KubeCon | CloudNativeCon

North America 2018

Deep Dive: Container Identity Working Group

Greg Castle
Container Identity WG lead
Twitter: @mrgcastle
Github: @destijl
Google

Mike Danese
Sig-auth chair
Twitter, Github:@mikedanese
Google

Container Identity Working Group

- Formed Aug 2017
- Turned down Oct 2018
- Folded back into sig-auth
  - Didn’t need the extra meeting
- Definitely not done with identity work
- WGs not meant to live forever
- Today:
  - Review use cases and current solutions
  - Deep dive on K8s service account tokens for workload identity
  - Future work discussion

Use Cases

- User identity
  - **Enterprise:** Manage user ID in LDAP/AD/Proprietary

- Workload identity inside cluster
  - **Enterprise:** Manage service ID centrally
  - **Secrets:** API key in Hashicorp Vault
  - **Multi-cloud:** Google ID for BigQuery, Amazon ID for S3, etc.
  - **Service-Service:** Intra + inter cluster
  - **Container:** Monitoring sidecar needs different ID to workload

- Workload identity outside cluster
  - **CI/CD:** Auth to K8s control plane

User Identity

User Identity Solutions

- Manage user ID in LDAP/AD/Proprietary. One src of truth
- Common solutions:
  - Federate using OIDC
  - Custom authenticating proxy, set user impersonation headers
  - Custom authenticator and exec-based auth provider (kubernetes/pull/59495 needs docs)
- Also available:
  - x509 client certs
  - passwords
  - static tokens
- Anti-pattern: users sharing K8s service accounts

Workload Identity

Goals

- Get ID to apps with ≈ 0 developer effort
- Provision ID for multiple external systems
- Segmentation: minimum blast radius for compromise
- Limited lifetime, auto-rotation
- Non-exportable where possible (e.g. TPM available)

External efforts (not covered here)

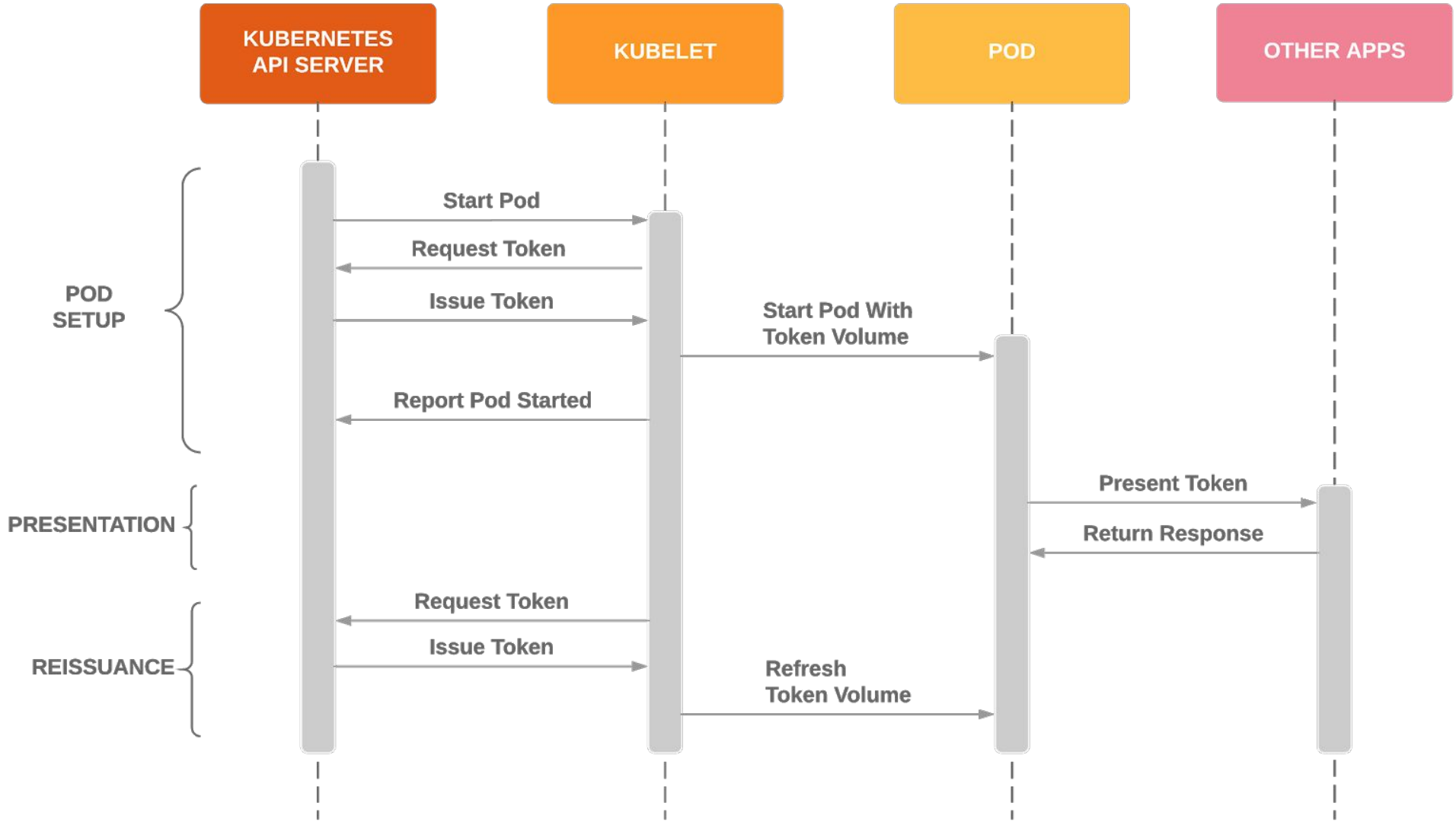
- SPIFFE and SPIRE (spiffe.io)
  - Application x509 ID and standard naming scheme, API for workloads to access ID, runtime env for attestation, rotation
- Istio (istio.io)
  - Lots of service management features. Identity: SPIFFE-named x509 certs to identify services
- Vault integration (goo.gl/ZuAPtn)
  - TODO: scope K8s SA tokens to vault


Workload Identity: Kubernetes Service Accounts

K8s Service Accounts: 2017

- Forever-tokens managed via secrets
- Token leak rotation pain (e.g. for goo.gl/EN6kff)
- Permissive file permissions 0644
- No audience binding

“New” K8s Service Accounts



K8s Service Accounts: 2018

- Audience bound
- Limited duration
- Forced rotation is easy: restart pod
- No central storage
- Kubelet handles reissuance
- K8s client library will handle re-read on token files
  - client.go already done

Changes visible to pod

“BoundServiceAccountTokenVolume”

- Need to re-read (but client lib will do it for you)
- Different permissions from 0644 -> 0600
- File-only injection: no environment variables
- PodSecurityPolicy now requires allowing projected volumes

Service account volume available at the same mount point

Migration Plan

kubernetes/kubernetes#70679

Rough timeline

- Announce alpha for testing: 1.13
  - Opt-in with cluster-level flag
- Beta: 1.15-ish
- GA 1.16-ish

Call to action

Watch this issue for docs to explain the feature:
kubernetes/kubernetes#70679

K8s distributors, Identity integrators:

- Try out these new APIs, give feedback

Security teams:

- Communicate to devs k8s service account changes coming

Developers that use SAs to interact with K8s API:

- Try out your application with “new” style service accounts

Future Work

Future Work

- Node component authn/z
  - kubernetes/kubernetes#62747
- Server authentication for cluster workloads (k8s and addons)
  - Solved by Istio, SPIRE etc.
  - What if you don’t want the dependency?
  - kubernetes/kubernetes#63732
- Non-exportable credentials

KubeCon | CloudNativeCon

North America 2018

Thanks!
Questions?

Greg Castle
Container Identity WG lead
Twitter: @mrgcastle
Github: @destijl
Google

Mike Danese
Sig-auth chair
Twitter, Github:@mikedanese
Google