|  |  |
| --- | --- |
| **DOCUMENT RULES:** | |
| **Task Number / Name:** | **ArgoCD** |
| **Task name & column name should be written:** | **Bold (CTRL+B)** |
| **Commands should be written in the after # sign:** | *Italic (CTRL+I) #hostname* |
| **Output photo should be cropped or compressed:**  **Photo could be more than one:**  **If you need extra lines, add the line next after it:** | ***Description photo should be with title bar (CTRL + I + B)*** |
| **All other text should be written:** | Standard |
| **Font name and text size:** | Calibri and 9 |
| **Group name:** | Dev\_ops\_ |
| **Student name and surname:** | Murad Abbaszade |
| **E-mail:** | [muradabbaszade6@gmail.com](mailto:muradabbaszade6@gmail.com) |
| **WhatsApp number:** | **+994703664205** |

|  |  |
| --- | --- |
| 1.ArgoCD App of Apps |  |
| Check if kubernetes is up and running(It can take some time):  kubectl cluster-info  Clone the sample App Of Apps implementation to current directory:  git clone <https://github.com/a1tan/appofappssample.git> |  |
| Install the ArgoCD application:  kubectl apply -k appofappssample/applications/argocd  kubectl get pods -n argocd –w |  |
| Apply the parent application definition:  kubectl apply -k appofappssample/parentapplication |  |
| Check if applications installed and synchronized:  kubectl get application -n argocd -owide -w |  |
| 2.ArgoCD ApplicationSet |  |
| Check if kubernetes is up and running(It can take some time):  kubectl cluster-info  Clone the sample App Of Apps implementation to current directory:  git clone https://github.com/a1tan/argoapplicationsets.git |  |
| Install the ArgoCD application:  kubectl apply -k argoapplicationsets/managementstack/argocd  kubectl get pods -n argocd -w |  |
| Apply the applicationset definition:  kubectl apply -k argoapplicationsets/managementstack/argocdapplications |  |
| 3.ArgoCD Autopilot |  |
| kubectl cluster-info  Run below commands to setup argocd autopilot:  VERSION=$(curl --silent "https://api.github.com/repos/argoproj-labs/argocd-autopilot/releases/latest" | grep '"tag\_name"' | sed -E 's/.\*"([^"]+)".\*/\1/')  curl -L --output - https://github.com/argoproj-labs/argocd-autopilot/releases/download/$VERSION/argocd-autopilot-linux-amd64.tar.gz | tar zx  mv ./argocd-autopilot-\* /usr/local/bin/argocd-autopilot  argocd-autopilot version |  |
| export GIT\_TOKEN=ghp\_Yzuo3ZyInG6F9Md3F544aKFQTV64u9122X0Q  git config --global user.email "muradabbaszade6@gmail.com"  git config --global user.name "MuradAbbaszade"  export GIT\_REPO=https://github.com/MuradAbbaszade/LoginAndRegister |  |
| Run command to create repository and bootstrap ArgoCD:  argocd-autopilot repo bootstrap |  |
| kubectl get pods -n argocd -w |  |
| kubectl get application -n argocd -owide -w |  |
| 4.Bootstrap flux with Bootstrap CLI |  |
| kubectl cluster-info |  |
| Install flux cli:  curl -s https://fluxcd.io/install.sh | sudo bash |  |
| export GITHUB\_TOKEN=ghp\_Yzuo3ZyInG6F9Md3F544aKFQTV64u9122X0Q |  |
| kubectl get pods -n flux-system -owide -w |  |
| Check if sources for Git Repository applied: kubectl get GitRepository -A -owide |  |
| Check if sources for Kustomization applied: kubectl get Kustomization -A -owide |  |
| 5.Bootstrap flux with terraform |  |
| Check if kubernetes is up and running(It can take some time):  kubectl cluster-info |  |
| Install terraform cli:  curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add - |  |
| sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb\_release -cs) main" |  |
| sudo apt-get update && sudo apt-get install terraform |  |
| Download terraform resource to be applied:  git clone https://github.com/a1tan/fluxterraformgithub.git  cd fluxterraformgithub |  |
| terraform init |  |
| terraform plan -out=tfplan |  |
| terraform apply tfplan |  |
| kubectl get pods -n flux-system -owide -w |  |
| Check if sources for Git Repository applied: kubectl get GitRepository -A -owide  Check if sources for Kustomization applied: kubectl get Kustomization -A -owide |  |
| Pull Request Ephemeral Environments |  |
| kubectl cluster-info  Clone the sample App Of Apps implementation to current directory:  git clone https://github.com/a1tan/ephemeralenvironments.git |  |
| Install the ArgoCD application:  kubectl apply -k ephemeralenvironments/managementstack/argocd  kubectl get pods -n argocd -w |  |
| kubectl apply -k ephemeralenvironments/managementstack/argocdapplications |  |
| kubectl get application -n argocd -owide -w |  |
| kubectl get ns |  |
| kubectl port-forward svc/sampleapp-ns-f-ephemeralpullrequest-3 -n sampleapp-ns-f-ephemeralpullrequest-3-namespace 8080:80 & |  |
| curl http://localhost:8080/Sample |  |
| kubectl get secret vc-cluster-f-ephemeralpullrequest-3 -n cluster-f-ephemeralpullrequest-3-namespace -ojsonpath={.data.config} | base64 -d > config.yaml  sed -i 's/cluster-f-ephemeralpullrequest-3.cluster-f-ephemeralpullrequest-3-namespace.svc.cluster.local/localhost:8081/g' config.yaml  Port forward to VCluster:  kubectl port-forward svc/cluster-f-ephemeralpullrequest-3 -n cluster-f-ephemeralpullrequest-3-namespace 8081:443 & |  |
| kubectl port-forward svc/sampleapp-f-ephemeralpullrequest-3 -n sampleapp-f-ephemeralpullrequest-3-namespace --kubeconfig config.yaml 8082:80 &  curl http://localhost:8082/Sample |  |