Container security: best practices & roadmap

October 2019

Google Cloud

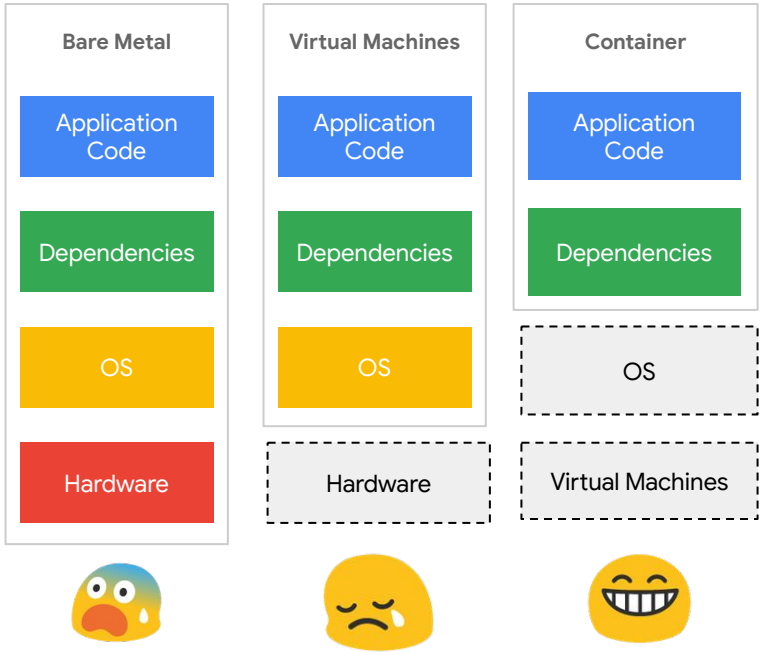
Container security 101

Google Cloud

First, what's a container?

Containers are all the rage these days, getting us close to the dream of write once, run anywhere.

They package and isolate applications and dependencies.



Google Cloud

What is Kubernetes?

- A portable, open-source, **container-centric** management platform
- Built-in primitives for **deployments, rolling upgrades, scaling, monitoring, and more**
- Inspired by **Google's internal systems**
- Get true **workload portability** and increased **infrastructure efficiency**

Containers come with security advantages inherent in their architecture

Containers are short lived and frequently re-deployed; **you can constantly be patching**.

Containers are intentionally immutable; a modified container is a **built-in security alert**.

Good security defaults are one line changes; **setting secure configurations is easy**.

With isolation technologies, **you can increase security without adding resources**.

Google Cloud

Container security threats & risks

INFRASTRUCTURE SECURITY

- Privilege escalation
- Credential compromise
- Kubernetes API compromise
- Over-privileged users

SOFTWARE SUPPLY CHAIN

- Unpatched vulnerability
- Supply chain vulnerability
- Zero day exploit on common library

RUNTIME SECURITY

- DDoS
- Node compromise and exploit
- Container escape
- Flood event pipeline

What is container security?

INFRASTRUCTURE SECURITY

Is my infrastructure **secure for developing** containers?

- How can I use Kubernetes security features to protect my identities, secrets, and network?
- How can I use native GCP functionality, like IAM, audit logging, and networking?

SOFTWARE SUPPLY CHAIN

Is my container image **secure to build and deploy?**

- How can I make sure my container images are vulnerability-free?
- How can I make sure the images I built aren't modified before they are deployed?

RUNTIME SECURITY

Is my container **secure to run?**

- How can I identify a container acting maliciously in production?
- How can I take action to protect and isolate my workload?
- How can I securely scale my containers deployment?

Kubernetes security features

Namespaces

Segment cluster resources

RBAC

Control access to cluster resources

NetworkPolicy

Limit pod to pod traffic

Pod Security Policy

Limit what permissions pods have

Secrets

Provide pods access to application secrets

Audit Logging

Capture operations to the Kubernetes API server

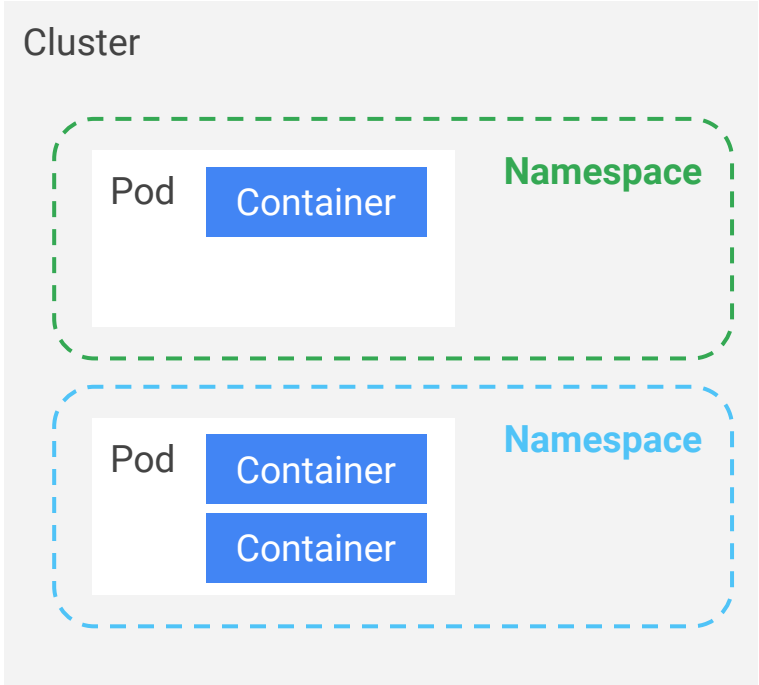
K8S: Kubernetes namespaces

Problem: I have too much stuff!

- Name collisions in the API
- Poor isolation between users
- Don't want to expose sensitive data

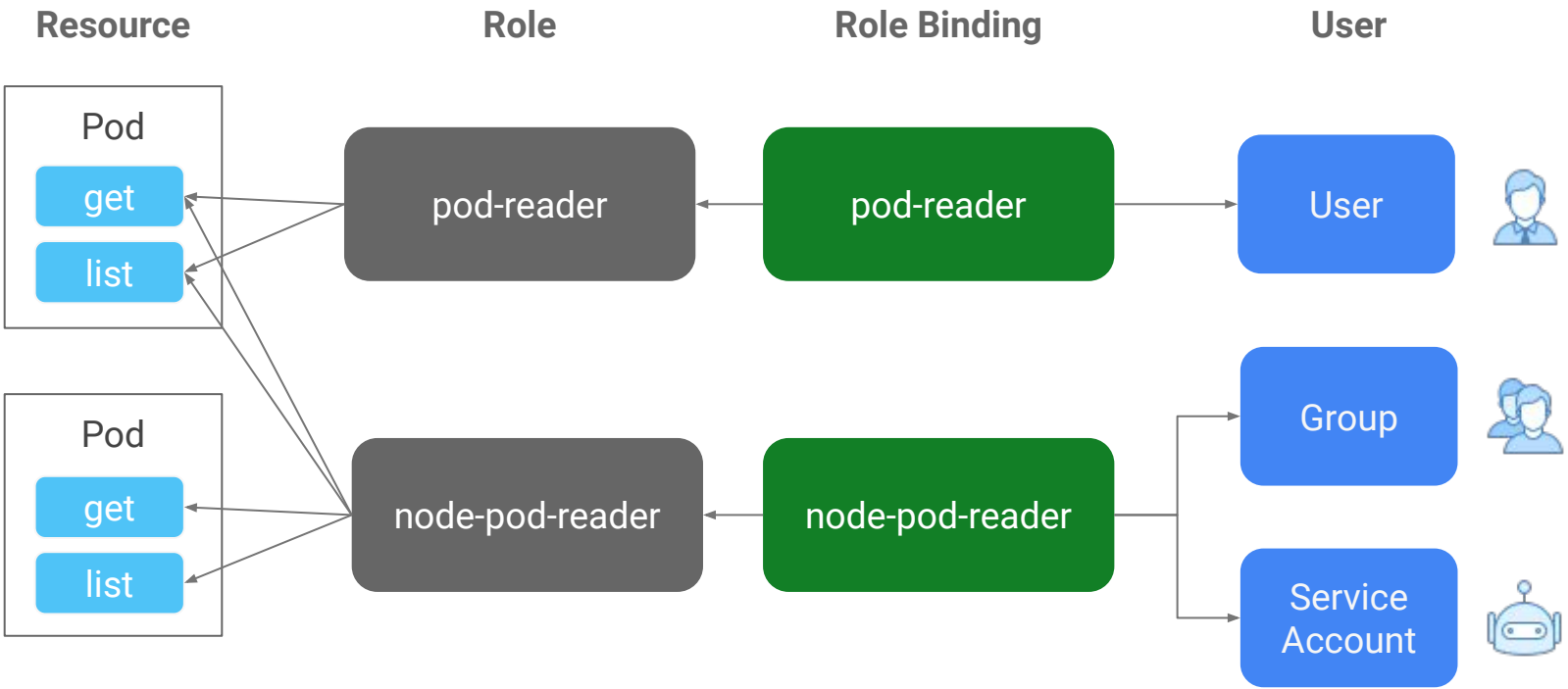
Solution: Slice up the cluster

- Create new namespaces as needed
  - per-user, per-app, per-department, etc.
- Namespaces are just another API object
- Provide each user community with its own:
  - Resources, with consumption limits
  - Policies, with delegated management



Google Cloud

K8S: RBAC

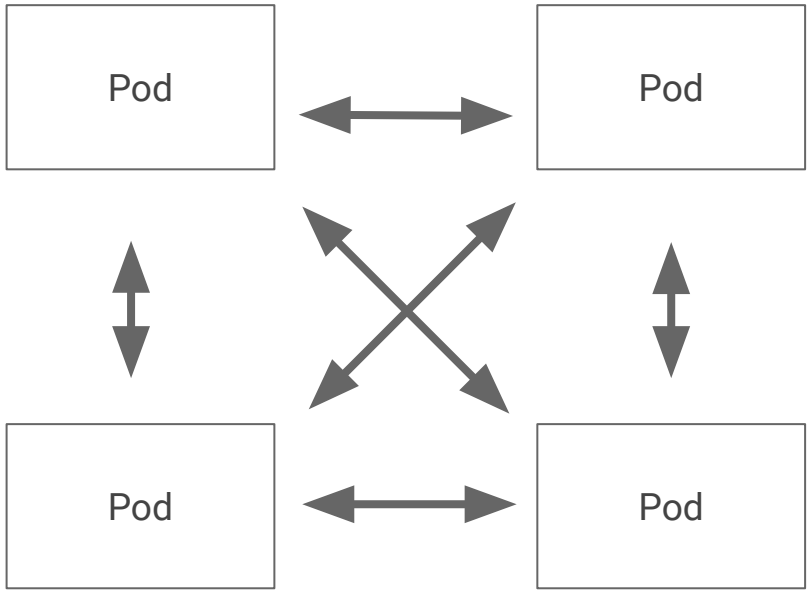


Google Cloud

K8S: NetworkPolicy

How does K8S networking work?

- IPs are per pod, **scoped to cluster**
- **Pods can reach each other directly**, without NAT, even across nodes



Google Cloud

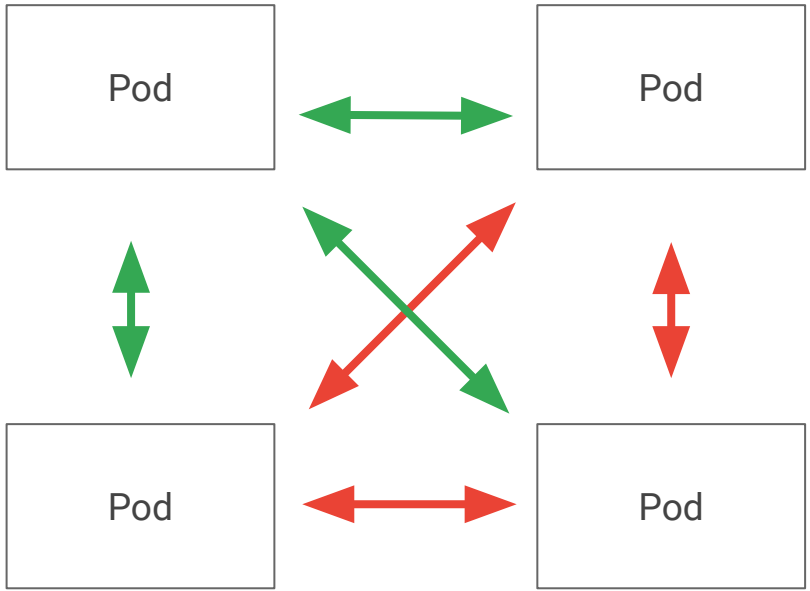
K8S: NetworkPolicy

How does K8S networking work?

- IPs are per pod, **scoped to cluster**
- **Pods can reach each other directly**, without NAT, even across nodes

With NetworkPolicy:

Restrict pod-to-pod traffic



Google Cloud

Best practices and current features

Google's container security philosophy

Prioritize security by default
We believe in giving users the power of Kubernetes without making them security experts

Integrate the best of Google Cloud Platform security
Integrations with IAM, Audit Logging, VPCs, encryption by default, node auto-upgrades, and the Cloud Security Command Center

Take advantage of a strong security culture
From our dedicated infra security team to security design requirements, security is an integral part of Google's culture, products, and operations

Google Cloud

Container security products & features

Google Cloud

INFRASTRUCTURE SECURITY

Best practices by default

RBAC, disabled Kubernetes dashboard, and hardening guide

Minimal OS

Hardened, purpose-built host OS based on Chromium

Node auto-upgrade

Automatic Kubernetes version upgrades for nodes, including security patches

Private clusters & authorized networks

Private IPs for your nodes, and restricted IP access to masters

Managed SSL certs BETA

HTTPs load balancers with automatically configured SSL certificates

Workload identity BETA

Google IAM service accounts for pods to authenticate outside the cluster

Shielded GKE nodes BETA

Verify boot integrity and secure kubelet credentials using vTPMs

Customer-managed encryption keys BETA

Encryption of persistent disks with a key in Cloud KMS

Application-layer secrets encryption BETA

Manage encryption of secrets in Kubernetes using a key in Cloud KMS

SOFTWARE SUPPLY CHAIN

Container Registry Vulnerability Scanning

Automatic scans of images and packages for known CVEs

Binary Authorization

Verification of provenance and blocking deployments that do not meet necessary requirements

Managed base images

Base images for common distributions patched for security vulnerabilities automatically by Google

RUNTIME SECURITY

Container runtime security partners

Aqua, Capsule8, StackRox, Sysdig and Twistlock integrations with Cloud SCC

GKE Sandbox BETA

Workload isolation at the pod level using hypervisor-like technology

Pod Security Policy BETA

Pod capability constraints, including Linux constructs like seccomp

GKE: Node auto-upgrade

Node upgrades help you keep your nodes up to date with the latest stable version of Kubernetes

- When a security patch is rolled out to nodes, GKE automatically patches your node and keeps it up to date
  - Security patch is rolled into the first possible release after public disclosure
  - Auto-upgrade is typically rolled out 2-4 weeks after update, depending on the severity of the vulnerability
- Available for both COS and Ubuntu nodes
- You can also schedule a maintenance window to control when maintenance happens

https://cloud.google.com/kubernetes-engine/docs/concepts/node-auto-upgrades

GKE: Minimal OS

Container-optimized OS (COS) based on Chromium OS, and maintained by Google

- **Built from source**: Since COS is based on Chromium OS, Google maintains all components and is able to rebuild from source if a new vulnerability is discovered and needs to be patched
- **Smaller attack surface**: Container-Optimized OS is purpose-built to run containers, has a smaller footprint, reducing your instance's potential attack surface
- **Locked-down by default**: Firewall restricts all TCP/UDP except SSH on port 22, and prevents kernel modules. Root file system is mounted read-only
- **Automatic Updates**: COS instances automatically download weekly updates in the background; only a reboot is necessary to use the latest updates. Google provides patches and maintenance

https://cloud.google.com/container-optimized-os/

GKE leverages Google Cloud security features

Identity & Access Management

Manage your users and permissions at the project level

This doesn't stop you from using Kubernetes RBAC - but lets you more easily manage this across projects

Cloud Audit Logging

Review, monitor and alert on audit logs centrally

This makes Kubernetes audit logs easier to consume along with all your other audit logs

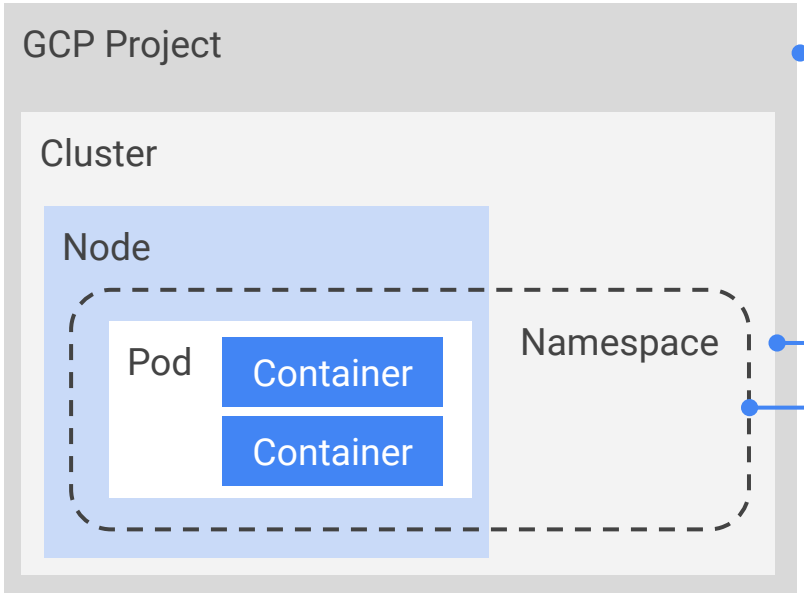
Virtual Private Cloud

Connect your cloud resources to each other on a private network

This lets you plug in Kubernetes into your broader network, the way you want to

Google Cloud

GKE: Using IAM and RBAC



Use IAM at the project level

Set roles for

- Cluster Admin: manage clusters
- Container Developer: API access within clusters

Use RBAC at the cluster and namespace level

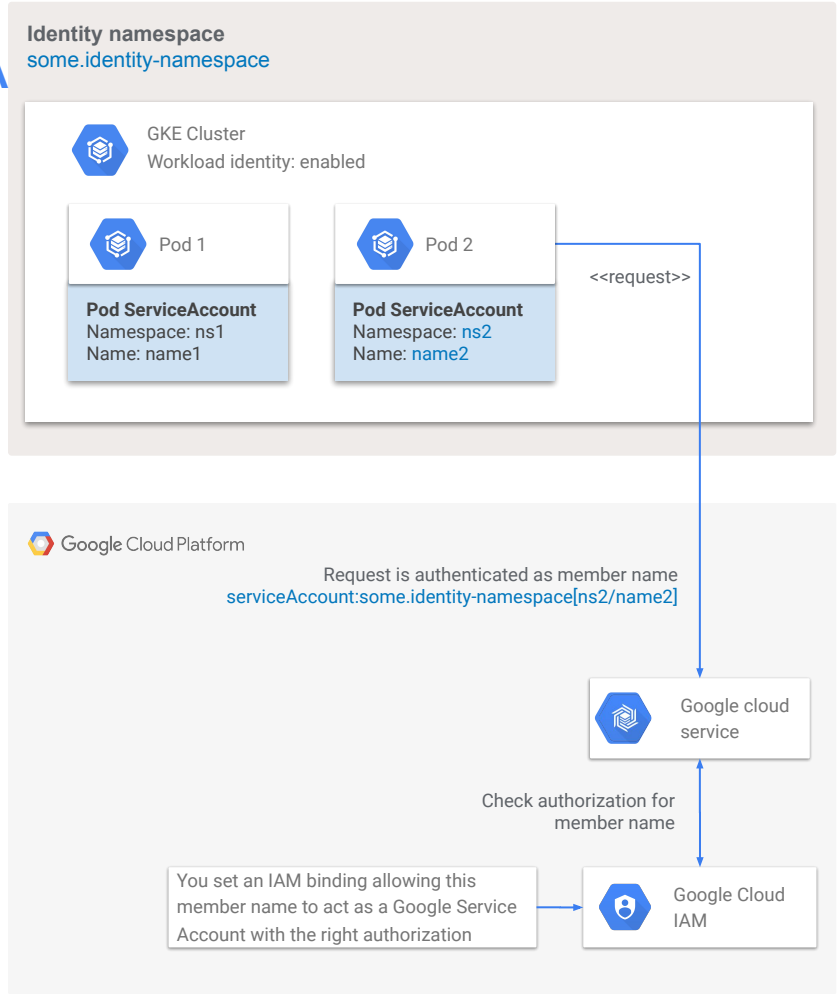
Set permissions on individual clusters and namespaces

Google Cloud

GKE: Workload identity BETA

Access Google Cloud services from Google Kubernetes Engine without managing any keys or secrets at all

- New IAM primitive, 'identity namespace,' enables Google cloud platform to authenticate Kubernetes Service Accounts directly
  - Use IAM to grant the desired authorization for a kubernetes service account
  - Google manages the identity namespace credentials; no rotation burden
  - Each workload only has minimal, necessary permissions
- Easy to assign identity and prove it to Google services

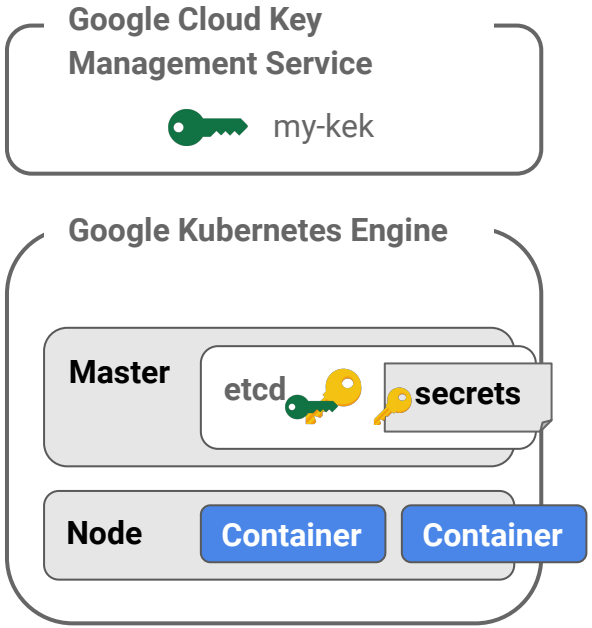


Google Cloud

GKE: Shielded GKE nodes BETA

- **Crypto anchors with vTPMs** offers isolation of secrets bound to a TPM, and exfiltration resistance
- How does it work?
  - Node Identity will be attested to ensure no arbitrary machine can join a cluster as a node
  - Any secret can be bound to a node (vTPM) and will not be globally shared

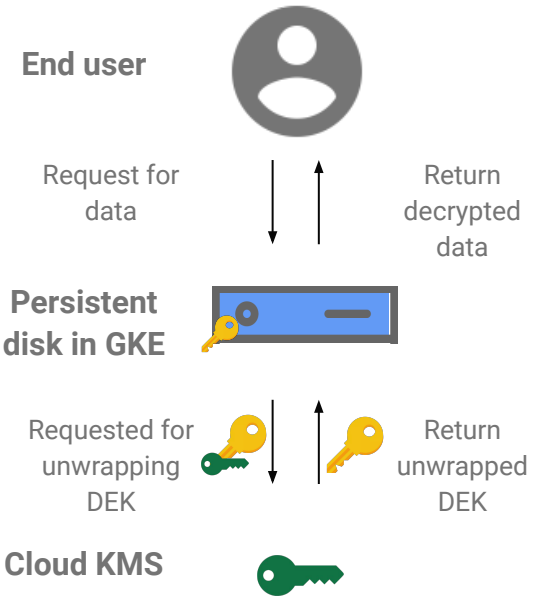
GKE: Application layer secrets encryption BETA



- Data in etcd, such as secrets, are encrypted locally with a data encryption key (DEK)
- The data encryption key is also stored in etcd, but encrypted with a key encryption key (KEK) in Cloud KMS, via a KMS plugin
- Each encryption generates a new DEK, rather than rotating the DEK

Google Cloud

GKE: Customer-managed encryption keys BETA



- Encryption key per cluster, for the GCE disks in that cluster
- You specify the Cloud KMS key you want to use
  - Must be in the same region
  - Must give permissions on the key to the GKE service account
- Key is specified as part of storage class
- Limitations
  - Only for dynamically attached persistent disks
  - Requires self-installation of CSI driver

Google Cloud

Managed base images

- Managed base images are patched for security vulnerabilities automatically by Google with the most recently available patches from upstream
- Most common distributions: Available for Debian, Ubuntu, and CentOS, on GCP Marketplace
- Scanned and patched: Scanned for vulnerabilities using Container Registry Vulnerability Scanning. Where a patch for these exists upstream, it is applied.
- Used by Google: Google maintains several base image for building its own applications, including Google Cloud services like Google App Engine

Google Cloud

https://cloud.google.com/container-registry/docs/managed-base-images

GCR: Vulnerability Scanning

- Scans all images in your private Google Container Registry for known Common Vulnerabilities and Exposures (CVEs)
- Examines images and packages installed in images
- Works for: Debian, Ubuntu and Alpine images
- Images are scanned when:
  - An image is added to the registry
  - There is an update to the vulnerability database

Google Cloud

https://cloud.google.com/container-registry/docs/vulnerability-scanning

Binary Authorization

Run only what you trust. Enforce that only images with required signatures, or from whitelisted repositories, can be deployed to production.

- Customer defines signatures requirements
- Images are signed throughout CI/CD process by build/test tools
- Image deployments are blocked/allowed at GKE control plane based on policy

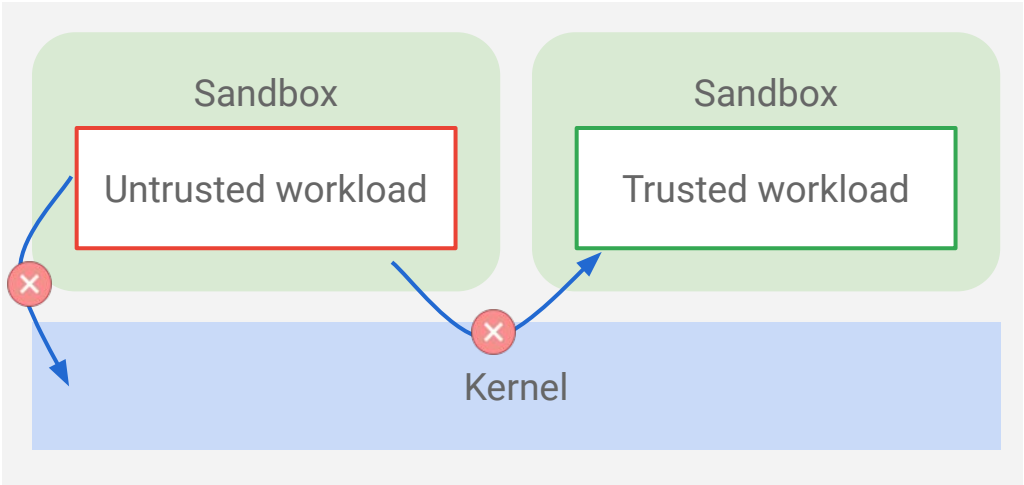
https://cloud.google.com/binary-authorization/

GKE: GKE Sandbox BETA

Run **trusted and untrusted** workloads on the same node

Rather than achieving isolation via separate VMs, you can run workloads of different trust levels on the same node

Performance improvements from not having to allocate a new cluster to achieve isolation



Google Cloud

https://cloud.google.com/kubernetes-engine/sandbox/

Who protects what?

Hardening the control plane is Google's responsibility

Google does most of the hard work to protect nodes - but it's the user's responsibility to upgrade to reap the benefits

Protecting workloads is still the user's responsibility

Responding to attacks on Google's infrastructure is Google's responsibility, whereas responding to attacks on user workloads is the user's responsibility

Upcoming products and features

Under NDA

Google Cloud

NDA

Container security: What's next?

INFRASTRUCTURE SECURITY

Tests in Security Health Analytics ALPHA
Automated checks for configurations following the GKE hardening guide

GKE CIS Benchmarks
Updated CIS benchmarks for Kubernetes, as well as new CIS benchmarks for GKE specifically

SOFTWARE SUPPLY CHAIN

Vulnerability deployment policy ALPHA
Enforce deployment policies based on the vulnerabilities and base images used

Deployment History EAP
Verify if your environment is affected by a newly discovered vulnerability

RUNTIME SECURITY

Container Threat Detection ALPHA
Monitor, detect, and react to common container attacks

Google Cloud

Container security

Infrastructure security

Tests in Security Health Analytics

Automated checks for configurations following the GKE hardening guide

Google Cloud

CSCC: Security Health Analytics ALPHA

Container Scanner

Category	Finding description
IP_ALIAS_DISABLED	Indicates that a GKE Cluster was created with Alias IP ranges enabled.
LEGACY_AUTHORIZATION_ENABLED	Indicates that Legacy Authorization is enabled on GKE Clusters.
MASTER_AUTHORIZED_NETWORKS_DISABLED	Indicates that Master authorized networks is not enabled on GKE Clusters.
MONITORING_DISABLED	Indicates that Stackdriver Monitoring is disabled on GKE Clusters.
NETWORK_POLICY_DISABLED	Indicates that Network policy is disabled on GKE Clusters.
POD_SECURITY_POLICY_DISABLED	Indicates that `PodSecurityPolicy` is disabled on a GKE Cluster.
PRIVATE_CLUSTER_DISABLED	Indicates that a GKE Cluster has a Private cluster disabled.
WEB_UI_ENABLED	Indicates that the GKE web UI (dashboard) is enabled.

- Checks for common misconfigurations in GCP
  - Such as those in the GKE hardening guide
- Results shown in the Cloud Security Command Center

Google Cloud

https://cloud.google.com/security-command-center/docs/how-to-enable-security-health-analytics

Container security

Software supply chain

Vulnerability deployment policy

Enforce deployment policies based on the vulnerabilities and base images used

Deployment History

Verify if your environment is affected by a newly discovered vulnerability

Signing keys in Cloud KMS

Manage signing keys in Binary Authorization in Cloud KMS

GCR: Deployment History[EAP]

- Works for all images stored in GCR and images deployed to GCP from other registries
- Automatically detects when container images are running in GKE or App Engine Flex
- Makes information available via the Container Analysis API about what's running and has been running in a project
- Enables identification of known vulnerabilities that apply to running containers

Questions

- How do you know what is currently deployed in your environment? And associated vulns?

Container security

Runtime security

Container Threat Detection

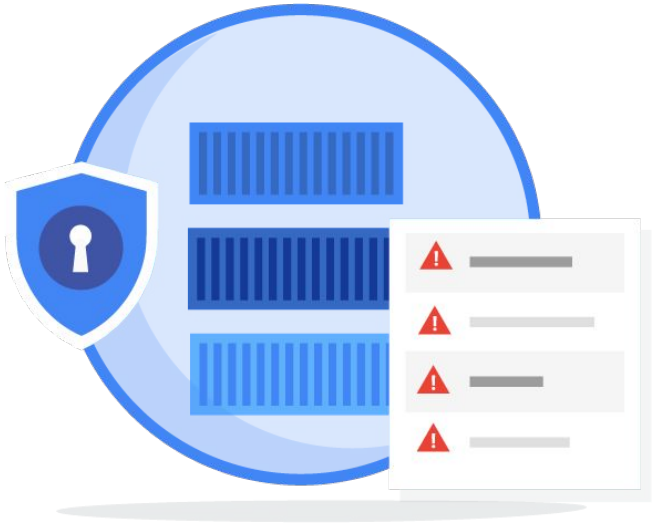
Monitor, detect, and react to common container attacks

Google Cloud

Container Threat Detection[Alpha]

Container Threat Detection will offer **detection capabilities that cover the most common threats**

- Pre-built detectors for the top attacks
  - Suspicious binary
  - Suspicious library
  - Reverse shell
- Integrations with
  - Cloud Security Command Center for surfacing security findings
  - GKE for one-click deployment
  - Stackdriver for events
- Kernel integrations only available in COS



Google Cloud

Learn more

cloud.google.com/containers/security

g.co/gke/security
g.co/gke/hardening
g.co/gke/securitybulletins