# 

**CSWG- Policy As Code Documentation**

Kubernetes Sentinel Policies

4.1.5 Ensure that default service accounts are not actively used.

***Sentinel Policy Name:***

* 4.1.5 Ensure that default service accounts are not actively used

***Category:***

* Kubernetes

***Description of Policy:***

* The default Kubernetes service account should not be used to ensure that rights granted to applications can be more easily audited and reviewed.

***Sentinel Policy Restriction:***

* The default Kubernetes service account should be configured such that it does not provide a service account token and does not have any explicit rights assignments.

***Terraform attributes:***

* + Provider Ref: [kubernetes\_default\_service\_account | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/default_service_account)

***Test cases:***

**Pass cases**

1. For 'google\_compute\_network' ensure that 'automount\_service\_account\_token = false'

"after" : 
"automunt service account token": 
false, 
"metadata" : 
" secret" : 
"timeouts" 
"annotations": null, 
"labels": 
"nane": 
"nanespace": 
[1, 
: null, 
null, 
"default" , 
"terraform -example" , 

**Fail case:**

1. For 'google\_compute\_network' the 'automount\_service\_account\_token' is set to something other than 'false'

"after" : 
"automount serwice account token": true 
n, 
"metadata" : 
"annotations" : 
"labels": 
"name" : 
"namespace" : 
null, 
null, 
"default", 
"terraform•example" , 

**Testcases Output:**

PASS - default Kube service accounts are rx)t used. sentinel 
PASS - test\ckfault Kube service accounts are rot 
logs: 
Create an explicit service account 'hen a Kubernetes wrkload requires $pcific access to the, 
- Rule "main 
Kubernetes API server. 
mt use the defaul 
t service account. 
trace: 
default Kube service accounts are mt used.sentinel 
Value: 
false 
default Kube service accounts are mt used.sentinel - Rule 
Value: 
false 
PASS - hc1 
default Kube service accounts are rnt used.sentinel:29:1 • Rule 
Value: 
true 
default Kube service accounts are r•nt used.sentinel:13:1 - Rule 
Value: 
"Kube Serv accounts are mt used" 
"main" 
"Kube Serv accounts are used" 

4.2.1 Minimize the admission of privileged containers

***Sentinel Policy Name:***

* + 4.2.1 Minimize the admission of privileged containers

***Category:***

* + Kubernetes

***Description of Policy:***

* + Privileged containers have access to all Linux Kernel capabilities and devices. A container running with full privileges can do almost everything that the host can do.
  + There should be at least one PodSecurityPolicy (PSP) defined which does not permit privileged containers.
  + This policy will ensure that the "privileged" attribute is set to false.

***Sentinel Policy Restriction:***

* + Sentinel policy will go through and check for the privileged attribute in the Kubernetes Pod Security policy and make sure it is set to false.

***Terraform attributes:***

* + - Provider Ref:[kubernetes\_pod\_security\_policy | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/pod_security_policy)

***Test cases:***

**Pass cases**

1. If attribute "privileged" is set to false then the pass case will always pass.

"privi leged" : 
false 

**Fail case:**

1. If attribute "privileged" is set to true then the pass case will always fail.

"privileged" : 
true, 

**Testcases Output:**

(base) alexcardona@A1exs—Air—2 CIS—4.2.1—Minimize—The—Admission—of—Privi1eged—Containters % sentinel test 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for test/minimize—the—admission—of—privileged—containers/fail.hcl 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for hcl 
PASS — sentinel 
PASS - test/minimize-the-admission-of-privileged-containers/fail .hcl 
1 Rule 
- Rule 
•148:1 - Rule 
—verbose 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
minimize—the—admission—of—privi leged—containers . sentinel : 148. 
Value: 
false 
mi leged—containers . sentinel : 104 
Value: 
false 
PASS - .hcl 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
run 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
_as_user rule is set to MustRunAsNonRoot 
= false 
nmain" 
"enforce_user" 
= false 
"main" 
ml leged—containers . sentinel . 
. sentinel . 
Value: 
true 
Value: 
true 
• 104:1 
— Rule nenforce_user" 

4.2.2 Minimize the admission of containers wishing to share the host process ID

***Sentinel Policy Name:***

* + 4.2.2 Minimize the admission of containers wishing to share the host process ID

***Category:***

* + Kubernetes

***Description of Policy:***

* + A container running in the host's PID namespace can inspect processes running outside the container. If the container also has access to ptrace capabilities this can be used to escalate privileges outside of the container.
  + There should be at least one PodSecurityPolicy (PSP) defined which does not permit containers to share the host PID namespace.
  + The main purpose of the policy is to not to permit containers to be run with the hostPID flag set to true.

***Sentinel Policy Restriction:***

* + Sentinel Policy will search and filter to find the "host\_pid" attribute and ensure it is set to false.

***Terraform attributes:***

* + - Provider Ref:[kubernetes\_pod\_security\_policy | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/pod_security_policy)

***Test cases:***

**Pass cases**

1. If attribute "host\_pid" is set to false then the pass case will always pass.

false, 

**Fail case:**

1. If attribute "host-pid" is set to true then the pass case will always fail.

"host_pid" : 
true, 

**Testcases Output:**

(base) alexcardona@A1exs—Air—2 CIS—4.2.1—Minimize—The—Admission—of—Privi1eged—Containters % sentinel test 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for test/minimize—the—admission—of—privileged—containers/fail.hcl 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for hcl 
PASS — sentinel 
PASS - test/minimize-the-admission-of-privileged-containers/fail .hcl 
1 Rule 
- Rule 
•148:1 - Rule 
—verbose 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
minimize—the—admission—of—privi leged—containers . sentinel : 148. 
Value: 
false 
mi leged—containers . sentinel : 104 
Value: 
false 
PASS - .hcl 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
run 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
_as_user rule is set to MustRunAsNonRoot 
= false 
nmain" 
"enforce_user" 
= false 
"main" 
ml leged—containers . sentinel . 
. sentinel . 
Value: 
true 
Value: 
true 
• 104:1 
— Rule nenforce_user" 

4.2.3 Minimize the admission wishing to share the host IPC namespace

***Sentinel Policy Name:***

* + 4.2.3 Minimize the admission wishing to share the host IPC namespace

***Category:***

* + Kubernetes

***Description of Policy:***

* + A container running in the host's IPC namespace can use IPC to interact with processes outside the container.
  + There should be at least one PodSecurityPolicy (PSP) defined which does not permit containers to share the host IPC namespace
  + The main purpose of the policy is to ensure that containers are not permitted to run with the "host\_ipc " flag set to true.

***Sentinel Policy Restriction:***

* + Sentinel Policy will search and filter to find the "host\_ipc" attribute and ensure it is set to false.

***Terraform attributes:***

* + - Provider Ref:[kubernetes\_pod\_security\_policy | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/pod_security_policy)

***Test cases:***

**Pass cases**

1. If attribute "host\_ipc" is set to false then the pass case will always pass.

" host_ipc" : 
false, 

**Fail case:**

1. If attribute "host\_ipc" is set to true then the pass case will always fail.

" host_ipc" : 
true, 

**Testcases Output:**

(base) alexcardona@A1exs—Air—2 CIS—4.2.1—Minimize—The—Admission—of—Privi1eged—Containters % sentinel test 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for test/minimize—the—admission—of—privileged—containers/fail.hcl 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for hcl 
PASS — sentinel 
PASS - test/minimize-the-admission-of-privileged-containers/fail .hcl 
1 Rule 
- Rule 
•148:1 - Rule 
—verbose 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
minimize—the—admission—of—privi leged—containers . sentinel : 148. 
Value: 
false 
mi leged—containers . sentinel : 104 
Value: 
false 
PASS - .hcl 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
run 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
_as_user rule is set to MustRunAsNonRoot 
= false 
nmain" 
"enforce_user" 
= false 
"main" 
ml leged—containers . sentinel . 
. sentinel . 
Value: 
true 
Value: 
true 
• 104:1 
— Rule nenforce_user" 

4.2.4 Minimize the admission of containers wishing to share the host network namespace

***Sentinel Policy Name:***

* + 4.2.3 Minimize the admission wishing to share the host IPC namespace

***Category:***

* + Kubernetes

***Description of Policy:***

* + A container running in the host's network namespace could access the local loopback device, and could access network traffic to and from other pods.
  + There should be at least one PodSecurityPolicy (PSP) defined which does not permit containers to share the host network namespace.
  + The main purpose of the policy is to ensure that the hostNetwork flag is not set to true

***Sentinel Policy Restriction:***

* + Sentinel Policy will search and filter to find the "host\_network" attribute and ensure it is set to false.

***Terraform attributes:***

* + - Provider Ref:[kubernetes\_pod\_security\_policy | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/pod_security_policy)

***Test cases:***

**Pass cases**

1. If attribute "host\_network" is set to false then the pass case will always pass.

"host network": 
false, 

**Fail case:**

1. If attribute "host\_network" is set to true then the pass case will always fail.

"host network" : 
true, 

**Testcases Output:**

(base) alexcardona@A1exs—Air—2 CIS—4.2.1—Minimize—The—Admission—of—Privi1eged—Containters % sentinel test 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for test/minimize—the—admission—of—privileged—containers/fail.hcl 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for hcl 
PASS — sentinel 
PASS - test/minimize-the-admission-of-privileged-containers/fail .hcl 
1 Rule 
- Rule 
•148:1 - Rule 
—verbose 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
minimize—the—admission—of—privi leged—containers . sentinel : 148. 
Value: 
false 
mi leged—containers . sentinel : 104 
Value: 
false 
PASS - .hcl 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
run 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
_as_user rule is set to MustRunAsNonRoot 
= false 
nmain" 
"enforce_user" 
= false 
"main" 
ml leged—containers . sentinel . 
. sentinel . 
Value: 
true 
Value: 
true 
• 104:1 
— Rule nenforce_user" 

4.2.5 Minimize the admission of containers with allowPrivilegeEscalation

***Sentinel Policy Name:***

* + 4.2.5 Minimize the admission of containers with allowPrivilegeEscalation

***Category:***

* + Kubernetes

***Description of Policy:***

* + A container running with the allowPrivilegeEscalation flag set to true may have processes that can gain more privileges than their parent.
  + There should be at least one PodSecurityPolicy (PSP) defined which does not permit containers to allow privilege escalation. The option exists (and is defaulted to true) to permit setuid binaries to run
  + The main purpose of the policy is to ensure that containers are not permitted to run with the allowPrivilegeEscalation flag set to true.

***Sentinel Policy Restriction:***

* + Sentinel Policy will search and filter to find the "allowPrivilegeEscalation" attribute and ensure it is set to false. If this attribute is not set to false then the resource will not be created.

***Terraform attributes:***

* + - Provider Ref:[kubernetes\_pod\_security\_policy | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/pod_security_policy)

***Test cases:***

**Pass cases**

1. If attribute "allow\_privilege\_escaltion" is set to false then the pass case will always pass.

" al Iow_p divi lege_escalat ion” : 
false, 

**Fail case:**

1. If attribute "allow\_privilege\_escaltion" is set to true then the pass case will always fail.



**Testcases Output:**

(base) alexcardona@A1exs—Air—2 CIS—4.2.1—Minimize—The—Admission—of—Privi1eged—Containters % sentinel test 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for test/minimize—the—admission—of—privileged—containers/fail.hcl 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for hcl 
PASS — sentinel 
PASS - test/minimize-the-admission-of-privileged-containers/fail .hcl 
1 Rule 
- Rule 
•148:1 - Rule 
—verbose 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
minimize—the—admission—of—privi leged—containers . sentinel : 148. 
Value: 
false 
mi leged—containers . sentinel : 104 
Value: 
false 
PASS - .hcl 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
run 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
_as_user rule is set to MustRunAsNonRoot 
= false 
nmain" 
"enforce_user" 
= false 
"main" 
ml leged—containers . sentinel . 
. sentinel . 
Value: 
true 
Value: 
true 
• 104:1 
— Rule nenforce_user" 

4.2.6 Minimize the admission of root containers

***Sentinel Policy Name:***

* + 4.2.6 Minimize the admission of root containers

***Category:***

* + Kubernetes

***Description of Policy:***

* + Containers may run as any Linux user. Containers which run as the root user, whilst constrained by Container Runtime security features still have a escalated likelihood of container breakout.
  + There should be at least one PodSecurityPolicy (PSP) defined which does not permit root users in a container
  + The main purpose of the policy is to ensure that containers are not permitted to run with the "run\_as\_user" flag set to root.

***Sentinel Policy Restriction:***

* + Sentinel Policy will search and filter to find the "run\_as\_user" attribute and ensure it is not set to root.

***Terraform attributes:***

* + - Provider Ref:[kubernetes\_pod\_security\_policy | Resources | hashicorp/kubernetes | Terraform Registry](https://registry.terraform.io/providers/hashicorp/kubernetes/latest/docs/resources/pod_security_policy)

***Test cases:***

**Pass cases**

1. If attribute "run\_as\_user" is set to MustRunAsNonRoot then the pass case will always pass.

" range" : 
[l, 
"MustRunAsNonRoot" , 

**Fail case:**

1. If attribute "run\_as\_user" is set to root then the pass case will always fail.

"run as 
user" : 
" range" : 
" rule": 
"Root" , 

**Testcases Output:**

(base) alexcardona@A1exs—Air—2 CIS—4.2.1—Minimize—The—Admission—of—Privi1eged—Containters % sentinel test 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for test/minimize—the—admission—of—privileged—containers/fail.hcl 
Installing test modules for 
— Module tfplan—functions marked for installation 
Installation complete for hcl 
PASS — sentinel 
PASS - test/minimize-the-admission-of-privileged-containers/fail .hcl 
1 Rule 
- Rule 
•148:1 - Rule 
—verbose 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
minimize—the—admission—of—privi leged—containers . sentinel : 148. 
Value: 
false 
mi leged—containers . sentinel : 104 
Value: 
false 
PASS - .hcl 
logs: 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
Ensure 
trace: 
that 
that 
that 
that 
that 
that 
that 
the 
the 
the 
the 
the 
the 
run 
admission of privileged containers are set to false 
host proccess ID is set to false. Example host_pid 
host_ipc is set to false. 
host_ipc is set to false. 
host network is set to false. 
allow_privilege_escalation is set to false. 
_as_user rule is set to MustRunAsNonRoot 
= false 
nmain" 
"enforce_user" 
= false 
"main" 
ml leged—containers . sentinel . 
. sentinel . 
Value: 
true 
Value: 
true 
• 104:1 
— Rule nenforce_user" 

4.2.8 Minimize the admission of containers with added capabilities

***Sentinel Policy Name:***

* + 4.2.8 Minimize the admission of containers with added capabilities

***Category:***

* + Kubenetes

***Description of Policy:***

* + As per CIS benchmark , It is not recommended to permit containers with capabilities assigned beyond the default set.
  + Containers run with a default set of capabilities as assigned by the Container Runtime. Capabilities outside this set can be added to containers which could expose them to risks of container breakout attacks.
  + Policy will simply check for the allowedCapabilities attribute and ensure this value is empty.

***Sentinel Policy Restriction:***

* + Sentinel policy will ensure that the attribute named "allowedCapabilities" in a Kubernetes Pod Security Policy is set to empty and that there are no values in there.

***Test cases:***

**Pass cases**

1. If attribute "allowed\_capabilites" is set to empty then the pass case will always pass.

'la t towed _ capabilities" : 
"al towed flex volumes" : 

**Fail case:**

1. If attribute "allowed\_capabiliteis" is set to anything then the fail case will always fail.

'la I Iow_p rivi lege_esca lation" : 
"al Iowed_+pabilities" : 
fa use, 
["test"] , 

**Testcases Output:**

(base) 
CIS—4.2.8 
— sentinel 
PASS 
PASS 
— hcl 
tes t 'pass . hcl 
PASS - 
% sentinel test 

5.1.1 Ensure Image Vulnerability Scanning using GCR Container Analysis or a third party provider

***Sentinel Policy Name:***

5.1.1 Ensure Image Vulnerability Scanning using GCR Container Analysis or a third party provider

***Category:***

* + Kubernetes

***Description of Policy:***

* + Scan images stored in Google Container Registry (GCR) for vulnerabilities.
  + Vulnerabilities in software packages can be exploited by hackers or malicious users to obtain unauthorized access to local cloud resources. GCR Container Analysis and other third party products allow images stored in GCR to be scanned for known vulnerabilities.

***Sentinel Policy Restriction:***

* + The policy checks whether "**service**" is set to "**containerscanning.googleapis.com**"

***Terraform attributes:***

* + - Provider Ref:

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/google_project_service>

* + - Terraform attribute: "service"

***Test cases:***

**Pass cases**

1. In pass case, if "service" is set to "containerscanning.googleapis.com" then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "service" is not set to "containerscanning.googleapis.com" then the pass case will always fail.

**No Screenshot of argument in mock file**

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test -run=s.l.l-kubernetes-ensure-image-vulnerability-scanning-using-gcr-container-analysis 
S . 1 . 1 - kubernetes - ensure - image - vulnerability - scanning- using- gcr - container - analysis . sentinel 
PASS 
test'S . 1 . 1 - kubernetes - ensure - image - vulnerability - scanning- using- gcr - container - analysis/fail . hcl 
PASS 
test'S . 1 . 1 - kubernetes - ensure - image - vulnerability - scanning- using- gcr - container - analysis/ pass . hcl 
PASS 

5.1.4 Minimize Container Registries to only those Approved

***Sentinel Policy Name:***

* + 5.1.4 Minimize Container Registries to only those Approved

***Category:***

* + Kubernetes

***Description of Policy:***

* + Allowing unrestricted access to external container registries provides the opportunity for malicious or unapproved containers to be deployed into the cluster. Allowlisting only approved container registries reduces this risk.

***Sentinel Policy Restriction:***

* + This policy will ensure the "evaluation\_mode" value under "binary\_authorization" for the google\_container\_cluster resource is set to: ***PROJECT\_SINGLETON\_POLICY\_ENFORCE***
  + This policy will ensure the "evaluation\_mode" value under "default\_admission\_rule" for the google\_binary\_authorization\_policy resource is set to: ***REQUIRE\_ATTESTATION*** or ***ALWAYS\_DENY*** *NOT* ***ALWAYS\_ALLOW***

***Terraform attributes:***

* + Provider Ref:
    1. <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    2. <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/binary_authorization_policy>
  + Filter:
    1. google\_container\_cluster: *binary\_authorization >* evaluation\_mode
    2. google\_binary\_authorization\_policy: default\_admission\_rule > evaluation\_mode

***Test cases:***

**Pass cases**

* + google\_container\_cluster filter set to CIS recommendation.

"binary_authorization" : [ 
null, 
"enabled": 
-evaluation mde": "PRmECT SINGLETON POLICY ENFORCE 

* + google\_binary\_authorization\_policy filter set to CIS recommendation.

"enforcenpnt node" : 
"evaluation mode": 
" requi : 
"ENFORCED BLOCK AUDIT 
"ALWAYS DENV 
null, 

**Fail case:**

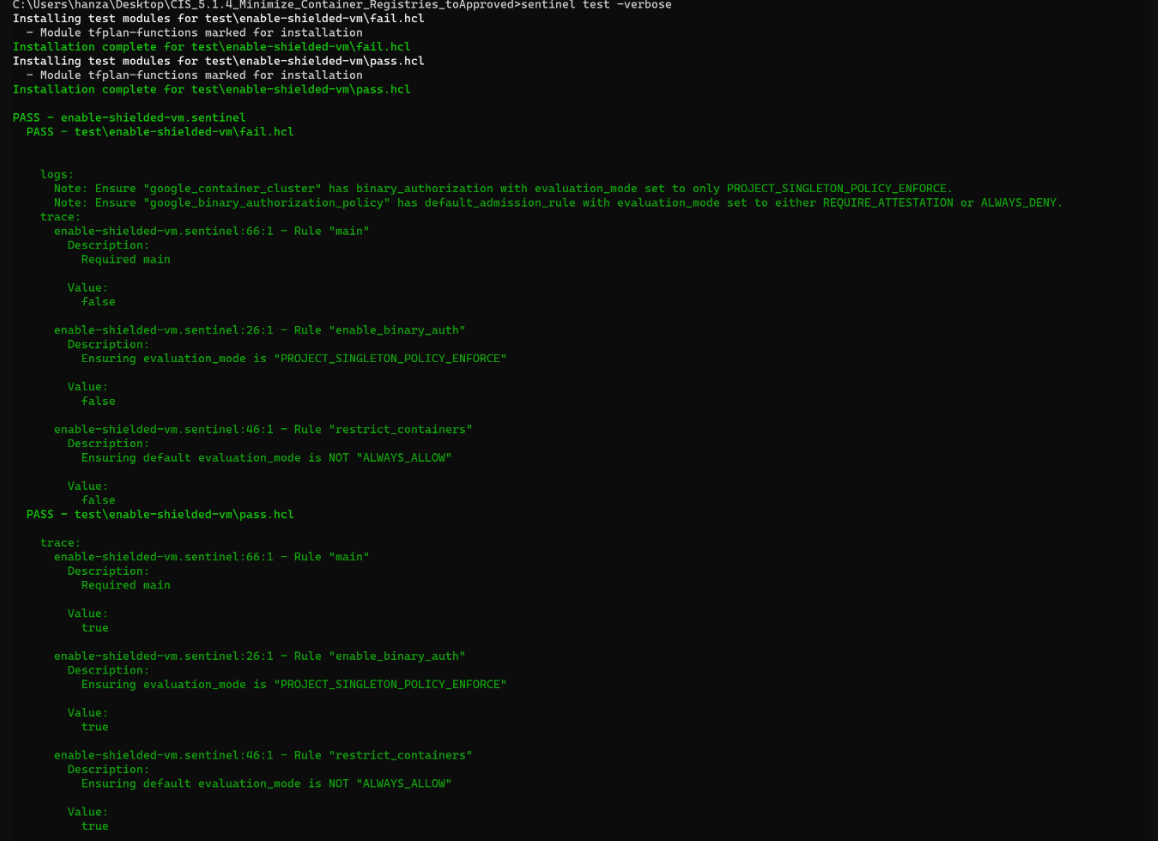
* 1. google\_container\_cluster filter not set to CIS recommendation.

"after" : 
"binary_authorization" : [ 
"enabl ed" : 
null, 
•evaluation . 
_ enabled% 

* 1. google\_binary\_authorization\_policy filter not set to CIS recommendation

"enforcement mode": 
"ENFORCED BLOCK AND AUDIT L(Xi", 
"evaluation mode": 
"require_attestations_by" • 
. null, 

**Testcases Output:**



5.2.1 Ensure GKE clusters are not running using the Compute Engine default service account

***Sentinel Policy Name:***

5.2.1 Ensure GKE clusters are not running using the Compute Engine default service account

***Category:***

* + Kubernetes

***Description of Policy:***

* + Create and use minimally privileged Service accounts to run GKE cluster nodes instead of using the Compute Engine default Service account. Unnecessary permissions could be abused in the case of a node compromise.
  + You should create and use a minimally privileged service account to run your Kubernetes Engine cluster instead of using the Compute Engine default service account, and create separate service accounts for each Kubernetes Workload.
  + Kubernetes Engine requires, at a minimum, the node service account to have the monitoring.viewer, monitoring.metricWriter, and logging.logWriter roles. Additional roles may need to be added for the nodes to pull images from GCR.

***Sentinel Policy Restriction:***

* + The policy checks whether "**service\_account**" is specified under the "**node\_config**" block

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "service\_account" inside the "node\_config" block

The "service\_account" would be used by the Node VMs. If not specified, the "default" service account is used.

***Test cases:***

**Pass cases**

1. In pass case, if "service\_account" is specified under the "node\_config" block for both resources then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "service\_account" is not specified under the "node\_config" block for both resources, then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-GKE-c1usters-are-not-running-using-computeengine-defau1t-service-account 
kubernetes - ensure -GKE - clusters - are - not - running- using- computeengine - default - service - account . sentinel 
PASS 
test/kubernetes - ensure -GKE - clusters - are - not - running- using- computeengine - default - service - account/fail . hcl 
PASS 
test/kubernetes - ensure -GKE - clusters - are - not - running- using- computeengine - default - service - account/ pass . hcl 
PASS 

5.2.2 Prefer using dedicated GCP Service Accounts and Workload Identity

***Sentinel Policy Name:***

5.2.2 Prefer using dedicated GCP Service Accounts and Workload Identity

***Category:***

* + Kubernetes

***Description of Policy:***

* + Kubernetes workloads should not use cluster node service accounts to authenticate to Google Cloud APIs. Each Kubernetes Workload that needs to authenticate to other Google services using Cloud IAM should be provisioned a dedicated Service account. Enabling Workload Identity manages the distribution and rotation of Service account keys for the workloads to use.
  + Manual approaches for authenticating Kubernetes workloads running on GKE against Google Cloud APIs are: storing service account keys as a Kubernetes secret (which introduces manual key rotation and potential for key compromise); or use of the underlying nodes' IAM Service account, which violates the principle of least privilege on a multitenanted node, when one pod needs to have access to a service, but every other pod on the node that uses the Service account does not.
  + Once a relationship between a Kubernetes Service account and a GCP Service account has been configured, any workload running as the Kubernetes Service account automatically authenticates as the mapped GCP Service account when accessing Google Cloud APIs on a cluster with Workload Identity enabled.

***Sentinel Policy Restriction:***

* + The policy checks whether "**workload\_pool**" matches regex expression "**[0-9]+.svc.id.goog**" where [0-9]+ represents the project id and **"mode"** is set to "**GKE\_METADATA"**

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
      * Terraform attribute: "workload\_pool" inside the "workload\_identity\_config" block
      * Terraform attribute: "mode" from "workload\_metadata\_config" inside the "node\_config" block
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
      * Terraform attribute: "mode" from "workload\_metadata\_config" inside the "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if

"workload\_pool" matches regex expression "[0-9]+.svc.id.goog" where [0-9]+ represents the project id and

"mode" is set to "GKE\_METADATA" then the pass case will always pass.

Screenshot of argument in mock file

For workload\_identity\_config" block:

"workload_pool " • " 
132794818113. svc . id. goog" 

For "node\_config" block:

"workload_metadata_config" : 
"mode": ' GKE METADATA", 

**Fail case:**

1. In pass case , if

"workload\_pool" does not match regex expression "[0-9]+.svc.id.goog" where [0-9]+ represents the project id and

"mode" is not set to "GKE\_METADATA" then the pass case will always fail.

**No screenshot of argument in mock file**

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test -run=S.2.2-kubernetes-prefer-using-dedicated-gcp-svc-accounts-and-work10ad-identity 
S . 2 .2-kubernetes-prefer -using- dedicated-gcp-svc-accounts-and-workload-identity . sentinel 
PASS 
test'S . 2 .2-kubernetes-prefer -using- dedicated-gcp-svc-accounts-and-workload-identity/fail . hcl 
PASS 
test'S . 2 .2-kubernetes-prefer -using- dedicated-gcp-svc-accounts-and-workload-identity/pass . hcl 
PASS 

5.3.1 Ensure Kubernetes Secrets are encrypted using keys managed in Cloud KMS

***Sentinel Policy Name:***

5.3.1 Ensure Kubernetes Secrets are encrypted using keys managed in Cloud KMS

***Category:***

* + Kubernetes

***Description of Policy:***

* + Encrypt Kubernetes secrets, stored in etcd, at the application-layer using a customer-managed key in Cloud KMS.
  + By default, GKE encrypts customer content stored at rest, including Secrets. GKE handles and manages this default encryption for you without any additional action on your part.

Application-layer Secrets Encryption provides an additional layer of security for sensitive data, such as user defined Secrets and Secrets required for the operation of the cluster, such as service account keys, which are all stored in etcd.

Using this functionality, you can use a key, that you manage in Cloud KMS, to encrypt data at the application layer. This protects against attackers in the event that they manage to gain access to etcd.

Impact:

To use the Cloud KMS CryptoKey to protect etcd in the cluster, the 'Kubernetes Engine Service Agent' Service account must hold the 'Cloud KMS CryptoKey Encrypter/Decrypter' role.

***Sentinel Policy Restriction:***

The policy checks whether if "state" is set to "ENCRYPTED" and key\_name contains value

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "state" and "key\_name" from ""database\_encryption" block

***Test cases:***

**Pass cases**

1. In pass case, if "state" is set to "ENCRYPTED" and key\_name contains value, then the pass case will always pass.

Screenshot of argument in mock file

" database_encryption" : 
" key _ name" : 
" state" . 
" projects [my -proj ect/ locat ions/globa 1 / keyRi ngs /my- ri ng/cryptoKeys /my- key" , 
"ENCRYPTED" , 

**Fail case:**

1. In pass case, if "state" is not set to "ENCRYPTED" or key\_name does not contain value, then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e 
,/ALL Wls/sentinel 
hashicorp/policytesting$ 
Installing test modules for 
Installing test modules for 
kubernetes-ensure-kubernetes-secrets 
kubernetes-ensure-kubernetes-secrets 
kubernetes-secrets-are 
kubernetes-secrets-are 
kubernetes-secrets-are 
sentinel test 
test/kubernetes-ensure- kubernetes- secrets- are-encrypted- using- keys-managed- in- cloudKmS/fai1. hcl 
test/ kubernetes- ensure- kubernetes- secrets- are- encrypted- usl ng- keys-managed- in- cloudKl•1S/ pass . hcl 
- run= 
kubernetes - ensure- kubernetes - secrets- are- encrypt 
-using- keys-manag 
-in- cloudKl•1S 
- ver 
se 
kubernetes - ensure - kubernetes - secrets - are - encrypted - using- keys -managed - in - cloudKMS . sentinel 
PASS 
test/kubernetes - ensure - kubernetes - secrets - are - encrypted - using- keys -managed - in - cloudKMS/fai1. hcl 
PASS 
logs : 
Ensure Kubernetes 
trace: 
Value: 
false 
Value: 
false 
Secrets stored in etcd 
-are 
-are 
are encrypted at the 
application layers using keys managed in Cloud K'•IS 
- encrypted - using- keys -managed - in - cloudK'•1S . sentinel : 34 : 1 
- encrypted - using- keys -managed-in - cloudK'•1S . sentinel : 18 : 1 
Rule 
Rule 
"main" 
" database _ 
PASS 
test/kubernetes - ensure - kubernetes - secrets - are - encrypted - using- keys -managed - in - cloudKMS/pass . hcl 
logs : 
Ensure Kubernetes 
Secrets stored in etcd 
are encrypted at the 
application layers using keys managed in Cloud K'•IS 
trace: 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
-ensure- 
-ensure- 
-ensure- 
- encrypted - using- keys -managed - in - cloudK'•1S . sentinel : 34 
- encrypted - using- keys -managed - in - cloudK'•1S . sentinel : 18 
- encrypted - using- keys -managed - in - cloudK'•1S . sentinel : 26 
Rule 
Rule 
Rule 
"main" 
" database _ 
"kms is enabled" 

5.4.1 Ensure legacy Compute Engine instance metadata APIs are Disabled

***Sentinel Policy Name:***

5.4.1 Ensure legacy Compute Engine instance metadata APIs are Disabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Disable the legacy GCE instance metadata APIs for GKE nodes. Under some circumstances, these can be used from within a pod to extract the node's credentials.
  + Without requiring a custom HTTP header when accessing the legacy GCE metadata endpoint, a flaw in an application that allows an attacker to trick the code into retrieving the contents of an attacker-specified web URL could provide a simple method for enumeration and potential credential exfiltration. By requiring a custom HTTP header, the attacker needs to exploit an application flaw that allows them to control the URL and also add custom headers in order to carry out this attack successfully.

***Sentinel Policy Restriction:***

* + The policy checks whether if "**disable-legacy-endpoints**" is set to true

***Terraform attributes:***

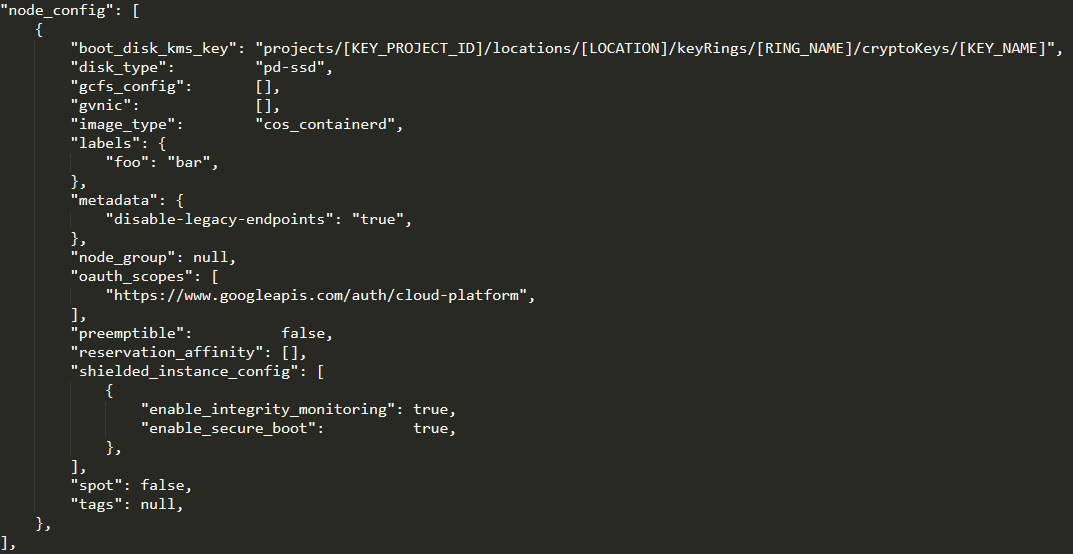
* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "disable-legacy-endpoints" under "metadata" inside the "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, If "disable-legacy-endpoints" is set to true for both resources then the pass case will always pass.

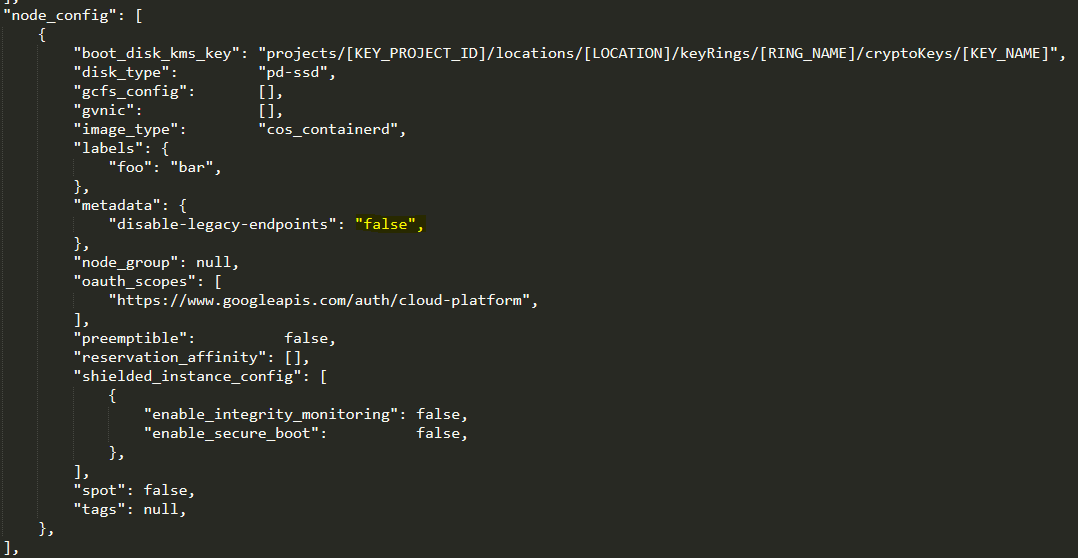
Screenshot of argument in mock file



**Fail case:**

1. In pass case, If "disable-legacy-endpoints" is not set to true for both resources then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e/ALL 
Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-1egacy-compute-engine-metadata-APIs-are-disab1ed 
kubernetes - ensure - legacy - compute - engine -metadata -APIs - are - disabled . sentinel 
PASS 
test/kubernetes - ensure - legacy - compute - engine -metadata -APIs - are - disabled/fail . hcl 
PASS 
test/kubernetes - ensure - legacy - compute - engine -metadata -APIs - are - disabled/ pass . hcl 
PASS 

5.4.2 Ensure the GKE Metadata Server is Enabled

***Sentinel Policy Name:***

5.4.2 Ensure the GKE Metadata Server is Enabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Running the GKE Metadata Server prevents workloads from accessing sensitive instance metadata and facilitates Workload Identity
  + Every node stores its metadata on a metadata server. Some of this metadata, such as kubelet credentials and the VM instance identity token, is sensitive and should not be exposed to a Kubernetes workload. Enabling the GKE Metadata server prevents pods (that are not running on the host network) from accessing this metadata and facilitates Workload Identity.

* + When unspecified, the default setting allows running pods to have full access to the node's underlying metadata server.

***Sentinel Policy Restriction:***

* + The policy checks whether "**mode**" is set to "**GKE\_METADATA**"

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
      * Terraform attribute: "mode" from "workload\_metadata\_config" inside the "node\_config" block

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>

* + - * Terraform attribute: "mode" from "workload\_metadata\_config" inside the "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "mode" is set to "GKE\_METADATA" then the pass case will always pass.

Screenshot of argument in mock file

"workload_metadata_config" : 
"mode": ' GKE METADATA", 

**Fail case:**

1. In pass case ,if "mode" is not set to "GKE\_METADATA" then the pass case will always fail.

**No Screenshot of argument in mock file**

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
S .4.2-kubernetes-ensure-gke -metadata-server -is-enabled . sentinel 
PASS 
test'S . 4.2-kubernetes-ensure-gke -metadata-server -is-enabled/fail . hcl 
PASS 
test'S . 4.2-kubernetes-ensure-gke -metadata-server -is-enabled/pass . hcl 
PASS 
- run-S . 4.2- kubernetes-ensure- gke-metadata- server- is-enabled 

5.5.1 Ensure Container-Optimized OS (COS) is used for GKE node images

***Sentinel Policy Name:***

5.5.1 Ensure Container-Optimized OS (COS) is used for GKE node images

***Category:***

* + Kubernetes

***Description of Policy:***

* + Use Container-Optimized OS (COS) as a managed, optimized and hardened base OS that limits the host's attack surface. COS is an operating system image for Compute Engine VMs optimized for running containers. With COS, you can bring up your containers on Google Cloud Platform quickly, efficiently, and securely.
  + NOTE : COS AND UBUNTU are deprecated as of GKE 1.24. In GKE version 1.24 and later, Docker-based node image types are not supported. In GKE version 1.23, you also cannot create new node pools with Docker node image types. You must migrate to a containerd node image type. To learn more about this change, see Migrating from Docker to containerd in GKE documentation.

***Sentinel Policy Restriction:***

* + The policy checks whether "**image\_type**" is set to "**cos**" or "**cos\_containerd**" for both resources

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "image\_type" inside the "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "image\_type" is set to "cos" or "cos\_containerd" for both resources, then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case, if "image\_type" is not set to "cos" or "cos\_containerd" for both resources, then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-container-optimized-osis-used-for-GKE-node-images 
kubernetes - ensure - container -optimized -osis - used -for -GKE - node - images . sentinel 
PASS 
test/kubernetes - ensure - container -optimized -osis - used -for -GKE - node - images/fail . hcl 
PASS 
test/kubernetes - ensure - container -optimized -osis - used -for -GKE - node - images/ pass . hcl 
PASS 

5.5.2 Ensure Node Auto-Repair is enabled for GKE nodes

***Sentinel Policy Name:***

5.5.2 Ensure Node Auto-Repair is enabled for GKE nodes

***Category:***

* + Kubernetes

***Description of Policy:***

* + Nodes in a degraded state are an unknown quantity and so may pose a security risk.
  + Kubernetes Engine's node auto-repair feature helps you keep the nodes in your cluster in a healthy, running state. When enabled, Kubernetes Engine makes periodic checks on the health state of each node in your cluster. If a node fails consecutive health checks over an extended time period, Kubernetes Engine initiates a repair process for that node.

***Sentinel Policy Restriction:***

* + The policy checks whether "**auto\_repair**" is set to **true**

***Terraform attributes:***

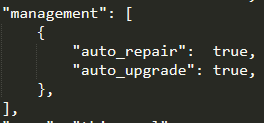
* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "auto\_repair" inside the "management" block

***Test cases:***

**Pass cases**

1. In pass case, if "auto\_repair" is set to true then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "auto\_repair" is not set to true then the pass case will always fail.

Screenshot of argument in mock file

management" . 
" auto 
repałr" : 
" auto_upgrade" . 
false, 
false, 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
kubernetes - ensure - auto- repair - is - enabled -for -GKE - nodes . sentinel 
PASS 
test/kubernetes - ensure - auto- repair - is - enabled -for -GKE - nodes/fail . hcl 
PASS 
test/kubernetes - ensure - auto- repair - is - enabled -for -GKE - nodes/ pass . hcl 
PASS 
- run=kubernetes- ensure- auto- repair- is- enabled- for- GKE- nodes 

5.5.3 Ensure Node Auto-Upgrade is enabled for GKE nodes

***Sentinel Policy Name:***

5.5.3 Ensure Node Auto-Upgrade is enabled for GKE nodes

***Category:***

* + Kubernetes

***Description of Policy:***

* + Node auto-upgrade keeps nodes at the current Kubernetes and OS security patch level to mitigate known vulnerabilities. Node auto-upgrade helps you keep the nodes in your cluster or Node pool up to date with the latest stable patch version of Kubernetes as well as the underlying node operating system. Node auto-upgrade uses the same update mechanism as manual node upgrades.
  + Node pools with node auto-upgrade enabled are automatically scheduled for upgrades when a new stable Kubernetes version becomes available. When the upgrade is performed, the Node pool is upgraded to match the current cluster master version. From a security perspective, this has the benefit of applying security updates automatically to the Kubernetes Engine when security fixes are released.

***Sentinel Policy Restriction:***

* + The policy checks whether "**auto\_upgrade**" is set to **true**

***Terraform attributes:***

* + Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
  + Terraform attribute: "auto\_upgrade" inside the "management" block

***Test cases:***

**Pass cases**

1. In pass case, if "auto\_upgrade" is set to true then the pass case will always pass.

Screenshot of argument in mock file

management" . 
" auto 
repałr" : 
" auto_upgrade" : 
true, 
true, 

**Fail case:**

1. In pass case ,if "auto\_upgrade" is not set to true then the pass case will always fail.

Screenshot of argument in mock file

management" . 
" auto 
repałr" : 
" auto_upgrade" . 
false, 
false, 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
kubernetes - ensure - node - autoupgrades - is - enabled -for -GKE - nodes . sentinel 
PASS 
test/kubernetes - ensure - node - autoupgrades - is - enabled -for -GKE - nodes/fail . hcl 
PASS 
test/kubernetes - ensure - node - autoupgrades - is - enabled -for -GKE - nodes/ pass . hcl 
PASS 
- run=kubernetes- ensure- node- autoupgrades- is- enabled- for- GKE- nodes 

5.5.4 When creating New Clusters - Automate GKE version management using Release Channels

***Sentinel Policy Name:***

5.5.4 When creating New Clusters - Automate GKE version management using Release Channels

***Category:***

* + Kubernetes

***Description of Policy:***

* + Subscribe to the Regular or Stable Release Channel to automate version upgrades to the GKE cluster and to reduce version management complexity to the number of features and level of stability required.
  + Release Channels signal a graduating level of stability and production-readiness. These are based on observed performance of GKE clusters running that version and represent experience and confidence in the cluster version.
  + The Regular release channel upgrades every few weeks and is for production users who need features not yet offered in the Stable channel. These versions have passed internal validation, but don't have enough historical data to guarantee their stability. Known issues generally have known workarounds.
  + The Stable release channel upgrades every few months and is for production users who need stability above all else, and for whom frequent upgrades are too risky. These versions have passed internal validation and have been shown to be stable and reliable in production, based on the observed performance of those clusters.
  + Critical security patches are delivered to all release channels.

***Sentinel Policy Restriction:***

* + The policy checks whether "**channel**" is set to either "**regula**r" or "**stable**"

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "channel" inside the "release\_channel" block

***Test cases:***

**Pass cases**

1. In pass case, if "channel" is set to either "regular" or "stable" , then the pass case will always pass.

Screenshot of argument in mock file

" release 
channel" 
" channel 
"STABLE" 

OR

" release 
channel" 
" channel 
"REGULAR", 

**Fail case:**

1. In pass case, if "channel" is NOT set to either "regular" or "stable" , then the pass case will always fail.

Screenshot of argument in mock file

" release 
channel" 
" channel 
"UNSPECIFIED", 

**OR**

" release 
channel" 
" channel 
"RAPID", 

**Testcases Output:**

Screenshot of test command successful

Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-automate-gke-version-management-using-release-channels 
kubernetes - automate - gke - version -management - using- release - channels . sentinel 
PASS 
PASS 
PASS 
PASS 
PASS 
test/kubernetes - automate - gke - version -management - using- release - channels/fail . hcl 
test/kubernetes - automate - gke - version -management - using- release - channels/fai12. hcl 
test/kubernetes - automate - gke - version -management - using- release - channels/ pass . hcl 
test/kubernetes - automate - gke - version -management - using- release - channels/ pass2 . hcl 

5.5.5 Ensure Shielded GKE Nodes are Enabled

***Sentinel Policy Name:***

5.5.5 Ensure Shielded GKE Nodes are Enabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Shielded GKE Nodes provides verifiable integrity via secure boot, virtual trusted platform module (vTPM)-enabled measured boot, and integrity monitoring.
  + Shielded GKE nodes protects clusters against boot- or kernel-level malware or rootkits which persist beyond infected OS.

* + Shielded GKE nodes run firmware which is signed and verified using Google's Certificate Authority, ensuring that the nodes' firmware is unmodified and establishing the root of trust for Secure Boot. GKE node identity is strongly protected via virtual Trusted Platform Module (vTPM) and verified remotely by the master node before the node joins the cluster. Lastly, GKE node integrity (i.e., boot sequence and kernel) is measured and can be monitored and verified remotely.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_shielded\_nodes**" is set to **true**

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enable\_shielded\_nodes"

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_shielded\_nodes" is set to true then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "enable\_shielded\_nodes" is not set to true then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
kubernetes - ensure - shielded - gke - nodes - are - enabled . sentinel 
PASS 
test/kubernetes - ensure - shielded - gke - nodes - are - enabled/fail . hcl 
PASS 
test/kubernetes - ensure - shielded - gke - nodes - are - enabled/ pass . hcl 
PASS 
- run=kubernetes- ensure- shielded- gke- nodes- are- enabled 

5.5.6 Ensure Integrity Monitoring for Shielded GKE Nodes is Enabled

***Sentinel Policy Name:***

5.5.6 Ensure Integrity Monitoring for Shielded GKE Nodes is Enabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Enable Integrity Monitoring for Shielded GKE Nodes to be notified of inconsistencies during the node boot sequence.
  + Integrity Monitoring provides active alerting for Shielded GKE nodes which allows administrators to respond to integrity failures and prevent compromised nodes from being deployed into the cluster.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_integrity\_monitoring**" is set to **true**

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "enable\_integrity\_monitoring" inside the "shielded\_instance\_config" block under "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_integrity\_monitoring" is set to true for both resources then the pass case will always pass.

Screenshot of argument in mock file

" shielded_instance_config" : 
"enable integrity_monitoring" : 
"enable secure boot" 
true , 
true, 

**Fail case:**

1. In pass case ,if "enable\_integrity\_monitoring" is not set to true for both resources then the pass case will always fail.

Screenshot of argument in mock file

" shielded_instance_config" : 
"enable integrity_monitoring . 
"enable secure boot" 
false, 
false, 

**Testcases Output:**

Screenshot of test command successful

Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-integrity-monitoring-for-shielded-gke-nodes-are-enabled 
kubernetes - ensure - integrity -monitoring-for - shielded - gke - nodes - are - enabled . sentinel 
PASS 
test/kubernetes - ensure - integrity -monitoring-for - shielded - gke - nodes - are - enabled/fail . hcl 
PASS 
test/kubernetes - ensure - integrity -monitoring-for - shielded - gke - nodes - are - enabled/ pass . hcl 
PASS 

5.5.7 Ensure Secure Boot for Shielded GKE Nodes is Enabled

***Sentinel Policy Name:***

5.5.7 Ensure Secure Boot for Shielded GKE Nodes is Enabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Enable Secure Boot for Shielded GKE Nodes to verify the digital signature of node boot components.
  + An attacker may seek to alter boot components to persist malware or root kits during system initialization. Secure Boot helps ensure that the system only runs authentic software by verifying the digital signature of all boot components, and halting the boot process if signature verification fails.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_secure\_boot**" is set to **true**

***Terraform attributes:***

* + - Provider Ref:
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "enable\_secure\_boot" inside the "shielded\_instance\_config" block under "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_secure\_boot" is set to true for both resources then the pass case will always pass.

Screenshot of argument in mock file

" shielded_instance_config" : 
"enable integrity_monitoring" : 
"enable secure boot" 
true , 
true, 

**Fail case:**

1. In pass case ,if "enable\_secure\_boot" is not set to true for both resources then the pass case will always fail.

Screenshot of argument in mock file

" shielded_instance_config" : 
"enable integrity_monitoring . 
"enable secure boot" 
false, 
false, 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e/ALL 
PASS 
PASS 
PASS 
Wls/sentinel_hashicorp/policytesting$ sentinel test 
kubernetes - ensure - secure - boot -for - shielded - gke - nodes - are - enabled . sentinel 
test/kubernetes - ensure - secure - boot -for - shielded - gke - nodes - are - enabled/fail . hcl 
test/kubernetes - ensure - secure - boot -for - shielded - gke - nodes - are - enabled/ pass . hcl 
- run = kubernetes- ensure- secure- boot - for- shielded- gke- nodes- are- enabled 

5.6.1 Enable VPC Flow Logs and Intranode Visibility

***Sentinel Policy Name:***

5.6.1 Enable VPC Flow Logs and Intranode Visibility

***Category:***

* + Kubernetes

***Description of Policy:***

* + Enable VPC Flow Logs and Intranode Visibility to see pod-level traffic, even for traffic within a worker node.
  + Enabling Intranode Visibility makes your intranode pod to pod traffic visible to the networking fabric. With this feature, you can use VPC Flow Logs or other VPC features for intranode traffic.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_intranode\_visibility**" is set to true

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enable\_intranode\_visibility"

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_intranode\_visibility" is set to true then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "enable\_intranode\_visibility" is not set to true then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e 
Installing test modules for 
Installing test modules for 
,/ALL Wls/sentinel 
hashicorp/policytesting$ 
sentinel test -run=kubernetes-ensure-vpcflowlogs-and-intran 
test/ kubernetes- ensure- vpcflowlogs- and- intranode- visi bility/fail . h cl 
test/ kubernetes- ensure- vpcflowlogs- and- intranode- vi si bi lity/ pass . h cl 
e-visibility 
- ver 
se 
kubernetes - ensure - vpcflowlogs - and - intranode - visibility . sentinel 
PASS 
test/kubernetes - ensure - vpcflowlogs - and - intranode - visibility 'fail . hcl 
PASS 
logs : 
Ensure VPC 
trace: 
Value: 
false 
Flow Logs and Intranode Visibility to see pod-level traffic, even for traffic within a worker node 
kubernetes - ensure - vpcflowlogs - and - intranode - visibility . sentinel : 18 : 1 
kubernetes - ensure - vpcflowlogs - and - intranode - visibility . sentinel : 12 : 1 
Value: 
false 
test/kubernetes - ensure - vpcflowlogs - and - intranode - visibility / pass . hcl 
Rule 
Rule 
"main" 
"enable intranode_visibility_enabled" 
PASS 
logs : 
Ensure VPC 
trace: 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
Flow Logs and Intranode Visibility to see pod-level traffic, even for traffic within a worker node 
-ensure- 
-ensure- 
vpcflowlogs - and- intranode - visibility . sentinel : 18 
vpcflowlogs - and- intranode - visibility . sentinel : 12 
Rule 
Rule 
"main" 
"enable intranode_visibility_enabled" 

5.6.2 Ensure use of legacy Compute of VPC-native clusters

***Sentinel Policy Name:***

5.6.2 Ensure use of VPC-native clusters

***Category:***

* + Kubernetes

***Description of Policy:***

* + Create Alias IPs for the node network CIDR range in order to subsequently configure IP-based policies and firewalling for pods. A cluster that uses Alias IPs is called a 'VPC-native' cluster.

***Sentinel Policy Restriction:***

* + The policy checks whether "**ip\_allocation\_policy**" block contains values

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "ip\_allocation\_policy" block

***Test cases:***

**Pass cases**

1. In pass case, if "ip\_allocation\_policy" block contains values, then the pass case will always pass.

Screenshot of argument in mock file

" cluster_secondary_range name" : 
name" : 
"pods " 
services 

**Fail case:**

1. In pass case ,if "ip\_allocation\_policy" block does not contains values, then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-use-of-VPC-native-c1usters 
Installing test modules for test/kubernetes-ensure-use-of-VPC-native-c1usters/fai1. hcl 
Installing test modules for test/kubernetes-ensure-use-of-VPC-native-c1usters/pass . hcl 
kubernetes-ensure-use-of-VPC-native-c1usters . sentinel 
PASS 
test/kubernetes-ensure-use-of-VPC-native-c1usters/fai1. hcl 
PASS 
- verbose 
logs : 
Ensure use of VPC native clusters (using Alias IPs) 
trace: 
kubernetes-ensure-use-of-VPC-native-c1usters . sentinel : 19: 1 
Value: 
false 
kubernetes-ensure-use-of-VPC-native-c1usters . sentinel : 12:1 
Value: 
false 
PASS 
test/kubernetes-ensure-use-of-VPC-native-c1usters/pass . hcl 
logs : 
Ensure use of VPC native clusters (using Alias IPs) 
trace: 
kubernetes-ensure-use-of-VPC-native-c1usters . sentinel : 19: 1 
Value: 
true 
kubernetes-ensure-use-of-VPC-native-c1usters . sentinel : 12:1 
Value: 
true 
Rule 
Rule 
Rule 
Rule 
"main" 
"cluster_alias_ip_ranges_is_enabled" 
"main" 
"cluster_alias_ip_ranges_is_enabled" 

5.6.3 Ensure Master Authorized Networks is Enabled

***Sentinel Policy Name:***

5.6.3 Ensure Master Authorized Networks is Enabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Enable Master Authorized Networks to restrict access to the cluster's control plane (master endpoint) to only an allowlist (whitelist) of authorized IPs.
  + Master Authorized Networks blocks untrusted IP addresses. Google Cloud Platform IPs (such as traffic from Compute Engine VMs) can reach your master through HTTPS provided that they have the necessary Kubernetes credentials.

***Sentinel Policy Restriction:***

* + The policy checks whether "**master\_authorized\_networks\_config**" block contains values

***Terraform attributes:***

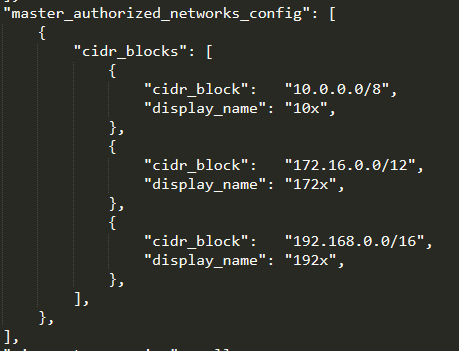
* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "master\_authorized\_networks\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "master\_authorized\_networks\_config" block contains values, then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case if "master\_authorized\_networks\_config" block does not contains values, then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
- run=kubernetes - ensure- master- authori zed- networks- is- set- to- enabled 
Installing test modules for test/kubernetes-ensure-master-authorized-networks-is-set-to-enabled/fail . hcl 
Installing test modules for test/kubernetes-ensure-master-authorized-networks-is-set-to-enabled/pass . hcl 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel 
PASS 
test/kubernetes - ensure -master - authorized - networks - is - set -to- enabled/fail . hcl 
PASS 
logs : 
Ensure master authorized networks is set to Enabled on Kubernetes Engine Clusters 
- verbose 
trace: 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 24 : 1 
Value: 
false 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 12 : 1 
Value: 
true 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 18 : 1 
Value: 
false 
Rule 
Rule 
Rule 
PASS 
test/kubernetes - ensure -master - authorized - networks - is - set -to- enabled/ pass . hcl 
logs : 
Ensure master authorized networks is set to Enabled on Kubernetes Engine Clusters 
trace: 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 24 : 1 
Value: 
true 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 12 : 1 
Value: 
true 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 18 : 1 
Value: 
true 
Rule 
Rule 
Rule 
"main" 
" deny _ 
"master _ authorized _ networks_config_is_configured" 
"main" 
" deny _ 
"master _ authorized _ networks_config_is_configured" 

5.6.4 Ensure clusters are created with Private Endpoint Enabled and Public Access Disabled

***Sentinel Policy Name:***

5.6.4 Ensure clusters are created with Private Endpoint Enabled and Public Access Disabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Disable access to the Kubernetes API from outside the node network if it is not required.
  + In a private cluster, the master node has two endpoints, a private and public endpoint. The private endpoint is the internal IP address of the master, behind an internal load balancer in the master's VPC network. Nodes communicate with the master using the private endpoint. The public endpoint enables the Kubernetes API to be accessed from outside the master's VPC network.
  + Although Kubernetes API requires an authorized token to perform sensitive actions, a vulnerability could potentially expose the Kubernetes publically with unrestricted access. Additionally, an attacker may be able to identify the current cluster and Kubernetes API version and determine whether it is vulnerable to an attack. Unless required, disabling public endpoint will help prevent such threats, and require the attacker to be on the master's VPC network to perform any attack on the Kubernetes API.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_private\_endpoint**" is set to true

***Terraform attributes:***

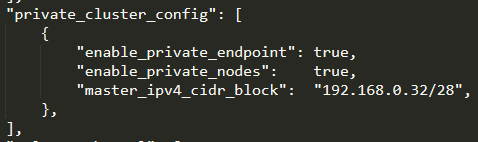
* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enable\_private\_endpoint" under "private\_cluster\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_private\_endpoint" is set to true then the pass case will always pass.

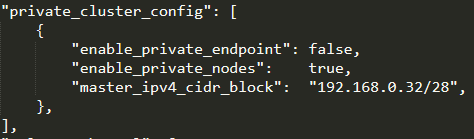
Screenshot of argument in mock file



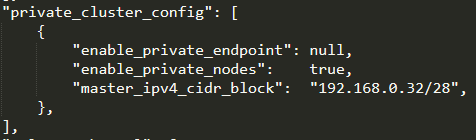
**Fail case:**

1. In pass case ,if "enable\_private\_endpoint" is not set to true then the pass case will always fail.

Screenshot of argument in mock file



**OR**



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-clusters-are-created-with-private-endpoint-enabled-public-access-disabled 
kubernetes - ensure - clusters - are - created -with - private - endpoint - enabled - public - access - disabled . sentinel 
PASS 
PASS 
PASS 
PASS 
test/kubernetes - ensure - clusters - are - created -with - private - endpoint - enabled - public - access - disabled/fail . hcl 
test/kubernetes - ensure - clusters - are - created -with - private - endpoint - enabled - public - access - disabled/fai12. hcl 
test/kubernetes - ensure - clusters - are - created -with - private - endpoint - enabled - public - access - disabled/ pass . hcl 

5.6.5 Ensure clusters are created with Private Nodes

***Sentinel Policy Name:***

5.6.5 Ensure clusters are created with Private Nodes

***Category:***

* + Kubernetes

***Description of Policy:***

* + Disable public IP addresses for cluster nodes, so that they only have private IP addresses. Private Nodes are nodes with no public IP addresses.
  + Disabling public IP addresses on cluster nodes restricts access to only internal networks, forcing attackers to obtain local network access before attempting to compromise the underlying Kubernetes hosts.
  + To enable Private Nodes, the cluster has to also be configured with a private master IP range and IP Aliasing enabled.

Private Nodes do not have outbound access to the public internet. If you want to provide outbound Internet access for your private nodes, you can use Cloud NAT or you can manage your own NAT gateway.

To access Google Cloud APIs and services from private nodes, Private Google Access needs to be set on Kubernetes Engine Cluster Subnets.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_private\_nodes**" is set to true

***Terraform attributes:***

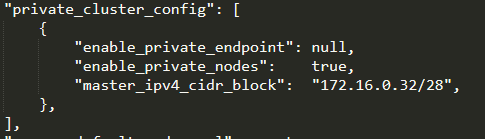
* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enable\_private\_nodes" under "private\_cluster\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_private\_nodes" is set to true then the pass case will always pass.

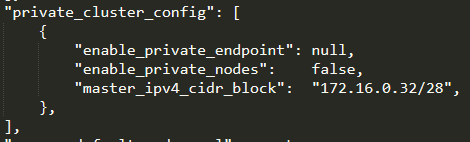
Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "enable\_private\_nodes" is not set to true then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-kubernetes-cluster-is-created-with-private-nodes 
kubernetes - ensure - kubernetes - cluster - is - created -with - private - nodes . sentinel 
PASS 
test/kubernetes - ensure - kubernetes - cluster - is - created -with - private - nodes/fail . hcl 
PASS 
test/kubernetes - ensure - kubernetes - cluster - is - created -with - private - nodes/ pass . hcl 
PASS 

Ensure Network Policy is Enabled and set as appropriate

***Sentinel Policy Name:***

5.6.7 Ensure Network Policy is Enabled and set as appropriate

***Category:***

* + Kubernetes

***Description of Policy:***

* + Use Network Policy to restrict pod to pod traffic within a cluster and segregate workloads.
  + By default, all pod to pod traffic within a cluster is allowed. Network Policy creates a pod-level firewall that can be used to restrict traffic between sources. Pod traffic is restricted by having a Network Policy that selects it (through the use of labels). Once there is any Network Policy in a namespace selecting a particular pod, that pod will reject any connections that are not allowed by any Network Policy. Other pods in the namespace that are not selected by any Network Policy will continue to accept all traffic.

Network Policies are managed via the Kubernetes Network Policy API and enforced by a network plugin, simply creating the resource without a compatible network plugin to implement it will have no effect. GKE supports Network Policy enforcement through the use of Calico.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enabled**" is set to true

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enabled" inside the "network\_policy" block

***Test cases:***

**Pass cases**

1. In pass case, if "enabled" is set to true then the pass case will always pass.

Screenshot of argument in mock file

"network_policy" : 
" enabled" : 
" provider" : 
true, 
null, 

**Fail case:**

1. In pass case ,if "enabled" is not set to true then the pass case will always fail.

Screenshot of argument in mock file

"network_policy" : 
" enabled" : 
" provider" : 
false, 
null, 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C: /mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ 
kubernetes - ensure - network- policy -is - enabled . sentinel 
PASS 
test/kubernetes - ensure - network- policy - is - enabled/fail . hcl 
PASS 
test/kubernetes - ensure - network- policy - is - enabled/ pass . hcl 
PASS 
sentinel test 
- run=kubernetes- ensure- network- pol i cy- is- enabled 

5.7.1 Ensure Stackdriver Kubernetes Logging and Monitoring is Enabled

***Sentinel Policy Name:***

* + 5.7.1 Ensure Stackdriver Kubernetes Logging and Monitoring is Enabled

***Category:***

* + Kubernetes : Managed services: Logging

***Description of Policy:***

* + Enabling stackdriver kubernetes logging and monitoring will send logs and metrics to a remote aggregator to mitigate

the risk of local tampering in the event of a breach.

***Sentinel Policy Restriction:***

* + Sentinel policy will check in terraform code whether below allowed types are present in terraform code
  + **Logging\_service** contain any of ["**logging.googleapis.com/kubernetes","logging.googleapis.com**"] And
  + **Monitoring\_services** contain any of [**"Monitoring.googleapis.com/Kubernetes","monitoring.googleapis.com**"]

***Terraform attributes:***

Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>

Terraform attribute: "logging\_service" and "monitoring\_service"

***Test cases:***

**Pass cases**

1. In pass case, if "logging\_service" is set to "logging.googleapis.com/kubernetes" or "logging.googleapis.com" AND

"monitoring\_service" is set to "monitoring.googleapis.com/kubernetes" or "monitoring.googleapis.com" for both resources, then the pass case will always pass.

Screenshot of argument in mock file



and



**Fail case:**

1. In pass case, if "logging\_service" is NOT set to "logging.googleapis.com/kubernetes" or "logging.googleapis.com" OR

"monitoring\_service" is NOT set to "monitoring.googleapis.com/kubernetes" or "monitoring.googleapis.com" for both resources, then the pass case will always fail.

Screenshot of argument in mock file

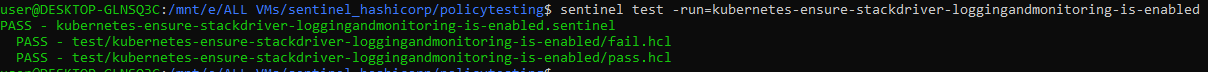


And



**Testcases Output:**

Screenshot of test command successful



5.8.2 Ensure authentication using Client Certificates is Disabled

***Sentinel Policy Name:***

5.8.2 Ensure authentication using Client Certificates is Disabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Disable Client Certificates, which require certificate rotation, for authentication. Instead, use another authentication method like OpenID Connect.
  + GKE manages authentication via gcloud for you using the OpenID Connect token method, setting up the Kubernetes configuration, getting an access token, and keeping it up to date. This means Basic Authentication using static passwords and Client Certificate authentication, which both require additional management overhead of key management and rotation, are not necessary and should be disabled.

***Sentinel Policy Restriction:***

* + The policy checks whether "issue\_client\_certificate" is set to false

***Terraform attributes:***

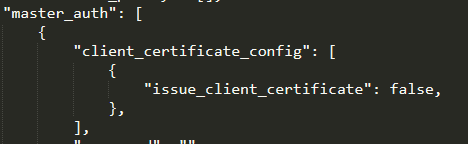
* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "issue\_client\_certificate" under "client\_certificate\_config" block under "master\_auth" block

***Test cases:***

**Pass cases**

1. In pass case, If "issue\_client\_certificate" is set to false then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case, If "issue\_client\_certificate" is not set to false then the pass case will always fail.

Screenshot of argument in mock file

"master auth" : 
issue client certificate": 
true, 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-authentication-using-client-certificate-disabled 
Installing test modules for test/kubernetes-ensure-authentication-using-client-certificate-disabled/fail.hcl 
Installing test modules for test/kubernetes-ensure-authentication-using-client-certificate-disabled/pass . hcl 
kubernetes - ensure - authentication - using- client - certificate - disabled . sentinel 
PASS 
test/kubernetes - ensure - authentication - using- client - certificate - disabled/fail . hcl 
PASS 
logs : 
Ensure Authentication using Client Certificate is Disabled. Instead use another authentication method like OpenlD Connect 
- verbose 
trace: 
kubernetes 
Value: 
false 
kubernetes 
Value: 
false 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
-ensure 
-ensure 
-ensure 
-ensure 
-authentication 
-authentication 
-authentication 
-authentication 
-using- 
-using- 
-using- 
-using- 
client-certificate-disabled . sentinel : 36 
client-certificate-disabled . sentinel : 26 
client-certificate-disabled . sentinel : 18 
client-certificate-disabled . sentinel : 12 
Rule 
Rule 
Rule 
Rule 
"main" 
"client certificate is disabled" 
" deny _ undefined _ client _ certificate _ config" 
" deny _ " 
PASS 
logs : 
test/kubernetes - ensure - authentication - using- client - certificate - disabled/pass . hcl 
Ensure Authentication using Client Certificate is Disabled. Instead use another authentication method like OpenlD Connect 
trace: 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
kubernetes 
Value: 
true 
-ensure 
-ensure 
-ensure 
-ensure 
-authentication 
-authentication 
-authentication 
-authentication 
-using- 
-using- 
-using- 
-using- 
client-certificate-disabled . sentinel : 36 
client-certificate-disabled . sentinel : 26 
client-certificate-disabled . sentinel : 18 
client-certificate-disabled . sentinel : 12 
Rule 
Rule 
Rule 
Rule 
"main" 
"client certificate is disabled" 
" deny _ undefined _ client _ certificate _ config" 
" deny _ " 

5.8.4 Ensure Legacy Authorization (ABAC) is Disabled

***Sentinel Policy Name:***

5.8.4 Ensure Legacy Authorization (ABAC) is Disabled

***Category:***

* + Kubernetes

***Description of Policy:***

* + Legacy Authorization, also known as Attribute-Based Access Control (ABAC) has been superseded by Role-Based Access Control (RBAC) and is not under active development. RBAC is the recommended way to manage permissions in Kubernetes.
  + As RBAC provides significant security advantages over ABAC, it is recommended option for access control. Where possible, legacy authorization must be disabled for GKE clusters.

***Sentinel Policy Restriction:***

* + The policy checks whether "enable\_legacy\_abac" is set to false

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enable\_legacy\_abac"

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_legacy\_abac" is set to false then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case, if "enable\_legacy\_abac" is not set to false then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
- run=kubernetes - ensure- master- authori zed- networks- is- set- to- enabled 
Installing test modules for test/kubernetes-ensure-master-authorized-networks-is-set-to-enabled/fail . hcl 
Installing test modules for test/kubernetes-ensure-master-authorized-networks-is-set-to-enabled/pass . hcl 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel 
PASS 
test/kubernetes - ensure -master - authorized - networks - is - set -to- enabled/fail . hcl 
PASS 
logs : 
Ensure master authorized networks is set to Enabled on Kubernetes Engine Clusters 
- verbose 
trace: 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 24 : 1 
Value: 
false 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 12 : 1 
Value: 
true 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 18 : 1 
Value: 
false 
Rule 
Rule 
Rule 
PASS 
test/kubernetes - ensure -master - authorized - networks - is - set -to- enabled/ pass . hcl 
logs : 
Ensure master authorized networks is set to Enabled on Kubernetes Engine Clusters 
trace: 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 24 : 1 
Value: 
true 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 12 : 1 
Value: 
true 
kubernetes - ensure -master - authorized - networks - is - set -to- enabled . sentinel : 18 : 1 
Value: 
true 
Rule 
Rule 
Rule 
"main" 
" deny _ 
"master _ authorized _ networks_config_is_configured" 
"main" 
" deny _ 
"master _ authorized _ networks_config_is_configured" 

5.9.1 Enable Customer-Managed Encryption Keys (CMEK) for GKE Persistent Disks (PD)

***Sentinel Policy Name:***

5.9.1 Enable Customer-Managed Encryption Keys (CMEK) for GKE Persistent Disks (PD)

***Category:***

* + Kubernetes

***Description of Policy:***

* + Use Customer-Managed Encryption Keys (CMEK) to encrypt node boot and dynamically-provisioned attached Google Compute Engine Persistent Disks (PDs) using keys managed within Cloud Key Management Service (Cloud KMS).
  + GCE persistent disks are encrypted at rest by default using envelope encryption with keys managed by Google. For additional protection, users can manage the Key Encryption Keys using Cloud KMS.

***Sentinel Policy Restriction:***

* + The policy checks whether "**disk\_type**" is set to "**pd-ssd**" or "**pd-standard**" and "**boot\_disk\_kms\_key**" contains a value.

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_node_pool>
    - Terraform attribute: "disk\_type" and "boot\_disk\_kms\_key" inside the "node\_config" block

***Test cases:***

**Pass cases**

1. In pass case, if "disk\_type" is set to "pd-ssd" or "pd-standard" and "boot\_disk\_kms\_key" contains a value for both resources, then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case if "disk\_type" is not set to "pd-ssd" or "pd-standard" or "boot\_disk\_kms\_key" does not contain a value for both resources, then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

Wls/sentinel_hashicorp/policytesting$ sentinel test -run=kubernetes-ensure-customer-managed-encryption-keys-for-GKE-Persistent-Disks 
kubernetes - ensure - customer -managed - encryption - keys -for -GKE - Persistent - Disks . sentinel 
PASS 
test/kubernetes - ensure - customer -managed - encryption - keys -for -GKE - Persistent - Disks/fail . hcl 
PASS 
test/kubernetes - ensure - customer -managed - encryption - keys -for -GKE - Persistent - Disks/ pass . hcl 
PASS 

5.10.5 Ensure use of Binary Authorization

***Sentinel Policy Name:***

5.10.5 Ensure use of Binary Authorization

***Category:***

* + Kubernetes

***Description of Policy:***

* + Binary Authorization helps to protect supply-chain security by only allowing images with verifiable cryptographically signed metadata into the cluster.
  + Binary Authorization provides software supply-chain security for images that you deploy to GKE from Google Container Registry (GCR) or another container image registry.

* + Binary Authorization requires images to be signed by trusted authorities during the development process. These signatures are then validated at deployment time. By enforcing validation, you can gain tighter control over your container environment by ensuring only verified images are integrated into the build-and-release process.

***Sentinel Policy Restriction:***

* + The policy checks whether "**evaluation\_mode**" is set to "**PROJECT\_SINGLETON\_POLICY\_ENFORCE**"
    - This is because the "**enabled**" attribute for binary authorization is deprecated. **PROJECT\_SINGLETON\_POLICY\_ENFORCE** is functionally equivalent to the deprecated enable\_binary\_authorization parameter being set to true as per provider registry docs.

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "evaluation\_mode" inside the "binary\_authorization" block

***Test cases:***

**Pass cases**

1. In pass case, if "evaluation\_mode" is set to "PROJECT\_SINGLETON\_POLICY\_ENFORCE" then the pass case will always pass.

Screenshot of argument in mock file

"binary_authorization" . 
" enabled" : 
null, 
"evaluation mode": " 
PROJECT SINGLETON POLICY ENFORCE" 

**Fail case:**

1. In pass case ,if "evaluation\_mode" is not set to "PROJECT\_SINGLETON\_POLICY\_ENFORCE" then the pass case will always fail.

Screenshot of argument in mock file

"binary_authorization" 
" enabled" : 
null, 
"evaluation mode : " 
" ALWAYS ALLOW" , 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e/ALL 
Wls/sentinel_hashicorp/policytesting$ 
kubernetes - ensure - use -of - binary - authorization . sentinel 
PASS 
test/kubernetes - ensure - use -of - binary - authorization/fail . hcl 
PASS 
test/kubernetes - ensure - use -of - binary - authorization/ pass . hcl 
PASS 
sentinel test 
- run=kubernetes-ensure- use-of- binary- authorization 

5.10.2 Ensure that Alpha clusters are not used for production workloads

***Sentinel Policy Name:***

5.10.2 Ensure that Alpha clusters are not used for production workloads

***Category:***

* + Kubernetes

***Description of Policy:***

* + Alpha clusters are not covered by an SLA and are not production-ready.
  + Alpha clusters are designed for early adopters to experiment with workloads that take advantage of new features before those features are production-ready. They have all Kubernetes API features enabled, but are not covered by the GKE SLA, do not receive security updates, have node auto-upgrade and node auto-repair disabled, and cannot be upgraded. They are also automatically deleted after 30 days.

***Sentinel Policy Restriction:***

* + The policy checks whether "**enable\_kubernetes\_alpha**" is not set to **true**

***Terraform attributes:***

* + - Provider Ref: <https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/container_cluster>
    - Terraform attribute: "enable\_kubernetes\_alpha"

***Test cases:***

**Pass cases**

1. In pass case, if "enable\_kubernetes\_alpha" is not set to true then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "enable\_kubernetes\_alpha" is set to true then the pass case will always fail.

Screenshot of argument in mock file



**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e/ALL 
Wls/sentinel_hashicorp/policytesting$ sentinel test -run=S.1ø.2-kubernetes-ensure-a1pha-c1usters-are-not-used-for-production 
S . 18.2-kubernetes-ensure-alpha-clusters-are-not-used-for -production . sentinel 
PASS 
test'S . 18.2-kubernetes-ensure-alpha-clusters-are-not-used-for -production/fail . hcl 
PASS 
test'S . 18.2-kubernetes-ensure-alpha-clusters-are-not-used-for -production/pass . hcl 
PASS 

5.10.6 Enable Cloud Security Command Center (Cloud SCC)

***Sentinel Policy Name:***

* + 5.10.6 Enable Cloud Security Command Center (Cloud SCC)

***Category:***

* + Kubernetes

***Description of Policy:***

* + Enable Cloud Security Command Center (Cloud SCC) to provide a centralized view of security for your GKE clusters.
  + Cloud Security Command Center (Cloud SCC) is the canonical security and data risk database for GCP. Cloud SCC enables you to understand your security and data attack surface by providing asset inventory, discovery, search, and management.

***Sentinel Policy Restriction:***

* + The policy checks whether "**service**" is set to "**securitycenter.googleapis.com**"

***Terraform attributes:***

* + - Provider Ref:

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/google_project_service>

* + - Terraform attribute: "service"

***Test cases:***

**Pass cases**

1. In pass case, if "service" is set to "securitycenter.googleapis.com" then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case ,if "service" is not set to "securitycenter.googleapis.com" then the pass case will always fail.

**No Screenshot of argument in mock file**

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP -GLNSQ3C : /mnt/e/ALL 
Wls/sentinel_hashicorp/policytesting$ sentinel test 
S . 18.6-kubernetes-enable-cloud-security -command-center . sentinel 
PASS 
test'S . 18.6-kubernetes-enable-cloud-security -command-center/fail . hcl 
PASS 
test'S . 18.6-kubernetes-enable-cloud-security -command-center/pass . hcl 
PASS 
- run-S . 18.6- kubernetes-enable- cloud- security- command- center 