

Contents



A person wearing a white shirt is holding a pen in their right hand, positioned as if about to write. The background is blurred, showing what appears to be a desk or table with some papers and a small object. The overall image has a light, warm tone.

DevOps Reference Series

Kubernetes

by

Kuzey Koç

Contents



Applied Notes

Notes

- kubernetesin mimarisinde 2 temel kavram vardır. **control plane** ve **worker node**
 - control plane
 - * kubernetes spesifik proseslerin çalıştığı master.
 - * control plane içinde çalışan önemli prosesler
 - api server
 - kısa ismi **api**
 - controller manager
 - kısa ismi **c-m**
 - cluster içini sürekli izler ve bir sorun çıktığında onarır
 - bir pod ya da container öldü diyelim. hemen onu tamir eder. replicaset buna örnek verilebilir
 - scheduler
 - kısa ismi **sched**
 - uygulamaları, worker nodelara dağıtır. bunu yaparken de workload ya da cpu ve ram kullanımlarına göre yapar
 - etcd
 - key value store. yani kubernetes clusterının ayarlarını saklar. configürasyonu, her nodun statelerini vs tutar
 - ayrıca bir clusterın backupını almak istiyorsak etcd'den almamız gerekir
 - * cluster içinde en az 2 control plane olması gerekir. çünkü biri düşerse diğer ayakta kalacak şekilde kurulması gerekir
 - worker node
 - * kullanıcı uygulamalarını çalıştığı ve control plane tarafından yönetilen serverlar
- kubernetes için var olan web tabanlı arayüzler vardır. bunlardan biri **kubernetes dashboard**
- kubernetes komponentleri
 - pod
 - * kubernetesdeki en temel birimdir
 - * bir ya da birden çok containerı barındırır
 - service
 - * pod içinde çalışan konteynerların kendi ip adresleri vardır ancak ephemeral oldukları için bu konteynerlar replicaset tarafından değiştirilir. bu olunca da ip adresi değişir. ip adresi değışse bile diğer uygulamaların bu pod içinde çalışan uygulamaya erişiminin devam edebilmesi için **service** diye bir k8s komponenti geliştirildi.
 - * service, bir pod'a atanmış statik ip adresi gibi düşünülebilir

- * servis aynı zamanda bir **load balancer** gibi de davranır.
- ingress
 - * pod içinde çalışan bir service browserdan erişmek istediğimizde ingress kullanabiliriz. burada ingress, service ile iletişim kurar. service ise gerekli konteynerin servisini ingresse açar.
- configmap
 - * pod içinde çalışan uygulamanın ayarları ya da ortam değişkenlerini buraya yazabiliriz ve k8s buradan okuyup işlem yapabilir
 - * configmap kullanmanın avantajı, bu değişkenleri dockerfile'a yazmayıp, uygulamayı her seferinde build etmememizi sağlar.
- secret
 - * configmap gibi ancak daha hassas verileri saklamak için kullanılır. örneğin api tokenları, şifreler gibi.
 - * secretlar **plain text** olarak etcd'de saklanır. api erişimi olan herkes bu bilgilere erişip secretları okuyup değiştirebilir. bu yüzden secretları encrypt etmek için 3rd party araçlar kullanılması gerekir. Vault gibi
- volume
 - * pod içinde çalışan ve veri üreten konteynerlerin bu verileri **persistence** yapması için bir disk alanı gerekir. bunun için volumelar vardır. veritabanı gibi servislerin kesinlikle buna ihtiyacı vardır.
 - * volume aynı makinede de saklanabilir, cloud providerların bir storage servisinde de saklanabilir
- deployment
 - * pod, konteynerlar arası bir soyutlama yapıyorsa, deployment da podlar üstünden bir soyutlama yapan kubernetes konseptidir.
 - * statei ya da bir volume'e ihtiyaç duymayan uygulamalar için kullanılır
 - * genellikle direk podlar ile değil de deploymentlar ile çalışırız
- statefulset
 - * **sts** kısa ismi
 - * database uygulamaları için özel olarak geliştirilmiştir. **mysql, mongodb, elasticsearch**. bu gibi uygulamalar her zaman stateful sets ile deploy edilmelidir. deployment ile değil
 - * database uygulamalarının stateful set ile deploy etmek biraz uğraştırıcıdır. bu yüzden veritabanı gibi uygulamaları kubernetes clusteri dışında tutmak daha mantıklıdır. yani veritabanını yine deployment ile kurup asıl database bağlantısını dışardaki veritabanına göndermek en mantıklısıdır.
- kubernetes has 3 different pluggable components
 - CRI (container runtime interface)

- * containerd and cri-o
- * docker was removed
- CNI (container networking interface)
 - * k8s has k-proxy by default for cluster networking but there are different plugable services as well
 - * flannel
 - * calico
 - * cilium
- CSI (container storage interface)
 - * this is for storage in cluster

• DaemonSet

- **DaemonSet** her node'da çalışması gereken servislerin tanımlandığı bir deployment türüdür. Her bir node'da sistem düzeyinde çalışması gereken servisler için kullanılır. Örnek olarak; **log toplama** (Fluentd, Logstash), **node monitoring** (prometheus node exporter), **container network interface** (calico, flannel) ve **security agents** gibi "node" ile ilgili olan servisler için kullanılır. ¹
- Yeni bir node cluster'a eklenirse, DaemonSet otomatik olarak o node'da da bir Pod oluşturur. Node cluster'dan silinirse, ilgili Pod da silinir. İstersen sadece belirli label'lara sahip node'lara Pod yerleştirebilirsin (nodeSelector, affinity vb.). ¹
- Bir daemonset oluşturmak için de şu örneğe bakabilirsin **kubernetes daemonset file example**
- **DaemonSet ve Deployment Farkları** ¹

özellik	DaemonSet	Deployment
pod sayısı	her node'a 1 tane	belirli sayıda (replicaset)
node'a bağlılık	evet	hayır
usecase	sistem/altyapı	aplikasyon

• Kubernetes Volumes

- kubernetes does not give you data persistency out of the box ²
- **Persistent Volume**
 - * belongs to cluster itself. it is a dedicated disk
- **Persistent Volume Claim**
- **Storage Class**

¹ChatGPT

²Kubernetes Volumes explained | Persistent Volume, Persistent Volume Claim & S...

Flashcards

Resources

Docs

- [Kubernetes Documentation | Kubernetes](#)

Videos

- [Complete Kubernetes Course - From BEGINNER to PRO - YouTube](#)
- [Kubernetes Crash Course for Absolute Beginners {NEW} - YouTube](#)

Books

Articles