[Publish] Document - LZA Environment Updates

Este documento proporciona una **descripción general completa** de la configuración del Landing Zone Accelerator desplegado en su entorno de AWS. Describe cómo se gestiona el entorno, detalla cada archivo de configuración y proporciona un flujo de trabajo estándar para realizar cambios futuros utilizando su *pipeline* basado en Git. Esta documentación está destinada a ser compartida con sus equipos de interés internos para mejorar la transparencia, la mantenibilidad y la capacidad de gestión a largo plazo del despliegue de LZA.

---

1. Descripción General del LZA

El **LZA** ofrece un marco de trabajo automatizado, escalable y seguro para gestionar un entorno de AWS con múltiples cuentas. Se basa en AWS Control Tower, extendiéndolo con servicios de seguridad adicionales, configuraciones de IAM, estandarización de redes y automatización del cumplimiento normativo.

Beneficios Clave

- Despliegue de infraestructura estandarizado y repetible.
- Configuración de cuentas segura y conforme a las mejores prácticas de AWS.
- Gestión de la configuración basada en GitOps utilizando Github.
- Despliegues automatizados mediante AWS CodePipeline y AWS CodeBuild.

Arquitectura del Despliegue

- **Control de Código Fuente**: Los archivos de configuración se almacenan en un repositorio Git de Github (`https://github.com/gs-esteban/landing_zone`).
- **AWS CodeConnection**: Conecta Github de forma segura a AWS CodePipeline.
- **Flujo de Trabajo de CodePipeline**:
  - **Etapa de Origen (Source)**: Obtiene la última configuración del repositorio de Github.
  - **Etapa de Construcción (Build)**: Valida la estructura YAML y la lógica de configuración usando AWS CodeBuild.

- **Etapa de Despliegue (Deploy)**: Aplica los cambios de configuración al entorno de AWS.

2. Flujo de Trabajo para Cambios de Configuración

Los cambios en la configuración siguen un enfoque **GitOps**. Esto asegura que todos los cambios estén versionados, revisados por pares y desplegados automáticamente.

Flujo de Trabajo Paso a Paso

1. **Clonar el Repositorio**

```
git clone https://Github.org/<tu-org>/<repo-lza>.git
cd <repo-lza>

```

2. **Crear una Rama de Característica (*Feature Branch*)**

```
git checkout -b feature/actualizar-config

```

3. Realizar Cambios

   Modifique uno o más de los archivos de configuración YAML, como:
   - `global-config.yaml`
   - `accounts-config.yaml`
   - `security-config.yaml`
   - `networking-config.yaml`
   - `organization-config.yaml`
   - `iam-config.yaml`
4. **Confirmar y Enviar Cambios (*Commit and Push*)**

```
git add .
git commit -m "Ejemplo: Actualizar config global con nuevos límites de presupuesto"
git push origin feature/actualizar-config

```

5. **Enviar un *Pull Request***
   - Abra un PR en Github apuntando a la rama `main`.
   - Solicite revisiones de código al equipo de seguridad/infraestructura designado.
6. **Despliegue**

- Al fusionar (*merge*) a la rama `main`.
- Vaya a la consola de AWS CodePipeline, seleccione el *pipeline* `AWSAccelerator-Pipeline` y haga clic en **Liberar cambio (Release change)**.
- Etapas del *pipeline*: Origen → Construcción → Aprobación → Despliegue.

7. **Monitorear y Validar**
   - Revise el estado en la consola de CodePipeline.
   - Use los registros de CloudWatch para solucionar problemas si es necesario.
   - Confirme que los cambios de configuración se hayan aplicado correctamente en la Consola de AWS.

3. Archivos de Configuración

global-config.yaml

Define la estructura fundamental para el entorno LZA.

Características:

- **Región de Origen (*Home Region*)** y Regiones Habilitadas.
- **Retención de Registros de CloudWatch**.
- Habilitación y Versión de **Control Tower**.
- **Barreras de Seguridad (*Guardrails*)** (p. ej., acceso root, uso obligatorio de MFA, bloqueo de acceso público a S3).
- Configuración de **Registros (*Logging*)** (*buckets*, reglas de ciclo de vida).
- **Alertas de Presupuesto** e Informes de Costo y Uso.

accounts-config.yaml

Define todas las cuentas de AWS en la organización.

Categorías de Cuentas:

- **Cuentas Obligatorias**:
  - **Management**: Cuenta raíz para gestión y aprovisionamiento.
  - **LogArchive**: Almacena registros centralizados.

- **Audit**: Cuenta de solo lectura y administrador de seguridad delegado.

- **Cuentas de Carga de Trabajo (*Workload*)**:
  - **Development**: Entorno para experimentación y pruebas no productivas.

Las cuentas están vinculadas a **Unidades Organizativas (OUs)** definidas en `organization-config.yaml`.

iam-config.yaml

Especifica los roles, políticas y perfiles de instancia de IAM que se desplegarán en todas las cuentas.

Definiciones Clave:

- **Conjuntos de Políticas (*Policy Sets*)**: Políticas reutilizables gestionadas por el cliente.
- **Conjuntos de Roles (*Role Sets*)**:
  - `EC2-Instance-Profile-SSM-Role`: Para instancias EC2 gestionadas.
  - `VPC-Flowlog-Role`: Usado para la entrega de registros de flujo.
  - Backup-Role: Permisos para los servicios de AWS Backup.

    Los roles de IAM se asignan con políticas de confianza y políticas gestionadas por AWS o personalizadas.

networking-config.yaml

Centraliza las definiciones de red para toda la Organización de AWS.

Capacidades:

- **Eliminación de VPC Predeterminada**: Desactivada por defecto para todas las cuentas.
- **Registros de Flujo de VPC Globales (*VPC Flow Logs*)**:
  - Tipo de tráfico: `ALL`.
  - Entrega a CloudWatch Logs (retención de 30 días).
  - Campos completos para auditoría y cumplimiento normativo.

    Actualmente no hay VPCs, Transit Gateways o Políticas de Endpoint definidas, pero la estructura está lista para implementarlas.

organization-config.yaml

Define la jerarquía de la Organización de AWS.

Unidades Organizativas (OUs):

- *Security*, *Infrastructure*, *Workloads*, con OUs anidadas para *Development*, *Staging* y *Production*.
- Las OUs sirven como objetivos para la configuración, SCPs y reglas de cumplimiento.

Políticas:

- **Políticas de Control de Servicios (SCPs)**: Ninguna desplegada actualmente, pero la estructura es compatible.
- **Políticas de Etiquetado y Respaldo**: La estructura está presente para uso futuro.

security-config.yaml

Proporciona una configuración centralizada para herramientas de seguridad, políticas de IAM, AWS Config y alarmas.

Capacidades Principales:

- Delega la cuenta de **Auditoría** como administrador de GuardDuty, Config y Security Hub.
- Aplica una **política de contraseñas de IAM** para toda la organización (14 caracteres, complejidad, rotación).
- Habilita **Access Analyzer** a nivel de organización.
- Estándares de **Security Hub** definidos pero no habilitados:
  - Mejores Prácticas de Seguridad Fundamentales de AWS (FSBP).
  - *Benchmark* de CIS AWS Foundations.
  - NIST 800-53 Rev. 5.
- **Reglas de AWS Config**:
  - Monitorea métricas críticas de cumplimiento como:
    - Uso de grupos de IAM.
    - Uso de *Internet Gateways*.
    - Cifrado de *buckets* S3.

        - Uso de KMS en Secrets Manager.
        - Disponibilidad de ELB y uso de certificados.
- **Alarmas de CloudWatch**:
    - Basadas en los controles del *benchmark* de CIS:
        - Uso de la cuenta raíz.
        - Cambios en políticas de IAM.
        - Cambios en CloudTrail.
        - Fallos de autenticación en la consola.
        - Eliminación de llaves KMS.
        - Cambios en S3 y grupos de seguridad.

4. Mejores Prácticas para Gestionar LZA

- Utilice **ramas y *pull requests*** para todos los cambios de configuración.
- Almacene políticas y documentos personalizados en directorios con una estructura clara.
- Monitoree el estado del *pipeline* usando AWS CodePipeline y CloudWatch.
- Habilite servicios de seguridad adicionales como Security Hub, GuardDuty y Macie según sea necesario.
- Revise regularmente los **marcos de trabajo de CIS/NIST** para alinear los controles.
- Mantenga una **separación clara** entre las cuentas de desarrollo, pruebas y producción usando OUs.