CLOUDBEES UNIVERSITY

CONTINUOUS DELIVERY WITH THE CLOUDBEES JENKINS PLATFORM

USER TRAINING

CLOUDBEES UNIVERSITY

TOC

- CI & CD, Jenkins Overview and CloudBees Introduction
- Jenkins Jobs/Projects and Setting up Freestyle Projects
- Monitoring and Organizing your jobs
- Plugins Management
- Working with Version Control
- Code Quality and Coverage Metrics
- Parameterized Builds
- Automated Deployment
- Folders & Folders Plus plugins and Security + RBAC plugins
- Playing with Git: Validated Merge Plugin and Pull-Request Builder for GitHub
- Pipeline (formerly known as Workflow)
- Template Plugin
- CloudBees Support
- Introduction to CloudBees Jenkins Operation Center

CLOUDBEES UNIVERSITY

CI / CD FUNDAMENTALS

CLOUDBEES UNIVERSITY

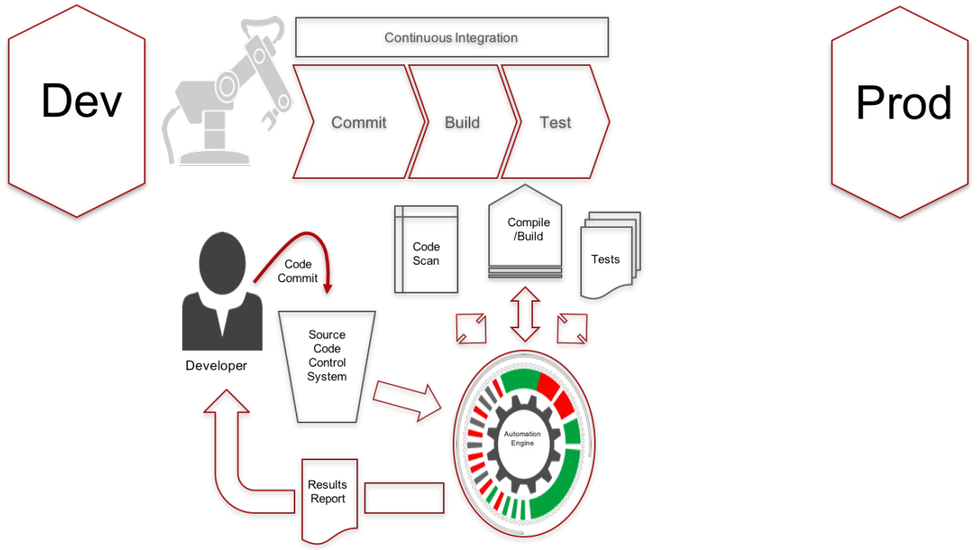
WHAT IS CONTINUOUS INTEGRATION (CI)?

- Original answer by Martin Fowler:
  - Integrate code often, at least daily, to make integration a non-event
  - Developers follow repository tip closely
- Modern Answer
  - Automate build and test executions of components
    - Build very often (e.g., per commit) to detect regressions quickly
    - Test often (unit and integration tests)
  - Automate regular code quality analysis executions

CLOUDBEES UNIVERSITY

CONTINUOUS INTEGRATION (CI)

- Automate Build & Test phases
- Continually build and integrate



CLOUDBEES UNIVERSITY

WHAT IS CONTINUOUS DELIVERY (CD) ?

- While CI lets you automate the software build, scan and test process, CD automates the full application delivery pipeline taking new features and code from development to staging to production
- With automated building, testing, environment provisioning, and deployment, CD becomes the foundation of your DevOps transformation

CLOUDBEES UNIVERSITY

WHAT IS CONTINUOUS DELIVERY (CD) ?

The ability to get:

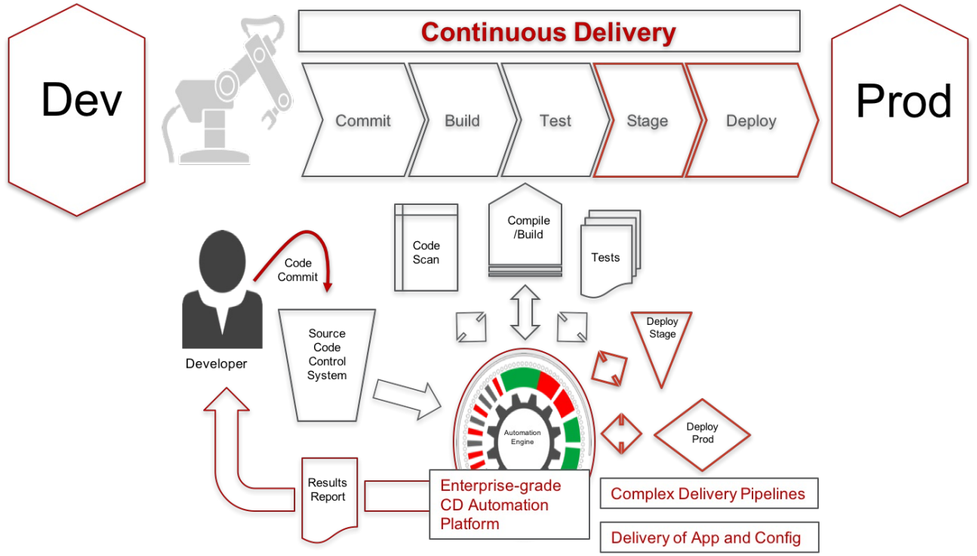
- Features
- Configuration changes
- Bug fixes
- Experiments

into production or into the hands of users safely & quickly in a sustainable way.

CLOUDBEES UNIVERSITY

CONTINUOUS DELIVERY (CD)

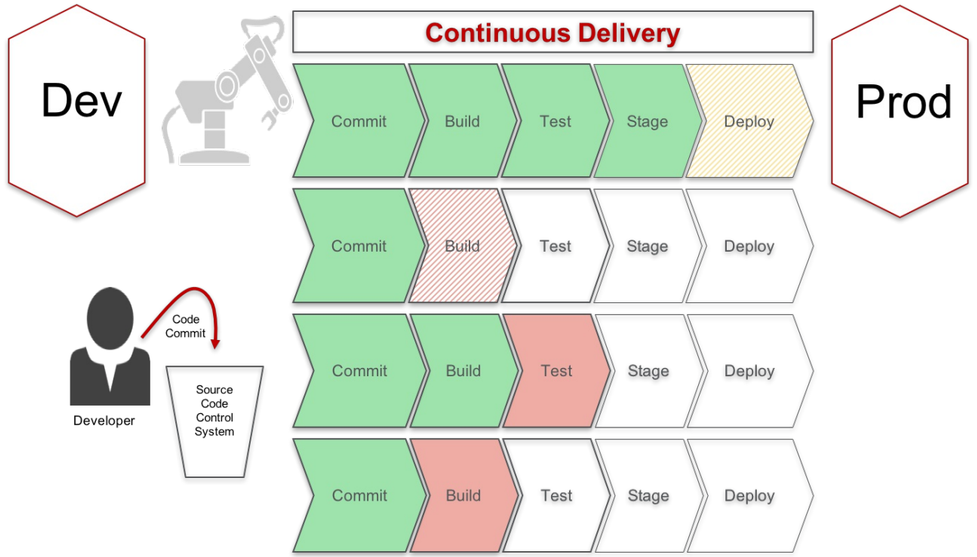
- CD starts with CI
- It then extends automation across the entire application lifecycle



CLOUDBEES UNIVERSITY

CONTINUOUS DELIVERY (CD)

- Enables continuous execution and monitoring of the entire software development pipeline



CLOUDBEES UNIVERSITY

CONTINUOUS DELIVERY (CD), WHY DO IT ?

- Make releases painless, low risk events
- Reduce time to market
- Increase software quality & stability
- Reduce cost of ongoing software development
- Increase customer & employee satisfaction
- Speed up the feedback loop

CLOUDBEES UNIVERSITY

RELEASE CADENCE

How long would it take to your organization to deploy a change that involves just one single line of code?

CLOUDBEES UNIVERSITY

JENKINS OVERVIEW

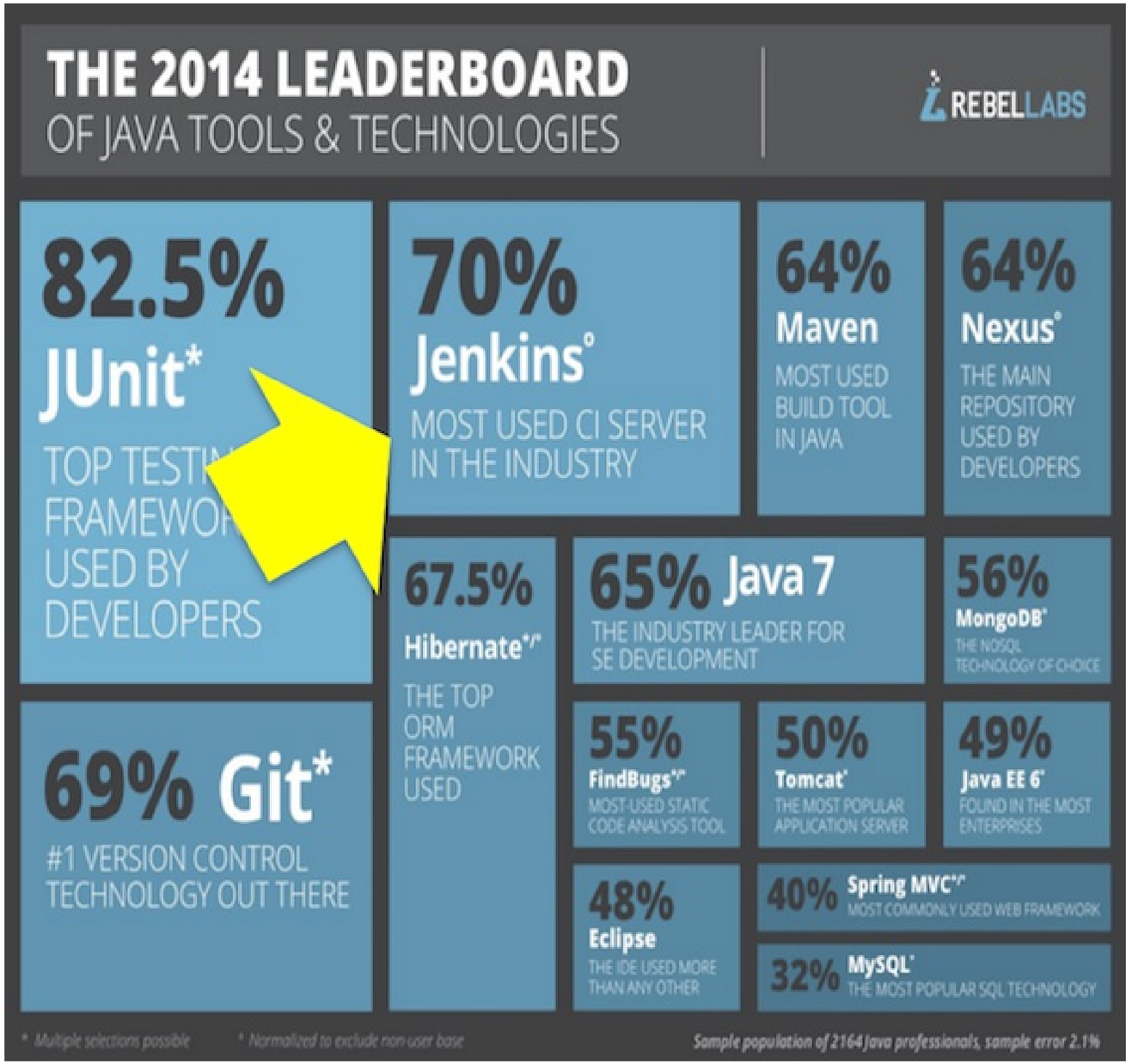
CLOUDBEES UNIVERSITY

MEET JENKINS…

- #1 Continuous Integration and Delivery server
- Created by Kohsuke Kawaguchi
- An independent and active community (jenkins-ci.org)
- 10 years old
- 500+ releases to date
- 100,000 active installations
- 300,000 Jenkins servers
- 1,200+ plugins

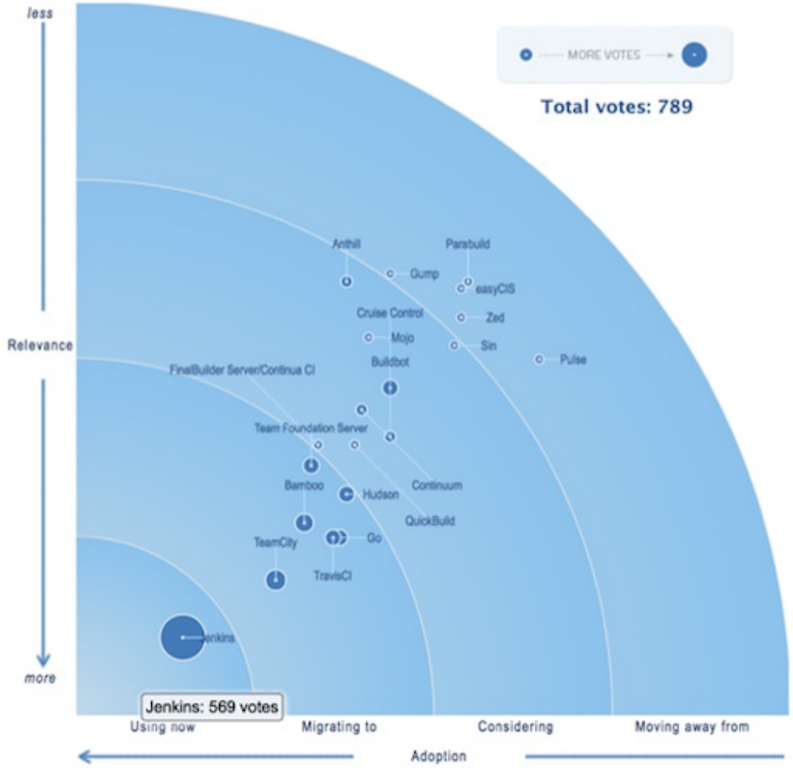
Source: 2014 Java Tools and Technologies Landscape – Rebel Labs

CLOUDBEES UNIVERSITY



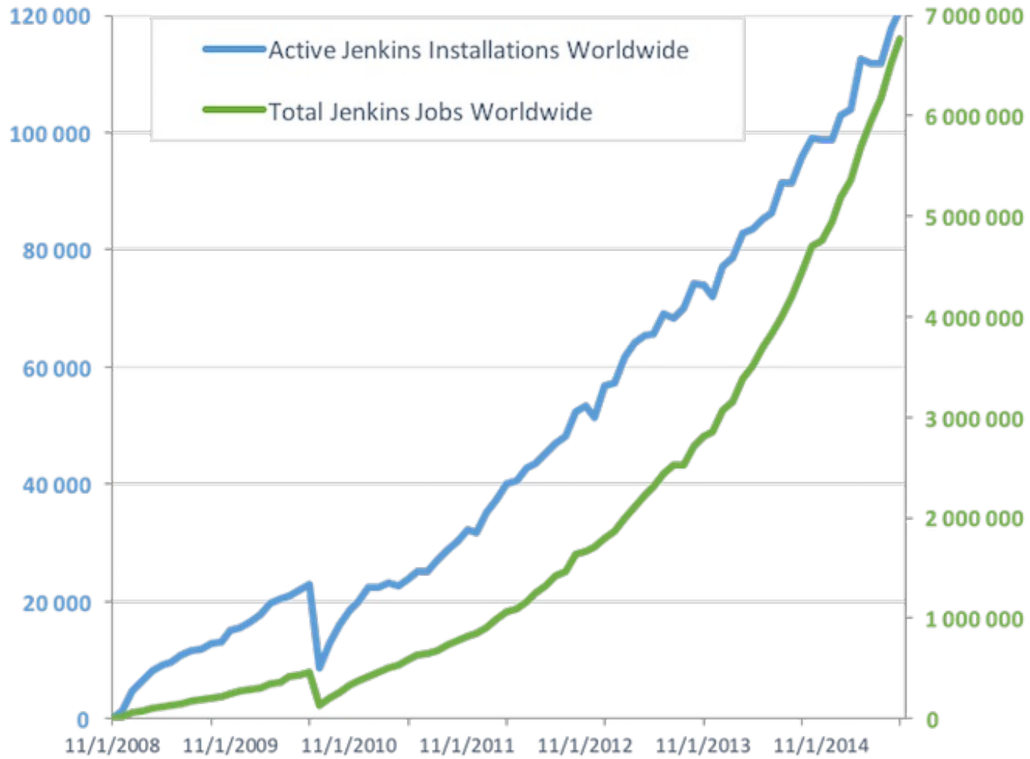
CLOUDBEES UNIVERSITY

JENKINS POPULARITY: THROUGH THE ROOF



http://stats.jenkins-ci.org/jenkins-stats

http://www.infoq.com/research/ci-server

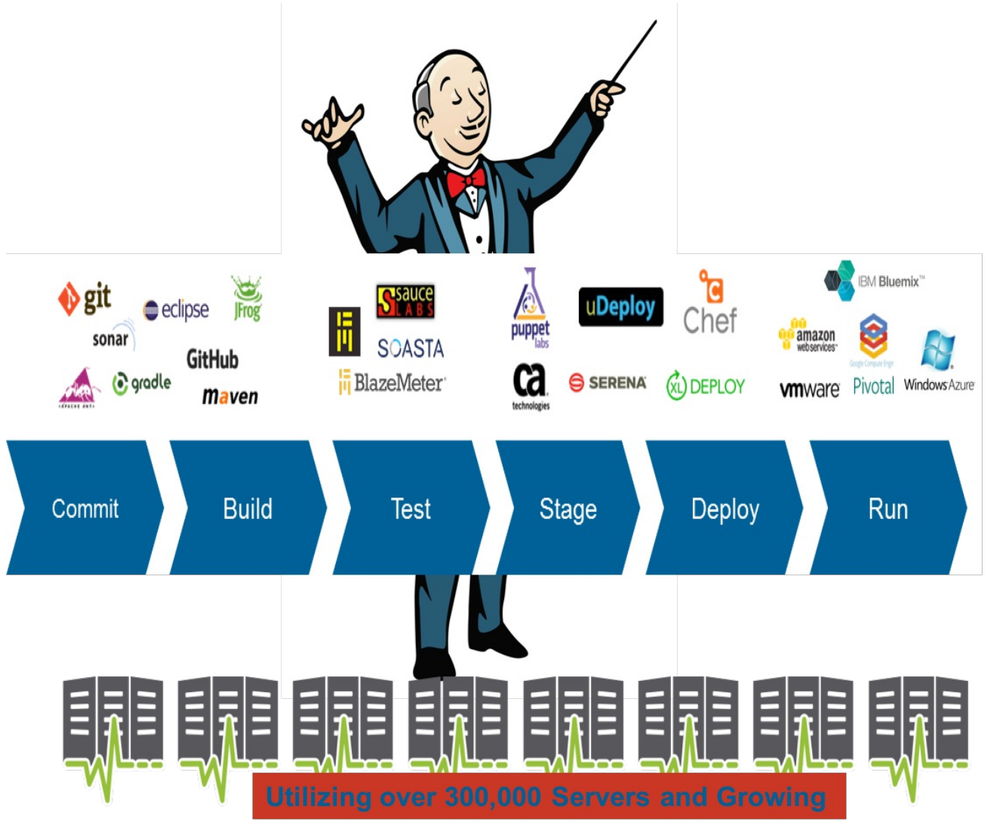


CLOUDBEES UNIVERSITY

WORLDWIDE ADOPTION

CLOUDBEES UNIVERSITY

JENKINS IS THE CD ORCHESTRATOR



CLOUDBEES UNIVERSITY

CLOUDBEES INTRODUCTION

CLOUDBEES UNIVERSITY

CLOUDBEES AND THE JENKINS COMMUNITY

- Kohsuke Kawaguchi
  - Community leader and CloudBees' CTO
- Code and Releases
  - CloudBees partners with the community on development
  - CloudBees engineers contribute a majority of Jenkins OSS code
  - CloudBees partners with the community on releases
  - CloudBees contributes fixes back to the community
- Produces Jenkins Quarterly Newsletter
- Conducts Jenkins User Conferences

CLOUDBEES UNIVERSITY

CLOUDBEES INC.

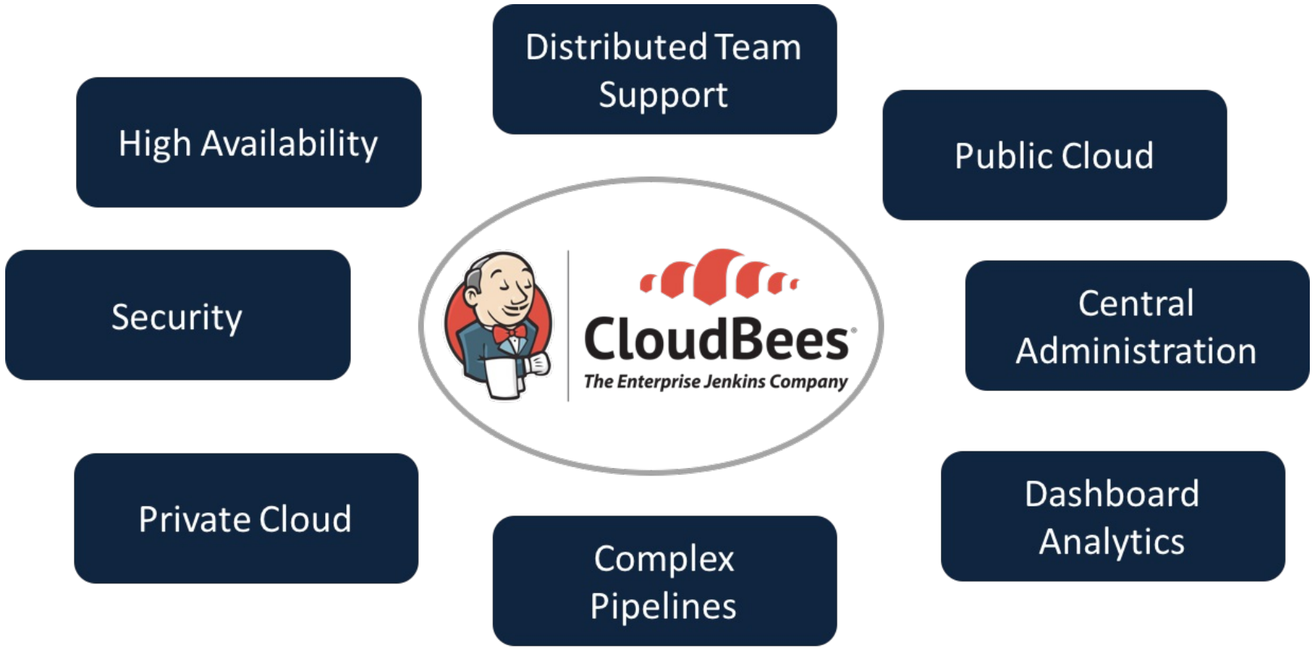
- Incorporated in April 2010
- ~ 200 people as of July 2016
- Headcount and revenue tripled between January 2015 and July 2016
- Presence in 14 countries
- Offices:
  - USA:
  - Raleigh, Richmond, San Jose
- Europe:
  - Sevilla (Spain), Reading (UK), Neuchatel (Switzerland)

CLOUDBEES UNIVERSITY

FOLLOW-THE-SUN 24/7 SUPPORT LOCATIONS

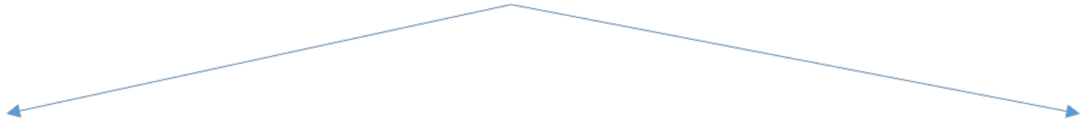
CLOUDBEES UNIVERSITY

WE DELIVER JENKINS @ ENTERPRISE SCALE



CLOUDBEES UNIVERSITY

CLOUDBEES JENKINS PLATFORM



Enterprise Edition
Scalable, Secure and Manageable

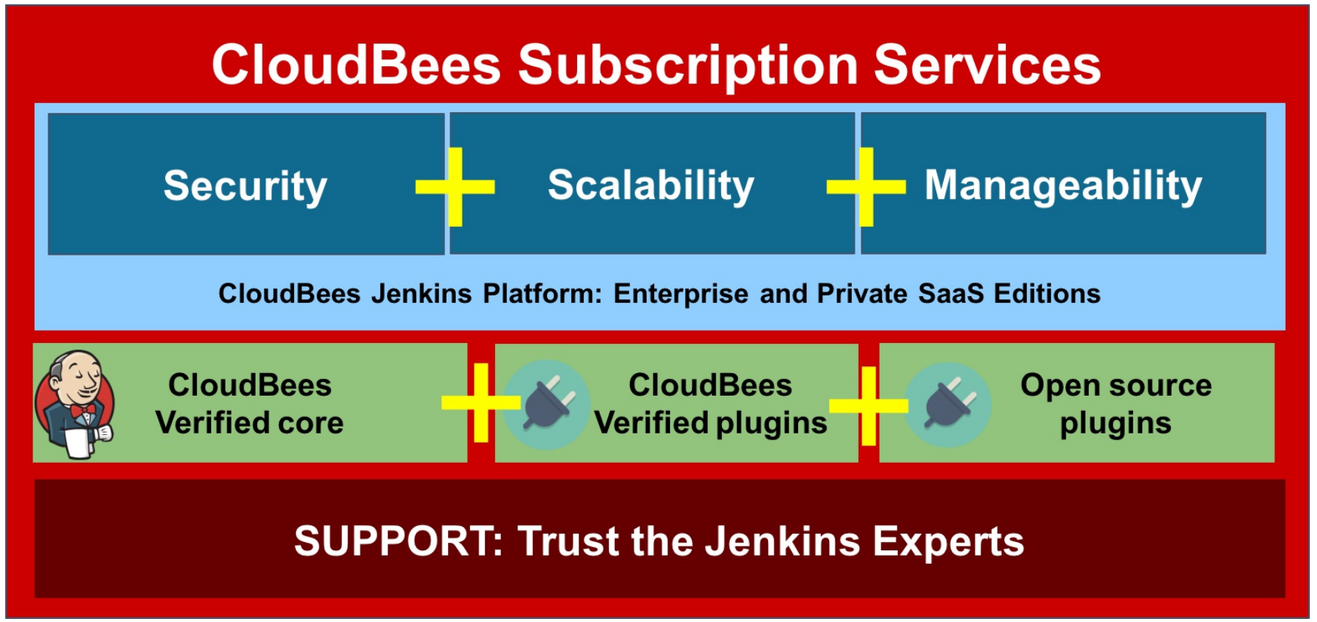
- High Availability
- Cluster Operations – multi-master environment control
- Shared Build Nodes
- Role Based Access Control and single sign-on
- Pipeline management
- Templates to automate best practices
- Analytics and actionable information

Private SaaS Edition
Continuous Delivery as a Service

- Deploy on your private cloud or VPC
- Federated, shared services across projects
- Deploy new Jenkins services in seconds
- Resilient cloud-based HA/DR
- Automated storage backups
- Optimize infrastructure cost & usage
- Operational visibility & analysis
- Oversight of total workflow & complex pipelines

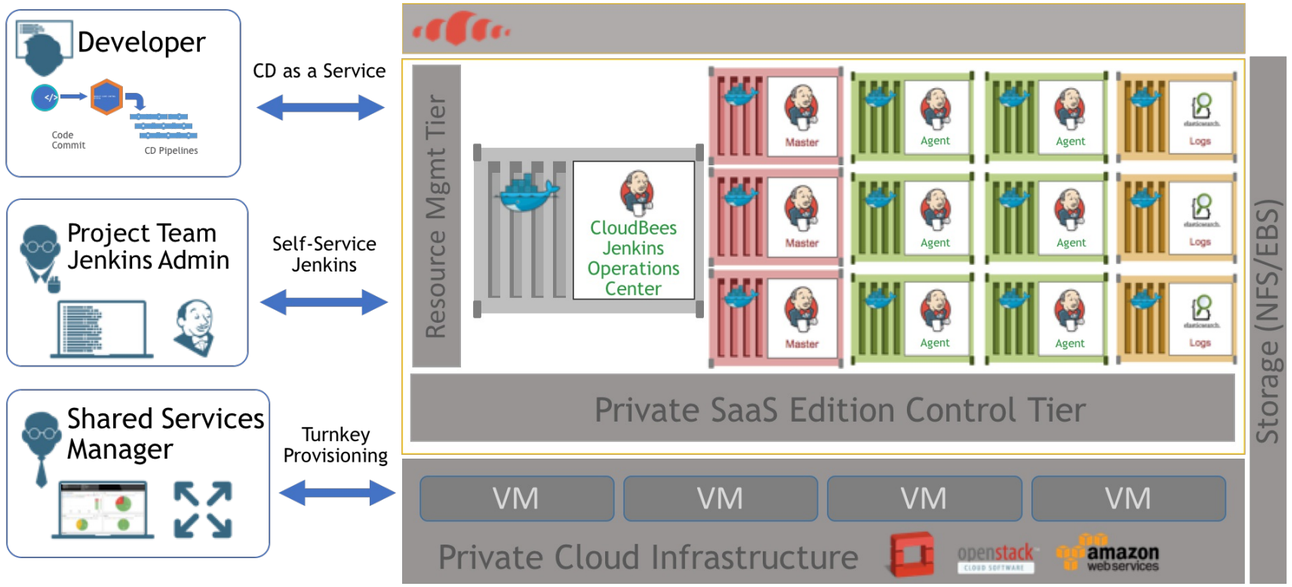
CLOUDBEES UNIVERSITY

CLOUDBEES JENKINS PLATFORM: ENTERPRISE EDITION



CLOUDBEES UNIVERSITY

CLOUDBEES JENKINS PLATFORM: PRIVATE SAAS EDITION



CLOUDBEES UNIVERSITY

JENKINS SERVICES FROM CLOUDBEES

Jenkins Certified Training

Jenkins Certification

Jenkins Consulting

Jenkins Newsletter

Jenkins User Conferences

CLOUDBEES UNIVERSITY

CLOUDBEES SUBSCRIPTION MODEL

- CloudBees sells subscriptions that entitle you to receive support for:
  - CloudBees Jenkins Platform
  - OSS Jenkins including 1,200+ Open Source Plugins
- Customers are charged an annual subscription fee per installation
  - Multi-year subscriptions are available
- A typical service subscription includes:
  - Software updates, bug fixes, and upgrades
  - Technical support
  - stable versions, stable APIs, and more

CLOUDBEES UNIVERSITY

CLOUDBEES RESOURCES

- Customer Engagement
  - Support
  - Knowledge Base
  - Diagnostics
- Professional Services
  - Architecture Assessment & CD Guidance
  - Bootstrap Implementation Services
  - Migration Assistance
- Certified Partners
  - Training
  - DevOps transformation
  - DevOps integration
  - Custom Development

CLOUDBEES UNIVERSITY

JENKINS JOBS/PROJECTS

CLOUDBEES UNIVERSITY

WHAT IS A JENKINS JOB/PROJECT

- Jobs are at the heart of the Jenkins build process.
- A Jenkins job is most of the time a particular task or step in your build process.
- An application may require several jobs
- A single job may have multiple build steps

CLOUDBEES UNIVERSITY

KEY JENKINS PROJECT (JOB) TYPES

- Pipeline Project (Brand New!)
- Freestyle Project
- Maven Project
- Monitor External Job
- Several more types supported

CLOUDBEES UNIVERSITY

FREESTYLE PROJECTS

- Most flexible
- Build any type of projects (Ant, Maven, Makefile, Shell script…)
- Supports most reporting plugins

CLOUDBEES UNIVERSITY

MAVEN PROJECTS

- Optimized for Maven 2/3 projects
- Jenkins reads the pom.xml file to make setting up the project easier.
- Build projects in the correct order based on the Maven dependencies
- Understands multi-module projects
- Default reporting of tests and code analysis tools

CLOUDBEES UNIVERSITY

SETTING UP FREESTYLE PROJECTS

CLOUDBEES UNIVERSITY

NEW FREESTYLE PROJECT: GENERAL INFO

- Discard old build data
  - Useful to cap the disk consumption
  - You can mark some to be kept forever
- Configure specific tools for the build to run with
  - Specify JDK
  - Specify Maven/Ant versions

CLOUDBEES UNIVERSITY

NEW FREESTYLE PROJECT: BUILD STEPS

- Build steps do the actual work
- Out of the box, many types available:
  - Unix shell or Windows batch script
  - Invoke an Ant script
  - Invoke a Maven script
- You can have several build steps
  - A shell script followed by an Ant or Maven build
  - An Ant build followed by a Maven build
- Can be reordered via Drag & Drop
- Plugins can add new build steps

CLOUDBEES UNIVERSITY

ENVIRONMENT VARIABLES

- Jenkins also sets environment variables, eg.
  - BUILD_NUMBER (eg “123”)
  - JOB_NAME (eg. “tax-api-unit-tests”)
  - BUILD_TAG (eg. “Jenkins-tax-api-unit-tests-123”)
  - SVN_REVISION or CVS_BRANCH
- You can use these in your build script

CLOUDBEES UNIVERSITY

POST-BUILD ACTIONS

- Collect the results of builds
  - Archiving
  - Reporting
  - Trigger other builds
  - Notifications

CLOUDBEES UNIVERSITY

POST-BUILD ACTION: ARCHIVING

- Store build artifacts for future reference
- Use Ant-style paths build/**/*.jar to determine what files to archive Stored artifacts are displayed on a build's page

CLOUDBEES UNIVERSITY

POST-BUILD ACTION: JUNIT REPORTS

- The build needs to run the tests and generate an XML JUnit report
- Use Ant-style path expressions
- Results appear on the project home page

CLOUDBEES UNIVERSITY

POST-BUILD ACTION: JAVADOC API

- The build needs to generate Javadoc
- Jenkins will capture javadoc
- Javadoc link appears on the build home page

CLOUDBEES UNIVERSITY

POST-BUILD ACTION: E-MAIL NOTIFICATION

- Email notification strategies
  - Sent when a build fails or is deemed “unstable”
  - And on the first successful build after a series of failures
- To fine-tune the behaviour, use “email-ext”

CLOUDBEES UNIVERSITY

POST-BUILD ACTION: TRIGGER OTHER BUILDS

- Specify other builds to be started when this build succeeds
- Useful for cascading builds
- Different view of the same data as “build trigger”

CLOUDBEES UNIVERSITY

LAB EXERCISES:

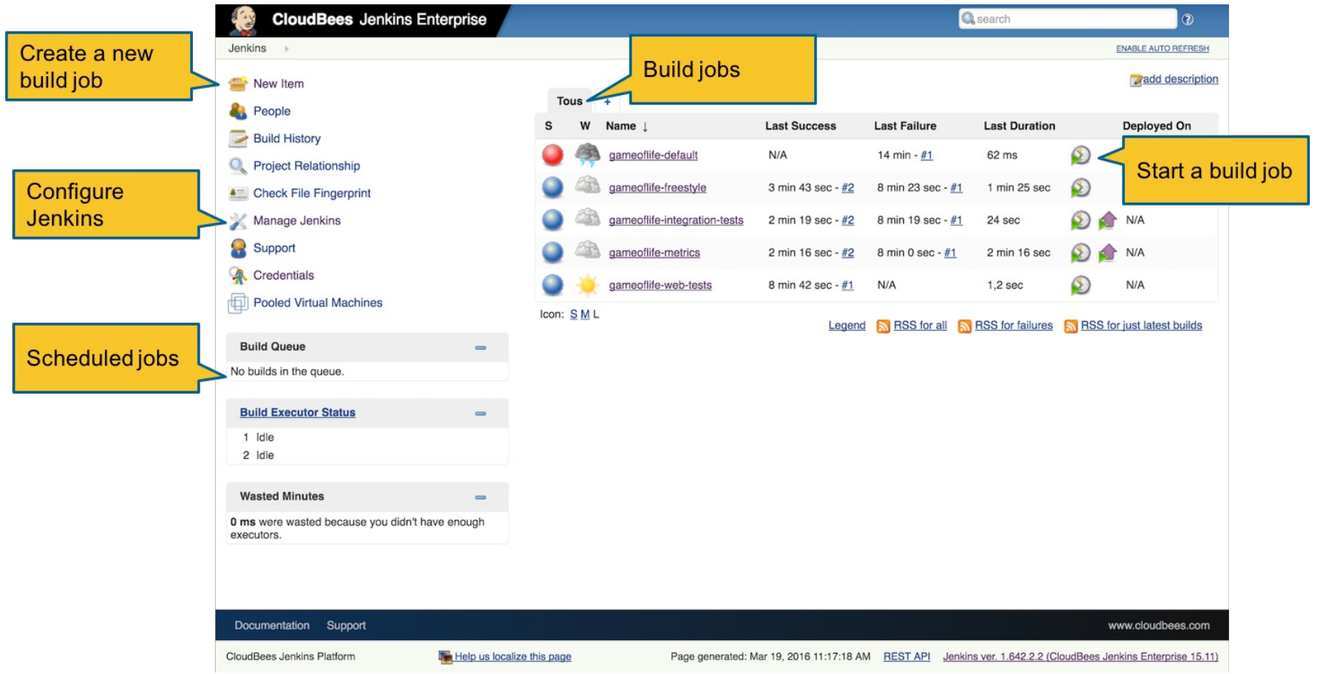
- Lab 1: Creating A Freestyle build Job
- Lab 2: Creating A Maven build Job

CLOUDBEES UNIVERSITY

MONITORING BUILD JOBS

CLOUDBEES UNIVERSITY

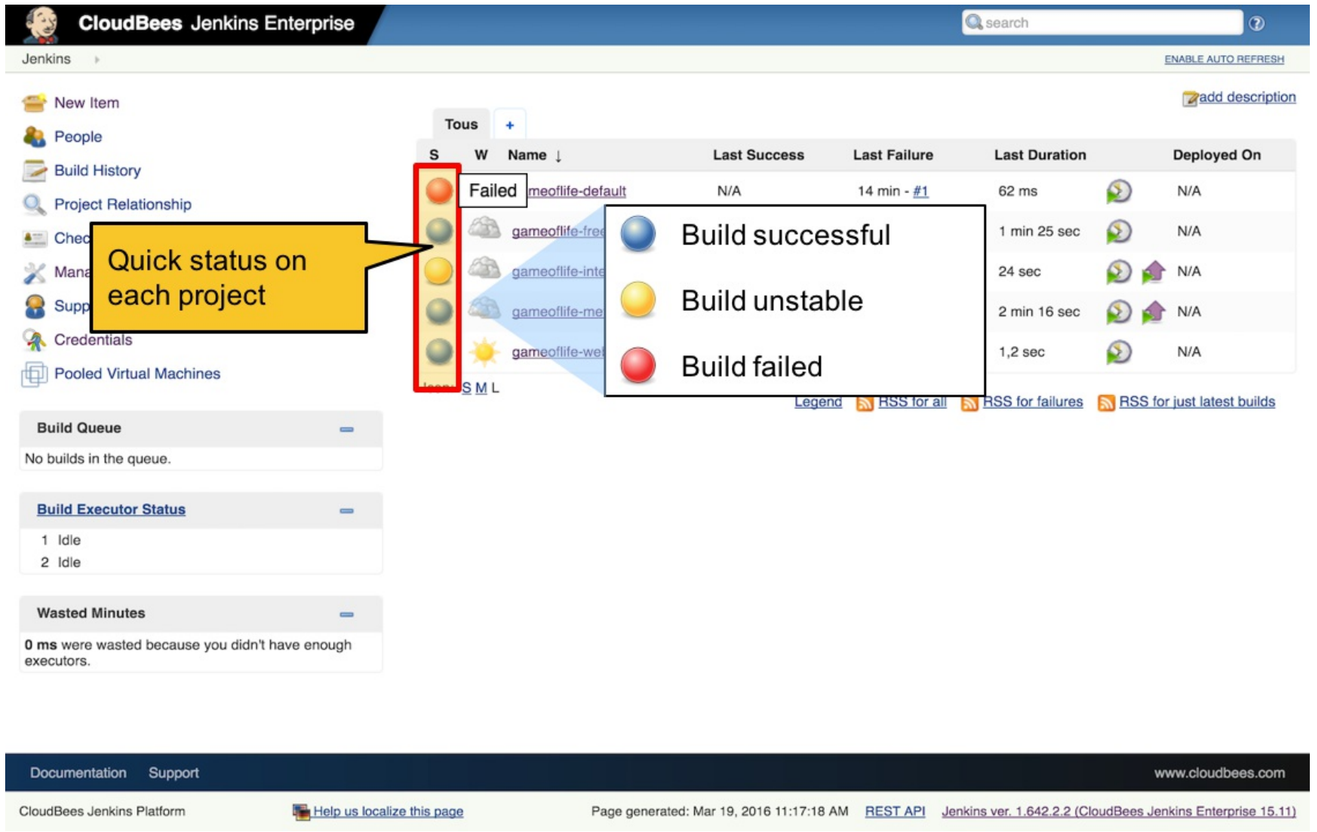
THE JENKINS DASHBOARD



CLOUDBEES UNIVERSITY

MONITORING BUILD JOBS

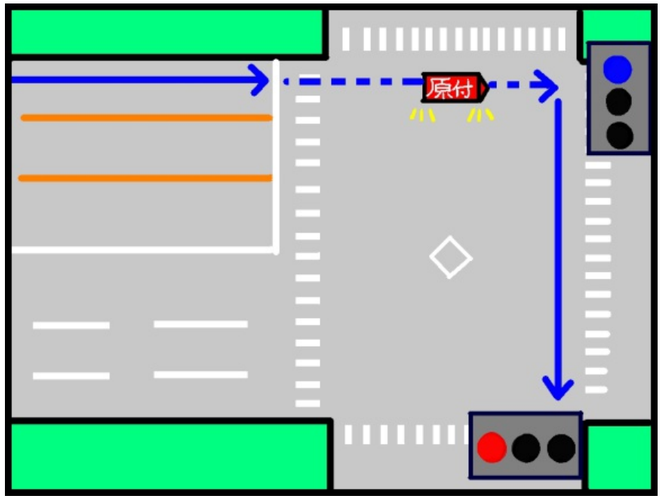
- What is the last build result?

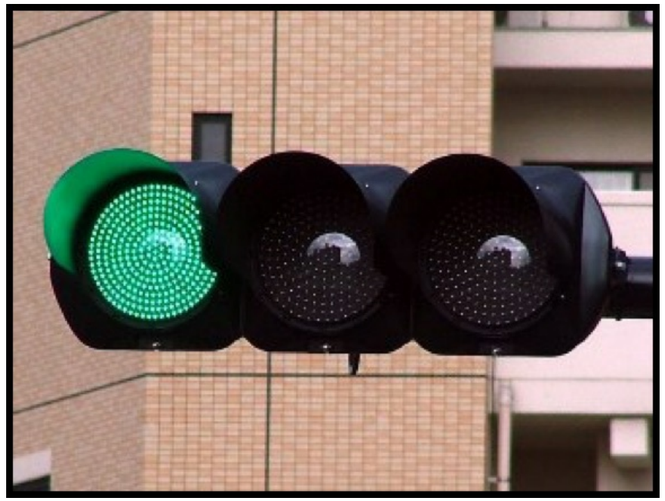


CLOUDBEES UNIVERSITY

WHY BLUE?

- In Japan, it is really called "blue light"
  - So people draw them as such
  - Even though the actual color is green!





CLOUDBEES UNIVERSITY

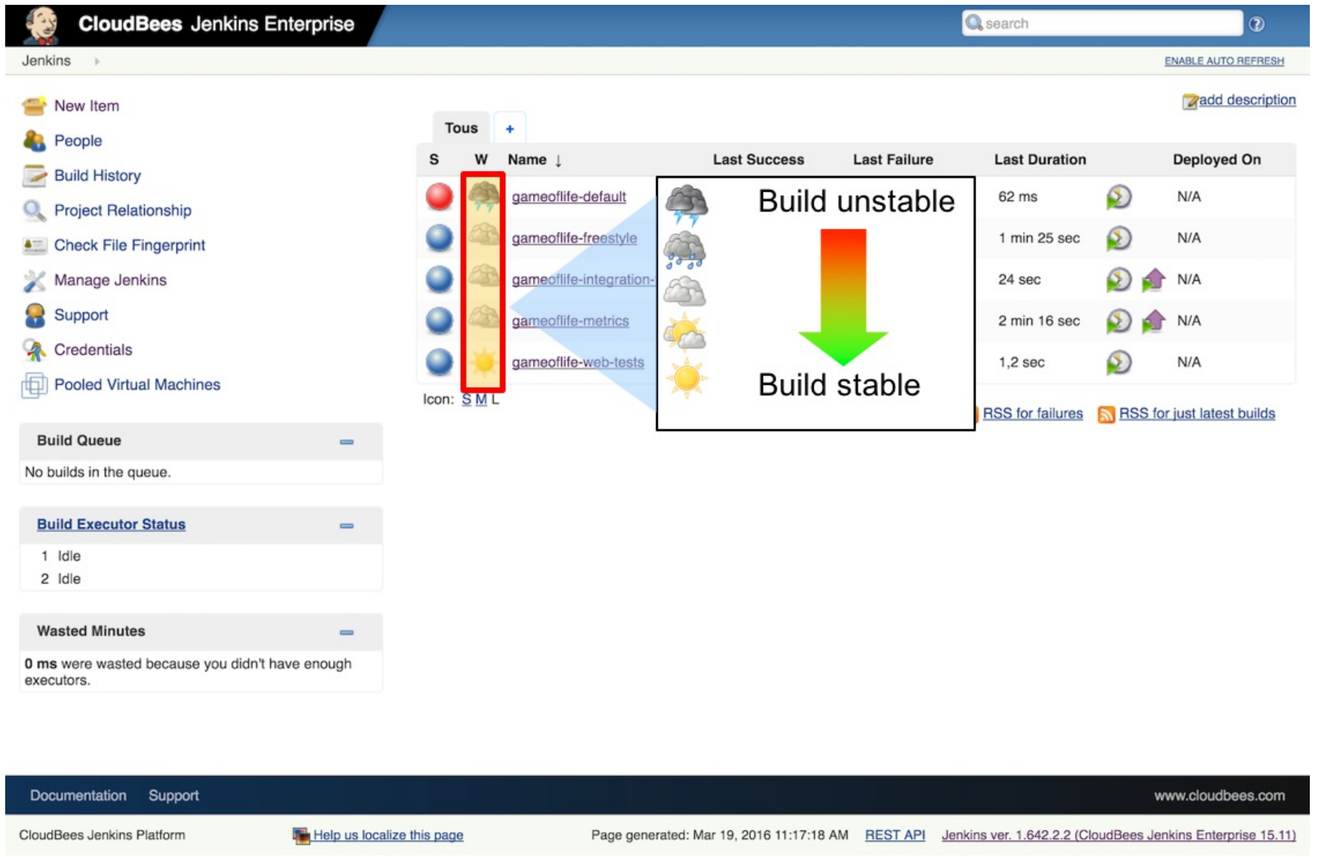
BENEFITS: QUICK OVERVIEW

- KISS principle:
  - Color for last build status
  - Weather for the build status trend
- One screen to see them all

CLOUDBEES UNIVERSITY

MONITORING BUILD JOBS

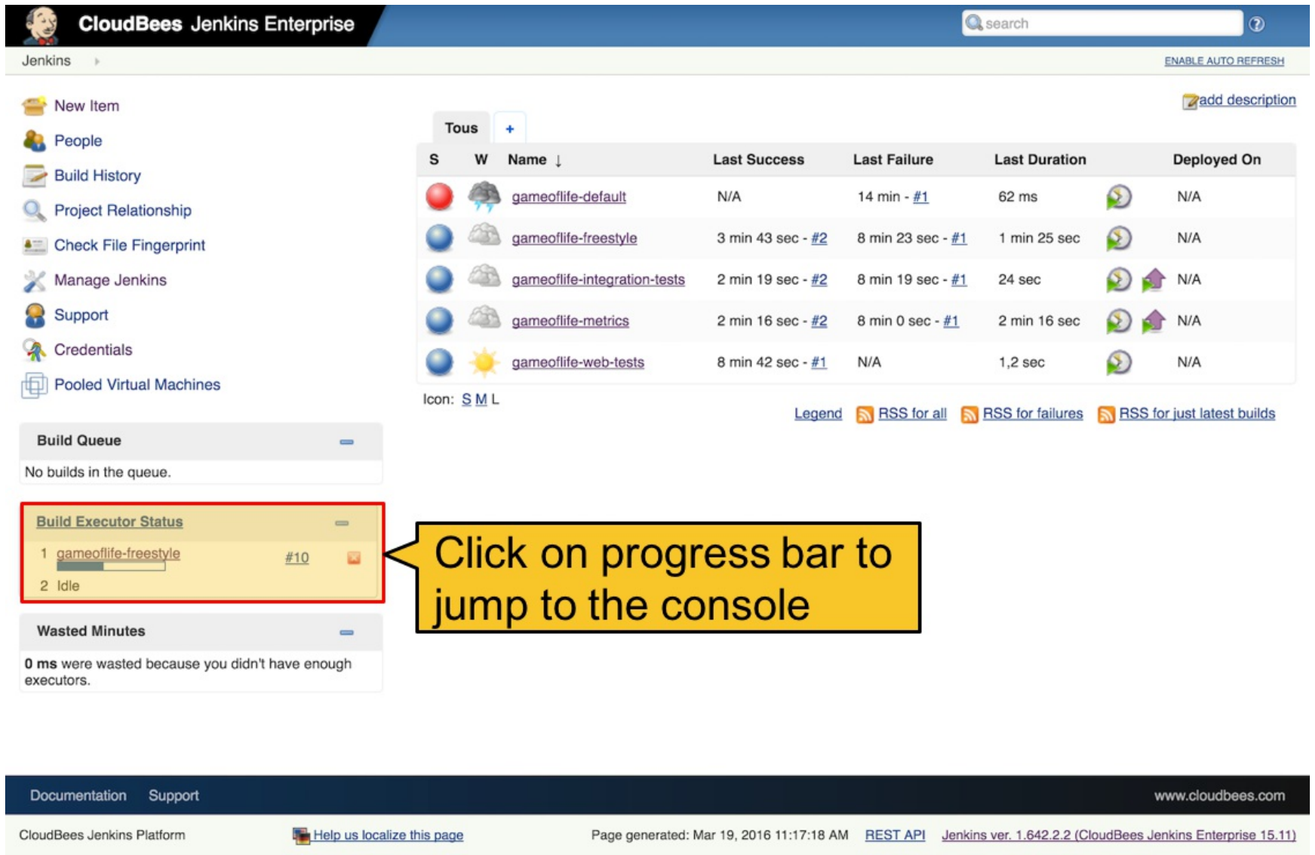
- How my jobs are behaving for the last 5 builds?



CLOUDBEES UNIVERSITY

MONITORING BUILD JOBS

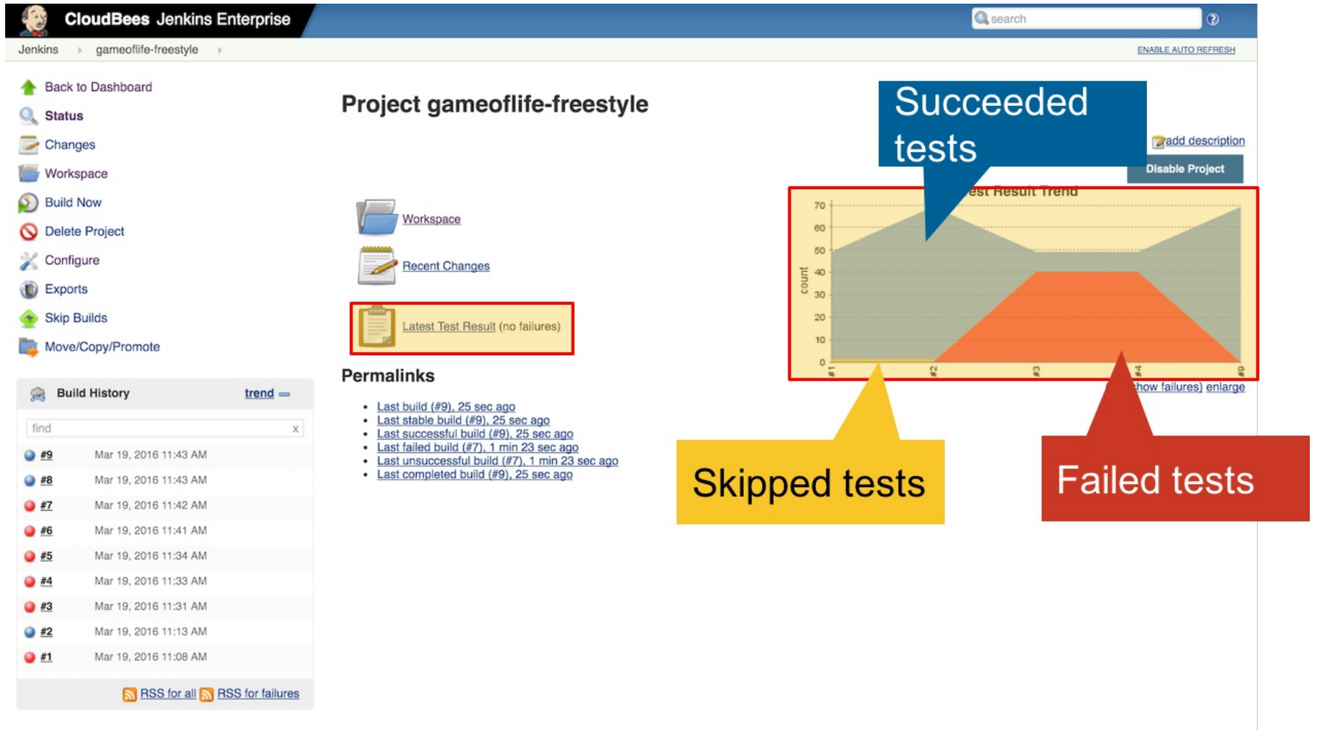
- What jobs are currently executed?



CLOUDBEES UNIVERSITY

BUILD DETAILS

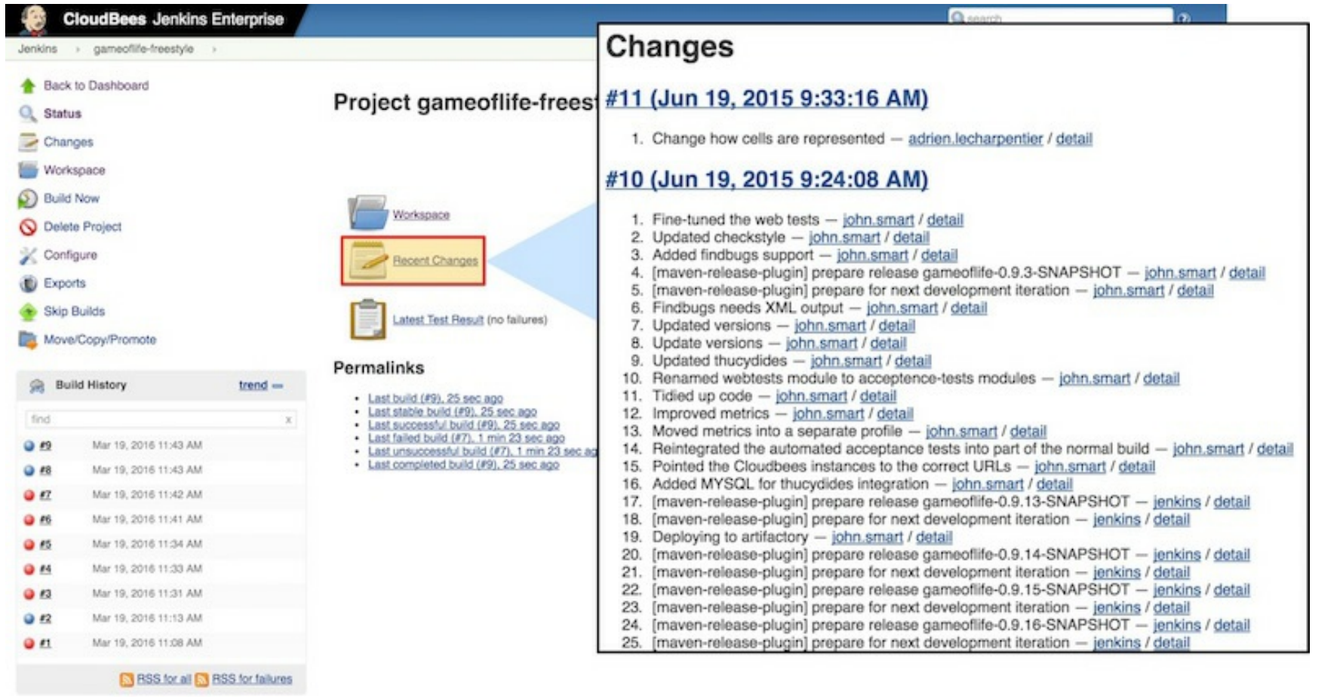
- See the tests results:



CLOUDBEES UNIVERSITY

BUILD DETAILS

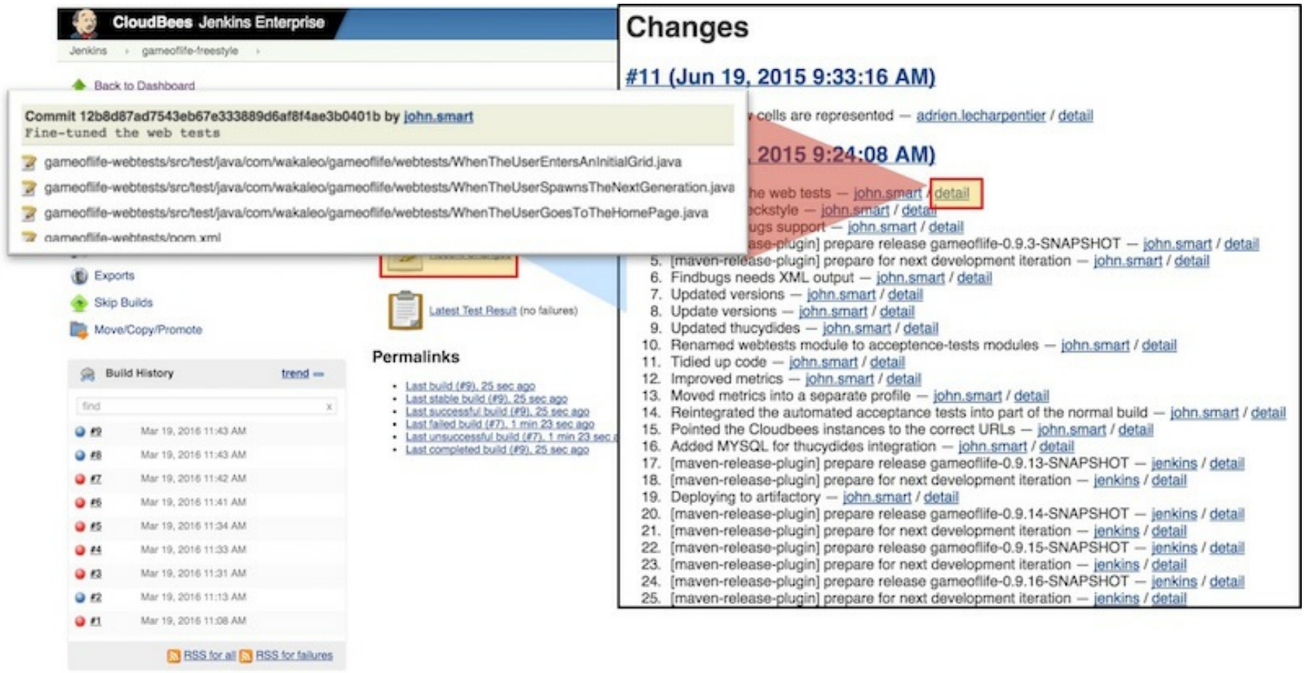
- See what has changed:



CLOUDBEES UNIVERSITY

BUILD DETAILS

- See what has changed:



CLOUDBEES UNIVERSITY

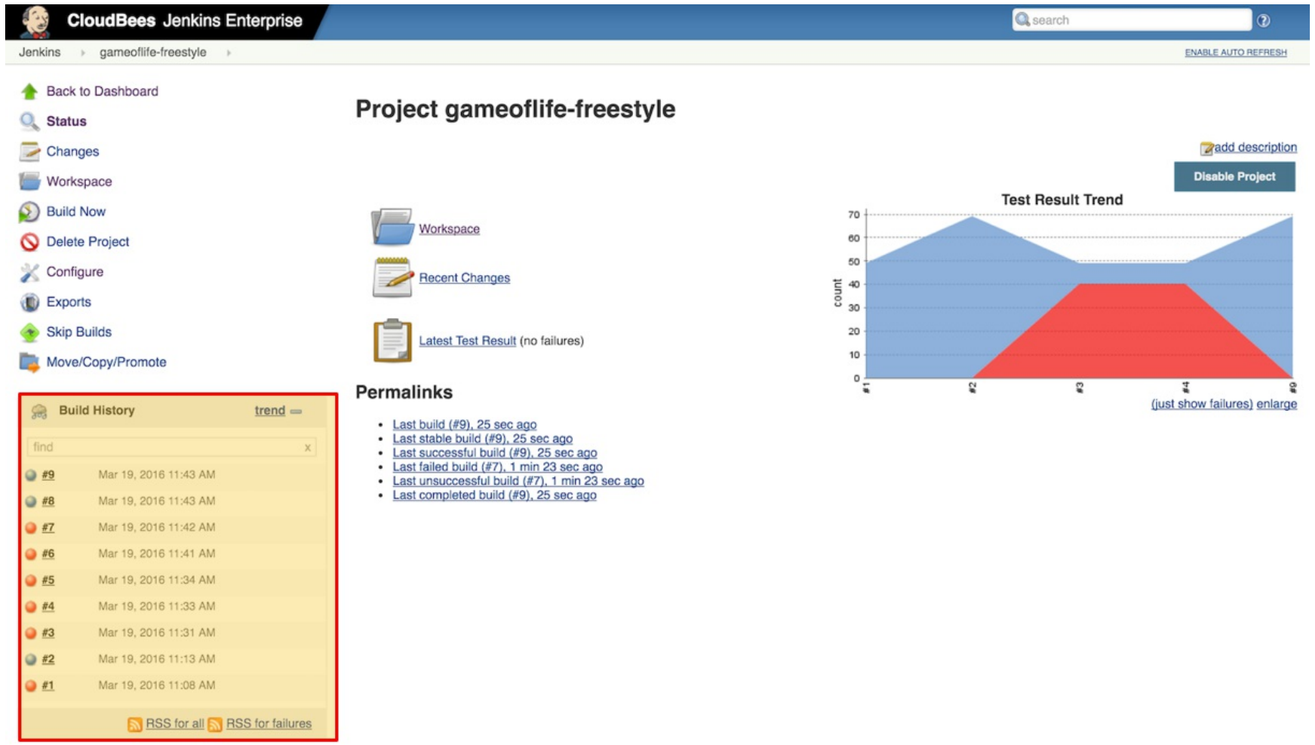
BUILD DETAILS

- Download latest archived artifact:

CLOUDBEES UNIVERSITY

BUILD DETAILS

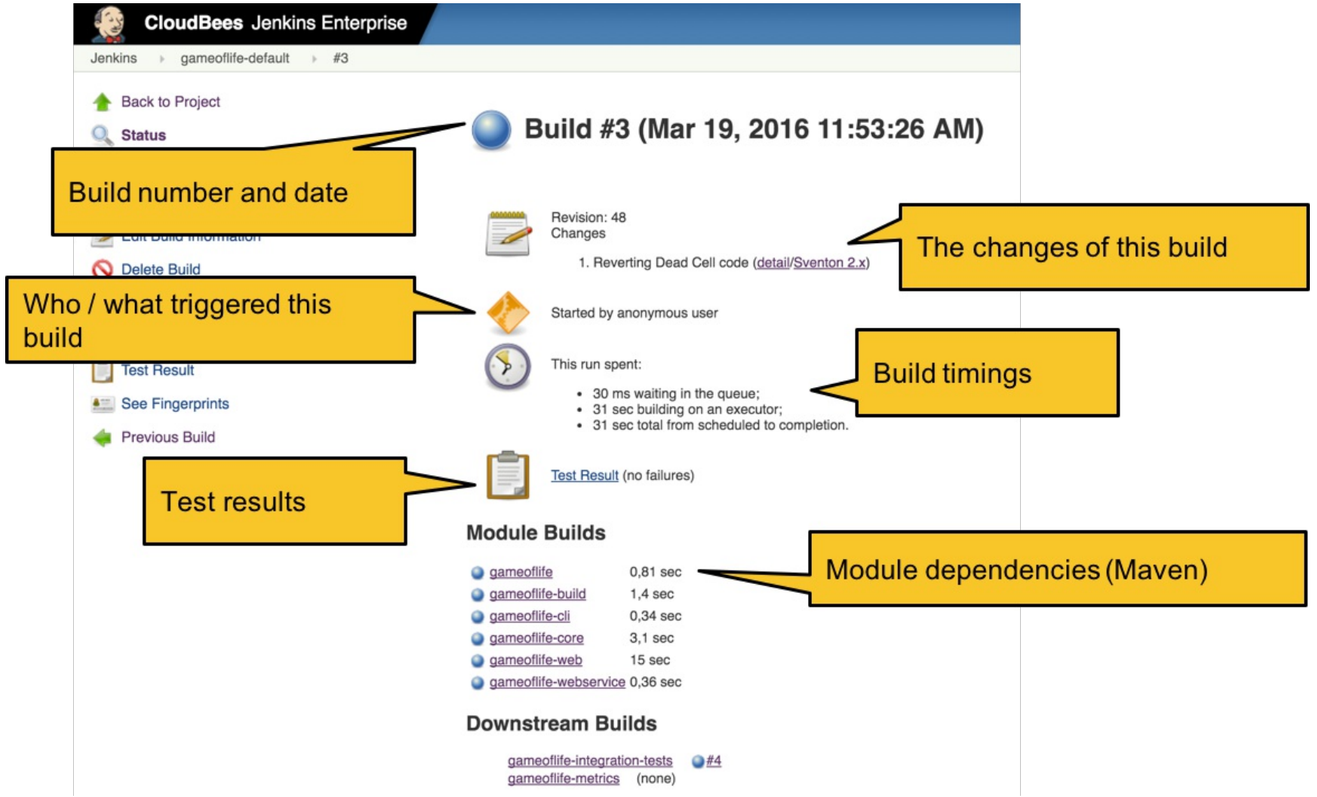
- Build history:



CLOUDBEES UNIVERSITY

BUILD DETAILS

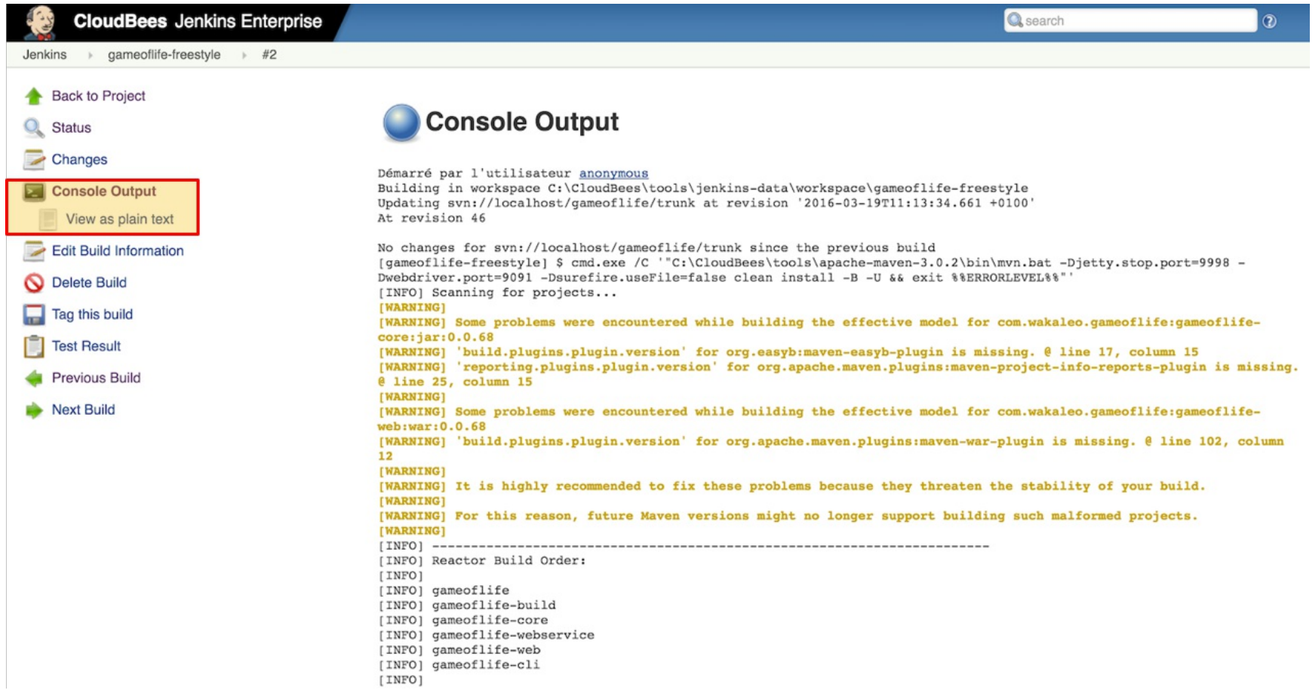
- View the details of any particular build:



CLOUDBEES UNIVERSITY

BUILD DETAILS

- View the output of the build:



CLOUDBEES UNIVERSITY

ORGANIZING BUILD JOBS

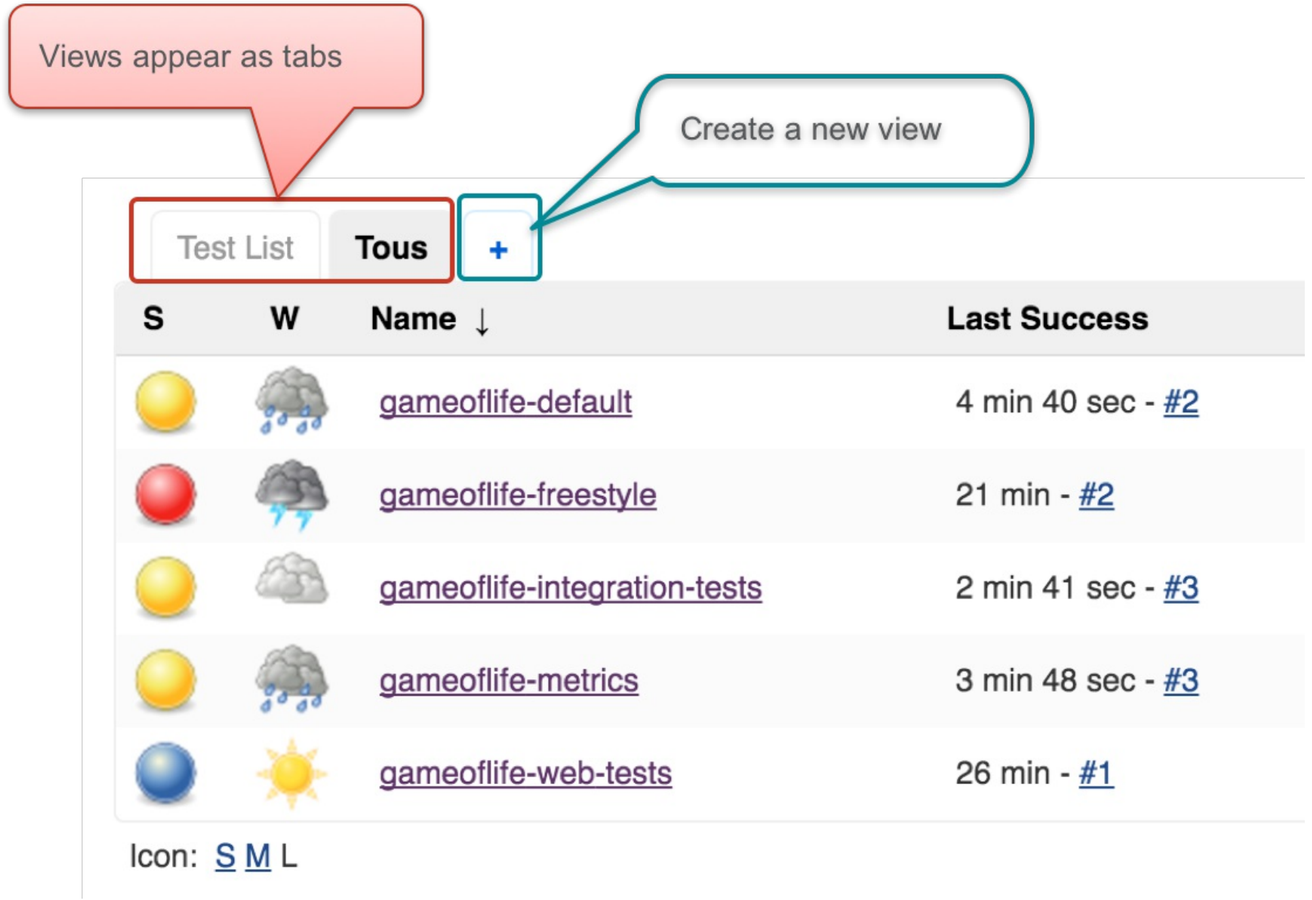
CLOUDBEES UNIVERSITY

ORGANIZING YOUR BUILD JOBS

- Don’t do everything in one job:
  - Create many job and divide things up
- Progressive refinement:
  - Do cheap things first and get fast feedback:
    - Compilation, unit tests
  - Do expensive things later:
    - Only after cheap things run successfully
    - Integration tests, cross-platform tests

CLOUDBEES UNIVERSITY

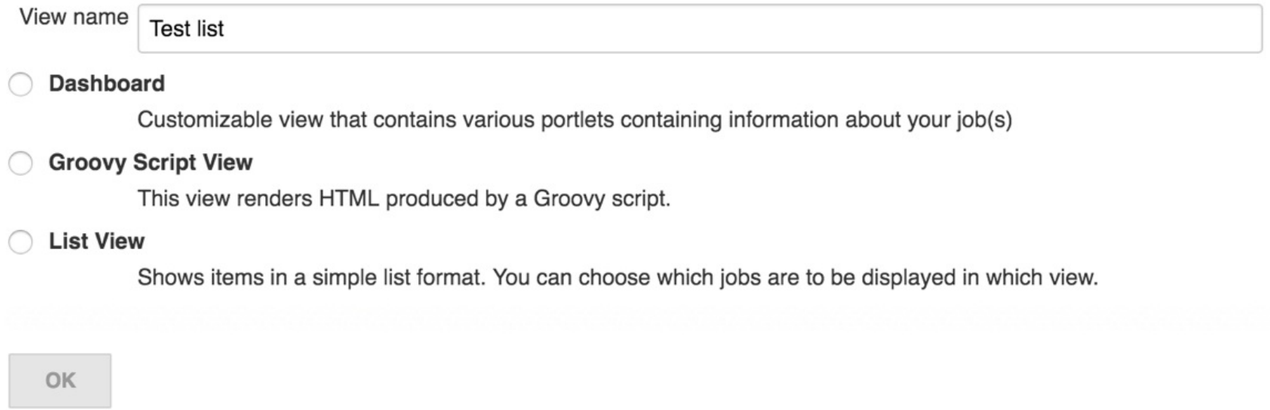
USING VIEWS



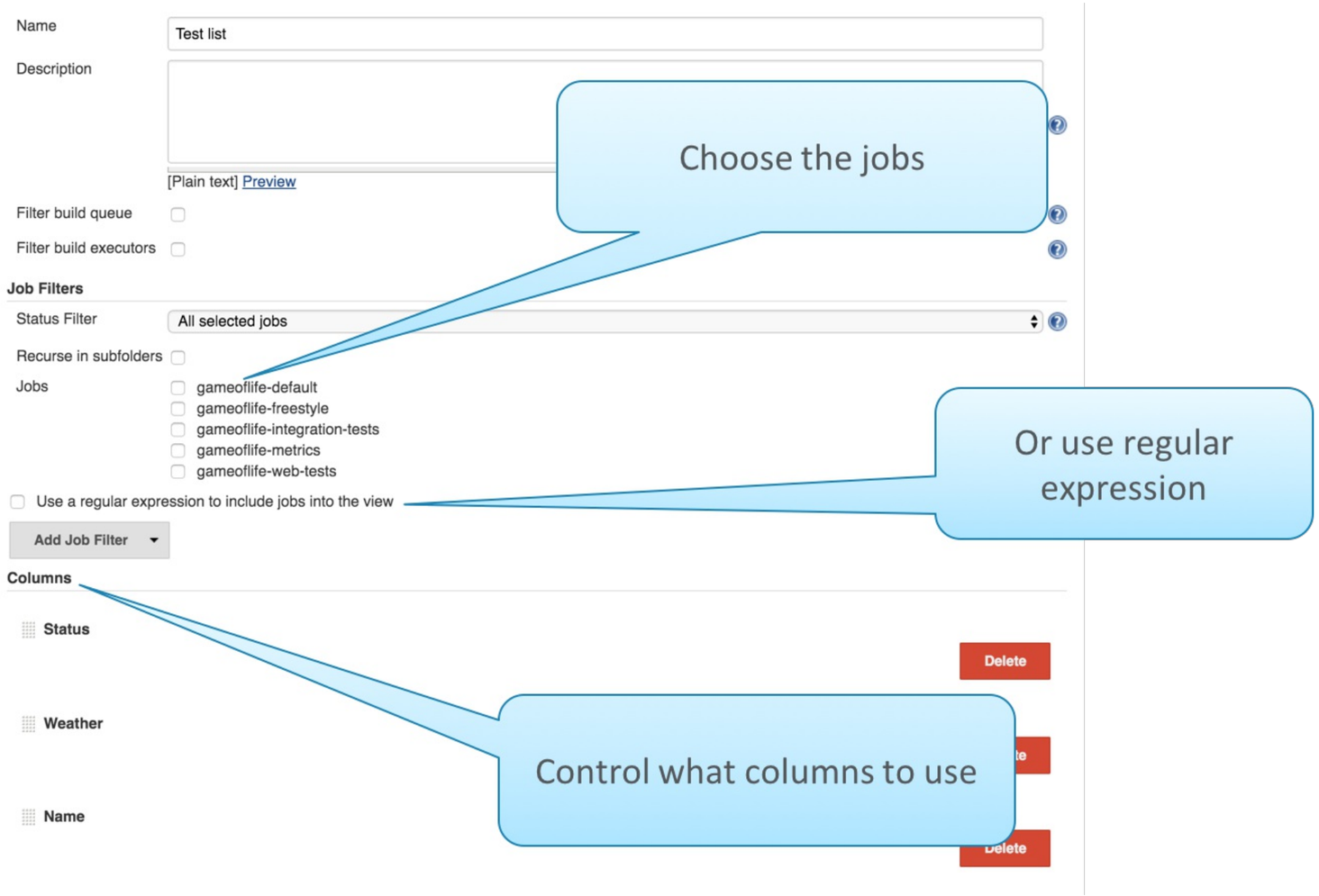
CLOUDBEES UNIVERSITY

LIST VIEW

- Contains a list of projects:



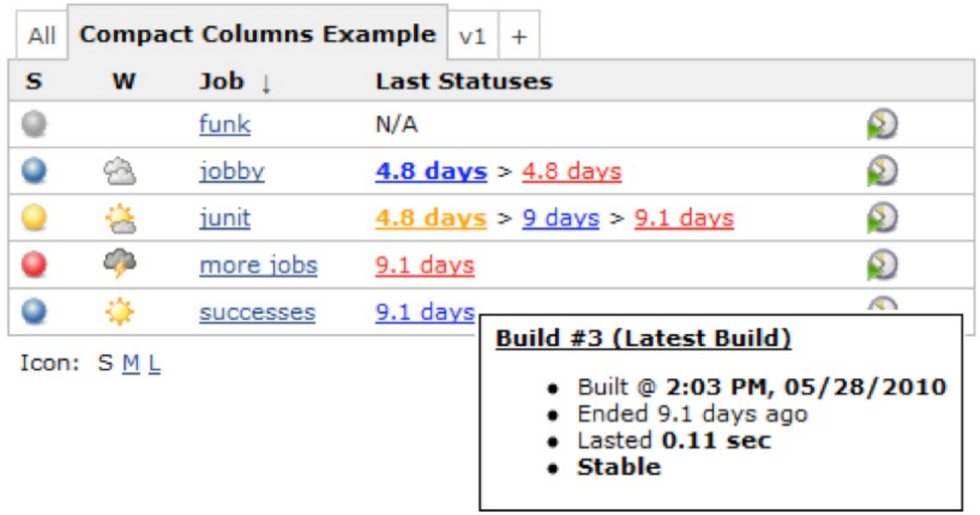
CREATING A LIST VIEW



CLOUDBEES UNIVERSITY

CUSTOMIZE LIST VIEW

- "View Job Filters" plugin
- Columns:
  - "Favorite" mark
  - Compact status column



CLOUDBEES UNIVERSITY

OTHER VIEW TYPES

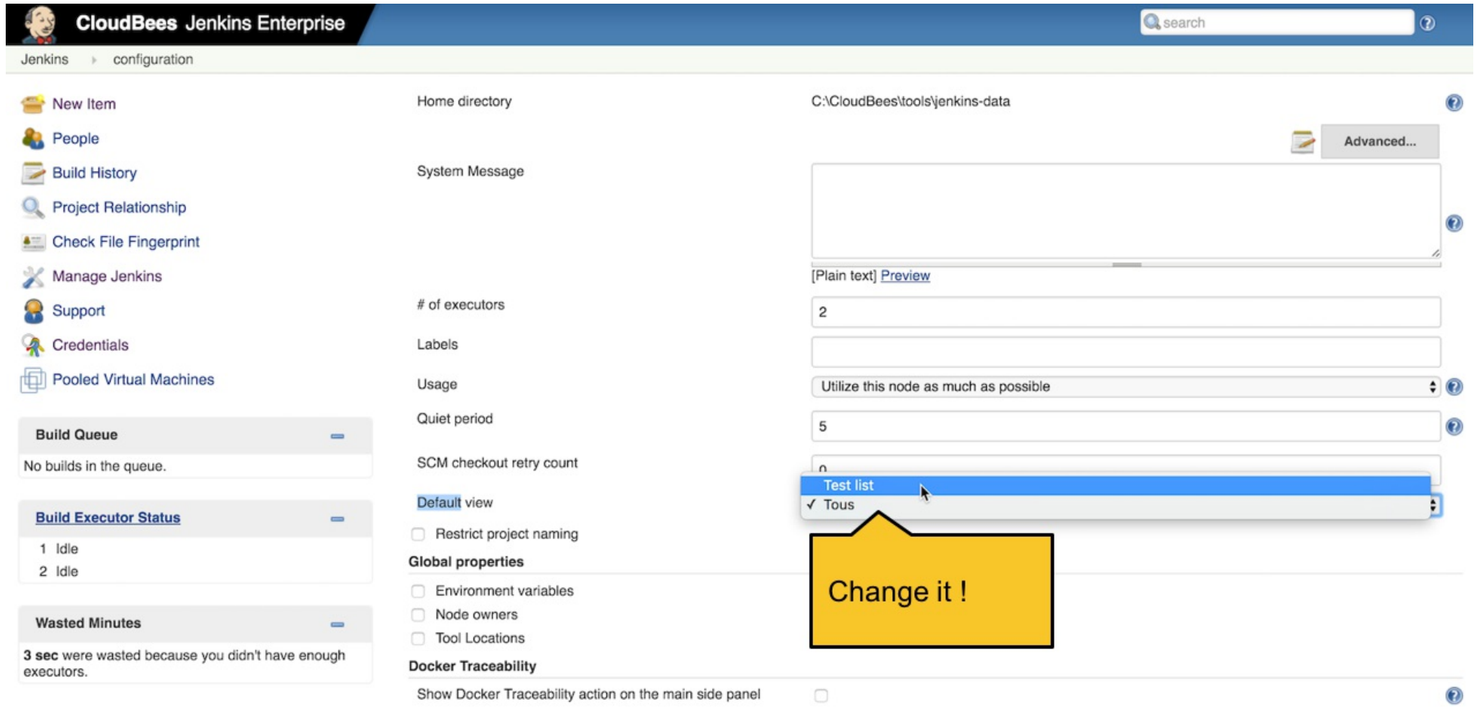
Radiator View Plugin:

- Highly-visible display of build status
  - Visible from a distance
  - May show all builds
  - Or only broken ones

CLOUDBEES UNIVERSITY

DEFAULT VIEW

- All view is the default, but if more views exist, this can be changed in $JENKINS_URL/configure:



CLOUDBEES UNIVERSITY

LAB EXERCISES:

- Lab 3: Displaying Test Results
- Lab 4: Creating an Integration Tests Build

CLOUDBEES UNIVERSITY

PLUGIN MANAGEMENT

CLOUDBEES UNIVERSITY

JENKINS PLUGINS

- Jenkins can be extended by writing plugins
- They are mainly used to integrate with 3rd-party tools
- Many already available
- Check out the Jenkins website

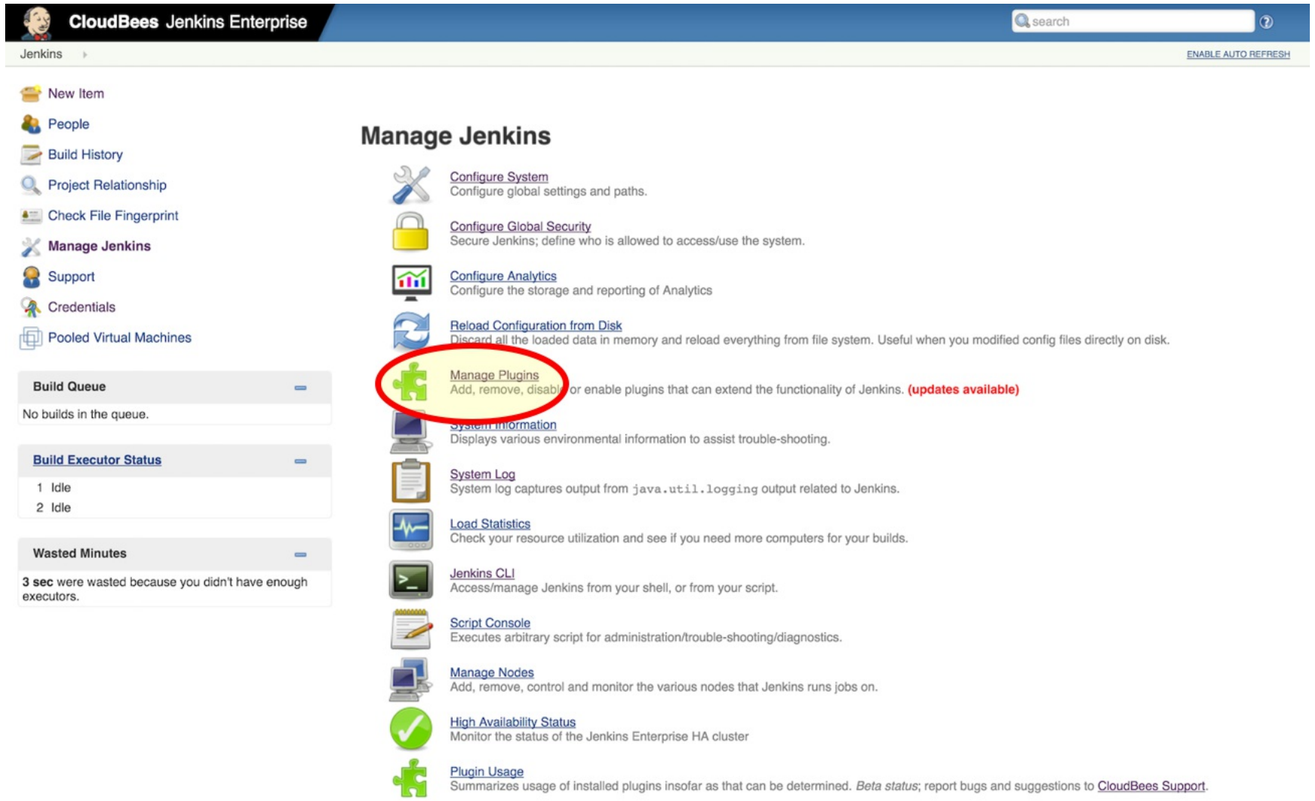
CLOUDBEES UNIVERSITY

PLUGIN EXAMPLES

- Source code management tools
- Build Tools
- Reporting tools
  - Code coverage, static code analysis
- Online source code browsers
- Issue tracker
- Notification tools
- Views and UI customizations
- Distributed builds

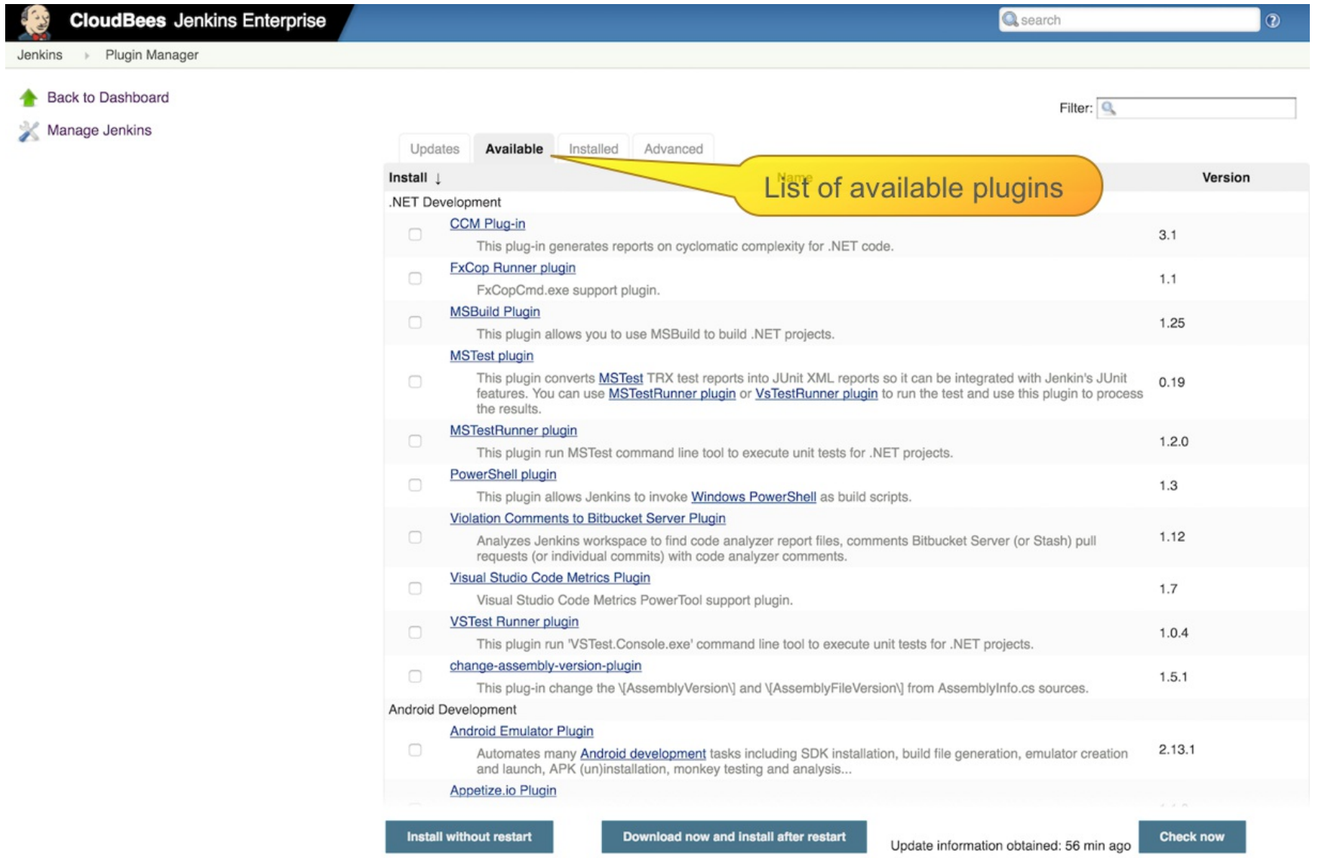
CLOUDBEES UNIVERSITY

INSTALLING PLUGINS



CLOUDBEES UNIVERSITY

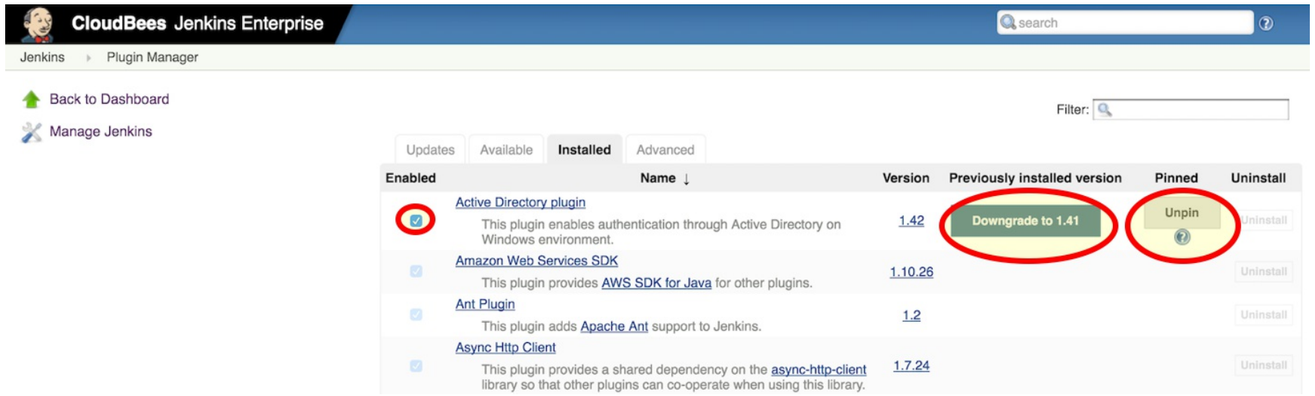
WHAT PLUGINS ARE AVAILABLE?



CLOUDBEES UNIVERSITY

INSTALLED PLUGINS

- You can downgrade, pin/unpin or disable plugins:



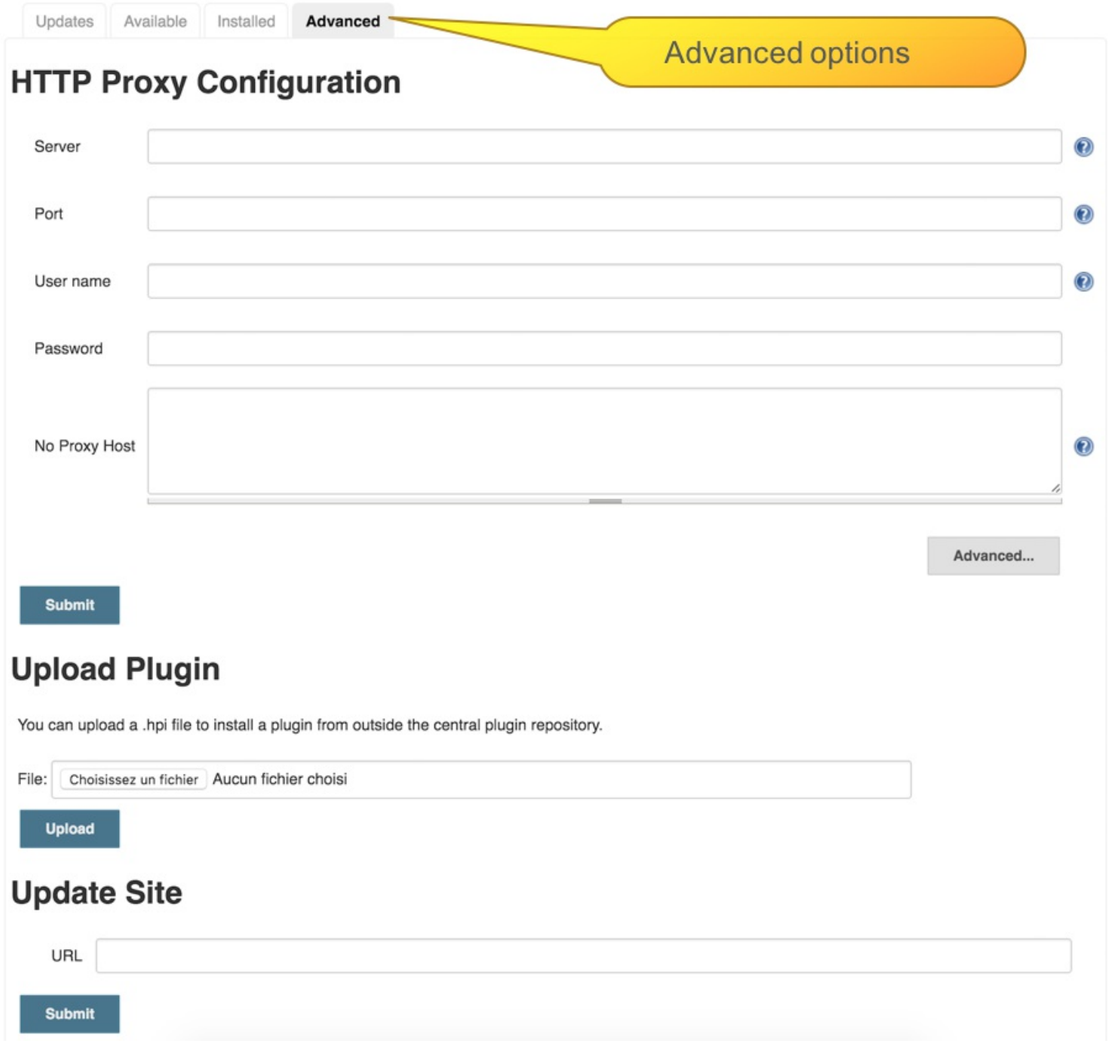
CLOUDBEES UNIVERSITY

PINNING PLUGINS:

- When you update bundled plugin, they get pinned
- Pinned plugins don't get overwritten automatically
- On disk, you'll see a file named foo.hpi.pinned

CLOUDBEES UNIVERSITY

MANUALLY INSTALLING PLUGINS



CLOUDBEES UNIVERSITY

EXAMPLE: VIOLATIONS PLUGIN

- Reports on code quality metrics:
- Checkstyle, CPD, FindBugs, PMD, and also FxCop and pylint
- Your build script needs to generate the raw XML data

CLOUDBEES UNIVERSITY

WORKING WITH VERSION CONTROL

CLOUDBEES UNIVERSITY

WORKING WITH VERSION CONTROL

- Integrating with your SCM
  - Build triggers
  - Integrating with a SCM browser
  - Integrate with SCM users

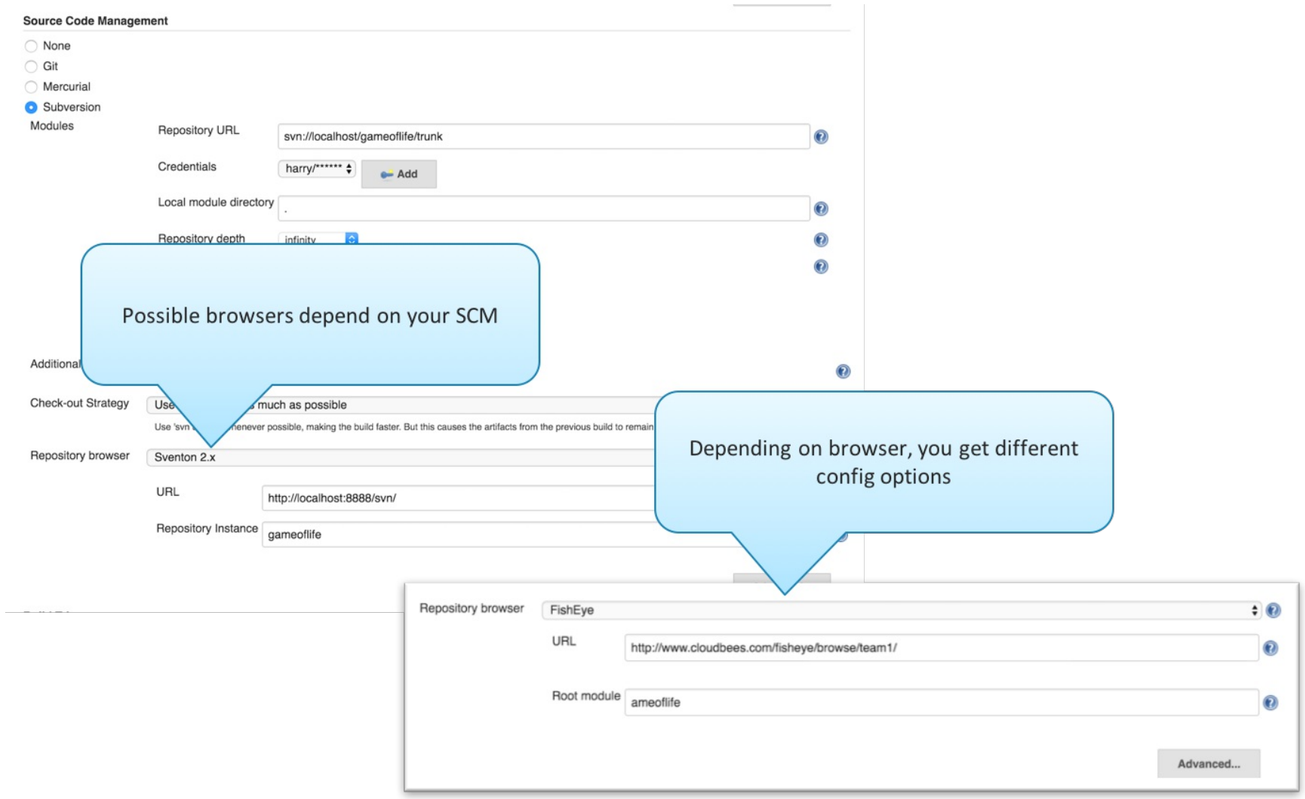
CLOUDBEES UNIVERSITY

SCM BROWSER INTEGRATION

- Why?
  - See details of what changes went to builds
  - You can share the pointer with colleagues
- Jenkins Integrates with browsers for many modern SCMs
  - Subversion: Trac, ViewSVN, WebSVN, Sventon, Fisheye, Collabnet,…
  - Git: gitweb, github,…
  - Mercurial: bitbucket, googlecode, hgweb,…

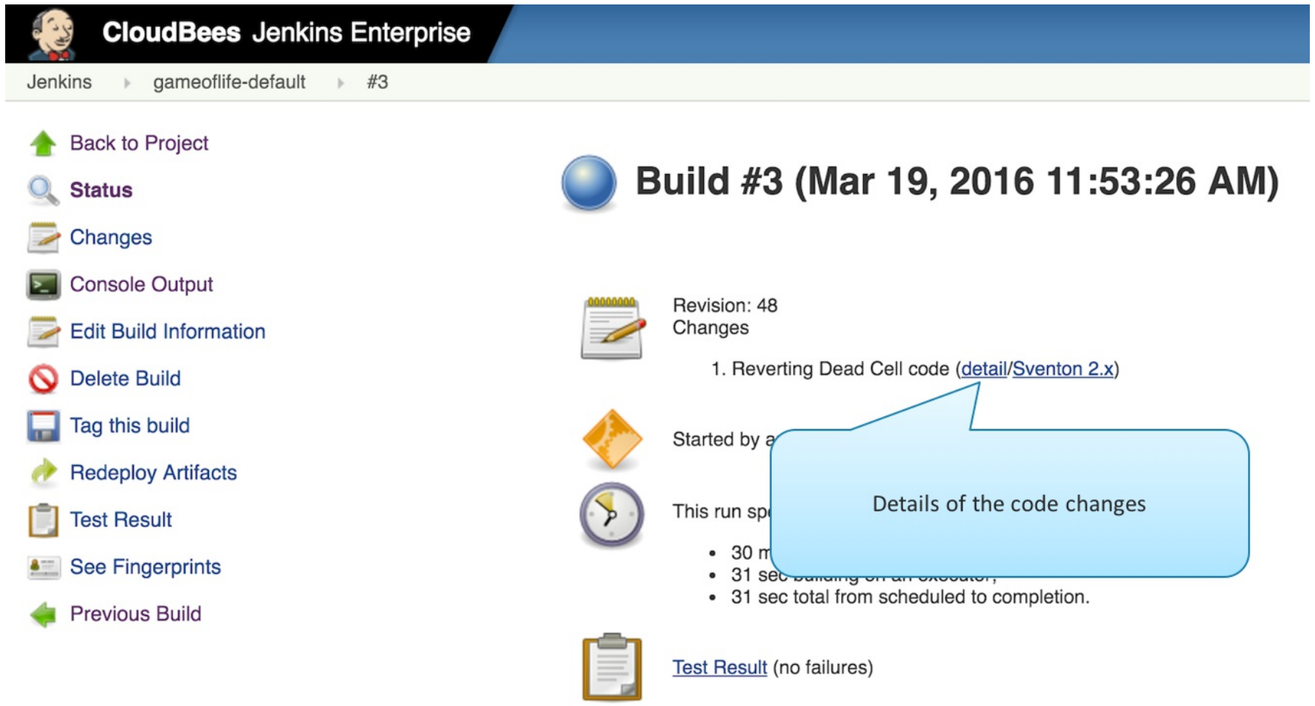
CLOUDBEES UNIVERSITY

EXAMPLE USING SVENTON



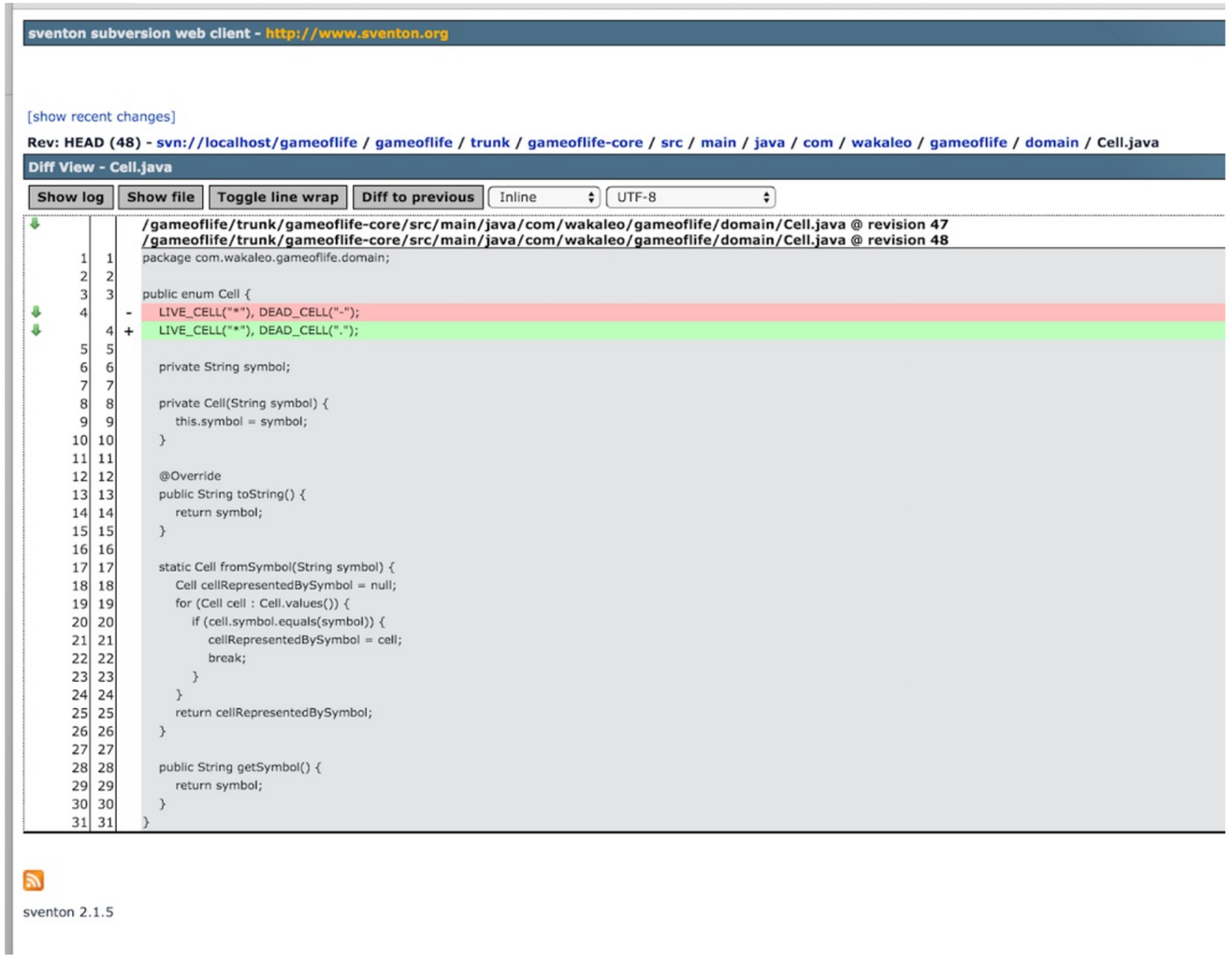
CLOUDBEES UNIVERSITY

EXAMPLE USING SVENTON



CLOUDBEES UNIVERSITY

EXAMPLE USING SVENTON



CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 5: Integrating with a SCM Browser

CLOUDBEES UNIVERSITY

CODE QUALITY METRICS

CLOUDBEES UNIVERSITY

CODE QUALITY METRICS

- Jenkins is very good at reporting on code quality metrics:
  - Static analysis (Checkstyle, FindBugs,…)
  - Code Coverage (Cobertura, Clover,…)
  - Code quality data mining(SonarQube)

CLOUDBEES UNIVERSITY

STATIC ANALYSIS

- Analyze code without running it
- Find potential bugs
- Examples:
  - Checkstyle
  - PMD
  - Findbugs
- Use the Jenkins 'Violations' plugin

CLOUDBEES UNIVERSITY

INTEGRATING CHECKSTYLE REPORTS

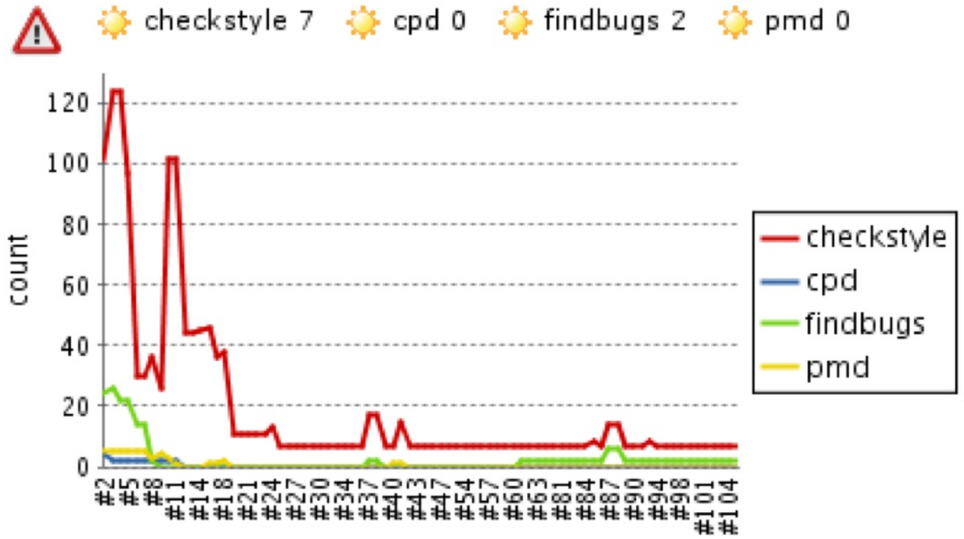
- Works on source code:
  - Enforce coding standards, javadoc, etc.
- Customizable rules
- Supports Ant, Maven, etc.

```
<project>
 <reporting>
    ...
    <plugin>
      <artifactId>maven-checkstyle-plugin</artifactId>
      <configuration>
        <configLocation>${basedir}/src/main/config/my-checkstyle-rules.xml</configLocation>
      </configuration>
    </plugin>
    ...
  </plugins>
 </reporting>
</project>
```

CLOUDBEES UNIVERSITY

INTEGRATING CHECKSTYLE REPORTS

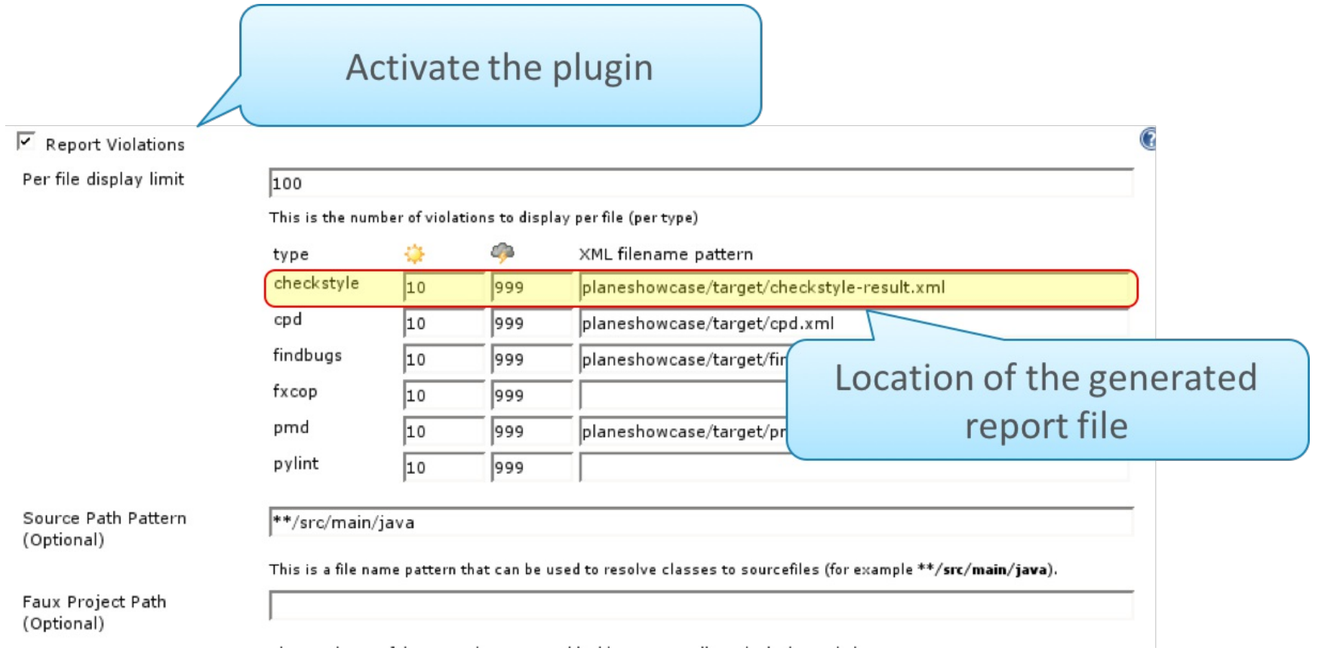
- Checkstyle statistics are included in the Violations graph



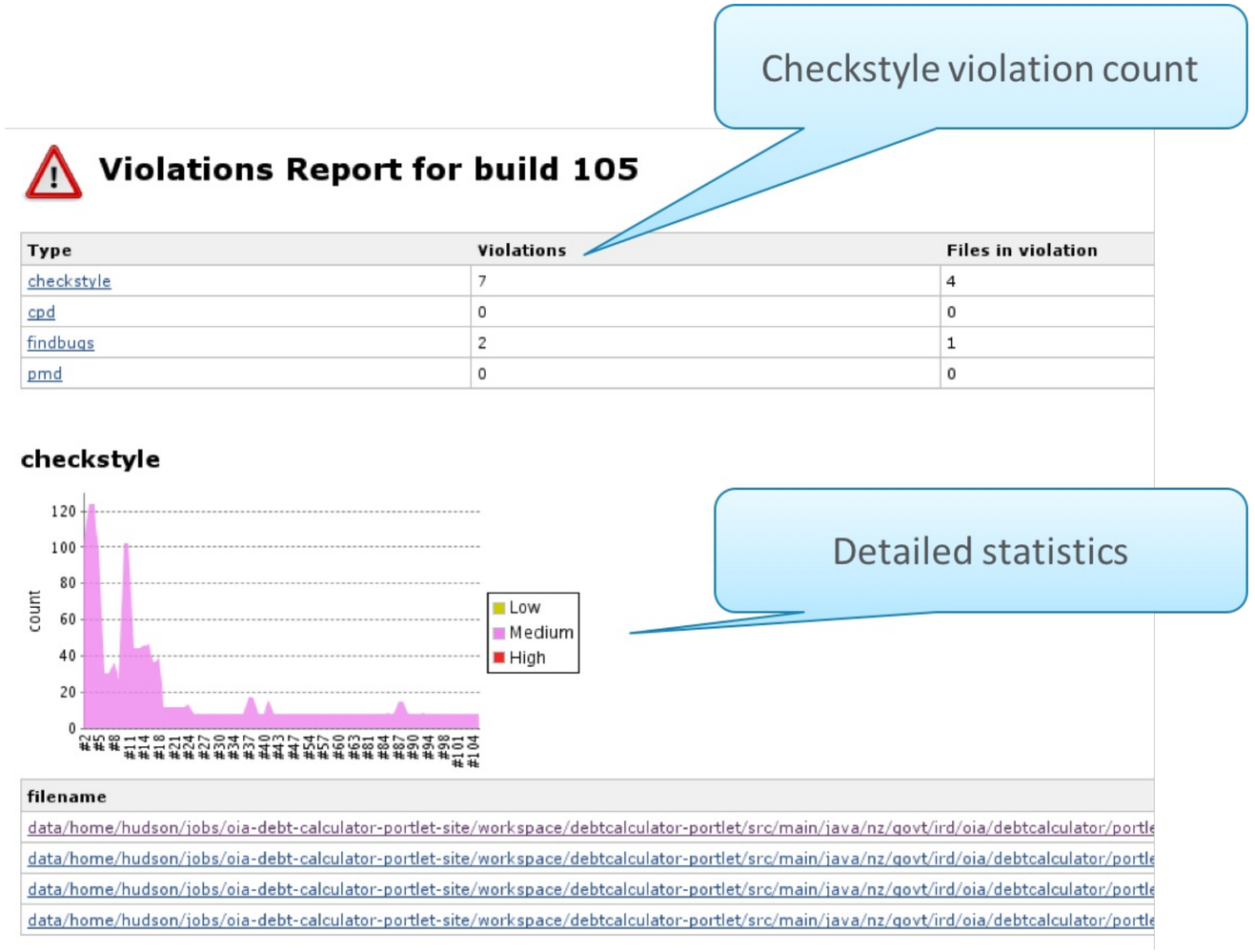
CLOUDBEES UNIVERSITY

CONFIGURING CHECKSTYLE PLUGIN

- Configured as a post-build plugin



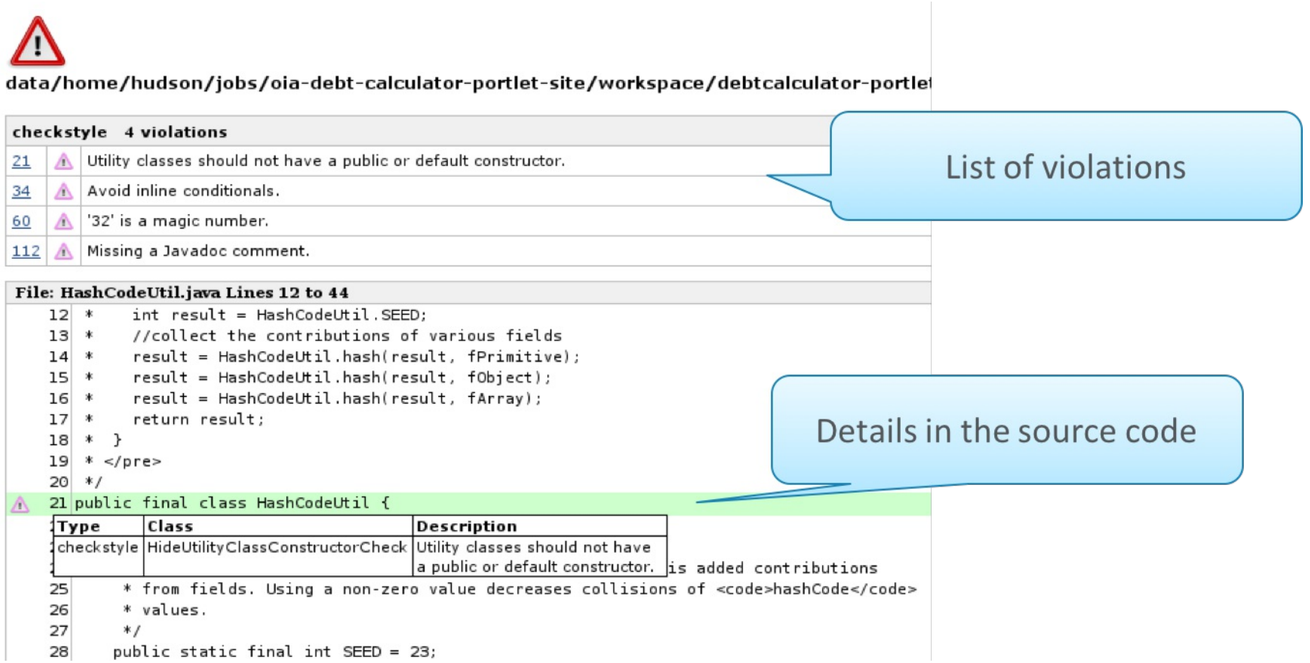
CHECKSTYLE DETAILS



CLOUDBEES UNIVERSITY

CHECKSTYLE SOURCE CODE OVERLAY

- View details for a particular class:



CLOUDBEES UNIVERSITY

VIOLATIONS PLUGIN SUPPORTS MORE

- PMD
  - Works on source code
    - Based on xpath matches from source tree.
  - Find coding problems
- CPD
  - Works on source code
  - Detects copy/paste in code
- FindBugs
  - Works on bytecode
  - Good at finding errors involving types

CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 6: Code Quality Metrics

CLOUDBEES UNIVERSITY

CODE COVERAGE METRICS:

Dynamic Analysis

CLOUDBEES UNIVERSITY

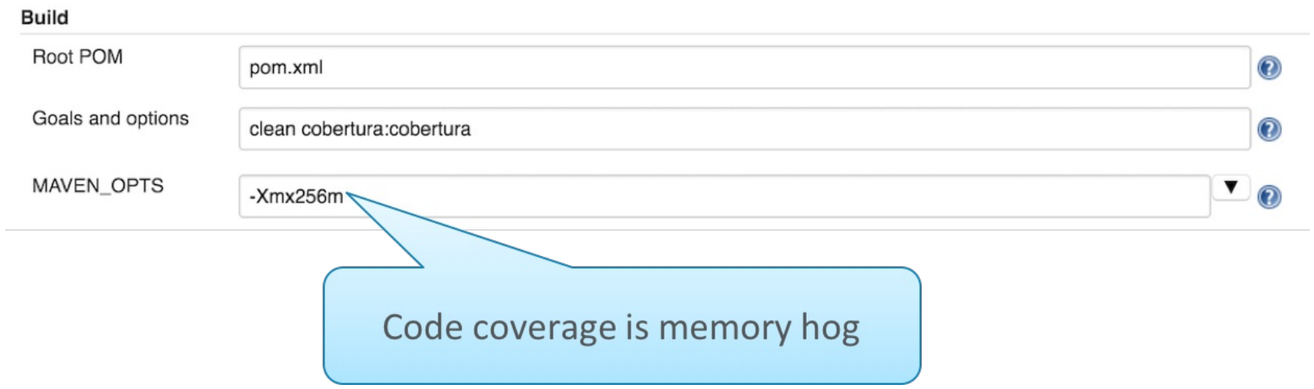
DYNAMIC ANALYSIS

- Most commonly used code coverage
  - But there are others, like deadlock analysis
- Code coverage measures how much code is executed by your tests
- Useful in guiding unit test effort
- Jenkins integrates with several Code Coverage tools:
  - JaCoCo, Clover, Cobertura, Emma, Crap4j

CLOUDBEES UNIVERSITY

INTEGRATING WITH COBERTURA

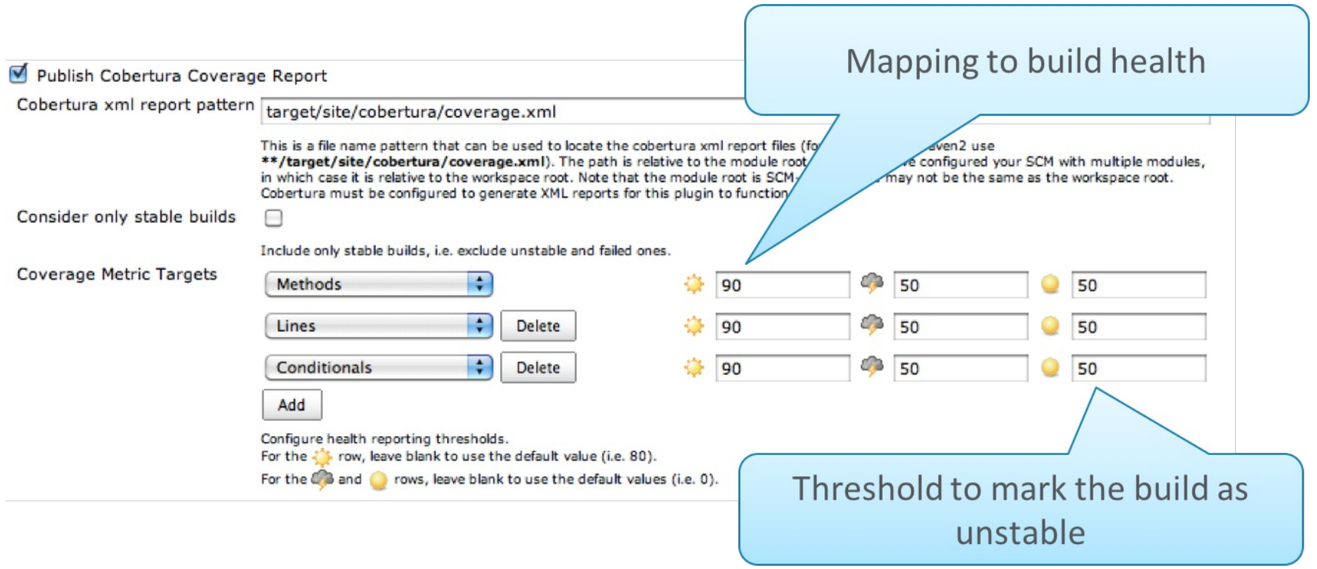
- Configuring projects to work with Cobertura:



CLOUDBEES UNIVERSITY

INTEGRATING WITH COBERTURA

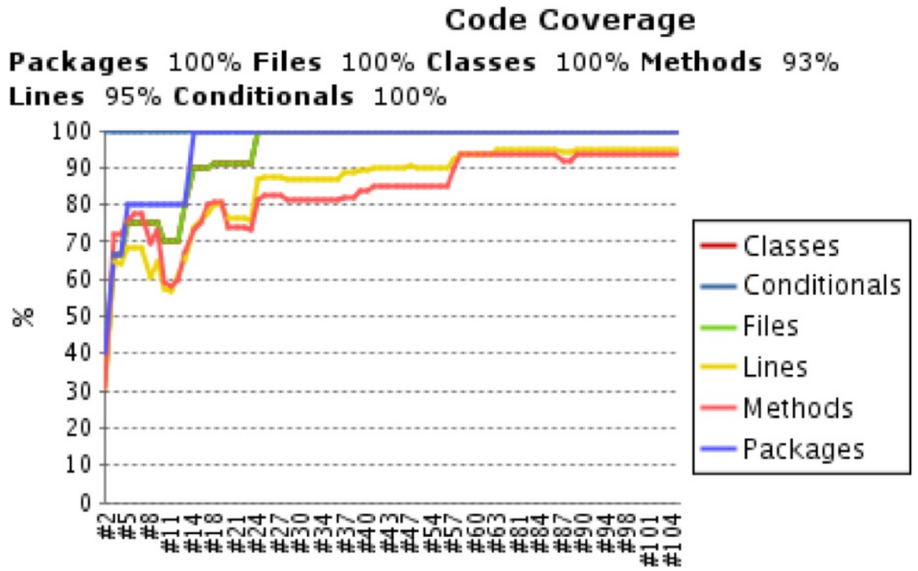
- Configure Cobertura in the Post-build Actions:



CLOUDBEES UNIVERSITY

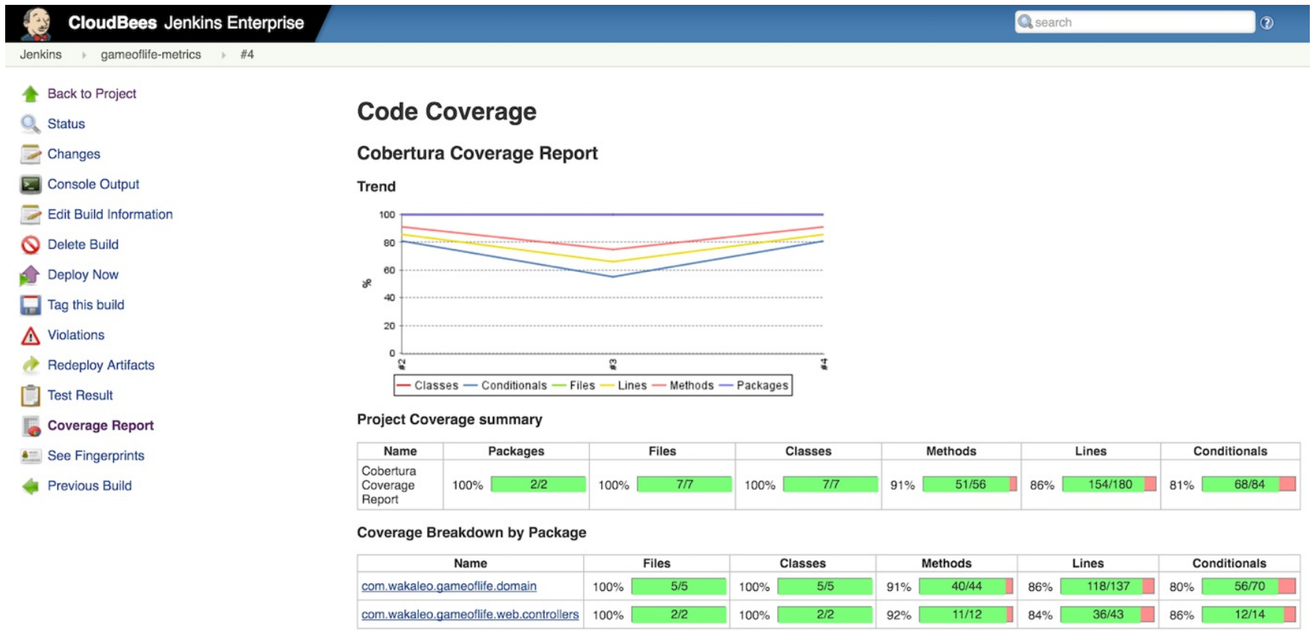
CODE COVERAGE REPORTS

- Code coverage statistics are displayed on the build results page



CLOUDBEES UNIVERSITY

CODE COVERAGE REPORT DRILL-DOWN



CLOUDBEES UNIVERSITY

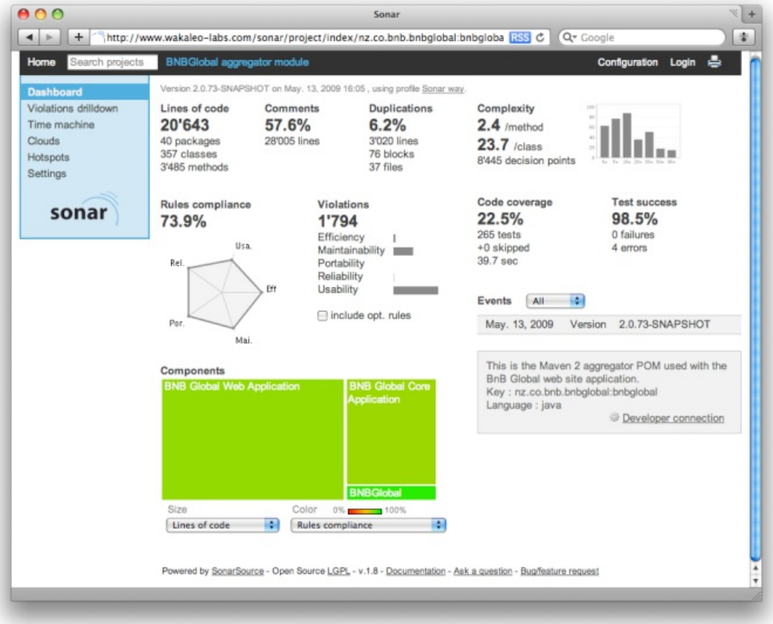
CODE COVERAGE LINE-BY-LINE REPORT

```
43      */
44     public Manifest getManifest() {
45  3      if (manifest == null) {
46  0          loadMetaInf();
47         }
48  3      return manifest;
49     }
50
51     public void setManifest(Manifest manifest) {
52  0      this.manifest = manifest;
53  0  }
54
55     public ServletContext getContext() {
56  0      return context;
57     }
58
59     public void setContext(ServletContext context) {
60  3      this.context = context;
61  3  }
62
63     /**
64      * Load the META-INF file.
65      */
66     public void loadMetaInf() {
67  3      String appServerHome = context.getRealPath("/");
68  3      File manifestFile = new File(appServerHome, "META-INF/MANIFEST.MF");
```

CLOUDBEES UNIVERSITY

COMBINED METRICS: SONARQUBE

- SonarQube gives overview of project metrics
- Projects publish metrics data to the SonarQube database
- Jenkins is obviousplace to run them



CLOUDBEES UNIVERSITY

PUBLISHING METRICS TO SONARQUBE

Step 1

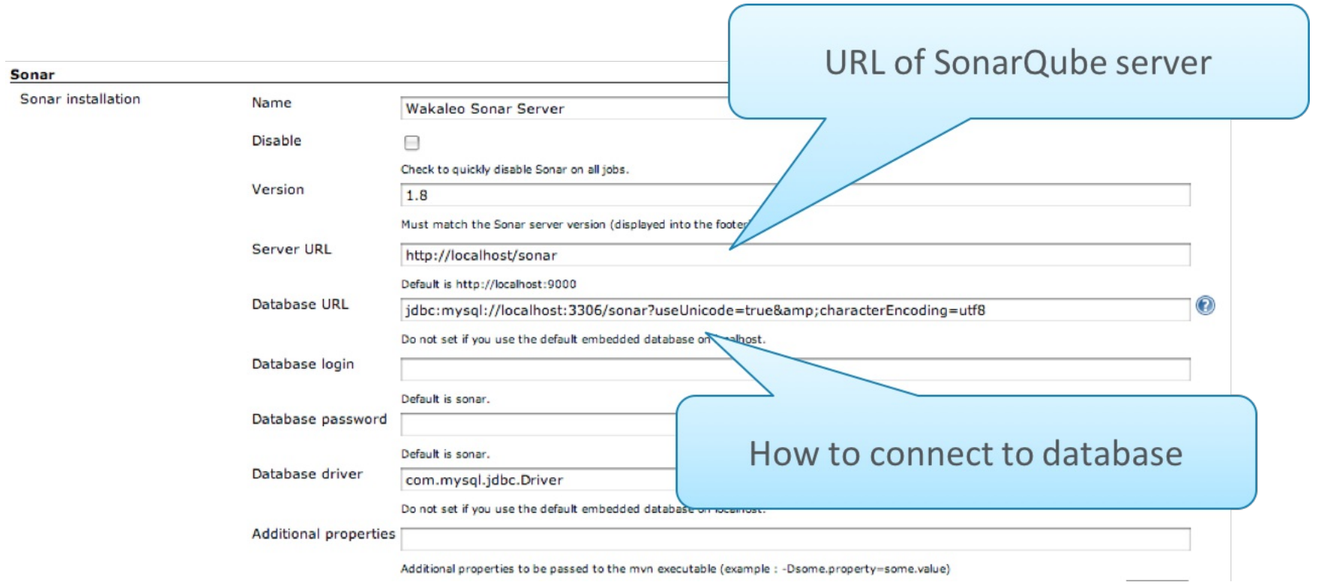
Configure the Jenkins SonarQube plugin:

Step 2

Configure SonarQube:

CLOUDBEES UNIVERSITY

CONFIGURING SONARQUBE PLUGIN



CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 7: Code Coverage metrics

CLOUDBEES UNIVERSITY

PARAMETERIZED BUILDS

CLOUDBEES UNIVERSITY

WHAT ARE PARAMETERIZED BUILDS?

- For the moment, jobs were no-arguments methods call

```
void doCIOfProjectGameOfLife() {...}
```

- but it is possible to add parameter to this method

```
void doCIOfProjectGameOfLife(testPlatform) {...}
```

CLOUDBEES UNIVERSITY

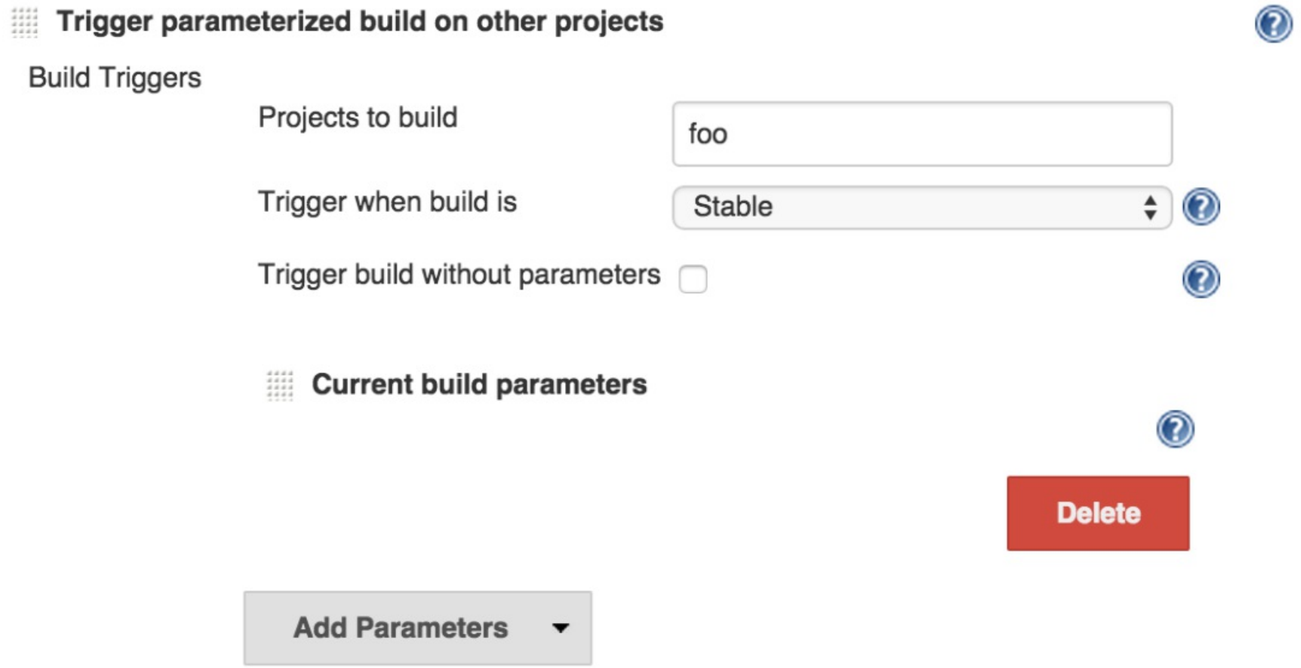
BENEFITS: GENERIC JOBS

- You don’t need to create a job for every platform you want to test / use
  - DRY : Do not Repeat Yourself
  - Maintenance is easier

CLOUDBEES UNIVERSITY

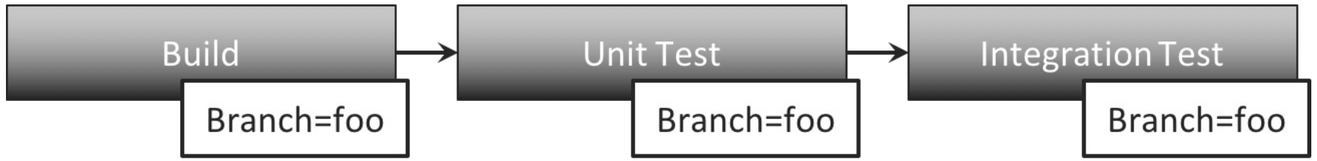
BENEFITS: CREATE A « PIPELINE »

- Using Parameterized Trigger Plugin, you can start another job with the current parameters:



CLOUDBEES UNIVERSITY

BENEFITS: CREATE A "PIPELINE"

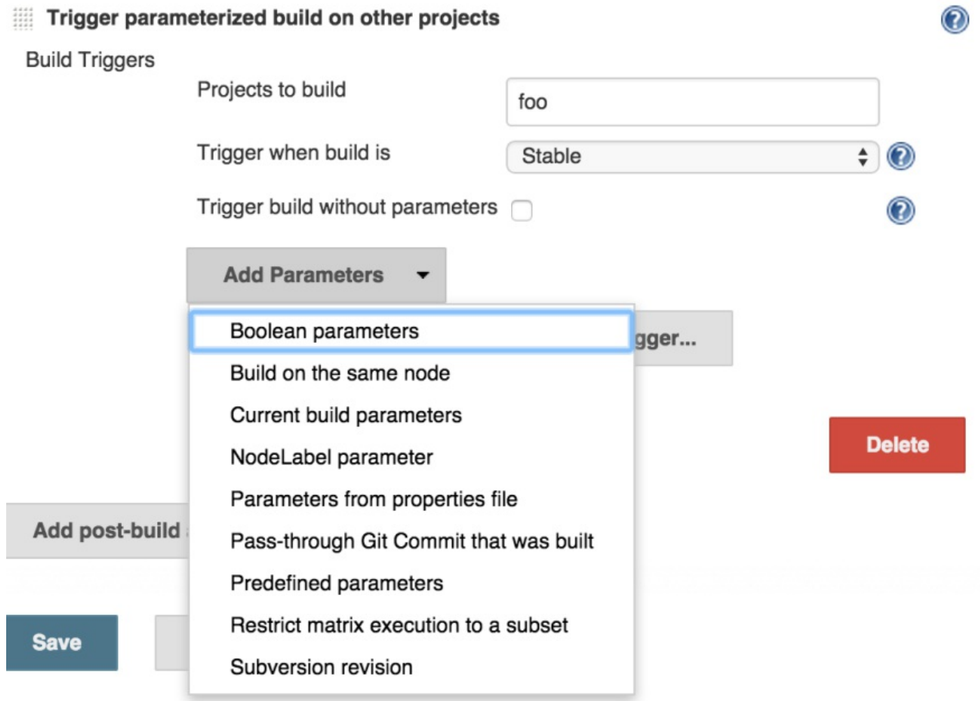


- Pass current parameters:
- Pass SCM revision/branch/tags
  - This makes downstream builds check out the exact same revision
  - E.g., Ensure unit tests and builds are synced
- Similar feature available for all SCMs

CLOUDBEES UNIVERSITY

BENEFITS: CREATE A « PIPELINE »

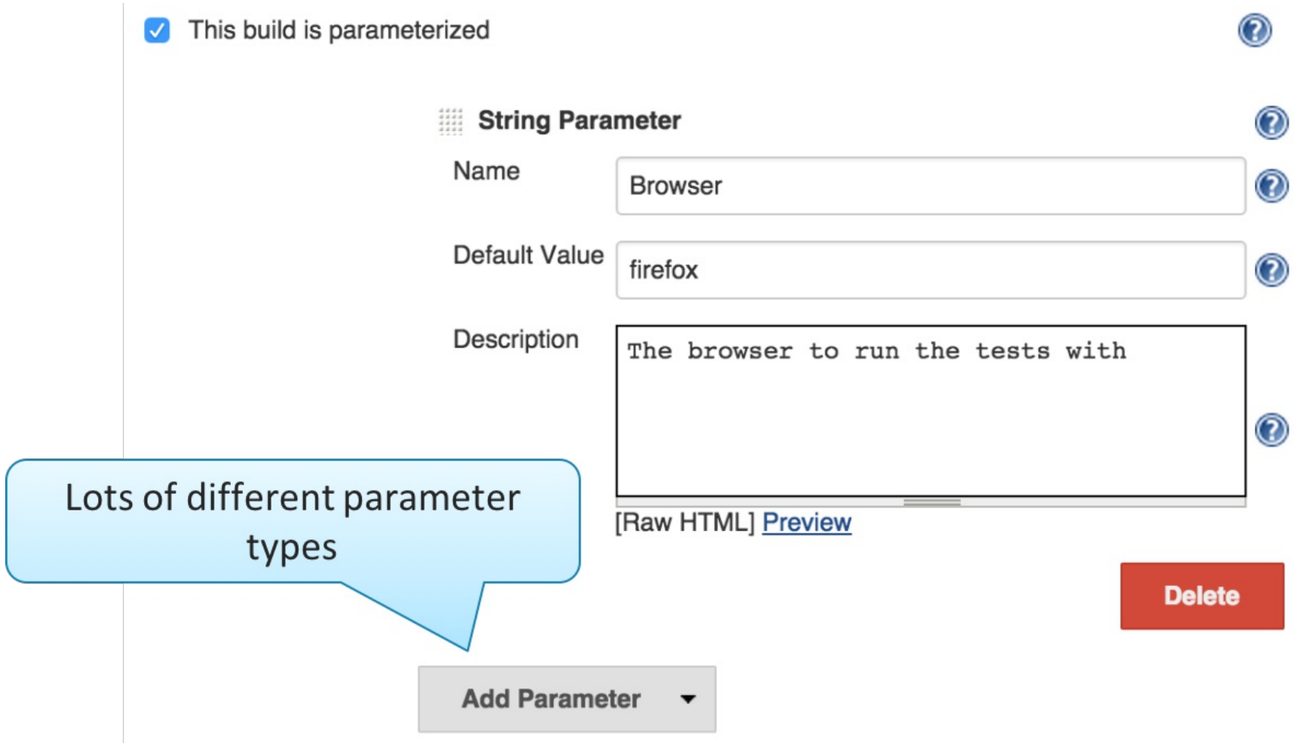
- It's also possible to provide more or different parameters to the next job:



CLOUDBEES UNIVERSITY

WHERE DO WE START? #1

- Define parameters



CLOUDBEES UNIVERSITY

WHERE DO WE START? #2

- Refer to parameter values
  - As variable expansions: ${Browser}
  - As environment variables from your builds
    - Watch out for cases!
  - Some parameter types expose data in different ways
    - File parameter

CLOUDBEES UNIVERSITY

WHERE DO WE START? #3

- Specify actual values when you launch a build:

- From curl/wget, use query parameter:

```
/job/foo/buildWithParameters?Browser=firefox
```

- From CLI:

```
-p Browser=firefox”
```

CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 8: Parameterized Builds

CLOUDBEES UNIVERSITY

AUTOMATED DEPLOYMENT

CLOUDBEES UNIVERSITY

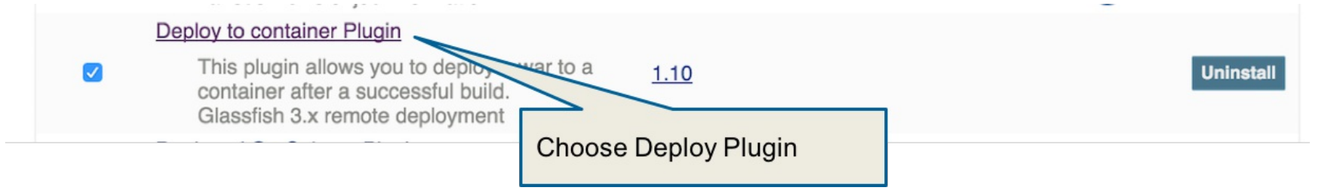
WHAT IS AUTOMATED DEPLOYMENT

- A way to deploy your application to a container:
  - Continuous Deployment
  - Manual Deployment to test servers

CLOUDBEES UNIVERSITY

WHERE DO WE START?

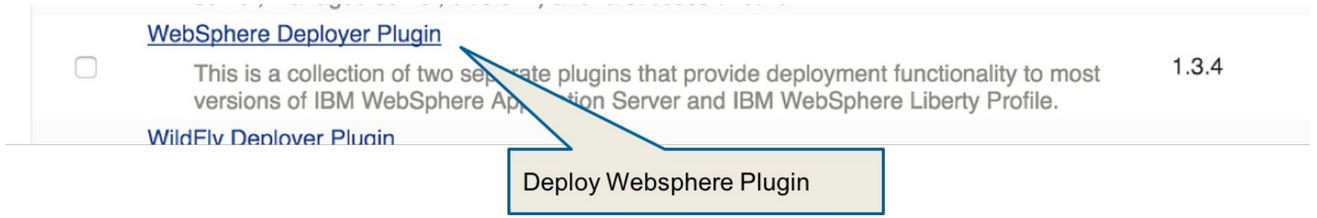
- Install Deploy Plugin
  - supports several application servers (Tomcat, JBoss, GlassFish)



CLOUDBEES UNIVERSITY

WHERE DO WE START?

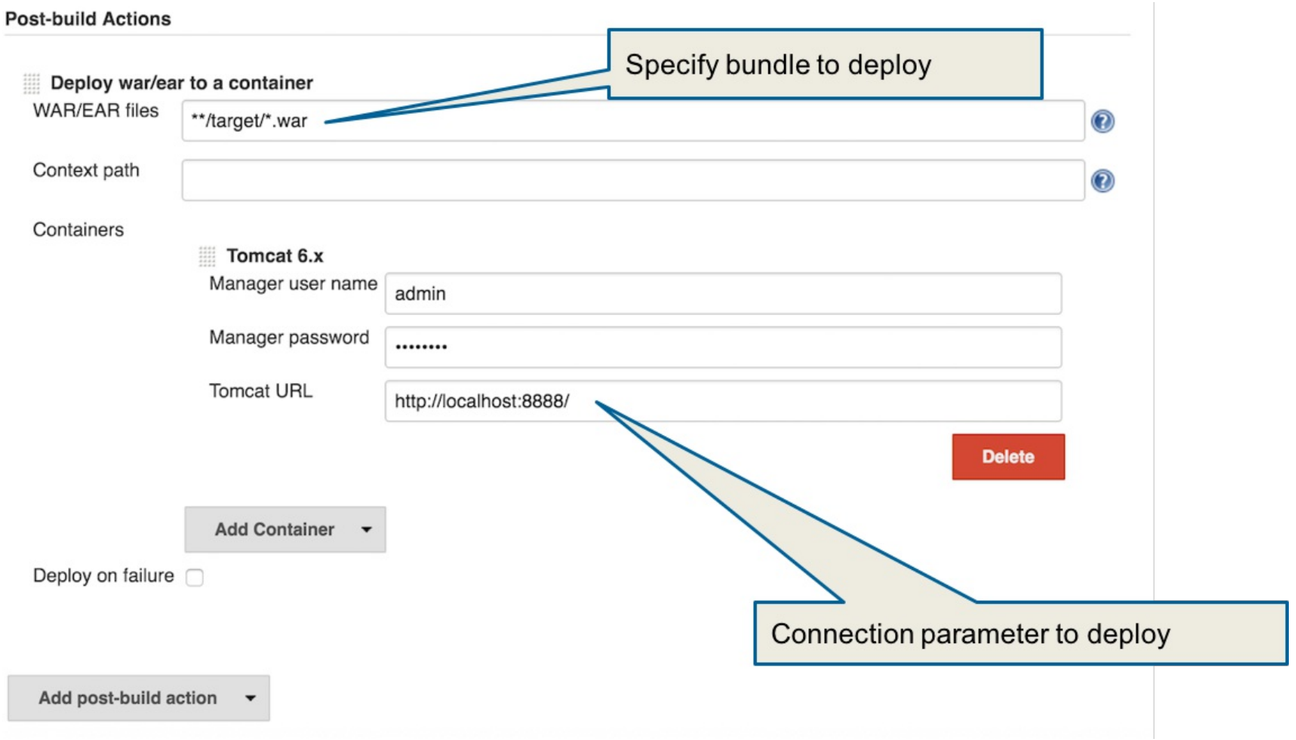
- Install Deploy Plugin Additions:
  - Some additional containers available as separate plugins



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Configure the WAR or EAR to be deployed



CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 9: Automated Deployment to Tomcat

CLOUDBEES UNIVERSITY

FOLDERS AND FOLDERS PLUS PLUGINS

CLOUDBEES UNIVERSITY

WHAT IS FOLDERS/FOLDERS PLUS?

- Just like operating system folders, unlimited nesting
  - Folder 1→Folder 2→Folder 3→…→Folder n
  - Help you model custom taxonomies
- Creates namespace
  - "build" job in Folder A is different than "build" job in Folder B
- Folders have their own views.

CLOUDBEES UNIVERSITY

BENEFITS: ISOLATION AND INHERITANCE

- When combined with the Role-Based Access Control (RBAC) plugin, Folders can have their own groups and permissions, effectively allowing teams to have their own “world” without needing their own Jenkins master
- Nested folders can inherit permissions from their parent folder, as well as templates in the parent folder.

CLOUDBEES UNIVERSITY

BENEFIT: GRANULAR CONTROL #1

- Each folder supports groups and roles configured on a particular folder.
- Permissions from root be inherited (default), but can also be excluded by filtering out all filterable root-level groups.
- Folders can limit what types of objects can be created in the folder.

CLOUDBEES UNIVERSITY

BENEFIT: GRANULAR CONTROL #2

- Folders allow for arbitrary key-value pairs to be set as environment variables.
  - Can be referenced from builds within folder
  - E.g.: $BRANCH_NAME or $DEPLOY_URL
- "Folders Plus" supports linking agents to specific folders, such that agents will only run that folders' builds.

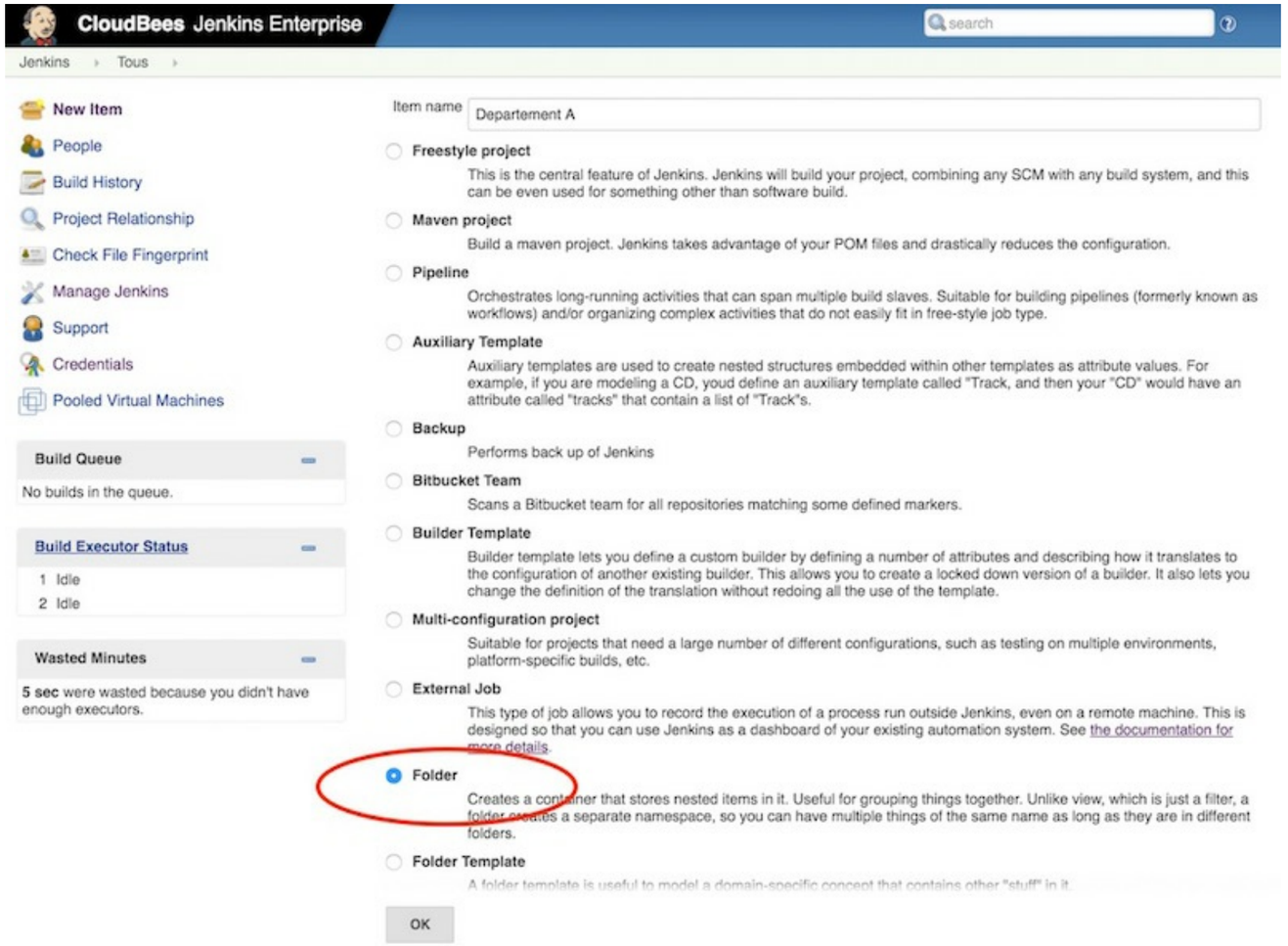
CLOUDBEES UNIVERSITY

BENEFITS: EASY BRANCHING

- When a branch of an existing project needs its own pipeline, the folder with the parent project's pipeline can simply be copied
  - Local references within the branch are kept intact
- Folders can also have environment variables for arbitrary key-value pairs, which jobs can reference
  - E.g.: $BRANCH_NAME or $DEPLOY_URL

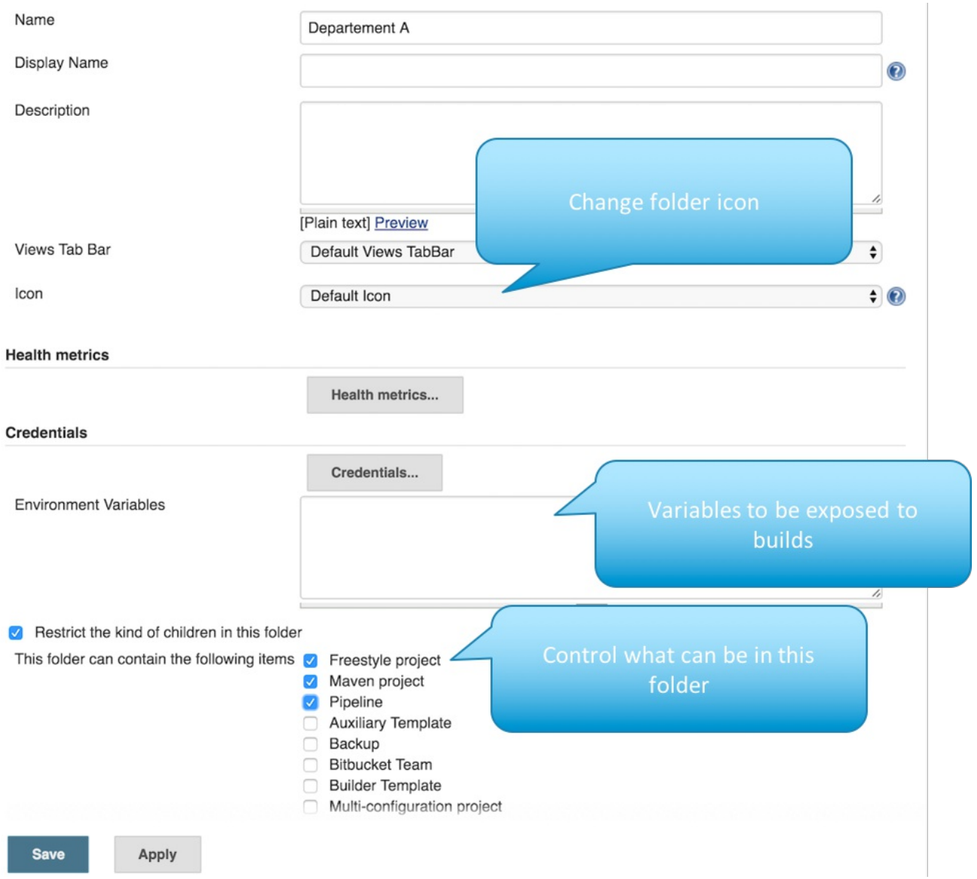
CLOUDBEES UNIVERSITY

WHERE DO WE START? #1



CLOUDBEES UNIVERSITY

WHERE DO WE START? #2



CLOUDBEES UNIVERSITY

SECURITY & ROLES BASE ACCESS CONTROL (RBAC)

CLOUDBEES UNIVERSITY

ENABLING SECURITY

Jenkins can integrate its security to a lot of existing assets:

- Active Directory
- Delegate to servlet container
- Google Apps SSO (with OpenID)
- Jenkins' own user database
- LDAP
- OpenID SSO
- Operations Center Single Sign On
- Unix user/group database

CLOUDBEES UNIVERSITY

PER-PROJECT ACCESS CONTROL

- If your Jenkins installation hosts sensitive projects that must be visible to a restricted set of people, define permissions at the individual project level so that different people have access to different sets of projects.
- Or if you just want to have a single set of access rights across all your projects, Jenkins can be configured that way, too.

CLOUDBEES UNIVERSITY

WHAT IS RBAC?

- Role-based access control can be configured as an authorization policy.
- RBAC comes as a CJE plugin.
- RBAC allows:
  - Creation of re-usable global and object-only roles.
  - Role define very granular permissions on all Jenkins objects (projects, view, folders, etc.)

CLOUDBEES UNIVERSITY

HOW DO I USE RBAC?

- Roles that are created must be mapped to groups defined in Jenkins:
  - These groups can be "shadow" groups for external groups in LDAP/AD
- Groups/roles can be defined on different objects in Jenkins:
  - Jobs
  - Maven modules
  - Agents
  - Views
  - Folders

CLOUDBEES UNIVERSITY

BENEFITS: GRANULARITY

- CJE RBAC supports very granular permissions for roles.
  - E.g. "Browse" role only has overall read permissions while "QA" role supports read/write/execute permissions on jobs only in a given folder.
- Roles and groups can be combined with folders and other Jenkins objects to limit users' permissions within those objects.
  - E.g. A team-only project folder containing all jobs in a pipeline, along with a job that is hidden from anyone not in the "QA Team" group.

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Ensure security is turned on and a valid security realm has been selected.
- Opt to use "role-based matrix authorization strategy" to enable CloudBees RBAC plugin to your authorization policy.
- Create groups and roles on jour CJE master configuration:
  - Assign roles to these groups
  - Add users to groups

CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 10: Job Organization and Security with Folders

CLOUDBEES UNIVERSITY

VALIDATED MERGE PLUGIN

aka the “Unbreakable Build”

CLOUDBEES UNIVERSITY

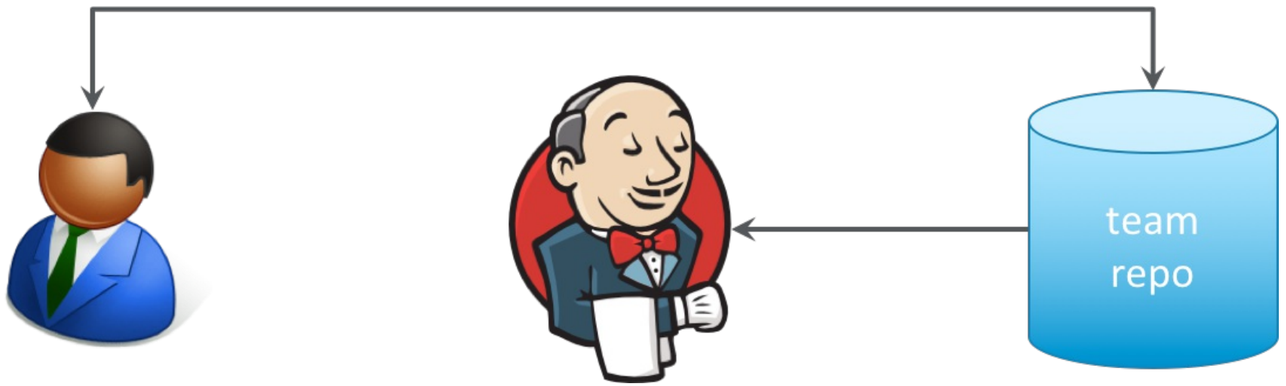
WHAT IS THE VALIDATED MERGE PLUGIN?

- How can you delegate more to Jenkins?
- Does your CI server shift work from laptops to servers?
  - Not if you have to share the changes to test: You end up testing locally before committing => FAIL
- OK, let's test it before sharing changes
  - By taking advantages of Git

CLOUDBEES UNIVERSITY

WHAT IS THE VALIDATED MERGE PLUGIN?

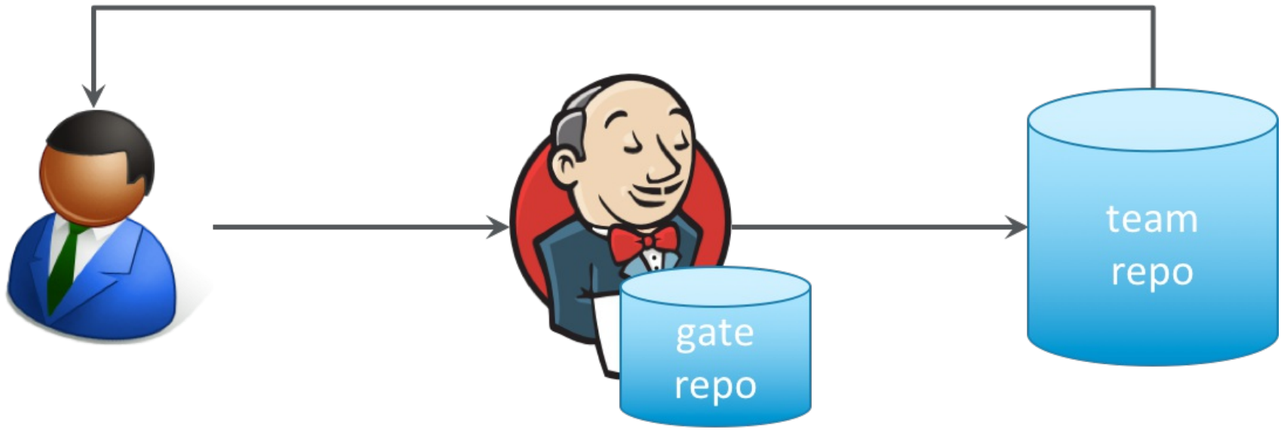
- Model before validated merge:
  - You push to team repo, then test



CLOUDBEES UNIVERSITY

WHAT IS THE VALIDATED MERGE PLUGIN?

- Model with Validated Merge:
  - Jenkins becomes git repo
  - You push to Jenkins
  - Jenkins tests, then push to team repo



CLOUDBEES UNIVERSITY

BENEFITS: NO BROKEN BUILDS

- A mistake will not block others
- Only push after changes are validated
- It is the Unbreakable Build pattern

CLOUDBEES UNIVERSITY

BENEFITS: TESTS ASYNCHRONOUSLY

Make

change and move on

Let

Jenkins follow through

CLOUDBEES UNIVERSITY

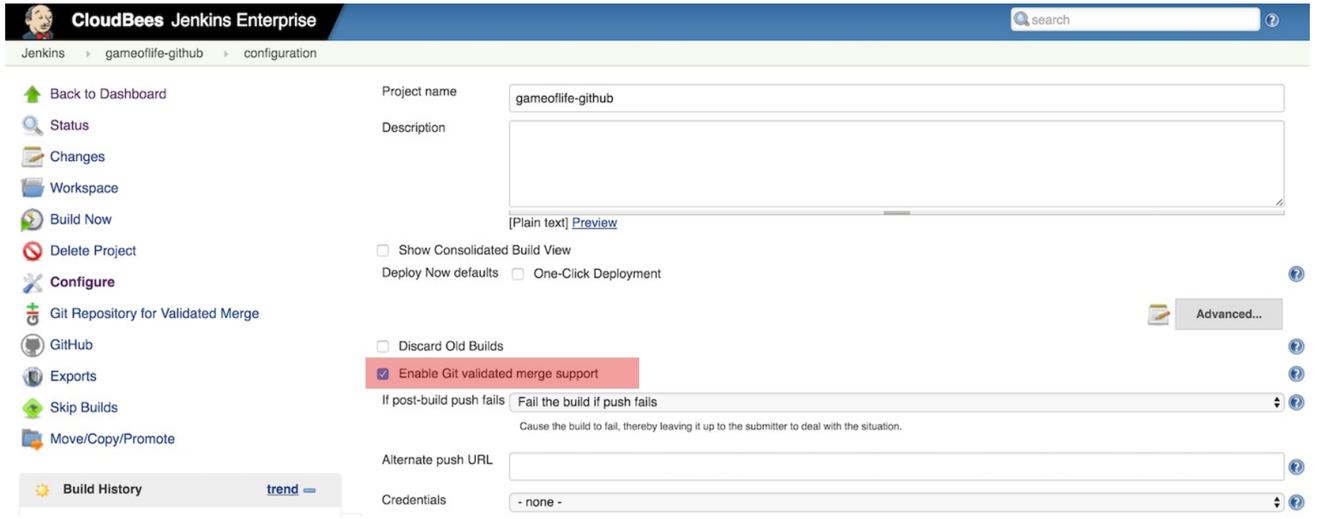
BENEFITS: TESTS ON THE SERVER

Your laptop has more important things to do Highly concurrent, environment-dependent test executions

CLOUDBEES UNIVERSITY

WHERE DO WE START?

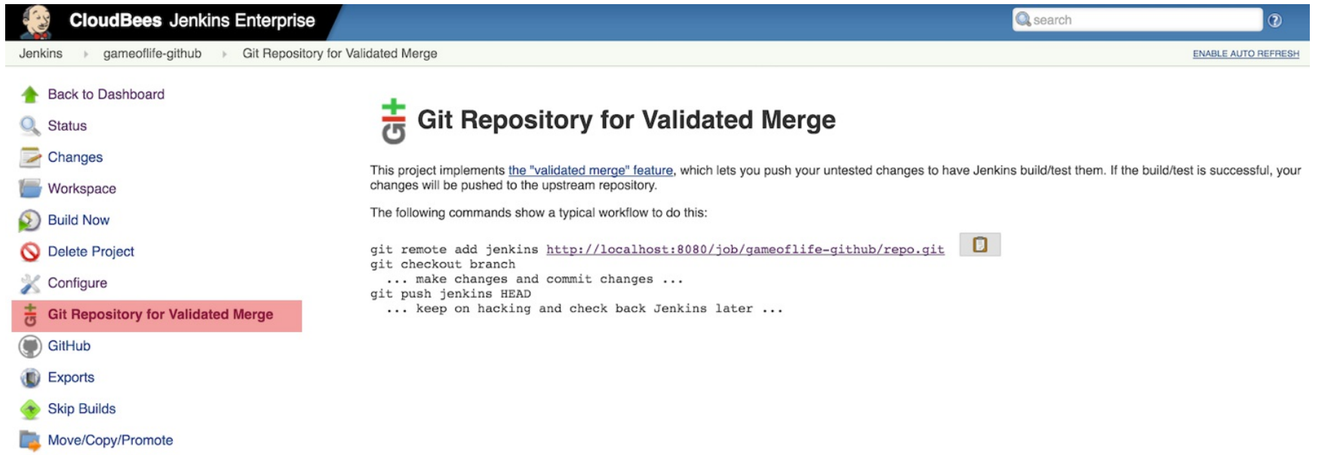
- To enable, just one checkbox:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Git access coordinate from project page:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

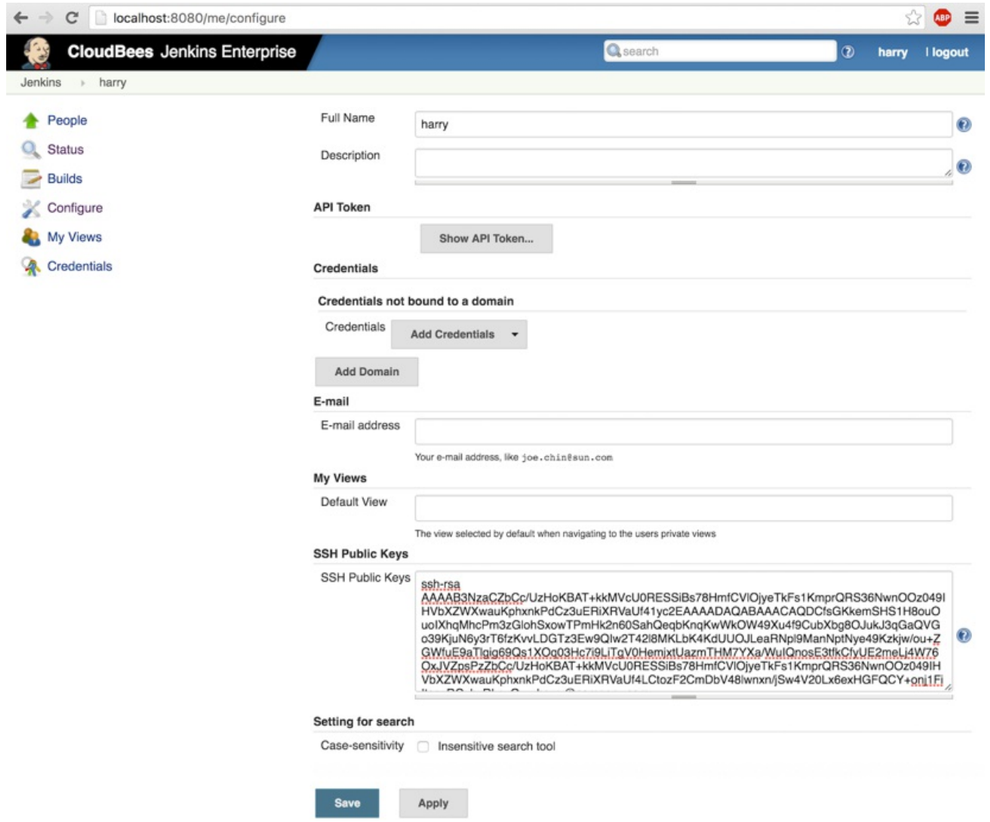
- Two protocols supported:
  - SSH
    - See system config to fix the port number for production use
  - HTTP
    - The same URL as the “Git repository” page in Jenkins Awkward to handle authentication

CLOUDBEES UNIVERSITY

WHERE DO WE START?

Developer Workflow

Register SSH public key : http://<JENKINS_URL>/me/configure



CLOUDBEES UNIVERSITY

WHERE DO WE START?

Developer Workflow

- Adds Jenkins as a new git remote repository

```
$ git remote add jenkins ssh://server:port/JOBNAME/
```

- Get changes from team repo

```
$ git pull origin mybranch
```

- Hack

```
$ vi .
$ git commit
```

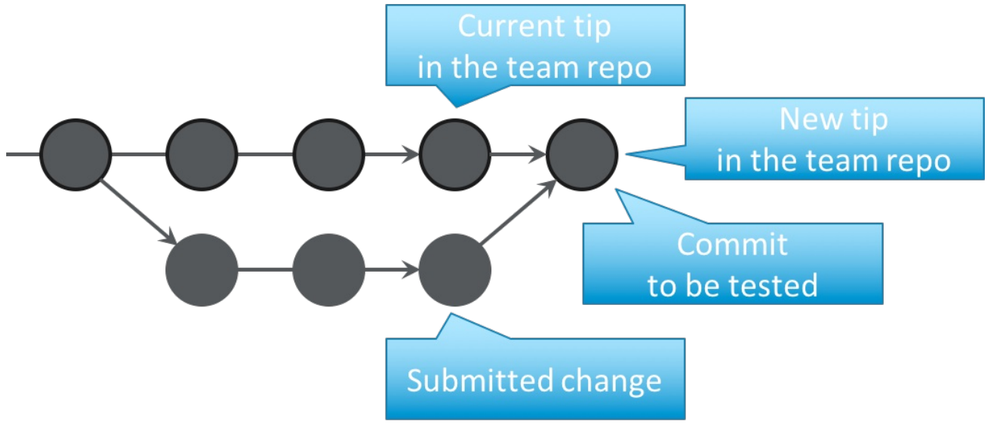
- Push to the « Jenkins » remote instead of team one

```
$ git push jenkins mybranch
```

CLOUDBEES UNIVERSITY

WHERE DO WE START?

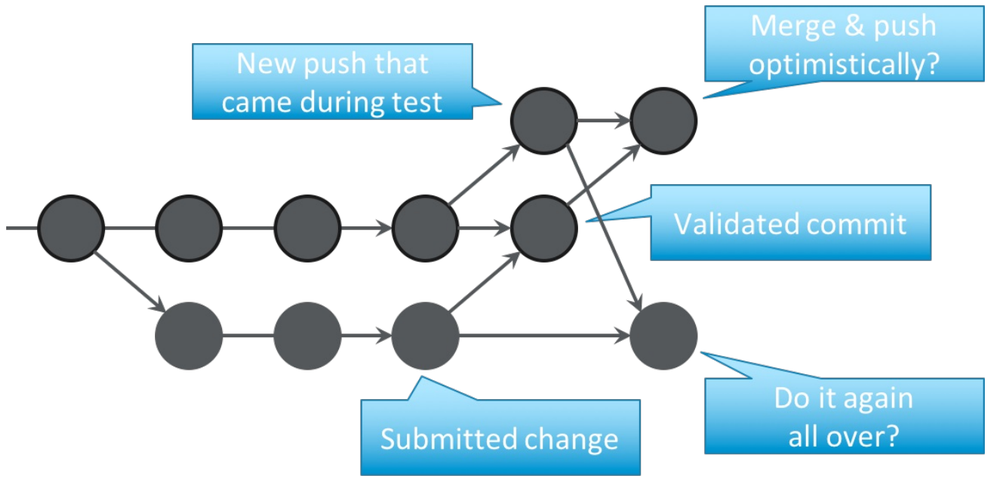
- Pre-build merge behavior
  - Jenkins needs to test what becomes the new tip:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Post-build merge behavior
  - What if new push is made to the team repo?



Or fail and leave it to the developer to resolve

CLOUDBEES UNIVERSITY

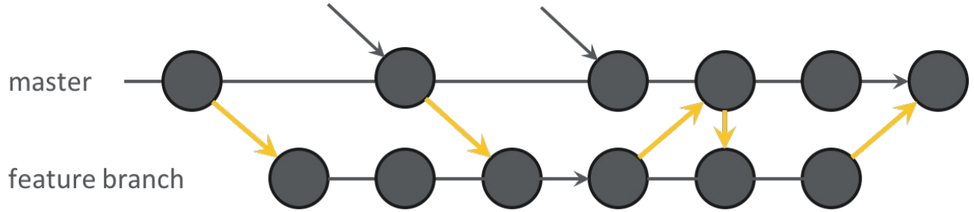
ALTERNATIVES: GERRIT

- Gerrit is a code review tool
- Gerrit trigger plugin for Jenkins:
  - Jenkins acts as a code reviewer (review===test)
- Works great if you want code review
- Problems:
  - Gerrit needs to own the repository
  - More client side changes required

CLOUDBEES UNIVERSITY

ALTERNATIVES: SUBVERSION?

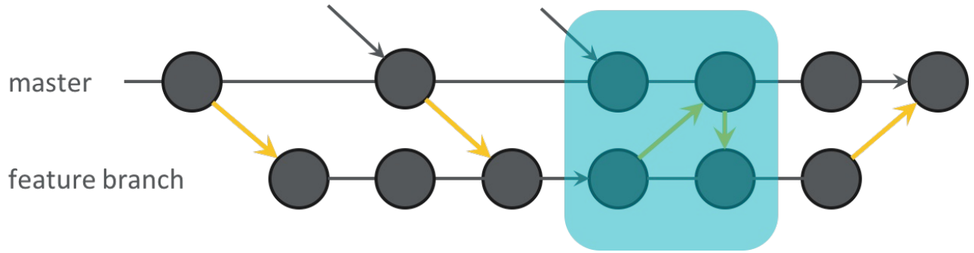
- Yes! Svnmerge plugin to the rescue
  - It's not Git, but Subversion has improved
- You commit to branches
- Jenkins will merge and commit to upstream



CLOUDBEES UNIVERSITY

CAVEAT

- Push and rebase need to happen in one go:



CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 11: Validated Merge

CLOUDBEES UNIVERSITY

PULL-REQUEST BUILDER FOR GITHUB

CLOUDBEES UNIVERSITY

WHAT IS PULL-REQUEST BUILDER FOR GITHUB

The Pull-Request Builder for GitHub plugin let you configure Jenkins to verify pull-request on a GitHub project and check the proposed changes validates Continuous Integration criteria set on your Jenkins instance.

CLOUDBEES UNIVERSITY

BENEFITS: EFFICIENCY

The GitHub Pull Requester Plugin can:

- automatically run unit/integration tests.
- merge the pull request upstream if it passes the unit/integration tests

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Download and install the GitHub Pull Request Builder Plugin.
- Configure the Plugin, add users to the Admin, and White List

CLOUDBEES UNIVERSITY

PIPELINE

Formerly known as Workflow

CLOUDBEES UNIVERSITY

WHAT IS A « JENKINS WORKFLOW »

CD is about setting a “pipeline”, from SCM commit to deployed application

Jenkins can chain jobs this way, but …

CLOUDBEES UNIVERSITY

JENKINS CD PIPELINE

a real world CD pipeline is way more complex !

- requires (complex) conditional logic
- requires resources allocation and cleanup
- involves human interaction for manual approval
- should be resumable at some point on failure

CLOUDBEES UNIVERSITY

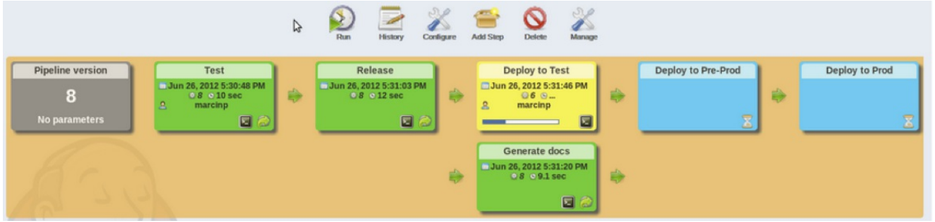
JENKINS CD PIPELINE

- many related plugins
  - conditional build steps
  - parameterized trigger
  - promotions
  - …
- Major issue : pipeline configuration is scattered in various jobs

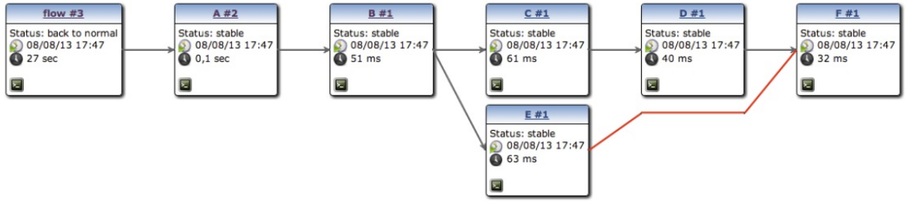
CLOUDBEES UNIVERSITY

CD PIPELINE VISUALIZATION

- pipeline plugin to render “linear” pipeline



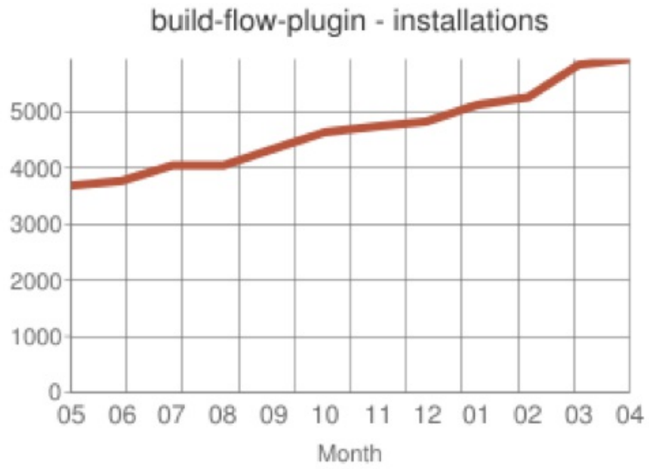
- build-graph-view to render complex ones



CLOUDBEES UNIVERSITY

BUILD-FLOW PLUGIN

- OSS, considered a proof of concept



- define CD pipeline as (possibly complex) job orchestration written with a Groovy DSL
- largely adopted
  - demonstrates direction is correct
- But major technical limitations

CLOUDBEES UNIVERSITY

CLOUDBEES PIPELINE PLUGIN

- inspired by build-flow
- Groovy DSL to orchestrate build steps (optionally jobs)
- extensible DSL syntax
- do supports savepoints
- advanced visualization
- plain text : Can be stored in SCM

CLOUDBEES UNIVERSITY

GROOVY DSL ?

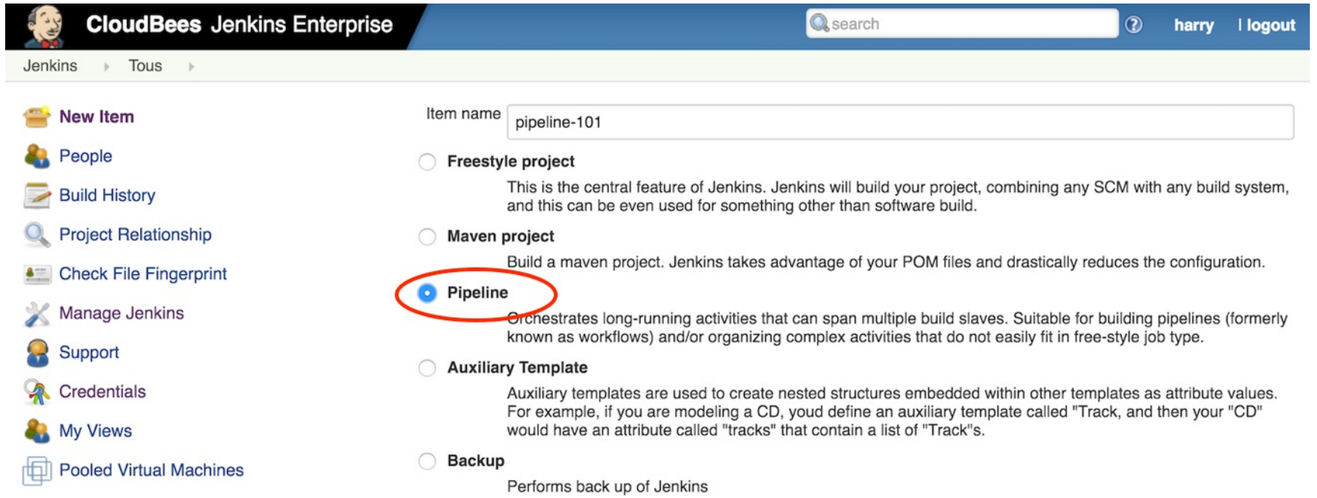
Groovy based “Domain Specific Language” to manage build step orchestration

- Groovy is widely used in Jenkins ecosystem. You can leverage on Groovy experience
- If you don’t like/care about Groovy, just consider the DSL as a dedicated simple syntax

CLOUDBEES UNIVERSITY

PIPELINE 101

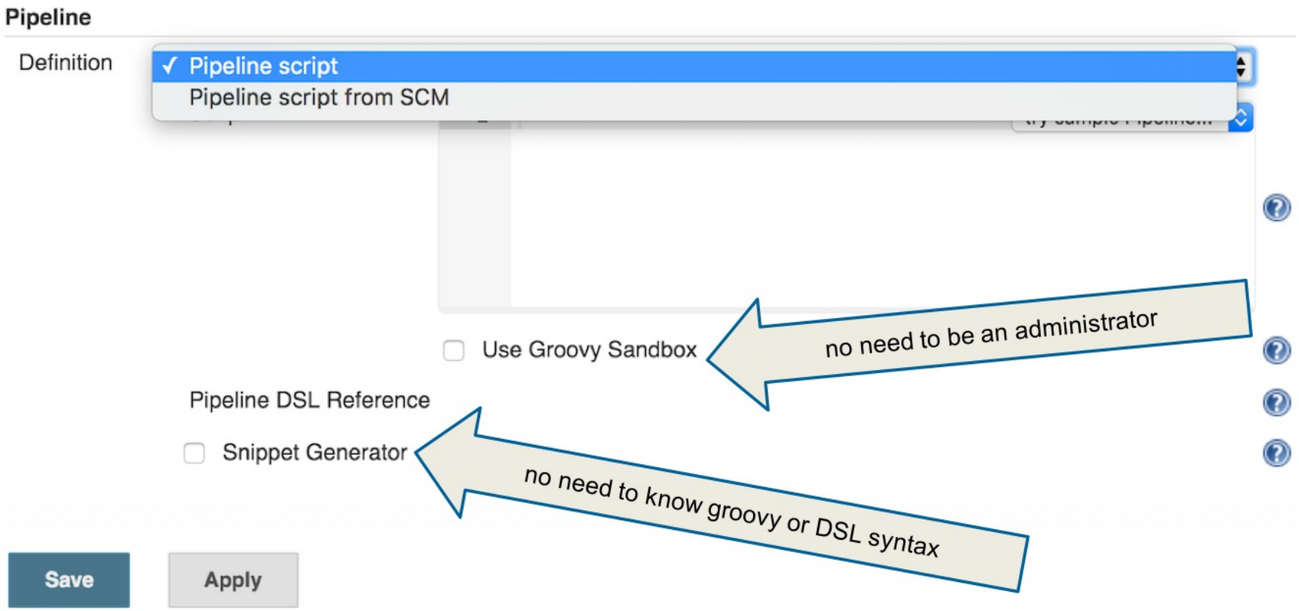
- A Pipeline is a new type of job:



CLOUDBEES UNIVERSITY

PIPELINE 101

Can define DSL in Jenkins, or store in SCM
= > possibly distinct pipeline per branch



CLOUDBEES UNIVERSITY

PIPELINE 101 - BASIC DSL SYNTAX

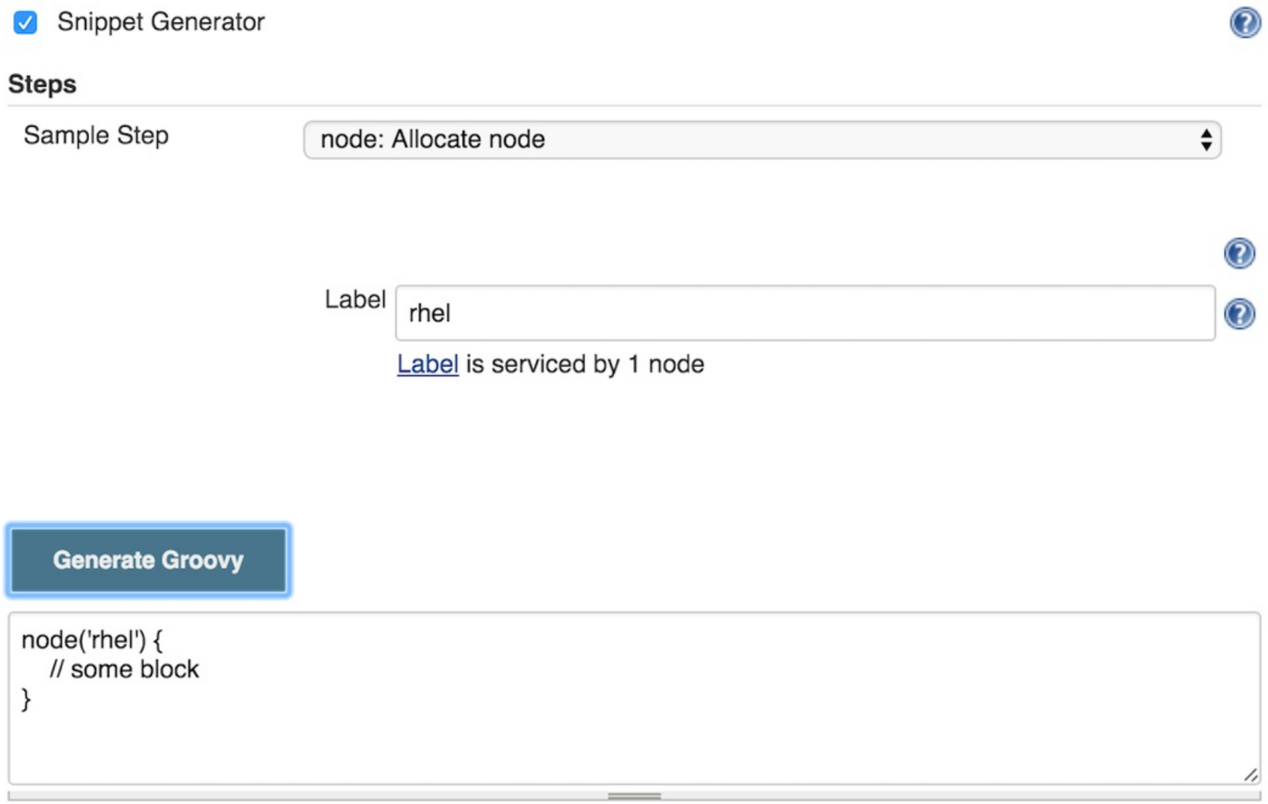
- DSL is used to define when/how to run build steps
- Those (most of them) will run on a node, i.e. a build executor
- Example with sh:

```
node('linux') { // run on node that are labelled as "linux"
  sh 'echo hello world' // a DSL keyword to run a shell script build step:
}
```

CLOUDBEES UNIVERSITY

PIPELINE SNIPPET GENERATOR

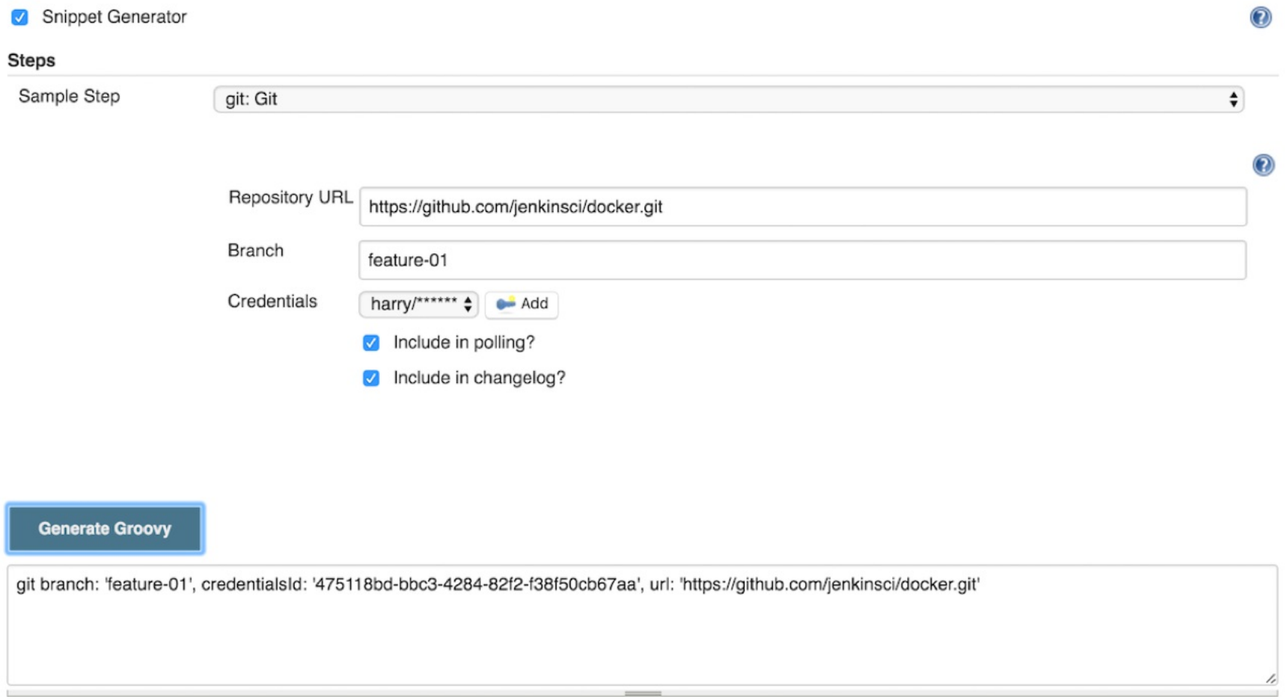
- No need to learn DSL syntax !



CLOUDBEES UNIVERSITY

PIPELINE 101 - BUILD STEPS

- Some have native DSL support



PIPELINE 101 - BUILD STEPS

Some don't (yet) have native DSL support and require some hack-ish invocation syntax - but still work:

CLOUDBEES UNIVERSITY

INVOCATION SYNTAX

- Most build steps can be used this way today, they may evolve to offer native pipeline support:

```
step( // DSL keyword to invoke arbitrary step
  [ // groovy map
      $class: 'build_step_class_name', // Same as in job's config.xml on disk
      constructor_argument: 'value',
      constructor_argument_2: 'value'
  ]
)
```

CLOUDBEES UNIVERSITY

HUMAN INTERACTION

A real world CD pipeline involves human interaction:

- manual validation steps
- approval

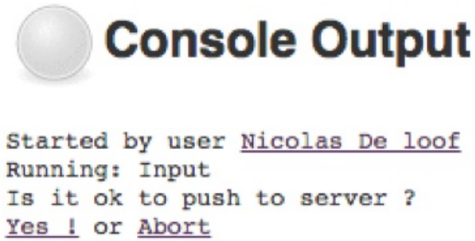
Jenkins users used to rely on Promoted Builds Plugin
but then lost the “pipeline execution flow” view

CLOUDBEES UNIVERSITY

HUMAN INTERACTION

```
input // DSL keyword for interactive step
  message: 'Is it ok to deploy ?',
  ok: 'Yes!', // Button caption
  submitter: 'boss' // Role restriction
```

- Will give this "output":



CLOUDBEES UNIVERSITY

EXECUTION CONTROL

- Try up to N times:

```
retry(10) {
  // some block
}
```

- Pause the flow:

```
sleep time: 10,
  unit: 'MINUTES'
```

- Wait for event:

```
waitUntil {
    // some block
}
```

- Timeout an operation:

```
timeout(time: 100,
   unit: 'SECONDS') {
    // some block
}
```

CLOUDBEES UNIVERSITY

EXECUTION CONTROL

You can also rely on Groovy control flow syntax:

```
while(something) {
   // do something
   if (something_else) {
      // do something else
   }
}
```

CLOUDBEES UNIVERSITY

CONCURRENCY

Pipeline can execute tasks in parallel :

```
parallel
  firstBranch: {
    // do something
  },
  secondBranch: {
    // do something else
  }
```

CLOUDBEES UNIVERSITY

FILE SYSTEM

- read file:

```
readFile file: 'some/file', encoding: 'UTF-8'
```

- write file:

```
writeFile file: 'some/file', text: 'hello'
```

CLOUDBEES UNIVERSITY

BUILD-FLOW COMPATIBILITY

- build-flow might be deprecated at some time
- Pipeline can trigger a job too:

```
build job: 'foo', wait: false
```

so could run a build flow script with some minor changes

CLOUDBEES UNIVERSITY

PIPELINE STAGE VIEW

CLOUDBEES UNIVERSITY

WHAT IS PIPELINE STAGE VIEW?

- When you have complex builds pipelines,
  it is useful to be able to see the progress of each stage.
- To that end, CJE includes an extended visualization
  of pipeline build history on the index page of a flow project.

Tip : (You can also click on Full Stage View to get a full-screen view.)

CLOUDBEES UNIVERSITY

DEFINING STAGES

- The stage keyword let you label parts of the pipeline :

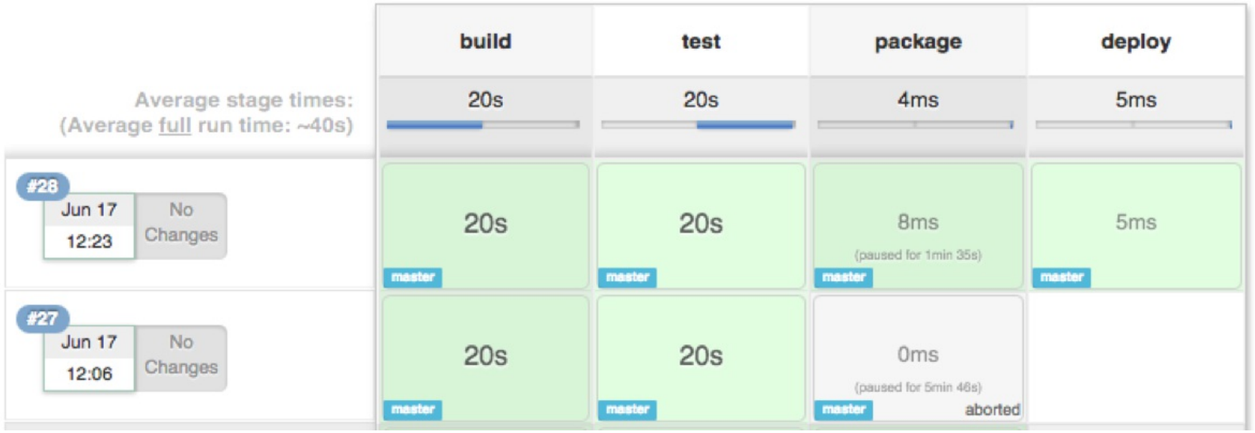
```
stage 'build'
  // Steps to build your application

stage 'test'
  // Steps to run tests against application

stage 'package'
  // Steps to packaging

stage 'deploy'
  // GO in production
```

This will be rendered by visualization

CLOUDBEES UNIVERSITY

VISUALIZING STAGES

After the first build, you can see your stages (and their states, logs, etc.) :



CLOUDBEES UNIVERSITY

CONCURENT STAGES

Stages also let you control concurrency:

```
stage name: 'load-tests',
    concurrency: 4 // spread across 4 nodes
```

CLOUDBEES UNIVERSITY

BENEFITS: SIMPLICITY

By adding Stage View to your Pipeline jobs it allows users to easily identify the progress of each stage within the job.

CLOUDBEES UNIVERSITY

CHECKPOINTS

CLOUDBEES UNIVERSITY

WHAT ARE CHECKPOINTS?

- The checkpoints plugin, offered through CloudBees Jenkins enterprise package, allows for Pipelines to be resumed at specified points.
- By adding a simple line of code to a pipeline script, a failed Pipeline build can be started at that specific checkpoint.
- Data is always preserved with restarted Pipelines and it is easy to pick up where the build left off.

CLOUDBEES UNIVERSITY

BENEFITS: RESTART PIPELINE WHERE THEY FAILED

- Without Checkpoints, failed long running Pipebuilds will need to be re-run from the start.
- With Checkpoints, you can simply place a checkpoint before a high risk operations. And if the build fails, Checkpoint will allow the build to be resumed from the checkpoint just before the failure.
- Caution: Checkpoints take additional disk space. You can delete a specific checkpoint to save space

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Create or modify an existing Pipeline to include a Checkpoint. Where is there a high risk operation? A task that depends on network communications or a large amount of disk space?
- Add a checkpoint before the high risk operation. For example:

```
checkpoint 'Built & Tested'
```

CLOUDBEES UNIVERSITY

WHERE DO WE START?

A note about Checkpoints:

- As started earlier, data is always preserved with Checkpoint'ed Pipelines and it is easy to pick up where the build left off
- This is accomplished by serializing the entire workspace and the pipeline script itself (hence the impact on disk usage)
- So updating a pipeline script and resuming it after a checkpoint will not run the new version of the pipeline
- This feature is really intended to resume the exact same pipeline from the checkpoint after an external issue has been fixed or resolved
- For example, compile, test, build stages pass, but deploy stage may fail due to connection issue to the deployment environment. You could resume from a checkpoint placed after the test stage to re-deploy once the connectivity issue is resolved

CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 12: Pipelines

CLOUDBEES UNIVERSITY

TEMPLATE PLUGIN

Template Plugin Overview

CLOUDBEES UNIVERSITY

WHAT ARE TEMPLATES?

- A template is a reusable piece of job configuration that only exposes a minimal and pre-defined set of parameters to users
- A template is internally translated to a “standard” Jenkins job configuration.
- A template can extend another template
- Changes to a template immediately apply to everywhere the template is used.

CLOUDBEES UNIVERSITY

WHAT ARE TEMPLATES?

attribute

named & typed variable

model

collection of attributes (& super model)

instance

particular values for all attributes in a model

transformer

translates model instance into real item

template

model + transform**er**

CLOUDBEES UNIVERSITY

WHAT ARE TEMPLATES?

- 5 different kinds of templates
  - Builder
  - Publisher
  - Job
  - Folder
  - Auxiliary (model only)

CLOUDBEES UNIVERSITY

BENEFITS: MAINTAINABILITY

- Manage a large set of jobs without duplicating configuration:
  - Configuration update is easily done in one place instead of doing a dangerous mass update using the groovy console for example
  - Easier onboarding of new teams/applications
- Manage a large set of jobs without unexpected subtle differences
  - You are explicitly controlling what may be different/customized

CLOUDBEES UNIVERSITY

BENEFITS: DELEGATION

- Delegate jobs creation without allowing users to do anything:
  - They can create jobs but they can customize only what the administrator allows
  - Enforce common enterprise standard across all applications (for example, Static Code Analysis to be done all projects)
- Delegate jobs creation without exposing to users all advanced options and switches that Jenkins may provide
  - Configuration settings are limited by the administrator who can specialize them to his targeted audience

CLOUDBEES UNIVERSITY

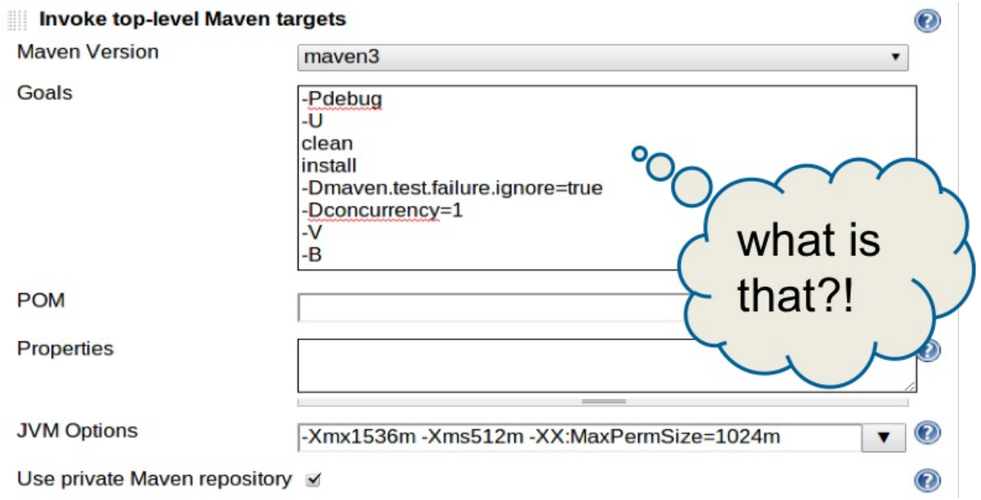
TEMPLATE PLUGIN

Builder templates

CLOUDBEES UNIVERSITY

WHAT ARE BUILDER TEMPLATES?

- A builder template allows to create your own simplified and reusable build step



CLOUDBEES UNIVERSITY

WHAT ARE BUILDER TEMPLATES?

- How does it work?
  - transformer produces a Jenkins build step
  - often a shell/batch script but need not be
  - transformation run during build, so can access live data
  - new builder immediately run

CLOUDBEES UNIVERSITY

BENEFITS: SIMPLICITY

- By using builder templates you avoid
  - copying & modifying too complex build steps
  - writing heavyweight Jenkins plugins (learning curve)

CLOUDBEES UNIVERSITY

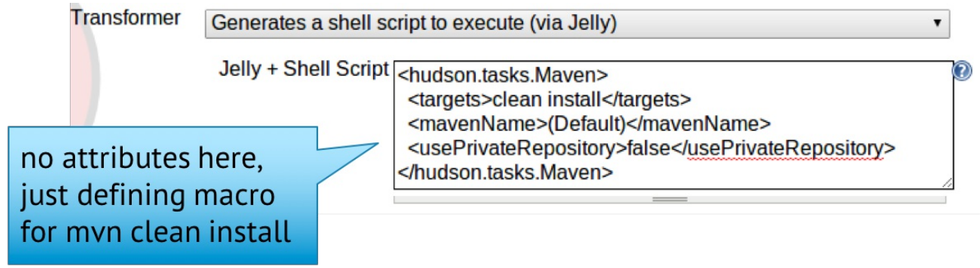
WHERE DO WE START?

- Using general transformers in a builder
  - awkward to do simple things
  - but can generate any builder: Gradle, wsadmin, &c.
  - useful if builder's plugin does tricky things
  - copy XML from a prototype job, then templatize
    - RFE in progress to load sample XML for you

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Using general transformers in a builder
  - Groovy for scripting power;
  - Jelly for XML hygiene (deprecated)



CLOUDBEES UNIVERSITY

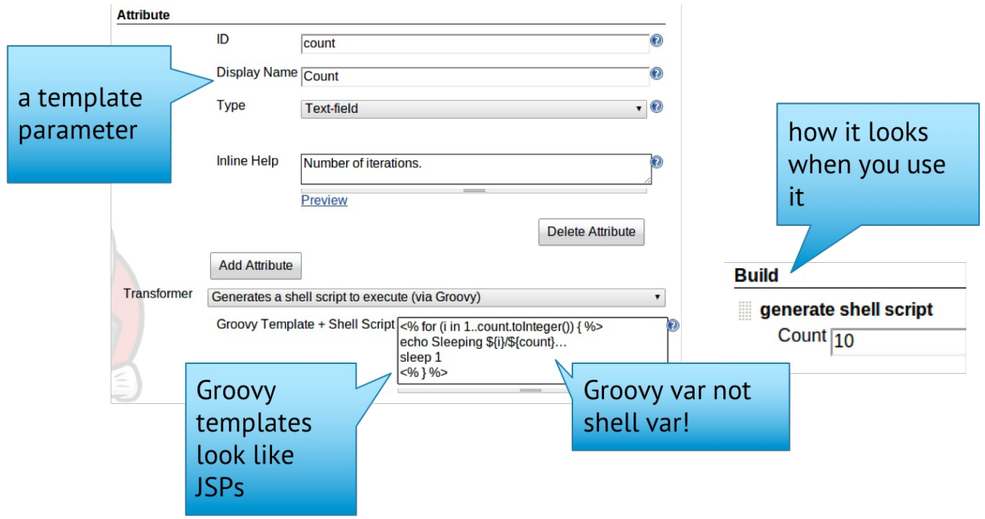
WHERE DO WE START?

- Generate shell scripts and run
  - easier if you would normally use a shell script anyway
  - write a template of a shell script (usually in Groovy)
  - do not confuse Groovy variables with shell variables

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Generate shell scripts and run



CLOUDBEES UNIVERSITY

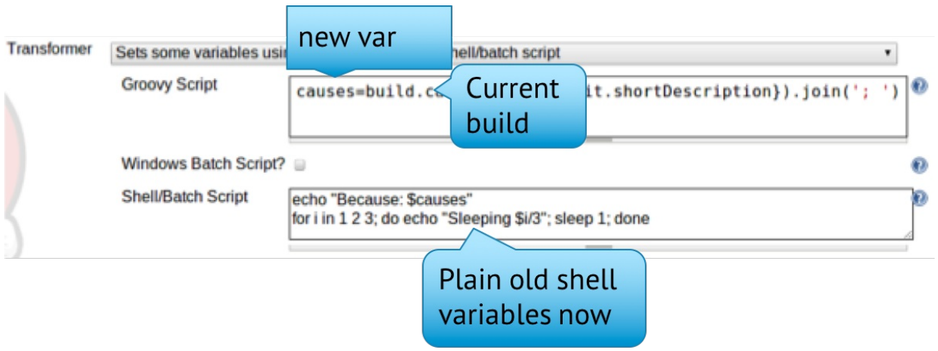
WHERE DO WE START?

- Predefine variables then run script
  - alternative keeping Groovy and shell syntax separated
  - supports Windows batch scripts as well
  - special characters (spaces, quotes, …) preserved
  - all control flow must be done by shell

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Predefine variables then run script
  - Definition:



  - Output:

```
[workspace] $ /bin/sh -e /tmp/hudson6506427550849825755.sh
Because: Started by user anonymous
Sleeping 1/3
Sleeping 2/3
Sleeping 3/3
```

CLOUDBEES UNIVERSITY

TEMPLATE PLUGIN

Publisher templates

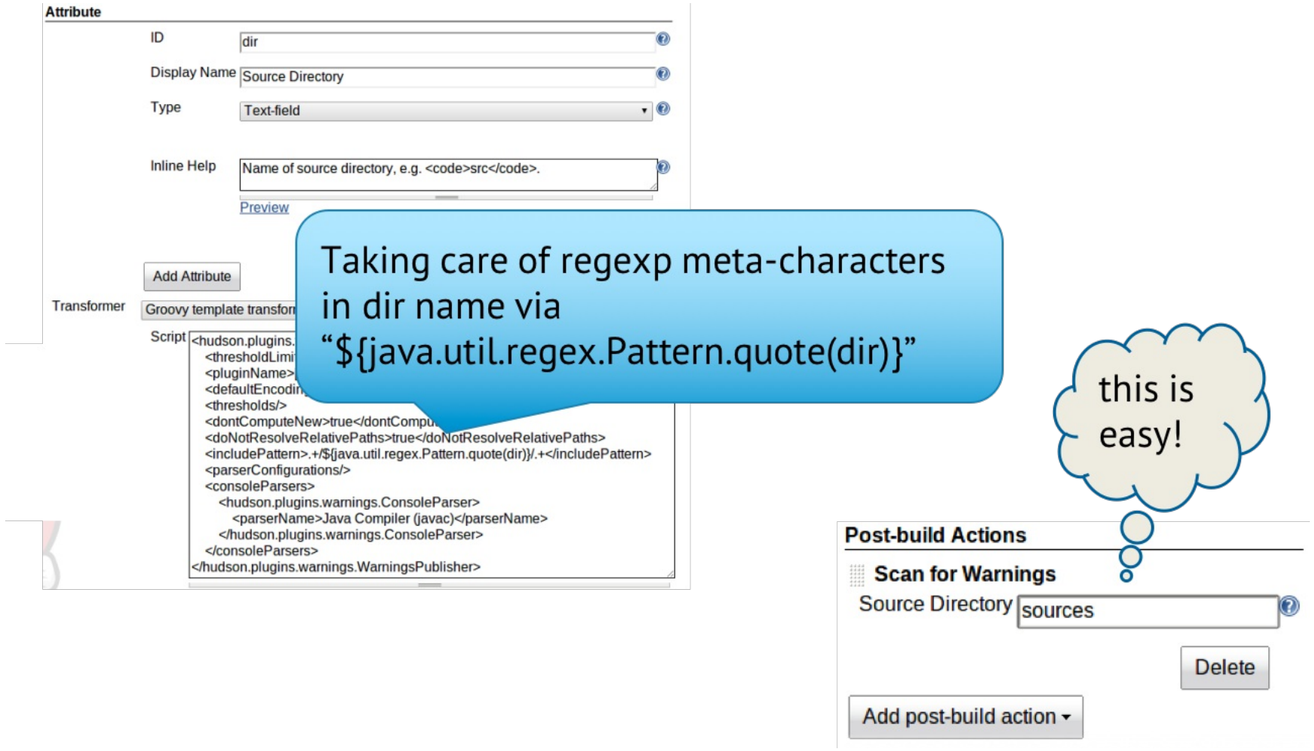
CLOUDBEES UNIVERSITY

WHAT ARE PUBLISHER TEMPLATES?

- much like builders, except for post-build actions
- publish HTML or test results, send email, …
- only general transformers (Groovy/Jelly) supported

CLOUDBEES UNIVERSITY

WHERE DO WE START?



CLOUDBEES UNIVERSITY

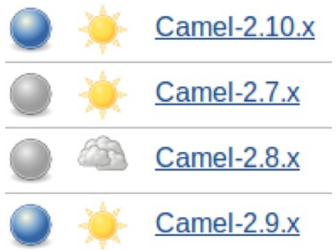
TEMPLATE PLUGIN

Job templates

CLOUDBEES UNIVERSITY

WHAT ARE JOB TEMPLATES?

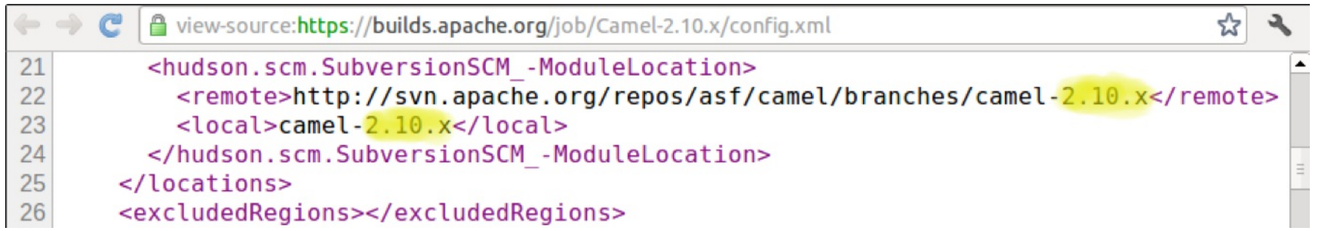
- A maze of twisty little projects, all alike
  - many Jenkins jobs are very similar
    - want to capture the common parts
  - not everyone creating a job is a Jenkins expert
  - managers want to know what varies and what does not
  - security setup may prohibitarbitrary job configs
  - job templates to the rescue!



CLOUDBEES UNIVERSITY

WHERE DO WE START?

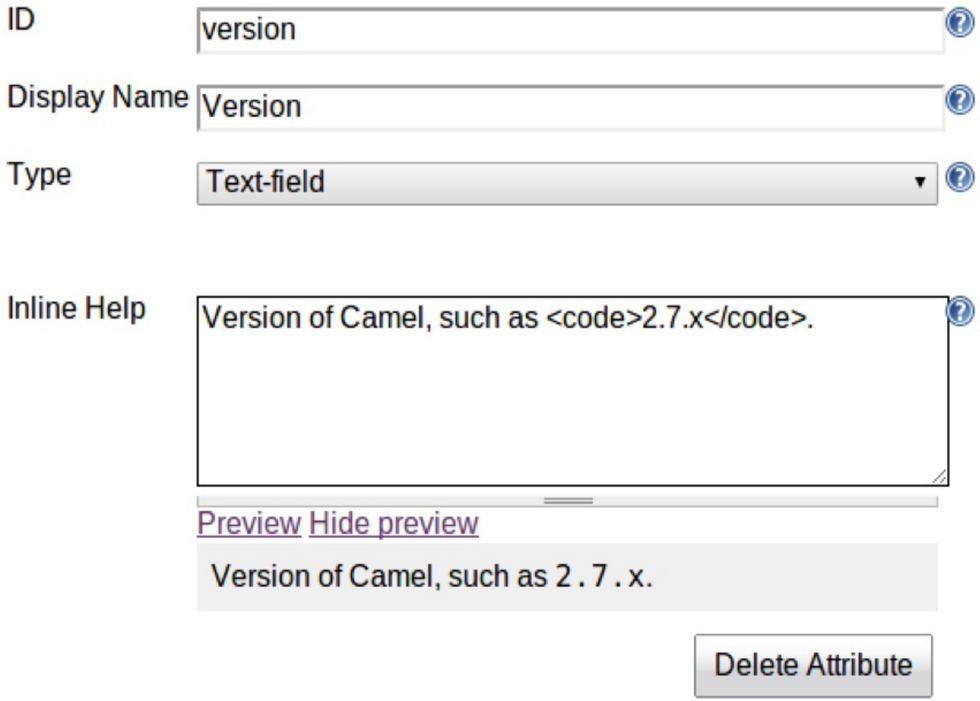
- Defining a job template
  - start with config.xml from a working job
  - decide which parts should be customizable



CLOUDBEES UNIVERSITY

WHERE DO WE START?

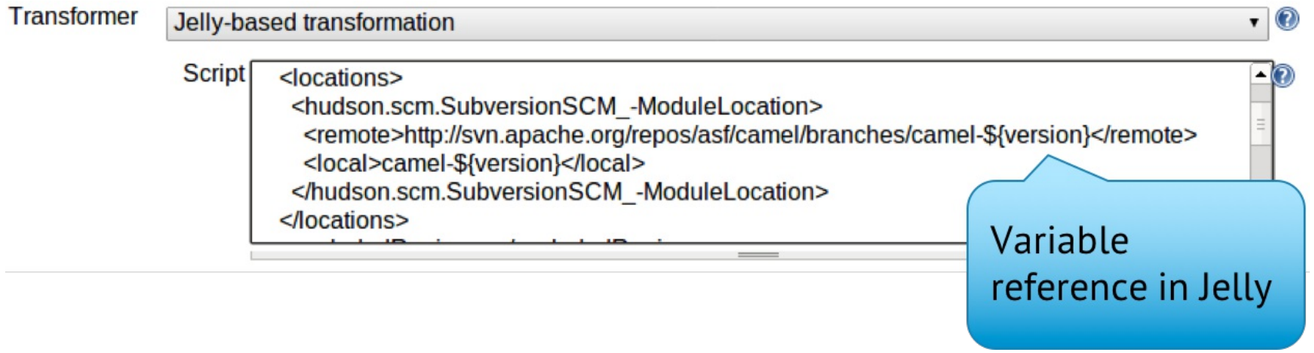
- Defining a job template
  - define and document attributes



CLOUDBEES UNIVERSITY

WHERE DO WE START?

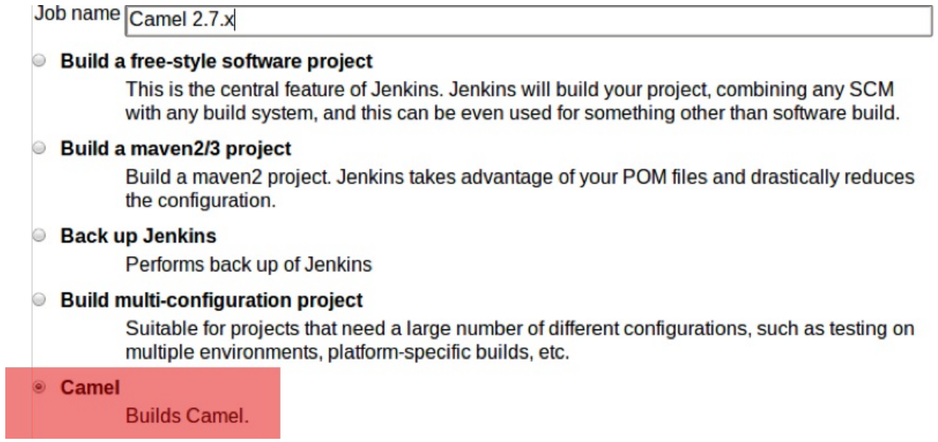
- Defining a job template
  - pick a transformation language (Groovy, Jelly)
  - make config.xml into a template



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Instantiating a job template
  - "New Job" page:



  - Configuration screen:

CLOUDBEES UNIVERSITY

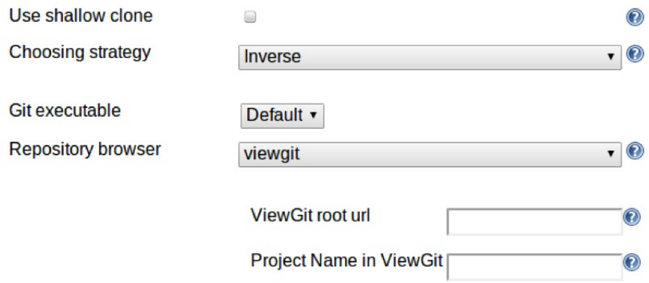
TEMPLATE PLUGIN

Auxiliary models and fancier template attributes

CLOUDBEES UNIVERSITY

WHAT ARE ATTRIBUTES?

- Attribute types
  - so far all attributes have been simple text fields
  - complex templates demand more!
  - after all, plain Jenkins jobs have a rich config UI…



CLOUDBEES UNIVERSITY

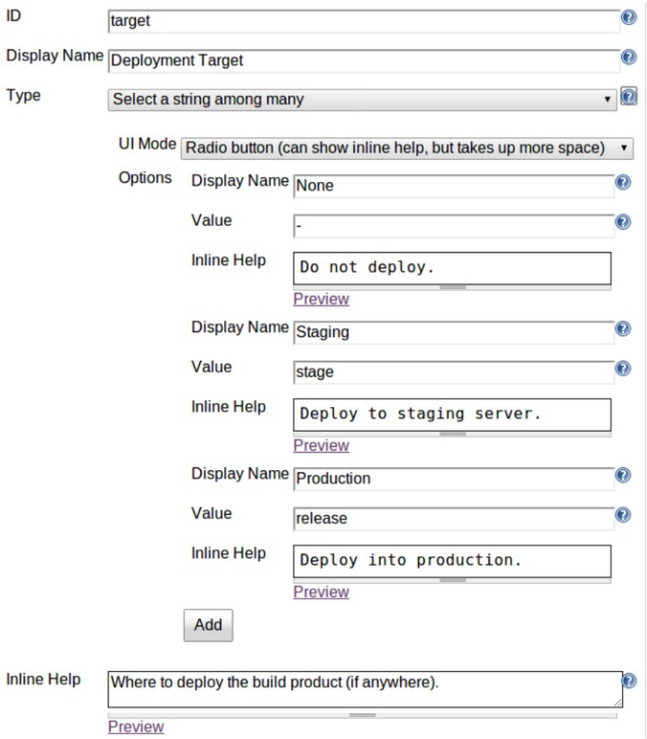
WHAT ARE ATTRIBUTES?

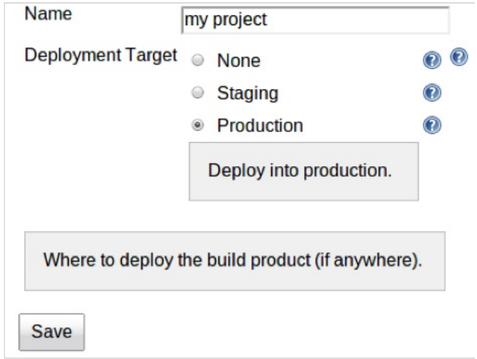
- Some basic attribute types:
  - text field ~ String (what we have seen)
  - text area ~ String (same but multiline)
  - checkbox ~ boolean

CLOUDBEES UNIVERSITY

WHAT ARE ATTRIBUTES?

- Select from enumerated items:





CLOUDBEES UNIVERSITY

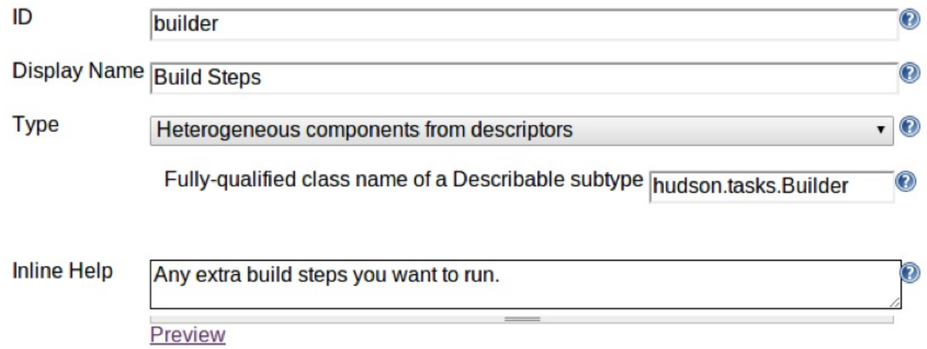
WHAT ARE ATTRIBUTES?

- Jenkins-specific attribute types
  - “item” (~ job/project)
  - installation of some type of tool (e.g. JDK)
  - “component” of some type, such as builder

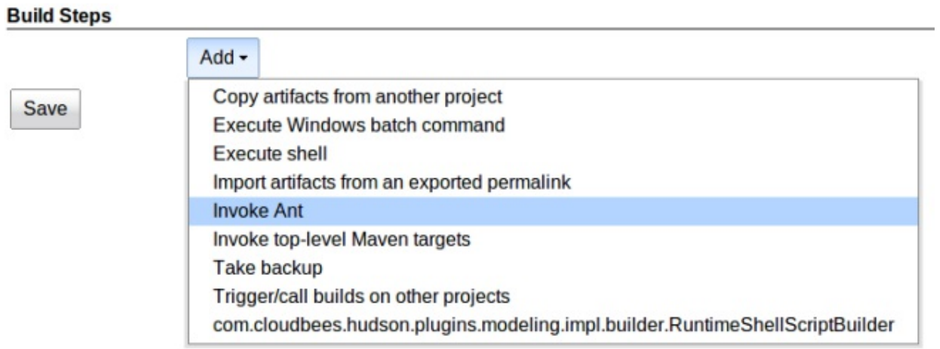
CLOUDBEES UNIVERSITY

WHAT ARE ATTRIBUTES?

- Jenkins-specific attribute types:
  - Definition:



  - What users see:



CLOUDBEES UNIVERSITY

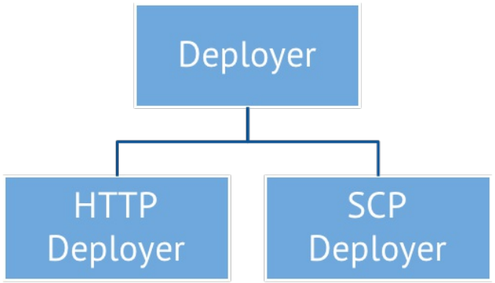
WHAT ARE AUXILIARY MODELS?

- Auxiliary models and inheritance:
  - like defining your own struct/class (with config GUI)
  - model only used inside other models
  - object-oriented style
    - “abstract” models
    - subtypes
    - polymorphic attributes
    - virtual attributes
  - used for complex, structured configuration

CLOUDBEES UNIVERSITY

WHERE DO WE START?

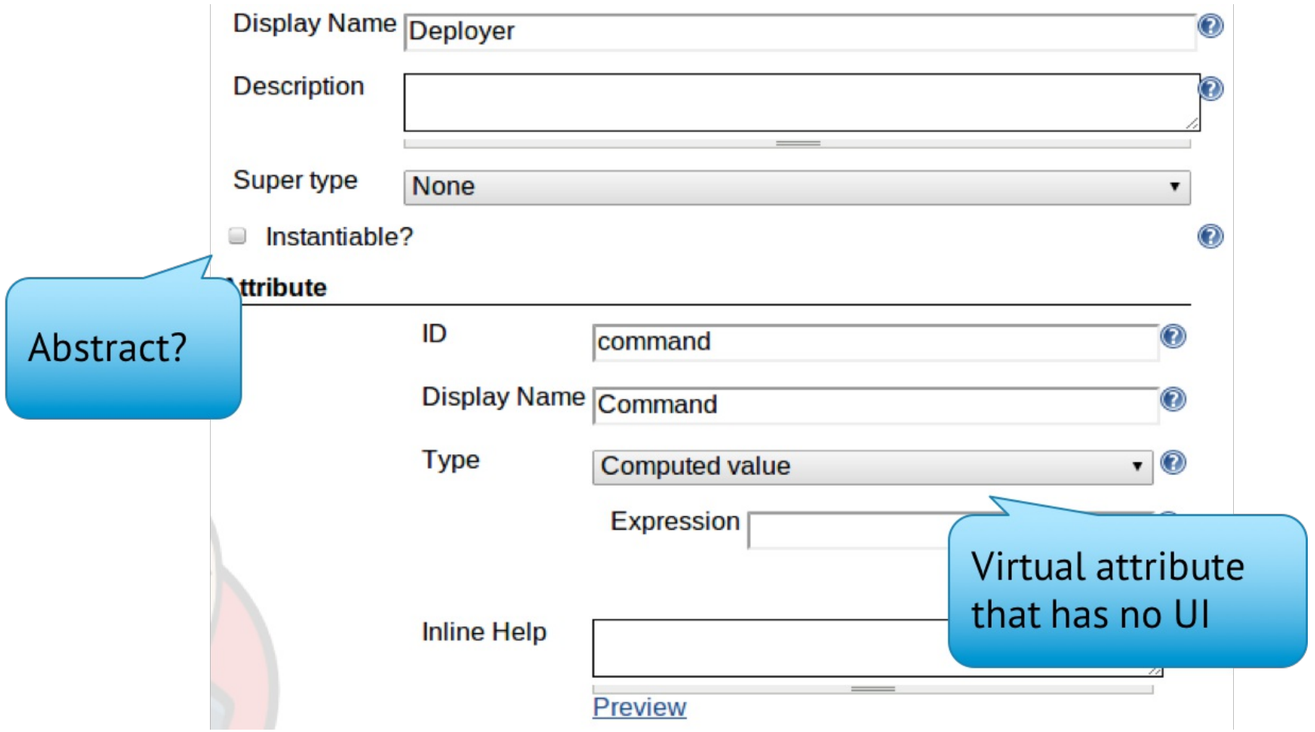
- Auxiliary model example:
  - Scenario
    - We build webapp
    - Team decides where to deploy
    - Two types of deployment (via HTTP or SCP)
  - How we model this
    - Via 3 models
    - With inheritance



CLOUDBEES UNIVERSITY

WHERE DO WE START?

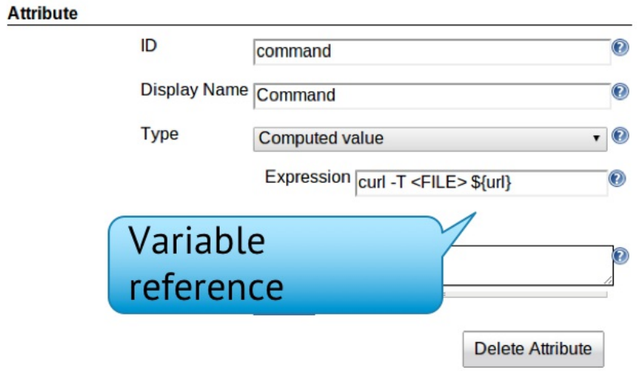
- Defining an abstract auxiliary model:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

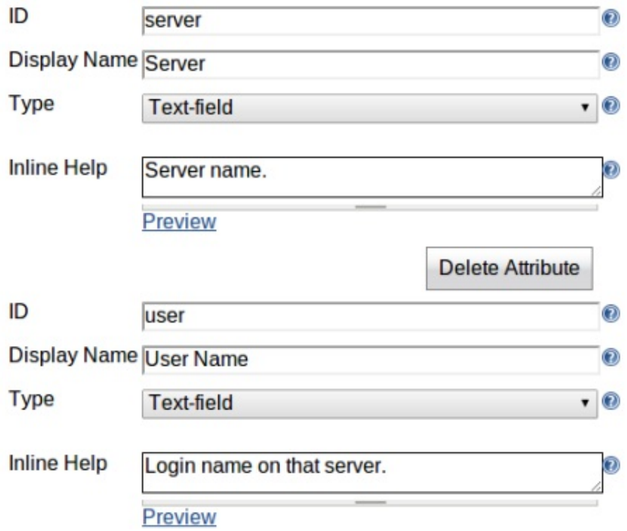
- Defining auxiliary model subtypes
  - Marked as “instantiable”
  - Extends from “Deployer”
  - “command” attribute concretely defined



CLOUDBEES UNIVERSITY

WHERE DO WE START?

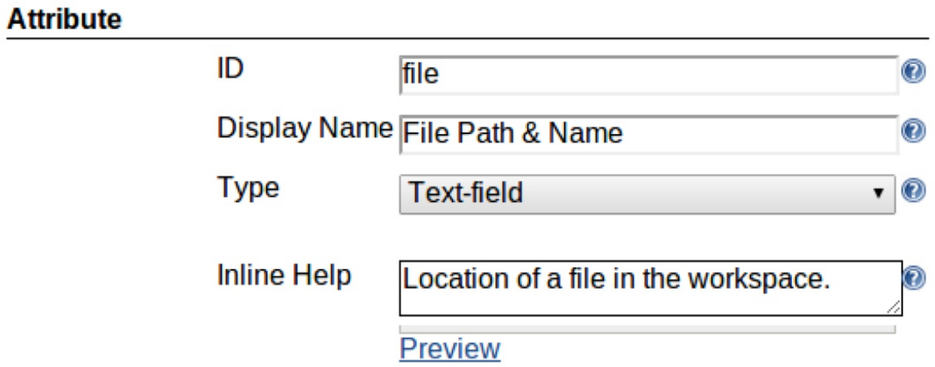
- Defining auxiliary model subtypes
  - Another subtype defined similarly



CLOUDBEES UNIVERSITY

WHERE DO WE START?

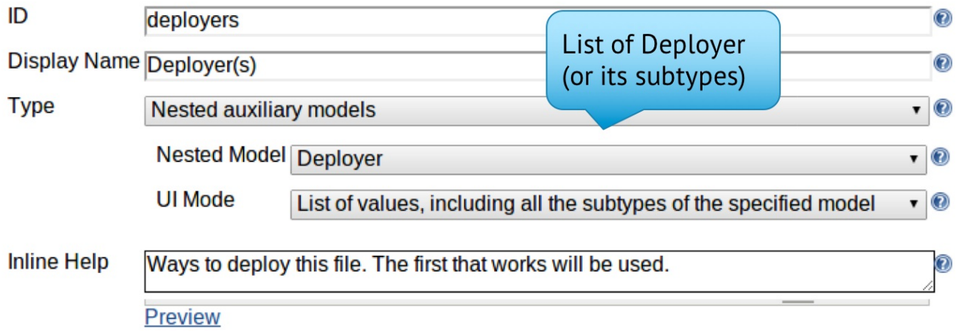
- Use deployer from builder template
  - First, define the file attribute:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Then let users specify deployers



CLOUDBEES UNIVERSITY

WHERE DO WE START?

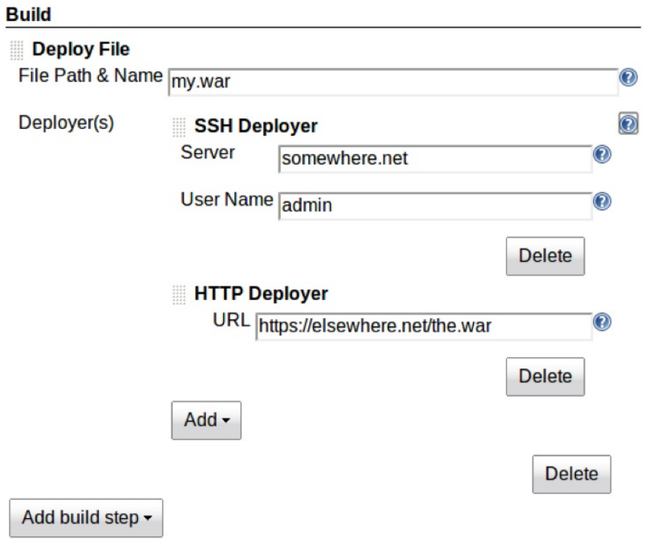
- Produce command that runs
  - Bit of Groovy to generate the command to run

```
cmd = deployers.
    toList()*.
    command*.
    toString()*.
    replace('<FILE>',file).
    join('\n')
```

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- And the user experience …




```
[workspace] $ /bin/sh -e /tmp/hudson8063706505027028169.sh
+ scp my.war admin@somewhere.net:~ || curl -T my.war https://elsewhere.net/the.war
Finished: SUCCESS
```

CLOUDBEES UNIVERSITY

TEMPLATE PLUGIN

Folder templates

CLOUDBEES UNIVERSITY

WHAT IS A FOLDER TEMPLATE?

- like a job template, but produces a whole folder
- may predefine:
  - job type restrictions
  - icon
  - columns
  - variables passed to jobs
- template produces config.xml as for a job
- can also define initial list of jobs
- admins can control what happens inside!

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Folder template example
- Scenario:
  - Team uses one branch for one sprint
  - Each branch requires build & test jobs
- How we model this:
  - Let's map one branch to one folder
  - Create a folder template so that build & test jobs are there from get-go

CLOUDBEES UNIVERSITY

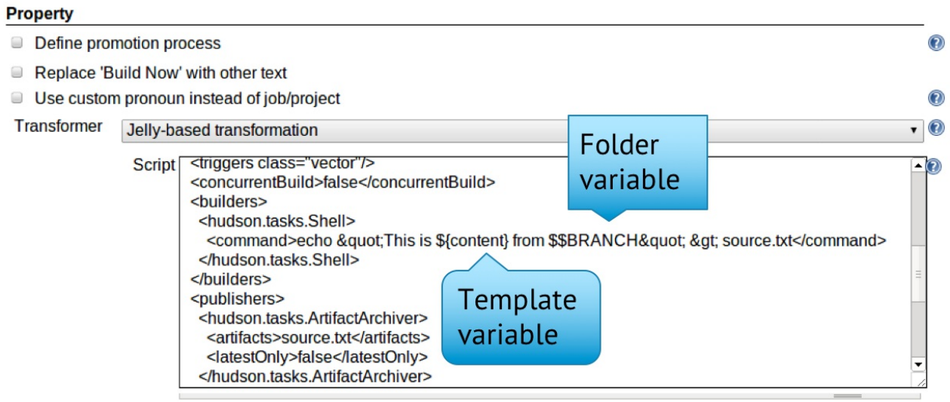
WHERE DO WE START?

- How to create them ?
- The same as before… by example!
  - Manually create a folder
  - Manually create two jobs in them
  - Take their config.xml and create templates

CLOUDBEES UNIVERSITY

WHERE DO WE START?

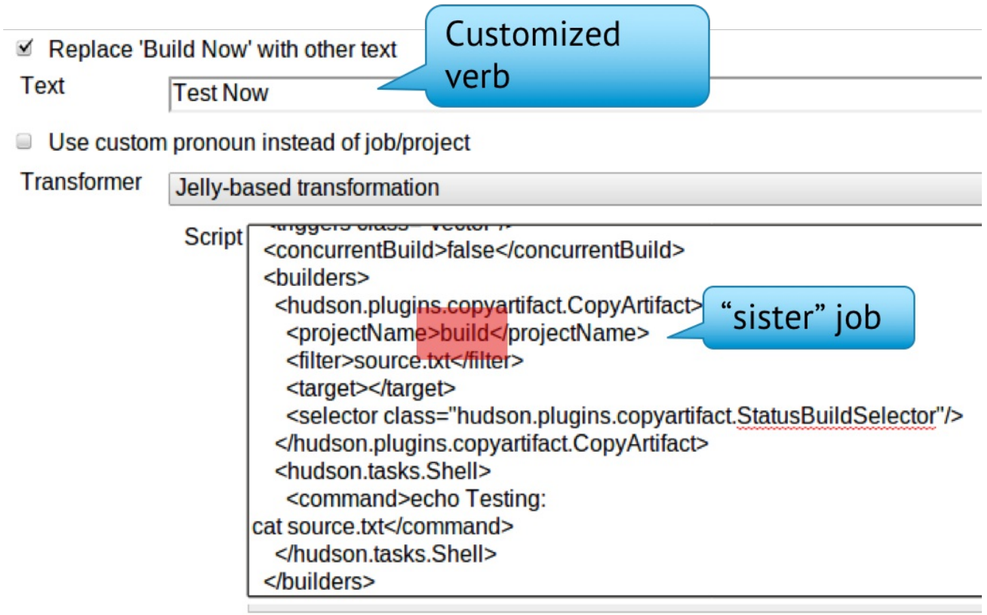
- Setup: job template build
  - Define “content” attribute



CLOUDBEES UNIVERSITY

WHERE DO WE START?

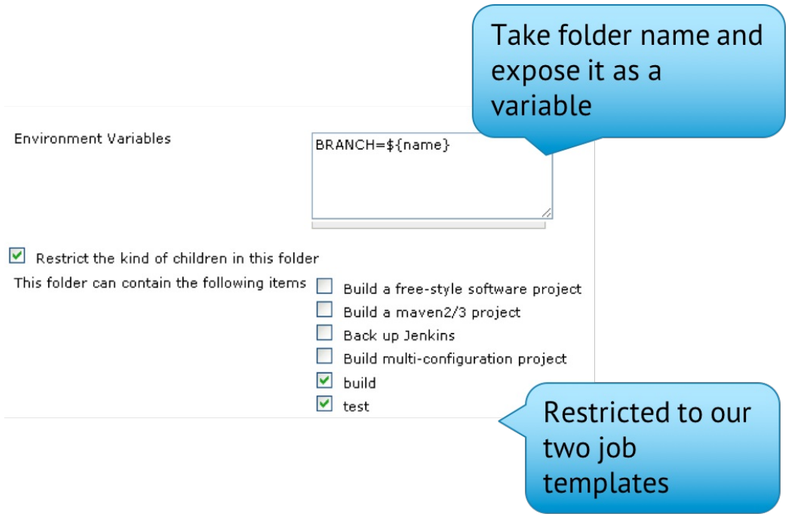
- Setup: job template test



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Create a folder that becomes a template
  - Create a folder by any name
  - Configure it the way we like

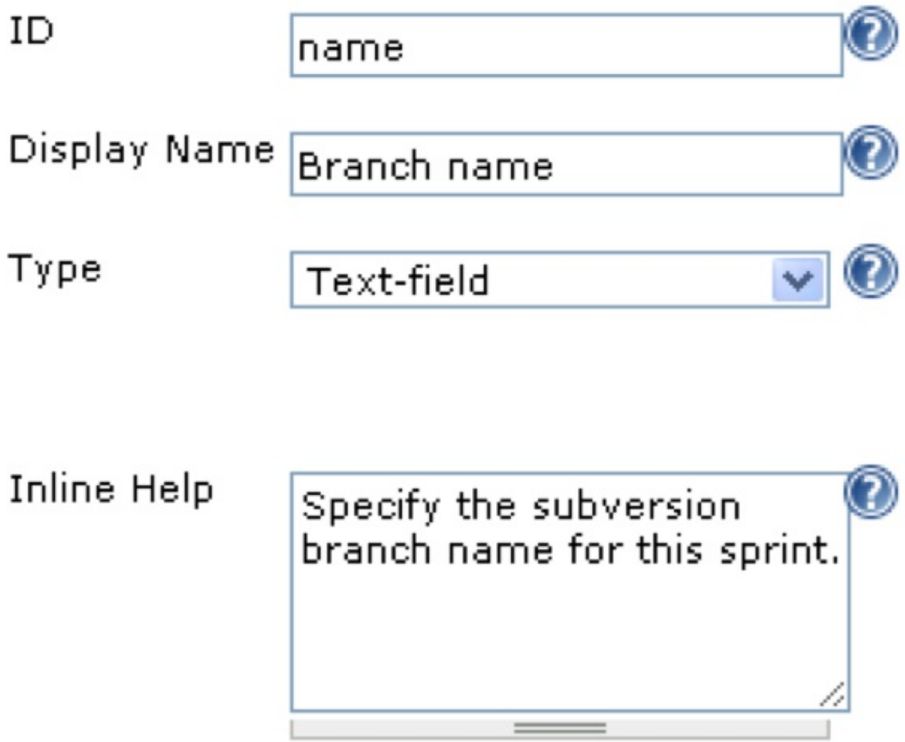


  - Copy config.xml into clipboard

CLOUDBEES UNIVERSITY

WHERE DO WE START?

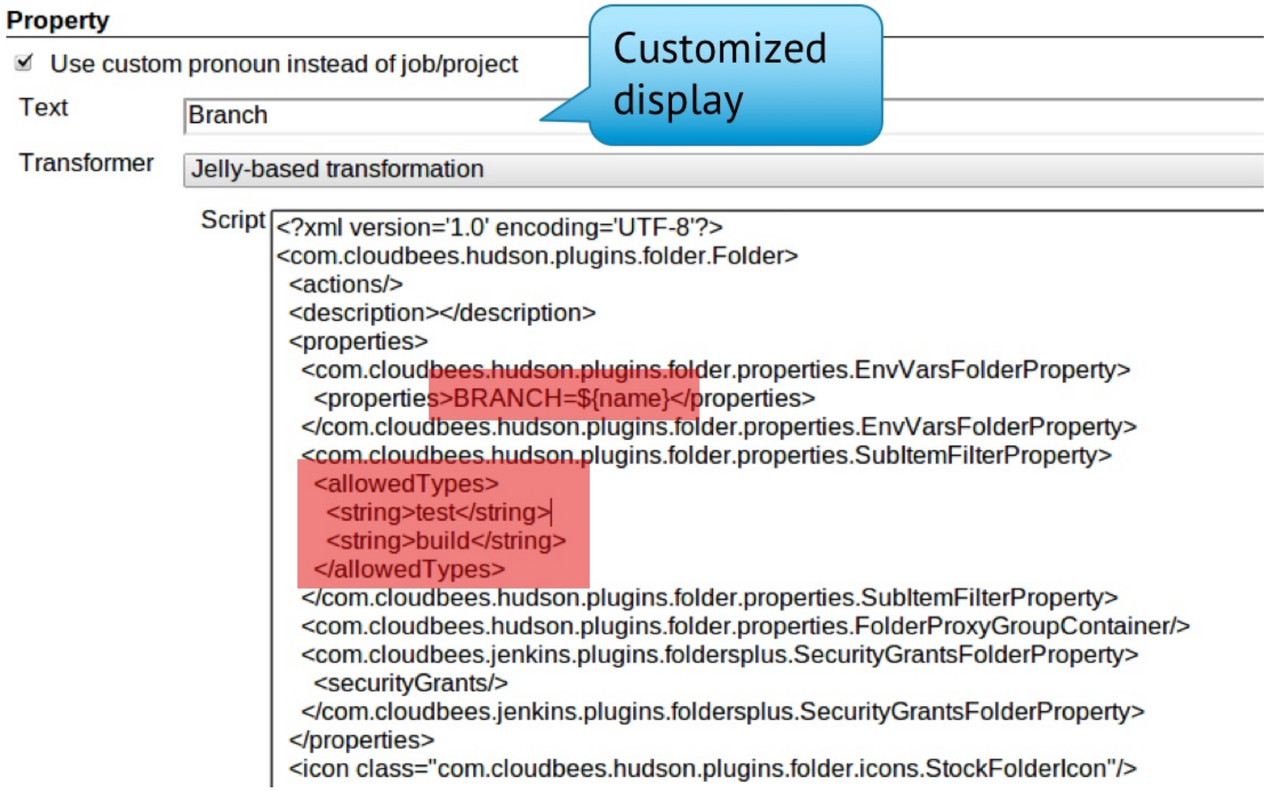
- Folder template branch
  - Define the branch name attribute:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

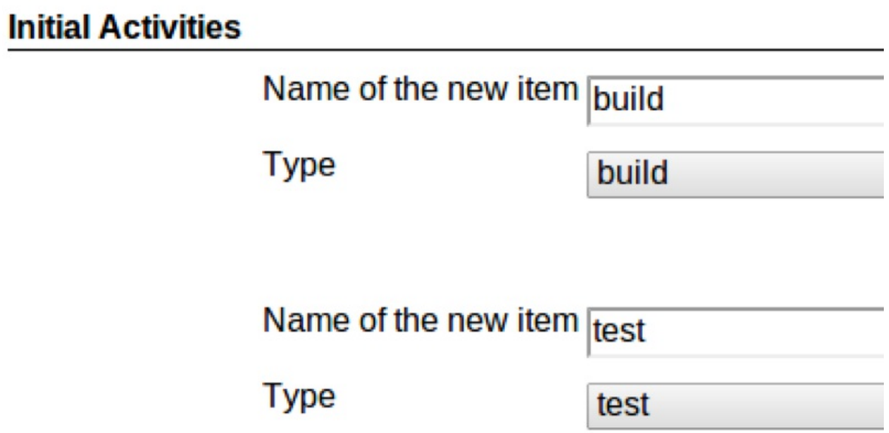
- Folder template branch:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Folder template branch
  - Create build & test jobs when a folder is created:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Going one more level!
  - Scenario:
    - Many teams work on different products in your org
  - Let's define "product":
    - As a collection of branches

CLOUDBEES UNIVERSITY

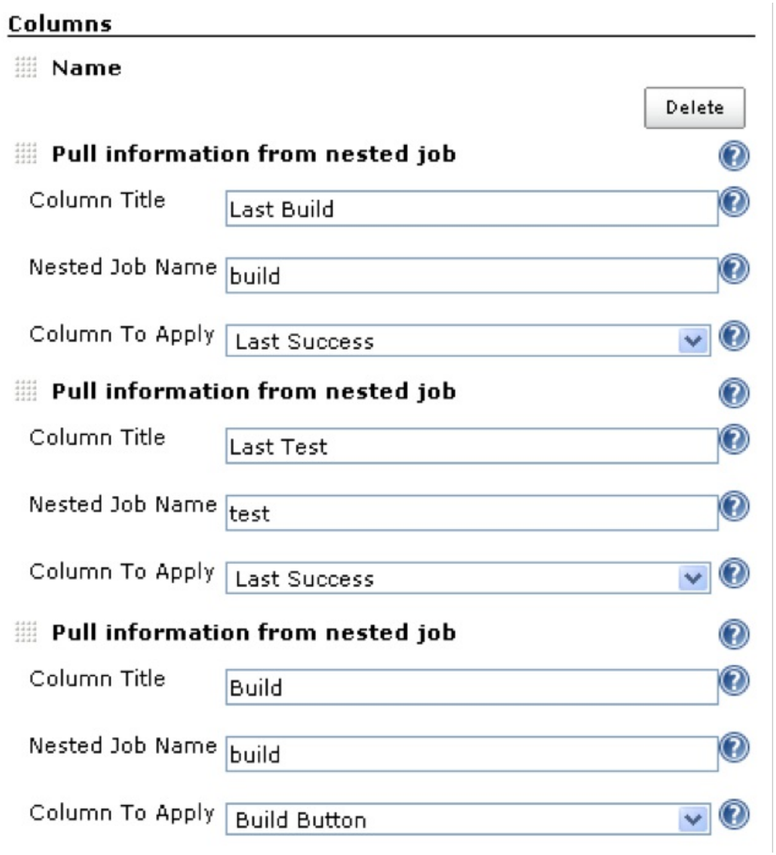
WHERE DO WE START?

- Create a folder that becomes a template
  - Create a folder by any name
  - Configure it the way we like
    - Restrict children to “branch”
    - Custom folder icon

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Configure it the way we like (cont'd)
  - Column configuration:



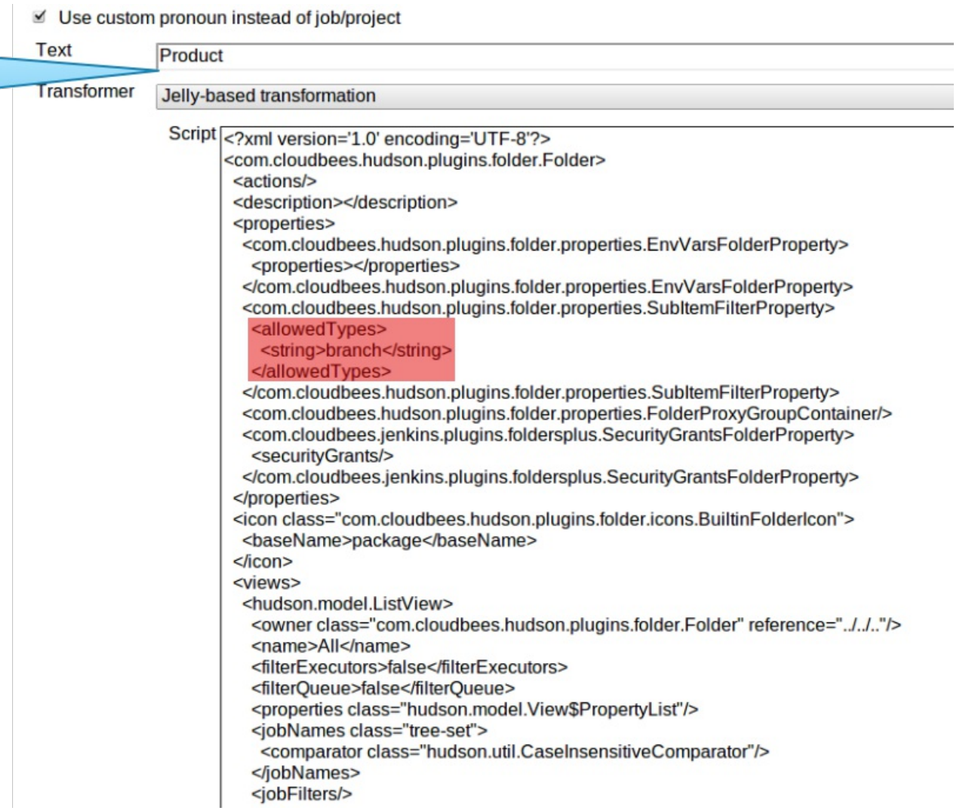
Show information about job inside the folder

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Take config.xml and paste that as XML:

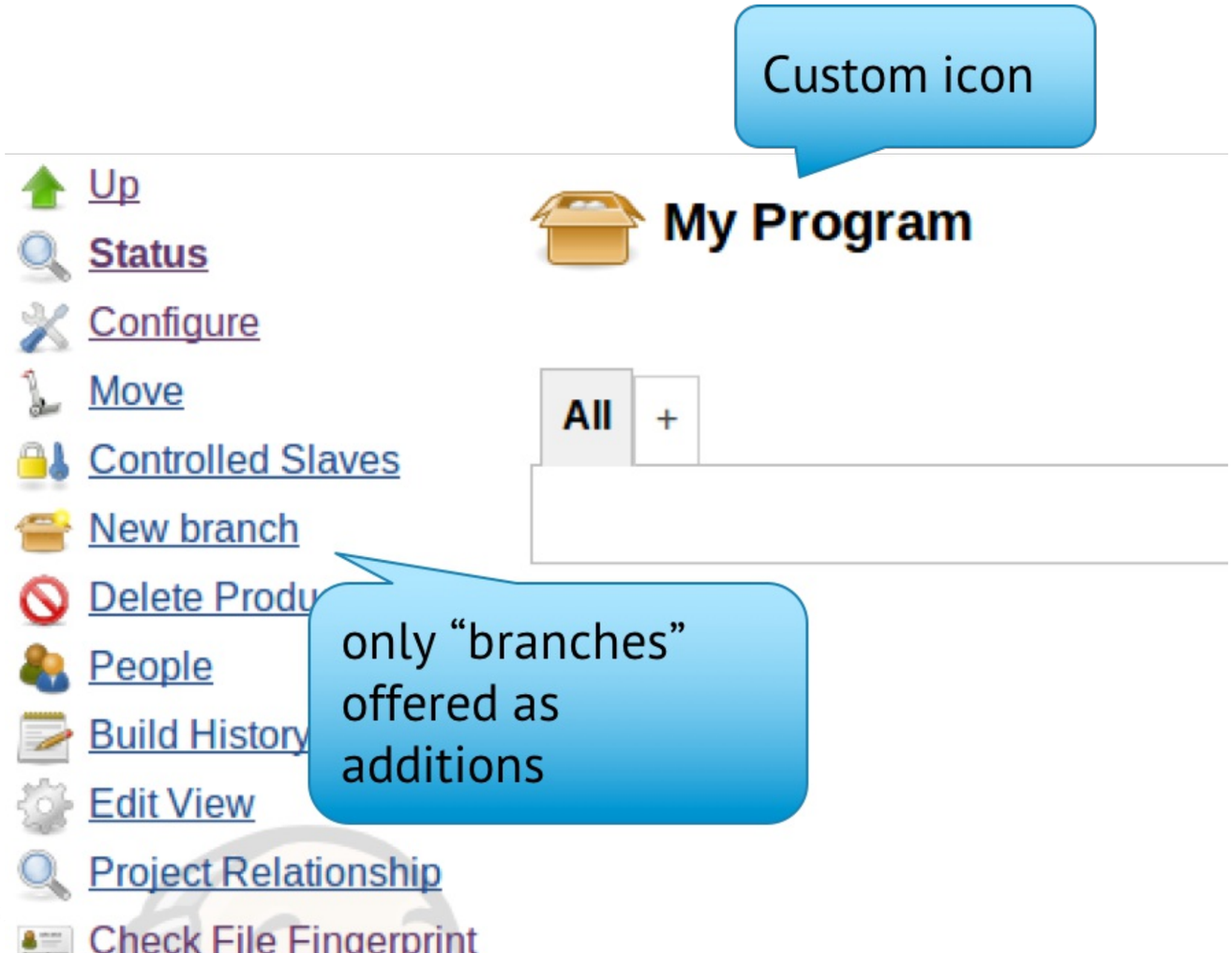
more custom display



CLOUDBEES UNIVERSITY

WHERE DO WE START?

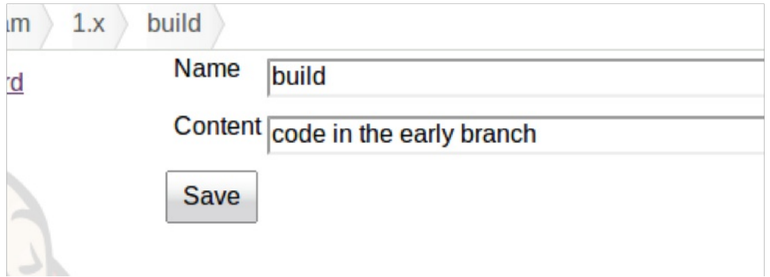
- Making a “product” and “branches”
  - initially empty…



CLOUDBEES UNIVERSITY

WHERE DO WE START?

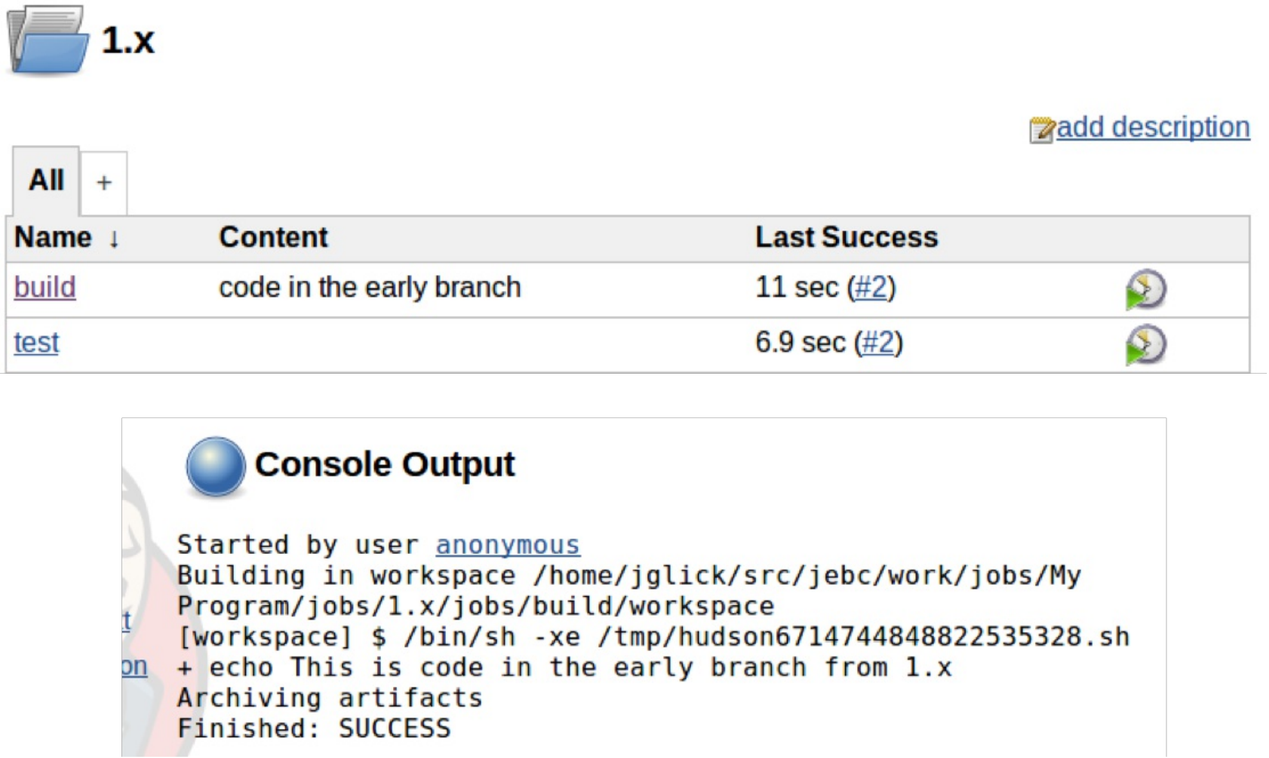
- Adding a branch and customizing its build job:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

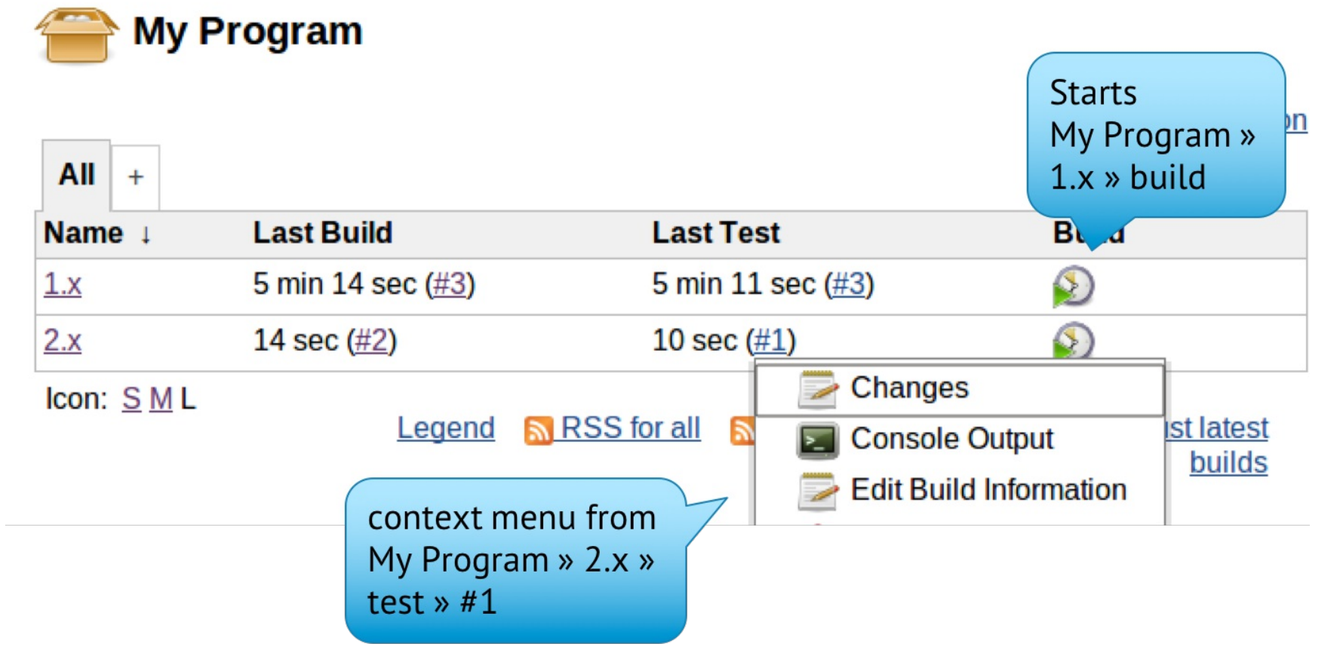
- Running build and test jobs inside a branch:



CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Product folder now shows overview of branches:



CLOUDBEES UNIVERSITY

LAB EXERCISE:

Lab 13: Templates

CLOUDBEES UNIVERSITY

CLOUDBEES SUPPORT PLUGIN

Accelerate diagnosis with CloudBees Support

CLOUDBEES UNIVERSITY

WHAT IS CLOUDBEES SUPPORT

- CloudBees customers have access to a whole set of services including Professional Support by Jenkins experts who are themselves developers and Jenkins maintainers.
- The Support plugin provides the ability to generate a bundle of all the commonly requested information used by CloudBees when resolving support issues.
  - Those bundles will be used within the CloudBees Support Portal

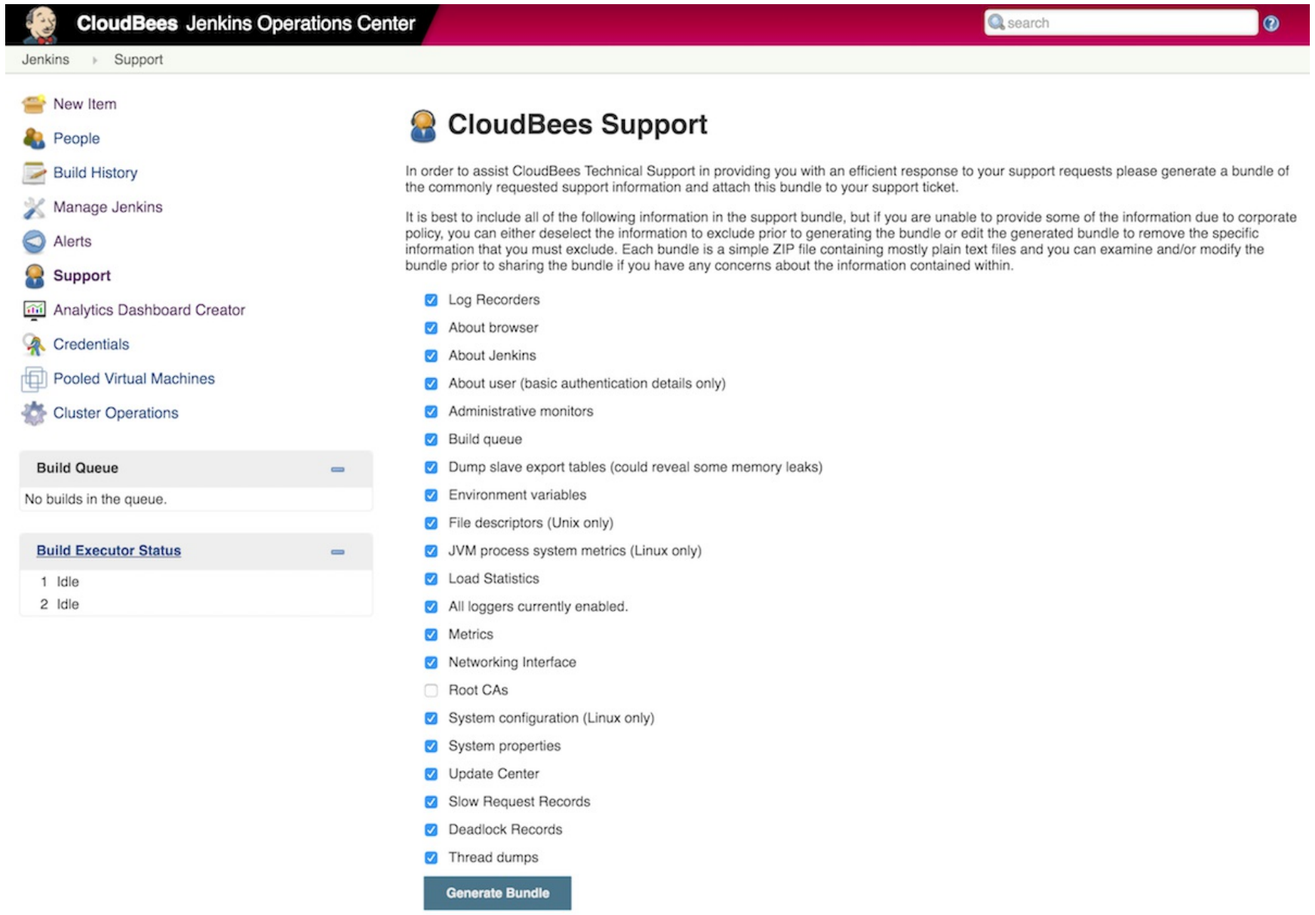
CLOUDBEES UNIVERSITY

WHERE DO WE START?

- Simply go to the plugin's release notes, pick the latest version, and click the link in the Download section, such as to http://jenkins-updates.cloudbees.com/download/plugins/cloudbees-support/latest/cloudbees-support.hpi). You can install the downloaded plugin using Manage Jenkins » Manage Plugins » Advanced » Upload Plugin
- To generate a bundle, simply go to the CloudBees Support link on the Jenkins instance.

CLOUDBEES UNIVERSITY

WHERE DO WE START?(CONT.)



CLOUDBEES UNIVERSITY

LAB EXERCISE

Lab 14: CloudBees Support

CLOUDBEES UNIVERSITY

INTRODUCTION TO CJOC

CloudBees Jenkins Operations Center

CLOUDBEES UNIVERSITY

JENKINS @ SCALE

CLOUDBEES UNIVERSITY

HOW JENKINS GROWS IN AN ORGANIZATION ?

~~Horizontally...~~ virally

Vertically

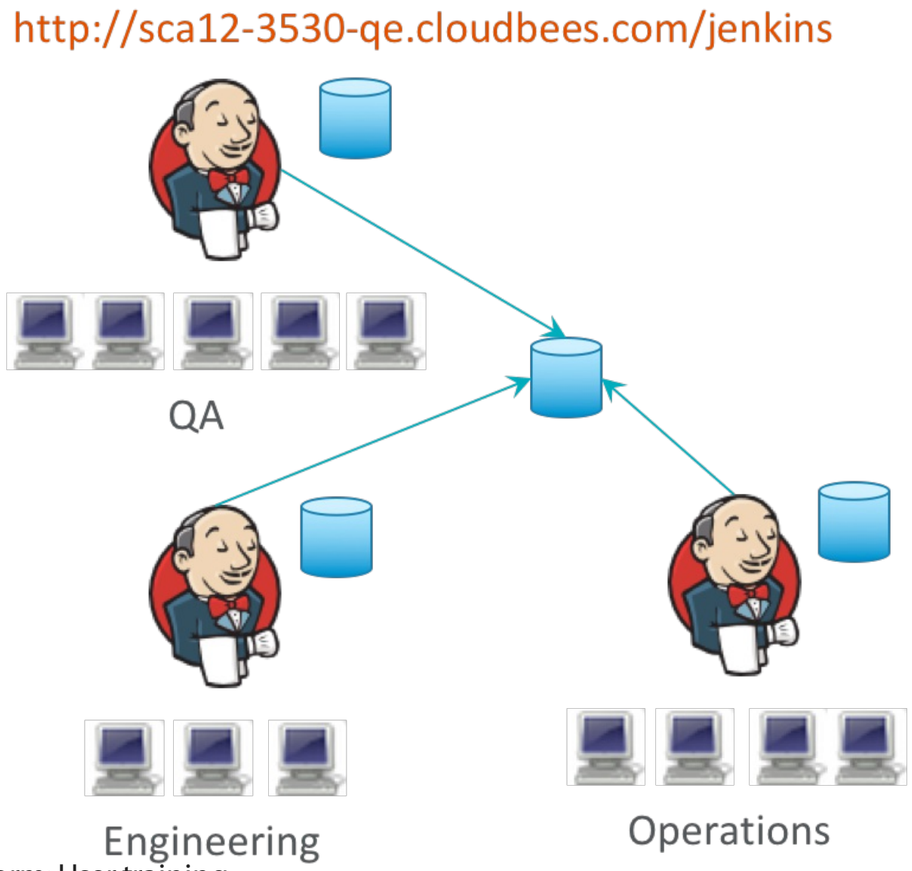
CLOUDBEES UNIVERSITY

VERTICAL SCALING ISSUES

- How long can I keep scaling Jenkins?
- Single point of failure:

HORIZONTAL SCALING ISSUES

- I would like to…
  - Share security information: Security Realm and roles
  - Easily navigate between masters



CLOUDBEES UNIVERSITY

INTRODUCING CLOUDBEES JENKINS OPERATIONS CENTER

aka "CJOC"

CLOUDBEES UNIVERSITY

WHAT IS CLOUDBEES JENKINS OPERATIONS CENTER?

- Operations console for Jenkins
  - Consolidated navigation
  - Shared agents
  - Centralized authentication and authorization
  - License management
  - Security enforcement
  - Analytics
  - Cluster Operations

CLOUDBEES UNIVERSITY

WHAT IS CLOUDBEES JENKINS OPERATIONS CENTER?

- Without CJOC, scaling with Jenkins can be painful.
  - Two ways to scale today:
    - Vertical
    - Horizontal
- CJOC provides alternative scaling model to avoid the pitfalls of a single point of failure and master sprawl.

CLOUDBEES UNIVERSITY

WHAT IS CLOUDBEES JENKINS OPERATIONS CENTER?

- CJOC treats agents and Jenkins masters like Jenkins jobs.
- Maintenance tasks, like plugin updates, can also be run as Jenkins jobs using a CJOC-only job type called “Cluster Operations”.
- CJOC tracks agent and master usages in a report format.

CLOUDBEES JENKINS OPERATIONS CENTER OVERVIEW

Scale

- Easily configured client masters
- Shared build resources
- Shared clouds

Secure

- Role-based Access Control (RBAC)
- Integration with LDAP and ActiveDirectory
- Single-Sign On
- Setting enforcement
- Plugin version enforcement

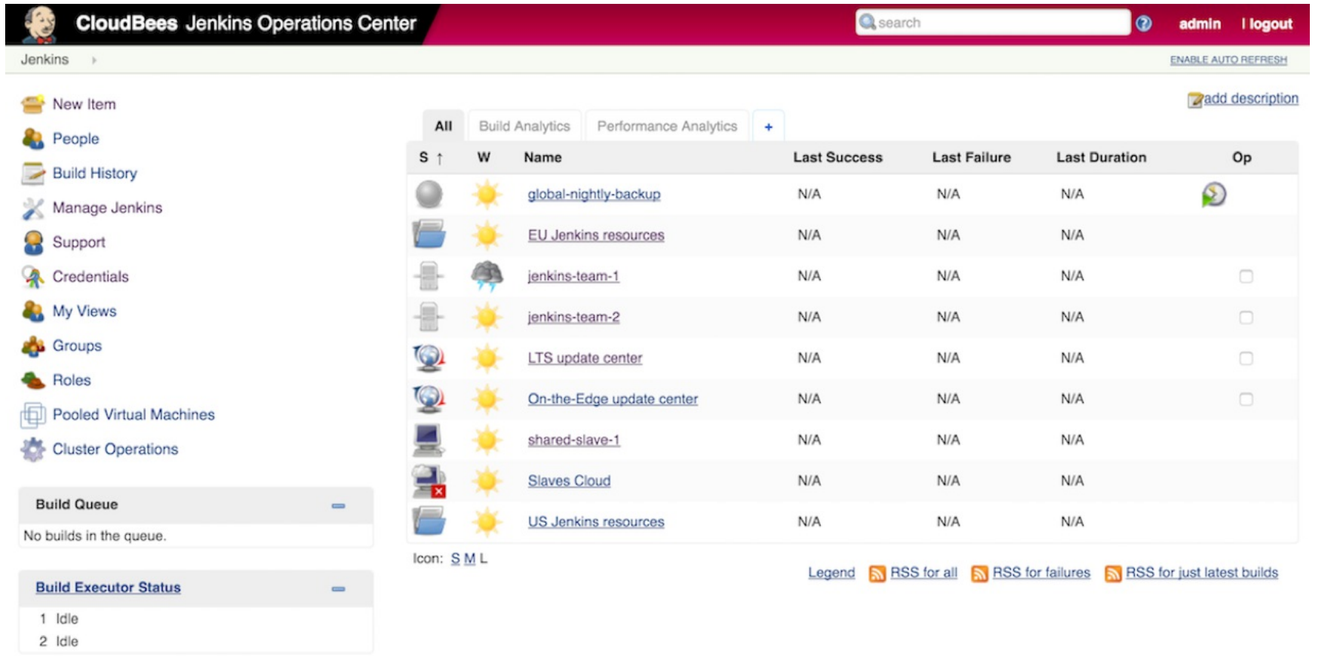
Manage

- License management
- CloudBees Jenkins Analytics
- Cluster Operations
- Test instance management

CLOUDBEES UNIVERSITY

CONCEPTS

- Operations Center server:



CLOUDBEES UNIVERSITY

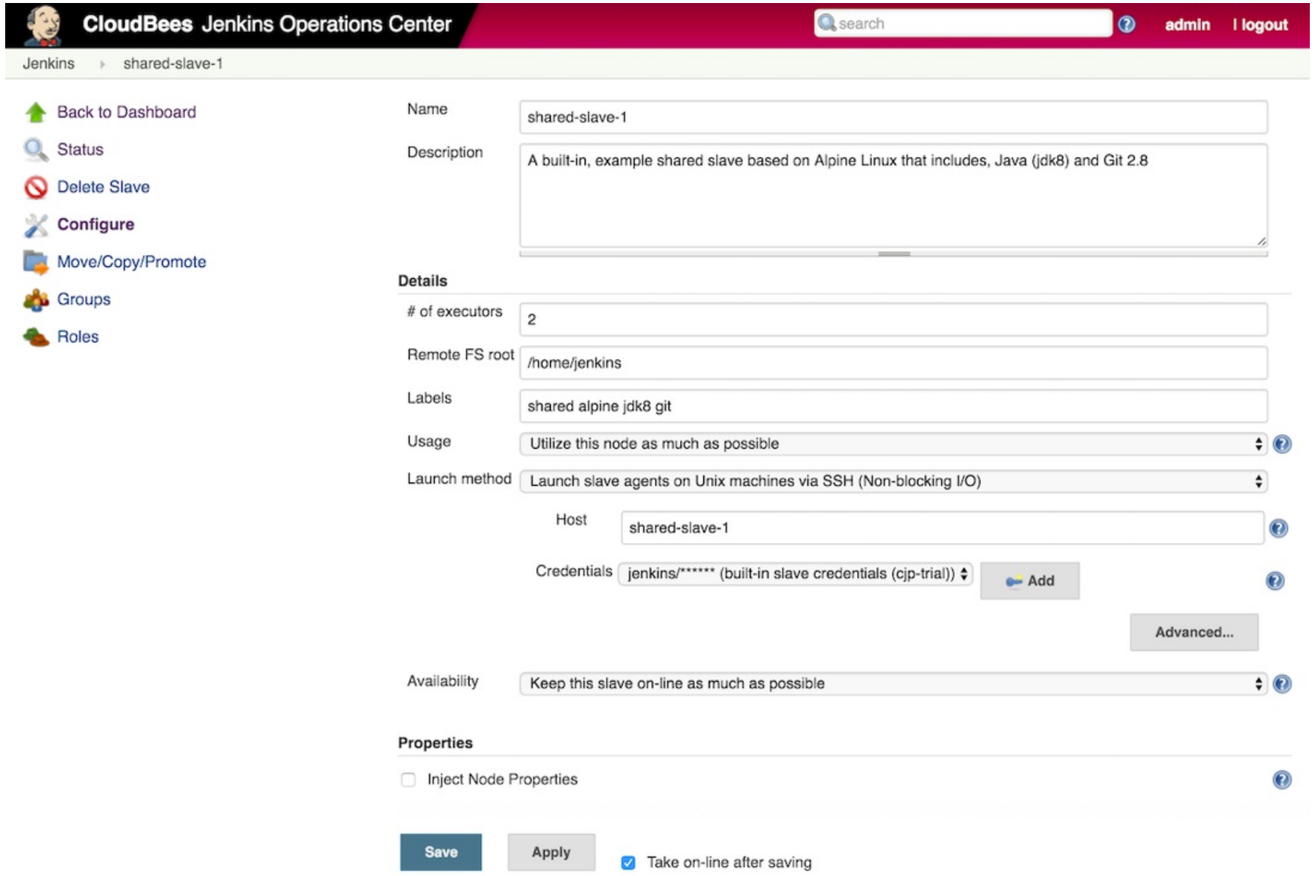
CONCEPTS

- Client master:

CLOUDBEES UNIVERSITY

CONCEPTS

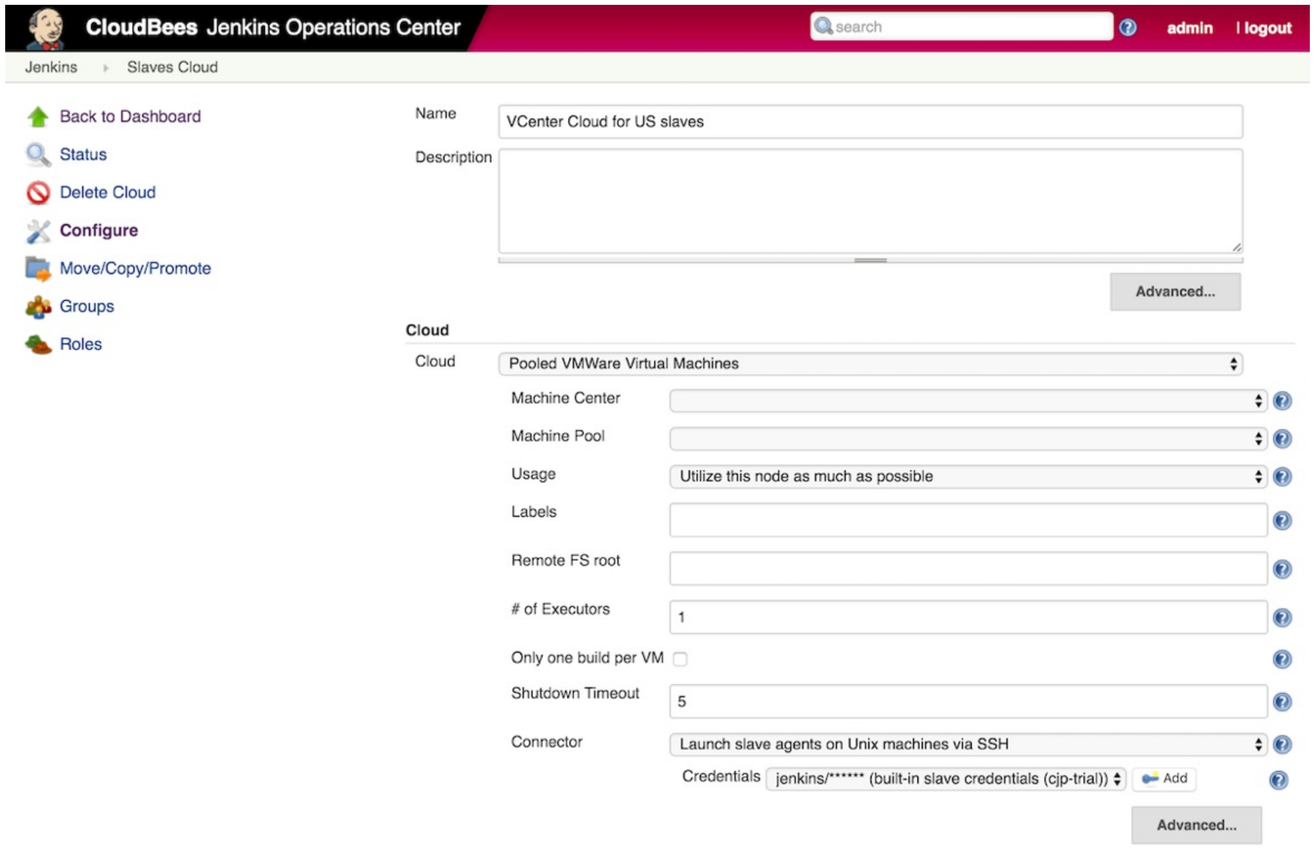
- Shared agent:



CLOUDBEES UNIVERSITY

CONCEPTS

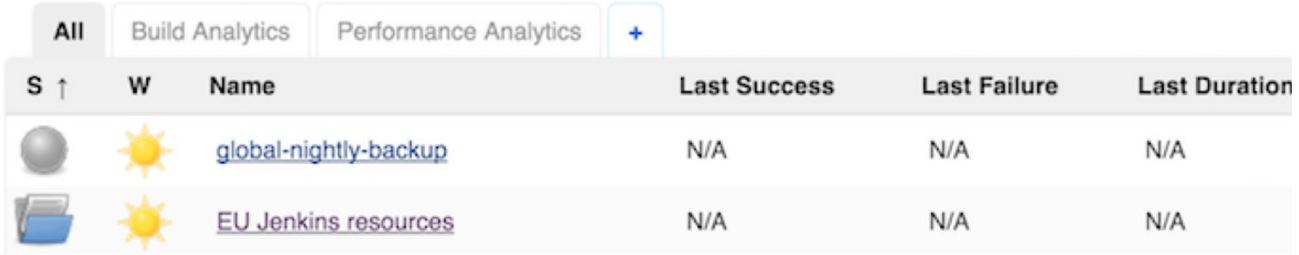
- Shared cloud:



CLOUDBEES UNIVERSITY

CONCEPTS

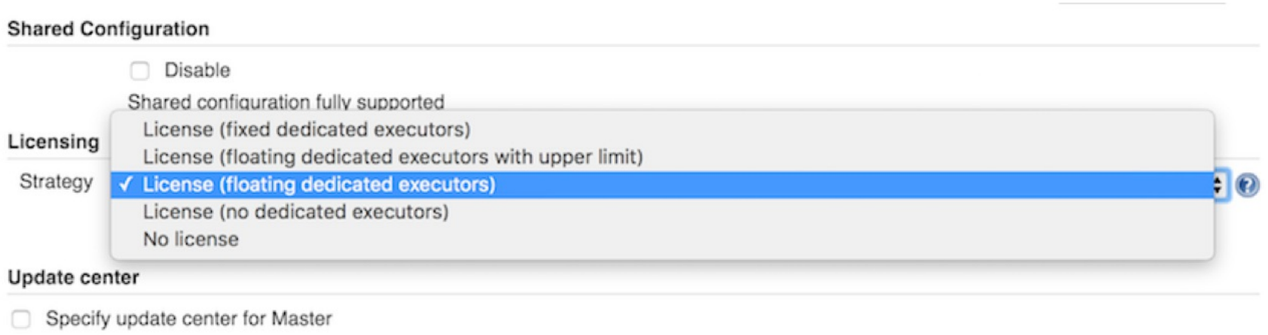
- Folders:



CLOUDBEES UNIVERSITY

CONCEPTS

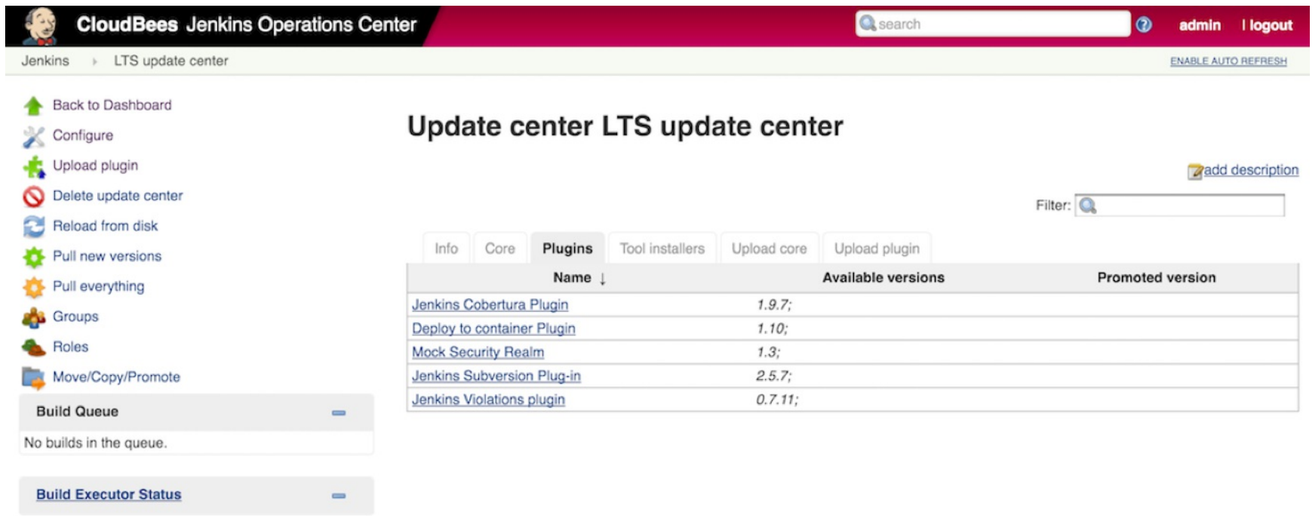
- Sub-licensing:



CLOUDBEES UNIVERSITY

CONCEPTS

- Update Center:



CLOUDBEES UNIVERSITY

BENEFIT: MAINTAINABILITY

- While some organizations seek to ease the pain of maintenance by offloading Jenkins to a managed service like CloudBees' DEV@cloud or PSE, CJOC offers an alternative solution.
- CJOC's Cluster Operations can automate maintenance tasks, like plugin and core updates, rollbacks, and master back ups.
  - Cluster Operations can be scheduled, triggered, etc. like regular Jenkins jobs.

CLOUDBEES UNIVERSITY

BENEFIT: AUDIT TRAIL

- Track:
  - who performed an update or plugin install,
  - which master or update center was affected,
  - and what the outcome was.
- CJOC's Analytics allows metrics on master/agent usage to be tracked and stored in an Elasticsearch instance.

CLOUDBEES UNIVERSITY

BENEFIT: SCALABILITY

- CJOC allows masters to be onboarded quickly, with new masters receiving pre-configured security authorization and authentication settings as well as the ability to “lease” agents from a shared pool.
- Shared agents allow build resources to be used more efficiently between Jenkins masters.

CLOUDBEES UNIVERSITY

BENEFIT: STABILITY

- Test masters for production instances can be set up and managed used CJOC.
  - Test instances are the best way to test for plugin/core conflicts before running updates on production masters
- CJOC and client masters are compatible with CloudBees' High Availability plugin.

CLOUDBEES UNIVERSITY

WHERE DO WE START?

- CJOC will need to be installed on a server
  - Run as .war, RPM, etc
  - At least 1 GB for CJOC's JVM