Method of Procedure

Les Schwab  
WOPR v2 – AWS Central Logging

Prepared By Brian Keith

Five Talent Software

1001 SW Emkay Dr., Suite 120

Bend, OR 97702

800-770-1868

www.FiveTalent.com

Contents

[1. Document History: 2](#_Toc2110430)

[2. Summary 2](#_Toc2110431)

[3. Supporting Information: 2](#_Toc2110432)

[4. Schedule and Downtime: 3](#_Toc2110433)

[5. Customer Notification Announcement 3](#_Toc2110434)

[6. Description of Work: 3](#_Toc2110435)

[7. Supporting Diagrams: 3](#_Toc2110436)

[8. Tasks Prior to Implementation: 3](#_Toc2110437)

[9. Implementation and configuration Tasks: 3](#_Toc2110438)

[9.1. Detail Tasks go here 3](#_Toc2110439)

[10. Tasks after Implementation: 4](#_Toc2110440)

[11. References 4](#_Toc2110441)

[*11.1.* *Add references here* 4](#_Toc2110442)

[12. Appendix A – CloudFormation Script 4](#_Toc2110443)

# Document History:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Person | Notes, Comments, Reasons |
| 0.1 | <<Date>> | Brian Keith | First Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Summary

This project will “provides infrastructure and configuration information for deploying a centralized logging solution that collects, analyzes, and displays logs on AWS across multiple accounts and AWS Regions. The solution uses Amazon Elasticsearch Service (Amazon ES), a managed service that simplifies the deployment, operation, and scaling of Elasticsearch clusters in the AWS Cloud, as well as Kibana, an analytics and visualization platform that is integrated with Amazon ES.” [Centralized Logging on AWS]

# Supporting Information:

**Work Locations:**

**Five Talent Project Leader:** Shelby Cunningham

**Five Talent Technical Leader:** Brian Keith 800-770-1868, brian.keith@fivetalent.com

**Customer Project Leader:**

**Customer Contact Information:**

**Manufacturer Contact Information:**

# Schedule and Downtime:

**Onsite Schedule: none expected**

**Downtime Scheduled - none expected**

**Applications and Services Impacted:**

# Customer Notification Announcement

**Description of how downtime was announced to customer and users:**

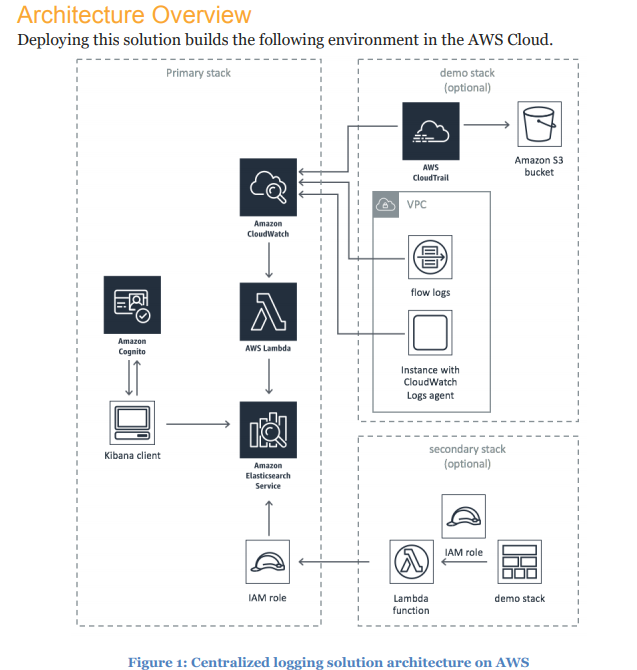
**Email announcement description:**

# Description of Work:

<<*Add work description here>>*

# Supporting Diagrams:

*<<Add diagrams>>*

**

# Tasks Prior to Implementation:

1. Task 1 –
2. Task 2 –
3. Task 3

# Implementation and configuration Tasks:

## Detail Tasks go here

|  |  |  |
| --- | --- | --- |
| Step | Process | Screen Shot |
|  | Log in to the AWS Master Account and navigate to the CloudFormation Service console |  |
|  | Click on Create Stack |  |
|  | Upload the template, FT\_LesSchwab\_WOPR\_centralized-logging-primary.template  Click **Next** |  |
|  | Enter a stack name and the necessary parameters | d |
|  | Enter Tags |  |
|  | Review settings, acknowledge the warning  Click **Create Stack** |  |
|  | An email will be received for SNS Notification subscription.  Click **Confirm Subscription** |  |
|  |  |  |
|  |  |  |
|  | Procedure Complete. |  |

1. Task 1a -
2. Task 2 – Milestone 1
3. Task 3 – Milestone 2
4. Task 4. – Milestone 3

# Tasks after Implementation:

1. Documentation
2. Service
3. Support
4. Time replication and adjust as needed.

# References

## Centralized Logging on AWS - <https://docs.aws.amazon.com/solutions/latest/centralized-logging/welcome.html>

# Appendix A – CloudFormation Scripts

*Include scripts here. Also keep as separate .json or .yml files for distribution and deployment*

# Centralized Logging Solution

#

# template for centralized-logging-solution

# \*\*DO NOT DELETE\*\*

#

# author: aws-solutions-builder@

AWSTemplateFormatVersion: 2010-09-09

Description: (SO0009) - AWS Centralized Logging Solution, primary template

Parameters:

# Name for ES Domain

DOMAINNAME:

Description: Name for the Amazon ES domain that this template will create. Domain names must start with a lowercase letter and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9.

Type: String

Default: centralizedlogging

# Email address for the Elasticsearch domain admin

DomainAdminEmail:

Description: The email address of the Elasticsearch domain admin who will receive CloudWatch alarm notifications.

Type: String

Default: esdomainadmin@example.com

AllowedPattern: '^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$'

Description: E-mail address of the Elasticsearch admin

# Email address for Cognito Admin user

CognitoAdminEmail:

Type: String

Default: cognitoadmin@example.com

AllowedPattern: '^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$'

Description: E-mail address of the Cognito admin

# ES cluster size

ClusterSize:

Description: Amazon ES cluster size; small (2 data nodes), medium (4 data nodes), large (10 data nodes)

Type: String

Default: Small

AllowedValues:

- Small

- Medium

- Large

# Demo template for sample logs

DemoTemplate:

Description: Deploy template for sample data and logs?

Type: String

Default: 'No'

AllowedValues:

- 'Yes'

- 'No'

# Spoke accounts which would use the same ES

SpokeAccounts:

Description: Account IDs which you want to allow for centralized logging (comma separated list eg. 11111111,22222222)

Type: CommaDelimitedList

# VPC CIDR for sample sources

DemoVPC:

Description: CIDR for VPC with sample sources (Only required if you chose 'Yes' above)

Type: String

MinLength: 9

MaxLength: 18

AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'

Default: 10.250.0.0/16

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x

# Subnet for sample web server

DemoSubnet:

Description: IP address range for subnet with sample web server (Only required if you chose 'Yes' above)

Type: String

MinLength: 9

MaxLength: 18

AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'

Default: 10.250.250.0/24

ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x

Metadata:

AWS::CloudFormation::Interface:

ParameterGroups:

- Label:

default: Elasticsearch Configuration

Parameters:

- DOMAINNAME

- DomainAdminEmail

- ClusterSize

- SpokeAccounts

- Label:

default: Cognito Configuration

Parameters:

- CognitoAdminEmail

- Label:

default: Do you want to deploy sample log sources?

Parameters:

- DemoTemplate

- DemoVPC

- DemoSubnet

ParameterLabels:

CognitoAdminEmail:

default: Cognito Admin email address

DomainAdminEmail:

default: Elasticsearch Domain Admin email address

DOMAINNAME:

default: Elasticsearch Domain name

ClusterSize:

default: Cluster Size

DemoTemplate:

default: Sample Logs

SpokeAccounts:

default: Spoke Accounts

DemoVPC:

default: VPC CIDR for Sample Sources

DemoSubnet:

default: Subnet for Sample Web Server

Mappings:

InstanceMap:

t2.micro: {"Arch":"HVM64"}

us-east-1: {"instancetype":"t2.micro"}

us-east-2: {"instancetype":"t2.micro"}

us-west-1: {"instancetype":"t2.micro"}

us-west-2: {"instancetype":"t2.micro"}

ca-central-1: {"instancetype":"t2.micro"}

eu-west-1: {"instancetype":"t2.micro"}

eu-central-1: {"instancetype":"t2.micro"}

eu-west-2: {"instancetype":"t2.micro"}

eu-west-3: {"instancetype":"t2.micro"}

ap-southeast-1: {"instancetype":"t2.micro"}

ap-southeast-2: {"instancetype":"t2.micro"}

ap-northeast-1: {"instancetype":"t2.micro"}

ap-northeast-2: {"instancetype":"t2.micro"}

ap-south-1: {"instancetype":"t2.micro"}

sa-east-1: {"instancetype":"t2.micro"}

send-data: {"SendAnonymousData": "Yes"}

# V57271571 - 10/09/2018 - update tshirt size

InstanceSizing:

elasticsearch:

Small: i3.large.elasticsearch

Medium: i3.2xlarge.elasticsearch

Large: i3.4xlarge.elasticsearch

MasterSizing:

elasticsearch:

Small: c4.large.elasticsearch

Medium: c4.large.elasticsearch

Large: c4.large.elasticsearch

NodeCount:

elasticsearch:

Small: '4'

Medium: '6'

Large: '6'

# Lambda source code mapping

SourceCode:

General:

S3Bucket: "solutions"

KeyPrefix: "centralized-logging/v2.2.0"

Conditions:

DemoData: !Equals [!Ref DemoTemplate, 'Yes']

SingleAccnt: !Equals [!Select [ 0, !Ref SpokeAccounts ], '']

Resources:

#

# Cognito and IAM

#

# Creates a user pool in cognito to auth against

UserPool:

Type: 'AWS::Cognito::UserPool'

Properties:

UserPoolName: !Sub ${DOMAINNAME}\_kibana\_access

AutoVerifiedAttributes:

- email

MfaConfiguration: 'OFF'

EmailVerificationSubject: !Ref AWS::StackName

Schema:

- Name: name

AttributeDataType: String

Mutable: true

Required: true

- Name: email

AttributeDataType: String

Mutable: false

Required: true

# Creates a needed group in Cognito for Kibana access

UserPoolGroup:

Type: "AWS::Cognito::UserPoolGroup"

Properties:

Description: 'User pool group for Kibana access'

GroupName: !Sub ${DOMAINNAME}\_kibana\_access\_group

Precedence: 0

UserPoolId: !Ref UserPool

# Creates a User Pool Client to be used by the identity pool

UserPoolClient:

Type: 'AWS::Cognito::UserPoolClient'

Properties:

ClientName: !Sub ${DOMAINNAME}-client

GenerateSecret: false

UserPoolId: !Ref UserPool

# Creates a federated Identity pool

IdentityPool:

Type: 'AWS::Cognito::IdentityPool'

Properties:

IdentityPoolName: !Sub ${DOMAINNAME}Identity

AllowUnauthenticatedIdentities: true

CognitoIdentityProviders:

- ClientId: !Ref UserPoolClient

ProviderName: !GetAtt UserPool.ProviderName

# Create a role for unauthorized access to AWS resources. Very limited access.

# Only allows users in the previously created Identity Pool

CognitoUnAuthorizedRole:

Type: 'AWS::IAM::Role'

Properties:

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: 'Allow'

Principal:

Federated: 'cognito-identity.amazonaws.com'

Action:

- 'sts:AssumeRoleWithWebIdentity'

Condition:

StringEquals:

'cognito-identity.amazonaws.com:aud': !Ref IdentityPool

'ForAnyValue:StringLike':

'cognito-identity.amazonaws.com:amr': unauthenticated

Policies:

- PolicyName: 'CognitoUnauthorizedPolicy'

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: 'Allow'

Action:

- 'mobileanalytics:PutEvents'

- 'cognito-sync:BulkPublish'

- 'cognito-sync:DescribeIdentityPoolUsage'

- 'cognito-sync:GetBulkPublishDetails'

- 'cognito-sync:GetCognitoEvents'

- 'cognito-sync:GetIdentityPoolConfiguration'

- 'cognito-sync:ListIdentityPoolUsage'

- 'cognito-sync:SetCognitoEvents'

- 'congito-sync:SetIdentityPoolConfiguration'

Resource: !Sub 'arn:aws:cognito-identity:${AWS::Region}:${AWS::AccountId}:identitypool/${IdentityPool}'

# Create a role for authorized access to AWS resources.

# Only allows users in the previously created Identity Pool

CognitoAuthorizedRole:

Type: 'AWS::IAM::Role'

Properties:

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: 'Allow'

Principal:

Federated: 'cognito-identity.amazonaws.com'

Action:

- 'sts:AssumeRoleWithWebIdentity'

Condition:

StringEquals:

'cognito-identity.amazonaws.com:aud': !Ref IdentityPool

'ForAnyValue:StringLike':

'cognito-identity.amazonaws.com:amr': authenticated

Policies:

- PolicyName: 'CognitoAuthorizedPolicy'

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: 'Allow'

Action:

- 'mobileanalytics:PutEvents'

- 'cognito-sync:BulkPublish'

- 'cognito-sync:DescribeIdentityPoolUsage'

- 'cognito-sync:GetBulkPublishDetails'

- 'cognito-sync:GetCognitoEvents'

- 'cognito-sync:GetIdentityPoolConfiguration'

- 'cognito-sync:ListIdentityPoolUsage'

- 'cognito-sync:SetCognitoEvents'

- 'congito-sync:SetIdentityPoolConfiguration'

- 'cognito-identity:DeleteIdentityPool'

- 'cognito-identity:DescribeIdentityPool'

- 'cognito-identity:GetIdentityPoolRoles'

- 'cognito-identity:GetOpenIdTokenForDeveloperIdentity'

- 'cognito-identity:ListIdentities'

- 'cognito-identity:LookupDeveloperIdentity'

- 'cognito-identity:MergeDeveloperIdentities'

- 'cognito-identity:UnlikeDeveloperIdentity'

- 'cognito-identity:UpdateIdentityPool'

Resource: !Sub 'arn:aws:cognito-identity:${AWS::Region}:${AWS::AccountId}:identitypool/${IdentityPool}'

CognitoESAccessRole:

Type: 'AWS::IAM::Role'

Properties:

ManagedPolicyArns:

- arn:aws:iam::aws:policy/AmazonESCognitoAccess

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: 'Allow'

Principal:

Service: 'es.amazonaws.com'

Action:

- 'sts:AssumeRole'

# Assigns the roles to the Identity Pool

IdentityPoolRoleMapping:

Type: 'AWS::Cognito::IdentityPoolRoleAttachment'

Properties:

IdentityPoolId: !Ref IdentityPool

Roles:

authenticated: !GetAtt CognitoAuthorizedRole.Arn

unauthenticated: !GetAtt CognitoUnAuthorizedRole.Arn

AdminUser:

Type: 'AWS::Cognito::UserPoolUser'

Properties:

DesiredDeliveryMediums:

- 'EMAIL'

UserAttributes:

- Name: email

Value: !Ref CognitoAdminEmail

Username: !Ref CognitoAdminEmail

UserPoolId: !Ref UserPool

# Custom resource to configure Cognito and ES

SetupESCognito:

Type: 'Custom::SetupESCognito'

Version: 1.0

Properties:

ServiceToken: !GetAtt LambdaESCognito.Arn

Domain: !Ref DOMAINNAME

CognitoDomain: !Sub ${DOMAINNAME}-${AWS::AccountId}

UserPoolId: !Ref UserPool

IdentityPoolId: !Ref IdentityPool

RoleArn: !GetAtt CognitoESAccessRole.Arn

LambdaESCognito:

Type: 'AWS::Lambda::Function'

Properties:

Description: Centralized Logging - Lambda function to enable cognito authentication for kibana

Environment:

Variables:

# V56536055 - 10/08/2018 - better logging capabilities

LOG\_LEVEL: 'INFO' #change to WARN, ERROR or DEBUG as needed

Handler: index.handler

Runtime: nodejs8.10

Timeout: 300

Role: !GetAtt LambdaESCognitoRole.Arn

Code:

S3Bucket: !Join ["-", [!FindInMap ["SourceCode", "General", "S3Bucket"], Ref: "AWS::Region"]]

S3Key: !Join ["/", [!FindInMap ["SourceCode", "General", "KeyPrefix"], "clog-auth.zip"]]

LambdaESCognitoRole:

Type: AWS::IAM::Role

DependsOn: ElasticsearchAWSLogs

Properties:

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Principal:

Service:

- lambda.amazonaws.com

Action:

- sts:AssumeRole

Path: "/"

Policies:

- PolicyName: root

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Action:

- logs:CreateLogGroup

- logs:CreateLogStream

- logs:PutLogEvents

Resource: arn:aws:logs:\*:\*:\*

- Effect: Allow

Action:

- es:UpdateElasticsearchDomainConfig

Resource: !Sub 'arn:aws:es:${AWS::Region}:${AWS::AccountId}:domain/${DOMAINNAME}'

- Effect: Allow

Action:

- cognito-idp:CreateUserPoolDomain

- cognito-idp:DeleteUserPoolDomain

Resource: !GetAtt UserPool.Arn

- Effect: Allow

Action:

- iam:PassRole

Resource: !GetAtt CognitoESAccessRole.Arn

#

# Primer Elasticsearch resources

# [LoggingMasterRole, LoggingMasterPolicies, ElasticsearchAWSLogs]

#

LoggingMasterRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Principal:

AWS:

Fn::If:

- SingleAccnt

- Ref: AWS::AccountId

- Ref: SpokeAccounts

Service: lambda.amazonaws.com

Action:

- sts:AssumeRole

Path: "/"

LoggingMasterPolicies:

Type: AWS::IAM::Policy

Properties:

PolicyName: !Sub logging-master-${AWS::Region}

PolicyDocument:

Version: '2012-10-17'

Statement:

- Effect: Allow

Action:

- es:ESHttpPost

Resource: !Sub arn:aws:es:${AWS::Region}:${AWS::AccountId}:domain/\*

Roles:

- !Ref LoggingMasterRole

ElasticsearchAWSLogs:

Type: AWS::Elasticsearch::Domain

DeletionPolicy: Retain

Properties:

DomainName: !Ref DOMAINNAME

ElasticsearchVersion: 6.3

# V5804310 - 10/09/2018 - ES Encryption at rest

EncryptionAtRestOptions:

Enabled: true

ElasticsearchClusterConfig:

DedicatedMasterEnabled: true

InstanceCount: !FindInMap [NodeCount, elasticsearch, !Ref ClusterSize]

ZoneAwarenessEnabled: true

InstanceType: !FindInMap [InstanceSizing, elasticsearch, !Ref ClusterSize]

DedicatedMasterType: !FindInMap [MasterSizing, elasticsearch, !Ref ClusterSize]

DedicatedMasterCount: 3

SnapshotOptions:

AutomatedSnapshotStartHour: '1'

AccessPolicies:

Version: 2012-10-17

Statement:

- Action: 'es:\*'

Principal:

AWS: !Sub ${LoggingMasterRole.Arn}

Effect: Allow

Resource: !Sub arn:aws:es:${AWS::Region}:${AWS::AccountId}:domain/\* #removing domain name due to cyclic dependency

- Action: 'es:\*'

Principal:

AWS: !Sub

- arn:aws:sts::${AWS::AccountId}:assumed-role/${AuthRole}/CognitoIdentityCredentials

- { AuthRole: !Ref CognitoAuthorizedRole }

Effect: Allow

Resource: !Sub 'arn:aws:es:${AWS::Region}:${AWS::AccountId}:domain/${DOMAINNAME}/\*'

# V57095985 - 10/08/2018 - ES Domain needed configurations

# https://github.com/awslabs/aws-centralized-logging/issues/2

AdvancedOptions:

rest.action.multi.allow\_explicit\_index: 'true'

indices.fielddata.cache.size: 40

#

# SNS Topic

#

Topic:

Type: 'AWS::SNS::Topic'

Properties:

DisplayName: 'Centralized Logging CloudWatch alarms notification topic'

TopicPolicy:

Type: 'AWS::SNS::TopicPolicy'

Properties:

PolicyDocument:

Id: Id1

Version: '2012-10-17'

Statement:

- Sid: Sid1

Effect: Allow

Principal:

AWS: !Sub '${AWS::AccountId}' # Allow CloudWatch Alarms

Action: 'sns:Publish'

Resource: '\*'

Topics:

- !Ref Topic

TopicEndpointSubscription:

DependsOn: TopicPolicy

Type: 'AWS::SNS::Subscription'

Properties:

Endpoint: !Ref DomainAdminEmail

Protocol: email

TopicArn: !Ref Topic

#

# CloudWatch Alarms

#

StatusYellowAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Replica shards for at least one index are not allocated to nodes in a cluster.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'ClusterStatus.yellow'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 60

Statistic: Maximum

Threshold: 1

StatusRedAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Primary and replica shards of at least one index are not allocated to nodes in a cluster.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'ClusterStatus.red'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 60

Statistic: Maximum

Threshold: 1

CPUUtilizationTooHighAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Average CPU utilization over last 45 minutes too high.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 3

MetricName: 'CPUUtilization'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 900

Statistic: Average

Threshold: 80

MasterCPUUtilizationTooHighAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Average CPU utilization over last 45 minutes too high.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 3

MetricName: 'MasterCPUUtilization'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 900

Statistic: Average

Threshold: 50

FreeStorageSpaceTooLowAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Cluster has less than 2GB of storage space.'

ComparisonOperator: LessThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'FreeStorageSpace'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 60

Statistic: Minimum

Threshold: 2000

IndexWritesBlockedTooHighAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Cluster is blocking incoming write requests.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'ClusterIndexWritesBlocked'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 300

Statistic: Maximum

Threshold: 1

JVMMemoryPressureTooHighAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Average JVM memory pressure over last 15 minutes too high.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'JVMMemoryPressure'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 900

Statistic: Average

Threshold: 80

MasterJVMMemoryPressureTooHighAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Average JVM memory pressure over last 15 minutes too high.'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'MasterJVMMemoryPressure'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 900

Statistic: Average

Threshold: 50

MasterNotReachableFromNodeAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'Master node stopped or not reachable. Usually the result of a network connectivity issue or AWS dependency problem.'

ComparisonOperator: LessThanThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'MasterReachableFromNode'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 60

Statistic: Minimum

Threshold: 1

AutomatedSnapshotFailureTooHighAlarm:

DependsOn: TopicEndpointSubscription

Type: 'AWS::CloudWatch::Alarm'

Properties:

AlarmActions:

- !Ref Topic

AlarmDescription: 'No automated snapshot was taken for the domain in the previous 36 hours (created by marbot).'

ComparisonOperator: GreaterThanOrEqualToThreshold

Dimensions:

- Name: ClientId

Value: !Ref 'AWS::AccountId'

- Name: DomainName

Value: !Ref DOMAINNAME

EvaluationPeriods: 1

MetricName: 'AutomatedSnapshotFailure'

Namespace: 'AWS/ES'

OKActions:

- !Ref Topic

Period: 60

Statistic: Maximum

Threshold: 1

#

# Log Streamer and Demo resources

# [LogStreamerRole, LogStreamer, LogStreamerInvokePermission, DemoStack]

#

LogStreamerRole:

Type: 'AWS::IAM::Role'

Properties:

AssumeRolePolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Principal:

Service: lambda.amazonaws.com

Action: 'sts:AssumeRole'

Path: /

Policies:

- PolicyName: !Sub logstreamer-${AWS::Region}

PolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Action:

- logs:CreateLogGroup

- logs:CreateLogStream

- logs:PutLogEvents

Resource: !Sub arn:aws:logs:${AWS::Region}:${AWS::AccountId}:log-group:/aws/lambda/\*

- Effect: Allow

Action:

- es:ESHttpPost

Resource: !Sub arn:aws:es:${AWS::Region}:${AWS::AccountId}:domain/\*

# - Effect: Allow

# Action:

# - sts:AssumeRole

# Resource: !Sub ${LoggingMasterRole.Arn}

LogStreamer:

Type: AWS::Lambda::Function

Properties:

Description: Centralized Logging - Lambda function to stream logs on ES Domain

Environment:

Variables:

# V56536055 - 10/08/2018 - better logging capabilities

LOG\_LEVEL: 'INFO' #change to WARN, ERROR or DEBUG as needed

DomainEndpoint: !Sub ${ElasticsearchAWSLogs.DomainEndpoint}

MasterRole: !Sub ${LoggingMasterRole}

SessionId: !Sub ${AWS::AccountId}-${AWS::Region}

Owner: Hub

Solution: SO0009

ClusterSize: !Ref ClusterSize

UUID: !Sub ${CreateUniqueID.UUID}

AnonymousData: !FindInMap [InstanceMap, send-data, SendAnonymousData]

Handler: index.handler

Role: !Sub ${LogStreamerRole.Arn}

Code:

S3Bucket: !Join ["-", [!FindInMap ["SourceCode", "General", "S3Bucket"], Ref: "AWS::Region"]]

S3Key: !Join ["/", [!FindInMap ["SourceCode", "General", "KeyPrefix"], "clog-indexing-service.zip"]]

Runtime: nodejs8.10

Timeout: 300

LogStreamerInvokePermission:

Type: AWS::Lambda::Permission

Properties:

FunctionName: !Sub ${LogStreamer}

Action: lambda:InvokeFunction

Principal: !Sub logs.${AWS::Region}.amazonaws.com

SourceAccount: !Sub ${AWS::AccountId}

DemoStack:

Type: AWS::CloudFormation::Stack

Condition: DemoData

Properties:

Parameters:

LogStreamerArn: !Sub ${LogStreamer.Arn}

DemoVPCCidr: !Sub ${DemoVPC}

DemoSubnet: !Sub ${DemoSubnet}

TemplateURL: !Join ["/", ["https://s3.amazonaws.com/solutions-reference", !FindInMap ["SourceCode", "General", "KeyPrefix"], "centralized-logging-demo.template"]]

#

# Solution Helper resources

# [SolutionHelperRole, SolutionHelper, SendingAnonymousData, CreateUniqueID]

#

SolutionHelperRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Principal:

Service: lambda.amazonaws.com

Action: sts:AssumeRole

Path: /

Policies:

- PolicyName: Custom\_Lambda\_Loader\_Permissions

PolicyDocument:

Version: 2012-10-17

Statement:

- Effect: Allow

Action:

- logs:CreateLogGroup

- logs:CreateLogStream

- logs:PutLogEvents

Resource: !Sub arn:aws:logs:${AWS::Region}:${AWS::AccountId}:log-group:/aws/lambda/\*

- Effect: Allow

Action:

- ec2:DescribeImages

Resource: "\*"

SolutionHelper:

Type: AWS::Lambda::Function

Properties:

Handler: solution-helper.lambda\_handler

Role: !Sub ${SolutionHelperRole.Arn}

Description: EFS Backup - This function is a CloudFormation custom lambda resource that generates UUID for each deployment.

Code:

S3Bucket: !Sub solutions-${AWS::Region}

S3Key: library/solution-helper/v3/solution-helper.zip

Runtime: python2.7

Timeout: 300

CreateUniqueID:

Type: Custom::LoadLambda

Properties:

ServiceToken: !Sub ${SolutionHelper.Arn}

Region: !Sub ${AWS::Region}

CreateUniqueID: true

Outputs:

DomainEndpoint:

Description: ES domain endpoint URL

Value: !Sub https://${ElasticsearchAWSLogs.DomainEndpoint}

KibanaLoginURL:

Description: Kibana login URL

Value: !Sub https://${ElasticsearchAWSLogs.DomainEndpoint}/\_plugin/kibana/

MasterRole:

Description: IAM role for ES cross account access

Value: !Sub ${LoggingMasterRole.Arn}

SpokeAccountIds:

Description: Accounts that are allowed to index on ES

Value: !Join [ ',', !Ref SpokeAccounts]

LambdaArn:

Description: Lambda function to index logs on ES Domain

Value: !Sub ${LogStreamer.Arn}

ClusterSize:

Description: Cluster size for the deployed ES Domain

Value: !Sub ${ClusterSize}