

What happens if a Kubernetes master node crashes?

If a **Kubernetes master node crashes**, the **cluster loses control functionality**—but not necessarily the running workloads—because the **control plane** components like the API server, scheduler, and controller-manager reside on the master node. Here's a quick summary and a demo you can try using kind (Kubernetes IN Docker):

What Happens When Master Node Crashes

Component Impact if Crashed Master

API Server No kubectl commands work (cluster is unreachable)

Scheduler No new pods get scheduled

Controller Manager Replicasets, jobs, failovers don't work

Running Pods Continue running if worker nodes are healthy

So, your application may keep working, but you **can't scale, deploy new apps**, or respond to node failures unless the master is back up.

Real-World Best Practice

In production:

- You should have multiple control plane nodes for HA (High Availability).
- Use external etcd with backups.
- Use cloud-managed K8s (like GKE, EKS) if you want control-plane HA handled for you.

What happens when the Kubernetes master node crashes?

Select all that apply.

- A. kubectl stops working
- **B.** New pods can't be scheduled
- C. Replicasets and failovers stop

- D. Running pods get deleted
- E. Running pods continue
- F. New apps can still be deployed
 - A. kubectl stops working: True. kubectl communicates directly with the API server on the master node. If the master is down, kubectl cannot reach the API server and therefore will not function.
 - **B. New pods can't be scheduled: True.** The scheduler, a component of the control plane running on the master node, is responsible for placing new pods onto worker nodes. If the master is down, the scheduler cannot function, and no new pods can be scheduled.
 - C. Replicasets and failovers stop: True. The controllers (like the ReplicaSet controller and other custom controllers) that manage the desired state of your application (including scaling and handling failures) run on the master node. If the master is down, these controllers cannot observe the cluster's state or take action to create new pods or handle failovers.
 - **D. Running pods get deleted: False**. The kubelet, which runs on each worker node, is responsible for managing the pods running on that node. Once pods are scheduled and running on worker nodes, they will continue to run even if the master node goes down. The kubelet will keep them alive. However, if a node itself fails, the pods on that node will be affected.
 - E. Running pods continue: True. As mentioned above, the kubelet on each worker node independently manages the running pods on that node. Therefore, existing pods will continue to run even if the master node is unavailable.
 - F. New apps can still be deployed: False. Deploying new applications involves creating new deployments, replicasets, etc., which requires interaction with the API server and the scheduler on the master node. If the master is down, these operations cannot be performed.

✓ Correct Answers:

A, **B**, **C**, **E**