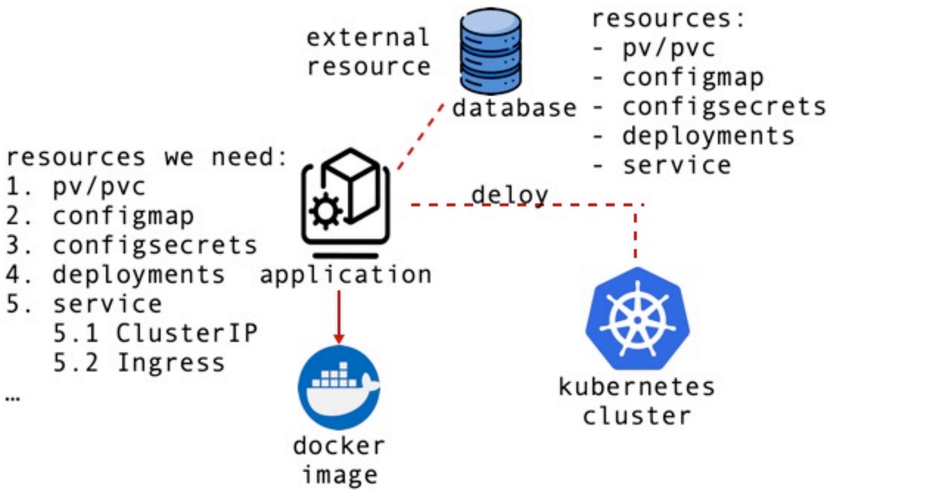
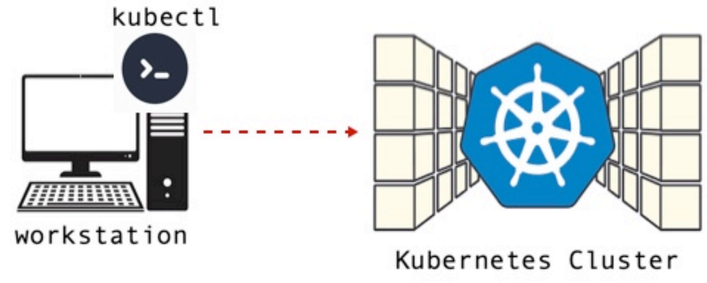
#1



1. identify the resources and their dependencies
2. write relevant manifest files to create these objects on the cluster

#2. how to rollout/deploy the application on to the kubernetes Cluster?
upon writing the manifests along with the dependent resource manifests, to rollout the application we need to do the below activities

1. clone/pull the ci repository onto the workstation
2. modify the manifest files with values pertaining to the env on which we are deploying the application
3. validate the changes we have made
4. identify the external dependent resources and have them deployed on the cluster based on their resource dependency order.
5. identify the application dependent resources, identify their dependency order and list them.
6. create each resource/object on after the another manually on the cluster and verify their status of the creation.
7. based on that deploy the next resource within the set.
8. once all the resources/objects including the external dependent resources are deployed, verify the health of the overall application.



From the above we can understand without Helm or any package manager, deploying the application on the cluster is a tedious job because:
1. we need to prepare spec files for external dependent resources, where most of the time these external resources like
 1.1 database servers (postgres, mysql)
 1.2 middleware servers like (redis, messaging queue, kafka)
etc
on common dependent resources that are required for many application deployments, and might have pre-created manifests for deploying them. Even then also there is no way to reuse these resources, rather we need to manually write the specs in deploying these common dependencies.

2. For each env, the values with which we want to deploy the objects will be different, so we need to manually modify the spec files which is errorneous
3. identifying the dependency order of these resources is a tedious job, unless he/she is the developer of those application manifests, understanding them and deriving dependencies takes huge amount of time and makes us complex as well
4. deploying each of these objects manually takes huge amount of time

So to overcome all of the above challenges in delivering an application on to the kubernetes cluster, The HELM package manager has been introduced.