1. Accelerator Configuration File Customization and Sample Configs

1.1. Sample Accelerator Configuration Files

- Sample config files can be found in this folder
- Unsure where to start, use the config.lite-example.json file

Samples with Descriptions:

1. **Full PBMM configuration** file (`config.example.json`)
   - The full PBMM 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.
2. **Light weight PBMM configuration** file (`config.lite-example.json`) **(Recommended for most new PBMM customers)**
   - To reduce solution costs and allow customers to grow into more advanced AWS capabilities, we created this lighter weight configuration that does not sacrifice functionality, but could limit performance. This config file:
     - 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
     - removes the perimeter VPC Interface Endpoints
     - reduces the Fortigate instance sizes from c5n.2xl to c5n.xl (VM08 to VM04)
     - removes the Unclass ou and VPC
   - The Accelerator allows customers to easily add or change this functionality in future, as and when required without any impact
3. **Ultra-Light sample configuration** file (`config.ultralite-example.json`)
   - 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 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
4. **Multi-Region sample configuration** file (`config.multi-region-example.json`)

- This configuration file was created to represent a more advanced multi-region version of the Full PBMM configuration file from bullet 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)

5. **Test PBMM configuration** file (`config.test-example.json`) **(Use for testing PBMM configuration)**
   - Further reduces solution costs, while demonstrating full solution functionality (NOT recommendend for production). This config file:
     - uses the Light weight PBMM configuration as the starting point
     - 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
     - reduces the Fortigate instance sizes from c5n.xl to c5.xl
     - reduces the rsyslog and RDGW instance sizes from t2.large to t2.medium
     - removes the second rsyslog node
     - 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
     - The most expensive individual component of this sample is the perimeter 3rd party firewalls
       - this example will be updated in the near future, removing the 3rd party firewalls
       - we will add a NATGW for egress. For ingress, customers will need to manually target the perimeter ALB to point to each backend-ALB's IP's and manually update the IP's when they change (the next major SEA code release will include functionality to automate this capability)

1.2. Deployment Customizations

- 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. This is documented
- 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 here
- 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

1.3. 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)
- 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 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 ALZ version of the Accelerator does not support sub-ou.
- v1.1.5 and above adds support for customer provided YAML config file(s) as well as JSON. 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 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.

1.4. 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.

2. NEW: 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.

...Return to Accelerator Table of Contents