AWS Secure Environment Accelerator

Amazon Web Services

Amazon Web Services

None

Table of contents

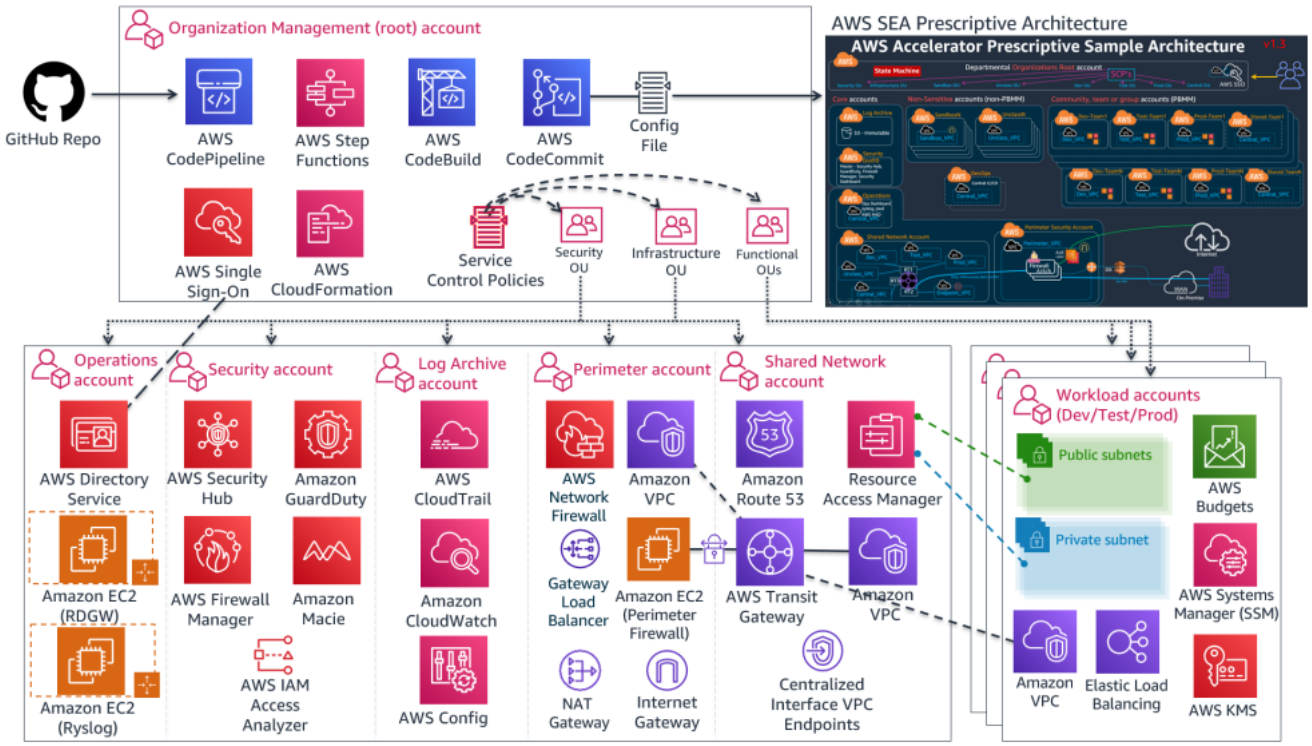
1. 1. AWS Secure Environment Accelerator

1.1 1.1. Overview

The AWS Accelerator is a tool designed to help deploy and operate secure multi-account, multi-region AWS environments on an ongoing basis. The power of the solution is the configuration file that drives the architecture deployed by the tool. This enables extensive flexibility and for the completely automated deployment of a customized architecture within AWS without changing a single line of code.

While flexible, the AWS Accelerator is delivered with a sample configuration file which deploys an opinionated and prescriptive architecture designed to help meet the security and operational requirements of many governments around the world. Tuning the parameters within the configuration file allows for the deployment of customized architectures and enables the solution to help meet the multitude of requirements of a broad range of governments and public sector organizations.

The installation of the provided prescriptive architecture is reasonably simple, deploying a customized architecture does require extensive understanding of the AWS platform. The sample deployment specifically helps customers meet NIST 800-53 and/or CCCS Medium Cloud Control Profile (formerly PBMM).



1.2 1.2. What specifically does the Accelerator deploy and manage?

A common misconception is that the AWS Secure Environment Accelerator only deploys security services, not true. The Accelerator is capable of deploying a complete end-to-end hybrid enterprise multi-region cloud environment.

Additionally, while the Accelerator is initially responsible for deploying a prescribed architecture, it more importantly allows for organizations to operate, evolve, and maintain their cloud architecture and security controls over time and as they grow, with minimal effort, often using native AWS tools. While the Accelerator helps with the deployment of technical security controls, it's important to understand that the Accelerator is only part of your security and compliance effort. We encourage customers to work with their AWS account team, AWS Professional Services or an AWS Partner to determine how to best meet the remainder of your compliance requirements.

The Accelerator is designed to enable customers to upgrade across Accelerator versions while maintaining a customer's specific configuration and customizations, and without the need for any coding expertise or for professional services. Customers have been able to seamlessly upgrade their AWS multi-account environment from the very first Accelerator beta release to the latest release (across more than 50 releases), gaining the benefits of bug fixes and enhancements while having the option to enable new features, without any loss of existing customization or functionality.

Specifically the accelerator deploys and manages the following functionality, both at initial accelerator deployment and as new accounts are created, added, or onboarded in a completely automated but customizable manner:

1.2.1 1.2.1. Creates AWS Account

- Core Accounts - as many or as few as your organization requires, using the naming you desire. These accounts are used to centralize core capabilities across the organization and provide `Control Panel` like capabilities across the environment. Common core accounts include:
- Shared Network
- Operations
- Perimeter
- Log Archive
- Security Tooling
- Workload Accounts - automated concurrent mass account creation or use AWS organizations to scale one account at a time. These accounts are used to host a customer's workloads and applications.
- Scalable to 1000's of AWS accounts
- Supports AWS Organizations nested OU's and importing existing AWS accounts
- Performs 'account warming' to establish initial limits, when required
- Automatically submits limit increases, when required (complies with initial limits until increased)
- Leverages AWS Control Tower

1.2.2 1.2.2. Creates Networking

- Transit Gateways and TGW route tables (incl. inter-region TGW peering)
- Centralized and/or Local (bespoke) VPC's
- Subnets, Route tables, NACLs, Security groups, NATGWs, IGWs, VGWs, CGWs
- **NEW** Outpost, Local Zone and Wavelength support
- VPC Endpoints (Gateway and Interface, Centralized or Local)
- Route 53 Private and Public Zones, Resolver Rules and Endpoints, VPC Endpoint Overloaded Zones
- All completely and individually customizable (per account, VPC, subnet, or OU)
- Layout and customize your VPCs, subnets, CIDRs and connectivity the way you want
- Static or Dynamic VPC and subnet CIDR assignments
- Deletes default VPC's (worldwide)
- AWS Network Firewall

1.2.3 1.2.3. Cross-Account Object Sharing

- VPC and Subnet sharing, including account level re-tagging (Per account security group 'replication')
- VPC attachments and peering (local and cross-account)
- Zone sharing and VPC associations
- Managed Active Directory sharing, including R53 DNS resolver rule creation/sharing
- Automated TGW inter-region peering
- Populate Parameter Store with all `user` objects to be used by customers' IaC

- Deploy and share SSM documents (4 provided out-of-box, ELB Logging, S3 Encryption, Instance Profile remediation, Role remediation)
- customer can provide their own SSM documents for automated deployment and sharing

1.2.4 1.2.4. Identity

- Creates Directory services (Managed Active Directory and Active Directory Connectors)
- Creates Windows admin bastion host auto-scaling group
- Set Windows domain password policies
- Set IAM account password policies
- Creates Windows domain users and groups (initial installation only)
- Creates IAM Policies, Roles, Users, and Groups
- Fully integrates with and leverages AWS SSO for centralized and federated login

1.2.5 1.2.5. Cloud Security Services

- Enables and configures the following AWS services, worldwide w/central designated admin account:
- GuardDuty w/S3 protection
- Security Hub (Enables designated security standards, and disables individual controls)
- Firewall Manager
- CloudTrail w/Insights and S3 data plane logging
- Config Recorders/Aggregator
- Conformance Packs and Config rules (95 out-of-box NIST 800-53 rules, 2 custom rules, customizable per OU)
- Macie
- IAM Access Analyzer
- CloudWatch access from central designated admin account (and setting Log group retentions)

1.2.6 1.2.6. Other Security Capabilities

- Creates, deploys and applies Service Control Policies
- Creates Customer Managed KMS Keys (SSM, EBS, S3), EC2 key pairs, and secrets
- Enables account level default EBS encryption and S3 Block Public Access
- Configures Systems Manager Session Manager w/KMS encryption and centralized logging
- Configures Systems Manager Inventory w/centralized logging
- Creates and configures AWS budgets (customizable per OU and per account)
- Imports or requests certificates into AWS Certificate Manager
- Deploys both perimeter and account level ALB's w/Lambda health checks, certificates and TLS policies
- Deploys & configures 3rd party firewall clusters and management instances (leverages marketplace)
- Gateway Load Balancer w/auto-scaling and VPN IPSec BGP ECMP deployment options
- Protects Accelerator deployed and managed objects
- Sets Up SNS Alerting topics (High, Medium, Low, Blackhole priorities)
- Deploys CloudWatch Log Metrics and Alarms
- Deploys customer provided custom config rules (2 provided out-of-box, no EC2 Instance Profile/Permissions)

1.2.7 1.2.7. Centralized Logging and Alerting

- Deploys an rsyslog auto-scaling cluster behind a NLB, all syslogs forwarded to CloudWatch Logs

- Centralized access to "Cloud Security Service" Consoles from designated AWS account
- Centralizes logging to a single centralized S3 bucket (enables, configures and centralizes)
- VPC Flow logs w/Enhanced metadata fields (also sent to CWL)
- Organizational Cost and Usage Reports
- CloudTrail Logs including S3 Data Plane Logs (also sent to CWL)
- All CloudWatch Logs (includes rsyslog logs)
- Config History and Snapshots
- Route 53 Public Zone Logs (also sent to CWL)
- GuardDuty Findings
- Macie Discovery results
- ALB Logs
- SSM Inventory
- Security Hub findings
- SSM Session Logs (also sent to CWL)
- Resolver Query Logs (also sent to CWL)
- Email alerting for CloudTrail Metric Alarms, Firewall Manager Events, Security Hub Findings incl. GuardDuty Findings
- **NEW** Optionally collect Organization and ASEA configuration and metadata in a new restricted log archive bucket

1.3 1.3. Relationship with AWS Landing Zone Solution (ALZ)

The ALZ was an AWS Solution designed to deploy a multi-account AWS architecture for customers based on best practices and lessons learned from some of AWS' largest customers. The AWS Accelerator draws on design patterns from the Landing Zone, and re-uses several concepts and nomenclature, but it is not directly derived from it, nor does it leverage any code from the ALZ. The Accelerator is a standalone solution with no dependence on ALZ.

1.4 1.4. Relationship with AWS Control Tower

The AWS Secure Environment Accelerator now leverages AWS Control Tower!

With the release of v1.5.0, the AWS Accelerator adds the capability to be deployed on top of AWS Control Tower. Customers get the benefits of the fully managed capabilities of AWS Control Tower combined with the power and flexibility of the Accelerators Networking and Security orchestration.

1.5 1.5. Accelerator Installation Process (Summary)

This summarizes the installation process, the full installation document can be found in the documentation section below.

- Create a config.json (or config.yaml) file to represent your organizations requirements (several samples provided)
- Create a Secrets Manager Secret which contains a GitHub token that provides access to the Accelerator code repository
- Create a unique S3 input bucket in the management account of the region you wish to deploy the solution and place your config.json and any additional custom config files in the bucket
- Download and execute the latest release installer CloudFormation template in your management accounts preferred 'primary' / 'home' region
- Wait for:
- CloudFormation to deploy and start the Code Pipeline (~5 mins)
- Code Pipeline to download the Accelerator codebase and install the Accelerator State Machine (~10 mins)
- The Accelerator State Machine to finish execution (~1.25 hrs Standalone version, ~2.25 hrs Control Tower Version)
- Perform required one-time post installation activities (configure AWS SSO, set firewall passwords, etc.)
- On an ongoing basis:
- Use AWS Organizations to create new AWS accounts, which will automatically be guardrailed by the Accelerator
- Update the config file in CodeCommit and run the Accelerator State Machine to:
- deploy, configure and guardrail multiple accounts at the same time (~25 min Standalone, ~50 min/account Control Tower)
- change Accelerator configuration settings (~25 min)

2. Installation & Upgrades

2.1 Accelerator Installation and Upgrades

This section contains information on the installation and upgrade procedures for ASEA.

- Installation
- Installation Guide
- Sample Configurations and Customization
- Calgary Region Configuration Sample
- State Machine Behavior
- Splitting the Config File
- Considerations with Existing Organizations
- Importing ALZ Accounts
- Open Releases
- Upgrades
- Upgrade Guide
- v1.5.0 Upgrade Instructions
- Functionality
- Services
- Pricing
- Architecture Diagrams
- Key Account & Capability Overview
- Centralized Logging Details
- Accelerator Object Naming
- Open Roadmap

2.2 Installation

2.2.1 1. Accelerator Installation Guide

1.1. Overview

We encourage customers installing the Accelerator to get the support of their local AWS account team (SA, TAM, CSM, ProServe) to assist with the installation of the Accelerator, as the Accelerator leverages, deploys, or orchestrates over 50 different AWS services.

Users are strongly encouraged to also read the Accelerator Operations/Troubleshooting Guide before installation and the FAQ while waiting for the installation to complete. The Operations/Troubleshooting Guide provides details as to what is being performed at each stage of the installation process, including detailed troubleshooting guidance.

These installation instructions assume one of the prescribed architectures is being deployed.

1.2. Prerequisites

1.2.1. GENERAL

- Management or root AWS Organization account (the AWS Accelerator cannot be deployed in an AWS sub-account)
- No additional AWS accounts need to be pre-created before Accelerator installation
- If required, a limit increase to support your desired number of new AWS sub-accounts (default limit is 10 sub-accounts)
- **recent changes to new AWS account limits are causing accelerator installation failures, please work with your local account team to increase your limits**
- Valid Accelerator configuration file, updated to reflect your requirements (see below)
- Determine your primary or Accelerator `control` or `home` region, this is the AWS region in which you will most often operate
- Government of Canada customers are still required to do a standalone installation at this time, please request standalone installation instructions from your Account SA or TAM
- The Accelerator *can* be installed into existing AWS Organizations - see caveats and notes here
- Existing AWS Landing Zone Solution (ALZ) customers are required to remove their ALZ deployment before deploying the Accelerator. Scripts are available to assist with this process.
- Changes to the Accelerator codebase are strongly discouraged unless they are contributed and accepted back to the solution. Code customization will block the ability to upgrade to the latest release and upgrades are encouraged to be done between quarterly to semi-annually. The solution was designed to be extremely customizable without changing code, existing customers following these guidelines have been able to upgrade across more than 50 Accelerator releases, while maintaining their customizations and gaining the latest bug fixes, features and enhancements without any developer or professional services based support. Please see this FAQ for more details.

1.3. Production Deployment Planning

1.3.1. GENERAL

For any deployment of the Accelerator which is intended to be used for production workloads, you must evaluate all these decisions carefully. Failure to understand these choices could cause challenges down the road. If this is a "test" or "internal" deployment of the Accelerator which will not be used for production workloads, you can leave the default config values.

Config file schema documentation (Draft)

1.3.2. OU STRUCTURE PLANNING

Plan your OU and core account structure carefully. By default, we suggest: `Security, Infrastructure, Central, Sandbox, Dev, Test, Prod`.

- The `Security` OU will contain the `Security` account, the `Log Archive` account, and the Organization `Management` account.
- The `Infrastructure` OU will hold the remainder of the accounts shared or utilized by the rest of the organization (`Shared Network`, `Perimeter`, and `Operations`).
- The remainder of the OUs correspond with major permission shifts in the SDLC cycle and NOT every stage an organization has in their SDLC cycle (i.e. QA or pre-prod would be included in one of the other OUs).
- The `Central` OU is used to hold accounts with workloads shared across Dev, Test, and Prod environments like centralized CI/CD tooling.
- The v1.5.0+ releases align the Accelerator OU and account structure with AWS multi-account guidance, splitting the `core` OU into the `Security` and `Infrastructure` OUs.

Note: While OUs can be renamed or additional OUs added at a later point in time, deployed AWS accounts CANNOT be moved between top-level OUs (guardrail violation), nor can top-level OUs easily be deleted (requires deleting all AWS accounts from within the OU first).

1.3.3. NETWORK CONFIGURATION PLANNING

If deploying the prescriptive architecture using the Full or Lite sample config files, you will need the following network constructs:

1. Six (6) RFC1918 Class B address blocks (CIDR's) which do not conflict with your on-premise networks (a single /13 block works well)
   - VPC CIDR blocks cannot be changed after installation, this is simply the way the AWS platform works, given everything is built on top of them. Carefully consider your address block selection.
   - one block for each OU, except Sandbox which is not routable (Sandbox OU will use a 7th non-routed address block)
   - the "core" Class B range will be split to support the Endpoint VPC and Perimeter VPC (with extra addresses remaining for future use)
   - Given a shared VPC architecture is leveraged (prevents stranded islands of CIDR blocks and reduces networking costs), we have assigned a class B address block to each VPC to future proof the deployment. Smaller customers can successfully deploy with a half class B CIDR block per shared VPC.
2. Two (2) RFC6598 /23 address blocks (Government of Canada (GC) requirement only)
   - Used for AWS Managed Active Directory (MAD) deployment and perimeter underlay network
   - non-GC customers can replace the RFC6598 address space with the extra unused addresses from the above RFC1918 CIDR range above (the App2 subnets in the Central VPC and the Perimeter VPC address space)
3. BGP ASN's for network routing, one for each of:
   - Transit Gateway (one unique ASN per TGW, multi-region example requires a second ASN)
   - IPSec VPN Firewall Cluster (if deployed)
   - VGW for Direct Connect connectivity (only shown in the config.multi-region-example.json)
   - For example: the Control Tower with Network Firewall example config requires a single BGP ASN for the TGW, the IPSec VPN example requires two BGP ASN's, and the multi-region example requires five unique BGP ASN's.

NOTE: Prior to v1.5.0 CIDR ranges were assigned to each VPC and subnet throughout the config file. This required customers to perform extensive updates across the config file when needing to move to specific IP ranges compatible with a customer's existing on-premise networks.

While this is still supported for those wanting to control exactly what address is used on every subnet, the solution has added support for dynamic CIDR assignments and the sample config files have been updated to reflect. New installs will have CIDR's pulled from CIDR pools, defined in the global-options section of the config file with state maintained in DynamoDB.

The v1.5.0 custom upgrade guide will provides details on the upgrade process and requirements to migrate to the new CIDR assignment system, if desired. A script was created to assist with this migration.

1.3.4. DNS, DOMAIN NAME, TLS CERTIFICATE PLANNING

If deploying the prescriptive architecture, you must decide on:

1. A unique Windows domain name (`organizationaws` / `organization.aws` , `organizationcloud` / `organization.cloud` , etc.). Given this is designed as the primary identity store and used to domain join all cloud hosted workloads, changing this in future is difficult. Pick a Windows domain name that does NOT conflict with your on-premise AD domains, ensuring the naming convention conforms to your organizations domain naming standards to ensure you can eventually create a domain trust between the MAD and on-premise domains/forests
2. DNS Domain names and DNS server IP's for on-premise private DNS zones requiring cloud resolution (can be added in future)
3. DNS Domain for a cloud hosted public zone `"public": ["organization.cloud-nuage.canada.ca"]` (can be added in future)
4. DNS Domain for a cloud hosted private zone `"private": ["organization.cloud-nuage.gc.ca"]` (can be added in future)
5. Wildcard TLS certificate for each of the 2 previous zones (can be added/changed in future)

1.3.5. EMAIL ADDRESS PLANNING

1. While you require a minimum of 6 ***unique*** email addresses (1 per sub-account being created), we recommend at least 20 ***unique*** email ALIASES associated with a single mailbox, never used before to open AWS accounts, such that you do not need to request new email aliases every time you need to create a new AWS account and they can all be monitored via a single mailbox. These email addresses can ***never*** have been used to previously open an AWS account.
2. You additionally require email addresses for the following additional purposes (these can be existing monitored mailboxes and do not need to be unique):

- Accelerator execution (state machine) notification events (1 address)
- High, Medium and Low security alerts (3 addresses if you wish to segregate alerts)
- Budget notifications

1.3.6. CENTRALIZED INGRESS/EGRESS FIREWALLS

As of v1.5.0 the Accelerator offers multiple automated firewall deployment options:

a) AWS Network Firewall (native AWS Cloud service)

```
- Defined in the config file as part of a VPC
```

b) 3rd party firewalls interconnected to the cloud tenancy via IPSec VPN (Active/Active using BGP + ECMP)

- Defined in the config file under deployments w/TGW VPN attachments
- this was the only automated option prior to v1.5.0
- a sample Fortinet Fortigate configuration is provided (both PAYGO and BYOL supported)
- For Fortinet BYOL, requires minimum 2 valid license files (evaluation licenses adequate) (can be added in future)

c) 3rd party firewalls interconnected to the cloud tenancy via Gateway Load Balancer (GWLB) in an auto-scaling group

- Defined in the config file under both deployments and load balancers
- a sample Checkpoint CloudGuard configuration is provided (both PAYGO and BYOL supported)

d) Customer gateway (CGW) creation, to enable connectivity to on-premises firewalls or manually deployed cloud firewalls

- Defined in the config file under deployments w/TGW VPN attachments (but without an AMI or VPC association)

Examples of each of the firewall options have been included as variants of the Lite config file example.

Note: While we only provide a single example for each 3rd party implementation today, the implementations are generic and should be usable by any 3rd party firewall vendor, assuming they support the required features and protocols. The two examples were driven by customer demand and heavy lifting by the 3rd party vendor. We look forward to additional vendors developing and contributing additional sample configurations. For new 3rd party integrations, we encourage the use of the GWLB approach.

1.3.7. OTHER

1. We recommend installing with the default Accelerator Name (`ASEA`) and Accelerator Prefix (`ASEA-`), but allow customization. Prior to v1.5.0 the defaults were (`PBMM`) and (`PBMMAccel-`) respectively.
   - the Accelerator name and prefix ***CANNOT*** be changed after the initial installation;
   - the Accelerator prefix including the mandatory dash cannot be longer than 10 characters.
2. New installations, which now leverage Control Tower, require the `organization-admin-role` be set to `AWSControlTowerExecution` . Existing standalone installations will continue to utilize their existing role name for the `organization-admin-role` , typically `OrganizationAccountAccessRole` , as this role is used by AWS Organizations by default when no role name is specified while creating AWS accounts through the AWS console.
   - the Accelerator leverages this role name to create all new accounts in the organization;
   - this role name, as defined in the config file, ***MUST*** be utilized when manually creating all new sub-accounts in the Organization;
   - existing installs wishing to change the role name are required to first deploy a new role with a trust to the root account, in all accounts in the organization.

1.4. Accelerator Pre-Install Steps

1.4.1. GENERAL

Before installing, you must first:

1. Login to the Organization **Management (root) AWS account** with `AdministratorAccess`.
2. ***Set the region to your desired*** ***`home`*** ***region*** (i.e. `ca-central-1`)
3. Install AWS Control Tower:

- Government of Canada customers are *required* to skip this step
- OU and account names can ONLY be customized during initial installation. These values MUST match with the values supplied in the Accelerator config file.

a. Go to the AWS Control Tower console and click `Set up landing zone`

b. Select your `home` region (i.e. `ca-central-1`) - the Accelerator home region must match the Control Tower home region

c. Leave the Region deny setting set to `Not enabled` - the Accelerator needs a customized region deny policy

d. Select *all* regions for `Additional AWS Regions for governance`, click `Next`

- The Control Tower and Accelerator regions MUST be properly aligned
- If a region is not `governed` by Control Tower, it must NOT be listed in `control-tower-supported-regions`
- To manage a region requires the region:
- be enabled in Control Tower (if supported)
- added to the config file `control-tower-supported-regions` list (if supported)
- added to the config file `supported-regions` list (even if not supported by Control Tower, as the Accelerator can manage regions not yet supported by Control Tower, but only when NOT listed in `control-tower-supported-regions`)
- While we highly recommend guardrail deployment for all AWS enabled by default regions, at minimum
- the home region MUST be enabled in Control Tower and must be listed in `control-tower-supported-regions`
- both the home-region and ${GBL*REGION} must be listed in `supported-regions`

e. For the `Foundational OU`, leave the default value `Security`

f. For the `Additional OU` provide the value `Infrastructure`, click `Next`

g. Enter the email addresses for your `Log Archive` and `Audit` accounts, change the `Audit` account name to `Security`, click `Next` - OU and account names can ONLY be customized during initial installation. OU names, account names and email addresses _must* match identically with the values supplied in the Accelerator config file.

h. Select `Enabled` for AWS CloudTrail configuration (if not selected), click `Next`

i. Click `Set up landing zone` and wait ~60 minutes for the Control Tower installation to complete

j. Select `Add or register organizational units`, Click `Add an OU`

k. Type `Dev`, click `Add`, wait until the OU is finished provisioning (or it will error)

l. Repeat step 9 for each OU (i.e. `Test`, `Prod`, `Central`, `Sandbox`)

m. Select `Account factory`, Edit, Subnets: 0, Deselect all regions, click `Save`

n. In AWS Organizations, move the Management account from the `root` OU into the `Security` OU

4. Verify:

a. AWS Organizations is enabled in `All features` mode

- if required, navigate to AWS Organizations, click `Create Organization`, `Create Organization`

b. Service Control Policies are enabled

- if required, in Organizations, select `Policies`, `Service control policies`, `Enable service control policies`

5. Verify the Organization Management (root) account email address

- In AWS Organizations, Settings, "Send Verification Request"
- Once it arrives, complete the validation by clicking the validation link in the email

6. Create a new KMS key to encrypt your source configuration bucket (you can use an existing key)
   - AWS Key Management Service, Customer Managed Keys, Create Key, Symmetric, and then provide a key name (`ASEA-Source-Bucket-Key`), Next
   - Select a key administrator (Admin Role or Group for the Organization Management account), Next
   - Select key users (Admin Role or Group for the Organization Management account), Next
   - Validate an entry exists to "Enable IAM User Permissions" (critical step if using an existing key)
   - `"arn:aws:iam::123456789012:root"` , where `123456789012` is your ***Organization Management*** account ID.
   - Click Finish
   - Select the new key, Select `Key Rotation` , `Automatically rotate this CMK every year` , click Save.
7. Enable `"Cost Explorer"` (My Account, Cost Explorer, Enable Cost Explorer)
   - With recent platform changes, Cost Explorer *may* now be auto-enabled (unable to confirm)
8. Enable `"Receive Billing Alerts"` (My Account, Billing Preferences, Receive Billing Alerts)
9. It is ***extremely important*** that ***all*** the account contact details be validated in the Organization Management (root) account before deploying any new sub-accounts.
   - This information is copied to every new sub-account on creation.
   - Subsequent changes to this information require manually updating it in ***each*** sub-account.
   - Go to `My Account` and verify/update the information lists under both the `Contact Information` section and the `Alternate Contacts` section.
   - Please ESPECIALLY make sure the email addresses and Phone numbers are valid and regularly monitored. If we need to reach you due to suspicious account activity, billing issues, or other urgent problems with your account - this is the information that is used. It is CRITICAL it is kept accurate and up to date at all times.

1.4.2. CREATE GITHUB PERSONAL ACCESS TOKEN AND STORE IN SECRETS MANAGER

As of v1.5.0, the Accelerator offers deployment from either GitHub or CodeCommit:

GitHub (recommended)

1. You require a GitHub access token to access the code repository
2. Instructions on how to create a personal access token are located here.
3. Select the scope `public_repo` underneath the section `repo: Full control over private repositories` .
4. Store the personal access token in Secrets Manager as plain text. Name the secret `accelerator/github-token` (case sensitive).
   - Via AWS console
   - Store a new secret, and select `Other type of secrets` , `Plaintext`
   - Paste your secret with no formatting no leading or trailing spaces (i.e. completely remove the example text)
   - Select the key you created above (`ASEA-Source-Bucket-Key`),
   - Set the secret name to `accelerator/github-token` (case sensitive)
   - Select `Disable rotation`

CodeCommit (alternative option)

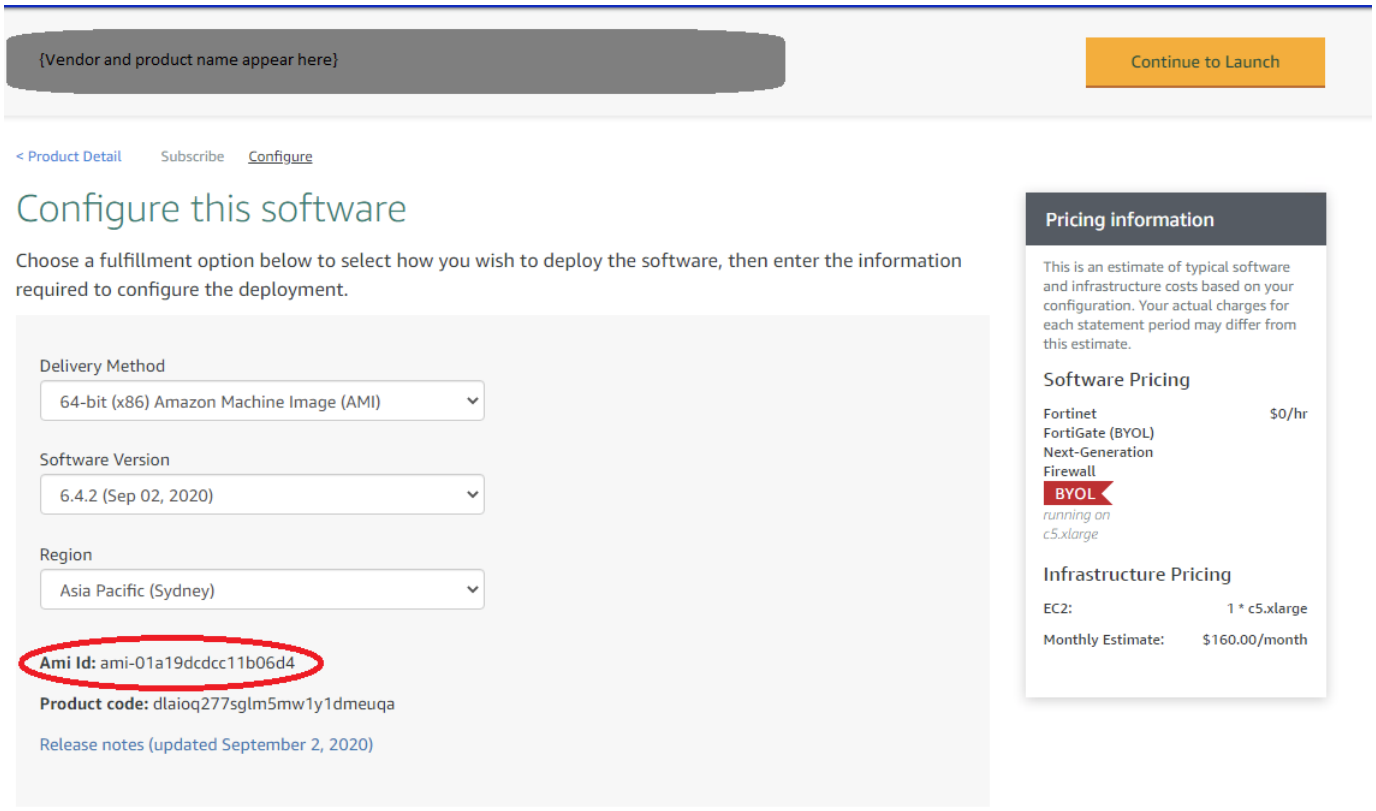
Multiple options exist for downloading the GitHub Accelerator codebase and pushing it into CodeCommit. As this option is only for advanced users, detailed instructions are not provided.

1. In your AWS Organization Management account, open CodeCommit and create a new repository named `aws-secure-environment-accelerator`
2. Go to GitHub and download the repository `Source code` zip or tarball for the release you wish to deploy
   - Do NOT download the code off the main GitHub branch, this will leave you in a completely unsupported state (and with beta code)
3. Push the extracted codebase into the newly created CodeCommit repository, maintaining the file/folder hierarchy (ensuring that the root of the repository on code commit looks the same as the root of the repository on github)
4. Set the default CodeCommit branch for the new repository to main
5. Create a branch following the Accelerator naming format for your release (i.e. `release/v1.5.5`)

1.4.3. AWS INTERNAL (EMPLOYEE) ACCOUNTS ONLY

If deploying to an internal AWS employee account and installing the solution with a 3rd party firewall, you need to enable Private Marketplace (PMP) before starting:

1. In the Organization Management account go here: https://aws.amazon.com/marketplace/privatemarketplace/create
2. Click `Create a Private Marketplace`, and wait for activation to complete
3. Go to the "Account Groups" sub-menu, click `Create account group`
4. Enter an Account Group Title (i.e. `Default`) and `Add` the Management (root) account number in `Associate AWS account`
5. Associate the default experience `New Private Marketplace`, then click `Create account group` and wait for it to create
6. Go to "Experiences" sub-menu, select `New Private Marketplace`
7. Select the "Settings" sub-tab, and click the `Not Live` slider to make it `Live` and wait for it to complete
8. Ensure the "Software requests" slider is set to `Requests off` and wait for it to complete
9. Change the name field (i.e. append `-PMP`) and change the color, so it is clear PMP is enabled for users, click `Update`
10. Go to the "Products" sub-tab, then select the `All AWS Marketplace products` nested sub-tab
11. Search Private Marketplace for the Fortinet or Checkpoint products and select
    - `Fortinet FortiGate (BYOL) Next-Generation Firewall` and
    - `Fortinet FortiManager (BYOL) Centralized Security Management` **or**
    - `CloudGuard Network Security for Gateway Load Balancer - BYOL` and
    - `Check Point Security Management (BYOL)`
12. Select "Add" in the top right
    - Due to PMP provisioning delays, this sometimes fails when attempted immediately following enablement of PMP or if adding each product individually - retry after 20 minutes.
13. While not used in this account, you must now subscribe to the two subscriptions and accept the EULA for each product (you will need to do the same in the perimeter account, once provisioned below)
    - To subscribe, select the "Approved products" tab
    - Click on the product you want to subscribe, in this case `Fortinet FortiGate (BYOL) Next-Generation Firewall` and `Fortinet FortiManager (BYOL Centralized Security Management` **or** `CloudGuard Network Security for Gateway Load Balancer - BYOL` and `Check Point Security Management (BYOL)`
    - Click on "Continue to Subscribe"
    - Click on "Accept Terms" and wait for subscription to be completed
    - If you are deploying in any region except ca-central-1 or wish to switch to a different license type, you need the new AMI IDs. After successfully subscribing, continue one more step and click the “Continue to Configuration”. When you get the below screen, select your region and version (**Fortinet** `v6.4.7`, **Checkpoint Mgmt** `R81.10-335.883` and **CloudGuard** `R80.40-294.374` recommended at this time). Marketplace will provide the required AMI ID. Document the two AMI IDs, as you will need to update them in your config.json file below.



1.5. Basic Accelerator Configuration

1. Select a sample config file as a baseline starting point

- **IMPORTANT: Use a config file from the GitHub code branch you are deploying from, as valid parameters change over time. The `main` branch is NOT the current release and often will not work with the GA releases.**
- sample config files can be found in this folder;
- descriptions of the sample config files and customization guidance can be found here;
- unsure where to start, use the `config.lite-CTNFW-example.json`, where CTNFW is for Control Tower w/NFW;
- These configuration files can be used, as-is, with only minor modification to successfully deploy the sample architectures;
- On upgrades, compare your deployed configuration file with the latest branch configuration file for any new or changed parameters;

2. At minimum, you MUST update the AWS account names and email addresses in the sample file:

- For existing accounts, they *must* match identically to both the account names and email addresses defined in AWS Organizations (including the management account);
- For new accounts, they must reflect the new account name/email you want created;
- All new AWS accounts require a unique email address which has never before been used to create an AWS account;
- When updating the budget or SNS notification email addresses within the sample config, a single email address for all is sufficient;
- Update the IP address in the "alarm-not-ip" variable with your on-premise IP ranges (used for the AWS-SSO-Authentication-From-Unapproved-IP alarm);
- If deploying the Managed AD, update the dns-domain, netbios-domain, log-group-name, as well as the AD users and groups that will be created;
- For a test deployment, the remainder of the values can be used as-is;
- While it is generally supported, we recommend not adding more than 1 or 2 workload accounts to the config file during the initial deployment as it will increase risks of hitting a limit. Once the Accelerator is successfully deployed, add the additional accounts to the config file and rerun the state machine.
- More information here on the fields in the config file that need to be updated.

3. A successful deployment of the prescriptive architecture requires VPC access to 9 AWS endpoints, you cannot remove both the perimeter firewalls (all public endpoints) and the 9 required central VPC endpoints from the config file (ec2, ec2messages, ssm, ssmmessages, cloudformation, secretsmanager, kms, logs, monitoring).
4. When deploying to regions other than `ca-central-1`, you need to modify your config file as follows (for Canada Central 1, the AMI IDs are pre-populated for you):

a. Update the firewall and firewall manager AMI IDs to reflect your home regions regional AMI IDs (see 2.3.3, item 13), making sure you select the right version and region per the recommendations.

b. Validate all the Interface Endpoints defined in your config file are supported in your home region (i.e. Endpoint VPC). Remove unsupported endpoints from the config file, add additional endpoints as available.

c. If you are installing into a home region which is explicitly named in any of the replacements\addl_regions_x, remove it from the list. If deploying in us-east-1, remove ${GBL_REGION}.

5. Create an S3 bucket in your Organization Management account `your-bucket-name`

- you must supply this bucket name in the CFN parameters *and* in the config file (`global-options\central-bucket`)
- the bucket name *must* be the same in both spots
- the bucket must have versioning enabled
- the bucket must be `S3-KMS` encrypted using the `ASEA-Source-Bucket-Key` created above

6. Place your customized config file(s), named `config.json` (or `config.yaml`), in your new bucket
7. If required, place the firewall configuration and license files in the folder and path defined in the config file

- For AWS Network Firewall: `nfw/nfw-example-policy.json`
- For Fortinet: `firewall/firewall-example.txt`, `firewall/license1.lic` and `firewall/license2.lic`
- We have made a sample available here: `./reference-artifacts/Third-Party/`
- the sample configures an active / active firewall pair with two tunnels per firewall
- If you updated your perimeter VPC subnet names, you must also make these changes in your firewall-example.txt file
- If you don't have any license files, update the config file with an empty array (`"license": []`). Do NOT use the following: `[""]` .
- The basic Checkpoint configuration is stored directly in config.json

8. Place any defined certificate files in the folder and path defined in the config file

- i.e. `certs/example1-cert.key` , `certs/example1-cert.crt`
- Sample available here: `./reference-artifacts/Certs-Sample/*`
- Ideally you would generate real certificates using your existing certificate authority
- Should you wish, instructions are provided to aid in generating your own self-signed certificates (Self signed certificates are NOT secure and simply for demo purposes)
- Use the examples to demonstrate Accelerator TLS functionality only

9. Detach ***ALL*** SCPs (except `FullAWSAccess` which remains in place) from all OU's and accounts before proceeding

- For Control Tower based installs do NOT detach Control Tower SCPs (i.e. aws-guardrails-xxxxxx)
- Installation **will fail** if this step is skipped

1.6. Installation

1. You can find the latest release in the repository here.

- We only support new installations of v1.5.5 or above (older releases continue to function)

2. Download the CloudFormation (CFN) template for the release you plan to install (either `AcceleratorInstallerXXX.template.json` for GitHub or `AcceleratorInstallerXXX-CodeCommit.template.json` for CodeCommit)
3. Use the provided CloudFormation template to deploy a new stack in your Management (root) AWS account

- As previously stated we do not support installation in sub-accounts

4. Login to your Organization Management account and ***make sure you are in your desired `home` region*** (i.e. `ca-central-1`) (your desired primary or control region)

5. Navigate to **CloudFormation** in the AWS Console and click `Create stack with new resources (standard)`, then
   - Select "Template is ready"
   - For the "Specify template" select "Upload a template file"
   - Select the `*.template.json` file you downloaded in step 2 above
   - Click Next
6. Fill out the required parameters - ***LEAVE THE DEFAULTS UNLESS SPECIFIED BELOW***
   - Specify `Stack Name` STARTING with `ASEA-` (case sensitive) suggest a suffix of `orgname` or `username`
   - Change `ConfigS3Bucket` to the name of the bucket you created above `your-bucket-name`
   - Add an `Email` address to be used for State Machine Status notification
   - The `GitHub Branch` should point to the release you selected
   - if upgrading, change it to point to the desired release
   - the latest stable branch is currently `release/v1.5.5`, case sensitive
   - click `Next`
7. Finish deploying the stack
   - Apply a tag on the stack, Key= `Accelerator`, Value= `ASEA` (case sensitive).
   - **ENABLE STACK TERMINATION PROTECTION** under `Stack creation options`
   - Click `Next`, Acknowledge resource creation, and click `Create stack`
   - The stack typically takes under 5 minutes to deploy.
8. Once deployed, you should see a CodePipeline project named `ASEA-InstallerPipeline` in your account. This pipeline connects to GitHub, pulls the code from the prescribed branch and deploys the Accelerator state machine.
   - if the CloudFormation fails to deploy with an `Internal Failure`, or, if the pipeline fails connecting to GitHub, then:
   - fix the issue with your GitHub secret created in section 2.3.2, then delete the Installer CloudFormation stack you just deployed, and restart at step 3 of this section.
9. For new stack deployments, when the stack deployment completes, the Accelerator state machine will automatically execute (in Code Pipeline). When upgrading you must manually `Release Change` to start the pipeline.
10. **While the pipeline is running:**
    - review the list of Known Installation Issues in the section below
    - review the Accelerator Basic Operation and Frequently Asked Questions (FAQ) Document
11. Once the pipeline completes (~10 mins), the main state machine, named `ASEA-MainStateMachine_sm`, will start in Step Functions
12. The state machine time is dependent on the quantity of resources being deployed. On an initial installation of a more complex sample configuration files, it takes approximately 2 hours to execute (depending on the configuration file). Timing for subsequent executions depends entirely on what resources are changed in the configuration file, but often takes as little as 20 minutes.
    - While you can watch the state machine in Step Functions, you will also be notified via email when the State Machine completes (or fails). Successful state machine executions include a list of all accounts which were successfully processed by the Accelerator.
13. The configuration file will be automatically moved into Code Commit (and deleted from S3). From this point forward, you must update your configuration file in CodeCommit.
14. You will receive an email from the State Machine SNS topic and the 3 SNS alerting topics. Please confirm all four (4) email subscriptions to enable receipt of state machine status and security alert messages. Until completed, you will not receive any email messages (must be completed within 7-days).
15. If the state machine **fails**:
    - Refer to the Troubleshooting Guide for instructions on how to inspect and retrieve the error
    - You can also refer to the FAQ and Known Installation Issues
    - Once the error is resolved, re-run the step function `ASEA-MainStateMachine_sm` using `{"scope": "FULL","mode": "APPLY"}` as input

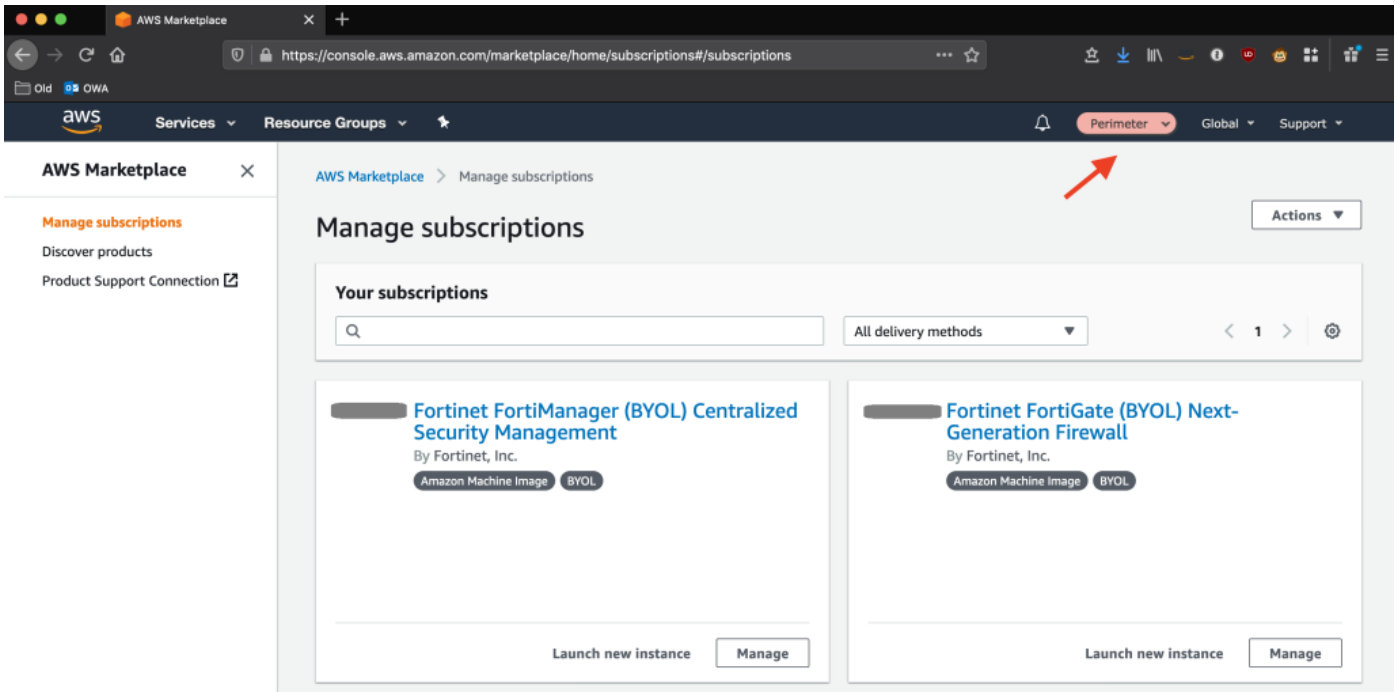
16. If deploying a prescriptive architecture with 3rd party firewalls, after the perimeter account is created in AWS Organizations, but before the Accelerator reaches Stage 2:

a. NOTE: If you miss the step, or fail to execute it in time, no need to be concerned, you will simply need to re-run the main state machine (`ASEA-MainStateMachine_sm`) to deploy the firewall (no SM input parameters required)

b. Login to the **perimeter** sub-account (Assume your `organization-admin-role`)

c. Activate the 3rd party vendor firewall and firewall manager AMI's in the AWS Marketplace

- Navigate back to your private marketplace
- Note: Employees should see the private marketplace, including the custom color specified in prerequisite step 4 above.
- Select "Discover products" from the side bar, then in the "Refine Results" select "Private Marketplace => Approved Products"
- Subscribe and Accept the Terms for each product (firewall and firewall manager)
- When complete, you should see the marketplace products as subscriptions **in the Perimeter account**:



17. If deploying the prescriptive architecture, once the main state machine (`ASEA-MainStateMachine_sm`) completes successfully, confirm the status of your perimeter firewall deployment

- If you have t2.micro ec2 instances running in any account which had the account-warming flag set to true, they will be removed on the next state machine execution;
- If your perimeter firewalls were defined but not deployed on first run, you will need to rerun the state machine. This happens when:

a. you were unable to activate the firewall AMI's before stage 2 (step 16)

b. we were not able to fully activate your account before we were ready to deploy your firewalls. This case can be identified by a running EC2 micro instance in the account, or by looking for the following log entry 'Minimum 15 minutes of account warming required for account'.

c. In these cases, simply select the `ASEA-MainStateMachine_sm` in Step Functions and select `Start Execution` (no SM input parameters required)

1.6.1. KNOWN INSTALLATION ISSUES

Current Issues:

- If dns-resolver-logging is enabled, VPC names containing spaces are not supported at this time as the VPC name is used as part of the log group name and spaces are not supported in log group names. By default in many of the sample config files, the VPC name is auto-generated from the OU name using a variable. In this situation, spaces are also not permitted in OU names (i.e. if any account in the OU has a VPC with resolver logging enabled and the VPC is using the OU as part of its name)
- On larger deployments we are occasionally seeing state machine failures when `Creating Config Recorders`. Simply rerun the state machine with the input of `{"scope": "FULL", "mode": "APPLY"}`.
- Occasionally CloudFormation fails to return a completion signal. After the credentials eventually fail (1 hr), the state machine fails. Simply rerun the state machine with the input of `{"scope": "FULL", "mode": "APPLY"}`
- If the State Machine fails on an initial execution of NEW-ACCOUNTS, it must be re-run to target the failed accounts (i.e. with `{"scope": "FULL", "mode": "APPLY"}`) or the new sub-accounts will fail to be properly guardrailed

Issues in Older Releases:

- New installs to releases prior to v1.5.5 are no longer supported.
- Upgrades to releases prior to v1.5.5 are no longer supported.
- Upgrades to v1.3.9 in preparation for an upgrade to v1.5.5 may be possible with manual workarounds.
- FROM 2022-08-07 to 2022-10-12: An issue with the version of cfn-init in the "latest" AWS standard Windows AMI will cause the state machine to fail during a new installation when deploying an RDGW host. RDGW hosts in existing deployments will fail to fully initialize if the state machine is or has been recently run and the auto-scaling group subsequently refreshes the host (default every 7 days).
- To temporarily workaround this issue, assume an administrative role in your `operations` account, open Systems Manager Parameter Store, and `Create parameter` with a Name of `/asea/windows-ami` and a value of `ami-0d336ea070bc06fb8` (which is the previous good AMI in ca-central-1), accepting the other default values. Update your config file to point to this new parameter by changing `image-path` (under \deployments\mad) to `/asea/windows-ami` instead of `/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-Base`. Rerun your state machine. If you have an existing RDGW instance it should be terminated to allow the auto-scaling group to redeploy it. In other regions you will need to lookup the previous working ami-id (you cannot use `ami-0d336ea070bc06fb8`)
- **This issue was resolved with the 2022-10-12 Windows AMI release. Customers that implemented this workaround must revert the above config file entry and rerun their state machines (the above AMI has been deprecated).**

1.7. Post Installation

The Accelerator installation is complete, but several manual steps remain:

1. Enable and configure AWS SSO in your `home` region (i.e. ca-central-1)

- **NOTE: AWS SSO has been renamed to AWS IAM Identity Center (IdC). The IdC GUI has also been reworked. The below steps are no longer click-by-click accurate. An update to the below documentation is planned, which will also include instructions to delegate AWS IdC administration to the Operations account enabling connecting IdC directly to MAD, rather than through an ADC.**
- Login to the AWS Console using your Organization Management account
- Navigate to AWS Single Sign-On, click `Enable SSO`
- Set the SSO directory to AD ("Settings" => "Identity Source" => "Identity Source" => click `Change`, Select Active Directory, and select your domain from the list)
- Under "Identity Source" section, Click `Edit` beside "Attribute mappings", then set the `email` attribute to: `${dir:email}` and click `Save Changes`
- Configure Multi-factor authentication, we recommend the following minimum settings:
- Every time they sign in (always-on)
- Security key and built-in authenticators
- Authenticator apps
- Require them to provide a one-time password sent by email to sign in
- Users can add and manage their own MFA devices
- Create all the default permission sets and any desired custom permission sets
- e.g. Select `AWS accounts` from the side bar, select "Permission sets" tab then `Create permission set`
- `Use an existing job function policy` => Next
- Select job function policy `AdministratorAccess`
- Add Tags, Review and Create
- repeat for each default permission set and any required custom permission sets
- For Control Tower based installations, remove the orphaned Permission Sets from each AWS accounts (select the account, expand Permission Sets, click Remove for each)
- Map MAD groups to permission sets and accounts
- Select `AWS accounts` from the side bar and select `AWS organization` tab
- Select the accounts you want to map to each MAD group and click `Assign users`
- Select your DNS domain e.g. `example.local`, and search for the group you would like to assign (e.g. `aws-` for the pre-created groups) and click `Search connected directory`
- Select the desired group `aws-log-archive-View`
- Select the permission set you would like to assign to the MAD group to (e.g. `ViewOnlyAccess`)
- Click `Finish` (Note: if it fails during provisioning, simply select the failed accounts and click on "Retry changes")
- AWS SSO should be the primary mechanism for all access to all AWS accounts in your Organization, to determine or update the login page for your organization:
- Click on `Dashboard` within the AWS SSO console and note the `User portal URL`
- Share this url with all your users
- NOTE: the url prefix can only be changed ONCE (ever) using the settings tab, so update with caution
- Any pre-created AWS MAD users passwords are available in secrets manager in the AWS management account. To reset these passwords login to the Operations account through AWS SSO, then:
- Navigate to "Directory Service" , select the directory => Actions => Reset user password
- Users can change their passwords from any MAD domain connected instance

2. Configure the new alb-forwarding feature (added in v1.5.0)

- AWS ALB's are published using DNS names which resolve to backing IP's which could silently change at any time due to a scaling event, maintenance, or a hardware failure. While published as a DNS name, ALB's can only target IP addresses. This presents a challenge as we need the ALB's in the perimeter account to target ALB's in the various back-end workload accounts;
- Alb-forwarding solves this problem by executing a small snippet of code every 60 seconds which updates managed ALB listeners with any IP changes, ensuring any managed flows do not go offline. This removes the requirement to leverage a 3rd party appliance to perform NAT to a DNS name;
- Alb-forwarding is enabled on the perimeter VPC in the perimeter account in many of the sample configs and can be used with Accelerator created or customer created ALB's. This setting causes a new DynamoDB table named `ASEA-Alb-Ip-Forwarding-vpc-*` to be created in the perimeter account. Additionally the parameter store in the perimeter account contains the details of all Accelerator deployed ALB's across the environment for easy reference;
- Steps to configure:
- First you need to manually create a listener on the front-end ALB (without a target group), multiple listeners are supported;
- Next, for each application that needs to be published, a record needs to be added to the DynamoDB table, see sample below;
- Records can be added to the table for any ALB in the account running the alb-forwarding tool. Records can be added at any time. DDB change logs will trigger the initial creation of the appropriate target group(s) and IP addresses will be verified and updated every 60 seconds thereafter.

mple DynamoDB JSON to add an entry to the table:

```
{
  "id": "App1",
  "targetAlbDnsName": "internal-Core-mydevacct1-alb-123456789.ca-central-1.elb.amazonaws.com",
  "targetGroupDestinationPort": 443,
  "targetGroupProtocol": "HTTPS",
  "vpcId": "vpc-0a6f44a80514daaaf",
  "rule": {
    "sourceListenerArn": "arn:aws:elasticloadbalancing:ca-central-1:123456789012:listener/app/Public-DevTest-perimeter-alb/b1b12e7a0c412bf3/ef9b022a4fdd8bdf",
    "condition": {
      "paths": ["/img/*", "/myApp2"],
      "hosts": ["aws.amazon.com"],
      "priority": 30
    }
  }
}
```

```
-   where `id` is any unique text, `targetAlbDnsName` is the DNS address for the backend ALB for this application (found in parameter store), `vpcId` is the vpc ID containing the front-end ALB (in this account), `sourceListenerArn` is the arn of the listener of the front-end ALB, `paths` and `hosts` are both optional, but one of the two must be supplied. Finally, `priority` must be unique and is used to order the listener rules. Priorities should be spaced at least 40 apart to allow for easy insertion of new applications and forwarder rules.

-   the provided `targetAlbDnsName` must resolve to addresses within a [supported](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html) IP address space.
```

3. On a per role basis, you need to enable the CWL Account Selector in the Security and the Operations accounts, in each account:

- Go to CloudWatch, Settings, Under `Cross-account cross-region` select `Configure`, Under `View cross-account cross-region` select `Edit`, choose `AWS Organization account selector`, click `Save changes`

4. Configure central Ingress/Egress firewalls, if deployed

- Layer 3/4 `appliance` based inspection is an optional feature

General

- If deployed, login to any 3rd party firewalls and firewall manager appliances and update any default passwords;
- Tighten security groups on the 3rd party firewall instances (using the Accelerator configuration file), further limiting access to firewall management interfaces to a set of designated and controlled CIDR ranges;
- Update the firewall configuration per your organization's security requirements and best practices;
- Diagrams reflecting perimeter traffic flows when NFW and/or GWLB are used can be found here on slides 6 through 9.

AWS Network Firewall

- The AWS Network Firewall policies and rules deployed by the Accelerator, can only be updated using the Accelerator. Customers wishing to manage the AWS Network Firewall from the console GUI, must create a new policy with new rules created through the console and then manually associate this new policy to the Accelerator deployed Network Firewall. Customers can choose either option, but they cannot be mixed to ensures that Accelerator updates do not overwrite console based updates.

Fortinet

- Manually update firewall configuration to forward all logs to the Accelerator deployed NLB addresses fronting the rsyslog cluster
- login to each firewall, select `Log Settings`, check `Send logs to syslog`, put the NLB FQDN in the `IP Address/FQDN` field (stored in parameter store of perimeter account)
- Manually update the firewall configuration to connect perimeter ALB high port flows through to internal account ALB's
- Note: while this option is still available, a new alb-forwarding mechanism is available in v1.5.0+ (see section 2 above) which simplifies and eliminates this more complicated "NAT to DNS name" option;
- login to each firewall, switch to `FG-traffic` vdom, select `Policies & Objects`, select `Addresses`, Expand `Addresses`
- Set `Prod1-ALB-FQDN` to point to a reliable sub-account ALB FQDN, this is used for full-path health checks on ***all*** ALB's
- Set additional `DevX-ALB-FQDN`, `TestX-ALB-FQDN` and `ProdX-ALB-FQDN` to point to workload account ALB FQDNs
- Two of each type of ALB FQDN records have been created, when you need more, you need to create BOTH an additional FQDN and a new VIP, per ALB
- Each new VIP will use a new high port (i.e. 7007, 7008, etc.), all of which map back to port 443
- Detailed steps can be read here.

Checkpoint

- Download and install Checkpoint SmartConsole client on your PC (only available for Windows, if you have a Mac, you can use Workspaces to install the client)
- Retrieve the firewall and firewall manager SSL certificate from secrets manager in the perimeter account and save to a pem file (convert to ppk on Windows)
- Login to the `AWS Console` for the `Perimeter` account
- Navigate to `Secrets Manager` and select `Secrets`
- Look for Secret name `ASEA/keypair/ASEA-Appliances` and click on it
- Go to the `Secret value` section and click on `Retrieve secret value`
- Make sure you have the `Plaintext` tab selected and copy the content to a .pem file using your preferred text editor. (the file will need to be converted to ppk format on Windows)
- Wait approx. 25 min. after the managers "Launch time" and then SSH into the Firewall Manager using the SSL certificate (.pem file retrieved above) and login user `admin`
- The firewall manager EC2 instance name is `ASEA-Checkpoint-FirewallMgr`
- Once you SSH successfully, execute the following commands:
- `set user admin password`
- `set expert-password`
- `set user admin shell /bin/bash`
- `save config`
- The following commands are useful for troubleshooting (in expert mode):
- `autoprov_cfg -v` (check cme at Take 155 or greater)
- `autoprov_cfg show all` (check cme configuration)
- `cat /var/log/aws-user-data.log` (validate bootstrap, file should end with `"Publish operation" succeeded (100%)`)
- `tail -f /var/log/CPcme/cme.log` (watch to ensure it finds the instances, establishes SIC and adds the nodes)
- Login to SmartConsole, and update the firewall policy per your organizations security requirements
- An outbound rule allowing http and https should exist
- From the RDGW host in Operations, test to see if outbound web browsing is enabled
- NOTES:
- No best practice or security configuration has been configured on the Checkpoint firewalls. These firewalls have been configured to work with GWLB, but otherwise have the default/basic Checkpoint out-of-box configuration installed
- Do NOT reboot the Checkpoint appliances until bootstrap is complete (~25 minutes for the manager), or you will be required to redeploy the instance

5. Recover root passwords for all sub-accounts and apply strong passwords

- Process documented here

6. Enable MFA for **all** IAM users and **all** root account users, recommendations:

- Yubikeys provide the strongest form of MFA protection and are strongly encouraged for all account root users and all IAM users in the Organization Management (root) account
- the Organization Management (root) account requires a dedicated Yubikey (if access is required to a sub-account root user, we do not want to expose the Organization Management accounts Yubikey)
- every ~50 sub-accounts requires a dedicated Yubikey (minimize the required number of Yubikeys and the scope of impact should a Yubikey be lost or compromised)
- each IAM breakglass user requires a dedicated Yubikey, as do any additional IAM users in the Organization Management (root) account. While some CSPs do not recommend MFA on the breakglass users, it is strongly encouraged in AWS
- all other AWS users (AWS SSO, IAM in sub-accounts, etc.) can leverage virtual MFA devices (like Google Authenticator on a mobile device)

7. Customers are responsible for the ongoing management and rotation of all passwords on a regular basis per their organizational password policy. This includes the passwords of all IAM users, MAD users, firewall users, or other users, whether deployed by the Accelerator or not. We do NOT automatically rotate any passwords, but strongly encourage customers do so, on a regular basis.
8. During the installation we request required limit increases, resources dependent on these limits will not be deployed

a. Limit increase requests are controlled through the Accelerator configuration file `"limits":{}` setting

b. The sample configuration file requests increases to your EIP count in the perimeter account and to the VPC count and Interface Endpoint count in the shared-network account

c. You should receive emails from support confirming the limit increases

d. On the next state machine execution, resources blocked by limits should be deployed (i.e. additional VPC's and Endpoints)

e. If more than 2 days elapses without the limits being increased, on the next state machine execution, they will be re-requested

9. Note: After a successful install the Control Tower `Organizational units'` dashboard will indicate `2 of 3` in the `Accounts enrolled` column for the Security OU, as it does not enable enrollment of the management account in guardrails. The Accelerator compliments Control Tower and enables guardrails in the management account which is important to high compliance customers.

1.8. Other Operational Considerations

- The Organization Management (root) account does NOT have any preventative controls to protect the integrity of the Accelerator codebase, deployed objects or guardrails. Do not delete, modify, or change anything in the Organization Management (root) account unless you are certain as to what you are doing. More specifically, do NOT delete, or change *any* buckets in the Organization Management (root) account.
- While generally protected, do not delete/update/change S3 buckets with cdk-asea-, or asea- in *any* sub-accounts.
- ALB automated deployments only supports Forward and not redirect rules.
- AWS generally discourages cross-account KMS key usage. As the Accelerator centralizes logs across an entire organization as a security best practice, this is an exception/example of a unique situation where cross-account KMS key access is required.
- Only 1 auto-deployed MAD in any mandatory-account is supported today.
- VPC Endpoints have no Name tags applied as CloudFormation does not currently support tagging VPC Endpoints.
- If the Organization Management (root) account coincidentally already has an ADC with the same domain name, we do not create/deploy a new ADC. You must manually create a new ADC (it won't cause issues).
- 3rd party firewall updates are to be performed using the firewall OS based update capabilities. To update the AMI using the Accelerator, you must first remove the firewalls and then redeploy them (as the EIP's will block a parallel deployment), or deploy a second parallel FW cluster and de-provision the first cluster when ready.
- When adding more than 100 accounts to an OU which uses shared VPC's, you must *first* increase the Quota `Participant accounts per VPC` in the shared VPC owner account (i.e. shared-network). Trapping this quota before the SM fails has been added to the backlog.
- The default limit for Directory Sharing is 125 accounts for an Enterprise Managed Active Directory (MAD), a quota increase needs to be manually requested through support from the account containing the MAD before this limit is reached. Standard MAD has a sharing limit of 5 accounts (and only supports a small quota increase). The MAD sharing limit is not available in the Service Quota's tools.

2.2.2 1. Accelerator Sample Configurations and Customization

1.1. Summary

- Sample config files can be found in this folder
- Most of the examples reflect a medium security profile (NIST, ITSG, FEDRAMP)
- Unsure where to start, use config.lite-CTNFW-example.json (CT w/NFW variant of option 2)
- Frugal and want something comprehensive to experiment with, use config.test-example.json (option 5)
- Config file schema documentation (Draft)
- Estimated monthly pricing for sample configurations

1.2. Sample Configuration Files with Descriptions

1.2.1. FULL CONFIGURATION (CONFIG.EXAMPLE.JSON)

- The full configuration file was based on feedback from customers moving into AWS at scale and at a rapid pace. Customers of this nature have indicated that they do not want to have to upsize their perimeter firewalls or add Interface endpoints as their developers start to use new AWS services. These are the two most expensive components of the deployed architecture solution.
- Default settings:
- AWS Control Tower: No
- Firewall: IPSec VPN with Active/Active Fortinet cluster (uses BGP+ECMP)

1.2.2. LITE WEIGHT CONFIGURATION FILES

- Four variants with differing central ingress/egress firewalls
- *Variant 1:* ***Recommended starting point*** (config.lite-CTNFW-example.json)
- Default Settings:
- AWS Control Tower: Yes
- Firewall: AWS Network Firewall
- *Variant 2:* ***Recommended for new GC PBMM customers*** (config.lite-VPN-example.json)
- requires 3rd party licensing (BYOL or PAYGO)
- Default Settings:
- AWS Control Tower: No
- Firewall: IPSec VPN with Active/Active Fortinet cluster (uses BGP+ECMP)
- *Variant 3:* (config.lite-NFW-example.json)
- Same as *Variant 1* config without AWS Control Tower
- Default Settings:
- AWS Control Tower: No
- Firewall: AWS Network Firewall
- *Variant 4:* (config.lite-GWLB-example.json)
- requires 3rd party licensing (BYOL or PAYGO)
- Default Settings:
- AWS Control Tower: No
- Firewall: Gateway Load Balancer with Checkpoint firewalls in an autoscaling group

- To reduce solution costs and allow customers to grow into more advanced AWS capabilities, we created these lite weight configurations that does not sacrifice functionality, but could limit performance. These config files:
- only deploys the 9 required centralized Interface Endpoints (removes 50 from full config). All services remain accessible using the AWS public endpoints, but require traversing the perimeter firewalls
- removes the perimeter VPC Interface Endpoints
- reduces the Fortigate instance sizes from c5n.2xl to c5n.xl (VM08 to VM04) in *Variant 2: IPSec VPN with Active/Active Fortinet cluster* option
- removes the Unclass ou and VPC
- AWS Control Tower can be implemented in all sample configs using *Variant 1: AWS Control Tower with AWS Network Firewall* as an example (new installs only).
- The Accelerator allows customers to easily add or change this functionality in future, as and when required without any impact

1.2.3. ULTRA-LITE SAMPLE CONFIGURATION

- *Variant 1:* (config.ultralite-CT-example.json)
- AWS Control Tower: Yes
- Firewall: None
- Networking: None
- *Variant 2:* (config.ultralite-example.json)
- AWS Control Tower: No
- Firewall: None
- Networking: None
- This configuration file was created to represent an extremely minimalistic Accelerator deployment, simply to demonstrate the art of the possible for an extremely simple config. This example is NOT recommended as it violates many AWS best practices. This config has:
- no `shared-network` or `perimeter` accounts
- no networking (VPC, TGW, ELB, SG, NACL, endpoints) or route53 (zones, resolvers) objects
- no Managed AD, AD Connector, rsyslog cluster, RDGW host, or 3rd party firewalls
- only enables/deploys AWS security services in 2 regions (ca-central-1, us-east-1) (Not recommended)
- only deploys 2 AWS config rules w/SSM remediation
- renamed log-archive (Logs), security (Audit) and operations (Ops) account names

1.2.4. MULTI-REGION SAMPLE CONFIGURATION (CONFIG.MULTI-REGION-EXAMPLE.JSON)

- This configuration file was created to represent a more advanced multi-region version of the Full configuration file from configuration 1 above. This config:
- adds a TGW in us-east-1, peered to the TGW in ca-central-1
- adds TGW static routes, including several dummy sample static routes
- adds a central Endpoint VPC in us-east-1 with us-east-1 endpoints configured
- adds a shared VPC for all UnClass OU accounts in us-east-1, connected to the us-east-1 TGW (accessible through ca-central-1)
- creates additional zones and resolver rules
- Sends us-east-1 CloudWatch Logs to the central S3 log-archive bucket in ca-central-1
- Deploys SSM documents to us-east-1 and remediates configured rules in UnClass OU
- adds a local account specific VPC, in us-east-1, in the account MyUnClass and connects it to the us-east-1 TGW (i.e. shares TGW)
- local account VPC set to use central endpoints, associates appropriate centralized hosted zones to VPC (also creates 5 local endpoints)
- adds a VGW for DirectConnect to the perimeter VPC
- adds the 3rd AZ in ca-central-1 (MAD & ADC in AZ a & b)

- Default Settings:
- AWS Control Tower: No
- Firewall: IPSec VPN with Active/Active Fortinet cluster (uses BGP+ECMP)

1.2.5. TEST CONFIGURATION (CONFIG.TEST-EXAMPLE.JSON) (USE FOR TESTING OR LOW SECURITY PROFILES)

- Further reduces solution costs, while demonstrating full solution functionality (NOT recommendend for production). This config file:
- uses the Lite weight configuration as the starting point (NFW variant)
- consolidates Dev/Test/Prod OU to a single Workloads OU/VPC
- only enables Security Hub, Config and Macie in ca-central-1 and us-east-1
- removes the Fortigate firewall cluster (per NFW variant)
- removes the rsyslog cluster
- reduces the RDGW instance sizes from t2.large to t2.medium
- reduces the size of the MAD from Enterprise to Standard edition
- removes the on-premise R53 resolvers (hybrid dns)
- reduced various log retention periods and the VPCFlow log interval
- removes the two example workload accounts
- adds AWS Network Firewall (NFW) and AWS NATGW for centralized ingress/egress (per NFW variant)
- Default Settings:
- AWS Control Tower: No
- Firewall: AWS Network Firewall

1.2.6. LITE WEIGHT MULTI-REGION CA-WEST-1 CONFIGURATION (CONFIG.LITE-VPN-MULTI-REGION-CA-WEST-1-EXAMPLE.JSON) FILES

- This configuration file was created to represent a more advanced multi-region version of the Full configuration file from configuration 1 above. This config:
- adds ca-west-1 to list of supported regions
- adds a TGW in ca-west-1
- adds a central Endpoint VPC in ca-west-1 with ca-west-1 endpoints configured
- adds a shared VPCs for Dev,Test,Prod,Unclass OU accounts in ca-west-1
- Sends ca-west-1 CloudWatch Logs to the central S3 log-archive bucket in ca-central-1
- Deploys SSM documents to ca-west-1 and remediates configured rules Dev,Test,Prod,Unclass OU
- adds VPC to Perimeter account in ca-west-1
- Deploys Fortigate Firewalls to Perimeter account in ca-west-1
- Disables Macie in ca-west-1 (Service not available yet)
- Deploys available AWS Config Rules to ca-west-1
- requires 3rd party licensing (BYOL or PAYGO)
- Default Settings:
- AWS Control Tower: No
- Firewall: IPSec VPN with Active/Active Fortinet cluster (uses BGP+ECMP)
- To reduce solution costs and allow customers to grow into more advanced AWS capabilities, we created these lite weight configurations that does not sacrifice functionality, but could limit performance. These config files:
- only deploys the 9 required centralized Interface Endpoints (removes 50 from full config). All services remain accessible using the AWS public endpoints, but require traversing the perimeter firewalls
- removes the perimeter VPC Interface Endpoints
- reduces the Fortigate instance sizes from c5n.2xl to c6i.xl (VM08 to VM04) in *Variant 2: IPSec VPN with Active/Active Fortinet cluster* option

- Review additional details here
- The Accelerator allows customers to easily add or change this functionality in future, as and when required without any impact

1.3. Deployment Customizations

1.3.1. MULTI-FILE CONFIG FILE AND YAML FORMATTING OPTION

- The sample configuration files are provided as single, all encompassing, json files. The Accelerator also supports both splitting the config file into multiple component files and configuration files built using YAML instead of json. Details can be found in the linked document.

1.3.2. SAMPLE SNIPPETS

- The sample configuration files do not include the full range of supported configuration file parameters and values, additional configuration file parameters and values can be found in the sample snippets document.

1.3.3. THIRD PARTY FIREWALL EXAMPLE CONFIGS

- The Accelerator is provided with a sample 3rd party configuration file to demonstrate automated deployment of 3rd party firewall technologies. Given the code is vendor agnostic, this process should be able to be leveraged to deploy other vendors firewall appliances. When and if other options become available, we will add them here as well.
- Automated firewall configuration customization possibilities
- Sample Fortinet Fortigate firewall config file

1.4. Other Configuration File Hints and Tips

- It is critical that all accounts that are leveraged by other accounts (i.e. accounts that any workload accounts are dependant on), are included in the mandatory-accounts section of the config file (i.e. shared-network, log-archive, operations)
- Account pointers within the config file point to the account key (i.e. (`mandatory-account-configs\account-key`) and NOT the account name field (`mandatory-account-configs\account-key\account-name: "account name"`). This allows for easy account names, duplicate account names, and no requirement to update account pointers during account renames.
- If any of the account pointers within `global-options` does not point to a valid mandatory account key, the State Machine will fail with the error `EnvironmentVariable value cannot be null` before starting CodeBuild Phase -1
- You cannot supply (or change) configuration file values to something not supported by the AWS platform
- For example, CWL retention only supports specific retention values (not any number)
- Shard count - can only increase/reduce by half the current limit. i.e. you can change from `1-2`, `2-3`, `4-6`
- Always add any new items to the END of all lists or sections in the config file, otherwise
- Update validation checks will fail (VPC's, subnets, share-to, etc.)
- To skip, remove or uninstall a component, you can often simply change the section header, instead of removing the section
- change "deployments"/"firewalls" to "deployments"/"xxfirewalls" and it will uninstall the firewalls and maintain the old config file settings for future use
- Objects with the parameter deploy: true, support setting the value to false to remove the deployment
- As you grow and add AWS accounts, the Kinesis Data stream in the log-archive account will need to be monitored and have its capacity (shard count) increased by setting `"kinesis-stream-shard-count"` variable under `"central-log-services"` in the config file
- Updates to NACL's requires changing the rule number (`100` to `101`) or they will fail to update
- When adding a new subnet or subnets to a VPC (including enabling an additional AZ), you need to:
- increment any impacted NACL id's in the config file (`100` to `101`, `32000` to `32001`) (CFN does not allow nacl updates)
- make a minor change to any impacted route table names (`MyRouteTable` to `MyRouteTable1`) (CFN does not allow updates to route table associated ids)
- The sample VPN firewall configuration uses an instance with **4** NIC's, make sure you use an instance size that supports 4 ENI's
- Firewall names, CGW names, TGW names, MAD Directory ID, account keys, and OU's must all be unique throughout the entire configuration file (also true for VPC names given NACL and security group referencing design)

- The configuration file *does* have validation checks in place that prevent users from making certain major unsupported configuration changes
- **The configuration file does *NOT* have extensive error checking. It is expected you know what you are doing. We eventually hope to offer a config file, wizard based GUI editor and add the validation logic in this separate tool. In most cases the State Machine will fail with an error, and you will simply need to troubleshoot, rectify and rerun the state machine.**
- You cannot move an account between top-level OU's. This would be a security violation and cause other issues. You can move accounts between sub-ou. Note: The Control Tower version of the Accelerator does NOT support sub-ou's.
- When using YAML configuration files, we only support the subset of yaml that converts to JSON (we do not support anchors)
- Security Group names were designed to be identical between environments, if you want the VPC name in the SG name, you need to do it manually in the config file
- Adding more than approximately 50 *new* VPC Interface Endpoints across *all* regions in any one account in any single state machine execution will cause the state machine to fail due to Route 53 throttling errors. If adding endpoints at scale, only deploy 1 region at a time. In this scenario, the stack(s) will fail to properly delete, also based on the throttling, and will require manual removal.
- We do not support Directory unsharing or ADC deletion, delete methods were not implemented. We only support ADC creation in mandatory accounts.
- If `use-central-endpoints` is changed from true to false, you cannot add a local VPC endpoint on the same state machine execution (add the endpoint on a prior or subsequent execution)
- If you update the 3rd party firewall names, be sure to update the routes and alb's which point to them. Firewall licensing occurs through the management port, which requires a VPC route back to the firewall to get internet access and validate the firewall license.
- Removing the AWS NFW requires 2 state machine executions, in the first you must remove all routes that reference the NFW, and in the second you can remove or xx out the NFW (also true for the GWLB implementation).

1.5. Config file and Deployment Protections

- The config file is moved to AWS CodeCommit after the first execution of the state machine to provide strong configuration history, versioning and change control
- After each successful state machine execution, we record the commit id of the config file used for that execution in secrets manager
- On ***every*** state machine execution, before making any changes, the Accelerator compares the latest version of the config file stored in CodeCommit with the version of the config file from the last successful state machine execution (after replacing all variables)
- If the config file includes any changes we consider to be significant or breaking, we immediately fail the state machine
- if a customer somehow accidentally uploads a different customers config file into their Accelerator CodeCommit repository, the state machine will fail
- if a customer makes what we consider to be a major change to the config file, the state machine will fail
- if a customer makes a change that we believe has a high likelihood to cause a deployment failure, the state machine will fail
- If a customer believes they understand the full implications of the changes they are making (and has made any required manual changes to allow successful execution), we have provided protection override flags. These overrides should be used with extremely caution:
- To provide maximum protection we have provided scoped override flags. Customers can provide a flag or flags to only bypass specific type(s) of config file validations or blocks. If using an override flag, we recommend customers use these scoped flags in most situations.
- If a customer is purposefully making extensive changes across the config file and wants to simply override all checks with a single override flag, we also have this option, but discourage it use.
- The various override flags and their format can be found in here.

1.6. Summary of Example Config File Minimum Changes for New Installs

At a minimum you should consider reviewing the following config file sections and make the required changes.

1.6.1. GLOBAL OPTIONS

- S3 Central Bucket
- `global-options/central-bucket` : "AWSDOC-EXAMPLE-BUCKET"
- replace with `your-bucket-name` as referenced in the Installation Guide Step #5

- Central Log Services SNS Emails
- `global-options/central-log-services/sns-subscription-emails` : "myemail+notifyT-xxx@example.com"
- update the 3 email addresses (high, medium and low) as required. Each address will receives alerts or alarms of the specified level. The same email address can be used for all three.
- The default dynamic CIDR pools (`global-options/cidr-pools`) listed below are used to assign ranges based on the subnet mask set in each VPC and subnet throughout the configuration file.
- `global-options/cidr-pools/0/cidr` : "10.0.0.0/13"
- The main address pool used to dynamically assign CIDR ranges for most VPCs
- `global-options/cidr-pools/1/cidr` : "100.96.252.0/23"
- Address pool used to dynamically assign CIDR ranges for the Managed Active Directory subnets in the Ops account
- `global-options/cidr-pools/2/cidr` : "100.96.250.0/23"
- Address pool used to dynamically assign CIDR ranges for the Perimeter VPC
- `global-options/cidr-pools/3/cidr` : "10.249.1.0/24"
- A non-routable pool of addresses used to dynamically assign CIDR ranges for the Active Directory Connector subnets in the Organization Management/root account

1.6.2. MANDATORY ACCOUNT CONFIGS

- All mandatory accounts specific to your config file, that are present under the `mandatory-account-config` section require you to assign a unique email address for each account listed below. Replace the email values in the JSON config file for these accounts with unique email addresses.
- `mandatory-account-configs/shared-network/email` : "myemail+aseaT-network@example.com----------REPLACE----------"
- `mandatory-account-configs/operations/email` : "myemail+aseaT-operations@example.com----------REPLACE----------"
- `mandatory-account-configs/perimeter/email` : "myemail+aseaT-perimeter@example.com----------REPLACE----------"
- `mandatory-account-configs/management/email` : "myemail+aseaT-management@example.com----------REPLACE----------" (Note: This is the email of your root account)
- `mandatory-account-configs/log-archive/email` : "myemail+aseaT-log@example.com----------REPLACE----------"
- `mandatory-account-configs/security/email` : "myemail+aseaT-sec@example.com----------REPLACE----------"
- Budget Alerts email addresses need to be replaced with an email address in your organization. It can be the same email address for all budget alerts. Config located at the following path (Multiple exist for different thresholds, update all under each account):
- `mandatory-account-configs/shared-network/budget/alerts/emails` : "myemail+aseaT-budg@example.com"
- `mandatory-account-configs/perimeter/budget/alerts/emails` : "myemail+aseaT-budg@example.com"
- `mandatory-account-configs/management/budget/alerts/emails` : "myemail+aseaT-budg@example.com"
- For the `shared-network` account, review and update the following (or delete the sections):
- `mandatory-account-configs/shared-network/vpc/on-premise-rules/zone` : "on-premise-privatedomain1.example.ca" (qty 2)
- `mandatory-account-configs/shared-network/vpc/zones/private` : "cloud-hosted-privatedomain.example.ca"
- `mandatory-account-configs/shared-network/vpc/zones/public` : "cloud-hosted-publicdomain.example.ca"
- For the `operations` account, review and update the following:
- `mandatory-account-configs/operations/deployments/mad/dns-domain` : "example.local"
- `mandatory-account-configs/operations/deployments/mad/netbios-domain` : "example"
- `mandatory-account-configs/operations/deployments/mad/log-group-name` : "/${ACCELERATOR_PREFIX_ND}/MAD/example.local" (replace example.local)
- `mandatory-account-configs/operations/deployments/mad/ad-users` (update user, email and group of each user as required)
- do not remove or change permissions on the `adconnector-usr`

- For `perimeter` account, review and update the following:
- `mandatory-account-configs/perimeter/certificates/priv-key` : "certs/example1-cert.key"
- `mandatory-account-configs/perimeter/certificates/cert` : "certs/example1-cert.crt"
- If you are using VPN config:
- `mandatory-account-configs/perimeter/deployments/firewalls/image-id` : "ami-0d8e2e78e928def11"
- Update AMI with the AMI collected from the Marketplace for **Fortinet FortiGate (BYOL) Next-Generation Firewall**
- `mandatory-account-configs/perimeter/deployments/xxfirewall-manager/image-id` : "ami-0e9f45c3ec34c3a9a"
- Update AMI with the AMI collected from the Marketplace for **Fortinet FortiManager (BYOL) Centralized Security Management**
- NOTE: Default config of "xxfirewall-manager" will prevent the firewall manager from being deployed. To deploy the firewall manager remove the "xx" to set the parameter to "firewall-manager"
- `mandatory-account-configs/perimeter/deployments/firewalls/license` : ["firewall/license1.lic", "firewall/license2.lic"]
- Two Fortinet FortiGate firewall licenses, if you don't have any license files, update the config file with an empty array ("license": []). Do NOT use the following: [""]
- Place files in a folder (eg. firewall) in the same S3 bucket in your Organization Management account as the deployment configuration file.
- `mandatory-account-configs/perimeter/deployments/firewalls/config` : "firewall/firewall-example.txt"
- The Fortinet configuration file to initially configure the firewalls. Sample configuration files can be found in the reference-artifacts/Third-Party folder
- Place file in a folder (eg. firewall) in the same S3 bucket in your Organization Management account as the deployment configuration file.
- If you are using GWLB config:
- `mandatory-account-configs/perimeter/deployments/firewalls/Checkpoint-Firewall - image-id` : "ami-0217611bf09d5b4c1"
- Update AMI with the AMI collected from the Marketplace for **CloudGuard Network Security for GWLB - BYOL**
- `mandatory-account-configs/perimeter/deployments/firewall-manager/image-id` : "ami-0071a3b4ef9ac766a"
- Update AMI with the AMI collected from the Marketplace for **Checkpoint Security Management**
- `mandatory-account-configs/perimeter/deployments/firewall-manager/version` : "R8110BYOLMGMT"
- Update version based on the selected ami-id version from the Private Marketplace
- For `management`, review and update the following:
- `mandatory-account-configs/management/account-name` : "ASEA-Main"
- Update this field with your Organization Management (root) account name, if it is not set to ASEA-Main.
- `mandatory-account-configs/management/iam/users`
- the names of your break-glass and ASEA operation users

1.6.3. WORKLOAD ACCOUNT CONFIGS

- As mentioned in the Installation Guide, we recommend not adding more than 1 or 2 workload accounts to the config file during the initial deployment as it will increase risks of hitting a limit. Once the Accelerator is successfully deployed, add the additional accounts back into the config file and rerun the state machine.
- Review the workload accounts in the config that you selected and change the name and email as desired
- Modify `mydevacct1` with the account name of your choosing
- Modify `mydevacct1/account-name` : "MyDev1" with the account name
- Modify `mydevacct1/email` : "myemail+aseaT-dev1@example.com----------REPLACE----------" with a unique email address for the account
- Modify `mydevacct1/description` : "This is an OPTIONAL SAMPLE workload account..." with a description relevant to your account
- Modify `mydevacct1/ou` : "Dev" with the OU that you would like the account to be attached to

1.6.4. ORGANIZATION UNITS

- For all organization units, update the budget alerts email addresses:
- `organizational-units/core/default-budgets/alerts/emails` : "myemail+aseaT-budg@example.com"
- `organizational-units/Central/default-budgets/alerts/emails` : "myemail+aseaT-budg@example.com"
- `organizational-units/Dev/default-budgets/alerts/emails` : "myemail+aseaT-budg@example.com"
- `organizational-units/Test/default-budgets/alerts/emails` : "myemail+aseaT-budg@example.com"
- `organizational-units/Prod/default-budgets/alerts/emails` : "myemail+aseaT-budg@example.com"
- `organizational-units/Sandbox/default-budgets/alerts/emails` : "myemail+aseaT-budg@example.com"
- For organization units with `certificates` , review the certificates and update as you see fit. These certificates are used in the `alb` section under `alb/cert-name` of each OU

2.2.3 1. CA-West-1 (Calgary) Region Configurations and Customizations

1.1. Introduction

1.1.1 SUMMARY

The configurations described in this documentation section explains how to enable the Calgary (ca-west-1) region. This currently depends on ASEA version > 1.6.1, and extends into ca-west-1 (i.e. ca-west-1 is NOT the home region). Before applying any of the configuration below, be sure to review the networking architecture, and deploy in a test ASEA instance first if possible.

1.1.2 ACTIVATING THE CALGARY OPT-IN REGION

Since March 20, 2019, when AWS adds a Region, the new Region is disabled by default. If you want your users to be able to create and manage resources in a new Region, you first need to enable that Region. The Calgary region (ca-west-1) is an 'Opt-in' region that requires enablement configuration for all AWS accounts.

To update the enabled Regions for member accounts of your AWS Organizations, perform the steps in the following procedure. 1. *Requires:* Enable trusted access for the AWS Account Management service. To set this up, see Enabling trusted access for AWS Account Management. 2. Sign in to the AWS Organizations console with your organization's management account credentials. 3. On the AWS accounts page, select the account that you want to update. 4. Choose the Account settings tab. 5. Under Regions, select the Region you want to enable or disable. 6. Choose Actions, and then choose either Enable or Disable option. 7. If you chose the Enable option, review the displayed text and then choose Enable region.

This can also be executed using the AWS CLI & SDKs, review this page for detail. Alternatively, you can also use the sample script provided here (insert hyperlink to reference artifacts) to enable or disable the Opt-in region programatically using the following instructions:

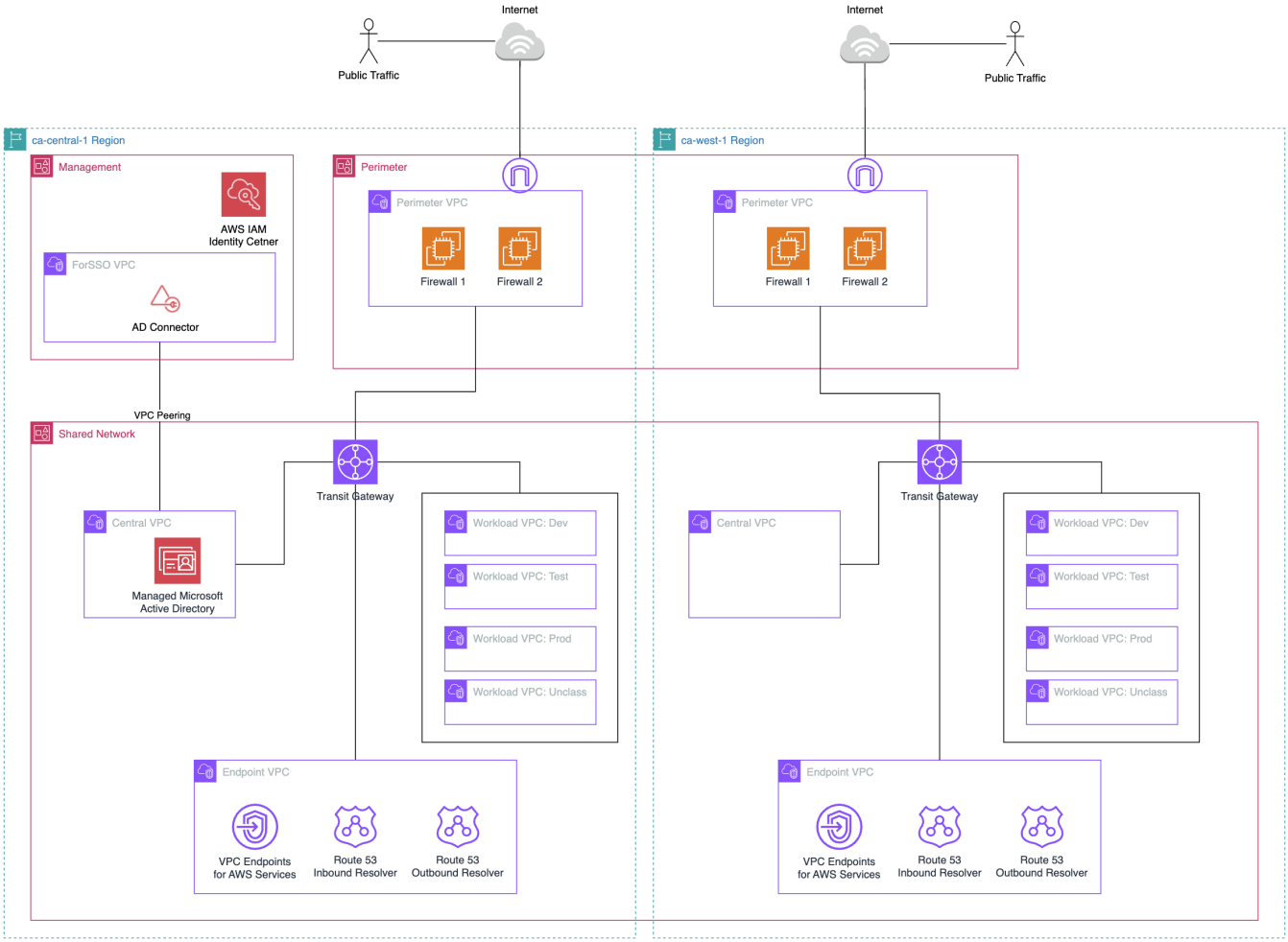
1. Log into the AWS console as a Full Administrator to the Organization Management account.
2. Start a CloudShell session.
3. Create a virtual python environment. `python3 -m venv env`
4. Activate the python environment. `source env/bin/activate`
5. Install the python3 required libaries (ex: `pip install -r requirements.txt`)
6. Make the Python script executable (ex: `chmod +x region_optin.py`)
7. Execute the script with the following parameters: `--OptInRegion` *region* `--Action` *enable / disable / status*

   Optional: `--IgnoreOU` *ou*

   Example: `python3 region_optin.py --OptInRegion ca-west-1 --Action=enable`

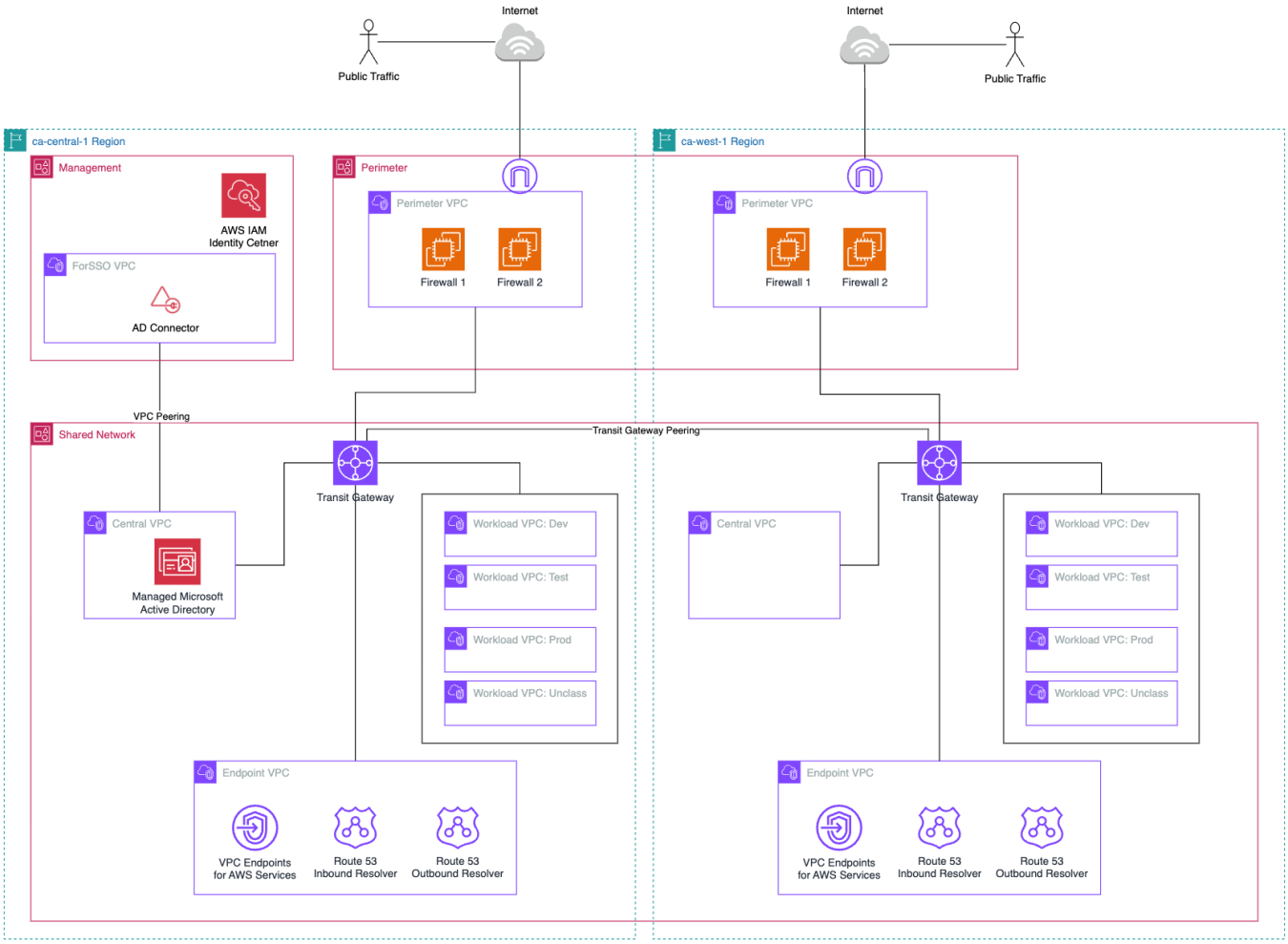
Note: These instructions will need to be repeated for all new accounts that are added in the future and that will be used for workloads that use the ca-west-1 region

1.2. Network Architecture -- Mirrored from Home Region



The *Mirrored from Home Region* network architecture mirrors the network architecture from the home region (e.g. ca-central-1). In the diagram above, ca-west-1 has its own Transit Gateway, same set of VPCs, Endpoint configuration, and Perimeter VPC/Firewall configuration. Additionaly, this configuration sample does not connect ca-central-1 with ca-west-1 via Transit Gateway Peering (see #1.3 below). Note that in the sample config provided, the IP CIDR ranges are different than the home region.

1.3. Network Architecture -- Cross Region Peering



The cross Region peering network architecture adds cross Region peering to enable cross Region communication. To continue following the Government of Canada Cloud guardrail "segment and separate" the workload VPCs would need individual segregated Transit Gateway Route Tables instead of the common Segregated route table to maintain segregation across Regions.

Region - ca-cental-1

Dev Segregated TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-central-1
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	blackhole - Dev VPC	ca-central-1
10.3.0.0/16	blackhole - Test VPC	ca-central-1
10.4.0.0/16	blackhole - Prod VPC	ca-central-1
10.5.0.0/16	blackhole - Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	Dev VPC	ca-west-1
10.99.0.0/16	blackhole - Test VPC	ca-west-1
10.100.0.0/16	blackhole - Prod VPC	ca-west-1
10.101.0.0/16	blackhole - Unclass VPC	ca-west-1

Test Segregated TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-central-1
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	blackhole - Dev VPC	ca-central-1
10.3.0.0/16	blackhole - Test VPC	ca-central-1
10.4.0.0/16	blackhole - Prod VPC	ca-central-1
10.5.0.0/16	blackhole - Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	blackhole - Dev VPC	ca-west-1
10.99.0.0/16	Test VPC	ca-west-1
10.100.0.0/16	blackhole - Prod VPC	ca-west-1
10.101.0.0/16	blackhole - Unclass VPC	ca-west-1

Prod Segregated TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-central-1
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	blackhole - Dev VPC	ca-central-1
10.3.0.0/16	blackhole - Test VPC	ca-central-1
10.4.0.0/16	blackhole - Prod VPC	ca-central-1
10.5.0.0/16	blackhole - Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	blackhole - Dev VPC	ca-west-1
10.99.0.0/16	blackhole - Test VPC	ca-west-1
10.100.0.0/16	Prod VPC	ca-west-1
10.101.0.0/16	blackhole - Unclass VPC	ca-west-1

Core TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-central-1
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	Dev VPC	ca-central-1
10.3.0.0/16	Test VPC	ca-central-1
10.4.0.0/16	Prod VPC	ca-central-1
10.5.0.0/16	Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
10.96.0.0/13	Peering TGW	ca-west-1

Shared TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-central-1
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	Dev VPC	ca-central-1
10.3.0.0/16	Test VPC	ca-central-1
10.4.0.0/16	Prod VPC	ca-central-1
10.5.0.0/16	Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
10.96.0.0/13	Peering TGW	ca-west-1

Region - ca-west-1

Dev Segregated TGW RT		
CIDR	Attachment ID	Region
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	Dev VPC	ca-central-1
10.3.0.0/16	blackhole - Test VPC	ca-central-1
10.4.0.0/16	blackhole - Prod VPC	ca-central-1
10.5.0.0/16	blackhole - Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
0.0.0.0/0	Perimeter VPN	ca-west-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	blackhole - Dev VPC	ca-west-1
10.99.0.0/16	blackhole - Test VPC	ca-west-1
10.100.0.0/16	blackhole - Prod VPC	ca-west-1
10.101.0.0/16	blackhole - Unclass VPC	ca-west-1

Test Segregated TGW RT		
CIDR	Attachment ID	Region
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	blackhole - Dev VPC	ca-central-1
10.3.0.0/16	Test VPC	ca-central-1
10.4.0.0/16	blackhole - Prod VPC	ca-central-1
10.5.0.0/16	blackhole - Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
0.0.0.0/0	Perimeter VPN	ca-west-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	blackhole - Dev VPC	ca-west-1
10.99.0.0/16	blackhole - Test VPC	ca-west-1
10.100.0.0/16	blackhole - Prod VPC	ca-west-1
10.101.0.0/16	blackhole - Unclass VPC	ca-west-1

Prod Segregated TGW RT		
CIDR	Attachment ID	Region
100.96.252.0/23	Central VPC	ca-central-1
10.1.0.0/16	Central VPC	ca-central-1
10.2.0.0/16	blackhole - Dev VPC	ca-central-1
10.3.0.0/16	blackhole - Test VPC	ca-central-1
10.4.0.0/16	Prod VPC	ca-central-1
10.5.0.0/16	blackhole - Unclass VPC	ca-central-1
10.0.0.0/22	Endpoint VPC	ca-central-1
0.0.0.0/0	Perimeter VPN	ca-west-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	blackhole - Dev VPC	ca-west-1
10.99.0.0/16	blackhole - Test VPC	ca-west-1
10.100.0.0/16	blackhole - Prod VPC	ca-west-1
10.101.0.0/16	blackhole - Unclass VPC	ca-west-1

Core TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	Dev VPC	ca-west-1
10.99.0.0/16	Test VPC	ca-west-1
10.100.0.0/16	Prod VPC	ca-west-1
10.101.0.0/16	Unclass VPC	ca-west-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.0.0.0/13	Peering TGW	ca-central-1

Shared TGW RT		
CIDR	Attachment ID	Region
0.0.0.0/0	Perimeter VPN	ca-west-1
10.97.0.0/16	Central VPC	ca-west-1
10.98.0.0/16	Dev VPC	ca-west-1
10.99.0.0/16	Test VPC	ca-west-1
10.100.0.0/16	Prod VPC	ca-west-1
10.101.0.0/16	Unclass VPC	ca-west-1
10.96.0.0/22	Endpoint VPC	ca-west-1
10.0.0.0/13	Peering TGW	ca-central-1

The sample segregated route tables have routes to shared resources in the central and endpoint VPCs of each region and only to the corresponding workload VPC in the remote region. Internet bound traffic would route to the local Region Perimeter firewalls. For example, lets look at the dev workload VPC and what it's allowed to route to based on the sample config.

```
dev (ca-central-1) <--> dev (ca-west-1)
dev (ca-central-1 and ca-west-1) <--> central (ca-central-1 and ca-west-1)
dev (ca-central-1 and ca-west-1) <--> endpoint (ca-central-1 and ca-west-1)
dev (ca-central-1) <--> perimeter (ca-central-1)
dev (ca-west-1) <--> perimeter (ca-west-1)
```

1.4. How to apply Mirrored from Home Region configuration

The general strategy is to compare your existing deployed configuration (**config.json** in CodeCommit) with the sample provided here. Using your preferred file compare tool (e.g. Visual Studio Code), you will see differences that need to be applied. Here is a list of changes that should be made: 1. Add the use of '${ALT_REGION}' 2. Set '${ALT_REGION}': 'ca-west-1' 3. Add 'ca-west-1' to list of Supported Regions 4. Add 'ca-west-1' to list of Macie Excluded Regions (until service is launched in region) 5. 'fw-mgr-alert-level': 'None' 6. Add 'ca-west-1' to additional-cwl-regions 7. Add '${ALT_REGION}' to list of ssm-automation regions (global and OU config sections) 8. Add '${ALT_REGION}' cidr-pools 9. Add TGW for '${ALT_REGION}' 10. Add firewalls (Fortinet) to deploy in '${ALT_REGION}' 1. Follow ASEA installation instructions for Marketplace and enabling the Fortigate Subscriptions in ca-west-1 11. AWS Config configuration is split into supported region rules. Remediate-regions updated with '${ALT_REGION}' 12. Endpoint VPC created in Shared-Network account in '${ALT_REGION}'. Note available Interface Endpoints is a subset of ca-central-1. Sample deploys minimum needed. 13. Dev/Test/Prod VPCs created in Shared-Network account in '${ALT_REGION}' with TGW attachments

Current Known Limitations:

1. Managed Active Directory should be manually 'shared' to ca-west-1 once the service is updated to support ca-west-1
2. Rsyslog servers (used as an option for Fortigate logging destination) can only be deployed to a single region. This would need to be configured outside ASEA (manually or with your own created IaC).
3. Fortigate firewalls config use c6i EC2 instance types in lieu of c5n until it becomes available in ca-west-1.

1.5. How to apply Cross Region Peering configuration

The general strategy is to compare your existing deployed configuration (**config.json** in CodeCommit) with the sample provided here. Using your preferred file compare tool (e.g. Visual Studio Code), you will see differences that need to be applied. Here is a list of changes that should be made:

1.5.1 Create Segregated Route Tables and Propogations

In preparation for the Transit Gateway peering, you need to create a segregated route table for each workload VPC in each Region. This allows you the flexibility to customize the routes specific to each workload VPC which is used to only allow routing to the corresponding workload VPC in the remote Region. You also need to propagate the routes from the Endpoint VPC, Central VPC, and Firewall attachments to maintain communication to these locations. 1. Create workload segregated Transit Gateway route tables by adding them to the home Region Transit Gateway `["mandatory-account-configs"]["shared-network"].deployments.tgw[0]["route-tables"]` and remote Region Transit Gateway `["mandatory-account-configs"]["shared-network"].deployments.tgw[1]["route-tables"]` sections.

```
"route-tables": [
  "core",
  "shared",
  "standalone",
  "segregated",
  "dev_segregated",
  "test_segregated",
  "prod_segregated",
  "unclass_segregated"
],
```

2. Add the workload segregated Transit Gateway route tables to the `tgw-rt-propagated` section under `tgw-attach` for the Endpoint VPCs, Central VPCs, and the Firewalls Transit Gateway attachments in Perimeters of each Region.

```
"tgw-rt-propagate": [
  "core",
  "shared",
  "standalone",
  "segregated",
  "dev_segregated",
  "test_segregated",
  "prod_segregated",
  "unclass_segregated"
],
```

3. Commit the changes and run the `ASEA-MainStateMachine_sm` State Machine (SM) with the input of `{"scope": "FULL", "mode": "APPLY", "verbose": "0"}`. Wait for successful completion. 4. Verify the new TGW route tables are created and have the routes to central, endpoint and firewall tgw attachments.

1.5.2 Associate Workload VPC to Workload Segregated Transit Gateway Route Table

This process will switch the workload VPC from the segregated TGW route table to the workload specific segregated TGW route table.

NOTE: Following this process will isolate the respective resources in the workload VPC. Any communication within the VPC will be unaffected however any communication that has to transfer through the Transit Gateway will be interrupted. Recommend performing this process on one workload VPC at a time during a maintenance window. For example, only start with the Dev VPC.

1. Undeploy the TGW attachment by prefixing the `tgw-attach` with "xx" to be `xxtgw-attach` to the corresponding workload VPC. This will be an unknown field, which is the same a deleting the section.

```
"xxtgw-attach": {
  "associate-to-tgw": "Main",
  "account": "shared-network",
  "associate-type": "ATTACH",
  "tgw-rt-associate": ["segregated"],
  "tgw-rt-propagate": ["core", "shared"],
  "blackhole-route": true,
  "attach-subnets": ["TGW"],
  "options": ["DNS-support"]
}
```

2. Commit the changes and run the `ASEA-MainStateMachine_sm` State Machine (SM) with the input of `{"scope": "FULL", "mode": "APPLY", "verbose": "0"}`. Wait for successful completion.
3. Redeploy the TGW attachment by removing the "xx" to be `tgw-attach` and update the `tgw-rt-associate` with the respective workload segregated TGW route table. For example changing from `segregated` to `dev_segregated`.

```
"tgw-attach": {
  "associate-to-tgw": "Main",
  "account": "shared-network",
  "associate-type": "ATTACH",
  "tgw-rt-associate": ["dev_segregated"],
  "tgw-rt-propagate": ["core", "shared"],
  "blackhole-route": true,
  "attach-subnets": ["TGW"],
  "options": ["DNS-support"]
}
```

4. Commit the changes and run the `ASEA-MainStateMachine_sm` State Machine (SM) with the input of `{"scope": "FULL", "mode": "APPLY", "verbose": "0"}`. Wait for successful completion.
5. Validate communication has been restored to original status.
6. Repeat steps 1-5 for each workload VPC.

1.5.3 Configure Transit Gateway Peering

The Transit Gateway peering process is achieved by creating a TGW peering attachment and creating static routes in each of the TGW route tables.
1. Create the Transit Gateway peering attachment by adding the following section to the remote Region TGW deployment section to associate to TGW in home Region. `json`

```
    "tgw-attach": {
        "associate-to-tgw": "Main",
        "account": "shared-network",
        "region": "${HOME_REGION}",
        "tgw-rt-associate-local": ["core"],
        "tgw-rt-associate-remote": ["core"]
    },
```

2. Commit the changes and run the `ASEA-MainStateMachine_sm` State Machine (SM) with the input of `{"scope": "FULL", "mode": "APPLY", "verbose": "0"}`. Wait for successful completion. 3. Create static routes for each of the TGW route tables in each Region. You are creating these

routes to allow workload traffic to its workload VPC peer, Central VPC and Endpoint VPC in remote Region. Refer to the `Segregated Route Tables` above and the sample multi-region config file for examples here. This is an example of the static routes in the `dev_segregated` TGW route table in the home Region assuming CIDR ranges follows example above.

```
{
  "name": "dev_segregated",
  "routes": [
    {
      "destination": "10.3.0.0/16",
      "blackhole-route": true
    },
    {
      "destination": "10.4.0.0/16",
      "blackhole-route": true
    },
    {
      "destination": "10.5.0.0/16",
      "blackhole-route": true
    },
    {
      "destination": "10.96.0.0/22",
      "target-tgw": "Main-${ALT_REGION}"
    },
    {
      "destination": "10.97.0.0/16",
      "target-tgw": "Main-${ALT_REGION}"
    },
    {
      "destination": "10.98.0.0/16",
      "target-tgw": "Main-${ALT_REGION}"
    },
    {
      "destination": "10.99.0.0/16",
      "blackhole-route": true
    },
    {
      "destination": "10.100.0.0/16",
      "blackhole-route": true
    }
  ]
},
```

4. Commit the changes and run the `ASEA-MainStateMachine_sm` State Machine (SM) with the input of `{"scope": "FULL", "mode": "APPLY", "verbose": "0"}`. Wait for successful completion. 5. Validate communication across the TGW peering connection between Regions.

2.2.4 1. State Machine Behavior and Inputs

1.1. State Machine Behavior

Accelerator v1.3.0 makes a significant change to the manner in which the state machine operates. These changes include:

1. Reducing the `default scope` of execution of the state machine to only target newly created AWS accounts and AWS accounts listed in the mandatory accounts section of the config file.

- `default scope` refers to running the state machine without any input parameters;
- This new default scope disallows any changes to the config file outside new accounts;
- NOTE: it is critical that accounts for which others are dependant upon, MUST be located within the `mandatory-account-configs` section of the config file (i.e. management, log-archive, security, operations, shared-network, perimeter, etc.).

2. The state machine now accepts a new input parameter, `scope`, which accepts the following values: `FULL` | `NEW-ACCOUNTS` | `GLOBAL-OPTIONS` | `ACCOUNT` | `OU`.

- when the `scope` parameter is supplied, you must also supply the `mode` parameter. At this time `mode` only accepts the value `APPLY`. To be specific `"mode":"APPLY"` is mandatory when running the state machine with the `"scope":` parameter.

3. Starting the state machine with `{"scope":"FULL","mode":"APPLY"}` makes the state machine execute as it did in v1.2.6 and below.

- The state machine targets all AWS accounts and allows changes across any section of the config file;
- The blocks and overrides described in section 1.4 above remain valid;
- `FULL` mode must be run at least once immediately after any Accelerator version upgrade. Code Pipeline automatically starts the state machine with `{"scope":"FULL","mode":"APPLY"}`. If the state machine fails for any reason after upgrade, the state machine must be restarted with these parameters until a successful execution of the state machine has completed.

4. Starting the state machine with `{"scope":"NEW-ACCOUNTS","mode":"APPLY"}` is the same as operating the state machine with the `default scope` as described in the first bullet

5. Starting the state machine with `{"scope":"GLOBAL-OPTIONS","mode":"APPLY"}` restricts changes to the config file to the `global-options` section.

- If any other portion of the config file was updated or changed, the state machine will fail;
- The global options scope executes the state machine on the entire managed account footprint.

6. Starting the state machine with `{"scope":"OU","targetOus":[X],"mode":"APPLY"}` restricts changes to the config file to the specified `organizational-units` section(s) defined by `targetOus`.

- When `scope=OU`, `targetOus` becomes a mandatory parameter;
- `X` can be any one or more valid OU names, or the value `"ALL"`;
- When `["ALL"]` is specified, the state machine targets all AWS accounts, but only allows changes to the `organizational-units` section of the config file;
- When OUs are specified (i.e. `["Dev","Test"]`), the state machine only targets mandatory accounts plus accounts in the specified OUs (Dev, Test), and only allows changes to the specified OUs sections (Dev, Test) of the config file;
- If any other portion of the config file was updated or changed, the state machine will fail.

7. Starting the state machine with `{"scope":"ACCOUNT","targetAccounts":[X],"mode":"APPLY"}` restricts changes to the config file to the specified `xxx-account-configs` section(s) defined by `targetAccounts`.

- When `scope=ACCOUNT`, `targetAccounts` becomes a mandatory parameter;
- `X` can be any one or more valid account numbers, the value `"NEW"`, or the value `"ALL"`;
- When `["ALL"]` is specified, the state machine targets all AWS accounts, but only allows changes to the `xxx-account-configs` sections of the config file;
- When specific accounts and/or `NEW` is specified (i.e. `["NEW", "123456789012", "234567890123"]`), the state machine only targets mandatory accounts plus the listed accounts and any newly created accounts. It also only allows changes to the specified accounts sections (New, 123456789012, 234567890123) of the config file;
- If any other portion of the config file was updated or changed, the state machine will fail.

Starting in v1.3.0, we recommend running the state machine with the parameters that most tightly scope the state machines execution to your planned changes and minimizing the use of `FULL` scope execution.

- should you accidentally change the wrong section of the config file, you will be protected;
- as you grow and scale to hundreds or thousands of accounts, your state machine execution time will remain fast.

NOTE 1: The `scope` setting has no impact on SCP application, limit requests, custom tagging, or directory sharing.

NOTE 2: All comparisons for config file changes are assessed AFTER all replacements have been made. Changing variable names which result in the same end outcome do NOT appear as a change to the config file.

1.2. Accelerator State Machine Inputs

1.2.1. REBUILD DYNAMODB TABLE CONTENTS

With the exception of the Outputs table, the contents of the Accelerator DynamoDB tables are rebuilt on every state machine execution. We recently started depending on the Outputs DynamoDB tables to ensure the parameters in parameter store are consistently maintained in the same order as objects are created and deleted. Should the CONTENTS of the tables be destroyed or corrupted, customers can force a rebuild of the CloudFormation Outputs in DynamoDB by starting the state machine with the parameter:

```
{ "storeAllOutputs": true }
```

This should be completed BEFORE running the state machine with a corrupt or empty DynamoDB table or the Accelerator is likely to reorder a customers parameters. If the DynamoDB tables were completely destroyed, they must be recreated before running the state machine with this parameter.

1.2.2. BYPASS ALL CONFIG FILE VALIDATION CHECKS

This parameter should be specified with extreme caution, as it bypasses all config file validation. The state machine typically has protections enabled preventing customers from making breaking changes to the config file. Under certain conditions with the support of a trained expert, bypassing these checks is required. Start the state machine with the parameter:

```
{ "overrideComparison": true }
```

Customers are encouraged to use the specific override variables below, rather than the all-inclusive override, to ensure they only bypasses intended config changes.

1.2.3. BYPASSING SPECIFIC CONFIG FILE VALIDATION CHECKS

Providing any one or more of the following flags will only override the specified check(s):

```
{
  "configOverrides": {
    "ov-global-options": true,
    "ov-del-accts": true,
    "ov-ren-accts": true,
    "ov-acct-email": true,
    "ov-acct-ou": true,
    "ov-acct-vpc": true,
    "ov-acct-subnet": true,
    "ov-acct-vpc-optin": true,
    "ov-tgw": true,
    "ov-mad": true,
    "ov-ou-vpc": true,
    "ov-ou-subnet": true,
    "ov-share-to-ou": true,
    "ov-share-to-accounts": true,
    "ov-nacl": true,
    "ov-nfw": true
  }
}
```

1.2.4. GENERATE VERBOSE LOGGING WITHIN STATE MACHINE

- Added "verbose": "1" state machine input options
- parameter is optional

- parameter defaults to 0

```
{ "scope": "FULL", "mode": "APPLY", "verbose": "1" }
```

1.2.5. STATE MACHINE SCOPING INPUTS

Summary of inputs, per section 1.1 above:

```
{ "scope": "FULL", "mode": "APPLY" }
```

```
{ "scope": "NEW-ACCOUNTS", "mode": "APPLY" }
```

```
{ "scope": "GLOBAL-OPTIONS", "mode": "APPLY" }
```

```
{ "scope": "OU", "targetOus": ["ou-name", "ou-name"], "mode": "APPLY" }
```

```
{ "scope": "ACCOUNT", "targetAccounts": ["123456789012", "234567890123"], "mode": "APPLY" }
```

1.2.6. EXAMPLE OF COMBINED INPUTS

```
{
  "scope": "FULL",
  "mode": "APPLY",
  "configOverrides": { "ov-ou-vpc": true, "ov-ou-subnet": true, "ov-acct-vpc": true }
}
```

2.2.5 1. Multi-file Accelerator Config file and YAML Support Details

1.1. Customers would like the ability to specify their configuration in YAML. This facilitates

- commenting out entire sections, which is unavailable in standard JSON
- annotating aspects of configuration (e.g. cidr: "10.100.0.0/16" # We chose this for \$reason.)
- aligning the Accelerator with CloudFormation, which supports JSON/YAML as input format

1.2. Customers would like the configuration file split into multiple files

- one file for Global options + Mandatory accounts
- one file per OU
- one file for every approx. 2000 lines of workload accounts (Code Commit diff stops working at 3000 lines, allow for adding to each file)

1.3. Benefits

1. Easier cut/paste/comparison of OU configurations
2. Allow CodeCommit diff functionality to function (File currently too large)
3. Allow easier updates to workload accounts (simple append)
4. Smaller scoped updates (de-risk accidentally changing the wrong section)
5. Both a customer request and something the team thought was a good idea

1.4. Steps FOR YAML

- The loadAcceleratorConfig functionality should no longer assume config.json as the config filename in the config repo and/or S3, instead it should look for config.yaml and config.json
- Check for the existence of config.yaml and config.json (initially in S3, but also in CodeCommit on future executions)
- If both files exist, fail with an error message
- **Infer the file type from the extension, and parse accordingly**
- Any other failure should also be an error, fail with an error message
- The accelerator will continue to use JSON formatting internally, if a yaml file is supplied, we are simply converting it to JSON for use by the Accelerator
- All examples throughout this document use config.json as the example, but also apply to YAML
- Both JSON and YAML input files will be equally supported
- Only one file format is supported across all config files, either JSON or YAML, customers can NOT mix YAML and JSON file formats

1.5. Steps For File Split

- When the `__LOAD` keyword is encountered, search relatively (from the same location as root config file) for the file, and insert into the config tree, recursively following `__LOAD` if necessary (to max depth of 2). Any file referenced in `__LOAD` must parse successfully in one of the two formats, otherwise FAIL.

```
{
   "core": {
      "__LOAD": "ous/core.json"
   }
}
```

Note that while we will provide sensible examples, there is no prescriptive requirement for file organization within a customer's configuration, customers can use the feature to break-out sections as is most effective for their deployment. Breaking out large repeatable sections like security groups is a good example and could be included off the main file, an account file, or off an ou file:

```
"security-groups": [ "__LOAD": "global/security-groups.json" ]
```

Examples:

1. All in one (single file like today):

```
.
├── config.json
```

1. Split along major sections:

```
.
├── config.json
├── ous
│ ├── core.json
│ ├── dev.json
│ └── test.json ---> could be one per ou, could only be for some ou's as determined by customer
├── accounts
│ ├── workload-accounts-group1.json
│ ├── workload-accounts-group2.json
│ ├── my-workload-accounts.json
│ └── more-accounts.json ---->we will encourage each file being as close to 2000 lines as possible (not one per account, not all in one file)
└── global
├── global-options.json
├── security-groups.json

etc
```

- Max depth of 2 means config.json can load ou/dev.json, which can load global/security-groups.json.
- security-groups.json CANNOT load another sub-file (unless security-groups.json was only directly loaded from config.json).

1.6. Dealing with Accelerator Automatic Config File Updates

When customers create AWS accounts directly through AWS Organizations, the Accelerator automatically updates the config file, adding these new accounts. If a customer renames an OU we automatically update the config file. With multi-part files, how do we know what source file to update? We require two mechanisms:

1. Add the following new parameters to the global-options section of the config file

```
    "workloadaccounts-param-filename": "accounts/more-accounts2.json",
    "workloadaccounts-prefix" : "accounts/more-accounts",
    "workloadaccounts-suffix" : 3,
```

- filename is set to `config.json`, and prefix to `config` in a single file configuration scenario (suffix is not used)
- While OU contents can be moved into `__LOADED` sub-files, it was decided the OU object itself must remain in the main config file
- The above parameters:
- are required to be in the main config file and cannot be `__LOAD`'ed
- Must be present or SM fails
- Are used to decide where to add new accounts to the config file

2. Add the following new parameter to each mandatory and workload account config

```
    "src-filename": "accounts/my-workload-accounts.json",
```

1.7. Accelerator Internal Operations

- when updating an account in the config file, we use the `"src-filename"` parameters to find and update an accounts `ou`, `ou-path`, `account-name`, and `email` parameters

- When creating new accounts (inserting into config file):
- if the update is not going to make the file larger than 2000 lines, insert the new account into the config file `"workloadaccounts-param-filename"`
- if the insert will push the file over 2000 lines:
- create the next unused filename for the given prefix in Code Commit (`{"workloadaccounts-prefix"}{"workloadaccounts-suffix"}.{customer file format}`), i.e. `"accounts/more-accounts3.json"`
- insert the new account into the new file in it's entirety
- update `"workloadaccounts-param-filename"` to: `{"workloadaccounts-prefix"}{"workloadaccounts-suffix"}.{customer file format}`
- add a new load stmt to the workload-accounts section of the config file with the name `{"workloadaccounts-prefix"}{"workloadaccounts-suffix"}.{customer file format}`
- update `"workloadaccounts-suffix"` to: `{"workloadaccounts-suffix"} + 1`
- be careful with comma's between files (JSON sections) when appending/connecting

1.8. Example

The entire main config file could be reduced to this:

```
{
   "global-options": {
      "workloadaccounts-param-filename": "accounts/more-accounts2.json",
      "workloadaccounts-prefix": "accounts/more-accounts",
      "workloadaccounts-suffix": 3,
      "__LOAD": "global/global-options.json"
   },
   "mandatory-account-configs": {
      "__LOAD": "accounts/mandatory-accounts.json"
   },
   "workload-account-configs": {
      "__LOAD": [
         "accounts/workload-accounts1.json",
         "accounts/my-other-accounts.json",
         "accounts/workload-accounts2.json"
      ]
   },
   "organizational-units": {
      "core": {
         "__LOAD": "ous/core.json"
      },
      "Central": {
         "__LOAD": "ous/central.json"
      },
      "Dev": {
         "__LOAD": "ous/dev.json"
      },
      "Test": {
         "__LOAD": "ous/test.json"
      },
      "Prod": {
         "__LOAD": "ous/prod.json"
      },
      "UnClass": {
         "__LOAD": "ous/unclass.json"
      },
      "Sandbox": {
         "__LOAD": "ous/sandbox.json"
      }
   }
}
```

1.9. Acceptance Criteria

- A new customer may start an Accelerator deployment with a config.json or config.yaml, and have it deploy as expected so long as the file is semantically correct according to structure and expected keys (and of course syntactically correct in either YAML or JSON)
- Accelerator should continue to function as it does today o i.e. on startup creates repo and copies all referenced config files, not just config.json to repo (json or YAML) o leverages config files in CodeCommit repo from this point forward (json or YAML as provided by customer) o SM runs against the commit id of each file at the start of the SM (i.e. don't allow changes to any file during execution)
- Accelerator leverages multiple config files to receive the same input parameters it previously did from one file
- All accelerator functionality both ALZ and Standalone versions continue to function as previously defined

• Customer can successfully provides multiple config files with the same result as the current one file

2.2.6 1. Existing Organizations / Accounts

1.1. Considerations: Importing existing AWS Accounts / Deploying Into Existing AWS Organizations

- The Accelerator *can* be installed into existing AWS Organizations
- our early adopters have all successfully deployed into existing organizations
- Existing AWS accounts *can* also be imported into an Accelerator managed Organization
- Caveats:
- Per AWS Best Practices, the Accelerator deletes the default VPC's in all AWS accounts, worldwide. The inability to delete default VPC's in pre-existing accounts will fail the installation/account import process. Ensure default VPC's can or are deleted before importing existing accounts. On failure, either rectify the situation, or remove the account from Accelerator management and rerun the state machine
- The Accelerator will NOT alter existing (legacy) constructs (e.g. VPC's, EBS volumes, etc.). For imported and pre-existing accounts, objects the Accelerator prevents from being created using preventative guardrails will continue to exist and not conform to the prescriptive security guidance
- Existing workloads should be migrated to Accelerator managed VPC's and legacy VPC's deleted to gain the full governance benefits of the Accelerator (centralized flow logging, centralized ingress/egress, no IGW's, Session Manager access, existing non-encrypted EBS volumes, etc.)
- Existing AWS services will be reconfigured as defined in the Accelerator configuration file (overwriting existing settings)
- We do NOT support *any* workloads running or users operating in the Organization Management (root) AWS account. The Organization Management (root) AWS account MUST be tightly controlled
- Importing existing *workload* accounts is fully supported, we do NOT support, recommend and strongly discourage importing mandatory accounts, unless they were clean/empty accounts. Mandatory accounts are critical to ensuring governance across the entire solution
- We've tried to ensure all customer deployments are smooth. Given the breadth and depth of the AWS service offerings and the flexibility in the available deployment options, there may be scenarios that cause deployments into existing Organizations to initially fail. In these situations, simply rectify the conflict and re-run the state machine.
- If the Firewall Manager administrative account is already set for your organization, it needs to be unset before starting a deployment.

1.2. Process to import existing AWS accounts into an Accelerator managed Organization

- Newly invited AWS accounts in an Organization will land in the root ou
- Unlike newly created AWS accounts which immediately have a Deny-All SCP applied, imported accounts are not locked down as we do not want to break existing workloads (these account are already running without Accelerator guardrails)
- In AWS Organizations, select ALL the newly invited AWS accounts and move them all (preferably at once) to the correct destination OU (assuming the same OU for all accounts)
- In case you need to move accounts to multiple OU's we have added a 2 minute delay before triggering the State Machine
- Any accounts moved after the 2 minute window will NOT be properly ingested, and will need to be ingested on a subsequent State Machine Execution
- This will first trigger an automated update to the config file and then trigger the state machine after a 2 minute delay, automatically importing the moved accounts into the Accelerator per the destination OU configuration
- As previously documented, accounts CANNOT be moved between OU's to maintain compliance, so select the proper top-level OU with care
- If you need to customize each of the accounts configurations, you can manually update the configuration file either before or after you move the account to the correct ou
- if before, you also need to include the standard 4 account config file parameters, if after, you can simply add your new custom parameters to the account entry the Accelerator creates
- if you add your imported accounts to the config file, moving the first account to the correct ou will trigger the state machine after a 2 minutes delay. If you don't move all accounts to their correct ou's within 2 minutes, your state machine will fail. Simply finish moving all accounts to their correct OU's and then rerun the state machine.

- If additional accounts are moved into OUs while the state machine is executing, they will not trigger another state machine execution, those accounts will only be ingested on the next execution of the state machine
- customers can either manually initiate the state machine once the current execution completes, or, the currently running state machine can be stopped and restarted to capture all changes at once
- Are you unsure if an account had its guardrails applied? The message sent to the state machine Status SNS topic (and corresponding email address) on a successful state machine execution provides a list of all successfully processed accounts.
- The state machine is both highly parallel and highly resilient, stopping the state machine should not have any negative impact. Importing 1 or 10 accounts generally takes about the same amount of time for the Accelerator to process, so it may be worth stopping the current execution and rerunning to capture all changes in a single execution.
- We have added a 2 min delay before triggering the state machine, allowing customers to make multiple changes within a short timeframe and have them all captured automatically in the same state machine execution.

1.3. Deploying the Accelerator into an existing Organization

- As stated above, if the ALZ was previously deployed into the Organization, please work with your AWS account team to find the best mechanism to uninstall the ALZ solution
- Ensure all existing sub-accounts have the role name defined in `organization-admin-role` installed and set to trust the Organization Management (root) AWS Organization account
- prior to v1.2.5, this role must be named: `AWSCloudFormationStackSetExecutionRole`
- if using the default role (`AWSCloudFormationStackSetExecutionRole`) we have provided a CloudFormation stack which can be executed in each sub-account to simplify this process
- As stated above, we recommend starting with new AWS accounts for the mandatory functions (shared-network, perimeter, security, log-archive accounts).
- To better ensure a clean initial deployment, we also recommend the installation be completed while ignoring most of your existing AWS sub-accounts, importing them post installation:
- create a new OU (i.e. `Imported-Accounts`), placing most of the existing accounts into this OU temporarily, and adding this OU name to the `global-options\ignored-ous` config parameter;
- any remaining accounts must be in the correct ou, per the Accelerator config file;
- install the Accelerator;
- import the skipped accounts into the Accelerator using the above import process, paying attention to the below notes
- NOTES:
- Do NOT move any accounts from any `ignored-ous` to the root ou, they will immediately be quarantined with a Deny-All SCP, they need to be moved directly to their destination ou
- As stated above, when importing accounts, there may be situations we are not able to fully handle
- If doing a mass import, we suggest you take a quick look and if the solution is not immediately obvious, move the account which caused the failure back to ignored-ous and continue importing the remainder of your accounts. Once you have the majority imported, you can circle back and import outstanding problem accounts with the ability to focus on each individual issue
- The challenge could be as simple as someone has instances running in a default VPC, which may require some cleanup effort before we can import (coming soon, you will be able to exclude single account/region combinations from default VPC deletion to gain the benefits of the rest of the guardrails while you migrate workloads out of the default VPC)

2.2.7 1. How to migrate an AWS Landing Zone (ALZ) account "as is" into an AWS Secure Environment Accelerator (ASEA)

1.1. Overview

This document describes the steps to migrate an existing linked account from an AWS Landing Zone (ALZ) to an AWS Secure Environment Accelerator (ASEA).

1.2. Prerequisites / Setup

1.2.1. CONFIRM ASEA SSO AND OU CONFIGURATION

On the ASEA, setup and run initial tests with SSO and permission sets with an account under the OU where the linked account will be migrated to. Confirm that SSO is properly configured with permissions required for the team members whose account is being migrated. This would include configuration of the ASEA's AWS Managed Active Directory (MAD) which should align with how the team migrating their account has their AWS SSO and MAD configured today.

1.2.2. SWITCH THE ALZ LINKED ACCOUNT PAYMENT METHOD TO INVOICING

If working with your AWS account team (TAM/SA) they will reach out to an internal team within AWS to have the linked account payment method switched to invoicing. This way the customer doesn't have to enter a credit card when making the account standalone in the upcoming steps.

1.2.3. CONFIRM CONSOLE ACCESS TO THE ALZ LINKED ACCOUNT AND ALSO TO THE EMAIL ACCOUNT

Confirm you have access to login as root to the ALZ linked account AWS console. Confirm you have access to the email account associated to the ALZ linked account. The upcoming steps will first make the account standalone (remove from ALZ organizations) so you need to make sure you have root access to the account. If required, you can reset the password following: https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_change-root.html

1.2.4. IF AN ENTERPRISE SUPPORT (ES) CUSTOMER, THEN CONFIRM ES IS ENABLED ON THE ALZ LINKED ACCOUNT

If the ALZ management account is on Enterprise Support (ES), then make sure ES is enabled on the linked account being migrated to the ASEA. If its not, then raise a support case to activate ES on the linked account. This is to make sure an ES support case can be created and escalated during step 2 if any unforeseen issue occurs.

1.2.5. CONFIRM THE ALZ CODEPIPELINE IS EXECUTING SUCCESSFULLY

Make sure the ALZ CodePipeline is still running successfully. Execute the ALZ CodePipeline from the management account to make sure it runs successfully.

- AWS Console -> CodePipeline
- Select “AWS-Landing-Zone-CodePipeline”
- Select “Release Change”
- Click on the pipeline and confirm it successfully runs through to completion

1.2.6. CONFIRM CLI ACCESS AND SETUP PYTHON AND THE AWS PYTHON SDK (BOTO3)

Confirm SSO temporary command line access from the management account with AdminAccess.

- SSO login → Select linked account → “Command line or programmatic access”
- Select Option 2 and add to your AWS credentials file under “[default]“
- This is required as the python script in step 3 takes a “profile” parameter
- Confirm you have the AWS CLI tool installed.
- https://aws.amazon.com/cli/
- Confirm by running a command such as “aws s3 ls”
- Confirm you have python3 and the AWS python library (boto3) installed which is required in step 2 to confirm the account has been disassociated from the landing zone correctly.
- https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html

1.3. Landing Zone - Disassociate the account from the ALZ

- Login to the ALZ management account, and go to “Service Catalog” -> “Provisioned products”
- Select “Access Filter” -> “Account” to see a list of the account products

1.3.1. SELECT THE PRODUCT FOR THE SPECIFIC LINKED ACCOUNT

- Put the linked account name in the provisioned products search bar
- This will narrow down the list and show a product name “AWS-Landing-Zone-Account-Vending-Machine” with a name *“lzapplicaitons*”*
- Select that product and then “Actions->Terminate”

1.3.2. CONFIRM THE PRODUCT SUCCESSFULLY TERMINATES

- The provisioned product entry will show a status of “Under change”
- You can also verify by going to CloudFormation→Stacks and you will see “DELETE IN PROGRESS” for the AVM Template stack being deleted.
- Go to the Resources tab to see the deleted resources associated to this stack.
- Once the provisioned product no longer says “Under change” move to the next step.
- Please note, this can take 1-2 hours.

1.3.3. GO TO THE LINKED ACCOUNT (ASSUME ROLE)

- From the management account, assume the role “AWSCloudFormationStackSetExecutionRole” to the linked account
- or optionally, SSO with console access to that account

1.3.4. UNDER “CLOUDFORMATION” VERIFY THAT THE ALZ STACKS (STACKSETS FROM ALZ MGMT) WERE DELETED

- There should be no stack left in the linked account with the prefix “StackSet-AWS-Landing-Zone-Baseline*". For example:
- StackSet-AWS-Landing-Zone-Baseline-CentralizedLoggingSpoke-
- StackSet-AWS-Landing-Zone-Baseline-EnableConfigRules-
- StackSet-AWS-Landing-Zone-Baseline-EnableNotifications-
- StackSet-AWS-Landing-Zone-Baseline-EnableConfigRulesGlobal-
- StackSet-AWS-Landing-Zone-Baseline-EnableConfig-
- StackSet-AWS-Landing-Zone-Baseline-ConfigRole-
- StackSet-AWS-Landing-Zone-Baseline-IamPasswordPolicy-
- StackSet-AWS-Landing-Zone-Baseline-SecurityRoles-
- StackSet-AWS-Landing-Zone-Baseline-EnableCloudTrail-

1.3.5. VERIFY THAT THE ACCOUNT IS READY TO BE INVITED AND BASELINED BY THE ASEA

- You need to ensure that resources don’t exist in the default VPC, there is no config recorder channel, no CloudTrail Trail and STS is active in all regions.
- This can be done manually, but ideally use this python script that can be run as well to automate the verification
- https://github.com/paulbayer/Inventory_Scripts/blob/mainline/ALZ_CheckAccount.py
- mkdir test; cd test
- git clone https://github.com/paulbayer/Inventory_Scripts.git
- python3 ALZ_CheckAccount.py -a LINKED ACCOUNT_HERE -p default

- It will run through 5 steps and output the following. If you were to run this script before the “terminate” step above is complete you would have warnings in steps 2 and 3 below.
- Step 0 completed without issues
- Checking account 111122223333 for default VPCs in any region
- Step 1 completed with no issues
- Checking account 111122223333 for a Config Recorders and Delivery Channels in any region
- Step 2 completed with no issues
- Checking account 111122223333 for a specially named CloudTrail in all regions
- Step 3 completed with no issues
- Checking account 111122223333 for any GuardDuty invites
- Step 4 completed with no issues
- Checking that the account is part of the AWS Organization.
- Step 5 completed with no issues
- We've found NO issues that would hinder the adoption of this account ****

1.4. Landing Zone (ALZ) - Remove the account from the ALZ organizations and make standalone

Removing the account from the ALZ organizations and making it standalone is required so it can be invited into the ASEA organization.

1.4.1. READ THE FOLLOWING SUMMARY/CONSIDERATIONS

- https://aws.amazon.com/premiumsupport/knowledge-center/organizations-move-accounts/

1.4.2. VERIFY ACCESS

- As stated in the previous sections, verify you have a mechanism to access the account post leaving the ALZ organization
- Former SSO roles will no longer function nor will the “AWSCloudFormationStackSetExecutionRole” role as it will have a trust relationship to the ALZ management account.
- Confirm the root credentials have been recovered and are usable
- As an alternative, confirm access with a new role/IAM user with Admin permissions on the account

1.4.3. VERIFY BILLING FLIPPED TO INVOICING

- As stated in the previous sections, verify the account payment method has been flipped to “invoicing” to avoid having to enter a Credit Card when going standalone. This can be done working with your AWS account team who will coordinate internally, or by raising a support case describing the use case.

1.4.4. REMOVE THE ACCOUNT FROM THE ORGANIZATIONS AND MAKE STANDALONE

- Follow the instructions on the following link to remove the account
- The short version is select the account from the ALZ mgmt account Organizations and select "remove"
- https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts_remove.html
- https://aws.amazon.com/blogs/security/aws-organizations-now-supports-self-service-removal-of-accounts-from-an-organization
- Note, when moving the account standalone do not select Enterprise Support. You shouldn't get a popup dialog asking for a Credit Card and the Support level since the account should have been moved to invoicing. Support can be reenabled on the linked account once it’s invited into the ASEA organization.

1.5. Accelerator - Invite the account into its organization

1.5.1. FROM THE ASEA MGMT ACCOUNT, SEND AN INVITE TO THE STANDALONE ACCOUNT

- Follow the instructions on the following link to invite the account

- The short version is go to the ASEA mgmt account organizations and select "Add an account" -> "Invite existing account" -> "enter the linked account account ID"
- https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts_invites.html

1.5.2. IN THE FORMER ALZ ACCOUNT, ACCEPT THE INVITATION

- https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_accounts_invites.html#orgs_manage_accounts_accept-decline-invite

1.5.3. KEEP THE LINKED ACCOUNT AT THE ROOT LEVEL OF THE ORGANIZATIONS

- Verify access to the linked account using your root login credentials
- If you had created an IAM role/user with Admin permissions, then verify access as well

1.5.4. ACTIVATE ENTERPRISE SUPPORT (ES) ON THIS LINKED ACCOUNT

- If ES is enabled on the ASEA management account, open a support case to enable ES on this linked account
- Go to the Support center and create a billing support case with "Account" and "Activation"
- Subject "Requesting ES enablement on linked account"
- Body "Requesting ES enablement on linked account "
- Your AWS TAM can escalate the case with the support team if it's time sensitive.
- This is to make sure an ES support case can be created and escalated during the next steps if any unforeseen issue occurs.

1.5.5. UPDATE (OR ADD) THE ORGANIZATION ADMING ROLE SO ONE CAN ASSUME THE ROLE INTO THE LINKED ACCOUNT

- Login to the linked account which just joined the organization.
- Create a new Organization Admin role, as defined in the customers config file: "organization-admin-role": "OrganizationAccountAccessRole".
- With newer customers the default is "OrganizationAccountAccessRole, with older customers it is "AWSCloudFormationStackSetExecutionRole".
- If "AWSCloudFormationStackSetExecutionRole" then you can edit the trust relationship directly
- Go to IAM -> Role -> AWSCloudFormationStackSetExecutionRole
- Update the trust relationship to have the management account ID of the ASEA (instead of the account ID of the previous ALZ)
- Verify that you can assume this role from the management account into the linked account

1.6. Accelerator - Move the linked account from the top level root OU into the appropriate OU managed by the ASEA

1.6.1. PLAN WHAT OU THIS ACCOUNT WILL BE MOVED INTO

- Option 1 - Create a new OU and move the account into that OU
- Before the migration, the team would have created a new OU (ie-similar to the sandbox OU).
- This would be needed if they need to isolate this account from TGW attachments/Networking and want to keep it isolated.
- The state machine will run and start to baseline the account.
- It will create a new VPC and deploy resources using CFN such as Config, CloudTrail, etc.
- Note, if the OU is setup similar to the sandbox OU it does not provide access to the shared VPCs that have the TGW attachments.
- Creating a new OU also requires adding that new OU and the OU persona to the config file in advance of the next state machine execution.
- Option 2 - Move account into an existing OU (ie-prod)
- The state machine will run and start to baseline the account.
- It will create a new VPC and deploy resources using CFN such as Config, CloudTrail, etc.
- The customers existing VPC will remain, as a 2nd DETACHED VPC.
- Mote. if it is non-compliant to security rules, it remains non-compliant and needs to be cleaned up and brought into compliance
- If the VPC is compliant and it has unique IP addresses, it could be attached to the TGW.

1.6.2. MOVE THE ACCOUNT FROM THE ROOT OU TO THE CORRECT OU

- THIS CANNOT BE EASILY UNDONE - MAKE SURE YOU MOVE TO THE CORRECT OU

- Follow the instructions on the following link to move the account to the correct OU
- The short version is go to the ASEA management account organizations and "select the account" -> "actions" -> "move" -> "select the correct OU"
- NOTE: The ASEA state machine will automatically start within 1-2 minutes of the account being moved into the OU
- https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_ous.html#move_account_to_ou
- Verify that the ASEA main state machine (under AWS->Step Functions) is triggered and runs cleanly (~30-45 minutes)

1.7. Accelerator (ASEA) - Verify access control with roles, SSO, etc

- Update and verify SSO and permission sets for the linked account now part of the ASEA
- Verify you still have access to the linked account via root (or other mechanisms)
- Verify you still can assume the operations role into the linked account

1.8. Landing Zone - Close down the ALZ core accounts and then the management account

Once all workloads have been migrated from the ALZ to the ASEA, then you may decide to shutdown your ALZ.

1.8.1. CLOSE DOWN THE ALZ LINKED ACCOUNTS

- Close all the linked accounts “as is” without making them standalone
- This will be the ALZ core linked accounts, but you might have some remaining workload accounts you decided not to migrate to the ASEA.
- https://aws.amazon.com/premiumsupport/knowledge-center/close-aws-account
- The management account will remain with organizations and the core accounts will show as suspended for 90 days.

1.8.2. CLOSE DOWN THE ALZ MANAGEMENT ACCOUNT

- After 90 days, the suspended linked accounts will be completely closed
- Go to the root account and turn off Organizations and then close the root account

2.3 Upgrades

2.3.1 1. Accelerator Upgrade Guide

1.1. General Upgrade Considerations

- Due to some breaking dependency issues, customers can only upgrade to v1.3.8 or above (older releases continue to function, but cannot be installed).
- While an upgrade path is planned, customers with a standalone Accelerator installation can upgrade to v1.5.x but need to continue with a standalone installation until the Control Tower upgrade option becomes available.
- Always compare your configuration file with the config file from the release you are upgrading to in order to validate new or changed parameters or changes in parameter types / formats.
- do NOT update to the latest firewall AMI - see the last bullet in section 1.8. Other Operational Considerations of the installation guide
- do NOT update the `organization-admin-role` - see item 2 in section 1.3.7. Other
- do NOT update account-keys (i.e. existing installations cannot change the internal values to `management` from `master`)
- do NOT make changes outside those required for the upgrade (those stated in the release notes or found through the comparison with the sample config file(s)). Customers wishing to change existing Accelerator configuration should either do so before their upgrade, ensuring a clean/ successful state machine execution, or after a successful upgrade.
- The Accelerator name and prefix ***CANNOT*** be changed after the initial installation
- Customers which customized any of the Accelerator provided default configuration files (SCPs, rsyslog config, ssm-documents, iam-policies, etc.) must manually merge the latest Accelerator provided updates with deployed customizations:
- it is important customers assess the new defaults and integrate them into their custom configuration, or Accelerator functionality could break or Accelerator deployed features may be unprotected from modification
- if customers don't take action, we continue to utilize the deployed customized files (without the latest updates)
- The below release specific considerations need to be cumulatively applied (an upgrade from v1.2.3 to v1.2.5 requires you to follow both v1.2.4 and v1.2.5 considerations)

1.2. Release Specific Upgrade Considerations:

- Upgrades to `v1.5.6-a and above` from `v1.5.5 and below`:
- In order to implement the VPC flow log fix (#1112) (b5dc19c):
- Before update: for every VPC of the configuration, change the “flow-logs” option to “CWL”
- Execute the State Machine using `{"scope": "FULL","mode": "APPLY"}`. Wait for successful completion
- Change the “flow-logs” option to the original value (“BOTH”) (don’t re-run the state machine)
- Follow the general instructions to upgrade ASEA
- Upgrades to `v1.5.1-a and above` from `v1.5.0` or `v1.5.1`:
- Do not add the parameter: `"ssm-inventory-collection": true` to OUs or accounts which already have SSM Inventory configured or the state machine will fail
- Follow the standard upgrade steps detailed in section 1.3 below
- `v1.5.1` was replaced by v1.5.1-a and is no longer supported for new installs or upgrades

- Upgrades to `v1.5.0` and `v1.5.1-a and above` from `v1.3.8 through v1.3.9`:
- We recommend upgrading directly to v1.5.1-a
- Due to the size and complexity of this upgrade, we require all customers to upgrade to `v1.3.8 or above` before beginning this upgrade
- While v1.5.0 supports Control Tower for *NEW* installs, existing Accelerator customers *CANNOT* add Control Tower to their existing installations at this time (planned enhancement for 22H1)
- Attempts to install Control Tower on top of the Accelerator will corrupt your environment (both Control Tower and the Accelerator need minor enhancements to enable)
- **The v1.5.x custom upgrade guide can be found here**
- Upgrades to `v1.3.9 and above` from `v1.3.8-b and below`:
- All interface endpoints containing a period must be removed from the config.json file either before or during the upgrade process
- i.e. ecr.dkr, ecr.api, transfer.server, sagemaker.api, sagemaker.runtime in the full config.json example
- If you remove them on a pre-upgrade State Machine execution, you can put them back during the upgrade, if you remove them during the upgrade, you can put them back post upgrade.
- Upgrades to `v1.3.3 and above` from `v1.3.2 and below`:
- Requires mandatory config file schema changes as documented in the release notes.
- These updates cause the config file change validation to fail and require running the state machine with the following input to override the validation checks on impacted fields: `{"scope": "FULL", "mode": "APPLY", "configOverrides": {"ov-ou-vpc": true, "ov-ou-subnet": true, "ov-acct-vpc": true }}`
- Tightens VPC interface endpoint security group permissions and enables customization. If you use VPC interface endpoints that requires ports/protocols other than TCP/443 (such as email-smtp), you must customize your config file as described here
- Upgrades from `v1.3.0 and below`:
- Please review the `Release Specific Upgrade Considerations` from ASEA v1.5.0 or below, they were removed from this release.

1.3. Summary of Upgrade Steps (all versions except v1.5.0)

1. Login to your Organization Management (root) AWS account with administrative privileges
2. Either: a) Ensure a valid Github token is stored in secrets manager (per the installation guide), or b) Ensure the latest release is in a valid branch of CodeCommit in the Organization Management account
3. Review and implement any relevant tasks noted in the General Upgrade Considerations section above
4. Update the config file in CodeCommit with new parameters and updated parameter types based on the version you are upgrading to (this is important as features are iterating rapidly)

- An automated script is available to help convert config files to the new v1.5.0 format
- Compare your running config file with the sample config file from the latest release
- Review the `Config file changes` section of the release notes for **all** Accelerator versions since your current deployed release

5. If you customized any of the other Accelerator default config files by overriding them in your S3 input bucket, merge the latest defaults with your customizations before beginning your upgrade
6. Download the latest installer template (`AcceleratorInstallerXYZ.template.json` or `AcceleratorInstallerXXX-CodeCommit.template.json`) from the `Assets` section of the latest release
7. Do ***NOT*** accidentally select the `ASEA-InitialSetup` CloudFormation stack **below**
8. If you are replacing your GitHub Token:

- Take note of the `AcceleratorName`, `AcceleratorPrefix`, `ConfigS3Bucket` and `NotificationEmail` values from the Parameters tab of your deployed Installer CloudFormation stack (`ASEA-what-you-provided`)
- Delete the Installer CloudFormation stack (`ASEA-what-you-provided`)
- Redeploy the Installer CloudFormation stack using the template downloaded in step 6, providing the values you just documented (changes to `AcceleratorName` or `AcceleratorPrefix` are not supported)
- The pipeline will automatically run and trigger the upgraded state machine

9. If you are using a pre-existing GitHub token, or installing from CodeCommit:

- Update the Installer CloudFormation stack using the template downloaded in step 5, updating the `GithubBranch` to the latest release (eg. `release/v1.5.1-a`)
- Go to AWS CloudFormation and select the stack: `ASEA-what-you-provided`
- Select Update, select Replace current template, Select Upload a template file
- Select Choose File and select the template you downloaded in step 6 (`AcceleratorInstallerXYZ.template.json` or `AcceleratorInstallerXXX-CodeCommit.template.json`)
- Select Next, Update `GithubBranch` parameter to `release/vX.Y.Z` where X.Y.Z represents the latest release
- Click Next, Next, I acknowledge, Update
- Wait for the CloudFormation stack to update (`Update_Complete` status) (Requires manual refresh)
- Go To Code Pipeline and Release the ASEA-InstallerPipeline

2.3.2 1. Accelerator v1.5.x Custom Upgrade Instructions

1.1. Overview

The upgrade from v1.3.8/v1.3.9 to v1.5.x is generally the same as any previous Accelerator upgrades, with a couple of key differences:

- the magnitude of this release has resulted in a requirement for significant updates to the config file
- we have provided a script to *assist* with this process. A manual verification of the changes and customer custom updates are often still required.
- we are re-aligning the OU structure with AWS guidance and that of AWS Control Tower (optional, but highly recommended)
- the core OU is being split into a "Security" OU and an "Infrastructure" OU
- we've added the capability to manage your IP addresses in DynamoDB, rather than with the config file
- this includes the ability to dynamically allocate CIDR ranges to VPCs and subnets
- more information on this features design can be found on this ticket
- the config file conversion script will:
- update your config file in a manner that supports both CIDR management schemes (but continues to leverage the previous mechanism)
- copy your currently configured CIDR ranges into the appropriate DynamoDB tables (optional, but recommended)
- you can change your IP address mechanism for any VPC at any time
- customers can mix and match IP address management mechanisms as they choose (`provided` , `lookup` , and `dynamic`)

1.2. Upgrade Caveats

1. **While an upgrade path is planned, customers with a Standalone Accelerator installation can upgrade to v1.5.x but need to continue with a Standalone installation until the Control Tower upgrade option becomes available.**
2. The script to assist with config file conversion and DynamoDB population only supports single file json based config files, customers that leverage YAML and/or multi-part config files, have several options:

- manually update your yaml or multi-part json config file to reflect the config file format for the latest release (similar to all previous upgrades)
- use the config.json file found in the `raw` folder of your CodeCommit repo to run the conversion script
- this version of the config file has resolved all variables with their final values, all variables will be removed from config.json in this scenario
- the new config file can be converted back to json/multi-part format before being placed back into your CodeCommit repository
- or it could be used to simply validate the changes you made using option a
- do not manually update the config file in the `raw` folder, as it will be overwritten based on the json or yaml file in the root of your repository
- use a 3rd party tool to manually convert your yaml / multi-part config files to a single file json file to run the conversion script
- the new config file can be converted back to json/multi-part format before being placed back into your CodeCommit repository

3. Config files which are significantly different than the example config files may not be properly converted. This includes config files which use different mandatory account keys or renamed the core OU.
4. This guide and its examples assume the existing accelerator deployment uses the `PBMMAccel-` accelerator prefix, if a different prefix is used on the existing installation, it is important it is specified when execution section 1.6 below.

1.3. Config File Conversion

- You must first upgrade to Accelerator v1.3.8 or v1.3.9
- Login to your AWS Organization Management account
- Pull your current config.json file from CodeCommit and save as a text file
- Locate the python conversion script and review its readme here
- To convert your configuration file execute: (completely offline process)

```
python update.py --Region ca-central-1 --LoadConfig --ConfigFile config.json
```

- This will output a new config file named: `update-config.json`
- Save *both* the original v13.8 and the new v1.5.0 config files for future reference/use
- After conversion, we recommend running the updated config file back prettier to simplify file comparisons
- While the conversion script often does much of the heavy lifting, we still require customers to manually verify the changes and make manual adjustments as appropriate:
- If you use a relatively standard config file you MAY not need to make any changes manually
- Ensure the value of `account-name` for the Organization Management account matches the actual account name of the Organization management account (the account key is generally either `management` or `master`).
- we recommend you change your `rdgw-instance-type` and `rsyslog-instance-type` from t2.*to t3.* (they will auto-replace on the next instance refresh) (Optional).
- optionally remove the `"API_GW_EXECUTION_LOGGING_ENABLED"` config rule throughout, as it overlaps with an identical Security Hub config rule.
- we added the capability to deploy a Config aggregator in any of the central services accounts (i.e. Log-archive, Security, Operations), by adding `"config-aggr": true` to *either*: `central-security-services`, `central-operations-services`, or `central-log-services`. The existing aggregator in the Org management account will remain. Do **not** set it in all 3 sections, as AWS only supports a maximum of 3 config aggregators.
- the optional attribute `endpoint-port-orverides` has been properly renamed to `endpoint-port-overrides`. If you have the `endpoint-port-orverides` in your config file you must rename it to `endpoint-port-overrides`.
- the new example config files also introduced several new internally resolvable variables (`${CONFIG::OU_NAME}` and `${CONFIG::VPC_NAME}`), which when used thoughtfully along with the new dynamic CIDR feature, enables multi-part config file customers to define the VPCs for multiple OU's in a single shared nested config file. These new variables should be ignored during an upgrade.
- the accelerator supports 3 types of CIDR ranges `provided`, `lookup`, and `dynamic`. The upgrade script sets the `cidr-src` to `provided`, meaning it uses the CIDR ranges provided in the config file, as per the previous release. The upgrade script also adds the additional required fields (`pool` and `size`) to every CIDR range defined in the config file to leverage the `lookup` type, but when set to `provided` these fields are NOT required and could be removed. They were added by the script for the sole purpose of making it easy to switch from `provided` to `lookup` in future. Once a customer switches to `lookup`, the `cidr\value` field is no longer used and can be removed from the config file. The `cidr-src` for should remain set at `provided` during upgrade.
- do **not** add the `cidr-pools` section to the config file during or before the upgrade, this section is only used for new installations.
- New description fields have been added to the config file to help provide context to certain objects. These will be used by a future GUI that is under development, and serve no functional purpose at this time. Customers can alter this text as they please.
- Most of the example config files have been converted to `dynamic` cidr-src as it provides simplier CIDR management for new customers. Two example config files ending in `-oldIP.json` have been maintained to aid upgrading customers in config file comparison.
- Be advised - in v1.5.0 we restructured the SCPs based on a) customer requests, and b) the addition of Control Tower support for new installs.
- customers are responsible for reviewing the SCPs to ensure they have not been altered in a manner that no longer meets an organizations security requirements;
- we reorganized and optimized our SCP's from 4 SCP files down to 3 SCP files, without removing any protections or guardrails;
- these optimizations have resulted in minor enhancements to the SCP protections and in some cases better scoping;
- the first two SCP files (Part-0 and Part-1) contain the controls which protect the integrity of the Accelerator itself;
- the third file (Sensitive, Unclass, Sandbox) contains customer data protection specific guardrails, which may change based on workload data classification or customer profiles and requirements;
- this frees the fourth SCP for use by Control Tower, or for use by customers for custom guardrails (Standalone installs only). As Control Tower leverages 2 SCP files on the Security OU, we have moved some of our SCP's to the account level (Control Tower installations only).
- The script and upgrade instructions above do not include the new config file parameters added in v1.5.1+. These new parameters can be added either during or after the upgrade. New parameters include: `"rdgw-enforce-imdsv2": true`, `"rsyslog-enforce-imdsv2": true`, `"ssm-inventory-collection": true` on each ou, and `"dynamic-s3-log-partitioning": [{values}]`

1.4. Upgrade process

- Before proceeding with your upgrade please review the General and Release Specific Upgrade Considerations in the Upgrade Guide
- upgrades directly from v1.3.8 need to ensure they include the extra step required for v1.3.9 upgrades (removal of endpoints with periods)
- Login to your AWS Organization Management account, in your home or default region
- Place your *updated and validated* config file back in the root folder of your CodeCommit repository
- json, yaml and multi-part config files continue to be supported
- Change to the us-east-1 region and open CloudWatch
- Select Rules under events in the left navigation pane
- Select the `PBMMAccel-MoveAccount_rule`, select `actions`, select `Disable`
- Select the `PBMMAccel-PolicyChanges_rule`, select `actions`, select `Disable`
- Open AWS Organizations
- Select the `core` OU and rename it to `Security` (case sensitive) (i.e. Actions, Rename)
- In the `Root` ou, create a new OU named `Infrastructure (case sensitive) (i.e. Actions, Create new)
- Expand the `Security` OU, select all EXCEPT the `security`, `log-archive`, and Organization `Management` account
- (i.e. the `Operations`, `Perimeter`, and `SharedNetwork` accounts)
- Select Actions, Move, then select the newly created `Infrastructure` OU (note these accounts temporarily missing guardrails)
- NOTE: the key to this being a possible move/change, is the new `Infrastructure` OU is defined identically to the old `core` OU. Typically you CANNOT move accounts between OU's without breaking something and violating security guardrails.
- Select Policies from the left navigation pane, then Service COntrol Policies
- Click the `PBMMAccel-Guardrails-Part-2` policy, and Select Targets
- Detach the policy from ALL OUs
- Change to the us-east-1 region and open CloudWatch
- Select Rules under events in the left navigation pane
- Select the `PBMMAccel-MoveAccount_rule`, select `actions`, select `Enable`
- Select the `PBMMAccel-PolicyChanges_rule`, select `actions`, select `Enable`
- Follow the Standard Upgrade instructions from the section `Summary of Upgrade Steps (all versions)` of the Installation and Upgrade guide, repeated verbatim below for ease of reference

1.5. "Summary of Upgrade Steps (all versions)" *(Copied from upgrade guide)*

1. Login to your Organization Management (root) AWS account with administrative privileges
2. Either:

   a) Ensure a valid Github token is stored in secrets manager, or

   b) Ensure the latest release is in a valid branch of CodeCommit in the Organization Management account. See this (section) of the installation guide for more details.
3. Review and implement any relevant tasks noted in the upgrade consideration sections (sections 1.1 and 1.2) of the Upgrade Guide
4. Update the config file in CodeCommit with new parameters and updated parameter types based on the version you are upgrading to (this is important as features are iterating rapidly)

- An automated script is available to help convert config files to the new v1.5.0 format
- Compare your running config file with the sample config file from the latest release
- Review the `Config file changes` section of the release notes for **all** Accelerator versions since your current deployed release

5. If you customized any of the other Accelerator default config files by overriding them in your S3 input bucket, merge the latest defaults with your customizations before beginning your upgrade
6. Download the latest installer template (`AcceleratorInstallerXYZ.template.json` or `AcceleratorInstallerXXX-CodeCommit.template.json`) from the `Assets` section of the latest release

7. Do ***NOT*** accidentally select the `PBMMAccel-InitialSetup` CloudFormation stack **below**
8. If you are replacing your GitHub Token:

- Take note of the `AcceleratorName`, `AcceleratorPrefix`, `ConfigS3Bucket` and `NotificationEmail` values from the Parameters tab of your deployed Installer CloudFormation stack (`PBMMAccel-what-you-provided`)
- Delete the Installer CloudFormation stack (`PBMMAccel-what-you-provided`)
- Redeploy the Installer CloudFormation stack using the template downloaded in step 6, providing the values you just documented (changes to `AcceleratorName` or `AcceleratorPrefix` are not supported)
- The pipeline will automatically run and trigger the upgraded state machine

9. If you are using a pre-existing GitHub token, or installing from CodeCommit:

- Update the Installer CloudFormation stack using the template downloaded in step 5, updating the `GithubBranch` to the latest release (eg. `release/v1.5.0`)
- Go to AWS CloudFormation and select the stack: `PBMMAccel-what-you-provided`
- Select Update, select Replace current template, Select Upload a template file
- Select Choose File and select the template you downloaded in step 6 (`AcceleratorInstallerXYZ.template.json` or `AcceleratorInstallerXXX-CodeCommit.template.json`)
- Select Next, Update `GithubBranch` parameter to `release/vX.Y.Z` where X.Y.Z represents the latest release
- Click Next, Next, I acknowledge, Update
- Wait for the CloudFormation stack to update (`Update_Complete` status) (Requires manual refresh)
- Go To Code Pipeline and Release the PBMMAccel-InstallerPipeline

1.6. Post Upgrade Follow-up Tasks for v1.5.x Upgrade

- Once the State Machine finishes in AWS Organizations, delete the `PBMMAccel-Guardrails-Part-2` SCP, as we do not remove policies from Organizations given they could be used by customers for other purposes.
- Optionally, populate DynamoDB with your allocated CIDR ranges:
- You *must* perform the following steps if you plan to:
- move from `"cidr-src": "provided"` to `"cidr-src": "lookup"` for your existing cidr blocks; or
- dynamically assign CIDR ranges to new VPCs using pools that overlap with existing VPC's currently defined in your config file.
- this step populates the DynamoDB tables (`PBMMAccel-cidr-vpc-assign` and `PBMMAccel-cidr-subnet-assign`) with your EXISTING utilized CIDR ranges using the upgrade script a second time *after* the upgrade is complete.
- Using the converted version of your config file (update-config.json)
- Validate you are happy with the `pool` names assigned to each VPC and subnet throughout the config file. Update as appropriate, pool names can be any alpha-numeric string, but a subnets pool must match one of its VPCs pools.
- Locate the python conversion script and review its readme here
- To load DynamoDB with your CIDR ranges, execute: (online, requires credentials to the Organization Management account)

```
python update.py --Region ca-central-1 --LoadDB --ConfigFile update-config.json --AcceleratorPrefix PBMMAccel-
```

- Note the use of the --LoadDB switch, the UPDATED config filename, and that the Accelerator prefix is specified
- If you run this command using your v1.3.8 or v1.3.9 config file, the `\cidr\pool:` values will be improperly assigned in DynamoDB.
- In the Organization Management account, in DynamoDB, select Items, and Verify that the tables have been properly populated.
- the script should have populated the following two DynamoDB tables: `PBMMAccel-cidr-vpc-assign` and `PBMMAccel-cidr-subnet-assign` with all your existing vpc and subnet assignments.
- if you plan to dynamically assign CIDR ranges for any new VPCs, you need to manually create the CIDR pools by adding new item(s) to the DynamoDB Table `PBMMAccel-cidr-pool`. The `PBMMAccel-cidr-pool` table stores CIDR ranges to select from for new CIDR assignments. This table works together with the other two DynamoDB tables to track, assign and maintain non-overlapping CIDR ranges based on a pool name and region.

nple DynamoDB JSON to add an entry to the `PBMMAccel-cidr-pool` table:

```
{
    "id": {
        "S": "1"
    },
    "cidr": {
        "S": "10.0.0.0/13"
    },
    "region": {
        "S": "ca-central-1"
    },
    "pool": {
        "S": "main"
    }
}
```

- where `id` is any unique text, `cidr` is the main cidr block from which VPC cidrs are taken. `region` is the AWS region where the pool is used. `pool` is the name of the pool

NOTES:

- You can populate the `cidr-pools` section of the config file/DynamoDB with values that overlap with the existing assigned ranges in your config file. In this situation, it is CRITICAL that you execute this entire process, to avoid issueing duplicate or overlapping CIDR ranges with those already issued. Alternatively, leverage new unique ranges when populating the `cidr-pools`.
- `cidr-pools` only needs to be populated when a VPC has a `cidr-src` set to `dynamic`.
- Optionally, change all the `cidr-src` values throughout your config file to `lookup`, and remove all the `cidr\value` fields. Once changed, CIDR values will be provided by DynamoDB. Switching to `lookup` requires completion of the previous optional step to first load DynamoDB.
- run the state machine with the input parameters `{"scope": "FULL","mode": "APPLY","verbose": "0"}`
- during the state machine execution, the Accelerator will compare the values returned by DynamoDB with the values from the previous successful state machine execution. If the DynamoDB values were incorrectly populated, the state machine will catch it with a comparison failure message and gracefully fail.

2.4 Functionality

2.4.1 Accelerator Service List

Services

This table indicates whether services are leveraged and/or orchestrated by the Accelerator.

CATEGORY	SERVICE	LEVERAGED	ORCHESTRATED
Compute			
	AWS Lambda	X	
	Amazon Elastic Compute Cloud (EC2)		X
Monitoring & Alerts			
	Amazon CloudTrail		X
	AWS Config		X
	Amazon CloudWatch	X	X
	Amazon EventBridge	X	X
	Amazon Simple Notification Service (SNS)	X	
	AWS Budgets		X
	Systems Manager Inventory		X
Infrastructure			
	AWS CodeCommit	X	
	AWS CodeBuild	X	
	AWS CodePipeline	X	
	AWS CloudFormation	X	
	AWS Cloud Development Kit (CDK) / Software Development Kit (SDK)	X	
	AWS Step Functions	X	
	Amazon Kinesis Data Stream	X	
	Amazon Kinesis Data Firehose	X	
	Amazon Simple Queue Service (SQS)	X	
Data			
	Amazon Simple Storage Service (S3)	X	X
	Amazon DynamoDB	X	
	Amazon Elastic Container Registry (ECR) (incl. ECR Public)	X	
	Systems Manager Parameter Store	X	X
	AWS Secrets Manager	X	
Networking			
	Amazon Virtual Private Cloud (VPC)		X
	AWS Transit Gateway		X
	AWS PrivateLink		X
	Elastic Load Balancer (ELB) (incl. ALB, NLB, GWLB)		X
	Route53		X

CATEGORY	SERVICE	LEVERAGED	ORCHESTRATED
	Route53 Resolver		X
Management			
	AWS Organizations	X	X
	AWS Resource Access Manager (RAM)		X
	AWS Identity and Access Management (IAM)	X	X
	AWS Single Sign-On (SSO)	X	
	AWS Directory Service (incl. AWS Managed AD and AD Connector)		X
	AWS Control Tower	X	X
	AWS IAM Access Analyzer		X
	AWS Cost and Usage Reports		X
	AWS Service Quotas		X
Security			
	AWS GuardDuty		X
	AWS Security Hub		X
	Amazon Macie		X
	Systems Manager Automation		X
	Systems Manager Session Manager		X
	AWS Key Management Service (KMS)	X	X
	AWS Security Token Service (STS)	X	
	AWS Firewall Manager		X
	AWS Network Firewall		X
	AWS Certificate Manager (ACM)		X
Third-Party			
	Fortinet FortiGate and FortiManager (Firewall & Mgmt)		X
	Checkpoint CloudGuard and Manager (Firewall & Mgmt)		X
	`rsyslog` on Amazon Linux 2		X
	Windows Remote Desktop Gateway Bastion		X

If we missed a service, let us know!

2.4.2 1. Accelerator Pricing

1.1. Overview

The AWS Secure Environment Accelerator (ASEA) is available free of charge as an open source solution on GitHub. **You are responsible for the cost of the AWS services enabled, configured, and deployed by the solution.**

The ASEA solution enables, configures and deploys two types of AWS services: services leveraged by the ASEA itself to deliver its capabilities; and services orchestrated by the ASEA to help create a secure multi-account AWS foundation for your users and workloads.

The pricing for services leveraged by the ASEA are relatively consistent and small. The pricing for services orchestrated by the ASEA can vary dramatically based on the underlying architecture, services and features selected by a customer through the customizable configuration file.

Most of the provided example ASEA configuration files (except ultra-lite) build a highly available and scalable multi-datacenter environment with hyperscale routing and enterprise grade security worldwide, something that would cost tens of millions of dollars on-premises and still not achieve the same results.

As shown below, different configuration files can dramatically change the monthly cost of running the solution from $30/month, to $1500/month, to $2400/month, to over $3700/month. The price of the deployed solution is 100% dependent on what the customer deploys, and not on the Accelerator automation engine itself. While the example deployment(s) may appear expensive when used solely for testing in a personal account, they typically only represent a very small percentage of a production customers AWS spend. The examples were designed to minimize costs as a customer scales.

This document is designed to assist customers in understanding the pricing associated with operating the example ASEA configuration files. For full pricing details, please refer to each services pricing page.

1.2. Example Configuration File Pricing

The pricing found in this document is provided as an example only. Pricing represents reasonably steady state, minimal activity or traffic flows, and only includes sample workload accounts when they exist in the example config files.

Pricing is based on the ca-central-1 region, a month with 31 days (744 hours), on-demand pricing and Bring Your Own Licensing (BYOL) for any 3rd party firewalls. This is estimated pricing, the solution is regularly updated and pricing is dependent on the actual version and configuration used to implement the solution.

Any changes to the example configuration file will impact the pricing. These estimates do not include any customer workloads, workloads must be independently priced.

1.2.1. PRICING BY CONFIGURATION FILE

The following table provides the estimated monthly pricing based on the example configuration. Additional information on each of the example config files can be found here.

Example Configuration	Description	Estimated Monthly Pricing
Ultra-Lite	This configuration file was created to represent an extremely minimalistic Accelerator deployment, to demonstrate the art of the possible for an extremely simple config. This example is NOT recommended as it violates many AWS best practices.	$30
Test	Designed to reduce solution costs, while demonstrating full solution functionality (Use for testing Full/Lite configurations or Low Security Profiles). Based on Lite Config w/AWS Network Firewall.	$1,500
Lite	Same as Full Config with the following changes: 1) Reduces the FortiGate instance sizes from c5n.2xl to c5n.xl (VM08 to VM04); 2) Only deploys the 9 required centralized Interface Endpoints (removes 50). All services remain accessible using the AWS public endpoints, but require traversing the perimeter firewalls; 3) Removes the perimeter VPC Interface Endpoints; 4) Removes the Unclass ou and VPC. Four variants of the lite configuration file are provided: - AWS Control Tower w/AWS Network Firewall instead of IPSEC VPN Firewalls **(recommended starting point)** - AWS Network Firewall instead of IPSEC VPN Firewalls - IPSEC VPN integrated 3rd party firewalls - AWS Gateway Load Balancer integrated 3rd party firewalls	$2,575 $2,550 $2,450 +FW lic. $2475 +FW lic.
Full	Large IPSEC VPN Firewalls w/Endpoints - The full configuration file was based on feedback from customers moving into AWS at scale and at a rapid pace. Customers of this nature have indicated that they do not want to have to upsize their perimeter firewalls or add Interface endpoints as their developers start to use new AWS services. These are the two most expensive components of the deployed architecture solution.	$4,200

1.2.2. PRICING BY AWS ACCOUNT (ALL CONFIGURATIONS)

The following table provides the estimated monthly pricing per AWS account for each of the example configuration files.

AWS Account	Description	Ultra Lite	Test	Lite	Full
Management	This is the organization management or root account. This account aggregates organization wide billing, and is used to manage the Accelerator, AWS SSO and SCPs. Access to this account must be highly restricted. This account should not contain any customer resources or workloads.	$10	$75	$140	$140
Operations	This Account is used for centralized IT operational resources (MAD, rsyslog, ITSM, etc.) which need to made available to all accounts in the organization and would generally be used and managed by the Cloud Operations team.	-	$275	$680	$680

AWS Account	Description	Ultra Lite	Test	Lite	Full
Security	The security account is generally used and managed by the customers security and compliance teams, and contains an organizations security tooling and consoles. This account functions as the organization administrative account for Security Hub, GuardDuty, Macie, Firewall Manager, and Access Analyzer. This account also has the ability to assume a view-only role in every account in the organization to conduct security investigations.	$5	$10	$25	$25
Log Archive	The log archive account provides a central aggregation and secure long-term storage location for all logs created within the AWS organization. Logs created in every account in the organization are centralized to an S3 bucket in this account.	$15	$35	$55	$55

AWS Account	Description	Ultra Lite	Test	Lite	Full
Perimeter	This account is used as the centralized internet facing ingress/egress point and contains edge security services for the organizations IaaS based workloads.	-	$590	$410-$700	$1,200
Shared Network	This account is used for centralized or shared networking resources and will typically contain a transit gateway to enable routing between different AWS based and on-premises networks. If a centralized or shared VPC architecture is deployed, this account will also contain VPCs (i.e. Dev, Test, Prod) which are shared via RAM sharing to accounts within designated OUs in the organization. If a spoke architecture is used, the Transit gateway is instead shared to the accounts within the organization.	-	$515	$825-$995	$1,950

AWS Account	Description	Ultra Lite	Test	Lite	Full
MyDev1	This is an optional sample workload account which lives in the Dev organizational unit. Dev accounts have a full set of security guardrails similar to a production accounts and are designed to be used by developers. These accounts leverage either local or centralized networking and are connected to the organizations network via the centralized transit gateway, which is used to access the internet via the perimeter security account or on-premises networks.	-	-	$80	$80

AWS Account	Description	Ultra Lite	Test	Lite	Full
TheFunAccount	This is an optional sample workload account that is created in Sandbox organizational unit. Sandbox accounts are designed for experimentation only, as they have the fewest guardrails, and provide the most cloud native experience. These accounts leverage localized networking and are fully isolated from all other organization networks, with no transit gateway connectivity and direct internet access via a local internet gateway.	-	-	$70	$70
TOTAL	**Estimated Monthly Pricing**	**$30**	**$1500**	**$2,450 - $2,575**	**$4,200**

1.2.3. DETAILED PRICING BY AWS SERVICE (LITE CONFIG – IPSEC VPN ACTIVE/ACTIVE FIREWALLS)

We picked a single example configuration file to provide detailed pricing per service.

The following table provides the estimated monthly pricing per AWS services provisioned by the Accelerator, across all accounts, for the Lite – IPSec VPN configuration.

AWS service	Quantity	Estimated Monthly Pricing
CloudTrail (All Regions)		$28
CloudWatch (All Regions)		$35
CloudWatch Events (All Regions)		$0
CodeBuild		$2
CodeCommit		$0
CodePipeline		$0
Config (All Regions)		$85
Data Transfer		$0
Directory Service	- Managed Active Directory (2 domain controllers) - Shared Directory (2 accounts) - Small AD Connector (1)	$444
DynamoDB		$0
EC2 Container Registry (ECR)		$0.2
Elastic Compute Cloud (EC2)	- NAT Gateway (1) - Remote Desktop Gateway (1 x Windows t3.large) - rsyslog Servers (2 x Linux t3.large) - Fortinet Firewalls (2 x Linux c5n.xlarge) - EBS Volumes (30 GB x 3 instances, 100 GB x 2 instances)	$669
Elastic Load Balancing	- Application Load Balancing (2) - Network Load Balancing (rsyslog) (1)	$55
GuardDuty (All Regions)		$41
Key Management Service (All Regions)		$44
Kinesis		$12
Kinesis Firehose		$2
Lambda (All Regions)		$0
Macie (All Regions)		$4
Route 53	- HostedZones (11) - Resolver Network Interfaces (4)	$378
Secrets Manager		$5
Security Hub (All Regions)		$97
Simple Notification Service (All regions)		$0
Simple Queue Service (All Regions)		$0
Simple Storage Service (All regions)		$6
Step Functions		$1
Systems Manager		$0

AWS service	Quantity	Estimated Monthly Pricing
Virtual Private Cloud	- VPC Endpoints (18) - VPN Connections (2) - Transit Gateway VPC Attachments (5) - Transit Gateway VPN Attachments (2)	$542
TOTAL	**Estimated Monthly Pricing**	$2,450

2.4.3 AWS Secure Environment Accelerator Deployment Capabilities

Overview

Deploys, creates, manages and updates the following objects across a multi-region, multi-account AWS environment

TASK	Accelerator - What happens, WHERE, under what condition, on each state machine execution
AWS Accounts	
- Creates mandatory accounts (accounts which other accounts are dependent on)	organization management (root) account, global scope
- Creates workload accounts (individually or in bulk), base personality determined by ou placement	organization management (root) account, global scope
- Supports native AWS Organization account and OU activities (OU and account rename, move account between OU's, create accounts, etc.)	organization management (root) account, global scope
- Applies a Deny All SCP on any newly created account(s) until successfully guardrailed	organization management (root) account, new account scope (failure to apply guardrails fails the Accelerator and leaves account blocked until remediated)
- Allows bulk parallel* account creation, configuration, updates and guardrail application	creates, guardrails and configures new accounts and regions in parallel per defined personas, organization management (root) account. Control Tower account ingestion is sequential at this time.
- Performs 'account warming' to establish initial limits, when required	state Machine region only, defined accounts (per region potential)
- Checks limit increases, when required (complies with initial limits until increased)	per account, per region (supported limits only)
- Automatically submits limit increases, when required	state Machine region only, defined accounts (per region potential)
- Leverages AWS Control Tower	Accelerator and Control Tower home regions must match, the Accelerator supports all on-by-default regions and will require a standalone install in regions not yet supported by Control Tower
Networking	
- Creates Transit Gateways and TGW route tables incl. static routes and inter-region TGW peering	in the defined region(s), defined account(s)
- Creates centralized and/or local account (bespoke) VPC's	in the defined region(s), defined account(s)
...all completely and individually customizable (per account, VPC, subnet, or OU), Static or Dynamic VPC and subnet CIDR assignments	
- Creates Subnets, Route tables, NACLs, Security groups, NATGWs, IGWs, VGWs, CGWs (per customer specs)	part of any VPC, in the defined region(s), defined account(s) - allows detailed CIDR allocation, and cross-account security group referencing
- Deletes default VPC's (worldwide)	in all regions, in all accounts, can disable regions (all accounts or specific account)
- Creates VPC Endpoints (Gateway and Interface)	part of any VPC, in the defined region(s), defined account(s)
- Configures centralized endpoints (R53 zones populated, shared and attached to local and cross-account VPC's)	configures regional central endpoints (only one 'central' VPC per region)
- Creates Route 53 Private and Public Zones	in the defined account(s), defined region(s), defined VPC(s), global scope
- Creates Resolver Rules and Resolver (inbound/outbound) Endpoints	part of a specific VPC(s), in the defined region(s), defined account(s) (i.e. per region possible)
...including MAD R53 DNS resolver rule creation	created in same region as MAD only, shared to same region VPC's when use-central-endpoints set
- Automatically creates R53 VPC Endpoint Overloaded Zones	same region(s), same account(s) as the endpoint and VPC(s)

TASK	Accelerator - What happens, WHERE, under what condition, on each state machine execution
- Deploys and configures AWS Network Firewall	on any VPC, any region, any account
Cross-Account Object Sharing	
- VPC and Subnet sharing, including account level retagging/ naming (and per account security group 'replication')	VPC's are shared to accounts within the SAME REGION as the source VPC only An OU could have additional VPC's defined for additional regions and would be shared to the appropriate accounts in the same additional regions
- VPC peering and TGW attachments (local and cross-account)	in the defined region, no cross-region attachments or peering supported
- Managed Active Directory sharing	state machine region only (consider same region as the MAD only)(unshare method not implemented)
- Automated TGW inter-region peering	cross-region, cross-account or same-account
- Shares SSM remediation documents	from defined account(s), to defined OU's, in defined regions
Zone sharing and VPC associations	
- Public Hosted Zones	no sharing, no association required (any account, any VPC, any region)
- Private Hosted Zones - i.e. Cloud DNS domains	associated worldwide to all VPCs with use-central-endpoints
- Endpoint Private Hosted Zones	associate within region, for all VPC use-central-endpoints (including cross-account)
- On-premise resolver rules	associate within region, for all VPC use-central-endpoints (including cross-account)
- MAD resolver rule association	same region as the MAD resolver only, assoc. w/all VPC use-central-endpoints
Identity	
- Creates Directory services (Managed Active Directory and Active Directory Connectors)	in a specific VPC, in the defined region, defined account - only 1 per account, therefore can't have a second region in the same account (ADC creation only supported in mandatory accounts)
- Creates Windows admin bastion host auto-scaling group	once per above MAD (once per account), same region as MAD
- Set Windows domain password policies (initial installation only)	once per above MAD (once per account), same region as MAD
- Set IAM account password policies	once per account, global scope
- Creates Windows domain users and groups (initial installation only)	once per above MAD (once per account), same region as MAD
- Creates IAM Policies, Roles, Users, and Groups	once per account, global scope
Cloud Security Services	
- Enables and configs the following AWS services, worldwide w/central specified admin account:	(each service can have specified regions disabled)
- GuardDuty w/S3 protection	enabled all regions, all accounts, admin account per region
- Security Hub (Enables specified security standards, and disables specified individual controls)	enabled all regions, all accounts, admin account per region
- Firewall Manager	enabled once per account (global scope), single admin account

TASK	Accelerator - What happens, WHERE, under what condition, on each state machine execution
- CloudTrail w/Insights and S3 data plane logging	enabled all regions (using Organization trail, stored in Organization Management account)
- Config Recorders/Aggregator	enabled all regions, all regions include global events, aggregator set to specified region in Organization Management account
- Macie	enabled all regions, admin account per region
- IAM Access Analyzer	enabled once per account (global scope), single admin account
- Enables CloudWatch access from central specified admin account	enabled once per account (global scope), two admin accounts (Ops & Security)
- Deploys customer provided SSM remediation documents (four provided out-of-box today)	customized per OU, defined regions, defined accounts
...remediates S3 buckets without KMS CMK encryption and ALB's without centralized logging	customized per OU, all regions, integrated w/SSM remediation, when desired
- Deploys AWS Config rules (managed and custom) including AWS Conformance packs (NIST 800-53 deployed by default + 2 custom)	customized per OU, all regions, all accounts integrated w/SSM remediation, when desired
Other Security Capabilities	
- Creates, deploys and applies Service Control Policies	at the top OU level only, sub-ou's managed directly through AWS Organizations
- Creates Customer Managed KMS Keys w/automatic key rotation (SSM, EBS, S3)	SSM and EBS keys are created if a VPC exists in the region, S3 if we need an Accelerator bucket in the region, per account
- Enables account level default EBS KMS CMK encryption	set if a VPC exists in the region, per account
- Enables S3 Block Public Access	once per account, global scope
- Configures Systems Manager Session Manager w/KMS CMK encryption and centralized logging	set if a VPC exists in the region, per account
- Imports or requests certificates into AWS Certificate Manager	State Machine region only (per region potential, required for ALB deployments)
- Deploys both perimeter and account level ALB's w/Lambda health checks, certs & TLS policies	State Machine region only (per region potential)
- Deploys & configures 3rd party firewall clusters and management instances	in the defined region(s), defined account(s)
...Gateway Load Balancer w/auto-scaling (NEW) and VPN IPSec BGP ECMP deployment options	
- Configuration is fully managed and maintained in AWS CodeCommit - full multi-account configuration history	organization management (root) account
...breaking configuration changes block Accelerator execution	Idempotent - extensive error handling and failure cleanup - Accelerator can be stopped, started, and rerun without implication
Centralized Logging	
- Deploys an rsyslog auto-scaling cluster behind an NLB, all syslogs forwarded to CWL	State Machine region only (per region potential)
- Centralizes logging to a single centralized S3 KMS CMK encrypted bucket (enables, configures and centralizes) incl:	Sets S3 ownership flag, sets bucket retentions

TASK	Accelerator - What happens, WHERE, under what condition, on each state machine execution
- VPC Flow logs (w/Enhanced metadata fields and optional CWL destination)	part of a specific VPC, in the defined region, defined account (to local account bucket in state machine region, replicated to log-archive primary region)
- Organizational Cost and Usage Reports	once per organization, global scope (to local account bucket in state machine region, replicated to log-archive primary region)
- CloudTrail Logs including S3 Data Plane Logs (also sent to CWL)	directly back to log-archive, specified primary region
- All CloudWatch Logs (includes rsyslog logs) (and setting Log group retentions)	State machine region, plus configured regions
- Config History and Snapshots	directly back to log-archive account specified primary region
- Route 53 Public Zone Logs, DNS Resolver Query Logs	to CloudWatch Logs in us-east-1 (which are sent to S3)
- GuardDuty Findings	directly back to log-archive, specified primary region
- Macie Discovery results	directly back to security, specified primary region, replicated to log-archive
- ALB Logs	State Machine region only (same as ALB deployment)
- SSM Session Logs (also sent to CWL)	All regions currently send back to central region, log-archive account
Extensibility	
- Populates each accounts Parameter Store with the Accelerator deployed objects (allows customer IaC to extend/leverage)	each account, defined regions (all ELB's across the environment are populated in specified accounts, i.e. perimeter, to enable automated end-to-end plumbing)
- Every execution outputs the execution status and a list of successfully guardrailed accounts to a SNS topic	allows 3rd party framework to execute after every Accelerator execution by hooking to SNS topic
...which emails a customer defined email address	...or hooking to the email alert
- Deploys roles with customized access (read-only,write) to the log-archive buckets (enabling customer SIEM deployments, SSM, EC2 CWL)	defined account, global scope
- Designed for Day 1, 2 and day 10. Customers get new features without any customization effort no matter the deployed architecture	Upgradable from any version to any version, no customization or professional services required (Customer production proven across multiple releases)
Alerting	
- Deploys global High, Medium, Low, Ignore priority SNS topics and email subscriptions	in the defined account, org accessible regional topics, each region subscribed to a single defined central region which has the email subscriptions
- Deploys customer defined CloudWatch Log Metrics and Alarms w/prioritized alarms (19 out-of-box)	all accounts, home region only, as this is where the Org/account CloudTrail exists
- Creates and configures AWS budgets w/alerting (customizable per OU and per account)	once per account, global scope
- Configures email alerting for CloudTrail Metric Alarms, Firewall Manager Events, Security Hub Findings incl. GuardDuty Findings	

General

- "defined" region, "defined" account, means "customer defined", either at installation, upgrade, or any time they decide to reconfigure

- all items are created per customer defined parameters and configurations and are fully customizable without changing a single line of code
- security services are enabled and deployed globally, but, each service can be disabled per region. A single region deployment is possible.
- customer can enable/disable features, or change the configuration of each feature in the Accelerator config file
- customers can evolve their configurations over time, as they evolve and as their requirements change, without the requirement for code changes or professional services

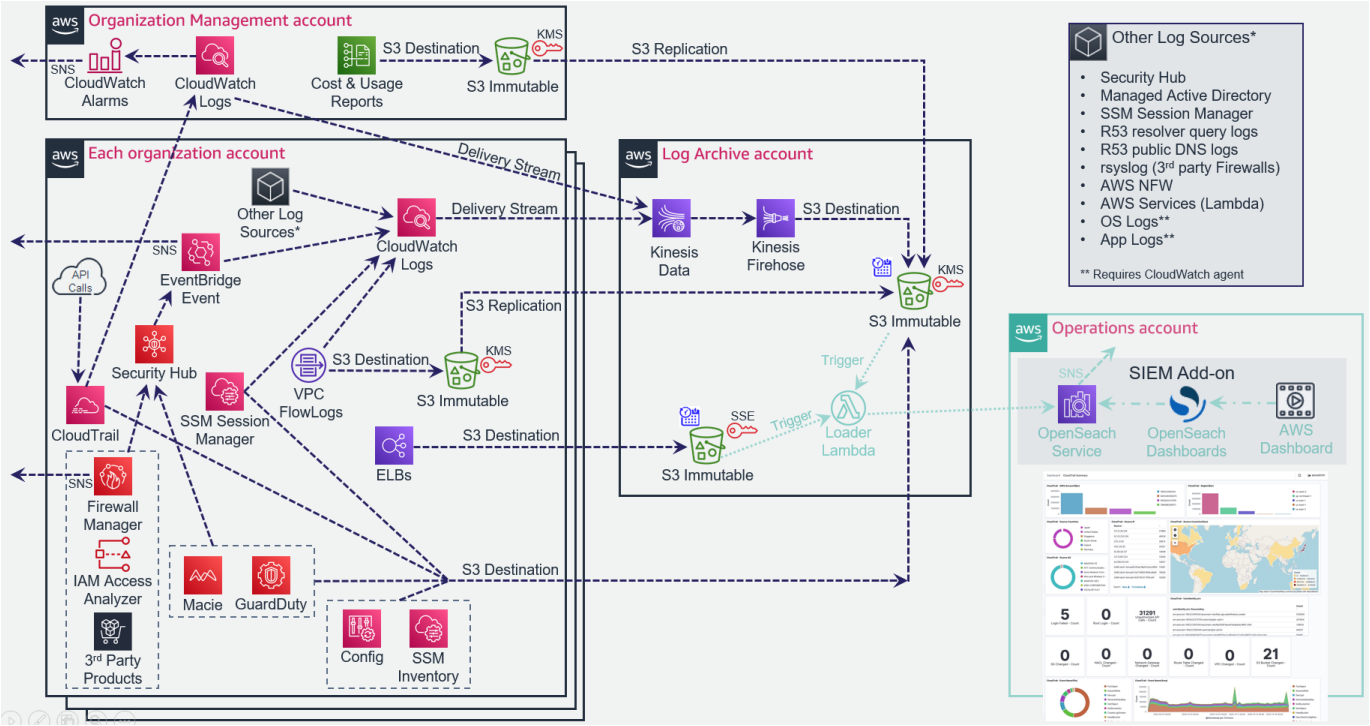
Region support

- All AWS commercial regions are supported. Lack of availability of CodeBuild, CodeCommit, or AWS Organizations in the Accelerator primary or installation region will prevent installation directly in that region. In these cases, customers can select a different installation region and the Accelerator can remotely deploy configurations and guardrails to that unsupported installation region.
- Prior to v1.2.5, we utilized a single StackSet, which blocked several additional installation regions. The Accelerator no longer leverages any StackSets, unblocking installing directly in several additional regions.
- As most features can be toggled on/off (per region), we expect most regions should be supportable both as a primary (or installation) region with the three above noted exceptions, and in these cases should still be fully supported as a managed (or secondary) region.
- Opt-in regions are not yet supported, but given enough demand, could easily be added.

...Return to Accelerator Table of Contents

2.4.4 1. Accelerator Central Logging Implementation and File Structures

The following diagram details the ASEA central logging implementation:



1.1. Accelerator Central Logging Buckets

Bucket Type	Bucket Name	Purpose
AES Encrypted Bucket	pbmmaccel-logarchive-phase0-aescacentral1-1py9vr4cdwuxu	ALB Logs - ALB's do not support logging to a KMS bucket
KMS Encrypted Bucket	pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo	All other AWS Accelerator initiated logs
AES or KMS Encrypted	aws-controltower-logs-123456789012-ca-central-1	All Control Tower initiated logs
AES or KMS Encrypted	aws-controltower-s3-access-logs-123456789012-ca-central-1	S3 Access logs for the Control Tower logs bucket

1.1.1. NOTES

- Every customer has two Accelerator logging buckets
- Control Tower installations have an additional two Control Tower logging buckets
- Customers could use any account name for their central logging account

- Bucket name format is: {Accel-Prefix}-{Account-Name}-{Accel-Phase}-xxx{Region}-{Random}
- {Accel-Prefix} defaults to 'asea' (previously 'pbmmaccel' for Canada)
- {Accel-Phase} should always be 'phase0'
- {region} should always be 'cacentral1' for Canada
- {account} is likely to be 'log-archive'
- xxx is either "aes" or "" (nothing)

1.2. Accelerator Bucket Folders

Log Type	Folder Path	Example
ELB (in AES bucket)	{account#}/elb-{elbname}/AWSLogs/{account#}/*	• s3://pbmmaccel-logarchive-phase0-aescacentral1-1py9vr4ucwuxu/123456789012/elb-Core-mydevacct1-alb/AWSLogs/123456789012/ELBAccessLogTestFile • s3://pbmmaccel-logarchive-phase0-aescacentral1-1py9vr4ucwuxu/123456789013/elb-Public-Prod-perimeter-alb/AWSLogs/123456789013/ELBAccessLogTestFile
VPC Flow Logs	{account#}/{vpc-name}/AWSLogs/{account#}/vpcflowlogs/{region}/{year}/{month}/{day}/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/123456789012/Test-East-lcl/AWSLogs/123456789012/vpcflowlogs/us-east-1/2020/08/31/123456789012_vpcflowlogs_us-east-1_fl-04af3543c74402594_20200831T1720Z_73d3922a.log.gz
Macie Reports	{account#}/macietestobject	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/123456789014/macie-test-object
Cost and Usage Reports	{account#}/cur/Cost-and-Usage-Report/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/123456789015/cur/Cost-and-Usage-Report/*
Config History*	AWSLogs/{account#}/Config/{region}/{year}/{month}/{day}/ConfigHistory/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/AWSLogs/123456789016/Config/ca-central-1/2020/8/31/ConfigHistory/123456789016_Config_ca-central-1_ConfigHistory_AWS::CloudFormation::Stack_20200831T011226Z_20200831T025845Z_1.json.gz
Config Snapshot*	AWSLogs/{account#}/Config/{region}/{year}/{month}/{day}/ConfigSnapshot/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/AWSLogs/123456789016/Config/ca-central-1/2020/8/30/ConfigSnapshot/123456789016_Config_ca-central-1_ConfigSnapshot_20200830T193058Z_5d173149-e6d0-41e4-af7f-031ff736f8c8.json.gz
GuardDuty	AWSLogs/{account#}/GuardDuty/{region}/{year}/{month}/{day}/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/AWSLogs/123456789014/GuardDuty/ca-central-1/2020/09/02/294c9171-4867-3774-9756-f6f6c209616f.jsonl.gz
CloudWatch Logs****	CloudWatchLogs/{year}/{month}/{day}/{hour}/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/CloudWatchLogs/2020/08/30/00/PBMMAccel-Kinesis-Delivery-Stream-1-2020-08-30-00-53-33-35aeea4c-582a-444b-8afa-848567924094
CloudTrail Digest***	{org-id}/AWSLogs/{org-id}/{account#}/CloudTrail-Digest/{region}/{year}/{month}/{day}/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/o-fxozgwu6rc/AWSLogs/o-fxozgwu6rc/123456789016/CloudTrail-Digest/ca-central-1/2020/08/30/123456789016_CloudTrail-Digest_ca-central-1_PBMMAccel-Org-Trail_ca-central-1_20200830T190938Z.json.gz
CloudTrail Insights**	{org-id}/AWSLogs/{org-id}/{account#}/CloudTrail-Insights/{region}/{year}/{month}/{day}/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/o-fxozgwu6rc/AWSLogs/o-fxozgwu6rc/123456789015/CloudTrail-Insight/ca-central-1/2020/09/23/123456789015_CloudTrail-Insight_ca-central-1_20200923T0516Z_KL5e9VCV2SS7IqzB.json.gz
CloudTrail***	{org-id}/AWSLogs/{org-id}/{account#}/CloudTrail/{region}/{year}/{month}/{day}/*	• s3://pbmmaccel-logarchive-phase0-cacentral1-1tr23emhncdzo/o-fxozgwu6rc/AWSLogs/o-fxozgwu6rc/123456789016/CloudTrail/ca-central-1/2020/08/30/123456789016_CloudTrail_ca-central-1_20200830T0115Z_3YQJxwt5qUaOzMtL.json.gz
CT S3 Access Logs	{no folders}	• s3://aws-controltower-s3-access-logs-123456789012-ca-central-1/2021-04-26-18-11-21-8647E1080048E5CB
SSM Inventory	ssm-inventory/{ssm-inventory-type}/accountid={account#}/region={region}/resourcetype={rt}/*	• s3://asea-logarchive-phase0-cacentral1-1tr23emhncdzo/ssm-inventory/AWS:Application/accountid=123456789012/region=ca-central-1/resourcetype=ManagedInstanceInventory/i-001188b4e152aecaf.json

1.2.1. NOTES

* Located in Control Tower bucket when installed, Control Tower adds the {org-id} (i.e. o-h9ho05hcxl/) as the top level folder

** Only available in Accelerator Standalone deployments

*** CloudTrail control plane logs located in Control Tower bucket when installed, Control Tower drops the {org-id} (i.e. o-h9ho05hcxl/) from the middle of the folder path. This may change when Control Tower migrates to Organization Trails. CloudTrail data plane logs remain in the Accelerator bucket.

**** v1.5.1 introduces the capability to split CloudWatch log groups starting with specific prefixes out into customer named subfolders. The folder/file structure is otherwise identical. The v1.5.1 example config files separate out MAD, RQL, Security Hub, NFW, rsyslog, and SSM logs by default. Example: Security Hub logs will be in the following structure: `CloudWatchLogs/security-hub/{year}/{month}/{day}/{hour}/`

- Account number is sometimes duplicated in path because logs replicated from another account always need to start with the source account number
- Macie reports will only appear in the {account#} for the central security account, and only if a customer schedules PII discovery reports
- All CloudWatch Logs from all accounts are mixed in the same folder, the embedded log format contains the source account information as documented here: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/ValidateLogEventFlow.html
- With the exception of CloudWatch Logs, all logs are in the original format provided by the log source/service.

2.4.5 Object Naming

ACCELERATOR OBJECT NAMING

- Resources will have the 'Name' tag assigned, where Name={name}{suffix}
- No prefix or suffix will be applied to DNS records/zones (as that breaks them)
- When _ is not supported, a - will be used
- Stacks/stacksets/functions and ***non-end user*** accessed objects deployed in all accounts will also start with the {AcceleratorPrefix} prefix (i.e. "***PBMMAccel-***" or "***ASEA-***")
- The prefix does not apply to objects like VPC's, subnets, or TGW's which customers need to directly access. This is for objects deployed to build the customer accessible objects
- This prefix will be protected by SCP's so customers don't break 'managed' features
- Resources will have the tag 'Accelerator={AcceleratorName}' assigned when tags are supported
- Stacks will have the tag 'AcceleratorName={AcceleratorName}' assigned, which will often (but not always) be inherited by objects created by the stack (due to TGW duplicate tag issue)

DEFAULTS

```
- the default {AcceleratorName} is 'PBMM' before v1.5.0 and 'ASEA' after v1.5.0
- the default {AcceleratorPrefix} is 'PBMMAccel-' before v1.5.0 and 'ASEA-' after v1.5.0
```

SUFFIX'S

suffix	object type
_vpc	VPC
_azN_net	Subnet
_azN_rt	RouteTable
_tgw	Transit Gateway
-key	KMS key
_pcx	Peering Connection
_sg	Security Group
_nacl	NACL
_alb	Application Load Balancer
_nlb	Network Load Balancer
_agw	Appliance Gateway
_vpce	VPC Endpoint
_AMI	AMI
_dhcp	DHCP option set
_snap	snapshot
_ebs	Block storage
_igw	internet gateway
_lgw	Local gateway
_nat	NAT gateway
_vpg	Virtual private gateway
_cgw	Customer gateway
_vpn	VPN Connection
_sm	Step Functions
_rule	CW Event Rule
_pl	CodeBuild

NO SUFFIX

suffix	object type
None	Stacks
None	CFN_Stack_Sets
None	Lambda
None	Cloud Trails
None	CWL Groups
None	Config Rules
None	OU
None	Service Control Policy

3. Upgrade to Landing Zone Accelerator

3.1 Upgrading from ASEA to Landing Zone Accelerator (LZA)

3.1.1 Overview

The AWS Secure Environment Accelerator (ASEA) launched in 2020 in order for Canadian customers to implement landing zones that complied with the Canadian Centre for Cyber Security Medium Cloud (CCCS-M) profile. As our services continued to evolve, a long-term strategy and plan were developed in 2021 to incorporate features and lessons learned from ASEA, as part of this strategy AWS launched Landing Zone Accelerator on AWS (with Control Tower) which is now the preferred solution for accelerating customer landing zones globally.

This technical guide assists customers in performing an in-place upgrade from ASEA Landing Zone to Landing Zone Accelerator (LZA). The target audience is technical personnel responsible for the deployment and operational management of landing zones.

This documentation package includes:

1. Step-by-step upgrade instructions
2. Command-line upgrade tools
3. Phase-by-phase execution guidance

The upgrade process is executed through a series of command-line scripts, designed to guide users through multiple upgrade phases systematically.

3.1.2 High-level process

To perform a successful upgrade, there is a sequence of tasks that must be completed before the upgrade can begin. The first task involves generating the configuration file for the upgrade tool. Subsequent tasks check that all ASEA resources currently deployed are in the correct state, update ASEA to the latest version, and remediate any resource drift of deployed ASEA resources using the provided scripts. Once the resources are remediated and ASEA is upgraded to the latest version, customers will then enable a new configuration option in the ASEA configuration file that will instruct the ASEA state machine to prepare the environment for upgrade by removing resources that are only necessary to run the ASEA state machine, and other ASEA specific tasks. This will also effectively disable all ASEA CloudFormation custom resources from modifying any of the resources that have been deployed. After the final ASEA state machine run, the ASEA installer stack can be removed from the environment to completely disable and remove ASEA.

Once the ASEA installer stack has been removed, the customer will run a script that will create a mapping of every resource in every account and region that ASEA has deployed, and store that file in Amazon S3 and AWS CodeCommit. This mapping will be used by the Landing Zone Accelerator (LZA) to identify ASEA specific resources that must be modified or referenced in later stages of the upgrade. Once the mapping file is generated, the LZA configuration file generation script can also be run. This file in conjunction with the mapping, will be used to create the LZA configuration files during the upgrade.

After the LZA configuration files are generated, they will be placed in a CodeCommit repository residing in the home installation region of ASEA. Then, the LZA can be installed and reference the configuration repository created above. During the installation, the LZA will reference the newly created configuration, and the LZA code pipeline will install two additional stages. The first stage created will evaluate and create references that the LZA specific resource stacks can reference based off of configuration changes. This stage is executed before any core LZA stages are executed. The last stage created for upgraded environments is executed after all LZA stages are executed. This stage is responsible for adding dependencies created by the LZA to ASEA created stacks to ensure that all resources are handled correctly during the execution of the LZA CodePipeline.

Once the LZA is installed, customer resources will continue to exist and are still modifiable, but interaction with some ASEA resources that remain are handled through the LZA configuration files. Management of LZA native environments and upgraded environments will see almost no difference.

Before starting we strongly encourage you to go through the full documentation and review the Key differences between ASEA and LZA and Feature specific considerations. The preparation steps can be done in advance, can be run multiple times and will not modify your current environment. The upgrade steps should be completed when you are ready to apply the upgrade to your environment.

- Key differences between ASEA and LZA
- Feature specific considerations
- Preparation

a. Pre-requisites and configuration

b. Resource mapping and drift detection

c. Handling drift from Resource mapping

d. Configuration conversion

e. Pre-upgrade validations

- Upgrade

a. Optional preparation steps

b. Disable ASEA

c. Install LZA

d. Finalize the upgrade

- FAQ and Troubleshooting
- FAQ
- Troubleshooting
- Rollback strategy
- ASEA Resource Handlers
- Known issues

3.2 Key differences between ASEA and LZA

3.2.1 Key differences between ASEA and LZA

This section highlights key differences between ASEA and LZA. For further documentation please refer to Landing Zone Accelerator on AWS solution documentation

Accelerator prefix

ASEA by default uses the `ASEA` prefix to identify resources deployed by the accelerator and protect them through SCPs. When LZA is installed during the upgrade process it keeps the existing prefix for existing and new resources to ensure compatibility with the guardrails and uniformity across resources created by ASEA and LZA.

This is different than the default prefix (`AWSAccelerator`) used for a regular LZA installation.

Pipeline execution role

ASEA used the `ASEA-PipelineRole` as the privileged role deployed to all accounts and used by the accelerator to manage resources. The LZA upgraded environment used the `ASEA-LZA-DeploymentRole` . This is defined with this configuration in the global-config.yaml file.

```
cdkOptions:
  centralizeBuckets: true
  useManagementAccessRole: false
  customDeploymentRole: ASEA-LZA-DeploymentRole
```

Service Control Policies (SCP)

During the upgrade, Service Control Policies are kept as-is and not modified. You retain all existing customizations. If you customized the SCPs in ASEA, review your changes to ensure that resources deployed by the accelerator can be modified by the accelerator Pipeline role. This is achieved by having the Organization Admin Role and -PipelineRole listed in several SCP conditions such as:

```
"Condition": {
        "ArnNotLike": {
          "aws:PrincipalARN": ["arn:aws:iam::*:role/${ACCELERATOR_PREFIX}*", "arn:aws:iam::*:role/${ORG_ADMIN_ROLE}"]
        }
      }
```

Verify you have not removed those in your customizations before starting the upgrade.

SSM Parameters to reference accelerator resources

Both accelerators make extensive use of SSM Parameters to store the id of resources created by the accelerator and reference them from other CloudFormation stacks. Most of this behavior is internal to the accelerator and transparent to the end-user.

If you deployed your own customizations using those accelerator created SSM Parameters or reference them in your own Infrastructure as Code, you need to be aware of structural differences between ASEA and LZA parameters.

For example several parameters are created to reference networking resources.

- In ASEA the parameters use a numerical index (e.g. `/ASEA/network/vpc/1/id` contains the ID of the first VPC deployed in the account and `/ASEA/network/vpc/1/net/1/aza/id` contains the ID of the first subnet in AZA of the first VPC)
- In LZA the parameters are indexed by the resource name defined in the network-config.yaml file (e.g. `/ASEA/network/vpc/Central_vpc/id` **** contains the of the VPC named `Central_vpc` and `/ASEA/network/vpc/Central_vpc/subnet/App2_Central_aza_net/id` contains the ID of the `App2_central_aza_net` subnet from the `Central_vpc`)

For AWS accounts created before the upgrade, both sets of parameters will co-exist. For new accounts ad resources created after the upgrade, only the LZA version of the parameters will exist.

Refer to the Landing Zone Accelerator Implementation Guide for a full list of Parameter Store outputs supported by LZA.

Centralized logging

LZA uses the same centralized logging architecture than ASEA to consolidate logs in a central S3 bucket in the Log Archive account. During the upgrade the configuration and dynamic partitioning rules are adapted to keep the same logging structure. If you have external integrations that depend on the logging structure and format, you should closely monitor the logs during the upgrade and review the current section to identify if the differences can impact your integration.

Reference: Landing Zone Accelerator Centralized Logging

VPC FLOW LOGS CLOUDWATCH LOG GROUPS

ASEA uses the following naming convention for the CloudWatch Log Groups names: `/{AcceleratorPrefix}/flowlogs/{vpc-name}``

LZA uses CDK naming which will produce a Log Group name with this pattern: `{Accelerator-Prefix}-NetworkVpcStack-{account}-region-*VpcFlowLogsGroup*.`

During the upgrade, the VPC Flow Logs are re-configured by LZA, therefore new CloudWatch Log Groups are created and new Flow Logs entries are sent to LZA Log Groups, while existing data remain in the ASEA Log Groups.

KINESIS DATA STREAM AND AMAZON DATA FIREHOSE ARE RE-DEPLOYED BY LZA

During the upgrade, LZA deploys new Kinesis Data Streams and Amazon Data Firehose resources in the Log Archive account to replace the ones that were deployed by ASEA. If you have external applications integrated to the logging Data Stream, you need to update them with the new Kinesis Data Stream resource.

LOG AGGREGATION FOR ALL ENABLED REGIONS

By default ASEA deploys the Kinesis Data Streams and Log Group subscription filters to send logs to the central logging bucket only to the home region. Additional regions can be configured with the additional_cwl_regions property.

In LZA, the logging infrastructure is deployed to all `enabledRegions`, this will result in increased number of logs being sent to the central S3 bucket as well as the deployment of a Kinesis Data Stream and Kinesis Data Firehose in the Logging account for every enabled regions.

During the upgrade, existing ASEA subscription filters are replaced by the LZA destination for existing Log Groups subscriptions. The LZA subscription filters are added to new Log Groups created after the upgrade.

SECURITY HUB TO CLOUDWATCH LOGS

When Security Hub is configured to send logs to CloudWatch, in ASEA the forwarding rule and the `/ASEA/SecurityHub` CloudWatch LogGroup is created only in the logging account.

In LZA, the forwarding rule and CloudWatch Log Groups are created in every account and enabled region. This will result in additional logs being sent to CloudWatch and the centralized S3 logging bucket.

ELB ACCESS LOGS

LZA creates new S3 buckets to store ELB access logs in every enabled regions in the central logs account (e.g. `asea-elb-access-logs-<account>-<region>`). ASEA stored the ELB access logs on the `asea-logarchive-phase0-aes<region>-<suffix>` bucket. After the upgrade, the `ASEA-LZA-ELB_LOGGING_ENABLED` AWS Config Rule will update the logging destination of all existing ELBs to use the new LZA buckets.

Customer Managed Keys

There are differences between how ASEA and LZA manage AWS KMS keys to provide encryption at rest capabilities for resources deployed by the solution. Detailed documentation is available in the Customer Managed Keys - Comparison of ASEA and LZA document.

Cost considerations

Due to architectural and operational differences between ASEA and LZA, you can see an increase of the AWS resources cost during and after the upgrade. We recommend that you monitor the costs of your environment on a daily basis to detect any anomaly.

DURING THE UPGRADE

The upgrade itself makes changes to a significant number of resources, therefore it is expected that applying the upgrade will incur a significant AWS Config cost the day the upgrade is applied. The same behavior can be seen when initially installing the accelerator or when a State Machine/pipeline run affects a large number of resources.

During the upgrade process it is expected that some resources will exist twice for some time. The ASEA created resource and the LZA created resource, until the cleanup process happens.

Both these impacts are temporary and the cost will stabilize when the upgrade is complete.

AFTER THE UPGRADE

LZA has the capability to deploy and configure more services than ASEA, during the upgrade new capabilities are not deployed unless required, you can choose to enable additional services once the upgrade is complete. LZA uses more granular KMS keys than ASEA, new Customer Manager Keys will be created as part of the upgrade, the impact on your total costs depends on the number of accounts and regions in use in your environment. Review the Customer Managed Keys - Comparison of ASEA and LZA document for more details.

By default LZA consolidate more logs than ASEA to CloudWatch Logs, review the section on Centralized logging to understand how the additional logging can impact costs.

3.2.2 Customer Managed Keys - Comparison of ASEA and LZA

There are differences between how ASEA and LZA manage AWS KMS keys to provide encryption at rest capabilities for resources deployed by the solution. In general, LZA uses more granular keys for each service as well as configuration options to control where the keys are deployed. Some AWS KMS keys are deployed to every account and Region managed by the solution, while others are centralized in a single core account.

This document provides details about important differences between the management of keys by ASEA and LZA and how the different keys are handled during the upgrade.

Refer to the Key management section of the **Landing Zone Accelerator on AWS Implementation Guide** for more details on the configuration options for AWS KMS keys offered by LZA.

Important

This documentation calls out specific cases where it might be possible for you to delete KMS keys that are no longer needed. This is a high-risk operation that requires careful assessment. Maintaining unused KMS keys does not affect accelerator functionality. **Deleting an AWS KMS key is destructive and potentially dangerous. It deletes the key material and all metadata associated with the KMS key and is irreversible. After a KMS key is deleted, you can no longer decrypt the data that was encrypted under that KMS key, which means that data becomes unrecoverable.** Refer to AWS documentation on Deleting AWS KMS keys for more information.

General approach

- In general the upgrade process aims to align the usage of Customer Managed Keys (CMK) to the default LZA configuration
- In existing accounts you may see the existence of ASEA created keys and LZA created keys
- New AWS accounts created after the upgrade will only have LZA created keys
- For cases where the LZA configuration supports `deploymentTargets` for keys, the convert-config process generates a configuration to create CMKs in regions with VPCs (which in most cases corresponds to the home region AND additional regions identified to deploy workloads). Customers can choose to modify this configuration to suit theirs needs.
- The upgrade process never schedules the deletion of an existing key. In most cases existing ASEA keys need to be kept for as long as there is data encrypted with the key. When applicable, we provide guidance on specific use cases where older keys can be manually deleted by the customer if they choose to.

Key specifics details

S3/BUCKET KEY

ASEA: Creates a `Bucket-Key` in all accounts in the home region. This key is used for other resources as well (see references below)

LZA: Creates an `s3` key in all accounts and regions to encrypt Amazon S3 buckets created by the solution. Deployment of the key can be controlled using deploymentTargets in the configuration. (ref: S3GlobalConfig)

Upgrade: The convert-config process generates a configuration to create a LZA `s3` key in all accounts and in regions which have VPCs deployed. The existing ASEA `Bucket-Key` in existing accounts is kept and needed to provide access to data already encrypted using the key. New AWS accounts created after the upgrade will only have the LZA `s3` key.

Central Logging bucket

ASEA: Uses the `Bucket-Key` in the Log Archive account

LZA: Creates a `central-logs/s3` in the Log Archive account

Upgrade: The existing central logging bucket in the Log Archive account is imported during the upgrade process. The existing ASEA `Bucket-Key` from the Log Archive account continues to be used to encrypt existing and new data stored on the central logging bucket.

The KMS resource-based policy for this CMK is copied to the LZA configuration repository (`kms-policies/central-log-bucket-key.json`) during configuration conversion. Any future changes needed to this policy need to be done by modifying that file and re-running the LZA pipeline. LZA won't dynamically update this policy.

A sample scenario where modifying this KMS policy might be needed is when adding an AWS opt-in region such as `ca-west-1` to the enabled regions of your configuration. The policy need to be modified to allow `guardduty.ca-west-1.amazon.com` as a service that can use the key. See Exporting generated GuardDuty findings to Amazon S3 buckets for more details.

CLOUDWATCH

ASEA: Uses service managed keys for CloudWatch encryption

LZA: Creates a `cloudwatch` key in all accounts and regions used to encrypt CloudWatch Logs groups created by the solution. Deployment of the key can be controlled using deploymentTargets in the configuration (ref: CloudWatchLogsConfig)

Upgrade: The convert-config process generates a configuration to create a LZA `cloudwatch` key in all accounts and in regions which have VPCs deployed.

SNS TOPICS

ASEA: Uses the bucket key to encrypt SNS topics

LZA: Creates a dedicated `snstopic` key in accounts where notification SNS topics are deployed

Upgrade: SNS Topics are deployed to the Management and Audit accounts and a dedicated `snstopic` key is deployed in those accounts.

LAMBDA

ASEA: Uses service managed keys for Lambda environment variables encryption

LZA: Creates a `lambda` key used to encrypt environment variables for Lambda functions created by the solution. Deployment of the key can be controlled using deploymentTargets in the configuration (ref: LambdaConfig)

Upgrade: The convert-config process generates a configuration that don't create keys for Lambda. Service managed keys will continue to be used for Lambda environment variables encryption. You can opt-in to have LZA create Customer Managed Keys by modifying the LZA configuration.

EBS

ASEA: Creates a `EBS-Key` in all regions with VPCs

LZA: Controlled by the EbsDefaultVolumeEncryptionConfig in security-config.yaml to be used for default encryption of Amazon EBS volumes.

Upgrade: The convert-config process generates a configuration to create a LZA `ebs/default-encryption` key in all accounts and regions which have VPCs deployed. The existing ASEA `EBS-Key` is kept in existing accounts for existing volumes and snapshot encrypted by the key. New volumes created after the upgrade will use the LZA `ebs/default-encryption` key by default.

Post-upgrade: Customers can decide to manually remove the ASEA `EBS-Key` from individual accounts once they confirm that no volumes, snapshots or other data and resources is using the key. **Deleting an AWS KMS key is destructive and potentially dangerous. It deletes the key material and all metadata associated with the KMS key and is irreversible. After a KMS key is deleted, you can no longer decrypt the data that was encrypted under that KMS key, which means that data becomes unrecoverable.** Refer to AWS documentation on Deleting AWS KMS keys for more information.

SYSTEM MANAGER SESSION MANAGER

ASEA: Creates a `SSM-Key` in all regions and accounts with VPCs. This key is used to encrypt Session Manager sessions AND the Session Manager Log Group

LZA: Creates a `sessionmanager-logs/session` key to encrypt Session Manager sessions if Session Manager logging is activated in the global-config.yaml file. In all accounts and regions. Uses the CloudWatch key to encrypt Session Manager Log Group.

Upgrade: Create `sessionmanager-logs/session` key to encrypt Session Manager sessions in all accounts and all regions with VPCs. The `CloudWatch` key is used to encrypt Session Manager Log Group.

Post-upgrade: Customers can decide to manually remove the ASEA `SSM-Key` from individual accounts once they confirm that not CloudWatch logs or other data and resources is using the key. The Session Manager sessions data is short-lived, however the `SSM-Key` is also used to encrypt the `/[<accelerator-prefix</SSM` Log Group. Deleting the key will prevent access to existing logs in this Log Group. Only delete the key once you confirm you no longer need access to the data from this log group according to your retention policy. Note that all Cloud Watch Log Groups logs are also delivered to the central logging bucket for long term retention. The central logging bucket uses the `ASEA-Bucket` key for encryption. **Deleting an**

AWS KMS key is destructive and potentially dangerous. It deletes the key material and all metadata associated with the KMS key and is irreversible. After a KMS key is deleted, you can no longer decrypt the data that was encrypted under that KMS key, which means that data becomes unrecoverable. Refer to AWS documentation on Deleting AWS KMS keys for more information.

3.2.3 Feature specific considerations

This section contains documentation about specific features that may require manual intervention because they can't be fully automated by this upgrade process. Review each item that applies to your environment. Several warnings that can be generated from the `convert-config` command refer to items from this section.

SYSTEM MANAGER DOCUMENTS

ASEA deploys System Manager documents through the `global-options/ssm-automation` configuration attributes and share those documents to other accounts. The configuration converter generates corresponding configuration with the `ssmAutomation` attribute in the `security-config.yaml` to re-create those documents through LZA.

The upgrade process does not remove the ASEA created documents, you need to review and remove them manually if needed. The documents created by ASEA are named `ASEA-<document-name>` and owned by the operations account. Those created by LZA are named `ASEA-LZA-<document-name>` and owned by the security account.

RSYSLOG SERVERS

> **convert-config warning message**
>
> *rsyslog servers are deployed in ${accountKey}. Please refer to documentation on how to manage these resources after the upgrade.*

ASEA can deploy rsyslog servers with an auto-scaling group and Network Load Balancer. These rsyslog servers are configured to forward logs to a CloudWatch log group. They are not designed to store long term data and can then be replaced with minimal impact.

During the upgrade the existing deployed resources are not modified and remain in the original ASEA CloudFormation stacks. No LZA configuration elements are generated automatically for rsyslog.

We recommend that you provision new rsyslog servers and NLB with LZA, reconfigure any appliance that send logs to these servers with the new NLB address and then decommission the resources provisioned by ASEA once you confirm all traffic is sent to the new servers.

How to deploy rsyslog servers with LZA?

To deploy rsyslog servers with LZA you can leverage the applications customization capability. A sample is available in the LZA CCCS Medium reference architecture.

How to remove the ASEA deployed rsyslog servers?

Once you confirm the rsyslog servers deployed from ASEA are no longer in use, you can delete them by running the following command from the migration tool to flag the rsyslog to be deleted and then run the LZA pipeline. They will be deleted in the `ImportAseaResources` stage of the pipeline.

```
yarn run post-migration remove-rsyslog
```

THIRD-PARTY FIREWALLS

> **convert-config warning message**
>
> *Third-Party firewalls ${firewall.name} are deployed in ${accountKey}. Please refer to documentation on how to manage these resources after the upgrade.*

Third-Party firewall appliances (such as FortiGate) can be deployed by ASEA and once deployed and configured their lifecycle are managed outside of the accelerator (i.e. patching and configuration changes are handled directly through the appliance UI or CLI).

During the upgrade, the existing deployed resources are not modified and remain in the original ASEA CloudFormation stacks. The firewalls can continue to be managed as before (i.e. outside the accelerator) and no other actions are needed in relation to the upgrade.

During the configuration conversion a `firewalls/instances` configuration block is added to the customizations-config.yaml file to allow the use of ${ACCEL_LOOKUP} variables in the network-config.file to reference the firewall instances.

Which configuration changes to ASEA Firewall instances are supported from LZA?

Only removing the Firewalls from the configuration file to decommission them is supported. Any other changes to the configuration (i.e. change the AMI used) will be ignored by the accelerator.

APPLICATION LOAD BALANCERS

ASEA has the ability to deploy ALBs in individual accounts (e.g. Perimeter account) or be configured at the OU level to deploy ALB in every accounts of the OU.

During upgrade, the existing deployed resources are not modified and remain in the original ASEA CloudFormation stacks.

The recommendation is to create new ALBs through LZA, reconfigure the workloads to use them, and then decommission the ASEA ALBs.

Which configuration changes to ASEA Application Load Balancers are supported from LZA?

The ASEA ALBs are not converted to the LZA configuration files, no changes are supported through LZA pipeline. You can modify ASEA ALB properties, modify and add rules to ALB listeners or change target groups directly through the AWS console or APIs.

Once you redeploy new ALBs with LZA those can be managed through the regular LZA configuration files.

How to define ALB to be created in every workload account of an OU?

To achieve the same pattern than ASEA where ALB are defined at the OU level and deployed in every workload account of the OU, you can refer to this example configuration from the LZA CCCS Medium reference architecture.

How to remove ASEA deployed Application Load Balancers?

In the future a `post-migration` command will be added to the LZA upgrade tool to remove ASEA Application Load Balancers. At the moment if you need to remove ASEA ALBs once you confirm they are are no longer in use you can do it through the AWS console or APIs.

ALB IP FORWARDER

If you are using ALB IP Forwarding in ASEA, (`"alb-forwarding": true` is set for a VPC in the ASEA configuration file), the following will occur as a result of the config-converter script:

- The AlbIpForwardingStack.template.json CloudFormation stack will be added to the LZA Configuration CodeCommit repository under the cloudformation path. ex: cloudformation/AlbIpForwardingStack.template.json
- The VPC Name containing the front-end ALBs will be determined (i.e. Perimeter VPC)
- A `customizations-config.yaml` file will be generated in the LZA Configuration CodeCommit repository in the root directory.
- In the `customizations-config.yaml` file, the following entry will be added for each VPC with ALB Forwarding enabled to the `customizations/cloudFormationStacks` section of the configuration:

```
  - name: <AcceleratorPrefix>-AlbIPForwardingStack
    template: cloudformation/AlbIpForwardingStack.template.json
    runOrder: 1
    parameters:
      - name: acceleratorPrefix
        value: <AcceleratorPrefix>
      - name: vpcName
        value: <VPC_NAME>
    terminationProtection: true
    deploymentTargets:
      accounts:
        - Perimeter
    regions:
      - ca-central-1
```

Once the Customizations stage of the pipeline has been successfully run with the configuration file above, a new DynamoDB table will be generated in the `deploymentTargets` account and region specified. This table should be named `Alb-Ip-Forwarding-<VPC_NAME>`. In the same region and account, a DynamoDB table named `<ASEA-Prefix>-Alb-Ip-Forwarding-<VPC-ID>` should exist. You will need to copy over all of these entries from the old ALB IP Forwarding table to the new one.

For more details about ALB Forwarding in LZA, refer to the post-deployment instructions of LZA CCCS Medium reference architecture.

MANAGED ACTIVE DIRECTORY

> **convert-config warning message**
>
> *Managed AD is deployed in ${accountKey}. Please refer to documentation on how to manage these resources after the upgrade.*

During the upgrade the existing Managed Active Directory resource is not modified, remain in the original ASEA CloudFormation stacks and you can continue to managed Active Directory objects through the Windows AD Management Tool from any instance joined to the domain.

Is there still an AD EC2 management instance (i.e. RDGW) created?

The management instance created by ASEA using the ASEA-RDGWAutoScalingGroup will still be present and you can continue to use it to manage the Active Directory objects.

Which configuration changes to ASEA Managed AD are supported from LZA configuration?

No changes to the Managed AD resources created by ASEA are supported through the LZA configuration. The configuration converter doesn't generate any corresponding block in LZA configuration.

LZA configurations support the creation of new Managed Active Directory using the ManagedActiveDirectoryConfig configuration. Do not declare a `managedActiveDirectories` block in your LZA configuration with the same domain than the one created in ASEA, this will be ignored.

How to decommission a Managed Active Directory that was deployed by ASEA?

The resources need to be decommissioned manually. In the future a flag could be added to the `post-migration` command to flag the resources for removal.

GATEWAY LOAD BALANCER

> **convert-config warning message**
>
> *The account ${accountKey} utilizes a Gateway Load Balancer: ${loadBalancerItem.name}. Please refer to documentation on how to manage these resources.* or *The organizational unit ${ouKey} utilizes a Gateway Load Balancer: ${loadBalancerItem.name}. Please refer to documentation on how to manage these resources.*

If you are using Gateway Load Balancers (GWLB) in ASEA, (`"type: "GWLB"` is set for one of your Load Balancers in the `alb` configuration), the configuration tool will not map the existing GWLB in ASEA to the LZA configuration.

> **Warning**
>
> Review the FAQ entry Gateway Load Balancer are not supported in the configuration conversion, how will this impact the workload availability? to assess the potential impact of your workload availability during the upgrade.

If you're looking to implement GWLBs in your environment, you can do so by referencing the central network services configuration within LZA. The LZA configuration allows end-users to define multiple GWLBs and VPC and subnets of where these resources are provisioned. End-users can also define which subnets the service endpoints are distributed to.

To set up GWLBs in your LZA environment, reference the `network-config.yaml` file and specify the `gatewayLoadBalancers` configuration within the `centralNetworkServices` configuration:

```
gatewayLoadBalancers:
  - name: <AcceleratorPrefix>-GWLB
    subnets:
      - Network-Inspection-Firewall-A
      - Network-Inspection-Firewall-B
    account: Network
    vpc: Network-Inspection
    deletionProtection: true
    endpoints:
      - name: Endpoint-A
        account: Network
        subnet: Network-Inspection-A
        vpc: Network-Inspection
      - name: Endpoint-B
        account: Network
        subnet: Network-Inspection-B
        vpc: Network-Inspection
```

GATEWAY LOAD BALANCER ENDPOINT ROUTES

To specify routes to the GWLB for inspection, reference the Subnet route tables configuration within LZA, for example taking in the above configuration:

```
- name: GwlbRoute
  destination: 0.0.0.0/0
  type: gatewayLoadBalancerEndpoint
  target: Endpoint-A
```

CUSTOM IAM ROLE TRUST POLICIES

convert-config warning message

The trust policy for the role ${role.role} ... Please refer to documentation on how to manage these resources.

The LZA solution supports multiple types of assumeRole policies. The following are supported with their respective LZA configurations, particularly as it relates to the `assumedBy` property for the IAM Role set configuration:

Using a policy to delegate access to AWS services:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "elasticmapreduce.amazonaws.com",
          "datapipeline.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

LZA configuration:

```
- name: EC2-Role
  instanceProfile: true
  assumedBy:
    - type: service
      principal: elasticmapreduce.amazonaws.com
    - type: service
      principal: datapipeline.amazonaws.com
  policies:
    awsManaged:
      - AmazonElasticMapReduceFullAccess
      - AWSDataPipeline_PowerUser
      - CloudWatchAgentServerPolicy
  boundaryPolicy: Default-Boundary-Policy
```

Using a policy to delegate access to all principals in an account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

LZA configuration:

```
- name: EC2-Readonly-Role
  assumedBy:
    - type: account
      principal: '123456789012'
  policies:
    awsManaged:
      - AmazonEC2ReadOnlyAccess
```

Using a policy to delegate access to cross-account principals

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/test-access-role"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

LZA configuration:

```
- name: Network-Security-Role
  assumedBy:
    - type: principalArn
      principal: 'arn:aws:iam::444455556666:role/test-access-role'
  policies:
    awsManaged:
      - AmazonSSMManagedInstanceCore
      - AmazonEC2ReadOnlyAccess
  boundaryPolicy: Default-Boundary-Policy
```

Using a policy to provide 3rd party access via external ID conditionals

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/test-access-role"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "111122223333",
        },
      },
    }
  ]
}
```

LZA configuration:

```
- name: Network-Security-Role
  assumedBy:
    - type: principalArn
      principal: 'arn:aws:iam::444455556666:role/test-access-role'
  externalIds:
    - 111122223333
  policies:
    awsManaged:
      - AmazonSSMManagedInstanceCore
      - AmazonEC2ReadOnlyAccess
  boundaryPolicy: Default-Boundary-Policy
```

Using a SAML Provider to Federate:

```
{
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Action": "sts:AssumeRoleWithSAML",
      "Principal": {"Federated": "arn:aws:iam::account-id:saml-provider/Test-SAML"},
      "Condition": {"StringEquals": {"SAML:aud": "https://signin.aws.amazon.com/saml"}}
    }
  }
```

LZA Configuration:

```
providers:
  - name: Test-SAML
    metadataDocument: path/to/metadata.xml

- name: Network-Security-Role
  assumedBy:
    - type: provider
      principal: Test-SAML
  externalIds:
    - 111122223333
  policies:
    awsManaged:
      - AmazonSSMManagedInstanceCore
      - AmazonEC2ReadOnlyAccess
  boundaryPolicy: Default-Boundary-Policy
```

If an assume role policy is needed outside of the scope of what's natively supported in LZA, it's recommended to lean on LZA to provision the IAM Role and trust policy through the customizations layer:

- Create your own CloudFormation template and add it to the `customizations-config.yaml` file, which will be generated in the LZA Configuration CodeCommit repository in the root directory.

PUBLIC AND PRIVATE HOSTED ZONES

convert-config warning message

The VPC ${vpcItem.name} in account ${accountKey} utilizes a public Route53 zone: ${zone}. Please refer to documentation on how to manage these resources. or *The VPC ${vpcItem.name} in OU ${ouKey} utilizes a public Route53 zone: ${zone}. Please refer to documentation on how to manage these resources.*

In ASEA you can create Route53 Public and Private Hosted Zone through the configuration file. Once the zone is created you need to manage its records outside of the accelerator.

e.g.

```
"zones": {
  "public": [
    "cloud-hosted-publicdomain.example.ca"
  ],
  "private": [
    "cloud-hosted-privatedomain.example.ca"
  ]
},
```

As of right now, LZA only supports the creation of private hosted zones in association with creating Vpc Interface Endpoints (for centralized distribution) as well as for Route 53 Resolver Rules. It does not support the creation of custom public or private hosted zone.

After the upgrade to LZA you can continue to manage records in the existing zones. To create new Route53 zones you will need to create your own CloudFormation template and add it to the `customizations-config.yaml` file, which will be generated in the LZA Configuration CodeCommit repository in the root directory.

VPC TEMPLATES

convert-config warning message

The VPC ${vpcItem.name} in OU ${ouKey} is set to deploy 'local' in each account. You need to add a vpcTemplate to your configuration to keep the same behavior for new accounts in this OU.

In ASEA you can define a VPC at the OU level with a `local` deployment. This can be used with dynamic or provided CIDR ranges.

For example, this is used in the sample config file to create local VPC in each Sandbox account.

```
"vpc": [
  {
    "deploy": "local",
    "name": "${CONFIG::OU_NAME}",
    "description": "This VPC is deployed locally in each Sandbox account and each account/VPC is deployed with the same identical CIDR range.  This VPC has no
access to the rest of the Organizations networking and has direct internet access and does not use the perimeter ingress/egress services.",
    "cidr-src": "dynamic",
    "cidr": [
      {
        "size": 16,
        "pool": "main"
      }
    ]
  ...
```

During the upgrade, each existing account using this feature will have its own VPC added to the configuration with the current CIDR range assigned to the VPC. To allow the creation of new accounts in this OU with a local VPC with a similar behavior than ASEA you need to add a vpcTemplate to your configuration.

Example using a provided CIDR range:

```
vpcTemplates:
  - name: Sandbox-Template
    region: {{ AcceleratorHomeRegion }}
    deploymentTargets:
      organizationalUnits:
        - Sandbox
      excludedAccounts:
        - Sandbox01
        - Sandbox02
    cidrs:
      - 10.100.0.0/20
    internetGateway: true
    enableDnsHostnames: true
    enableDnsSupport: true
    instanceTenancy: default
    routeTables:
      - name: Network-Sandbox-A
        routes:
          - name: NatRoute
            destination: 0.0.0.0/0
            type: natGateway
            target: Nat-Network-Sandbox-A
          - name: S3Gateway
            type: gatewayEndpoint
            target: s3
          - name: DynamoDBGateway
            type: gatewayEndpoint
            target: dynamodb
      - name: Network-Sandbox-B
        routes:
          - name: NatRoute
            destination: 0.0.0.0/0
            type: natGateway
            target: Nat-Network-Sandbox-B
          - name: S3Gateway
            type: gatewayEndpoint
            target: s3
          - name: DynamoDBGateway
            type: gatewayEndpoint
            target: dynamodb
      - name: Network-Sandbox-Nat-A
        routes:
          - name: IgwRoute
            destination: 0.0.0.0/0
            type: internetGateway
            target: IGW
      - name: Network-Sandbox-Nat-B
        routes:
          - name: IgwRoute
            destination: 0.0.0.0/0
            type: internetGateway
            target: IGW
    subnets:
      - name: Network-Sandbox-A
        availabilityZone: a
        routeTable: Network-Sandbox-A
        ipv4CidrBlock: 10.100.0.0/24
      - name: Network-Sandbox-B
        availabilityZone: b
        routeTable: Network-Sandbox-B
        ipv4CidrBlock: 10.100.1.0/24
      - name: Network-SandboxNat-A
        availabilityZone: a
        routeTable: Network-Sandbox-Nat-A
        ipv4CidrBlock: 10.100.2.0/28
      - name: Network-SandboxNat-B
        availabilityZone: b
        routeTable: Network-Sandbox-Nat-B
        ipv4CidrBlock: 10.100.2.16/28
    natGateways:
      - name: Nat-Network-Sandbox-A
        subnet: Network-SandboxNat-A
      - name: Nat-Network-Sandbox-B
        subnet: Network-SandboxNat-B
    gatewayEndpoints:
      defaultPolicy: Default
      endpoints:
        - service: s3
        - service: dynamodb
```

Existing accounts

It is important to add the existing accounts that were upgraded from ASEA to the `deploymentTargets/excludedAccounts` list to avoid creating new VPC into the existing accounts.

CUSTOM AWS CONFIG RULES

convert-config warning message

Custom AWS Config Rule with detection needs an IAM policy written Rule Name: "${configRule.name}". Policy file name: "${detectionPolicyPath} or *Custom AWS Config Rule with remediation needs an IAM policy written Rule Name: "${configRule.name}". Policy file name: "${remediationPolicyPath}*

Custom AWS Config Rules need to have the appropriate permissions to execute the detection Lambda and the remediation SSM Documents. The way those permissions are provided are different between ASEA and LZA.

During the configuration conversion process, empty policy files will be generated under the `custom-config-rules` folder. You need to provide the appropriate policies in those files prior to the LZA installation. LZA will create the IAM Roles with the permissions provided in those files that are referenced with the `rolePolicyFile` property in `security-config.yaml`. The `AutomationAssumeRole` parameter is automatically provided by LZA with the created role to the remediation document and no longer should be provided as an explicit parameter.

SUSPENDED ACCOUNTS

All suspended accounts in your organization should be under a specific OU that is ignored by the accelerator.

See ASEA FAQ 1.1.0 How do I suspend an AWS account? for more details.

The presence of Suspended accounts in *regular* OUs (i.e. Dev, Test, Prod) will generate errors during the upgrade.

3.3 Preparation

3.3.1 Preparation

The preparation steps can be done in advance, can be run multiple times and will not modify your current environment. We encourage customers to start the preparation as soon as possible, even if not planning to apply the upgrade immediately.

The preparations steps are

- Preparation:

a. Pre-requisites and configuration

b. Resource mapping and drift detection

c. Handling drift from Resource mapping

d. Configuration conversion

e. Pre-upgrade validations

Optional inventory step

There is an optional read-only inventory script that you can run as part of your early preparation to to help identify features or configurations that require extra planning or considerations.

3.3.2 Upgrade pre-requisites and configuration

Prerequisites

- You are running the latest version of ASEA. If you are not running ASEA version 1.5.11 then upgrade ASEA before starting the ASEA to LZA upgrade process
- Confirm all suspended accounts are under a specific OU that is ignored by the accelerator. (see Suspended accounts)
- Confirm you don't have any empty nested OU without active AWS Accounts that are not referenced from the ASEA configuration files (i.e. `Dev/nestedOU`). The convert-config tool won't generate empty nested OUs in the configuration.
- You can run the scripts from your local workstation. If you are filtering egress traffic from your corporate network you need to ensure outbound connectivity to AWS service endpoints.
- You will need Git, AWS CLI, NodeJS and Yarn installed.
- We highly recommend having appropriate AWS Support plans on all AWS Accounts of your landing zone. For any issues encountered during the upgrade process you need to open a support case to get assistance and exchange relevant information with AWS staff. At a minimum Developer support is needed on the management account and core landing zones accounts (Logging, Security, Networking and Perimeter) to troubleshoot any cross-account issues. Business support is the minimum recommended tier if you have production workloads in AWS
- Monitor and manage your service quotas. See the FAQ Which Service Quotas should be monitored for the upgrade?
- If using an AWS opt-in region, you need to enable to set STS Session tokens to be valid in All AWS Regions.
- Upgrading your landing zone from ASEA to LZA requires advanced knowledge of configuring and operating ASEA and LZA landing zones. This operation should be led by your most-experienced resources responsible for your current landing zone operations. Review all the documentation in this upgrade guide and Landing Zone Accelerator implementation guide.

TECHNICAL PREREQUISITES

Before running the upgrade tools, ensure you meet the following requirements:

> **Environment Requirements**
>
> ✅ **Recommended Environment:** Linux or MacOS with a Bash-like shell
>
> ⚠️ **Important Note:** Windows compatibility is limited as tools have not been extensively tested on this platform

Verify npm installation

Node.js uses the npm package manager to help you install tools and frameworks for use in your application. To confirm you have npm installed you can run the following command:

```
npm -v
```

Set your node heap size

Set your node heap size to at least 4k

```
export NODE_OPTIONS=--max-old-space-size=4096
```

Install Yarn

Utilizing the npm package manager, you can install yarn globally using the following command:

```
npm install -g yarn
```

CLONE THE ASEA REPO

In order to prepare the ASEA environment for upgrade you will need to clone the ASEA GitHub repository: https://github.com/aws-samples/aws-secure-environment-accelerator.git

```
git clone https://github.com/aws-samples/aws-secure-environment-accelerator.git
```

INSTALL THE UPGRADE SCRIPTS PROJECT DEPENDENCIES AND BUILD THE PROJECT

- Navigate to the directory which contains the upgrade scripts:

```
cd aws-secure-environment-accelerator
cd reference-artifacts/Custom-Scripts/lza-upgrade
```

- Install dependencies and build the project:

```
yarn install
yarn build
```

Note: The `<root-dir>` placeholder in further instructions in this document corresponds to the current working directory.

Configuration

RETRIEVE TEMPORARY IAM CREDENTIALS VIA AWS IDENTITY CENTER

Prior to running the upgrade scripts, you will need temporary IAM credentials in order to run the script. In order to retrieve these, follow the instructions here and set the temporary credentials in your environment: https://aws.amazon.com/blogs/security/aws-single-sign-on-now-enables-command-line-interface-access-for-aws-accounts-using-corporate-credentials/

CREATE UPGRADE TOOL CONFIGURATION FILE AND PREPARE ENVIRONMENT

Creates the configuration file used by the upgrade tool. The configuration file will be created in the directory `<root-dir>/src/input-config/input-config.json`.

```
cd <root-dir>
yarn run migration-config
```

Info

By default the upgrade tool uses `ca-central-1` as the home region. If you use a different home region you need to set the AWS_REGION environment variable before running `migration-config`. e.g. `AWS_REGION=eu-west-1 yarn run migration-config`

Detailed information

This command will also deploy a CloudFormation template and create two CodeCommit repositories. The CloudFormation template will create an S3 bucket for the resource mapping files. The first CodeCommit repository will also be used for the resource mapping files. The second CodeCommit repository will be used for the Landing Zone Accelerator configuration files that will be created in a later step.

To skip the creation of these resources and only generate the local configuration file, you can use the `local-update-only` argument.

```
yarn run migration-config local-update-only
```

CONFIRM OUTPUTS

Navigate to `<rootDir>/src/input-config/input-config.json` and confirm the file has been generated with values corresponding to your environment. It is not expected that these values will need to be modified.

Two CodeCommit repositories have been created

- `<prefix-name>-Mappings` to store resource mapping
- `<prefix-name>-LZA-config` to store LZA configuration

Detailed documentation of input-config.json

- `aseaPrefix` : The ASEA prefix used for ASEA deployed resources. This can be found in the initial ASEA Installer CloudFormation template `Parameters` under `AcceleratorPrefix` . Ex: `ASEA-`
- `acceleratorName` : The ASEA accelerator name. This can be found as a parameter in the initial ASEA Installer CloudFormation template.
- `repositoryName` : The ASEA Repository name used to store ASEA Configuration files. This can be found either in the initial ASEA Installer CloudFormation template `Parameters` under `ConfigRepositoryName` or in the CodeCommit Service.
- `assumeRoleName` : The name of the role which will be assumed during the upgrade process. Ex: `<prefix-name>-PipelineRole`
- `parametersTableName` : The name of the DynamoDB Table where ASEA account metadata is stored. This can be found by:
- Navigating to the DynamoDB service home page
- Selecting `Tables` from the drop down on the left side of the console.
- Finding the table name similar to `<prefix-name>-Parameters` .
- `homeRegion` : Home Region for ASEA. This field can be retrieved from the ASEA Configuration file
- `mappingBucketName` : Name of the S3 bucket to write the mapping output to. Ex: `asea-lza-resource-mapping-<management-account-id>`
- `aseaConfigBucketName` : Name of ASEA created phase-0 central bucket, will be used to copy and convert assets for LZA.
- `operationsAccountId` : Operations Account Id.
- `installerStackName` : The name of the ASEA installer CloudFormation stack.
- `centralBucket` : The name of the ASEA Phase 0 configuration bucket. Ex: `asea-management-phase0-configcentral1-ocqiyas45i27`
- `mappingRepositoryName` : The name of the CodeCommit repository resource mapping repository. Ex. `ASEA-Mappings` . Do not modify this value.
- `lzaConfigRepositoryName` : The name of the CodeCommit repository that will store the LZA configuration files. Ex. `ASEA-LZA-config` . Do not modify this value.
- `lzaCodeRepositorySource` : This value will be used when deploying the LZA installer CloudFormation stack. Ex. `github`
- `lzaCodeRepositoryOwner` : This value will be used when deploying the LZA installer CloudFormation stack. Ex. `awslabs`
- `lzaCodeRepositoryName` : This value will be used when deploying the LZA installer CloudFormation stack. Ex. `landing-zone-accelerator-on-aws`
- `lzaCodeRepositoryBranch` : This value will be used when deploying the LZA installer CloudFormation stack. Ex. `release/v1.11.0`
- `managementAccountEmail` : This value will be used when deploying the LZA installer CloudFormation stack.
- `logArchiveAccountEmail` : This value will be used when deploying the LZA installer CloudFormation stack.
- `auditAccountEmail` : This value will be used when deploying the LZA installer CloudFormation stack.
- `controlTowerEnabled` : This value will be used when deploying the LZA installer CloudFormation stack. Possible values `Yes` or `No`

3.3.3 Resource Mapping and Drift Detection Scripts

> **Warning**
>
> When ready to apply the upgrade you will need to re-run the resource mapping. Or if you make changes to ASEA resources to fix drifted resources.

OVERVIEW

The Resource Mapping script will generate the ASEA mapping file which will be used throughout the ASEA to LZA Upgrade process. In order to accomplish this task, the script needs to do the following:

- Ensure that the S3 Bucket exists and has proper object versioning enabled
- Retrieve all ASEA Enabled Regions from the ASEA Configuration File.
- Retrieve all ASEA Enabled Accounts from the ASEA Parameters Table.
- Assume a role into each account and create a unique AWS CloudFormation client for each environment (region/account combination). For each unique environment:
- List every CloudFormation Template associated with ASEA (This is a filtered down list operation)
- List every Resource that is associated with the CloudFormation Template.
- Detect Drift on each individual resource
- The outputs of these will be saved in the S3 Bucket.

RESOURCE MAPPING COMMANDS

```
cd <root-dir>
yarn run resource-mapping
```

CONFIRM RESOURCE MAPPING OUTPUTS

After running the `resource-mapping` script, the following artifacts should be generated inside the S3 bucket which has been deployed via CloudFormation and passed in the config file as `mappingBucketName`.

- Drift Detection File (per account/per region/per stack)
- Stack Resource File (per account/per region/per stack)
- Aggregate Drift Detection File (All drifted resources)

The file `AllDriftDetectedResources.csv` contains an aggregate of resources that have drifted from their original configuration. Review the next section of this guide to analyze and handle the drift results

In order to validate the output artifacts, you should verify that the following files have been created inside the S3 Bucket (*Output-Mapping-Bucket*).

Detailed information for drift files

- Resource Mapping File
- Look for file which matches *Output-Mapping-File-Name* from configuration file.
- Aggregated Drift Detection File
- Look for a file named `AllDriftDetectedResources.csv`
- See Further instructions on analyzing the drift results
- Drift Detection Files
- For a more granular look at Drift Detection, this is available on an account/region/stack basis as well:
- Navigate to `migration/<account-name>/<region>/<stack-name>/<stack-name>-drift-detection.csv`
- See Further instructions on analyzing the drift results
- Stack Resource List Output
- For each Account, Region, and Stack:
- Navigate to `migration/<account-name>/<region>/<stack-name>/<stack-name>-resources.csv`

CUSTOM RESOURCE DRIFT DETECTION

Custom Resource Drift Detection Overview

The above section covers Drift Detection on CloudFormation native resources. However, ASEA and LZA both utilize many Lambda-backed custom-resources as well. To successfully detect drift during the upgrade process, there is a snapshot tool that records the state of custom resources. The snapshot tool supports the following commands:

- yarn run snapshot pre
- yarn run snapshot post
- yarn run snapshot report
- yarn run snapshot reset

Snapshots will be taken before the upgrade to collect information that will be available for future troubleshooting. Optionally you can capture the snapshot after the upgrade as well.

Detailed information about snapshot commands

Each subcommand of the snapshot tool and its associated actions can be found below:

- `yarn run snapshot pre` - This command should be run `before` the upgrade process. Describes all custom resource states before the upgrade and saves the results in `${aseaPrefix}-config-snapshot`
- `yarn run snapshot post` - This command should be run `after` the upgrade process. Describes all custom resource states after the upgrade and saves the results in `${aseaPrefix}-config-snapshot`
- `yarn run snapshot report` - This command should be run `after` the pre and post snapshot commands have been run. Runs a diff on the Pre and Post snapshot resources and outputs a list of the diffs.
- `yarn run snapshot reset` - Deletes the DynamoDB table `${aseaPrefix}-config-snapshot`

In order to do this, the tool does the following:

- Creates DynamoDB table in the `${homeRegion}` to store snapshot data. The table is named `${aseaPrefix}-config-snapshot`:
- Assume a role into each account and makes AWS api calls to describe the state of each service managed by a custom resources. In each account/region:
- For each custom resource type, retrieve associated AWS resource, attributes, and state
- The data will then be stored in the DynamoDB table with the following fields:
- `AccountRegion` - `${AccountKey}:${Region}` key to identify what account and region the resource lives in
- `ResourceName` - Custom Resource Id
- `PreMigrationJson` (Created after snapshot pre) - This field contains the metadata and state of the resource(s) associated with the Custom Resource prior to the upgrade.
- `PreMigrationHash` (Created after snapshot pre) - This field contains a hashed value of the pre-upgrade json.
- `PostMigrationJson` (Created after snapshot post) - This field contains the metadata and state of the resource(s) associated with the Custom Resource after the upgrade is complete.
- `PostMigrationHash` (Created after snapshot post) - This field contains a hashed value of the post-upgrade json.

Custom Resource Drift Detection Commands

```
cd <root-dir>
yarn run snapshot pre
```

Custom Resource Drift Detection Outputs

In order to validate the snapshot behaviors, you will need to do the following:

- Navigate to `DynamoDB` in the AWS console.
- Click on `Tables` on the left side of the page.
- On the `Tables` page, select the radio-button next to the table `${aseaPrefix}-config-snapshot`
- Once you have selected the radio-button, click on the `Explore Table Items` button in the top right.
- This table should be populated with the following fields:
- AccountRegion
- ResourceName
- PreMigrationJson
- PreMigrationHash

3.3.4 Handling Drift from Resource Mapping

After executing the Resource Mapping and Drift Detection Scripts you need to check the status of drifted resources and properly handle each case.

At the root of the **Mapping Output Bucket** a file named `AllDriftDetectedResources.csv` is created with a summary of all ASEA resources that have drifted. Download this file and inspect each row.

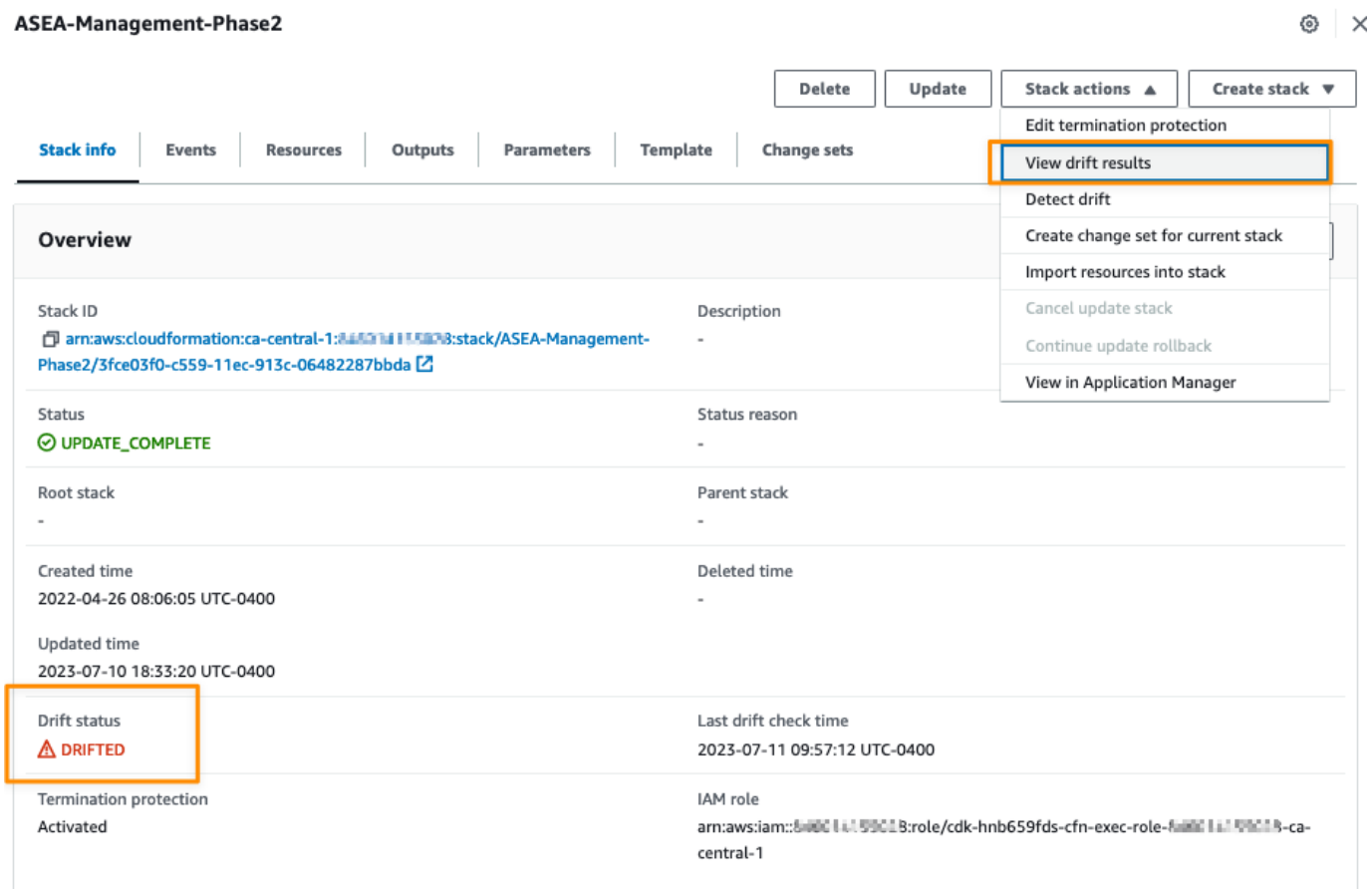
For a more granular look at Drift Detection, this is available on an account/region/stack basis as well:

- Navigate to `migration/<account-name>/<region>/<stack-name>/<stack-name>-drift-detection.csv`
- The possible values for the resources are:
- IN_SYNC - there is no drift detected in the CloudFormation Resource
- MODIFIED - drift has been detected in the CloudFormation Resource. The metadata in the `PropertyDifferences` column describes the drift that needs to be fixed.
- NOT_SUPPORTED means that CloudFormation does not support drift-detection on that specific resource.
- If there is drift detected, this drift needs to be manually fixed. The specific resource and configurations which need to be addressed will be available in the drift-detection.csv file under `PropertyDifferences` or by Detecting Drift manually in the CloudFormation console (https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/detect-drift-stack.html)

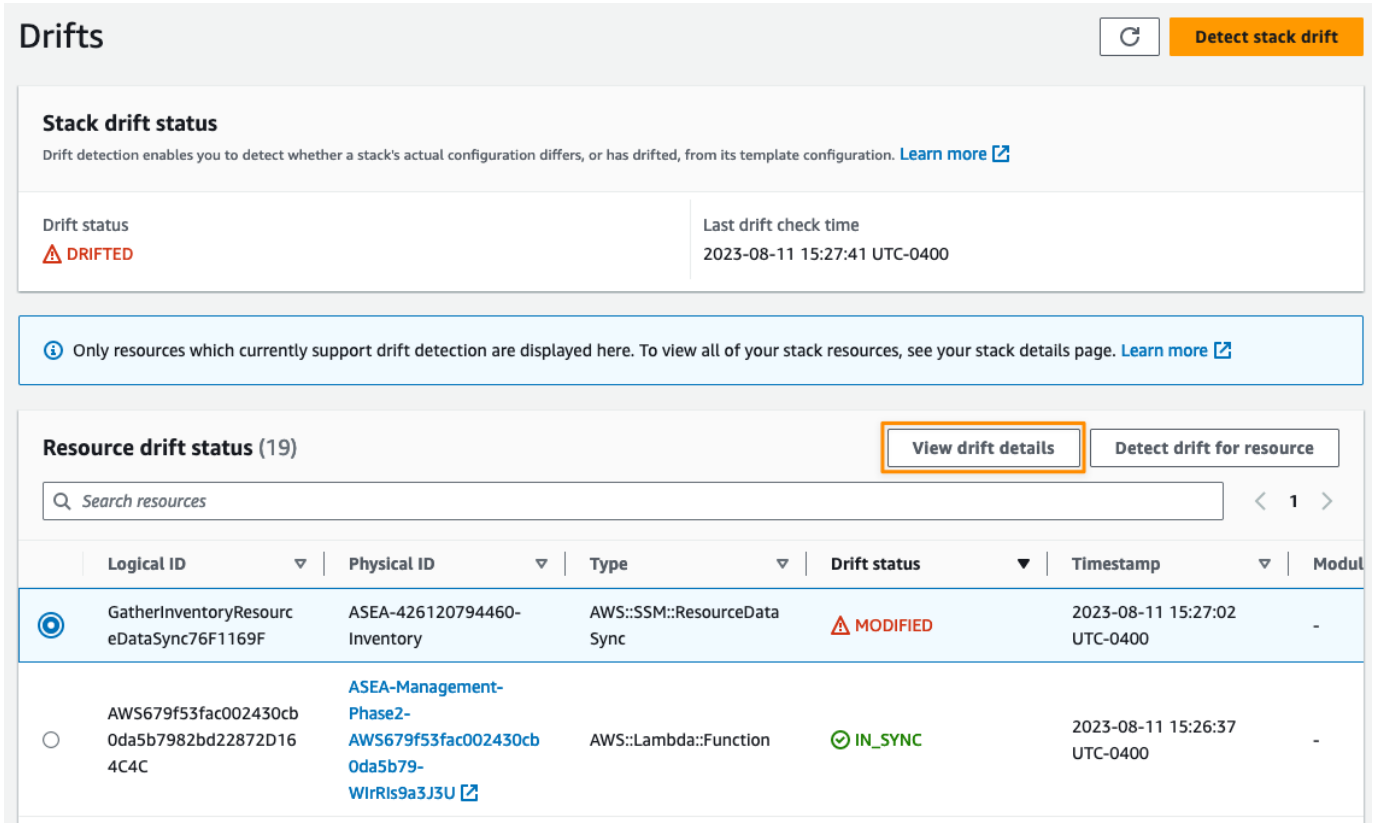
INSPECT THE DRIFTED RESOURCES

You can find more details about each occurrence by inspecting the drift results in AWS Console.

- Login to your Management Account
- Use the **Switch role** feature to switch to the target account using the management role (i.e. `ASEA-PipelineRole`)
- Go to the CloudFormation console and find the relevant stack
- The **Drift status** should be **Drifted**. In the **Stack actions** menu, select **View drift results**



• Find the resources in the **Modified** state, select the radio button on the left and click **View drift details**



• The next screen will show the detailed differences. For most resources you will also have a link to open the resource in the console

ANALYZE AND FIX THE DRIFTED RESOURCES

Each change should be analyzed, confirm if it is expected and why and determine if a corrective action is needed

- Some changes are expected as part of ASEA operations and can be safely ignored (see next section)
- If a change was manually done outside of the ASEA config file:
- If it could have been done through the config file, you should update the config file accordingly and re-run the state machine to remove the drift
- If the change was done manually because it cannot be done through ASEA (i.e. Direct Connect configuration, adding rules to ALB listeners, etc.) you should document the change in a central registry. Special attention should be given to those elements during the upgrade and in the post-upgrade testing phase.

> **Warning**
>
> Failure to detect and fix drift before the upgrade can result in the configuration of resources to be reverted to the state described in the configuration file.

Pay special attention to drift on networking resources such as route tables and security groups. Drift detection will only detect drift on resources deployed by ASEA and not all resource types support CloudFormation drift detection. (see the Resource type support table). Review the Considerations when detecting drift section to understand edge cases where CloudFormation may not be able to always return accurate drift results.

As an example, drift detection is not supported for resource type `AWS::EC2::SubnetRouteTableAssociation`. If you created route tables outside the accelerator and manually associated these route tables with an ASEA subnet, this change won't be detected in drift detection and the route table association will be reverted to the one defined in the configuration when installing LZA.

> **Tip**
>
> There is a script available to help detect drift on networking resources that are not detected my CloudFormation. The script is available in the tools/network-drift-detection folder in the ASEA to LZA upgrade tools.

> **Warning**
>
> Remember that drift detection is a tool to help you identify manual changes done outside the accelerator, but it won't identify every change. Carefully review and analyze ALL changes done to accelerator resources outside of the configuration file.

Expected drift (can generally be ignored)

Some of the resources deployed by ASEA are modified by other mechanisms in the normal course of operations of the accelerator (e.g. the `EC2-INSTANCE-PROFILE-PERMISSIONS` Config Rule that dynamically attaches permissions to IAM instance profiles roles). These resources will show as drifted, but they can be safely ignored. A `CanIgnore` flag in `AllDriftDetectedResources.csv` can help you identify drift instances that can generally be safely ignored.

Stack	LogicalResourceID	Notes
Multiple	GatherInventoryResourceDataSync	Multiple occurrence of this finding can be reported in multiple accounts and regions
ASEA-LogArchive-Phase0	CWLKinesisStreamRole	Inline policy dynamically added to role
ASEA-Operations-Phase1	IAMAssetsOperationsIAMRoleASEARsyslogRole	Inline policy dynamically added to role
ASEA-Perimeter-Phase1-VpcStackPerimeterNestedStack...	PerimeterTgwAttach	Difference in tags
ASEA-SharedNetwork-Phase1-VpcStack...	*TgwAttach	Difference in tags
ASEA-SharedNetwork-Phase2	FlowLog[VPC]cloudwatchlogs	One occurrence per VPC. Difference in tags
ASEA-SharedNetwork-Phase2-VpcEndpoints1	EndpointEndpoint...	Private hosted zone for interface endpoints are shared to additional VPCs
ASEA-SharedNetwork-Phase3	*private domain name*	Private hosted zone for interface endpoints are shared to additional VPCs

In addition, drift on Load Balancer resources (i.e. `AWS::ElasticLoadBalancingV2::Listener`, `AWS::ElasticLoadBalancingV2::LoadBalancer`) can also be ignored as the LZA upgrade won't modify Application Load Balancers.

Note about tags and drift

ASEA uses CloudFormation stack-level tags to apply tags to all supported resources in a stack. Tags applied at stack-level can generate false positives on drift detection. You can review the column `PropertyDifferencesPaths` from the `AllDriftDetectedResources.csv` file to verify the properties that have drifted to confirm if only tags are drifted on the resource.

3.3.5 Convert Configuration

Convert Configuration Overview

In order to accomplish the upgrade, the existing ASEA configuration file needs to be converted into LZA configuration files (https://docs.aws.amazon.com/solutions/latest/landing-zone-accelerator-on-aws/using-configuration-files.html). The `convert-config` script parses through the ASEA configuration file and for each resource block does the following:

- Reads in the ASEA configuration object
- Decides the ASEA Object Type
- Maps object and resource metadata file to LZA Object
- Creates proper Deployment Targets for the LZA Object (This defines which accounts the resource will be deployed to)
- Once the entire ASEA configuration file has been converted, the output LZA configuration files will be stored locally in the current directory in a sub-directory named `outputs\lza-config`. The files will also be created in the CodeCommit repository name `<prefix-name>-LZA-config`

Convert Configuration Commands

```
cd <root-dir>
yarn run convert-config
```

Option to generate files locally only

If you used the `local-update-only` in the configuration step, you should also use the `local-update-only` with the convert-config command to generate the files locally only as the CodeCommit repo wasn't created. This can be useful in your early preparation phase to validate the generated configuration without impacting your environment.

```
yarn run convert-config local-update-only
```

Option to enable termination protection

By default the tool sets termination protection to false on CloudFormation stacks to facilitate troubleshooting and retries in case of errors. It is recommended to enable this feature through the LZA global configuration file after the initial LZA pipeline run is successful. The `enable-termination-protection` flag can be used to enable termination protection for the LZA deployed stacks in the initial installation.

```
yarn run convert-config enable-termination-protection
```

Tip

If an ASEA account resides in an Organizational Unit which is in the `ignored-ous` section of `global-config` block, that account will not be added to the resulting `accounts-config.yaml` output file. This is due to the way that the LZA handles accounts which it manages as well as logic in the config validator.

Confirm Convert Configuration Outputs

After running the `convert-config` script, the following artifacts should be generated in the current directory in a subdirectory named `outputs/lza-config` and in the CodeCommit repository named `<prefix-name>-LZA-config`:

- Configuration Files
- `accounts-config.yaml`
- `global-config.yaml`
- `iam-config.yaml`
- `network-config.yaml`
- `organization-config.yaml`
- `security-config.yaml`
- Dynamic Partitioning preferences
- `dynamic-partitioning/log-filters.json`
- IAM Policies
- `iam-policies/*`
- Service Control Policies (SCPs)
- `service-control-policies/*`
- SSM Documents
- `ssm-documents/*`

3.3.6 Pre-upgrade validations

The Landing Zone Accelerator has tools that can be used to validate the configuration locally. This can help catch errors locally before applying the upgrade in the actual AWS environment.

Obtain and build the Landing Zone Accelerator code

To run those tools you need to download and build the Landing Zone Accelerator code.

The following commands should be run in a dedicated folder, outside of the current folder with the upgrade scripts, to store the LZA code base (referred as `<lza-code>` in instructions)

```
cd <lza-code>
git clone https://github.com/awslabs/landing-zone-accelerator-on-aws/
cd source
yarn install
yarn build
```

To run the next commands you need to confirm you have valid temporary credentials to your management account as mentioned at the beginning of this guide.

Validating LZA configuration files

LZA has a tool to validate your configuration files. We strongly recommend you run this tool on the generated LZA configuration file to spot any errors.

See Configuration Validator section in the LZA developer guide for more details.

To run the configuration validation, run the following commands from the LZA source directory by passing the path to the LZA config file as an argument.

```
cd <lza-code>/source
yarn validate-config <root-dir>/outputs/lza-config
```

Validate Service Control Policies size

The upgrade to LZA generally does not modify your current SCP statements. The only exception is that the `organization-admin-role` can be added to the `SSM` and `S3` statements of the `Guardrails-Part-0` and `Guardrails-Part-1` SCPs if it is not already there. This can potentially bring those SCP over the limit if the existing content was close to the limit.

We recommend that you verify the number of characters of all SCP files to confirm they are not over the 5120 characters limit. You can run the following command from within the `outputs/lza-config` folder to print the size of each SCP file.

```
for FILE in service-control-policies/*; do echo -n $FILE; echo -n ' '; cat $FILE | sed -e 's/[[:space:]]//g' |  tr -d '\r' | tr -d '\n' | wc -c; done
```

Validation complete

You have successfully validated the configuration and the preparation steps.

> **Warning**
>
> Stop here if you are not ready to proceed with the ASEA to LZA upgrade. Otherwise move to the next section to start the upgrade.

3.4 Upgrade

3.4.1 ASEA to LZA Upgrade

Before starting the upgrade process you need to make sure you recently went through all the preparation steps and have a copy of your LZA configurations files in the CodeCommit repository named `<prefix-name>-LZA-config`.

Re-confirm pre-requisites

- Confirm you are on the latest ASEA version and that the last state machine execution was successful.
- Confirm all suspended accounts are under a specific OU that is ignored by the accelerator. (see Suspended accounts)
- Confirm you don't have any empty nested OU without active AWS Accounts that are not referenced from the ASEA configuration files (i.e. `Dev/nestedOU`). The convert-config tool won't generate empty nested OUs in the configuration.

> **Warning**
>
> The following steps will start applying changes to your environment by uninstalling ASEA and installing LZA. Only move ahead when ready to go through the full upgrade.

The upgrade steps are

- Upgrade

a. Optional preparation steps

b. Disable ASEA

c. Install LZA

d. Finalize the upgrade

3.4.2 Optional preparation steps

Additional preparation steps are recommended depending on your configuration

Configure Interface Endpoints for S3 and DynamoDB

CONTEXT

During the upgrade process, LZA creates new route tables and associates them with the existing subnets to replace the previous ASEA route tables. This is mostly transparent as the LZA route tables are identical to the ASEA route tables defined in the ASEA configuration. However, the routes pointing to the prefix list for Gateway Endpoints (S3 and DynamoDB) are only added at a later stage of the upgrade process. Therefore the Gateway Endpoints won't be available from your VPCs between the NetworkVPC stage and PostImportASEAResources stage of the LZA installation. Communication to S3 and DDB will fall back to using the public endpoints going through your Perimeter VPC using the default route. This traffic will be allowed or denied based on your egress rules in the perimeter firewall.

WORKAROUND

If your workloads cannot tolerate a communication disruption to S3 and DynamoDB, or if they require communication through a Private Endpoint, we recommend temporarily deploying Interface Endpoints for the duration of the upgrade.

Warning

Gateway endpoints are offered at no cost. Interface endpoints have an hourly cost and data transferred through the interface endpoint is charged. This is why we recommended only deploying the S3 and DynamoDB interface endpoint as a temporary measure during the upgrade.

Prior to executing the LZA upgrade

- In the Shared Networking account, create Interface Endpoints for S3 and DynamoDB in the Endpoint VPC.
- Create a security group that allows HTTPS from anywhere (0.0.0.0/0)
- For S3
- Do not select the option "Enable DNS Name"
- Select the security group previously created
- For DynamoDB
- Do not select the option "Enable DNS Name"
- Select the security group previously created
- Go to **Route 53** and create a Private Hosted Zones for the endpoints.
- For S3
- Domain name: s3.ca-central-1.amazonaws.com (adjust as needed based on your region)
- Type: Private hosted zone
- VPCs to associate with the hosted zone: Select only the Endpoint_vpc for now
- Add records to the Private Hosted Zone
- 1) Create top-level A record
- Subdomain: (Leave empty)
- Record Type: A
- Alias: Selected
- Route traffic to: Alias to VPC Endpoint
- Select the S3 endpoint previously created
- 2) Create wildcard A record
- Subdomain: *
- Record Type: A
- Alias: Selected
- Route traffic to: Alias to VPC Endpoint
- Select the S3 endpoint previously created
- Once the record is created, edit the hosted zone and associate it with all your VPC (Dev, Test, Prod, Central)
- For DynamoDB
- Domain name: dynamodb.ca-central-1.amazonaws.com (adjust as needed based on your region)
- Type: Private hosted zone
- VPCs to associate with the hosted zone: Select only the Endpoint_vpc for now
- Add record to the Private Hosted Zone
- Subdomain: (Leave empty)
- Record Type: A
- Alias: Selected
- Route traffic to: Alias to VPC Endpoint
- Select the DynamoDB endpoint previously created
- Once the record is created, edit the hosted zone and associate it with all your VPC (Dev, Test, Prod, Central)

If you have VPCs deployed locally in workload accounts outside of the shared-network account (i.e. Spoke VPC topology) you will need to create this association using the AWS CLI, SDK or API. Refer to the documentation on Associating an Amazon VPC and a private hosted zone that you created with different AWS accounts

Removal of endpoints after the LZA installation

Once LZA upgrade is complete

- Confirm that Gateway Endpoints are associated with the route tables of your subnets
- Remove the S3 and DynamoDB Private Hosted Zones and Interface Endpoints that were previously created by doing the steps in reverse order:
- Un-associate the Private Hosted Zone from all VPCs except Endpoint_vpc
- Remove all record from the zone except the SOA and NS records
- Delete the Private Hosted Zone
- Delete the Interface endpoint (don't delete the Gateway endpoints)

Disable Security Hub forwarding to CloudWatch Log Groups

ASEA uses an Event Bridge rule and a Lambda function to forward all Security Hub findings to a CloudWatch Log Group in the Security Audit account. The centralized logging architecture then forward all the CloudWatch Log entries to the central S3 bucket. During the LZA installation, a LZA specific Event Bridge rule will be deployed to achieve the same outcome. The LZA rule directly targets the CloudWatch Log Group without a Lambda, the process is thus more efficient.

We recommend disabling the Event Bridge rule **before** the LZA installation to avoid duplicate findings being delivered. On large environments, timeout issues related to Lambda rate limiting have been reported during the upgrade.

If you require all findings to be logged in CloudWatch Logs and S3 we recommend you instead disable the rule **after** the LZA installation, be advised that you will see duplicate findings being delivered. In all cases, Security Hub findings will continue to be available in the Security Hub console and through SNS Topics notifications if they are configured, this only affect the delivery of the findings to CloudWatch and S3.

DISABLE THE EVENT BRIDGE RULE

1. Login to your Management account using an administrative role
2. Assume the privileged role (i.e. `{prefix-name}-PipelineRole`) into the Security Audit account
3. Go to the Event Bridge console in the Rules page
4. Locate the `{prefix-name}-SecurityHubFindingsImportToCWLs` rule
5. Disable the rule
6. Repeat this for every AWS Region enabled in your configuration file

Alternatively you can run the following command using AWS Cloud Shell from the Security Audit account to disable the rule in all regions (you need to use the appropriate rule name if using a different accelerator prefix)

```
for region in `aws ec2 describe-regions --query "Regions[].RegionName" --output text`; do aws events disable-rule --region $region --name ASEA-SecurityHubFindingsImportToCWLs; done
```

3.4.3 Disable and uninstall ASEA

Disable ASEA Custom Resource Delete Behaviors

To complete the upgrade process, we will need to disable ASEA Custom Resource deletions. In order to do this, we have added a new parameter called `LZAMigrationEnabled`. Setting this to true during CloudFormation stack update will enable this behavior. In order disable the resources, complete the following:

DEPLOY THE UPGRADE ASEA INSTALLER STACK

You will need to update the existing CloudFormation Installer stack:

- Download the AcceleratorInstaller stack from the latest ASEA Release on GitHub (i.e. v1.6.0 or later)
- Navigate to the AWS CloudFormation console
- Select the existing installer stack then **Update Stack**
- On the **Update Stack** page, select the radio button for:
- **Replace current template** under **Prepare Template Section**`
- Click **Next**
- **Upload a Template File** under **Specify Template Section**
- Select **Choose File** and navigate to the file downloaded from GitHub release page
- Click **Next**
- On the **Specify Stack Details** in the Parameters section update only the parameter named `LZAMigrationEnabled`. Change the value to `true`.
- Update the parameter named `RepositoryBranch`. Change the value to the latest ASEA release (e.g. `release/v1.6.0`)
- Click **Next**
- On the **Configure Stack Options** don't make any changes.
- Click **Next**
- On the **Review**
- In **Capabilities** section, select the box **I acknowledge the AWS CloudFormation might create IAM resources with custom names.**
- Click **Next**
- Wait for the stack to finish updating

Execute the ASEA installer pipeline and state machine

- Navigate to AWS CodePipeline console
- Locate the ASEA-InstallerPipeline under the Pipeline/Pipelines section
- Select the pipeline and then click on **Release change**
- Wait for the pipeline execution to complete
- The last step of the pipeline will start the ASEA main state machine
- Monitor the progress of the main state machine
- Navigate to the AWS Step Function console
- The `ASEA-MainStateMachine_sm` should be running
- Wait until the `ASEA-MainStateMachine_sm` is finished before moving to the next section

Re-run resource mapping script

When the `ASEA-MainStateMachine_sm` has completed successfully, re-run the resource mapping script.

```
cd <root-dir>
yarn run resource-mapping
```

Prepare ASEA Environment

PREPARE ASEA ENVIRONMENT OVERVIEW

This step will prepare the ASEA environment for upgrade to the Landing Zone Accelerator on AWS. In this step the upgrade scripts tool will delete the CDK Toolkit CloudFormation stacks in the Management account. Which includes deleting ECR images from the CDK Toolkit ECR repository. Deleting the ASEA CloudFormation installer stack and finally the ASEA InitialSetup stack. You will also be emptying the ASEA artifacts bucket in order for the installer CloudFormation stack to be deleted. In order to empty the artifacts S3 bucket you will need to navigate to S3 console.

- Find the bucket that has the string `artifactsbucket` in the name
- Click the radio button next to the bucket
- Click the **Empty** button in the upper right
- Type the string **permanently delete** in the confirmation text box
- Click the **Empty** button
- Wait until a green bar appears with the text **Successfully emptied bucket**
- Switch back to your CLI environment and run the commands below

PREPARE ASEA ENVIRONMENT COMMANDS

```
cd <root-dir>
yarn run asea-prep
```

Wait until the commands complete.

3.4.4 Installing the Landing Zone Accelerator

> **Warning**
>
> Once LZA is installed and the LZA pipeline has run, rollback to ASEA won't be possible anymore. Make sure you are ready to proceed and that you executed all the recommended preparation steps.

Installing the LZA Pipeline

You are ready to deploy AWS Landing Zone Accelerator. This step will deploy a CloudFormation template, creates two AWS CodePipeline pipelines, an installer and the core deployment pipeline along with associated dependencies. This solution uses AWS CodeBuild to build and deploy a series of CDK-based CloudFormation stacks that are responsible for deploying supported resources in the multi-account, multi-Region environment. The CloudFormation template will first create the `${prefix-name}-Installer`, which in turn will create the accelerator pipeline, `${prefix-name}-Pipeline`

- For more details on the deployment pipelines, take a look here: https://docs.aws.amazon.com/solutions/latest/landing-zone-accelerator-on-aws/deployment-pipelines.html

INSTALLING THE LZA PIPELINE COMMANDS

```
cd <root-dir>
yarn run lza-prep
```

Installing the LZA Pipeline Confirmation

Navigate to the AWS CloudFormation console and confirm that the stack named `<prefix-name>-Installer` deployed successfully.

Run the LZA Pipeline

- For general LZA Pipeline deployment details, refer to the LZA Implementation Guide here: https://docs.aws.amazon.com/solutions/latest/landing-zone-accelerator-on-aws/awsaccelerator-pipeline.html
- During the Landing Zone Accelerator pipeline deployment, there are two ASEA upgrade specific stages `ImportAseaResources` and `PostImportAseaResources`. These two stages allow the LZA to manage and interact with resources that were originally managed in the scope of ASEA. The current ASEA Resource Handlers exist in the table here: ASEA Resource Handlers.
- **ImportAseaResources**: This stage uses the `CFNInclude` module to include the original ASEA Managed CloudFormation resources. This allows the resources to be managed in the context of the LZA CDK Application. SSM Parameters are created for these resources so that they can be interacted with during the LZA Pipeline run.
- **PostImportAseaResources**: This stage runs at the end of the LZA Pipeline, it allows the LZA pipeline to modify original ASEA Managed Cloudformation resources. This requires a separate stage because it allows the prior LZA stages to interact with ASEA resources and then modifies all ASEA resources (as opposed to CFN Including the ASEA resources in every stage).

3.4.5 Finalize the upgrade

> **Warning**
>
> The following steps will delete ASEA resources that are no longer needed because they have been replaced by LZA resources. Please confirm that all resources are deployed and working as expected before proceeding with this step.

REMOVE TEMPORARY INTERFACE ENDPOINTS FOR S3 AND DYNAMODB

If you created temporary Interface Endpoints for S3 and DynamoDB in the optional preparation steps you can now remove them according to the instructions.

Post upgrade Overview

This step will perform post upgrade actions which includes following

- Copy ASEA ACM Certificate assets from ASEA Central Bucket to LZA created Assets bucket. `copy-certificates`
- Delete Outputs from ASEA stacks. `remove-stack-outputs`
- Marks duplicate SNS Topics, Subscriptions and Policies for removal. `remove-sns-resources`
- Marks duplicate Config Rules and Remediation Configurations for removal. `remove-asea-config-rules`
- Marks duplicate RSyslog resources for removal. `remove-rsyslog`
- Marks duplicate CloudWatch Alarm resources for removal. `remove-cloudwatch-alarms`
- Marks duplicate CloudWatch Metrics resources for removal. `remove-cloudwatch-metrics`
- Marks duplicate Budget resources for removal. `remove-budgets`
- Marks duplicate logging resources for removal. `remove-logging`

Each of the above steps has a corresponding flag that can be set during the post-migration step. These flags determine which actions are performed by the post-migration step.

Post upgrade Commands

```
cd <root-dir>
yarn run post-migration remove-stack-outputs copy-certificates remove-sns-resources remove-asea-config-rules remove-cloudwatch-alarms remove-cloudwatch-metrics
remove-budgets remove-logging
```

After the commands has been run, go the the CodePipeline console and release the `ASEA-Pipeline`. Resources that have been flagged for removal will be deleted in the `ImportAseaResources` stage.

Enabling Termination Protection on CloudFormation stacks

During the initial LZA installation, termination protection was set to false on CloudFormation stacks to facilitate troubleshooting and retries in case of errors. Now that LZA is installed we recommend that customers enable termination protection on all LZA stacks.

Change the setting in the `global-config.yaml` file and run the LZA pipeline.

```
terminationProtection: true
```

Upgrade complete

At this point the upgrade to LZA is complete. Further updates to the environment will require updating the LZA configuration and then executing the LZA pipeline.

Review the section Feature specific considerations for further steps that may be needed based on your configuration.

3.5 FAQ and Troubleshooting

3.5.1 FAQ

Does the upgrade affect availability of the workloads in the Landing Zone?

The upgrade is designed to be as transparent and automated as possible. Only resources initially deployed by ASEA are touched during the upgrade, any resources deployed outside the accelerator and in workloads accounts are not impacted during the upgrade.

Changes to the shared networking resources managed by the accelerator may have an impact on the availability of workloads using the shared VPCs and perimeter resources. During the upgrade process, LZA creates new route tables and NACLs and associates them with the existing subnets to replace the previous ASEA route tables and NACLs. We recommend customers plan for a one to two minute network disruption for the traffic going through the Perimeter VPC.

- The LZA route table and NACLs are re-created based on the ASEA configuration. It is critical to identify drift or any manual modifications done to these resources prior to the upgrade.
- When the route tables are replaced in the NetworkVPC stage of the LZA installation, minimal packet loss (i.e. few seconds) can be observed. This affects all traffic going through the Transit Gateway.
- For deployments using AWS Network Firewall, the routes targeting the network firewall endpoints are re-created in the NetworkVpcEndpointsStack that is deployed immediately after the NetworkVPCStack. This causes a network disruption of all ingress/egress traffic going through the Perimeter VPC between 1 and 2 minutes.
- For deployments using third-party Firewalls (i.e. FortiGate), the routes targeting the firewall ENIs are re-created in the NetworkAssociationsGwlbStack. This doesn't affect workload traffic flowing through the firewalls but can impact connectivity to the firewall management interface.
- There is a period between the **NetworkVPC** and **PostImportASEAResources** stages where route tables to VPC Gateway Endpoints for S3 and DynamoDB are not available. See the section on Optional preparation steps for more details and recommended workaround.

What if we made manual changes to subnet route tables outside the accelerator?

As detailed in the previous entry, LZA creates new route tables and NACLs based on the ASEA configuration and associates them with the existing subnets to replace the ASEA route tables and NACLs. Any changes made to subnet route tables outside the accelerator will be reverted during the upgrade.

The preferred resolution is to align the ASEA configuration to incorporate the manual changes in ASEA before the upgrade to remove the drift.

If this is not possible, you should record all route table information before the upgrade to identify manually created entries. After the upgrade is complete, these entries need to be recreated.

Note: Transit Gateway route tables are not replaced during the upgrade, these guidelines only apply to subnet route tables.

There is a script available to help detect drift on networking resources that are not detected by CloudFormation. The script is available in the tools/network-drift-detection folder in the ASEA to LZA upgrade tools.

Gateway Load Balancer are not supported in the configuration conversion, how will this impact the workload availability?

As covered in the Feature specific considerations section, the configuration tool will not map the existing GWLB in ASEA to the LZA configuration. The already deployed firewall instances and Gateway Load Balancer endpoints will remain untouched. However, you should carefully review the route tables to confirm if the entries sending traffic to the GWLB Endpoints will be properly configured when LZA re-creates the subnet route tables.

Review the related route table entries in the network-config.yaml file and compare the entries with the entries of the route tables currently deployed in your environment. Make the necessary modifications in network-config.yaml to match the current configuration. Reference the RouteTableEntryConfig documentation for more details on how to configure route tables in LZA.

In this scenario, you won't be able to use the `gatewayLoadBalancerEndpoint` destination type to reference a GWLB that was not deployed by LZA. As an alternative you can use the `networkInterface` type to directly reference the ENIs of the GWLB endpoints. Please note, that the LZA route tables are created and associated in the NetworkVPC stack, but the route entries of type `networkInterface` are created in the NetworkAssociations stack which is triggered later in the pipeline, resulting in a period of time where the route entries will be missing.

Another alternative is to manually add the missing route entries to the LZA route tables as soon as the NetworkVPC stage completes to minimize network downtime.

Do you recommend having specific monitoring in place for the upgrade?

We highly recommend that you monitor the availability of all your workloads deployed in the Landing Zone as well as the main network flows to be alerted of any impact during the upgrade process.

Example of important network flows to monitor:

- Ingress traffic from Internet to workloads through Perimeter
- Egress traffic from workloads to Internet through Perimeter
- Ingress and egress traffic between on-premises networks and AWS VPCs (i.e. Direct Connect or VPN)
- East-West traffic between your VPCs through Transit Gateway

You can use CloudWatch Network Synthetic Monitor and CloudWatch Synthetics (Canaries) in combination with CloudWatch Alarms to setup monitoring of the important network flows and of your applications.

Why does CloudTrail configuration show as disabled in the LZA configuration files?

The convert-config tool generates a configuration block in `global-config.yaml` with CloudTrail showed as disabled.

```
logging:
  account: LogArchive
  centralizedLoggingRegion: ca-central-1
  cloudtrail:
    enable: false
    organizationTrail: false
```

This is because for upgraded environment, there is already an existing organizational trail configured by ASEA or ControlTower that will continue to be used. We don't recommend changing this to `true` as this will instruct LZA to create a new trail in addition to the existing one created by ASEA.

Which Service Quotas should be monitored for the upgrade?

Depending on your configuration, the LZA installation can create over 500 IAM Roles in each account. If you already have several IAM Roles in your accounts and using the default limit of 1000, the installation could be blocked by this service quota.

You can make an AWS Config query using the organization aggregator to list the current number of IAM Roles in each account, and request a limit increase proactively.

```
SELECT
  accountId,
  COUNT(*)
WHERE
  resourceType = 'AWS::IAM::Role'
GROUP BY
  accountId
ORDER BY
  COUNT(*) DESC
```

For more information about LZA related Quotas, refer to the LZA Documentation about Quotas as well as this note about CodeBuild concurrency

3.5.2 Troubleshooting

Failure in ImportASEAResourceStage

If the LZA pipeline fails in the ImportASEAResources stage and you need to restart the pipeline from the beginning, you will need to remove a file from the `asea-lza-resource-mapping-<accountId>` bucket. The name of the file is `asearesources.json`. Download a copy of the file and then delete it from the S3 bucket. The file will be recreated when the pipeline is rerun.

Failure creating new account after upgrade when using Control Tower

Error messages:

- Account creation failed error message in the Prepare stage.
- AWS Control Tower failed to deploy one or more stack set instances: StackSet Id: AWSControlTowerBP-VPC-ACCOUNT-FACTORY-V1

If you are adding a new Control Tower account, ensure that there are no regions where VPCs are automatically created when an account is provisioned. To do this:

- Navigate to the Control Tower Home Page
- Select 'Account Factory' on the left of the page
- Click the 'Edit' button on the 'Network configuration' section
- Ensure that none of the regions are selected under 'Regions for VPC Creation'

Timeout issues on large environments

When upgrading an ASEA environment with a large number of accounts (>100), you may encounter specific timeout issues and need to do manual changes to workaround the issues.

JAVASCRIPT HEAP OUT OF MEMORY ERRORS

Cause: CodeBuild does not have enough memory to synthesize very large CloudFormation stacks

Workaround: Increase the resources allocated to CodeBuild and increase NodeJS `max_old_space_size`

1. Go to CodeBuild console and locate the `ASEA-ToolkitProject` project
2. Edit the project, in the Environment section change the Compute size to the next larger size available (70 GB Memory, 36 vCPU)
3. In the Environment variables section: a) change the value of the `NODE_OPTIONS` variable to `--max_old_space_size=32768`
4. Release the accelerator pipeline again

Note: this manual change will need to be re-applied every time you upgrade to a new LZA version or re-run the LZA installer pipeline.

ERROR IN SECURITY STACK - CLOUDFORMATION DID NOT RECEIVE A RESPONSE FROM YOUR CUSTOM RESOURCE

Cause: Throttling can happen based on the concurrent Lambda execution quota.

Workaround: Disable the Event Bridge rule `ASEA-SecurityHubFindingsImportToCWLs` in the Security account.

ERROR IN SECURITYRESOURCE STACK - AWS CONFIG RATE EXCEEDED ERROR

Cause: Too many resources are deployed in parallel, leading to rate limiting errors.

Workaround: Increase the resources allocated to CodeBuild and increase NodeJS `max_old_space_size`

1. Go to CodeBuild console and locate the `ASEA-ToolkitProject` project
2. Edit the project, in the Environment variables section: a) change the value of the `MAX_CONCURRENT_STACKS` variable to `75`
3. Release the accelerator pipeline again

Note: this manual change will need to be re-applied every time you upgrade to a new LZA version or re-run the LZA installer pipeline.

CREDENTIALSPROVIDERERROR IN BOOTSTRAP STAGE

Bootstrap stage fails with the following error

```
error | utils-common-functions | {"name":"CredentialsProviderError","tryNextLink":false}
Could not load credentials from any providers
```

Workaround: Increase the **Number of retries** in the SDK configuration.

1. Go to CodeBuild console and locate the `ASEA-ToolkitProject` project
2. Edit the project, in the Environment variables section: a) add a new environment variable named `NUMBER_OF_RETRIES` b) set the value of the a higher value (default: 12)
3. Release the accelerator pipeline again

Use of opt-in region - "InvalidClientTokenId: The security token included in the request is invalid"

If an AWS opt-in region (e.g. ca-west-1) is enabled in your ASEA environment you need to change the region compatibility of STS session tokens to be valid in all AWS Regions.

1. Sign in with administrative privileges in your Management account.
2. Open the IAM console. In the navigation pane, choose Account settings.
3. Under Security Token Service (STS) section Session Tokens from the STS endpoints. The Global endpoint indicates Valid only in AWS Regions enabled by default. Choose Change.
4. In the Change region compatibility dialog box, select All AWS Regions. Then choose Save changes.

Documentation: Managing global endpoint session tokens

Network timeout or connectivity issue running the upgrade tool

To run the upgrade tool, you need to have valid credentials to your management account. The upgrade tool makes API calls to several AWS services to gather information about your configuration and create the resource mapping. It reads information from the accelerator S3 buckets, DynamoDB tables and make calls to AWS Organizations as well as AWS CloudFormation in all regions.

If running the tool from within an AWS VPC, it will use the available VPC endpoints to reach the respective service endpoints. If no VPC endpoints are available or to make calls to regions other than the home region, the pubic service endpoints will be used and you need to make sure that any egress filtering you have in place allow those calls.

If running the tool from within your corporate network, you need to make sure that any egress filtering you have in place allow those calls.

The following endpoints can be used by the `migration-config,` , `resource-mapping` and `convert-config` command of the upgrade tool. If you have configured additional `supported-regions` or use a home region other than `ca-central-1` , the list needs to be updated accordingly.

```
organizations.us-east-1.amazonaws.com
sts.amazonaws.com
sts.us-east-1.amazonaws.com
codecommit.ca-central-1.amazonaws.com
s3.ca-central-1.amazonaws.com
dynamodb.ca-central-1.amazonaws.com
kms.ca-central-1.amazonaws.com
ssm.ca-central-1.amazonaws.com
cloudformation.ca-central-1.amazonaws.com
cloudformation.ap-northeast-1.amazonaws.com
cloudformation.ap-northeast-2.amazonaws.com
cloudformation.ap-northeast-3.amazonaws.com
cloudformation.ap-south-1.amazonaws.com
cloudformation.ap-southeast-1.amazonaws.com
cloudformation.ap-southeast-2.amazonaws.com
cloudformation.eu-central-1.amazonaws.com
cloudformation.eu-north-1.amazonaws.com
cloudformation.eu-west-1.amazonaws.com
cloudformation.eu-west-2.amazonaws.com
cloudformation.eu-west-3.amazonaws.com
cloudformation.sa-east-1.amazonaws.com
cloudformation.us-east-1.amazonaws.com
cloudformation.us-east-2.amazonaws.com
cloudformation.us-west-1.amazonaws.com
cloudformation.us-west-2.amazonaws.com
```

Different S3 buckets deployed by the accelerator are accessed by the tool, those calls will be made using the `<bucket-name>.s3.ca-central-1.amazonaws.com` endpoint.

Security stack failure during LZA pipeline run after adding an opt-in region

You encounter the following error during an LZA pipeline run after adding an opt-in region such as `ca-west-1` to your enabled regions.

The stack named ASEA-SecurityStack--ca-west-1 failed creation, it may need to be manually deleted from the AWS console: ROLLBACK_COMPLETE: Received response status [FAILED] from custom resource. Message returned: BadRequestException: The request failed because the GuardDuty service principal does not have permission to the KMS key or the resource specified by the destinationArn parameter. Refer to https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_exportfindings.html

See information about the Central Logging bucket CMK for more details and how to fix the issue.

Cannot exceed quota for RolesPerAccount error

You encounter an error similar to this one during LZA installation:

```
Deployment failed: Error: The stack named ASEA-SecurityResourcesStack-<account>-<region> failed creation, it may need to be manually deleted from the AWS
console: ROLLBACK_COMPLETE: Resource handler returned message: "Cannot exceed quota for RolesPerAccount: 1000 (Service: Iam, Status Code: 409, Request
ID: )" (RequestToken: , HandlerErrorCode: ServiceLimitExceeded)
```

You need to request a limit increase for the RolesPerAccount Quota. See the FAQ Which Service Quotas should be monitored for the upgrade?

Service Limit Exceeded Exception on Cloud Watch Logs resource policy

You encounter an error in the Network_Prepare or Security_Resources when trying to add a CloudWatch Logs resource policy.

The error code is `LimitExceededException` in the CloudTrail event for calls to `Logs:PutResourcePolicy`.

In the Security_Resources stack the root error might not be surfaced and this will show up as a timeout on a custom resource.

```
ASEA-SecurityResourcesStack-<account>-<region> | CREATE_FAILED        | AWS::CloudFormation::CustomResource   | SecurityHubEventsLog/
SecurityHubEventsFunction/SecurityHubEventsFunctionResource/Default (SecurityHubEventsLogSecurityHubEventsFunctionSecurityHubEventsFunctionResourceF6D56745)
Received response status [FAILED] from custom resource. Message returned: Error: Task timed out after 180.00 seconds
```

Cause: There is a hard limit of 10 CloudWatch Logs resource policies per account. LZA needs to create two.

Workaround: Remove existing CloudWatch Logs resource policies in the problematic account and region to free up sufficient space for LZA. You can use the AWS CLI describe-resource-policies command to list existing resource policies.

3.5.3 ASEA to LZA Upgrade Rollback Strategy

Rollback Strategy Overview

The existing ASEA to LZA upgrade process relies on a combination of automated and manual mechanisms to accomplish the upgrade. This is due to AWS service limits as well as resource collisions of resources which exist in both the ASEA and LZA solutions.

If an issue occurs during the upgrade process, there needs to be a rollback plan in place. Since the upgrade process utilizes both automated and manual steps, we will roll back in an automated fashion where possible and require manual steps for others. The high-level rollback steps are below.

Warning

Carefully review the current documentation to understand when rolling back is applicable. The rollback steps are intended as a last resort mechanism and cannot be applied once the LZA pipeline as run. Make sure you complete all the validation steps proposed before starting the upgrade procedures in your production environment.

Rollback Steps

The rollback steps are designed to re-install ASEA. These steps are only needed if you uninstalled ASEA by running the `asea-prep` command. The steps are only possible if you haven't started the LZA installation by running the `lza-prep` command.

- Confirm that the `${Prefix}-CDK-Toolkit` stacks have been deleted in all regions and accounts where the accelerator is deployed
- In the management account, empty and delete the CDK assets bucket (`cdk-hnb659fds-assets-<account>-ca-central-1`). This bucket is part of the `${Prefix}-CDK-Toolkit` stack and has a retention policy to retain, therefore it needs to be deleted manually
- Review, backup and delete the ASEA DynamoDB Tables
- In the management account, backup the content of the following DynamoDB tables: `ASEA-cidr-pool`, `ASEA-cidr-subnet-assign`, `ASEA-cidr-vpc-assign`, `ASEA-Output-Utils`, `ASEA-Outputs`, `ASEA-Parameters`
- If using dynamic IP allocation, the `ASEA-cidr-*` tables contain important data that you need to keep
- The content of the `ASEA-Output-Utils`, `ASEA-Outputs`, `ASEA-Parameters` will be regenerated in the ASEA install
- After backing up their content, delete the DynamoDB Tables, they will be re-created in ASEA install
- Run ASEA Installer Stack
- Download the Installer Stack from: https://github.com/aws-samples/aws-secure-environment-accelerator/releases
- Navigate to the CloudFormation homepage
- Click on the “Create Stack” button
- Choose the option “Upload a template file” and click on the “Choose file” button.
- Navigate to the Installer Stack template on your local machine, select the file, and click “Next”
- On the “Specify Stack details” page, provide a stack name for your deployment.
- Fill in the CloudFormation Parameters.
- Complete CloudFormation deployment.
- Run the ASEA-InstallerPipeline
- After deploying the CloudFormation template, a new CodePipeline pipeline will be created. This Pipeline will be called `{$Prefix}-InstallerPipeline`. - The Code Pipeline will automatically trigger an execution and begin running when created
- This pipeline runs a CodeBuild job which does a number of things – most importantly, create the ASEA State Machine.
- **IMPORTANT** If using dynamic IP allocation, you need to repopulate the data in the `ASEA-cidr-*` DDB tables that you backed up in an earlier step before running the ASEA State Machine. The state machine will be automatically triggered at the end of the ASEA `{$Prefix}-InstallerPipeline`, stop the pipeline execution when it reaches the `Execute` stage or stop the ASEA State Machine execution as soon as it starts.
- Run the ASEA State Machine
- After the InstallerPipeline has successfully run, the ASEA State Machine will be kicked off which will ensure that ASEA features are rolled back to match the ASEA configuration.

3.5.4 ASEA Resource Handlers

In order to accomplish upgrading from ASEA to LZA, the solution relies on a concept called ASEA Resource Handlers. These resource handlers utilize the CFN Include module to allow the LZA engine to manage ASEA resources in their original CloudFormation stacks. By using the CFN Include Module, the LZA application can modify certain properties of CloudFormation constructs. The current state of supported resources can be found in the table below:

Resource Type	Resource Deletion Supported	Resource Update Supported	Modifiable Attributes
Application Load Balancers	FALSE	FALSE	
EC2 Firewall Instance (Fortinet)	FALSE	FALSE	
ELB Target Group	FALSE	FALSE	
IAM Groups	TRUE	TRUE	Group Name Managed Policy Arns
IAM Managed Policies	TRUE	TRUE	Managed Policy Name Managed Policy Document
IAM Roles	TRUE	TRUE	Permissions Boundary Managed Policy Arns Assume Role Policy Document Instance Profile
IAM Users	TRUE	TRUE	Groups Permissions Boundary
Internet Gateway (IGW)	FALSE	FALSE	
ManagedAD	FALSE	FALSE	
NACL Subnet Associations	FALSE	TRUE	NACL Id Subnet Id
NAT Gateway	FALSE	TRUE	Subnet Id
Network Firewall	TRUE	TRUE	Firewall Logging Configuration
Network Firewall Policy	TRUE	FALSE	
Network Firewall Rule Group	TRUE	FALSE	
Route53 Hosted Zone	FALSE	FALSE	
Route53 Query Logging Association	FALSE	FALSE	
Route53 Record Set	FALSE	FALSE	
Route53 Resolver Endpoint	FALSE	FALSE	
Security Groups	FALSE	TRUE	Security Group Ingress Rules Security Group Egress Rules
Shared Security Group	FALSE	FALSE	
SSM Association	FALSE	FALSE	
SSM Resource Data Sync	FALSE	FALSE	
Subnets	FALSE	TRUE	Subnet CIDR Block Subnet Availability Zone Subnet Map Public IP on Launch
Transit Gateway Associations	FALSE	FALSE	
Transit Gateway Black Hole Routes	FALSE	FALSE	

Resource Type	Resource Deletion Supported	Resource Update Supported	Modifiable Attributes
Transit Gateway Propagations	FALSE	FALSE	
Transit Gateway Route Tables	FALSE	FALSE	
Transit Gateway Routes	FALSE	FALSE	
Transit Gateways	FALSE	TRUE	Amazon Side ASN Auto Accept Shared Attachments Default Route Table Associations Default Route Table Propagations DNS Support VPN ECMP Support
Virtual Private Gateway	FALSE	TRUE	Amazon Side ASN
VPC	FALSE	TRUE	CIDR Blocks Enable DNS Host Names Enable DNS Support Instance Tenancy
VPC Endpoint	TRUE	FALSE	None, Including associated security group. Must re-create endpoint
VPC Endpoint (Gateway)	FALSE	TRUE	Route Table Ids
VPC Peering Connection	FALSE	FALSE	

3.5.5 Known Issues

This is a list of known issues at the time of release. This list will be updated when new versions of the upgrade tools are released. Contact your AWS account teams for more details if these issues impact your upgrade.

Unsupported configurations

The following configurations are not handled automatically by the current version of the upgrade tools. Also review the Feature-specific considerations section of the documentation for additional details.

SITE-TO-SITE VPNS

Description: Site-to-site VPNs attached to Transit Gateway configured with ASEA are not converted to LZA configuration.

Symptom or error message: The customer gateway and VPN connections configurations are not generated in the LZA `network-config.yaml` file.

Resolution or workaround: While the configurations are not present in the LZA configuration files, the already deployed resources won't be affected during the upgrade. Thus the VPN connection will still be in place and no network disruption on the VPN tunnel is anticipated during the upgrade. After the upgrade you can plan deploying new VPN configurations natively using LZA and delete the original resources created by ASEA.

Upgrade known issues

The following issues can result in errors during the ASEA to LZA upgrade and should be fixed in the LZA configuration files before starting the LZA installation.

CONVERT-CONFIG SHOULD NOT CONVERT AUTOMATIONASSUMEROLE PARAMETERS FOR AWS CUSTOM CONFIG RULES REMEDIATION

Description: LZA handles the IAM Roles and Policies for custom config rule detection Lambda and remediation SSM document differently than ASEA. In the current state, convert-config generates a configuration that will generate an error at deployment time if the `remediation-params` in ASEA configuration contains a parameter named `AutomationAssumeRole`.

Symptom or error message: The SecurityResourcesStack stack fails on the creation of custom AWS Config Rule with the error `InvalidParameterValueException`.

Resolution or workaround: The `AutomationAssumeRole` parameter is automatically provided by LZA with the created role. You can comment the `AutomationAssumeRole` parameters in the remediation section of your custom config rules in security-config.yaml. See the Custom AWS Config Rules section in the Feature Specific Considerations for more details about AWS Custom Config Rules.

STACK EXCEEDS THE ALLOWED MAXIMUM OF 500 RESOURCES

Description: During LZA installation you receive an error message about exceeding the maximum number of resources allowed in a CloudFormation stack. This more commonly happen in the NetworkVPC stack.

```
Number of resources in stack 'ASEA-NetworkVpcStack-111222333444-ca-central-1': 571 is greater than allowed maximum of 500
```

Root cause: This issue is documented at the LZA level, and a fix is currently being developed and will be available in a future version of LZA. The new version will distribute resources differently between stacks to avoid the limit. Reference: NetworkVpc Stack exceeds the allowed maximum of 500 resources

In the context of an ASEA to LZA upgrade, existing ASEA resources remain in the ASEA stacks, and certain resources are recreated in LZA stacks, such as NACLs, route tables, and the association of route tables to subnets. Therefore, only these types of resources can be problematic, particularly the NetworkVpcStack of the shared networking account.

Resolution or workaround: The only possible workaround is to adjust the configuration to reduce the number of resources to be created in the stack (e.g. optimize the number of NACL rules) or wait for the fix to be available in a future LZA version. **If you added several VPC, Subnets and NACL rules from the default ASEA configuration you are likely to face this issue and should make additional verifications before attempting the LZA upgrade**.

During the preparation steps when you generate the LZA configuration files you can confirm the number of Subnets and NACLs that will be created in the NetworkVPC stack of the Shared Network account.

The following commands can be used to estimate from the `network-config.yaml` file the number of resources that have the most impact on the total number of resources in the NetworkVpcStack. (requires installation of yq)

```
# Number of subnets in the networking account (corresponds to AWS::EC2::SubnetRouteTableAssociation)
cat network-config.yaml | yq '.vpcs[] | select(.account == "shared-network" and .region == "ca-central-1") | .subnets[].name' | wc -l

# Number of shared subnets in the networking account (corresponds to AWS::RAM::ResourceShare)
cat network-config.yaml | yq '.vpcs[] | select(.account == "shared-network" and .region == "ca-central-1") | .subnets[] | select(. | has("shareTargets")) | .name' | wc -l

# Number of NACLs in the networking account (corresponds to AWS::EC2::NetworkAclEntry)
cat network-config.yaml | yq '[.vpcs[] | select(.account == "shared-network" and .region == "ca-central-1") | [.networkAcls[].inboundRules, .networkAcls[].outboundRules]] | flatten | length'
```

If the sum of those three type of resources is above 380 in a single account and region, further investigation is recommended before attempting the upgrade.

Landing Zone Accelerator known issues

The following issues will not prevent a successful upgrade from ASEA to LZA, but can impact functionalities and operations in the upgraded Landing Zone.

RESOURCES ARE NOT DELETED AFTER BEING REMOVED FROM CONFIGURATION FILE

Description: You attempt to remove a resource that was deployed by ASEA from the LZA configuration file and it is not removed after a successful LZA pipeline run.

Symptom or error message: The LZA pipeline runs with success, but the resource is not deleted.

Resolution or workaround: Not all ASEA resources support deletion through the LZA configuration and pipeline. Review the ASEA Resource Handlers page for the current state of supported handlers.

3.5.6 Fixed Issues

Fixed in LZA v1.11.1

The following issued were fixed as part of LZA v1.11.1 release.

ERROR ADDING A NEW ROUTE TARGETING FIREWALL INSTANCE

Description: After a successful upgrade, you try to add in `network-config.yaml` a route entry that targets ENI 0 of a firewall appliance using the lookup variable `${ACCEL_LOOKUP::EC2:ENI_0:Firewall_azA:Id}`

Symptom or error message: Error in the NetworkAssociationsStack after adding a route targeting ENI 0 of a firewall appliance.

```
Resource handler returned message: "Invalid id: "${ACCEL_LOOKUP::EC2:ENI_0:Firewall_azA:Id}" (expecting "eni-...")
```

Resolution or workaround: Fixed in LZA v1.11.1

SOME AWS CONFIG RULES DO NOT EVALUATE AFTER THE UPGRADE

Description: Some AWS Config Rules deployed by LZA do not evaluate (i.e Last successful detective evaluation appears as 'Not Available' in the console). The equivalent ASEA Config Rule evaluates correctly.

Symptom or error message: The scope of changes for Config Rule is set to an empty list of Resource types instead of scoped to **All changes** as in ASEA.

Resolution or workaround: Fixed in LZA v1.11.1

Fixed in LZA v1.12.0

ERROR: TRANSIT GATEWAY STATIC ROUTE ALREADY EXIST

Description: During LZA install, LZA attempts to re-create a transit gateway static route table entry that already exists.

Symptom or error message: Error in the NetworkAssociationsStack during LZA initial installation.

```
ASEA-NetworkAssociationsStack-xxxxxx-ca-central-1 failed: Error: The stack named ASEA-NetworkAssociationsStack-xxxxxx-ca-central-1 failed creation, it may
need to be manually deleted from the AWS console: ROLLBACK_COMPLETE: tgw-rtb-xxxxx|x.x.x.x/yy already exists in stack arn:aws:cloudformation:ca-
central-1:xxxxxxx:stack/ASEA-SharedNetwork-Phase2
```

Root cause: The static route was deployed in an ASEA stack and doesn't need to be deployed by LZA.

Resolution or workaround: Fixed in LZA v1.12.0. If you had commented the static route in the `network-config.yaml` file as the previous documented workaround, you can uncomment the route to have LZA start to manage the resource.

REMOVAL OF INTERFACE ENDPOINTS FAILS IN IMPORTASEARESOURCES STAGE

Description: Failure when attempting to remove an interface endpoint that was deployed by ASEA prior to LZA upgrade.

Symptom or error message: Failure in ImportAseaResources

```
ASEA-SharedNetwork-Phase2-VpcEndpoints1 failed: Error [ValidationError]: Template format error: Unresolved resource dependencies
[SsmParamEndpointVpccodecommitDns] in the Resources block of the template
```

Resolution or workaround: Fixed in LZA v1.12.0

4. 1. Accelerator Basic Operation and Frequently asked Questions

4.1 1.1. Operational Activities

1.1.1. How do I add new AWS accounts to my AWS Organization?

How do I add new AWS accounts to my AWS Organization?

We offer three options and all can be used in the same Accelerator deployment. All options work with AWS Control Tower, ensuring the account is both ingested into Control Tower and all Accelerator guardrails are automatically applied.

No matter the mechanism you choose, new accounts will automatically be blocked from use until fully guardrailed, the Accelerator will automatically execute, and accounts will automatically be ingested into AWS Control Tower (if deployed).

Option 1

Users can simply add the following five lines to the configuration file `workload-account-configs` section and rerun the state machine. The majority of the account configuration will be picked up from the OU the AWS account has been assigned. You can also add additional account specific configuration, or override items like the default OU budget with an account specific budget. This mechanism is often used by customers that wish to programmatically create AWS accounts using the Accelerator and allows for adding many new accounts at one time.

```
"fun-acct": {
  "account-name": "TheFunAccount",
  "email": "myemail+aseaT-funacct@example.com",
  "src-filename": "config.json",
  "ou": "Sandbox"
}
```

Option 2

We've heard consistent feedback that our customers wish to use native AWS services and do not want to do things differently once security controls, guardrails, or accelerators are applied to their environment. In this regard, simply create your new AWS account in AWS Organizations as you did before**, either by a) using the AWS Console or b) by using standard AWS account creation API's, CLI or 3rd party tools like Terraform.

- ** **IMPORTANT:** When creating the new AWS account using AWS Organizations, you need to specify the role name provided in the Accelerator configuration file `global-options\organization-admin-role`, otherwise we cannot bootstrap the account. In Control Tower installations, this **MUST** be set to ***`AWSControlTowerExecution`***, for customers who installed prior to v1.2.5 this value is ***`AWSCloudFormationStackSetExecutionRole`*** and after v1.2.5 we were recommending using the role ***`OrganizationAccountAccessRole`*** as this role is used by default by AWS Organizations if no role name is specified when creating AWS accounts through the AWS console or cli.
- On account creation we will apply a quarantine SCP which prevents the account from being used by anyone until the Accelerator has applied the appropriate guardrails
- Moving the account into the appropriate OU triggers the state machine and the application of the guardrails to the account, once complete, we will remove the quarantine SCP.
- **NOTE:** Accounts CANNOT be moved between OU's to maintain compliance, so select the proper top-level OU with care
- In AWS Organizations, select ALL the newly created AWS accounts and move them all (preferably at once) to the correct destination OU (assuming the same OU for all accounts)
- In case you need to move accounts to multiple OU's we have added a 2 minute delay before triggering the State Machine
- Any accounts moved after the 2 minute window will NOT be properly ingested, and will need to be ingested on a subsequent State Machine Execution.

Option 3

Create your account using Account Factory in the AWS Control Tower console.

1.1.2. I tried to enroll a new account via Control Tower but it failed?

I tried to enroll a new account via Control Tower but it failed?

or "The state machine failed during the 'Load Organization Configuration' step with the error 'The Control Tower account: ACCOUNT_NAME is in a failed state ERROR'"

If account enrollment fails within Control Tower, you will need to follow the troubleshooting steps here. A common reason for this is not having the `ControlTowerExectution` role created in the account you are trying to enroll. Even after you successfully enroll the account, it is possible the state machine will fail at `Load Organization Configuration`. If you look at the CloudWatch logs you will see the error message:

```
There were errors while loading the configuration: The Control Tower account: ACCOUNT_NAME is in a failed state ERROR.
```

This is because the Accelerator checks that there are no errors with Control Tower before continuing. In some cases Control Tower can leave an orphaned Service Catalog product in an **Error** state. You need to cleanup Control Towers Service Catalogs Provisioned Products so there are no products remaining in an error or tainted state before you can successfully re-run the state machine.

1.1.3. Can I use AWS Organizations for all tasks I currently use AWS Organizations for?

Can I use AWS Organizations for all tasks I currently use AWS Organizations for?

In AWS Organizations you can continue to:

- create and rename AWS accounts
- move AWS accounts between OU's
- create, delete and rename OU's, including support for nested OU's
- create, rename, modify, apply and remove SCP's

What can't I do:

- modify Accelerator or Control Tower controlled SCP's
- add/remove SCP's on top-level OU's (these are Accelerator and/or Control Tower controlled)
- users can change SCP's on non-top-level OU's and non-Accelerator controlled accounts as they please
- add/remove SCP's on specific accounts that have Accelerator controlled SCPs
- move an AWS account between top-level OU's (i.e. `Sandbox` to `Prod` is a security violation)
- moving between `Prod/sub-ou-1` to `Prod/sub-ou2` or `Prod/sub-ou2/sub-ou2a/sub-ou2ab` is fully supported
- create a top-level OU (need to validate, as they require config file entries)
- remove quarantine SCP from newly created accounts
- we do not support forward slashes (`/`) in OU names, even though the AWS platform does

More details:

- If an AWS account is renamed, an account email is changed, or an OU is renamed, on the next state machine execution, the config file will automatically be updated.
- If you edit an Accelerator controlled SCP through Organizations, we will reset it per what is defined in the Accelerator configuration files.
- If you add/remove an SCP from a top-level OU or Accelerator controlled account, we will put them back as defined in the Accelerator configuration file.
- If you move an account between top-level OU's, we will put it back to its original designated top-level OU.
- The Accelerator fully supports nested OU's, customers can create any depth OU structure in AWS Organizations and add/remove/change SCP's *below* the top-level as they desire or move accounts between these OU's without restriction. Users can create OU's to the full AWS OU structure/depth
- Except for the Quarantine SCP applied to specific accounts, we do not 'control' SCP's below the top level, customers can add/create/customize SCP's
- as of v1.3.3 customers can optionally control account level SCP's through the configuration file

1.1.4. How do I make changes to items I defined in the Accelerator configuration file during installation?

How do I make changes to items I defined in the Accelerator configuration file during installation?

Simply update your configuration file in CodeCommit and rerun the state machine! In most cases, it is that simple.

If you ask the Accelerator to do something that is not supported by the AWS platform, the state machine will fail, so it needs to be a supported capability. For example, the platform does not allow you to change the CIDR block on a VPC, but you can accomplish this as you would today by using the Accelerator to deploy a new second VPC, manually migrating workloads, and then removing the deprecated VPC from the Accelerator configuration.

Below we have also documented additional considerations when creating or updating the configuration file.

It should be noted that we have added code to the Accelerator to block customers from making many 'breaking' or impactful changes to their configuration files. If someone is positive they want to make these changes, we also provide override switches to allow these changes to be attempted forcefully.

1.1.5. Can I update the config file while the State Machine is running? When will those changes be applied?

Can I update the config file while the State Machine is running? When will those changes be applied?

Yes. The state machine captures a consistent input state of the requested configuration when it starts. The running Accelerator instance does not see or consider any configuration changes that occur after it has started. All configuration changes occurring after the state machine is running will only be leveraged on the *next* state machine execution.

1.1.6. What if I really mess up the configuration file?

What if I really mess up the configuration file?

The Accelerator is designed with checks to compare your current configuration file with the version of the config file from the previous successful execution of the state machine. If we believe you are making major or breaking changes to the config file, we will purposefully fail the state machine. See Config file and Deployment Protections for more details.

With the release of v1.3.0 we introduced state machine scoping capabilities to further protect customers, detailed here.

1.1.7. What if my State Machine fails? Why? Previous solutions had complex recovery processes, what's involved?

What if my State Machine fails? Why? Previous solutions had complex recovery processes, what's involved?

If your main state machine fails, review the error(s), resolve the problem and simply re-run the state machine. We've put a huge focus on ensuring the solution is idempotent and to ensure recovery is a smooth and easy process.

Ensuring the integrity of deployed guardrails is critical in operating and maintaining an environment hosting protected data. Based on customer feedback and security best practices, we purposely fail the state machine if we cannot successfully deploy guardrails.

Additionally, with millions of active customers each supporting different and diverse use cases and with the rapid rate of evolution of the AWS platform, sometimes we will encounter unexpected circumstances and the state machine might fail.

We've spent a lot of time over the course of the Accelerator development process ensuring the solution can roll forward, roll backward, be stopped, restarted, and rerun without issues. A huge focus was placed on dealing with and writing custom code to manage and deal with non-idempotent resources (like S3 buckets, log groups, KMS keys, etc.). We've spent a lot of time ensuring that any failed artifacts are automatically cleaned up and don't cause subsequent executions to fail. We've put a strong focus on ensuring you do not need to go into your various AWS sub-accounts and manually remove or cleanup resources or deployment failures. We've also tried to provide usable error messages that are easy to understand and troubleshoot. As new scenarios are brought to our attention, we continue to adjust the codebase to better handle these situations.

Will your state machine fail at some point in time, likely. Will you be able to easily recover and move forward without extensive time and effort, YES!

1.1.8. How do I update some of the supplied sample configuration items found in reference-artifact, like SCPs and IAM policies?

How do I update some of the supplied sample configuration items found in reference-artifact, like SCPs and IAM policies?

To override items like SCP's or IAM policies, customers simply need to provide the identically named file in their input bucket. As long as the file exists in the correct folder in the customers input bucket, the Accelerator will use the customers supplied version of the configuration item, rather than the Accelerator version. Customer SCP's need to be placed into a folder named `scp` and IAM policies in a folder named `iam-policy` (case sensitive).

The Accelerator was designed to allow customers complete customization capabilities without any requirement to update code or fork the GitHub repo. Additionally, rather than forcing customers to provide a multitude of config files for a standard or prescriptive installation, we provide and auto-deploy with Accelerator versions of most required configuration items from the reference-artifacts folder of the repo. If a customer provides the required configuration file in their Accelerator S3 input bucket, we will use the customer supplied version of the configuration file rather than the Accelerator version. At any time, either before initial installation, or in future, a customer can place new or updated SCPs, policies, or other supported file types into their input bucket and we will use those instead of or in addition to Accelerator supplied versions. Customer only need to provide the specific files they wish to override, not all files.

Customers can also define additional SCPs (or modify existing SCPs) using the name, description and filename of their choosing, and deploy them by referencing them on the appropriate organizational unit in the config file.

Prior to v1.2.5, if we updated the default files, we overwrote customers customizations during upgrade. Simply updating the timestamp *after* upgrade on the customized versions and then rerunning the state machine re-instates customer customizations. In v1.2.5 we always use the customer customized version from the S3 bucket. Its important customers assess newly provided defaults during an upgrade process to ensure they are incorporating all the latest fixes and improvements. If a customer wants to revert to Accelerator provided default files, they will need to manually copy it from the repo into their input bucket.

NOTE: Most of the provided SCPs are designed to protect the Accelerator deployed resources from modification and ensure the integrity of the Accelerator. Extreme caution must be exercised if the provided SCPs are modified. In v1.5.0 we restructured the SCPs based on a) customer requests, and b) the addition of Control Tower support for new installs.

- we reorganized and optimized our SCP's from 4 SCP files down to 3 SCP files, without removing any protections or guardrails;
- these optimizations have resulted in minor enhancements to the SCP protections and in some cases better scoping;
- the first two SCP files (Part-0 and Part-1) contain the controls which protect the integrity of the Accelerator itself;
- the third file (Sensitive, Unclass, Sandbox) contains customer data protection specific guardrails, which may change based on workload data classification or customer profiles and requirements;
- this freed the fourth SCP for use by Control Tower. As Control Tower leverages 2 SCP files on the Security OU, we have moved some of our SCP's to the account level.

1.1.9. I deployed AWS Managed Active Directory (MAD) as part of my deployment, how do I manage Active Directory domain users, groups, and domain policies after deployment?

I deployed AWS Managed Active Directory (MAD) as part of my deployment, how do I manage Active Directory domain users, groups, and domain policies after deployment?

Customers have clearly indicated they do NOT want to use the Accelerator to manage their Active Directory domain or change the way they manage Active Directory on an ongoing basis. Customer have also indicated, they need help getting up and running quickly. For these reasons, the Accelerator only sets the domain password policy, and creates AD users and groups on the initial installation of MAD. After the initial installation, customers must manage Windows users and groups using their traditional tools. A bastion Windows host is deployed as a mechanism to support these capabilities. Passwords for all newly created MAD users have been stored, encrypted, in AWS Secrets Manager in the Management (root) Organization AWS account.

To create new users and groups:

- RDP into the ASEA-RDGW bastion host in the Ops account
- Run ADUC and create users and groups as you please under the NETBIOSDOMAIN (example) tree
- Or run the appropriate PowerShell command
- Go to AWS SSO and map the Active Directory group to the appropriate AWS account and permission set

The Accelerator will not create/update/delete new AD users or groups, nor will it update the domain password policy after the initial installation of Managed Active Directory. It is your responsibility to rotate these passwords on a regular basis per your organizations password policy. (NOTE: After updating the admin password it needs to be stored back in secrets manager).

1.1.10. How do I suspend an AWS account?

How do I suspend an AWS account?

Suspending accounts is blocked via SCP and purposely difficult, two options exist:

1. Modify SCP method (not desired)
2. Leverage the UnManaged OU

• validate your config file contains the value: `"ignored-ous": ["UnManaged"]`

• the state machine must be executed at least once after this value is added to the config file

• In AWS Organizations create an OU named `UnManaged` in the root of the OU tree, if it does not exist

• Change to the `us-east-1` region and open CloudWatch and navigate to Rules

• Select the `PBMMAccel-MoveAccount_rule`, select actions, select `Disable`

• In Organizations move the account to be suspended to the `UnManaged` OU

• Change to the `us-east-1` region and open CloudWatch and navigate to Rules

• Select the `PBMMAccel-MoveAccount_rule`, select actions, select `Enable`

• login to the account to be suspended as the account root user

• suspend the account through `My Account`

• Run the state machine (from the Organization management account), the account will:

• have a deleted=true value added to the config file

• be moved to the suspended OU (OU value and path stays the same in the config file)

• deleted=true causes OU validation to be skipped on this account on subsequent SM executions

• If the AWS account was listed in the mandatory-accounts section of the config file the SM will fail (expected)

• after the above tasks have been completed, remove all references to the suspended mandatory account from the config file

• rerun the state machine, specifying: `{ "overrideComparison": true }`

• Deleted accounts will continue to appear under the `Suspended` OU for 90-days

1.1.11. I need a new VPC, where shall I define it?

I need a new VPC, where shall I define it?

You can define a VPC in one of four major sections of the Accelerator configuration file:

- within an organization unit (this is the recommended and preferred method);
- within an account in mandatory-account-configs;
- within an account in workload-account-configs;
- defined within an organization unit, but opted-in within the account config.

We generally recommend most items be defined within organizational units, such that all workload accounts pickup their persona from the OU they are associated and minimize per account configuration. Both a local account based VPC (as deployed in the Sandbox OU accounts), or a central shared VPC (as deployed in the Dev/Test/Prod OU accounts in many of the example configs) can be defined at the OU level.

As mandatory accounts often have unique configuration requirements, for example the centralized Endpoint VPC, they must be configured within the account's configuration. Customers can define VPC's or other account specific settings within any account's configuration, but this requires editing the configuration file for each account configuration.

Prior to v1.5.0, local VPC's defined at the OU level were each deployed with the same CIDR ranges and therefor could not be connected to a TGW. Local VPC's requiring centralized networking (i.e. TGW connectivity) were required to be defined in each account config, adding manual effort and bloating the configuration file.

The addition of `dynamic` and `lookup` CIDR sources in v1.5.0 resolves this problem. Local VPCs can be defined in an OU, and each VPC will be dynamically assigned a unique CIDR range from the assigned CIDR pool, or looked up from the DynamoDB database. Customers can now ensure connected, templated VPCs are consistently deployed to every account in an OU, each with unique IP addresses.

v1.5.0 also added a new opt-in VPC capability. A VPC is defined in an OU and a new config file variable is added to this VPC `opt-in: true`. When opt-in is set to true, the state machine does NOT create the VPC for the accounts in the OU, essentially ignoring the VPC definition. Select accounts in the OU can then be opted-in to the VPC(s) definition, by adding the value `accountname\opt-in-vpcs: [“opt-in-vpc-name1”, “opt-in-vpc-name2”, “opt-in-vpc-nameN”]` to the specific accounts which need the VPC(s). A VPC definition with the specified name (i.e. `opt-in-vpc-name1`) and the value `opt-in: true`, must exist in the OU config for the specified account. When these conditions apply, the VPC will be created in the account per the OU definition. Additional opt-in VPCs can be added to an account, but VPC's cannot be removed from the opt-in-vpcs array. VPC's can be TGW attached, assuming `dynamic` cidr-src is utilized, or DynamoDB is prepopulated with the required CIDR ranges using `lookup` mode. `cidr-src` provided is suitable for disconnected Sandbox type accounts.

The Future: While Opt-In VPCs are powerful, we want to take this further. Why not deploy an AWS Service Catalog template which contains the names of all the available opt-in VPCs for the accounts OU, inside each account. An account end user could then request a new VPC for their account from the list of available opt-in patterns. A user's selection would be sent to a centralized queue for approval (w/auto-approval options), which would result in the opt-in-vpc entry in that account being updated with the end users requested VPC pattern and the personalized VPC being created in the account and attached to the centralized TGW (if part of the pattern). This would ensure all VPC's conformed to a set of desirable design patterns, but also allow the end-user community choices based on their desired development and app patterns. If you like this idea, please +1 this feature request.

1.1.12. How do I modify and extend the Accelerator or execute my own code after the Accelerator provisions a new AWS account or the state machine executes?

How do I modify and extend the Accelerator or execute my own code after the Accelerator provisions a new AWS account or the state machine executes?

Flexibility:

- The AWS Secure Environment Accelerator was developed to enable extreme flexibility without requiring a single line of code to be changed. One of our primary goals throughout the development process was to avoid making any decisions that would result in users needing to fork or branch the Accelerator codebase. This would help ensure we had a sustainable and upgradable solution for a broad customer base over time.
- Functionality provided by the Accelerator can generally be controlled by modifying the main Accelerator configuration file.
- Items like SCP's, rsyslog config, PowerShell scripts, and iam-policies have config files provided and auto-deployed as part of the Accelerator to deliver on the prescriptive architecture (these are located in the \reference-artifacts folder of the GitHub repo for reference). If you want to alter the functionality delivered by any of these additional config files, you can simply provide your own by placing it in your specified Accelerator bucket in the appropriate sub-folder. The Accelerator will use your provided version instead of the supplied repo reference version.
- As SCP's and IAM policies are defined in the main config file, you can simply define new policies, pointing to new policy files, and provide these new files in your bucket, and they will be used.
- While a sample firewall config file is provided in the \reference-artifacts folder, it must be manually placed in your S3 bucket/folder on new Accelerator deployments
- Any/all of these files can be updated at any time and will be used on the next execution of the state machine
- Over time, we predict we will provide several sample or reference architectures and not just the current single PBMM architecture (all located in the \reference-artifacts\SAMPLE_CONFIGS folder).

Extensibility:

- Every execution of the state machine sends a state machine status event to a state machine SNS topic
- These status events include the Success/Failure status of the state machine, and on success, a list of all successfully processed AWS accounts
- While this SNS topic is automatically subscribed to a user provided email address for user notification, users can also create additional SNS subscriptions to enable triggering their own subsequent workflows, state machines, or custom code using any supported SNS subscription type (Lambda, SQS, Email, HTTPS, HTTPS)
- Additionally, objects deployed within an account have been populated in Parameter Store, see answer 1.3.2 for details

Example:

- One of our early adopter customers has developed a custom user interface which allows their clients to request new AWS environments. Clients provide items like cost center, budget, and select their environment requirements (i.e. Sandbox, Unclass or full sensitive SDLC account set). On appropriate approval, this pushes the changes to the Accelerator configuration file and triggers the state machine.
- Once the state machine completes, the SNS topic triggers their follow-up workflow, validates the requested accounts were provisioned, updates the customer's account database, and then executes a collection of customer specific follow-up workflow actions on any newly provisioned accounts.

1.1.13. How can I easily access my virtual machines or EC2 instances?

How can I easily access my virtual machines or EC2 instances?

The preferred and recommended method to connect to instances within the Accelerator is by using AWS Systems Manager Session Manager. Session Manager allows access to instances without the need to have any open firewall ports. Session Manager allows for Command line access to instances (both Windows and Linux) directly through the AWS console, with all activity logged to CloudWatch Logs. Session Manager enables customers to connect to Windows instances with a native RDP client and Linux instances with a native SSH client, if desired. Customers can gain quick access to instances through the AWS console, or using their preferred clients.

General

- Both the RDGW and rsyslog instances deployed in the Ops account are properly configured for Systems Manager Session Manager
- We have implemented automation such that all instances are also automatically configured for Session Manager (i.e. configured with the appropriate launch role, has a recent session manager agent installed (most amazon AMI's do), has access to an SSM endpoint)

NEW - System Manager Fleet Manager is now available, which allows connecting graphically to Windows desktops directly from the AWS console without the need for any command line access or tools, and without any requirement for an RDSH/ RDP client. As the Accelerator configures Systems Manager, this feature "simply works" for most instances deployed within the Accelerator.

```
- Navigate to Systems Manager, open Fleet Manager, select an instance, click "Node Actions" and select "Connect with Remote
Desktop". If you are logging in via AWS SSO, you can seamlessly connect. IAM users must supply valid credentials.
```

Connecting to an Instance

- From the AWS Console
- Go to: EC2, Instances, select the instance (i.e. ASEA-RDGW), click “Connect”, select Session Manager, Connect
- Ideal for Linux or Windows PowerShell users
- Everything is fully logged for future reference
- Directly through local RDP client using Session Managers tunnel capability:
- Provides direct access to your instances/host without any open internet ports on the RDGW instance, using a local/fat client tool
- Install AWS CLI v2 on your PC - available here (uninstall CLIv1 first, if installed)
- Install the SSM plugin on your PC - available here
- Get AWS temporary credentials from AWS SSO for the account your workload resides (i.e. Ops account when accessing the ASEA-RDGW instance) by selecting “Command line or programmatic access” instead of “Management Console” and paste them into a command prompt
- i.e. via logging in here: https://xxxxxxx.awsapps.com/start or
- This blog describes the process to use SSO to get credentials for the AWS CLI directly without the GUI
- Then enter: aws ssm start-session --target "i-12345678901234567" --document-name AWS-StartPortForwardingSession --parameters portNumber="3389",localPortNumber="56789"--region ca-central-1
- Command syntax is slightly different on Linux/Mac
- Replace i-1111adddce582b23c with the instance id of your RDGW instance
- A tunnel will open
- As these are tunnels to proprietary protocols (i.e. RDP/screen scraping) session content is not logged.
- Run mstsc/rdp client and connect to 127.0.0.1:56789
- By replacing 3389 with a new port for another applications (i.e. SSH running on a Linux instance), you can connect to a different application type
- You can change the local port by changing 56789 to any other valid port number (i.e. connecting to multiple instances at the same time)
- Login with the windows credentials discussed above in the format NETBIOSDOMAIN\User1 (i.e. example\user1)
- Your Netbios domain is found here in your config file: "netbios-domain": "example",
- Connect to your desktop command line to command line interface of remote Windows or Linux servers, instead of through console (i.e. no tunnel):
- aws ssm start-session --target "i-090c25e64c2d9d276""--region ca-central-1
- Replace i-xxx with your instance ID
- Everything is fully logged for future reference
- If you want to remove the region from your command line, you can:
- Type: “aws configure” from command prompt, hit {enter} (key), {enter} (secret), enter: ca-central-1, {enter}

1.1.14. I ran the state machine but it failed when it tried to delete the default VPC?

I ran the state machine but it failed when it tried to delete the default VPC? The state machine cannot delete the default VPC (Error: VPC has dependencies and cannot be deleted)

You need to ensure that resources don't exist in the default VPC or else the state machine won't be able to delete it. If you encounter this error, you can either delete the resources within the VPC or delete the default VPC manually and run the state machine again.

4.2 1.2. Existing Accounts / Organizations

1.2.1. How do I import an existing AWS account into my Accelerator managed AWS Organization (or what if I created a new AWS account with a different Organization trust role)?

How do I import an existing AWS account into my Accelerator managed AWS Organization (or what if I created a new AWS account with a different Organization trust role)?

- Ensure you have valid administrative privileges for the account to be invited/added
- Add the account to your AWS Organization using standard processes (i.e. Invite/Accept)
- this process does NOT create an organization trust role
- imported accounts do NOT have the quarantine SCP applied as we don't want to break existing workloads
- Login to the account using the existing administrative credentials
- Execute the Accelerator provided CloudFormation template to create the required Accelerator bootstrapping role - in the GitHub repo here: `reference-artifacts\Custom-Scripts\Import-Account-CFN-Role-Template.yml`
- add the account to the Accelerator config file and run the state machine
- If you simply created the account with an incorrect role name, you likely need to take extra steps:
- Update the Accelerator config file to add the parameter: `global-options\ignored-ous` = `["UnManagedAccounts"]`
- In AWS Organizations, create a new OU named `UnManagedAccounts` (case sensitive)
- Move the account to the `UnManagedAccounts` OU
- You can now remove the Quarantine SCP from the account
- Assume an administrative role into the account
- Execute the Accelerator provided CloudFormation template to create the required Accelerator bootstrapping role

1.2.2. Is it possible to deploy the Accelerator on top of an AWS Organization that I have already installed the AWS Landing Zone (ALZ) solution into?

Is it possible to deploy the Accelerator on top of an AWS Organization that I have already installed the AWS Landing Zone (ALZ) solution into?

Existing ALZ customers are required to uninstall their ALZ deployment before deploying the Accelerator. Please work with your AWS account team to find the best mechanism to uninstall the ALZ solution (procedures and scripts exist). It is often easier to migrate AWS accounts to a new Accelerator Organization, per the process detailed in the next FAQ question. Additionally, please reference the following section of the Installation Guide for additional considerations.

1.2.3. What if I want to move an account from an AWS Organization that has the ALZ deployed into an AWS Organization running the Accelerator?

What if I want to move an account from an AWS Organization that has the ALZ deployed into an AWS Organization running the Accelerator?

Before removing the AWS account from the source organization, terminate the AWS Service Catalog product associated with the member account that you're interested in moving. Ensuring the product terminates successfully and that there aren't any remaining CloudFormation stacks in the account that were deployed by the ALZ. You can then remove the account from the existing Organization and invite it into the new organization. Accounts invited into the Organization do NOT get the `Deny All` SCP applied, as we do not want to break existing running workloads. Moving the newly invited account into its destination OU will trigger the state machine and result in the account being ingested into the Accelerator and having the guardrails applied per the target OU persona.

For a detailed procedure, please review this document.

4.3 1.3. End User Environment

1.3.1. Is there anything my end users need to be aware of? Why do some of my end users struggle with CloudWatch Log groups errors?

Is there anything my end users need to be aware of? Why do some of my end users struggle with CloudWatch Log groups errors?

CloudWatch Log group deletion is prevented for security purposes and bypassing this rule would be a fundamental violation of security best practices. This protection does NOT exist solely to protect ASEA logs, but ALL log groups. Users of the Accelerator environment will need to ensure they set CloudFormation stack Log group retention type to RETAIN, or stack deletes will fail when attempting to delete a stack (as deleting the log group will be blocked) and users will encounter errors. As repeated stack deployments will be prevented from recreating the same log group name (as it already exists), end users will either need to check for the existence of the log group before attempting creation, or include a random hash in the log group name. The Accelerator also sets log group retention for all log groups to value(s) specified by customers in the config file and prevents end users from setting or changing Log group retentions. When creating new log groups, end users must either *not* configure a retention period, or set it to the default `NEVER expire` or they will also be blocked from creating the CloudWatch Log group. If applied by bypassing the guardrails, customer specified retention periods on log group creation will be overridden with the Accelerator specified retention period.

While a security best practice, some end users continue to request this be changed, but you need to ask: Are end users allowed to go in and clean out logs from Windows Event Viewer (locally or on domain controllers) after testing? Clean out Linux kernel logs? Apache log histories? The fundamental principal is that all and as many logs as possible will be retained for a defined retention period (some longer). In the "old days", logs were hidden deep within OS directory structures or access restricted by IT from developers - now that we make them all centralized, visible, and accessible, end users seem to think they suddenly need to clean them up. Customers need to establish a usable and scalable log group naming standard/convention as the first step in moving past this concern, such that they can always find their active logs easily. As stated, to enable repeated install and removal of stacks during test cycles, end user CloudFormation stacks need to set log groups to RETAIN and leverage a random hash in log group naming (or check for existence, before creating).

The Accelerator provided SCPs (guardrails/protections) are our recommendations, yet designed to be fully customizable, enabling any customer to carefully override these defaults to meet their individual requirements. If insistent, we'd suggest only bypassing the policy on the Sandbox OU, and only for log groups that start with a very specific prefix (not all log groups). When a customer wants to use the delete capability, they would need to name their log group with the designated prefix - i.e. opt-in to allow CloudWatch log group deletes.

1.3.2. How can I leverage Accelerator deployed objects in my IaC? Do I need to manually determine the arn's and object id's of Accelerator deployed objects to leverage them in my IaC?

How can I leverage Accelerator deployed objects in my IaC? Do I need to manually determine the arn's and object id's of Accelerator deployed objects to leverage them in my IaC?

Objects deployed by the Accelerator which customers may need to leverage in their own IaC have been populated in parameters in AWS parameter store for use by the IaC tooling of choice. The Accelerator ensures parameters are deployed consistently across accounts and OUs, such that a customer's code does not need to be updated when it is moved between accounts or promoted from Dev to Test to Prod.

Objects of the following types and their associated values are stored in parameter store: VPC, subnet, security group, ELB (ALB/NLB w/DNS address), IAM policy, IAM role, KMS key, ACM cert, SNS topic, and the firewall replacement variables.

Additionally, setting "populate-all-elbs-in-param-store": true for an account will populates all Accelerator wide ELB information into parameter store within that account. The sample PBMM configuration files set this value on the perimeter account, such that ELB information is available to configure centralized ingress capabilities.

1.3.3. How do I deploy AWS Elastic Beanstalk instances?

How do I deploy AWS Elastic Beanstalk instances?

If your deployed environment contains an SCP enforcing volume encryption of EC2 instances, your Elastic Beanstalk deployment will fail.

The SCP will contain an entry like this:

```
{
  "Sid": "EBS1",
  "Effect": "Deny",
  "Action": "ec2:RunInstances",
  "Resource": "arn:aws:ec2:*:*:volume/*",
  "Condition": {
    "Bool": {
      "ec2:Encrypted": "false"
    }
  }
},
```

A solution is to encrypt the root volume of the AMI that Elastic Beanstalk uses for your selected platform, and perform a custom AMI deployment of your Elastic Beanstalk application.

You can gather the AMI that Elastic Beanstalk uses via CLI with the following command:

```
aws elasticbeanstalk describe-platform-version --region <YOUR_REGION> --platform-arn <ARN_EB_PLATFORM>
```

Once you have gathered the AMI ID successfully, go to the EC2 console and:

- Click on the ‘AMIs’ option in the left navigation pane
- Search for your AMI after selecting ‘Public Images’ from the dropdown list.
- Select the AMI
- Go to Actions and Copy AMI
- Click on the checkbox to enable ‘Encryption’ and then select "Copy AMI".

Once the AMI is successfully copied, you can use this AMI to specify a custom AMI in your Elastic Beanstalk environments with root volume encrypted.

4.4 1.4. Upgrades

1.4.1. Can I upgrade directly to the latest release, or must I perform upgrades sequentially?

Can I upgrade directly to the latest release, or must I perform upgrades sequentially?

Yes, currently customers can upgrade from whatever version they have deployed to the latest Accelerator version. There is no requirement to perform sequential upgrades. In fact, we strongly discourage sequential upgrades.

Given the magnitude of the v1.5.0 release, we have added a one-time requirement that all customers upgrade to a minimum of v1.3.8 before attempting to upgrade to v1.5.0.

1.4.2. Why do I get the error "There were errors while comparing the configuration changes:" when I update the config file?

Why do I get the error "There were errors while comparing the configuration changes:" when I update the config file?

In v1.3.0 we added protections to allow customers to verify the scope of impact of their intended changes to the configuration file. In v1.3.0 and above, the state machine does not allow changes to the config file (other than new accounts) without providing the `scope` parameter. Please refer to the State Machine behavior and inputs Guide for more details.

4.5 1.5. Support Concerns

1.5.1. The Accelerator is written in CDK and deploys CloudFormation, does this restrict the Infrastructure as Code (IaC) tools that I can use?

The Accelerator is written in CDK and deploys CloudFormation, does this restrict the Infrastructure as Code (IaC) tools that I can use?

No. Customers can choose the IaC framework or tooling of their choice. The tooling used to deploy the Accelerator has no impact on the automation framework customers use to deploy their applications within the Accelerator environment. It should be noted that the functionality deployed by the Accelerator is extremely platform specific and would not benefit from multi-platform IaC frameworks or tooling.

1.5.2. What happens if AWS stops enhancing the Accelerator?

What happens if AWS stops enhancing the Accelerator?

The Accelerator is an open source project, should AWS stop enhancing the solution for any reason, the community has access to the full codebase, its roadmap and history. The community can enhance, update, fork and take ownership of the project, as appropriate.

The Accelerator is an AWS CDK based project and synthesizes to native AWS CloudFormation. AWS sub-accounts simply contain native CloudFormation stacks and associated custom resources, when required. The Accelerator architecture is such that all CloudFormation stacks are native to each AWS account with no links or ties to code in other AWS accounts or even other stacks within the same AWS account. This was an important initial design decision.

The Accelerator codebase can be completely uninstalled from the organization management (root) account, without any impact to the deployed functionality or guardrails. In this situation, guardrail updates and new account provisioning reverts to a manual process. Should a customer decide they no longer wish to utilize the solution, they can remove the Accelerator codebase without any impact to deployed resources and go back to doing things natively in AWS as they did before they deployed the Accelerator. By adopting the Accelerator, customers are not locking themselves in or making a one-way door decision.

1.5.3. What level of Support will the ASEA have from AWS Support?

What level of Support will the ASEA have from AWS Support?

The majority of the solution leverages native AWS services which are fully supported by AWS Support. Additionally, the Accelerator is an AWS CDK based project and synthesizes to native AWS CloudFormation. AWS sub-accounts simply contain native CloudFormation stacks and associated custom resources (when required). The Accelerator architecture is such that all CloudFormation stacks are native to each AWS account with no direct links or ties to code in other AWS accounts (no stacksets, no local CDK). This was an important project design decision, keeping deployed functionality in independent local CloudFormation stacks and decoupled from solution code, which allows AWS support to effectively troubleshoot and diagnose issues local to the sub-account.

As the Accelerator also includes code, anything specifically related to the Accelerator codebase will be only supported on a "best effort" basis by AWS support, as AWS support does not support custom code. The first line of support for the codebase is typically your local AWS team (your SA, TAM, ProServe and/or AWS Partner). As an open source project, customers can file requests using GitHub Issues against the Accelerator repository or open a discussion in GitHub discussions. Most customer issues arise during installation and are related to configuration customization or during the upgrade process.

1.5.4. What does it take to support the Accelerator?

What does it take to support the Accelerator?

We advise customers to allocate a 1/2 day per quarter to upgrade to the latest Accelerator release.

Customers have indicated that deploying the Accelerator reduces their ongoing operational burden over operating in native AWS, saving hours of effort every time a new account is provisioned by automating the deployment of the persona associated with new accounts (guardrails, networking and security). The Accelerator does NOT alleviate a customer's requirement to learn to effectively operate in the cloud (like monitoring security tooling/carrying out Security Operation Center (SOC) duties). This effort exists regardless of the existence of the Accelerator.

1.5.5. Is the Accelerator only designed and suitable for Government of Canada or PBMM customers?

Is the Accelerator only designed and suitable for Government of Canada or PBMM customers?

No. The Accelerator is targeted at ***any AWS customer*** that is looking to automate the deployment and management of a comprehensive end-to-end multi-account environment in AWS. It is ideally suited for customers interested in achieving a high security posture in AWS.

The Accelerator is a sophisticated deployment framework that allows for the deployment and management of virtually any AWS multi-account "Landing Zone" architecture without any code modifications. The Accelerator is actually delivering two separate and distinct products which can each be used on their own:

1. the Accelerator the tool, which can deploy virtually any architecture based on a provided config file (no code changes), and;
2. the Government of Canada (GC) prescriptive PBMM architecture which is delivered as a sample configuration file and documentation.

The tooling was purposely built to be extremely flexible, as we realized that some customers may not like some of the opinionated and prescriptive design decisions we made in the GC architecture. Virtually every feature being deployed can be turned on/off, not be used or can have its configuration adjusted to meet your specific design requirements.

We are working on building a library of sample config files to support additional customer needs and better demonstrate product capabilities and different architecture patterns. In no way is it required that the prescriptive GC architecture be used or deployed. Just because we can deploy, for example, an AWS Managed Active Directory, does not mean you need to use that feature of the solution. Disabling or changing these capabilities also requires zero code changes.

While the prescriptive sample configuration files were originally developed based on GC requirements, they were also developed following AWS Best Practices. Additionally, many security frameworks around the world have similar and overlapping security requirements (you can only do security so many ways). The provided architecture is applicable to many security compliance regimes around the world and not just the GC.

4.6 1.6. Deployed Functionality

1.6.1. I wish to be in compliance with the 12 GC TBS Guardrails, what don't you cover with the provided sample architecture?

I wish to be in compliance with the 12 GC TBS Guardrails, what don't you cover with the provided sample architecture?

The AWS SEA allows for a lot of flexibility in deployed architectures. If used, the provided PBMM sample architecture was designed to help deliver on the technical portion of *all* 12 of the GC guardrails, when automation was possible.

What don't we cover? Assigning MFA to users is a manual process. Specifically, you need to procure Yubikeys for your root/break glass users, and enable a suitable form of MFA for *all* other users (i.e. virtual, email, other). The guardrails also include some organizational processes (i.e. break glass procedures, or signing an MOU with CCCS) which customers will need to work through independently.

While AWS is providing the tools to help customer be compliant with the 12 PBMM guardrails (which were developed in collaboration with the GC) - it's up to each customers ITSec organization to assess and determine if the deployed controls actually meet their security requirements.

Finally, while we started with a goal of delivering on the 12 guardrails, we believe we have extended well beyond those security controls, to further help customers move towards meeting the full PBMM technical control profile (official documentation is weak in this area at this time).

1.6.2. Does the ALB perform SSL offloading?

Does the ALB perform SSL offloading?

As configured - the perimeter ALB decrypts incoming traffic using its certificate and then re-encrypts it with the certificate for the back-end ALB. The front-end and back-end ALB's can use the same or different certs. If the Firewall needs to inspect the traffic, it also needs the backend certificate be manually installed.

1.6.3. What is the recommended approach to manage the ALB certificates deployed by the Accelerator?

What is the recommended approach to manage the ALB certificates deployed by the Accelerator?

The Accelerator installation process allows customers to provide their own certificates (either self-signed or generated by a CA), to enable quick and easy installation and allowing customers to test end-to-end traffic flows. After the initial installation, we recommend customers leverage AWS Certificate Manager (ACM) to easily provision, manage, and deploy public and private SSL/TLS certificates. ACM helps manage the challenges of maintaining certificates, including certificate rotation and renewal, so you don’t have to worry about expiring certificates.

The Accelerator provides 3 mechanisms to enable utilizing certificates with ALB's:

- **Method 1** - IMPORT a certificate into AWS Certificate Manager from a 3rd party product
- When using a certificate that does not have a certificate chain (usually this is the case with Self-Signed)

```
"certificates": [
  {
   "name": "My-Cert",
   "type": "import",
   "priv-key": "certs/example1-cert.key",
   "cert": "certs/example1-cert.crt"
  }
 ]
```

- When using a certificate that has a certificate chain (usually this is the case when signed by a Certificate Authority with a CA Bundle)

```
"certificates": [
  {
   "name": "My-Cert",
   "type": "import",
   "priv-key": "certs/example1-cert.key",
   "cert": "certs/example1-cert.crt",
   "chain": "certs/example1-cert.chain"
  }
 ]
```

- this mechanism allows a customer to generate certificates using their existing tools and processes and import 3rd party certificates into AWS Certificate Manager for use in AWS
- Self-Signed certificates should NOT be used for production (samples were provided simply to demonstrate functionality)
- both a `.key` and a `.crt` file must be supplied in the customers S3 input bucket
- "cert" must contain only the certificate and not the full chain
- "chain" is an optional attribute that contains the certificate chain. This is generally used when importing a CA signed certificate
- this will create a certificate in ACM and a secret in secrets manager named `accelerator/certificates/My-Cert` in the specified AWS account(s), which points to the newly imported certificates ARN

- **Method 2** - REQUEST AWS Certificate Manager generate a certificate

```
"certificates": [
  {
    "name": "My-Cert",
    "type": "request",
    "domain": "*.example.com",
    "validation": "DNS",
    "san": ["www.example.com"]
  }
]
```

- this mechanism allows a customer to generate new public certificates directly in ACM
- both `DNS` and `EMAIL` validation mechanisms are supported (DNS recommended)
- this requires a ***Public*** DNS zone be properly configured to validate you are legally entitled to issue certificates for the domain
- this will also create a certificate in ACM and a secret in secrets manager named `accelerator/certificates/My-Cert` in the specified AWS account(s), which points to the newly imported certificates ARN
- this mechanism should NOT be used on new installs, skip certificate and ALB deployment during initial deployment (removing them from the config file) and simply add on a subsequent state machine execution
- Process:
- you need a public DNS domain properly registered and configured to publicly resolve the domain(s) you will be generating certificates for (i.e. example.com)
- domains can be purchased and configured in Amazon Route53 or through any 3rd party registrar and DNS service provider
- in Accelerator phase 1, the cert is generated, but the stack does NOT complete deploying (i.e. it waits) until certificate validation is complete
- during deployment, go to the AWS account in question, open ACM and the newly requested certificate. Document the authorization CNAME record required to validate certificate generation
- add the CNAME record to the zone in bullet 1 (in Route53 or 3rd party DNS provider) (documented here)
- after a few minutes the certificate will validate and switch to `Issued` status
- Accelerator phase 1 will finish (as long as the certificate is validated before the Phase 1 credentials time-out after 60-minutes)
- the ALB will deploy in a later phase with the specified certificate

- **Method 3** - Manually generate a certificate in ACM
- this mechanism allows a customer to manually generate certificates directly in the ACM interface for use by the Accelerator
- this mechanism should NOT be used on new installs, skip certificate and ALB deployment during initial deployment (removing them from the config file) and simply add on a subsequent state machine execution
- Process:
- go to the AWS account for which you plan to deploy an ALB and open ACM
- generate a certificate, documenting the certificates ARN
- open Secrets manager and generate a new secret of the format `accelerator/certificates/My-Cert` (of type `Plaintext` under `Other type of secrets`), where `My-Cert` is the unique name you will use to reference this certificate
- In all three mechanisms a secret will exist in Secrets Manager named `accelerator/certificates/My-Cert` which contains the ARN of the certificate to be used.

- In the Accelerator config file, find the definition of the ALB for that AWS account and specify `My-Cert` for the ALB `cert-name`

```
"alb": [
  {
    "cert-name": "My-Cert"
  }
]
```

- The state machine will fail if you specify a certificate in any ALB which is not defined in Secrets Manager in the local account.

We suggest the most effective mechanism for leveraging ACM is by adding CNAME authorization records to the relevant DNS domains using Method 2, but may not appropriate right for all customers.

1.6.4. Why do we have rsyslog servers? I thought everything was sent to CloudWatch?

Why do we have rsyslog servers? I thought everything was sent to CloudWatch?

The rsyslog servers are included to accept logs for appliances and third party applications that do not natively support the CloudWatch Agent from any account within a customers Organization. These logs are then immediately forwarded to CloudWatch Logs within the account the rsyslog servers are deployed (Operations) and are also copied to the S3 immutable bucket in the log-archive account. Logs are only persisted on the rsyslog hosts for 24 hours. The rsyslog servers are required to centralize the 3rd party firewall logs (Fortinet Fortigate).

1.6.5. Can you deploy the solution without Fortinet Firewall Licenses?

Can you deploy the solution without Fortinet Firewall Licenses?

Yes, if license files are not provided, the firewalls will come up configured and route traffic, but customers will have no mechanism to manage the firewalls/change the configuration until a valid license file is added. If invalid licence files are provided, the firewalls will fail to load the provided configuration, will not enable routing, will not bring up the VPN tunnels and will not be manageable. Customers will need to either remove and redeploy the firewalls, or manually configure them. If performing a test deployment, please work with your local Fortinet account team to discuss any options for temporary evaluation licenses.

Additionally, several additional firewall options are now available, including using AWS Network Firewall, a native AWS service.

1.6.6. I installed additional software on my Accelerator deployed RDGW / rsyslog host, where did it go?

I installed additional software on my Accelerator deployed RDGW / rsyslog host, where did it go?

The RDGW and rsyslog hosts are members of auto-scaling groups. These auto-scaling groups have been configured to refresh instances in the pool on a regular basis (7-days in the current sample config files). This ensures these instances are always clean. Additionally, on every execution of the Accelerator state machine the ASG are updated to the latest AWS AMI for the instances. When the auto-scaling group refreshes its instances, they will be redeployed with the latest patch release of the AMI/OS. It is recommended that the state machine be executed monthly to ensure the latest AMI's are always in use.

Customers wanting to install additional software on these instances should either a) update the automated deployment scripts to install the new software on new instance launch, or b) create and specify a custom AMI in the Accelerator configuration file which has the software pre-installed ensuring they are also managing patch compliance on the instance through some other mechanism.

At any time, customers can terminate the RDGW or rsyslog hosts and they will automatically be re-created from the base images with the latest patch available at the time of the last Accelerator State Machine execution.

1.6.7. Some sample configurations provide NACLs and Security Groups. Is that enough?

Some sample configurations provide NACLs and Security Groups. Is that enough?

Security group egress rules are often used in 'allow all' mode (`0.0.0.0/0`), with the focus primarily being on consistently allow listing required ingress traffic (centralized ingress/egress controls are in-place using the perimeter firewalls). This ensures day to day activities like patching, access to DNS, or to directory services access can function on instances without friction.

The Accelerator provided sample security groups in the workload accounts offer a good balance that considers both security, ease of operations, and frictionless development. They allow developers to focus on developing, enabling them to simply use the pre-created security constructs for their workloads, and avoid the creation of wide-open security groups. Developers can equally choose to create more appropriate least-privilege security groups more suitable for their application, if they are skilled in this area. It is expected as an application is promoted through the SDLC cycle from Dev through Test to Prod, these security groups will be further refined by the extended customers teams to further reduce privilege, as appropriate. It is expected that each customer will review and tailor their Security Groups based on their own security requirements. The provided security groups ensures day to day activities like patching, access to DNS, or to directory services access can function on instances without friction, with the understanding further protections are providing by the central ingress/egress firewalls.

The use of NACLs are general discouraged, but leveraged in this architecture as a defense-in-depth mechanism. Security groups should be used as the primary access control mechanism. As with security groups, we encourage customers to review and tailor their NACLs based on their own security requirements.

1.6.8. Can I deploy the solution as the account root user?

Can I deploy the solution as the account root user?

No, you cannot install as the root user. The root user has no ability to assume roles which is a requirement to configure the sub-accounts and will prevent the deployment. As per the installation instructions, you require an IAM user with the `AdministratorAccess` policy attached.

1.6.9. Is the Organizational Management root account monitored similarly to the other accounts in the organization?

Is the Organizational Management root account monitored similarly to the other accounts in the organization?

Yes, all accounts including the Organization Management or root account have the same monitoring and logging services enabled. When supported, AWS security services like GuardDuty, Macie, and Security Hub have their delegated administrator account configured as the "security" account. These tools can be used within each local account (including the Organization Management account) within the organization to gain account level visibility or within the Security account for Organization wide visibility. For more information about monitoring and logging refer to architecture documentation.

1.6.10. How are the perimeter firewall configurations and licensing managed after deployment?

How are the perimeter firewall configurations and licensing managed after deployment?

While you deploy the perimeter firewalls with the Accelerator you will continue to manage firewall updates, configuration changes, and license renewals from the respective firewall management interface and not from the Accelerator config file. As these changes are not managed by the Accelerator you do not need to rerun the state machine to implement or track any of these changes. You can update the AMI of the 3rd party firewalls using the Accelerator, you must first remove the existing firewalls and redeploy them (as the Elastic IP's (EIP's) will block a parallel deployment) or deploy a second parallel firewall cluster and de-provision the first cluster when ready.

1.6.11. Can the Fortinet Firewall deployments use static private IP address assignments?

Can the Fortinet Firewall deployments use static private IP address assignments?

Yes, the `"port"` stanza in the configuration file can support a private static IP address assignment from the AZ and subnet. Care must be exercised to assure the assigned IP address is within the correct subnet and availability zone. Consideration must also be given to the Amazon reserved IP addresses (first three addresses, and the last) within subnets when choosing an IP Address to assign.

Using the `config.example.json` as a reference, static IP Assignments would look like this in the `ports:` stanza of the firewall deployment.

```
"ports": [
  {
    "name": "Public",
    "subnet": "Public",
    "create-eip": true,
    "create-cgw": true,
    "private-ips": [
      {
        "az": "a",
        "ip": "100.96.250.4"
      },
      {
        "az": "b",
        "ip": "100.96.250.132"
      }
    ]
  },
  {
    "name": "OnPremise",
    "subnet": "OnPremise",
    "create-eip": false,
    "create-cgw": false,
    "private-ips": [
      {
        "az": "a",
        "ip": "100.96.250.68"
      },
      {
        "az": "b",
        "ip": "100.96.250.196"
      }
    ]
  }
  ...
],
```

Where `private-ips` are not present for the subnet or availability zone an address will be assigned automatically from available addresses when the firewall instance is created.

1.6.12. I've noticed CloudTrail logs and in certain situation VPC flow logs are stored in the centralized log-archive account logging bucket twice?

I've noticed CloudTrail logs and in certain situation VPC flow logs are stored in the centralized log-archive account logging bucket twice?

Yes. CloudTrail is configured to send its logs directly to S3 for centralized immutable log retention. CloudTrail is also configured to send it's logs to a centralized Organizational CloudWatch Log group such that the trail can be a) easily queried online using CloudWatch Insights across all AWS accounts in the organization, and b) to enable alerting based on undesirable API activity using CloudWatch Metrics and Alarms. All CloudWatch Log groups are also configured to be sent, using Amazon Kinesis, to S3 for centralized immutable log retention.

VPC flow log destinations can be configured in the config file. The example config files are set to send the VPC flow logs to both S3 and CloudWatch Logs by default for the same reasons as CloudTrail.

To reduce the duplicate long-term storage of these two specific CloudWatch Log types, customers can set `cwl-glbl-exclusions` under `central-log-services` to: `["/${ACCELERATOR_PREFIX_ND}/flowlogs/*", "/${ACCELERATOR_PREFIX_ND}/CloudTrail*"]` to prevent these specifically named log groups from being stored on S3. This setting also prevents the Accelerator from setting the customer desired log group retention period defined in the config file, once implemented, for those log groups. Therefore, we do not recommend this exception be applied during the initial installation, as the retention setting on these CWL groups will remain the default (infinite). If `cwl-glbl-exclusions` is set after initial install, the defined retention will be configured during install and will remain set to the value present when the exception was applied to those log groups. This allows logs to be stored in CloudWatch Logs for quick and easy online access (short-retention only), and stored in S3 for long-term retention and access.

Side note: CloudTrail S3 data plane logs are enabled at the Organizational level, meaning all S3 bucket access is logged. As CloudTrail is writing to a bucket within the Organization, CloudTrail itself is accessing the bucket, seemingly creating a cyclical loop. As CloudTrail writes to S3 in 5-10min batches, CloudTrail will actually only cause one extra log 'entry' every 5-10minutes and not per S3 event, mitigating major concerns. Today, with an Organization trail logging data plane events for all buckets - there is no way to exclude any one bucket. But - having clear view of who accessed/changed logs, including AWS services, is important.

1.6.13. I need a Route53 Private Hosted Zone in my workload account. How shall I proceed?

I need a Route53 Private Hosted Zone in my workload account. How shall I proceed?

The workload account requires creating a temporary local VPC before creating the Private Hosted Zone (PHZ). Creating a PHZ in Route53 requires association with a VPC. You cannot specify a shared VPC when creating the PHZ, hence the need for this workaround.

Create the temporary workload account VPC

You can create the temporary VPC during AWS account creation via the ASEA config (preferred way). Insert the "vpc" JSON object like shown below when using the ASEA config to create an AWS account.

If you don't use the ASEA config you will need to assume the proper ASEA elevated IAM role in the workload account in order to create the VPC manually.

```
"mydevacct": {
    "account-name": "MyDev1",
    "email": "dev1-main@super-corp.co",
    "src-filename": "config.json",
    "ou": "dev",
    "vpc": [
        {
            "deploy": "local",
            "name": "Local",
            "description": "This VPC Temp VPC to create the local hosted zone.",
            "cidr-src": "provided",
            "cidr": [
                {
                    "value": "192.168.100.0/24"
                }
            ],
            "region": "${HOME_REGION}"
        }
    ]
}
```

Create in the workload account a Private Hosted Zone

Using an IAM role assumed in the workload account:

List the VPCs.

```
aws ec2 describe-vpcs
```

Then retrieve the VpcId attribute for the newly created VPC as well as the Id for the shared VPC.

Create the Private Hosted Zone

```
aws route53 create-hosted-zone --name <MY_DOMAIN> --hosted-zone-config PrivateZone=true --vpc VPCRegion=<VPC_REGION>,VPCId=<VPC_ID> --
caller-reference <YOUR_REFERENCE_ID>
```

Insert the proper values for:

- `<MY_DOMAIN>`
- `<VPC_REGION>`
- `<VPC_ID>` (id of new the local VPC)
- `<YOUR_REFERENCE_ID>` (can be any value)

Take note of the newly created hosted zone id by looking at the output of the command. The Id is the value after `/hostedzone/` from the Id attribute. For example, the value is `Z0123456NWOWQ4HNN40U` from `"Id": "/hostedzone/Z0123456NWOWQ4HNN40U"`.

Create an authorization to associate with this new zone

While still in the workload account; you need to create an association request authorization to allow the shared VPC to associate with this newly created Route53 PHZ.

```
aws route53 create-vpc-association-authorization --hosted-zone-id <ZONE_ID> --vpc VPCRegion=<SHARED_VPC_REGION>,VPCId=<SHARED_VPC_ID>
```

Insert the proper values for:

- `<ZONE_ID>`
- `<SHARED_VPC_REGiON>`
- `<SHARED_VPC_ID>`

Confirm the association request for the shared VPC

After switching to an IAM role in the SharedNetwork account associate the Private Hosted Zone from the workload account.

```
aws route53 associate-vpc-with-hosted-zone --hosted-zone-id <ZONE_ID> --vpc VPCRegion=<SHARED_VPC_REGION>,VPCId=<SHARED_VPC_ID>
```

Insert the proper values for:

- `<ZONE_ID>`
- `<SHARED_VPC_REGiON>`
- `<SHARED_VPC_ID>`

Validate Association and clean-up

Back in the workload account and assuming its IAM role, validate the association using the below command. You should see two VPCs attached. The local VPC and the shared VPC.

```
aws route53 get-hosted-zone --id <ZONE_ID>
```

Insert the proper values for:

- `<ZONE_ID>`

You can now dissociate the local VPC from the zone.

```
aws route53 disassociate-vpc-from-hosted-zone --hosted-zone-id <ZONE_ID> --vpc VPCRegion=<VPC_REGION>,VPCId=<VPC_ID>
```

Insert the proper values for:

- `<ZONE_ID>`
- `<VPC_REGiON>`
- `<VPC_ID>`

You can now delete the local VPC. We recommend you leverage the ASEA configuration file. Simply the remove the `vpc` section from the workload account:

```
"mydevacct": {
    "account-name": "MyDev1",
    "email": "dev1-main@super-corp.co",
    "src-filename": "config.json",
    "ou": "dev"
}
```

and rerun the State Machine.

1.6.14. How do I create a role which has read access to the log-archive bucket to enabling log forwarding to my favorite SIEM solution?

How do I create a role which has read access to the log-archive bucket to enabling log forwarding to my favorite SIEM solution?

You can update the ASEA config file to provision an IAM role that has cross-account access to the Log Archive S3 Buckets. Attempting to do this outside the ASEA config file is blocked by security guardrails. Additionally, even if the guardrails are bypassed, it is likely the ASEA will revert any manual changes on subsequent State Machine executions. The below example creates a Lambda role which is provided permissions to Amazon OpenSearch, S3 Read Only, Lambda VPC Execution, the Log Archive S3 buckets and the KMS key. Update the below example with the least-privilege policies needed to meet the requirements of your chosen SIEM solution.

The primary trick, is the use of the `"ssm-log-archive-read-only-access": true` flag.

As we generally recommend the SIEM be deployed into the Operations account, add the following to the roles array within the **Operations** account section in the ASEA config file:

```
{
  "role": "SIEM-Lambda-Processor",
  "type": "lambda",
  "ssm-log-archive-read-only-access": true,
  "policies": [
    "AmazonOpenSearchServiceFullAccess",
    "service-role/AWSLambdaVPCAccessExecutionRole",
    "AmazonS3ReadOnlyAccess"
  ],
  "boundary-policy": "Default-Boundary-Policy"
}
```

1.6.15. How do I create a role for use by Azure Sentinel using the new S3 Connector method?

How do I create a role for use by Azure Sentinel using the new S3 Connector method?

This process is very similar to FAQ #1.6.14, except we need to allow for a cross-cloud role assumption. This will be done in the Log Archive account, instead of the Operations account.

The following config snippet should be added to the roles array within the **Log Archive** account section in the ASEA config file:

```
{
  "role": "MicrosoftSentinelRole",
  "type": "account",
  "ssm-log-archive-read-only-access": true,
  "policies": [
    "AmazonSQSReadOnlyAccess",
    "service-role/AWSLambdaSQSQueueExecutionRole",
    "AmazonS3ReadOnlyAccess"
  ],
  "boundary-policy": "Default-Boundary-Policy",
  "trust-policy": "sentinel-trust-policy.json",
  "source-account": "log-archive",
  "source-account-role": "OrganizationAccountAccessRole"
}
```

- The value of the `source-account-role` above needs to be replaced with the value of `organization-admin-role` from your config file (OrganizationAccountAccessRole, AWSCloudFormationStackSetExecutionRole, or AWSControlTowerExecution).

The above role uses a custom trust policy, and also requires a file of the name `sentinel-trust-policy.json` be placed into the `iam-policy` folder of the customers S3 input bucket. This file must contain the following text:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::197857026523:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "{CUSTOMER-VALUE-HERE}"
        }
      }
    }
  ]
}
```

- The IAM account number listed above is a value provided by Microsoft in their documentation (hard-coded to the same value for all customers).
- The value of `sts:ExternalId`, shown as `{CUSTOMER-VALUE-HERE}` above, must be replaced with the ID of the Log Analytics Workspace in your Azure tenant.
- This information is based on the requirements published here as of 2022-03-10.

1.6.16. Does the ASEA include a full SIEM solution?

Does the ASEA include a full SIEM solution?

We've found a diverse set of differing customer needs and requirements across our customer base. The ASEA:

- enables AWS security services like Amazon GuardDuty (a Cloud native IDS solution) and centralizes the consoles of these tools in the Security account;
- audits the entire environment for compliance and consolidates findings from AWS security services in the Security Hub console in the Security account;
- sends prioritized email alerts for Security Hub Findings, Firewall Manager alerts and customizable CloudWatch Alarms;
- centralizes logs across the environment in a central bucket in the Log Archive account;
- in addition, retains logs locally in CloudWatch Logs for simple query using CloudWatch Insights.

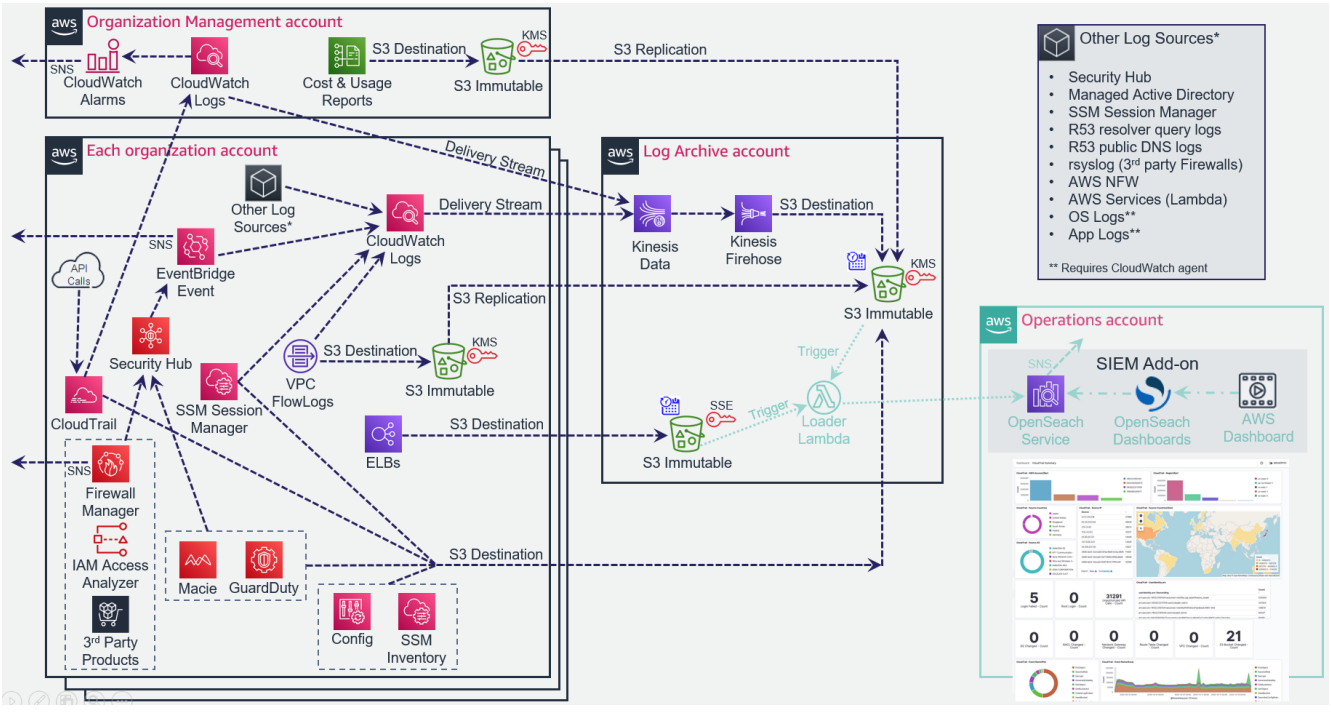
This makes it extremely simple to layer a customer's preferred SIEM solution on top of the ASEA, enabling easy consumption of the comprehensive set of collected logs and security findings.

Customers ask for examples of what this integration looks like. We've also had a number of customers ask for a reasonably functional and comprehensive open source SIEM-like solution to provide more advanced dashboarding, log correlation and search capabilities.

While not a part of the ASEA, we've made the SIEM on Amazon OpenSearch Service available as an ASEA **Add-on** to satisfy these requirements.

This independent solution can easily and quickly be deployed on top of the ASEA by following the documentation and using the scripts available here. This process takes less than an hour.

The overall logging architecture is represented in this diagram:



1.6.17. Why are only select interface endpoints provisioned in the sample configuration files?

Why are only select interface endpoints provisioned in the sample configuration files?

For economic reasons, most of the sample configuration files only include the following minimum set of required interface endpoints:

"ec2", "ec2messages", "ssm", "ssmmessages", "secretsmanager", "cloudformation", "kms", "logs", "monitoring"

The full sample configuration file included all interface endpoints that existed in the Canada (Central) region at the time the configuration file was originally developed:

"access-analyzer", "acm-pca", "application-autoscaling", "appmesh-envoy-management", "athena", "autoscaling", "autoscaling-plans", "awsconnector", "cassandra", "clouddirectory", "cloudformation", "cloudtrail", "codebuild", "codecommit", "codepipeline", "config", "datasync", "ebs", "ec2", "ec2messages", "ecr.api", "ecr.dkr", "ecs", "ecs-agent", "ecs-telemetry", "elasticbeanstalk", "elasticbeanstalk-health", "elasticfilesystem", "elasticloadbalancing", "elasticmapreduce", "email-smtp", "events", "execute-api", "git-codecommit", "glue", "kinesis-firehose", "kinesis-streams", "kms", "license-manager", "logs", "macie2", "monitoring", "notebook", "sagemaker.api", "sagemaker.runtime", "secretsmanager", "servicecatalog", "sms", "sns", "sqs", "ssm", "ssmmessages", "states", "storagegateway", "sts", "synthetics", "transfer", "transfer.server", "workspaces"

Since that time these additional endpoints have been launched in the ca-central-1 region and can be optionally added to customer configuration files to make them accessible from private address space:

"airflow.api", "airflow.env", "airflow.ops", "app-integrations", "appstream.api", "appstream.streaming", "auditmanager", "backup", "backup-gateway", "batch", "cloudhsmv2", "codedeploy", "codedeploy-commands-secure", "codestar-connections.api", "comprehend", "comprehendmedical", "databrew", "dms", "elasticache", "emr-containers", "finspace", "finspace-api", "fis", "fsx", "greengrass", "imagebuilder", "inspector2", "iot.data", "iot.fleethub.api", "iotsitewise.api", "iotsitewise.data", "kendra", "lakeformation", "lambda", "memory-db", "mgn", "models-v2-lex", "nimble", "panorama", "profile", "qldb.session", "rds", "rds-data", "redshift", "redshift-data", "rekognition", "runtime-v2-lex", "sagemaker.featurestore-runtime", "securityhub", "servicecatalog-appregistry", "ssm-contacts", "ssm-incidents", "sync-states", "textract", "transcribe", "transcribestreaming", "translate", "xray"

The aws.sagemaker.ca-central-1.studio interface endpoint was also launched, but cannot be auto-deployed by the Accelerator at this time as it does not utilize standardized naming and requires a code update to enable deployment.

Additional endpoints may exist in other AWS regions. Any endpoint can be added to any Accelerator configuration file, as long as it follows the standardized endpoint naming convention (e.g. com.amazonaws.{region}.{service}).

1.6.18. How can centralized EC2 patch management be deployed?

How can centralized EC2 patch management be deployed?

With Quick Setup, a capability of AWS Systems Manager, you can create patch policies powered by Patch Manager. A patch policy defines the schedule and baseline to use when automatically patching your Amazon Elastic Compute Cloud (Amazon EC2) instances and other managed nodes. This solution needs modification to deploy into the ASEA. See the guide here to learn how.

4.7 1.7. Network Architecture

1.7.1. We want to securely connect our on-premises networks/datacenters to our AWS Cloud PBMM tenancy, what does AWS you recommend?

We want to securely connect our on-premises networks/datacenters to our AWS Cloud PBMM tenancy, what does AWS you recommend?

We recommend customers create a new AWS sub-account in your organization in the Infrastructure OU to “own” the Direct Connect (DX), segregating Direct Connect management and billing from other organization activities. Once provisioned you would create a Public VIF on the DX in this account. You can also create additional Private VIF’s when and if required, and share them directly with any sub-account that needs to consume them.

We recommend customers then inter-connect directly to the Transit Gateway, in the Shared Network sub-account, from your on-premises network/datacenters.

- Initiate IPSec VPN tunnels from on-premises to the TGW using BGP w/ECMP to scale and balance the traffic. Equal Cost Multi-Pathing (ECMP) is used to balance the traffic across the available VPN tunnels.
- You need to create as many VPN attachments to the TGW as is required to meet your bandwidth requirements or DX capacity. Today IPSec attachments are limited to 1.25 Gbps each (10 Gbps would require 8 attachments) and is scalable to 50 Gbps.
- Each VPN attachment would comprise two tunnels (active/passive), each connecting to a different on-premises firewall/ VPN appliance.

The VPN attachments would then be connected to an appropriately configured route table on the TGW. TGW route tables provide VRF like segregation capabilities, allowing customers to control which of their cloud based networks are allowed to communicate on-premises, or visa-versa.

This architecture is fully managed and easy to manage, highly available, scalable, cost effective, and enables customers to reserve all their 3rd party Perimeter firewall capacity for public or internet facing traffic.

(This guidance will be updated once MACSEC is broadly available across AWS transit centers)

1.7.2. Does this configuration violate PBMM / ITSG-22/38/33 principals?

Does this configuration violate PBMM / ITSG-22/38/33 principals?

No. Data center interconnects are not zoning boundaries (or ZIPs). Additionally, in many cases the on-premises VPN termination device used to interconnect to the cloud either contains, or is placed in-line with firewall and/or inspection devices. Customers insistent on placing a firewall between datacenters can enable the appropriate filtering or inspection on these on-premise devices. Enabling the same capabilities inside AWS would mean a customer is inspecting both ends of the same wire, a pointless activity. The TGW approach is being used by several gov’t PBMM customers.

Additionally, it should be noted that workloads in all the AWS accounts are fully protected using AWS Security Groups (stateful firewalls) wrapped around each and every instance comprising a workload.

1.7.3. Why do you NOT recommend using a VGW on the perimeter VPC?

Why do you NOT recommend using a VGW on the perimeter VPC?

The VGW solution was not designed to support an enterprise cloud environment – it was designed to provide single VPC connectivity. The VGW solution offers lower availability than other options as it relies on VPC route tables to steer traffic, which need to be updated using custom scripts in the event the failure of an appliance or availability zone. The VGW solution is typically harder to maintain and troubleshoot. The VGW solution has limited scalability, as the VGW only supports a single active connection and does not support BGP or ECMP (i.e. supports a maximum bandwidth of 1.25Gbps). Most customers providing enterprise cloud connectivity have switch away from this approach. This approach is highly discouraged.

1.7.4. Why do you NOT recommend connecting directly to the 3rd party firewall cluster in the perimeter account? (not GWLB, not NFW)

Why do you NOT recommend connecting directly to the 3rd party firewall cluster in the perimeter account? (not GWLB, not NFW)

This approach was common with AWS customers before the TGW was introduced, with many customers upgrading or considering upgrading to the TGW approach. We also have some customers using this architecture based on a very specific limitation of the customer's Direct Connect architecture, these customers would also like to migrate to the TGW approach, if they could.

While viable, this approach adds unneeded complexity, reduces cloud availability, is expensive to scale, and reduces bandwidth to internet facing workloads. This solution doubles the IPSec VPN tunnels using BGP w/ECMP requirements as it needs tunnels on both sides of the firewall. In this configuration each firewall appliance typically only provides a single pair of IPSec connections supporting marginally more bandwidth than the TGW VPN attachments. Adding tunnels and bandwidth requires adding firewall appliances. Stateful capabilities typically need to be disabled due to performance and asymmetric routing challenges. This typically means a very expensive device is being deployed inside AWS simply to terminate a VPN tunnel.

1.7.5. What if I really want to inspect this traffic inside AWS, but like the TGW architecture?

What if I really want to inspect this traffic inside AWS, but like the TGW architecture?

Customers who insist on inspecting the ground to cloud traffic inside AWS *can* do this with the proposed TGW architecture. The TGW route tables can be adjusted to hairpin the traffic through either a dedicated Inspection VPC, or to the Perimeter account firewall cluster for inspection. The Inspection VPC option could leverage 3rd party firewalls in an autoscaling group behind a Gateway Load Balancer, or leverage AWS network firewall to inspection traffic. To maximize internet throughput, the Inspection VPC option is generally recommended. While we do not feel inspection is needed in this situation, it is possible.

1.7.6. What does the traffic flow look like for an application running in a workload account?

What does the traffic flow look like for an application running in a workload account?

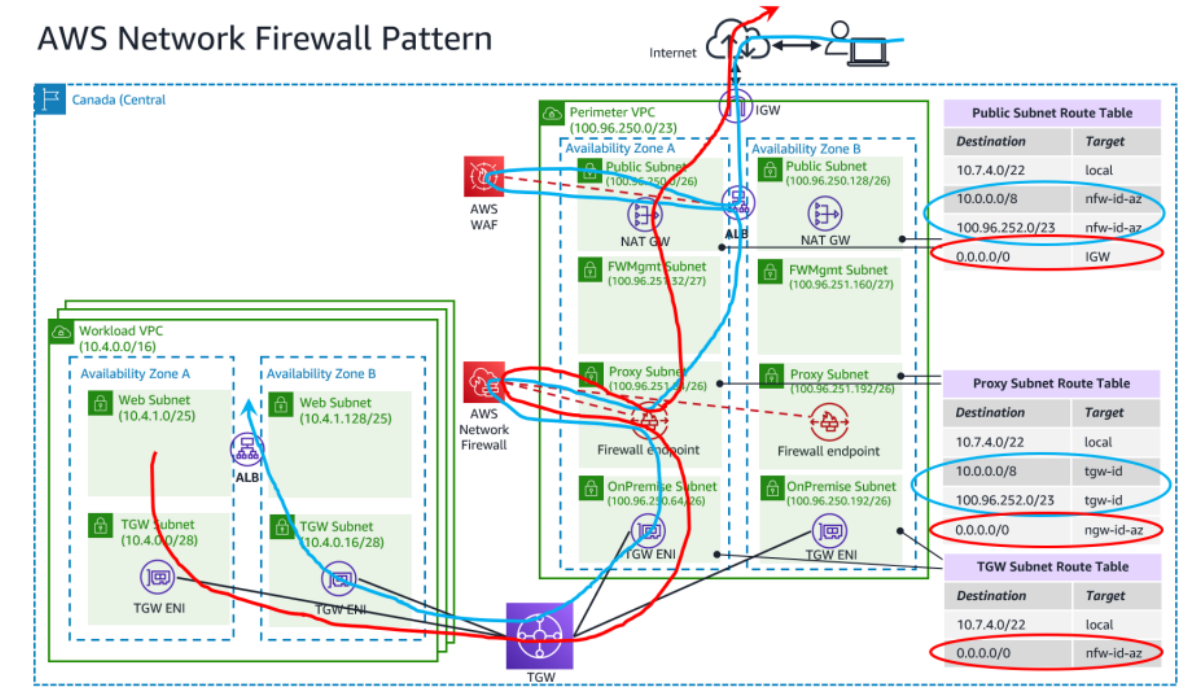
The perimeter (ingress/egress) account typically contains two ALB's, one for production workloads and another for Dev/Test workloads. The Dev/Test ALB should be locked to restrict access to on-premises users (using a security group) or have authentication enabled to prevent Dev/Test workloads from being exposed to the internet. Additionally, each workload account (Dev/Test/Prod) contains a local (back-end) ALB.

AWS Web Application Firewall (WAF) should be enabled on both front-end and back-end ALB's. The Front-end WAF would contain rate limiting, scaling and generic rules. The back-end WAF would contain workload specific rules (i.e. SQL injection). As WAF is essentially a temporary fix for broken applications before a developer can fix the issue, these rules typically require the close involvement of the application team. Rules can be centrally managed across all WAF instances using AWS Firewall Manager from the Security account.

The front-end ALB is then configured to target the back-end ALB using the process described in the Post Installation section of the installation guide, step 2 `(Configure the new alb-forwarding feature (added in v1.5.0)`. This enables configuring different DNS names and/or paths to different back-end ALB's using the ASEA's alb-forwarder. We recommend moving away from the NAT to DNS mechanism used in previous released as it was too complex, does not work with bump-in-the-wire inspection devices (NFW, GWLB), and only available on a limited number of 3rd party firewalls.

This implementation allows workload owners to have complete control of workloads in a local account including the ELB configuration, and allow site names and paths to be defined and setup at sub-account creation time (instead of during development) to enable publishing publicly or on-premises in a rapid agile manner.

This overall flow is depicted in this diagram:



NOTE1: Distinct route tables required per AZ, which targets the local AZ's nfw, gwlb or ngw endpoint

1.7.7. How does CloudFront and API Gateway fit with the answer from question 1.7.6?

How does CloudFront and API Gateway fit with the answer from question 1.7.6?

The perimeter account is focused on protecting legacy IaaS based workloads. Cloud Native applications including CloudFront and API Gateway should be provisioned directly in the same account as the workload and should NOT traverse the perimeter account.

These services must still be appropriately configured. This includes ensuring both WAF and logging are enabled on each endpoint.

The GC guidance on Cloud First patterns and anti-patterns can be downloaded here.

5. Operations & Troubleshooting

5.1 Accelerator Operations & Troubleshooting Guide

This document is targeted at individuals installing or executing the AWS Secure Environment Accelerator. It is intended to guide individuals who are executing the Accelerator by providing an understanding as to what happens at each point throughout execution and to assist in troubleshooting state machine failures and/or errors. This is one component of the provided documentation package and should be read after the Installation Guide, but before the Developer Guide.

5.2 1. System Overview

This document is targeted at individuals installing or executing the AWS Secure Environment Accelerator. It is intended to guide individuals who are executing the Accelerator by providing an understanding as to what happens at each point throughout execution and to assist in troubleshooting state machine failures and/or errors. This is one component of the provided documentation package and should be read after the Installation Guide, but before the Developer Guide.

5.2.1 1.1. Overview

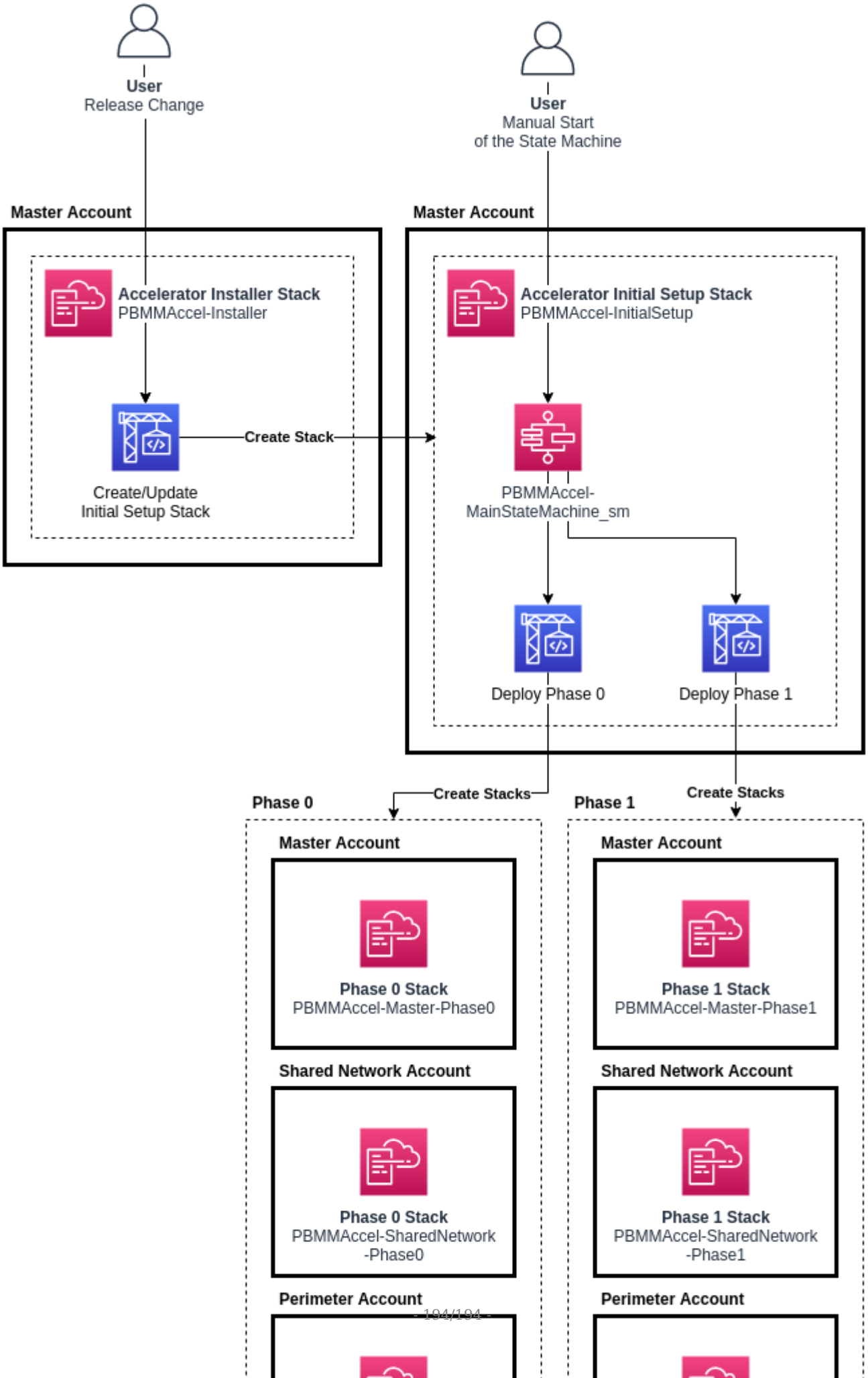
The system can be thought of in two levels. The first level of the system consists of Accelerator stacks and resources. Let's call these the Accelerator-management resource. The second level of the system consists of stacks and resources that are deployed by the Accelerator-management resource. Let's call these the Accelerator-managed resources. The Accelerator-management resources are responsible for deploying the Accelerator-managed resources.

There are two Accelerator-management stacks:

- the `Installer` stack that is responsible for creating the next listed stack;
- the `Initial Setup` stack. This stack is responsible for reading configuration file and creating Accelerator-managed resources in the relevant accounts.

There are multiple Accelerator-managed stacks. Currently there are as many as twelve Accelerator-managed stacks per managed account.

The figure below shows a zoomed-out overview of the Accelerator. The top of the overview shows the Accelerator-management resources, i.e. the `Installer` stack and the `Initial Setup` stack. The bottom of the overview shows the Accelerator-managed resources in the different accounts.


User
Release Change
User
Manual Start
of the State Machine
Master Account
Accelerator Installer Stack
PBMMAccel-Installer
Create Stack
Create/Update
Initial Setup Stack
Master Account
Accelerator Initial Setup Stack
PBMMAccel-InitialSetup
PBMMAccel-
MainStateMachine_sm
Deploy Phase 0
Deploy Phase 1
Create Stacks
Create Stacks
Phase 0
Phase 1
Master Account
Phase 0 Stack
PBMMAccel-Master-Phase0
Shared Network Account
Phase 0 Stack
PBMMAccel-SharedNetwork
-Phase0
Perimeter Account
Master Account
Phase 1 Stack
PBMMAccel-Master-Phase1
Shared Network Account
Phase 1 Stack
PBMMAccel-SharedNetwork
-Phase1
Perimeter Account