

de.NBI - Cloud Usermeeting 2021

Introduction to Kubernetes II: Deployments/Networking/Volumes

Sebastian Beyvers

`sebastian.beyvers@computational.bio.uni-giessen.de`

Marius Dieckmann

`marius.dieckmann@computational.bio.uni-giessen.de`

18 November 2021

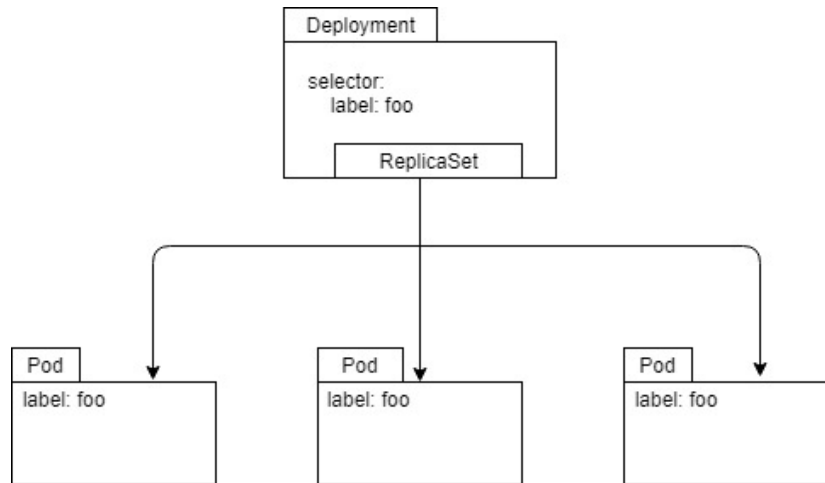




kubernetes

- The standard resource for regular, long-running services
- Build on top of replica sets
- Rule of thumb: Do not manage Pods/ReplicaSets created by Deployments directly
- Offers:
 - Replication
 - Rolling updates/Rollbacks
 - (Auto-)Scaling

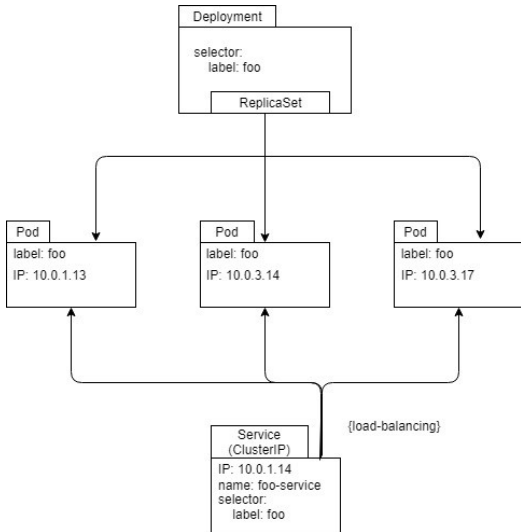
Deployments - Selectors



- Similar actions use the same configuration pattern
 - Containers in pods/deployments/jobs/... Are defined exactly the same → API cross references
- "Advanced features" for pods
 - Init-container (same as container, but runs beforehand)
 - Liveness probe
 - Imagepullpolicy
 - DNS config
 - Image pull secrets

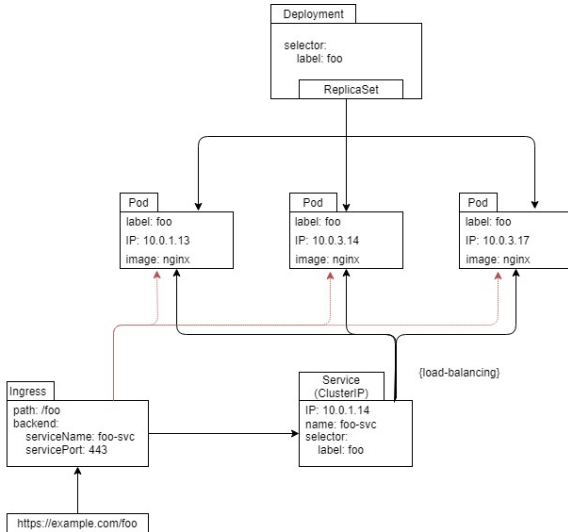
- From the Documentation: "...Pods are mortal. They are born and they die..." → pods do not have a stable IP
- Services group multiple pods with a single IP
 - Services use selectors to find pods
 - Non-selector services: Provide external endpoints
 - (optional) DNS-addons can create DNS entries for services
- Services have types:
 - ClusterIP (default): Expose a cluster internal IP
 - NodePort: Expose the service on a node externally
 - LoadBalancer: Expose the service externally using an external load balancer (Mainly public clouds)

Service - Selectors



- Ingress: most widely used option to route traffic into Kubernetes
 - Mostly used for HTTP(S) traffic (but TCP and UDP are also possible)
- "An Ingress is a collection of rules that allow inbound connections to reach the cluster services."
- Offers more advanced options compared to standalone services
 - Domain based routing
 - Path based routing
 - Automated TLS
- Multiple implementations available: nginx, Traefik,...

Ingress - Selectors



- "Pods are mortal" → data stored inside pods is not persisted
- Multiple pods might want to share a single data source
- Pods might be rescheduled and end up on a different node
- Docker: Volumes → local directories
 - Not suitable for k8s (pod rescheduling)
- Kubernetes provides volumes/storage via plugins
 - Details are dependent on the implementation (storage class)
 - Plugins can be provided by the k8s team or from external developers

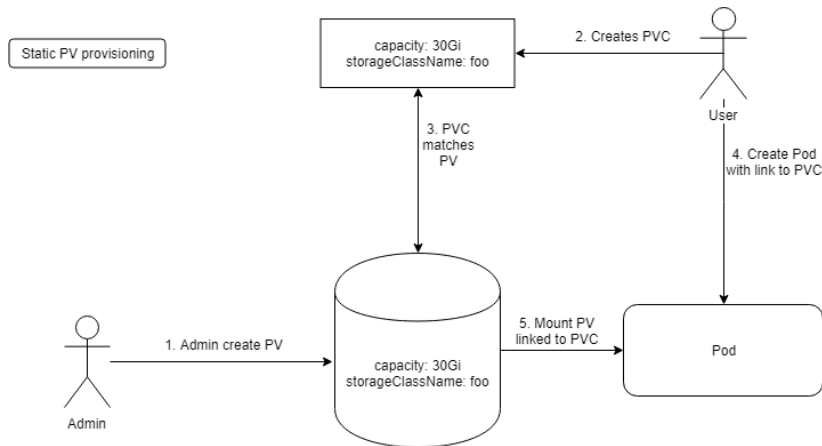
Storage Types

- Temporary storage:
 - emptyDir → empty directory, (fast) scratch storage
- Local storage:
 - hostPath → Mounts a path on the host into the container (similar to docker -v)
- Persistent (network) storage:
 - CephRBD
 - CephFS
 - NFS
 - Longhorn
 - Cinder
 - ...
- ReclaimPolicies: Retain, Delete, Recycle
- AccessModes: ReadWriteOnce, ReadOnlyMany, ReadWriteMany, (ReadWriteOncePod)

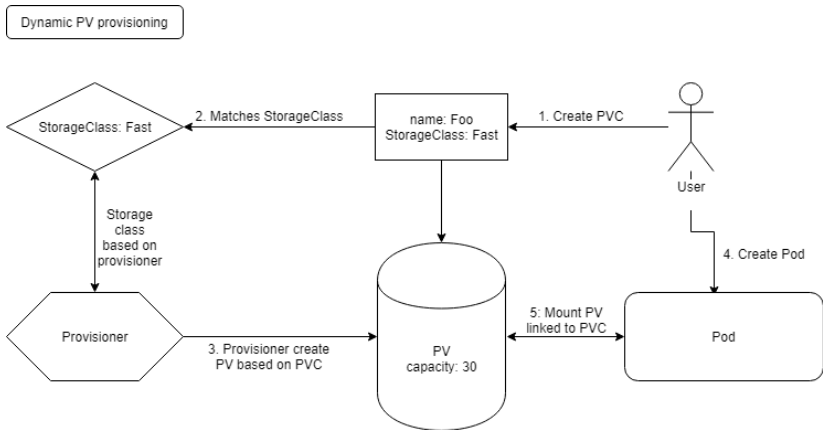
Storage organization

- Volumes
 - Externally mounted storage object
- Persistent volume (PV)
 - Storage object
 - 1. Static provisioning:
 - Created by an admin
 - 2. Dynamic provisioning:
 - Predefined storage classes
 - Managed by a provisioner
 - Provisioner create the volume
 - Many implementations → Names and properties are cluster dependent
- Persistent volume claim (PVC)
 - Storage request by a user
 - Satisfied by the provisioner → creation of PV

Static PV provisioning



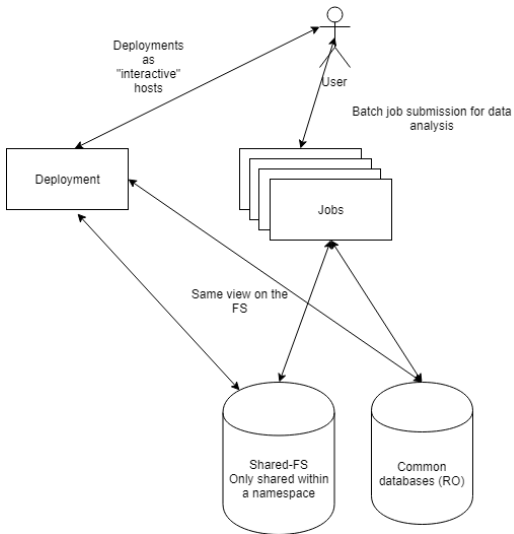
Dynamic PV provisioning



Additional storage options

- Use StatefulSets if pods should hold state
 - PVC template so each pod will get its own PV
- ConfigMap
 - Used to push key value pairs into your cluster
 - Can be used as persistent list of env-vars
 - Useful if you want to share configurations across multiple pods
- Secrets
 - Similar to ConfigMaps, but with "hidden" data
 - Used for usernames, passwords, keyfiles (certificates) etc.

Example: K8s as HTC replacement



Questions?