

































-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML



## CloudFormation static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your CLOUDFORMATION code

- All rules 27
-  Vulnerability 3
-  Security Hotspot 20
-  Code Smell 4

### "Log Groups" should be configured with a retention policy

 Code Smell

### Defining a short backup retention duration is security-sensitive

 Security Hotspot

### Using unencrypted EFS file systems is security-sensitive

 Security Hotspot

### Using unencrypted SQS queues is security-sensitive

 Security Hotspot

### Using unencrypted SNS topics is security-sensitive

 Security Hotspot

### Using unencrypted SageMaker notebook instances is security-sensitive

 Security Hotspot

### Using unencrypted Elasticsearch domains is security-sensitive

 Security Hotspot

### Using unencrypted RDS databases is security-sensitive

 Security Hotspot

### Using unencrypted EBS volumes is security-sensitive

 Security Hotspot

### Disabling logging is security-sensitive

 Security Hotspot

### "Log Groups" should be declared explicitly

 Code Smell

### Administration services access should be restricted to specific IP addresses

 Vulnerability

### Disabling versioning of S3 buckets is security-sensitive

 Security Hotspot

Tags

Search by name...



### "Log Groups" should be configured with a retention policy

Analyze your code

 Code Smell  Critical  convention aws

Log streams created on AWS will stay forever unless the `AWS::Logs::LogGroup` to which they belong to was configured with a retention policy. `Log Groups` should have their “RetentionInDays” property set with a valid value to be sure the log events are kept only for the expected duration.

When the property is not set, the log events will be kept for ever or will be deleted only when the Log Group is removed.

Keeping the logs for ever doesn't come for free: AWS will charge for keeping these logs. Also from a security point of view, keeping the data for ever may be not compliant with company policy or regulatory rules.

Note: this rule doesn't check if the value provided to "RetentionInDays" is valid because AWS CloudFormation Linter (cfn-lint) do it already

#### Noncompliant Code Example

```
MyLambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    Runtime: nodejs12.x
    Description: Example of Lambda Function

MyFunctionLogGroup:
  Properties:
    LogGroupName: !Join ['/', ['aws/lambda', !Ref MyLambdaFunction]]
    # Noncompliant: "RetentionInDays" property is not set: logs are kept for ever
```

#### Compliant Solution

```
MyLambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    Runtime: nodejs12.x
    Description: Example of Lambda Function

MyFunctionLogGroup:
  Properties:
    LogGroupName: !Join ['/', ['aws/lambda', !Ref MyLambdaFunction]]
    RetentionInDays: 30
```

Available In:

sonarcloud  | sonarqube 