



Web deployment made easy

Introduction to OpenShift

Alberto Rodriguez Peon IT-CDA-WF
Alvaro Gonzalez Alvarez IT-CDA-WF

Web deployment @ CERN

— Effort to set up

— Flexibility

+ Effort to set up

+ Flexibility



Web deployment @ CERN

- Effort to set up
- Flexibility

- + Effort to set up
- + Flexibility



Other central
web services



Web deployment @ CERN

- Effort to set up
- Flexibility

- + Effort to set up
- + Flexibility



Other central web services

Deployment in dedicated servers



Web deployment @ CERN

- Effort to set up
- Flexibility



- + Effort to set up
- + Flexibility



Other central web services



Based on Kubernetes and Docker containers, **OpenShift** provides a **balance between configuration effort and flexibility** to deploy any kind of web application

Deployment in dedicated servers





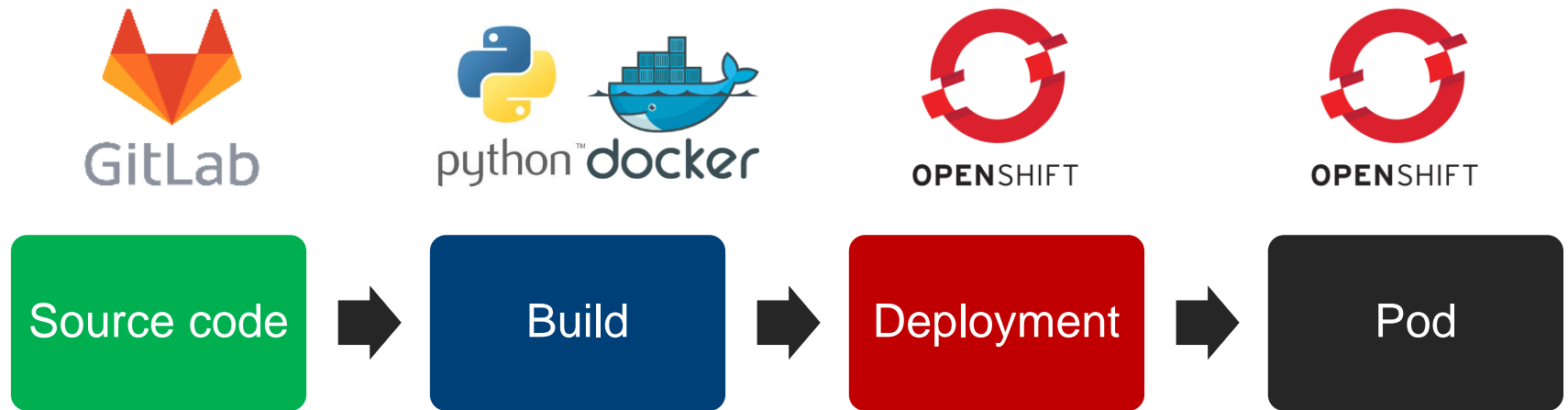
OpenShift is a Kubernetes distribution **customized for web hosting**

- **Kubernetes** provides **automating deployment, scaling and management** of containerized applications

On top of **Kubernetes**, brings:

- An **internal Software Defined Network (SDN)** optimized for multi-tenancy, enabling communication between containers of the same application
- An **internal Docker registry** to store automatically generated Docker images
- An **HAProxy** router to load balance external requests to containers running in the cluster
- A **web dashboard to graphically** manage your application

Source code → Production



OpenShift provides the platform to easily **automate the deployment** of applications:

- ❑ Built-in Docker images for **most common programming languages**
- ❑ **Orchestration** of you application components, including rolling updates
- ❑ **Load balancing** to the container(s) running your application

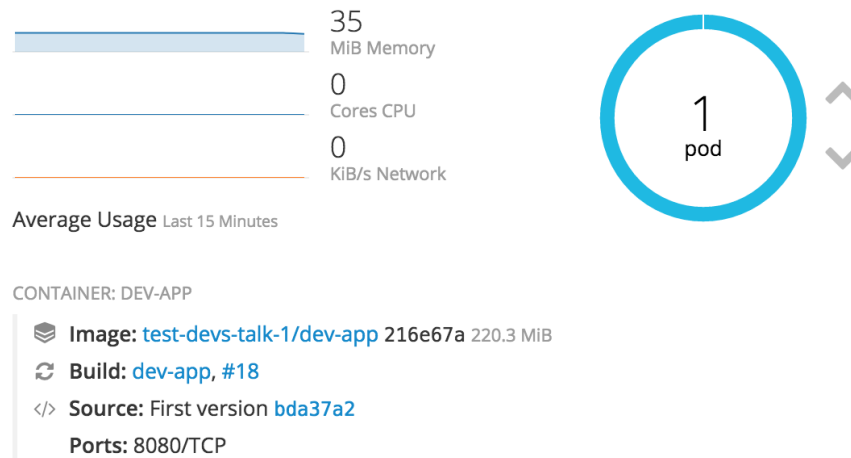
Pods and containers

In OpenShift, applications run as a set of pods

A pod is one or more containers running together, providing a single (micro)service

Pods are **immutable** and **expendable**, usually, not managed directly by users but by other OpenShift components

Pods **can communicate together** through internal IP addresses in the **OpenShift SDN**



Building a Docker image

OpenShift provides **automation** for **building Docker images** from **source code**, including **built-in Docker images** for the **most common programming languages**

These images allow **out-of-the-box build process** without writing a Dockerfile

Maintained by Red Hat

By default, **updates to the base image will rebuild and redeploy** your application, ensuring your application **always get the latest security fixes**

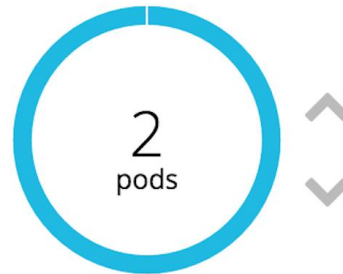


Using a **Dockerfile** for other programming languages and/or **more complex applications** is also possible

Orchestrating the lifecycle of your application

The **lifecycle** of applications is also fully automated by describing **how the application should run**. This includes:

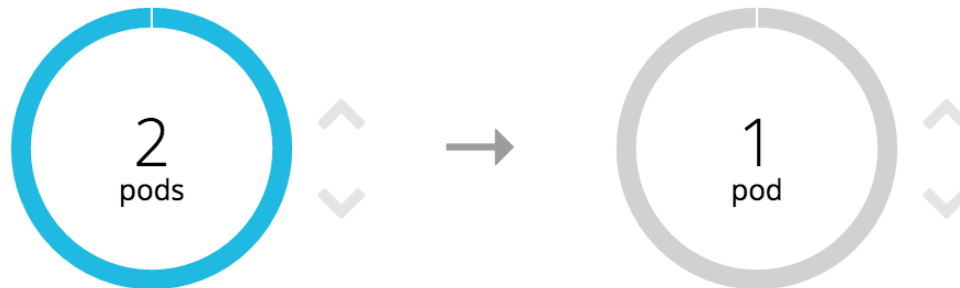
- **Triggers for automatic redeployment** after a change
- **Versioning of the application**, supporting rollbacks to a previous version
- Manual or automatic **scaling of components**
- **Different deployment strategies**, including allowing **graceful shutdown**, running **custom hooks** at a given point of the deployment or **rolling updates**



Orchestrating the lifecycle of your application

The **lifecycle** of applications is also fully automated by describing **how the application should run**. This includes:

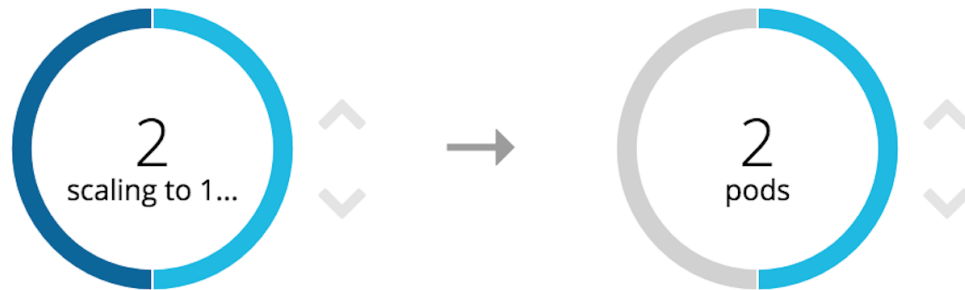
- **Triggers for automatic redeployment** after a change
- **Versioning of the application**, supporting rollbacks to a previous version
- Manual or automatic **scaling of components**
- **Different deployment strategies**, including allowing **graceful shutdown**, running **custom hooks** at a given point of the deployment or **rolling updates**



Orchestrating the lifecycle of your application

The **lifecycle** of applications is also fully automated by describing **how the application should run**. This includes:

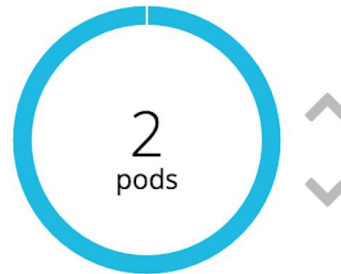
- **Triggers for automatic redeployment** after a change
- **Versioning of the application**, supporting rollbacks to a previous version
- Manual or automatic **scaling of components**
- **Different deployment strategies**, including allowing **graceful shutdown**, running **custom hooks** at a given point of the deployment or **rolling updates**



Orchestrating the lifecycle of your application

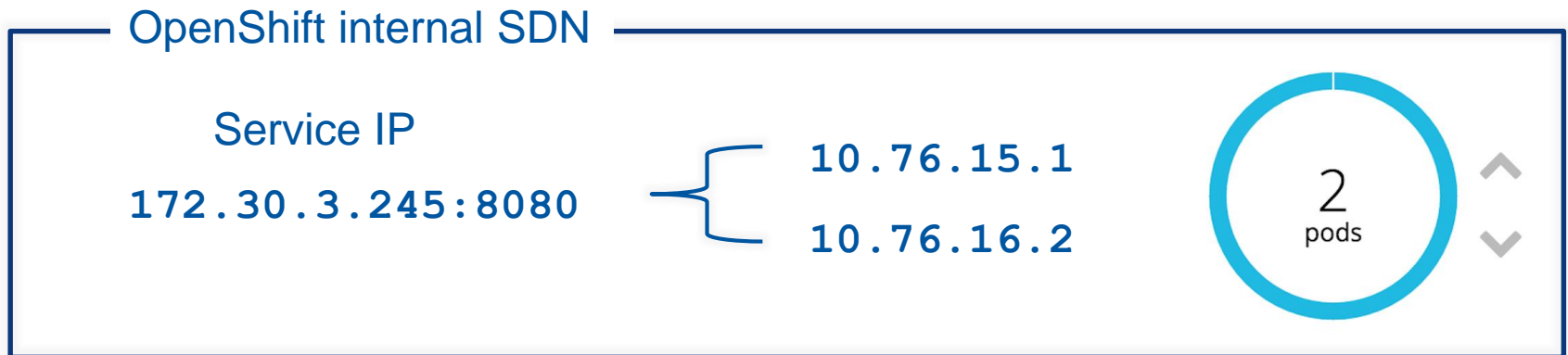
The **lifecycle** of applications is also fully automated by describing **how the application should run**. This includes:

- **Triggers for automatic redeployment** after a change
- **Versioning of the application**, supporting rollbacks to a previous version
- Manual or automatic **scaling of components**
- **Different deployment strategies**, including allowing **graceful shutdown**, running **custom hooks** at a given point of the deployment or **rolling updates**



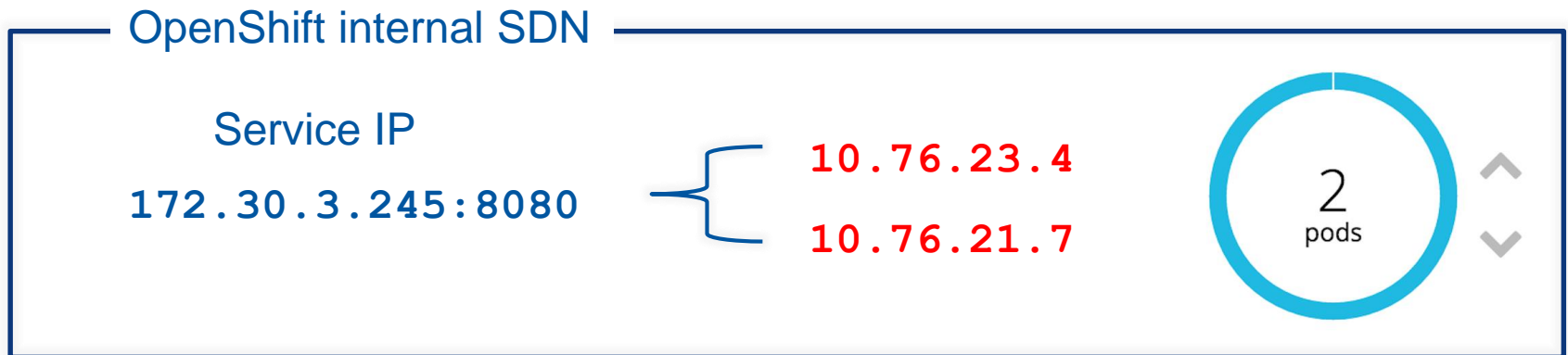
Load-balancing requests to your app

- A **Service** is an **internal** IP address and port that **load balances requests** to pods backed by a single Deployment Config



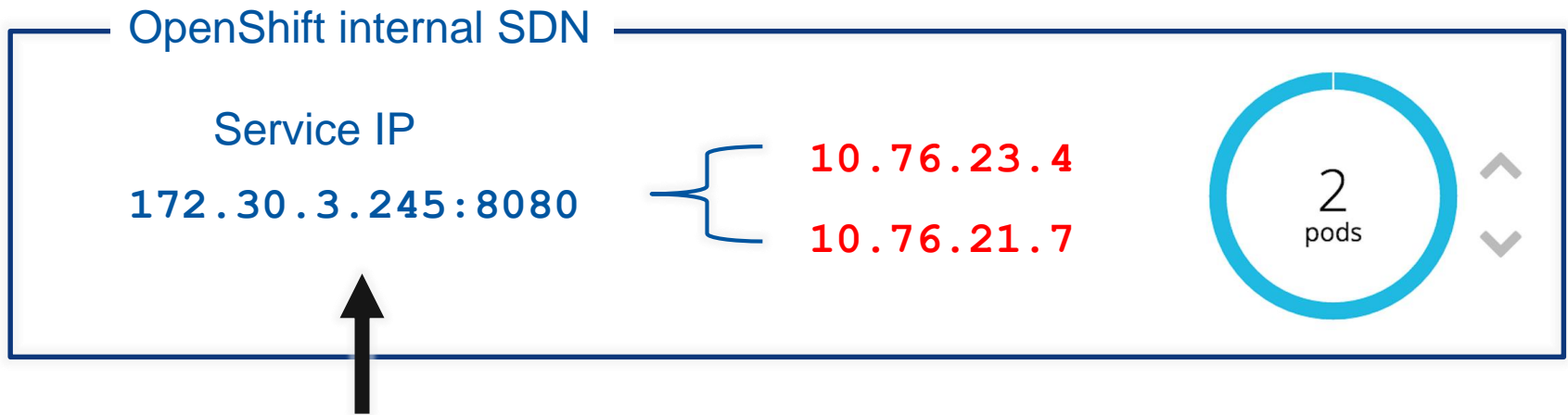
Load-balancing requests to your app

- A **Service** is an **internal** IP address and port that **load balances requests** to pods backed by a single Deployment Config



Load-balancing requests to your app

- A **Service** is an **internal IP** address and port that **load balances requests** to pods backed by a single Deployment Config
- A **Route** **exposes a Service at a host name**, so external clients can reach it by name
 - OpenShift registers on the DNS `<your-site>.web.cern.ch` for you
 - Routes in OpenShift can be secured with TLS (no need to provide a certificate!)



<https://myapp.web.cern.ch>

Persisting data across pod restarts

- By design, on-disk files stored in a pod **are always lost after a crash or a redeployment**, as containers will start back **with a clean state**
- To overcome this, Persistent Volumes provides external storage to applications
 - **Standard volumes** in the OpenShift cluster at CERN are based on **NFS**, which means **several pods can read and write to it at the same time**
 - All data stored in the persistent volumes **are regularly backed up**
 - **All pods replicas** of a **single DeploymentConfig** will mount the same volume
- Persistent Volumes with access to **EOS** and **CVMFS** are also available

OpenShift @ CERN

OpenShift projects are managed through the central web services portal at CERN:
<https://cern.ch/web>

Create new site [\[Open help \]](#)

Site category: [According to website lifecycle policy, test websites expire in 6 months](#)

Site name: [Build name >>](#)

Description:

Site type:

- Centrally hosted on DFS [?](#)
- EOS folder [?](#)
- Collaboration workspace [?](#)
- Drupal [?](#)
- PaaS Web Application [?](#)
- Social Community [?](#)

[How to choose?](#)

Create communities about topics that interest you, find answers to your questions and connect with others.

Owner (CERN login): [?](#)

I have read and agreed to the [CERN Computing Rules](#) and taken into account the [design guidelines](#) for websites and the [website lifecycle policy](#).

(All fields are mandatory)

DEMO 1

- Flask python deployment with OpenShift:
 - <https://gitlab.cern.ch/paas-tools/test-devs-cern>

DEMO 2

- Protect with SSO the previous demo

But wait, there's more...

- ImageStreams
- Autoscaling
- Secrets and ConfigMaps
- CronJobs
- Integrated monitoring
- CLI usage
- Interactive shell access
- ...

Documentation links

- PaaS Web applications KB articles
 - Getting started with OpenShift
- Openshift v3.6 Upstream documentation

The End

- Questions:
 - Now
 - Or later, if there is interest, a dedicated session can be arranged
- Contact:
 - Mattermost OpenShift channel
 - ServiceNow request