AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

A Jenkins Experiment from 2017

Henrik Bærbak Christensen

AARHUS UNIVERSITET

Goal

- *To get dirty hands on using a build server system for*
  - *Continuous integration*
    - Unit testing, *real integration testing (accept test/service test/end-to-end test) using live services*
  - *Continuous deployment*
    - Packing services into containers (Docker images)
- Map conceptual framework to real system
  - Pipeline, stages, resources, …

Learning Process

- Tried Concourse
  - Simple model
    - Tasks, Resources, Jobs
  - Failed, did not solve issues with security (timeboxing limit)
- Tried Jenkins
  - Big pile of mud model
    - Two UIs,
    - Two different approaches for defining pipelines (UI / Jenkins file)
    - Two variants of syntax for pipelines in Jenkins files
    - Unholy mix of special syntax and plain old bash scripts
  - But – made it work!

AARHUS UNIVERSITET

First Light

Goal 1: Compile and unit-test TS17* in Jenkins

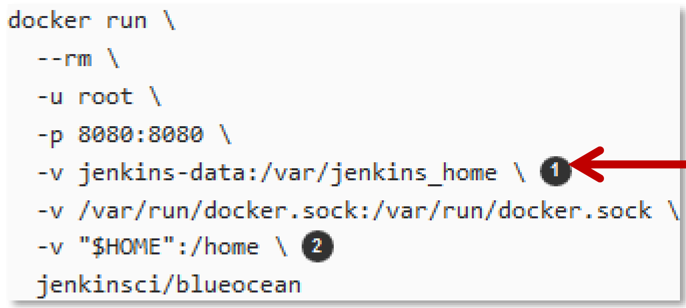
*) TS17 is a RSA case study system

Mise en Place

- To cook that up, we need
  - A running Jenkins server
  - Telling Jenkins to
    - Checkout my TS17D source code (SSH git, means keys)
    - Compile and unit test it (‘gradle test’)
- The Ingrediens
  - Start a Jenkins server (easiest using a docker hub image)
  - Jenkins require an *agent* (~ a machine/node) as execution context
    - Modern default is a docker container
  - Configure with secure key for SSH to ‘git clone’

Jenkins in a Container

- Find some spare CPU/RAM/Disk and use a containerized Jenkins.
  - I took my starting point in
    - https://jenkins.io/doc/tutorials/build-a-java-app-with-maven/

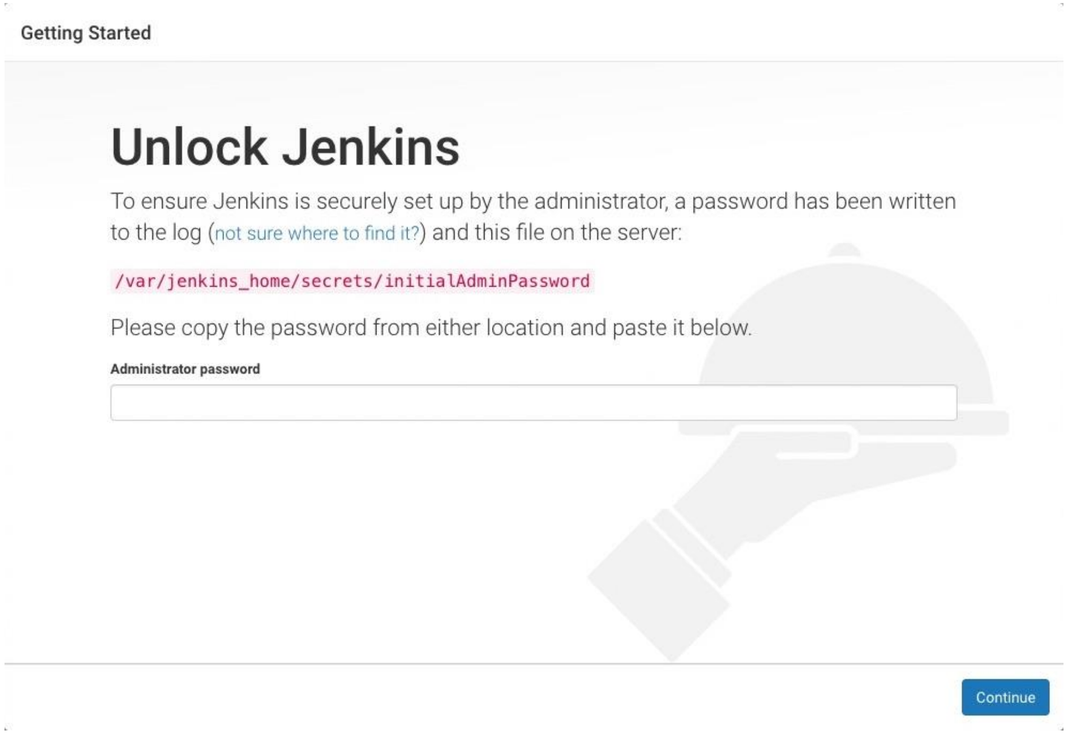


```
docker run \
  --rm \
  -u root \
  -p 8080:8080 \
  -v jenkins-data:/var/jenkins_home \ 1
  -v /var/run/docker.sock:/var/run/docker.sock \
  -v "$HOME":/home \ 2
  jenkinsci/blueocean
```

Better:
-v ~/jenkins-data:/var/jenkins_home

Unlocking

- You need to unlock it, follow the tutorial



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

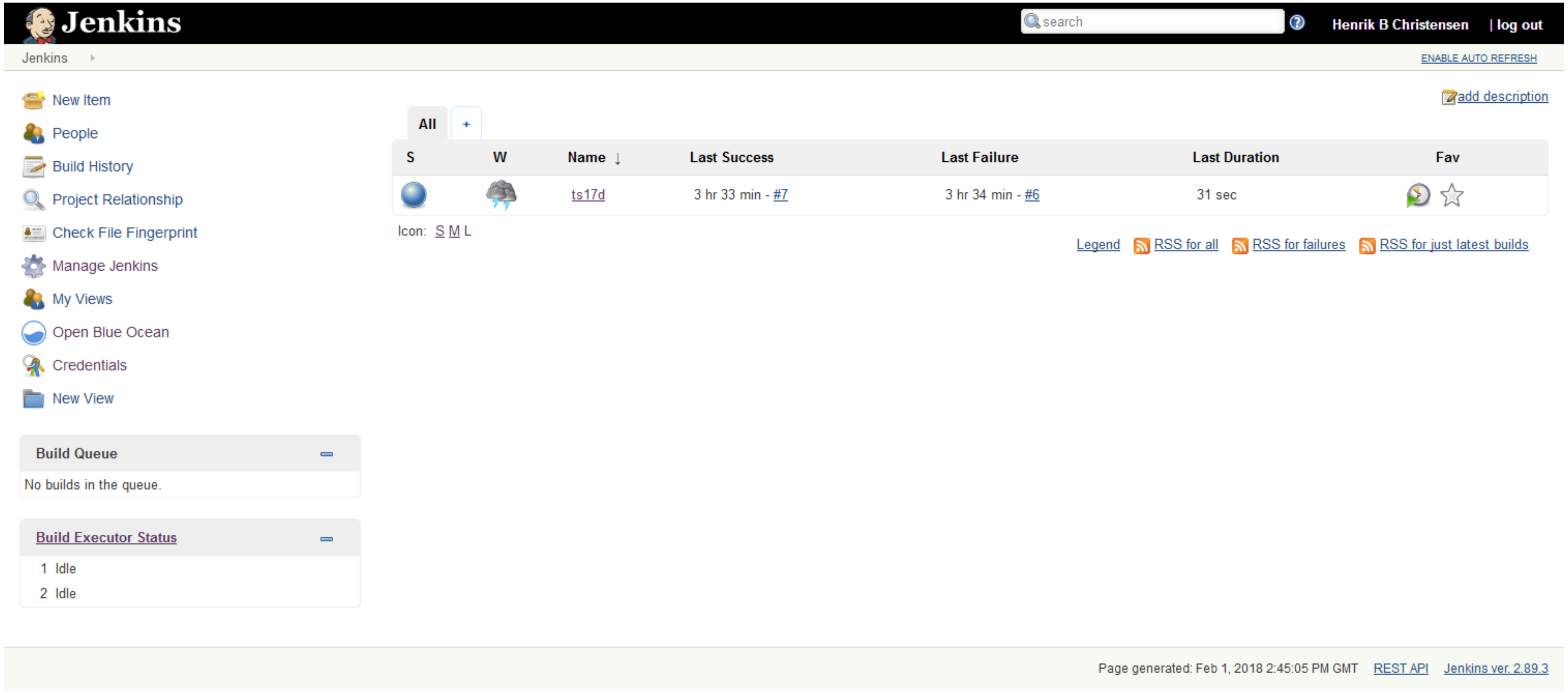
Please copy the password from either location and paste it below.

Administrator password

Note: If you restart the container before it is entered, you are deadlocked.

AARHUS UNIVERSITET

Dashboard



Jenkins

search

Henrik B Christensen | log out

Jenkins ›

ENABLE AUTO REFRESH

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Open Blue Ocean
- Credentials
- New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
●		ts17d	3 hr 33 min - #7	3 hr 34 min - #6	31 sec	☆

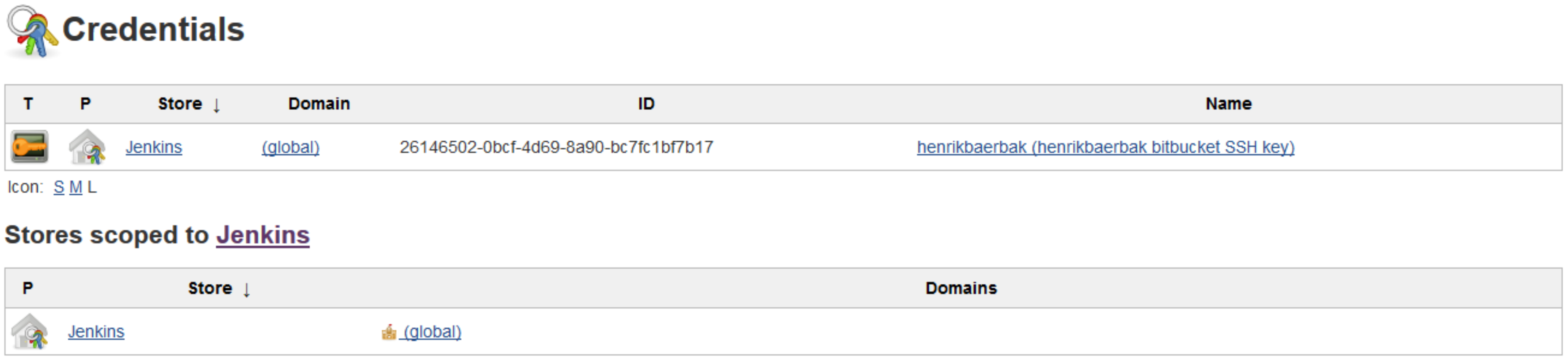
Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

Page generated: Feb 1, 2018 2:45:05 PM GMT REST API Jenkins ver. 2.89.3

Secure Key

- Usual frustration…
  - Need to provide the private key when checking out of bitbucket



Credentials

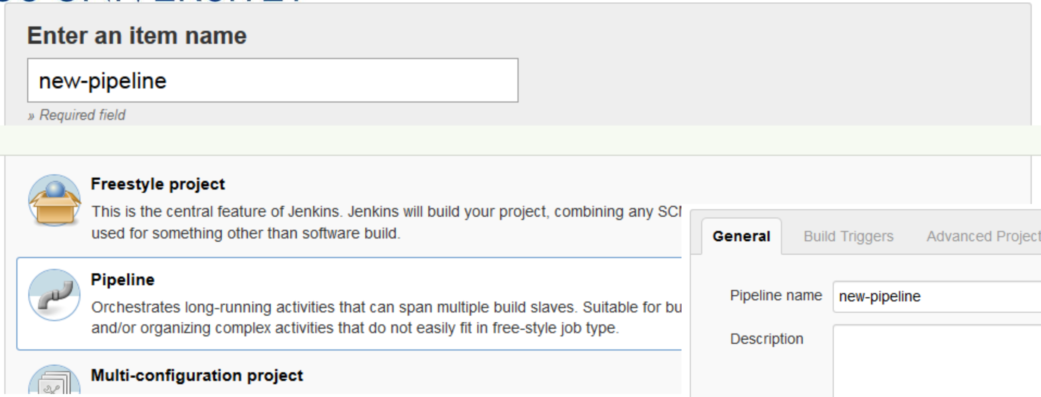
T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	26146502-0bcf-4d69-8a90-bc7fc1bf7b17	henrikbaerbak (henrikbaerbak bitbucket SSH key)

Icon: S M L

Stores scoped to Jenkins

P	Store ↓	Domains
	Jenkins	(global)

Define a Pipeline



Enter an item name

new-pipeline

» Required field

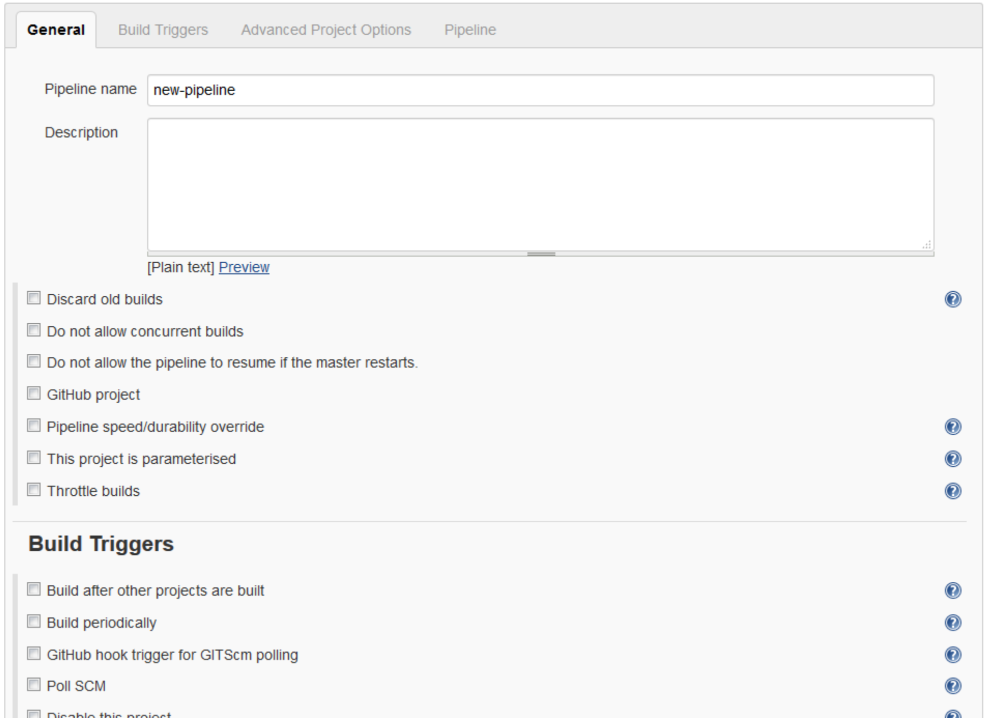
Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCI
used for something other than software build.

Pipeline

Orchestrates long-running activities that can span multiple build slaves. Suitable for bu
and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project



Pipeline name: new-pipeline

Description:

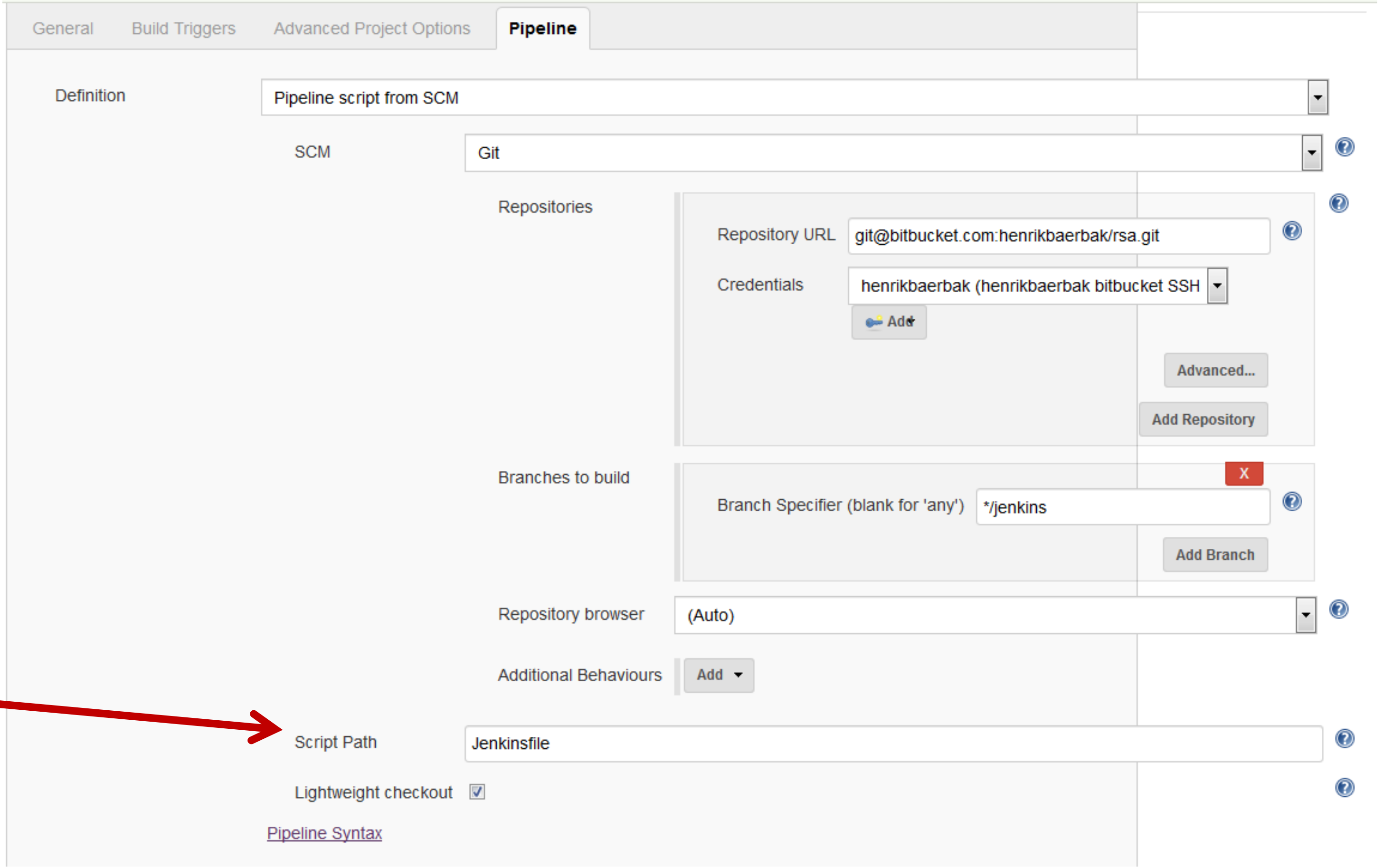
[Plain text] Preview

- ☐ Discard old builds
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the master restarts.
- ☐ GitHub project
- ☐ Pipeline speed/durability override
- ☐ This project is parameterised
- ☐ Throttle builds

Build Triggers

- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ GitHub hook trigger for GITScm polling
- ☐ Poll SCM
- ☐ Disable this project

Jenkinsfile under SCM



General | Build Triggers | Advanced Project Options | **Pipeline**

Definition: Pipeline script from SCM

SCM: Git

Repositories
- Repository URL: git@bitbucket.com:henrikbaerbak/rsa.git
- Credentials: henrikbaerbak (henrikbaerbak bitbucket SSH

Branches to build
- Branch Specifier (blank for 'any'): */jenkins

Repository browser: (Auto)

Additional Behaviours

Script Path: Jenkinsfile

Lightweight checkout: ✓

Pipeline Syntax

SCM Monitoring

- Rightclick pipeline name and choose ‘configure’

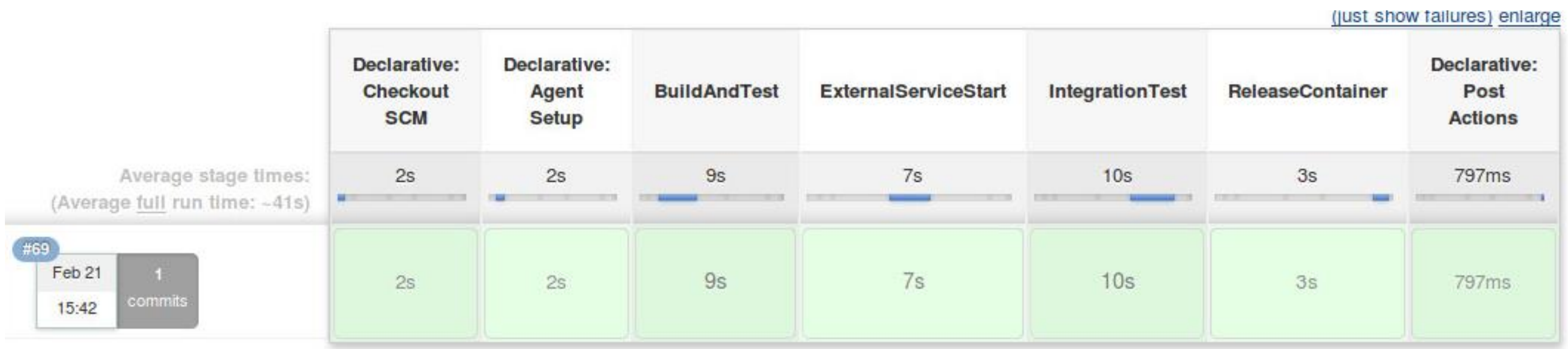
- Poll SCM   To run the pipeline every 5 minutes
  - H/5 * * * *   = poll every 5 minutes (randomly distribution)

  - Will poll, and *only rerun pipeline in case of changes!*

- *I do not think you can trigger builds if your Jenkins is locally installed*
  - *POST from bitbucket to local IP? Nay…*

Pipeline

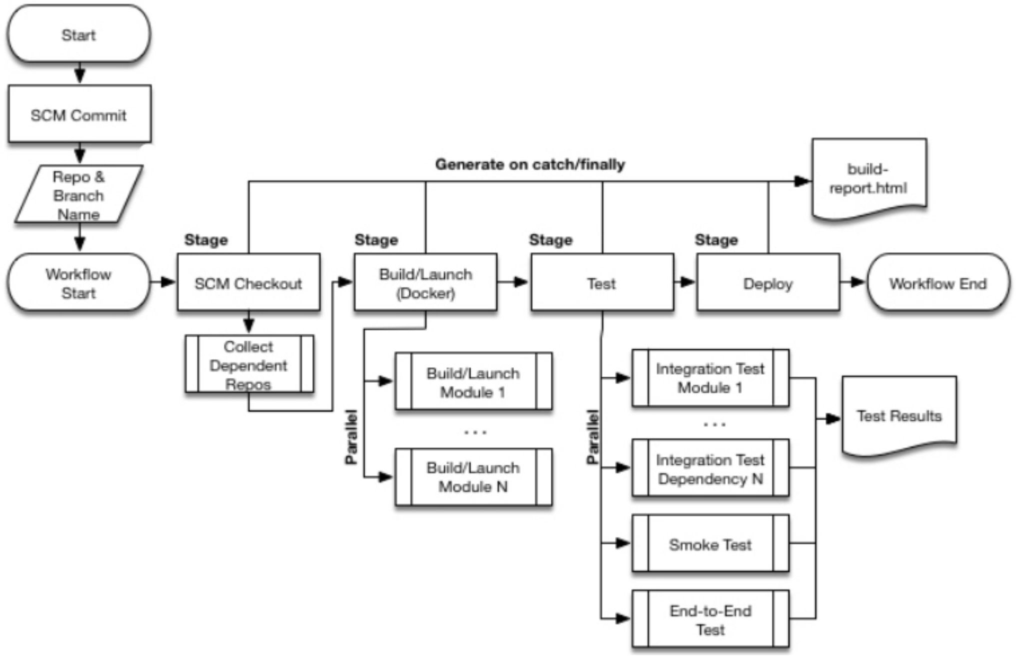
- About 69 commits later (Sigh!)…
  - Commit stage, Accept stage



(just show failures) enlarge

	Declarative: Checkout SCM	Declarative: Agent Setup	BuildAndTest	ExternalServiceStart	IntegrationTest	ReleaseContainer	Declarative: Post Actions
Average stage times: (Average full run time: ~41s)	2s	2s	9s	7s	10s	3s	797ms
#69 Feb 21 15:42 1 commits	2s	2s	9s	7s	10s	3s	797ms

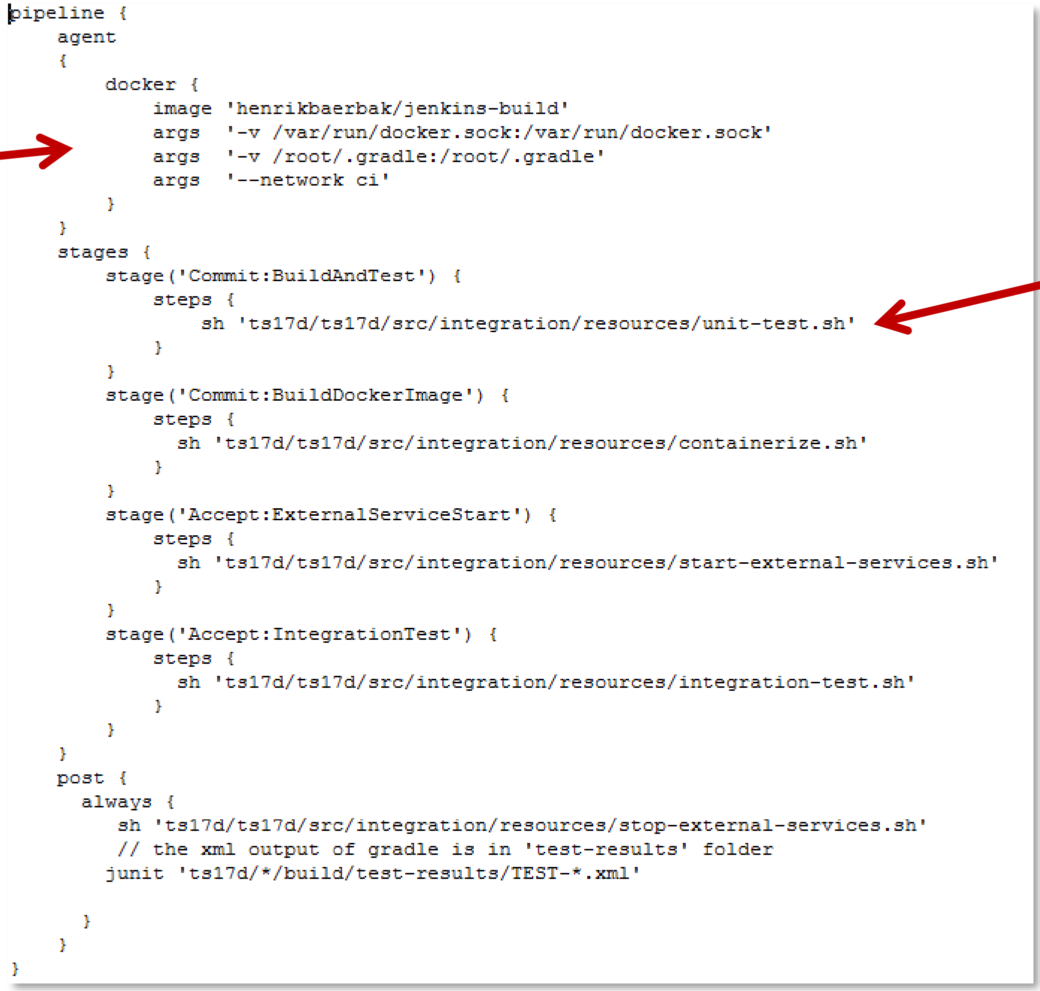
Workflow



https://jenkins.io/doc/book/pipeline/

AARHUS UNIVERSITET

Jenkinsfile

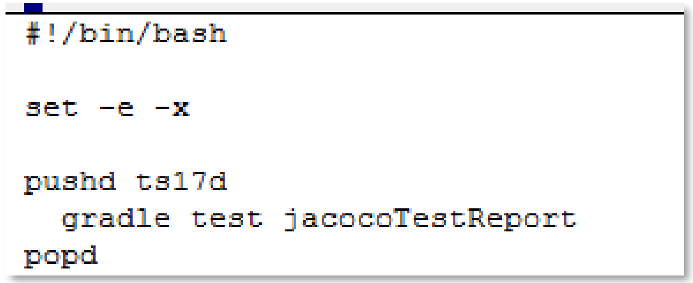


```
pipeline {
    agent
    {
        docker {
            image 'henrikbaerbak/jenkins-build'
            args  '-v /var/run/docker.sock:/var/run/docker.sock'
            args  '-v /root/.gradle:/root/.gradle'
            args  '--network ci'
        }
    }
    stages {
        stage('Commit:BuildAndTest') {
            steps {
                sh 'ts17d/ts17d/src/integration/resources/unit-test.sh'
            }
        }
        stage('Commit:BuildDockerImage') {
            steps {
              sh 'ts17d/ts17d/src/integration/resources/containerize.sh'
            }
        }
        stage('Accept:ExternalServiceStart') {
            steps {
              sh 'ts17d/ts17d/src/integration/resources/start-external-services.sh'
            }
        }
        stage('Accept:IntegrationTest') {
            steps {
              sh 'ts17d/ts17d/src/integration/resources/integration-test.sh'
            }
        }
    }
    post {
      always {
         sh 'ts17d/ts17d/src/integration/resources/stop-external-services.sh'
         // the xml output of gradle is in 'test-results' folder
        junit 'ts17d/*/build/test-results/TEST-*.xml'

      }
    }
}
```

AARHUS UNIVERSITET

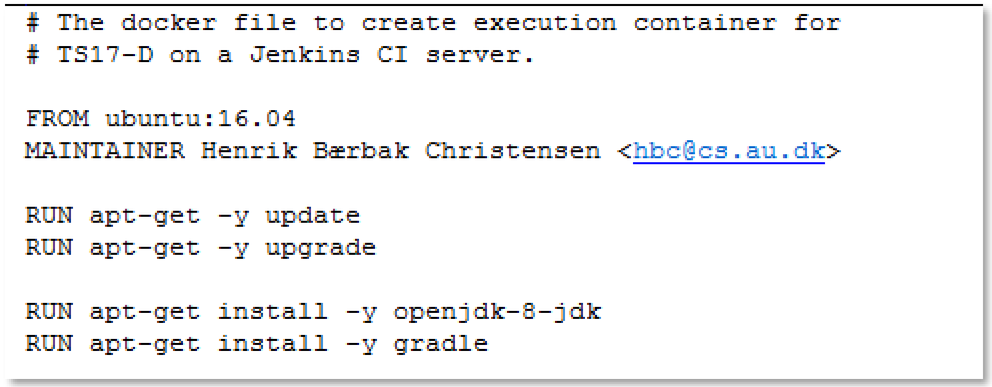
The Easy Part: Unit Test

- Stage: BuildAndTest
- Unit-test.sh $\longrightarrow$



```
#!/bin/bash

set -e -x

pushd ts17d
  gradle test jacocoTestReport
popd
```

- But
  - Require an ‘agent’ = execution context = docker container that includes Java8 and Gradle!
    - Thus you have to build that (or find it)



```
# The docker file to create execution container for
# TS17-D on a Jenkins CI server.

FROM ubuntu:16.04
MAINTAINER Henrik Bærbak Christensen <hbc@cs.au.dk>

RUN apt-get -y update
RUN apt-get -y upgrade

RUN apt-get install -y openjdk-8-jdk
RUN apt-get install -y gradle
```

AARHUS UNIVERSITET

Service Tests

Tricky, in a Docker context

The Tricky Part

- Service Tests / Integration Test / End-to-End test
- Why?
  - Because they test TS17D connected to *external services!*
    - MongoDB, Mountebank, …
- Issues
  - You cannot have the tests in the normal gradle structure (/test)
  - You need to start external services, i.e. ‘docker run’ from scripts
    - But – these scripts are within a docker container!
    - Thus – you need to install docker – in a docker container
  - You need a network
    - So ts17d can see a host that has MongoDB etc.

AARHUS UNIVERSITET

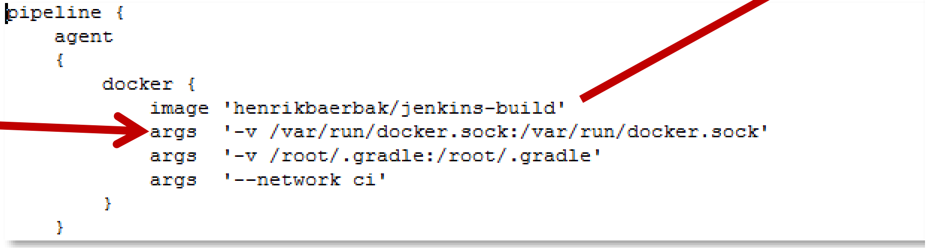
Issue 1: Gradle Integration tests

- Integration (or manual) tests in Gradle
  – Default ‘gradle test’ will execute everything in src/test ☹
- Requires
  – Create new source set ‘src/integration’
  – Create ‘integration’ task which runs as JUnit test all tests defined in ‘src/integration/test’
- Major gradle magic

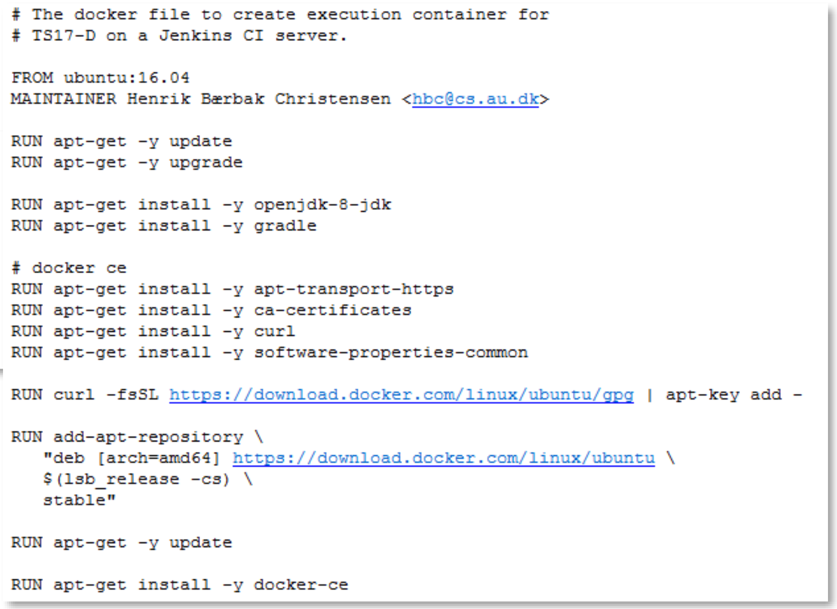
https://www.petrikainulainen.net/programming/gradle/getting-started-with-gradle-integration-testing/

Issue 2: Starting Containers

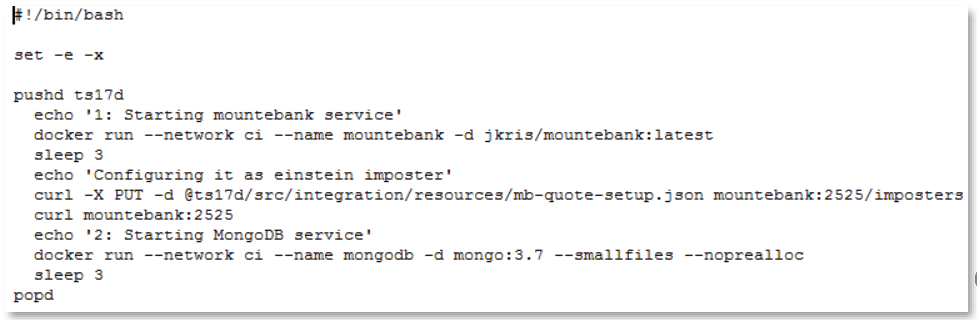
- Start docker containers – inside a docker container ???
  - Yes, you need to
    - Install ‘docker ce’ in the jenkins agent container
    - Mount the docker socket



```
pipeline {
    agent
    {
        docker {
            image 'henrikbaerbak/jenkins-build'
            args  '-v /var/run/docker.sock:/var/run/docker.sock'
            args  '-v /root/.gradle:/root/.gradle'
            args  '--network ci'
        }
    }
```



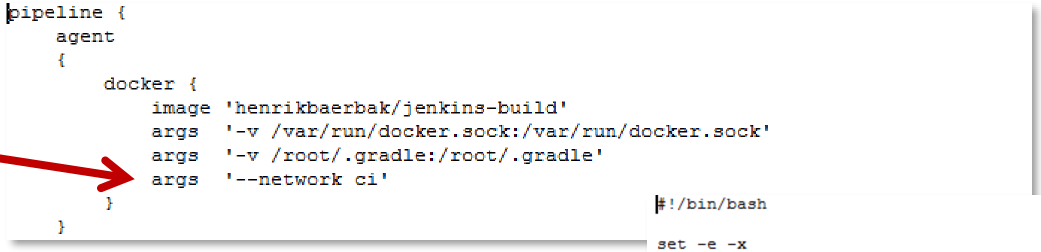
```
# The docker file to create execution container for
# TS17-D on a Jenkins CI server.

FROM ubuntu:16.04
MAINTAINER Henrik Bærbak Christensen <hbc@cs.au.dk>

RUN apt-get -y update
RUN apt-get -y upgrade

RUN apt-get install -y openjdk-8-jdk
RUN apt-get install -y gradle

# docker ce
RUN apt-get install -y apt-transport-https
RUN apt-get install -y ca-certificates
RUN apt-get install -y curl
RUN apt-get install -y software-properties-common

RUN curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -

RUN add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"

RUN apt-get -y update

RUN apt-get install -y docker-ce
```



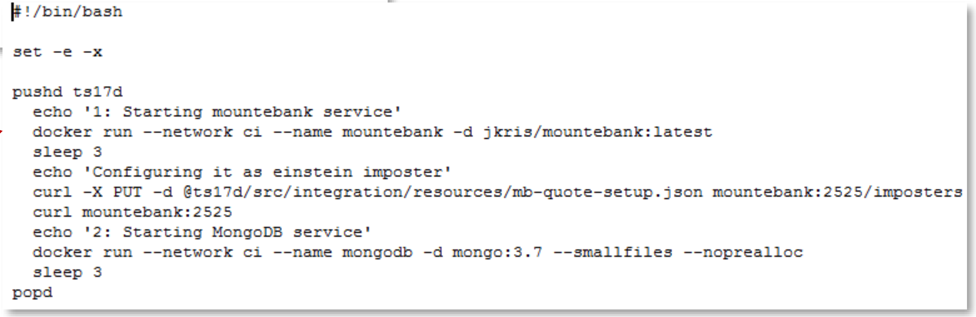
```
#!/bin/bash

set -e -x

pushd ts17d
  echo '1: Starting mountebank service'
  docker run --network ci --name mountebank -d jkris/mountebank:latest
  sleep 3
  echo 'Configuring it as einstein imposter'
  curl -X PUT -d @ts17d/src/integration/resources/mb-quote-setup.json mountebank:2525/imposters
  curl mountebank:2525
  echo '2: Starting MongoDB service'
  docker run --network ci --name mongodb -d mongo:3.7 --smallfiles --noprealloc
  sleep 3
popd
```

Issue 3a: Networking

- Network for TS17D that has e.g. MongoDB deployed
  - No! You cannot use portmapping for this (argue why)
  - Solution: *Named networks* in docker
- On Physical HOST 'docker network create ci'



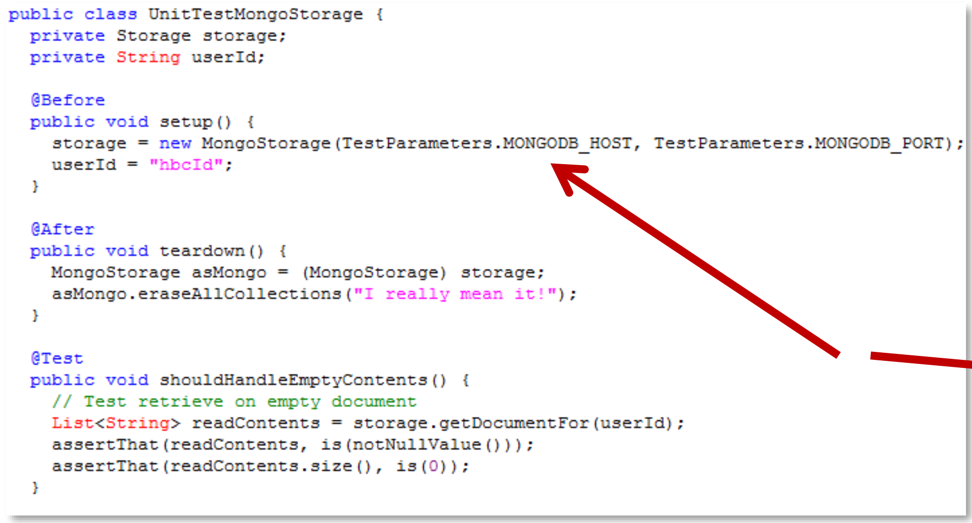
```
pipeline {
    agent
    {
        docker {
            image 'henrikbaerbak/jenkins-build'
            args  '-v /var/run/docker.sock:/var/run/docker.sock'
            args  '-v /root/.gradle:/root/.gradle'
            args  '--network ci'
        }
    }
```



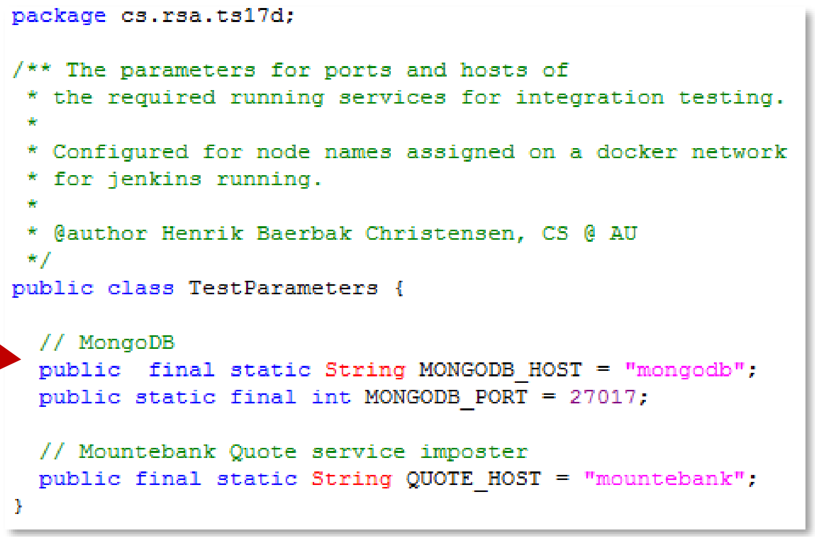
```
#!/bin/bash

set -e -x

pushd ts17d
  echo '1: Starting mountebank service'
  docker run --network ci --name mountebank -d jkris/mountebank:latest
  sleep 3
  echo 'Configuring it as einstein imposter'
  curl -X PUT -d @ts17d/src/integration/resources/mb-quote-setup.json mountebank:2525/imposters
  curl mountebank:2525
  echo '2: Starting MongoDB service'
  docker run --network ci --name mongodb -d mongo:3.7 --smallfiles --noprealloc
  sleep 3
popd
```

Issue 3b: Networking

- All containers on network ‘ci’ can see each other under the *name assigned to the container*
  - *mongodb:27017* *is db server on container mongodb*
  - *mountebank:6777* *is imposter quote service*

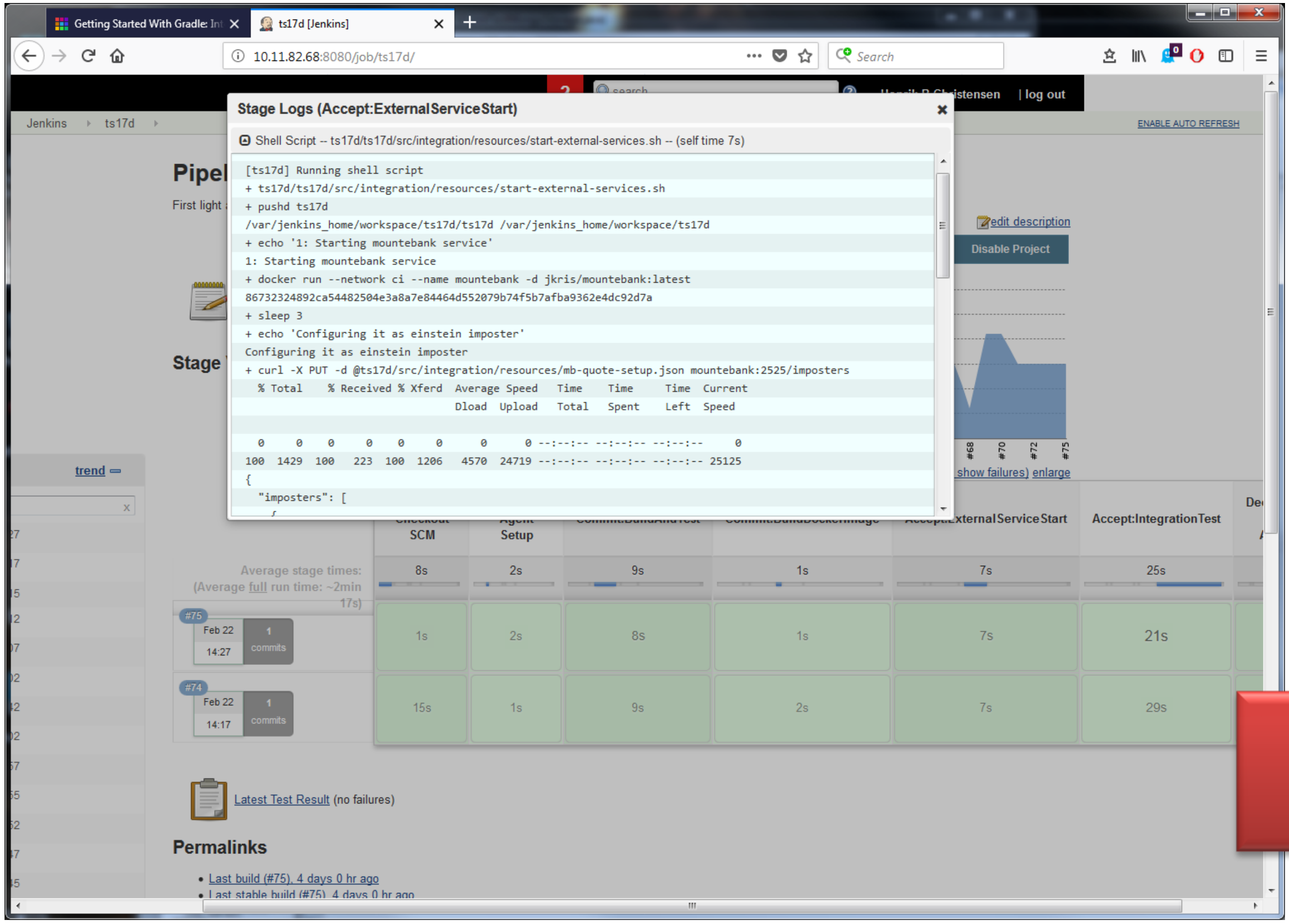


```
public class UnitTestMongoStorage {
  private Storage storage;
  private String userId;

  @Before
  public void setup() {
    storage = new MongoStorage(TestParameters.MONGODB_HOST, TestParameters.MONGODB_PORT);
    userId = "hbcId";
  }

  @After
  public void teardown() {
    MongoStorage asMongo = (MongoStorage) storage;
    asMongo.eraseAllCollections("I really mean it!");
  }

  @Test
  public void shouldHandleEmptyContents() {
    // Test retrieve on empty document
    List<String> readContents = storage.getDocumentFor(userId);
    assertThat(readContents, is(notNullValue()));
    assertThat(readContents.size(), is(0));
  }
```



```
package cs.rsa.ts17d;

/** The parameters for ports and hosts of
 * the required running services for integration testing.
 *
 * Configured for node names assigned on a docker network
 * for jenkins running.
 *
 * @author Henrik Baerbak Christensen, CS @ AU
 */
public class TestParameters {

  // MongoDB
  public  final static String MONGODB_HOST = "mongodb";
  public static final int MONGODB_PORT = 27017;

  // Mountebank Quote service imposter
  public final static String QUOTE_HOST = "mountebank";
}
```

Phew…



Stage Logs (Accept:ExternalServiceStart)

Shell Script -- ts17d/ts17d/src/integration/resources/start-external-services.sh -- (self time 7s)

```
[ts17d] Running shell script
+ ts17d/ts17d/src/integration/resources/start-external-services.sh
+ pushd ts17d
/var/jenkins_home/workspace/ts17d/ts17d /var/jenkins_home/workspace/ts17d
+ echo '1: Starting mountebank service'
1: Starting mountebank service
+ docker run --network ci --name mountebank -d jkris/mountebank:latest
86732324892ca54482504e3a8a7e84464d552079b74f5b7afba9362e4dc92d7a
+ sleep 3
+ echo 'Configuring it as einstein imposter'
Configuring it as einstein imposter
+ curl -X PUT -d @ts17d/src/integration/resources/mb-quote-setup.json mountebank:2525/imposters
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  1429  100   223  100  1206   4570  24719 --:--:-- --:--:-- --:--:-- 25125
{
  "imposters": [
```

	Checkout SCM	Agent Setup	Commit:BuildAndTest	Commit:BuildDockerImage	Accept:ExternalServiceStart	Accept:IntegrationTest
Average stage times: (Average full run time: ~2min 17s)	8s	2s	9s	1s	7s	25s
#75 Feb 22 14:27 1 commits	1s	2s	8s	1s	7s	21s
#74 Feb 22 14:17 1 commits	15s	1s	9s	2s	7s	29s

Latest Test Result (no failures)

Permalinks

- Last build (#75), 4 days 0 hr ago
- Last stable build (#75), 4 days 0 hr ago

This was a *long* journey…

AARHUS UNIVERSITET

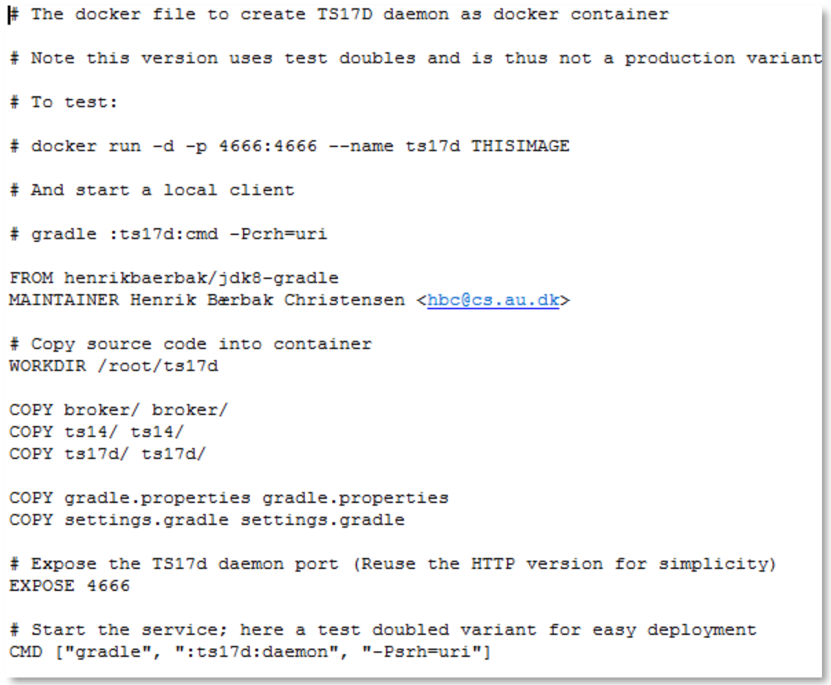
Release

Jez Humble: Commit Stage

- Recall: Commit stage
  - Compile code, run unit tests, *create binaries for later stages*

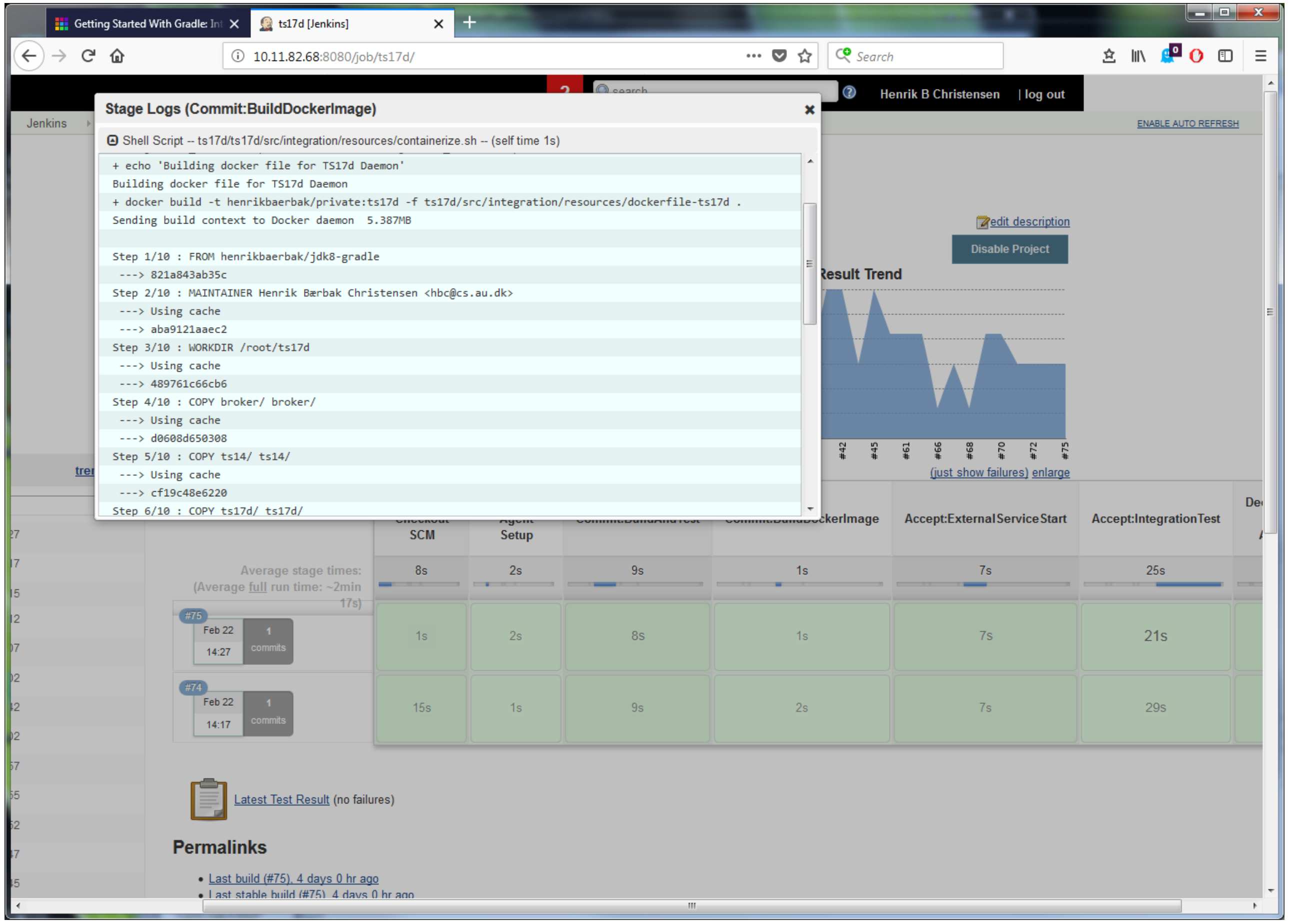
- *Docker context*
  - *Build an image for TS17D*

Note: This image actually builds from source code! Binary distribution pending



```
# The docker file to create TS17D daemon as docker container

# Note this version uses test doubles and is thus not a production variant

# To test:

# docker run -d -p 4666:4666 --name ts17d THISIMAGE

# And start a local client

# gradle :ts17d:cmd -Pcrh=uri

FROM henrikbaerbak/jdk8-gradle
MAINTAINER Henrik Bærbak Christensen <hbc@cs.au.dk>

# Copy source code into container
WORKDIR /root/ts17d

COPY broker/ broker/
COPY ts14/ ts14/
COPY ts17d/ ts17d/

COPY gradle.properties gradle.properties
COPY settings.gradle settings.gradle

# Expose the TS17d daemon port (Reuse the HTTP version for simplicity)
EXPOSE 4666

# Start the service; here a test doubled variant for easy deployment
CMD ["gradle", ":ts17d:daemon", "-Psrh=uri"]
```

Docker build



Stage Logs (Commit:BuildDockerImage)

Shell Script -- ts17d/ts17d/src/integration/resources/containerize.sh -- (self time 1s)

```
+ echo 'Building docker file for TS17d Daemon'
Building docker file for TS17d Daemon
+ docker build -t henrikbaerbak/private:ts17d -f ts17d/src/integration/resources/dockerfile-ts17d .
Sending build context to Docker daemon  5.387MB

Step 1/10 : FROM henrikbaerbak/jdk8-gradle
 ---> 821a843ab35c
Step 2/10 : MAINTAINER Henrik Bærbak Christensen <hbc@cs.au.dk>
 ---> Using cache
 ---> aba9121aaec2
Step 3/10 : WORKDIR /root/ts17d
 ---> Using cache
 ---> 489761c66cb6
Step 4/10 : COPY broker/ broker/
 ---> Using cache
 ---> d0608d650308
Step 5/10 : COPY ts14/ ts14/
 ---> Using cache
 ---> cf19c48e6220
Step 6/10 : COPY ts17d/ ts17d/
```

	Checkout SCM	Agent Setup	Commit:BuildAndTest	Commit:BuildDockerImage	Accept:ExternalServiceStart	Accept:IntegrationTest
Average stage times: (Average full run time: ~2min 17s)	8s	2s	9s	1s	7s	25s
#75 Feb 22 14:27 1 commits	1s	2s	8s	1s	7s	21s
#74 Feb 22 14:17 1 commits	15s	1s	9s	2s	7s	29s

Getting Started With Gradle: Int
ts17d [Jenkins]
10.11.82.68:8080/job/ts17d/
Henrik B Christensen
| log out
ENABLE AUTO REFRESH
Jenkins
edit description
Disable Project
Result Trend
#42 #45 #61 #66 #68 #70 #72 #75
(just show failures) enlarge
Latest Test Result (no failures)
Permalinks
Last build (#75), 4 days 0 hr ago
Last stable build (#75), 4 days 0 hr ago

AARHUS UNIVERSITET

Deploy to Production

AARHUS UNIVERSITET

Hmmm…

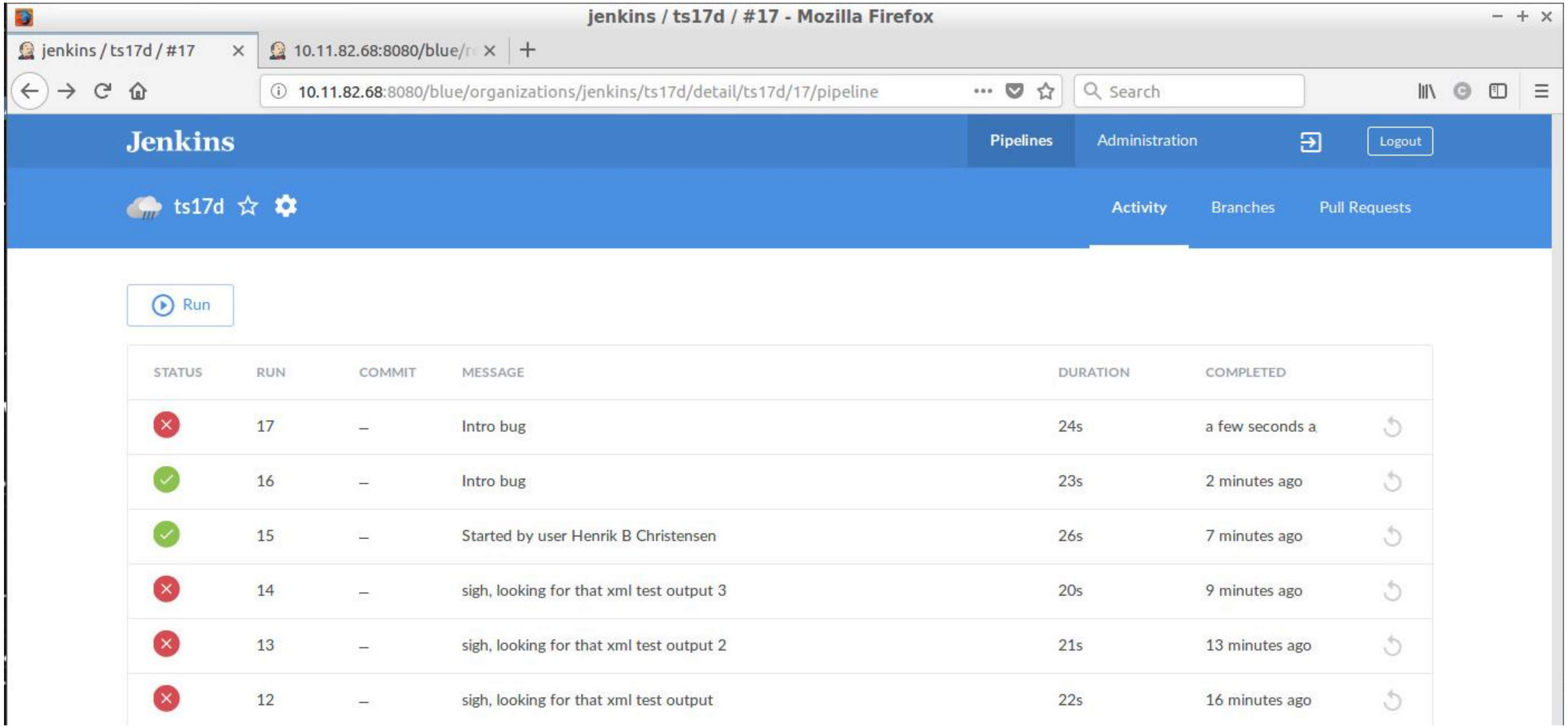
- I have not done it…
  - Use Droplet API, use AWS API, etc, from a shell script in a stage in Jenkinsfile
- Or
  - Uber uDeploy and uOrchestrate
  - …
- Or
  - Rancher/Kubernetes/whatever UI

AARHUS UNIVERSITET

Monitoring / Jenkins UI

AARHUS UNIVERSITET

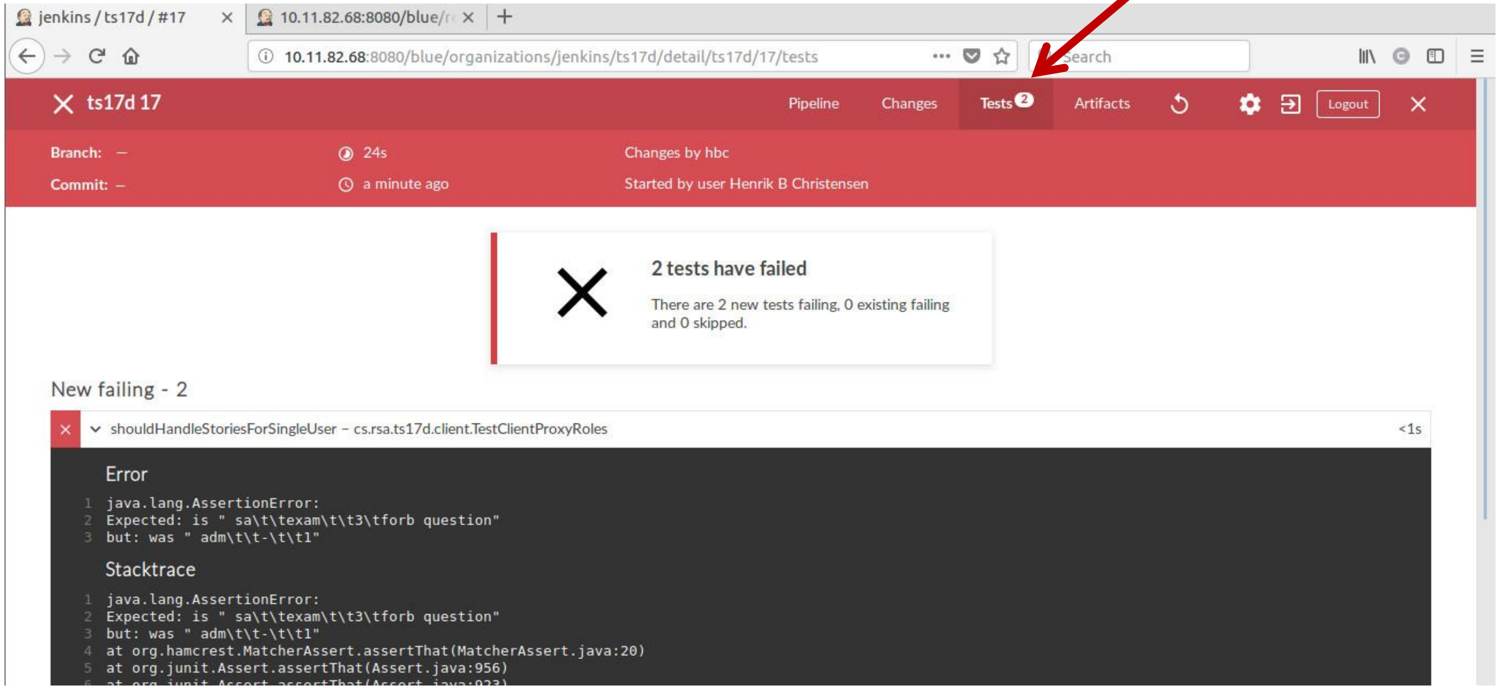
GUI # 2: Blue Ocean



STATUS	RUN	COMMIT	MESSAGE	DURATION	COMPLETED
✗	17	–	Intro bug	24s	a few seconds a
✓	16	–	Intro bug	23s	2 minutes ago
✓	15	–	Started by user Henrik B Christensen	26s	7 minutes ago
✗	14	–	sigh, looking for that xml test output 3	20s	9 minutes ago
✗	13	–	sigh, looking for that xml test output 2	21s	13 minutes ago
✗	12	–	sigh, looking for that xml test output	22s	16 minutes ago

Failed Test cases

- Jenkins can fetch JUnit reports



Branch: –
Commit: –
24s
a minute ago
Changes by hbc
Started by user Henrik B Christensen

2 tests have failed

There are 2 new tests failing, 0 existing failing and 0 skipped.

New failing - 2

shouldHandleStoriesForSingleUser – cs.rsa.ts17d.client.TestClientProxyRoles <1s

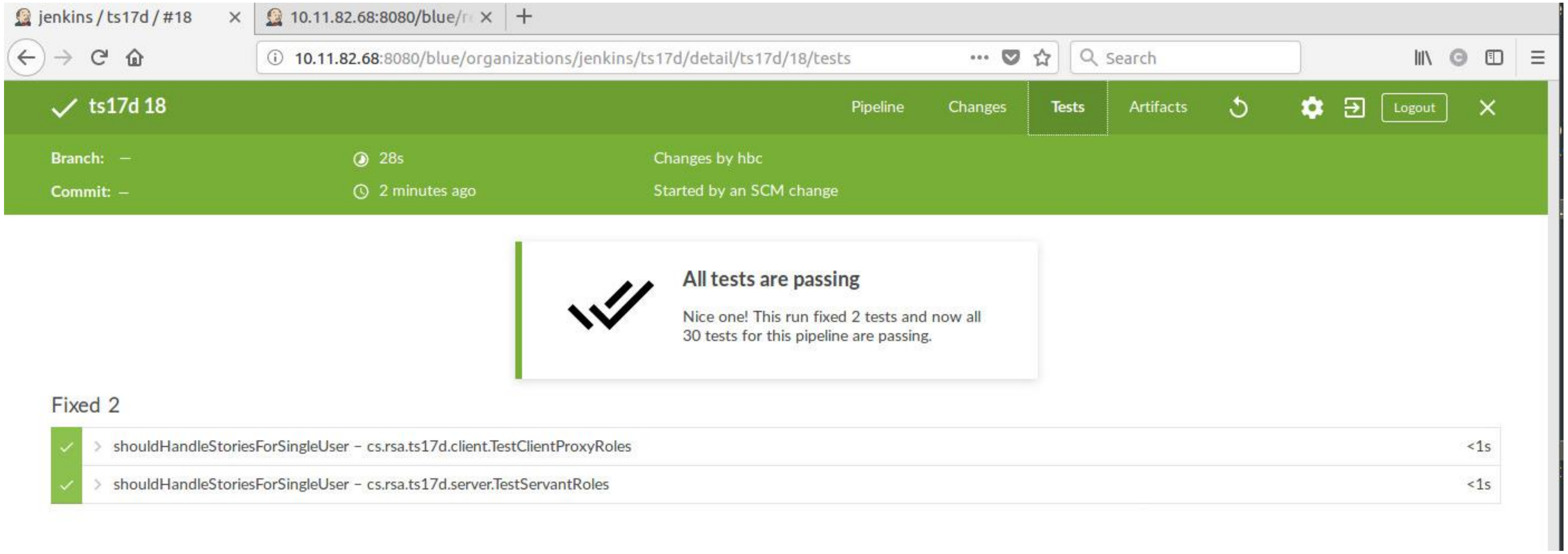
Error

```
java.lang.AssertionError:
Expected: is " sa\t\texam\t\t3\tforb question"
but: was " adm\t\t-\t\t1"
```

Stacktrace

```
java.lang.AssertionError:
Expected: is " sa\t\texam\t\t3\tforb question"
but: was " adm\t\t-\t\t1"
at org.hamcrest.MatcherAssert.assertThat(MatcherAssert.java:20)
at org.junit.Assert.assertThat(Assert.java:956)
```

Back On Track



✓ ts17d 18

Branch: –
Commit: –
28s
2 minutes ago
Changes by hbc
Started by an SCM change

All tests are passing

Nice one! This run fixed 2 tests and now all 30 tests for this pipeline are passing.

Fixed 2

✓	shouldHandleStoriesForSingleUser – cs.rsa.ts17d.client.TestClientProxyRoles	<1s
✓	shouldHandleStoriesForSingleUser – cs.rsa.ts17d.server.TestServantRoles	<1s

Discussion

Discussion

- Lots and lots of hard bindings ☹
  - Language mix (Jenkins+Shell) instead of dedicated DSL
    - And ‘something else’ for Windows…
  - Lots of magic constants, no means of abstraction
    - Jenkins refers to *named* shell scripts
    - Scripts refers to *named* images, resources, …
    - Identical *names* in the JUnit test code
  - Quite a few environmental dependencies
    - Proper setup of Jenkins container, agent containers
    - Dependency on previously made docker network

Consistency!
Refactoring!

Discussion

- Slow development turn around
  - Fail? Change a bit, commit, and push, and hit ‘build now’ in Jenkins, and review
    - Slow and manual
    - Pollute git branch with stupid commits ala “does this work then???”
  - Ex: Change ‘--ti’ to ‘-ti’, commit, push, jenkins build, review failure…

- Differences in environment (jenkins/manual)
  - Can I just run the integration test scripts? No!
    - Hard couplings to special environment in Jenkins

Discussion

- The *evolutionary model is much too visible*
  - Two different GUIs
    - Classic and 'Blue Ocean' look very different
  - Jenkins file can be in one of two different formats
    - Scripted and Declarative
  - Pipelines can be created using the GUI alone

- Which means:
  - Every tutorial/guide you find on the net *explains the solution to your problem using another specification model than the one you have adopted!!!*

Discussion

- The documentation issue…
- I have unfortunately seen this issue very often
  - Superficial tutorial material
    - Explaining 10% using non-relevant examples
      - “No, I will not use Jenkins to echo ‘This is stage 1’ onto the terminal!!!”
  - Combined with reference material suitable for experts only
  - Took a lot of effort to crack the ‘start some services’ nut…
- Ant, Ivy, Gradle, Jenkins follow this unfortunate pattern

Conclusion

- Mixed…

- *The concepts and motivation are ‘right’*

- *The Jenkins tool is not IMO*

AARHUS UNIVERSITET

Pipeline Syntax

Internal notes ☺

Where am I?

- Point of confusion:
  - The folders involved, where is ‘current folder’ in Jenkins?

- Given git project ‘rsa’ you will clone to ‘~/rsa’
  - Put Jenkinsfile in the project root ~/rsa/Jenkinsfile
  - In stages, files are *referenced from this folder*
    - *Sh ‘ts17d/ci/test.sh’* if test.sh is in ~/rsa/ts17d/ci folder
  - In the shell scripts the same folder is the ‘current’
    - *Pushd ts17d* will change to ~/rsa/ts17