Title: Unknown lifecycle phase 'clean install test -Dsurefire.suiteXmlFiles=testng.xml' while running the testng xml from docker file
Post Body:

I'm running my testng xml from docker file. When I tried docker run, it throws 'You must specify a valid lifecycle phase or a goal in the format'

The same command is executing in the CLI mode. Below is my docker file

```
FROM maven:3.6.0-jdk-8 ADD pom.xml pom.xml RUN mvn compile ADD . . CMD ['mvn','clean install test -Dsurefire.suiteXmlFiles=tes
```

Accepted Answer: None
Highest Rated Answer:

your

```
CMD ['mvn','clean install test -Dsurefire.suiteXmlFiles=testng.xml']
```

is the equivalent of

```
mvn 'clean install test -Dsurefire.suiteXmlFiles=testng.xml'
```

outside the container. It other words the whole `clean install test -Dsurefire.suiteXmlFiles=testng.xml` is treated as maven phase and of course it can not be found.

What you are trying to do can be correctly written like this:

```
CMD ['mvn','clean', 'install', 'test', '-Dsurefire.suiteXmlFiles=testng.xml']
```

That said, there are few other issues with your approach:

- You run `mvn compile` after just adding the POM (before adding the working dir). Depending on what is in the POM that may do nothing or fail.
- There is no need to call `install` in a container. It will add the resulting artifact in the local (to the container) repo which will be destroyed together with the container. Unless there is something else in the container that expects to find your artifact in the local repo this is pointless. Normally `verify` or `package` (if there are no integration tests) is more than enough
- you run `install` first and `test` later. However both are part of the same build lifecycle. So what will happen is, all phases (including `test`) up to `install` will be executed. Then all phases up to `test` will be executed again. That is probably not what you want