AWS Secure Environment Accelerator (ASEA)

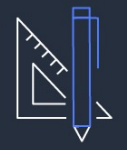
Introductory workshop

Agenda

- The high-level solution
- AWS Secure Environment Accelerator (ASEA) Architecture Review
- List of Requirements
- The main stages of installation
- What is a “State Machine”
- How AWS SEA Accelerator State Machine works
- Demo & review of the file “config.json”

The high-level solution

Feature Summary (1/3)



- **AWS Organization with Multiple Accounts:**
  - An AWS Organization is a grouping structure for a number of separate AWS accounts that are controlled by a single client entity. This provides consolidated billing, organizational units, and facilitates the deployment of pan-organizational security controls such as AWS CloudTrail logs, VPC flow logs, and service control policies. Separate accounts provide solid isolation of the control plane and data plane between workloads and/ or environments.
- **Encryption:**
  - AWS KMS with customer-managed CMK keys is widely used for all data stored at rest, in S3 buckets, EBS volumes, RDS encryption.
- **Services Control Policies:**
  - SCP provides a security check mechanism primarily used to deny entire categories of API operations at an AWS account, organizational unit, or organization level. These can be used to ensure that workloads are deployed only in prescribed regions, to ensure that only whitelist services are used, or to prevent the disabling of detection/prevention controls. Normative SCPs are provided.

aws

Feature Summary (2/3)

- **Centralized and isolated network:**
  - Virtual Private Clouds (VPCs) are used to create a data plane isolation between workloads, centralized in a shared network account. Connectivity to on-premises environments, Internet exit, shared resources, and AWS APIs are publicized at a central point of entry and exit through the use of Transit Gateway, site-to-site VPN, next-generation firewall, and AWS Direct Connect (if applicable).
- **Centralized DNS Management:**
  - Amazon Route 53 is used to provide unified public and private hosted zones in the cloud environment. Inbound and outbound Amazon Route 53 resolvers extend this unified view of DNS to local networks.
- **Full logging:**
  - AWS CloudTrail logs are enabled across the organization to ensure auditability in the cloud environment. Amazon CloudWatch is used to log application activities, as well as information from VPC flows, and these are centralized. Deletion is prevented via SCPs.

aws

Feature Summary (3/3)

- **Detective Security Checks:**
  - Potential security threats have emerged in the cloud environment through the automatic deployment of detection security controls such as Amazon GuardDuty, AWS Config, and AWS Security Hub.
- **Single Sign-On (SSO)**:
  - AWS SSO manses user access and permissions for all of your accounts in AWS Organizations centrally. AWS SSO automatically configures and manages all necessary permissions for your accounts, without requiring additional configuration in individual accounts.

aws

Automated implementation of technical Guardrail

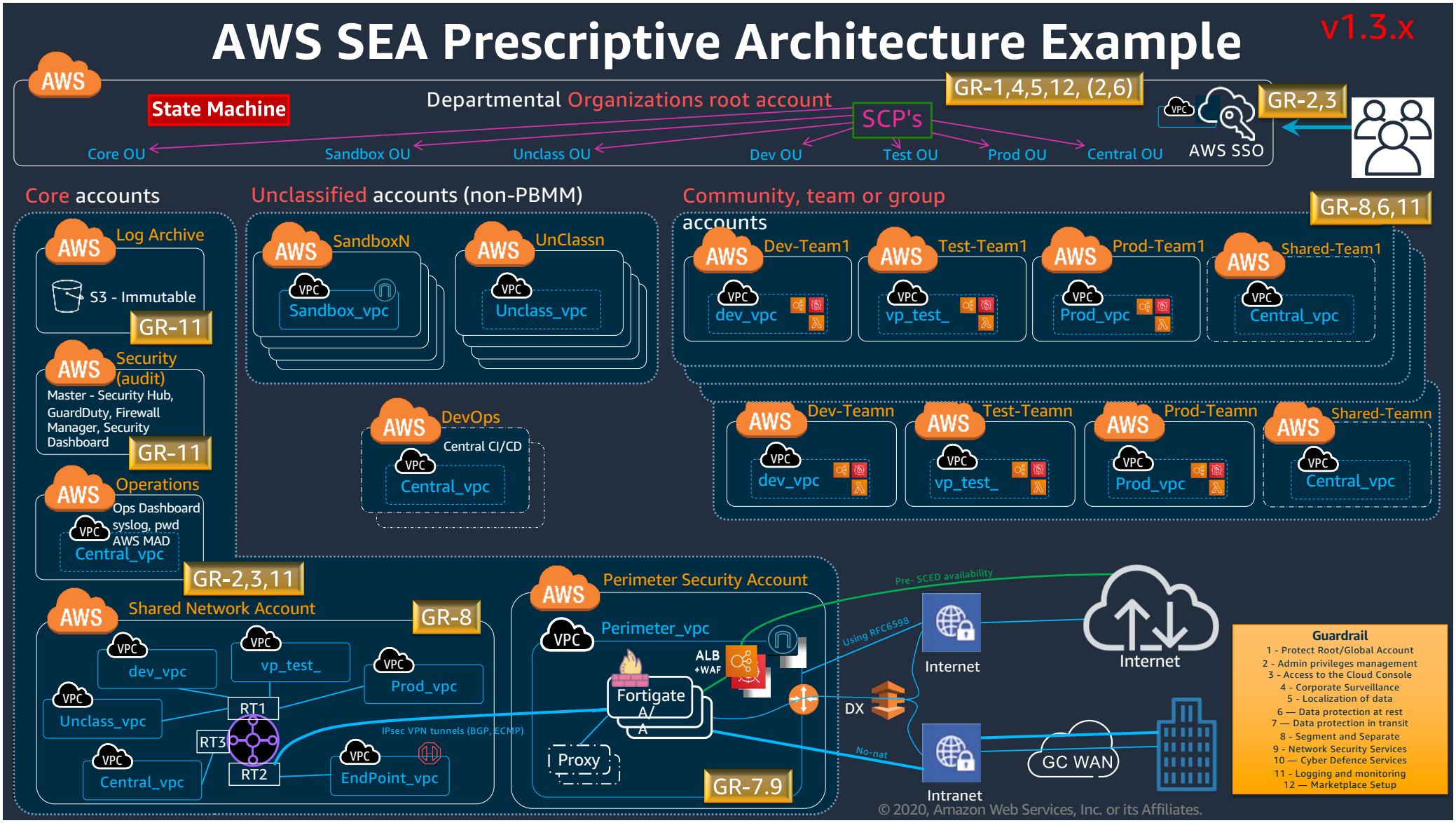
#	Guardrail	Coverage of AWS SEA
1	Protect Root/Global Administrator Account	✓
2	Managing Administrative Privileges	✓
3	Access to the Cloud Console	✓
4	Corporate Supervisory Accounts	✓
5	Location of data	✓
6	Data protection at rest	✓
7	Data Protection in Transit	✓
8	Segment and Separate	✓
9	Network Security Services	✓
10	Cyber Defense Services	✓
11	Logging and monitoring	✓
12	Marketplace Setup	✓

* Some railings require manual tracking

aws

AWS SEA Prescriptive Architecture Example

v1.3.x

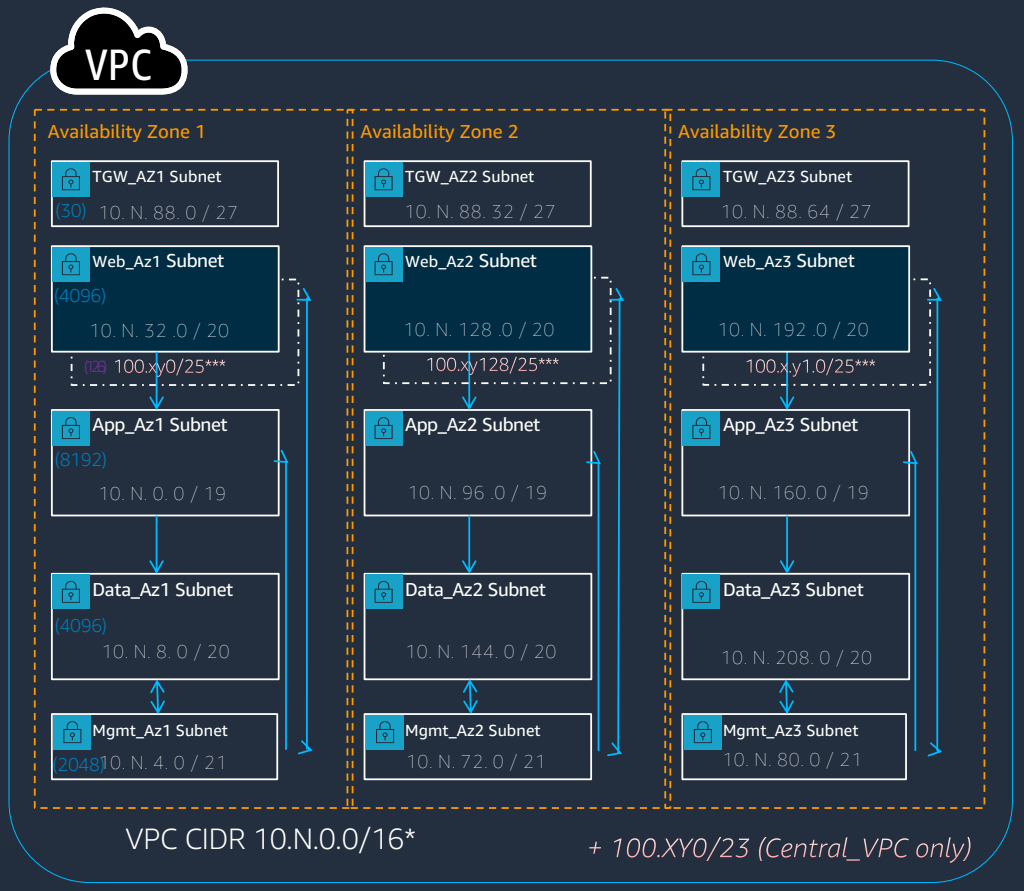


AWS Accelerator Standard VPC Design

v1.3.x

(Used for Clearing, Dev, Test, Prod, Central VPCs) - Class B

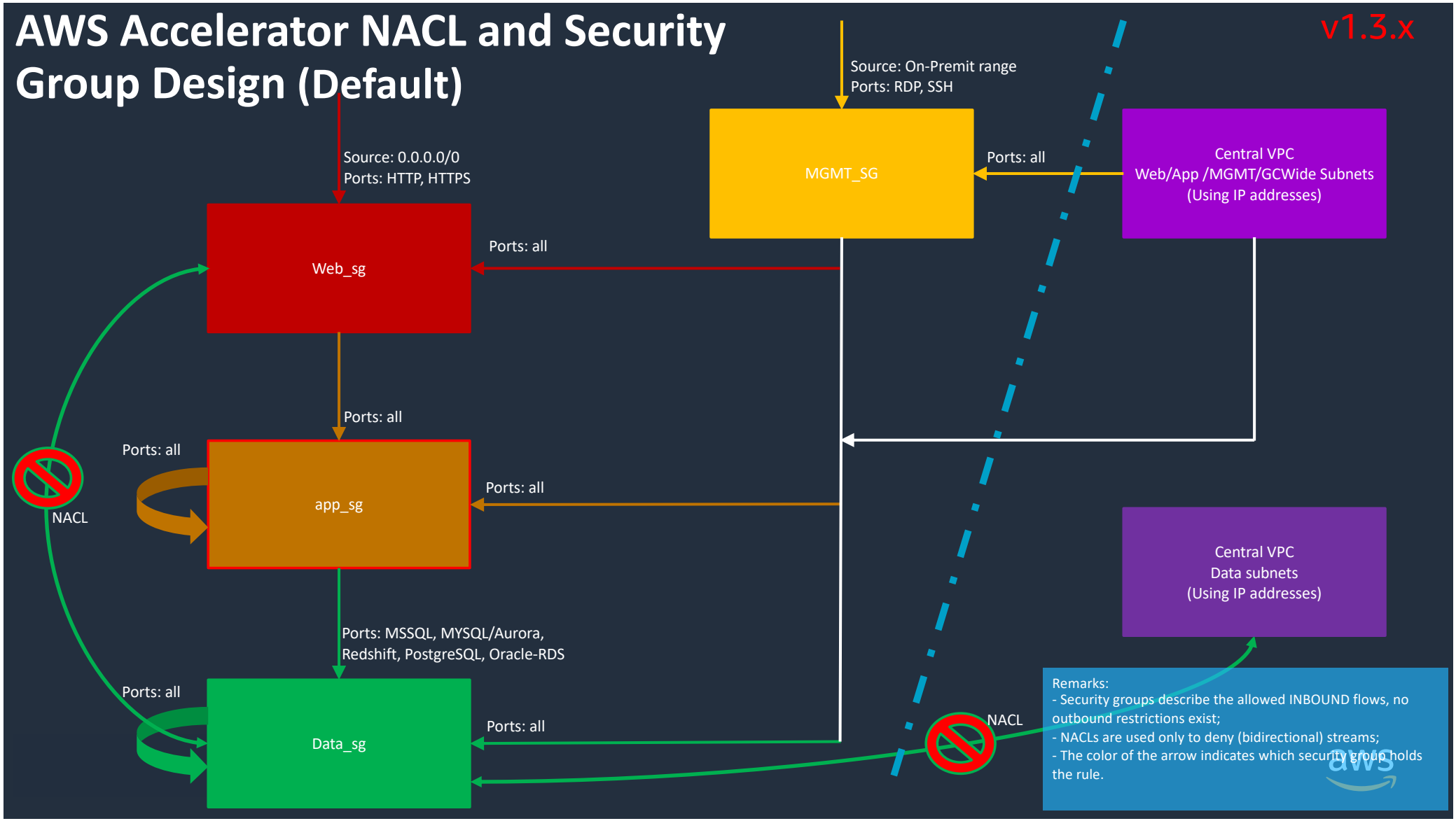
(The half-class B option exists)



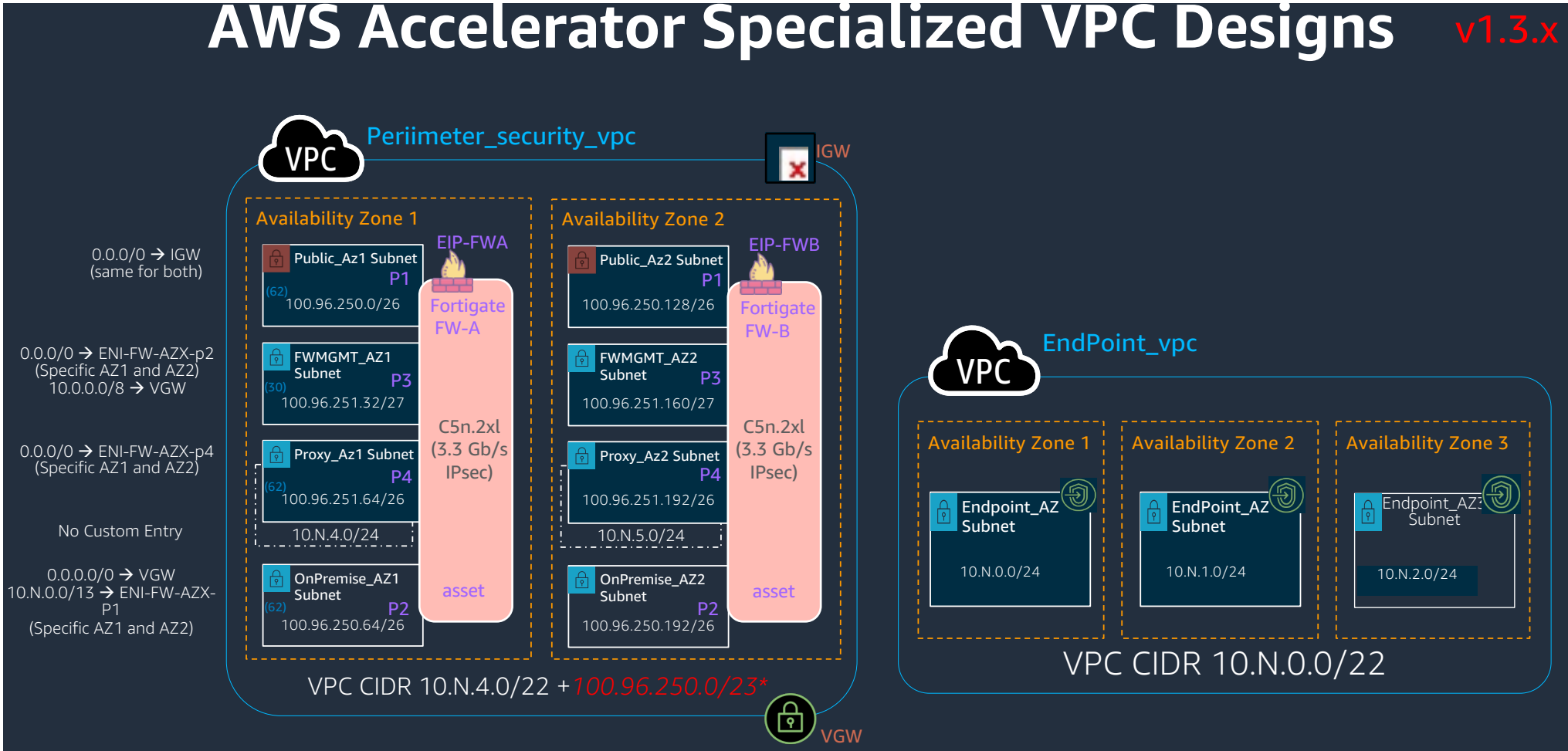
NOTE: Subnets are NOT ZIPs. Security groups are used as a zoning boundary/ZIP. This design takes advantage of the concept of many Micro-Zip, potentially one per application, per zone.

NOTE: TGW subnets are not shared. Sandbox_VPCRemoves TGW subnets, web subnets become public with IGW and NATGW for private subnets. Subnets of the central VPC RFC6598 namedSGCwide_AZX.

* We assign a full /16 to each VPC (i.e. 10.10.0.0/16 for Dev, 10.11.0.0/16 for Test, etc.). The client may optionally use other CIDR blocks RFC1918. It is essential that these CIDR ranges do not conflict with CIDR ranges in an on-premise department because there is NO Nat'ing for floor-cloud communications (mark as "used for cloud" in the departmental on-premises IPAM system).
** Note: 10.N.224.0/19, 10.N.88.96-10.N.95.255, and 100.x.y1.128/25 are available for future assignment.
*** Central VPC CIDR has been extended with an RFC6598 CIDR range (internal web subnets) to host MAD and other services that may require cross-department access.


AWS Accelerator NACL and Security Group Design (Default)
v1.3.x
Source: On-Premit range
Ports: RDP, SSH
MGMT_SG
Ports: all
Central VPC
Web/App /MGMT/GCWide Subnets
(Using IP addresses)
Source: 0.0.0.0/0
Ports: HTTP, HTTPS
Web_sg
Ports: all
Ports: all
Ports: all
app_sg
Ports: all
NACL
Central VPC
Data subnets
(Using IP addresses)
Ports: MSSQL, MYSQL/Aurora,
Redshift, PostgreSQL, Oracle-RDS
Ports: all
Data_sg
Ports: all
NACL
Remarks:
- Security groups describe the allowed INBOUND flows, no outbound restrictions exist;
- NACLs are used only to deny (bidirectional) streams;
- The color of the arrow indicates which security group holds the rule.
aws

AWS Accelerator Specialized VPC Designs v1.3.x



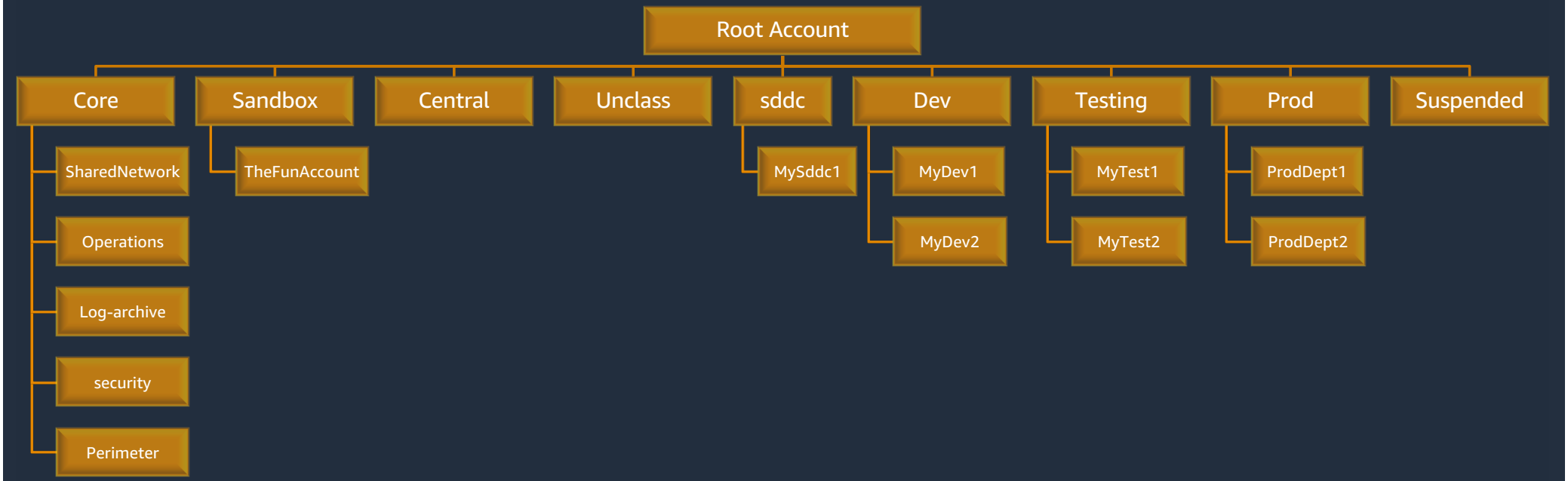
* 100.96.250.0/23 is an example of block RFC6598, customers must each use their own block assigned by SSC. Departments also need SSC to assign unique PMO DSOs.
** Note: 10.n.4.0/22 should be used to create a VPC because you cannot extend a 100 subnet block. *, this is a FortiSandbox detonation subnet
*** Additional 100.96.252.0/23 required for the overlay network (Fortigates inside the VPN tunnel). Before GCCAP is available, the public subnet will contain ELBs for public applications.
**** Remaining available addresses: 100.96.251.0/27 and 100.96.251.128/27 (32 per AZ)

List of Requirements

AWS SEA Configuration for Production - Required Information

THIS REQUIRES EXTENSIVE PREPARATION AND PLANNING

AWS Organizations



SEA Configuration for Production - Information Required

THIS REQUIRES EXTENSIVE PREPARATION AND PLANNING

1. Plan the structure of your OU, we suggest you:
    - core, Central, Sandbox, Unclass, Dev, Test, Prod
2. 6* RFC1918 Class B or Half Class B (CIDR) address blocks that do not conflict with your on-premise networks
    - (one for each OR, except Sandbox which is not routable)
    - “Main” Class B range will be divided to support perimeter VPC Endpoint and VPC
3. 1 * Address blocks RFC6598/23
4. 2* ASN BGP (TGW, FW Cluster) (a third is required if you are deploying a VGW for DX connectivity)
5. A unique Windows domain name (deptaws/dept.aws, deptcloud/dept.cloud, etc.)
6. DNS domain names and DNS server IP addresses for on-premises private DNS zones that require cloud resolution
7. DNS domain for a public zone hosted in the “public” cloud: ["dept.cloud-nuage.canada.ca"]
8. DNS domain for a private zone hosted in the “private” cloud: ["dept.cloud-nuage.gc.ca"]
9. Wildcard TLS certificate for each of the previous 2 zones
10. 2 * Fortinet FortiGate firewall licenses
11. We also recommend at least 20 unique email ALIASES associated with a single mailbox, never used before to open AWS accounts, so you don't need to request new email aliases every time you need to create a new AWS account.

The main stages of installation

Basic Requirements

❖ Deploying AWS SEA requires support from your on-premises AWS account team. Attempts to deploy the accelerator without the support of your AWS SA, TAM, ProServe, or installation will fail because new AWS accounts do not have appropriate quotas set up to facilitate installation.

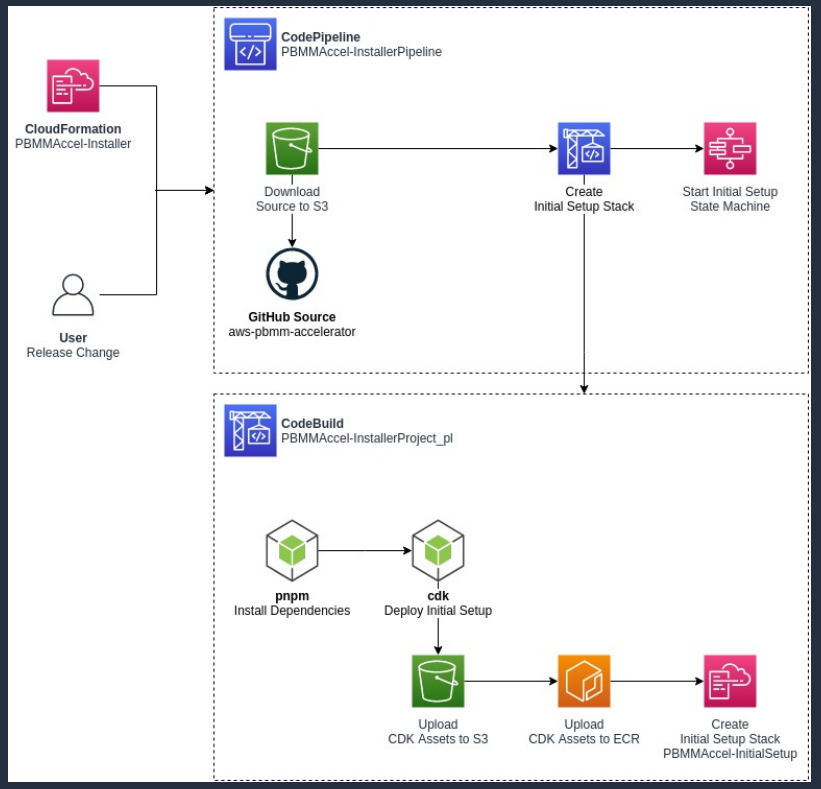
❖ Installing AWS SEA native architecture is prescriptive. It requires an increase in your quotas to support a minimum of 6 AWS accounts in AWS Organizations plus all additional workload accounts required.

The main stages of installation

1. Create the various configuration files and drop them into an S3 Bucket
2. Starting the installation via CloudFormation
3. Accelerator State Machine Initialization Using AWS CodeCommit, AWS CodeBuild, and AWS CodePipeline
4. Running the “Accelerator State Machine” using AWS Step Functions
5. Manual configurations to finalize the installation

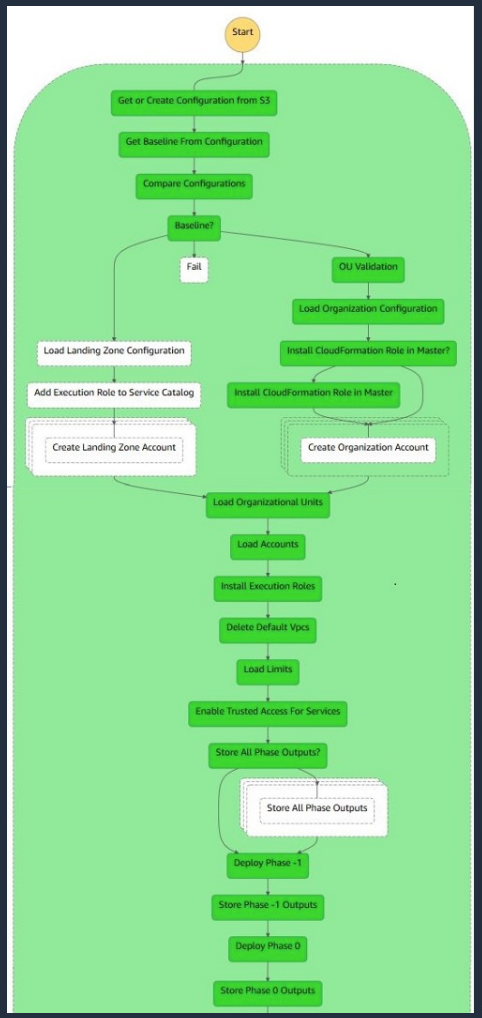
The main stages of installation

1. Create the various configuration files and drop them into an S3 Bucket
2. Starting the installation via CloudFormation
3. Accelerator State Machine Initialization Using AWS CodeCommit, AWS CodeBuild, and AWS CodePipeline



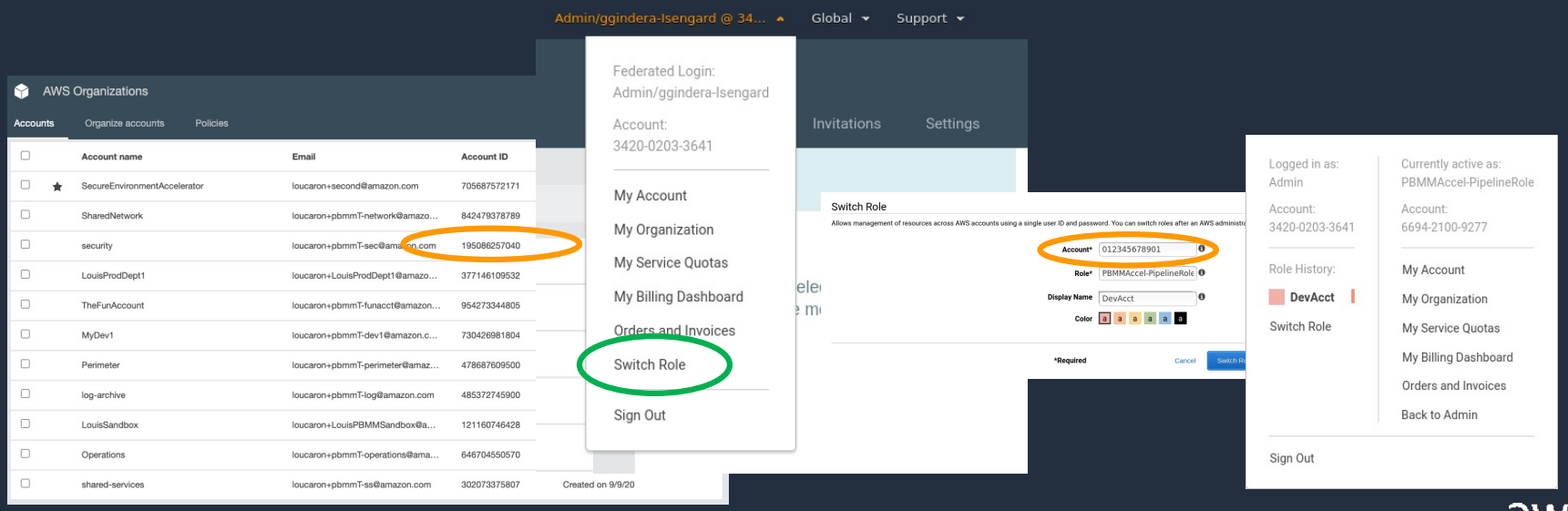
The main stages of installation

4. Running the “Accelerator State Machine” using AWS Step Functions



The main stages of installation

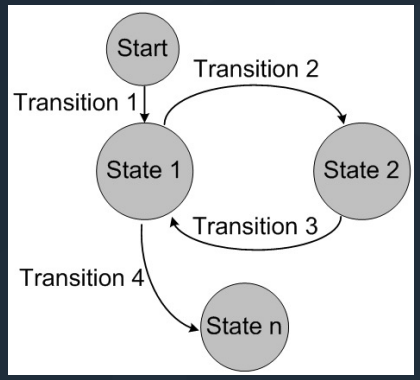
5. Manual configurations to finalize the installation
    - Change password for root and Firewall accounts
    - Initialize MFA for each root account



What is a “State Machine”

What is a “State Machine”

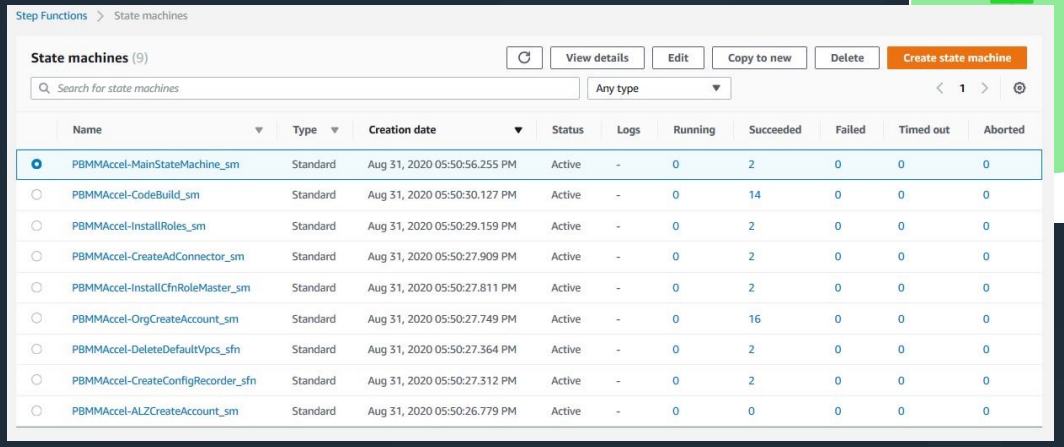
A “State Machine” is a concept used in the design of computer programs or digital logic. A finite state machine consists of a finite number of states, transitions, and actions that can be modeled with flow charts, where the logic path can be detected when conditions are met.



How the Accelerator State Machine works

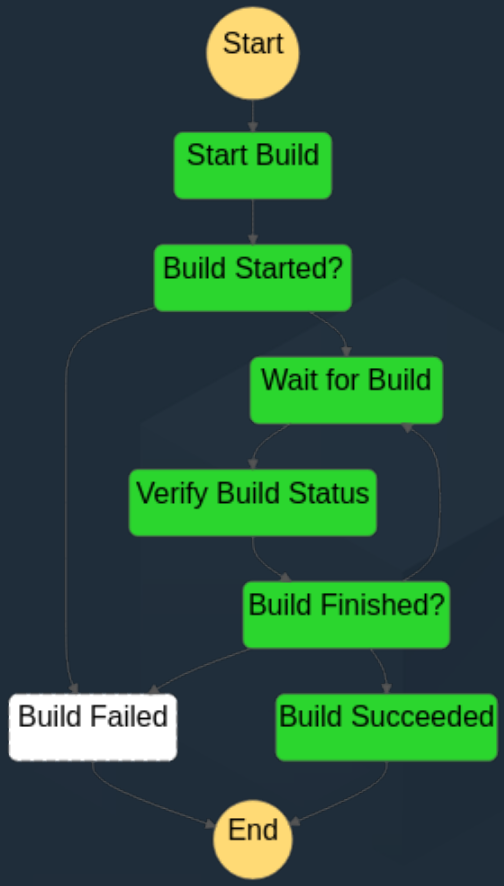
Accelerator State Machine

- The accelerator consists of a primary finite state machine PBMMAccel-MainStateMachine_SM and nine supporting secondary state machines (starting with version 1.2.1). The client only execute PBMMAccel-MainStateMachine_SM. All troubleshooting will usually start with PBMMAccel-MainStateMachine_SM.



How the Accelerator State Machine works

- Consists of 6 installation phases



How the Accelerator State Machine works

- When an error occurs, the installation stops and sends a notification to the administrator via mail
- The complete configuration of the solution is done via the file “config.json”
- The” Accelerator State Machine” generates dynamic AWS CloudFormation scripts via CDKusage
  - AWS CloudFormation is used to make installations and configurations
  - “Accelerator State Machine” code runs as a Lambda function using AWS Step Functions to orchestrate it

How the Accelerator State Machine works

- Multiple deployment phases are used due to a restriction of AWS CloudFormation that prevents us from making references between accounts and Regions and using similar names for Stack names across different Regions.
- Exchange of information between Installation Phases is done via AWS SecretManager and S3 during installation
- All shared resources use Tags to identify them

How the Accelerator State Machine works

- The status machine can be shut down and restarted at any time. The accelerator has been designed to be able to return to a stable state, so that the state machine can be stopped or fail for some reason
- The accelerator is idempotent - it can be run as many or as few times as you like without any negative effect.
- The state machine, primarily using the capabilities of the CDK, will evaluate the delta between the previously deployed old configuration and the new configuration and update the environment as appropriate.

How the Accelerator State Machine works

The state machine will run:

- automatically after each execution of the code pipeline (new installations, code upgrades, or manual pipeline runs)
- automatically when new AWS accounts are moved to an Accelerator Controller OU in AWS Organizations
- when someone starts it manually: Step Functions, PBMMAccel-MainStateMachine_SM, Start Execution, Start Execution (leave default values in name and json area)

How the Accelerator State Machine works

- The status machine prevents users from accidentally making some major changes, especially unsupported AWS platform changes, changes that will not be deployed, or changes that could be catastrophic to users.
- If someone knows exactly what they are doing and what the consequences of these changes are, we offer the opportunity to override these checks.
- Customers should expect that the items we have blocked cannot be changed after the accelerator is installed.

Demo &
Review of the file “config.json”

Full PBMM configuration file

(config.example.json)

- The complete PBMM configuration file based on feedback from customers who were migrating to AWS at a large scale and at a rapid pace. Customers of this nature indicated that they do not want to have to increase their perimeter firewalls or add interface endpoints when their developers start using new AWS services. These are the two most expensive components of the deployed architecture solution.

Light weight PBMM configuration file

(config.lite-example.json)

(Recommended for most new PBMM customers)

❖To reduce solution costs and enable customers to become more advanced AWS capabilities, we've created this lighter configuration that doesn't sacrifice functionality, but could limit performance. This configuration file:

- deploys only the required 6 centralized interface endpoints (removes 56). All services remain accessible using AWS Public Endpoints, but require traversing perimeter firewalls
- removes perimeter VPC interface endpoints
- reduces Fortigate instances size from c5n.2xl to c5n.xl (VM08 to VM04)
- removes Unclass OR and VPC

❖Accelerator allows customers to easily add or modify this feature in the future, as needed, without any impact

Ultra-Light sample configuration file
(config.ultralite-example.json)

❖This configuration file was created to represent an extremely minimalist accelerator deployment, simply to demonstrate the art of the possible for an extremely simple configuration. This configuration has:

- no shared network or perimeter accounts
- no networking objects (VPC, TGW, ELB, SG, NACL, endpoints) or route53 (zones, resolvers)
- no managed AD, AD connector, rsyslog cluster, RDGW host, or third-party firewall
- active/deploys AWS security services only in 2 regions (ca-central-1, us-east-1) (not recommended)
- deploys only 2 AWS configuration rules with SSM remediation
- rename for log archive (logs), security (audit), and operations (Ops) account names

Multi-Region sample configuration file
(config.multi-region-example.json)

❖This configuration file was created to represent a more advanced multiregional version of the Full PBMM configuration file. This configuration:

- adds a TGW in us-east-1, paired to the TGW in ca-central-1
- adds TGW static routes, including several examples of dummy static routes
- adds a central Endpoint VPC in us-east-1 with configured us-east-1 endpoints
- adds a shared VPC for all UnClass accounts OR in us-east-1, connected to the us-east-1 TGW (accessible via 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 fixes rules configured in UnClass OR
- adds a specific VPC to the local account, in us-east-1, into the MyUnclass account and connects it to the us-east-1 TGW (i.e. share the TGW)
  - Local account VPC configured to use central endpoints, associates centralized hosted zones appropriate to the VPC (also creates 5 local endpoints)
- Adds a VGW for DirectConnect to the Perimeter VPC
- adds 3rd AZ in ca-central-1 (MAD & ADC in AZ a and b)

Or find AWS SEA in GitHub?

- Main GitHub Site:
  - https://github.com/aws-samples/aws-secure-environment-accelerator
- Configuration File Template:
  - https://github.com/aws-samples/aws-secure-environment-accelerator/tree/master/reference-artifacts/SAMPLE_CONFIGS
- Description of the various configuration files:
  - https://github.com/aws-samples/aws-secure-environment-accelerator/blob/master/docs/installation/customization-index.md

Questions?

aws