A NEW PLATFORM FOR A NEW ERA

Java Applications in the Cloud

Ben Hale, Cloud Foundry Java Experience

https://github.com/nebhale/sf-meetup-2014

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Cloud-friendly Applications

- Many (most ?) applications will eventually run in the cloud
- While this may be technically feasible, it does not mean that all applications are well-suited to it
- There is a continuum of suitability from “runs like it always has” to “thrives at cloud-scale”

- If applications *can* thrive, how do we help them do that?

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Disclaimer

- I’m a Spring guy
  - If you don’t love Spring Boot, you’re not looking hard enough
- Basic patterns are applicable to *all* application, Spring, Java or otherwise
- Implementation details are likely to be Spring and Spring Boot-specific
  - Replicate the functionality

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Cloud-friendly Applications

- State
- Configuration
- External Resources
- Serviceability
- Monitoring and Management
- Local Development

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

State is EVIL!

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

State is EVIL!

- State, especially shared state is the enemy of scalability
- Any coordination of state impacts performance
- Coordination of state across peer instances *really* impacts performance
- Java as a language (and you as a developer) aren't great at coordinating state
  - Leave it for applications (and developers) who live and breath it

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

State is EVIL!

- Making code immutable helps avoid state
  - Good rule of thumb is to question anything not marked `final`
- Obvious to push data back to data stores
  - Relational, Graph, Document, Key-Value

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

State is EVIL!

- Don’t forget files
- Amazing how many applications persist data to the filesystem
  - Looking at you Jenkins
- Even in cases where you *could* use/share a filesystem, don’t
  - Shared state impacts performance

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Configuration

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Configuration

- Many (most ?) applications read from myriad configuration files
- In a PaaS, it's difficult to get configuration files deployed with an application
  - Repackage deployment artifact
  - Must have original artifact for any configuration change

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Configuration

- Configuration *should* come from “the environment”
- For Java developers, “the environment” typically means system properties
- System properties are difficult to set in Cloud Foundry
  - `cf set-env JAVA_OPTS “-Dtest.property=test-value”`
  - Doesn’t even work before `cf` 6.1.2!

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Configuration

- “The environment” actually means environment variables
  - Java developers didn’t invent the PaaS, so we need to accommodate this
- `cf set-env TEST_PROPERTY test-value`
- `System.getenv(“TEST_PROPERTY”)`
  - Unexpected for Java users/developers

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Configuration

- Spring Boot Relaxed Binding
- No need for an exact match between desired properties and names
- Intuitive mapping between system properties and environment variables
  - `test.property <=> TEST_PROPERTY`
- `@Value("${test.property}") String testProperty`

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Service Binding

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Service Binding

- Most common configuration is service endpoint information
- Each environment needs a different configuration for the same service
  - How do you package for deployment?
  - How do you ensure that the package for each environment is identical
- Applications should depend on *pre-configured* services

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Service Binding

- Java Buildpack Auto-reconfiguration
  - Inspects `ApplicationContext` for beans that match exposed services
  - Transparently replaces bean definitions
- Reconfigures JDBC, Hibernate, JPA, MongoDB, RabbitMQ, and Redis
- Builds on Spring Cloud

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Service Binding

- Use Spring Cloud directly
- Get configured bean
  - `getSingletonServiceConnector(DataSource.class, null)`
- Get service information and configure bean
  - `new SimpleDriverDataSource(new Driver(), info.getJdbcUrl(), info.getUserName(), info.getPassword())`

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Serviceability

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Serviceability

- Many (most ?) applications write out to myriad log files
- Files, logging or otherwise, are pain to get the contents of
  - `cf files <APP-NAME> <PATH> | …`
- No integration with Cloud Foundry's logging
  - `cf log <APP-NAME> [--recent]`

© 2014 Pivotal Software, Inc. All rights reserved.

Serviceability

- Easiest application integration with logging is through `stdout` and `stderr`
  - Enhancements coming to allow direct library access

- Myriad file names become myriad eye catchers
  - `logging_controller.log => [LOGGING CONTROLLER]`

© 2014 Pivotal Software, Inc. All rights reserved.

Serviceability

- Cloud Foundry will write output to syslog endpoints
  - Splunk, Papertrail, etc.

- Uses a special kind of user-provided service
  - `cf cups <SERVICE> -l syslog://<HOST>:<PORT>`

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Monitoring and Management

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Monitoring and Management

- Logs and analysis only takes you so far
- Important to have real-time monitoring of applications
  - Uptime, performance, etc.

- Only a single inbound port is open to an application
  - No JMX connections
  - Can be hacked, but you shouldn't

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Monitoring and Management

- Services to the rescue!
- New Relic
  - SaaS
- AppDirect
  - SaaS or on-prem

- Open an issue if you want support for something new

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Monitoring and Management

- Spring Boot Metrics
  - Simple API for simple needs
  - Exposed via REST and JMX
- Jolokia
  - JMX over REST
  - `http://<HOST>/jolokia/read/<OBJECT-NAME>`

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Local Development, Cloud Production

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Local Deployment, Cloud Production

- No matter how fast your cloud is, it's not as fast as running locally
- Developers *should* do lots of work locally before pushing applications.
  - Not authoritative, but a good quick-turnaround option
- Therefore, applications **must** run both locally and in the cloud without modification

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Local Deployment, Cloud Production

- Always use runtimes that run equally locally and in the cloud
  - Servlet containers like Tomcat and Jetty start quickly
  - Spring Boot embeds container directly removing that variable
- Always use uniform and familiar configuration
  - Ideally, a relaxed binding like Spring Boot provides
  - Otherwise favor environment variables

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Local Deployment, Cloud Production

- Use Spring profiles to encapsulate conditional application configuration
- Spring profiles, specifically the “cloud” profile
-
```
@Profile(“cloud”)
@Configuration
public static class DataSourceConfiguration {
    @Bean
    DataSource dataSource() {…}
}
```

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Local Deployment, Cloud Production

- Run the buildpack locally
  - *(shh… my secret weapon)*
- Simple API
  - `bin/detect  <APP-DIR>`
  - `bin/compile <APP-DIR> <CACHE-DIR>`
  - `bin/release <APP-DIR>`

Pivotal

© 2014 Pivotal Software, Inc. All rights reserved.

Cloud-friendly Applications

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

Cloud-friendly Applications

- Statelessness leads to scalability
- No filesystem access
- Configuration via the environment
- Connection information via services
- Console-based logging
- Monitoring via services and frameworks
- Local development, cloud production

Pivotal™

© 2014 Pivotal Software, Inc. All rights reserved.

A NEW PLATFORM FOR A NEW ERA