"Cloudier"

- Monitoring your apps in the cloud
  - Spring Boot provided endpoints
  - Writing custom health checks
- Deploy your Spring Boot apps to the cloud using Docker

Monitoring with Spring Boot

Introducing Spring Boot Actuator

- Production ready **monitoring** and **management features** out of the box
  - Health, autoconfig report, beans, etc
- HTTP or JMX
  - Feed into Nagios / Zabbix / New Relic
- Easy to add your own

```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

pom.xml

Adding Spring Boot Actuator to Your Project

Builtin Production Ready Endpoints

`/autoconfig` **for report**

`/beans` **for all beans**

`/configprops` **for all config**

`/dump` **for memory dump**

`/health` **to check application**

Many more ...

http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#production-ready

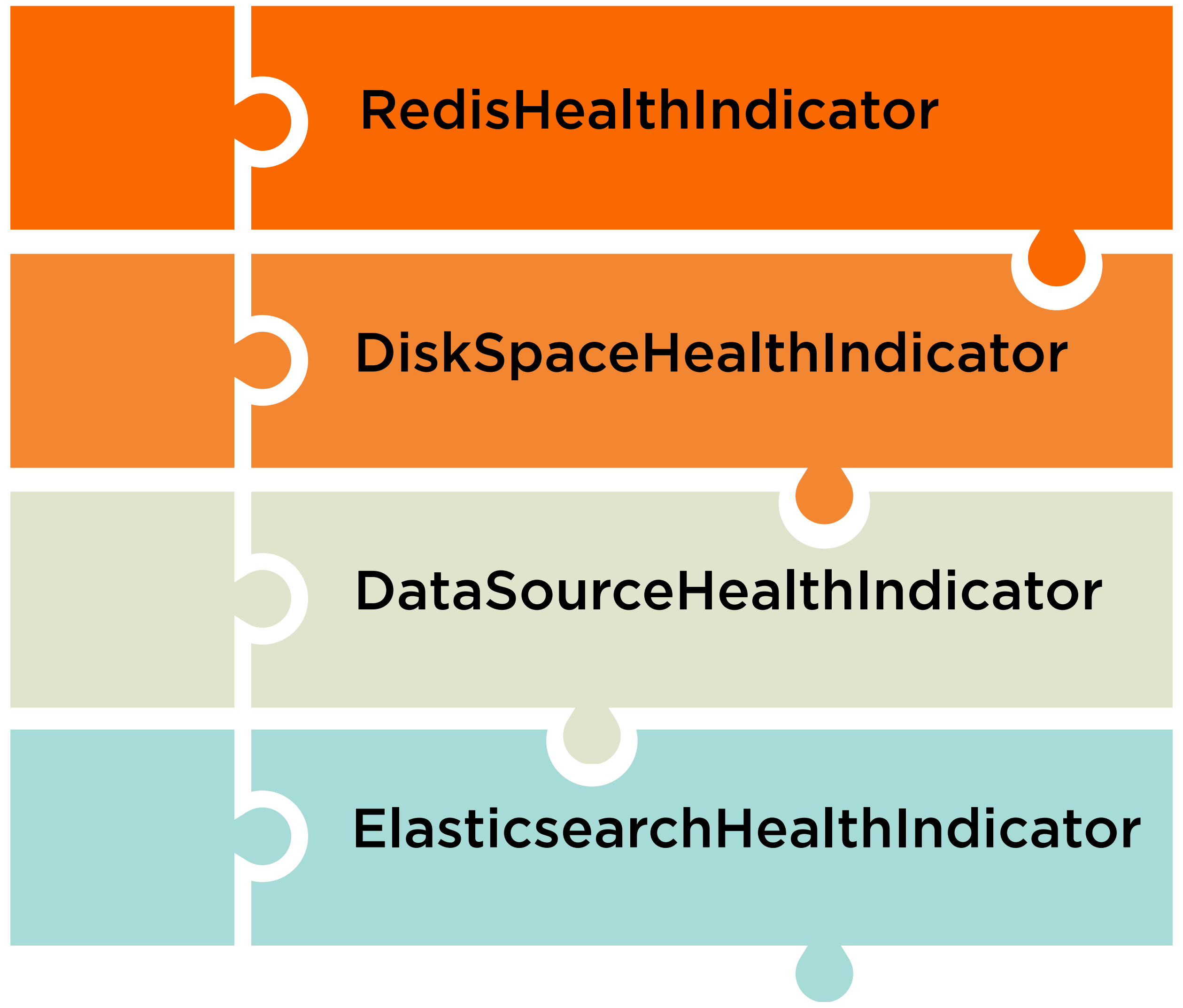
Is your application healthy?

Spring Boot HealthIndicator's

Autoconfigured

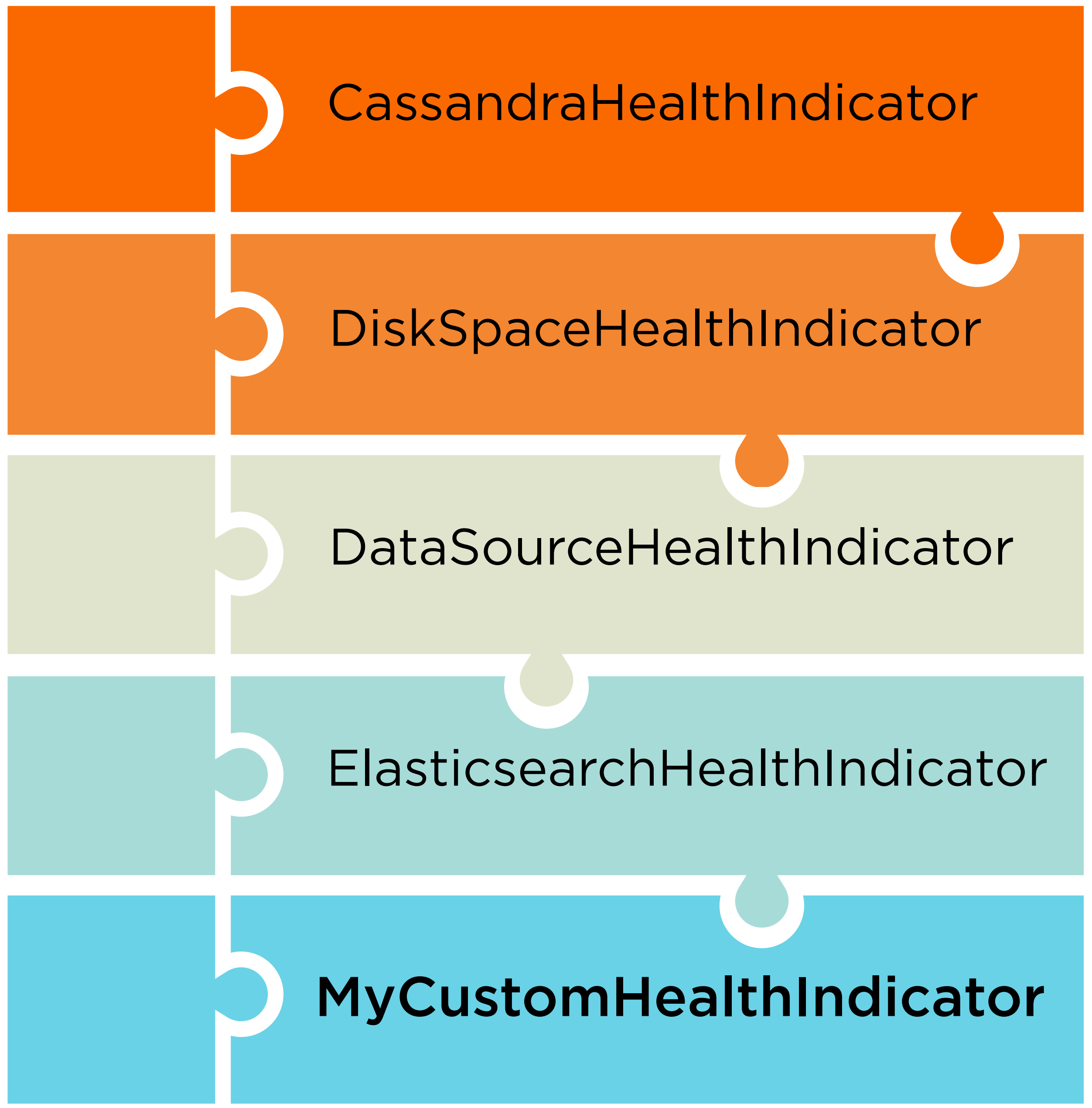
Custom

Autoconfigured HealthIndicator’s



http://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#_auto_configured_healthindicators

Custom HealthIndicator's



```java
@Component
public class MyCustomHealthIndicator implements HealthIndicator {






}
```

Defining Your Own `HealthIndicator`’s

Define a class that’s annotated with `@Component` **and implements** `HealthIndicator`

```java
@Component
public class MyCustomHealthIndicator implements HealthIndicator {

    @Override
    public Health health() {
        ...
    }
}
```

Defining Your Own `HealthIndicator`’s

Implement the single `health()` method

```
// Condition failed
return Health.down().build();

// Condition failed with details (authentication required)
return Health.down().withDetail("someKey", "someValue").build();

// Condition is ok
return Health.up().build();

// Condition is unknown
return Health.unknown().build();
```

Defining Your Own `HealthIndicator`’s

Use the `Health` class’s static builder methods to build `Health` object

Preparing Our Application for the Cloud

But wait, what if I don’t know anything about

docker

?

What Is Docker?

- Virtualization management software for containers and images:
  - Build images
  - Deploy images into containers
  - Manage containers

What’s a container?

“A container is a stripped-to-basics version of a Linux operating system.”

Docker documentation

What’s an image?

The software you run in a container is called an ***image***

Why Docker?

Easy

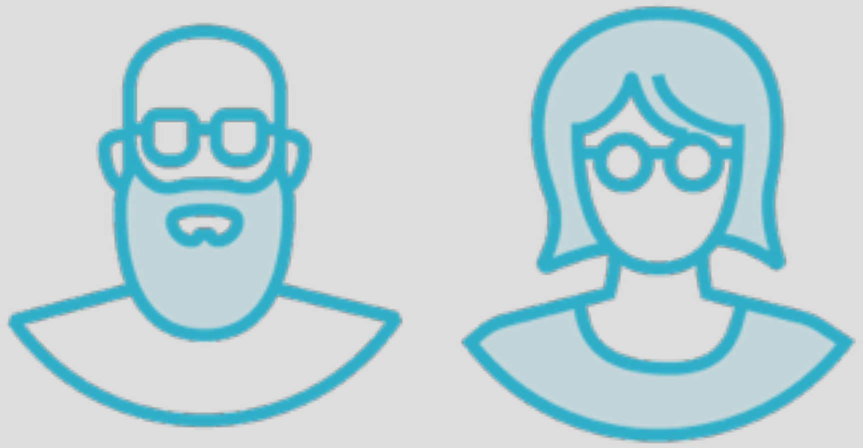
Lightweight

Cloud Agnostic

Scales

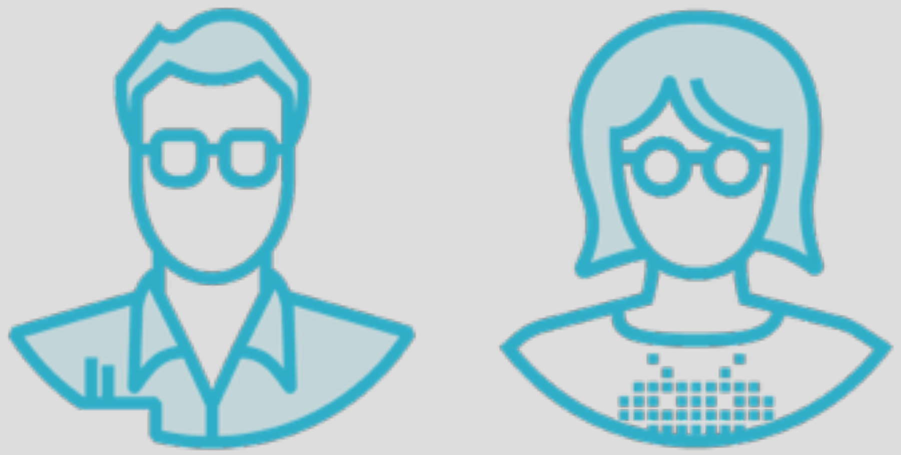
Installing Docker

Linux



https://docs.docker.com/engine/installation/linux/

Mac



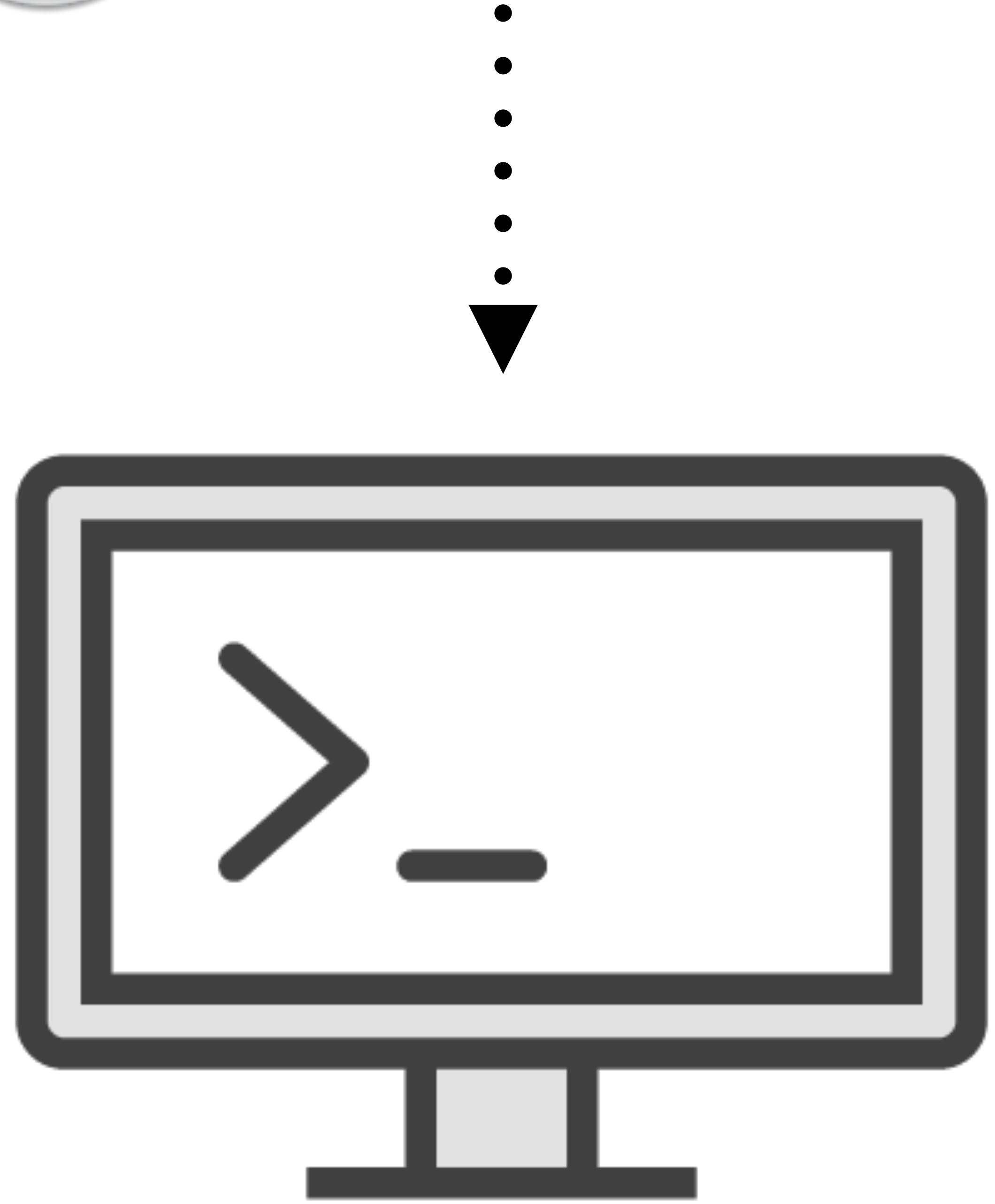
https://docs.docker.com/engine/installation/mac/

Windows

https://docs.docker.com/engine/installation/windows/

Docker Toolbox

Double click **Docker Quickstart** to initialize Docker



Deploying Spring Boot to the Cloud

Amazon EC2 Container Service

- A container management service in the cloud
  - Supports Docker
- Highly scalable
  - Runs as a cluster
  - Can grow and shrink as needed
  - Load balancing (ELB)
- No additional charge

https://aws.amazon.com/

AWS Command Line Tool

https://aws.amazon.com/cli/

In Review...

- Spring Boot Actuator
- Spring Boot + Docker
- Deploying to the cloud (AWS)