Descripción mapa de red:

Aclaración: El mapa queda lo más resumido posible y hay ciertas características que no pude plasmar ya que la versión gratuita de Lucidchart me limitó a cantidad de figuras para usar. Por ejemplo la primer figura de “usuario” está faltante pero iría como primera acción llamando al Route53 como comienzo del flujo.

Por medio de la resolución de dominio en route53 poseemos acceso a la aplicación frontend desplegada en cloudfront + s3 permitiendo el desarrollo en js como por ejemplo algún desarrollo en react.js. Al estar en cloudfront permite la alta disponibilidad tras poder estar en HA distribuidos en multiples AZs.

A su vez el acceso al Internet gw dentro de la vpc (con su respectiva acl) cuya primer comunicación posee el Aplication Load Balancer el cual elegí este por ser el más óptimo para resolución de rutas para la aplicación backend y comunicación con el IngressController. En caso de necesidad de baja latencia y manejo de servicios internos sin ingress (manejo de red capa 4) se podría usar un NLB.

La aplicación backend está deployada sobre EKS brindando HA y escalabilidad en aws. Esta misma se encuentra en 2 nodos, cada uno en distintas AZs y dentro de target group permitiendo balanceo de carga (utilizando affinity) y usando autoscaling groups permitiendo escalabilidad dinámica.

Las bases de datos decidí por practicidad usar los servicios de aws mongo atlas y rds de postgresql, el cual por medio de security groups permitirían el acceso por ip al estilo White list, o bien se podría implementar un ha-proxy permitiendo el acceso por medio de una ip fija. El mongo atlas posee su propia White list. Como medida de seguridad deben llevar roles IAM además de security groups.

Porque elegí deployarlas en subnets privadas? Principalmente lo pensé que el acceso sea dentro del cluster al estilo llamadas de apis de ws. Si fuese necesario se podrían atachar a subnets publicas.

Como se expone eks hacia internet? Por medio de nat gateways permitiendo el aislamiento de las subnets privadas hacia internet sin exponerse lo cual permitiría el consumo de webservices externos.

Tener en cuenta que para mejor disponibilidad se podría generar el despliegue del cluster en 3 AZs pero para modo de ejemplo representé simplemente 2 (us-east-2 y us-east-1) Ohio y virginia, como ejemplo de situación real donde estas 2 regiones cuentan con costos mas bajos que el resto (ejemplo Brasil o chile) en estos casos que se despliegue de cero una infraestructura habría que analizar lo que se denomina principios de compliance, latency, cost, service and features en ese orden. Ademas se podrían desplegar mas nodos en diferentes subnets privadas, pero por el mismo motivo de ejemplificación lo deployé en 2 nodos.