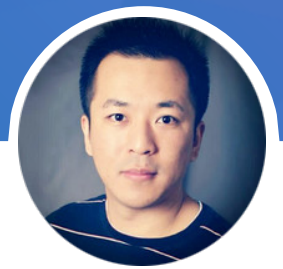
# Lesson learned on Running Hadoop on Kubernetes



**Chen Qiang**

Data SRE@LinkedIn

cqiang@linkedin.com

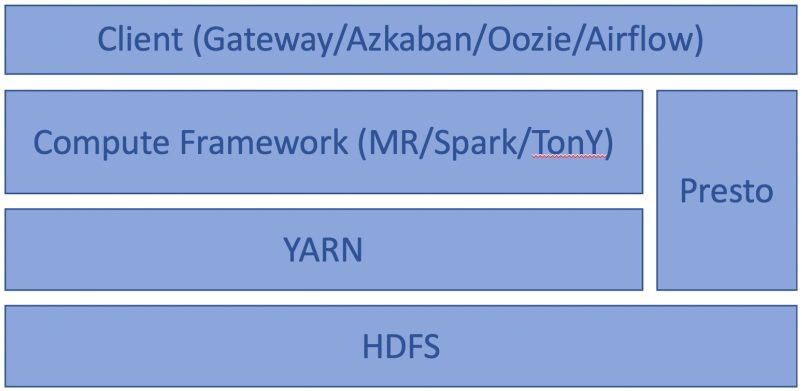
www.linkedin.com/in/cqiang/

# Today's agenda

- What's Hadoop?
- Hadoop@LinkedIn
- Initiatives
- Architecture
- Challenges
- What's next?
- Key take-ways
- Demo

# What is Hadoop?

“Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and big data processing”




### HDFS

Hadoop Distributed File System (HDFS)

- POSIX-like file system
- Designed for distributed computing frameworks
- Highly reliable and resilient

### YARN

Yet Another Resource Negotiator (YARN)

- Distributed workflow scheduler
- Resource allocation
- Job monitoring

### Compute Framework

MapReduce

- Hive (Interactive SQL)
- Pig

Spark

TonY (TensorFlow on Yarn)

Presto

### Client

Gateway

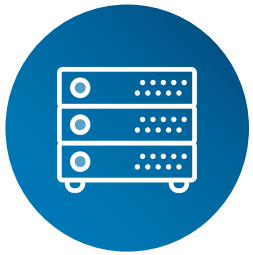
- CLI access

Azkaban/Oozie/Airflow

- Workflow scheduler and orchestrator

Hosted Notebook (e.g. Jupyter)

# Hadoop@LinkedIn



## Footprint

10+ Hadoop clusters ^

Largest Hadoop cluster is 7000+ servers with capacity of 400+ PB

Multiple large clusters with 4000+ servers

## Users

Thousands of individual and service accounts

Thousands of groups for data access

## Throughput

HDFS
- Avg read 600+GB/s
- Avg write 600+GB/s

YARN
- 300,000+ daily jobs with hundred millions of containers
- 1000+ container allocation per second

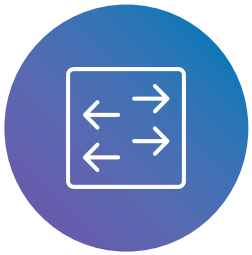
## QPS

Avg HDFS Namenode RPC 100K+

LDAP 150K+

KDC 5K+

DNS 95K+



## Network

Inbound and outbound network traffic is over 15Tbps

^ None of production cluster is running on Kubernetes

# Initiatives

## Testing

Every individual has own cluster that has very similar set up to production clusters

Immutable images, i.e. issue is always reproducible

## Efficiency

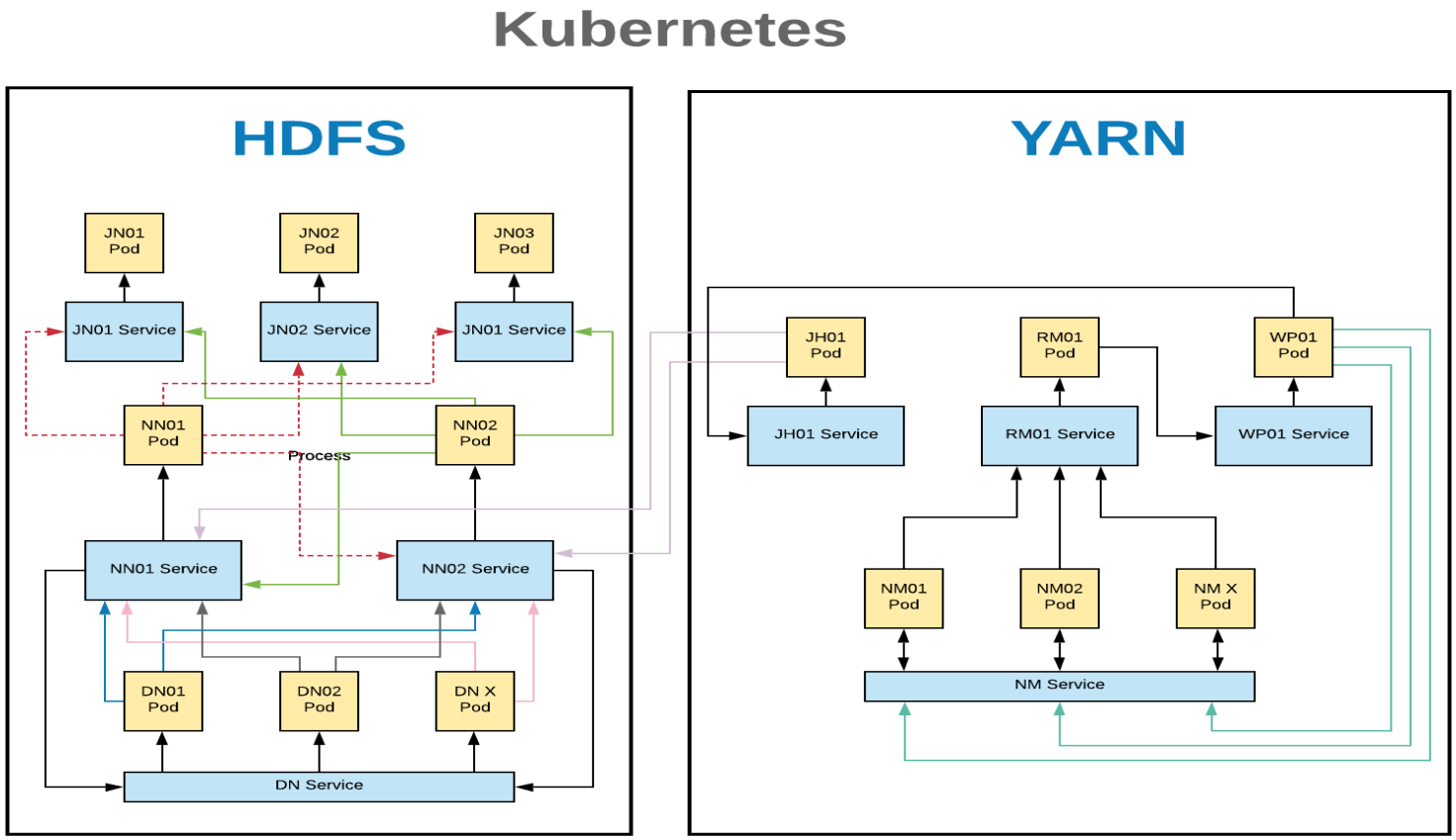
3-mins to spin up a secured(Kerberos-enabled) Hadoop cluster

No inter-team and/or cross-team dependencies

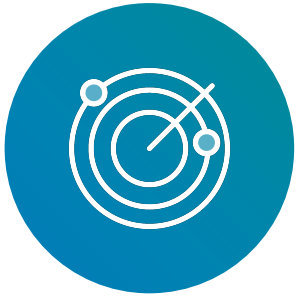
## Technology

Explore new technologies to boost efficiency and productivity

# Architecture



# Challenges



## DNS

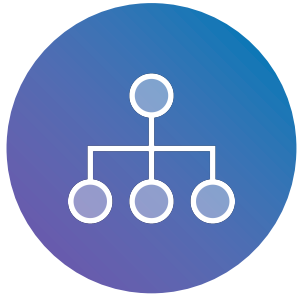
Each worker pod does not have global DNS resolvable hostname

## Network

No VIP or fixed IP address for Hadoop master services

## Identity

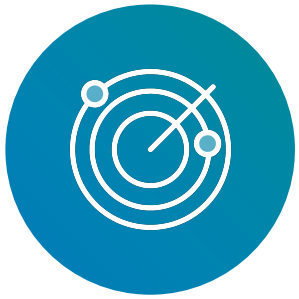
Cannot pre-generate Kerberos keytab due to unknown IP address and/or hostname



## Orchestration

Hadoop components requires to start in certain order

# DNS



**DNS**

- Each worker node pod does not have global DNS resolvable hostname

**Problem**

- Hadoop worker (Yarn nodemanger and HDFS datanode) does not have a global resolvable hostname.
- In secure Hadoop cluster, authentication also requires hostname matching from reverse lookup with node's kerberos principal, i.e. yarn/_HOST@KERBEROS.REALM
- webHDFS (HDFS REST API) while namenode sends redirect request back to client with hostname of a designated datanode.
- MapReduce application master or Spark driver's hostname needs to be advertised to all its workers

**Solution**

- StatefulSet with headless service gives unique resolvable hostname to every pod, however lookups are not consistent

```
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# hostname –f
kubecon2020-demo-hdfs-dn-1.kubecon2020-demo-hdfs-dn.k8s.svc.kube.mydomain.com
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# host kubecon2020-demo-hdfs-dn-1.kubecon2020-demo-hdfs-dn.k8s.svc.kube.mydomain.com
Host kubecon2020-demo-hdfs-dn-1.kubecon2020-demo-hdfs-dn.k8s.svc.kube.mydomain.com not found: 3(NXDOMAIN)
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# host 10.244.97.181
181.97.244.10.in-addr.arpa domain name pointer 10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com.
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# cat /etc/hosts
10.244.97.181  kubecon2020-demo-hdfs-dn-1.kubecon2020-demo-hdfs-dn.k8s.svc.kube.mydomain.com
kubecon2020-demo-hdfs-dn-1
```

- Inject resolvable hostname into /etc/hosts in main container

```
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# cat /etc/hosts
10.244.97.181  10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com kubecon2020-demo-hdfs-dn-1.kubecon2020-
demo-hdfs-dn.k8s.svc.kube.mydomain.com kubecon2020-demo-hdfs-dn-1
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# hostname –f
10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com.
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# host 10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com
10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com has address 10.244.97.181
[cqiang@kubecon2020-demo-hdfs-dn-1 /]# host 10.244.97.181
181.97.244.10.in-addr.arpa domain name pointer 10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com.
```

# Network

**Network**

- No VIP ( or fixed IP addresses) for Hadoop master services

**Problem**

- There are many components in Hadoop, and they are communicating with each other while the cluster is up and running.
  - Datanode initiates connection to Namenode
  - Nodemanager initiates connection to ResourceManager
- And many other intercommunication between components
- Master hostname needs to be in Hadoop configuration across all components

**Solution**

- Create Kubernetes Service for every Hadoop admin instance. The service is to provide a pre-defined and structured DNS resolvable hostname which can be pre-determined in Hadoop configuration files. e.g. using service hostnames as namenodes in hdfs-site.xml

```
<property>
   <name>dfs.namenode.http-address.kubecon2020-demo.ha1</name>
   <value>kubecon2020-demo-hdfs-nn1-svc.k8s.svc.kube.mydomain.com:50070</value>
</property>
<property>
   <name>dfs.namenode.https-address.kubecon2020-demo.ha2</name>
   <value>kubecon2020-demo-hdfs-nn2-svc.k8s.svc.kube.mydomain.com:50070 </value>
</property>
```

# Identity

**Identity**

- Cannot pre-generate keytabs due to unknown IP address and/or hostname

**Problem**

- This problem ONLY applies to secure Hadoop cluster with Kerberos-enabled.
- Secure Hadoop cluster uses keytab files for authentication and usually it has hostname as part of the Kerberos service principal, e.g. hdfs/10-244-97-181.kubecon2020-demo-hdfs-dn-svc.k8s.svc.kube.mydomain.com@KERBEROS.REALM.
- Since hostname has random IP address, thus keytab cannot be pre-generated.

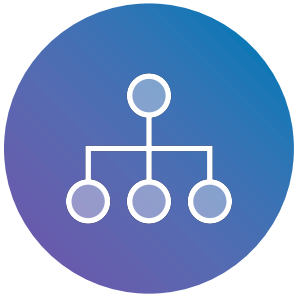
**Solution**

- Introduced Keytab Delivery Service (KDS) in Kubernetes to serve keytab generation request from other pods. When a Hadoop pod is launched, it calls KDS for requesting a keytab as part of initContainer. KDS authenticates the request and replies with QA only keytab back to client. It uses the same authentication mechanism from recently LinkedIn Open-sourced Kube2Hadoop*.

```
initContainers:
  - name: keytab
    image: container-registry/keytab_fetch_image:latest
    env:
      - name: SVCPRINC
        value: "hdfs/_IP_.kubecon2020-demo-hdfs-dn-svc. k8s.svc.kube.mydomain.com"
      - name: HTTPPRINC
        value: "HTTP/_IP_.kubecon2020-demo-hdfs-dn-svc. k8s.svc.kube.mydomain.com"
    volumeMounts:
      - name: shared-data
        mountPath: /common/keytab/location
```

* https://engineering.linkedin.com/blog/2020/open-sourcing-kube2hadoop
https://github.com/linkedin/kube2hadoop

# Orchestration



## Orchestration

- Hadoop components requires to start in certain order

**Problem**

- There is a strong dependency in Hadoop components, hence the bootstrap order is very critical. For example HDFS consists of 3 major components, journalnodes (metadata edits), namenodes( serves metadata) and datanodes (data). Starting sequence must be first Jouanalnodes then Namenodes
- A simple solution is to orchestrate such sequence externally, but it introduces additional complexity and long deployment duration.

**Solution**

- Built-in dependency using initContainer with Kubernetes service discovery, hence all pods can be deployed simultaneously. It cuts the cluster deployment time down to 2 mins for a full secure Hadoop cluster including HDFS and Yarn.

```
initContainers:
  - name: init
    image: container-registry/dependency_check_image:latest
     command: ["/bin/sh"]
      args: ["-c", "sh /checking_dep journalnode"]

    env:
        - name: "CLUSTER_NAME"
          value: "KUBECON2020_DEMO"
```

# What's next?

- Extend ephemeral environment to have more big data components – in progress
  - Spark
  - Hive
  - Presto
  - Azkaban
- Long running Hadoop cluster on Kubernetes – in progress
- Kubernetes Custom Resources Definition (CRD) and Hadoop Operator

# Audience take-away

- Opens the opportunity to combine different workload onto same resources management platform, i.e. Kubernetes
- May change the way how big data components are running
- Make Hadoop to Kubernetes migration/transition easier

# Demo