1. Accelerator Basic Operation and Frequently asked Questions

1.1. Operational Activities

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

- We offer two options and both can be used in the same deployment:
  - 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+pbmmT-funacct@example.com",
  "src-filename": "config.json",
  "ou": "Sandbox"
}
```

  - 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**.
    - ** **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`, ***prior to v1.2.5, the ONLY supported value is `AWSCloudFormationStackSetExecutionRole`***, otherwise we cannot bootstrap the account.
    - 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

1.1.2. Can I use AWS Organizations for all tasks I currently use AWS Organizations for? (Standalone Version Only)

- 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 controlled SCP's

  - add/remove SCP's on top-level OU's (these are Accelerator controlled) or specific accounts that have Accelerator controlled SCPs
    * users can change SCP's on non-top-level ou's and non-Accelerator controlled accounts as they please
  - 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

1.1.3. 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.

1.1.4. 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.

1.1.5. 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 1.4. 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

1.1.6. 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 scenario's 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!

1.1.7. 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 there 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. It's 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. We will be improving documentation as to which SCPs deliver security functionality versus those protecting the Accelerator itself in a future release.

1.1.8. 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.

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).

1.1.9. How do I suspend an AWS account?

- Prior to v1.2.4, suspending accounts were blocked via SCP:
  - a defect exists in prior releases which could cause SM failures after an account was suspended
  - this required modifications to both the Part1 and Part2 SCPs
- To suspend an account in v1.2.4 and above, follow this process:
  - the AWS account must remain in the source OU
  - login to account to be suspended as the account root user
  - suspend the account through `My Account`
  - Run 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

1.1.10. I need a new VPC, where shall I define it?

You can define a VPC in one of three 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.

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 VPC (as deployed in the Dev.Test/Prod OU accounts) can be defined in an OU. It should be noted that local VPC's will each be deployed with the same CIDR ranges (at this time) and therefor should not be connected to a TGW.

As mandatory accounts often have unique configuration requirements, VPC's like the Endpoint VPC, are configured within the mandatory account configuration. Customers can also define VPC's within each workload account configuration, but this requires editing the configuration file for each account configuration.

1.1.11. 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 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)

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 PBMM 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.

1.2. 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)?

- 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?

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). Additionally, please reference section 4 of the Instation and Upgrade Guide. It may be easier to migrate AWS accounts to a new Accelerator Organization, per the process detailed in FAQ #1.2.3.

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?

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.

1.3. End User Enviroment

1.3.1. Is there anything my end users need to be aware of?

CloudWatch Log group deletion is prevented for security purposes. Users of the Accelerator environment will need to ensure they set CFN stack Log group retention type to RETAIN, or stack deletes will fail when attempting to delete a stack and your users will complain.

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?

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 customers 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.

1.4. Upgrades

1.4.1. 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.

1.4.2. After upgrading to v1.3.0, 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 section 2 of the `Accelerator Configuration File Customization` Guide for more details.

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?

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?

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?

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?

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 customers 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?

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.

1.6. Deployed Functionality

1.6.1. I wish to be in compliance with the 12 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?

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?

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 (usally 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?

The rsyslog servers are included to accept logs for appliances and third party applications that do not natively support the CloudWwatch Agent from any account within a customers Organization. These logs are then immediatly 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?

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.

1.6.6. 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 (30-days in the sample config files). This ensures these instances are always clean. Additionally, if the Accelerator State Machine has been run in the previous 30-days, the ASG will have also been updated with 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.

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?

The Accelerator provided sample security groups

Security group egress rules are often used in 'allow all' mode (`0.0.0.0/0`), with the focus primarily being on consistently whitelisting 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?

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?

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.

...Return to Accelerator Table of Contents