OpenSecOps Foundation

Installation Manual

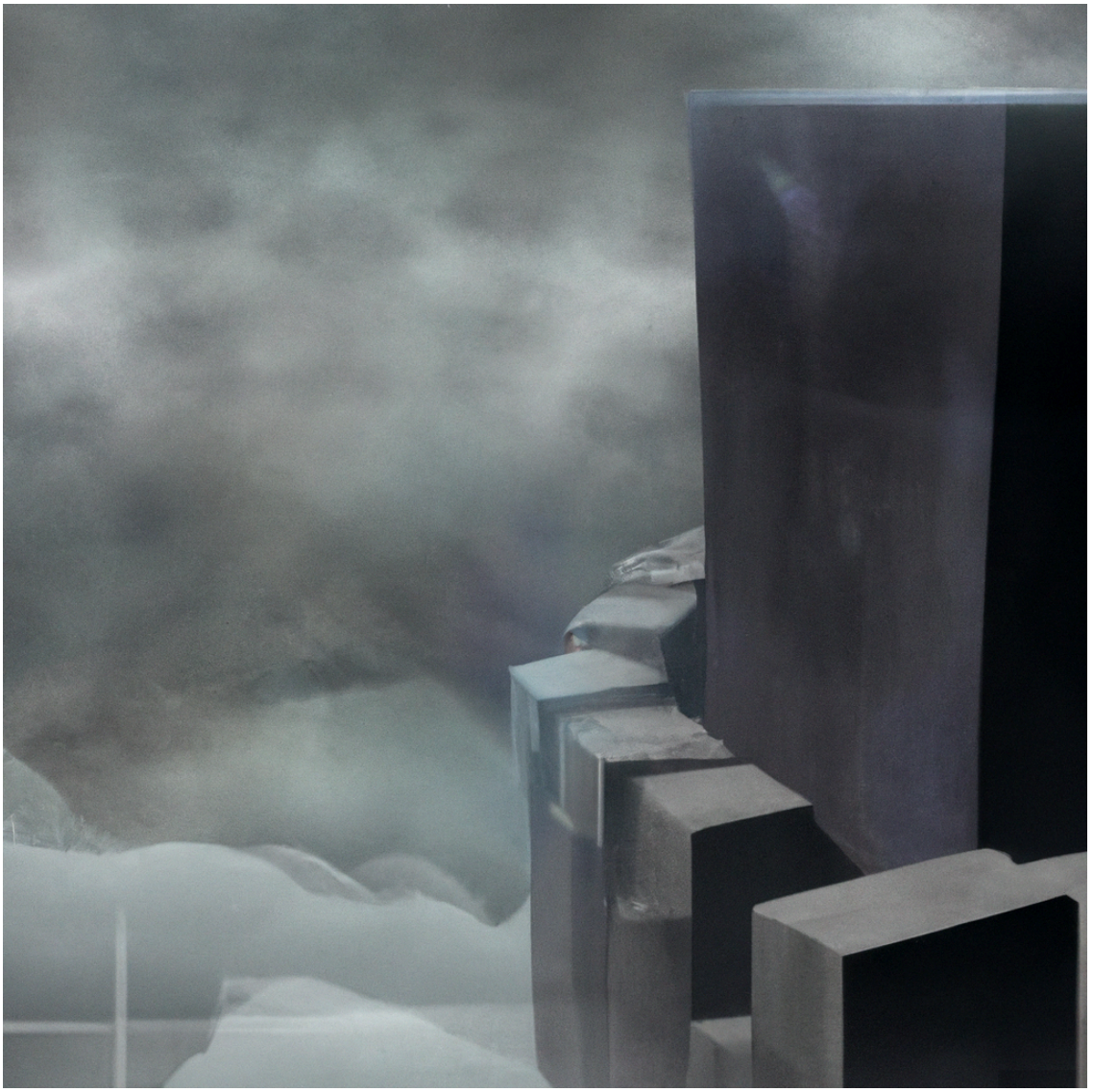


Table of Contents

Document Versions

Version	Date	Changes	Author
1.3	2025-04-07	Replaced “Delegat” with “OpenSecOps” throughout	Peter Bengtson
1.2	2025-03-13	Added section about Resource Control Policies	Peter Bengtson
1.1	2024-10-17	Added section about Support and SLAs	Peter Bengtson
1.0	2024-09-22	First version	Peter Bengtson

Introduction

This manual describes the OpenSecOps Foundation installation procedure.

Audience

This installation manual is intended for an experienced system administrator with AWSAdministratorAccess permissions for all accounts in the system.

To you, an instruction like “Install Account Factory for Terraform (AFT)”, which actually is part of the setup, is something you take in your stride. You should be intimately familiar with:

- Zsh
- The AWS CLI
- Git
- GitHub / GitLab / BitBucket
- CloudFormation
- Terraform, including common strategies for state files
- SAM, the AWS Serverless Application Model and its CLI

Time Required

A very experienced system administrator should be able to install Foundation in 2-3 days under ideal conditions. However, several factors could impact this timeline:

1. Familiarity with AWS services: The administrator would need to be very familiar with AWS Control Tower, AFT, IAM, and other related services.

2. Existing environment complexity: If installing into an existing AWS Organization, additional time might be needed for adapting the current setup.

3. Troubleshooting: Any issues that arise during installation could significantly extend the timeline.

4. Organisation-specific customizations: If extensive customisation is needed, this could add time to the process.

5. Approval processes: In larger organisations, obtaining necessary approvals for changes might extend the timeline.

6. Testing and verification: Thorough testing of all components after installation is crucial and could take additional time.

A more conservative estimate might be 3-5 days for most scenarios, allowing for potential complications and thorough testing. However, for a highly experienced administrator in a well-prepared environment with minimal customisation needs, 2-3 days is achievable.

Prerequisites

- You must have full Administrator privileges in all accounts, including the administrative root account. Nothing must prevent your full access. (You’re Sudo Sue.)

- The system can be brand new, in which case there is only a single account, or it can be an existing system, potentially with a large number of member accounts. This installation guide is written primarily for the new installation scenario.

- The procedure is the same for a pre-existing system setup. The only thing that has to be adapted is the AFT setup, to include the existing accounts. You might also have to adjust things already in place, such as the OU arrangement.

Preparations

Create a mailing list for your account-related information. We suggest you simply call it "`accounts@yourcompany.com`". You can then use + notation for all account email addresses.

Installing AWS Control Tower

1. Create a new account or use an existing one. We will refer to this account as your "Org" account. Don't use the company's name for this account. If you have a choice, we recommend that you name or rename the account "Org", and its email address "`accounts+org@yourcompany.com`"

2. Log into the Org account that is, or is to be, the administrative account for your AWS Organization *using root credentials*. The username is the email address of the account. Set up MFA when you log in the first time if the account is new.

3. Go to the region which will be your main region.

4. Go to AWS Control Tower.

5. Click on "Set up Landing Zone".

6. Select the regions in which you want to operate. We strongly recommend including `us-east-1`, even if you don't intend to use it. Access can be restricted by the SCPs to be installed later.

7. Enable Region deny.

8. Accept the default OU names.

9. Set up the Log Archive email: "`accounts+log-archive@yourcompany.com`"

10. Set up theAudit email: "`accounts+audit@yourcompany.com`"

11. Let AWS Control Tower set up AWS account access configuration.

12. Enable CloudTrail.

13. Set up Log Retention.

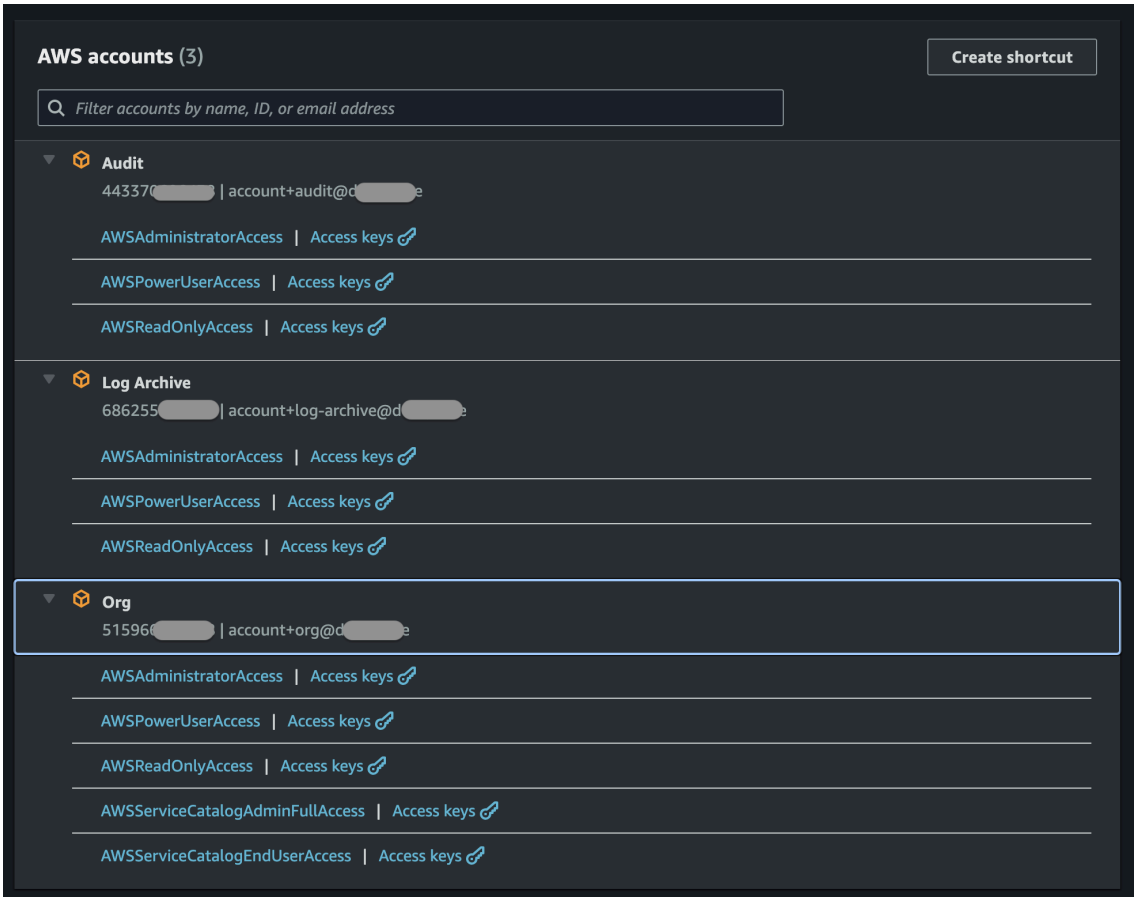
14. Do NOT enable KMS encryption. It's not necessary and the log files are tamper-proof. More importantly, it can't be enabled if the centralised log collection from member accounts is to work. Moreover, AWS Control Tower does not support multi-Region keys or asymmetric keys.

15. Verify all information on the next page.
16. Click on “Set up landing zone”.
17. Wait until installation is complete. This will take about an hour.
18. Accept the invitation to the Org account (“Invitation to join AWS IAM Identity Center”) and set a long password. Store it in a secure location.
19. Confirm the SNS subscriptions via email.

Creating an installation SSO User

You should now create an SSO User for yourself. NB: After this user has been created, you no longer should use the root account user, only the SSO User you’re now creating. The rest of the installation should be done exclusively with the SSO user.

1. Click on “Create user” in IAM Identity Center.
2. Enter your personal details.
3. Select all groups.
4. Create the user.
5. Confirm the email invitation.
6. Log in to AWS using the link in the email. It’s typically of the form “`https://d-c12345abc678.awsapps.com/start/`”
7. Set up MFA during the login.
8. Your AWS Account selection screen should now look something like this:



9. Log into the Org account using AWSAdministratorAccess. From now on, you should only use your new SSO user.

Configure SSO using IAM Identity Center

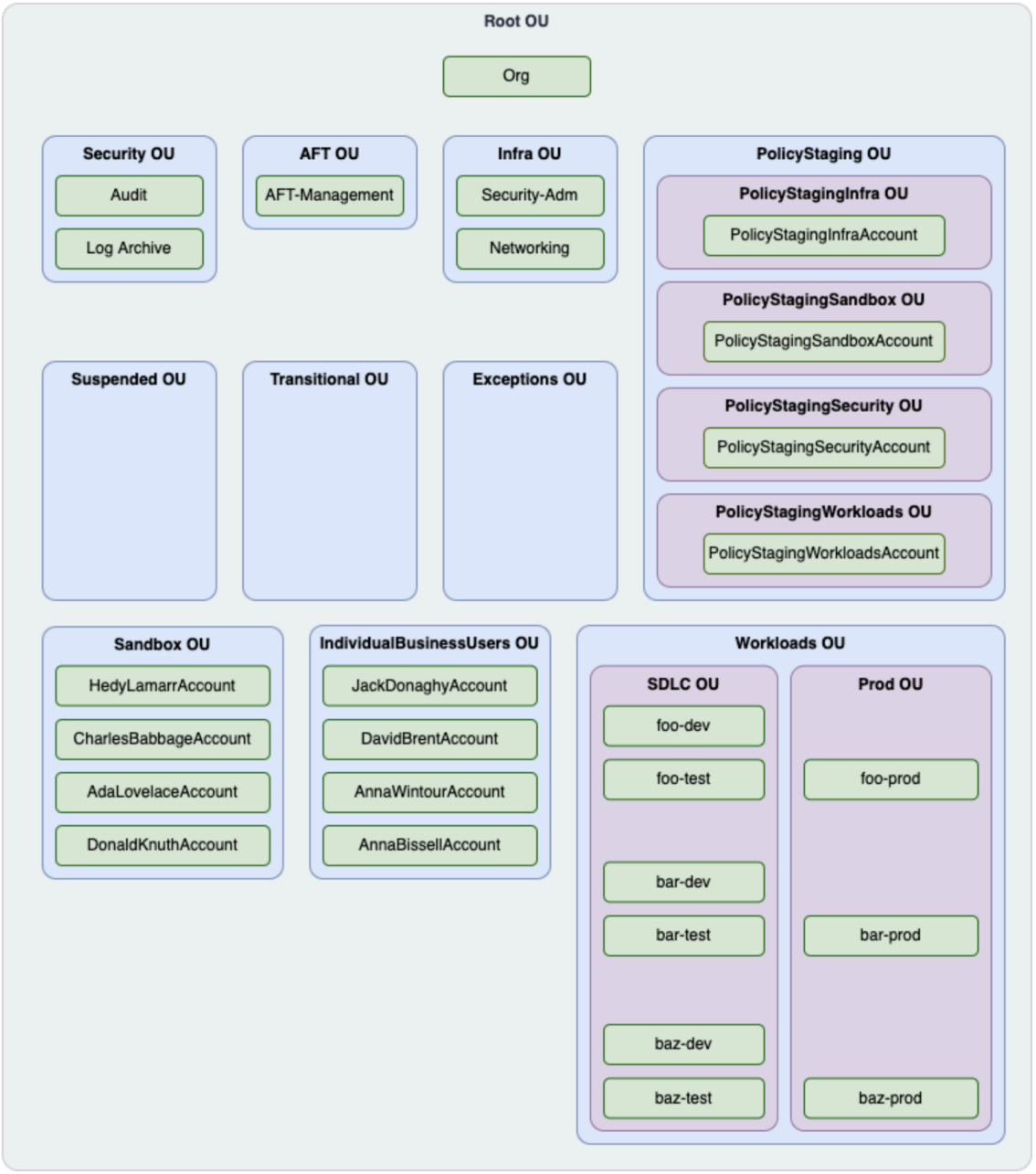
1. Go to the IAM Identity Center (formerly known as AWS SSO).
2. Go to Settings.
3. Configure the Identity Source, if necessary.
4. Configure your desired MFA behaviour. Recommendations:
    a. "Every time they sign in (always-on)"
    b. "Require them to register an MFA device at sign in"
5. Customise the AWS access portal URL. By default, it's something like "`https://d-c12345abc678.awsapps.com/start`". We recommend changing it to your company name or function, e.g. "`https://yourcompany.awsapps.com/start`".
6. You may want to set the session duration for the AWSAdministratorAccess permission set to 8 hours whilst working with the installation. Just be sure to set it back to 1 hour when you're done.

Configure Control Tower Account Factory VPC settings

1. With your SSO user, go to AWS Control Tower, then to Account factory.
2. Click Edit and configure the settings.
3. We recommend that you uncheck all boxes that automatically create VPCs for new accounts.

Setting up Organisational Units

Before we can install Account Factory for Terraform (AFT), we need to set up a best-practice hierarchy of Organisational Units (OUs). We recommend the following:

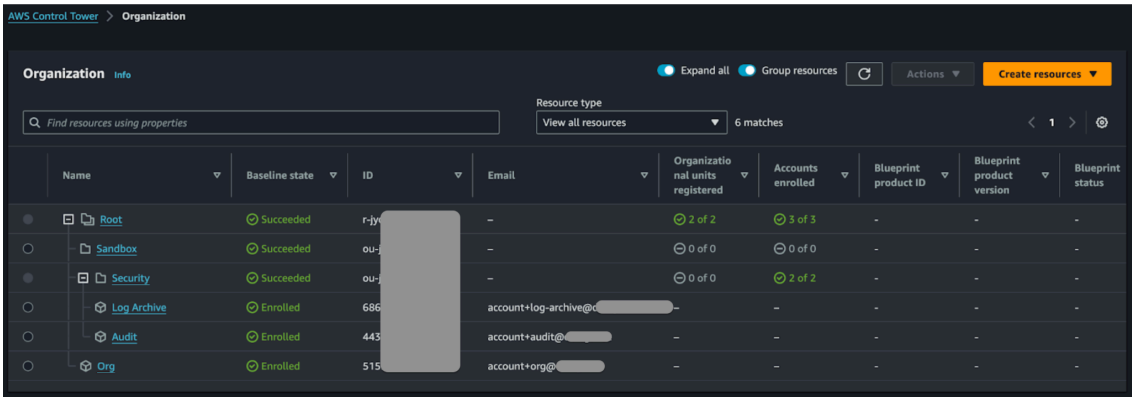


For a detailed explanation of this arrangement, please refer to the OpenSecOps Foundation TDS.

New Systems

If the system is brand new, you will only have the three accounts “Org”, “Audit”, and “Log Archive”. This means that there is nothing to migrate, and you can proceed to the OU setup without further ado.

In this scenario, logging in to Org using your administrator SSO user and going to AWS Control Tower and then on to “Organization”, you’ll see the following, after enabling “Expand all” and “Group resources”:



Existing AWS Organizations

If the system already exists, you might have many more than three accounts, in which case you need to have a conversation with the system administrators to align and modify the OU structure in a way that doesn’t break the existing setup.

It’s entirely possible to transition to the above OU structure without breaking production environments, but this must be done with care. But as you’re undoubtedly experienced enough to be trusted with full system access, your expertise will include how to perform such migrations.

In this scenario, you’ll obviously see a lot more elements when viewing the AWS Organization. You can still create the proposed OU structure, then gradually migrate existing accounts into it.

Scope

It’s important to point out that we’re only setting up the OUs at this point; we’re not creating any new accounts quite yet. In order to install AFT later – which will handle all account creation and mutation in OpenSecOps Foundation, we must manually create an account for AFT (“AFT-Management”). At this point, however, we only need to set up the OUs according to the diagram above, that is, everything that’s not green.

Create the OUs

1. After the Control Tower installation, these OUs already exist:
    a. Root (contains the Org account)
    b. Sandbox (empty)
    c. Security (contains Log Archive and Audit)

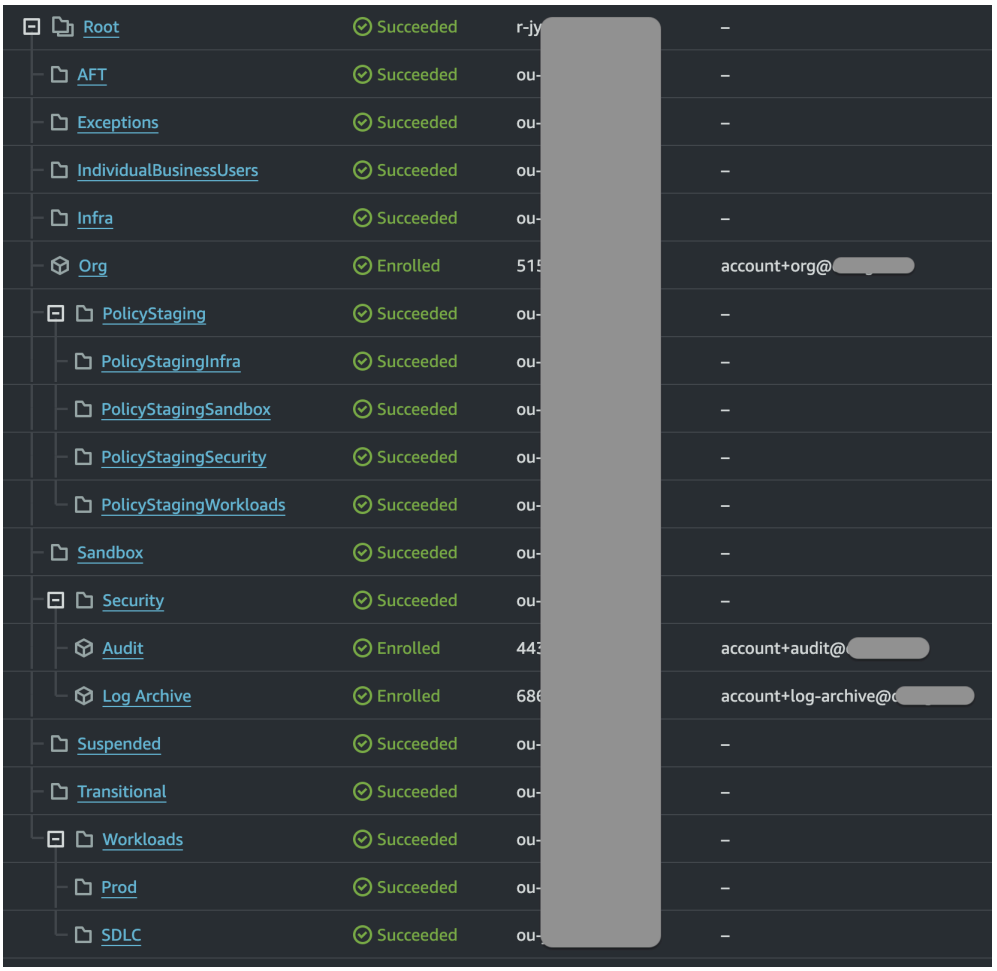
2. Create the following OUs under Root:
    a. AFT
    b. Infra
    c. Suspended
    d. Transitional
    e. Exceptions
    f. IndividualBusinessUsers
    g. PolicyStaging

   h. Workloads

3. Under PolicyStaging, set up the following nested OUs:
   a. PolicyStagingInfra
   b. PolicyStagingSandbox
   c. PolicyStagingSecurity
   d. PolicyStagingWorkloads

4. Under Workloads, set up the following nested OUs:
   a. SDLC
   b. Prod

5. The OU structure as viewed in AWS Control Tower should now look like this:



   If there already existed an AWS Organization setup, you might have more OUs and accounts.

Account Factory for Terraform (AFT)

Preparations

Installing AFT is a fairly long process with quite a few steps. The full instructions are here, on Hashicorp's site, in the form of a tutorial:

https://developer.hashicorp.com/terraform/tutorials/aws/aws-control-tower-aft

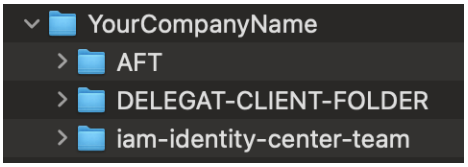
However, as parts of the installation described in the tutorial already have been done – such as creating OUs – we have streamlined the process considerably by providing tailored repos for your use.

Take a look at the Hashicorp instructions now. Then follow the instructions below rather than those in the tutorial until you get to the section called "Deploying AFT" where further instructions will tell you how to proceed using the Hashicorp instructions.

First, we need to do four things:

Set up a directory structure for administration

Create the following directory structure on your laptop or workstation, or whatever you're using for this installation procedure. It should look like this:



The directories can be empty for now. However, when installation is complete, you should copy the entire directory structure and hand it over to your client so they can continue administrating the system. "YourCompanyName" can be anything you like, such as "OpenSecOps Products", "SystemAdministration", etc.

Create an administration account for AFT

1. The complete instructions are here: https://docs.aws.amazon.com/controltower/latest/userguide/provision-as-end-user.html

2. We've boiled them down for you to this:
   a. With your SSO user in the Org account, go to Service Catalog
   b. Click on Products
   c. Select AWS Control Tower Account Factory, then choose the Launch product button
   d. Provisioned product name: "`AFT-Management`"
   e. AccountEmail: "`accounts+aft-management@yourcompany.com`"
   f. AccountName: "`AFT-Management`"
   g. ManagedOrganizationalUnit: "`AFT`"
   h. SSOUserEmail: "`accounts+org@yourcompany.com`"
   i. SSOUserFirstName: "`AWS Control Tower`"
   j. SSOUserLastName: "`Admin`"

3. Important: the last three items, SSOUserEmail, SSOUserFirstName, and SSOUserLastName should be constant when creating all accounts. This applies even when subsequently creating accounts using AFT. DO NOT enter the details of any real person here. Access permissions to real users will be assigned using AFT and an open-source OpenSecOps extension later on.

4. Click on "Launch product".

5. Wait until the process is complete.

6. You should have received a welcome email from AWS for the new account.

7. Click on “Provisioned products”. This is where all your accounts will appear - that is, except Org, Log Archive, and Audit which are handled separately.

8. Check the AWS access portal, which now should list AFT-Management as an available account.

9. Go to AWS Control Tower and verify that the new account appears where it should.

Enable trusted access for Service Catalog

Obtain the access keys for the Org from the AWS access portal, then paste them into a terminal, then type:

```
aws organizations enable-aws-service-access --service-principal
servicecatalog.amazonaws.com
```

This will enable trusted access for Service Catalog in your AWS organization.

Enable EC2 access

1. Log into the Org account using your SSO user.
2. Go to EC2 and start three t3 instances. There’s no need to create PEM files or allow access to the instances in any way.
3. Log into the AFT-Management account using your SSO user.
4. If the account lacks a VPC, create a default one.
5. Go to EC2 and start three t3 instances. There’s no need to create PEM files or allow access to the instances in any way.
6. Let these six instances run for a while. It will register the accounts for the use of EC2, and it will also push up your quotas.
7. After a few hours, or the next morning, terminate all six instances.
8. Delete any VPC you created.

Setting up the Git Repositories

You now need to decide where to store the Git repositories AFT uses. AWS has discontinued CodeCommit. You *must* use an external provider or a local Git server unless you already are using CodeCommit in your organisation. Chances are you aren’t, but if you are, use it – it’s easier.

Your choices are limited: GitHub, GitHub Enterprise Server, or BitBucket. You can find more information about how to set up what is referred to as “alternative VCS” here:

https://docs.aws.amazon.com/controltower/latest/userguide/aft-alternative-vcs.html

Read all instructions carefully before you begin and plan ahead.

Forking the AFT repos

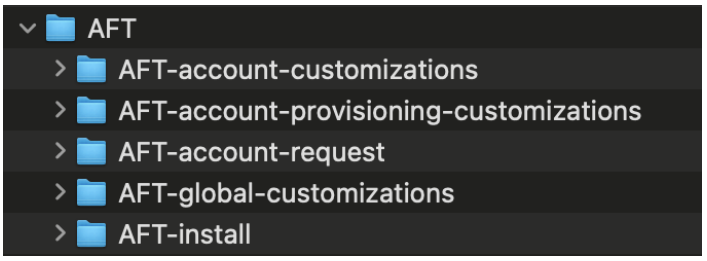
You need to fork five repositories from OpenSecOps’s GitHub organisation to your own location on the external Git provider of your choice. Our repos are in their turn forked from Hashicorp originals, with modifications to make installation of OpenSecOps Foundation easier.

Follow the Hashicorp tutorial instructions, but when you fork the repositories, follow these instructions instead:

Fork the following repos *to your external provider*, given your organisation's name on the provider is "AcmeInc":

1. Fork https://github.com/OpenSecOps-Org/Foundation-AFT-install to "`AcmeInc/AFT-install`".
2. Fork https://github.com/OpenSecOps-Org/Foundation-AFT-account-request to "`AcmeInc/AFT-account-request`".
3. Fork https://github.com/OpenSecOps-Org/Foundation-AFT-account-customizations to "`AcmeInc/AFT-account-customizations`".
4. Fork https://github.com/OpenSecOps-Org/Foundation-AFT-account-provisioning-customizations to "`AcmeInc/AFT-account-provisioning-customizations`".
5. Fork https://github.com/OpenSecOps-Org/Foundation-AFT-global-customizations to "`AcmeInc/AFT-global-customizations`".

You should then clone these five repos from your external provider – *not from the originals* – to your local computer's AFT folder, which then should look like this:



Tailoring AFT-install

When the Hashicorp guide tells you to go to the "`learn-terraform-aws-control-tower-aft`" repository, you will find it under the much better name "`AFT-install`". (This is, after all, not a learning situation.)

The files to modify here are:

1. `main.tf`
2. `terraform.tfvars`

To VPC or not to VPC

`main.tf` has a configuration parameter called `aft_enable_vpc`. It is by default set to true.

The usage patterns where disabling the `aft_enable_vpc` parameter (setting it to false) might help reduce operating costs typically involve scenarios where:

1. Public networking is sufficient: If your organization doesn't require the enhanced security and network isolation provided by a VPC for AFT operations, and public internet connectivity is acceptable.

2. Low-volume account provisioning: If you're not frequently provisioning or updating accounts, the cost savings from not running constant NAT Gateways and VPC endpoints might be significant.

3. Non-sensitive environments: For non-production or test environments where the additional security layer of a VPC is not critical.

4. Simple AFT implementations: If your AFT setup doesn't require complex networking or you're not using features that benefit from private networking.

5. Cost-sensitive deployments: For organizations prioritizing cost optimization over network isolation, especially if they have other security measures in place.

6. Temporary or short-term use: If you're using AFT for a limited time or for specific projects rather than as a long-term, always-on solution.

7. Environments with existing network infrastructure: If you already have a network setup that can securely handle AFT operations without needing a dedicated VPC.

8. Small-scale deployments: For smaller organizations or projects where the scale doesn't justify the additional networking costs.

9. Regions with high NAT Gateway costs: In AWS regions where NAT Gateway pricing is particularly high, disabling the VPC might lead to more significant cost savings.

10. Simplified management: If you prefer a simpler networking setup and are willing to trade off some network isolation for easier management and potentially lower costs.

It's important to note that while disabling `aft_enable_vpc` can reduce costs, it also means giving up the security and isolation benefits of a VPC. Organizations should carefully consider their security requirements, compliance needs, and overall AWS architecture before making this decision. In many cases, especially for larger enterprises or those dealing with sensitive data, the security benefits of using a VPC might outweigh the potential cost savings.

Running AFT without a dedicated VPC with NAT

AFT can still function without a VPC:

1. AWS API Access:
   - With VPC: AFT uses VPC endpoints for private access to AWS services.
   - Without VPC: AFT communicates with AWS services over the public internet using HTTPS.

2. Lambda Functions:
   - With VPC: Lambda functions run inside the VPC.
   - Without VPC: Lambda functions run in the AWS-managed infrastructure, connecting to services via public endpoints.

3. Step Functions:
   - These are serverless and don't require a VPC to operate.

4. S3 Buckets:
   - These are globally accessible and don't require a VPC.

5. DynamoDB Tables:
   - These are also accessible without a VPC.

6. SNS Topics and SQS Queues:
   - These services can be used without a VPC.

7. CodePipeline and CodeBuild:

- These can operate without a VPC, using public endpoints for source and artifact storage.

The key difference is that without a VPC, all communication between AFT components and AWS services happens over the public internet rather than through private network connections. AWS ensures security through:

1. HTTPS encryption for all API calls
2. IAM roles and policies for access control
3. AWS signature v4 for request authentication

So, AFT can indeed work without a VPC. The trade-off is between the enhanced network isolation and security of a VPC versus the simplicity and potential cost savings of operating without one. The choice depends on your specific security requirements and risk tolerance.

Tailoring AFT-global-customizations

There's nothing to modify here at this point, but after the basic AFT installation is done, you will uncomment and modify api_helpers/pre-api-helpers.sh which has been prepopulated with commented-out bash commands to support OpenSecOps's open-source utilities

1. https://github.com/OpenSecOps-Org/AFT-SSO-account-configuration and
2. https://github.com/OpenSecOps-Org/AFT-DNS-subdomain-OpenSecOpsion.

Tailoring AFT-account-customizations

The only thing you need to do here is update the email addresses in `sandbox-customizations/terraform/budget.tf`.

Tailoring AFT-account-provisioning-customizations

There is nothing to modify here.

Tailoring AFT-account-request

This is where you will do most of your work with AFT as this repo contains the module definitions for your accounts. These files are the ones to modify:

- `terraform/OU-AFT.tf-DISABLED`
- `terraform/OU-Infra.tf-DISABLED`
- `terraform/OU-PolicyStaging.tf-DISABLED`
- `terraform/OU-Sandbox.tf-DISABLED`
- `terraform/OU-Suspended.tf-DISABLED`
- `terraform/OU-Workloads.tf-DISABLED`

OU-AFT contains the definition for the already created AFT-Management in the AFT OU. Update "`yourcompany.com`" to your actual email domain, change "`yourcompany`" in the group access definition `custom_fields` section to whatever group prefix you'll be using later to set up the SSO Groups, and also update `change_requested_by` to your own name.

NB: In `OU-PolicyStaging.tf` and `OU-Workloads.tf` you must also specify the actual OU IDs with the parentheses of the value of `ManagedOrganizationalUnit`. The values to replace are of the form "`PolicyStagingInfra (ou-3f6y-7djh234xy)`" and the appropriate values can be found in Control Tower.

Deploying AFT

The next step is to deploy AFT using Terraform CLI commands.

As we mentioned at the top of the AFT installation section, the full instructions are here, on Hashicorp's site, in the form of a tutorial:

https://developer.hashicorp.com/terraform/tutorials/aws/aws-control-tower-aft

(For your reference, AWS' instructions are here:

https://docs.aws.amazon.com/controltower/latest/userguide/aft-getting-started.html)

We have already created:

1. An OU called AFT,
2. An account, AFT-Management, in that OU.
3. Forked, cloned and tailored the Git repositories AFT uses, and,
4. Stored local copies of these in the AFT folder on our local computer.

You can thus start at the section "Apply configuration", as we have already done all the preparatory steps. In a terminal, `cd` to `AFT/AFT-install`.

Authenticate

It's up to you how you authenticate, but the easiest way at this point is to simply copy and paste the AWSAdministratorAccess keys for the Org account from the SSO login screen.

Validate

You can then do "`terraform init`", "`terraform validate`", and "`terraform plan`" to verify that everything works.

Apply

When you are ready to deploy, type "`terraform apply`". The process will take about 30 minutes.

NB: If you're running without a VPC, "`terraform apply`" might fail the first time. Just try again; it should succeed the second time.

Post-Installation

Elevate your Privileges in AFT-Management

At this point, you need to:

1. Log into the Org account using SSO,
2. Go to IAM Identity Center (formerly known as AWS SSO),
3. Then to "AWS accounts",
4. Open the "AFT" widget,
5. Select "AFT-Management",
6. Click "Assign users or groups",
7. Click on "Users",
8. Then select your personal user,
9. Click on "Next",

10. Select “AWSAdministratorAccess”,
11. Click on “Next”, then “Submit”,
12. Wait for the update to complete.

The reason for this is that we’re not yet using the OpenSecOps Foundation permission sets and groups, and Control Tower will by default not assign you `AWSAdministratorAccess` to new accounts but rather `AWSOrganizationsFullAccess`, which isn’t sufficient for you to complete the installation.

Enable the CodeStar Connection

1. Log in to the AFT-Management account using your SSO User.
2. Search for “codesuite”.
3. Go to Settings, then to Connections.
4. Click on the pending connection, which should have a name similar to “`ct-aft-github-connection`”.
5. Follow the Hashicorp instructions.

Grant AFT access to Service Catalog portfolio

Similarly, follow the instructions to configure Service Catalog for AFT.

1. You will find that the tab no longer is called “Groups, Roles and Users” but simply “Access”.
2. Click on Grant Access.
3. Click on Roles.
4. Search for “AWSAFTExecution” and grant access.

Rerun the account provisioning pipeline

Next, follow the instructions to rerun the account provisioning pipeline.

- If restarting the pipeline fails in the Source stage, wait and retry. The connection to your external provider may take up to 24 hours to take.
- If you accidentally didn’t install the GitHub app, create a temporary connection named “temp” or anything you like, and use it to install the app in the correct organisation on GitHub. Then delete “temp”.

Verifying operations

We’re now ready to test the whole GitOps chain.

Update the AFT-Management account

1. In a terminal, go to your `AFT` folder and then on to `AFT-account-request`.
2. In the `terraform` folder, rename “`OU-AFT.tf-DISABLED`” to “`OU-AFT.tf`”.
3. Update the following fields:
    a. `AccountEmail` (to fit your company email domain and account mailing list name)
    b. `SSOUserEmail` (to match the system Control Tower Admin user)
    c. `soar:team:email` (to fit your company email domain and team mailing list name)
    d. `change_requested_by` (enter your name here)
    e. `change_reason` (if you wish)
4. Add and commit these changes, then push to the external provider (such as GitHub).
5. Use your SSO user to log into the AFT-Administration account. Check that:
    a. In CodePipeline, check that `ct-aft-account-request` completed successfully.

   b. In Step Functions, check that none of the state machines have recent errors. If any of them are running, let them run to completion.
6. Use your SSO user to log into the Org account. Go to AWS Organizations and verify that the "`AFT-Management`" account has the tags specified in `OU-AFT.tf`.

Run Account Customizations Manually

Since we created the AFT-Management manually in Service Catalog, it has no dedicated customisation pipeline. To verify that the system can create one, trigger the account customisation pipeline manually to automatically create one.

1. Use your SSO user to log into the AFT-Administration account.
2. Go to Step Functions, then onto State machines.
3. Click on "`aft-invoke-customizations`".
4. Click on "Start execution" and paste the following into the input field:

```
{ "include": [{"type": "all"}]}
```

5. Click on "Start execution".
6. Watch the state machine. As it executes, check the aft-account-provisioning-framework state machine: it should build the pipeline for AFT-Management.
7. After everything completes successfully, log into the Org account and check that the tags of AFT-Management have been updated.
8. We have now verified that AFT is fully functional for creating and updating accounts and their associated pipelines.

Provision the Security-Adm and Networking accounts

The next step is to create the remaining two accounts needed for installing OpenSecOps Foundation, `Security-Adm` and `Networking`.

To do so:

1. In a terminal, go to your `AFT` folder and then on to `AFT-account-request`.
2. In the `terraform` folder, rename "`OU-Infra.tf-DISABLED`" to "`OU-Infra.tf`".
3. Update all relevant fields.
4. Add and commit your changes, then push to your external Git provider (such as GitHub).
5. Log into the AFT-Administration account and check that `ct-aft-account-request` completes successfully.
6. This will trigger activity in Service Catalog in your Org account, so log into Org, go to Service Catalog and:
   a. Click on Provisioned Products
   b. Set "Access filter" to Account
   c. Watch the accounts being created, one after the other.
7. When the first account has been created, check that it is being enrolled in AWS Control Tower and that it appears in AWS Organizations.
8. When it's completely enrolled, log into AFT-Management and see that its pipeline is being created and executed.
9. When the account pipeline has run to completion, log back into Org and check that the tags have been set properly in AWS Organizations.
10. When both accounts are done, we have verified that AFT is fully operational in all respects.

Elevate your Privileges

As a final step, elevate your privileges in both these new accounts in the same way as you did for `AFT-Management`.

Extending AFT

The next step is to extend the functionality of AFT so that access and subdomain OpenSecOpsion will be set up automatically from now on. For this, we need to use OpenSecOps Install before we install the two open source SAM projects required.

OpenSecOps Install is the tool clients will use after you're finished the first install to update and maintain the OpenSecOps Foundation installation. We will use it to install the first repos defining:

- OpenSecOps Foundation boundary permission policies,
- OpenSecOps Foundation SCPs, and,
- the OpenSecOps Foundation core SSO configuration.

Then, we'll manually install two open-source SAM projects:

- https://github.com/OpenSecOps-Org/AFT-SSO-account-configuration
- https://github.com/OpenSecOps-Org/AFT-DNS-subdomain-OpenSecOpsion

When they have been fully installed, AFT will process the following `custom_field` data during account customisation:

```
custom_fields = {
  "sso_groups" = jsonencode({
    "yourcompany-security-administration" = ["SecurityAdministratorAccess", "ReadOnlyWideAccess"]
    "yourcompany-network-administration"  = ["NetworkAdministratorAccess", "ReadOnlyWideAccess"]
    "yourcompany-platform-team"           = ["DeveloperAccess", "ReadOnlyWideAccess"]
    "yourcompany-auditors"                = "ReadOnlyWideAccess"
  })

  "sso_users" = jsonencode({
    "john.doe@yourcompany.com" = "DeveloperAccess"
  })

  subdomain_delegations = "jd.yourcompany.cloud, john.yourcompany.cloud"
}
```

That is, access for SSO groups (`sso_groups`) and individual users (`sso_users`), as well as any subdomain OpenSecOpsions (`subdomain_OpenSecOpsions`) we wish to assign to the account.

Setting up CLI access to AWS

OpenSecOps Install is a command-line tool. It requires CLI access to your accounts. For this, you can use something like the following in your `~/.aws/config` file:

```
[default]
region = eu-north-1
sso_start_url = https://yourcompany.awsapps.com/start
sso_region = eu-north-1
```

```
sso_account_id = 111111111111
sso_role_name = AWSAdministratorAccess

[profile Org]
region = eu-north-1
sso_start_url = https://yourcompany.awsapps.com/start
sso_region = eu-north-1
sso_account_id = 111111111111
sso_role_name = AWSAdministratorAccess

[profile Log-Archive]
region = eu-north-1
sso_start_url = https://yourcompany.awsapps.com/start
sso_account_id = 222222222222
sso_role_name = AWSAdministratorAccess
sso_region = eu-north-1

[profile Audit]
region = eu-north-1
sso_start_url = https://yourcompany.awsapps.com/start
sso_account_id = 333333333333
sso_role_name = AWSAdministratorAccess
sso_region = eu-north-1

[profile AFT-Management]
region = eu-north-1
sso_start_url = https://yourcompany.awsapps.com/start
sso_region = eu-north-1
sso_account_id = 444444444444
sso_role_name = AWSAdministratorAccess

[profile Security-Adm]
region = eu-north-1
sso_start_url = https://yourcompany.awsapps.com/start
sso_region = eu-north-1
sso_account_id = 555555555555
sso_role_name = AWSAdministratorAccess
```

Update the `region`, the `sso_start_url`s, the `sso_account_id`s, and, if you wish, the `profile` names. Note that the `default` and the `Org` profile account details are the same; this is intentional.

Setting up OpenSecOps Install

Clone the repo

1. In your terminal, navigate to your `AWS` folder, then `cd` to `OpenSecOps-CLIENT-FOLDER`.
2. `git clone git@github.com:OpenSecOps-Org/Installer.git`

Copy the configuration directory

1. `cd Installer`
2. `cp -rf apps.example apps`

Set up source control for the configuration

You will notice that the new `apps` folder is excluded from source control. As this folder will contain the entire configuration for OpenSecOps Foundation, it's *strongly*

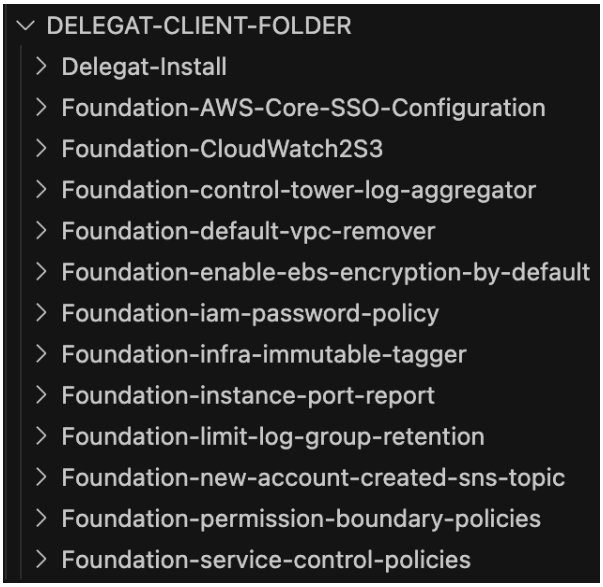
recommended that you create a Git repository for the `apps` folder in a central location, so that the configuration can be shared between cloud administrators.

You clearly know how to set this up, so please do it now before you start tailoring it to the client's installation.

Download the Foundation repos

1. `cd` to `Installer`.
2. `./init Foundation`

Your OpenSecOps-CLIENT-FOLDER should now look like this:



Configure Account IDs and SSO profile names

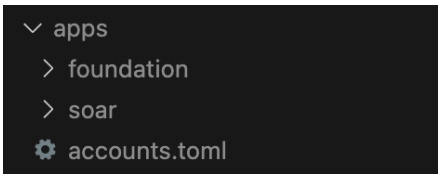
The first thing you need to modify is `Installer/apps/accounts.toml`. It looks like this:

```
# Account IDs and SSO profile names.
# NB: Enter the SSO profile name, not the account name.
# Below, they happen to be the same, which is quite OK.
# The profiles must have SystemAdministrator privileges.
#

admin-account.id               = "111111111111"
admin-account.profile          = "Org"

security-account.id            = "222222222222"
security-account.profile       = "Security-Adm"

log-archive-account.id         = "333333333333"
log-archive-account.profile    = "Log-Archive"

aft-management-account.id      = "444444444444"
aft-management-account.profile = "AFT-Management"

audit-account.id               = "555555555555"
audit-account.profile          = "Audit"
```

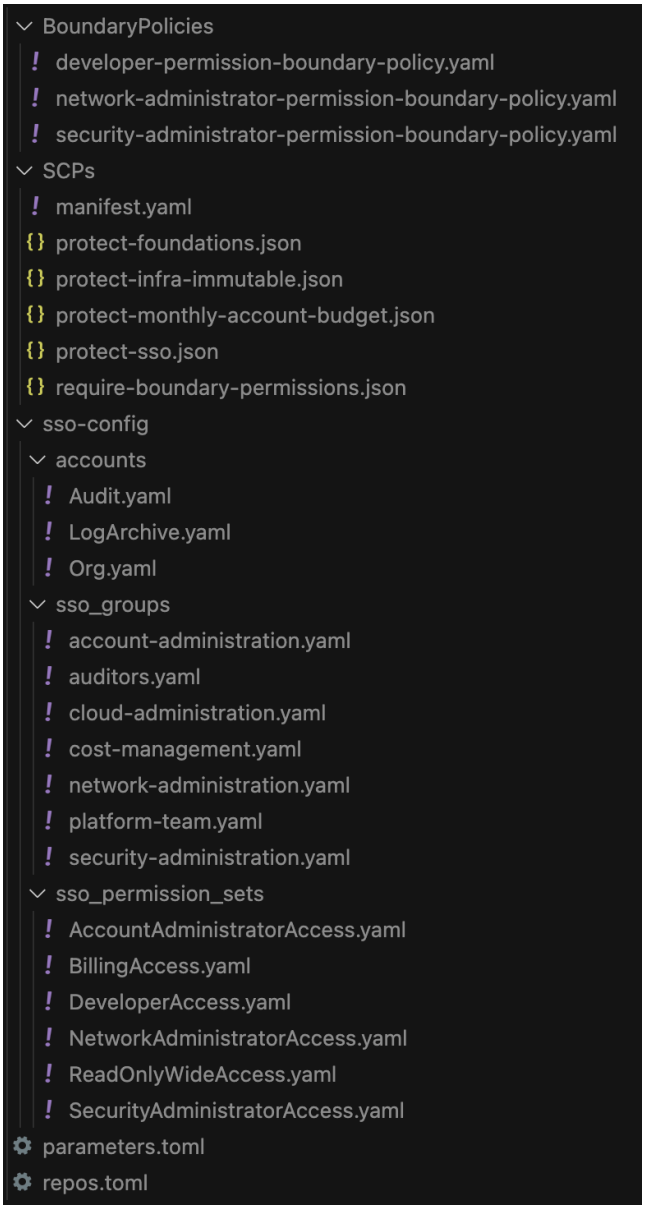
If you've named your profiles as per the example in "Setting up CLI access to AWS", you only need to update the account ids.

Configure Foundation parameters

OpenSecOps Foundation, like OpenSecOps SOAR, is configured through text files. They each have their configuration folders, next to the account configuration file:



The foundation folder is structured like this:


BoundaryPolicies
developer-permission-boundary-policy.yaml
network-administrator-permission-boundary-policy.yaml
security-administrator-permission-boundary-policy.yaml
SCPs
manifest.yaml
protect-foundations.json
protect-infra-immutable.json
protect-monthly-account-budget.json
protect-sso.json
require-boundary-permissions.json
sso-config
accounts
Audit.yaml
LogArchive.yaml
Org.yaml
sso_groups
account-administration.yaml
auditors.yaml
cloud-administration.yaml
cost-management.yaml
network-administration.yaml
platform-team.yaml
security-administration.yaml
sso_permission_sets
AccountAdministratorAccess.yaml
BillingAccess.yaml
DeveloperAccess.yaml
NetworkAdministratorAccess.yaml
ReadOnlyWideAccess.yaml
SecurityAdministratorAccess.yaml
parameters.toml
repos.toml

parameters.toml

The first file we will configure is `parameters.toml`. First modify the first section:

```
# Parameters for all Delegat Foundation repos

# ------------------------------------------------------------
#
# Global parameters (used across repos)
#
# ------------------------------------------------------------

# AWS Organizations
org-id                      = 'o-xxxxxxxxxx'
root-ou                     = 'r-xxxx'

# Regions
main-region                 = 'eu-xxxxx-1'
other-regions               = ['us-xxxx-1']

# Either AWSControlTowerExecution (under Control Tower)
# or OrganizationAccountAccessRole (under plain AWS Organizations)
cross-account-role          = "AWSControlTowerExecution"

# The email domain to use for notifications
email-domain                = 'example.com'

# Group prefix for SSO
sso-group-prefix            = "acme-inc-"
```

You will find `org-id` and `root-ou` in AWS Organizations in your Org account.

The `main-region` should be set to the region where you installed AWS Control Tower.

Set `other-regions` to the list of additional regions you have activated in AWS Control tower. If your main region is `eu-north-1` and your additional regions are `eu-central-1` and `us-east-1`, set this value to `["eu-central-1", "us-east-1"]`.

`email-domain` is the email domain from which the weekly port report will send its emails. It must be a domain to which SES has rights to send. You may want to register a domain for this purpose using Route 53 and authorise it appropriately. We also recommend you to enable DKIM.

Finally, `sso-group-prefix` is the prefix that will be used for all SSO Groups OpenSecOps Foundation sets up. It must end with a hyphen.

The only other thing you need to modify at this point is this section:

```
# ------------------------------------------------------------
#
# Foundation-instance-port-report
#
# ------------------------------------------------------------

[Foundation-instance-port-report.SAM]

EmailRecipients  = "accounts@example.com"   # Configure this
EmailCC          = "soc@example.com"        # Configure this
EmailBCC         = ""                       # Configure this

CrossAccountRole = '{cross-account-role}'
Regions          = "{all-regions}"
EmailSender      = "no-reply@{email-domain}"
EmailReturnPath  = "no-reply@{email-domain}"
```

EmailRecipients is the comma-separated list of recipients of the weekly instance port report. EmailCC, which likewise can be comma-separated, will get a CC copy. EmailBCC, another comma-separated list, will receive blind carbon copies.

Configure Foundation policies and core SSO

At this point you need to configure the security policies and the core SSO setup for the installation. There are three folders you need to consider: BoundaryPolicies, SCPs, and sso-config.

BoundaryPolicies folder

This folder contains three policies:

1. `developer-permission-boundary-policy.yaml`
2. `network-administrator-permission-boundary-policy.yaml`
3. `security-administrator-permission-boundary-policy.yaml`

The only one you need to consider at this point is the developer one, the one for Builders. This is the boundary policy that developers need to attach to all roles they create. The permissions are further restricted by SCPs. The permissions here restrict roles that the developers *create*, not the permissions Builders *have themselves*.

Therefore, this file should be kept in sync with `sso-config/sso_permission_sets/DeveloperAccess.yaml`, where Builder permissions are defined.

The only section you probably will want to modify at this point is this one, in both `DeveloperAccess.yaml` and `developer-permission-boundary-policy.yaml`:

```
  -
    Sid: AllowLocalServices
    Effect: Allow
    Resource: "*"
    Condition:
        StringEquals:
            aws:RequestedRegion:
              - eu-north-1
              - eu-west-2
```

Under `aws:RequestedRegion`, specify the regions in which Builders/Developers are allowed to work. In the example above, this is Stockholm and Ireland. Developers will not be able to access any other regions.

If you decide to change `NetworkAdministratorAccess` and/or `SecurityAdministratorAccess`, always do this in tandem with changing the corresponding boundary permissions, making sure the boundary permissions are equivalent or, ideally, more restrictive than the person-based SSO Access Permission Sets.

SCPs folder

There is no need to update anything here, unless you want to wait with enforcing the use of boundary permissions. If you do, change this:

```
- name: require-boundary-permissions
  description: |
    Requires builder principals to use an appropriate boundary policy for creating IAM Roles.
    Protects boundary permissions from being tampered with.
  resource_file: require-boundary-permissions.json
  deployment_targets:
    organizational_units:
      - Root
```

To this:

```
- name: require-boundary-permissions
  description: |
    Requires builder principals to use an appropriate boundary policy for creating IAM Roles.
    Protects boundary permissions from being tampered with.
  resource_file: require-boundary-permissions.json
  deployment_targets:
    organizational_units: []
    #  - Root
```

When, at a later date, you want to enable boundary permissions, revert the change and redeploy.

sso-config folder

This folder contains three subordinate folders:

- `accounts` – this folder contains permission assignments to the accounts not managed by AFT: `Org`, `Log Archive`, and `Audit`. There's nothing to change here.

- `sso_groups` – contains the SSO Groups OpenSecOps Foundation will create. You can add more if you like. There's nothing to change here. Note that the SSO Group names will be prefixed by the `sso-group-prefix` you specified in `parameters.toml`. If the `sso_group_name` is "`auditors`" and the prefix is "`acme-inc-`", the effective SSO Group name will be "`acme-inc-auditors`".

- `sso_permission_sets` – these are the SSO Permission Sets that OpenSecOps Foundation creates and maintains in IAM Identity Center (formerly known as AWS SSO). There's probably no need to change any of them, although you certainly can do so, apart from setting the allowed developer regions. See the discussion above about keeping

`DeveloperAccess.yaml` and `developer-permission-boundary-policy.yaml` in sync.

Create the AWSControlTowerExecution role in Org

In order for OpenSecOps Foundation to work in the Org account, we need to create `AWSControlTowerExecution`, which is present in all other AWS Control Tower accounts. This is in order so that it can be assumed from its own account, just like for any other account in your Organization. We are thus creating parity in all accounts.

1. Log into the Org account
2. Go to IAM
3. Click on "Roles", then on "Create role"
4. Select "AWS Account"
5. Check that "This account" is selected. This is important. Only principals in the Org account should be able to assume the new role.
6. Click on "Next"
7. Find and select "AdministratorAccess".
8. Click on "Next"
9. Name the role "AWSControlTowerExecution"

Enable Service Managed Stack Sets

1. Log in to the Org account
2. Go to CloudFormation, then click on StackSets, then on Enable trusted access.
3. Go to AWS Organizations, then to Services and verify that trusted access has been enabled.

Deploy Foundation-permission-boundary-policies

1. cd to Foundation-permission-boundary-policies
2. aws sso login
3. First do a dry run, so you can see what Installer is going to do:
   ```
   ./deploy --dry-run
   ```
4. If you're satisfied, run the command without the dry run arg:
   ```
   ./deploy
   ```
5. Wait for the deployment to finish
6. Verify that the Stacks and StackSets are in place and that they have defined their boundary policies.

Deploy Foundation-service-control-policies

1. cd to Foundation-service-control-policies
2. Do a dry run using
   ```
   ./deploy --dry-run
   ```
3. Then deploy for real:
   ```
   ./deploy
   ```
4. Go to AWS Organizations, then to Policies, then to Service control policies and verify that the SCPs are in place

Deploy Foundation-resource-control-policies

5. In AWS Organizations Services, enable Resource control policies.
6. cd to Foundation-resource-control-policies
7. Do a dry run using
   `./deploy --dry-run`
8. Then deploy for real:
   `./deploy`
9. Go to AWS Organizations, then to Policies, then to Resource control policies and verify that the RCPs are in place

Deploy Foundation-AWS-Core-SSO-Configuration

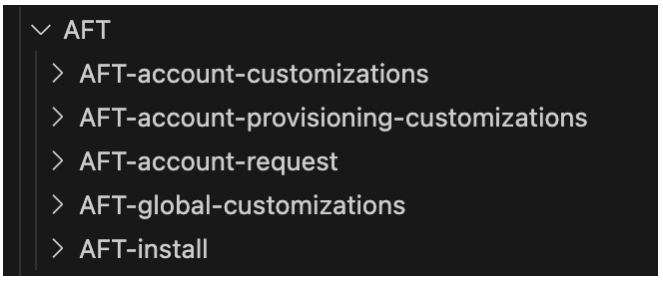
Proceed similarly with Foundation-AWS-Core-SSO-Configuration.

Deploy AFT-SSO-account-configuration

We now have all the SSO resources and configuration in place. This means that we're now ready to deploy the open-source SAM project AFT-SSO-account-configuration which uses the AFT `custom_data` to set up permissions for SSO groups and SSO users for accounts managed by AFT.

This repo is not managed by Installer, so we must install it manually. This is a very straightforward process.

1. In a terminal, `cd` to your AFT folder. It should have the following contents:



2. `git clone git@github.com:OpenSecOps-Org/AFT-SSO-account-configuration.git`
3. `cd AFT-SSO-account-configuration`
4. `sam build`
5. `sam deploy --guided`
6. Stack name: INFRA-AFT-SSO-account-configuration
7. AWS Region: your main region where AWS Control Tower lives
8. Parameter OrgId: Your AWS Organization ID
9. Parameter CloudAdministrationGroupName: the name of the cloud-administrators SSO Group with your prefix, e.g. "`acme-inc-cloud-administration`".
10. Parameter CloudAdministrationGroupPermissionSets: `AWSAdministratorAccess,ReadOnlyWideAccess`

When deployment has finished, simply proceed to the next installation. We'll update the `AFT-global-customizations` repo for both installations at the same time.

Deploy AFT-DNS-subdomain-OpenSecOpsion

1. `cd` to your AFT folder again.
2. `git clone git@github.com:OpenSecOps-Org/AFT-DNS-subdomain-OpenSecOpsion.git`
3. `cd AFT-DNS-subdomain-OpenSecOpsion`
4. `sam build`
5. `sam deploy --guided`
6. Stack name: INFRA-AFT-DNS-subdomain-OpenSecOpsion
7. AWS Region: your main region where AWS Control Tower lives
8. Parameter OrgId: Your AWS Organization ID
9. Parameter NetworkingAccountId: the 12-digit account ID of the Networking account.

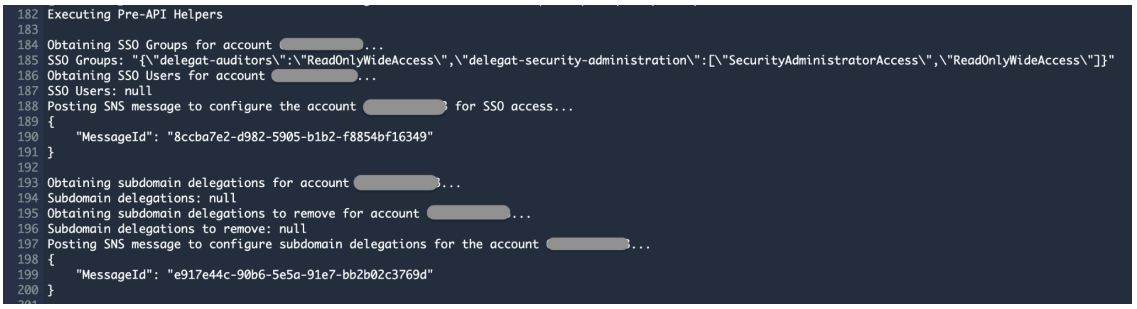
Activating the AFT extensions

To make things easier, this step has already been prepared for you.

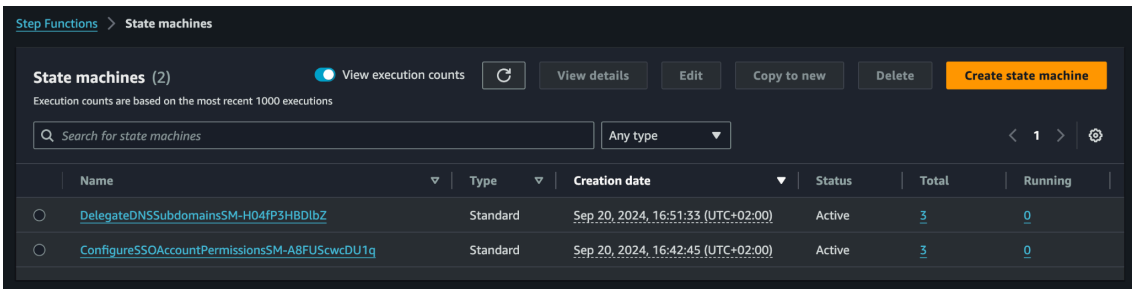
1. cd to `AFT-global-customizations`
2. Open `api_helpers/pre-api-helpers.sh`
3. Uncomment the lines after the first “echo”.
4. Replace “111111111111” with your Org account number.
5. Add, commit, and push.
6. Log on to the AFT-Management account.
7. Go to Step Functions.
8. Click on the “`aft-invoke-customizations`” state machine. (You have run it once before when setting up AFT.)
9. Start an execution with the input:

```
{ "include": [{"type": "all"}]}
```

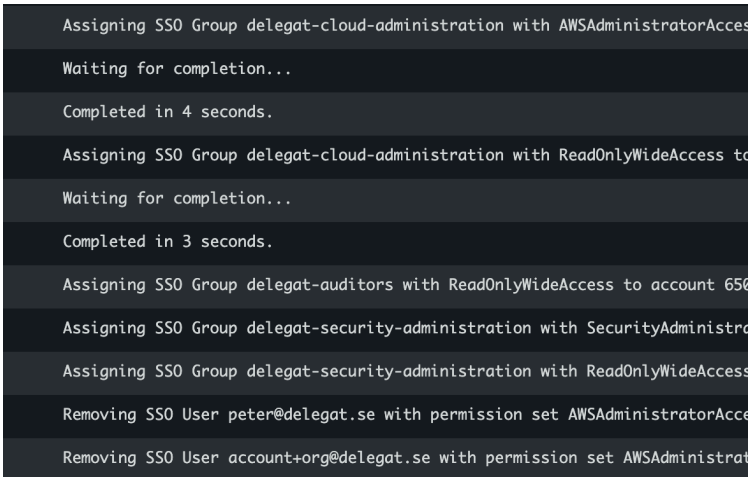
10. Click on “Start execution”.
11. Watch the state machine.
12. Also check the state machine “xxxxxxxx” which should have one execution for each account in your system handled by AFT (in the default clean install situation, there should be three of them).
13. Also check CodePipeline, where there should be the same number of pipelines running. You can check the AFT-Global-Customizations stage logs; you should see something like:



14. Log into the Org account and go to Step Functions. You should see something like the following:



15. Check that both have successful executions.
16. For ConfigureSSOAccountPermissions state machine, follow the links to its CloudWatch log group and verify that the logs contain something like the following:

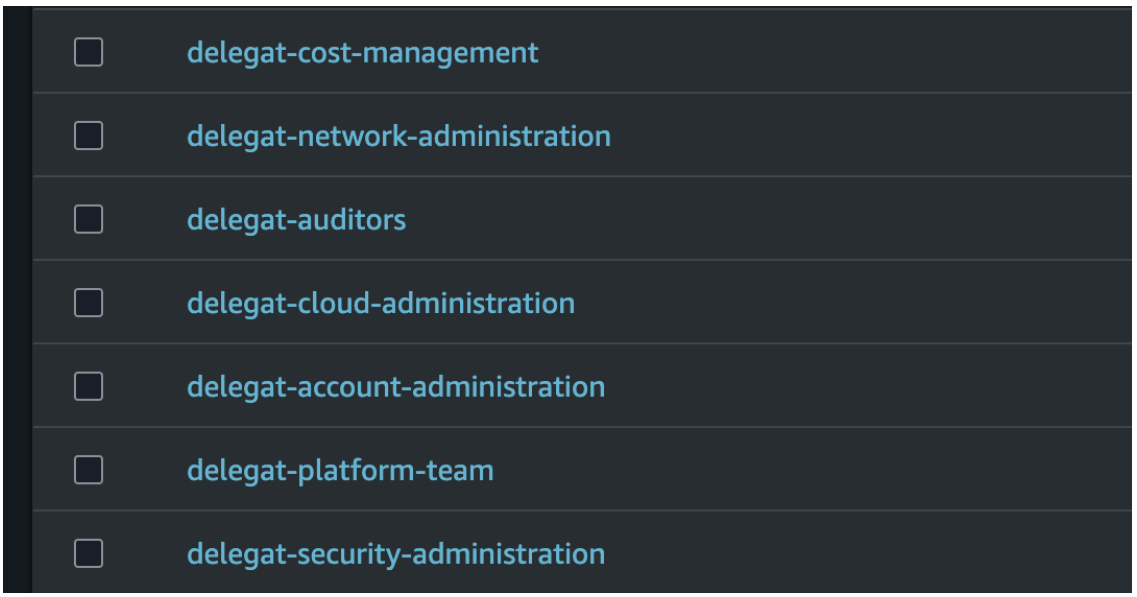


    If permissions are successfully assigned and removed, the AFT extensions are fully operational.
17. There is no need to protect the settings as indicated in the open-source README, as the SCPs you already have installed include such protection.

Adjust SSO Group memberships

Now that AFT can handle the assignment of all permissions for groups and users, you can remove your old group assignments:

1. Log on to the Org account using your SSO user.
2. Go to IAM Identity Center (formerly known as AWS SSO), then on to your SSO User.
3. Add yourself to all the new groups beginning with your `sso-group-prefix`.
4. Remove yourself from all other groups.
5. The final result should look something like this, given that your `sso-group-prefix` is "`OpenSecOps-`":



NB: this is something that must be done with every SSO User in the system, in order to ensure secure operations. If the AWS system already existed and had SSO Users already, you will need to do this with all of them. However, you may wish to do so gradually, to ensure their access isn't broken during the transition period.

The only exception to this is the break-glass SSO user, "`accounts+org@yourcompany.com`", which should be left as-is.

Thus, essentially, we're replacing all existing SSO Groups with OpenSecOps Foundation's new ones. This is essential in order to protect the system against escalation of privileges. When you're done, the old groups shouldn't be used by *any* user except the `accounts+org` one.

For full information about who should go in which groups, see the OpenSecOps Foundation TDS and SOP manuals.

6. Sign out and in again to see the new permissions on the AWS access portal.

Activations & OpenSecOpsions

AWS Config

1. In the Org account, enable AWS Config in your main region. Remove the filter: in this region, you want to record IAM global events.
2. Enable AWS Config in your other enabled regions. Do not remove the IAM global filter in these regions.

IAM Identity Center

1. Register the AFT-Management account as the OpenSecOpsed administrator for IAM Identity Center (formerly known as AWS SSO).

GuardDuty

1. Enable GuardDuty in all your activated regions.

2. OpenSecOpse administration of GuardDuty to the Security-Adm account in all your activated regions.
3. Log in to Security-Adm, enable and set up auto-enable in all your activated regions.

IAM Access Analyzer

1. In Org, OpenSecOpse administration of IAM Access Analyzer to the Security-Adm account.
2. In Security-Adm, set up an organisation-wide IAM Access Analyzer for external access in all your regions.
3. Set up an organisation-wide IAM Access Analyzer for unused access in your main region only.

Amazon Detective

1. In Org, OpenSecOpse Amazon Detective in all your regions to Security-Adm (the GUI will suggest this account automatically).
2. In Security-Adm, configure Detective in all your selected regions.

Amazon Inspector

1. In the Org account, OpenSecOpse administration of Amazon Inspector to the Security-Adm account in all your chosen regions.

2. In the Security-Adm account, configure Amazon Inspector in each chosen region. Note that you must activate/invite the individual existing member accounts in each region as well as enable automatic activation of new accounts.

AWS Security Hub

1. In the Org account, OpenSecOpse administration to the Security-Adm account as you enable Security Hub in all your enabled regions.

2. In the Security-Adm account, set up central configuration and consolidated findings in all your enabled regions.

3. Set up two policies, one for PROD and one for DEV accounts. Make sure that auto-enabling of new controls is *not* enabled. Also make sure that you select exactly the controls you need, one by one. The PROD policy will include things like multi-zone deployment, inclusion in backup plans, and deletion protection of resources. The DEV policy should not require these things, to ease development. Deleting a KMS key is a no-no in PROD, but should be allowed in DEV, for instance.
    a. Assign the PROD policy to the org root
    b. Assign the DEV policy to IndividualBusinessUsers, Sandbox, and SDLC OUs.

4. Suppress all findings in all regions to let the system start over with the new settings. Relevant findings will be regenerated within 24 hours. It’s a good idea to wait 24 hours to verify your control setup.

Installing the rest of OpenSecOps Foundation

Deployment

In a terminal, `cd` to your Installer folder.

Log into AWS using "`aws sso login`".

Before starting the installation, it's a good idea to check what the installation will do using "`./deploy-all --dry-run`". This will refresh the repositories, build the applications, upload the artifacts and inform you of what the installer will do.

If you encounter any problems, you might want to run the command again, but with the added switch "`--verbose`", thus: "`./deploy-all --dry-run --verbose`".

When you're ready, give the command "`./deploy-all`" and wait for completion.

Post-Deployment

The first three of these post-deployment actions are strictly speaking not necessary as they are run automatically on schedule once a day. You can wait 24 hours or, if you prefer, invoke them manually.

Foundation-infra-immutable-tagger

1. In the Org account, go to Lambda and search for "`INFRA-infra-immutable-tagger-TriggerFunction-`"
2. Start a Test run with the JSON input data "`{"AccountId": "ALL"}`"
3. On the Step Functions page, verify that `InfraImmutableTaggerSM` has run successfully.

Foundation-default-vpc-remover

1. In the Org account, go to Lambda and search for "`INFRA-default-vpc-remover-TriggerFunction-`"
2. Start a Test run with the JSON input data "`{"AccountId": "ALL"}`"
3. On the Step Functions page, verify that `RemoveDefaultVpcsFromAccount` has run successfully.

Foundation-limit-log-group-retention

1. In the Org account, go to Step Functions and click on the `LimitLogGroupRetention` state machine.
2. Start an execution. The JSON input is not used.
3. Verify that execution completes successfully.

Foundation-control-tower-log-aggregator

If you have old Control Tower log files, you might want to process them so everything follows the same format. The log processor will only process log files from the day before, so if the system has existed for some time, or if installation has taken more than a day, you might want to check out the instructions at https://github.com/OpenSecOps-Org/Foundation-control-tower-log-aggregator, section "Processing Old Main Logs". Note that the date range includes the `end_date`.

When done, verify that the S3 bucket "`all-aggregated-logs-111111111111-`*`xx-yyyy-9`*" no longer is empty.

You may also want to trigger the state machine "`CombineLogFilesSM`" to process all files from the previous day, not just Control Tower logs. This includes the application logs collected from member accounts by Foundation-CloudWatch2S3.

Increase Lambda Quota in Log Archive

If the system is new, or if the Log Archive account hasn't had much activity, you will need to request a quota increase for concurrent Lambda executions in that account, or log processing will take an inordinate amount of time each day. Use Service Quotas. It may be set to 10; request a new value of 1000.

Deploy the remaining AFT accounts

In the following, please check that you don't exceed the account number quota. For a newly created system, this is 10 accounts. This will allow you to create the PolicyStaging accounts, but if you also wish to create Sandbox accounts, you must raise the quota before you attempt to create the new accounts.

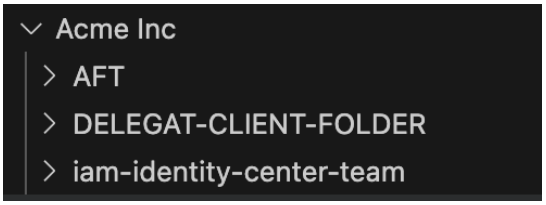
1. In a terminal, `cd` to your AFT folder, then `cd` to `AFT-account-request/terraform`.
2. Rename `OU-PolicyStaging.tf-DISABLED` to `OU-PolicyStaging.tf`.
3. Customise as necessary if you didn't do this already.
4. `git add`, `git commit`, and `git push`.
5. The pipeline will create the four accounts in order. This will take some time: up to 30 minutes per account. When done, access and permissions have been assigned as specified in the modules.

6. If you want to create personal sandbox accounts, tailor and rename `OU-Sandbox.tf-DISABLED` in the same manner. Note that `account_customizations_name` should be set to `"sandbox-customizations"` for this OU.
7. `git add`, `git commit`, and `git push`.
8. Wait for completion.

Install Temporary Elevated Access Management (TEAM)

Follow the instructions at https://aws.amazon.com/blogs/security/temporary-elevated-access-management-with-iam-identity-center/ to install Just-In-Time access management (TEAM).

Download the TEAM repo to your client folder, the one containing your AFT folder and your OpenSecOps-CLIENT-FOLDER. After the clone it should look like this:



`cd` into iam-identity-center-team and continue the installation from there.

Install TEAM in the `AFT-Management` account, not your Org account. Permissions have been set up to support deploying TEAM in `AFT-Management`.

This is the default `parameters.sh` file:

```
IDC_LOGIN_URL=https://d-90676dxxxx.awsapps.com/start
REGION=us-east-1
TEAM_ACCOUNT=123456789101
ORG_MASTER_PROFILE=org_master_profile
TEAM_ACCOUNT_PROFILE=team_account_profile
TEAM_ADMIN_GROUP="team_admin_group_name"
TEAM_AUDITOR_GROUP="team_auditor_group_name"
TAGS="project=iam-identity-center-team environment=prod"
CLOUDTRAIL_AUDIT_LOGS=arn:aws:cloudtrail:us-east-1:123456789101:eventdatastore/e646f20d-7959-4682-be84-6c5b8a37cf15
```

Comments:

- **IDC_LOGIN** is your AWS SSO login URL
- **REGION** is your main region, the one where Control Tower was installed
- **TEAM_ACCOUNT** is the ID of your AFT-Management account
- **ORG_MASTER_PROFILE** is the name of the Org SSO profile ("`Org`"), not the account name
- **TEAM_ACCOUNT_PROFLE** is the name of the AFT-Management SSO profile ("`AFT-Management`"), not the account name
- **TEAM_ADMIN_GROUP** can be set to "`TEAM-admins`"
- **TEAM_AUDITOR_GROUP** can be set to "`TEAM-auditors`"

SystemAdministrator Session Duration

If you raised the session time of the SystemAdministrator permission set, you should now reset it to 1 hour.

The SystemAdministrator permission set should not be used for day-to-day work; the installation you've just completed defines other roles and permissions that should be used instead.

Please refer to the OpenSecOps Foundation TDS for more information.

Updating and Recovering OpenSecOps Foundation

The procedure is described in the Installer README, at https://github.com/OpenSecOps-Org/Installer/blob/main/README.md.

Essentially, all you need to do is go to your Installer folder in a terminal and then:

```
1. git pull
2. ./deploy-all Foundation --dry-run
3. ./deploy-all Foundation
```

If you want to update a single component, go to its repo directory and do:

```
1. ./deploy --dry-run
2. ./deploy
```

It's generally prudent to do a dry run first to see what is going to change, but the dry run is of course completely optional.

NB: You can always delete all repositories, including the `Installer` one, provided you keep `Installer/apps` where all your configuration lives. You should already

have set up source control for the `apps` directory. There is no configuration in the individual repositories.

If you delete the component repo directories, all you need to do is "`./init Foundation`" and fresh copies will be downloaded from OpenSecOps's central repositories.

Technical Support & SLAs

We offer post-installation technical support and Service Level Agreements tailored to the client's needs and use case. For more information, please contact OpenSecOps AB's sales representatives.

Conclusion

This concludes the installation of OpenSecOps Foundation.