

Migrando On-Premises Datacenters a la Nube

Lecciones Aprendidas



Catherin Cruz



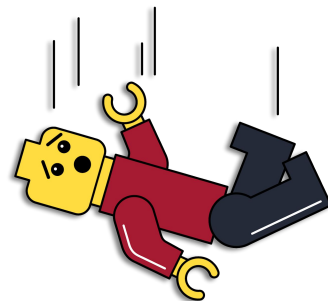
[@UserTwoGG](https://twitter.com/UserTwoGG)



github.com/twogg-git



linkedin.com/in/cruz-catherine



30 y 31 de Julio 2020

#DevopsdaysMDE2020





Lecciones Aprendidas



Definiendo las bases del proyecto



Preparando al equipo para la nube



Encapsulando servicios en contenedores



Provisionando infraestructura básica



Implementando CI/CD y GitOps



Pruebas en ambientes y Producción





Definiendo las bases del proyecto

Toda la migración exitosa **inicia conociendo al cliente**, ¿por qué quieren migrar? ¿cuales son los beneficios esperados?, ¿los procesos y ambientes?

- ✓ Estudia casos de éxito y también los que fracasaron
- ✓ Escoge bien el proveedor de servicios
- ✓ Haz un inventario, listados, mapas, investiga!
- ✓ Crea un plan de trabajo escalonado por equipos
- ✓ Documenta todo y actualiza constantemente





Definiendo las bases del proyecto



Muchas veces el cliente o tú quieren migrar primero los repositorios más estables, los pequeños que darán menos problemas, el **casi listo para la nube, pero...**

- ? Evalúa la prioridad y el beneficio
- ? Dependencias, librerías, scripts
- ? El equipo a cargo del repo
- ? Servicios a usar en el proveedor





Preparando al equipo para la nube

Momento de implementar el plan, **todos deben aprender idioma nube**, comenzar los entrenamientos, las tareas de investigación, la revisión de repos, y siempre tener presente los beneficios de la migración!

Motiva a la participación de las partes interesadas, (desarrollo, operaciones, testing), muchas veces ellos tienen identificados los procesos a mejorar.

Experiencia + Conocimiento + Equipo = Nube!!!





Preparando al equipo para la nube

- ★ Responsabilidades compartidas
- ★ Herramientas y flujos de trabajo
- ★ Grupos pequeños y enfocados
- ★ Socializar errores e investigarlos
- ★ Colaboración interdisciplinar
- ★ Objetivo general y por equipo





Encapsulando servicios en contenedores



Más del 75% de migraciones basadas en estrategias de contenedores reportan una reducción inicial y creciente del 10% del tiempo invertido en desarrollo.

Como unidades independientes estos servicios son más **fáciles de actualizar, y desplegar**, pero serán más difíciles de configurar, asegurar, y monitorear.

Sin embargo la nube ama los microservicios...





Encapsulando servicios en contenedores

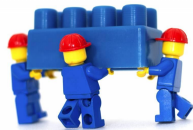
Contenedores



- ✓ Mayor escalabilidad
- ✓ Ahorra recursos (por consumo)
- ✓ Menor tiempo de desarrollo
- ✓ Mayor flexibilidad en cómputo

- ✗ Riesgos de seguridad
- ✗ SO menos flexible

Agarre y Vamonos



- ✓ Ahorra costo y tiempo
 - ✓ Reduce costos on-premises
 - ✓ Migración sencilla y rápida
- ✗ Costos a largo plazo
 - ✗ No usas la nube o sus servicios





Provisionando infraestructura básica

Infraestructura como código (IaC), el universo mágico donde operaciones configura y actualiza los servicios en el proveedor de forma controlable y versionada.

Aunque sea la infra básica se debe pensar en un **despliegue tipo producción**, tener en cuenta la seguridad, la validación de estado actual, los servicios actualizados con cada cambio, y el debug de errores.





Provisionando infraestructura básica

En este punto ya hemos identificado los componentes del datacenter y gracias a todo lo aprendido del proyecto y el proveedor, podemos decidir la infraestructura y/o configuración obsoleta, migrable, ó reusable...



- ✓ **Acceso:** VPC, IAM, CloudTrail
- ✓ **Red:** ALB, IG, Security groups
- ✓ **Servicios:** RDS, Bastion Host





Implementando CI/CD y GitOps

Listos usuarios, servidores, base de datos, red, y ahora?... **EL CÓDIGO**



Definir los pipeline y el ambiente inicial



Reusar los artefactos y entregables



Validar paso final, artefacto ó despliegue?



Establecer un formato y buenas prácticas



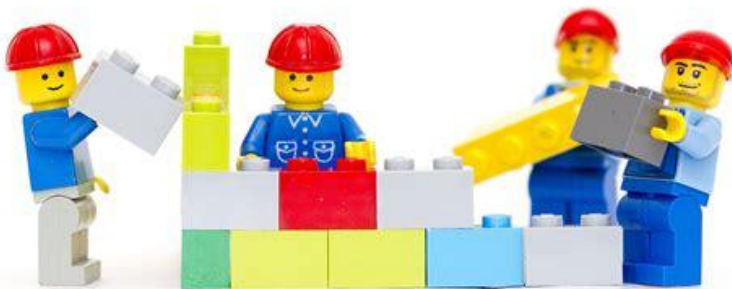


Implementando CI/CD y GitOps

Versionamento CI/CD



Infraestructura declarativa



- ♥ Fácil de migrar entre proveedores
- ♥ Mayor consistencia en configuración
- ♥ Automatizar tareas repletibles
- ♥ Implementación de buenas prácticas
- ♥ Administrar con Git las operaciones
- ♥ Monitoreo y retroalimentación





Pruebas en ambientes y Producción

Aunque no se implementa la misma seguridad ó data que producción, gran parte de **los errores se identifican en el ambiente inicial de desarrollo.**

Cada fase tiene su enfoque, no es lo mismo testear 200MB en una instancia mediana, que 5TB, 3 réplicas, 2 zonas, y sus certificados SSL por dominio.



Desarrollo > Testing > Staging > Producción





Pruebas en ambientes y Producción

Estamos en producción!, no fue fácil, podemos disfrutar unos 15min y de vuelta al laboratorio.

Momento de enfocarnos en optimizar y usar los recursos sacando la mayor ventaja de la nube.

Realizar monitoreo constante de los servicios, los logs de cada aplicación y ambiente.

Planear casos de recuperación a desastres...



Chaos Engineering



Serverless & Kubeless



DevSecOps Sabores





Referencias y Recursos



Migrate with AWS

aws.amazon.com/cloud-migration/
awsstatic.com/Migrating-to-aws-ebook
awscloud.com/AWS_Migration_8_Best_Practices-ebook



Learn to provision infrastructure with HashiCorp Terraform

learn.hashicorp.com/terraform



How to configure a production-grade CI-CD workflow for IaC

gruntwork.io/guides/automations/production-grade-ci-cd



Effective DevOps - Jennifer Davis, Ryn Daniels

oreilly.com/library/view/effective-devops



LEGO (All media and copyrights reserved)

lego.com/en-us/





Preguntas?...



GRACIAS!!!

Catherin Cruz

 @UserTwoGG

 github.com/twogg-git



30 y 31 de Julio 2020

#DevopsdaysMDE2020

