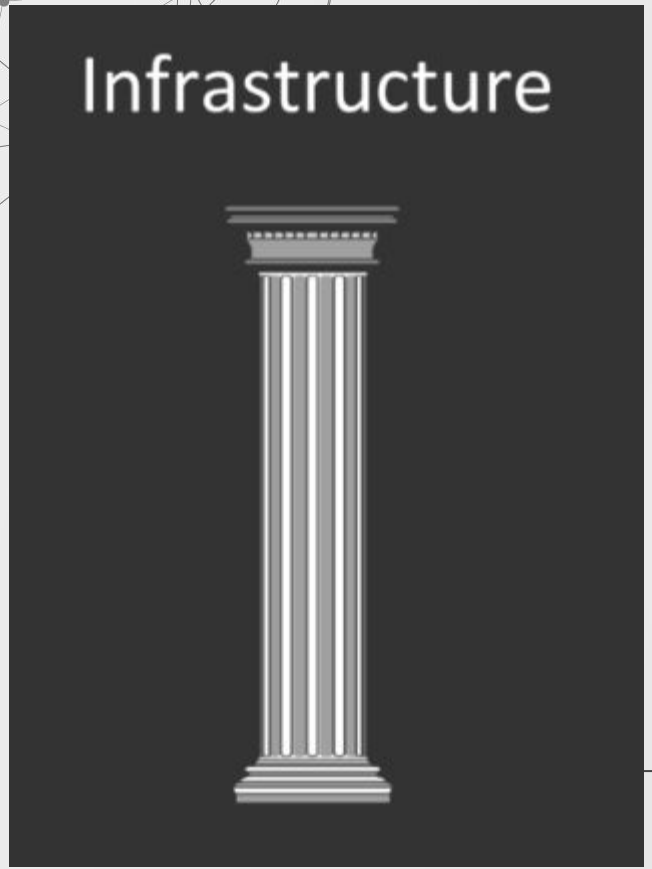
KUBERNETES TROUBLESHOOTING

Application introspection and debugging

Develop a Hypothesis

Clue 1
+ Clue 2
+ Kubernetes Knowledge

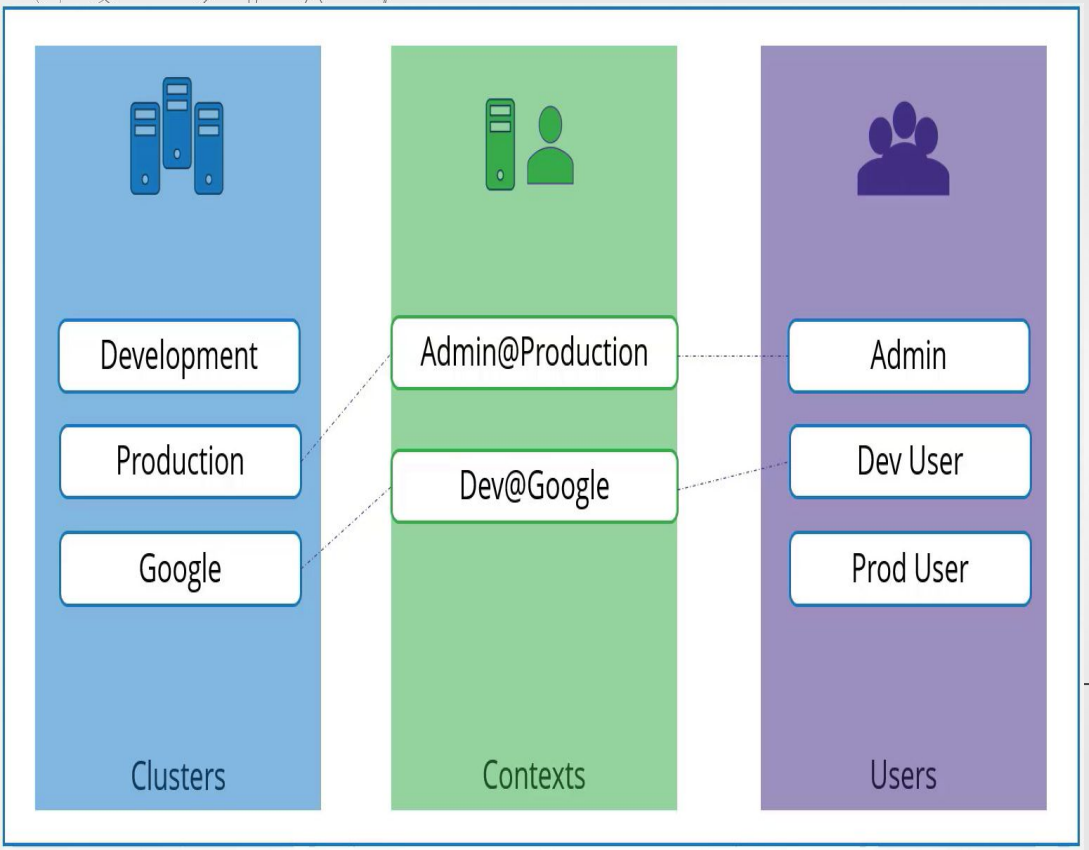
Probable Cause



Self-managed or cloud-managed?

...

$HOME/.kube/config



How can I access RTE clusters?

...

1st scenario

Deploy two Pods in a Namespace. Check the Pod's statuses and suggest a solution to the problem you face.

2nd scenario

Failed Pod. Someone in your organization has come to you for help. They are having trouble deploying a container.

3rd scenario

You deployed a voting application, but votes are not being cast and the worker Pod is not healthy. Your colleagues suspect the problem is at the application level. Check the Pod's health and suggest a solution

4th scenario

Several weeks later on a team video chat, your coworker says there's a failed Pod again.
This time, the same app. Kuard, isn't working. Your teammate explains that after some testing, it can't be re-deployed. They go on to say that the Kubernetes manifest hasn't changed. You begin to investigate.

5th scenario

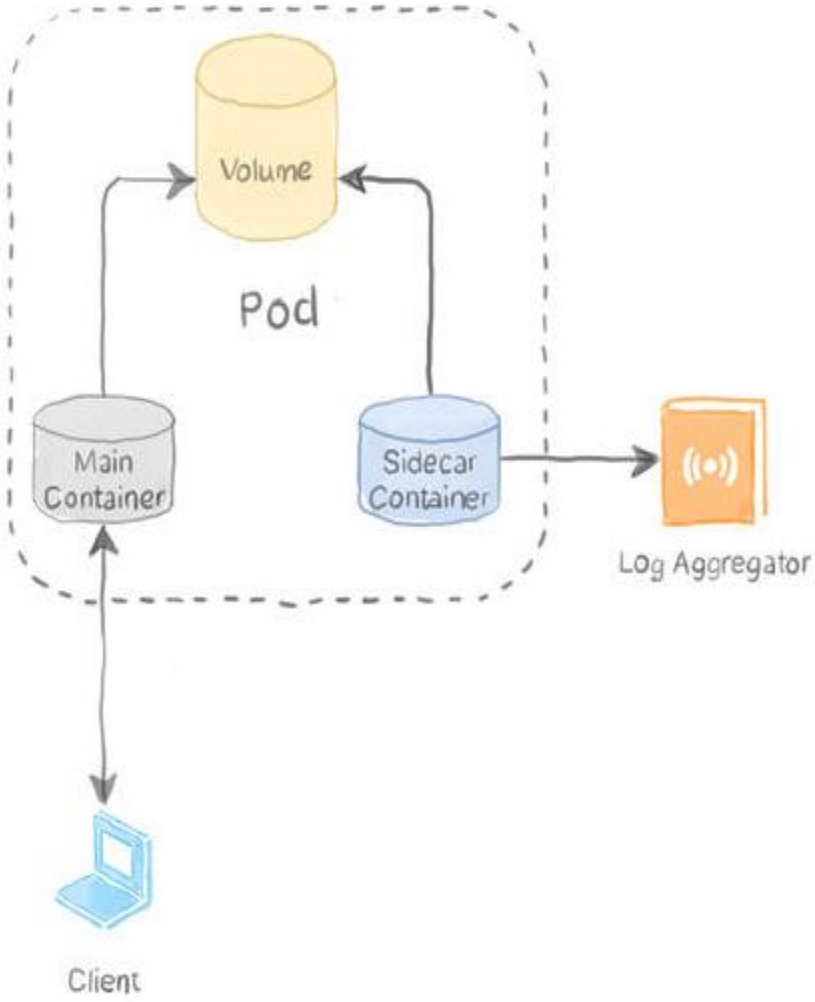
You receive a call from an employee who needs help with their new application. The user states that after deploying their containers, they are unable to access them through a web browser.

6th scenario

After deploying some nginx Pods and a Service, you notice, when trying to access your Service by IP from one of your Pods, in different namespaces, that it is not working properly. Correct the mistake and run some tests again.

7th scenario

Deploy and debug your kuard Pod, now with some Init Containers.


Volume
Pod
Main Container
Sidecar Container
Log Aggregator
Client

9th scenario

Deploy and debug a Pod with a Sidecar container.

1. User makes changes to a CRD
2. The operator tracks the CRD and identifies change events
3. The operator reconciles the CRD state with the desired state
4. The operator adjusts cluster state to the desired state