

The AFP Project: Secure Cloud Authentication for Machines and Humans

Valentin Haenel @esc_____

2015-12-15 @ AWS Meetup Berlin

Version: v1.0-1-g162bdaffec <https://github.com/immobilienscout24/afp-talk>



This work is licensed under the *Creative Commons Attribution-ShareAlike 3.0 License*.

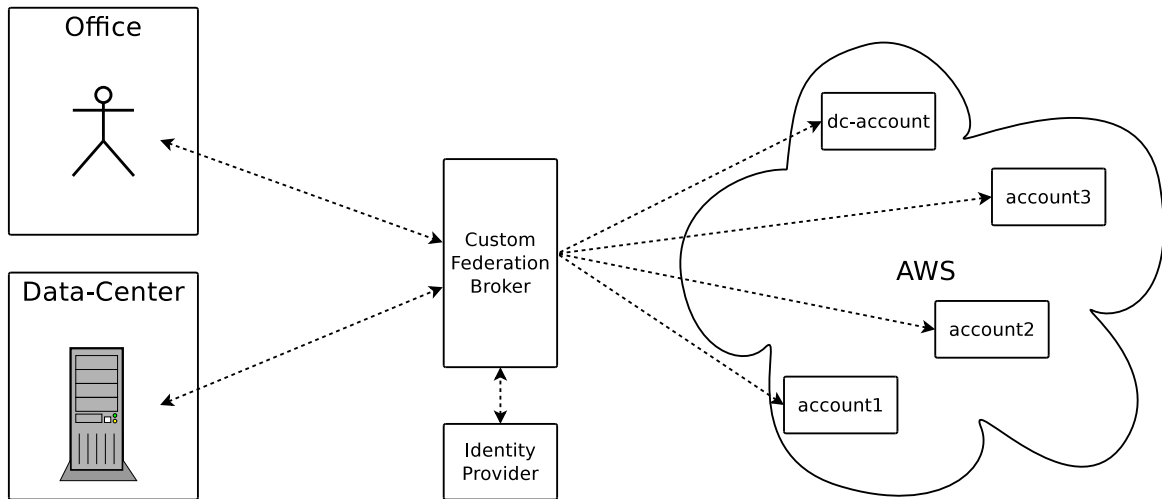
- 1 Introduction
- 2 Our Problem
- 3 afp-core
- 4 afp-web
- 5 afp-cli
- 6 afp-alppaca
- 7 Epilogue

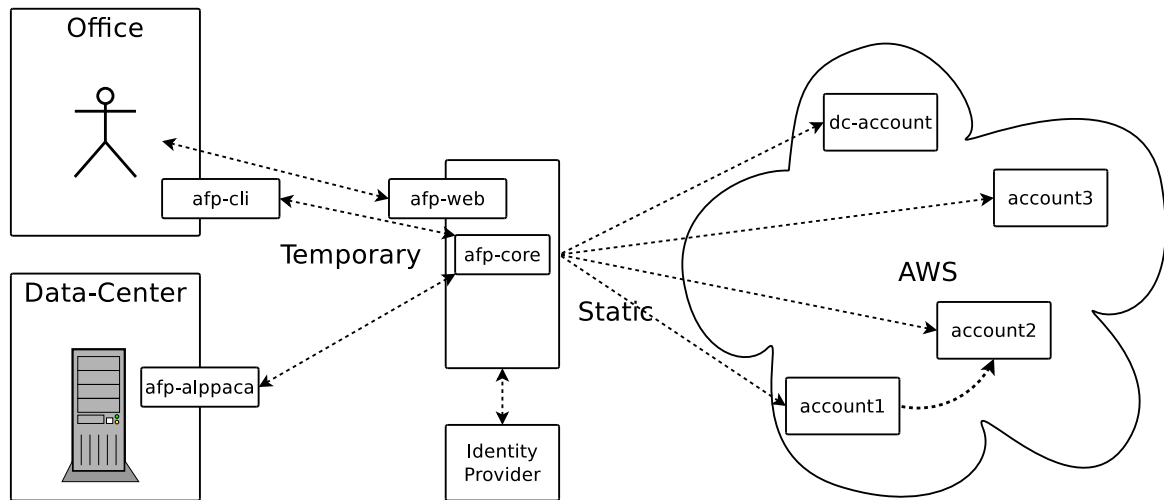
- Immobilienscout24
 - @IS24_Developer
-
- 2000 Machines in data-center
 - 17 Years in IT

- Senior System Developer
- Doing Python and Cloud
- @esc_

- 1 Introduction
- 2 Our Problem
- 3 afp-core
- 4 afp-web
- 5 afp-cli
- 6 afp-alppaca
- 7 Epilogue

- Connect the office to the cloud (no VPN)
- Making the 2000 machines in the data-center *cloud-ready*
- Easy and secure AWS service usage
- Temporary credentials for machines and humans





- *Identity and Access Management (IAM) Users*
 - Static Credentials
 - AWS_ACCESS_KEY_ID etc...
 - ssh-key
 - Basically anything w/o and expiry date
 - This is Bad™
-
- → Mine Bitcoins if lost

Multi-Account Paradigm

- We use multiple AWS accounts
- Goals:
 - Reduce blast radius in case of compromise
 - Modularise our data-center(s)
 - Feasible and incremental exit-strategy
- Rules:
 - Use well-defined interfaces
 - No VPN connections between accounts

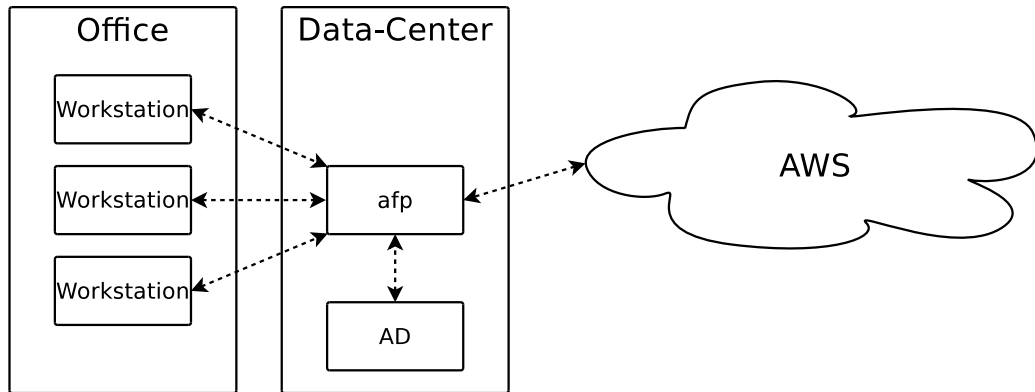
- 1 Introduction
- 2 Our Problem
- 3 **afp-core**
- 4 afp-web
- 5 afp-cli
- 6 afp-alppaca
- 7 Epilogue

- *AWS Federation Proxy (AFP)*
- *Custom Federation Broker*
- Different providers
 - Linux group provider (humans)
 - IP provider (machines)

- A webapplication
- One of the few places left that *does* have static credentials
- Uses *Secure Token Service* (STS) and *AssumeRole*

Connect *Active Directory* (AD) → AWS (Humans)

- Special groups in AD: `aws-<ACCOUNT>-<ROLE>`
 - AD users are members of such groups
 - E.g. `aws_account1_read-only`
 - → role: `read_only` in account `account1`
 - (Yes, the regex it is configurable)
-
- Note: in our case it is AD, but it could be anything
 - LDAP, Linux Users and Groups etc...



- 1 Introduction
- 2 Our Problem
- 3 afp-core
- 4 afp-web**
- 5 afp-cli
- 6 afp-alppaca
- 7 Epilogue

- Web-frontend for afp-core
 - Give an overview of all available accounts and their usable roles for the logged-in user
 - Get and show temporary credentials for the selected account and role
 - Automatic login to the AWS management console for the desired account
-
- See also: [Creating a URL that Enables Federated Users to Access the AWS Management Console \(Custom Federation Broker\)](#)

IMMOBILIEN
SCOUT 24

AWS Federation Proxy

Logged in as Valentin

Favorites

account1★
full-access: » Credentials » AWSConsole

account2★
full-access: » Credentials » AWSConsole

Accounts

account3★
full-access: » Credentials » AWSConsole

account4★
full-access: » Credentials » AWSConsole

account5★
full-access: » Credentials » AWSConsole

account6★
full-access: » Credentials » AWSConsole

AWS Federation Proxy WebUI is licensed under the [Apache License Version 2.0](#) . Version 2.0.0.is24

bash

raw

```
export AWS_ACCESS_KEY_ID=ASIAII2WL67DOVVBUQAQ
export AWS_SECRET_ACCESS_KEY=CQ+DrEOClc+mG+Pvrio3Nqx4L6M6s3l2ax1qEd75
export
AWS_SESSION_TOKEN=AQoDYXdzEEgakAIFyTJUI61MyQIZQqCm+wuyTuDnC5Etm6p8rtVfkNy6Fvg6xumjDbTSjt
oijp/jVopuu9WQwemlqDsdU0SVvQQ8qdVscvFFu8tTgrQ0gVxa5Dt58xBSOtHsBxHFE2BWat1RXuGed4l5HENwnrsxk
r5D8913+nafwwqiSpGgWgFzyVdvGBrEp55iyRobOuOC9ewGrBDufYVXMEMIn6
/jG8bGHvxqkePvoAsb3lY96E4pnVbKW7o8+xLaRre+a+lofXWn
/EoDI4SKT2pU6sTmbiHJes9l9Ruv2ndlTnhHEtYJq+tY1qUqQ6bcmjK6Ko+pyEiTuXUG9YgwO
/8o4OhDGivRrARYNbd8NDZn1ccFw3ceyCY18CzBQ==
export AWS_SECURITY_TOKEN=${AWS_SESSION_TOKEN}
```

Expires at 2015-12-15T16:02:48Z

Close

AWS Federation Proxy WebUI is licensed under the [Apache License Version 2.0](#) . Version 2.0.0.is24

- 1 Introduction
- 2 Our Problem
- 3 afp-core
- 4 afp-web
- 5 afp-cli**
- 6 afp-alppaca
- 7 Epilogue

- Console/shell frontend for afp-core
- List available accounts and roles
- Open subshell with exported AWS creds
- Alternatively: display or write

```
$ pip install afp-cli
```

- At our company: this just works!
- (Mainly because `https://afp` can be resolved)
 - Otherwise use:
 - → `index-url(config)`
 - → `--index-url(command-line)`

```
$ afp
account1      read-only
account2      access-all-areas, read-only
$ afp smarthome
...
(AWS: smarthome/access-all-areas 59 min) $ aws s3 ls
...
```

Alternatives → see README

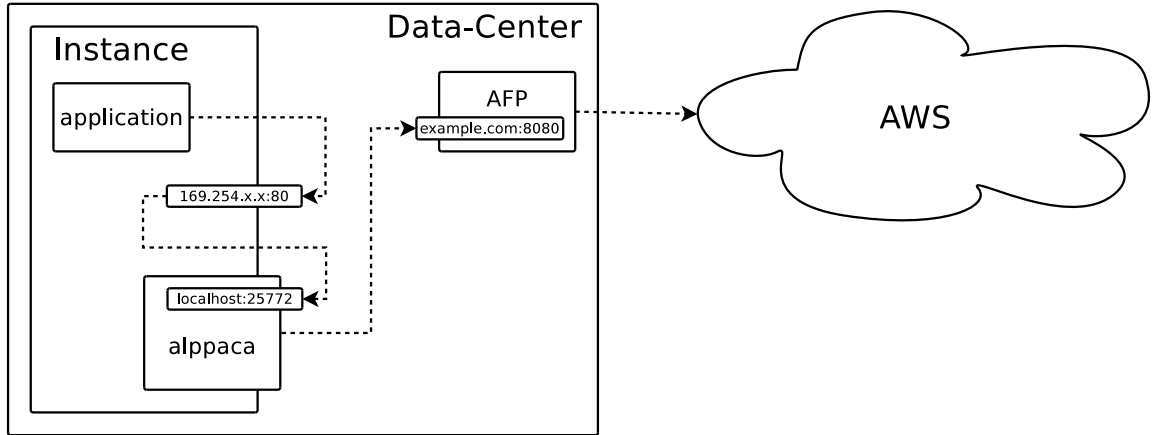
- 1 Introduction
- 2 Our Problem
- 3 afp-core
- 4 afp-web
- 5 afp-cli
- 6 afp-alppaca**
- 7 Epilogue

Instance Metadata Service (IMS)

- Available on 169.254.169.254.169:80 on every EC2 instance
- Provides information about the machines
- Also provides temporary security credentials

```
$ curl 169.254.169.254/latest/meta-data/iam/security-credentials/  
test_role  
$ curl 169.254.169.254/latest/meta-data/iam/security-credentials/test_role  
'{"Code": "Success",  
  "AccessKeyId": "ASIAI",  
  "SecretAccessKey": "oieDhF",  
  "Token": "6jmePdXNehjPVt7CZ1WMkKrqB6zDc34d2vpLej",  
  "Expiration": "2015-04-17T13:40:18Z",  
  "Type": "AWS-HMAC"}'
```

- *A Local Prefetch Proxy for Amazon Credentials (alppaca)*
 - Enables temporary credentials for machines
 - Provides an *Instance Metadata Service (IMS)* compatible interface
-
- → Machines in data-center are *cloud-ready*
 - (Yes, all 2000 of them can do this today!)



- Listens on 169.254.169.254:80
 - Uses Linux firewall magic (iptables)
- Prefetches temporary credentials and stores them in memory
- Always has a valid set of credentials
 - Always able to deliver them in sub 1 second
 - No changes to Amazon SDKs needed
- Roles are in the special dc-account

- `afp-alppaca` can be configured to perform an `AssumeRole` under the hood
- The data-center machine can appear to be in any one of our accounts
- Requires a role for that machine in that account and a *trusted principle* to enable the `AssumeRole` call

Further Uses: on EC2 in the Cloud

- Run afp-alppaca on an EC2 instance?
- Yes, to allow cross-account access

- More Linux magic needed...

- 1 Introduction
- 2 Our Problem
- 3 afp-core
- 4 afp-web
- 5 afp-cli
- 6 afp-alppaca
- 7 Epilogue**

- hashicorp/vault
- AdRoll/hologram
- You tell me?

Enrico Heine, Konrad Hosemann, Marco Hoyer, Michael Kuehne, Oliver Schoenherr, Ravi Yadav, Robert J. Will, Schlomo Schapiro, Sebastian Spoerer, Sebastian Stiehl, Sergej Domme, Stefan Neben, Stefan Nordhausen, Tobias Höynck, Tobias Vollmer, Valentin Haenel,

As per git-log on 15.12.2015

- All of it is open-source!
- <https://immobilienscout24.github.io/afp/>
- Questions?