

Install AWS Cli

- aws <https://aws.amazon.com/cli/>

Table of Contents

- [EC2](#)
 - [List Instance ID, Type and Name](#)
 - [List Instances with Public IP Address and Name](#)
 - [List of VPCs and CIDR IP Block](#)
 - [List of Subnets for a VPC](#)
 - [List of Security Groups](#)
 - [Print Security Groups for an Instance](#)
 - [Edit Security Groups of an Instance](#)
 - [Print Security Group Rules as FromAddress and ToPort](#)
 - [Add Rule to Security Group](#)
 - [Delete Rule from Security Group](#)
 - [Edit Rules of Security Group](#)
 - [Delete Security Group](#)
- [S3](#)
 - [List Buckets](#)
 - [List Files in a Bucket](#)
 - [Create Bucket](#)
 - [Delete Bucket](#)
 - [Download S3 Object to Local](#)
 - [Upload Local File as S3 Object](#)
 - [Delete S3 Object](#)
 - [Download Bucket to Local](#)
 - [Upload Local Directory to Bucket](#)
 - [Share S3 Object without Public Access](#)
- [API Gateway](#)
 - [List of API Gateway IDs and Names](#)
 - [List of API Gateway Keys](#)
 - [List API Gateway Domain Names](#)
 - [List of Resources for API Gateway](#)
 - [Find Lambda for API Gateway Resource](#)
- [ELB](#)
 - [List of ELB Hostnames](#)
 - [List of ELB ARNs](#)
 - [List of ELB Target Group ARNs](#)
 - [Find Instances for a Target Group](#)
- [RDS](#)
 - [List of DB Clusters](#)
 - [List of DB Instances](#)

- [Take DB Instance Snapshot](#)
 - [Take DB Cluster Snapshot](#)
- [ElastiCache](#)
 - [List of ElastiCache Machine Type and Name](#)
 - [List of ElastiCache Replication Groups](#)
 - [List of ElastiCache Snapshots](#)
 - [Create ElastiCache Snapshot](#)
 - [Delete ElastiCache Snapshot](#)
 - [Scale Up/Down ElastiCache Replica](#)
- [Lambda](#)
 - [List of Lambda Functions, Runtime and Memory](#)
 - [List of Lambda Layers](#)
 - [List of Source Event for Lambda](#)
 - [Download Lambda Code](#)
- [Cloudwatch](#)
 - [List of CloudWatch Alarms and Status](#)
 - [Create Alarm for EC2 High CPUUtilization](#)
 - [Create Alarm for EC2 High StatusCheckFailed_Instance](#)
- [Route53](#)
 - [List Domains](#)
 - [List Records for a Domain \(Zone\)](#)
- [SNS](#)
 - [List of SNS Topics](#)
 - [List of SNS Topic and related Subscriptions](#)
 - [Publish to SNS Topic](#)
- [DynamoDB](#)
 - [List of DynamoDB Tables](#)
 - [Get All Items from a Table](#)
 - [Get Item Count from a Table](#)
 - [Get Item using Key](#)
 - [Get Specific Fields from an Item](#)
 - [Delete Item using Key](#)
- [SQS](#)
 - [List Queues](#)
 - [Create Queue](#)
 - [Count Messages in Queue](#)
 - [Send Message](#)
 - [Receive Message](#)
 - [Delete Message](#)
 - [Purge Queue](#)
 - [Delete Queue](#)
- [CloudFront](#)
 - [List of CloudFront Distributions and Origins](#)
 - [Create Cache Invalidation](#)
 - [Check Cache Invalidation Status](#)
- [Amplify](#)
 - [List of Amplify Apps and Source Repository](#)

- [Cognito](#)
 - [List of User Pool IDs and Names](#)
 - [List of Phone and Email of All Users](#)
- [IAM User](#)
 - [List of UserId and UserName](#)
 - [Get Single User](#)
 - [Add User](#)
 - [Delete User](#)
 - [List Access Keys for User](#)
 - [Delete Access Key for User](#)
 - [Activate/Deactivate Access Key for User](#)
 - [Generate New Access Key for User](#)
- [IAM Group](#)
 - [List Groups](#)
 - [Add/Delete Groups](#)
 - [List of Policies and ARNs](#)
 - [List of User/Group/Roles for a Policy](#)
 - [List Policies for a Group](#)
 - [Add Policy to a Group](#)
 - [Add User to a Group](#)
 - [Remove User from a Group](#)
 - [List Users in a Group](#)
 - [List Groups for a User](#)
 - [Attach/Detach Policy to a Group](#)

Pro Tip!

👉 If you have multiple AWS Accounts, you can use bash alias like the following. So you no longer need to pass `--profile` to `aws` tool.

```
alias aws-prod="aws --profile work-prod"
alias aws-dev="aws --profile work-dev"
alias aws-self="aws --profile personal"
alias aws="aws --profile work-dev"
```

👉 To format `aws` command output into tables, you can pipe output to `column -t`.

```
# aws ec2 describe-instances | jq ...
i-0f112d652ecf13dac c3.x2large fisher.com
i-0b3b5128445a332db t2.nano robinson.com

# aws ec2 describe-instances | jq ... | column -t
i-0f112d652ecf13dac c3.x2large fisher.com
i-0b3b5128445a332db t2.nano robinson.com
```

EC2

List Instance ID, Type and Name

```
aws ec2 describe-instances | jq -r '.Reservations[].Instances[]|.InstanceId+" "+.InstanceType+" "+(.Tags[] | select(.K
i-0f112d652ecf13dac c3.x2large fisher.com
i-0b3b5128445a332db t2.nano robinson.com
```

```
i-0d1c1cf4e980ac593  t2.micro    nolan.com
i-004ee6b792c3b6914  t2.nano    grimes-green.net
i-00f11e8e33c971058  t2.nano    garrett.com
```

List Instances with Public IP Address and Name

👉 Tip: You can directly put this to your `/etc/hosts`

```
aws ec2 describe-instances --query 'Reservations[*].Instances[?not_null(PublicIpAddress)]' | jq -r '.[*].PublicIpAdd
223.64.72.64    fisher.com
198.82.207.161  robinson.com
182.139.20.233  nolan.com
153.134.83.44   grimes-green.net
202.32.63.121   garrett.com
```

List of VPCs and CIDR IP Block

```
aws ec2 describe-vpcs | jq -r '.Vpcs[]|.VpcId+" "+(.Tags[]|select(.Key=="Name").Value)+" "+.CidrBlock'
vpc-0d1c1cf4e980ac593  frontend-vpc  10.0.0.0/16
vpc-00f11e8e33c971058  backend-vpc   172.31.0.0/16
```

List of Subnets for a VPC

```
aws ec2 describe-subnets --filter Name=vpc-id,Values=vpc-0d1c1cf4e980ac593 | jq -r '.Subnets[]|.SubnetId+" "+.CidrBloc
subnet-0dae5d4daa47fe4a2  10.0.128.0/20  Public Subnet 1
subnet-0641a25faccb01f0f  10.0.32.0/19   Private Subnet 2
subnet-09fb8038641f1f36f  10.0.0.0/19    Private Subnet 1
subnet-02a63c67684d8deed  10.0.144.0/20  Public Subnet 2
```

List of Security Groups

```
aws ec2 describe-security-groups | jq -r '.SecurityGroups[]|.GroupId+" "+.GroupName'
sg-02a63c67684d8deed  backend-db
sg-0dae5d4daa47fe4a2  backend-redis
sg-0a56bff7b12264282  frontend-lb
sg-0641a25faccb01f0f  frontend-https
sg-09fb8038641f1f36f  internal-ssh
```

Print Security Groups for an Instance

```
aws ec2 describe-instances --instance-ids i-0dae5d4daa47fe4a2 | jq -r '.Reservations[]|.Instances[]|.SecurityGroups[]|.G
sg-02a63c67684d8deed  backend-db
sg-0dae5d4daa47fe4a2  backend-redis
```

Edit Security Groups of an Instance

👉 You have to provide existing Security Group IDs as well

```
aws ec2 modify-instance-attribute --instance-id i-0dae5d4daa47fe4a2 --groups sg-02a63c67684d8deed sg-0dae5d4daa47fe4a2
```

Print Security Group Rules as FromAddress and ToPort

```
aws ec2 describe-security-groups --group-ids sg-02a63c67684d8deed | jq -r '.SecurityGroups[]|.IpPermissions[]|. as $par
223.64.72.64/32    3306
198.82.207.161/32  3306
```

```
168.244.58.160/32 3306
202.0.149.202/32 3306
212.143.80.102/32 3306
```

Add Rule to Security Group

```
aws ec2 authorize-security-group-ingress --group-id sg-02a63c67684d8deed --protocol tcp --port 443 --cidr 35.0.0.1
```

Delete Rule from Security Group

```
aws ec2 revoke-security-group-ingress --group-id sg-02a63c67684d8deed --protocol tcp --port 443 --cidr 35.0.0.1
```

Edit Rules of Security Group

👉 You have to provide All IP Ranges as well

```
aws ec2 update-security-group-rule-descriptions-ingress --group-id sg-02a63c67684d8deed --ip-permissions 'ToPort=443,I
```

Delete Security Group

```
aws ec2 delete-security-group --group-id sg-02a63c67684d8deed
```

S3

List Buckets

```
aws s3 ls
2020-01-28 18:49:50 customer-data-primary
2020-01-28 18:50:22 customer-data-backup
2020-01-28 18:50:54 wordpress-cdn
2020-01-28 18:52:25 backend-artifacts-20200220-deployment
```

List Files in a Bucket

```
aws s3 ls wordpress-cdn/wp-content/uploads/2019/10/04/
2019-10-04 15:02:02 133557 amazing-content.jpg
2019-10-04 15:02:02 2986 amazing-content-103x50.jpg
2019-10-04 15:02:02 5640 amazing-content-120x120.jpg
2019-10-04 15:02:02 7924 amazing-content-150x150.jpg
```

Create Bucket

```
aws s3 mb s3://my-awesome-new-bucket
make_bucket: my-awesome-new-bucket
```

Delete Bucket

```
aws s3 rb s3://my-awesome-new-bucket --force
```

Download S3 Object to Local

```
aws s3 cp s3://my-awesome-new-bucket .
download: ./backup.tar from s3://my-awesome-new-bucket/backup.tar
```

Upload Local File as S3 Object

```
aws s3 cp backup.tar s3://my-awesome-new-bucket
upload: ./backup.tar to s3://my-awesome-new-bucket/backup.tar
```

Delete S3 Object

```
aws s3 rm s3://my-awesome-new-bucket/secret-file.gz .
delete: s3://my-awesome-new-bucket/secret-file.gz
```

Download Bucket to Local

```
aws s3 sync s3://my-awesome-new-bucket/ /media/Passport-Ultra/Backup
```

Upload Local Directory to Bucket

```
aws s3 sync /home/minhaz/Downloads s3://my-awesome-new-bucket/
```

Share S3 Object without Public Access

```
aws s3 presign s3://my-awesome-new-bucket/business-reports.pdf --expires-in 3600
https://my-awesome-new-bucket.s3.amazonaws.com/business-reports.pdf?AWSAccessKeyId=AKISUENSAKSIEUAA&Expires=1582876994
```

API Gateway

List of API Gateway IDs and Names

```
aws apigateway get-rest-apis | jq -r '.items[] | .id+" "+.name'
5e3221cf8 backend-api
69ef7d4c8 frontend-api
bb1e3c281 partner-api
f99796943 internal-crm-api
ee86b4cde import-data-api
```

List of API Gateway Keys

```
aws apigateway get-api-keys | jq -r '.items[] | .id+" "+.name'
ee86b4cde backend-api-key
69ef7d4c8 partner-api-key
```

List API Gateway Domain Names

```
aws apigateway get-domain-names | jq -r '.items[] | .domainName+" "+.regionalDomainName'
backend-api.mdminhazulhaque.io d-ee86b4cde.execute-api.ap-southeast-1.amazonaws.com
frontend-api.mdminhazulhaque.io d-bb1e3c281.execute-api.ap-southeast-1.amazonaws.com
```

List of Resources for API Gateway

```
aws apigateway get-resources --rest-api-id ee86b4cde | jq -r '.items[] | .id+" "+.path'
ee86b4cde /{proxy+}
69ef7d4c8 /
```

Find Lambda for API Gateway Resource

```
aws apigateway get-integration --rest-api-id ee86b4cde --resource-id 69ef7d4c8 --http-method GET | jq -r '.uri'
arn:aws:lambda:ap-southeast-1:987654321:function:backend-api-function-5d4daa47fe4a2:live/invocations
```

ELB

List of ELB Hostnames

```
aws elbv2 describe-load-balancers --query 'LoadBalancers[*].DNSName' | jq -r 'to_entries[] | .value'
frontend-lb-1220186848339.ap-southeast-1.elb.amazonaws.com
backend-lb-6208709163457.ap-southeast-1.elb.amazonaws.com
```

List of ELB ARNs

```
aws elbv2 describe-load-balancers | jq -r '.LoadBalancers[] | .LoadBalancerArn'
arn:aws:elasticloadbalancing:ap-southeast-1:987654321:loadbalancer/app/frontend-lb/1220186848339
arn:aws:elasticloadbalancing:ap-southeast-1:987654321:loadbalancer/app/backend-lb/6208709163457
```

List of ELB Target Group ARNs

```
aws elbv2 describe-target-groups | jq -r '.TargetGroups[] | .TargetGroupArn'
arn:aws:elasticloadbalancing:ap-southeast-1:987654321:targetgroup/frontend/b6da07d35
arn:aws:elasticloadbalancing:ap-southeast-1:987654321:targetgroup/backend/97ad3b13c
```

Find Instances for a Target Group

```
aws elbv2 describe-target-health --target-group-arn arn:aws:elasticloadbalancing:ap-southeast-1:987654321:targetgroup/i-0b3b5128445a332db
i-0d1c1cf4e980ac593
i-00f11e8e33c971058
```

RDS

List of DB Clusters

```
aws rds describe-db-clusters | jq -r '.DBClusters[] | .DBClusterIdentifier+" "+.Endpoint'
backend-prod backend-prod.cluster-b6da07d35.ap-southeast-1.rds.amazonaws.com
internal-prod internal-dev.cluster-b6da07d35.ap-southeast-1.rds.amazonaws.com
```

List of DB Instances

```
aws rds describe-db-instances | jq -r '.DBInstances[] | .DBInstanceIdentifier+" "+.DBInstanceClass+" "+.Endpoint.Address'
backend-dev db.t3.medium backend-prod.b6da07d35.ap-southeast-1.rds.amazonaws.com
internal-dev db.t2.micro internal-dev.b6da07d35.ap-southeast-1.rds.amazonaws.com
```

Take DB Instance Snapshot

```
aws rds create-db-snapshot --db-snapshot-identifier backend-dev-snapshot-0001 --db-instance-identifier backend-dev
aws rds describe-db-snapshots --db-snapshot-identifier backend-dev-snapshot-0001 --db-instance-identifier general
```

Take DB Cluster Snapshot

```
aws rds create-db-cluster-snapshot --db-cluster-snapshot-identifier backend-prod-snapshot-0002 --db-cluster-identifier
aws rds describe-db-cluster-snapshots --db-cluster-snapshot-identifier backend-prod-snapshot-0002 --db-cluster-identif
```

ElastiCache

List of ElastiCache Machine Type and Name

```
aws elasticache describe-cache-clusters | jq -r '.CacheClusters[] | .CacheNodeType+" "+.CacheClusterId'
cache.t2.micro      backend-login-hk
cache.t2.micro      backend-login-vn
cache.t2.micro      backend-login-sg
```

List of ElastiCache Replication Groups

```
aws elasticache describe-replication-groups | jq -r '.ReplicationGroups[] | .ReplicationGroupId+" "+.NodeGroups[].Prim
backend-login-hk  backend-login-hk.6da35.ng.0001.apse1.cache.amazonaws.com
backend-login-vn  backend-login-vn.6da35.ng.0001.apse1.cache.amazonaws.com
backend-login-sg  backend-login-sg.6da35.ng.0001.apse1.cache.amazonaws.com
```

List of ElastiCache Snapshots

```
aws elasticache describe-snapshots | jq -r '.Snapshots[] | .SnapshotName'
automatic.backend-login-hk-2020-02-27-00-27
automatic.backend-login-vn-2020-02-27-00-27
automatic.backend-login-sg-2020-02-27-00-27
```

Create ElastiCache Snapshot

```
aws elasticache create-snapshot --snapshot-name backend-login-hk-snap-0001 --replication-group-id backend-login-hk --c
```

Delete ElastiCache Snapshot

```
aws elasticache delete-snapshot --snapshot-name backend-login-hk-snap-0001
```

Scale Up/Down ElastiCache Replica

```
aws elasticache increase-replica-count --replication-group-id backend-login-hk --apply-immediately
aws elasticache decrease-replica-count --replication-group-id backend-login-hk --apply-immediately
```

Lambda

List of Lambda Functions, Runtime and Memory


```
aws lambda list-functions | jq -r '.Functions[] | .FunctionName+" "+.Runtime+" "+(.MemorySize|tostring)'
backend-api-function      nodejs8.10  512
backend-signup-email-function nodejs10.x  128
partner-api-8XJAP1VVLYA7 python3.7   128
marketing-promo-sqs-function nodejs10.x  128
```

List of Lambda Layers

```
aws lambda list-layers | jq -r '.Layers[] | .LayerName'
imagemagik-layer
django-layer
nodejs-extra-layer
```

List of Source Event for Lambda

```
aws lambda list-event-source-mappings | jq -r '.EventSourceMappings[] | .FunctionArn+" "+.EventSourceArn'
arn:aws:lambda:function:backend-api-function      arn:aws:dynamodb:table/prod-user-list/stream
arn:aws:lambda:function:backend-signup-email-function arn:aws:dynamodb:table/prod-user-email/stream
arn:aws:lambda:function:partner-api-8XJAP1VVLYA7  arn:aws:sqs:partner-input-msg-queue
arn:aws:lambda:function:marketing-promo-sqs-function arn:aws:sqs:promo-input-msg-queue
```

Download Lambda Code

```
aws lambda get-function --function-name DynamoToSqs | jq -r .Code.Location
https://aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/987654321/backend-api-function-1fda0de7-a751
```

Cloudwatch

List of CloudWatch Alarms and Status

```
aws cloudwatch describe-alarms | jq -r '.MetricAlarms[] | .AlarmName+" "+.Namespace+" "+.StateValue'
backend-autoscale  AWS/EC2      OK
backend-lb         AWS/ApplicationELB OK
partner-hk        AWS/ECS      ALARM
partner-vn        AWS/ECS      ALARM
partner-sg        AWS/ECS      ALARM
userdata-read     AWS/DynamoDB OK
userdata-write    AWS/DynamoDB OK
```

Create Alarm for EC2 High CPU Utilization

```
aws cloudwatch put-metric-alarm --alarm-name high-cpu-usage --alarm-description "Alarm when CPU exceeds 70 percent" --
```

Create Alarm for EC2 High StatusCheckFailed_Instance

```
aws cloudwatch put-metric-alarm --alarm-name EC2-StatusCheckFailed-AppServer --alarm-description "EC2 StatusCheckFailed"
```

Route53

List Domains

```
aws route53 list-hosted-zones | jq -r '.HostedZones[]|.Id+" "+.Name'
/hostedzone/ZEB1PAH4U mysite.com.
/hostedzone/ZQU0HGH3G yoursite.com.
/hostedzone/ZEAD0A0CO staywith.us.
```

List Records for a Domain (Zone)

```
aws route53 list-resource-record-sets --hosted-zone-id /hostedzone/ZEB1PAH4U | jq -r '.ResourceRecordSets[]| if (.Alias
A      mysite.com.          dualstack.mysite-lb-967522168.ap-southeast-1.elb.amazonaws.com.
A      mysite.com.          11.22.33.44
TXT    _amazonSES.mysite.com. 6c6d761371f0480bbe60de0df275b550
A      test.mysite.com.      55.66.77.88
CNAME  www.mysite.com.      mysite.com
```

SNS

List of SNS Topics

```
aws sns list-topics | jq -r '.Topics[]|.TopicArn'
arn:aws:sns:ap-southeast-1:987654321:backend-api-monitoring
arn:aws:sns:ap-southeast-1:987654321:dynamodb-count-check
arn:aws:sns:ap-southeast-1:987654321:partner-integration-check
arn:aws:sns:ap-southeast-1:987654321:autoscale-notifications
```

List of SNS Topic and related Subscriptions

```
aws sns list-subscriptions | jq -r '.Subscriptions[]|.TopicArn+" "+.Protocol+" "+.Endpoint'
arn:aws:sns:ap-southeast-1:autoscale-notifications    lambda    arn:aws:lambda:function:autoscale-function
arn:aws:sns:ap-southeast-1:backend-api-monitoring     email    alert@mdminhazulhaque.io
arn:aws:sns:ap-southeast-1:dynamodb-count-check       email    alert@mdminhazulhaque.io
arn:aws:sns:ap-southeast-1:partner-integration-check  lambda    arn:aws:lambda:function:partner-function
arn:aws:sns:ap-southeast-1:autoscale-notifications    lambda    arn:aws:lambda:function:autoscale-function
```

Publish to SNS Topic

```
aws sns publish --topic-arn arn:aws:sns:ap-southeast-1:987654321:backend-api-monitoring \
--message "Panic!!!" \
--subject "The API is down!!!"
```

DynamoDB

List of DynamoDB Tables

```
aws dynamodb list-tables | jq -r '.TableNames[]'
userdata_hk
userdata_vn
userdata_sg
providers
events
```

Get All Items from a Table

! This command will stream ALL items until SIGINT is sent

```
aws dynamodb scan --table-name events
```

Get Item Count from a Table

```
aws dynamodb scan --table-name events --select COUNT | jq .ScannedCount
726119
```

Get Item using Key

```
aws dynamodb get-item --table-name events --key '{"email": {"S": "admin@mdminhazulhaque.io"}}'
{
  "Item": {
    "email": {
      "S": "admin@mdminhazulhaque.io"
    },
    "created_at": {
      "N": "1554780667296"
    },
    "event_type": {
      "S": "DISPATCHED"
    }
  }
}
```

Get Specific Fields from an Item

```
aws dynamodb get-item --table-name events --key '{"email": {"S": "admin@mdminhazulhaque.io"}}' --attributes-to-get eve
{
  "Item": {
    "event_type": {
      "S": "DISPATCHED"
    }
  }
}
```

Delete Item using Key

```
aws dynamodb delete-item --table-name events --key '{"email": {"S": "admin@mdminhazulhaque.io"}}'
```

SQS

List Queues

```
aws sqs list-queues | jq -r '.QueueUrls[]'
https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo
https://ap-southeast-1.queue.amazonaws.com/987654321/user-signup
```

Create Queue

```
aws sqs create-queue --queue-name public-events.fifo | jq -r .QueueUrl
https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo
```

Count Messages in Queue

```
aws sqs get-queue-attributes --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo | jq
arn:aws:sqs:ap-southeast-1:987654321:events.fifo 42
```

Send Message

```
aws sqs send-message --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo --message-bod
{
  "MD5OfMessageBody": "37b51d194a7513e45b56f6524f2d51f2",
  "MessageId": "4226398e-bab0-4bee-bf5a-8e7ae18c855a"
}
```

Receive Message

```
aws sqs receive-message --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo | jq -r '.
Hello
```

Delete Message

```
aws sqs delete-message --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo --receipt-h
```

Purge Queue

```
aws sqs purge-queue --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo
```

Delete Queue

```
aws sqs delete-queue --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo
```

CloudFront

List of CloudFront Distributions and Origins

```
aws cloudfront list-distributions | jq -r '.DistributionList.Items[] | .DomainName+" "+.Origins.Items[0].DomainName'
d9d5bb1e3c281f.cloudfront.net  frontend-prod-hk.s3.amazonaws.com
d12b09e8a0a996.cloudfront.net  frontend-prod-vn.s3.amazonaws.com
db64e7e9b3cc22.cloudfront.net  frontend-prod-sg.s3.amazonaws.com
d5e3221cf8b921.cloudfront.net  cdn.mdminhazulhaque.io
```

Create Cache Invalidation

```
aws cloudfront create-invalidation --distribution-id D12B09E8A0A996 --path /blog/* /blog/assets/* | jq -r '.Invalid
IALJ5AL93ZD79
```

Check Cache Invalidation Status

```
aws cloudfront get-invalidation --distribution-id D12B09E8A0A996 --id IALJ5AL93ZD79 | jq -r '.Invalidation.Status'
Completed
```

Amplify

List of Amplify Apps and Source Repository

```
aws amplify list-apps | jq -r '.apps[] | .name+" "+.defaultDomain+" "+.repository'
fe-vn d9d5bb1e3c281f.amplifyapp.com https://bitbucket.org/aws/frontend-vn
fe-hk db64e7e9b3cc22.amplifyapp.com https://bitbucket.org/aws/frontend-hk
fe-sg d5e3221cf8b921.amplifyapp.com https://bitbucket.org/aws/frontend-sg
```

Cognito

List of User Pool IDs and Names

```
aws cognito-idp list-user-pools --max-results 60 | jq -r '.UserPools[] | .Id+" "+.Name'
ap-southeast-1_b6da07d35 prod-users
ap-southeast-1_b6da07d34 dev-users
```

List of Phone and Email of All Users

```
aws cognito-idp list-users --user-pool-id ap-southeast-1_b6da07d35 | jq -r '.Users[].Attributes | from_entries | .sub
585fb96e-525c-4f9b-9d41-865d2dffde9b +601122334455 admin@mdminhazulhaque.io
71f2778c-8e21-4775-94dc-e363c77d1ae1 +601122334455 foo@bar.com
8fc1882e-e661-49db-88e6-45d370bc352a +601122334455 cli@aws.com
```

IAM User

List of UserId and UserName

```
aws iam list-users | jq -r '.Users[]|.UserId+" "+.UserName'
AIDAZBWIOJIIQFOLNBXXCVSUQ kaiser
AIDAZCTWYVXYOKSHVWXPYPLR thornton
AIDAZUYALCGFQJENBCZFJTVX maldonado
AIDAZKQAFIGQJWOKKSKRBLGE key
AIDAZXUDGQVQCEWBFGIJOWWY nelson
```

Get Single User

```
aws iam get-user --user-name kaiser
```

Add User

```
aws iam create-user --user-name audit-temp
```

Delete User

```
aws iam delete-user --user-name audit-temp
```

List Access Keys for User

```
aws iam list-access-keys --user-name audit-temp | jq -r .AccessKeyMetadata[].AccessKeyId
AKIABWIOJIIQFOLNBXXCVSUQ
AKIACTWYVXYOKSHVWXPYPLR
AKIAUYALCGFQJENBCZFJTVX
```

Delete Access Key for User

```
aws iam delete-access-key --user-name audit-temp --access-key-id AKIABWIOJIQFOLNBXXCVSUQ
```

Activate/Deactivate Access Key for User

```
aws iam update-access-key --status Inactive --user-name audit-temp --access-key-id AKIABWIOJIQFOLNBXXCVSUQ
aws iam update-access-key --status Active --user-name audit-temp --access-key-id AKIABWIOJIQFOLNBXXCVSUQ
```

Generate New Access Key for User

```
aws iam create-access-key --user-name audit-temp | jq -r '.AccessKey | .AccessKeyId+" "+.SecretAccessKey'
AKIABWIOJIQFOLNBXXCVSUQ p9ge02ebLX9jobdQKmfikRqCiEw3HBylwHyXq0z
```

IAM Group

List Groups

```
aws iam list-groups | jq -r .Groups[].GroupName
developers
administrators
testers
marketing-ro
```

Add/Delete Groups

```
aws iam create-group --group-name business-ro
aws iam delete-group --group-name business-ro
```

List of Policies and ARNs

```
aws iam list-policies | jq -r '.Policies[]|.PolicyName+" "+.Arn'
aws iam list-policies --scope AWS | jq -r '.Policies[]|.PolicyName+" "+.Arn'
aws iam list-policies --scope Local | jq -r '.Policies[]|.PolicyName+" "+.Arn'
```

List of User/Group/Roles for a Policy

```
aws iam list-entities-for-policy --policy-arn arn:aws:iam::987654321:policy/Marketing-ReadOnly
```

List Policies for a Group

```
aws iam list-attached-group-policies --group-name business-ro
```

Add Policy to a Group

```
aws iam attach-group-policy --group-name business-ro --policy-arn arn:aws:iam::aws:policy/DynamoDBReadOnlyAccess
```

Add User to a Group

```
aws iam add-user-to-group --group-name business-ro --user-name marketing-michael
```

Remove User from a Group

```
aws iam remove-user-from-group --group-name business-ro --user-name marketing-alice
```

List Users in a Group

```
aws iam get-group --group-name business-ro
```

List Groups for a User

```
aws iam list-groups-for-user --user-name qa-bob
```

Attach/Detach Policy to a Group

```
aws iam detach-group-policy --group-name business-ro --policy-arn arn:aws:iam::aws:policy/DynamoDBFullAccess  
aws iam attach-group-policy --group-name business-ro --policy-arn arn:aws:iam::aws:policy/DynamoDBFullAccess
```