## Architecting with Google Kubernetes Engine: Workloads

#### Scope:

1. Kubernetes Operations module

- kubectl command line utility

- use it to connect to clusters

- create and delete objects (e.g. pods)

- inspect cluster and objects

- view pods console output

- interact with pods

2. Deployments, Jobs and scaling module

- create and use Deployments

- create jobs and cron jobs

- Helm (<https://helm.sh/>) package manager for Kubernetes

- Google Cloud Kubernetes marketplace

3. Networking module

- Services

- use of Load Balancers to expose Services to external clients

4. Persistent data and storage module

- different Kubernetes storage abstractions

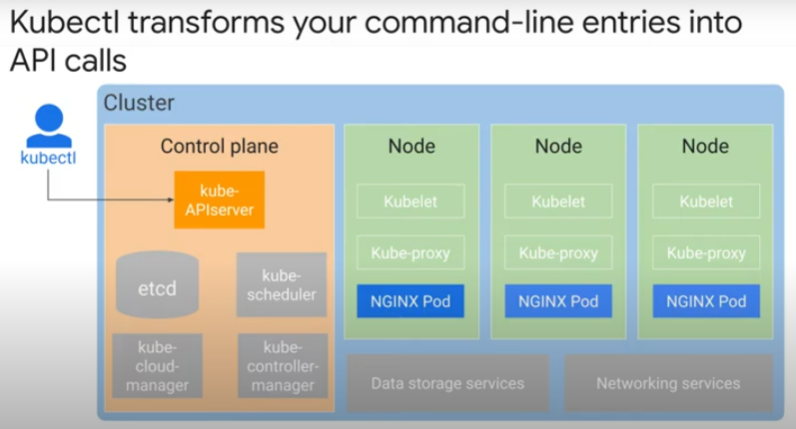
#### Kubernetes Operations module

Kubectl

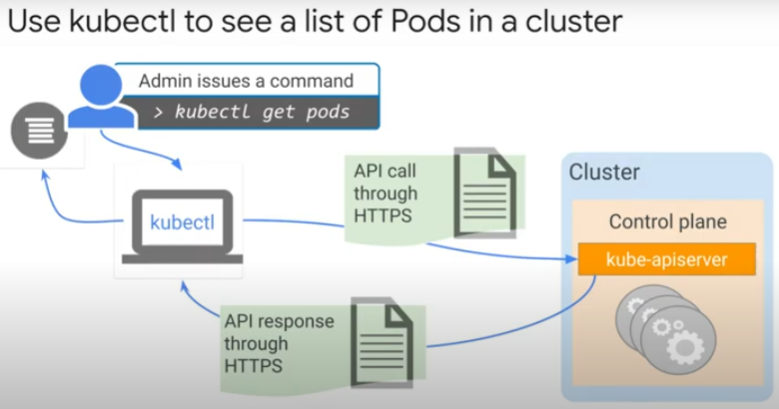
* command line utility
* before use, kubectl must be configured with location of Kubernetes cluster, and credentials to authenticate with cluster, e.g. by HTTPS
  + kubectl stores its default config file in home ($HOME) directory, hidden .kube folder
  + $HOME/.kube/config
  + can set list of kubeconfig files using *KUBECONFIG* environmental variable
  + config file contains context info (cluster, namespace, user) and info on authentication mechanisms (certificates, tokens)
  + https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/
  + to see configuration of kubectl command: *kubectl config view*, or manually open file
  + to retrieve credentials from GKE for specific cluster, use

*gcloud container clusters get-credentials [cluster name] –zone [zone name]*

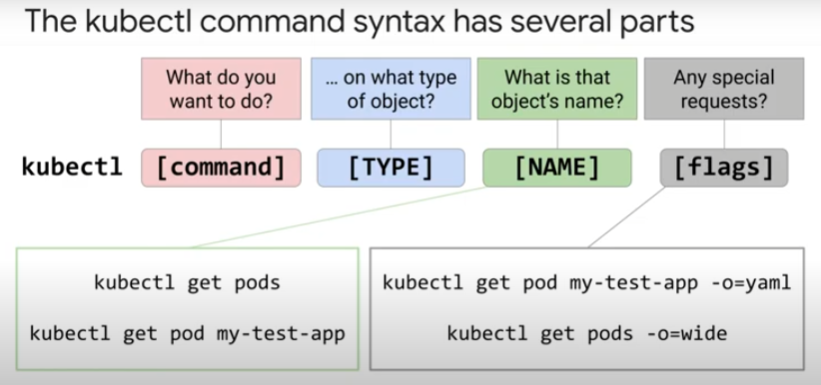
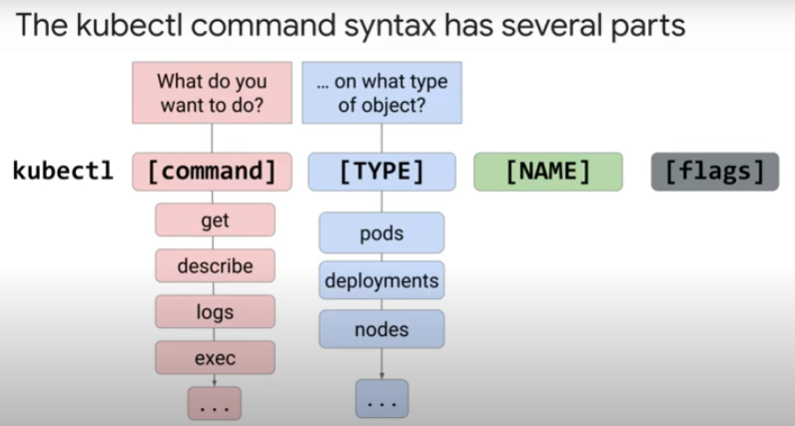
* + - gcloud get-credentials command writes info into kubeconfig file in .kube directory
    - if re-run command for different cluster, will update and over-ride kubeconfig file with credentials for cluster referenced in last run.
    - Run once per cluster before using kubectl commands
      * because kubectl commands need credentials to work
      * kubectl commands automatically refer to kubeconfig file once configured
  + gcloud commands are for authorized users to interact with Google Cloud
    - *gcloud container clusters get-credentials* allow authorized users to connect to GKE clusters
  + kubectl commands are for administering internal state of existing clusters
    - when need to create/change/teardown clusters, get cluster info, need to use *gcloud container clusters* commands.
    - <https://cloud.google.com/sdk/gcloud/reference/container/clusters>



* transforms command line entries into API calls to kube-apiserver on control plane of cluster
* for cluster configuration data, state data, metadata, kube-apiserver queries etcd.
  + etcd is a strongly consistent key-value store for large-scale distributed systems.
  + e.g. when administrator wants to get list of pods in cluster: *kubectl get pods*.
* Kube-apiserver responds to kubectl, through HTTPS.
* Kubectl interpretes API response and prints output to command prompt for administrator to read



kubectl command syntax



* *kubectl get pods* returns list of all pods
* *kubectl get pod [pod name]*returns info only on named pod
* *-o=yaml* flag is to output in yaml
* *-o=wide* flag to display list of pods in wide format, with additional columns of data
* *kubectl config –kubeconfig=....* to alter kubeconfig file
* *kubectl config –use/set-context* to work with context in kubeconfig files
  + a context is an entity defined in kubeconfig file to alias parameters
* https://kubernetes.io/docs/reference/kubectl/cheatsheet/
* [https://www.containiq.com/post/kubectl-config-set-context-tutorial-and-best-practices#:~:text=In%20Kubernetes%2C%20a%20context%20is,with%20a%20human%2Dreadable%20name](https://www.containiq.com/post/kubectl-config-set-context-tutorial-and-best-practices" \l ":~:text=In Kubernetes%2C a context is,with a human-readable name).