

OpenShift Pt.2

COMPASS NUM-APP

28.04.2021



Objectives

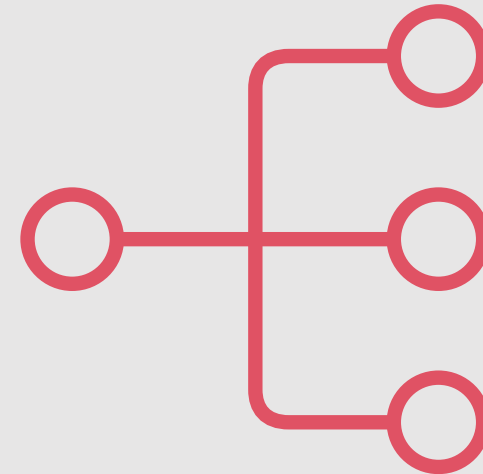
After this session, you should be able to:

- Interact with an OpenShift cluster via UI and CLI
- Understand the advantages of source-to-image and templates



Agenda

- Recap: Kubernetes & OpenShift
- OpenShift CLI
- Source To Image
- Templates



Necessary Preparation

Installation of OpenShift CLI

https://docs.openshift.com/container-platform/4.7/cli_reference/openshift_cli/getting-started-cli.html#installing-openshift-cli

RedHat Developer Sandbox

<https://developers.redhat.com/developer-sandbox>



Recap Kubernetes & OpenShift

Limitations of Containers

Production environment requirements:

- Communication between large number of containers
- Resource limits
- Need to increase/decrease number of running containers
- Quick response to service degradation
- Roll out of new service releases

Kubernetes - Overview

“Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.”

Kubernetes - Features

Kubernetes provides:

- Service discovery and load balancing
- Horizontal scaling
- Health-checks and self-healing
- Automated rollout
- Secrets and configuration management

OpenShift - Overview

What is Red Hat OpenShift Container Platform?

- Built on top of Kubernetes
- Set of modular components and services
- Provides production relevant capabilities for
 - Monitoring and Auditing
 - Security
 - Multitenancy
 - Application life-cycle management
 - Many more...

OpenShift - Features

Features provided by OpenShift:

- Integrated developer workflow
- Routes
- Metrics and logging
- Unified UI

Relevant OpenShift Resources

- Pod
 - A Pod is a group of one or more containers deployed to a single node.
- Service
 - A *Service* is a set of replicated pods. It decouples work definitions from the pods.
- Route
 - A *Route* is a load balancing mechanism used to expose services externally.
- Build Config
 - A Build Configuration (BC) defines a build process for new container images.
- Deployment Config
 - A Deployment Configuration (DC) defines the template for a pod and manages deploying new images or configuration changes.
- Secret
 - A Secret is an object that contains a small amount of sensitive data such as a password, a token, or a key.

Relevant OpenShift Resources

- Config Map
 - A ConfigMap (CM) is an API object used to store non-confidential data in key-value pairs.
- Image
 - An *Image* is a portable package containing all content, binaries, and configuration data that define a container instance
- Image Stream
 - An image stream comprises one or more Docker images identified by tags. It presents a single virtual view of related images, similar to a Docker image repository
- Persistent Volume
 - A *PersistentVolume* (PV) is a piece of storage in the cluster.
- Persistent Volume Claim
 - A *PersistentVolumeClaim* (PVC) is a request for storage by a user.

Demo

Quiz Time



[This Photo](#) by Unknown Author is licensed under [CC BY](#)

Quiz Time

Choose 3

Which of the following statements are correct regarding container limitations?

- A. Containers are easily orchestrated in large numbers.
- B. Lack of automation increases response time to problems.
- C. Containers do not manage application failure inside them.
- D. Containers are not load-balanced.
- E. Containers are heavily isolated packaged applications.

Quiz Time

Choose 3

Which of the following statements are correct regarding container limitations?

- A. Containers are easily orchestrated in large numbers.
- B. Lack of automation increases response time to problems.**
- C. Containers do not manage application failure inside them.**
- D. Containers are not load-balanced.**
- E. Containers are heavily isolated packaged applications.

Quiz Time

Choose 2

Which of the following statements are correct regarding Kubernetes?

- A. Kubernetes is a container.
- B. Kubernetes can only use Docker containers.
- C. Kubernetes is a container orchestration system.
- D. Kubernetes simplifies management, deployment, and scaling of containerized applications.
- E. Applications managed in a Kubernetes cluster are harder to maintain.

Quiz Time

Choose 2

Which of the following statements are correct regarding Kubernetes?

- A. Kubernetes is a container.
- B. Kubernetes can only use Docker containers.
- C. Kubernetes is a container orchestration system.**
- D. Kubernetes simplifies management, deployment, and scaling of containerized applications.**
- E. Applications managed in a Kubernetes cluster are harder to maintain.

Quiz Time

Choose 3

Which of the following statements are true regarding Red Hat OpenShift?

- A. OpenShift provides additional features to a Kubernetes infrastructure.
- B. Kubernetes and OpenShift are mutually exclusive.
- C. OpenShift hosts use Red Hat Enterprise Linux as the base operating system.
- D. OpenShift simplifies development incorporating a Source-to-Image technology and CI/CD pipelines.
- E. OpenShift simplifies routing and load balancing.

Quiz Time

Choose 3

Which of the following statements are true regarding Red Hat OpenShift?

- A. OpenShift provides additional features to a Kubernetes infrastructure.**
- B. Kubernetes and OpenShift are mutually exclusive.
- C. OpenShift hosts use Red Hat Enterprise Linux as the base operating system.
- D. OpenShift simplifies development incorporating a Source-to-Image technology and CI/CD pipelines.**
- E. OpenShift simplifies routing and load balancing.**

Exercise



Objective: Create instance of PostgreSQL database

Steps:

- 1. Login into Sandbox*
- 2. Use the +Add menu to open available db templates*
- 3. Use PostgreSQL to create an instance*

OpenShift CLI

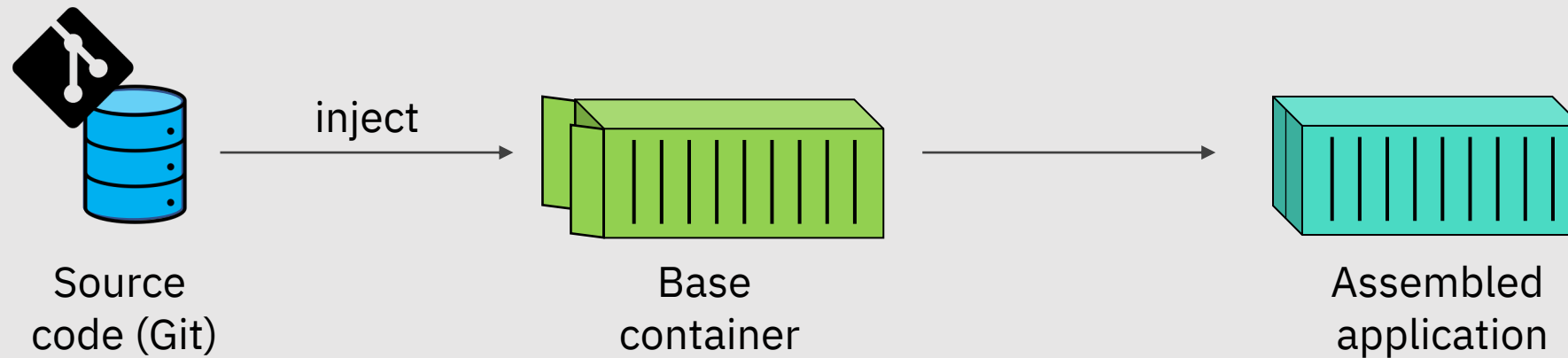
OpenShift CLI

- Command line interface to a cluster
- Should be used when:
 - Working directly with project source code
 - Scripting OpenShift operations
 - Bandwidth resources are limiting the use of the web console
- Client binary (oc) can be installed from web console
- Login credentials are available via web console

Demo

Source To Image

Source-to-Image (S2I) Process



S2I Build Process

S2I builder image: image containing required runtime environment for the application

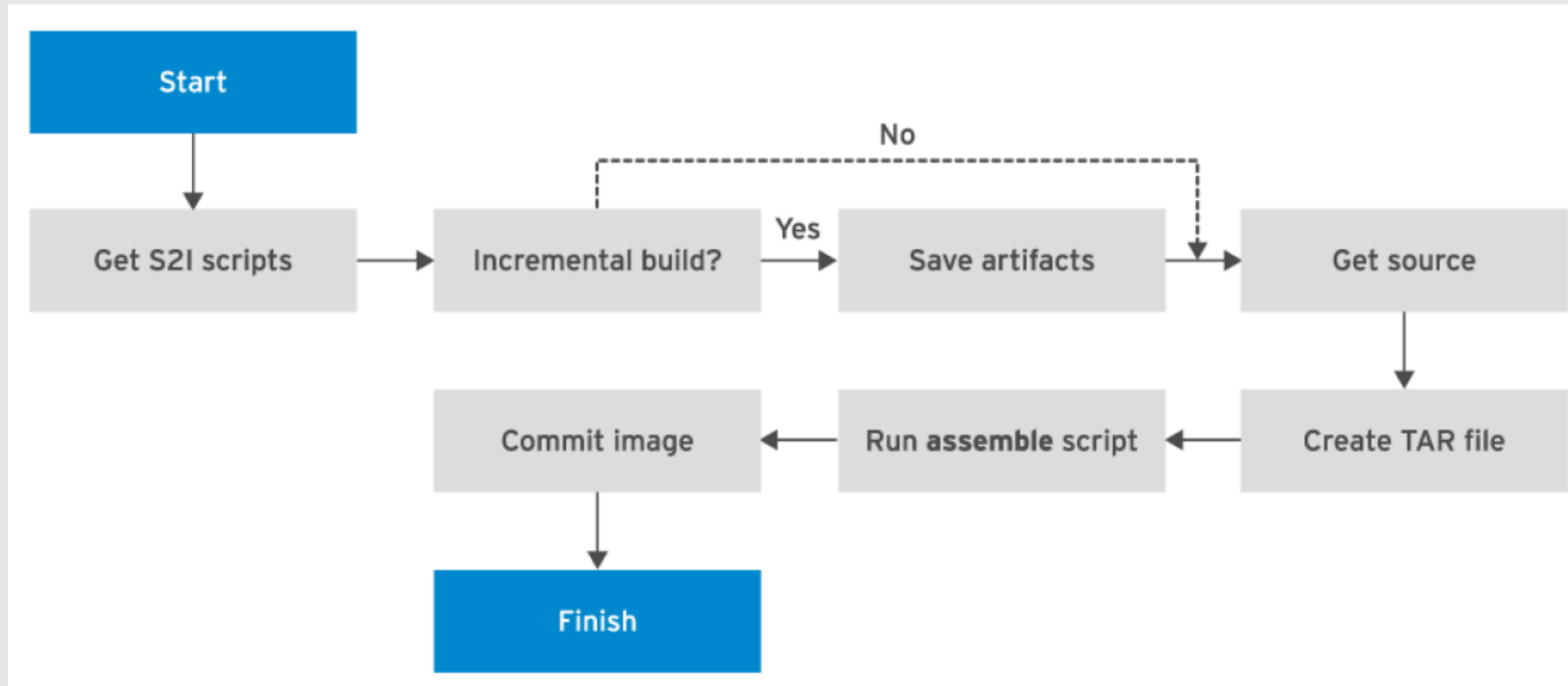
S2I scripts:

- assemble: build app and place in correct directories (mandatory)
- run: application execution (mandatory)
- save-artifacts: saved dependencies to tar file for subsequent builds
- Usage: usage description
- test/run: enables verification that image runs correctly

—————→ Scripts can be overwritten (.s2i/bin)

—————→ Usual script location: `io.openshift.s2i.scripts-url="image:///usr/libexec/s2i"`

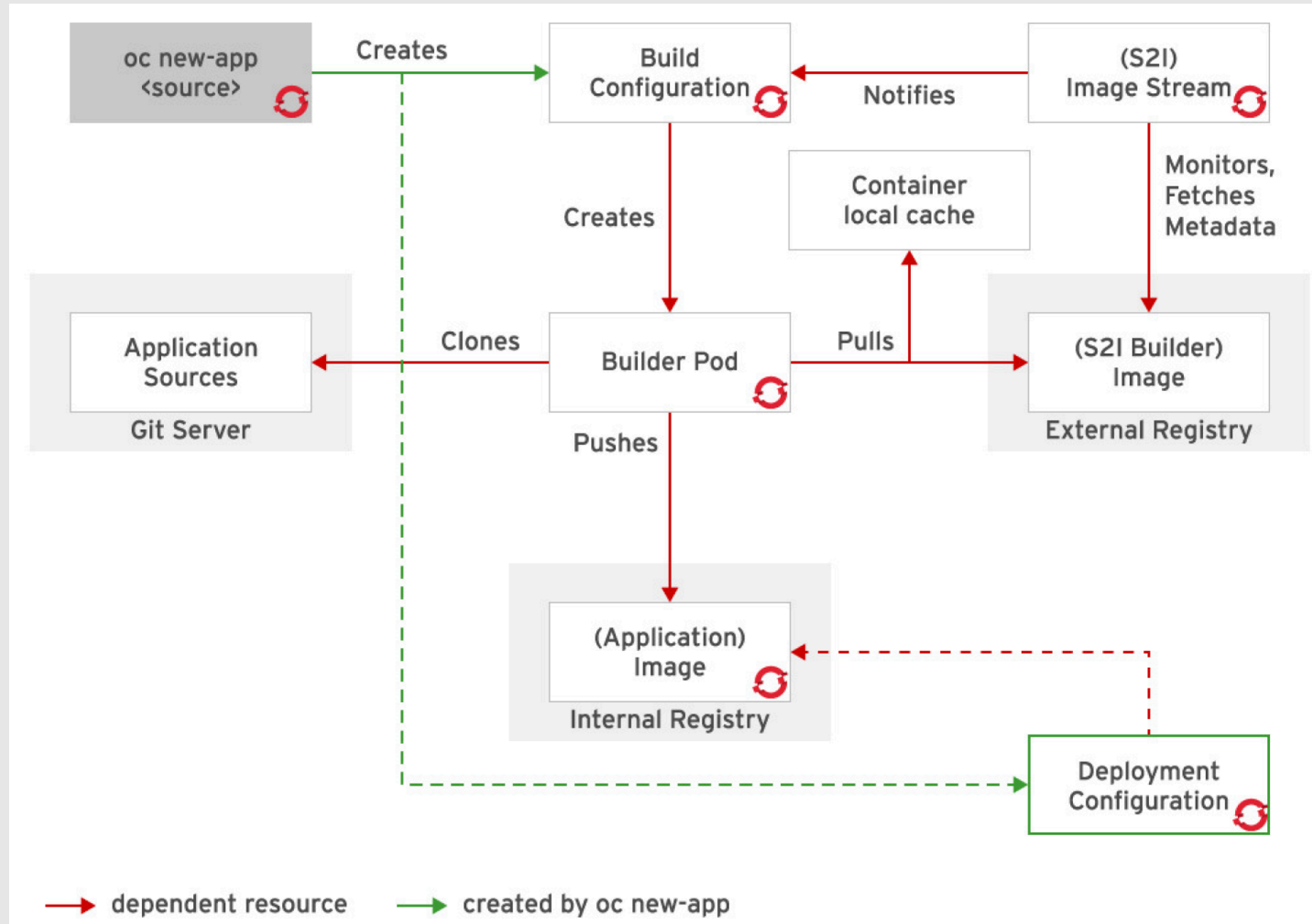
S2I Build Process



Source: RedHat

Demo

S2I with 'oc new-app'



Source: RedHat

S2I with "oc new-app"

```
oc new-app https://github.com/user/repo --source-secret=yoursecret
```



Using a private repository with S2I requires a **deploy key** with read access

Demo

Templates

Templates

YAML or JSON file consisting of a set of OpenShift resources

Enables deployment of a set of resources as a single unit
No need for deploying them individually

Syntax

1. Template Resource type
2. Optional annotations
3. Resource list
4. Reference to a template parameter
5. Parameter list
6. Label list

```

apiVersion: template.openshift.io/v1
kind: Template ❶
metadata:
  name: mytemplate
  annotations:
    description: "Description" ❷
objects: ❸
- apiVersion: v1
  kind: Pod
  metadata:
    name: myapp
  spec:
    containers:
      - env:
          - name: MYAPP_CONFIGURATION
            value: ${MYPARAMETER} ❹
        image: myorganization/myapplication
        name: myapp
        ports:
          - containerPort: 80
            protocol: TCP
parameters: ❺
- description: Myapp configuration data
  name: MYPARAMETER
  required: true
labels: ❻
  mylabel: myapp

```

Parameters

1. Generate value based on regex
2. Set mandatory parameters
3. Set default value

```
parameters:
- description: ACME cloud provider API key
  name: APIKEY
  generate: expression ❶
  from: "[a-zA-Z0-9]{12}"
- description: MyApp configuration data
  name: MYPARAMETER
  required: true ❷
- description: MyApp configuration data
  name: MYPARAMETER
  value: /etc/myapp/config.ini ❸
```

Creating Templates

1. Export existing resources

```
oc get -o yaml --export is,bc,dc,svc,route > mytemplate.yaml
```

2. Remove runtime information (e.g. status, creationTimestamp, uid, image, annotation...)

```
oc explain route
```

Hint: Copy existing templates and adapt them

Creating an App from a Template

List required parameters

```
oc process -f mytemplate.yaml --parameters
```

Create application from template

```
oc new-app --file mytemplate.yaml -p PARAM1=value1 -p PARAM2=value2
```

Demo

Exercise



Set-up Database

1. `oc get templates -n openshift`
2. `oc get templates -n openshift | grep postgresql`
3. `oc describe template postgresql-persistent -n openshift`
4. `oc get template postgresql-persistent -o yaml -n openshift`
5. `oc process --parameters -n openshift postgresql-persistent`
6. `oc new-app postgresql-persistent -p POSTGRESQL_DATABASE=num`
7. `oc status`

Exercise

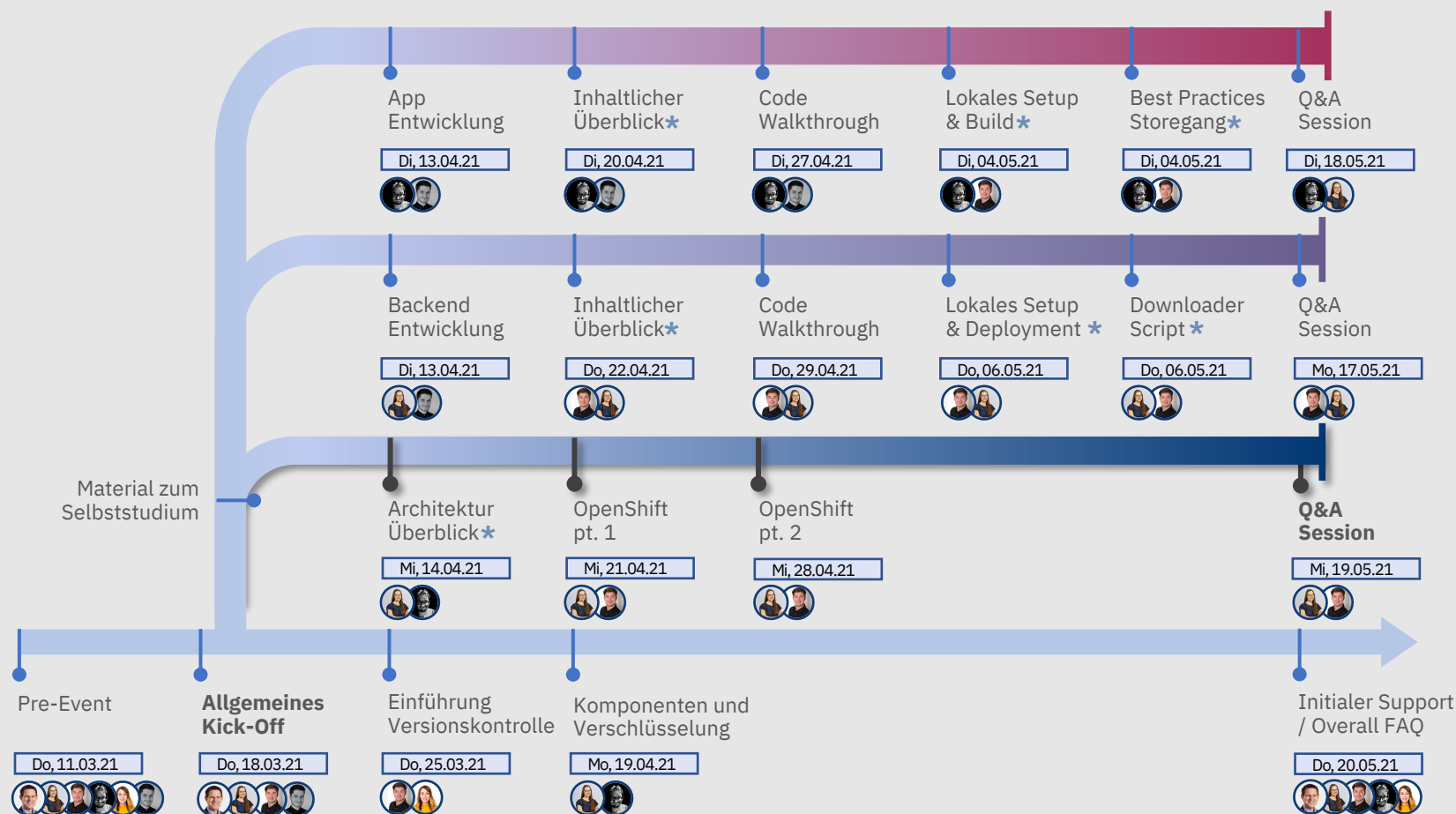


Verify database set-up

1. `oc get pods`
2. `oc rsh <container-name>`
3. `psql -d num -U userGUE`
4. `\dt`
5. `CREATE TABLE accounts (user_id serial PRIMARY KEY, password VARCHAR (50) NOT NULL, email VARCHAR (255) UNIQUE NOT NULL, created_on TIMESTAMP NOT NULL, last_login TIMESTAMP);`
6. `\q`
7. `exit`
8. Delete resources in cluster

Outlook

Platform Contributor Track



Links

Links

Containers

<https://www.docker.com/resources/what-container>

OC Cli

https://docs.openshift.com/container-platform/4.5/cli_reference/openshift_cli/getting-started-cli.html

Q&A

Q&A

*What questions
do you have?*

