with
COMMUNICATIONS

AWS CLI: Amazon Web Service Command Line Interface

January 24, 2017

The Amazon Web Service Command Line Interface (AWS CLI for short) expands the options for a user interacting with the AWS. It makes AWS tools accessible via the API because while a user can log in to the web console through a browser, not all the options are exposed through that console. Without the AWS CLI, extra code is required to talk to the API to gain access to the full catalogue of potential interactive options with AWS.

Server-side encryption settings for S3 buckets, for example, can have specific keys encrypted by the AES-256 encryption mechanism when a user instructs the AWS to perform the encryption through the web console. However, users are unable to utilize the key management service (KMS keys) directly with AWS S3 without using the API. This CLI uses the API to expose hidden features that would normally have to be coded through the command line. It isn't limited to just one command line either - the CLI can be fed various lines. Essentially, the user acts as if they are utilizing the API from a command line in order to configure options missing from inside the web console.

For most people, the AWS CLI proves to be a nice utility. Its abilities mean that those users who are savvy with command lines and who are seeking greater use of the AWS don't always have to go to the web interface to interact because the web console is made more accessible.

To illustrate, consider the difference between Windows and Linux. People who use Windows typically click around to find all the various configuration screens and options; however, these options that are present in Window's display are not all available from the web browser in the case of the AWS. When the AWS is used to its full potential through access to the CLI, it ends up being more like Linux than Windows. People who use Linux can enter a few commands on the command line to avoid searching through UI's to find all of these options. In short, they can accomplish their task directly without the hassle of searching.

It almost goes without saying that one of the biggest benefits to using a CLI is speed. This holds true for AWS API's as well because using this particular CLI allows for changes to be made faster than if a user had to log into and click through a web browser to input commands. Similarly, it expedites the coding process by allowing users to script a set of commands or deployments (i.e. attaching EBS block devices, encrypting said block devices, or setting up a cloud front server in an ELB) in the CLI that will repeat, saving time for the coder.

The AWS CLI also saves thee user time when it comes to installation: just pick the package management system for the platform of choice and said platform will initiate the installation. For example, Mac users can install the CLI through Homebrew, and there are plenty of options for Windows and Linux users as well since the software is based on Python Boto3 API.

Conveniently, the CLI documentation is thorough and all its functions mimic the Boto3 API documents as well. If a user knows Boto3, they should easily be able to transfer that knowledge to their experience with this CLI.

Users have the option to set environment variables for their AWS secret key using their API key and their secret key. Typically, it is good to set up both which region (i.e. U.S. West or U.S. East) they want to connect to, as well as an .AWS directory document containing credentials and a configuration to set up profiles.

It is possible to have multiple AWS accounts managed by one system simultaneously using document files such as a default, personal account or a company account. While CLI tools are running, the user can specify which profile they want to run and can even switch back and forth between profiles without having to reset environment variables. Just set up an .AWS folder, enter credentials and profiles, then run AWS commands directly: type AWS, then type S3 and/or a sub command.

The same goes for using an LS and a bucket – a list can be generated to show the keys in that bucket. In addition, typing CP creates a copy which can be sent to an S3 bucket.

The list doesn’t end there, other potential functions of this CLI include:

- Synchronizing an entire folder item into S3
- Manipulating metadata using the S3 API
- Placing bucket policies
- Setting encryption policies using specific KMS keys
- Using a custom reviving key (an advanced feature where the user controls the keys instead of the Amazon KMS system)
- Compiling a list of a bucket’s metadata
- Setting permissions of ownership
- Creating data into websites

The list of options is numerous because the AWS CLI and the S3 API sub command can do anything the S3 can do through a console, and then some.

While the benefits of the AWS CLI are great, one critique of the software stems from a seeming lag in the Boto3 Python library that may cause documents to appear wrong or incomplete if they are not kept up-to-date. There have been a few instances where a document is known to be in the API; however, the documentation lags behind the development of the code. Users should be wary of this if they try to complete an action that they do not believe exists, because it just might.

http://www.sixfeetup.com/blog/hidden-features-via-aws-cli