[Publish] - LZA Environment

Este documento proporciona una descripción completa de la configuración del **AWS Landing Zone Accelerator (LZA)** en tu entorno de AWS. Explica cómo se gestiona el entorno, detalla cada archivo de configuración y proporciona un flujo de trabajo estándar para realizar cambios futuros utilizando Git. Esta documentación está diseñada para compartirse con los equipos internos a fin de mejorar la transparencia, la mantenibilidad y la gestión a largo plazo de la implementación de LZA.

---

1. Descripción general del despliegue del Landing Zone Accelerator

El **AWS Landing Zone Accelerator (LZA)** ofrece un marco 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 red y automatización de cumplimiento.

Beneficios clave:

- Implementación de infraestructura estandarizada y repetible
- Configuración de cuentas segura y conforme con las mejores prácticas de AWS
- Gestión de configuraciones basada en GitOps
- Despliegues automatizados con AWS CodePipeline y AWS CodeBuild

Arquitectura de despliegue:

- **Control de código fuente:** Los archivos de configuración están almacenados en un repositorio Git (`https://github.org/org/lza-config-repo/`)
- **AWS CodeConnection:** Conecta el repositorio de forma segura con AWS CodePipeline
- **Flujo de trabajo en CodePipeline:**
  - *Etapa de origen:* Obtiene la última configuración desde el repositorio
  - *Etapa de construcción:* Valida la estructura YAML y la lógica de configuración usando AWS CodeBuild
  - *Etapa de despliegue:* Aplica los cambios al entorno de AWS

---

2. Flujo de trabajo para cambios en la configuración

Los cambios en la configuración siguen un enfoque **GitOps**, lo cual 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>/<lza-repo>.git
cd <lza-repo>

```

2. **Crear una rama de funcionalidad**

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

```

3. **Realizar cambios**

Editar uno o más de los archivos YAML de configuración, por ejemplo:

- `global-config.yaml`
- `accounts-config.yaml`
- `security-config.yaml`
- `networking-config.yaml`
- `organization-config.yaml`
- `iam-config.yaml`

4. **Hacer commit y push**

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

```

5. **Crear un Pull Request**

- Abre un PR en Bitbucket dirigido a la rama `main`
- Solicita revisión de código al equipo de seguridad/infraestructura designado

6. **Despliegue**

Una vez que se hace *merge* a `main`:

- Ve a la consola de AWS CodePipeline, selecciona `AWSAccelerator-Pipeline` y haz clic en *Release change*

- Etapas de la canalización: `Source → Build → Approval → Deploy`

7. **Monitorear y validar**

- Verifica el estado en la consola de CodePipeline
- Usa los logs de CloudWatch si es necesario para depurar
- Confirma los cambios exitosos en la consola de AWS

---

3. Archivos de configuración

`global-config.yaml`

Define la estructura base para el entorno LZA.

Características:

- Región principal y regiones habilitadas
- Retención de logs en CloudWatch
- Habilitación y versión de Control Tower
- Límites de seguridad (acceso root, MFA, acceso público a S3)
- Configuración de logging (buckets, reglas de ciclo de vida)
- Alertas de presupuesto y reportes de uso y costos

`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 logs centralizados
  - `Audit`: Cuenta de solo lectura y administración de seguridad delegada
- **Cuentas de carga de trabajo:**
  - `Development`: Entorno de pruebas y experimentación no productiva

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

---

4. Mejores prácticas para gestionar LZA

- Usar ramas y Pull Requests para todos los cambios de configuración

- Almacenar políticas personalizadas y documentos en directorios estructurados
- Monitorear el estado de la canalización con AWS CodePipeline y CloudWatch
- Habilitar servicios de seguridad adicionales como Security Hub, GuardDuty y Macie según se necesite
- Revisar regularmente los marcos de CIS/NIST para alineación de controles
- Mantener una separación clara entre cuentas de desarrollo, pruebas y producción mediante OUs