neueda

futureproof your workforce

DATA PERSISTENCE

NEUEDA.COM

Objectives

- Data Persistence in OpenShift
- Persistent Volumes using NFS
- Persistent Volume Claims
- OpenShift Volumes
- Data Persistence Using Databases
- OpenShift Databases

Data Persistence in OpenShift

- In our previous lab we were able to deploy an application, but any data we put in was lost when the app updated.
- Actually, by design Kubernetes Pods are *ephemeral*.
  - Any given Pod could be replaced at any time.
- However persistent state is a critical aspect of many applications.

Example: Pods are Ephemeral

- Let’s look at the Image Uploader deployment again
- Upload an image and see how it displays in the website
- Now scale the application down to 0 pods, then scale back up to 1 pod
- Open the URL again - no image!
- What happened? Pods (and their memory) are ephemeral
    - When the Pod is deleted, all its memory goes with it.

Handling Permanent Data

- OpenShift makes persistent storage available for
  - Data that is shared between Pods
  - Data that needs to persist past the lifetime of any particular Pod
- Permanent storage in pods is handled using **Persistent Volumes** (PVs)
- This can be supplied from a variety of storage solutions
  - NFS, HostPath (local directories), AWS EBS, Azure Disk, etc

Persistent Volumes using NFS

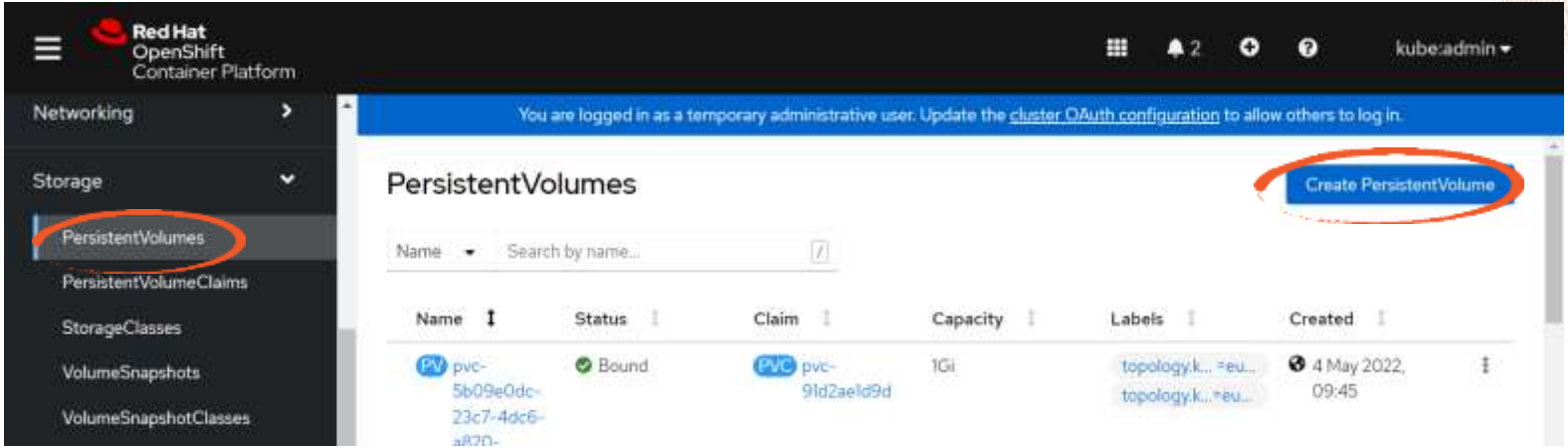
- In OpenShift the process of creating persistent file storage has three steps:

1. Create a Persistent Volume (these are cluster-wide resources)
2. Create a **Persistent Volume Claim** (PVC) in your project.
3. Mount the persistent storage as a **Volume** into your application's Pods

Creating a Persistent Volume

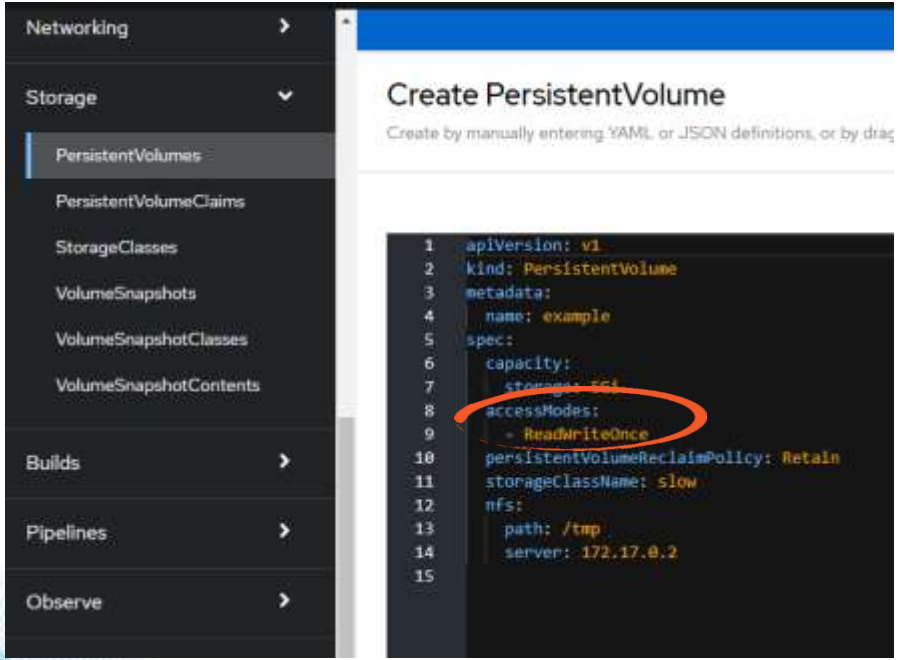
- Persistent Volumes are cluster-wide resources and really should only be allocated by the Cluster Administrator
- Go to the Administrator perspective in your Web Console and click on the Storage item.
  - See there is no “PersistentVolumes” option.
- In the Cluster Admin console you can see that option is available.

Creating a Persistent Volume



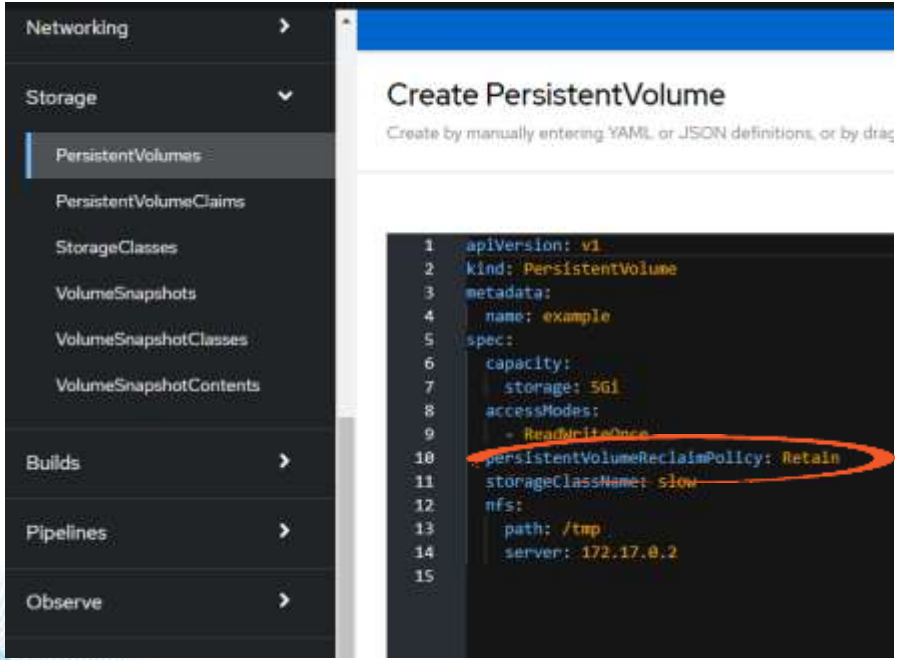
neueda
future proof your workforce

Configuring a PersistentVolume: accessMode



- The **accessMode** can be
  - Read/Write once (RWO) - Can be mounted as read/write by a single node in the cluster
  - Read-only many (ROX) - Can be mounted as read-only by many nodes
  - Read/Write many (RWX) - Can be mounted as read/write by multiple nodes in the cluster.

Configuring a PersistentVolume: Reclaim Policy



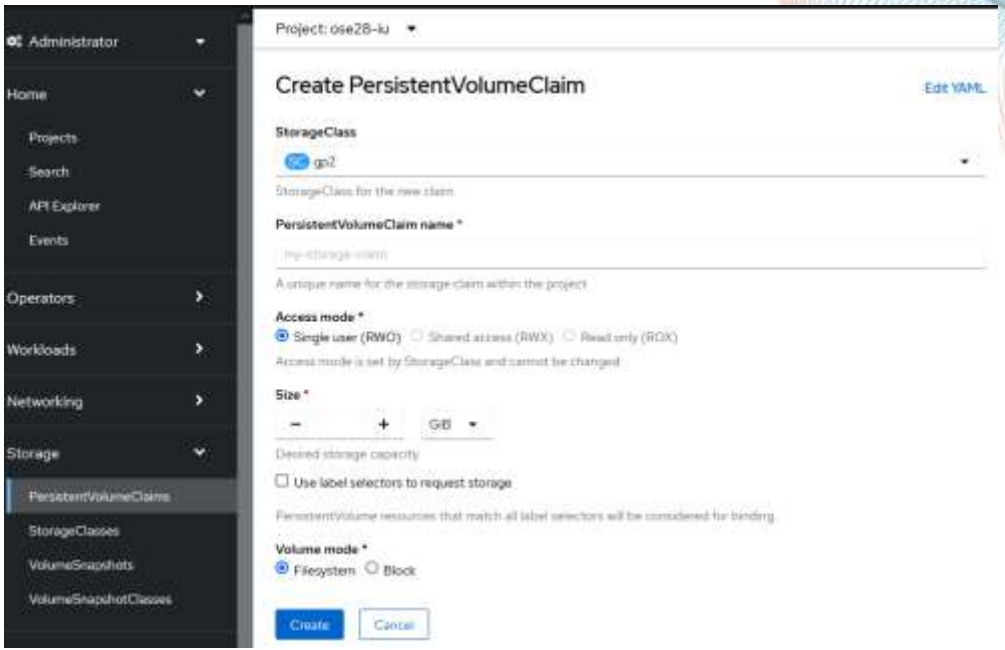
- The reclaim policy dictates how a PV handles reclaiming space after a storage claim on the PV is no longer required. It can be
  - Retain - all data is retaining in the PV, you have to reclaim space manually
  - Recycle - all data is automatically removed when the claim is deleted.

Using Persistent Storage

- To take advantage of a PV, you need to make a claim on that PV.
- PVs represent available storage, and PVCs represent an application's need for that storage.
- When you create a PVC, OpenShift looks for the best fit among the available PVs and reserves it for use by the PVC.
  - PV size vs PVC need - use the smallest available PV
  - Access Mode - use a PV with the same or greater access privileges than that required by the PVC

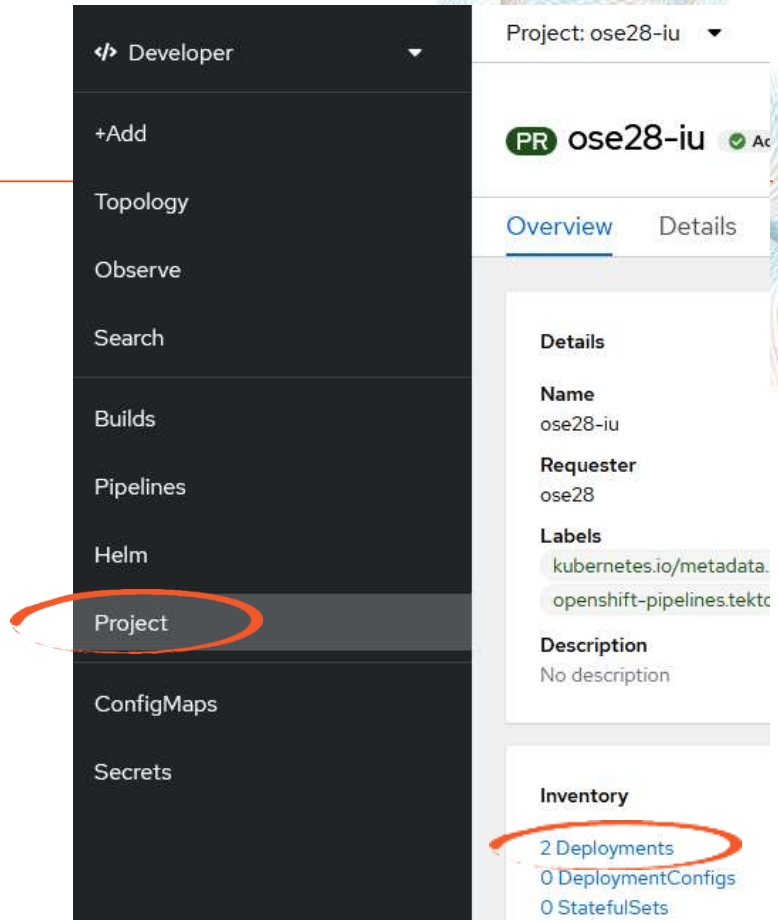
Adding a Persistent Volume Claim

- PersistentVolumeClaims can be created by accessing the PersistentVolumeClaims item under storage in the Administrator perspective

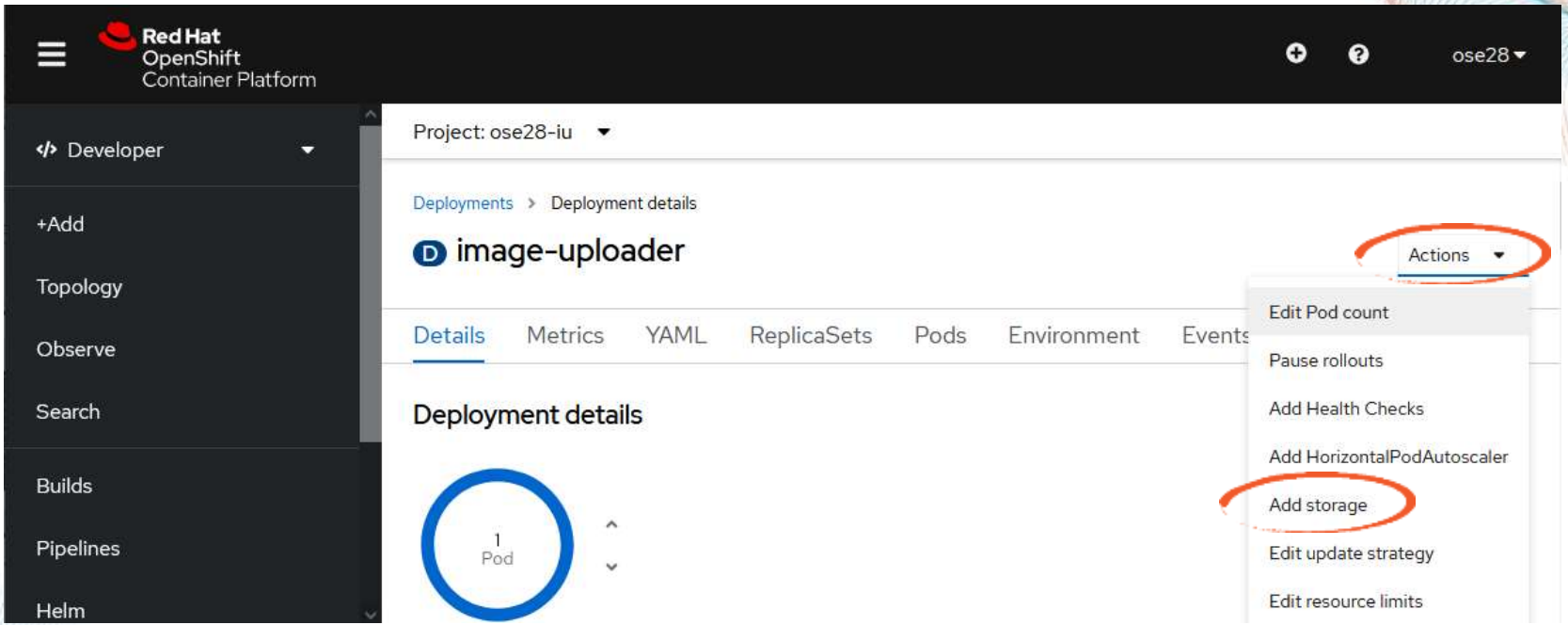


OpenShift Volumes

- A *volume* is any filesystem, file or data mounted into an application's Pods to provide persistent data.
- To attach a PVC as a volume in our app using the Web Console you need to get to the project overview page and select the Deployment.



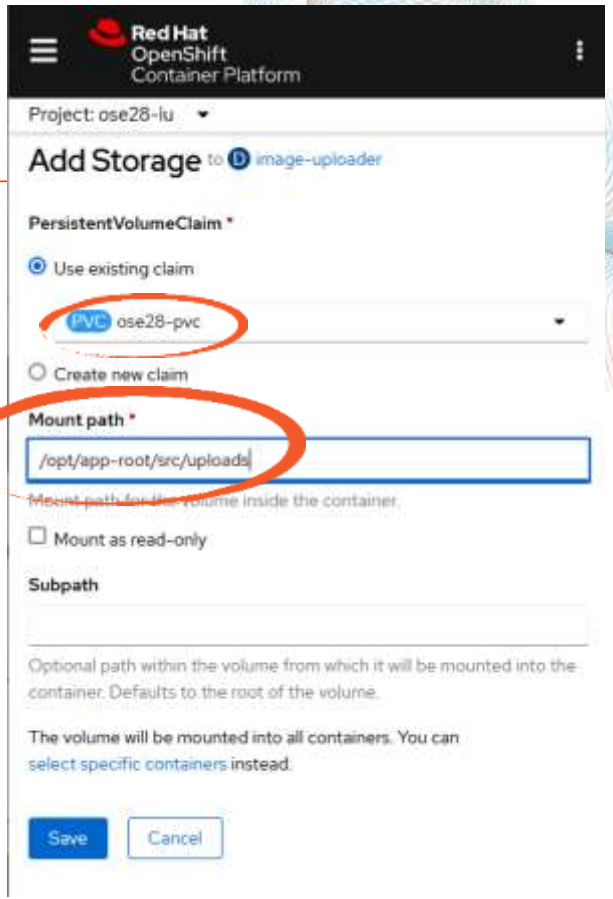
Creating a Volume



neueda
futureproof your workforce

Creating a Volume

- Just select your existing PVC
- The Mount Path is the path within each pod at which the persistent volume should be available - it is application dependent.
- On “Save” the volume is attached.



neueda
future proof your workforce

Lab 3: Add Persistent Data to the Image Uploader

- This is a quick lab to implement what we have just seen in your own project.
- Go to /labs/3-image-uploader-persistent-storage.md

Volumes Separate Data

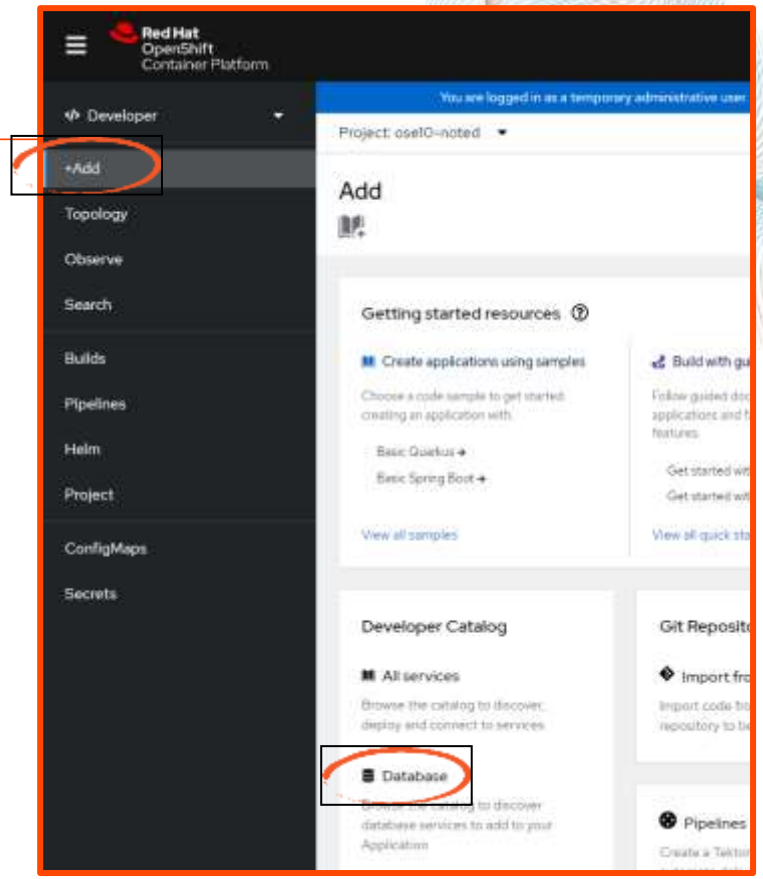
- Suppose everyone set up Volumes on our cluster - why don't you see everyone else's data?
  - Each application deployment uses its own NFS volume to store data.
  - Each NFS volume is then mounted into its own application's mount space.
  - So the data is always separated on the cluster.

Data Persistence Using Databases

- The Persistent Volume technology is only one way to data persistence.
- You can also hook your application to a database
  - If the database is not running on your cluster, then it supplies its own data persistence
  - If the database is running on your cluster, then data persistence is achieved behind the scenes.

OpenShift Databases

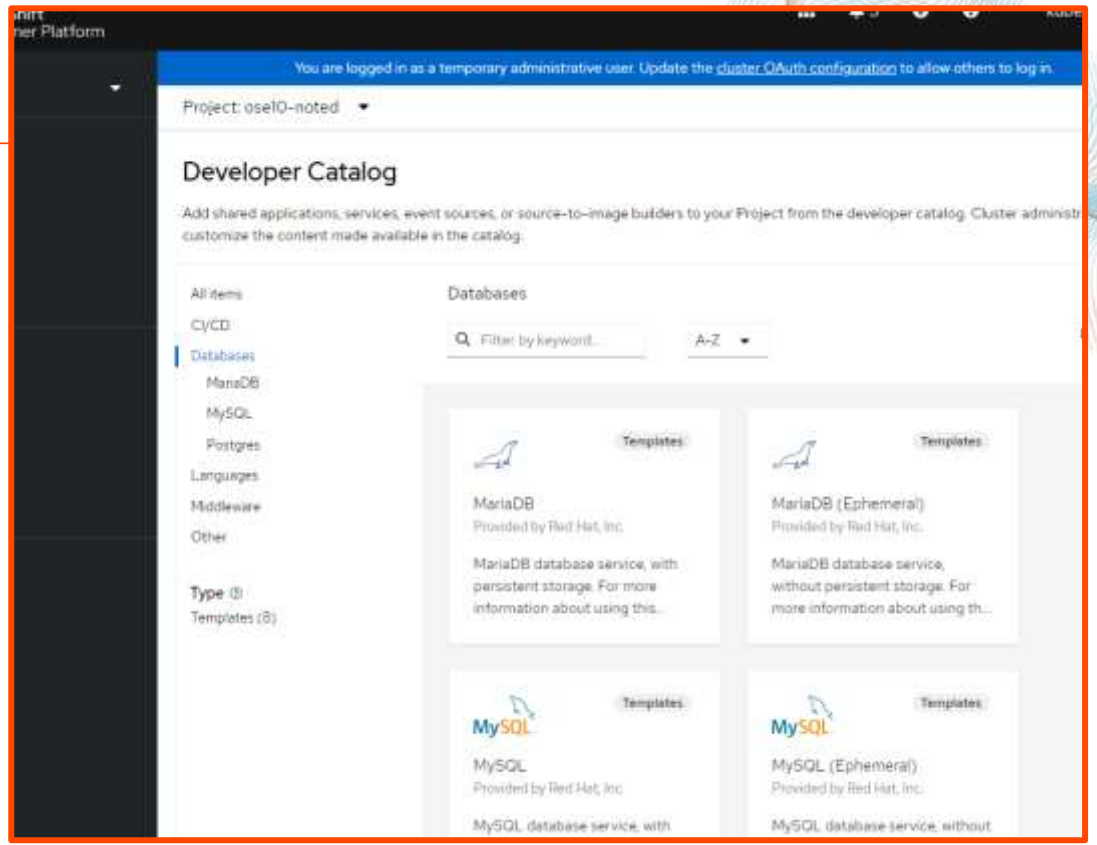
- OpenShift has pre-configured MySQL, MariaDB and PostgreSQL databases available in the Developer Catalogue



neueda
futureproof your workforce

OpenShift Databases

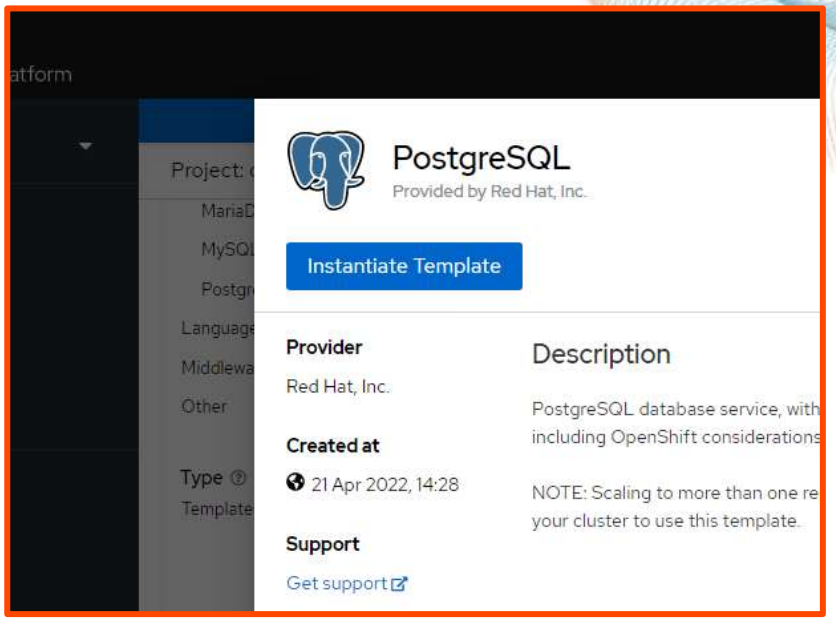
- The databases all offer ephemeral (for testing) and persistent options.



neueda
future proof your workforce

OpenShift Databases

- During Instantiation you can supply a username, password and other configuration details.
- If you use a template you will need to manually configure your app to inject the environment variables needed to connect.



neueda
future proof your workforce

Demo: Petclinic

- In this demo we will deploy a Spring Boot app and a MySQL database on the same cluster and link the two.

Connecting to Databases

- The database deployment we saw is quite manual, OpenShift has a number of more automatic connection methods.
- One of the simplest of these is using a Secrets object.
- Secrets objects store one or more values that are intended to be obscured (like passwords, or certificate files).
- Secrets objects can be made available to Pods in a namespace without divulging the contents to users.
- We will see Secrets for connecting a database later on.

Summary

- Data Persistence in OpenShift
- Persistent Volumes using NFS
- Persistent Volume Claims
- OpenShift Volumes
- Data Persistence Using Databases
- OpenShift Databases

Questions and Comments?

neueda
futureproof your workforce