

IBM

GRUNDING – IBM Tag 2

Alex Studer
Master@IBM Software Engineer

Alexander L. Schorno
Partner Technical Specialist – Data & AI



Who are we?



Alex Studer

Master@IBM Software Engineer

alex.studer@ibm.com



Alexander L. Schorno

Partner Technical Specialist – Data & AI

alexander.schorno@ibm.com

Agenda

Teil 1 – Theorie

- Repetition Cloud Computing
- Repetition Containerisierung
- Einführung PaaS

Teil 2 – Cloud Onboarding

- Getting Access to the IBM Cloud
- IBM Cloud Tour
- Red Hat OpenShift Tour

Teil 3 – Hands-on PaaS

- Exercise 1: SuperMario in the Cloud
- Exercise 2: Minimal Webserver
- Exercise 3: Source-to-Image Workflow



Repetition

Cloud Computing

- Wieso Cloud Computing?
- Was ist Cloud Computing?
- Wo steht die Cloud?
- Wie nutze ich die Cloud?



Vorteile einer Cloud

Flexibilität

- **Scalability:** Cloud Infrastruktur lässt sich bei Bedarf skalieren, um schwankende Arbeitslasten zu bewältigen.
- **Storage Options:** Je nach Sicherheitsanforderungen und anderen Faktoren können Benutzer zwischen öffentliche, privaten, oder hybriden Speicherangeboten wählen.
- **Control choices:** Mit As-a-Service-Optionen können Unternehmen den Grad ihrer Kontrolle selbst bestimmen. (SaaS, PaaS und IaaS)
- **Tool selection:** Benutzer können aus einem Angebot an vorgefertigten Tools und Funktionen wählen, um eine Lösung zu erstellen, die ihren spezifischen Anforderungen entspricht.
- **Security features:** Virtuelle private Cloud, Verschlüsselung und API-Schlüssel sorgen für die Sicherheit der Daten.

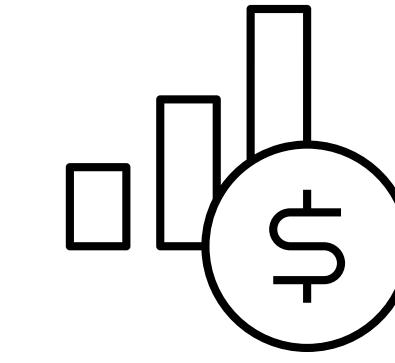
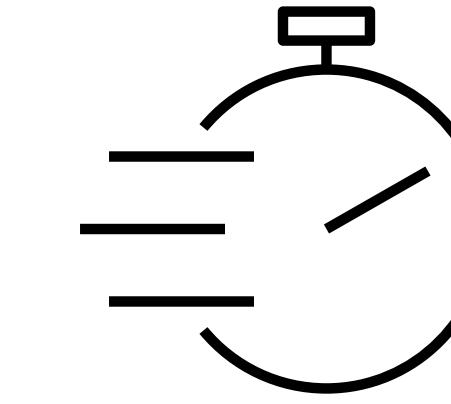
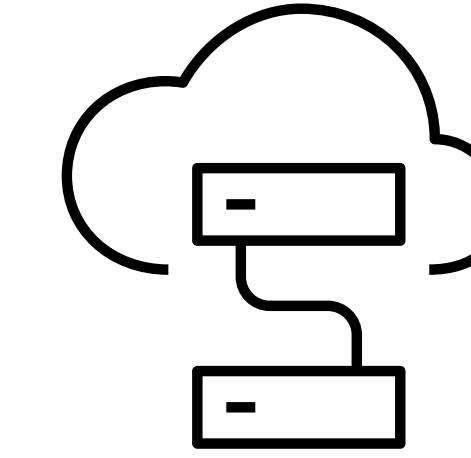
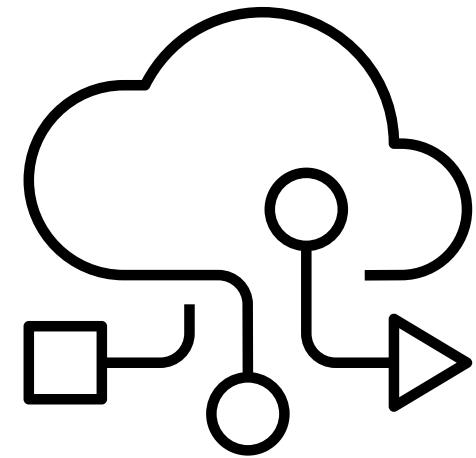
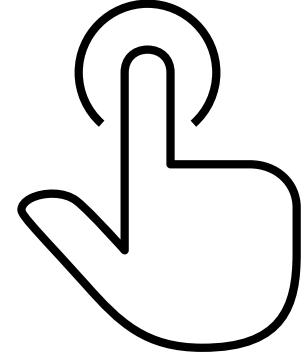
Effizienz & Wirtschaftlichkeit

- **Accessibility:** Cloud-basierte Anwendungen und Daten sind von praktisch jedem Gerät mit Internetanschluss zugänglich.
- **Speed to market:** Die Entwicklung in der Cloud ermöglicht es den Benutzern, ihre Anwendungen schnell auf den Markt zu bringen.
- **Data security:** Hardwareausfälle führen aufgrund der vernetzten Backups nicht zu Datenverlusten.
- **Savings on equipment:** Hardwareausfälle führen aufgrund der vernetzten Backups nicht zu Datenverlusten.
- **Pay structure:** Eine "Utility"-Bezahlstruktur bedeutet, dass die Benutzer nur für die Ressourcen zahlen, die sie auch nutzen.

Strategischer Wert

- **Streamlined work:** Cloud Service Provider (CSPs) verwalten die zugrunde liegende Infrastruktur, so dass sich Unternehmen auf die Anwendungsentwicklung und andere Prioritäten konzentrieren können.
- **Regular updates:** Dienstanbieter aktualisieren ihre Angebote regelmäßig, um den Nutzern die aktuellste Technologie zur Verfügung zu stellen.
- **Collaboration:** Weltweiter Zugriff bedeutet, dass Teams von verschiedenen Standorten aus zusammenarbeiten können.
- **Competitive edge:** Unternehmen können sich schneller bewegen als Konkurrenten, die IT-Ressourcen für die Verwaltung der Infrastruktur aufwenden müssen.

Was ist Cloud Computing? Eigenschaften



On demand self-service

Man erhält Zugriff auf Cloudressourcen über eine einfache Oberfläche ohne dass es dafür zwischenmenschliche Interaktion benötigt mit dem Anbieter.

Broad Network Access

Cloud Computerressourcen können über standard Netzwerkmechanismen auf verschiedenen Plattformen genutzt werden. Mobiltelefone, Tablets, Labtops, und Workstations.

Resource pooling

Die Computerressourcen des Cloudanbieters sind zu einem Pool von Ressourcen zusammengefasst. Diese physischen und virtuellen Ressourcen können dann dynamisch Konsumenten zugeordnet werden.

Rapid elasticity

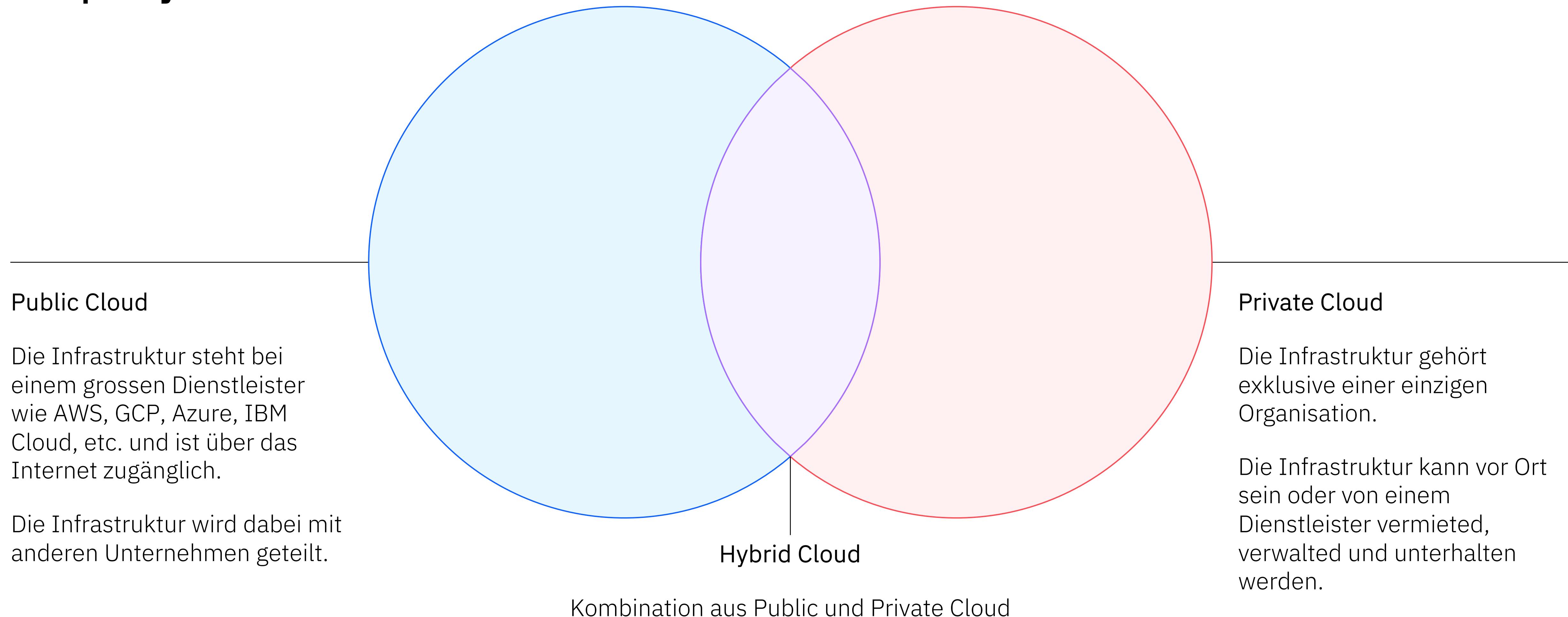
Cloudressourcen können dynamisch zur Verfügung gestellt und wieder freigegeben werden. Dies geschieht automatisiert und kann selbstständig skalieren. Für Konsumenten scheinen dabei die Ressourcen unbegrenzt zu sein und können jederzeit auf seine Bedürfnisse angepasst werden.

Measured Services

Man bezahlt nur was man nutzt oder für die Nutzung reserviert. Die Ressourcennutzung wird überwacht, gemessen und transparent rapportiert basierend auf deren Nutzung.

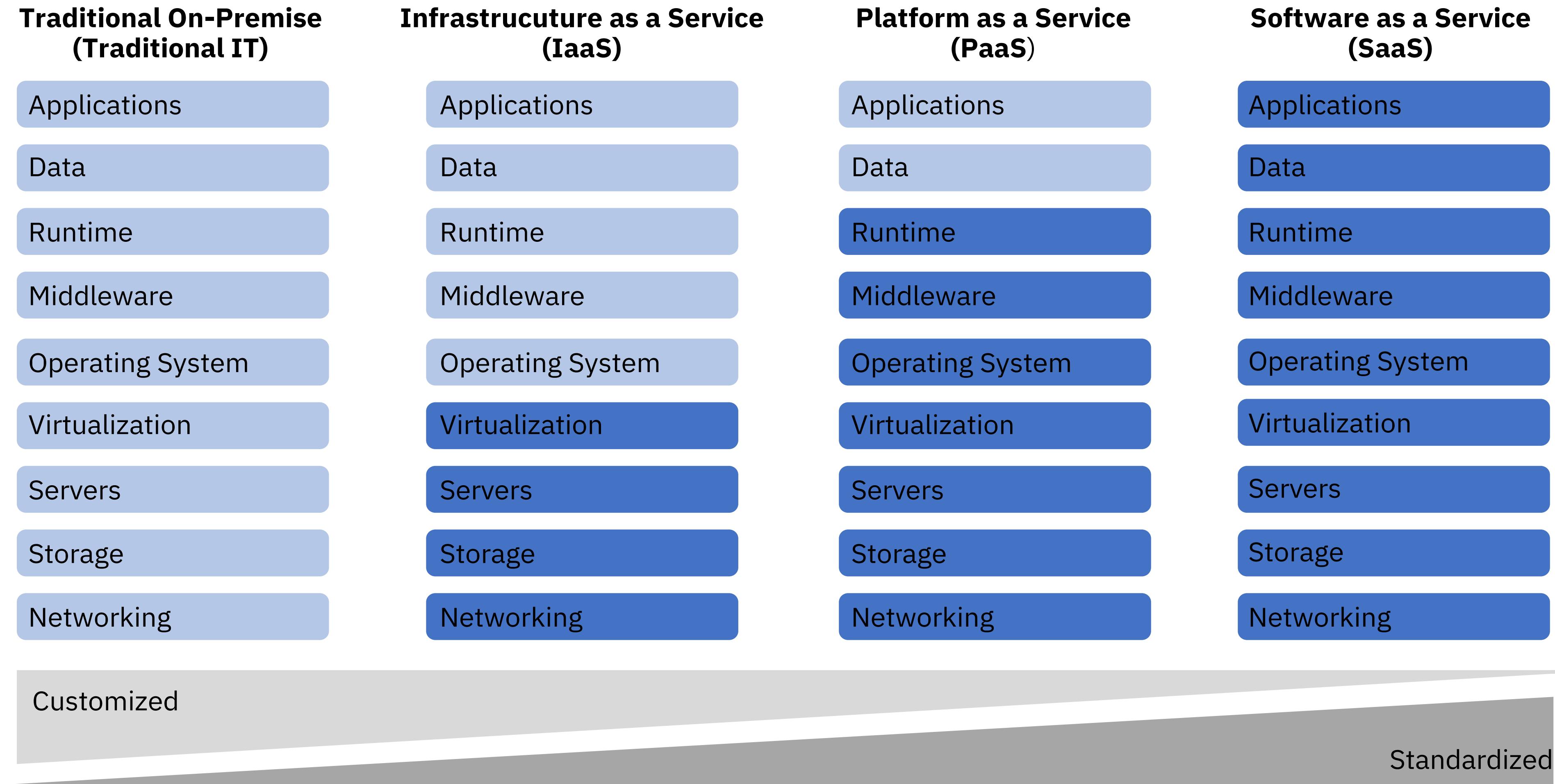
Wo steht die Cloud?

Deployment models



Wie nutze ich die Cloud?

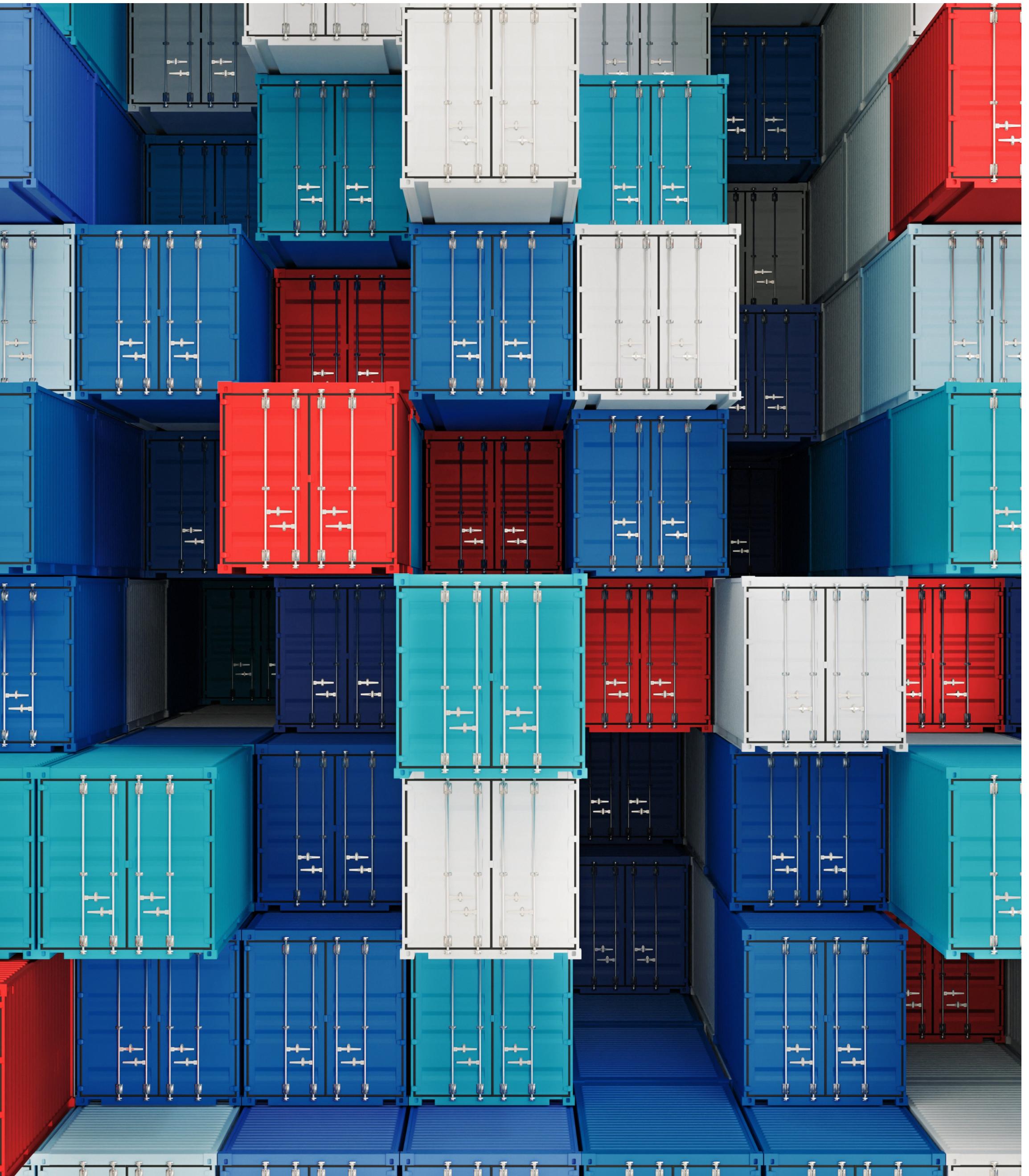
Service models



Repetition

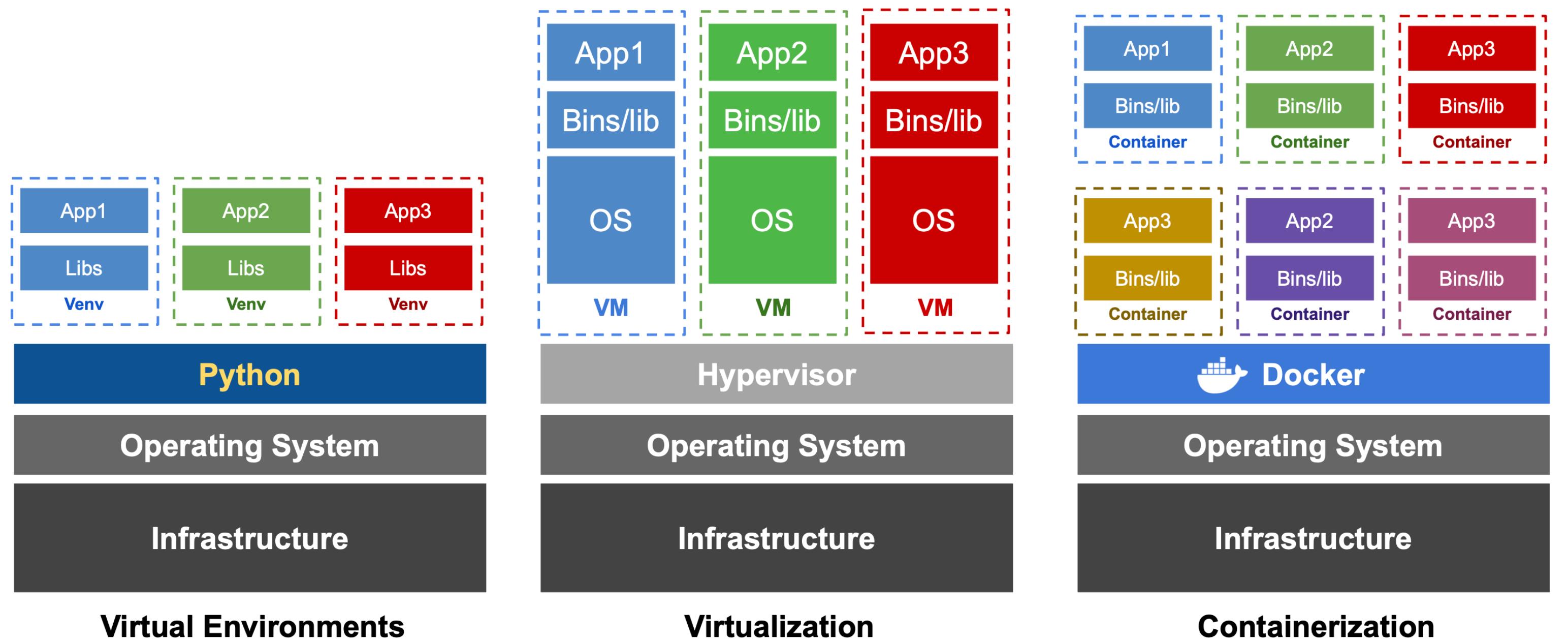
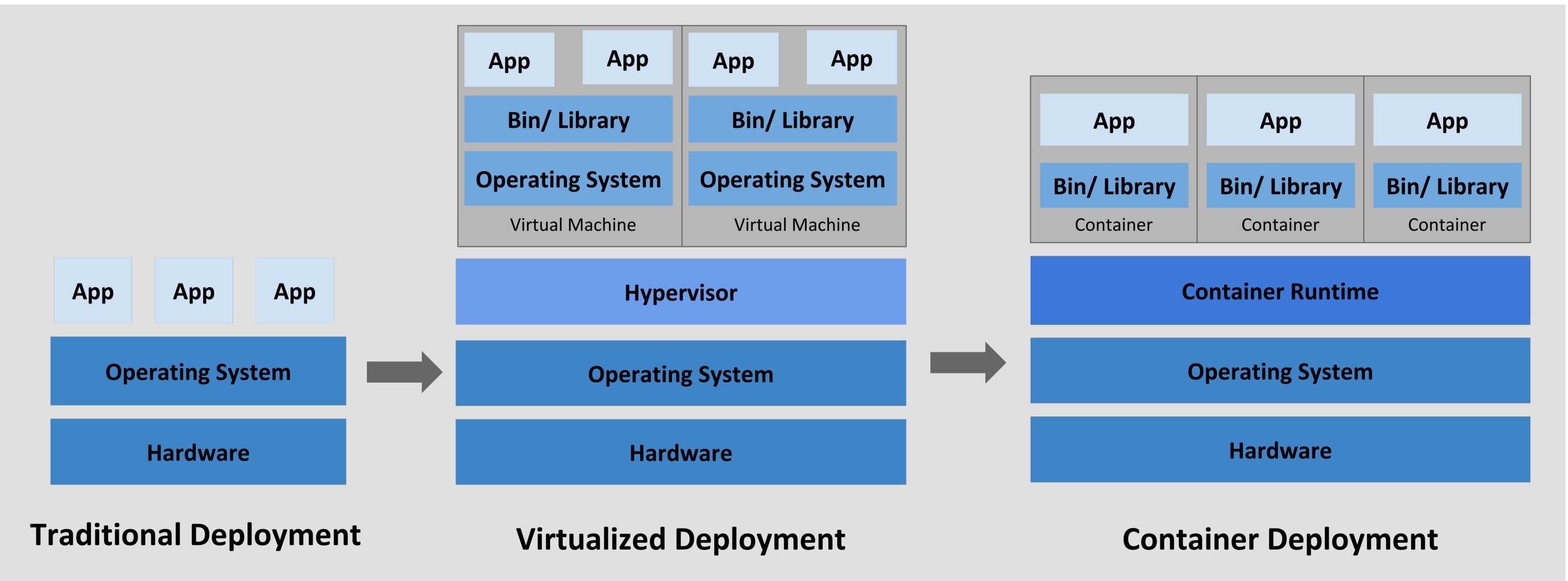
Containerisierung

- Was nützt uns Containerisierung?
- Wie unterscheiden sich Container von virtuellen Maschinen?
- Welche docker commands um ein Container image lokal auszuführen?
- Wo findet man Informationen für Software Tools wie Docker?



Eine kleine Geschichte der Virtualisierung

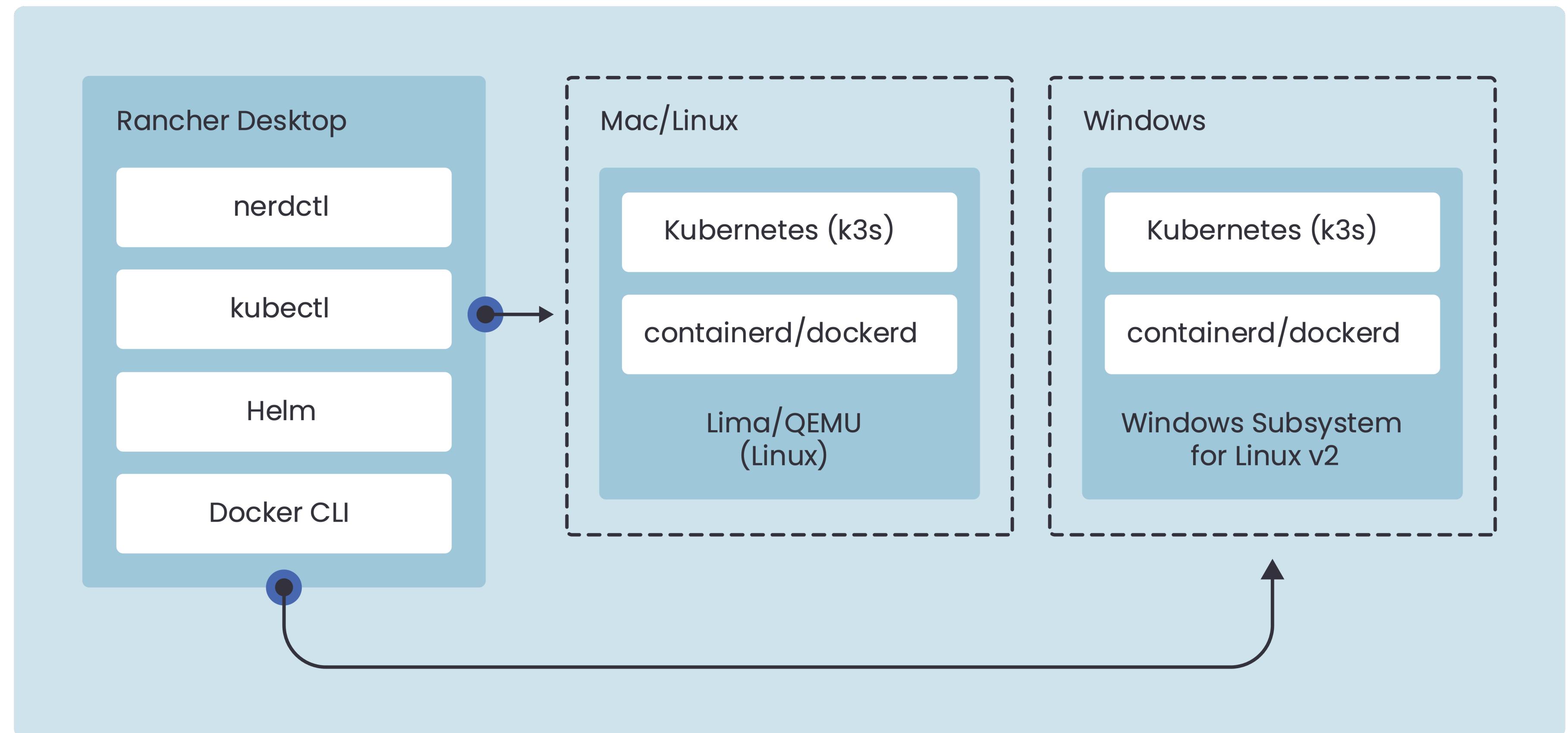
- 1960s: Time sharing on mainframes
- 1990s/2000s: Wide-spread adoption of hardware virtualization (server virtualization, virtual machines)
- 2008: Release Linux Containers (LXC)
- 2013: Release Docker
- 2017: Release OCI Image Spec



Rancher Desktop



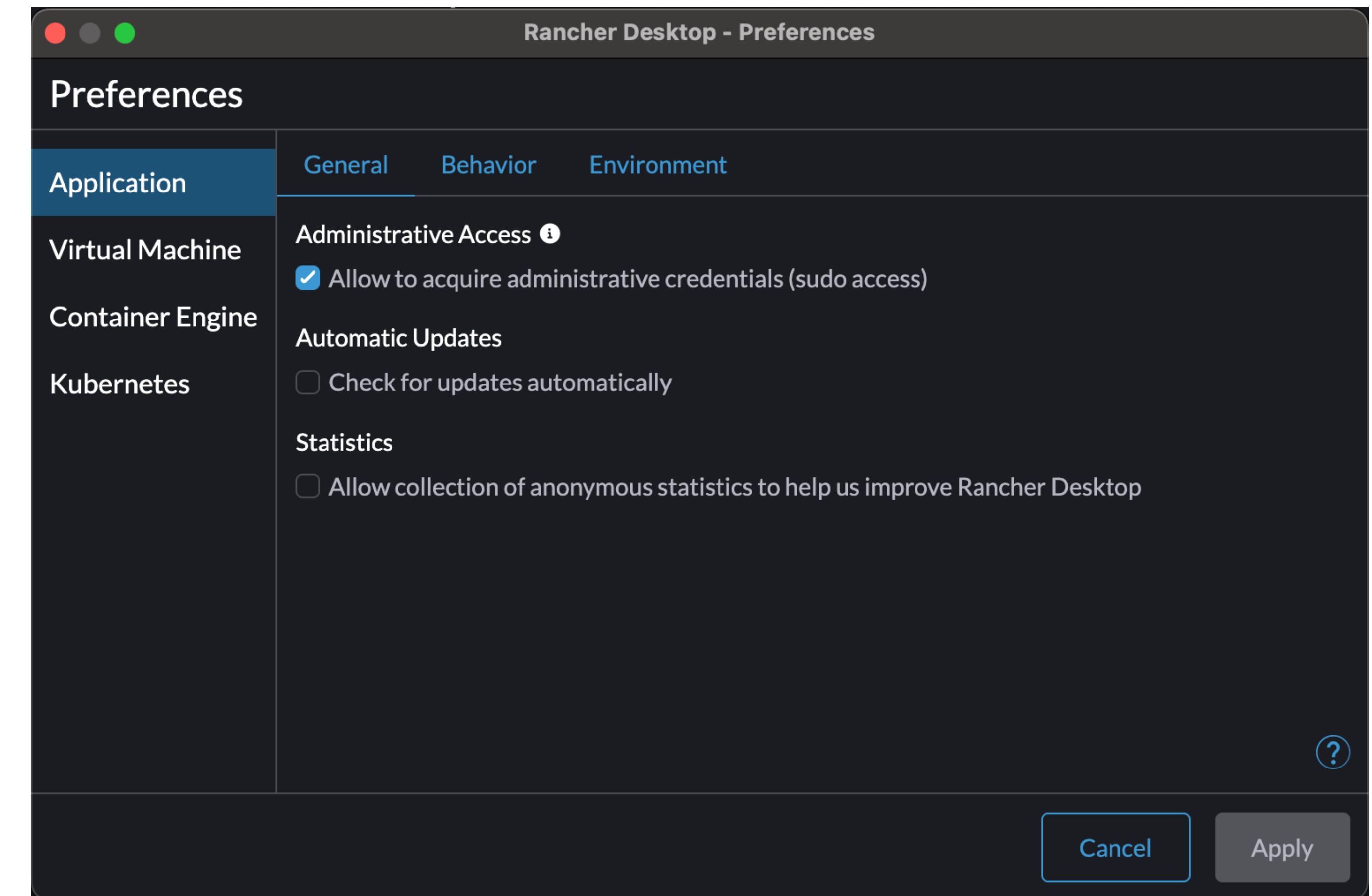
RANCHER DESKTOP



Install Rancher Desktop

Windows & MacOS

- Folge den Installations Anleitung unter:
<https://docs.rancherdesktop.io/getting-started/installation>
- Wähle Container Engine: dockerd (moby)
- Schalte Kubernetes nicht ein; setze kein Häckchen bei «Enable Kubernetes»



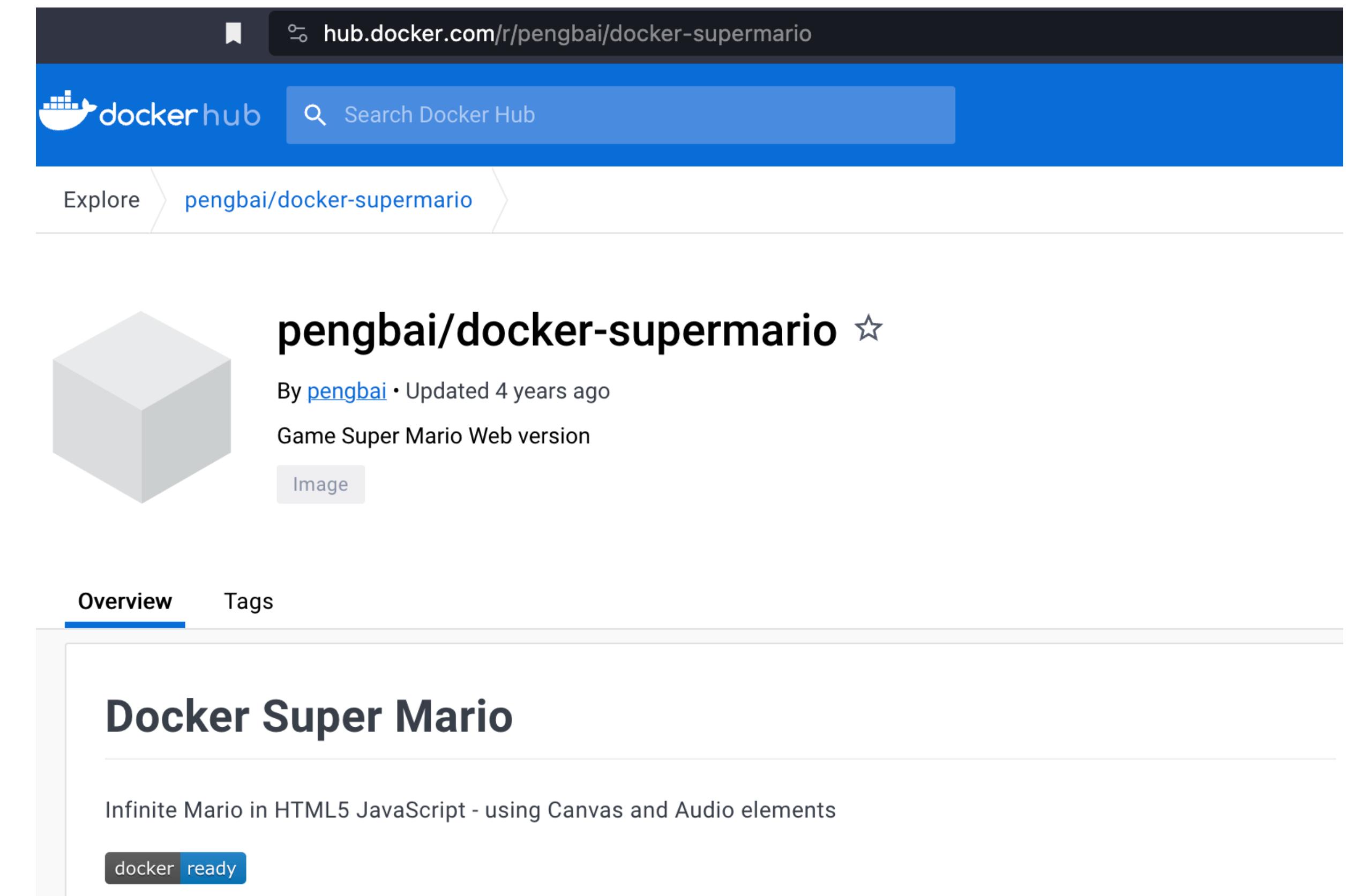
Download Container Image

1. Gehe auf Docker Hub und suche das Container Image ersichtlich auf dem Bild rechts.
2. Wir laden ein Container Image von der Container Registry «DockerHub» herunter. Dazu führe den folgenden Befehl in deiner Shell aus:

```
docker pull <image_name>
```

3. Prüfe ob das Container Image heruntergeladen wurde im GUI unter Images oder mit dem Command:

```
docker image ls
```



Run a Container locally

Aufgabe 1:

- Führt die selben Kommandos wie auf dem Bild unten aus.
- Bevor ihr «docker stop» ausführt, unterbricht und öffnet ein Webbrowserfenster
- Gebt folgende URL ein: <http://localhost:8600> und der Spielspass kann losgehen.
- Sobald ihr genug gespielt habt, schliesst das Browserfenster und macht mit docker stop weiter.

Aufgabe 2:

- Findet heraus was die Kommandos genau machen.
Wir lösen am Ende der Stunde gemeinsam auf.
- Der Befehl «help» kann hinter ein Kommando angehängt werden um Information direkt in der Shell zu erhalten. Beispiel
`docker help`

```
~ ➔ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
~ ➔ docker run -d -p 8600:8080 pengbai/docker-supermario
6d0b96aed14c96711042036bf6959b9f77c2eee363a0679b2f0c88ded38f99b3
~ ➔ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
6d0b96aed14c   pengbai/docker-supermario   "catalina.sh run"   4 seconds ago   Up 3 seconds   0.0.0.0:8600→8080/tcp,  :::8600→8080/tcp   gracious_swartz
~ ➔ docker stop gracious_swartz
gracious_swartz
~ ➔ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
```

Hands-on Mario Jump and Run



Plattform as a Service

PaaS

Platform as a Service (PaaS)

Applications

Data

Runtime

Middleware

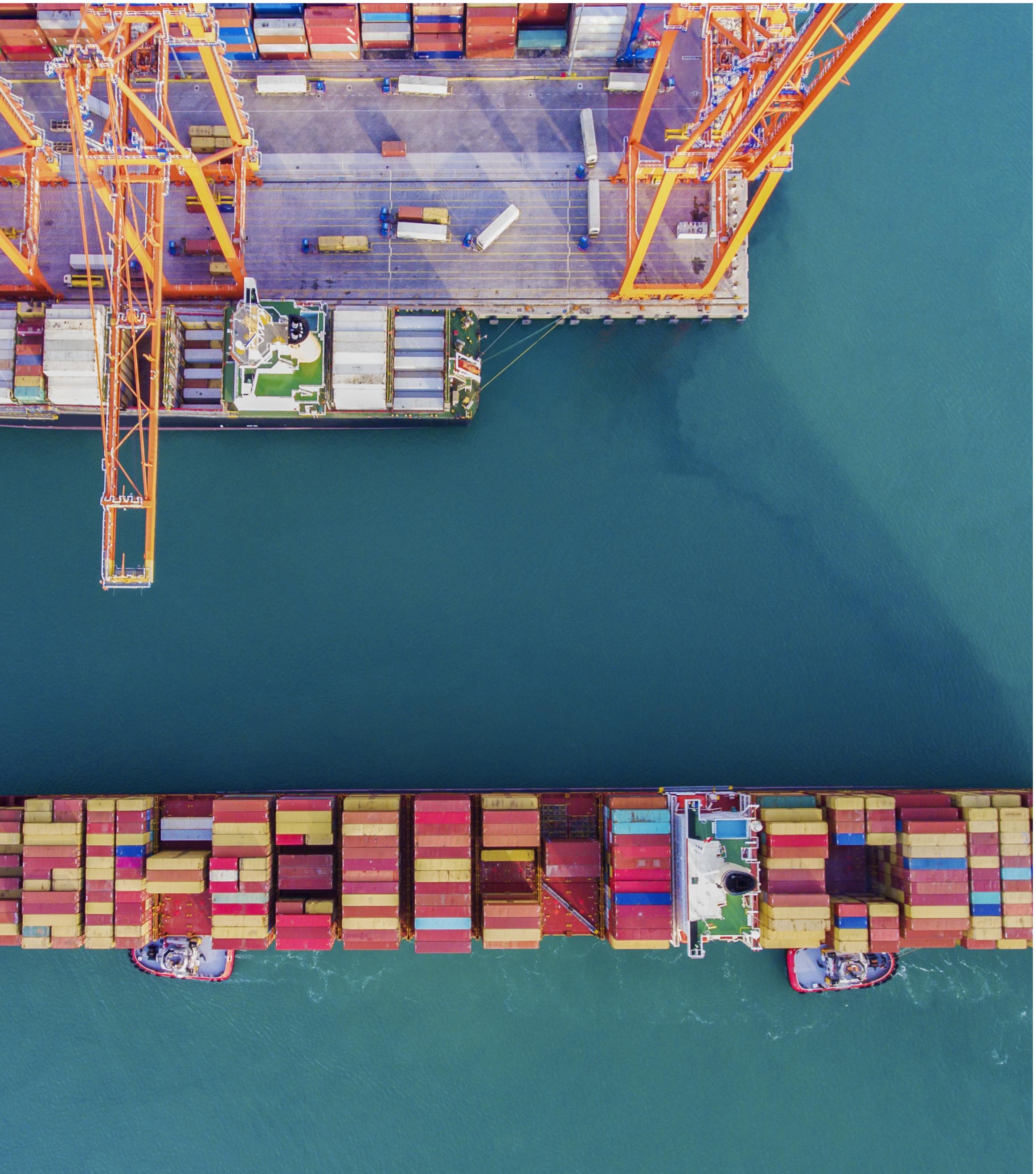
Operating System

Virtualization

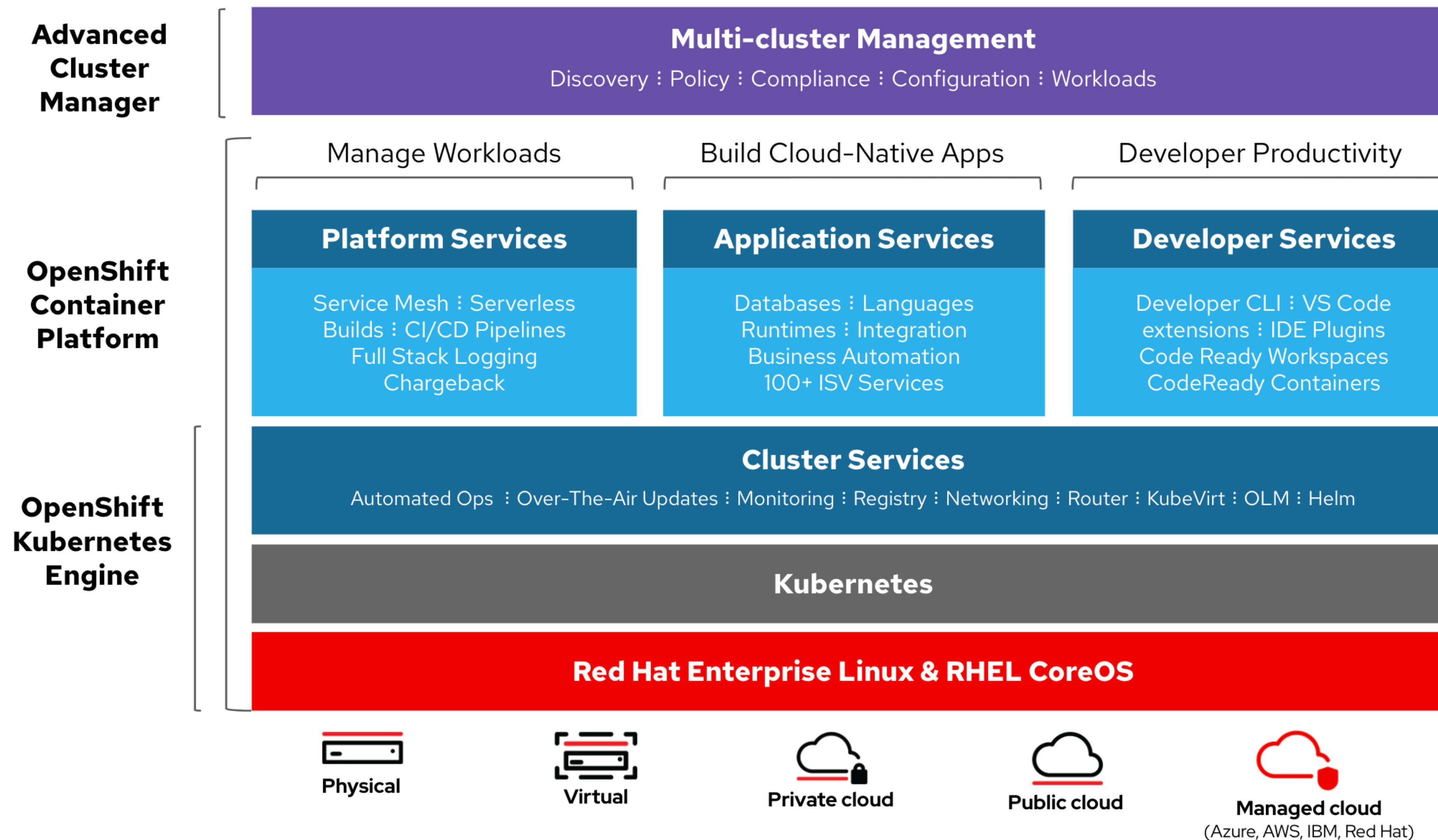
Servers

Storage

Networking



Red Hat OpenShift



IBM Cloud Onboarding



IBM Cloud

Account

Action required: You are invited to join an account in IBM Cloud

Type: Account
Update Time: 1 Nov 2023, 10:30 AM UTC

View notification (in subscribed accounts):

Hi alstuder@ethz.ch,

Alex Studer invited you to join the following IBM Cloud account: HSLU Digital Engineer (Account ID: 78d74408dc21484dad51b28f0415ecbc).

Join now.
The link to join the account expires in 30 days.

Thank you,
IBM Cloud

In case of any issues, please contact [IBM Cloud support](#).

© Copyright IBM Corporation 2014, 2023.

IBM



Dashboard

[Edit dashboard](#)[Create resource](#)

For you

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.



Build a machine learning model

Build a predictive machine learning model with an API to be used in applications, and retrain the model with feedback data.

[Recommended](#)

2 hr

[Getting started](#)

5 min

[Popular](#)

3 min

[Popular](#)

2 min

[Recommended](#)

3 min

[Getting started](#)

1 min

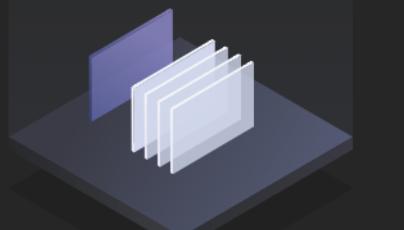
[Popular](#)

News

[View all](#)

- Leveraging IBM Cloud for electronic design automation (EDA) workloads
- Operationalize and automate FinOps with Apptio Cloudability and IBM Turbonomic
- Enterprise-managed IAM: An SRE team case study
- 3 keys to building a robust hybrid cloud risk strategy

Recent support cases

[View all](#)

You can view a summary of your support cases here after you submit them. [Learn more about how to get support.](#)

[View usage](#)

Usage

Planned maintenance

[View all](#)

Clear skies!

You can view your scheduled maintenance events here.

IBM Cloud status

[View all](#)

No issues

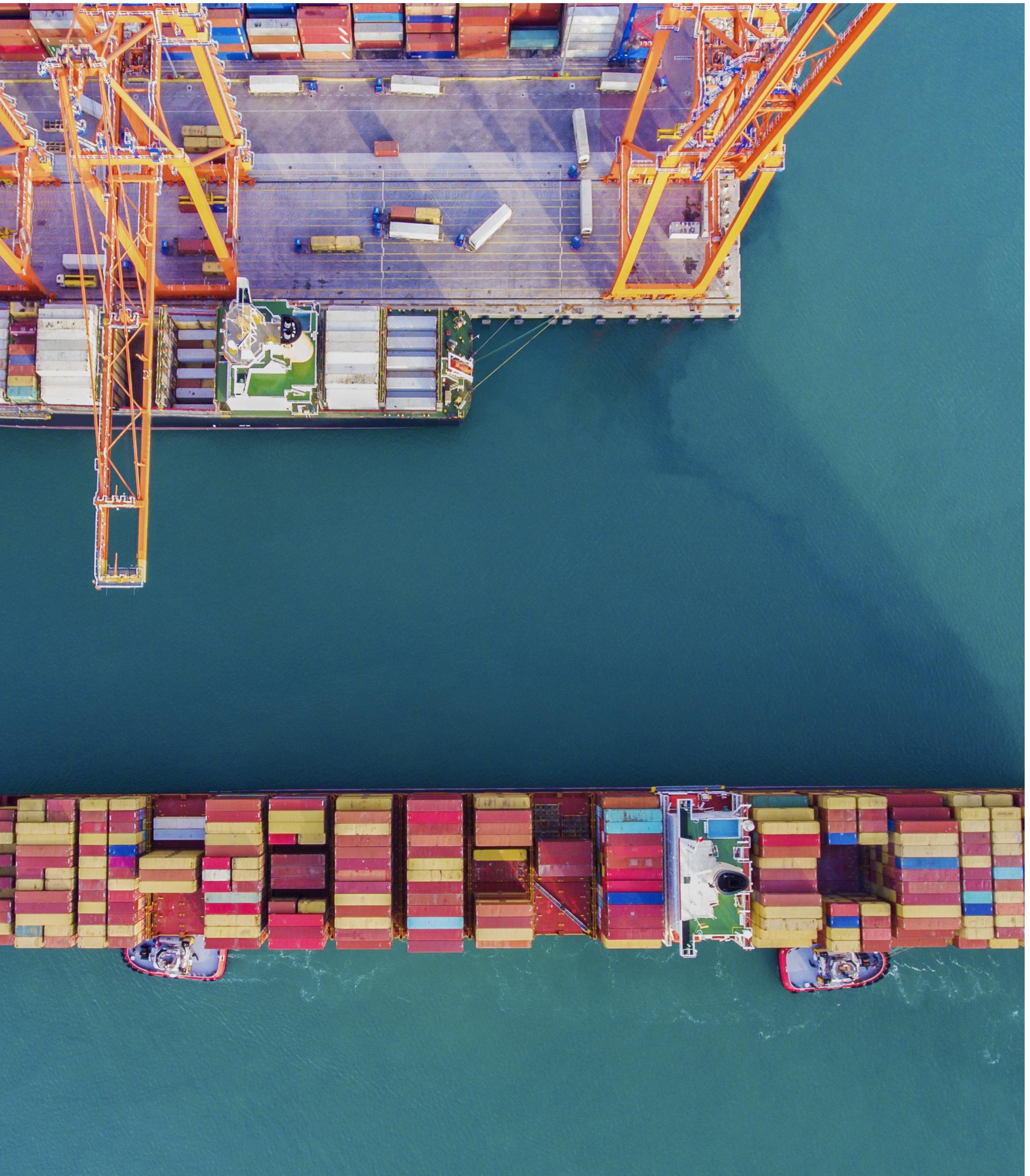


User access

[Manage users](#)

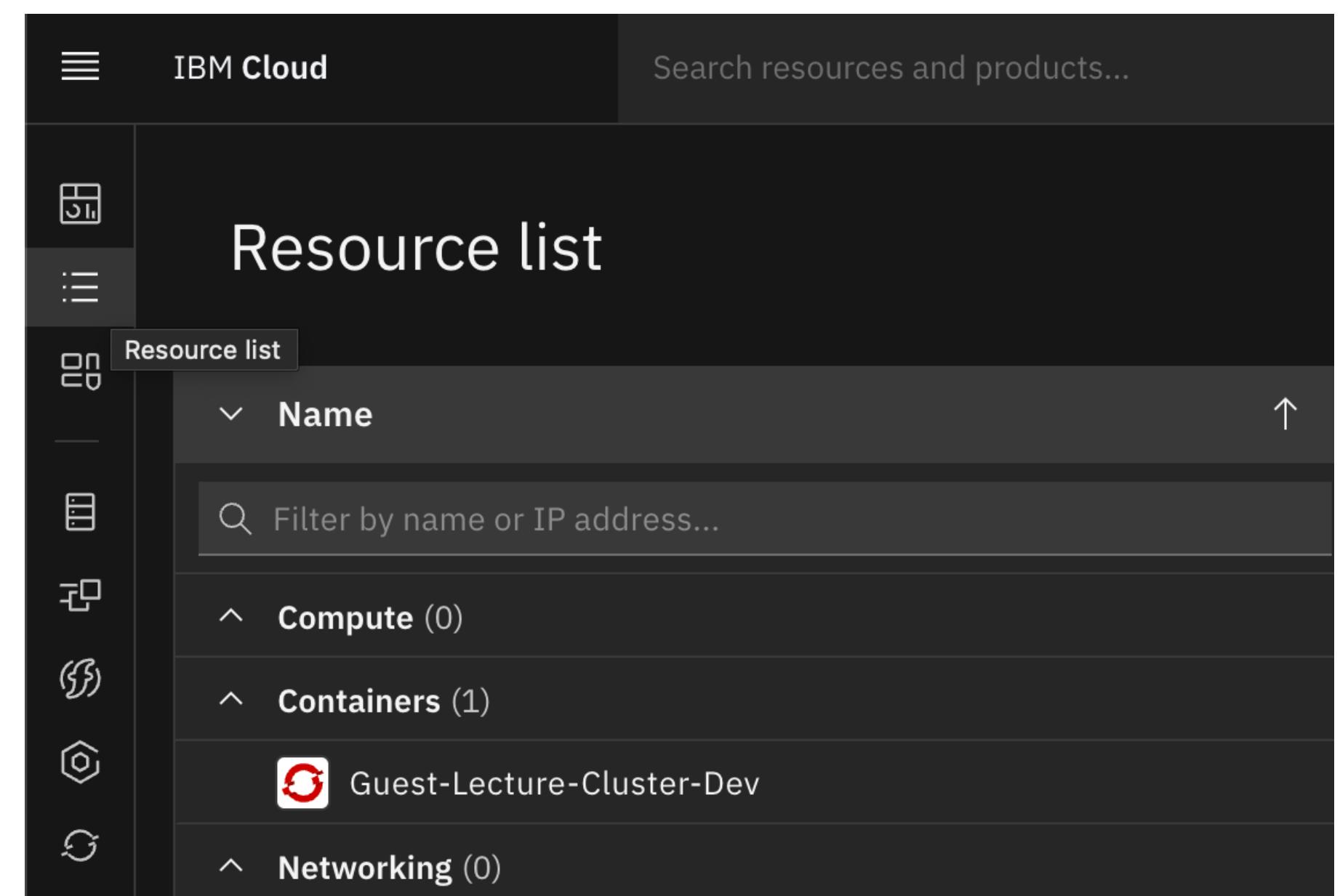
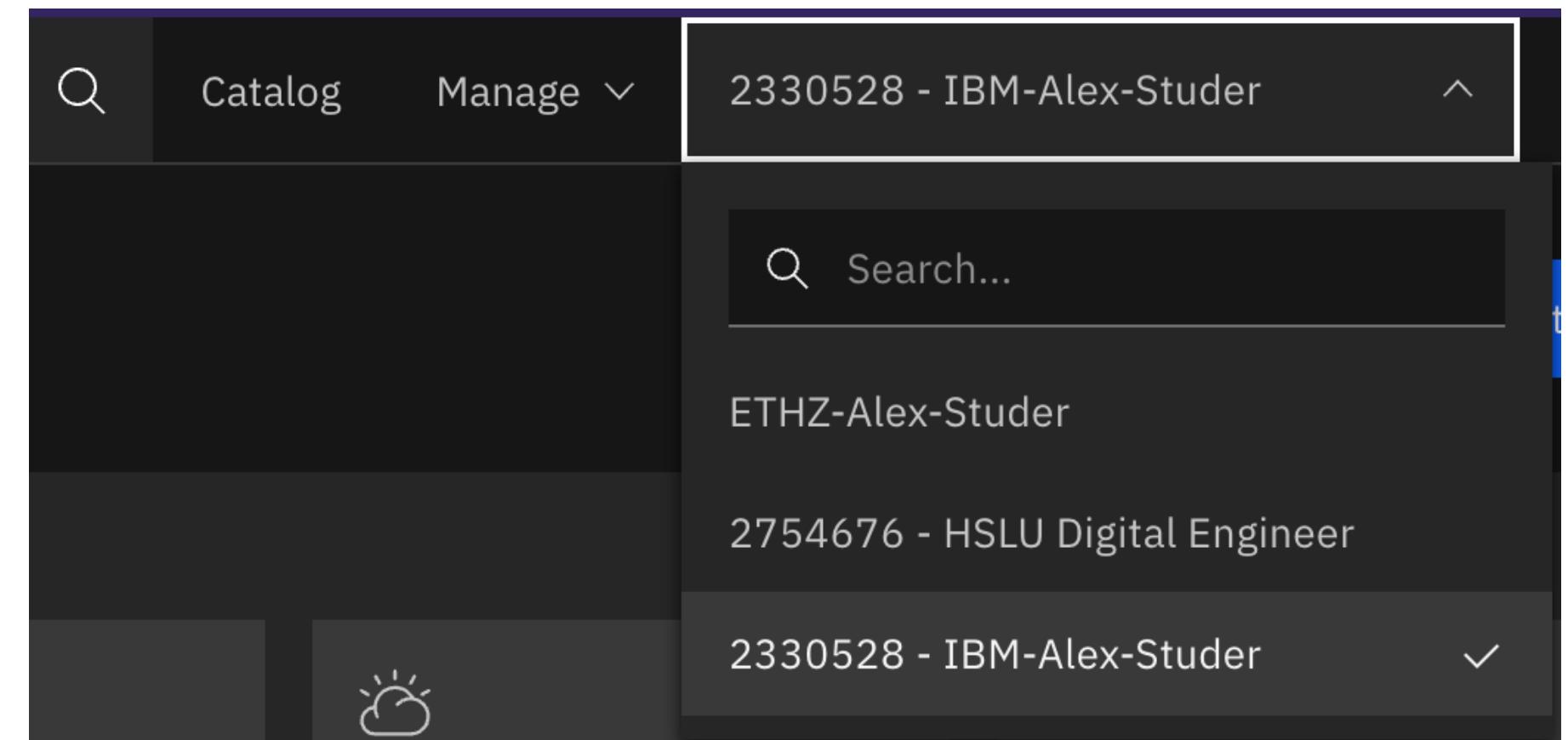
Hands-on PaaS

with Red Hat OpenShift



Open Red Hat OpenShift

- Wechselt auf den Account «IBM-Alex-Studer» mittels drop-down Menu oben rechts (siehe Bild oben rechts)
- Öffnet die Resourcen Liste und navigiert zur OpenShift Platform «Guest-Lecture-Cluster-Dev» unter Containers (siehe Bild unter rechts)
- Wenn der Service geöffnet ist, könnt ihr via der blauen Schaltfläche «OpenShift web consol» das GUI von OpenShift öffnen.



Red Hat OpenShift

- Wechselt in OpenShift auf die «Developer» Ansicht
- Mit dem drop-down Menu «Project» erreicht ihr den Button «Create Project» mit dem ihr ein neues Projekt anlegen könnt (siehe Bild rechts)
- Bennt euer Projekt nach dem folgenden Muster «student-vorname-nachname» und erstellt dieses (siehe Bild rechts)
- Navigiert mit dem Project drop-down Menü zu eurem projekt.

Create Project

An OpenShift project is an alternative representation of a Kubernetes namespace.

[Learn more about working with projects ↗](#)

Name * ⓘ

Display name

Description

Project: All Projects ▾

Select project...

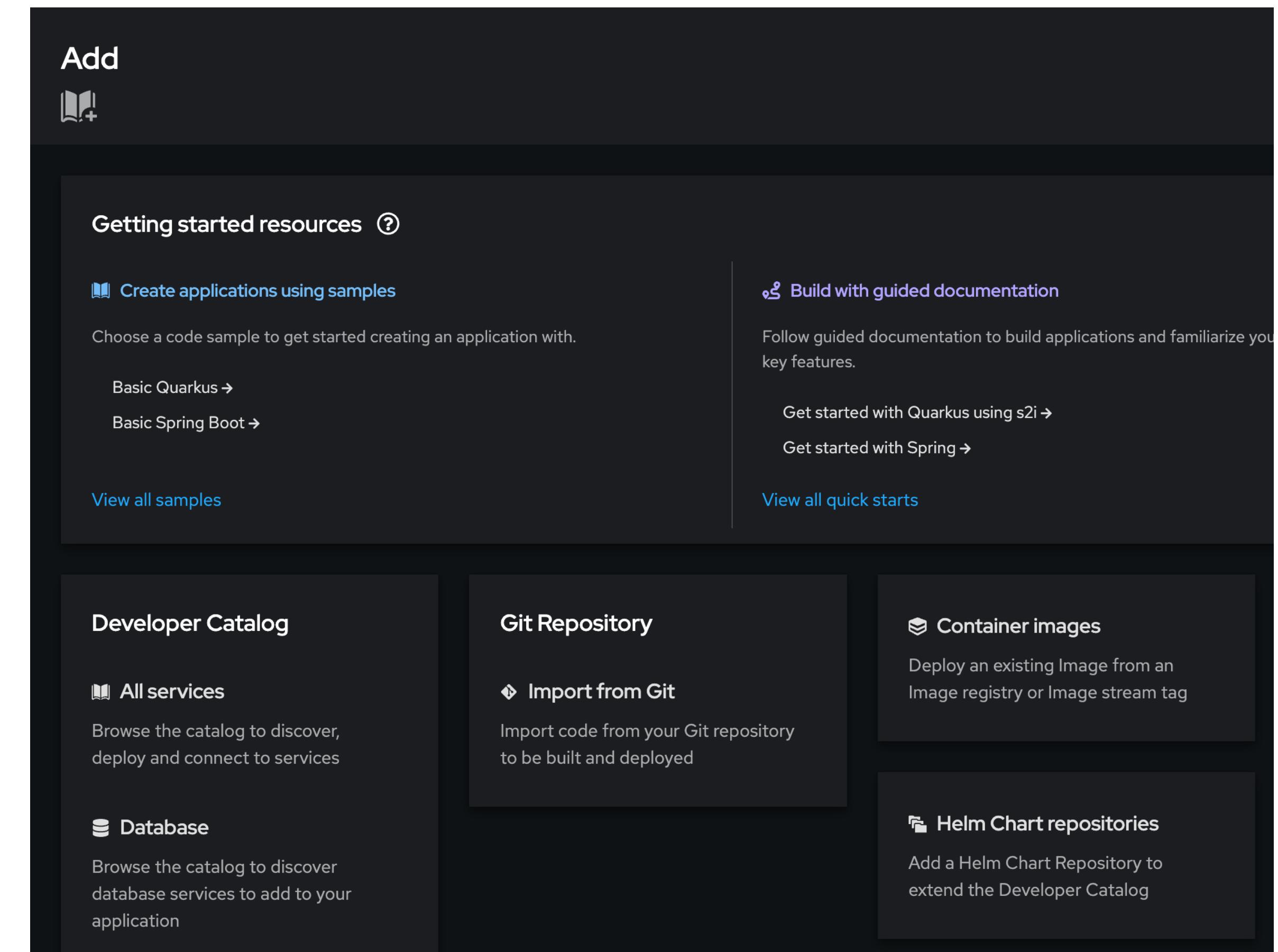
Show default projects

Projects	Star
All Projects	★
calico-system	★
demo-game	★
ibm-cert-store	★
ibm-odf-validation-webhook	★
ibm-system	★
student-alex-studer	★
student-alexander-schorno	★
tigera-operator	★

Exercise 1

Mario Bros. on the cloud

- Bleibt in der «Developer» Ansicht für dieses und alle folgenden Übungen
- Wechselt zur «Add» Oberfläche über die Navigation auf der Linkenseite des GUIs
- Wählt die Schaltfläche «Container Images» aus



Exercise 1

Mario Bros. on the cloud

- Fügt den Namen des Container Images ein (analog zum Bild rechts) -> pengbai/docker-supermario
- Alles andere sollte dann automatisch ergänzt werden.
- Drückt auf «Create» und wechselt anschliessend auf die «Topology» Ansicht.

Deploy Image

Image

Deploy an existing Image from an Image Stream or Image registry.

Image name from external registry
pengbai/docker-supermario ✓

Validated

To deploy an Image from a private registry, you must [create an Image pull secret](#) with your Image registry credentials.

Allow Images from insecure registries

Image stream tag from internal registry

Runtime icon

 openshift ▼

The icon represents your Image in Topology view. A label will also be added to the resource defining the icon.

General

Application name

docker-supermario-app

A unique name given to the application grouping to label your resources.

Name *

docker-supermario

A unique name given to the component that will be used to name associated resources.

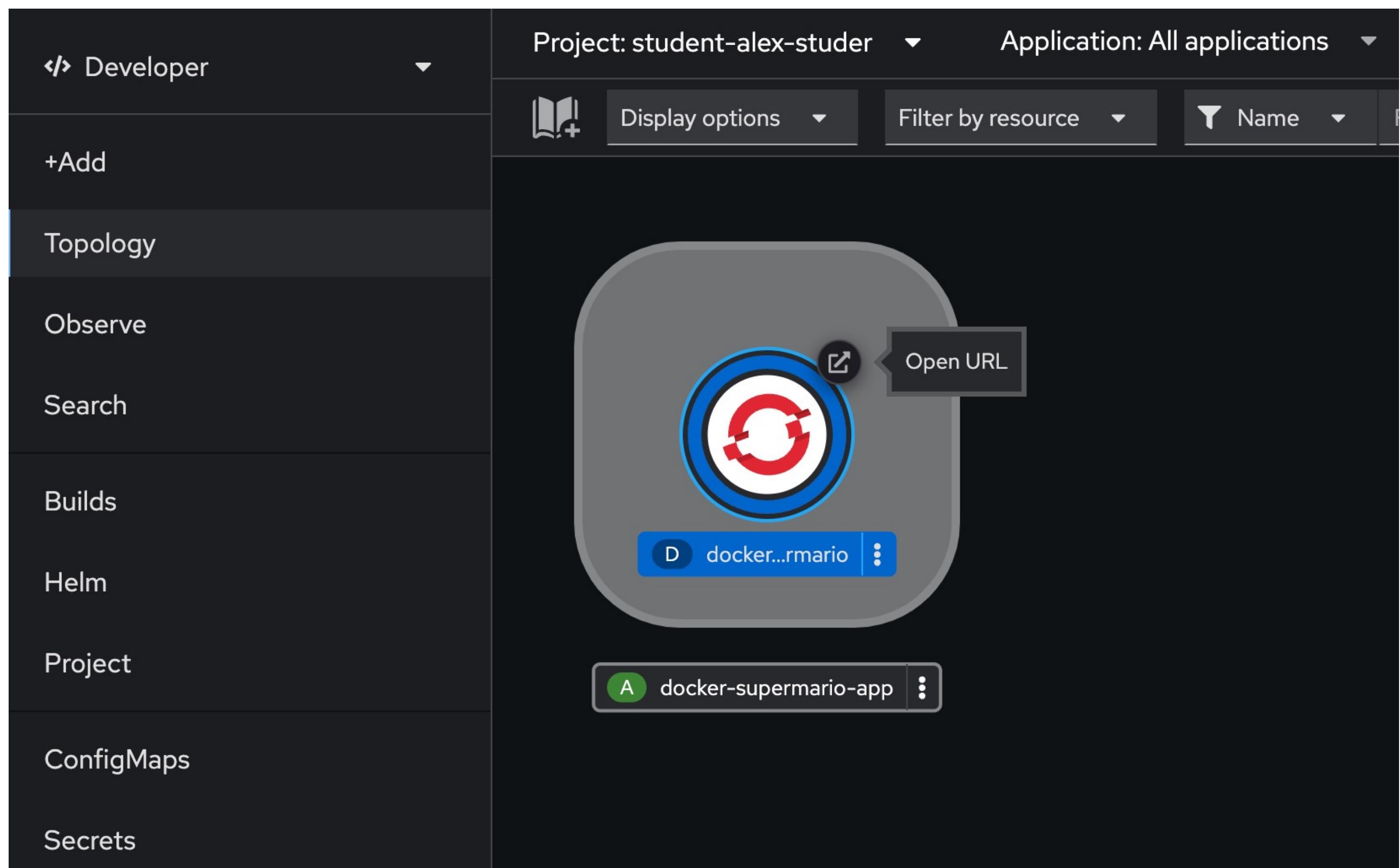
Resource type

Create **Cancel**

Exercise 1

Mario Bros. on the cloud

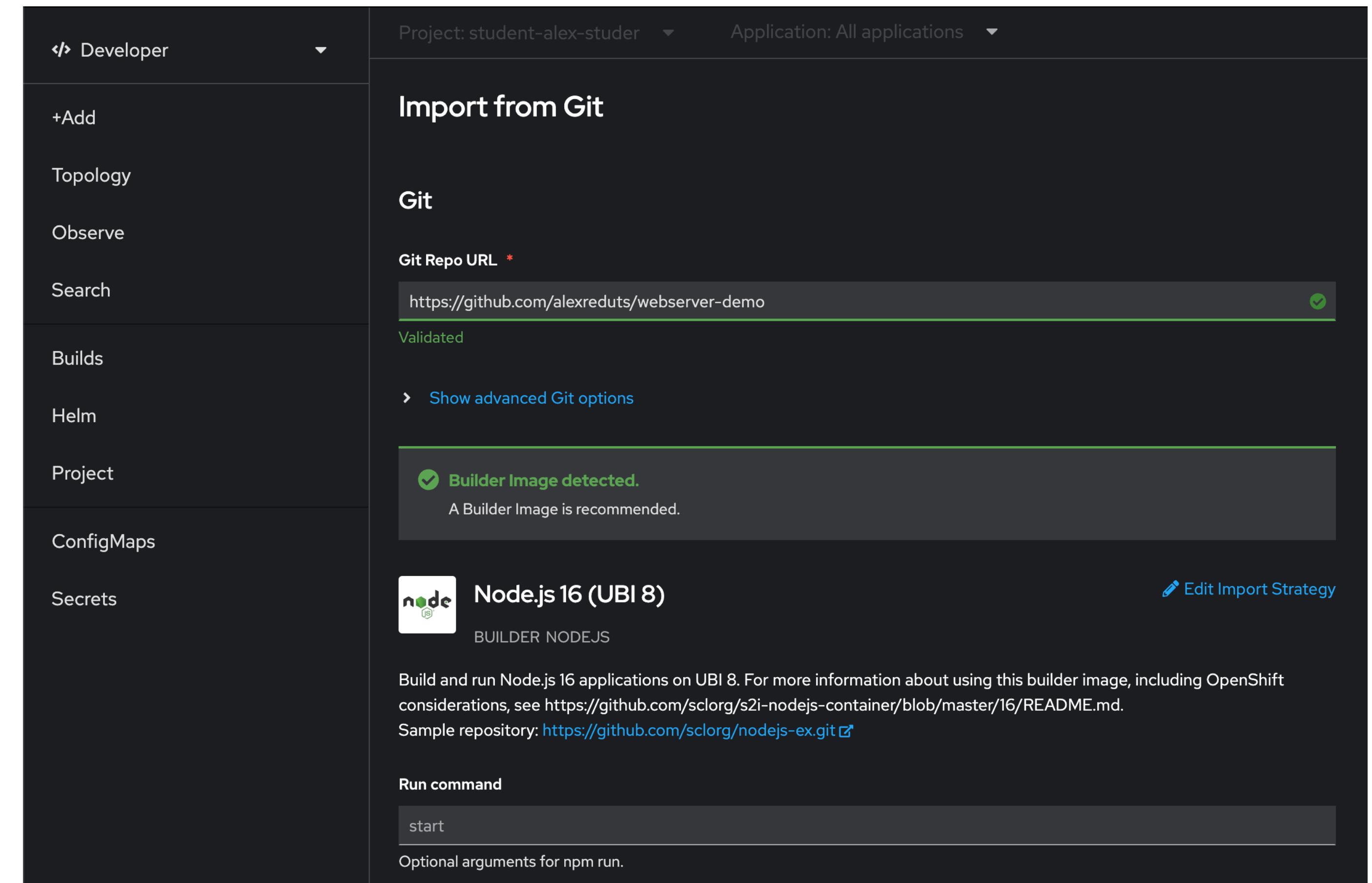
- Sobald der Ring um das OpenShift Logo dunkelblau eingefärbt ist, solltet ihr den «Open URL» Button sehen.
- Wenn ihr diesen drückt, sollte in einem neuen Fester Supermario laufen.
- Öffnet die selbe URL auf eurem Mobiltelefon oder einem andere Gerät.
Habt ihr zugriff darauf?



Exercise 2

Hello World Webserver

- Wählt auf der «+Add» Ansicht «Import from Git» aus.
- Kopiert die folgende Git Repo URL (siehe Bild rechts):
<https://github.com/alexreduts/webserver-demo>



The screenshot shows the Red Hat OpenShift developer interface. On the left, a sidebar menu includes options like +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area is titled "Import from Git" under the "Git" section. It displays a "Git Repo URL" field containing "https://github.com/alexreduts/webserver-demo", which is marked as "Validated". A note indicates "Builder Image detected" with a Node.js 16 (UBI 8) icon. Below this, there's a "Run command" section with "start" listed and an optional argument field. The top navigation bar shows "Project: student-alex-studer" and "Application: All applications".

Exercise 2

Hello World Webserver

- Unter General>>Application wählt im drop-down Menu «Create application» aus. (siehe Bild rechts untern)
- Gebt der neuen Applikation einen eigenen Namen unter «webserver-demo»
- Wählt im drop-down Menü «Resource type» «DeploymentConfig» aus.
- Den Rest lässt ihr unverändert und führt wieder «Create» aus

General

Application

Create application
Select an Application to group this component.

Application name *
webserver-demo
A unique name given to the application grouping to label your resources.

Name *
webserver-demo
A unique name given to the component that will be used to name associated resources.

Resource type
DeploymentConfig
Resource type to generate. The default can be set in [User Preferences](#).

Advanced options

Target port
8080
Target port for traffic.

Create a route
Exposes your component at a public URL

[Show advanced Routing options](#)

General

Application

docker-supermario-app
 Select an application

Create application
No application group

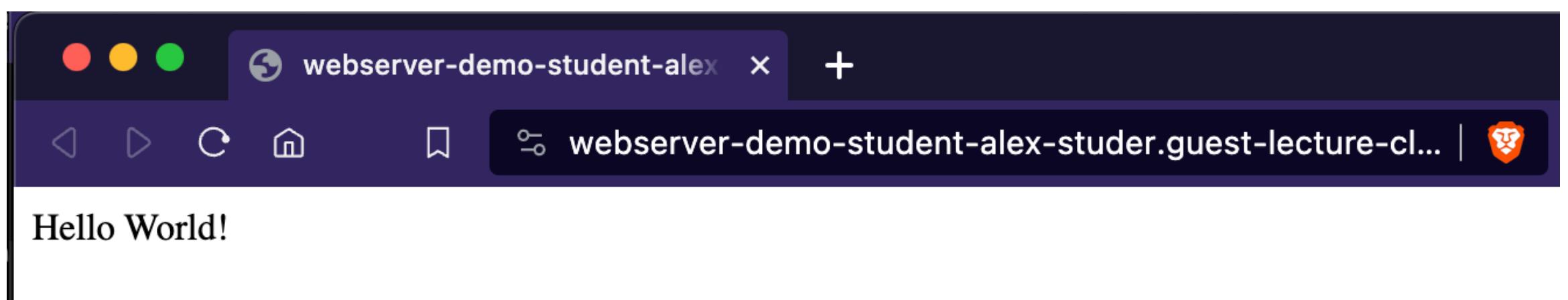
docker-supermario-app

Resource type to generate. The default can be set in [User Preferences](#).

Exercise 2

Hello World Webserver

- Wenn ihr diesmal auf das OpenShift Symbol in der «Topology» Ansicht drückt, solltet ihr eine Ansicht analog dem Bild rechts sehen. Wartet bis unter «Pods» eine Pod ressource existiert, die als «running» gekennzeichnet ist.
- Sobald der Container läuft könnt ihr auf den Link unter «Routes» clicken und ein neuer Tab sollte sich öffnen mit einer simplen Hello World Nachricht.



DC webserver-demo Actions ▾

Details Resources Observe

Pods
No Pods found for this resource.

Builds
BC webserver-demo Start Build
Build #1 is running (Just now) View logs

Services
S webserver-demo Service port: 8080-tcp → Pod port: 8080

Routes
RT webserver-demo Location: <https://webserver-demo-student-alex-studer.guest-lecture-cluster-dev-607d4c962c42edb56975f6e066c57197-0000.eu-de.containers.appdomain.cloud>

Exercise 3

Source-to-Image Workflow

Ziel: Kreiert eine weiter Applikation basierend auf dem «Hello World Webserver» aber mit einem anderen Text.

Hints:

- Geht zum Github Repository vom Webserver und kreiert euren eigenen «Fork» davon
- Ändert im app.js file den Text von Hello World
- Legt eine neue Applikation in OpenShift an das neu euer eigenes Git Repository verwendet.

