Propuesta Azure Kubernetes Service para kilimo

Deadline

Contexto

Actualmente, la aplicación de MMRV está alojada en la nube de Amazon (AWS), pero debido a los compromisos contractuales que tiene el proyecto de MMRV, el equipo plantea una migración desde su entorno actual en AWS para la nube de Microsoft (Azure).
La aplicación actualmente usa una solución contenerizada en AWS, por lo cual una opción sencilla y segura, sería adoptar la tecnología de Kubernetes en Azure, con el servicio Azure Kubernetes Service (AKS), ya que puede administrar y orquestar los contenedores que hoy existen en AWS, solo necesitando cambios menores y no una re-factorización de la aplicación.

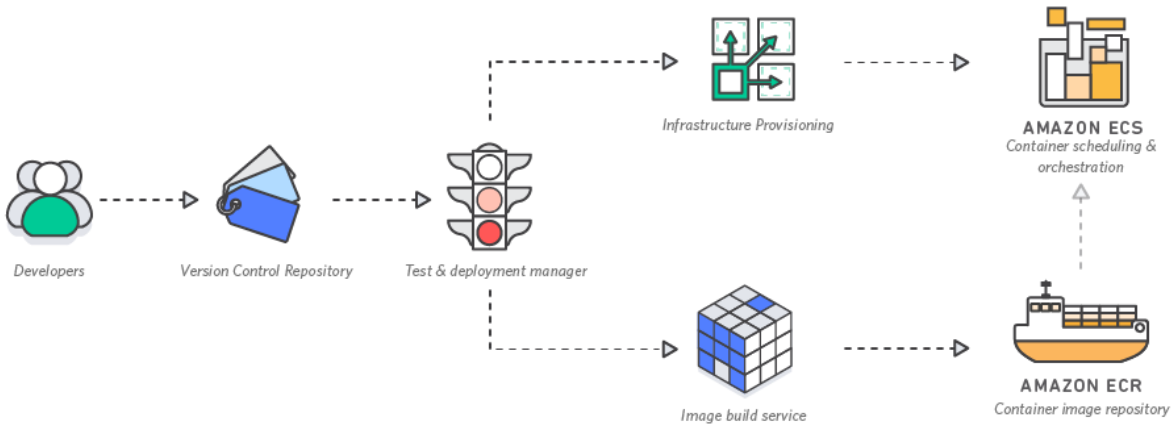
Decision

Kubernetes sería la mejor opción para homologar todo el servicio y nos acerca un poco más a ser cloud agnostic. Kubernetes es un software open source que puede nos puede dar la flexibilidad de llevar MMRV escalonadamente hacia Azure, ya que es posible seguir comunicándose con el resto de aplicaciones alojadas en AWS.

Infraestructura actual

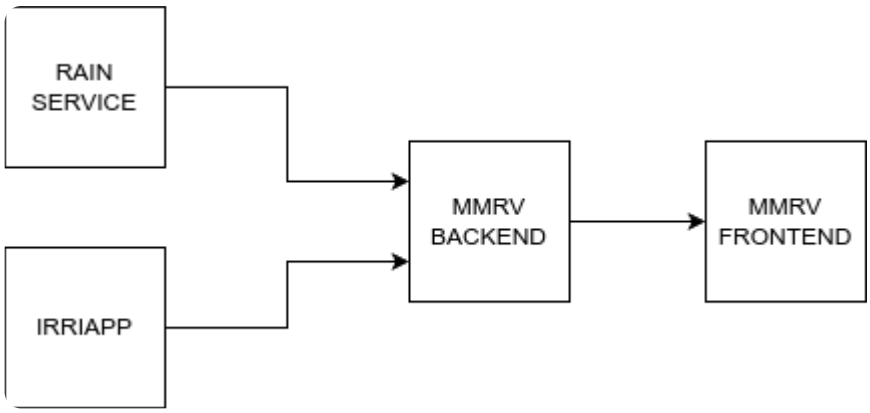
El servicio de MMRV tiene actualmente el código en el repositorio Github y de alli a través de un pipeline son construidas y enviadas las imágenes Docker a un repositorio en Amazon

ECR, y contenerizada en una instancia de Amazon ECS.



Front-end y Back-end

Actualmente, el Front-end de MMRV está construido con Next.js, es y desplegado con la solución serverless de AWS (amplify), el back-end, por otro lado, se encuentra contenerizado en una instancia de ECS. Tanto el back-end y el front-end actualmente son orquestados de manera independiente. Con la implementación de una solución Kubernetes, ambas partes de la aplicación puede ser orquestadas por un mismo recurso, simplificando su monitoreo, self-healing, seguridad, comunicación entre componentes, entre otros.



Base de datos

Actualmente, el servidor de base de datos ya se encuentra alojado en Azure

Implementación

Propongo un cronograma de fases para ser ágiles y mantener la entrega continua.

FASE 1 - staging

1. Creación de la siguiente infraestructura necesaria:

- Azure Container Registry(ACR)
- Azure Kubernetes Service (AKS)
- Managed Service Identity (MSI)
- (opcional) Firewall

2. Adaptación/creación de pipelines para el envío de la imagen Docker de MMRV Back-end
3. Despliegue MMRV back-end

FASE 2 - production

1. Creación de la siguiente infraestructura necesaria:

- Azure Container Registry(ACR)
- Azure Kubernetes Service (AKS)
- Managed Service Identity (MSI)
- (opcional) Firewall

2. Despliegue MMRV back-end

FASE 3 - staging

1. Adaptación/creación de pipelines para el envío de la imagen Docker de MMRV Front-end
2. Despliegue MMRV Front-end

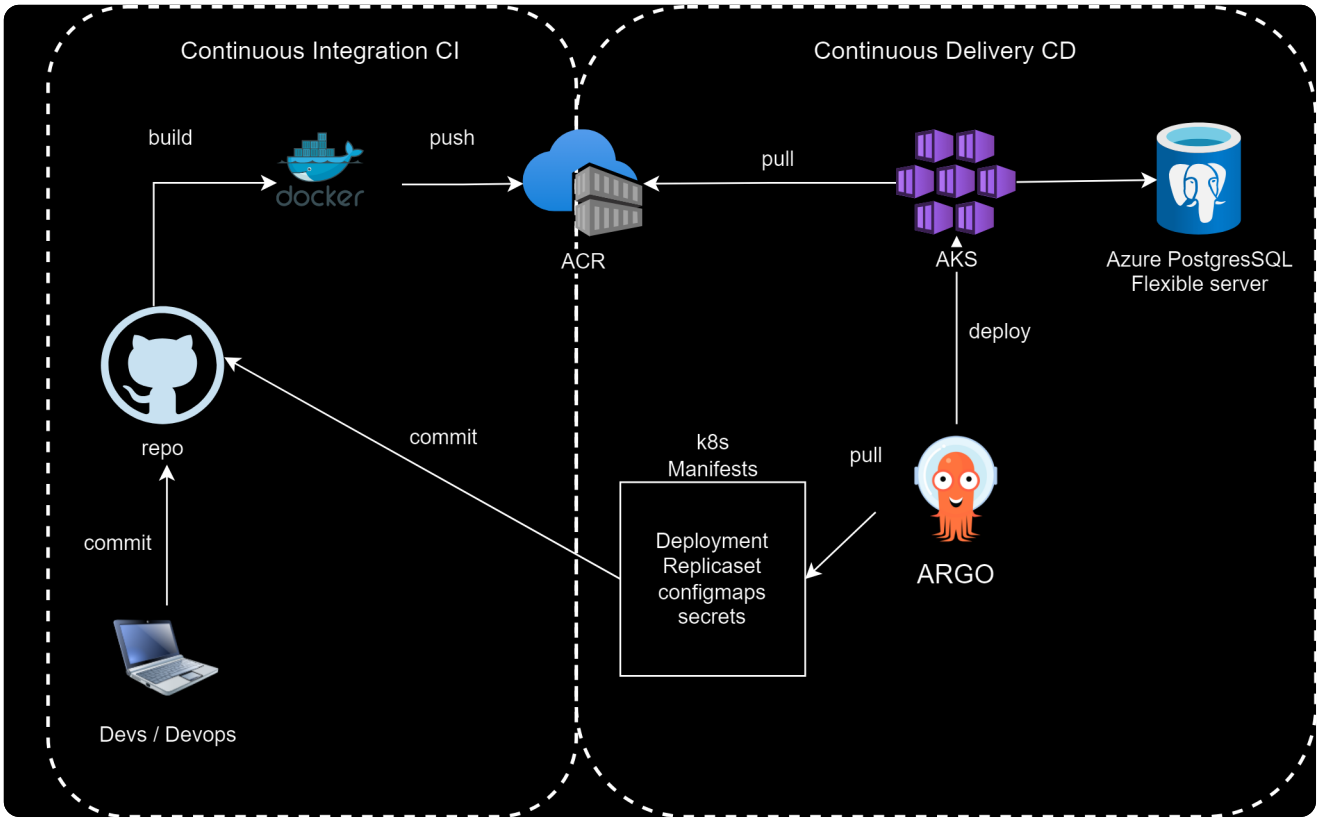
FASE 4 - production

1. Adaptacion/creacion de pipelines para el envio de la imagen Docker de MMRV Front-end
2. Despliegue MMRV Front-end

FASE 5

1. Adopción de metodologías GitOps
2. "Remedy" de deuda técnica generada en algunas de las fases anteriores.
3. Revisión del estado de la seguridad de todo el proyecto
4. Crear planes de contingencia para la recuperación de datos (DRP)

Futuro diagrama



Consecuencias

- **Desafíos técnicos imprevistos**: por ejemplo, una aplicación puede tener tantas dependencias que la refactorización o el cambio de plataforma pueden ser mucho más complejos y consumir más tiempo de lo que se pensaba en un principio.
- **Costos imprevistos**: sin una planificación adecuada, las empresas pueden incurrir en gastos que no habían presupuestado, como nuevas tasas de licencia o costos de capacitación en las nuevas herramientas para los empleados.
- **Tiempo de inactividad inesperado**: los cambios importantes en una aplicación pueden generar conflictos o problemas que provoquen un tiempo de inactividad no planificado, tanto en la aplicación como en los sistemas conectados o dependientes.
- **Adaptación de pipelines y homologación de otros servicios en Azure:** ejemplo empezar a depositar las imágenes Docker en la registry de Azure. Usar Azure key vault en vez de AWS secret manager, entre otros.

Estado

En revision 22/03/2024

Propetario

Equipo DevOps

Fuente

https://thenewstack.io/primer-how-kubernetes-came-to-be-what-it-is-and-why-you-should-care/

https://medium.com/@cuemby/qu%C3%A9-es-ser-cloud-agnostic-y-por-qu%C3%A9-debes-usarlo-3ae2b8c3553

https://aws.amazon.com/es/ecs/getting-started/

Anexo