Maddie’s Sprint 1 Research for [TASK 432](https://dev.azure.com/F22-Team02-Wilson-Bondura-Clay-Hayes-Saad/F22-Team02-Wilson.Bondura.Clay.Hayes.Saad/_workitems/edit/432) EC2/S3 Tutorial &Notes

Story: As a developer, I must research our backend technology choices.

* EC2 Tutorial:  [AWS EC2 Tutorial Beginners to Advance - Full course on Amazon EC2 in 40 mins with Hands On Labs 2022](https://www.youtube.com/watch?v=8SQnGqp3YZM)
  + \*\*set up budget in AWS so we don't have overages
  + EC2 capabilities
    - Virtual machines
    - Storing data (EBS)
    - Distributing load (ELB)
    - Auto scaling
  + We may need to play with configuration: CPU, RAM, instance type (instanceclassGeneration.size)
    - Network attached (EBS)
    - Hardware (EC2)
    - \*\*we will be t2.micro b/c free
  + EC2 user data automates boot tasks
    - Bootstrapping = launching pre-configured commands when machine starts
    - Launched at first start of instance
  + Launching
    - AWS console -> compute
      * Do our east region
      * AMIs: look for marketplace of what want or just linux
      * \*\*\*what AMI will out project require
      * Key pair = pem for mac
      * Security group = http or shh and anywhere
      * User data -> advanced settings
        + Linux script for angular web server?
      * Launch it
    - Can edit after launch: iam rules…
  + Ssh command then put in the IP address to get into the instance and make changes --- use instance connect for ssh
  + Select ec2 instance connect
    - Can do this to get inside the instance and make any changes
    - Sudo yum install git -> inside instance
  + \*\*NOTE: stopping and restarting instance will change public IP which we don t want
    - STATIC IP address = elastic IP option
      * Allocate elastic
      * Amazons pool ipv4
    - Once allocate -> actions associate IP address
      * Select our instance to associate with static now can stop and start
  + EC2 instance type more ingo
    - Balance: compute memory and networking
  + Security groups = act as firewall on instance
    - Control in/outbound network, ports, ip ranges
    - Lock to a region or VPC
    - Maintain one group for SSH access
    - NOTE:
      * App time out = security group issue
      * "connection refused" = app error
    - We will need HTTP from anywhere
      * Port 80/443 for http and https
  + Ec2 instance connect in depth
    - No need for key file b/c within amazon
    - \*\*only uses amazon linux 2
  + Attach IAM role
    - Go to instance -> actions -> security IAM role
      * Have to have IAM role already created
  + Using AWS CLI
    - Aws configure -> access key
  + EC2 launch types
    - On demand: pay for what use, short-term workload
      * Seemed like what we will use
      * Spot instances - for temporary workloads if need but not for critical jobs/dbs
* [Create and Deploy Angular Application to AWS S3 - Step by Step Guide](https://www.youtube.com/watch?v=O06eyJ3vMyc)
  + Main steps:
    - Create the angular app
    - Build the app for production
    - Using the AWS mgt console
      * Create s3 bucket
      * Upload the angular files to the bucket
      * Deploy on S3
  + AWS
    - GO to IAM
      * Let computer be user (add user): admin.angular
        + Admin access, other presets
    - Logging in to user
      * Aws --version
        + install awscli
      * Node and NPM install
        + Check: nmp -v, node -v (if blank then install)
        + Also need to have brew on mac and nvm installed
      * "aws configure" in terminal
        + Access key = find all information when download csv after creating IAM user
        + \*ours are none???
      * Result: terminal logged in with created user
    - Now can create or use angluar app
      * Ng new app name
        + Already created-> cd to the app
      * Ng serve --open
        + Connects to server and adds to local host
      * Pull up app running in local host
      * Then do
    - Setting up in aws
      * Go to S3: where store information
      * New bucket with unique name
      * Permissions
        + Remove them because want to be able to access it to the public?
    - Uploading to bucket
      * Create resources
        + Ng build = build production ready

Dist folder for deployable version to upload

* + - * Click upload
        + Place dist folder in bucket
        + Grant public access
        + Standard
      * Static website hosting
        + Properties tab and use static website hosting
        + Index.html for index and error document

Endpoint link is the link to the website

* + - * + Permissions: bucket policy

Copy and paste this to give public access on s3

{

"Version":"2012-10-17", "Statement":[

{ "Sid":"PublicRead",

"Effect":"Allow",

"Principal": "\*",

"Action":[

"s3:GetObject"

], "Resource":[

"arn:aws:s3:::BUCKETNAME/\*"

]

}

]

}

* CODE
  + Go to package.json file
    - And add this to "scripts"
      * "aws deploy": "aws s3 sync dist/APPNAME/ s3://BUCKETNAME"
  + Once all done
    - Ng build && npm run aws-deploy
      * Re-deploy with the new json and command
  + Completed then
    - Go to your s3 bucket
    - Properties
    - And should see that static website is hosting by clicking the link
* Helpful guide in addition to video: <https://angular.io/guide/setup-local>
  + <https://angular.io/cli>
* \*\*\*NOTE:
  + Frontend doesn't require processing on server side so s3 is better b/c on ec2 it is a waste of resources

Make sure install nvm, node, npm, angular/cli

* Can check with node –v, npm –v
* Text

  Description automatically generated with medium confidenceText

  Description automatically generated