Bigtop 101

Guodong Xu
Biography

Guodong Xu

Joined Linaro in 2012

Worked on Linux kernel, and Big Data

My Shared Page:

[1] Bigtop (v1.3.0) Build, Smoketest, and Deploy on Multiple Physical Machines
[2] How to 'Consume' a Bigtop release - to get a repository URL
[4] Issues I met when running NutchIndexing and How I fixed them

Pages from Linaro Big Data Team:

[1] https://collaborate.linaro.org/display/BDTS/Big+Data+Team+Space
Abstract

This session walks you through Bigtop building, testing and deploying. Commands examples are provided, so it can be easily followed.

- What is Bigtop?
- Build from Source
- Smoke test
- Deploy on multiple physical machines

Note: There is opportunities to get hands-on with me via Linaro DevCloud.
Note: Reference environment is based on a CentOS 7. (Other supported OS includes: Debian 9, Fedora 26, Ubuntu 16.04, OpenSUSE 42)

Bigtop is an Apache Foundation Project.

- for Big Data
- for packaging
- for testing
- for deploying
What is Bigtop? - 2/ components

As of 1.3.0, it contains components:

- Alluxio
- Ambari
- Apex
- Crunch
- Datafu
- Flink
- Flume
- Giraph
- Gpdb
- Hadoop
- Hama
- Hbase
- Hive
- Kafka
- Mahout
- Phoenix
- Qfs
- Solr
- Spark
- Sqoop
- Tajo
- Tez
- Ycsb
- Zeppelin
- Zookeeper
Part I - Build from Source
Build from Source Code - 1/ docker install

- **Install docker**

  No secret here. Just follow docker’s official guide: [https://docs.docker.com/install/linux/docker-ce/centos/#set-up-the-repository](https://docs.docker.com/install/linux/docker-ce/centos/#set-up-the-repository)

  ```
  $ sudo yum install docker-ce docker-ce-cli containerd.io
  $ sudo systemctl start docker
  $ sudo systemctl enable docker
  $ sudo docker run hello-world
  $ sudo docker ps
  ```

  Note: If you would like to use Docker as a non-root user, you should now consider adding your user to the “docker” group with something like:

  ```
  $ sudo usermod -aG docker <your-user>
  ```

  Note: Remember to log out and back in for this to take effect!
Build from Source Code - 2/ docker-compose and ruby install

$ sudo yum install git docker-compose ruby

Docker-compose and ruby are required by `docker-hadoop.sh`, which is a tool for creation of container-based cluster on a single machine.
Build from Source Code - source code downloading

Source code can be fetched from github.  

https://github.com/apache/bigtop

To use **Bigtop Official Release 1.3.0**, which is the first one with complete Arm Server support.

```
$ git clone https://github.com/apache/bigtop.git
$ cd bigtop/
$ git checkout -b working-rel/1.3.0 rel/1.3.0
```
Build from Source Code - start a build container

Here you should know that to ensure same experience and same setup for following builds, Bigtop supports being built from a container. (The container images itself is built from source too. But it's not this slide's focus.)

Before starting the container, give other users `w` access to `bigtop` home directory. It is required for gradle installation as 'jenkins' users. Otherwise, you will see this error when run 'gradlew tasks'. FAILED: Could not create service of type CrossBuildFileHashCache using BuildSessionScopeServices.createCrossBuildFileHashCache().

```
$ cd ..
$ chmod a+w bigtop/
$ cd bigtop
```

Now, you can start the container based on Image: "bigtop/slaves:1.3.0-centos-7-aarch64".

```
$ docker run -it --rm -u jenkins --workdir /ws -v `pwd`:/ws bigtop/slaves:1.3.0-centos-7-aarch64 bash -l
```

Note:
A. User 'jenkins' is employed. It exists by default in the root docker image of centos 7.0.
B. It's not allowed using 'root' to build bigtop. Some component refuses to be built in root.
C. Image "bigtop/slaves:1.3.0-centos-7-aarch64" will be retrieved from docker hub on live.
Once the container is created, you can enter it, and prepare the env. a little bit.

```bash
[jenkins@eb7597605841 ws]$ . /etc/profile.d/bigtop.sh
```

Note: `bigtop.sh` sets environment variables such as: JAVA_HOME, MAVEN_HOME, ANT_HOME, GRADLE_HOME, etc.

```
$ cd /ws
$ ./gradlew tasks
```

Note:
A. This will initiate gradle installation in current docker container.
B. Using `$ ./gradlew tasks | grep "^-rpm"` to see all supported components.
   - alluxio-rpm - Building RPM for alluxio artifacts
   - ambari-rpm - Building RPM for ambari artifacts
   - ... ...
   - zeppelin-rpm - Building RPM for zeppelin artifacts
   - zookeeper-rpm - Building RPM for zookeeper artifacts

Totally 32 components.
C. same thing to '-deb'.
Build from Source Code  - 6/  Build rpm, yum, and repo

This is the major part of building. Time consuming. And heavily rely on your network and local computing power.

$ ./gradlew allclean

$ ./gradlew rpm  
   Note: rpm - Build all RPM packages

$ ./gradlew yum  
   Note: yum - Creating YUM repository

$ ./gradlew repo  
   Note:
   ● repo - Invoking a native repository target yum
   ● It equals to 'createrpo ...'.
   ● This command creates ./repodata folder under [bigtop]/output. 'repodata' directory holds the metadata information for the newly created repository.

Note: upon completion, you can find the build result in '[bigtop]/output' folder.
Part II - Smoke Test
Here is what we will setup:

- on **ONE** physical machine
- **Multi-container** based cluster
- bigdata components deployed.

```
A multi-node hadoop cluster was created by bigtop:
$ cd provision/docker/
$ docker-hadoop.sh -C erp-18.66_dedian-9.yaml -C 5
```
Deploy & Smoke Test w/ Docker 1/ Yaml config file

Prelude: Bigtop uses docker as an easy way to deploy multi-node cluster and to do smoke tests. Here is how. To start with, you need a .yaml config file.

Bigtop provides sample config files. Find them in folder \\

```
[bigtop]/provisioner/docker/
```

Here is what I use:
```
$ cat [bigtop]/provisioner/docker/working-erp-18.06_centos-7.yaml
```

```yaml
docker:
  memory_limit: "16g"
  image: "bigtop/puppet:1.3.0-centos-7-aarch64"
  repo: "file://bigtop-home/output"
  distro: centos
  components: [hdfs, yarn, mapreduce, zookeeper, hbase, hive, spark]
  enable_local_repo: false
  smoke_test_components: [hdfs, spark]
```

Note: 'bigtop-home' don't change it. Find it in docker files.

Note: 'bigtop/puppet:1.3.0-centos-7-aarch64': it's in docker hub. And you can build your own's too.
With the yaml config file, we can start a container cluster, and deploy bigdata components at the same time.

```
$ cd provisioner/docker/
$ ./docker-hadoop.sh -C working-erp-18.06_centos-7.yaml -c 3
```

Note: -C: config file
Note: -c: number of containers

It takes about 10 minutes to finish. It doesn't download anything from external network.

Check status with "$ docker ps -n 3"
Deploy & Smoke Test w/ Docker 3/ smoke test

Edit config file (working-erp-18.06_centos-7.yaml) to set which components to smoke test.

Eg.

```
smoke_test_components: [hdfs,spark]
```

Then run the smoke test:

```
$ ./docker-hadoop.sh -C working-erp-18.06_centos-7.yaml -s
```

Note: -s: run smoke test.
Part III - Deploy on Multiple Nodes
Deploy on Multiple Nodes

- 1/ target

Step 1

Node 001
- hdfs
- yarn
- mapreduce
- zookeeper

Node 002
- hdfs
- yarn
- mapreduce
- kafka

Node 003
- hdfs
- yarn
- mapreduce
- flink

Site.yaml

Bigtop Release Repo

Step 2

Puppet apply
Set Hostname, Update /etc/hosts

Bigtop configuration requires FQDN for each machine in the cluster. Here we name our servers in the following rules:

```
node-<%03d>bigtop.deploy
```

To Set FQDN on each machine, do the following:

- Set FQDN for the machine
  ```
  $ sudo hostnamectl set-hostname node-001.bigtop.deploy
  $ sudo hostname
  ```

- Update /etc/hosts
  ```
  $ sudo vi /etc/hosts
  -append these lines:
    192.168.10.10 node-001.bigtop.deploy node-001
    192.168.10.11 node-002.bigtop.deploy node-002
    192.168.10.12 node-003.bigtop.deploy node-003
  ```

Note: run a ping test to verify hostname settings.
Deploy on Multiple Nodes - 3/ Disable firewall

Bigtop components (hdfs, yarn, etc.) use a lot of port for receiving services and connections between nodes. To make them work well, we need disable `firewalld` so all ports can be accessed through.

Method to disable firewall on CentOS 7, please refer: [https://linuxize.com/post/how-to-stop-and-disable-firewalld-on-centos-7/](https://linuxize.com/post/how-to-stop-and-disable-firewalld-on-centos-7/)

```bash
$ sudo systemctl stop firewalld
$ sudo systemctl disable firewalld
$ sudo systemctl mask --now firewalld
```

Note: (NOT RECOMMENDED) An alternative way is to open each and every port specifically. Difficulty of doing this is that it is not easy to list all ports completely.
Deploy on Multiple Nodes  - 4/ env. preparation

- Install openJDK 8 JAVA: [https://openjdk.java.net/install/](https://openjdk.java.net/install/)
  
  ```
  $ sudo yum -y install java-1.8.0-openjdk
  $ java -version
  
  openjdk version "1.8.0_191"
  ```

- Install unzip curl (it's required by gradle installation)
  
  ```
  $ sudo yum -y install unzip curl
  ```

Note: About puppet and puppet modules installation, refer to:

- Install puppet and puppetlabs-stdlib
  
  ```
  $ sudo yum updateinfo
  
  # BIGTOP-3088: pin puppetlabs-stdlib to 4.12.0 as the one provided
  # by distro (4.25.0) has conflict with puppet<4. Should be removed
  # once puppet in distro is updated.
  
  $ sudo yum -y install hostname curl sudo unzip wget puppet
  $ sudo puppet module install puppetlabs-stdlib --version 4.12.0
  ```

- Install puppet-modules (depends on puppet)
  
  ```
  $ sudo ./gradlew toolchain-puppetmodules
  
  BUILD SUCCESSFUL in 2m 18s
  ```
Deploy on Multiple Nodes - hiera and hieradata

Note: before doing that, you need to 'git clone' bigtop source code onto each of these nodes.

Copy hiera and hieradata to /etc for puppet

$ cd [your-bigtop-code-folder]
$ sudo cp bigtop-deploy/puppet/hiera.yaml /etc/puppet
$ sudo mkdir -p /etc/puppet/hieradata
$ sudo rsync -a --delete bigtop-deploy/puppet/hieradata/site.yaml bigtop-deploy/puppet/hieradata/bigtop
/etc/puppet/hieradata/
Deploy on Multiple Nodes

Edit /etc/puppet/hieradata/site.yaml

$ sudo vi /etc/puppet/hieradata/site.yaml

bigtop::hadoop_head_node: "node-001.bigtop.deploy"

hadoop::hadoop_storage_dirs:
  - /mnt/sda2
  - /mnt/sdd1
  - /mnt/sde1

hadoop_cluster_node::cluster_components:
  - hdfs
  - yarn
  - mapreduce
  - zookeeper
  - kafka
  - spark
  - flink

bigtop::bigtop_repo_uri:
"http://node-001.bigtop.deploy/releases/1.3.0/centos/2/aarch64"

Fields in 'site.yaml' are explained next.
Deploy on Multiple Nodes

- 7/site.yml - hadoop_storage_dirs

hadoop::hadoop_storage_dirs:

- These are folders (physical drives) which allocated to HDFS. You want to give more physical drives to HDFS as possible to increase I/O parallelism.
- One folder for one physical drive.
- More physical drive, more efficient parallel I/O.

If your disks are managed by LVM, please release them.

1. Release physical disks from LVM.
2. Format newly released physical drives and mount them.
Deploy on Multiple Nodes

- site.yaml
- bigtop_repo_uri

bigtop::bigtop_repo_uri: "http://node-001.bigtop.deploy/releases/1.3.0/centos/2/aarch64"

This is the URL where to retrieve Bigtop build artifacts. There are three major ways regarding where Bigtop artifacts can be retrieved.

A. Use Bigtop official release URL. Ref:
   https://www.apache.org/dyn/closer.lua/bigtop/bigtop-1.3.0/repos/
   a. Eg. for CentOS, download bigtop.repo and find 'baseurl=...'.

B. Create an offline Bigtop release repository by downloading all the bigtop repositories to local, using 'reposync' command.
   a. This is useful when you want to install bigtop when there is no internet access and you cannot build from source either.
   b. Details how, please check slide in backup section.

C. Build from source, then publish via Nginx. (This is not covered)

Further reading: How to 'Consume' a Bigtop release - to get a repository URL
Deploy on Multiple Nodes

hadoop_cluster_node:: **cluster_components**:

This is the list of which components to install on this node. Just component names are required, no need to care about roles. 'puppet apply' script will figure out proper roles.

- Note: each node should have its own components list.

bigtop:: **hadoop_head_node**:

Head node (aka. master node) is specified here. 'puppet apply' use this to decide which roles should be launched on which nodes.

- Note: all nodes should point to the same 'head node'. Bigtop will handle the difference in deployed roles.

Node 001
- hdfs
- yarn
- mapreduce
- zookeeper

Node 002
- hdfs
- yarn
- mapreduce
- kafka

Node 003
- hdfs
- yarn
- mapreduce
- flink
Deploy on Multiple Nodes

Edit /etc/puppet/hieradata/site.yaml

```
$ sudo vi /etc/puppet/hieradata/site.yaml

bigtop::hadoop_head_node: "node-001.bigtop.deploy"

hadoop::hadoop_storage_dirs:
  - /mnt/sda2
  - /mnt/sdd1
  - /mnt/sde1

hadoop_cluster_node::cluster_components:
  - hdfs
  - yarn
  - mapreduce
  - zookeeper
  - kafka
  - spark
  - flink

bigtop::bigtop_repo_uri:
  "http://node-001.bigtop.deploy/releases/1.3.0/centos/2/aarch64"
```
Deploy on Multiple Nodes

$ cd ~/bigtop/
$ sudo puppet apply -d --parser future  
--modulepath="bigtop-deploy/puppet/modules:/etc/puppet/modules"  
bigtop-deploy/puppet/manifests

Notice: Roles to deploy: [resourcemanager, nodemanager, mapred-app, hadoop-client,  
zookeeper-server, zookeeper-client, namenode, datanode]

... ...
Notice: Finished catalog run in 663.71 seconds

Note: To confirm the installation is correct:
$ sudo jps -l

to see running java applications. It should cover roles appeared above.
Deploy on Multiple Nodes

There are two ways to confirm that the newly deployed slave nodes are registered well on master.

- **HDFS**
  
  \$ hdfs dfsadmin -printToplogy

  
  Or,

  \$ hdfs dfsadmin -report

  Live datanodes (3):

  Note: the number of datanodes, and their details such as IP, Hostname:Port.

- **Yarn**

  \$ yarn node -list

  Total Nodes: 3

  Note: the number of nodes, and their details such as Hostname:Port.
Hands On
END