

# Threats Detection & Response in Public Cloud Infrastructure

By Cloud Security Team in Tokopedia

<b>Introduction</b>	<b>1</b>
<b>A Brief Story</b>	<b>1</b>
Cloud Infrastructure	2
Cloud Computing	2
Cloud Security	2
Shared Responsibility Model	2
Cloud Security Challenges & Risks	3
Threat Detection & Response	4
Threat Detection & Response in AWS	4
AWS CloudTrail	4
AWS EventBridge	5
AWS GuardDuty	5
AWS Simple Notification Service (SNS)	6
AWS Lambda	6
<b>Hands-On Time !</b>	<b>7</b>
1. Infrastructure Preparation Step	7
1.1 Choosing Region	7
1.2 Creating an IAM user in your AWS account	8
<b>1.3 Attach a policy to a Role, and attach it to an IAM user</b>	<b>10</b>
<b>1.4 Create EC2 SSH Key Pair</b>	<b>10</b>
1.5 Create AWS Role For Auto-remediation Tools	13
1.6 Switch To Previously Created IAM User	15
1.7 CloudFormation (Before Threat Detection & Response)	16
EC2	16
Security Group	16
S3	17
Creating a stack on the AWS CloudFormation console	17
Go Through Created Resources & Its Vulnerability	17
Restrict S3 Bucket Access Control List (ACL)	22
Restrict EC2 Instance Firewall	23

2. Threat Detection Kit	23
2.1 Amazon CloudTrail	24
2.2 Amazon GuardDuty	24
2.3 Amazon SNS	25
2.4 EventBridge	27
<b>3. Threat Response Kit</b>	<b>29</b>
3.1 Auto-remediation script with Lambda	29
3.2 Auto-remediation script Deployment Process	29
3.3 Setting Up Trigger	29
Detection & Response Simulation	29
4.1 Redeploy Vulnerable CloudFormation Template	29
4.2 Validate Notification & Remediation Result	29
Cleanup Steps	29
<b>References</b>	<b>30</b>

## Introduction

This handbook will cover topics and activities related to the threat detection and response in public cloud infrastructure. The goal of this workshop will allow you to be able to enrich your knowledge and give you a live hands-on experience when implementing security related stuff to the public cloud. Those experiences are derived from Tokopedia business as usual activities of Cloud Security Team. Hopefully this workshop will boost your ideas so you can contribute to the cloud security threat detection and response.

## A Brief Story

*Threat is everywhere at any time.* Those are the words we've to put in our mind. Threat itself can come from many angles in our cloud infrastructure, and it might lead to security incidents, such as misconfiguration, system disruption, data breach, etc. We cannot use the term “we safe 100%” from any threats, since it is impossible to do and if “we 100% safe”, then the topic of threat response is not relevant again. However, we are focussing on how we can enable the most suitable security solution in the use cases that we face, then we have to strengthen our security stuff every single minute.



## 1. Cloud Infrastructure

### 1.1 Cloud Computing

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider. Cloud computing has many benefits, such as:

- **Agility**  
The cloud gives you easy access to a broad range of technologies so that you can innovate faster and build nearly anything that you can imagine.
- **Elasticity**  
With cloud computing, you don't have to over-provision resources up front to handle peak levels of business activity in the future. Instead, you provision the amount of resources that you actually need.
- **Cost savings**  
The cloud allows you to trade fixed expenses (such as data centers and physical servers) for variable expenses, and only pay for IT as you consume it.
- **Deploy globally in minutes**  
With the cloud, you can expand to new geographic regions and deploy globally in minutes.

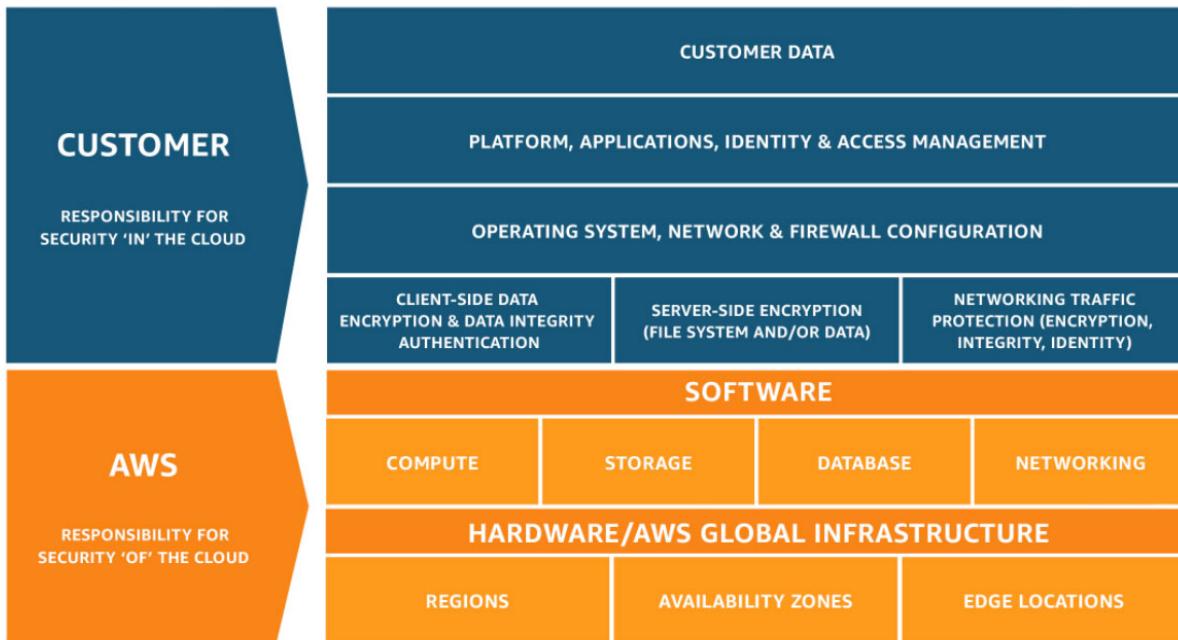
### 1.2 Cloud Security

General explanation for cloud security or cloud computing security is sets of policies, controls, procedures, and technology that works together to protect cloud based systems. This explanation might raise a common question, why cloud based systems need to be protected, isn't it the cloud provider responsibility? To answer this question, introducing Shared Responsibility Model.

#### Shared Responsibility Model

Shared responsibility model is a model that distinguish between security in the cloud and non cloud / on-premises and defines responsibility sharing between us as the cloud user, and cloud provider as the service provider. Below is shared responsibility model from Amazon Web Service

(AWS)



From the diagram above, we can see that in AWS the customer is responsible for security 'in' the cloud while AWS is responsible for security 'of' the cloud.

### Cloud Security Challenges & Risks

The nature of cloud infrastructure is extremely dynamic, therefore it introduce a lot of security challenges & risks, some of them are:

- Data Breaches
- Visibility
- Dynamic Workloads
- Misconfigurations
- Unsecured APIs
- Access Control / Unauthorized APIs
- Securing The Control Plane
- Security Compliance & Auditing

In this workshop session, we will be focused on challenges & risks below:

- Visibility
- Misconfigurations

through threat detection & threat response.

## 2. Threat Detection & Response

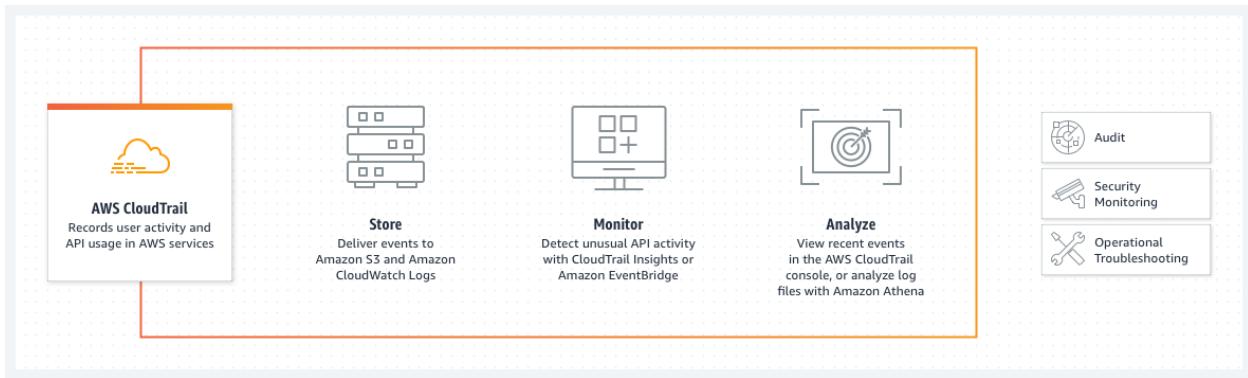
Threat detection is the practice of analyzing the entirety of a security ecosystem to identify any malicious activity that could compromise the network. If a threat is detected, then mitigation efforts must be enacted to properly neutralize the threat before it can exploit any present vulnerabilities.

### 2.1 Threat Detection & Response in AWS

Below are some native services that are useful for doing threat detection & response in AWS.

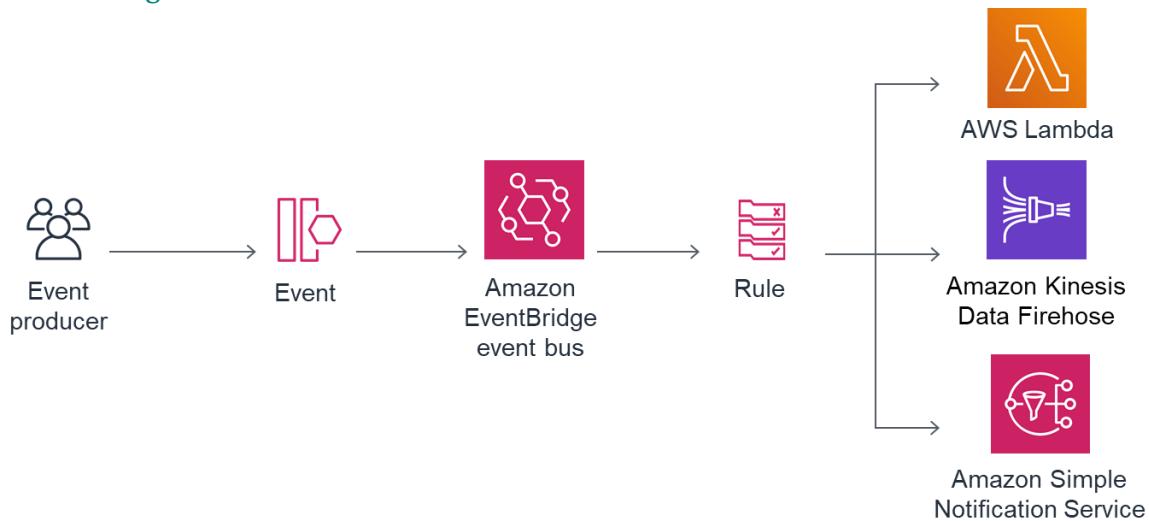
#### AWS CloudTrail

Amazon CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your Amazon Web Services account.



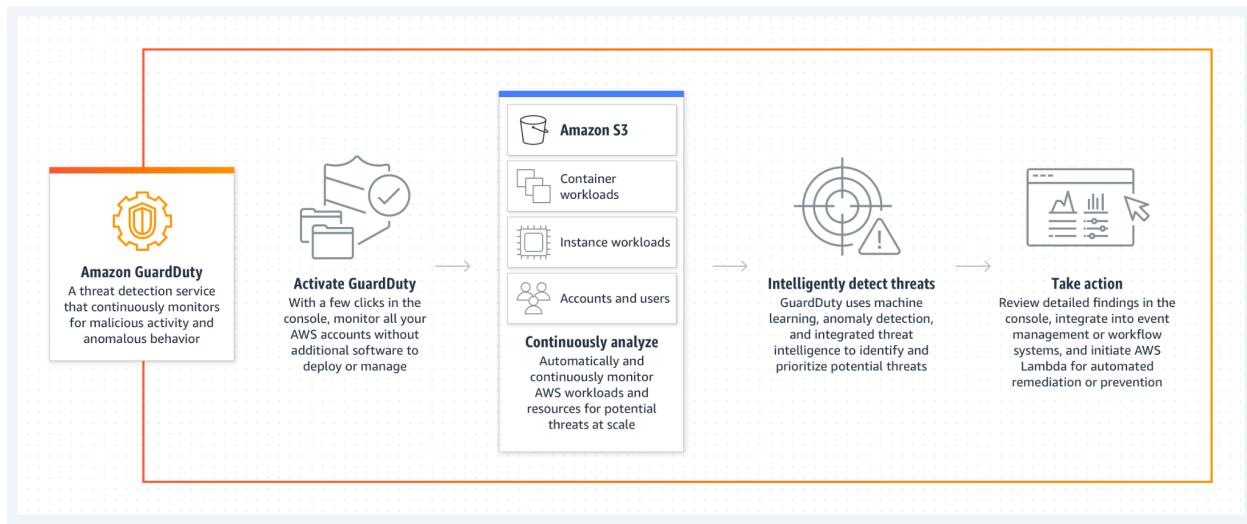
Using Cloudtrail we can deliver events to a specific location, this is of course very important for threat detection, we can create monitoring and detection from specific events that we gathered, e.g. Audit activity on a certain region or activity on a certain AWS service.

## AWS EventBridge



Amazon EventBridge is a serverless event bus that makes it easier to build event-driven applications at scale using events generated from your applications, integrated Software-as-a-Service (SaaS) applications, and AWS services. Using eventbridge we can take events coming from Cloudtrail and redirect them to wherever we want it to go, this is of course very helpful when designing threat detection in the Cloud.

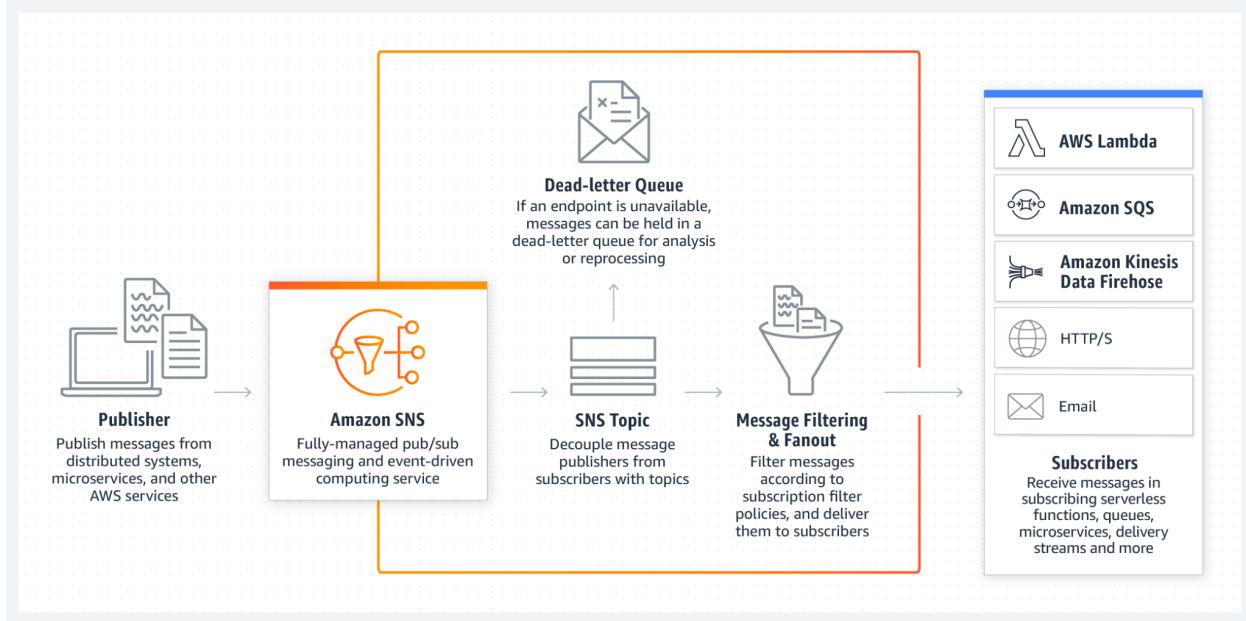
## AWS GuardDuty



Amazon GuardDuty is a threat intelligence detection service from AWS to perform continuous monitoring of our AWS environment. It detects malicious activity and delivers those findings in detail. For respectable service from Google Cloud Platform, they have a Security Command Center (SCC). Details on what GuardDuty can detect can be seen [here](#).

## AWS Simple Notification Service (SNS)

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.



In this workshop we will use SNS heavily to notify us when a threat is detected on our AWS environment.

## AWS Lambda

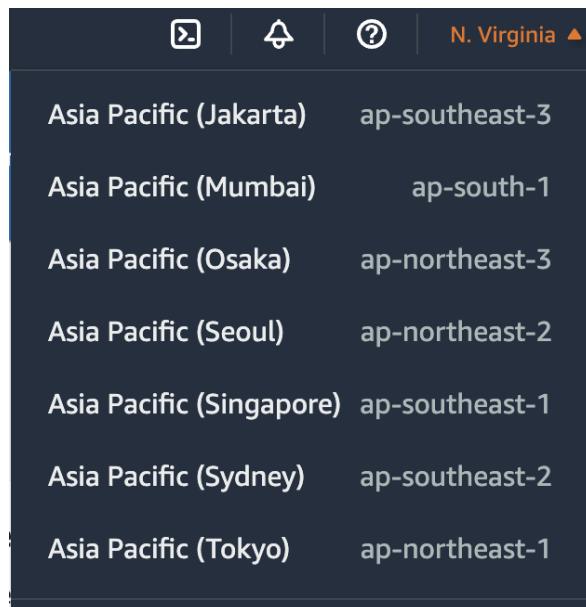
AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. In this workshop we will use Lambda as a threat response mechanism to remediate security misconfiguration or vulnerability that got introduced into the environment.

# Hands-On Time !

## 2. Infrastructure Preparation Step

### 2.1 Choosing Region

Amazon Web Services is available in different regions all over the world, and the console lets you provision resources across multiple regions. You usually choose a region that best suits your business needs to optimize your customer's experience, but we recommend to use the region that is not impacting your personal resources. For this session use the **ap-southeast-2** (Sydney) region.



### 2.2 Creating an IAM user in your AWS account

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources. When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.

1. In the navigation pane, choose **Users** and then choose **Add users**.

The screenshot shows the AWS IAM 'Users' page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and a search bar. Under 'Access management', 'User groups' is collapsed, and 'Users' is selected. The main area shows 'Users (2) Info' with a description of what an IAM user is. There's a search bar labeled 'Find users by username or access key'. Below it is a table header with columns: User name, Groups, Last activity, MFA, Password a..., and Active. A blue 'Add users' button is located in the top right corner of the main content area.

2. Type “**workshop-iamuser1**” for the username of the new user. This is the sign-in name for AWS.
3. Select the type of access this set of users will have. In this workshop we will choose **Password - AWS Management Console access**.

#### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*	<input type="text" value="workshop-iamuser1"/>
<a href="#">+ Add another user</a>	

#### Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type*	<input type="checkbox"/> <b>Access key - Programmatic access</b> Enables an <b>access key ID</b> and <b>secret access key</b> for the AWS API, CLI, SDK, and other development tools.
	<input checked="" type="checkbox"/> <b>Password - AWS Management Console access</b> Enables a <b>password</b> that allows users to sign-in to the AWS Management Console.

Console password*	<input checked="" type="radio"/> Autogenerated password <input type="radio"/> Custom password
-------------------	--

Require password reset	<input type="checkbox"/> User must create a new password at next sign-in Users automatically get the <b>IAMUserChangePassword</b> policy to allow them to change their own password.
------------------------	---

4. Choose **Autogenerated password**
5. Uncheck User must create a new password at next sign-in

Require password reset	<input type="checkbox"/> User must create a new password at next sign-in Users automatically get the <b>IAMUserChangePassword</b> policy to allow them to change their own password.
------------------------	---

6. Choose **Next: Permissions**.

7. On the Set permissions page, choose **Attach existing policies directly** and pick policy below:

- Managed policy AmazonEC2FullAccess
- Managed policy AmazonS3FullAccess
- Managed policy AWSCloudFormationFullAccess
- Managed policy AmazonGuardDutyFullAccess
- Managed policy AWSLambda\_FullAccess
- Managed policy AmazonEventBridgeFullAccess
- Managed policy CloudWatchFullAccess
- Managed policy AmazonSNSFullAccess
- Managed policy AWSCloudTrail\_FullAccess
- Managed policy CloudWatchLogsFullAccess

8. Choose **Next: Tags**

9. Choose **Next: Review**

10. On the review page make sure that the configuration looks like below.

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	workshop-lamuser1
AWS access type	AWS Management Console access - with a password
Console password type	Autogenerated
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	<a href="#">AmazonEC2FullAccess</a>
Managed policy	<a href="#">AmazonS3FullAccess</a>
Managed policy	<a href="#">AWSCloudFormationFullAccess</a>
Managed policy	<a href="#">AmazonGuardDutyFullAccess</a>
Managed policy	<a href="#">AWSLambda_FullAccess</a>
Managed policy	<a href="#">AmazonEventBridgeFullAccess</a>
Managed policy	<a href="#">CloudWatchFullAccess</a>
Managed policy	<a href="#">AmazonSNSFullAccess</a>
Managed policy	<a href="#">AWSCloudTrail_FullAccess</a>

Tags

[Cancel](#) [Previous](#) [Create user](#)

11. Choose: **Create User**

12. After user creation is completed, click at **Download .csv** link to get credential from previously created user and keep it somewhere safe.

## Add user

1 2 3 4 5

**Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://535863243830.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Password	Email login instructions
	workshop-iamuser1	***** Show	<a href="#">Send email</a>

## 2.3 Create AWS Role For Auto-remediation Tools

1. Go to IAM menu > Roles > Create role

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and links for 'Dashboard', 'Access management' (User groups, Users), and 'Roles'. The main area has a header 'Roles (11) Info' with a description: 'An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.' Below this is a search bar and a table with columns: 'Role name', 'Trusted entities', and 'Last activity'. One row is visible: 'aws-vncflowlns-cloudwatch-role', 'AWS Service: vnc-flow-lns', and '6 days ago'.

2. In Step 1, Select trusted entity
  - **Trusted entity type:** AWS service
  - **Use case:** Lambda

**Select trusted entity**

**Trusted entity type**

- AWS service
 

Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account
 

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity
 

Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation
 

Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy
 

Create a custom trust policy to enable others to perform actions in this account.

**Use case**

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- EC2
 

Allows EC2 instances to call AWS services on your behalf.
- Lambda
 

Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case ▾

**Cancel** **Next**

3. Click Next
4. In Step 2, Add permissions – select the following policies:
  - AmazonEC2FullAccess
  - AmazonS3FullAccess

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

**Add permissions**

**Permissions policies (Selected 1/751)**

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter

"S3full" X Clear filters

Policy name	Type	Description
AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.

▶ **Set permissions boundary - optional**

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

**Cancel** **Previous** **Next**

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

## Add permissions

**Permissions policies (Selected 2/751)**  
Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter

"ec2full" X Clear filters

Policy name Type Description

 AmazonEC2FullAcc... AWS m... Provides full access to Amazon EC2 via the AWS Management Console.

► Set permissions boundary - *optional*  
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

5. Hit **Next**
6. Enter Role name called **auto-remediation-lambda-role**
7. Scroll down and click **Create role**

IAM > Roles > Create role

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

## Name, review, and create

**Role details**

Role name  
Enter a meaningful name to identify this role.

Description  
Add a short explanation for this policy.

Maximum 1000 characters. Use alphanumeric and '+-=\_,@-\_` characters.

Step 1: Select trusted entities Edit

```

1  [
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ]
9     }
10    ]
11  ]

```

## 2.4 Switch To Previously Created IAM User

To sign in to an AWS account as an IAM user using an IAM users you will need credentials and IAM Users sign-in URL, you can find this on **.csv file** that was previously downloaded in **Step 2.2**

User name	Password	Access key ID	Secret access key	Console login link
workshop-iamuser1				URL

Or go to IAM > Users > workshop-iamuser1 > Security credentials > **Console sign-in link (copy the link)**

The screenshot shows the AWS IAM console interface. On the left, there's a navigation sidebar with options like Dashboard, Access management, Users, and Access reports. The 'Users' section is currently selected. In the main area, under the 'workshop-iamuser1' user, the 'Summary' tab is active. It displays details such as User ARN, Path, and Creation time. Below the summary, there are tabs for Permissions, Groups, Tags, Security credentials, and Access Advisor. The 'Security credentials' tab is highlighted with a red box. Under this tab, the 'Sign-in credentials' section shows a 'Console sign-in link' which is also highlighted with a red box.

Click or open the url that was specified inside **Console Login Link** and fill up the form with **Username** and **Password** that specified in the CSV file and click **Sign In**.

## Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

**Sign in**

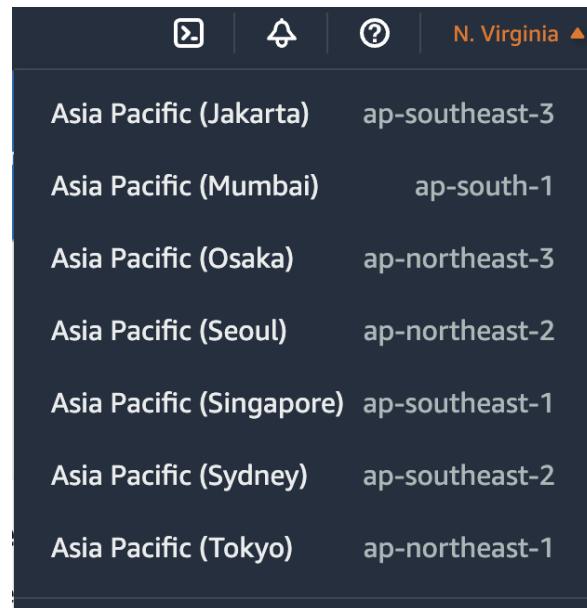
[Sign in using root user email](#)

[Forgot password?](#)

## 2.5 Create EC2 SSH Key Pair

If you do not already have one, create a new key pair in the Amazon EC2 console for each AWS Region where you plan to receive data. Use the steps below.

1. In your AWS Management Console, choose **ap-southeast-2 (Sydney)** AWS Region. You need to create a key pair for every AWS Region you choose.



2. Choose Services > EC2 > Network & Security > Key Pairs, and then choose Create Key Pair.

## Create key pair Info

**Key pair**  
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

**Name**  
 The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type** Info  
 RSA  
 ED25519

**Private key file format**  
 .pem  
 For use with OpenSSH  
 .ppk  
 For use with PuTTY

**Tags (Optional)**  
No tags associated with the resource.  
[Add tag](#)  
 You can add 50 more tags.

[Cancel](#) **Create key pair**

- 3 . Enter a friendly name like **workshop-toko-academy-key**
- 4 . Save the private key, make it accessible to your ssh utility of choice, and set the ownership/permissions as needed (e.g. on unix system run: `chmod 400 <key name>.pem`).

## 2.6 Amazon GuardDuty

For this workshop, let's enable Amazon GuardDuty.

1. Go to the search bar and search “GuardDuty”
2. Select **ap-southeast-2 (Sydney) Region**

The screenshot shows the AWS search interface. The search bar at the top contains the text "guardduty". Below the search bar, there is a message: "The new AWS Console Starting April 2022," followed by "Search results for 'guardduty'". On the left, there is a sidebar with categories: Services (1), Blogs (29), Documentation (38,428), Knowledge Articles (23), Events (3), and Marketplace (58). The main content area is titled "Services" and features a card for "GuardDuty" with the subtext "Intelligent Threat Detection to Protect Your AWS Accounts and Workloads". At the bottom right of the main content area, there is a link "See all 29 results ▶".

- 3 . Go to **GuardDuty** and click **Get Started**
- 4 . Turn it on by clicking **Enable GuardDuty**

The screenshot shows the "Welcome to GuardDuty" page. At the top, it says "Welcome to GuardDuty" and "30 day free trial". Below that, there is a section titled "Service permissions" with the text: "When you enable GuardDuty, you grant GuardDuty permissions to analyze AWS CloudTrail logs, VPC Flow Logs, and DNS query logs to generate security findings. [Learn more](#)". There is a button "View service role permissions". Below this, there is a note: "Note: GuardDuty doesn't manage AWS CloudTrail logs, VPC Flow Logs, and DNS query logs or make their events and logs available to you. You can configure the settings of these data sources through their respective consoles or APIs. You can suspend or disable GuardDuty at any time to stop it from processing and analyzing events and logs. [Learn more](#)". Further down, it says: "When you enable GuardDuty for the first time, your AWS account is automatically enrolled in a 30 day [GuardDuty free trial](#). Learn more about [GuardDuty pricing](#)". At the bottom right, there is a large orange button labeled "Enable GuardDuty".

## 2.7 CloudFormation

### 2.7.1 What is CloudFormation

AWS CloudFormation is an infrastructure as code (IaC) service that allows you to easily model, provision, and manage AWS and third-party resources.

### 2.7.2 S3

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

### 2.7.3 EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. EC2 offers many options that enable you to build and run virtually any application. With these possibilities, getting started with EC2 is quick and easy to do. This page provides you the resources to get you started with EC2 instances.

### 2.7.4 Security Group

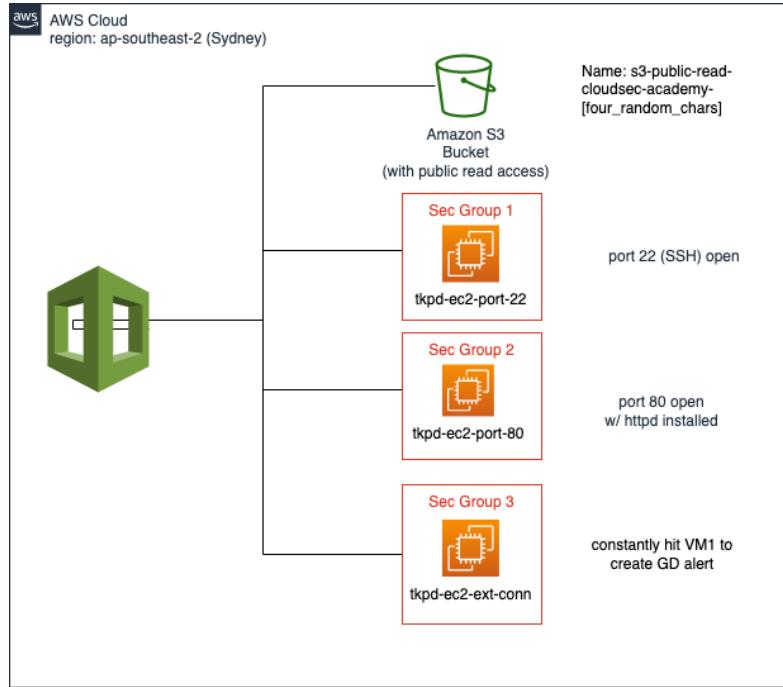
A security group acts as a virtual firewall, controlling the traffic that is allowed to reach and leave the resources that it is associated with. For example, after you associate a security group with an EC2 instance, it controls the inbound and outbound traffic for the instance.

### 2.7.5 Creating a stack on the AWS CloudFormation console

Use the prepared cloud formation template (.yaml) that available in the github  
<https://github.com/tokopedia/cloudsec-academy-2022>

### 2.7.6 Go Through Created Resources & Its Vulnerability

In this cloud formation template, you will create the following resources in your AWS account:

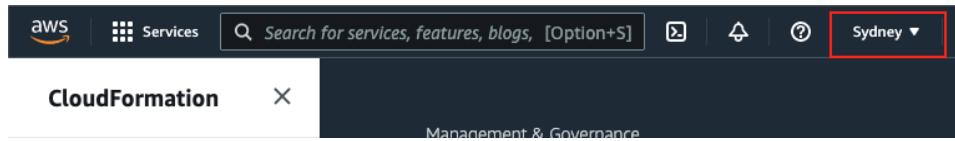


1. 1x S3 Bucket where the Bucket Policy is configured to be public ("s3-public-read-cloudsec-academy-[four\_random\_chars]")
2. 3x EC2 instances with port 80 & 22 open, and Test VM
  - a. **tkpd-ec2-port-22** EC2 Instance port 22 open
  - b. **tkpd-ec2-port-80** EC2 Instance that runs https that display welcome page
  - c. **tkpd-ec2-ext-conn** EC2 Instance that create simulated malicious traffic activity (cryptomining, dnsDataExfil, etc)
3. 3x Security Groups for each EC2 instances to configure firewall
  - a. **InstanceSecurityGroup80** Firewall for tkpd-ec2-port-80
  - b. **InstanceSecurityGroup22** Firewall for tkpd-ec2-port-22
  - c. **InstanceSecurityGroupTester** Firewall for tkpd-ec2-ext-conn

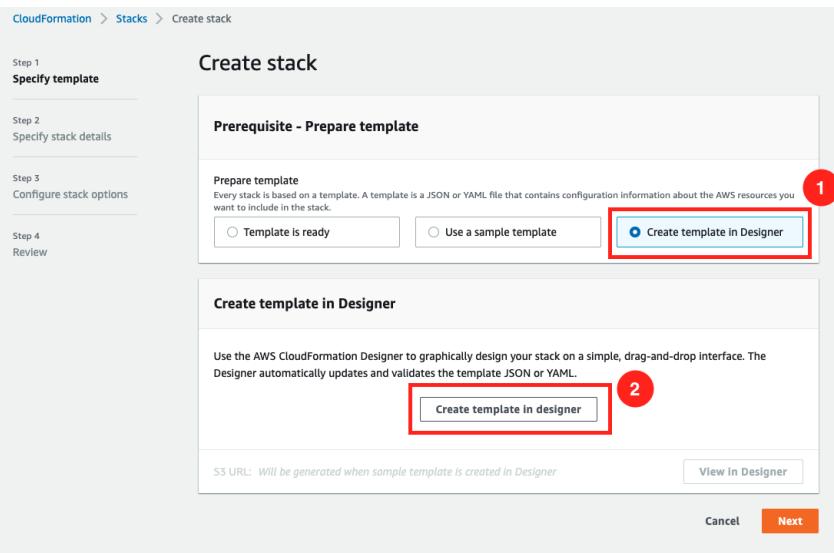
## Starting the Create Stack wizard

### To create a stack on the CloudFormation console

1. Open the AWS CloudFormation console at <https://console.aws.amazon.com/cloudformation>.
2. Make sure you have selected ap-southeast-2 (Sydney) region



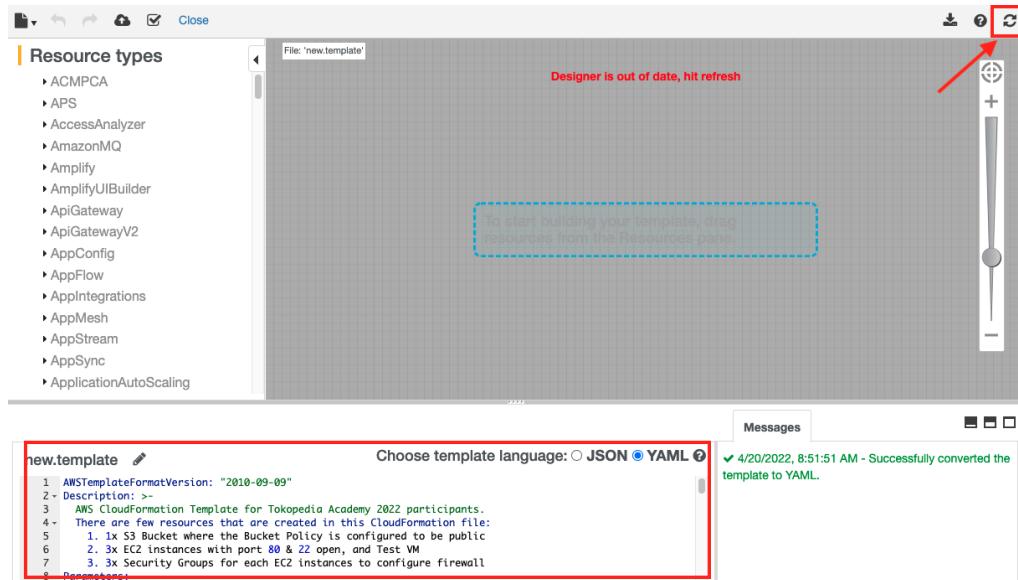
3. Create a new stack by using one of the following options:
  - Choose **Create Stack > With new resources (standard)**
4. Select “Create template in Designer”(1) and click “Create template in designer”(2), which will redirect you to a *Designer*\* page.



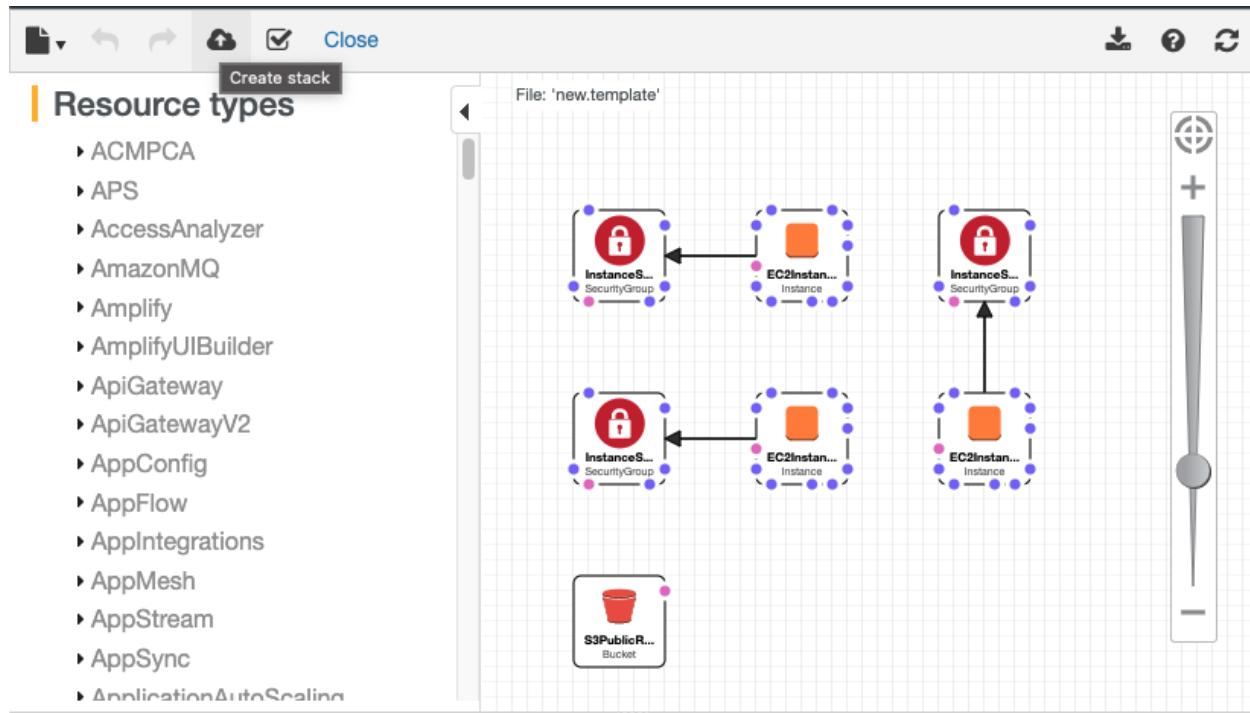
5. Select Parameters (1) > Template (2) > choose YAML (3)



6. Copy and paste the *main.yaml* code from the [github repository](#) to the inline code feature of *Designer* then hit refresh on the top-right corner



7. You will see the visual diagram of the to-be created resource. Once you have verified that the resources are correct, click the *Create stack button* (💡)



8. Noticed that there is a S3 URL generated. AWS created a S3 bucket that stores the cloudformation template file (.yaml) that you just created. Click "Next"

**Create stack**

**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://s3-ap-southeast-1.amazonaws.com/cf-templates-1aoqjdgo528ya-ap-southeast-1/2022110wLy-new.templategcqz8`

Amazon S3 template URL

S3 URL: `https://s3-ap-southeast-1.amazonaws.com/cf-templates-1aoqjdgo528ya-ap-southeast-1/2022110wLy-new.templategcqz8` View in Designer

**Cancel** **Next**

## 9. Follow the prompts

- Stack name : **Tokopedia-Academy-April-2022**
- BucketChars : enter 4 random characters (eg. a3w8)
- InstanceType : **t2.micro**
- KeyName : Select the KeyPair that you have created in [step 1.4 Create EC2 SSH Key Pair](#)

**Specify stack details**

**Stack name**

Stack name  
`Tokopedia-Academy-April-2022`

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.

**BucketChars**  
Please enter 4 random digit characters  
`a3w8`

**InstanceType**  
EC2 instance type  
`t2.micro`

**KeyName**  
The existing of EC2 KeyPair name to enable SSH access to the instance  
`jason.gautama`

**Cancel** **Previous** **Next**

## 10. Click “Next” > “Next” > Review the input > “Create stack”

11. You can hit the refresh button to check the progress. Once the CloudFormation status is “**CREATE\_COMPLETE**”, you can go through the tabs (“Events”, “Resources”, “Outputs”) and explore the available information

Stack ID	Description
arn:aws:cloudformation:ap-southeast-1:330032264459:stack/Tokopedia-Academy-April-2022/11aba120-c050-11ec-8c68-02251b8105e4	AWS CloudFormation Template for Tokopedia Academy 2022 participants. There are few resources that are created in this CloudFormation file: 1. 1x S3 Bucket where the Bucket Policy is configured to be public 2. 3x EC2 instances with port 80 & 22 open, and Test VM 3. 3x Security Groups for each EC2 instances to configure firewall

Status	Status reason
<b>CREATE_COMPLETE</b>	-

\*Note: *AWS CloudFormation Designer* (Designer) is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor.

We have successfully created the resources using CloudFormation! **Kudos to you!**  
Now let's explore each newly created resources and how to remediate them

## Restrict S3 Bucket Access Control List (ACL)

- Search for “S3” on the AWS console search bar or <https://s3.console.aws.amazon.com/>
- Find buckets by name “s3-public-read-cloudsec-academy” > click the bucket name
- Go to Permissions > Block public access (bucket settings). You will see that **Block all public access: Off**. This means the object inside the public can be public

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLS), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Edit**

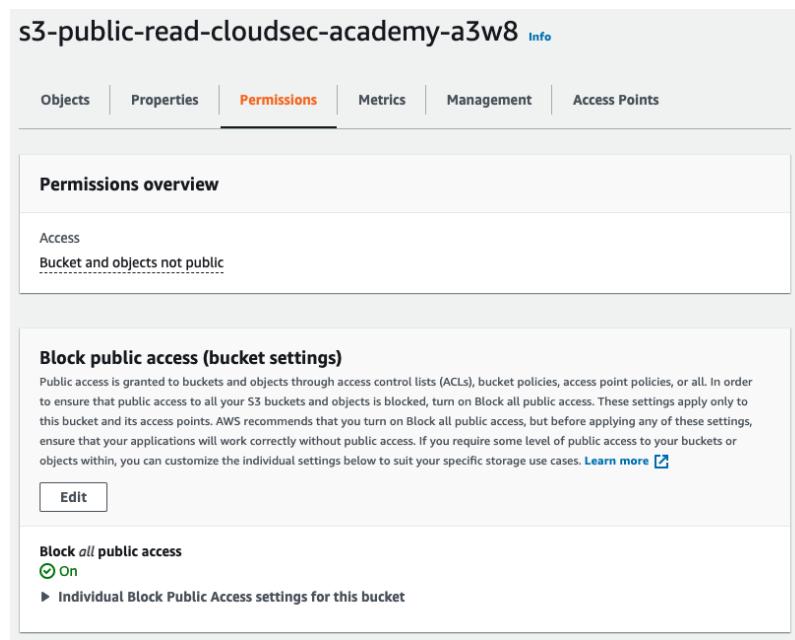
**Block all public access**

**Off**

**Individual Block Public Access settings for this bucket**

- Block public access to buckets and objects granted through *new access control lists (ACLS)*  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through *any access control lists (ACLS)*  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through *new public bucket or access point policies*  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through *any public bucket or access point policies*  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

- Before we remediate, let's test if the bucket is open to the public by **uploading a file** and making it public.
  - Select “Upload” > Add files > “*upload any pictures*”
  - Expand “Permissions” > Predefined ACLs > Grant public-read access > tick the box “I understand the risk of granting public-read access to the specified objects.” > Upload
  - You can check the photo that everyone in the internet can see the photo you just uploaded (Open the link in Incognito to verify)
- To remediate the findings, go to Permissions tab > Under **Block public access (bucket settings)**, click Edit > tick the box **Block all public access** > Save changes > type “confirm” > click **Confirm** button



- Open the same image you have uploaded to the S3 bucket, you will notice that you get an access denied message.

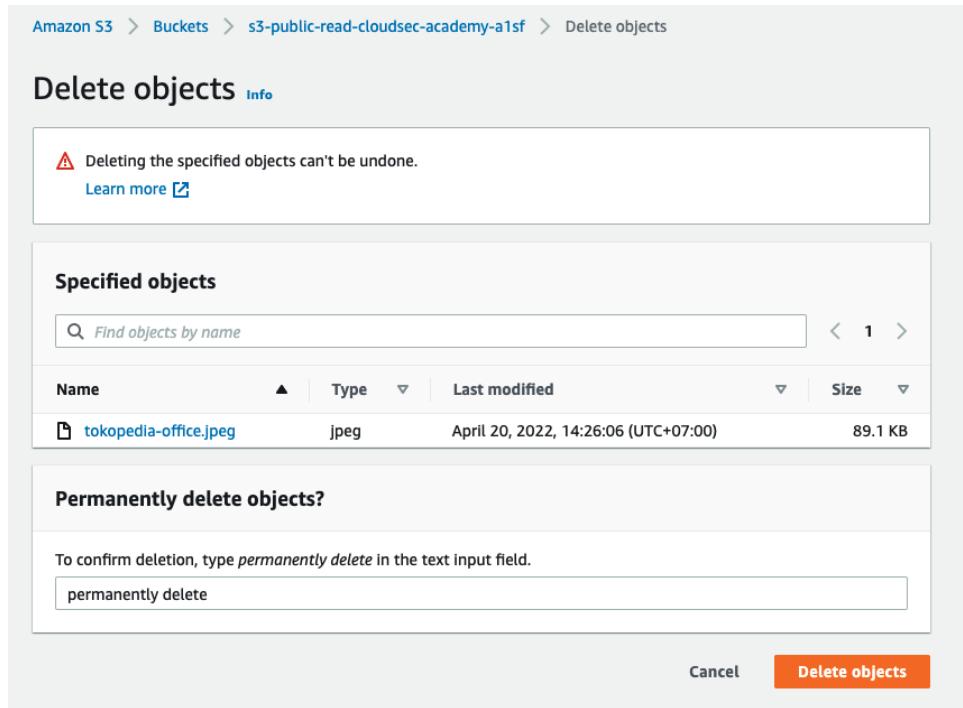
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

▼<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>2NVNRV2SWN9622TF</RequestId>
  <HostId>15W+LrAOVGCG1sd+3WD30ojFwlPNiC8dPyAp1zqQkHRMZubKXyTh4YJ9xpqXIPMXeYLxZDpGlfk==</HostId>
</Error>

```

- Before we proceed, let's delete the image on S3 bucket before you go to the next step. **It is important since we need to remove all objects in S3 before we can delete the CloudFormation stack in the next step!**



## Restrict EC2 Instance Firewall (aka Security Group)

Opening EC2 instance to all internet (0.0.0.0/0) is highly discouraged as attacker can try to perform brute force attacks and other malicious acts that can gain them access to the instance (and worst cases to the whole organization infrastructure)

The scenario is we want to test if by removing the Inbound rule on the Security Group will result in the user unable to access the page that is hosted by tkpd-ec2-port-80 instance. So before we start, **open a new tab and specify the IP address of the tkpd-ec2-port-80**.

To find the IP address / DNS of **tkpd-ec2-port-80**, you can go to the created CloudFormation stacks > Tokopedia-Academy-April-2022 > Outputs > and find **PublicDNSport80** or **PublicIPport80**.

Key	Value	Description	Export name
CurrentAvailabilityZone	ap-southeast-2b	Availability Zone of the newly created EC2 Instance	-
InstanceIdEC2Tester	i-07f287517489f95dd	InstanceId of EC2 instance ("tkpd-ec2-ext-conn")	-
InstanceIdEC2p22	i-03d4036356caa63f8	InstanceId of EC2 Instance with port 22 open	-
InstanceIdEC2p80	i-0a31f24b5c0bc3ee5	InstanceId of EC2 Instance with port 80 open	-
PublicDNSport80	<a href="http://ec2-54-206-51-140.ap-southeast-2.compute.amazonaws.com">ec2-54-206-51-140.ap-southeast-2.compute.amazonaws.com</a>	Use this DNS url to visit the welcome page	-
PublicIPport80	54.206.51.140	Public IP address for EC2 port 80 ("tkpd-ec2-port-80" instance)	-
PublicReadBucket	s3-public-read-cloudsec-academy-a3w8	S3 Bucket that is made to be public	-

If you received a “This site can’t be reached” error, make sure that you put **http://** in front of the IP address.



To start remediation process for this instance you can follow steps below:

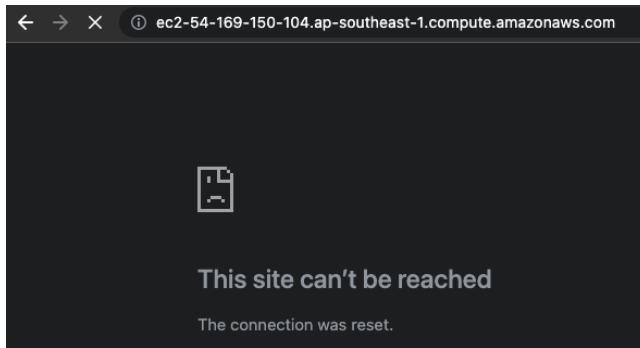
1. Go to EC2 > Security Group > search for tkpd-sg-port-80
2. Tick the checkbox on the left side of “tkpd-sg-port-80” > select **Inbound rules** tab > click **Edit inbound rules**
3. Notice that we have Inbound TCP port 80 with 0.0.0.0/0 set. Click **Delete** and Save rules

The screenshot shows the 'Edit inbound rules' page in the AWS Management Console. A single rule is listed:

- Security group rule ID:** sgr-0aa450a20124e94b4
- Type:** HTTP
- Protocol:** TCP
- Port range:** 80
- Source:** Custom (0.0.0.0/0)
- Description - optional:** (empty)

Buttons at the bottom include 'Add rule', 'Delete', 'Cancel', 'Preview changes', and 'Save rules'.

- Verify that you are no longer able to view the page that you opened



You can repeat steps #1-3 for tkpd-ec2-port-22 and tkpd-ec2-ext-conn Security Group if you wish. Try and restrict the CIDR range (0.0.0.0/0) with your own IP (ex. 36.79.212.123/32). You can use tools like <https://whatismyipaddress.com/> to find your IP address and check if you are still able to access it.

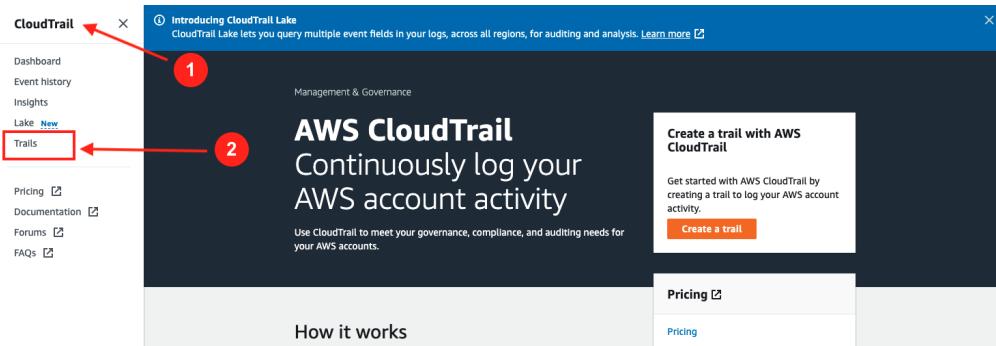
### 3. Threat Detection Kit

Here are the kits we need to set up threat detection. Overall we need to setup 2 services, Amazon GuardDuty for threat intelligence detection service, and AWS Lambda for the auto remediation as a response to the threat

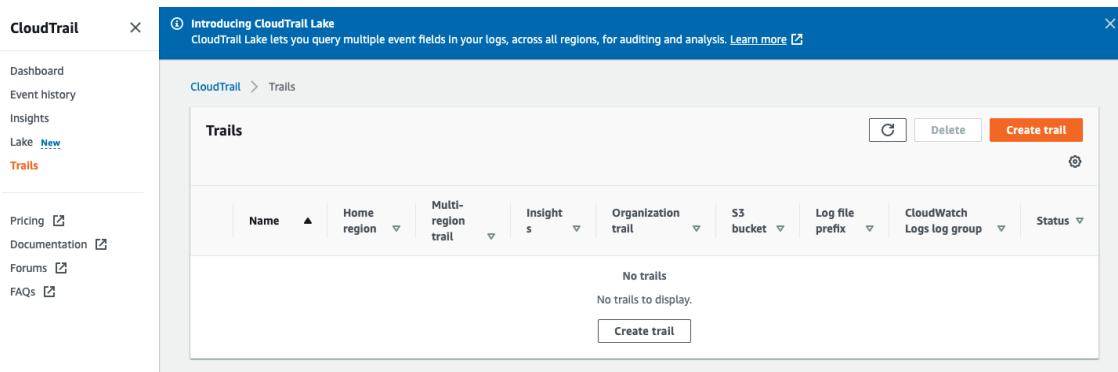
#### 3.1 Amazon CloudTrail

On this following section, you will enable AWS CloudTrail

- Go to the search bar and type “CloudTrail”
- Expand the sandwich bar and click **Trails**



### 3. Click Create Trail



#### 4. Fill the necessary field

- Trail Name:** security-trail
  - Storage Location** -> Create new S3 bucket
  - Log file SSE-KMS encryption** -> Un-select it
  - Leave the other fields as its default, then click next
  - Event type** -> Select the **Management Events**
  - Scroll down to the bottom and click next
- Review your config and it will look like this
  - Choose create trail
  - Done !!!

## 3.2 Amazon SNS

The AWS CloudTrail has now logged the events on our infrastructure. Now, we have to set up the notifications in order to allow us to know the events via email. For this section, we will have 2 part: Firstly is setting up the **Topics** and secondly the **Subscriptions** part

- Topics Part
  - On the search bar, type “SNS”

- 1.2. On the left pane, click the three-horizontal sandwich and click on **Topics**

The screenshot shows the AWS Amazon SNS Topics page. The left sidebar has a 'Topics' link under the 'Dashboard' section. The main content area shows a table titled 'Topics (0)' with columns 'Name', 'Type', and 'ARN'. A message says 'No topics' and 'To get started, create a topic.' At the bottom is a large orange 'Create topic' button.

- 1.3. Select the “Create Topic” button
- 1.4. For the Type, we will use the “Standard”
- 1.5. **Name** for Topic: **ec2-and-s3-event-topic**
- 1.6. For the rest, you can leave it as default value
- 1.7. Hit “Create topic”

## 2. Subscriptions Part

- 2.1. Still on the Amazon SNS page, now you should be on the Topics’s page that you have just created. If not, Go to the Topics on the left pane and choose your created topic previously.
- 2.2. Scroll down and hit “Create subscription”

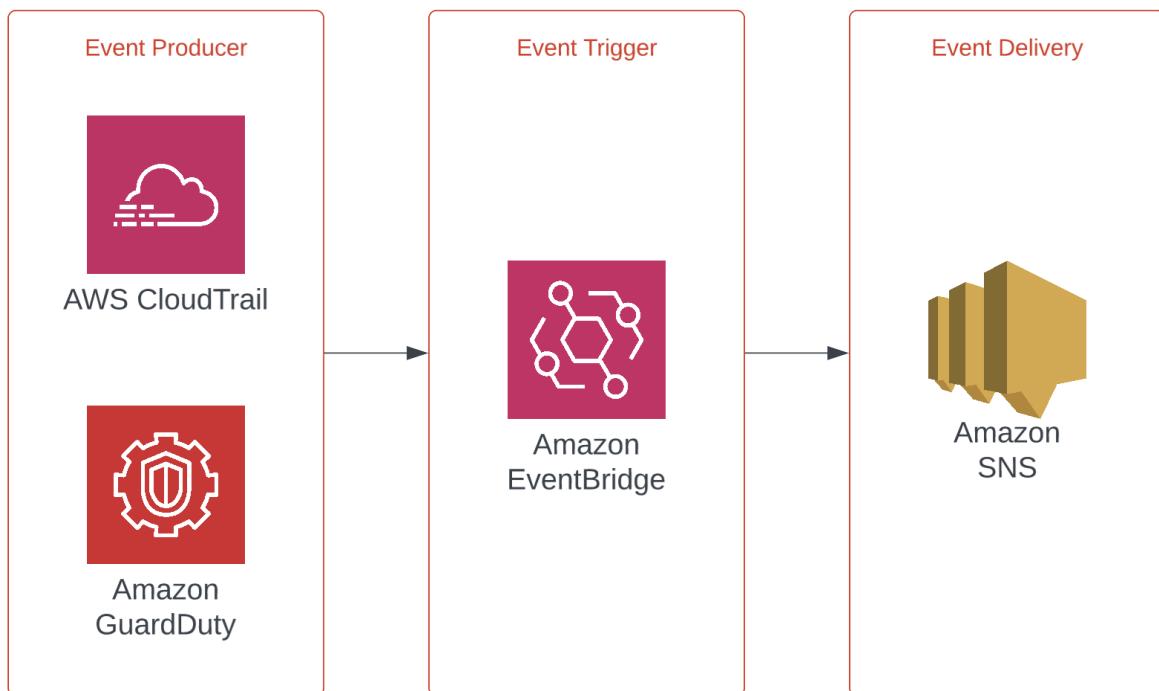
The screenshot shows the AWS Amazon SNS Subscriptions page. The top navigation bar has 'Subscriptions' selected. The main content area shows a table titled 'Subscriptions (0)' with columns 'ID', 'Endpoint', 'Status', and 'Protocol'. A message says 'No subscriptions found' and 'You don't have any subscriptions to this topic.' At the bottom is a large orange 'Create subscription' button.

- 2.3. On the protocol field, choose “Email”
- 2.4. Fill the endpoint with your email that will receive the notification
- 2.5. Next up is clicking “Create subscription”
- 2.6. Now, check your mail and confirm the message by clicking the sent URL
- 2.7. Finish, now you are ready to receive your notification, and last but not least, we have to set up the event trigger on EventBridge section below

Details	
ARN	Status
arn:aws:sns:ap-southeast-2:312653230926:ec2-and-s3-event-topic:bae2c641-ce0b-4c8f-bf1c-0e9a25a9b400	Confirmed
Endpoint	Protocol
jason.gautama@tokopedia.com	EMAIL
Topic	
ec2-and-s3-event-topic	

### 3.3 EventBridge

Now we are arriving at the last configuration for the Threat Detection Kit. A bit summary, we have had CloudTrail and GuardDuty as the event producer, followed by Amazon SNS as the event delivery. **EventBridge will be residing between event producer and event delivery as an event trigger.** Here is the schema to illustrate:



## 1. Event Trigger for EC2

- 1.1. Go to the search bar and type “EventBridge” then click it
- 1.2. On the left pane, choose “Rules” and you will see this page

The screenshot shows the AWS Amazon EventBridge Rules interface. The left sidebar has a tree view with 'Events' expanded, showing 'Rules' selected. The main content area has a heading 'Rules' and a sub-section 'Select event bus' with a dropdown menu set to 'default'. Below this is a table titled 'Rules (0/0)' with a single row 'No rules to display' and a 'Create rule' button.

- 1.3. Choose “Create rule”

- 1.4. Let’s fill the field

- 1.4.1. Name : **ec2-event-trigger**
- 1.4.2. Description : create EC2 event trigger for Tokopedia Academy
- 1.4.3. Choose Next
- 1.4.4. Scroll down and look for Event Pattern section
- 1.4.5. Choose “Custom Patterns (JSON editor)” and paste this JSON Script

```
{
  "source": ["aws.ec2"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": {
    "eventSource": ["ec2.amazonaws.com"],
    "eventName": ["AuthorizeSecurityGroupIngress"]
  }
}
```

- 1.4.6. Click Next

The screenshot shows the 'Event pattern' configuration screen. At the top, there are two tabs: 'Event pattern' (selected) and 'Info'. Below the tabs are two buttons: 'Event pattern form' and 'Custom patterns (JSON editor)'. A dropdown menu says 'Select matching pattern' with a downward arrow, and an 'Insert' button. To the right of these is a radio button labeled 'Content-based filter syntax'.

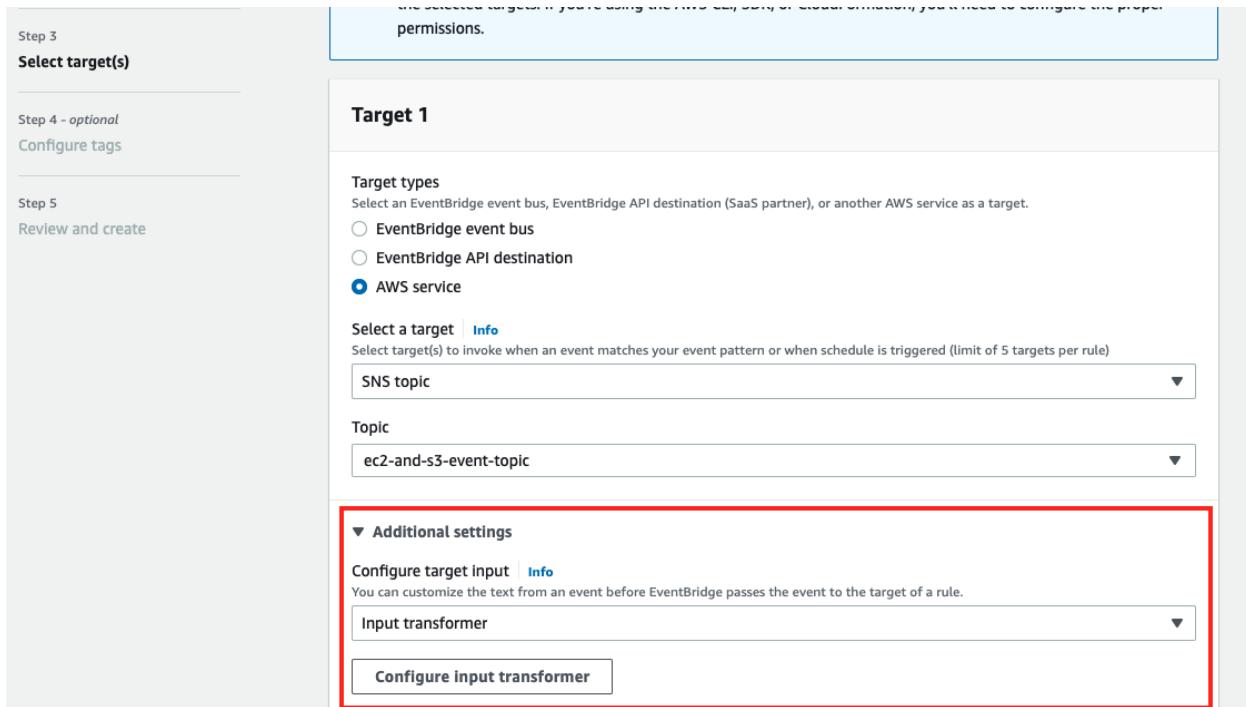
The main area contains a JSON code editor with the following content:

```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["AWS API Call via CloudTrail"],
4   "detail": {
5     "eventSource": ["ec2.amazonaws.com"],
6     "eventName": ["AuthorizeSecurityGroupIngress", "RevokeSecurityGroupIngress"]
7   }
8 }
```

Below the code editor, a message says 'JSON is valid' with a circular icon. At the bottom of the editor are four buttons: 'Copy', 'Prettify JSON', 'Event pattern form', and 'Test pattern'.

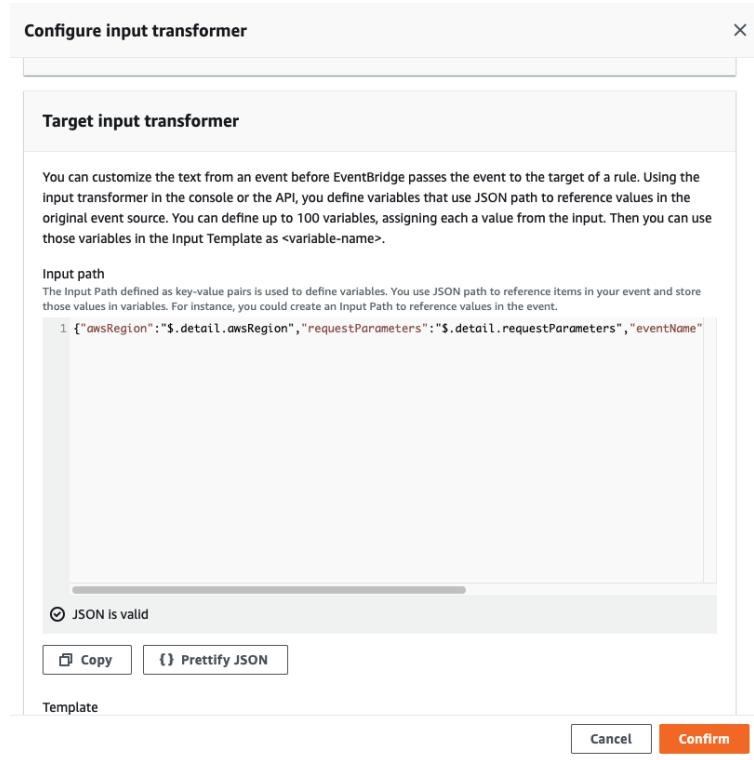
At the very bottom of the interface are three buttons: 'Cancel', 'Previous', and 'Next'.

- 1.4.7. Select a target, choose **SNS topic** as the target
- 1.4.8. On the Topic, choose “ec2-and-s3-event-topic”
- 1.4.9. On the additional settings, choose input transformation and click configure input transformation



- 1.4.10. On the **Target input transformer (not sample event section)**, fill it with this in the **Target input transformer**

```
{"awsRegion":("$.detail.awsRegion", "requestParameters": "$.detail.requestParameters", "eventName":("$.detail.eventName", "eventTime":("$.detail.eventTime")})}
```



#### 1.4.11. On the **Template**, fill by this

"Hi, there is a new EC2 Security Group <eventName> event at <eventTime> in <awsRegion> region. Here is the request <requestParameters>"

##### Template

The Input Template is a template for the information you want to pass to your target. You can create a template that passes either a string or JSON to the target.

```
1 "Hi, there is a new EC2 Security Group <eventName> event at <eventTime> in <awsRegion> region."
```

**Copy**

#### 1.4.12. Click next until the last page, then select “Create rule”

## 2. Event Trigger for S3

- 2.1. Make sure you are still in the Rules page of EventBridge. If not, go to the Rules page as it demonstrates on point 1.1 - 1.2 above
- 2.2. Let's fill the field
  - 2.2.1. Name : **s3-event-trigger**
  - 2.2.2. Description : create s3 event trigger for Tokopedia Academy
  - 2.2.3. Choose next
  - 2.2.4. Scroll down and look for Event Pattern section
  - 2.2.5. Choose “Custom Patterns (JSON editor)” and paste this JSON Script

```
{
  "source": ["aws.s3"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": {
    "eventSource": ["s3.amazonaws.com"],
    "eventName": [
      "CreateBucket",
      "DeleteBucketPublicAccessBlock",
      "PutBucketPublicAccessBlock"
    ]
  }
}
```

- 2.2.6. Click next
- 2.2.7. Select a target, choose **SNS topic** as the target
- 2.2.8. On the Topic, choose “**ec2-and-s3-event-topic**”
- 2.2.9. Expand the Additional settings, go to **Configure target input** > select **input transformer** > click **Configure input transformer**
- 2.2.10. On the **target input transformer**, fill it with this

```
{"awsRegion":("$.detail.awsRegion", "bucketName":("$.detail.requestParameters.bucketName", "eventName":("$.detail.eventName", "eventTime":("$.detail.eventTime")})
```

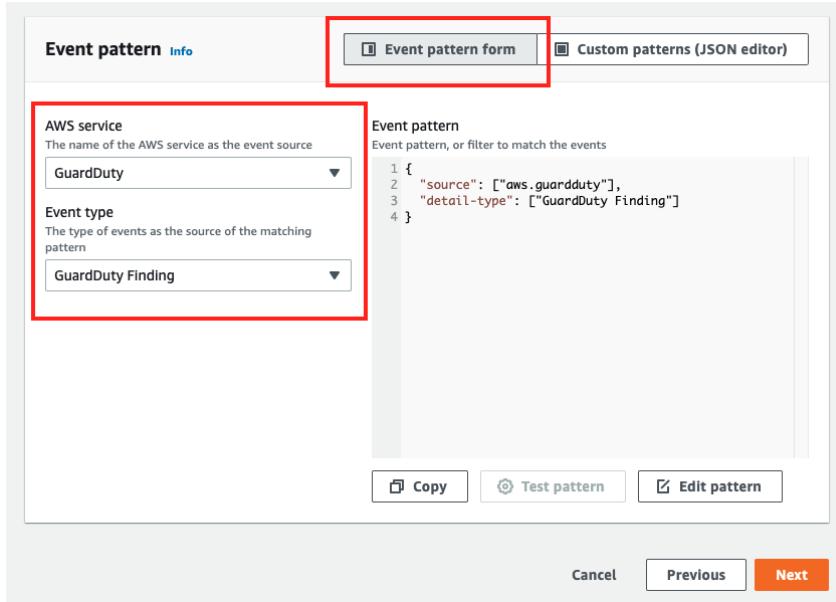
- 2.2.11. On the **template**, fill by this

"Hi, there is a new S3 <eventName> event at <eventTime> which affected bucket <bucketName> and it is residing in <awsRegion> region"

- 2.2.12. Click next until the last page, then select "Create rule"

### 3. Event Trigger for GuardDuty Findings

- 3.1. Make sure you are still in the Rules page of EventBridge. If not, go to the Rules page as it demonstrates on point 1.1 - 1.2 above
- 3.2. Let's fill the field
  - 3.2.1. Name : **guardduty-event-trigger**
  - 3.2.2. Description : create guardduty event trigger for Tokopedia Academy
  - 3.2.3. Choose next
  - 3.2.4. Scroll down and look for **Event Pattern** section
  - 3.2.5. Choose Event pattern form:
    - 3.2.5.1. AWS Service : **GuardDuty**
    - 3.2.5.2. Event Type : **GuardDuty Finding**
  - 3.2.6. Click next



- 3.2.7. On the Target, select **SNS topic** as the target
- 3.2.8. On the Topic, choose "**ec2-and-s3-event-topic**"
- 3.2.9. Click next until the last page, then select "Create rule"

## 4. Threat Response Kit

### 4.1 Auto-remediation script with Lambda

For the Threat Response Kit, we will utilize Lambda to perform the remediation script. We will focus on remediate a few thing below:

1. Revoke EC2 Security Group port 22 SSH widely open/public facing
2. Revoke EC2 Security Group port 3389 RDP widely open/public facing
3. Revoke S3 public bucket policy

Let's start with the EC2 Security Group

1. Open Service **Lambda** on AWS Dashboard
2. Choose **Create function**
3. Let's fill the form
  - a. Function name -> **ec2-security-group-remediation**
  - b. Runtime -> **Python 3.9**
  - c. Permissions
    - i. Execution Role -> Use an existing role -> **auto-remediation-lambda-role**
4. Select **Create function**
5. Paste the code below on the code editor (or go to here:  
<https://github.com/tokopedia/cloudsec-academy-2022/blob/main/auto-remediation/ec2-overly-permissive-remediation.py>)

```
import json
import boto3
import os

def lambda_handler(event, context):
    # Check config
    if os.environ.get('is_active', 'true').lower() == 'false':
        return {
            'statusCode': 200,
            'body': 'EC2 overly permissive auto-remediation is triggered, but it set to not active'
        }
    else:
        # Your remediation logic here
        pass
```

```

# Get Security Group ID

securityGroupID = event['detail']['requestParameters']['groupId']

client = boto3.client('ec2')


# Revoke overly permissive on port 22 SSH

responsePort22 = client.revoke_security_group_ingress(
    CidrIp='0.0.0.0/0',
    FromPort=22,
    GroupId=securityGroupID,
    IpProtocol='TCP',
    ToPort=22
)

# Revoke overly permissive on port 3389 RDP

responsePort3389 = client.revoke_security_group_ingress(
    CidrIp='0.0.0.0/0',
    FromPort=3389,
    GroupId=securityGroupID,
    IpProtocol='TCP',
    ToPort=3389
)

return {
    'statusCode': 200,
    'port22Response': responsePort22,
    'port3389Response': responsePort3389
}

```

6. Click **Deploy** to deploy the code
7. On the menu bar, choose **Configuration**

8. On the edit **General Configuration** and set timeout to **1 minute**, then save
9. On the edit **Environment Variables** and set key to “is\_active” with value “True”, then save the configuration
10. Done for EC2 Security Group set up

Next, the S3 Public Access Policy Remediation

1. Choose **Create function**
2. Let's fill the form
  - a. Function name : **s3-public-bucket-remediation**
  - b. Runtime : **Python 3.9**
  - c. Permissions
    - i. Execution Role -> Use an existing role -> **auto-remediation-lambda-role**
3. Select **Create function**
4. Paste the code below on the code editor (or go to here:  
<https://github.com/tokopedia/cloudsec-academy-2022/blob/main/auto-remediation/s3-public-bucket-remediation.py>)

```
import json

import boto3

import os

def lambda_handler(event, context):

    # Check config

    if os.environ.get('is_active', 'true').lower() == 'false':
        return {

            'statusCode': 200,
            'body': 's3 public bucket auto-remediation is triggered, but it set to not active'
        }
    
```

```

# Retrieve bucket name

bucketName = event['detail']['requestParameters']['bucketName']


# Check whitelisted bucket

whitelistedBuckets = os.environ.get('whitelisted', '')


for whitelistedBucket in whitelistedBuckets.split(','):

    if whitelistedBucket == bucketName:

        # Return response since bucket whitelisted

        return {

            'statusCode': 200,

            'body': "bucket whitelisted"

        }

# S3 public bucket remediation

client = boto3.client('s3')

response = client.get_public_access_block(

    Bucket=bucketName

)

publicAccessConfig = response['PublicAccessBlockConfiguration']

if publicAccessConfig['BlockPublicAcls'] == False or
publicAccessConfig['IgnorePublicAcls'] == False or
publicAccessConfig['BlockPublicPolicy'] == False or
publicAccessConfig['RestrictPublicBuckets'] == False:

    responseRemediation = client.put_public_access_block(

```

```

        Bucket=bucketName,

        PublicAccessBlockConfiguration={

            'BlockPublicAcls': True,
            'IgnorePublicAcls': True,
            'BlockPublicPolicy': True,
            'RestrictPublicBuckets': True
        },
    )

# Return response

return {

    'statusCode': 200,
    'body': responseRemediation
}

# Return response

return {

    'statusCode': 200,
    'body': response
}

```

5. Click **Deploy** to deploy the code
6. On the menu bar, choose **Configuration**
7. On the **edit General Configuration** and set timeout to 1 minute, then save
8. On the **edit Environment Variables** and set the key to “**is\_active**” with value “**True**”.
9. Next up, set new key to “**whitelisted**” with value “**testapp-workshop-xxx**”
10. Save the configuration
11. Done for S3 Public Access set up

## 4.2 Setting Up Trigger

To trigger those 2 remediation scripts on Lambda, we need set rules on EventBridge as the same as Amazon SNS above. Here is the schema looks like

4. Event Trigger for EC2 Remediation
  - 4.1. Go to the search bar and type “EventBridge” then click it
  - 4.2. On the left pane, choose “Rules”
    - 4.2.1. Search for **ec2-event-trigger**
    - 4.2.2. Edit that **Rule**
    - 4.2.3. Click continue until you find **Target**, then **Add Target**
    - 4.2.4. On the Target, type **Lambda** as the target
    - 4.2.5. On the function, choose **ec2-security-group-remediation**
    - 4.2.6. Click next until the last page, then select **Create rule**
  
5. Event Trigger for S3 Remediation
  - 5.1. Go to the search bar and type “EventBridge” then click it
  - 5.2. On the left pane, choose “Rules”
    - 5.2.1. Search for **s3-event-trigger**
    - 5.2.2. Edit that **Rule**
    - 5.2.3. Click continue until you find **Target**, then **Add Target**
    - 5.2.4. On the Target, select **Lambda** as the target
    - 5.2.5. On the function, choose **s3-public-bucket-remediation**
    - 5.2.6. Click next until the last page, then select **Create rule**

Voila, now your Threat Response Kit has been set up !!!

Let's see the magic what Threat Response Kit can do

## 5. Detection & Response Simulation

### 5.1 Verify GuardDuty Findings

We have enabled GuardDuty at the beginning of our workshop. GuardDuty is a powerful service because it provides visibility on your cloud environment when abnormal activity is detected.

1. Go to **GuardDuty > Findings** – you will notice the amount of findings detected as we spawn the CloudFormation template

Finding type	Resource	Last ...	Count
Backdoor:EC2/C&CActivity.BIDNS	Instance: i-0e0b41ec9cced9072	17 hours ago	2
CryptoCurrency:EC2/BtcTool.BIDNS	Instance: i-0e0b41ec9cced9072	17 hours ago	4
CryptoCurrency:EC2/BtcTool.BIDNS	Instance: i-0e0b41ec9cced9072	17 hours ago	8
Policy:S3/BucketBlockPublicAccessDisabled	workshop-lamuser1: ASIAJNR4FSLJKUGAIKA	17 hours ago	1
Trojan:EC2/DNSDataExfiltration	Instance: i-09d15404ba22f0ac5	18 hours ago	8
CryptoCurrency:EC2/BtcTool.BIDNS	Instance: i-09d15404ba22f0ac5	18 hours ago	5
Backdoor:EC2/C&CActivity.BIDNS	Instance: i-09d15404ba22f0ac5	18 hours ago	3
CryptoCurrency:EC2/BtcTool.BIDNS	Instance: i-09d15404ba22f0ac5	18 hours ago	10
Policy:S3/BucketBlockPublicAccessDisabled	workshop-lamuser1: ASIAUJZP5JTREDRYXJJA	18 hours ago	1

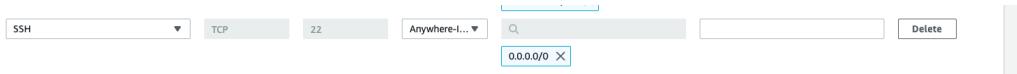
## 5.2 Validate Notification & Remediation Result

Steps specified inside 5.2 will recreate a misconfigured resource, we can follow steps below to check Remediation result and notification that were sent to pre configured email.

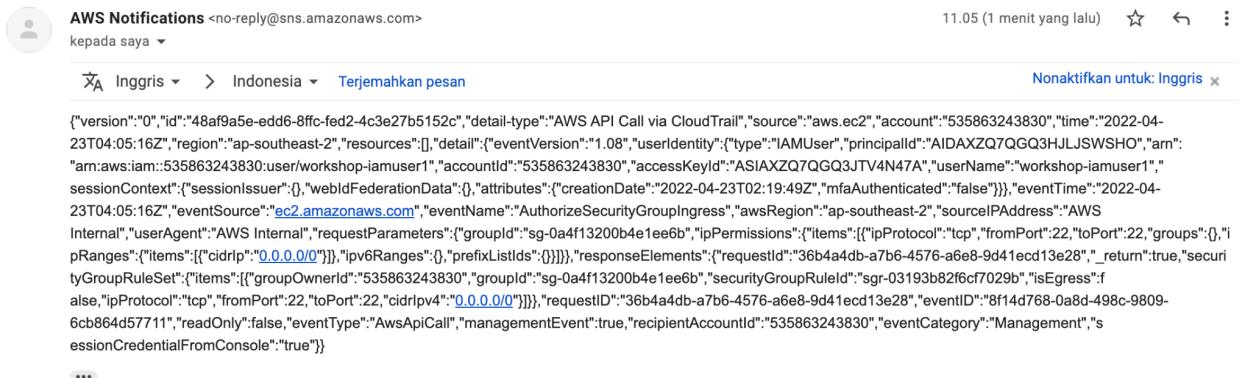
### EC2 Security Group Remediation Result & Notification

To test the remediation process you can follow the steps below:

1. Go to EC2 > Security Groups
2. Find **tkpd-sg-port-22**
3. Click Action > Inbound Rules
4. Click Add rule > Choose “RDP” in Type field & fill up source with **0.0.0.0/0**



5. When rule created you will get notified about someone is adding security group



6. And when checking previously created rule, you will see that the rule is removed by Lambda function

## S3 Public Block Remediation Result & Notification

To test the remediation process you can follow the steps below:

1. Go to S3
2. Find your previously created s3 bucket ([s3-public-read-cloudsec-academy-xxxx](#)) and click it
3. Go to Permissions Tab
4. Click Edit on **Block public access (bucket settings)**
5. Unchecks **Block all public access**

**Edit Block public access (bucket settings) Info**

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

6. Click Save & write **confirm** on the dialog.

7. The result should look like this

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, and this bucket and its access points. AWS recommends that you turn on Block all public access, to prevent public access to objects within, you can customize the individual settings below to suit your specific storage needs.

**Edit**

**Block all public access**

**⚠ Off**

► Individual Block Public Access settings for this bucket

8. Try to refresh the page, now the block public access setting is reverted to enable

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, and this bucket and its access points. AWS recommends that you turn on Block all public access, to prevent public access to objects within, you can customize the individual settings below to suit your specific storage needs.

**Edit**

**Block all public access**

**✔ On**

► Individual Block Public Access settings for this bucket

9. And on your previously configured email alert, you should receive notification

AWS Notifications <no-reply@sns.amazonaws.com>  
kepada saya ▾

11.16 (1 menit yang lalu) ☆ ↵ ⋮

Inggris ▾ > Indonesia ▾ Terjemahkan pesan Nonaktifkan untuk: Inggris ×

"Hi, there is a new S3 PutBucketPublicAccessBlock event at 2022-04-23T04:16:43Z which affected bucket s3-public-read-cloudsec-academy-9m8s and it is residing in ap-southeast-2 region"

...

### GuardDuty Notification (Optional)

As we configured GuardDuty notification earlier, if GuardDuty already detected some threat it will show up in your pre configured email (This notification delivery might take hours to arrive)

## 6. Cleanup Steps

1. Disable Cloudtrail
  - a. In the navigation pane, choose **Trails**, and then choose **security-trail**, choose **Stop logging** to turn off logging for the trail.
2. Delete Cloudtrail
  - a. Open the **Trails** page of the CloudTrail console
  - b. Choose **security-trail**
  - c. At the top of the trail details page, choose **Delete**
  - d. When you are prompted to confirm, choose **Delete** to delete the trail permanently. The trail is removed from the list of trails.
  - e. **Remainder** Log files that were already delivered to the Amazon S3 bucket are not deleted
3. Empty bucket object (if there's an object)
  - a. In the **Buckets** list, select the option next to the name of the bucket, and then choose **Empty** at the top of the page
  - b. On the **Empty bucket** page, confirm that you want to permanently delete all objects in the bucket by typing **permanently delete** into the text field, and then choose **Empty**.
4. Delete S3 Bucket for Log Trails
  - a. In the **Buckets** list, select the option next to the name of the bucket that you create for trail log, and then choose **Delete** at the top of the page.
  - b. On the **Delete bucket** page, confirm that you want to delete the bucket by entering the bucket name into the text field, and then choose **Delete bucket**.
  - c. To verify that you've deleted the bucket, open the **Buckets** list and enter the name of the bucket that you deleted. If the bucket can't be found, your deletion was successful.
5. Delete S3 Bucket for CF-Templates (*cf-templates-\**)
  - a. In the **Buckets** list, select the option next to the name of the bucket that you create for trail log, and then choose **Delete** at the top of the page.
  - b. On the **Delete bucket** page, confirm that you want to delete the bucket by entering the bucket name into the text field, and then choose **Delete bucket**.
  - c. To verify that you've deleted the bucket, open the **Buckets** list and enter the name of the bucket that you deleted. If the bucket can't be found, your deletion was successful.
6. Delete Cloudformation
  - a. On the **Stacks** page in the CloudFormation console, select **Tokopedia-Academy-April-2022**. The stack must be currently running

- b. In the stack details pane, choose **Delete**.
  - c. Select **Delete stack** when prompted.
7. Delete Eventbridge Rule
- a. Open the Amazon EventBridge
  - b. In the navigation pane, choose **Rules**.
  - c. Under **Event bus**, select the event bus that is associated with the rule (Default)
  - d. To delete a rule, select the button next to the rule and choose **Actions, Delete, Delete**.
  - e. Enter the name of the rule **ec2-event-trigger** to acknowledge that it is a managed rule and that deleting it may stop functionality in the service that created the rule. To continue, enter the rule name and choose **Force delete**
8. Delete Amazon SNS Topics and Subscriptions
- a. When you delete a topic, Amazon SNS deletes the subscriptions associated with the topic.
  - b. In the left navigation pane, choose **Topics**.
  - c. On the **Topics** page, select **ec2-and-s3-event-topic**, and then choose **Delete**
  - d. In the **Delete topic** dialog box, enter delete me, and then choose **Delete**. The console deletes the topic
9. Disable Guarduty
- a. Open the GuardDuty console
  - b. In the navigation pane, Choose **Settings**
  - c. Within the **Suspend GuardDuty** section, choose **Suspend GuardDuty** or **Disable GuardDuty**, then confirm your action in the next panel
10. Delete Lambda
- a. Go to AWS Lambda service. Find AWS lambda functions created.  
**ec2-security-group-remediation, s3-public-bucket-remediation**
  - b. Observe that there is a radio button across each of the AWS Lambda functions. Select the function.
  - c. The **Action** dropdown which was earlier grayed out is highlighted now. Now, open the combo box.
  - d. Select the **Delete** button to delete the AWS Lambda function.
  - e. **Reminder** deleting lambda will not delete the role linked, therefore also need to delete the lambda role.
11. Switch Account -> Root
12. Delete Role (User & Lambda role)
- a. In the navigation pane, choose **Roles**, and then select the check box next to the role name that you want to delete **auto-remediation-lambda-role**
  - b. At the top of the page, choose **Delete**.

- c. In the confirmation dialog box, review the last accessed information, which shows when each of the selected roles last accessed an AWS service. This helps you to confirm if the role is currently active. If you want to proceed, enter the name of the role in the text input field and choose **Delete**. If you are sure, you can proceed with the deletion even if the last accessed information is still loading.
13. Delete User that created earlier
- a. In the navigation pane, choose **Users**, and then select the check box next to the username **workshop-iamuser1** to delete.
  - b. At the top of the page, choose **Delete**.
  - c. In the confirmation dialog box, enter the username **workshop-iamuser1** in the text input field to confirm the deletion of the user. Choose **Delete**.

## 7. Improvements That Can Be made

From our workshop session, below are some improvements that surely will increase resilient of Threat Detection & Response.

1. Make the notification that being sent to receiver more informational and actionable
2. Create separated Lambda function to check existing resource that were misconfigured
3. Exploring another auto-remediation for other services
4. Auto-remediate the security group when it detects 0.0.0.0/0 on port 22 – currently it is not supported

## References

- <https://docs.aws.amazon.com/guardduty/>
- <https://docs.aws.amazon.com/AWSCloudFormation/>
- <https://docs.aws.amazon.com/ec2/>
- <https://cloud.google.com/compute>
- <https://www.alibabacloud.com/product/ecs>
- <https://docs.aws.amazon.com/iam/>
- <https://cloud.google.com/iam>
- <https://www.alibabacloud.com/product/ram>
- <https://aws.amazon.com/what-is-cloud-computing/>
- <https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cloud-security/>
- <https://aws.amazon.com/compliance/shared-responsibility-model/>
- <https://www.crowdstrike.com/cybersecurity-101/cloud-security/>
- <https://www.rapid7.com/fundamentals/threat-detection/#:~:text=Threat%20detection%20is%20the%20practice,can%20exploit%20any%20present%20vulnerabilities.>
- <https://aws.amazon.com/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc>
- <https://aws.amazon.com/lambda/>