kubeflow setting

nfs 서버 설치 및 설정(스토리지 서버를 할 서버 하나 선택해서)

nfs 서버로 제공할 공간 확보

```
$ sudo mkdir -p /mnt/storage/nfs_storage
$ sudo chmod -R 777 /mnt/storage/nfs_storage
```

nfs 서버 설치

```
$ sudo apt-get -y install nfs-common nfs-kernel-server rpcbind portmap
```

ip 확인 : master node와 worker node의 ip를 확인합니다.

```
$ kubectl get node -o wide

#  NAME                STATUS   ROLES    AGE   VERSION   INTERNAL-IP    EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION       CON
#  koock-coach-master   Ready    master   42h   v1.15.5   172.27.0.82    <none>        Ubuntu 18.04.3 LTS   4.15.0-118-generic   doc
#  koock-coach-worker   Ready    <none>   42h   v1.15.5   172.27.0.129   <none>        Ubuntu 18.04.3 LTS   4.15.0-118-generic   doc
```

nfs 서버 설정

```
$ sudo nano /etc/exports
```

```
/mnt/storage/nfs_storage 마스터노드IP(rw,insecure,sync,no_root_squash,no_subtree_check)
/mnt/storage/nfs_storage 워커노드IP(rw,insecure,sync,no_root_squash,no_subtree_check)
```

```
$ sudo exportfs -a
$ sudo systemctl restart nfs-kernel-server
```

nfs client설치(worker node)

```
$ sudo apt install nfs-common
```

nfs 서버 테스트 : 클라이언트

```
$ mkdir test_mount
$ sudo mount nfs서버IP:/mnt/storage/nfs_storage test_mount
$ touch test_mount/test.txt
```

nfs 서버 테스트 : 서버

```
$ ls /mnt/storage/nfs_storage
```

nfs 서버 테스트 종료 : 클라이언트

```
$ umount test_mount
```

동적 프로비저닝을 위한 스토리지 클래스 설치

로컬 패스 스토리지

```
$ kubectl apply -f https://raw.githubusercontent.com/rancher/local-path-provisioner/master/deploy/local-path-storage.yaml
```

helm 설치

```
$ curl https://raw.githubusercontent.com/helm/helm/release-2.16/scripts/get > get_helm.sh
$ chmod 700 get_helm.sh
$ ./get_helm.sh
```

nfs-client-provisioner 설치

```
$ helm repo add stable https://charts.helm.sh/stable
$ helm install --generate-name  --set nfs.server=172.27.0.121 --set nfs.path=/mnt/storage/nfs_storage stable/nfs-client-provisioner
```

storageclass 세팅

```
$ kubectl patch storageclass nfs-client -p '{"metadata":{"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
$ kubectl get storageclass
```

kubernetes상에서 그래픽카드 사용을 위한 플러그인 설치

```
kubectl create -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/1.0.0-beta4/nvidia-device-plugin.yml
```

docker private registry 배포

```
$ kubectl apply -f https://raw.githubusercontent.com/mojokb/handson-kubeflow/master/registry/kubeflow-registry-deploy.yaml
$ kubectl apply -f https://raw.githubusercontent.com/mojokb/handson-kubeflow/master/registry/kubeflow-registry-svc.yaml
```

docker daemon 세팅(마스터노드 워커노드 둘다)

```
$ sudo nano /etc/docker/daemon.json
```

```
{
    "exec-opts": ["native.cgroupdriver=systemd"],
    "log-driver": "json-file",
    "log-opts": {
        "max-size": "100m"
    },
    "data-root": "/mnt/storage/docker_data",
    "storage-driver": "overlay2",
    "default-runtime" : "nvidia",
    "runtimes" : {
            "nvidia" : {
                    "path": "/usr/bin/nvidia-container-runtime",
                    "runtimeArgs" : []
            }
    },
    "insecure-registries": [
```

```
            "kubeflow-registry.default.svc.cluster.local:30000"
    ]
}
```

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart docker
```

host ip 등록(마스터노드 워커노드 둘다 적용)

```
$ sudo nano /etc/hosts
```

```
마스터노드IP   kubeflow-registry.default.svc.cluster.local
```

registry 확인

```
$ curl kubeflow-registry.default.svc.cluster.local:30000/v2/_catalog
```

k9s 설치

```
$ wget https://github.com/derailed/k9s/releases/download/v0.13.7/k9s_0.13.7_linux_i386.tar.gz
$ tar zxvf k9s_0.13.7_linux_i386.tar.gz
$ sudo mv k9s /usr/bin
```

설치 확인

```
$ k9s
```

kfctl 설치

```
$ wget https://github.com/kubeflow/kfctl/releases/download/v1.0.2/kfctl_v1.0.2-0-ga476281_linux.tar.gz
$ tar zxvf kfctl_v1.0.2-0-ga476281_linux.tar.gz
$ sudo mv kfctl /usr/bin
```

kubeflow 설치

```
$ export KF_NAME=my-kubeflow
$ export BASE_DIR=/opt/kubeflow
$ export KF_DIR=${BASE_DIR}/${KF_NAME}
$ export CONFIG_URI="https://raw.githubusercontent.com/kubeflow/manifests/v1.0-branch/kfdef/kfctl_k8s_istio.v1.0.2.yaml"
$ sudo mkdir -p ${KF_DIR}
$ sudo chmod 777 ${KF_DIR}
$ cd ${KF_DIR}
$ kfctl apply -V -f ${CONFIG_URI}
```

kubeflow 설치확인

```
$ kubectl get pods -n kubeflow -o wide
```