Build and Install Apache Ambari V2.6.1 on AArch64

Introduction

Apache Ambari project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.

Ambari enables System Administrators to:

- Provision a Hadoop Cluster
- Manage a Hadoop Cluster
- Monitor a Hadoop Cluster

On this collaborate page we try to explain how to build and install the Ambari on CentOS 7.4 and Debian 9.0

Build

Sources

- Upstream: https://github.com/apache/ambari.git
- Release-2.6.1 is used in this Wiki

Setup Environment

- Debian 9.0 64bit for AArch64, or CentOS-7.4 64bit for AArch64
- jdk8u-server-release-1804

Pre-Requisites

jdk8u-server-release-1804

Dependencies

maven@v3.5.3, nodejs@v4.2.6, npm@2.14.12, brunch@1.7.10, phantomjs@2.1.1, python>=2.6, python-dev, rpm, yum, g++

Build Steps

Install Pre-requisites

For Debian 9.0:

```bash
sudo apt install git python python-dev rpm yum build-essential libfreetype6 libfreetype6-dev fontconfig
fontconfig-config libfontconfig1-dev libssl-dev openssl findbugs -y
```
For CentOS7

```
sudo yum groupinstall "Development Tools"
sudo yum install git python-python-devel openssl-devel openssl openssl-libs freetype freetype-devel fontconfig-devel fontconfig gcc gcc-c++ make build autoconf automake cppunit-devel cmake bzip2 rpm-build
```

**Setup maven**

To setup maven 3.5.3

```
wget http://www-eu.apache.org/dist/maven/maven-3/3.5.3/binaries/apache-maven-3.5.3-bin.tar.gz
tar xvf apache-maven-3.5.3-bin.tar.gz
cd apache-maven-3.5.3/bin
export PATH=$PWD:$PATH
```

Make sure the version of Maven is 3.5.3 when the following command is issued.

```
mvn --version
```

```
Apache Maven 3.5.3 (3383c37e1f9e9b3bc3df5050c29c8aff9f295297; 2018-02-24T19:49:05Z)
Maven home: /home/centos/maven/apache-maven-3.5.3
Java version: 1.8.0-release, vendor: Oracle Corporation
Java home: /home/centos/jdk8u/jdk8u-server-release-1804/jre
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "4.12.0-1.1.aarch64", arch: "aarch64", family: "unix"
```

**Setup python tools**

- **For python 2.6, download**

  ```
  wget http://pypi.python.org/packages/2.6/s/setuptools/setuptools-0.6c11-py2.6.egg#md5=bfa92100bd772d5a213eedd356d64086
  sudo sh setuptools-0.6c11-py2.6.egg
  ```

- **For python 2.7, download**

  ```
  wget https://pypi.python.org/packages/2.7/s/setuptools/setuptools-0.6c11-py2.7.egg#md5=fef9f97bc72225116870bc7919059ea
  sudo sh setuptools-0.6c11-py2.7.egg
  ```

  The python 2.6 didn't work for me, hence I have just created a softlink of v2.6 for v2.7 python

  ```
  $ sudo ln -s /usr/bin/python2.7 /usr/bin/python2.6
  ```

**Setup nodejs/npm**

Nodejs and npm come with different versions along with Ubuntu/Debian

For Ubuntu/Debian, nodejs/npm can be installed by:
sudo apt-get install -y nodejs npm
cd /usr/bin && sudo ln -s nodejs node
sudo npm install -g brunch@1.7.10

Note that if you are using Debian 9 stretch then please follow the below steps https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions
sudo apt-get install curl
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install -y nodejs
cd /usr/bin && sudo ln -s nodejs node
sudo npm install -g brunch@1.7.10

For CentOS7, nodejs/npm need to be built from source.

git clone https://github.com/nodejs/node.git
cd node
git checkout -b 4.2.6 v4.2.6
./configure --prefix=/usr && make -j8
sudo make install
sudo npm install -g brunch@1.7.10

The version of built out binaries are: nodejs@v4.2.6, npm@2.14.12.

As long as they are installed, pom.xml in ambari-admin needs to be changed to reflect these versions. The target nodejs/npm version are defined in "configuration" field of "frontend-maven-plugin".

Build PhantomJS

The following steps explain about AArch64 supported phantomjs v2.1.1. Note that you have to install all the dependency packages before you proceed further. Refer collaborate page for PhantomJS

git clone https://github.com/ariya/phantomjs.git
cd phantomjs
git checkout -b v2.1.1 2.1.1
./build.py -c -j $(getconf _NPROCESSORS_ONLN)

When the build is finished, create tar file for deployment

cd deploy
./package.sh

You can test phantomjs build by issuing:

./bin/phantomjs test/run-tests.js

Install phantomjs-2.1.1-linux-aarch64.tar.bz2 to the system and add phantomjs to $PATH. Check if phantomjs is properly installed by doing:

ten-user@ambari-dk:$ phantomjs --version
2.1.1

Replace frontend-maven-plugin

Ambari uses fronend-maven-plugin@v0.0.16, which doesn't support AArch64. Do following to rebuild this plugin for AArch64.

git clone https://github.com/eirslett/frontend-maven-plugin.git
cd frontend-maven-plugin
git checkout -b 0.0.16 frontend-plugins-0.0.16
git apply frontend-maven.patch
mvn clean -DskipTests install
Replace leveldbjni

leveldbjni is used in Ambari-metrics. It only provides x86/x86_64 version in maven repo. So AArch64 version of leveldbjni needs to be built and installed.

```
wget http://pkgs.fedoraproject.org/repo/pkgs/snappy/snappy-1.0.5.tar.gz/4c0af044e654f5983f4a6bf00d1ac236/snappy-1.0.5.tar.gz
    tar -xf snappy-1.0.5.tar.gz
    cd snappy-1.0.5
     ./configure --disable-shared --with-pic --host aarch64-unknown-linux --build aarch64-unknown-linux
    make -j4
    cd ..
    git clone git://github.com/chirino/leveldb.git
    git clone git://github.com/fusesource/leveldbjni.git
    export SNAPPY_HOME=`cd snappy-1.0.5; pwd`
    export LEVELDB_HOME=`cd leveldb; pwd`
    export LEVELDBJNI_HOME=`cd leveldbjni; pwd`
    cd leveldb
    export LIBRARY_PATH=${SNAPPY_HOME}
    export C_INCLUDE_PATH=${LIBRARY_PATH}
    export CPLUS_INCLUDE_PATH=${LIBRARY_PATH}
    git apply ../leveldbjni/leveldb.patch
    wget https://raw.githubusercontent.com/google/leveldb/master/port/atomic_pointer.h -O port/atomic_pointer.h
    make libleveldb.a
    cd ${LEVELDBJNI_HOME}
    git checkout -b 1.8 leveldbjni-1.8
    mvn clean install -P all -P linux64 -DskipTests=true
```

Build Ambari

To build Ambari, a certain version number should be provided. This version number IS 5-DIGITS, not "4-digits" mentioned on Ambari’s Wiki Page. The last digit may vary but the first 3 digits should be same as Ambari source/release version. In our case this is 2.6.1. Patch is provided to make Ambari built on AArch64. Apply all the patches before you are going for the build. You can directly clone and build my AMBARI git repository - https://git.linaro.org/people/naresh.bhat/apache/ambari.git

```
git clone https://github.com/apache/ambari.git
    cd ambari
    git checkout release-2.6.1

Download and apply following patches

    git am 0001-ambari-build-aarch64-2.6.1.patch
    git am 0002-ambari-metrics-grafana-Add-jdeb-support.patch
    git am 0003-ambari-funtest-Add-jdeb-support.patch
    git am 0004-ambari-logsearch-Add-jdeb-support.patch
    git am 0005-ambari-Add-jdeb-arm64-support.patch

    mvn versions:set -DnewVersion=2.6.1.0.0
    pushd ambari-metrics
    mvn versions:set -DnewVersion=2.6.1.0.0
    popd

On CentOS 7.4 to generate rpm's you can issue below command.

    mvn -B clean install package rpm:rpm -DskipTests -Dpython.ver="python >= 2.6" -Preplaceurl -Drat.ignoreErrors=true

On Debian 9 to generate Debian packages you can issue below command.

    mvn -B clean install jdeb:jdeb -DnewVersion=2.6.1.0.0 -DskipTests -Dpython.ver="python >= 2.6" -Drat.ignoreErrors=true
```

Ambari Server will create following packages

- RPM will be created under AMBARI_DIR/ambari-server/target/rpm/ambari-server/RPMS/aarch64.

Ambari Agent will create following packages

- RPM will be created under AMBARI_DIR/ambari-agent/target/rpm/ambari-agent/RPMS/aarch64.
Ambari Metrics will create following packages

- RPM will be created under `AMBARI_DIR/ambari-metrics/ambari-metrics-timelineservice/target/rpm/ambari-metrics-collector/RPMS/noarch`.

### Patches

- 0004-ambari-log...b-support.patch
- 0003-ambari-fun...b-support.patch
- 0002-ambari-m...-support.patch
- 0005-ambari-Ad...-support.patch
- 0001-ambari-bui...h64-2.6.1.patch
Running Ambari

Run Ambari Server

First, install Pre-Requisites

```bash
sudo yum install postgresql
sudo yum install postgresql-server
```

Then install the Ambari Server RPM.

```bash
sudo yum install ambari-server/target/rpm/ambari-server/RPMS/aarch64/ambari-server-*-rpm
```

Initialize Ambari Server:

```bash
sudo ambari-server setup
```

Start up Ambari Server:

```bash
sudo ambari-server start
```

To access Ambari, go to: `http://{ambari-server-hostname}:8080`

The initial username/password is admin/admin.

Run Ambari Agent

Install the Ambari Agent RPM.

```bash
sudo yum install ambari-agent/target/rpm/ambari-agent/RPMS/aarch64/ambari-agent-2.4.2.0-0.aarch64.rpm
```

Then edit the location of Ambari Server in `/etc/ambari-agent/conf/ambari-agent.ini` by editing the `hostname` line.

Start Ambari Agent:

```bash
sudo ambari-agent start
```