Adapting to a unified and pluggable cluster management platform

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

Maestro (Intent Based Service Blueprint)

RACE
Resource Allocation Control Engine

ORCA
Short Lived Jobs Orchestrator

Nuage
Storage Self Service

Monitoring
Alerting
Logging
DataVault
Cert Management ...

RAIN
Resource Allocation at LinkedIn

LID
LinkedIn Deployment Service

InOps
Hardware Asset Management Service
RAIN

- Allocates resources from the common available pool of resources in LinkedIn’s data centers.
- Manages resource pools and host lifecycle
- Generates the deployment plan and sends to the host launcher, which starts the application on physical host
- Enforces RBAC and other deployment policies
• LinkedIn’s abstraction layer (over runc) for providing containerization on Linux hosts
• Resource and Namespaces Isolation
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

User

Rain Scheduler

Deployment DB

Cert Server

INGraphs

Rain app
Pod
Machine

Deployment CLI

Kubernetes cluster 1
control plane

ETCD Storage

RAIN-agent

Jupyterhub cluster

Notebook

Prometheus

Coredns

Kubelet

Tensorflow cluster

Notebook

tfjob

Prometheus

Kubelet

Kubelet
Jupyterhub on K8s

- User
- ML Platform UI
- Jupyterhub on Kubernetes
- Cert Server
- Notebook (includes Secret, Sparkmagic)
- Kubelet
- HDFS (includes Name Node, Data Node, Livy)
Jupyterhub on K8s
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 2
K8s Day 2: War Story 2
K8s Day 2: War Story 2

User

ML Platform UI

Jupyterhub on Kubernetes

Cert Server

Jupyterhub

Flanneld
Kubelet

Notebook

InterSecret
Sparkmagic

HDFS

Name Node

Data Node

Data Node

Livy
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=\n  jupyterhub.apiserver.linkedin.com

get-certs
KUBERNETES_SERVICE_HOST=\1.apiserver.linkedin.com

nginx
KUBERNETES_SERVICE_HOST=""

hub

Kubelet
Distributed ML Training

Tensorflow K8s Cluster

- Notebook
  - TokenSecret
  - Fairing

- tf-operator
  - tf-job
  - tf-job worker
  - nvidia device plugin

- tf-job master
  - tf-job
  - tf-job worker
  - nvidia device plugin

Kubelet

HDFS

- Name Node
  - Data Node
  - Data Node
Distributed ML Training
Distributed ML Training
Online Model Serving
Online Model Serving
New ML Pipeline

- Feature Management
- Model Creation
- Model Deployment
- Model Serving
- Model Maintenance

Logos: Kubeflow, YARN
Secure HDFS Access
Secure HDFS Access
Secure HDFS Access

1. Login using Password
2. Get Cert from Greslin
3. Job Submit / Query (passing Cert)

User

Kubernetes

Spark Operator
TensorFlow Operator

RBAC against Cert
APIServer
ETCD Storage
Token Controller (NEW)

Holdem HDFS

Name Node

Data Node

Token Service Supervision
Kubelet

TF-Worker
Kubelet

KDC (Kerberos Server)

5. Get TGT using Supernode Keytab
6. Get and Renew Delegation Token on behalf of job owner using TGT
7. Authenticate TGT

10. Token Delete upon worker termination
9. Secure Data Access using Delegation Token
8. Mount Token as Secrets

......
Secure HDFS Access

1. Login using Password
2. Get Cert from Greslin
3. Job Submit / Query (passing Cert)
4. Get Token
5. Get TGT using Supervisor Keypair
6. Get and Renew Delegation Token on behalf of job owner using TGT
7. Authenticate TGT
8. Mount Token as Secrets
9. Secure Data Access using Delegation Token

User

Kubernetes

RBAC against Cert

ETCD Storage

Spark Operator

TensorFlow Operator

Spark Worker

TF Worker

Token Controller (NEW)

Token Service Supersede

Kubelet

Kubelet

Kubelet

KDC (Kerberos Server)

Hadoop HDFS

Name Node

Data Node

Data Node

Data Node
Unified Cluster Management Platform

Maestro (Intent Based Service Blueprint)

RACE
Resource Allocation Control Engine

ORCA
Short Lived Jobs Orchestrator

Nuage
Storage Self Service

ML Platform

Monitoring
Alerting
DataVault
ACL Service
...

RAIN
Resource Allocation at LinkedIn

LID
LinkedIn Deployment Service

K8s

InOps
Hardware Asset Management Service

ARM
Azure Resource Manager
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