La pyme Digital Market ha decidido llevar su negocio a la nube, por lo cual ha tomado la decisión de trabajar en la plataforma AWS por lo cual su primer trabajo será llevar la estructura web a esta plataforma, donde deberá tener una arquitectura escalable y lo más autónoma posible. Para ello crear lo siguiente:

-Una arquitectura en una red VPC llamada RED-PRINCIPAL-WEB con el Bloque CIDR IPv4 172.10.0.0/16 la cual necesitamos que tenga dos subredes en el rango de ip 172.10.2.0/24 - 172.10.3.0/24 (Personalizar bloques CIDR) Como el ejercicio no pide subredes privadas, le asignamos 0.

Para poder dar una capa de seguridad a la arquitectura se necesita tener un grupo de seguridad de nombre : Grupo-Seguridad-Web el cual permita las conexiones desde el servicio ssh (puerto 22. Como no especifica quienes son los que pueden ingresar, seleccionar 0.0.0.0/0.

Una vez creado el grupo de seguridad, se crean las instancias.

Esta arquitectura deberá contar con 3 servidores con las características t2.micro con AMI Amazon Linux y su almacenamiento por defecto, estos equipos deben tener los nombres Servidor1, Servidor2 y Servidor3 (Escribir nombre de la instancia “servidor” sin el número. Después, en la pestaña resumen le colocamos 3. Editamos config de red y la ponemos en la vpc creada, y una subred disponible 1ª. En esa pestaña aparece el nunero de ip por si llegaran a pedir una específica para la subred. IMPORTANTE: cada vez que seleccionamos una vpc que no sea la predeterminada la pestaña “asignar automáticamente la ip pública” se desactiva, por lo que hay q procurar mantenerla habilitada). Seleccionamos el grupo de seguridad existente. Finalmente en detalles avanzados vamos al final y escribimos el siguiente script para automatizar la actualización de las tres máquinas virtuales:

#!/bin/bash

yum update -y

yum upgrade -y

Este script se ejecutará como sudo en las 3 instancias. LANZAMOS INSTANCIAS.

Voy a instancias y edito el nombre. Habrá 3 instancias con el nombre servidor. Agregarle el numero a cada una.

Al finalizar esto tendrá que comprobar la conectividad entre todas las máquinas virtuales. Es decir que la instancia 1,2 y 3 puedan verse entre sí. Para eso voy a necesitar las ip privada de cada máquina, que se encuentra al darle click a cada instancia (Direcciones ipv4 privadas.) Le damos conectar y luego conectar. Abrimos 3 pestañas ordenadas.

Para comprobar si puedo llegar de una máquina a otra debiera hacer un “ping ip privada de la máquina a la qu quiero llegar” ( ping 172.10.3.122). Sin embargo, eso no se ejecuta correctamente PORQUE EL PING NECESITA EL PROTOCOLO ICMP EN EL GRUPO DE SEGURIDAD ANTES CREADO. Entonces, en cualquier instancia, voy a detalles y luego a seguridad. Una vez ahí voy a ver el grupo de seguridad y darle click y luego en el panel “Editar reglas de entrada”. Agregar regla Todos los icmp ipv4, desde cualquier lugar 0.0.0.0/0. Guardamos regla.

Después hacemos ping entre las máquinas con sus ips privadas y deberían poder conectarse entre ellas.

Como primera tarea tendrá que aumentar el tamaño del disco en la máquina Servidor1 de 8GB a 15GB. Click en servidor1 y en detalles ir a almacenamiento. Ahí puedo saber que disco tiene asociado (id del volumen). Damos click ahí. Click nuevamente al volumen y arriba a la derecha Modificar. Le damos tamaño GB 15 y luego click abajo en Modificar. No se ve reflejado automáticamente en en panel de ec2. Hay que esperar que se actualice.

Además, usted como especialista en arquitectura tendrá que disponer de un almacenamiento en común para todas las máquinas, utilizando EFS, el que debe ser de nombre Almacenamiento\_Principal donde como requisito todas las instancias deberán tener la posibilidad de conectarse a este disco (dentro de la misma vpc), por lo cual tendrá que asociarlo a cada una de ellas y dejarlo configurado para que cada vez que la máquina se encienda este disco se encuentre asociado. Vamos a EFS y creamos sistema de archivos con el nombre señalado. Creamos el almacenamiento. Abrimos el efs creado -> red-> asociamos el grupo de seguridad antes creado en la pestaña Administrar. Guardamos. Luego, para asociar una EFS con los servidores, abrimos el EFS (click en el nombre) y Asociar. Nos va a dar un código sudo mount el cual debemos copiar. Ahora abrimos el servidor1 y creamos el directorio:

sudo mkdir /mnt/servidorWeb

Si no na error quiere decir que la carpeta fue creada.

Luego pegamos el código sudo mount. Este código terminará en com:/ efs. Nosotros demos reemplazarlo para que quede así: com:/ /mnt/servidorWeb ya que ese es el directorio donde debemos montar el efs.

Comprobamos con un df -h y en la última lpinea nos dice que está o no montado.  
REPETIMOS ESTO EN LAS DOS INSTANCIAS RESTANTES. (sudo mkdir /mnt/servidorWeb y luego pegar el comando sudo mount reemplazando el directorio al final)

Este disco tendrá que estar dentro de la misma red, por lo cual tendrá que estar asociado a la misma vpc.

Ahora, para satisfacer la condición “y dejarlo configurado para que cada vez que la máquina se encienda este disco se encuentre asociado.” que pide el encabezado, debemos editar un archivo que está dentro de /etc/fstab

Este archivo debe ser editado en las tres instancias. El fstab lleva una estructura para poder construirlo, donde le tenemos que pasar el efs, luego le damos la ruta de montaje y finalmente parámetros. Para eso debemos estar viendo los detalles del efs asociado

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Entonces:

Sudo nano /etc/fstab y ejecutamos.

En la tercera línea (inmediatamente debajo de las dos de arriba) colocamos el disco (fs-….hasta .com:/), luego donde se va a montar y finalmente los parámetros(Todo separado por espacios).

fs-……com:/ /mnt/servidorWeb nfs4

ctrl + o para guardar

Enter guardo

Ctrl +x me salgo

Compruebo con sudo mount -a

Si no da error quiere decir que está bien.

Recapitulando, al editar el fstab estamos haciendo que cada vez que la máquina se encienda monete de manera automática el efs. REPETIMOS EN LAS OTRAS DOS INSTANCIAS. (sudo nano /etc/fstab y escribimos en tercera línea)

El disco tendrá que estar montado en el directorio: /mnt/servidorWeb.

Dentro de las máquinas debes crear usuarios para poder administrarlas:

Servidor1 : admin1

Servidor2 : admin2

servidor3 : admin3

Donde todos los usuarios deberán pertenecer al grupo adminweb.

Para eso voy al servidor1 y creo primero el grupo

sudo groupadd adminweb.

De esa manera, luego puedo decirle “crea el admin1 y asócialo al grupo adminweb.

sudo useradd admin1 -g adminweb

comprobamos con cat /etc/group y el grupo adminweb tendrá un id 1001

Ahora, si mor los usuarios con cat /etc/passwd

Debería mostrar admin1: grupo 1001

Ahí queda demostrado que se cumplió el paso.

REPETIR EN LAS OTRAS DOS INSTANCIAS PERO USANDO admin2 y admin 3 SEGÚN CORRESPONDA. Desde sudo groupadd adminweb en adelante

Dentro de la ruta /mnt/servidorWeb cree el archivo pagina.txt el cual debe tener permisos 640.

\*Recordar que /mnt/servidorweb es el punto de montaje del efs. Las tres máquinas tienen el efs ahí. Eso quiere decir que si yo creo el archivo pagina.txt en una de las máquinas, debiera poder verlo en las otras dos de forma simultánea. Por eso el archivo se crea en una sola máquina.

Entonces, en la máquina1

sudo touch /mnt/servidorweb/pagina.txt

Ahora, para comprobar, entro al servidor2 y hago

cd /mnt (ejecuto)

cd servidorweb/ (ejecuto)

ls (y debería aparecer pagina.txt)

Para dar el permiso 640 vamos a cualquier instancia

sudo chmod 640 /mnt/servidorweb/pagina.txt

6= propietario =lectura y escritura 4+2

4= grupo=solamente lectura

0= otros=ninfun permiso

Recordar que :

r=4, w=2, x=1

Compruebo con

ls -l /mnt/servidorweb/pagina.txt (con eso arroja los permisos)

Como método de redundancia cree un grupo de autoscaling en el cual deberá agregarse a los tres servidores y esto debe pertenecer a la misma red este tendrá que estar asociado al balanceador de carga creado anteriormente. Donde tendrá que establecer política de escalado automático basado en la utilización de CPU de modo que cuando el consumo de la CPU supere el 80%(política de escalamiento) crezca de manera automática a un máximo de 6 con un deseado de 3.

Para este último paso vamos al panel principal de la izquierda de la pantalla y damos click en autoscaling grupo de autoscaling y creamos u grupo y le damos un nombre.

Recordar que el autoscaling es lo que permite que nuestras instancias o arquitecturas crezcan bajo una necesidad. Si lo vemos como diagrama, el grupo de autoscaling tiene 3 instancias adentro y estas aumentan a 6 según las condiciones. Cuando el requerimiento disminuye, se borran las instancias. Eso es mínimo deseado 3 y máximo 6. Cuando la cpu de las máquinas llegue al 80% durante 5 min, entonces crece. Para eso se crea una PLANTILLA DE LANZAMIENTO, que diga que cada instancia que se va a crear sea una amazonlinux, t2.micro, 8gb, asociado al efs, dentro de la vpc y asociado al grupo de seguridad.

Damos click en “crear una configuración de lanzamiento”

Damos nombre y descripción.

Luego, en imágen de aplicación-> inicio rápido-> amazonlinux

Luego tipo de instancia -> t2.micro

Par de claves-> debe ser un par de claves existente porque la máquina se va a crear de manera automática.

Luego en configuración de red dedemos seleccionarle el grupo de seguridad creado, porque éste lleva consigo la VPC. Por ese motivo, en subred, no incluimos una subred, ya que el grupo de seguridad indica en qué VPC va a funcionar.

Luego detalles avanzados ->datos de usuario (lo más abajo) configuramos nyestra plantilla de lanzamiento de manera tal que cada vez que se cree una máquina, ésta se asocie al efs de manera automática. Para eso:

#!/bin/bash

yum update -y

yum upgrade -y

mkdir /mnt/servidorweb

(pasarle el efs, copiando toda la dirección que sale en la imagen1, pero reemplazando al final .com:/ /mnt/servidorWeb)

Damos a crear plantilla de lanzamiento.

Como ya se creó la plantilla de lanzamiento, voy a la pestaña donde estaba creando el autoscaling (ec2->panel izquierda->autoscaling) Le damos un nombre al grupo de autoscaling y seleccionamos la plantilla de lanzamiento recién creada y click en siguiente.

Luego en red fijarse en seleccionar la vpc creada y en subred seleccionamos ambas. Luego siguiente.

Balanceador de carga:

Recordemos que el balanceador de carga es la herramienta que va a permitir dividir de manera equilibrada las peticiones que se están creando entre las instancias del grupo autoscaling. El ejercicio no pide balanceador de carga, pero lo vamos a crear igual y que pasa?

Entonces en tipo de balanceador -> aplicación load balancer.

Esquema del balanceador -> internet facing.

Debe poder ver la vpc y ambas subredes.

Protocolo ->crear grupo de destino y coloca nombre automáticamente.

Bajamos a “Comprobar Estado”

Cada 300 segundos (5min)

Siguiente

“Tamaño del grupo”

Capacidad deseada: 3

Capacidad deseada mínima: no especifica así que lo dejamos en 3

Capacidad deseada máxima: 6

“Escalamiento automático”

Marcar política de escalado de seguimiento de destino”

Valor destino: 80

Política mantenimiento de instancia: por defecto

SIGUIENTE

SIGUIENTE

Etiquetas:

Clave: Nombre

Valor: ServidorEscalado (le estamos diciendo que cada máquina que cree se llame servidorEscalado

SIGUIENTE

Finalmente damos a CREAR GRUPO AUTOSCALING

\*Importante: el grupo autoscaling va a comenzar a crear instancias de manera inmediata, porque “piensa” que las necesitamos ahora. En este caso crea 3(deseado)

El ejercvicio pide que las máquinas creadas al principio (servidor1,2 y 3) también estén asociadas al grupo autoscaling. Para eso la seleccionamos->acciones->configuracions de instancia->asociar a grupo autoscaling. Esto se hace para que el autoscaling empiece a administrar también estas instancias, de manera que pueda eliminarlas cuando lo necesite.

Para finalizar cree un script que permita guardar un respaldo del archivo pagina.txt que se encuentra dentro de /mont/servidorWeb

Para ello nos conectamos al servidor1

ls

touch respaldo.sh

ls

Hasta ahí tenemos el script creado, ahora hay q editarlo:

nano respaldo.sh (ejecutamos)

#!/bin/bash

cp -r /mnt/servidorWeb /home/ec2-user/respaldo

ctrl + o (guardo)

ctrl + x (me salgo)

ese script copia lo que hay en /mnt/servidorWeb en la dirección /home/ec2-user/respaldo

Como aun no creamos la carpeta respaldo, la creamos

mkdir respaldo

ls

Muestra respaldo (como carpeta en letras celestes) y respaldo.sh que sería el script

Ahora, los script, para ejecutarse necesitan permiso de ejecución, por lo cual hacemos:

sudo chmod +x respaldo.sh (ejecutamos)

si ahora hacemos un ls el respaldo.sh cambia de color a verde, eso quiere decir que tiene permiso de ejecución.

Para poder ejecutarlo hacemos:

sudo ./respaldo.sh (ejecutamos y hacemos ls)

Ahora entramos al respaldo:

cd respaldo/

ls

Debería traer la carpeta servidorWeb completa en letra celeste.

Entonces si miro el servidorWeb (cd servidorWeb/ luego ls) debería mostrarme el archivo pagina.txt

REPETIR EN LAS OTRAS 2 INSTANCIAS

\*chmod = cambiar permisos de acceso de un archivo o un directorio (777 o 640)

Chown = cambiar el propietario de un archivo

Chgrp = cambiar grupo de usuarios de un archivo o directorio