HPE Container Platform – Kubernetes Application Image Development

Lab 5: Convert Standard EPIC Image to KubeDirector App (H2O) via Web Terminal

Module 5 “Convert Standard EPIC Image to KubeDirector App (H20) via Web Terminal” 2

Task 1 “Build H20 Docker Image“ 2

Create Dockerfile 2

Build Docker Image 2

Push Docker Image to Docker Hub 2

Task 2 “Deploy KubeDirector Application (H20) in HPE Container Platform” 4

Login 4

Create KubeDirectorApp file 4

Launch & access the application 6

Task 3 “Deleting Application” 8

# Module 5 “Convert Standard EPIC Image to KubeDirector App (H20) via Web Terminal”

This lab is to convert a standard EPIC image to Kubernetes Application. Using it, pods can be spun up.

## Task 1 “Build H20 Docker Image“

This section is to:

1. Create Dockerfile
2. Build Docker Image
3. Push Docker Image to Docker Hub

### Create Dockerfile

Follow the below procedure to create Dockerfile:

1. SSH to a host
2. Create a directory & navigate to it

[root@controller]# **mkdir KD-H2O**

[root@controller]# **cd KD-H2O/**

1. Copy the required tgz file to the working directory

[root@controller KD-H2O]# **cp /mnt/HPECP-AppImage-K8s/KD-H2O/appconfig.tgz ~/KD-H2O/**

1. Copy the Dockerfile to the working directory

[root@controller KD-H2O]# **cp /mnt/HPECP-AppImage-K8s/KD-H2O/Dockerfile ~/KD-H2O/**

### Build Docker Image

Follow the below procedure to build Docker image

1. Execute the command to build Docker image, it may take appx 5-6min to build (replace <tag> with your name)

|  |  |
| --- | --- |
|  | **NOTE: Use your name as tag. Here, its hpe (docker build -t h2o:hpe .). The dot (.) in the command is to represent the current directory. Here, o is the small alphabet o and not zero.** |

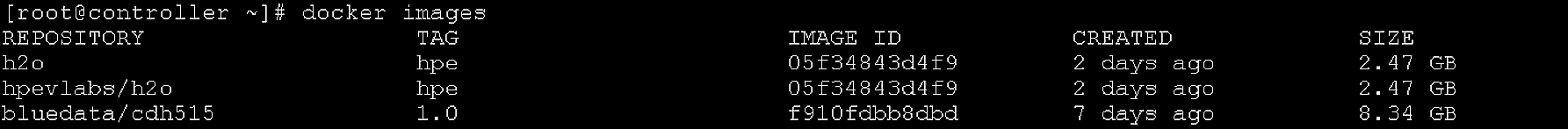
[root@controller KD-H2O]# **docker build -t h2o:<tag> .**

### Push Docker Image to Docker Hub

Follow the below procedure to push Docker image to Docker Hub

1. View the created Docker Image

[root@controller KD-H2O]# **docker images**



1. Login to Docker

[root@controller KD-H2O]# **docker login**

|  |  |
| --- | --- |
|  | **NOTE: Use the following DockerHub credentials:**  **• Username: hpevlabs**  **• Password: Pa$$w0rd2020** |

1. Tag the Docker image

|  |  |
| --- | --- |
|  | **NOTE: hpe is the tag name here, replace it with the name you provided while building Docker image in Build Docker Image section.** |

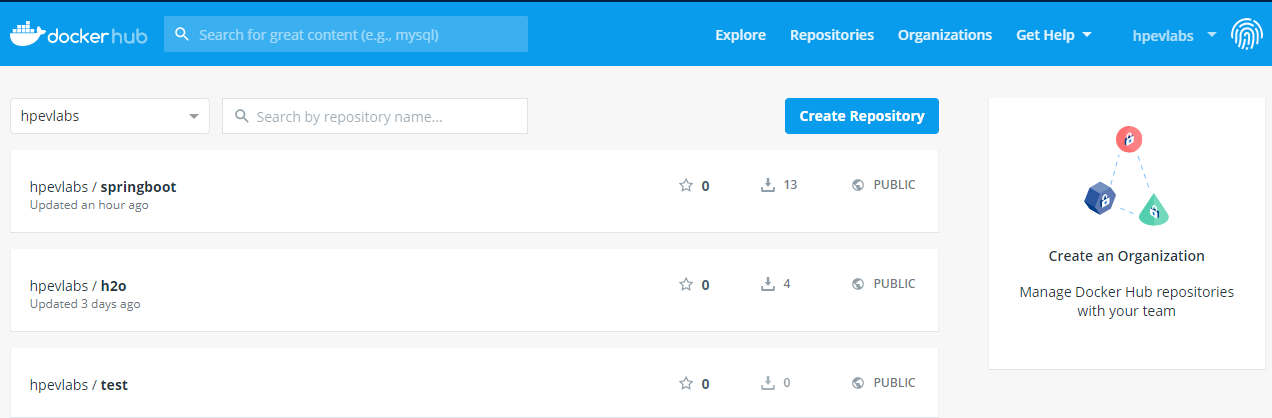
[root@controller KD-H2O]# **docker tag h2o:hpe hpevlabs/h2o:hpe**

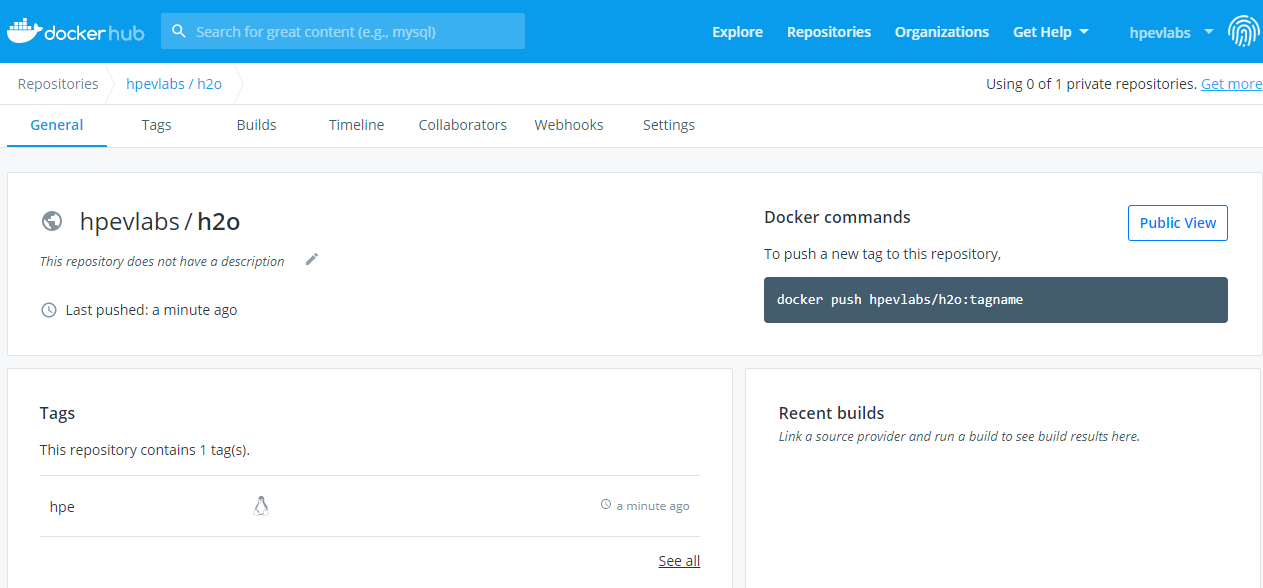
1. Push the Docker image, it may take appx 4-5min to push

|  |  |
| --- | --- |
|  | **NOTE: Restart the Docker daemon if docker push gives HTTPS error. Restart Docker by executing “systemctl restart docker”** |

[root@controller KD-H2O]# **docker push hpevlabs/h2o:hpe**

1. Verify the image in Docker Hub
   1. Open <https://www.docker.com/hpevlabs> on the browser with following credentials
      1. Username: hpevlabs
      2. Password: Pa$$w0rd2020
   2. You will get a list of repository present in your DockerHub account, click on the repository we are using I.e. h2o



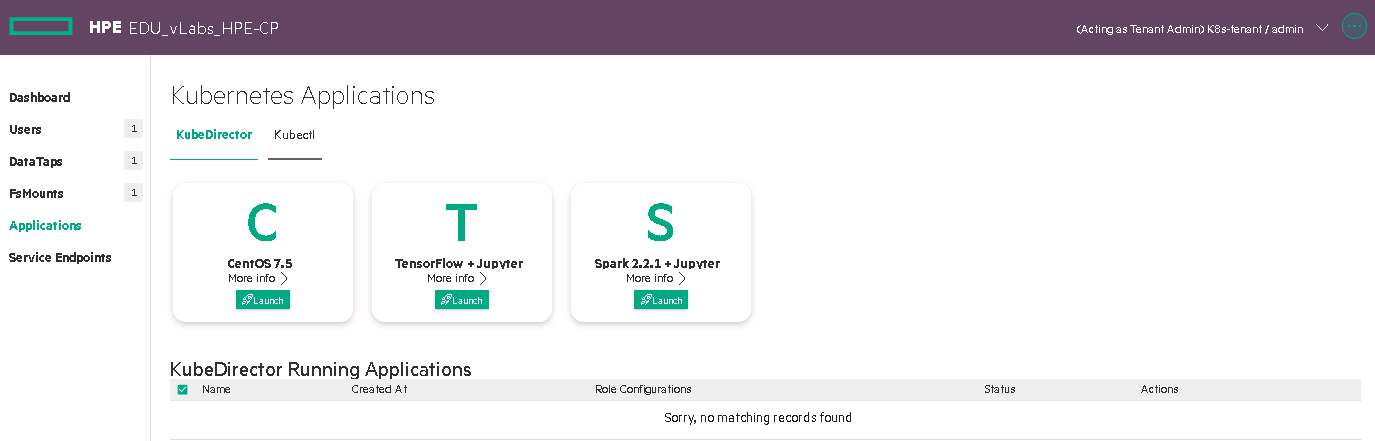


## Task 2 “Deploy KubeDirector Application (H20) in HPE Container Platform”

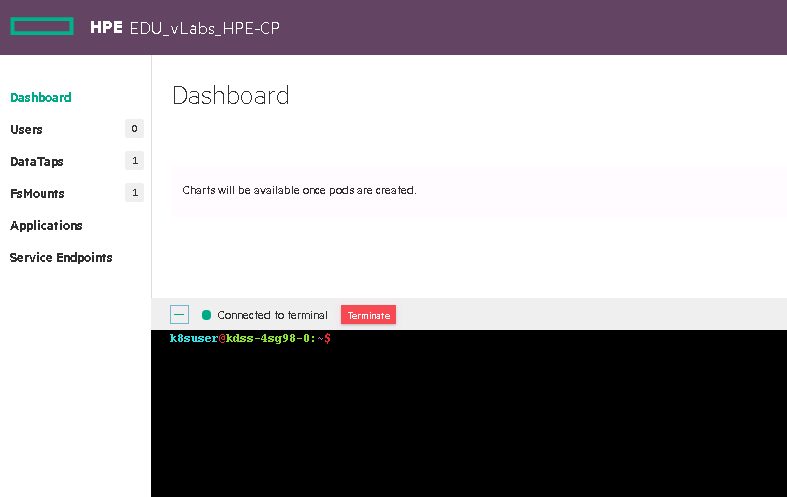
### Login

Follow the procedure

1. Login to HPE Container Platform Web UI
2. From the left-hand menu, click on **Kubernetes** -> **Tenants**
3. Enter any Kubernetes Tenants
4. Click on **Applications**



1. Open the Kubernetes Web Terminal from below

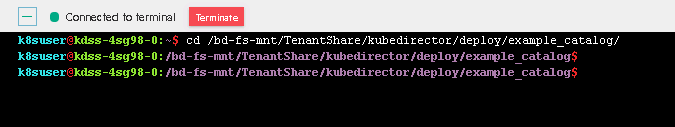


### Create KubeDirectorApp file

Follow the below procedure

1. Navigate to the given location

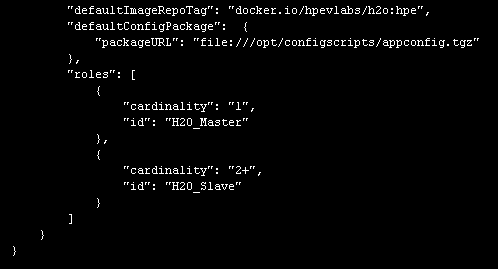
k8suser@kdss-4sg98-0:~$ **cd /bd-fs-mnt/TenantShare/kubedirector/deploy/example\_catalog/**



1. Create **cr-app-h2oflow.json** file

k8suser@kdss-4sg98-0:/bd-fs-mnt/TenantShare/kubedirector/deploy/example\_catalog$ **sudo** **vi** **cr-app-h2oflow.json**

1. Copy the content from **C:\bluedata\_share\HPECP-AppImage-K8s\KD-H2O\cr-app-h2oflow.json** file and paste it in **cr-app-h2oflow.json** file in web terminal
2. Update the tag in **cr-app-h2oflow.json** file in **"defaultImageRepoTag": "docker.io/hpevlabs/h2o:hpe"** (Here, tag used was hpe replace it with name you used while building the docker image)



1. Save and exit the file
2. Create **cr-cluster-h2oflow.yaml** file

k8suser@kdss-4sg98-0:/bd-fs-mnt/TenantShare/kubedirector/deploy/example\_catalog$ **sudo** **vi** **cr-cluster-h2oflow.yaml**

1. Copy the content from **C:\bluedata\_share\HPECP-AppImage-K8s\KD-H2O\cr-cluster-h2oflow.yaml** file and paste it in **cr-cluster-h2oflow.yaml** file in web terminal
2. Save and exit the file
3. Create the JSON file

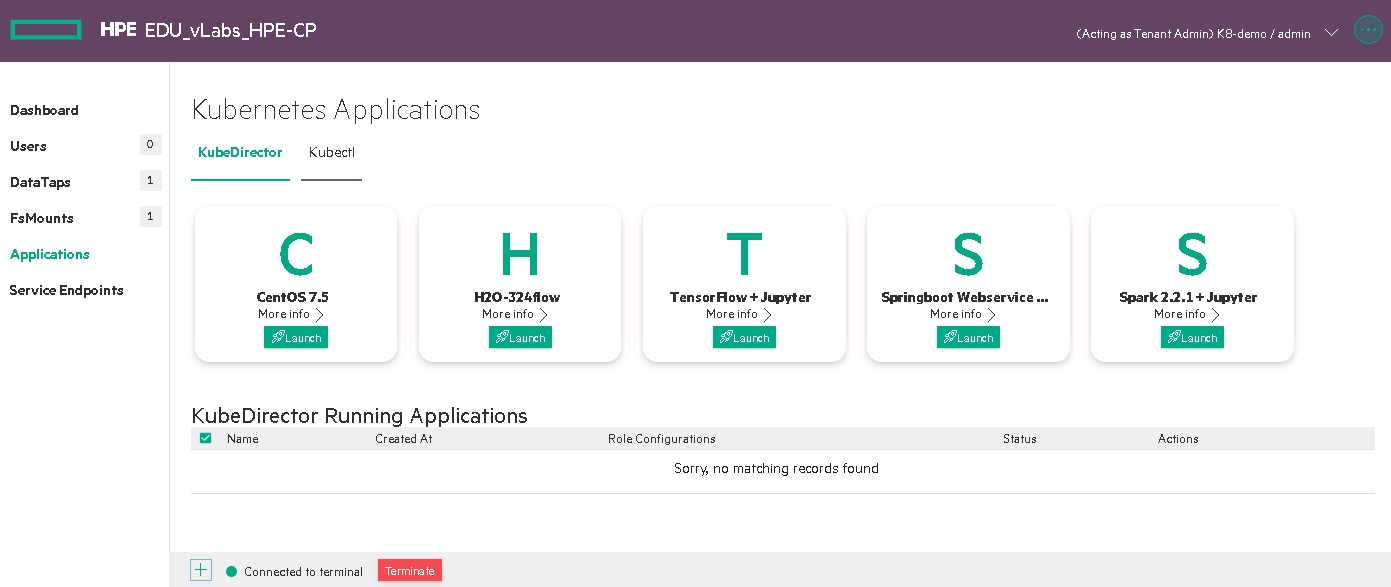
k8suser@kdss-4sg98-0:/bd-fs-mnt/TenantShare/kubedirector/deploy/example\_catalog$ **kubectl create –f cr-app-h2oflow.json**



1. Navigate to the root directory

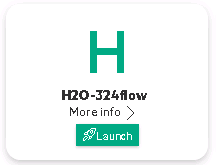
k8suser@kdss-4sg98-0:/bd-fs-mnt/TenantShare/kubedirector/deploy/example\_catalog$ **cd ~**

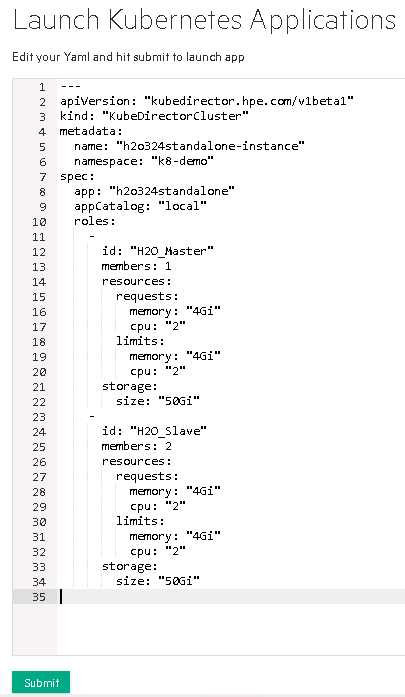
1. Minimize the terminal
2. On the Kubernetes Application page, a new application will appear (Refresh the browser, if required)



### Launch & access the application

1. Click on the **Launch** button on the newly created Application, a Launch Kubernetes Application page will come (make edits, if required) and click on **Submit**



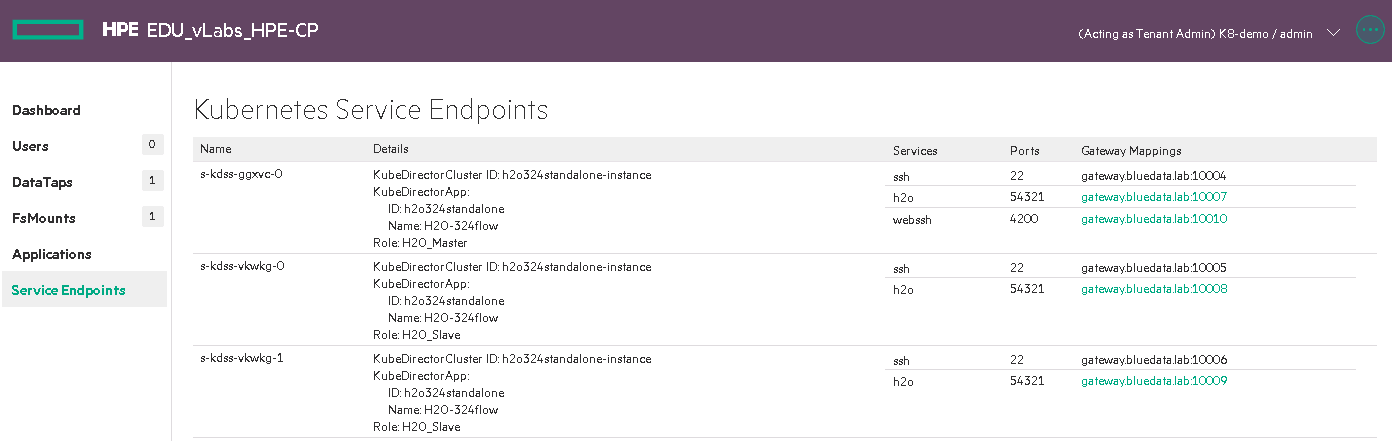


1. On the KubeDirector Running Applications section, the new application will start getting created (it may take appx 10mins, watch the Status column)



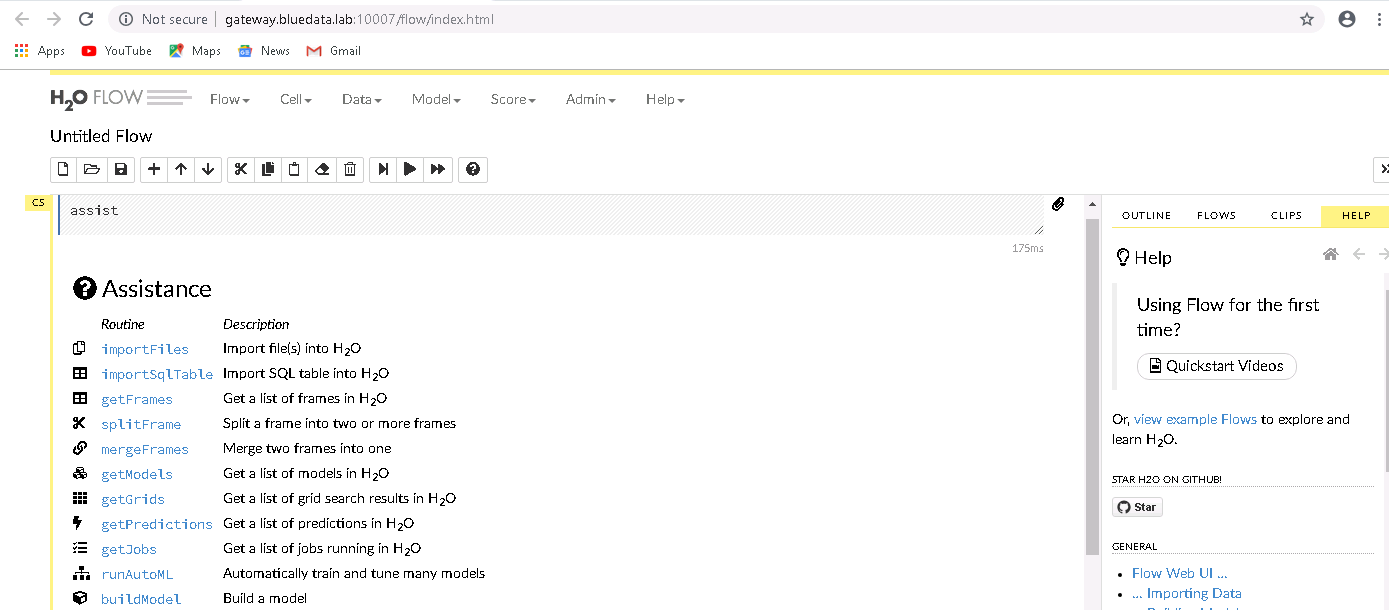


1. Once the application is ready, it can be accessible via service endpoints



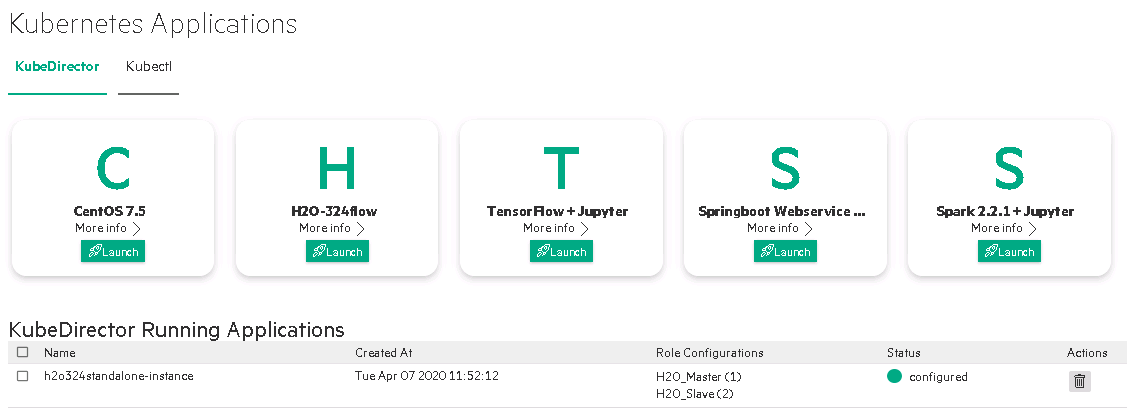
The highlighted shows the role i.e. H2O Master in Details column and in Gateway Mappings column it is showing the hostname of gateway and port which will be used to access the H2O Web UI.

1. Clicking on h2o service **Gateway mapping** for H2O\_Master role will open-up a new tab with H2O Web UI



## Task 3 “Deleting Application”

1. Click on **Applications**



1. In KubeDirector Running Applications section, click on  under **Actions** column
2. A confirmation box will appear, click on **OK** and It will start deleting the application

