Sesión 11 Jenkins

big data (Cloudera)

cloudera-introduction.pdf

- MapReduce → Spark
  - MapReduce : repartir la carga entre X servidores que no tengan colisiones y meterlo en un lugar común
  - Hive : SQL to MapReduce
  - Pig : lenguaje para no SQL (pero que se le parece tanto...)
  - kafka : cluster
  - impala : consumo intensivo RAM, accede al FS directamente (sin pasar por HDFS)
    - HDFS : Hadoop Distributed FS (Java) → lento
  - SOLR : indexador
  - Kite : ¿api?
  - Resource Management
    - YARN : Yet Another Resource Negotation
    - sistema de colas
      - Filesystem: HDFS
      - Relational: kudu
  - Security
    - Sentry : kerberos
    - RecordService: ¿registro a qye se ha accedido?
      - NoSQL: HBase (el perdedor contra Casandra)
  - Storage
    - Batch
      - sqoop
    - Real-Time
      - Kafka
- Hadoop Cluster
  - NameNode (de 2 a 3, este último para Journal)
    - zookeeper : service discovery
    - secondaryNameNode : helper
    - Resource manager (YARN)
  - DataNode
    - Just a Band of Disc
    - RAID1 para disco OS
    - hacen el trabajo
  - HDFS
    - sistema raid
    - bloques de 128MG
    - replicación (3 dataNode)
    - reglas de afinidad de la replicación
    - sueltas el archivo de Gb o Tb
  - Beowulf cluster
    - https://es.wikipedia.org/wiki/Cluster_Beowulf
    - https://es.wikipedia.org/wiki/Grace_Murray_Hopper
      - mejor 2 equipos que un equipo con el doble

Jenkins

en otras sesiones

GitLab demo 1: GitLab demo 1 Job:




```
docker run --name demo1-${BUILD_NUMBER} --rm -d gitlab/nginx-demo1
sleep 15
CONTAINER_IP=$(docker inspect --format '{{ .NetworkSettings.IPAddress }}' demo1-
${BUILD_NUMBER})
CODE_STATUS=$(curl -sI ${CONTAINER_IP} | grep -c "^HTTP/1.1 200")
docker stop demo1-${BUILD_NUMBER}

#[[ ${CODE_STATUS} == 1 ]] && { echo "OK"; exit 0; } || { echo "KO";exit 1; }
if [ ${CODE_STATUS} -eq 1 ]; then {
    exit 0;
} else {
    exit 1;
}
fi
```

gitlab demo 1 pipeline:

```
node {
   stage("build") {
        def job = build job: 'gitlab demo1'
   }

   stage("execute") {
        def job = build job: 'gitlab demo1 job'
   }
   stage("publish") {
        def job = build job: 'gitlab demo1 publish'
   }
}
```

gitlab demo1 publish:

```
node ("docker-agent") {
    stage ("tag") {
        sh ('docker tag gitlab/nginx-demo1:latest
registry.floss.cat/mate/demo1:latest')
    }
    stage ("push") {
        withCredentials([usernamePassword(credentialsId: 'registry.floss.cat',
usernameVariable: 'USERNAME', passwordVariable: 'PASSWORD')]) {
            sh('''
                docker login --username=${USERNAME} --password=${PASSWORD}
registry.floss.cat
                docker push registry.floss.cat/mate/demo1:latest
            ''')
        }
    }
}
```

pipelines

- https://jenkins.io/doc/book/pipeline/docker/
- https://jenkins.io/doc/book/pipeline/syntax/
- https://jenkins.io/doc/book/pipeline/jenkinsfile/
- https://github.com/stefanprodan/dockerdash/blob/master/Jenkinsfile
- blue-ocean
    - Jenkinsfile:
        - SCM
        - pipeline as code

k8s

- https://medium.com/@geraldcroes/kubernetes-traefik-101-when-simplicity-matters-957eeede2cf8
    - traefik + kubernetes
    - deployment for traefik: https://docs.traefik.io/user-guide/kubernetes/
- calico: driver red para kubernetes + policy (para establecer como se ven los servicios de un namespace)
- https://kubernetes.io/docs/tasks/tools/install-minikube/

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/info:cursos:pue:devops:sesion11

Last update: **22/03/2019 20:45**