Authentication

In the AWS Console, create an IAM user and click "Download .csv". Then, configure interactively:

```
aws configure
```

Or, configure non-interactively:

```
aws configure set aws_access_key_id $(cat
new_user_credentials.csv | sed -n 2p | awk 'BEGIN
{ FS = "," } ; { printf "%s", $1 }')
aws configure set aws_secret_access_key $(cat
new_user_credentials.csv | sed -n 2p | awk 'BEGIN
{ FS = "," } ; { printf "%s", $2 }')
```

Browser-based authentication for Azure and GCP:

```
az login
gcloud auth login
```

GCP authentication with a JSON key file:

```
gcloud auth activate-service-account
--key-file <Path to your key file>
gcloud config set project $(cat <Path to your key
file> | jq -r ".project_id")
```

Show the signed-in user:

```
aws sts get-caller-identity
az ad signed-in-user show
gcloud auth list
```

SSH to a Public Cloud Virtual Machine

```
ssh ubuntu@$(aws ec2 describe-instances --filters
Name=instance-state-name,Values=running Name=tag-
value,Values=<Your instance name> --query
"Reservations[0].Instances[0].PublicIpAddress" --
output text)

ssh ubuntu@$(az vm list-ip-addresses --query
"[?virtualMachine.name=='<Your VM
name>'].virtualMachine.network.publicIpAddresses[0].
ipAddress" --output tsv)

gcloud compute ssh <Your VM name> --ssh-key-file
~/.ssh/id_rsa
```

Filtering and Querying

AWS

All AWS commands support a **query** option. This specifies a JMESPath string to extract a portion of the output:

```
aws iam list-users --query
"Users[0].UserName" # "cloudsecurity"
```

Azure

Azure CLI commands support a **query** option identical to AWS's:

```
az network vnet list
--query '[0].subnets[0].addressPrefix'
# "10.0.0.0/24"
```

GCP

While GCP commands do not support data extraction via a **query** option, they support a **filter** option. This will make the command only return items that match the provided Boolean expression:

```
gcloud sql instances list -filter
'name = <Instance name> AND
serverCaCert.expirationTime.date("%Y") >=
"2020"' # Table of matching instances
```

jq

When the built-in filtering and querying capabilities fall short, you can process and create JSON with **jq**:

```
aws s3api list-buckets | jq -r '.Buckets[]
| select(.Name | startswith("sec510"))'

gcloud projects list --format json |
jq -r '.[] |
select(.lifecycleState=="ACTIVE").name'
# <Your active project's name>
```

SANS CLOUD SECURITY

MULTICLOUD COMMAND-LINE INTERFACE

By Brandon Evans

Cheat Sheet v1.2.1.2

SANS.ORG/CLOUD-SECURITY

Use CLIs to interact with the three most popular cloud platforms: Amazon Web Services (AWS), Microsoft Azure, and the Google Cloud Platform (GCP).

CLI Version Details

All commands, unless stated otherwise, have been tested in the SEC510 course VM using the following CLI versions:

```
aws --version     # aws-cli/2.0.35
Python/3.7.3 Linux/4.15.0-58-generic
botocore/2.0.0dev39

az --version      # azure-cli 2.2.0
gcloud --version  # Google Cloud SDK 286.0.0
gsutil --version  # gsutil version: 4.48
jq --version      # jq-1.5-1-a5b5cbe
```

You must be authenticated and have the appropriate Identity and Access Management (IAM) permissions to run these commands.

MULTIPLE CLOUDS REQUIRE MULTIPLE SOLUTIONS

Enumerate Contents of Storage

Enumerate all buckets or storage accounts in an account:

```
aws s3 ls s3://
az storage account list
gsutil ls gs://
```

Enumerate all containers in an Azure storage account:

```
az storage container list --account-name
<Your storage account name>
```

Enumerate all objects or blobs in a bucket or container:

```
aws s3 ls s3://<Your bucket name>

az storage blob list --account-name <Your
storage account name> --container-name
<Your container name>

gsutil ls gs://<Your bucket name>
```

Upload and Download Files from Storage

Uploading

```
aws s3 cp file.txt s3://<Your bucket name>

az storage blob upload --account-name <Your
storage account name> --container-name <Your
container name> --name file.txt --file file.txt

gsutil cp file.txt gs://<Your bucket name>
```

Downloading

```
aws s3 cp s3://<Your bucket name>/file.txt .

az storage blob download --account-name <Your
storage account name> --container-name <Your
container name> --name file.txt --file file.txt

gsutil cp gs://<Your bucket name>/file.txt .
```

Encrypt and Decrypt Data

AWS

```
aws kms encrypt --key-id <Your key ARN or
alias> --plaintext SANS | jq -r
'.CiphertextBlob' | base64 -d > encrypted.txt

aws kms decrypt --key-id <Your key ARN or
alias> --ciphertext-blob fileb://encrypted.txt
| jq -r '.Plaintext' # SANS
```

Azure

Azure Key Vault only supports asymmetric encryption. `az keyvault key encrypt` and `decrypt` were added in version 2.8.0 on June 23rd, 2020. These commands were tested for that version on macOS:

```
az keyvault key encrypt --algorithm RSA1_5 --
vault-name <Your Key Vault name>
--name <Your key name> --value SANS
| jq -r '.result' > encrypted.txt

az keyvault key decrypt --algorithm RSA1_5 --
vault-name <Your Key Vault name>
--name <Your key name> --value "$(cat
encrypted.txt)" | jq -r '.result' # SANS
```

GCP

```
echo "SANS" > plaintext.txt
gcloud kms encrypt --plaintext-file
plaintext.txt --ciphertext-file encrypted.txt
--keyring <Your keyring name>
--location <Your location, such as us-
central1> --key <Your key name>

gcloud kms decrypt --plaintext-file new-
plaintext.txt --ciphertext-file encrypted.txt
--keyring <Your keyring name>
--location <Your location, such as us-
central1> --key <Your key name>

cat new-plaintext.txt # SANS
```

Alternative Cryptography Commands for Azure

Here are alternative commands supported in 2.2.0:

```
export REQUEST_BODY='{"alg": "RSA1_5", "value":
"SANS"}'
az rest --resource https://vault.azure.net --method
POST --headers "Content-Type=application/json" --
uri "https://<Your Key Vault
name>.vault.azure.net/keys/<Your key
name>/encrypt?api-version=7.0" --body
"$REQUEST_BODY" | jq -r '.value' > encrypted.txt

export REQUEST_BODY=$(echo '{"alg": "RSA1_5"}' | jq
--arg value value ". + {value: \"$(cat
encrypted.txt)\"}")
az rest --resource https://vault.azure.net --method
POST --headers "Content-Type=application/json" --
uri "https://<Your Key Vault
name>.vault.azure.net/keys/<Your key
name>/decrypt?api-version=7.0" --body
"$REQUEST_BODY" | jq -r '.value' # SANS
```

Other Tips and Tricks

- The AWS Systems Manager Session Manager can establish shell sessions to private EC2 instances: `aws ssm start-session`
- The Azure API can be invoked using HTTP requests with `az rest` These will use the same credentials used with all other commands.
- Delete the default GCP firewall rules: `gcloud compute firewall-rules delete default-allow-<Repeat for icmp, rdp, ssh, and internal>`