redhat.

Fuse Integration Services 2.0

A systems integration lecture

Mgr. David Simansky
Quality Engineer, Red Hat Middleware

2017-03-03

FIS 2.0

Introduction
Relation to JBoss Fuse

Karaf
Is OSGi in containers still needed?

Spring Boot
Is Winter really ending soon?

Fabric8 Maven Plugin
Maven workflow for OpenShift development

CI/CD
Connecting the dots/pipelines

FIS 2.0

What are the boundaries?

- OpenShift extension for JBoss Fuse
- Versions aligned to 6.3 release
- S2I Docker images
  - fis-java-openshift:2.0 (Spring Boot / generic uber-jar apps)
  - fis-karaf-openshift:2.0 (Karaf micro distributions)
- Karaf 4 Maven Plugin
- Spring Boot support in Camel 2.18
- Fabric8 Maven Plugin (f-m-p or FMP)

Karaf

Karaf

Is it still viable in cloud?

- Microservice ready
  - Define your own small distor with Karaf 4 Maven plugin
  - Health/Readiness checks
  - Benefit from separated classloader
  - Reload config from ConfigMap
- Container in container overhead (inception)
- There might be other drawbacks
  - Containers are immutable
  - No use for runtime patching / hot deployment
  - No need for service discovery

Karaf

Karaf 4 Maven Plugin

```
// Too long for snippet field
https://github.com/fabric8-quickstarts/karaf2-camel-rest-sql/blob/master/pom.xml
```

Karaf

Fabric8 Karaf Config Admin

```
amq.usr = $[k8s:secret:$[env:ACTIVEMQ_SERVICE_NAME]/username]
amq.pwd = $[k8s:secret:$[env:ACTIVEMQ_SERVICE_NAME]/password]
amq.url = tcp://$[env+service:ACTIVEMQ_SERVICE_NAME]
```

Spring Boot

Spring Boot

Veni, vidi, spingbooted

- Spring application that you can “just run”
- Opinionated defaults out of the box
  - Ease up development
  - Reduce configuration complexity
  - Focus on business logic not framework options
- No code generation
- No requirement for XML configuration
- Embedded HTTP server
- Autoconfiguration
- Readiness/Health checks

Spring Boot

Autoconfigure

- Attempts to automatically configure your application
- Based on class path analysis
- @Enable*
    - @EnableAutoConfiguration
    - @EnableWebMvc
    - @EnableTransactionManagement
- @Conditional
    - @ConditionOnClass
    - @ConditionalOnMissingBean

Spring Boot & Camel

- Camel version 2.18
- Components starters out of the box
- Autoconfiguration support
- Auto detection of Camel routes

Spring Boot & Camel

HelloWorld

```
@SpringBootApplication
public class Application extends RouteBuilder {

     // must have a main method spring-boot can run
     public static void main(String[] args) {
          SpringApplication.run(Application.class, args);
     }

     @Override
     public void configure() throws Exception {
     from("timer://foo?period=5000")
          .setBody().constant("Hello World")
          .log(">>> ${body}");
     }
}
```

Spring Boot & Camel

Auto detect routes

```
@Component
public class MyRouter extends RouteBuilder {

  @Override
  public void configure() throws Exception {
    from("jms:invoices").to("file:/invoices");
  }
}
```

Spring Boot & Camel

That simple?

```
<dependency>
   <groupId>org.apache.camel</groupId>
   <artifactId>camel-spring-boot-starter</artifactId>
</dependency>
```

- https://github.com/fabric8-quickstarts/spring-boot-camel-amq

Fabric8 Maven Plugin

Fabric8 Maven Plugin

https://maven.fabric8.io/

- Purpose
  - Build Docker images from source (S2I)
  - Create OpenShift resources DC, BC, services, routes
- Based on opinionated defaults (as usual)
- Ironed-out and simplified workflow
  - “mvn clean install” is enough
- Detect cloud platform K8s vs. OpenShift
- Invoke S2I binary build

Fabric8 Maven Plugin

That simple?

```
<plugin>
     <groupId>io.fabric8</groupId>
     <artifactId>fabric8-maven-plugin</artifactId>
     <version>${fabric8.maven.plugin.version}</version>
     <executions>
          <execution>
          <goals>
               <goal>resource</goal>
               <goal>build</goal>
          </goals>
          </execution>
     </executions>
     </plugin>
</plugins>
```

Fabric8 Maven Plugin

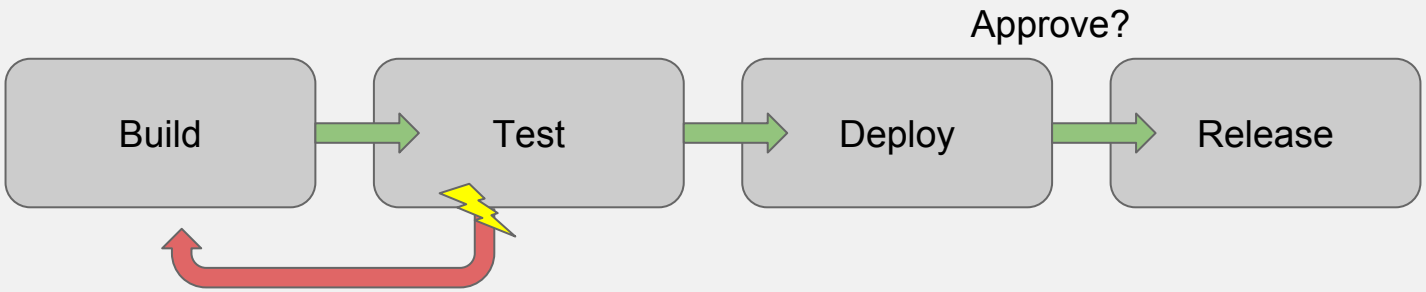
Goals

```
fabric8:help

fabric8:resource

fabric8:resource-apply

fabric8:build

fabric8:deploy
```

CI CD Pipelines

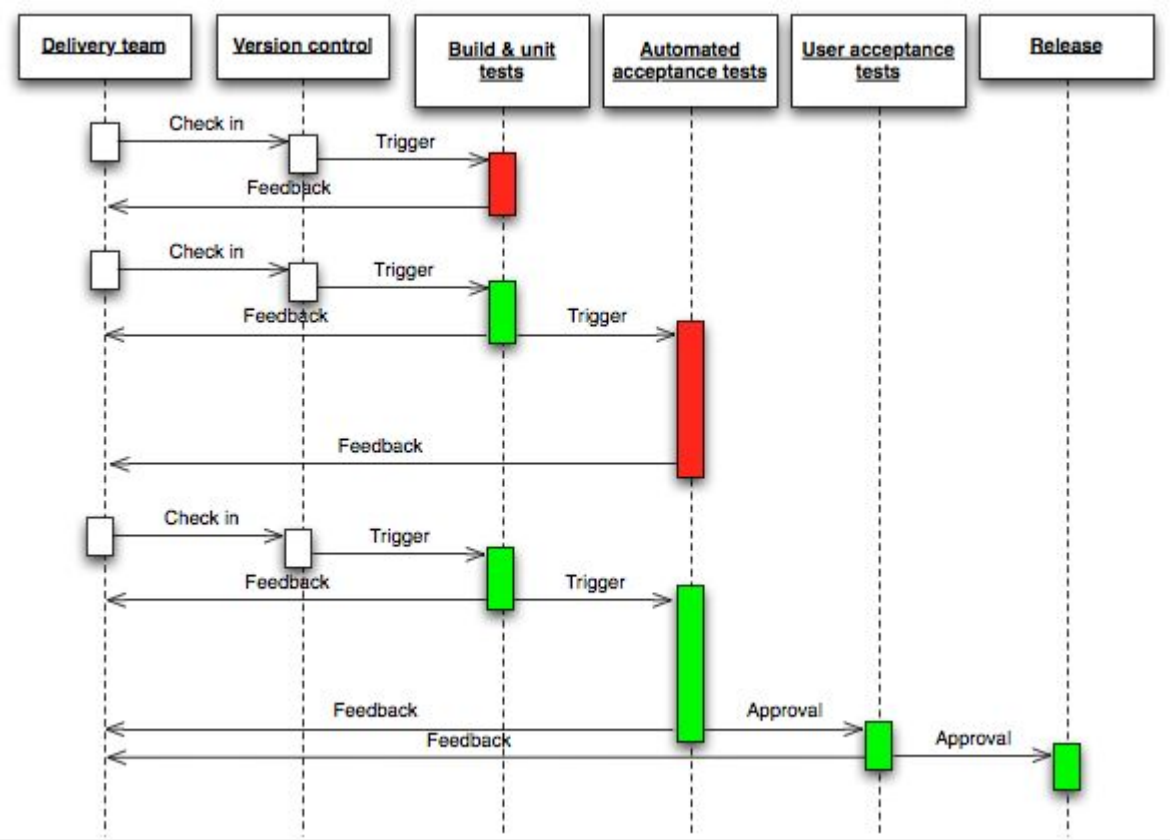
CI CD Pipelines

Motivation



CI CD Pipelines

Motivation



Source: https://en.wikipedia.org/wiki/Continuous_delivery

CI CD Pipelines

Jenkins (Workflow) Pipeline

- Freedom to define the steps of CI job
- Scripted workflow
- Mix shell commands with Jenkins plugins
- Build steps, post-build actions, build environments
- Deploy to cloud, stage, production
- Trigger other jobs or services

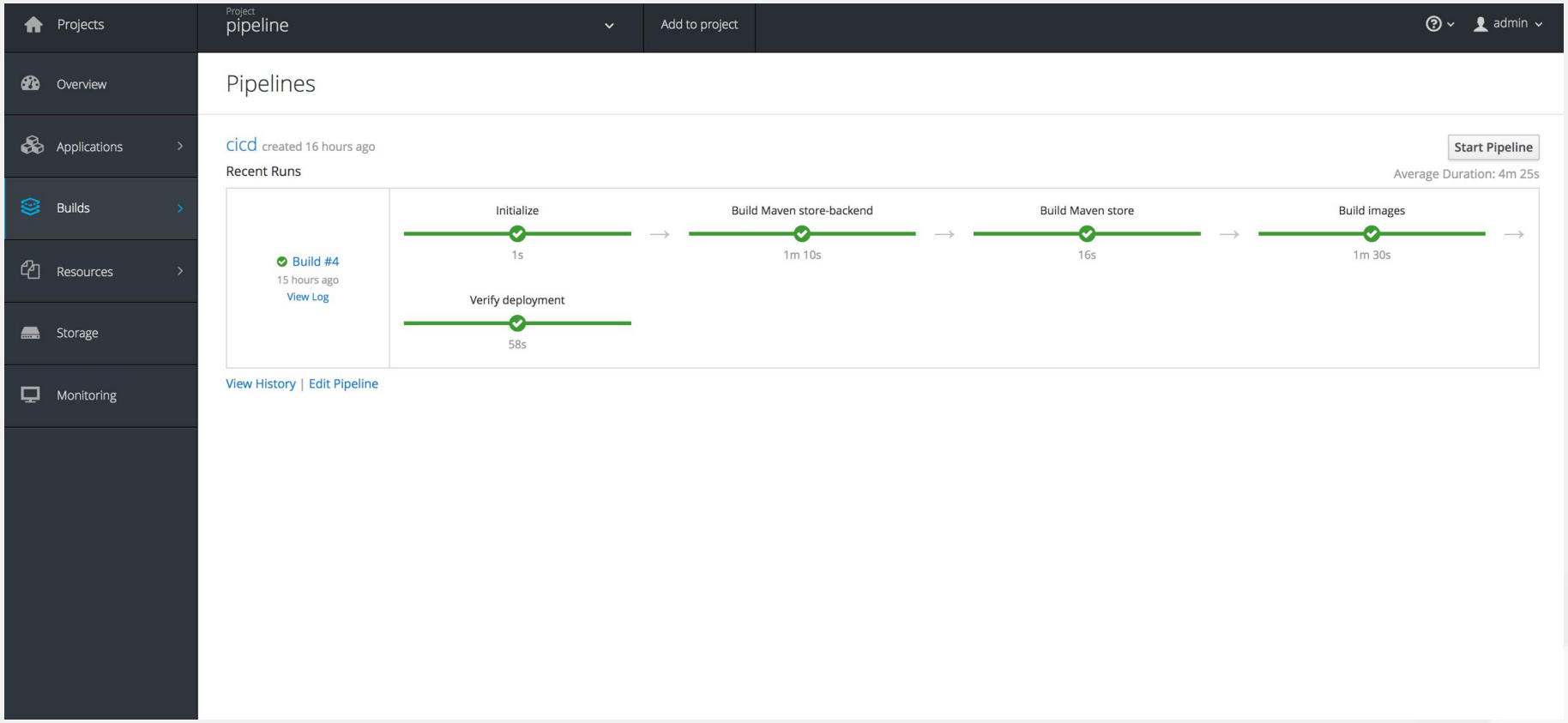
CI CD Pipelines

OpenShift

https://github.com/isss-apps/workshop/tree/master/cicd-pipeline

CI CD Pipeline

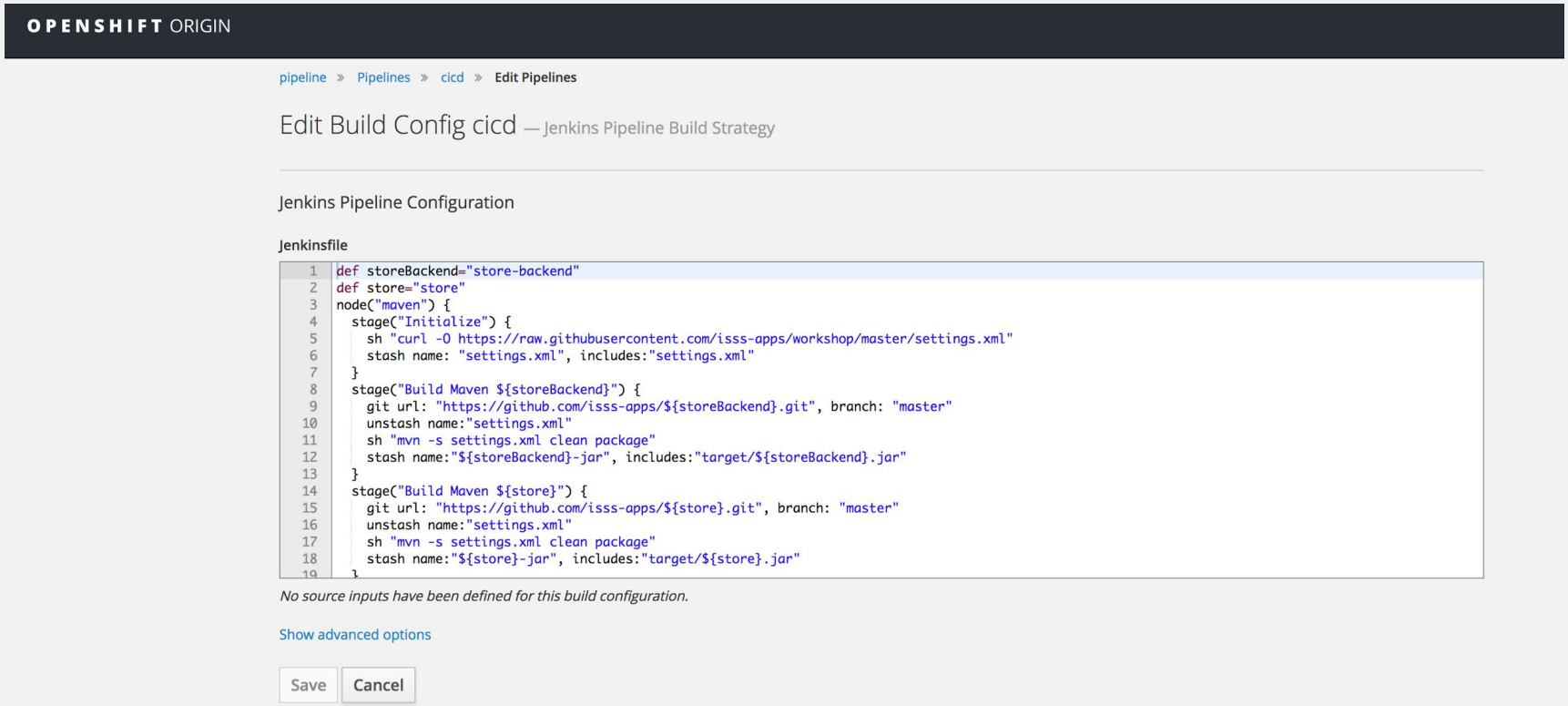
OpenShift console view



redhat.

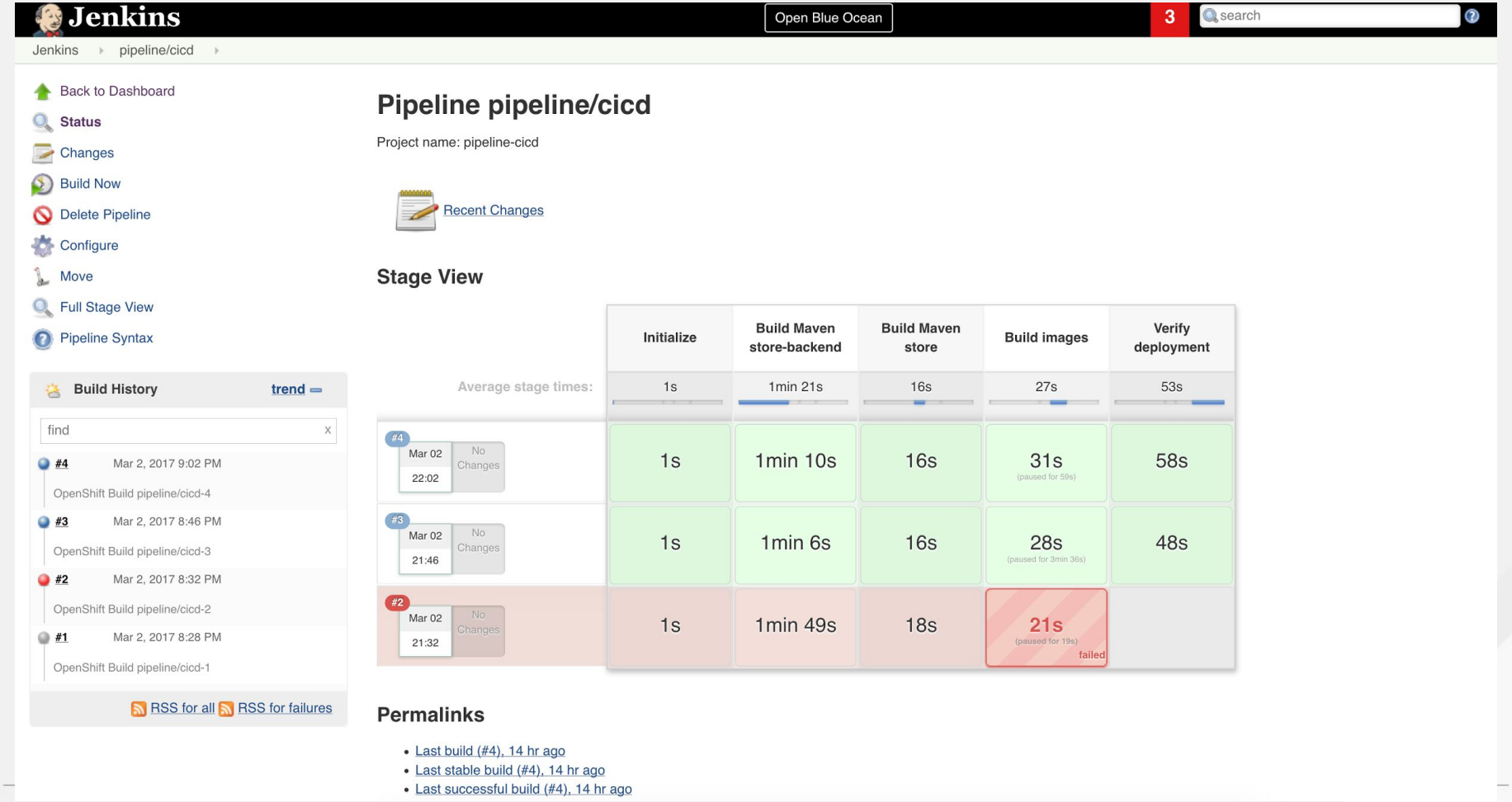
CI CD Pipeline

OpenShift console edit view



CI CD Pipeline

Jenkins console view



redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos

CODE SNIPPET

When referencing code snippets, use the template below. Resize the snippet box to the appropriate size for your text.

```
Type code snippet here
```