# Objective: Recover the Cloud Ring

Much of the basic knowledge you need to work in the cloud is the same as you need when your hardware lives on-premises: operating systems (especially Linux), networking, scripting, and cryptography for starters. For cloud technology you need to add another layer of scripting using the provider’s tools to assemble the pieces of cloud functions into a solution. This challenge will teach you some of the basics for using the AWS cloud. The terminals all have the AWS CLI installed, so you won’t need to do that.

A picture containing text, sign

Description automatically generated

## AWS CLI Intro

The AWS Command Line Interface (CLI) is a set of tools provided by [AWS that you install](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html) on your workstation or server. AWS also supports a [web-based management console](https://aws.amazon.com/console/), but for an installation of any size it is cumbersome and time consuming. Large installations are almost always created and maintained through AWS CLI or through a scripting language like [Terraform](https://aws.amazon.com/blogs/apn/terraform-beyond-the-basics-with-aws/) that leverages the AWS CLI. The CLI is based on an executable named aws, so all the commands start with aws.

Graphical user interface

Description automatically generated with medium confidence

A screenshot of a video game

Description automatically generated

Text

Description automatically generated

Graphical user interface

Description automatically generated  
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/sts/get-caller-identity.html>

The terminal leads you through the commands you need, so there will not be much explanation given here. The commands and links from the terminal are supplied in case you get stuck.  
Text

Description automatically generated

### Configure the aws cli credentials

The credentials are given in the terminal:  
access key: AKQAAYRKO7A5Q5XUY2IY  
secret key: qzTscgNdcdwIo/soPKPoJn9sBrl5eMQQL19iO5uf  
region: us-east-1 .  
<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html#cli-configure-quickstart-config>

A screenshot of a computer

Description automatically generated with medium confidence

### Use aws sts

The commands are  
aws sts help  
aws sts get-caller-identity  
Text

Description automatically generated  
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/sts/index.html>

### Trufflehog Search

Unless the developer takes [explicit actions to remove them](https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/removing-sensitive-data-from-a-repository), any credentials mistakenly submitted to a Git repository remain there for the life of the repository. Attackers can find these credentials and use them to compromise your installation.

A picture containing text, toy

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Text

Description automatically generated with low confidence<https://haugfactory.com/asnowball/aws_scripts.git>

You need to find the AWS secrets that Alabaster [left in this repository](https://haugfactory.com/asnowball/aws_scripts.git) using the Trufflehog tool. Gerty does not have a terminal, but you can skip ahead to Sulfrod’s terminal which has Trufflehog installed. Alabaster’s repository is on the public Internet, so you can also install Trufflehog on your own machine. Interestingly, the version of Trufflehog I installed found the credentials immediately without needing to open previous commits.

A picture containing text

Description automatically generated

### Question

Use Trufflehog to find the name of the file where Alabaster stored his credentials. If you use Sulfrod’s tmux terminal, the results will scroll off the top. You can [regain your scrolling functions in tmux](https://superuser.com/questions/209437/how-do-i-scroll-in-tmux) by pressing CNTL-b, then [. When you want to enter commands again, use CNTL-c.

### Answer

Trufflehog finds a string, AKIAAIDAYRANYAHGQOHD, that could be a key.  
A screenshot of a computer

Description automatically generated

The hint, Checkout Old Commits, suggests we use the command  
git checkout <commitNumberHere>, but I find git show <commitNumberHere> to be easier.  
To do that, we clone the repo with  
git clone <https://haugfactory.com/asnowball/aws_scripts.git>  
git show 106d33e1ffd53eea753c1365eafc6588398279b5

Text

Description automatically generated

There are keys.  
aws\_access\_key\_id="AKIAAIDAYRANYAHGQOHD" aws\_secret\_access\_key="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"

The file that contains them is put\_policy.py, so enter that into the objective.

## Exploitation via the AWS CLI

This challenge picks up where we left off at Sulfrod’s terminal. You will use the AWS credentials that Alabaster left in the Git repository to gain access to his AWS to see what you can find, and learn about AWS policies on the way. The talk by Chris Elgee is a good place to start, especially the demonstration towards the end.

Text

Description automatically generated  
<http://www.youtube.com/watch?v=t-xDvVUialo> Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated  
<https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html>

The terminal leads you through this challenge, although you do have to read the reference URLs carefully. Therefore, other than the starting configuration, I’ll just give you the questions and answers.

### Answers

Text

Description automatically generated

A screenshot of a computer

Description automatically generated  
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/index.html>   
aws iam list-attached-user-policies --user-name haug

Graphical user interface, text

Description automatically generated  
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/index.html>   
aws iam get-policy --policy-arn arn:aws:iam::602123424321:policy/TIER1\_READONLY\_POLICY

Graphical user interface, text, application

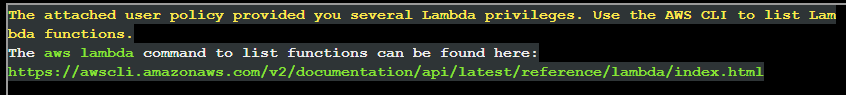
Description automatically generated  
aws iam get-policy-version --policy-arn arn:aws:iam::602123424321:policy/TIER1\_READONLY\_POLICY --version-id v1

Graphical user interface, text, application

Description automatically generated  
aws iam list-user-policies --user-name haug

A screenshot of a computer

Description automatically generated  
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3api/index.html>   
aws s3api list-objects --bucket smogmachines3

  
<https://awscli.amazonaws.com/v2/documentation/api/latest/reference/lambda/index.html>   
aws lambda list-functions

A screenshot of a computer

Description automatically generated with medium confidenceaws lambda get-function-url-config --function-name smogmachine\_lambda

The URL we recover just prints an AWS error message. It would be nice to have something we could show off to prove our hacking prowess; this nifty ring will have to be enough.

A picture containing text, device

Description automatically generated