AWS CLI Cheatsheet

http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html https://www.youtube.com/watch?v=_wiGpBQGCjU

Setup

Overview

- Virtualbox
- Ubuntu 14.04 LTS VM, 64-bit http://releases.ubuntu.com/14.04/ubuntu-14.04.4-desktop-amd64.iso
- create new machine, settings
    - System / Processor
    - Enable PAE/NX
    - System / Acceleration
    - Paravirtualization Interface: Default
    - Enable VT-x/AMD-V
    - Enable Nested Paging
    - Display / Screen
    - Video Memory: 128MB
    - Acceleration: Enable 3D Acceleration
- boot
- install

install Virtualbox Guest Additions, passwordless sudo

```
echo $USER
sudo echo "$USER ALL=(ALL) NOPASSWD:ALL" | sudo tee -a /etc/sudoers
sudo su
apt-get update
apt-get install -y build-essential dkms linux-headers-$(uname -r)
cd /media/aws-admin/
sh ./VBoxLinuxAdditions.run
shutdown now
```

install AWS CLI

```
sudo apt-get install -y python-dev python-pip
sudo pip install awscli
aws --version
aws configure
```

Bash one-liners

```
cat <file> # output a file
tee # split output into a file
cut -f 2 # print the 2nd column, per line
sed -n '5{p;q}' # print the 5th line in a file
sed 1d # print all lines, except the first
tail -n +2 # print all lines, starting on the 2nd
head -n 5 # print the first 5 lines
tail -n 5 # print the last 5 lines

expand # convert tabs to 4 spaces
unexpand -a # convert 4 spaces to tabs
wc # word count
tr ' ' \\t # translate / convert characters to other characters

sort # sort data
uniq # show only unique entries
paste # combine rows of text, by line
join # combine rows of text, by initial column value
```

Cloudtrail - Logging and Auditing

http://docs.aws.amazon.com/cli/latest/reference/cloudtrail/ 5 Trails total, with support for resource level permissions

```
# list all trails
aws cloudtrail describe-trails

# list all S3 buckets
aws s3 ls

# create a new trail
aws cloudtrail create-subscription \
    --name awslog \
    --s3-new-bucket awslog2016

# list the names of all trails
aws cloudtrail describe-trails --output text | cut -f 8

# get the status of a trail
aws cloudtrail get-trail-status \
    --name awslog

# delete a trail
aws cloudtrail delete-trail \
    --name awslog

# delete the S3 bucket of a trail
aws s3 rb s3://awslog2016 --force

# add tags to a trail, up to 10 tags
aws cloudtrail add-tags \
    --resource-id awslog \
    --tags-list "Key=log-type,Value=all"

# list the tags of a trail
aws cloudtrail list-tags \
    --resource-id-list

# remove a tag from a trail
aws cloudtrail remove-tags \
    --resource-id awslog \
    --tags-list "Key=log-type,Value=all"
```

IAM

Users

https://blogs.aws.amazon.com/security/post/Tx15CIT22V4J8RP/How-to-rotate-access-keys-for-IAM-users http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-limits.html Limits = 5000 users, 100 group, 250 roles, 2 access keys / user

http://docs.aws.amazon.com/cli/latest/reference/iam/index.html

```
# list all user's info
aws iam list-users

# list all user's usernames
aws iam list-users --output text | cut -f 6

# list current user's info
aws iam get-user

# list current user's access keys
aws iam list-access-keys

# crate new user
aws iam create-user \
    --user-name aws-admin2

# create multiple new users, from a file
allUsers=$(cat ./user-names.txt)
for userName in $allUsers; do
    aws iam create-user \
        --user-name $userName
done

# list all users
aws iam list-users --no-paginate

# get a specific user's info
aws iam get-user \
    --user-name aws-admin2

# delete one user
aws iam delete-user \
    --user-name aws-admin2

# delete all users
# allUsers=$(aws iam list-users --output text | cut -f 6);
allUsers=$(cat ./user-names.txt)
for userName in $allUsers; do
    aws iam delete-user \
        --user-name $userName
done
```

Password policy

http://docs.aws.amazon.com/cli/latest/reference/iam/

```
# list policy
# http://docs.aws.amazon.com/cli/latest/reference/iam/get-account-password-policy.html
aws iam get-account-password-policy

# set policy
# http://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html
aws iam update-account-password-policy \
    --minimum-password-length 12 \
    --require-symbols \
    --require-numbers \
    --require-uppercase-characters \
    --require-lowercase-characters \
    --allow-users-to-change-password

# delete policy
# http://docs.aws.amazon.com/cli/latest/reference/iam/delete-account-password-policy.html
aws iam delete-account-password-policy
```

Access Keys

http://docs.aws.amazon.com/cli/latest/reference/iam/

```
# list all access keys
aws iam list-access-keys

# list access keys of a specific user
aws iam list-access-keys \
    --user-name aws-admin2

# create a new access key
aws iam create-access-key \
    --user-name aws-admin2 \
    --output text | tee aws-admin2.txt

# list last access time of an access key
aws iam get-access-key-last-used \
    --access-key-id AKIAINA6AJZY4EXAMPLE

# deactivate an acccss key
aws iam update-access-key \
    --access-key-id AKIAI44QH8DHBEXAMPLE \
    --status Inactive \
    --user-name aws-admin2

# delete an access key
aws iam delete-access-key \
    --access-key-id AKIAI44QH8DHBEXAMPLE \
    --user-name aws-admin2
```

Groups, Policies, Managed Policies

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html http://docs.aws.amazon.com/cli/latest/reference/iam/

```
# list all groups
aws iam list-groups

# create a group
aws iam create-group --group-name FullAdmins

# delete a group
aws iam delete-group \
    --group-name FullAdmins

# list all policies
```

```
aws iam list-policies

# get a specific policy
aws iam get-policy \
    --policy-arn <value>

# list all users, groups, and roles, for a given policy
aws iam list-entities-for-policy \
    --policy-arn <value>

# list policies, for a given group
aws iam list-attached-group-policies \
    --group-name FullAdmins

# add a policy to a group
aws iam attach-group-policy \
    --group-name FullAdmins \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

# add a user to a group
aws iam add-user-to-group \
    --group-name FullAdmins \
    --user-name aws-admin2

# list users, for a given group
aws iam get-group \
    --group-name FullAdmins

# list groups, for a given user
aws iam list-groups-for-user \
    --user-name aws-admin2

# remove a user from a group
aws iam remove-user-from-group \
    --group-name FullAdmins \
    --user-name aws-admin2

# remove a policy from a group
aws iam detach-group-policy \
    --group-name FullAdmins \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

# delete a group
aws iam delete-group \
    --group-name FullAdmins
```

S3

https://docs.aws.amazon.com/cli/latest/reference/s3api/index.html#cli-aws-s3api

```
# list existing S3 buckets
aws s3 ls

# create a bucket name, using the current date timestamp
bucket_name=test_$(date "+%Y-%m-%d_%H-%M-%S")
echo $bucket_name

# create a public facing bucket
aws s3api create-bucket --acl "public-read-write" --bucket $bucket_name

# verify bucket was created
aws s3 ls | grep $bucket_name

# check for public facing s3 buckets (should show the bucket name you created)

aws s3api list-buckets --query 'Buckets[*].[Name]' --output text | xargs -I {} bash -c 'if [[ $(aws s3api get-bucket-acl --bucket {} --query '"'"'Grants[?Grantee.URI==`http://acs.amazonaws.com/groups/global/AllUsers` && Permission==`R

# check for public facing s3 buckets, updated them to be private

aws s3api list-buckets --query 'Buckets[*].[Name]' --output text | xargs -I {} bash -c 'if [[ $(aws s3api get-bucket-acl --bucket {} --query '"'"'Grants[?Grantee.URI==`http://acs.amazonaws.com/groups/global/AllUsers` && Permission==`R

# check for public facing s3 buckets (should be empty)

aws s3api list-buckets --query 'Buckets[*].[Name]' --output text | xargs -I {} bash -c 'if [[ $(aws s3api get-bucket-acl --bucket {} --query '"'"'Grants[?Grantee.URI==`http://acs.amazonaws.com/groups/global/AllUsers` && Permission==`R
```

EC2

keypairs

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html

```
# list all keypairs
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-key-pairs.html
aws ec2 describe-key-pairs

# create a keypair
# http://docs.aws.amazon.com/cli/latest/reference/ec2/create-key-pair.html
aws ec2 create-key-pair \
    --key-name <value> --output text

# create a new local private / public keypair, using RSA 4096-bit
ssh-keygen -t rsa -b 4096

# import an existing keypair
# http://docs.aws.amazon.com/cli/latest/reference/ec2/import-key-pair.html
aws ec2 import-key-pair \
    --key-name keyname_test \
    --public-key-material file:///home/apollo/id_rsa.pub

# delete a keypair
# http://docs.aws.amazon.com/cli/latest/reference/ec2/delete-key-pair.html
aws ec2 delete-key-pair \
    --key-name <value>
```

Security Groups

http://docs.aws.amazon.com/cli/latest/reference/ec2/index.html

```
# list all security groups
aws ec2 describe-security-groups

# create a security group
aws ec2 create-security-group \
    --vpc-id vpc-1a2b3c4d \
    --group-name web-access \
    --description "web access"

# list details about a securty group
aws ec2 describe-security-groups \
    --group-id sg-0000000

# open port 80, for everyone
aws ec2 authorize-security-group-ingress \
    --group-id sg-0000000 \
    --protocol tcp \
    --port 80 \
    --cidr 0.0.0.0/24

# get my public ip
my_ip=$(dig +short myip.opendns.com @resolver1.opendns.com);
echo $my_ip

# open port 22, just for my ip
aws ec2 authorize-security-group-ingress \
    --group-id sg-0000000 \
    --protocol tcp \
    --port 80 \
    --cidr $my_ip/24

# remove a firewall rule from a group
aws ec2 revoke-security-group-ingress \
    --group-id sg-0000000 \
    --protocol tcp \
    --port 80 \
    --cidr 0.0.0.0/24

# delete a security group
aws ec2 delete-security-group \
    --group-id sg-00000000
```

Images

https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-images.html

```
# list all private AMI's, ImageId and Name tags
aws ec2 describe-images --filter "Name=is-public,Values=false" \
    --query 'Images[].[ImageId, Name]' \
    --output text | sort -k2

# delete an AMI, by ImageId
aws ec2 deregister-image --image-id ami-00000000
```

Instances

http://docs.aws.amazon.com/cli/latest/reference/ec2/index.html

```
# list all instances (running, and not running)
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instances.html
aws ec2 describe-instances

# list all instances running
aws ec2 describe-instances --filters Name=instance-state-name,Values=running

# create a new instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html
aws ec2 run-instances \
    --image-id ami-f0e7d19a \
    --instance-type t2.micro \
    --security-group-ids sg-00000000 \
    --dry-run

# stop an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/terminate-instances.html
aws ec2 terminate-instances \
    --instance-ids <instance_id>

# list status of all instances
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-status.html
aws ec2 describe-instance-status

# list status of a specific instance
aws ec2 describe-instance-status \
    --instance-ids <instance_id>

# list all running instance, Name tag and Public IP Address
aws ec2 describe-instances \
  --filters Name=instance-state-name,Values=running \
  --query 'Reservations[].Instances[].[PublicIpAddress, Tags[?Key==`Name`].Value | [0] ]' \
  --output text | sort -k2
```

Tags

```
# list the tags of an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-tags.html
aws ec2 describe-tags

# add a tag to an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/create-tags.html
aws ec2 create-tags \
    --resources "ami-1a2b3c4d" \
    --tags Key=name,Value=debian

# delete a tag on an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/delete-tags.html
aws ec2 delete-tags \
    --resources "ami-1a2b3c4d" \
    --tags Key=Name,Value=
```

Cloudwatch

Log Groups

http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/WhatIsCloudWatchLogs.html http://docs.aws.amazon.com/cli/latest/reference/logs/index.html#cli-aws-logs

create a group

http://docs.aws.amazon.com/cli/latest/reference/logs/create-log-group.html

```
aws logs create-log-group \
    --log-group-name "DefaultGroup"
```

list all log groups

http://docs.aws.amazon.com/cli/latest/reference/logs/describe-log-groups.html

```
aws logs describe-log-groups

aws logs describe-log-groups \
    --log-group-name-prefix "Default"
```

delete a group

http://docs.aws.amazon.com/cli/latest/reference/logs/delete-log-group.html

```
aws logs delete-log-group \
    --log-group-name "DefaultGroup"
```

Log Streams

```

# Log group names can be between 1 and 512 characters long. Allowed
# characters include a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen),
# '/' (forward slash), and '.' (period).

# create a log stream
# http://docs.aws.amazon.com/cli/latest/reference/logs/create-log-stream.html
aws logs create-log-stream \
    --log-group-name "DefaultGroup" \
    --log-stream-name "syslog"

# list details on a log stream
# http://docs.aws.amazon.com/cli/latest/reference/logs/describe-log-streams.html
aws logs describe-log-streams \
    --log-group-name "syslog"

aws logs describe-log-streams \
    --log-stream-name-prefix "syslog"

# delete a log stream
# http://docs.aws.amazon.com/cli/latest/reference/logs/delete-log-stream.html
aws logs delete-log-stream \
    --log-group-name "DefaultGroup" \
    --log-stream-name "Default Stream"
```

Cloudwatch - Monitoring

http://docs.aws.amazon.com/cli/latest/reference/cloudwatch/index.html