CI/CD Implementation

Jenkins is hosted on my local machine.

**Global Tool Configuration**

Ensure that the following software’s are mapped to their respective PATH on the machine or provide an installer for the following:

* JAVA
* Maven
* Git
* Docker
* Maven Configuration

Ensure that if you use existing settings.xml (leave as-is) else a jenkins specific settings.xml mention the path here.

Ensure that you add the following piece into the settings.xml file

<server>

<id>jenkins</id>

<privateKey>~/.ssh</privateKey>

<passphrase></passphrase>

</server>

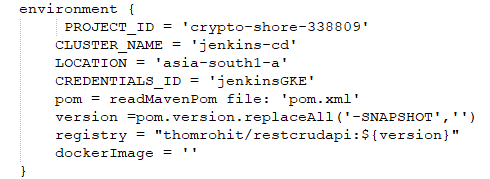
This ensures that a maven release can occur correctly. Which will update the pom.xml to the next version and also create a tag branch with the code.

**Necessary Plugins**

Ensure that you have the following plugins Google Kubernetes Engine Plugin, pipeline-utility-steps, Docker, Kuberenetes apart from the basic plugins needed in Jenkins.

**Pipeline**

The Declarative Pipeline consists of environment ,triggers, options, stages and post blocks



The Environment block has a set of global variable defined to be used throughout the pipeline, notice that we **read the pom.xml** here to get the version of the application so that we can use this to build the Docker image and update the Kuberentes yaml file accordingly.

This helps update an existing image and also creates a new image when a release is done.

**Triggers block**

A poll (10 mins) gets triggered when a commit has happened

**Options block**

Disable concurrent build to occur at the same time

Discard older builds and keep only last 14

**Stages block**

The stages is spilt into different stage, each stage performs the task assigned and moves to the next stage accordingly.

The Docker build Stage has a section commented out, as I was facing an issue which needed a Script approval, and if that is not enabled in your Jenkins, then the pipeline would fail. But that seems to be the recommended way which is why I left it.

The alternative was to run the docker commands directly with the help of the credentials plugin.

**Post block**

Finally we have the post block, which will always send out an email with the logs attached and the link to the Job build URL.