Kubernetes From Scratch

Cloud Native Application Development Series

Kubernetes Cluster Components

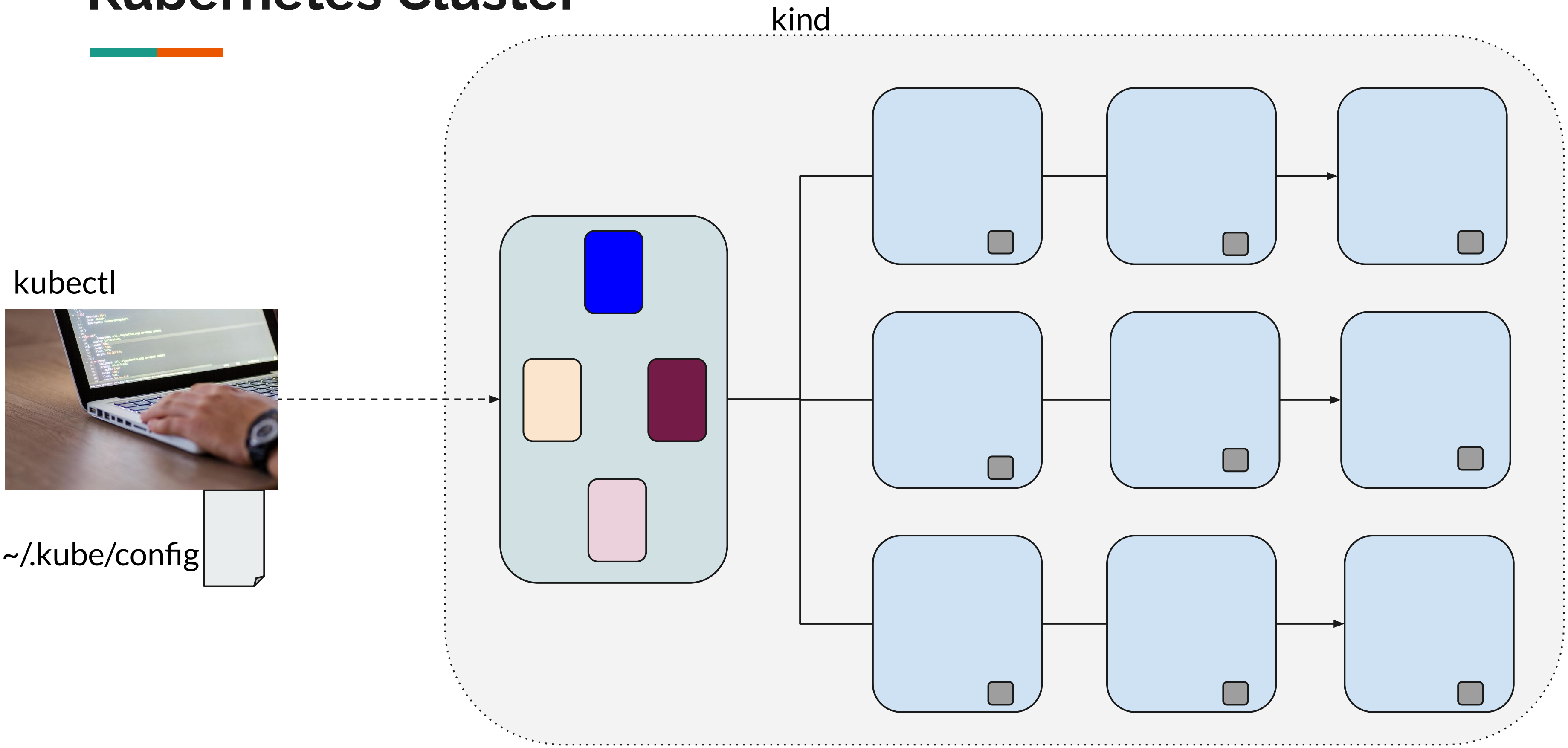
High level description to give you an idea.

	Component	Description
master / control-plane	**api-server**	APIs for clients to talk to the cluster and create workloads
	etcd (et-c-d)	A distributed key-value store to store cluster data
	controller-manager	A process which continuously monitors workloads / nodes etc
	scheduler	Workload scheduler
node	**kubelet**	An agent which creates containers and monitors
	container runtime (docker)	To create container, we need to have container runtime on the node
	kube-proxy	Maintains the network rules on the node for communication among workloads in the cluster

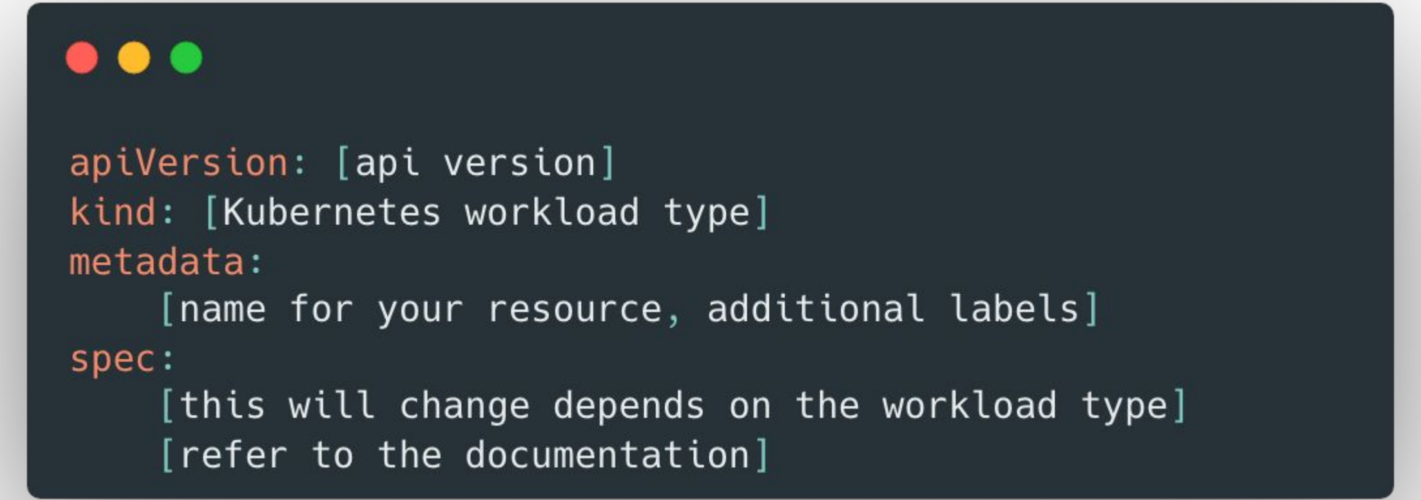
kube Config

- A config file to organize cluster info
- **$HOME/.kube/config**
- environment variable **KUBECONFIG=/a/b/c**

Kubernetes Cluster



Kubernetes Resource Yaml format



```
apiVersion: [api version]
kind: [Kubernetes workload type]
metadata:
    [name for your resource, additional labels]
spec:
    [this will change depends on the workload type]
    [refer to the documentation]
```

Pod Status

Status	Description
Pending	Node is yet to be assigned
ContainerCreating	Kubelet is working on creating container
Running	Kubelet started the container
ErrImagePull / ImagePullBackOff	Failed pulling image. Kubelet will retry with some delay
Completed	Container exited successfully
Error	Container exited with error
CrashLoopBackOff	There is a problem in running the container. Kubelet is retrying with delay. (no issues in pulling the image)
Terminating	Pod is getting deleted

Pod Container Restart Policy

Policy	Description
Never	Completed / Error → Do NOT Restart
Always	Completed / Error → Restart
OnFailure	Error → Restart

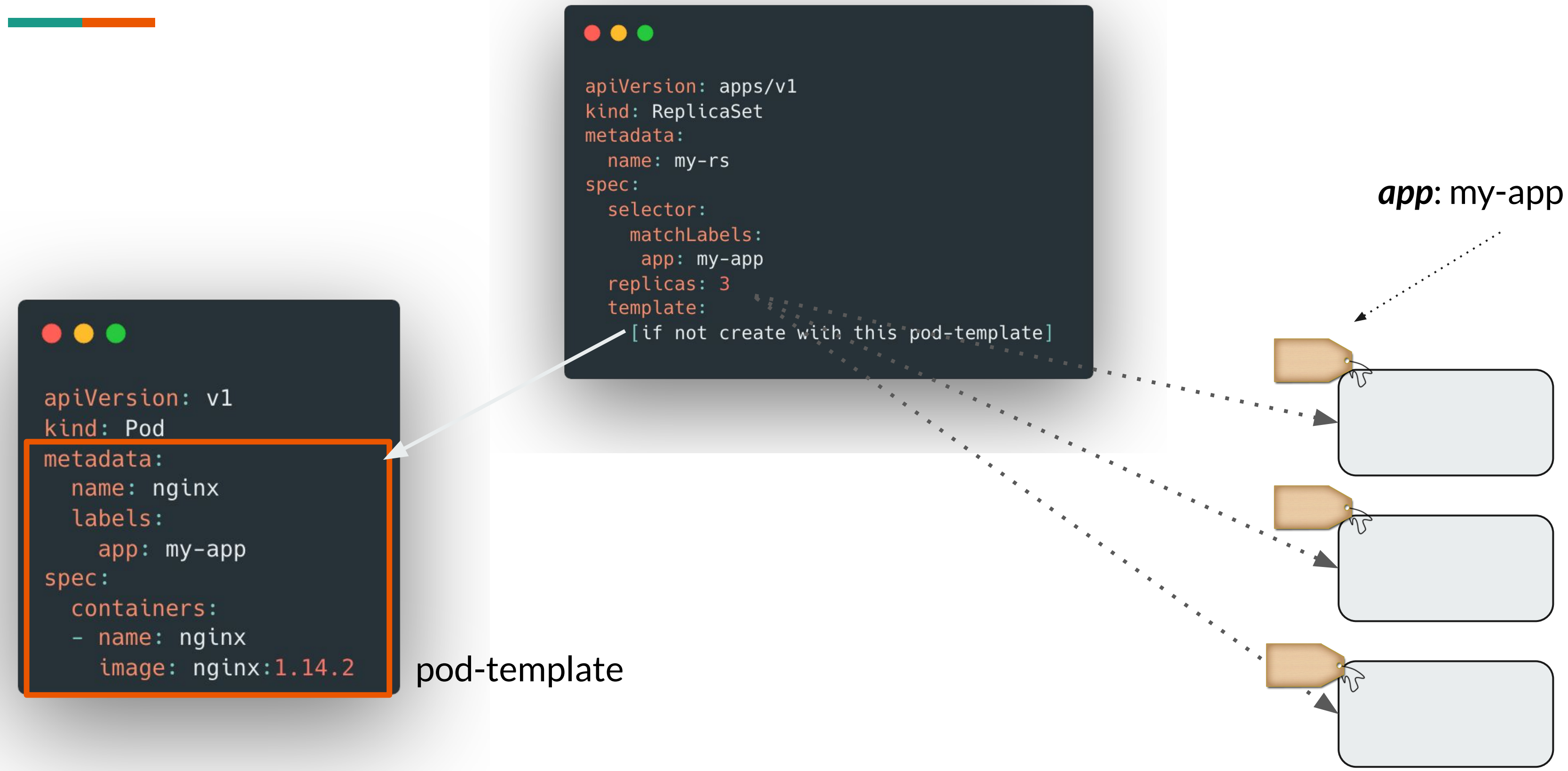
Kubectl Commands!

Command	Description
kubectl get nodes	To list the nodes in the cluster
kubectl create -f [file-name]	To create resources in the given yaml
kubectl delete -f [file-name]	To delete resources in the given yaml
kubectl get pod	To list all the pods (in the default namespace)
kubectl get pod –show-labels	To list pod labels
kubectl get pod -l team=team-a **kubectl get pod -l team!=team-a**	To query pods based on labels
kubectl get pod [pod-name] -o yaml	To provide pod info in the yaml format

Kubectl Commands!

Command	Description
kubectl describe pod [pod-name]	To describe pod
kubectl delete pod [pod-name]	To delete the given pod
kubectl delete pod --all	To delete all pods
kubectl logs [pod-name]	To check the pod log (assuming single container)
kubectl logs [pod-name] -c [container-name]	To check the specific container log for a pod when there are multiple containers
kubectl exec -it [pod-name] -c [container-name] -- bash	To access the container of the pod
kubectl port-forward [pod-name] 8080:80	To access our application APIs from our host for debugging

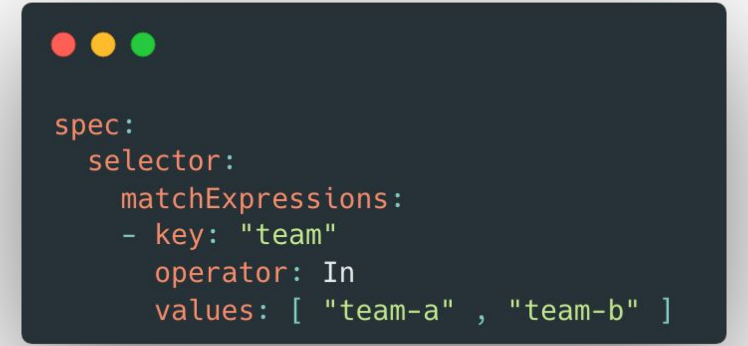
ReplicaSet



```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: my-rs
spec:
  selector:
    matchLabels:
      app: my-app
  replicas: 3
  template:
    [if not create with this pod-template]
```

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    app: my-app
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
```

ReplicaSet Match Expressions



```yaml
spec:
  selector:
    matchExpressions:
    - key: "team"
      operator: In
      values: [ "team-a" , "team-b" ]
```

Kubectl create!

Command	Description
kubectl create -f [file-name]	To create resources in the given file
kubectl create -f .	To create resources from all the yamls in the current directory
kubectl create -f http://vinsguru.com/k8s.yaml	To create resources in the given url

Kubectl apply!

Command	Description
kubectl apply -f [file-name]	To ***create/update*** resources in the given file
kubectl apply -f .	To ***create/update*** resources from all the yamls in the current directory
kubectl apply -f http://vinsguru.com/k8s.yaml	To ***create/update*** resources in the given url

Kubectl Commands!

Command	Description
kubectl get [kind]	To list resources of specific kind
kubectl get [kind] [resource-name]	To get the specific resource
kubectl get [kind] --show-labels	To show labels
kubectl get [kind] -l app=my-app	To query resources
kubectl get [kind] -o yaml	To format the output in yaml

Kubectl Commands!

Command	Description
kubectl describe [kind] [resource-name]	To describe the resource
kubectl delete [kind] [resource-name]	To delete the given resource
kubectl logs [kind]/[resource-name]	To check the log
kubectl exec -it [kind]/[resource-name] – bash	To access the container of the pod
kubectl port-forward [kind]/[resource-name] 8080:80	To access our application APIs from our host for debugging

Kubectl Rollout

Command	Description
kubectl rollout history deploy/[name]	To get the rollout history
kubectl rollout undo deploy/[name]	To rollback to the previously deployed version
kubectl rollout undo deploy/[name] --to-revision=3	To deploy a specific version
kubectl rollout history deploy/[name] --revision=3	To see the pod template changes

Deployment Strategy!

<table>
<tr><th>Strategy</th><th>Description</th></tr>
<tr><td><b>recreate</b></td><td>Terminate the old pods and create the new pods</td></tr>
<tr><td><b>rolling update</b></td><td>Gradually roll out the changes. We can have a mix of old and new pods temporarily.
Below properties can be a number or %
<table>
<tr><th>Properties</th><th>Description</th></tr>
<tr><td><b>maxSurge</b></td><td>Max number of additional pods that can be created</td></tr>
<tr><td><b>maxUnavailable</b></td><td>Max number of pods that can be terminated</td></tr>
</table>
</td></tr>
</table>

Service Types

Type	Description
ClusterIP	For communication within the k8s cluster. Can not be accessed from outside the cluster. (AWS/GCP cloud - private subnet communication) . This is the default option if we do not specify. Mostly this is what we would use.
NodePort	Can be accessed from outside via k8s master/nodes via Specific port (can be used for testing)
LoadBalancer	To be used in AWS/ GCP/Azure..cloud providers. Can be used to receive traffic from outside

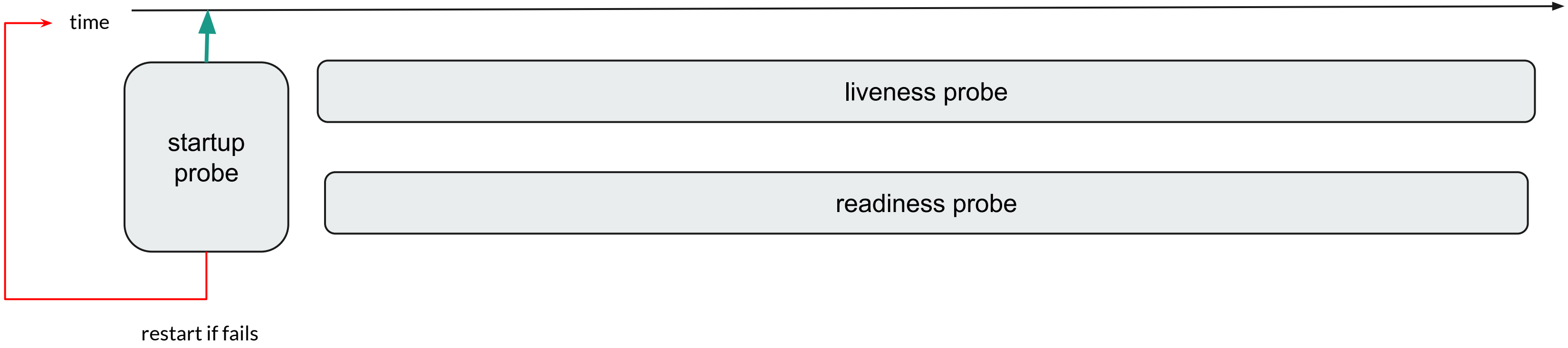
Kubectl Commands

Command	Description
kubectl create ns dev	To create a namespace called dev
kubectl get pod -n dev	To get the pods from dev namespace
kubectl delete ns dev	To delete namespace dev

Probe Types

Probes	Description	Action If Fails
startupProbe	To check if the application inside the container has started.	restart
livenessProbe	To check if the application is still alive.	restart
readinessProbe	To check if the application is ready to take the requests from ***service***	remove from service

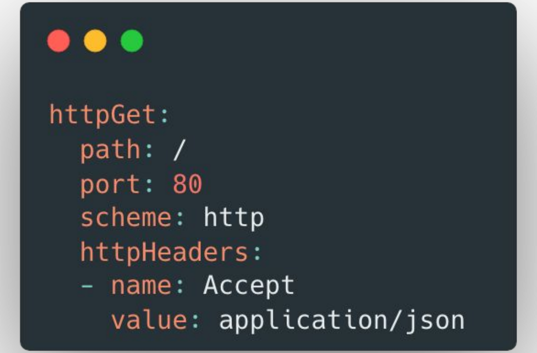
Probe Execution Phase



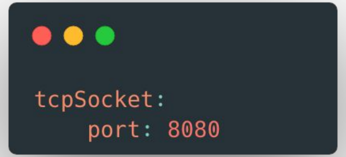
Probe

Options	Description
exec	Execute any command to check. for ex: ***cat /tmp/app.log***
httpGet	To invoke a http endpoint. for ex: ***/health***
tcpSocket	To check if the app started listening on specific port.

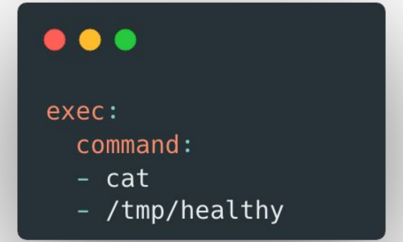
Probe



```yaml
httpGet:
  path: /
  port: 80
  scheme: http
  httpHeaders:
  - name: Accept
    value: application/json
```



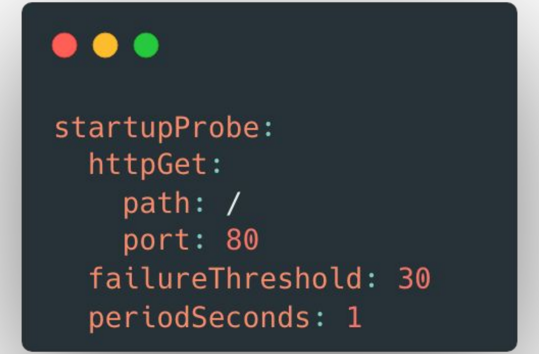
```yaml
tcpSocket:
    port: 8080
```



```yaml
exec:
  command:
  - cat
  - /tmp/healthy
```

Probe Properties

Properties	Default Values
initialDelaySeconds	0
periodSeconds	10
timeoutSeconds	1
successThreshold	1
failureThreshold	3



```
startupProbe:
  httpGet:
    path: /
    port: 80
  failureThreshold: 30
  periodSeconds: 1
```