



Handson Kubeflow (o)

(주) 두다지 - 이명환
kb@dudaji.com

CONTENTS

1. Deploying Kubeflow on Existing Clusters

Minimum system requirements

The Kubernetes cluster must meet the following minimum requirements:

- Your cluster must include at least one worker node with a minimum of:
 - 4 CPU
 - 50 GB storage
 - 12 GB memory
- The recommended Kubernetes version is 1.14. Kubeflow has been validated and tested on Kubernetes 1.14.
 - Your cluster must run at least Kubernetes version 1.11.
 - Kubeflow **does not work** on Kubernetes 1.16.
 - Older versions of Kubernetes may not be compatible with the latest Kubeflow versions. The following matrix provides information about compatibility between Kubeflow and Kubernetes versions.

Kubernetes Versions	Kubeflow 0.4	Kubeflow 0.5	Kubeflow 0.6	Kubeflow 0.7
1.11	compatible	compatible	incompatible	incompatible
1.12	compatible	compatible	incompatible	incompatible
1.13	compatible	compatible	incompatible	incompatible
1.14	compatible	compatible	compatible	compatible
1.15	incompatible	compatible	compatible	compatible
1.16	incompatible	incompatible	incompatible	incompatible

1. Deploying Kubeflow on Existing Clusters

vm on GCP (1)

kubeflow가 설치될 kubernetes 환경은 google Cloud의 custom vm에서 microk8s 아니면 kubernetes 1.15버전을 사용합니다.

<https://cloud.google.com/>

핸즈온용 VM 인스턴스 사양 :

8 CPU

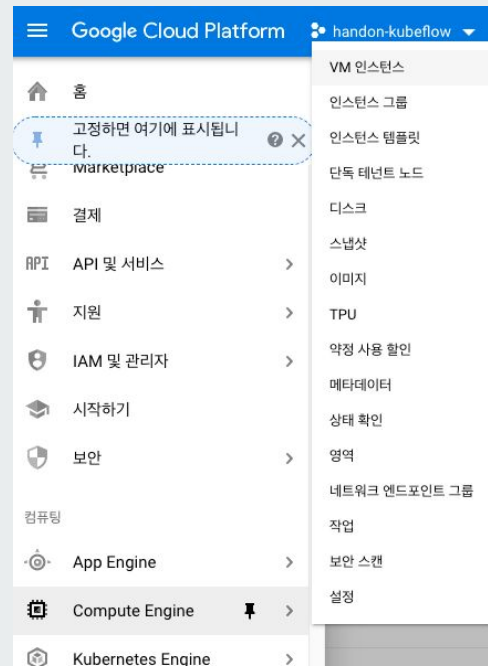
20GB Memory

100gb,

Ubuntu 18.04 minimal

(approx. 0.31\$/hour)

핸즈온 끝나시면 꼭 VM 삭제 부탁드립니다..



1. Deploying Kubeflow on Existing Clusters

vm on GCP (2)

이름 [?]
handson-kubeflow

[?] Name can't be changed once the 인스턴스는 created

리전 [?] asia-northeast1(도쿄) 영역 [?] asia-northeast1-b 월 \$226.62 예상
시간당 약 \$0.31
사용한 만큼만 비용 지불: 선불 비용 없이 초당 청구

마신 구성 [?]

마신 계열
일반 용도

일반적인 작업 부하에 적합한 마신 유형이며 가격 및 유연성을 위해 최적화되었습니다.

시리즈
N1
Intel Skylake CPU 플랫폼 또는 이전 버전의 플랫폼에서 제공

마신 유형
커스텀

코어
8 vCPU 1 - 96

메모리
20 GB 7.25 - 52

메모리 확장 [?]

[?] CPU 플랫폼 및 GPU

컨테이너 [?]
 이 VM 인스턴스에 컨테이너 이미지를 배포합니다. 자세히 알아보기

부팅 디스크 [?]

새로운 100GB 표준 영구 디스크 이미지
Ubuntu 18.04 LTS Minimal 변경

ID 및 API 액세스 [?]

서비스 계정 [?]
Compute Engine default service account

액세스 범위 [?]

기본 액세스 허용
 모든 Cloud API에 대한 전체 액세스 허용
 각 API에 액세스 설정

방화벽 [?]
태그 및 방화벽 규칙을 추가하여 인터넷에서 특정 네트워크 트래픽을 허용합니다.

HTTP 트래픽 허용
 HTTPS 트래픽 허용

[?] 관리, 보안, 디스크, 네트워크, 단독 임대

이 인스턴스의 요금이 청구됩니다. [Compute Engine 가격 책정](#) [?]

만들기 취소

방화벽 설정에서 HTTP/HTTPS 트래픽 허용을 체크!

1. Deploying Kubeflow on Existing Clusters

vm on GCP (3)

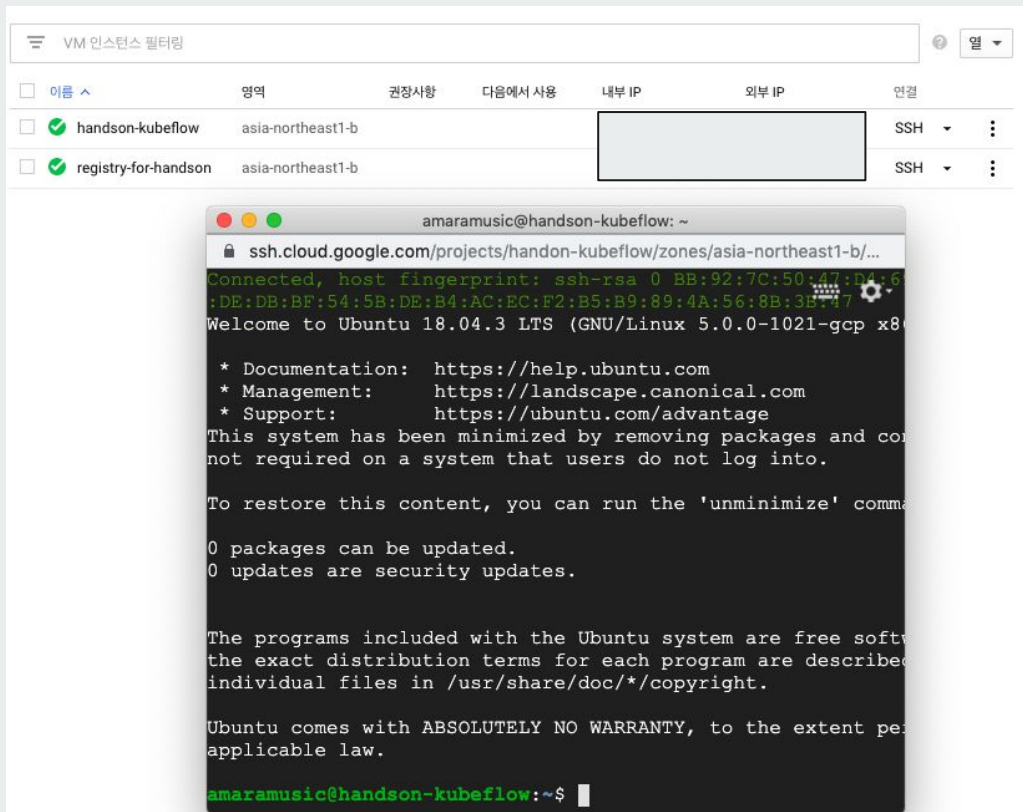


<input type="checkbox"/>	이름 [▲]	영역	권장사항	다음에서 사용	내부 IP	외부 IP	연결
<input type="checkbox"/>	✔ handson-kubeflow	asia-northeast1-b					SSH
<input type="checkbox"/>	✔ registry-for-handson	asia-northeast1-b					SSH

생성후 ssh를 눌러 콘솔 창을 엽니다.

1. Deploying Kubeflow on Existing Clusters

vm on GCP (4)



The image shows a GCP VM instances page with two instances: 'handson-kubeflow' and 'registry-for-handson'. Below it is a terminal window showing the SSH connection to the 'handson-kubeflow' instance, displaying the Ubuntu 18.04.3 LTS login screen.

<input type="checkbox"/>	이름 ^	영역	권장사항	다음에서 사용	내부 IP	외부 IP	연결
<input checked="" type="checkbox"/>	handson-kubeflow	asia-northeast1-b					SSH ▾ ⋮
<input checked="" type="checkbox"/>	registry-for-handson	asia-northeast1-b					SSH ▾ ⋮

```
amaramusic@handson-kubeflow: ~  
ssh.cloud.google.com/projects/handson-kubeflow/zones/asia-northeast1-b/...  
Connected, host fingerprint: ssh-rsa 0 BB:92:7C:50:47:D4:6  
;DE:DB:BF:54:5B:DE:B4:AC:EC:F2:B5:B9:89:4A:56:8B:3B:47  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1021-gcp x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
This system has been minimized by removing packages and con  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' comm  
  
0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free softw  
the exact distribution terms for each program are describ  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent pe  
applicable law.  
amaramusic@handson-kubeflow:~$
```

1. Deploying Kubeflow on Existing Clusters

microk8s



Install microk8s

```
# in vm
$ sudo apt-get update
$ sudo apt-get install -y git snap vim docker.io
$ sudo gpasswd -a $USER docker | newgrp docker
$ sudo snap install microk8s --classic --channel=1.13/stable
$ sudo usermod -a -G microk8s ${USER}

# re login
# add alias to profile
alias k=microk8s.kubectl
alias kubectl=microk8s.kubectl
$ k get no
$ sudo microk8s.kubectl config view --raw > $HOME/.kube/config

# for hostpath-provisioner
$ microk8s.enable storage dns
```

1. Deploying Kubeflow on Existing Clusters

by kubernetes 1.15 (1)

```
$ apt-get update
### docker-ce install , 18.09
https://docs.docker.com/install/linux/docker-ce/ubuntu/
$ sudo apt-get install -y apt-transport-https ca-certificates curl gnupg-agent software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
$ sudo apt-get update
$ sudo apt-get install -y docker-ce=5:18.09.9~3-0~ubuntu-bionic docker-ce-cli=5:18.09.9~3-0~ubuntu-bionic containerd.io vim

##### be root! - experimental...

$ cat > /etc/docker/daemon.json <<EOF
{
  "exec-opts": ["native.cgroupdriver=systemd"],
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "100m"
  },
  "storage-driver": "overlay2"
}
EOF
$ mkdir -p /etc/systemd/system/docker.service.d
$ systemctl daemon-reload
$ systemctl restart docker
```


1. Deploying Kubeflow on Existing Clusters

by kubernetes 1.15 - nvidiadocker

```
### OPTIONAL::: install nvidia-driver and install nvidia-docker for gpu -> https://hiseon.me/linux/ubuntu/install\_nvidia\_driver/ https://jybaek.tistory.com/796
$ release=$(lsb_release -sr | sed -e "s/./g")
$ sudo apt install sudo gnupg
$ sudo apt-key adv --fetch-keys "http://developer.download.nvidia.com/compute/cuda/repos/"$release"/x86_64/7fa2af80.pub"
$ sudo sh -c 'echo "deb http://developer.download.nvidia.com/compute/cuda/repos/"$release"/x86_64/" > /etc/apt/sources.list.d/nvidia-cuda.list'
$ sudo sh -c 'echo "deb http://developer.download.nvidia.com/compute/machine-learning/repos/"$release"/x86_64/" > /etc/apt/sources.list.d/nvidia-machine-learning.list'
$ sudo apt update

$ apt-cache search nvidia
$ sudo apt-get install -y nvidia-XXX
$ sudo apt-get install -y dkms nvidia-modprobe

$ reboot
$ sudo cat /proc/driver/nvidia/version | nvidia-smi

$ curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -
$ distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
$ curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list
$ sudo apt-get update
## if docker 19.03 이후는 https://github.com/NVIDIA/nvidia-docker/tree/master#upgrading-with-nvidia-docker2-deprecated 참조
$ sudo apt-get install -y nvidia-docker2
$ sudo vi /etc/docker/daemon.json
"default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
      "path": "/usr/bin/nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
# then reboot docker....
$ sudo docker run --runtime=nvidia --rm nvidia/cuda nvidia-smi
```

1. Deploying Kubeflow on Existing Clusters

by kubernetes 1.15 (2)

install kubernetes

- <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/>

- <https://vitux.com/install-and-deploy-kubernetes-on-ubuntu/>

```
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
```

```
$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"
```

```
$ sudo apt-get install -y kubelet=1.15.5-00 kubeadm=1.15.5-00 kubectl=1.15.5-00
```

```
$ sudo apt-mark hold kubelet kubeadm kubectl
```

```
$ sudo systemctl net.bridge.bridge-nf-call-iptables=1
```

```
$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16
```

```
$ mkdir -p $HOME/kube
```

```
$ sudo cp -i /etc/kubernetes/admin.conf $HOME/kube/config
```

```
$ sudo chown $(id -u):$(id -g) $HOME/kube/config
```

```
$ kubectl taint nodes --all node-role.kubernetes.io/master-
```

```
$ kubectl apply -f https://docs.projectcalico.org/v3.8/manifests/calico.yaml
```

install storage Class - named local-path

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

set default storage class

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

OPTIONAL: Install nvidia-gpu-plugin, installed nvidia-docker2

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

OPTIONAL : install nfs-client-provisioner, ReadManyWrite를 위해 로컬의 볼륨을 nfs로 설정하여 등록합니다.

<https://github.com/helm/charts/tree/master/stable/nfs-client-provisioner#configuration>

<https://vitux.com/install-nfs-server-and-client-on-ubuntu/>

```
mojokb@ubuntu:~$ k get po -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-5c98db65d4-2zbfz            1/1    Running   1           29m
coredns-5c98db65d4-gp2r2            1/1    Running   1           29m
etcd-ubuntu                          1/1    Running   1           29m
kube-apiserver-ubuntu                1/1    Running   1           28m
kube-controller-manager-ubuntu       1/1    Running   1           28m
kube-flannel-ds-amd64-j72js          1/1    Running   1           29m
kube-proxy-6whpx                     1/1    Running   1           29m
kube-scheduler-ubuntu                1/1    Running   1           28m
nvidia-device-plugin-daemonset-j67wr 1/1    Running   0           53s
mojokb@ubuntu:~$
```

1. Deploying Kubeflow on Existing Clusters

Deploy kubeflow (1)

<https://www.kubeflow.org/docs/started/k8s/overview/#community-maintained>

kubeflow는 community maintained 버전을 사용합니다.

0.7.0이 나왔습니다!

(<https://docs.google.com/document/d/13fv-D0m40v1ugE1Pt8kXe0T7xA5MSS1yno3dUK732r8/edit#heading=h.s5sf36exyj4f>)

<https://github.com/kubeflow/kubeflow/releases/tag/v0.7.0>

```
$ wget https://github.com/kubeflow/kubeflow/releases/download/v0.7.0/kfctl_v0.7.0_linux.tar.gz
```

```
$ tar -xvf kfctl_v0.7.0_linux.tar.gz
```

```
$ sudo mv kfctl /usr/bin
```

```
$ export KF_NAME=handson-kubeflow
```

```
$ export BASE_DIR=/home/${USER}
```

```
$ export KF_DIR=${BASE_DIR}/${KF_NAME}
```

```
$ export
```

```
CONFIG_URI="https://raw.githubusercontent.com/kubeflow/manifests/v0.7-branch/kfdef/kfctl_k8s_istio.0.7.0.yaml"
```

1. Deploying Kubeflow on Existing Clusters

Deploy kubeflow (2)

```
$ mkdir -p ${KF_DIR}
$ cd ${KF_DIR}
$ kfctl build -V -f ${CONFIG_URI}
```

```
amaramusic@handson-kubeflow:~/handson-kubeflow$ kfctl build -V -f ${CONFIG_URI}
INFO[0000] Downloading https://raw.githubusercontent.com/kubeflow/manifests/v0.7-branch/kfdef/kfctl_k8s_istio.0.7.0.yaml
INFO[0000] Downloading https://raw.githubusercontent.com/kubeflow/manifests/v0.7-branch/kfdef/kfctl_k8s_istio.0.7.0.yaml
70"
INFO[0000] App directory /home/amaramusic/handson-kubeflow already exists filename="coordinator/coordinator.go:274"
INFO[0000] Writing KfDef to kfctl_k8s_istio.0.7.0.yaml filename="coordinator/coordinator.go:277"
INFO[0000] No name specified in KfDef.Metadata.Name; defaulting to handson-kubeflow based on location of config file: /
rdinator/coordinator.go:202"
INFO[0000] :340"
***** :340"
Notice anonymous usage reporting enabled using spartakus :340"
To disable it :340"
If you have already deployed it run the following commands: :340"
  cd $(pwd) :340"
  kubectl -n ${K8S_NAMESPACE} delete deploy -l app=spartakus :340"
For more info: https://www.kubeflow.org/docs/other-guides/usage-reporting/ :340"
***** :340"
filename="coordinator/coordinator.go:120" :340"
INFO[0000] Creating directory /home/amaramusic/handson-kubeflow/.cache filename="kfconfig/types.go:437" :340"
INFO[0000] Fetching https://github.com/kubeflow/manifests/archive/v0.7-branch.tar.gz to /home/amaramusic/handson-kubefl; :340"
INFO[0002] Fetch succeeded; LocalPath /home/amaramusic/handson-kubeflow/.cache/manifests/manifests-0.7-branch filename; :340"
INFO[0002] Processing application: istio-crds filename="kustomize/kustomize.go:340" :340"
INFO[0002] Processing application: profiles filename="kustomize/kustomize.go:340" :340"
INFO[0002] Processing application: seldon-core-operator filename="kustomize/kustomize.go:340" :340"
amaramusic@handson-kubeflow:~/handson-kubeflow$
```

1. Deploying Kubeflow on Existing Clusters

Deploy kubeflow (3)

```
$ export CONFIG_FILE=${KF_DIR}/kfctl_k8s_istio.0.7.0.yaml  
$ kfctl apply -V -f ${CONFIG_FILE}
```

```
amaramusic@handson-kubeflow:~/handson-kubeflow$ kfctl apply -V -f ${CONFIG_FILE}  
INFO[0000] No name specified in KfDef.Metadata.Name; defaulting to handson-kubeflow based on location of config file: /home/ama  
tor/coordinator.go:202"  
INFO[0000]  
*****  
Notice anonymous usage reporting enabled using spartakus  
To disable it  
If you have already deployed it run the following commands:  
  cd $(pwd)  
  kubectl -n ${K8S_NAMESPACE} delete deploy -l app=spartakus  
  
For more info: https://www.kubeflow.org/docs/quick-start/quick-start-istio/  
*****  
filename="coordinator/coordinator.go:202"  
INFO[0000] Deleting cachedir /home/amaramusic/handson-kubeflow  
INFO[0000] Fetching https://github.com/kubeflow/manifests/releases/download/v0.7.0/manifests.yaml  
INFO[0003] Fetch succeeded; LocalPath /home/amaramusic/handson-kubeflow/manifests.yaml  
INFO[0003] folder /home/amaramusic/handson-kubeflow/manifests.yaml  
INFO[0003] /home/amaramusic/handson-kubeflow/manifests.yaml  
INFO[0003] namespace: kubeflow  
application.app.k8s.io/manifests created  
virtualservice.networking.istio.io/kfam created  
customresourcedefinition.apiextensions.k8s.io/seldondeployments.machinelearning.seldon.io/seldon-deployments created  
serviceaccount/seldon-manager created  
clusterrole.rbac.authorization.k8s.io/seldon-operator-manager-role created  
clusterrolebinding.rbac.authorization.k8s.io/seldon-operator-manager-rolebinding created  
configmap/seldon-config created  
secret/seldon-operator-webhook-server-secret created  
service/seldon-operator-controller-manager-service created  
service/webhook-server-service created  
statefulset.apps/seldon-operator-controller-manager created  
application.app.k8s.io/seldon-core-operator created  
INFO[0011] Applied the configuration Successfully! filename="cmd/apply.go:72"  
amaramusic@handson-kubeflow:~/handson-kubeflow$ k get po
```


1. Deploying Kubeflow on Existing Clusters

Deploy kubeflow (4)

\$ kubectl get po -n kubeflow

```
amaramusic@handson:~/handson-kubeflow$ k get po -n kubeflow
NAME                                READY   STATUS    RESTARTS   AGE
admission-webhook-bootstrap-stateful-set-0      0/1    Pending   0           10s
admission-webhook-deployment-78d899bf68-ncsh4   0/1    Pending   0           10s
application-controller-stateful-set-0          0/1    Pending   0           11s
argo-ui-55b859f7d7-tkzp4                      0/1    Pending   0           11s
centraldashboard-75474d6f94-9tgwl             0/1    Pending   0           10s
jupyter-web-app-deployment-6c8f4c8997-9817c     0/1    Pending   0           10s
katib-controller-7ddd4c8b8c-4z14d              0/1    Pending   0           5s
katib-db-7b679f6f8c-g7dtr                     0/1    Pending   0           5s
katib-manager-84c4fb876b-2gxnx                 0/1    Pending   0           5s
katib-ui-5d454c75c7-s45qg                     0/1    Pending   0           5s
kfserving-controller-manager-0                 0/2    Pending   0           8s
metadata-db-5dd459cc-zp7th                     0/1    Pending   0           10s
metadata-deployment-b745d8bcf-gp8jt            0/1    Pending   0           10s
metadata-deployment-b745d8bcf-rscfd           0/1    Pending   0           10s
metadata-envoy-deployment-7ccf5c4f74-xxmlc      0/1    Pending   0           10s
metadata-grpc-deployment-6496f66c8c-6g2vb      0/1    Pending   0           9s
metadata-grpc-deployment-6496f66c8c-9mld8      0/1    Pending   0           9s
metadata-ui-78f5b59b56-4rvjk                   0/1    Pending   0           9s
minio-6f48db9cc4-hgjxz                         0/1    Pending   0           4s
ml-pipeline-844645fd-w7g27                     0/1    Pending   0           5s
ml-pipeline-ml-pipeline-visualizationserver-865894f5f7-hjcz 0/1    Pending   0           3s
ml-pipeline-persistenceagent-66f89b56d9-ffm2n   0/1    Pending   0           4s
ml-pipeline-scheduledworkflow-57445ddf88-xtzcx 0/1    Pending   0           3s
ml-pipeline-ui-5c64b6c666-vvw59                0/1    Pending   0           4s
ml-pipeline-viewer-controller-deployment-7cc8d77468-6j4vg 0/1    Pending   0           3s
mysql-749f87bff5-tgsjf                         0/1    Pending   0           4s
notebook-controller-deployment-6c887454f7-irc4b 0/1    Pending   0           9s
profiles-deployment-bd576fd8f-4b4wf            0/2    Pending   0           3s
pytorch-operator-84c58df794-wnqw9             0/1    Pending   0           9s
seldon-operator-controller-manager-0           0/1    Pending   0           6s
spartakus-volunteer-9768df654-ckbxx           0/1    Pending   0           7s
tensorboard-6544748d94-46hnb                   0/1    Pending   0           6s
tf-job-operator-db676465c-hxcnf                0/1    Pending   0           6s
workflow-controller-67648d796-gsr4w           0/1    Pending   0           11s
amaramusic@handson:~/handson-kubeflow$
```

```
amaramusic@handson-kubeflow:~/handson-kubeflow$ k get po -n kubeflow
NAME                                READY   STATUS    RESTARTS   AGE
admission-webhook-bootstrap-stateful-set-0      1/1    Running   0           9m48s
admission-webhook-deployment-78d899bf68-w2bld   1/1    Running   0           8m16s
application-controller-stateful-set-0          1/1    Running   0           9m49s
argo-ui-55b859f7d7-7zhfl                       1/1    Running   0           9m48s
centraldashboard-75474d6f94-fb68r             1/1    Running   0           9m48s
jupyter-web-app-deployment-6c8f4c8997-22fw8     1/1    Running   0           9m47s
katib-controller-7ddd4c8b8c-9p7jl             1/1    Running   1           9m43s
katib-db-7b679f6f8c-s2x7r                     1/1    Running   0           9m43s
katib-manager-84c4fb876b-44fxx                 1/1    Running   2           9m43s
katib-ui-5d454c75c7-dspgk                      1/1    Running   0           9m42s
kfserving-controller-manager-0                 2/2    Running   1           9m45s
metadata-db-5dd459cc-gh819                     1/1    Running   0           9m47s
metadata-deployment-b745d8bcf-c71jj            1/1    Running   0           9m47s
metadata-deployment-b745d8bcf-hn87w           1/1    Running   0           9m47s
metadata-envoy-deployment-7ccf5c4f74-7gmzv      1/1    Running   0           9m47s
metadata-grpc-deployment-6496f66c8c-fmmsw      1/1    Running   4           9m47s
metadata-grpc-deployment-6496f66c8c-jb5s       1/1    Running   4           9m47s
metadata-ui-78f5b59b56-p4fp8                   1/1    Running   0           9m47s
minio-6f48db9cc4-rvbb6                         1/1    Running   0           9m42s
ml-pipeline-844645fd-629zm                     1/1    Running   0           9m42s
ml-pipeline-ml-pipeline-visualizationserver-865894f5f7-2ndt9 1/1    Running   0           9m41s
ml-pipeline-persistenceagent-66f89b56d9-sg5gn   1/1    Running   0           9m42s
ml-pipeline-scheduledworkflow-57445ddf88-pp5sr 1/1    Running   0           9m41s
ml-pipeline-ui-5c64b6c666-fwgdb               1/1    Running   0           9m41s
ml-pipeline-viewer-controller-deployment-7cc8d77468-k25qn 1/1    Running   0           9m41s
mysql-749f87bff5-18gwg                         1/1    Running   0           9m42s
notebook-controller-deployment-6c887454f7-c2df2 1/1    Running   0           9m46s
profiles-deployment-bd576fd8f-9bwx            2/2    Running   0           9m40s
pytorch-operator-84c58df794-jrqfr             1/1    Running   0           9m46s
seldon-operator-controller-manager-0           1/1    Running   1           9m42s
spartakus-volunteer-9768df654-2jjgg           1/1    Running   0           9m44s
tensorboard-6544748d94-g2p6                    1/1    Running   0           9m44s
tf-job-operator-db676465c-s98mm                1/1    Running   0           9m44s
workflow-controller-67648d796-r9g8l           1/1    Running   0           9m48s
amaramusic@handson-kubeflow:~/handson-kubeflow$
```

모든 컨테이너가 다 올라가는데 10분 정도 걸립니다.

1. Deploying Kubeflow on Existing Clusters

k9s

<https://github.com/derailed/k9s>

from release ver.

```
$ wget https://github.com/derailed/k9s/releases/download/0.9.3/k9s_0.9.3_Linux_x86_64.tar.gz
```

from brew

```
$ brew install derailed/k9s/k9s
```

```
Context: kubernetes-admin@kubern... <0> all      <ctrl-d> Delete      <s> Shell
Cluster: kubernetes                <1> default  <d> Describe   <y> YAML
User:    kubernetes-admin          <e> Edit
K9s Rev: 0.9.3                     <ctrl-k> Kill
K8s Rev: v1.15.7                   <l> Logs
CPU:    n/a                         <shift-l> Logs Previous
MEM:    n/a                         <ctrl-s> Save
```



Pod(all)[63]

NAMESPACE↑	NAME	READY	STATUS	RS	CPU	MEM	%CPU	%MEM	IP	NODE
auth	dex-d468d6b64-rcq25	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
cert-manager	cert-manager-cainjector-74966df6bf-mkgnk	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
cert-manager	cert-manager-f9ddb-8pbmd	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
cert-manager	cert-manager-webhook-68d879df6-sngh7	0/1	ContainerCreating	2	n/a	n/a	n/a	n/a	n/a	kubeflow
default	nfs-client-prov-nfs-client-provisioner-5785d564f-dxgdd	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
istio-system	authservice-8d74b5cb6-nq2hg	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
istio-system	grafana-86f89dbd84-tt44w	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
istio-system	istio-citadel-74966f47d6-rz55q	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
istio-system	istio-cleanup-secrets-1.1.6-9q9cj	0/1	Completed	0	n/a	n/a	n/a	n/a	192.168.85.146	kubeflow
istio-system	istio-egressgateway-5c64d575bc-lgm9z	0/1	ContainerCreating	1	n/a	n/a	n/a	n/a	n/a	kubeflow
istio-system	istio-galley-784b9f6d75-76bqj	0/1	ContainerCreating	0	n/a	n/a	n/a	n/a	n/a	kubeflow

1. Deploying Kubeflow on Existing Clusters

Kubeflow Dashboard (1)

istio에 설치된 ingressgateway port를 확인

\$ k get svc -n istio-system

```
istio-galley ClusterIP 10.101.35.228 <none> 443/TCP,15014/TCP,9901/TCP
istio-ingressgateway NodePort 10.105.19.36 <none> 15020:30186/TCP,80:31380/TCP,443:31390/TCP,31400:31400/TCP,15029:30580/TCP,15030:32573/
32363/TCP,15032:31217/TCP,15443:30641/TCP 42m
istio-pilot ClusterIP 10.108.12.33 <none> 15010/TCP,15011/TCP,8080/TCP,15014/TCP
```

(gcp일 경우) 여기서 확인된 31380, 31390을 vm의 네트워크 세부정보 - 방화벽에 추가 해줍니다.

The screenshot shows the configuration for a VM network interface in GCP. The network interface is named 'nic0' and is connected to the 'default' VPC network. The configuration page shows various settings, including network tier (Premium), IP address (10.0.0.0/24), and firewall rules. A red box highlights the '세부정보 보기' (View details) button for the firewall rules. Below this, a table lists the firewall rules:

Rule Name	Direction	Target	IP Range	Ports	Priority	Allow
default-allow-https	Ingress	all VMs	0.0.0.0/0	tcp:443	1000	허용
healthy	Ingress	all VMs	IP 범위: 0.0.0.0/0	tcp:6789	1000	허용
kubeflow-istio-http	Ingress	all VMs	IP 범위: 0.0.0.0/0	tcp:31380,31390	1000	허용
kubernetes	Ingress	all VMs	IP 범위: 0.0.0.0/24	tcp:6443	1000	허용

1. Deploying Kubeflow on Existing Clusters

Kubeflow Dashboard (2)



브라우저에서 {vm IP}:31380로 접속하면,

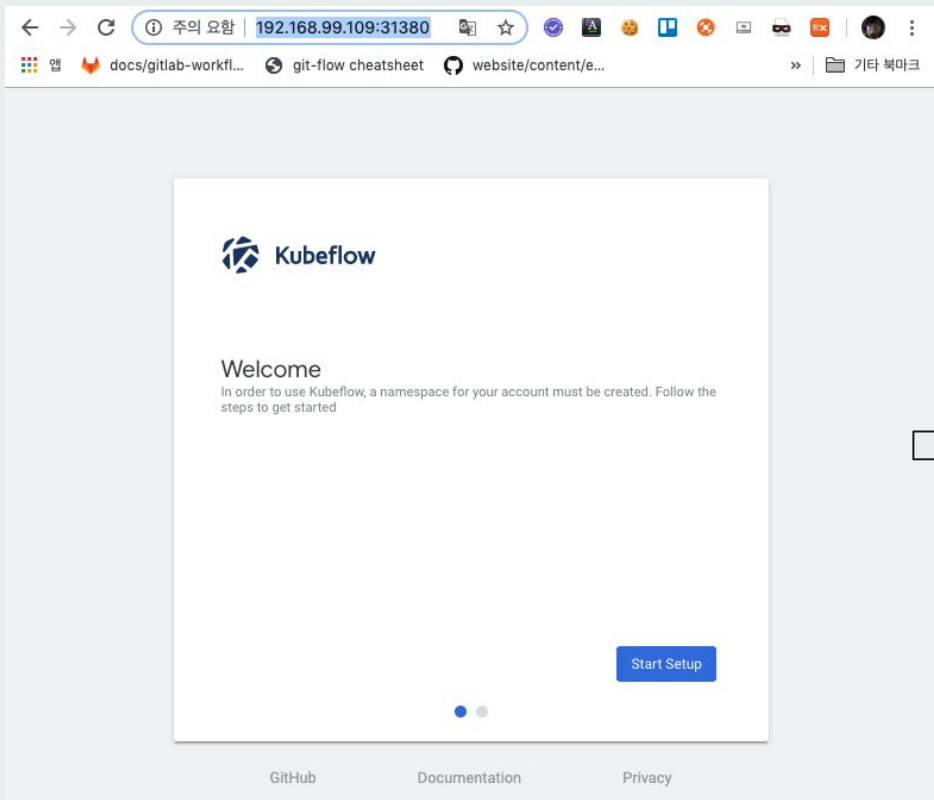
user 생성을 위한 `automatic profile creations`이 진행됩니다.

여기선 `test` 로 생성합니다.

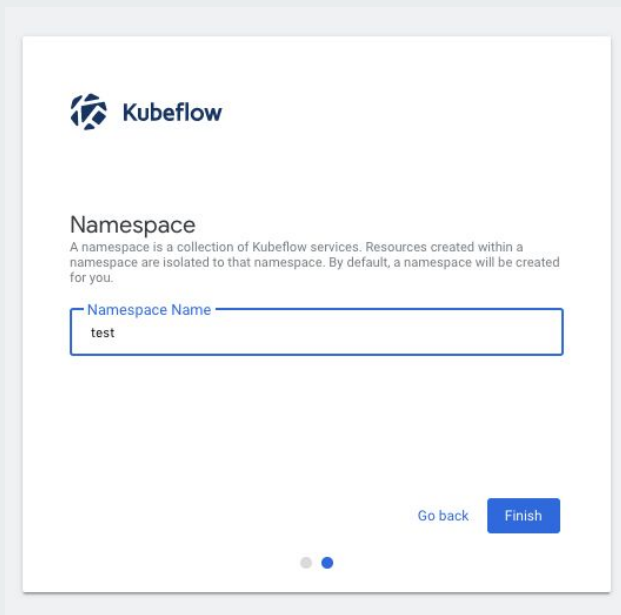
(`test`라는 namespace 와 profile이 생성됩니다.)

1. Deploying Kubeflow on Existing Clusters

Kubeflow Dashboard (3)



A screenshot of a web browser displaying the Kubeflow dashboard. The browser's address bar shows the URL `192.168.99.109:31380`. The page features the Kubeflow logo at the top left, followed by the heading "Welcome" and a sub-heading "In order to use Kubeflow, a namespace for your account must be created. Follow the steps to get started". A blue button labeled "Start Setup" is positioned at the bottom right of the main content area. At the very bottom of the page, there are three links: "GitHub", "Documentation", and "Privacy".



A screenshot of the Kubeflow dashboard's "Namespace" configuration page. It features the Kubeflow logo at the top left, the heading "Namespace", and a descriptive paragraph: "A namespace is a collection of Kubeflow services. Resources created within a namespace are isolated to that namespace. By default, a namespace will be created for you." Below this text is a text input field labeled "Namespace Name" with the value "test" entered. At the bottom right, there are two buttons: "Go back" and "Finish".

1. Deploying Kubeflow on Existing Clusters

Kubeflow Dashboard (4)

profile을 정의 하고 나면 대쉬보드를 확인 하실 수 있습니다!

The screenshot displays the Kubeflow Dashboard interface. On the left is a dark blue navigation sidebar with the following items: Home, Pipelines, Notebook Servers, Katib, Artifact Store, GitHub (with an external link icon), and Documentation (with an external link icon). At the bottom of the sidebar are links for Privacy and Usage Reporting, and the text 'build version v1beta1'. The main content area has a header with the Kubeflow logo and a dropdown menu set to 'test'. Below the header are two tabs: 'Dashboard' (active) and 'Activity'. The dashboard contains several widgets: 'Quick shortcuts' with five lightning bolt icons and links to 'Upload a pipeline', 'View all pipeline runs', 'Create a new Notebook server', 'View Katib Studies', and 'View Metadata Artifacts'; 'Recent Notebooks' showing 'No Notebooks in namespace test'; 'Recent Pipelines' listing five sample pipelines with their creation times; and 'Documentation' with links to 'Getting Started with Kubeflow', 'MiniKF', 'Microk8s for Kubeflow', 'Minikube for Kubeflow', 'Kubeflow on GCP', 'Kubeflow on AWS', and 'Requirements for Kubeflow'.

1. Deploying Kubeflow on Existing Clusters

Container Registry for kubeflow (1)

Local registry in kubernetes

쿠버네티스에서 local registry를 deploy하는 방법입니다.
kubernetest svc가 제공하는 domain을 레지스트리 도메인으로 사용합니다.

<https://github.com/mojokb/handson-kubeflow/blob/master/registry/kubeflow-registry-deploy.yaml>

<https://github.com/mojokb/handson-kubeflow/blob/master/registry/kubeflow-registry-svc.yaml>

registry url : kubeflow-registry.default.svc.cluster.local:30000

클러스터 호스트에서 lookup 될 수 있게 /etc/hosts 에서도 추가해 줍니다.

```
127.0.0.1 localhost kubeflow-registry.default.svc.cluster.local
```

1. Deploying Kubeflow on Existing Clusters

Container Registry for kubeflow (2)

External registry

공인인증서가 적용된 registry가 있다면 사용하시면 되고, docker-hub를 사용하셔도 좋습니다!
레지스트리를 사용하기 위해서 만드신 프로필의 서비스어카운트에 imagePullSecrets 정보를 등록합니다.
여기서 예제로 registry.zipsacoding.com * 을 사용합니다.

```
# in cluster
```

```
$ docker login registry.zipsacoding.com
```

```
$ kubectl create secret generic regcred --from-file=.dockerconfigjson=/home/amaramusic/.docker/config.json \
--type=kubernetes.io/dockerconfigjson -n test
```

```
$ kubectl patch serviceaccount default -p '{"imagePullSecrets": [{"name": "regcred"}]}' -n test
```

```
$ kubectl patch serviceaccount default-editor -p '{"imagePullSecrets": [{"name": "regcred"}]}' -n test
```

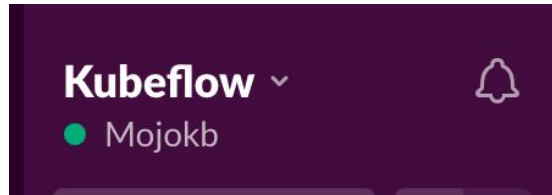
```
$ kubectl patch serviceaccount default-viewer -p '{"imagePullSecrets": [{"name": "regcred"}]}' -n test
```

또한 노트북에서 .docker/config.json 파일도 생성해줍니다. (Notebook 생성 후 작업)

config.json 의 내용을 노트북의 /home/jovyan/.docker/config.json 로 옮겨줍니다.

* registry.zipsacoding.com : handson을 위해 만든 프라이빗 레지스트리입니다 .

More questions? Let's talk!
kubeflow.slack.com/#kubeflow-korea-chat



#kubeflow-korea-chat
☆ | 👤 9 | 🌟 0 | ✎ Add a topic



사람이 없어서
인르워요