Quarkus - Building a Custom SubstrateVM

Prerequisites

Install JVMCI

Before anything can be done you must first install a JVMCI build.

These can be downloaded from OTN here

Note, if you are installing on an alternative system, you can use the developer tools on chrome or firefox to capture a direct URL with an auth param.

After obtaining the download install it and set JAVA_HOME

```
wget -O jvmci.tgz
http://download.oracle.com/otn/utilities_drivers/oracle-
labs/labsjdk-8u202-jvmci-0.56-darwin-
amd64.tar.gz?AuthParam=[GENERATED AUTH TOKEN HERE]
tar -xzvf jvmci.tgz -C /opt
# On Mac do labsjdk1.8.0_202-jvmci-0.56/Contents/Home
export JAVA_HOME=/opt/labsjdk1.8.0_202-jvmci-0.46/
export PATH=$JAVA_HOME/bin:$PATH
```

NOTE
Get the latest version available as graal tends to keep updating the minimum requirements.

Install MX

Now you need to install Graal's special build tool, mx.

```
git clone https://github.com/graalvm/mx.git
export PATH=`pwd`/mx:$PATH
```

Checkout and build SubstrateVM

You can now check-out and build Substrate:

```
git clone https://github.com/oracle/graal.git
cd graal/substratevm
mx build
```

Once built, you can quickly run the tools in place

```
echo "public class HelloWorld { public static void main(String[]
args) { System.out.println(\"Hello World\"); } }" > HelloWorld.java
javac HelloWorld.java
mx native-image HelloWorld
./helloworld
```

Building a custom GraalVM distribution

In order to create a custom GraalVM distribution (necessary for usage in quarkus) you must switch to the vm subdirectory, and use special build options to pick the modules in your distribution.

For a simple subset VM with just substrate and its native tools execute the following:

```
cd ../vm
mx --dy /substratevm,/tools build
tar -czvf my-custom-graal.tgz -C latest_graalvm .
```

For a stock CE build the following will do:

```
mx --dy /substratevm,/tools,/sulong,/graal-nodejs build
tar -czvf my-custom-graal.tgz -C latest_graalvm .
```