AWS Landing Zone – Design Considerations (pt1)

Amazon Web Services

24 July 2018

Design Considerations

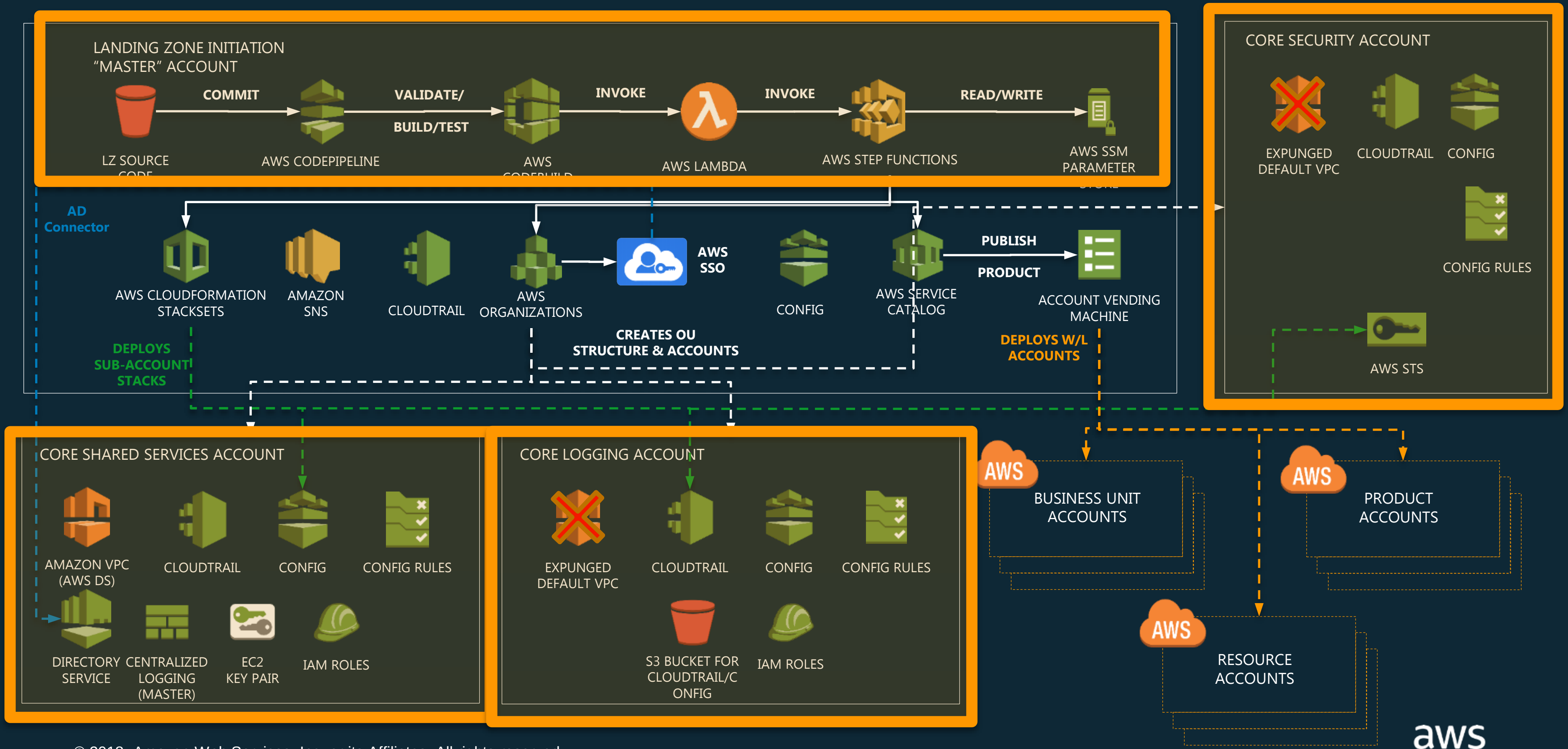
The Landing Zone Pipeline

- The baseline accounts deployed as part of the example implementation

Example Implementation - Core Accounts

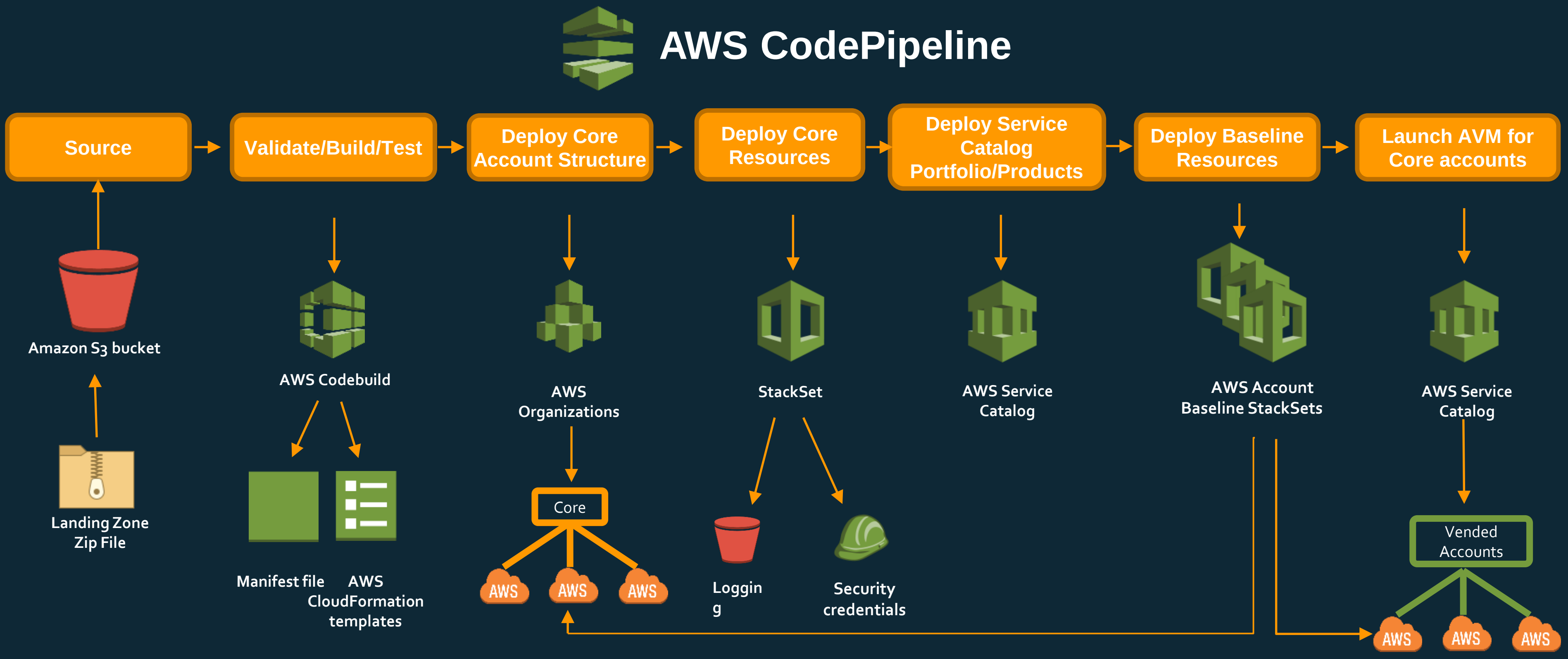
- The baseline accounts deployed as part of the example implementation

AWS Landing Zone – Pipeline and Core Accounts

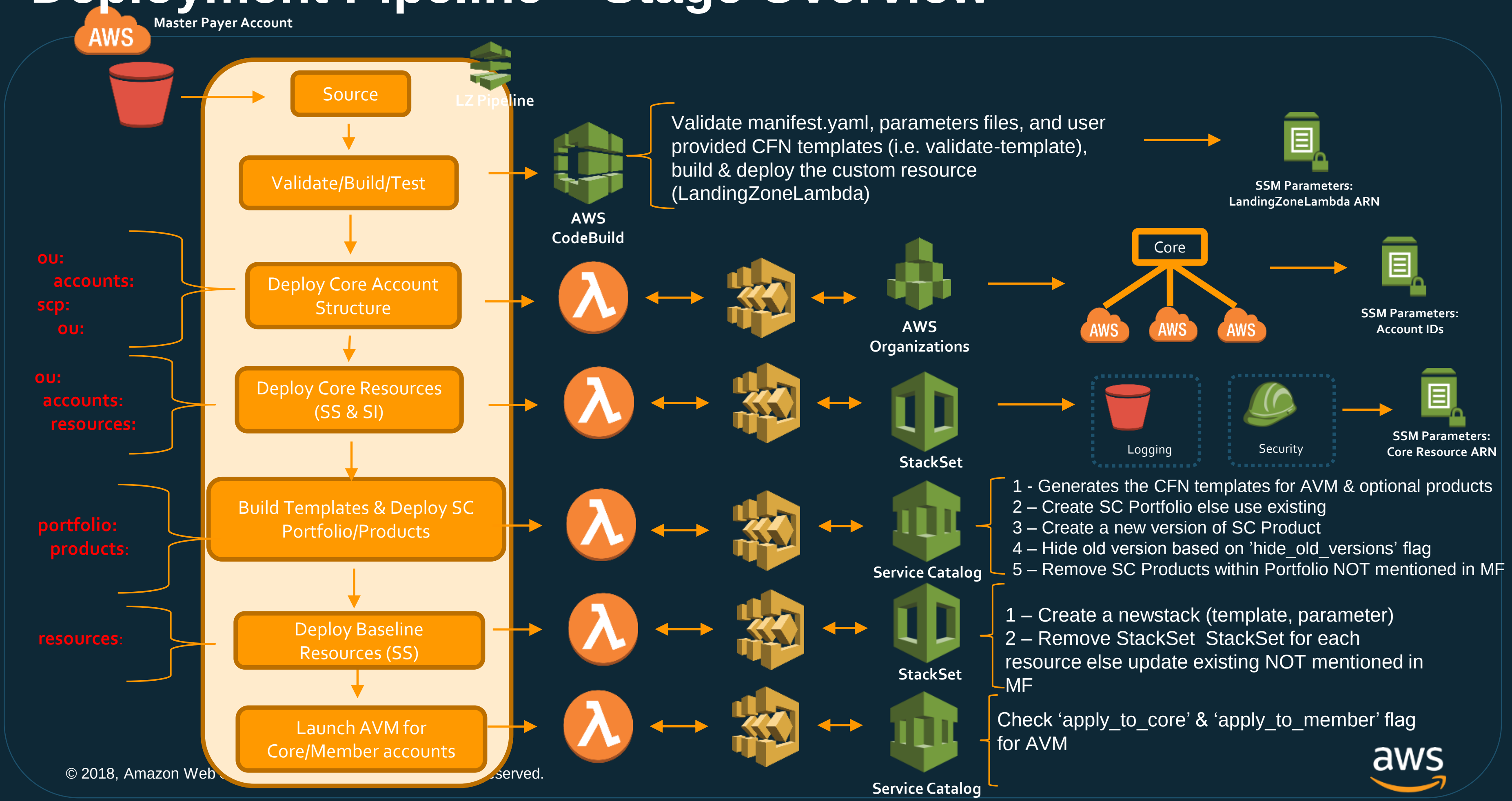


The Landing Zone Pipeline

Deployment Pipeline High Level



Deployment Pipeline – Stage Overview



Master Payer Account
AWS
LZ Pipeline
Source
Validate/Build/Test
Deploy Core Account Structure
Deploy Core Resources (SS & SI)
Build Templates & Deploy SC Portfolio/Products
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts
ou:
accounts:
scp:
ou:
ou:
accounts:
resources:
portfolio:
products:
resources:
AWS CodeBuild
SSM Parameters: LandingZoneLambda ARN
AWS Organizations
Core
AWS
AWS
AWS
SSM Parameters: Account IDs
StackSet
Logging
Security
SSM Parameters: Core Resource ARN
Service Catalog
StackSet
Service Catalog
aws

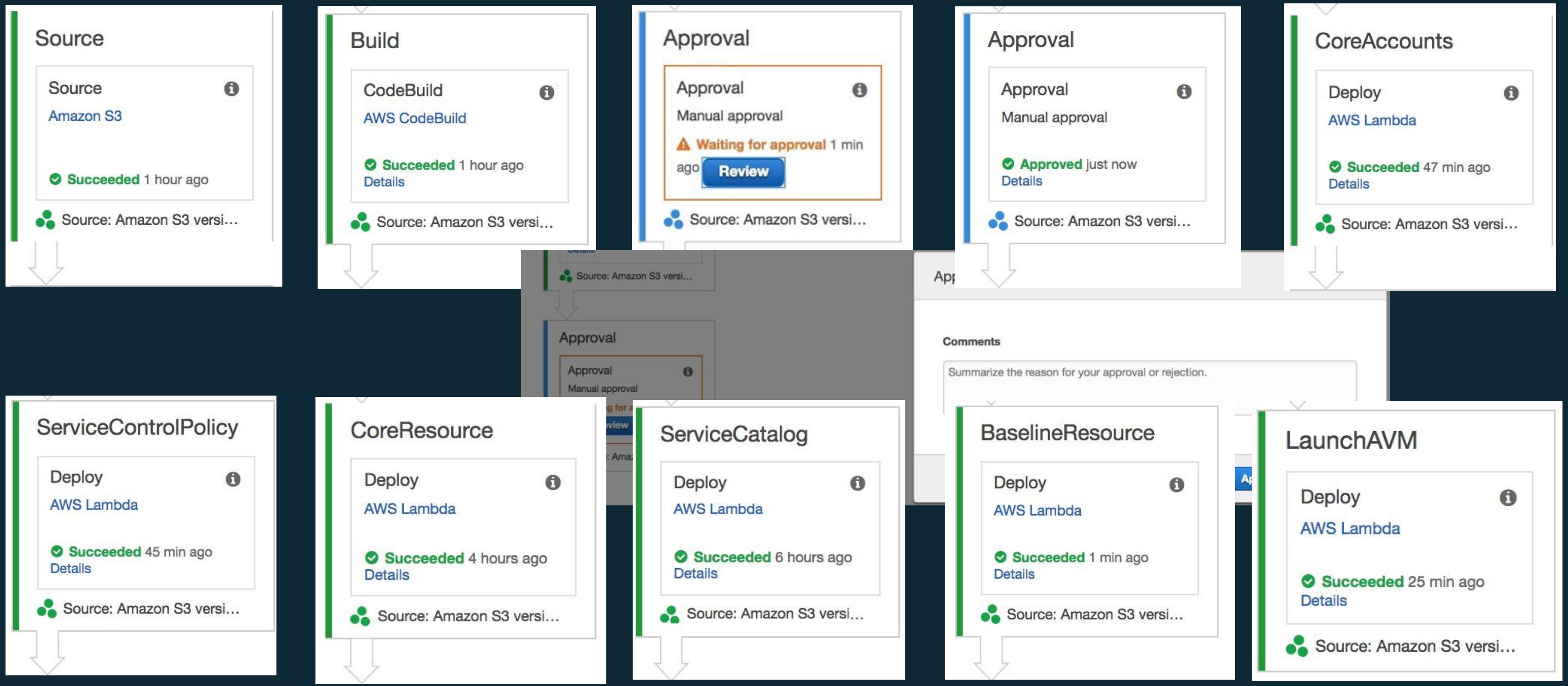
Validate manifest.yaml, parameters files, and user provided CFN templates (i.e. validate-template), build & deploy the custom resource (LandingZoneLambda)

1 - Generates the CFN templates for AVM & optional products
2 – Create SC Portfolio else use existing
3 – Create a new version of SC Product
4 – Hide old version based on ’hide_old_versions’ flag
5 – Remove SC Products within Portfolio NOT mentioned in MF

1 – Create a newstack (template, parameter)
2 – Remove StackSet StackSet for each resource else update existing NOT mentioned in MF

Check ‘apply_to_core’ & ‘apply_to_member’ flag for AVM

Implementation Pipeline - Stages



Source
Source ℹ
Amazon S3
✓ Succeeded 1 hour ago
Source: Amazon S3 versi…

Build
CodeBuild ℹ
AWS CodeBuild
✓ Succeeded 1 hour ago
Details
Source: Amazon S3 versi…

Approval
Approval ℹ
Manual approval
⚠ Waiting for approval 1 min ago Review
Source: Amazon S3 versi…

Approval
Approval ℹ
Manual approval
✓ Approved just now
Details
Source: Amazon S3 versi…

CoreAccounts
Deploy ℹ
AWS Lambda
✓ Succeeded 47 min ago
Details
Source: Amazon S3 versi…

Comments
Summarize the reason for your approval or rejection.

ServiceControlPolicy
Deploy ℹ
AWS Lambda
✓ Succeeded 45 min ago
Details
Source: Amazon S3 versi…

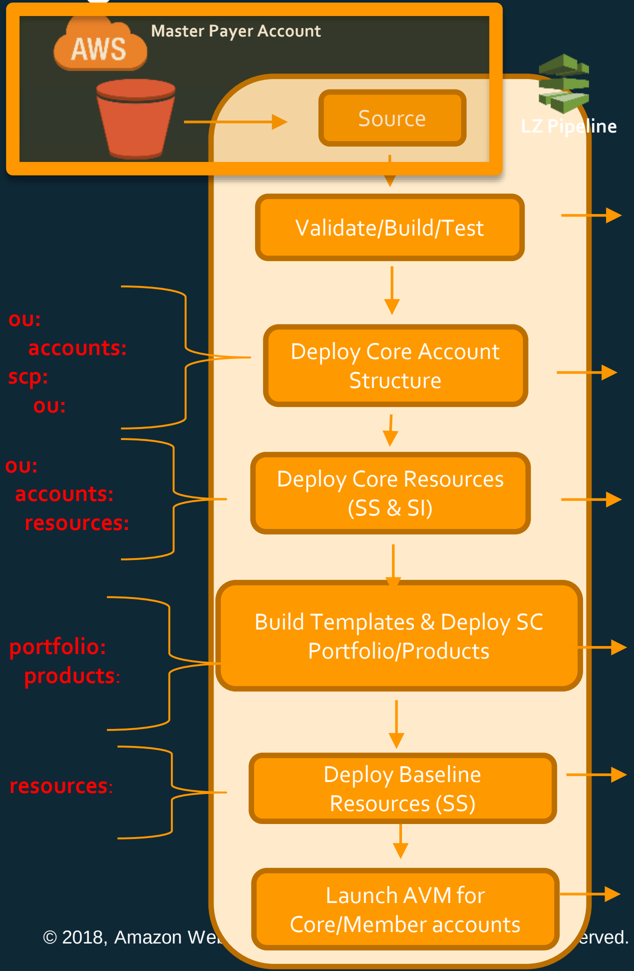
CoreResource
Deploy ℹ
AWS Lambda
✓ Succeeded 4 hours ago
Details
Source: Amazon S3 versi…

ServiceCatalog
Deploy ℹ
AWS Lambda
✓ Succeeded 6 hours ago
Details
Source: Amazon S3 versi…

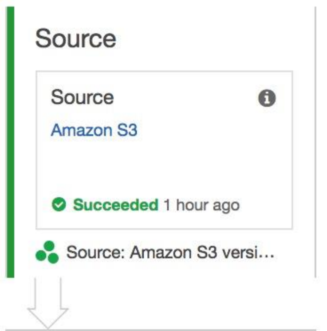
BaselineResource
Deploy ℹ
AWS Lambda
✓ Succeeded 1 min ago
Details
Source: Amazon S3 versi…

LaunchAVM
Deploy ℹ
AWS Lambda
✓ Succeeded 25 min ago
Details
Source: Amazon S3 versi…

Stage 1 - Source code Artefact S3 Bucket


AWS
Master Payer Account
Source
LZ Pipeline
Validate/Build/Test
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts

Snippet from aws-landing-zone-deployment.template


Source

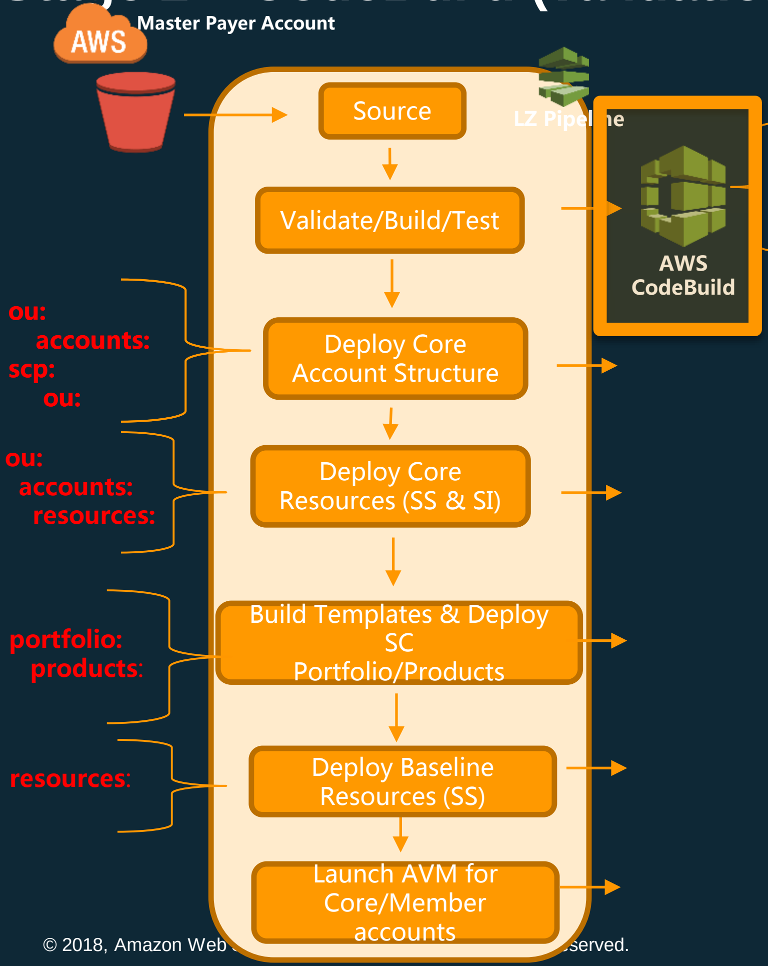
Source
Amazon S3

Succeeded 1 hour ago

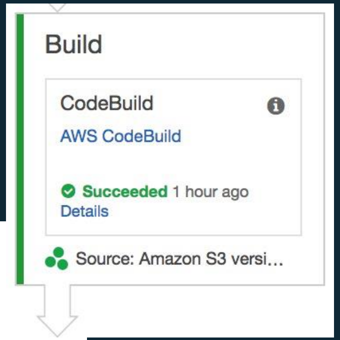
Source: Amazon S3 versi…

```
Parameters:
  LandingZonePipelineS3Bucket:
    Type: "String"
    Description: "CodePipeline S3 bucket"
    Default: "landing-zone-framework-us-east-1"
  LandingZonePipelineS3Key:
    Type: "String"
    Description: "CodePipeline S3 Key"
    Default: "lz.zip"
  LandingZonePipelineArtifactS3Bucket:
    Type: "String"
    Description: "CodePipeline Artifacts S3 bucket"
```

Stage 2 – CodeBuild (validation and packaging engine)


Master Payer Account
AWS
LZ Pipeline
Source
Validate/Build/Test
AWS CodeBuild
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts

Validate manifest.yaml, parameters files, and user provided CFN templates (i.e. validate-template), build & deploy the custom resource (LandingZoneLambda)


Build

CodeBuild

AWS CodeBuild

✓ Succeeded 1 hour ago

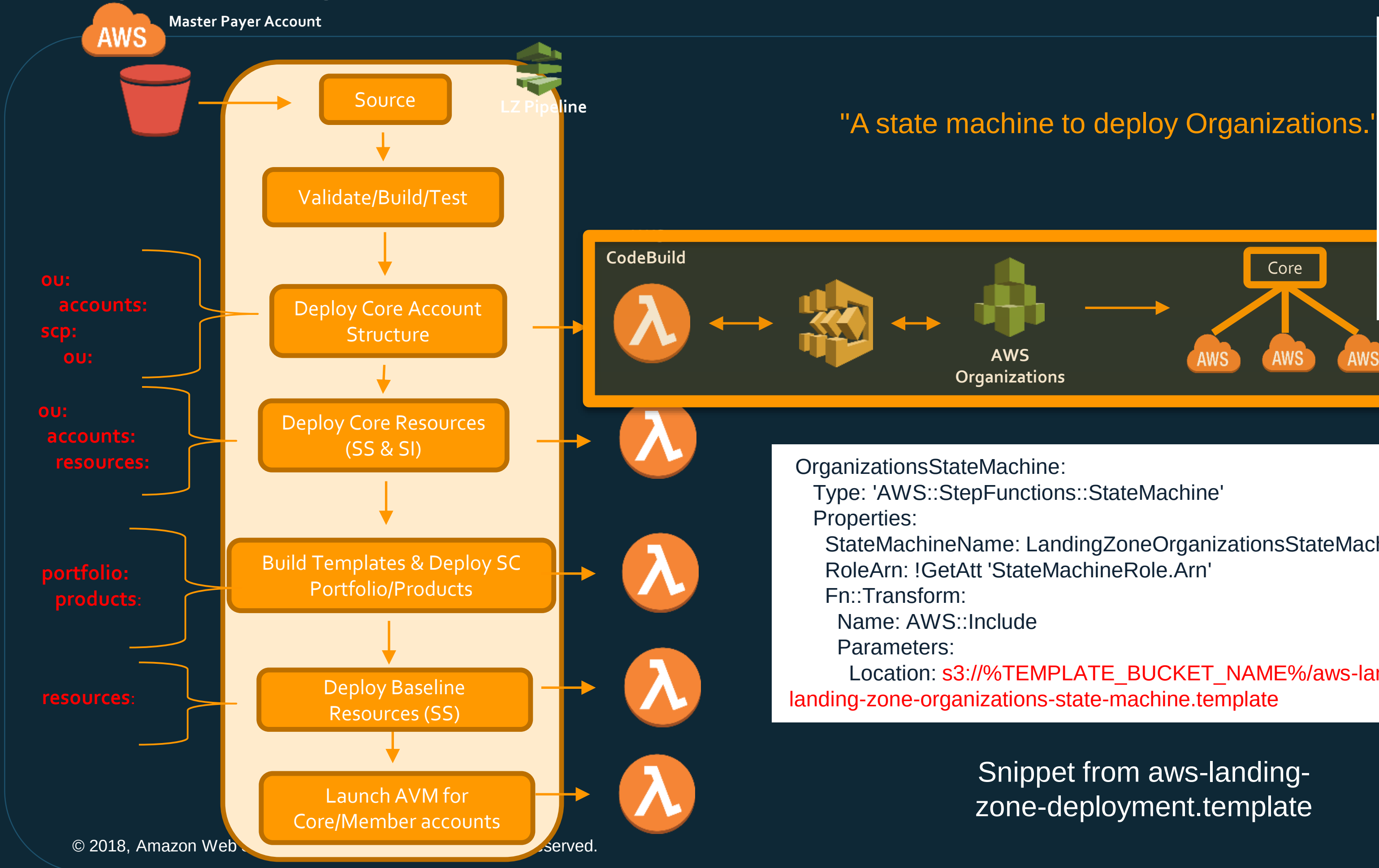
Details

Source: Amazon S3 versi...

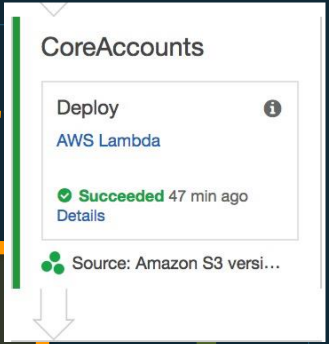
```
LandingZoneCodeBuild:
    Type: AWS::CodeBuild::Project
    Properties:
      Name: AWS-Landing-Zone-CodeBuild
      ServiceRole: !GetAtt LandingZoneCodeBuildRole.Arn
      Source:
        Type: CODEPIPELINE
      Environment:
        ComputeType: BUILD_GENERAL1_SMALL
        Image: "aws/codebuild/nodejs:6.3.1"
        Type: LINUX_CONTAINER
      Artifacts:
        Name: !Sub ${LandingZonePipelineArtifactS3Bucket}-
Built
        Type: CODEPIPELINE
```

Snippet from aws-landing-zone-deployment.template

Stage 3 – Organizations State Machine (Step Functions)


AWS
Master Payer Account
LZ Pipeline
Source
Validate/Build/Test
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts
CodeBuild
AWS Organizations
Core
AWS
AWS
AWS
SSM Parameters: Account IDs

"A state machine to deploy Organizations."



CoreAccounts

Deploy
AWS Lambda

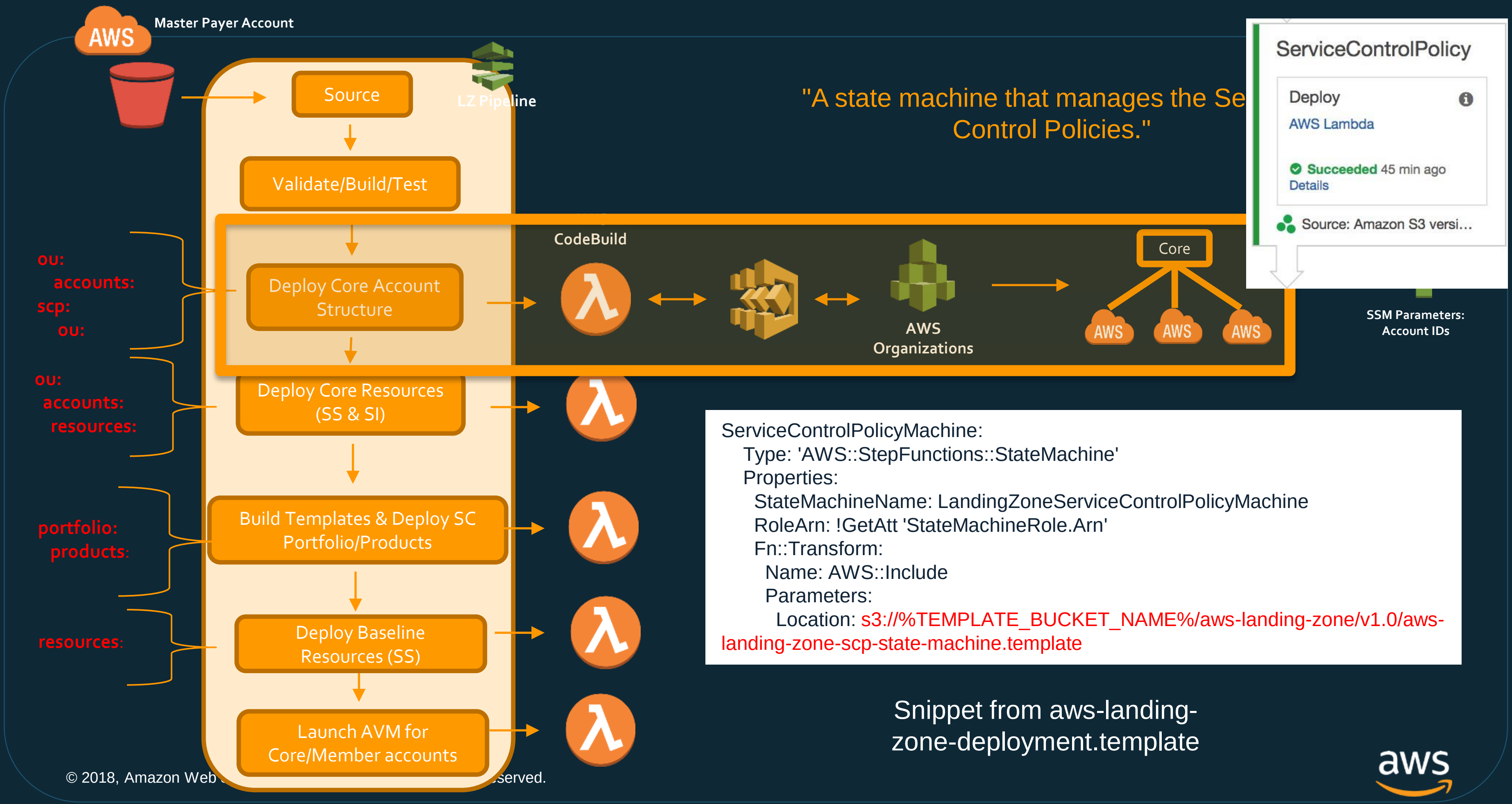
Succeeded 47 min ago
Details

Source: Amazon S3 versi...

```
OrganizationsStateMachine:
  Type: 'AWS::StepFunctions::StateMachine'
  Properties:
    StateMachineName: LandingZoneOrganizationsStateMachine
    RoleArn: !GetAtt 'StateMachineRole.Arn'
    Fn::Transform:
      Name: AWS::Include
      Parameters:
        Location: s3://%TEMPLATE_BUCKET_NAME%/aws-landing-zone/v1.0/aws-landing-zone-organizations-state-machine.template
```

Snippet from aws-landing-zone-deployment.template

Stage 3 – Service Control Policy State Machine (Step Functions)


Master Payer Account
Source
LZ Pipeline
Validate/Build/Test
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
CodeBuild
AWS Organizations
Core
AWS
AWS
AWS
SSM Parameters: Account IDs
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts

"A state machine that manages the Se Control Policies."

ServiceControlPolicy

Deploy

AWS Lambda

Succeeded 45 min ago

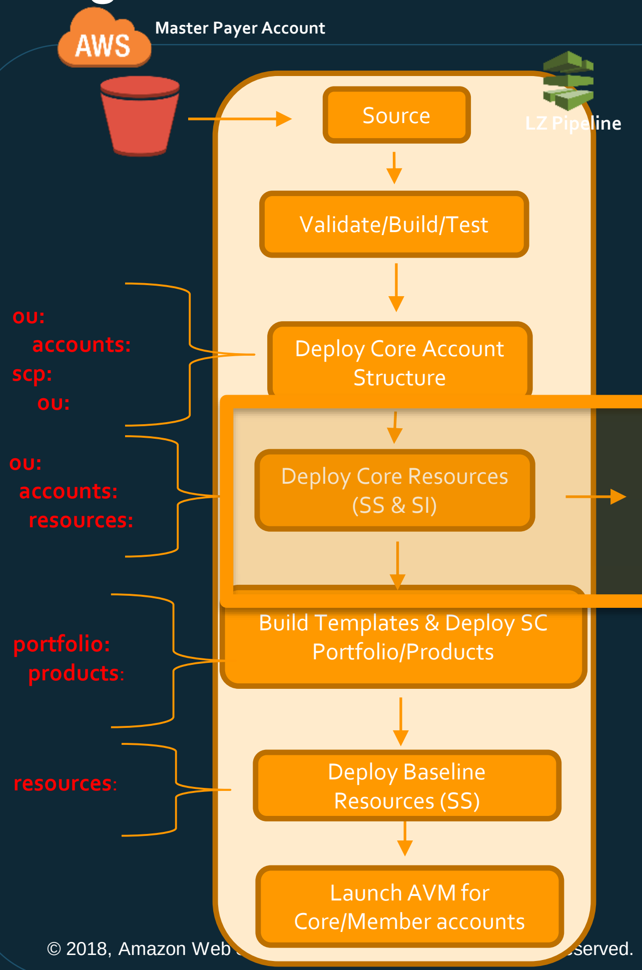
Details

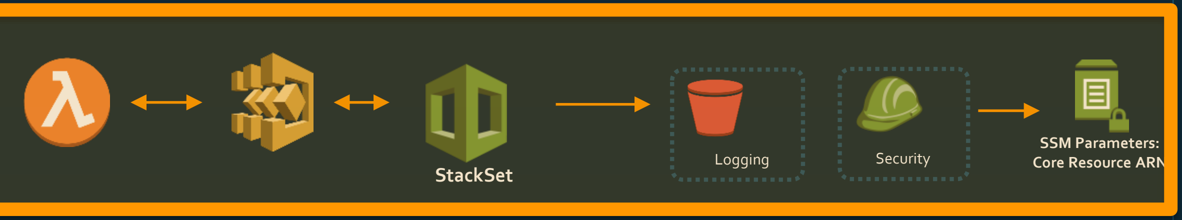
Source: Amazon S3 versi...

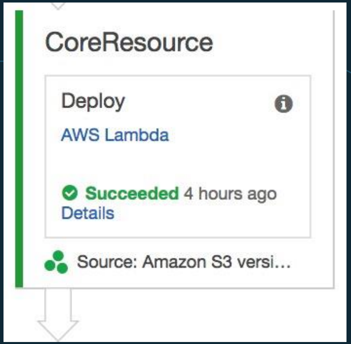
```
ServiceControlPolicyMachine:
    Type: 'AWS::StepFunctions::StateMachine'
    Properties:
      StateMachineName: LandingZoneServiceControlPolicyMachine
      RoleArn: !GetAtt 'StateMachineRole.Arn'
      Fn::Transform:
        Name: AWS::Include
        Parameters:
          Location: s3://%TEMPLATE_BUCKET_NAME%/aws-landing-zone/v1.0/aws-landing-zone-scp-state-machine.template
```

Snippet from aws-landing-zone-deployment.template

Stage 4 – Core Resources


AWS
Master Payer Account
LZ Pipeline
Source
Validate/Build/Test
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts





CoreResource

Deploy

AWS Lambda

Succeeded 4 hours ago

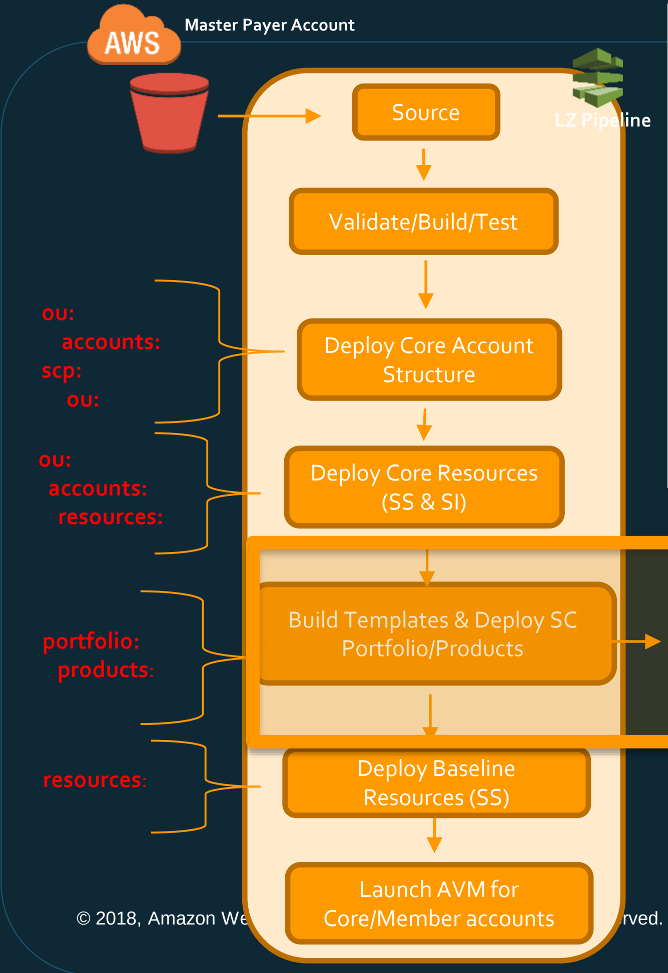
Details

Source: Amazon S3 versi...

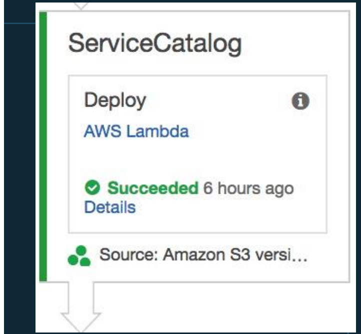
```
 if account_id is not None:
          #Deploying Core resource Stack Set
          stack_name = "AWS-Landing-Zone-{}".format(resource.name)
          sm_input =
self._create_stack_set_state_machine_input_map(stack_name,
template_full_path, params, [str(account_id)], resource.regions, ssm_map)
      else:
          #Deploying Baseline resource Stack Set
          stack_name = "AWS-Landing-Zone-Baseline-{}".format(resource.name)
          sm_input =
self._create_stack_set_state_machine_input_map(stack_name,
template_full_path, params, [], [], ssm_map)
```

Snippet from LandingZoneStateMachineTrigger Lambda

Stage 5 – Service Catalog


AWS
Master Payer Account
LZ Pipeline
Source
Validate/Build/Test
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts

```
def start_service_catalog_sm(self, sm_arn_sc):
        try:
            logger.info("Processing Service catalogs section from {}
file".format(self.manifest_file_path))
            list_sm_exec_arns = []
            for portfolio in self.manifest.portfolios:
                for product in portfolio.products:
                    sm_input =
self._create_service_catalog_state_machine_input_map(portfolio, product)
                    self._run_or_queue_state_machine(sm_input, sm_arn_sc,
list_sm_exec_arns, product.name)
            self._save_sm_exec_arn(list_sm_exec_arns)
            return
        except Exception as e:
            message = {'FILE': __file__.split('/')[-1], 'METHOD': inspect.stack()[0][3],
'EXCEPTION': str(e)}
            self.logger.exception(message)
            raise
```

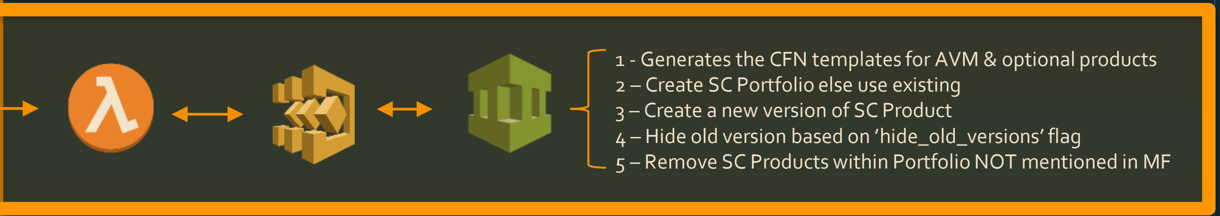

ServiceCatalog

Deploy
AWS Lambda

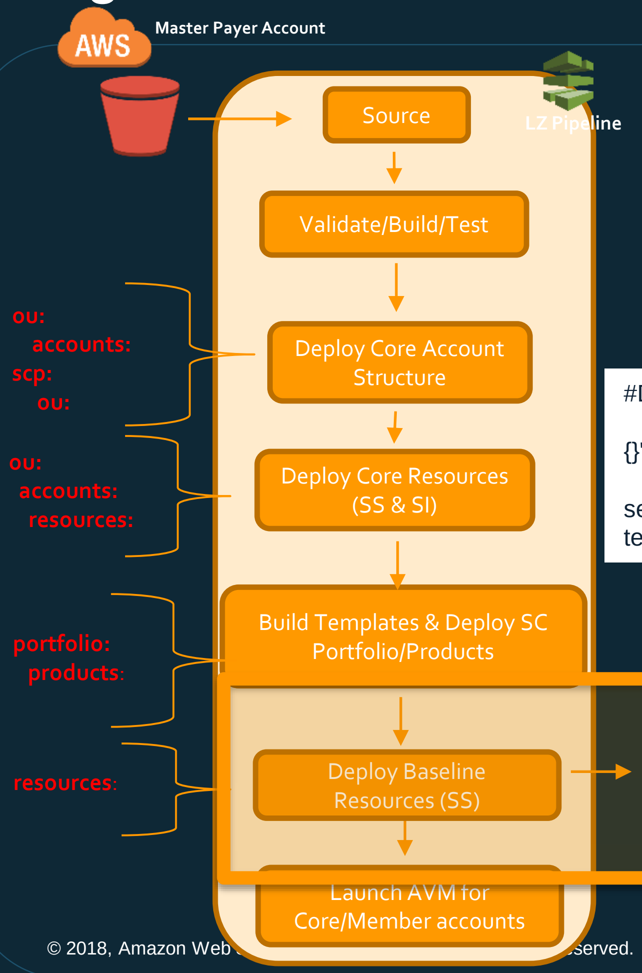
✓ Succeeded 6 hours ago
Details

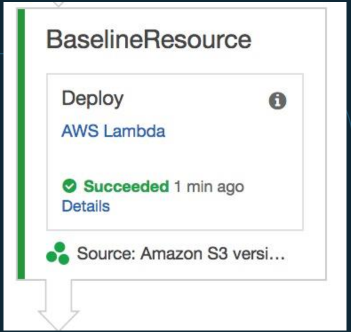
Source: Amazon S3 versi...

Snippet from LandingZoneStateMachineTrigger Lambda


1 - Generates the CFN templates for AVM & optional products
2 – Create SC Portfolio else use existing
3 – Create a new version of SC Product
4 – Hide old version based on 'hide_old_versions' flag
5 – Remove SC Products within Portfolio NOT mentioned in MF

Stage 6 – Baseline Resources


AWS
Master Payer Account
LZ Pipeline
Source
Validate/Build/Test
Deploy Core Account Structure
Deploy Core Resources (SS & SI)
Build Templates & Deploy SC Portfolio/Products
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts
ou:
accounts:
scp:
ou:
ou:
accounts:
resources:
portfolio:
products:
resources:



BaselineResource

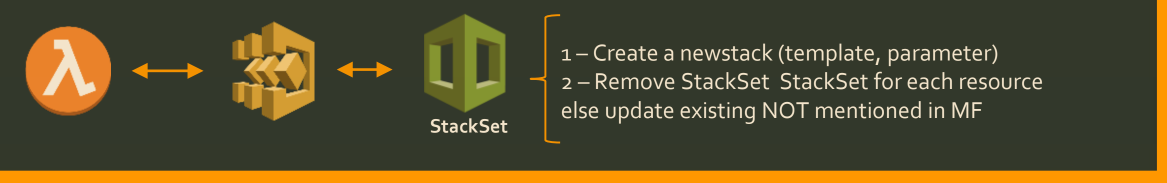
Deploy
AWS Lambda

Succeeded 1 min ago
Details

Source: Amazon S3 versi...

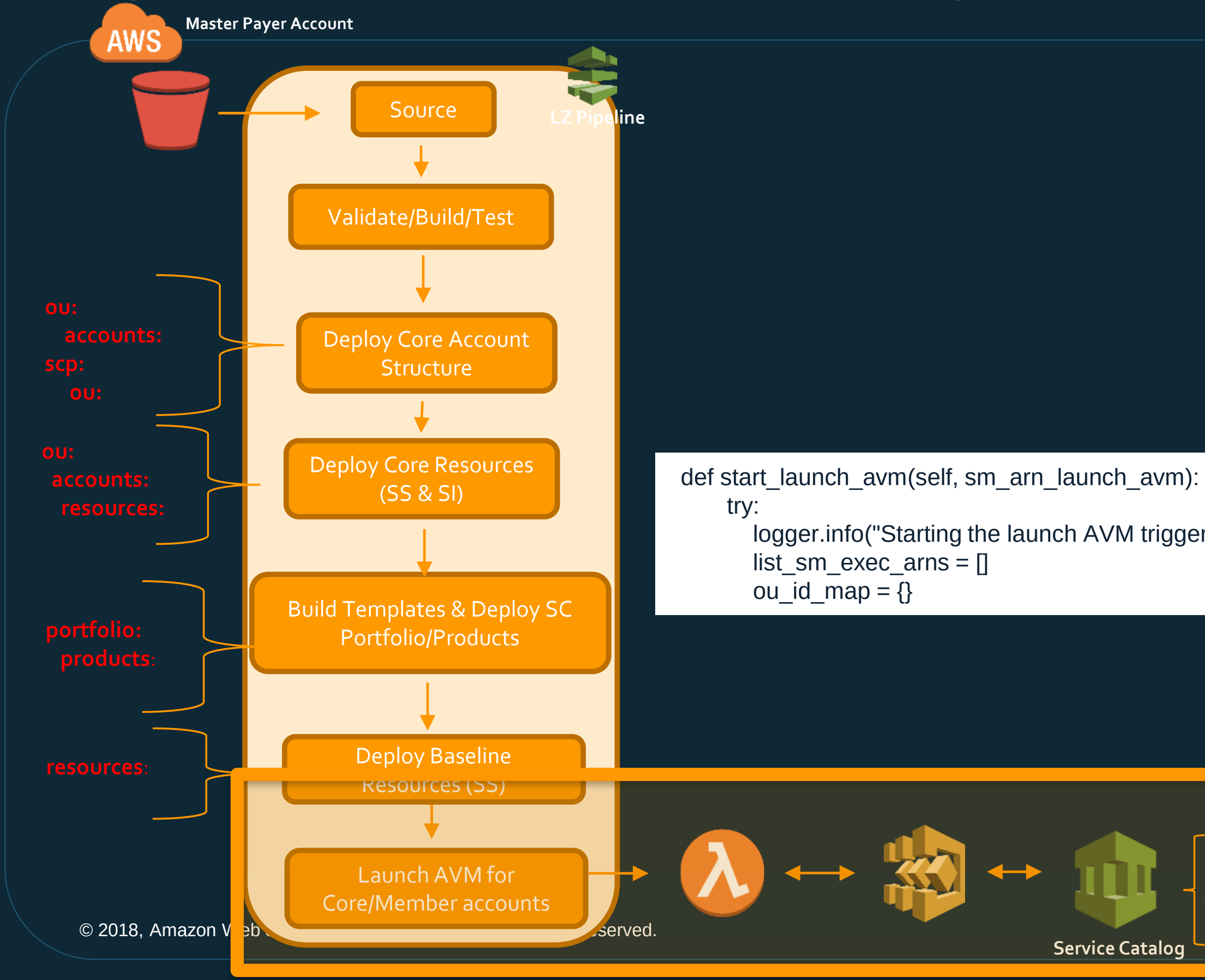
Snippet from
LandingZoneState
MachineTrigger
Lambda

```
#Deploying Baseline resource Stack Set
            stack_name = "AWS-Landing-Zone-Baseline-
{}".format(resource.name)
            sm_input =
self._create_stack_set_state_machine_input_map(stack_name,
template_full_path, params, [], [], ssm_map)
```



1 – Create a newstack (template, parameter)
2 – Remove StackSet StackSet for each resource else update existing NOT mentioned in MF

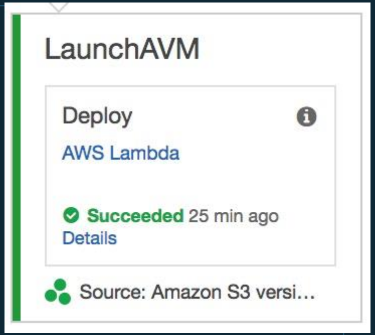
Stage 7 – Launch Automated Vending Machine (AVM)


AWS
Master Payer Account
LZ Pipeline
Source
Validate/Build/Test
ou:
accounts:
scp:
ou:
Deploy Core Account Structure
ou:
accounts:
resources:
Deploy Core Resources (SS & SI)
portfolio:
products:
Build Templates & Deploy SC Portfolio/Products
resources:
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts
Service Catalog

Check 'apply_to_core' & 'apply_to_member' flag for AVM

```
def start_launch_avm(self, sm_arn_launch_avm):
    try:
        logger.info("Starting the launch AVM trigger")
        list_sm_exec_arns = []
        ou_id_map = {}
```

Snippet from LandingZoneState MachineTrigger Lambda



LaunchAVM

Deploy

AWS Lambda

Succeeded 25 min ago

Details

Source: Amazon S3 versi...

The Core Account Structure

Core Accounts

The following Accounts are created automatically as part of the **Example Multi-Account Implementation:**

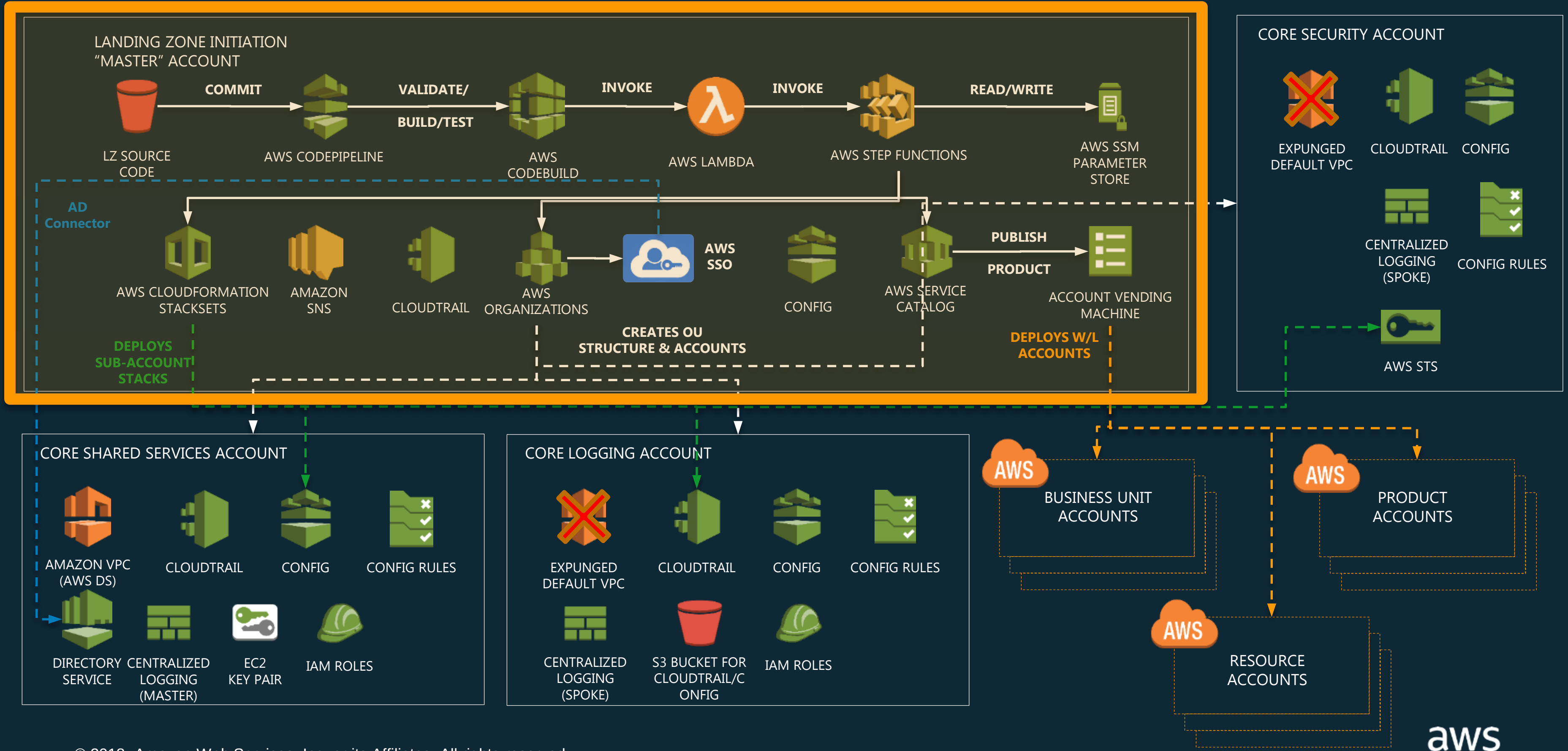
- AWS Organizations Master Account
- Shared Services Account
- Security Account
- Logging Account

Core Accounts

The following Accounts are created automatically as part of the **Example Multi-Account Implementation:**

- AWS Organizations Master Account
- Shared Services Account
- Security Account
- Logging Account

Organizations Master Account

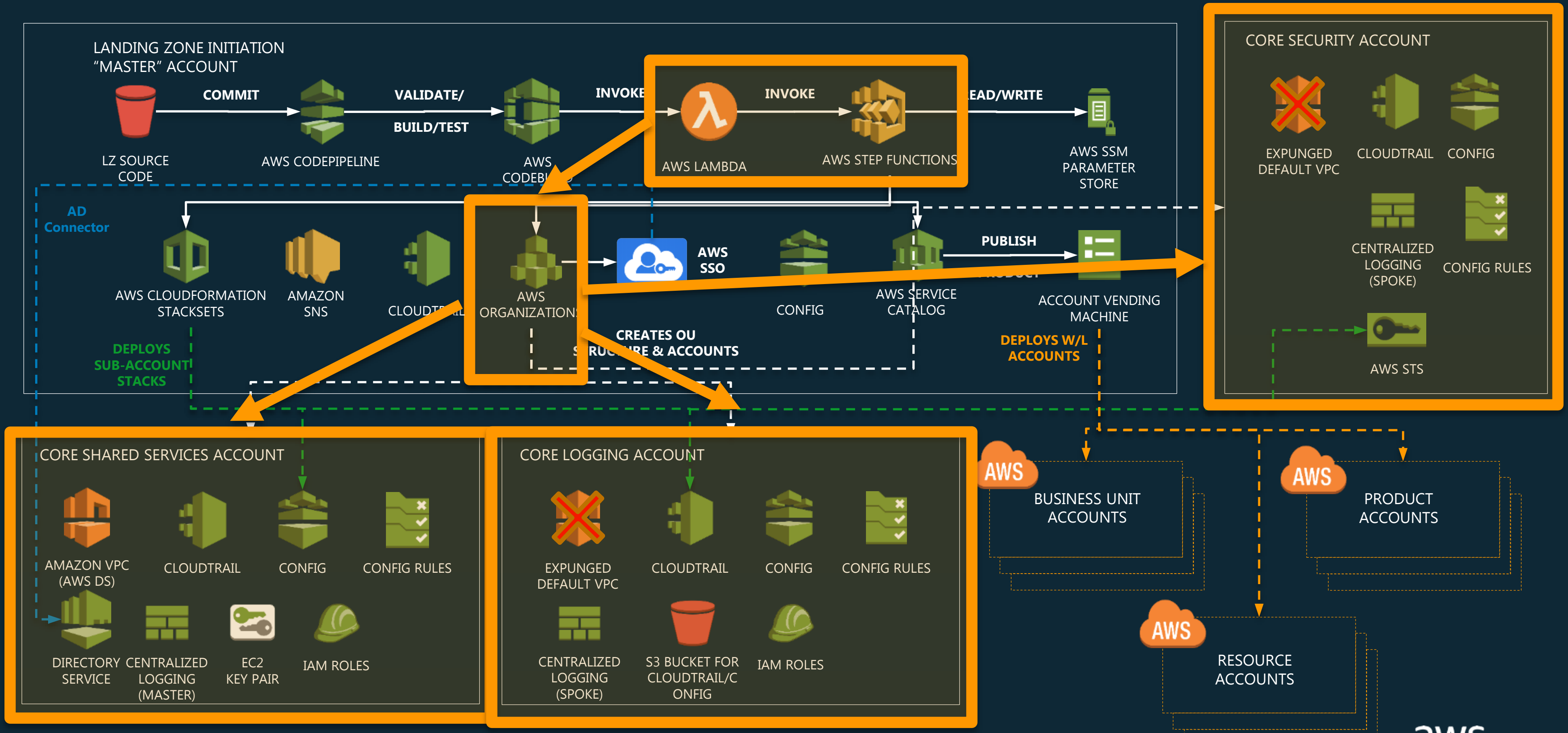


Organizations Master Account - Functionality

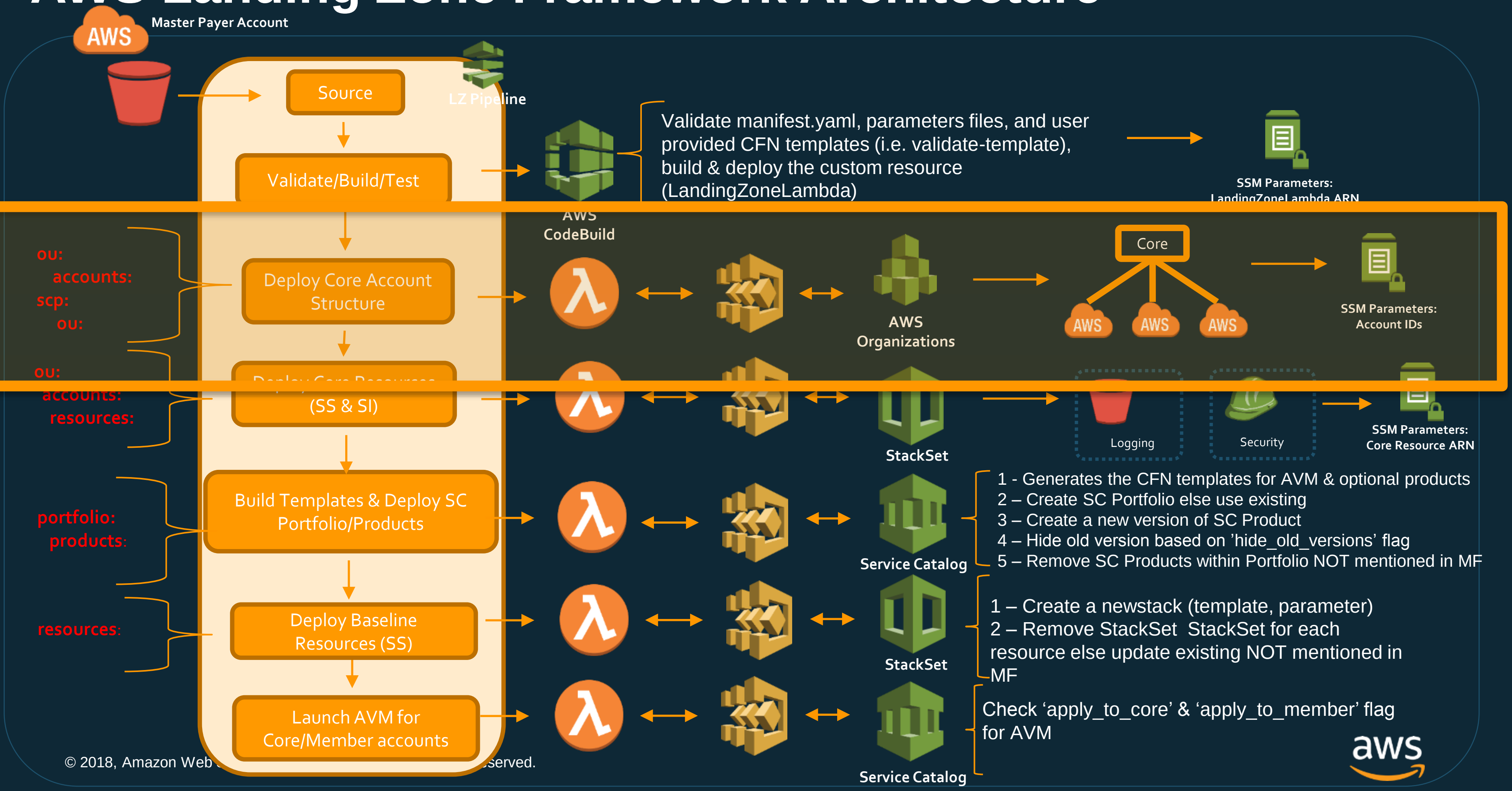
The 1st and only manually created account in the process containing:

- **Deployment Pipeline** – the implementation and configuration pipeline
- **Account Management** – creates / manages new AWS accounts with AWS Organizations
- **Account Vending Machine** – hosts a self-service Account creation product for end user account creation.
- **Service Catalog** – this is the engine presenting the Account Vending Machine and Future products (Custom-built, Marketplace)
  - The account creation could be integrated with Ticketing Systems such as ServiceNow for example.

Organizations and Core OU within Master Account



AWS Landing Zone Framework Architecture


AWS
Master Payer Account
LZ Pipeline
Source
Validate/Build/Test
AWS CodeBuild
Deploy Core Account Structure
Deploy Core Resources (SS & SI)
Build Templates & Deploy SC Portfolio/Products
Deploy Baseline Resources (SS)
Launch AVM for Core/Member accounts
ou:
accounts:
scp:
ou:
ou:
accounts:
resources:
portfolio:
products:
resources:
AWS Organizations
Core
AWS
AWS
AWS
SSM Parameters: LandingZoneLambda ARN
SSM Parameters: Account IDs
StackSet
Logging
Security
SSM Parameters: Core Resource ARN
Service Catalog
StackSet
Service Catalog
aws

Validate manifest.yaml, parameters files, and user provided CFN templates (i.e. validate-template), build & deploy the custom resource (LandingZoneLambda)

1 - Generates the CFN templates for AVM & optional products
2 – Create SC Portfolio else use existing
3 – Create a new version of SC Product
4 – Hide old version based on 'hide_old_versions' flag
5 – Remove SC Products within Portfolio NOT mentioned in MF

1 – Create a newstack (template, parameter)
2 – Remove StackSet StackSet for each resource else update existing NOT mentioned in MF

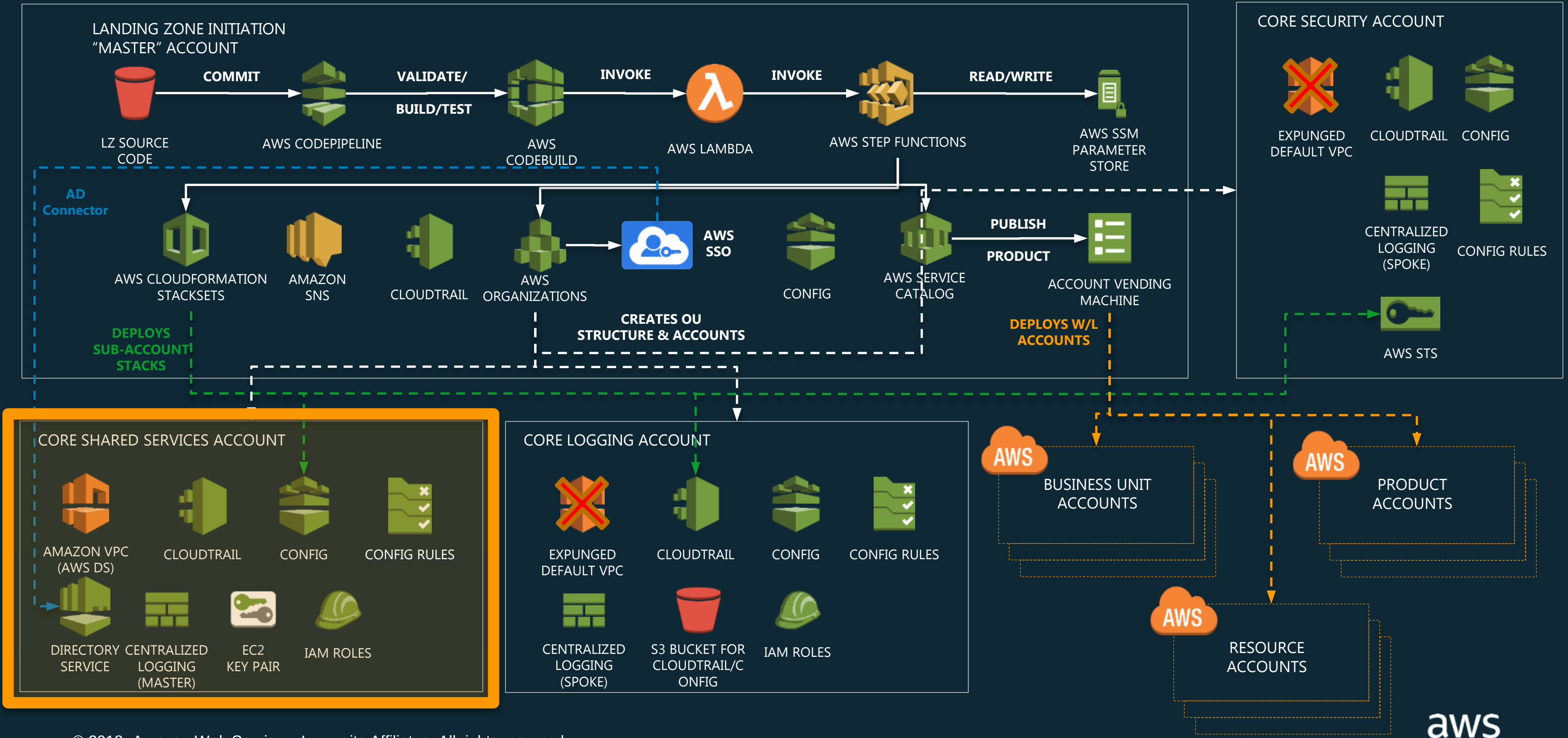
Check 'apply_to_core' & 'apply_to_member' flag for AVM

Core Accounts

The following Accounts are created automatically as part of the **Example Multi-Account Implementation:**

- AWS Organizations Master Account
- Shared Services Account
- Security Account
- Logging Account

Shared Services Account



Shared Services Account

A core shared services account is created for hosting landing zone infrastructure dependencies including

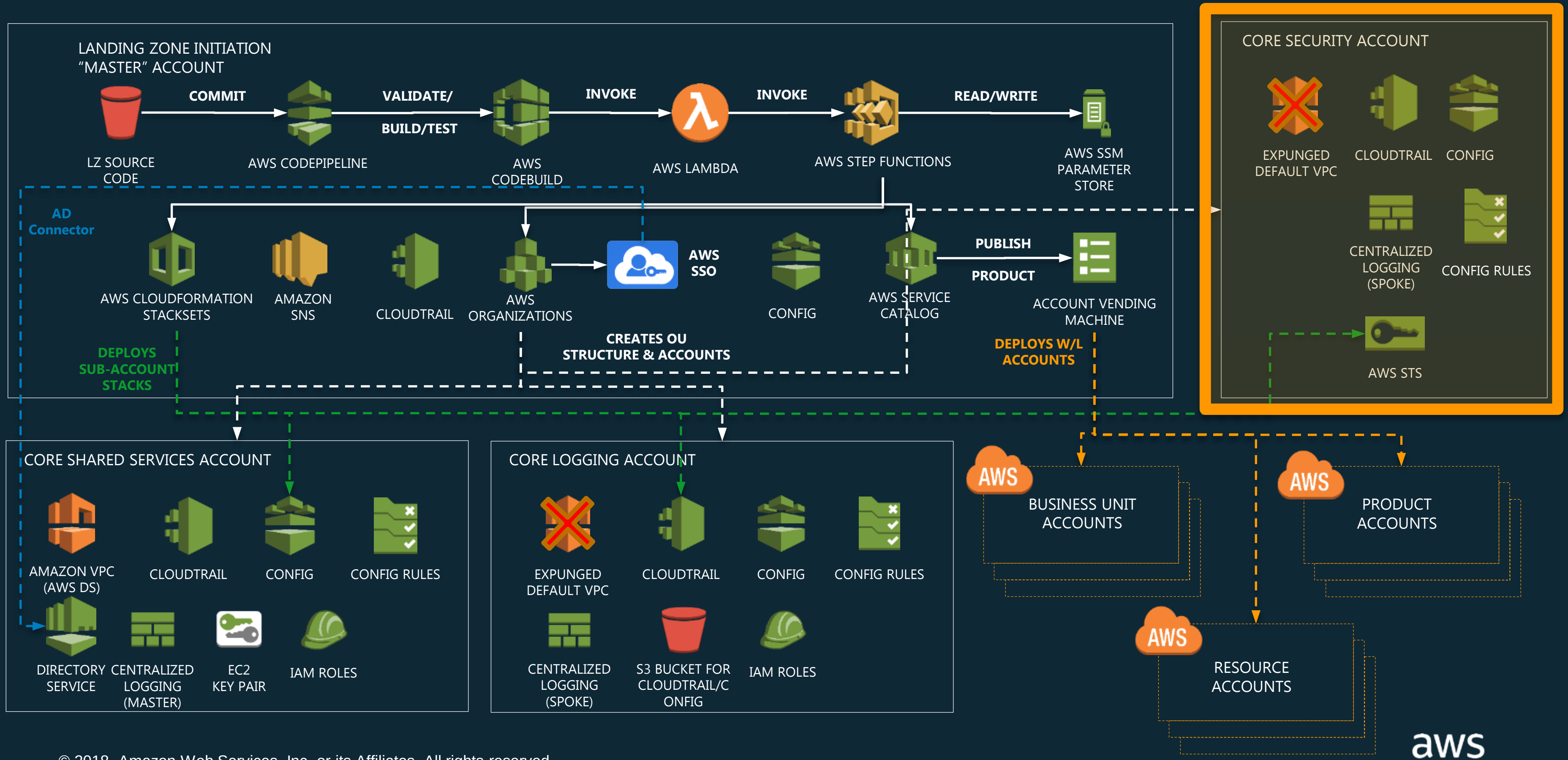
- An LDAP Directory Service for managing user access for Single Sign-On
- An initial location for log analytics capabilities
  - Elasticsearch is an option as part of the Deployment
  - Could just as easily be Splunk instead or even build in another account

Core Accounts

The following Accounts are created automatically as part of the **Example Multi-Account Implementation:**

- AWS Organizations Master Account
- Shared Services Account
- Security Account
- Logging Account

Security Account



Security Account

A core security account is created to facilitate:

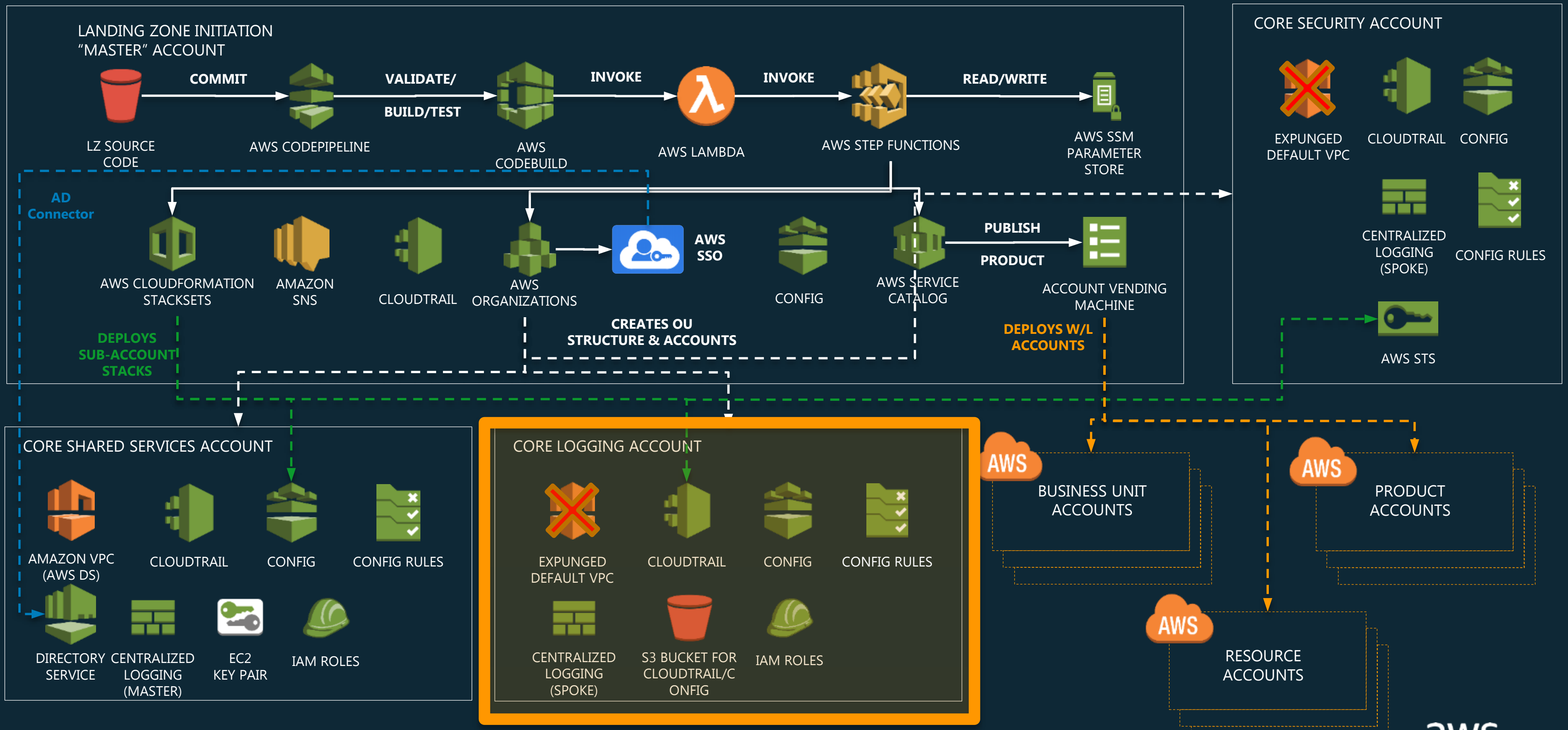
- Customer’s security department-controlled audit and security remediation access to all accounts.
- Access to this account should be highly controlled
- Cross-account read-only and administrative access will be pre-provisioned to all new accounts as part of the security baseline
- This provides ‘break-glass’ access to the security team

Core Accounts

The following Accounts are created automatically as part of the **Example Multi-Account Implementation:**

- AWS Organizations Master Account
- Shared Services Account
- Security Account
- Logging Account

Logging Account



Logging Account

A dedicated account for securely storing logs for archiving and forensic activities. This Account Contains

- Centralized location for copies of every account’s Audit and Configuration compliance logs
- Additional buckets in the Logging Account could and should be used to centralize and store other audit and compliance logs

Lab 2 – Review Deployment

What’s next?

- ✓ Module 1 / Lab 1 – Initial Thought Process / Deploy Example Landing Zone
- ✓ Module 2 / Lab 2 – Design Considerations (pt1) / Review Deployment
- ○ Module 3 - Design Considerations (pt2)
- ○ Module 4 - Design Considerations (pt3)
  - Lab 3 - Configure AD and SSO
  - Lab 4 - Deploy a Member Account
  - Lab 5 - Deploy Centralised Logging Hub
- ○ Module 5 – Extending the Landing Zone
- ○ Lab 6 – Configure Centralised Logging Spoke and new Config Rule