Using Service Catalog as a Preventive Control

May 2019

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

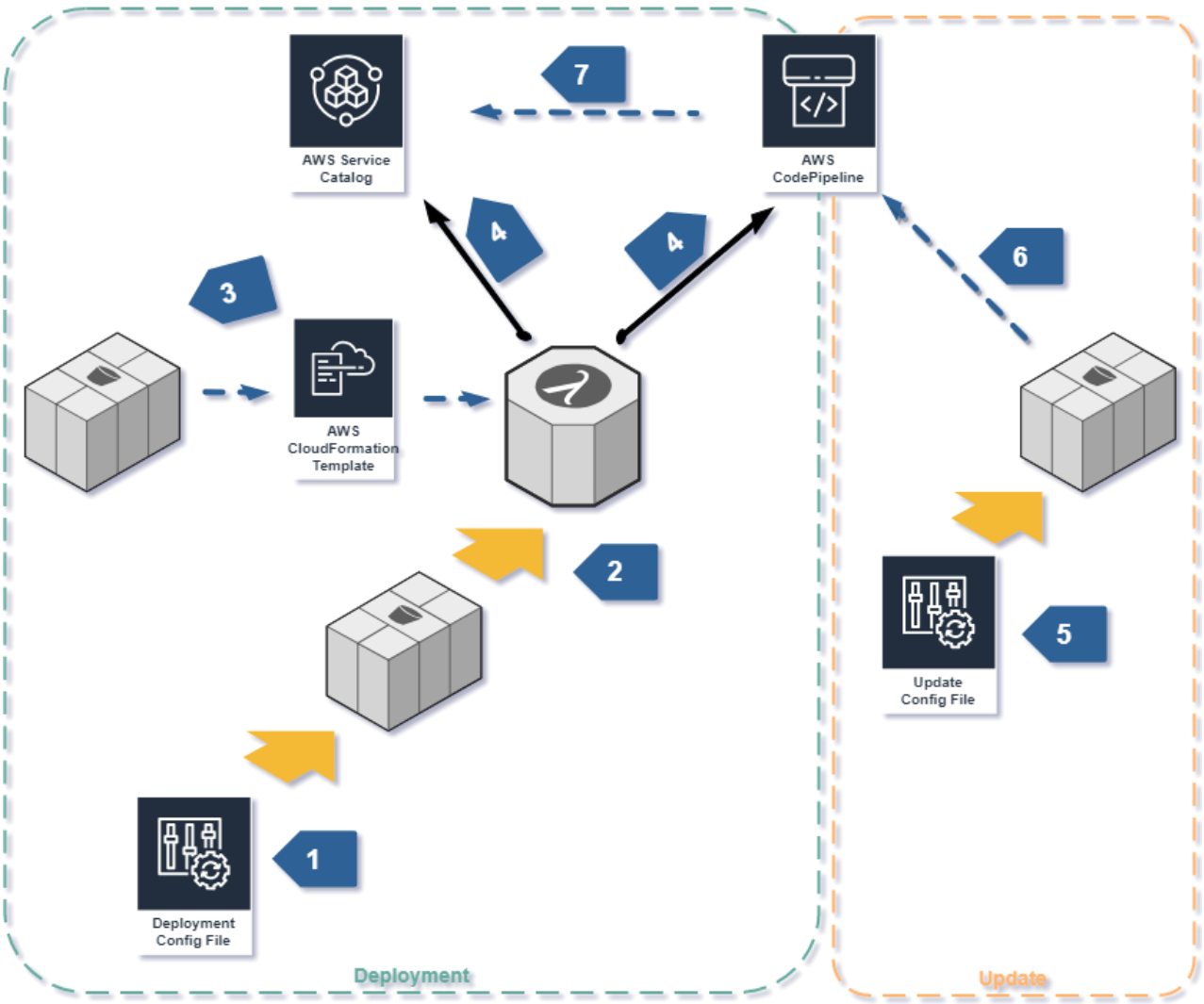
Contents

Solution Overview

Introduction

Large enterprises try to find a balance between controlling risk and empowering their developers in alignment with DevOps practices. Ideally developers are able to leverage AWS services and create optimized architectures for their applications. This solution addresses the risk and empowerment concerns by using AWS Service Catalog to exposed hardened AWS services to developers. By leveraging AWS Service Catalog products for each AWS Service; developers can create their own architectures with a self-service experience. This solution provides AWS CloudFormation templates to make it easier for developers to automate the provisioning of the AWS Service Catalog products.

Process Flow



The above diagram illustrates deploying and updating Service Catalog product.

Deployment Process

1. User uploads a configuration file to deployment Amazon S3 bucket
2. This upload triggers the AWS Lambda Deployment function
3. The Deployment function pulls the CFN product deployment template from an Amazon S3 bucket and executes it
4. AWS CloudFormation, based on the configuration file, deploys the product to AWS Service Catalog and creates a product update AWS CodePipeline

For more details about the deployment process go to: Product Deployment Lambda

Update

1. User uploads an updated configuration file to an Amazon S3 bucket
2. Upload triggers an AWS CodePipeline
3. The pipeline executes and updates AWS Service Catalog with the new product version

For more details about update process go to: Product Update CodePipeline

Solution Deployment

Deployment Content

Below are the descriptions of the content in each solution folder:

deployment-lambda – source code of the AWS Lambda Function that handles the initial deployment of a Service Catalog product. For more information go to Product Deployment Lambda.

docs – documentation for this solution.

init – AWS CloudFormation templates to create the required foundational infrastructure for the solution such as; IAM policies and roles, the AWS Service Catalog portfolio and an Amazon S3 bucket.

- deploy.sh – located in Init folder. Initial deployment shell script.
- cleanup.sh – located in Init folder. This script deletes all resources created by the deployment script

products-config – (empty) – placeholder for product configuration files. The deployment script will copy the deployment templates from the templates\deployment folder to this folder and update it based on the configuration set in the deployment script.

product-selector-lambda – source code of the AWS Lambda Function to support provisioning products. For more information go to Product Selector Lambda.

resource-compliance-lambda – source code of the AWS Lambda Function to validate parameters when provisioning products. For more information go to Resource Compliance Lambda.

resource-selector-lambda – source code of the AWS Lambda Function to support deployment by easily finding AWS resource such as vpc, subnet, security group, and other using tags and filters. For more information go to Resource Selector Lambda.

s3-upload-files – contains the AWS CloudFormation products deployment template and the AWS CloudFormation products template. The entire content of this folder will be copied to the deployment Amazon S3 bucket during initial solution deployment.

templates – various configuration and AWS CloudFormation templates:

- Deployment – product deployment configuration templates. See products-config folder description above.
- Examples – example of the AWS CloudFormation templates to provision each product from AWS Service Catalog.
- Updates – product update configuration files. For more information about update go to Product Update CodePipeline.
- sc-(product)-update.json – product update configuration template.
- deny-policy.yml – the AWS CloudFormation template to create deny IAM policy. This policy can be attached to IAM users or roles to prevent users create AWS resources that are supported by AWS Service Catalog, from AWS Management Console, cli, api, etc.

Deployment Script

The **deploy.sh** script located in the init folder is used to create the initial infrastructure for the described solution.

This script creates following AWS resources:

- Product Update AWS CodePipeline IAM role
- Product Deployment AWS Lambda IAM role
- Product Selector AWS Lambda IAM role
- AWS Service Catalog Product IAM policy
- AWS Service Catalog Portfolio
- Deployment Amazon S3 Bucket
- Product Deployment AWS Lambda function
- Product Selector AWS Lambda function
- Resource Selector AWS Lambda function
- Deploy products example to AWS Service Catalog

Prerequisites

Before running the deployment script ensure that the following prerequisites have been satisfied.

1. The script assumes you have valid AWS credentials. Make sure you have configured your workstation with valid AWS CLI credentials. For more information visit: AWS CLI quick configuration documentation
2. Open the deployment script in your favorite editor and make changes in the "Initial Deployment Configuration Section". Below is the list of configuration parameters that you should review and change. The default values are displayed in parenthesis below.
    - resources_cfn_stack_name ("sc-product-resources") – name of the AWS CloudFormation stack. This value will be use to name the AWS CloudFormation resource stack
    - deployment_lambda_function_name ("sc-product-deployment-lambda") – name of the product deployment AWS Lambda function
    - deployment_lambda_role_name ("sc-product-deployment-lambda-role") – name of the product deployment AWS Lambda IAM role
    - deployment_s3_bucket_name – name of the deployment S3 bucket
    - product_selector_lambda_role_name ("sc-product-selector-lambda-role") – name of the product selector AWS Lambda IAM role
    - pipeline_role_name ("sc-product-update-codepipeline-role") – name of the product update AWS CodePipeline IAM role
    - sc_product_policy_name ("service-catalog-product-policy") – name of the AWS Service Catalog IAM policy
    - sc_portfolio_description ("Security Product Allow Deploy by Developers") – description of the AWS Service Catalog portfolio

- sc_portfolio_name ("security-products") – name of the AWS Service Catalog portfolio
- account_access_role_name – name of an existing IAM role that will have full access to AWS Service Catalog portfolio, for instance: admin.
- deployer_config_file_suffix ("deployer") – suffix of the product deployment configuration files. If you decide to change it, you will need to change extension of all files in the templates\deployment folder.

Deployment

To deploy the solution, download this zip file from GitHub and unzip it on your computer (if you are familiar with git; you can also clone this repo). From a terminal session; change directory to the init subfolder where you unzipped the solution. Next run the deployment script from the init folder.

```
sh ./deploy.sh
```

Cleanup

To delete all AWS resources created by deployment script you can run the cleanup script from the init folder:

```
sh ./cleanup.sh
```

Product Deployment Lambda

"sc-product-deployment-lambda" is the AWS Lambda function responsible for adding new products to AWS Service Catalog as well as creating the product's IAM Service Catalog role.

To add a new product to AWS Service Catalog, you will need to upload a product configuration file to the Amazon S3 deployment bucket. The configuration file has to have the extension specified in the solution deployment script. The default extension is ".deployer".

Configuration File Format

The configuration file's content has to be in JSON readable format. Most editors have a json editing tool that can be used to validate json syntax. Below is an example of the configuration file.

```
{
  "Parameters": {
    "PorfolioStack": "<name of the CloudFormation stack used to create Service Catalog portfolio>",
    "ProductName": "<product name>",
    "ProductDescription": "<product description>",
    "ProductVersion": "<product initial version e.g. 1.0>",
    "ProductVersionDescription": "<product version description>",
    "ProductTemplateUrl": "<path to product CloudFormation template, including S3 bucket name>",
    "ProductRoleName": "<name of the Service Catalog product IAM role>",
    "ProductPolicyName": "<name of the product IAM policy",
    "ProductRoleTemplateUrl": "<path to product IAM role Cloudformation template, including S3 bucket
name>",
    "TemplateRuleConstraint": "<optional, Service Catalog template rule constraint>"
    "DeploymentBucket": "<deployment bucket>",
    "DeployUpdatePipeline": "< true/false>",
    "UpdateConfigFileName": "<name of the file to trigger update pipeline without extension>"
  }
}
```

The values for the parameters: PorfolioStack, ProductPolicyName and DeploymentBucket should be the same values provided in the solution deployment script.

Copies of the configuration files can be found in products-config folder after running the solution deployment script. This can be used to validate your current configuration. For example, the value of "TemplateRuleConstraint" can be found in the products-config\sc-product-elasticsearch.deployer file.

Deployment

We are organizing all of the product assets within the same S3 prefix. As an example, <s3-deployment-bucket>/products/. Each product will have its own prefix as well. For instance, < s3-deployment-bucket>/products/sqs. To deploy a new product, follow these steps:

1. Upload the IAM Role and product CFN templates to the Amazon S3 bucket (<s3-deployment-bucket>) using the S3 prefix <products/productname>.
2. Create a configuration file pointing to the location where templates where uploaded
3. Upload the configuration file to <s3-deployment-bucket>/deployment-cfg folder

The upload to S3 will trigger the sc-product-deployment-lambda function. This function will launch the product deployment CloudFormation template located at <s3-deployment-bucket>/deployment-cfn/sc-product-deployment.yml.

AWS CloudFormation parameters value will be read from the configuration file.

Product Update CodePipeline

When a new product is created, there is an option to create an AWS Codepipeline to manage product updates. This behavior is set by changing the "DeployUpdatePipeline" value to true in the configuration file. See Product Deployment Lambda. Here are some important things to know about this feature:

- Each product pipeline will be triggered by watching a specific S3 path and looking for a newly uploaded zipped configuration file.
- The S3 path is <s3-deployment-bucket>/deployment-cfg/<UpdateConfigFileName>.zip
    - The value for <UpdateConfigFileName> is provided in the deployment configuration file under the "UpdateConfigFileName" parameter.
- The content of zipped configuration file indicated above should contain the following files.
    - A product AWS CloudFormation template: this is the updated version of the AWS Service Catalog product
    - An update configuration file ${UpdateConfigFileName}.json.
        - The value for ${UpdateConfigFileName} is provided in the deployment configuration file under "UpdateConfigFileName" parameter.

Update configuration file has to be in valid JSON format and should contain these keys with values.

```
{
  "SchemaVersion": "1.0",
  "ProductVersionName": "<product version>",
  "ProductVersionDescription": "<product version description>",
  "ProductType": "CLOUD_FORMATION_TEMPLATE",
  "Properties": {
    "TemplateFilePath": "/<name of product template that is included in zip>"
  }
}
```

Deploy Product Update

1. Create a zip file containing both the product template and a configuration file. Be sure to follow the original guidance above
2. Upload zip file to the <s3-deployment-bucket>/deployment-cfg folder
3. If update zip file already exists in this location, overwrite it

Product Selector Lambda

Product Selector is an AWS Lambda Function that was designed to be called from within an AWS CloudFormation as a custom resource.

AWS Service Catalog Product Identification

Return resource ids of product and artifact (version) required to launch an AWS Service Catalog product from AWS CloudFormation. This improves the end user experience because they do not need to remember specific resource identifiers.

Example Syntax from with CloudFormation:

```
ProductSelector:
  Type: "Custom::ProductSelector"
  Version: "1.0"
  Properties:
    ServiceToken: !Sub 'arn:aws:lambda:${AWS::Region}:${AWS::AccountId}:function:sc-product-selector'
    ProductName: <product name e.g. sqs>
    Version: <version of product to return id>
```

Note: Version parameter is optional. If not provided, latest version will be return.

Example Syntax of obtaining the Returned Values:

```
Product Id: !GetAtt ProductSelector.ProductId
Provisioning Artifact Id: !GetAtt ProductSelector.ArtifactId
```

Example of Usage:
Please refer to the AWS CloudFormation product provision templates in the templates\examples folder.

Resource Compliance Lambda

AWS CloudFormation custom resources provide the ability to run custom logic during a CFN template's execution. This feature can allow us to perform additional compliances checks or configuration steps on resources we are creating. The Resource Compliance Lambda is an example of this and provides the following capabilities:

Bring You Own Key (BYOK)

This capability validates if the provided KMS key has external key material using the KMS Import Key feature.

Syntax:

```
ProductSelector:
 Type: "Custom:: ResourceCompliance"
 Version: "1.0"
 Properties:
  ServiceToken: !Sub 'arn:aws:lambda:${AWS::Region}:${AWS::AccountId}:function: sc-resource-compliance'
  Action:
    Name: byok
    Parameters:
        Key: <KMS Key Id to validate>
```

Return:

If provided KMS Key doesn't have EXTERNAL origin, this indicated it is not an imported key. The function will return a FAILURE status back to AWS CloudFormation. This will cause the stack to fail.

JSON

This capability convert provided string to JSON object. The input string has following format:

"Key1=Value1,Key2=Value2,…"

Example:

"Name=My Cluster,Environment=Dev"

The input string can be convert to four different output formats which are:

Tags:

Tags type convert input string to following format:

[{"Key":Key1, "Value": Valu1},{"Key":Key2, "Value",Value2},…]

Example:

[{"Key":"Name", "Value": "My Cluster"},{"Key":"Environment", "Value","Dev"}]

Converted string is return as JSON object back to CFN templates where can be use as value for any resource Tags parameter. For usage example see CFN templates in templates/examples/ location

DynamoDBSchema:

This type convert input string to Amazon DynamoDB Attribute Definition format:

[{"AttributeName":Key1,"AttributeType":Valu1},{"AttributeName":Key2, "AttributeType",Value2},….]

Example:

[{"AttributeName":"Name", "AttributeType": "S"},{" AttributeName":"Id", "AttributeType","S"}]

Converted string is return as JSON object back to CFN where can be apply to Amazon DynamoDB product. For usage example see CFN template in templates/examples/ sc-provision-dynamodb-cft.yml

For supported AttributeType value visit:
https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_AttributeDefinition.html

DynamoDBKey

This type convert input string to Amazon DynamoDB Key Schema format:

[{"AttributeName":Key1, "KeyType": Valu1},{" AttributeName":Key2, "KeyType",Value2},…]

Example:

[{"AttributeName":"Name", "KeyType": "HASH"},{" AttributeName":"Id", "KeyType","RANGE"}]

Converted string is return as JSON object back to CFN where can be apply to Amazon DynamoDB product. For usage example see CFN template in templates/examples/ sc-provision-dynamodb-cft.yml

For supported "KeyType" value visit:
https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_KeySchemaElement.html

SQS

This type convert input string to tags and apply them to Amazon SQS, as currently AWS CloudFormation does not support Amazon SQS tagging.

{"Key":Key1, "Value": Valu1},{"Key":Key2, "Value",Value2},…

No JSON return
Note: the SQS queue URL must be provided under SQS parameter
For usage example see AWS Service Catalog SQS product template:
s3-upload-files/products/sqs/sc-sqs.yml

Syntax:

```
ResourceCompliance:
 Type: "Custom:: ResourceCompliance"
 Version: "1.0"
 Properties:
  ServiceToken: !Sub 'arn:aws:lambda:${AWS::Region}:${AWS::AccountId}:function: sc-resource-compliance'
  Action:
    Name: json
    Parameters:
     JSON: '<comma delimiter key=value string>'
     Type: <Convert type: tags, sqs, dynamodbschema, dynamodbkey>
     SQS: <SQS Queue URL. Require when Type is sqs>
```

Return:
This capability return JSON object, except when type is 'sqs'.

Example of Usage
See links under each convert type

Principal

This capability ensures that the provided principal for the access policy is not a wildcard (*). If a wildcard value is passed, the wildcard will be replaced with AWS account Id of the account.

Syntax:

```
ProductSelector:
 Type: "Custom:: ResourceCompliance"
 Version: "1.0"
 Properties:
  ServiceToken: !Sub 'arn:aws:lambda:${AWS::Region}:${AWS::AccountId}:function: sc-resource-compliance'
  Action:
    Name: principal
    Parameters:
     Account: <AWS Account Id>
     Principal: <List of principals>
```

Return:
This capability does not return any values.

Example of Usage
See sc-elasticsearch.yml cfn template in s3-upload-files\products\elasticsearch location

Resource Selector Lambda

One of the considerations when building this solution was scale. How much work is required from the team that administers the AWS Service Catalog portfolio and the development teams that uses the AWS Service Catalog products? In order to truly make this solution self-service we needed to provide a method for development teams to easily obtain AWS account specific resource ids that are required by the AWS Service Catalog products. The Resource Selector Lambda provides this service. The Resource Selector Lambda is called as a customer resource in your AWS CloudFormation template. The Lambda function will return the AWS account specific values that are dependent on the parameters you pass into the function from the custom resource definition.

Resource Selector AWS Lambda function supports returning the following resource ids:

- VPC
- Subnets
- Security Groups
- AWS Certificate Manager
- KMS Keys
- IAM Policy
- IAM Roles
- Spot Price
- Image Id (AMI)

Usage

Below is an AWS CloudFormation snippet with demonstrates the syntax and options supported by each resource:

```
ResourceSelector:
  Type: "Custom::ResourceSelector"
  Version: "1.0"
  Properties:
   ServiceToken: !Sub 'arn:aws:lambda:${AWS::Region}:${AWS::AccountId}:function:sc-resource-selector'
   Options:
    OnError : [optional] (skip,failed => default failed)
   Resources:
    vpc:
     Tags:
      - Key: <key name>
        Value: <value>
     Options:
      Output: [optional](single, all => default: all)
      Match: [optional](all, any => default: any)
      OnError: [optiona] (skip,failed => default: failed)
    subnet:
     Tags:
      - Key: <key name>
```

```
    Value: <value>
  Options:
   Output: [optional](single, all => default: all)
   Match: [optional](all, any => default: any)
   OnError: [optiona] (skip,failed => default: failed)
   AvailableIP: [optional] (number => default: 5)
sg:
  Tags:
   - Key: <key name>
     Value: <value>
  Options:
   Output: [optional](single, all => default: all)
   Match: [optional](all, any => default: any)
   OnError: [optiona] (skip,failed => default: failed)
   GroupName: [optional] <security group name>

acm:
  Tags:
   - Key: <key name>
     Value: <value>
  Options:
   Output: [optional](single, all => default: all)
   Match: [optional](all, any => default: any)
   OnError: [optiona] (skip,failed => default: failed)
   Domain: [optional] <certificate domain name>
kms:
  Options:
   Output: [optional](single, all => default: all)
   OnError: [optiona] (skip,failed => default: failed)
   KMSAlias: [optional] <kms key alias>
   KMSOutput: [optional] (id, alias => default: id)
policy:
  Options:
   Output: [optional](single, all => default: all)
   OnError: [optiona] (skip,failed => default: failed)
   PolicyName: [optional] <iam policy name>
role:
  Tags:
   - Key: <key name>
     Value: <value>
  Options:
   Output: [optional](single, all => default: all)
   Match: [optional](all, any => default: any)
   OnError: [optiona] (skip,failed => default: failed)
   RoleName: [optional] <iam role name>
   RolePath: [optional] <iam role path>
spot:
  Options:
   InstanceType: [require] <EC2 Instance Type>
   InstanceOS: [require] (Linux, Windows, RHEL)
ami:
  Tags:
   - Key: <key name>
     Value: <value>
```

Options:
ImageOwner: [require](self,<account id>, amazon, etc.)
ImageName: [optiona] < AMI image name>

Parameters

All parameters are optional. If a parameter is not passed into the function, all resources will be returned.

Tags

Allow a developer to search for resources by tags. This is a list; so you can use a single or multiple tags in the search.
Key – is the name of tag key. This value must match tag name associate with resource (is not case sensitive)
Value – is the value of tag key. It can be a whole name, partial name or regular expression (this is case sensitive)
Note: KMS and IAM Policy do not support search by tags

Output

Specify how many resources should be returned, if more than one resource id matches the search criteria.
Allowed values:

single –returns the first resource that matches your criteria
all – returns all resources matching the criteria

If this value is not provided, all resources matching the criteria will be returned

Match

If multiple tags are provided, indicate if resource need to match on all or any tags
Allow values: all, any
If this value is not provided, all resources that match at least one tag will be returned

OnError

Define how the AWS CloudFormation stack will behave if no resources match our criteria.
Allow values:

skip – CFN stack will ignore empty value and keep continue running. Be aware that this might cause the CFN stack to fail if template was not design to handle an empty value
failed – CFN stack will fail if resources are not found

If this value is not provided, the default behavior is to cause the stack to fail.

This option can be described in two places. If it is described as a property of the function definition, then this value is applied globally to all subsequent resources. If it is described within a specific resource, this value only applies to that resource.

AvailableIP

This parameter allows a developer to specify how many IPs have to be available in the subnet for the function to return it. The default is 5 IPs.

Domain

Name of domain on the ACM certificate. The search domain can be provided as the whole or partial name

KMSAlias

KMS alias to search. The search alias can be provided as a whole or partial name.

KMSOutput

Define if the function should return the found key(s) as an alias or a KMS id. Allow values: id, alias. Default value is alias.

PolicyName

A full or partial name of IAM policy to search for. Keep in mind; IAM policy do not support Tags

RoleName

The full or partial name of the IAM role to search

RolePath

IAM role path

InstanceType

Appling to Spot Pricing - EC2 Instance Type for which the spot price should be return. Example: t2.micro

InstanceOS

Appling to Spot Pricing – EC2 Instance operation system for which the spot price should be return. Allow values: Linux, Windows, RHEL

ImageOwner

Applying for AMI – the owner of image. Can be self, account id, amazon, microsoft, aws-marketpalce

ImageName

Applying for AMI – full or partial name of AMI image

Note: To narrow search of subnets or security group to specific VPC, define criteria for vpc resource as well. If VPC is not found, all subnets/security group will be return. To avoid this, set OnError = failed option under vpc resource.

Summary

In the solution we have demonstrated how to leverage AWS Service Catalog to provide developers with a secure and self-service solution to deploy AWS resources. We have also discussed how to manage the lifecycle of this solution. This includes adding and updating the product catalog as well as deploying these changes into an AWS account(s). This solution can be deployed and ready for use in matter of minutes. The appendix A includes documentation for each product that can be easily imported into your company's internal confluence/wiki/help portal.

Authors

Remek Hetman

Remek is a Senior Cloud Infrastructure Architect with AWS Professional Services Financial Services Practice. He works with AWS financial enterprise customers providing technical guidance and assistance for Infrastructure, Security, DevOps, and Big Data to help them make the best use of AWS services. Outside of work, he enjoys spending time actively, and pursuing his passion – astronomy.

Jim Long

Jim Long is a Sr. Cloud / IT Transformation Architect in the AWS Professional Services Financial Services Practice based out of Boston Massachusetts. He works with large enterprise customers to accelerate their Cloud adoption journey.

References

https://aws.amazon.com/servicecatalog/

https://aws.amazon.com/blogs/mt/how-to-launch-secure-and-governed-aws-resources-with-aws-cloudformation-and-aws-service-catalog/

https://aws.amazon.com/blogs/mt/automate-account-creation-and-resource-provisioning-using-aws-service-catalog-aws-organizations-and-aws-lambda/

https://aws.amazon.com/blogs/mt/create-an-approval-workflow-for-aws-service-catalog-in-servicenow/

https://aws.amazon.com/blogs/mt/create-a-security-partition-for-your-applications-using-aws-service-catalog-and-aws-lambda/

https://aws.amazon.com/blogs/mt/secure-serverless-development-using-aws-service-catalog/

https://aws.amazon.com/blogs/mt/how-to-set-up-a-multi-region-multi-account-catalog-of-company-standard-aws-service-catalog-products/

Appendix A

Provision Products from Service Catalog Developers Guideline

SQS Service Catalog Product

Introduction

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware, and empowers developers to focus on differentiating work.

Product Link

https://aws.amazon.com/sqs/

Provision Product Name

Name of the product when calling product selector lambda: sqs

Product IAM Role

SQS product will be launch under following IAM role: **sc-sqs-product-role**
Make sure to grant permission for that role to access the KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt message body using KMS Key

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **QueueName**: A name for the queue. To create a FIFO queue, the name of your FIFO queue must end with the ".fifo" suffix
- **FifoQueue**: Indicates whether this queue is a FIFO queue. Allowed values:
    - true
    - false
- **ContentBasedDeduplication**: For first-in-first-out (FIFO) queues, specifies whether to enable content-based deduplication. During the deduplication interval, Amazon SQS treats messages that are sent with identical content as duplicates and delivers only one copy of the message. Allow values:
    - true
    - false
- **MessageRetentionPeriod**: The number of seconds that Amazon SQS retains a message. (default: 345600 seconds (4 days); min: 60 seconds (1 minute); max: 1209600 seconds (14 days))
- **DelaySeconds**: The time in seconds that the delivery of all messages in the queue is delayed. (Range 0-900s; default 0)

- **ReceiveMessageWaitTimeSeconds**: Specifies the duration, in seconds, that the ReceiveMessage action call waits until a message is in the queue in order to include it in the response, as opposed to returning an empty response if a message isn't yet available. (Range:0-20 s)
- **VisibilityTimeout**: The length of time during which a message will be unavailable after a message is delivered from the queue. This blocks other components from receiving the same message and gives the initial component time to process and delete the message from the queue. (Range: 0 - 43200 seconds (12 hours))
- **KMSId**: The id of encryption key that should be used to encrypt the SQS.
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the SQS product. You can use these values within your CloudFormation template using ImportValue

- **QueueFifoArn**: The Amazon Resource Name (arn) for the SQS (FIFO queue)
- **QueueFifoURL**: The DNS name for the SQS's HTTPS endpoint. (FIFO queue)
- **QueueStdArn**: The Amazon Resource Name (arn) for the SQS (Standard queue)
- **QueueStdURL**: The DNS name for the SQS's HTTPS endpoint. (Standard queue)

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-sqs-cft.yml

SNS Service Catalog Product

Introduction

Amazon Simple Notification Service (SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging.

Product Link

https://aws.amazon.com/sns/

Provision Product Name

Name of the product when calling product selector lambda: sns

Product IAM Role

SNS product will be launch under following IAM role: **sc-sns-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt SNS message using KSM Key
- Prevent open SNS to public access

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **TopicName**: The name of the SNS topic. This value will be used for Display Name as well
- **KMSId**: The id of encryption key that should be used to encrypt the SNS.
- **PolicyPrincipal** (optional): along with PolicyAction allow overwrite default SNS policy. Multiple principals can be enter as comma separated. Public open "*" is not allow and will scope access to root account.
- **PolicyAction** (optional): along with PolicyAction allow overwrite default SNS policy. Multiple actions cab enter as comma separated.

List of Outputs

The following values are outputs of the SNS product. You can use these values within your CloudFormation template using ImportValue

- **SNSArn**: ARN of SNS topic

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-sns-cft.yml

Kinesis Service Catalog Product

Introduction

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information. Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application.

Product Link

https://aws.amazon.com/kinesis/

Provision Product Name

Name of the product when calling product selector lambda: kinesis

Product IAM Role

Kinesis product will be launch under following IAM role: **sc-kinesis-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt Kinesis Stream using KMS Key

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **Name**: The name of the Kinesis stream
- **RetentionPeriodHours**: The number of hours for the data records that are stored in shards to remain accessible. (Range: 24-168; Default: 24)
- **ShardCount**: The number of shards that the stream uses. (Range: 1 - 30; Default 1)
- **KMSId**: The id of encryption key that should be used to encrypt the Kinesis Stream data.
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the Kinesis product. You can use these values within your CloudFormation template using ImportValue

- **KinesisArn**: The Amazon Resource Name (arn) for the Kinesis Stream
- **KinesisId**: The Kinesis stream name (physical ID).

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-kinesis-cft.yml

ElasticSearch Service Catalog Product

Introduction

The Amazon Elasticsearch Service (Amazon ES) is a managed service that makes it easy to create a domain and deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. Elasticsearch is a popular open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and clickstream analytics.

Product Link

https://aws.amazon.com/elasticsearch-service/

Provision Product Name

Name of the product when calling product selector lambda: elasticsearch

Product IAM Role

ElasticSearch product will be launch under following IAM role: **sc-elasticsearch-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt store data using KMS Key
- Prevent policy to be open for public access to ElasticSearch
- Deploy inside VPC

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **DomainName**: The name for the Amazon Elasticsearch domain. This should adhere to AM Tech naming standards for Elastisearch
- **ElasticsearchVersion**: The version of Elasticsearch (Default is 6.2)
- **InstanceType**: The instance type for your data nodes. Be aware that some Elasticsearch instance types do not support KMS encryption. Allowed values:
    - m4.large.elasticsearch
    - m4.xlarge.elasticsearch
    - r4.large.elasticsearch
    - r4.xlarge.elasticsearch
- **InstanceCount**: The number of data nodes (instances) to use in the Amazon ES domain. (default 1; max 4)
- **DedicatedMasterType**: The hardware configuration of the instance that hosts the dedicated master node. Allow values:
    - m4.large.elasticsearch
    - m4.xlarge.elasticsearch

    - r4.large.elasticsearch
    - r4.xlarge.elasticsearch
- **DedicatedMasterCount**: the number of instances to use for the master node. (default 0; max 2)
    - 0 – means no master node
- **VolumeSize**: The size of the EBS volume for each data node. (Range 10-100GB; default 10)
- **EnableZoneAwareness**: Indicates whether to enable zone awareness for the Amazon ES domain. When enabled the InstanceCount above need to be either 2 or 4
- **ESAccessPrincipalFull**: the IAM role(s) that is allowed to interact with the ES endpoint using HTTPS over port 443 (using the following HTTP methods Get, Post Put, and Delete). Typically, this is the IAM role used by the EC2 instance that is pushing data into the Elasticsearch domain. This value is used to create the Elasticsearch domain's access policy. For multiple roles use comma separation
- **ESAccessPrincipalReadOnly**: Same as above but allowing only HTTP Get method. If you need grand only one (either Full or ReadOnly) permission, for the other enter None as value.
- **KMSId**: The id of encryption key that should be used to encrypt the Elasticsearch domain's data at rest.
- **SubnetIds**: VPC Subnet(s) in which the ES will be launch
- **SecurityGroupIds**: Security group to access ES
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the ElasticSearch product. You can use these values within your CloudFormation template using ImportValue

- **DomainArn**: The Amazon Resource Name (arn) for the Elasticsearch Domain
- **DomainEndpoint**: The DNS name for the Elasticsearch domain's HTTPS endpoint.

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-elasticsearch-cft.yml

ElastiCache Service Catalog Product

Introduction

Amazon ElastiCache offers fully managed Redis. Seamlessly deploy, run, and scale popular open source compatible in-memory data stores. Build data-intensive apps or improve the performance of your existing apps by retrieving data from high throughput and low latency in-memory data stores.

Product Link

https://aws.amazon.com/elasticache/

Provision Product Name

Name of the product when calling product selector lambda: elasticache

Product IAM Role

ElastiCache product will be launch under following IAM role: **sc-elasticache-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encryption at rest and in transit enabled by default

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **ClusterName**: The name for the Amazon ElastiCache cluster. This should adhere to AM Tech naming standards for Elasticache
- **NodeType**: The instance type for your data nodes. Allowed values:
    - cache.t2.micro
    - cache.t2.small
    - cache.t2.medium
    - cache.m4.large
    - cache.m4.xlarge
    - cache.r5.large
    - cache.r5.xlarge
- **NumberClusters**: The number of cache clusters. (default 2; max 6)
- **AuthToken** [Optional]: The password that's used to access a password-protected server
- **SNSTopicArn**: (Optional) - ARN of SNS NotificationTopic
- **CacheSubnetGroupName**: The name of a cache subnet group to use for this replication group.as
- **SecurityGroupIds**: A list of Amazon Virtual Private Cloud (Amazon VPC) security groups to associate with this replication group
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the ElastiCache product. You can use these values within your CloudFormation template using ImportValue

- **ElastiCacheId**: The Id of ElastiCache cluster
- **PrimaryEndPoint**: The DNS name for the Elasticache cluster's HTTPS endpoint.

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-elasticache-cft.yml

EFS Service Catalog Product

Introduction

Amazon Elastic File System (Amazon EFS) provides simple, scalable, elastic file storage for use with AWS Cloud services and on-premises resources. Amazon EFS is built to elastically scale on demand without disrupting applications, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

Product Link

https://aws.amazon.com/efs/

Provision Product Name

Name of the product when calling product selector lambda: efs

Product IAM Role

EFS product will be launch under following IAM role: **sc-efs-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt EFS using KMS Key

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **KMSId**: The id of encryption key that should be used to encrypt the EFS data at rest.
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the EFS product. You can use these values within your CloudFormation template using ImportValue

- **EFSId**: the logical Id of Elastic File System

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-efs-cft.yml

EBS Service Catalog Product

Introduction

Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.

Product Link

https://aws.amazon.com/ebs/

Provision Product Name

Name of the product when calling product selector lambda: ebs

Product IAM Role

EBS product will be launch under following IAM role: **sc-ebs-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt EBS using KMS Key

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **AutoEnableIO**: Indicates whether the volume is auto-enabled for I/O operations. (Default: false). Allowed values:
    - true
    - false
- **AvailabilityZone**: The Availability Zone in which to create the new volume.
- **VolumeSize**: The size of the volume, in gibibytes (GiBs).
- **SnapshotId**: (optional) the snapshot from which to create the new volume.
- **KMSId**: The id of encryption key that should be used to encrypt the EBS data at rest.
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the EBS product. You can use these values within your CloudFormation template using ImportValue

- **EBSId**: The id of EBS volume

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-ebs-cft.yml

DMS Replication Instance Service Catalog Product

Introduction

The AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases.

Product Link

https://aws.amazon.com/dms/

Provision Product Name

Name of the product when calling product selector lambda: dmsinstance

Product IAM Role

DMS Replication Instance product will be launch under following IAM role: **sc-dms-instance-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt DMS using KMS Key

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **ReplicationInstanceIdentifier**: The name for the DSM replication instance. This should adhere to AM Tech naming standards for DMS
- **ReplicationInstanceClass**: The instance type for your DMS. Allowed values:
    - dms.t2.large
    - dms.r4.large
    - dms.r4.xlarge
- **EngineVersion**: The DMS Engine version. Supported version are (default: 3.1.2):
    - 3.1.2
    - 2.4.4
- **Storage**: The amount of storage (in gigabytes) to be initially allocated for the replication instance. (default 50; max 500)
- **KMSId**: The id of encryption key that should be used to encrypt the DMS data at rest.
- **ReplicationSubnetGroupIdentifier**: A subnet group to associate with the replication instance.
- **SecurityGroupsIds**: The VPC security group(s) to be used with the replication instance
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the DMS Replication Instance product. You can use these values within your CloudFormation template using ImportValue

- **ReplicationInstance**: the DMS Replication Instance Id

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-dms-replication-instance-cft.yml

DMS Endpoint Service Catalog Product

Introduction

The AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database. The AWS Database Migration Service can migrate your data to and from most widely used commercial and open-source databases.

Product Link

https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html

Provision Product Name

Name of the product when calling product selector lambda: dmsendpoint

Product IAM Role

DMS Endpoint product will be launch under following IAM role: **sc-dms-endpoint-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt DMS Endpoint using KMS Key

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **EndpointIdentifier**: The name for the DSM endpoint. This should adhere to AM Tech naming standards for DMS
- **EndpointType**: The type of endpoint. Allowed values:
    - source
    - target
- **EngineName**: The source or destination engine. Currently only oracle is supported
- **DatabaseName**: The name of endpoint database (SID)
- **ServerName**: The name of the server where the endpoint database resides.
- **Port**: The port used by the endpoint database
- **UserName**: The user name to be used to log in to the endpoint database.
- **Password**: The password to be used to log in to the endpoint database.
- **KMSId**: The id of encryption key that should be used to encrypt the DMS data at rest.
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the DMS Endpoint product. You can use these values within your CloudFormation template using ImportValue

- Endpoint: the DMS endpoint Id

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-dms-endpoint-cft.yml

DynamoDB Service Catalog Product

Introduction

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and support peaks of more than 20 million requests per second.

Product Link

https://aws.amazon.com/dynamodb/

Provision Product Name

Name of the product when calling product selector lambda: dynamodb

Product IAM Role

DynamoDB product will be launch under following IAM role: **sc-dynamodb-product-role**

Security Features

List of security feature included by default in product:

- Server-side encryption enabled by default

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **TableName**: A name for the table.
- **ReadCapacityUnits**: The desired minimum number of consistent reads. Default 5.
- **WriteCapacityUnits**: The desired minimum number of consistent writes. Default 5.
- **KeySchema**: Specifies the attributes that make up the primary key for the table. For more information check: DynamoDBKey
- **AttributeDefinitions**: A list of attributes that describe the key schema for the table and indexes. For more information check: DynamoDBSchema:
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the DynamoDB product. You can use these values within your CloudFormation template using ImportValue

- **DynamoDB**: the name of DynamoDB table
- **DynamoDBArn**: the ARN of DynamoDB table

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-dynamodb-cft.yml

FSx for Windows Service Catalog Product

Introduction

Amazon FSx for Windows File Server provides a fully managed native Microsoft Windows file system so you can easily move your Windows-based applications that require file storage to AWS. Built on Windows Server, Amazon FSx provides shared file storage with the compatibility and features that your Windows-based applications rely on, including full support for the SMB protocol and Windows NTFS, Active Directory (AD) integration, and Distributed File System (DFS).

Product Link

https://aws.amazon.com/fsx/windows/

Provision Product Name

Name of the product when calling product selector lambda: fsx

Product IAM Role

FSx product will be launch under following IAM role: **sc-fsx-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt FSx using KMS Key
- Deploy inside VPC

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **KMSId**: The id of encryption key that should be used to encrypt the FSx data at rest.
- **StorageCapacity**: The storage capacity (GB) of the file system. (Min: 300, Max: 65536, Default: 300)
- **SubnetIds**: VPC Subnets Ids as comma separated list
- **SecurityGroupIds**: VPC Security Groups Ids as comma separated list
- **ActiveDirectoryId**: The ID for an existing Microsoft Active Directory
- **ThroughputCapacity**: The throughput of an Amazon FSx file system, measured in megabytes per second. (Min: 8, Max: 2048, Default: 8)
- **AutomaticBackupRetentionDays**: The number of days to retain automatic backups. Setting this to 0 disables automatic backups. (Min: 0, Max: 35, Default: 0)
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the FSx product. You can use these values within your CloudFormation template using ImportValue

- **FSxId**: the logical Id of FSx

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-fsx-cft.yml

SageMaker Service Catalog Product

Introduction

Amazon SageMaker provides every developer and data scientist with the ability to build, train, and deploy machine learning models quickly. Amazon SageMaker is a fully-managed service that covers the entire machine learning workflow to label and prepare your data, choose an algorithm, train the model, tune and optimize it for deployment, make predictions, and take action. Your models get to production faster with much less effort and lower cost.

Product Link

https://aws.amazon.com/sagemaker/

Provision Product Name

Name of the product when calling product selector lambda: sagemaker

Product IAM Role

SageMaker product will be launch under following IAM role: **sc-sagemaker-product-role**
Make sure to grant permission for that role to access provided KMS key id used to provision product.

Security Features

List of security feature included by default in product:

- Encrypt SageMaker using KMS Key
- Deploy inside VPC

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **KMSId**: The id of encryption key that should be used to encrypt the SageMaker data at rest.
- **InstanceType**: The type of ML compute instance to launch for the notebook instance
- **NotebookInstanceName**: The name of the new notebook instance.
- **RoleArn**: ARN of SageMaker IAM Role.
- **SubnetId**: The ID of the subnet in a VPC to which you would like to have a connectivity from your ML compute instance
- **SecurityGroupIds**: The VPC Security Groups Ids as comma separated list
- **VolumeSizeInGB**: The size, in GB, of the ML storage volume to attach to the notebook instance. (Min: 5, Max: 16384, Default: 5)
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the SageMaker product. You can use these values within your CloudFormation template using ImportValue

- **NotebookArn**: the ARN of SageMaker Notebook Instance

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/sc-provision-sagemaker-cft.yml

AutoScaling Service Catalog Product

Introduction

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.

Product Link

https://aws.amazon.com/autoscaling/

Provision Product Name

Name of the product when calling product selector lambda: autoscaling

Product IAM Role

AutoScaling product will be launch under following IAM role: **sc-autoscaling-product-role**

Security Features

List of security feature included by default in product:

- Deploy inside VPC
- Working directly with ALB over HTTPS

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **ALBTargetGroupStack**: The ALB Target Group ARN
- **AppInstanceType**: The EC2 instance type
- **AppEC2IAMRole**: The name of EC2 Instance profile
- **AppHealthCheckGracePeriod**: Number of seconds after instance launch ALB begins health checks. (Min: 40, Max: 5000, Default: 300)
- **HealthCheckType**: The service you want the health status from, Amazon EC2 or Elastic Load Balancer. Allow values: ELB, EC2. Default ELB
- **AppMinCount**: Minimum EC2 Instances count. (Min: 0, Max: 5, Default: 1)
- **AppMaxCount**: Maximum EC2 Instances count. (Min: 0, Max: 10, Default: 1)
- **SNSTopicARN**: (optional) SNS Notification Topic ARN
- **SNSTopicStackName**: (optional) Notification Topic Service Catalog SNS Stack
- **EnableScalePolicy**: Enable Auto Scaling Policy. Allow values: true, false. Default false
- **ScaleOutCooldown**: The amount of time, in seconds, after a scale-out activity completes before another scale-out activity can start. Default: 3600
- **ScaleOutAdjustment**: The number of instances by which to scale-out. (Min: 1, Max: 5, Default: 1)
- **ScaleInCooldown**: The amount of time, in seconds, after a scale-in activity completes before another scale in activity can start. Default: 1200
- **ScaleInAdjustment**: The number of instances by which to scale-in. (Min: -1, Max: -5, Default: -1)

- **ImageId**: AMI Image Id
- **SecurityGroupIds**: the VPC Security Groups Ids as comma separated list
- **SubnetIds**: the VPC Subnets Ids as comma separated list
- **UserData**: (optional) User Data
- **KeyName**: (optional) Key pair name
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the AutoScaling product. You can use these values within your CloudFormation template using ImportValue

- **AppASGName**: the name of AutoScaling group
- **ScaleOutPolicy**: the ARN of Scale Out policy
- **ScaleInPolicy**: the ARN of Scale In policy

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/lab/web-server-deployment-cfn.yml

Application Load Balancer Service Catalog Product

Introduction

Application Load Balancer operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers, IP addresses and Lambda functions based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications.

Product Link

https://aws.amazon.com/elasticloadbalancing/features/#Details_for_Elastic_Load_Balancing_Products

Provision Product Name

Name of the product when calling product selector lambda: alb

Product IAM Role

ALB product will be launch under following IAM role: **sc-alb-product-role**

Security Features

List of security feature included by default in product:

- Deploy inside VPC

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **ALBName**: The name of ALB
- **SecurityGroupIds**: The VPC Security Groups Ids as comma separated list
- **SubnetIds**: The VPC Subnets Ids as comma separated list
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the ALB product. You can use these values within your CloudFormation template using ImportValue

- **ALBArn**: the ARN of ALB
- **ALBDNSName**: the DNS of ALB
- **ALBName**: the name of ALB
- **ALBFullName**: the Full name of ALB

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/lab/web-server-deployment-cfn.yml

Application Load Balancer Target Group Service Catalog Product

Introduction

Each target group is used to route requests to one or more registered targets. When you create each listener rule, you specify a target group and conditions. When a rule condition is met, traffic is forwarded to the corresponding target group. You can create different target groups for different types of requests. For example, create one target group for general requests and other target groups for requests to the microservices for your application.

Product Link

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html

Provision Product Name

Name of the product when calling product selector lambda: albtarget

Product IAM Role

ALB Target Group product will be launch under following IAM role: **sc-alb-target-product-role**

Security Features

List of security feature included by default in product:

- Allow only encrypted connection over HTTPS
- Deploy inside VPC

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **ALBTargetName**: The ALB Target Group name
- **HealthCheckIntervalSeconds**: The approximate number of seconds between health checks for an individual target.
- **HealthCheckPath**: The ping path destination where Elastic Load Balancing sends health check requests.
- **HealthCheckPort**: The port that the load balancer uses when performing health checks on the targets.
- **HealthCheckProtocol**: The protocol that the load balancer uses when performing health checks on the targets
- **HealthCheckTimeoutSeconds**: The number of seconds to wait for a response before considering that a health check has failed.
- **HealthyThresholdCount**: The number of consecutive successful health checks that are required before an unhealthy target is considered healthy. (Min: 2, Max: 10, Default: 5)
- **AppPort**: Application Port. Allow values: 443, 8443
- **UnhealthyThresholdCount**: The number of consecutive failed health checks that are required before a target is considered unhealthy. (Min: 2, Max: 10, Default: 5)

- **VpcId**: The VPC Id
- **Tags**: (optional) tags formatted string. For more information check: Tags

List of Outputs

The following values are outputs of the ALB Target Group product. You can use these values within your CloudFormation template using ImportValue

- **ALBTargetId**: the logical Id of ALB Target Group
- **ALBTargetGroupName**: the ALB Target Group name
- **ALBTargetGroupFullName**: the ALB Target Group full name

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/lab/web-server-deployment-cfn.yml

Application Load Balancer Listener Service Catalog Product

Introduction

A listener is a process that checks for connection requests, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to the targets in one or more target groups.

Product Link

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html

Provision Product Name

Name of the product when calling product selector lambda: alblistener

Product IAM Role

ALB Listener product will be launch under following IAM role: **sc-alb-listener-product-role**

Security Features

List of security feature included by default in product:

- Require HTTPS connection
- Require SSL Certificate

List of Parameters

The following parameters need to be described in your CloudFormation (see sample template snippet below) and provided values.

- **CertificateArn**: The ARN of ACM certificate to apply to ALB
- **ALBTargetGroupStack**: The ALB Target Group ARN
- **ALBStack**: The ARN of ALB
- **AppPort**: Application Port. Allow values: 443, 8443. Default 443

List of Outputs

The following values are outputs of the ALB Listener product. You can use these values within your CloudFormation template using ImportValue

- **ALBListenerArn**: the ARN of ALB listener

Sample CloudFormation Template

https://github.com/aws-samples/aws-service-catalog-preventive-control/blob/master/templates/examples/lab/web-server-deployment-cfn.yml