1. **Cual es la diferencia entre nube pública, privada e híbrida**

Nube pública: se pudiese definir como un conjunto de servicios de computación ofrecidos por proveedores externos. Estos servicios son mantenidos por estos proveedores y están a la disposición de todo el qué pueda contratarlos. Los servicios más comunes son los de:

- IaaS: Infraestructura como servicio
- PaaS: Plataforma como servicios
- SaaS: Software como servicios

Nube Privada: Es el conjunto de recursos de computación qué están a disposición únicamente de una organización y los mismo son provisionados y mantenidos por esta. El acceso a estos recursos se realiza a través de conexión de área local y/o VPNs.

Nube Híbrida: Es un estrategia qué se utiliza para desplegar soluciones en entornos de nube pública y en premisa (local) haciendo uso de los beneficios qué ofrece cada uno. Por ejemplo existen aplicaciones o servicios qué por temas regulatorios necesitan correr en premisas mientras algunos servicios no críticos pudieran desplegarse en entornos de nube pública.

2. **Describa tres prácticas de seguridad en la nube**
    a. **Utilización de segmentos de red**: Desplegar recursos en distintos segmentos de red o subnets permite aumentar la seguridad entre servicios y aplicaciones limitando el acceso no necesario a estas. Una práctica muy sugerida es el uso de subredes privadas para recursos críticos como bases de datos donde no se tenga acceso desde internet. Esto permite además poder aplicar controles de tráfico entre redes.

    b. **Utilización de encriptación in transit and at rest**: Esto permite proteger la información qué viaja de un punto a otro de forma segura; evitando qué se transmita en texto plano y pueda ser leída en caso de ser interceptada por algún ataque de ciberseguridad y en el caso de la encriptación at rest ayuda a qué la información qué está guardada o en reposo pueda ser utilizada en caso de algún evento de seguridad.

    c. **Implementar controles de accesos y autenticación a recursos**: Esto permite limitar el acceso a los recursos en la nube a aquellos usuarios qué hemos definido y la aplicación de controles de acceso permite qué estos usuarios una

vez autenticados solo puedan realizar acciones sobre los recursos qué tienen previamente definidos.

3. Qué es IAC, y cuales son sus principales características

Iac: infraestructura como código permite la automatización en la creación de recursos de computación. Esto permite lograr qué el proceso de creación de infraestructura sea replicable y consistente. El uso de Iac permite qué se puedan aplicar buenas prácticas de desarrollo de software a la gestión de la infraestructura como el uso de control de versiones además de qué reduce el error humano

- Terraform:
    - Permite el aprovisionamiento de infraestructura utilizando HCL (hashicorp language). Lo cual permite una fácil adopción y una gran flexibilidad.
    - Idempotente: Terraform lleva un control del estado de infraestructura qué se creó y si intentamos correr nuevamente no sobreescribirá los recursos creados.
    - Integración con una gran cantidad de proveedores de cloud.
- Asible:
    - Es una herramienta de infraestructura como código con enfoque especial en el manejo de configuración, escrita en python.
    - Utiliza jugadas o playbook para describir los pasos de configuración a ejecutar
    - No necesita la instalación de agentes en los host donde se ejecutarán los playbooks

4. ¿Qué métricas considera esenciales para el monitoreo de soluciones en nube ?

- Métricas de uso de recursos: Van a variar de acuerdo al servicio cloud qué se utilice pero el objetivo es poder tener visibilidad del uso de recursos provisionados a cada servicio y de esta forma gestionar la capacidad de una forma más eficiente.
    - Uso de memoria RAM
    - Uso de CPU
    - Uso de espacio de Almacenamiento

- Métricas asociadas a la disponibilidad:
    - Uptime: es el tiempo de disponibilidad
    - MTTR: Es el promedio de tiempo entre fallas

- Métricas asociadas al rendimiento:
    - Latencia : El tiempo qué tardan los datos en moverse de un punto A a un punto B
    - Tasa de error: Es la relación entre las solicitudes fallida y el total de solicitudes
    - Throughput: La cantidad de datos qué se pueden procesar en un tiempo determinado.

5. ¿Qué es docker y cuales son sus componentes principales?

Docker es un tecnología qué permite crear contenedores. Los contenedores permiten correr procesos de forma aislada en un host.

Componentes principales

- Client: Es el API qué interactúa con el daemon de docker
- CLI: Es la interfaz de línea de comandos de docker, qué permite la interacción con el docker engine de una forma más fácil.
- Dockerd: : Es el daemon de docker, es el servicio que crea las imágenes de docker.
- Contenedor: Es una instancia de una imagen de docker
- Imagen: Las imágenes son archivos estáticos formados en capas qué contiene la información de la aplicación. Son una especie de molde para la creación de contenedores.
- Dockerfile: Es un archivo qué describe una serie de pasos necesarios para la creación de una imagen. Es la receta para la creación de una imagen
- Registry: Es un repositorio de imágenes, sirve para compartir y distribuir imágenes públicas o privadas.
- Docker-compose: Es una herramienta de docker qué facilita la creación de aplicaciones qué cuentan con varios contenedores. Permite la definición de recursos en un archivo YAML.

6. **Caso Practico**

Diseño de arquitectura para una aplicación nativa de nube.

Caracteristicas:

- Frontend: una aplicación web qué los clientes utilizarán para la navegación.
- Backend: servicios qué se comunican con la base de datos y el frontend
- Base de datos: Un sistema de gestión de base de datos qué almacene información.
- Almacenamiento de objetos: para gestionar imágenes y contenido estático

Detalles del diseño

Proveedor de nube: AWS

1. **Servicios utilizados:**
    - **Route53:** Gestión de dominio y enrutar el tráfico hacia cloudfront
    - **Cloudfront:** Es un servicio de CDN que distribuye archivo estáticos, mejora la latencia y ayuda a la distribución de contenido en varias zonas geográficas mejorando así la experiencia de usuario.
    - **ACM:** amazon certificate manager, nos permitirá administrar los certificados de seguridad ssl/tsl, asegurando qué los datos de nuestra aplicación estén encriptados cuando viajen por internet.
    - **Amazon WAF:** Es el servicio de Firewall web. Es la primera capa de seguridad de la aplicación permite la aplicación de reglas de seguridad para limitar el tráfico malicioso.
    - **Amazon s3:** Servicio de almacenamiento de archivos estáticos
    - **EC2:** Es el servicio de instancias o servidores donde están siendo desplegados los componentes de la aplicación web.
    - **ELB:** El servicio de elastic load balancer, estará balanceando el tráfico entre las diferentes instancias.
    - **Amazon RDS MySQL :** Es el servicio de bases de datos relacionales gestionado qué ofrece aws.
    - **IAM:** Este servicio permitirá gestionar los aspectos de autenticación, control de acceso y definir los roles y permisos para los servicios.

- **Cloudwatch:** Será nuestro sistema de monitoreo y gestión de logs de nuestra infraestructura..

2. **Descripción**
    a. **Frontend**

    En esta capa de la aplicación de cara a los usuarios. Esta es la única capa de la aplicación qué podrá ser alcanzada desde interntet. El frontend de la aplicación será desplegado en un grupo de escalamiento de instancias de ec2 en dos zonas de disponibilidad de la región us-east-1.

    Las instancias serán desplegadas en dos subredes públicas, una por cada zona de disponibilidad.El tráfico entre instancias será distribuido por un ALB.

    b. **Backend**

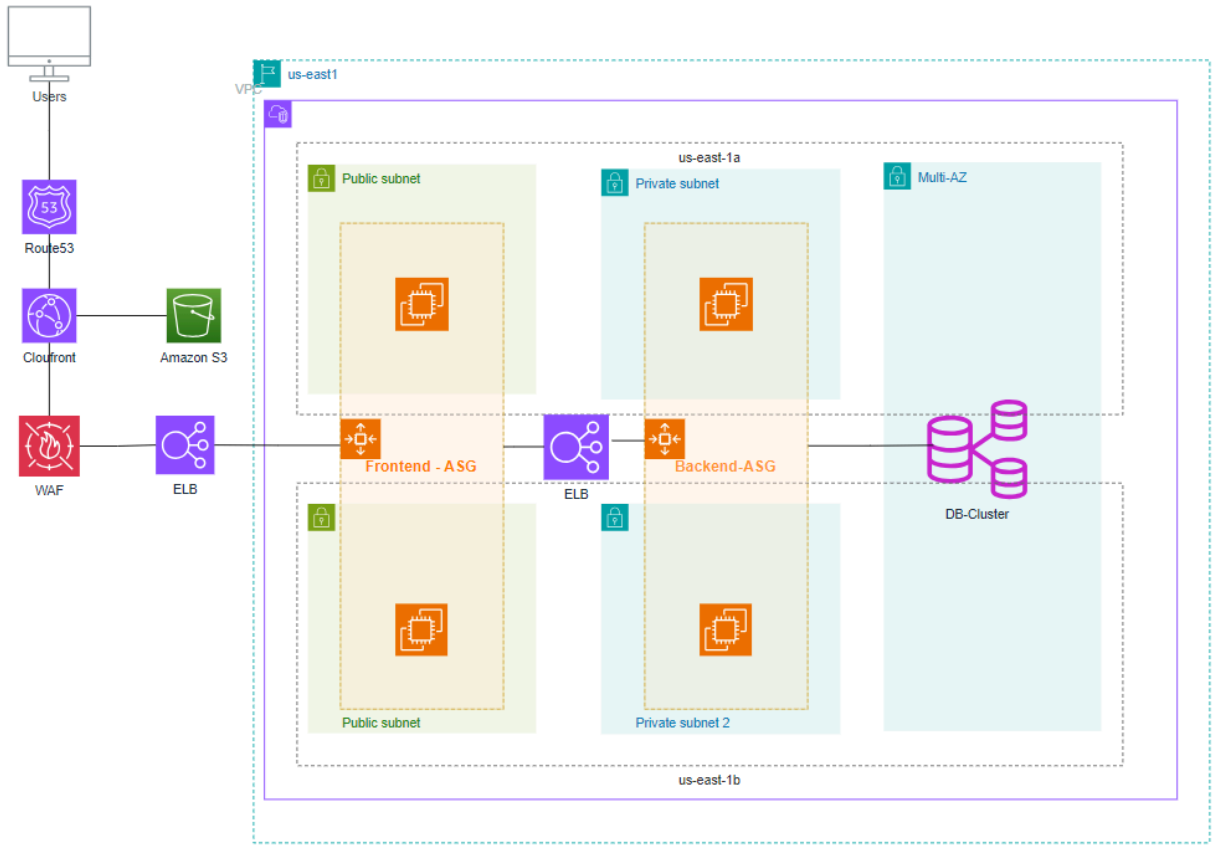
    El componente de backend de la aplicación será desplegado en un grupo de escalamiento de instancia de ec2 en subredes privadas en dos zonas de disponibilidad en la región us-east-1. El tráfico entre instancia será distribuido por un elb. El único tráfico permitido hacia esta subred será proveniente del segurity group configurado para el frontend.

    Este es el único componente de la aplicación con acceso a la base de datos.

    c. **Datos**

    Para la capa de datos estaremos desplegando un clúster de Amazon RDS con Mysql. El cluster será desplegado en multi-az deployment. Con dos instancias. Una instancia de escritura/master y una instancia read replica(solo lectura.) Para asegurar estabilidad y disponiblididad.

Diagrama de la aplicion.



Git repo: https://github.com/wilmost/wepapplication.git