Why pod failed: There could be 2 reasons- 1) Startup failure- Container don’t start. 2) Runtime failure- Application code fail after pod start.

3) Kubernetes ImagePullBackOff [or] ErrImagePull

## What Does an ImagePullBackOff Error Mean?

The ImagePull part of the ImagePullBackOff error primarily relates to your Kubernetes container runtime being unable to pull the image from a private or public container registry. The Backoff part indicates that Kubernetes will continuously pull the image with an increasing backoff delay. Kubernetes will keep on increasing the delay with each attempt until it reaches the limit of five minutes.

**backoff strategy** is a technique that we can use to retry failing function calls after a given delay

It seems like a generalized statement to say that container runtime (be it Docker, containerd, etc.) fails to pull the image from the registry, but let’s try to understand the possible causes for this issue.

Here are some of the possible causes behind your pod getting stuck in the ImagePullBackOff state:

* Image doesn’t exist.
* Image tag or name is incorrect.
* Image is private, and there is an authentication failure.
* Network issue.
* Registry name is incorrect.
* Container registry rate limits.

There could be 3 primary reasons for it 1) Image name is not correct, 2) tag is not correct 3) authentication while pulling the image. Kubectl get pod 

kubectl describe pod <pod-name> This command is used to check event in the pod kubectl edit pod <pod-name>

A pod is the smallest unit in Kubernetes (K8S). They should run until they are replaced by a new deployment. Because of this, **there is no way to restart a pod**, instead, it should be replaced.