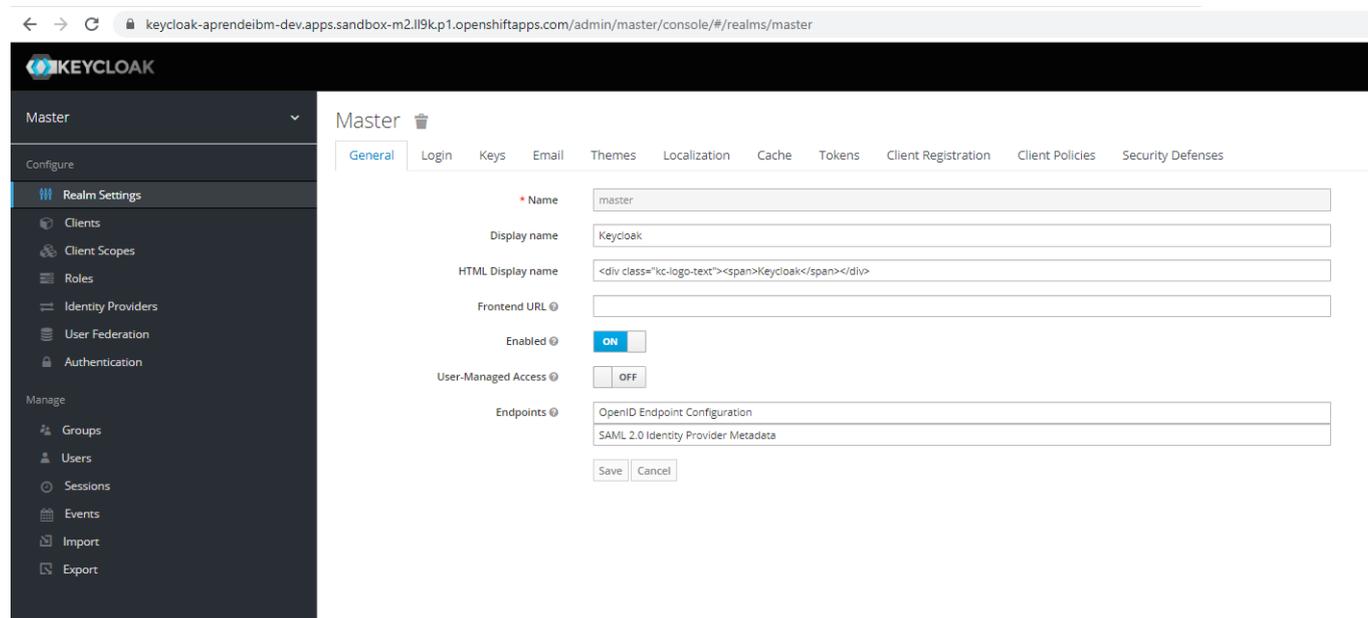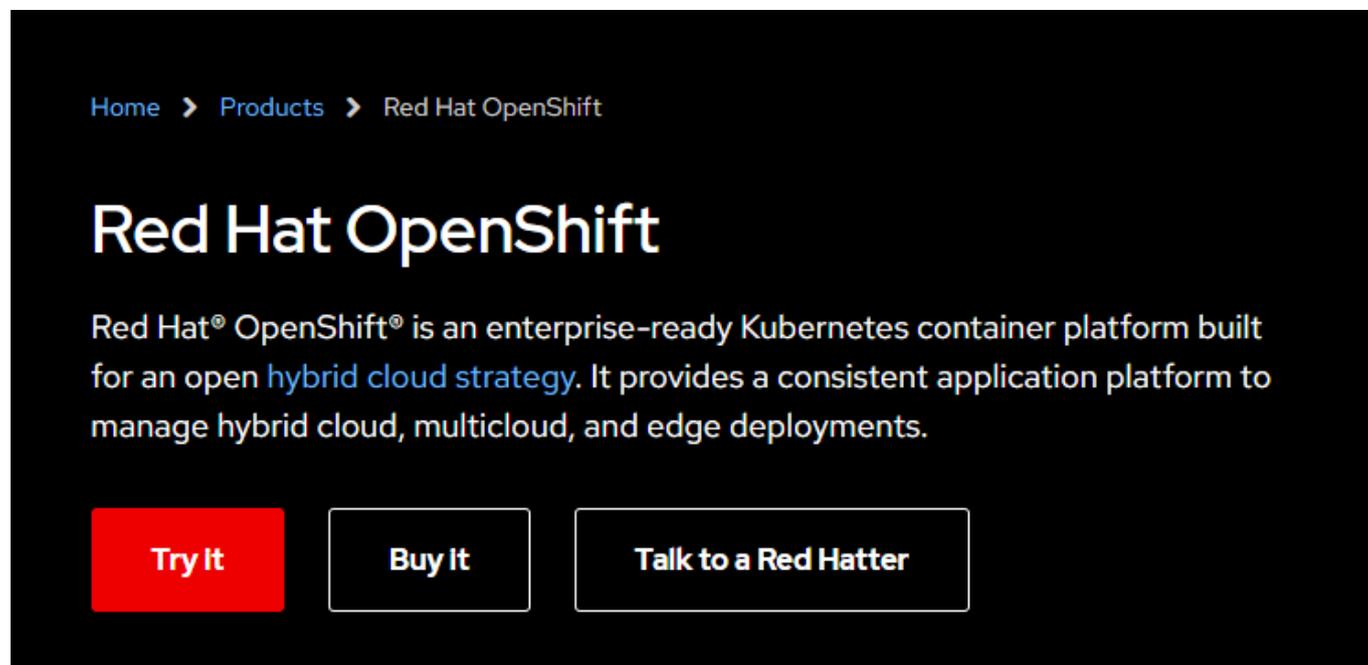En este artículo vamos a ver los pasos para desplegar **KeyCloak** en un ambiente de **RedHat OpenShift**.

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

_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**)
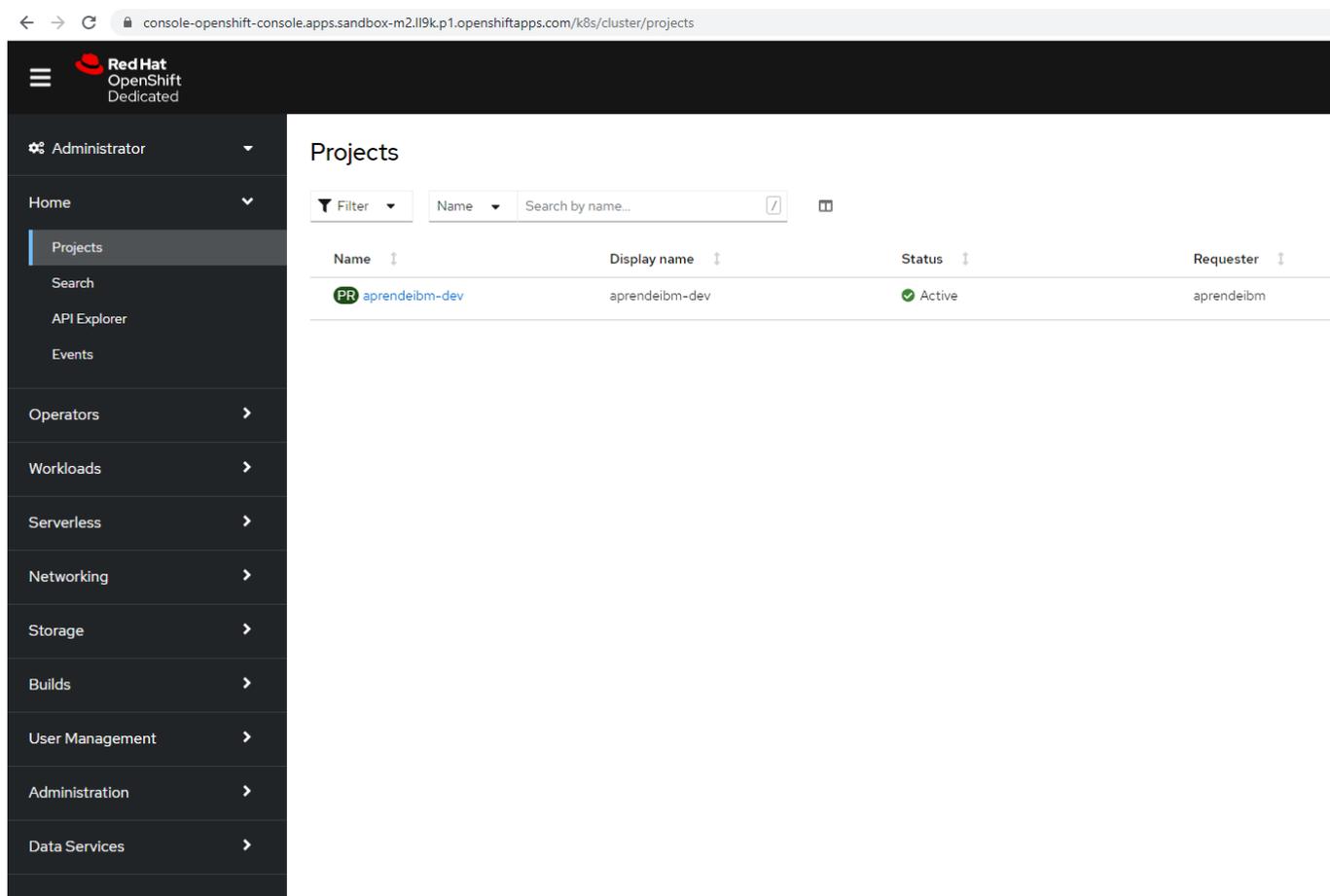
_3. Ahora vamos a crear los siguientes objetos

- 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
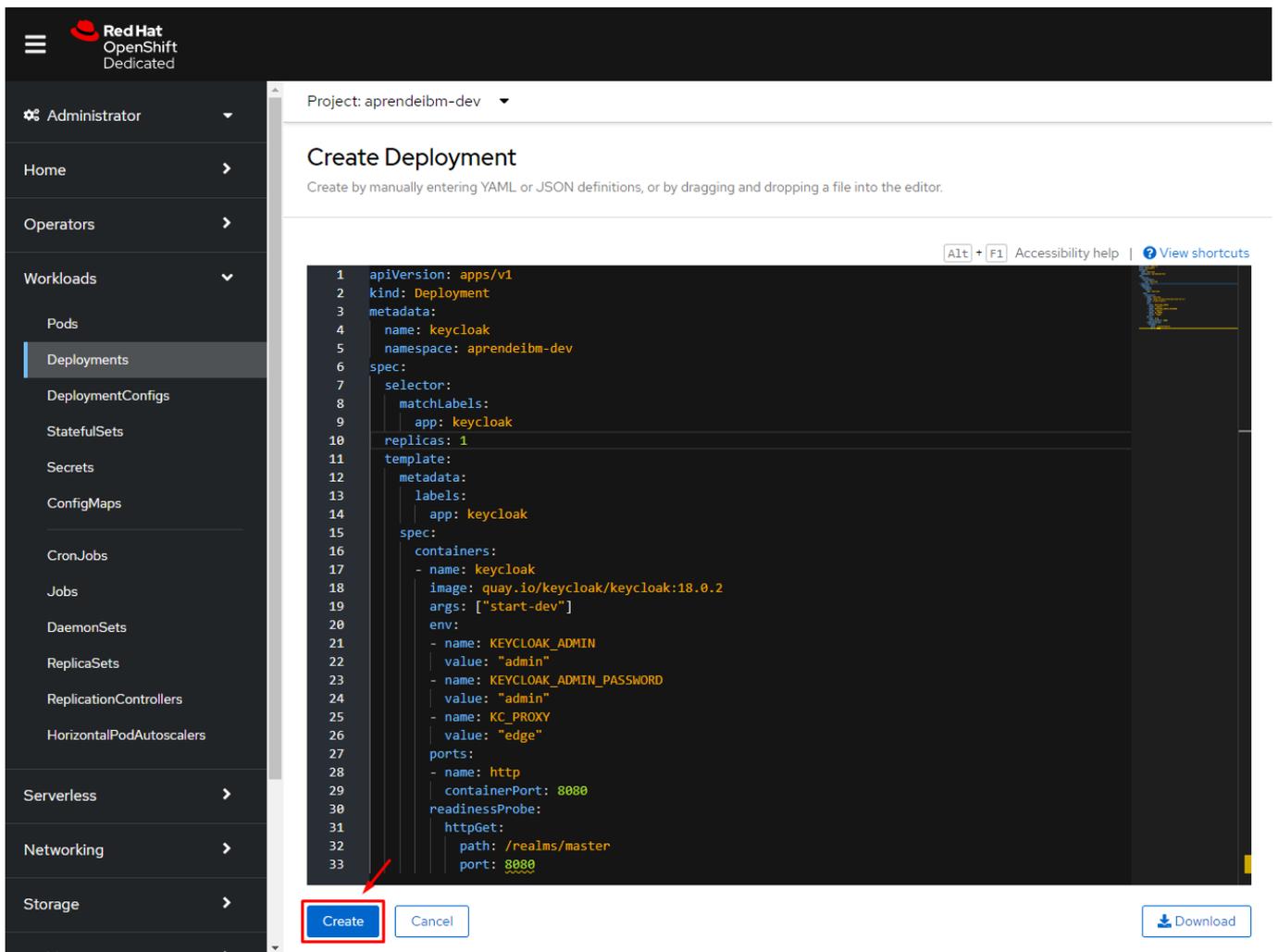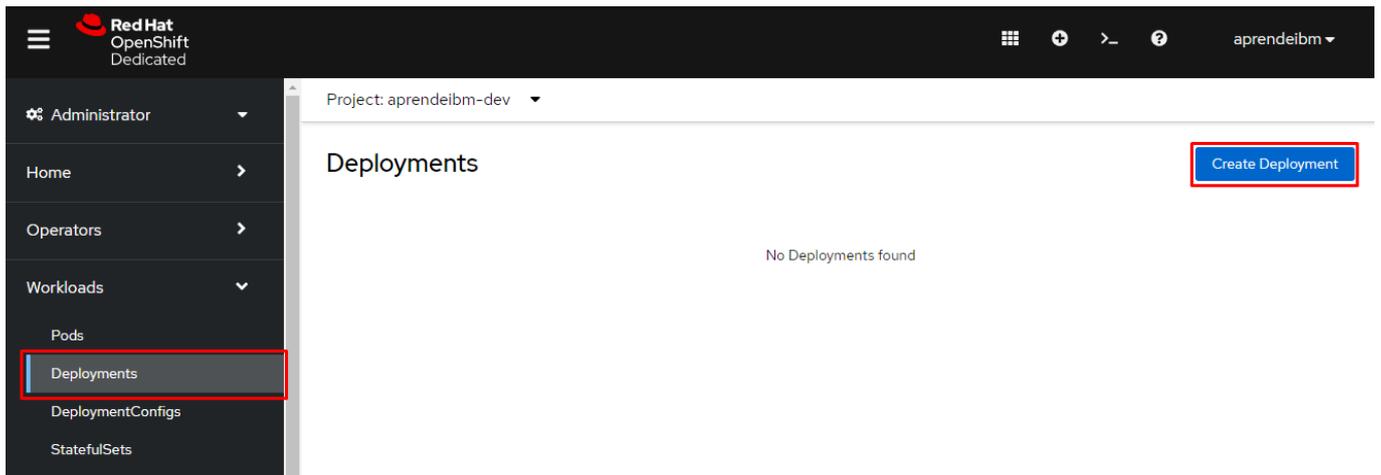
Deployments > Deployment details

D keycloak

Details   Metrics   YAML   ReplicaSets   Pods   Environment   Events

Deployment details

0
Scaling to 1

Name
keycloak

Namespace
NS aprendeibm-dev

Labels   Edit ✏
No labels

Update strategy
RollingUpdate

Max unavailable
25% of 1 pod

Max surge
25% greater than 1 pod

Project: aprendeibm-dev ▼

Deployments

Name ▼   Search by name...   /

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).

```
1    apiVersion: v1
2    kind: Service
3   metadata:
4       name: keycloak
5       namespace: aprendeibm-dev
6   spec:
7       selector:
8           app: keycloak
9       ports:
10          - protocol: TCP
11            port: 80
12            targetPort: 8080
```

Project: aprendeibm-dev ▾

Services > Service details

Ⓢ keycloak

Actions ▾

Details   YAML   Pods

Service details

Name
keycloak

Namespace
NS aprendeibm-dev

Labels   Edit ✎

No labels

Pod selector
🔍 app=keycloak

Annotations
0 annotations ✎

Session affinity
None

Service routing

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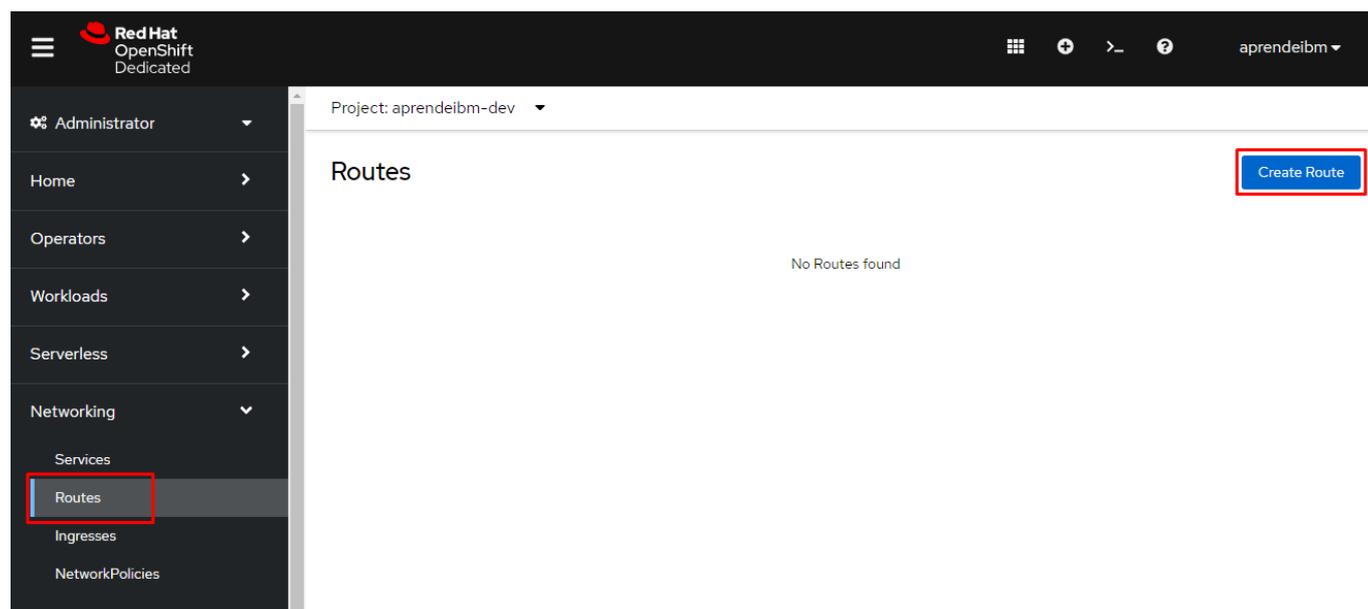keycloak.aprendeibm-dev.svc.cluster.local
Accessible within the cluster only

Service address

Type		Location
Cluster IP		172.30.108.171
Accessible within the cluster only		

Service port mapping

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

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

Red Hat OpenShift Dedicated

⚙️ **Administrator** ▾

Home ›

Operators ›

Workloads ›

Serverless ›

Networking ▾

   Services

   Routes

   Ingresses

   NetworkPolicies

Project: aprendeibm-dev ▾

Routes   Create Route

No Routes found

aprendeibm ▾

Project: aprendeibm-dev ▼

Create Route

Edit YAML

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 *

⑤ 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

▼ Filter ▾   Name ▾   Search by name...   /

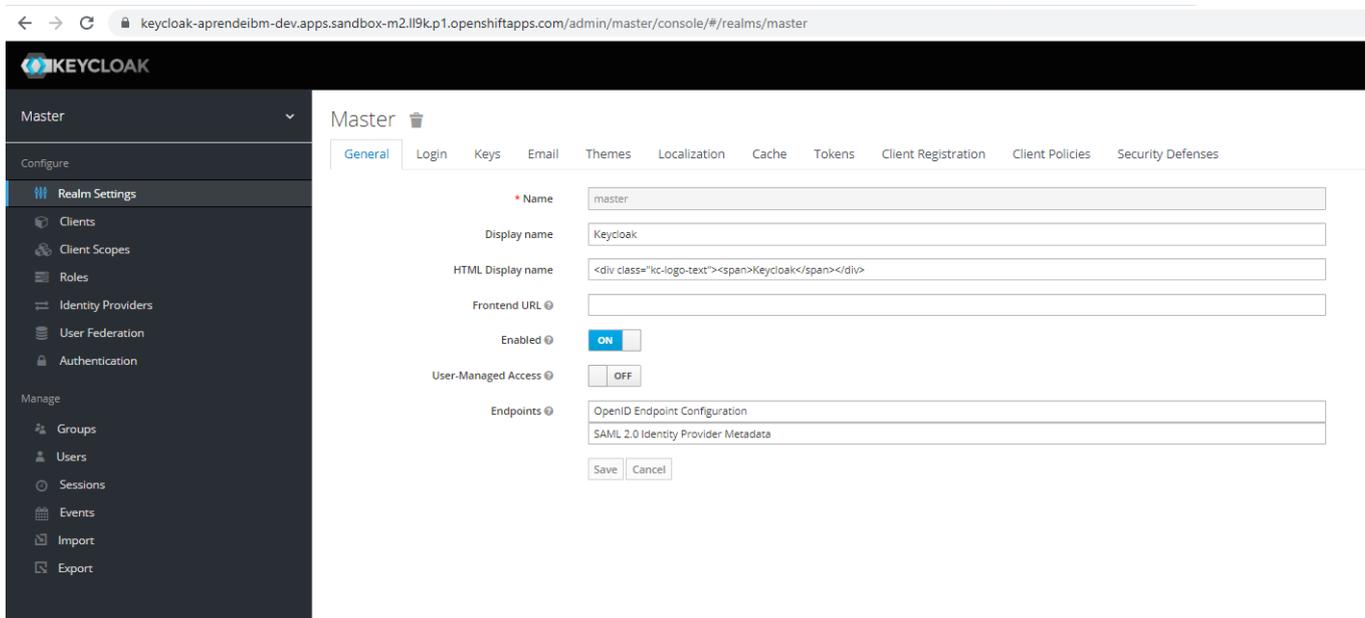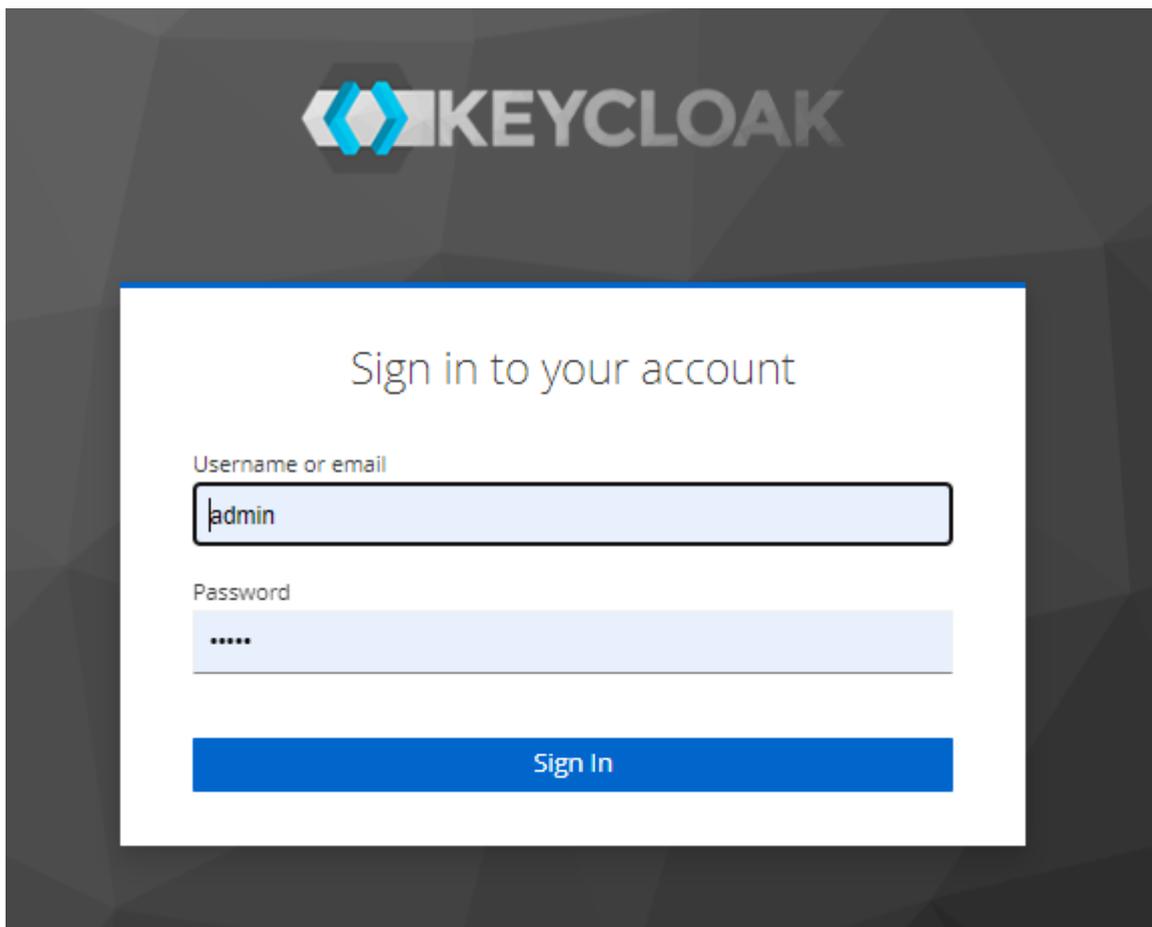| Name	Status	Location	Service
RT keycloak	✓ Accepted	https://keycloak-aprendeibm-dev.apps.sandbox-m2.ll9k.p1.openshiftapps.com ⬈ 📋	**S** keycloak

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

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

KEYCLOAK

Welcome to **Keycloak**

👥 Administration Console ›	📄 Documentation ›	⚙ Keycloak Project ›
Centrally manage all aspects of the Keycloak server	User Guide, Admin REST API and Javadocs	
		✉ Mailing List ›
		🐞 Report an issue ›

JBoss Community

ArquitecturaIBM Consulting