**🔧 Install AWS Cli**

* aws <https://aws.amazon.com/cli/>

**📢 Table of Contents**

* [EC2](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#ec2)
  + [List Instance ID, Type and Name](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-instance-id-type-and-name)
  + [List Instances with Public IP Address and Name](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-instances-with-public-ip-address-and-name)
  + [List of VPCs and CIDR IP Block](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-vpcs-and-cidr-ip-block)
  + [List of Subnets for a VPC](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-subnets-for-a-vpc)
  + [List of Security Groups](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-security-groups)
  + [Print Security Groups for an Instance](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#print-security-groups-for-an-instance)
  + [Edit Security Groups of an Instance](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#edit-security-groups-of-an-instance)
  + [Print Security Group Rules as FromAddress and ToPort](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#print-security-group-rules-as-fromaddress-and-toport)
  + [Add Rule to Security Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#add-rule-to-security-group)
  + [Delete Rule from Security Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-rule-from-security-group)
  + [Edit Rules of Security Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#edit-rules-of-security-group)
  + [Delete Security Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-security-group)
* [S3](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#s3)
  + [List Buckets](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-buckets)
  + [List Files in a Bucket](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-files-in-a-bucket)
  + [Create Bucket](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#create-bucket)
  + [Delete Bucket](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-bucket)
  + [Download S3 Object to Local](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#download-s3-object-to-local)
  + [Upload Local File as S3 Object](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#upload-local-file-as-s3-object)
  + [Delete S3 Object](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-s3-object)
  + [Download Bucket to Local](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#download-bucket-to-local)
  + [Upload Local Directory to Bucket](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#upload-local-directory-to-bucket)
  + [Share S3 Object without Public Access](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#share-s3-object-without-public-access)
* [API Gateway](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#api-gateway)
  + [List of API Gateway IDs and Names](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-api-gateway-ids-and-names)
  + [List of API Gateway Keys](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-api-gateway-keys)
  + [List API Gateway Domain Names](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-api-gateway-domain-names)
  + [List of Resources for API Gateway](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-resources-for-api-gateway)
  + [Find Lambda for API Gateway Resource](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#find-lambda-for-api-gateway-resource)
* [ELB](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#elb)
  + [List of ELB Hostnames](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-elb-hostnames)
  + [List of ELB ARNs](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-elb-arns)
  + [List of ELB Target Group ARNs](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-elb-target-group-arns)
  + [Find Instances for a Target Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#find-instances-for-a-target-group)
* [RDS](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#rds)
  + [List of DB Clusters](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-db-clusters)
  + [List of DB Instances](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-db-instances)
  + [Take DB Instance Snapshot](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#take-db-instance-snapshot)
  + [Take DB Cluster Snapshot](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#take-db-cluster-snapshot)
* [ElastiCache](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#elasticache)
  + [List of ElastiCache Machine Type and Name](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-elasticache-machine-type-and-name)
  + [List of ElastiCache Replication Groups](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-elasticache-replication-groups)
  + [List of ElastiCache Snapshots](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-elasticache-snapshots)
  + [Create ElastiCache Snapshot](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#create-elasticache-snapshot)
  + [Delete ElastiCache Snapshot](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-elasticache-snapshot)
  + [Scale Up/Down ElastiCache Replica](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#scale-updown-elasticache-replica)
* [Lambda](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#lambda)
  + [List of Lambda Functions, Runtime and Memory](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-lambda-functions-runtime-and-memory)
  + [List of Lambda Layers](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-lambda-layers)
  + [List of Source Event for Lambda](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-source-event-for-lambda)
  + [Download Lambda Code](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#download-lambda-code)
* [Cloudwatch](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#cloudwatch)
  + [List of CloudWatch Alarms and Status](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-cloudwatch-alarms-and-status)
  + [Create Alarm for EC2 High CPUUtilization](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#create-alarm-for-ec2-high-cpuutilization)
  + [Create Alarm for EC2 High StatusCheckFailed\_Instance](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#create-alarm-for-ec2-high-statuscheckfailed_instance)
* [Route53](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#route53)
  + [List Domains](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-domains)
  + [List Records for a Domain (Zone)](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-records-for-a-domain-zone)
* [SNS](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#sns)
  + [List of SNS Topics](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-sns-topics)
  + [List of SNS Topic and related Subscriptions](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-sns-topic-and-related-subscriptions)
  + [Publish to SNS Topic](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#publish-to-sns-topic)
* [DynamoDB](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#dynamodb)
  + [List of DynamoDB Tables](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-dynamodb-tables)
  + [Get All Items from a Table](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#get-all-items-from-a-table)
  + [Get Item Count from a Table](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#get-item-count-from-a-table)
  + [Get Item using Key](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#get-item-using-key)
  + [Get Specific Fields from an Item](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#get-specific-fields-from-an-item)
  + [Delete Item using Key](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-item-using-key)
* [SQS](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#sqs)
  + [List Queues](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-queues)
  + [Create Queue](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#create-queue)
  + [Count Messages in Queue](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#count-messages-in-queue)
  + [Send Message](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#send-message)
  + [Receive Message](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#receive-message)
  + [Delete Message](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-message)
  + [Purge Queue](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#purge-queue)
  + [Delete Queue](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-queue)
* [CloudFront](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#cloudfront)
  + [List of CloudFront Distributions and Origins](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-cloudfront-distributions-and-origins)
  + [Create Cache Invalidation](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#create-cache-invalidation)
  + [Check Cache Invalidation Status](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#check-cache-invalidation-status)
* [Amplify](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#amplify)
  + [List of Amplify Apps and Source Repository](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-amplify-apps-and-source-repository)
* [Cognito](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#cognito)
  + [List of User Pool IDs and Names](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-user-pool-ids-and-names)
  + [List of Phone and Email of All Users](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-phone-and-email-of-all-users)
* [IAM User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#iam-user)
  + [List of UserId and UserName](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-userid-and-username)
  + [Get Single User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#get-single-user)
  + [Add User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#add-user)
  + [Delete User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-user)
  + [List Access Keys for User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-access-keys-for-user)
  + [Delete Access Key for User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#delete-access-key-for-user)
  + [Activate/Deactivate Access Key for User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#activatedeactivate-access-key-for-user)
  + [Generate New Access Key for User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#generate-new-access-key-for-user)
* [IAM Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#iam-group)
  + [List Groups](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-groups)
  + [Add/Delete Groups](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#adddelete-groups)
  + [List of Policies and ARNs](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-policies-and-arns)
  + [List of User/Group/Roles for a Policy](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-of-usergrouproles-for-a-policy)
  + [List Policies for a Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-policies-for-a-group)
  + [Add Policy to a Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#add-policy-to-a-group)
  + [Add User to a Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#add-user-to-a-group)
  + [Remove User from a Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#remove-user-from-a-group)
  + [List Users in a Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-users-in-a-group)
  + [List Groups for a User](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#list-groups-for-a-user)
  + [Attach/Detach Policy to a Group](https://gist.github.com/belkar30d/473e13155207480ffcc18a8fa35350c2#attachdetach-policy-to-a-group)

**ℹ️ 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(.Key == "Name").Value)'

i-0f112d652ecf13dac c3.xlarge 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 '.[][]|.PublicIpAddress+" "+(.Tags[]|select(.Key=="Name").Value)'

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+" "+.CidrBlock+" "+(.Tags[]|select(.Key=="Name").Value)'

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[]|.GroupId+" "+.GroupName'

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 $parent|(.IpRanges[].CidrIp+" "+($parent.ToPort|tostring))'

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,IpProtocol=tcp,IpRanges=[{CidrIp=202.171.186.133/32,Description=Home}]'

**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&Signature=kizOEA93kaIHw7uv25wSFIKLmAx

**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/wordpress-ph/88f517d6b5326a26 | jq -r '.TargetHealthDescriptions[] | .Target.Id'

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 backend-prod

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

**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[].PrimaryEndpoint.Address'

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 --cache-cluster-id backend-login-hk

**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://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/987654321/backend-api-function-1fda0de7-a751-4586-bf64-5601a410c170

**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 CPUUtilization**

aws cloudwatch put-metric-alarm --alarm-name high-cpu-usage --alarm-description "Alarm when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --dimensions "Name=InstanceId,Value=i-123456789" --evaluation-periods 2 --alarm-actions arn:aws:sns:ap-southeast-1:987654321:System-Alerts --unit Percent

**Create Alarm for EC2 High StatusCheckFailed\_Instance**

aws cloudwatch put-metric-alarm --alarm-name EC2-StatusCheckFailed-AppServer --alarm-description "EC2 StatusCheckFailed for AppServer" --metric-name StatusCheckFailed\_Instance --namespace AWS/EC2 --statistic Average --period 60 --threshold 0 --comparison-operator GreaterThanThreshold --dimensions "Name=InstanceId,Value=i-123456789" --evaluation-periods 3 --alarm-actions arn:aws:sns:ap-southeast-1:987654321:System-Alerts --unit Count

**Route53**

**List Domains**

aws route53 list-hosted-zones | jq -r '.HostedZones[]|.Id+" "+.Name'

/hostedzone/ZEB1PAH4U mysite.com.

/hostedzone/ZQUOHGH3G yoursite.com.

/hostedzone/ZEADEA0CO staywith.us.

**List Records for a Domain (Zone)**

aws route53 list-resource-record-sets --hosted-zone-id /hostedzone/ZEB1PAH4U | jq -r '.ResourceRecordSets[]| if (.AliasTarget!=null) then .Type+" "+.Name+" "+.AliasTarget.DNSName else .Type+" "+.Name+" "+.ResourceRecords[].Value end'

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 untill 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 event\_type

{

"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 -r '.Attributes | .QueueArn + " " + .ApproximateNumberOfMessages'

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-body Hello

{

"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 '.Messages[] | .Body'

Hello

**Delete Message**

aws sqs delete-message --queue-url https://ap-southeast-1.queue.amazonaws.com/987654321/public-events.fifo --receipt-handle "AQEBpqKLxNb8rIOn9ykSeCkKebNzn0BrEJ3Cg1RS6MwID2t1oYHCnMP06GnuVZGzt7kpWXZ5ieLQ=="

**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 '.Invalidation.Id'

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 + " " + .phone\_number + " " + .email'

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'

AIDAZBWIOJIQFOLNBXXCVSUQ 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

AKIABWIOJIQFOLNBXXCVSUQ

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