Lab1:

In this session we will be using ubuntu 22.04 server, All the prerequisites already followed:

- Docker installation: https://www.openwriteup.com/?page_id=785
- Kubernetes setup: https://www.openwriteup.com/?page_id=866
- Untainted the master (As we are using single host): https://www.openwriteup.com/?page_id=887

*Note: In case you are doing first time, please follow the guideline provided for docker and Kubernetes*

Setup Argo Cli

```
# Download the binary

curl-sLO https://github.com/argoproj/argo-workflows/releases/download/v3.5.6/argo-linux-amd64.gz

# Unzip

gunzip argo-linux-amd64.gz

# Make binary executable

chmod +x argo-linux-amd64

# Move binary to path

mv ./argo-linux-amd64 /usr/local/bin/argo

# Test installation

argo version
```

```
Destroy complete! Resources: 2 destroyed.
root@devvm:~/base/system# argo version
argo: v3.2.6
  BuildDate: 2021-12-17T20:59:31Z
  GitCommit: db7d90a1f609685cfda73644155854b06fa5d28b
  GitTreeState: clean
  GitTag: v3.2.6
  GoVersion: go1.16.12
  Compiler: gc
  Platform: linux/amd64
```

Controller and Server

```
kubectl create namespace argo
kubectl apply -n argo -f https://github.com/argoproj/argo-workflows/releases/download/v3.5.6/install.yaml
```

```
root@devvm:/home/amit/argowf# kubectl create namespace argo
kubectl apply -n argo -f https://github.com/argoproj/argo-workflows/releases/download/v3.5
namespace/argo created
customresourcedefinition.apiextensions.k8s.io/clusterworkflowtemplates.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/cronworkflows.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/workflowartifactgctasks.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/workfloweventbindings.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/workflows.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/workflowtaskresults.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/workflowtasksets.argoproj.io created
customresourcedefinition.apiextensions.k8s.io/workflowtemplates.argoproj.io created
serviceaccount/argo created
serviceaccount/argo-server created
role.rbac.authorization.k8s.io/argo-role created
clusterrole.rbac.authorization.k8s.io/argo-aggregate-to-admin created
clusterrole.rbac.authorization.k8s.io/argo-aggregate-to-edit created
clusterrole.rbac.authorization.k8s.io/argo-aggregate-to-view created
clusterrole.rbac.authorization.k8s.io/argo-cluster-role created
clusterrole.rbac.authorization.k8s.io/argo-server-cluster-role created
rolebinding.rbac.authorization.k8s.io/argo-binding created
clusterrolebinding.rbac.authorization.k8s.io/argo-binding created
clusterrolebinding.rbac.authorization.k8s.io/argo-server-binding created
configmap/workflow-controller-configmap created
service/argo-server created
priorityclass.scheduling.k8s.io/workflow-controller created
deployment.apps/argo-server created
deployment.apps/workflow-controller created
```

kubectl get pods -n argo

```
root@devvm:/home/amit/argowf# kubectl get pods -n argo
NAME                                     READY   STATUS    RESTARTS   AGE
argo-server-579d446586-w526j             1/1     Running   0          37s
workflow-controller-566fcc48f-pdm8b      1/1     Running   0          37s
root@devvm:/home/amit/argowf#
```