



ARQUITECTURA E IMPLEMENTACIÓN

Jonathan A. Vega Ortega

pragma



AGENDA


1. Análisis de caso
2. Marco de trabajo
3. Estrategia Organizacional
4. Estrategia Infraestructura
5. Estrategia Devops
6. IaC Cloudformation
7. Finalización

pragma

ANALISIS DE CASO

CCS es la Compañía Colombiana de Seguimiento de Vehículos. CCS se encarga del monitoreo y seguimiento de vehículos de carga. CCS instala sensores en camiones de carga, de forma que es posible en todo momento conocer la localización, velocidad, dirección, estado de la carga, temperatura de la carga, detenciones planeadas y no planeadas, y accidentes que pueda tener cada camión. Igualmente, cada vehículo tiene una cámara interna que graba todo lo que ocurre al interior de la cabina, y una cámara que graba todo el recorrido del camión. Finalmente, cada camión cuenta con un botón de pánico en caso de que se presente una emergencia.

pragma



CCS tiene una central en la que se consolidan todas las señales provenientes de los camiones. En esta central se analizan todos los estados recibidos y en caso de detectar una situación anómala o recibir una señal de emergencia, se informe tanto a las autoridades respectivas, cómo a organismos de socorro e interesados (propietario del camión). CSS actualmente tiene un proceso de ventas en el que requiere que un cliente vaya a una oficina, haga la solicitud para contratar la instalación de sensores para una serie de vehículos. En el proceso de venta se revisa la identidad del usuario y los ingresos de la persona o empresa que va a contratar los sensores para sus vehículos y de acuerdo con unas reglas de negocio establecidas en la empresa se da o no la autorización del contrato.

pragma

pragma



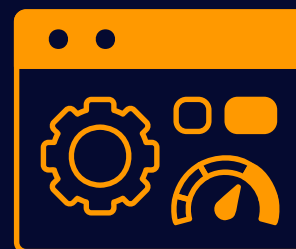
Cuando el contrato contempla la autorización de más de 50 vehículos se requiere la firma del Manager. Por otro lado, los clientes están pidiendo un servicio que les permita visualizar las estadísticas de localización, distancia de los recorridos, tiempo de movimiento de cada camión diario, mensual y anual, esto debido a que actualmente el sistema solo es utilizado para enviar alertas en casos específicos, por esta razón el manager de la compañía le gustaría crear un servicio plus para los clientes con el que les pueda entregar las funcionalidades que están solicitando y que adicional contempla un módulo de administración en el que el cliente pueda modificar el tipo de alertas que le gustaría recibir, en que correos o dispositivos le gustaría recibirlas, un módulo financiero en dónde pueda recibir todas las facturas y hacer los pagos mensuales y un módulo en el que pueda crear y modificar los usuarios que tendrían acceso al sistema con diferentes roles (Administración, Visualización, Compra y Aprobación de nuevos productos o planes).

MARCO DE TRABAJO



Excelencia Operacional

- Operar y monitorear de forma eficiente.



Eficiencia en rendimiento

- Uso óptimo de recursos y escalabilidad.



Seguridad

- Protección de datos y sistemas.



Optimización de costos

- Evitar gastos innecesarios.



Fiabilidad

- Tolerancia a fallos y recuperación.



Sostenibilidad

- Minimizar el impacto ambiental.

pragma

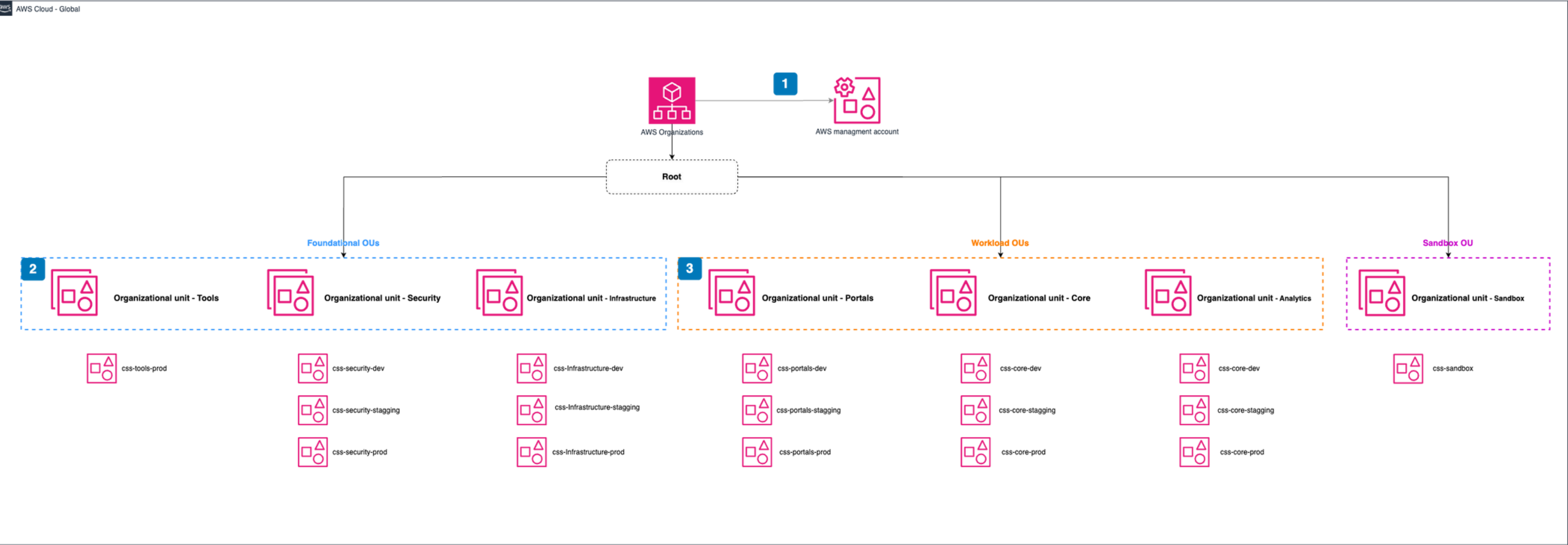


ESTRATEGIA ORGANIZACIONAL

pragma

Organizations Strategy - Compañía Colombiana de Seguimiento de Vehiculos

Arquitecto Cloud - Jonathan A. Vega Ortega



1 The management account manages the creation and control of all other accounts.

This central account should not be used for workloads; only for administration and consolidated billing.

2 This defines the basis of the architecture, grouping the accounts necessary for shared and security services. It includes:

- ✓ OU - Tools: Contains the `css-tools-prod` account, which hosts shared monitoring, management, and other common resources.
- ✓ OU - Security: Contains several accounts for dev, staging, and prod environments. This accounts houses the services that manage security for the entire organization.
- ✓ OU - Transversal: Centralizes auditing and logging. Security and compliance logs generated by all accounts and workloads are stored in these accounts (dev, staging, prod).
- ✗ Objective: Keep security, common tools, and auditing outside of the accounts where the applications run.

3 These OUs are focused on business applications and are organized by solution type or functional domain:

- ✓ OU - Portals: Contains the accounts (dev, staging, prod) for the company's portals or frontends.
- ✓ OU - Core: This houses the core services and applications (core-dev, core-staging, core-prod) that support the core business operations.
- ✓ OU - Analytics: Groups the accounts where analytics and business intelligence data are processed. Like the others, it has separate environments (dev, staging, prod).
- ✗ Objective: Separate and isolate each workload to improve governance, security, and reduce the blast radius in the event of an incident.

Well Architect Framework

Security: Account separation, the application of SCPs, and the centralization of logs and auditing strengthen security and reduce blast radius.

Reliability: An architecture with separate environments and isolated workloads facilitates recovery from failures and ensures resilience.

Operational Excellence: Improves visibility, control, and monitoring of each environment through organization by account and OU.

Cost Optimization: Allows you to identify costs per workload and optimize resource use more precisely.

Performance Efficiency: Facilitates independent scaling of each workload or environment according to operational and business requirements.



ESTRATEGIA INFRAESTRUCTURA

"Todo niño es un artista. El problema es cómo seguir
siendo artista una vez que crecemos."

— Pablo Picasso

pragma

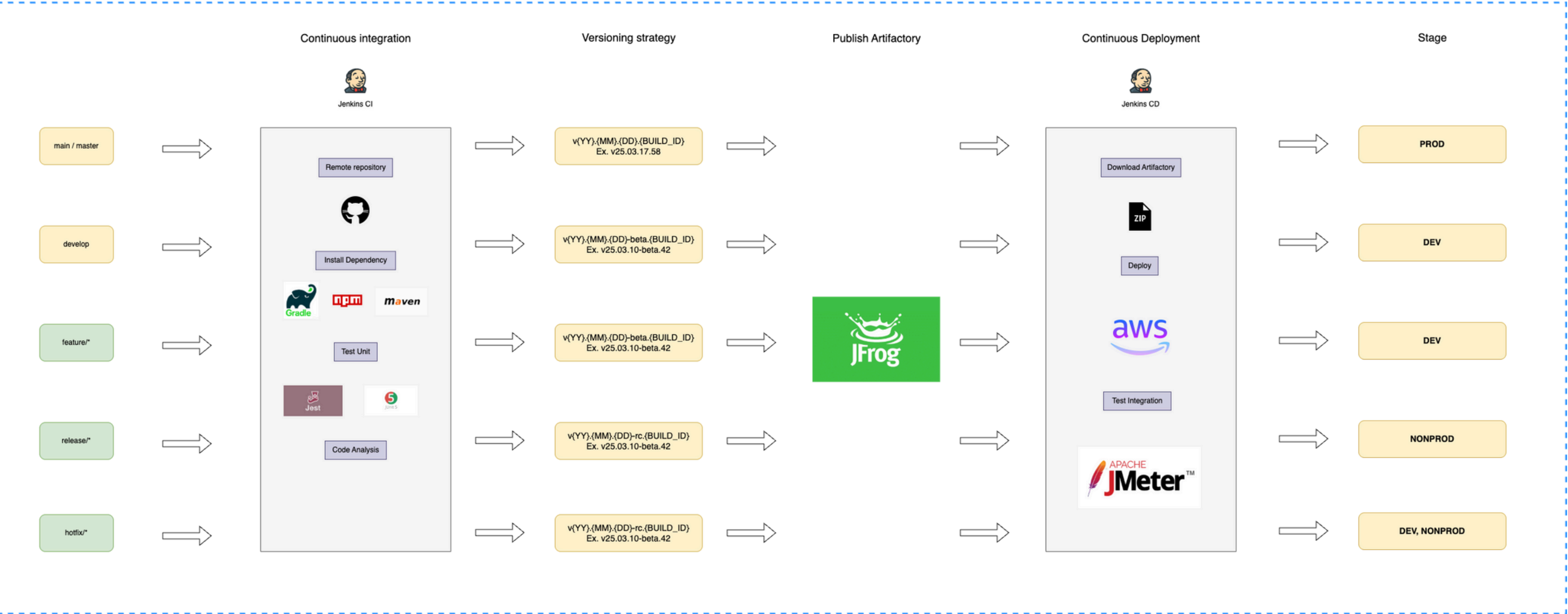


ESTRATEGIA DEVOPS

pragma

Devops strategy - Compañía Colombiana de Seguimiento de Vehiculos

Arquitecto Cloud - Jonathan A. Vega Ortega



Distributed DevOps

El modelo DevOps distribuido separa (o distribuye) las responsabilidades de operaciones de ingeniería de aplicaciones y operaciones de ingeniería de infraestructura entre los equipos de ingeniería, siguiendo la metodología Cloud operations and platform enablement (COPE).



IAC CLOUDFORMATION

pragma



GitHub



Jonathan1021/**css- cloudformation**



Implementation of infrastructure as code in cloudformation.

1

Contributor

0

Issues

0

Stars

0

Forks



Jonathan1021/css-cloudformation: Implementation of infrastructure as code in cloudformation.

Implementation of infrastructure as code in cloudformation. - Jonathan1021/css-cloudformation

 GitHub

[Ver Aquí](#)

The background is a solid dark blue color. On the left and right sides, there is a decorative pattern of orange lines forming a hexagonal grid. The lines are thick and the pattern is partially visible, extending from the edges towards the center.

GRACIAS