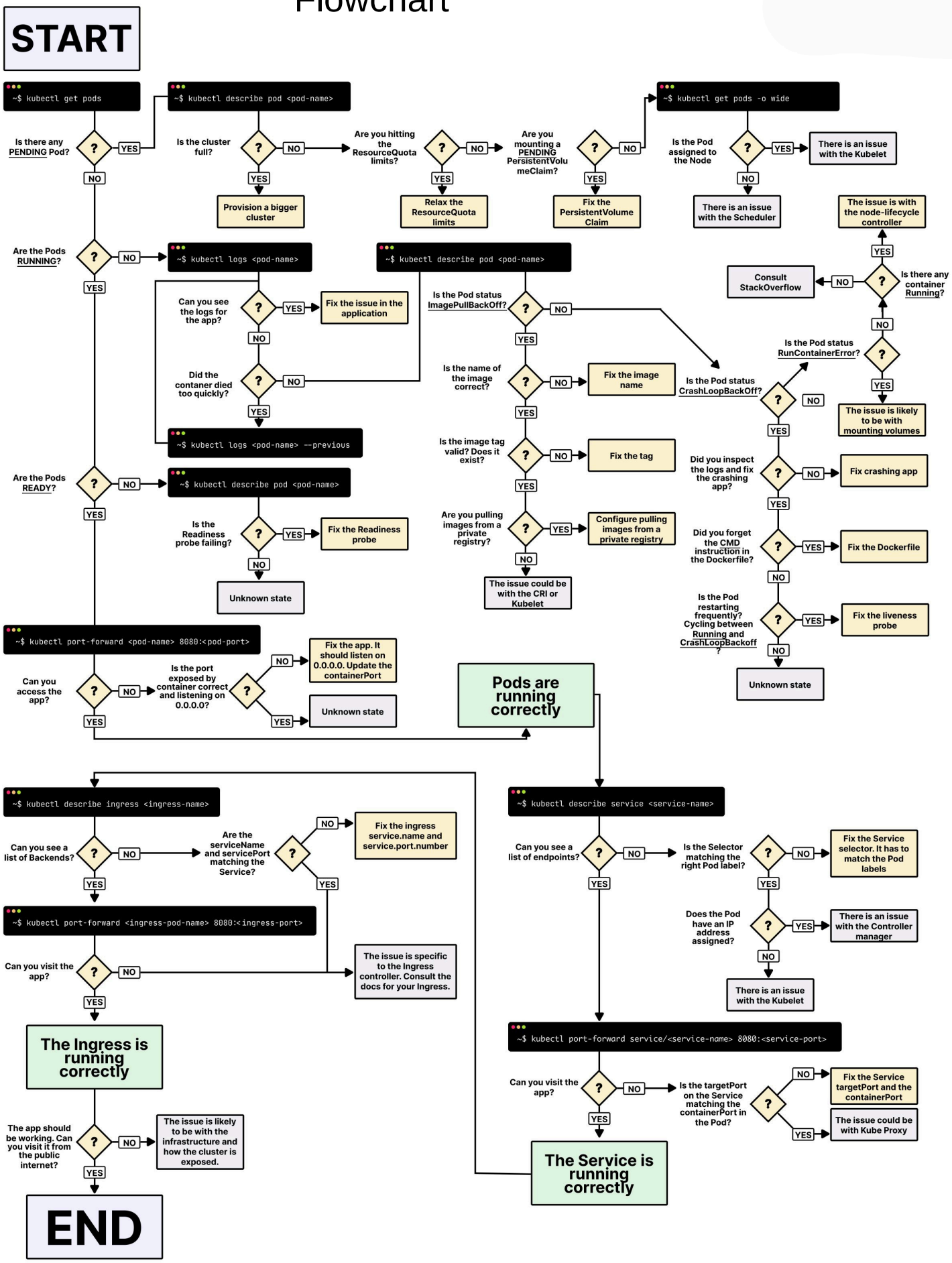
Flowchart


START
~$ kubectl get pods
Is there any PENDING Pod?
YES
NO
~$ kubectl describe pod <pod-name>
Is the cluster full?
YES
Provision a bigger cluster
NO
Are you hitting the ResourceQuota limits?
YES
Relax the ResourceQuota limits
NO
Are you mounting a PENDING PersistentVolumeClaim?
YES
Fix the PersistentVolume Claim
NO
~$ kubectl get pods -o wide
Is the Pod assigned to the Node
YES
There is an issue with the Kubelet
NO
There is an issue with the Scheduler
Are the Pods RUNNING?
NO
~$ kubectl logs <pod-name>
Can you see the logs for the app?
YES
Fix the issue in the application
NO
Did the container died too quickly?
NO
YES
~$ kubectl logs <pod-name> --previous
~$ kubectl describe pod <pod-name>
Is the Pod status ImagePullBackOff?
NO
YES
Is the name of the image correct?
NO
Fix the image name
YES
Is the image tag valid? Does it exist?
NO
Fix the tag
YES
Are you pulling images from a private registry?
YES
Configure pulling images from a private registry
NO
The issue could be with the CRI or Kubelet
Is the Pod status CrashLoopBackOff?
NO
YES
Did you inspect the logs and fix the crashing app?
NO
Fix crashing app
YES
Did you forget the CMD instruction in the Dockerfile?
YES
Fix the Dockerfile
NO
Is the Pod restarting frequently? Cycling between Running and CrashLoopBackoff?
YES
Fix the liveness probe
NO
Unknown state
Is the Pod status RunContainerError?
YES
The issue is likely to be with mounting volumes
NO
Is there any container Running?
YES
The issue is with the node-lifecycle controller
NO
Consult StackOverflow
Are the Pods READY?
NO
~$ kubectl describe pod <pod-name>
Is the Readiness probe failing?
YES
Fix the Readiness probe
NO
Unknown state
~$ kubectl port-forward <pod-name> 8080:<pod-port>
Can you access the app?
NO
Is the port exposed by container correct and listening on 0.0.0.0?
NO
Fix the app. It should listen on 0.0.0.0. Update the containerPort
YES
Unknown state
YES
Pods are running correctly
~$ kubectl describe ingress <ingress-name>
Can you see a list of Backends?
NO
Are the serviceName and servicePort matching the Service?
NO
Fix the ingress service.name and service.port.number
YES
~$ kubectl port-forward <ingress-pod-name> 8080:<ingress-port>
Can you visit the app?
NO
The issue is specific to the Ingress controller. Consult the docs for your Ingress.
YES
The Ingress is running correctly
The app should be working. Can you visit it from the public internet?
NO
The issue is likely to be with the infrastructure and how the cluster is exposed.
YES
END
~$ kubectl describe service <service-name>
Can you see a list of endpoints?
NO
Is the Selector matching the right Pod label?
NO
Fix the Service selector. It has to match the Pod labels
YES
Does the Pod have an IP address assigned?
YES
There is an issue with the Controller manager
NO
There is an issue with the Kubelet
YES
~$ kubectl port-forward service/<service-name> 8080:<service-port>
Can you visit the app?
NO
Is the targetPort on the Service matching the containerPort in the Pod?
NO
Fix the Service targetPort and the containerPort
YES
The issue could be with Kube Proxy
YES
The Service is running correctly

TROUBLESHOOTING TECHNIQUES IN K8S

CRASHLOOPBACKOFF: it error indicates that will automatically starting, crash and then restart the pod again repeatedly.

1. check the logs of the pods : kubectl logs pod-name
2. describe the pod : kubectl describe pod pod-name
3. ENSURE THAT THE POD CONFIGURATIONS AND ENVIRONMENT VARIABLES ARE CORRECTLY SET.
4. Verify the entrypoint of a container
5. Ensure that there are no resource limitations preventing the pod from running.

IMAGEPULLBACKOFF: this error occurs when k8s cant able to pull image from registry (Dockerhub, ECR)

1. describe the pod : kubectl describe pod pod-name
2. Check The imagename of pod config file (YAML)
3. Check the secrets : kubectl get secret

PENDINGPODS: This error will occurs when a pod is in pending state when it is unable to schedule on to a node

10 pods (worker : t2.micro)

11th pod pending

if any pod was deleted from that 10 pods, then 11th pod will be created

1. describe the pod : kubectl describe pod pod-name
2. check the nodes : kubectl get no

NODENOTREADY: A node is in a NoTReady State when its unable to participate in the cluster

1. describe the node : kubectl describe node node-name
2. Review the node logs : kubectl logs node node-id

UNAUTHORIZED ERROR: This error indicates a failure in API Authentication

1. Check your credentials
2. describe the resource : kubectl describe resource resource-name
3. Check the RBAC policies
4. Check the service account tokens are valid or not
5. check the kubeconfig file configured correctly or not

FailedScheduling: This error occurs when the scheduler cant find any suitable node for a pod.

1. describe the pod : kubectl describe pod pod-name
2. check the resources of a pod (CPU, Memory)

OOMKILLED : This error will occur when a pod was killed itself because it exceeds the memory limits.

1. check the resources of pods
2. describe the pod
3. increase the pod memory limit
4. optimising the app to use less memory

ERROR CREATING LOADBALANCER: it error will occurs when k8s cant communicate with cloud provider.

1. describe the service : kubectl describe service service-name
2. check the IAM permissions on AWS account
3. Check the cloud provider integration configuration is successfully configured or not
4. Verify the service file annotations (YAML)

ContainerCreatingerror: this error will occurs when we didnt pass correct container configurations on yaml file

1. describe the pod : kubectl describe pod pod-name
2. check the container configurations on YAML file
3. Imagepull Issue check
4. Verify the node has sufficient resources or not
5. Ensure that there are no networking issues.