KUBERNETES V1.22 WHAT YOU NEED TO KNOW

SENTHIL RAJA CHERMAPANDIAN

SPEAKER INTRO

Senthil Raja Chermapandian

Principal Software Engineer, Ericsson

Architecting Cloud-native AI/ML Platforms

Organizer: KCD Chennai 2022 (kcdchennai.in)

Maintainer: github.com/senthilrch/kube-fledged

Tech blogger: medium.com/@senthilrch

Twitter: @senthilrch

LinkedIn: linkedin.com/in/senthilrch

NOTABLE FEATURES/ENHANCEMENTS

- The kubernetes release cadence was officially changed from four to three releases yearly.
- This is the first longer-cycle release related to that change.
  - Removal of several deprecated beta APIs
  - API changes and improvements for ephemeral containers
  - A new podsecurity admission alpha feature is introduced, intended as a replacement for podsecuritypolicy
  - A new alpha feature to enable API server tracing
  - As an alpha feature, all kubernetes node components (including the kubelet, kube-proxy, and container runtime) can be run as a non-root user
  - External credential providers now stable
  - Quality of service for memory resources
  - Node system swap support

REMOVAL OF SEVERAL DEPRECATED BETA APIS

- The **v1.22** release will stop serving the API versions we've listed immediately below. These are all beta apis that were previously deprecated in favor of newer and more stable API versions.
  - Beta versions of the validatingwebhookconfiguration and mutatingwebhookconfiguration api (the **admissionregistration.k8s.io/v1beta1** API versions)
  - The beta customresourcedefinition API (**apiextensions.k8s.io/v1beta1**)
  - The beta apiservice API (**apiregistration.k8s.io/v1beta1**)
  - The beta tokenreview API (**authentication.k8s.io/v1beta1**)
  - Beta API versions of subjectaccessreview, localsubjectaccessreview, selfsubjectaccessreview (API versions from **authorization.k8s.io/v1beta1**)
  - The beta certificatesigningrequest API (**certificates.k8s.io/v1beta1**)
  - The beta lease API (**coordination.k8s.io/v1beta1**)
  - All beta ingress apis (the **extensions/v1beta1** and **networking.k8s.io/v1beta1** API versions)

KUBECTL CONVERT

- There is a plugin to kubectl that provides the kubectl convert subcommand. It's an official plugin that you can download as part of kubernetes. See download kubernetes for more details.
- You can use kubectl convert to update manifest files to use a different api version. For example, if you have a manifest in source control that uses the beta ingress API, you can check that definition out, and run kubectl convert -f <manifest> --output-version <group>/<version>. You can use the kubectl convert command to automatically convert an existing manifest.
- For example, to convert an older ingress definition to networking.k8s.io/v1, you can run:
  - kubectl convert -f ./legacy-ingress.Yaml --output-version networking.k8s.io/v1

API CHANGES AND IMPROVEMENTS FOR EPHEMERAL CONTAINERS

- Ephemeral containers are useful for interactive troubleshooting when kubectl exec is insufficient because a container has crashed or a container image doesn't include debugging utilities.
- In particular, distroless images enable you to deploy minimal container images that reduce attack surface and exposure to bugs and vulnerabilities. Since distroless images do not include a shell or any debugging utilities, it's difficult to troubleshoot distroless images using kubectl exec alone.
- When using ephemeral containers, it's helpful to enable process namespace sharing so you can view processes in other containers.

A NEW PODSECURITY ADMISSION ALPHA FEATURE IS INTRODUCED

- Pod security policy is deprecated as of kubernetes v1.21. There were numerous problems with PSP that lead to the decision to deprecate it, rather than promote it to GA
- Replace podsecuritypolicy with a new built-in admission controller that enforces the pod security standards.
- Policy enforcement is controlled at the namespace level through labels
- Policies can be applied in 3 modes. Multiple modes can apply to a single namespace.
  - Enforcing: policy violations cause the pod to be rejected
  - Audit: policy violations trigger an audit annotation, but are otherwise allowed
  - Warning: policy violations trigger a user-facing warning, but are otherwise allowed
- An optional per-mode version label can be used to pin the policy to the version that shipped with a given kubernetes minor version (e.g. v1.18)

A NEW ALPHA FEATURE TO ENABLE API SERVER TRACING

- Along with metrics and logs, traces are a useful form of telemetry to aid with debugging incoming requests.
- The API server currently uses a poor-man's form of tracing (see github.com/kubernetes/utils/trace)
- Butwe can make use of distributed tracing to improve the ease of use and enable easier analysis of trace data.
- Since this feature is for diagnosing problems with the kube-api server, it is targeted at cluster operators and cloud vendors that manage kubernetes control-planes.

KUBELET-IN-USERNS (AKA ROOTLESS MODE)

- This feature allows running the entire Kubernetes components (kubelet, CRI, OCI, CNI, and all kube-*) as a non-root user on the host, by running them in a user namespace.
- Protect the host from potential container-breakout vulnerabilities. This is the main motivation.
- Allow users of shared machines to run Kubernetes without the risk of accidentally breaking their colleagues' environments. **Not recommended for real multi-tenancy where the users cannot be trusted.**
  - Safe kind: Kubernetes inside Rootless Docker/Podman.
  - Safe Kubernetes-on-Kubernetes, to isolate workloads more strictly than Kubernetes API namespaces.

EXTERNAL CREDENTIAL PROVIDERS NOW STABLE

- **k8s.io/client-go** and tools using it such as kubectl and kubelet are able to execute an external command to receive user credentials.
- This feature is intended for client side integrations with authentication protocols not natively supported by k8s.io/client-go (LDAP, Kerberos, OAuth2, SAML, etc.).
- The plugin implements the protocol specific logic, then returns opaque credentials to use.
- Almost all credential plugin use cases require a server side component with support for the webhook token authenticator to interpret the credential format produced by the client plugin.

QUALITY OF SERVICE FOR MEMORY RESOURCES

- In cgroup v1, and prior to this feature, the container runtime never took into account and effectively ignored **spec.containers[].resources.requests["memory"]**.
- Because there is no way to throttle memory usage, if a container goes past its memory limit it will be terminated by the kernel with an OOM (Out of Memory) kill.
- Memory QoS uses the memory controller of cgroup v2 to guarantee memory resources in Kubernetes.
- Memory requests and limits of containers in pod are used to set specific interfaces memory.min and memory.high provided by the memory controller.
- When memory.min is set to memory requests, memory resources are reserved and never reclaimed by the kernel
- This is how Memory QoS ensures the availability of memory for Kubernetes pods.

NODE SYSTEM SWAP SUPPORT

- In prior releases, kubernetes did not support the use of swap memory on linux
- As part of kubernetes' earlier design, swap support was considered out of scope
- A kubelet would by default fail to start if swap was detected on a node.
- Swap configuration on a node is exposed to a cluster admin via the memoryswap in the kubeletconfiguration.
- As a cluster administrator, you can specify the node's behaviour in the presence of swap memory by setting memoryswap.Swapbehavior.
- This is possible through the addition of a memory_swap_limit_in_bytes field to the container runtime interface (cri).

RESOURCES

- Kubernetes 1.22: reaching new peaks | kubernetes
- Kubernetes API and feature removals in 1.22: here's what you need to know | kubernetes
- Authenticating | kubernetes (external credential plugin)
- Quality-of-service for memory resources | kubernetes
- New in kubernetes v1.22: alpha support for using swap memory | kubernetes
- Debug running pods | kubernetes (EPHEMERAL CONTAINERS)
- Pod security admission | kubernetes
- Alpha in kubernetes v1.22: API server tracing | kubernetes
- Running kubernetes node components as a non-root user | kubernetes

THANK YOU

SENTHIL RAJA CHERMAPANDIAN