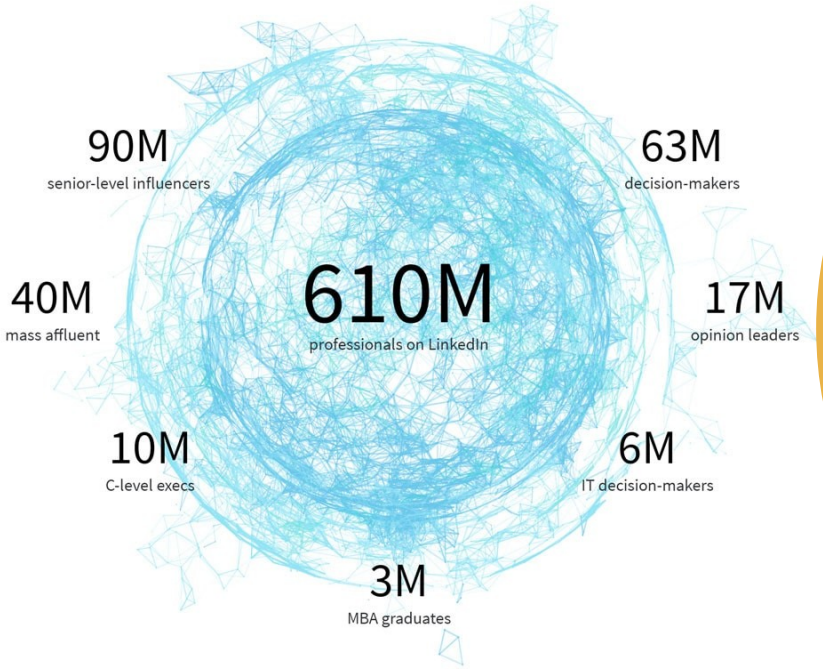
Adapting to a unified and pluggable cluster management platform

Tengfei Mu, Engineering Manager, LinkedIn
Abin Shahab, Staff Engineer, LinkedIn

Jun 25, 2019

LinkedIn



Vision

Create economic opportunity for every member of the global workforce

LPS (LinkedIn Platform as Service)

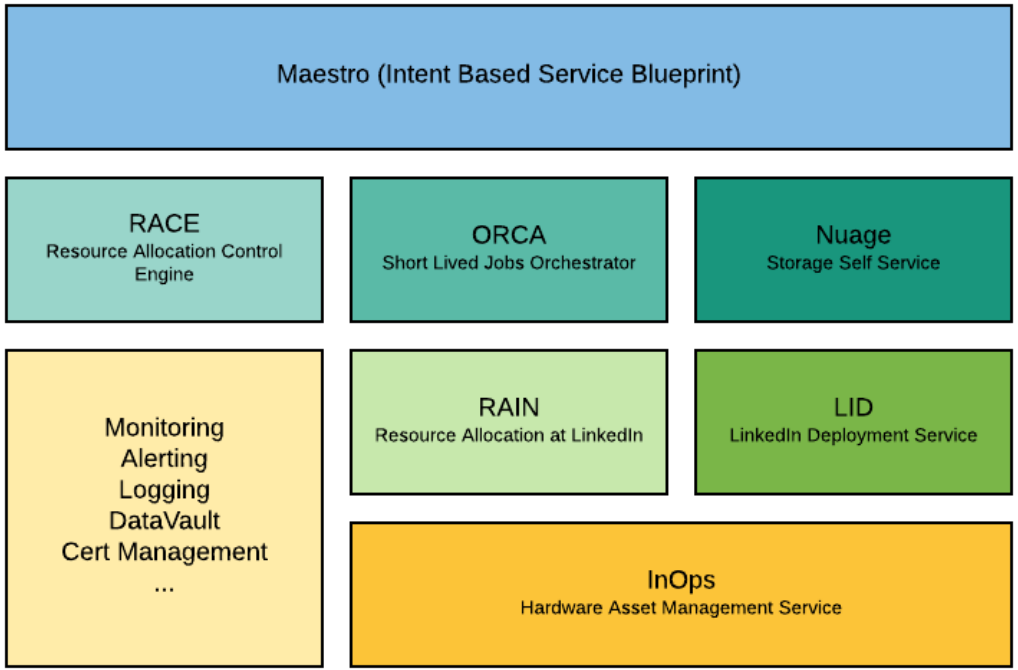
Provides a compute platform
for LinkedIn Engineers to be productive
and make efficient use of LinkedIn's
pool of hardware resources

- Developer Productivity
- Operations Productivity
- Hardware Utilization
- Enable Innovation

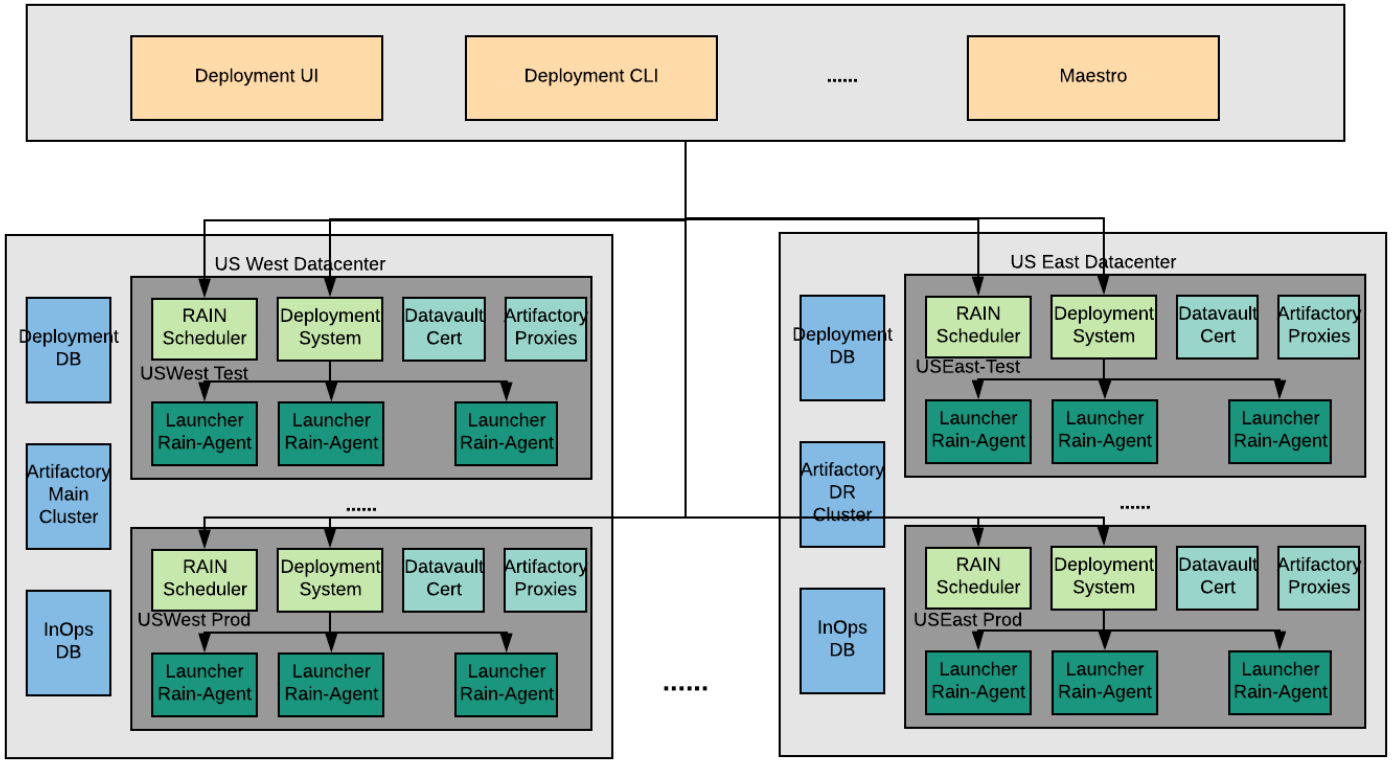
LPS Scale

- 10,000s of services
- 100,000s of builds daily
- 10,000s of deployments daily
- 100,000s of hosts
- 1,000,000s of containers

Existing Cluster Management Platform

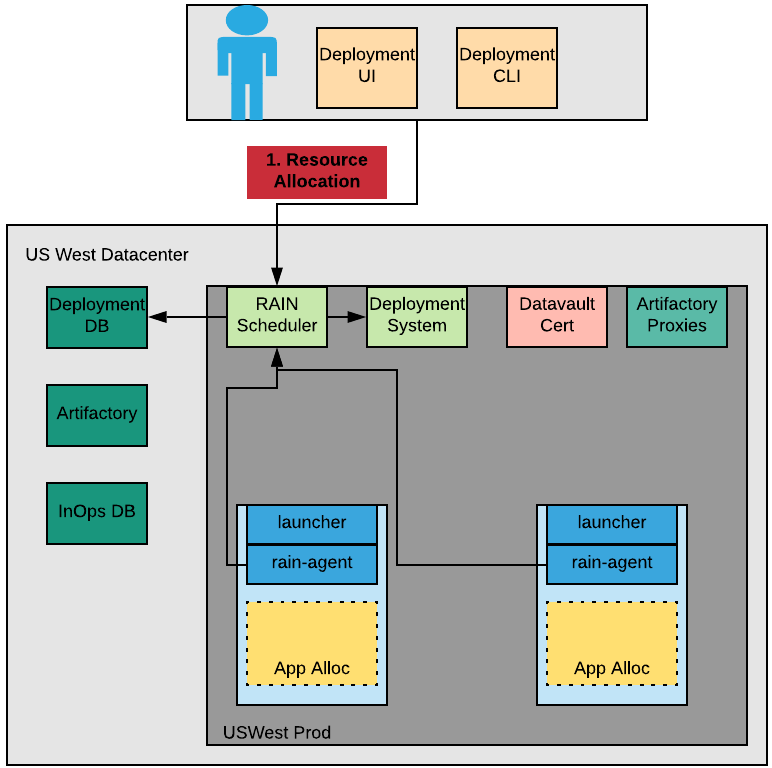


Architecture



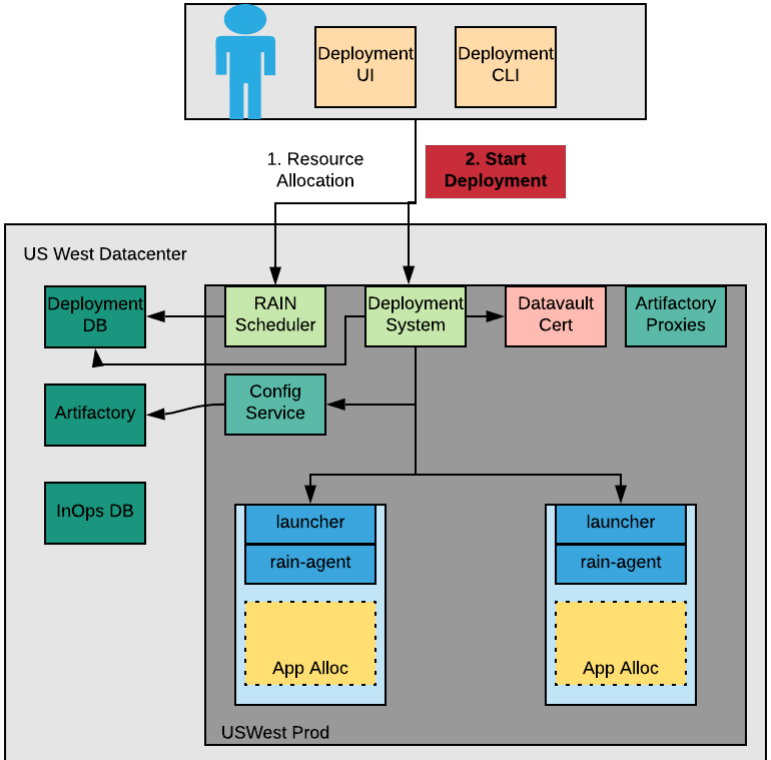
RAIN

- Allocates resources from the common available pool of resources in LinkedIn's data centers.
- Manages resource pools and host lifecycle



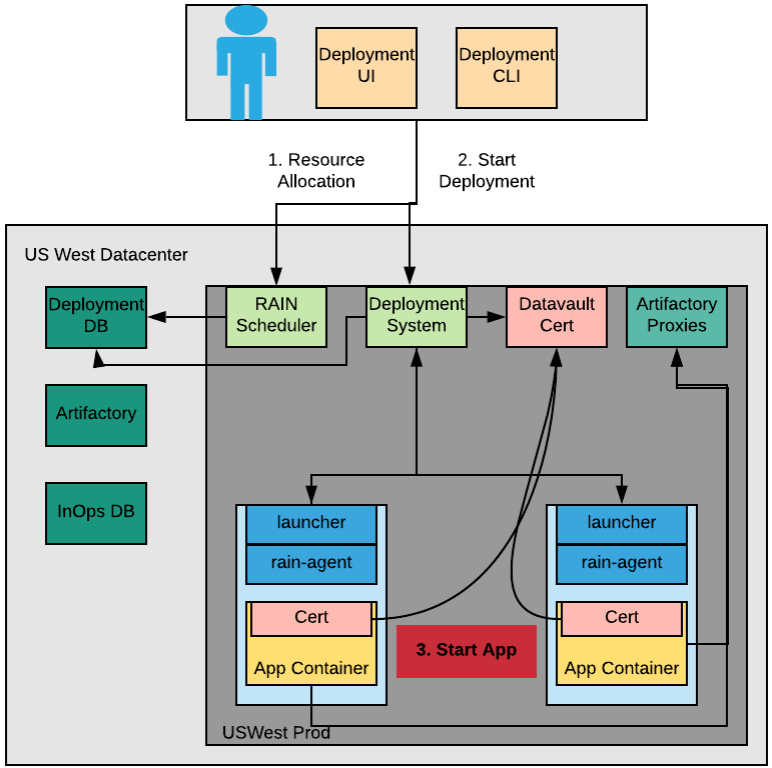
LID

- Generates the deployment plan and sends to the host launcher, which starts the application on physical host
- Enforces RBAC and other deployment policies



Locker

- LinkedIn's abstraction layer (over runc) for providing containerization on Linux hosts
- Resource and Namespaces Isolation



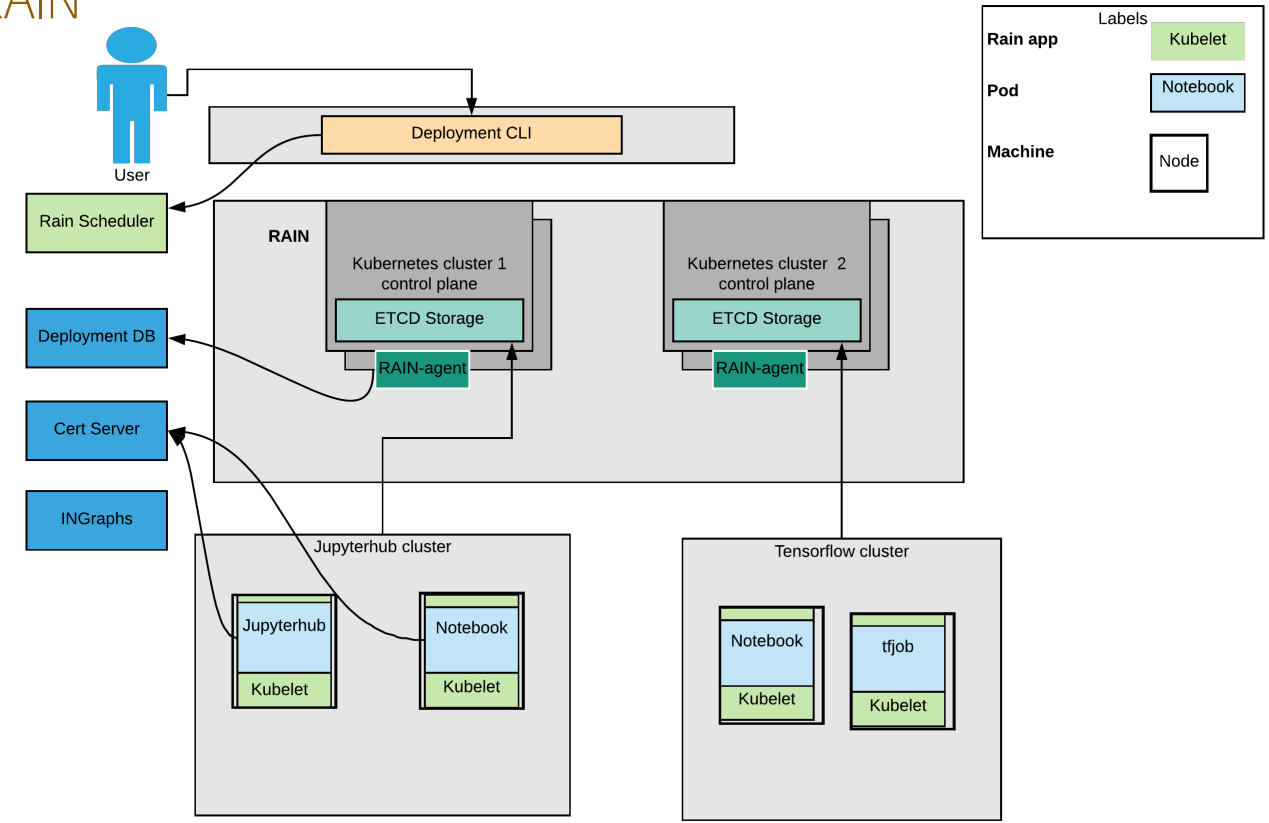
ORCA

- Create a tree of short running jobs with priorities
- Utilize spare capacity in common pool
- Collect results and exit status
- Replace Hudson's job scheduling with Orca

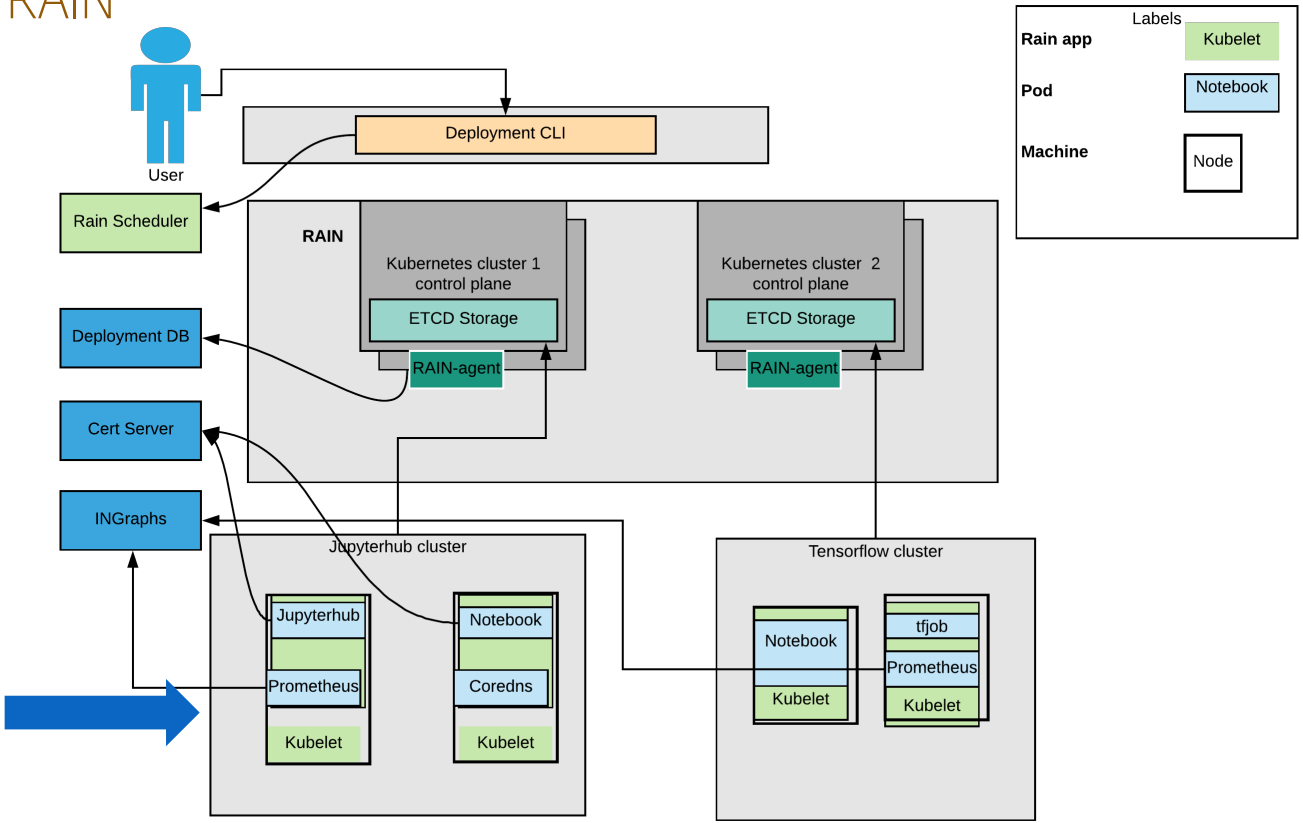
Challenges

- Capability to support growing batch workloads
- Extensibility to model emerging workloads
- Portability across on-premise and cloud

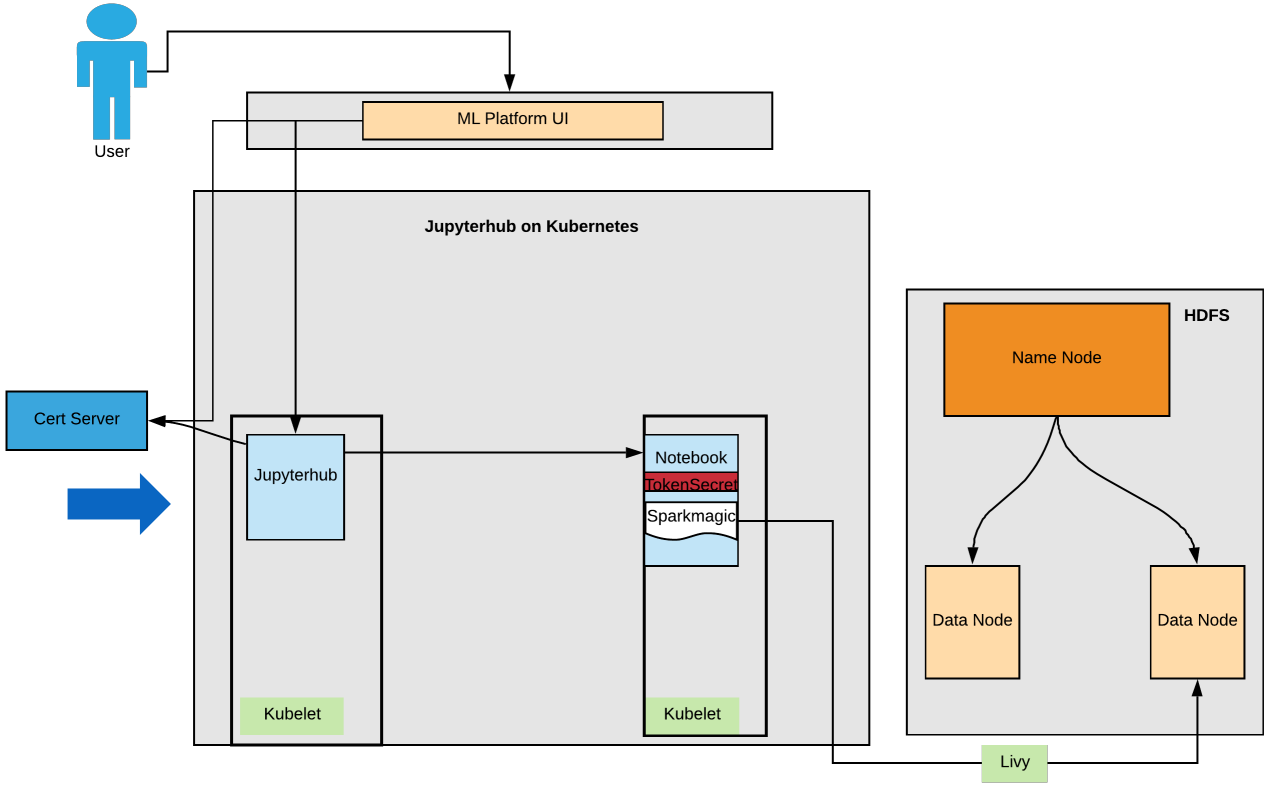
K8s on RAIN


Labels
Rain app
Kubelet
Pod
Notebook
Machine
Node
User
Deployment CLI
Rain Scheduler
RAIN
Kubernetes cluster 1
control plane
ETCD Storage
RAIN-agent
Kubernetes cluster 2
control plane
ETCD Storage
RAIN-agent
Deployment DB
Cert Server
INGraphs
Jupyterhub cluster
Jupyterhub
Kubelet
Notebook
Kubelet
Tensorflow cluster
Notebook
Kubelet
tfjob
Kubelet

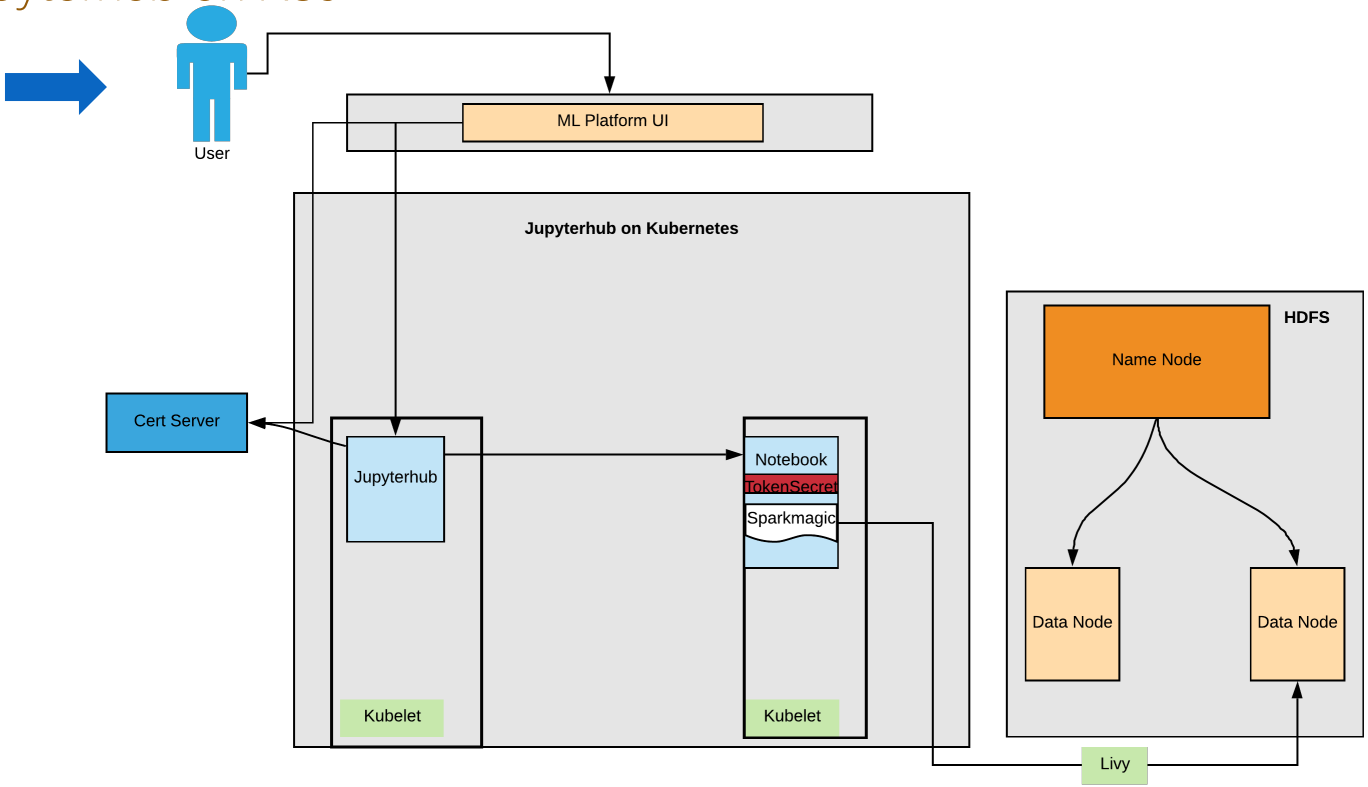
K8s on RAIN


Labels
Rain app
Kubelet
Pod
Notebook
Machine
Node
User
Deployment CLI
Rain Scheduler
RAIN
Kubernetes cluster 1
control plane
ETCD Storage
RAIN-agent
Kubernetes cluster 2
control plane
ETCD Storage
RAIN-agent
Deployment DB
Cert Server
INGraphs
Jupyterhub cluster
Jupyterhub
Prometheus
Kubelet
Notebook
Coredns
Kubelet
Tensorflow cluster
Notebook
Kubelet
tfjob
Prometheus
Kubelet

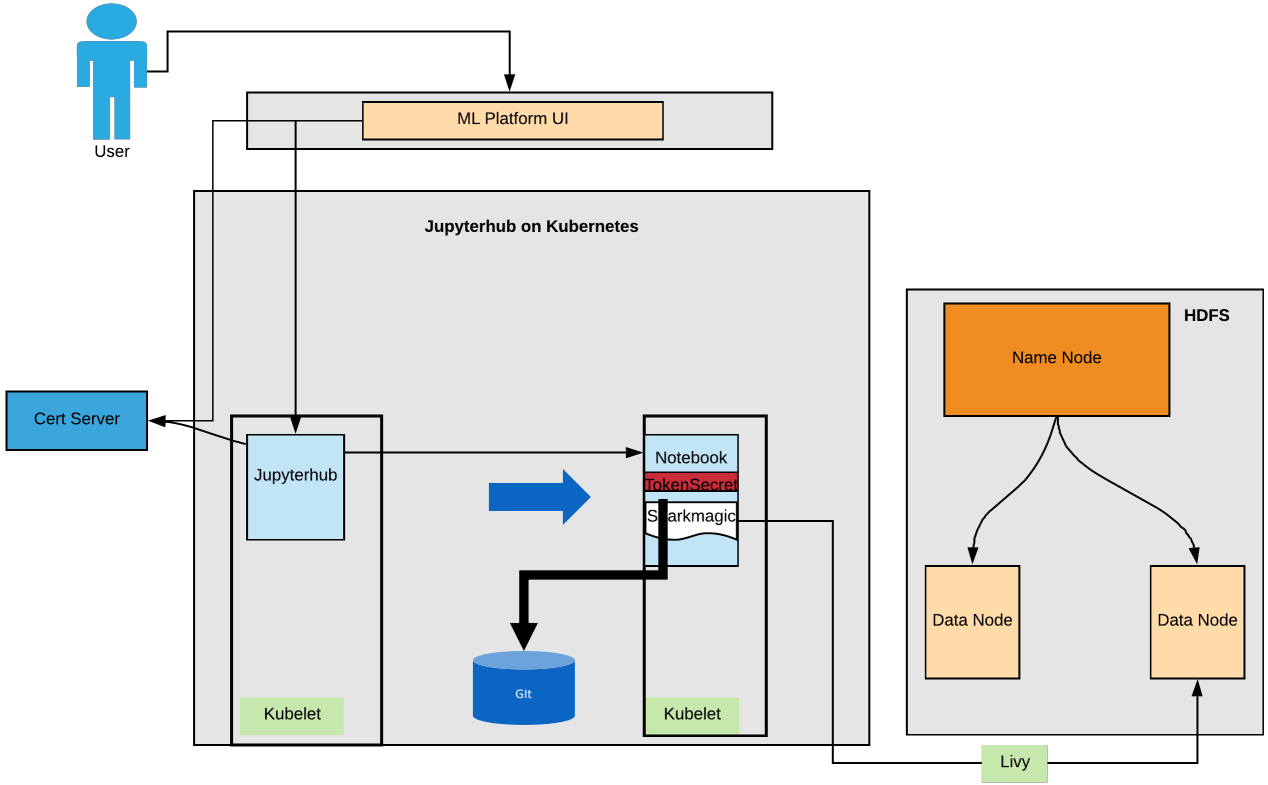
Jupyterhub on K8s



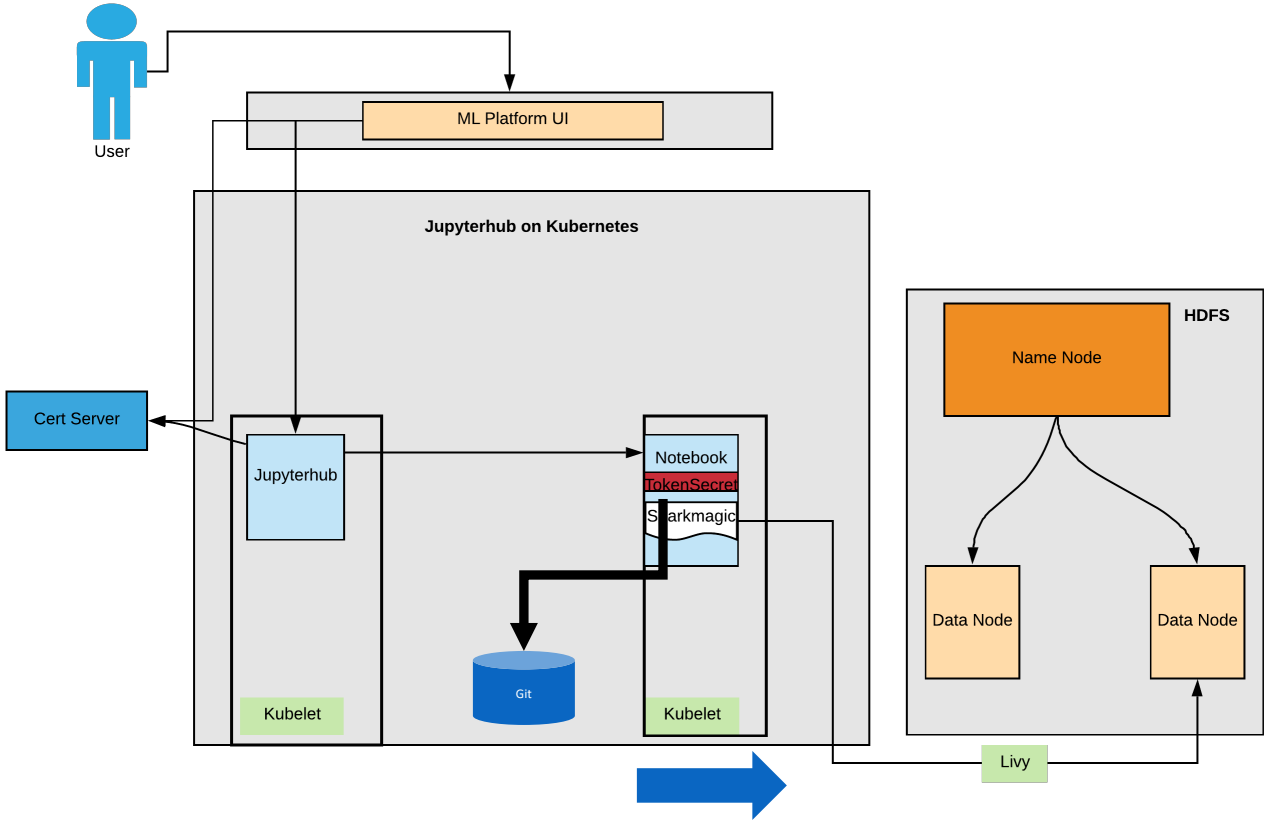
Jupyterhub on K8s



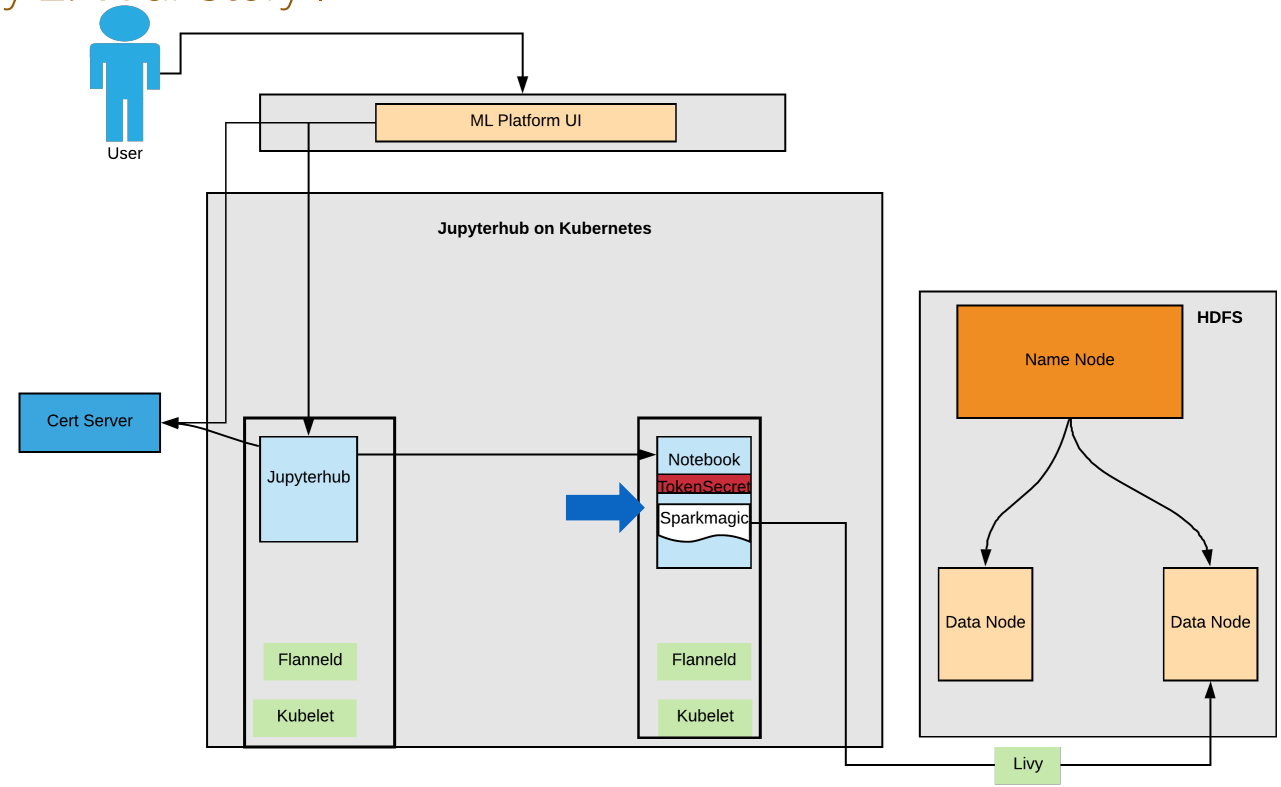
Jupyterhub on K8s



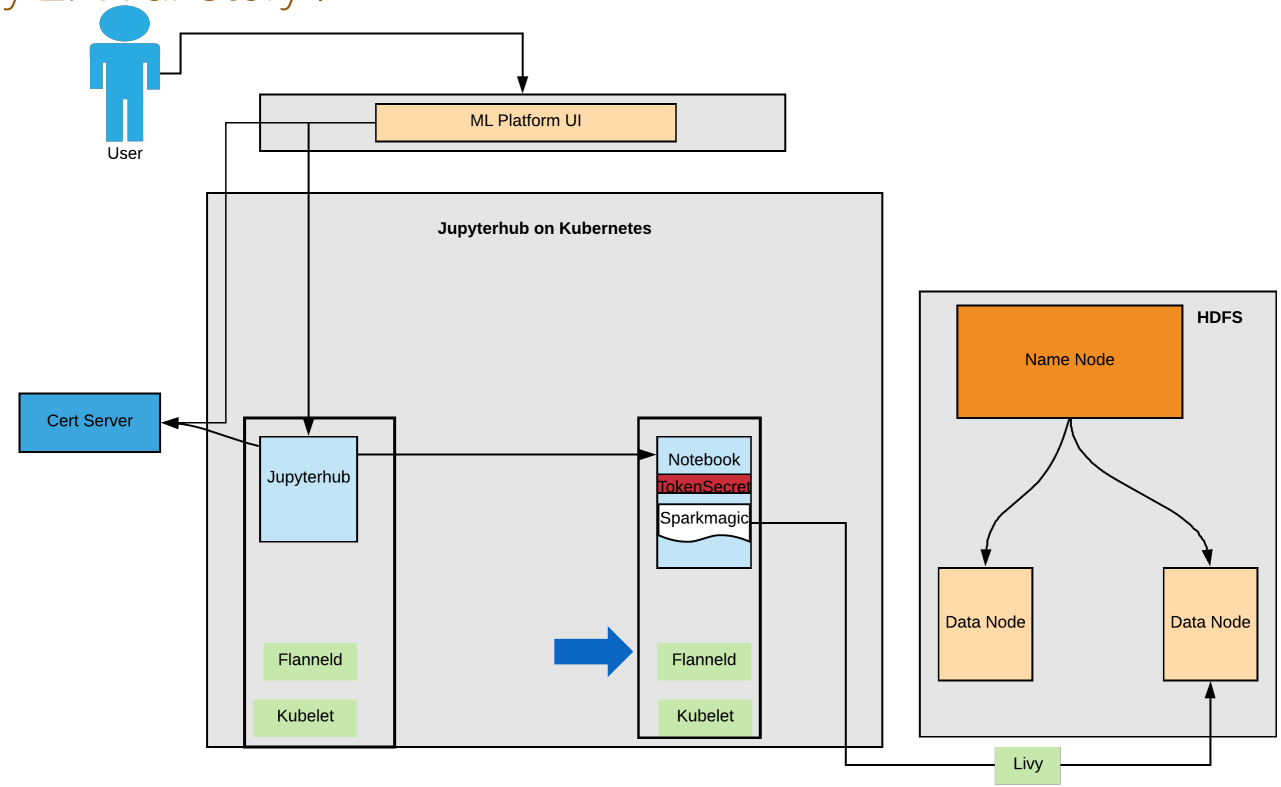
Jupyterhub on K8s



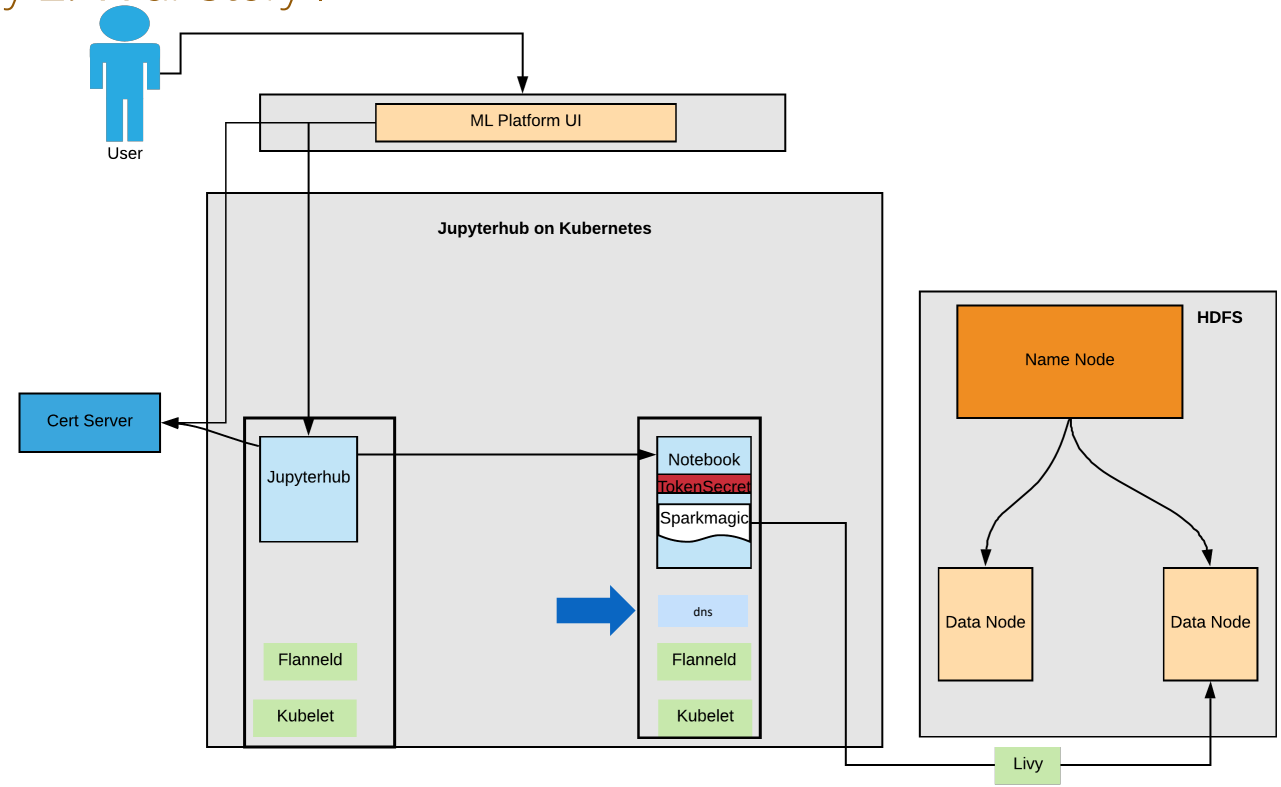
K8s Day 2: War Story I



K8s Day 2: War Story I



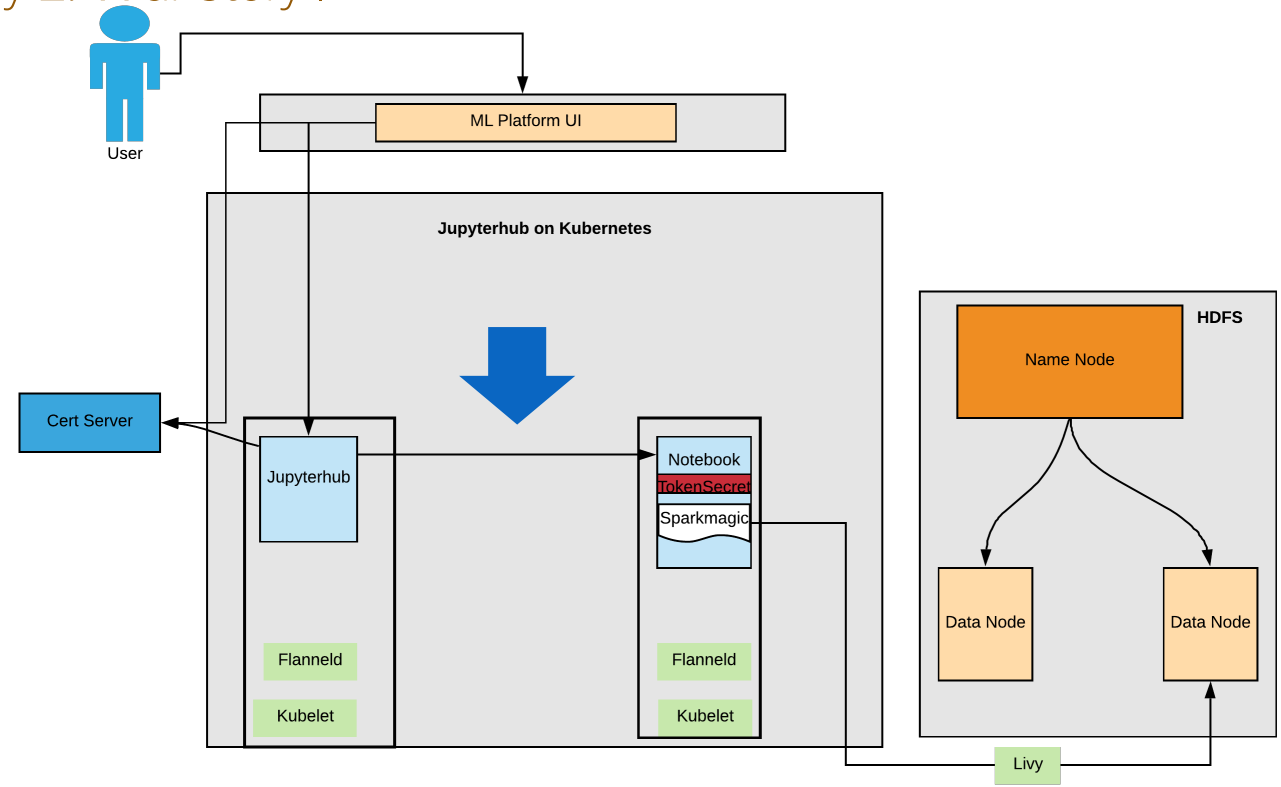
K8s Day 2: War Story I

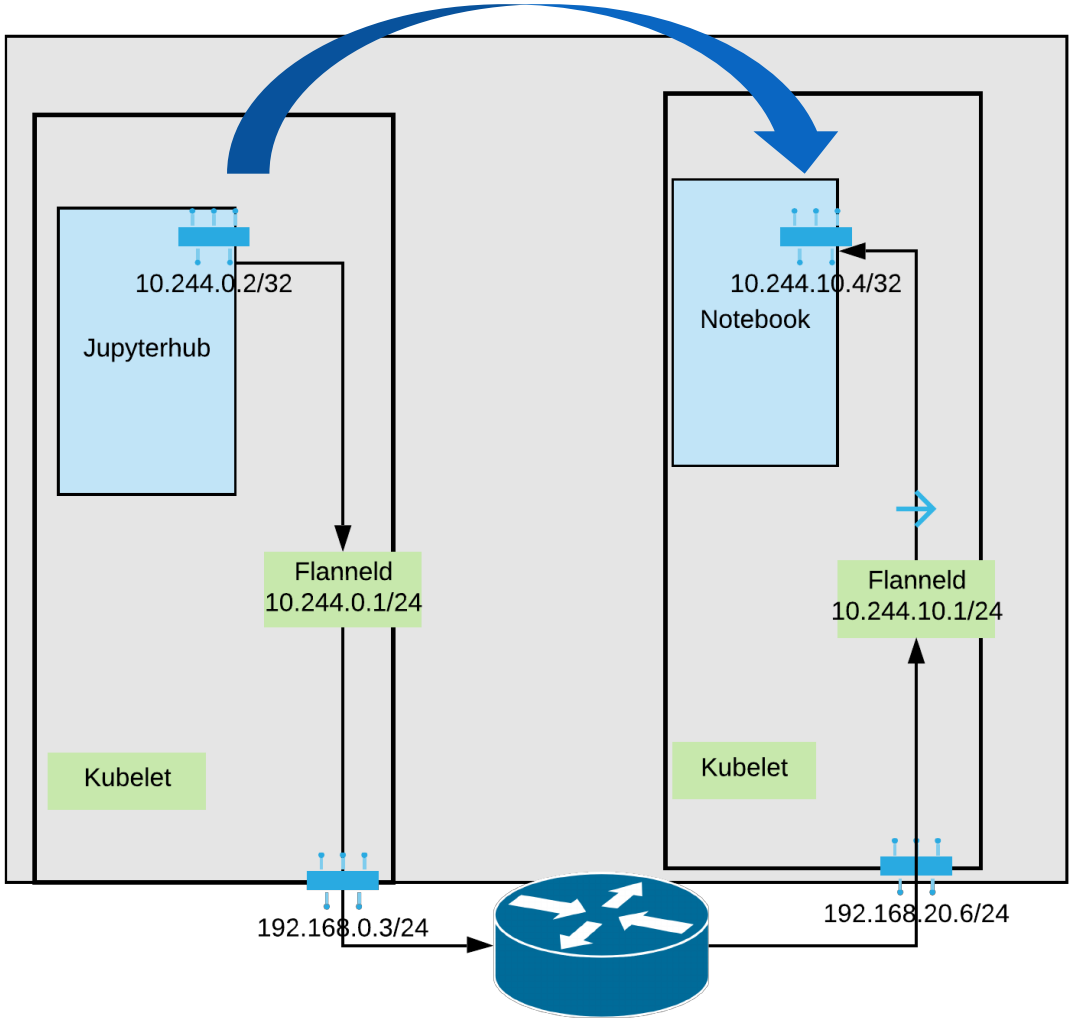


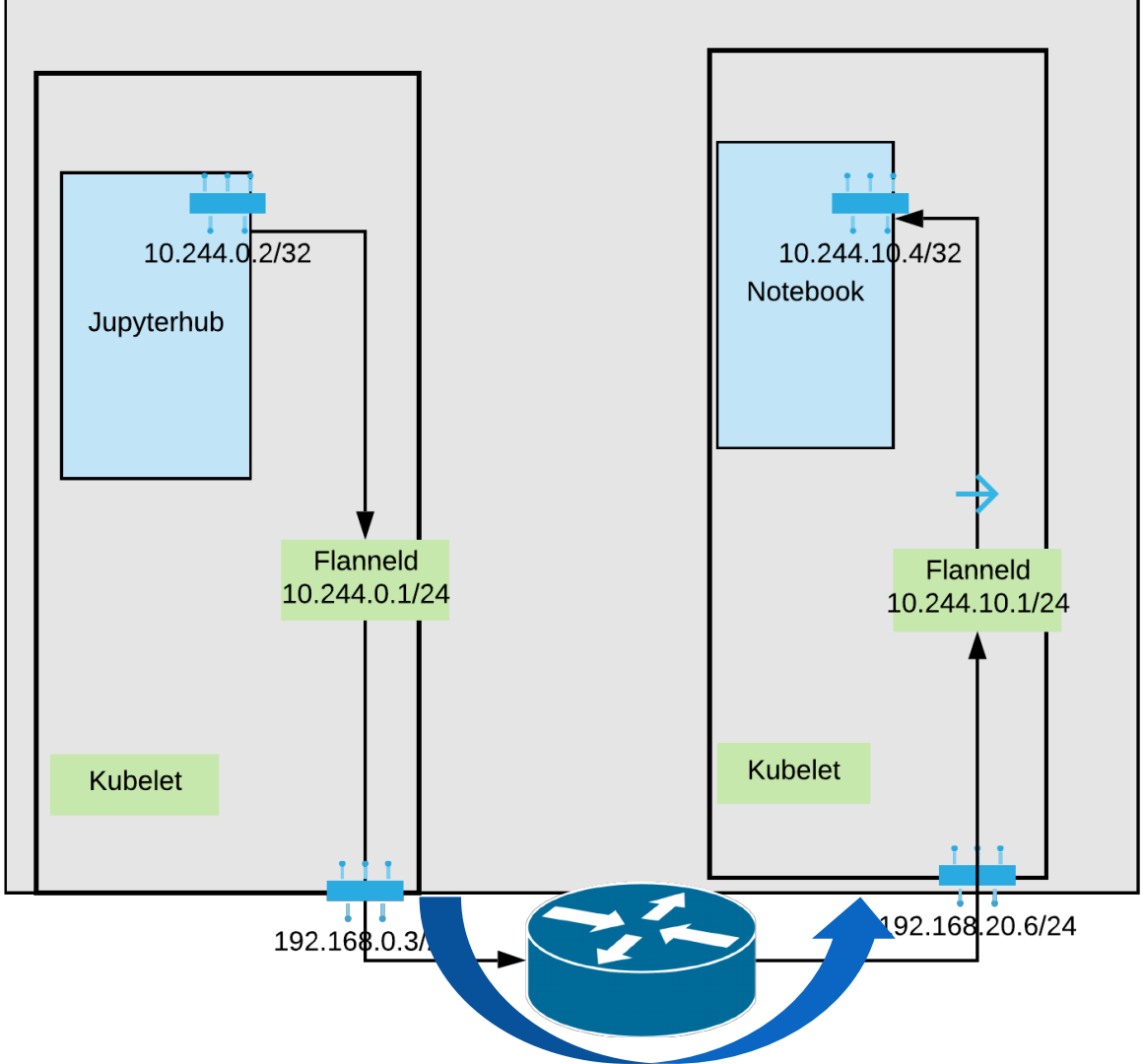
K8s Day 2: War Story I

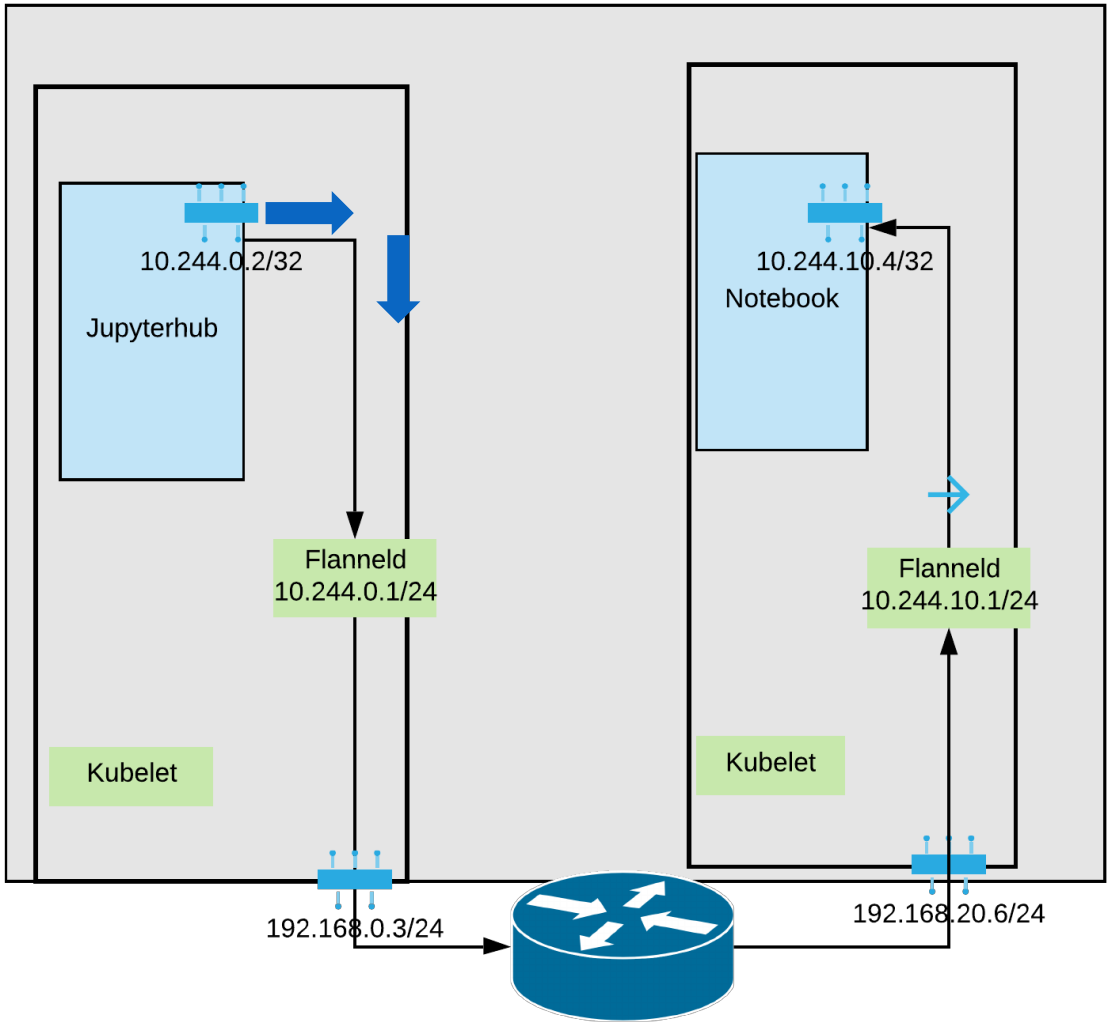
"Why are you hesitant about nuking the cluster and creating a new one? it'll take much less time and the odds of having a working cluster are higher"

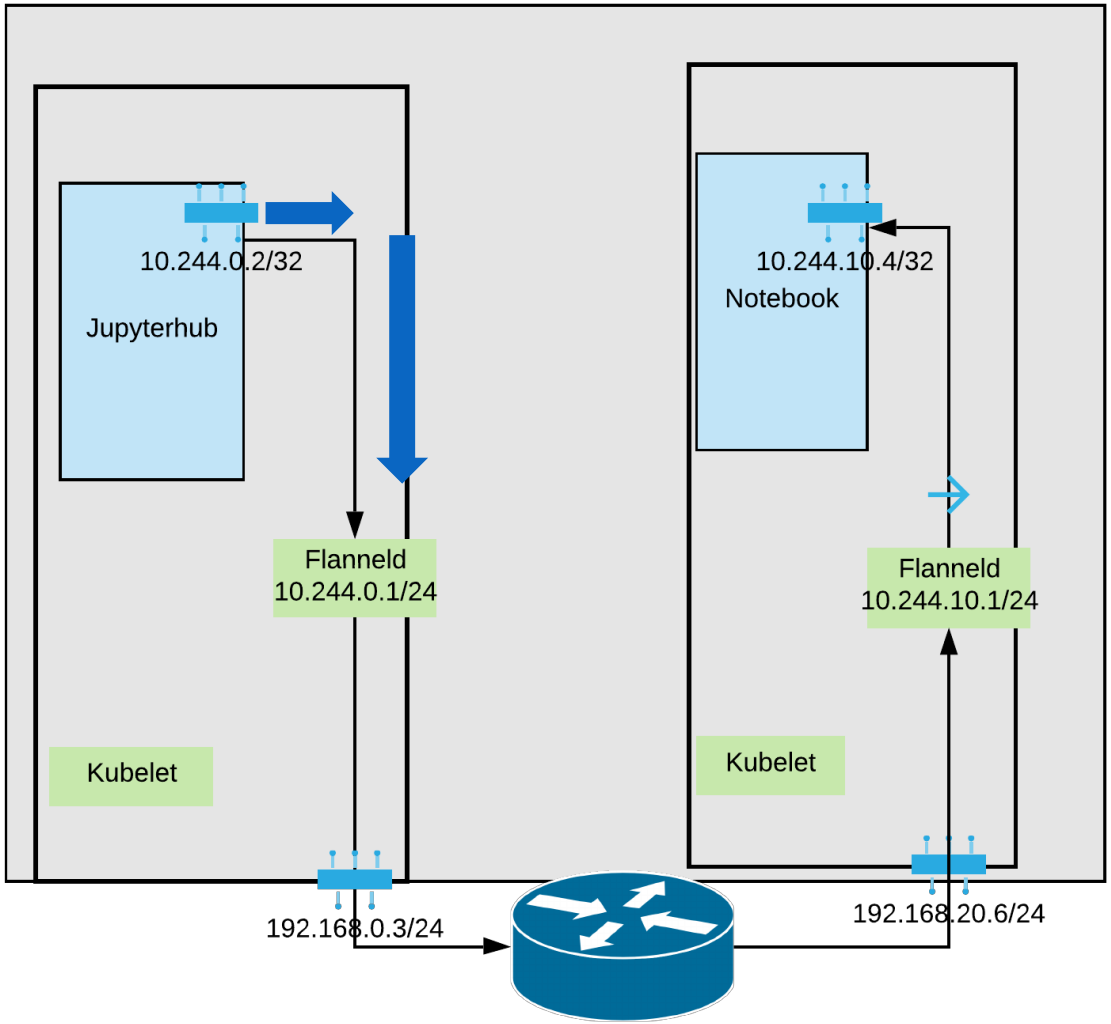
K8s Day 2: War Story I

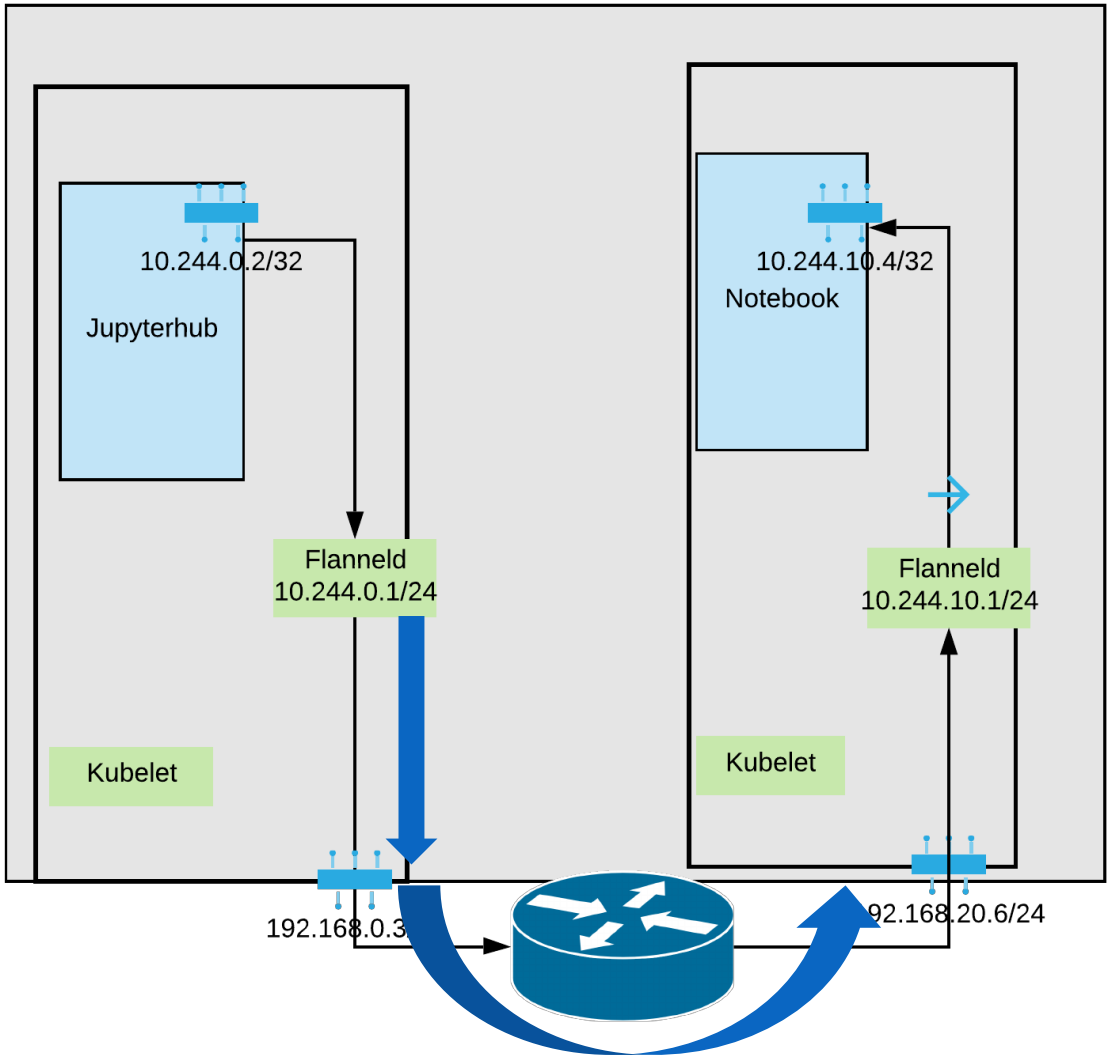


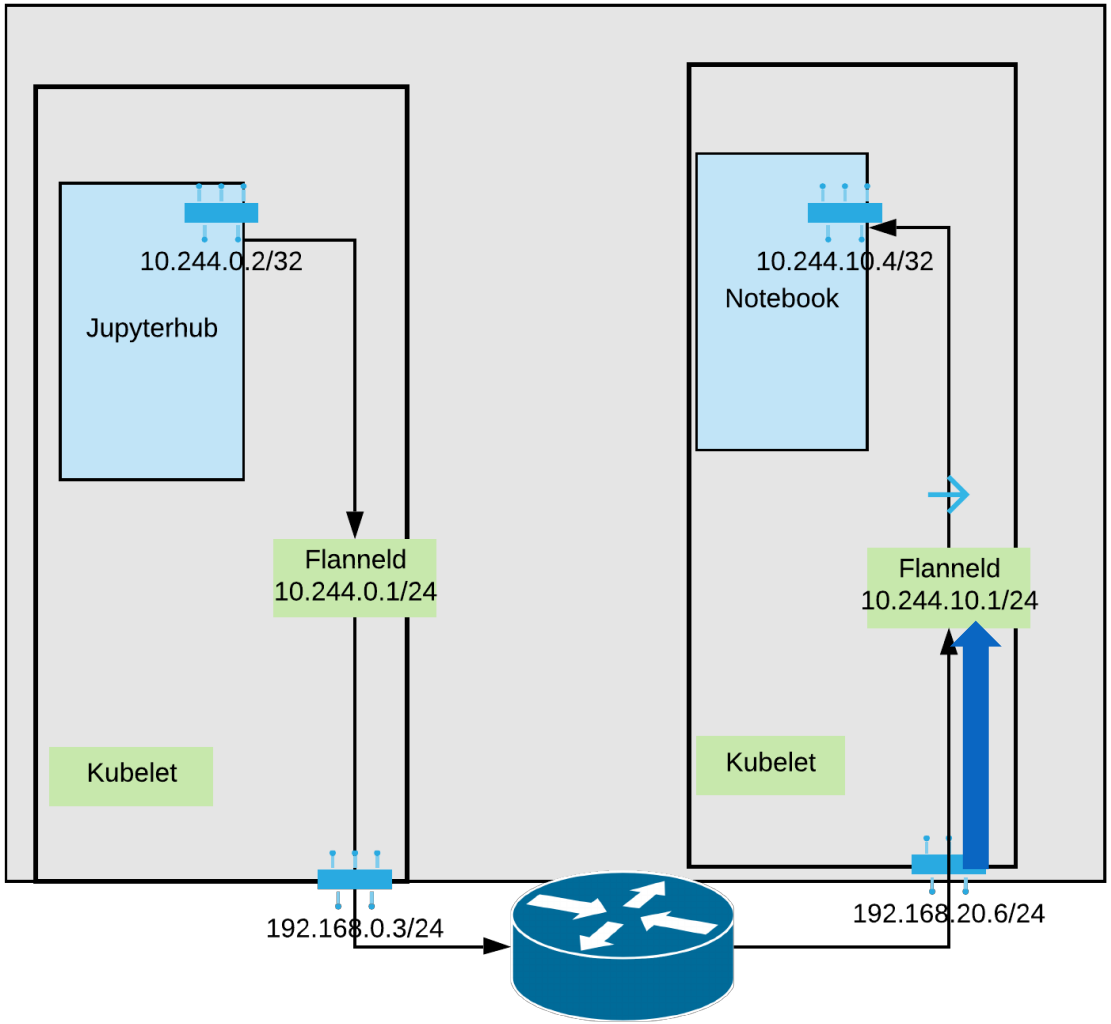

10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
192.168.0.3/24
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.20.6/24

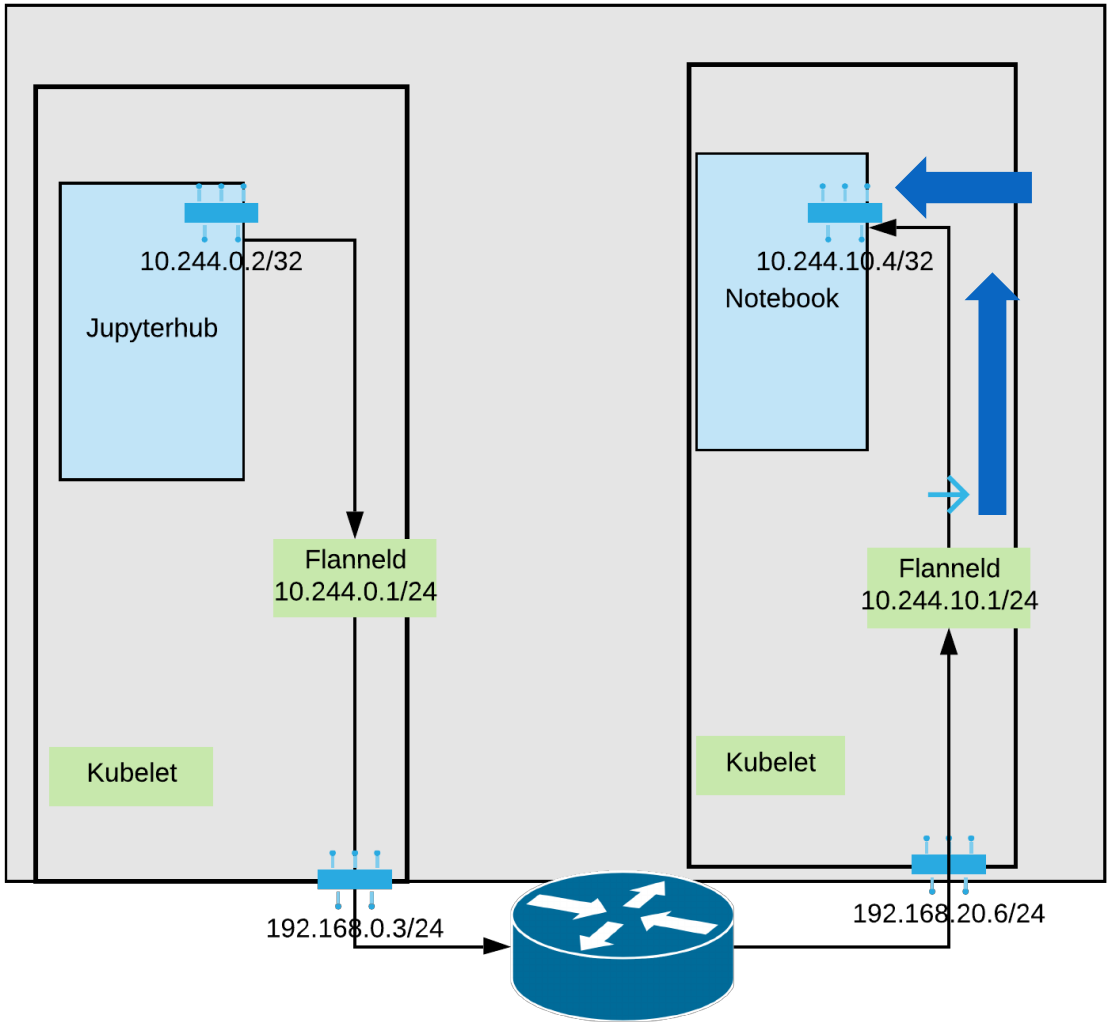

10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.20.6/24

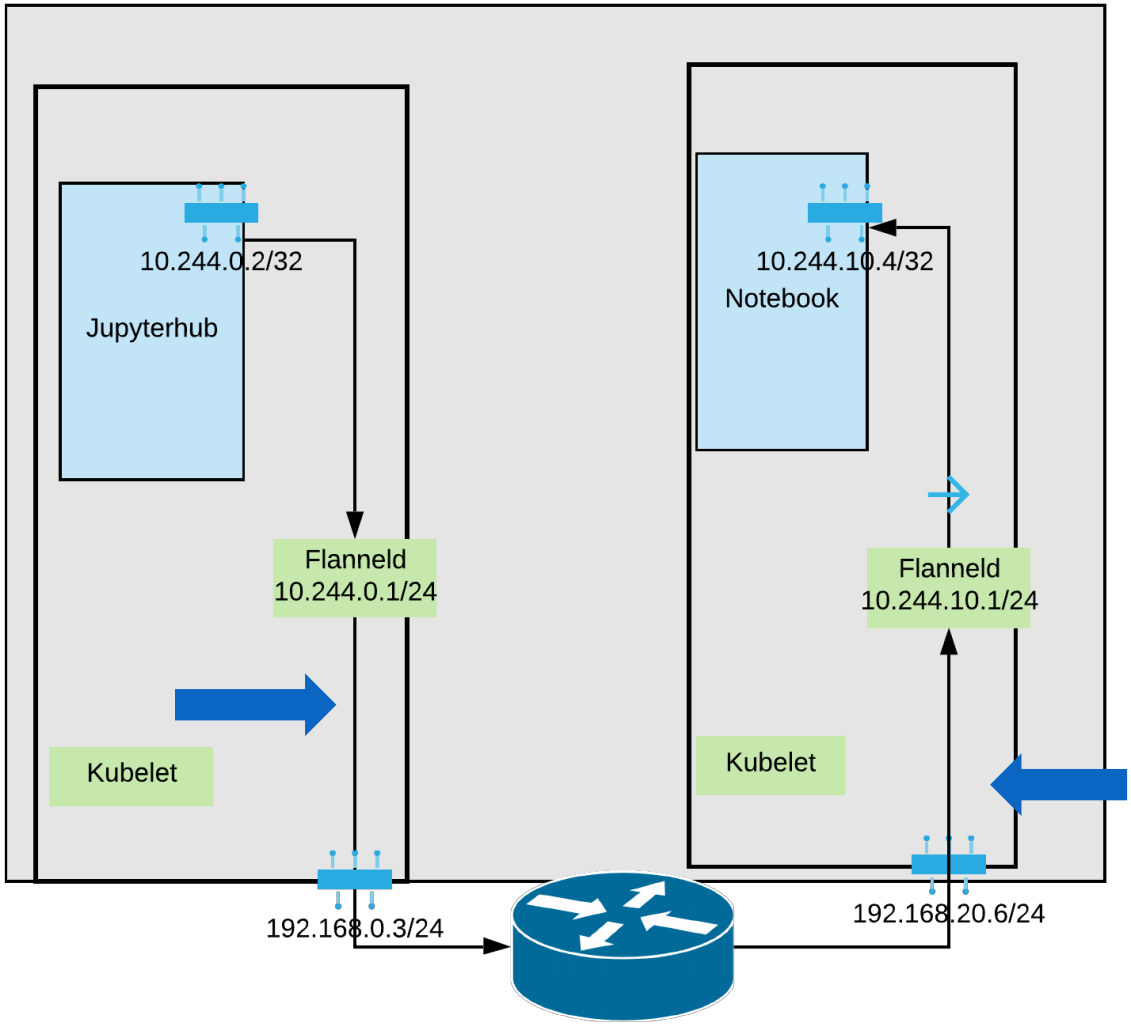

10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.0.3/24
192.168.20.6/24

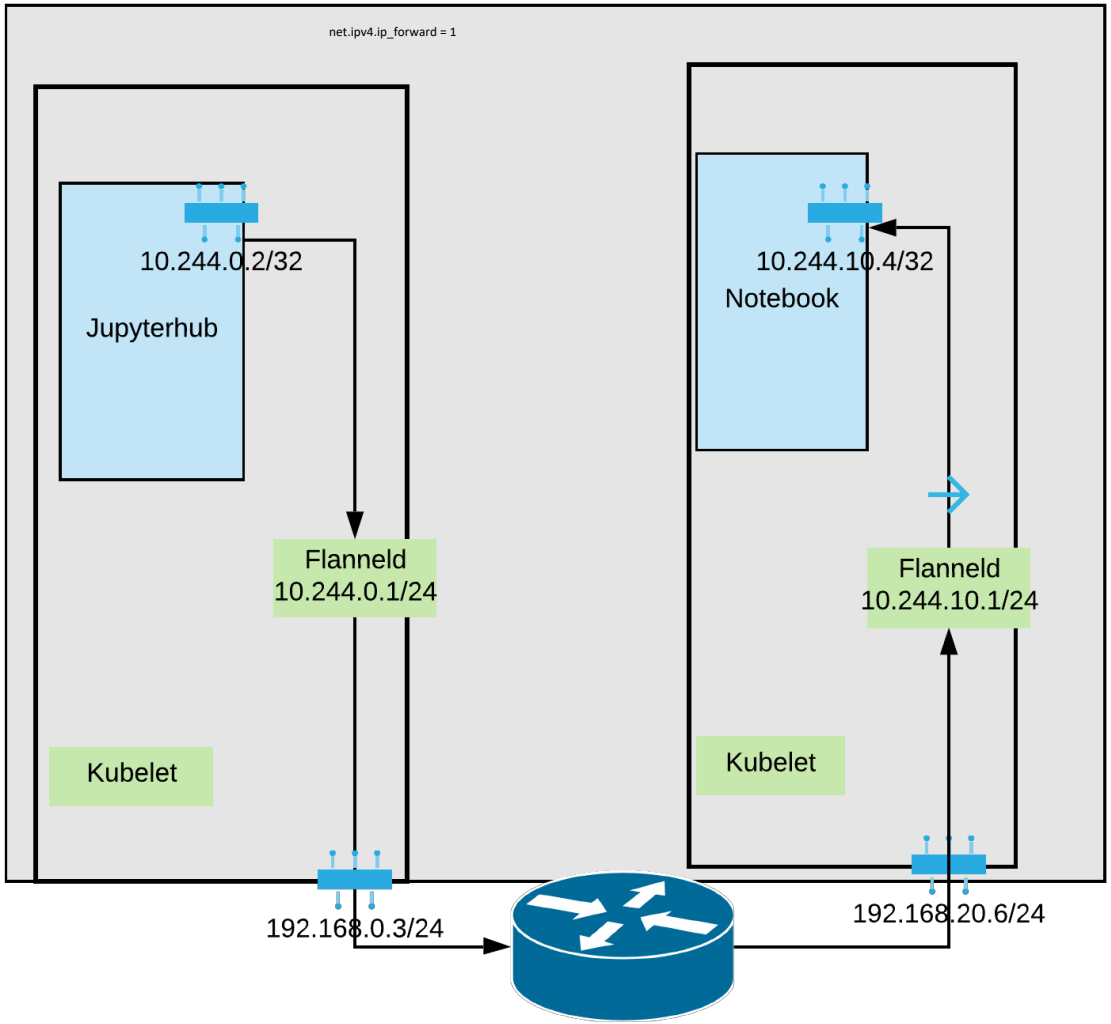

10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.0.3/24
192.168.20.6/24

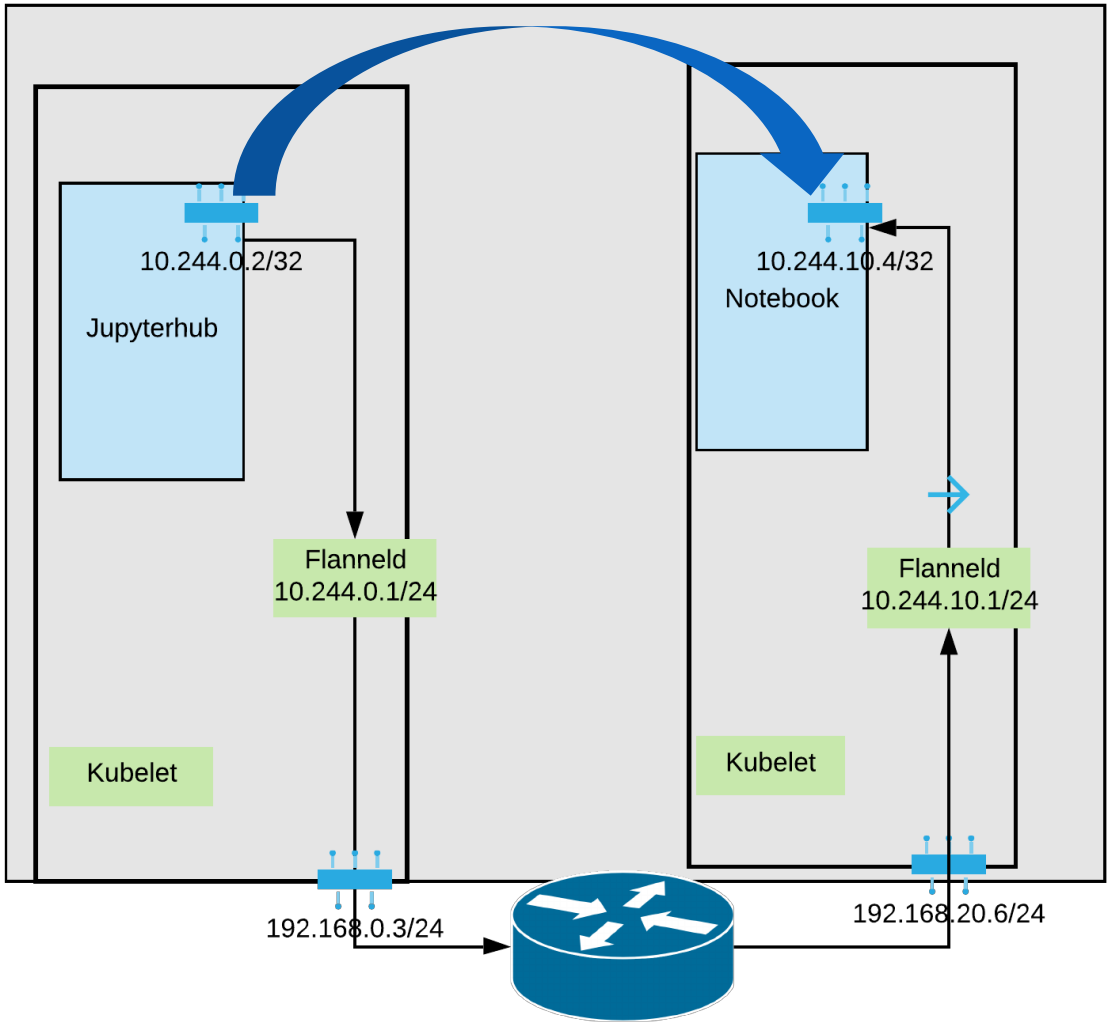

10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.0.3
92.168.20.6/24


10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
192.168.0.3/24
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.20.6/24

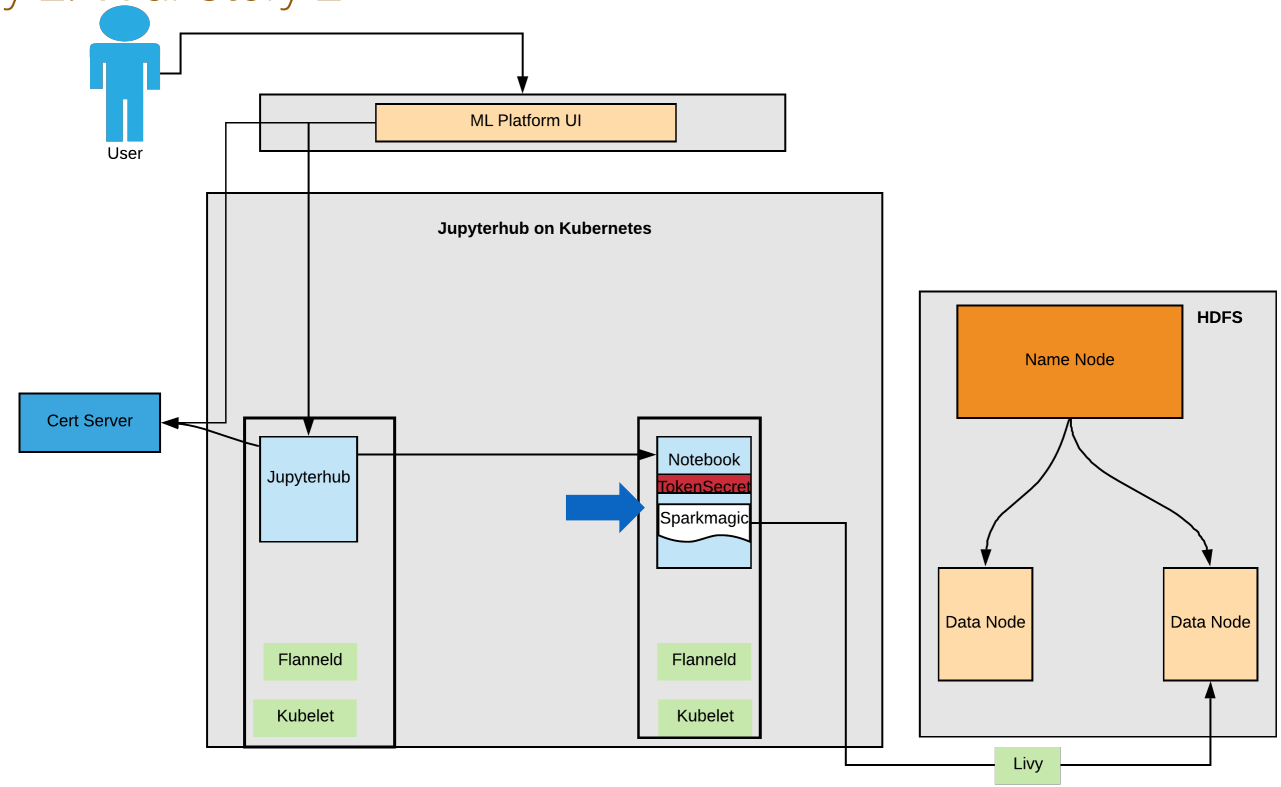

10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
192.168.0.3/24
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.20.6/24


10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
192.168.0.3/24
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.20.6/24

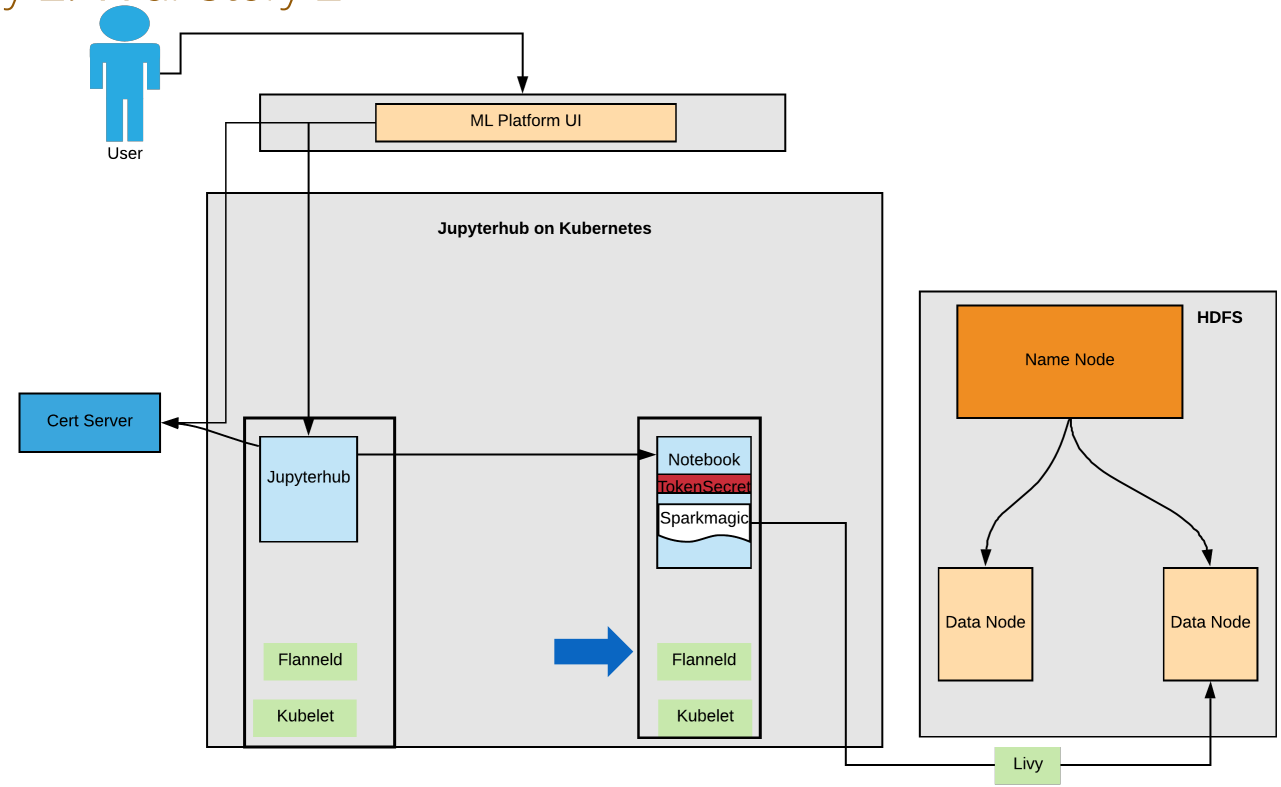

net.ipv4.ip_forward = 1
10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.0.3/24
192.168.20.6/24


10.244.0.2/32
Jupyterhub
Flanneld
10.244.0.1/24
Kubelet
192.168.0.3/24
10.244.10.4/32
Notebook
Flanneld
10.244.10.1/24
Kubelet
192.168.20.6/24

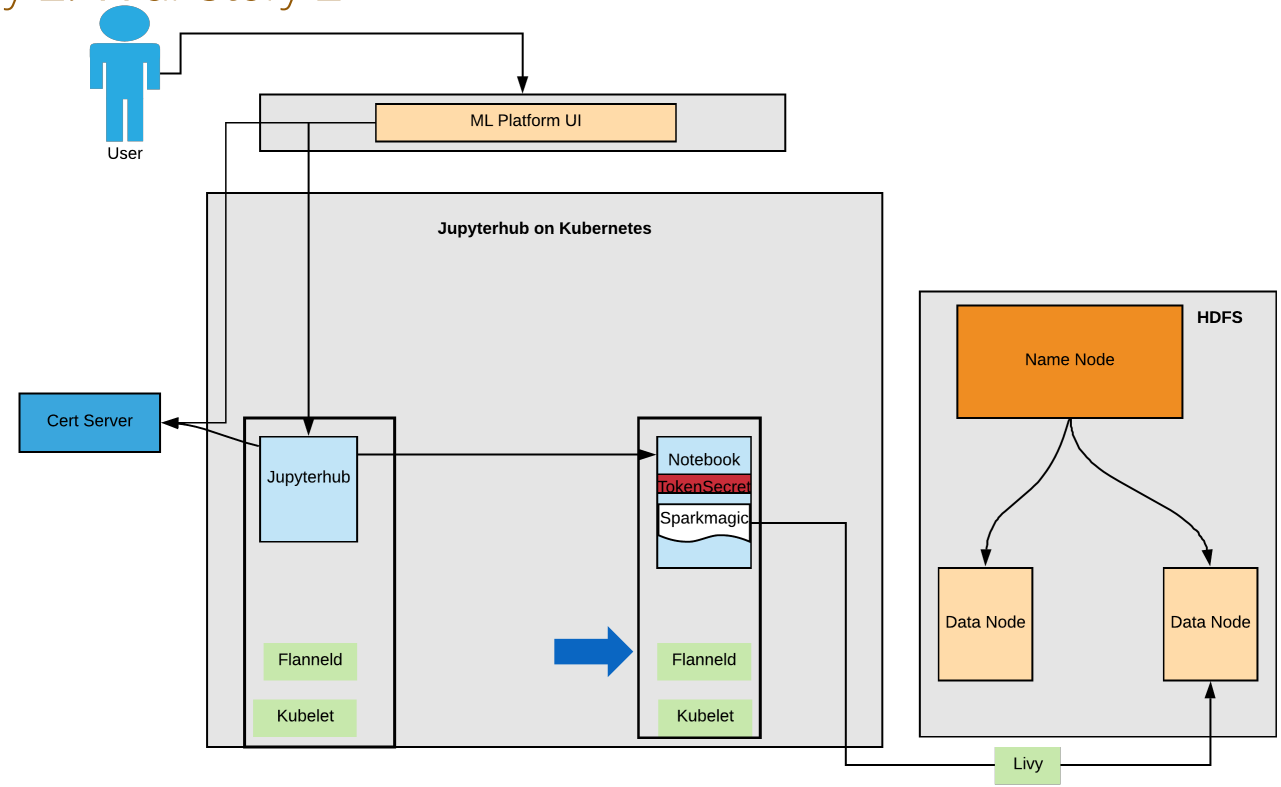
K8s Day 2: War Story 2



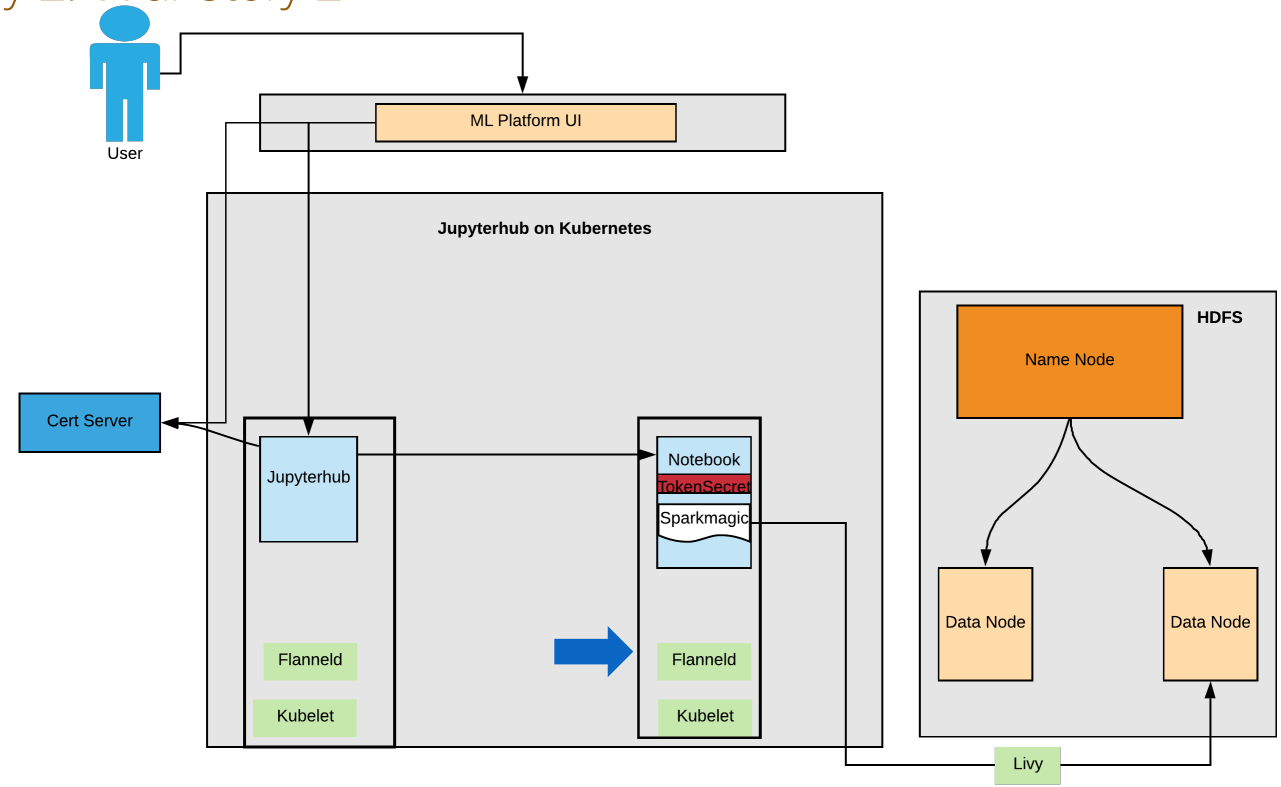
K8s Day 2: War Story 2



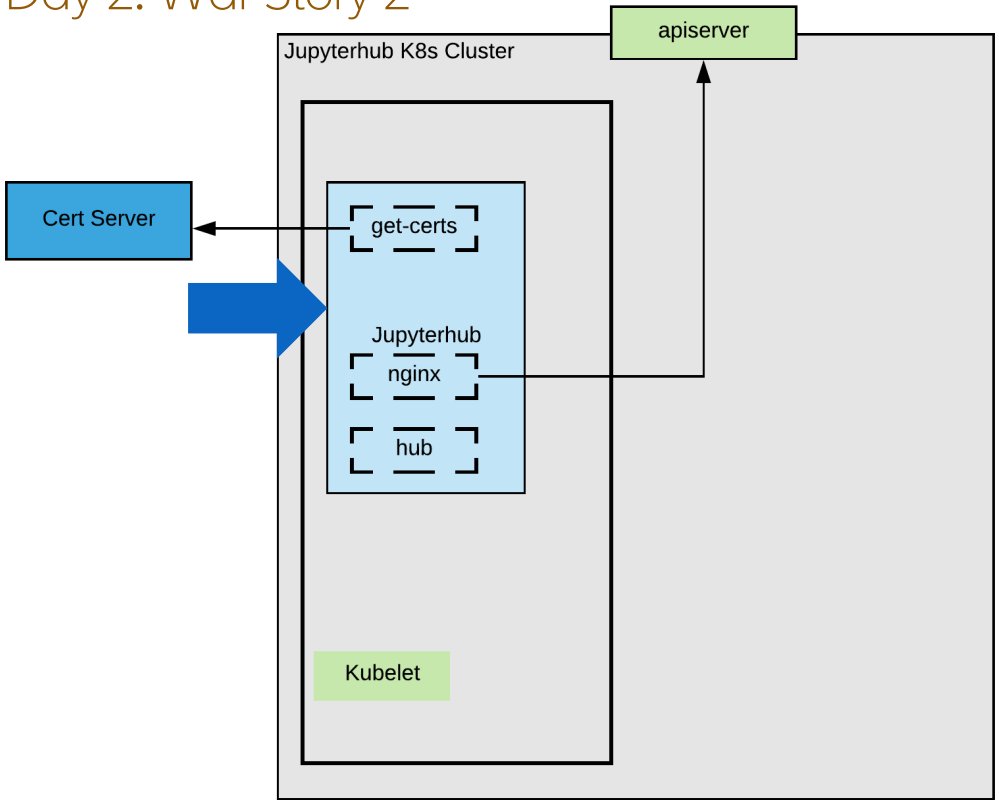
K8s Day 2: War Story 2



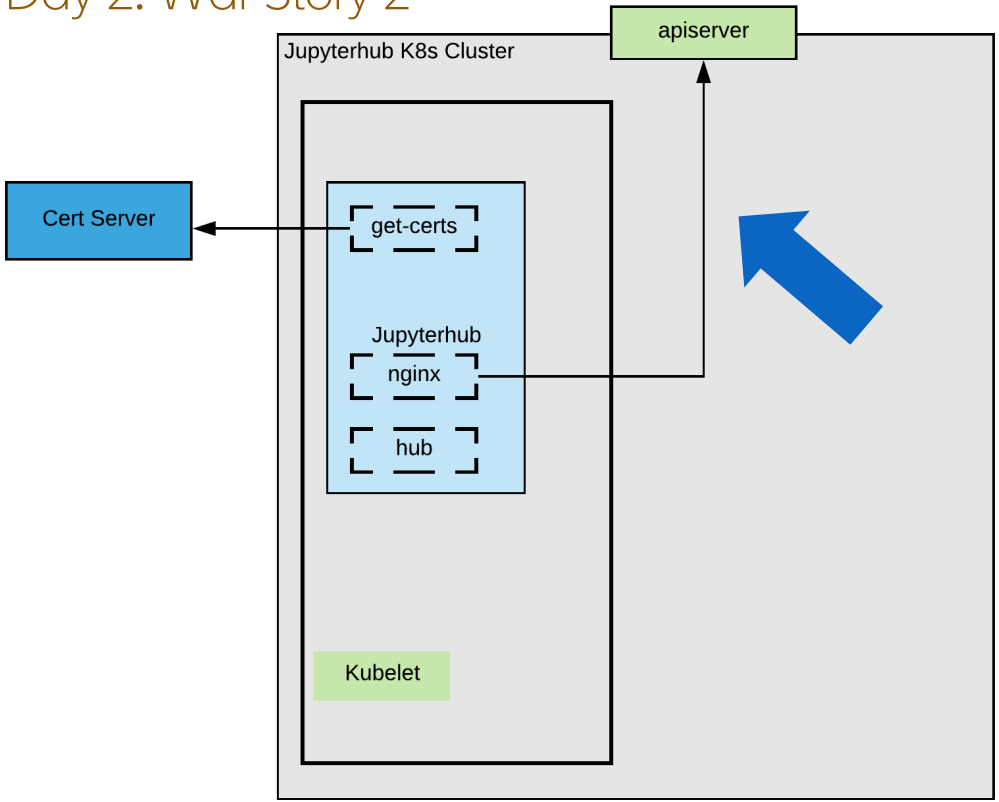
K8s Day 2: War Story 2

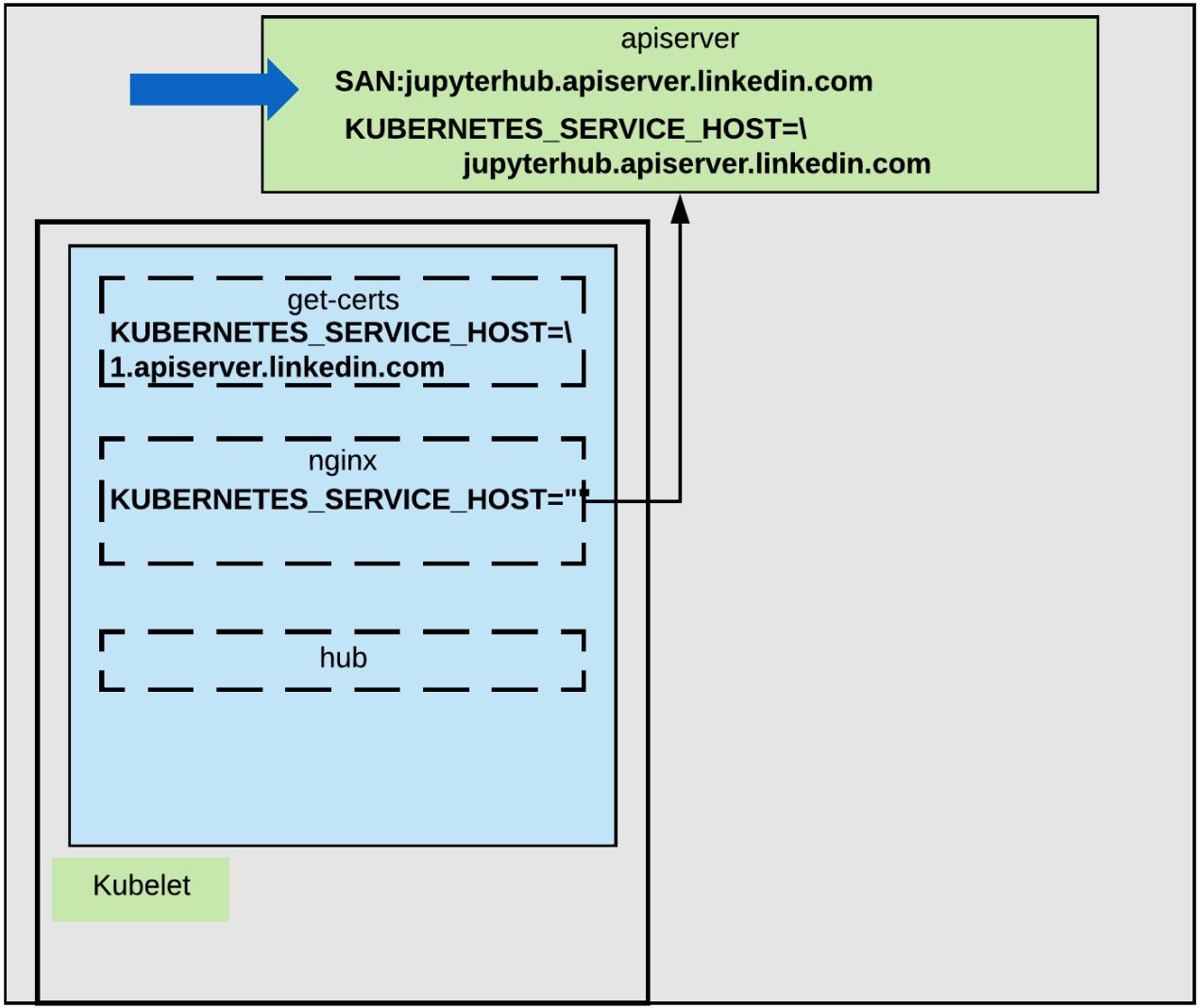


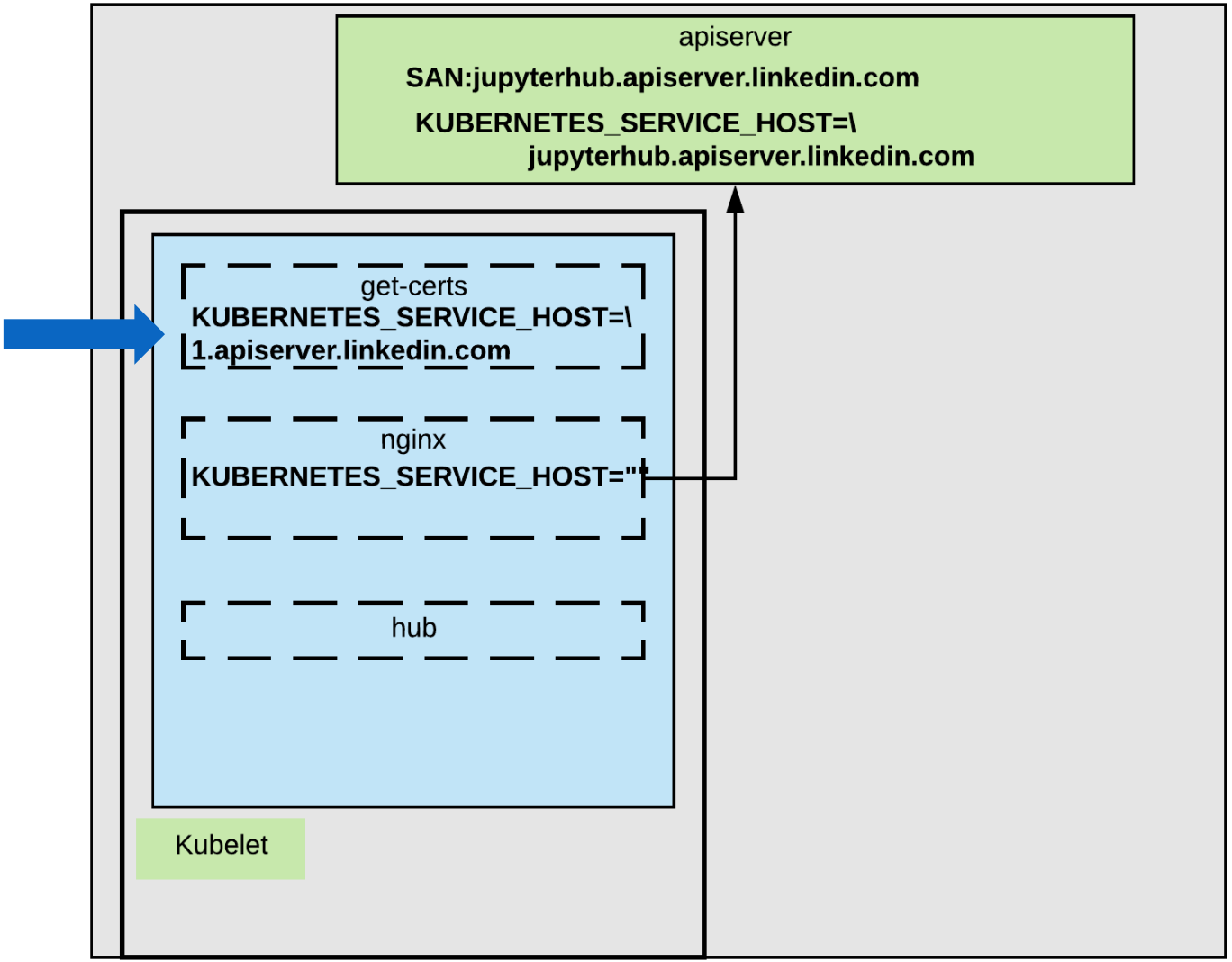
K8s Day 2: War Story 2

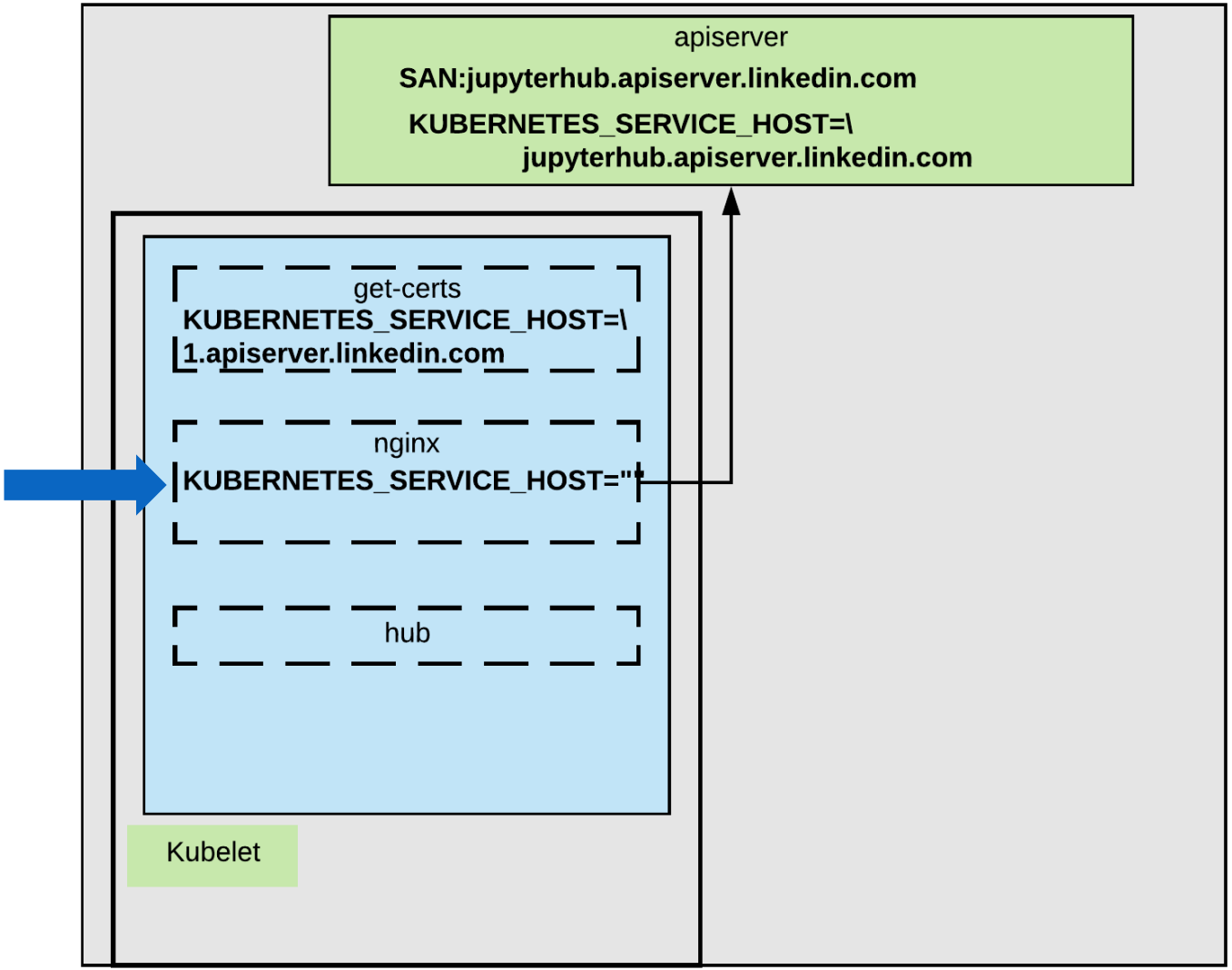


K8s Day 2: War Story 2

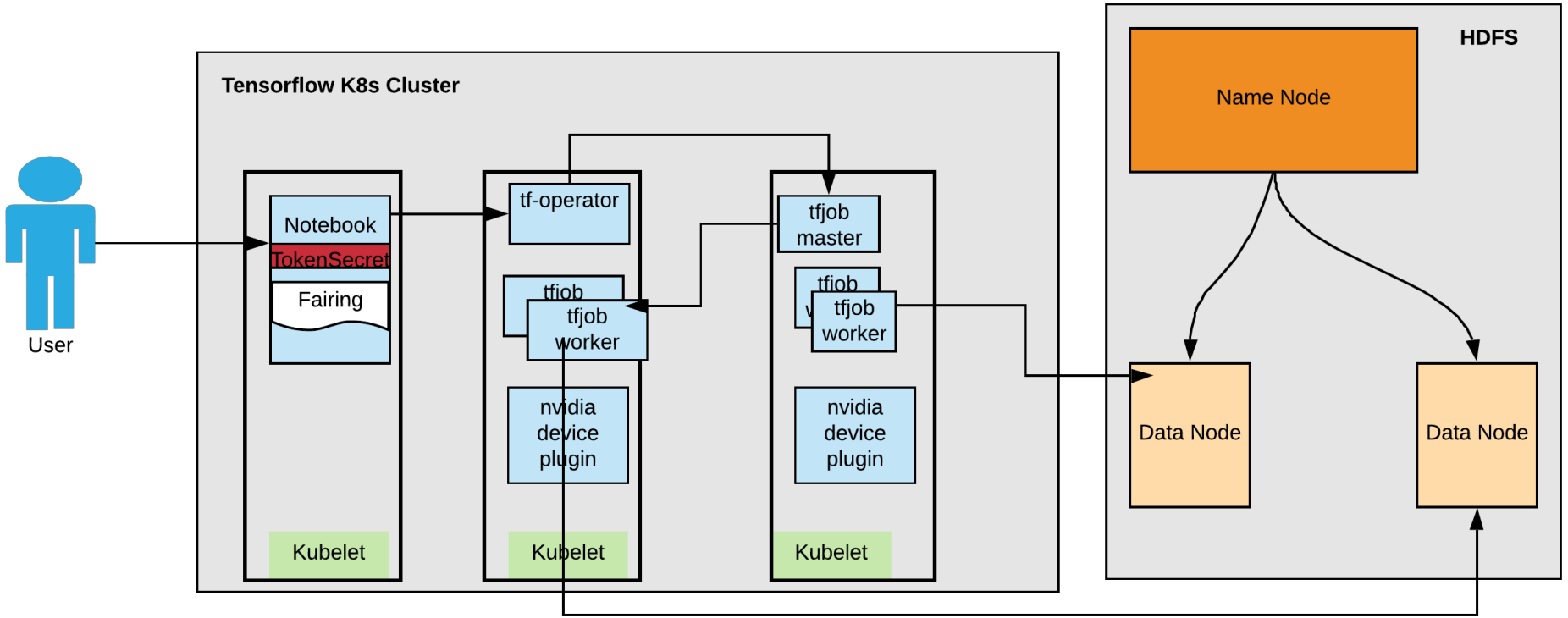



apiserver
SAN:jupyterhub.apiserver.linkedin.com
KUBERNETES_SERVICE_HOST=\
jupyterhub.apiserver.linkedin.com
get-certs
KUBERNETES_SERVICE_HOST=\
1.apiserver.linkedin.com
nginx
KUBERNETES_SERVICE_HOST="
hub
Kubelet

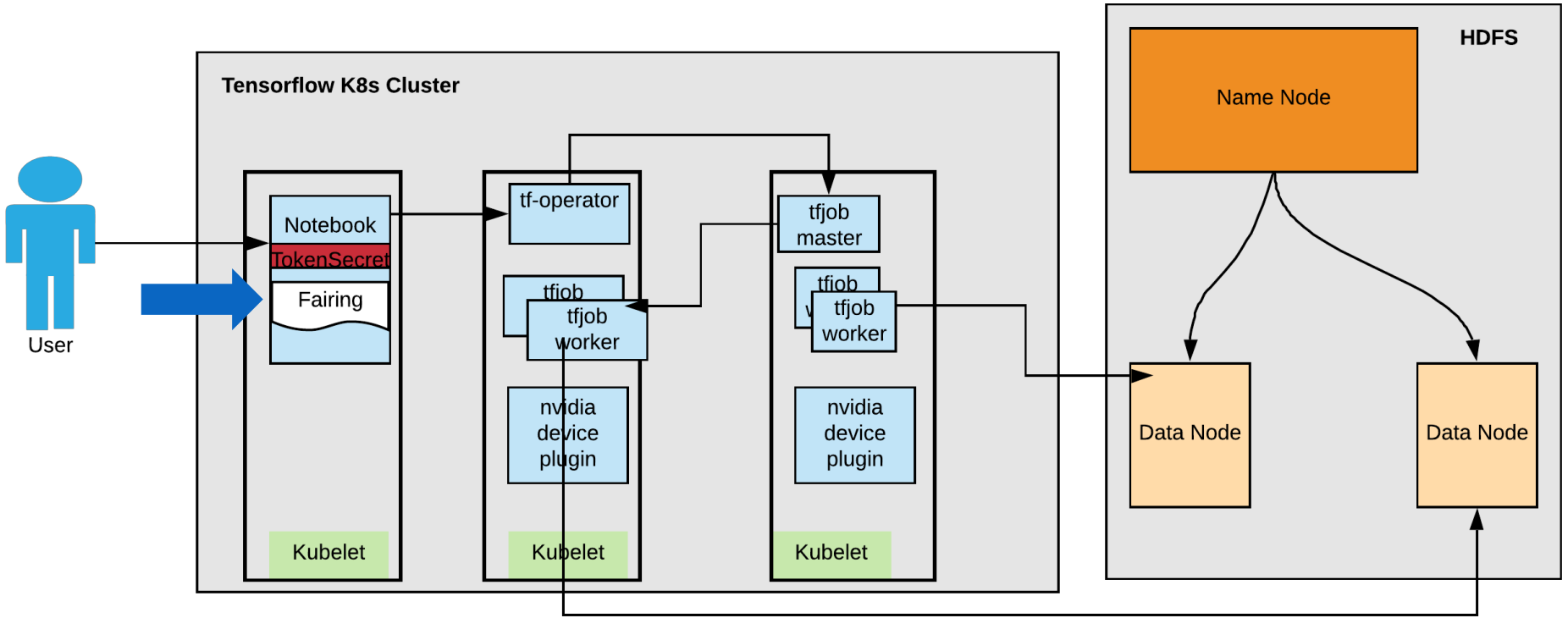

apiserver
SAN:jupyterhub.apiserver.linkedin.com
KUBERNETES_SERVICE_HOST=\
jupyterhub.apiserver.linkedin.com
get-certs
KUBERNETES_SERVICE_HOST=\
1.apiserver.linkedin.com
nginx
KUBERNETES_SERVICE_HOST="
hub
Kubelet


apiserver
SAN:jupyterhub.apiserver.linkedin.com
KUBERNETES_SERVICE_HOST=\
jupyterhub.apiserver.linkedin.com
get-certs
KUBERNETES_SERVICE_HOST=\
1.apiserver.linkedin.com
nginx
KUBERNETES_SERVICE_HOST="
hub
Kubelet

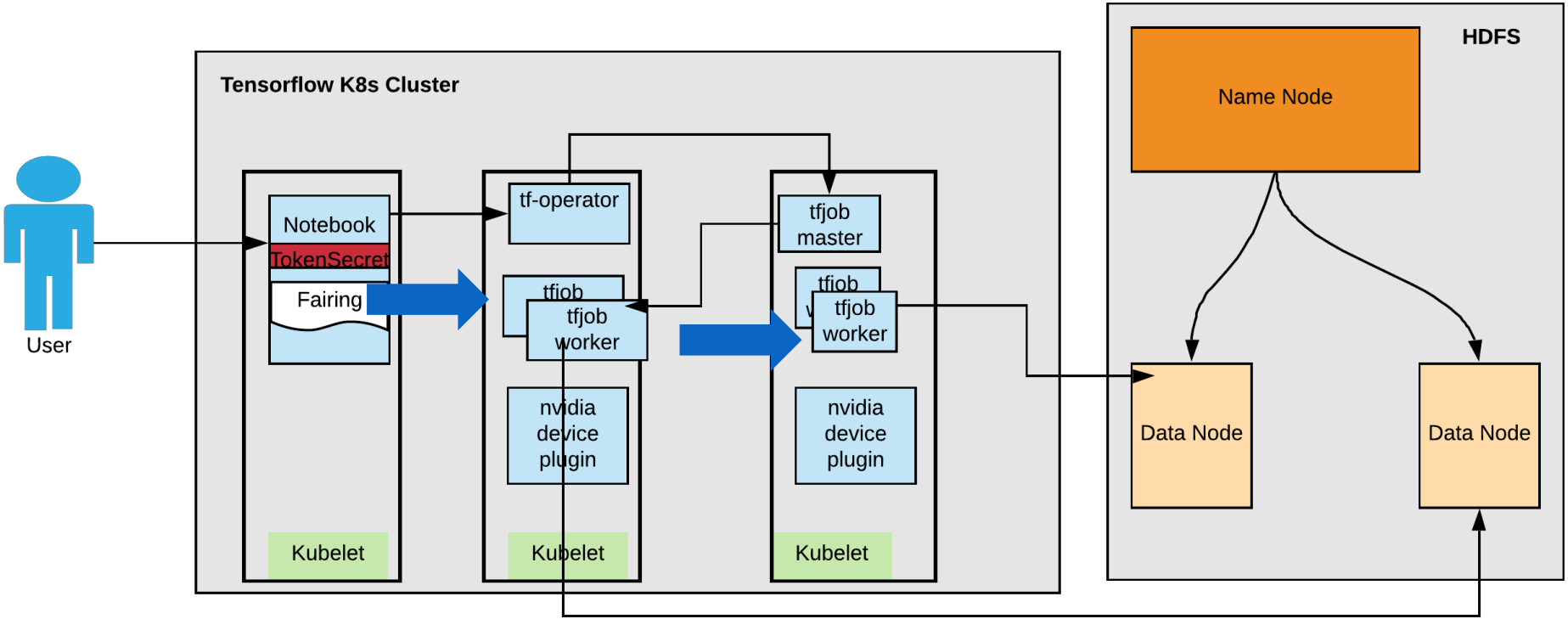
Kubeflow ML Training



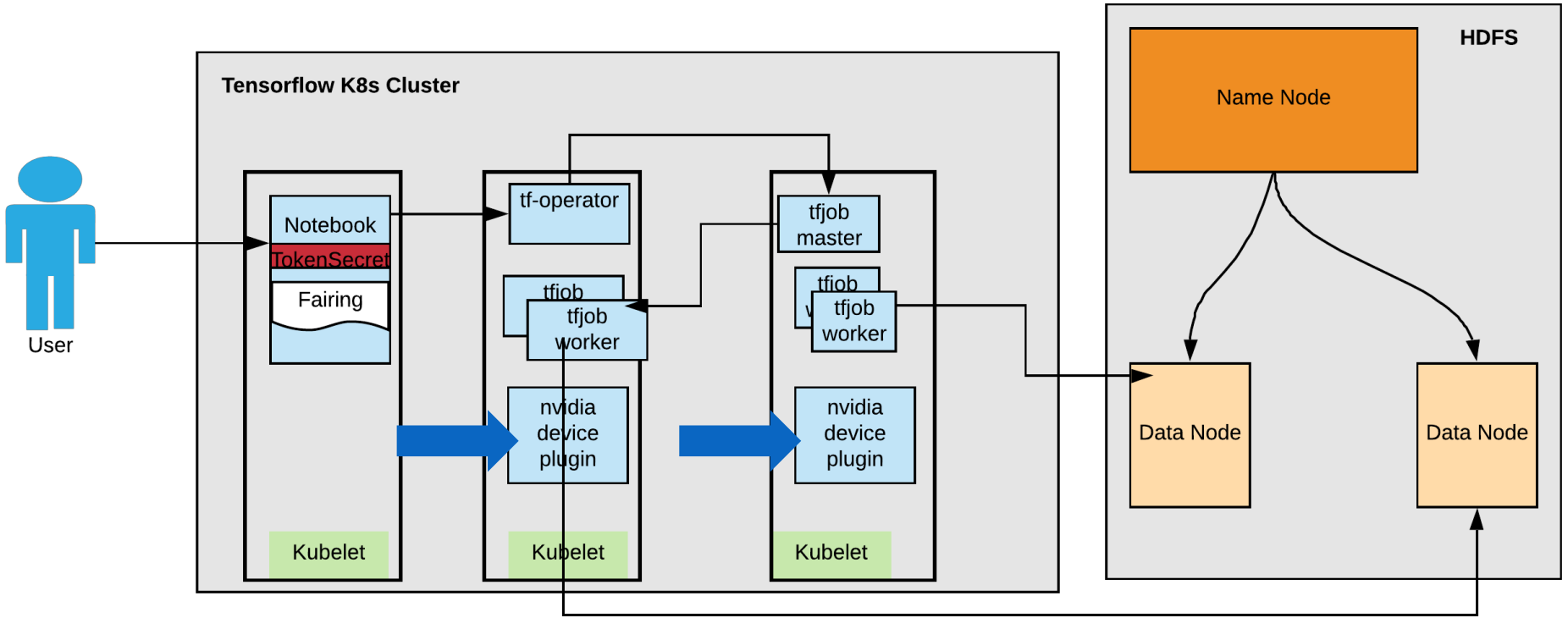
Distributed ML Training



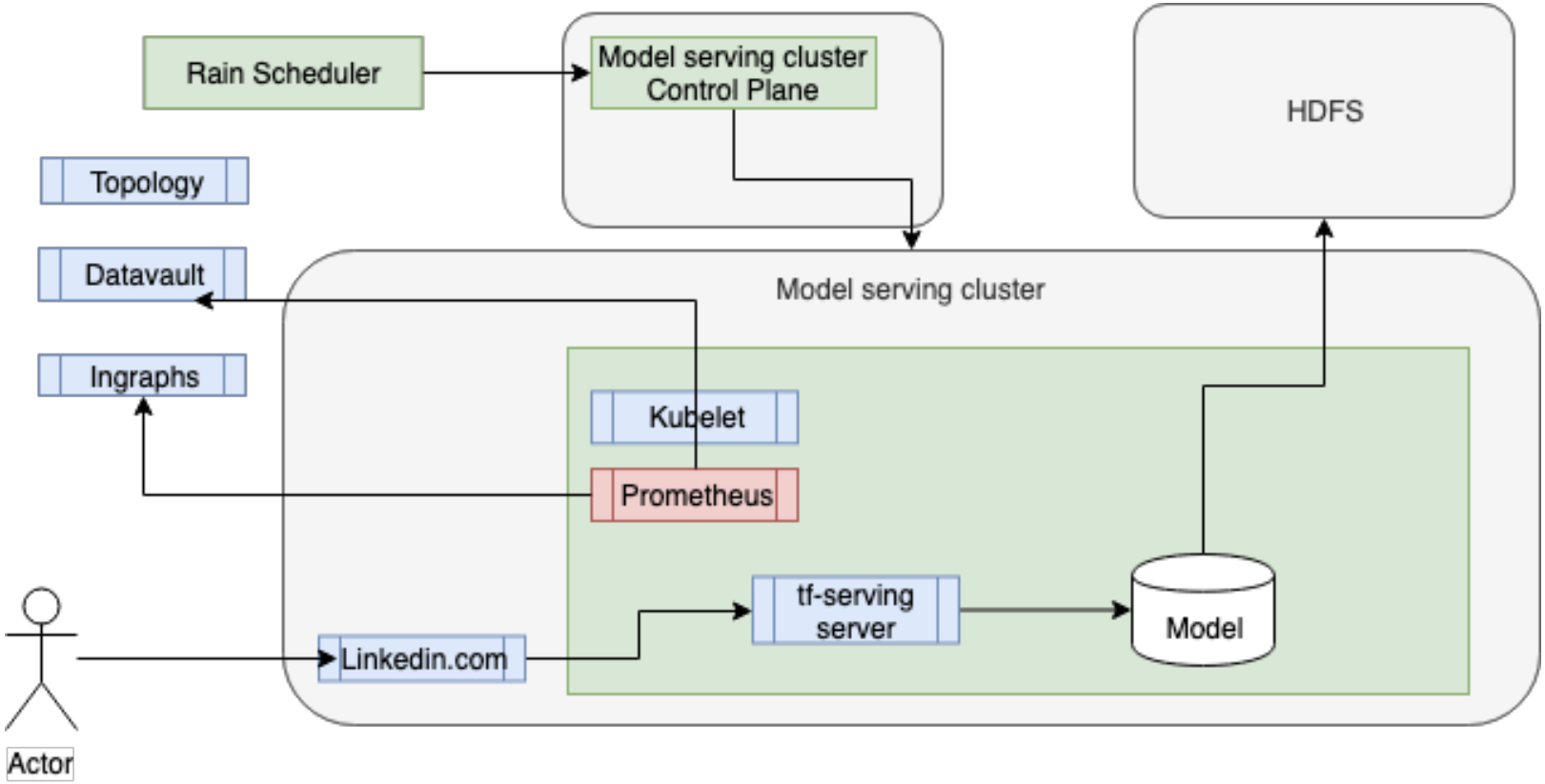
Distributed ML Training



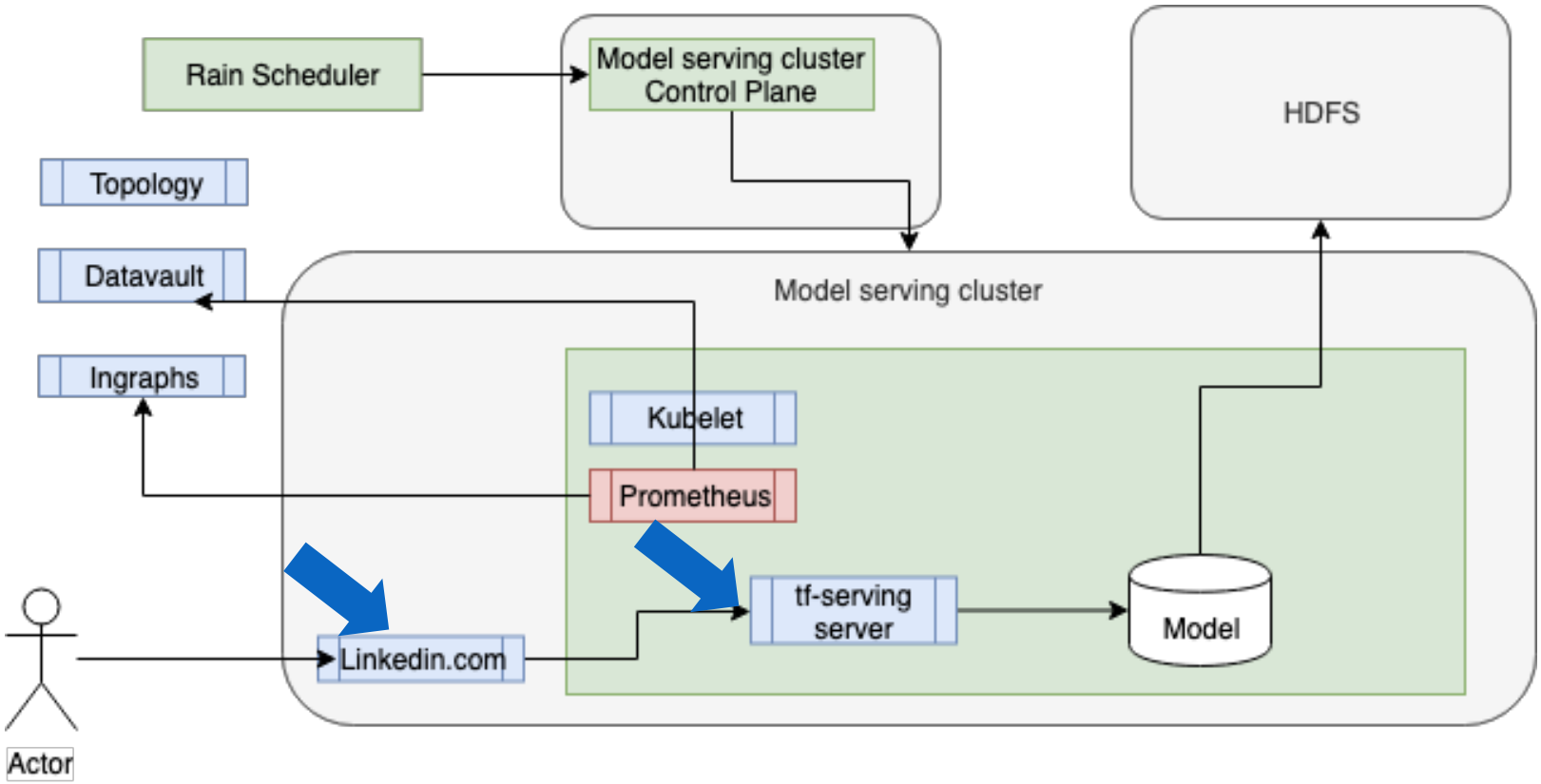
Distributed ML Training



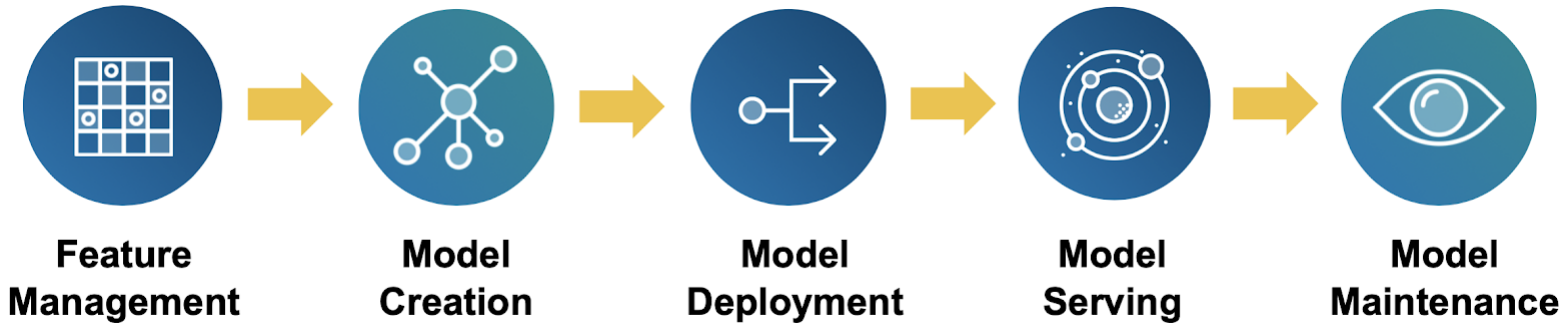
Online Model Serving



Online Model Serving



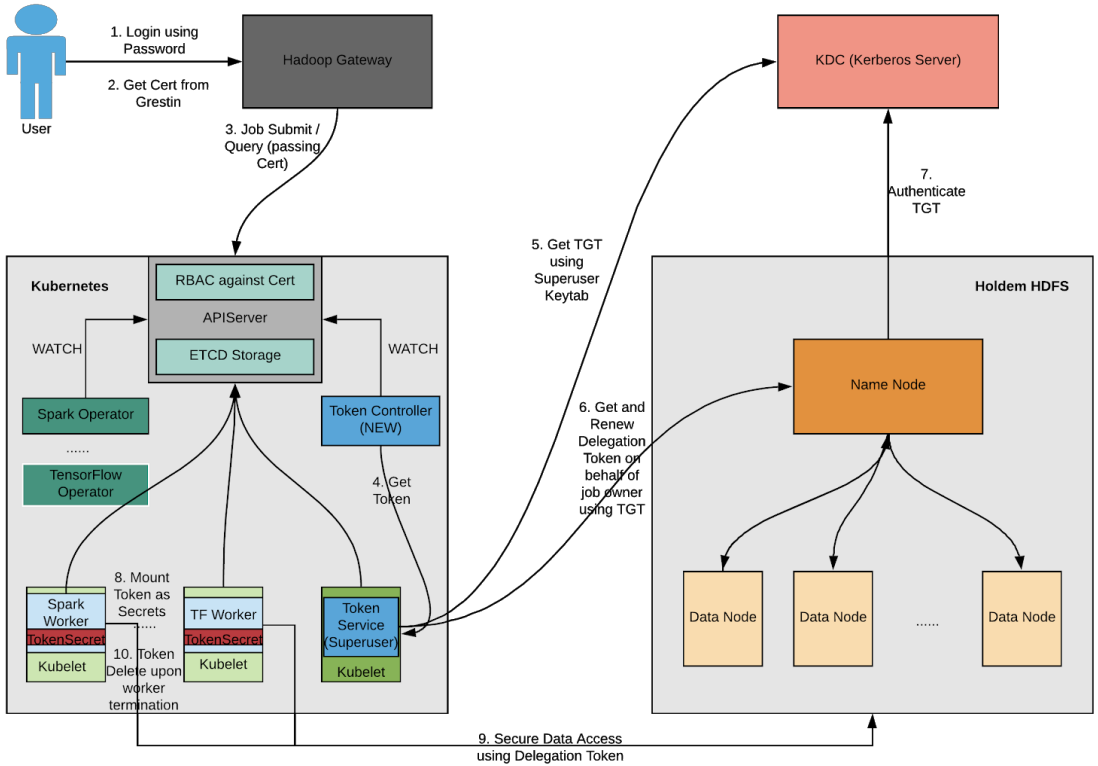
New ML Pipeline



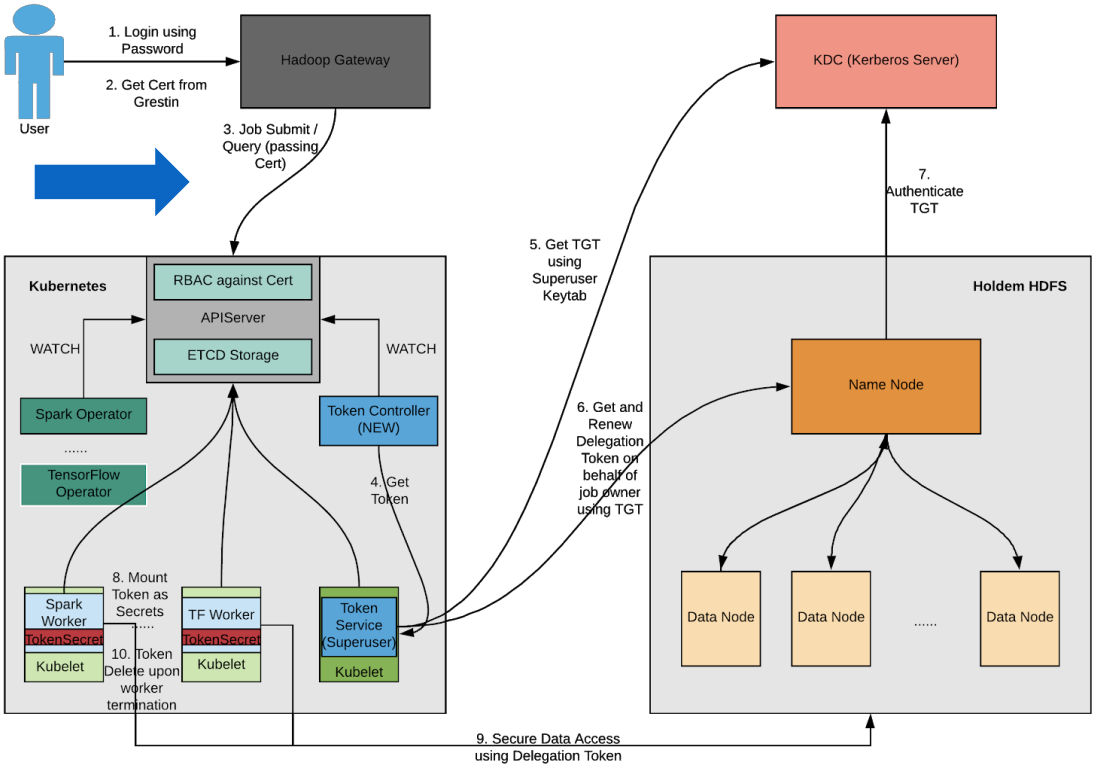
Kubeflow

hadoop YARN

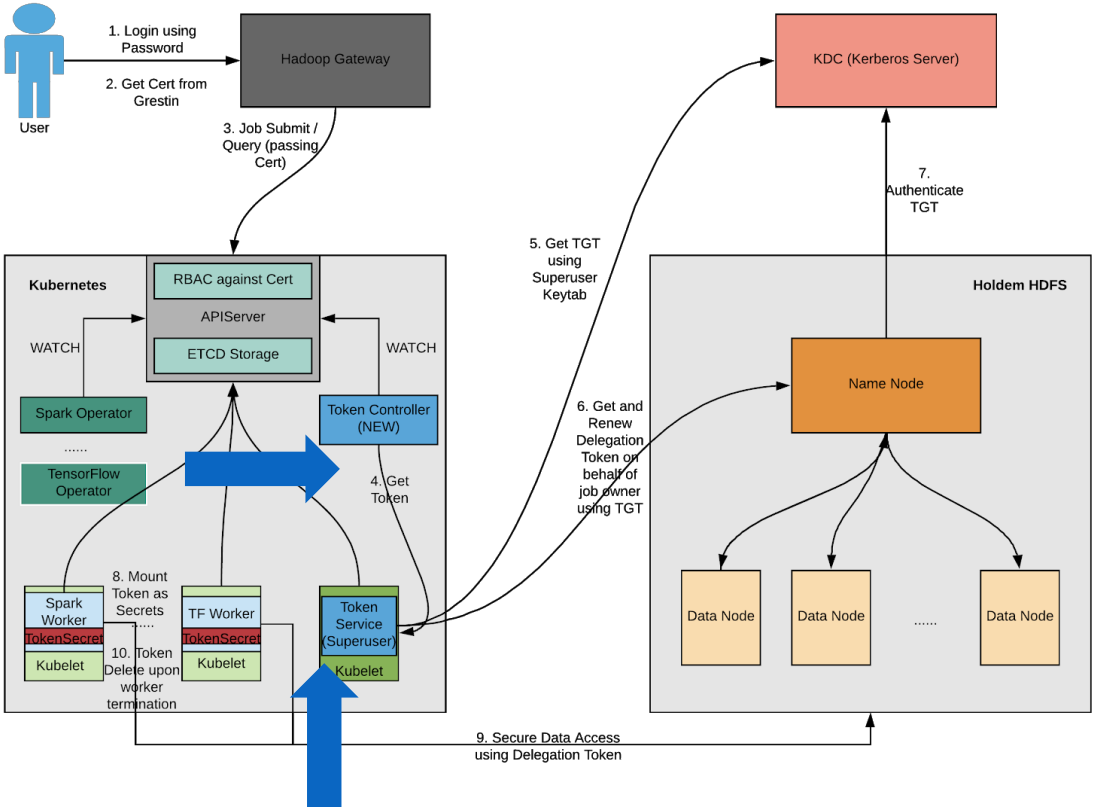
Secure HDFS Access



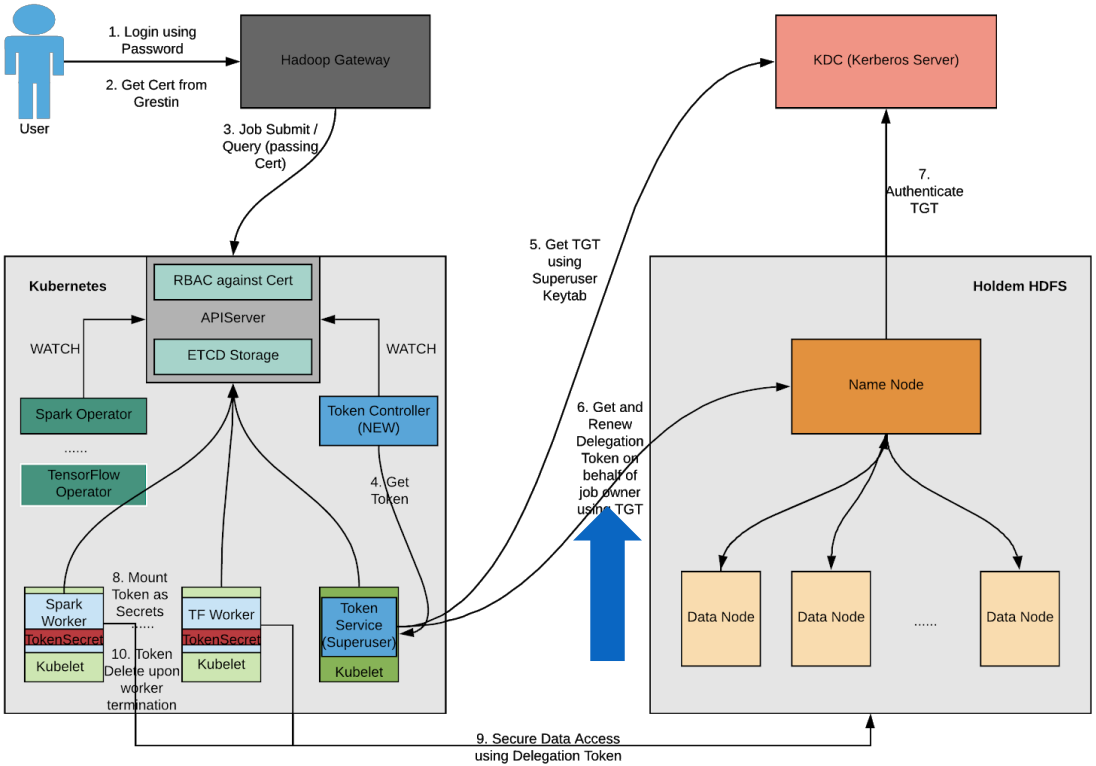
Secure HDFS Access



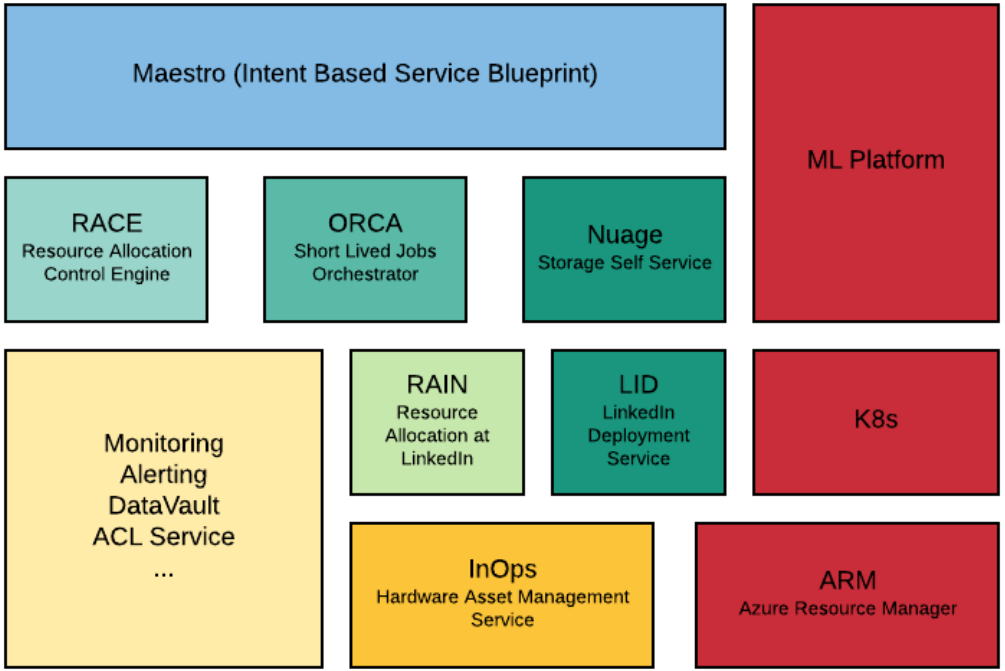
Secure HDFS Access



Secure HDFS Access



Unified Cluster Management Platform



Ecosystem Integrations

- PaaS integrations
- RBAC and Cert integrations
- Logging and Monitoring integration

Onbox Integration

- K8s CRI implementation for Locker
- Init container abstracts LinkedIn's application runtime logic

Wrap Up

- K8s ecosystem is powerful and growing fast
- Day 2 operation and integration are non-trivial
- Supporting emerging AI workloads to show value is a good start
- ‘Cloud native’ has huge value for the enterprise

Thank you

LinkedIn