

Deploying applications



Alberto Pimpo

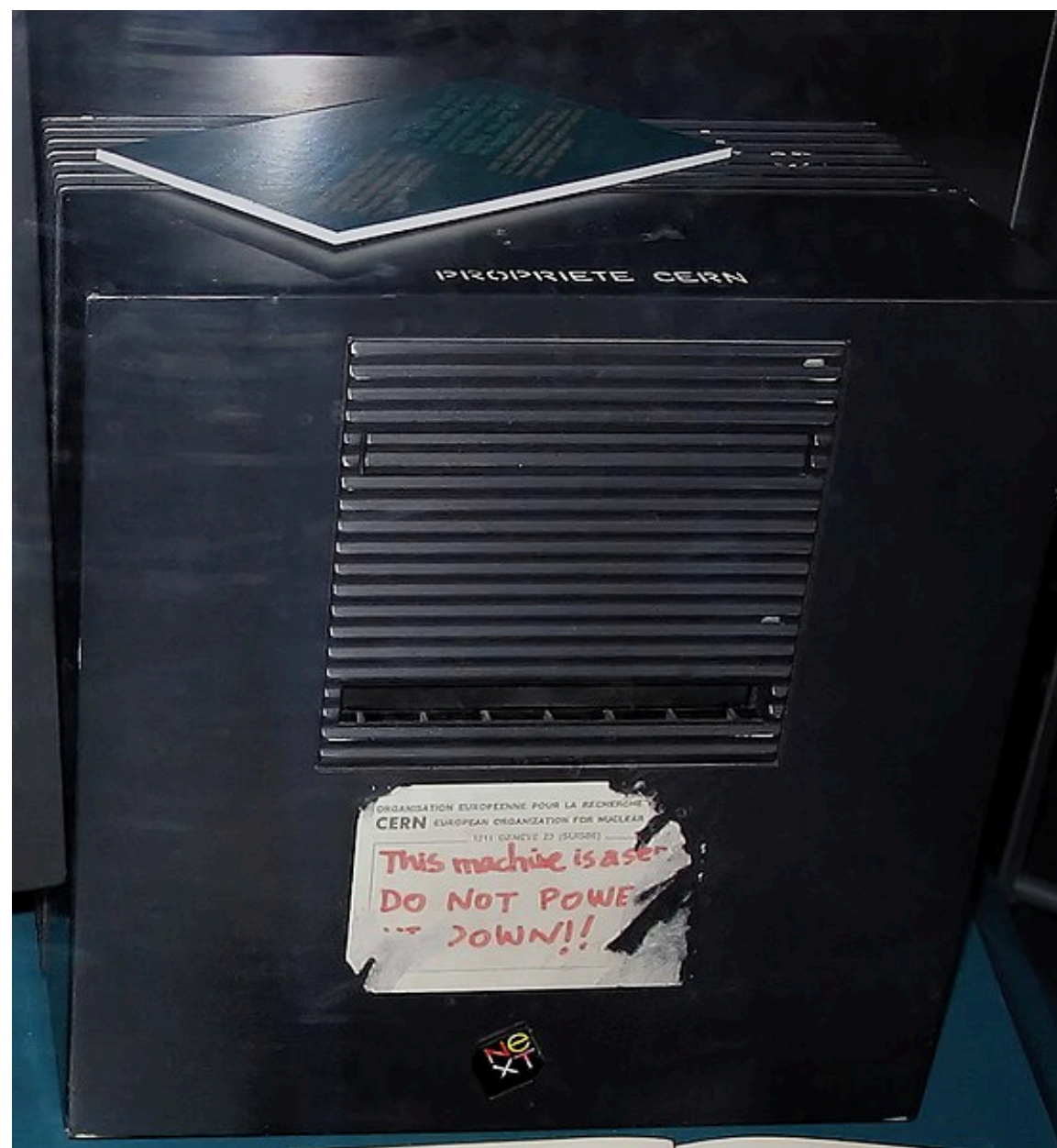
```
[ber@apimpo-laptop]~% whoami
```

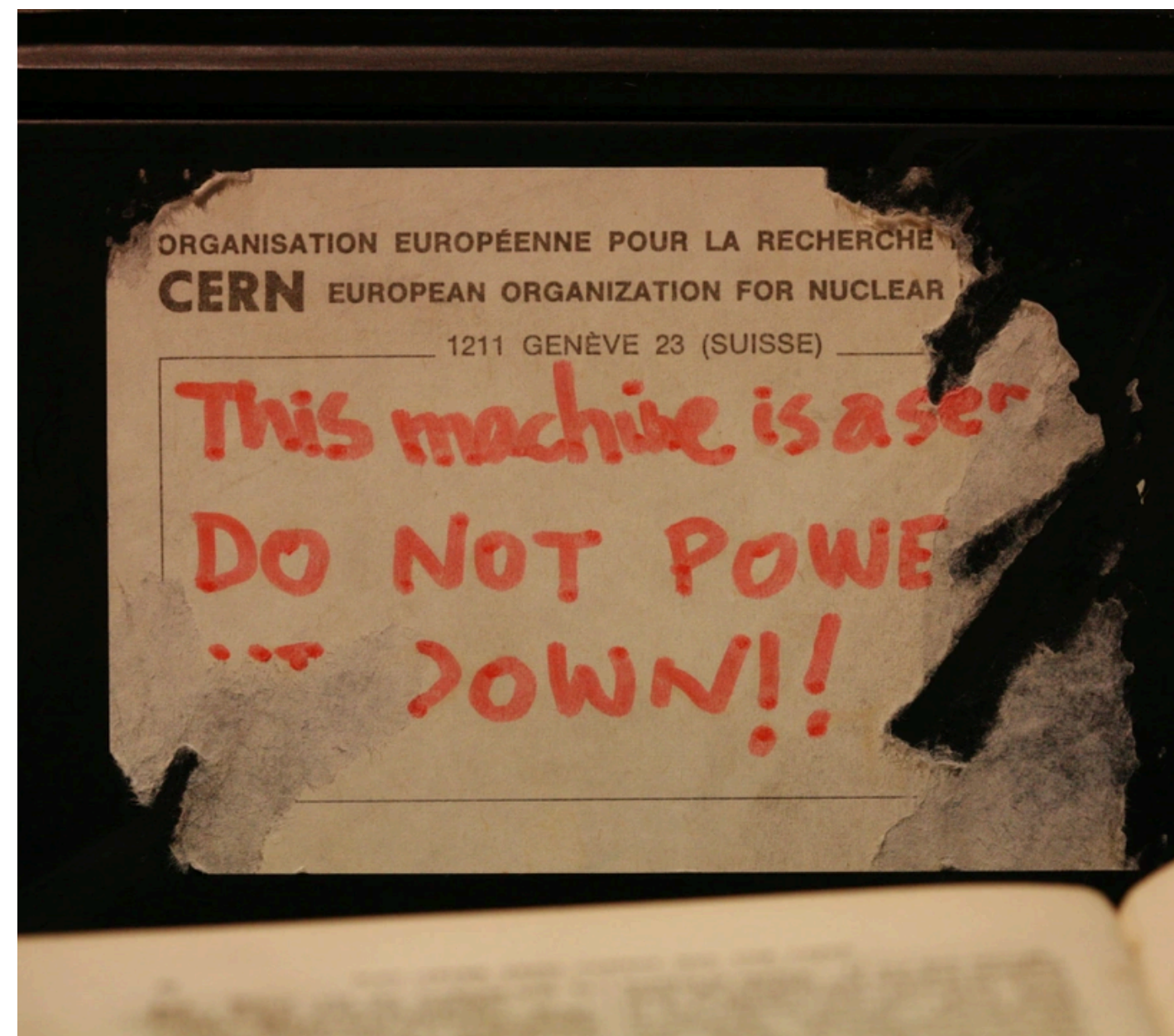
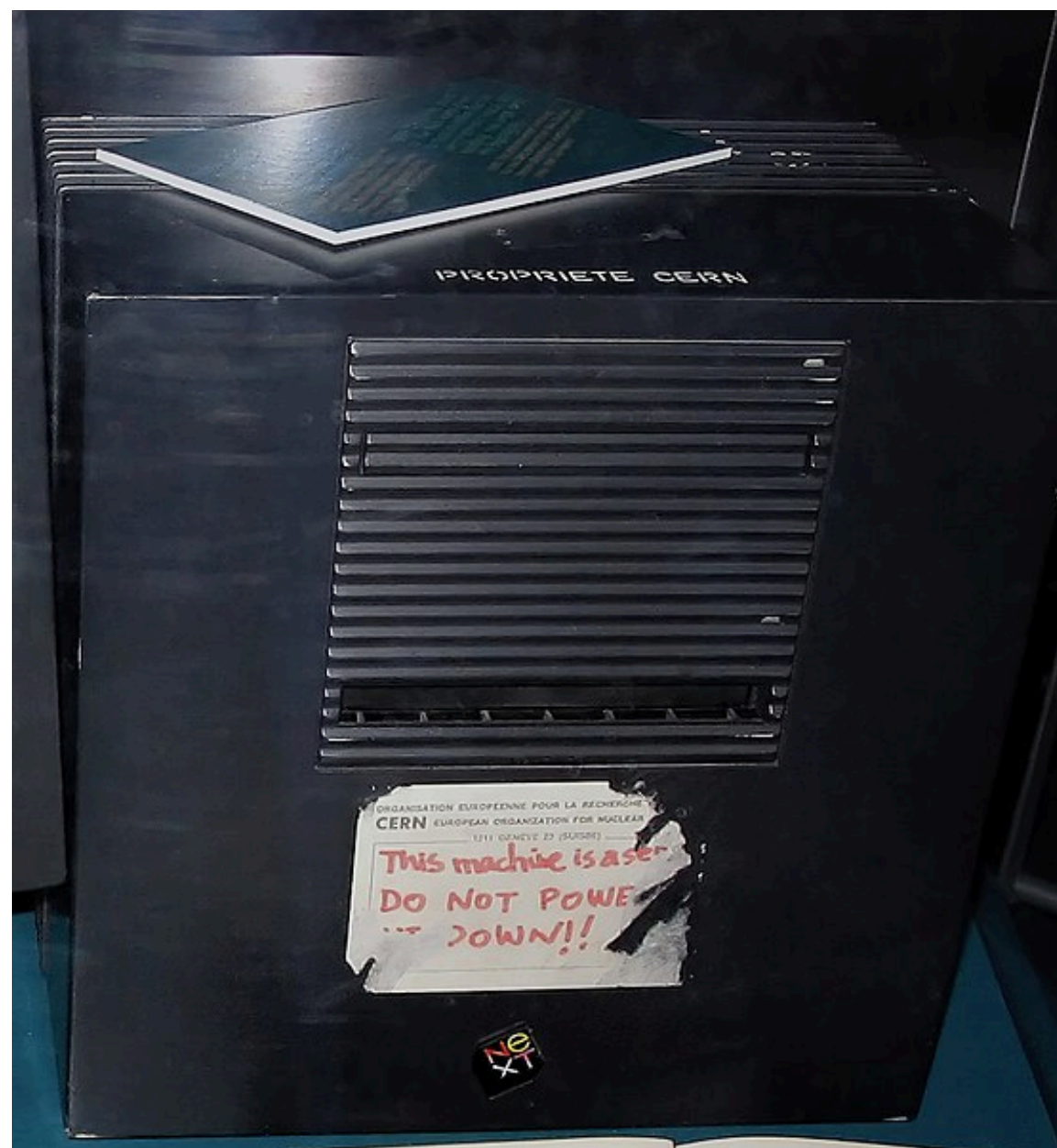
```
Alberto Pimpo
```

```
Dev Ops Engineer (IT-PW-AH)
```

```
Formerly BE-ICS
```

```
[ber@apimpo-laptop]~% █
```





The image features decorative graphic elements in the corners. In the top-right and bottom-left corners, there are several overlapping, rounded rectangular shapes in shades of blue and grey, creating a layered effect. The central text is positioned between these decorative elements.

What can go wrong?



**Deploying correctly is hard,
we need to rely on services.**

Let's make a pizza

Buy all the ingredients



Let's make a pizza

Buy all the ingredients



Buy a premade dough



Let's make a pizza

Buy all the ingredients



Buy a premade dough



Just order it





Paradigms of cloud computing



- **IaaS**
 - virtualized components that replicates the physical architectures (VMs, Block Storage, etc...)
- **PaaS**
 - a platform for building and deploying applications without managing the infrastructure
- **SaaS**
 - just use the software

Some examples

SaaS



PaaS



IaaS



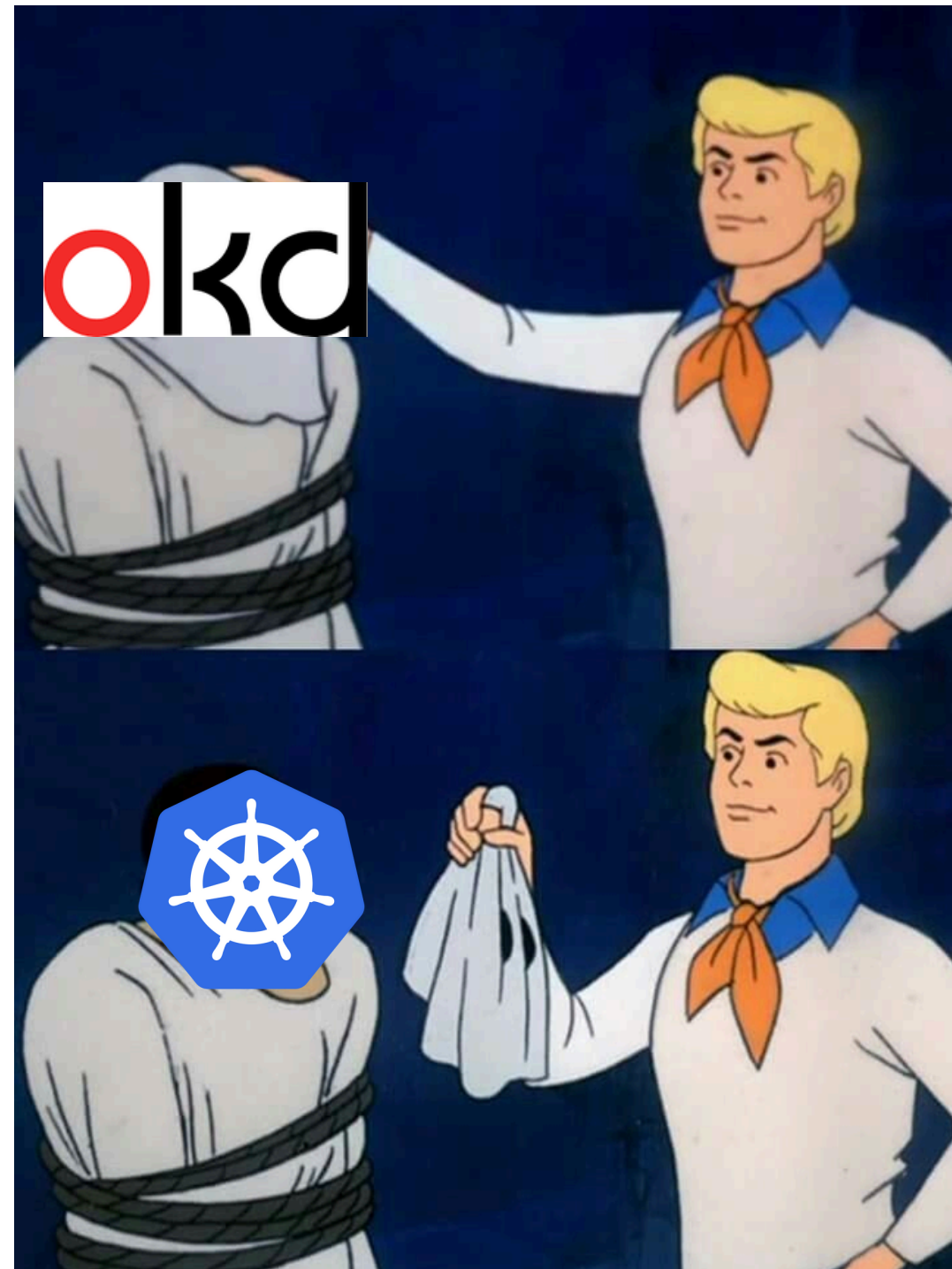
Our PaaS platform

okcd

Let's see together how it works

<https://paas.cern.ch/>

It's Kubernetes under the hood



A package manager for kubernetes



Let's see together an example



Thanks!





Exercise 1



- Create a project on paas-stg (use webservices-portal) with *Test* category
- Deploy the application from *Modern application and deployment* lecture
 - using S2I (Import from git)
 - deploying the container image directly
- Look at the logs of your app on logs.paas.cern.ch



Exercise 2

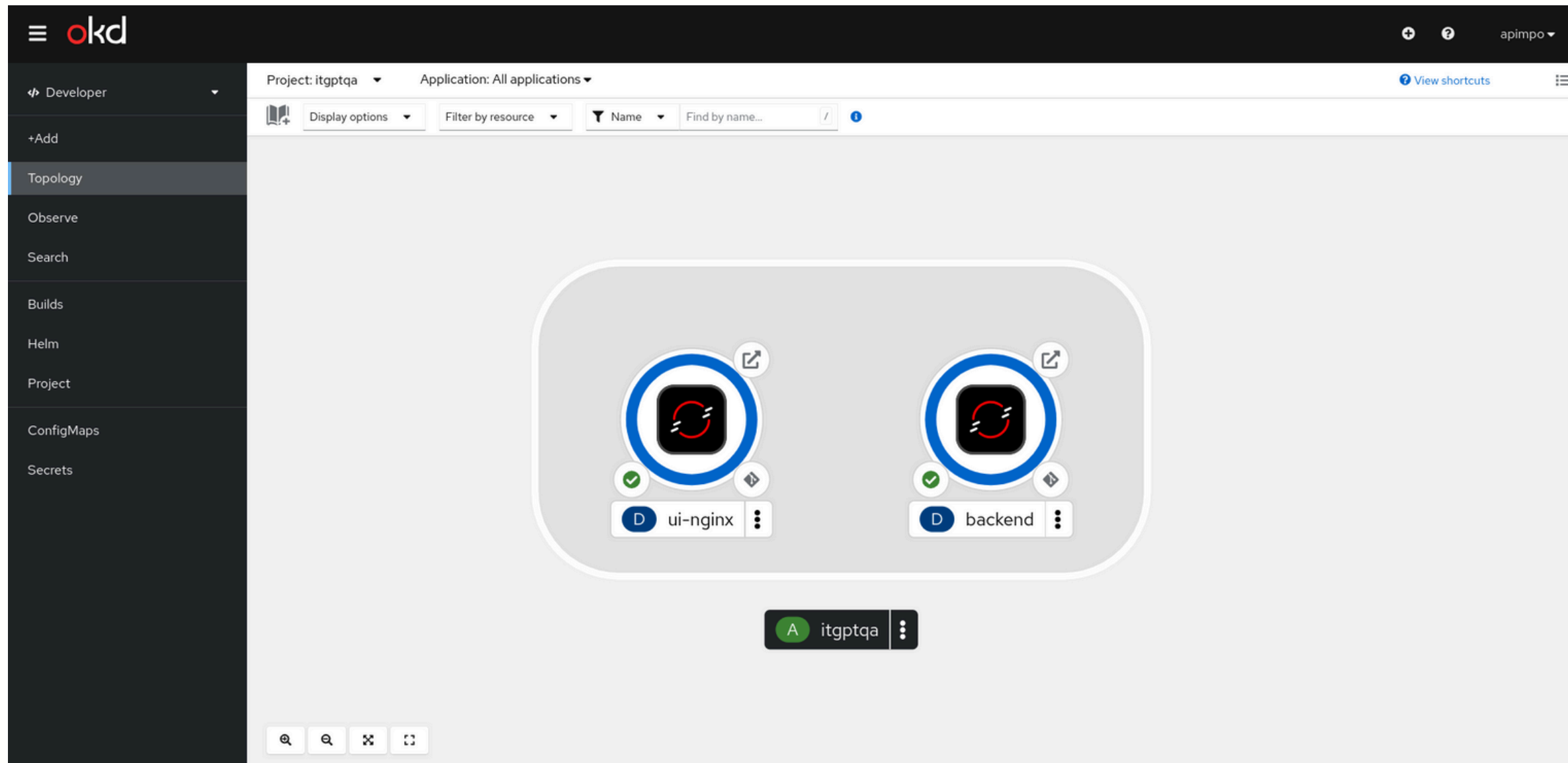


- Deploy openwebui helm chart with a custom values file
 - disable ollama and pipelines components
 - set the hostname to *myname-test.app.cern.ch*
 - *add [1] to the ingress config to make it visible outside CERN*
 - increase the replicas to 3
 - use image *gitlab-registry.cern.ch/apimpo/open-webui/myimage* with tag *latest-x86_64*
- Extra:
 - connect it with a DBoD Postgres instance (extra envvar `DATABASE_URL`)

[1]

```
annotations:  
  haproxy.router.openshift.io/ip_whitelist: ''  
  route.openshift.io/termination: edge  
tls: false
```


Screenshots



Screenshots

The screenshot displays the OpenShift Developer console interface. The top navigation bar includes the 'okd' logo, a user profile 'apimpo', and a 'View shortcuts' link. The main content area shows a deployment overview for the 'itgptqa' project. Two deployment units are visible: 'ui-nginx' and 'backend', both in a 'Running' state. A central application icon 'itgptqa' is also shown. The right-hand sidebar provides detailed information for the 'ui-nginx' deployment, including a health check warning, a table of running pods, and a list of completed builds.

Project: itgptqa **Application:** All applications

Deployment Overview:

- Deployment: **ui-nginx** (Running)
- Deployment: **backend** (Running)
- Application: **itgptqa** (Running)

Health checks: Container ui-nginx does not have health checks to ensure your application is running correctly. [Add health checks](#)

Pods:

Name	Status	View logs
ui-nginx-85c498c686-5rtkr	Running	View logs
ui-nginx-85c498c686-qdqv2	Running	View logs
ui-nginx-85c498c686-jdmwz	Running	View logs

Builds:

Build	Status	View logs
ui-nginx	View all 9	Start Build
Build #52	was complete (180 days ago)	View logs
Build #51	was complete (180 days ago)	View logs
Build #50	was complete (180 days ago)	View logs

Screenshots

The screenshot displays the OpenShift Developer console interface. The top navigation bar includes the 'okd' logo, a user profile 'apimpo', and a 'View shortcuts' link. The main content area is titled 'Project: itgptqa' and 'Application: All applications'. A sidebar on the left lists navigation options: Developer, +Add, Topology (selected), Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The central 'Topology' view shows a deployment graph with two service endpoints: 'ui-nginx' and 'backend', both with a 'D' icon and a status indicator. Below them is the application 'itgptqa' with an 'A' icon. The right-hand panel provides details for the 'ui-nginx' deployment, including a 'Health checks' warning, a '3 Pods' gauge with an 'Increase the Pod count' button, and a list of configuration parameters such as 'Update strategy', 'Max unavailable', 'Max surge', 'Progress deadline seconds', and 'Min ready seconds'.

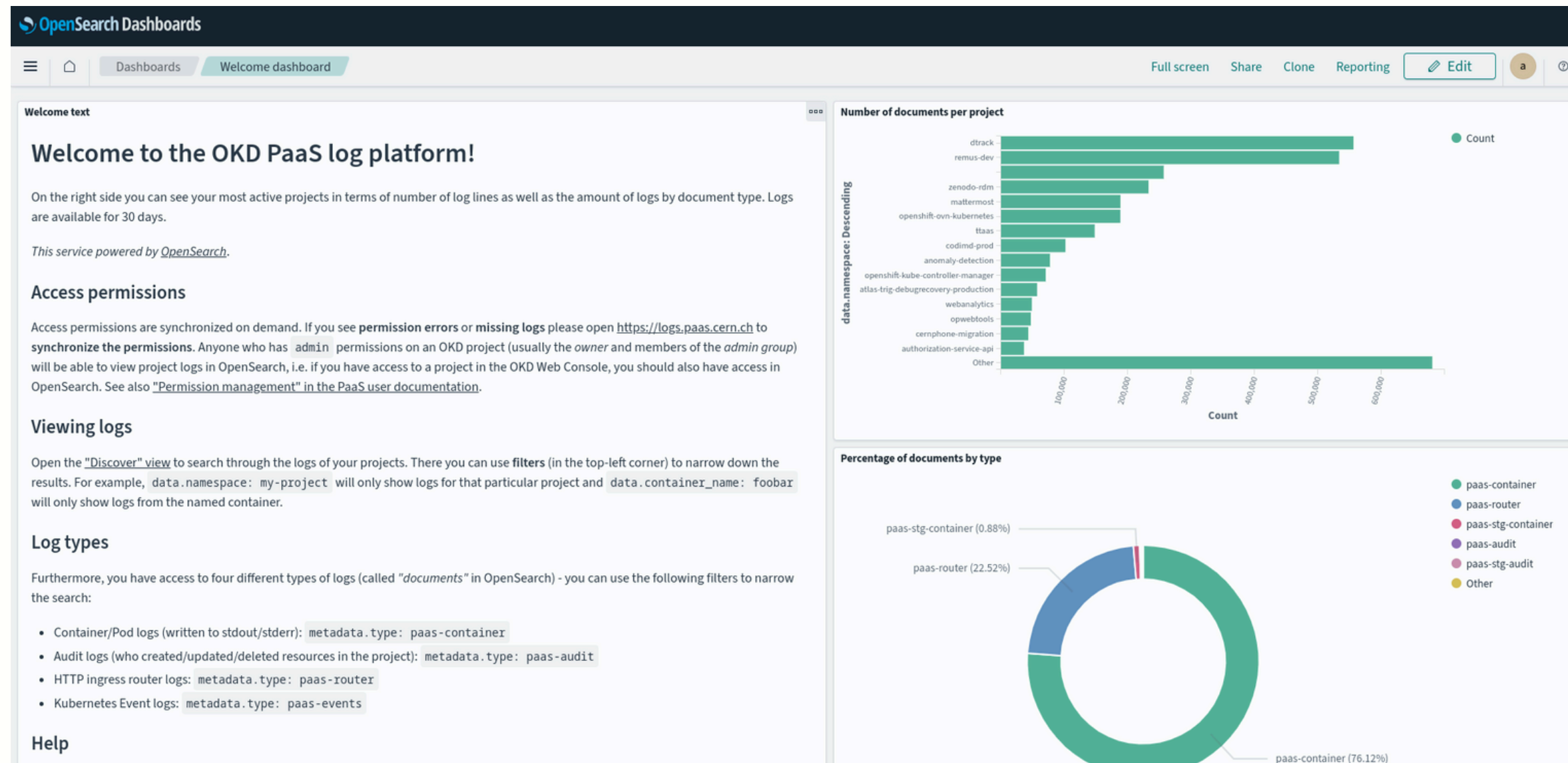
Parameter	Value
Name	ui-nginx
Update strategy	RollingUpdate
Namespace	itgptqa
Max unavailable	25% of 3 pods
Max surge	25% greater than 3 pods
Progress deadline seconds	600 seconds
Min ready seconds	Not configured

Screenshots

The screenshot displays the OKD web console interface. On the left is a dark sidebar with navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main area shows the 'Project: itgptqa' and 'Pod details' for a pod named 'ui-nginx-85c498c686-5rtrk' which is in a 'Running' state. Below this, there are tabs for 'Details', 'Metrics', 'YAML', 'Environment', 'Logs', 'Events', and 'Terminal'. The 'Logs' tab is active, showing a 'Log streaming...' interface with a search bar and controls for 'View log history', 'Show full log', 'Wrap lines', 'Raw', 'Download', and 'Expand'. The log content shows a series of HTTP requests from various clients to the pod, including requests for robots.txt, elmah.axd, .htaccess, and various API endpoints like /chat/api/config and /chat/api/user/conversations. The log entries include timestamps, IP addresses, and user-agent strings.

```
377 10.76.123.2 - - [23/Oct/2024:22:00:20 +0000] "GET /elmah.axd HTTP/1.1" 200 417 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0" "137.138.175.218"
378 10.76.12.2 - - [23/Oct/2024:22:00:35 +0000] "GET /robots.txt HTTP/1.1" 200 417 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0" "137.138.175.218"
379 10.76.12.2 - - [23/Oct/2024:22:00:38 +0000] "GET /elmah.axd HTTP/1.1" 200 417 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0" "137.138.175.218"
380 10.76.12.2 - - [23/Oct/2024:22:00:38 +0000] "GET /.htaccess HTTP/1.1" 200 417 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0" "137.138.175.218"
381 10.76.123.2 - - [25/Oct/2024:08:28:00 +0000] "GET / HTTP/1.1" 200 417 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0" "2001:1458:202:6::101:35e0"
382 10.76.123.2 - - [25/Oct/2024:08:28:00 +0000] "GET /assets/index-4adb2002.js HTTP/1.1" 200 829279 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
383 10.76.123.2 - - [25/Oct/2024:08:28:00 +0000] "GET /assets/index-13a236e8.css HTTP/1.1" 200 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
384 10.76.123.2 - - [25/Oct/2024:08:28:00 +0000] "GET /chat/api/config HTTP/1.1" 200 1003 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
385 10.76.123.2 - - [25/Oct/2024:08:28:00 +0000] "GET /chat/api/user/conversations HTTP/1.1" 401 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
386 10.76.123.2 - - [25/Oct/2024:08:28:09 +0000] "GET /chat/api/config HTTP/1.1" 200 1003 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
387 10.76.123.2 - - [25/Oct/2024:08:28:09 +0000] "GET /icon.png HTTP/1.1" 200 186894 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
388 10.76.123.2 - - [25/Oct/2024:08:28:09 +0000] "GET /chat/api/user/conversations HTTP/1.1" 401 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:132.0) Gecko/20100101 Firefox/132.0"
389 10.76.53.2 - - [29/Oct/2024:07:38:09 +0000] "GET / HTTP/1.1" 200 417 "-" "Iframely/1.3.1 (+https://iframely.com/docs/about) Atlassian" "18.209.183.169"
390 10.76.12.2 - - [30/Oct/2024:22:24:48 +0000] "GET / HTTP/1.1" 200 417 "-" "Mozilla/5.0 Firefox/33.0" "2001:1458:201:afc0::100:25a"
391 10.76.123.2 - - [03/Nov/2024:14:46:43 +0000] "GET / HTTP/1.1" 200 417 "https://paas.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
392 10.76.123.2 - - [03/Nov/2024:14:46:43 +0000] "GET /assets/index-4adb2002.js HTTP/1.1" 200 829279 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
393 10.76.123.2 - - [03/Nov/2024:14:46:43 +0000] "GET /assets/index-13a236e8.css HTTP/1.1" 200 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
394 10.76.123.2 - - [03/Nov/2024:14:46:43 +0000] "GET /chat/api/config HTTP/1.1" 200 1003 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
395 10.76.123.2 - - [03/Nov/2024:14:46:43 +0000] "GET /icon.png HTTP/1.1" 200 186894 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
396 10.76.123.2 - - [03/Nov/2024:14:46:44 +0000] "GET /chat/api/user/conversations HTTP/1.1" 401 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
397 10.76.123.2 - - [03/Nov/2024:14:46:44 +0000] "GET /?state=3c6a19df7d514830a05d5178d91a7de7&session_state=3fa42446-dffd-49fc-acc3-66e97d8402ff&code=5e90e408-e8ca-4e1b-a30c-fae81e69e2de.3" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
398 10.76.123.2 - - [03/Nov/2024:14:46:44 +0000] "GET /chat/api/config HTTP/1.1" 200 1003 "https://itgptqa.app.cern.ch/?state=3c6a19df7d514830a05d5178d91a7de7&session_state=3fa42446-dffd-49fc-acc3-66e97d8402ff&code=5e90e408-e8ca-4e1b-a30c-fae81e69e2de.3" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
399 10.76.123.2 - - [03/Nov/2024:14:46:44 +0000] "GET /chat/api/user/conversations HTTP/1.1" 401 34 "https://itgptqa.app.cern.ch/?state=3c6a19df7d514830a05d5178d91a7de7&session_state=3fa42446-dffd-49fc-acc3-66e97d8402ff&code=5e90e408-e8ca-4e1b-a30c-fae81e69e2de.3" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
400 10.76.123.2 - - [03/Nov/2024:14:46:45 +0000] "GET /chat/api/user/conversations HTTP/1.1" 401 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
401 10.76.123.2 - - [03/Nov/2024:14:46:47 +0000] "GET /chat/api/user/conversations HTTP/1.1" 401 34 "https://itgptqa.app.cern.ch/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
```

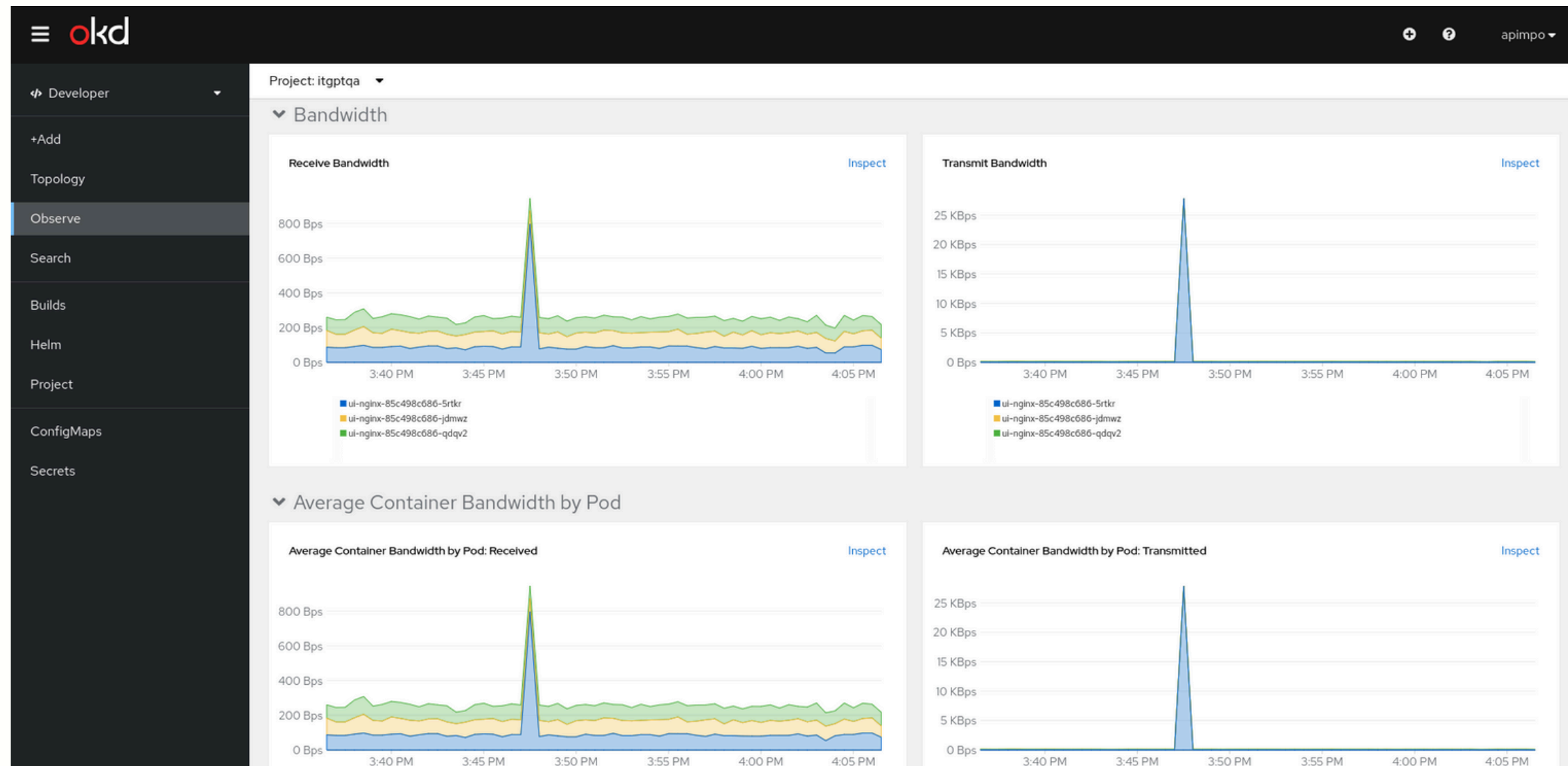
Screenshots



Screenshots

The screenshot displays the OpenShift Kubernetes Dashboard interface. The top navigation bar includes the 'okd' logo, a hamburger menu, and user information 'apimpo'. The main header shows 'Project: itgptqa' and 'Application: All applications'. Below this, there are filters for 'Display options', 'Filter by resource', and a search bar for 'Name'. The central area features a topology view with two containers: 'ui-nginx' and 'backend', both marked with a green checkmark. Below them is the 'itgptqa' application icon. The right sidebar is open to the 'ui-nginx' container details, showing a 'Health checks' warning: 'Container ui-nginx does not have health checks to ensure your application is running correctly. Add health checks'. Below this, the 'Metrics' section displays a 'CPU usage' graph with a peak at 3:45 PM. The legend for the graph includes three entries: 'ui-nginx-85c498c686-5rtkr', 'ui-nginx-85c498c686-jdmwz', and 'ui-nginx-85c498c686-qdqv2'.

Screenshots



Screenshots

The screenshot displays the OKD Observability console interface. The top navigation bar includes the OKD logo, a user profile 'apimpo', and a search icon. The left sidebar contains navigation options: Developer, +Add, Topology, Observe (selected), Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area shows the 'Project: itgptqa' and the 'Observe' section with sub-tabs for Dashboard, Metrics, Alerts, Silences, and Events. A search bar is present with the text 'Search by name...'. Below this is a table of alerts:

Name	Severity	Alert state	Notifications
VolumeUsageAlert	Warning	-	-

Screenshots

The screenshot displays the OpenShift Developer console interface. At the top left, the 'okd' logo is visible. The top right corner shows a user profile 'apimpo' and a help icon. The main interface is divided into a left sidebar and a main content area. The sidebar, under the 'Developer' perspective, includes options like '+Add', 'Topology', 'Observe', 'Search', 'Builds', 'Helm', 'Project', 'ConfigMaps', and 'Secrets'. The main content area is titled 'Project: test-ber' and features three top-level cards: 'Create applications using samples', 'Build with guided documentation', and 'Explore new developer features'. Below these are several categorized cards: 'Developer Catalog' (with sub-sections for All services, Database, Operator Backed, and Helm Chart), 'Git Repository' (with 'Import from Git'), 'Container images', 'Sharing', 'Samples', and 'From Local Machine' (with 'Import YAML' and 'Upload JAR file'). A URL bar at the bottom left shows 'https://paas.cern.ch/import/ns/test-ber'.

Screenshots

The screenshot displays the OpenShift Developer console interface. On the left is a dark sidebar with navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area shows the configuration for a build. At the top, it indicates 'Project: test-ber' and 'Application: All applications'. A 'Hide advanced Git options' toggle is visible. The 'Git reference' field is empty, with a note that it is optional. The 'Context dir' field is set to '/realworld'. The 'Source Secret' dropdown is set to 'Select Secret name'. A green notification banner states 'Builder Image detected. A Builder Image is recommended.' Below this, the selected builder image is 'Python 3.11 (UBI 9)' with the tag 'BUILDER PYTHON' and an 'Edit Import Strategy' link. A descriptive paragraph explains that this builder image is used for Python 3.11 applications on UBI 9, with a link to a sample repository. The 'General' section is partially visible, showing the 'Application' field. At the bottom, there are 'Create' and 'Cancel' buttons.

Screenshots

The screenshot shows the 'Import from Git' dialog in the OpenShift Developer console. The interface includes a sidebar with navigation options like 'Developer', '+Add', 'Topology', 'Observe', 'Search', 'Builds', 'Helm', 'Project', 'ConfigMaps', and 'Secrets'. The main content area is titled 'Import from Git' and shows the 'Git' section with a 'Git Repo URL' field containing 'https://github.com/nickjj/docker-django-example'. Below this, a 'Validated' status is shown, followed by a link to 'Show advanced Git options'. A green notification box indicates 'Multiple import strategies detected' and recommends the Dockerfile at Dockerfile. The 'Dockerfile' section includes a logo and a link to 'Edit Import Strategy'. The 'General' section has an 'Application name' field with 'docker-django-example-app' and a 'Name' field with 'docker-django-examle'. At the bottom, there are 'Create' and 'Cancel' buttons.

Project: test-ber Application: All applications

Import from Git

Git

Git Repo URL *

 ✓

Validated

[Show advanced Git options](#)

✓ Multiple import strategies detected
The Dockerfile at Dockerfile is recommended.

Dockerfile [Edit Import Strategy](#)

General

Application name

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

Name *

[Create](#) [Cancel](#)

Screenshots

The screenshot shows the OpenShift Developer console interface. On the left is a dark sidebar with the 'okd' logo and a navigation menu including 'Developer', '+Add', 'Topology', 'Observe', 'Search', 'Builds', 'Helm', 'Project', 'ConfigMaps', and 'Secrets'. The main area is titled 'Project: test-ber' and displays the 'Add Storage' dialog for a service named 'csc-development-exercise-2'. The dialog has two radio buttons: 'Use existing claim' (selected) and 'Create new claim'. Below the 'Use existing claim' option is a dropdown menu labeled 'Select claim'. The 'Mount path' field is empty, with a note below it stating 'Mount path for the volume inside the container.' There is also an unchecked checkbox for 'Mount as read-only'. The 'Subpath' field is also empty, with a note below it stating 'Optional path within the volume from which it will be mounted into the container. Defaults to the root of the volume.' At the bottom, there is a note: 'The volume will be mounted into all containers. You can [select specific containers](#) instead.' and two buttons: 'Save' and 'Cancel'.

Project: test-ber

Add Storage to [csc-development-exercise-2](#)

PersistentVolumeClaim *

Use existing claim

Select claim

Create new claim

Mount path *

Mount path for the volume inside the container.

Mount as read-only

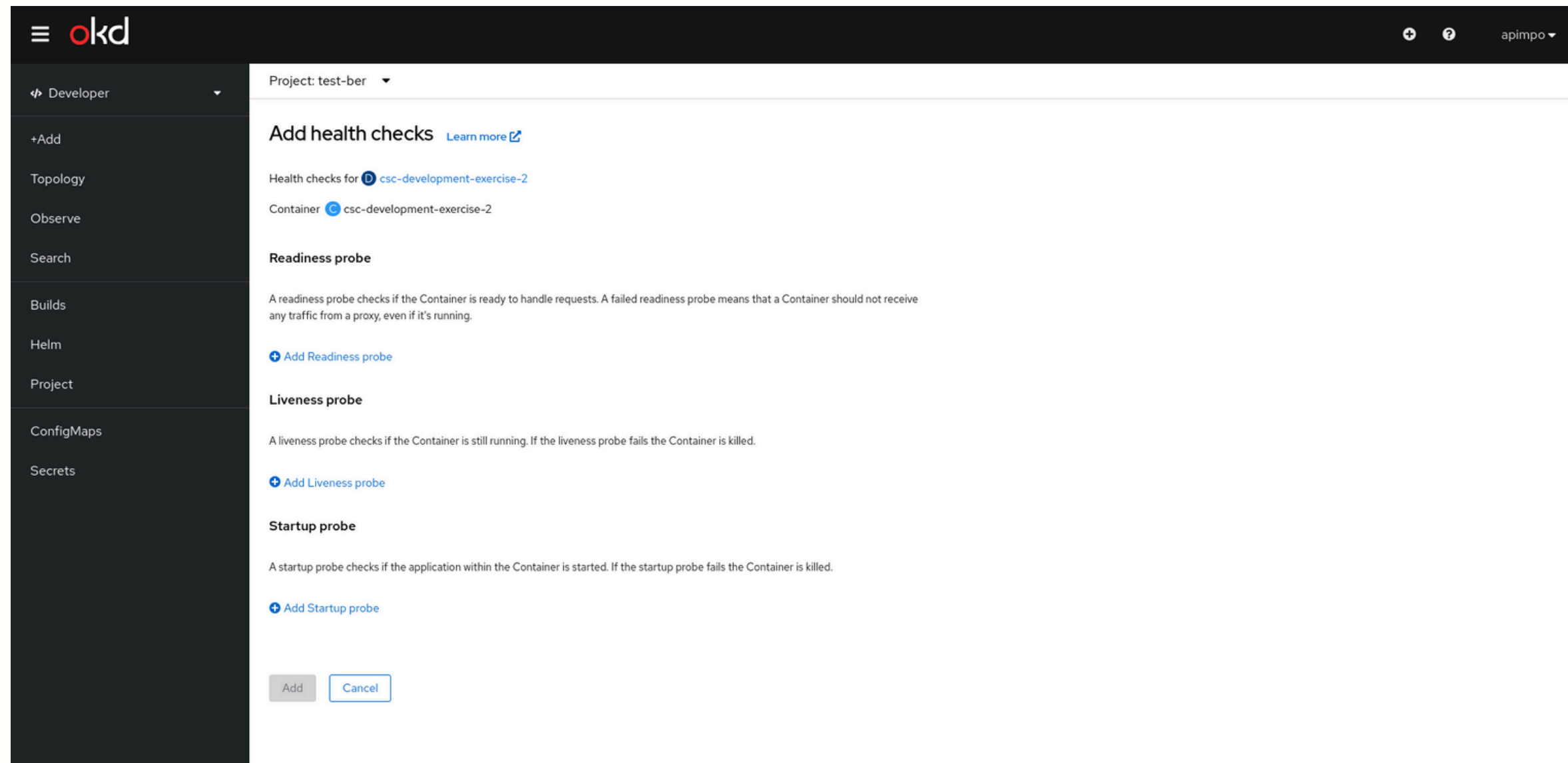
Subpath

Optional path within the volume from which it will be mounted into the container. Defaults to the root of the volume.

The volume will be mounted into all containers. You can [select specific containers](#) instead.

Save Cancel

Screenshots



The screenshot shows the OpenShift console interface. On the left is a dark sidebar with navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area is titled 'Project: test-ber' and displays the 'Add health checks' dialog for the container 'csc-development-exercise-2'. The dialog includes sections for 'Readiness probe', 'Liveness probe', and 'Startup probe', each with a brief description and an 'Add' button. At the bottom of the dialog are 'Add' and 'Cancel' buttons.

okd + ? apimpo ▾

Project: test-ber ▾

Add health checks [Learn more](#)

Health checks for **csc-development-exercise-2**

Container **csc-development-exercise-2**

Readiness probe

A readiness probe checks if the Container is ready to handle requests. A failed readiness probe means that a Container should not receive any traffic from a proxy, even if it's running.

[+ Add Readiness probe](#)

Liveness probe

A liveness probe checks if the Container is still running. If the liveness probe fails the Container is killed.

[+ Add Liveness probe](#)

Startup probe

A startup probe checks if the application within the Container is started. If the startup probe fails the Container is killed.

[+ Add Startup probe](#)

Screenshots

The screenshot displays the OpenShift Developer console interface for creating a new claim. The top navigation bar shows the 'okd' logo, a search icon, a help icon, and the user 'apimpo'. The left sidebar contains navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area is titled 'Project: test-ber' and shows the 'Create new claim' form. The 'StorageClass' dropdown is set to 'cephfs'. Below the dropdown is a search box labeled 'Select StorageClass'. A list of available StorageClasses is shown, including 'cephfs', 'cephfs-no-backup', 'cephfs-ssd', 'cephfs-ssd-no-backup', 'cvmfs', and 'eos'. The 'Filesystem' radio button is selected. Below the list are input fields for 'Mount path' and 'Subpath', with descriptive text for each.

Project: test-ber

Create new claim

StorageClass

cephfs

Select StorageClass

- cephfs (default) | Standard CephFS volume | cephfs.manila.csi.openstack.org | Meyrin CephFS
- cephfs-no-backup CephFS persistent volume without automatic backup. It is not possible to recover a previous state of the volume. Appropriate for non-critical, frequently-changing or rolling data. | cephfs.manila.csi.openstack.org | Meyrin CephFS
- cephfs-ssd CephFS SSD volume, in a different failure domain than the standard CephFS volume. | cephfs.manila.csi.openstack.org | Meyrin CephFS Vault SSD A
- cephfs-ssd-no-backup CephFS persistent volume on fast SSD storage, without automatic backup. It is not possible to recover a previous state of the volume. Appropriate for non-critical, frequently-changing or rolling data with high performance requirements. | cephfs.manila.csi.openstack.org
- cvmfs cvmfs.csi.cern.ch
- eos Allows accessing the EOS filesystem. Storage size is irrelevant for EOS, use '1Gi'. | eosxd.csi.cern.ch

Filesystem Block

Mount path *

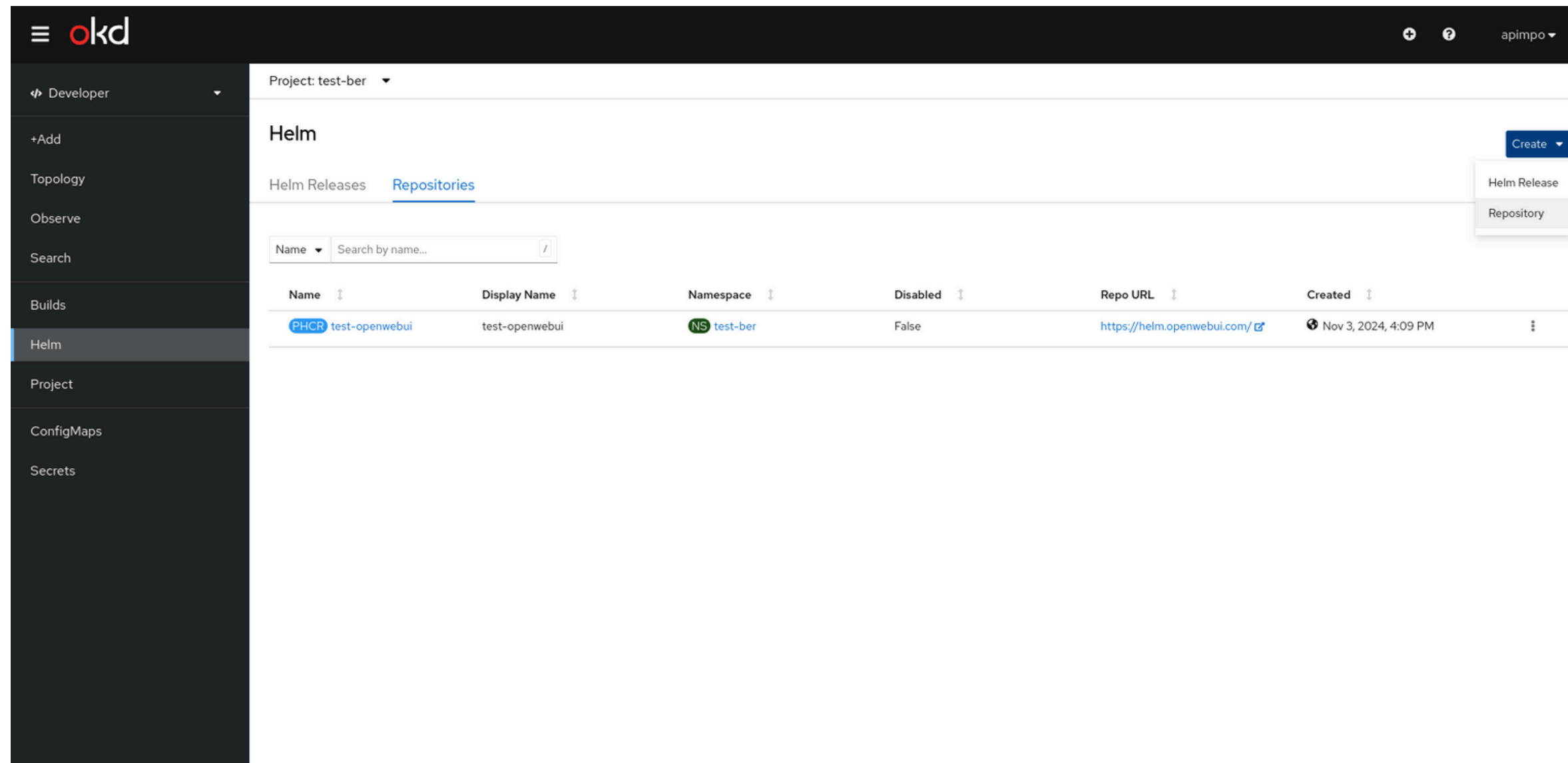
Mount path for the volume inside the container.

Mount as read-only

Subpath

Optional path within the volume from which it will be mounted into the container. Defaults to the root of the volume.

Screenshots



The screenshot shows the OKD web console interface. The top navigation bar includes the OKD logo, a user profile 'apimpo', and a search icon. The left sidebar contains a navigation menu with options: Developer, +Add, Topology, Observe, Search, Builds, Helm (selected), Project, ConfigMaps, and Secrets. The main content area is titled 'Helm' and shows the 'Repositories' tab. A 'Create' button is visible in the top right. Below the header, there is a search bar and a table of Helm repositories. The table has columns for Name, Display Name, Namespace, Disabled, Repo URL, and Created. One repository is listed: 'test-openwebui' in the 'test-ber' namespace, with a URL of 'https://helm.openwebui.com/' and a creation time of 'Nov 3, 2024, 4:09 PM'.

Project: test-ber

Helm

Helm Releases [Repositories](#)

Create

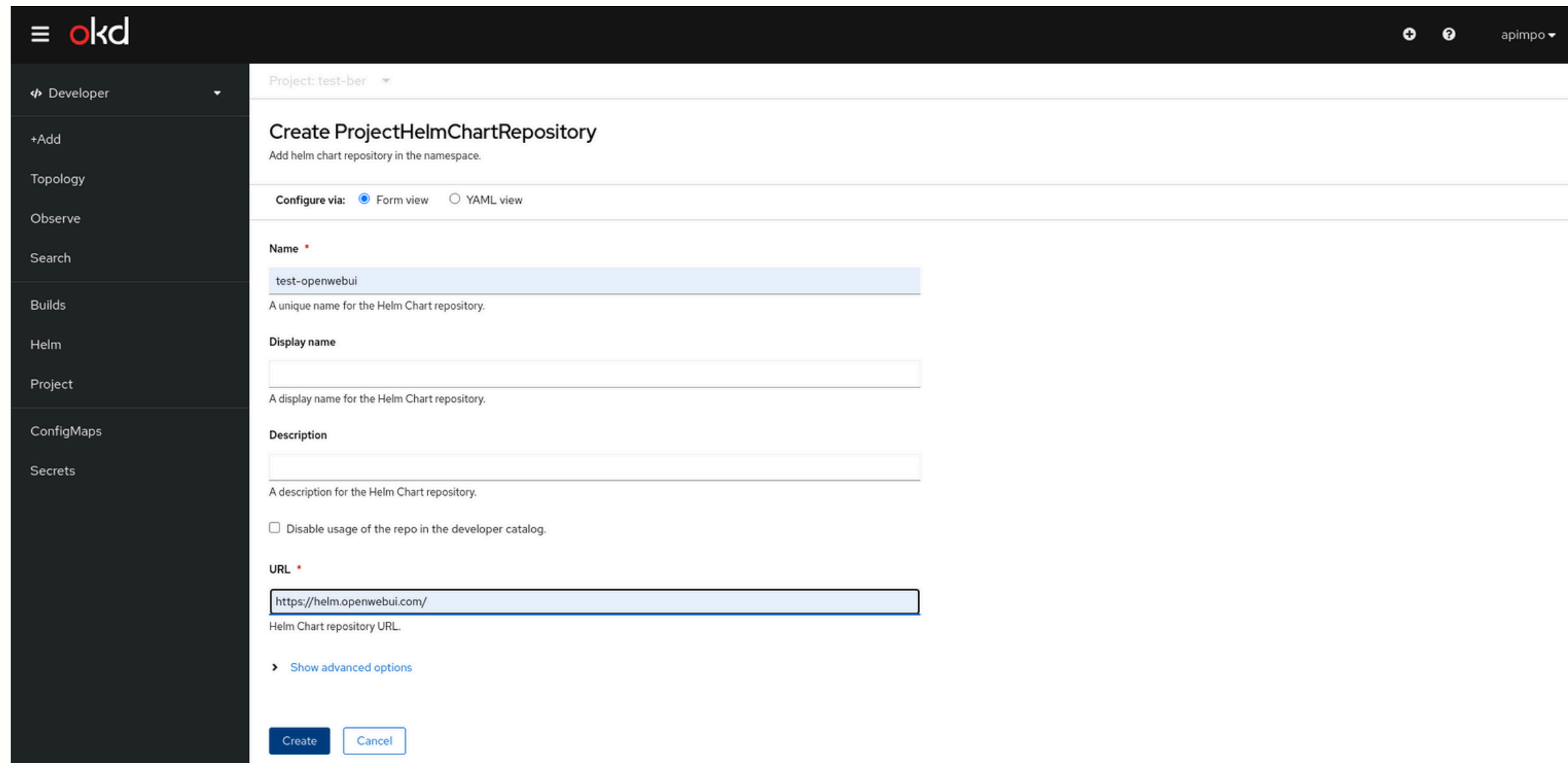
Helm Release

Repository

Name Search by name... /

Name	Display Name	Namespace	Disabled	Repo URL	Created
PHCR test-openwebui	test-openwebui	NS test-ber	False	https://helm.openwebui.com/	Nov 3, 2024, 4:09 PM

Screenshots



The screenshot shows the OpenShift console interface for creating a Helm Chart repository. The top navigation bar includes the 'okd' logo, a search icon, a help icon, and the user 'apimpo'. The left sidebar shows the 'Developer' menu with options like '+Add', 'Topology', 'Observe', 'Search', 'Builds', 'Helm', 'Project', 'ConfigMaps', and 'Secrets'. The main content area is titled 'Create ProjectHelmChartRepository' and includes a sub-header 'Add helm chart repository in the namespace.' Below this, there are radio buttons for 'Form view' (selected) and 'YAML view'. The form contains several fields: 'Name' with the value 'test-openwebui', 'Display name' (empty), 'Description' (empty), and 'URL' with the value 'https://helm.openwebui.com/'. There is also a checkbox for 'Disable usage of the repo in the developer catalog.' and a 'Show advanced options' link. At the bottom, there are 'Create' and 'Cancel' buttons.

Project: test-ber

Create ProjectHelmChartRepository

Add helm chart repository in the namespace.

Configure via: Form view YAML view

Name *

test-openwebui

A unique name for the Helm Chart repository.

Display name

A display name for the Helm Chart repository.

Description

A description for the Helm Chart repository.

Disable usage of the repo in the developer catalog.

URL *

https://helm.openwebui.com/

Helm Chart repository URL.

[Show advanced options](#)

Screenshots

The screenshot shows the OKD Developer Catalog interface. The top navigation bar includes the OKD logo, a hamburger menu, and user information (apimpo). The left sidebar contains navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area is titled 'Helm Charts' and shows a list of three chart repositories: 'Cern Auth Proxy', 'Open Webui', and 'Pipelines'. A search bar and a sort dropdown are visible above the chart cards. The URL at the bottom of the screenshot is: <https://paas.cern.ch/catalog/ns/test-ber?catalogType=HelmChart&selectedId=test-openwebui--https%3A%2F%2Fhelm.openwebui.com%2Fopen-webui-3.4.3.tgz>

Project: test-ber

Developer Catalog > Helm Charts

Helm Charts

Browse for charts that help manage complex installations and upgrades. Cluster administrators can customize the content made available in the catalog. Alternatively, developers can [try to configure their own custom Helm Chart repository](#).

All items 3 items

Filter by keyword... A-Z

Chart Repositories

- OKD4 PaaS Helm Charts (1)
- test-openwebui (2)

Cern Auth Proxy
Helm chart for installing an SSO Proxy on CERN PaaS

Open Webui
Open WebUI: A User-Friendly Web Interface for Chat Interactions 🧡

Pipelines
Pipelines: UI-Agnostic OpenAI API Plugin Framework

<https://paas.cern.ch/catalog/ns/test-ber?catalogType=HelmChart&selectedId=test-openwebui--https%3A%2F%2Fhelm.openwebui.com%2Fopen-webui-3.4.3.tgz>

Screenshots

The screenshot displays the OKD Helm Charts interface. The left sidebar contains navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area shows the 'Helm Charts' section for the 'test-ber' project. It includes a search bar and a list of chart repositories. The 'Open Webui' chart is selected, and its details are shown in a modal window on the right.

Open Webui

Create

Latest Chart version
3.4.3

Product version
0.3.35

Source
N/A

Provider
N/A

Home page
<https://www.openwebui.com/>

Repository
test-openwebui

Maintainers
N/A

Created at
Oct 27, 2024, 4:54 AM

Support
N/A

Description
Open WebUI: A User-Friendly Web Interface for Chat Interactions 🌟

README
open-webui
Open WebUI: A User-Friendly Web Interface for Chat Interactions 🌟
Homepage: <https://www.openwebui.com/>

Source Code

- <https://github.com/open-webui/helm-charts>
- <https://github.com/open-webui/open-webui/pkgs/container/open-webui>
- <https://github.com/otwld/ollama-helm/>
- <https://hub.docker.com/r/ollama/ollama>

Installing
Before you can install, you need to add the `open-webui` repo to `Helm`

```
helm repo add open-webui https://helm.openwebui.com/  
helm repo update
```

Now you can install the chart:

Screenshots

The screenshot displays the OpenShift Developer console interface. The top navigation bar includes the 'okd' logo, a search icon, a help icon, and the user name 'apimpo'. The left sidebar contains navigation options: Developer, +Add, Topology, Observe, Search, Builds, Helm, Project, ConfigMaps, and Secrets. The main content area is titled 'Create Helm Release' and includes a sub-header: 'The Helm Release can be created by manually entering YAML or JSON definitions. For more information on the chart, refer to this [README](#)'. Below this, there are two input fields: 'Release name' with the value 'open-webui' and 'Chart version' with the value '3.4.3 / App Version 0.3.35 (Provided by Test Openwebui)'. A note below the 'Release name' field states: 'A unique name for the Helm Release.' Below the input fields, there are radio buttons for 'Configure via': 'Form view' (unselected) and 'YAML view' (selected). A notification banner at the top of the code editor area reads: 'Form view is disabled for this chart because the schema is not available'. The code editor displays the following YAML configuration:

```
1 affinity: {}
2 annotations: {}
3 clusterDomain: cluster.local
4 containerSecurityContext: {}
5 extraEnvVars:
6   - name: OPENAI_API_KEY
7     value: 0p3n-w3bu!
8 image:
9   pullPolicy: IfNotPresent
10  repository: ghcr.io/open-webui/open-webui
11  tag: ''
12 imagePullSecrets: []
13 ingress:
14   annotations: {}
15   class: ''
16   enabled: false
17   existingSecret: ''
18   host: ''
19   tls: false
```

At the bottom of the form, there are two buttons: 'Create' and 'Cancel'.