The Game of DevOps: Applying Jenkins Best Practices in a Dynamic Industry

Victor Martinez

Jenkins World

A global DevOps event

2017

Who am I

- Victor Martinez
- Configuration Engineer @ARM
- Twitter: @aninfinite
- Active user in the Jenkins community
- JAM Co-Organiser

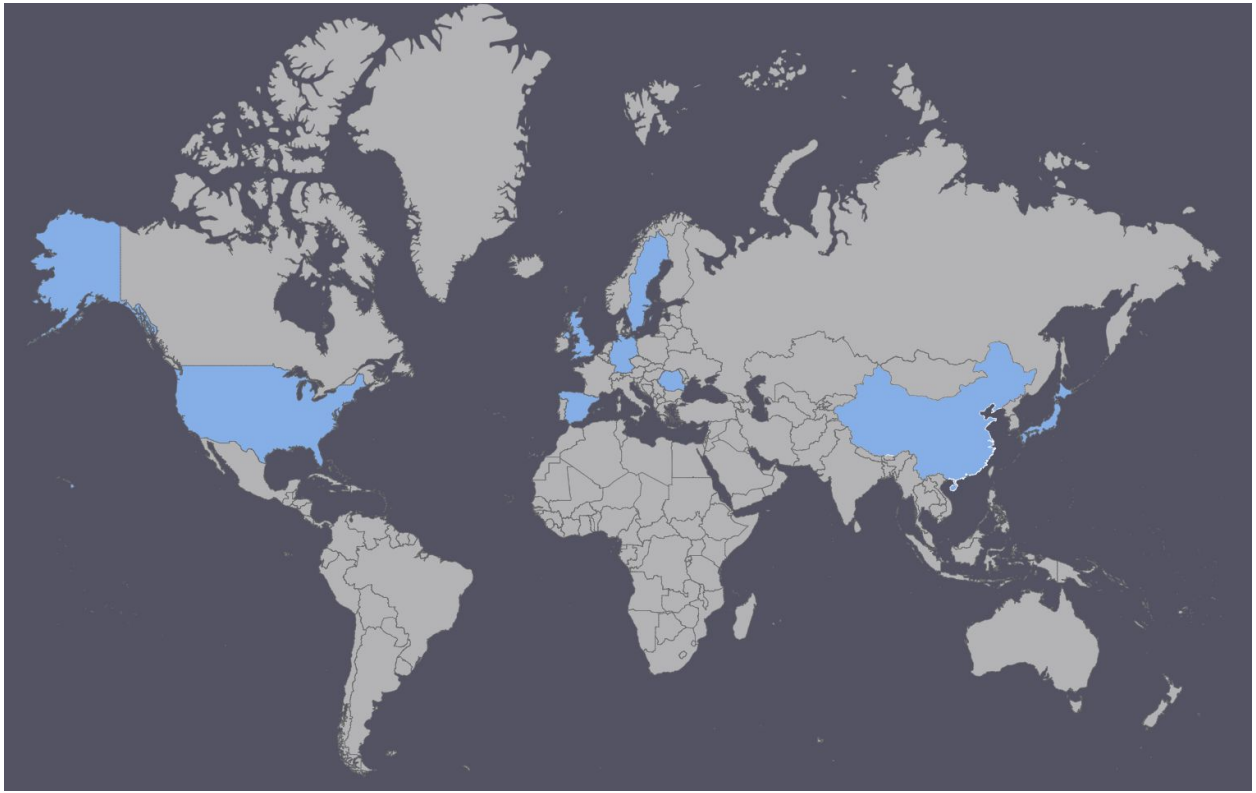
https://www.rnd2016.eng.cam.ac.uk/images/skyline/

Dynamic Industry

http://redcarpetsf.com/real-estate-update-bay-area-still-hot/#jp-carousel-11154

The Game of DevOps

ACME

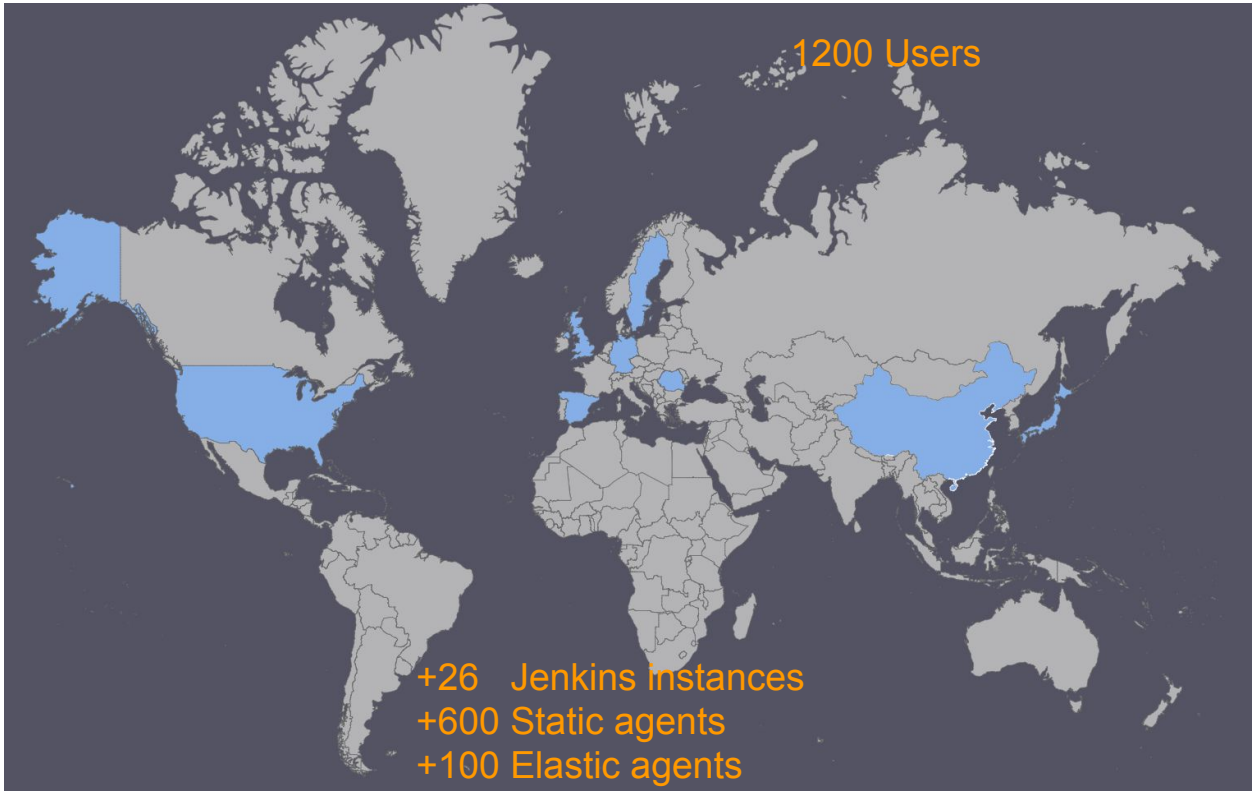


https://www.amcharts.com/visited_countries/

ACME

https://www.amcharts.com/visited_countries/

ACME

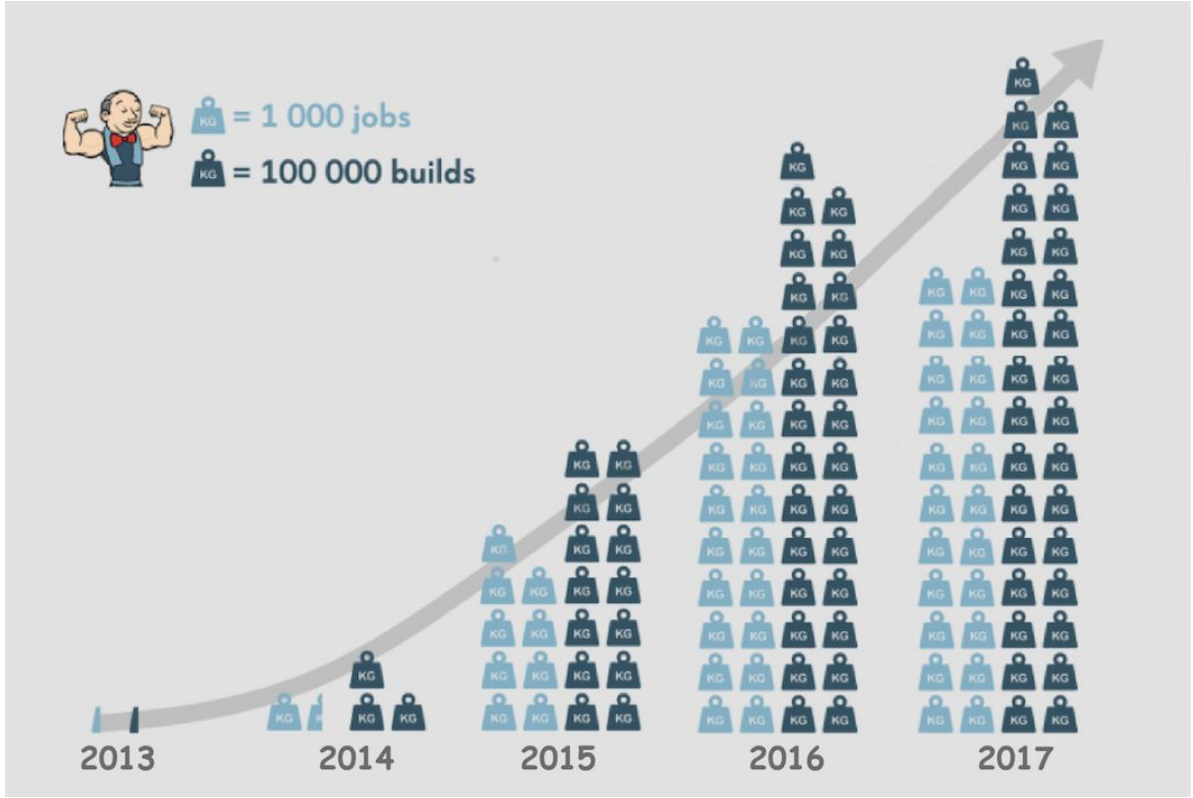


https://www.amcharts.com/visited_countries/

Guidelines

Recommended Action	Benefits	More Information
Name the jobs using the convention **[game].[branch].[action].[platform]**, as detailed here. Organise jobs in tabs using **[game]** as the tab name.	Better organisation and project view in Jenkins. Jobs that are more traceable.	Jenkins Job Naming Convention
Set up a different job/project for each maintenance or development branch you create. Use different jobs for building, testing, and deploying.	The build flow becomes more understandable and has a better high-level view.	
Set a label on your jobs to build on the slaves and, if possible, avoid using the name of the node as the label. Use the label that is geographically closest to you (for example, bcn-osx-pool if you are in Barcelona).	Avoiding Jenkins memory outages. Sharing the the build load between nodes. Shorter build times and faster downloads from repositories.	
Use a policy for log rotation. We recommend a maximum of 90 builds, 30 days for keeping build logs, 3 days for keeping artifacts, and 7 builds for keeping artifacts.	This allows Jenkins and its slaves to save space in the workspaces and delete unused artifacts.	☑ ES-3901 - JENKINS: Set reasonable Log Rotation automatically REOPENED
Avoid polling the repository and use hooks instead. When polling, use randomised polling and set an appropriate time for polling (not every minute).	Polling must die, as it is a waste of resources to ask repositories for changes every minute. If every job uses polling, Jenkins can	Polling must die: triggering Jenkins builds from hooks

But what if there is an exponential usage?



https://www.cloudbees.com/sites/default/files/jenkins-docker-muscles.jpg

Best Practices

#1 Description

http://keywordsuggest.org/gallery/643734.html

#1 Description


```groovy
jenkins.model.Jenkins.instance?.
getAllItems()?.
findAll { !it.getDescription() }?.
each {
  println "'${it.fullName}' doesn't explain what it does."
}
```

#2 Spaces

#2 Spaces


```groovy
jenkins.model.Jenkins.instance?.
  getAllItems()?.
  findAll { it.name.contains(' ') }?.
  each {
    println "'${it.fullName}' contains spaces!"
}
```

#3 Label

https://psy317lauwaijen.wordpress.com/2013/01/18/running-on-empty/

#3 Label


```
1  jenkins.model.Jenkins.instance?.
2  getAllItems()?.
3  findAll { !it.getAssignedLabel() }?.
4  each {
5    println "'${it.fullName}' doesn't have any assigned agent/label"
6  }
```

#4 Workspace

https://hbr.org/2015/01/why-a-messy-workspace-undermines-your-persistence

#4 Workspace


```
jenkins.model.Jenkins.instance?.
getAllItems()?.
findAll {
  !(it.getPublishersList()?.find {it.getClass().getName().contains('WsCleanup')}) &&
  !(it.getBuildWrappers()?.find{ k,v -> k.getClass().getName().contains('PreBuildCleanup') } != null ) &&
  !(it.scm && it.scm.getClass().getName().contains('hudson.plugins.git.GitSCM') &&
       it.scm.getExtensions()?.find { it instanceof hudson.plugins.git.extensions.impl.WipeWorkspace } != null )
   }?.
each {
  println "'${it.fullName}' is not cleaning up its workspace"
}
```

#5 Log rotator

http://www.balavto.com/production/excavator-attachments/grabs/hanging-grab-log-manoeuvring

#5 Log rotator


```
jenkins.model.Jenkins.instance?.
getAllItems(Job.class)?.
each {
  found = true
  if (it.getBuildDiscarder() )
    found = it.getBuildDiscarder().getArtifactDaysToKeep() < 0 && it.getBuildDiscarder().getArtifactNumToKeep() < 0 &&
            it.getBuildDiscarder().getDaysToKeep() < 0 && it.getBuildDiscarder().getNumToKeep() < 0
  if (found)
    println "'${it.fullName}' doesn't have any log rotator configuration"
}
```

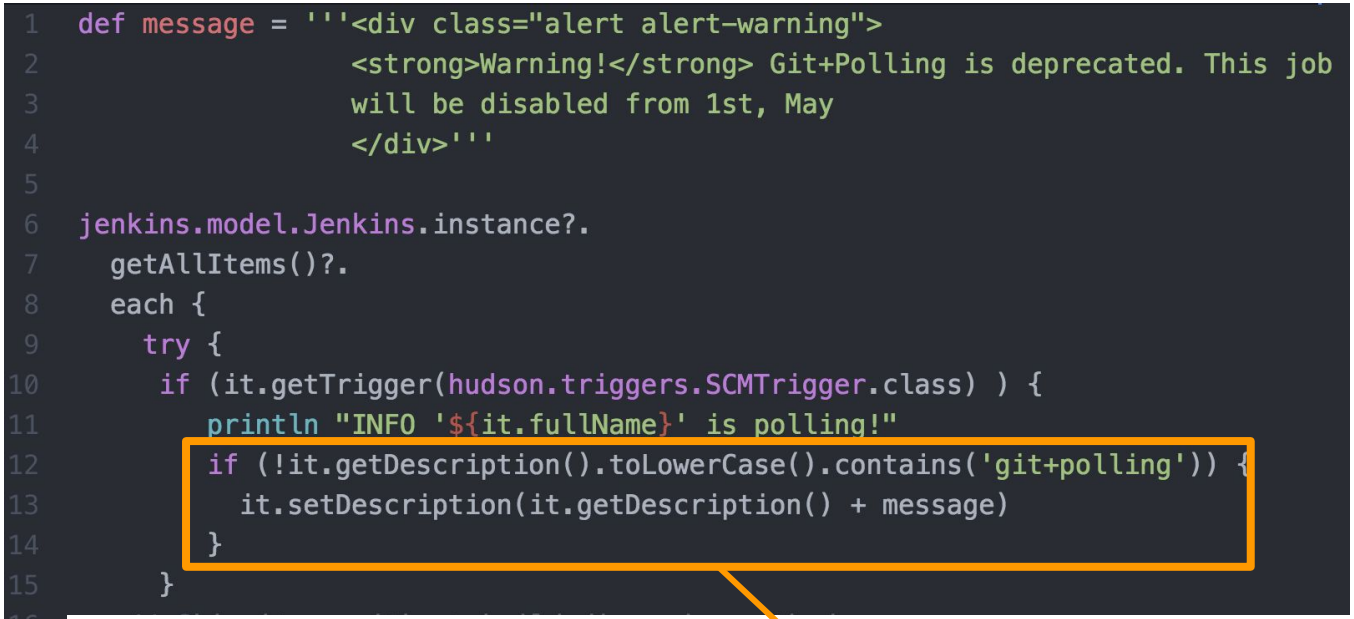
#6 Pushing

https://twitter.com/fr3dg/status/481828080676536320

#6 Pushing


```groovy
jenkins.model.Jenkins.instance?.
  getAllItems()?.
  each {
    try {
      if (it.getTrigger(hudson.triggers.SCMTrigger.class) ) {
        println "'${it.fullName}' is polling!"
      }
    // Skip items without build discarder method
    } catch (groovy.lang.MissingMethodException mme) { }
}
```

#6 Pushing



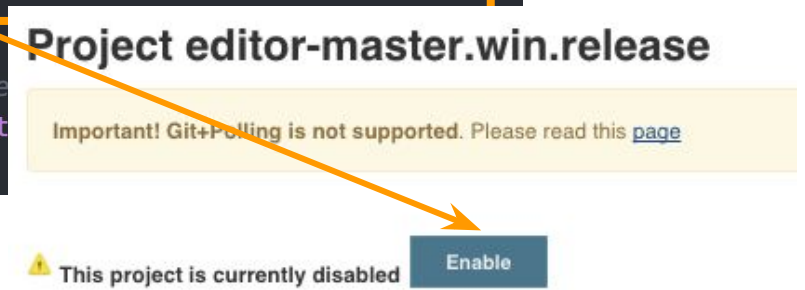
```groovy
def message = '''<div class="alert alert-warning">
                 <strong>Warning!</strong> Git+Polling is deprecated. This job
                 will be disabled from 1st, May
                 </div>'''

jenkins.model.Jenkins.instance?.
  getAllItems()?.
  each {
    try {
     if (it.getTrigger(hudson.triggers.SCMTrigger.class) ) {
        println "INFO '${it.fullName}' is polling!"
        if (!it.getDescription().toLowerCase().contains('git+polling')) {
          it.setDescription(it.getDescription() + message)
        }
     }
```

#6 Pushing

```groovy
def message = '''<div class="alert alert-warning">
                 <strong>Important! Git+Polling is not supported</strong>.
                 </div>'''

jenkins.model.Jenkins.instance?.
  getAllItems()?.
  each {
    try {
     if (it.getTrigger(hudson.triggers.SCMTrigger.class) ) {
        println "INFO '${it.fullName}' is polling!"
        if (!it.getDescription().toLowerCase().contains('git+polling')) {
          it.setDescription(it.getDescription() + message)
          it.disable()
        }
    }
   // Skip items without build discarder me
   } catch (groovy.lang.MissingMethodExcept
}
```



#7 Shallow

https://www.tes.com/lessons/er6Hsk17a1Q0vw/concept-shallow-and-deep

#7 Shallow


```groovy
import hudson.plugins.git.extensions.impl.CloneOption
jenkins.model.Jenkins.instance?.
getAllItems()?.
findAll { try { it.getScm() } catch (groovy.lang.MissingMethodException mme) {} }?.
findAll { it.getScm() instanceof hudson.plugins.git.GitSCM }?.
each {
  isShallow = false
  if (it.getScm().getExtensions() && it.getScm().getExtensions()?.
                                        find {
                                          it instanceof CloneOption &&
                                          it.isShallow()
                                        }
     ) {
    isShallow = true
  }
  if (!isShallow)
    println "'${it.fullName}' is not using shallow cloning"
}
```

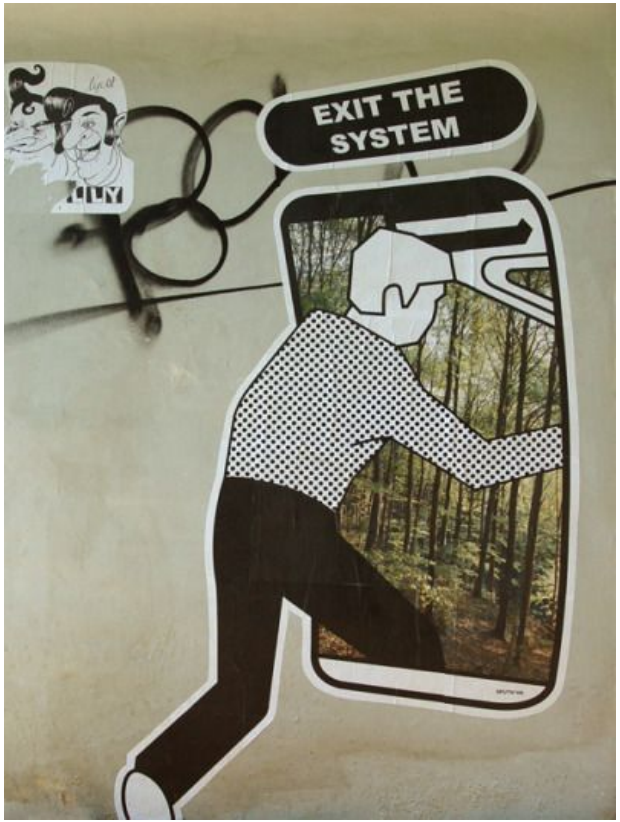
#8 Reference

https://s3.amazonaws.com/libapps/accounts/1687/images/reference.jpg

#8 Reference


```groovy
import hudson.plugins.git.extensions.impl.CloneOption
jenkins.model.Jenkins.instance?.
getAllItems()?.
findAll { it.getScm().getType().toLowerCase().contains('git') }?.
each { job ->
  if (job.scm.getExtensions().empty) { // When no options
    println "'${job.fullName}' is not using Git Reference Repo"
  }
  job.scm.getExtensions()?.findAll { it instanceof CloneOption }.each { p ->
    if (!p.getReference().trim()){
      println "'${job.fullName}' is not using Git Reference Repo"
    }
  }
}
```

#9 System exit



http://fucknobigbrother.tumblr.com/post/12913270051/via-wooster-collective-exit-the-system

#9 System exit


```groovy
jenkins.model.Jenkins.instance?.
getAllItems(AbstractProject.class)?.
findAll { it.getPublishersList() }?.each {
  if (it.getPublishersList()?.
          findAll { it.getClass().getName().endsWith("GroovyPostbuildRecorder") &&
                    it.getScript()?.getScript()?.toLowerCase().contains("system.exit")}
      ) {
    println "'${it.fullName}' is using System Exit"
  }
}
```

#10 Groovy Sandbox

http://chicagokidsisland.com/wp-content/uploads/2013/11/indoor-sandbox.jpg

#10 Groovy Sandbox


```
1    jenkins.model.Jenkins.instance?.
2      getAllItems(org.jenkinsci.plugins.workflow.job.WorkflowJob.class)?.
3      findAll { !it.getDefinition()?.isSandbox() }?.
4      each {
5        println "Pipeline -> '${it.fullName}' doesn't use Groovy Sandbox"
6    }
```

#11 Maven Jobs

https://es.wikipedia.org/wiki/MAVEN#/media/File:Mars-MAVEN-Orbiter-20140921.jpg

#11 Maven Jobs


```groovy
jenkins.model.Jenkins.instance?.
  getAllItems(hudson.maven.MavenModuleSet.class)?.
  each {
    println "'${it.fullName}' is a Maven Project!"
}
```

#12 Abort building

http://maxpixel.freegreatpicture.com/Abort-Renovation-House-Demolition-Demolition-41855

#12 Abort building


```groovy
import hudson.plugins.build_timeout.BuildTimeoutWrapper
jenkins.model.Jenkins.instance?.
getAllItems()?.
findAll {
  try {
    !it.getBuildWrappers()?.find { k,v -> v instanceof BuildTimeoutWrapper }
  } catch (groovy.lang.MissingMethodException mee) {}
}?.each {
  println "'${it.fullName}' doesn't use Build Timeout Plugin"
}
```

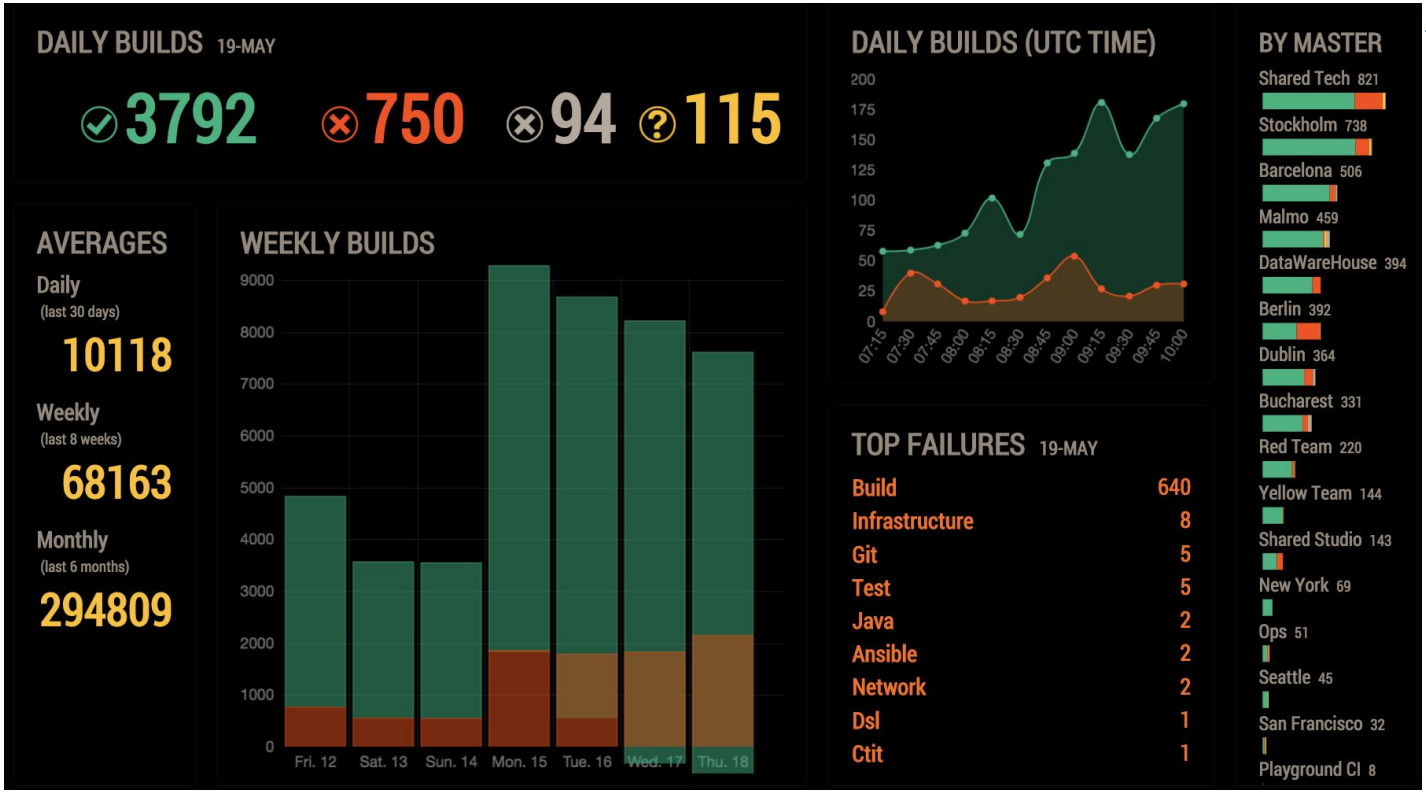
#13 Build Failure Plugin

http://energinnovation.com/category/collaborative-innovation

#13 Build Failure Plugin


```groovy
import com.sonyericsson.jenkins.plugins.bfa.model.ScannerJobProperty
jenkins.model.Jenkins.instance?.
getAllItems()?.
each {
  def bfa = it.getAllProperties()?.find { prop -> prop instanceof ScannerJobProperty }
  if (!bfa || bfa?.isDoNotScan()) {
    println "'${it.fullName}' is not using BFA"
  }
}
```

#13 Build Failure Plugin



Control Comments

Control Comments

Project MyFreestyleJob

This is only for testing purpose

```
disable:git:shallow_cloning
```

```
jenkins.model.Jenkins.instance?.
  getAllItems()?.
  findAll { try { it.getScm() } catch (groovy.lang.MissingMethodException mme) {} }?.
  findAll { it.getScm() instanceof hudson.plugins.git.GitSCM }?.
  findAll { !it.description.contains("disable:git:shallow_cloning") }?.
  each {
    isShallow = false
    if (it.getScm().getExtensions() && it.getScm().getExtensions()?.
                                        find {
                                          it instanceof hudson.plugins.git.extensions.impl.CloneOption &&
                                          it.isShallow()
                                        }
       ) {
      isShallow = true
    }
    if (!isShallow)
      println "'${it.fullName}' is not using shallow cloning"
}
```

How to evangelise those practices?

- They are coded practices
- Easy to automate/collaborate
- DryRun
- Visualise
- Poke Owners (JobOwnership Plugin helped)
- Levels:
  - Deprecated policy
  - Disable policy
  - Modify policy

Automated Linting

sonar

Pmd

DON'T SHOOT THE MESSENGER

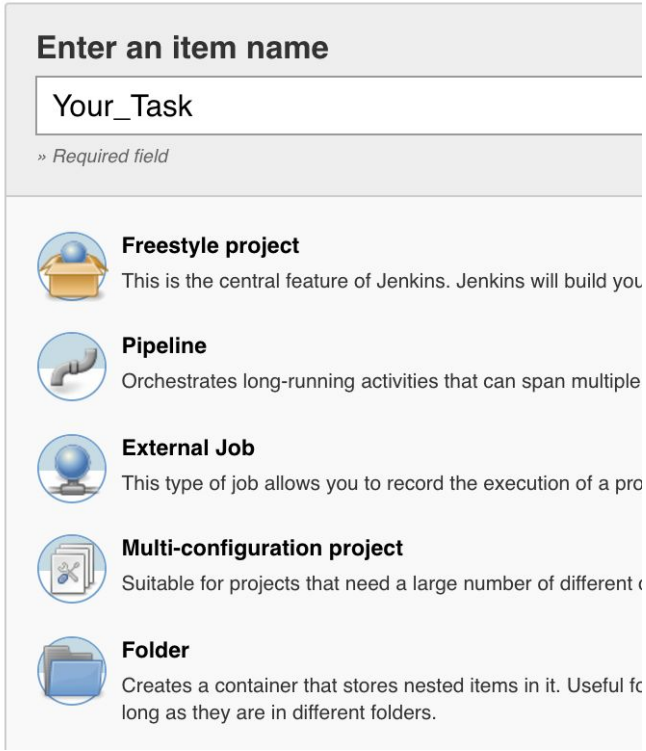
checkstyle

Jenkins Lint: Items supported

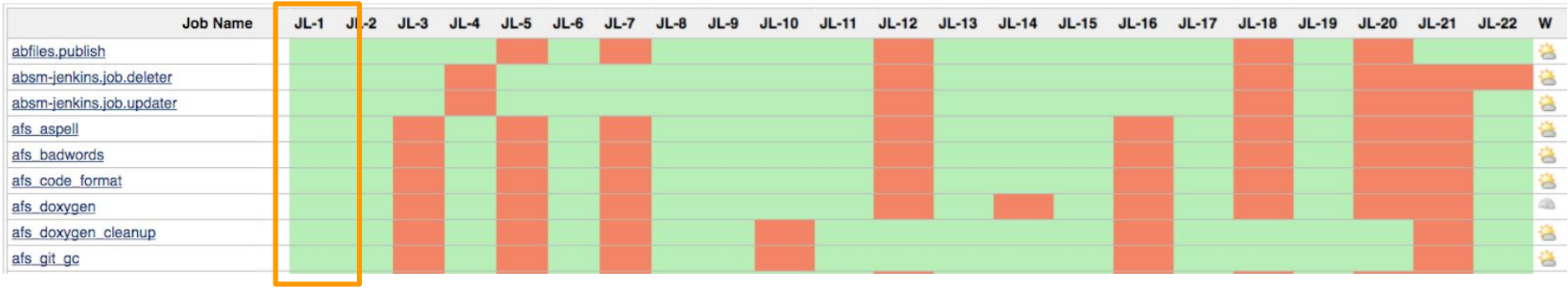
Stage View

Permalinks

- Last build (#1), 2 hr 14 min ago
- Last stable build (#1), 2 hr 14 min ago
- Last successful build (#1), 2 hr 14 min ago

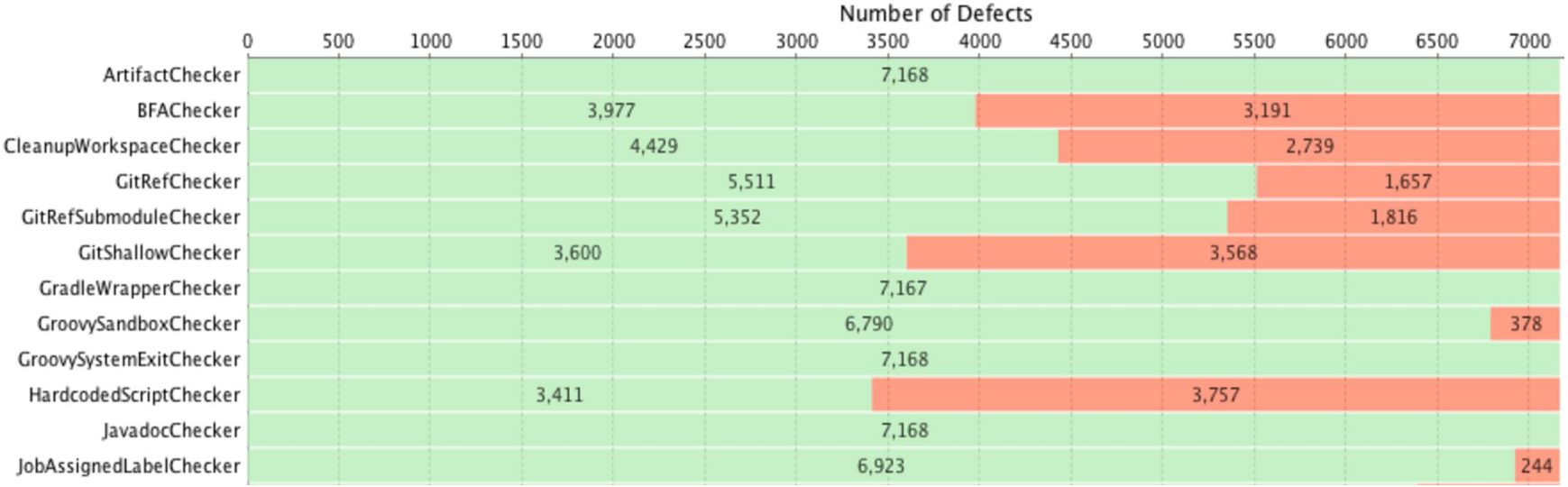


Jenkins Lint: jobs view

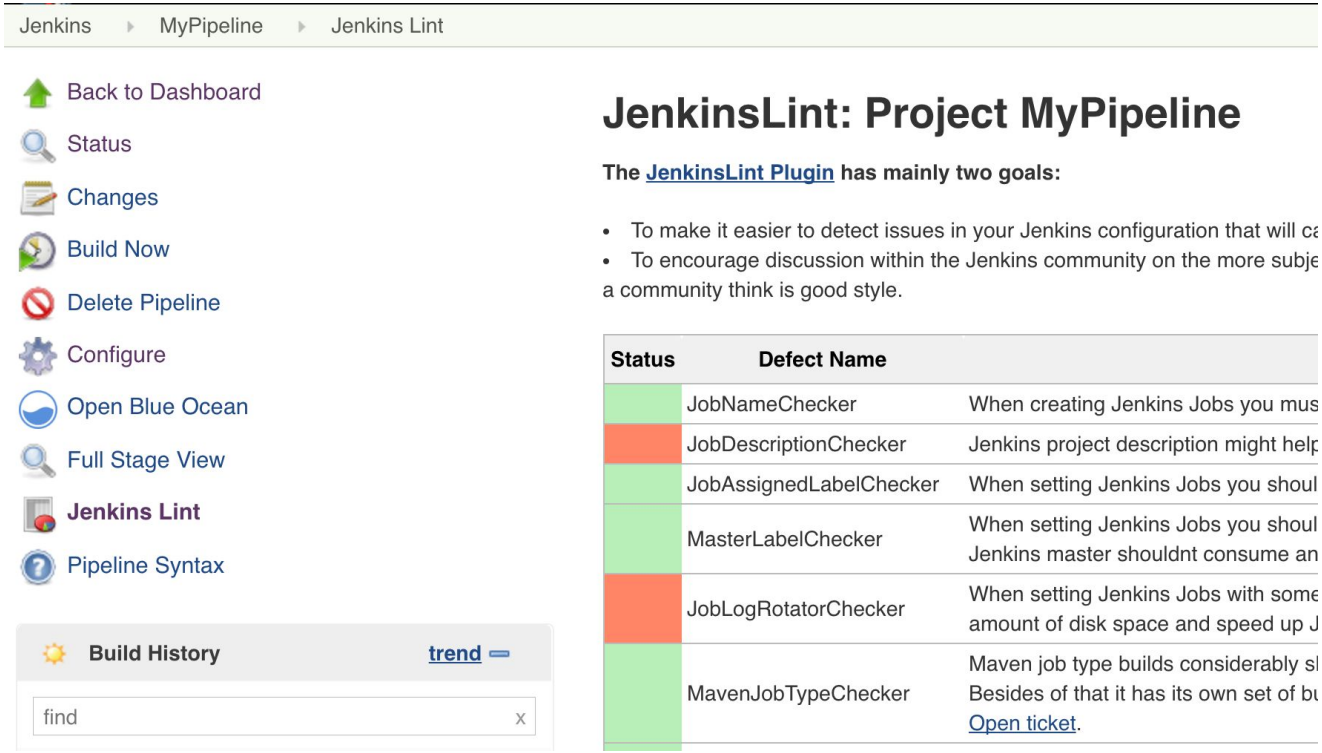


Lint Defects

Jenkins Lint: defects view



Jenkins Lint: job view



Jenkins Lint: overall lint job view

Pipeline MyPipeline

Recent Changes

add description

Disable Project

Lint stability: 90%

Jenkins Lint: agent view

Jenkins ▸ Nodes ▸ Terminator ▸ Jenkins Lint ▸

- Back to List
- Status
- Delete Agent
- Configure
- Build History
- Load Statistics
- Log
- **Jenkins Lint**
- Open Blue Ocean

Build Executor Status

JenkinsLint: Computer Terminator

The JenkinsLint Plugin has mainly two goals:

- To make it easier to detect issues in your Jenkins configuration that will
- To encourage discussion within the Jenkins community on the more sub

good style.

Status	Defect	
	SlaveDescriptionChecker	Jenkins slave description might h
	SlaveVersionChecker	Jenkins slave version should mat
	SlaveLabelChecker	When setting Jenkins Slaves you
	WindowsSlaveLaunchChecker	This launch method relies on DC(Start instead, which also permitsinstallation as a

Jenkins Lint: docs

JL-1: JobNameChecker
Description: When creating Jenkins Jobs you must avoid whitespace. In order to comply with the style guide.
Severity: High
Control Comment: To disable this checker you can add `lint:ignore:JobNameChecker` to your Jenkins project.

JL-2: JobDescriptionChecker
Description: Jenkins project description might help you to know what it does and further details.
Severity: Medium
Control Comment: To disable this checker you can add `lint:ignore:JobDescriptionChecker` to your Jenkins project.

JL-3: JobAssignedLabelChecker
Description: When setting Jenkins Jobs you should set where those Jobs can run.
Severity: Medium
Control Comment: To disable this checker you can add `lint:ignore:JobAssignedLabelChecker` to your Jenkins project.

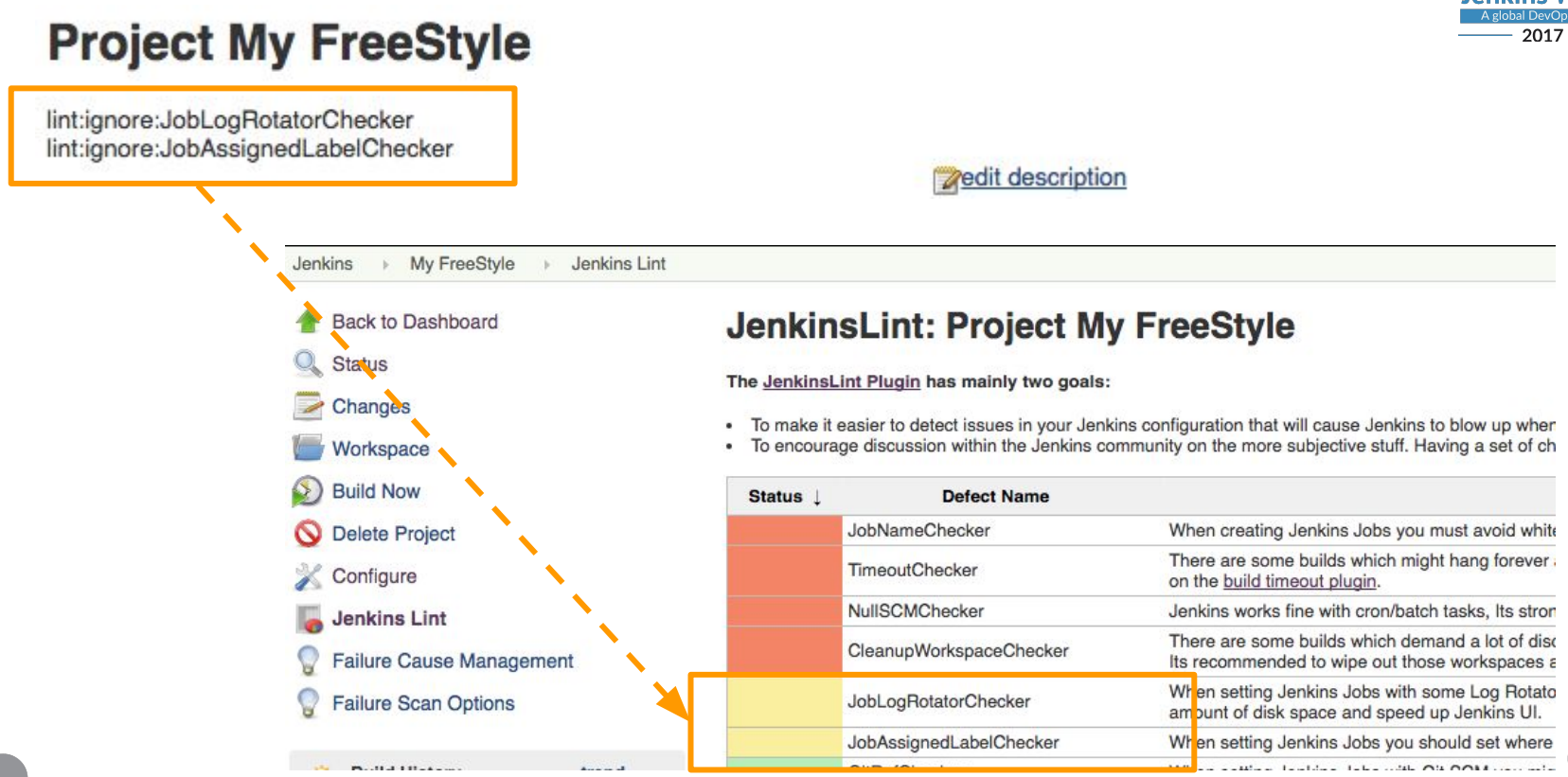
Jenkins Lint: disable lints

Project My FreeStyle

```
lint:ignore:JobLogRotatorChecker
lint:ignore:JobAssignedLabelChecker
```

edit description

Jenkins Lint: disable lints



Jenkins Lint: global settings

Jenkins ▸ configuration

Jenkins Lint

Enabled ☑

If Jobs should be scanned or not.

Enable analysis of disabled Jobs ☑

Whether to lint disabled jobs or not.

Enable Job floating box ☑

Whether to display JobAction floatingBox or not.

Jenkins Lint Checkers

Artifact Checker ☑

BFA Checker ☑

Cleanup Workspace Checker ☑

Git Ref Checker ☑

Future

- Fully Pipeline support.
- Jenkins lint command line tool to lint Pipelines and JobDSL.
- Auto-fix trivial lint defects, such as:
  - Log rotator
  - Groovy sandbox
  - Workspace cleanup
- Disable projects automatically.
- Code Quality integration as a single source of truth.

Want to try Jenkins Lint?

https://plugins.jenkins.io/jenkinslint

Q&A

Thanks

https://github.com/v1v/JenkinsWorld2017

Jenkins World

A global DevOps event

2017