

Criando pipelines de entrega contínua multilinguagem com Docker e Jenkins

Camilo Ribeiro
Klarna AB





Camilo Ribeiro

Software Test Engineer at Klarna
ex-Senior QA Consultant at Thoughtworks

Blogger on www.bugbang.com.br



@camiloribeiro



HQ em Estocolmo/Suécia
Maior empresa de pagamento online da
europa, atuando na Suécia, Alemanha,
Noruega, Dinamarca, Austria, Reino Unido,
Estados Unidos e vários outros países.

Foco em entregas rápidas, confiáveis e
automação para testes, deploys e
configuração

www.klarna.com



<https://github.com/camiloribeiro/cdeasy>

README.md

cdeasy - Continuous Delivery Made Easy

```
##      .
## ## ##    ==
## ## ## ##   ===
/"""""""""""\_/_ ===
~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
\____ o      _/
 \_\ \_/_/
 \_\_\_/_/
<3 Docker <3
```

Example of how docker and automation can help you to imporove your continuous delivery process.

This is not a production ready framework or template to start using in your current project. Please take this as an inspiration to improve or implement a Continuous Develivey pipeline for your own needs :)

<https://github.com/camiloribeiro/cdeasy>





1



2



3



4

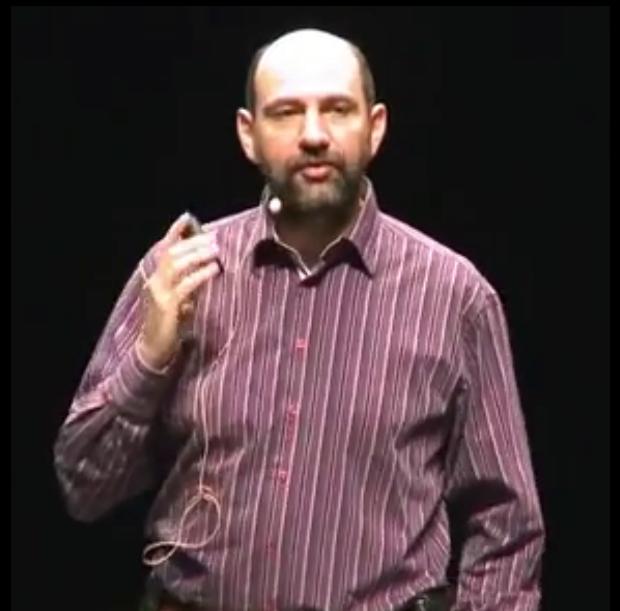


5

Continuous Delivery

“ Entrega Continua é uma disciplina de desenvolvimento de software aonde você desenvolve software de uma maneira que ele possa ser entregue em produção a qualquer momento. ”

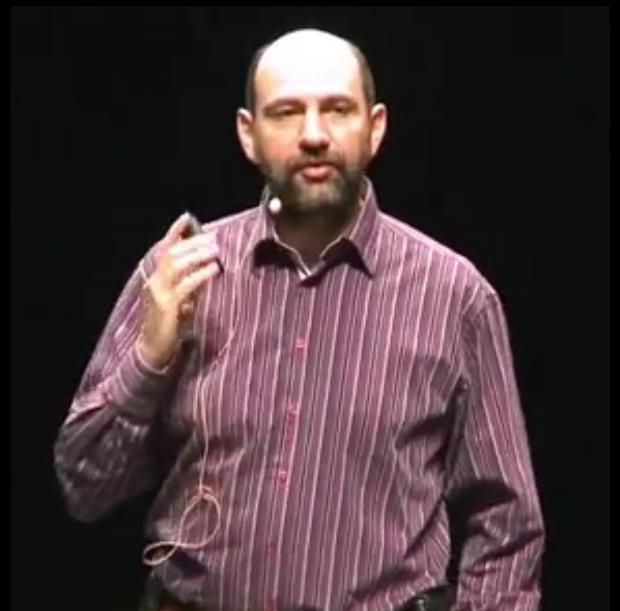
Martin Fowler



Continuous Delivery

“ Entrega Continua é uma disciplina de desenvolvimento de software aonde você desenvolve software de uma maneira que ele **possa ser entregue em produção a qualquer momento.** “

Martin Fowler





ubuntu



Java



MySQL



CentOS



fedora



NGINX



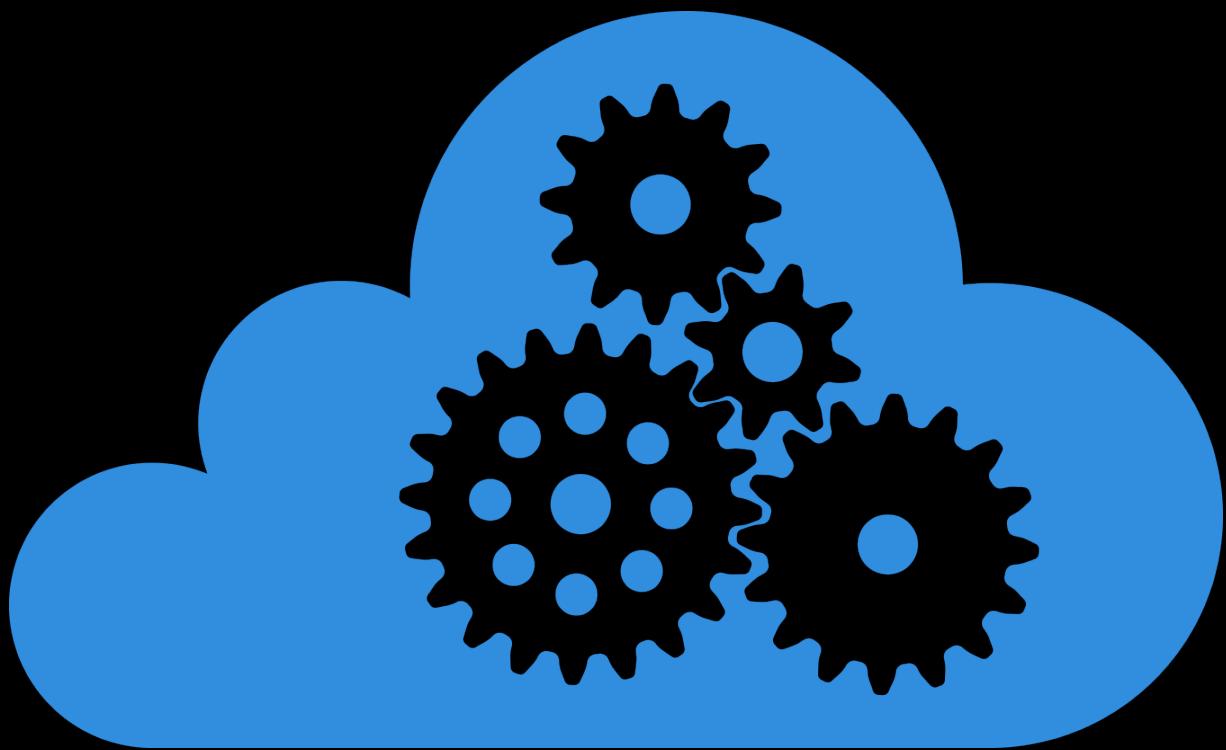
Cucumber

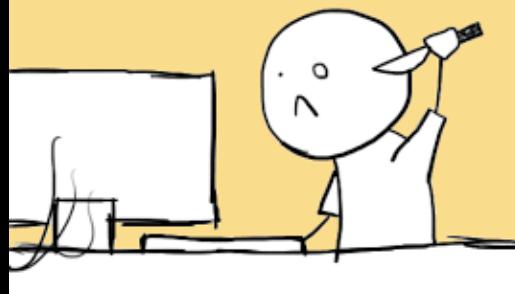
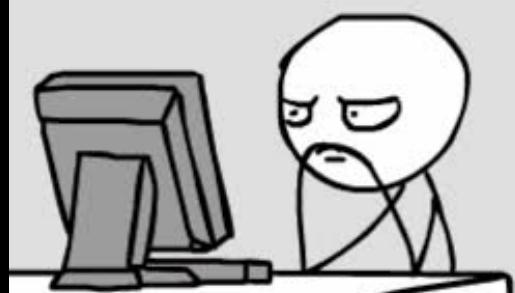
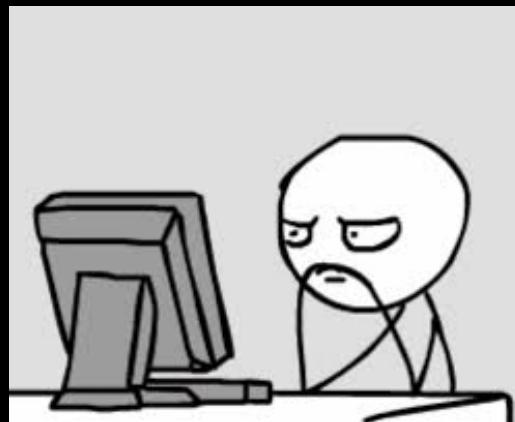


ERLANG



ubuntu





Project name: running_java_with_docker

Description:

(Escaped HTML) [Preview]

- Discard Old Builds
- Docker Container
- This build is parameterized
- Delivery Pipeline configuration
- Disable Build (No new builds will be executed until the project is re-enabled.)
- Execute concurrent builds if necessary

Advanced Project Options

Source Code Management

- None
- CVS
- CVS Projectset
- Git
- Repositories

Repository URL: https://github.com/camiloribeiro/cucumber-gradle-parallel.git

Credentials: - none - Add

Branches to build

Branch Specifier (blank for 'any') ::

Add Branch Delete Branch

Repository browser (Auto)

Additional Behaviours Add

Subversion

Build Triggers

- Build after other projects are built
- Built periodically
- Poll SCM

Build Environment

- Create Delivery Pipeline version
- Build

Execute shell

Command: docker pull niaquinto/gradle:2.5

See the list of available environment variables

Delete

Execute shell

Command: docker run -v \$WORKSPACE:/gradle -w /gradle niaquinto/gradle:2.5 clean build runInParallel

See the list of available environment variables

Delete

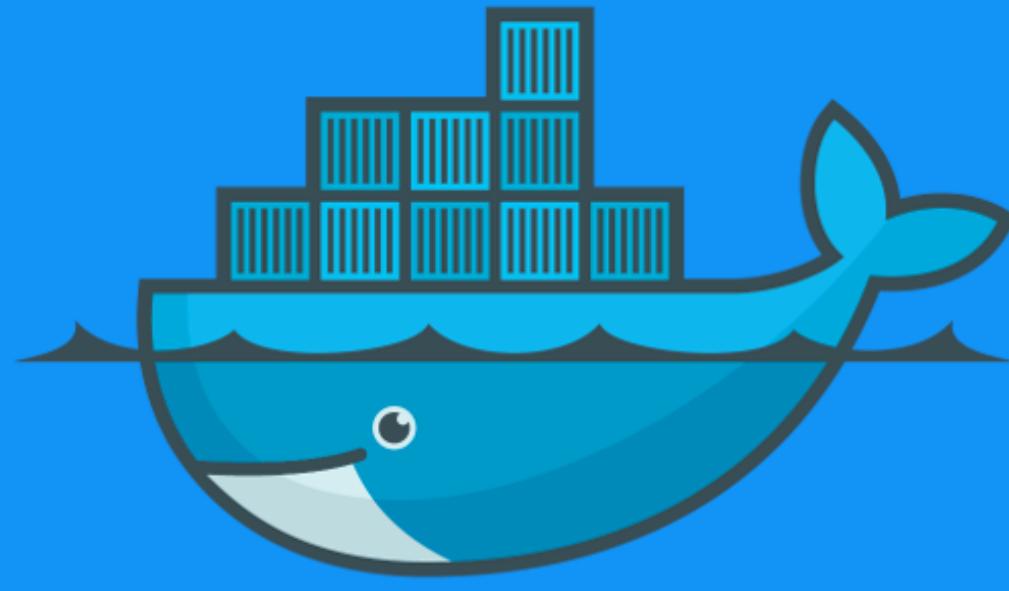
Add build step ·

Post-build Actions

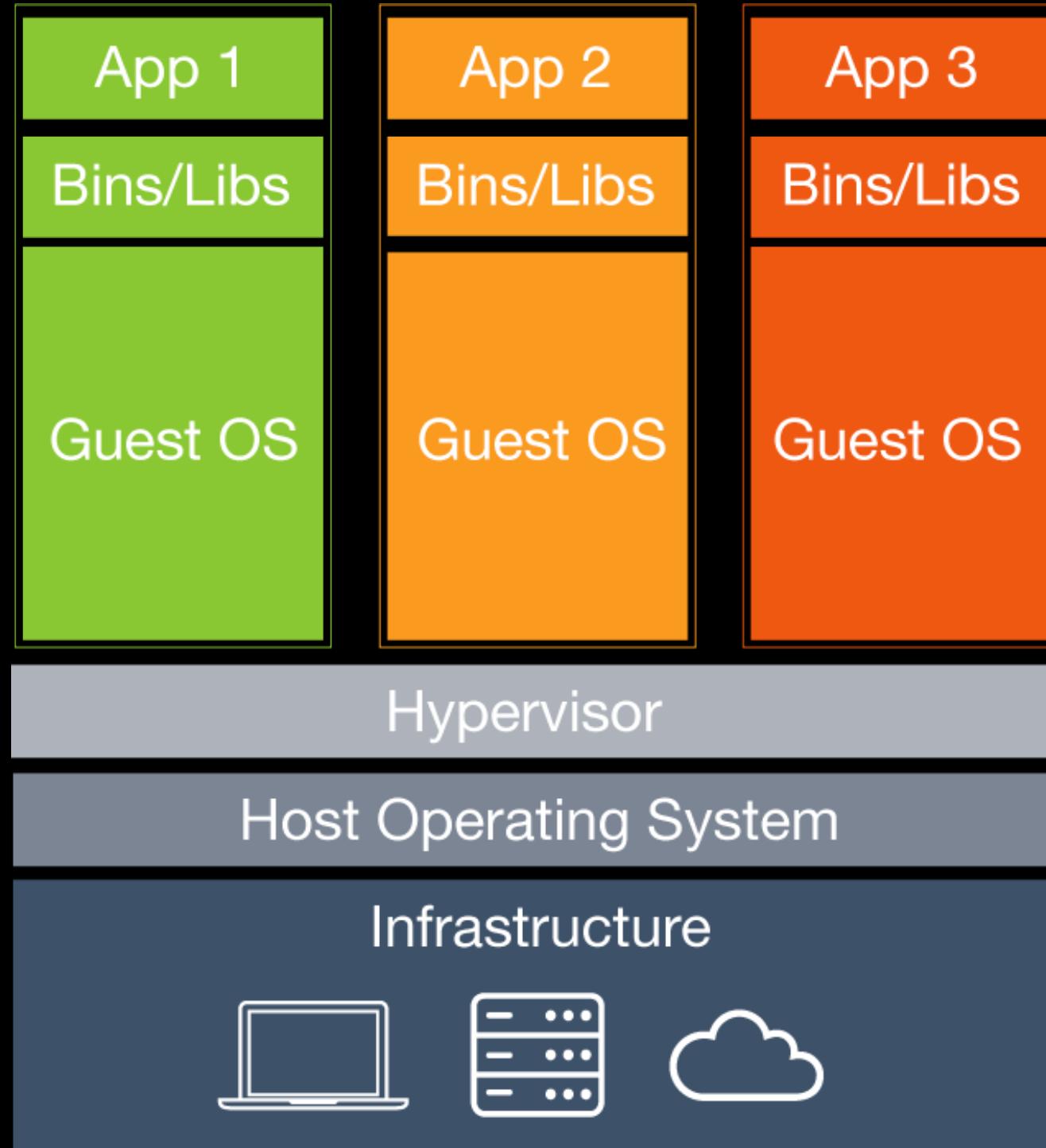
```

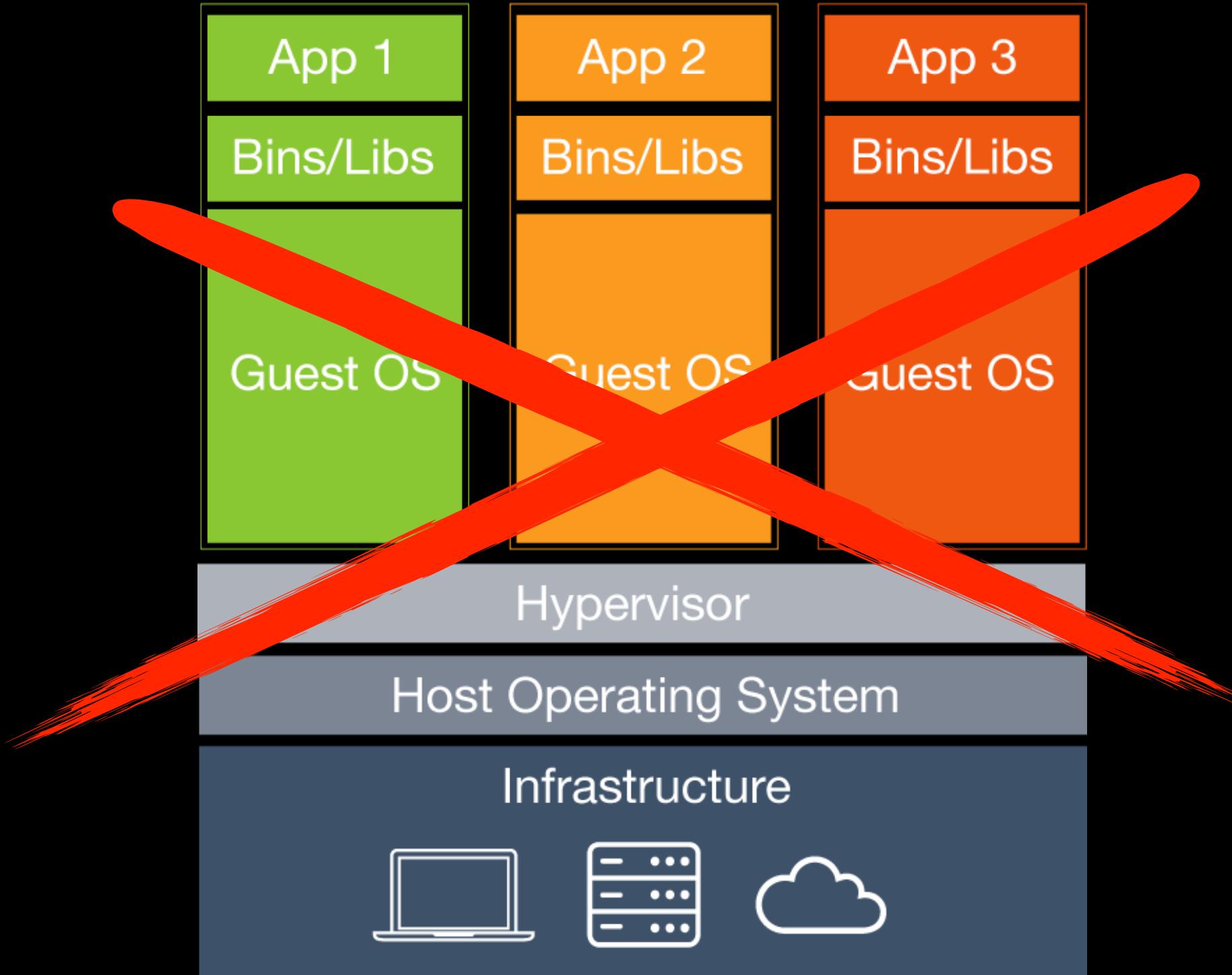
<project>
    <actions></actions>
    <description></description>
    <keepDependencies>false</keepDependencies>
    <properties></properties>
    <canRoam>true</canRoam>
    <disabled>false</disabled>
    <blockBuildWhenDownstreamBuilding>false</blockBuildWhenDownstreamBuilding>
    <blockBuildWhenUpstreamBuilding>false</blockBuildWhenUpstreamBuilding>
    <triggers class='vector'></triggers>
    <concurrentBuild>false</concurrentBuild>
    <builders>
        <hudson.tasks.Shell>
            <command>docker pull niaquinto/gradle:2.5</command>
        </hudson.tasks.Shell>
        <hudson.tasks.Shell>
            <command>docker run -v $WORKSPACE:/gradle -w /gradle niaquinto/gradle:2.5 clean build
runInParallel</command>
        </hudson.tasks.Shell>
    </builders>
    <publishers>
        <htmlpublisher.HtmlPublisher>
            <reportTargets>
                <htmlpublisher.HtmlPublisherTarget>
                    <reportName>Cucumber Report</reportName>
                    <reportDir>build/reports/cucumber</reportDir>
                    <reportFiles>feature-overview.html</reportFiles>
                    <keepAll>true</keepAll>
                    <allowMissing>true</allowMissing>
                    <alwaysLinkToLastBuild>true</alwaysLinkToLastBuild>
                    <wrapperName>htmlpublisher-wrapper.html</wrapperName>
                </htmlpublisher.HtmlPublisherTarget>
            </reportTargets>
        </htmlpublisher.HtmlPublisher>
    </publishers>
    <buildWrappers></buildWrappers>
    <scm class='hudson.plugins.git.GitSCM'>
        <userRemoteConfigs>
            <hudson.plugins.git.UserRemoteConfig>
                <name>origin</name>
                <url>https://github.com/camiloribeiro/cucumber-gradle-parallel.git</url>
            </hudson.plugins.git.UserRemoteConfig>
        </userRemoteConfigs>
        <branches>
            <hudson.plugins.git.BranchSpec>
                <name>**</name>
            </hudson.plugins.git.BranchSpec>
        </branches>
        <configVersion>2</configVersion>
        <disableSubmodules>false</disableSubmodules>
        <recursiveSubmodules>false</recursiveSubmodules>
        <doGenerateSubmoduleConfigurations>false</doGenerateSubmoduleConfigurations>
        <authorOrCommitter>false</authorOrCommitter>
        <clean>false</clean>
        <wipeOutWorkspace>false</wipeOutWorkspace>
        <pruneBranches>false</pruneBranches>
        <remotePoll>false</remotePoll>
        <ignoreNotifyCommit>false</ignoreNotifyCommit>
        <gitTool>Default</gitTool>
        <skipTag>true</skipTag>
    </scm>
</project>

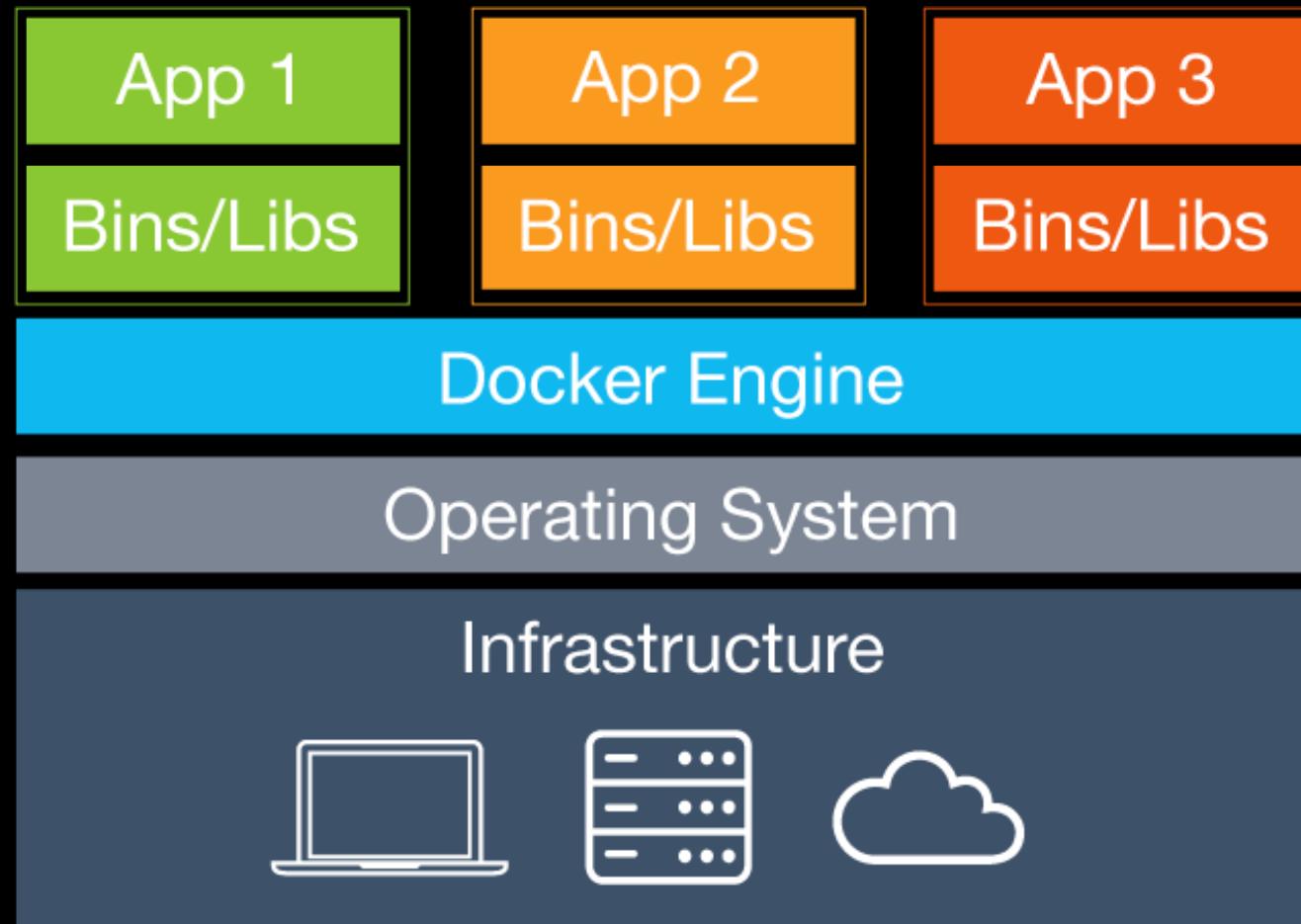
```

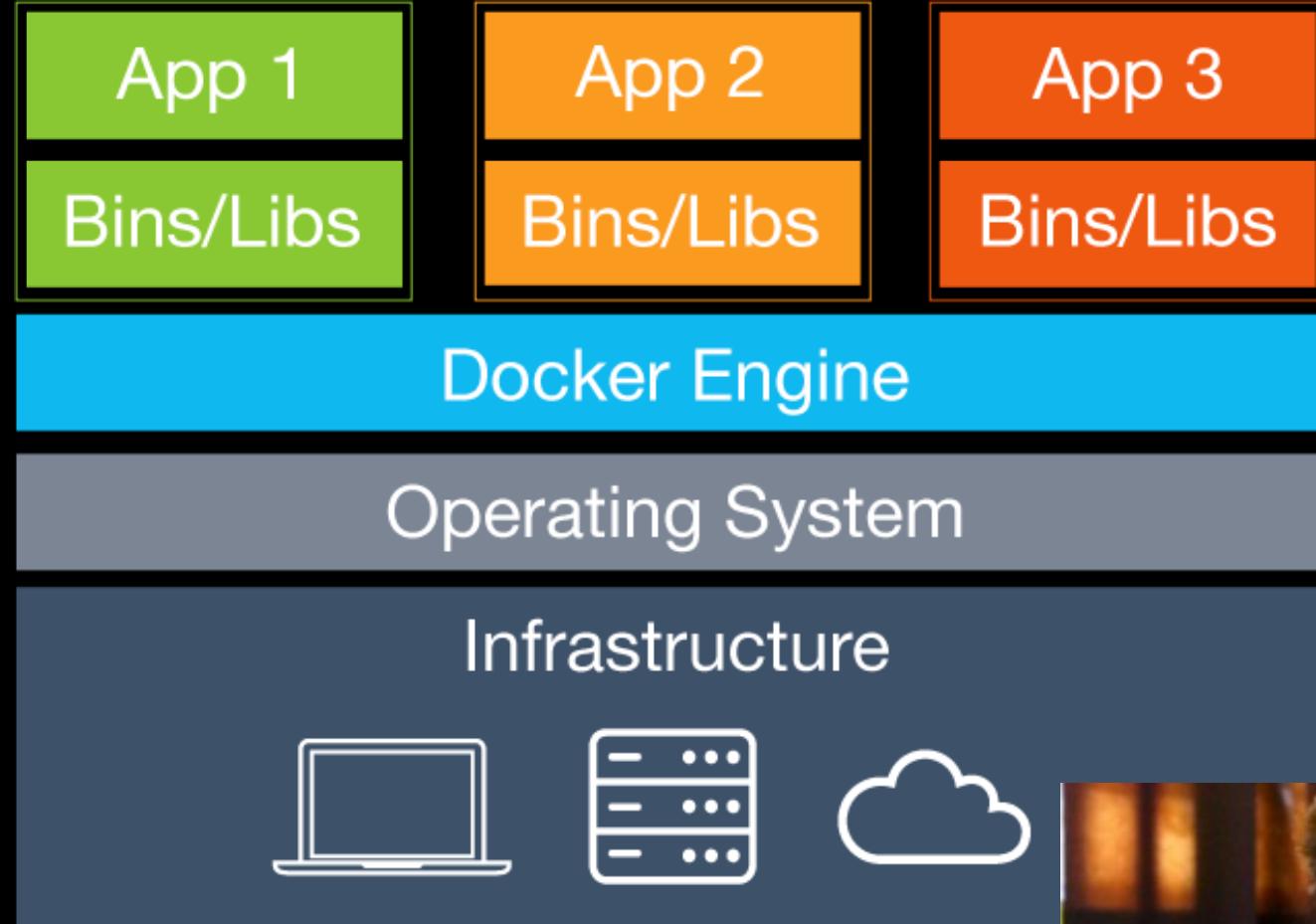


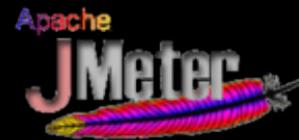
docker



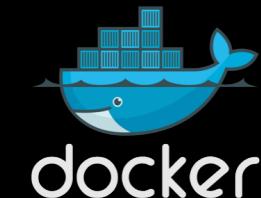
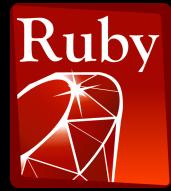








Cucumber



CentOS



ubuntu



fedora



docker

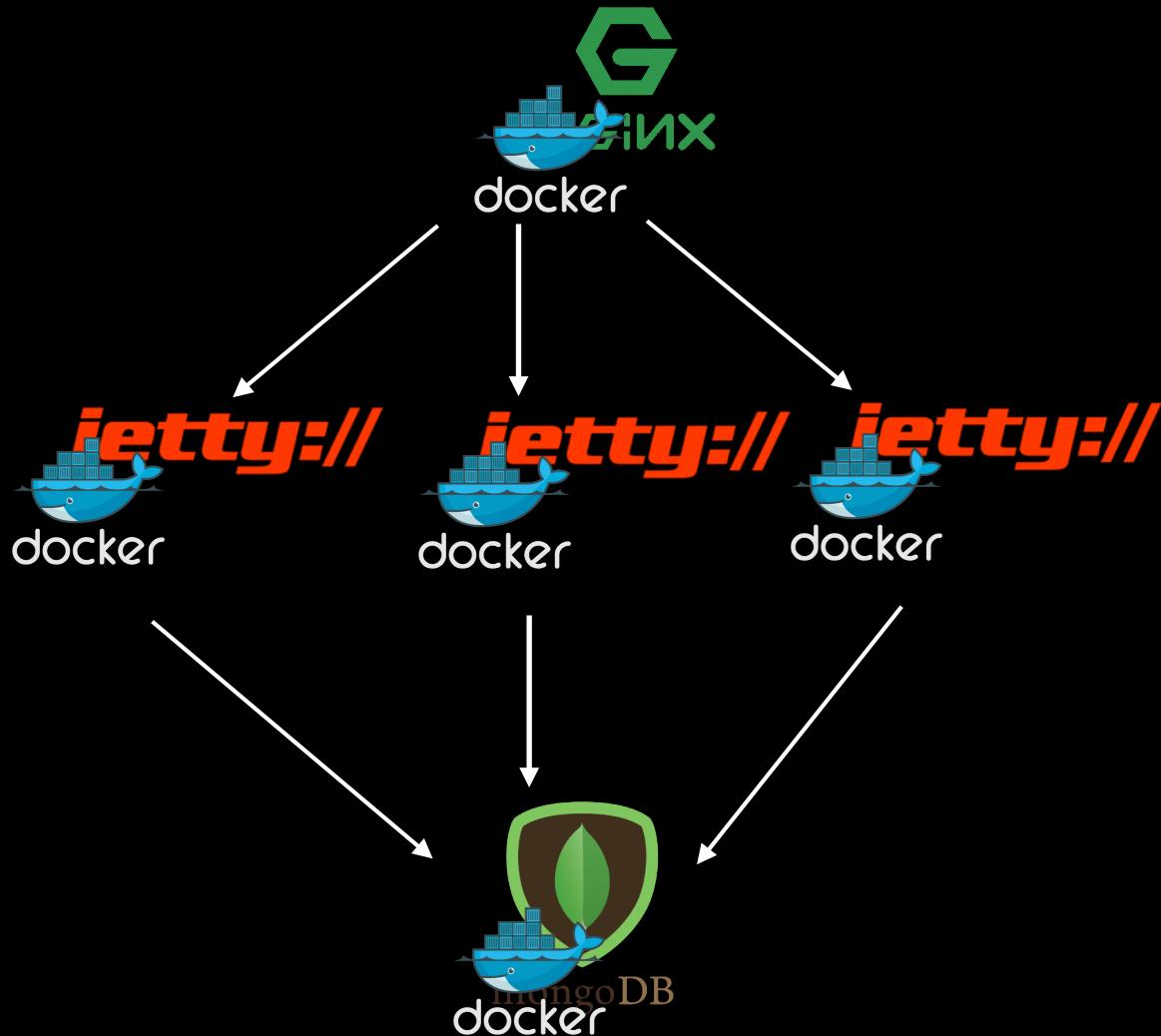
docker pull gradle

docker run gradle clean build runInParallel

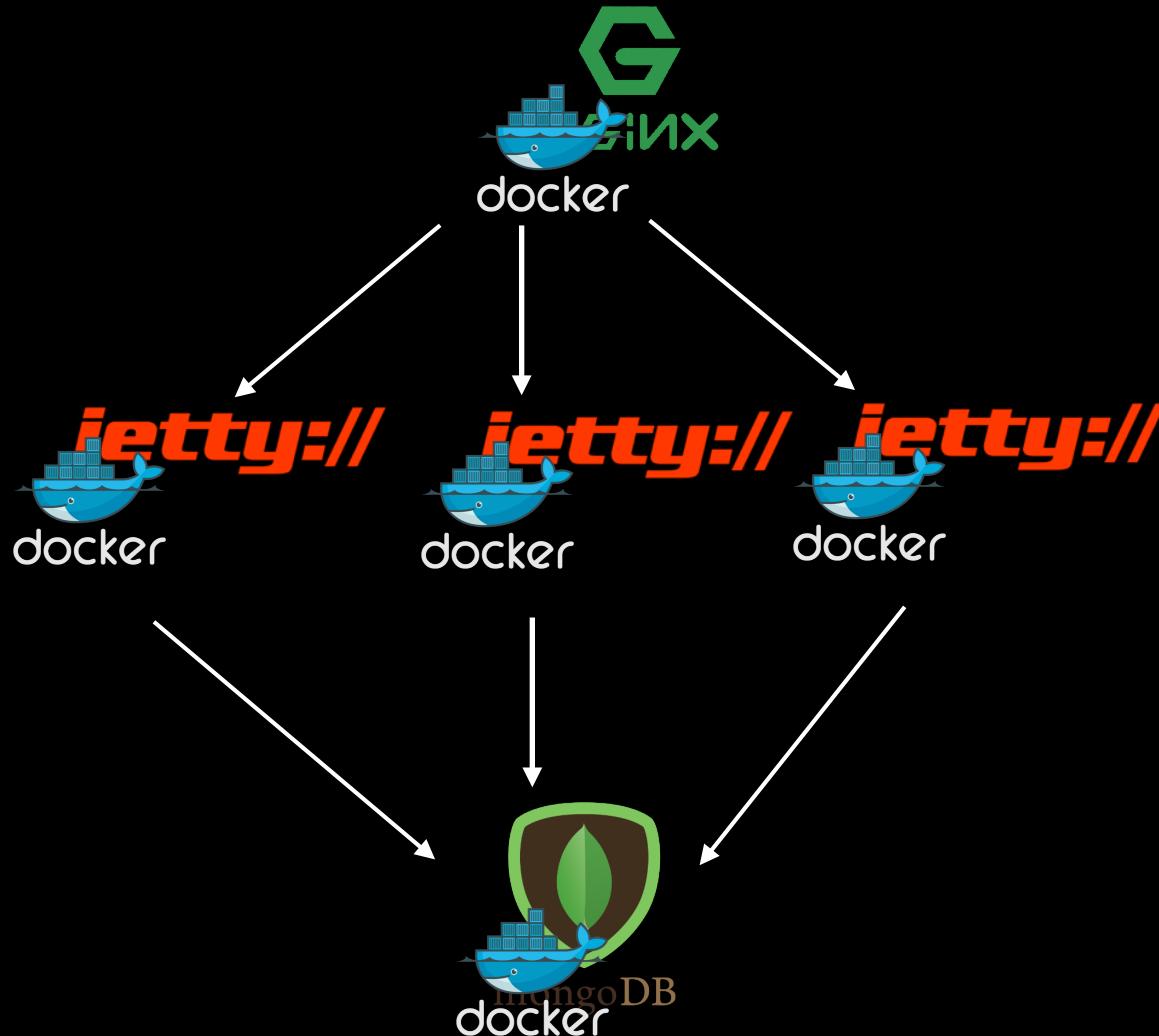
```
docker pull ruby  
docker run ruby bundle install && rake test
```

```
docker pull node  
docker run node npm install && node app.js
```

“Poder de cloud” no seu local



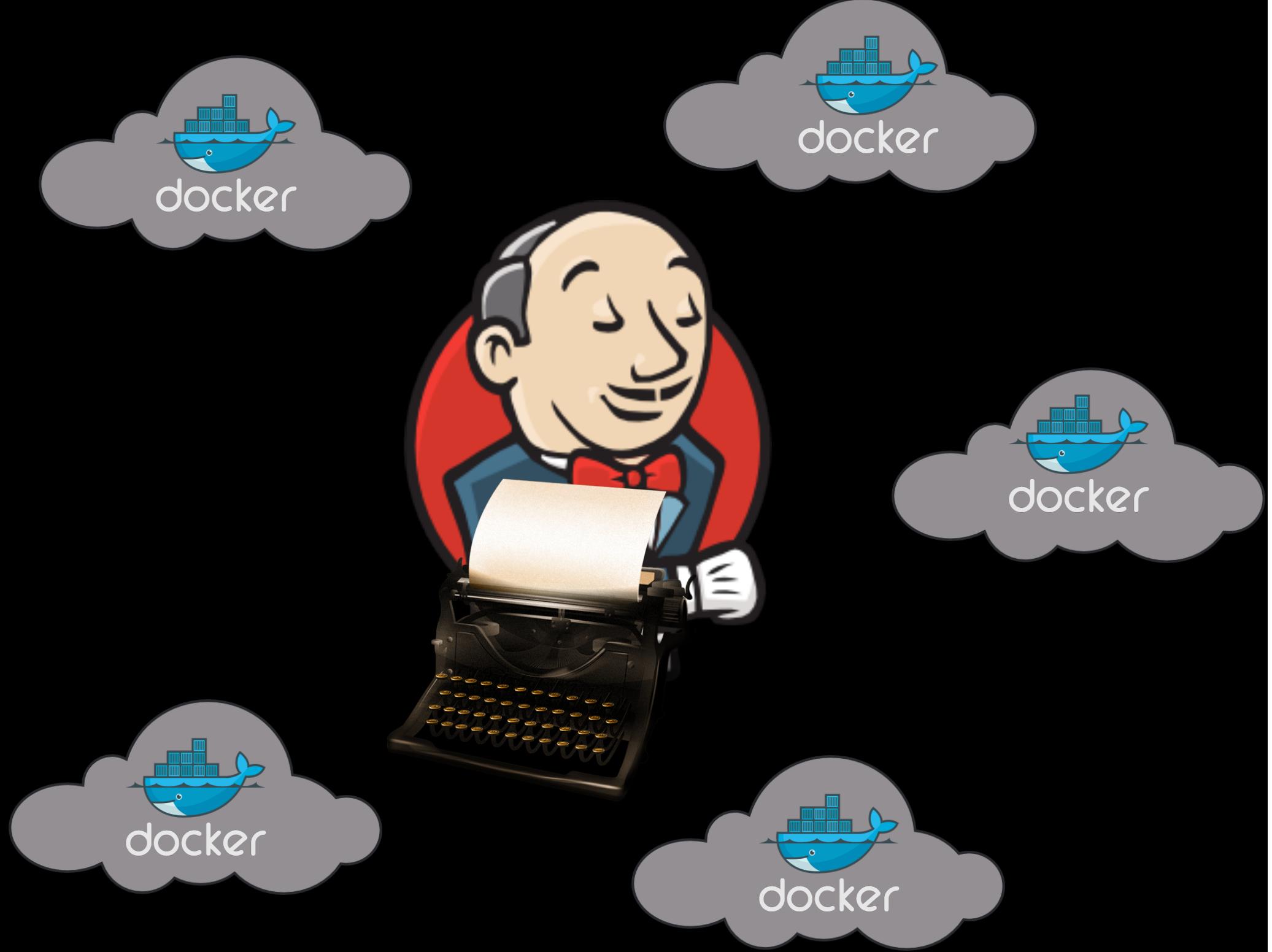
docker-compose



```
nginx:  
  build: docker/nginx  
  ports:  
    - "8080:8080"  
  tty: true  
  links:  
    - server
```

```
server:  
  build: docker/server  
  links:  
    - database  
  volumes:  
    - ./src:/tmp  
  command: /tmp/setup-jenkins.sh
```

```
database:  
  build: docker/mongo  
  ports:  
    - "27017:27017"  
    - "27018:27018"  
    - "27019:27019"  
    - "28017:28017"  
  tty: true
```



Jenkins Job DSL



```
job("running_java_with_docker") {
    scm {
        git {
            remote {
                name('origin')
                url ("https://github.com/camiloribeiro/cucumber-gradle-parallel.git")
            }
        }
    }
    triggers {
        scm 'H/5 * * * *'
    }
    steps {
        shell 'docker pull niaquinto/gradle:2.5'
        shell 'docker run -v $WORKSPACE:/gradle -w /gradle niaquinto/gradle:2.5 clean build runInParallel'
    }
    publishers {
        publishHtml {
            report('build/reports/cucumber') {
                reportName('Cucumber Report')
                reportFiles('feature-overview.html')
                keepAll()
                allowMissing()
                alwaysLinkToLastBuild()
            }
        }
    }
}
```

```
job("running_java_with_docker") {  
    scm {  
        git {  
            remote {  
                name('origin')  
                url ("https://github.com/...")  
            }  
        }  
    }  
    triggers {  
        scm 'H/5 * * * *'  
    }  
    steps {  
        shell 'docker pull cucumber/cucumber:latest'  
        shell 'docker run -it cucumber/cucumber --tags @cucumber'<br/>  
    }  
    publishers {  
        publishHtml {  
            report('build/reports/cucumber') {  
                reportName('Cucumber Report')  
                reportFiles('feature-overview.html')  
                keepAll()  
                allowMissing()  
                alwaysLinkToLastBuild()  
            }  
        }  
    }  
}
```

Project name

running_java_with_docker

Description

[Escaped HTML] [Preview](#)

Discard Old Builds

Docker Container

This build is parameterized

'nParallel'

```
job("running_java_with_docker") {  
    scm {  
        git {  
            remote {  
                name('origin')  
                url ("https://github.com/camiloribeiro/cucumber-gradle-parallel.git")  
            }  
        }  
    }  
    triggers {  
        scm 'H/5 * * * *'  
    }  
    steps {  
        shell 'docker pull nia'  
        shell 'docker run -v $'  
    }  
    publishers {  
        publishHtml {  
            report('build/reports')  
            reportName('Cucu')  
            reportFiles('feature')  
            keepAll()  
            allowMissing()  
            alwaysLinkToLastBuild()  
        }  
    }  
}
```

Source Code Management

None
 CVS
 CVS Projectset
 Git

Repositories

Repository URL:

Credentials:

Branches to build:

Repository browser:

Additional Behaviours:

Subversion

```
job("running_java_with_docker") {  
    scm {  
        git {  
            remote {  
                name('origin')  
                url ("https://github.com/camiloribeiro/cucumber-gradle-parallel.git")  
            }  
        }  
    }  
    triggers {  
        scm 'H/5 * * * *'  
    }  
    steps {  
        shell 'docker pull niaquinto/gradle:2.5'  
        shell '  
    }  
    publish {  
        publish {  
            repository {  
                repository {  
                    keepBuilds(5)  
                    allowParallel(true)  
                    alwaysKeep(true)  
                }  
            }  
        }  
    }  
}
```

Build Triggers

Build after other projects are built

Build periodically

Poll SCM

Schedule

H/5 * * * *

Would last have run at Sunday, August 23, 2015 9:00:35 AM

Ignore post-commit hooks

```
job("running_java_with_docker") {
    scm {
        git {
            remote {
                name('origin')
                url ("https://github.com/camiloribeiro/gradle-docker")
            }
        }
    }
    triggers {
        scm 'H/5 * * * *'
    }
    steps {
        shell 'docker pull niaquinto/gradle:2.5'
        shell 'docker run -v $WORKSPACE/:/gradle -w /gradle niaquinto/gradle:2.5 clean build runInParallel'
    }
    publishers {
        publishHtml {
            report('build/reports/cucumber') {
                reportName('Cucumber Report')
                reportFiles('feature-overview.html')
                keepAll()
                allowMissing()
                alwaysLinkToLastBuild()
            }
        }
    }
}
```

Build

Execute shell

Command `docker pull niaquinto/gradle:2.5`

See [the list of available environment variables](#)

Execute shell

Command `docker run -v $WORKSPACE/:/gradle -w /gradle niaquinto/gradle:2.5 clean build runInParallel`

See [the list of available environment variables](#)

Add build step ▾

Post-build Actions

```
job("running_java_with_docker"
scm {
  git {
    remote {
      name('origin')
      url ("https://github.com/cam")
    }
  }
}
triggers {
  scm 'H/5 * * * *'
}
steps {
  shell 'docker pull niaquinto/gr'
  shell 'docker run -v $WORKSP
}
publishers {
  publishHtml {
    report('build/reports/cucumber') {
      reportName('Cucumber Report')
      reportFiles('feature-overview.html')
      keepAll()
      allowMissing()
      alwaysLinkToLastBuild()
    }
  }
}
```

Post-build Actions

Publish HTML reports

Reports

HTML directory to archive
build/reports/cucumber

Index page[s]
feature-overview.html

Report title
Cucumber Report

Add

```
configure { project -> project / publishers << 'join.JoinTrigger' {
    'joinProjects'{}
    'joinPublishers' {
        'hudson.plugins.parameterizedtrigger.BuildTrigger' {
            'configs' {
                'hudson.plugins.parameterizedtrigger.BuildTriggerConfig' {
                    'configs' {
                        'hudson.plugins.parameterizedtrigger.CurrentBuildParameters' {}
                    }
                    projects('promote_rpm_to_stage')
                    condition('SUCCESS')
                    triggerWithNoParameters('true')
                }
            }
        }
    }
    evenIfDownstreamUnstable('false')}
}
```

jenkinsci / job-dsl-plugin
forked from JavaPosseRoundup/job-dsl-plugin

Unwatch 220 Unstar 416 Fork 250

Job reference

Daniel Spilker edited this page 19 days ago · 175 revisions

This is the in-depth documentation of the methods available on inside the *job* part of the DSL.

Free Style Job

```
freeStyleJob(String name) { // since 1.30
    name(String name) // deprecated since 1.30

    // DSL specific methods
    using(String templateName)
    configure(Closure configBlock)
    previousNames(String regex) // since 1.29

    batchTask(String name, String script)
    blockOn(String projectNames)
    blockOn(Iterable<String> projectNames)
    blockOn(String projectNames, Closure closure) // since 1.36
    blockOn(Iterable<String> projectNames, Closure closure) // since 1.36
    blockOnDownstreamProjects()
    blockOnUpstreamProjects()
    checkoutRetryCount(int times = 3)
    compressBuildLog() // since 1.36
    concurrentBuild(boolean allowConcurrentBuild = true) // since 1.21
    customWorkspace(String workspacePath)
    deliveryPipelineConfiguration(String stageName, String taskName = null) // since 1.36
    description(String description)
    disabled(boolean shouldDisable = true)
    displayName(String displayName)
```

Pages 16

Find a Page...

Home

Deprecation Policy

Extending the DSL

Folder Reference

Frequently Asked Questions

Handling Credentials

IDE Support

Jenkins Job DSL Architecture

Job DSL Commands

Job reference

Migration

Real World Examples

The Configure Block

Tutorial Using the Jenkins Job DSL

User Power Moves

Show 1 more pages...



ubuntu



Java



MySQL



CentOS



fedora



NGINX



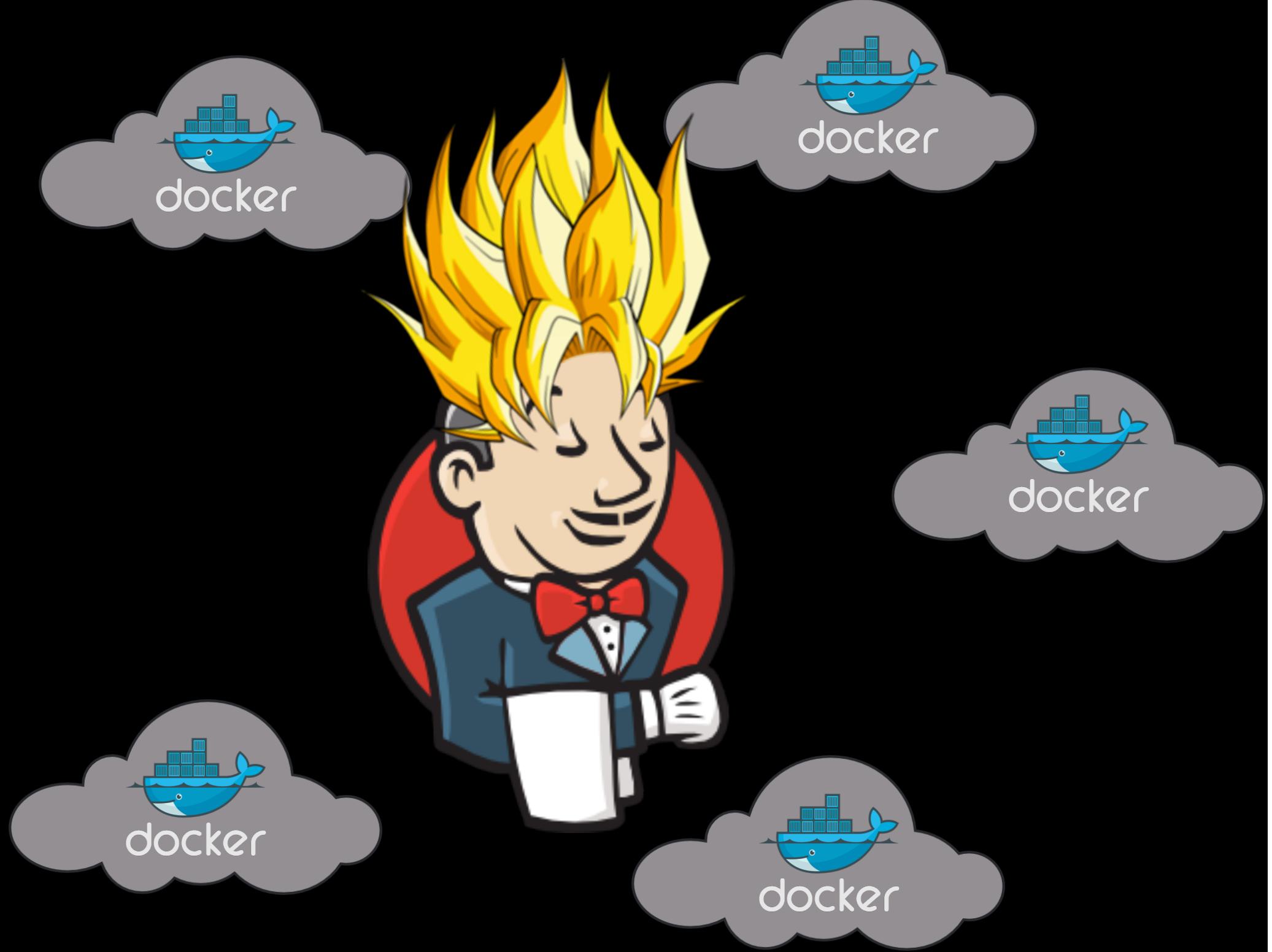
Cucumber



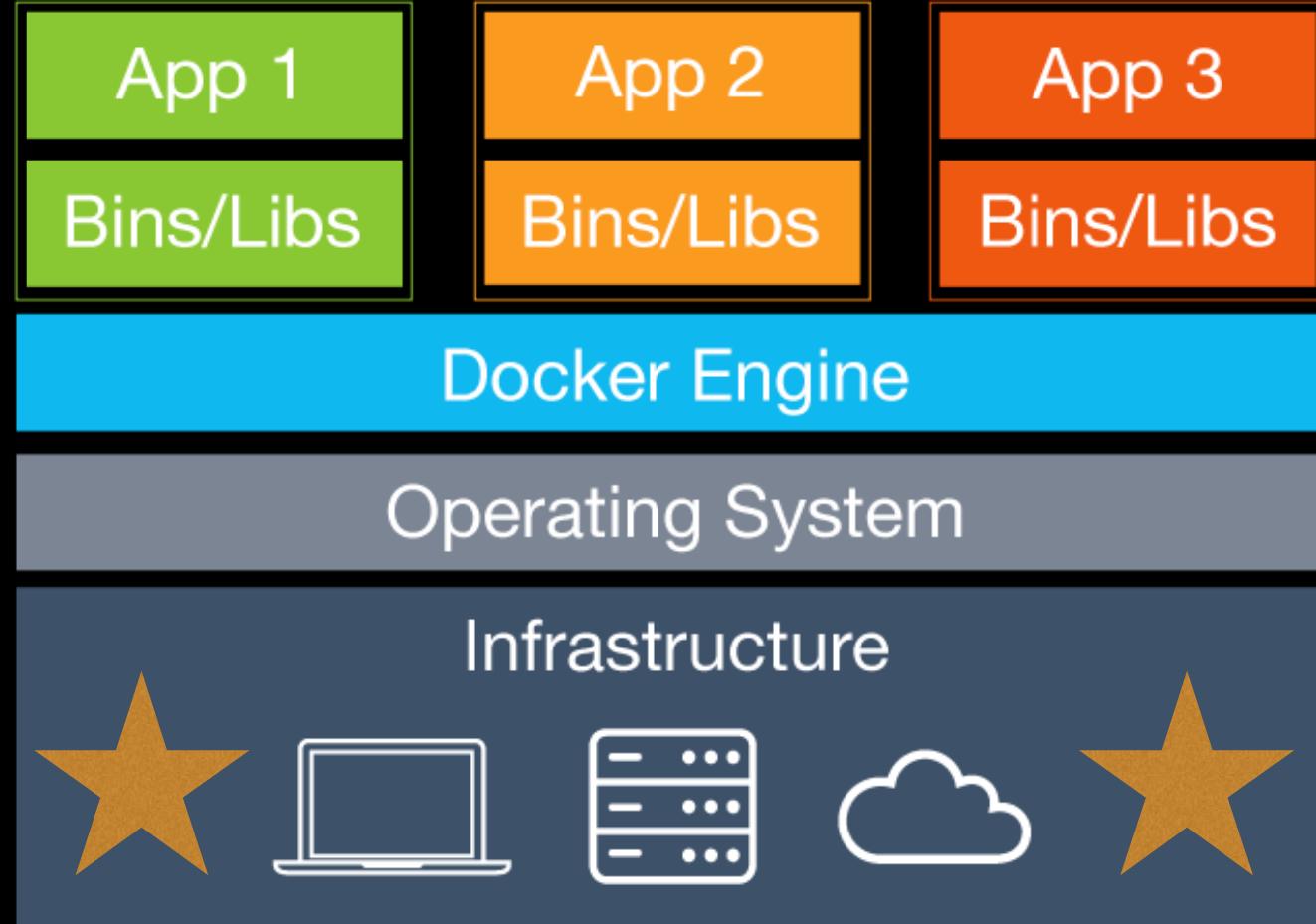
ERLANG



ubuntu

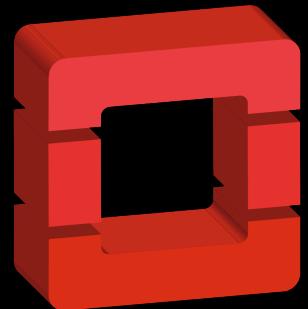








Google



openstack®

CLOUD SOFTWARE



Windows Azure



PUBLIC | AUTOMATED BUILD

cgswong/aws ☆

Last pushed: 10 days ago

Repo Info Tags Dockerfile Build Details

What is docker-aws

PASSED

Docker image for running the and `s3cmd`. It creates a Docker image containing all dependencies needed to run `aws` and `s3cmd`. That way, you can run these tools in a Docker container without setting the dependencies on the host system.

PUBLIC | AUTOMATED BUILD

google/cloud-sdk ☆

Last pushed: 11 hours ago

Repo Info Tags Dockerfile Build Details

cloud-sdk-docker

`google/cloud-sdk` is a Docker image bundling of the Google Cloud SDK:

- App Engine SDK for Go
- App Engine SDK for Java
- App Engine SDK for Python and PHP

PUBLIC | AUTOMATED BUILD

azukiapp/deploy-digitalocean ☆

Last pushed: 5 days ago

Repo Info Tags Dockerfile Build Details

This image is maintained by Azuki, the creator and maintainer of . is an orchestrator for Docker's containers in local development env

Using with azk

You can visit for more instructions on how to use this image with a

PUBLIC | AUTOMATED BUILD

heroku/go ☆

Last pushed: 2 days ago

Repo Info Tags Dockerfile Build Details

Heroku Go Docker image

For use with the heroku docker plugin.

Image tags

You can see all of the tags [here](#).

The `latest` tag will generally refer to the latest, possibly unsupported, release of Go, including betas and release candidates.

Once a final version is cut a separate tag will be used and updated.

 klarna/nerve public	0 STARS	1239 PULLS	DETAILS
 klarna/zookeeper public	0 STARS	451 PULLS	DETAILS
 klarna/synapse public	0 STARS	2572 PULLS	DETAILS
 klarna/jruby public	0 STARS	43 PULLS	DETAILS
 klarna/zookeeper-aws public	0 STARS	563 PULLS	DETAILS
 klarna/java8-el7 public	0 STARS	373 PULLS	DETAILS
 klarna/pulp-admin public	0 STARS	646 PULLS	DETAILS
 klarna/centos7-base public	0 STARS	274 PULLS	DETAILS
 klarna/ansible public	0 STARS	6 PULLS	DETAILS

Demo

Prayed 403 times today



New high score

camilo@camiloribeiro.com



**KEEP
CALM
AND
ASK
QUESTIONS**

thank you!

www.klarna.com/jobs

Klarna®