Quarkus - Container Images

Quarkus provides extensions for building (and pushing) container images. Currently it supports:

- jib
- docker
- s2i

Container Image extensions

JIB

The extension `quarkus-container-image-jib` is powered by Jib for performing container builds. The major benefit of using Jib with Quarkus, is that all dependencies (everything found under target/lib) are cached in a different layer than the actual application making rebuilds extra fast and extra small (when it comes to pushing). Another important benefit of using this extension is that it provides the ability to create a container image without having to have any dedicated client side tooling (like Docker) or running daemon processes (like the Docker daemon) when all that is needed is the ability to push to a container image registry.

To use this image extension, you just need to add the following dependency into your project.

```xml
    <dependency>
      <groupId>io.quarkus</groupId>
      <artifactId>quarkus-container-image-jib</artifactId>
    </dependency>
```

Docker

The extension `quarkus-container-image-docker` is using the docker binary and the generated Dockerfiles under `src/main/docker` in order to perform docker builds.

To use this image extension, you just need to add the following dependency into your project.

```xml
    <dependency>
      <groupId>io.quarkus</groupId>
      <artifactId>quarkus-container-image-docker</artifactId>
    </dependency>
```

S2i

The extension quarkus-container-image-s2i is using s2i binary builds in order to perform container builds inside the Openshift cluster. The idea behind the binary build is that you just upload the the artifact and its dependencies to the cluster and during the build they will be merged to a builder image (defaults to fabric8/s2i-java).

The benefit of this approach, is that it can be combined with Openshift's DeploymentConfig that makes it easy to rollout changes to the cluster.

To use this image extension, you just need to add the following dependency into your project.

```
    <dependency>
      <groupId>io.quarkus</groupId>
      <artifactId>quarkus-container-image-s2i</artifactId>
    </dependency>
```

S2i builds, require creating a BuildConfig and two ImageStream resources, one of the builder image and one for the output image. The creation of such objects is being taken care of by the Quarkus kubernetes extension.

Building

To build a container image for your project, you just need to specify quarkus.container-image.build=true either to the application.properties or as a system property.

```
mvn clean package -Dquarkus.container-image.build=true
```

Setting the environment variable QUARKUS_CONTAINER_IMAGE_EXECUTION to build can be used instead of the system propery.

Pushing

If the execution is set to push then after the build, the image will also be pushed to an image registry. The registry can be specified using quarkus.container-image.registry and will default to docker.io.

Customizing

For customizing the extension the following properties are available:

Table 1. Container Image

Property	Type	Description	Default Value

quarkus.container-image.group	String	The group/repository of the image	The ${user.name}
quarkus.container-image.name	String	The name of the image	The application name
quarkus.container-image.tag	String	The tag of the image	The application version
quarkus.container-image.registry	String	The registry to use for pushing	
quarkus.container-image.username	String	The registry username	
quarkus.container-image.password	String	The registry password	
quarkus.container-image.insecure	Boolean	Flag to allow insecure registries	false
quarkus.container-image.build	Boolean	Flag to enable building of a container image	false
quarkus.container-image.push	Boolean	Flag to enable pushing of a container image to the configured registry	false

Jib Options

On top the the global container image options, the following jib specific options are available:

Property	Type	Description	Default Value
quarkus.container-image-jib.base-jvm-image	String	The base image to use for the jib build	fabric8/java-alpine-openjdk8-jre
quarkus.container-image-jib.base-native-image	String	The base image to use for the native build	registry.access.redhat.com/ubi8/ubi-minimal
quarkus.container-image-jib.jvm-arguments	String	The arguments to pass to java	-Dquarkus.http.host=0.0.0.0,-Djava.util.logging.manager=org.jboss.logmanager.LogManager
quarkus.container-image-jib.native-arguments	String	The arguments to pass to the native application	-Dquarkus.http.host=0.0.0.0

quarkus.container-image-jib.environment-variables	Map<String, String>	The container environment variables	

Docker Options

On top the the global container image options, the following docker specific options are available:

Property	Type	Description	Default Value
quarkus.container-image-docker.dockerfile-jvm-path	String	Path to the JVM Dockerfile	${project.root}/src/main/docker/Dockerfile.jvm
quarkus.container-image-docker.dockerfile-native-path	String	Path to the native Dockerfile	${project.root}/src/main/docker/Dockerfile.native

S2i Options

On top the the global container image options, the following s2i specific options are available:

Property	Type	Description	Default Value
quarkus.container-image-s2i.base-jvm-image	String	The base image to use for the s2i build	fabric8/java-alpine-openjdk8-jre
quarkus.container-image-s2i.base-native-image	String	The base image to use for the native build	registry.access.redhat.com/ubi8/ubi-minimal