https://www.office.com/  
  
root@6f44fbd6f5cb:/# export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-arm64

root@6f44fbd6f5cb:/# export PATH=$PATH:$JAVA\_HOME/bin

root@6f44fbd6f5cb:/# java -version

export PATH=$JAVA\_HOME/bin:$PATH

to resolve this first download android-sdk with apt-get install android-sdk  
then find the location of sdk with whereis android-sdk command   
then set path as   
export ANDROID\_HOME=/drivea/java/android/android-sdk  
export PATH=$PATH:$ANDROID\_HOME/tools/bin  
export PATH=$PATH:$ANDROID\_HOME/platform-tools Note : set path acc to ur sdk location

./gradlew assemble -x signArchives

gradle build -x signArchives

export JAVA\_TOOL\_OPTIONS=-Dfile.encoding=UTF8

mohneesh\_ubuntu\_maven

wget [www.scala-lang.org/files/archive/scala-2.13.0.deb](http://www.scala-lang.org/files/archive/scala-2.13.0.deb)

dpkg -i scala-version.deb

mvn install -Dgpg.skip=true

sudo apt install python3.7 python3-venv python3.7-venv

python3.7 -m venv py37-venv

. py37-venv/bin/activate

docker run -it --name prakriti\_python ubuntu:latest

docker build . -t abhi\_image -f Dockerfiles/Dockerfile-awskops

docker image inspect protokube:1.19.0-beta.2 | grep "Architecture"

docker buildx build --push --platform linux/arm64,linux/amd64 -t logspout:${CIRCLE\_BRANCH} -t logspout:latest -f <filename> .

15m 21s

DOCKER\_USERNAME

DOCKER\_PASSWORD

pip3 install -r requirements.txt

docker run -it --rm <image name >

docker image inspect <image\_name>

root@094f8e1844:/nishant/liferay-portal# file /nishant/liferay-portal/build/node/bin/node

/nishant/liferay-portal/build/node/bin/node: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.18, BuildID[sha1]=1337e280fe687c3d526558f3f3289f735cda4834, with debug\_info, not stripped

root@06094f8e1844:/nishant/liferay-portal#

grep -inr "architecture"

grep -inr "arm64"

grep -inr "x86\_64"

grep -inr "amd64"

2>&1 | tee outfile.txt

apt-get install openjdk-8-jdk

mvn clean install -DskipTests

grep -inr getObjectMapper | grep -v new

wget http://mirrors.estointernet.in/apache//ant/binaries/apache-ant-1.9.15-bin.zip

unzip apache-ant-1.9.14-bin.zip

mv apache-ant-1.9.14 /opt/

ln -s /opt/apache-ant-1.9.15/ /opt/ant

ln -s /opt/ant/bin/ant /usr/bin/ant

export ANT\_HOME=/opt/ant

cd /opt/apache-ant-1.9.14/

ant -f fetch.xml -Ddest=system

git clone --branch sync-3.1.0-ga1 --depth=1 <https://github.com/liferay/liferay-portal>

sbt makePom

wget www.scala-lang.org/files/archive/scala-2.13.0.deb

dpkg -i scala\*.deb

echo "deb https://dl.bintray.com/sbt/debian /" | sudo tee -a /etc/apt/sources.list.d/sbt.list

apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 2EE0EA64E40A89B84B2DF73499E82A75642AC823

apt-get update

apt-get install sbt -y

docker cp <containerId>:/file/path/within/container /host/path/target

mvn -Dmaven.javadoc.skip=true verify

Hi All,

If you want to keep some packages on build for longer time and it is affecting due to vpn disconnection. Please use screen utility.

I have used it and the compilation does not get affected if vpn disconnects in between.

Steps to use screen utility:

1) Install screen utility in your docker or server(wherever you are working).

2) Create a session by command: screen -S session\_name

3)Put your package for compilation.

4) Detach your screen by command: ctrl+a +d

5) list your screen sessions by command: screen -ls

6)Reattach your screen by command: screen -r screen\_session\_id

For more details please refer the following link:

https://linuxize.com/post/how-to-use-linux-screen/

How To Use Linux Screen

https://linuxize.com

https://vertx.io/docs/apidocs/io/vertx/core/eventbus/EventBus.html#send-java.lang.String-java.lang.Object-io.vertx.core.eventbus.DeliveryOptions-io.vertx.core.Handler-

<https://docs.travis-ci.com/user/docker/>

export PATH=$PATH:$HOME/.cargo/bin

wget https://services.gradle.org/distributions/gradle-5.0-bin.zip

unzip -d /opt/gradle gradle-5.0-bin.zip

export GRADLE\_HOME=/opt/gradle/gradle-5.0

export GRADLE\_HOME=/opt/gradle/gradle-5.0

export PATH=${GRADLE\_HOME}/bin:${PATH}

gradle -v

mv gradle-5.0-bin.zip gradle

mv gradle /opt/

103.206.163.18:4433

Abhishek.nishant

abhishek138

LocalDomain

- if [ "${TRAVIS\_CPU\_ARCH}" == "arm64" ]; then

mvn clean install >> 2>&1 | tee rahul.txt;

cd ./modules/openapi-generator-cli;

cd ../../;

docker build -t rahulimage -f ./modules/openapi-generator-cli/Dockerfile .;

docker images;

else

mvn -e --no-snapshot-updates --quiet --batch-mode --show-version clean install -Dorg.slf4j.simpleLogger.defaultLogLevel=error;

mvn -e --no-snapshot-updates --quiet --batch-mode --show-version verify -Psamples -Dorg.slf4j.simpleLogger.defaultLogLevel=error;

fi

--- a/modules/openapi-generator-cli/Dockerfile

+++ b/modules/openapi-generator-cli/Dockerfile

@@ -1,4 +1,4 @@

-FROM java:8-jre-alpine

+FROM openjdk:8-jre-alpine

https://github.com/dappnode/compose/releases/tag/1.25.5

jar -xvf =🡺 to extraxt jar file

jar cf jar-file input-file ==🡺 to create a jar file

git hub: kulabh

password: abhishek138

mkdir -p $HOME/dotnet && tar zxf dotnet-sdk-3.1.406-linux-arm64.tar.gz -C $HOME/dotnet

export DOTNET\_ROOT=$HOME/dotnet

export PATH=$PATH:$HOME/dotnet

slack

id : odidev

passward : nibble@123

<https://betterprogramming.pub/install-go-1-11-on-ubuntu-18-04-16-04-lts-8c098c503c5f>

docker build . -t abhi\_image -f Dockerfiles/Dockerfile-awskops

2.86

ip 192.168.3.132

login: ubuntu\_arm

pass: root@123

3.169

ip 192.168.3.137

login: arm137

pass: root@123

2.118

ip 192.168.2.139

login: arm139

pass: root@123

serial port detail

0=2.137

1=2.132

2=2.38

3=2.139

COPY / /XYZ

COPY <binary> /usr/local/bin

x64\_server@192.168.2.172

prabhat181998  
abhishek138

premprabhat

abhishek138

apt-get -y --no-install-recommends install iproute2 libavcodec58 firefox=89.0+build2-0ubuntu0.20.04.2

Git hub action build and push link  
  
https://github.com/odidev/dcind/commit/b07191399108c398e0a56cdb49761a56d618b3ad

Hi Team,

I am trying to use the**iron/node** image on the arm64 platform but it seems it is not available for arm64.

I have successfully built the image using the command **docker build -t image\_name .** on the arm64 platform without making any changes in the Dockerfile.

I have used **Travis-CI**to build and push the image for both the platforms.

**Commit Link** - https://github.com/odidev/dockers/commit/f886f6825d1a91c85ecec605d29fe78253d31776

**Travis-CI link** - <https://travis-ci.com/github/odidev/dockers/builds/231921421>

**Docker Hub Link** - <https://hub.docker.com/repository/registry-1.docker.io/odidev/node_image/tags?page=1&ordering=last_updated>

Do you have any plans on releasing arm64 images on [dockerhub](https://www.docker.com/products/docker-hub" \t "_blank).

It will be very helpful if an arm64 image is available. If interested, I will raise a PR.

Please refer the issue [#77](https://github.com/iron-io/dockers/issues/77) in the github repository aswell for the same.

ghp\_Felde1MT8lYI49pYEfRB6sItqMtUoM19ZzlO

. 2> /dev/null;

https://docs.docker.com/engine/install/ubuntu/#install-using-the-convenience-script