

Kubernetes Pod States | An Introduction

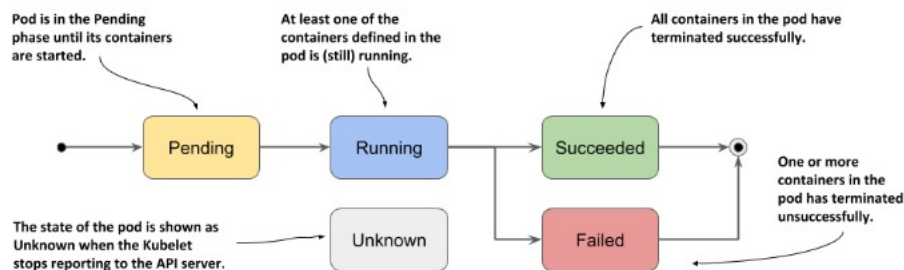
*There are five Pod States namely Pending, Running, Succeeded, Failed and Unknown on Kubernetes. Here our **Support team** will explain each state briefly.*

Bobcares, as part of our **Server Management service**, handles all inquiries, large or small, that come our way.

Do you want to learn more? Read on and get in contact with us if you have any additional questions.

Kubernetes Pod States

The status field of a Pod is a PodStatus object with a phase field. A Pod's phase is a high-level summary of where the Pod is in its lifecycle.



The following are the possible values for phase:

1. **Pending:** The Kubernetes system accepts the Pod, but doesn't create one or more of the Container images. This includes time spent prior to being scheduled as well as time spent downloading images over the network, which may take some time.
2. **Running:** The Pod associates with a node, and then creates all containers. At least one Container is still active or is in the process of being started or restarted.
3. **Succeeded:** All Containers in the Pod successfully terminates and will not restart.
4. **Failed:** All Containers in the Pod have terminated, and at least one of them has failed. That is, either the Container exited with a non-zero status or the system terminated it.
5. **Unknown:** The state of the Pod is unknown for some reason, most likely due to a communication error with the Pod's host.

Pod Conditions

The phase of a pod says little about the condition of the pod. The main pod conditions are:

1. **PodScheduled:** Shows whether or not the pod has been scheduled to a node.
2. **Initialized:** The pod's init containers have all completed successfully.
3. **ContainersReady:** All containers in the pod indicate that they are

ready.

4. **Ready:** The pod is ready to provide services to its clients.

Kubernetes tracks the state of each container within a Pod in addition to the overall phase of the Pod. Container lifecycle hooks can trigger events at specific points in a container's lifecycle.

Container States

While looking into Kubernetes Pod States, we also need to know about Container States. Once a Node assigns a Pod by the scheduler, the kubelet begins creating containers for that Pod using a container runtime. Waiting, Running, and Terminated are the three possible container states.

1. **Waiting:** A container is Waiting if it is not in the Running or Terminated states. A container in the Waiting state is still performing the operations necessary for it to complete startup.
2. **Running:** The Running status indicates that a container is running normally.
3. **Terminated:** A container in the Terminated state started execution and then either finished or failed for some reason.

[Looking for a solution to another query? [We are just a click away.](#)]

Conclusion

Pods have a defined lifecycle that begins with the Pending phase, progresses to the Running phase if at least one of its primary containers starts normally, and then to the Succeeded or Failed phases depending on whether any container in the Pod terminated in failure. To summarize, our **Support team** has went over the Kubernetes Pod States details.

PREVENT YOUR SERVER FROM CRASHING!

Never again lose customers to poor server speed! Let us help you.

Our server experts will monitor & maintain your server 24/7 so that it remains lightning fast and secure.

[GET STARTED](#)

Related posts:

4 [Managed Server Maintenance Packages](#)