

Laboratorio de desafíos: crear un sitio web dinámico para la cafetería

En este laboratorio implementaremos una app web en una instancia EC2 en una región, una vez funcione, crearemos una AMI de esta y la implementaremos en otra región.

Empezamos con esto ya creado:



Tarea 1: analizar la instancia de EC2 existente

<input checked="" type="checkbox"/>	Name ↗	ID de la instancia	Estado de la instancia	Tipo de instancia	Comp
<input checked="" type="checkbox"/>	aws-cloud9-CafeWebServer-55c20e74b26e4f889e...	i-0832e81edaf873a8a	En ejecución	t2.micro	2/2

i-0832e81edaf873a8a (aws-cloud9-CafeWebServer-55c20e74b26e4f889eb0e595455d8003)		🔍 🔍
Detalles Estado y alarmas Novedad Monitoreo Seguridad Redes Almacenamiento Etiquetas		
▼ Resumen de instancia Información		
ID de la instancia 🔍 i-0832e81edaf873a8a (aws-cloud9-CafeWebServer-55c20e74b26e4f889eb0e595455d8003)	Dirección IPv4 pública 🔍 3.215.117.137 dirección abierta 🔗	Direcciones IPv4 privadas 🔍 10.0.0.224
Dirección IPv6 -	Estado de la instancia 🟢 En ejecución	DNS de IPv4 pública 🔍 ec2-3-215-117-137.compute-1.amazonaws.com dirección abierta 🔗

Vemos que la instancia tiene esa IP publica y que está asociada a un cloud9.

Preguntas sobre la instancia:

¿La instancia está en una subred pública?

i-0832e81edaf873a8a (aws-cloud9-CafeWebServer-55c20e74b26e4f889eb0e595455d8003)

Detalles Estado y alarmas **Novedad** Monitoreo Seguridad **Redes** Almacenamiento Etiquetas

▼ Detalles de redes Información

Dirección IPv4 pública 3.215.117.137 dirección abierta	Direcciones IPv4 privadas 10.0.0.224	ID de VPC vpc-0ba5fd6a62eac84ed (Lab VPC)
DNS de IPv4 pública ec2-3-215-117-137.compute-1.amazonaws.com dirección abierta	Nombre DNS de IP privada (solo IPv4) ip-10-0-0-224.ec2.internal	
ID de subred subnet-0c6fbbf7d09ea8d4a (Public Subnet)	Direcciones IPv6 -	Direcciones IPv4 privadas secundarias -
Zona de disponibilidad us-east-1a	Direcciones IP del operador (efímeras) -	ID de Outpost -

Si, si lo está.

¿La instancia de EC2 tiene asignada una dirección IP pública IPv4?

Si, esta 3.215.117.137

¿Qué números de puerto TCP de entrada están abiertos para esta instancia?

Para ver esto, nos vamos a la pestaña de Seguridad y reglas de entrada:

▼ Reglas de entrada

Filtrar reglas				
Nombre	ID de la regla del grupo d...	Intervalo de pu...	Protocolo	Origen
-	sgr-0e11a8a9c98d5ef93	22	TCP	35.172.155.96/27
-	sgr-09ae2fb1fcb69d3f5	22	TCP	35.172.155.192/27

Vemos que tiene abierto el puerto 22, y solo permitido a un rango específico.

¿La instancia de EC2 tiene un rol AWS Identity and Access Management (IAM) asociado?

Para ver eso nos vamos a los detalles de la instancia y buscamos Rol de IAM:

Rol de IAM

IMDSv2
Optional

Vemos que no tiene ninguno asociado.

Tarea 2: conectarse al IDE en la instancia de EC2

Para ello usaremos cloud9. Elegimos el entorno de CafeWebServer y le damos a abrir:

Entornos (1)						
<div>Eliminar Ver detalles Abrir en Cloud9 Crear entorno</div> <div>Mis entornos</div>						
	Nombre	Cloud9 IDE	Tipo de entorno	Conexión	Permiso	ARN del propietario
<input type="radio"/>	CafeWebServer	Abrir	Instancia de EC2	Secure Shell (SSH)	Propietario	arn:aws:sts::169520914979:assumed-role/voclabs/user3191730=Ruiz_Ruiz,_Nicol_s

Esto nos abrirá una “Maquina Virtual” en la que alojaremos nuestra app en un futuro.

Tarea 3: analizar el entorno de la pila LAMP y confirmar que el servidor web es accesible

Podemos empezar viendo la versión del sistema operativo. Para ellos tenemos que tener en cuenta que se trata de una máquina Linux, así que los comandos serán muy parecidos:

```
bash - "ip-10-0-0-224.ec2" x Immediate x +  
voclabs:~/environment $ cat /proc/version  
Linux version 5.10.215-203.850.amzn2.x86_64 (mockbuild@ip-10-0-33-79) (gcc10-gcc (GCC) 10.5.0 20230707 (Red Hat 10.5.0-1), GNU ld version 2.35.2-9.amzn2.0.1) #1 SMP Tue Apr 23 20:32:19 UTC 2024  
voclabs:~/environment $
```

Podemos ver(o no) que se trata de una versión de Amazon Linux parecida a Red Hat. Ahora vamos ver si los servicios necesarios para la app están correctamente instalados u operativos:

Servicio http:

```
voclabs:~/environment $ sudo httpd -v  
Server version: Apache/2.4.59 ()  
Server built: Apr 22 2024 13:07:26  
voclabs:~/environment $ service httpd status  
Redirecting to /bin/systemctl status httpd.service  
• httpd.service - The Apache HTTP Server  
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)  
Drop-In: /usr/lib/systemd/system/httpd.service.d  
└─php-fpm.conf  
Active: inactive (dead)  
Docs: man:httpd.service(8)
```

Servicio sql(mariadb):

```
voclabs:~/environment $ mariadb --version  
mariadb Ver 15.1 Distrib 10.5.23-MariaDB, for Linux (x86_64) using Editline wrapper  
voclabs:~/environment $ service mariadb status  
Redirecting to /bin/systemctl status mariadb.service  
• mariadb.service - MariaDB 10.5 database server  
Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)  
Active: inactive (dead)  
Docs: man:mariadb(8)  
https://mariadb.com/kb/en/library/systemd/
```

php para finalizar:

```
voclabs:~/environment $ php --version
PHP 8.2.9 (cli) (built: Aug 24 2023 20:33:12) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.9, Copyright (c) Zend Technologies
with Xdebug v3.2.2, Copyright (c) 2002-2023, by Derick Rethans
```

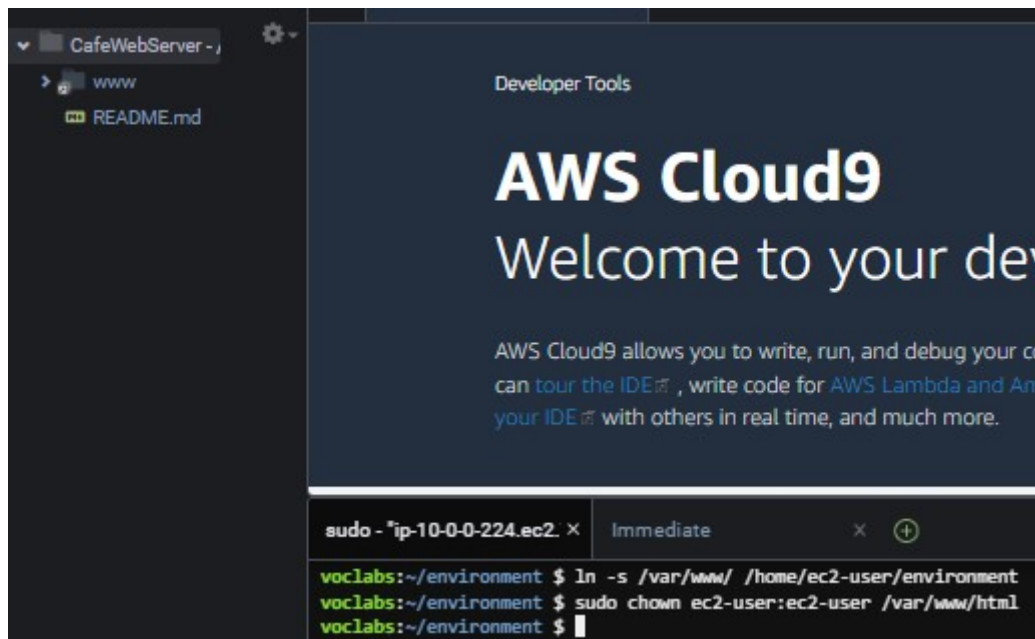
Vemos que todo está instalado correctamente, mas no está habilitado.

Vamos a activar los servicios:

```
voclabs:~/environment $ sudo chkconfig httpd on
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
voclabs:~/environment $ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
voclabs:~/environment $ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
   Active: active (running) since Mon 2024-05-20 14:59:01 UTC; 1s ago
     Docs: man:httpd.service(8)
   Main PID: 16580 (httpd)
   Status: "Processing requests..."
     Tasks: 47
    Memory: 12.5M
```

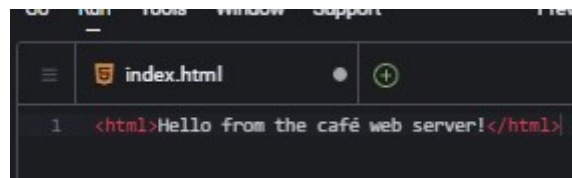
```
voclabs:~/environment $ sudo chkconfig mariadb on
Note: Forwarding request to 'systemctl enable mariadb.service'.
Created symlink from /etc/systemd/system/mysql.service to /usr/lib/systemd/system/mariadb.service.
Created symlink from /etc/systemd/system/mysqld.service to /usr/lib/systemd/system/mariadb.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
voclabs:~/environment $ sudo service mariadb start
Redirecting to /bin/systemctl start mariadb.service
voclabs:~/environment $ sudo service mariadb status
Redirecting to /bin/systemctl status mariadb.service
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-05-20 15:00:08 UTC; 3s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 17377 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
   Process: 17157 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
   Process: 17133 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
```

Una vez todo activado, vamos a crear un enlace débil a nuestro entorno de trabajo para poder trabajar más cómodamente.

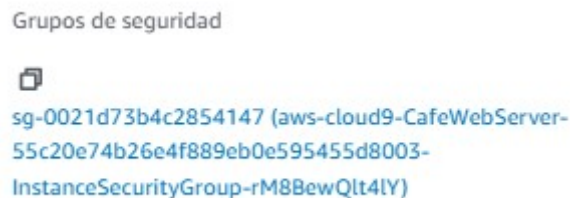


Arriba a la izquierda podemos apreciar que en nuestro home aparece un enlace simbólico a la carpeta de trabajo.

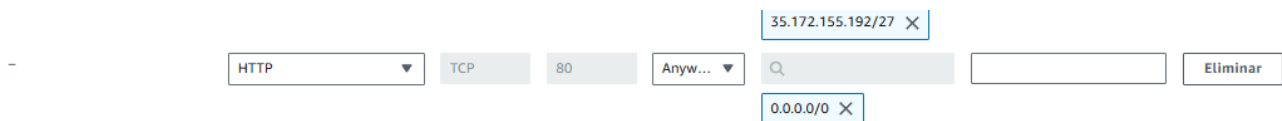
Ahora que está todo preparado, vamos a empezar a programar nuestra app, nos dirigimos a la siguiente ruta: “www/html”, aquí creamos un archivo .html con el contenido de abajo:



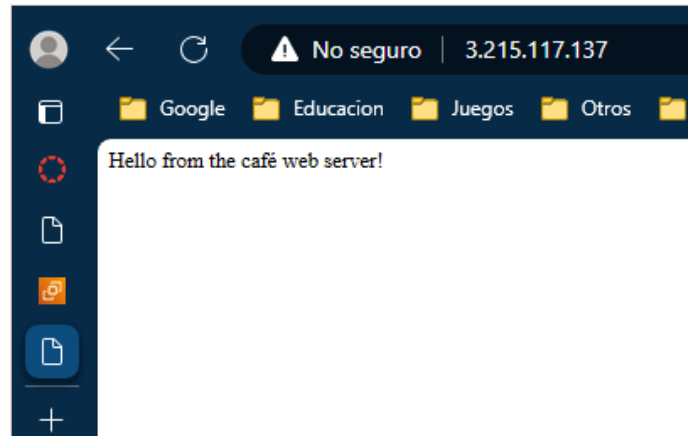
Guardamos los cambios. Ahora solo queda hacer que la app sea visible desde internet, para ello nos vamos al servicio EC2, elegimos nuestra instancia y nos vamos a la pestaña de Seguridad y damos click en el grupo de seguridad:



Aquí debemos añadir una regla de entrada, tiene que verse tal que así:



Ahora si ponemos la IP de la instancia en nuestro navegador, veremos lo siguiente:



Nuevo requisito empresarial: instalar una aplicación de sitio web dinámico en la instancia de EC2 (desafío n.º 2)

Tarea 4: instalar la aplicación de la cafetería

Una vez habilitados los servicios, accedido a la app mediante internet y visto que funciona, vamos a crear una app en condiciones, para ellos nos descargamos los siguientes repositorios(alojados en un bucket S3):

Los comandos son los mismos, coger de internet este archivo x y descomprimirlo.

```
cd ~/environment
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/setup.tar.gz
```

```
tar -zxvf setup.tar.gz
```

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/db.tar.gz
```

```
tar -zxvf db.tar.gz
```

```
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/mod4-challenge/cafe.tar.gz
```

```
tar -zxvf cafe.tar.gz
```

Vamos a copiar todos los archivos al directorio de trabajo:

```
mv cafe /var/www/html/
```

Abrimos `html/caf /index.php` y vemos como est  compuesto el c digo html:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Caf !</title>
6     <link rel="stylesheet" href="css/styles.css">
7 </head>
8
9 <body class="bodyStyle">
10
11     <div id="header" class="mainHeader">
12         <hr>
13         <div class="center">Caf </div>
14     </div>
15     <br>
16     <?php
17         // Get the application environment parameters from the Parameter Store.
18         include ('getAppParameters.php');
19
20         // Display the server metadata information if the showServerInfo parameter is true.
21         include('serverInfo.php');
```

Vemos que la l nea 18 hace referencia a un archivo, vamos a abrirlo tambi n:

```
$result = $ssm_client->GetParametersByPath(['Path' => '/caf ']);

$showServerInfo = "";
$timeZone = "";
$currency = "";
$db_url = "";
$db_name = "";
$db_user = "";
$db_password = "";

foreach($result['Parameters'] as $p) {
    if ($p['Name'] == '/caf /showServerInfo') $showServerInfo = $p['Value'];
    if ($p['Name'] == '/caf /timeZone') $timeZone = $p['Value'];
    if ($p['Name'] == '/caf /currency') $currency = $p['Value'];
    if ($p['Name'] == '/caf /dbUrl') $db_url = $p['Value'];
    if ($p['Name'] == '/caf /dbName') $db_name = $p['Value'];
    if ($p['Name'] == '/caf /dbUser') $db_user = $p['Value'];
    if ($p['Name'] == '/caf /dbPassword') $db_password = $p['Value'];
}
```

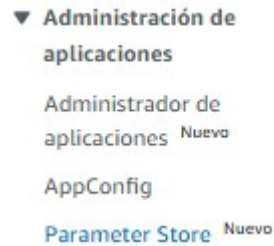
B sicamente, un programa que recopila informaci n de la base de datos.

Vamos a ejecutarlo para que se vea en la app(que ahora pondremos):

```
voclabs:~/environment $ cd setup
voclabs:~/environment/setup $ ./set-app-parameters.sh

Setting the default AWS region...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload    Total   Spent    Left   Speed
100    10  100    10    0    0    738      0  --:--:-- --:--:-- --:--:--    769
Region = us-east-1
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload    Total   Spent    Left   Speed
100    41  100    41    0    0   3099      0  --:--:-- --:--:-- --:--:--   3153
```

Vamos a usar un nuevo servicio, Systems Manager, y a la izquierda elegimos Parameter Store:



Aquí podremos ver valores “importantes” de nuestra base de datos, nos interesa /cafe/dbPassword
Re:Start!9

Volvemos a nuestra terminal y ejecutamos los siguientes scripts:

(Como dan fallo, lo que he hecho es modificar directamente la contraseña del root)

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Re:Start!9';

```
voclabs:~/environment/db $ ./create-db.sh
Create Database script completed.
Please check the create-db.log file to verify successful execution.
```

Si ahora entramos en la base de datos con el usuario root y la contraseña Re:Start!9, podemos ver la base de datos creada:

```
MariaDB [(none)]> use cafe_db;
Reading table information for completion
You can turn off this feature to get a q

Database changed
MariaDB [cafe_db]> show tables;
+-----+
| Tables_in_cafe_db |
+-----+
| order              |
| order_item         |
| product            |
| product_group      |
+-----+
4 rows in set (0.001 sec)
```

Actualicemos ahora la configuración de la zona horaria:

```
voclabs:~/environment/db $ sudo sed -i "2i date.timezone = \"America/New_York\" \" /etc/php.ini
voclabs:~/environment/db $ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
```


Es hora de probar nuestra app, entramos en la ip de la instancia y agregamos un /cafe/:



Café

Problema a resolver, la app funciona de manera errónea, esto es debido a los permisos que tiene, para cambiar esto vamos a otorgarle un rol IAM adecuado. Para ello, elegimos nuestra instancia EC2, le damos a acciones/seguridad/cambiar rol IAM. Aquí elegimos el CafeRol, le damos a guardar y reiniciamos nuestra instancia. Con esto debería ser más que suficiente, pero la página está equivocada(no lo digo yo, lo hemos puesto en común varios compañeros)

Tarea 5: probar la aplicación web

En este paso, debería entrar al menú y hacer un pedido(pero no puedo)

Nuevo requisito empresarial: crear sitios web de desarrollo y producción en diferentes regiones de AWS (desafío n.º 3)

Esto puede ser útil para tener un sitio en el que poder trabajar sin molestar a los clientes, y otro sitio que sirva para el tema de pedidos.

Tarea 6: crear una AMI y lanzar otra instancia de EC2

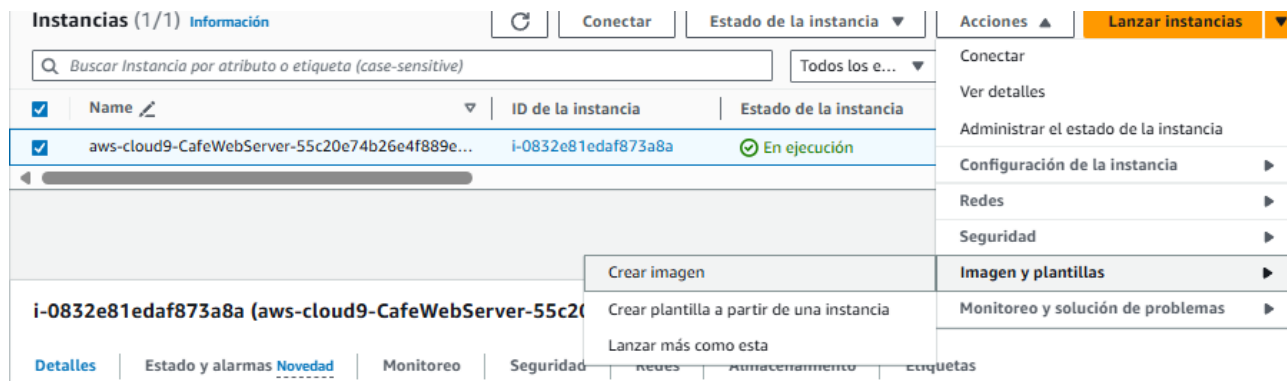
Una AMI es una imagen de una instancia(como una ova). Previamente crearemos un par de claves en nuestra instancia EC2:

```
voclabs:~/environment $ sudo hostname cafeserver
voclabs:~/environment $ ssh-keygen -t rsa -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/id_rsa.
```

Si nos pide algo, lo dejamos en blanco. Ahora vamos a hacer que la clave se pueda usar a través de ssh.

```
voclabs:~/environment $ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
voclabs:~/environment $
```

Ahora si, creamos la AMI, nos vamos a EC2/instancias/actions/images and templates/create images:



Preguntas sobre el las AMI

Al crear una AMI a partir de una instancia, ¿se reiniciará la instancia?

Descripción de la imagen: *opcional*

Descripción de la imagen

255 caracteres como máximo

Sin reiniciar

☐ Habilitar

Vemos que tenemos la opción de que no lo haga, pero por defecto lo hará.

¿De qué manera se pueden modificar las propiedades del volumen raíz cuando se crea una AMI a partir de una instancia?

Tipo de almacenamiento	Dispositivo	Instantánea	Tamaño	Tipo de volumen	IOPS	Rendimiento	Eliminar cuando termine	Cifrado
EBS	/dev/...	Crear una nueva instan...	10	SSD de uso general de ...	3000		<input checked="" type="checkbox"/> Habilitar	<input type="checkbox"/> Habilitar

Solo podemos cambiar el tamaño y el eliminar cuando termine.

¿Se pueden añadir más volúmenes a una AMI creada a partir de una instancia que solo tenga un volumen?

Si

Tipo de almacenamiento	Dispositivo
EBS	/dev/...

Agregar volumen

De vuelta a la consola, rellenamos los siguientes parámetros:

Nombre de la imagen

CafeServer

Máximo de 127 caracteres. No se pu

Le damos a crear imagen. El menú izquierdo elegimos AMI y esperamos a que el estado sea disponible.

Vamos a crear una AMI en otra región de AWS, para ello elegimos la AMI, pulsamos en acciones y copiar AMI, aquí elegimos Oregon y le damos a copiar.

Descripción de la copia de AMI

[Copied ami-0954c8c5c7e9d6673 from us-east-1] CafeServer

Región de destino

Se creará una copia de la AMI original en la región de destino.

EE.UU. Oeste (Oregon)

☐ Copiar etiquetas

Incluye las etiquetas de AMI definidas por el usuario al copiar la AMI

Imágenes de Amazon Machine Image (AMI) (1) Información

De mi propiedad

Buscar AMI por atributo o etiqueta

	Name	Nombre de AMI	ID de AMI	Origen	Propietario	Visib
		CafeServer	ami-028b4...	169520914979/CafeServer	169520914979	Priva

Aquí la vemos en la otra región. Crearemos la instancia de la cafetería desde la AMI copiada. Para ello, lanzamos una instancia y elegimos nuestra AMI:

Mis AMI Inicio rápido

☒ De mi propiedad ☐ Compartido conmigo

Imágenes de máquina de Amazon (AMI)

CafeServer
ami-028b4a5c30ee24c87
2024-05-20T16:26:01.000Z Virtualización: hvm Activado para ENA: true Tipo de dispositivo raíz: e

Descripción

[Copied ami-0954c8c5c7e9d6673 from us-east-1] CafeServer

Arquitectura ID de AMI

x86_64 ami-028b4a5c30ee24c87

Ponemos la configuración requerida dada por la web y creamos la instancia

ec2-34-213-222-239.us-west-2.compute.amazonaws.com (DNS de la nueva instancia)

Ahora tendremos que editar un parámetro en el archivo **set-app-parameters.sh** que encontraremos en el IDE cloud9 de la región primera:

```
10 echo "Setting the default region"
11 az=$(curl http://169.254.169.254/latest/meta-data/availability-zone/)
12 region="us-west-2"
13 export AWS_DEFAULT_REGION=$region
14 echo "Region = " $AWS_DEFAULT_REGION
```

Ponemos esto en la línea 12.

```
17 #
18 publicDNS="ec2-34-213-222-239.us-west-2.compute.amazonaws.com"
19 echo "Public DNS =" $publicDNS
20 echo
```

En la 18 ponemos la DNS de la nueva región.

Le damos a RUN para ver que todo funcione y seguimos.

Tarea 7: verificar la nueva instancia de la cafetería

Ponemos ahora la ip de la instancia de la región de Oregon y vemos si funciona(no funciona por el mismo error de antes)



Café

Este error lo hemos tenido bastantes compañeros, yo mismo lo he buscado en internet la solución del laboratorio, y es exactamente lo que hemos hecho en la tarea 4 al final, cambiar el Rol IAM al del caferol, con eso debería de ser suficiente.