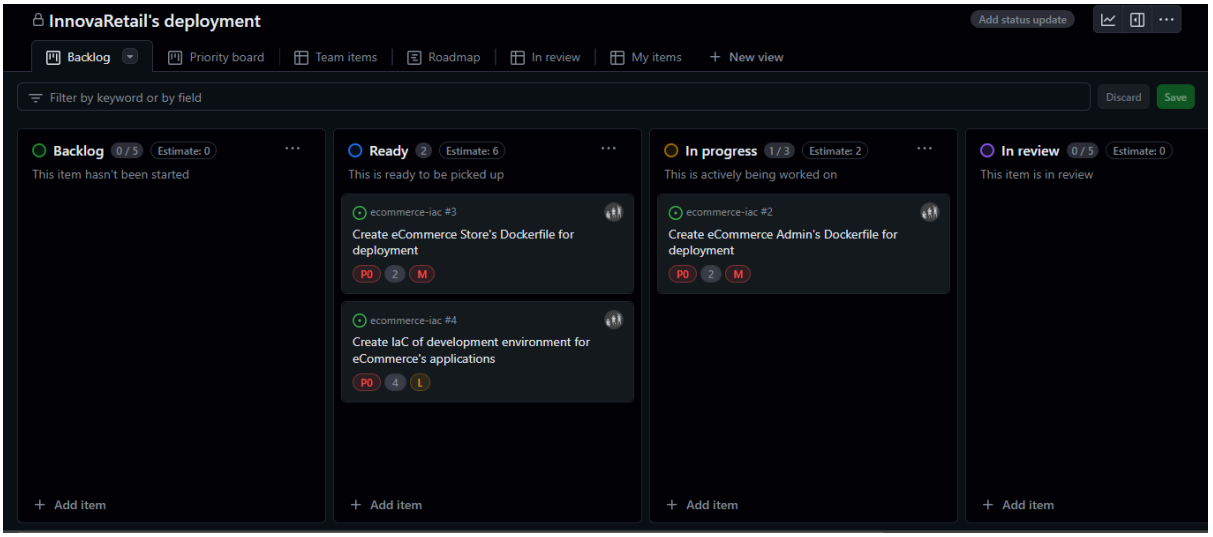
¿Quienes somos?

Automat Systems es una empresa vanguardista especializada en la automatización de infraestructura DevOps. Con un enfoque en la eficiencia y la innovación, proporcionamos soluciones personalizadas para el despliegue de servicios en la nube y la automatización de procesos. Nuestro equipo de expertos trabaja incansablemente para asegurar que cada cliente alcance la excelencia operativa, reduciendo costos y mejorando la seguridad y la resiliencia de sus sistemas.

Metodología:



Cómo metodología para desarrollar este proyecto, se decidió utilizar la metodología kanban, que permite: Visualización del flujo de trabajo: al utilizar tableros visuales para representar el flujo de trabajo. Esto permite a los equipos ver de manera clara y concisa las tareas pendientes, en progreso y completadas. Para un proyecto DevOps, esto es útil para rastrear las etapas de desarrollo, pruebas, implementación y monitoreo.

1. Flexibilidad y adaptabilidad: Kanban se basa en principios ágiles y se adapta bien a los cambios. En un entorno DevOps, donde las prioridades y los requisitos pueden cambiar rápidamente, Kanban permite ajustar el flujo de trabajo según sea necesario.
2. Limitación del trabajo en progreso (WIP): Kanban fomenta la limitación del WIP. Esto significa que solo se trabaja en un número específico de tareas a la vez. Para un proyecto DevOps, esto ayuda a evitar la sobrecarga y a mantener un enfoque constante en las tareas más importantes.
3. Mejora continua: Kanban promueve la mejora continua. Los equipos pueden analizar métricas como el tiempo de ciclo y la eficiencia para identificar áreas de optimización en el proceso DevOps.
4. Colaboración y transparencia: Kanban fomenta la colaboración entre los miembros del equipo y la transparencia en el proceso. Esto es fundamental

para un enfoque DevOps, donde la comunicación y la colaboración son esenciales.

Branching:

Para la estrategia de branching, decidimos utilizar Branching basado en ambientes, ya que, el uso de ramas basado en ambientes es una práctica que permite gestionar el ciclo de vida del software en diferentes etapas, desde desarrollo hasta producción.
Esto permite que haya:

1. Seguridad y control: Las ramas de ambiente permiten controlar quién puede realizar cambios en cada etapa.
2. Pruebas específicas: Cada rama de ambiente se puede configurar con pruebas automatizadas específicas para ese entorno.
3. Despliegue gradual: Las ramas de ambiente facilitan la implementación gradual.
4. Retroalimentación temprana: Al tener una rama de desarrollo separada, los equipos pueden recibir comentarios tempranos de los stakeholders y corregir problemas antes de llegar a producción.

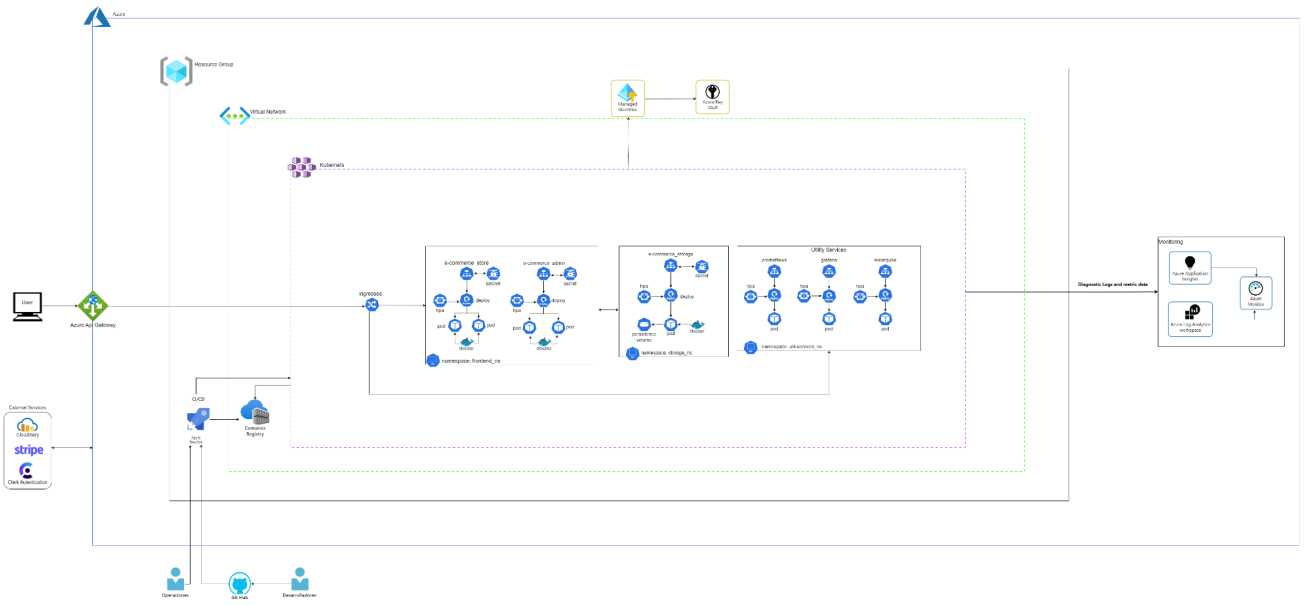
Contexto del problema

La empresa InnovaRetail Corp se encuentra en la necesidad de mejorar su presencia en línea y optimizar la gestión interna de su negocio. Para lograrlo, han decidido implementar una plataforma de comercio electrónico que abarque tanto una tienda en línea como un sistema de administración avanzado (CMS)

Solución

Se busca desplegar una plataforma de comercio electrónico conformada por una tienda en línea y un sistema de administración cms implementados en un cluster de microservicios en kubernetes junto con los servicios de microsoft azure

Arquitectura de la solución:



Enlace al diagrama en draw.io:
https://drive.google.com/file/d/17qFggd-xnOjAS7iI7Sr_biwl6dbrr8kD/view?usp=sharing

Esta arquitectura está alineada con una solución basada en la nube la cual busca desplegar una plataforma de comercio electrónico conformada por una tienda en línea y un sistema de administración cms implementados en un cluster de microservicios en kubernetes junto con los servicios de microsoft azure

Componentes de Azure

Grupo de recursos: es un contenedor que almacena los recursos relacionados con una solución de Azure. El grupo de recursos puede incluir todos los recursos de la solución o solo aquellos que se desean administrar como grupo.

Redes virtuales Azure: son entornos aislados y altamente seguros para ejecutar máquinas virtuales (VM) y aplicaciones. Esta arquitectura de referencia utiliza una topología de red virtual hub-spoke. La red virtual central contiene el cortafuegos Azure y las subredes Azure Bastion. La red virtual radial contiene las subredes del sistema AKS y del grupo de nodos de usuario y la subred de Azure Application Gateway.

Azure Kubernetes Service (AKS): ofrece la forma más rápida de empezar a desarrollar e implementar aplicaciones nativas de nube en Azure, centros de datos o en el perímetro con canalizaciones integradas de código a nube y límites de protección.

Azure Application Gateway: es un equilibrador de carga de tráfico web que administra el tráfico a las aplicaciones web. Este tipo de enrutamiento se conoce como equilibrio de carga de capa de aplicación.

Azure Key Vault: para almacenar y acceder de forma segura a secretos, como claves de API, contraseñas y certificados.

Azure DevOps admite una cultura colaborativa y un conjunto de procesos que reúnen a desarrolladores, administradores de proyectos y colaboradores para desarrollar software.

Azure Container Registry: permite compilar, almacenar y administrar imágenes y artefactos de contenedor en un registro privado para todo tipo de implementaciones de contenedor. Use los registros de contenedor de Azure con el desarrollo de contenedores y las canalizaciones de implementación existentes.

Azure Application Insights: es un servicio que ofrece la nube de Microsoft (Azure), que proporciona herramientas para monitorizar, analizar y detectar errores de rendimiento en las aplicaciones alojadas en dicha nube. Además, también ofrece la posibilidad de insertar trazas personalizadas y registrar errores en estas aplicaciones.

Log Analytics: es una herramienta de Azure Portal que se usa para editar y ejecutar consultas de registros en los datos que se recopilan en los registros de Azure Monitor, y analizar los resultados de forma interactiva.

Azure Monitor: es un nombre que abarca una colección de herramientas diseñadas para proporcionar visibilidad del estado del sistema. Ayuda a entender el rendimiento de los servicios nativos de nube y permite identificar de manera proactiva los problemas que les afectan.

Workflow

El usuario interactúa con nuestra solución a través del Azure Application Gateway, que actúa como punto de entrada. Este servicio dirige el tráfico al clúster de Kubernetes de Azure, donde se despliegan los microservicios necesarios para el funcionamiento del e-commerce. En total, desplegamos 6 microservicios:

1. e-commerce_store y e-commerce_admin: Contienen la lógica de funcionamiento del frontend y backend de la aplicación respectivamente.
2. e_commerce_storage: Gestiona el almacenamiento utilizado por la aplicación, utilizando MySQL como base de datos.
3. Prometheus, Grafana y SonarQube: Mejoran la administración y el rendimiento de los pods, proporcionando monitoreo y análisis.

Cada microservicio cuenta en su mayoría con dos réplicas en sus pods y tiene un Horizontal Pod Autoscaler (HPA) que supervisa y ajusta automáticamente el número de réplicas en función de la carga.

Para el despliegue de los microservicios, utilizamos un Deployment en Kubernetes. Todas las imágenes necesarias para el despliegue se encuentran en contenedores Docker. La seguridad de los microservicios está gestionada por el Managed Identity de Azure y Azure Key Vault, asegurando un acceso seguro a las credenciales y secretos.

La gestión del clúster de Kubernetes se divide en dos roles: Operaciones y Desarrollo. El rol de Operaciones realiza cambios en el clúster a través del portal de Azure DevOps, mientras que el rol de Desarrollo interactúa con el clúster a través de GitHub, Azure DevOps y el Azure Container Registry para realizar el push de imágenes actualizadas o nuevas.

Por último, implementamos un sistema de monitoreo tanto interno como externo. Para el monitoreo interno de los microservicios, utilizamos las herramientas Prometheus, Grafana y SonarQube. Para el monitoreo de recursos, como el API Gateway, el Ingress y el uso de recursos de los servicios de Azure, empleamos Azure Monitor, que nos proporciona visibilidad completa y capacidades de análisis avanzadas.

Pricing:

Application Gateway	Nivel Basic, tamaño de instancia Pequeña: 1 instanci...	Por adelantado: 0,00 ...	Mensualmente: 18,25 ...
Azure Kubernetes Service (AKS)	Gratis (no es de producción); La administración de c...	Por adelantado: 0,00 ...	Mensualmente: 43,34...
Azure SQL Database	Base de datos única, Núcleo virtual, Uso general, Ap...	Por adelantado: 0,00 ...	Mensualmente: 15,69...
Azure DevOps	3 Usuarios de licencias del plan Básico, 0 usuarios d...	Por adelantado: 0,00 ...	Mensualmente: 0,00 ...
Registro de contenedor	Nivel Basic, registro 1 x 30 días, 10 GB Almacenamie...	Por adelantado: 0,00 ...	Mensualmente: 6,00 ...
Key Vault	Almacén: 0 operaciones, 0 operaciones avanzadas, ...	Por adelantado: 0,00 ...	Mensualmente: 0,00 ...

Soporte

SOPORTE:

Incluido — 0,00 US$

Seleccione su programa u oferta

PROGRAMA DE LICENCIAS:

Contrato de cliente de Microsoft (MCA) — Iniciar sesión para ver los precios de su contrato Azure.

Mostrar Precios De Desarrollo Y Pruebas

Costo inicial estimado	0,00 US$
Coste mensual estimado	83,28 US$

Se calcularon los costos estimados de los servicios a utilizar para el despliegue de la solución escogida, teniendo en cuenta un application gateway, un AKS, un azures SQL database, Azure DevOps, un registro de contenedor y Key vault, todos dimensionados en una escala pequeña/mediana por las dimensiones de la infraestructura ya existente.
Este cálculo arrojó un bajo costo de operación para un despliegue adecuado.

Implementación

Despliegue de la Aplicación de Ecommerce en Kubernetes

A continuación se detallaran los pasos que se siguieron para desplegar la aplicación de ecommerce en kubernetes

Paso 1: Configuración del Volumen Persistente para la Base de Datos MySQL
Se creó un volumen persistente de 2GB para almacenar los datos de la base de datos MySQL. Este volumen persistente se configuró con acceso de lectura y escritura una sola vez.

Paso 2: Despliegue de la Base de Datos MySQL
Se configuró un despliegue de MySQL con una réplica única utilizando la última imagen de MySQL disponible. Se montó el volumen persistente creado anteriormente en el contenedor de MySQL para almacenar los datos de la base de datos. Se configuraron variables de entorno para establecer la contraseña de root y el nombre de la base de datos.

Paso 3: Configuración del Servicio para MySQL
Se creó un servicio de tipo ClusterIP para exponer el puerto 3306 utilizado por MySQL dentro del clúster de Kubernetes. Este servicio se asoció al despliegue de MySQL utilizando etiquetas de selección.

Paso 4: Despliegue de la Interfaz de Usuario de Administración de Ecommerce
Se configuró un despliegue para la interfaz de usuario de administración de ecommerce. Este despliegue utiliza una réplica única y la última imagen disponible de la aplicación de administración de ecommerce. Se establecieron variables de entorno para configurar la URL de la base de datos, las claves de acceso a servicios externos y otras configuraciones necesarias.

Paso 5: Configuración del Servicio para la Interfaz de Usuario de Administración
Se creó un servicio de tipo LoadBalancer para exponer la interfaz de usuario de administración de ecommerce fuera del clúster de Kubernetes. Este servicio se asoció al despliegue de la interfaz de usuario de administración utilizando etiquetas de selección y se mapeó el puerto 3000 utilizado por la aplicación.

Con estos pasos, se completó con éxito el despliegue de la aplicación de ecommerce en un clúster de Kubernetes, asegurando la disponibilidad y la escalabilidad de la aplicación.

Implementacion en terraform

Configuración de Infraestructura en Microsoft Azure para Kubernetes

Paso 1: Configuración del Grupo de Recursos
Se creó un grupo de recursos en Azure para organizar los recursos relacionados con la aplicación. Este recurso se configuró con el nombre y la ubicación proporcionados por las variables de entrada.

Paso 2: Configuración del Registro de Contenedores
Se creó un registro de contenedores en Azure para almacenar las imágenes de contenedores de la aplicación. Este registro se asoció al grupo de recursos creado anteriormente y se configuró con una SKU básica. Se habilitó la opción de administrador para permitir el acceso al registro.

Paso 3: Configuración del Clúster de Kubernetes
Se creó un clúster de Kubernetes en Azure utilizando el servicio Azure Kubernetes Service (AKS). El clúster se configuró con un nombre único, una ubicación y se asoció al grupo de recursos creado anteriormente. Se estableció un prefijo DNS para el clúster y se configuró un grupo de nodos predeterminado con una sola instancia de máquina virtual de tamaño estándar.

Paso 4: Asignación de Rol para el Acceso al Registro de Contenedores
Se asignó un rol al identificador principal del clúster de Kubernetes para permitir el acceso al registro de contenedores. Se configuró el rol de "AcrPull" para permitir la recuperación de imágenes de contenedor desde el registro. Se desactivó la verificación del principal de servicio AAD para evitar conflictos.

Paso 5: Configuración de Salidas
Se configuraron salidas para proporcionar información importante sobre la configuración del clúster de Kubernetes. Se incluyó la certificación del cliente y la configuración de Kubernetes en las salidas, marcadas como sensibles para proteger la información confidencial.

Con estos pasos, se completó con éxito la configuración de la infraestructura en Microsoft Azure para alojar y gestionar un clúster de Kubernetes, asegurando la disponibilidad y la seguridad de la aplicación desplegada.

Bibliografía

- https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices
- https://learn.microsoft.com/en-us/azure/architecture/example-scenario/serverless/microservices-with-container-apps
- https://learn.microsoft.com/en-us/azure/architecture/web-apps/app-service/architectures/basic-web-app?tabs=cli
- https://learn.microsoft.com/en-us/azure/architecture/example-scenario/apps/ecommerce-scenario
- https://learn.microsoft.com/en-us/azure/architecture/web-apps/
- https://learn.microsoft.com/en-us/azure/architecture/web-apps/app-service/architectures/baseline-zone-redundant
- https://learn.microsoft.com/en-us/azure/architecture/web-apps/serverless/architectures/web-app
- https://learn.microsoft.com/en-us/azure/architecture/web-apps/app-service/architectures/multi-region
- https://learn.microsoft.com/en-us/azure/architecture/guide/devsecops/devsecops-on-aks
- https://learn.microsoft.com/es-es/azure/azure-resource-manager/management/manage-resource-groups-portal
- https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices-advanced