2.7b. Security Directives_EN

Document Status: DRAFT

Document Owner

INFRASTRUCTURE

@José Miguel Calavia Jurado

1. General

The safety guidelines described in this section are oriented to each service. These guidelines should be applied during resource creation, regardless of the technology used for creation.

The guidelines follow a basis established by Sanoma.

1.1 Encryption

Encryption of data at rest and in transit on AWS is critical to ensuring the security of information stored and transmitted in the cloud.

Two approaches are considered when discussing encryption::

- **Encryption of data at rest:** Protecting data stored on disks, databases and other storage media from unauthorised access. Throughout this document, each AWS service described in this document considers the configurations necessary to ensure data encryption using, in most cases, the service's default encryption keys. For the moment we do not use our own keys or those provided by Sanoma.
- **Data encryption in transit:** Encryption in transit from end users to the AWS resource/service that exposes the application is guaranteed by using HTTPS protocol, all applications use HTTPS protocol with a TLS 1.2 and 1.3 protocol version. This first part of communication is and must be encrypted in all applications without exception. The second part of communication, from the AWS service to the server/container is not encrypted, it is a plaintext communication, the third part of communication from the backend to the data layer is not encrypted so far. Details are being reviewed to make an encrypted End-to-End communication.

1.2 Environments

- Three environments must be maintained to ensure that applications have the correct tests and validations by the different actors involved in the applications.
- The environments to consider are:
  - **dev** - Dedicated and exclusive environment for development teams
  - **acc** - Also known as a pre-production environment, it must be an identical environment to the production environment, this environment is normally oriented to the Quality or Business team.
  - **pro** - The application's production environment, the environment where end users interact with the application.
- Each environment must have its own AWS account, in order to avoid mixing production data with those of low environments and reduce access to production.
- At the security level, to reduce exposure, DEV and ACC environments must be protected by a whitelist, that is, only SANOMA IPs and those of the providers that need to interact with the application must be allowed. See document: https://sanomaeducacion.atlassian.net/wiki/spaces/ISL/pages/890929154

1.3 Access

- Access to AWS accounts must be done by assuming roles from STL accounts: **there should be no IAM users for administrative, support, resource creation tasks** in the accounts.
- The permissions for developers in the dev account can be **PowerUser** (for now).
- The PRE environment should always have permissions considering the minimum privilege for people who use the environment. In this case, as previously mentioned, this environment is normally used for validations, so Developers who want to access this environment should have limited access (Read Only).
- The prod accesses should be only granted for administration of the Cloud, hence Read Only privileges for people who need to do Log reviews, etc.
- The connection to the different databases must be done only via Session Manager, through the use of a Bastion Host that must be provided by the Cloud team. (See **database access guide**)
- Write Permission to databases in the development environment may be granted for developers
- Access to PRE databases should be limited to people of interest, who interact with this environment considering read or read / write only.
- Access to production databases should be read / write only for application users and whose credentials must be registered with the SecretManager service. For support issues, reading access should only be provided to the necessary schemes after approval from the product PO to business / development / etc users. For these, the databases must have the audit log activated (see RDS Service)

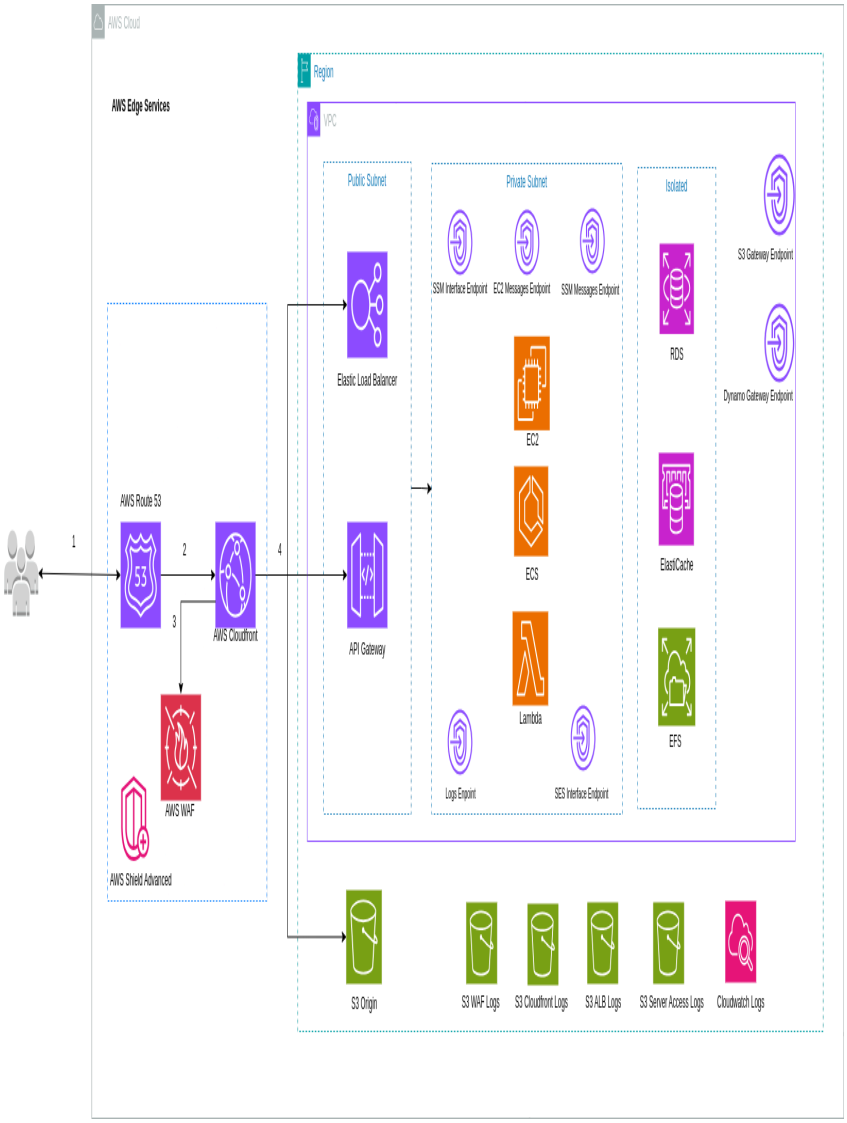
1.4 Suppliers

All Santillana / Sanoma providers that develop applications and need to interact with applications exposed to the internet in the Dev and Pre environments, must use the Sanoma VPN, that is, the project PO must manage accounts with Sanoma so that the providers can connect to the VPN and access protected environments without the need for the provider to provide their IP Pool.

Note: There may be exceptional cases where you have to add temporary test IPs or any developer who needs to access the applications and does not have a VPN account, in which case, the user must provide their public IP and this can be added to the specific IPSet (See WAF Section).

1.5 Reference architecture

It should include the use of AWS services that help improve the resilience of the application to attacks by DDoS, global services. In that sense, **every application** must have the Cloudfront service as the main front, this service will be associated with Route 53 and WAF to publish the applications. Cloudfront in conjunction with Route53 and WAF offers advantages for mitigation, isolation and rapid response to these types of attacks in the different layers of the OSI model.



2. AWS services

2.1 ACCOUNT

- Sanoma, must provide an account for the creation of a new project, this account will have to have the access role for administration by default, this role must be assumed from the jump account (STL).
- It must be verified that the account is completely clean without the use of services such as EC2, S3, ECS, RDS, etc. The default services enabled by Sanoma are SecurityHub, AWS Config, AWS Cloudtrail, AWS Guardduty.
- It must be verified that the account does not have existing IAM users, except in exceptional cases indicated by Sanoma / Santillana.
- The creation of IAM users within accounts **is forbidden,** except exceptions made for legacy applications that still need IAM keys, this must be justified and approved by Santillana / Sanoma, and subject to a mandatory key rotation every 90 days (**See IAM service**).
- One of the first actions in the account is the creation of buckets for the storage of logs, centralized buckets must be created in each account to store logs, the first

bucket to be created and to which the others are related is “osc-s3 -log-bucket ”, the bucket must be created using a Cloudformation stack called “**osc-s3-log-bucket**” (the name of the stack is a mandatory requirement). The other Buckets for Load Balancers, Cloudfront, WAF, will be named accordingly. Below are the templates for the buckets:

  - **Stack: osc-s3-log-bucket -** 2.7.1 Templates Cloudformation
  - **Stack: osc-alb-log-bucket -** 2.7.1 Templates Cloudformation
  - **Stack: osc-waf-log-bucket -** 2.7.1 Templates Cloudformation
  - **Stack: osc-cloudfront-log-bucket -** 2.7.1 Templates Cloudformation
- The use of AWS services should be limited only for the use of the VPC, a protection perimeter should be established at the IAM level with the following template:
  - **Stack: iam-vpc perimeter** - 2.7.1 Templates Cloudformation
  - The previously created policy should be used when creating an IAM user (exceptional case). In general, IAM users for the applications that use S3, SES, etc. shouldn’t be created, but if this is the case for any Legacy application, this policy should be associated with the IAM entity.
- Activate the subscription of **AWS SHIELD Advanced**. At the organization level, Sanoma covers the cost for the use of Shield, so the expense will not be reflected in the account.

2.2. VPC

The creation of the VPC has a dependency with Sanoma, the specification of **CIDR** must be provided by the Techops team or the Sanoma Networking team, this is because there are private network segments that must not match the reserved Sanoma segments.

- The VPC must have three types of subnets:
  - **Public** - Internet exposure via Internet Gateway
  - **Private** - Internet access through Nat Gateway
  - **Isolate** - No internet output (the data layer in the applications must reside in these subnets)
- The minimum number of subnets is 2 for each type, more can be created depending on the need for the application or project.
- The default ACL used by all Subnets must allow all traffic between the VPC as the first Entry rule (Lower Identifier 100). Additionally, subsequent rules should be created to block Subred-level traffic to ports 20, 22, 3306, 5432, 3389 (TCP), 3389 (UDP), with rule 9999 being weight or identifier rule to allow all traffic. Finally, a rule is observed that AWS defaults, a rule with Deny to everything else (*). Example:

- The Outbound, at this point, must allow all traffic.
- VPC creation must include creation **VPC Endpoints** Interface or gateway type, according to the specific case. **Mandatory** for:

- **S3** - **Gateway** (All types of subnets must be included for them to use the VPC Endpoint, it is not enough to associate only private subnets, even though we take into account that we should not create instances in public subnets, we lose nothing by associating all subnets to this endpoint)
- **SSM - Interface** (It is complemented by two other endpoints for server administration, it is necessary to create enpoints also for **SSM Messages**, **EC2 Messages.** Interface type endpoints can be associated with Private type subnets only.
- **SES - Interface** (Important if IAM Credentials are still maintained for sending emails. **Take into account the Policy iam-vpc-perimeter**). Interface type endpoints can be associated with Private type subnets only.
- **Dynamo - Gateway** (All types of subnets must be included, it is not enough to associate only private subnets, even though we take into account that we should not create instances in public subnets, we lose nothing by associating all subnets with this endpoint)

- **Is prohibited** the creation of security groups with 0.0.0.0/0 as Ingress to any port.

2.3. ACM

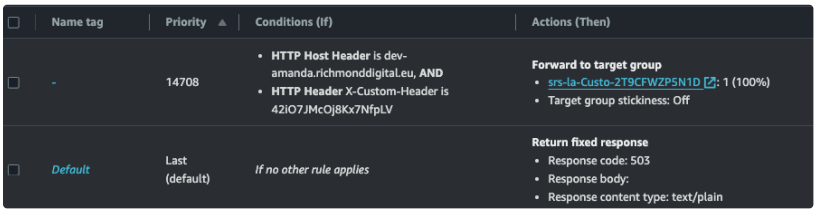
- Digital certificates must always be created with Wildcards for subdomains, this reduces visibility in the recognition stages for malicious actors. For example:
  - **Domain:** application1.example.com
  - **Certificate to request:** * .example.com
- The encryption algorithm for the certificate to be used in Cloudfront Balancers and Distributions must be: **ECDSA P 256.** This algorithm offers less computational effort and greater security compared to the default algorithm (RSA).
- The creation of certificates must always have validation by domains, the records must be sent to the person in charge of the administration of Sanoma DNS, indicating the CNAMES records and the account from which you want to use the certificate.

2.4. ALB

Load balancers must have the following attributes:

- **Drop Invalid Headers Fields: On**. This feature blocks poorly formed, poorly delimited, poorly codified Headers according to the HTTP / 1 and HTTP / 2 standards. It is useful for applications that have not considered the aspects to respond to the type of malicious headers, which represents a vulnerability that can be exploited.
- **Desync Mitigation Mode: Strictest.** This feature analyzes both the headers and the body of the HTTP request, a more complete review and prevents suspicious requests from causing inconsistency between the backend and the balancer, taking advantage of the fact that both components of the infrastructure (load balancer and backend) interpret the request differently.
- The logging of the balancer must be active, the deposit of logs must be the bucket that was initially created and must always have the name of the balancer as a prefix. Example:
  - **Access logs** : For this type of logs the following standard will be used: **s3: // < osc-balancer-logs > / < balancer-name >**

    - **Connection logs:** For this type of logs the following standard will be used: **s3: // < osc-balancer-logs > / connection-logs / < balancer-name >**
- The balancer must have active only the Listener **HTTPS 443**, and the default rule must respond with a code **503.** The rule must be used by default to direct traffic to a target group.
- The Listener 443 must have a previously created and validated ACM security certificate associated with it. The listener's security policy must be:
    - **ELBSecurityPolicy-TLS13-1-2-2021-06**, unless the balancer is the source of cloudfront, in that case:
    - **ELBSecurityPolicy-TLS13-1-3-2021-06**, or:
        - Use policy **ELBSecurityPolicy-TLS13-1-2-2021-06** if CloudFront returns an error 502 – Bad Request.
- The balancer **must not enable port 80** because exposure to end users is through Cloudfront, that is, the domain of the application being built must be associated with a Cloudfront distribution. Such distribution must originate from the public balancer through HTTPS communication only.
- The rules created to redirect Cloudfront traffic to the balancer and target group should specify the **Host Header** or **Path Based,** depending on the application, and also place the condition of a **HTTP Custom Header**, which must be a text string previously configured in the Cloudront distribution. Example:

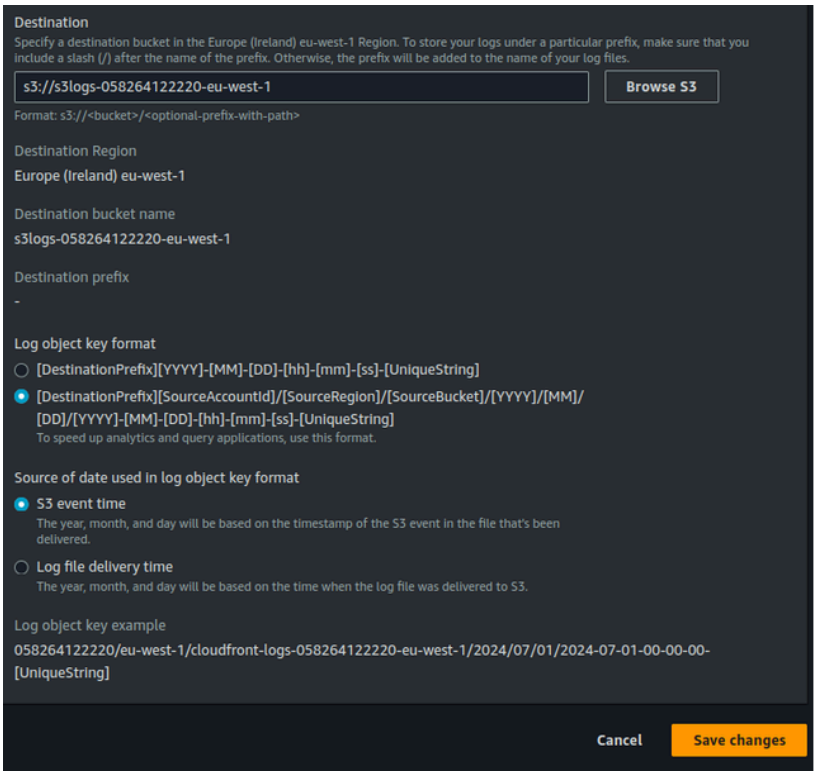


Name tag	Priority	Conditions (If)	Actions (Then)
-	14708	• **HTTP Host Header** is dev-amanda.richmonddigital.eu, **AND** • **HTTP Header** X-Custom-Header is 42iO7JMcOj8Kx7NfpLV	**Forward to target group** • srs-la-Custo-2T9CFWZP5N1D: 1 (100%) • Target group stickiness: Off
Default	Last (default)	*If no other rule applies*	**Return fixed response** • Response code: 503 • Response body: • Response content type: text/plain

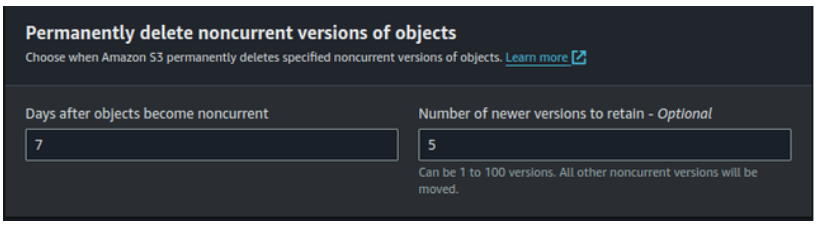
- The load balancer must be assigned a security group, this security group must only have a rule of entry to port 443 and as the origin only the **Prefix List** from Cloudfront.

2.5 s3

- Buckets must have versioning enabled.
- All buckets must be private. **There should be no public buckets**
- ACLs must be disabled
- The “Block all Public Access ” settings must be in mode **On.**
- Encryption for data at rest should be the **SSE S3.**
- The record of **Server Access Logs** must be enabled to send logs to one of the initially created buckets (**oc-s3-log-bucket**). **A Prefix should not be established** for deposit of logs and deposit must be based on **EVENT_TIME (data-based partitioning).** Example:



- Life cycle policies must be created for objects, three rules are **mandatory**:
  - `daily-housekeeping`: Delete objects listed as “Incomplete multiparts Upload ” and Delete Objects marked with Delete “Delete expired Object Delete Markers ” after **3** days of creation.
  - `expire-noncurrent-objects` **:** Deletion of NON-Current versions after **7** days of creation for Prod and **3** days for low environments, likewise, only 5 versions of objects should be kept.



- `transition-current-objects-to-intelligent-tiering`: Transition of objects larger than 128 kiB to Intelligent Tiering and number of days 0. “Transition current versions of objects between storage classes”
- **Mandatory**. The bucket **must contain the following policy**, in addition to the rules that the application needs:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "DenyAllWithoutHttps",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::BUCKET-NAME",
                "arn:aws:s3:::BUCKET-NAME/*"
            ],
            "Condition": {
                "Bool": {
                    "aws:SecureTransport": "false"
                }
            }
        },
        {
            "Sid": "DenyInsufficientEncryptionInTransit",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::BUCKET-NAME",
                "arn:aws:s3:::BUCKET-NAME/*"
            ],
            "Condition": {
                "NumericLessThanIfExists": {
                    "s3:TlsVersion": "1.2"
                }
            }
        },
        {
            "Sid": "DenyAllWithoutSigV4",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::BUCKET-NAME",
                "arn:aws:s3:::BUCKET-NAME/*"
            ],
            "Condition": {
                "StringNotEqualsIfExists": {
                    "s3:signatureversion": "AWS4-HMAC-SHA256"
                }
            }
        }
    ]
}
```

Note: When creating the CDK bucket, you should be careful with the specification of **Principal,** Default CDK states “AWS ”: “ * ”, what you want is that not only AWS services like Users, Services, Accounts have this, but also other identities like **Canonical Users**, **Federated users**. For this reason “ * ” is placed.

2.6. Cloudfront

- The Cloudfront distribution must have the Price Class of **“only North America and Europe ”** (Level 100 in Cloudformation or CDK).

- The security policy associated with the distribution of Cloudfront should be: **TLSv1.2_2021**
- Distribution must have default file Default Root Object, normally **“index.html ”**
- The log deposit must be active, the destination bucket is the bucket indicated in the “Account” part of this document. For a better analysis of the logs, a prefix must be configured: this must be the “Alternate Domain Name ” of the distribution (if there are several “Alternate Domain Name ”, select the first from the list), for example: **s3: // < bucket-logs-cloudfront > /example.santillana.es**
- The “Cookie Logging ” option Must be active, in “On ” mode
- IPv6 option must be in “On ” mode
- The HTTP / 3 and HTTP / 2 versions must be active.
- The CDN distribution must have a description, it is recommended to put a brief description or simply the URL of the service.
- The Cloudfront distribution must have a WAF ACL associated: WAF settings are detailed in the following section. When you create the Cloudfront distribution it should be noted that ACL should be used, which means that you must have previously created the ACL.
- **For S3 Origins:**
  - Access to origin must be limited by Origin Access Control (OAC), the policy must be added to the bucket, in addition to the previously established security policies.
  - As far as possible, the configuration of the Viewer Protocol Policy should be in “HTTPS Only ”, if it is an application accessible from Browsers for end users, you can set “Redirect HTTP to HTTPS ”.
  - If the application is only to distribute content, the allowed HTTP methods should only be “Get, Head ”.
  - Some content does not need restriction for its distribution - for example, the distribution of content of type CSS, JS, IMAGES, FONTS, in many cases it is free and does not pose a risk. However, for private content, the restriction must be activated and the display method must be through signed URLS or signed Cookies, depending on the use case. In both scenarios, the creation of the key must be done by environment and must be stored in a service such as Secret Manager with Limited permissions.
  - Cache policy should be that recommended by AWS “CachingOptimized ”,
  - A Response Headers policy must be created, this policy will be included in the “Behavior” settings of the “distribution **Response Headers Policy ”**. The basic configuration should be as follows:

```
const originResponseHeadersPolicy = new cloudfront.ResponseHeadersPolicy(this, 'ResponseHeadersPolicy', {
        responseHeadersPolicyName: xxxxxxxxxxxxxx
        securityHeadersBehvior: {
          strictTransportSecurity: { accessControlMaxAge: cdk.Duration.seconds(31536000), includeSubdomains: true, override: true, preload: true },
          contentTypeOptions: { override: true }
          xssProtection: { protection: true, modeBlock: true, override: true },
          frameOptions: { frameOption: cloudfront.HeadersFrameOption.DENY, override: true },
          referrerPolicy: { referrerPolicy: cloudfront.HeadersReferrerPolicy.STRICT_ORIGIN_WHEN_CROSS_ORIGIN,
```

```
override: true },
        },
        removeHeaders: ['Server', 'X-Powered-By'],
      });
```

- **For ALB Origins**
  - The protocol to go to the origin must be “HTTPS Only ” on port 443 and the Security Policy must be **ELBSecurityPolicy-TLS13-1-3-2021-06**. If the balancer is going to be associated with the domain, for cases where there is no Cloudfront ahead (not recommended), for compatibility reasons the Security Policy should be: **ELBSecurityPolicy-TLS13-1-2-2021-06**
  - A Custom Header must be added with the Header Name: **x-custom-header** and the **value: XXXXXXXXXXXXXX** (It can be a string of 20 or 30 characters, doesn’t include non-ASCII characters and the header name must be lowercase). This Custom Header will be added to the load balancer rule that the distribution has as Origin and prevent other distributions outside our Scope from accessing the balancer.
  - As much as possible, the configuration of the Viewer Protocol Policy should be in “HTTPS Only ”, if it is an application accessible from Browsers for end users, you can set “Redirect HTTP to HTTPS ”, but in the case of APIs, redirection is not necessary.
  - For this type of origin, the allowed HTTP Methods can include PUT, POST, depending on the need for the application.
  - Cache policies should be those recommended by AWS (UseOriginCacheControlHeaders and AllViewer)
  - A Response Headers policy must be created, this policy will be included in the “Behavior ” settings of the “distribution **Response Headers Policy ”**. It can be the same policy indicated for origin s3.
- Cloudfront distribution must be associated with **Shield Advanced**, for this, the subscription must have been previously made, as indicated in the first part of this document “Accounts”. An easy way to associate AWS Shield Advanced with Cloudfront distribution via CDK:

```
new shield.CfnProtection(this, 'WafShield', {
        name: `${project}-shield`,
        resourceArn:
`arn:aws:cloudfront::${this.account}:distribution/${cloufront.distri
butionId}`,
        tags: [
          {
            key: 'Environment',
            value: 'Production',
          },
        ],
        applicationLayerAutomaticResponseConfiguration: {
          status: 'ENABLED',
          action: {
            block: {} },
        },
      });
```

2.7. WAF

The WAF rules follow SANOMA's recommendations, they are a mixture of AWS rules and company rules.

The template follows the recommendations indicated in the **Environments** section, the default action is **“Block ”**, this is useful for Dev and Pre environments, for Prod you should adjust the parameter to **“Allow ”**. Template creates three **IP Sets,** two of them belonging to Sanoma IPs, to tools that perform periodic security scans on applications. The third **IP Set** called **“Testers ”** It is made to place some IP of the providers, who, in case of emergency, need to access the Dev and Pre environments and do not have a Sanoma VPN account. (See General - Suppliers).

Sanoma / Santillana applications are not global, that is, they do not have a globally distributed target audience, that is why only certain countries should be able to access the applications, in that sense, there is a Geographic blocking rule, any request that does not come from a country in the following list, it will be blocked:

```
Belgium - BE
Finland - FI
Netherlands - NL
Norway - NO
Poland - PL
Spain - ES
Sweden - SE
```

The Geo Blocking rule should be the last rule, above the AWS Shield rule that is automatically added after associating Shield Advanced.

The List of allowed Countries depends of the Product Owner of the application, some applications probably needs to be open for more countries.

Note: There is a little guide on how to deploy the WAF stack with CDK, the guide is found in the following link: 2.7.1 Templates Cloudformation | Desplegar el Stack con CDK

2.8. RDS

- Due to performance review issues, Performance Insights must be activated at the cluster level.
- Audit logs must be active: Audit Log, Error Log, General Log, Slow query log. The first generates a large amount of data, the other three do not generate as much data.
  - In order for logs to be displayed in Cloudwatch, you have to create a custom Parameter Group for the cluster, it is not enough to modify the database. The cluster parameter must have the following values:

```
'general_log': '1',
'slow_query_log': '1',
'server_audit_logging': '1',
'server_audit_events': 'CONNECT,QUERY,QUERY_DDL,QUERY_DCL',
```

- The previous logs will start generating data in Cloudwatch, if we do not specify a retention period we will cause unnecessary expenses. Retention must be configured for 6 months, this can be easily accomplished on CDK with the line:

```
cloudwatchLogsExports: ['audit', 'error', 'general', 'slowquery'],
cloudwatchLogsRetention: cdk.aws_logs.RetentionDays.SIX_MONTHS,
```

- The “Enable Auto minor version upgrade ” box must be active. Minor updates include security patches and updates for performance enhancement. They are generally not disruptive, but it is recommended to program the RDS maintenance window on a less frequent user schedule.
- The “Enable deletion protection ” box must be active.
- For Mysql and Aurora SQL, Graviton processor databases work perfect and are cheaper than Intel processors, we should use this instances in the most of cases.
- The backup for productive environments should be 30 days and for lower environments should be 7 days.

2.8. ECS/ECR

- Ensure Container Insights with enhanced observability is enabled during the creation of the cluster.
- Use Fargate (serverless) in the most of cases, we should to avoid to use EC2 servers.
- The encryption of the ECR must be AES-256.
- In General Settings, Enhanced Scanning , Continues Scan all Respositories and Scan on Push boxes should be active.
- Lifecycle policy should permit only the last 5 images inside of the repository.