Contents

[Prerequisite 1](#_Toc35417707)

[Create a job base image 1](#_Toc35417708)

[Create Persistent Volume Claim 1](#_Toc35417709)

[Create a schedule job to backup 3](#_Toc35417710)

[Restore database from a SQL file 8](#_Toc35417711)

# Prerequisite

A site (with a database) was already set up (and running) in the same AWS Openshift project.

Go to the project you want to by the command line: **oc project <project name>**

(in this sample, **oc project dn-story-test**)

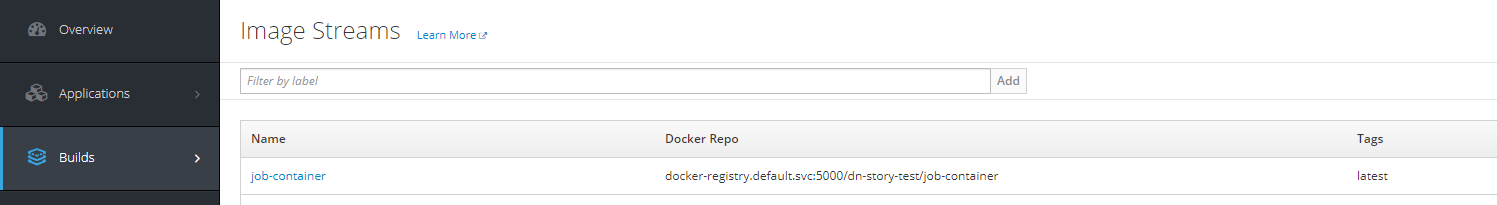
# Create a job base image

1. Create an image using build config via command line:

**oc apply -f <path to template folder>\base\_container\_job\_dn-story-test.yaml**

Then build it in Openshift or from command line

**oc start-build dn-story-container –n dn-story-test**



(Or hit the “Start Build” button on the UI of Openshift to build)

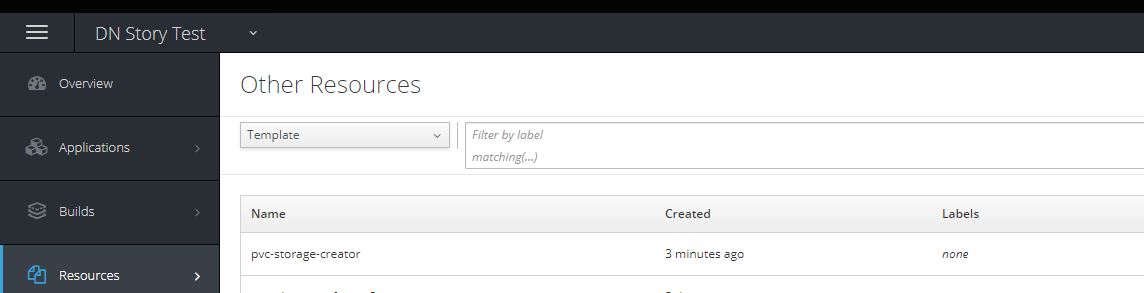
# Create Persistent Volume Claim

This PVC will store backup files of the scheduled job.

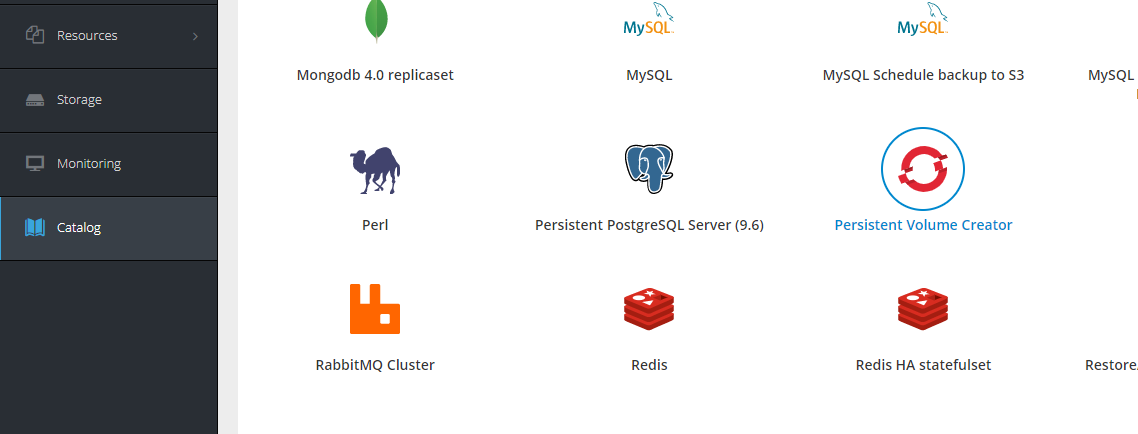
1. Create PVC using yaml via command line:

**oc apply -f <path to template folder>\pvc\_creator.yaml**

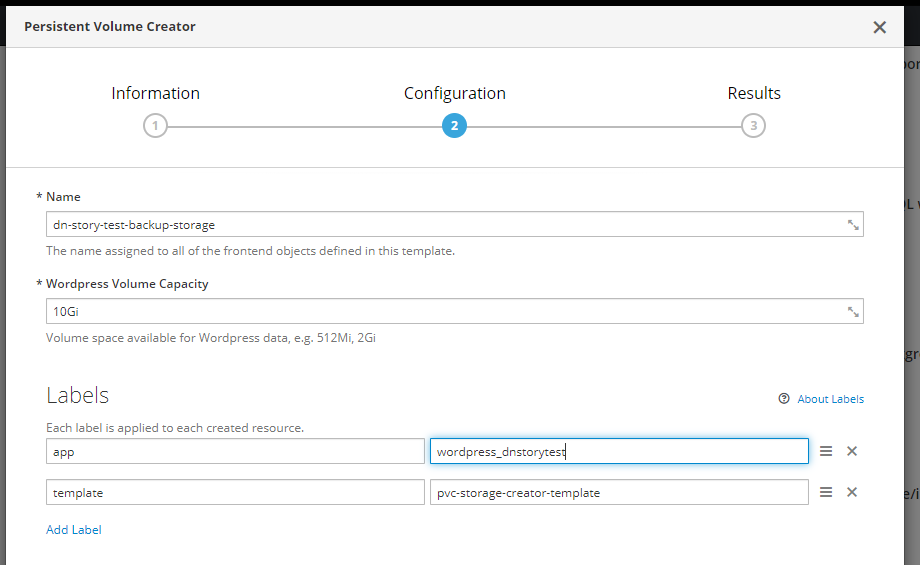
It should then appear in the template resources list,



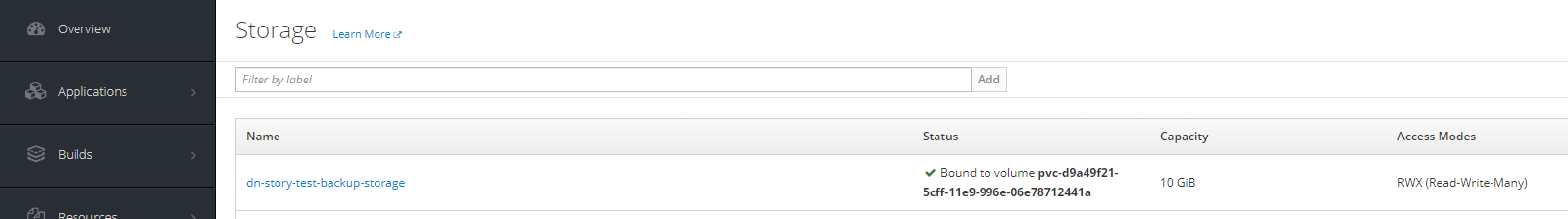
and in the catalog list



1. Click on the template in the Catalog, fill in the info and start experiencing



1. After created, we can see it in the Storage tab



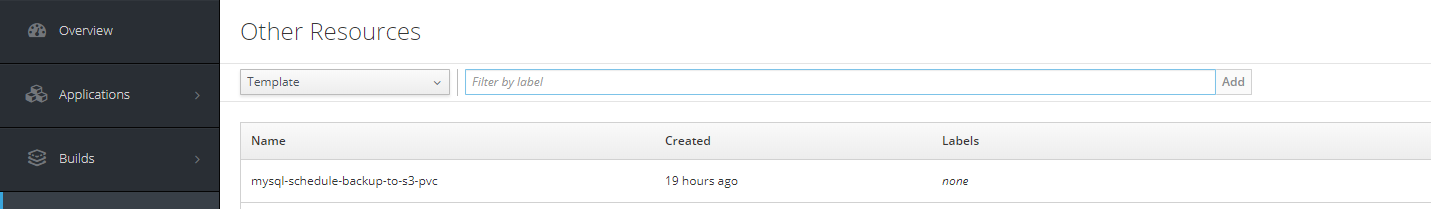
You can also confirm that the PVC was successfully created by running: **oc get pvc**

# Create a schedule job to backup

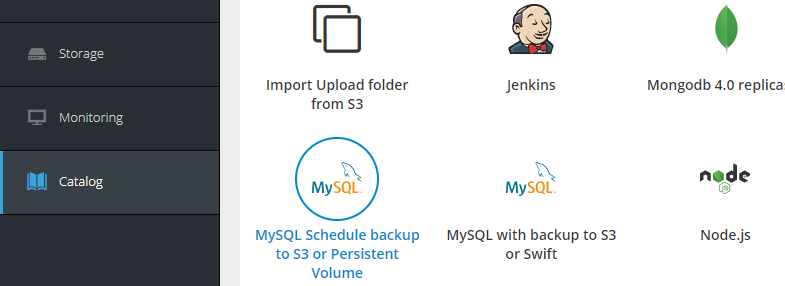
1. Create resources using template via command line:

**oc create -f <path to template folder>\mysql\_cronjob\_backup\_s3\_pvc.yaml**

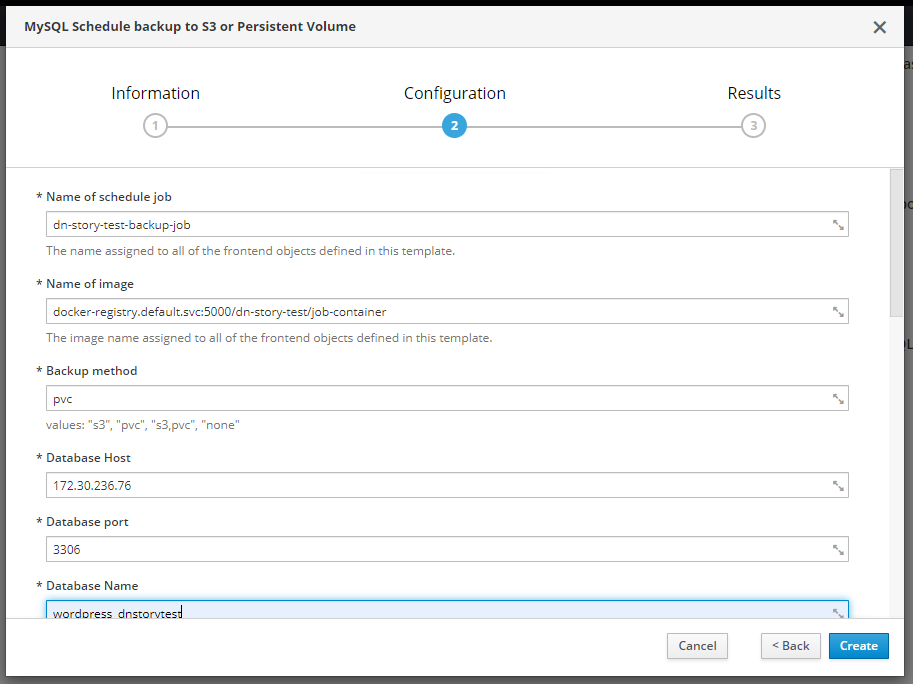
It should then appear in the template resources list.

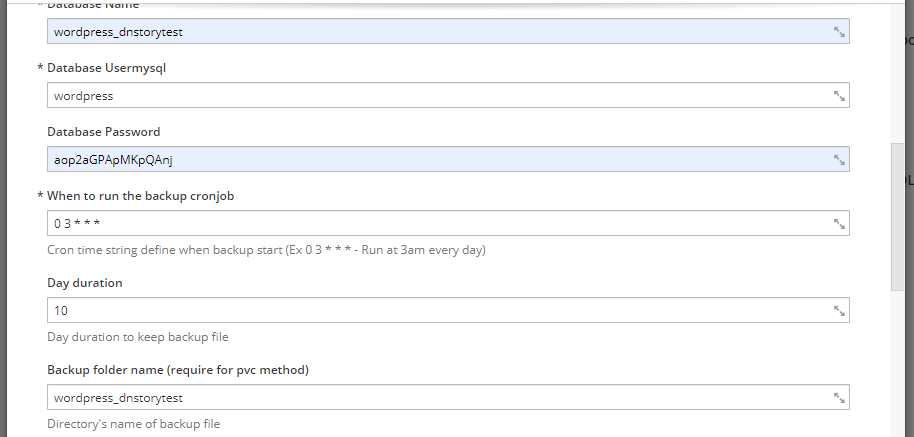


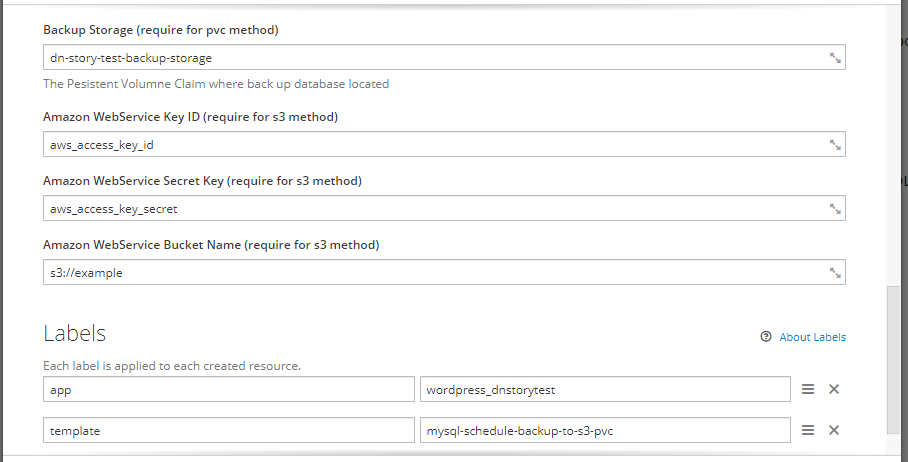
And catalog list

****

1. Click on the template in the Catalog, fill in the info and start experiencing



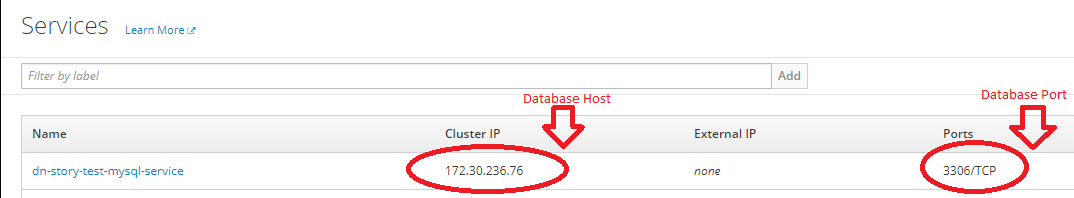




* ***Name of image***: This (image) has been built in the “Create job base image” section.
* ***Backup method***: This field will provide the method(s) to store the SQL file.
  + S3: Store data in Amazon S3 Service
  + PVC: Store data in Persistent Volume Claim

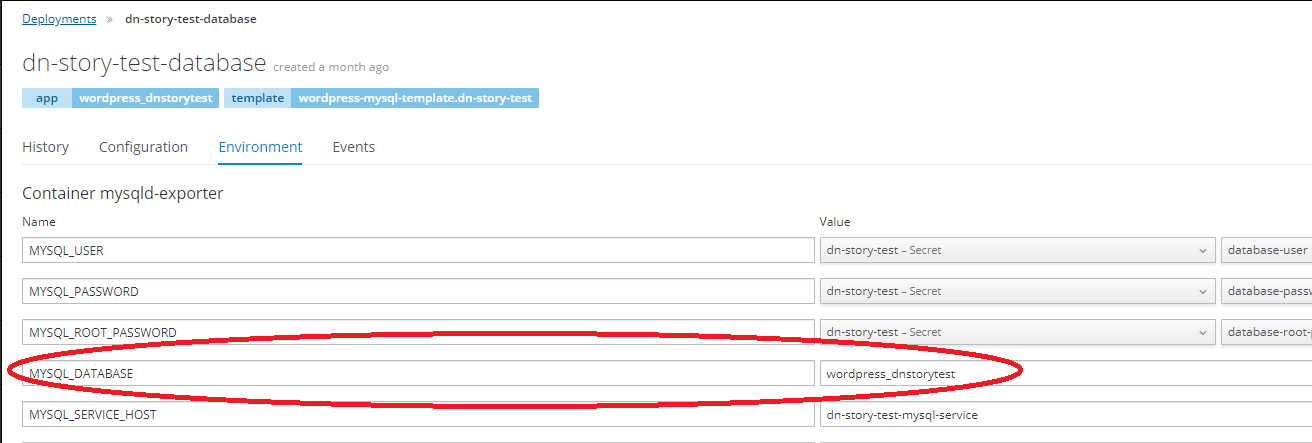
We can fill more than one method for backup (see above screenshot).

* ***Database Host*** and ***Database port*** get from Service which created:

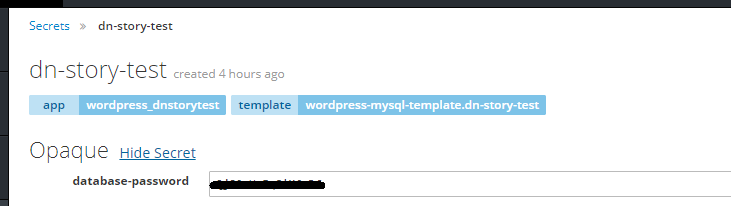


You can also take the Name of the database service to use as Host (instead of IP Address).

* ***Database Name***: You can get the value for this field from Deployments tab (Openshift UI)



* ***Database Password***: You can get the value for this field from Secret tab (Openshift UI)

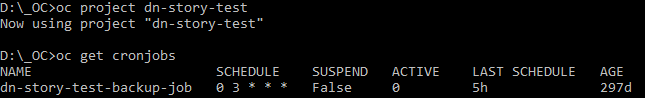


* ***Day duration***: if SQL files are older than this day, the script will remove them.
* ***Backup Storage***: **this field is only required when using PVC method**. This value is the name of storage which created from “Create Persistent Volume Claim” section
* ***Backup folder name***: this is a folder where SQL files are stored after backup.
* ***Amazon WebService Access Key ID & Amazon WebService Secret Key***: these fields are the values provided to access AWS S3 services. Reference:

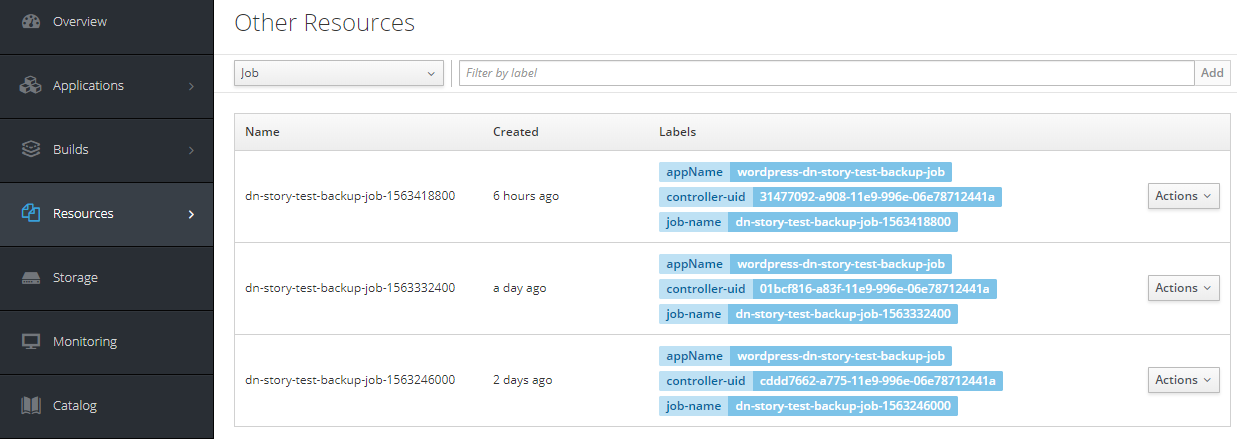
<https://docs.aws.amazon.com/general/latest/gr/managing-aws-access-keys.html>

* ***Amazon WebService Bucket Name***: this path is the path of folder that will create backup folder and store SQL file into.

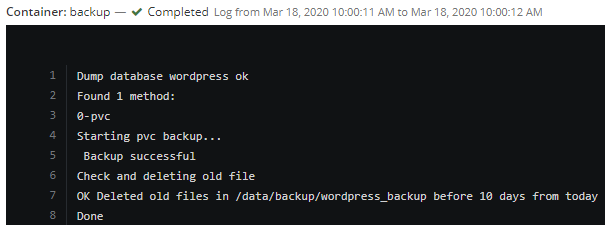
1. If the job was created successfully, you will be able to see it in the list by running the following command: **oc get cronjobs**



And you will be able to see the job instances later on after it runs (after the scheduled time).



Check log of the pod, it should say the database backup was done successfully.



1. Advanced: If by a mistake you created a cronjob and want to delete it, go for command: **oc delete cronjob/<cron\_job\_name>**

*Reference:* <https://docs.openshift.com/container-platform/3.5/dev_guide/cron_jobs.html>

1. If you have a demand to see the created backup files, or even copy them to your local workstation, you need to mount the PVC to a pod (a new pod or an existed pod).

*Reference:* <https://docs.openshift.com/container-platform/3.5/install_config/storage_examples/privileged_pod_storage.html> *(“Define your pod to access the claim”)*

To copy a pod directory to a local directory, use **rsync** command:

$ oc rsync <source> <destination> [-c <container>]

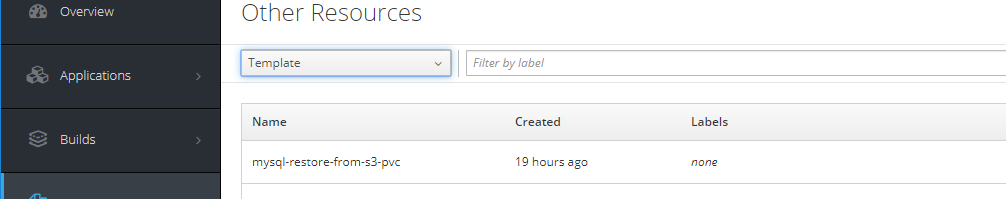
*Reference:* <https://docs.openshift.com/container-platform/3.3/dev_guide/copy_files_to_container.html>

# Restore database from a SQL file

1. Create resources using template via command line:

**oc create -f <path to template folder>\mysql\_restore\_from\_s3\_pvc.yaml**

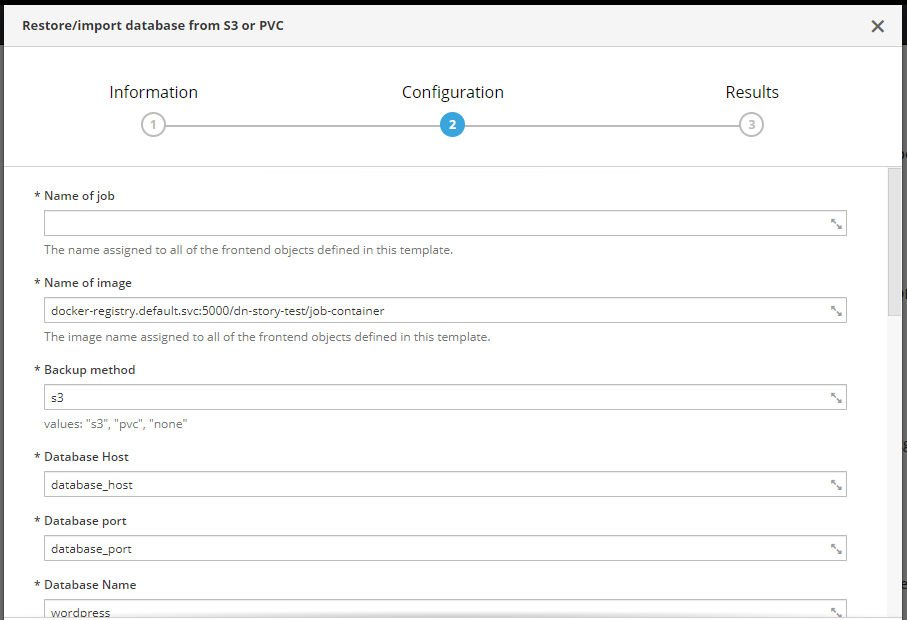
It should then appear in the template resources list.

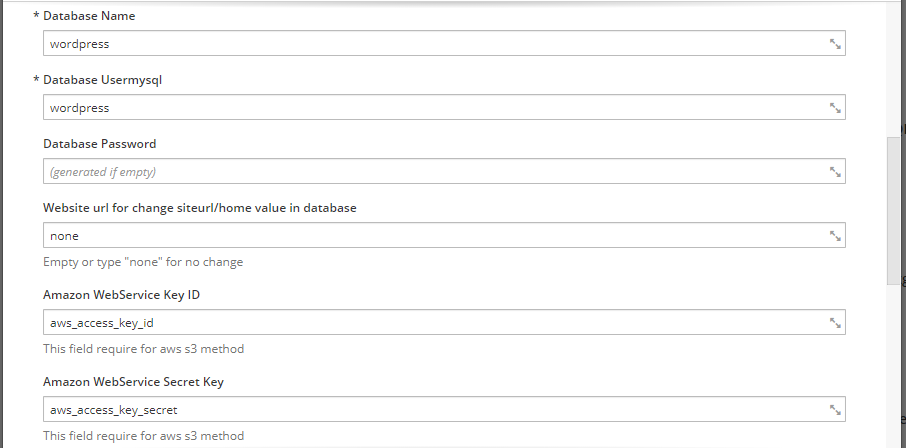


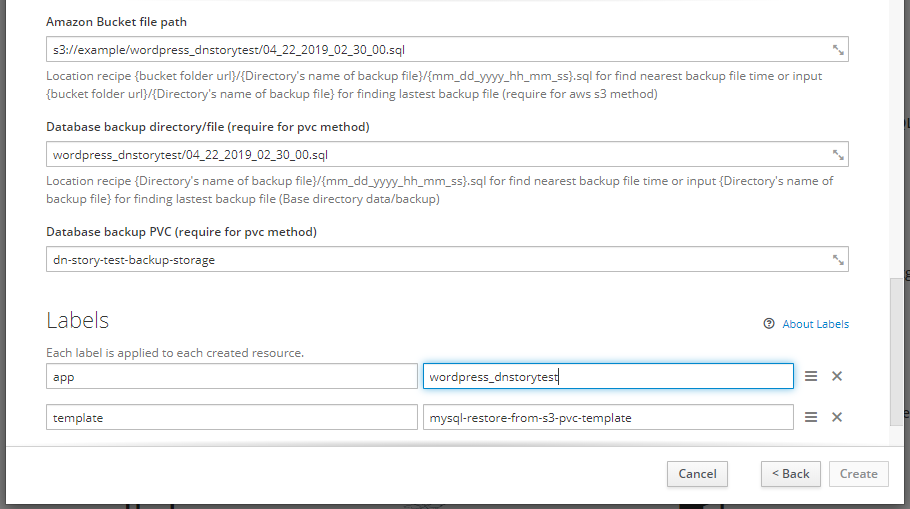
And catalog list



1. Click on the template in the Catalog, fill in the info and start experiencing







* ***Name of image***: This (image) has been built in the “Create job base image” section.
* ***Backup method***: This field will provide the method to store the SQL file.
  + S3: Store data in Amazon S3 Service
  + PVC: Store data in Persistent Volume Claim

NOTE: This job only accepts only one method per execution

Restore options:

* Restore with Persistent Volume Claim method
  + ***Backup Storage*** *(database backup PVC)*: The value is the name of storage created from the “Create Persistent Volume Claim” section.
  + ***Database backup directory/file:***

Restore a specific SQL file:fill this field with the full path of the SQL file you want to restore. If this file cannot be found, the script will find the nearest (by date) SQL file based on the file name.

Restore the latest file***:*** fill this field with the folder that contains SQL files. The script will find the latest (by date) file in this directory to restore.

* Restore with Amazon S3 method
  + ***Amazon WebService Access Key ID & Amazon WebService Secret Key***: these fields are the values provided to access AWS S3 services. Reference:

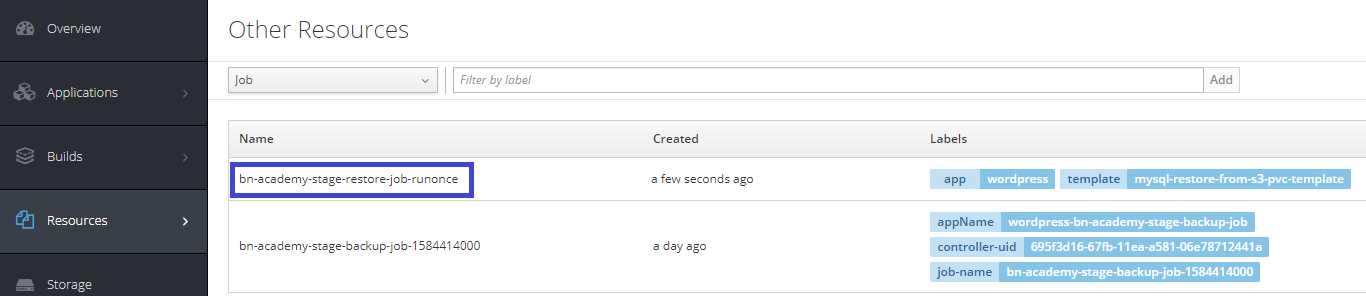
<https://docs.aws.amazon.com/general/latest/gr/managing-aws-access-keys.html>

* + ***Amazon Bucket file path***:

Restore a specific file: fill this field with the full path of the SQL file you want to restore. If this file cannot be found, the script will find the nearest (by date) SQL file based on the file name.

Restore the latest file: fill this field with the folder that contains SQL files. The script will find the latest (by date) file in this directory to restore.

1. You should be able to see a job created and run successfully



Check the log of the pod, it should say that the database was restored successfully.

