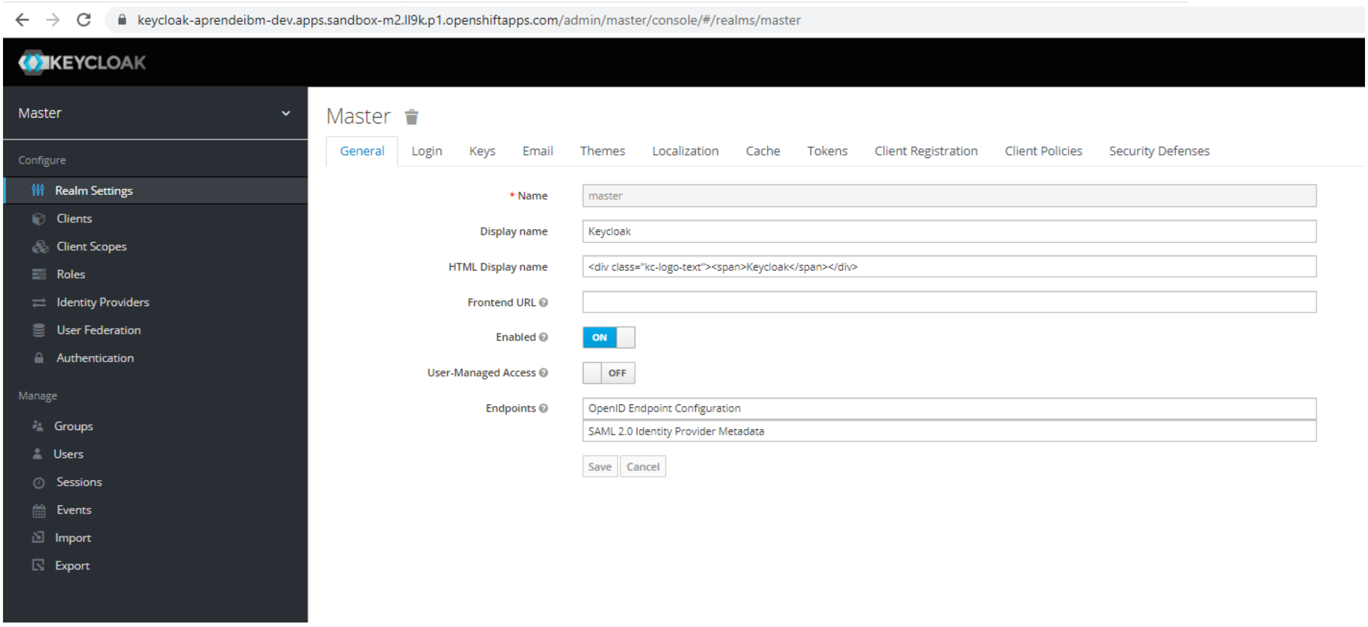
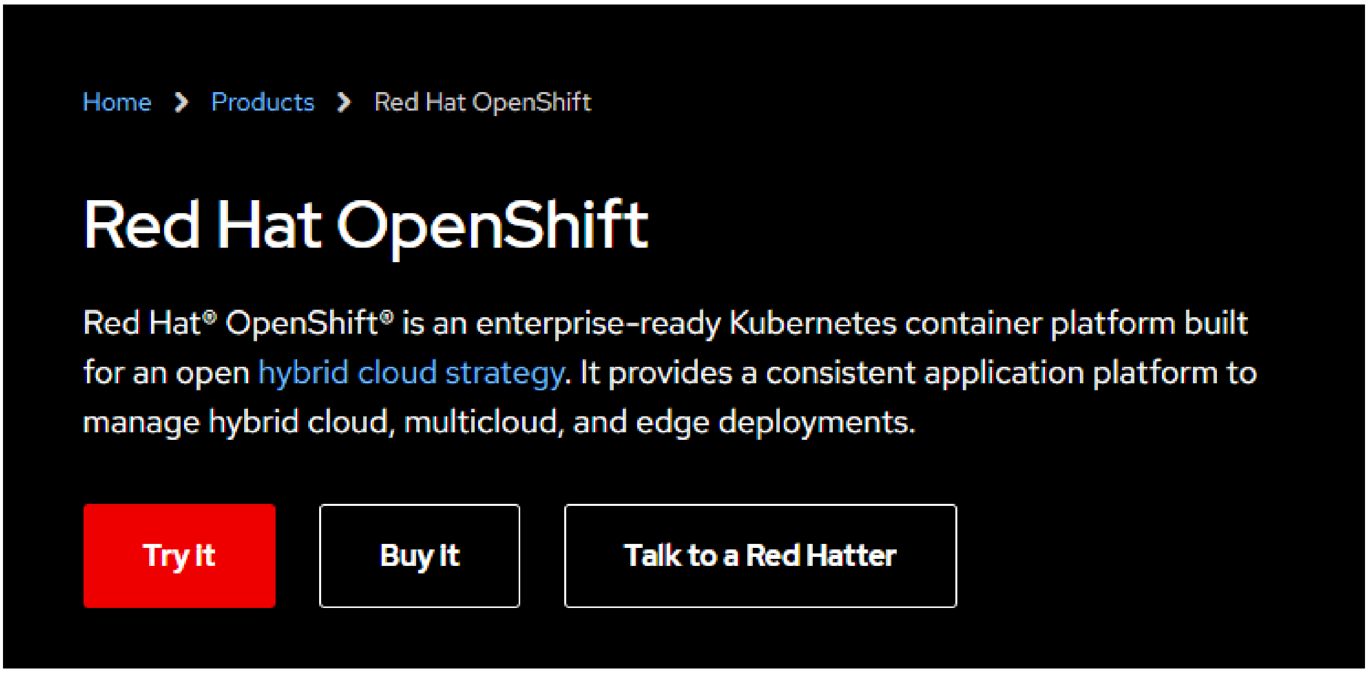
En este artículo vamos a ver los pasos para desplegar **KeyCloak** en un ambiente de **RedHat OpenShift**.


KEYCLOAK
Master
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Master
General
Login
Keys
Email
Themes
Localization
Cache
Tokens
Client Registration
Client Policies
Security Defenses
Save
Cancel

keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com/admin/master/console/#/realms/master

* Name: master
Display name: Keycloak
HTML Display name: <div class="kc-logo-text">Keycloak</div>
Frontend URL:
Enabled: ON
User-Managed Access: OFF
Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

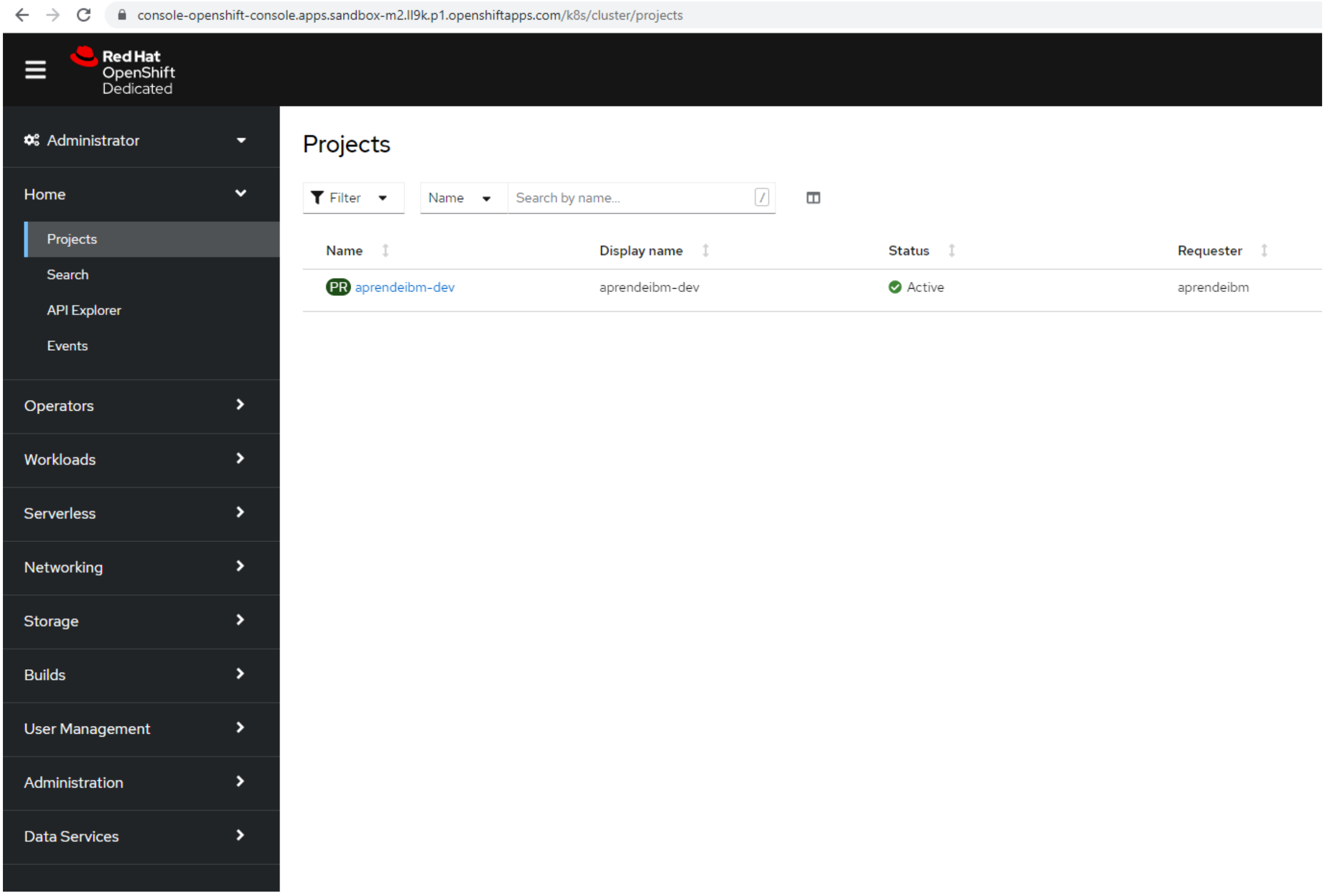
_1. Crear una cuenta trial de **OpenShift** en https://www.redhat.com/en/technologies/cloud-computing/openshift


Home
Products
Red Hat OpenShift
Try It
Buy It
Talk to a Red Hatter

Red Hat OpenShift

Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform built for an open hybrid cloud strategy. It provides a consistent application platform to manage hybrid cloud, multicloud, and edge deployments.

_2. Luego de disponer de una cuenta ingresamos al panel de administración, dependiendo de los datos ingresados, nos van a crear un proyecto, en este caso el proyecto tiene por nombre (**aprendeibm-dev**)


console-openshift-console.apps.sandbox-m2.ll9k.p1.openshiftapps.com/k8s/cluster/projects
Red Hat
OpenShift
Dedicated
Administrator
Home
Projects
Search
API Explorer
Events
Operators
Workloads
Serverless
Networking
Storage
Builds
User Management
Administration
Data Services
Filter
Name
Search by name...

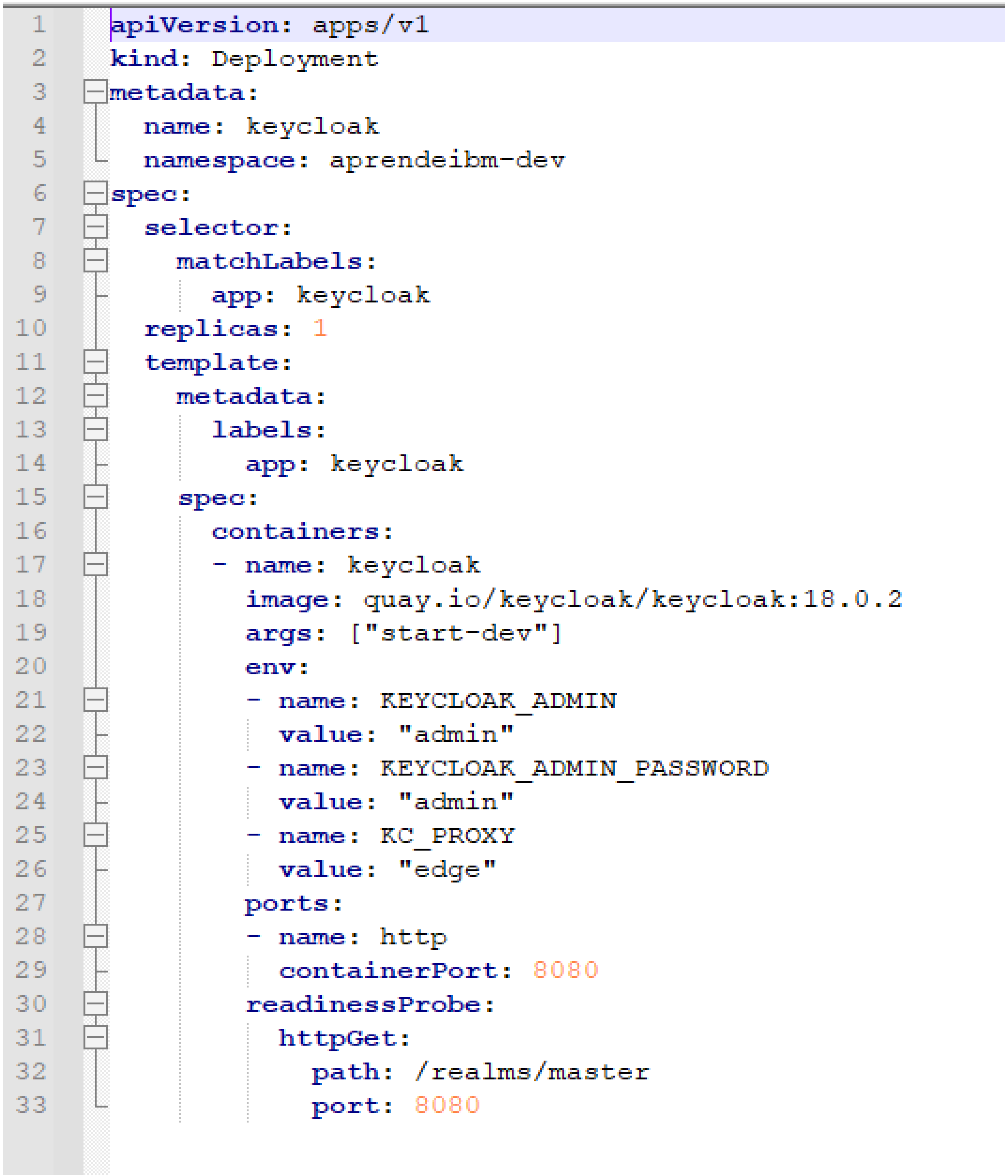
Projects

Name	Display name	Status	Requester
PR aprendeibm-dev	aprendeibm-dev	✓ Active	aprendeibm

_3. Ahora vamos a crear los siguientes objetos

- Deployment
- Service
- Route

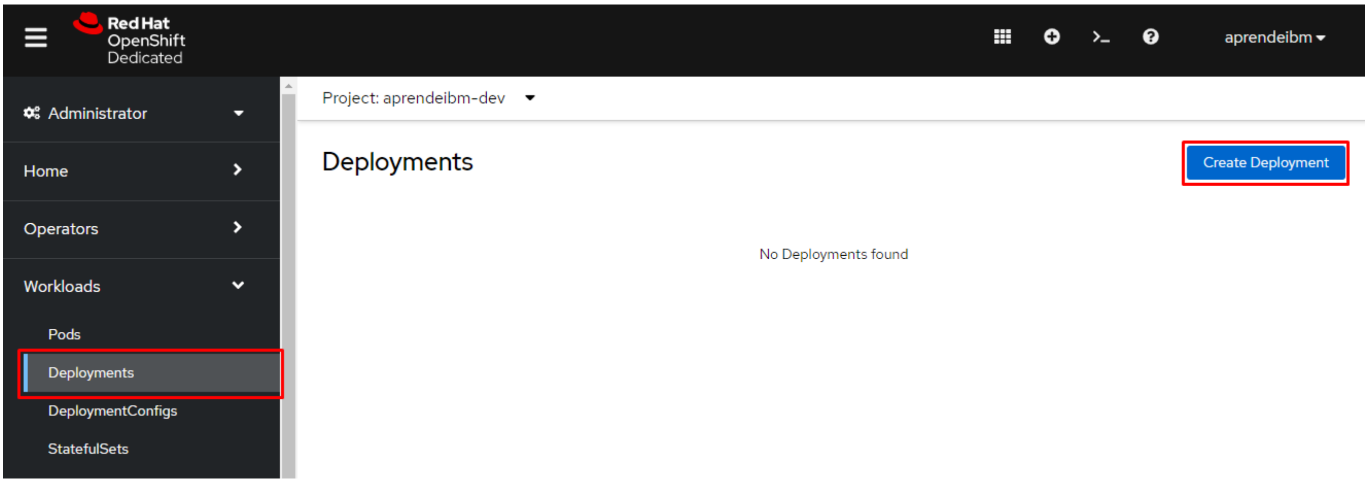
_4. **Crear Deployment**: en el menú nos vamos a ***Workloads>>Deployments*** y luego dar clic en «**Create Deployment**«, adjunto archivo con el contenido a copiar y pegar (only-keycloak-deployment.txt), luego le damos en «**Create**«.



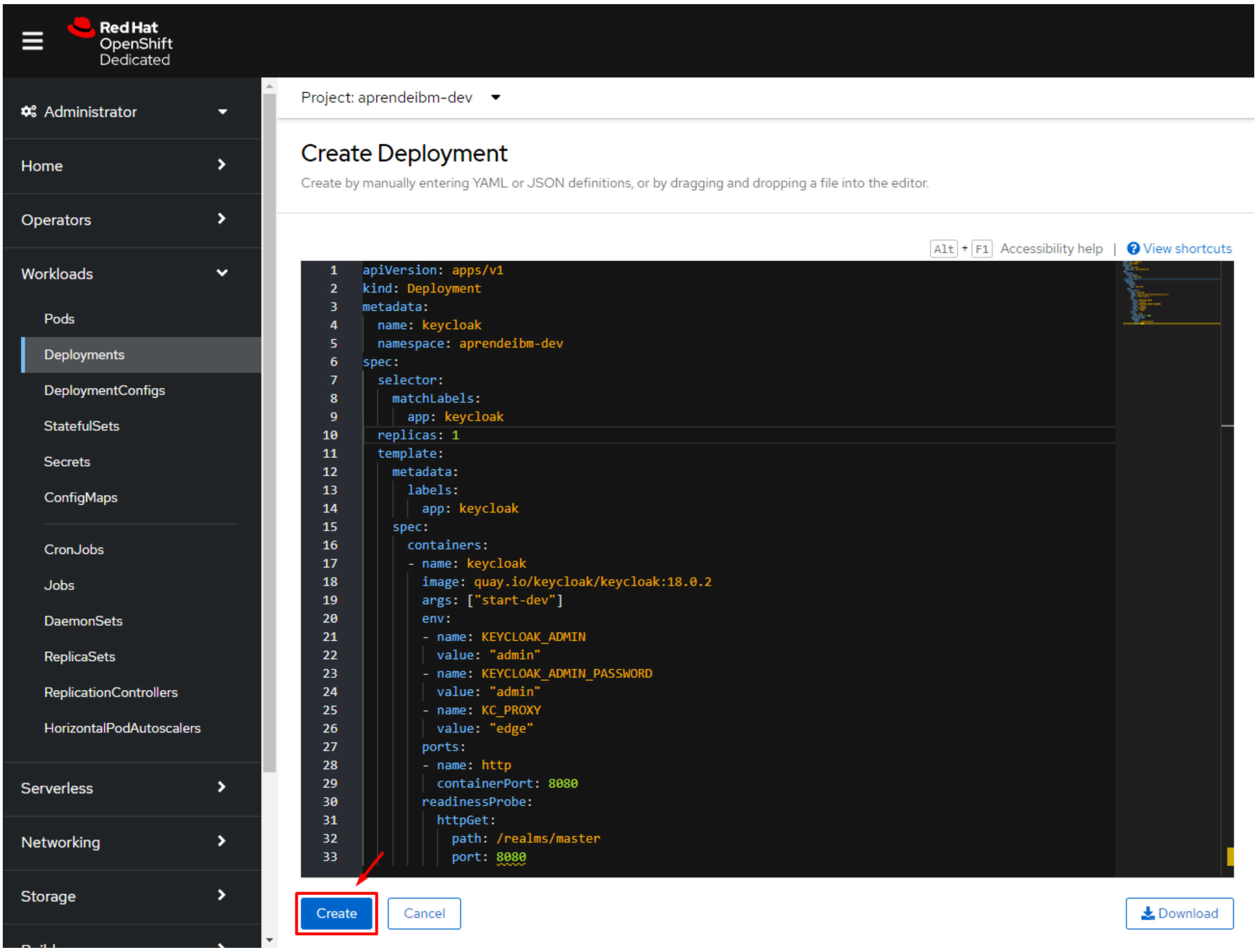
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: keycloak
  namespace: aprendeibm-dev
spec:
  selector:
    matchLabels:
      app: keycloak
  replicas: 1
  template:
    metadata:
      labels:
        app: keycloak
    spec:
      containers:
      - name: keycloak
        image: quay.io/keycloak/keycloak:18.0.2
        args: ["start-dev"]
        env:
        - name: KEYCLOAK_ADMIN
          value: "admin"
        - name: KEYCLOAK_ADMIN_PASSWORD
          value: "admin"
        - name: KC_PROXY
          value: "edge"
        ports:
        - name: http
          containerPort: 8080
        readinessProbe:
          httpGet:
            path: /realms/master
            port: 8080
```

En este apartado es bueno revisar la versión de la imagen que vamos a desplegar de **KeyCloak** ya que en nuevas versiones cambia el **look and feel**, en este caso vamos a utilizar la versión **18.0.2**

Para revisar las versiones disponibles a desplegar en un entorno de **Kubernetes**, visitar: https://www.keycloak.org/getting-started/getting-started-kube


Red Hat
OpenShift
Dedicated
aprendeibm
Administrator
Home
Operators
Workloads
Pods
Deployments
DeploymentConfigs
StatefulSets
Project: aprendeibm-dev
Deployments
Create Deployment

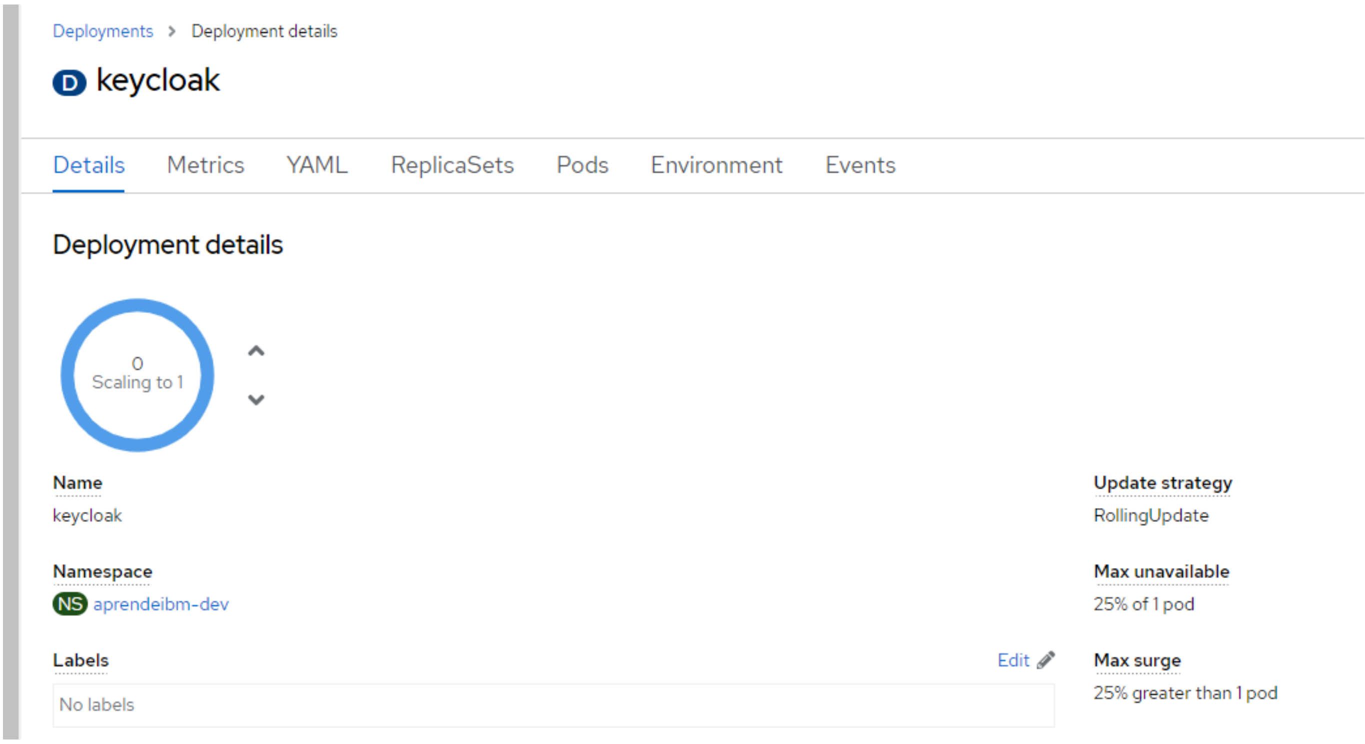
No Deployments found


Red Hat
OpenShift
Dedicated
Administrator
Home
Operators
Workloads
Pods
Deployments
DeploymentConfigs
StatefulSets
Secrets
ConfigMaps
CronJobs
Jobs
DaemonSets
ReplicaSets
ReplicationControllers
HorizontalPodAutoscalers
Serverless
Networking
Storage
Project: aprendeibm-dev
Alt + F1 Accessibility help | View shortcuts
Create
Cancel
Download

Create Deployment

Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: keycloak
  namespace: aprendeibm-dev
spec:
  selector:
    matchLabels:
      app: keycloak
  replicas: 1
  template:
    metadata:
      labels:
        app: keycloak
    spec:
      containers:
      - name: keycloak
        image: quay.io/keycloak/keycloak:18.0.2
        args: ["start-dev"]
        env:
        - name: KEYCLOAK_ADMIN
          value: "admin"
        - name: KEYCLOAK_ADMIN_PASSWORD
          value: "admin"
        - name: KC_PROXY
          value: "edge"
        ports:
        - name: http
          containerPort: 8080
        readinessProbe:
          httpGet:
            path: /realms/master
            port: 8080
```


Deployments > Deployment details
D keycloak
Details
Metrics
YAML
ReplicaSets
Pods
Environment
Events
0
Scaling to 1
Edit

Deployment details

Name
keycloak

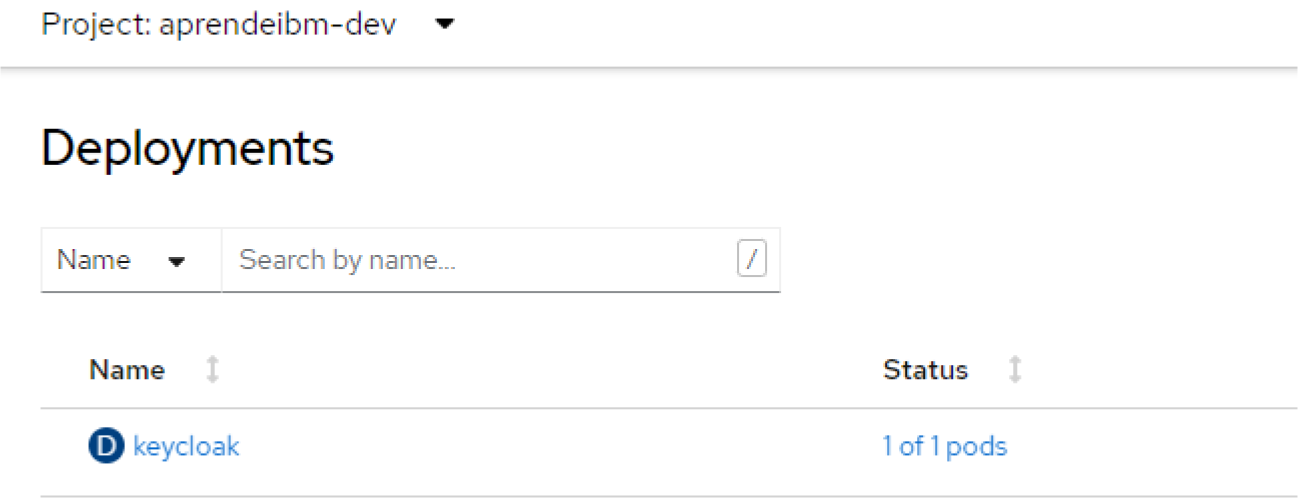
Namespace
NS aprendeibm-dev

Labels
No labels

Update strategy
RollingUpdate

Max unavailable
25% of 1 pod

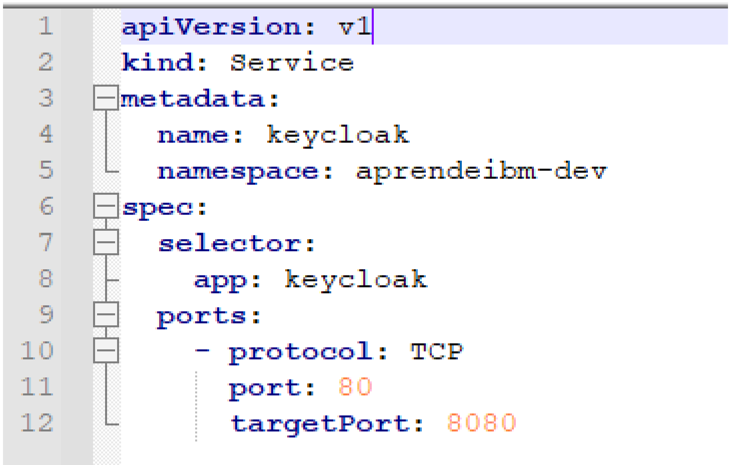
Max surge
25% greater than 1 pod


Project: aprendeibm-dev
Name
Search by name...
/

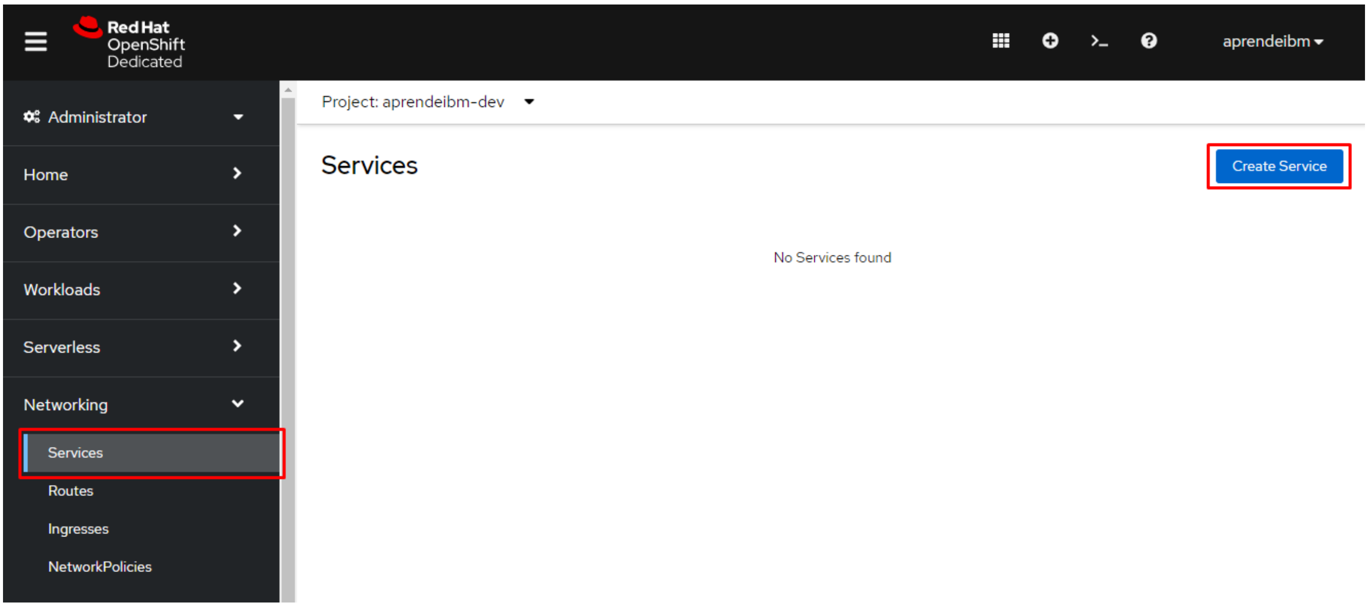
Deployments

Name	Status
D keycloak	1 of 1 pods

_5. **Crear Service**: Ingresar al menú ***Networking >> Services*** y clic en «**Create Service**«, ahí debemos colocar lo siguiente, se adjunta ([only-keycloak-service.txt]()).



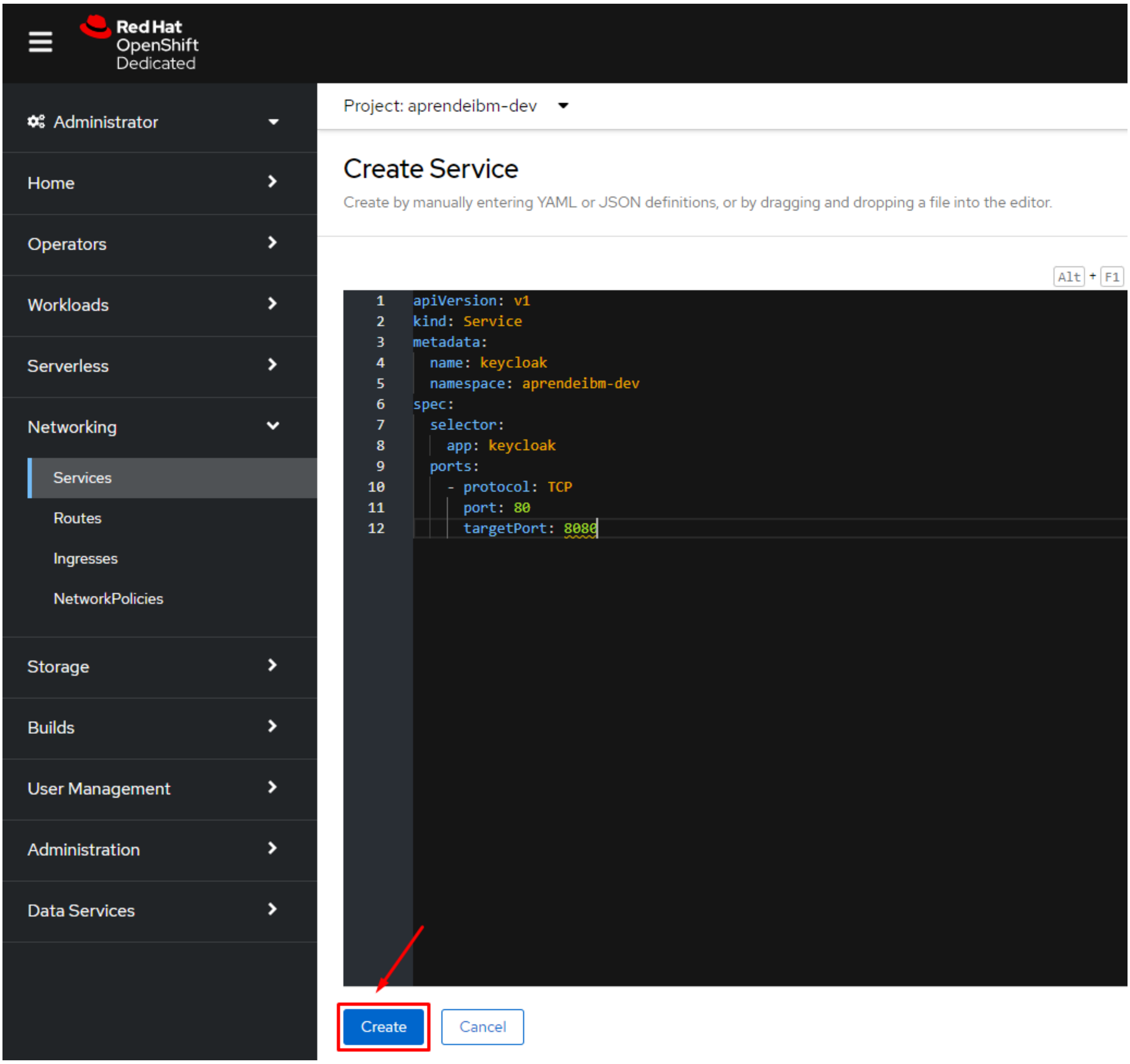
```
apiVersion: v1
kind: Service
metadata:
  name: keycloak
  namespace: aprendeibm-dev
spec:
  selector:
    app: keycloak
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```



Red Hat
OpenShift
Dedicated
aprendeibm
Administrator
Home
Operators
Workloads
Serverless
Networking
Services
Routes
Ingresses
NetworkPolicies
Services
Create Service

Project: aprendeibm-dev

No Services found

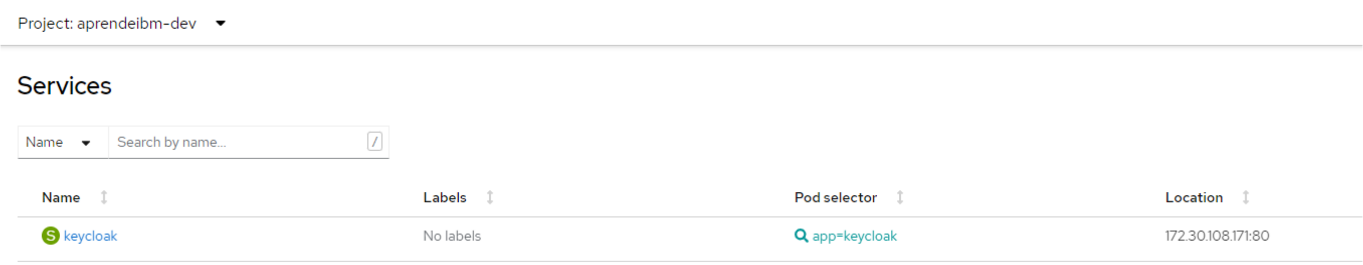

Red Hat
OpenShift
Dedicated
Administrator
Home
Operators
Workloads
Serverless
Networking
Services
Routes
Ingresses
NetworkPolicies
Storage
Builds
User Management
Administration
Data Services
Alt + F1
Create
Cancel

Project: aprendeibm-dev

Create Service

Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

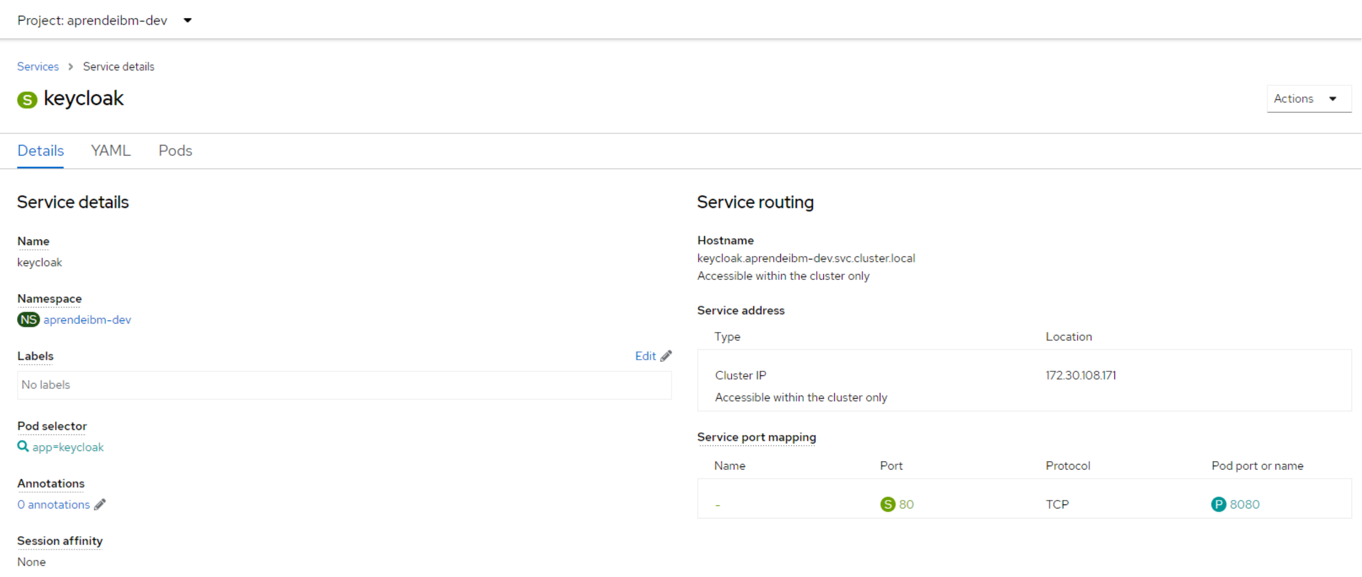
```yaml
apiVersion: v1
kind: Service
metadata:
  name: keycloak
  namespace: aprendeibm-dev
spec:
  selector:
    app: keycloak
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```


Name
Search by name...
/

Project: aprendeibm-dev

Services

Name	Labels	Pod selector	Location
keycloak	No labels	app=keycloak	172.30.108.171:80



Project: aprendeibm-dev

Services > Service details

S keycloak

Actions

Details YAML Pods

Service details

Name
keycloak

Namespace
NS aprendeibm-dev

Labels
No labels

Pod selector
app=keycloak

Annotations
0 annotations

Session affinity
None

Service routing

Hostname
keycloak.aprendeibm-dev.svc.cluster.local
Accessible within the cluster only

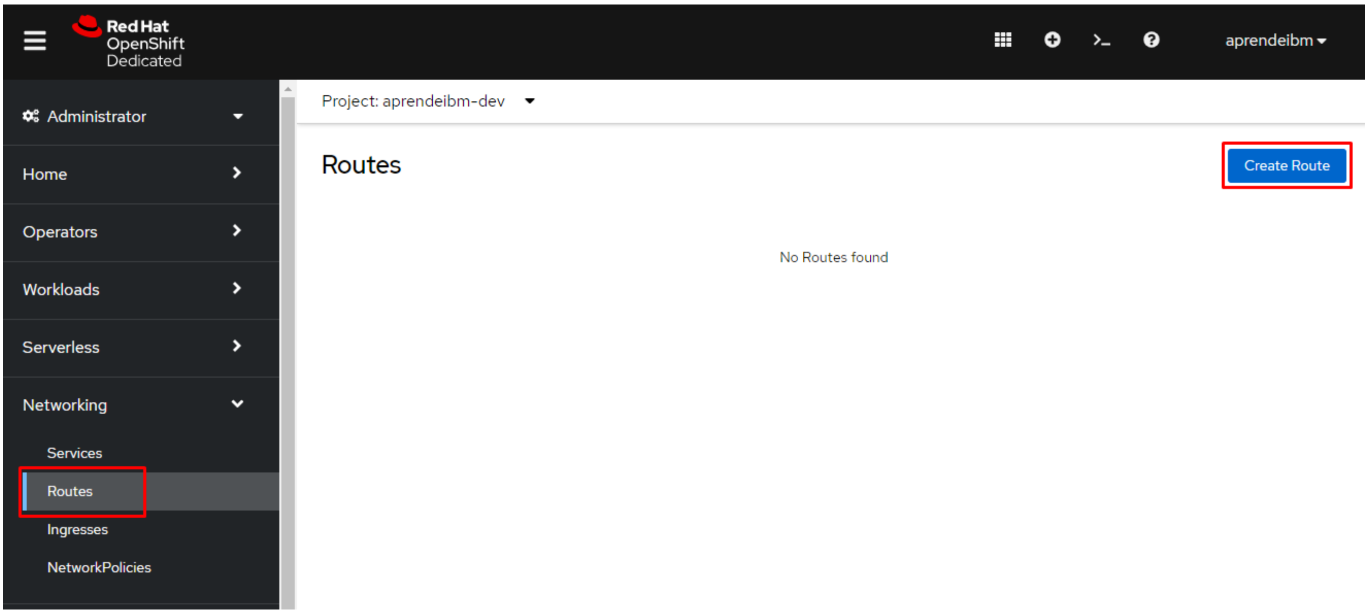
Service address

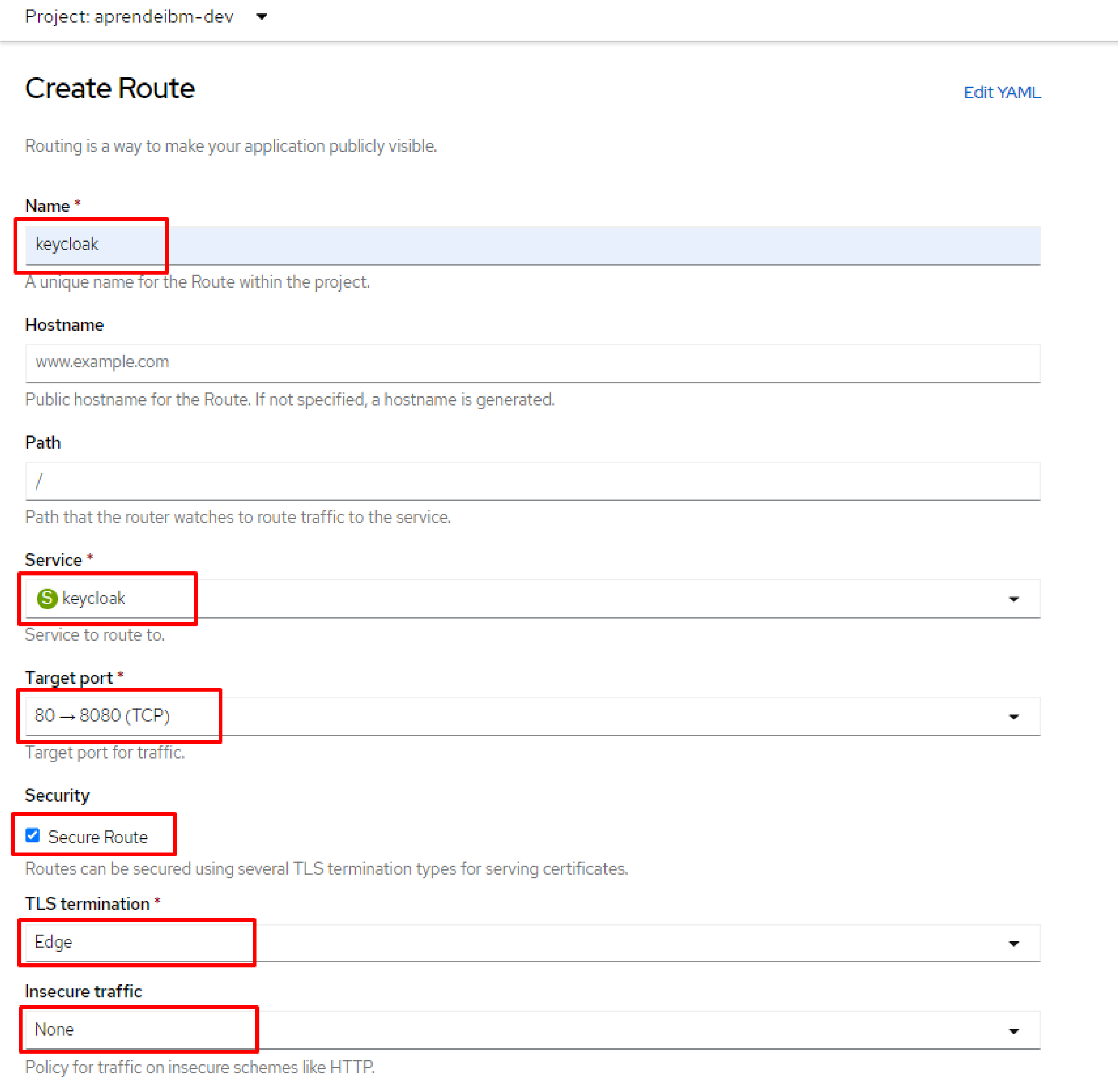
Type	Location
Cluster IP Accessible within the cluster only	172.30.108.171

Service port mapping

Name	Port	Protocol	Pod port or name
-	S 80	TCP	P 8080

_6. **Crear Route**: Ingresar al menú, ***Networking>>Routes*** y luego clic en «**Create Route**»


Red Hat
OpenShift
Dedicated
aprendeibm
Administrator
Home
Operators
Workloads
Serverless
Networking
Services
Routes
Ingresses
NetworkPolicies
Project: aprendeibm-dev
Routes
Create Route
No Routes found



Create Route

Routing is a way to make your application publicly visible.

Name *
keycloak
A unique name for the Route within the project.

Hostname
www.example.com
Public hostname for the Route. If not specified, a hostname is generated.

Path
/
Path that the router watches to route traffic to the service.

Service *
S keycloak
Service to route to.

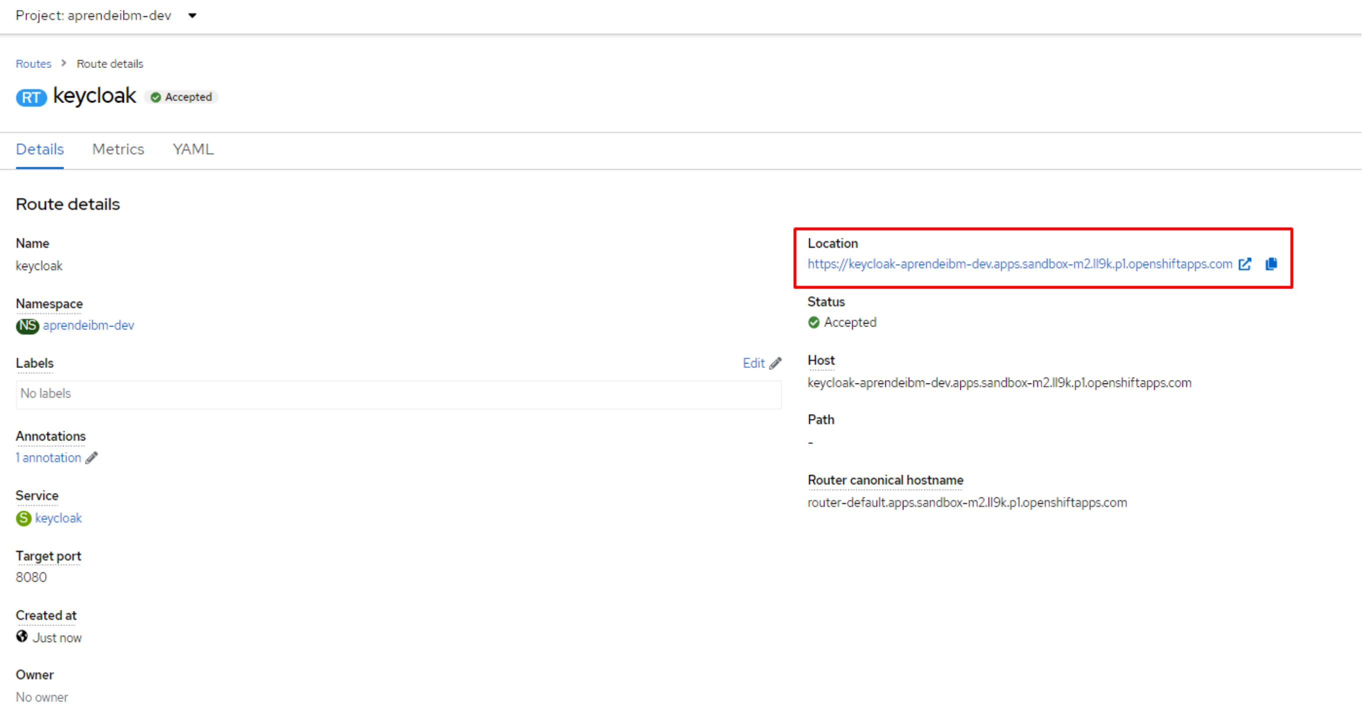
Target port *
80 → 8080 (TCP)
Target port for traffic.

Security
☑ Secure Route
Routes can be secured using several TLS termination types for serving certificates.

TLS termination *
Edge

Insecure traffic
None
Policy for traffic on insecure schemes like HTTP.

Aquí debemos seleccionar el **Service**, **Target port**, **Security**, **TLS terminarion (Edge** para tener https en la ruta que va a exponer nuestro Route**)**, **Insecure traffic** según la imagen. Luego presionar en «**Create**«.



Project: aprendeibm-dev

Routes > Route details

RT keycloak ✓ Accepted

Details | Metrics | YAML

Route details

Name
keycloak

Namespace
NS aprendeibm-dev

Labels Edit
No labels

Annotations
1 annotation

Service
S keycloak

Target port
8080

Created at
Just now

Owner
No owner

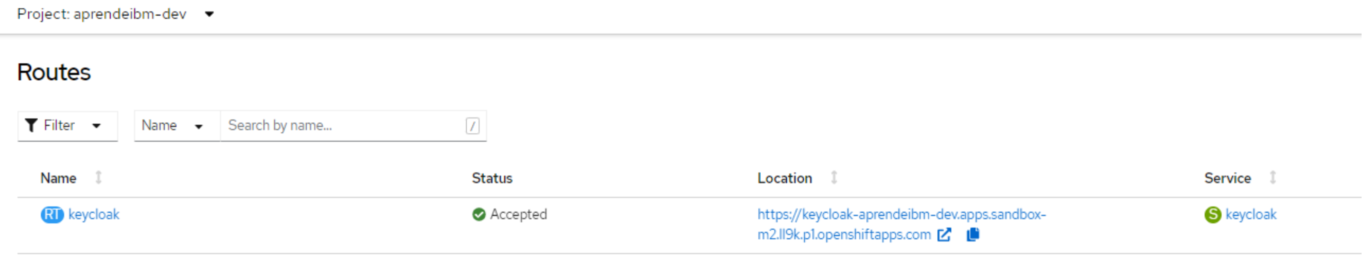
Location
https://keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com

Status
✓ Accepted

Host
keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com

Path
-

Router canonical hostname
router-default.apps.sandbox-m2.ll9k.p1.openshiftapps.com

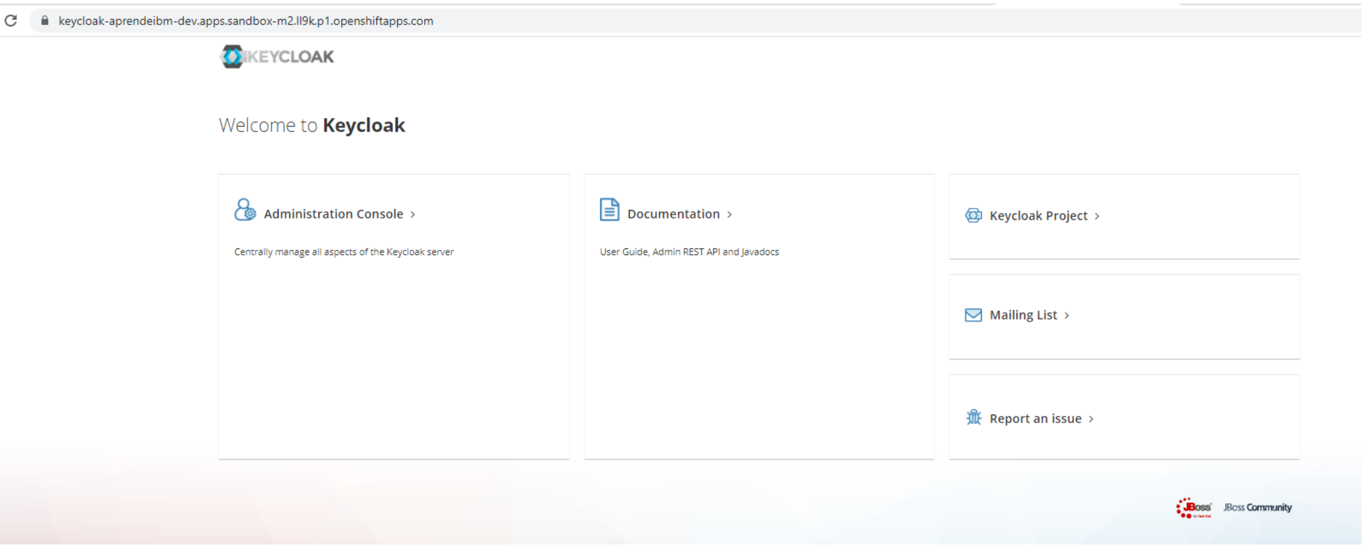


Project: aprendeibm-dev

Routes

Name	Status	Location	Service
RT keycloak	✓ Accepted	https://keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com	S keycloak

_7. Ingresar a la ruta indicada en el «**Location**»

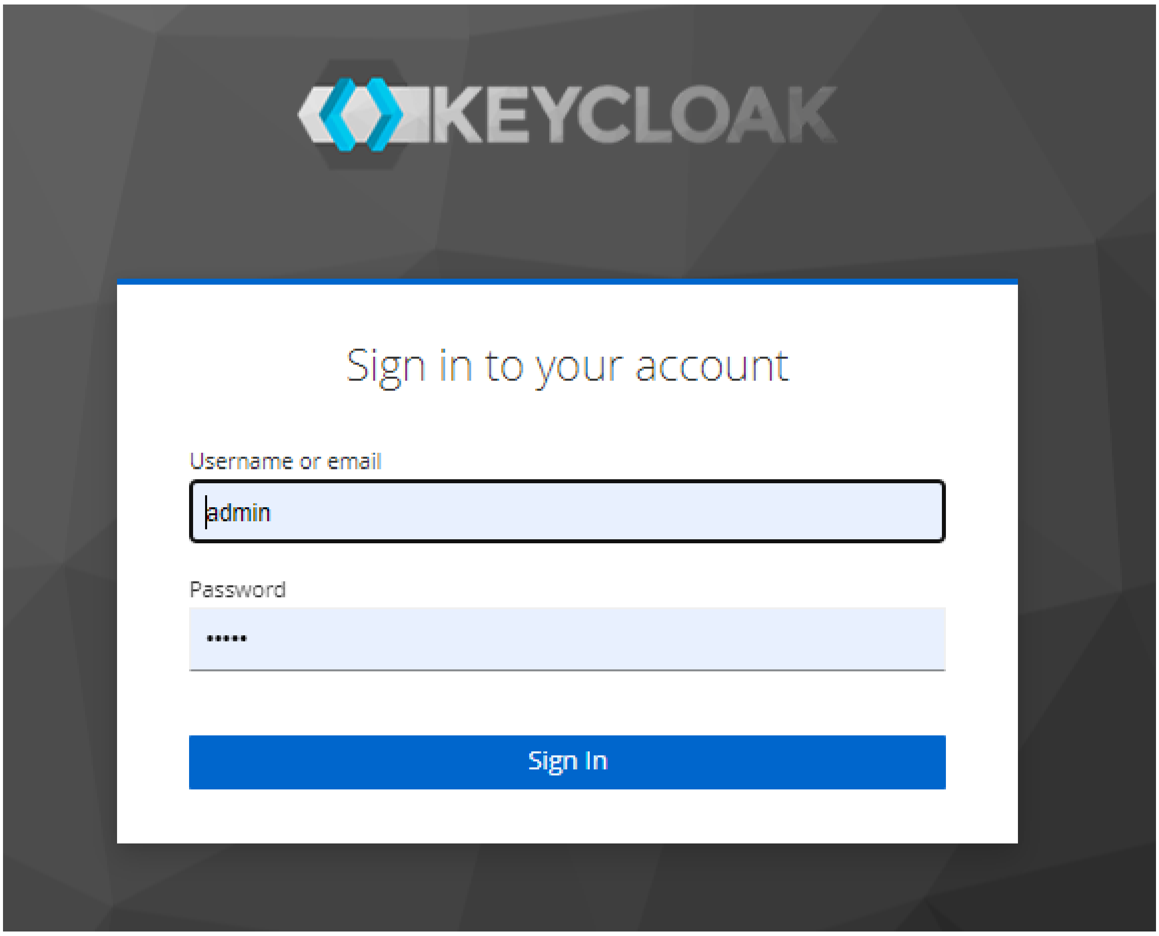


keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com

Welcome to **Keycloak**

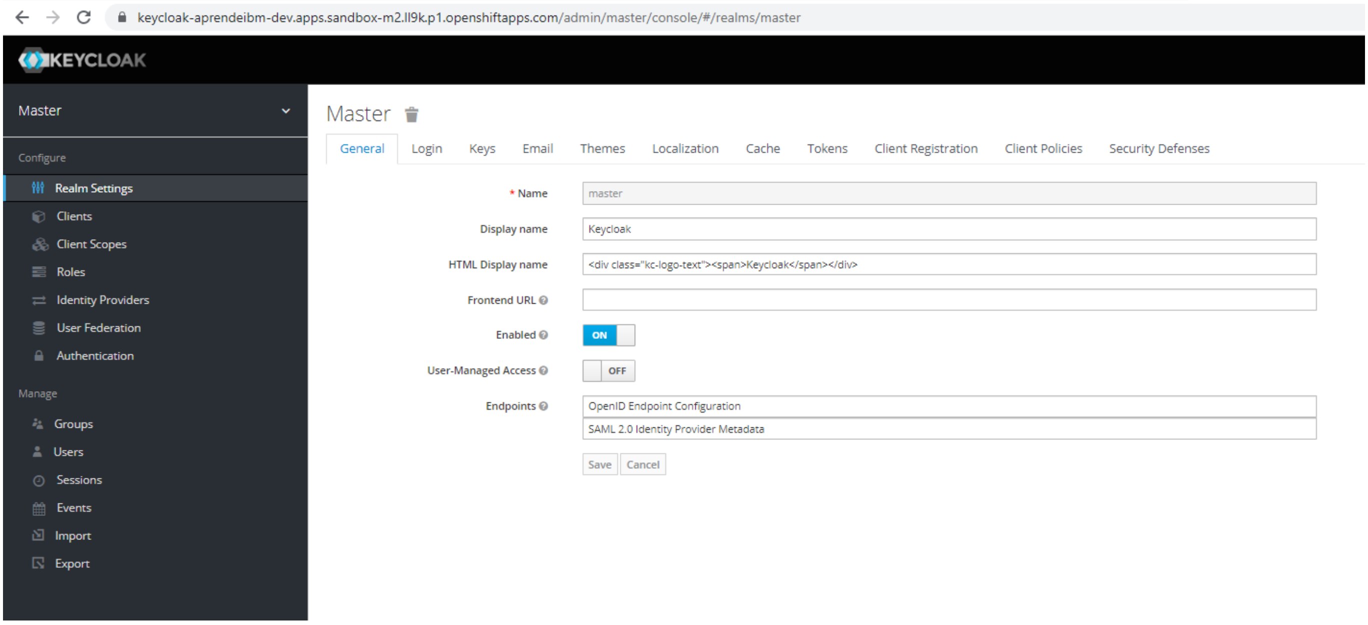
Administration Console ›
Centrally manage all aspects of the Keycloak server

Documentation ›
User Guide, Admin REST API and Javadocs


KEYCLOAK
Sign in to your account
Sign In

Username or email: admin

Password: •••••


keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com/admin/master/console/#/realms/master

KEYCLOAK
Master
Configure
Realm Settings
Clients
Client Scopes
Roles
Identity Providers
User Federation
Authentication
Manage
Groups
Users
Sessions
Events
Import
Export
Master
General
Login
Keys
Email
Themes
Localization
Cache
Tokens
Client Registration
Client Policies
Security Defenses
Save
Cancel

* Name: master

Display name: Keycloak

HTML Display name: <div class="kc-logo-text">Keycloak</div>

Frontend URL:

Enabled: ON

User-Managed Access: OFF

Endpoints: OpenID Endpoint Configuration
SAML 2.0 Identity Provider Metadata

Julio Pari (IT Architect IBM)

Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíame un mensaje a: (info@juliopari.com) o sino a través de Linkedin: https://www.linkedin.com/in/juliopari/