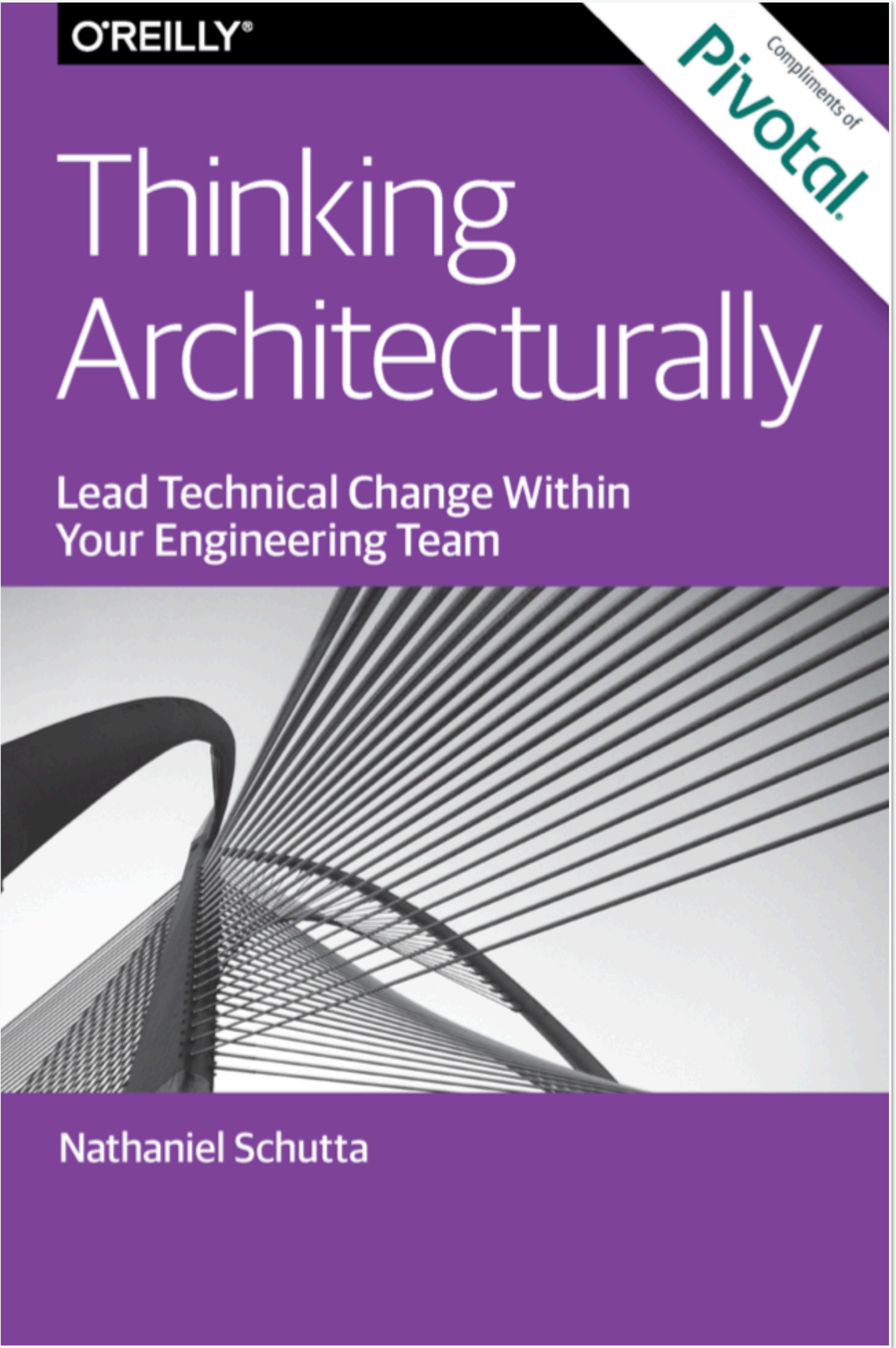
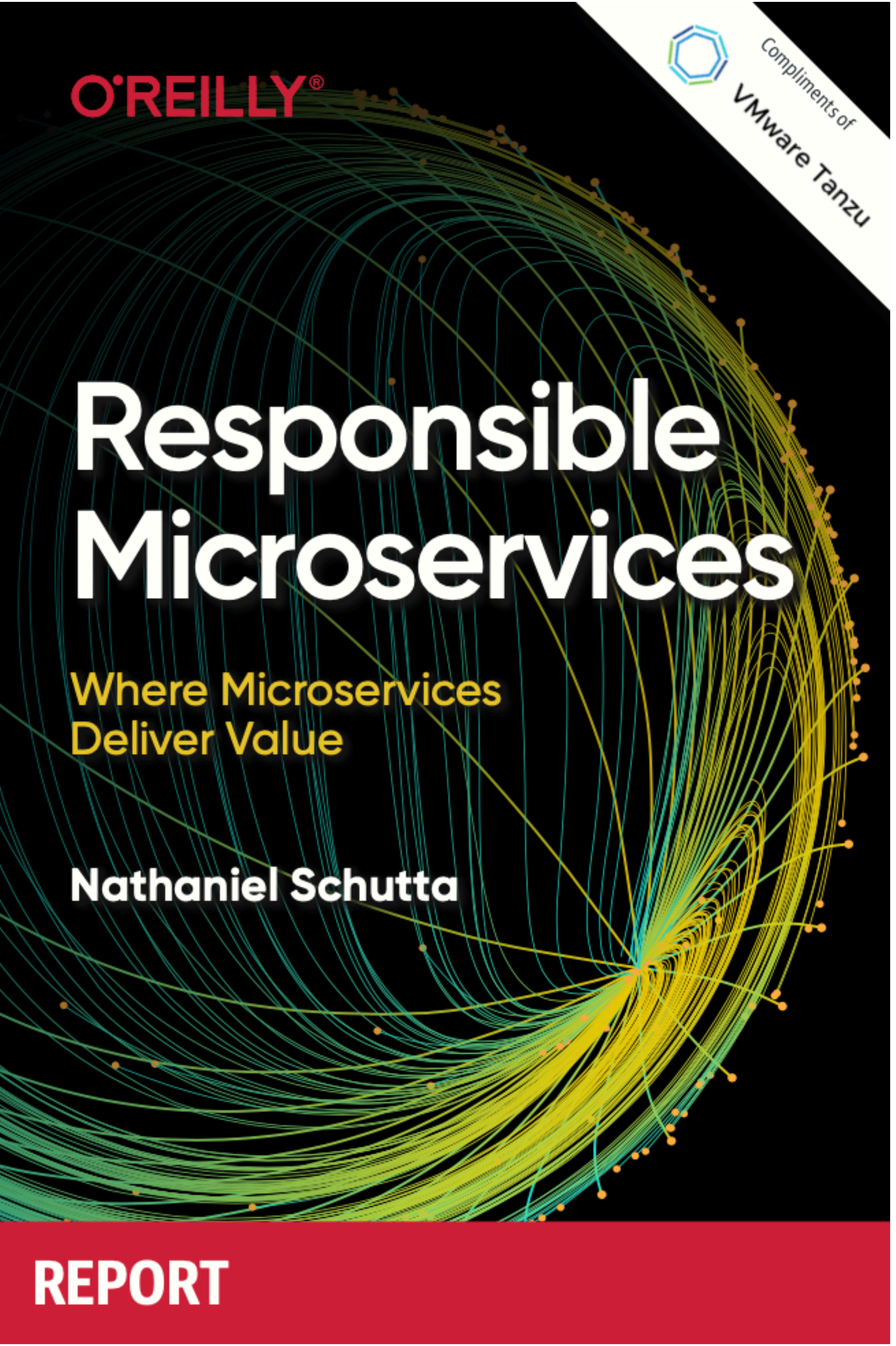
# Spring for Architects

Nathaniel Schutta
@ntschutta
ntschutta.io

Jakub Pilimon
@jakubpilimon


O'REILLY
Compliments of Pivotal
Thinking Architecturally
Lead Technical Change Within Your Engineering Team
Nathaniel Schutta

https://tanzu.vmware.com/content/ebooks/thinking-architecturally



https://tanzu.vmware.com/content/ebooks/responsible-microservices-ebook

It used to be so simple.

You had a monolith. Maybe two.

You released new versions semi annually.

Your team all worked on the same floor.

Or at least within walking distance.

But that isn’t the case today is it?

Now you have dozens, hundreds… maybe thousands of services.

New versions drop daily.

Your team is scattered around the globe.

Architecting was never easy!

Now? Massively distributed apps with geographically dispersed teams.

We are spread thin.

Can’t be everywhere at all times!

We can’t be involved with every decision.

We must empower our teams.

## Distributed decision making.

We can establish principles.

Leverage the power of defaults.

jstor.org

Have library access? Log in through your library

JSTOR

Search JSTOR

Log in | Register

Advanced Search   Browse   Tools

About   Support

JOURNAL ARTICLE

# The Power of Suggestion: Inertia in 401(k) Participation and Savings Behavior

**Brigitte C. Madrian and Dennis F. Shea**

The Quarterly Journal of Economics
Vol. 116, No. 4 (Nov., 2001), pp. 1149-1187 (39 pages)
Published By: Oxford University Press

https://www.jstor.org/stable/2696456

Cite this Item

**Read and download**
Log in through your school or library

**Alternate access options**
For independent researchers

## Abstract

This paper analyzes the impact of automatic enrollment on 401(k) savings behavior. We have two key findings. First, 401(k) participation is significantly higher under automatic enrollment. Second, a substantial fraction of 401(k) participants hired under automatic enrollment retain both the default contribution rate and fund allocation even though few employees hired before automatic enrollment picked this particular outcome. This "default" behavior appears to result from participant inertia and from employee perceptions of the default as investment advice. These findings have implications for the design of 401(k) savings plans as well as for any type of Social Security reform that includes personal accounts over which individuals have control. They also shed light more generally on the importance of both economic and noneconomic (behavioral) factors in the determination of individual savings behavior.

## Journal Information

The Quarterly Journal of Economics (QJE) is the oldest professional journal of economics in the English language. Edited at Harvard University's Department of Economics, it covers all aspects of the field -- from the journal's traditional emphasis on microtheory, to both empirical and theoretical macroeconomics. QJE is invaluable to professional and academic economists and students around the world.

THE
QUARTERLY JOURNAL
OF ECONOMICS

Vol. CXVI    November 2001    Issue 4

THE POWER OF SUGGESTION: INERTIA IN 401(k)
PARTICIPATION AND SAVINGS BEHAVIOR*

BRIGITTE C. MADRIAN AND DENNIS F. SHEA

This paper analyzes the impact of automatic enrollment on 401(k) savings behavior. We have two key findings. First, 401(k) participation is significantly higher under automatic enrollment. Second, a substantial fraction of 401(k) participants hired under automatic enrollment retain both the default contribution rate and fund allocation even though few employees hired before automatic enrollment picked this particular outcome. This "default" behavior appears to result from participant inertia and from employee perceptions of the default as investment advice. These findings have implications for the design of 401(k) savings plans as well as for any type of Social Security reform that includes personal accounts over which individuals have control. They also shed light more generally on the importance of both economic and noneconomic (behavioral) factors in the determination of individual savings behavior.

I. INTRODUCTION

In this paper we analyze the 401(k) savings behavior of employees in a large U. S. corporation before and after an interesting change in the company 401(k) plan. Before the plan change, employees who enrolled in the 401(k) plan were required to affirmatively elect participation. After the plan change, employees were automatically enrolled in the 401(k) plan immedi-

* Research support from the National Institute on Aging is gratefully acknowledged. A special thanks to Hewitt Associates for their help in providing the data. Thanks also to Richard Thaler, Anna Lusardi, Amil Petrin, Judith Chevalier, Kara Anderson, and David Docherty for helpful discussions. Comments from seminar participants at the University of North Carolina, Brigham Young University, Western Michigan University, the University of Chicago, the University of Wisconsin, Harvard University, the Massachusetts Institute of Technology, the U. S. Bureau of Labor Statistics, and the National Bureau of Economic Research are also appreciated.

*The Quarterly Journal of Economics*, November 2001

1149

View Preview

Behavioral Economics…

Powerful enough to earn
Richard Thaler a Noble Prize.

“…if you want to get somebody to do something, make it easy.”

–Richard Thaler

https://www.mckinsey.com/industries/public-and-social-sector/our-insights/nudging-the-world-toward-smarter-public-policy-an-interview-with-richard-thaler

Use that to *our* advantage.

Wait? What?

Architects must wield the power of defaults.

Make the right choice
the easy choice.

Distributed systems
have similar needs.

Monitoring. Circuit breakers.
Consumer Driven Contracts.

Gateways. Streams.
Externalized configuration.

Functions. Service discovery. Load balancing. Documentation.

We can’t afford to reinvent the wheel on every project.

Spring frees architects to focus on critical design decisions.

While empowering teams to
solve critical business problems.

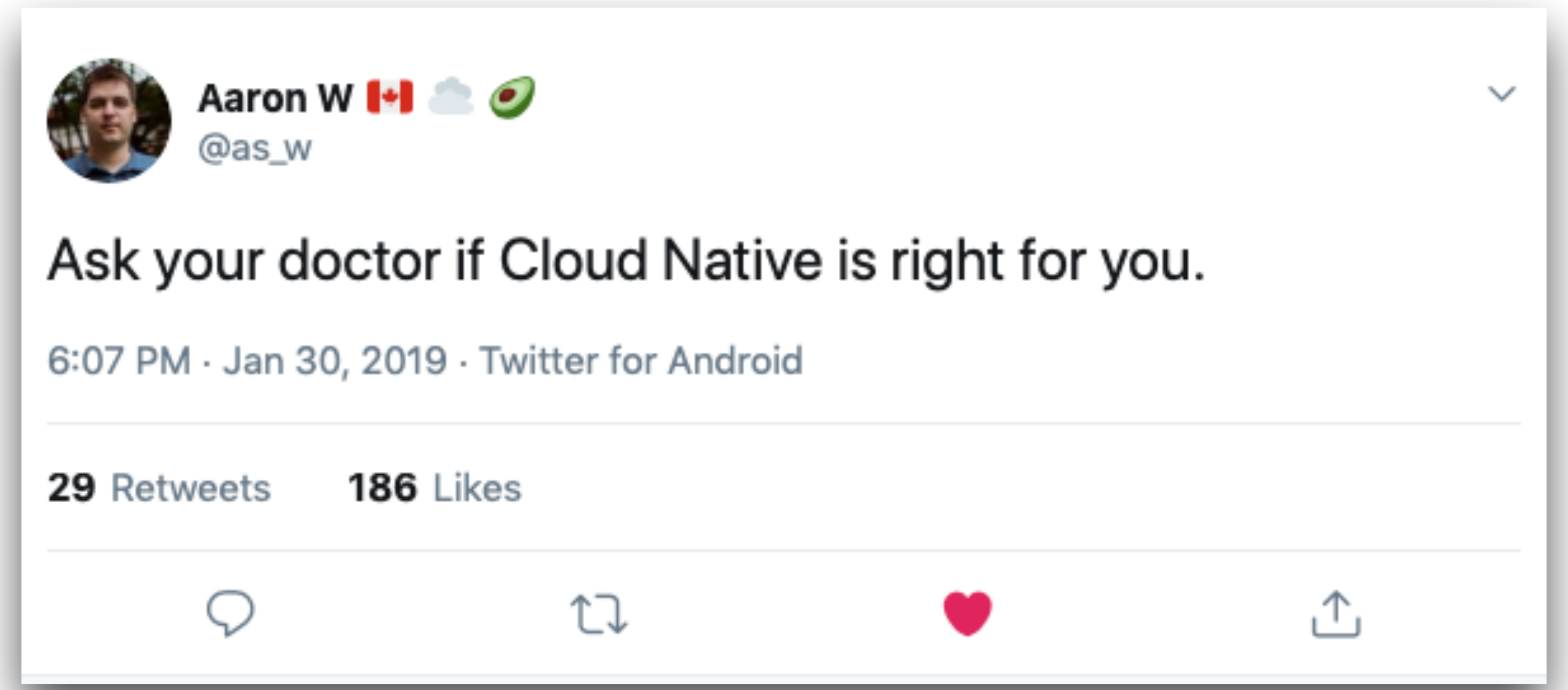
There are many ways to fail with distributed applications.

Spring is here to help you, your teams and your applications.

Help everyone sleep better at night.

Ultimately, it is about delivering business value to production.

## What is Cloud Native?


Aaron W 🇨🇦 ☁️ 🥑
@as_w

Ask your doctor if Cloud Native is right for you.

6:07 PM · Jan 30, 2019 · Twitter for Android

**29** Retweets **186** Likes

https://mobile.twitter.com/as_w/status/1090763452241534976

Applications designed to take advantage of cloud computing.

Fundamentally about how we create and deploy applications.

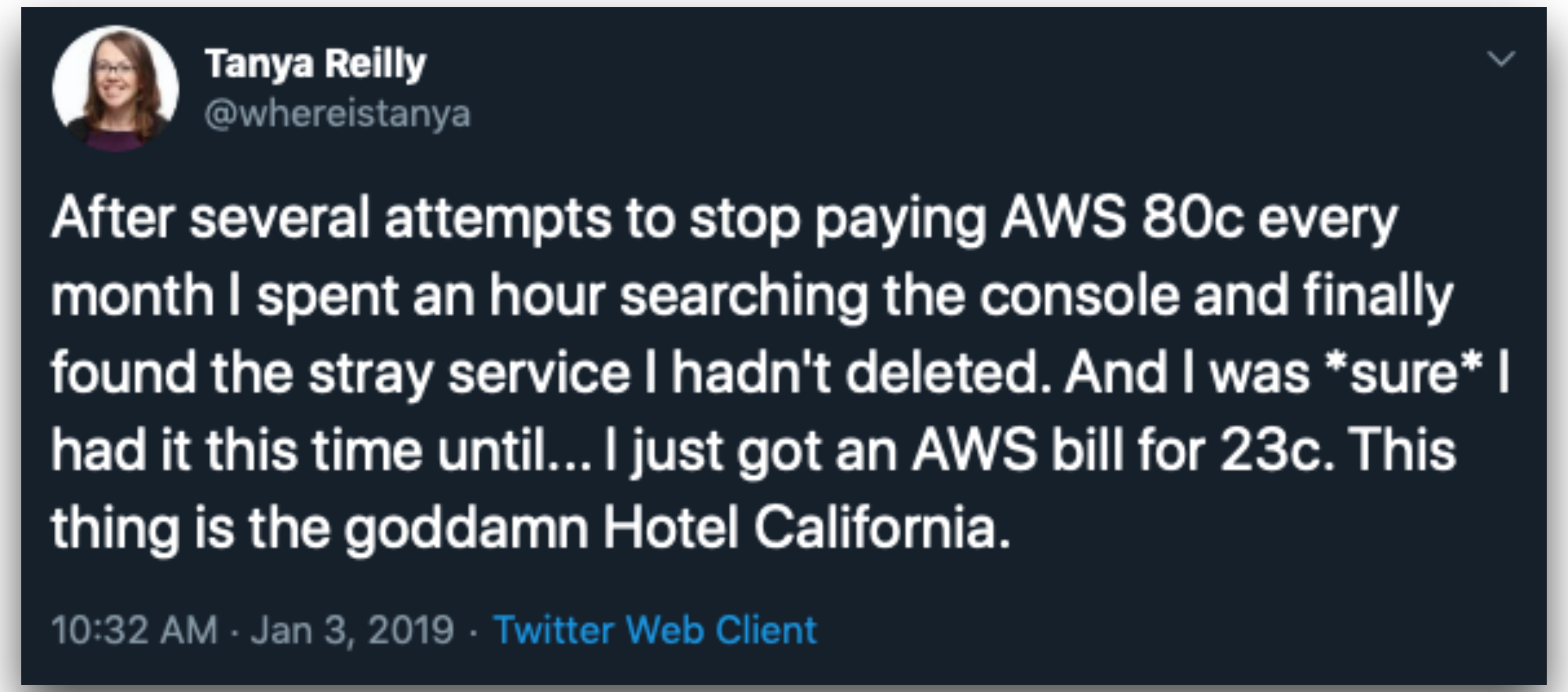
Cloud computing gives us some very interesting abilities.

Scale up. Scale down. On demand.

Limitless compute.*

* Additional fees may apply.

Said fees can be…opaque.

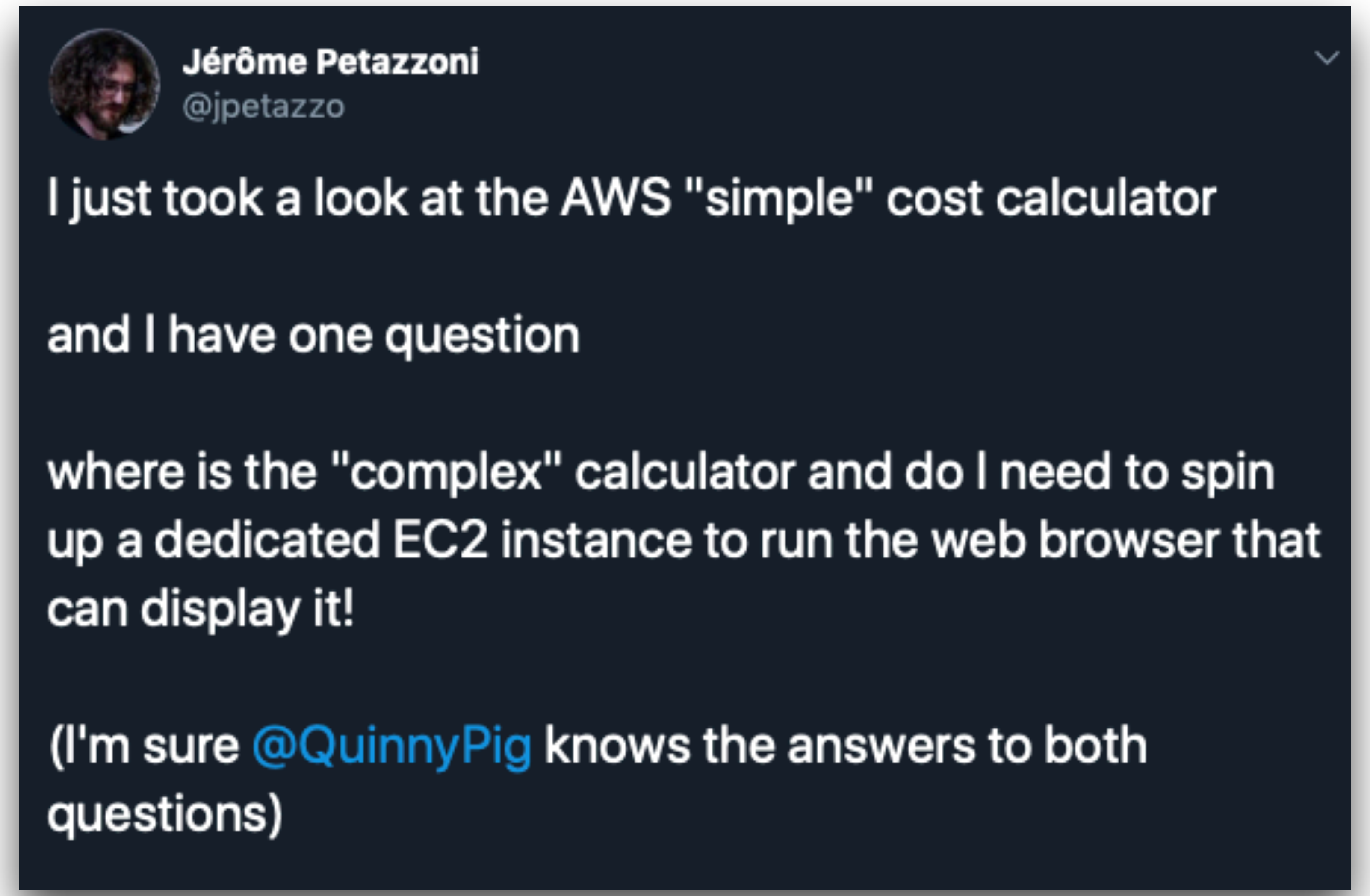


Tanya Reilly
@whereistanya

After several attempts to stop paying AWS 80c every month I spent an hour searching the console and finally found the stray service I hadn't deleted. And I was *sure* I had it this time until... I just got an AWS bill for 23c. This thing is the goddamn Hotel California.

10:32 AM · Jan 3, 2019 · Twitter Web Client

https://mobile.twitter.com/whereistanya/status/1080864493108776961


Jérôme Petazzoni
@jpetazzo

I just took a look at the AWS "simple" cost calculator

and I have one question

where is the "complex" calculator and do I need to spin up a dedicated EC2 instance to run the web browser that can display it!

(I'm sure @QuinnyPig knows the answers to both questions)

https://mobile.twitter.com/jpetazzo/status/1227638126602080256

calculator.s3.amazonaws.com

SIMPLE MONTHLY CALCULATOR

Language: English

Need Help? Watch the Videos or Read How AWS Pricing Works or Contact Sales

Get Started with AWS: Learn more about our Free Tier or Sign Up for an AWS Account »

[✓] FREE USAGE TIER: New Customers get free usage tier for first 12 months

Reset All

Services | Estimate of your Monthly Bill ($ 0.00)

Choose region: US East (N. Virginia)

Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

Amazon EC2
Amazon S3
Amazon Route 53
Amazon CloudFront
Amazon RDS
Amazon Elastic Load Balancing
Amazon DynamoDB
Amazon ElastiCache
Amazon CloudWatch
Amazon SES
Amazon SNS
Amazon Elastic Transcoder
Amazon WorkSpaces
Amazon WorkDocs
AWS Directory Service
Amazon Redshift
Amazon Glacier
Amazon SQS
Amazon SWF
Amazon Elastic MapReduce
Amazon Kinesis Streams
Amazon CloudSearch
AWS Snowball
AWS Direct Connect
Amazon VPC
Amazon SimpleDB

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances.

Clear Form

**Compute: Amazon EC2 Instances:**

| | Description | Instances | Usage | Type | Billing Option | Monthly Cost |
|---|---|---|---|---|---|---|
| ⊕ | Add New Row | | | | | |

**Compute: Amazon EC2 Dedicated Hosts:**

| | Description | Number of Hosts | Usage | Type | Billing Option |
|---|---|---|---|---|---|
| ⊕ | Add New Row | | | | |

**Storage: Amazon EBS Volumes:**

| | Description | Volumes | Volume Type | Storage | IOPS | Baseline Throughput | Snapshot Storage |
|---|---|---|---|---|---|---|---|
| ⊕ | Add New Row | | | | | | |

**Compute: Amazon Elastic Graphics:**

| | Description | Number of Elastic Graphics | Usage | Elastic Graphics Size and Memory |
|---|---|---|---|---|
| ⊕ | Add New Row | | | |

**Additional T2/T3 Unlimited vCPU Hours per month:**

| | |
|---|---|
| For Linux, RHEL and SLES: | 0 |
| For Windows and Windows with SQL Web: | 0 |

**Elastic IP:***

[✓ Enter values below] Calculate

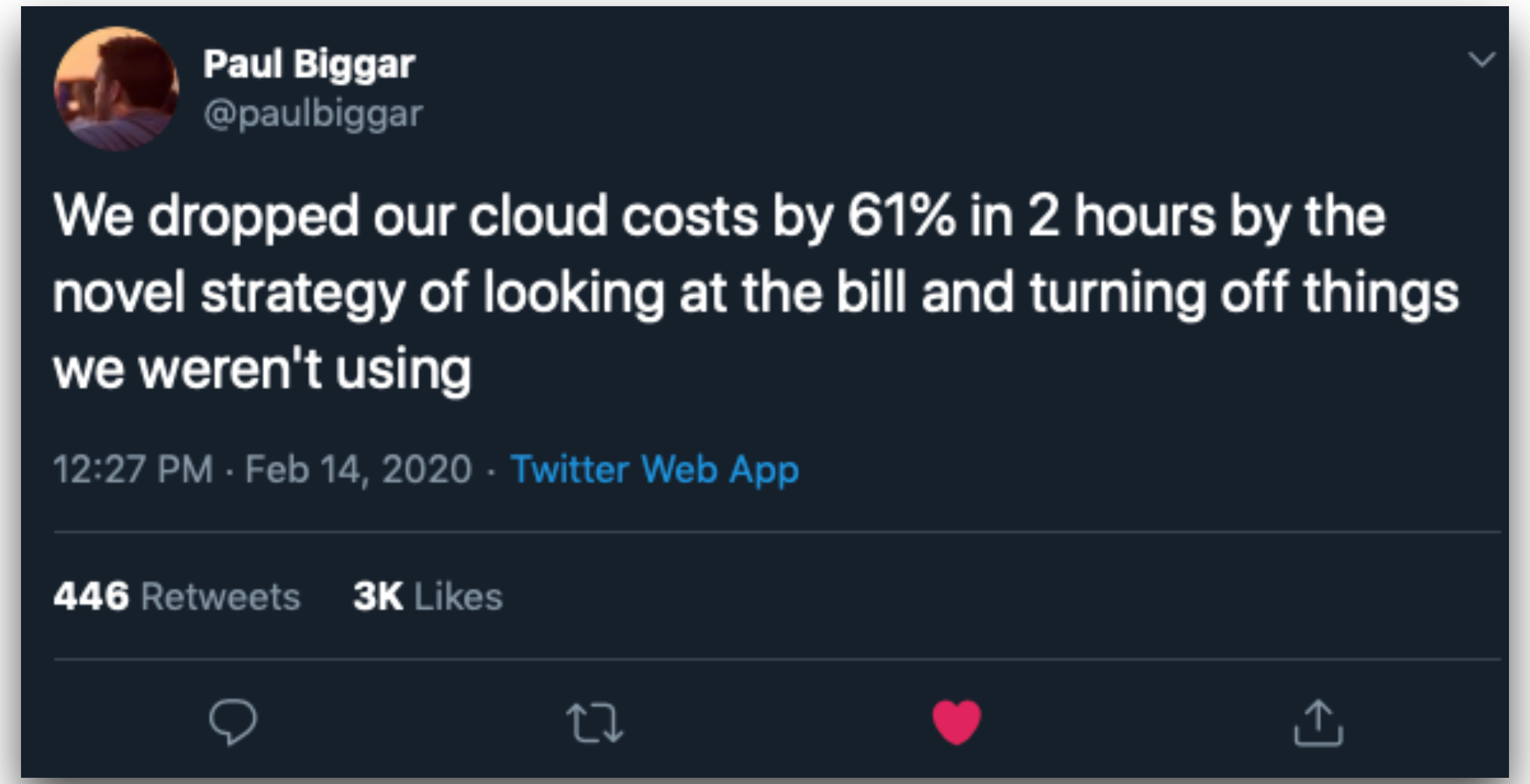
| | | |
|---|---|---|
| Total time the additional Elastic IPs are attached to running EC2 instances**: | 0 | Hours/Month |
| Total Non-attached time for all the Elastic IPs: | 0 | Hours/Month |
| Number of Elastic IP Remaps: | 0 | Per Month |

**Data Transfer:**

| | | |
|---|---|---|
| Inter-Region Data Transfer Out: | 0 | GB/Month |
| Data Transfer Out: | 0 | GB/Month |
| Data Transfer In: | 0 | GB/Month |

Common Customer Samples

Free Website on AWS
AWS Elastic Beanstalk Default
Marketing Web Site
Large Web Application (All On-Demand)
Media Application
European Web Application
Disaster Recovery and Backup


Paul Biggar
@paulbiggar

We dropped our cloud costs by 61% in 2 hours by the novel strategy of looking at the bill and turning off things we weren't using

12:27 PM · Feb 14, 2020 · Twitter Web App

446 Retweets 3K Likes

https://mobile.twitter.com/paulbiggar/status/1228385370439467009

Cloud native isn’t just an architectural pattern.

Combination of practices, techniques, technologies.

## Agile development.

## Continuous delivery.

## Automation.

## Containers.

Microservices.

## Functions.

Changes our culture.

DevOps.

Infrastructure is a different game today isn’t it?

We’ve seen this massive shift.

Servers used to be home grown.

Bespoke. Artisanal.

Spent days hand crafting them.

Treated them like pets…

Did whatever it took to keep them healthy and happy.

Servers were a heavily constrained resource.

They were really expensive!

Had to get our money’s worth…

Thus was born app servers.

Put as many apps as possible on a server.

Maximize the return on investment.

But that has some unintended side effects.

Shared resources.

One application’s bug could take down multiple apps.

Coordinating changes hurts.

"Your app can't get this feature until all other apps are ready."

Currency === 18 months of freezes, testing, frustration.

Organizations ignored currency issues…pain wasn’t “worth it”.

“Fear is the path to the dark side. Fear leads to anger. Anger leads to hate. Hate leads to suffering.”

–Yoda

#YodaOps

Move code from one server to another…

## Worked in dev…but not test.

Why?!?

The environments are
the same…right?

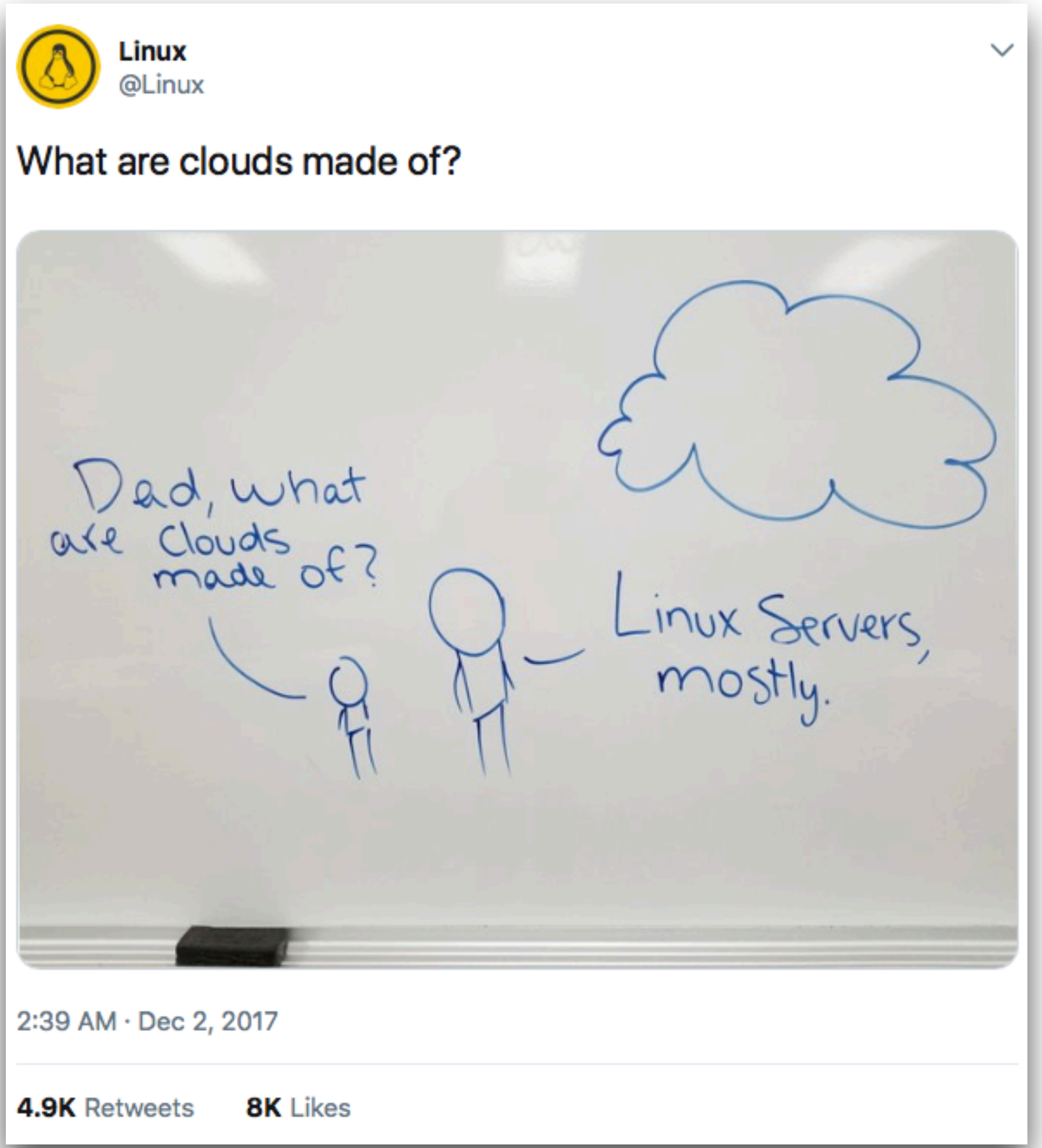
“Patches were applied in a different order…”

## Can I change careers?

Things started to change.

Servers became commodities.

Linux and Intel chips replaced custom OS on specialized silicon.



Linux
@Linux

What are clouds made of?

2:39 AM · Dec 2, 2017

4.9K Retweets 8K Likes

https://mobile.twitter.com/linux/status/936877536780283905?lang=en

Prices dropped.

Servers were no longer the constraining factor.

People costs eclipsed hardware costs.

Heroku, AWS, Google App Egine, Cloud Foundry, Azure.

Shared servers became a liability.

Treat them like cattle…when they get sick, get a new one.

15

New abstractions.

Containers and PaaS changed the game.

Package the app up with everything it needs.

Move *that* to a different environment.

Works in dev? You're testing the exact same thing in test.

So. Much. Win.

Your app needs a spiffy new library? Go ahead!

It doesn’t impact any other app because you are isolated.

Moves the value line.

Less “undifferentiated heavy lifting”.

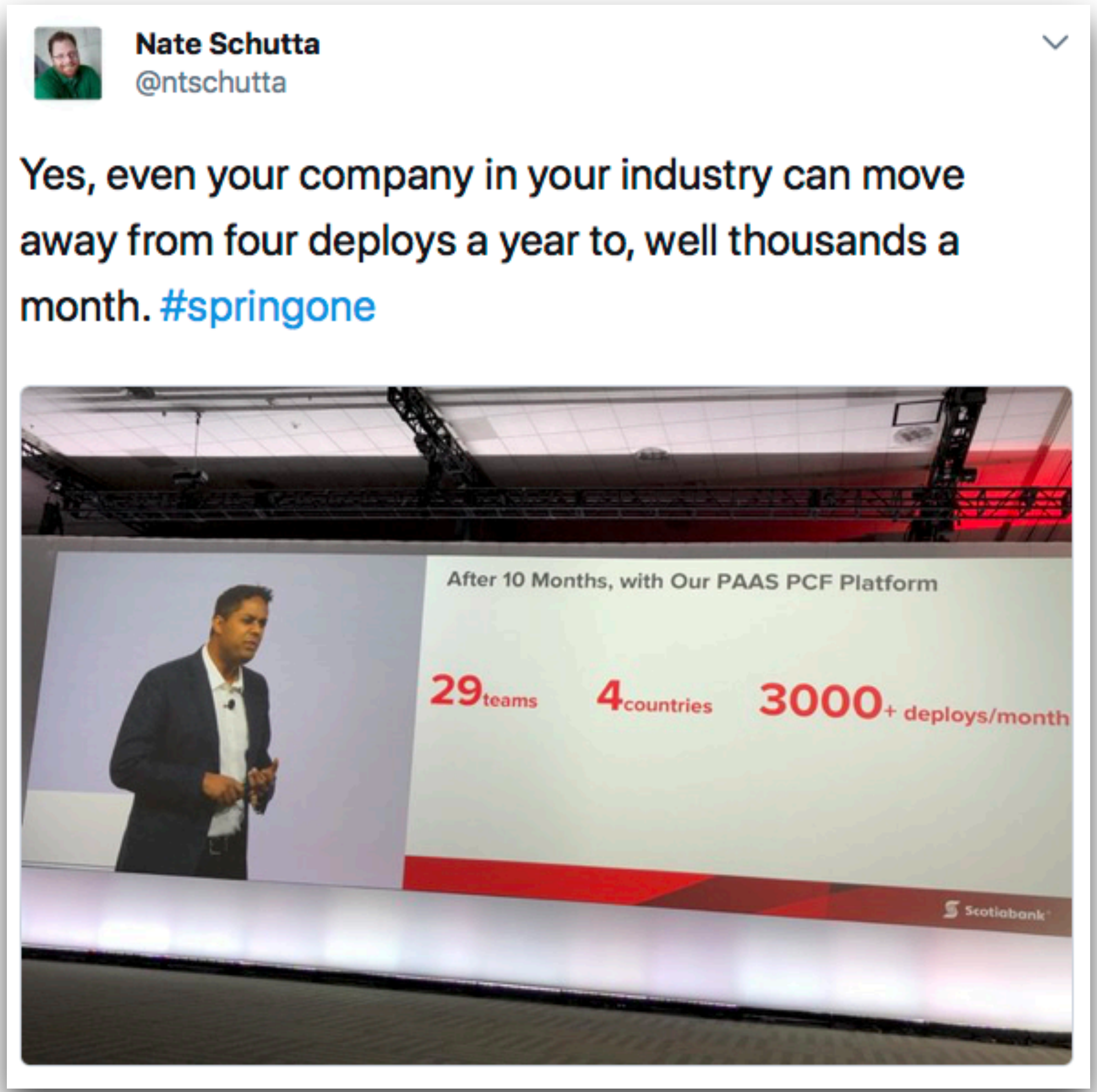
Changes development.

Always be changing.

Run experiments. A/B testing.

Respond to business changes.

Deliver in days not months.



Nate Schutta
@ntschutta

Yes, even your company in your industry can move away from four deploys a year to, well thousands a month. #springone

https://mobile.twitter.com/ntschutta/status/938109379995353088

Speed matters.

Disruption impacts every business.

Your industry is not immune.

Amazon Prime customers can order from Whole Foods.

Some insurance companies view Google as a competitor.

We're all technology companies today.

# 12 FACTORS

# Twelve Factor App.

https://12factor.net

## Characteristics shared by successful apps.

At least at Heroku.

1. One codebase in version control, multiple deploys.
2. Explicitly define your dependencies.
3. Configuration must be separate from the code.
4. Backing services are just attached resources.
5. Build, release, run.

6. Stateless - share nothing.
7. Export services via port binding.
8. Scale via process.
9. Start up fast, shut down gracefully.
10. Dev/Prod parity.
11. Logs as event streams.
12. Admin tasks run as one off processes.

## I. One codebase in version control, multiple deploys.

Version control isn’t controversial. Right?!?

Sharing code? It better be in a library then…

## II. Explicitly define your dependencies.

Do not rely on something just “being there” on the server.

If you need it, declare it.

III. Configuration must be separate from the code.

The things that vary from environment to environment.

Could you open source that app right now?

## IV. Backing services are just attached resources.

Should be trivial to swap out a local database for a test db.

In other words, loose coupling.

## V. Build, release, run.

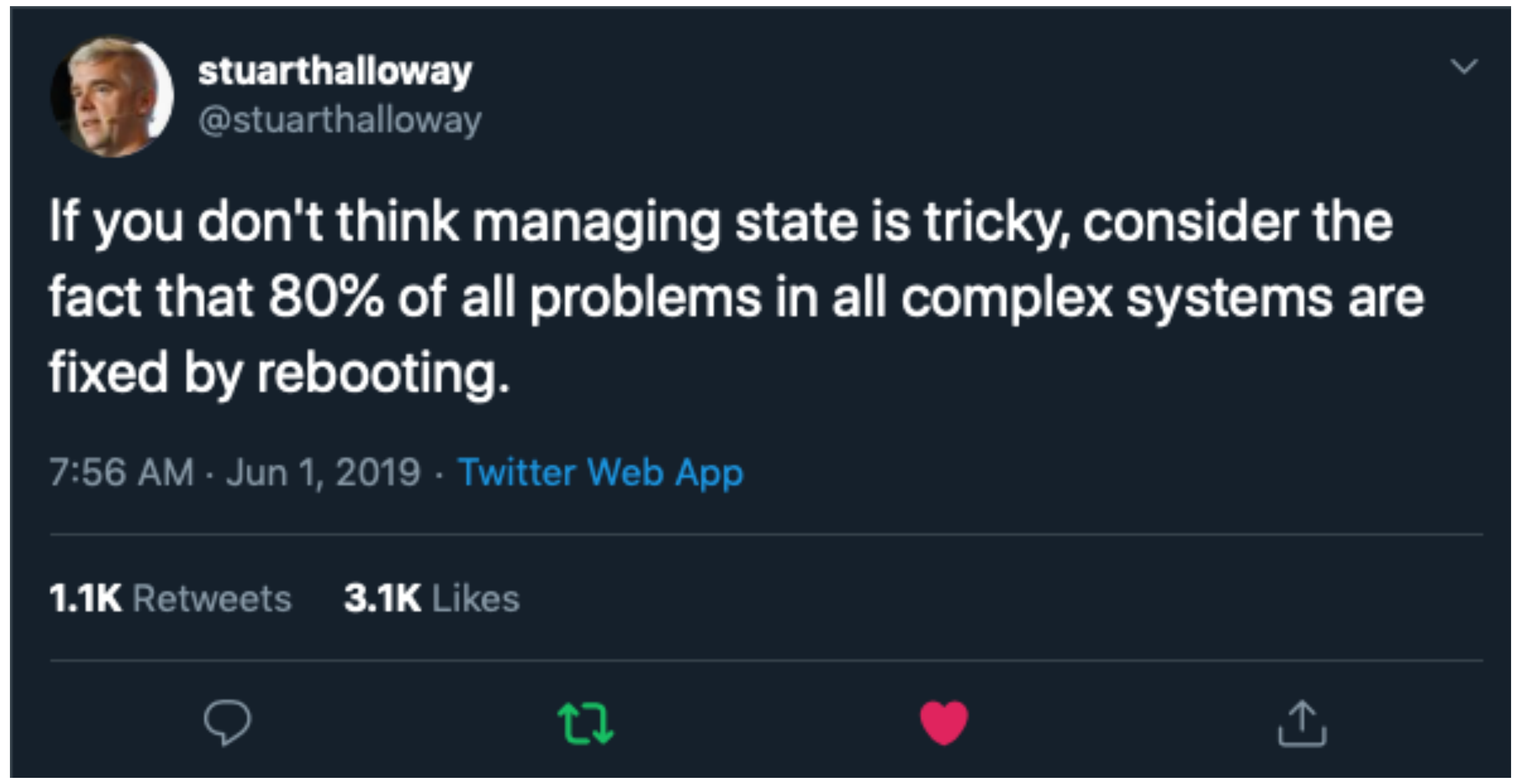
Deployment pipeline anyone?

## Build the executable…

Deploy the executable with the proper configuration…

Launch the executable in a given environment.

## VI. Stateless - share nothing.


stuarthalloway
@stuarthalloway

If you don't think managing state is tricky, consider the fact that 80% of all problems in all complex systems are fixed by rebooting.

7:56 AM · Jun 1, 2019 · Twitter Web App

1.1K Retweets 3.1K Likes

https://mobile.twitter.com/stuarthalloway/status/1134806008528809985

State must be stored via some kind of backing service.

In other words, you cannot rely on the filesystem or memory.

Recovery. Scaling.

## VII. Export services via port binding.

App exports a port, listens for incoming requests.

Localhost for development,
load balancer for public facing.

## VIII. Scale via process.

In other words, scale horizontally.

## IX. Start up fast, shut down gracefully.

Processes aren’t pets,
they are disposable.

Processes can be started (or stopped) quickly and easily.

Ideally, start up is seconds.

Also can handle unexpected terminations!

## X. Dev/Prod parity.

From commit to production should be hours…maybe days.

Definitely not weeks.

Developers should be involved in deploys and prod ops.

Regions should be identical. Or as close as possible to identical.

Backing services should be the same in dev and prod.

Using one DB in dev and another in prod invites pain.

## XI. Logs as event streams.

Don’t write logs to the filesystem!

It won’t be there later…

Write to stdout.

Stream can be routed any number of places.

And then consumed via a wide variety of tools.

## XII. Admin tasks run as one off processes.

Database migrations for instance.

## REPL for the win.

Run in an identical environment to the long running processes.

Your legacy apps will violate some factors.

Maybe all 12!

In general…

## II. Explicitly define your dependencies.

Probably one of the harder ones to satisfy.

Do we really need this library?

"It works, don't touch it."

III. Configuration must be separate from the code.

Many an app has hardcoded credentials.

Hardcoded database connections.

## VI. Stateless - share nothing.

Also can be challenging.

Many apps were designed around a specific flow.

Page 2 left debris for Page 3!

“Just stash that in session”.

## IX. Start up fast, shut down gracefully.

Many apps take way
too long to start up…

Impacts health checks.

## X. Dev/Prod parity.

Environments should be consistent!

Shorten code to prod cycle.

"It worked in test…"

Do your applications have to be fully 12 factor compliant?

Nope.

## Is it a good goal?

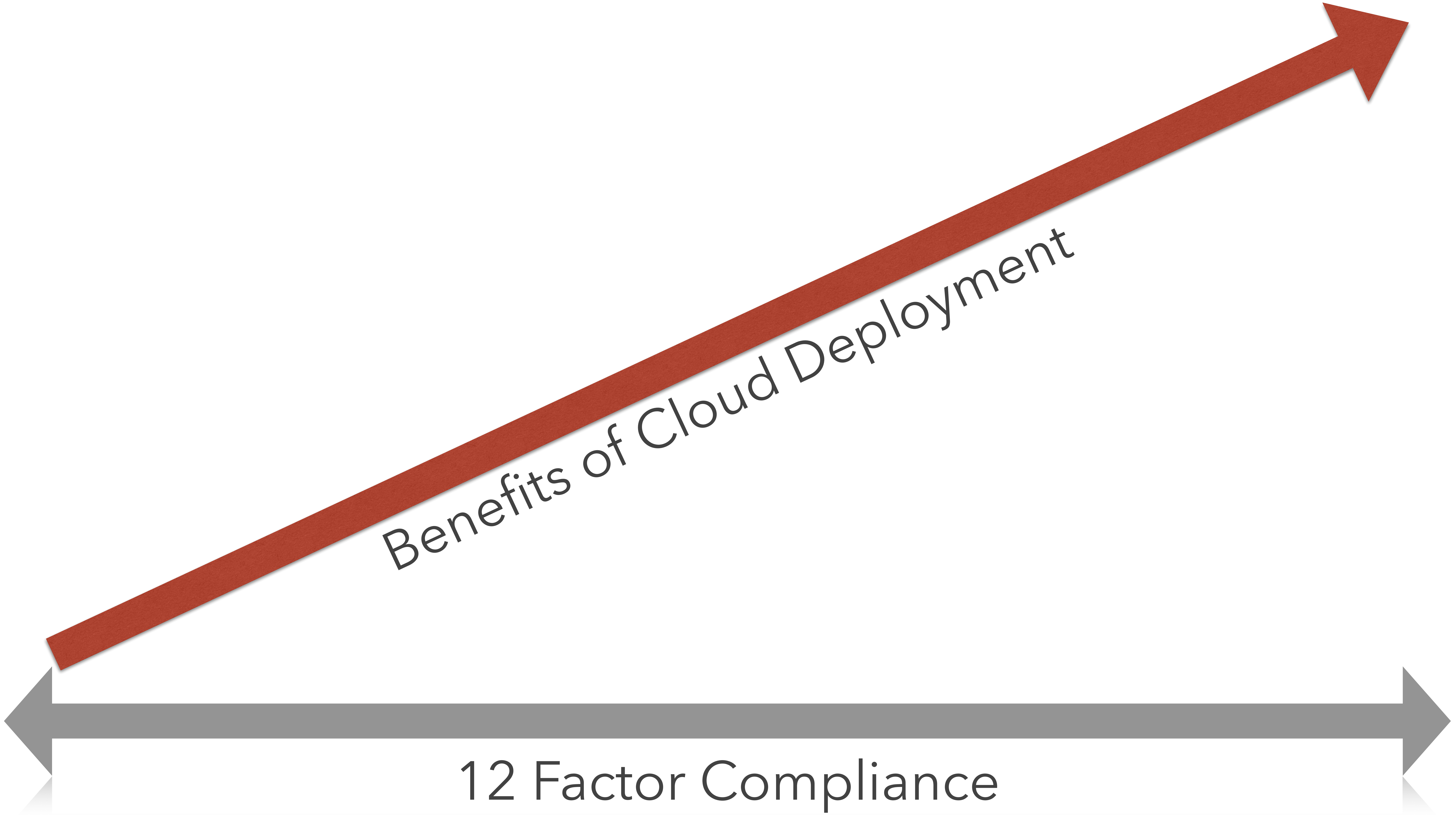
Sure.

But be pragmatic.

Certain attributes lessen the advantages of cloud.

Long startup time hurts elastic scaling & self healing.

Think of it as a continuum.


Benefits of Cloud Deployment
12 Factor Compliance

Developers also talk about 15 factor apps.

aka Beyond the Twelve-Factor App.

https://content.pivotal.io/blog/beyond-the-twelve-factor-app

However you define it…

To maximize what
the cloud gives us…

Applications need to be designed properly.

Legacy applications will fall short.

## Opportunistically refactor!

## Building greenfield?

## Go cloud native!

Don’t build legacy.

## What is Spring?

## Integration framework.

Combines a lot of different things together.

Consistent programming model.

Simplify Java development.

Supports other JVM languages,
Kotlin and Groovy.

Family of projects built atop the Spring Framework.

Provides support for any number of application architectures.

Message driven. Web applications.
Reactive. Microservices.

Spring provides choices.

Want to switch out your message broker? No problem.

Time for a different datastore?

No worries.

Backwards compatible.

Range of JDK versions,
minimize breaking changes.

Thoughtful APIs.

## User centered API design.

High code quality.

Clean code with top notch documentation.

Mature - first release in mid 2003.

Twitter, Inc.

Thread

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
It's said that "senior" developers tend to be afraid of learning new things. This has always struck me as absurd, because who hasn't learned a ton of new things over the course of 10+ years of programming? Almost impossible not to.
4 5 18

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
It almost feels as though less experienced developers are projecting their own fears onto others with more experience.
1 1 8

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
I get the sense that many programmers are afraid to learn *old* things. As if the older things are impossibly arcane and complex, compared to the supposed simplicity and elegance of new tech.
1 6 18

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
They have a strong preference for the new. They'd rather use unfinished, buggy, unproven technology than older, refined, stable, tested tech. Even if it reinvents or indeed breaks the wheel.
2 7 15

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
So now there's a of excitement about a "UnicornKit" that combined UIKit and AppKit into one. But do people realize that unless UnicornKit is simply UIKit, you're going to have to learn something else anyway?
1 3 9

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
Why is learning UnicornKit harder than learning AppKit? Many developers have successfully written both Mac and iOS apps. It's not as hard as you may think if you have no experience with AppKit.
2 3 8

Jeff Johnson (Mac/iOS developer for hire) @lapcats... · Dec 21
Also, any new technology is going to be chock full of bugs. AppKit has its bugs, yes, but it's assuredly vastly more reliable and stable than UnicornKit

Built to simplify JEE development.

Deliver the promise of EJB without…the overhead.

POJO - Plain Old Java Object.

## Inversion of Control aka Dependency Injection.

## Loose coupling.

## Declarative programming.

## Eliminate boilerplate.

Has grown considerably…

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community

Spring Boot
Spring Framework
Spring Data
Spring Cloud
Spring Cloud Data Flow
Spring Security
Spring Session
Spring Integration
Spring HATEOAS
Spring REST Docs
Spring Batch
Spring AMQP
Spring for Android
Spring CredHub
Spring Flo
Spring for Apache Kafka
Spring LDAP
Spring Mobile
Spring Roo
Spring Shell
Spring Statemachine
Spring Vault
Spring Web Flow
Spring Web Services

# Spring Boot 2.4.1

OVERVIEW
LEARN
SAMPLES

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

If you're looking for information about a specific version, or instructions about how to upgrade from an earlier release, check out the project release notes section on our wiki.

## Features

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

## Getting Started

- Super quick — try the Quickstart Guide.
- More general — try Building an Application with Spring Boot
- More specific — try Building a RESTful Web Service.
- Or search through all our guides on the Guides homepage.

## Spring Boot.

Opinionated view of Spring, simplifies building apps.

Not an application server.

Embeds a servlet container.

Doesn’t implement Java specs,
configures beans that do.

It is not a code generator.

Automatically configures beans.

Frees you from boilerplate configuration.

## Spring Initializr.

start.spring.io

spring initializr

**Project**

[✓ Maven Project] Gradle Project

**Language**

[✓ Java] Kotlin Groovy

**Spring Boot**

2.5.0 (SNAPSHOT) 2.4.2 (SNAPSHOT) [✓ 2.4.1] 2.3.8 (SNAPSHOT) 2.3.7

**Project Metadata**

| | |
|---|---|
| Group | com.example |
| Artifact | demo |
| Name | demo |
| Description | Demo project for Spring Boot |
| Package name | com.example.demo |
| Packaging | [✓ Jar] War |
| Java | 15 [✓ 11] 8 |

**Dependencies**

ADD DEPENDENCIES... ⌘ + B

*No dependency selected*

GENERATE ⌘ + ↵ EXPLORE CTRL + SPACE SHARE...

Pick your language, build preferences, Boot version.

Set the project metadata, select packaging and Java level.

Allows you to select the dependencies your project needs.

start.spring.io
spring initializr
ADD DEPENDENCIES... ⌘ + B

Project
[✓ Maven Project] Gradle Project

Spring Boot
2.5.0 (SNAPSHOT) 2.4.2 (SNAPSHOT
2.3.7

Project Metadata

| | |
|---|---|
| Group | com.example |
| Artifact | demo |
| Name | demo |
| Description | Demo project for Spring Boot |
| Package name | com.example.demo |
| Packaging | [✓ Jar] War |
| Java | 15 [✓ 11] 8 |

Web, Security, JPA, Actuator, Devtools...

Press ⌘ for multiple adds

DEVELOPER TOOLS

**Spring Boot DevTools**
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

**Lombok**
Java annotation library which helps to reduce boilerplate code.

**Spring Configuration Processor**
Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).

WEB

**Spring Web**
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

**Spring Reactive Web**
Build reactive web applications with Spring WebFlux and Netty.

**Rest Repositories**
Exposing Spring Data repositories over REST via Spring Data REST.

**Spring Session**
Provides an API and implementations for managing user session information.

**Rest Repositories HAL Explorer**
Browsing Spring Data REST repositories in your browser.

**Rest Repositories HAL Browser**
Browsing Spring Data REST repositories in your browser.
Requires Spring Boot >= 2.0.0.RELEASE and < 2.2.0.M1.

**Spring HATEOAS**
Eases the creation of RESTful APIs that follow the HATEOAS principle when working with Spring / Spring

Generates a starter project.

## Don’t like GUIs? There is a CLI.

https://docs.spring.io/spring-boot/docs/current/reference/html/spring-boot-cli.html

Open the project in your favorite IDE and away you go!

## Spring Cloud.

Distributed applications share a number of patterns.

Spring Cloud is an umbrella project with out the box solutions.

spring.io

Why Spring | Learn | Projects | Training | Support | Community

- Spring Boot
- Spring Framework
- Spring Data
- Spring Cloud
  - Spring Cloud Azure
  - Spring Cloud Alibaba
  - Spring Cloud for Amazon Web Services
  - Spring Cloud Bus
  - Spring Cloud CLI
  - Spring Cloud for Cloud Foundry
  - Spring Cloud - Cloud Foundry Service Broker
  - Spring Cloud Cluster
  - Spring Cloud Commons
  - Spring Cloud Config
  - Spring Cloud Connectors
  - Spring Cloud Consul
  - Spring Cloud Contract
  - Spring Cloud Function
  - Spring Cloud Gateway
  - Spring Cloud GCP
  - Spring Cloud Netflix
  - Spring Cloud Open Service Broker
  - Spring Cloud Pipelines
  - Spring Cloud Schema Registry
  - Spring Cloud Security
  - Spring Cloud Skipper

## Spring Cloud 2020.0.0

OVERVIEW | LEARN | SAMPLES

Spring Cloud provides tools for developers to quickly build some of the common patterns in distributed systems (e.g. configuration management, service discovery, circuit breakers, intelligent routing, micro-proxy, control bus, one-time tokens, global locks, leadership election, distributed sessions, cluster state). Coordination of distributed systems leads to boiler plate patterns, and using Spring Cloud developers can quickly stand up services and applications that implement those patterns. They will work well in any distributed environment, including the developer's own laptop, bare metal data centres, and managed platforms such as Cloud Foundry.

### Features

Spring Cloud focuses on providing good out of box experience for typical use cases and extensibility mechanism to cover others.

- Distributed/versioned configuration
- Service registration and discovery
- Routing
- Service-to-service calls
- Load balancing
- Circuit Breakers
- Global locks
- Leadership election and cluster state
- Distributed messaging

Useful defaults for Cloud Native applications.

Cloud agnostic.

Generate a project via
start.spring.io

Externalized configuration.

Integration with various Netflix components.

Security. Distributed tracing.
Event driven applications.

Deployment pipelines. Service Discovery. Connectors.

And on and on…

We’ll talk more about this topic!

## Spring Data.

Spring based model
for data access.

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community

Spring Boot

Spring Framework

Spring Data

- Spring Data JDBC
- Spring Data JPA
- Spring Data LDAP
- Spring Data MongoDB
- Spring Data Redis
- Spring Data R2DBC
- Spring Data REST
- Spring Data for Apache Cassandra
- Spring Data for Apache Geode
- Spring Data for Apache Solr
- Spring Data for Pivotal GemFire
- Spring Data Couchbase
- Spring Data Elasticsearch
- Spring Data Envers
- Spring Data Neo4j
- Spring Data JDBC Extensions
- Spring for Apache Hadoop

Spring Cloud

Spring Cloud Data Flow

Spring Security

Spring Session

Spring Integration

Spring HATEOAS

# Spring Data 2020.0.2

OVERVIEW LEARN

Spring Data's mission is to provide a familiar and consistent, Spring-based programming model for data access while still retaining the special traits of the underlying data store.

It makes it easy to use data access technologies, relational and non-relational databases, map-reduce frameworks, and cloud-based data services. This is an umbrella project which contains many subprojects that are specific to a given database. The projects are developed by working together with many of the companies and developers that are behind these exciting technologies.

## Features

- Powerful repository and custom object-mapping abstractions
- Dynamic query derivation from repository method names
- Implementation domain base classes providing basic properties
- Support for transparent auditing (created, last changed)
- Possibility to integrate custom repository code
- Easy Spring integration via JavaConfig and custom XML namespaces
- Advanced integration with Spring MVC controllers
- Experimental support for cross-store persistence

## Main modules

- Spring Data Commons - Core Spring concepts underpinning every Spring Data module.
- Spring Data JDBC - Spring Data repository support for JDBC.

Simplify data access regardless of the datastore.

Relational, non-relational, cloud based, map-reduce.

Broad support for a variety of databases.

We could go on and on…

In fact there are full talks just on the topic!

Schedule

# What Is Spring?

What Is Spring?

Glenn Renfro, Software Developer at VMware

SpringOne

LEARN MORE

Presentation slides

What is Spring?

Have you asked yourself, “What is Spring and what does it do?” Well, this talk is for you! We’ll begin with a brief history of Spring Framework. From there, we’ll discuss the layers of Spring  and what each layer does. As we cover each layer, we’ll give an overview on the key projects that comprise the layer.

manning.com

MANNING PUBLICATIONS
Quality is many small things done right

search

all products MEAP liveBook liveVideo liveProject liveAudio free content live sign in

Save half on *GraphQL in Action* and more TODAY ONLY!

Spring in Action, Sixth Edition

3 reviews

775 views in the last week

Craig Walls

MEAP began May 2020 · Publication in Summer 2021 *(estimated)*

ISBN 9781617297571 · 520 pages *(estimated)* · printed in black & white

free previous edition eBook included

An eBook copy of the previous edition of this book is included at no additional cost. It will be automatically added to your Manning Bookshelf within 24 hours of purchase.

To me, this has always been the defacto standard for documentation on the Spring Framework. I bought the 1st edition when it first came out as we were converting alegacy app to Spring and this book was essential in learning how the current version worked.

Tony Sweets

A new edition of the classic bestseller! *Spring in Action, 6th Edition* covers all of the new features of Spring 5.3 and Spring Boot 2.4 along with examples of reactive programming, Spring Security for REST Services, and bringing reactivity to your databases. You'll also find the latest Spring best practices, including Spring Boot for application setup and configuration.

Look Inside

resources

Source code

Book Forum

*show all*

MEAP

**Manning Early Access Program (MEAP)**
Read chapters as they are written, get the finished eBook as soon as it's ready, and receive the pBook long before it's in bookstores.

8 of 19 chapters available

print book ~~$59.99~~ $29.99
*pBook + eBook + liveBook*
*includes previous edition eBook*
Additional shipping charges may apply

add to cart

eBook ~~$47.99~~ $23.99
*3 formats + liveBook*
*includes previous edition eBook*

add to cart

enable 2-click buy

customers also bought

oreilly.com
O'REILLY
ENTERPRISE
PRICING
SIGN IN
TRY NOW
Search for books, videos, live events, and more
Contents
Learning Path
Spring and Spring Boot Fundamentals
Start

LEARNING PATH

# Spring and Spring Boot Fundamentals

Instructor Ken Kousen

TIME TO COMPLETE:
7h 20m

TOPICS:
Spring

PUBLISHED BY:
O'Reilly Media, Inc.

CREATED:
April 2019

**What is this learning path about, and why is it important?**

For many developers, Spring is the go-to framework for quickly and easily creating web-based enterprise applications. With its comprehensive ecosystem that includes an extensive array of tools and testing capabilities, Spring relieves developers of much of the drudge work when building out web-based, RESTful applications. Spring Boot helps out even further by enabling autoconfiguration of many of the tedious chores that you need to do when starting a

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Blog Tanzu.TV Community

Spring

Spring is the world's most popular framework for writing high performing and easily testable Java code. There are many Spring projects, covering everything from configuration to security, web apps to integration, and plenty more. Start with the Spring guides and learn even more at Spring.io.

Guides

Spring Cloud Gateway: What Is It?

Read more

Spring Cloud Stream: What Is It?

Read more

Prometheus and Grafana: Gathering Metrics from Spring Boot on Kubernetes

Read more

Spring Boot: Building an API

Read more

Spring Cloud Gateway: Getting Started

Read more

Wavefront for Spring Boot: Getting Started

Read more

# DOCUMENTATION

Documentation?
You’re kidding right?

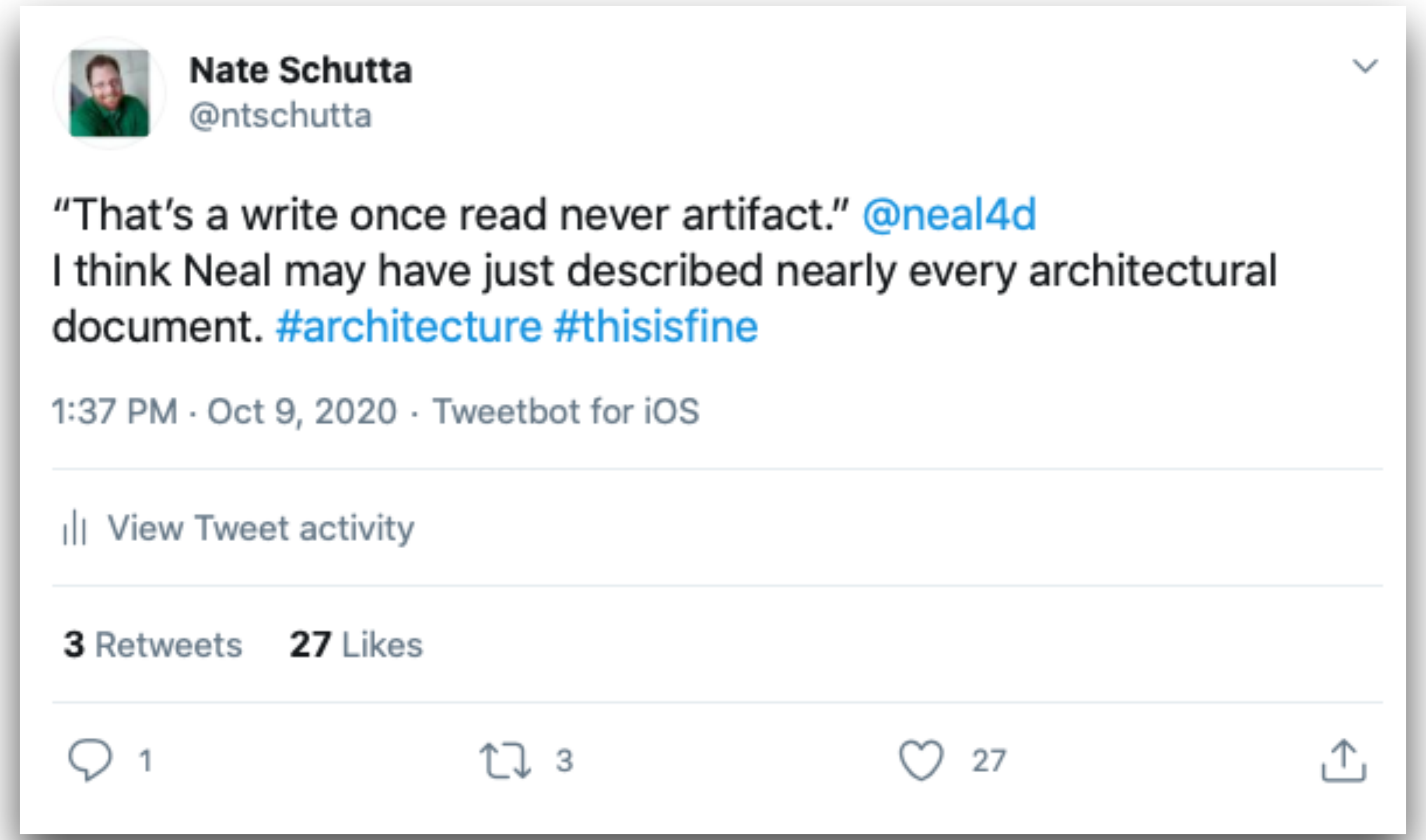
I know what some of you are thinking…

I don’t have time for all this.

We need to MOVE FAST.
And break things…

We’re Agile. With a capital A.

Besides, “the documentation is useless” #amirite?



Nate Schutta
@ntschutta

“That’s a write once read never artifact.” @neal4d
I think Neal may have just described nearly every architectural document. #architecture #thisisfine

1:37 PM · Oct 9, 2020 · Tweetbot for iOS

View Tweet activity

**3** Retweets **27** Likes

1 3 27

https://twitter.com/ntschutta/status/1314636196270739457

That is not, in fact, an inviolable requirement.

Documentation doesn’t have to be high ceremony.

Should answer basic questions!

## What does your service do?

## How does it work?

## What does it depend on?

## Golden rule!

Do it for those that come after you.

Don’t forget, sometimes *you* are the person that comes after you!

How long does it take for a new team member to be productive? Weeks?

## Months?

Solid onboarding guide.

Make sure it is updated.

Documentation should be easy to find.

Probably a website/wiki.

Updating the wiki should be part of the developer workflow.

Consider a simple (low ceremony) template.

Description - what does your service do? Don’t skimp here.

An architectural diagram or three.

Contact information as well as the on call rotation.

Links to helpful things like the repo, dashboard link, on call book.

## FAQ.

Onboarding/development guide.

## Coding standards.

## Development pipeline.

## Glossary.

Whatever helps the team understand.

Everyone should “get it” and be able to describe it. So have them do it.

Spring can help!

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community

Spring Boot
Spring Framework
Spring Data
Spring Cloud
Spring Cloud Data Flow
Spring Security
Spring Session
Spring Integration
Spring HATEOAS
Spring REST Docs
Spring Batch
Spring AMQP
Spring for Android
Spring CredHub
Spring Flo
Spring for Apache Kafka
Spring LDAP
Spring Mobile
Spring Roo
Spring Shell
Spring Statemachine
Spring Vault
Spring Web Flow
Spring Web Services

# Spring REST Docs 2.0.5.RELEASE

OVERVIEW LEARN SAMPLES

Spring REST Docs helps you to document RESTful services.

It combines hand-written documentation written with Asciidoctor and auto-generated snippets produced with Spring MVC Test. This approach frees you from the limitations of the documentation produced by tools like Swagger.

It helps you to produce documentation that is accurate, concise, and well-structured. This documentation then allows your users to get the information they need with a minimum of fuss.

## Spring Boot Config

Spring Boot provides an `@AutoConfigureRestDocs` annotation to leverage Spring REST Docs in your tests.

**Quickstart Your Project**
Bootstrap your application with Spring Initializr.

Word processors don’t lend themselves to a pipeline.

Takes hand crafted Asciidoctor (or Markdown) text…

Combined with autogenerated snippets from test code.

Output is HTML, style away.

And deriving documentation from tests keeps it up to date.

Your developers focus on describing requests & responses.

Change implementation details to your heart’s content!

Shouldn’t be a static thing!

Documentation should be reviewed along with the architecture.

youtube.com

Search

0:04 / 56:59

Documenting RESTful APIs with Spring REST Docs - Jenn Strater

11,097 views • Dec 14, 2017

103
15
SHARE
SAVE

SpringDeveloper
146K subscribers

SUBSCRIBE

RESTful APIs are eating the world, yet all too often the documentation can cause indigestion for the APIs' developers and their users. Developers have to deal with annotation overload, repetition, and an unpleasant writing environment. Users are then left with documentation that's inaccurate and

From Zero to Hero with Spring Boot - Brian Clozel
SpringDeveloper
145K views • 3 years ago
1:09:19

Documenting RESTful APIs
SpringDeveloper
43K views • 5 years ago
1:22:40

Test Driven Development with Spring Boot - Sannidhi Jalukar,...
SpringDeveloper
83K views • 3 years ago
59:27

Bootiful Testing
SpringDeveloper
21K views • 2 years ago
1:18:16

Spring Cloud Function: Write Once, Run Anywhere (For Real!)
SpringDeveloper
5.4K views • 1 year ago
1:10:02

Security Patterns for Microservice Architectures
SpringDeveloper
4K views • 3 months ago
40:30

Introduction to Spring Data
SpringDeveloper
32K views • 4 years ago
31:09

If Hemingway Wrote JavaDocs
SpringDeveloper
2.5K views • 3 months ago
54:30

Mix - SpringDeveloper
YouTube

springone.io

SpringOne

FAQ Sign Out

Schedule

# If Hemingway Wrote JavaDocs

If Hemingway Wrote JavaDocs

Watch later Share

If Hemingway Wrote JavaDocs

Jay Bryant, Senior Technical Writer at VMware;
Josh Cummings, Software Engineer at VMware

SpringOne

LEARN MORE

Presentation slides

Building an Accessibility Culture

You write the docs, but does anyone read them?

This talk covers how to write so well that developers value your documentation. It also addresses the related issues of SEO, internationalization, and accessibility.

# MONITORING

Monitoring is vital to a thriving distributed architecture.

## Four components to monitoring.

## Logging.

What would you say my service is doing?

Log anything that is useful.

Just don’t put in any personally identifying information (PII).

Ever.

Some things alone aren’t PII but when combined with other items…

Tracing can be difficult.

Can’t just put in a breakpoint and step through the code…

Calls bounce between 5 or 10 (or more) services.

Correlation IDs help.

Speaking of which…

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community
Spring Boot
Spring Framework
Spring Data
Spring Cloud
Spring Cloud Azure
Spring Cloud Alibaba
Spring Cloud for Amazon Web Services
Spring Cloud Bus
Spring Cloud CLI
Spring Cloud for Cloud Foundry
Spring Cloud - Cloud Foundry Service Broker
Spring Cloud Cluster
Spring Cloud Commons
Spring Cloud Config
Spring Cloud Connectors
Spring Cloud Consul
Spring Cloud Contract
Spring Cloud Function
Spring Cloud Gateway
Spring Cloud GCP
Spring Cloud Netflix
Spring Cloud Open Service Broker
Spring Cloud Pipelines
Spring Cloud Schema Registry
Spring Cloud Security
Spring Cloud Skipper
OVERVIEW
LEARN
SAMPLES
COPY

# Spring Cloud Sleuth 3.0.0

Spring Cloud Sleuth provides Spring Boot auto-configuration for distributed tracing.

## Features

Sleuth configures everything you need to get started. This includes where trace data (spans) are reported to, how many traces to keep (sampling), if remote fields (baggage) are sent, and which libraries are traced.

Specifically, Spring Cloud Sleuth...

- Adds trace and span ids to the Slf4J MDC, so you can extract all the logs from a given trace or span in a log aggregator.
- Instruments common ingress and egress points from Spring applications (servlet filter, rest template, scheduled actions, message channels, feign client).
- If `spring-cloud-sleuth-zipkin` is available then the app will generate and report Zipkin-compatible traces via HTTP. By default it sends them to a Zipkin collector service on localhost (port 9411). Configure the location of the service using `spring.zipkin.baseUrl`.

## Spring Boot Config

Add Sleuth to your classpath:

Maven

```
<dependencyManagement>
    <dependencies>
        <dependency>
            <groupId>org.springframework.cloud</groupId>
            <artifactId>spring-cloud-dependencies</artifactId>
            <version>${release.train.version}</version>
            <type>pom</type>
            <scope>import</scope>
```

## Auto-configured for distributed tracing!

Covers spans, sampling and key:value pairs.

Adds trace and span IDs and to stock ingress & egress points.

Instruments common ingress & egress points.

Generates Zipkin compatible traces if desired.

Basically add Sleuth to your classpath…

## And your Boot app can generate trace data!

## Dashboards.

View the health of a service.

Metrics should be displayed on a dashboard of some sort.

But we should be alerted when things start to go wonky.

## Alerting.

A key metric is out of band.

Allows us to detect an issue and fix it before our customers even notice.

Pager duty.

Must be sustainable.

Provide clear, concise on call documentation.

“We don't rise to the level of our expectations, we fall to the level of our training.”

– Archilochus

Vital that we think about just what we should be monitoring.

## What *is* a key metric?

Some pertain solely to the infrastructure our service runs on.

CPU utilization, RAM utilization, threads, database connections…

These often impact more than just our service.

Others key metrics are specific to our service.

Additionally we need to know the availability, latency, response time…

Basically anything that we identified earlier as part of our SLO.

Monitor errors and exceptions as well.

Identify normal, warning and critical thresholds for your metrics.

Can be hard to figure out early on.
Need a certain amount of history.

Not just a prod thing. We need to monitor staging.

Validates the monitors.

Metrics should be displayed on a dashboard of some sort.

But we should be alerted when things start to go wonky.

We shouldn’t be staring at our dashboards all day!

Alert on all of our key metrics, SLOs etc.

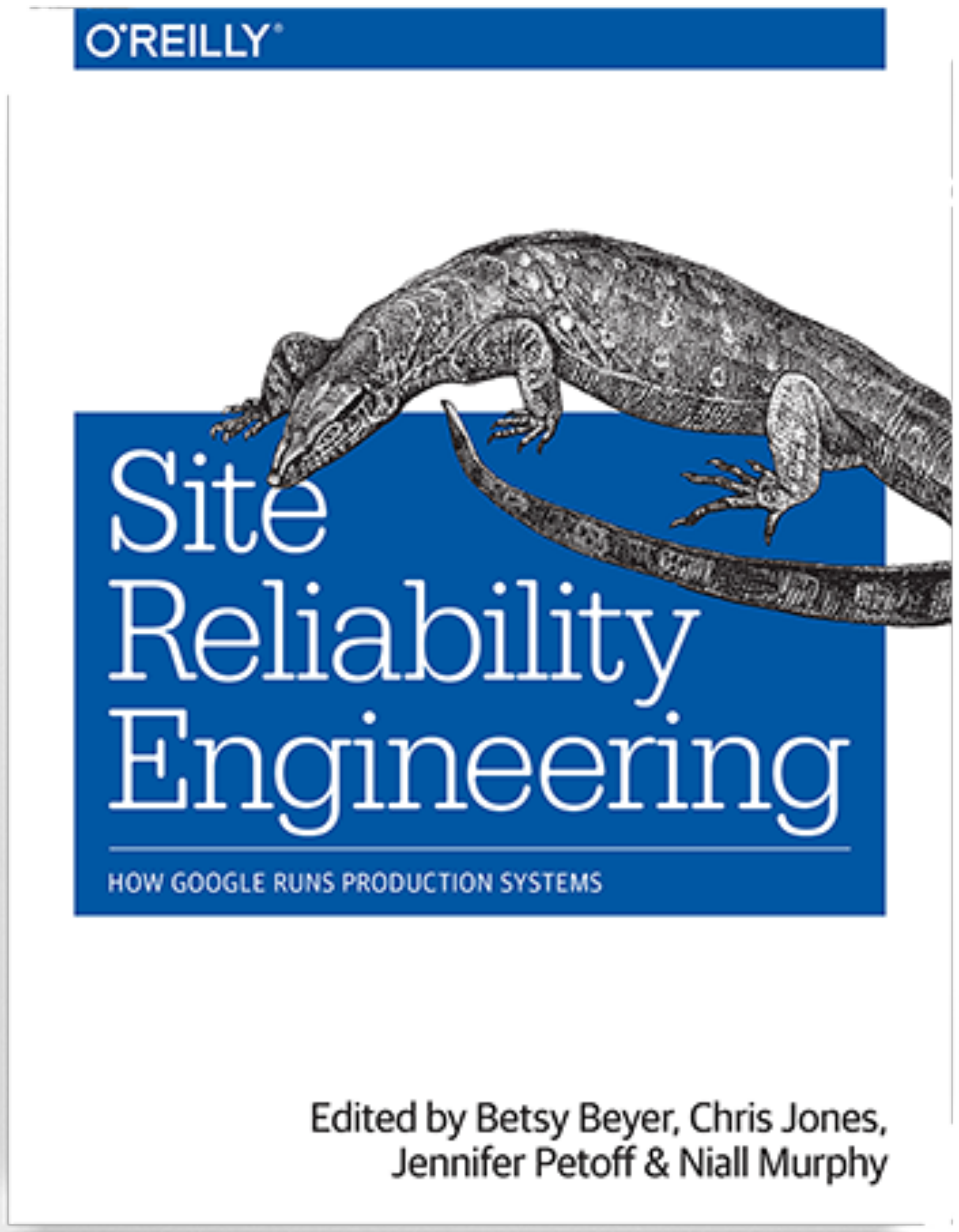
Absence of a key metric is also an alertable offense!

Alerts should be actionable.

Alerts should be urgent.

Alerts should require human intervention.

System can “fix” itself? Not an alert - monitor and/or report.


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy

https://landing.google.com/sre/book.html

## Four Golden Signals.

https://landing.google.com/sre/book/chapters/monitoring-distributed-systems.html#xref_monitoring_golden-signals

Latency - how long does it take to service a request.

Traffic - level of demand on the system. Requests/second. I/O rate.

Errors - failed requests. Can be explicit, implicit or policy failure.

Saturation - how much of a constrained resource is left.

Important to consider the sampling frequency.

High resolution can be costly.

Aggregate data.

Number of tools from Wavefront to Dynatrace to New Relic.

## Spring Boot Actuator!

https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-metrics.html

docs.spring.io

## 6. Metrics

Spring Boot Actuator provides dependency management and auto-configuration for Micrometer, an application metrics facade that supports numerous monitoring systems, including:

- AppOptics
- Atlas
- Datadog
- Dynatrace
- Elastic
- Ganglia
- Graphite
- Humio
- Influx
- JMX
- KairosDB
- New Relic
- Prometheus
- SignalFx
- Simple (in-memory)
- Stackdriver
- StatsD
- Wavefront

To learn more about Micrometer's capabilities, please refer to its reference documentation, in particular the concepts section.

### 6.1. Getting started

Spring Boot auto-configures a composite `MeterRegistry` and adds a registry to the composite for each of the supported implementations that it finds on the classpath. Having a dependency on `micrometer-registry-{system}` in your runtime classpath is enough for Spring Boot to configure the registry.

Metrics can’t afford to be hand-rolled solutions.

Take advantage of Actuator’s built in endpoints.

## 2. Endpoints

Actuator endpoints let you monitor and interact with your application. Spring Boot includes a number of built-in endpoints and lets you add your own. For example, the `health` endpoint provides basic application health information.

Each individual endpoint can be enabled or disabled and exposed (made remotely accessible) over HTTP or JMX. An endpoint is considered to be available when it is both enabled and exposed. The built-in endpoints will only be auto-configured when they are available. Most applications choose exposure via HTTP, where the ID of the endpoint along with a prefix of `/actuator` is mapped to a URL. For example, by default, the `health` endpoint is mapped to `/actuator/health`.

The following technology-agnostic endpoints are available:

| ID | Description |
|---|---|
| `auditevents` | Exposes audit events information for the current application. Requires an `AuditEventRepository` bean. |
| `beans` | Displays a complete list of all the Spring beans in your application. |
| `caches` | Exposes available caches. |
| `conditions` | Shows the conditions that were evaluated on configuration and auto-configuration classes and the reasons why they did or did not match. |
| `configprops` | Displays a collated list of all `@ConfigurationProperties`. |
| `env` | Exposes properties from Spring's `ConfigurableEnvironment`. |
| `flyway` | Shows any Flyway database migrations that have been applied. Requires one or more `Flyway` beans. |
| `health` | Shows application health information. |
| `httptrace` | Displays HTTP trace information (by default, the last 100 HTTP request-response exchanges). Requires an `HttpTraceRepository` bean. |
| `info` | Displays arbitrary application info. |
| `integrationgraph` | Shows the Spring Integration graph. Requires a dependency on `spring-integration-core`. |
| `loggers` | Shows and modifies the configuration of loggers in the application. |
| `liquibase` | Shows any Liquibase database migrations that have been applied. Requires one or more `Liquibase` beans. |

You can create your own custom endpoints as well.

Takes time to get monitoring right.

## Do you even SRE?

Beware the metric that is easy to measure…

Might not be meaningful. Sorry.

Also key to understand the business drivers.

What could cause a spike in demand?

## How does that translate to specific services?

Be realistic!

We can’t all be a third of internet traffic!

dzone.com

DZone Java Zone

Over a million developers have joined DZone. Log In / Join

REFCARDZ RESEARCH WEBINARS | Agile AI Big Data Cloud Database DevOps Integration IoT Java Microservices Open Source Performance Security Web Dev

TREND MICRO Build confidently with continuous security, compliance, and governance Learn more

DZone > Java Zone > Application Monitoring With Spring Boot Actuator

# Application Monitoring With Spring Boot Actuator

The Spring Boot Actuator is a module built into Spring Boot that has a number of features that make it easy to manage and monitor applications.

by Sanjoy Kumer Deb ‹/›CORE · Mar. 27, 20 · Java Zone · Tutorial

Like (17) Comment (2) Save Tweet 26.58K Views

Join the DZone community and get the full member experience. JOIN FOR FREE

Internet of Things Product Selector

Digi-Key's IoT Component Selector is your one-stop-shop for the IoT. Learn More ▶

Monitoring production is an important part of a software service provider. Many companies providing monitoring systems for maintaining the production environment. Spring Boot comes with different awesome modules that developers can easily configure and maintain development and production environments with. The actuator module provides production-ready features by which we can easily maintain the production environment. The actuator exposes JMX and HTTP endpoints.

## Features

- **Endpoints:** Spring Boot Actuator provides some default endpoints by which we can access application information. We can also monitor the production environment with those endpoints. Endpoints can also be accessed by third-party monitoring tools.

Digi-Key ELECTRONICS

TREND MICRO

Don't let cloud misconfigurations slow you down

Build confidently with continuous security, compliance, and governance

Learn more

youtube.com

YouTube

Search

Mastering Spring Boot's Actuator

21,573 views • Oct 4, 2018

261 5 SHARE SAVE

0:04 / 1:05:47

SpringDeveloper
146K subscribers

SUBSCRIBE

Spring Boot's Actuator provides a powerful set of production-ready features that have been recently updated for Spring Boot 2.0. In this talk we'll look in detail at the Actuator, focusing on the new features including the new endpoint infrastructure that introduces support for Jersey and Web Flux

Performance Monitoring Backend and Frontend using...
SpringDeveloper
3.4K views • 1 year ago
56:34

All | Spring Framework | Related | From Spring

How Fast is Spring?
SpringDeveloper
14K views • 2 years ago
1:15:25

Introducing Micrometer Application Metrics - Jon...
SpringDeveloper
19K views • 3 years ago
56:46

How to Get Productive with Spring Boot
SpringDeveloper
10K views • 1 year ago
1:03:56

The Beginner's Guide To Spring Cloud - Ryan Baxter
SpringDeveloper
90K views • 3 years ago
1:08:42

Reactive Spring - Josh Long, Mark Heckler
SpringDeveloper
20K views • 3 years ago
1:09:39

Mix - SpringDeveloper
YouTube

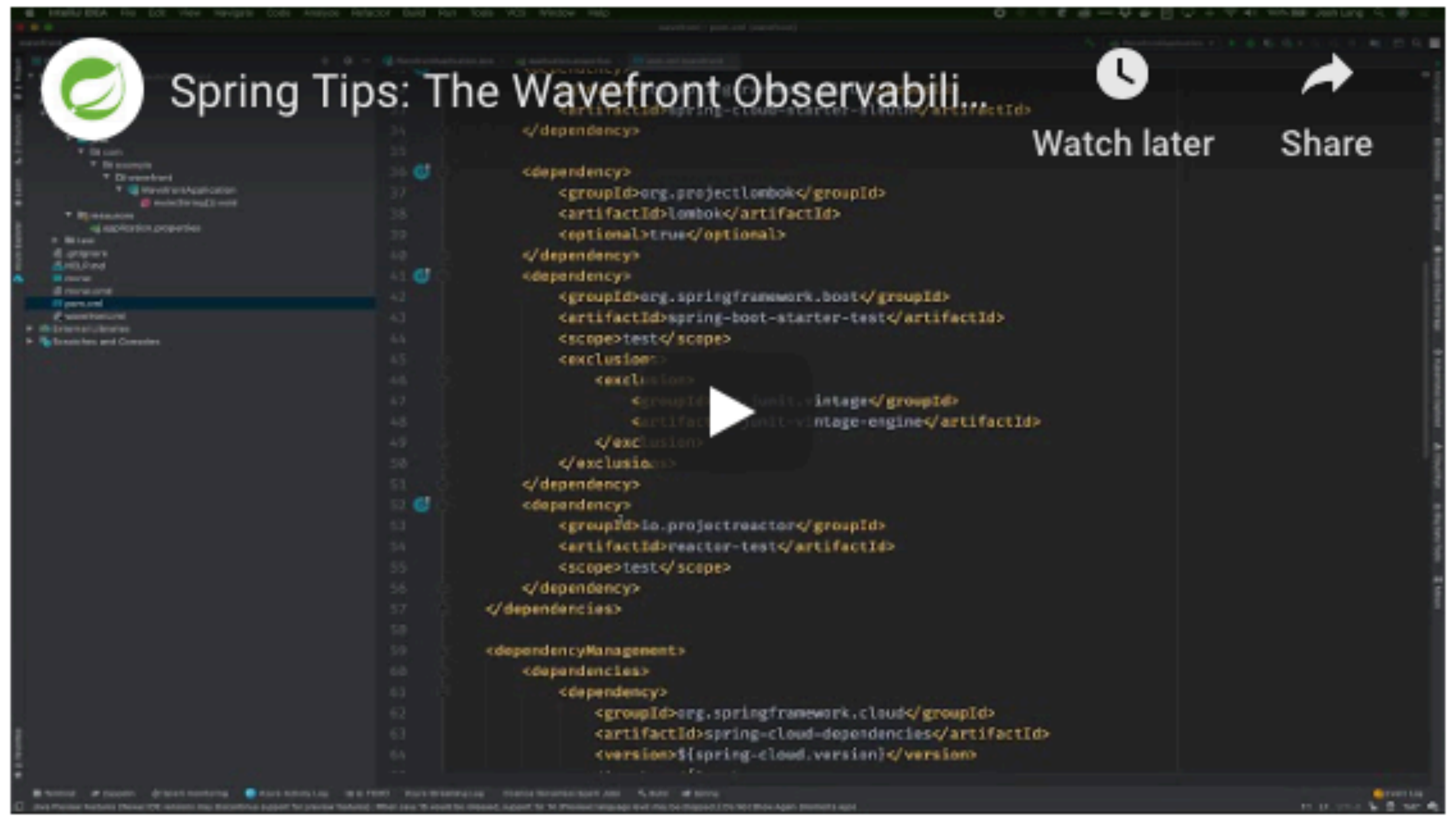
JUnit 5: the Next Step in Automated Testing
SpringDeveloper
21K views • 2 years ago
30:29

From Zero to Hero with Spring Boot - Brian Clozel
SpringDeveloper
145K views • 3 years ago
1:09:19

# Spring Tips: The Wavefront Observability Platform

ENGINEERING | JOSH LONG | APRIL 29, 2020 0 COMMENT

speaker: Josh Long (@starbuxman)



```
        </dependency>

        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
            <exclusions>
                <exclusion>
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>io.projectreactor</groupId>
            <artifactId>reactor-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>org.springframework.cloud</groupId>
                <artifactId>spring-cloud-dependencies</artifactId>
                <version>${spring-cloud.version}</version>
```

Hi, Spring fans! Welcome to another installment of Spring Tips! In this installment, we'll revisit two topics that we've addressed in two previous videos (distributed tracing and metrics collection) in terms of the superb Tanzu Wavefront observability platform.

The first video of the two videos, as mentioned above, dating way back in early 2017, looked at distributed tracing with spring cloud sleuth and openzipkin. Spring Cloud Sleuth is an abstraction for capturing the flow of messages from one node to another. It's useful to help you see how messages move through a system. Spring cloud sleuth integrates with all the usual ingress and egress points in a Spring Boot application. Make an HTTP request using either the `Restteplat` or the reactive `WebClient` or Spring Cloud Feign? It works. Receive an HTTP request to a traditional (Servlet-based) or reactive HTTP endpoint built with Spring? It works. Send or receive a message using Spring Cloud Stream or Spring Integration? Yep. You guessed it. It just works. You don't have to do anything

Get the Spring newsletter

Email Address

☐ Yes, I would like to be contacted by The Spring Team and VMware for newsletters, promotions and events per the terms of VMware's Privacy Policy

SUBSCRIBE

youtube.com

Search

SpringOne

0:01 / 25:31

Spring Boot Observability

1,187 views • Sep 25, 2020

27 0 SHARE SAVE

SpringDeveloper
146K subscribers

SUBSCRIBE

The best person to create observability for your application is you. Going from reactive monitoring and alerting to proactive observability enables you to deliver your apps with confidence. Make yourself, the business, SRE, and ops teams happy by delivering observability, with your code, to

Spring Security Patterns
SpringDeveloper
2.8K views • 3 months ago

All | Spring Framework | Related | From Spring

Spring Tips: The Wavefront Observability Platform
SpringDeveloper
9.7K views • 8 months ago

Building Robust and Resilient Apps Using Spring Boot and...
SpringDeveloper
8.8K views • 1 year ago

Spring Boot Loves K8s
SpringDeveloper
1.9K views • 3 months ago

Performance Monitoring Backend and Frontend using...
SpringDeveloper
3.4K views • 1 year ago

Mix - SpringDeveloper
YouTube

It's a Kind of Magic: Under the Covers of Spring Boot - Brian...
SpringDeveloper
34K views • 3 years ago

Guide to "Reactive" for Spring MVC Developers
SpringDeveloper
31K views • 2 years ago

Mastering Spring Boot's Actuator
SpringDeveloper
21K views • 2 years ago

# FAULT TOLERANCE

Distributed systems are not islands unto themselves.

Services fail.

Failures, uh find a way.

Our customers don’t care why.

You cannot prevent failure…but you can be prepared for it.

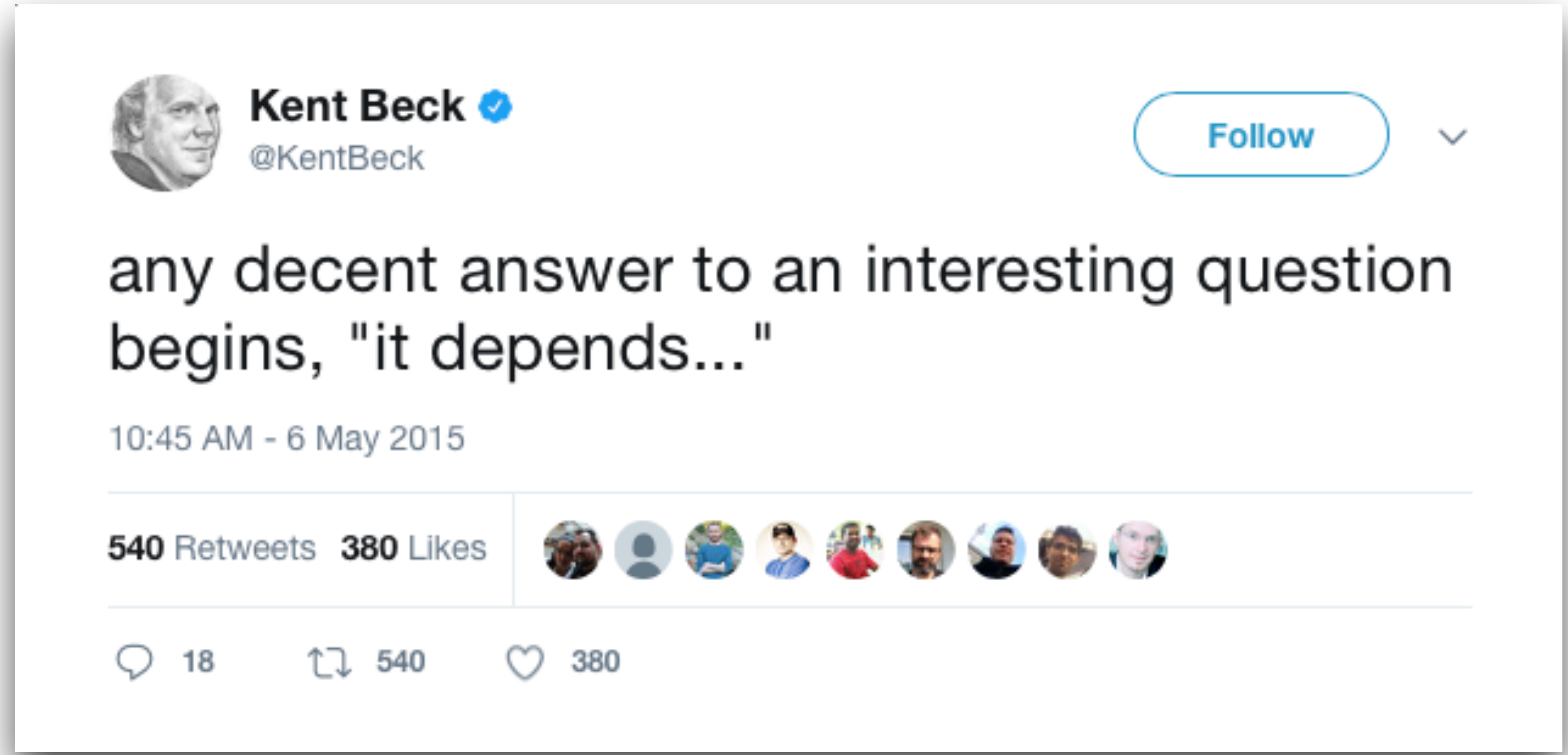
## How should we react?

Error message?

Call a backup service?

## Do we need to cache data?

Do we return a default answer?


Kent Beck
@KentBeck
Follow

any decent answer to an interesting question begins, "it depends..."

10:45 AM - 6 May 2015

540 Retweets 380 Likes

18 540 380

https://twitter.com/KentBeck/status/596007846887628801

## The circuit breaker pattern.

https://martinfowler.com/bliki/CircuitBreaker.html

Originally described by Michael Nygard.

Closed
on call / pass through
call succeeds / reset count
call fails / count failure
threshold reached / trip breaker
trip breaker
Open
on call / fail
on timeout / attempt reset
trip breaker
reset
Half-Open
on call / pass through
call succeeds / reset
call fails / trip breaker

Circuit breaker watches the calls.

Once they exceed a failure threshold, the circuit is opened.

Redirects to the
fallback mechanism.

Periodically checks to see if the service is repaired.

If so, circuit is closed.

Multiple circuit breaker implementations to pick from.

## Spring Cloud Circuit Breaker to the rescue!

spring.io

Why Spring Learn Projects Training Support Community

# Spring Cloud Circuit Breaker

1.0.4.RELEASE

OVERVIEW LEARN SAMPLES

## Introduction

Spring Cloud Circuit breaker provides an abstraction across different circuit breaker implementations. It provides a consistent API to use in your applications allowing you the developer to choose the circuit breaker implementation that best fits your needs for your app.

### Supported Implementations

- Netfix Hystrix
- Resilience4J
- Sentinel
- Spring Retry

## Core Concepts

To create a circuit breaker in your code you can use the `CircuitBreakerFactory` API. When you include a Spring Cloud Circuit Breaker starter on your classpath a bean implementing this API will automatically be created for you. A very simple example of using this API is given below

COPY

```
@Service
public static class DemoControllerService {
        private RestTemplate rest;
        private CircuitBreakerFactory cbFactory;

        public DemoControllerService(RestTemplate rest, CircuitBreakerFactory cbFactory) {
                this.rest = rest;
                this.cbFactory = cbFactory;
        }

        public String slow() {
                return cbFactory.create("slow").run(() -> rest.getForObject("/slow",
```

Consistent API, allows developers to pick the implementation.

Supports Netflix Hystrix, Resilience4j, Sentinel, Spring Retry.

Add the proper starter to the class path, call the factory…

You can now inject the circuit breaker wherever you see fit.

Each circuit breaker can be individually configured.

Can also create default configuration for all circuit breakers.

Free to change failure thresholds, slow call thresholds…

## Minimum number of calls, sliding window size…

Circuit breakers are vital for a healthy micro(services)biome.

It isn’t hard to add!

Your customers will thank you…

And you can avoid 3 AM pages.

spring.io

Why Spring Learn Projects Training Support Community

ALL GUIDES

# Circuit Breaker

This guide walks you through the process of applying circuit breakers to potentially failing method calls by using the Netflix Hystrix fault tolerance library.

## What You Will Build

You will build a microservice application that uses the circuit breaker pattern to gracefully degrade functionality when a method call fails. Use of the Circuit Breaker pattern can let a microservice continue operating when a related service fails, preventing the failure from cascading and giving the failing service time to recover.

## What You Need

- About 15 minutes
- A favorite text editor or IDE
- JDK 1.8 or later
- Gradle 4+ or Maven 3.2+
- You can also import the code straight into your IDE:
  - Spring Tool Suite (STS)
  - IntelliJ IDEA

## How to complete this guide

Like most Spring Getting Started guides, you can start from scratch and complete each step or you can bypass basic setup steps that are already familiar to you. Either way, you end up with working code.

Get the Code

Go To Repo

Spring Tips: Spring Cloud Circuit Breaker

13,781 views • Apr 24, 2019

263 3 SHARE SAVE

SpringDeveloper
146K subscribers

SUBSCRIBE

Hi Spring fans! In this installment we look at the just-announced Spring Cloud Circuit Breaker project, which provides an abstraction atop Netflix' Hystrix, Resilience4J, Alibaba's Sentinel and Spring Retry and supports reactive and non-reactive circuits.

All | Spring Framework | Related | From Spring

| | |
|---|---|
| 52:02 | Building Robust and Resilient Apps Using Spring Boot and... SpringDeveloper 8.8K views • 1 year ago |
| 1:10:39 | Spring Cloud Gateway - Spencer Gibb, Sree Tummidi SpringDeveloper 12K views • 3 years ago |
| 22:12 | Spring Tips: Circuit Breakers SpringDeveloper 18K views • 4 years ago |
| 40:00 | Spring Tips: Spring Cloud Gateway SpringDeveloper 23K views • 3 years ago |
| 36:53 | Spring Tips: Reactive Web Views SpringDeveloper 11K views • 1 year ago |
| 48:39 | Spring Tips: Organizational Consistency in your Spring Bo... SpringDeveloper 16K views • 1 year ago |
| 1:15:25 | How Fast is Spring? SpringDeveloper 14K views • 2 years ago |
| 1:30:59 | 12 Factor, or Cloud Native Apps for Spring Developers SpringDeveloper 21K views • 5 years ago |
| 44:15 | Spring Tips: Spring Batch and Apache Kafka SpringDeveloper 21K views • 1 year ago |

SC NETFLIX

Should be obvious …distributed systems have similar needs.

## Common patterns!

## Service discovery.

## Circuit breaker (we already touched on this!)

## Routing. Client side load balancing.

You know who has a lot of distributed app experience?

Netflix.

## Who has built out several OSS components to help?

Netflix.

Add some annotations and you’re good to go!

Integration of Netflix OSS into Boot apps.

Eureka.

Service discovery is a key part of distributed applications.

Services come and go, they’re scaled up and down.

Don’t try to configure by hand!

Two pieces - Eureka Server and Eureka Client.

Eureka Server can be configured to be HA.

Default is to run multiple instances and peer them.

Can also run standalone.

Clients register with Eureka.

Provide common info: port, host, health check, etc.

Put the Eureka client starter on the classpath & apps auto register.

Eureka gets a regular heartbeat from the service instances.

Heartbeat fails? Removed from the registry.

Uses the default `/info` and `/health` actuator endpoints.

Can be configured to register secure applications.

It can take up to 3 heartbeats to get everyone on the same page.

You can shorten the default heartbeat time period.

EurekaClient can then be use to find service instances.

There are alternative clients.

## Running in multiple zones?

You can configure it to use services in the same zone.

We don't just need service discovery though do we?

## Ribbon - client side load balancer.

Works with (or without) Eureka.

Add the starter.

Bet you saw that coming.

External properties configured via Boot configuration files.

You can create default configurations…

Can also customize by setting properties by environment.

You can also directly access the Ribbon API.

Configuration can be lazy loaded on first request.

Or set to load it eagerly.

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community

ALL GUIDES

# Service Registration and Discovery

This guide walks you through the process of starting and using the Netflix Eureka service registry.

## What You Will Build

You will set up a Netflix Eureka service registry and then build a client that both registers itself with the registry and uses it to resolve its own host. A service registry is useful because it enables client-side load-balancing and decouples service providers from consumers without the need for DNS.

## What You Need

- About 15 minutes
- A favorite text editor or IDE
- JDK 1.8 or later
- Gradle 4+ or Maven 3.2+
- You can also import the code straight into your IDE:
  - Spring Tool Suite (STS)
  - IntelliJ IDEA

## How to complete this guide

Like most Spring Getting Started guides, you can start from scratch and complete each step or you can bypass basic setup steps that are already familiar to you. Either way, you end up with working code.

To **start from scratch**, move on to Starting with Spring Initializr.

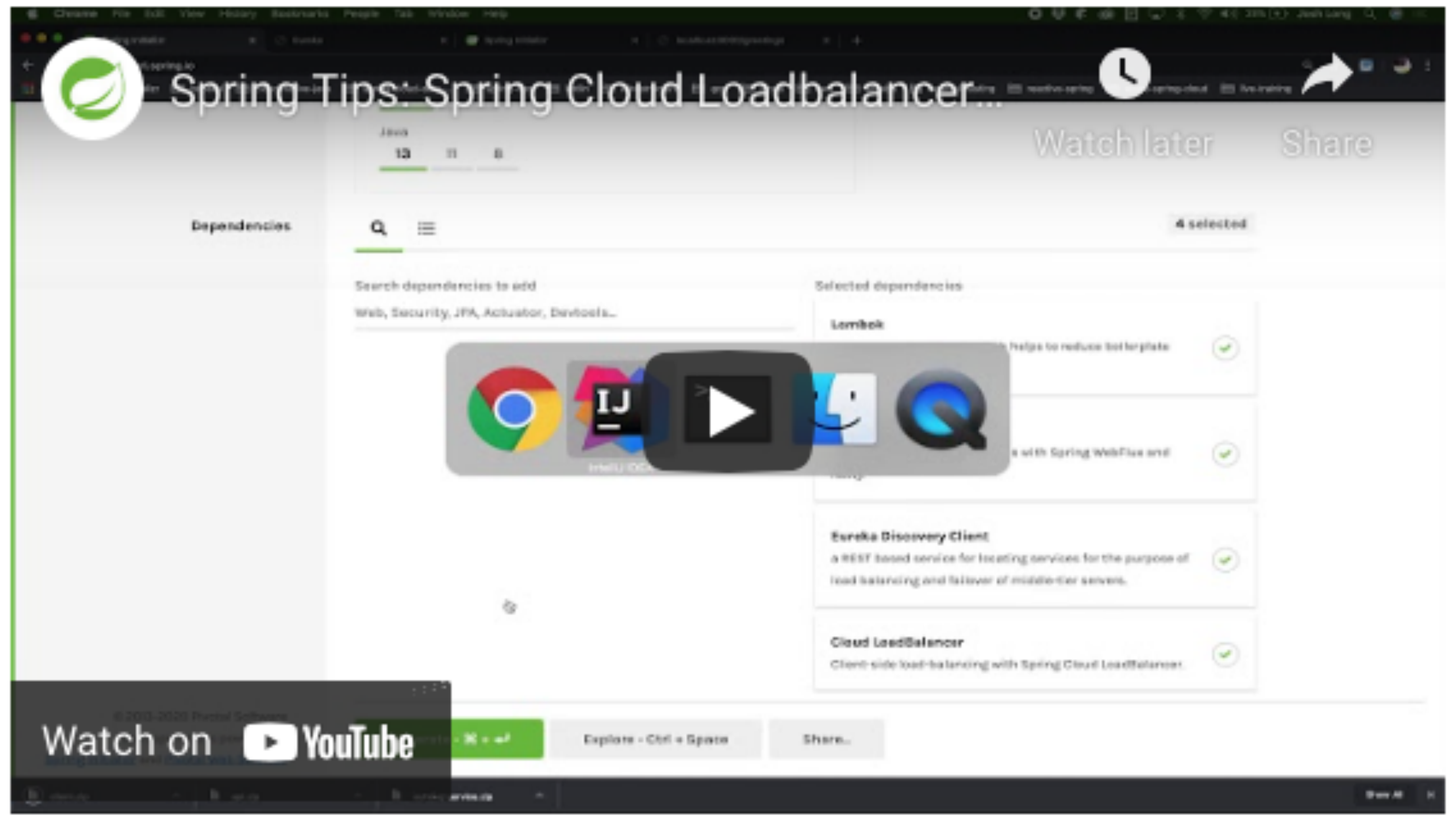
Get the Code

Go To Repo

# Spring Tips: Spring Cloud Loadbalancer

**ENGINEERING | JOSH LONG | MARCH 25, 2020 14 COMMENTS**

speaker: Josh Long (@starbuxman)



Hi, Spring fans! Welcome to another installment of Spring Tips! In this installment, we're going to look at a new feature in Spring Cloud, Spring Cloud Loadbalancer. Spring Cloud Loadbalancer is a generic abstraction that can do the work that we used to do with Netflix's Ribbon project. Spring Cloud still supports Netflix Ribbon, but Netflix Ribbons days are numbered, like so much else of the Netflix microservices stack, so we've provided an abstraction to support an alternative.

## The Service Registry

For us to use the Spring Cloud Load Balancer, we need to have a service registry up and running. A service registry makes it trivial to programmatically query for the location of a given service in a system. There are several popular implementations, including Apache Zookeeper, Netflix's Eureka, Hashicorp Consul, and others. You can even use Kubernetes and Cloud Foundry as service registries. Spring Cloud provides an abstraction, `DiscoveryClient`, that you can use to talk to these service

## Wait? Aren’t parts of Spring Cloud Netflix in maintenance mode?

youtube.com

YouTube

Search

Find the words.

Get Grammarly
Ad www.grammarly.com
DOWNLOAD

All | Spring Framework | Related | From Sp

SpringOne Platform by Pivotal

How to live in a post-Spring-Cloud-Netflix world

October 7–10, 2019
Austin Convention Center

SpringOne Platform by Pivotal | October 7–10 / Austin, TX

0:11 / 1:09:31

How to Live in a Post-Spring-Cloud-Netflix World

11,251 views • Oct 15, 2019

170 | 10 | SHARE | SAVE

SpringDeveloper
158K subscribers

SUBSCRIBED

Zuul? Gateway? Should we get rid of Ribbon? What is going on with Hystrix? If you've ever faced those questions, come and listen to this talk.

Guide to "Reactive" for Spring MVC Developers
SpringDeveloper
38K views • 2 years ago
1:04:27

Mix - SpringDeveloper
YouTube

Best Practices to Spring to Kubernetes Easier and Faster
SpringDeveloper
16K views • 1 year ago
1:07:42

Cloud Native Java - Josh Long
SpringDeveloper
47K views • 4 years ago
1:49:20

Streaming with Spring Cloud Stream and Apache Kafka
SpringDeveloper
24K views • 1 year ago
59:26

Building Robust and Resilient Apps Using Spring Boot and...
SpringDeveloper
12K views • 1 year ago
52:02

How to Get Productive with Spring Boot
SpringDeveloper
14K views • 1 year ago
1:03:56

Living on the Edge with Spring Cloud Gateway

## SC STREAM

Many distributed apps utilize event driven architectures.

Wait. What do *you* mean by events?

Depends on who you ask!

# What do you mean by “Event-Driven”?

Martin Fowler

07 February 2017

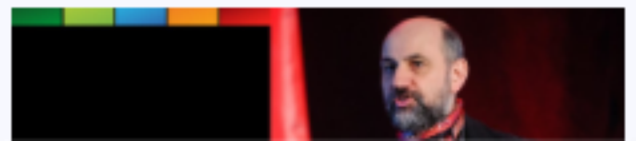
**Translations:** Japanese
DESIGN
EVENT ARCHITECTURES

Towards the end of last year I attended a workshop with my colleagues in ThoughtWorks to discuss the nature of “event-driven” applications. Over the last few years we've been building lots of systems that make a lot of use of events, and they've been often praised, and often damned. Our North American office organized a summit, and ThoughtWorks senior developers from all over the world showed up to share ideas.

The biggest outcome of the summit was recognizing that when people talk about “events”, they actually mean some quite different things. So we spent a lot of time trying to tease out what some useful patterns might be. This note is a brief summary of the main ones we identified.

## Event Notification

This happens when a system sends event messages to notify other systems of a change in its domain. A key element of event notification is that the source system doesn't really care much about the response. Often it doesn't expect any answer at all, or if there is a response that the source does care about, it's indirect. There would be a marked separation between the logic flow that sends the event and any logic flow that responds to some reaction to that event.



Event notification is nice because it implies a low level of coupling, and is pretty simple to set up. It can become problematic, however, if there really is a logical flow that runs over various event notifications. The problem is that it can be hard

## In the eye of the beholder?

There are multiple event patterns.
Which one are you using?

Event notification.

Something happens, source system shouts into the void.

"A new customer signed up!"

Event emitter doesn’t care what happens next.

Highly asynchronous.

## The 0th Law of Computer Science:

High cohesion, low coupling…

But there are downsides.

What would you say you do around here?

Hard to debug, difficult to reason about the system.

Monitoring, monitoring, monitoring.

Easy to lose sight of the flow.

Event-carried state transfer.

Event includes details.

Customer address updated and here is the new address.

Event subscribers don’t have to ask.

Aka Tell Don‘t Ask.

You know, object oriented programming 101.

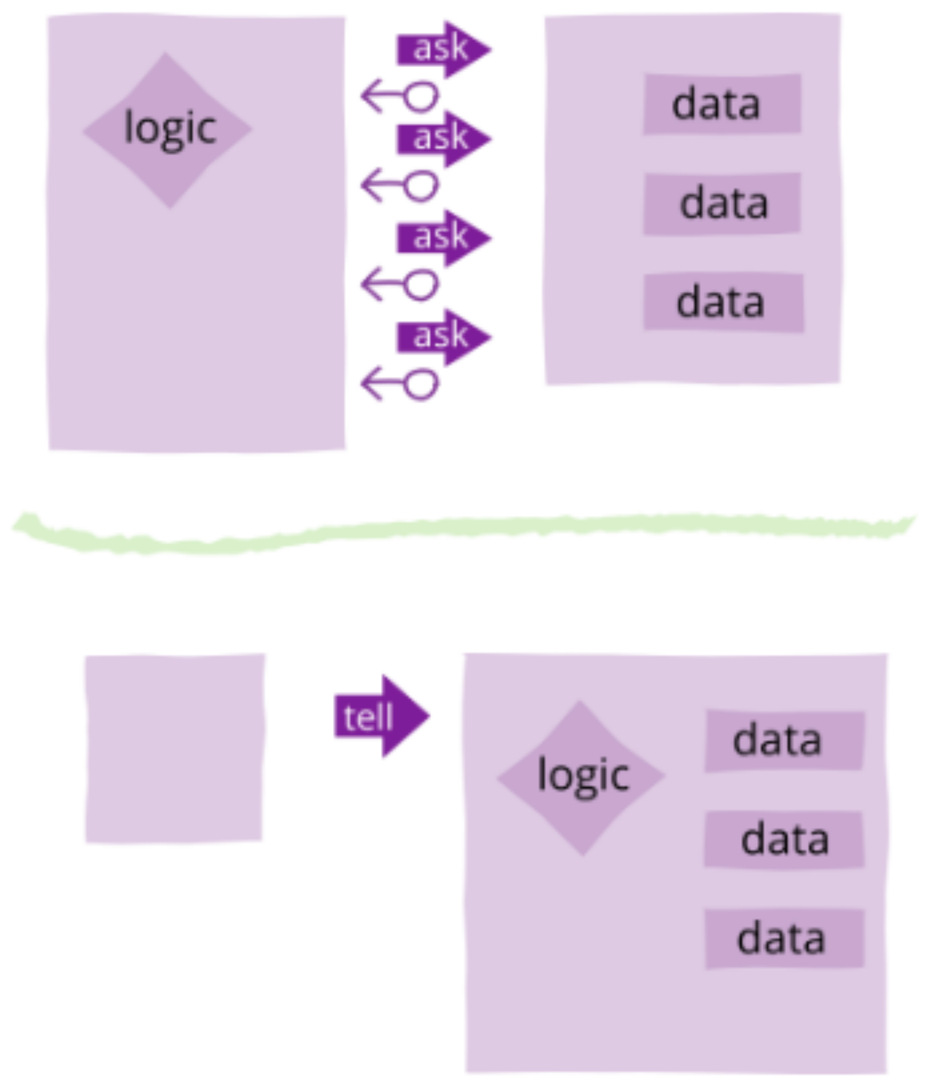
# TellDontAsk

5 September 2013

Martin Fowler

ENCAPSULATION
API DESIGN
OBJECT COLLABORATION DESIGN

Tell-Don't-Ask is a principle that helps people remember that object-orientation is about bundling data with the functions that operate on that data. It reminds us that rather than asking an object for data and acting on that data, we should instead tell an object what to do. This encourages to move behavior into an object to go with the data.



Let's clarify with an example. Let's imagine we need to monitor certain values, signaling an alarm should the value rise above a certain limit. If we write this in an "ask" style, we might have a data structure to represent these things…

*class AskMonitor…*

```
  private int value;
  private int limit;
  private boolean isTooHigh;
```

Reduced latency.

Lower overhead on the source systems.

Lots of data thrown around.

And receivers have to handle state.

Event sourcing.

Record every state change.

Event store is the source of truth.

## Did someone say Kafka?

# INTRODUCTION

*Everything you need to know about Kafka in 10 minutes*

## What is event streaming?

Event streaming is the digital equivalent of the human body's central nervous system. It is the technological foundation for the 'always-on' world where businesses are increasingly software-defined and automated, and where the user of software is more software.

Technically speaking, event streaming is the practice of capturing data in real-time from event sources like databases, sensors, mobile devices, cloud services, and software applications in the form of streams of events; storing these event streams durably for later retrieval; manipulating, processing, and reacting to the event streams in real-time as well as retrospectively; and routing the event streams to different destination technologies as needed. Event streaming thus ensures a continuous flow and interpretation of data so that the right information is at the right place, at the right time.

## What can I use event streaming for?

Event streaming is applied to a wide variety of use cases across a plethora of industries and organizations. Its many examples include:

- To process payments and financial transactions in real-time, such as in stock exchanges, banks, and insurances.

Strong audit log. Easy to recreate history.

Run hypotheticals.

Evolving schema can hurt.

Challenging to replay when we interact with outside systems.

CQRS.

Command Query Responsibility Segregation.

# CQRS

14 July 2011

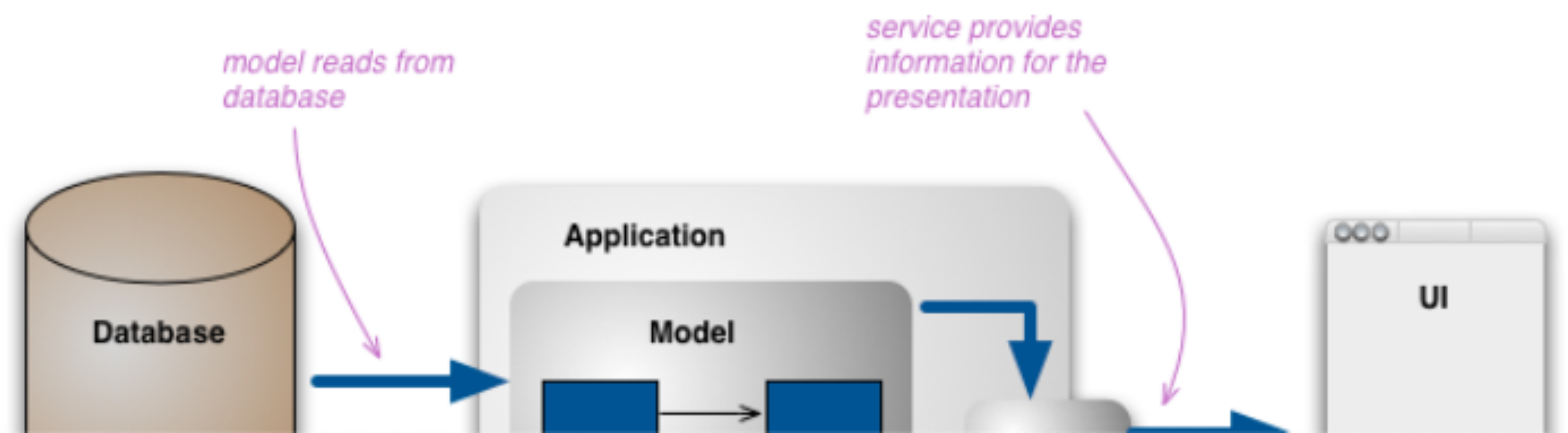
Martin Fowler

DOMAIN DRIVEN DESIGN
APPLICATION ARCHITECTURE
API DESIGN
EVENT ARCHITECTURES

CQRS stands for **Command Query Responsibility Segregation**. It's a pattern that I first heard described by Greg Young. At its heart is the notion that you can use a different model to update information than the model you use to read information. For some situations, this separation can be valuable, but beware that for most systems CQRS adds risky complexity.

The mainstream approach people use for interacting with an information system is to treat it as a CRUD datastore. By this I mean that we have mental model of some record structure where we can **c**reate new records, **r**ead records, **u**pdate existing records, and **d**elete records when we're done with them. In the simplest case, our interactions are all about storing and retrieving these records.

As our needs become more sophisticated we steadily move away from that model. We may want to look at the information in a different way to the record store, perhaps collapsing multiple records into one, or forming virtual records by combining information for different places. On the update side we may find validation rules that only allow certain combinations of data to be stored, or may even infer data to be stored that's different from that we provide.


model reads from database
service provides information for the presentation
Database
Application
Model
UI

One data structure for reads,
another for writes.

Not really event driven per se.

But often combined with.

## Don’t just take my word for it…

youtube.com

Search

Software

Two Days

SUBSCRIBE;

0:25 / 50:05 • What people mean by EDA

#GOTOcon #GOTOchgo #EventDrivenArchitecture

GOTO 2017 • The Many Meanings of Event-Driven Architecture • Martin Fowler

319,355 views • May 11, 2017

5.4K 107 SHARE SAVE

GOTO Conferences
211K subscribers

SUBSCRIBE

This presentation was recorded at GOTO Chicago 2017. #GOTOcon #GOTOchgo
http://gotochgo.com

Greg Young — A Decade of DDD, CQRS, Event Sourcing
Domain-Driven Design Europe
122K views • 4 years ago
48:04

All | Microservices | Kubernetes | Databases

Creating event-driven microservices: the why, how...
Devoxx
32K views • 1 year ago
39:01

GOTO 2012 • Introduction to NoSQL • Martin Fowler
GOTO Conferences
843K views • 8 years ago
54:52

GOTO 2020 • When To Use Microservices (And When Not...
GOTO Conferences
397K views • 5 months ago
38:44

Four Distributed Systems Architectural Patterns by Tim...
Devoxx
544K views • 3 years ago
50:01

GOTO 2019 • "Good Enough" Architecture • Stefan Tilkov
GOTO Conferences
175K views • 10 months ago
41:41

What is DDD - Eric Evans - DDD Europe 2019
Domain-Driven Design Europe
71K views • 1 year ago
57:06

GOTO 2018 • Functional Programming in 40 Minutes •...
GOTO Conferences
504K views • 2 years ago
41:35

AWS re:Invent 2019: [REPEAT 1] Moving to event-driven...
AWS Events
19K views • 1 year ago
53:25

# Event Streaming

Event streaming can be thought of as flipping the flow of requests in the opposite direction. Instead of a standard call-response cadence to communicate with each other, services simply emit an “event”—any significant change-of-state—that is open for any interested services to consume. Some frameworks like Spring Cloud Stream ease the integration with these messaging services.

## Guides

**Integrate Spring Cloud Data Flow Applications with a Scalable MongoDB Deployment on Kubernetes**

Integrate a Spring Cloud Data Flow app with a MongoDB service running on Kubernetes.

Read more

**Build a Scalable, Fault-Tolerant Messaging Cluster on Kubernetes with Apache Kafka and MongoDB**

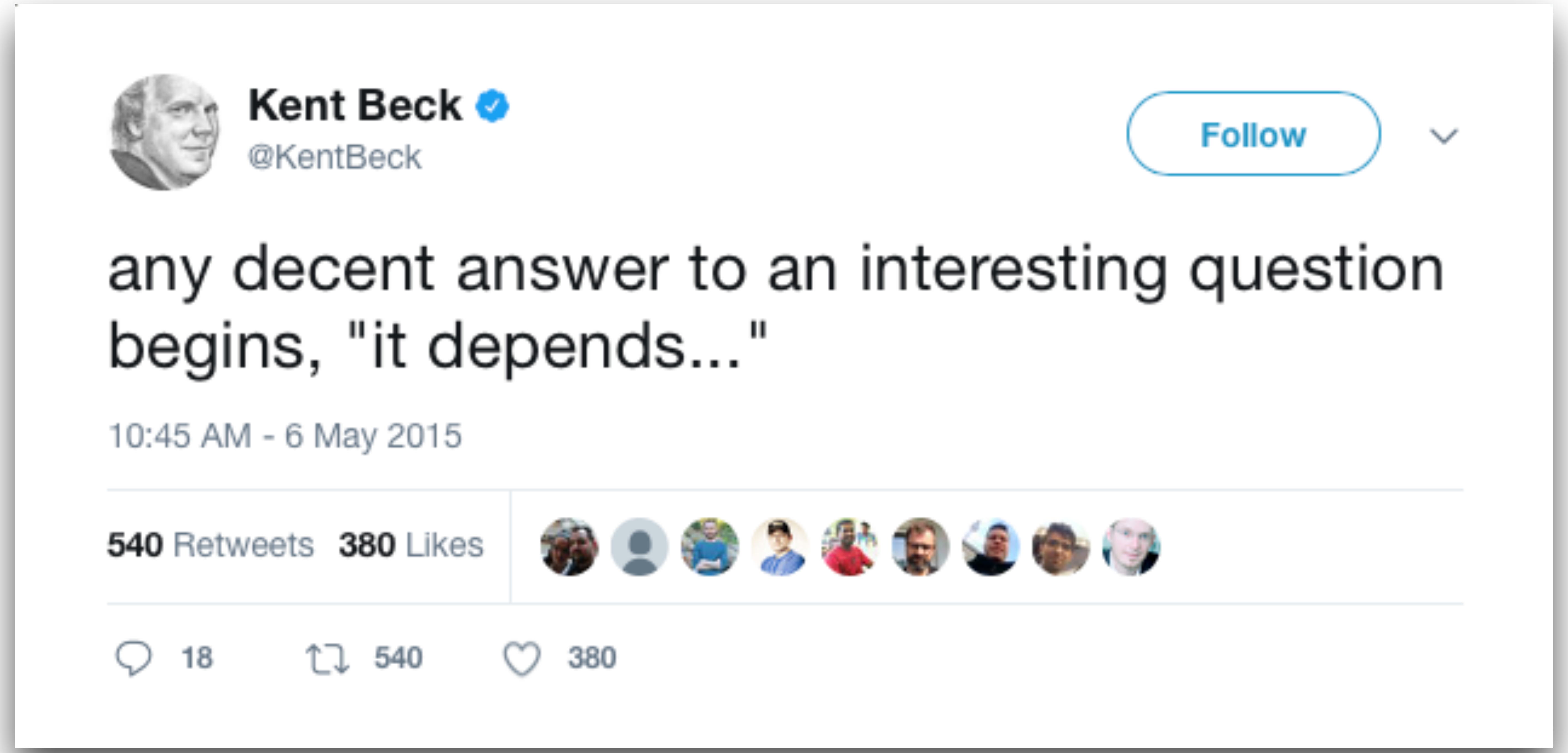
Integrate Kafka with MongoDB to create scalable, fault-tolerant messaging on Kubernetes

Read more

**Getting Started with Spring Cloud Stream**

Read more

## Which approach is right for you?



Kent Beck
@KentBeck

Follow

any decent answer to an interesting question begins, "it depends..."

10:45 AM - 6 May 2015

540 Retweets 380 Likes

18 540 380

https://twitter.com/KentBeck/status/596007846887628801

All about trade offs!

But you knew that.

By the way, this is usually when someone asks…

“How do distributed transactions work in the cloud?”

They don’t!

It’s like the real world.

You buy a shirt at a store.
There is a return policy.

They don’t keep the transaction open until the return period expires!

The sale is committed!

You return the shirt?

## Compensating transactions!

Put the shirt back into inventory.
Issue you a credit.

Same thing for us.

## OK, so how does Spring help us out when it comes to events?

## Spring Cloud Stream.

Here’s the thing.

As architects, we want flexibility.

One constant - change.

Architecture is often defined as the decisions that are hard to change.

Or the decisions we wish we got right.

But we *know* things will change!

We don’t want to paint ourselves into a corner…

SCS lets you swap brokers.

Use what is right for your team.

Supports what you’d expect.

Kafka, RabbitMQ, Kinesis plus various partner maintained bits.

Provides a binder to the external brokers.

Middleware neutral.

Also includes a test binder for integration testing.

You can alway build
your own binder…

Destination binder connects you to your messaging system.

Handles the boilerplate configuration bits.

Bindings are the bridge between your app and the broker.

Your functions then consume and produce messages.

There are binder specific health indicators.

And so very much more!

youtube.com

Search

All | Spring Framework | Related | From Sp

https://github.com/joshlong/**spring-tips** | spring.io/guides

spring tips

Josh Long

龍之春, 龙之春, जोश, ಜೋಶ

@starbuxman
josh@joshlong.com
joshlong.com
github.com/joshlong

0:11 / 30:42

Spring Tips: Spring Cloud Stream

26,885 views • Nov 21, 2016

261 | 5 | SHARE | SAVE

SpringDeveloper
158K subscribers

SUBSCRIBED

Speaker: Josh Long
https://www.twitter.com/starbuxman

Spring Cloud Stream - Developer Recipes, What's... (1:11:19)
SpringDeveloper
5.7K views • 2 years ago

Spring Tips: Spring Native 0.10.0 (1:25:04)
SpringDeveloper
11K views • 4 weeks ago

Webinar: Stream Processing in the Cloud with Data... (1:12:50)
SpringDeveloper
3.1K views • 4 years ago

Spring Tips: Spring Cloud Stream Kafka Streams (54:23)
SpringDeveloper
47K views • 3 years ago

Mix - SpringDeveloper
YouTube

Building microservices with event sourcing and CQRS (1:24:33)
SpringDeveloper
89K views • 5 years ago

From Zero to Hero with Spring WebSocket (1:31:00)
SpringDeveloper
87K views • 5 years ago

Introduction to Spring Cloud (52:12)
SpringDeveloper
11K views • 9 months ago

A Deep Dive into Spring Application Events (52:25)
SpringDeveloper
5.1K views • 9 months ago

# Spring Cloud Stream - demystified and simplified

**ENGINEERING | OLEG ZHURAKOUSKY | OCTOBER 14, 2019 2 COMMENTS**

This is the first post in a series of blog posts meant to clarify and preview what's coming in the upcoming releases of spring-cloud-stream and spring-cloud-function (both 3.0.0).

Recently, I had a discussion with a user and heard something that prompted me to begin a series of blog posts (starting with this one) with the goal of both demystifying the true goals of *Spring Cloud Stream* and *Spring Cloud Function* projects as well as demonstrating their new features.

## Spring Integration Wrapper?

The specific phrase that prompted all this was - *"Spring Cloud Stream, being a light Spring Integration input/output router…"*. That's an interesting perception, but I have to disagree. While it may have been inspired by Enterprise Integration Patterns (EIP) and builds on top of Spring Integration (SI), that last part is really just an implementation detail. Spring Cloud Stream (SCSt) as a framework was never about *"being a light Spring Integration input/output router"*. In fact, this statement shows part of the problem, where SI (the framework of choice to support some of the internal requirements of SCSt) was somehow perceived to be the core of SCSt in such way that many perceive SCSt to be an extension or a wrapper to SI. It is not. It has always been about pure microservices and binding them to *sources* and *targets* of data (i.e., messaging systems) . Simple as that.
If you abstract yourself far enough from knowing the internals of SCSt, you quickly realize that it is really a binding and activation framework. It binds a piece of code (provided by the user) to source/target of data exposed by the binder and activates such code according to binder implementation (for example, message arrival and so on). That is pretty much it.

## To Function or Not to Function?

Historically, Spring Cloud Stream exposed an annotation-based configuration model that required the user to provide a lot of information that could be otherwise easily inferred, thus simplifying

# Spring Cloud Stream - Event Routing

ENGINEERING | OLEG ZHURAKOUSKY | OCTOBER 31, 2019 0 COMMENTS

Welcome to another post in a series of posts showcasing the new features of Spring Cloud Stream (SCSt).
In previous posts (available here, here and here), we tried to provide justification for our shift to a functional programming model in Spring Cloud Stream (SCSt). It is less code and less configuration, and your code remains completely decoupled from the internals of SCSt.

Today, we'll talk about routing with functions.
Routing, in the context of SCSt, is the ability to either *a) route events to a particular event subscriber* or *b) route an event produced by an event subscriber to a particular destination*. To help more with the context, let's quickly look at how things work in the annotation-based programming model. In this post, we'll refer to it as route 'TO' and route 'FROM'.

For routing ***TO*** an event subscriber, we used the `condition` attribute of the `StreamListener` annotation, as follows:

```
@StreamListener(target = Sink.INPUT, condition = "headers['type']=='order'")
public void receiveOrders(Order order) {...}
```

Here are more details on this approach.

And, for routing ***FROM*** an event subscriber, we used Dynamically Bound Destinations - the approach that allows framework to bind to a destination based on some instruction provided within the individual event.

## Event Routing with Functions

With the functional approach, we can do all of the above in a more clean and concise way with a few additional features.

< ALL GUIDES

Event Streaming

Spring Cloud Stream >

Spring Cloud Stream Kafka >

Guides / Event Streaming

# Event Streaming

Event streaming can be thought of as flipping the flow of requests in the opposite direction. Instead of a standard call-response cadence to communicate with each other, services simply emit an “event”—any significant change-of-state—that is open for any interested services to consume. Some frameworks like Spring Cloud Stream ease the integration with these messaging services.

## Spring Cloud Stream

Discover how to use Spring Cloud Stream, a framework for building highly scalable, event-driven microservices connected with shared messaging systems.

## Spring Cloud Stream Kafka

A simple demonstration of how to implement your Java application with Kafka (Spring Cloud Stream) with the least amount of code in your Spring Boot application.

youtube.com
YouTube
Search
CONFLUENT
Welcome to #LiveStreams
with Viktor Gamov
1:29
Streams
0:03 / 1:09:30 • Streams starts
Top chat replay
HIDE CHAT REPLAY
All
Apache Kafka
Computer Application

Live chat replay is on. Messages that appeared when the stream was live will show up here.

Viktor Gamov are you ready for today's stream?
Gnu Gnu hello
Maksym Koval Hi! I have a question regarding AWS XRay and Kafka
Maksym Koval I wanted to create ConsumerInterceptor but it consumes multiple records. Any options?
Luan Carvalho Hello from Brazil !
Gnu Gnu Nice to meet you from S.Korea!

#confluent #apachekafka #streamprocessing

Streaming Processing and Testing with Spring Cloud Stream | Livestreams 020

1,000 views • Streamed live on Feb 9, 2021

36 1 SHARE SAVE

Confluent
27K subscribers

SUBSCRIBE

In this episode of Livestreams, Developer Advocate Viktor Gamov talks about how to use Kafka Streams' TopologyTestDriver with a Spring Cloud Stream app. Also, Viktor shows how you can enable Protobuf support in a Spring Cloud Stream app using Confluent Schema Registry.

Spring Cloud Stream and Confluent Cloud | Livestream...
Confluent
1.1K views •
Streamed 5 months ago
1:02:50

Reactive Kafka
SpringDeveloper
30K views • 4 years ago

youtube.com

YouTube

Search

https://github.com/joshlong/**spring-tips** | spring.io/guides

spring tips

Josh Long

龍之春, 龙之春, जोश, ಜೋಶ

@starbuxman
josh@joshlong.com
joshlong.com
github.com/joshlong

0:11 / 54:22

Spring Tips: Spring Cloud Stream Kafka Streams

47,045 views • Apr 10, 2018

423 11 SHARE SAVE

SpringDeveloper
158K subscribers

SUBSCRIBED

Hi Spring fans! In this installment of Spring Tips we look at stream processing in Spring Boot applications with Apache Kafka, Apache Kafka Streams and the Spring Cloud Stream Kafka Streams binder.

Health & Safety Program

Plan A Getaway
Ad GreatWolf.com

BOOK TODAY

All | Spring Framework | Related | From Sp

Streaming with Spring Cloud Stream and Apache Kafka
SpringDeveloper
24K views • 1 year ago
59:26

Spring Tips: Spring Cloud Gateway (Redux)
SpringDeveloper
48K views • 4 months ago
1:39:09

Cloud Native Java - Josh Long
SpringDeveloper
47K views • 4 years ago
1:49:20

Spring Cloud Stream - Developer Recipes, What's...
SpringDeveloper
5.7K views • 2 years ago
1:11:19

Spring Tips: Spring Cloud Stream
SpringDeveloper
26K views • 4 years ago
30:43

Developing Real-Time Data Pipelines with Apache Kafka
SpringDeveloper
142K views • 5 years ago
1:30:40

Kafka Streams - From the Ground Up to the Cloud -...
SpringDeveloper
8K views • 3 years ago
37:57

Mix - SpringDeveloper
YouTube

# CONTRACTS

Services evolve.

To be expected!

## How do you avoid breaking changes to consumers?

You might not even know who is calling your service!

## Consumer Driven Contracts.

https://martinfowler.com/articles/consumerDrivenContracts.html

spring.io

spring
Why Spring
Learn
Projects
Training
Support
Community

Spring Boot

Spring Framework

Spring Data

Spring Cloud

- Spring Cloud Azure
- Spring Cloud Alibaba
- Spring Cloud for Amazon Web Services
- Spring Cloud Bus
- Spring Cloud CLI
- Spring Cloud for Cloud Foundry
- Spring Cloud - Cloud Foundry Service Broker
- Spring Cloud Cluster
- Spring Cloud Commons
- Spring Cloud Config
- Spring Cloud Connectors
- Spring Cloud Consul
- Spring Cloud Contract
- Spring Cloud Function
- Spring Cloud Gateway
- Spring Cloud GCP
- Spring Cloud Netflix
- Spring Cloud Open Service Broker
- Spring Cloud Pipelines
- Spring Cloud Schema Registry
- Spring Cloud Security
- Spring Cloud Skipper

# Spring Cloud Contract 3.0.0

OVERVIEW LEARN SAMPLES

Spring Cloud Contract is an umbrella project holding solutions that help users in successfully implementing the Consumer Driven Contracts approach. Currently Spring Cloud Contract consists of the Spring Cloud Contract Verifier project.

Spring Cloud Contract Verifier is a tool that enables Consumer Driven Contract (CDC) development of JVM-based applications. It is shipped with Contract Definition Language (DSL) written in Groovy or YAML. Contract definitions are used to produce following resources:

- by default JSON stub definitions to be used by WireMock (HTTP Server Stub) when doing integration testing on the client code (client tests). Test code must still be written by hand, test data is produced by Spring Cloud Contract Verifier.
- Messaging routes if you're using one. We're integrating with Spring Integration, Spring Cloud Stream and Apache Camel. You can however set your own integrations if you want to.
- Acceptance tests (by default in JUnit or Spock) used to verify if server-side implementation of the API is compliant with the contract (server tests). Full test is generated by Spring Cloud Contract Verifier.

Spring Cloud Contract Verifier moves TDD to the level of software architecture.

To see how Spring Cloud Contract supports other languages just check out this blog post.

## Features

When trying to test an application that communicates with other services then we could do one of two things:

- deploy all microservices and perform end to end tests
- mock other microservices in unit / integration tests

Both have their advantages but also a lot of disadvantages. Let's focus on the latter. Deploy all

Contracts can be written in Groovy or YAML.

Add the Spring Cloud Contract Verifier dependency.

Running a clean build will generate test stubs.

You implement the test code.

Publish the stub artifacts along with your production code.

Living documentation!

Evolves with your services.

Consumers leverage Spring Cloud Contract Stub Runner.

Add the dependency and install the producer stubs.

Add the proper annotation to your test class.

Your test will get a stubbed version of the HTTP response.

spring.io

spring

Why Spring · Learn · Projects · Training · Support · Community

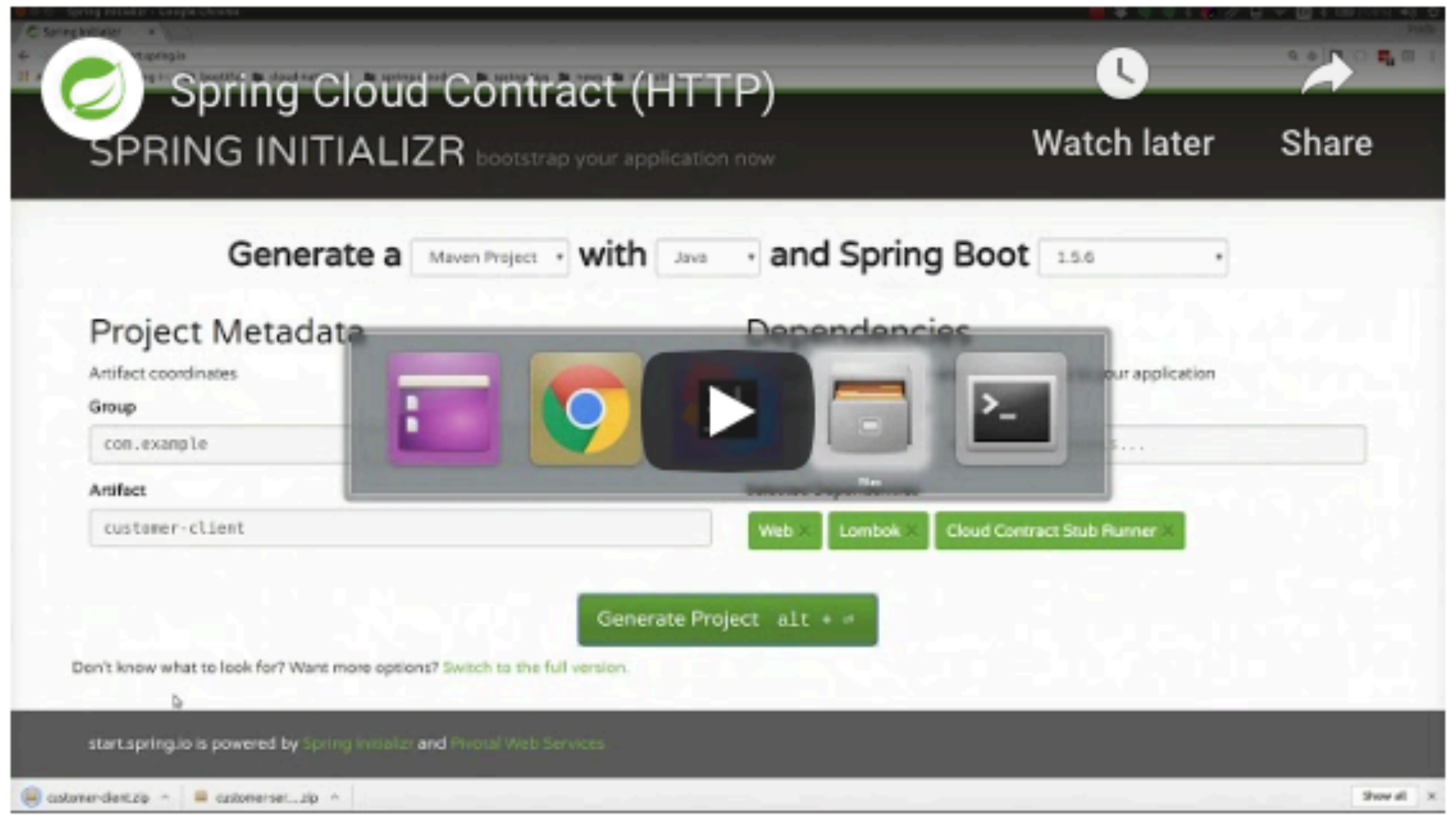
**Spring Blog**  All Posts · Engineering · Releases · News and Events

# Spring Tips: Spring Cloud Contract (HTTP)

**ENGINEERING | JOSH LONG | OCTOBER 24, 2017 11 COMMENTS**

Speaker: Josh Long

Hi Spring fans! In this tip, we'll look at how to get fast-feedback *and* do integration testing with Spring Cloud Contract. In this video, we'll focus on HTTP-based services.


Spring Cloud Contract (HTTP)
SPRING INITIALIZR bootstrap your application now
Watch later
Share
Generate a Maven Project with Java and Spring Boot 1.5.6
Project Metadata
Artifact coordinates
Group
com.example
Artifact
customer-client
Dependencies
Web
Lombok
Cloud Contract Stub Runner
Generate Project alt + ⏎
Don't know what to look for? Want more options? Switch to the full version.
start.spring.io is powered by Spring Initializr and Pivotal Web Services

**11 Comments**  spring.io  Disqus' Privacy Policy  Login

Recommend 2  Tweet  Share  Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

## Get the Spring newsletter

Email Address

☐ Yes, I would like to be contacted by The Spring Team and VMware for newsletters, promotions and events per the terms of VMware's Privacy Policy

SUBSCRIBE

youtube.com
YouTube
Search
All
Spring Framework
Related
From Spring
0:04 / 1:07:53
SHARE
SAVE
SUBSCRIBE

Consumer Driven Contract Testing with Spring Cloud Contract

8,494 views • Oct 3, 2018

66 6

SpringDeveloper
146K subscribers

In the microservices world, testing can sound easy since developers should focus in just one domain. In the real world, testing microservices can be tough as we also need to take into account the communication between the many services in our system. In this session about Spring Cloud

| | |
|---|---|
| Consumer Driven Contracts and Your Microservice Architectur... | SpringDeveloper · 6K views • 3 years ago · 1:03:33 |
| Webinar: Consumer Driven Contracts and Your... | SpringDeveloper · 7.1K views • 4 years ago · 1:24:21 |
| How Fast is Spring? | SpringDeveloper · 14K views • 2 years ago · 1:15:25 |
| The Beginner's Guide To Spring Cloud - Ryan Baxter | SpringDeveloper · 90K views • 3 years ago · 1:08:42 |
| Testing Your Message-Driven Application | SpringDeveloper · 2.8K views • 2 years ago · 1:02:38 |
| Living on the Edge with Spring Cloud Gateway | SpringDeveloper · 15K views • 2 years ago · 28:42 |
| Introduction to Spring Cloud | SpringDeveloper · 4K views • 3 months ago · 52:12 |
| Multi-Service Reactive Streams Using Spring, Reactor, and... | SpringDeveloper · 7.1K views • 2 years ago · 1:13:19 |
| Introducing Spring Cloud Gateway and API Hub for... | SpringDeveloper · 1K views • 3 months ago · 55:10 |

# REACTIVE

## Remember Moore’s Law?

“the number of transistors in a [chip] doubles about every two years.”

Processor clock speeds got faster and faster.

A new chip would be twice as fast as a six month old model.

Today? Not so much..but, we have more cores.

That reality changes things for how we architect systems.

Take advantage of the cores!

Many apps need high throughput and low latency.

## Non-blocking, asynchronous applications to the rescue!

Do more with fewer resources.

Spring gives you two stacks:
Reactive and Servlet.

Spring MVC builds on the Servlet API.

## Synchronous, blocking IO.

Aka - the traditional approach.

And that’s fine!

Still fits a number of use cases!

But we do have options today…

WebFlux is non-blocking, taking advantage of multi-core processors.

Designed for massive concurrent connections.

Project Reactor interacts with functional API of Java.

Twp APIs: Flux and Mono.

Back pressure ready,
fully non-blocking.

Low memory footprint, tens of millions of messages per second.

Datastores have evolved too.

There are a set of reactive repositories.

Spring Data has native support for Mongo, Redis and Cassandra.

Others are supported via R2DBC.

You may not always need reactive architectures…

But when you do,
Spring has you covered!

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community

# Reactive

Reactive systems have certain characteristics that make them ideal for low-latency, high-throughput workloads. Project Reactor and the Spring portfolio work together to enable developers to build enterprise-grade reactive systems that are responsive, resilient, elastic, and message-driven.

**What is reactive processing?**

Reactive processing is a paradigm that enables developers build non-blocking, asynchronous applications that can handle back-pressure (flow control).

**Why use reactive processing?**

Reactive systems better utilize modern processors. Also, the inclusion of back-pressure in reactive programming ensures better resilience between decoupled components.

## Project Reactor

Project Reactor is a fully non-blocking foundation with back-pressure support included. It's the foundation of the reactive stack in the Spring ecosystem and is featured in projects such as Spring WebFlux, Spring Data, and Spring Cloud Gateway.

Learn more

youtube.com

YouTube

Search

DEVOXX™

https://github.com/joshlong/flux-flix-service | spring.io/guides

REACTIVE
SPRING

Josh Long
(龍之春, 龙之春, जोश, ಜೊಶ)
@starbuxman
josh@joshlong.com
github.com/joshlong

UNITED STATES    @DevoxxUS

0:05 / 56:28

Reactive Spring by Josh Long

54,021 views • Apr 11, 2017

537  28  SHARE  SAVE

Devoxx
122K subscribers

SUBSCRIBE

Subscribe to Devoxx on YouTube @ https://bit.ly/devoxx-youtube
Like Devoxx on Facebook @ https://www.facebook.com/devoxxcom
Follow Devoxx on Twitter @ https://twitter.com/devoxx

All  Spring Framework  Microservices

Reactive Spring • Josh Long • GOTO 2019
GOTO Conferences
38K views • 2 years ago
53:42

lofi hip hop radio - beats to relax/study to
Lofi Girl
33K watching
LIVE NOW

JUnit 5: the Next Step in Automated Testing
SpringDeveloper
23K views • 2 years ago
30:29

ITkonekt 2019 | Robert C. Martin (Uncle Bob), Clean...
IT konekt
122K views • 2 years ago
1:11:59

Design Microservice Architectures the Right Way
InfoQ
459K views • 2 years ago
48:30

Mix - Devoxx
YouTube

Reactive Frontends with RxJS and Angular - Sergi Almar
SpringDeveloper
12K views • 3 years ago
1:06:30

Reactive Spring Security 5.1 by Example
SpringDeveloper
20K views • 2 years ago
1:08:47

What is Spring Webflux and when to use it?
Defog Tech
148K views • 2 years ago
17:46

youtube.com

Search

All · Spring Framework · Microservices

Spring Tour - Spring Tips and Reactive Spring Day 1 - with Josh Long

2,059 views • Jul 22, 2020

48 · 0 · SHARE · SAVE

VMware Tanzu
22.3K subscribers

ANALYTICS · EDIT VIDEO

Hi, Spring fans! Spring is an inversion-of-control container. Inversion of control insulates objects from the knowledge of how a collaborating object on which they depend are created. It also insulates objects from their environment, letting them exist blissfully unaware of where

Servlet or Reactive Stacks: The Choice is Yours. Oh No..... — SpringDeveloper — 14K views • 3 years ago — 1:11:06

Mix - VMware Tanzu — YouTube

What Is Right To Repair? — Marques Brownlee — 2.9M views • 2 months ago — 20:16

Reactive Spring with Josh Long — SouJava — 1.4K views • Streamed 9 months ago — 1:42:15

Tanzu Talk: Best Practices for Running Containers and... — VMware Tanzu — 170 views • Streamed 3 weeks ago — 44:05

Reactive Spring. Josh Long, Pivotal — Devoxx — 12K views • 2 years ago — 57:13

Kubernetes Tutorial for Beginners [FULL COURSE in ... — TechWorld with Nana — 1.4M views • 8 months ago — 3:36:55

Inside the mind of a master procrastinator | Tim Urban — TED — 40M views • 5 years ago — 14:04

Bootiful TDD by Josh Long @ Spring I/O 2019 — Spring I/O — 14K views • 2 years ago — 54:15

# SC GATEWAY

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community
Spring Boot
Spring Framework
Spring Data
Spring Cloud
Spring Cloud Azure
Spring Cloud Alibaba
Spring Cloud for Amazon Web Services
Spring Cloud Bus
Spring Cloud CLI
Spring Cloud for Cloud Foundry
Spring Cloud - Cloud Foundry Service Broker
Spring Cloud Cluster
Spring Cloud Commons
Spring Cloud Config
Spring Cloud Connectors
Spring Cloud Consul
Spring Cloud Contract
Spring Cloud Function
Spring Cloud Gateway
Spring Cloud GCP
Spring Cloud Netflix
Spring Cloud Open Service Broker
Spring Cloud Pipelines
Spring Cloud Schema Registry
Spring Cloud Security
Spring Cloud Skipper
OVERVIEW
LEARN
SAMPLES
COPY

# Spring Cloud Gateway 3.0.0

This project provides a library for building an API Gateway on top of Spring WebFlux. Spring Cloud Gateway aims to provide a simple, yet effective way to route to APIs and provide cross cutting concerns to them such as: security, monitoring/metrics, and resiliency.

## Features

Spring Cloud Gateway features:

- Built on Spring Framework 5, Project Reactor and Spring Boot 2.0
- Able to match routes on any request attribute.
- Predicates and filters are specific to routes.
- Circuit Breaker integration.
- Spring Cloud DiscoveryClient integration
- Easy to write Predicates and Filters
- Request Rate Limiting
- Path Rewriting

## Getting Started

```
 @SpringBootApplication
public class DemogatewayApplication {
        @Bean
        public RouteLocator customRouteLocator(RouteLocatorBuilder builder) {
                return builder.routes()
                        .route("path_route", r -> r.path("/get")
                                .uri("http://httpbin.org"))
                        .route("host_route", r -> r.host("*.myhost.org")
```

## A Spring approach to the gateway problem!

Based on Spring 5,
Reactor and Boot 2.

## Non blocking IO.

## Backpressure.

## Event loop!

## Spring WebFlux.

Lives along side Spring MVC.

Non-blocking, reactive.

Streams!

HandlerMapping - what code is going to handle this request.

WebFilter - manipulate the request/response.

Predicate - test some aspect of the request…

And determine whether to route it.

ServerWebExchange: access all parts of the http request/response.

Configure routes in Java, YAML
or via repositories.

Can route on path, host, headers, parameters…

Anything in the request.

## Filters!

Rewrite path.

Add or remove request/ response headers.

## Rate limiting.

## Circuit Breaker integration.

With so many options, why should I use Spring Cloud Gateway?

It is programmer centric routing.

Antithesis of tickets with a side of tickets. And more tickets.

Would you rather
refresh a configuration?

Or fill out another ticket?

Java centric, Spring centric, configuration centric.

You are in control.

Instead of one of these…

You can craft your own…

Except *you* decide what tools, blades etc. you want.

Lightweight, simple.

Use it as you will. You aren’t forced down a certain path.

Think of it as an ESB with inversion of control.

It is not a SaaS, it is a tool.

You can just “run” SC Gateway.

It is just an app.

Developer focussed.

You know how to build
and run applications.

You build it, you push it.

It is in your hands, not some random enterprise group.

Anything you could do in Zuul 1 is supported in SC Gateway.

## But what about performance?

There was a benchmark published in December 2017.

SC Gateway was not officially released at that time.

There are no performance issues today.

Many large companies rely on it.

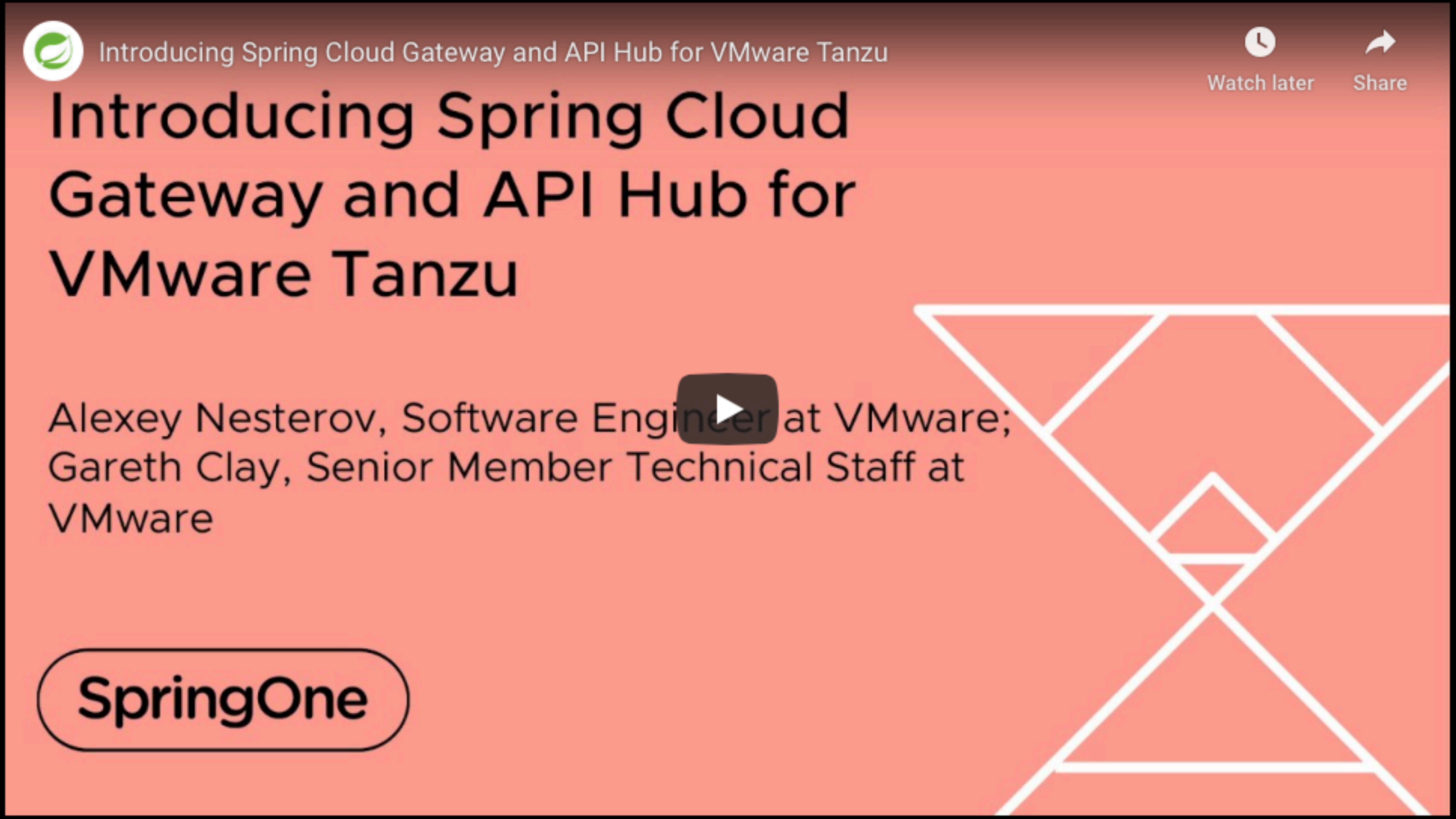
springone.io

SpringOne

FAQ Sign Out

Schedule

# Introducing Spring Cloud Gateway and API Hub for VMware Tanzu


Introducing Spring Cloud Gateway and API Hub for VMware Tanzu

Watch later Share

Introducing Spring Cloud Gateway and API Hub for VMware Tanzu

Alexey Nesterov, Software Engineer at VMware; Gareth Clay, Senior Member Technical Staff at VMware

SpringOne

LEARN MORE

Presentation slides

Spring Cloud Gateway: Getting Started

Spring Cloud Gateway for VMware Tanzu was made generally available in January of this year. In this talk, we'll introduce this new product, which gives developers the ability to create easily configurable, on-demand API Gateway

youtube.com
YouTube
Search
0:04 / 1:10:31
92
0
SHARE
SAVE
SUBSCRIBE
All
Spring Framework
Related
From Spring

Reactive Architectures with RSocket and Spring Cloud Gateway

5,371 views • Oct 16, 2019

SpringDeveloper
146K subscribers

Spring Cloud Gateway introduced a flexible API for configuring the edge service of our applications. Built upon Project Reactor, Spring WebFlux, and Spring Boot 2.0, Spring Cloud Gateway provided a modern and efficient architecture to handle a large number of concurrent requests. However, as

Multi-Service Reactive Streams Using Spring, Reactor, and...
SpringDeveloper
7.1K views • 2 years ago
1:13:19

Living on the Edge with Spring Cloud Gateway
SpringDeveloper
15K views • 2 years ago
28:42

The Value Of Reactive Systems
SpringDeveloper
3.9K views • 1 year ago
1:04:38

Spring Cloud Gateway - Spencer Gibb, Sree Tummidi
SpringDeveloper
12K views • 3 years ago
1:10:39

Mastering Spring Boot's Actuator
SpringDeveloper
21K views • 2 years ago
1:05:48

Spring Boot Omakase: A Fast-Paced "Chef's Choice" Dive int...
SpringDeveloper
858 views • 3 months ago
56:09

Spring Cloud on AWS
SpringDeveloper
24K views • 3 years ago
30:29

Spring Cloud Stream - Developer Recipes, What's Ne...
SpringDeveloper
5K views • 2 years ago
1:11:19

Mix - SpringDeveloper
YouTube

youtube.com
YouTube
Search
0:04 / 36:31
SHARE
SAVE
SUBSCRIBE
All
Spring Framework
Related
From Spring

Spring Cloud Gateway for Stateless Microservice Authorization

17,838 views • Oct 16, 2019

206 16

SpringDeveloper
146K subscribers

Improving and maintaining tech agility, time to market, and application modernization is challenging as the number of microservices we own and manage grows. How do you track who uses your applications? How can you establish and enforce common policies or flows to

Implementing Microservices Security Patterns and Protocol...
SpringDeveloper
19K views • 1 year ago
1:10:35

Developing microservices with aggregates - Chris Richardson
SpringDeveloper
171K views • 3 years ago
1:09:50

Spring Tips: Spring Cloud Gateway
SpringDeveloper
23K views • 3 years ago
40:00

Cloud Native Java - Josh Long
SpringDeveloper
43K views • 4 years ago
1:49:20

Living on the Edge with Spring Cloud Gateway
SpringDeveloper
15K views • 2 years ago
28:42

Multi-tenancy OAuth with Spring Security 5.2
SpringDeveloper
11K views • 1 year ago
1:11:23

Securing OAuth 2.0 Resources in Spring Security 5.0
SpringDeveloper
35K views • 2 years ago
1:11:14

Reactive Spring Security 5.1 by Example
SpringDeveloper
17K views • 2 years ago
1:08:47

Spring Cloud Gateway - Spencer Gibb, Sree Tummidi
SpringDeveloper
12K views • 3 years ago
1:10:39

# Spring Cloud Kubernetes

## Did someone say Kubernetes?

## Spring Cloud Kubernetes!

Spring Boot

Spring Framework

Spring Data

Spring Cloud

- Spring Cloud Azure
- Spring Cloud Alibaba
- Spring Cloud for Amazon Web Services
- Spring Cloud Bus
- Spring Cloud Circuit Breaker
- Spring Cloud CLI
- Spring Cloud for Cloud Foundry
- Spring Cloud - Cloud Foundry Service Broker
- Spring Cloud Cluster
- Spring Cloud Commons
- Spring Cloud Config
- Spring Cloud Connectors
- Spring Cloud Consul
- Spring Cloud Contract
- Spring Cloud Function
- Spring Cloud Gateway
- Spring Cloud GCP
- Spring Cloud Kubernetes
- Spring Cloud Netflix
- Spring Cloud Open Service Broker
- Spring Cloud OpenFeign
- Spring Cloud Pipelines
- Spring Cloud Schema Registry
- Spring Cloud Security
- Spring Cloud Skipper

## Spring Cloud Kubernetes 2.0.2

OVERVIEW LEARN SAMPLES

Spring Cloud Kubernetes provides implementations of well known Spring Cloud interfaces allowing developers to build and run Spring Cloud applications on Kubernetes. While this project may be useful to you when building a cloud native application, it is also not a requirement in order to deploy a Spring Boot app on Kubernetes. If you are just getting started in your journey to running your Spring Boot app on Kubernetes you can accomplish a lot with nothing more than a basic Spring Boot app and Kubernetes itself. To learn more, you can get started by reading the Spring Boot reference documentation for deploying to Kubernetes and also working through the workshop material Spring and Kubernetes.

### Features

- Kubernetes awareness
- `DiscoveryClient` implementation
- `PropertySource` objects configured via ConfigMaps
- Client side loadbalancing via Netflix Ribbon

### Getting Started

The easiest way to get started is by including the Spring Cloud BOM and then adding `spring-cloud-starter-kubernetes-all` to your application's classpath. If you don't want to include all of the Spring Cloud Kubernetes features you can add individual starters for the features you would like. By default Spring Cloud Kubernetes will enable the `kubernetes` profile when it detects it is running inside a Kubernetes cluster. You can take advantage of this by creating a `kubernetes-application` configuration properties for anything specific to Kubernetes you might want to configure. Once the starter is on the classpath the application should behave as any other Spring Cloud application.

**Quickstart Your Project**

Bootstrap your application with Spring Initializr.

First off, not required.

Boot is ready to roll on a multitude of popular cloud options.

Spring Boot will auto detect Kubernetes.

You can export the K8s probes via Actuator.

## Liveness and Readiness.

Just point your configuration to the actuator endpoints.

## What does Spring Cloud Kubernetes give me?

Kuberentes awareness.

DiscoveryClient implementation.

PropertySource objects.

## Client side load balancing via Ribbon.

You are free to add individual K8s starters…

Or pull in everything.

You can also use Spring to extend Kubernetes.

## Custom Resource Definition.

## Evolving space!

spring.io

Why Spring  Learn  Projects  Training  Support  Community

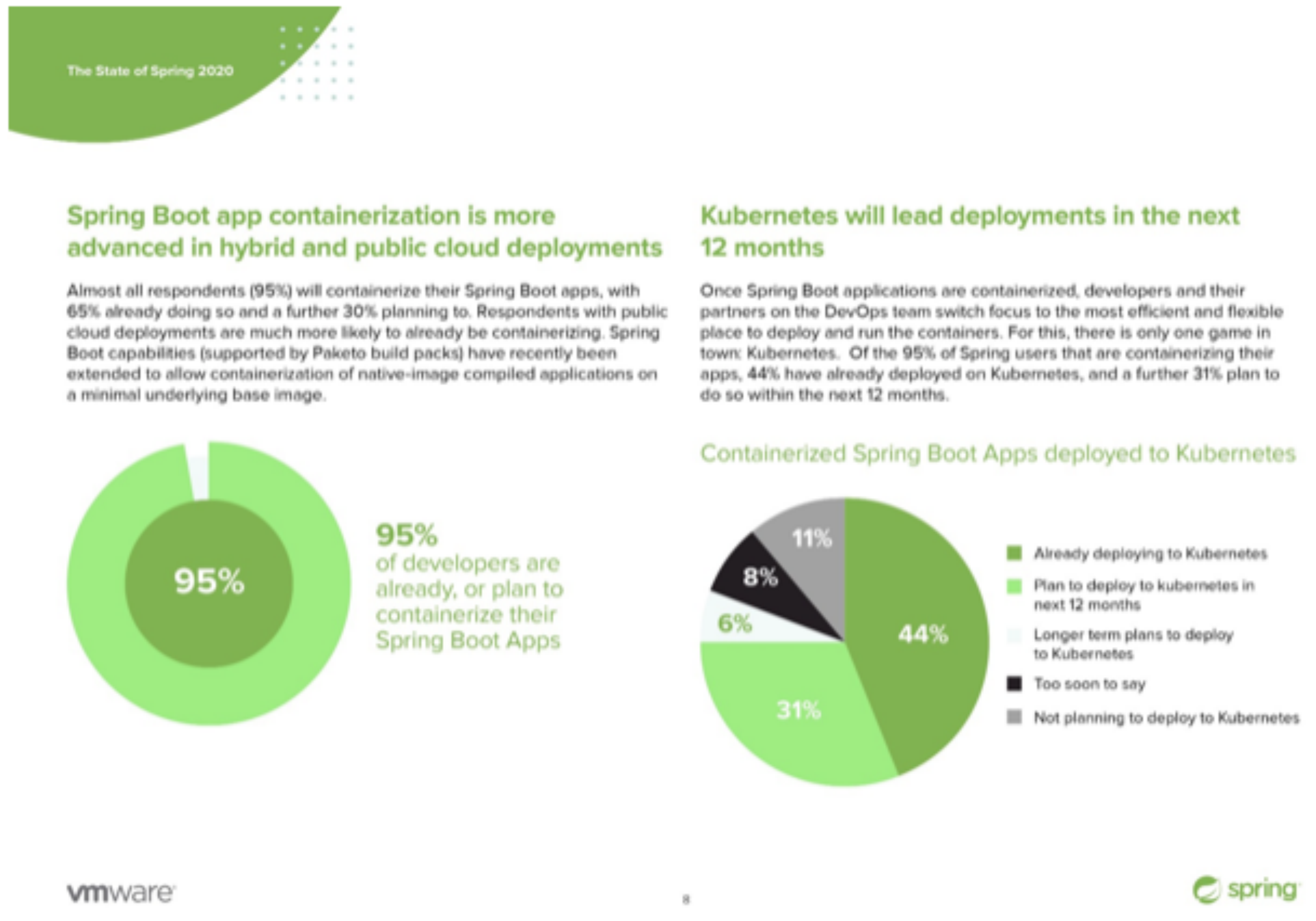
ALL GUIDES

# Spring on Kubernetes

**Get the Code**

Go To Repo

When it comes to building Java apps that run in the cloud, Spring and Spring Boot are clear favorites. It is also increasingly clear that technologies such as Docker and Kubernetes play an important role in the Spring community.


The State of Spring 2020

**Spring Boot app containerization is more advanced in hybrid and public cloud deployments**

Almost all respondents (95%) will containerize their Spring Boot apps, with 65% already doing so and a further 30% planning to. Respondents with public cloud deployments are much more likely to already be containerizing. Spring Boot capabilities (supported by Paketo build packs) have recently been extended to allow containerization of native-image compiled applications on a minimal underlying base image.

**Kubernetes will lead deployments in the next 12 months**

Once Spring Boot applications are containerized, developers and their partners on the DevOps team switch focus to the most efficient and flexible place to deploy and run the containers. For this, there is only one game in town: Kubernetes. Of the 95% of Spring users that are containerizing their apps, 44% have already deployed on Kubernetes, and a further 31% plan to do so within the next 12 months.

95%
95%
of developers are already, or plan to containerize their Spring Boot Apps
Containerized Spring Boot Apps deployed to Kubernetes
44%
31%
6%
8%
11%
Already deploying to Kubernetes
Plan to deploy to kubernetes in next 12 months
Longer term plans to deploy to Kubernetes
Too soon to say
Not planning to deploy to Kubernetes
vmware
8
spring

Packing your Spring Boot app in a Docker container and deploying that application to Kubernetes has been possible for a while now and took very little effort. Due to the "make jar not war" motto, all that was required to containerize a Spring Boot app was a container with a JRE to run the jar. Once you had a Docker container, running the containerized Spring Boot application in Kubernetes was just a matter of running the container.

That said, as more and more of you deployed Spring Boot applications to Kubernetes, it became clear we could do better. To that end, we have made several enhancements in Spring Boot 2.3 and are making even more in the forthcoming Spring Boot 2.4 release to make running Spring Boot on

youtube.com
YouTube
Search
0:02 / 34:17

Getting Started with Spring Cloud Kubernetes

19,377 views • Jul 3, 2019

355 8 SHARE SAVE

SpringDeveloper
155K subscribers

SUBSCRIBED

In this video Ryan Baxter of Pivotal takes a look at Spring Cloud Kubernetes, a project that allows developers to use native Kubernetes features in their Spring applications.

Get Our 8 Figure Copy & Paste Template

Start Copying
Ad www.monetarico.com/live
COPY US

All | Spring Framework | Kubernetes | Relat

Best Practices to Spring to Kubernetes Easier and Faster
SpringDeveloper
14K views • 1 year ago

Spring Tips: Spring Cloud Gateway (Redux)
SpringDeveloper
40K views • 2 months ago

Mix - SpringDeveloper
YouTube

Spring Boot Loves K8s
SpringDeveloper
4.9K views • 7 months ago

Whats New in Spring Boot 2 4
SpringDeveloper
89K views • 3 months ago

Webinar: Introducing Spring Cloud Task
SpringDeveloper
8K views • 4 years ago

Spring Tips: Spring Batch and Apache Kafka
SpringDeveloper
23K views • 1 year ago

Kubernetes for the Spring

Tanzu Tuesdays - Top tips for running Spring Boot applications on Kubernetes with Ollie Hughes

244 views • Jan 28, 2021

VMware Tanzu
21.7K subscribers

Deploying a Spring Application on K8s has become increasingly straightforward with new features in Spring Boot, such as OCI image building and micrometer metric exporting. In this talk, Ollie will demonstrate a number of tools and techniques that make deploying apps to Kubernetes as

Cloud Native Architectures and Microservices – Kenny Bastani
VMware Tanzu
10K views • 3 years ago
49:25

Mix - VMware Tanzu
YouTube

Build Images with Confidence and Ease with Cloud Native...
SpringDeveloper
1.4K views • 1 year ago
20:42

Tanzu Tuesdays - Spring On Kubernetes with Ryan Baxter
VMware Tanzu
421 views • 8 months ago
1:25:01

Docker for Beginners: Full Free Course!
KodeKloud
598K views • 1 year ago
2:09:28

Tanzu TV - Between Chair and Keyboard - The one with...
VMware Tanzu
80 views • 2 weeks ago
1:24:52

Devnexus 2021 Josh Long Bootiful kubernetes
devnexus
244 views • 2 months ago
33:08

Microservices using SpringBoot | Full Example
Daily Code Buffer
167K views • 6 months ago
1:21:39

How to travel the world with almost no money | Tomislav...
TEDx Talks
7.3M views • 6 years ago
18:19

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Tanzu.TV Community

Workshops / Spring on Kubernetes

# Spring on Kubernetes

Create a Spring Boot application. Containerize it, and push the container to a registry. Deploy it to Kubernetes.

Approximate time: 120 minutes

START WORKSHOP *(Login Required)*

During this workshop you will learn the finer details of how to create, build, run, and debug a basic Spring Boot app on Kubernetes by doing the following:

- Create a basic Spring Boot app
- Build a Docker image for the app
- Push the image to a Docker registry
- Deploy and run the app on Kubernetes
- Test the app using port-forwarding and ingress
- Use skaffold to iterate easily as you work on your app
- Use kustomize to manage configurations across environments
- Externalize application configuration using ConfigMaps
- Use service discovery for app-to-app communication
- Deploy the Spring PetClinic App with MySQL

youtube.com
YouTube
Search
#BOOTERNETES
spring.io/video · tanzu.tv · springonetour.io
FEATURING:
JOSH LONG
NATE SCHUTTA
MARK HECKLER
CORA IBERKLEID
PAUL CZARKOWSKI
TIFFANY JERNIGAN
0:00 / 2:08:53
285
2
SHARE
SAVE
SUBSCRIBED
SHOW CHAT REPLAY
SpringOne Tour 2021
SpringDeveloper - 1 / 1
All
Angular
Computer Application

SpringOne Tour 2021: # Booternetes

10,229 views • Premiered Mar 24, 2021

SpringDeveloper
155K subscribers

Join Josh Long, Nate Schutta, Mark Heckler, Cora Iberkleid, Paul Czarkowski, and Tiffany Jernigan for this journey from code with Spring Boot to production on Kubernetes. For more great Tanzu, Kubernetes, and Spring content, check out our VMware Tanzu channel:

SpringOne Tour 2021: # Booternetes
SpringDeveloper
2:08:54

Cloud Native Java - Josh Long
SpringDeveloper
45K views • 4 years ago
1:49:20

Mix - SpringDeveloper
YouTube

Do's and Don'ts: Avoiding First-Time Reactive Programmer...
SpringDeveloper
20K views • 1 year ago
58:38

Developing microservices with aggregates - Chris Richardson
SpringDeveloper
191K views • 4 years ago
1:09:50

How to Get Productive with Spring Boot
SpringDeveloper
13K views • 1 year ago
1:03:56

From Zero to Hero with Spring Boot - Brian Clozel
SpringDeveloper
151K views • 3 years ago
1:09:19

Introduction to Apache Cassandra Workshop
SpringDeveloper

SPRING NATIVE

## How fast is Spring?

Glad you asked!

youtube.com

Search

All | Computer Application | Related | From

How Fast is Spring by Dave Syer @ Spring I/O 2019

2,027 views • Jun 11, 2019

32 | 1 | SHARE | SAVE

Spring I/O
15.6K subscribers

SUBSCRIBE

Spring I/O 2019 - Barcelona, 16-17 May

Slides: http://presos.dsyer.com/decks/how-fas...

0:02 / 53:49

How Fast is Spring?
SpringDeveloper
15K views • 2 years ago
1:15:25

Mix - Spring I/O
YouTube

Serverless Spring with Dave Syer
VMware Tanzu
1.2K views • 1 year ago
1:14:05

How to secure your Spring apps with Keycloak by Thomas...
Spring I/O
9.9K views • 1 year ago
53:15

The State of Java Relational Persistence by Maciej...
Spring I/O
6.7K views • 1 year ago
46:10

Dave Syer, Pivotal—Choosing Software Abstractions,...
SpringDeveloper
4.6K views • 2 years ago
15:40

Spring On Kubernetes by Dave Syer @ Spring I/O Bridge
Spring I/O
2.7K views • 10 months ago
39:46

Colored iMacs? Let's Talk About Apple's 4/20 Event!
Marques Brownlee
4.4M views • 6 days ago
New
16:12

Running Spring Boot applications as GraalVM nativ...
Devoxx
8.4K views • 1 year ago
35:23

The Spring team has always prioritized performance.

And there are a number of things you can tweak.

And now Spring Native.

Currently beta!

Allows you to compile to GraalVM native images.

Nearly instant startup experience and lower memory usage.

But longer build times and fewer runtime optimizations.

## Trade offs!

Native images have different characteristics. That’s the point.

## Ahead of time transformations.

Not everything can be inferred…

Useful for functions.

Lower overhead Microservices.

Supports Java and Kotlin.

Don’t forget. It is in beta.

Expect breaking changes.

Let me repeat that.
Expect breaking changes.

docs.spring.io
spring

# Spring Native documentation

Version 0.9.2 - Andy Clement · Sébastien Deleuze · Filip Hanik · Dave Syer · Esteban Ginez · Jay Bryant · Brian Clozel · Stéphane Nicoll · Josh Long

## 1. Overview

Spring Native provides support for compiling Spring applications to native executables using the GraalVM native-image compiler.

Compared to the Java Virtual Machine, native images can enable cheaper and more sustainable hosting for many types of workloads. These include microservices, function workloads, well suited to containers, and Kubernetes

Using native image provides key advantages, such as instant startup, instant peak performance, and reduced memory consumption.

There are also some drawbacks and trade-offs that the GraalVM native project expect to improve on over time. Building a native image is a heavy process that is slower than a regular application. A native image has fewer runtime optimizations after warmup. Finally, it is less mature than the JVM with some different behaviors.

The key differences between a regular JVM and this native image platform are:

- A static analysis of your application from the main entry point is performed at build time.
- The unused parts are removed at build time.
- Configuration is required for reflection, resources, and dynamic proxies.
- Classpath is fixed at build time.
- No class lazy loading: everything shipped in the executables will be loaded in memory on startup.
- Some code will run at build time.
- There are some limitations around some aspects of Java applications that are not fully supported.

The goal of this project is to incubate the support for Spring Native, an alternative to Spring JVM, and provide a native

youtube.com

Search

code|cademy

Join Codecademy
Ad www.codecademy.com
LEARN MORE

All | Spring Framework | Related | From Spr

0:01 / 17:41 • Introduction (Sébastien Deleuze)

Announcing Spring Native Beta!

46,740 views • Mar 11, 2021

1K 22 SHARE SAVE

SpringDeveloper
155K subscribers

SUBSCRIBED

Sébastien Deleuze, Andy Clement, Brian Clozel, Jens Schauder, Ria Stein & David Syer announce Spring Native Beta v0.9.0. For more details check out the blog post at https://spring.io/blog/2021/03/11/ann...

| | |
|---|---|
| Whats New in Spring Boot 2 4 | SpringDeveloper • 89K views • 3 months ago • 49:43 |
| Cloud Native Java - Josh Long | SpringDeveloper • 45K views • 4 years ago • 1:49:20 |
| Explain it to Me Like I'm 5: Oauth2 and OpenID | SpringDeveloper • 12K views • 7 months ago • 47:50 |
| Spring Tips: Spring Cloud Gateway (Redux) | SpringDeveloper • 40K views • 2 months ago • 1:39:09 |
| Do's and Don'ts: Avoiding First-Time Reactive Programmer... | SpringDeveloper • 20K views • 1 year ago • 58:38 |
| The Path Towards Spring Boot Native Applications | SpringDeveloper • 5.4K views • 7 months ago • 48:54 |
| Best Practices to Spring to Kubernetes Easier and Faster | SpringDeveloper • 14K views • 1 year ago • 1:07:42 |
| It's a Kind of Magic: Under the | |

Sage 50cloud
Ad sage.com/en-us/products/sage-...

The Path Towards Spring Boot Native Applications

5,423 views • Sep 25, 2020

107 1 SHARE SAVE

SpringDeveloper
155K subscribers

SUBSCRIBED

"In this talk, Andy and Sébastien are going to share the latest status of their work on Spring GraalVM Native, the incubating project that allows Spring Boot applications to run as native images.

Cloud Native Java - Josh Long
SpringDeveloper
45K views • 4 years ago
1:49:20

Mix - SpringDeveloper
YouTube

Whats New in Spring Boot 2 4
SpringDeveloper
89K views • 3 months ago
49:43

Spring Tips: The Spring Graal Native Feature
SpringDeveloper
8.7K views • 1 year ago
41:26

Spring Security Patterns
SpringDeveloper
8K views • 7 months ago
54:26

Whats New in Spring Boot 2.3
SpringDeveloper
54K views • 10 months ago
34:51

The Beginner's Guide To Spring Cloud - Ryan Baxter
SpringDeveloper
99K views • 3 years ago
1:08:42

Walking Through Spring Cloud

It isn’t easy to architect cloud native applications.

Lot of moving parts.

Distributed architectures require a fair amount of plumbing.

Spring can help!

Hopefully something here caught your eye.

Remember…

Make the right choice
the easy choice.

How do you keep up with everything happening in Spring?

## What is your learning style?

## Like videos?

youtube.com
YouTube
Search
Home
Explore
Subscriptions
Library
Website
SUBSCRIBED
HOME
VIDEOS
PLAYLISTS
COMMUNITY
CHANNELS
ABOUT
PLAY ALL

SpringDeveloper
155K subscribers

Juergen Hoeller and Ben Hale at SpringOne Platform 2019
SpringDeveloper • 2K views • 1 year ago
SpringOne Platform 2019 - Day 1 Keynote Speakers: Juergen Hoeller, Pivotal; Ben Hale, Pivotal Watch the presentations from SpringOne Platform 2019, filmed in Austin, TX. Slides: Coming Soon

SpringOne Platform 2019
Watch the presentations from SpringOne Platform 2019, filmed in Austin, TX. Learn more about how Pivotal can help you transform your business, not just your IT. https://pv.tl/2IZJYn2

| Title | Channel | Views | Age | Length |
|---|---|---|---|---|
| SpringOne Platform 2019 | VMware Tanzu | 4.5K views | 1 year ago | 3:06 |
| Event Driven with Spring | SpringDeveloper | 33K views | 1 year ago | 1:07:17 |
| Scalable, Cloud-Native Data Applications by Example | SpringDeveloper | 5.5K views | 1 year ago | 1:11:40 |
| Domain-Driven Design with Relational Databases Using… | SpringDeveloper | 32K views | 1 year ago | 1:10:30 |
| Simple Data Movement Patterns: Legacy Applicatio… | VMware Tanzu | 798 views | 1 year ago | 22:44 |
| Reactive Event Processing with Apache Geode | SpringDeveloper | 3.4K views | 1 year ago | 30:44 |

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Tanzu.TV Community

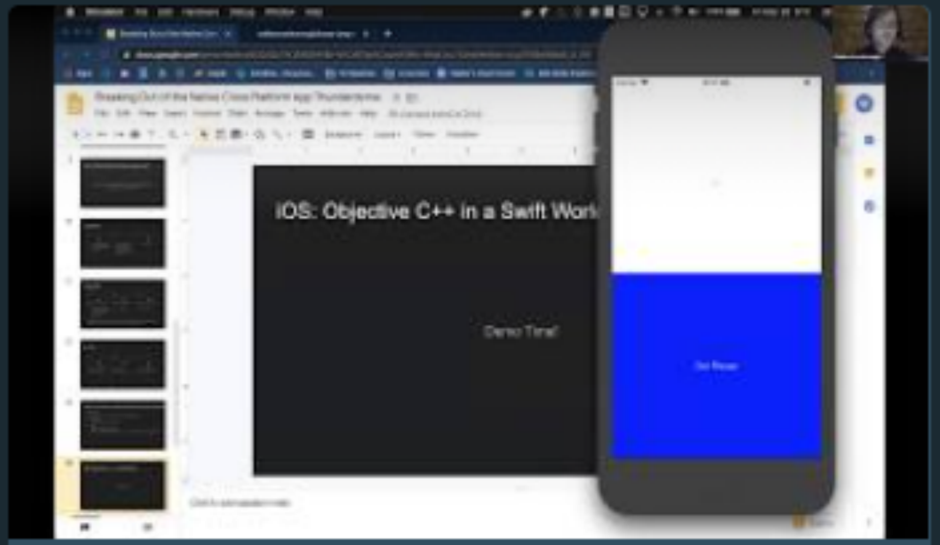
Tanzu.TV / Spring Live

# Spring Live

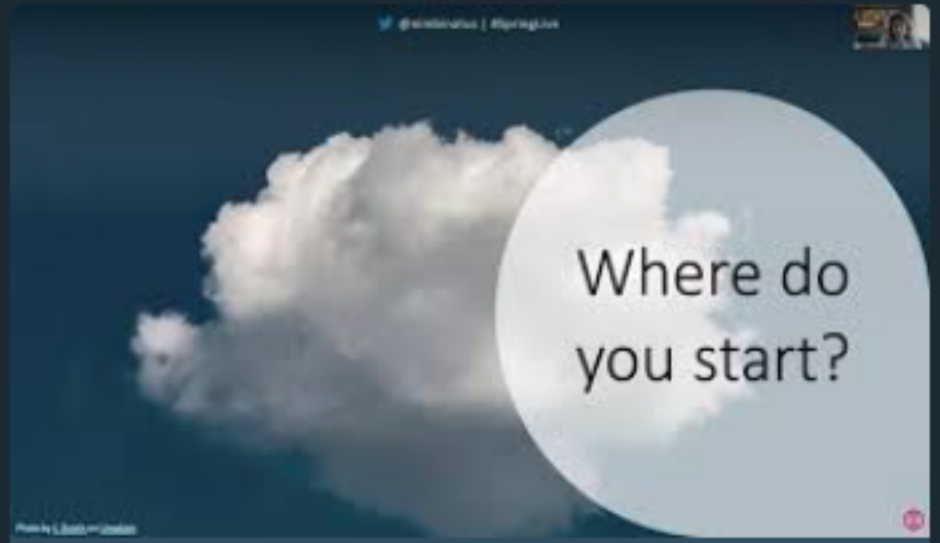
24 HOURS OF CLOUD NATIVE CONTENT.

Production-Ready Spring Boot Applications

How to implement and manager an award winning platform to enable developers



Breaking Out Of The Cross Platform Mobile App Tooling Thunderdome



Shifting to the Where? Exploring the paths to a cloudless world

Building community for your company's open source projects

Security Instrumentation Is the Future of All Software

Prefer to comment along to live streams? We’ve got you!

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Tanzu.TV Community

## Tanzu.TV Shows

Explore Tanzu.TV on the VMware Tanzu Developer Center for live demos of modern application development technologies, webinars, and episodes covering coding, Kubernetes, and more!

### Tanzu Tuesdays

LIVE EVERY TUESDAY AT 12PM PT

LIVE TODAY
April 27th 2021
Argo CD with Knative and Cloud Native Buildpacks with Boskey Savla

LIVE IN 7 DAYS
May 4 2021
GitOps for APIs: Enterprise Microservices in the Age of Kubernetes with Chris Sterling

April 20th 2021
Tracing Issues in Your Application with Marcin Grzejszczak

### Code

LIVE EVERY WEDNESDAY AT 12PM PT

How design can kill your testability with Jakub Pilimon

Spring Integration + Kotlin : Holy cow this is productivity with Mario Gray

Don't Cross the Streams! Using Spring Cloud Stream to bust the

Pick a day.

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Tanzu.TV Community

Tanzu.TV / Between Chair and Keyboard

# Between Chair and Keyboard

JOIN NATE SCHUTTA AND A SPECIAL GUEST ON TWITCH, MOST MONDAYS AT 10AM PT.

WATCH ON TWITCH

Interviews and discussions with open source committers, thought leaders, friends and interesting people in the software space. No preparation, no live coding, no slides just a conversation with “cool”, smart people. An “Inside the Actors Studio” but for technologists if we may be so bold.

## Upcoming Episodes

Streaming May 3 on Twitch

The one with Luca Mezzalira

Streaming May 10 on Twitch

The one with Adib Saikali

tanzu.vmware.com
vmware Tanzu Developer Center
Learn
Topics
Tanzu.TV
Community
WATCH ON TWITCH
LIVE TODAY
April 27th 2021
LIVE IN 7 DAYS
May 4 2021

Tanzu.TV / Tanzu Tuesdays

# Tanzu Tuesdays

LIVE DEMOS ON TWITCH, EVERY TUESDAY AT 12PM PT.

## Upcoming Episodes

Streaming April 27 on Twitch

Argo CD with Knative and Cloud Native Buildpacks with Boskey Savla

Streaming May 4 on Twitch

GitOps for APIs: Enterprise Microservices in the Age of Kubernetes with Chris Sterling

## Watch Past Episodes

# Code

LIVE CODING ON TWITCH, EVERY WEDNESDAY AT 12PM PT.

Each week our fearless team of Spring Developer Advocates will be tackling real world scenarios and complex technical topics and will be live coding solutions to them. This is your chance to learn tips, tools, and techniques from world class Spring developers.

## Watch Past Episodes

How design can kill your testability with Jakub Pilimon

Spring Integration + Kotlin : Holy cow this is productivity with Mario Gray

Don't Cross the Streams! Using Spring Cloud Stream to bust the ghosts of interservice communication ...

tanzu.vmware.com
vmware Tanzu Developer Center
Learn
Topics
Tanzu.TV
Community

Tanzu.TV / TGIK

TGIK

LIVE DISCUSSION ABOUT KUBERNETES ON YOUTUBE, EVERY FRIDAY AT 1PM PT.

WATCH ON YOUTUBE

Watch Past Episodes

vmware kubernetes TGI kubernetes Ep 152 Fridays at 1pm pacific

TGI Kubernetes 152: Cluster Testing with Sonobuoy

vmware kubernetes TGI kubernetes Ep 151 Fridays at 1pm pacific

# TGIK 151 - Cluster Diagnotics with Crashd

vmware kubernetes TGI kubernetes Ep 150 Fridays at 1pm pacific

TGI Kubernetes 150: Production Kubernetes Book Discussion

vmware kubernetes TGI kubernetes Ep 149 Fridays at 1pm pacific

TGI Kubernetes 149: GitOps with

vmware kubernetes TGI kubernetes Ep 148 Fridays at 1pm pacific

TGI Kubernetes 148: Gateway APIs

vmware kubernetes TGI kubernetes Ep 147 Fridays at 1pm pacific

TGI Kubernetes 147: CoreDNS

tanzu.vmware.com
vmware Tanzu Developer Center
Learn
Topics
Tanzu.TV
Community
Tanzu.TV / Tanzu Talk
WATCH ON TWITCH
Tanzu Talk
MOVING FROM PROJECT TO PRODUCT AT YAPI KREDI
BEYOND FAX MACHINES HEALTHCARE
THE NEW APP PLATFORM
with Ajay

# Tanzu Talk

DAILY INTERVIEWS AND DISCUSSIONS ABOUT CLOUD NATIVE SOFTWARE ON TWITCH, EVERY WEEKDAY AT 11AM CET.

## Watch Past Episodes

Tanzu.TV / SpringOne Tour

# SpringOne Tour

A series of two-day, live online events where your favorites from the community go in depth on different topics at the intersection of Spring and Kubernetes. Day 1 features a mix of presentations and interactive demos on Twitch. Day 2 includes hands-on workshops in groups, along with 1:1 interaction with an instructor.

Workshop attendees will receive special edition swag

## Upcoming Episodes

### Booternetes Europe

Workshop

### Booternetes II: The YAML Strikes Back

Livestream and Workshop

How about some
live online training?

oreilly.com

O'REILLY ENTERPRISE PRICING SIGN IN TRY NOW

Search for books, videos, live events, and more

VIEW ALL EVENTS

SPRING

# Reactive Spring and Spring Boot

## Using the Spring WebFlux module to build high-performance reactive systems

What you'll learn Is this course for you? Schedule

The latest major releases of the Spring Framework, Spring 5 and Spring Boot 2, take full advantage of the new functional features provided by Java SE 8 as well as the Reactive Streams specification. The new WebFlux module adds Spring's ease of use and Spring Boot's ease of configuration to the design and development of reactive applications, while Reactive Streams allow applications to more efficiently make use of limited resources by adding nonblocking capabilities.

Join expert Ken Kousen to gain the foundation you need to take advantage of reactive programming, the Reactive Streams specification (a widely adopted industry standard), and the WebFlux module.

APAC friendly time.

## What you'll learn and how you can apply it

**By the end of this live online course, you'll understand:**

- Spring's convention over configuration and dependency injection
- Spring Boot autoconfiguration and rapid prototyping capabilities
- The Reactive Streams specification and how it contributes to building asynchronous, reactive applications

May 4, 2021
4:00 a.m. - 8:00 a.m. CDT

67 spots available

Sign up for a free trial!

or sign in.

Registration closes May 3, 2021 5:00 p.m. CDT

YOUR INSTRUCTOR

Ken Kousen

Ken Kousen is the author of the Kotlin Cookbook (O'Reilly), Modern Java Recipes (O'Reilly), Gradle Recipes for Android (O'Reilly), and Making Java Groovy (Manning), as well as O'Reilly video courses in Android, Groovy,

Read more

Start your free 10-day trial

Learn more

## On Twitter? So are we!

@springcentral @habuma @ryanjbaxter
@springjuergen @david_syer
@kenkousen @csterwa
@michaelminella
@olga_maciaszek @cppwfs @springbootbook
@violeta_g_g
@starbuxman @Audrey_Neveu
@mariogray
@maciejwalkowiak
@spencerbgibb
@royclarkson
@ilayaperumalg @ReactiveSpring
@springcloud @ciberkleid
@bellalleb_bai @tiffanyfayj @ntschutta
@madhurabhave23
@rob_winch @MGrzejszczak @dturanski
@smaldini
@phillip_webb
@SpringTipsLive
@SpringOne @BootifulPodcast @cote
@SpringSecurity @JakubPilimon
@benbravo73
@scottyfred @fifthposition @SpringData @snicoll
@mkheck

I’m sure I missed several…

And I am very sorry for that!

Follow away!

Prefer to read?

manning.com

MANNING PUBLICATIONS
Quality is many small things done right

search

all products MEAP liveBook liveVideo liveProject liveAudio free content ● live sign in

Save half on *GraphQL in Action* and more TODAY ONLY!

Spring in Action, Sixth Edition

3 reviews

775 views in the last week

Craig Walls

MEAP began May 2020 · Publication in Summer 2021 *(estimated)*

ISBN 9781617297571 · 520 pages *(estimated)* · printed in black & white

print book ~~$59.99~~ $29.99
*pBook + eBook + liveBook*
includes *previous edition eBook*
Additional shipping charges may apply

add to cart

eBook ~~$47.99~~ $23.99
*3 formats + liveBook*
includes *previous edition eBook*

add to cart

free previous edition eBook included

An eBook copy of the previous edition of this book is included at no additional cost. It will be automatically added to your Manning Bookshelf within 24 hours of purchase.

enable 2-click buy

Look Inside

resources

Source code

Book Forum

*show all*

To me, this has always been the defacto standard for documentation on the Spring Framework. I bought the 1st edition when it first came out as we were converting alegacy app to Spring and this book was essential in learning how the current version worked.

Tony Sweets

customers also bought

MEAP

**Manning Early Access Program (MEAP)**
Read chapters as they are written, get the finished eBook as soon as it's ready, and receive the pBook long before it's in bookstores.

8 of 19 chapters available

A new edition of the classic bestseller! *Spring in Action, 6th Edition* covers all of the new features of Spring 5.3 and Spring Boot 2.4 along with examples of reactive programming, Spring Security for REST Services, and bringing reactivity to your databases. You'll also find the latest Spring best practices, including Spring Boot for application setup and configuration.

leanpub.com

Books Bundles Courses Search Leanpub

Reactive Spring

$30.00 MINIMUM PRICE

$30.00 SUGGESTED PRICE

YOU PAY $30.00

AUTHOR EARNS $24.00

YOU PAY (US$) $30.00

**EU customers:** Price excludes VAT. VAT is added during checkout.

Add Ebook to Cart

Add to Wish List

This book is 100% complete

COMPLETED ON 2020-09-19

Josh Long

Join Spring Developer Advocate Josh Long (@starbuxman) for an introduction to reactive programming and its implementation in the Spring ecosystem.

1,225 READERS

378 PAGES

ENGLISH

PDF

EPUB

MOBI

About the Book

oreilly.com

O'REILLY TEAMS INDIVIDUALS FEATURES CONTENT SPONSORSHIP SIGN IN TRY NOW

See everything available through O'Reilly online learning and start a free trial. Explore now. Search

## Spring Boot: Up and Running

by Mark Heckler
Released February 2021
Publisher(s): O'Reilly Media, Inc.
ISBN: 9781492076988

Explore a preview version of *Spring Boot: Up and Running* right now.

O'Reilly members get unlimited access to live online training experiences, plus books, videos, and digital content from 200+ publishers.

BUY ON AMAZON BUY FROM O'REILLY START YOUR FREE TRIAL

## Book description

With over 75 million downloads per month, Spring Boot is the most widely used Java framework available. Its ease and power have revolutionized application development from monoliths to microservices. Yet Spring Boot's simplicity can also be confounding. How do developers learn enough to be productive immediately? This practical book shows you how to use this framework to write successful mission-critical applications.

Mark Heckler from VMware, the company behind Spring, guides you through Spring Boot's architecture and approach, covering topics such as debugging, testing, and deployment. If you want to develop cloud native Java or Kotlin applications with Spring

Not sure where to start?

spring.io

spring

Why Spring Learn Projects Training Support Community

Spring makes Java cloud-ready.

WHY SPRING QUICKSTART

NEWS Announcing Spring Native Beta! | SpringOne returns Sep 1-2, 2021

What Spring can do

**Microservices**

Quickly deliver production-grade features with independently evolvable microservices.

**Reactive**

Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.

**Cloud**

Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.

**Web apps**

Frameworks for fast, secure, and responsive web applications connected to any data store.

spring.io
spring
Why Spring
Learn
Projects
Training
Support
Community

# Projects

From configuration to security, web apps to big data—whatever the infrastructure needs of your application may be, there is a Spring Project to help you build it. Start small and use just what you need—Spring is modular by design.

**Spring Boot**

Takes an opinionated view of building Spring applications and gets you up and running as quickly as possible.

**Spring Framework**

Provides core support for dependency injection, transaction management, web apps, data access, messaging, and more.

**Spring Data**

Provides a consistent approach to data access – relational, non-relational, map-reduce, and beyond.

**Spring Cloud**

Provides a set of tools for common patterns in distributed systems. Useful for building and deploying microservices.

Deep dive documentation?

spring.io

Why Spring | Learn | Projects | Training | Support | Community

- Spring Boot
- Spring Framework
- Spring Data
- Spring Cloud
- Spring Cloud Data Flow
- Spring Security
- Spring Session
- Spring Integration
- Spring HATEOAS
- Spring REST Docs
- Spring Batch
- Spring AMQP
- Spring for Android
- Spring CredHub
- Spring Flo
- Spring for Apache Kafka
- Spring LDAP
- Spring Mobile
- Spring Roo
- Spring Shell
- Spring Statemachine
- Spring Vault
- Spring Web Flow
- Spring Web Services

# Spring Boot 2.4.5

OVERVIEW | LEARN | SAMPLES

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

If you're looking for information about a specific version, or instructions about how to upgrade from an earlier release, check out the project release notes section on our wiki.

## Features

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

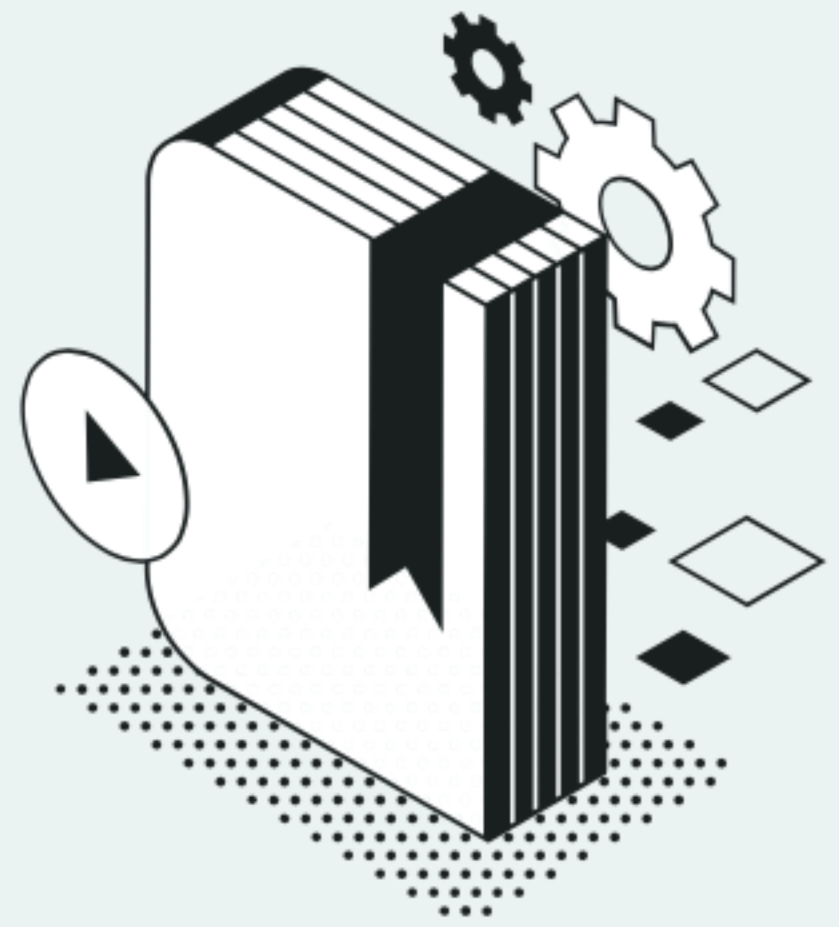
## Getting Started

- Super quick — try the Quickstart Guide.
- More general — try Building an Application with Spring Boot
- More specific — try Building a RESTful Web Service.
- Or search through all our guides on the Guides homepage.

Want some hands on coding?

spring.io

Why Spring Learn Projects Training Support Community



Guides

Whatever you're building, these guides are designed to get you productive as quickly as possible – using the latest Spring project releases and techniques as recommended by the Spring team.

**Getting Started Guides**
15-30 minutes

**Topical Guides**
60 minutes or less

**Tutorials**
2-3 hours

Getting Started Guides

Designed to be completed in 15-30 minutes, these guides provide quick, hands-on instructions for building the "Hello World" of any development task with Spring. In most cases, the only prerequisites are a JDK and a text editor.

Find a guide

Building a RESTful Web Service

Scheduling Tasks

Prefer to listen?

soundcloud.com
SOUNDCLOUD
Home
Stream
Library
Search for artists, bands, tracks, podcasts
Sign in
Create account
Upload
A Bootiful Podcast with Josh Long (@starbuxman)
spring

A Bootiful Podcast

Josh Long

San Francisco, United States

PRO UNLIMITED

All | Popular tracks | Tracks | Albums | Playlists | Reposts

Station | Follow | Share

Recent

A Bootiful Podcast — 1 year ago
"The Phoenix Project" author Gene Kim
# Technology
1:05:18
5 | Repost | Share | More | 2,907

A Bootiful Podcast — 1 year ago
Neo4j's mad scientist Michael Hunger on graphs, databases and relation-ships
# Technology
1:05:58
6 | 1 | Share | More | 2,195 | 1

A Bootiful Podcast — 1 year ago
Spring Cloud Services teammate Bella（Yuxin）Bai
# Technology
45:51
5 | Repost | Share | More | 2,361

| Followers | Following | Tracks |
|---|---|---|
| 1,005 | 1 | 57 |

I'm Josh Long (twitter.com/Starbuxman), a humble Spring advocate at @Pivotal and this is a (twitter.com/BootifulPodcast) a Bootiful Podcast, a celebration of the real heroes that drive ecosystems.

- @starbuxman on Twitter
- the Spring Initializr
- my blawg
- @Bootiful_Podcast on Twitter
- iTunes
- Google Play
- Patreon
- Spotify

Fans also like — Refresh

| | | | |
|---|---|---|---|
| O'Reilly Programming | 2,062 | 19 | Follow |
| Dockercast | 1,186 | 6 | Follow |
| ThoughtWorks | 3,073 | 121 | Follow |

1 following — View all

Rich ecosystem with a large, thriving community.

## Resources abound!

Ask questions!

Good luck!

# Resources

- The Power of Suggestion: Inertia in 401(k) Participation and Savings Behavior
  https://www.jstor.org/stable/2696456?seq=1
- What Is Spring?
  https://springone.io/2020/sessions/what-is-spring
- Why Spring?
  https://spring.io/why-spring
- spring initializr
  https://start.spring.io
- The Beginner's Guide To Spring Cloud
  https://www.youtube.com/watch?v=aO3W-lYnw-o
- Spring in Action
  https://www.manning.com/books/spring-in-action-sixth-edition

# Resources

- Spring and Spring Boot Fundamentals
  https://www.oreilly.com/learning-paths/learning-path-spring/9781492055334/
- Tanzu Developer Portal
  https://tanzu.vmware.com/developer/
- Spring Cloud Circuit Breaker
  https://spring.io/projects/spring-cloud-circuitbreaker
- Spring Cloud Circuit Breaker Guide
  https://spring.io/guides/gs/circuit-breaker/
- Spring Tips: Spring Cloud Circuit Breaker
  https://www.youtube.com/watch?v=s5-leUCti5o
- Spring Cloud Sleuth
  https://spring.io/projects/spring-cloud-sleuth

## Resources

- ✦ Spring Boot Actuator
  https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-features.html
- ✦ Application Monitoring With Spring Boot Actuator
  https://dzone.com/articles/application-monitoring-with-spring-boot
- ✦ Spring Tips: The Wavefront Observability Platform
  https://spring.io/blog/2020/04/29/spring-tips-the-wavefront-observability-platform
- ✦ Spring Boot Observability
  https://www.youtube.com/watch?v=zGiBpUlg9mk
- ✦ Mastering Spring Boot's Actuator
  https://www.youtube.com/watch?v=otcYECeFS6Y

## Resources

- Spring REST Docs
  https://spring.io/projects/spring-restdocs
- If Hemingway Wrote JavaDocs
  https://springone.io/post-event/sessions/if-hemingway-wrote-javadocs
- Documenting RESTful APIs with Spring REST Docs
  https://www.youtube.com/watch?v=CaARz49u1Mc
- Spring Cloud Contract
  https://spring.io/projects/spring-cloud-contract
- Spring Tips: Spring Cloud Contract
  https://spring.io/blog/2017/10/25/spring-tips-spring-cloud-contract-http
- Consumer Driven Contract Testing with Spring Cloud Contract
  https://www.youtube.com/watch?v=QHlhYQQa7bg

## Resources

- ✦ Spring Cloud Gateway
  https://spring.io/projects/spring-cloud-gateway#overview
- ✦ Introducing Spring Cloud Gateway and API Hub for VMware Tanzu
  https://springone.io/post-event/sessions/introducing-spring-cloud-gateway-and-api-hub-for-vmware-tanzu
- ✦ Reactive Architectures with RSocket and Spring Cloud Gateway
  https://www.youtube.com/watch?v=PfbycN_eqhg
- ✦ Spring Cloud Gateway for Stateless Microservice Authorization
  https://www.youtube.com/watch?v=RRMO4oNptoQ
- ✦ Top tips for running Spring Boot applications on Kubernetes with Ollie Hughes
  https://www.youtube.com/watch?v=R9mNUfvp8Dg
- ✦ Spring on Kubernetes Workshop
  https://tanzu.vmware.com/developer/workshops/spring-on-kubernetes/

# Resources

- ✦ SpringOne Tour 2021: # Booternetes
  https://www.youtube.com/watch?v=LfbU5xuR7Ck
- ✦ Spring on Kubernetes
  https://spring.io/guides/topicals/spring-on-kubernetes/
- ✦ Getting Started with Spring Cloud Kubernetes
  https://www.youtube.com/watch?v=u64jexEX_RY
- ✦ Spring Native documentation
  https://docs.spring.io/spring-native/docs/current/reference/htmlsingle/
- ✦ Announcing Spring Native Beta!
  https://www.youtube.com/watch?v=96n_YpGx-JU
- ✦ The Path Towards Spring Boot Native Applications
  https://www.youtube.com/watch?v=Um9djPTtPe0

# Resources

- How Fast is Spring?
  https://www.youtube.com/watch?v=T22i3WAa6dI
- SpringDeveloper on YouTube
  https://www.youtube.com/channel/UC7yfnfvEUlXUIfm8rGLwZdA
- SpringOne Tour
  https://tanzu.vmware.com/developer/tv/springone-tour/
- Tanzu.TV Shows
  https://tanzu.vmware.com/developer/tv/
- Spring.io
  https://spring.io
- Spring Documentation
  https://spring.io/projects

## Resources

- ✦ Spring Tips: Spring Cloud Loadbalancer
  https://spring.io/blog/2020/03/25/spring-tips-spring-cloud-loadbalancer
- ✦ How to Live in a Post-Spring-Cloud-Netflix World
  https://www.youtube.com/watch?v=mINNQ3zpRrE
- ✦ Service Registration and Discover
  https://spring.io/guides/gs/service-registration-and-discovery/
- ✦ Spring Tips: Spring Cloud Stream
  https://www.youtube.com/watch?v=HQ00E60kB6c
- ✦ Spring Tips: Spring Cloud Stream Kafka Streams
  https://www.youtube.com/watch?v=YPDzcmqwCNo
- ✦ Streaming Processing and Testing with Spring Cloud Stream
  https://www.youtube.com/watch?v=7QNkYqPcVpI

# Resources

- ✦ Spring Cloud Stream - demystified and simplified
  https://spring.io/blog/2019/10/14/spring-cloud-stream-demystified-and-simplified
- ✦ Spring Cloud Stream - Event Routing
  https://spring.io/blog/2019/10/31/spring-cloud-stream-event-routing
- ✦ Reactive
  https://spring.io/reactive
- ✦ Spring Tips and Reactive Spring
  https://www.youtube.com/watch?v=_LR0Cxnn-kw
- ✦ Reactive Spring by Josh Long
  https://www.youtube.com/watch?v=zVNIZXf4BG8

# THANK YOU!

Nathaniel Schutta
@ntschutta
ntschutta.io

Jakub Pilimon
@JakubPilimon

## Between Chair and Keyboard

Most Mondays,
around noon Central
https://www.twitch.tv/vmwaretanzu

Nate Schutta
Software Architect
VMware
@ntschutta

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Tanzu.TV Community

# Tanzu.TV Shows

Explore Tanzu.TV on the VMware Tanzu Developer Center for live demos of modern application development technologies, webinars, and episodes covering coding, Kubernetes, and more!

Follow us on Twitch and subscribe to our YouTube channels VMware Tanzu & VMware Cloud Native Apps to stay up to date with the latest Tanzu.TV content.

## Tanzu Tuesdays

LIVE EVERY TUESDAY AT 12PM PT

LIVE TODAY

Let's talk about SpringOne 2021!

AUG 31 2021

Clear-up Coté's Kubernetes Questions with Michael Coté

AUG 24 2021

Staying Ahead of the Curve with Spring and Cassandra 4.0 with David Gi...

AUG 10 2021

## Code

LIVE EVERY WEDNESDAY AT 12PM PT

tanzu.vmware.com

vmware Tanzu Developer Center

Learn Topics Tanzu.TV Community

## Upcoming Episodes

NOVEMBER 17

**Europe: Monitoring, observability, and replatforming**

Details coming soon

DECEMBER 8–9

**Platforms built on K8s**

Details coming soon

## Watch Past Episodes

Do or Do Not, There is No Try{} in Production

Booternetes II: The YAML Strikes Back (Europe)

Booternetes II: The YAML Strikes Back