Scheduling (S3)

Manual scheduling

- Scheduler searches for Pods without nodeName
    - Selects best node and sets the nodeName for the Pod
    - nodeName can be set manually in definition file
    - another way is with pod binding object (via definition yaml) and send POST request

Labels and Selectors

- labels: properties attached to items
    - added in definition file under metadata
- selectors: filters based on labels
    - *kubectl get <object-type> --selector <label>=<label-value>*
    - also used in replicasets, deployments, services, etc.
- annotations: additional information added to a object
    - added under metadata

Taints and Tolerations

- used to set restrictions to schedule which pods can go on which nodes
    - will not guarantee that a pod goes on a certain node
- Nodes have Taints
    - By default no taints on worker nodes
    - Master node has a default taint
    - *Kubectl taint nodes <node-name> key=<value>:<taint-effect>*
- Pods can have Tolerations
    - Default no tolerations
    - Specified in definition file under spec→tolerations

Node Selectors

- Limitation for pods to specify nodes
    - In pod definition file under spec > nodeSelector
    - Works with labels on nodes

Node Affinity

- To ensure pods are hosted on particular nodes
    - Specified under spec > affinity > nodeAffinity in pod definition
    - More complex than node selectors, also more options
- requiredDuringSchedulingIgnoredDuringExecution
    - node has to match or pod will not be scheduled
    - affinity ignored during execution
- preferredDuringSchedulingIgnoredDuringExecution
    - matching node is preferred but other nodes are used either
    - affinity ignored during execution
- requiredDuringSchedulingRequiredDuringExecution
    - node has to match or pod will not be scheduled
    - node has to match or pod will not be scheduled during execution

Taints and Tolerations vs Node Affinity

- for some cases taints and node affinity has to be used in combination

Resource Requirements and Limits

- scheduler taks resources in consideration for deploying pods
    - if not enough resources in any node POD does'nt get deployed
- Default Requests: 0.5 CPU (1=1vCPU), 256 Mi Memory, disk
    - Can be specified in deployment definition under spec > resources > requests
- Default Limits: 1 CPU, 512 Mi
    - Can be specified in deployment definition under spec > resources > limits
    - Limits can't be exceeded for CPU

DeamonSets

- Like replica sets but runs one copy of a POD on each node of the cluster
- Good for monitoring, logging, kube-proxy, networking, etc.
- Similar to replicaset definition just kind is different
- Uses default scheduler and nodeAffinity

Static Pods

- Nodes can be managed independently with kubelet
    - Pod definitions have to be placed under /etc/kubernetes/manifests
        - Path is option of the service pod-manifest-path or in kubeconfig.yaml
        - Only pods can be created like this
    - Kubelet tries to keep pods alive
    - Pods can be viewd with docker ps

Multiple Schedulers

- You can deploy your own scheduler
    - Can be specified with definition yaml too
- Cluster can have multiple schedulers
    - You can specifiy the scheduler when deploying a Pod
        - In the definition under spec > schedulerName
- View events with kubectl get events
- Kubectl logs <scheduler-name> --name-space<namespace>