# 

**CSWG- Policy as Code Documentation**

BigQuery Sentinel Policies

7.1 Ensure That BigQuery Datasets Are Not Anonymously or Publicly Accessible.

***Sentinel Policy Name:***

* + 7.1 Ensure That BigQuery Datasets Are Not Anonymously or Publicly Accessible.

***Category:***

* + BigQuery

***Description of Policy:***

* + Granting permissions to 'allUsers' or 'allAuthenticatedUsers' allows anyone to access the BigQuery dataset. Such access might not be desirable if sensitive data is being stored in the dataset. Therefore, ensure that anonymous and/or public access to a dataset is not allowed.

***Sentinel Policy Restriction:***

* + It is recommended that the IAM policy on BigQuery datasets does not allow anonymous and/or public access.

***Terraform attributes:***

* + - Provider Ref: [google\_bigquery\_dataset\_iam | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/bigquery_dataset_iam#google_bigquery_dataset_iam_policy)

***Test cases:***

**Pass cases**

* + For resource 'google\_bigquery\_dataset\_iam\_binding' and 'google\_bigquery\_dataset\_iam\_policy' the 'members' attribute does NOT contain 'allUsers' or 'allAuthenticatedUsers'

"after" : 
"condition" : 
"dataset id": 
"members": [ 
o, 
"example_dataset" , 

**Fail case:**

* + For resource 'google\_bigquery\_dataset\_iam\_binding' and 'google\_bigquery\_dataset\_iam\_policy' the 'members' attribute contains 'allUsers' and/or 'allAuthenticatedUsers'

"values": 
"condition" : 
"dataset id"• 
"nembers": [ 
o, 
"example_dataset", 
uwthenticatedUsers" , 

**Testcases Output:**

PASS - Ensure That Datasets Are Arnrmmusly or Publicly Accessible. sentinel 
PASS - test\Ensure That Ihtasets Are or Publicly Accessible\fail.hcl 
trace : 
Ensure That BiQJery Datasets Are Anonymously or Publicly Accessible. sentinel - 
Rule "main" 
Value : 
false 
Ensure That BiQJery Datasets Are ret Anonymusly or Publicly Accessible. sentinel : 17 : 1 
- Rule Dat 
Value : 
false 
PASS - test\Ensure That Biß2tpt•y Datasets Are Anonpusly or Publicly Accessible\pass. hcl 
Ensure That Datasets Are ArK»nymus1y or Publicly 
Value : 
Ensure That Datasets Are Anonymously or Publicly 
Value : 
Accessible. sentinel : 27 : 1 
Accessible. sentinel : 17 : 1 
- Rule "main" 
- Rule 

7.2 Ensure That All BigQuery Tables Are Encrypted With Customer Managed Encryption Key (CMEK)

***Sentinel Policy Name:***

7.2 Ensure That All BigQuery Tables Are Encrypted With Customer Managed Encryption Key (CMEK)

***Category:***

* + Big Query

***Description of Policy:***

* + BigQuery by default encrypts the data as rest by employing Envelope Encryption using Google managed cryptographic keys. The data is encrypted using the data encryption keys and data encryption keys themselves are further encrypted using key encryption keys. This is seamless and do not require any additional input from the user. However, if you want to have greater control, Customer-managed encryption keys (CMEK) can be used as encryption key management solution for BigQuery Data Sets. If CMEK is used, the CMEK is used to encrypt the data encryption keys instead of using google-managed encryption keys.

* + For greater control over the encryption, customer-managed encryption keys (CMEK) can be used as encryption key management solution for BigQuery tables. The CMEK is used to encrypt the data encryption keys instead of using google-managed encryption keys. BigQuery stores the table and CMEK association and the encryption/decryption is done automatically.

Applying the Default Customer-managed keys on BigQuery data sets ensures that all the new tables created in the future will be encrypted using CMEK but existing tables need to be updated to use CMEK individually.

***Sentinel Policy Restriction:***

* + The policy checks whether "**encryption\_configuration**" block contains "**kms\_key\_name**"

***Terraform attributes:***

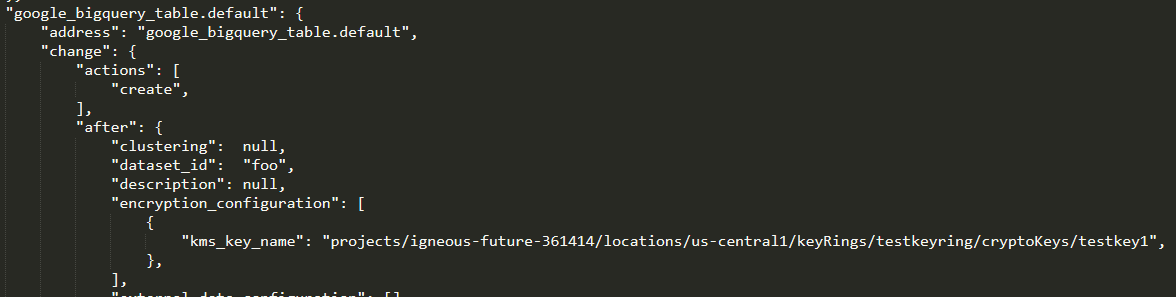
* + - Provider Ref:[google\_bigquery\_table | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/bigquery_table)
    - Terraform attribute: "encryption\_configuration" block

***Test cases:***

**Pass cases**

1. In pass case, if "encryption\_configuration" block contains "kms\_key\_name" then the pass case will always pass.

Screenshot of argument in mock file



**Fail case:**

1. In pass case, if "encryption\_configuration" block does not contains "kms\_key\_name" then the pass case will always fail.

Screenshot of argument in mock file

" . default" : { 
" address" : " default" , 
"change" • 
actions" : 
create" 
" after 
"clustering 
"dataset id": 
"description" : 
encryption_configuration" : 
null, 
null, 

**Testcases Output:**

Screenshot of test command successful

user@DESKTOP-GLNSQ3C:/mnt/e/ALL Wls/sentinel_hashicorp/policytesting$ sentinel test 
Installing test modules for test/ensure-bigquerytable-has-customerkey/fail . hcl 
module tfplan-functions marked for installation 
Installation complete for test/ensure-bigquerytable-has-customerkey/fail.hcl 
Installing test modules for test/ensure-bigquerytable-has-customerkey/pass . hcl 
module tfplan-functions marked for installation 
Installation complete for test/ensure-bigquerytable-has-customerkey/pass . hcl 
ensure - bigquerytable - has - customerkey . sentinel 
PASS 
test/ ensure - bigquerytable - has - customerkey 'fail . hcl 
PASS 
logs : 
- run-ensure- bigquerytable- has- customerkey 
- verbose 
google_bigquery_table.default has encryption _ configuration.8 that is missing, null, or is not a map or a list. It should have had these items: [kms_key_name] 
Policy failed because BigQuery Table referenced in the mock file is not encrypted with Customer managed Encryption Key (CMEK). Please use encryption _ configuration to include key. 
trace: 
ensure-bigquerytable-has-customerkey . sentinel : 28: 1 
Rule 
Value: 
false 
test/ ensure - bigquerytable - has - customerkey / pass . hcl 
PASS 
trace: 
ensure-bigquerytable-has-customerkey . sentinel : 28: 1 
Value: 
true 
Rule 
"main" 
"main" 

7.3 Ensure That a Default Customer-Managed Encryption Key (CMEK) Is Specified for All BigQuery Data Sets

***Sentinel Policy Name:***

7.3 Ensure That a Default Customer-Managed Encryption Key (CMEK) Is Specified for All BigQuery Data Sets

***Category:***

* + Big Query

***Description of Policy:***

* + BigQuery by default encrypts the data as rest by employing Envelope Encryption using Google managed cryptographic keys. The data is encrypted using the data encryption keys and data encryption keys themselves are further encrypted using key encryption keys. This is seamless and do not require any additional input from the user. However, if you want to have greater control, Customer-managed encryption keys (CMEK) can be used as encryption key management solution for BigQuery Data Sets.
  + For greater control over the encryption, customer-managed encryption keys (CMEK) can be used as encryption key management solution for BigQuery Data Sets. Setting a Default Customer-managed encryption key (CMEK) for a data set ensure any tables created in future will use the specified CMEK if none other is provided.

***Sentinel Policy Restriction:***

* + The policy checks whether "**default\_encryption\_configuration**" block contains "**kms\_key\_name**"

***Terraform attributes:***

* + - Provider Ref:[google\_bigquery\_dataset | Resources | hashicorp/google | Terraform Registry](https://registry.terraform.io/providers/hashicorp/google/latest/docs/resources/bigquery_dataset)
    - Terraform attribute: "default\_encryption\_configuration" block

***Test cases:***

**Pass cases**

1. In pass case, if "default\_encryption\_configuration" block contains "kms\_key\_name" then the pass case will always pass.

Screenshot of argument in mock file

'I default 
encryption configuration" : 
"kms_key_name": "projects/igneous-future-361414/10cations/us-centra11/keyRings/testkeyring/cryptoKeys/testkey1" , 

**Fail case:**

1. In pass case, if "default\_encryption\_configuration" block does not contains "kms\_key\_name" 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-ensure-bigquerydataset-has-defaultcustomerkey 
Installing test modules for test/ensure-bigquerydataset-has-defaultcustomerkey/fail . hcl 
module tfplan-functions marked for installation 
Installation complete for test/ensure-bigquerydataset-has-defaultcustomerkey/fail.hcl 
Installing test modules for test/ensure-bigquerydataset-has-defaultcustomerkey/pass . hcl 
module tfplan-functions marked for installation 
Installation complete for test/ensure-bigquerydataset-has-defaultcustomerkey/pass . hcl 
ensure - bigquerydataset - has - defaultcustomerkey . sentinel 
PASS 
test/ ensure - bigquerydataset - has - defaultcustomerkey 'fail . hcl 
PASS 
logs : 
- verbose 
google_bigquery_dataset.dataset has default _ encryption _ configuration.a that is missing, null, or is not a map or a list. It should have had these items: [kms_key_name] 
Policy failed because BigQuery datasets referenced in the mock file is not encrypted with Default Customer managed Encryption Key (CMEK). Please use default _ encryption _ configuration to include key. 
trace: 
ensure - bigquerydataset - has - defaultcustomerkey . sentinel : 18 : 1 
Rule 
Value: 
false 
test/ ensure - bigquerydataset - has - defaultcustomerkey / pass . hcl 
PASS 
trace: 
ensure - bigquerydataset - has - defaultcustomerkey . sentinel : 18 : 1 
Value: 
true 
Rule 
"main" 
Act 
"main" 
Go t' 