RUNNING JAVA APPLICATIONS ON CLOUD FOUNDRY

SCOTT FREDERICK

Community Engineer on Cloud Foundry at Pivotal

@scottyfred

http://www.linkedin.com/in/scottfrederick

https://github.com/scottfrederick

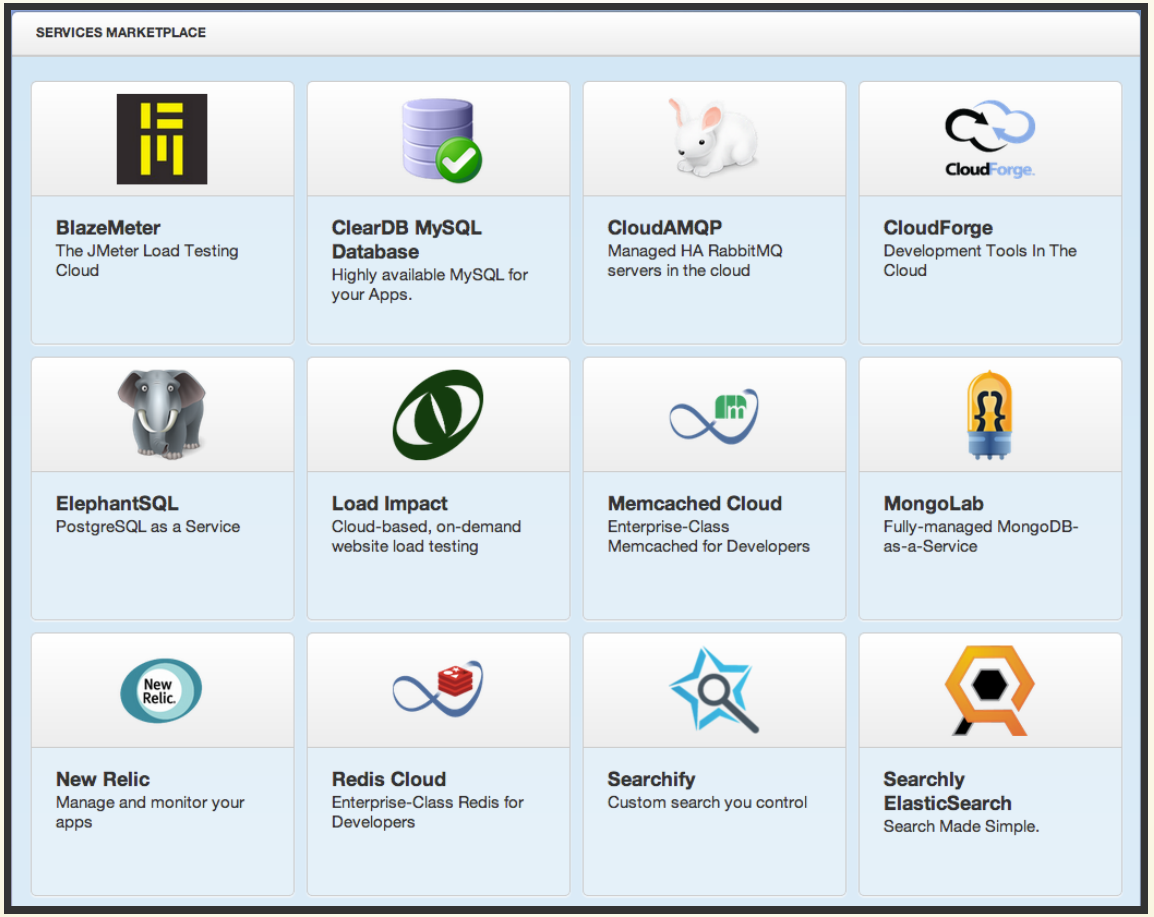
CLOUD FOUNDRY

OPEN PLATFORM-AS-A-SERVICE

- sponsored by Pivotal
- enjoys a vibrant community and ecosystem
  - http://cloudfoundry.org/about
- deployable to VMware vSphere, OpenStack, and AWS
  - additional infrastructure options in the works, including Google Compute Engine and Microsoft Azure

PIVOTAL CF HOSTED

- run.pivotal.io
- deployment of Cloud Foundry on AWS
- includes the Services Marketplace



PIVOTAL CF

- commercial enterprise distribution of Cloud Foundry
- deployable to VMware vSphere
  - other infrastructure options in the future
- simple wizard-based configuration and installation
- supports Pivotal One services
  - Pivotal HD
  - Pivotal Analytics
  - Pivotal RabbitMQ
  - MySQL

http://bit.ly/pivotal_cf_demo

LET'S PUSH AN APP

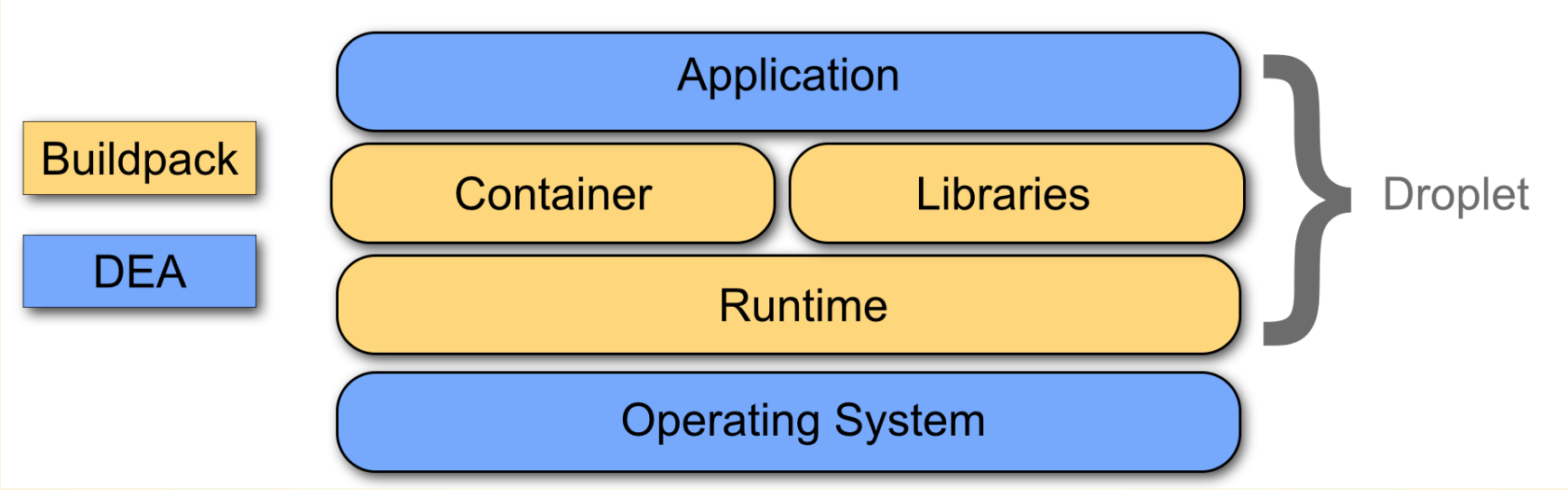
WHAT HAPPENED?

- an isolated container was created in a virtual machine
- the app was uploaded to the container
- a JRE was downloaded and installed
- Tomcat was downloaded, installed, and configured
- a MySQL database was created and exposed to the app

BUILDPACKS

Buildpacks are responsible for setting up the environment for an application to run in.

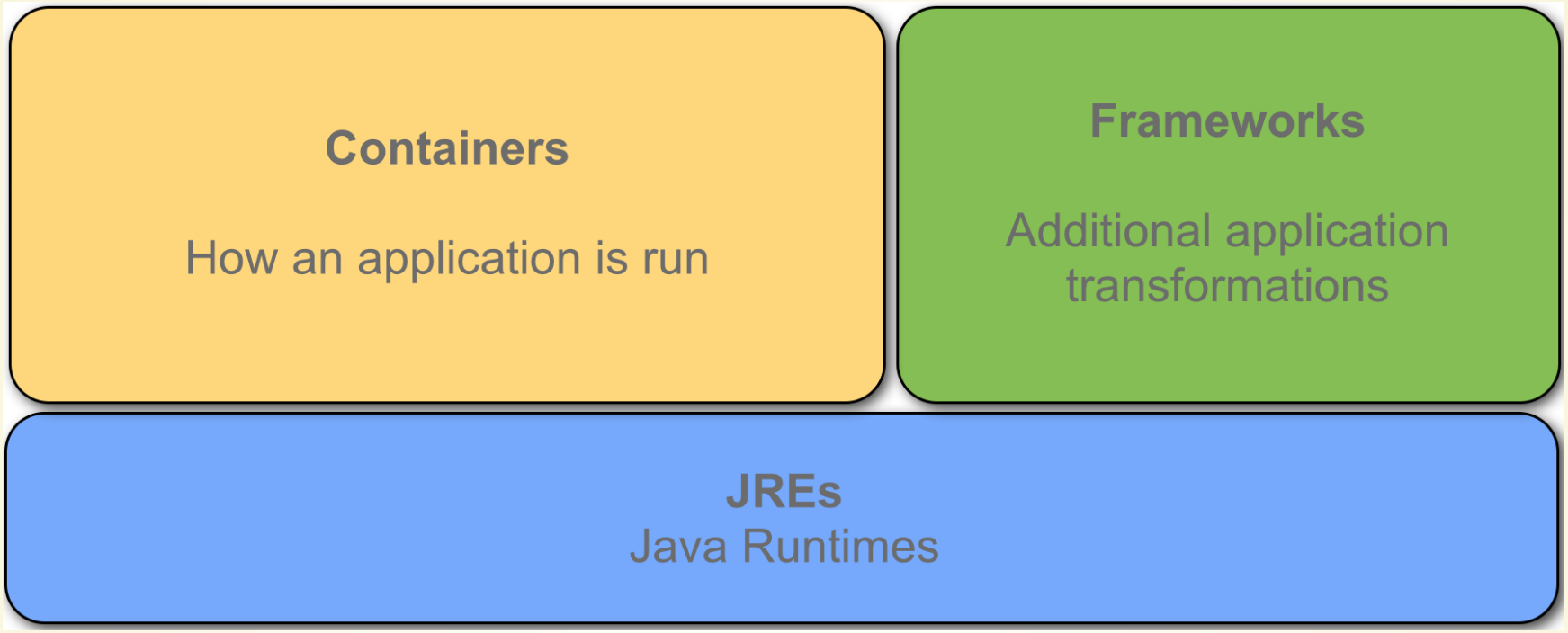
- including decisions about
  - runtimes - JVM, Ruby runtime, Node.js runtime
  - web containers - Tomcat, Jetty, OSGi
  - library and package management - Ruby gems, NPM packages



CLOUD FOUNDRY JAVA BUILDPACK

- easiest way to run Java applications
- automatic detection of application configuration based on inspection of application bits
- configurable and extensible by forking the GitHub repository
- zero-touch services integration

JAVA BUILDPACK CONCEPTS



JAVA BUILDPACK CONTAINERS

Supported containers and detection criteria

Java `main()`	`META-INF/MANIFEST.MF` exists with `Main-class` attribute set
Tomcat	`WEB-INF` directory exists
Groovy	`.groovy` file with a `main()` method, or `.groovy` file with no classes, or `.groovy` file with a shebang (`#!`) declaration
Spring Boot CLI	one or more POGO `.groovy` files with no `main()` method, and no `WEB-INF` directory
Play	`start` script and `lib/play.play_*.jar` exist

choose zero or one

LET'S SEE CONTAINERS IN ACTION

JAVA BUILDPACK FRAMEWORKS

Supported frameworks and detection criteria

Spring	`spring-core*.jar` exists
Play config	Play application detected
Play JPA config	play-java-jpa plugin exists in app
New Relic agent	New Relic service bound to app
AppDynamics agent	AppDynamics service bound to app

choose all that apply

LET'S SEE FRAMEWORKS IN ACTION

JAVA BUILDPACK CUSTOMIZATION

- customizations should be done for a large set of applications, not individual applications
  - reduces configuration drift
  - eases auditing of applications for security concerns

BUILDPACK CUSTOMIZATION BY CONFIGURATION

- some choices can be made in `config/*.yml` files
  - versions of JRE, Tomcat, other artifacts
  - repository locations
  - JRE memory configuration

LET'S CONFIGURE THE BUILDPACK

BUILDPACK CUSTOMIZATION BY EXTENSION

- extensions are additive
  - containers, frameworks, JREs
  - `config/components.yml`
- well-defined API
  - `initialize()`
  - `detect()`
  - `compile()`
  - `release()`

LET'S EXTEND THE BUILDPACK

JAVA TOOLING FOR CLOUD FOUNDRY

- Maven plugin
- Gradle plugin
- Java client library

CLOUD FOUNDRY MAVEN PLUGIN

```
<plugin>
  <groupId>org.cloudfoundry</groupId>
  <artifactId>cf-maven-plugin</artifactId>
  <version>1.0.1</version>
  <configuration>
    <server>mycloudfoundry-instance</server>
    <target>http://api.run.pivotal.io</target>
    <org>mycloudfoundry-org</org>
    <space>development</space>
    <appname>my-app</appname>
    <url>my-app.cfapps.io</url>
    <memory>512</memory>
  </configuration>
</plugin>
```

```
$ mvn package cf:push
```

https://github.com/cloudfoundry/cf-java-client/tree/master/cloudfoundry-maven-plugin

CLOUD FOUNDRY GRADLE PLUGIN

```
buildscript {
  dependencies {
    classpath group: 'org.cloudfoundry', name: 'cf-gradle-plugin', version: '
  }
}

apply plugin: 'cloudfoundry'

cloudfoundry {
  target = 'https://api.run.pivotal.io'
  organization = 'mycloudfoundry-org'
  space = 'development'
  application = 'my-app'
  file = new File("${war.archivePath}")
  uri = 'http://my-app.run.pivotal.io'
  memory = '512'
}
```

```
$ gradle assemble cf-push
```

https://github.com/cloudfoundry/cf-java-client/tree/master/cloudfoundry-gradle-plugin

CLOUD FOUNDRY JAVA CLIENT LIBRARY

```
CloudCredentials credentials = new CloudCredentials(user, password);
CloudFoundryClient client = new CloudFoundryClient(credentials, getTargetURL(
client.login();

System.out.println("\nSpaces:");
for (CloudSpace space : client.getSpaces()) {
    System.out.println(space.getName() + ":" + space.getOrganization().getNam
}

System.out.println("\nApplications:");
for (CloudApplication app : client.getApplications()) {
    System.out.println(app.getName());
}

System.out.println("\nServices");
for (CloudService service : client.getServices()) {
    System.out.println(service.getName() + ":" + service.getLabel());
}
```

http://docs.cloudfoundry.com/docs/using/managing-apps/libs/java-client.html

GO FORTH AND PUSH!