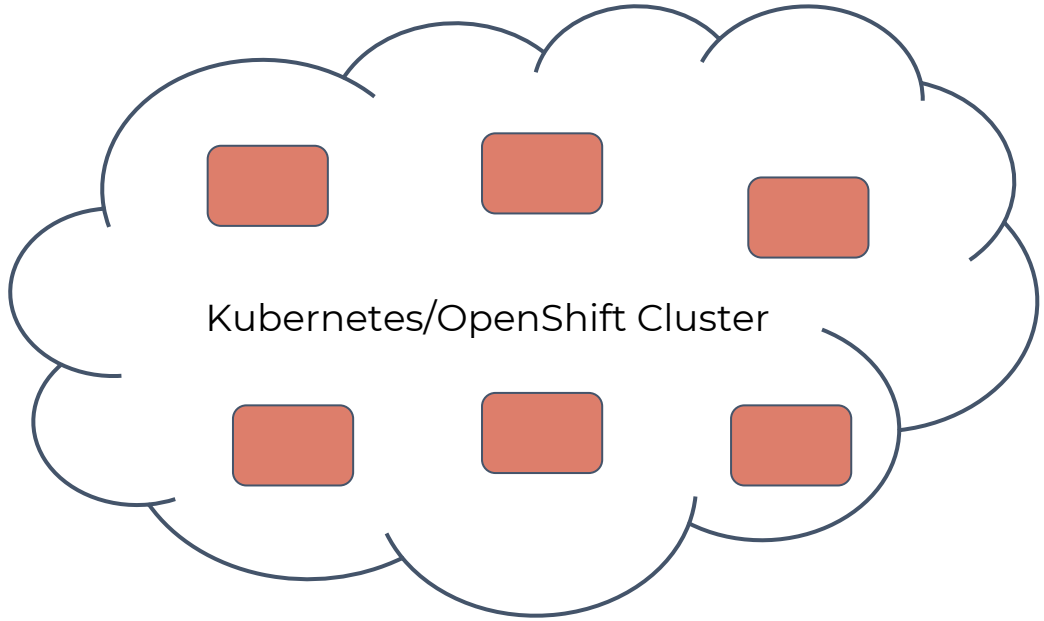
cloudbees®

CloudBees Core



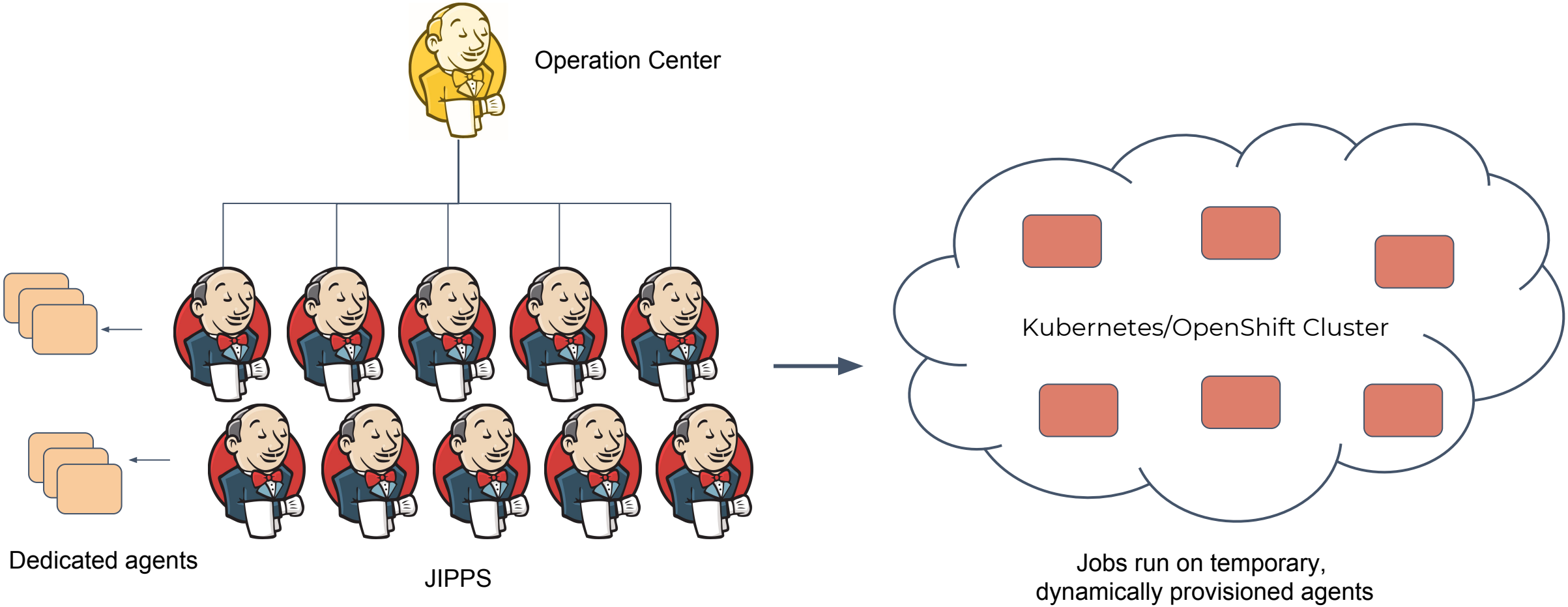
CloudBees Core



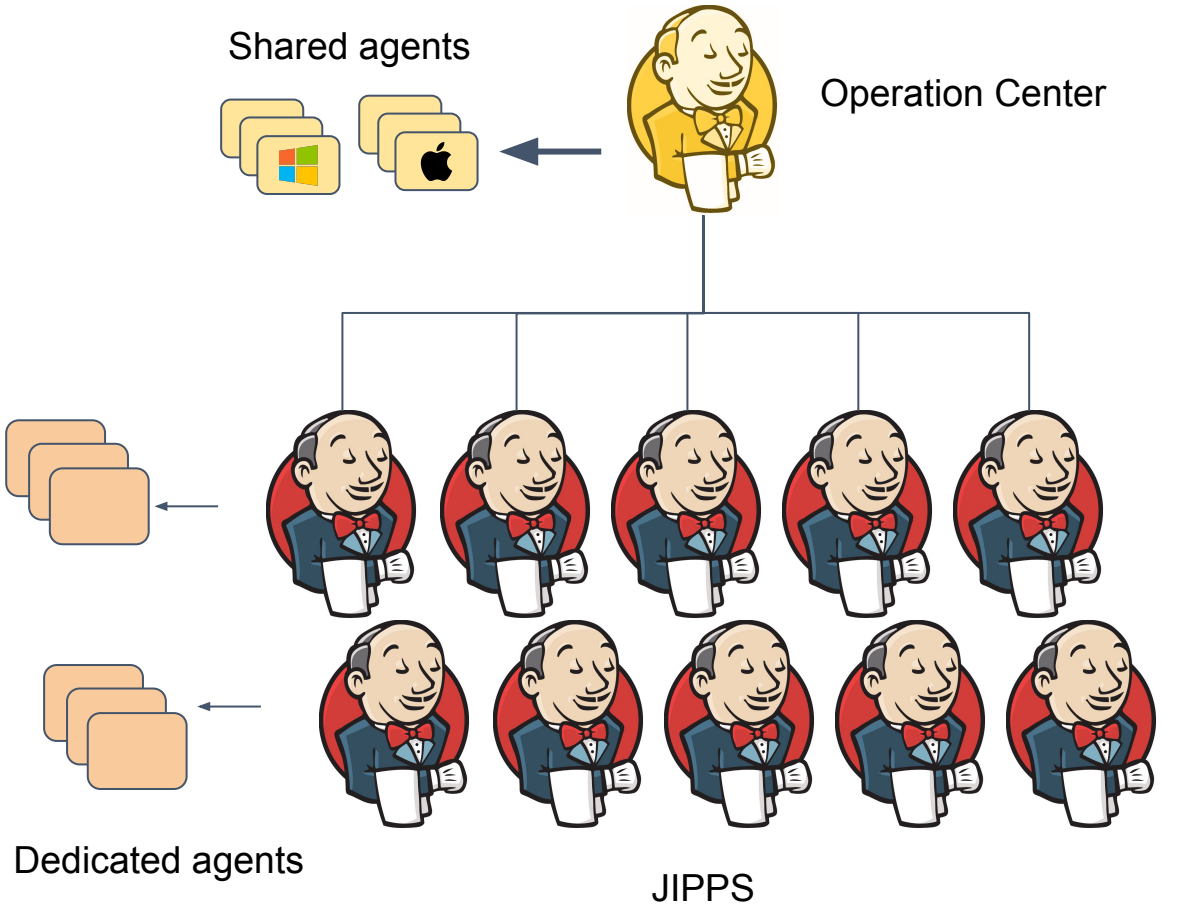


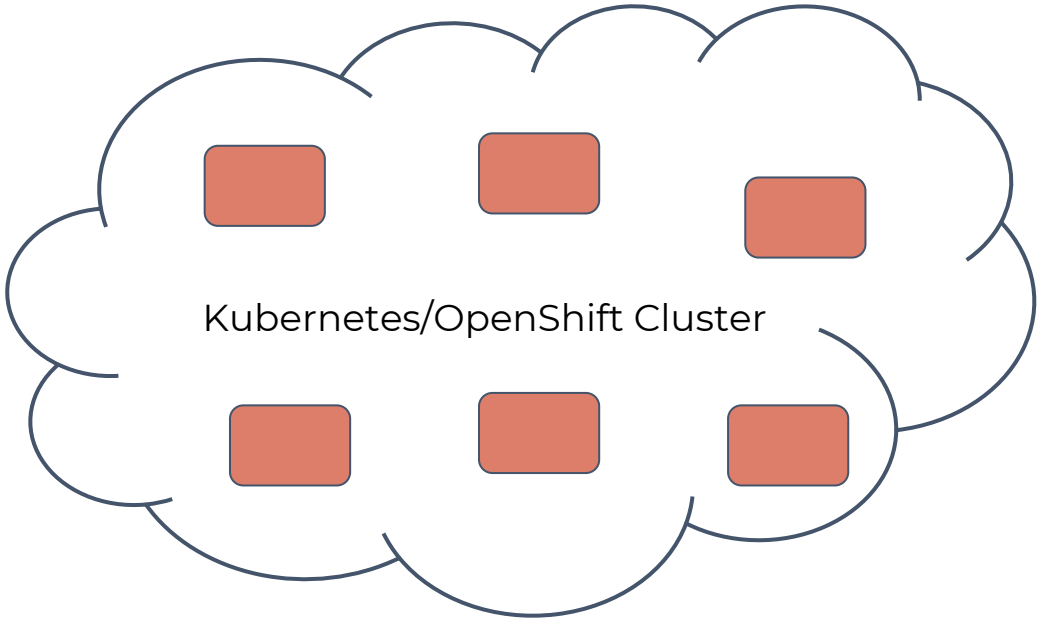
Jobs run on temporary,
dynamically provisioned agents

CloudBees Core



CloudBees Core





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

JAKARTA EE

ECLIPSE FOUNDATION

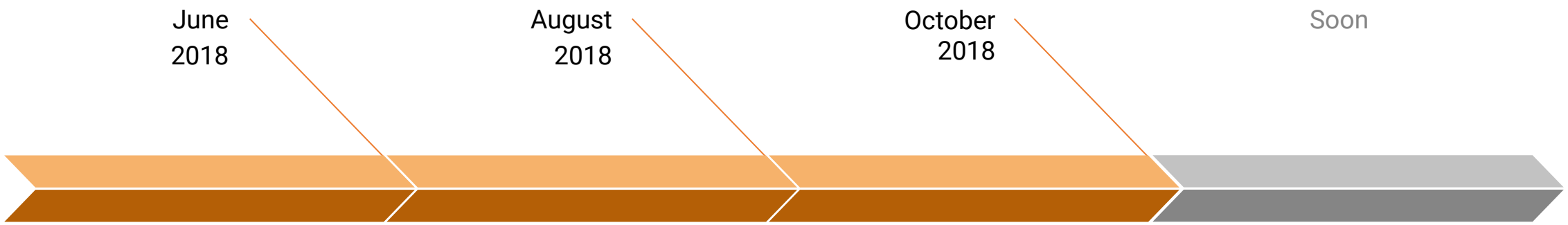
CBC and cluster problems

- 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

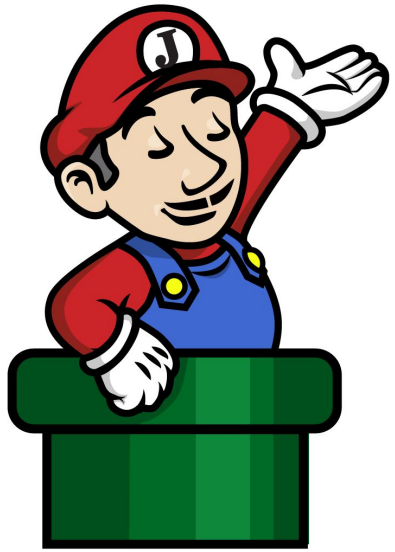
- Why does my build take so long and how can I make it faster?

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!