OpenSecOps Foundation

Technical Design Specification

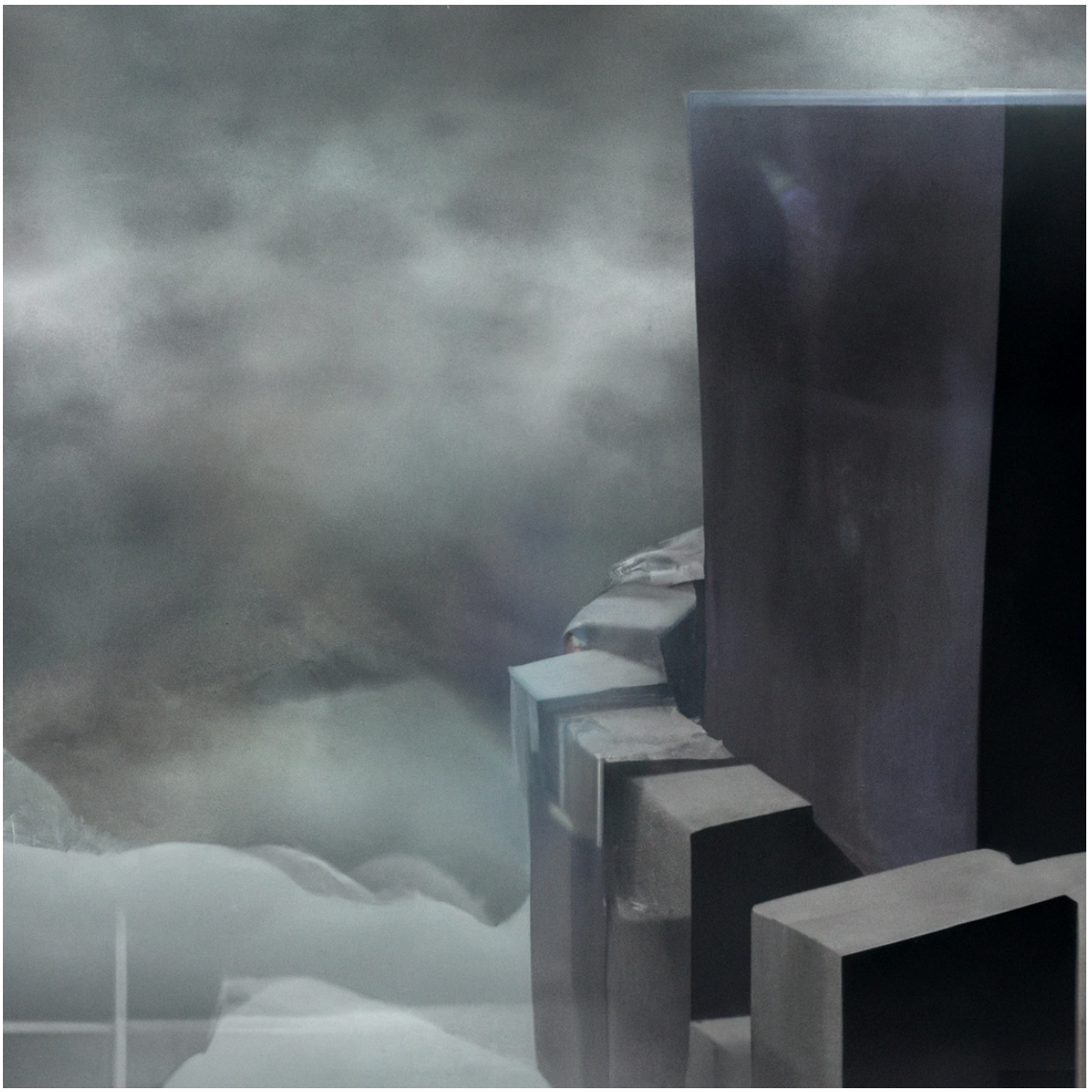


Table of Contents

Document Versions

Version	Date	Changes	Author
1.0	2024-02-14	First version	Peter Bengtson
1.1	2024-09-25	Update for FTR	Peter Bengtson
1.2	2025-03-27	References to the Delegat Foundation reference system and AWS Copilot removed	Peter Bengtson
1.3	2025-04-07	"Delegat" replaced by "OpenSecOps" throughout	Peter Bengtson

1 Introduction

This TDS describes the OpenSecOps Foundation system design goals, infrastructural and developmental architectural principles, and their implementation.

Scope of TDS

This document is intended to give a comprehensive high-level outline of how the OpenSecOps Foundation system is designed and set up. For deeper information outside of this scope, see the TDSs and SOPs relevant to the system level, such as

- OpenSecOps Foundation system SOP
- OpenSecOps SOAR TDS
- OpenSecOps SOAR SOP

For operational information pertaining to the account level:

- OpenSecOps AWS Account Properties SOP
- OpenSecOps KMS Keys SOP
- OpenSecOps S3 Buckets SOP
- OpenSecOps DynamoDB SOP

2 System Design Goals

Self-Managed Application Teams
- System designed to support agile team work modes
- Architecture as Code
- Application Developers are empowered by unusually wide permissions

Security
- Security by design
- Escalation of Privileges is prevented at every level of the system, thereby eliminating a plethora of attack vectors
- System-Level Builders and Administrators are also under restrictions
- Extensive SOAR automation to enforce security and development standards

Cost Efficiency
- Considerably reduced reliance on operations teams
- No container operations team needed
- Serverless technology everywhere reduces operational costs

Ease of Governance
- Best cloud practices built-in
- Single Sign-On
- Declarative account vending machinery
- Declarative assignment of permissions to accounts and groups
- Declarative subdomain OpenSecOpsion
- Incident handling for infra and applications integrated with Jira Cloud
- Severity-based incident SLA escalation in Jira Cloud
- Automatic log collection and archival from all accounts
- Log aggregation to reduce long-term archival costs to a minimum
- Quick setup – a new OpenSecOps Foundation system can be deployed in under a week

In the OpenSecOps Foundation system, the overarching principle is to empower Builders both on the application level and the infrastructural level to manage their own infrastructure; furthermore, to do so in a way which eliminates the risk of escalation of privileges.

Builder Teams themselves set up and manage individual VPCs and serverless container clusters and their pipelines. Moreover, they do this through Architecture as Code, meaning they don’t have to know how to architect infrastructure on AWS.

This also means that the need for a specialised container operations team is removed entirely, dramatically reducing operational costs and the TCO (Total Cost of Ownership) for the compute layer.

Application teams work with Architecture as Code and with the platform-agnostic container as the deliverable. This means that team members do not need to know how to configure infrastructure on AWS. This reduces the reliance on developers with specialised AWS infrastructure expertise and instead allows the team to concentrate on developing feature functionality.

The OpenSecOps Foundation system is a highly scalable production system for state-of-the art best-practice installations on AWS and can be deployed in less than less than a week.

3 Teams, Functions & Definitions

The following are the types of teams created for the most central functions in the OpenSecOps Foundation system and the kind of access they have. Other roles will be added when mandated by changes in team functions, tasks, or structure.

The norm for everyone in the system is to work with tailored, least-privilege access and, if a Builder, with protection against the escalation of privileges.

The structure implemented by these teams and their various functions, permissions and requirements is strictly enforced by Service Control Policies (SCPs). Provided the framework they establish is kept to, escalation of privileges cannot occur.

Infra

Simply short for Foundation Team.

Foundation Team

The Foundation Team consists of the people working closest to the cloud provider infrastructure, managing it for the rest of the organisation. The Foundation Team consists of Cloud, Security, and Network Administrators.

Foundation Team access modes are extremely powerful. They give maximum freedom to each Infra team. Thus, they must be closely matched to job function according to least-privilege principles when assigned to people.

The Foundation Team is Infra.

Cloud Administrators

Cloud Administrators are the only ones to have unrestricted permissions in all accounts. People in this group are trusted implicitly. Membership should be restricted to one or two people, and general paranoia should be the norm. None but these one or two "`sudo`" people should ever work with unrestricted access, and then only when absolutely necessary.

Indeed, following the principle of Separation of Concerns, Cloud Administrators should choose to use an access mode as specific as possible for the type of work undertaken, just like everyone else. Only when the foundations of the system need to change should a Cloud Administrator use unrestricted access.

Cloud Administrators are Infra.

Security Administrators

Security administrators manage security services and tools across the environment, ensuring all services and workloads run on a secured infrastructure. This function has access to the environment to remediate any possible security threat.

Security Administrators are Infra.

Network Administrators

Network Administrators have full access permissions to AWS services and actions required to set up and configure AWS network resources, including DNS.

Network Administrators are Infra.

Platform Team

The traditional task of a Platform Team is to provide the infrastructure, environments, deployment pipelines and other internal services that enable internal customers (usually application development teams) to build, deploy, and run their applications.

In the OpenSecOps Foundation system, the teams can do most traditional Platform Team activities themselves. The Platform Team provides the application teams with tooling, perhaps helps with the initial project setup, and assists and advises when required.

The OpenSecOps Platform Team is thus the intermediary between the Foundation Team and all types of Builders.

Platform team members can be a mixture of Infra and application-level depending on their tasks.

Builders

Anyone writing code for the system is a Builder. This includes application developers as well as system-level AWS engineers.

For the code produced by Builders, specifically the Roles created, escalation of privileges is always a risk. Therefore, all types of Builders always work with strict Permission Boundaries enforced by SCPs. This includes Infra Builders as well as application developers.

System-level Builders are Infra, application Builders are not.

Cost Managers

Cost Managers manage the spending of the environment and create budgets and alerts based on forecasting expenses. This function is also responsible for paying invoices.

Cost Managers are not Infra.

SOC

Security Operations Centre (SOC) is a centralised function employing people, processes, and technology to continuously monitor and improve an organisation's security posture while preventing, detecting, analysing, and responding to

cybersecurity incidents.

A SOC acts like the hub or central command post taking in telemetry from across an organisation's IT infrastructure, including its networks, devices, appliances, and information stores, wherever those assets reside. The proliferation of advanced threats places a premium on collecting context from diverse sources. Essentially, the SOC is the correlation point for every event logged within the organisation that is being monitored. For each of these events, the SOC must decide how they will be managed and acted upon.

SOC personnel are most definitely not Infra and should not have any kind of mutation access. Instead, SOC interacts with the local Security Team and with the account owners who are then responsible for mutating their infrastructure if required.

NOC

A Network Operations Centre (NOC, pronounced like the word “knock”) is one or more locations from which network monitoring and control, or network management, is exercised over a computer network.

NOC personnel are responsible for monitoring one or many networks for certain conditions that may require special attention to avoid degraded service.

NOC is Infra if part of the Network Administrator group.

4 Basic Configuration

AWS Control Tower

The OpenSecOps Foundation system uses AWS Control Tower to manage the foundational aspects of the system, as Control Tower offers a straightforward way to set up and govern an AWS multi-account environment following prescriptive best practices.

AWS Control Tower orchestrates the capabilities of several other AWS services, including, but not limited to, AWS Organizations, AWS Service Catalog, AWS CloudTrail, AWS GuardDuty, and AWS SSO. AWS Control Tower orchestration extends the capabilities of AWS Organizations.

The following two extensions to Control Tower, both supported by AWS, are used in OpenSecOps to manage accounts and policies in a declarative fashion.

AFT: Account Factory for Terraform

Account Factory for Terraform, or AFT, is the account vending machine for Control Tower. It is controlled by four Git repositories in the `AFT-Management` account, where AFT resides, and thus is declarative in nature.

The Account Administration group has access to the `AFT-Management` account as well as to the Service Catalog console in the `Org` account for account, group, and user management.

AFT has been extended with declarative facilities for SSO Group and SSO User assignment to accounts, described here: https://github.com/OpenSecOps-AB/AFT-SSO-account-configuration. Other extensions, such as for specifying capped budgets for personal user accounts, can be found in the configuration repos.

For more information about AFT, see the repos in `AFT-Management` and the AFT documentation at https://docs.aws.amazon.com/controltower/latest/userguide/taf-account-provisioning.html. Also, see the OpenSecOps Foundation system SOP.

Regions

Any region or regions can be used. The OpenSecOps Foundation system configures regional access in a fully secure way. You decide which regions are to be available. The rest cannot be accessed in any way, not even by administrators. Only Cloud Administrators can enable or disable regions.

Foundational AWS Services

These are the AWS services that constitute the foundations of the system.

AWS Single Sign-On

AWS Single Sign-On, or IAM Identity Centre as is its newer name, is the recommended approach for workforce authentication and authorisation on AWS for organisations of any size and type. OpenSecOps Foundation uses federated AWS SSO Users for all authentication and authorisation, disallowing the creation of IAM Users entirely for anyone not a Cloud Administrator.

AWS SSO administration has been OpenSecOpsed to the `AFT-Management` account.

AWS CloudTrail

There is exactly one organisation-wide, multi-regional CloudTrail in the system. It collects CloudTrail events from all accounts and stores them in the Log Archive account for archival purposes. The CloudTrail event logs are secure, encrypted, tamper-proof, and CIS-compliant.

There is no need for anyone to create further CloudTrails, and the capability to do so is restricted to Cloud Administrators only.

AWS CloudTrail has no global administration aspect to be OpenSecOpsed.

AWS Config

AWS Config is enabled everywhere and is used heavily by Security Hub, GuardDuty and other security services.

AWS Config administration has been OpenSecOpsed to the `Security-Adm` account.

AWS GuardDuty

Amazon GuardDuty is a threat detection service that continuously monitors for malicious activity and unauthorised behaviour using machine learning and other techniques. GuardDuty findings automatically appear in Security Hub.

AWS GuardDuty administration has been OpenSecOpsed to the `Security-Adm` account.

AWS Security Hub

AWS Security Hub is used to aggregate all security data in the OpenSecOps Foundation system. This includes info and findings from AWS native security services such as GuardDuty, CloudTrail, VPC Flow Logs, IAM Access Analyzer, AWS Inspector, AWS Macie, and others, and also from external security software through its many integrations.

AWS Security Hub findings in the ASFF format also form the basis on which the OpenSecOps SOAR security automation is built and through which the high security posture of the OpenSecOps Foundation system is upheld.

Security Hub findings cannot be manipulated by non-admins. This means that issues can’t be eliminated except by fixing the underlying problem. Thanks to this, users do not have to close Jira tickets manually - OpenSecOps SOAR will do it for them when it detects that the issue has been resolved properly.

It should be noted that the Security Hub controls for global services such as IAM are enabled in `eu-north-1` only, to avoid multiple tickets for the same issue. It does not matter in which region a global service actually resides.

AWS Security Hub administration has been OpenSecOpsed to the `Security-Adm` account.

AWS CloudFormation

AWS CloudFormation lets you model, provision, and manage AWS and third-party resources by treating infrastructure as code.

AWS CloudFormation administration has been OpenSecOpsed to the `IaC` account. At present, the OpenSecOpsion remains unused: all CloudFormation templates are currently deployed in the Org account.

Additional AWS Services

AWS Inspector

AWS Inspector has been enabled in all accounts. The results of the security scans are available in the AWS Inspector console as well as in the form of findings in Security Hub.

Account owners can create any number of local Inspector runs, for instance, to assess only certain instances or instances with specific tags.

AWS Detective

Amazon Detective makes it easy to investigate, analyse, and quickly identify the root cause of potential security issues or suspicious activities. Amazon Detective automatically collects log data from AWS resources and uses machine learning, statistical analysis, and graph theory to help teams and admins visualise and conduct faster and more efficient security investigations. It is available both separately and as a seamlessly integrated part of AWS Security Hub.

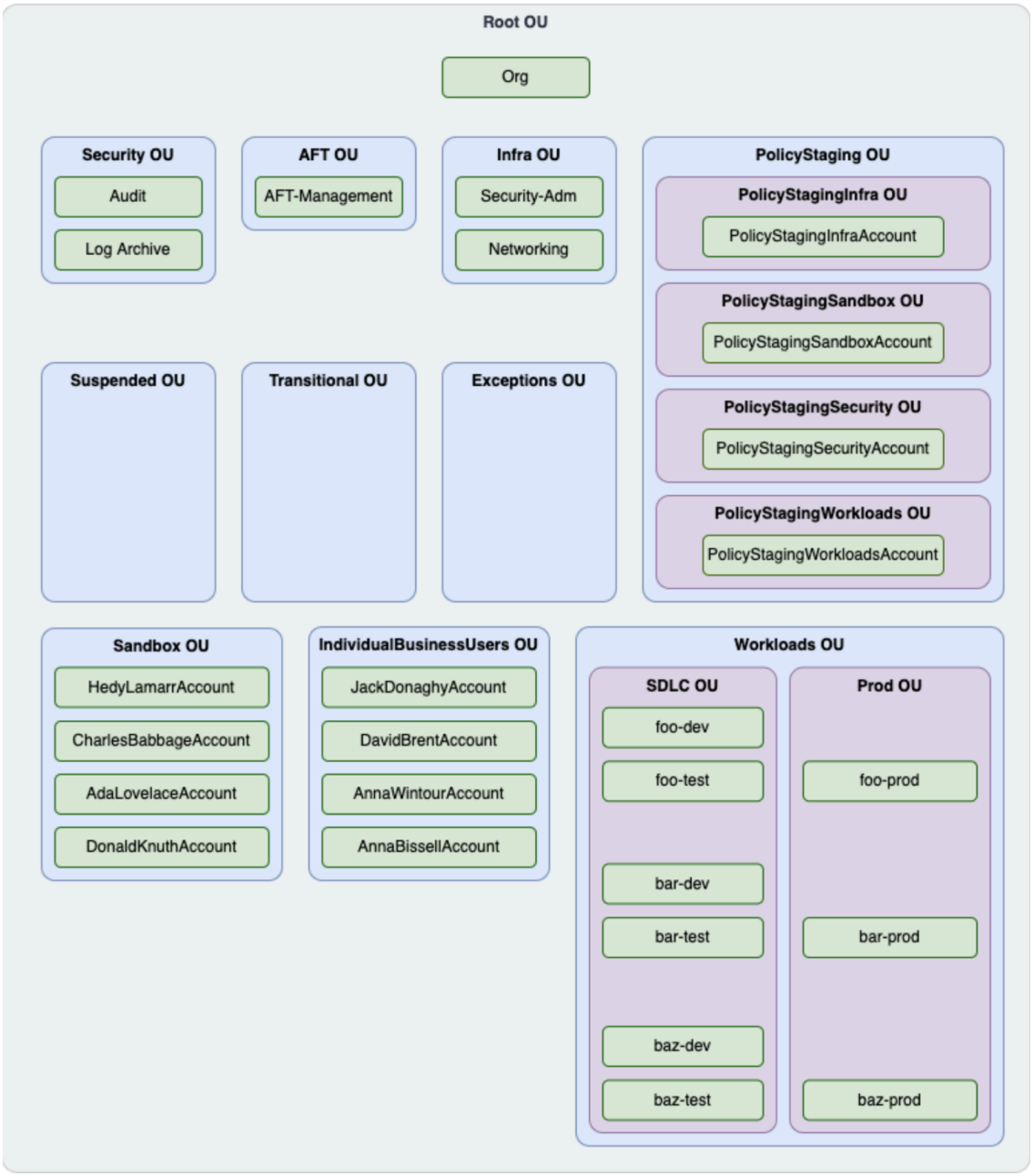
AWS Compute Optimizer

Compute Optimizer identifies workload patterns and recommends optimal AWS resources. Compute Optimizer analyses the configuration and resource utilisation of workloads to identify dozens of defining characteristics, for example, if a workload is CPU-intensive, if it exhibits a daily pattern, or if a workload accesses local storage frequently. The service processes these characteristics and identifies the hardware resource required by the workload. Compute Optimizer infers how the workload would have performed on various hardware platforms (e.g. Amazon EC2 instance types) or using different configurations (e.g. Amazon EBS volume IOPS settings, and AWS Lambda function memory sizes) to offer recommendations.

AWS Compute Optimizer is available to teams and admins in all accounts.

Organisational Units

The following diagram describes the hierarchical arrangement of OpenSecOps Foundation's Organisational Units (OUs) and the accounts they contain. OU names are shown in bold, and accounts in green:



Root OU

The Root OU encompasses all other OUs. It holds only a single account, `Org`, the AWS Organisations root administration account for the whole hierarchy of OUs and member accounts.

Security OU

The Security OU (formerly known as Core by AWS) contains the two security-related accounts deployed by AWS Control Tower: `Audit` and `Log Archive`.

AFT OU

The AFT OU contains one single account, `AFT-Management`, for account and SSO management.

Infra OU

The Infra OU contains two accounts: `Security-Adm` and `Networking`. These are builder accounts for the Infra team. All provisioning of system-related infrastructure should take place here. To avoid the escalation of privileges, the Infra team must use dedicated permission boundaries when creating IAM Roles, just like the application builders in the Workloads OU. Additional accounts will be added to the Infra OU as necessary.

Suspended OU

The Suspended OU holds suspended accounts awaiting destruction. Its SCPs are extremely restrictive.

Transitional OU

The Transitional OU holds accounts under migration to the AWS Organisation.

Exceptions OU

The Exceptions OU holds accounts that do not fit any other category. The number of accounts here should be kept as low as possible at all times.

PolicyStaging OU

The PolicyStaging OU, which in its turn contains the four sub-OUs PolicyStagingInfra, PolicyStagingSandbox, PolicyStagingSecurity, and PolicyStagingWorkloads, which in turn each contain a single account.

These OUs and accounts are intended for the development and testing of new policies and SCPs for the Infra OU, Sandbox OU, Security OU, and Workloads OU before they are applied to the live OUs.

They are builder accounts and have access modes with boundary permission restrictions for creating IAM Roles as well as more unrestricted access modes available.

Sandbox OU

The Sandbox OU contains personal sandbox accounts for builders. They require boundary permissions for all created IAM Roles.

IndividualBusinessUsers OU

The IndividualBusinessUsers OU is like the Sandbox OU but for business people using prepackaged software and SaaS services. Accounts in this OU are not builder accounts; infrastructure cannot be created here by the account owners, only by administrators. Should an admin create a Role in this account, a boundary permission policy must be specified.

Workloads OU

The Workloads OU is further subdivided into two OUs, the SDLC OU and the Prod OU. The SDLC (Software Development Life Cycle) OU contains accounts for

development and testing. The Prod OU contains production accounts. Account names in the Workloads OU follow a consistent convention based on the service name and the environment.

To prevent escalation of privileges, boundary permission policies are required for all IAM Roles in this OU.

The SDLC and Prod OUs are *not* further subdivided into service or team OUs. Remember: OUs are not meant to, and *should not*, mirror any company structures as they are purely technical.

For more information about the philosophy behind this arrangement of Organisational Units, see https://aws.amazon.com/blogs/mt/best-practices-for-organizational-units-with-aws-organizations/.

Account Structure

Org Account

Used for billing for all accounts in an organisation, to create new accounts and to manage access to all accounts, although OpenSecOpsion has been set up to allow this to be done from the `AFT-Management` account instead.

The `Org` account is unaffected by all SCPs (Service Control Policies). As little as possible should be deployed to the `Org` account, and as few people as possible should have any form of access to it. This includes read-only permissions. Any access granted, such as to Billing Administrators, should be as restricted and as least-privilege as possible.

Log Archive Account

The Log Archive account is dedicated to ingesting and archiving all logs. Storage in the `Log Archive` account is versioned, encrypted, tamper-proof, has access logging enabled, is sensibly life-cycled and saved for a total duration of 10 years.

There is also additional automation to concatenate log files to a size suitable for true low-cost long-term Glacier deep archival storage: https://github.com/OpenSecOps-AB/control-tower-log-aggregator

Audit Account

The audit account should be restricted to security and compliance teams with auditor (read-only) and administrator (full-access) cross-account roles to all accounts in the landing zone. These roles are intended to be used by security and compliance teams to

- Perform audits through AWS mechanisms, such as hosting custom AWS Config rule Lambda functions.
- Perform automated security operations, such as remediation actions.

The audit account also receives security notifications through the Amazon Simple Notification Service (Amazon SNS) service. Security administrators may wish to subscribe to these.

AFT-Management Account

The `AFT-Management` account is the central account for account and SSO management. It contains the AFT infrastructure for declarative account creation and configuration. AWS SSO management has also been OpenSecOpsed to this account.

Security-Adm Account

The `Security-Adm` account is the central account for security-related management and infrastructure. Administration has been OpenSecOpsed to this account for AWS Config, AWS GuardDuty, AWS Security Hub, AWS Inspector, AWS Detective, AWS IAM Access Analyzer, and AWS Macie.

Networking Account

The `Networking` account is intended for central configuration of network-related resources. It also hosts the central DNS domains for which subdomains are OpenSecOpsed to member accounts.

Personal Sandbox Accounts

Every engineer has a personal sandbox account for their personal use in which experiments and tests can be carried out. These accounts are named after the individual (e.g. `DiamandaGalasAccount`, `ScottyBowersAccount`, `KatyaZamolodchikovaAccount`) and have a monthly spend cap and budget. Notifications will be sent when the budgeted max spend is approached or projected to be exceeded. The personal sandbox accounts may contain anything, including VPCs, but any VPC created cannot be reached from the internet.

Non-technical users such as business administrators also have personal sandbox accounts from which they can access the services and business tools they need.

All personal accounts whether technical or non-technical are subject to the same requirements as all other accounts: they are monitored for infrastructural issues, must have contact information valid at all times, the owner must be reachable via email and Jira tickets, and they are subject to escalating SLAs.

Policy Staging Accounts

The PolicyStaging accounts are intended for the development and testing of new policies and SCPs for the Infra, Sandbox, Security, and Workloads OUs before they are applied to the live OUs.

They are builder accounts and have access modes with boundary permission restrictions for creating IAM Roles as well as more unrestricted access modes available.

Workload Accounts

Workload accounts are of two kinds: accounts for development and testing, and accounts used for running production loads. Workload account names follow a consistent convention based on the service name and the environment. They are furthermore placed in two different OUs: the Prod OU is used for production workload accounts and the SDLC (Software Development Life-Cycle) OU for all other workload accounts.

5 SOAR: Security Orchestration, Automation & Response

OpenSecOps SOAR (Security Orchestration, Automation, and Response) is built on top of AWS Security Hub and monitors and maintains the general security posture of the OpenSecOps Foundation system.

The SOAR will follow up on all infrastructural misconfigurations and deviations from best practices and generate work orders for the teams to fix them with appropriate SLAs and escalation. It will also issue incidents to responsible parties such as the builder teams, the cloud administrator team, SOC, and NOC. In this way, no security issue can be overlooked or ignored and the security posture is kept high at all times.

Tickets

OpenSecOps SOAR integrates with Jira Cloud or ServiceNow to handle the ticketing of all issues, including incidents and security controls. It can also send incidents to Microsoft Sentinel.

Jira Cloud or ServiceNow boards are intended to be used daily by the teams, not just for security-related tickets but for all their ticketing needs for project management, feature development, bug fixes, etc. The idea here is that incidents and infrastructural work orders are interwoven with and made an integral part of daily development work.

Account Tagging

All accounts in the AWS Organisation are tagged with information about the account – such as environment, project ID, and so forth – but also with contact information used by the SOAR to reach whoever is responsible for the account via email and Jira tickets.

Each account is also tagged with the ID of the team's Jira/ServiceNow board, which means that all notifications of incidents and controls will appear in the right place for teams to attend to. Jira/ServiceNow boards can be assigned per account, which is the norm for application builder teams, but they can also be shared, which is typical for infrastructural teams such as Cloud Administrators.

Issues

OpenSecOps SOAR is built around two main types of issues, incidents and controls. They are similar in many ways.

Incidents

For **incidents**, such as a server going rogue, or a user failing to authenticate too many times, we have a degree of severity and perhaps a need for immediate action. OpenSecOps SOAR will create a ticket to SOC for each incident. If the incident is severe, OpenSecOps SOAR will also snapshot and terminate any rogue instance and create a ticket to SOC for further investigation.

Controls

For **controls**, which represent ongoing security conditions such as a security group being too open, we have severity but also state: controls in a FAILED state need to be remedied with various degrees of urgency.

Automatic Remediation

If the control can be auto-remediated – there are around 30 auto-remediations available – OpenSecOps SOAR performs the remediation and notifies the team. If the control cannot be automatically fixed, OpenSecOps SOAR instead creates a ticket to the team responsible for creating the issue. Tickets are created with various levels of severity and SLAs.

Fixed Tickets are Closed Automatically

Security Hub automatically revisits all control findings periodically to re-evaluate them for changes in compliance, typically several times a day or when the configuration of the infrastructure is detected to have changed. Should a FAILED finding become PASSED, meaning that the underlying issue was fixed, the corresponding ticket will be closed automatically, relieving the teams from keeping track manually.

Statistics

For each failed control, counts and weighted statistics are kept in several dimensions (team, project, account, environment type, etc). This data could potentially be used to create dashboards of the most compliant/non-compliant accounts and teams or to determine where to best concentrate educational efforts. Actions, including auto-remediatory ones, can also be based on this data.

More Information

For more information about OpenSecOps SOAR, see Chapter 10 and also the OpenSecOps SOAR TDS and OpenSecOps SOAR SOP.

6 Networking

DNS

Domains are centrally managed from the `Networking` account. Automation is in place to automatically create subdomain OpenSecOpsions as required, for sandbox accounts or for production.

Subdomains can be declaratively OpenSecOpsed using AFT to any account, where they can be configured freely by account owners and Network Administrators. The most common use case is to set up DNS routing for application services created and managed in Builder accounts. Another use case is to provide custom service endpoints for application APIs behind AWS API Gateway.

For more information on how to configure subdomain OpenSecOpsion declaratively using AFT, see the OpenSecOps Foundation system SOP.

Default VPCs

The default VPCs in each region of an account have all been deleted. They behave slightly differently than other VPCs; for this reason, it is considered best practice to avoid using them at all.

There is automation in place to delete any new default VPCs created, for instance when AWS adds a new region. This automation runs nightly.

VPC Flow Logs

VPCs created without Flow Logs will automatically have Flow Logs enabled through auto-remediation to dump REJECTed packages to a new CloudWatch Log group using a new IAM Role and Policy.

7 Authentication & Authorisation

SSO Groups

Access to accounts is set up by first assigning people to SSO Groups. These groups have no permissions in themselves. They confer no rights but are containers whose only purpose is to group people according to job function.

However, when an SSO Group is assigned to a specific AWS account, the permissions the group is to have in that particular account must be specified using an SSO Permission Set. This means that an SSO group can have quite different permissions in different accounts.

AWS provides several predefined SSO Groups. Some of these are automatically assigned to an account at creation time (with a suitable predefined AWS permission set) to ensure that access is guaranteed for auditors and administrators.

There are also custom SSO Groups created explicitly for the OpenSecOps Foundation system. Some of these are for administrative purposes, but every builder team also has an SSO Group to link the people in a team to their specific accounts.

The custom SSO Groups all start with a common prefix such as "`OpenSecOps-`". You specify this prefix in your configuration text file for OpenSecOps Foundation. The standard AWS SSO Groups do not have a consistent common prefix; almost all of them start with "`AWS`". Below are our custom SSO Groups and the standard AWS ones we use.

OpenSecOps

We have our own SSO Groups tailored for our system and use cases which are to be used in place of the predefined AWS SSO Groups. For the very few exceptions to this rule, see the AWS section. The string "`OpenSecOps-`" in the following sections will of course be replaced with the prefix of choice for your installation.

OpenSecOps-cloud-administration

This group has unrestricted permissions in all accounts. Membership should be restricted to one or two people. All others work with dedicated roles and permission boundaries at all times.

Members have full access to all accounts using `AWSAdministratorAccess`, and, as always, `AWSReadOnlyAccess`. The latter is to be used when observing, the former only when mutating things in an account.

Again, be extremely restrictive, even paranoid, about who you put in this group. They can see and change absolutely everything, including SCPs and all permissions, but also application data and logs. There are no guardrails for Cloud Administrators. People in this group are trusted implicitly.

OpenSecOps-security-administration

Users in this group have full administrative rights, just like the users in `OpenSecOps-cloud-administration`, except that they, like all other roles, are bound by the system SCPs. Security administrators can, however, unlike Developers and Network administrators, create and manipulate IAM Users and Groups.

This group is assigned to all accounts except `Org`, which is excluded as there is no way to protect the system from rogue Security Administrators given that SCPs do not apply in the `Org` account. Thus, even Security Administrators cannot have mutation rights to the `Org` account.

The permission sets used are `SecurityAdministratorAccess` and, as usual, `AWSReadOnlyAccess`. The boundary permission policy used is `security-administrator-permission-boundary-policy`.

OpenSecOps-network-administration

Members of this group can set up and manage network-related tasks, including Route53 DNS configuration and cross-account ENIs.

This group is assigned to

- The `Networking` account,
- all `Workload OU` accounts for development, testing, and production,
- all personal `Sandbox OU` accounts, and,
- all personal `IndividualBusinessUsers OU` accounts.

The permission sets used are `NetworkAdministratorAccess` and, as usual, `AWSReadOnlyAccess`. The boundary permission policy used is `network-administrator-permission-boundary-policy`.

OpenSecOps-account-administration

Members of this group can handle on- and offboarding of new co-workers and assign them to SSO groups through which they receive the type of access they need.

They can also create and manage accounts through the use of AFT (Account Factory for Terraform) and configure SSO Group and SSO User access to those accounts declaratively.

Account Administrators also define new SSO Groups for Builder teams. Each team has a dedicated SSO Group which is used to assign `DeveloperAccess` and `AWSReadOnlyAccess` rights for team members to their designated shared team accounts. In most cases, each team will have three accounts: one for dev, test, and prod, respectively, with an SSO group named "`OpenSecOps-foo-team`" to specify the members of the team.

Account Administrators need not be deeply tech savvy, as their tasks mainly are carried out using the AFT git repository and the AWS SSO console, but they should be familiar with the principles behind Control Tower and AFT. They can call for help from the Foundation Team or Platform Team should something go wrong. They do not need to be able to fix account configuration issues themselves.

Account Administrators have access to two very specific accounts:

- the `AFT-Management` account where they can create, modify, and delete user accounts using AFT and its repositories, manage SSO groups and add and remove users from them, plus use the AFT state machines. This is where the major part of their work is done, using the SSO Permission Set `AccountAdministratorAccess`. There's also `AWSReadOnlyAccess`, to be used when just observing.

- the `Org` account, which is the management account for the AWS Organization, where they have admin access to AWS Service Catalog, mainly for observation, using the SSO Permission Set `AWSServiceCatalogAdminFullAccess`.

OpenSecOps-platform-team

This group is the glue between the Foundation Team and the developers/builders. They offer help with setting up and maintaining the development platform for application developers and builders in general.

As such, they share the wide access that developers already have, but they have no elevated powers particular only to the Platform Team.

Members of this group have read-only rights as well as `DeveloperAccess` rights to `Workloads`, `Sandbox`, and `IndividualBusinessUsers` OU accounts. The boundary permission policy used is `developer-permission-boundary-policy`.

OpenSecOps-cost-management

Members of this group can manage AWS billing, payment, budgeting, and cost reporting in the AWS Organisation account.

Members only have restricted access to one single account, the AWS Organisation account, `Org`, using the permission set `BillingAccess`.

AWS

Most of the AWS SSO Groups are not used in OpenSecOps Foundation as we have our own SSO Groups tailored for our system and use cases. The predefined AWS Group permissions and account assignments are too wide or general and thereby too unsafe for general use in OpenSecOps Foundation.

AWSControlTowerAdmins

Users in this group have full Admin rights to all AWS Control Tower core accounts and all provisioned accounts. This role is not used in OpenSecOps Foundation; instead, users on this level are assigned to the group `OpenSecOps-cloud-administration`.

AWSSecurityAuditPowerUsers

Users in this group have Power User access to all accounts for security audits. This role is not to be used in OpenSecOps; instead, users on this level are assigned to the group `OpenSecOps-security-administration`.

AWSSecurityAuditors

Users in this group have Read-only access to all accounts for security audits. This role is not to be used in OpenSecOps unless the user is a real auditor as it will give read access to *all* accounts in the organisation, including the `Org` account.

AWSAuditAccountAdmins

Users in this group have Admin rights to the cross-account audit account. This role is not to be used in OpenSecOps unless the user is a real auditor who needs mutation rights – and this is a very rare combination indeed.

AWSLogArchiveViewers

Users in this group have Read-only access to the log archive account. This role is not used in OpenSecOps.

AWSLogArchiveAdmins

Users in this group have Admin rights to the log archive account. This role is not used in OpenSecOps.

AWSServiceCatalogAdmins

Users in this group have Admin rights to the account factory in AWS Service Catalog. This role is not to be used in OpenSecOps. Instead, use the group `OpenSecOps-account-administration` which includes this permission set in the `Org` account and also allows the user to manage SSO in the `AFT-Management` account.

AWSAccountFactory

Users in this group have Read-only access to the account factory in AWS Service Catalog for end users. This role is not used in OpenSecOps.

SSO Users

Access to an account can also be given directly to individual SSO Users. This is not recommended except in the case of personal accounts in the `Sandbox` or `IndividualBusinessUsers` OUs.

SSO Users work in the same way as SSO Groups in this respect: you assign each user with one or more SSO Permission Sets to a specific account.

There is automation in place to allow this to be done from AFT, in the same way as for SSO Groups.

SSO Permission Sets

All SSO Permission Sets defined by AWS end in "`Access`" and use camel case. For clarity, we have kept to this convention for our OpenSecOps custom SSO Permission Sets. Please continue doing so.

OpenSecOps

AccountAdministratorAccess

Users with this permission set can create, modify, and delete user accounts in the AFT-Management account using AFT and its repositories, manage SSO groups and add and remove users from them, plus use the AFT state machines.

Users of this permission set aren't builders; they don't have permission to create or manipulate IAM Roles or IAM Policies in any way, meaning there is no risk of escalation of privileges. Thus this permission set does not use a boundary policy.

BillingAccess

Members of this group can manage AWS billing, payment, budgeting, and cost reporting in the AWS Organisation account.

Users of this permission set aren't builders; they don't have permission to create or manipulate IAM Roles or IAM Policies in any way, meaning there is no risk of escalation of privileges. Thus this permission set does not use a boundary policy.

DeveloperAccess

A permission set for application builders/developers. It includes read-only permissions for everything application developers need, plus mutation rights for things that AWS Copilot might deploy or access. This permission set will need to be updated as needs change; any modifications here should also be reflected in `developer-permission-boundary-policy`, the Boundary Permission used for application developers. Application builders are regionally restricted to `eu-north-1`, Stockholm.

As the users of this permission set can create roles and policies, there is a risk of escalation of privileges. Therefore, this permission set uses `developer-permission-boundary-policy` which must exist in the accounts this permission set is applied to.

SCPs require all users working with DeveloperAccess to attach the developer boundary policy to all IAM Roles they create. The SCPs also prevent Developers from modifying or deleting any such assignment.

NetworkAdministratorAccess

Grants full access permissions to AWS services and actions required to set up and configure AWS network resources, including full access to Route53.

As the users of this permission set can create roles and policies, there is a risk of escalation of privileges. Therefore, this permission set uses `network-administrator-permission-boundary-policy` which must exist in the accounts this permission set is applied to.

SCPs require all users working with NetworkAdministratorAccess to attach the network administrator boundary policy to all IAM Roles they create. The SCPs also prevent Network Administrators from modifying or deleting any such assignment.

SecurityAdministratorAccess

Grants full access permissions to AWS services and actions, including the permission to create and manipulate IAM Users and Roles. Security administrators are, though, like all other roles except `AWSAdministratorAccess`, bound by the system SCPs.

As the users of this permission set can create roles and policies, there is a risk of escalation of privileges. Therefore, this permission set uses `security-administrator-permission-boundary-policy` which must exist in the accounts this permission set is applied to.

SCPs require all users working with SecurityAdministratorAccess to attach the security administrator boundary policy to all IAM Roles they create. The SCPs also prevent Security Administrators from modifying or deleting any such assignment.

AWS

AWSAdministratorAccess

Provides full access to AWS services and resources. This is the permission set assigned to users in the `OpenSecOps-cloud-administration` group and for which SCPs make explicit exceptions for all restrictions.

AWSOrganizationsFullAccess

Provides full access to AWS Organizations. Unused in OpenSecOps Foundation.

AWSPowerUserAccess

Provides full access to AWS services and resources but does not allow management of Users and groups. Unused in OpenSecOps Foundation.

AWSServiceCatalogAdminFullAccess

Provides full access to AWS Service Catalog admin capabilities. Only used in the `Org` account, and only for members of the SSO group `OpenSecOps-account-administration`.

AWSServiceCatalogEndUserAccess

Provides access to the AWS Service Catalog end user console. Unused in OpenSecOps Foundation.

AWSReadOnlyAccess

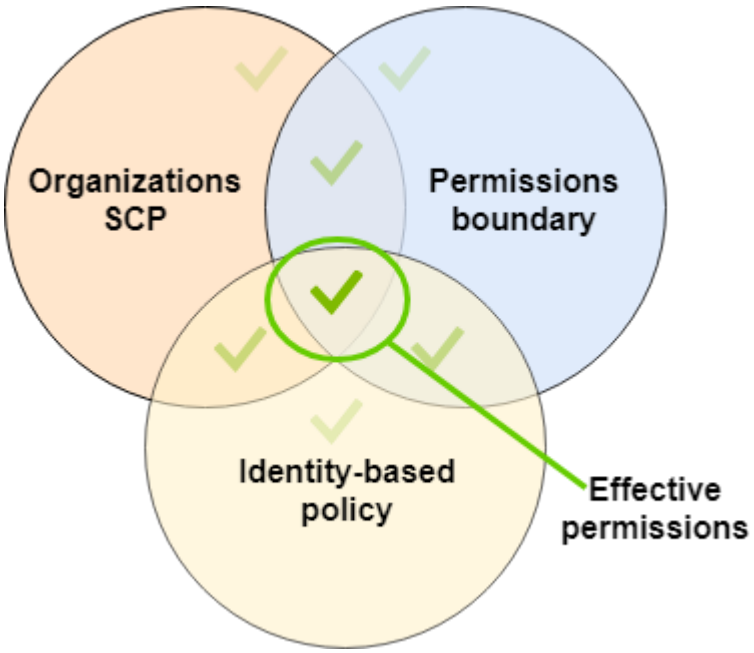
Grants permissions to view resources and basic metadata across all AWS services. Used by all OpenSecOps Foundation groups with mutation permissions to provide an alternative access method when *not* changing anything in the account.

Service Control Policies (SCPs)

Service control policies (SCPs) are a type of organisation policy used to manage permissions in an AWS organisation. SCPs offer central control over the maximum available permissions for an Organisational Unit (OU) or an account.

SCPs alone are not sufficient to grant permissions to the accounts in the organisation. No permissions are granted by an SCP. An SCP defines a guardrail to set limits to the actions that can be OpenSecOpsed to IAM and SSO users and roles. The administrator must still attach identity-based or resource-based policies to IAM and SSO users and IAM roles or the resources in an account to actually grant permissions.

The effective permissions are the logical intersection between what is allowed by
1. any SCP,
2. the IAM and resource-based policies, and
3. any Boundary Permission attached to the IAM Role or IAM User.



The SCPs in OpenSecOps Foundation have the following overall effect on all permissions except those given by `AWSSystemAdministrator`:

- they prevent the manipulation of CloudTrail, Config, and GuardDuty
- they block the creation and manipulation of IAM Users and Groups (except for System Administrators who are allowed to perform these actions)
- they prevent IAM password policies from being changed
- they block AFT custom fields from being seen and/or changed (such as budget limits or other AFT account-specific configuration)
- they prevent the tag `infra:immutable` from being set, changed or removed
- they prevent anything tagged `infra:immutable` from being changed or removed
- they prevent CloudFormation stacks and stack sets with names starting "`INFRA-`" from being created, changed, or deleted
- they prevent Lambdas with names starting "`INFRA-`" from being created, changed, or deleted
- they block changes to monthly budgets and their alarms
- they prevent SSO data such as permission sets in the AFT management account from being created, changed, or deleted.
- they prevent the creation of IAM Roles without the corresponding pre-defined Foundation boundary permission role
- they prevent the pre-defined Foundation boundary permission roles from being changed or deleted.

The individual SCPs that accomplish the above are:

protect-foundations

Protects CloudTrail, Config, and GuardDuty from creation, modification, and deletion by people other than Cloud Administrators. Also prevents IAM Users and Boundary Permission Policies from being manipulated in any way except by Cloud Administrators and Security Administrators. Protects IAM password policies and AFT custom fields.

Deployed to the Root OU.

protect-infra-immutable
Allows the tag `infra:immutable` to be set, updated, or deleted only by the system and by Cloud Administrators. Prevents the infrastructure thus tagged from modification/deletion except by the system and by Cloud Administrators.

Covered: IAM, SNS, Step Functions, EventBridge, CloudWatch, GuardDuty, IAM Access Analyzer, Inspector v2, KMS, RDS, Resource Groups, Secrets Manager.

Not coverable due to lack of tagging conditionals: Lambda, SSM, DynamoDB, DAX, SQS, EC2, S3, Security Hub.

Deployed to the Root OU.

protect-security-hub-settings
Protects the Security Hub settings for standards and controls, etc, from being modified except by Cloud Administrators.

Deployed to the Root OU.

protect-monthly-account-budget
Protects AWS Budgets named monthly-account-budget from modification by anyone but Cloud Administrators.

Deployed to the Sandbox OU and the IndividualBusinessUsers OU.

require-boundary-permissions
Requires all IAM Roles created to use a particular boundary permission policy unless the principal is a Cloud Administrator. Prevents builders from removing the boundary permission from already created roles and from modifying the pre-supplied boundary permission policies.

Deployed to the Workloads OU, the IndividualBusinessUsers OU, the PolicyStaging OU, and the Sandbox OU.

Boundary Permission Policies

AWS supports permissions boundaries for IAM entities (users or roles). A permissions boundary is an advanced feature for using a managed policy to set the maximum permissions that an identity-based policy can grant to an IAM entity. An entity's permissions boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.

OpenSecOps Foundation uses Boundary Permissions in all Builder accounts, both for application developers but also for administrators, using an SCP to enforce the use of a predefined boundary permission policy for all IAM Roles created. This ensures that escalation of privileges can’t happen no matter what the policies created by the builders say: the real limit is always defined by the mandatory, pre-supplied boundary permission and the overarching system SCPs.

The following Boundary Permission policies exist in OpenSecOps Foundation:

- `developer-boundary-permission-policy`, for application Builders.

- `security-administrator-permission-boundary-policy`, for Security Administrators.

- `network-administrator-permission-boundary-policy`, for Network Administrators.

The boundary policies must be attached to all IAM Roles created by each category of system users or role creation will fail. The policies cannot be detached, modified, or deleted.

8 Logging

System Logs

System logs from AWS Control Tower, including CloudTrail and AWS Config logs for all accounts and regions, are automatically collected and deposited in S3 buckets in the system-wide account for logging, `Log Archive`. This also applies to GuardDuty and Security Hub findings.

Local CloudWatch Logs

All local CloudWatch Log groups in an account are automatically streamed to the Log Archive account, where they are processed by log analysis software. This includes VPC Flow Logs and all logs from services that can log to CloudWatch.

This is also the way application logging is done: just make sure your application logs to CloudWatch and its logs will automatically be picked up, no matter their type. ECS/Fargate logs to CloudWatch Logs by default.

The retention time of CloudWatch log groups lacking an explicit setting will be set to 14 days by automation.

Load Balancer & CloudFront Logs

In addition to CloudWatch logs, any S3 bucket created locally to contain Elastic Load Balancer (ALB/ELB/NLB) or CloudFront logs will be automatically detected and its contents replicated to the `Log Archive` account. The bucket owner will also receive a Jira incident ticket to inform them that logs now are being streamed from their bucket, and suggested further steps they might want to take in the local account.

There is an auto-remediation for Elastic Load Balancers that creates an S3 bucket and sets up access logging for the load balancer to that bucket if missing. This means that in OpenSecOps Foundation, all load balancer access logs are guaranteed to be aggregated without any user intervention whatsoever.

Log Integrity & Retention

All logs in the `Log Archive` account are versioned, encrypted, tamper-proof, have access logging enabled, are sensibly life-cycled and saved for a total duration of 10 years by default. This can be customised as you wish.

Log Post-Processing

There is automation to concatenate log files to a size suitable for true low-cost long-term Glacier deep archival storage. For full details, see https://github.com/PeterBengtson/control-tower-log-aggregator.

9 Incident Management

All incidents are handled by OpenSecOps SOAR (Security Orchestration, Automation, and Response), which is built on top of AWS Security Hub and also monitors and maintains the general security posture of the OpenSecOps Foundation system.

The SOAR will issue incidents to responsible parties such as the builder teams, the cloud administrator team, SOC, and NOC. In this way, no security issues can be overlooked and the security posture is kept high at all times.

For more information on the operation of the OpenSecOps SOAR and what it means for daily work in the system, see Chapter 6, the OpenSecOps AWS Account Properties SOP, and also the OpenSecOps SOAR TDS and SOP, respectively.

Incident Tickets

OpenSecOps SOAR will automatically create tickets for all system incidents. These tickets will be assigned to the team responsible for the account and are subject to SLAs and escalation.

Builders can also create tickets by leveraging the mechanisms described under "Programmatic Creation of Incidents" below.

Programmatic Creation of Incidents

In the OpenSecOps Foundation system, CloudWatch Alarms can be used to create incidents on both the infrastructural level and the application level simply by naming them appropriately, such as `INFRA-FooBar-Alarm-CRITICAL`. This unifies incident handling across the entire system and all its applications.

There is also an API Gateway endpoint for programmatically creating incidents. This is useful for interfacing with third-party software such as ELK.

For the details, see the OpenSecOps SOAR SOP and the OpenSecOps AWS Account Properties SOP.

Unified Incident Handling

It is important, indeed imperative, that all incident handling in the OpenSecOps Foundation system, be it infrastructural or application-related, goes through the mechanism described in this chapter. To ensure the long-term scalability and manageability of the system, there must not be multiple incident integration points or multiple incident management systems.

For more information, see the OpenSecOps SOAR SOP.

10 Foundational CloudFormation Stacks & StackSets

The following list of StackSets and Stack does not include any templates installed by Control Tower, only those that have been added after Control Tower installation.

Also not included are templates managed by SAM and other tools for deployment purposes as these are administrative in nature and add no functionality in themselves.

It should also be noted that all CloudFormation template names (and thus all SAM project names) begin with "`INFRA-`". This gives protection through various system SCPs for the infrastructure deployed through the template, and against modification of the template itself.

Org Account StackSets

All StackSets in the Org account are deployed in `eu-north-1`. The Deployment column in the following table indicates to which regions and/or Organisational Units (OUs) the resulting Stacks are deployed.

Template Name	Description & Notes	Deployment
`INFRA-aft-sso-account-configuration`	This configures the SSO permissions of an account. Responds to an SNS message sent from an AFT customisation to assign SSO permission sets to an account using custom AFT data.	
`INFRA-CloudWatch2S3-additional-account`	Continuously dump all matching CloudWatch Log groups to a bucket in a central account for long-term storage.	
`INFRA-customizations-for-aws-control-tower`	CfCT: customizations-for-aws-control-tower Solution.	
`INFRA-default-vpc-remover`	This removes all default VPCs in all regions for a newly created account. Also runs periodically to remove VPCs in any region AWS may add in the future.	
`INFRA-detect-bucket-lifecycle`	Send all S3 CreateBucket and DeleteBucket events to a custom event bus in the organisation account so we can detect when a bucket contains logs and replicate it to the corresponding Log Archive bucket.	
`INFRA-detect-bucket-tagging`	Send all S3 PutBucketTagging events to a custom event bus in the organisation account	
`INFRA-detect-log-buckets`	Whenever a bucket is created, we want to detect whether it contains log files from CloudFront or AWS Elastic Load Balancing Services. If so, we replicate the contents of	

Template Name	Description & Notes	Deployment
	such buckets to the Log Archive account.	
INFRA-detect-stack-drift	Creates a weekly report of stack and stack set drift in a region.	
INFRA-diskMember	CloudFormation template for deploying member IR role and policy	
INFRA-enable-ebs-encryption-by-default	This CloudFormation stack turns on automatic EBS volume encryption for an account.	
INFRA-iam-password-policy	Sets an IAM Password Policy for an account. It uses a custom resource to manage the policy. Note that IAM password policies are global, and this will apply to all regions.	
INFRA-instance-port-report	Creates a weekly report of open instance ports across the whole organisation.	
INFRA-limit-log-group-retention	Sets the retention time of all CloudWatch LogGroups with unspecified (=infinite) retention to two weeks.	
INFRA-local-alarm-events-to-sec-hub-bus	This template is used to create a StackSet in the Org account, deploying to all accounts in the two regions we use. For the security account, this template will define an EventBusPolicy for the default event bus that allows other accounts in the organisation to use PutEvents to transfer events to the security account. For all other accounts, it will set up a role and a rule to transfer local alarm events to the security account.	
INFRA-new-ct-account-created-sns-topic	This stack creates an SNS Topic which receives events signalling the completion of the successful creation of an account via AWS Control Tower. The topic is intended to be used for post-creation account configuration. Deploy in the main region. The Topic can be accessed by any principal. Note that the topic can't be encrypted using the standard SNS key, as CloudWatch/EventBridge won't be able to publish to it; the reason being that the standard key doesn't permit Decrypt and GenerateDataKey.	
INFRA-s3-log-replication-source-account-role	Creates the IAM Role required for S3 log bucket replication to the Log Archive account.	
INFRA-sec-hub-custom-event-bus	Sets up a custom Event bus in the organisation account, allowing all accounts in the organisation to post events to it which then are picked up in the Org account. Used for syncing SecHub Controls between all accounts in a region, for sending Security Hub findings for all accounts (from the Security account where they are aggregated), and for	

Template Name	Description & Notes	Deployment
	S3 bucket tagging events to detect requests for public S3 buckets.	
INFRA-sec-hub-role	Defines roles for Security Hub-related access and for the custom event bus.	
INFRA-soar	This project contains nested state machines for SOAR processing, ticketing, and auto-remediation.	
INFRA-soc-incident-when-s3-tag-applied	Whenever certain tags are applied to an S3 bucket, this SAM project creates incidents for SOC to investigate.	

Org Account Stacks

The Regions column in the following table indicates in which regions the Stack is deployed.

StackSets can’t be automatically deployed to the Org account. Thus, some templates deployed as a StackSet (listed in the previous section) to multiple member accounts and/or OUs have also been deployed manually, as a separate step, to the Org account.

Template Name	Regions	Description & Notes
INFRA-aft-sso-account-configuration	eu-north-1	This configures the SSO permissions of an account. Responds to an SNS message sent from an AFT customisation to assign SSO permission sets to an account using custom AFT data.
INFRA-CloudWatch2S3-additional-account	eu-north-1 us-east-1	Continuously dump all matching CloudWatch Log groups to a bucket in a central account for long-term storage.
INFRA-customizations-for-aws-control-tower	eu-north-1	CfCT: customizations-for-aws-control-tower Solution.
INFRA-default-vpc-remover	eu-north-1	This removes all default VPCs in all regions for a newly created account. Also runs periodically to remove VPCs in any region AWS may add in the future.
INFRA-detect-bucket-lifecycle	eu-north-1 us-east-1	Send all S3 CreateBucket and DeleteBucket events to a custom event bus in the organisation account so we can detect when a bucket contains logs and replicate it to the corresponding Log Archive bucket.
INFRA-detect-bucket-tagging	eu-north-1 us-east-1	Send all S3 PutBucketTagging events to a custom event bus in the organisation

Template Name	Regions	Description & Notes
		account
INFRA-detect-log-buckets	eu-north-1 us-east-1	Whenever a bucket is created, we want to detect whether it contains log files from CloudFront or AWS Elastic Load Balancing Services. If so, we replicate the contents of such buckets to the Log Archive account.
INFRA-detect-stack-drift	eu-north-1 us-east-1	Creates a weekly report of stack and stack set drift in a region.
INFRA-diskMember	eu-north-1	CloudFormation template for deploying member IR role and policy
INFRA-enable-ebs-encryption-by-default	eu-north-1 us-east-1	This CloudFormation stack turns on automatic EBS volume encryption for an account.
INFRA-iam-password-policy	eu-north-1	Sets an IAM Password Policy for an account. It uses a custom resource to manage the policy. Note that IAM password policies are global, and this will apply to all regions.
INFRA-instance-port-report	eu-north-1	Creates a weekly report of open instance ports across the whole organisation.
INFRA-limit-log-group-retention	eu-north-1	Sets the retention time of all CloudWatch LogGroups with unspecified (=infinite) retention to two weeks.
INFRA-local-alarm-events-to-sec-hub-bus	eu-north-1 us-east-1	This template is used to create a StackSet in the Org account, deploying to all accounts in the two regions we use. For the security account, this template will define an EventBusPolicy for the default event bus that allows other accounts in the organisation to use PutEvents to transfer events to the security account. For all other accounts, it will set up a role and a rule to transfer local alarm events to the security account.
INFRA-new-ct-account-created-sns-topic	eu-north-1	This stack creates an SNS Topic which receives events signalling the completion of the successful creation of an account via AWS Control Tower. The topic is intended to be used for post-creation account configuration. Deploy in the main region. The Topic can be accessed by any principal. Note that the topic can't be encrypted using the standard SNS key, as CloudWatch/EventBridge won't be able to publish to it; the reason being that the standard key doesn't permit Decrypt and GenerateDataKey.
INFRA-s3-log-replication-source-account-role	eu-north-1	Creates the IAM Role required for S3 log bucket replication to the Log Archive

Template Name	Regions	Description & Notes
		account.
INFRA-sec-hub-custom-event-bus	eu-north-1 us-east-1	Sets up a custom Event bus in the organisation account, allowing all accounts in the organisation to post events to it which then are picked up in the Org account. Used for syncing SecHub Controls between all accounts in a region, for sending Security Hub findings for all accounts (from the Security account where they are aggregated), and for S3 bucket tagging events to detect requests for public S3 buckets.
INFRA-sec-hub-role	eu-north-1	Defines roles for Security Hub-related access and for the custom event bus.
INFRA-soar	eu-north-1	This project contains nested state machines for SOAR processing, ticketing, and auto-remediation.
INFRA-soc-incident-when-s3-tag-applied	eu-north-1 us-east-1	Whenever certain tags are applied to an S3 bucket, this SAM project creates incidents for SOC to investigate.

Security-Adm Account Stacks

TO DO

Log Archive Account Stacks

TO DO

Member Account Stacks

TO DO

11 Additional System Management Tools

AWS Core SSO Configuration

As there is no good place in AWS Control Tower to set up the core configuration of SSO, we use an open-source utility for this purpose. For full details, see https://github.com/OpenSecOps-AB/AWS-Core-SSO-Configuration.

12 Builders & Application Development

Account Recommendations

Sole Ownership

Each account in the OpenSecOps Foundation system should be owned by exactly one team, or in the case of personal sandbox accounts, by exactly one person. If you want a system that is manageable in the long run there should be no shared ownership of infrastructure within an account, nor should there be more than one team deploying to the same account.

The team or person owning the account is the sole contact point for the account, technical or otherwise. This is important for incident response, for enforcing SLAs to do with detected security issues, in attack and DR situations, etc.

The above applies to application Builder accounts as well as to system-level Builder accounts. (It of course also applies to any repos a team uses, which should not be shared between teams in terms of commits.)

One Service and One Environment per Account

Each account should host only one service and one environment (e.g., DEV, TEST, PROD). Security and compliance rules, requirements, controls, and auto-remediations vary with the type of environment.

NB: A "service", in this context, may consist of subservices grouped together in a functional whole to form an "application" exposed to the world. The application would be exposed to internal or external consumers – but not necessarily its component subservices.

Services should not share accounts even if they are managed by the same team. A service may sometimes need to be reassigned to another team: this should never require migration.

This means that each team should use *at least* three accounts per service: one account for managing the product's development cycle, another account for testing it, and another account for running the product in production. Under no circumstances should an account be used for multiple products or environments.

Email List

Each team should have a designated email distribution list to which automation can send email. Such email must reach all team members. There must be a process in place for each team to examine all email and to keep the distribution list up to date at all times.

Ticketing and SLAs

Each team should also have a process for dealing with tickets in Jira. Again, there must be a process to involve the whole team, and the process must consider the urgency of the issues. Some issues must be dealt with immediately, others can wait a few days.

Infrastructural fixes should be allowed, when their severity level demands it, to take full priority over application feature development. The team manager should

be prepared to allocate time accordingly and at times stop feature development *for the whole team* until critical issues have been fixed.

Deployment

Container Recommendations

To ensure the feasibility of retargeting to another cloud provider without application redesign, it is strongly recommended that Docker containers:

- do not assume anything about the system that orchestrates them.
- do not assume they are being orchestrated by a particular orchestrator such as ECS, EKS, GKE, Cloud Run, Docker Swarm, etc.
- always interface to the world in terms of only two things:
  - ports, and,
  - environment variables.
- directly interface to platform-independent technology (such as SQL databases, Redis, Kafka, etc) using environment variables and ports.
- use platform-specific services (such as AWS DynamoDB) but should use a library interface to abstract away the specifics and allow other technologies to be substituted in a platform-agnostic manner.

The above also means that applications can always and should always be tested and run locally during development. The platform independence of containers must be maintained at all times and in all stages of development.

Elastic Container Registry (ECR)

The counterpart to Docker Registry on AWS is ECR, the Elastic Container Registry. Docker images are uploaded to ECR for storage and to be automatically scanned for security vulnerabilities. It is from ECR that ECS will fetch the images to deploy.

The following automations are in place for ECR registries:

- ECR registries not configured to scan for security flaws will be set by auto-remediation to scan uploaded images using enhanced scanning. The scanning is done once for each image, at the time it is uploaded. The results can be examined in the console; they can also be used as the basis of alarms and further automation.
- ECR registries without lifecycle configuration will be configured by auto-remediation to keep only the last uploaded image.
- OpenSecOps does not require ECR image tags to be set immutable as this might interfere with workflows using the `latest` tag. Should those workflows not be used, there is an auto-remediation available to enforce immutable ECR tags. However, it is currently disabled.

The above applies to all ECR registries and their images, no matter how they are created and/or populated.

Elastic Container Service (ECS)

The following automations are in place for ECS clusters:

- ECS services must not be assigned public IPs automatically by the cluster. If this feature is enabled, automation will disable it.

- If an ECS cluster doesn’t have Container Insights enabled, auto-remediation will turn it on. Results collected by Container Insights can be seen in the console, and it is also possible to build CloudWatch alarms, incidents and further automation on them.

13 Databases

Database Logging

Databases will be discovered by automation. Logging, if not already configured, will automatically be enabled for Aurora Postgres DBs (including its Serverless variants) and plain vanilla RDS Postgres DBs. All database logs are visible to the developers in their account and are also automatically aggregated to the Log Archive account for long-term archival.

14 File Storage

S3 Buckets

S3 buckets are file servers. There is SOAR automation to protect them from inadvertent and/or unauthorised access.

World Readable and/or Writable S3 Buckets

Automation will close all buckets open to the world unless tagged appropriately.

S3 buckets configured publicly readable will be automatically closed for public access unless the tag "`soar:s3:request-publicly-readable`" is present on the bucket.

S3 buckets configured publicly writable will be automatically closed for public access unless the tag "`soar:s3:request-publicly-writable`" is present on the bucket.

When either or both of these tags appear, SOC will be notified of the request via an incident notification in Jira. SOC will then contact the owner of the S3 bucket about their reasons for having created a file server open to the world.