LAB-188

ADMINISTRACIÓN DE RECURSOS MEDIANTE EL ETIQUETADO

Fabricio Cervantes, Sony Etcheverry, Agustín Esteche, Facundo Morales, Benjamin Sabaño, Fernanda Urman, Balter Velazquez

OBJETIVOS

1	Aplicar etiquetas a recursos existentes de AWS
2	Buscar recursos en función de las etiquetas
3	Utilizar la AWS CLI o AWS SDK para PHP para detener y terminar instancias de Amazon EC2 en función de determinados atributos del recurso

INTRODUCCIÓN

En este laboratorio, iniciará una sesión en Command Host y ejecutará algunos comandos para encontrar y cambiar la etiqueta Version (Versión) en todas las instancias de desarrollo. Ejecutará varios ejemplos que muestran cómo puede utilizar la sintaxis de JMESPath admitida por la opción --query de la AWS CLI con el fin de devolver un resultado con un formato completo. A continuación, utilizará un conjunto de scripts previamente provistos para detener y reiniciar todas las instancias pertenecientes al entorno de desarrollo, es decir que tienen la etiqueta development (desarrollo).

TAREA 1: UTILIZAR ETIQUETAS PARA ADMINISTRAR RECURSOS

En esta tarea, iniciará sesión en la instancia Command Host y utilizará la AWS CLI para encontrar un conjunto de recursos por sus etiquetas.

ENCONTRAR INSTANCIAS DE DESARROLLO PARA EL PROYECTO

Ahora que ha iniciado sesión, puede utilizar la AWS CLI para encontrar los recursos de su subred privada que pertenecen al proyecto ERPSystem y que están en el entorno llamado development (desarrollo). También verá cómo utilizar la opción --query de la AWS CLI para producir resultados con un buen formato.

Para encontrar todas las instancias en su cuenta que tienen la etiqueta Project (Proyecto) y un valor de ERPSystem, copie el siguiente comando y ejecútelo en la ventana del terminal de Linux:

aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem"

ENCONTRAR INSTANCIAS DE DESARROLLO PARA EL PROYECTO

Copie el siguiente comando y ejecútelo en la ventana de terminal de Linux para incluir tanto el ID de la instancia como la zona de disponibilidad de cada instancia en el resultado de retorno:

aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem" --query 'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone}'

ENCONTRAR INSTANCIAS DE DESARROLLO PARA EL PROYECTO

Para incluir el valor de la etiqueta Project (Proyecto) en el resultado, copie y ejecute el siguiente comando en el terminal de Linux:

aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem" --query 'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone,Project:Tags[?Key==`Project`] | [0].Value}'

ENCONTRAR INSTANCIAS DE DESARROLLO PARA EL PROYECTO

Copie y ejecute el siguiente comando para incluir también las etiquetas Environment (Entorno) y Version (Versión) en el resultado:

```
aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem" --query
'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone,Project:Tags[?
Key==`Project`] | [0].Value,Environment:Tags[?Key==`Environment`] |
[0].Value,Version:Tags[?Key==`Version`] | [0].Value}'
```

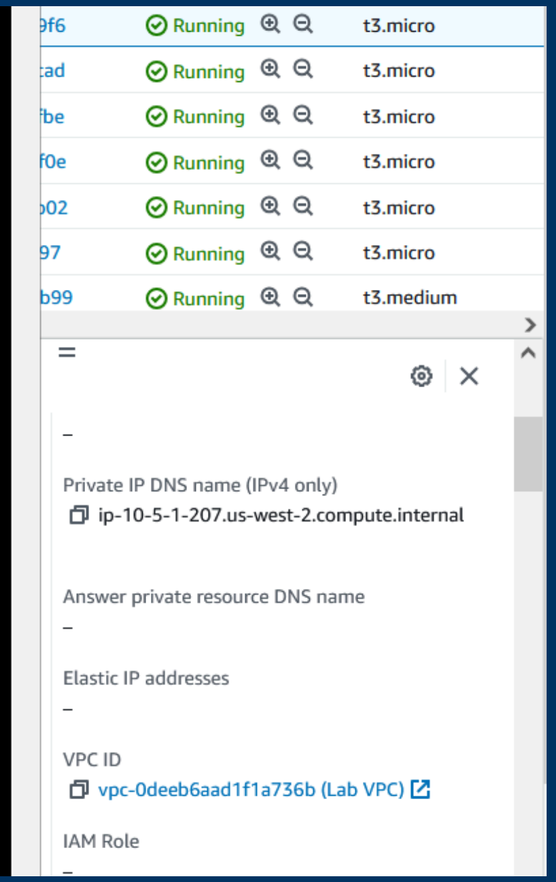
ENCONTRAR INSTANCIAS DE DESARROLLO PARA EL PROYECTO

Finalmente, agregue un segundo filtro de etiquetas para ver solo las instancias asociadas al proyecto denominado ERPSystem que pertenecen al Environment (Entorno) denominado development (desarrollo):

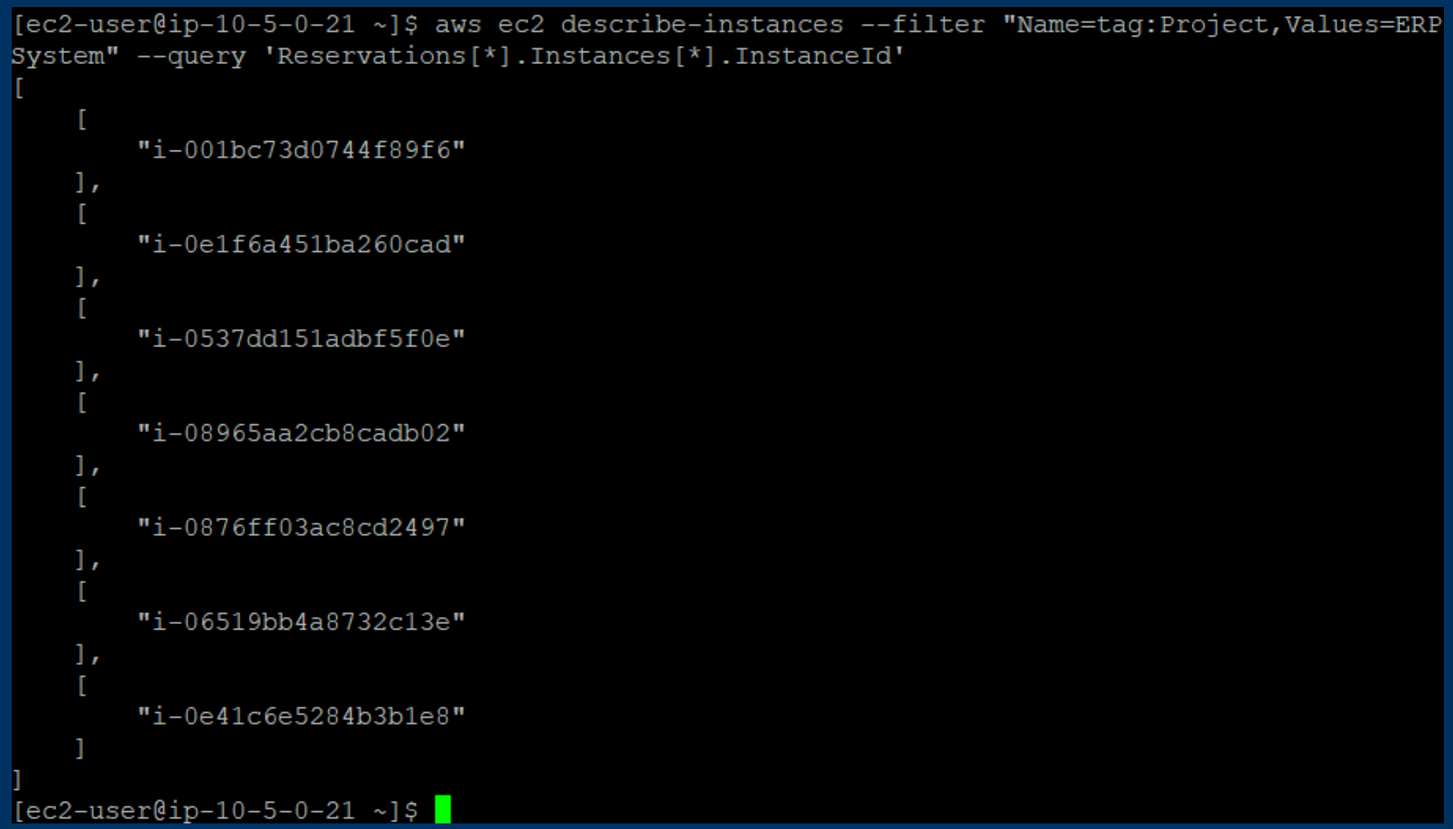
```
aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem"
"Name=tag:Environment,Values=development" --query 'Reservations[*].Instances[*].
{ID:InstanceId,AZ:Placement.AvailabilityZone,Project:Tags[?Key==`Project`] |
[0].Value,Environment:Tags[?Key==`Environment`] | [0].Value,Version:Tags[?Key==`Version`]
| [0].Value}'
```

Desplegamos todas las instancias en la cuentas que posean la etiqueta proyecto y el valor de ERPSystem

```
[ec2-user@ip-10-5-0-21 ~]$ aws ec2 describe-instances --filter "Name=tag:Project
,Values=ERPSystem"
{
    "Reservations": [
        {
            "Instances": [
                {
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "PublicDnsName": "",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "EbsOptimized": false,
                    "LaunchTime": "2024-07-15T21:16:51.000Z",
                    "PrivateIpAddress": "10.5.1.207",
                    "ProductCodes": [],
                    "VpcId": "vpc-0deeb6aad1f1a736b",
                    "CpuOptions": {
                        "CoreCount": 1,
                        "ThreadsPerCore": 2
                    },
                    "StateTransitionReason": "",
                    "InstanceId": "i-001bc73d0744f89f6",
                    "EnaSupport": true,
                    "ImageId": "ami-03789dbbf779d9790",
                    "PrivateDnsName": "ip-10-5-1-207.us-west-2.compute.internal"
,
                    "KeyName": "vockey",
                    "SecurityGroups": [
```



El comando debería mostrar el conjunto completo de parámetros disponibles para las siete instancias etiquetadas como Project=ERPSystem.
El resultado sería muy extenso y gran parte no sería de utilidad en este laboratorio.
En el siguiente paso, utilizará el parámetro --query para limitar los resultados



Utilizando el parámetro --query podemos limitar los resultados del comando anterior a solo el ID de la instancias descubierta:

```
[ec2-user@ip-10-5-0-21 ~]$ aws ec2 describe-instances --filter "Name=tag:Project,Values=ERP
System" --query 'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone}
'
[
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-001bc73d0744f89f6"
        }
    ],
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-0e1f6a451ba260cad"
        }
    ],
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-0537dd151adbf5f0e"
        }
    ],
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-08965aa2cb8cadb02"
        }
    ],
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-0876ff03ac8cd2497"
        }
    ],
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-06519bb4a8732c13e"
        }
    ],
    [
        {
            "AZ": "us-west-2a",
            "ID": "i-0e41c6e5284b3b1e8"
        }
    ]
```

El comando --query utilizado en este ejemplo utiliza la sintaxis de comodín JMESPath para especificar que se deben realizar iteraciones en todas las reservas y en todas las instancias con el fin de devolver la variable InstanceId de cada instancia en los resultados de retorno.

ENCONTRAR INSTANCIAS DE DESARROLLO PARA EL PROYECTO

Utilice el parámetro --query para limitar el resultado del comando anterior a solo el ID de la instancia descubierta:

aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem" --query 'Reservations[*].Instances[*].InstanceId'

Copie el siguiente comando y ejecútelo en la ventana de terminal de Linux para incluir tanto el ID de la instancia como la zona de disponibilidad de cada instancia en el resultado de retorno:

```
[ec2-user@ip-10-5-0-21 ~]$ aws ec2 describe-instances --filter "Name=tag:Project,Values=ERP
System" --query 'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone,
Project:Tags[?Key==`Project`] | [0].Value}'
[
    [
        {
            "Project": "ERPSystem",
            "AZ": "us-west-2a",
            "ID": "i-001bc73d0744f89f6"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "AZ": "us-west-2a",
            "ID": "i-0e1f6a451ba260cad"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "AZ": "us-west-2a",
            "ID": "i-0537dd151adbf5f0e"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "AZ": "us-west-2a",
            "ID": "i-08965aa2cb8cadb02"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "AZ": "us-west-2a",
            "ID": "i-0876ff03ac8cd2497"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "AZ": "us-west-2a",
            "ID": "i-06519bb4a8732c13e"
        }
    ],
    [
        {
```

Se devuelven dos pares nombre-valor por cada resultado.
Este comando se basa en el uso que el comando anterior hace de la sintaxis JMESPath: utiliza llaves para especificar una consulta de múltiples propiedades en cada instancia devuelta:

object.{Alias1:PropertyName1,Alias2:PropertyName2,[…]}

Como se ve aquí, se puede especificar un alias para cada propiedad con el fin de que el formato del resultado sea más abreviado.

Con este resultado, puede ver claramente que su filtro funcionó, y que solo está viendo las instancias que están asociadas con el proyecto ERPSystem. Sin embargo, con base en esta información, es probable que siga sin poder identificar qué instancias se devuelven. En los próximos pasos, verá cómo incluir el valor de las etiquetas personalizadas en el resultado de retorno.

Para incluir el valor de la etiqueta Project (Proyecto) en el resultado, copiamon y ejecutamos el siguiente comando en el terminal de Linux:

aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem" --query 'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone,Project:Tags[?Key==`Project`] | [0].Value}'

```
[ec2-user@ip-10-5-0-21 ~]$ aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPS
ystem" --query 'Reservations[*].Instances[*].{ID:InstanceId,AZ:Placement.AvailabilityZone,Pr
oject:Tags[?Key==`Project`] | [0].Value,Environment:Tags[?Key==`Environment`] | [0].Value,Ve
rsion:Tags[?Key==`Version`] | [0].Value}'
[
    [
        {
            "Project": "ERPSystem",
            "Environment": "staging",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-001bc73d0744f89f6"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "staging",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0e1f6a451ba260cad"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "staging",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0537dd151adbf5f0e"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "production",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-08965aa2cb8cadb02"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "development",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0876ff03ac8cd2497"
```

Copiamos y ejecute el siguiente comando para incluir también las etiquetas Environment (Entorno) y Version (Versión) en el resultado:

```
ystem" "Name=tag:Environment,Values=development" --query 'Reservations[*].Instances[*].{ID:I
nstanceId,AZ:Placement.AvailabilityZone,Project:Tags[?Key==`Project`] | [0].Value,Environmen
t:Tags[?Key==`Environment`] | [0].Value,Version:Tags[?Key==`Version`] | [0].Value}'
[
    [
        {
            "Project": "ERPSystem",
            "Environment": "development",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0876ff03ac8cd2497"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "development",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0e41c6e5284b3b1e8"
        }
```

Finalmente añadimos una segunda etiqueta (Tag) para poder catalogar las instancias pertenecientes al pryecto “ERPSystem” que, a su vez, pertenece al entorno “development”

```
[ec2-user@ip-10-5-0-21 ~]$ aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPS
ystem" "Name=tag:Environment,Values=development" --query 'Reservations[*].Instances[*].{ID:I
nstanceId,AZ:Placement.AvailabilityZone,Project:Tags[?Key==`Project`] | [0].Value,Environmen
t:Tags[?Key==`Environment`] | [0].Value,Version:Tags[?Key==`Version`] | [0].Value}'
[
    [
        {
            "Project": "ERPSystem",
            "Environment": "development",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0876ff03ac8cd2497"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "development",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0e41c6e5284b3b1e8"
        }
    ]
```

CAMBIAR LA ETIQUETA VERSION PARA EL PROCESO DE DESARROLLO

En este procedimiento, cambiará todas las etiquetas Version (Versión) en las instancias marcadas como development (desarrollo) dentro del proyecto ERPSystem.

En Command Host, abra el archivo **/home/ec2-user/change-resource-tags.sh**:

nano change-resource-tags.sh

Cierre el editor nano y ejecute este comando
desde la línea de comandos de Linux:

```
[ec2-user@ip-10-5-0-21 ~]$ ./change-resource-tags.sh
```

Para verificar que el número de versión en estas instancias se haya incrementado y que otras cajas de no desarrollo en el proyecto ERPSystem no se hayan visto afectadas, copie y ejecute el siguiente comando:

```
[ec2-user@ip-10-5-0-21 ~]$ aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPS
ystem" --query 'Reservations[*].Instances[*].{ID:InstanceId, AZ:Placement.AvailabilityZone,
Project:Tags[?Key==`Project`] |[0].Value,Environment:Tags[?Key==`Environment`] | [0].Value,V
ersion:Tags[?Key==`Version`] | [0].Value}'
[
    [
        {
            "Project": "ERPSystem",
            "Environment": "staging",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-001bc73d0744f89f6"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "staging",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0e1f6a451ba260cad"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "staging",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-0537dd151adbf5f0e"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "production",
            "AZ": "us-west-2a",
            "Version": "1.0",
            "ID": "i-08965aa2cb8cadb02"
        }
    ],
    [
        {
            "Project": "ERPSystem",
            "Environment": "development",
            "AZ": "us-west-2a",
            "Version": "1.1",
            "ID": "i-0876ff03ac8cd2497"
```

CAMBIAR LA ETIQUETA VERSION PARA EL PROCESO DE DESARROLLO

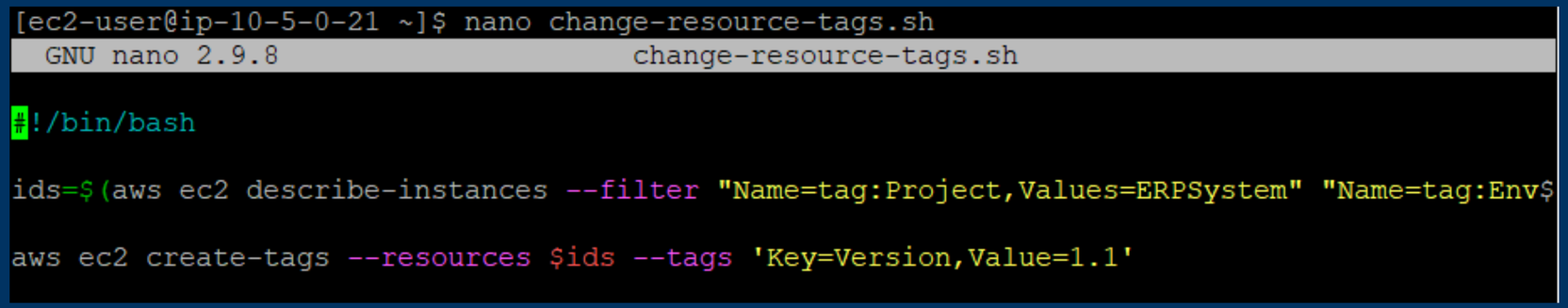
Examine el contenido del script:

#!/bin/bash

ids=$(aws ec2 describe-instances --filter "Name=tag:Project,Values=ERPSystem" "Name=tag:Environment,Values=development" --query 'Reservations[*].Instances[*].InstanceId' --output text)

aws ec2 create-tags --resources $ids --tags 'Key=Version,Value=1.1'

Para incluir el valor de la etiqueta Project (Proyecto) en el resultado, copiamon y ejecutamos el siguiente comando en el terminal de Linux:



TAREA 2: DETENER E INICIAR RECURSOS EN FUNCIÓN DE LA ETIQUETAS

En esta tarea, utilizará un script suministrado de forma previa para detener e iniciar un conjunto de instancias etiquetadas como instancias de desarrollo.

REVISAR EL SCRIPT STOPINATOR

REVISAR EL SCRIPT STOPINATOR

En la instancia Command Host, desplácese al directorio aws-tools, dentro del directorio principal, con el comando cd:

cd aws-tools

bra el archivo stopinator.php y examine su contenido:

nano stopinator.php

El script stopinator.php se utiliza para detener y reiniciar instancias de Amazon EC2 basadas en conjuntos de etiquetas especificados.

Puede operar en todas las regiones de AWS y acepta argumentos como `-t` para definir etiquetas específicas en formato `nombre=valor` y `-s` para iniciar las instancias en lugar de detenerlas.

Si no se proporcionan etiquetas, el script detendrá todas las instancias en ejecución en la cuenta.

```
[ec2-user@ip-10-5-0-21 ~]$ cd aws-tools
[ec2-user@ip-10-5-0-21 aws-tools]$ nano stopinator.php
  GNU nano 2.9.8                          stopinator.php

#!/usr/bin/php
<?php
# A simple PHP script to start all Amazon EC2 instances and Amazon RDS
# databases within all regions.
#
# USAGE: stopinator.php [-t stop-tags] [-nt exclude-tags]
#
# If no arguments are supplied, stopinator stops every Amazon EC2 and
# Amazon RDS instance running in an account.
#
# -t stop-tag: The tags to inspect to determine if a resource should be
# shut down. Format must follow the same format used by the AWS CLI.
#
# -e exclude-id: The instance ID of an Amazon EC2 instance NOT to terminate. Useful
# when running the stopinator from an Amazon EC2 instance.
#
# -p profile-name: The name of the AWS configuration section to use for
# credentials. Configuration sections are defines in your .aws/credentials file.
# If not supplied, will use the default profile.
#
# -s start: If present, starts instead of stops instances.
# PREREQUISITES
# This app assumes that you have defined an .aws/credentials file.

require 'vendor/autoload.php';
use Aws\Ec2\Ec2Client;

date_default_timezone_set('UTC');

# Obtain the profile name.
$profile = "default";
$opts = getopt('p::t::e::s::');
if (array_key_exists('p', $opts)) { $profile = $opts['p']; }

$excludeID = "";
if (array_key_exists('e', $opts)) {
        $excludeID = $opts['e'];
}

$start = false;
if (array_key_exists('s', $opts)) {
        $start = true;
}
```

DETENER Y REINICIAR EL PROCESO DE DESARROLLO DE ERPPROJECT

En esta tarea, utilizará el script stopinator.php para detener y restaurar el entorno de desarrollo del proyecto ERPSystem.

Desde el shell de Linux, ejecute el script stopinator.php:

./stopinator.php -t"Project=ERPSystem;Environment=development"

DETENER Y REINICIAR EL PROCESO DE DESARROLLO DE ERPPROJECT

En el menú Servicios, haga clic en **EC2.**
En el panel de navegación, haga clic en **Instancias.**
Verifique que dos instancias se están deteniendo o ya se hayan detenido.
Vuelva a la sesión de SSH de la instancia **Command Host** y, desde el símbolo de sistema de Linux, reinicie las instancias con el siguiente comando:

./stopinator.php -t"Project=ERPSystem;Environment=development" -s

Vuelva a la ventana de la Consola de administración de Elastic Compute Cloud y verifique que las dos instancias que se detuvieron anteriormente se estén reiniciando.

Se usó el script stopinator.php para gestionar el entorno de desarrollo del proyecto ERPSystem en AWS. Primero, se detuvieron dos instancias específicas con -t "Project=ERPSystem;Environment=development". Se verificó en la consola de EC2 que las instancias estaban apagadas. Luego, se reiniciaron las mismas instancias con -s y se confirmó que se estaban iniciando nuevamente en la consola de EC2.

```
[ec2-user@ip-10-5-0-21 aws-tools]$ ./stopinator.php -t"Project=ERPSystem;Environment=develop
ment"
Region is ap-south-1
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 1
20

        No instances to stop in Array.
Region is eu-north-1
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 1
20

        No instances to stop in Array.
Region is eu-west-3
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 1
20

        No instances to stop in Array.
Region is eu-west-2
PHP Notice:  Array to string conversion in /home/ec2-user/aws-tools/stopinator.php on line 1
20
```

Name	Instance ID	Instance state	Instance type
web server	i-001bc73d0744f89f6	Running	t3.micro
web server	i-0e1f6a451ba260cad	Running	t3.micro
NAT	i-034bc67c52eb1efbe	Running	t3.micro
app server	i-0537dd151adbf5f0e	Running	t3.micro
web server	i-08965aa2cb8cadb02	Running	t3.micro
web server	i-0876ff03ac8cd2497	Stopped	t3.micro
Command Host	i-04ba3e6a732147b99	Running	t3.medium
web server	i-06519bb4a8732c13e	Running	t3.micro
app server	i-0e41c6e5284b3b1e8	Stopped	t3.micro

TAREA 3.1: REVISAR EL SCRIPT TAG-OR-TERMINATE (ETIQUETA O TERMINACIÓN)

Abra el archivo terminate-instances.php con el editor nano.

nano terminate-instances.php

```
#!/usr/bin/php

<?php
require 'vendor/autoload.php';
use Aws\Ec2\Ec2Client;

$region = "us-west-2";
$subnetid = "";
$profile = "default"; # Only needed if not using IAM roles

# Necessary to quell a PHP error.
date_default_timezone_set('America/Los_Angeles');

array_shift($argv);
if (count($argv>0)) {
        do {
                $elem = array_shift($argv);
                if ($elem == "-region") {
                        $region = array_shift($argv);
                } elseif ($elem == "-subnetid") {
                        $subnetid = array_shift($argv);
                }
        } while (count($argv) > 0);
}

# Iterate through all available AWS regions.
$ec2 = Ec2Client::factory(array(
        'profile' =>$profile,
        'region' => $region
));

# Obtain a list of all instances with the Environment tag set.
$goodInstances = array();
$terminateInstances = array();

$tagArgs = array();
array_push($tagArgs,  array(
        'Name' => 'tag-key',
        'Values' => array('Environment')
));
$ec2DescribeArgs['Filters'] = $tagArgs;
```

Una vez dentro del editor de texto, notamos que el script toma dos argumentos: la región actual (region) y el ID de una subred (subnetid). Utiliza subnetid para identificar instancias no conformes.

Primero, el script usa el método **describeInstances()** para encontrar todas las instancias con la etiqueta **Environment** y guarda sus IDs en una tabla hash.

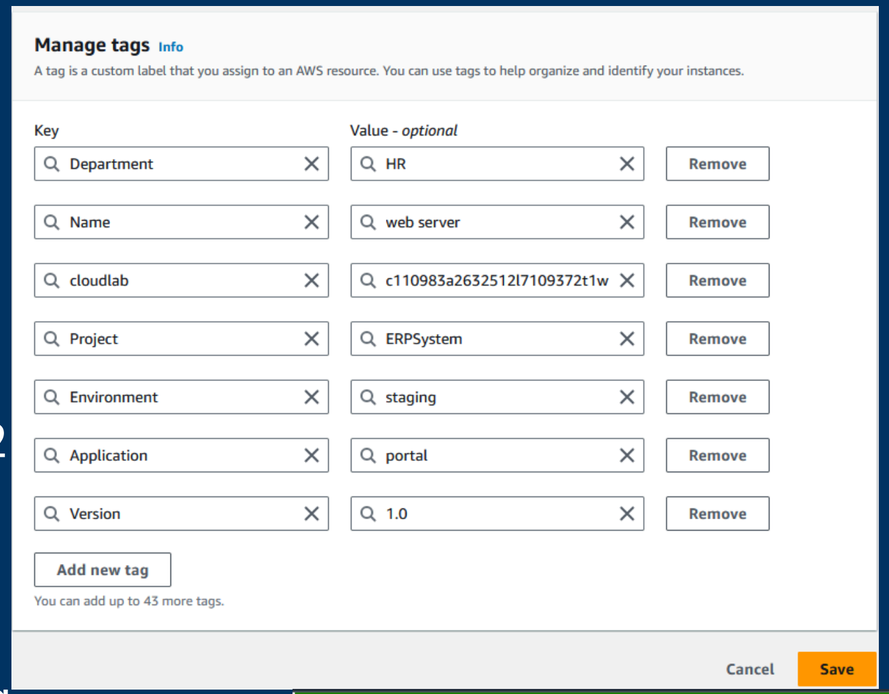
Luego, compara todas las instancias en la subred con la lista de instancias etiquetadas. Las instancias no etiquetadas se añaden a una lista para terminar.

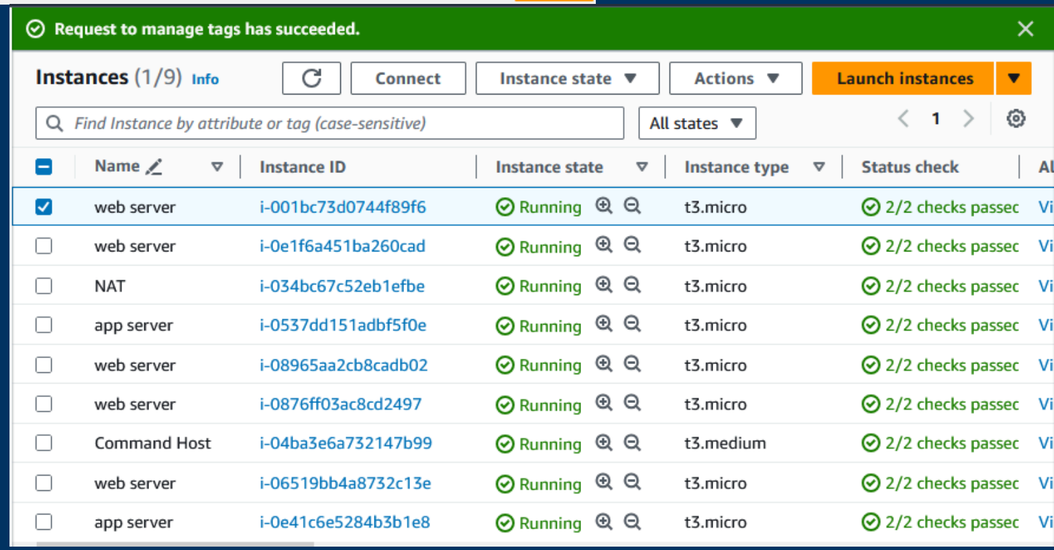
Finalmente, el script utiliza la lista de IDs no conformes como argumento para el método terminateInstances, finalizando esas instancias.

Configuramos el entorno para probar el script

Antes de ejecutar el script, se necesita ajustar algunas instancias en el laboratorio para eliminar la etiqueta Environment.

Regresamos a la Consola de Administración de EC2 y revisamos las instancias en el entorno de laboratorio. Elegimos una instancia en la subred privada. En la pestaña de Etiquetas, hacemos clic en Agregar/Editar Etiquetas. Localizamos la etiqueta Environment y hacemos clic en el ícono de eliminar. Guardamos los cambios. Repetimos este proceso para otra instancia en la subred privada.

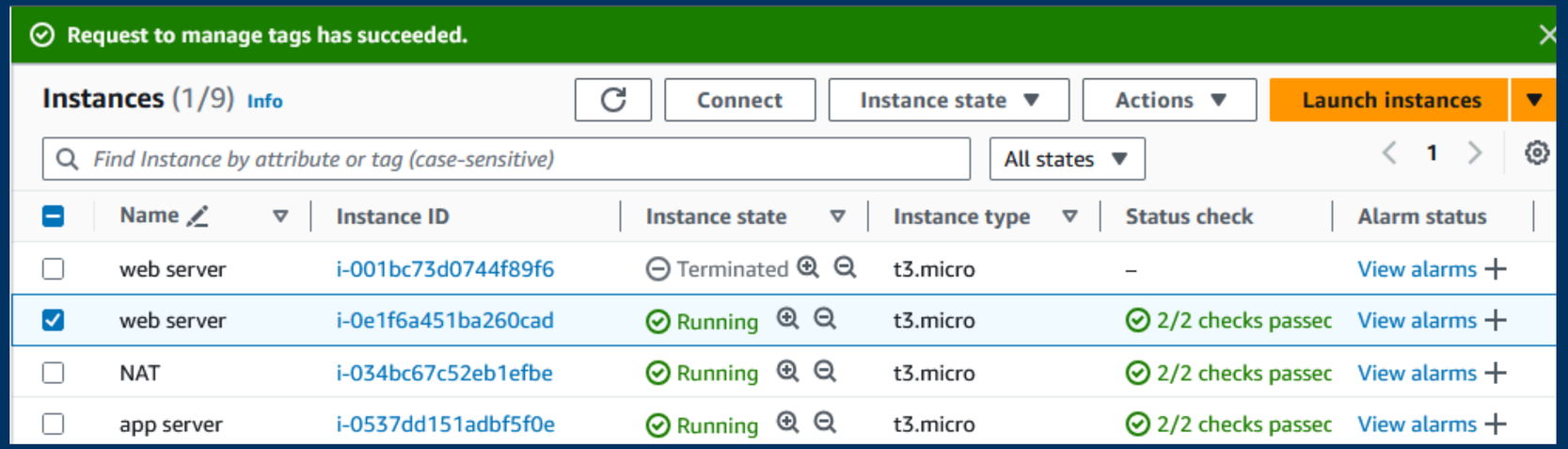




Ejecutamos el script

```
[ec2-user@ip-10-5-0-21 aws-tools]$ ./terminate-instances.php -region us-west-2 -subnetid sub
net-0e3308e80c84c289a

Checking i-001bc73d0744f89f6
Checking i-0e1f6a451ba260cad
Checking i-0537dd151adbf5f0e
Checking i-08965aa2cb8cadb02
Checking i-0876ff03ac8cd2497
Checking i-06519bb4a8732c13e
Checking i-0e41c6e5284b3b1e8
Terminating instances...
Instances terminated.
```



En este caso vamos a ir a nuestras intancias y elegiremos una de la subned privada.
Copiaremos el campo de az menos la última letra y el id de la misma.
En esta linea sustituremos la az en region y la subnet donde se indica
./terminate-instances.php -region <region> -subnetid <subnet-id>

¡GRACIAS POR SU ATENCIÓN!