Assignment 2 *Software Architecture / Cloud Computing*
Ximena Lizeth González Plascencia
February 4th, 2023

Create a static website using AWS S3 and explain in great technical detail what you did. 30 points

- Website files creation:
    - Go to the open source platform CodePen and select a template for the portfolio website to use as a blueprint for my own website.
    - Create `index.html`, `main.css` and `main.js` files and modify to display my name, the link to my GitHub repository and links to future projects.

- aws s3api API vs aws s3 API
    - In comparison to the aws s3 API, aws s3api API provides direct access to all Amazon S3 API operations and allows for the user to implement some advanced and more granular operations that would otherwise be impossible in the s3 API (s3api API is considered more flexible).
        - s3 API commands are easier to use (they're more high-level, but more limited), if the desired operation can be done with s3 API, it is the recommended first option.
    - The s3api API is a separate way of using the s3 API and provides a direct mapping to other AWS services (EC2, Lambda, etc.).
        - It provides every operation a user would want to do with a S3 bucket (creating, deleting, copying files, etc.).
        - It functions entirely based on JSON models (all responses are in this format, and most commands accept JSON models as arguments).
            - Most of the other services have a REST API, which communicates in JSON format (regardless of what service is being "talked" to).

        - s3api is basically a way of talking in the same language of these other service's API.
    - Commands in s3 API are built on top of the commands of the s3api API.
        - S3 is intended to provide a way to do the low level things, such as copying files from local machines to an S3 bucket, moving files, removing files, synchronization of local directories and S3 buckets, etc.
            - The s3 API doesn't function only on JSON models, and provides a way to easily manipulate S3 files (s3 API is similar to a very big filesystem).

- Bucket creation:
    - Use aws s3api CLI to create the bucket with the create-bucket command.

```
aws s3api create-bucket --bucket ximena.cetystijuana.com --region us-east-1
```

```
{
    "Location": "/ximena.cetystijuana.com"
}
```

        - The command creates a new S3 bucket.
        - The argument –-bucket is the name of the bucket to create.
            - Not every string can be used, there are a set of restrictions regarding length, special characters, etc. But the most important is that a bucket name cannot be used by another AWS account (it must be universally unique).
        - The argument --region is the region in which the bucket is created.
            - The default region is US East (N. Virginia) (us-east-1), in this case this region was explicitly declared.
            - The way to choose the bucket's region to optimize latency, reduce costs, or address regulatory requirements, is to pick any AWS Region that is geographically close to the user.

- Use the aws s3 CLI to verify the bucket was created using the ls command.

```
(base) Ximenas-MacBook-Air:~ ximenagonzalez$ aws s3 ls
2023-02-03 18:15:46 ulises.cetystijuana.com
2023-01-22 12:54:53 www.cetystijuana.com
2023-02-03 18:16:09 ximena.cetystijuana.com
```

  - The command lists all the S3 buckets owned by the user, similar to running the command on a local machine and listing files inside a directory.
    - As output it provides the timestamp and date the bucket was created (it can change if any modification is done to the bucket) and the name of the bucket.

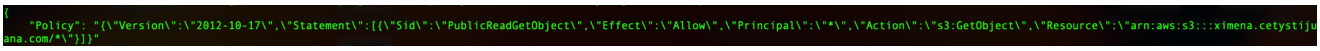
- Adding bucket policy:
  - Since the intention of the bucket is to host a static website, it's necessary to make it accessible to the public, in order to do this, a bucket policy is needed to provide read access.
    - A bucket policy is a policy used to edit and grant access permissions to a S3 bucket and its contents (the policy applies to all bucket content).
    - Only the bucket owner can edit a bucket's policy.
  - The bucket policy is created in a JSON format.

```json
{
"Version": "2012-10-17",
"Statement": [
  {
  "Sid": "PublicReadGetObject",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::ximena.cetystijuana.com/*"
  }
  ]
}
```

- The current policy version is 2012-10-17 (the only other one is the previous 2008-10-17). This defines the version of the policy language syntax rules (allowed attributes and values) that AWS uses to process a policy. The Version element should always be included and set to the most recent version in order to use policy variables introduced in the newest installation.
- The Statement element is the main element for a policy. It contains other statements indicating policy specifications.
    - The Sid is the statement ID and is an optional identifier for the policy statement, it's a description for what the policy is intended for.
    - The Effect element specifies whether the statement results in allowing or denying access to the bucket. It accepts only Allow and Deny as values (case sensitive).
    - The Principal element specifies the user, account, service or entity that is allowed or denied access to the bucket. The * (wildcard) grants permission to everyone (anonymous access).
    - The Action element specifies the action(s) that will be allowed or denied on the bucket. There's a list of these, but in this case the GetObject action is intended for read permission to the bucket.
    - The Resource element specifies the object(s) that the statement (policy) applies to. It's specified using and ARN (Amazon Resource Name) format:
        - *arn:aws:s3:::EXAMPLE-BUCKET/**
            - This example refers to all items within a specific S3 bucket.
            - The * indicates access to all of the bucket's contents.

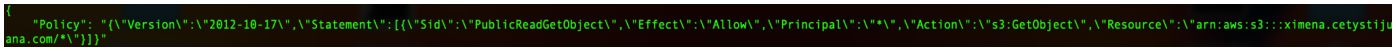
- Use the as3api put-bucket-policy command to apply the bucket policy to the S3 bucket.

```
aws s3api put-bucket-policy --bucket ximena.cetystijuana.com --policy file:///Users/ximenagonzalez/Desktop/policy.json
```



  - The –-bucket argument is the name of the bucket to which the policy is intended for.
  - The –-policy argument is the JSON file of the bucket policy, in this case a path to the file is provided.
    - file::// specifies that the provided value is a file in said path.
- Use the aws s3api get-bucket-policy to confirm the policy was created

```
aws s3api get-bucket-policy --bucket ximena.cetystijuana.com
```



- Uploading website files to bucket:
  - In order to make the bucket and actual website, it must have in it at least the index.html and error.html file (configuration files).
  - Use the command aws s3 website command to set the website configuration for a bucket.

```
aws s3 website s3://ximena.cetystijuana.com --index-document index.html --error-document error.html
```

  - The S3Uri (URI in for S3 resources) must be provided to indicate the URI (unique resource identifier) of the bucket.
  - The --index-document argument is the name of the main html of the website (name must be index.html).
  - The ---error-document argument is the name of the main html to display when a 4XX class error occurs (an error related to the user's request; indicates a problem occurred on the client's side; for example 404 Not Found).
  - By default this creates empty html files, no content will be displayed yet.

- In order to have a visual website, a sync with the local website directory and the remote bucket is necessary.
- The aws s3 sync command recursively copies new and updated files from the source directory to the destination.

```
aws s3 sync . s3://ximena.cetystijuana.com
```

```
upload: ./main.css to s3://ximena.cetystijuana.com/main.css
upload: ./main.js to s3://ximena.cetystijuana.com/main.js
upload: ./error.html to s3://ximena.cetystijuana.com/error.html
upload: ./index.html to s3://ximena.cetystijuana.com/index.html
```

    - The source directory . indicates that the source directory is the working directory (in this case the directory with the website's files)
    - The destination is the S3Uri to the bucket.
- Now in the browser the website files are visible through the link http://ximena.cetystijuana.com.s3-website-us-east-1.amazonaws.com/

Link your website to a subdomain of cetystijuana.com and explain in great technical detail what you did. 30 points.

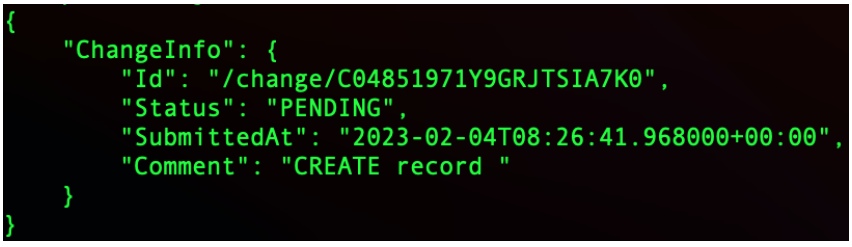
- Amazon Route 53:
    - It's a Domain Name System web service, it connects user requests to internet applications hosted on AWS or on-premises.
    - In this case it is useful to link the S3 bucket to a subdomain of cetystijuana.com (which is the bucket name).
- Using a custom URL by adding alternate domain names:
    - An alternate domain name or CNAME, allows the user to use their own domain name instead of the bucket's default URL.
    - In order to do this, a resource record set has to be created for the bucket, through a change batch request.
        - A change batch is considered a transactional element, is a JSON model that is sent to AWS Route 53 in order to validate the request and either make all or none of the changes in the batch request.
        - A resource record set is also a JSON model that contains authoritative DNS information for a specified domain name or subdomain name. It allows routing traffic from one domain to another.

```
{
  "Comment": "CREATE record ",
  "Changes": [{
  "Action": "CREATE",
              "ResourceRecordSet": {
                            "Name": "ximena.cetystijuana.com",
                            "Type": "CNAME",
                            "TTL": 300,
                          "ResourceRecords": [{ "Value":  "ximena.cetystijuana.com.s3-website-us-east-1.amazonaws.com"}]
}}]
}
```

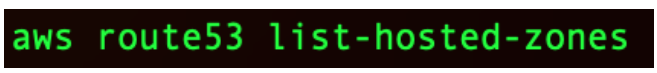
- The Changes element specifies the changes being requested in the change batch (in this case the creating a resource record set).
- The Comment element is optional documentation of what the JSON model is for.

- The Action element indicates that a record set is to be created, other values include DELETE and UPSERT.
- The ResourceRecordSet element specifies all the other elements that make up the resource record set.
    - The Name element indicates the domain or subdomain that the user wants to route traffic to (in this case the subdomain of cetystijuana.com).
    - The Type element specifies the DNS record type, in this case a CNAME record maps DNS queries for the name of the current domain, to another domain (the original bucket URL to the subdomain).
    - The TTL (time to live) element specifies the seconds that the DNS resolver will cache information about the record before submitting another request to AWS Route 53 to get the current values for that record (seconds taken to “refresh” the values of the record).
    - The ResourceRecords element specifies the S3 bucket to which the new subdomain will redirect traffic to.

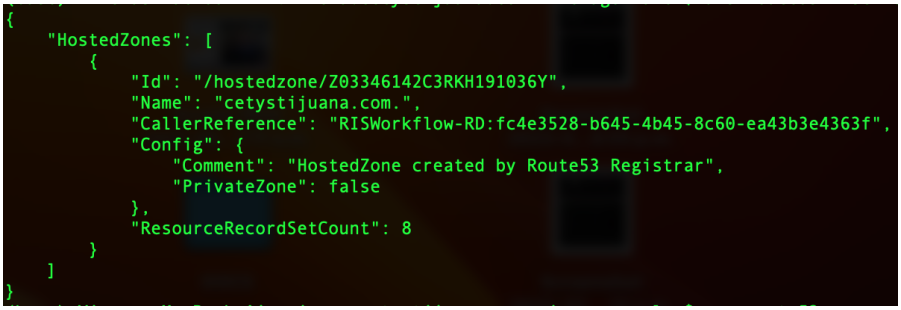
- Using the aws route53 change-resource-record-sets command to apply the changes of the change batch and create the resource record set.



    - The --hosted-zone-id argument is the ID of the hosted zone that contains the resource record sets that want to be changed.
        - This is obtained by running the command aws route53 list-hosted-zones.



- The command lists the public and private hosted zones that



  are associated with the current Amazon Web Services account.
  - The --change-batch argument accepts the JSON file of the change batch, in this case a path to the file is provided.
- Now in the browser the website files are visible through the link ximena.cetystijuana.com

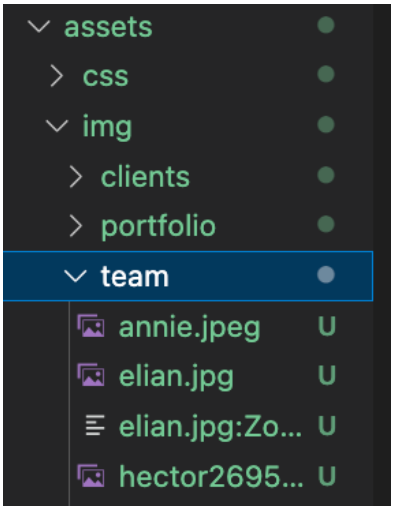
Add yourself to the list of students in cetystijuana.com. 15 points

- Using the aws s3 sync command to sync the contents of the cetystijuana.com bucket to a local directory, in order to be able to download the website's files and make changes to them locally.
    - The source directory is the S3Uri to the bucket.
    - The destination . is the local directory.

```
aws s3 sync s3://cetystijuana.com .
```

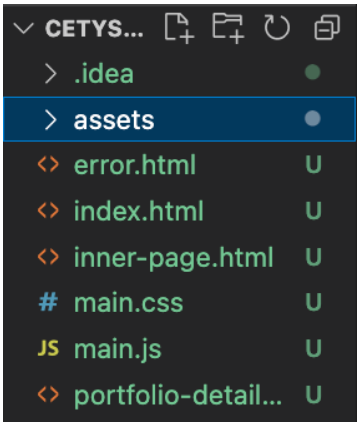
```
download: s3://cetystijuana.com/.idea/aws.xml to .idea/aws.xml
download: s3://cetystijuana.com/.idea/cetyswebsite.iml to .idea/cetyswebsite.iml
download: s3://cetystijuana.com/.idea/modules.xml to .idea/modules.xml
download: s3://cetystijuana.com/.idea/.gitignore to .idea/.gitignore
download: s3://cetystijuana.com/.DS_Store to ./.DS_Store
download: s3://cetystijuana.com/assets/img/apple-touch-icon.png to assets/img/apple-touch-icon.png
download: s3://cetystijuana.com/assets/.DS_Store to assets/.DS_Store
download: s3://cetystijuana.com/.idea/workspace.xml to .idea/workspace.xml
download: s3://cetystijuana.com/assets/img/clients/client-1.png to assets/img/clients/client-1.png
download: s3://cetystijuana.com/assets/img/.DS_Store to assets/img/.DS_Store
download: s3://cetystijuana.com/assets/img/clients/client-3.png to assets/img/clients/client-3.png
download: s3://cetystijuana.com/assets/img/clients/client-2.png to assets/img/clients/client-2.png
download: s3://cetystijuana.com/assets/img/clients/client-5.png to assets/img/clients/client-5.png
download: s3://cetystijuana.com/assets/img/clients/client-4.png to assets/img/clients/client-4.png
download: s3://cetystijuana.com/assets/img/clients/client-6.png to assets/img/clients/client-6.png
download: s3://cetystijuana.com/assets/css/style.css to assets/css/style.css
download: s3://cetystijuana.com/assets/img/favicon.png to assets/img/favicon.png
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-3.jpg to assets/img/portfolio/portfolio-3.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-5.jpg to assets/img/portfolio/portfolio-5.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-4.jpg to assets/img/portfolio/portfolio-4.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-1.jpg to assets/img/portfolio/portfolio-1.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-6.jpg to assets/img/portfolio/portfolio-6.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-9.jpg to assets/img/portfolio/portfolio-9.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-8.jpg to assets/img/portfolio/portfolio-8.jpg
download: s3://cetystijuana.com/assets/img/hero-img.png to assets/img/hero-img.png
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-7.jpg to assets/img/portfolio/portfolio-7.jpg
download: s3://cetystijuana.com/assets/img/team/elian.jpg to assets/img/team/elian.jpg
download: s3://cetystijuana.com/assets/img/portfolio/portfolio-2.jpg to assets/img/portfolio/portfolio-2.jpg
```

- Once with the files downloaded, edit the index. html file to include personal info.
    - Edit the Students Section in the html file.
    - Replace the <img src="assets/img/team/team-3.jpg" class="img-fluid" alt=""> with the path to the student's picture in the directory.
        - The picture has to be uploaded in the assets/img/team/ directory.



- In this case: <img src="assets/img/team/ximena.jpg" class="img-fluid" alt="">

- Replace the <h4>Student 3</h4> with the student's name.
- Replace the Student 3 Website with the website link and the string “Student’s Website”
  - In this case: Ximena's Website
- Replace the Quote with the chosen quote.
  - In this case: "I'm not superstitious, but I'm a little stitious." - Michael Scott
- Replace the   with the link to the student’s linkedin profile.
  - In this case:




```
<div class="col-lg-6">
  <div class="member d-flex align-items-start" data-aos="zoom-in" data-aos-delay="100">
    <div class="pic"><img src="assets/img/team/ximena.jpg" class="img-fluid" alt=""></div>
    <div class="member-info">
      <h4>Ximena González</h4>
      <span><a href="//ximena.cetystijuana.com">Ximena's Website </a></span>
      <p>"I'm not superstitious, but I'm a little stitious." – Michael Scott</p>
      <div class=
        <a href="https://www.linkedin.com/in/ximena-gonz%C3%A1lez-b9b228229/"> <i class="ri-linkedin-box-fill"></i> </a>
      </div>
    </div>
  </div>
</div>
```

Follow link (cmd + click)

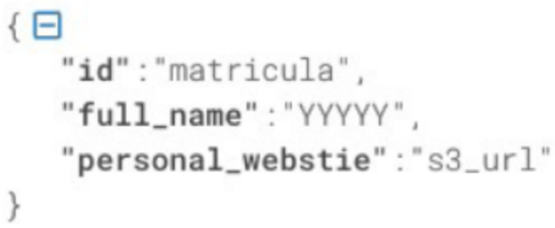
- Once the changes are made and saved, run the same aws s3 sync command to sync and upload the contents of the local directory to the remote bucket.
    - Note how the source directory now is . (local directory), and the destination is the S3Uri to the bucket.

```
aws s3 sync . s3://cetystijuana.com
```

```
upload: ./main.js to s3://cetystijuana.com/main.js
upload: ./main.css to s3://cetystijuana.com/main.css
upload: ./index.html to s3://cetystijuana.com/index.html
upload: assets/img/team/ximena.jpg to s3://cetystijuana.com/assets/img/team/ximena.jpg
```

- Now in the browser the changes made to cetystijuana.com are reflected.

Create an entry in the Students DynamoDB table using the cli with the following model: 15 points.



- Amazon DynamoDB is a cloud base serverless NoSQL database, meaning it uses a non-relational system. It automatically distributes data and traffic over servers to dynamically manage each user's requests. Thanks to its non-relational system, it provides fast performance and scalability.
    - It supports document and key-value data models, in this case the latter is of relevance.
- In order to create an entry, the aws dynamodb put-item command is needed.

```
aws dynamodb put-item --table-name Students --item file:///Users/ximenagonzalez/Desktop/db-entry.json
```

    - The --table-name argument is the name of the table in which to create the entry.
    - The --item argument accepts the JSON file for the table entry, in this case a path to the file is provided.
        - The way to create the JSON file is as follows:

```
{
    "id" : { "S" : "32438"},
    "full_name" : { "S" : "Ximena Lizeth González Plascencia"},
    "personal_website": { "S" : "ximena.cetystijuana.com"}
}
```

            - The key "id" is the name of one of the attributes, and for the value, it has to be described as a name-value pair. The name is the data type, and the value is the data itself. In this case the value is the 32438 string ("S"). The data syntaxis is the same for the other attributes and values.
- Using the aws dynamodb scan command it is confirmed that the entry was created.
    - The --table-name argument is the name of the table to scan.

```
aws dynamodb scan --table-name Students
```

Watch the Not Just Code Monkeys by Martin Fowler and write your comments in your personal static website. 10 points.

- Comments

References

Amazon Web Services. (2021). *How do I create alias records for services hosted in AWS?* https://aws.amazon.com/premiumsupport/knowledge-center/route-53-create-alias-records/

Amazon Web Services. (2021). *How do I create a simple resource record set in Amazon Route 53 using the AWS CLI?* https://aws.amazon.com/premiumsupport/knowledge-center/simple-resource-record-route53-cli/

Amazon Web Services. (2021). *Leveraging the s3 and s3api Commands*. https://aws.amazon.com/blogs/developer/leveraging-the-s3-and-s3api-commands/

Amazon Web Services. (2021). *Leveraging the s3 and s3api Commands*. https://aws.amazon.com/blogs/developer/leveraging-the-s3-and-s3api-commands/

Amazon Web Services. (2021). *Writing IAM Policies: How to Grant Access to an Amazon S3 Bucket*. https://aws.amazon.com/blogs/security/writing-iam-policies-how-to-grant-access-to-an-amazon-s3-bucket/

Amazon Web Services. (n.d.). *Amazon Route 53 concepts*. https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/route-53-concepts.html

Amazon Web Services. (n.d.). *Amazon S3 actions*. https://docs.aws.amazon.com/AmazonS3/latest/userguide/using-with-s3-actions.html

Amazon Web Services. (n.d.). *AWS JSON policy elements: Principal*. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_principal.html

Amazon Web Services. (n.d.). *Bucket naming rules*. https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html

Amazon Web Services. (n.d.). *Buckets overview*. https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingBucket.html

Amazon Web Services. (n.d.). *IAM JSON policy elements: Effect*. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_effect.html

Amazon Web Services. (n.d.). *IAM JSON policy elements: Resource*. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_resource.html

Amazon Web Services. (n.d.). *IAM JSON policy elements: Sid*. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_sid.html

Amazon Web Services. (n.d.). *IAM JSON policy elements: Statement*. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_statement.html

Amazon Web Services. (n.d.). *IAM JSON policy elements: Version*. https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_version.html

Amazon Web Services. (n.d.). *Principals*. https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-bucket-user-policy-specifying-principal-intro.html

Amazon Web Services. (n.d.). *Supported DNS record types*. https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/ResourceRecordTypes.html

Amazon Web Services. (n.d.). *Using Amazon S3 with the AWS CLI*. https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3.html

Amazon Web Services. (n.d.). *Using API-Level (s3api) commands with the AWS CLI - AWS Command Line Interface*. https://docs.aws.amazon.com/cli/latest/userguide/cli-services-s3-apicommands.html

Amazon Web Services. (n.d.). *Using bucket policies*. https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html

Amazon Web Services. (n.d.). *Values specific for simple alias records*. https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-values-alias.html

Amazon Web Services. (n.d.). *What is Amazon Route 53? - Amazon Route 53*. https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html

AWS CLI Command Reference. (n.d.). *change-resource-record-sets*. https://docs.aws.amazon.com/cli/latest/reference/route53/change-resource-record-sets.html

AWS CLI Command Reference. (n.d.). *create-bucket*. https://docs.aws.amazon.com/cli/latest/reference/s3api/create-bucket.html

AWS CLI Command Reference. (n.d.). *dynamodb*. https://docs.aws.amazon.com/cli/latest/reference/dynamodb/index.html

AWS CLI Command Reference. (n.d.). *list-hosted-zones*. https://docs.aws.amazon.com/cli/latest/reference/route53/list-hosted-zones.html

AWS CLI Command Reference. (n.d.). *ls*. https://docs.aws.amazon.com/cli/latest/reference/s3/ls.html

AWS CLI Command Reference. (n.d.). *put-bucket-policy*. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3api/put-bucket-policy.html

AWS CLI Command Reference. (n.d.). *s3*. https://docs.aws.amazon.com/cli/latest/reference/s3/

AWS CLI Command Reference. (n.d.). *s3api*. https://docs.aws.amazon.com/cli/latest/reference/s3api/index.html

AWS CLI Command Reference. (n.d.). *sync*. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3/sync.html

AWS CLI Command Reference. (n.d.). *website*. https://docs.aws.amazon.com/cli/latest/reference/s3/website.html

AWS CLI update CNAME value. (2023, February 2). Stack Overflow. https://stackoverflow.com/questions/75324577/aws-cli-update-cname-value

Logic, R. (2022, June 30). *How to download all files in an S3 Bucket using AWS CLI*. Radish Logic. https://www.radishlogic.com/aws/s3/how-to-download-all-files-in-an-s3-bucket-using-aws-cli/

TutorialsPoint. (2021). *DynamoDB - Overview*. https://www.tutorialspoint.com/dynamodb/dynamodb_overview.htm