

머신러닝 파이프라인

# 쿠베플로우 소개

송호연



# 목차

## 쿠베플로우 소개

1-1. 쿠베플로우 개요

1-2. 쿠베플로우 기본 개념

1-3. 쿠베플로우 설치



# 학습목표



## 쿠베플로우 소개



### 01. 쿠베플로우 개요에 대해 이해한다.

쿠베플로우라는 기술이 왜 중요한 지, 어떤 기능을 갖고 있는지 이해한다.



### 02. 쿠베플로우의 기본 개념을 이해한다.

쿠베플로우의 기본 개념들에 대해 공부한다.



### 03. 쿠베플로우 설치 방법에 대해 이해한다.

쿠베플로우 설치를 따라해보면서 익혀봅니다.

# 쿠버플로우 개요



01

# 쿠베플로우 개요

## 쿠베플로우가 필요한 이유



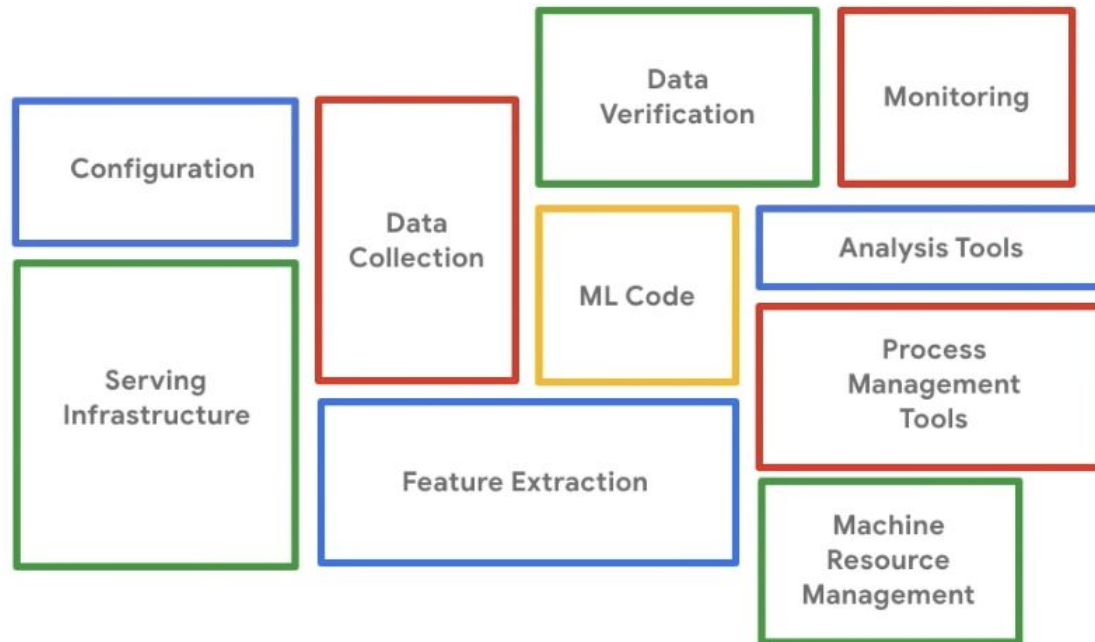
\*출처 : 출처 작성



# 쿠베플로우 개요



## 쿠베플로우가 필요한 이유



\*출처 : 출처 작성

# ● 쿠베플로우 개요

## ○ 쿠베플로우가 필요한 이유



\*출처 : 출처 작성



# 쿠베플로우 개요



## 쿠베플로우의 디자인과 핵심 컴포넌트

쿠베플로우는 다음 3가지 기능으로 정의할 수 있습니다.

### 1. 조합가능성 **Composability**

쿠베플로우의 핵심 구성 요소는 이미 머신러닝 실무자들에게 익숙한 데이터 과학 도구를 사용합니다. 이들은 기계 학습의 특정 단계를 용이하게 하기 위해 독립적으로 사용되거나, 엔드 투 엔드 파이프라인을 형성하기 위해 함께 구성될 수 있다.

### 2. 이식성 **Portability**

컨테이너 기반 설계를 갖추고 **Kubernetes** 및 클라우드 네이티브 아키텍처를 활용함으로써 **Kubeflow**는 특정 개발환경에 종속될 필요가 없습니다. 랩톱에서 실험 및 프로토타입 작업을 수행할 수 있으며, 프로덕션 환경에 손쉽게 배포할 수 있습니다.

### 3. 확장성 **Scalability**

**Kubernetes**를 사용하면 기본 컨테이너와 기계의 수와 크기를 변경하여 클러스터의 요구에 따라 동적으로 확장할 수 있습니다.





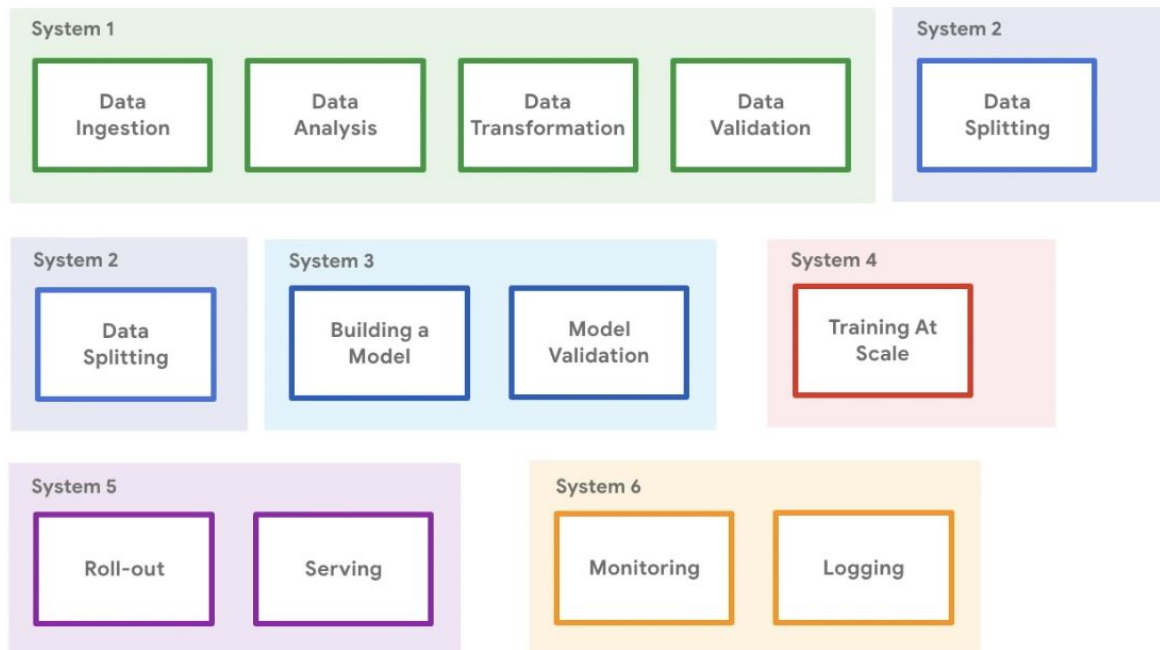
# 쿠베플로우 개요



## 조합가능성

조합가능성 Composability

### Each ML Stage is an Independent System



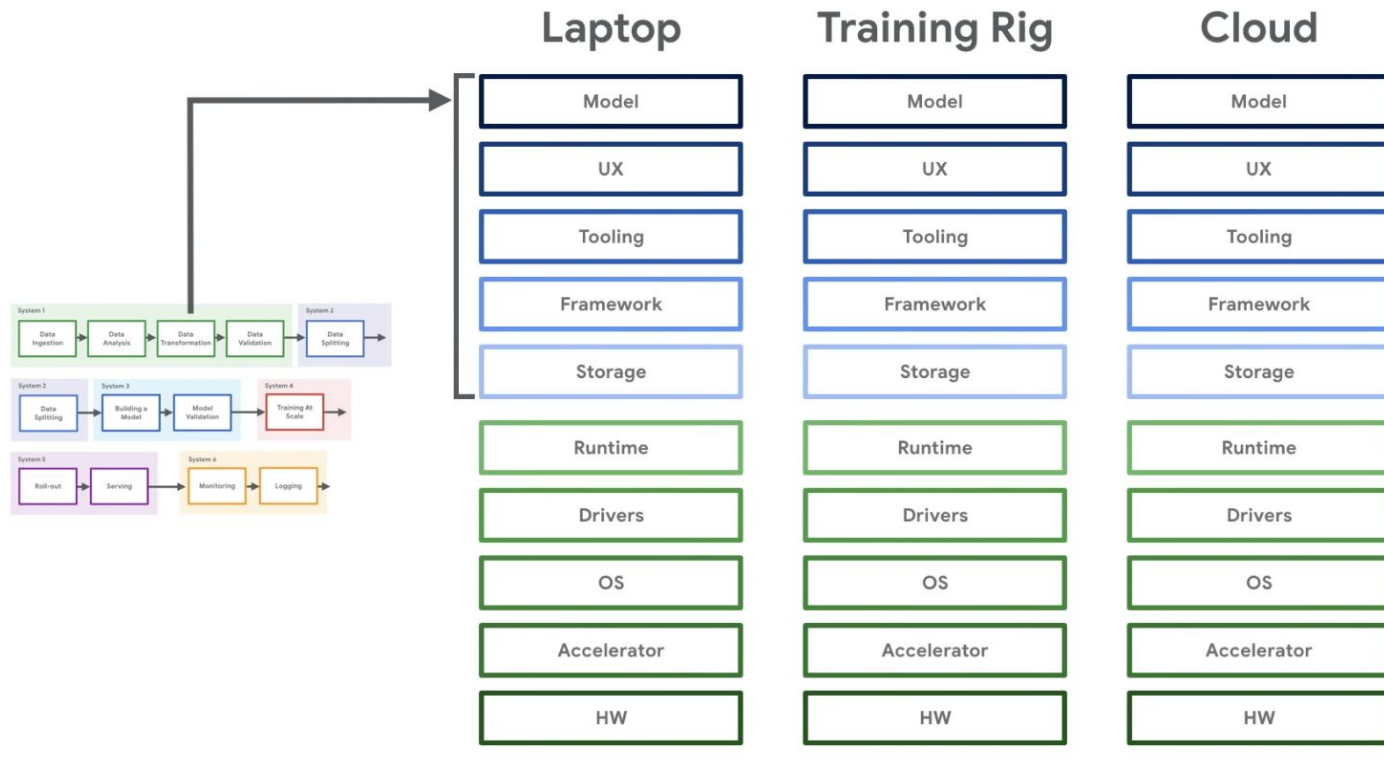


# 쿠베플로우 개요



## 이식성

### 이식성 Portability



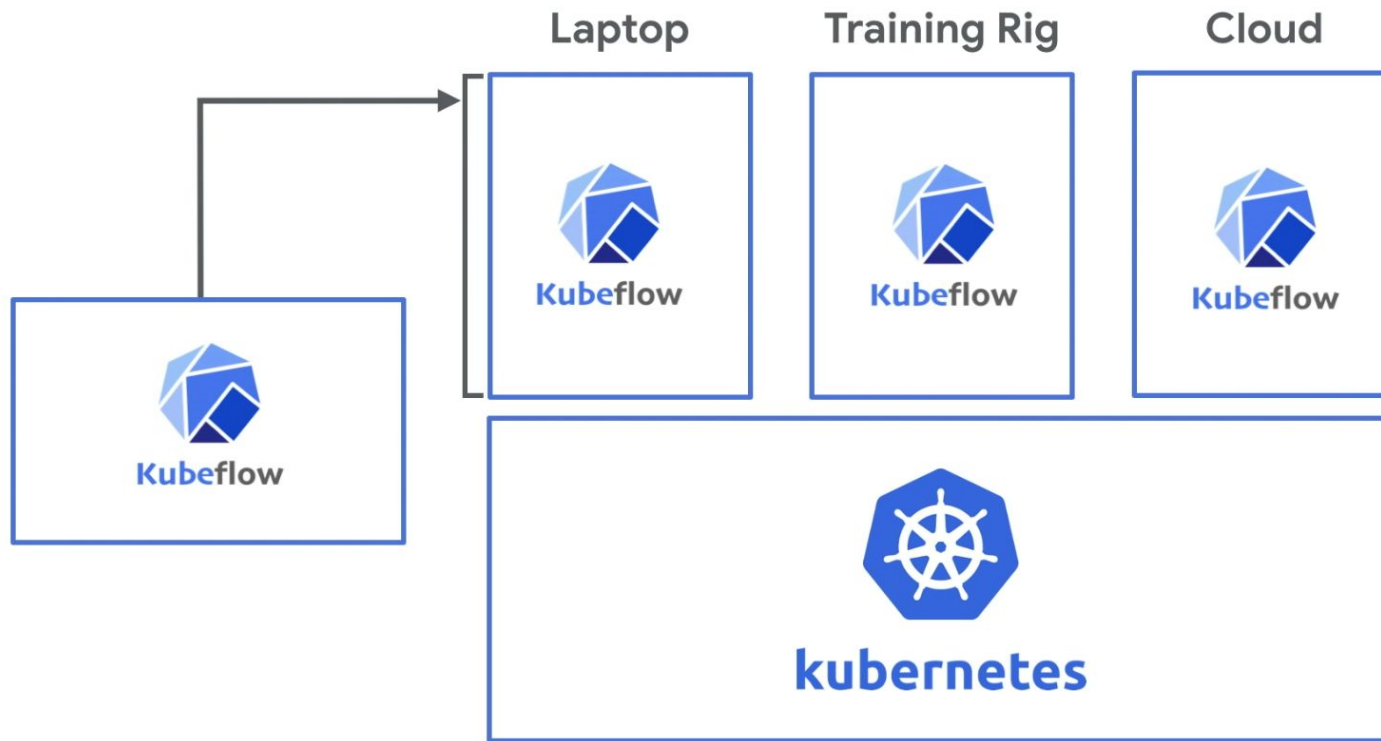


# 쿠베플로우 개요



## 이식성

이식성 Portability



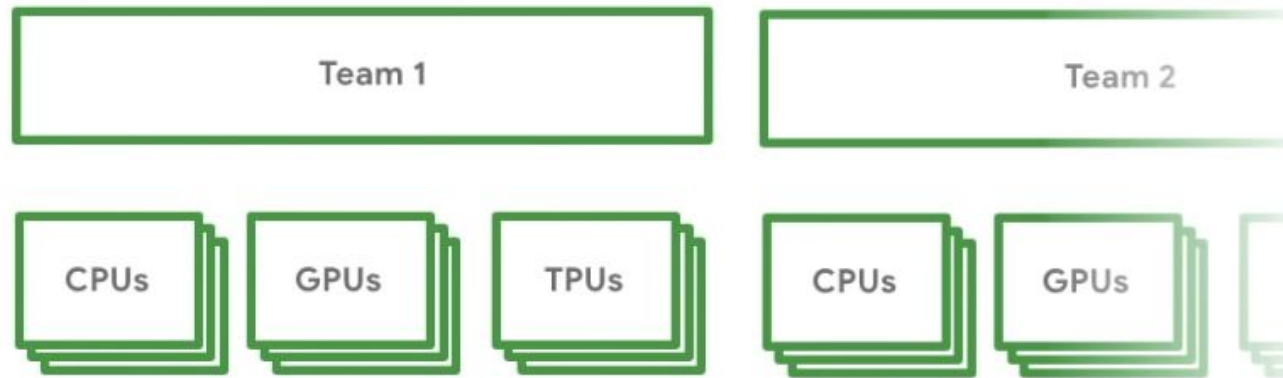


# 쿠베플로우 개요



## 확장성

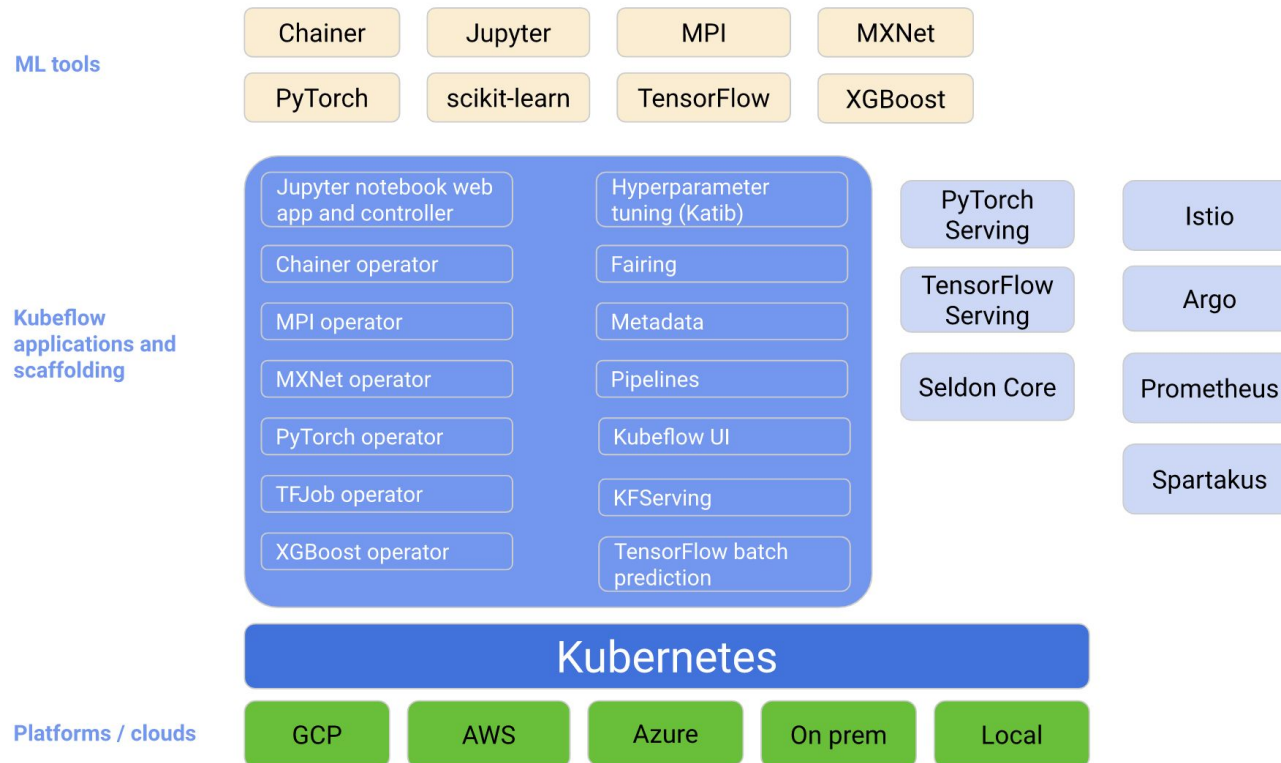
확장성 Scalability



# 쿠베플로우 개요

## 쿠베플로우가 필요한 이유

Kubeflow는 Kubernetes 용 ML 툴킷입니다. 다음 다이어그램은 Kubernetes를 기반으로 ML 시스템의 구성 요소를 배열하기 위한 플랫폼으로서 Kubeflow를 보여줍니다.



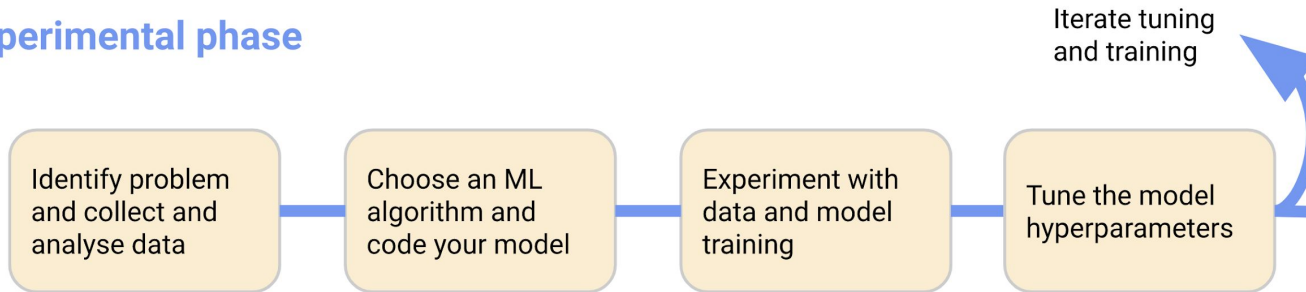
\* 출처 : 출처 작성

# 쿠베플로우 개요

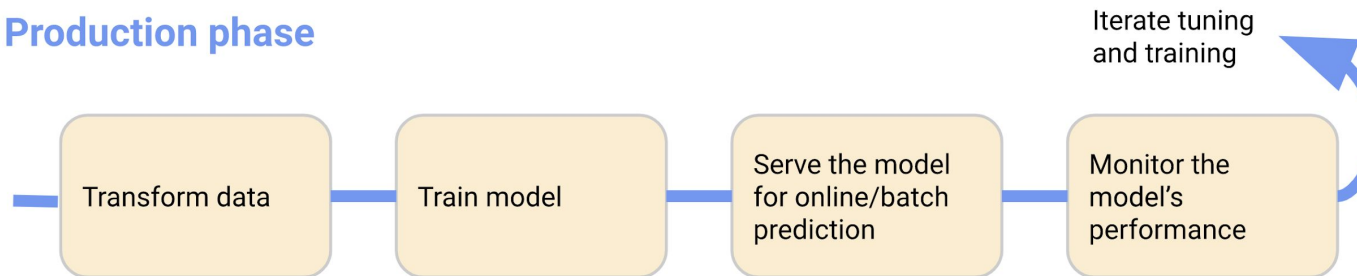
## 쿠베플로우가 필요한 이유

Kubeflow는 Kubernetes 용 ML 툴킷입니다. 다음 다이어그램은 Kubernetes를 기반으로 ML 시스템의 구성 요소를 배열하기 위한 플랫폼으로서 Kubeflow를 보여줍니다.

### Experimental phase



### Production phase



\*출처 : 출처 작성

# 쿠버플로우 기본 개념

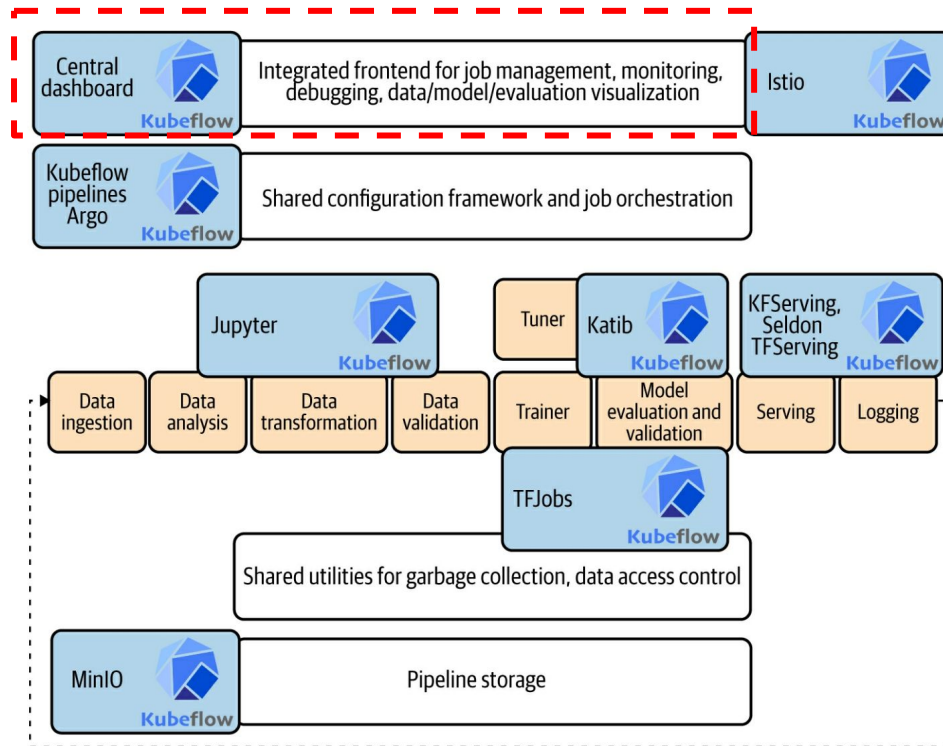


02

# 쿠베플로우 기본 개념

## 쿠베플로우 대시보드

쿠베플로우의 주요 인터페이스는 대시보드입니다.



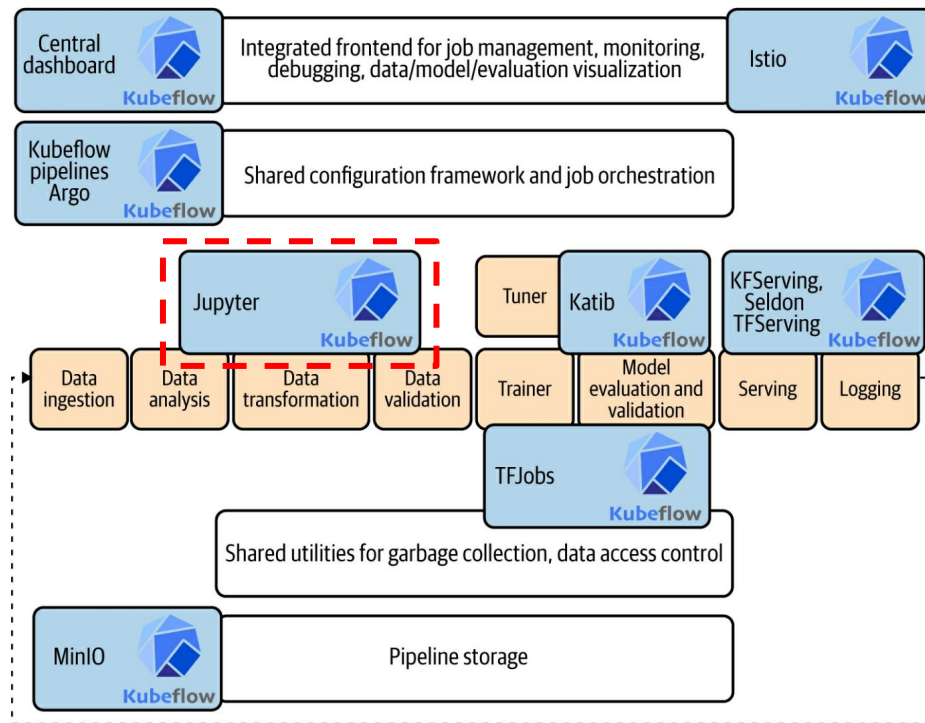
\* 출처 : 출처 작성



# 쿠베플로우 기본 개념

## 쿠베플로우 노트북

머신러닝 프로젝트의 첫 시작점은 바로 프로토타이핑과 실험입니다. 쿠베플로우의 경우에 이 단계를 담당하는 모듈이 바로 **쿠베플로우 노트북(JupyterHub)**입니다.

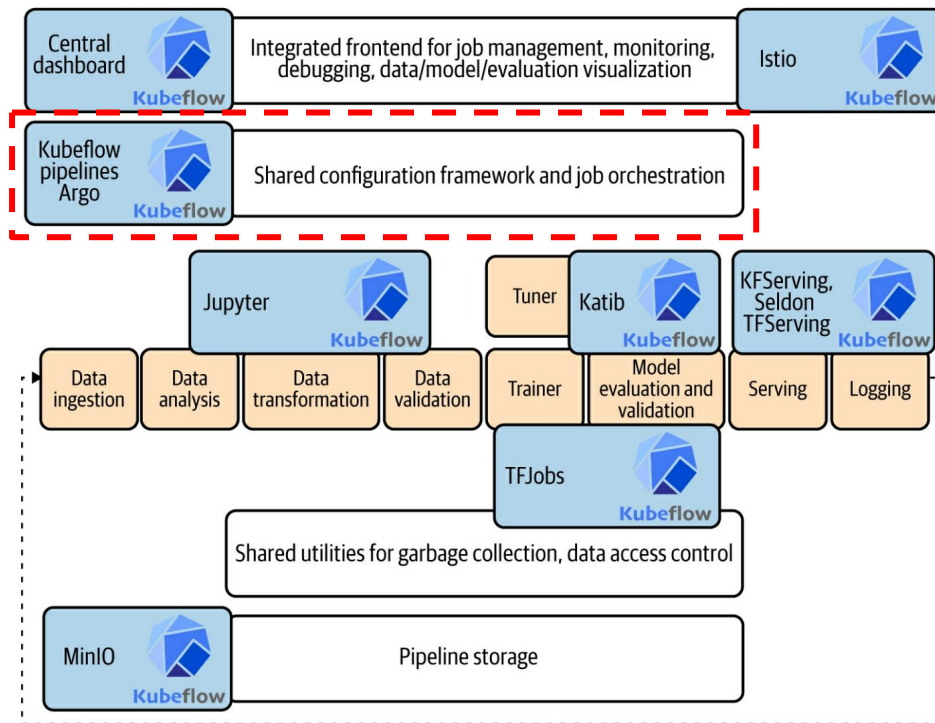


\* 출처 : 출처 작성

# 쿠베플로우 기본 개념

## 쿠베플로우 파이프라인

쿠베플로우의 핵심적인 구성요소는 바로 **쿠베플로우 파이프라인**입니다.  
파이프라인 오케스트레이션을 수행해주는 툴입니다. (Airflow와 비슷합니다.)



\* 출처 : 출처 작성



# 쿠베플로우 기본 개념



## 쿠베플로우 파이프라인

my\_python\_func 함수에다가  
**@kfp.dsl.python\_component**  
decorator를 적어주시면 됩니다.

Creating components within your application code

my\_python\_func

Decorate your function with  
`@kfp.dsl.python_component`

Pipeline component

`@kfp.dsl.python_component`  
my\_python\_func

\* 출처 : 출처 작성

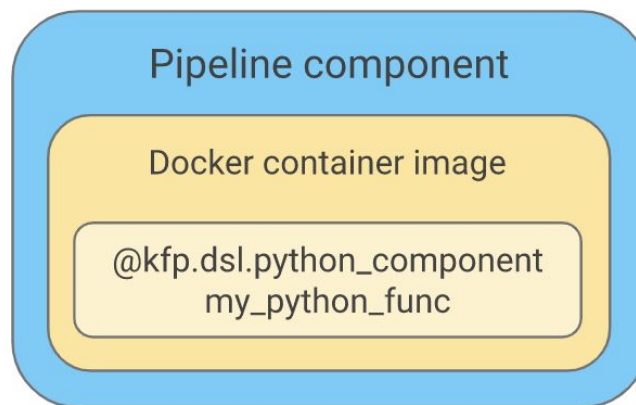
# 쿠베플로우 기본 개념

## 쿠베플로우 파이프라인

DSL decorator가 적힌 파이썬 함수를 도커 이미지로 패키징합니다.

그리고 이 도커 이미지를 활용하여 쿠베플로우 컴포넌트 <sup>Kubeflow Component</sup>를 만들 수 있습니다.

Create a container image for the component:  
`kfp.compiler.build_python_component`



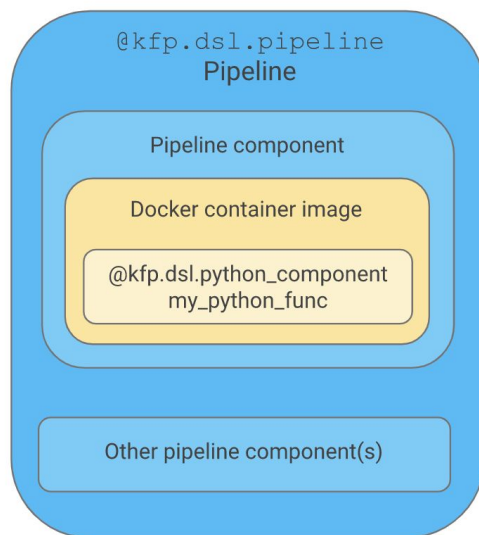
\*출처 : 출처 작성

# 쿠베플로우 기본 개념

## 쿠베플로우 파이프라인

마지막으로 `@kfp.dsl.pipeline` 데코레이터를 활용하여 해당 쿠베플로우 컴포넌트를 **파이프라인**으로 패키징 할 수 있습니다.

Define a pipeline and include all components:  
decorate your function with  
`@kfp.dsl.pipeline`



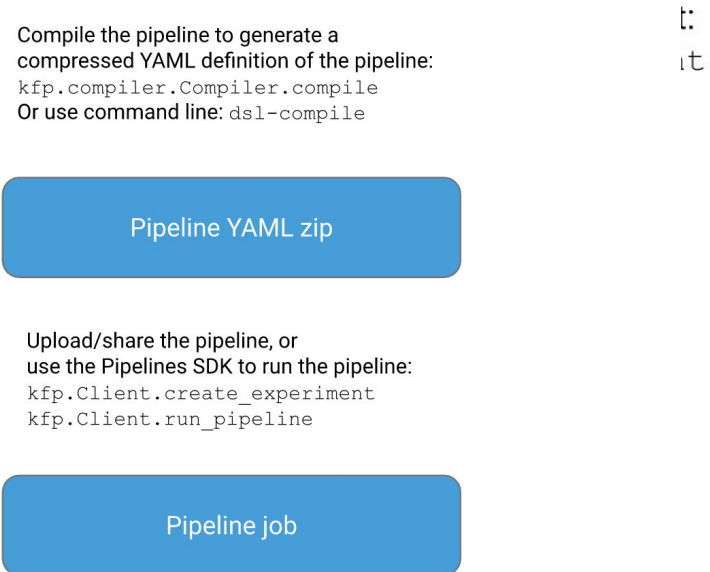
\* 출처 : 출처 작성

# 쿠베플로우 기본 개념

## 쿠베플로우 파이프라인

파이프라인 구성을 완료하면 해당 파이프라인을 **YAML** 형태로 패키징할 수 있고, 이를 "파이프라인을 컴파일"한다고 합니다.

그리고 이 **YAML** 파일을 **쿠베플로우 대시보드**에서 업로드하거나 **kfctl 명령**으로 업로드할 수 있습니다

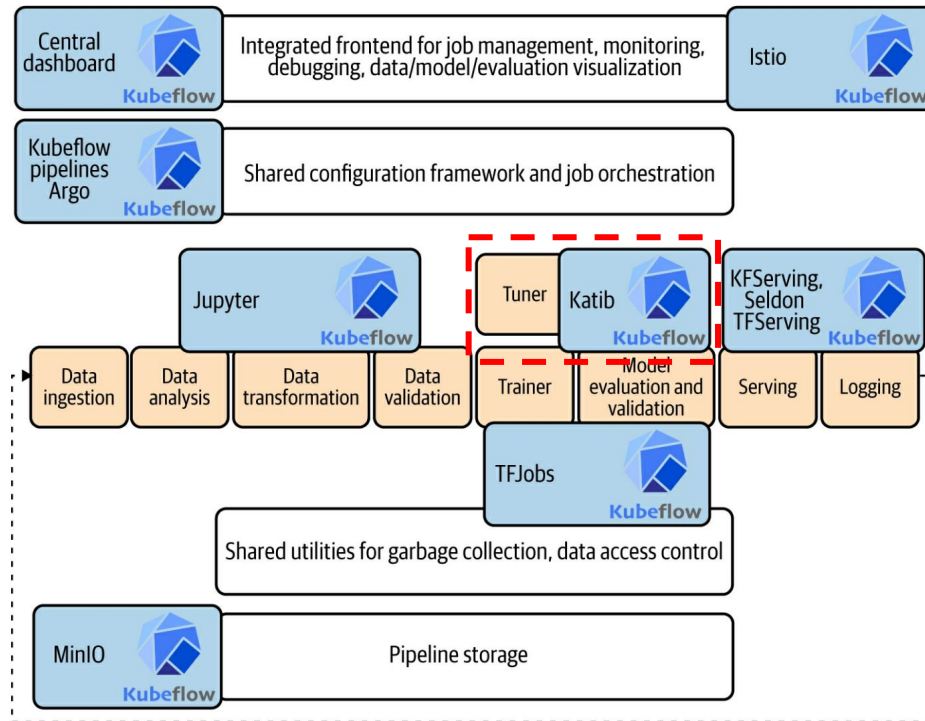


\* 출처 : 출처 작성

# 쿠베플로우 기본 개념

## 쿠베플로우 Katlib

쿠베플로우에서 모델 튜닝(AutoML)을 담당하는 모듈은 바로 **쿠베플로우 Katlib**입니다.  
기본적인 하이퍼파라미터 최적화와 여러 고도화된 툴들이 계속 추가되고 있습니다.

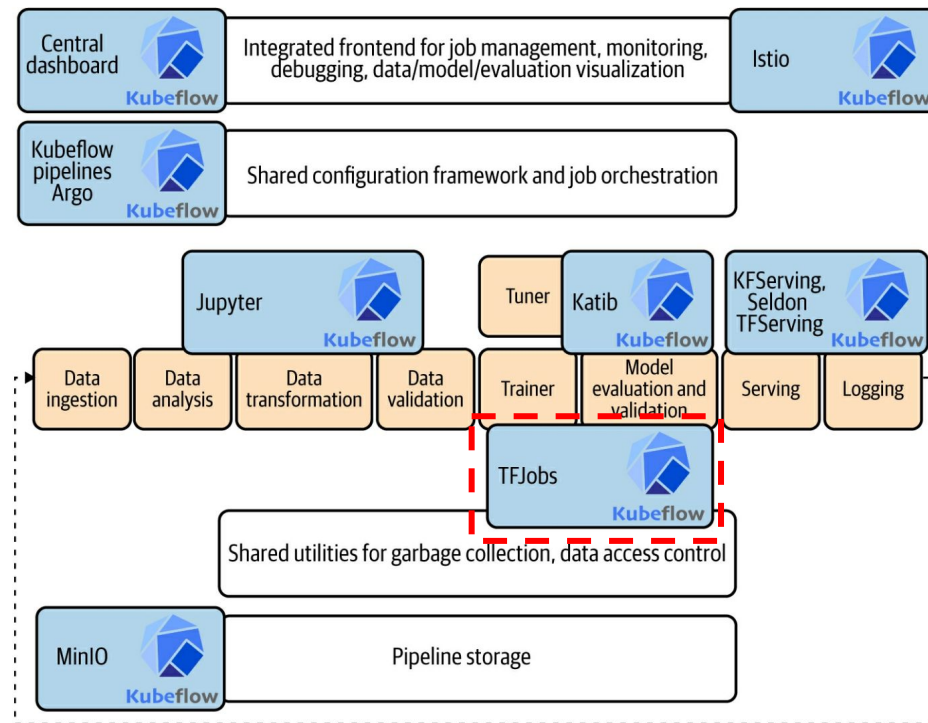


\* 출처 : 출처 작성

# 쿠베플로우 기본 개념

## 쿠베플로우 TFJobs

비동기로 학습을 하거나 오프라인 추론을 할 때 **TFJobs**를 사용할 수 있습니다.



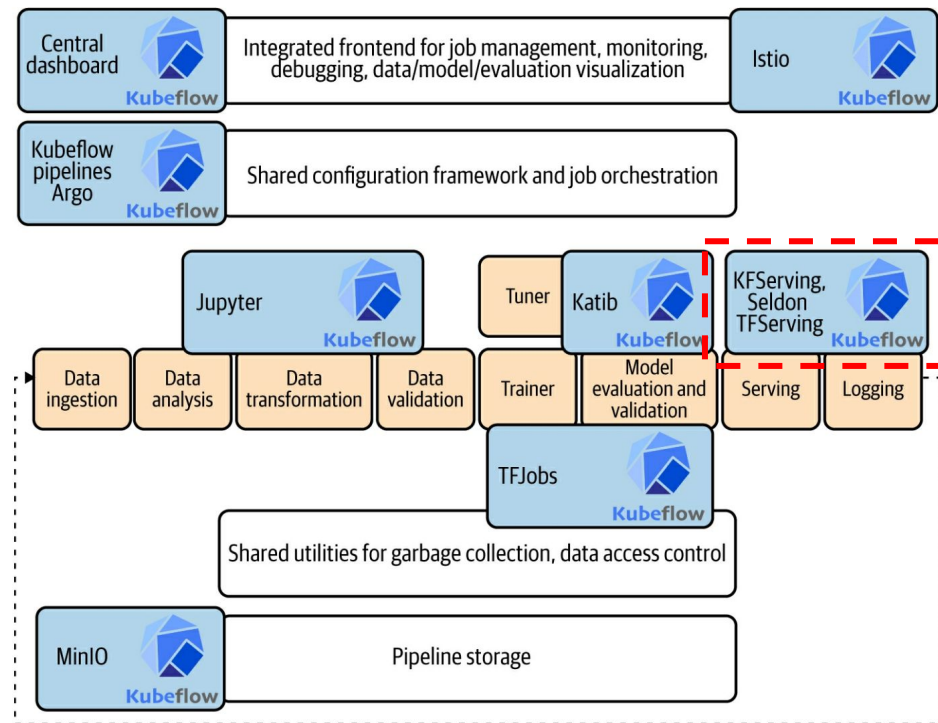
\* 출처 : 출처 작성



# 쿠베플로우 기본 개념

## 쿠베플로우 KFServing

온라인 인퍼런스 서버를 KFServing으로 배포할 수 있습니다.

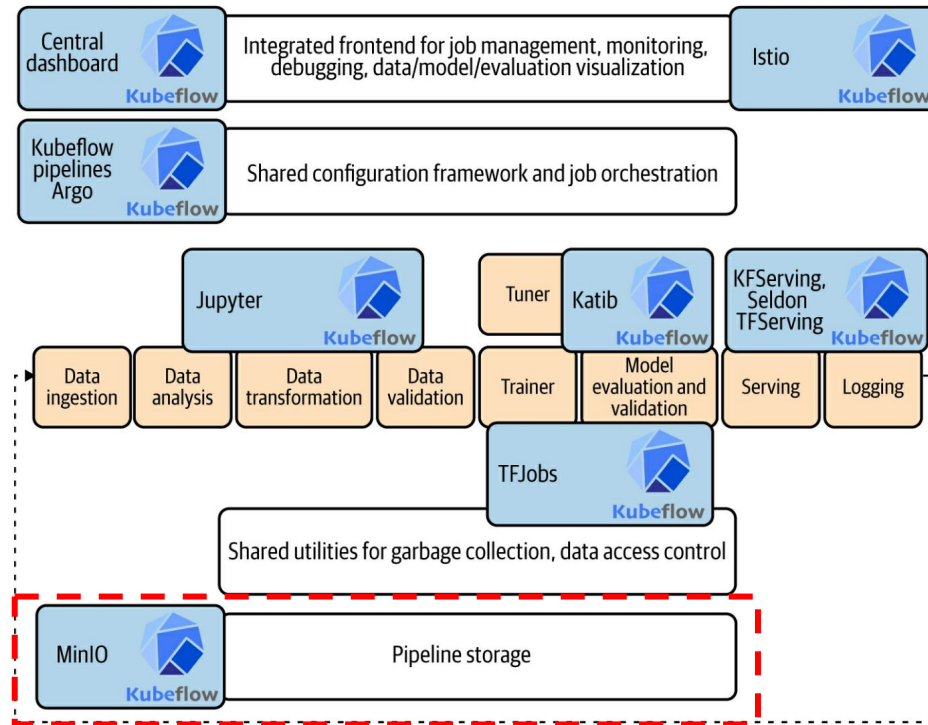


\* 출처 : 출처 작성

# 쿠베플로우 기본 개념

## 쿠베플로우 MinIO

쿠베플로우는 파이프라인 간의 저장소 기능을 **MinIO**라는 솔루션으로 구성하였습니다.  
MinIO에 파이프라인 중간에 생기는 부산물들을 저장 할 수 있습니다.



\* 출처 : 출처 작성

# 쿠버플로우 설치



03



# 쿠베플로우 설치



## 쿠베플로우 설치

### Docker for Desktop 설치

#### Mac

<https://desktop.docker.com/mac/stable/Docker.dmg>

#### Windows

<https://desktop.docker.com/win/stable/Docker%20Desktop%20Installer.exe>

\* 출처 : <https://www.docker.com/products/docker-desktop>



# 쿠베플로우 설치

## 쿠베플로우 설치

kfctl을 설치합니다. kfctl은 Kubeflow를 배포하고 관리하는 콘솔입니다.

```
export PLATFORM=$(uname) # Either Linux or Darwin
export KUBEFLOW_TAG=1.0.0
KUBEFLOW_BASE="https://api.github.com/repos/kubeflow/kfctl/releases"
# Or just go to https://github.com/kubeflow/kfctl/releases
wget https://github.com/kubeflow/kfctl/releases/download/v1.0/kfctl_v1.0-0-g94c35cf_darwin.tar.gz
KFCTL_FILE=${KFCTL_URL##*/}
tar -xvf "${KFCTL_FILE}"
sudo mv ./kfctl /usr/local/bin/
rm "${KFCTL_FILE}"
```

\* 출처 : <https://github.com/chris-chris/kubeflow-tutorial/tree/master/setup>



# 쿠베플로우 설치



## 쿠베플로우 설치

kfctl을 사용하여 istio 로 KubeFlow를 배포합니다.

```
export MANIFEST_BRANCH=${MANIFEST_BRANCH:-v1.0-branch}
export MANIFEST_VERSION=${MANIFEST_VERSION:-v1.0.0}

export KF_PROJECT_NAME=${KF_PROJECT_NAME:-hello-kf}
mkdir "${KF_PROJECT_NAME}"
pushd "${KF_PROJECT_NAME}"

manifest_root=https://raw.githubusercontent.com/kubeflow/manifests/
FILE_NAME=kfctl_k8s_istio.${MANIFEST_VERSION}.yaml
KFDEF=${manifest_root}${MANIFEST_BRANCH}/kfdef/${FILE_NAME}
kfctl apply -f $KFDEF -V
echo $?

popd
```

\* 출처 : <https://github.com/chris-chris/kubeflow-tutorial/tree/master/setup>



## 쿠베플로우 설치



### 쿠베플로우 포트포워딩

kubectl port-forward 명령으로 istio-ingressgateway를 8080 포트에 연결합니다.

```
kubectl port-forward svc/istio-ingressgateway -n istio-system 8080:80
```

\* 출처 : <https://github.com/chris-chris/kubeflow-tutorial/tree/master/setup>



# 짚어보기



## 쿠베플로우 소개



### 01. 쿠베플로우 개요에 대해 이해한다.

쿠베플로우라는 기술이 왜 중요한 지, 어떤 기능을 갖고 있는지 이해한다.



### 02. 쿠베플로우의 기본 개념을 이해한다.

쿠베플로우의 기본 개념들에 대해 공부한다.



### 03. 쿠베플로우 설치 방법에 대해 이해한다.

쿠베플로우 설치를 따라해보면서 익혀봅니다.



머신러닝 파이프라인

# 쿠베플로우 소개

송호연



# 감사합니다.

THANKS FOR WATCHING





# 쿠베플로우 설치



## 쿠베플로우 설치

<https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>

\* 출처 : <https://minikube.sigs.k8s.io/docs/start/>



## 쿠베플로우 설치



### 쿠베플로우 설치

Windows 또는 MacOS에서 Ubuntu 터미널을 얻는 간단한 방법은 Multipass 를 다운로드하여 설치 한 다음 Kubeflow 배포를 위한 최소 요구 사항과 일치하는 Multipass VM을 시작하는 것입니다.

```
multipass launch --name kubeflow --mem 16G --disk 50G --cpus 4
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



## 쿠베플로우 설치



### 쿠베플로우 설치

그런 다음 생성한 VM에 접근해보겠습니다.

```
multipass shell kubeflow
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



# 쿠베플로우 설치



## 쿠베플로우 설치

### 3. MicroK8s 설치

이 단계에서는 거의 모든 컴퓨터에서 실행하고 사용할 수 있는 최소한의 경량 Kubernetes 인 MicroK8s를 설치합니다. 다음 명령으로 설치할 수 있습니다.

```
sudo snap install microk8s --classic
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



## 쿠베플로우 설치



### 쿠베플로우 설치

그런 다음 설치 성공 여부를 확인하십시오.

```
sudo microk8s status --wait-ready
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



# 쿠베플로우 설치



## 쿠베플로우 설치

### "microk8s"그룹에 가입

MicroK8s는 관리자 권한이 필요한 명령을 원활하게 사용할 수 있도록 그룹을 생성합니다.

현재 사용자를 그룹에 추가하고 **.kube** 캐싱 디렉토리에 대한 액세스 권한을 얻으려면 다음 두 명령을 실행하십시오.

```
sudo usermod -a -G microk8s $USER
```

```
sudo chown -f -R $USER ~/.kube
```

\*출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>





# 쿠베플로우 설치



## 쿠베플로우 설치

### 4. Kubeflow 배포

마지막으로 Kubeflow 뒤에 모든 Kubernetes 서비스를 배포 할 수 있습니다.  
dns, dashboard 및 storage 서비스를 활성화하여 시작하겠습니다.

```
microk8s.enable dns dashboard storage
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



## 쿠베플로우 설치



### 쿠베플로우 설치

그런 다음 **Kubeflow**를 시작하려면 다음 명령을 실행하세요.

```
microk8s.enable kubeflow
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



## 쿠베플로우 설치



### 쿠베플로우 설치

이러한 서비스를 시작하는 데 몇 분 정도 걸릴 수 있습니다. 다음 명령을 사용하여 이 프로세스의 상태를 추적 할 수 있습니다.

```
microk8s.kubectl get all --all-namespaces
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



# 쿠베플로우 설치



## 쿠베플로우 설치

배포에 성공하면 다음과 같은 메시지가 표시됩니다.

```
Congratulations, Kubeflow is now available.  
The dashboard is available at http://10.64.140.43.xip.io/  
  
Username: admin  
Password: L0YQB2SCOZ6XXAQ0LS01W6P09KHTUG  
  
To see these values again, run:  
  
microk8s juju config dex-auth static-username  
microk8s juju config dex-auth static-password  
  
To tear down Kubeflow and associated infrastructure, run:  
  
microk8s disable kubeflow  
  
ubuntu@ip-172-31-73-59:~$ █
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



# 쿠베플로우 설치



## 쿠베플로우 설치

가상 머신에서 **Ubuntu**를 실행하는 경우 **SOCKS** 프록시를 생성해야 합니다. 다음과 같이 수행 할 수 있습니다.

- 1) 현재 세션에서 로그 아웃
- 2) 를 사용하여 머신에 다시 연결 **ssh**하고 **-D9999**매개 변수로 **SOCKS** 프록시를 활성화합니다 .

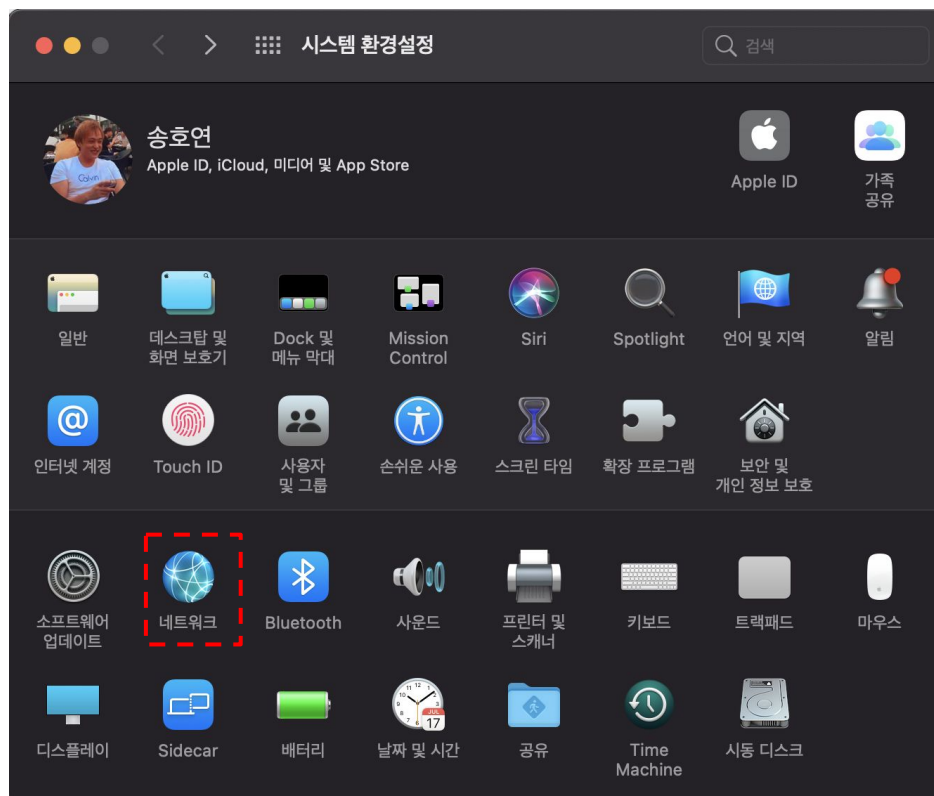
```
ssh -D9999 ubuntu@<machine_public_ip>
```

\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>

# 쿠베플로우 설치

## 쿠베플로우 설치

컴퓨터에서로 이동 **Settings > Network > Network Proxy**하여 다음을 가리키는 **SOCKS** 프록시를 활성화합니다. 127.0.0.1:9999

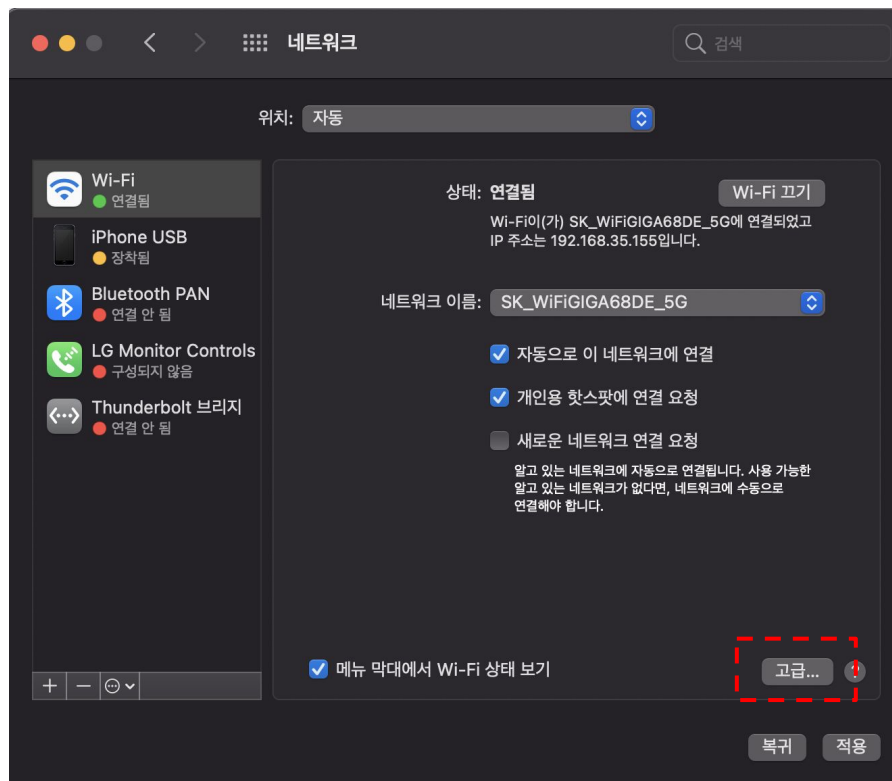


\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>

# 쿠베플로우 설치

## 쿠베플로우 설치

컴퓨터에서로 이동 **Settings > Network > Network Proxy**하여 다음을 가리키는 **SOCKS** 프록시를 활성화합니다. 127.0.0.1:9999



\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>

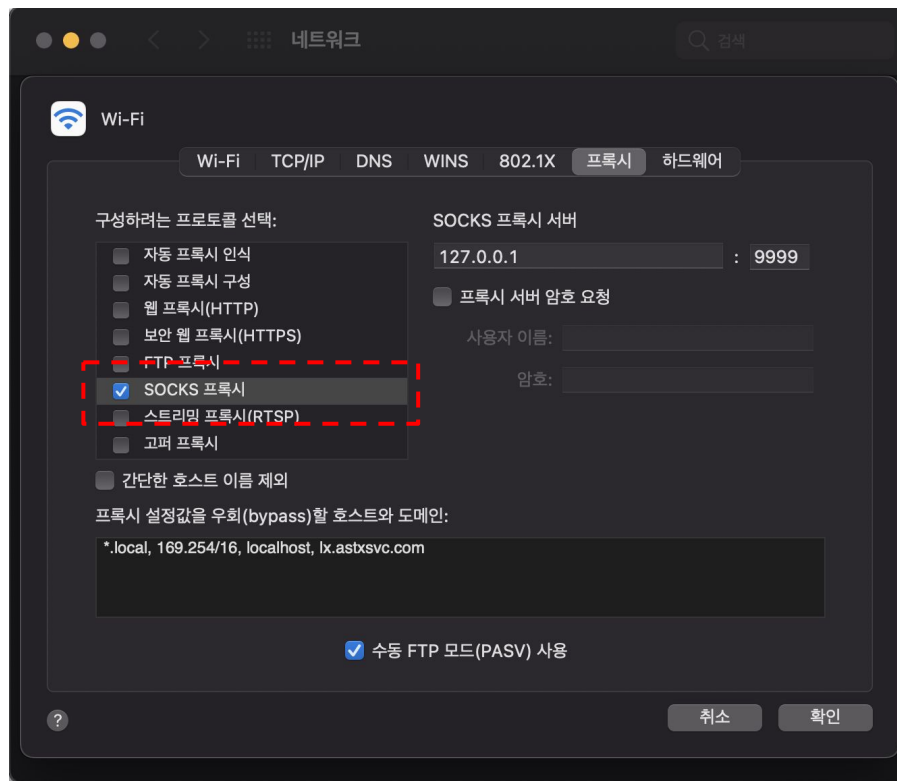


# 쿠베플로우 설치



## 쿠베플로우 설치

컴퓨터에서로 이동 **Settings > Network > Network Proxy**하여 다음을 가리키는 **SOCKS** 프록시를 활성화합니다. 127.0.0.1:9999



\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>



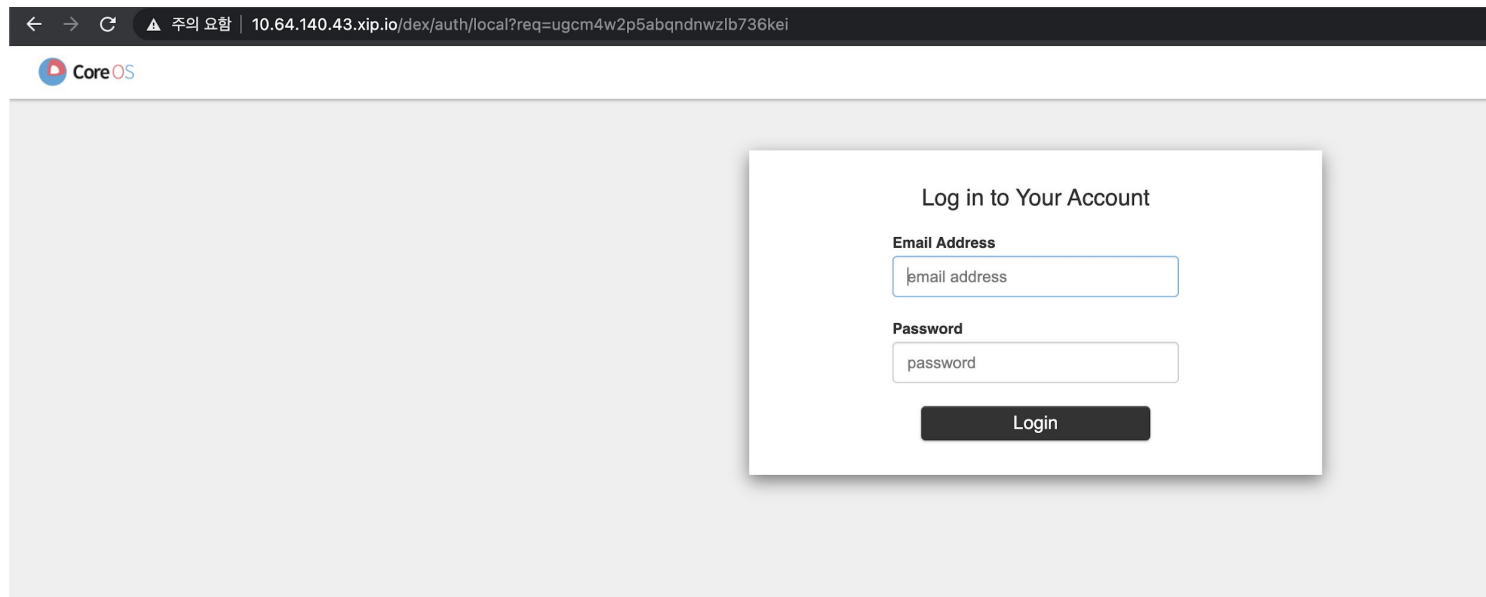


# 쿠베플로우 설치



## 쿠베플로우 설치

새 브라우저 창에서 이전 단계에서 제공된 링크에 액세스하십시오.  
이 예에서이 링크는 다음과 같습니다. <http://10.64.140.43.xip.io>

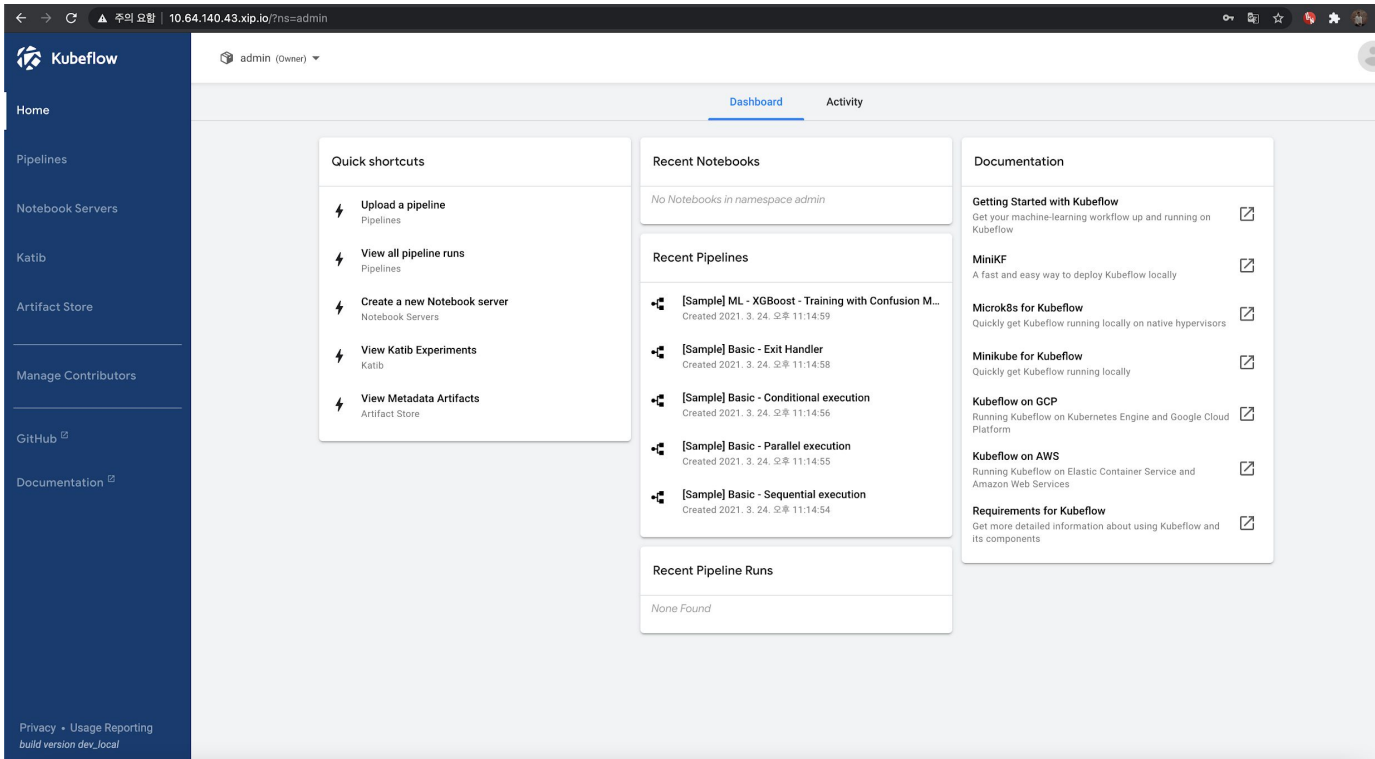


\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>

# 쿠베플로우 설치

## 쿠베플로우 설치

새 브라우저 창에서 이전 단계에서 제공된 링크에 액세스하십시오.  
이 예에서이 링크는 다음과 같습니다. <http://10.64.140.43.xip.io>



\* 출처 : <https://ubuntu.com/tutorials/deploy-kubeflow-ubuntu-windows-mac#1-overview>