de.NBI Cloud Usermeeting - 2023

Introduction to Kubernetes II:
Deployments / Networking / Volumes

Sebastian Beyvers
sebastian.beyvers@cb.jlug.de

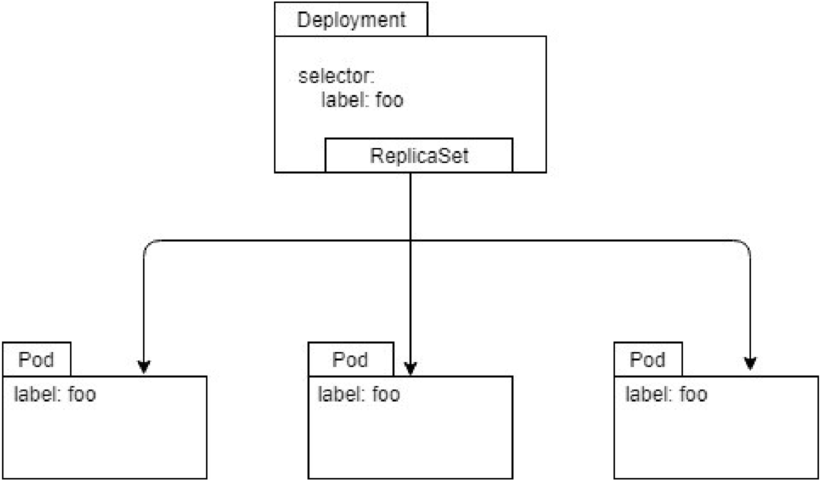
28 November 2023

Deployments

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: Inherited from ReplicaSets
  - Rolling updates/Rollbacks: Versioned upgrades
  - (Auto-)Scaling

Deployments - Selectors



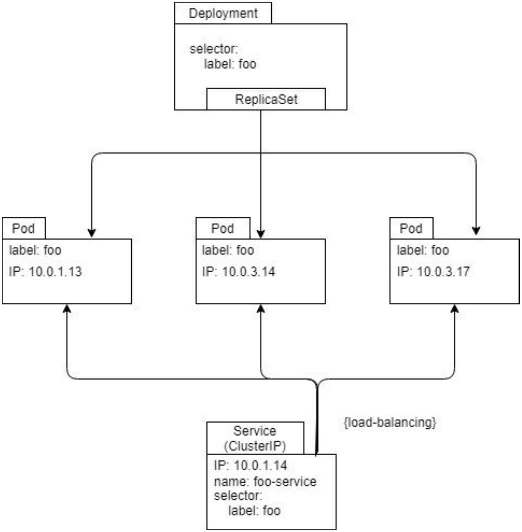
Container - Configuration

- Similar actions use the same configuration pattern
  - Containers in pods/deployments/jobs/... are defined exactly the same → API cross references / inheritance
- ”Advanced features” for pods
  - Init-container (container that runs beforehand)
  - Liveness & startup probe: Is my service alive ?
  - Imagepullpolicy: When should an image be pulled ?
  - DNS config: Which network should be preferred ?
  - Image pull secrets: Private container registries

Services

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

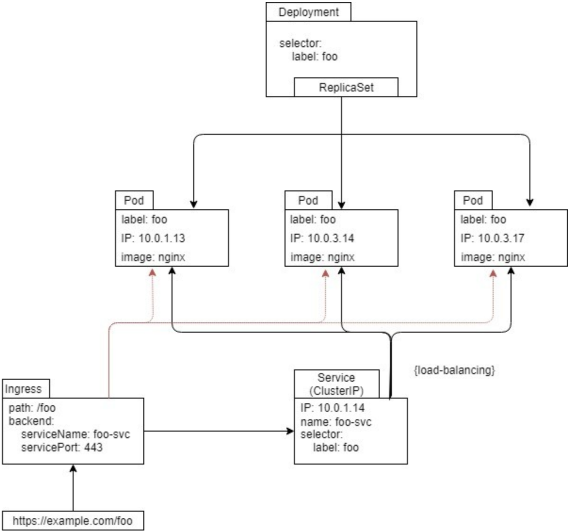
Service - Selectors



Ingress

- Ingress: most widely used option to route traffic into Kubernetes
  - Used for HTTP(S) traffic (but TCP and UDP are also possible)
  - May be partially replaced by the API Gateway concept
- ”An Ingress is a collection of rules that allow inbound connections to reach the clusters services.”
- Offers more advanced options compared to standalone services
  - Domain based routing: e.g. example.com
  - Path based routing: e.g. example.com/foo
  - Automated TLS via addons: https://example.com/foo
- Multiple implementations available: nginx, Traefik,...

Ingress - Selectors



Storage

- ”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 are provided by the k8s team or from external developers

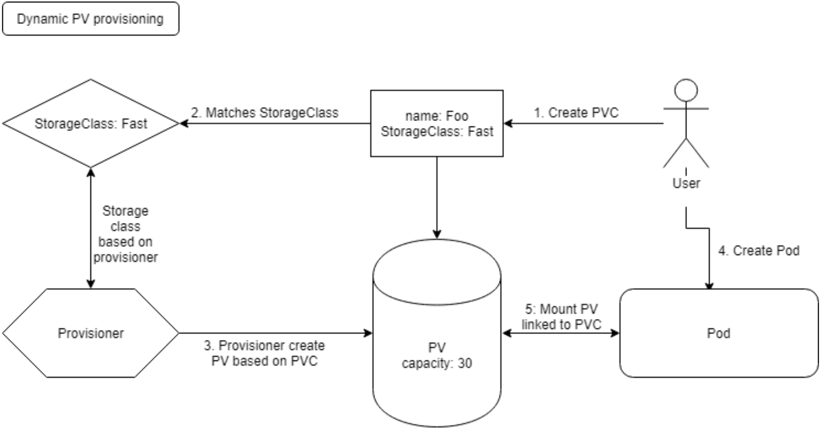
Storage Types

- 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
    - Provisoner 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

Dynamic PV provisioning



Additional storage options

- Use StatefulSets if pods should hold state
  - PVC templates can be used so that each pod will gets its own PV
- ConfigMap
  - Used to push key value pairs into your cluster
  - Can be used as persistent list of env-vars or (config) files!
  - 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.

Questions?