Postmortem: `myapp-deployment` CrashLoopBackOff Due to Missing ConfigMap

Incident Time:

June 25, 2025 – 07:10 PM EDT

The Kubernetes deployment `myapp-deployment` was observed to be stuck in the `CrashLoopBackOff` state. Upon investigation, the root cause was identified as a missing ConfigMap named `myapp-config`, which was required for mounting a volume inside the container.

What Happened?

- At approximately 3:10 PM, we rolled out a new version of `myapp-deployment` into the `production` environment. Shortly after, the pod failed to start and entered a `CrashLoopBackOff` state. Initial inspection of the pod status showed it never fully transitioned to `Running`. Instead, it was stuck in a repeated failure cycle, never getting past the container creation phase.

Investigation:

- We ran →
    - kubectl describe pod myapp-deployment-78c7449747-m7l8n

  Command result we found repeated warning messages from the kubelet:

    - Warning FailedMount kubelet MountVolume.SetUp failed for volume "config-volume" : configmap "myapp-config" not found

      This pointed us toward a missing dependency: the `myapp-config` ConfigMap, which is required to mount application configuration files into the container.

    - We confirmed the issue by running:
        - kubectl get configmap myapp-config -n test
        - Result: Error from server (NotFound): configmaps "myapp-config" not found

Root Cause:

The deployment manifest referenced a ConfigMap named `myapp-config`, used to mount configuration data into the container. However, this ConfigMap had not been created in the `test` namespace prior to the deployment.

Because Kubernetes could not mount the expected volume, the pod could not fully initialize, resulting in repeated container startup failures.

Resolution:

Once identified, the issue was resolved with the following steps:

- The missing ConfigMap was created:
    - kubectl create configmap myapp-config --from-file=config.yaml -n test #created in the repo for reference.
- The deployment was restarted:
    - kubectl rollout restart deployment myapp-deployment -n test
- Checked the pod status:
    - kubectl get po -n test | grep myapp-deployment
        - Result: found the service up and running.

What We Learned?

- Kubernetes will not start containers if a volume source (e.g., ConfigMap) does not exist.
- ConfigMaps are namespace-scoped and must be created in the correct namespace before deployment.
- Even simple mistakes, such as the incorrect ordering of resource applications, can cause cascading failures in a startup.

What to improve?

- We will update our CI/CD pipeline to apply ConfigMaps and Secrets before dependent workloads.
- We'll include a pre-deploy validation step to ensure all referenced ConfigMaps exist using: kubectl get configmap myapp-config -n test. (can also adopt helm to solve this)
- For future deployments, we'll include readiness checks and dry-run validations for YAML dependencies.
- We need to add proper alerts and monitors to where we can catch these issues before the customers are affected and start to report issues to improve the reliability.