CLUSTER CONTINUOUS DELIVERY WITH OOZIE

Clay Baenziger – Bloomberg Hadoop Infrastructure

ApacheCon Big Data – 5/18/2017

ABOUT BLOOMBERG

BIG DATA AT BLOOMBERG

Bloomberg quickly and accurately delivers business and financial information, news and insight around the world.

A sense of scale:

- 550 exchange feeds and over 100 billion market data messages a day
- 400 million emails and 17 million IM’s daily across the Bloomberg Professional Service
- More than 2,400 journalists in over 120 countries
  - Produce more than 5,000 stories a day
  - Reaching over 360 million homes world wide

BLOOMBERG BIG DATA APACHE OPEN SOURCE

Solr: 3 committers – commits in every Solr release since 4.6

Project	JIRAs	Project	JIRAs	Project	JIRAs
Phoenix	24	HBase	20	Spark	9
Zookeeper	8	HDFS	6	Bigtop	3
Oozie	4	Storm	2	Hive	2
Hadoop	2	YARN	2	Kafka	2
Flume	1	HAWQ	1	**Total***	**86**

** Reporter or assignee from our Foundational Services group and affiliated projects*

APACHE OOZIE

What is Oozie:

- Oozie is a workflow scheduler system to manage Apache Hadoop jobs.
- Oozie workflow jobs are Directed Acyclical Graphs (DAGs) of actions.
- Oozie coordinator jobs are recurrent Oozie workflow jobs triggerd by time and data availability.
- Oozie is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs as well as system specific jobs out of the box.
- Oozie is a scalable, reliable and extensible system.

Paraphrased from: http://oozie.apache.org/

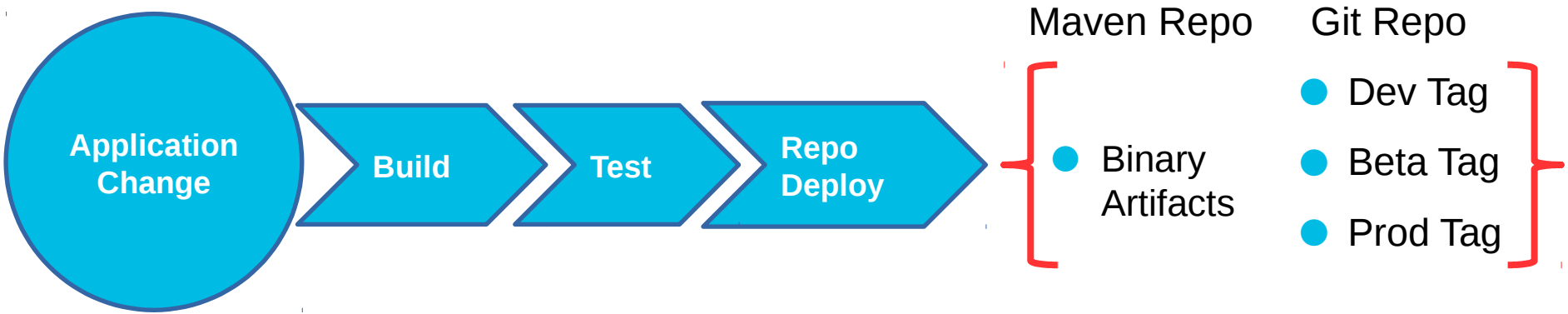
Actions:

- Map/Reduce
- Hive
- Pig
- HDFS
- Java
- Shell
- Spark
- Sub-Workflow
- E-Mail
- Decision
- Fork
- Join

LET’S TALK DEPLOYMENT

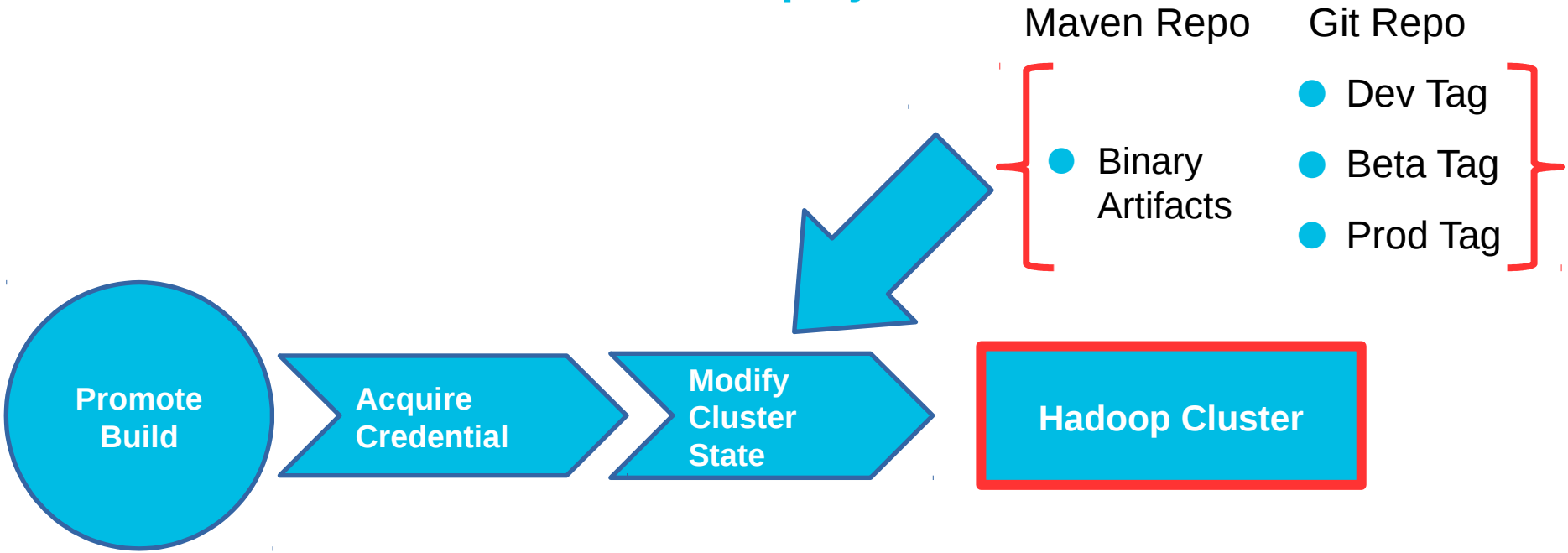
CONTINUOUS INTEGRATION MODEL

Jenkins / Build Server Driven Builds:



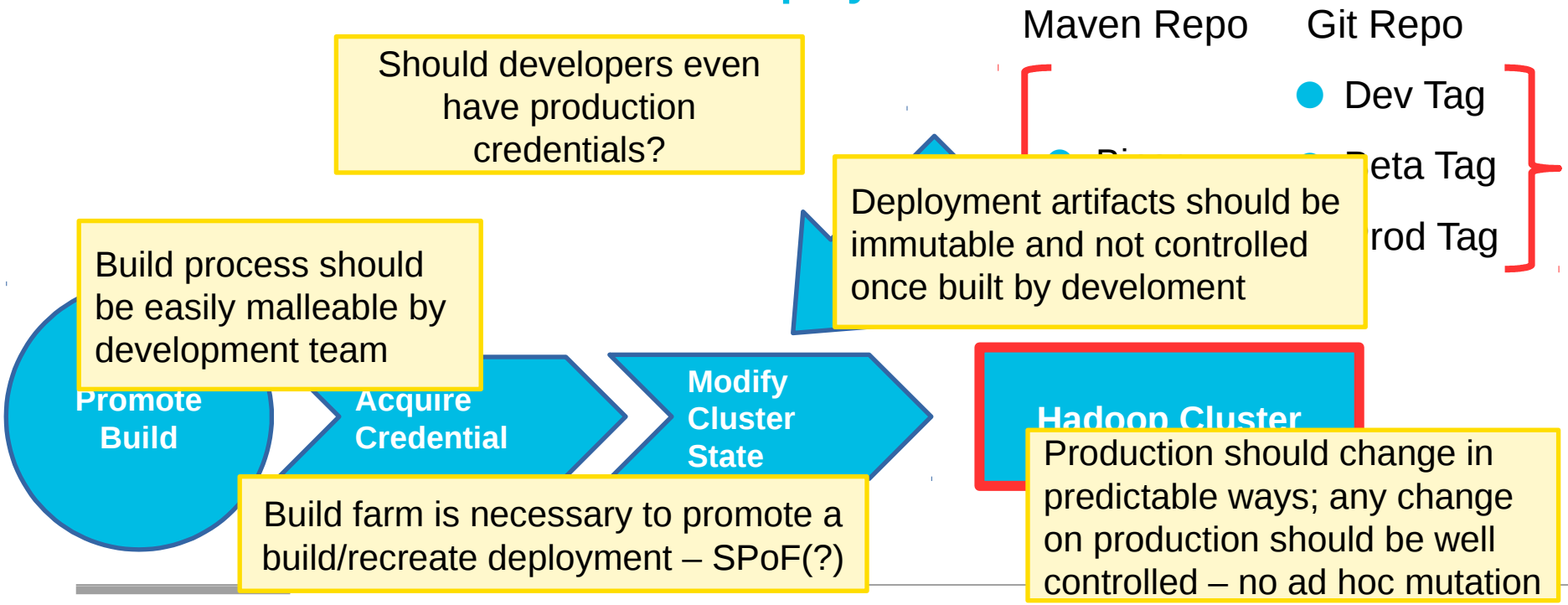
CONTINUOUS DELIVERY MODELS

Jenkins / Build Server Driven Deployments:



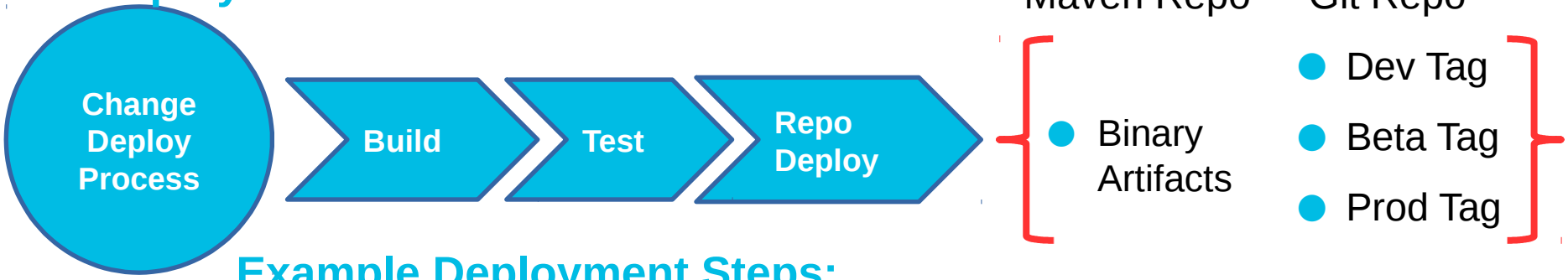
CONTINUOUS DELIVERY MODELS

Jenkins / Build Server Driven Deployments:



CONTINUOUS INTEGRATION MODEL

Deployment Process as Code:



Example Deployment Steps:

- Update Apache HBase tables and schemas – e.g. code in Java
- Create Apache HDFS directories – e.g. code in shell scripts
- Stage code in HDFS which actually processes data – e.g. Spark
- Submit Apache Oozie workflow to execute product – how code runs

CONTINUOUS INTEGRATION MODEL

Deployment Process as Code Axioms:

- **Idempotency** – able to be called repeatedly and produce the same result
- **Cleanliness** – will need to clean up unused deployment artifacts
- **Separation of Config from Code** – to allow the same deployment process to work for different environments configuration must be separate and environments (dev, beta, and prod.) should be as similar as possible
- **Build, Release, and Run Stages** – code is separately built, released/deployed and run

Adapted from "The Twelve-Factor App" - https://12factor.net

EXAMPLE DEPLOYMENT ARTIFACTS

Binary – Apache Maven:

- Apache Hadoop Map/Reduce Programs and Dependencies
- Apache Spark Programs and Dependencies
- Apache Slider Tarballs

ASCII – Git:

- Apache Oozie Workflows
- Apache Pig, Apache Hive, PySpark Scripts
- Apache HBase, Apache Phoenix and Apache Hive Schemas

OOZIE ACTIONS

OOZIE-2877 - OOZIE GIT ACTION

```
<workflow-app xmlns="uri:oozie:workflow:0.4" name="git-example">
    <start to="Clone_Repo"/><action name="Clone_Repo">
        <git xmlns="uri:oozie:git-action:0.1">
            <job-tracker>yarnName:8032</job-tracker><name-node>hdfs://hdfsName</name-node>
            <git-uri>git@github.com:apache/oozie</git-uri>
            <ssh-key-path>mySecureKey</ssh-key-path>
            <destination-uri>myRepoDir</destination-uri>
        </git>
        <ok to="end"/><error to="kill_job"/>
    </action><kill name="kill_job"><message>Job failed</message></kill><end name="end"/>
</workflow-app>
```

GIT ACTION OPTIONS

Various Git Options Supported:

- Clone (overwrite) a whole repo
- Clone (overwrite) a collection of paths
- Use an ssh-key for secured repositories
- What else? – Please provide your feedback via JIRA (OOZIE-2877)

Various Pitfalls Avoided:

- Secure key management in YARN container
- Don’t have to worry about error handling
- Not another shell action

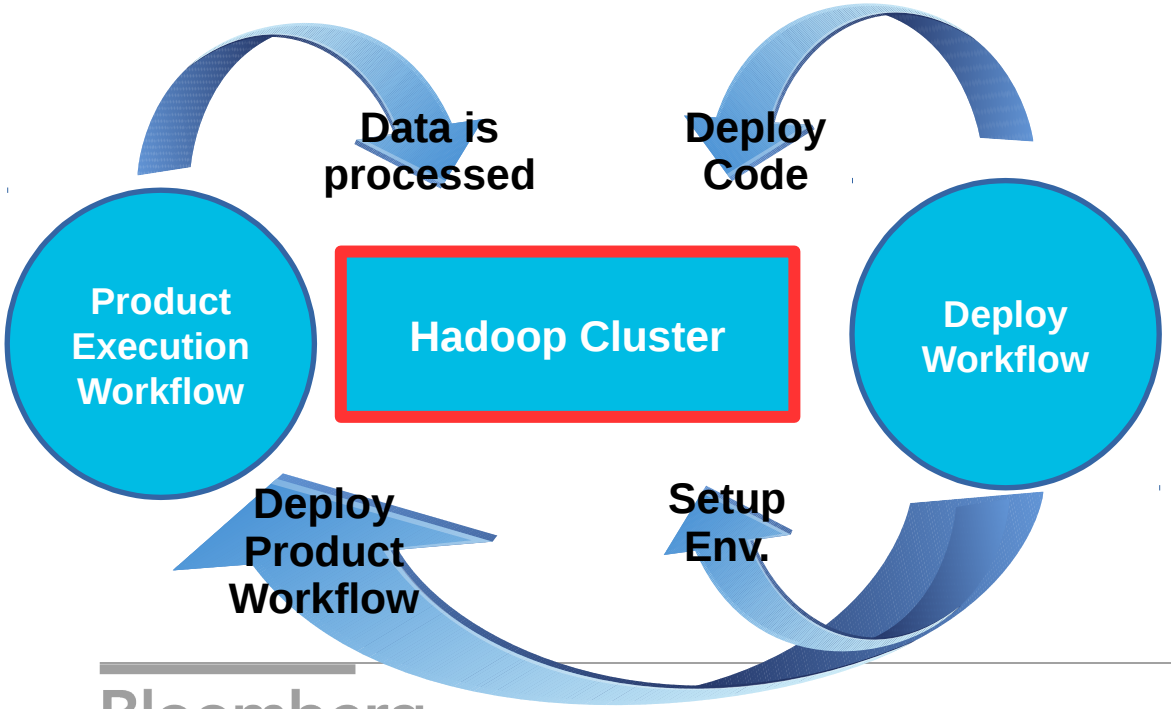
OOZIE-2878 – OOZIE MAVEN ACTION

A Means to Deploy Binaries:

- Build an assembly on the cluster?
  - Not attributable to a build then
- Use Maven Dependency Plugin?
  - Pulls dependencies
  - Artifacts are unmodified
- Need to handle Maven's desire to maintain a local repository
- Need to handle providing Maven a view of on "disk" (HDFS) state

CONTINUOUS DELIVERY MODELS

Cluster Driven Deployments:



Maven Repo

- Binary Artifacts

Git Repo

- Dev Tag
- Beta Tag
- Prod Tag

SEED JOB

SEED JOB

Seed Job: a means to deploy an application's deployment workflow.

- It is the only superuser executed operation.
- Everything for the application grows from it.
- Should be as simple and lightweight as possible.

A Seed Job Provides:

- Scripted deployment of developer provided deployment workflow.
- A standardized process for applications to "on-board" a new cluster.
- Allows for operational personnel to deploy an application without direct knowledge of the application.
- Allows for developers to control their deployment process running as their application account – without having permission to assume that account themselves.

SEED JOB – EXAMPLE STEPS

Super User Deploys Seed Job:

1. Coordinator job – *privileged code runs as super user*
2. Clones deployment workflow from Git and provisions any privileged resources
3. Submits deployment workflow to Oozie to run under application team role account

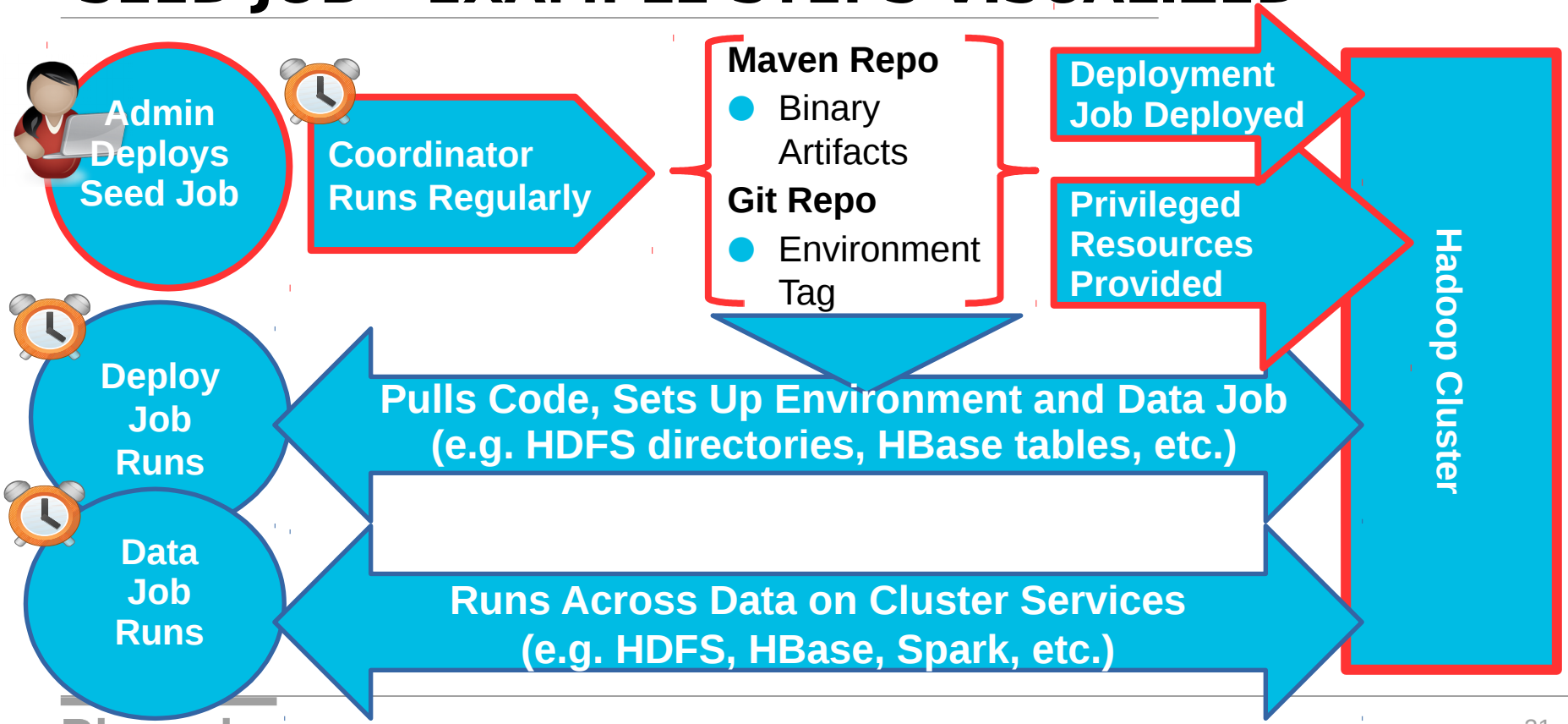
Application Role Account Runs Deployment Job:

4. Downloads application code
5. Verifies and creates/updates any HBase, Hive schemas or HDFS structures
6. Submits coordinator job(s) for data processing

Application Role Account Runs Data Processing Job:

7. Does any normal data processing as would any standard Oozie workflow

SEED JOB – EXAMPLE STEPS VISUALIZED



RECAPING CLUSTER PROS/CONS…

Hadoop Cluster Advantages

1. Highly-Available – Oozie HA
2. Visible/Recorded Logs – YARN
3. Supports Privilege Delegation (i.e. for initial seed job) – Delegation Tokens
4. Seed Jobs Provide Entry Point for Applications – No further administrative hand-holding is necessary

Hadoop Cluster Disadvantages

1. Conf. mgmt tools have no primitives to build upon – can't use Chef, Puppet, Ansible
    - Conf. mgmt. tools not cluster aware – races among competing converges
    - Not Hadoop native – HDFS, Spark, etc. are not first-class citizens
2. Oozie jobs run on arbitrary machines – no stable home directory

REAL WORLD EXAMPLE

HOW DO WE DEPLOY APPLICATIONS

Combination of Tools Available and Tools In-Use

- YARN Queue – Chef Based (First-Class Resource)
- HDFS Quota – Chef Based (Bash Resource)
- HBase Quota – TBD
- Project & Home HDFS Directory – Chef Based (Lots of Ruby and Shell)
- Seed Job – Oozie Based
- Deployment Job – Oozie Based

DEPLOY YARN QUEUE

Chef Example:

```
fair_share_queue 'application' do
     schedulingPolicy 'DRF'
     aclSubmitApps '@applicationTeamGroup'
     aclAdministerApps '@applicationTeamGroup'
     minResources '2960000mb, 650vcores'
     parent_resource 'fair_share_queue[groups]'
     subscribes :register, 'fair_share_queue[groups]',
                :immediate
     action :register
 end
```

http://bit.ly/2oWJkPr – GitHub.COM/Bloomberg/Chef-BACH/...

DEPLOY HDFS QUOTA

Chef Example:

```
bash 'set applicationTeam directory quota' do
  code <<-EOH
  hdfs dfsadmin -setSpaceQuota \
    #{node['...']['applicationTeam_quota']['space']} \
    #{node['...']['hdfs_url']}/groups/applicationTeam/ && \
  hdfs dfsadmin -setQuota \
    #{node[...']['applicationTeam_quota']['files']} \
    #{node['...']['hdfs_url']}/groups/applicationTeam/
  EOH
  user 'hdfs'
end
```

http://bit.ly/2oWJkPr – GitHub.COM/Bloomberg/Chef-BACH/...

Ew!

DEPLOY HBASE QUOTA

HBase Shell Action?

- Can be written as a Java or Shell action but…
- Could make schema changes easier
- Easy to provide a script; testing is easy
- Could even be used to return small row lookups for other uses

DEPLOY HDFS DIRECTORIES

Initially lots of LDAP scraping and heuristics:

In Chef, walk all provisioned users looking for:

1. A role account provisioned to the cluster
2. Walk its groups
3. Check each group for human users
4. Check if the humans are members of the cluster too
5. See if group is on a blacklist
6. If all checks out, create a /groups directory entry with permissive group permissions

Ew!

http://bit.ly/2pEguAi – GitHub.COM/Bloomberg/Chef-BACH/...

QUESTIONS?

Clay Baenziger - Bloomberg Hadoop Infrastructure

https://GitHub.COM/Bloomberg/Chef-BACH

Hadoop@Bloomberg.NET

Join the discussion:

OOZIE-2876 - Provide Deployment Primitives

SHORT OOZIE DEV LESSONS

Process of Writing an Oozie sharedlib/Action:

- Build with Hadoop 2 and dependencies: bin/mkdistro.sh -Phadoop-2,uber -Dhadoop.version=2.7.1
- Using modern libs for HBase will need to update pom's to pull hbase-client JAR
- Five key touch points:
    - sharelib.xml – actually rolls into sharelib tarball
    - OozieCLI.java – supposedly not needed post 4.2.0?
    - oozie-default.xml – recall if you override your oozie-site.xml too!
    - A few more pom.xml's
    - The action's code!