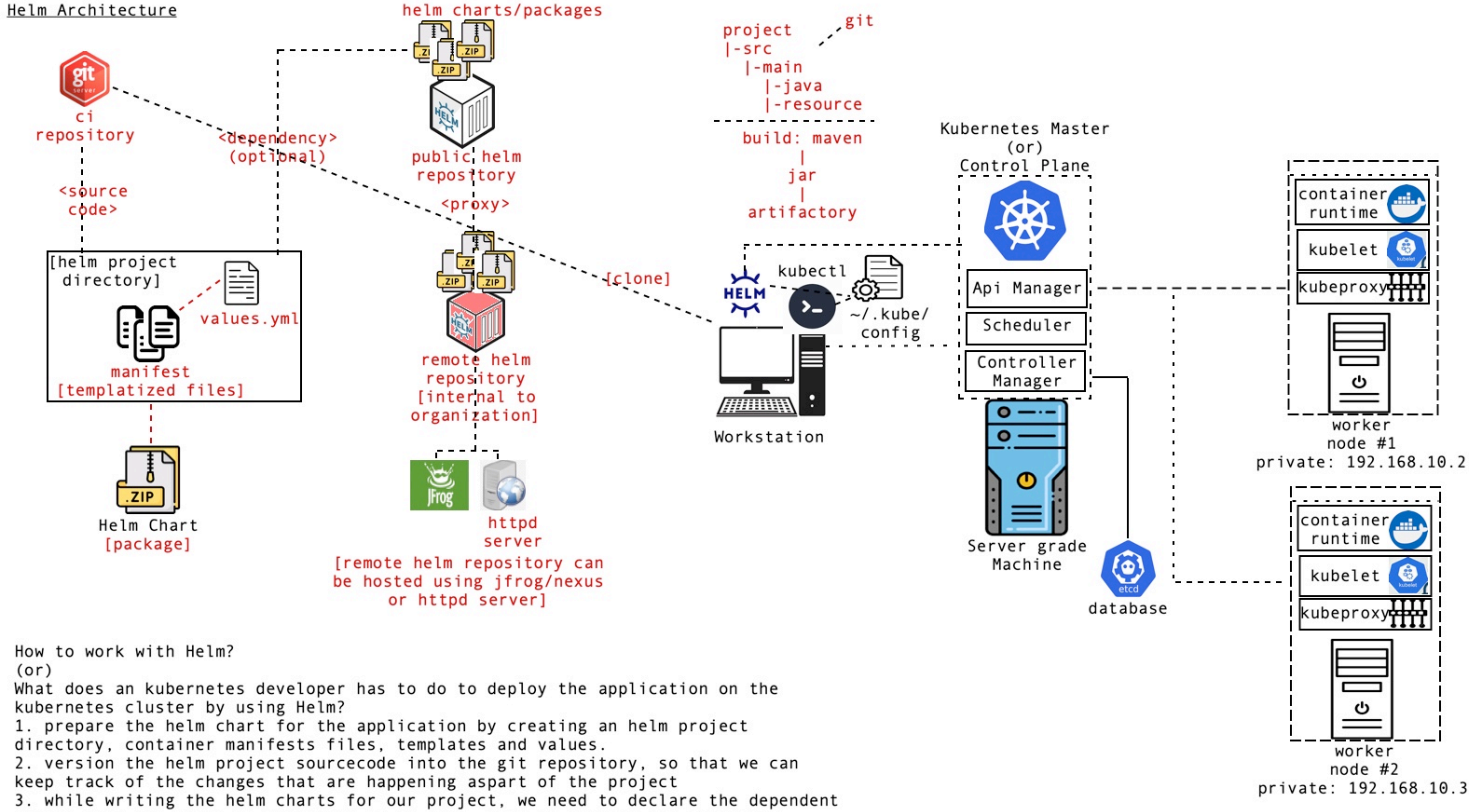
Helm Architecture



How to work with Helm?
(or)
What does an kubernetes developer has to do to deploy the application on the kubernetes cluster by using Helm?

1. prepare the helm chart for the application by creating an helm project directory, container manifests files, templates and values.
2. version the helm project sourcecode into the git repository, so that we can keep track of the changes that are happening aspart of the project
3. while writing the helm charts for our project, we need to declare the dependent charts that our chart is dependent on as well. When we specify the dependent charts for our chart, helm takes care of identifying and downloading the dependent charts from the public/organization repository while deploying the objects
4. for different envs we want to deploy the application with different values, so we need to write values.yml overriding the default values that are written aspart of the chart, so that the charts can be deployed across different env
5. run the helm cli by passing the helm chartname and values.yml as an input file, so that the helm cli
   1. validates the template files that are written
   2. substitutes the template/token values with the values that we have supplied aspart of values.yml

and deploys the objects on the cluster.

The devops engineer dont need to have the knowledge of the chart being deployed, and the dependencies, all that he has to do is pass the chart to helm cli asking to deploy on the cluster.

How does helm deploys the objects onto the Cluster?

1. HELM CLI downloads the charts and the dependent charts from the HELM repository
2. extracts the charts zip file and replaces configuration/templatized tokens in the manifest files with the values.yml values we supplied.
3. validates the manifests that are generated
4. determines the dependency order of the objects or resources we defined in those manifests
5. applies/creates those objects onto the cluster

This entire proccess is called an “Helm Release”