* Course Overview
* Introduction to the AWS CLI
  + Use \ to tell CLI the command is not finished yet
* Creating and Using an IAM Ad..
  + Create users with just enough privileges
  + In AWS
    - Search ‘iam’
    - Click ‘Uers’
    - Click ‘Add user’
    - Click ‘programmatic access’ box
    - Click ‘Next’
    - Click ‘Attach existing policies directly’
    - Click ‘AdministratorAcess’
    - Click ‘Next;
    - Click ‘review’
    - Click ‘Create’
    - Store access key somewhere safe
* Installing the AWS CLI
  + Search the web for ‘install aws cli’
  + Download aws cli
* Configuring the AWS CLI
  + Can never retrieve lost key
  + Can generate a new one
  + Click on ‘user’ on upper right corner
  + Click on ‘My Security Credentials’
  + In a CLI type ‘aws configure’
    - For region name put your region
    - For output format put text(options: json, table, text)
  + Type ‘aws s3 ls’
    - aws: use the aws cli
    - s3: s3 service
    - ls: list all buckets in account
* Configuring Name Profiles fo…
  + Profiles: which account of aws to access
  + Aws cli stores credentials and configurations settings in two files in hidden directory in home directory
    - The hidden directory is .aws
    - config: contains the format and region preferences
    - Credentials: holds access key id and the key itself
  + aws configure --profile [profile name]
    - Add one or more additional account settings
  + Then to run commands for the account add --profile [profile name] after aws
  + ex)
    - aws --profile account2 s3 ls
  + Aws cli
    - Automate aws operations
    - Using IAM users
      * Never give root user cli access only IAM users
    - Installing the aws cli
* Getting Help and Guidance for Your AWS CLI Operations
  + Aws cli command syntax
    - aws [options] <command> <subcommand> [parameters]
  + Command: usually invoke specific aws services
  + Subcommands: usually verbs
    - ex) create, delete, enable ….
  + Options
    - ex) --region, --output, --profile, --dry-run
  + Use --dry-run to test syntax
* Finding Syntax Help for AWS…
  + Type ‘aws help’
    - Gives brief overview and options
  + ‘aws s3 help’
    - Gives information and subcommands etc
  + Aws docs is also a helpful source
  + Using Both the S3 and S3API
    - ‘aws s3 ls’
    - ‘aws s3api list-buckets’
    - Gives same results be formatted differently
  + S3
    - Sync file systems
    - Stream data
    - Transfer large file in multiple parts
  + S3API
    - Interact with the aws api
    - Granular control
* Configuring IAM Users, Group…
  + Look down root user and create IAM users to perform day to day administration
  + Create an IAM group
    - Attach a policy giving members just enough access
    - Sign IAM users
  + Type ‘aws iam create-user --user-name [user name]’
    - Creates an iam user
  + Type ‘aws iam get-user --user-name mike’ to get user info
  + Type ‘aws iam list-access-keys --user-name [username]’ to show access keys
  + Type ‘aws iam create-access-key --user-name [username]’ to create keys
  + Type ‘aws iam create-group --group-name [group name]’ to create group
  + To define group permissions
    - Can create json format policy document and attach to group
  + Type ‘aws iam list-policies | grep AmazonEC2 | grep Access’
    - Return descriptions of IAM policies in region
  + Type ‘aws iam attach-group-policy \ --policy-arn [policy arn] \ --group-name [group name]
  + Type ‘aws iam-add-user-to-group \ --group-name [group name] \ --user-name [user name]’
    - To add user to group
* Creating an S3-based Static …
  + Type ‘aws s3 mb s3://[bucket name]’
    - mb creates bucket
  + type ‘aws s3api put-bucket-acl --bucket [bucket name] --acl public-read’
    - make bucket publicly readable
  + type ‘aws s3 sync . s3://[bucket name] --acl public-read’
    - to give files the same permissions as the bucket
  + type ‘aws s3 website s3://[bucket name] --index-document index.html --error-document error.html’
    - define purpose of index.html and error.html
  + type ‘aws s3api get-bucket-website --bucket [bucket name]’
    - to confirm that the previous operation works
  + the url is then ‘http://[bucket name].s3-website-[region].amazonaws.com/’
* Provisioning and Launching a…
  + Type ‘aws ec2 create-security-group --group-name [group name] --description “description…”
    - To create a security group
  + Add rules to security group to control traffic one at a time
    - aws ec2 authorize-security-group-ingress --group-name [group name] --protocol tcp --port [port number] --cidr [ip address]
    - run the same command with different port numbers to open access to those ports
  + type ‘aws ec2 describe-security-groups --group-names [group name]’
    - to see configurations
  + type ‘aws --output table ec2 describeimages --filters “Name=description,Values=\*Amazon Linux 2\*” “Name=owner-alias,Values=amazon”
    - to filter out images with descript and owner as listed
  + copy image id
  + type ‘aws ec2 describe-subnets’
    - to get list of subnets available in the specified region
  + run instance
    - aws ec2 run-instances --image-id [image id] --count [number of instances] --instance-type t2.micro --key-name [key name] --security-group-ids [group ids] --subnet-id [subnet id] --user-date <file://my_script.sh> --tag-specifications ‘ResourceType=Instance,Tags=[{Key-webserver,Value=production}]’
  + can also add database using RDS
    - aws rds created-db-instance --db-instance-identifier sg-cli-test --allocated-storage 20 --db-instance-class db.m1.small --engine mysql --master-username myawsuser --master-user-password mypassword’
* Course Review