1) Crear Security Group para Cluster kafka.
    a) EC2
    b) Redes y Seguridad – SECURITY GROUP
        1) Crear nuevo Security Group
            a) Nombre : kafka-cluster
            b) Descripción: Texto descriptivo
            c) Reglas de Entrada
                i. Agregar Regla de Entrada
                ii. Se abre puerto 9092 (Puerto de Kafka)
                iii. Origen: Desde cualquier lugar
                iv. Crear Grupo de Seguridad
                v. Se genera Security Group
            d) No se modifica Regla de Salida
            e) Se vuelve a Grupo de Seguridad
                i. Se ve que Grupo de Seguridad Esta Creado
                ii. Se accede y se ve que puerto 9092 es accesible desde cualquier lugar Origen: 0.0.0.0/0

2) Creando Security Group para cliente
    a) EC2
    b) Redes y Seguridad – SECURITY GROUP
        1) Crear nuevo Security Group
            a) Nombre kafka-client
            b) Descripción: Texto descriptivo
            c) Se mantiene VPC que muestra
            d) Reglas de Entrada
                i. Agregar Regla de Entrada
                ii. Tipo: SSH, Abre Puerto 22
                iii. Origen: Desde cualquier lugar
                iv. Se genera Security Group
            e) No se Modifica Regla de Salida
            f) Se vuelve a Grupo de Seguridad
                i. Se ve que el Grupo de Seguridad para cliente Esta Creado

3) Creando Cluster de Kafka en MSK (Es Pagado)
    a) Ingresar a Amazon MSK
    b) Se selecciona Create Cluster with custom settings
        1) Cluster name : cluster-finytec
        2) Apache Kafka Version: Seleccionar la recomendada.
        3) Configuration: Use MSK default configuration
        4) Seleccionar VPC: Se puede crear, pero se usa la vpc por defecto
        5) Number of Availabitility Zones : ***2***
        6) Availability Zone: use-east-1a
        7) Subnet: Aceptar la que ofrece
        8) Second Availabilty Zone: us-east-1b
        9) Subnet: Seleccionar la ofrecida
        10) Brokers
            a) Broker Instance Type: permite seleccionar el tamaño (**small**, large,…)
            b) Number of broker per Availability Zone: 2 (El total de Broker es el doble, en este caso ***4***)
        11) Storage no se modifica
        12) Encryption
            a) No enable encryption
            b) Both TLS encrypted and plaintext traffic allowed
        13) Monitoring mediante **CLOUDWATCH**
            a) Basic Monitoring
        14) Advanced SETTINGS
            a) Customize settings
                i. Se ven los dos security group definidos (Kafka y Client)

ii. Se selecciona el de Kafka-cluster

15) FINALMENTE create cluster

4) Revisando Creación de Cluster
   a) Amazon MSK → Clusters → <Clusters Name>
   b) Recurso de AWS – Cluster ARN
   c) View cliente information
      1) Se ve Bootstarp Server (Tiene las URL para la conexión y que se deben actualizar en el Código Spring)
         a) TLS
         b) Plaintest
      2) Se Ve ZooKeeper connect (Sale Gratuito)
   d) Resumen de los Brokers (2 por Zone total 4 brokers) (En documentación de kafka se ve cuantos brokers por Instance, ejemplo, t3.small → 300 particiones por brokers.
   e) Monitoring, se ve las métricas del broker

5) Creando una máquina EC2 para conectarse a Kafka-cluster.
   a) 2 Brokers no tienen acceso externos, se deben acceder desde una maquina en la red de AWS
   b) EC2
      1) Instancias en ejecución
      2) Launch Instancia
      3) Seleccionar Amazon Linux gratis)
         a) t2.micro
         b) Review lunch
         c) Se accede a parte de los Security Groups
         d) En vez de crear uno nuevo se selecciona uno existente
            i. Se selecciona la de Kafka-client no la de kafka-cluster
            ii. Lunch
               1) se debe seleccionar llave existente (CREDENCIALES PARA CONECTARSE A LA MAQUINA EC2
               2) Seleccionar create a new key pair
               3) Asignar nombre de la key
               4) Download Key Pair
                  1) Se debe guardar en forma segura, es un archivo .pem
               5) Finalmente Launch Instance
      4) Verificar ejecución de la instancia
         a) EC2
            i. pasa desde pendiente hasta en running

6) Conexión a la maquina EC2 desde local y configuraciones
   a) cambiar permisos a archivo .pem,
      1) chmod 400 <file>.pem
   b) En terminal local donde se encuentra <file>.pem
      1) ssh -i <file>.pem ec2-user@<servidor>
         a) El servidor esta en la instancia de EC2 en public DNS(iov4) <enter>
         b) Se obtiene la conexión a la maquina EC2
      2) Siguiente, actualizar máquina
         a) sudo yum update
      3) Instalar java
         a) sudo yum install java-<version>

7) Copiar archivo jar a EC2
   a) scp -i /ruta/a/tu/clave.pem /ruta/al/archivo.jar ec2-user@<dirección-ip-ec2>:/ruta/de/destino

8) Habilitar Kafka en EC2 AWS
   a) ir a kafka, acceder a Download
   b) copiar dirección HTTPS
   c) wget dirección HTTPS
   d) Se descarga kafka en formato tgz

e) se descomprime descarga
    1) tar -xvf <archivo descargado>
f) se ingresa a la carpeta descomprimida
g) Se ejecutan los comandos siguientes: (similares a los de local, cambia el localhost:9200 por dns plaintext)
    1) Listar topics con plaintext de MSK de AWS (DISPONIBLE EN INFORMACIÓN DE MSK AWS)
        a) bin/kafka-topics.sh .list –bootstrap-server <DNS de plaintext)
        b) Crear topics con --replication-factor 4
        c) Describe
        d) iniciar produces en bootstrap de AWS plaintext
        e) iniciar consumer en bootstrap de AWS plaintext
        f) Se producen y reciben los mensaje

9) Configurando ejemplo práctico.
    a) Se reemplaza en BOOTSTRAP_SERVER_CONFIG localhost:9092 por los BOOTSTRAP_SERVER DE AWS.

10) Creando Cluster de ElasticSearch en AWS
    a) Buscar Amazon ElasticSearch Service en Amazon
        1) Crear un nuevo dominio
        2) Elegir implementación de Desarrollo y pruebas con la última versión de ES
        3) Nombre del Dominio:
        4) Tipo de Instancia predeterminada
        5) Numero de Nodos : 1
        6) Tipo de almacenamiento de nodos de datos: EBS (Elastic block storage)
        7) Tipo de volumen de EBS : por defecto
        8) Tamaño de almacenamiento de EBS por nodo: 10
        9) Siguiente
        10) Acceso mediante VPC (Recomendado) no se elige
        11) Se elige Acceso público
        12) Crear Usuario Maestro:
            a) usuario: devs4j (ejemplo)
            b) Contraseña:
            c) Confirmar contraseña:
        13) NO SE USA COGNITO
        14) Política de acceso: Permitir el acceso libre al dominio
        15) SIGUIENTE
        16) Permite revisar cluster de ES
        17) Confirmar
    b) Prueba de Cluster ES Usando Postman
        1) URL : Punto de enlace de Cluster ES.
            a) En postman se debe usar Basic Auth (Claves de Usuario Maestro)
            b) Ejecutar y responde
                i. Se aplican los comandos similares al de Local
                    1) *_cat/*health?v
                    2) _cat/nodes
        2) URL KIBANA (se usa las claves de Usuario Maestro) se usa en Browser
    c) Indexando documentos con Rest
        1) En postman PUT /devs4j-transactions (CREA EL INDICE)
        2) En postman GET /devs4j-transactions (DEVUELVE EL INDICE CREADO)
        3) En postman POST *devs4j-transactions/transactions*/1 (POSTEA DOCUMENTOS)
        4) En postman GET *devs4j-transactions/transactions*/1 (OBTIENE DOCUMENTOS CREADO)

11) Configurar ES en aplicación
    a) Se debe considerar la autenticacion en el código
        1) Se usa Clase Usuario Maestro
        2) Se usa URL de KIBANA en RestHighLevelClient
            a) No se pone https de la URL de KIBANA y se usa puerto 443, https

3) LA PRUEBA EN LOCAL ES DISTINTA A LA DE AWS, se modifica el config de ES y se inyecta data desde el main

12) SUBIR APP.JAR A AWS con .pem y comando scp a la EC2
13) Ejecutar con ssh i .pem ec2-user@<>
14) La aplicación se ejecuta con java -jar en consola de AWS
15) Crear DashBoard en KIBANA

FINALMENTE LIMPIAR AWS

EC2
MSK
ESS

NOTA: devs4j criptomonedas REVISAR