CLOUDBEES UNIVERSITY

CLOUDBEES JENKINS PLATFORM

Pipeline with Docker

CLOUDBEES UNIVERSITY

TOC

CLOUDBEES UNIVERSITY

PREREQUISITES

CLOUDBEES UNIVERSITY

PREREQUISITES

- If you do not already have an SSH client, please install Secure Shell Chrome Extension
- Confirm that you can login to the assigned server (IP has been sent by email). Use cb as both the user and the password.

CLOUDBEES UNIVERSITY

JENKINS OVERVIEW

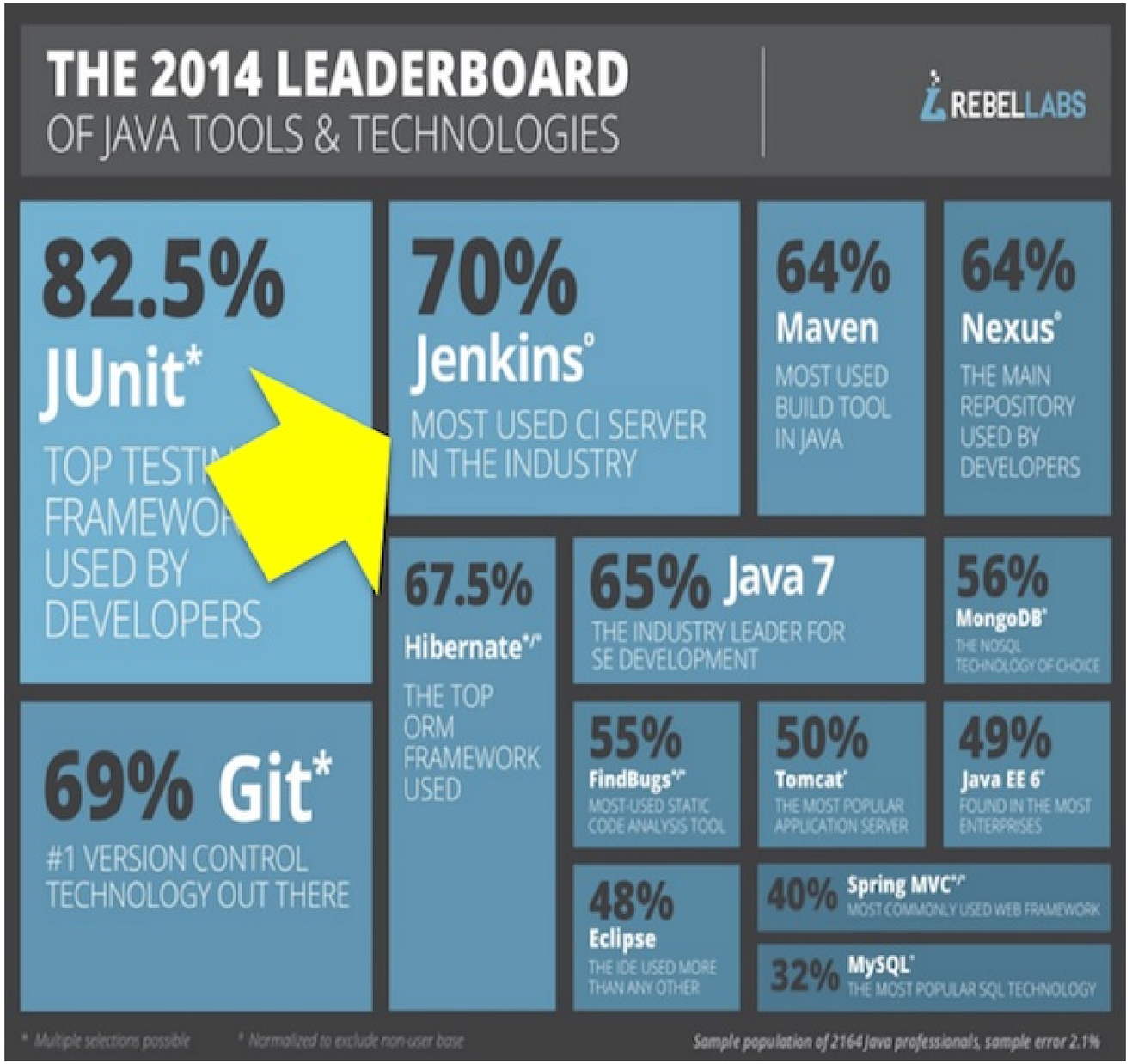
CLOUDBEES UNIVERSITY

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

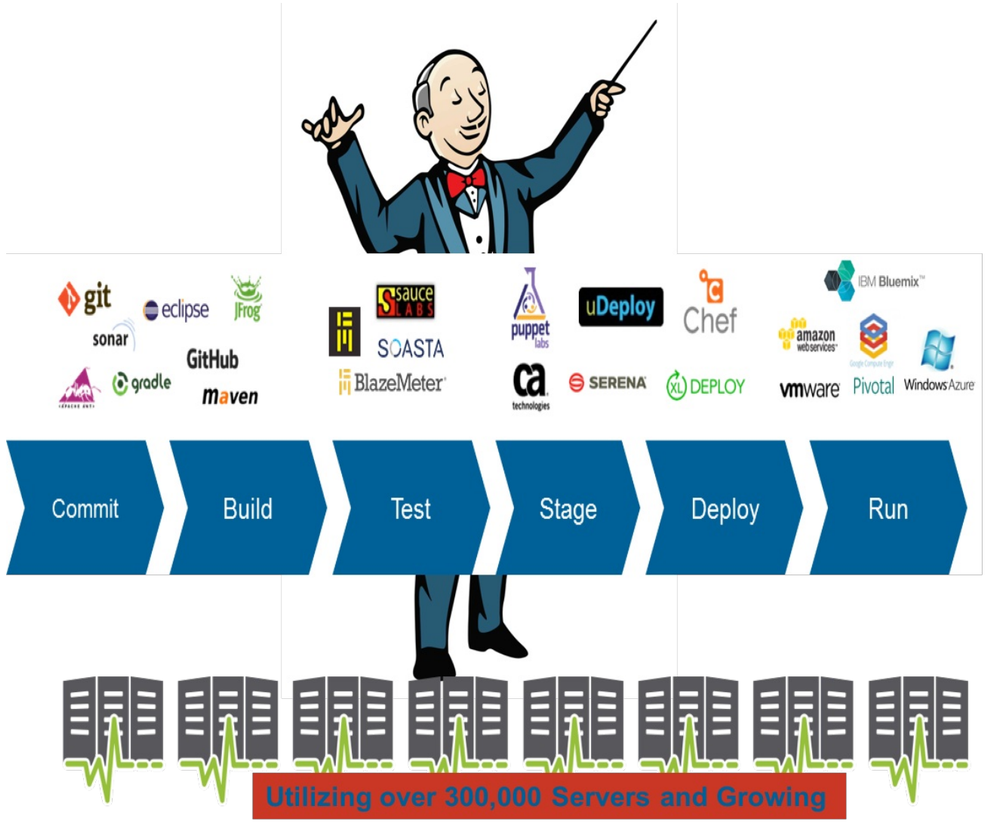
Source: 2014 Java Tools and Technologies Landscape – Rebel Labs

CLOUDBEES UNIVERSITY



CLOUDBEES UNIVERSITY

JENKINS IS THE CD ORCHESTRATOR



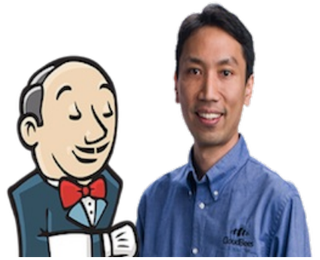
CLOUDBEES UNIVERSITY

CLOUDBEES INTRODUCTION

CLOUDBEES UNIVERSITY

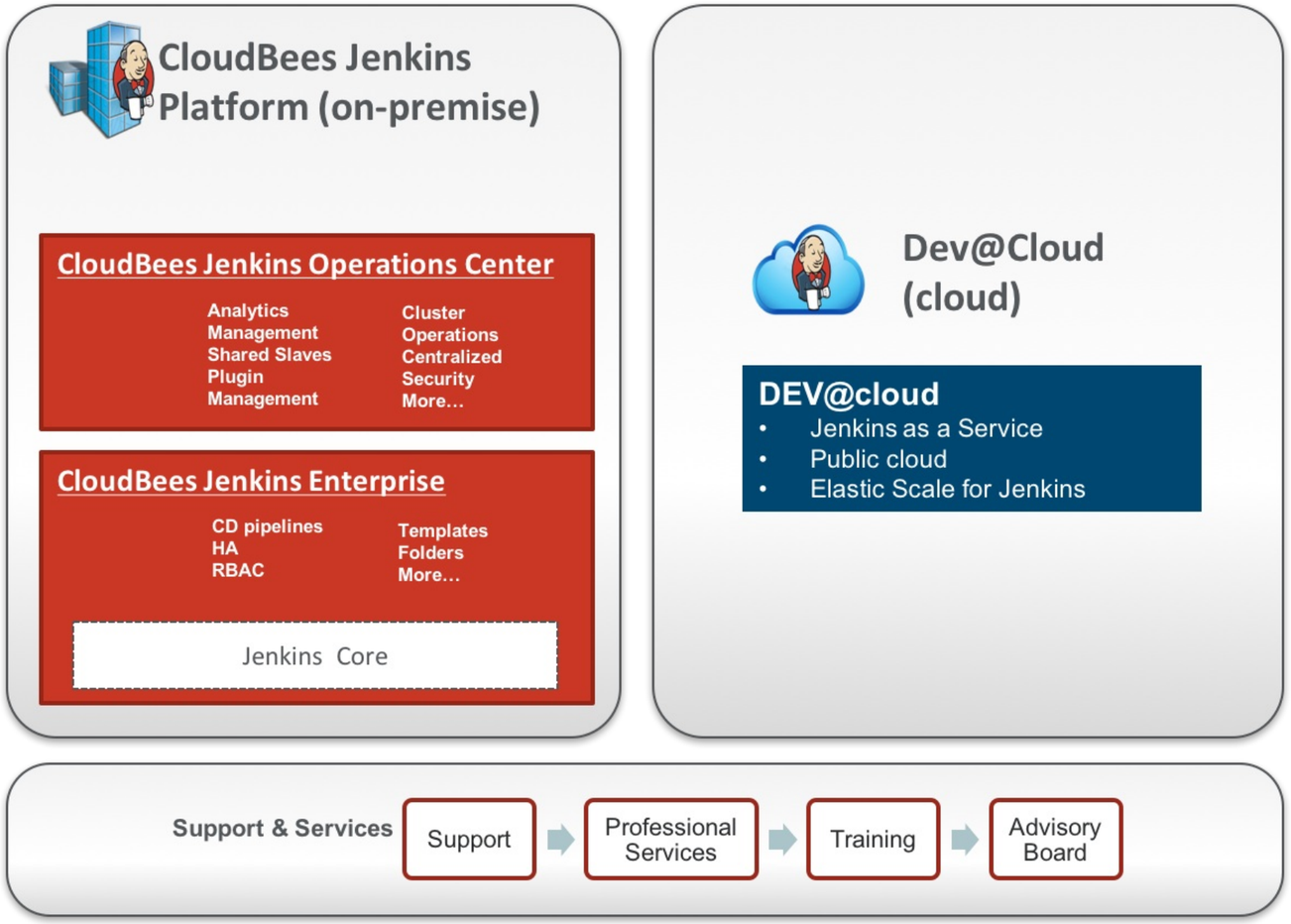
CLOUDBEES AND THE JENKINS COMMUNITY

- 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
- Produce Jenkins Quarterly Newsletter
- Conduct Jenkins User Conferences



CLOUDBEES UNIVERSITY

CLOUDBEES PRODUCTS



CLOUDBEES UNIVERSITY

CLOUDBEES JENKINS PLATFORM: ENTERPRISE EDITION

For enterprise DevOps or infrastructure teams that want to:

- Have the best in class Jenkins technical support
- Operate Jenkins at scale to support a large number of developers, teams, or projects
- Monitor usage and share resources across the Jenkins infrastructure
- Optimize performance and efficiency of the CD platform

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
- Partners
  - Training
  - Integration
  - Custom Development

CLOUDBEES UNIVERSITY

PIPELINE INTRODUCTION (DAY 1)

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL LEARN

- The need for pipeline
- Pipeline use cases
- Key pipeline DSL
- Pipeline structure and syntax
- Execution control

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL BE ABLE TO

- Recognize pipeline purpose and use cases

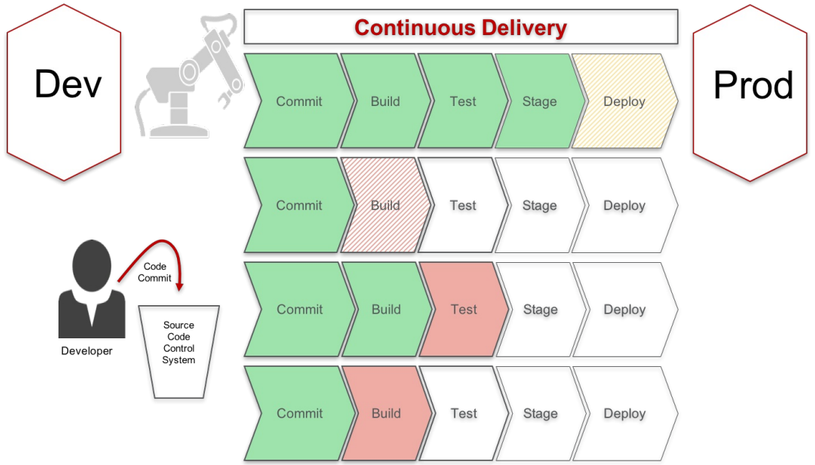
CLOUDBEES UNIVERSITY

THE NEED FOR PIPELINE?

- Reduction of a number of jobs
- Easier maintenance
- Deployment decentralization
- Easier specification through code

CLOUDBEES UNIVERSITY

WHAT IS CD FLOW?



CD is about setting a "flow", from SCM commit to deployed application

Jenkins can chain jobs this way, but …

CLOUDBEES UNIVERSITY

JENKINS CD FLOW

A Real-world CD Flow 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

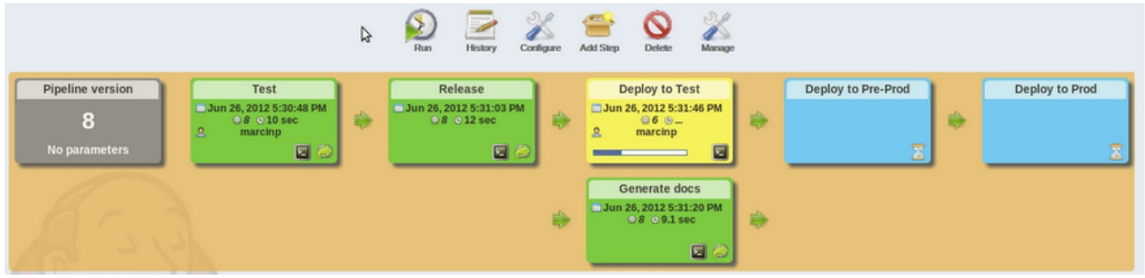
JENKINS CD FLOW

- Many Related Plugins
  - Conditional build steps
  - Parameterized trigger
  - Promotions
  - …
- Major Issue: Flow Configuration Is Scattered In Various Jobs

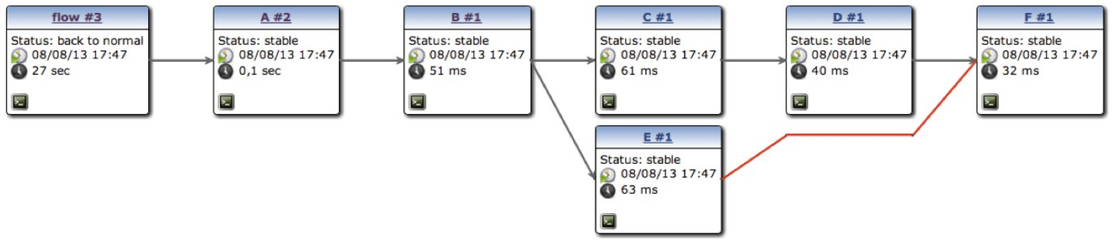
CLOUDBEES UNIVERSITY

CD FLOW VISUALIZATION

Build Pipeline Plugin To Render “Linear” Pipeline



Build-graph-view To Render Complex Ones



CLOUDBEES UNIVERSITY

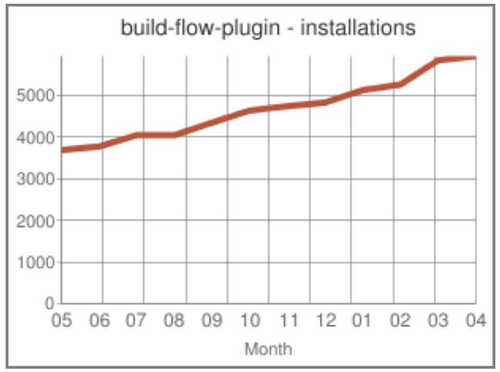
REALITY CHECK

- How do you use Jenkins now?
- Given the functionality, where do you see your team using these tools and plugins in your jobs?
- Do you practice CI or CD?
- Describe your CI/CD flow needs and processes.

CLOUDBEES UNIVERSITY

BUILD FLOW PLUGIN

- OSS
- Proof of concept
- Define CD flow as code
- Largely adopted
- Technical limitations



CLOUDBEES UNIVERSITY

CLOUDBEES PIPELINE PLUGIN

- Inspired By Build-flow
- Groovy DSL To Orchestrate Build Steps
- Extensible DSL Syntax
- Supports Checkpoint
- Advanced Visualization
- SCM Friendly

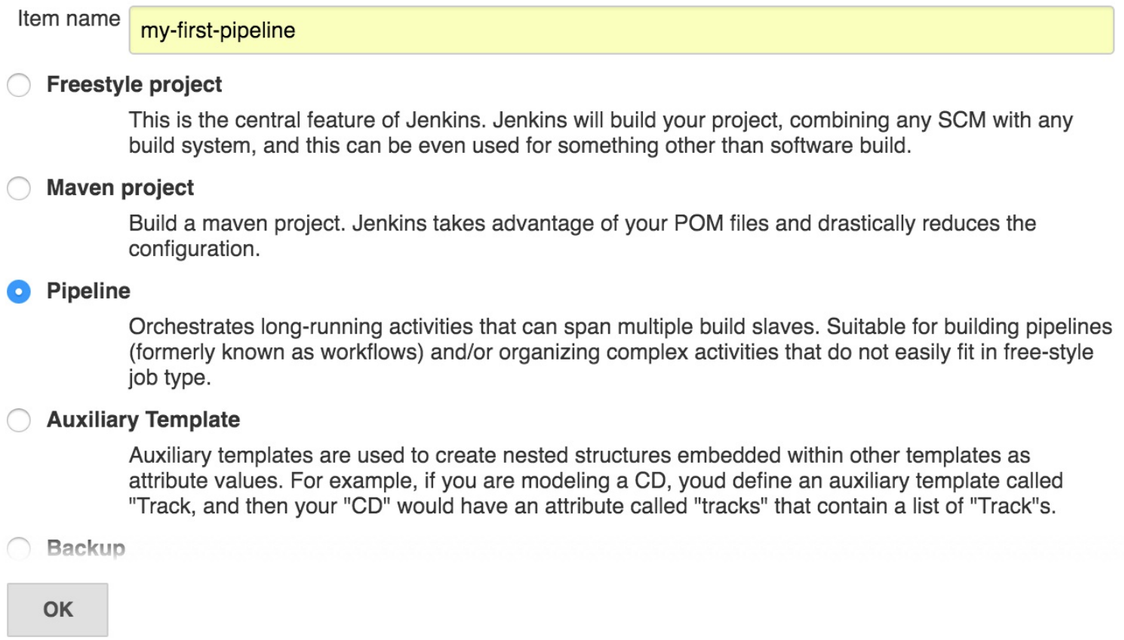
CLOUDBEES UNIVERSITY

MID-BREAK

(10) minutes for learner re-integration.

CLOUDBEES UNIVERSITY

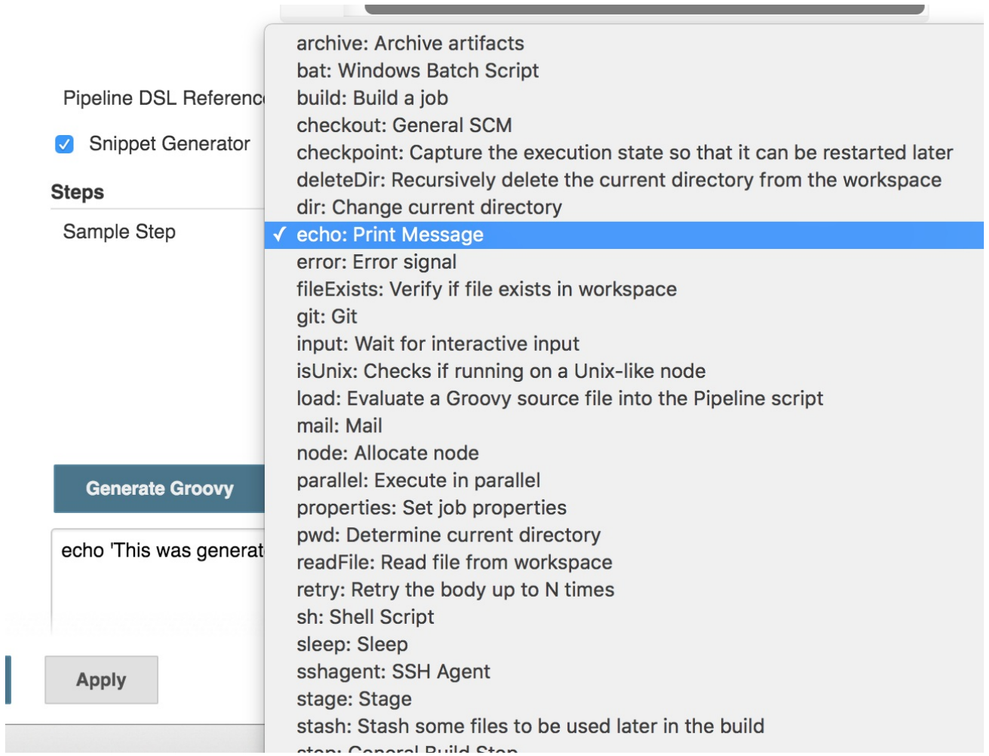
CREATING A SIMPLE PIPELINE



More info: Getting Started With Pipeline

CLOUDBEES UNIVERSITY

SNIPPET GENERATOR



CLOUDBEES UNIVERSITY

PIPELINE INTRODUCTION: REVIEW

CLOUDBEES UNIVERSITY

PIPELINE INTRODUCTION: REVIEW

- CD Flow
- Older Solutions
- Build Flow Plugin
- Cloudbees Pipeline Plugin
- Create Pipeline Jobs
- Use The Snippet Generator

CLOUDBEES UNIVERSITY

PIPELINE INTRODUCTION: EXERCISE

Pipeline Introduction: Exercise

CLOUDBEES UNIVERSITY

DOCKER INTRODUCTION (DAY 2)

CLOUDBEES UNIVERSITY

REVIEW OF DAY 1 CONCEPTS AND EXERCISE

- CD flow
- Older solutions
- Build Flow plugin
- CloudBees Pipeline Plugin
- Create Pipeline jobs & use the Snippet Generator
- Exercise

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL LEARN

- How to prepare the environments
- Docker containers
- Docker use cases
- Docker tools

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL BE ABLE TO

- Prepare the CD flow environments
- Understand Docker containers and use cases
- Have A Working Knowledge Of Docker Tools

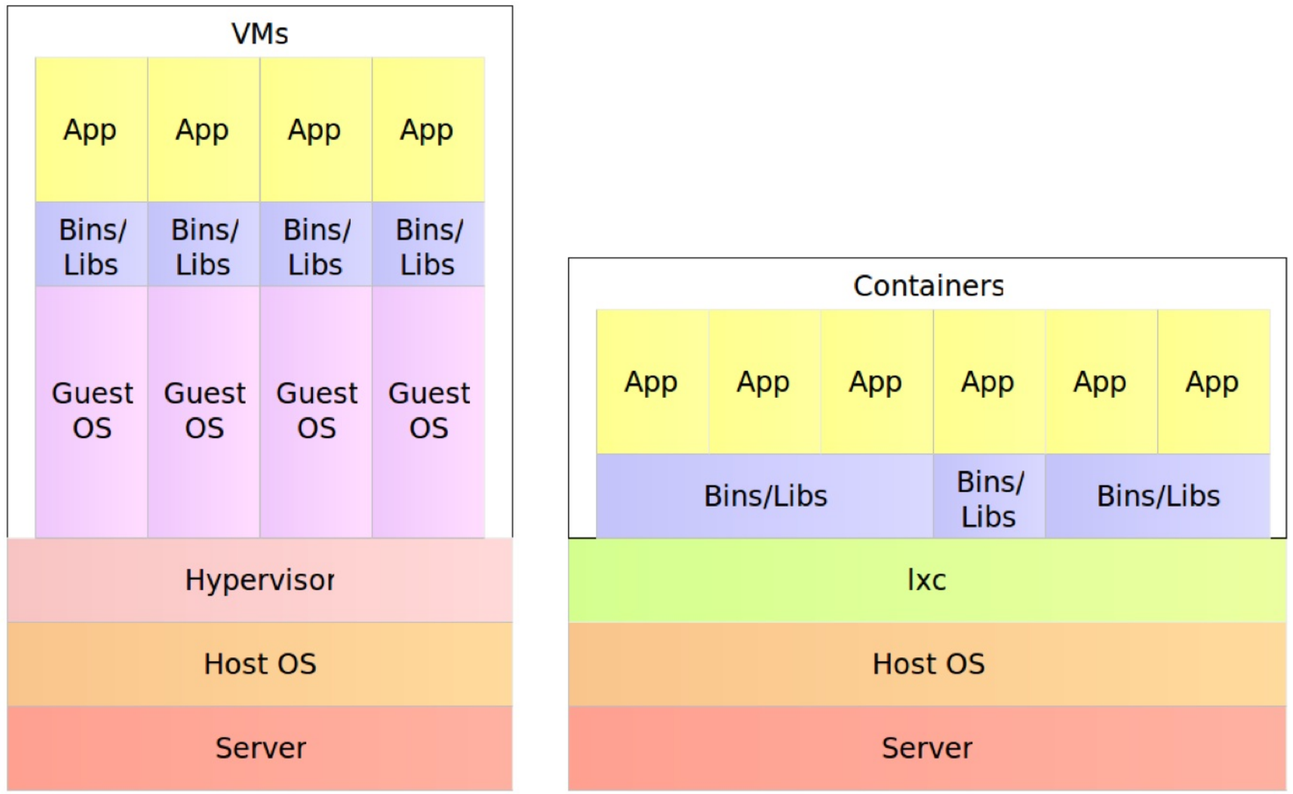
CLOUDBEES UNIVERSITY

DOCKER CONTAINERS

- Self-sufficient
- Isolated
- Immutable
- Reliable
- Scalable

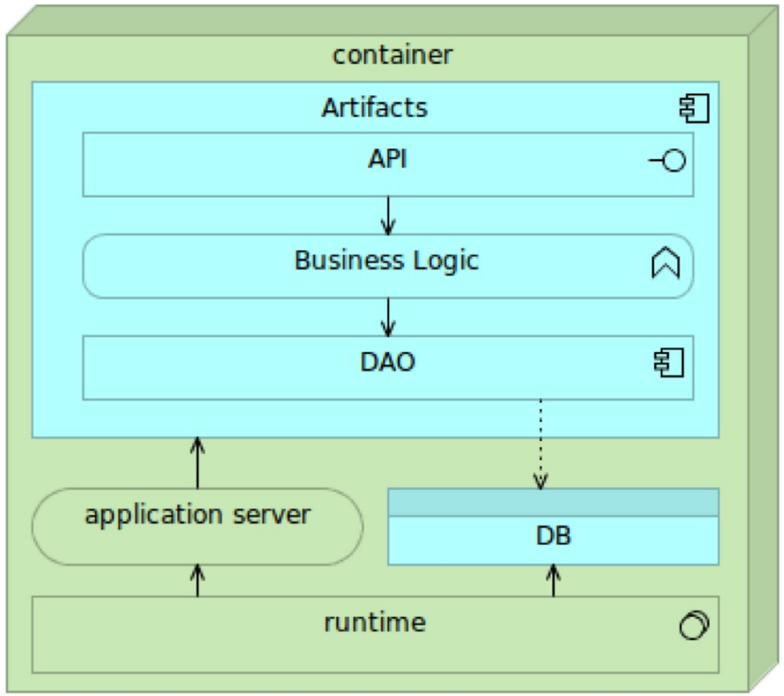
CLOUDBEES UNIVERSITY

VMS VS DOCKER CONTAINERS



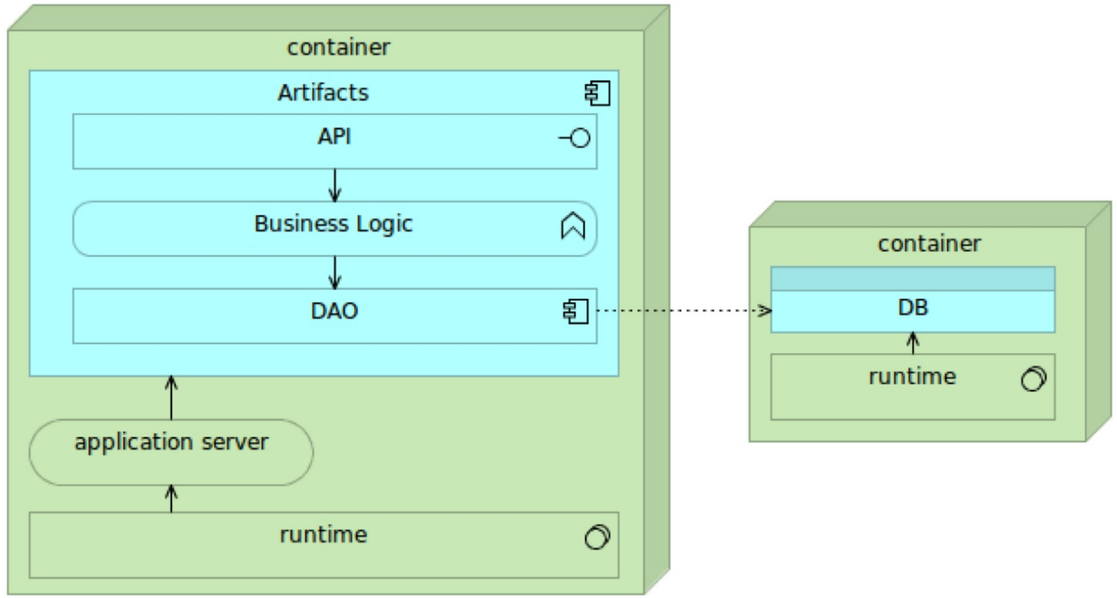
CLOUDBEES UNIVERSITY

SINGLE-CONTAINER SERVICE



CLOUDBEES UNIVERSITY

LINKED OR NETWORKED CONTAINERS



CLOUDBEES UNIVERSITY

DOCKER USE CASES

- Local Development And Testing
- Continuous Integration, Delivery, Or Deployment
- Reliable And Repeatable Deployments

CLOUDBEES UNIVERSITY

REALITY CHECK

- Questions About Containers?
- Examples Of Scalability In Your Current Use Of Containers?
- How Is CI Valuable In Your Environments?
- Are You Using Containers In Deployment Now?

CLOUDBEES UNIVERSITY

DOCKER TOOLS

- Docker Hub/Registry
- Docker Engine
- Docker Compose
- Docker Swarm
- Docker Machine
- Kitematic

CLOUDBEES UNIVERSITY

DOCKER HUB AND REGISTRY

- Docker Hub: Cloud hosted service that provides registry capabilities for public and private content.
- Docker Registry: An open source application dedicated to the storage and distribution of your Docker images.
- Docker Trusted Registry: Allows storing and managing Docker images on-premise. It supports security or regulatory compliance requirements.

CLOUDBEES UNIVERSITY

DOCKER HUB AND REGISTRY: PULLING AND PUSHING CONTAINERS

Please use the SSH client to connect to the server you were assigned.

```
docker pull cloudbees/training-books-ms-tests

docker tag cloudbees/training-books-ms-tests \
    localhost:5000/training-books-ms-tests

docker push localhost:5000/training-books-ms-tests
```

CLOUDBEES UNIVERSITY

DOCKER ENGINE

- Docker Engine is a lightweight runtime and robust tooling that builds and runs Docker containers.

CLOUDBEES UNIVERSITY

DOCKER ENGINE: RUNNING TESTS INSIDE CONTAINERS

```
cd /mnt/training-books-ms

docker run -it --rm \
    -v $PWD/client/components:/source/client/components \
    -v $PWD/client/test:/source/client/test \
    -v $PWD/src:/source/src \
    -v $PWD/target/scala-2.10:/source/target/scala-2.10 \
    --env TEST_TYPE=all \
    localhost:5000/training-books-ms-tests

ll target/scala-2.10/

docker ps -a | grep books
```

CLOUDBEES UNIVERSITY

DOCKER ENGINE: BUILDING CONTAINERS AND PUSHING THEM TO THE PRIVATE REGISTRY

```
cat Dockerfile

docker build -t localhost:5000/training-books-ms .

docker push localhost:5000/training-books-ms
```

CLOUDBEES UNIVERSITY

MID-BREAK

(10) minutes for learner re-integration.

CLOUDBEES UNIVERSITY

DOCKER ENGINE: RUNNING CONTAINERS

```
docker run -d --name books-ms-db mongo

docker run -d --name books-ms \
    -p 1234:8080 \
    --link books-ms-db:db \
    localhost:5000/training-books-ms

docker exec -it books-ms env | grep DB

docker ps

curl -I localhost:1234/api/v1/books
```

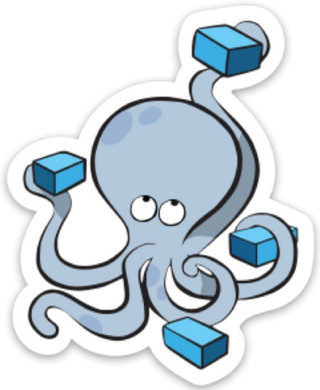
CLOUDBEES UNIVERSITY

DOCKER ENGINE: LOGGING, STOPPING AND REMOVING CONTAINERS

```
docker logs books-ms

docker logs books-ms-db

docker stop books-ms-db books-ms

docker ps -a

docker rm books-ms-db books-ms

docker ps -a
```

CLOUDBEES UNIVERSITY

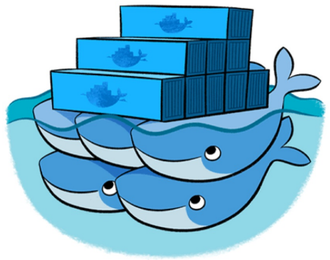
DOCKER COMPOSE



- Docker Compose allows defining multi-container application with all of its dependencies in a single file, then spin your application up in a single command.

CLOUDBEES UNIVERSITY

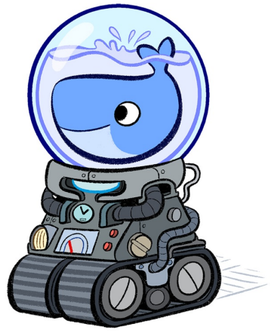
DOCKER SWARM



- Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine. With these pooled resources, you can scale out your application as if it were running on a single, huge computer.

CLOUDBEES UNIVERSITY

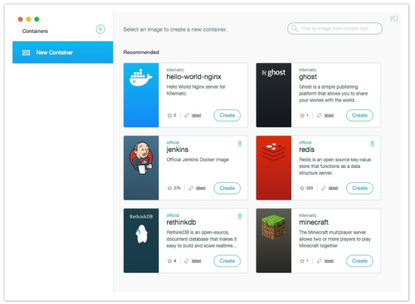
DOCKER MACHINE



- Docker Machine automatically sets up Docker on your computer, on cloud providers, and inside your data center. Docker Machine provisions the hosts, installs Docker Engine on them, and then configures the Docker client to talk to the Docker Engines.

CLOUDBEES UNIVERSITY

KITEMATIC



- Kitematic is a completely automated process that installs and configures the Docker environment on your machine. Build and run containers through a simple, yet powerful graphical user interface (GUI).

CLOUDBEES UNIVERSITY

DOCKER INTRODUCTION: REVIEW

CLOUDBEES UNIVERSITY

DOCKER INTRODUCTION: REVIEW

- Docker benefits & advantages
- Docker use cases
- Docker Hub and Registry, Engine, Compose, Swarm, Machine, Kitematic

CLOUDBEES UNIVERSITY

DOCKER INTRODUCTION: EXERCISE

Docker Introduction: Exercise

CLOUDBEES UNIVERSITY

THE PROJECT - PART 1 (DAY 3)

CLOUDBEES UNIVERSITY

REVIEW OF DAY 2 CONCEPTS AND EXERCISE

- Docker benefits & advantages
- Docker use cases
- Docker Hub and Registry, Engine, Compose, Swarm, Machine, Kitematic

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL LEARN

- How to combine CloudBees Pipeline plugin with Docker
- How to implement most commonly used steps required for CI/CD flow

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL BE ABLE TO

- Create deployment lifecycle with Jenkins Pipeline and Docker

CLOUDBEES UNIVERSITY

THE PROJECT

- Run pre-deployment tests inside a Docker container
- Build artefacts
- Build and push the service container
- Request manual permission to deploy the service container to production

CLOUDBEES UNIVERSITY

THE PROJECT (CONT.)

- Pull the service container and all dependent containers to production
- Run the service container and all dependent containers in production
- Run post-deployment tests (integration tests) inside a Docker container

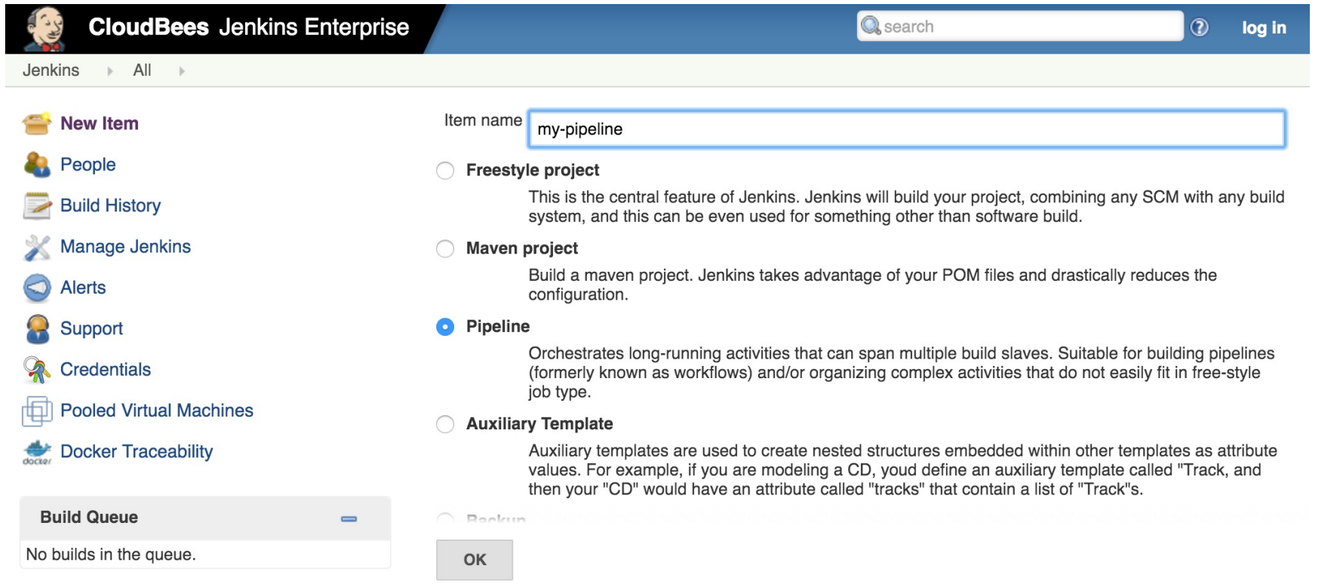
CLOUDBEES UNIVERSITY

REALITY CHECK

- Questions on the preparation and the project?
- Does this workflow differ from your practices?
- Using containers now?

CLOUDBEES UNIVERSITY

CREATE A PIPELINE JOB CALLED MY-PIPELINE



CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – NODE: TASK

Specify the node cd, run the pipeline, and confirm that it is running inside the cd node by looking at logs.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – NODE: SOLUTION

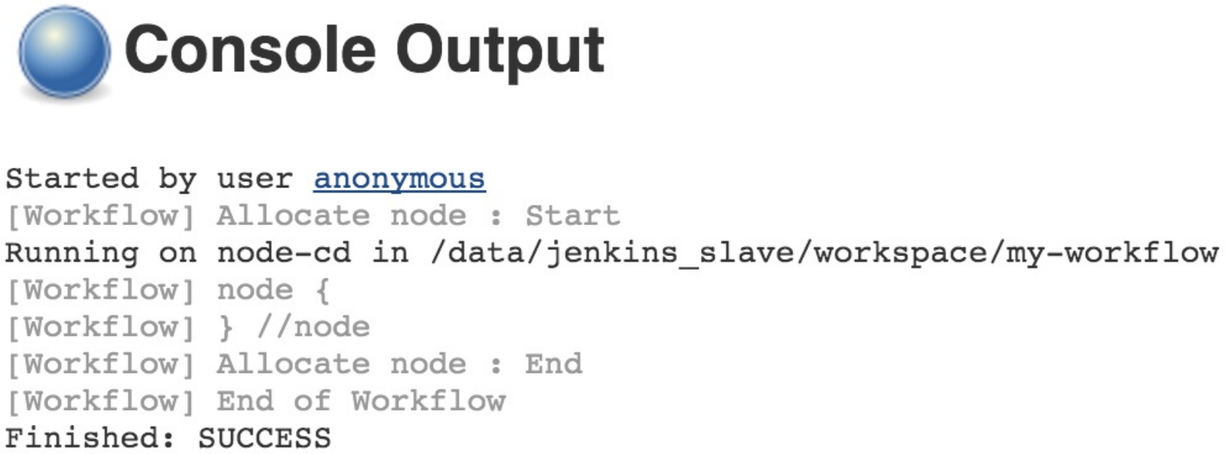
```
node("cd") {
}
```

CLOUDBEES UNIVERSITY

RUN THE JOB

Open http://<IP>:8080/job/my-pipeline/build?delay=0sec

Open http://<IP>:8080/job/my-pipeline/lastBuild/console



CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – GIT: TASK

Clone the code from the repository https://github.com/cloudbees/training-books-ms.git

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – GIT: SOLUTION

```
    git "https://github.com/cloudbees/training-books-ms.git"
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – VARIABLES, PWD AND SH: TASK

Assign the current job workspace directory to the dir variable, create directory db inside the workspace, and assign full permissions to all users.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – VARIABLES, PWD AND SH: SOLUTION

```
def dir = pwd()
sh "mkdir -p ${dir}/db"
sh "chmod 0777 ${dir}/db"
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – STAGE: TASK

Create the pre-deployment tests stage.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – STAGE: SOLUTION

```
    stage "pre-deployment tests"
```

CLOUDBEES UNIVERSITY

MID-BREAK

(10) minutes for learner re-integration.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – DOCKER: TASK

Pull the Docker image localhost:5000/books-ms-tests and run the run_tests.sh script inside the container. Host volume db should be mounted as /data/db inside the container.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – DOCKER: SOLUTION

```
def tests = docker.image("localhost:5000/training-books-ms-tests")
tests.pull()
tests.inside("-v ${dir}/db:/data/db") {
    sh "./run_tests.sh"
}
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – DOCKER: TASK

Build the Docker image localhost:5000/books-ms and push the container to the private registry.
Use the stage build for these steps.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – DOCKER: SOLUTION

```
    stage "build"
    def service = docker.build "localhost:5000/training-books-ms"
    service.push()
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – STASH AND UNSTASH: TASK

Pull containers and run (through Docker Compose target app) the localhost:5000/books-ms container in the production node. Use stash to archive docker-compose-dev.yml file while in the cd node and unstash to retrieve it when inside the production node. Before running the service, make sure that both the service and mongo containers are pulled.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – STASH AND UNSTASH: SOLUTION

```
node("cd") {
...
    stash includes: "docker-compose*.yml", name: "docker-compose"
}
node("production") {
    stage "deploy"
    unstash "docker-compose"
    docker.image("localhost:5000/training-books-ms").pull()
    docker.image("mongo").pull()
    sh "docker-compose -p books-ms up -d app"
}
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – ENV AND WITHENV: TASK

Run post-deployment tests in the node cd. The run_tests.sh script expects two environment variables: TEST_TYPE=integ and DOMAIN=<IP>:8081. Tests should be run inside the training-books-ms container.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – ENV AND WITHENV: SOLUTION

```
node("cd") {
    stage "post-deployment tests"
    def tests = docker.image("localhost:5000/training-books-ms-tests")
    tests.inside() {
        withEnv(["TEST_TYPE=integ", "DOMAIN=http://[IP]:8081"]) {
            sh "./run_tests.sh"
        }
    }
}
```

CLOUDBEES UNIVERSITY

THE PROJECT - PART 1: REVIEW

CLOUDBEES UNIVERSITY

THE PROJECT - PART 1: REVIEW

- Defined the project steps
- Created a new Pipeline job
- Defined steps that clone the code
- Defined steps that run pre-deployment tests
- Defined steps that build and push the container
- Defined steps that deploy the container
- Defined steps that run post-deployment tests

CLOUDBEES UNIVERSITY

THE PROJECT - PART 1: EXERCISE

The Project - Part 1: Exercise

CLOUDBEES UNIVERSITY

THE PROJECT - PART 2 (DAY 4)

CLOUDBEES UNIVERSITY

REVIEW OF DAY 3

- Defined the project steps
- Created a new Pipeline job
- Defined steps that clone the code
- Defined steps that run pre-deployment tests
- Defined steps that build and push the container
- Defined steps that deploy the container
- Defined steps that run post-deployment tests

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL LEARN

- How to use more advanced features of the CloudBees Pipeline plugin and Docker

CLOUDBEES UNIVERSITY

IN THIS UNIT: YOU WILL BE ABLE TO

- Create advanced deployment lifecycle with Jenkins Pipeline and Docker

CLOUDBEES UNIVERSITY

THE PROJECT (CONT.)

- Request manual confirmation before deployment
- Run steps in parallel
- Control the steps execution
- Add checkpoints on critical or after long running processes
- Visualize the deployment pipeline
- Convert the job into a template
- Leverage multi branch pipeline and Jenkinsfile features

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – INPUT: TASK

- Add an input with the message Please confirm deployment to production and the button with the text Submit.
- The input should be requested prior to deployment to production.
- Add the parameter for additional notes and echo the value.
- Make sure that only the user manager is allowed to use this input.
- Use user/pass manager/manager to login before approving the deployment.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – INPUT: SOLUTION

```
def response = input message: 'Please confirm deployment to production',
    ok: 'Submit',
    parameters: [[
        $class: 'StringParameterDefinition',
        defaultValue: '',
        description: 'Additional comments',
        name: ''
    ]],
    submitter: 'manager'
echo response
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – PARALLEL: TASK

Modify the script so that service and db containers are pulled in parallel.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – PARALLEL: SOLUTION

```
//    docker.image("localhost:5000/training-books-ms").pull()
//    docker.image("mongo").pull()
    def pull = [:]
    pull["service"] = {
        docker.image("localhost:5000/training-books-ms").pull()
    }
    pull["db"] = {
        docker.image("mongo").pull()
    }
    parallel pull
```

CLOUDBEES UNIVERSITY

REALITY CHECK

- Questions on the prep and the Project?
- Clear on the input and parallel DSL?
- Ready for the execution control?

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – EXECUTION CONTROL

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

sleep time: 10, unit: 'MINUTES'

waitUntil {
    // some block
}

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

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

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – EXECUTION CONTROL: TASK

Change the script so that the execution of post-deployment tests is retried in case of a failure.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – EXECUTION CONTROL: SOLUTION

```
retry(2) {
    sh "./run_tests.sh"
}
```

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – EXECUTION CONTROL: TASK

Change the script so that there is a pause of 2 seconds after the service is deployed.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – EXECUTION CONTROL: SOLUTION

```
sleep 2
```

CLOUDBEES UNIVERSITY

MID-BREAK

(10) minutes for learner re-integration.

CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – CHECKPOINT: TASK

Add checkpoint deploy between the nodes cd and production.

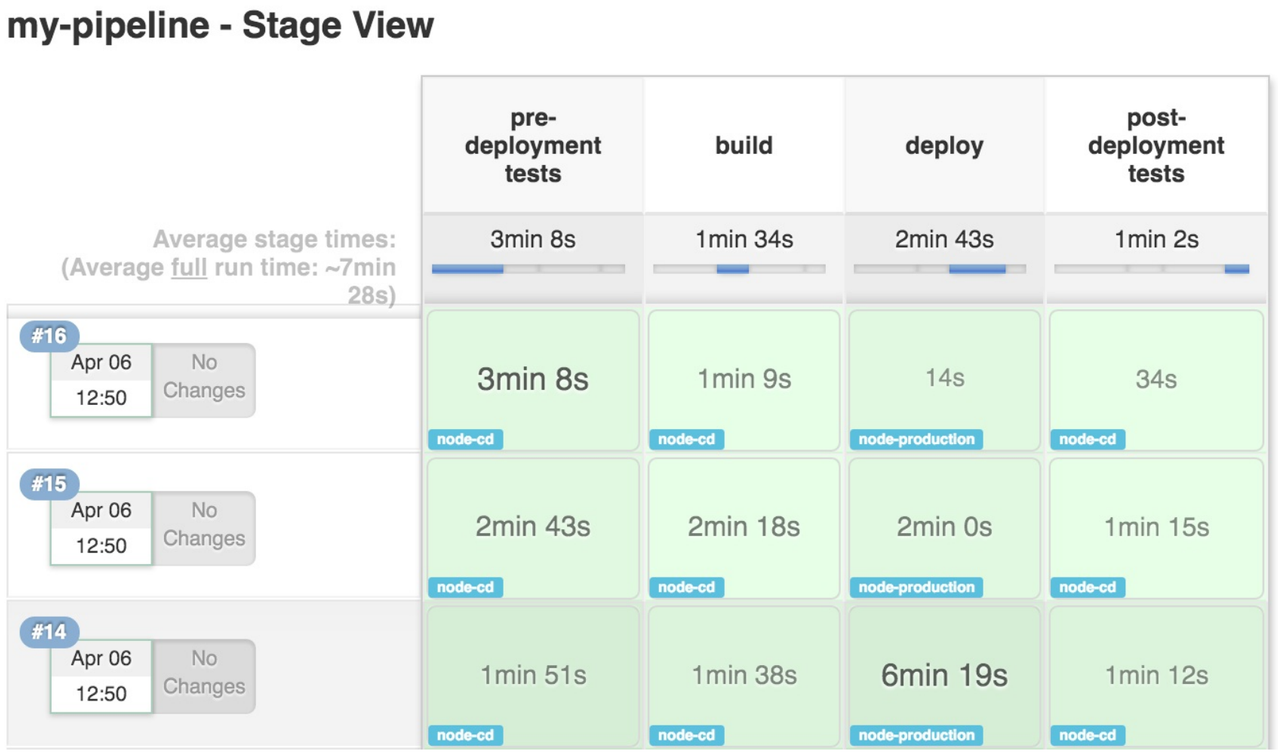
CLOUDBEES UNIVERSITY

KEY PIPELINE DSL – CHECKPOINT: SOLUTION

```
checkpoint "deploy"
```

CLOUDBEES UNIVERSITY

PIPELINE STAGE VIEW



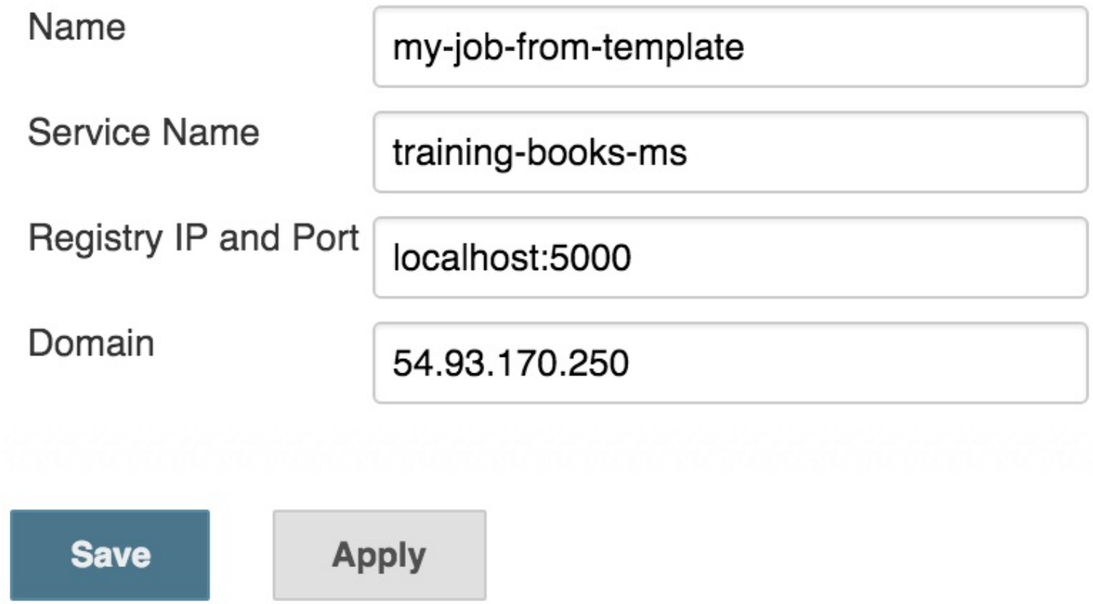
CLOUDBEES UNIVERSITY

PIPELINE AND JOB TEMPLATE: TASK

- Convert the deployment pipeline script into a template named my-template.
- Replace the service name (training-books-ms), the registry IP and port (localhost:5000) and domain (http://<IP>:8081) with variables.
- Create a new job called my-pipeline-from-template. The job type should be my-template.

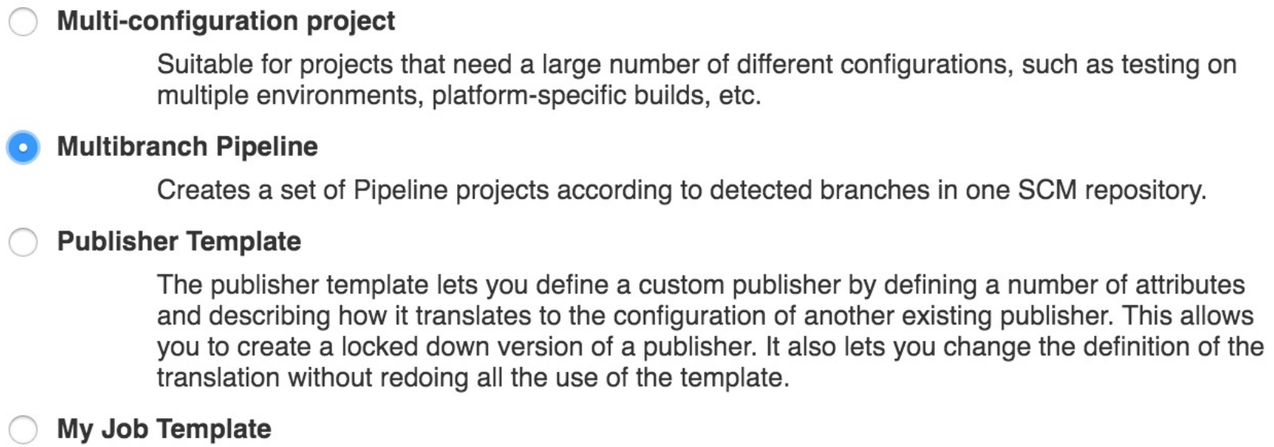
CLOUDBEES UNIVERSITY

PIPELINE AND JOB TEMPLATE: SOLUTION



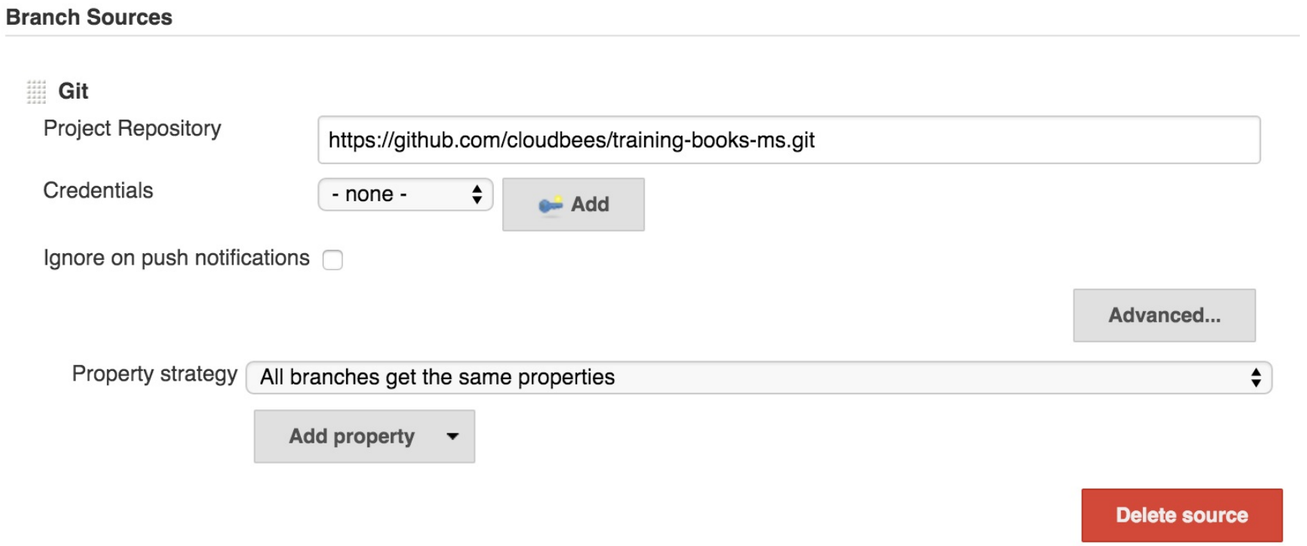
CLOUDBEES UNIVERSITY

MULTI-BRANCH PIPELINE AND JENKINSFILE



CLOUDBEES UNIVERSITY

MULTI-BRANCH PIPELINE AND JENKINSFILE



CLOUDBEES UNIVERSITY

MULTI-BRANCH PIPELINE AND JENKINSFILE

```
def serviceName = "training-books-ms"
def registry = "localhost:5000"
def flow

node("cd") {
    git "https://github.com/cloudbees/${serviceName}.git"
    flow = load "/mnt/scripts/pipeline-common.groovy"
    flow.runPreDeploymentTests(serviceName, registry)
    flow.build(serviceName, registry)
}
checkpoint "deploy"
node("cd") {
    flow.deploy(serviceName, registry)
    flow.runPostDeploymentTests(serviceName, registry, "http://<IP>:8081")
}
```

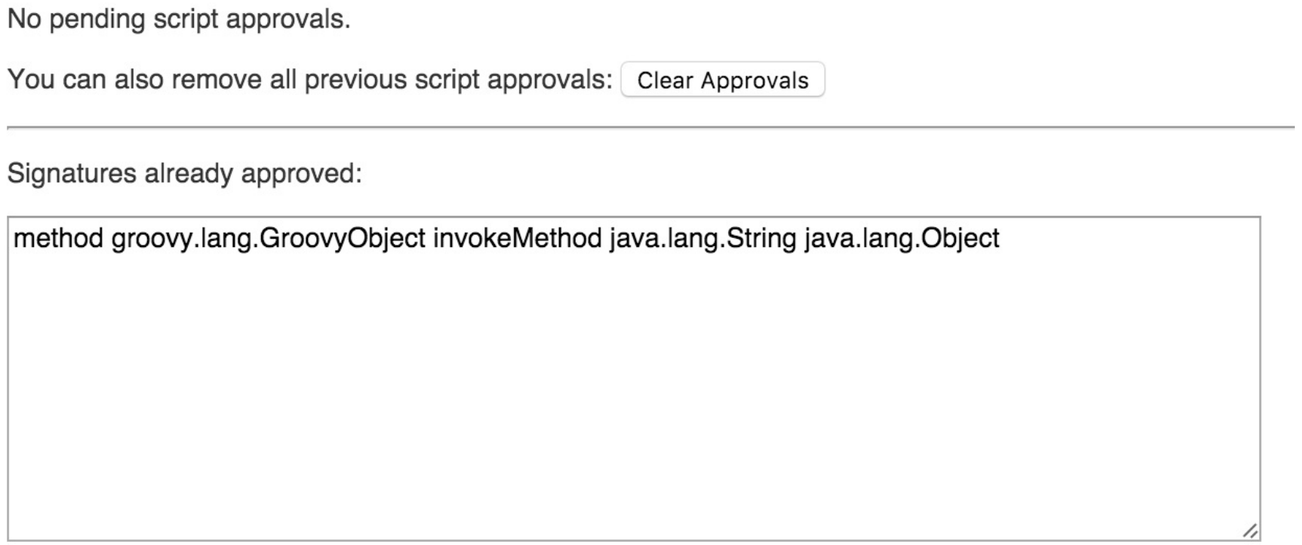
CLOUDBEES UNIVERSITY

MULTI-BRANCH PIPELINE AND JENKINSFILE

```
[Workflow] Allocate node : End
[Workflow] } //node
[Workflow] Allocate node : End
[Workflow] End of Workflow
org.jenkinsci.plugins.scriptsecurity.sandbox.RejectedAccessException: Scripts not permitted to use
method groovy.lang.GroovyObject invokeMethod java.lang.String java.lang.Object
(org.jenkinsci.plugins.workflow.cps.CpsClosure2 sleep java.lang.Integer)
        at
org.jenkinsci.plugins.scriptsecurity.sandbox.whitelists.StaticWhitelist.rejectMethod(StaticWhitelist
.java:155)
        at
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SandboxInterceptor.onMethodCall(SandboxIntercept
or.java:77)
        at
org.jenkinsci.plugins.scriptsecurity.sandbox.groovy.SandboxInterceptor.onMethodCall(SandboxIntercept
or.java:68)
        at org.kohsuke.groovy.sandbox.impl.Checker$1.call(Checker.java:149)
        at org.kohsuke.groovy.sandbox.impl.Checker.checkedCall(Checker.java:146)
        at com.cloudbees.groovy.cps.sandbox.SandboxInvoker.methodCall(SandboxInvoker.java:15)
        at Script1.deploy(Script1.groovy:36)
        at Unknown.Unknown(Unknown)
        at ___cps.transform___(Native Method)
        at com.cloudbees.groovy.cps.impl.ContinuationGroup.methodCall(ContinuationGroup.java:69)
        at
```

CLOUDBEES UNIVERSITY

MULTI-BRANCH PIPELINE AND JENKINSFILE



CLOUDBEES UNIVERSITY

PIPELINE GLOBAL LIBRARY

- Ability to share common parts of Pipeline scripts across multiple jobs
- Keep scripts DRY
- Shared library script Git repository
- Pipeline step load vs global library

Please visit Pipeline Global Library (https://github.com/jenkinsci/workflow-cps-global-lib-plugin/blob/master/README.md) for more info.

CLOUDBEES UNIVERSITY

THE PROJECT - PART 2: REVIEW

CLOUDBEES UNIVERSITY

THE PROJECT - PART 2: REVIEW

- Pipeline Job
- Job Template
- Multibranch Pipeline And Jenkinsfile

CLOUDBEES UNIVERSITY

COURSE REVIEW

CLOUDBEES UNIVERSITY

COURSE REVIEW

- The Need For The Pipeline
- What Is Cloudbees Pipeline?
- The Syntax And The Snippet Generator
- Docker Containers
- Docker Tools
- The Project

CLOUDBEES UNIVERSITY

THE PROJECT - PART 2: EXERCISE

The Project - Part 2: Exercise

CLOUDBEES UNIVERSITY

SYLLABUS AND REFERENCES

- CloudBees Pipeline Plugin Suite (http://documentation.cloudbees.com/docs/cje-user-guide/workflow.html)
- CloudBees Docker Pipeline Plugin (http://documentation.cloudbees.com/docs/cje-user-guide/docker-workflow.html)
- Docker Docs (https://docs.docker.com/)