Maintaining Your Build Farm

Chris B. Behrens
SOFTWARE ARCHITECT
@chrisbbehrens

Upgrading Jenkins

It’s easy to get behind on Jenkins updates

This creates an anti-pattern

You don’t upgrade often, which makes upgrades painful...

Which means you don’t upgrade often enough

If something hurts,
do it often.

If you thought upgrading would be harder

- You were WRONG
- Containerization will (or at least can) force you to do things the right way

We're running on Jenkins LTS

LTS changes over time

When that happens

- Your Jenkins instance changes
- But this is part of the plan

Jenkins auto-updates the volume content if necessary

Upgrading Plug-ins

Jenkins LTS includes plug-in versions

But manually upgraded plug-ins will not automatically upgrade

Manually upgrade them to the LTS version

Uh, what is that version?

Execute your docker container with PLUGINS_FORCE_UPGRADE

PLUGINS_FORCE_UPGRADE

```
docker run -p 2119:8080 -p 50000:50000
-e PLUGINS_FORCE_UPGRADE=true
-v c://Docker/Volumes/jenkins-master:/var/jenkins_home
--name jenkins-master jenkins/jenkins:lts
```

Upgrading Plug-ins

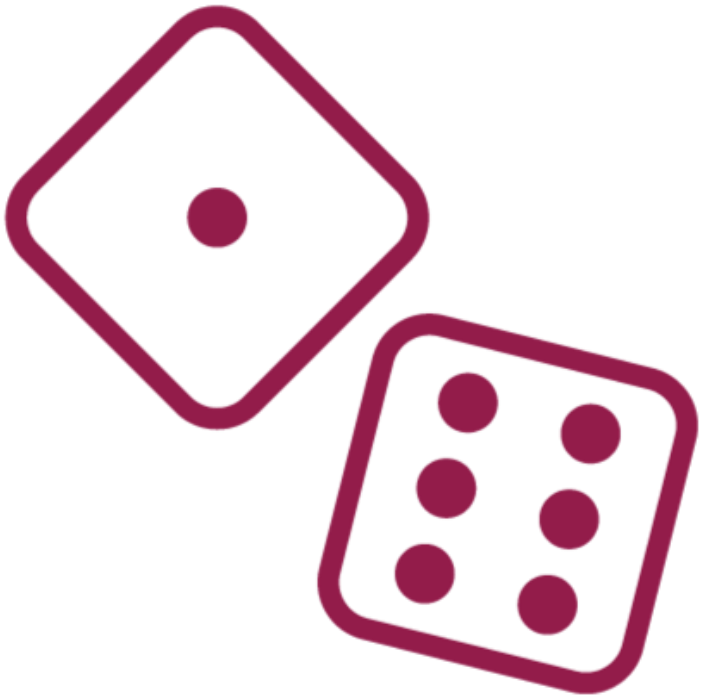
This does nothing for plug-ins not part of LTS

Restart your Jenkins container to upgrade

And for no other reason

Live Restore does nothing for this

The Risk Associated with Upgrading



Upgrading entails risk

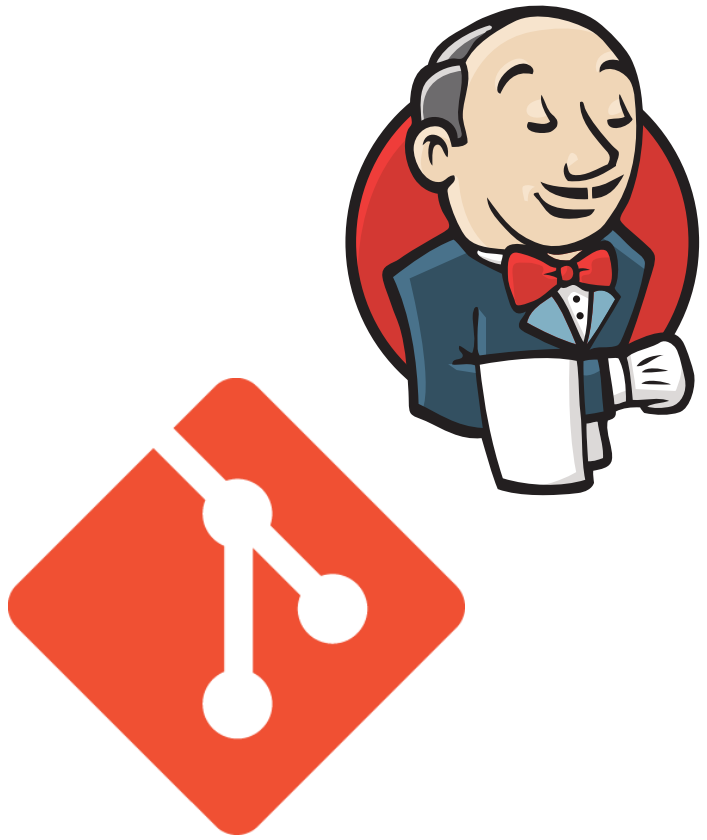
Broken plug-ins will break Production

Before you can sort it out on Github

Test your upgrades in a test environment, just like anything else mission-critical

Back up your volumes!

Jenkins in Git?



You just thought we were experimental

Storing *jenkins_home* in Git

Lots of good version-controllable assets

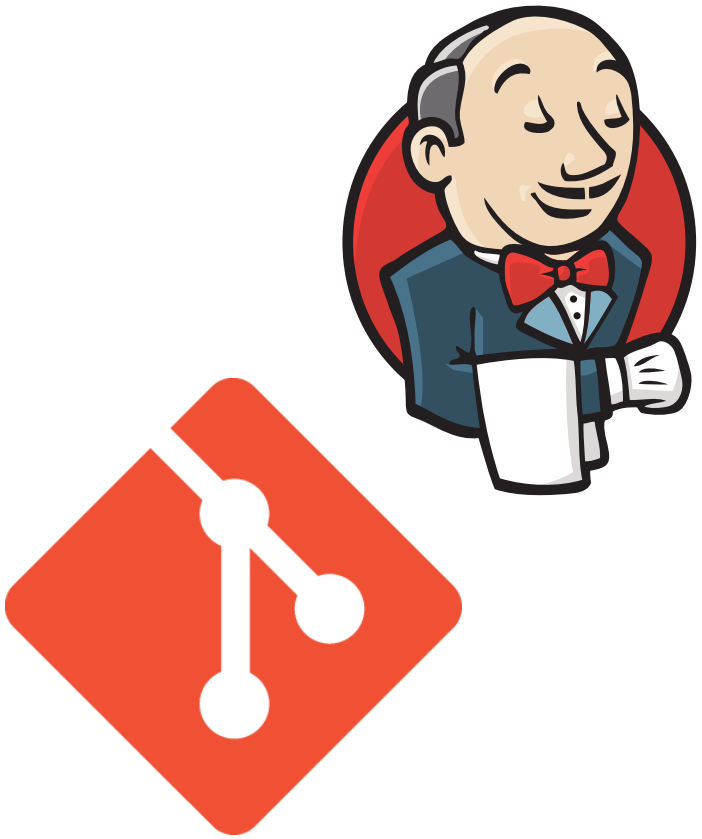
But also other stuff that is not

Create a strong .gitignore

There are actually several online

- But none are perfect

Jenkins in Git How-To



1. Copy *jenkins_home* to another box
2. Check that into a Git repo
   - Minus the binaries

Version history

24-hour backup

Don’t Commit, Round Two

Commit is low-level, and again – OPAQUE

If you want detail, get ready to crack open tarballs

Use version control or backup

Things to Think About

/var/run/docker.sock:/var/run/docker.sock

Needed to make Docker in Docker work

Be careful about WHICH images you trust

We can trust Jenkins

If we keep our Dockerfiles minimal

Running a Minimal Plug-in Profile

Run a minimal plug-in profile – a Jenkins principle

Every plug-in has:

- Security surface area
- Performance impact (under certain circumstances)
- Potential defects

If you don't need it

You don't want it

https://app.pluralsight.com/library/courses/automating-jenkins-groovy

End Credits

Thanks to Docker buddies:

- Floyd May
  - https://app.pluralsight.com/profile/author/floyd-may
- Piotr Gaczkowski
  - https://app.pluralsight.com/profile/author/piotr-gaczkowski
- Matt Milner
  - https://app.pluralsight.com/profile/author/matt-milner

THANK YOU VERY MUCH
FOR WATCHING!!!