# 

**CSWG- Policy As Code Documentation**

SQL Sentinel Policies

6.4 Ensure That the Cloud SQL Database Instance Requires All Incoming Connections to Use SSL

***Sentinel Policy Name:***

* + 6.4 Ensure That the Cloud SQL Database Instance Requires All Incoming Connections to Use SSL

***Category:***

* + Cloud SQL

***Description of Policy:***

* + SQL database connections if successfully trapped (MITM); can reveal sensitive data like credentials, database queries, query outputs etc.
  + For security, it is recommended to always use SSL encryption when connecting to our instance.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. require\_ssl parameter is set to true.

" ip_configuration" : 
" allocated_ip_range" : 
"authorized networks" : 
" ipv4_enab1ed" : 
'private network" : 
' require_ssl " : 
null, 
true, 
null, 
true, 

**Fail Cases:**

1. require\_ssl parameter is set to false.

" ip_configuration" : 
" allocated_ip_range" : 
"authorized networks" : 
" ipv4_enab1ed" : 
'private network" : 
' require_ssl " : 
null, 
true, 
null, 
false, 

2. ip\_configuration block is null.

" ip_configuration" : 

**Testcases Output:**

PS C: \Users\1939847\Documents\f01der3\c10ud_sq1> sentinel test 
PASS - ensure-sql-instance-require-all-incoming-connections-to-use-ssl. sentinel 
PASS - test\ensure-sql-instance-require-all-incoming-connections-to-use-ssl\fail.hcl 
PASS - test\ensure-sql-instance-require-all-incoming-connections-to-use-ssl\faill.hcl 
PASS - test\ensure-sql-instance-require-all-incoming-connections-to-use-ssl\pass.hcl 

6.5 Ensure That Cloud SQL Database Instances Do Not Implicitly Whitelist All Public IP Addresses

***Sentinel Policy Name:***

* 6.5 Ensure That Cloud SQL Database Instances Do Not Implicitly Whitelist All Public IP Addresses

***Category:***

* Cloud SQL

***Description of Policy:***

* Database Server should accept connections only from trusted Network(s)/IP(s) and restrict access from public IP addresses.
* To minimize attack surface on a Database server instance, only trusted/known and required IP(s) should be white-listed to connect to it.

***Sentinel Policy Restriction:***

* For 'google\_sql\_database\_instance', an authorized network should not have IPs/networks configured to 0.0.0.0/0 which will allow access to the instance from anywhere in the world.

***Terraform attributes:***

* + Provider Ref: [google\_sql\_database\_instance | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance)

***Test cases:***

**Pass cases**

1. For resource 'google\_sql\_database\_instance' the attribute 'authorized\_networks.value' is set to something other than '0.0.0.0/0'

"ip_configuration": [ 
"al 
null, 
"expiration time": " 
"name": 
"value" : 
"192.168.1ø.131/g2", 

**Fail case:**

1. For resource 'google\_sql\_database\_instance' the attribute 'authorized\_networks.value' is set '0.0.0.0/0'

"ip_configuration": [ 
null, 
"authorized networks": 
"expiration_time"; " 
"name" 
"value": 

**Testcases Output:**

PASS - Ensure That (101K] Database Instances 1k) Implicitly. sentinel 
PASS - test\Ensure That Clotx] Database Instances Do Implicitly\fail. hcl 
logs : 
Your infrastructure is non- carwliant. Please ensure that •value • IS IOT e.e.e.e/e. (referring to the value arguanent with in the 
d_neti.nrks• attribute of the resource This effort Will improve security. 
Implicitly. sentinel : 28 
- Rule "main 
Implicitly. sentinel : 13 
• auttX)ri ze 
trace: 
Ensure That Cloud SQL Database Instances Do r•klt Implicitly. sentinel :28:1 
Value: 
false 
Ensure That Cloud SQL Database Instances Do Implicitly. sentinel 
resses" 
Value : 
false 
- Rule 
- Rule 
PASS - That SW Database Instances Do Implicitly\pass.hcl 
trace : 
Ensure That 
Value : 
Ensure That 
Value : 
Cloud 
Cloud 
Database 
Database 
Instances 
Instances 
- Rule 
"main" 
" SQL _ c tel ist_al l_Publ i c _IP_Add 
all Public IP_Add 

6.6 Ensure That Cloud SQL Database Instances Do Not Have Public IPs

***Sentinel Policy Name:***

* 6.6 Ensure That Cloud SQL Database Instances Do Not Have Public IPs

***Category:***

* Cloud SQL

***Description of Policy:***

* It is recommended to configure SQL database instance to use private IPs instead of public IPs.
* To lower the organization's attack surface, Cloud SQL databases should not have public IPs. Private IPs provide improved network security and lower latency for your application.

***Sentinel Policy Restriction:***

* For 'google\_sql\_database\_instances' the default configuration assigns the asset a public IP. This policy will enforce a configuration where public IP can not be defined

***Terraform attributes:***

* + Provider Ref: [google\_sql\_database\_instance | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance)

***Test cases:***

**Pass cases**

1. For resource 'google\_sql\_database\_instance' the attribute 'ipv4\_enabled = false'

"ip_configuration": [ 
"al : 
"authorized networks . 
"require_ssl": 
null, 
false, 
null, 

**Fail case:**

1. For resource 'google\_sql\_database\_instance' the attribute 'ipv4\_enabled' is something other than 'false'

"ip_configuration": [ 
"authorized networks . 
"require SSI": 
null, 
true, 
null, 

**Testcases Output:**

PASS - • _ 
- . hcl 
trace: 
QL sentinel : 26: 1 - Rule "main 
Value : 
false 
QL have a Public IP. sentinel : 15:1 - Rule 
Value : 
false 
PASS - . hcl 
- Rule "maln 
trace: 
QL have a Public IP. sentinel : 26: 1 
Value: 
QC _ have a Public_IP. sentinel : 15:1 - 
Value: 
Rule 
"SQL _ have_a Public IP" 
have_a Public IP" 

6.7 Ensure That Cloud SQL Database Instances Are Configured with Automated Backups.

***Sentinel Policy Name:***

* + 6.7 Ensure That Cloud SQL Database Instances Are Configured with Automated Backups.

***Category:***

* + Cloud SQL

***Description of Policy:***

* + Backups provide a way to restore a Cloud SQL instance to recover lost data or recover from a problem with that instance.
  + Automated backups need to be set for any instance that contains data that should be protected from loss or damage.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, in backup\_configuration block, “enabled” parameter is set to “true”

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. In backup\_configuration block, “enabled” parameter is set to “true”.

"backup _ configuration" : 
" enabled " : 
" location" : 
'point_in_time recovery 
"start time" : 
enabled" : 
null, 
true, 
null, 
null, 

**Fail Cases:**

1. In backup\_configuration block, “enabled” parameter is set to “false”.

"backup _ configuration" : 
" enabled " : 
" location" : 
'point_in_time recovery 
"start time" : 
enabled" : 
null, 
false, 
null, 
null, 

2. When the backup\_configuration block is empty.

" backup _ configuration " : 

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_sq1\c10ud_sq1_1> sentinel test 
PASS - ensure-automatic-backup-is-enabled. sentinel 
PASS - test\ensure-automatic-backup-is-enabled\fail.hcl 
PASS - hcl 
PASS - test\ensure-automatic-backup-is-enabled\pass.hcl 

6.1.1 Ensure That a MySQL Database Instance Does Not Allow Anyone To Connect With Administrative Privileges

***Sentinel Policy Name:***

* CIS 6.1.1 Ensure That a MySQL Database Instance Does Not Allow Anyone To Connect With Administrative Privileges

***Category :***

* Cloud SQL

***Description of Policy:***

* It is recommended to set a password for the administrative user (rootby default) to prevent unauthorized access to the SQL database instances
* This recommendation is applicable only for MySQL Instances. PostgreSQL does not offer any setting for No Password from the cloud console.

***Sentinel Policy Restriction:***

* At the time of MySQL Instance creation, not providing an administrative password allows anyone to connect to the SQL database instance with administrative privileges. The root password should be set to ensure only authorized users have these privileges.
* This policy enforces Password Validation on all MySQL database Admins/ Superusers/ and regular users

***Terraform attributes:***

* + Provider Ref: [google\_sql\_database\_instance | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance)

***Test cases:***

**Pass cases**

1. For resource 'google\_sql\_database\_instance' the attribute 'enable\_password\_policy' is 'true'
2. For resource 'google\_sql\_database\_instance' the attribute 'disallow\_username\_substring' is 'true'
3. For resource 'google\_sql\_database\_instance' the attribute 'min\_length' is in the 'allsupported\_min\_length' (per the Sentinel Policy)
4. For resource 'google\_sql\_database\_instance' the attribute 'reuse\_interval' is in the 'allsupported\_reuse\_interval' (per the Sentinel Policy)

"password_val idation_policy" : 
" complexity" : 
"disallow_username_substring": true, 
cy" : 
"min _ length" : 
nterval" : 
"reuse interval": 
null, 
true , 
null, 

**Fail case:**

1. For resource 'google\_sql\_database\_instance' the attribute 'enable\_password\_policy' is 'false'
2. For resource 'google\_sql\_database\_instance' the attribute 'disallow\_username\_substring' is 'false'
3. For resource 'google\_sql\_database\_instance' the attribute 'min\_length' is something outside of the 'allsupported\_min\_length' (per the Sentinel Policy)
4. For resource 'google\_sql\_database\_instance' the attribute 'reuse\_interval' is something outside of the 'allsupported\_reuse\_interval' (per the Sentinel Policy)

" icy" : 
"complexity" : 
"min_length": 
"pas " : 
"reuse interval": 
true , 
true, 
true , 
null, 

**Testcases Output:**

PASS - enforced. sentinel 
PASS - 
logs : 
In order to be canpliant with this ensure that the values in the •passmrd_va1idationÄicy• 
he values are: canplexity: null disallch.a usernane_substring: true true min_length: 
terval: null reuse interval: 3 
trace : 
enforced. sentinel : 38: 1 
- Rule "maln 
Value : 
false 
enforced. sentinel : 15:1 
QL Instances" 
Value : 
false 
- Rule "main 
are set correctly. T 
- Rule 
PASS - hcl 
trace : 
mySQL_Aånin_superuser_passmrd_enforced. sentinel : 38 : 1 
Value : 
true 
sentinel : IS: 1 - Rule 
QL_Instances" 
Value : 
for 

6.1.2 Ensure ‘Skip\_show\_database’ Database Flag for Cloud SQL MySQL Instance Is Set to ‘On’.

***Sentinel Policy Name:***

* + 6.1.2 Ensure ‘Skip\_show\_database’ Database Flag for Cloud SQL MySQL Instance Is Set to ‘On’.

***Category:***

* + Cloud SQL(MySQL)

***Description of Policy:***

* + 'skip\_show\_database' database flag prevents people from using the SHOW DATABASES statement if they do not have the SHOW DATABASES privilege.
  + This can improve security if we have concerns about users being able to see databases belonging to other users.
  + When the skip\_show\_database' database flag is set to 'on', the SHOW DATABASES statement is permitted only to users with the above privilege.
  + When the skip\_show\_database' database flag is set to 'off', the SHOW DATABASES statement is permitted to all users, but displays the names of only those databases for which the user has the SHOW DATABASES or other privilege.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy Restriction:***

* + This sentinel policy ensures that “skip\_show\_database” flag is set to “on”,

***Pass and fail cases of the above sentinel policy:***

**Pass Case:**

1. ‘skip\_show\_database’ database flag is set to ‘on’.

" database _ flags " : 
"name" : 
"skip _ show database" , 
"value": "On 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘skip\_show\_database’ database flag is set to ‘off’.

"database _ flags " : 
" name" : 
"skip _ show database" , 
"value": "Off 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: skip database show database flag is on> sentinel 
test 
PASS - ensure-skip-database-show-database-flag-is-on. sentinel 
PASS 
PASS 
PASS 
PASS 
- test\ensure-skip-database-show-database-flag-is-on\fail.hcl 
- test\ensure-skip-database-show-database-flag-is-on\faill.hcl 
- test\ensure-skip-database-show-database-f1ag-is-on\fai12.hc1 
- test\ensure-skip-database-show-database-flag-is-on\pass.hcl 

6.1.3 Ensure That the ‘Local\_infile’ Database Flag for a Cloud SQL MySQL Instance Is Set to ‘Off’.

***Sentinel Policy Name:***

* + 6.1.3 Ensure That the ‘Local\_infile’ Database Flag for a Cloud SQL MySQL Instance Is Set to ‘Off’.

***Category:***

* + Cloud SQL(MySQL)

***Description of Policy:***

* + 1. The local\_infile flag controls the server-side LOCAL capability for LOAD DATA statements.
  + 2. Depending on the 'local\_infile' database flag setting, the server refuses or permits local data loading by clients that have LOCAL enabled on the client side.
  + 3. Due to security issues associated with the local\_infile flag, it is recommended to disable it.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

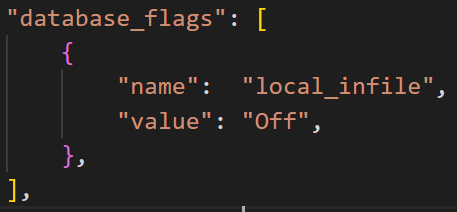
***Sentinel Policy restriction:***

* + In this sentinel policy, “local\_infile” database flag is set to “off”.

***Pass and fail cases of the above sentinel policy:***

**Pass Case:**

1. ‘local\_infile’ database flag is set to ‘off’.



**Fail Cases:**

1. When the database flag is null.



2. When the ‘local\_infile’ database flag is set to ‘on’.

" database _ flags " : 
"name" : 
"local infile", 
"value": 'On 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: local infile database flag is set to off> sentine 
1 test 
PASS - ensure-local-infile-database-flag-is-off. sentinel 
- test\ensure-local-infile-database-flag-is-off\fail.hcl 
PASS 
- test\ensure-local-infile-database-flag-is-off\faill.hcl 
PASS 
- test\ensure-10ca1-infi1e-database-f1ag-is-off\fai12.hc1 
PASS 
- test\ensure-local-infile-database-flag-is-off\pass.hcl 
PASS 

6.2.1 Ensure ‘Log\_error\_verbosity’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘DEFAULT’ or Stricter

***Sentinel Policy Name:***

* + 6.2.1 Ensure ‘Log\_error\_verbosity’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘DEFAULT’ or Stricter.

***Category:***

* + Cloud SQL(PostgreSQL)

***Description of Policy:***

* + The 'log\_error\_verbosity' flag controls the verbosity/details of messages logged.
  + Valid values are 'terse' or 'default' or 'verbose'
  + TERSE excludes the logging of DETAIL, HINT, QUERY, and CONTEXT error information.
  + VERBOSE output includes the SQLSTATE error code, source code file name, function name and line number that generated the error.
  + So, we have to set ‘log\_error\_verbosity’ database flag to ‘default’ or ‘verbose’.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “log\_error\_verbosity” database flag is set to “default” or stricter.

***Pass and fail cases of the above sentinel policy:***

**Pass Cases:**

1. ‘log\_error\_verbosity’ database flag is set to ‘default’.

" database _ flags " : 
'name": 
"value": "default", 

2. ‘log\_error\_verbosity’ database flag is set to ‘verbose’.

"database _ flags " : 
" name" : 
"value": "verbose", 

**Fail Cases:**

1.’log\_error\_verbosity’ database flag is set to ‘terse’.

" database _ flags " : 
'name": 
"value": "terse" , 

2. When the database flag is null.

" database _ flags " : 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PASS - ensure-log-error-verbosity-database-flag-is-default-or-stricter.sentinel 
PASS 
PASS 
PASS 
PASS 
PASS 
- test\ensure-log 
- test\ensure-log 
- test\ensure-log 
- test\ensure-log 
- test\ensure-log 
-error 
-error 
-error 
-error 
-error 
-verbosity-database-flag-is-default -or- stricter\fail. hcl 
-verbosity-database-flag-is -default -or- stricter\faill. hcl 
-verbosity-database-flag-is -default -or- stricter\fai12. hcl 
-verbosity-database-flag-is-default -or- stricter\pass. hcl 
-verbosity-database-flag-is-default -or- stricter\passl. hcl 

6.2.2 Ensure That the ‘Log\_connections’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘On’.

***Sentinel Policy Name:***

* + 6.2.2 Ensure That the ‘Log\_connections’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘On’.

***Category:***

* + Cloud SQL(PostgreSQL)

***Description of Policy:***

* + PostgreSQL does not log attempted connections by default.
  + Enabling the log\_connections setting will create log entries for each attempted connection as well as successful completion of client authentication which can be useful in troubleshooting issues and to determine any unusual connection attempts to the server

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “log\_connections” database flag is set to “on”.

***Pass and fail cases of the above sentinel policy:***

**Pass Case:**

1. ‘log\_connections’ database flag is set to ‘on’.

" database _ flags " : 
"name" : 
" log_connections " , 
"value": 'on 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘log\_connections’ database flag is set to ‘off’.

"database _ flags " : 
" name" : 
" log _ connections " , 
"value": "Off 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_postgresq1\postgresq1> sentinel test 
PASS - ensure-postgreSQL-10g-connections-database-f1ag-is-on. sentinel 
PASS 
PASS 
PASS 
PASS 
- test\ensure-postgreSQL-10g-connections-database-f1ag-is-on\fai1.hc1 
- test\ensure-postgreSQL-10g-connections-database-f1ag-is-on\fai11.hc1 
- test\ensure-postgreSQL-10g-connections-database-f1ag-is-on\fai12.hc1 
- test\ensure-postgreSQL-10g-connections-database-f1ag-is-on\pass.hc1 

6.2.3 Ensure That the ‘Log\_disconnections’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘On’

***Sentinel Policy Name:***

* + 6.2.3 Ensure That the ‘Log\_disconnections’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘On’.

***Category:***

* + Cloud SQL(PostgreSQL)

***Description of Policy:***

* + PostgreSQL does not log session details such as duration and session end by default.
  + Enabling the 'log\_disconnections' setting will create log entries at the end of each session which can be useful in troubleshooting issues and determine any unusual activity across a time period.
  + The 'log\_disconnections' and 'log\_connections' work hand in hand and generally, the pair would be enabled/disabled together.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, ”log\_disconnections” database flag is set to “on”.

***Pass and fail cases of the above sentinel policy:***

**Pass Case:**

1. ‘log\_disconnections’ database flag is set to ‘on’.

"database _ flags " : 
"name" : 
"log_disconnections" , 
"value": 'on 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘log\_disconnections’ database flag is set to ‘off’.

" database _ flags " : 
"name" : 
"log_disconnections" , 
"value": 'Off 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_postgresq1\postgresq1_1> sentinel test 
PASS - ensure-postgreSQL-10g-disconnections-database-f1ag-is-on. sentinel 
PASS 
PASS 
PASS 
PASS 
- test\ensure-postgreSQL-10g-disconnections-database-f1ag-is-on\fai1.hc1 
- test\ensure-postgreSQL-10g-disconnections-database-f1ag-is-on\fai11.hc1 
- test\ensure-postgreSQL-10g-disconnections-database-f1ag-is-on\fai12.hc1 
- test\ensure-postgreSQL-10g-disconnections-database-f1ag-is-on\pass.hc1 

6.2.4 Ensure ‘Log\_statement’ Database Flag for Cloud SQL PostgreSQL Instance Is Set Appropriately

***Sentinel Policy Name:***

* CIS 6.2.4 Ensure ‘Log\_statement’ Database Flag for Cloud SQL PostgreSQL Instance Is Set Appropriately

***Category:***

* Cloud SQL

***Description of Policy:***

* For PostgreSQL database instances, the 'log\_statement' database flag ensures that audit logs are collected in an effective manner
* The database flag 'log\_statement' controls which SQL statements are logged.
* As per the CIS benchmarks, this policy impacts only PostgreSQL database instances (not MySQL or SQL server)

***Sentinel Policy Restriction:***

* For 'google\_sql\_database\_instance' resources with database\_version containing "POSTGRES", this policy ensures that 'log\_statement' database flag is active within each PostgreSQL database instance. This policy also checks to ensure that the 'value' is 'ddl'
* The value 'ddl' ensures that data definition statements such as 'CREATE', 'ALTER', DROP' are collected and logged

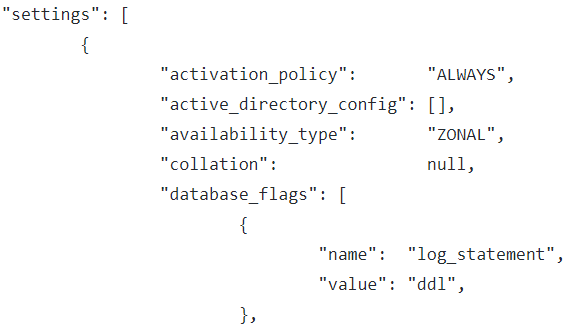
***Terraform attributes:***

* + Provider Ref: [google\_sql\_database\_instance | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance)

***Test cases:***

**Pass cases**

1. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.name' attribute contains "log\_statement"
2. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.value' attribute contains "ddl"



**Fail case:**

1. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.name' attribute contains something other than "log\_statement"
2. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.value' attribute contains something other than "ddl"

"settings": 
"acti vation_policy" : 
"ALWAYS", 
"acti ve_di rectory_config" • 
"availability_type" : 
"ZONAL", 
"collation": 
null, 
"database_flags": 
"name": 
"value": "dl 

**Testcases Output:**

PASS - sentinel 
PASS - hcl 
logs: 
Please enable ' log_statanent• database flag is enabled and the value is set it 
sentinel - Rule 
sentinel : 14:1 - Rule 
trace : 
sentinel : 29 : 1 
- Rule 
Value : 
false 
sentinel : 14 : 1 
- Rule 
Value : 
false 
PASS - hcl 
trace : 
Value : 
true 
Value : 
"main" 
"DB_f1ags" 
"main" 
"DB_f1ags" 

6.2.5 Ensure ‘Log\_hostname’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to 'on'.

***Sentinel Policy Name:***

* + 6.2.5 Ensure ‘Log\_hostname’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to 'on'.

***Category:***

* + Cloud SQL(PostgreSQL)

***Description of Policy:***

* + Logging hostnames allows for the association of hostname to IP address at the time of connection.
  + This information can aid with incident response efforts particularly in an environment that utilized dynamic IP addresses.
  + Logging hostnames may incur overhead on server performance as for each statement logged, DNS resolution will be required to convert IP address to hostname.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “log\_hostname” database flag is set to “on”.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. ‘log\_hostname’ database flag is set to ‘on’.

" database _ flags " : 
'name": 
" log_hostname " , 
"value": 'on 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘log\_hostname’ database flag is set to ‘off’.

"database _ flags " : 
" name" : 
" log_hostname " , 
"value": "Off 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_postgresq1\postgresq1_2> sentinel test 
PASS - ensure-postgreSQL-10g-hostname-is-on. sentinel 
PASS 
PASS 
PASS 
PASS 
- hcl 
- test\ensure-postgreSQL-10g-hostname-is-on\fai11.hc1 
- test\ensure-postgreSQL-10g-hostname-is-on\fai12.hc1 
- hcl 

6.2.6 Ensure That the ‘Log\_min\_messages’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to at least 'Warning'.

***Sentinel Policy Name:***

* + 6.2.6 Ensure That the ‘Log\_min\_messages’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to at least 'Warning'.

***Category:***

* + Cloud SQL(PostgreSQL)

***Description of Policy:***

* + The 'log\_min\_messages' flag defines the minimum message severity level that is considered as an error statement.
  + Messages for error statements are logged with the SQL statement.
  + Valid values include 'DEBUG5', 'DEBUG4', 'DEBUG3', 'DEBUG2', 'DEBUG1', 'INFO', 'NOTICE', 'WARNING', 'ERROR', 'LOG', 'FATAL', and 'PANIC'.
  + Each severity level includes the subsequent levels mentioned above.
  + ERROR is considered the best practice setting.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

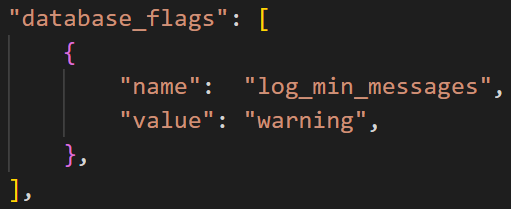
***Sentinel Policy restriction:***

* + In this sentinel policy, “log\_min\_messages” database flag is set to at least “warning”.

***Pass and fail cases of the above sentinel policy***

**Pass Cases:**

1. ‘log\_min\_messages’ database flag is set to ‘warning’.



2. ‘log\_min\_messages’ database flag is set to ‘error’.

"database _ flags " : 
"name" : 
" log_min_messages " , 
"value": "error" 

3. ‘log\_min\_messages’ database flag is set to ‘log’.

" database _ flags " : 
" name" : 
" log_min_messages " , 
"value": "log 

4. ‘log\_min\_messages’ database flag is set to ‘fatal’.

" database _ flags " : 
'name": 
" log_min_messages " , 
"value": "fatal", 

5. ‘log\_min\_messages’ database flag is set to ‘panic’.

" database _ flags " : 
" name" : 
" log_min_messages " , 
"value": "panic" 

**Fail Cases:**

1. When the database flag is null



2. When the ‘log\_min\_messages’ database flag is set to ‘info’

"database _ flags " : 
"name" : 
" log_min_messages " , 
"value": "info" , 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: log min messages database flag is at least set to W 
arning> sentinel test 
PASS - ensure-log-min-messages-database-flag-is-atleast-set-to-warning. sentinel 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
-database-flag-is-atleast-set 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
- test\ensure-log-min 
-messages 
-messages 
-messages 
-messages 
-messages 
-messages 
-messages 
-messages 
-messages 
-messages 
-to-warning\fail. hcl 
-to-warning\faill. hcl 
-to-warning\fai12. hcl 
-to-warning\fai13. hcl 
-to-warning\fai14. hcl 
-to-warning\pass. hcl 
-to-warning\passl. hcl 
-to-warning\pass2. hcl 
-to-warning\pass3. hcl 
-to-warning\pass4. hcl 

6.2.7 Ensure ‘Log\_min\_error\_statement’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘Error’ or Stricter.

***Sentinel Policy Name:***

* + 6.2.7 Ensure ‘Log\_min\_error\_statement’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘Error’ or Stricter.

***Category:***

* + Cloud SQL(PostgreSQL)

***Description of Policy:***

* + The 'log\_min\_error\_statement' flag defines the minimum message severity level that are considered as an error statement.
  + Messages for error statements are logged with the SQL statement.
  + Valid values include DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, NOTICE, WARNING, ERROR, LOG, FATAL, and PANIC.
  + Each severity level includes the subsequent levels mentioned above.
  + We need to ensure a value of 'ERROR' or stricter is set.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “log\_min\_error\_statement” database flag is set to “error” or stricter.

***Pass and fail cases of the above sentinel policy***

**Pass Cases:**

1. ‘log\_min\_error\_statement’ database flag is set to ‘error’.

" database _ flags " : 
"name" : 
"value": "error" 

2. ‘log\_min\_error\_statement’ database flag is set to ‘log’.

" database _ flags " : 
" name" : 
"value": "log 

3. ‘log\_min\_error\_statement’ database flag is set to ‘fatal’.

" database _ flags " : 
"value": "fatal" , 
"name" : 
"log_min_error_statement" 

4. ‘log\_min\_error\_statement’ database flag is set to ‘panic’.

" database _ flags " : 
"name" : 
"value": "panic" 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘log\_min\_error\_statement’ database flag is set to ‘info’.

"database _ flags " : 
" name" : 
"value": "info" , 

3. When the database flag parameter is not present in the sentinel mock.

4. ‘log\_min\_error\_statement’ database flag is set to ‘debug3’.

" database _ flags " : 
"name" : 
"value": "debug3" , 

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_postgresq1\postgresq1_4> sentinel test 
PASS - ensure-postgreSQL-10g-min-error-statement-is-set-to-error-or-stricter.sentine1 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
-error- statement-is-set -to-error 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
PASS 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
- test\ensure-postgreSQL-10g-min 
-or- stricter\fail. hcl 
-or-stricter\faill .hcl 
-or- stricter\fai12. hcl 
-or- stricter\fai13. hcl 
-or- stricter\pass. hcl 
-or-stricter\passl.hcl 
-or- stricter\pass2. hcl 
-or- stricter\pass3. hcl 

6.2.8 Ensure That the ‘Log\_min\_duration\_statement’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘-1′ (Disabled).

***Sentinel Policy Name:***

* + 6.2.8 Ensure That the ‘Log\_min\_duration\_statement’ Database Flag for Cloud SQL PostgreSQL Instance Is Set to ‘-1′ (Disabled).

**Category:**

* + Cloud SQL(PostgreSQL)

**Description of Policy:**

* + The log\_min\_duration\_statement flag defines the minimum amount of execution time of a statement in milliseconds, where the total duration of the statement is logged.
  + SQL statements may include sensitive information, and therefore should not be logged.
  + Ensure that log\_min\_duration\_statement is disabled, i.e., a value of -1 is set.

**Terraform Providers:**

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

**Sentinel Policy restriction:**

* + In this sentinel policy, "log\_min\_duration\_statement” database flag is set to '-1'.

**Pass and fail cases of the above sentinel policy**

**Pass Case:**

1. ‘log\_min\_duration\_statement’ database flag is set to ‘-1’.

" database _ flags " : 
" name" : 
"value": " 1 

**Fail Cases:**

1. When the database flag is null



2. When the ‘log\_min\_duration\_statement’ database flag is set to ‘0’.

" database _ flags " : 
" name" : 
"value": "0 

3. When the ‘log\_min\_duration\_statement’ database flag is set to a value greater than 0.

" database _ flags " : 
'name": 
"value": "2 

4. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_postgresq1\postgresq1_3> sentinel test 
PASS - ensure-postgreSQL-10g-min-duration-statement-is-set-to--1. sentinel 
• -duration-statement-is-set-to- -I\pass.hcl 
PASS 
PASS 
PASS 
PASS 
PASS 
- test\ensure-postgreSQL-10g 
- test\ensure-postgreSQL-10g 
- test\ensure-postgreSQL-10g 
- test\ensure-postgreSQL-10g 
- test\ensure-postgreSQL-10g 
• -duration-statement-is-set 
-nun 
• -duration-statement-is-set 
-nun 
• -duration-statement-is-set 
-nun 
• -duration-statement-is-set 
-nun 
-nun 
-to--l\fail.hcl 

6.2.9 Ensure That 'cloudsql.enable\_pgaudit' Database Flag for each Cloud SQL PostgreSQL Instance Is Set to 'on' For Centralized Logging

***Sentinel Policy Name:***

* 6.2.9 Ensure That 'cloudsql.enable\_pgaudit' Database Flag for each Cloud SQL PostgreSQL Instance Is Set to 'on' For Centralized Logging

***Category:***

* Cloud SQL

***Description of Policy:***

* Ensure 'cloudsql.enable\_pgaudit' database flag for PostgreSQL database instance is set to onto allow for centralized logging.
* This flag and installing the extension (open- source pgaudit extension within PostgreSQL) enables database auditing in PostgreSQL through the open-source pgAudit extension. This extension provides detailed session and object logging to comply with government, financial, & ISO standards and provides auditing capabilities to mitigate threats by monitoring security events on the instance.
* As per the CIS benchmarks, this policy impacts only PostgreSQL database instances (not MySQL or SQL server)

***Sentinel Policy Restriction:***

* For 'google\_sql\_database\_instance' resources with database\_version containing "POSTGRES", this policy ensures that 'cloudsql.enable\_pgaudit' is active within each PostgreSQL database instance

***Terraform attributes:***

* + Provider Ref: [google\_sql\_database\_instance | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance)

***Test cases:***

**Pass cases**

1. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.name' attribute contains "cloudsql.enable\_pgaudit"
2. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.value' attribute contains "on"

"settings": 
"activation_policy" : 
"ALWAYS" , 
"active_directory_config 
"availability_type" : 
"ZONAL" 
"collation": 
null, 
"database_flags " : 
"cloudsql. , 
"value": "on 

**Fail case:**

1. For resource 'google\_sql\_database\_instance' with database\_version containing "POSTGRES", 'database\_flags.name' attribute contains something other than "cloudsql.enable\_pgaudit"

"settings": 
"activation_policy" : 
"ALWAYS" , 
"active_di rectory_config" • 
"avai lability_type" : 
"ZONAL", 
"collation" : 
null, 
"database_flags": 
"name" : "null", 
"value" 

**Testcases Output:**

PASS - 
PASS - hc1 
logs: 
Ensure that database flag .enable_pgaudit• 
is enabled and is set to 
trace: 
- Rule 
Value: 
fal se 
- Rule 
Value: 
fal se 
PASS - hc1 
trace: 
- Rule 
Value: 
- Rule 
Value: 
"main" 
" googl i ns t ance_fl ag_ i s_on" 
"main" 
" googl at abase_ i ns t ance_fl ag_i s_on" 

6.3.1 Ensure 'external scripts enabled' database flag for Cloud SQL SQL Server instance is set to 'off'.

***Sentinel Policy Name:***

* + 6.3.1 Ensure 'external scripts enabled' database flag for Cloud SQL SQL Server instance is set to 'off'.

***Category:***

* + Cloud SQL (SQL Server)

***Description of Policy:***

* + ‘external scripts enabled’ database flag will enable the execution of scripts with certain remote language extensions.
  + By default, this database flag is set to off, but when the Advanced Analytics service is installed, then it can set the value to ‘on’.
  + As the External Scripts Enabled feature allows scripts external to SQL such as files located in an R library to be executed, then it could adversely affect the security of the system.
  + So, ‘external scripts enabled’ database flag should be set to ‘off’.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “external scripts enabled” database flag is set to “off”.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. ‘external scripts enabled’ database flag is set to ‘off’.

"database _ flags " : 
" name" : 
"external scripts enabled" , 
"value": "off 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘external scripts enabled’ database flag is set to ‘on’.

"database _ flags " : 
" name" : 
"external scripts enabled", 
"value": "On 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_sq1server\sq1server_2> sentinel test 
PASS - ensure-external-scripts-database-flag-is-off.sentinel 
PASS 
PASS 
PASS 
PASS 
- hcl 
- test\ensure-external-scripts-database-flag-is-off\faill.hcl 
- test\ensure-externa1-scripts-database-f1ag-is-off\fai12.hc1 
- test\ensure-external-scripts-database-flag 
hcl 

6.3.2 Ensure that the 'cross db ownership chaining' database flag for Cloud SQL SQL Server instance is set to 'off'.

***Sentinel Policy Name:***

* + 6.3.2 Ensure that the 'cross db ownership chaining' database flag for Cloud SQL SQL Server instance is set to 'off'.

***Category:***

* + Cloud SQL (SQL Server)

***Description of Policy:***

* + Enabling 'cross db ownership chaining' will allow you to control cross-database ownership chaining at the database level or to allow cross-database ownership for all databases.
  + Enabling cross db ownership is not recommended, unless all of the databases hosted by the instance of SQL Server must participate in cross database ownership chaining.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “cross db ownership chaining” database flag is set to “off”.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. ‘cross db ownership chaining’ database flag is set to ‘off’.

"database _ flags " : 
" name" : 
"cross db ownership chaining", 
"value": "off 

**Fail Cases:**

1. When the database flag is null



2. When the ‘cross db ownership chaining’ database flag is set to ‘on’.

"database _ flags " : 
" name" : 
"cross db ownership chaining" , 
"value": "on 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_sq1server\sq1server_1> sentinel test 
PASS - ensure-sql-server-cross-db-ownership-database-flag-is-off. sentinel 
- test\ensure-sql-server-cross-db-ownership-database-flag-is-off\fail.hcl 
PASS 
- test\ensure-sql-server-cross-db-ownership-database-flag-is-off\faill.hcl 
PASS 
- test\ensure-sq1-server-cross-db-ownership-database-f1ag-is-off\fai12.hc1 
PASS 
- test\ensure-sql-server-cross-db-ownership-database-flag-is-off\pass.hcl 
PASS 

6.3.3 Ensure 'user Connections' Database Flag for Cloud Sql Sql Server Instance Is Set to a Non-limiting Value.

***Sentinel Policy Name:***

* + 6.3.3 Ensure 'user Connections' Database Flag for Cloud Sql Sql Server Instance Is Set to a Non-limiting Value.

***Category:***

* + Cloud SQL (SQL Server)

***Description of Policy:***

* + The user connections option specifies the maximum number of simultaneous user connections that are allowed on an instance of SQL Server.
  + SQL Server allows a maximum of 32,767 user connections.
  + user connections is by default a self-configuring option, SQL Server adjusts the maximum number of user connections automatically as needed, up to the maximum value allowable.
  + So, user connections database flag is set an appropriate value.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “user connections” database flag is set to a non-limiting value.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. ‘user connections’ database flag is set to an appropriate value.

"database _ flags " : 
" name" : 
"user connections", 
"value": "500 

**Fail Cases:**

1. When the database flag is null



2. When the ‘user connections’ database flag is set to ‘0’.

" database _ flags " : 
"name" : 
"user connections", 
"value": "0 

3. When the database flag parameter is not present in the sentinel mock.

4. user connections database flag is set to a value less than 10.

" database _ flags " : 
"name" : 
"user connections", 
"value": "9 

**Testcases Output:**

PS C: sql server user connections database flag is not 
sentinel test 
PASS - ensure-sql-server-user-connections-database-flag-is-not-e. sentinel 
PASS 
PASS 
PASS 
PASS 
PASS 
- test\ensure-sql 
- test\ensure-sql 
- test\ensure-sql 
- test\ensure-sql 
- test\ensure-sql 
-server-user 
-server-user 
-server-user 
-server-user 
-server-user 
-connections-database-flag 
-connections-database-flag 
-connections-database-flag 
-connections-database-flag 
-connections-database-flag 
• -not-e\fail.hcl 
• -not-e\faill.hcl 
• -not-e\fai12.hc1 
• -not-e\fai13.hc1 
• -not-e\pass. hcl 

6.3.4 Ensure 'user options' database flag for Cloud SQL Server instance is not configured

***Sentinel Policy Name:***

* + 6.3.4 Ensure 'user options' database flag for Cloud SQL Server instance is not configured

***Category:***

* + Cloud SQL

***Description of Policy:***

* + It is recommended that the 'user options' database flag for SQL Server instance should not be configured. 'The user options' flag specifies global defaults for all users. A user can override these defaults by using the SET statement, which introduces security concerns.

***Sentinel Policy Restriction:***

* + The 'user options' database flag is not configured on SQL Server instances

***Terraform attributes:***

* + - Provider Ref: [google\_sql\_database\_instance | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance)

***Test cases:***

**Pass cases**

* + For resource 'google\_sql\_database\_instance' the attribute 'database\_flags.name' is NOT 'user options'

"settings": 
"availability_type": 
"ZONAL" , 
"collation": 
null, 
"database_flags" 
" ame": "null", 
"value : " 

**Fail case:**

* + For resource 'google\_sql\_database\_instance' the attribute 'database\_flags.name' is 'user options'

"settings": [ 
" activation_policy" : 
"active_directory_config 
"availability_type" : 
"collation": 
"database_flags": 
"name" : 
"value". 
"ALWAYS" , 
"ZONAL", 
null, 
"user options" 
"1024% 

**Testcases Output:**

PASS - sentinel 
PASS - hcl 
logs: 
Make sure that the •User c»tions• database flag is rot selected/ enabled/ configured 
SQL_Server_User_Options_flag_is_not_configured.sentinel:3e:1 - Rule 
Value: 
false 
SQL_Server_User_Options_flag_is_not_configured.sentinel:16:1 • Rule 
Value: 
false 
PASS - 
trace: 
: 3e:1 - Rule 
Value: 
- Rule 
Value: 
"main" 
"DB_f1ags" 
"main" 
"DB_f1ags" 

6.3.5 Ensure 'remote access' database flag for Cloud SQL SQL Server instance is set to 'off'.

***Sentinel Policy Name:***

* + 6.3.5 Ensure 'remote access' database flag for Cloud SQL SQL Server instance is set to 'off'.

***Category:***

* + Cloud SQL (SQL Server)

***Description of Policy:***

* + The remote access option controls the execution of stored procedures from local or remote servers on which instances of SQL Server are running.
  + By Default, remote access option is set to 'on'
  + To prevent local stored procedures from being run from a remote server or remote stored procedures from being run on the local server, this must be disabled.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, ‘remote access” database flag is set to “off”.

***Pass and fail cases of the above sentinel policy:***

**Pass Case:**

1. ‘remote access’ database flag is set to ‘off’.

" database _ flags " : 
"name" : 
"remote access", 
"value": "off 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘remote access’ database flag is set to ‘on’.

"database _ flags " : 
"name" : 
"remote access", 
"value": 'on 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_sq1server\sq1server_3> sentinel test 
PASS - ensure-remote-access 
PASS 
PASS 
PASS 
PASS 
- test\ensure-remote 
- test\ensure-remote 
- test\ensure-remote 
- test\ensure-remote 
-database-flag-is -off. sentinel 
-access-database-flag-is-off\fail. hcl 
-access-database-flag-is-off\faill. hcl 
-access-database-flag-is-off\fai12. hcl 
-access-database-flag-is-off\pass.hcl 

6.3.6 Ensure '3625 (trace flag)' database flag for all Cloud SQL Server instances is set to 'off'.

***Sentinel Policy Name:***

* + 6.3.6 Ensure '3625 (trace flag)' database flag for all Cloud SQL Server instances is set to 'off'.

***Category:***

* + Cloud SQL (SQL Server)

***Description of Policy:***

* + Trace flags are frequently used to diagnose performance issues or to debug stored procedures or complex computer systems.
  + 3625(trace log) Limits the amount of information returned to users who are not members of the sysadmin fixed server role, by masking the parameters of some error messages using '\*\*\*\*\*'.
  + This can help prevent disclosure of sensitive information, hence this is recommended to disable this flag.
  + It is recommended to set this flag globally to off to prevent the flag having been left on, or turned on by bad actors.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “3625” database flag is set to “off”.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. ‘3625’ database flag is set to ‘off’.

" database _ flags " : 
"name" : 
"3625" , 
"value": "off 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘3625’ database flag is set to ‘on’.

" database _ flags " : 
"name" : 
"3625" , 
"value": 'on 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_sq1server\sq1server_4> sentinel test 
PASS - ensure-3625-database-f1ag-is-off. sentinel 
PASS 
PASS 
PASS 
PASS 
- test\ensure-3625-database-f1ag-is-off\fai1.hc1 
- hcl 
- hcl 
- test\ensure-3625-database-f1ag-is-off\pass.hc1 

6.3.7 Ensure that the 'contained database authentication' database flag for Cloud SQL on the SQL Server instance is set to 'off'.

***Sentinel Policy Name:***

* + 6.3.7 Ensure that the 'contained database authentication' database flag for Cloud SQL on the SQL Server instance is set to 'off'.

***Category:***

* + - Cloud SQL (SQL Server)

***Description of Policy:***

* + A contained database includes all database settings and metadata required to define the database and has no configuration dependencies on the instance of the Database Engine where the database is installed.
  + Setting contained database authentication to 'on' will enable the use of Contained databases, which is not recommend for most cases due to potential security threats.
  + Most of the threats are related to the USER WITH PASSWORD authentication process, which moves the authentication boundary from the Database Engine level to the database level.

***Terraform Providers:***

<https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/sql_database_instance> - resource block “google\_sql\_datadase\_instance” is used.

***Sentinel Policy restriction:***

* + In this sentinel policy, “contained database authentication” database flag is set to “off”.

***Pass and fail cases of the above sentinel policy***

**Pass Case:**

1. ‘contained database authentication’ database flag is set to ‘off’.

"database _ flags " : 
" name" : 
"contained database authentication" , 
"value": "off 

**Fail Cases:**

1. When the database flag is null.



2. When the ‘contained database authentication’ database flag is set to ‘on’.

" database _ flags " : 
"name" : 
" contained 
"value": 'on 
database 
authentication" , 

3. When the database flag parameter is not present in the sentinel mock.

**Testcases Output:**

PS C: \Users\1939847\Documents\c10ud_sq1server\sq1server_5> sentinel test 
PASS - ensure-contained-database-authentication-database-flag-is-off. sentinel 
PASS 
PASS 
PASS 
PASS 
- test\ensure-contained-database-authentication-database-flag-is-off\fail.hcl 
- test\ensure-contained-database-authentication-database-flag-is-off\faill.hcl 
- test\ensure-contained-database-authentication-database-f1ag-is-off\fai12.hc1 
- test\ensure-contained-database-authentication-database-flag-is-off\pass.hcl 