Lab 6. Cloud Native CI/CD with tekton and argocd

> Tekton uses kubernetes CRDs - using kubernetes resources - to run CICD pipelines
>
> Light weight / Resource optimized
>
> Reusable tasks - Tekton Hub

0) Install Gitea, Docker Registry

- Login k8sadm@vm01

@vm01

```
$ kc rke-vm01
$ kcg

# install storage-class, set default
$ k apply -f charts/local-path/local-path-storage.yaml
$ k get sc
$ kubectl patch storageclass local-path -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class":"true"}}}'

$ k apply -f charts/gitea/deploy-gitea.yml
```

- http://gitea.vm01
- Set Gitea Base URL : http://gitea.gitea:3000
- Install Gitea
- Register User ID : tekton, Password: 12345678
- New migration > Git > https://github.com/flytux/kw-mvn.git
- New migration > Git > https://github.com/flytux/kw-mvn-deploy.git
- Install Docker Registry & Docker In-secure Setttings

@vm01

```
$ helm install docker-registry -f charts/docker-registry/values.yaml \
  charts/docker-registry -n registry --create-namespace
$ curl -v vm01:30005/v2/_catalog
```

@vm01

```
$ sudo vi /etc/docker/daemon.json
# replace below and save
{
  "insecure-registries": ["vm01:30005"]
}

$ sudo systemctl restart docker

$ sudo docker login vm01:30005
# ID / Password > tekton / 1
```

- Add Project "DEVOPS" in Rancher

1) Install Tekton, Dashboard, Triggers

@vm01

```
$ kubectl apply -f https://storage.googleapis.com/\
  tekton-releases/pipeline/previous/v0.29.1/release.yaml
$ k apply -f charts/tekton/tekton-dashboard-release.yaml
$ kubectl apply -f https://storage.googleapis.com/\
  tekton-releases/triggers/previous/v0.17.1/release.yaml
$ kubectl apply -f https://storage.googleapis.com/\
  tekton-releases/triggers/previous/v0.17.1/interceptors.yaml
```

- http://tekton.vm01
- http://rancher.vm01
- Move gitea, registry, tekton-pipelines namespace to "DEVOPS" project

2) Install Pipeline

```
$ k create ns build
$ kn build
$ k apply -f charts/tekton/pipeline
```

- Add Project "APPS" in Rancher
- Move build namespace to "APPS" project

3) Install ArgoCD

```
$ kubectl create namespace argocd
$ kubectl apply -n argocd -f charts/argocd/
$ kubectl -n argocd get secret argocd-initial-admin-secret \
  -o jsonpath="{.data.password}" | base64 -d # Get admin password
```

4) ArgoCD login and create App

- Cluster rke > Namespaces > Select argocd > Move to DEVOPS Project
- Cluster rke > System > Resources > Workloads > nginx-ingress-controller > ... > Edit
- Show Advanced options > Command > Command > Add "--enable-ssl-passthrough" to the end of arguments > Save
- Login argoCD : https://argocd.vm01
- ID : admin
- Password : # Get admin password

Option1)

@vm01

```
$ k apply -f charts/tekton/argo-app-kw-mvn.yml
```

Option2)

- Manage > Reposotories > Connect Repo Using HTTPS > Project : default
- Repository URL : http://gitea.gitea:3000/tekton/kw-mvn-deploy.git
- Username: tekton, Password: 12345678 > Connect
- ... > Create Application > Application Name : kw-mvn-deploy > Project Name : default
- Revison > main > Path : .
- Cluster URL : https://kubernetes.default.svc
- Namespace : deploy
- Directory Recurse : Check > Create
- Sync > Auto-create Namespace : Check

5) Create argocd-token

- Rancher > Cluster rke > DEVOPS > Resource > Config > argocd-cm > Edit
- add data > Key: accounts.admin Value: apiKey, login
- Save

- Login argocd : https://argocd.vm01
- Manage > Account > admin > Tokens > Generate New
- Copy New Token:
- Login vm01
- $ vi charts/tekton/argo-token.sh
- Replace ARGOCD_AUTH_TOKEN value with New Token value and wq
- $ charts/tekton/argo-token.sh

6) Run Pipeline

```
$ kcg
$ kc rke
$ kn build
$ k create -f charts/tekton/pipeline/pr-kw-build.yml
$ tkn pr logs -f
```

- http://tekton.vm01/#/namespaces/build/pipelineruns

- Check application : http://vm01:30088/

7) Add Webhook to Gitea Repo

- http://gitea.vm01/tekton/kw-mvn
- Settings > Webhooks > Add Webhook > Gitea
- Target URL : http://el-build-listener.build:8080
- Add Webhook
- Click Webhook > Test Delivery
- Check Pipeline Runs

8) Git push source repo will trigger tekton pipeline

- Edit source and commit
- Check Pipeline Runs
- Check argocd app deployment status
- Check application : http://vm01:30088/