﻿﻿﻿﻿Cloud Hacking - Learn and exploit a public AWS s3 bucket.  
  
🎄 Learn the AWS Basics  
🎄 Understand Simple Storage Service & IAM Roles  
🎄 Discover & leak files from an s3 bucket

In a move to taunt the Best Festival Company, Grinch Enterprises sends out an email to the entire company with everyone's name and date of birth. McSkidy looks quite stressed with the breach and thinks about the potential legal consequences. She talks to McInfra to try to determine the origin of the breach.

Shadow IT

Sometimes business units go around corporate IT, procurement, legal, and security when they need to get the job done quickly. This leads to security teams not knowing what they need to protect and systems not built to IT or Security standards.

Public Cloud is an easy way for business units to engage in shadow IT. And the most accessible public cloud to get started with is AWS.

What are the differences between these two images? (Hint: right-click and look at the image addresses)

One of these is an external link from an Amazon S3 Bucket.

Getting Started

You'll need to start your AttackBox to run commands using the AWS CLI for today's lesson. Your target will be an AWS account and some resources hosted by TryHackMe for this year's Advent of Cyber.

**Please note:** If you are on the TryHackMe free plan, the attack box does not have internet access and cannot reach AWS. You will need to install curl and the AWS CLI on your own machine in order to complete this challenge. Instructions for installing the AWS CLI are here: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

Today's Learning Objective

Today we'll be covering the basics of AWS - one of the leading public cloud providers, and two of its most common services - [Amazon S3 (Simple Storage Service)](https://aws.amazon.com/s3/) and [AWS IAM (Identity and Access Management](https://docs.aws.amazon.com/IAM/latest/UserGuide/introduction.html)).

We'll show you how to get started with the AWS CLI, then describe how to discover public S3 buckets, look at what's inside. We will also look at how you could leverage an IAM Access Key & Secret.

Amazon AWS

Amazon AWS is a public cloud service provider. As of their most recent financial disclosures, AWS accounts for the bulk of Amazon's profit. Most major enterprises leverage AWS in some form or another for Compute Services, Big Data or Machine Learning, Data Archive, Video Streaming, IoT, etc. The number of services AWS supports is so vast that we can barely fit it all in this screenshot. 

Your eyes don't deceive you. You can access robots, blockchain, satellites, and quantum computing from AWS.

[AWS divides its infrastructure into Regions](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-available-regions), mostly independent clusters of datacenters. Within each region are availability zones (AZ). Each AZ in a region leverages separate power grids and usually are located in different flood plains. This redundancy allows you to establish highly resilient architectures to withstand significant weather or geological events, or more frequently, hardware or facility failures.

Because regions are independent - you'll get different answers to questions depending on the region you are querying. You can specify a region with the --region option to the AWS CLI.

You can access AWS via the AWS Console, AWS CLI, AWS API, or the associated SDKs for your favorite programming languages.

Amazon S3

[Amazon S3](https://aws.amazon.com/s3/) (Simple Storage Service) is their hosted object storage service. Objects are stored in Buckets. To highly simplify the concept of object storage, Buckets are key-value stores, with the Object Key being a full pathname for a file and the value being the contents of the file. S3 is a publicly hosted service - it doesn't exist behind a corporate firewall, making it convenient for hosting public content. AWS has an entire feature set around [hosting a public website in S3](https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html).

AWS Buckets use a global namespace. Only one AWS customer can create a bucket named bestfestivalcompany-images.

Amazon S3 is used for more than public hosting. It has many uses for data archive, video processing, regulatory record retention, etc. The challenge for Best Festival Company, like any enterprise using S3, is that sometimes data gets mixed up, and data that shouldn't be public gets made public.

Discovering Bucket Names

There are many ways to discover the names of Buckets. One of the easiest ways is when a company embeds content hosted in S3 on their website. Images, PDFs, etc., can all be hosted cheaply in S3 and linked from another site. These links will look like this:

http://BUCKETNAME.s3.amazonaws.com/FILENAME.ext

or

http://s3.amazonaws.com/BUCKETNAME/FILENAME.ext

In both these cases, it is easy to identify the name of the S3 bucket. Now, what can we do with that information?

Listing the Contents of Buckets

Amazon S3 is one of AWS's oldest services. It's so old that it has two different methods of access control: [Bucket Policies](https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html) and [S3 ACLs](https://docs.aws.amazon.com/AmazonS3/latest/userguide/managing-acls.html). This leads to great confusion for developers who must manage policies, ACLs, and the differences between [Any User and Authenticated Users](https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html#specifying-grantee-predefined-groups).

Many buckets that contain public information allow you to list the contents of the bucket. In your AttackBox, try running the command:

curl http://irs-form-990.s3.amazonaws.com/

That massive pile of XML is a listing of all the IRS Form 990 filings for US Tax-Exempt corporations. AWS makes this data available as a [public dataset](https://registry.opendata.aws/irs990/).

If mentally parsing XML that contains no line breaks isn't your cup of tea, the AWS CLI also provides the ability to list the contents of a bucket (You probably want to hit Ctrl-C after a few seconds, there are a lot of US non-profit organizations).

aws s3 ls s3://irs-form-990/ --no-sign-request

The option --no-sign-request allows you to request data from S3 without being an AWS Customer.

Downloading Objects

Downloading an object from S3 is also easy. You can use curl:

curl http://irs-form-990.s3.amazonaws.com/201101319349101615\_public.xml

or the AWS CLI:

aws s3 cp s3://irs-form-990/201101319349101615\_public.xml . --no-sign-request

Note the two different URIs for an object. Objects can be addressed with http:// or via s3://

The different levels of Amazon S3 Authentication

In Amazon S3, Object permissions are different from Bucket permissions. Bucket permissions allow you to list the objects in a bucket, while the object's permissions will enable you to download the object. In the case of the irs-form-990 bucket, both the bucket and all the objects in the bucket are publicly readable. But that doesn't have to be the case. Objects can be readable while the bucket is not, or the bucket can be publicly readable, but the Objects are not.

Note: you can also have public write permissions to a Bucket. This is generally a bad idea and has been the vector of several [crypto-mining incidents](https://www.theregister.com/2018/02/22/la_times_amazon_aws_s3/).

There are also two levels of public buckets and objects. The first level is "Anyone." This is what you experienced with the irs-form-990 bucket. You could just hit that URL from your local browser. The second level is just as public - and that is public to Any AWS Customer (what AWS foolishly called AuthenticatedUsers for many years). Anyone with a credit card can create an AWS account; therefore, Authenticated Users doesn't provide much data protection.

|  |  |  |
| --- | --- | --- |
| **ACL Name** | **BUCKET** | **OBJECT** |
| **Anyone** | Anonymously list contents of  the bucket via curl or with  aws s3 ls --no-sign-request | Ability to download via curl or  aws s3 cp --no-sign-request |
| **AuthenticatedUsers** | Can only list the bucket with  active AWS keys via  aws s3 ls | You can only download the object with  active AWS Keys via  aws s3 cp |

AWS IAM

Excluding a few older services like Amazon S3, all requests to AWS services must be signed. This is typically done behind the scenes by the AWS CLI or the various Software development Kits that AWS provides. The signing process leverages IAM Access Keys. These access keys are one of the primary ways an AWS account is compromised.

IAM Access Keys

IAM Access Keys consist of an Access Key ID and the Secret Access Key. 

Access Key IDs always begin with the letters AKIA and are 20 characters long. These act as a user name for the AWS API.  The Secret Access Key is 40 characters long. AWS generates both strings; however, AWS doesn't make the Secret Access Key available to download after the initial generation.

There is another type of credentials, short-term credentials, where the Access Key ID begins with the letters ASIA and includes an additional string called the Session Token.

Conducting Reconnaissance with IAM

When you find credentials to AWS, you can add them to your AWS [Profile](https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html) in the AWS CLI. For this, you use the command:

aws configure --profile PROFILENAME

This command will add entries to the .aws/config and .aws/credentials files in your user's home directory.

Once you have configured a new profile with the new access keys, you can execute any command using this other set of credentials. For example, to list all the S3 Buckets in the AWS account you have found credentials for, try:

aws s3 ls --profile PROFILENAME

ProTip: Never store a set of access keys in the [default] profile. Doing so forces you always to specify a profile and never accidentally run a command against an account you don't intend to.

A few other common AWS reconnaissance techniques are:

1. Finding the Account ID belonging to an access key:

aws sts get-access-key-info --access-key-id AKIAEXAMPLE

1. Determining the Username the access key you're using belongs to

aws sts get-caller-identity --profile PROFILENAME

1. Listing all the EC2 instances running in an account

aws ec2 describe-instances --output text --profile PROFILENAME

1. Listing all the EC2 instances running in an account in a different region  
   aws ec2 describe-instances --output text --region us-east-1 --profile PROFILENAME

AWS ARNs

An Amazon ARN is their way of generating a unique identifier for all resources in the AWS Cloud. It consists of multiple strings separated by colons.

The format is:

arn:aws:<service>:<region>:<account\_id>:<resource\_type>/<resource\_name>

Challenge

Somehow, the Grinch has managed to get hold of all the Elves' names and email addresses. How could this have happened? Given the scope of the breach, McSkidy believes someone in HR must be involved. You know that HR recently launched a new portal site using WordPress. You also know that HR didn't request any infrastructure from IT to deploy this portal site. Where is that portal hosted?

Here is the image HR sent out announcing the new site:



Based on that, can you figure out how the Grinch got access to the employee database?

***Answer the questions below***

What is the name of the S3 Bucket used to host the HR Website announcement?



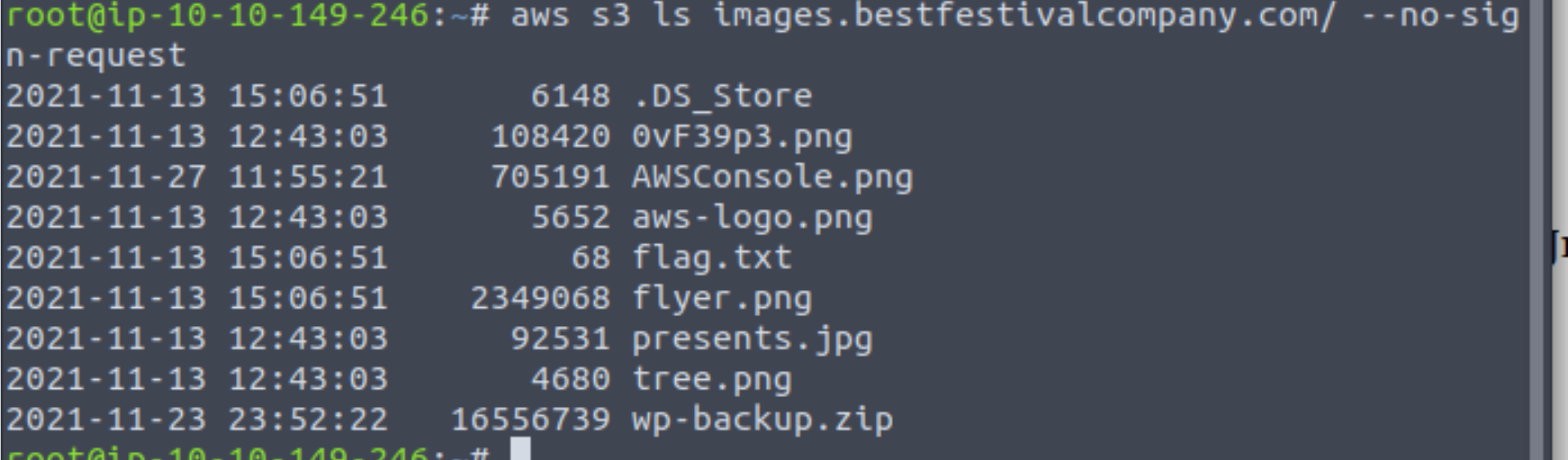
 Submit

What is the message left in the flag.txt object from that bucket?

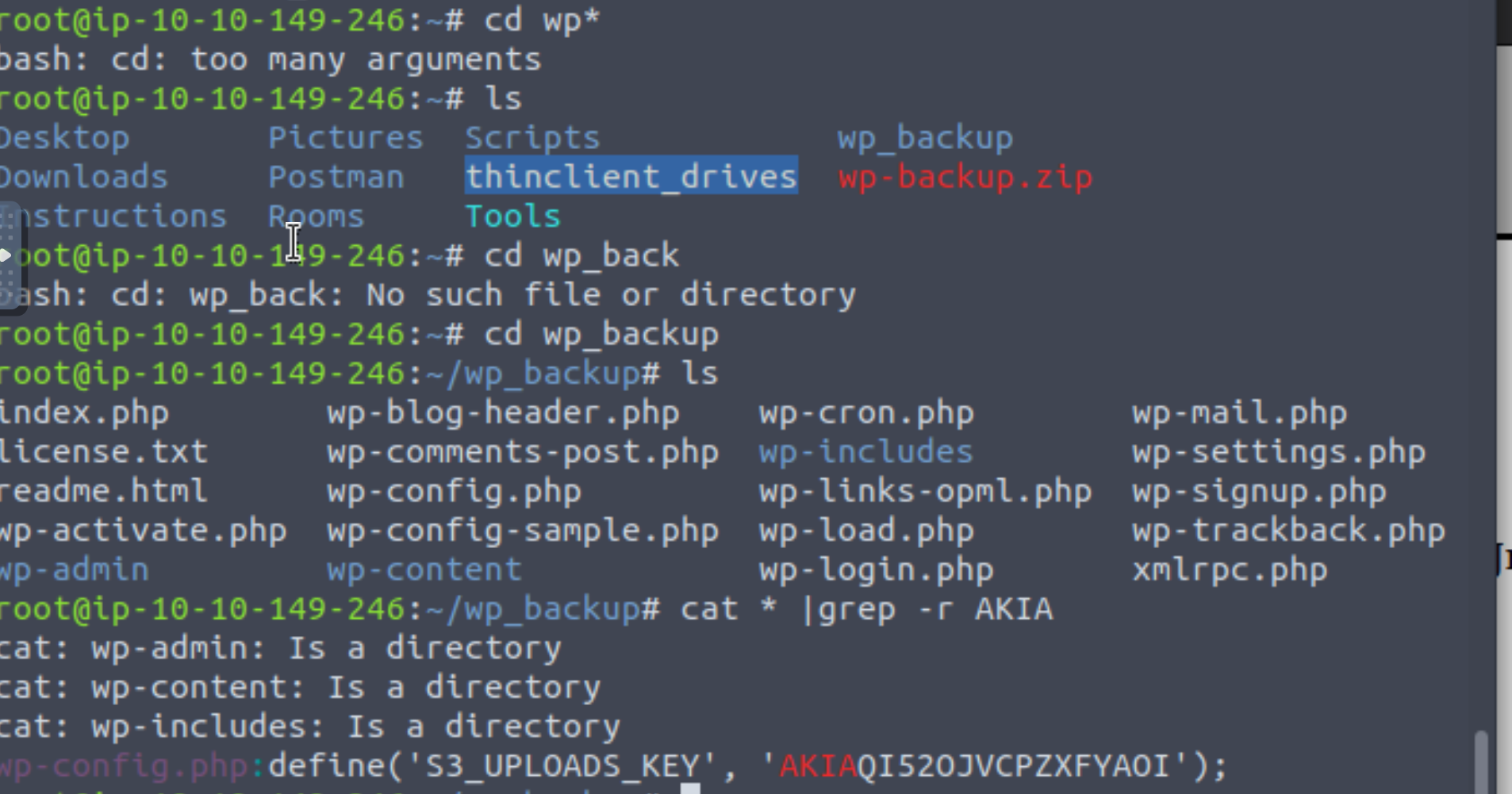


 Submit

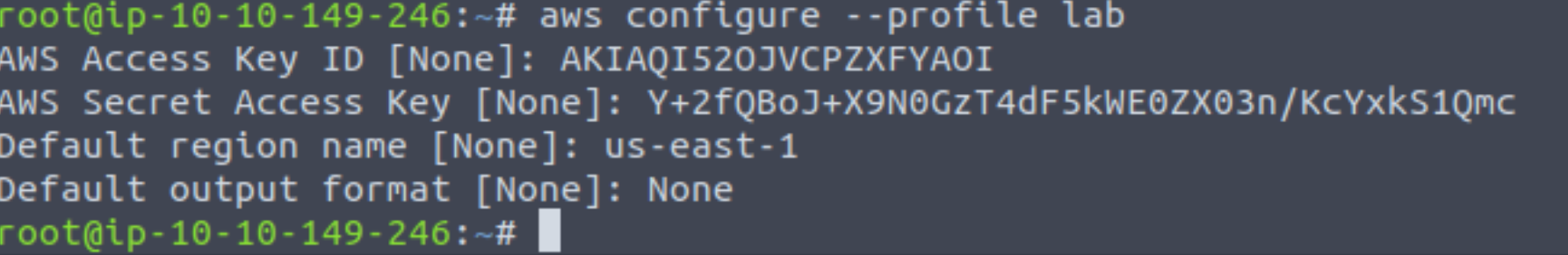
What other file in that bucket looks interesting to you?



What is the AWS Access Key ID in that file?

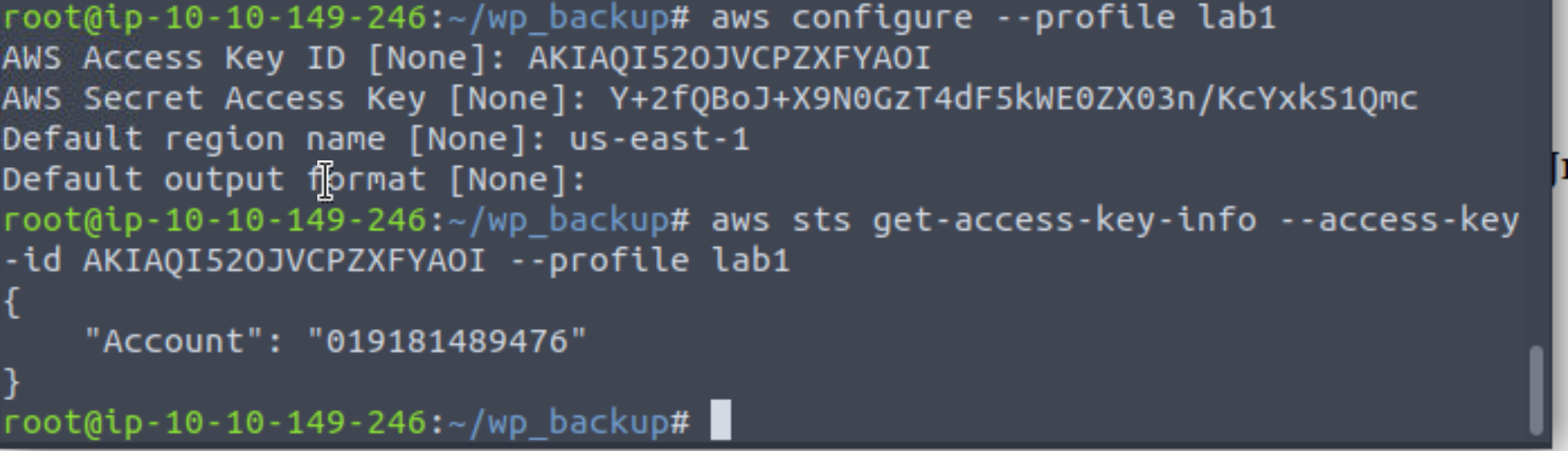


What is the AWS Account ID that access-key works for?

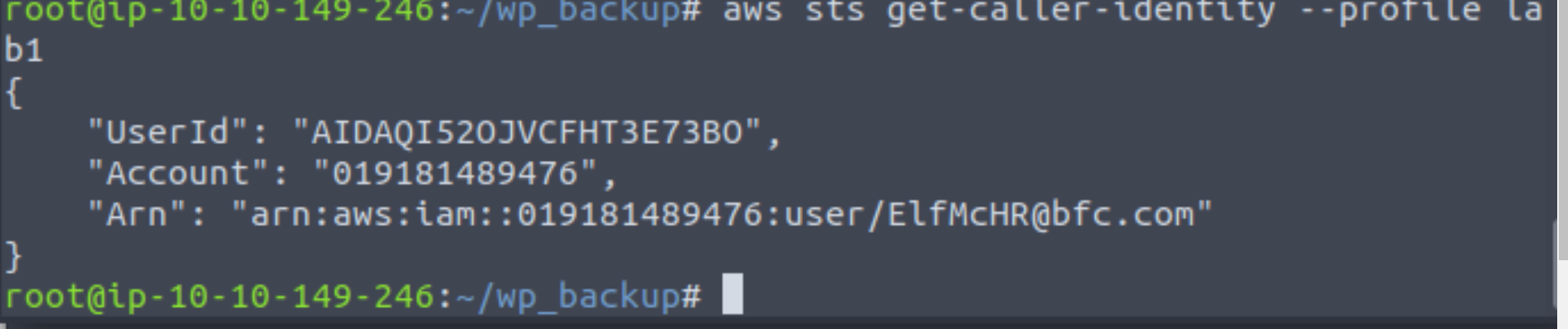




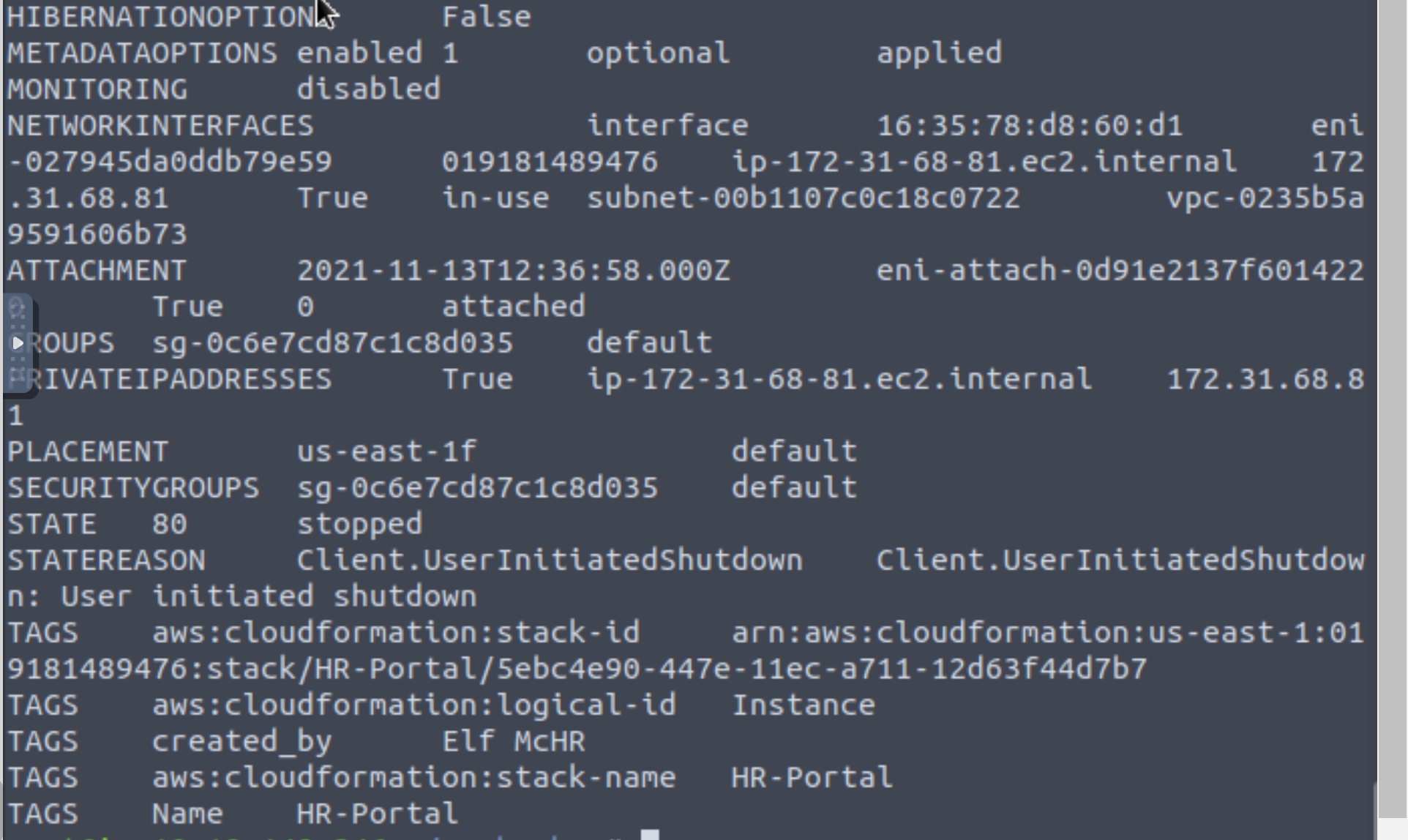
Cant do it unless profile setup



What is the Username for that access-key?

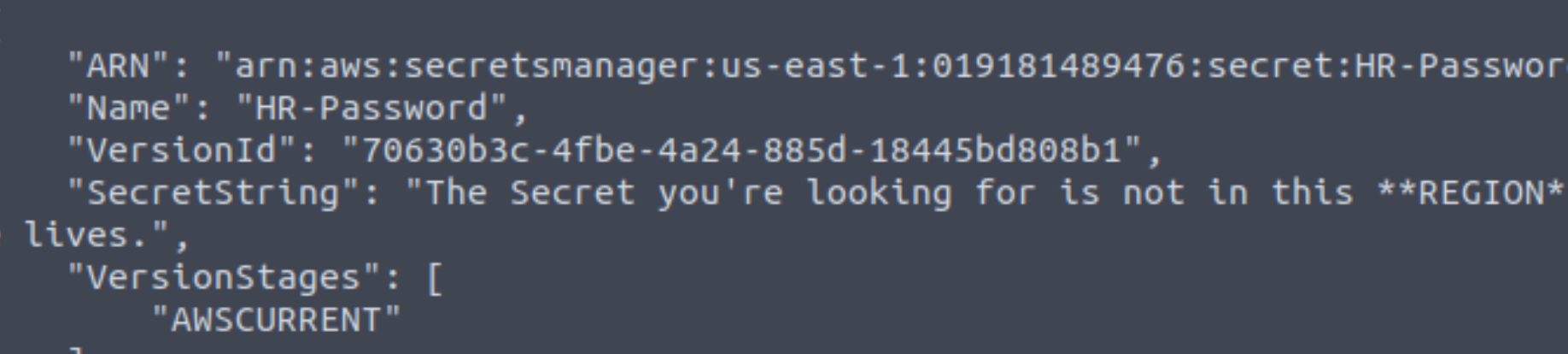
There is an EC2 Instance in this account. Under the TAGs, what is the Name of the instance?

aws ec2 describe-instances --output text --profile PROFILENAME



What is the database password stored in Secrets Manager?

Aws secretsmanager list-secret –profile lab1 //give name of scret db



Have to go by region to get the latency out

