Becoming a Command Line Expert with the AWS CLI

James Saryerwinnie, Amazon Web Services

November 14, 2013
AWS Command Line Interface
AWS Command Line Interface

Unified tool to manage your AWS services
One Tool
One Tool

Installation
One Tool

Installation

Configuration
One Tool

Installation

Configuration
Installation

Windows:
  32-bit MSI: http://s3.amazonaws.com/aws-cli/AWSCLI32.msi
  64-bit MSI: http://s3.amazonaws.com/aws-cli/AWSCLI64.msi

Bundled Installer

$ unzip awscli-bundle.zip
$ ./install
$ ~/.local/lib/aws/bin/aws --version
Pip

$ pip install --upgrade awsclli
One Tool

✓ Installation

Configuration
# Credential Configuration

<table>
<thead>
<tr>
<th>IAM Role</th>
<th>Environment</th>
<th>Config File: ~/.aws/config</th>
</tr>
</thead>
</table>
| Automatic  | AWS_ACCESS_KEY_ID
            | AWS_SECRET_ACCESS_KEY               | aws_access_key_id
            |                                   | aws_secret_access_key              |
Configuring

$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [json]:
Configuring

[default]
aws_access_key_id = EXAMPLE
aws_secret_access_key = EXAMPLEKEY
region = us-west-2
output = json
One Tool

✓ Installation

✓ Configuration
One Tool

✓ Installation

✓ Configuration
Demo
Let’s run a command
$ aws ec2 describe-instances
{
    "Reservations": []
}

$ aws ec2 describe-instances
{
    "Reservations": []
}

$ aws ec2 describe-instances
{
  "Reservations": []
}

$ aws ec2 describe-instances
{
    "Reservations": []
}

$ aws ec2 describe-instances
{
    "Reservations": []
}

$ aws ec2 describe-instances

service (command)  operation (subcommand)
Command with Arguments

$ aws ec2 import-key-pair \
   --key-name mykey \
   --public-key-material file:///home/user/.ssh/id_rsa.pub
Command with Arguments

$ aws ec2 import-key-pair \
   --key-name mykey \
   --public-key-material file:///home/user/.ssh/id_rsa.pub
Command with Arguments

$ aws ec2 import-key-pair
  --key-name mykey
  --public-key-material file://home/user/.ssh/id_rsa.pub
Feature

For any parameter value

- `file://<filename>`
- `http://<url>`
- `https://<url>`
How do I know what arguments to use?
$ aws ec2 create-security-group help

CREATE-SECURITY-GROUP()

NAME
create-security-group -

DESCRIPTION
The create-security-group operation creates a new security group.
Every instance is launched in a security group. If no security group is
specified during launch, the instances are launched in the default
security group. Instances within the same security group have unre-
stricted network access to each other. Instances will reject network
access attempts from other instances in a different security group. As
the owner of instances you can grant or revoke specific permissions
using the authorize-security-group-ingress and revoke-security-
group-ingress operations.

SYNOPSIS
create-security-group
[--dry-run | --no-dry-run]
--group-name <value>
--description <value>
[--vpc-id <value>]

OPTIONS
--dry-run | --no-dry-run (boolean)
$ aws ec2 create-security-group help

EXAMPLES
To create a security group
This example creates a security group named MySecurityGroup.
Command:

c    aws ec2 create-security-group --group-name MySecurityGroup --description "My security group"

Output:

{  "return": "true"
  "GroupId": "sg-9b380f8"}

create-security-group

Description

The create-security-group operation creates a new security group.

Every instance is launched in a security group. If no security group is specified during launch, the instances are launched in the default security group. Instances within the same security group have unrestricted network access to each other. Instances will reject network access attempts from other instances in a different security group. As the owner of instances you can grant or revoke specific permissions using the authorize-security-group-ingress and revoke-security-group-ingress operations.

Synopsis

```
create-security-group
[--dry-run | --no-dry-run]
--group-name <value>
```
Feature

Append help to any command

- `aws service operation help`
- `aws service help`
- `aws help`
Tab Completion
Bash  $ complete -C aws_completer aws
zsh   $ source bin/aws_zsh_completer.sh
tcsh $ complete aws 'p/*/`aws_completer`/''x
$ aws ec2 describe-instances
{
"Reservations": [
{
"OwnerId": "",
"ReservationId": "r-12345",
"Groups": [
{
"GroupName": "SSH",
"GroupId": "sg-abcdefg"
}
],
"Instances": [
{
"State": {
"Code": 16,
"Name": "running"
},
"KeyName": "mykey",
"InstanceType": "t1.micro",
}
]
},
...
}
## Output Formats

<table>
<thead>
<tr>
<th>JSON</th>
<th>Table</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmatic processing</td>
<td>Interactive browsing</td>
<td>Piping to text tools</td>
</tr>
<tr>
<td>Integrate with JSON tools</td>
<td>Easier to visually parse</td>
<td>Easy to parse</td>
</tr>
</tbody>
</table>
Feature

Multiple output formats

- --output json
- --output table
- --output text
Demo
Regions[ ].RegionName

```json
{
  "Regions": [
    {
      "Endpoint": "…",
      "RegionName": "eu-west-1"
    },
    {
      "Endpoint": "…",
      "RegionName": "us-east-1"
    }
  ]
}
```
Reservations[*].Instances[*].[InstanceId,State.Name]

```json
{"Reservations": [
  "Instances": [{
    "InstanceId": "i-1",
    "State": {"Name": "running"}
  }, {
    "InstanceId": "i-2",
    "State": {"Name": "stopped"}
  }]
]}
```

i-1     running
i-2     stopped
Reservations[*].Instances[*].{ID: InstanceId, State: State.Name}

```
{"Reservations": [
    "Instances": [
        {
            "InstanceId": "i-1",
            "State": {"Name": "running"}
        },
        {
            "InstanceId": "i-2",
            "State": {"Name": "stopped"}
        }
    ]}
```

<table>
<thead>
<tr>
<th>DescribeInstances</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>State</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>i-1</td>
<td>running</td>
</tr>
<tr>
<td>i-2</td>
<td>stopped</td>
</tr>
<tr>
<td>Data</td>
<td>Query</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td><code>{&quot;foo&quot;: &quot;bar&quot;}</code></td>
<td>foo</td>
</tr>
<tr>
<td><code>{&quot;foo&quot;: {&quot;bar&quot;: &quot;baz&quot;}}</code></td>
<td>foo.bar</td>
</tr>
<tr>
<td><code>{&quot;foo&quot;: [0, 1]}</code></td>
<td>foo[1]</td>
</tr>
<tr>
<td><code>{&quot;bar&quot;: 1, &quot;baz&quot;: 2}</code></td>
<td>foo or bar</td>
</tr>
<tr>
<td><code>{&quot;a&quot;: 1, &quot;b&quot;: 2, &quot;c&quot;: 3}</code></td>
<td>[a, b]</td>
</tr>
<tr>
<td><code>{&quot;a&quot;: 1, &quot;b&quot;: 2, &quot;c&quot;: 3}</code></td>
<td>{a: a, other: b}</td>
</tr>
<tr>
<td>[&quot;a&quot;: 1, &quot;a&quot;: 2}, {&quot;a&quot;: 3}, {&quot;a&quot;: 4}]</td>
<td>[*].a</td>
</tr>
</tbody>
</table>
Query response data

- Use `--query` to create the exact output you want.

aws ec2 describe-instances

... --filters Name=instance-state-name,Values=running
--filters (list)
A list of filters used to match properties for Instances. For a complete reference to the available filter keys for this operation, see the Amazon EC2 API reference.

**Shorthand Syntax:**

Key value pairs, where values are separated by commas.

```
--filters Name=string1,Values=string1,string2
```

**JSON Syntax:**

```
[
  {
    "Name": "string",
    "Values": ["string", ...]
  }
  ...
]
```
--filters (list)

A list of filters used to match properties for Instances. For a complete reference to the available filter keys for this operation, see the Amazon EC2 API reference.

**Shorthand Syntax:**

Key value pairs, where values are separated by commas.

--filters Name=string1,Values=string1,string2

**JSON Syntax:**

```json
[
  {
    "Name": "string",
    "Values": ["string", ...]
  }
  ...
]
```
--filters (list)
A list of filters used to match properties for Instances. For a complete reference to the available filter keys for this operation, see the Amazon EC2 API reference.

Shorthand Syntax:

Key value pairs, where values are separated by commas.
--filters Name=string1,Values=string1,string2

JSON Syntax:

[  
  {
    "Name": "string",
    "Values": ["string", ...]
  }
  ...
]
--filters

Name=instance-state-name,Values=running

[{
  "Name": "instance-state-name",
  "Values": ["running"]
}]
Feature

Shorthand Syntax

• Use Shorthand Syntax to specify parameter values
Large Responses
--max-items 3500
--max-items 3500 --starting-token <blob>
Feature

Pagination

Use --starting-token and --max-items to paginate results
We've Learned

- `file://`
- `aws help`
- Tab Completion
- `--output`
- `--query`
- Shorthand Syntax
- Pagination
Static Blog with Amazon S3

- Create blog locally
- Sync to Amazon S3
- AWS Identity and Access Management (IAM)
- Amazon Route53
Amazon S3

$ aws s3 mb s3://www.reinvent-cli-blog-demo.com/
$ aws s3 website www.reinvent-cli-blog-demo.com \--index-document index.html
Create a user that has access only to the static blog

Full access only to the www.reinvent-cli-blog-demo.com bucket
AWS IAM

$ aws iam create-user --user-name static-blog

{
    "User": {
        "UserName": "static-blog",
        "Path": "/",
        "CreateDate": "2013-10-18T18:46:39.044Z",
        "UserId": "EXAMPLEUSERID",
        "Arn": "arn:aws:iam::12345:user/static-blog"
    }
}
AWS IAM

$ aws iam create-access-key --user-name static-blog
{
    "AccessKey": {
        "UserName": "static-blog",
        "Status": "Active",
        "CreateDate": "2013-10-18T18:47:38.913Z",
        "SecretAccessKey": "SECRET_KEY",
        "AccessKeyId": "ACCESS_KEY"
    }
}

AWS IAM

$ aws iam put-user-policy --user-name static-blog \
   --policy-name static-blog-s3-access \
   --policy-document file://singlebucket.json
AWS IAM

$ aws iam put-user-policy --user-name static-blog
   --policy-name static-blog-s3-access
   --policy-document file://singlebucket.json
AWS IAM

{
   "Statement": [
       {
           "Sid": "Stmt12345678",
           "Action": [
               "s3:*"
           ],
           "Effect": "Allow",
           "Resource": [
               "arn:aws:s3:::www.reinvent-cli-blog-demo.com/*",
               "arn:aws:s3:::www.reinvent-cli-blog-demo.com"
           ]
       }
   ]
}
How do I tell the CLI about this new user?
aws configure --profile staticblog
[default]
...

[profile staticblog]
aws_access_key = ACCESS_KEY
aws_secret_access_key = SECRET_KEY
region = us-west-2
Pushing Blog Content

```bash
aws s3 sync . s3://www.reinvent-cli-blog-demo.com/ 
  --acl public-read 
  --delete 
  --profile staticblog
```
Pushing Blog Content

```plaintext
aws s3 sync . s3://www.reinvent-cli-blog-demo.com/ \
    --acl public-read \ 
    --delete \ 
    --profile staticblog
```
Pushing Blog Content

```bash
aws s3 sync . s3://www.reinvent-cli-blog-demo.com/ \
  --acl public-read \
  --delete \
  --profile staticblog
```
Pushing Blog Content

aws s3 sync . s3://www.reinvent-cli-blog-demo.com/ \
    --acl public-read \
    --delete \
    --profile staticblog
Pushing Blog Content

```bash
aws s3 sync . s3://www.reinvent-cli-blog-demo.com/ \
  --acl public-read \
  --delete \
  --profile staticblog
```
Amazon S3 Sync
Amazon S3 Sync
Amazon S3 Sync

- Sync new/changed files
- Files uploaded in parallel
- Large files split into chunks
Amazon Route53
Amazon Route53

```bash
$ aws route53 create-hosted-zone --name www.reinvent-cli-blog-demo.com \\
--caller-reference reinvent-cli-blog-demo
{
    "HostedZone": {
        "Id": "/hostedzone/Z1TI9W0V4R87XY",
        "Name": "www.reinvent-cli-blog-demo.com"
    },
    "DelegationSet": {
        "NameServers": [
            "ns-abc.awsdns-20.com",
            "ns-abcd.awsdns-49.org"
        ]
    }
}
```
Amazon Route53

aws route53 change-resource-record-sets --hosted-zone-id "/hostedzone/Z1TI9W0V4R87XY" \
--change-batch file://changebatch.json

{
  "ChangeInfo": {
    "Status": "PENDING",
    "Comment": "Add S3 Bucket",
    "SubmittedAt": "2013-10-31T18:37:34.281Z",
    "Id": "/change/C1AG4RL3JT78JG"
  }
}
Amazon Route53

file://changebatch.json
{
  "Comment": "Add S3 Bucket",
  "Changes": [
  {
    "Action": "CREATE",
    "ResourceRecordSet": {
      "Name": "www.reinvent-cli-blog-demo.com",
      "Type": "A",
      "AliasTarget": {
        "HostedZoneId": "Z3BJ6K6RIION7M",
        "EvaluateTargetHealth": false,
        "DNSName": "s3-website-us-west-2.amazonaws.com"
      }
    }
  }
  ]
}
Demo
Additional Topics
BotoCore

_regions.json  ec2.json  redshift.json
_retry.json  elasticache.json  route53.json
autoscaling.json  elasticbeanstalk.json  s3.json
cloudformation.json  elastictranscoder.json  ses.json
cloudfront.json  elb.json  sns.json
cloudsearch.json  emr.json  sqs.json
cloudwatch.json  iam.json  storagegateway.json
datapipeline.json  importexport.json  sts.json
directconnect.json  opsworks.json  support.json
dynamodb.json  rds.json  swf.json
"api_version": "2013-10-01",
"type": "query",
"signature_version": "v2",
"service_full_name": "Amazon Elastic Compute Cloud",
"service_abbreviation": "Amazon EC2",
"endpoint_prefix": "ec2",
"operations": {
  "ActivateLicense": {
    "name": "ActivateLicense",
    "input": {...},
    "output": {...},
  }
}
"dynamodb": {
    "__default__": {
        "max_attempts": 10,
        "delay": {
            "type": "exponential",
            "base": 0.05,
            "growth_factor": 2
        }
    },
    "policies": {
        "throughput_exceeded": {
            "applies_when": {
                "response": {
                    "service_error_code": "ProvisionedThroughputExceededException",
                    "http_status_code": 400
                }
            }
        }
    }
}
Plugins
Next Steps

• Check out the code on Github: [http://github.com/aws/aws-cli](http://github.com/aws/aws-cli)
• Ask questions on our Forums: [https://forums.aws.amazon.com/forum.jspa?forumID=150](https://forums.aws.amazon.com/forum.jspa?forumID=150)
• Chat with us in the Developer Lounge (Boto 1:30pm, CLI 3:30pm)
Please give us your feedback on this presentation

TLS304

As a thank you, we will select prize winners daily for completed surveys!

Thank You