## 6. 基础配置(maven必须3.3版本以上)

### 6.1. fabric8:setup

①通过命令添加fabric8依赖

mvn io.fabric8:fabric8-maven-plugin:3.1.92:setup

②手动加依赖

<plugin>

<groupId>io.fabric8</groupId>

<artifactId>fabric8-maven-plugin</artifactId>

<version>3.1.92</version>

<configuration>

....

<images>

*<!-- A single's image configuration -->*

<image>

...

<build>

....

</build>

</image>

....

</images>

</configuration>

*<!-- Connect fabric8:resource and fabric8:build to lifecycle phases -->*

<executions>

<execution>

<id>fabric8</id>

<goals>

<goal>resource</goal>

<goal>build</goal>

<goal>helm</goal>

</goals>

</execution>

</executions>

</plugin>

|  |  |
| --- | --- |
|  | This plugin definition includes the configuration to generate the Docker image. |

The setup can be influenced with the following configuration options:

Table 20. Setup configuration

| **Element** | **Description** | **Property** |
| --- | --- | --- |
| **updateVersion** | If set to true then an already existing plugin configuration will be updated. Otherwise an existing configuration is left untouched. Default is true. | updateVersion |
| **useVersionProperty** | Whether we should use a version property for the plugin which is defined in a dedicates <properties> section with the name fabric8.maven.plugin.version | useVersionProperty |
| **generateBackupPoms** | Controls whether a backup pom should be created when the pom.xml is modified. Ddefault is true. | generateBackupPoms |
| **backupPomFileName** | Name of the backup file to create. Default is ${basedir}/pom.xml-backup | backupPomFileName |

### 6.2.开启本地kubernetes测试环境

#fabric8:cluster-start

#### 6.2.1. 前提条件

Maven3.3以上版本

依赖驱动:

* Windows users 参考 [enable Hyper-V on Windows 10](https://msdn.microsoft.com/en-us/virtualization/hyperv_on_windows/quick_start/walkthrough_install) or [Windows 7](https://blogs.technet.microsoft.com/schadinio/2010/07/09/installing-hyper-v-manager-on-windows-7/).
* OS X 参考 [install the xhyve driver](https://github.com/kubernetes/minikube/blob/master/DRIVERS.md" \l "xhyve-driver)
* Linux 参考 [install the kvm driver](https://github.com/kubernetes/minikube/blob/master/DRIVERS.md" \l "kvm-driver)

#### 6.2.2. Starting the cluster

mvn fabric8:cluster-start

This will internally invoke the **[fabric8:install](https://maven.fabric8.io/" \l "fabric8:install)** goal to ensure that all the required binaries are installed (like [gofabric8](https://github.com/fabric8io/gofabric8/) and for kubernetes: [kubectl](http://kubernetes.io/docs/user-guide/kubectl-overview/) and [minikube](https://github.com/kubernetes/minikube) or for OpenShift: [oc](https://docs.openshift.com/enterprise/latest/cli_reference/basic_cli_operations.html) and [minishift](https://github.com/jimmidyson/minishift))

By default the binaries are installed in ~/.fabric8/bin

#### 6.2.3. Using OpenShift

By default fabric8:cluster-start will use minikube to create a local single node kubernetes cluster. To specify OpenShift use:

mvn fabric8:cluster-start -Dfabric8.cluster.kind=openshift

This will then use minishift instead to create a single node local OpenShift cluster.

#### 6.2.4. VM drivers

By default the VM drivers used will be hyperv on Windows, xhyve on OS X and kvm on Linux.

If you wish to switch to a different VM driver you can specify the fabric8.vm.driver property. For example if you have installed [VirtualBox](https://www.virtualbox.org/wiki/Downloads) and wish to use that then type:

mvn fabric8:cluster-start -Pfmp-snapshot -Dfabric8.vm.driver=virtualbox

Note that we highly recommend using the default VM drivers (hyperv on Windows, xhyve on OS X and kvm on Linux) as they tend to work better and use less resources on your laptop than the alternatives.

#### 6.2.5. Configure apps

By default the cluster contains only the [fabric8 developer console](http://fabric8.io/guide/console.html) as often developers laptops don’t have lots of RAM.

If you want to deploy the full fabric8 platform (with Nexus, Jenkins, Gogs, JBoss Forge etc) then use the following command:

mvn fabric8:cluster-start -Dfabric8.cluster.app=platform

#### 6.2.6. Configure cluster resources

You can specify the number of CPUs or memory via additional parameters:

mvn fabric8:cluster-start -Dfabric8.cluster.cpus=2 -Dfabric8.cluster.memory=4096

The above configures 2 CPUs and 4Gb of memory

#### 6.2.7. Stop

You can stop the cluster at any time via **[fabric8:cluster-stop](https://maven.fabric8.io/" \l "fabric8:cluster-stop)**

mvn fabric8:cluster-stop

Once stopped you can restart again with all the images, resources and pods intact later on by running fabric8:cluster-start again

mvn fabric8:cluster-start

### 6.3. fabric8:cluster-stop

This goal will stop a local kubernetes cluster.

This goal stops the VM running the local cluster so it will free up resources on your machine (memory + CPU) though the VM is not destroyed; it can restarted.

mvn fabric8:cluster-stop

#### 6.3.1. Restarting

You can restart the cluster at any time via **[fabric8:cluster-start](https://maven.fabric8.io/" \l "fabric8:cluster-start)**

mvn fabric8:cluster-start

Once restarted all the images, resources and pods should come back

#### 6.3.2. Deleting

If you wish to destroy the cluster VM and all the data inside it then you can pass the fabric8.cluster.deleteparameter with a value of true:

mvn fabric8:cluster-stop -Dfabric8.cluster.delete=true

### 6.4. fabric8:install

Ensures that the fabric8 binaries are installed on the current machine such as [gofabric8](https://github.com/fabric8io/gofabric8/) and for kubernetes: [kubectl](http://kubernetes.io/docs/user-guide/kubectl-overview/) and [minikube](https://github.com/kubernetes/minikube) or for OpenShift: [oc](https://docs.openshift.com/enterprise/latest/cli_reference/basic_cli_operations.html) and [minishift](https://github.com/jimmidyson/minishift)

mvn fabric8:install

An alternative is to just run the **[fabric8:cluster-start](https://maven.fabric8.io/" \l "fabric8:cluster-start)** goal to install the binaries and start a local cluster

By default the binaries are installed in ~/.fabric8/bin