**Tokopedia Workshop**
Apr 23, 2022

# Threats Detection & Response in Public Cloud Infrastructure

**By Cloud Security Team in Tokopedia**

# 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.

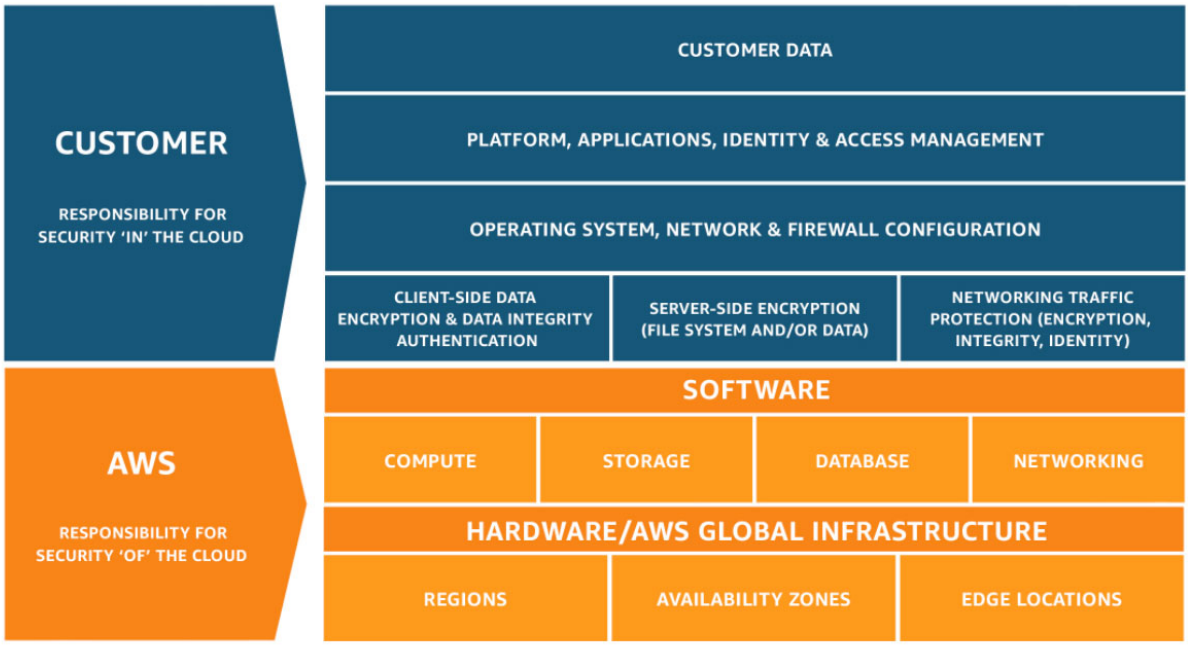
## 1.2 Cloud Security

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 alot 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

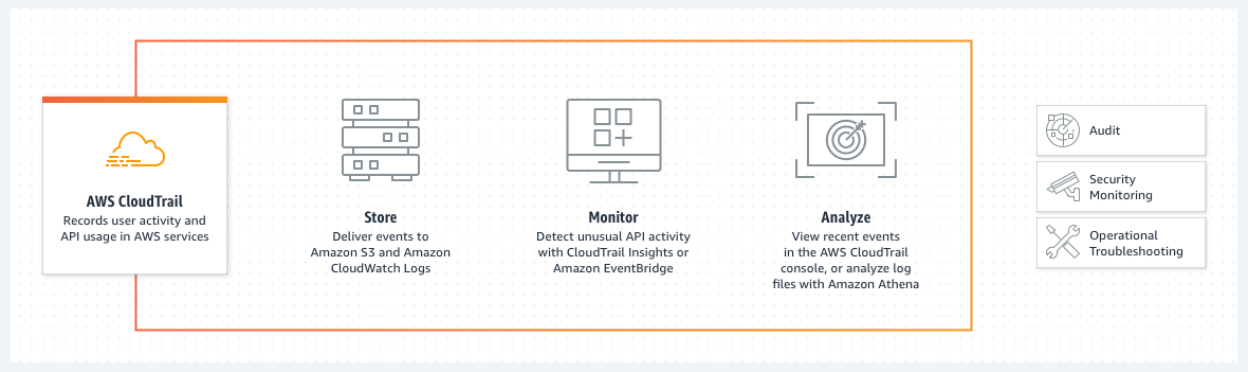
Threat detection is the practice of analyzing the entirety of a security ecosystem to identify any malicious activity that could compromise the network. If a threat is detected, then mitigation efforts must be enacted to properly neutralize the threat before it can exploit any present vulnerabilities.

## 2.1 Threat Detection & Response in AWS

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

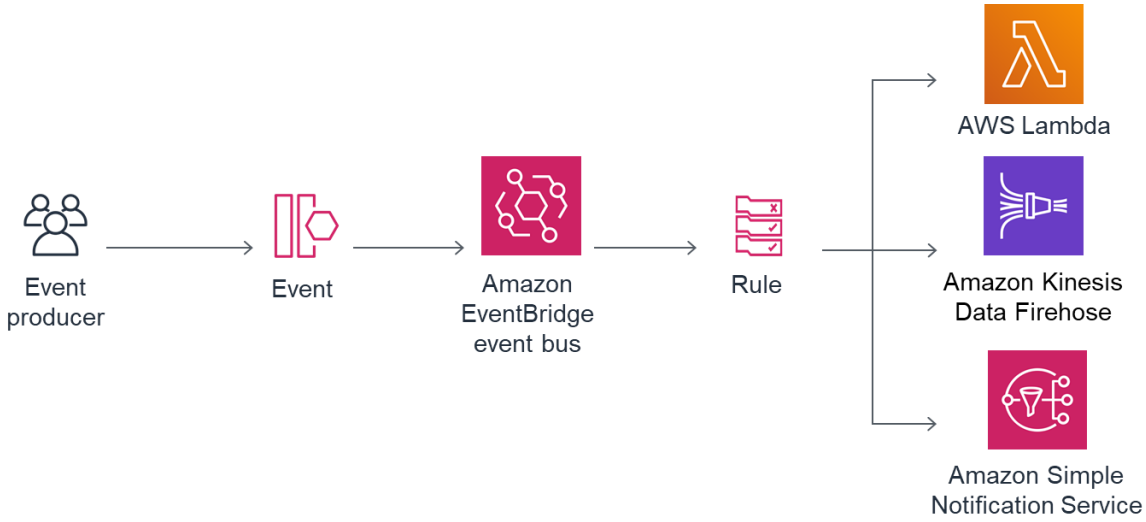
### AWS CloudTrail

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



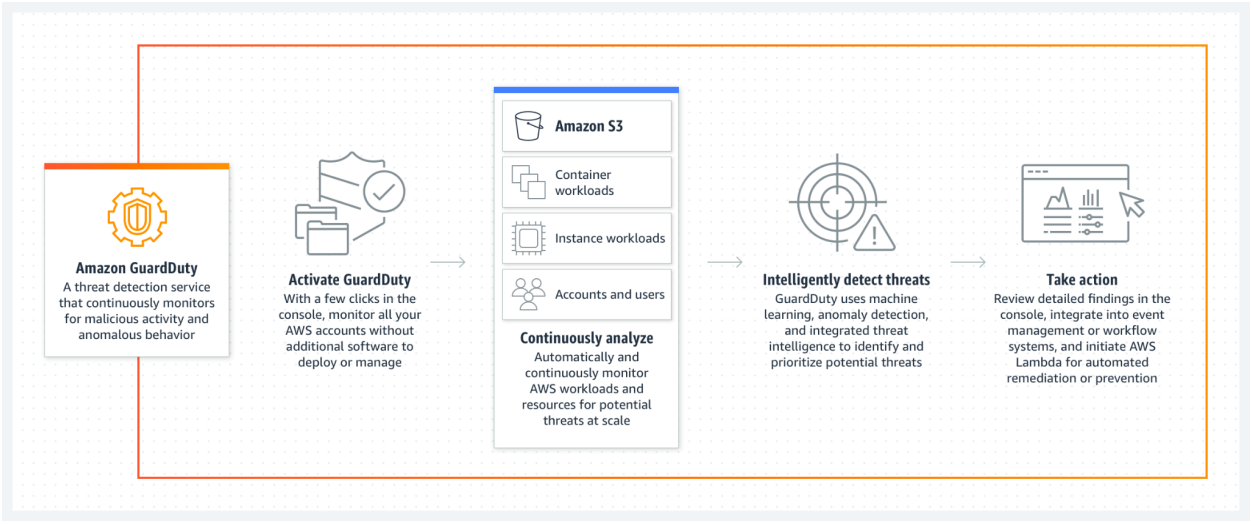

Using Cloudtrail we can deliver events to a specific location, this is of course very important for threat detection, we can create monitoring and detection from specific events that we gathered, e.g. Audit activity on a certain region or activity on a certain AWS service.

## 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.

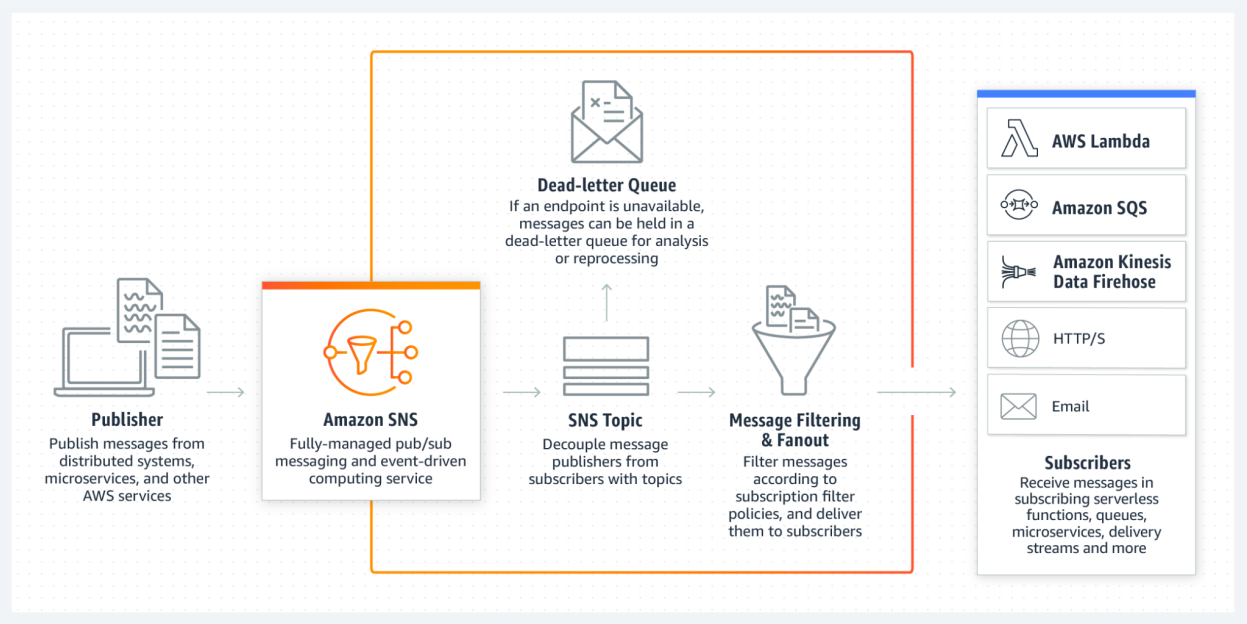
## AWS GuardDuty




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

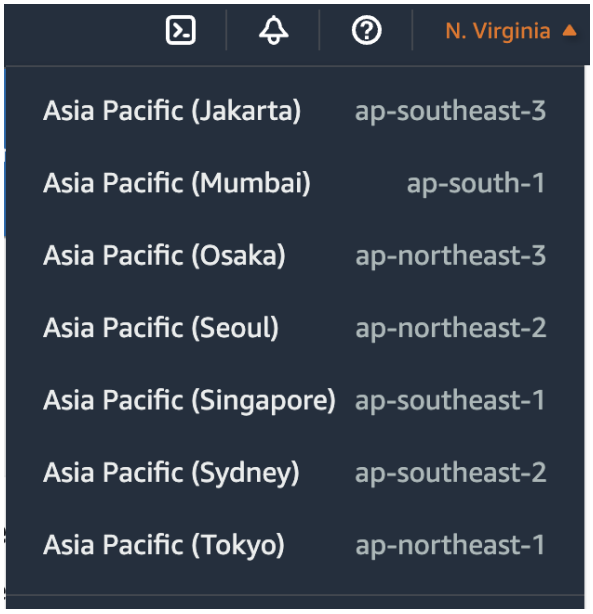
AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. In this workshop we will use Lambda as a threat response mechanism to remediate security misconfiguration or vulnerability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

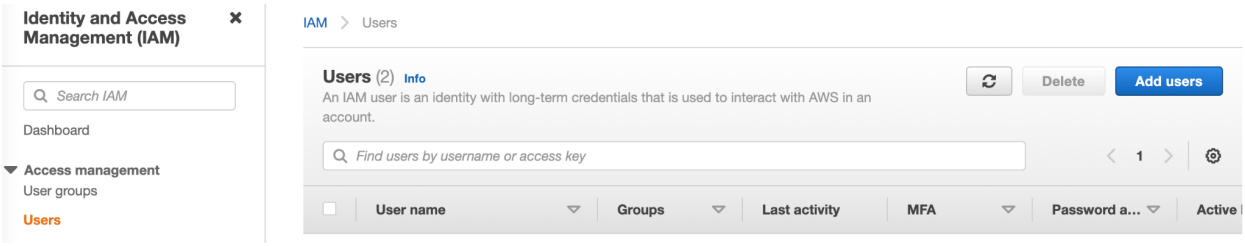



### 2.2 Creating an IAM user in your AWS account

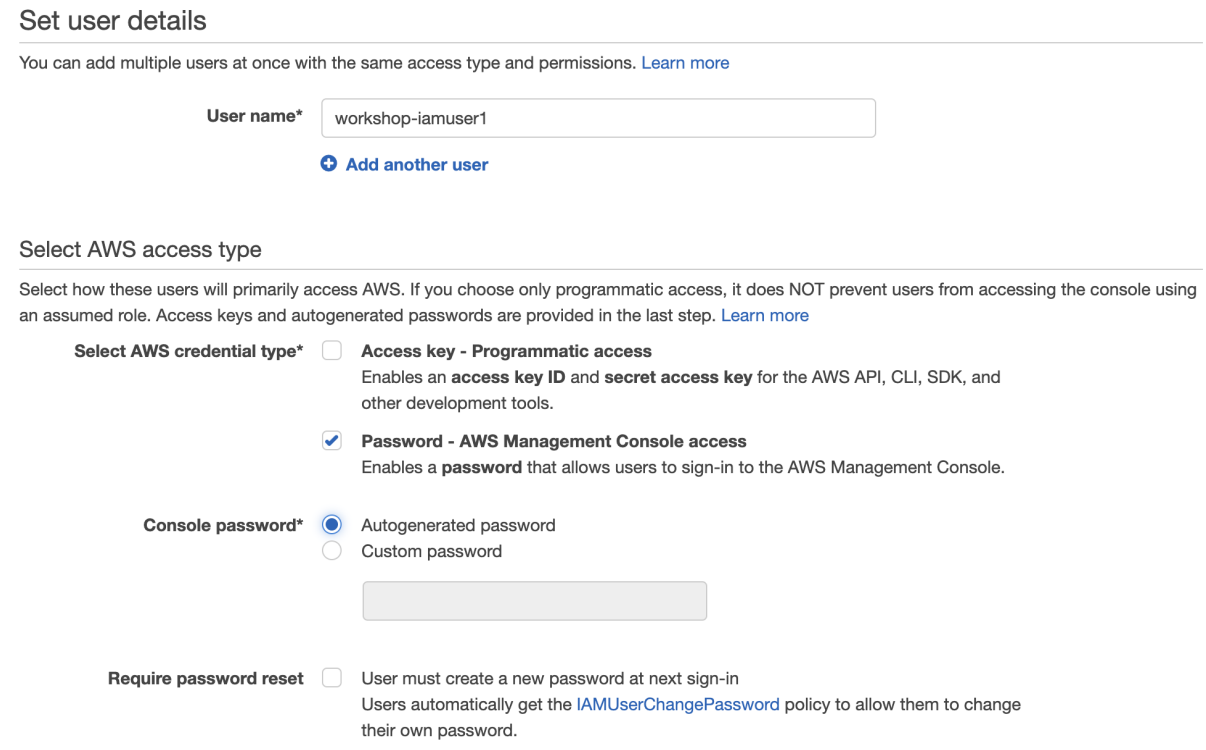
AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources. When you first create an AWS account, you begin with a single sign-in identity 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/.

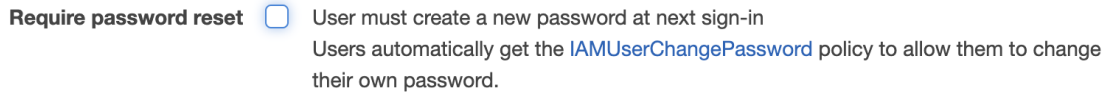
1. In the navigation pane, choose **Users** and then choose **Add users**.




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.**




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

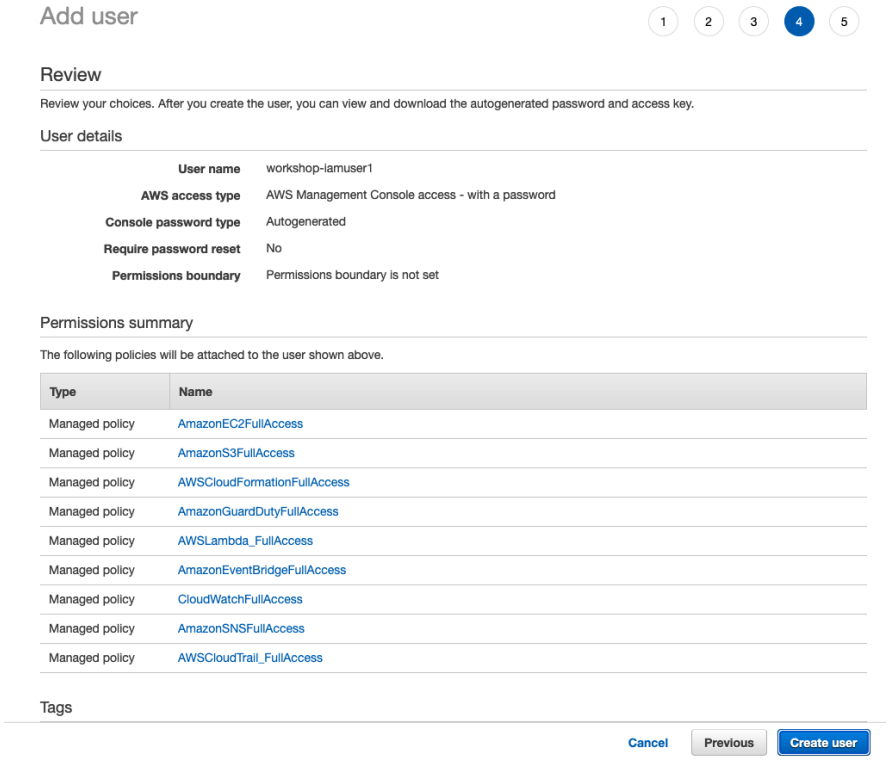



6. Choose **Next: Permissions.**

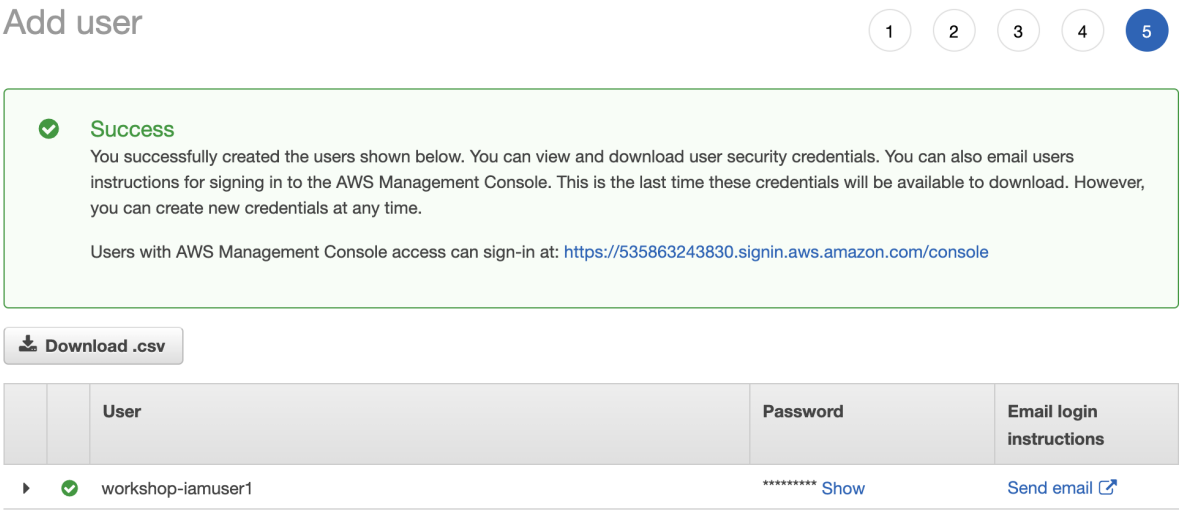
7. On the Set permissions page, choose **Attach existing policies directly** and pick policy below:

- 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.

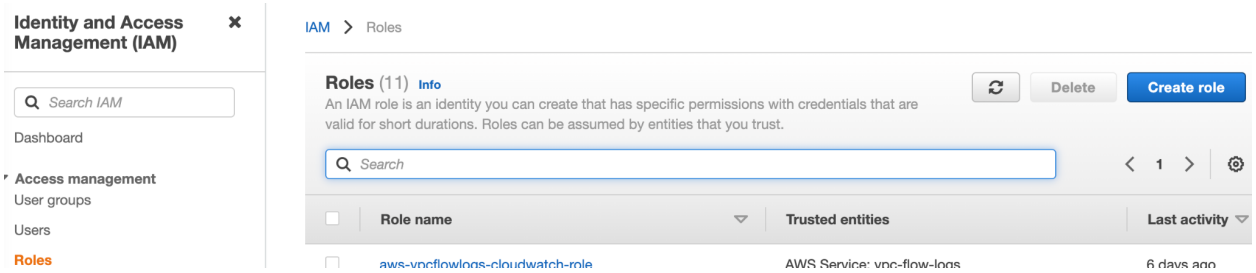



11. Choose: **Create User**
12. After user creation is completed, click at **Download .csv** link to get credential from previously created user and keep it somewhere safe.



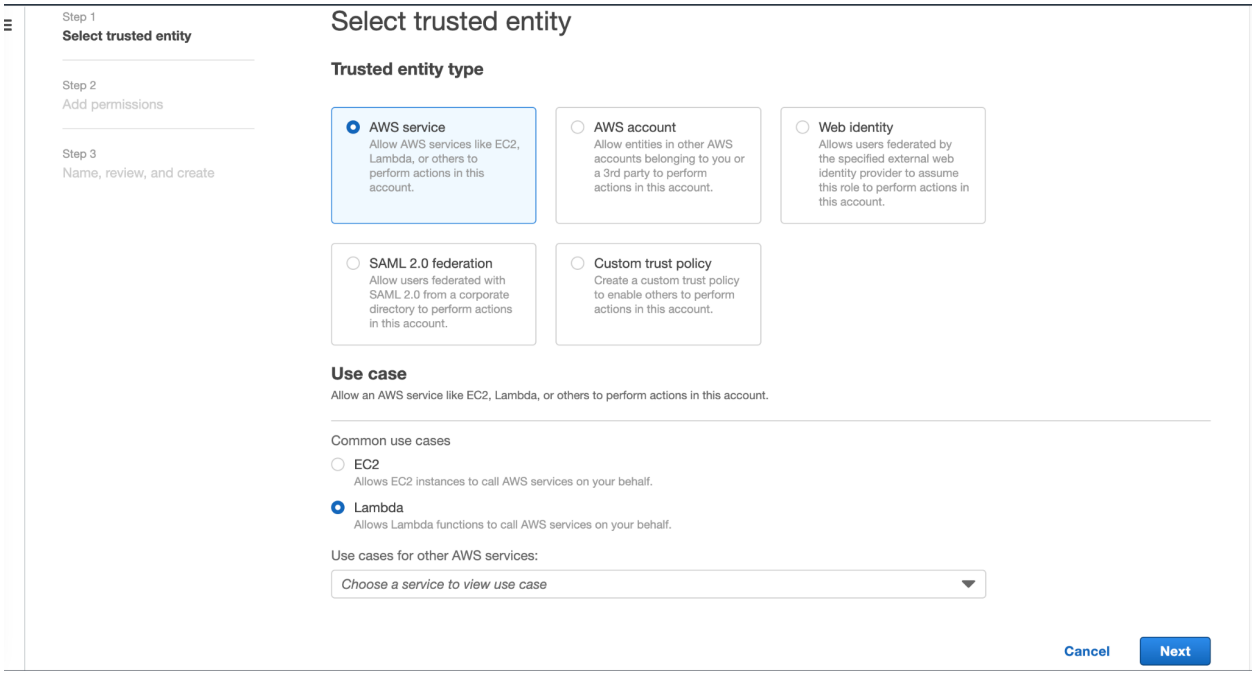

## 2.3 Create AWS Role For Auto-remediation Tools

1. Go to IAM menu > Roles > Create role

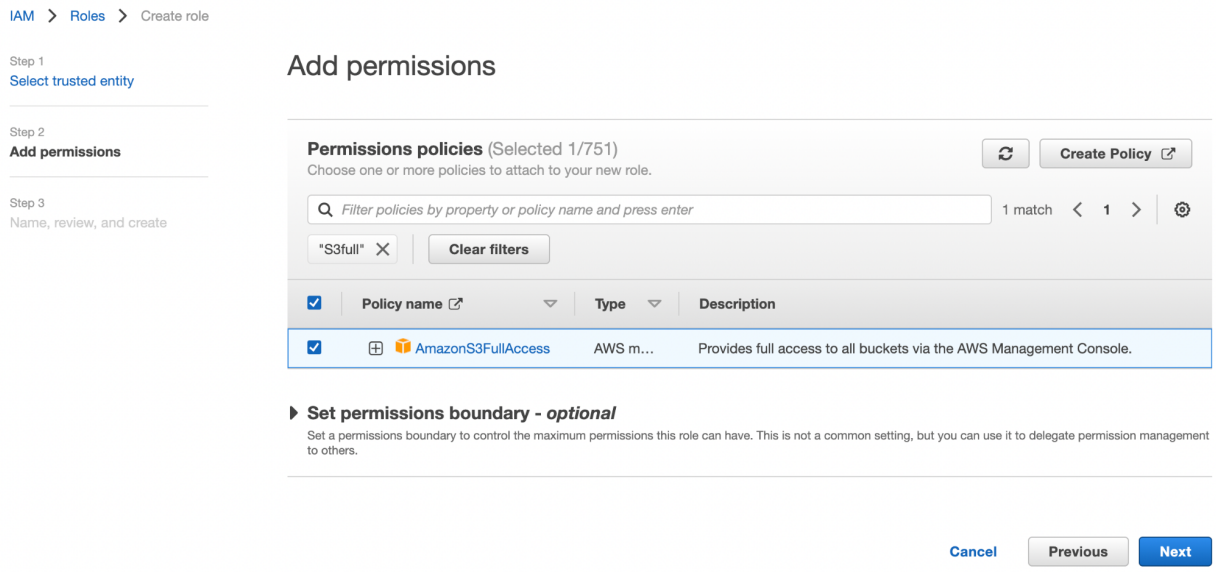


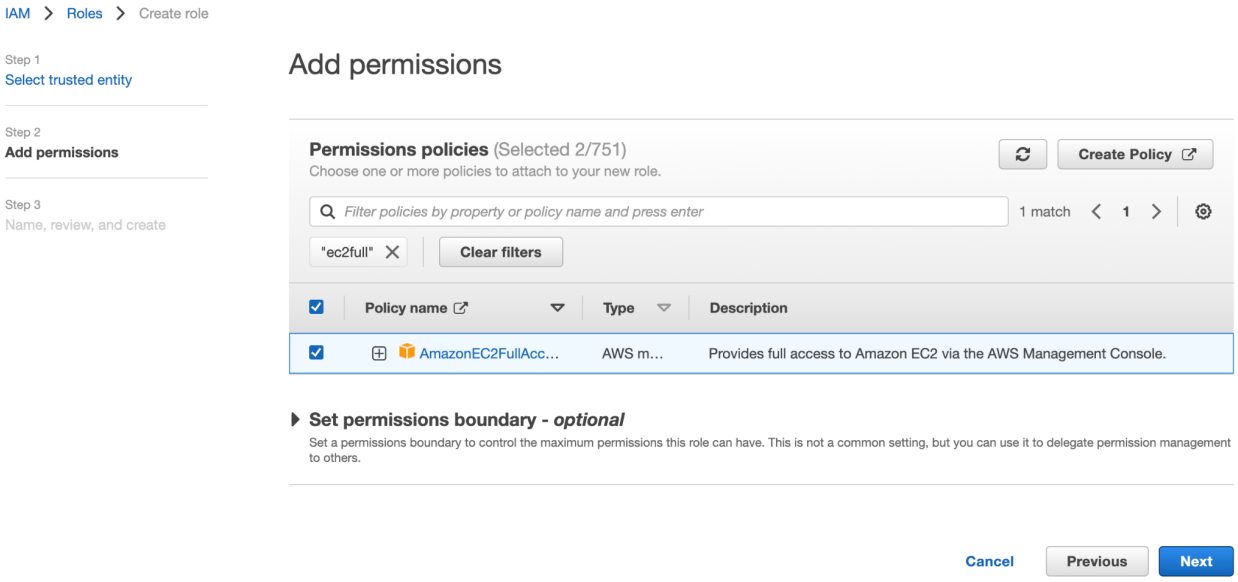

2. In Step 1, Select trusted entity

- **Trusted entity type:** AWS service
- **Use case:** Lambda

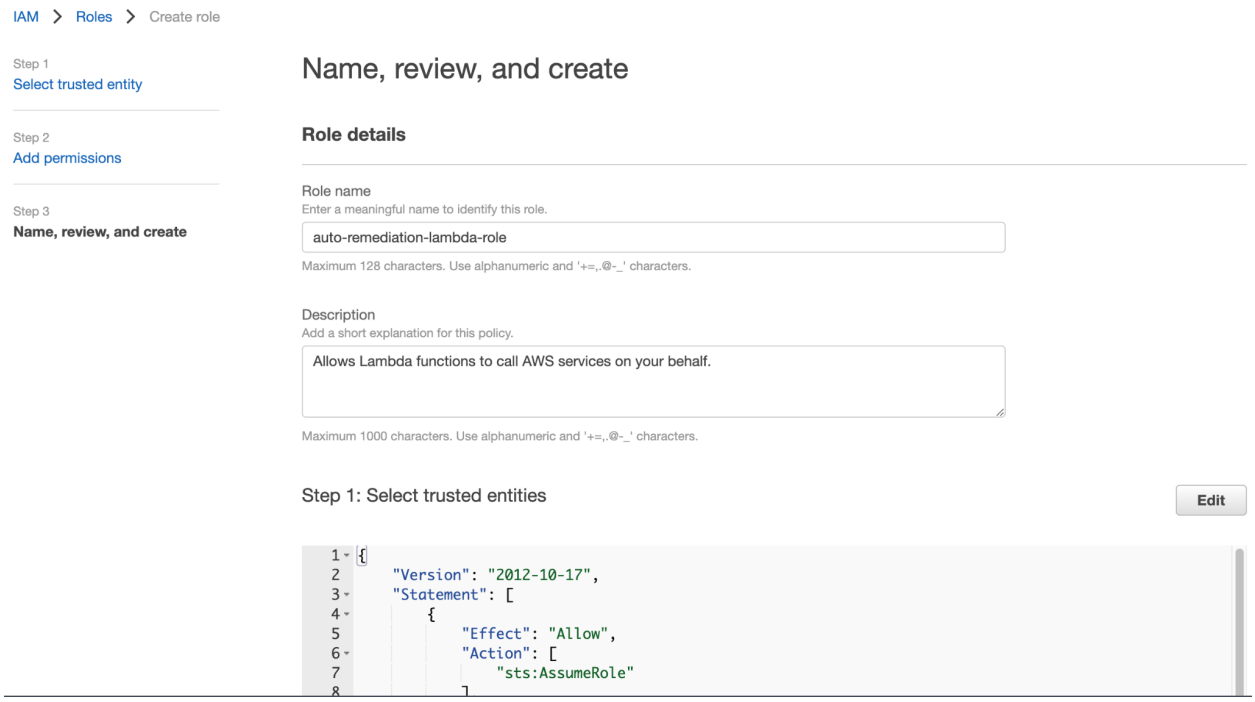



3. Click Next
4. In Step 2, Add permissions – select the following policies:
   - AmazonEC2FullAccess
   - AmazonS3FullAccess






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

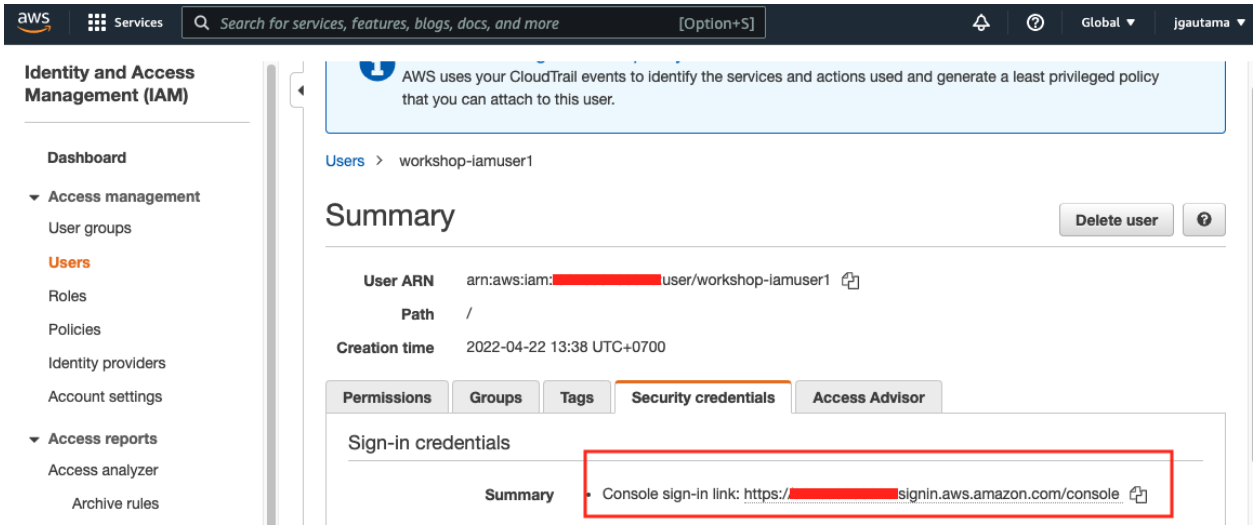


## 2.4 Switch To Previously Created IAM User

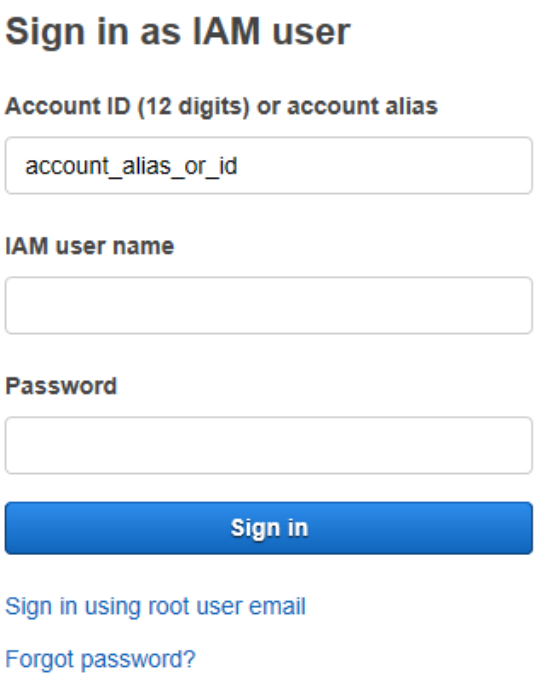
To sign in to an AWS account as an IAM user using an IAM users you will need credentials and IAM Users sign-in URL, you can find this on **.csv file** that was previously downloaded in **Step 2.2**

| 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)**



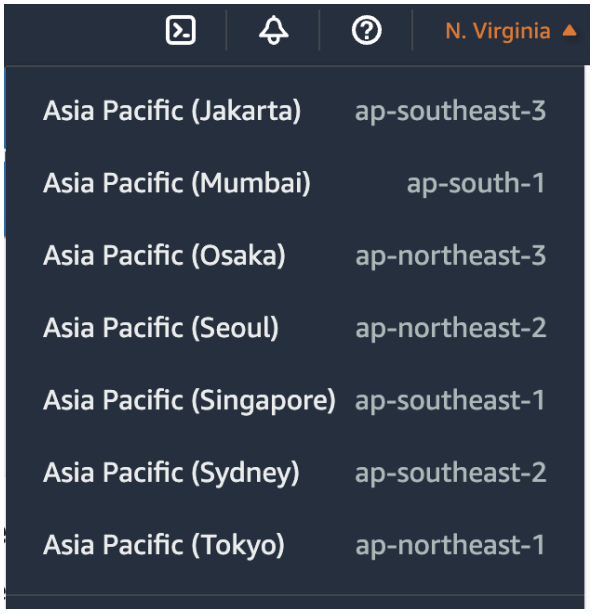

Click or open the url that was specified inside **Console Login Link** and fill up the form with **Username** and **Password** that specified in the CSV file and click **Sign In**.



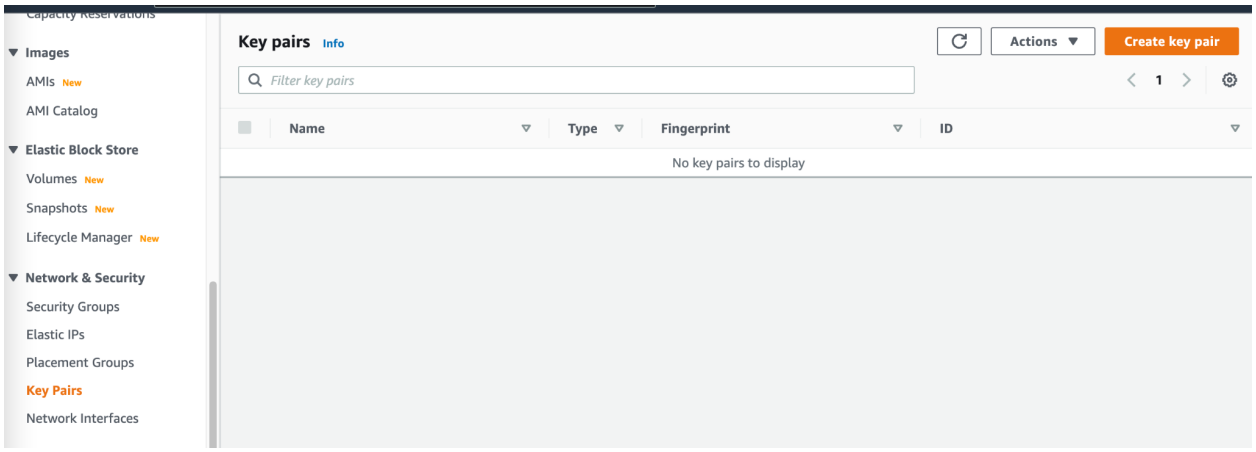

## 2.5 Create EC2 SSH Key Pair

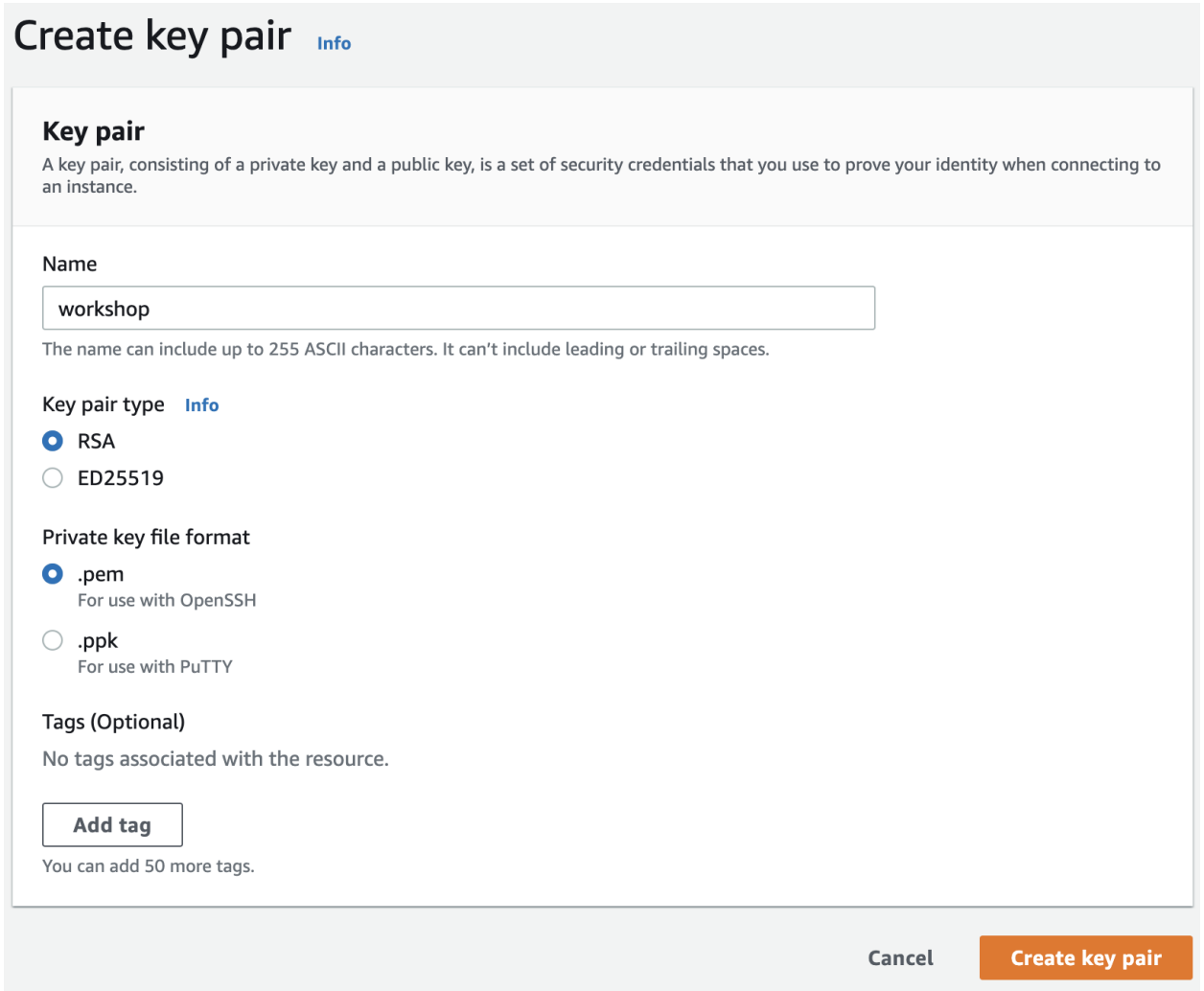
If you do not already have one, create a new key pair in the Amazon EC2 console for each AWS Region where you plan to receive data. Use the steps below.

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.



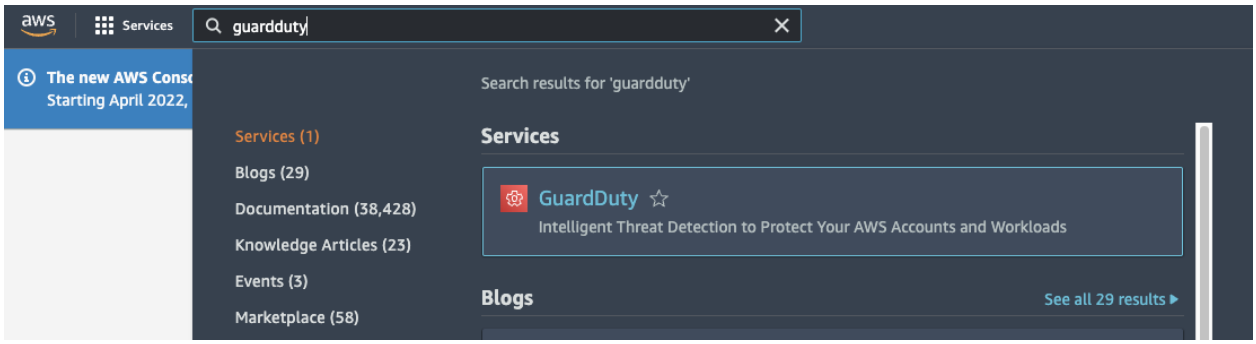
3. Enter a friendly name like **workshop-toko-academy-key**

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`).

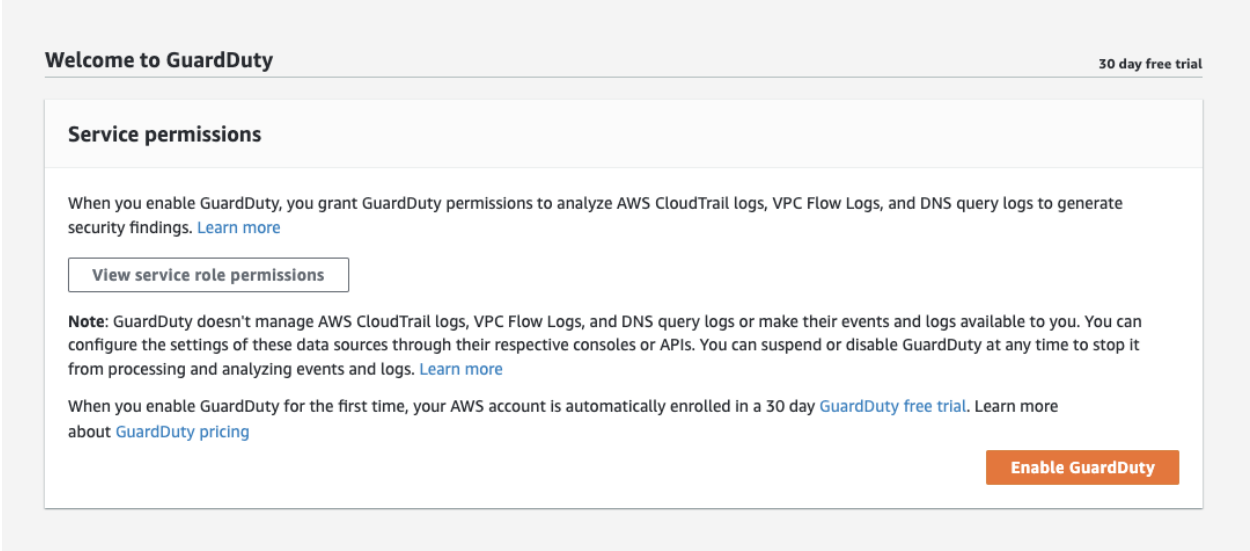
## 2.6 Amazon GuardDuty

For this workshop, let's enable Amazon GuardDuty.

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




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

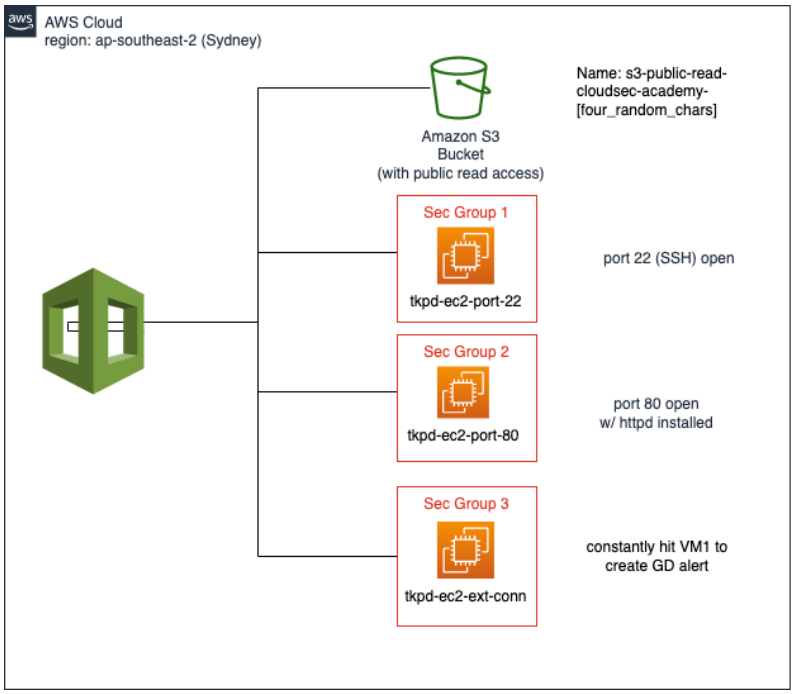
A security group acts as a virtual firewall, controlling the traffic that is allowed to reach and leave the resources that it is associated with. For example, after you associate a security group with an EC2 instance, it controls the inbound and outbound traffic for the instance.

### 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:

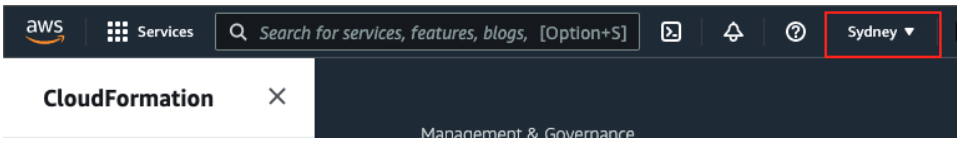



1. 1x S3 Bucket where the Bucket Policy is configured to be public ("**s3-public-read-cloudsec-academy-[four_rand_char]**")
2. 3x EC2 instances with port 80 & 22 open, and Test VM
    a. **tkpd-ec2-port-22** EC2 Instance port 22 open
    b. **tkpd-ec2-port-80** EC2 Instance that runs https that display 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

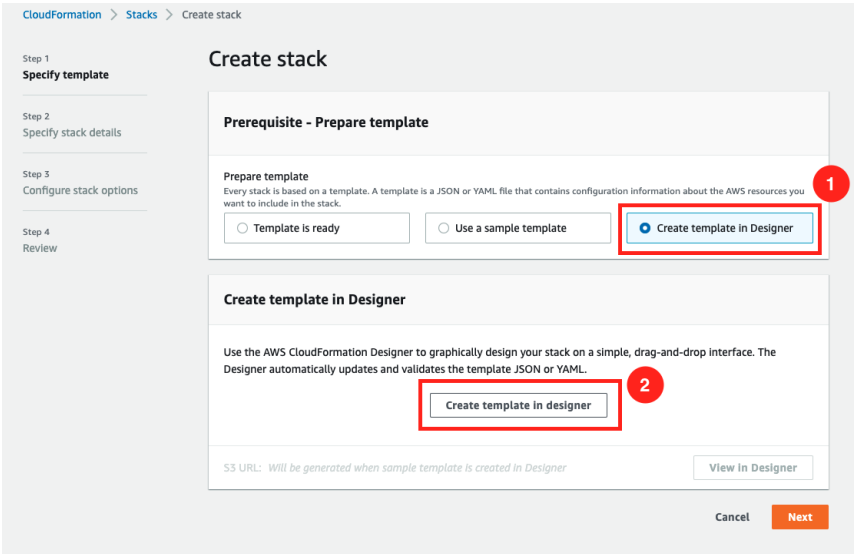
**Starting the Create Stack wizard**

**To create a stack on the CloudFormation console**

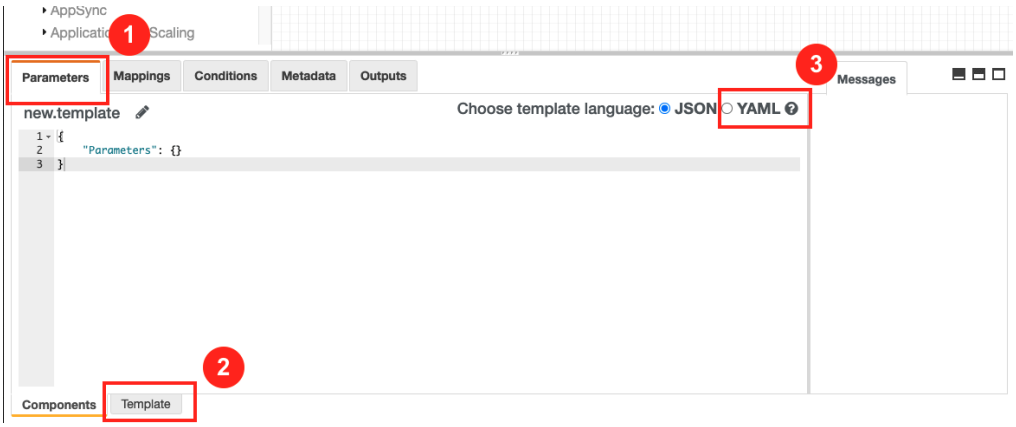
1. Open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation.
2. Make sure you have selected ap-southeast-2 (Sydney) region



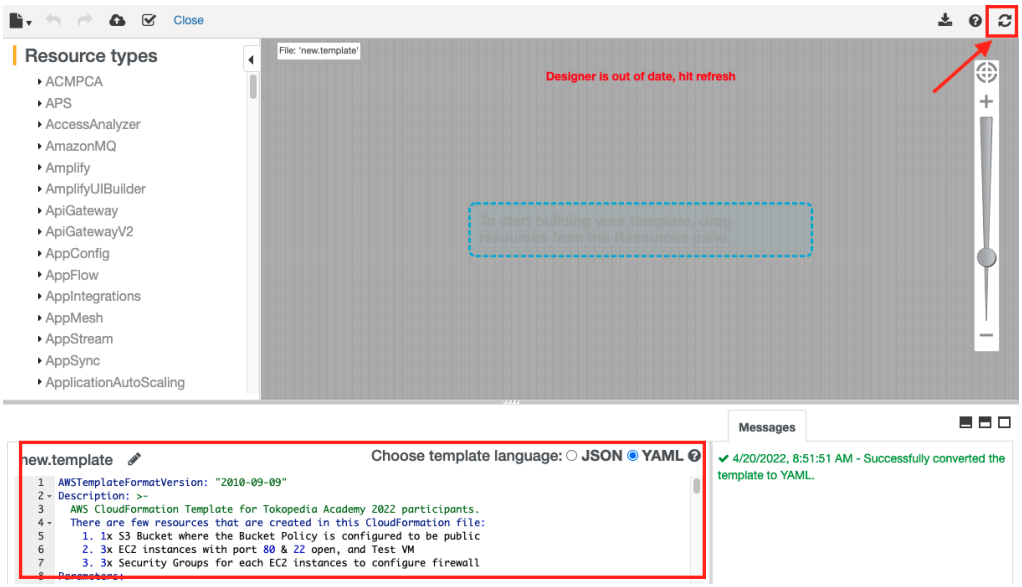

3. Create a new stack by using one of the following options:
   - Choose **Create Stack** > **With new resources (standard)**
4. Select “Create template in Designer”(1) and click “Create template in designer”(2), which will redirect you to a *Designer** page.



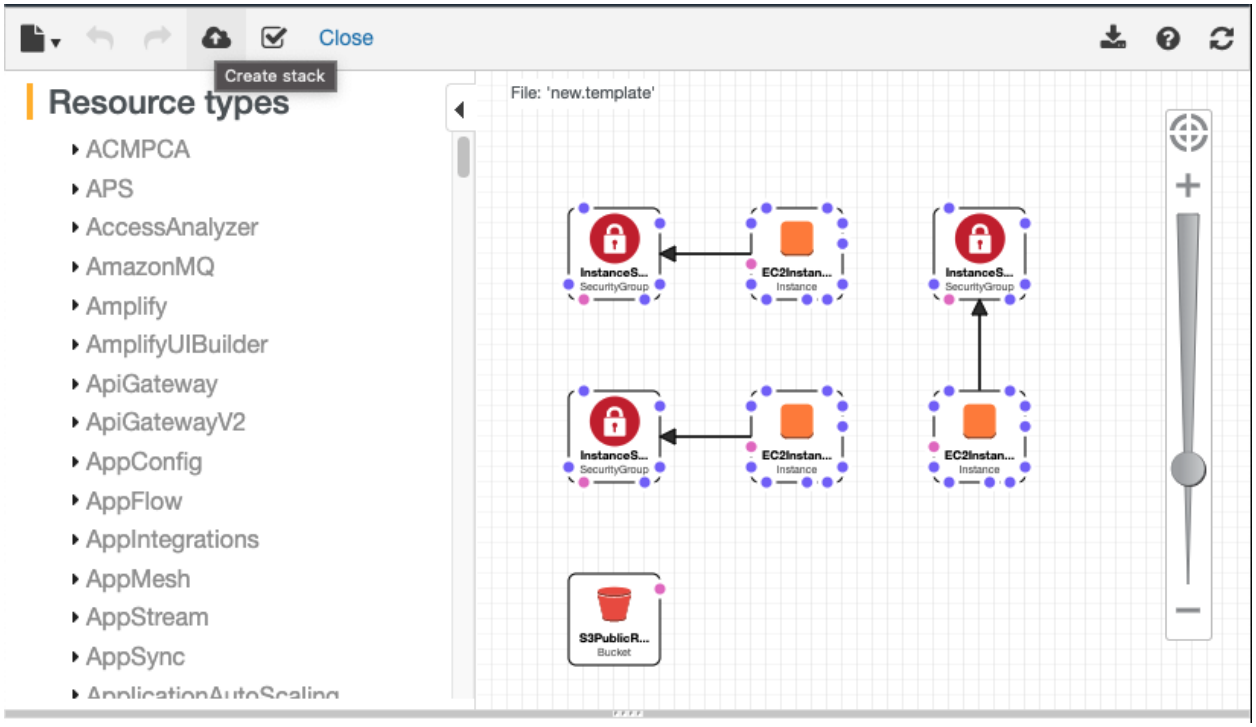

5. Select Parameters (1) > Template (2) > choose YAML (3)



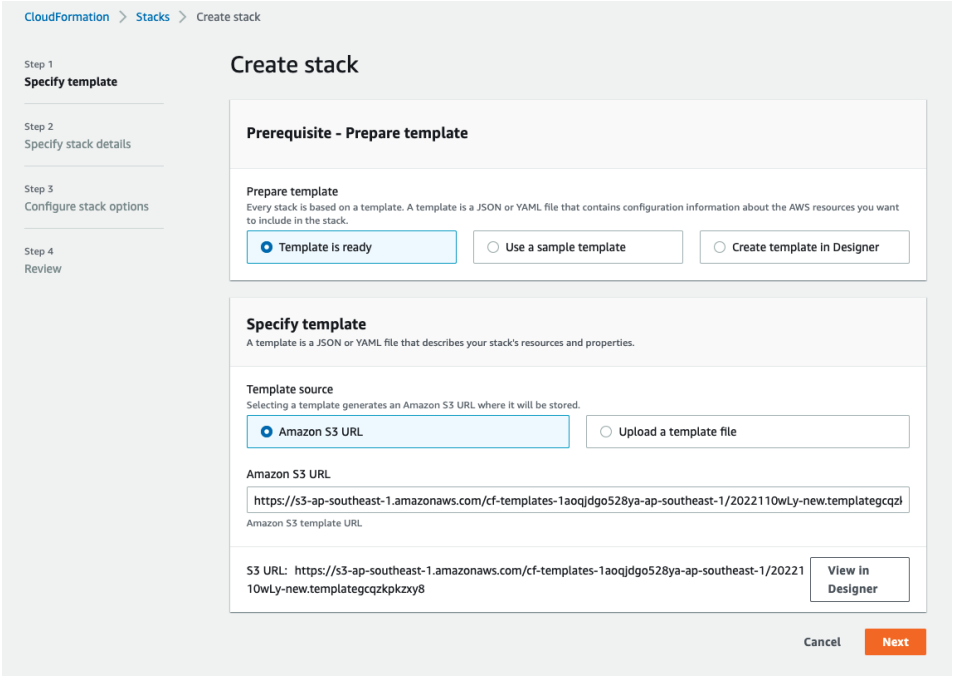
6. Copy and paste the *main.yaml* code from the [github repository] to the inline code feature of *Designer* then hit refresh on the top-right corner



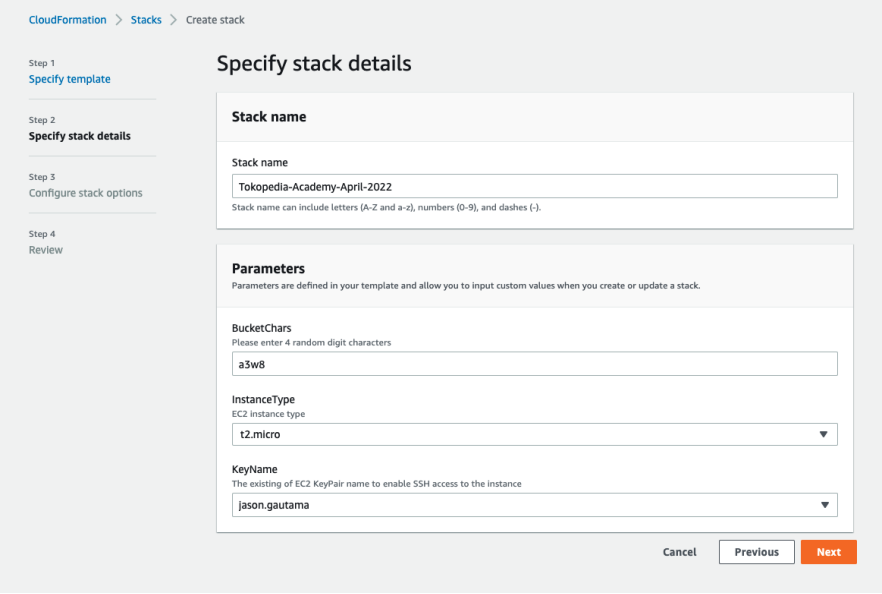

7. You will see the visual diagram of the to-be created resource. Once you have verified that the resources are correct,click the *Create stack* ***button*** (⛅)




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”

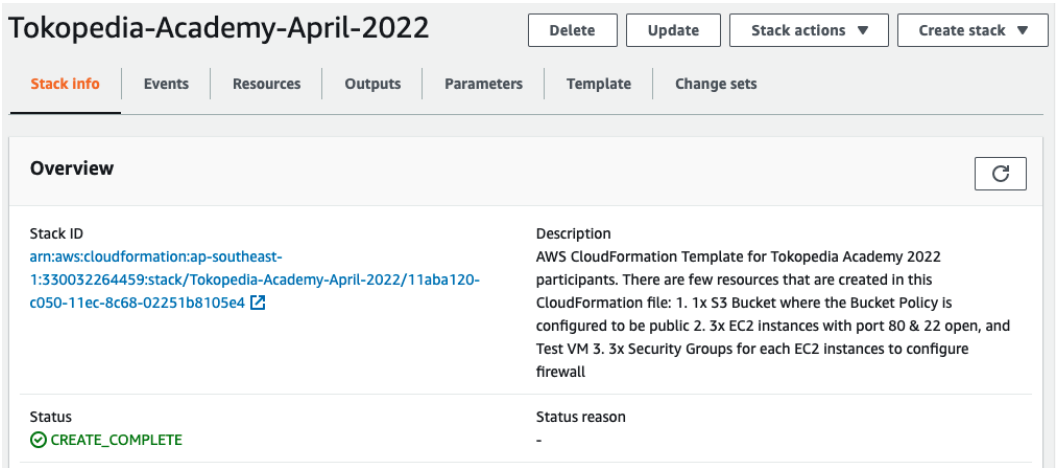



9. Follow the prompts
   - Stack name : **Tokopedia-Academy-April-2022**
   - BucketChars : enter 4 random characters (eg. a3w8)
   - InstanceType : **t2.micro**
   - KeyName : Select the KeyPair that you have created in step 1.4 Create EC2 SSH Key Pair




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

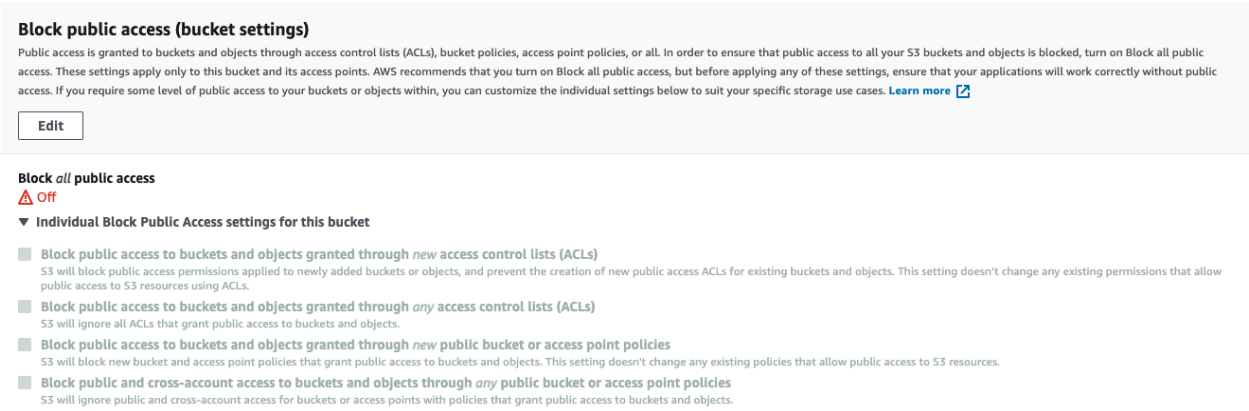



**Note: AWS CloudFormation Designer* (Designer) is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor.

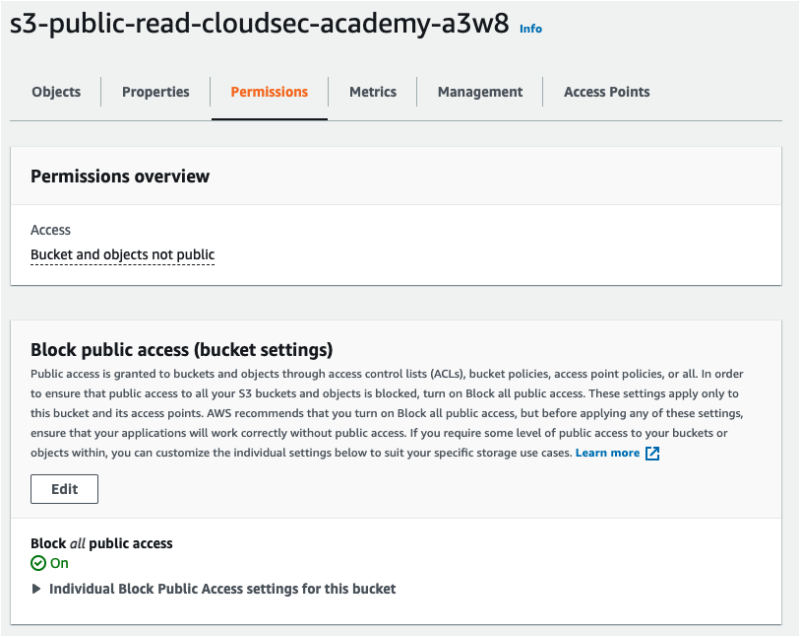
We have successfully created the resources using CloudFormation! **Kudos to you!**
Now **let's explore <u>each newly created resources</u>** and **how to remediate them**

## Restrict S3 Bucket Access Control List (ACL)

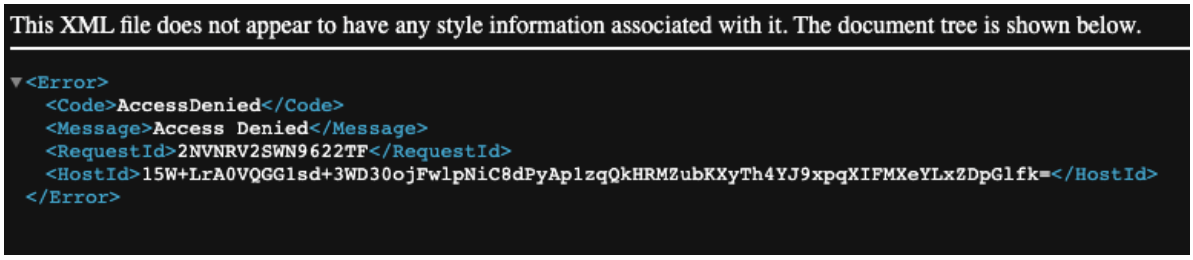
- Search for "S3" on the AWS console search bar or [https://s3.console.aws.amazon.com/](https://s3.console.aws.amazon.com/)
- Find buckets by name "s3-public-read-cloudsec-academy" > click the bucket name
- Go to Permissions > Block public access (bucket settings). You will see that **Block *all* public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



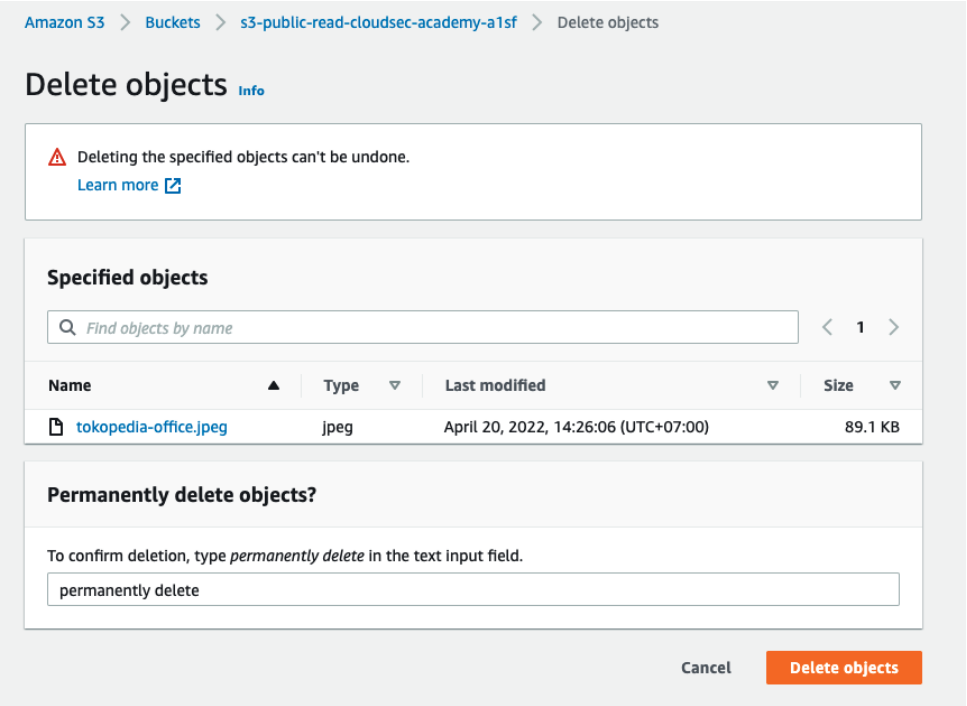
- Before we remediate, let's test if the bucket is open to the public by **uploading a file** and making it public.
    - Select "Upload" > Add files > *"upload any pictures"*
    - Expand "Permissions" > Predefined ACLs > Grant public-read access > tick the box "I understand the risk of granting public-read 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.




- 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!**

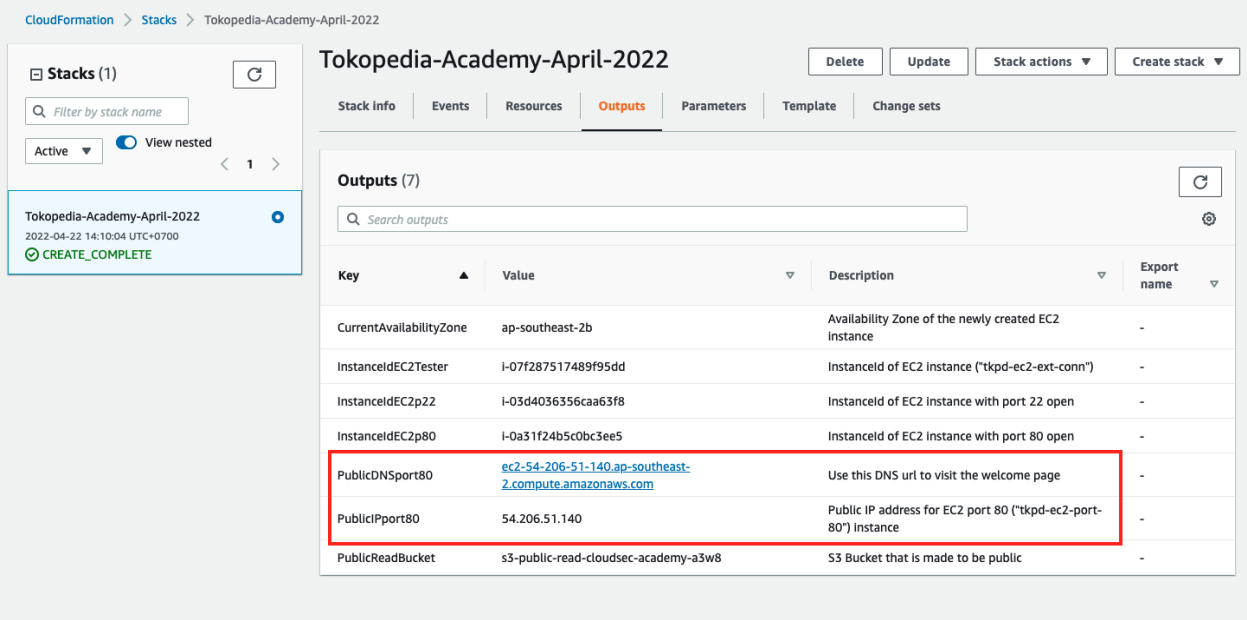



## Restrict EC2 Instance Firewall (aka Security Group)

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.**

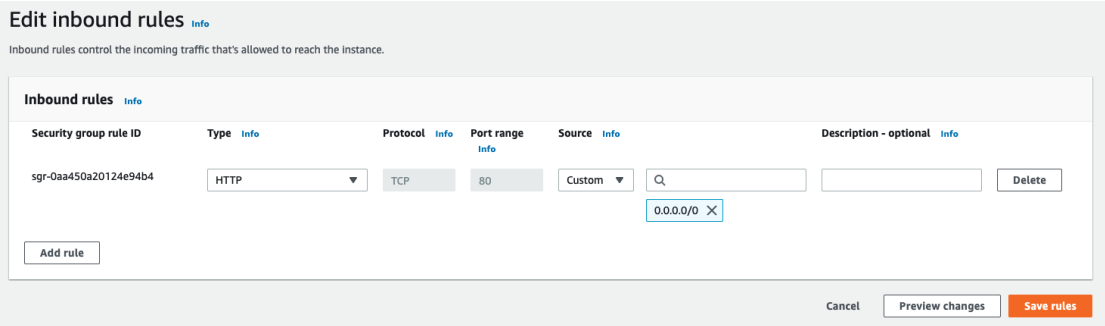
To find the IP address / DNS of **tkpd-ec2-port-80**, you can go to the created CloudFormation stacks > Tokopedia-Academy-April-2022 > Outputs > and find **PublicDNSport80** or **PublicIPport80.**



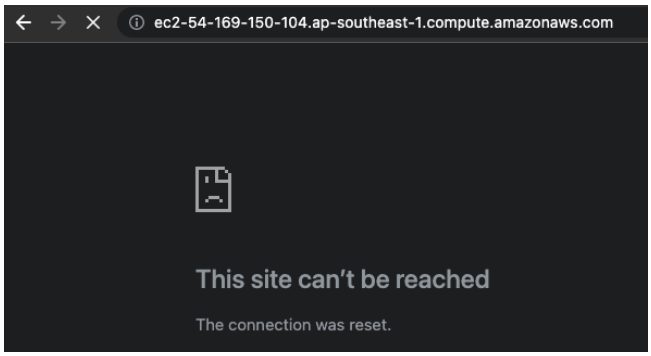

If you received a "This site can't be reached" error, make sure that you put **http://** in front of the IP address.


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




4. Verify that you are no longer to view the page that you open




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.

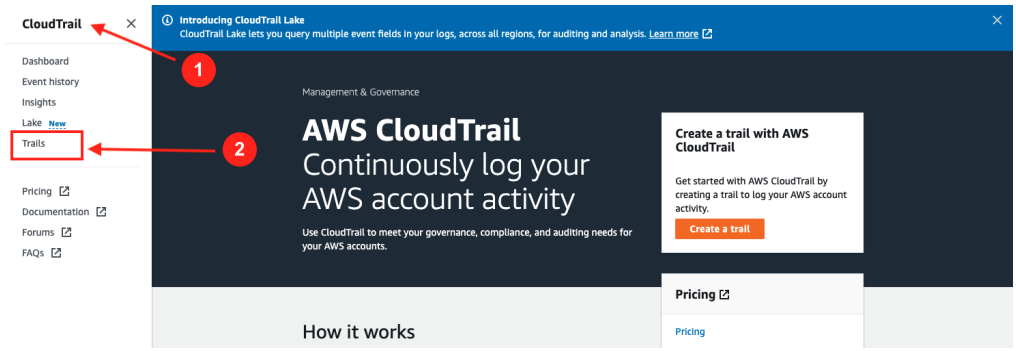
# 3. Threat Detection Kit

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

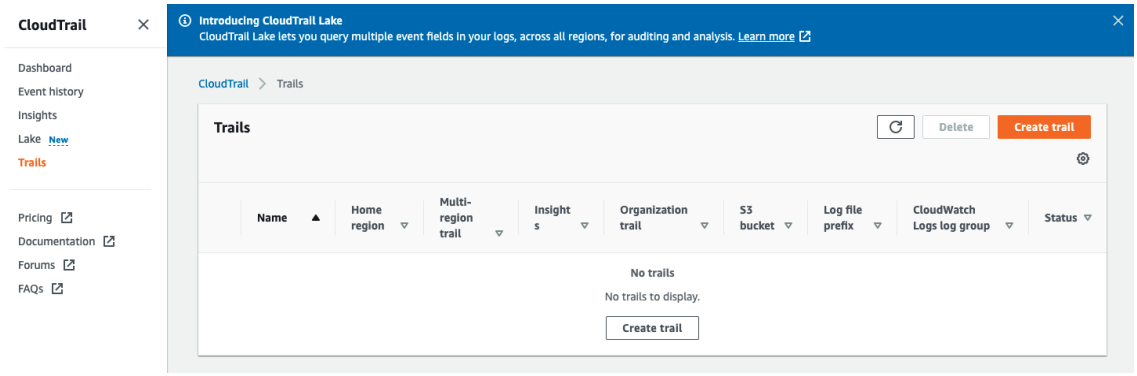
## 3.1 Amazon CloudTrail

On this following section, you will enable AWS CloudTrail

1. Go to the search bar and type “CloudTrail”
2. Expand the  sandwich bar and click **Trails**




3. Click **Create Trail**



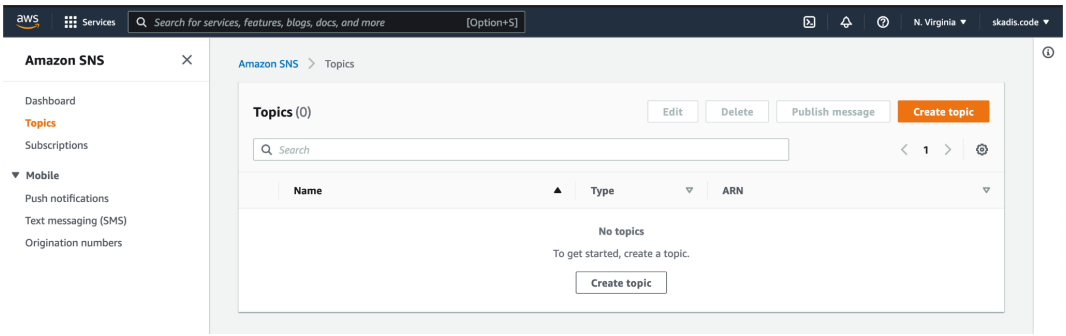

4. Fill the necessary field
    a. **Trail Name:** security-trail
    b. **Storage Location** -> Create new S3 bucket
    c. **Log file SSE-KMS encryption** -> Un-select it
    d. Leave the other fields as its default, then click next
    e. **Event type** -> Select the **Management Events**
    f. Scroll down to the bottom and click next
5. Review your config and it will look like this
6. Choose create trail
7. 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

1. Topics Part
    1.1. On the search bar, type “SNS”

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




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**

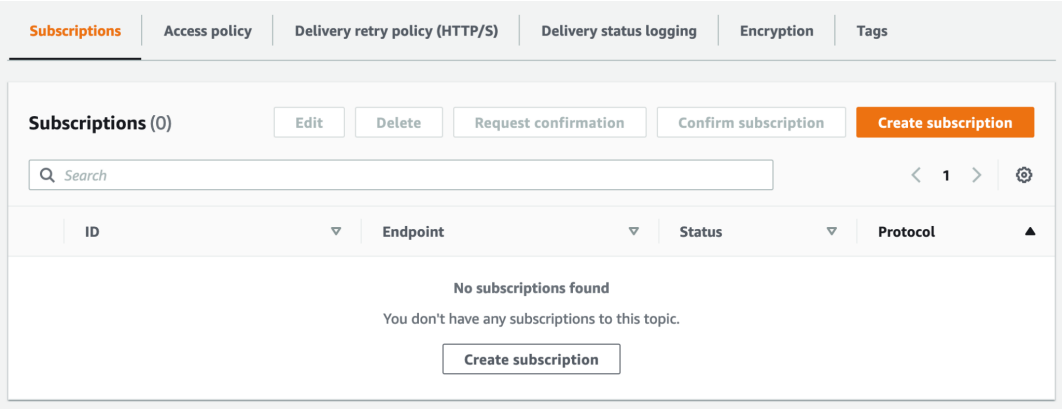
1.6. For the rest, you can leave it as default value

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”



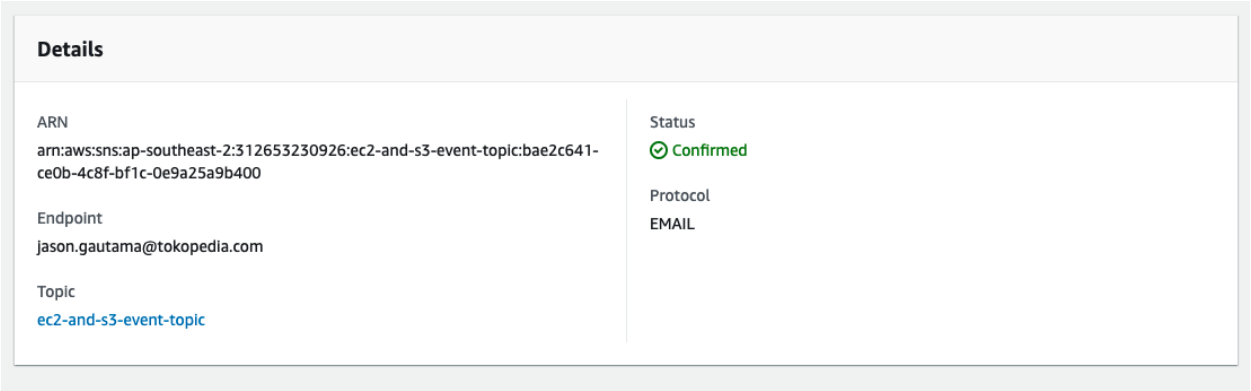

2.3. On the protocol field, choose “Email”

2.4. Fill the endpoint with your email that will receive the notification

2.5. Next up is clicking “Create subscription”

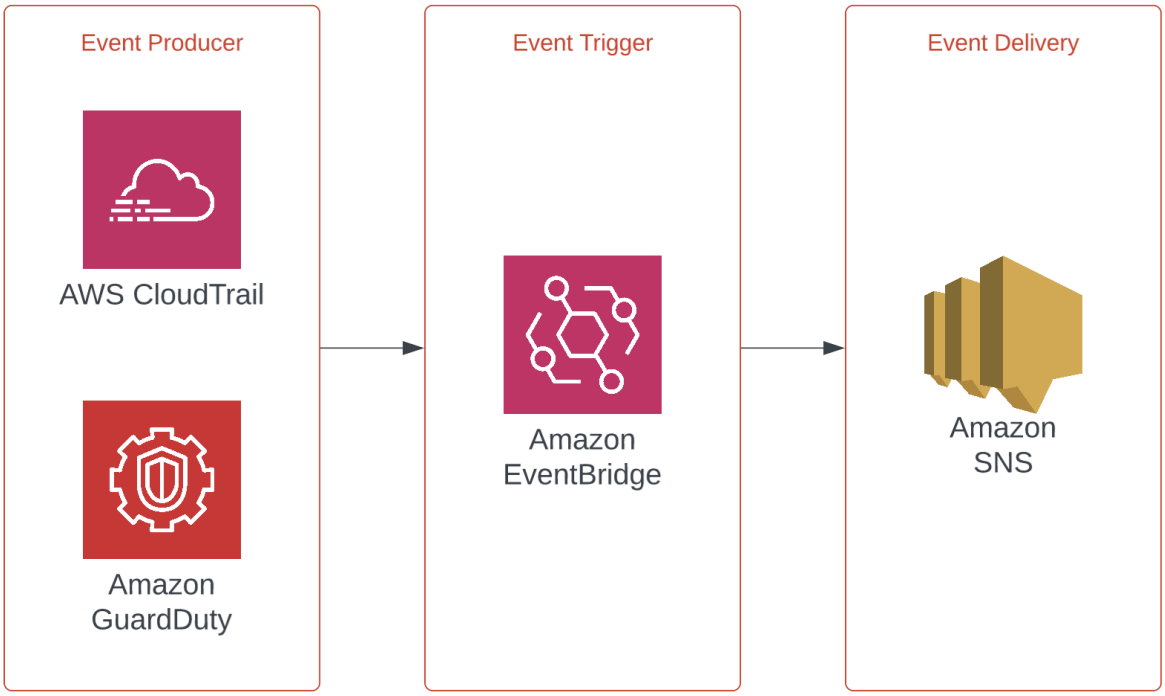
2.6. Now, check your mail and confirm the message by clicking the sent URL

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




## 3.3 EventBridge

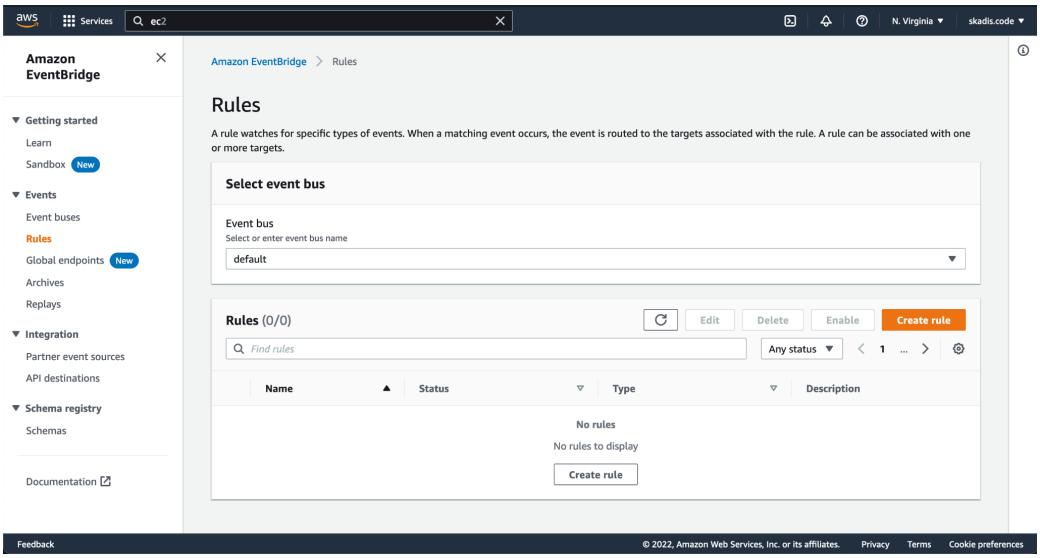
Now we are arriving at the last configuration for the Threat Detection Kit. A bit summary, we have had CloudTrail and GuardDuty as the event producer, followed by Amazon SNS as the event delivery. **EventBridge will be residing between event producer and event delivery as an event trigger.** Here is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




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

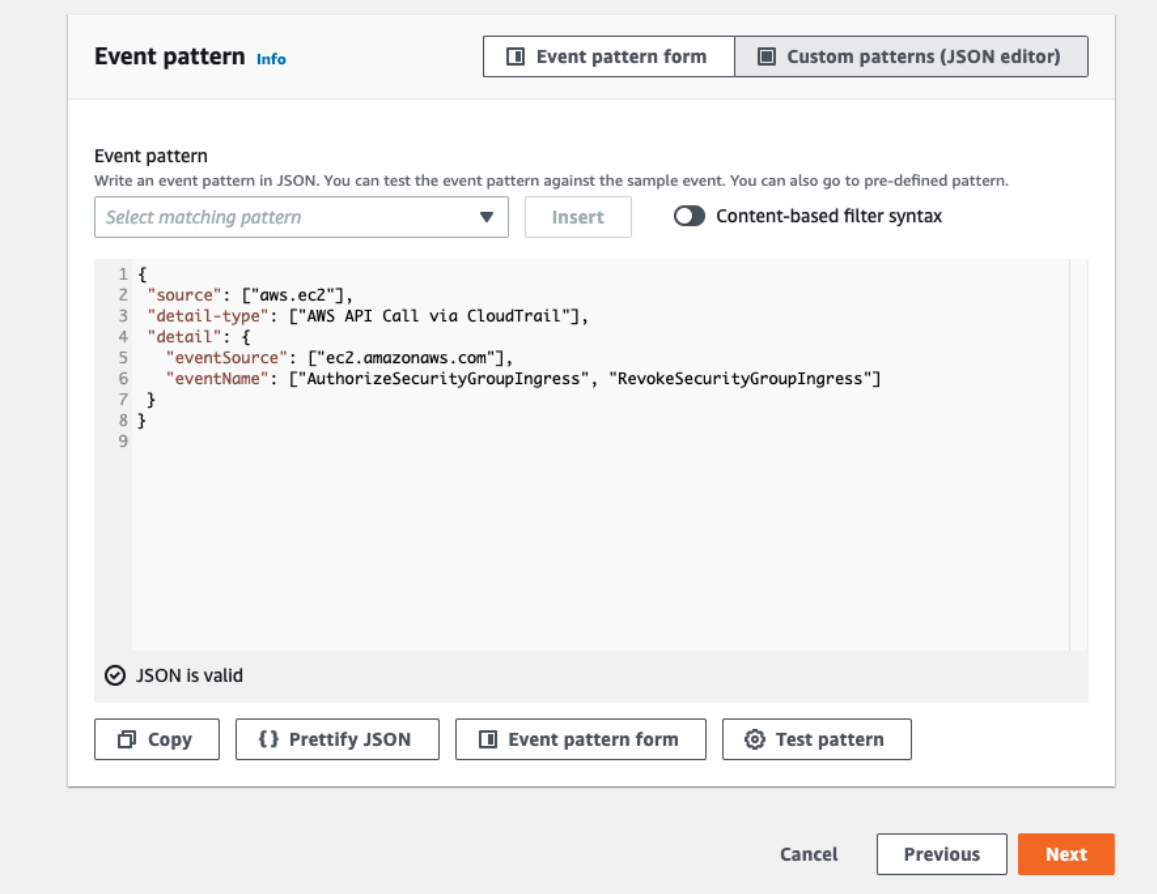
1.4.3. Choose Next

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"]
 }
}
```

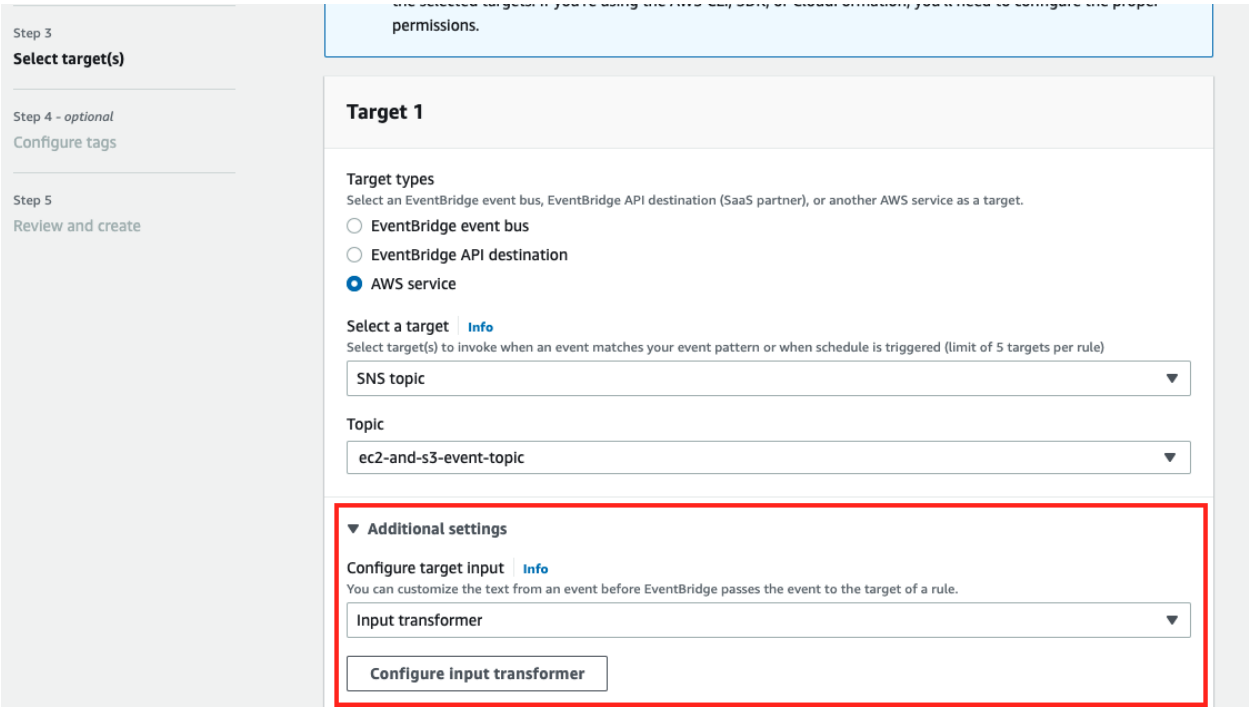
1.4.6. Click Next




1.4.7. Select a target, choose **SNS topic** as the target

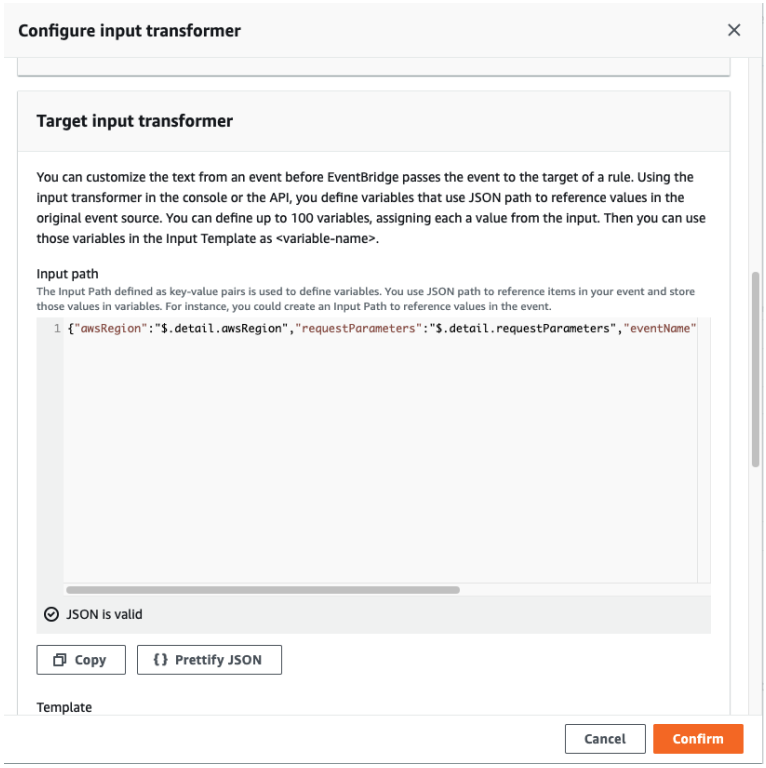
1.4.8. On the Topic, choose “ec2-and-s3-event-topic”

1.4.9. On the additional settings, choose input transformation and click configure input transformation



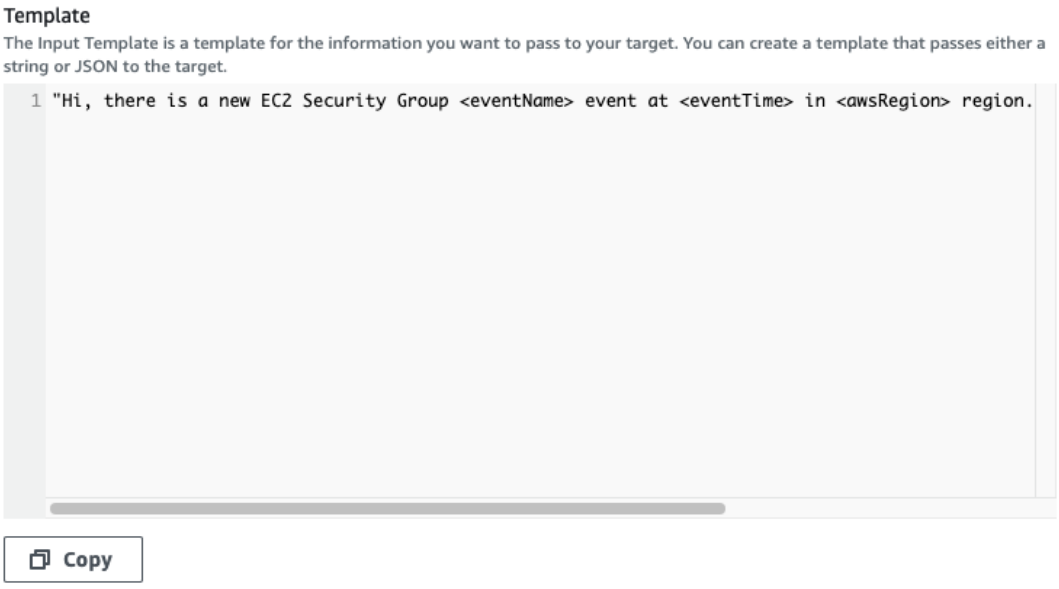

1.4.10. On the Target input transformer (**not sample event section)**, fill it with this in the **Target input transformer**

```
{"awsRegion":"$.detail.awsRegion","requestParameters"
:"$.detail.requestParameters","eventName":"$.detail.e
ventName","eventTime":"$.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>"
```




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

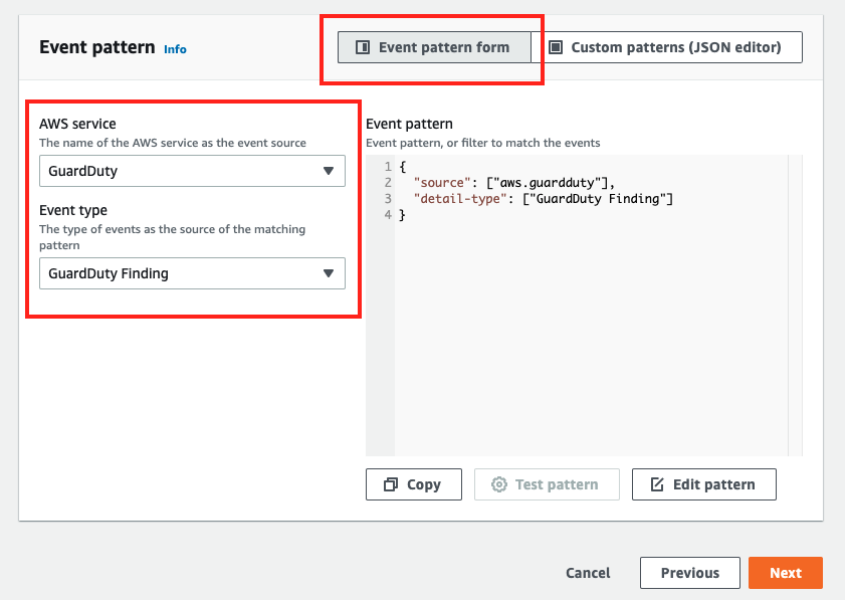
3.2.4. Scroll down and look for **Event Pattern** section

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'

        }
```

```
    # 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 !!!

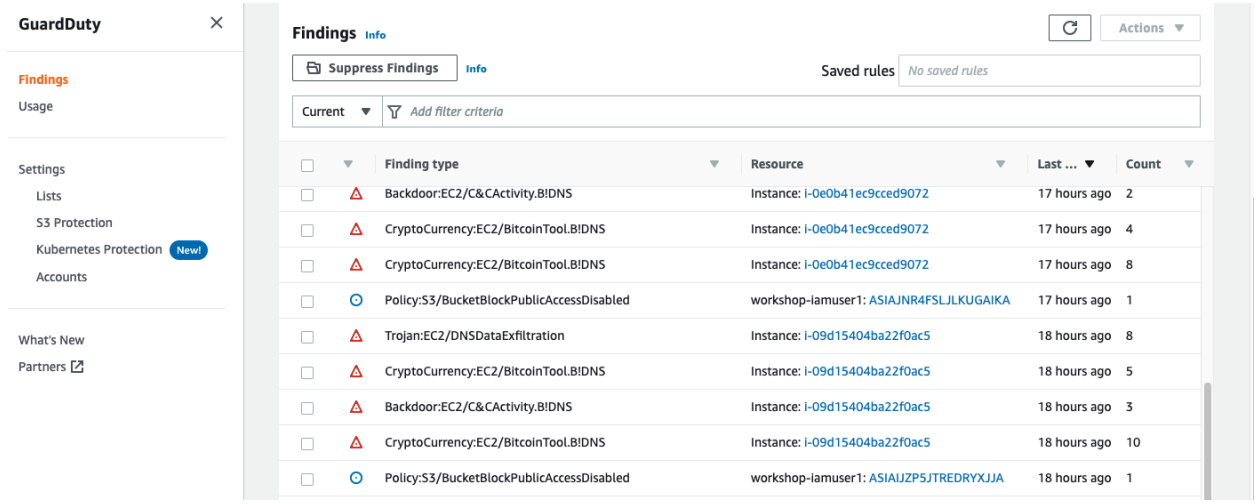
Let's see the magic what Threat Response Kit can do

# 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



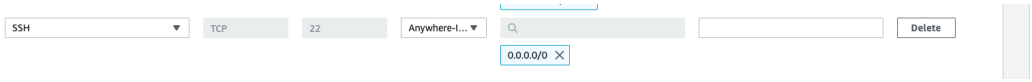

## 5.2 Validate Notification & Remediation Result

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.

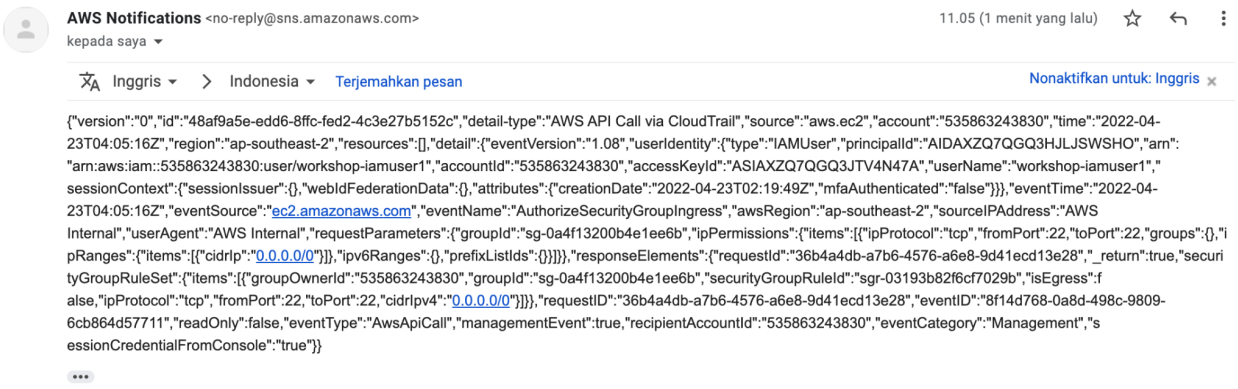
### EC2 Security Group Remediation Result & Notification

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

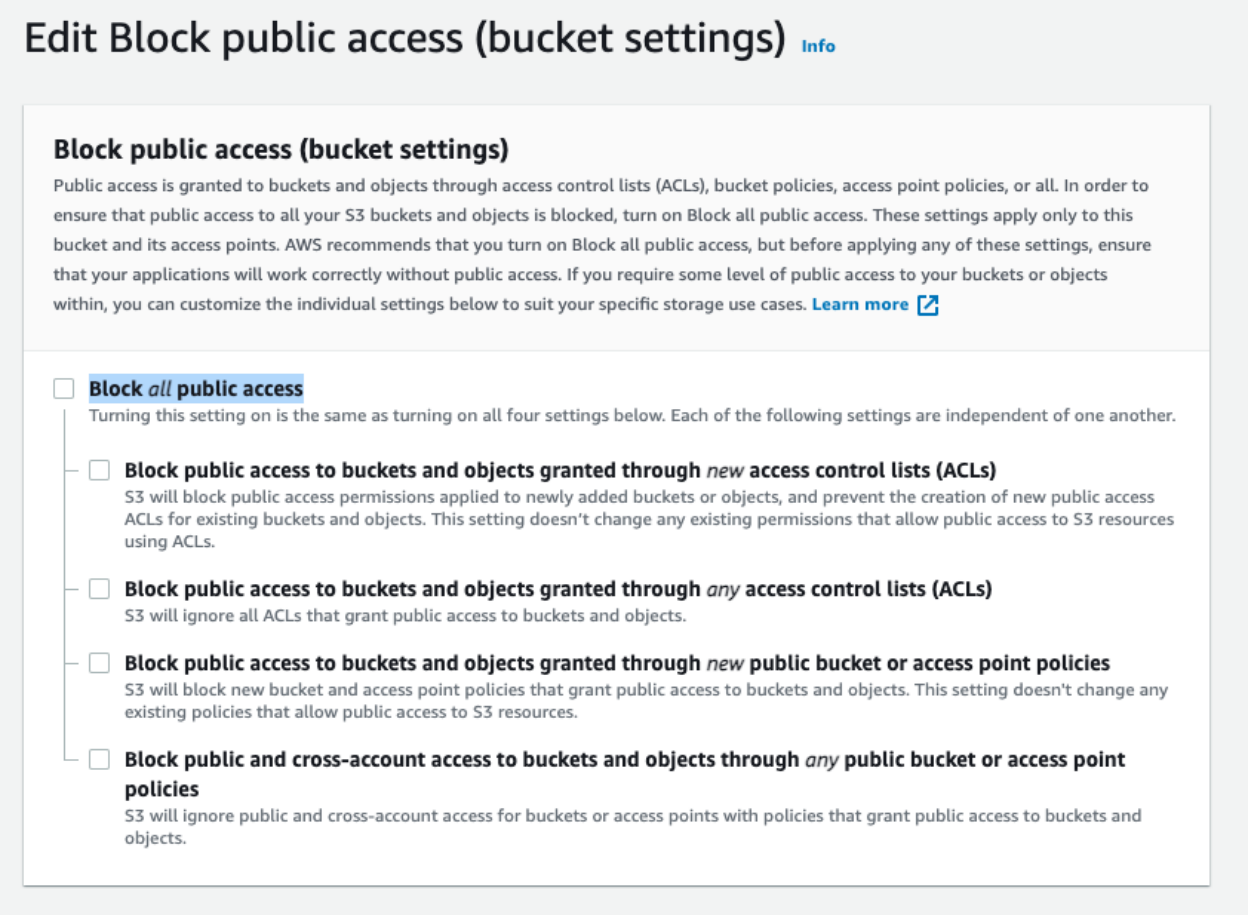



6. And when checking previously created rule, you will see that the rule is removed by Lambda function

## S3 Public Block Remediation Result & Notification

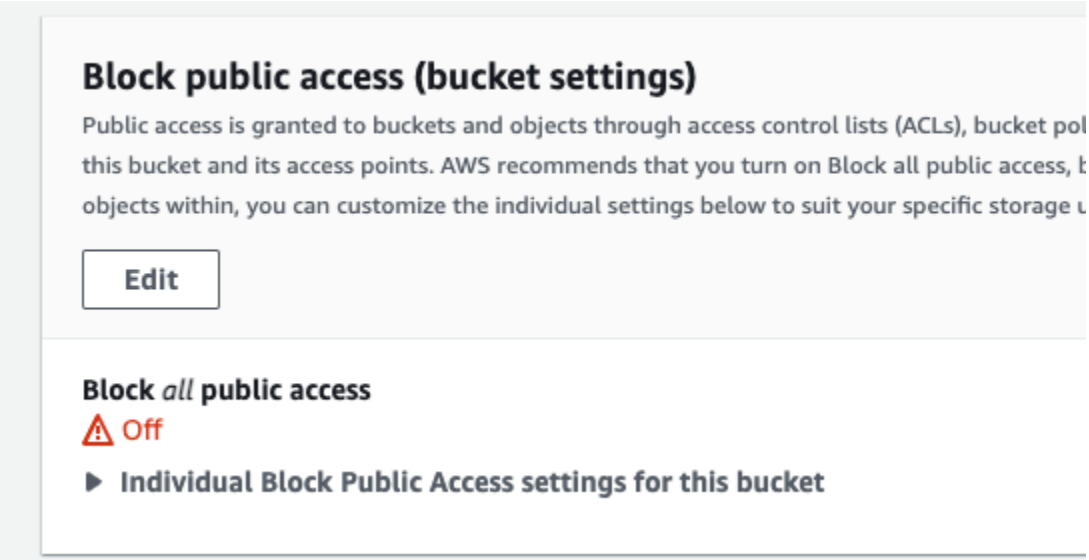
To test the remediation process you can follow the steps below:

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**

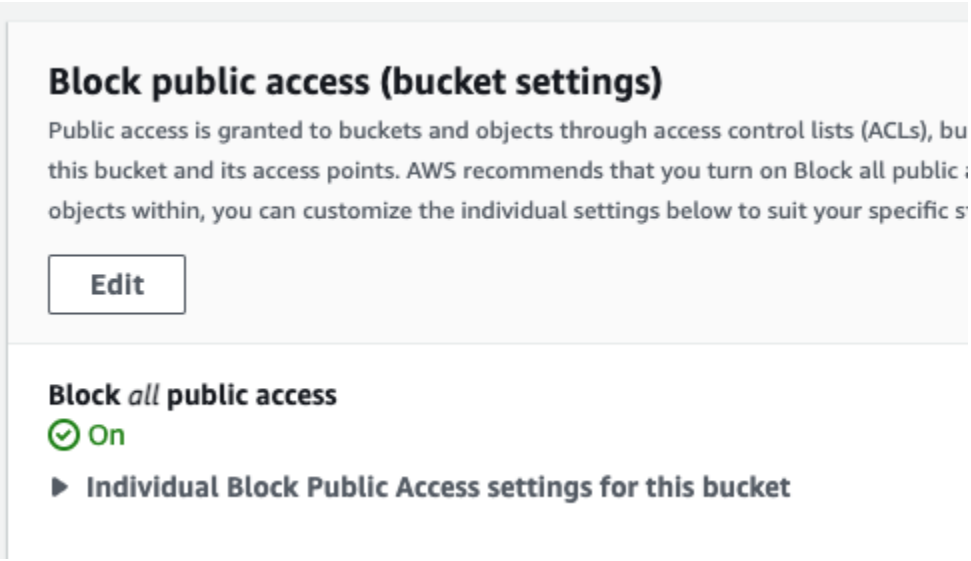



6. Click Save & write **confirm** on the dialog.

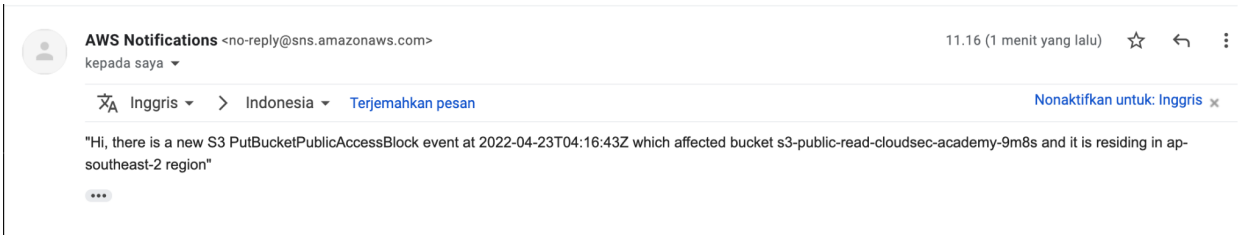
7. The result should look like this




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




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




## 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/