CIS Benchmarks

This document explains what the CIS Kubernetes and GKE Benchmarks are, how to audit your compliance with the Benchmarks, and what GKE configures where you cannot directly audit or implement a recommendation yourself.

Using the CIS Benchmarks

The Center for Internet Security (CIS) releases benchmarks for best practice security recommendations. The CIS Kubernetes Benchmark (https://www.cisecurity.org/benchmark/kubernetes/) is a set of recommendations for configuring Kubernetes to support a strong security posture. The Benchmark is tied to a specific Kubernetes release. The CIS Kubernetes Benchmark is written for the open source Kubernetes distribution and intended to be as universally applicable across distributions as possible.

With a managed service like GKE, not all items on the Benchmark are your responsibility, and there are recommendations that you cannot audit or remediate directly yourself. **If you are running on GKE, use the CIS GKE Benchmark** (#accessing-gke-benchmark)**, which is a child benchmark of the CIS Kubernetes Benchmark, meant specifically to be applied to the GKE distribution.** This draws from the existing CIS Benchmark, but removes items that are not configurable or managed by the user and adds additional controls that are Google Cloud-specific.

Multiple set of Benchmarks

With GKE, you can use CIS Benchmarks for: GKE, Kubernetes, Docker, and Linux. Note that Container-Optimized OS (COS) (/container-optimized-os/docs), the default node OS for GKE, does not have a CIS Benchmark; and that the container runtime containerd (/kubernetes-engine/docs/concepts/using-containerd) also does not have a CIS Benchmark.

Some tools attempt to analyze Kubernetes nodes against multiple CIS Benchmarks (e.g. Linux, Docker, and Kubernetes) and combine the results. This often results in confusing and potentially contradictory advice because those benchmarks weren't designed to be combined and applied in a Kubernetes environment.

Shared responsibility model

In GKE, under the Shared responsibility model (https://cloud.google.com/blog/products/containers-kubernetes/exploring-container-security-the-shared-responsibility-model-in-gke-container-security-shared-responsibility-model-gke), Google manages the following Kubernetes components:

- The control plane (including the master VMs, API server, other components on the master VMs, and etcd)
- The Kubernetes distribution
- The nodes' operating system

Configurations related to these items are generally not available for you to audit or modify in GKE.

You are still responsible for upgrading the nodes that run your workloads, and the workloads themselves. You can generally audit and remediate any recommendations to these components.

Ability to audit and remediate

The CIS GKE Benchmark draws from the existing CIS Kubernetes Benchmark, but remove items that are not configurable or managed by the user, and add additional controls that are Google Cloud-specific.

The sections of the CIS GKE Benchmark are:

- **Control Plane Components, etcd, and Control Plane Configuration (sections 1, 2 and 3)** are from the CIS Kubernetes Benchmark. These generally cannot be audited or remediated on GKE.
- **Worker Nodes (section 4)** is from the CIS Kubernetes Benchmark. Some of these items can be audited or remediated on GKE, but instructions may be different.
- **Policies (section 5)** is also from the CIS Kubernetes Benchmark. These are generally directly applicable to GKE, without any change in instructions.
- **Managed services (section 6)** in the CIS GKE Benchmark is net new content for GKE specifically. This section contains all of the Recommendations that are specific to Google Cloud controls. These can be audited and remediated on GKE.

For the items that cannot be audited or remediated on GKE, see the section on Default values (#default-values) to understand how a default cluster created in GKE performs against the CIS Kubernetes Benchmark.

Versions

Note that the version numbers for different Benchmarks may not be the same.

This document refers to these versions:

Kubernetes version	CIS Kubernetes Benchmark version	CIS GKE Benchmark version
1.15	1.5.0	1.0.0

CIS Kubernetes Benchmark

Accessing the Benchmark

The CIS Kubernetes Benchmark is available on the CIS website (https://www.cisecurity.org/benchmark/kubernetes/).

Recommendation Levels

In the CIS Kubernetes Benchmark,

Level	Description
Level 1	Recommendations intend to: • be practical and prudent; • provide a clear security benefit; and • not inhibit the utility of the technology beyond acceptable means.
Level 2	Extends the Level 1 profile. Recommendations exhibit one or more of the following characteristics: • are intended for environments or use cases where security is paramount; • acts as defense in depth measure; or • may negatively inhibit the utility or performance of the technology.

Recommendation Scoring

In the CIS Kubernetes Benchmark,

Scoring	Description
Scored	Failure to comply with these recommendations will decrease the final benchmark score.
Not Scored	Failure to comply with these recommendations will not decrease the final benchmark score.

Evaluation on GKE

We use the following values to specify the status of Kubernetes Recommendations in GKE:

Status	Description
Pass	Complies with a Benchmark Recommendation.
Fail	Does not comply with a Benchmark Recommendation.
Equivalent Control	Does not comply with the exact terms in the Benchmark Recommendation, but other mechanisms in GKE exist to provide equivalent security controls.
Depends on Environment	GKE does not configure items related to this Recommendation. The user's configuration determines whether their environment complies with a Benchmark Recommendation.

Jnless specified, the values for workloads pertain to the environment you are running on GKE, not to GKE system containers.

Status on GKE

When creating a new GKE cluster with the specified version (#version), here's how it will perform against the CIS Kubernetes Benchmark.

The Scoring for the CIS Kubernetes Benchmark and the CIS GKE Benchmark are different, as some controls cannot be audited or remediated in GKE. The follow evaluates a new GKE cluster against the CIS Kubernetes Benchmark, referring to the controls in sections 1-5. When evaluating your own environment, you shoul S GKE Benchmark to perform an audit.

Status of default GKE cluster:

#	Recommendation	Scored/ Not Scored	Level	Default Status
1	*Control Plane Components*			
1.1	**Master Node Configuration Files**			
1.1.1	Ensure that the API server pod specification file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.2	Ensure that the API server pod specification file ownership is set to `root:root`	Scored	L1	Pass
1.1.3	Ensure that the controller manager pod specification file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.4	Ensure that the controller manager pod specification file ownership is set to `root:root`	Scored	L1	Pass
1.1.5	Ensure that the scheduler pod specification file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.6	Ensure that the scheduler pod specification file ownership is set to `root:root`	Scored	L1	Pass
1.1.7	Ensure that the etcd pod specification file permissions are set to `644` or more restrictive	Scored	L1	Pass

#	Recommendation	Scored/ Not Scored	Level	Default Status
1.1.8	Ensure that the etcd pod specification file ownership is set to `root:root`	Scored	L1	Pass
1.1.9	Ensure that the Container Network Interface file permissions are set to `644` or more restrictive	Not Scored	L1	Pass
1.1.10	Ensure that the Container Network Interface file ownership is set to `root:root`	Not Scored	L1	Pass
1.1.11	Ensure that the etcd data directory permissions are set to `700` or more restrictive	Scored	L1	Pass
1.1.12	Ensure that the etcd data directory ownership is set to `etcd:etcd`	Scored	L1	Pass
1.1.13	Ensure that the admin.conf file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.14	Ensure that the admin.conf file ownership is set to `root:root`	Scored	L1	Pass
1.1.15	Ensure that the scheduler.conf file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.16	Ensure that the scheduler.conf file ownership is set to `root:root`	Scored	L1	Pass
1.1.17	Ensure that the controller-manager.conf file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.18	Ensure that the controller-manager.conf file ownership is set to `root:root`	Scored	L1	Pass
1.1.19	Ensure that the Kubernetes PKI directory and file ownership is set to `root:root`	Scored	L1	Pass
1.1.20	Ensure that the Kubernetes PKI certificate file permissions are set to `644` or more restrictive	Scored	L1	Pass
1.1.21	Ensure that the Kubernetes PKI key file permissions are set to `600`	Scored	L1	Pass
1.2	**API Server**			

#	Recommendation	Scored/ Not Scored	Level	Default Status
1.2.1	Ensure that the --anonymous-auth argument is set to false	Not Scored	L1	Fail
1.2.2	Ensure that the --basic-auth-file argument is not set	Scored	L1	Pass
1.2.3	Ensure that the --token-auth-file parameter is not set	Scored	L1	Fail
1.2.4	Ensure that the --kubelet-https argument is set to true	Scored	L1	Pass
1.2.5	Ensure that the --kubelet-client-certificate and --kubelet-client-key arguments are set as appropriate	Scored	L1	Pass
1.2.6	Ensure that the --kubelet-certificate-authority argument is set as appropriate	Scored	L1	Pass
1.2.7	Ensure that the --authorization-mode argument is not set to AlwaysAllow	Scored	L1	Pass
1.2.8	Ensure that the --authorization-mode argument includes Node	Scored	L1	Pass
1.2.9	Ensure that the --authorization-mode argument includes RBAC	Scored	L1	Pass
1.2.10	Ensure that the admission control plugin EventRateLimit is set	Not Scored	L1	Fail
1.2.11	Ensure that the admission control plugin AlwaysAdmit is not set	Scored	L1	Pass
1.2.12	Ensure that the admission control plugin AlwaysPullImages is set	Not Scored	L1	Fail
1.2.13	Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used	Not Scored	L1	Fail
1.2.14	Ensure that the admission control plugin ServiceAccount is set	Scored	L1	Pass
1.2.15	Ensure that the admission control plugin NamespaceLifecycle is set	Scored	L1	Pass

#	Recommendation	Scored/ Not Scored	Level	Default Status
1.2.16	Ensure that the admission control plugin PodSecurityPolicy is set	Scored	L1	Fail
1.2.17	Ensure that the admission control plugin NodeRestriction is set	Scored	L1	Pass
1.2.18	Ensure that the --insecure-bind-address argument is not set	Scored	L1	Pass
1.2.19	Ensure that the --insecure-port argument is set to 0	Scored	L1	Pass
1.2.20	Ensure that the --secure-port argument is not set to 0	Scored	L1	Pass
1.2.21	Ensure that the --profiling argument is set to false	Scored	L1	Fail
1.2.22	Ensure that the --audit-log-path argument is set	Scored	L1	Equivalent Control
1.2.23	Ensure that the --audit-log-maxage argument is set to 30 or as appropriate	Scored	L1	Equivalent Control
1.2.24	Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate	Scored	L1	Equivalent Control
1.2.25	Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate	Scored	L1	Equivalent Control
1.2.26	Ensure that the --request-timeout argument is set as appropriate	Scored	L1	Pass
1.2.27	Ensure that the --service-account-lookup argument is set to true	Scored	L1	Pass
1.2.28	Ensure that the --service-account-key-file argument is set as appropriate	Scored	L1	Pass
1.2.29	Ensure that the --etcd-certfile and --etcd-keyfile arguments are set as appropriate	Scored	L1	Fail
1.2.30	Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate	Scored	L1	Pass
1.2.31	Ensure that the --client-ca-file argument is set as appropriate	Scored	L1	Pass

#	Recommendation	Scored/ Not Scored	Level	Default Status
1.2.32	Ensure that the --etcd-cafile argument is set as appropriate	Scored	L1	Fail
1.2.33	Ensure that the --encryption-provider-config argument is set as appropriate	Scored	L1	Pass
1.2.34	Ensure that encryption providers are appropriately configured	Scored	L1	Fail
1.2.35	Ensure that the API Server only makes use of Strong Cryptographic Ciphers	Not Scored	L1	Pass
1.3	**Controller Manager**			
1.3.1	Ensure that the --terminated-pod-gc-threshold argument is set as appropriate	Scored	L1	Pass
1.3.2	Ensure that the --profiling argument is set to false	Scored	L1	Fail
1.3.3	Ensure that the --use-service-account-credentials argument is set to true	Scored	L1	Fail
1.3.4	Ensure that the --service-account-private-key-file argument is set as appropriate	Scored	L1	Pass
1.3.5	Ensure that the --root-ca-file argument is set as appropriate	Scored	L1	Pass
1.3.6	Ensure that the RotateKubeletServerCertificate argument is set to true	Scored	L2	Equivalent Control
1.3.7	Ensure that the --bind-address argument is set to 127.0.0.1	Scored	L1	Pass
1.4	**Scheduler**			
1.4.1	Ensure that the --profiling argument is set to false	Scored	L1	Fail
1.4.2	Ensure that the --bind-address argument is set to 127.0.0.1	Scored	L1	Pass
2	***etcd***			

#	Recommendation	Scored/ Not Scored	Level	Default Status
2.1	Ensure that the --cert-file and --key-file arguments are set as appropriate	Scored	L1	Fail
2.2	Ensure that the --client-cert-auth argument is set to true	Scored	L1	Fail
2.3	Ensure that the --auto-tls argument is not set to true	Scored	L1	Pass
2.4	Ensure that the --peer-cert-file and --peer-key-file arguments are set as appropriate	Scored	L1	Equivalent Control
2.5	Ensure that the --peer-client-cert-auth argument is set to true	Scored	L1	Equivalent Control
2.6	Ensure that the --peer-auto-tls argument is not set to true	Scored	L1	Equivalent Control
2.7	Ensure that a unique Certificate Authority is used for etcd	Not Scored	L2	Pass
3	*Control Plane Configuration*			
3.1	**Authentication and Authorization**			
3.1.1	Client certificate authentication should not be used for users	Not Scored	L2	Pass
3.2	**Logging**			
3.2.1	Ensure that a minimal audit policy is created	Scored	L1	Pass
3.2.2	Ensure that the audit policy covers key security concerns	Not Scored	L2	Pass
4	*Worker Nodes*			
4.1	**Worker Node Configuration Files**			
4.1.1	Ensure that the kubelet service file permissions are set to 644 or more restrictive	Scored	L1	Pass

#	Recommendation	Scored/ Not Scored	Level	Default Status
4.1.2	Ensure that the kubelet service file ownership is set to `root:root`	Scored	L1	Pass
4.1.3	Ensure that the proxy kubeconfig file permissions are set to `644` or more restrictive	Scored	L1	Pass
4.1.4	Ensure that the proxy kubeconfig file ownership is set to `root:root`	Scored	L1	Pass
4.1.5	Ensure that the kubelet.conf file permissions are set to `644` or more restrictive	Scored	L1	Pass
4.1.6	Ensure that the kubelet.conf file ownership is set to `root:root`	Scored	L1	Pass
4.1.7	Ensure that the certificate authorities file permissions are set to `644` or more restrictive	Scored	L1	Pass
4.1.8	Ensure that the client certificate authorities file ownership is set to `root:root`	Scored	L1	Pass
4.1.9	Ensure that the kubelet configuration file has permissions set to `644` or more restrictive	Scored	L1	Pass
4.1.10	Ensure that the kubelet configuration file ownership is set to `root:root`	Scored	L1	Pass
4.2	**Kubelet**			
4.2.1	Ensure that the --anonymous-auth argument is set to false	Scored	L1	Pass
4.2.2	Ensure that the --authorization-mode argument is not set to AlwaysAllow	Scored	L1	Pass
4.2.3	Ensure that the --client-ca-file argument is set as appropriate	Scored	L1	Pass
4.2.4	Ensure that the --read-only-port argument is set to 0	Scored	L1	Fail
4.2.5	Ensure that the --streaming-connection-idle-timeout argument is not set to 0	Scored	L1	Pass
4.2.6	Ensure that the --protect-kernel-defaults argument is set to true	Scored	L1	Fail

#	Recommendation	Scored/ Not Scored	Level	Default Status
4.2.7	Ensure that the --make-iptables-util-chains argument is set to true	Scored	L1	Pass
4.2.8	Ensure that the --hostname-override argument is not set	Not Scored	L1	Pass
4.2.9	Ensure that the --event-qps argument is set to 0 or a level which ensures appropriate event capture	Not Scored	L2	Fail
4.2.10	Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate	Scored	L1	Equivalent Control
4.2.11	Ensure that the --rotate-certificates argument is not set to false	Scored	L1	Equivalent Control
4.2.12	Ensure that the RotateKubeletServerCertificate argument is set to true	Scored	L1	Equivalent Control
4.2.13	Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers	Not Scored	L1	Pass
5	*Policies*			
5.1	**RBAC and Service Accounts**			
5.1.1	Ensure that the cluster-admin role is only used where required	Not Scored	L1	Depends on Environment
5.1.2	Minimize access to secrets	Not Scored	L1	Depends on Environment
5.1.3	Minimize wildcard use in Roles and ClusterRoles	Not Scored	L1	Depends on Environment
5.1.4	Minimize access to create pods	Not Scored	L1	Depends on Environment

#	Recommendation	Scored/ Not Scored	Level	Default Status
5.1.5	Ensure that default service accounts are not actively used	Scored	L1	Depends on Environment
5.1.6	Ensure that Service Account Tokens are only mounted where necessary	Not Scored	L1	Depends on Environment
5.2	**Pod Security Policies**			
5.2.1	Minimize the admission of privileged containers	Not Scored	L1	Depends on Environment
5.2.2	Minimize the admission of containers wishing to share the host process ID namespace	Scored	L1	Depends on Environment
5.2.3	Minimize the admission of containers wishing to share the host IPC namespace	Scored	L1	Depends on Environment
5.2.4	Minimize the admission of containers wishing to share the host network namespace	Scored	L1	Depends on Environment
5.2.5	Minimize the admission of containers with allowPrivilegeEscalation	Scored	L1	Depends on Environment
5.2.6	Minimize the admission of root containers	Not Scored	L2	Depends on Environment
5.2.7	Minimize the admission of containers with the NET_RAW capability	Not Scored	L1	Depends on Environment

#	Recommendation	Scored/ Not Scored	Level	Default Status
5.2.8	Minimize the admission of containers with added capabilities	Not Scored	L1	Depends on Environment
5.2.9	Minimize the admission of containers with capabilities assigned	Not Scored	L2	Depends on Environment
5.3	**Network Policies and CNI**			
5.3.1	Ensure that the CNI in use supports Network Policies	Not Scored	L1	Pass
5.3.2	Ensure that all Namespaces have Network Policies defined	Scored	L2	Depends on Environment
5.4	**Secrets Management**			
5.4.1	Prefer using secrets as files over secrets as environment variables	Not Scored	L1	Depends on Environment
5.4.2	Consider external secret storage	Not Scored	L2	Depends on Environment
5.5	**Extensible Admission Control**			
5.5.1	Configure Image Provenance using ImagePolicyWebhook admission controller	Not Scored	L2	Depends on Environment
5.6	**General Policies**			
5.6.1	Create administrative boundaries between resources using namespaces	Not Scored	L1	Depends on Environment

#	Recommendation	Scored/ Not Scored	Level	Default Status
5.6.2	Ensure that the seccomp profile is set to docker/default in your pod definitions	Not Scored	L2	Depends on Environment
5.6.3	Apply Security Context to Your Pods and Containers	Not Scored	L2	Depends on Environment
5.6.4	The default namespace should not be used	Scored	L2	Depends on Environment

Default values on GKE

Where the default for a new GKE cluster does not pass a Recommendation from the CIS Kubernetes Benchmark, here are the default values used in GKE, with an explanation. Some of these Recommendations can be remediated, following the remediation procedures laid out in the CIS GKE Benchmark. Items that can be automatically audited are marked as Scored in the CIS GKE Benchmark.

Default values for Recommendations which Fail or Depends on Environment in a default GKE cluster:

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
1	*Control Plane Components*						
1.2	API Server						

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
1.2.1	Ensure that the --anonymous-auth argument is set to false	Not Scored	L1	Fail	`true`	Some GKE monitoring components use anonymous authentication to obtain metrics. Although GKE allows anonymous authentication for the kubelet, the exposure is identical to the read-only port as GKE disables the additional debugging handlers.	Not Scored
1.2.3	Ensure that the --token-auth-file parameter is not set	Scored	L1	Fail	Set	Some master components are bootstrapped using static tokens, which are then used to authenticate to the API server.	Not Scored
1.2.10	Ensure that the admission control plugin EventRateLimit is set	Not Scored	L1	Fail	Not set	GKE does not support the Event Rate Limit admission controller as it is a Kubernetes Alpha feature.	Not Scored
1.2.12	Ensure that the admission control plugin AlwaysPullImages is set	Not Scored	L1	Fail	Not set	GKE does not enable the Always Pull Images admission controller, as this overrides the ImagePullStrategy field. Always Pulling images may also limit the ability to scale, as compared to preloaded images.	Not Scored
1.2.13	Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used	Not Scored	L1	Fail	Not set	GKE does not enable the Security Context admission controller by default. Using a Pod Security Policy allows more control and is preferred.	Not Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
1.2.16	Ensure that the admission control plugin PodSecurityPolicy is set	Scored	L1	Fail	Not set	GKE does not enable the Pod Security Policy admission controller by default, as this requires a policy to be set. GKE customers can enable PodSecurityPolicy.	Not Scored, see also 6.10.3
1.2.21	Ensure that the --profiling argument is set to false	Scored	L1	Fail	Not set	GKE uses profiling for debugging.	Not Scored
1.2.22	Ensure that the --audit-log-path argument is set	Scored	L1	Equivalent Control	Not set	GKE captures audit logs, but does not use these flags for auditing. See GKE Audit policy (https://cloud.google.com/kubernetes-engine/docs/concepts/audit-policy) for more details.	Not Scored
1.2.23	Ensure that the --audit-log-maxage argument is set to 30 or as appropriate	Scored	L1	Equivalent Control	Not set	GKE captures audit logs, but does not use these flags for auditing. See GKE Audit policy (https://cloud.google.com/kubernetes-engine/docs/concepts/audit-policy) for more details.	Not Scored
1.2.24	Ensure that the --audit-log-maxbackup argument is set to 10 or as appropriate	Scored	L1	Equivalent Control	Not set	GKE captures audit logs, but does not use these flags for auditing. See GKE Audit policy (https://cloud.google.com/kubernetes-engine/docs/concepts/audit-policy) for more details.	Not Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
1.2.25	Ensure that the --audit-log-maxsize argument is set to 100 or as appropriate	Scored	L1	Equivalent Control	Not set	GKE captures audit logs, but does not use these flags for auditing. See GKE Audit policy (https://cloud.google.com/kubernetes-engine/docs/concepts/audit-policy) for more details.	Not Scored
1.2.29	Ensure that the --etcd-certfile and --etcd-keyfile arguments are set as appropriate	Scored	L1	Fail	Not set	GKE does not currently use mTLS to protect connections between the API server to etcd. Note that etcd listens on localhost. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored
1.2.32	Ensure that the --etcd-cafile argument is set as appropriate	Scored	L1	Fail	Not set	GKE does not currently use mTLS to protect connections between the API server to etcd. Note that etcd listens on localhost. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
1.2.34	Ensure that encryption providers are appropriately configured	Scored	L1	Fail	`identity`	GKE encrypts customer content at rest by default (https://cloud.google.com/security/encryption-at-rest/default-encryption/). To further encrypts secrets, use Application-layer Secrets Encryption (https://cloud.google.com/kubernetes-engine/docs/how-to/encrypting-secrets).	Not Scored, see also 6.3.1
1.3	**Controller Manager**						
1.3.2	Ensure that the --profiling argument is set to false	Scored	L1	Fail	`true`	GKE uses profiling for debugging.	Not Scored
1.3.6	Ensure that the RotateKubeletServerCertificate argument is set to true	Scored	L2	Equivalent Control	`false`	GKE rotates kubelet certificates, but does not use this flag.	Not Scored
1.4	**Scheduler**						
1.4.1	Ensure that the --profiling argument is set to false	Scored	L1	Fail	`true`	GKE uses profiling for debugging.	Not Scored
2	***etcd***						

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
2.1	Ensure that the --cert-file and --key-file arguments are set as appropriate	Scored	L1	Fail	Not set	GKE does not currently use mTLS to protect connections between the API server to etcd. Note that etcd listens on localhost. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored
2.2	Ensure that the --client-cert-auth argument is set to true	Scored	L1	Fail	Not set	GKE does not currently use mTLS to protect connections between the API server to etcd. Note that etcd listens on localhost. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored
2.4	Ensure that the --peer-cert-file and --peer-key-file arguments are set as appropriate	Scored	L1	Equivalent Control	Not set	GKE uses mTLS for peer traffic between instances of etcd. These flags are used for regional clusters but not zonal clusters, as there is only one instance of etcd in a zonal cluster. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
2.5	Ensure that the --peer-client-cert-auth argument is set to true	Scored	L1	Equivalent Control	Not set	GKE uses mTLS for peer traffic between instances of etcd. These flags are used for regional clusters but not zonal clusters, as there is only one instance of etcd in a zonal cluster. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored
2.6	Ensure that the --peer-auto-tls argument is not set to true	Scored	L1	Equivalent Control	Not set	GKE uses mTLS for peer traffic between instances of etcd. These flags are used for regional clusters but not zonal clusters, as there is only one instance of etcd in a zonal cluster. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Not Scored
4	*Worker Nodes*						
4.2	**Kubelet**						
4.2.4	Ensure that the --read-only-port argument is set to 0	Scored	L1	Fail	`10255`	Some GKE monitoring components use the kubelet read-only port to obtain metrics.	Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
4.2.6	Ensure that the --protect-kernel-defaults argument is set to true	Scored	L1	Fail	`false`	GKE doesn't protect kernel defaults from Kubernetes, as customer workloads may want to modify these.	Scored
4.2.9	Ensure that the --event-qps argument is set to 0 or a level which ensures appropriate event capture	Not Scored	L2	Fail	5	Events are Kubernetes objects stored in etcd. To avoid overwhelming etcd they are only kept for one hour, and are not an appropriate security auditing mechanism. Allowing unlimited events as suggested in this control exposes the cluster to unnecessary DoS risk and contradicts the recommendation to use admission EventRateLimits. Security relevant events that need permanent storage should be sent to logs.	Scored
4.2.10	Ensure that the --tls-cert-file and --tls-private-key-file arguments are set as appropriate	Scored	L1	Equivalent Control	Not Set	GKE uses mTLS for kubelet to API server traffic. GKE uses TLS for API server to kubelet traffic, which is authenticated for GKE v1.12+ clusters. GKE does not use these flags but rather this is specified in the kubelet config file. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
4.2.11	Ensure that the --rotate-certificates argument is not set to false	Scored	L1	Equivalent Control	Not set	GKE rotates server certificates for GKE v1.12+ clusters. GKE does not use these flags but rather this is specified in the kubelet config file. GKE does not rotate client certificates, unless Shielded GKE Nodes are enabled. In this case, GKE does not use these flags but runs a separate process for certificate rotation. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Scored
4.2.12	Ensure that the RotateKubeletServerCertificate argument is set to true	Scored	L1	Equivalent Control	Not set	GKE rotates server certificates for GKE v1.12+ clusters. GKE does not use these flags but rather this is specified in the kubelet config file. See Cluster trust (https://cloud.google.com/kubernetes-engine/docs/concepts/cluster-trust) for more details.	Scored
5	*Policies*						
5.1	**RBAC and Service Accounts**						
5.1.1	Ensure that the cluster-admin role is only used where required	Not Scored	L1	Depends on Environment	n/a		Not Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
5.1.2	Minimize access to secrets	Not Scored	L1	Depends on Environment	n/a		Not Scored
5.1.3	Minimize wildcard use in Roles and ClusterRoles	Not Scored	L1	Depends on Environment	n/a		Not Scored
5.1.4	Minimize access to create pods	Not Scored	L1	Depends on Environment	n/a		Not Scored
5.1.5	Ensure that default service accounts are not actively used	Scored	L1	Depends on Environment	n/a		Scored
5.1.6	Ensure that Service Account Tokens are only mounted where necessary	Not Scored	L1	Depends on Environment	n/a		Not Scored
5.2	**Pod Security Policies**						
5.2.1	Minimize the admission of privileged containers	Not Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.2	Minimize the admission of containers wishing to share the host process ID namespace	Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
5.2.3	Minimize the admission of containers wishing to share the host IPC namespace	Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.4	Minimize the admission of containers wishing to share the host network namespace	Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.5	Minimize the admission of containers with allowPrivilegeEscalation	Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.6	Minimize the admission of root containers	Not Scored	L2	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.7	Minimize the admission of containers with the NET_RAW capability	Not Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.8	Minimize the admission of containers with added capabilities	Not Scored	L1	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.2.9	Minimize the admission of containers with capabilities assigned	Not Scored	L2	Depends on Environment	n/a	No Pod Security Policy is set by default.	Scored
5.3	**Network Policies and CNI**						

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
5.3.2	Ensure that all Namespaces have Network Policies defined	Scored	L2	Depends on Environment	n/a		Scored
5.4	**Secrets Management**						
5.4.1	Prefer using secrets as files over secrets as environment variables	Not Scored	L1	Depends on Environment	n/a		Not Scored
5.4.2	Consider external secret storage	Not Scored	L2	Depends on Environment	n/a		Not Scored
5.5	**Extensible Admission Control**						
5.5.1	Configure Image Provenance using ImagePolicyWebhook admission controller	Not Scored	L2	Depends on Environment	Not enabled	GKE does not enable the Image Policy Webhook admission controller by default. Image Provenance using Binary Authorization is not set by default, as this requires a policy to be set.	Not Scored, see also 6.10.5
5.6	**General Policies**						
5.6.1	Create administrative boundaries between resources using namespaces	Not Scored	L1	Depends on Environment	n/a		Not Scored

#	Recommendation	Scored/ Not Scored on CIS Kubernetes Benchmark	Level	Default Status	Default Value	Justification	Scored/ Not Scored on CIS GKE Benchmark
5.6.2	Ensure that the seccomp profile is set to docker/default in your pod definitions	Not Scored	L2	Depends on Environment	n/a	No seccomp profile is set by default.	Not Scored
5.6.3	Apply Security Context to Your Pods and Containers	Not Scored	L2	Depends on Environment	n/a	No Security Context is set by default.	Not Scored
5.6.4	The default namespace should not be used	Scored	L2	Depends on Environment	n/a		Scored

CIS GKE Benchmark

Although the only additional Recommendations in the CIS GKE Benchmark are in section 6, some of the audit and remediation procedures for Recommendations 1-5 are different in the CIS GKE Benchmark from the CIS Kubernetes Benchmark.

Accessing the Benchmark

The CIS GKE Benchmark is available on the CIS website:

- Go to the full list of CIS Benchmarks (https://www.cisecurity.org/cis-benchmarks/).

- Under the Kubernetes heading, click **Expand to see related content**.
- The CIS GKE Benchmark is listed for download.

Recommendation Levels

In the CIS GKE Benchmark,

Level	Description
Level 1	Recommendations are meant to be widely applicable. These should be applied to almost all environments.
Level 2	Recommendations result in a more stringent security environment, but are not necessarily applicable to all cases. These may have performance impact, or may not be able to be applied in concert with other Recommendations. These should be evaluated for your environment before being applied. In some cases, for example multi-tenant workloads, these Recommendations may be more relevant.

Recommendation Scoring

In the CIS GKE Benchmark,

Scoring	Description
Scored	Recommendations are easily tested using an automated method, and has a value that can be definitively evaluated. These Recommendations only include Generally Available (https://cloud.google.com/products/#product-launch-stages) products or features.

Scoring	Description
Not Scored	Recommendations cannot be easily assessed using automation or requires evaluation to determine the exact implementation appropriate for your workload. These Recommendations may use Beta (https://cloud.google.com/products/#product-launch-stages) products or features. For example, Pod Security Policy (https://cloud.google.com/kubernetes-engine/docs/how-to/pod-security-policies) requires the use of a policy specific to your workload, and is a Beta feature, so is Not Scored.

Since many configurations in the control plane cannot be audited or remediated in GKE, this means that some controls, though Scored in the CIS Kubernetes mark, are Not Scored in the CIS GKE Benchmark.

Evaluation on GKE

For GKE-specific Recommendations (section 6), since these are all configurable such that they can be configured to Pass in your environment, we use the following values to specify the default values:

Status	Description
Default	A new cluster complies with a Benchmark Recommendation by default.
Not Default	A new cluster does not comply with a Benchmark Recommendation by default.
Depends on Environment	GKE does not configure items related to this Recommendation. The user's configuration determines whether their environment complies with a Benchmark Recommendation.

Default values on GKE

When creating a new GKE cluster with the specified version (#version), here's how it will perform against the CIS Kubernetes Benchmark.

Status of default GKE cluster:

#	Recommendation	Scored/ Not Scored	Level	Default Status
6	*Managed services*			
6.1	**Image Registry and Image Scanning**			
6.1.1	Ensure Image Vulnerability Scanning using GCR Container Analysis or a third party provider	Scored	L1	Not Default
6.1.2	Minimize user access to GCR	Scored	L1	Depends on Environment
6.1.3	Minimize cluster access to read-only for GCR	Scored	L1	Not Default
6.1.4	Minimize Container Registries to only those approved	Not Scored	L2	Not Default
6.2	**Identity and Access Management (IAM)**			
6.2.1	Prefer not running GKE clusters using the Compute Engine default service account	Scored	L2	Not Default
6.2.2	Prefer using dedicated GCP Service Accounts and Workload Identity	Not Scored	L1	Not Default
6.3	**Cloud Key Management Service (Cloud KMS)**			
6.3.1	Consider encrypting Kubernetes Secrets using keys managed in Cloud KMS	Scored	L1	Not Default
6.4	**Node Metadata**			

#	Recommendation	Scored/ Not Scored	Level	Default Status
6.4.1	Ensure legacy Compute Engine instance metadata APIs are Disabled	Scored	L1	Default
6.4.2	Ensure the GKE Metadata Server is Enabled	Not Scored	L2	Not Default
6.5	**Node Configuration and Maintenance**			
6.5.1	Ensure Container-Optimized OS (COS) is used for GKE node images	Scored	L2	Default
6.5.2	Ensure Node Auto-Repair is enabled for GKE nodes	Scored	L1	Default
6.5.3	Ensure Node Auto-Upgrade is enabled for GKE nodes	Scored	L1	Default
6.5.4	Consider automating GKE version management using Release Channels	Not Scored	L1	Not Default
6.5.5	Ensure Shielded GKE Nodes are Enabled	Not Scored	L1	Not Default
6.5.6	Ensure Integrity Monitoring for Shielded GKE Nodes is Enabled	Not Scored	L1	Not Default
6.5.7	Ensure Secure Boot for Shielded GKE Nodes is Enabled	Not Scored	L2	Not Default
6.6	**Cluster Networking**			
6.6.1	Consider enabling VPC Flow Logs and Intranode Visibility	Not Scored	L2	Not Default
6.6.2	Prefer VPC-native clusters	Scored	L1	Not Default
6.6.3	Ensure Master Authorized Networks is Enabled	Scored	L1	Not Default
6.6.4	Ensure clusters are created with Private Endpoint Enabled and Public Access Disabled	Scored	L2	Not Default
6.6.5	Ensure clusters are created with Private Nodes	Scored	L1	Not Default

#	Recommendation	Scored/ Not Scored	Level	Default Status
6.6.6	Consider firewalling GKE worker nodes	Not Scored	L1	Not Default
6.6.7	Ensure Network Policy is Enabled and set as appropriate	Not Scored	L1	Not Default
6.6.8	Consider using Google-managed SSL Certificates	Not Scored	L2	Depends on Environment
6.7	**Logging**			
6.7.1	Ensure Stackdriver Kubernetes Logging and Monitoring is Enabled	Scored	L1	Default
6.7.2	Consider enabling Linux auditd logging	Not Scored	L2	Not Default
6.8	**Authentication and Authorization**			
6.8.1	Ensure Basic Authentication using static passwords is Disabled	Scored	L1	Default
6.8.2	Ensure authentication using Client Certificates is Disabled	Scored	L1	Default
6.8.3	Consider managing Kubernetes RBAC users with Google Groups for GKE	Not Scored	L2	Not Default
6.8.4	Ensure Legacy Authorization (ABAC) is Disabled	Scored	L1	Default
6.9	**Storage**			
6.9.1	Consider enabling Customer-Managed Encryption Keys (CMEK) for GKE persistent disks (PDs)	Not Scored	L1	Not Default
6.10	**Other Cluster Configurations**			
6.10.1	Ensure Kubernetes Web UI is Disabled	Scored	L1	Default

#	Recommendation	Scored/ Not Scored	Level	Default Status
6.10.2	Ensure that Alpha clusters are not used for production workloads	Scored	L1	Default
6.10.3	Ensure Pod Security Policy is Enabled and set as appropriate	Not Scored	L1	Not Default
6.10.4	Consider GKE Sandbox for running untrusted workloads	Not Scored	L2	Not Default
6.10.5	Prefer enabling Binary Authorization and configuring policy as appropriate	Not Scored	L2	Not Default
6.10.6	Prefer enabling Cloud Security Command Center (Cloud SCC)	Not Scored	L1	Not Default

How to audit Benchmarks

Specific instructions for auditing each Recommendation is available as part of the relevant CIS Benchmark. However, you may wish to automate some of these checks to simplify the verification of these controls in your environment. The tools listed below can help with this.

Note that this does not allow you to audit Recommendations from the Kubernetes CIS Benchmark that are not auditable on GKE. For components that you cannot directly audit, see Default values (#default-values) to understand how your environment is already configured by GKE.

Automated auditing of the CIS Kubernetes Benchmark

You can use an open-source tool `kube-bench` (https://github.com/aquasecurity/kube-bench) to test your cluster configuration against the CIS Kubernetes Benchmark. Note that you will be unable to run the `kube-bench master` tests against your GKE workloads, since you do not

have access to the master node directly; and will only be able to run the `kube-bench node` tests.

Make sure to specify the appropriate version, for example,

```
bench node --benchmark cis-1.5
```

Automated auditing of the CIS GKE Benchmark

Security Health Analytics (/security-command-center/docs/how-to-manage-security-health-analytics) identifies common misconfigurations in your environment, such as open firewalls or public buckets. This includes GKE security recommendations (/security-command-center/docs/concepts-security-health-analytics-findings#container-findings). By enabling Security Health Analytics, you'll be notified of cluster misconfigurations you may have in Cloud Security Command Center (/security-command-center).

Many Level 1 Scored Recommendations are covered by corresponding findings in Security Health Analytics.

#	Recommendation	Scored/ Not Scored	Level	Security Health Analytics Finding(s)
6.1	**Image Registry and Image Scanning**			
6.1.1	Ensure Image Vulnerability Scanning using GCR Container Analysis or a third party provider	Scored	L1	n/a
6.1.2	Minimize user access to GCR	Scored	L1	n/a
6.1.3	Minimize cluster access to read-only for GCR	Scored	L1	n/a
6.1.4	Minimize Container Registries to only those approved	Not Scored	L2	n/a

#	Recommendation	Scored/ Not Scored	Level	Security Health Analytics Finding(s)
6.2	**Identity and Access Management (IAM)**			
6.2.1	Prefer not running GKE clusters using the Compute Engine default service account	Scored	L2	`OVER_PRIVILEGED_ACCOUNT` and `OVER_PRIVILEGED_SCOPES`
6.2.2	Prefer using dedicated GCP Service Accounts and Workload Identity	Not Scored	L1	`WORKLOAD_IDENTITY_DISABLED`
6.3	**Cloud Key Management Service (Cloud KMS)**			
6.3.1	Consider encrypting Kubernetes Secrets using keys managed in Cloud KMS	Scored	L1	n/a
6.4	**Node Metadata**			
6.4.1	Ensure legacy Compute Engine instance metadata APIs are Disabled	Scored	L1	`LEGACY_METADATA_ENABLED`
6.4.2	Ensure the GKE Metadata Server is Enabled	Not Scored	L2	n/a
6.5	**Node Configuration and Maintenance**			
6.5.1	Ensure Container-Optimized OS (COS) is used for GKE node images	Scored	L2	`COS_NOT_USED`
6.5.2	Ensure Node Auto-Repair is enabled for GKE nodes	Scored	L1	`AUTO_REPAIR_DISABLED`
6.5.3	Ensure Node Auto-Upgrade is enabled for GKE nodes	Scored	L1	`AUTO_UPGRADE_DISABLED`

#	Recommendation	Scored/ Not Scored	Level	Security Health Analytics Finding(s)
6.5.4	Consider automating GKE version management using Release Channels	Not Scored	L1	n/a
6.5.5	Ensure Shielded GKE Nodes are Enabled	Not Scored	L1	n/a
6.5.6	Ensure Integrity Monitoring for Shielded GKE Nodes is Enabled	Not Scored	L1	n/a
6.5.7	Ensure Secure Boot for Shielded GKE Nodes is Enabled	Not Scored	L2	n/a
6.6	**Cluster Networking**			
6.6.1	Consider enabling VPC Flow Logs and Intranode Visibility	Not Scored	L2	`FLOW_LOGS_DISABLED`
6.6.2	Prefer VPC-native clusters	Scored	L1	`IP_ALIAS_DISABLED`
6.6.3	Ensure Master Authorized Networks is Enabled	Scored	L1	`MASTER_AUTHORIZED_NETWORKS_DISABLED`
6.6.4	Ensure clusters are created with Private Endpoint Enabled and Public Access Disabled	Scored	L2	n/a
6.6.5	Ensure clusters are created with Private Nodes	Scored	L1	`PRIVATE_CLUSTER_DISABLED` checks for the existence of a private cluster config, but not that the nodes are private
6.6.6	Consider firewalling GKE worker nodes	Not Scored	L1	n/a
6.6.7	Ensure Network Policy is Enabled and set as appropriate	Not Scored	L1	`NETWORK_POLICY_DISABLED`

#	Recommendation	Scored/ Not Scored	Level	Security Health Analytics Finding(s)
6.6.8	Consider using Google-managed SSL Certificates	Not Scored	L2	n/a
6.7	**Logging**			
6.7.1	Ensure Stackdriver Kubernetes Logging and Monitoring is Enabled	Scored	L1	`CLUSTER_LOGGING_DISABLED` and `CLUSTER_MONITORING_DISABLED`
6.7.2	Consider enabling Linux auditd logging	Not Scored	L2	n/a
6.8	**Authentication and Authorization**			
6.8.1	Ensure Basic Authentication using static passwords is Disabled	Scored	L1	n/a
6.8.2	Ensure authentication using Client Certificates is Disabled	Scored	L1	n/a
6.8.3	Consider managing Kubernetes RBAC users with Google Groups for GKE	Not Scored	L2	n/a
6.8.4	Ensure Legacy Authorization (ABAC) is Disabled	Scored	L1	`LEGACY_AUTHORIZATION_ENABLED`
6.9	**Storage**			
6.9.1	Consider enabling Customer-Managed Encryption Keys (CMEK) for GKE persistent disks (PDs)	Not Scored	L1	n/a
6.10	**Other Cluster Configurations**			
6.10.1	Ensure Kubernetes Web UI is Disabled	Scored	L1	`WEB_UI_ENABLED`

#	Recommendation	Scored/ Not Scored	Level	Security Health Analytics Finding(s)
6.10.2	Ensure that Alpha clusters are not used for production workloads	Scored	L1	n/a
6.10.3	Ensure Pod Security Policy is Enabled and set as appropriate	Not Scored	L1	`POD_SECURITY_POLICY_DISABLED`
6.10.4	Consider GKE Sandbox for running untrusted workloads	Not Scored	L2	n/a
6.10.5	Prefer enabling Binary Authorization and configuring policy as appropriate	Not Scored	L2	n/a
6.10.6	Prefer enabling Cloud Security Command Center (Cloud SCC)	Not Scored	L1	n/a

What's Next

- Read the GKE security overview (/kubernetes-engine/docs/concepts/security-overview)
- Follow security best practices in the GKE hardening guide (/kubernetes-engine/docs/how-to/hardening-your-cluster)
- Learn more about the shared responsibility model in GKE (https://cloud.google.com/blog/products/containers-kubernetes/exploring-container-security-the-shared-responsibility-model-in-gke-container-security-shared-responsibility-model-gke)

Last updated 2020-06-26 UTC.