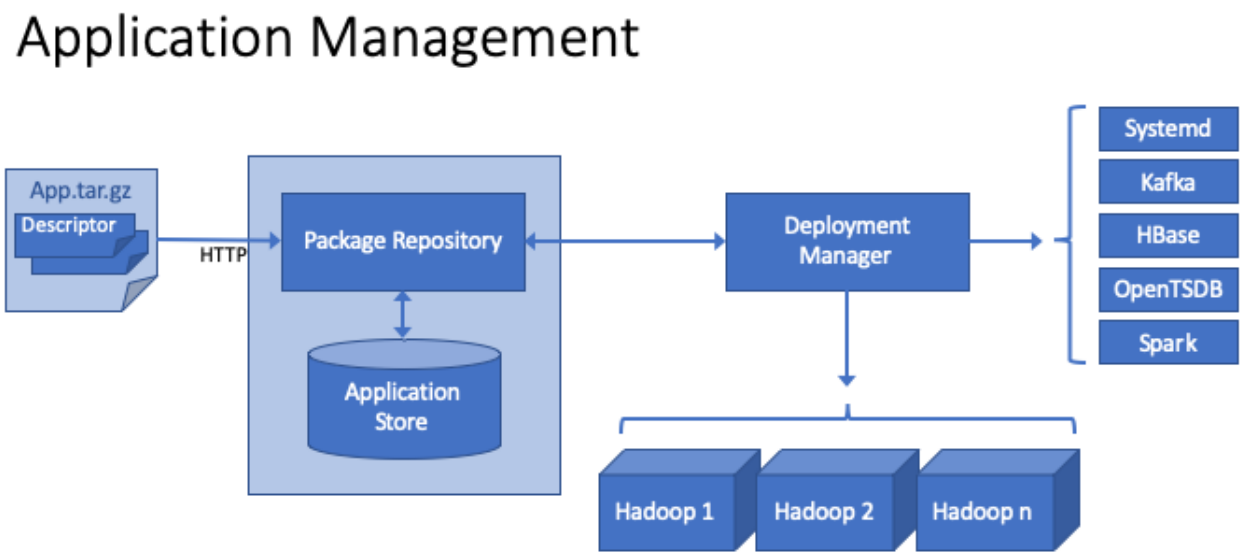
deployment-manager

PNDA Applications

Applications are managed on PNDA by two main components:

- Package Repository – provides an API for uploading or deleting application packages from the system. The packages are stored in a Kubernetes persistent volume.
- Deployment Manager – provides an API for deploying applications from the package repository to the cluster.

Information about PNDA application management in PNDA 5.0 can be found here: http://pnda.io/pnda-guide/applications/



Deployment Manager

The deployment manager is responsible for deploying apps into the PNDA cluster and configuring all components of the cluster, using descriptor files in the application .tar file. The deployment manager currently was quickly ported to Kubernetes, with sufficient changes to allow the PNDA console to work.

Deployment-manager defines several plugins to deploy different types of components in an application: oozie, sparkStreaming, flinkStreaming, jupyter.

Currently the only plugin modified to deploy into Kubernetes is jupyter.

Jupyter

the old jupyter plugin distributes jupyter notebooks in applications by ssh copy the files into the jupyter server. Since cloud-native PNDA deploys jupterhub-k8s, this method does not work.

Jupyterhub on kubernetes deploys a jupyter-pod (a jupyter singleuser server) for each user that logs into jupyterhub. The pod attache a persistent volume that stores notebooks for the user.

Jupyterhub documentation describe several ways to add files to the user storage here.

We leverage a git server to synchronize notebooks to users. Deployment-manager run a nginx server with fcgi-wrapper and git-http-backend, to provide a git server through http. Repositories are stored in /data/git-repos.

When a user deploys an application (currently only pnda user is enabled) deployment-manager copies the application notebooks in /data/git-repos /jupyter-$username/$applicationName.

Jupyterhub configuration is modified to create a postStart hook that clones the user repo in apps-notebooks folder.

```
c.KubeSpawner.lifecycle_hooks = {
            "postStart": {
              "exec": {
                "command": [
                  "sh",
                  "-c",
                  "rm -rf /home/jovyan/apps-notebooks && git clone http://deployment-manager-git:8099
/jupyter-${JUPYTERHUB_USER} /home/jovyan/apps-notebooks"]
              }
            }
          }
```

Testing the deployment of jupyter notebooks

To test the deployment of jupyter-notebooks we have created an example app in example-pnda-apps/sparkstreaming-notebook-example-0.1.0.tgz.

To upload the package to the cloud-pnda platform:

It is possible to temporarily expose the package repository API with kubectl port forwarding:

```
kubectl -n pnda port-forward service/pnda-package-repository 8888
```

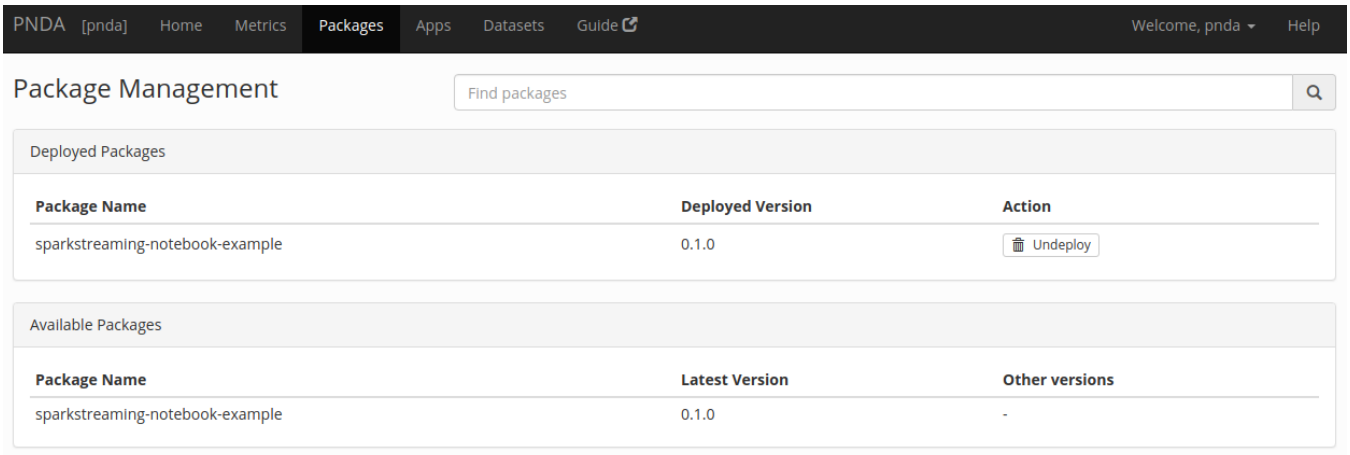
Then you can use curl to post the package to the package repository:

```
curl -XPUT "http://localhost:8888/packages/sparkstreaming-notebook-example-0.1.0.tgz?user.name=root" --data-
binary "@example-pnda-apps/sparkstreaming-notebook-example-0.1.0.tgz"
```

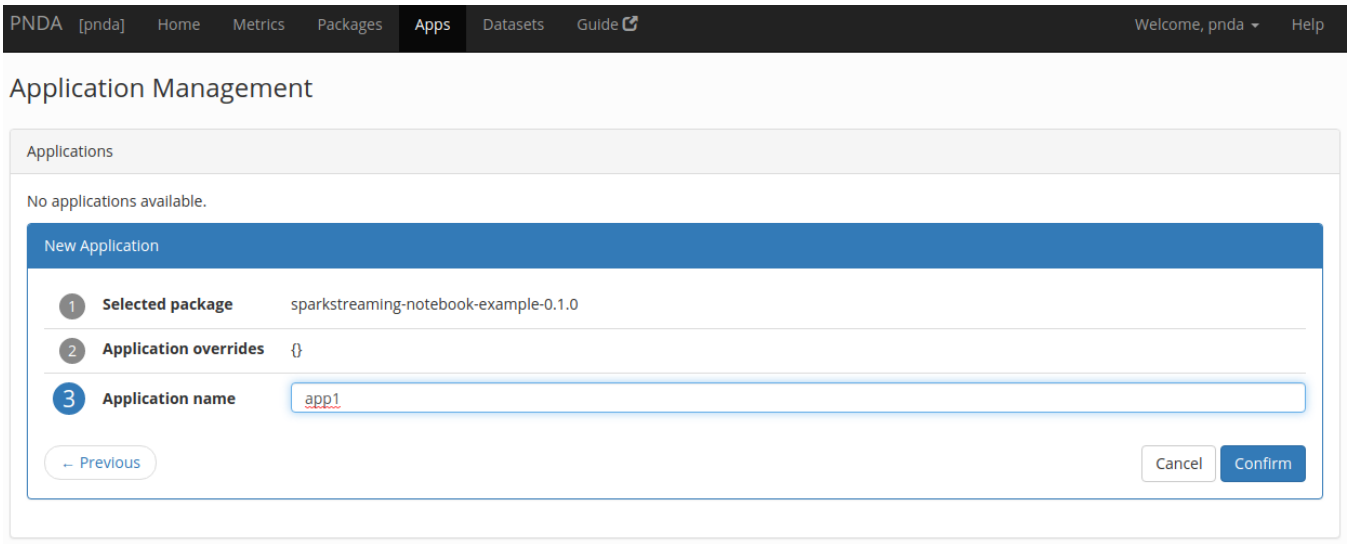
Then you can verify the upload by fetching the list of packages:

```
curl "localhost:8888/packages?user.name=root"
[{"name": "app", "latest_versions": [{"version": "0.1.0", "file": "sparkstreaming-notebook-example-0.1.0.
tgz"}]}]
```

Then you can deploy the package in console UI Package Tab:



Then install the application in the Apps Tab:



Then access jupyterhub with the pnda user and check the notebooks are available in the app1 folder: