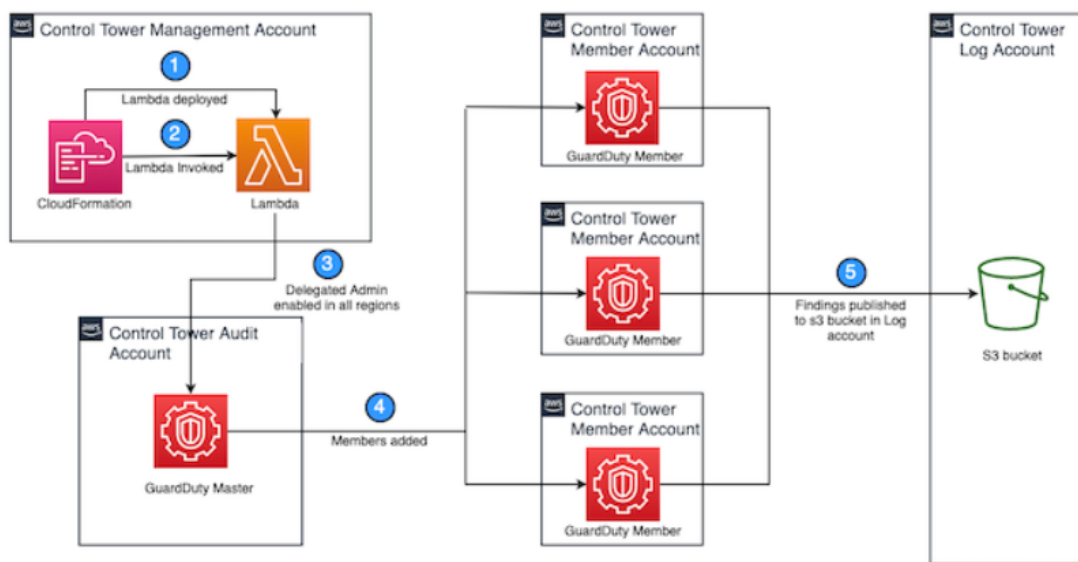


## Synctree Flosports | GuardDuty

GuardDuty can be used to continuously monitor any AWS account or workload for malicious activity and unauthorized behavior. It uses machine learning and integrated threat intelligence to identify abnormal behavior and suspected attackers. This is done from billions of events recorded via AWS CloudTrail, Amazon Virtual Private Cloud (VPC) Flow logs, and Domain Name System (DNS) logs. In this example, we implement GuardDuty to protect accounts that have been created and are governed by AWS Control Tower.

We deploy GuardDuty using the GuardDuty delegated administrator feature. This feature allows you to manage multiple GuardDuty accounts in an AWS Organization, and is broadly applicable to any AWS Organization. Where AWS Control Tower is an ideal use case, it is not a prerequisite for using GuardDuty or the GuardDuty delegated administrator feature.



### Let's Do practically:

- **GDDelegatedAdminAccountId:** The Account ID of the account you would like to be your GuardDuty administrator. This is typically the AWS Control Tower audit account.
- **OrganizationId:** The ID of the organization (in a format such as o-xxxxxxxxxx), which can be found on the Settings tab of the AWS Organizations console.
- **RoleToAssume:** IAM role to be assumed in child accounts to enable GuardDuty. The default is `AWSControlTowerExecution` that is created by AWS Control Tower and has necessary permission.
- **S3Key:** The path to the function.zip file that contains the `EnableGDDelegatedAdminLambda` function. Path of S3-bucket where template file are added in S3 -
- **S3-SourceBucket:** The S3 bucket that hosts the function.zip file that contains the `EnableGDDelegatedAdminLambda` function. The function.zip file is hosted in the `aws-service-catalog-reference-architectures` S3 bucket. we added like this :  
`guardduty/guardduty_enabler.zip`

# Start Creating Stack of Cloud-Formation:

## Step-1 :

**Create stack**

Step 1  
**Create stack**

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

**Prerequisite - Prepare template**

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**  
Selecting a template generates an Amazon S3 URL, where it will be stored.

☒ Amazon S3 URL ☐ Upload a template file

Amazon S3 URL  
`https://cf-templates-x4tlk28liqb4-us-east-2.s3.us-east-2.amazonaws.com/guardduty/aws-control-tower-guardduty-enabler.template`

Amazon S3 template URL  
S3 URL: `https://cf-templates-x4tlk28liqb4-us-east-2.s3.us-east-2.amazonaws.com/guardduty/aws-control-tower-guardduty-enabler.template` [View in Designer](#)

Cancel **Next**

## Step-2 :

**Specify stack details**

Step 1  
**Create stack**

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review guardDuty

**Stack name**

Stack name  
`guardDuty`  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**ComplianceFrequency**  
How frequently (in minutes, between 1 and 3600, default is 60) should organizational compliance be checked?  
`60`

**OrganizationId**  
AWS Organizations ID for the Control Tower. This is used to restrict permissions to least privilege.  
`o-6c7qdp2db`

**RegionFilter**  
Should GuardDuty be enabled for all GuardDuty supported regions, or only Control Tower supported regions?  
`ControlTower`

**RoleToAssume**  
What role should be assumed in child accounts to enable GuardDuty? The default is `AWSControlTowerExecution` for a Control Tower Environment.  
`AWSControlTowerExecution`

**S3SourceBucket**  
Which S3 bucket contains the `guardduty_enabler.zip` file for the `GuardDutyEnabler` lambda function?  
`cf-template-x4tlk28liqb4-us-east-1`

**S3SourceFile**  
What is the S3 path to the zip file for the `GuardDutyEnabler` lambda function?  
`guardduty/guardduty_enabler.zip`

**SecurityAccountId**  
Which account will be the GuardDuty Admin? Enter the AWS account ID. (This is generally the AWS Control Tower Audit account)  
`785581945575`

Add your resource parameters as above in stack

- ComplianceFrequency set 60 by default & other parameters add your own.

- Click on **Next** and reviewed once all parameters

### Step-3 :

Finally check guardDuty working in AWS Console.

