CREATIVE IDEA: SINGLE-SIGN-ON WITH G SUITE FOR DEVELOPMENT CLIENTS!

GSuite | Single-Sign-On (SSO)

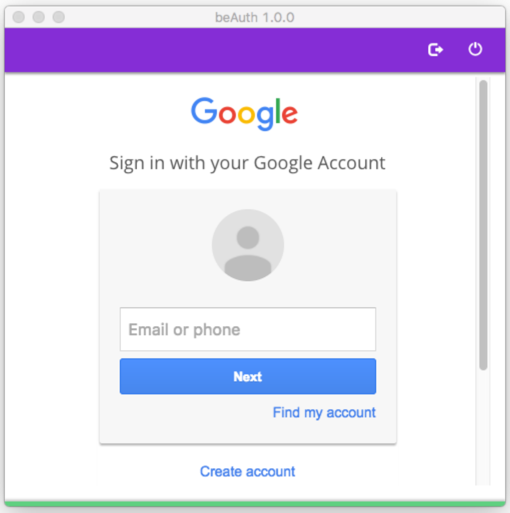
Simone Merlini | 4 May 2017

In the **last article**, we discussed how to use corporate G Suite accounts to log in via **Single-Sign-On** on the **Amazon Web Services** web console.

Access to the web console only covers some of the needs of people who work with AWS every day. In particular, developers and DevOps almost always require an **access key/secret key pair** on their PCs to use the **AWS CLI**, to call single AWS APIs (such as the ones for new AI services such as **Rekognition** and **Lex**), and to be able to use all the desktop applications (for example, the various file managers based on S3—such as the excellent **CloudBerry File Explorer**, or Git clients for using **CodeCommit**) which in turn use the AWS APIs.

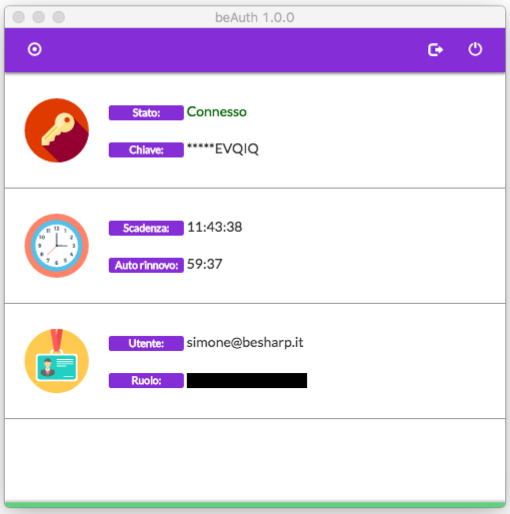
Access keys and secret keys are **not directly bound to one IAM role** (whose use through the AssumeRole API we have already seen to be security best practice) but **require a dedicated IAM user**, which would make it pointless to assume an AWS role with centralised credentials.

With no way around it, this limitation required a somewhat creative solution, and so we at **beSharp came up with beAuth.**



At startup, the program shows the G Suite login screen (which may be associated with Google's two-factor authentication)

beAuth is a small piece of software that can be installed as an agent within the operating system and which uses G Suite's credentials and the **SAML protocol**-based SSO mechanism (which we saw in the previous article) to generate temporary access key/secret key pairs that are linked to the assumed IAM role and that are rotated at predetermined intervals (typically 1 hour) within the configuration of the AWS CLI. By using this, in addition to the CLI itself, all the services that rely on it can access AWS resources by temporarily inheriting the permissions of the assumed role, without the need for a dedicated IAM user.

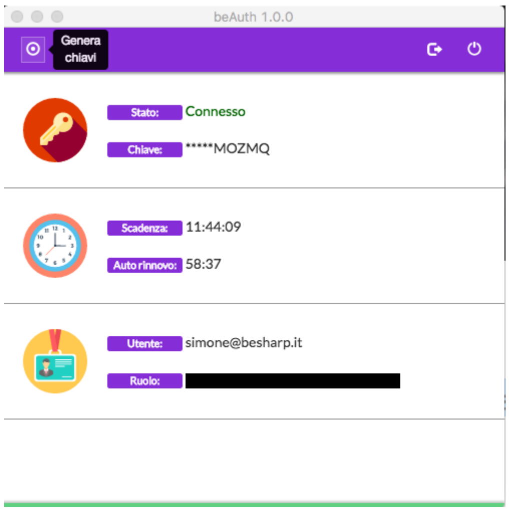


Stato: Connesso
Chiave: *****EVQIQ
Scadenza: 11:43:38
Auto rinnovo: 59:37
Utente: simone@besharp.it
Ruolo: [redacted]

beAuth's control panel shows the session information (which lasts 12 hours by default) and the validity period of the access/secret key pair (which expires after one hour), identifies the user who is logged in, the IAM role assumed, and the keys that are active at any given moment. The active key is configured automatically by default in the AWS CLI

Furthermore, since one IAM role can be even assumed **cross-account** with the same G Suite credentials (properly configured), access/secret key pairs can be obtained for different accounts, **which is an extremely useful solution should the company have multiple AWS accounts** (such as test—staging—production) or in the event that you administrate multiple AWS accounts on behalf of several clients.

For all software that does not rely on CLI but needs access/secret key pairs to be directly inserted, **beAuth allows you to generate disposable pairs** that last up to 1 hour and can be manually inserted as needed for individual calls or working sessions.



Stato: Connesso
Chiave: *****MOZMQ
Scadenza: 11:44:09
Auto rinnovo: 58:37
Utente: simone@besharp.it
Ruolo: [redacted]

These keys can be regenerated as needed and copied and pasted into any third-party application

There are many advantages to this solution:

- **Convenience**: you can access ALL AWS resources (not just the CLI) with a single centralised account, which is the same as the G Suite account, without having to implement complex Active Directory infrastructures.
- **Security**: the whole thing is based on the mechanism of **AssumeRole and STS tokens**, and therefore fully complies with security best practices as dictated by AWS, the same that are fundamental to services such as S3, **considered secure by many Fortune 500 companies**. The fact that the keys can be rotated very frequently makes the whole thing even more rock solid!
- **Simplicity**: beAuth configures itself automatically, so once you have done the initial setup (5 minutes), all that's required is for the user to log in with their G Suite credentials, with no need to manage complicated scripts or spend all day switching between different credentials and accounts, so they can concentrate on their core business.
- **Portability**: beAuth was made in **Electron**, so it is portable to any desktop operating system (Windows, Mac OS, Linux), has a very small footprint in terms of consumption of system resources, is completely under the user's control, and does not require special permissions to run.
- **Costs**: all this generates 0 (ZERO) costs on the AWS side, which can't hurt

What do you think of our creative solution? Are you interested in implementing beAuth within your company? Do you want to create something similar in house? Do you have any suggestions or questions? **Contact us** and comment below to get all the answers!

Simone Merlini

CEO e co-fondatore di beSharp, Cloud Ninja ed early adopter di qualsiasi tipo di soluzione *aaS. Mi divido tra la tastiera del PC e quella a tasti bianchi e neri; sono specializzato nel deploy di cene pantagrueliche e nel test di bottiglie d'annata.

Get in touch

beSharp.it
proud2becloud@besharp.it

Copyright © 2011-2021 by beSharp srl - P.IVA IT02415160189