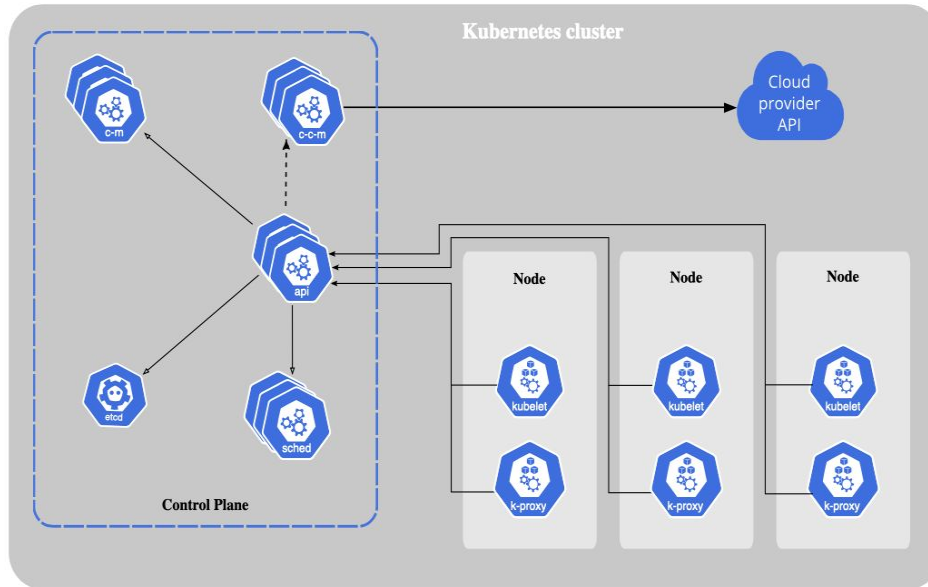




Cloud Computing Project : Kubernetes

Group 3 : Shuang Chen, Samuel D'Apréa, Lukas Atkinson, Louis-César Pagès.

Architecture and Concepts of Kubernetes



- Deployment
- StatefulSet
- Service
- Ingress



Kubernetes setup

- Minikube
- Kubectl
- Interacting with the cluster

```
kubectl get pods -A  
minikube dashboard
```

```
kubectl create deployment hello-minikube --image=k8s.gcr.io/echoserver:1.4  
kubectl expose deployment hello-minikube --type=NodePort --port=8080
```

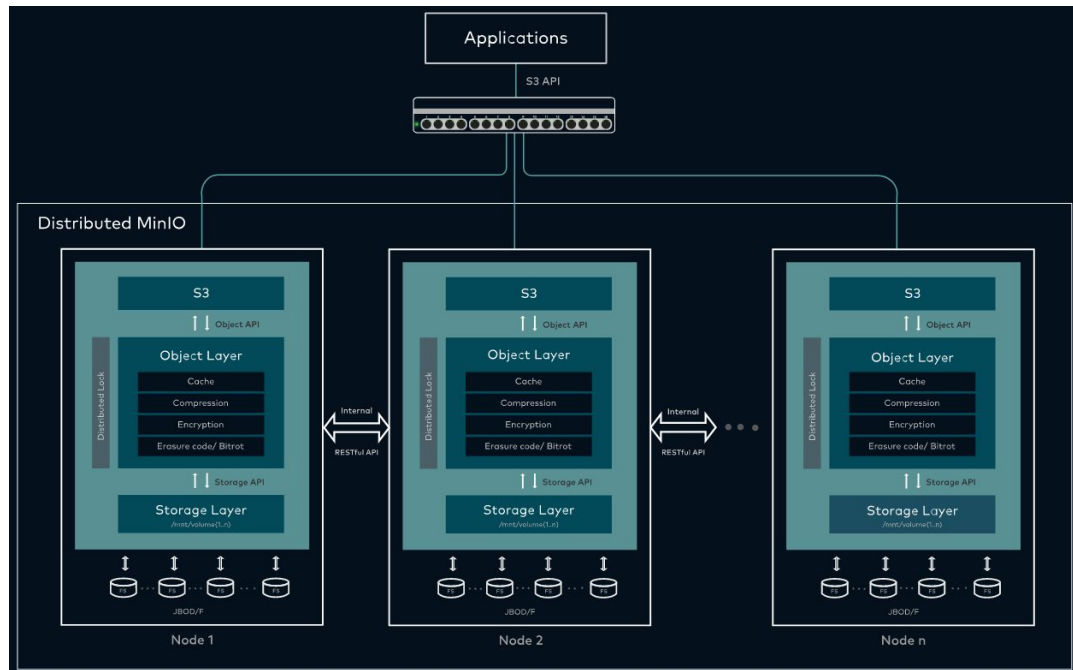
```
kubectl get services hello-minikube  
minikube service hello-minikube  
kubectl port-forward service/hello-minikube 7080:8080
```

Features and Architecture of MinIO

- **Open-source**
- High-performance **object storage**
- Max. supported object size of **5TB**
- **S3-compatible API**
- **Scalable and lightweight**
- Compression
- Server-side **encryption** (e.g. AES-256-GCM)
- **Bitrot** protection / **Erasure coding**
- **Client SDK: Java and Python**

➔ Example demo for using MinIO with Python Client API

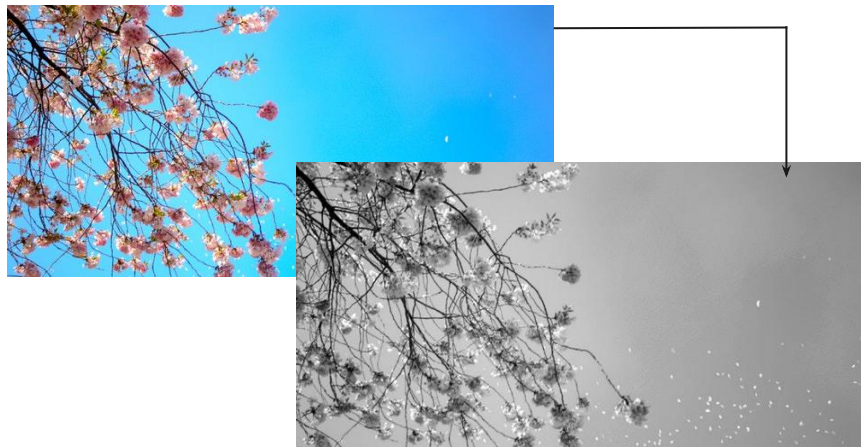
MINIO



<https://min.io/product/overview#architecture>

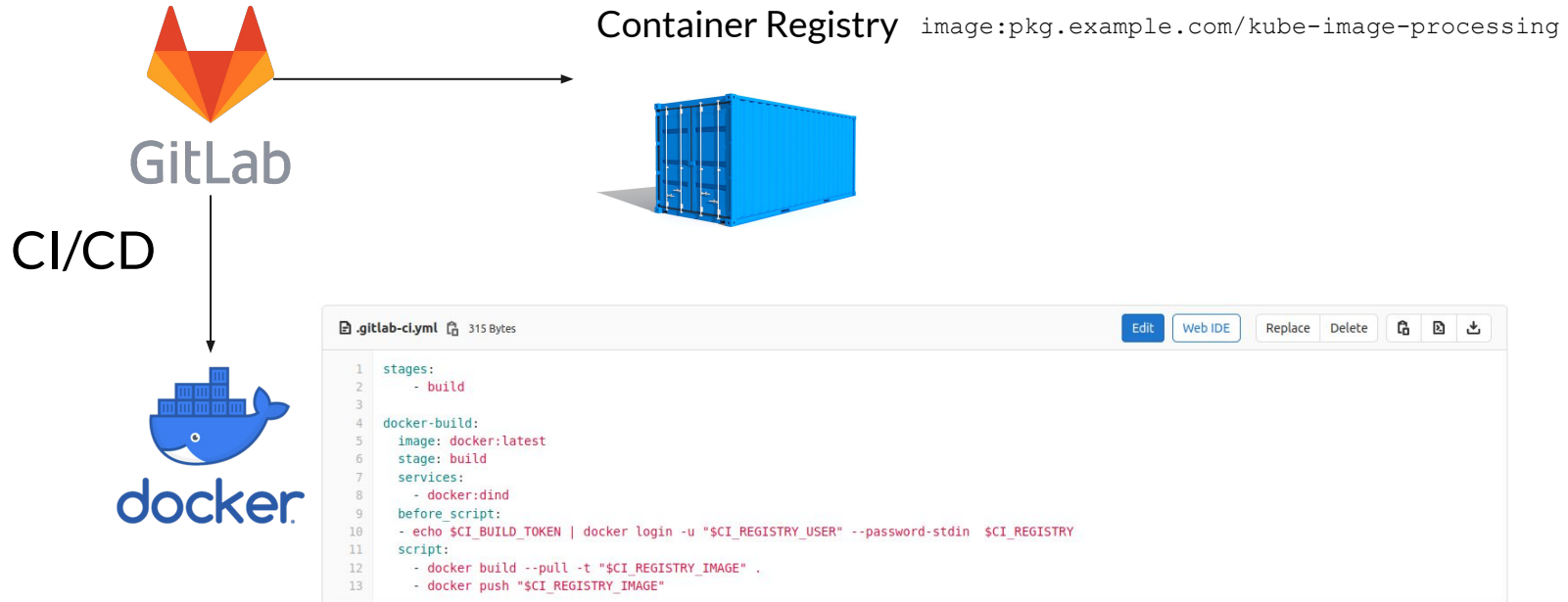
Which application is containerized ?

- REST API
 - GET /
 - POST /incoming
 - POST /processed
 - GET /processed/<name>
- Image processing App
- Python + Flask web framework
+ Minio SDK for Python
- JSON input



<https://images.pexels.com/photos/72161/pexels-photo-72161.jpeg?dl&fit=crop&w=640&h=318>

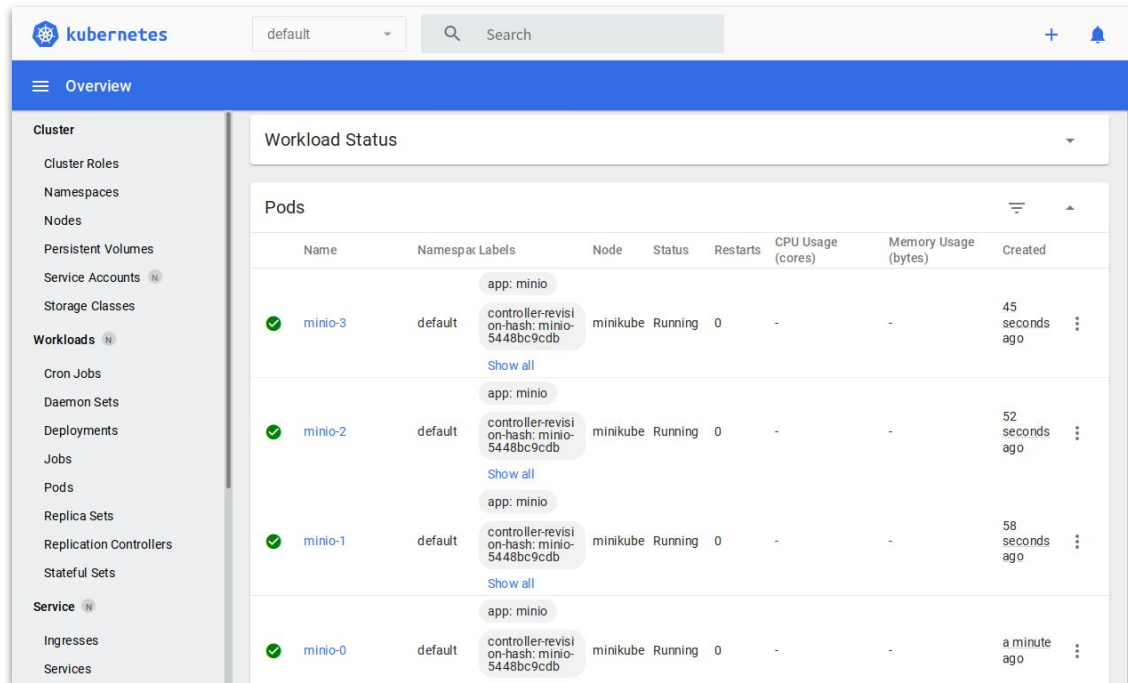
How is the App set up in Kubernetes Cluster ?









Demo!

Demo: Dashboard with MinIO



The screenshot displays the Kubernetes Dashboard interface. The top navigation bar includes the Kubernetes logo, a dropdown menu set to 'default', a search bar, and a notification bell. The left sidebar contains a navigation menu with categories like Cluster, Namespaces, Nodes, Workloads, and Service. The main content area is titled 'Workload Status' and shows a table of Pods.

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
 minio-3	default	app: minio controller-revisi on-hash: minio- 5448bc9cdb	minikube	Running	0	-	-	45 seconds ago
 minio-2	default	app: minio controller-revisi on-hash: minio- 5448bc9cdb	minikube	Running	0	-	-	52 seconds ago
 minio-1	default	app: minio controller-revisi on-hash: minio- 5448bc9cdb	minikube	Running	0	-	-	58 seconds ago
 minio-0	default	app: minio controller-revisi on-hash: minio- 5448bc9cdb	minikube	Running	0	-	-	a minute ago



Demo: Config: Pods

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-deployment
  labels:
    app: web.app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: web.app
  template:
    metadata:
      labels:
        app: web.app
    spec:
      containers:
      - name: web-container
        image: example.com/kube-image-processing
        ports:
        - containerPort: 5000
        env:
        - name: minio_hostname
          value: minio:9000
        - name: minio_access_key
          value: minio
        - name: minio_secret_key
          value: minio123
      imagePullSecrets:
      - name: kube-image-processing-regcred
```



Demo: Config: Service

```
---  
apiVersion: v1  
kind: Service  
metadata:  
  name: web-service  
spec:  
  selector:  
    app: web.app  
  type: NodePort  
  ports:  
  - protocol: TCP  
    port: 80  
    targetPort: 5000
```



Demo: Config: Ingress

```
---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: web-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web-service
            port:
              number: 80
```

Demo: Secrets

The screenshot shows the Kubernetes dashboard interface. At the top, there is a navigation bar with the Kubernetes logo, a dropdown menu set to 'default', a search bar, and a notification bell. Below this is a blue header bar with the breadcrumb 'Config and Storage > Secrets'. On the left side, there is a sidebar menu with various Kubernetes resource types. The 'Secrets' resource type is selected and highlighted. The main content area displays a table of secrets. The first row, 'kube-image-processing-regcred', is highlighted with a red box. The table has columns for Name, Namespace, Labels, Type, and Created. The 'Created' column shows the time since the secret was created, and there are pagination controls at the bottom right of the table.

Name	Namespace	Labels	Type	Created ↑
kube-image-processing-regcred	default	-	kubernetes.io/docker	27 days ago
regcred	default	-	kubernetes.io/docker	2 months ago
default-token-ldvtg	default	-	kubernetes.io/service-account-token	2 months ago

1 - 3 of 3 |< < > >|



Demo: Apply Configuration

```
kubectl apply -f imageProcessingDeployment.yaml
```

Demo: Dashboard with Deployment

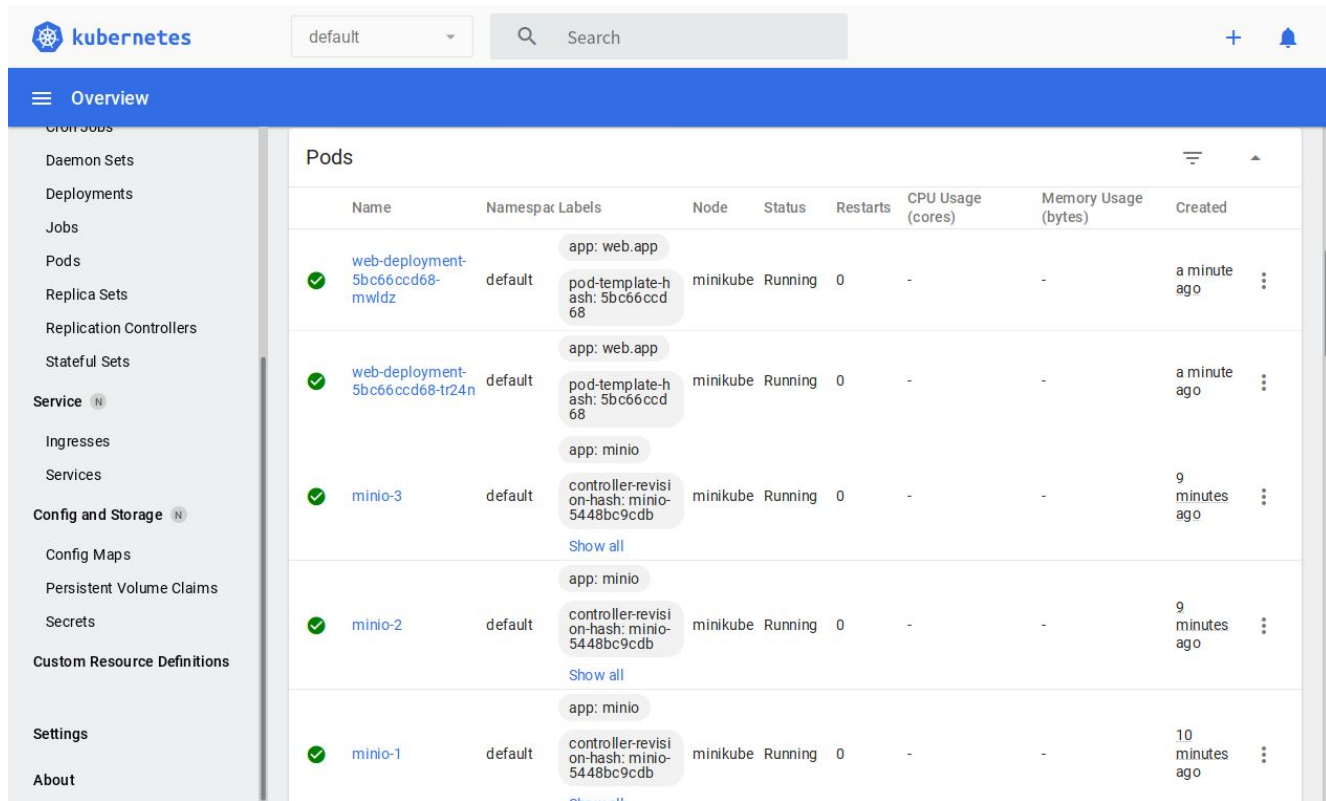
The screenshot displays the Kubernetes Dashboard interface. At the top, the 'kubernetes' logo is visible on the left, a 'default' namespace selector in the center, and a search bar on the right. A blue navigation bar contains the 'Overview' menu. The left sidebar lists various Kubernetes resources: CronJobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service (with a notification icon), Ingresses, Services, Config and Storage (with a notification icon), Config Maps, Persistent Volume Claims, Secrets, Custom Resource Definitions, Settings, and About.

The main content area is titled 'Workloads' and features a 'Workload Status' section with four large green circles representing the status of Deployments, Pods, Replica Sets, and Stateful Sets. Below this is a 'Deployments' table with the following data:

Name	Namespace	Labels	Pods	Created ↑	Images
web-deployment	default	app: web.app	2 / 2	39 seconds ago	pkg.latk.de/edu-cloud-computing-20ws/kube-image-processing

At the bottom right of the table, there is a pagination indicator '1 - 1 of 1' and navigation arrows.

Demo: Dashboard with Deployment



The screenshot shows the Kubernetes Dashboard interface. At the top, there is a navigation bar with the Kubernetes logo, a dropdown menu set to 'default', a search bar, and a notification bell. Below the navigation bar is a blue header with the word 'Overview'. On the left side, there is a sidebar menu with various Kubernetes resource types: CronJobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service (with a count of 1), Ingresses, Services, Config and Storage (with a count of 1), Config Maps, Persistent Volume Claims, Secrets, Custom Resource Definitions, Settings, and About. The main content area is titled 'Pods' and displays a table of running pods. The table has columns for Name, Namespace, Labels, Node, Status, Restarts, CPU Usage (cores), Memory Usage (bytes), and Created. There are five rows of pods, all with a status of 'Running' and '0' restarts. The first two rows are for a 'web-deployment' and the last three are for a 'minio' deployment. Each row includes a green checkmark icon and a vertical ellipsis menu icon.

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
web-deployment-5bc66ccd68-mwldz	default	app: web.app pod-template-hash: 5bc66ccd68	minikube	Running	0	-	-	a minute ago
web-deployment-5bc66ccd68-tr24n	default	app: web.app pod-template-hash: 5bc66ccd68	minikube	Running	0	-	-	a minute ago
minio-3	default	app: minio controller-revision-hash: minio-5448bc9cdb	minikube	Running	0	-	-	9 minutes ago
minio-2	default	app: minio controller-revision-hash: minio-5448bc9cdb	minikube	Running	0	-	-	9 minutes ago
minio-1	default	app: minio controller-revision-hash: minio-5448bc9cdb	minikube	Running	0	-	-	10 minutes ago



Demo: Scaling

```
kubectl scale --replicas=3 deployment/web-deployment
```


Demo: Dashboard with scaled Deployment

The screenshot displays the Kubernetes Dashboard interface. The top navigation bar includes the Kubernetes logo, a namespace dropdown set to 'default', a search bar, and a notification bell. The left sidebar contains a navigation menu with categories like 'Overview', 'Jobs', 'Pods', 'Replica Sets', 'Replication Controllers', 'Stateful Sets', 'Service', 'Ingresses', 'Services', 'Config and Storage', 'Custom Resource Definitions', 'Settings', and 'About'. The main content area is titled 'Pods' and shows a table of running pods. The table has columns for Name, Namespace, Labels, Node, Status, Restarts, CPU Usage (cores), Memory Usage (bytes), and Created. There are six pods listed, all in a 'Running' state. The first three pods are part of a 'web-deployment' and are running on 'minikube' nodes. The last two pods are part of a 'minio' deployment and are also running on 'minikube' nodes. Each pod row includes a green checkmark icon, a 'Show all' link, and a vertical ellipsis menu.

Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
web-deployment-5bc66ccd68-s9kqt	default	app: web.app pod-template-hash: 5bc66ccd68	minikube	Running	0	-	-	24 seconds ago
web-deployment-5bc66ccd68-tr24n	default	app: web.app pod-template-hash: 5bc66ccd68	minikube	Running	0	-	-	2 minutes ago
web-deployment-5bc66ccd68-mwldz	default	app: web.app pod-template-hash: 5bc66ccd68	minikube	Running	0	-	-	2 minutes ago
minio-3	default	app: minio controller-revision-hash: minio-5448bc9cdb	minikube	Running	0	-	-	11 minutes ago
minio-2	default	app: minio controller-revision-hash: minio-5448bc9cdb	minikube	Running	0	-	-	11 minutes ago

Demo: Dashboard with Ingress

Ingress at IP
192.168.39.210

The screenshot shows the Kubernetes dashboard interface. At the top, there's a header with the Kubernetes logo, a namespace dropdown menu set to 'default', and a search bar. Below the header, a blue navigation bar indicates the current path: 'Discovery and Load Balancing > Ingresses'. The left sidebar contains a list of resource categories: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service (with a notification icon), Ingresses (highlighted), Services, Config and Storage (with a notification icon), Config Maps, Persistent Volume Claims, Secrets, Custom Resource Definitions, Settings, and About. The main content area displays a table titled 'Ingresses'. The table has columns for Name, Namespace, Labels, Endpoints, and Created. A single entry is shown: 'web-ingress' in the 'default' namespace with no labels, endpoints at '192.168.39.210', and created '5 minutes ago'. At the bottom of the table, it shows '1 - 1 of 1' and navigation arrows.

Name	Namespace	Labels	Endpoints	Created ↑
web-ingress	default	-	192.168.39.210	5 minutes ago

Demo: REST API

open <http://192.168.39.210>

Logs from web-container ▾ in web-deployment-5bc66ccd68-mwldz ▾

```
* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.4 - - [11/Feb/2021 15:57:03] "GET / HTTP/1.1" 200 -
```

JSON

Raw Data

Headers

Save

Copy

Collapse All

Expand All

Filter JSON

routes:

0:

method: "POST"

path: "/incoming"

1:

method: "POST"

path: "/processed"

2:

method: "GET"

path: "/incoming/<name>"

3:

method: "GET"

path: "/processed/<name>"

Demo: Upload data

```
curl http://192.168.39.210/incoming  
  -H 'Content-Type: application/json'  
  --data '{"url":"http://192.168.1.103:8123/norway.jpg"}'
```



Response:

```
{"path":"/incoming/56df5bf60b0500d5  
e9a6e85193ca4b52b21024b8dc5ce6f8c48  
82bfaacea24b1"}
```

Demo: Convert image

```
curl http://192.168.39.210/processed  
  -H 'Content-Type: application/json'  
  --data '{"path":"/incoming/56df5bf..."}'
```



Response:

```
{"path":"/processed/011e8ba0a6ace92  
8861845837e84578efbf75382d8c1595bb1  
aa66e33b621e4f"}
```