KubeFlow：允许用户在Kubeflow集群中操作Tensorboard实例

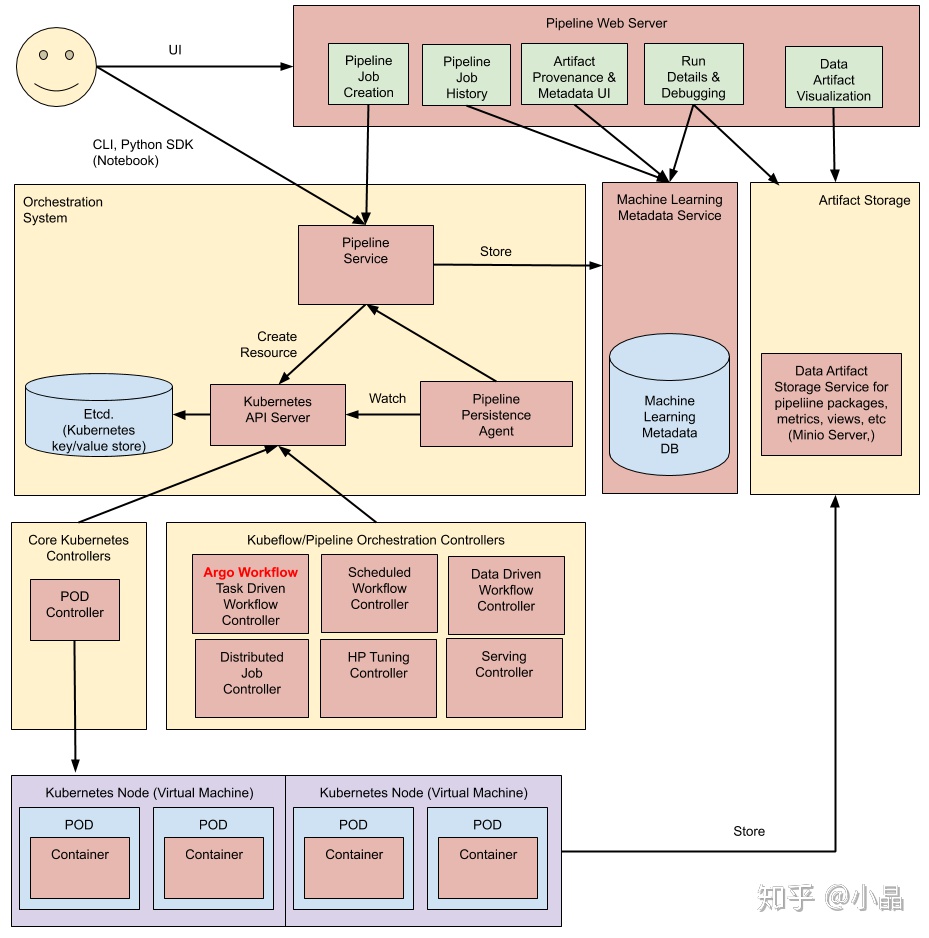
Kubernets的机器学习工具包。

KubeFlow关注实验阶段和生产阶段。

KubeFlow有一系列的组件，用于完成在不同阶段的不同环节的任务。

KubeFlow的主要组件：

1. Central Dashboard：看板（看板中包含的信息有哪些？它们的用途是什么？）
2. Metadata：用于跟踪各数据库、作业与模型（数据库的类型有哪些？数据库可接入的类型或管理的类型有哪些？作业跟踪在具体跟踪什么？模型跟踪跟踪的信息是什么？）
3. **Jupyter Notebooks**：交互式业务IDE编码环境（是否只支持这一个开发环境？如果是，那么其结果如何被约束在用于开发模型和实践模型？还是可以将它用于其他的Python应用开发？）
4. Frameworks for Training：支持的ML框架（PyTorch是主要框架，是否支持sklearn等其他框架和/或工具？）
5. Hyperparameter Tuning：Katib用于支持调参实验结果管理，是一个超参数服务器（如何管理？只是管理静态的结果？还是管 理动态的训练和静态的结果展示？如何管理不同的框架？）
6. **Pipelines**：基于Argo实现的ML工作流组件，可以定义复杂的ML工作流，进行创建、编排调度和管理ML工作流（如何定义和管理工作流？可以定义的复杂ML工作流体现在哪里？有Web UI，什么样？能做什么？是展示还是可以自定义？）



1. Tools for Serving：在Kubenets上对机器学习模型的服务化部署和离线预测的工具，
2. Multi-Tenancy in KubeFlow：KubeFlow中的多租户[租户只有使用权，没有修改权，那么KubeFlow的多租户不能修改什么？能够增加什么？]（在ML工作流的实施过程中，为何需要多租户？）
3. Fairing：用于code打包构建镜像（image）的组件（打包的标准化流程）
4. Operator：针对不同的机器学习框架提供资源调度和分布式训练的能力（PyTorch-Operator是我们的重点，是否可以扩展介入其他的框架，如sklearn？如何扩展？在调度资源和分布式训练时如何调度？以及如何调优？）