



# *CLI QUICK REFERENCE*

**EC2**

# INSTANCES

- list all instances (running, and not running): `aws ec2 describe-instances`
- list all instances running: `aws ec2 describe-instances --filters Name=instance-state-name,Values=running`
- create a new instance: `aws ec2 run-instances --image-id ami-a0b1234 --instance-type t2.micro --security-group-ids sg-000000000 --dry-run`
- stop an instance: `aws ec2 terminate-instances --instance-ids <instance_id>`
- list status of all instances: `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`

# SECURITY GROUP

- 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-server --description "web server access"`
- list details about a security group: `aws ec2 describe-security-groups --group-id sg-00000000`
- open port 80 for everyone: `aws ec2 authorize-security-group-ingress --group-id sg-00000000 --protocol tcp --port 80 --cidr 0.0.0.0`
- 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-00000000 --protocol tcp --port 80 --cidr $my_ip/24`
- remove a firewall rule from a group: `aws ec2 revoke-security-group-ingress --group-id sg-00000000 --protocol tcp --port 80 --cidr 0.0.0.0/24`
- delete a security group: `aws ec2 delete-security-group --group-id sg-00000000`

# IMAGES

- 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`
- delete an AMI, by ImageId: `aws ec2 deregister-image --image-id ami-000000000`

# TAGS

- list the tags of an instance: `aws ec2 describe-tags`
- add a tag to an instance: `aws ec2 create-tags --resources "ami-1a2b3c4d" --tags Key=name, Value=debian`
- delete a tag on an instance: `aws ec2 delete-tags --resources "ami-1a2b3c4d" --tags Key=Name, Value=`

# KEYPAIRS

- list all keypairs: `aws ec2 describe-key-pairs`
- create a keypair: `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: `aws ec2 import-key-pair --key-name keyname_test --public-key-material file:///home/rkumar/id_rsa.pub`
- delete a keypair: `aws ec2 delete-key-pair --key-name <value>`