Applied Containers & Orchestration

Topics

1. Low level container building blocks
    a. Container vs Virtual Machine
    b. Namespaces
        i. ipc
        ii. mnt
        iii. net
        iv. pid
        v. time
        vi. user
        vii. uts
    c. Cgroups
    d. What is a container runtime?
    e. What does docker do?
    f. What does containerd do?
    g. Images/Containers/Volumes/Registries - what's the difference?
    h. Docker commands you should know
        i. Docker info
        ii. Docker run
        iii. Docker network
        iv. Docker volume
        v. Docker image
        vi. Docker * inspect
    i. Linuxkit
    j. Docker-Compose
        i. Services
        ii. Networking
        iii. Volumes
        iv. What’s coming in version 2
2. Building Images
    a. Buildkit
    b. Layers
    c. “Copy on Write” and how containers start so fast
    d. Image Layer Caching
    e. Storage Drivers & their differences
        i. vfs
        ii. btrfs / zfs
        iii. overlay2
    f. Writing better dockerfiles
        i. Optimizing builds for caching

        ii. Spotting common mistakes
    g. Internals of the final image
        i. What makes up a container image?
        ii. Other than the filesystem, what is attached?
        iii. Which information is purely metadata?
3. Sidecars, Ambassadors, and Adapters
    a. What’s a sidecar container?
    b. How can sidecars be used to modernize existing applications?
        i. SSL termination
        ii. Http static caching / acceleration
        iii. Metrics aggregation
    c. Ambassador containers for connecting to external services
        i. Abstracting complex service APIs
    d. Adaptor containers for converting and exporting
        i. Healthchecks / Metrics
    e. Caching existing / request deduplication
        i. Three stooges problem
        ii. Simple response etag caching
        iii. Request deduplication
4. Container networking
    a. Bridge networking
    b. Overlay networking
    c. Host networking
    d. How does iptables play in?
    e. Security within docker networks
    f. How does dns work in containers?
5. Container networking 2
    a. What is a CNI?
    b. Calico vs Cilium
    c. How does networking between machines in a cluster work?
    d. Networking policies - how to secure a network
    e. Tricks for networking policies
        i. Cilium Editor
        ii. Minikube gotchas
    f. Micro services for spreading load
6. Container Security
    a. What is container isolation
    b. Trusted images
    c. Cgroups again
    d. Don’t run root unless you need to
    e. Linux Capabilities instead of root
    f. What does a privileged container have access to?
    g. What if your sidecar gets popped?
    h. How to handle tokens/passwords/credentials

   i. Leaving sensitive things in image layers
   j. Thinking through attack surface of and single instance
   k. Network isolation
   l. Limiting blast radius once a service is popped
7. Kubernetes 1
   a. Components that make up control plane
      i. Etcd
      ii. Apiserver
      iii. Scheduler
      iv. Control Manager
   b. Kubelet & Kube Proxy
   c. Namespaces
   d. Kubernetes yaml and resource intro
      i. General format for files
      ii. Versioning in yaml vs Versioning of k8s
   e. Pods vs Containers
      i. What's shared between containers in a pod
      ii. How do those sidecars work
      iii. Resource Requests & Limits
      iv. Image Pull Policy
      v. Security Context
   f. Deployments
   g. Minikube for debugging
      i. How to start a cluster
      ii. What can you do in minikube?
      iii. What can't you do in minikube?
   h. Kubernetes Dashboard
      i. k9s
   i. Metrics Server
8. Kubernetes 2
   a. Healthchecks / Readiness probes for advanced serving and rollouts
   b. Services
      i. Multiple services behind a single dns/ip address
      ii. Where in the networking stack do LBs live
   c. DaemonSets
   d. Jobs
   e. Cronjobs
   f. Services
   g. Secrets
   h. ConfigMaps
   i. Labels
   j. Service Accounts
      i. RBAC
      ii. Tokens in pods

        iii. Image Pull Secrets

9. Kubernetes 3
    a. What's the difference between Deployments, StatefulSets, ReplicaSets?
    b. How does the container registry fit in?
    c. Node affinity
    d. Annotations
    e. CRD - Custom Resource Definitions
    f. Readiness/Liveness/Startup probes
    g. Service Load Balancers
    h. Self Hosted vs Cloud Provider
        i. Which managed k8s should you use?
    i. Rollouts
        i. Surge update
        ii. Max unavailable
        iii. Startup probes for safety
        iv. Revision history
        v. Undoing a rollout
10. Kubernetes 4
    a. Vertical vs Horizontal Scaling
    b. Horizontal Pod Autoscalers
        i. Pod resource requests
        ii. Metrics Servers
        iii. Versioning for definitions
        iv. Tuning
    c. Provisioning new nodes when resources run out
    d. Services + Load Balancing + Autoscaling
    e. Scaling nodes vs HPAs
        i. Scaling nodes varies between hosting providers
    f. Optimize your containers for faster scaling
    g. Helm charts
        i. Installing your infrastructure
        ii. Helm templating
        iii. Values file
11. Kubernetes 5
    a. Taints
    b. Cordoning a node
    c. Draining a node
    d. Resource pressures
    e. Prometheus & Grafana
    f. Node Exporter
    g. Time Series metrics
    h. Alerting on pressures
    i. Building basic dashboards