Aws Cloud Computing curse

Assignment 1

Submitter: Shlomit Ashkenazi, 323024034

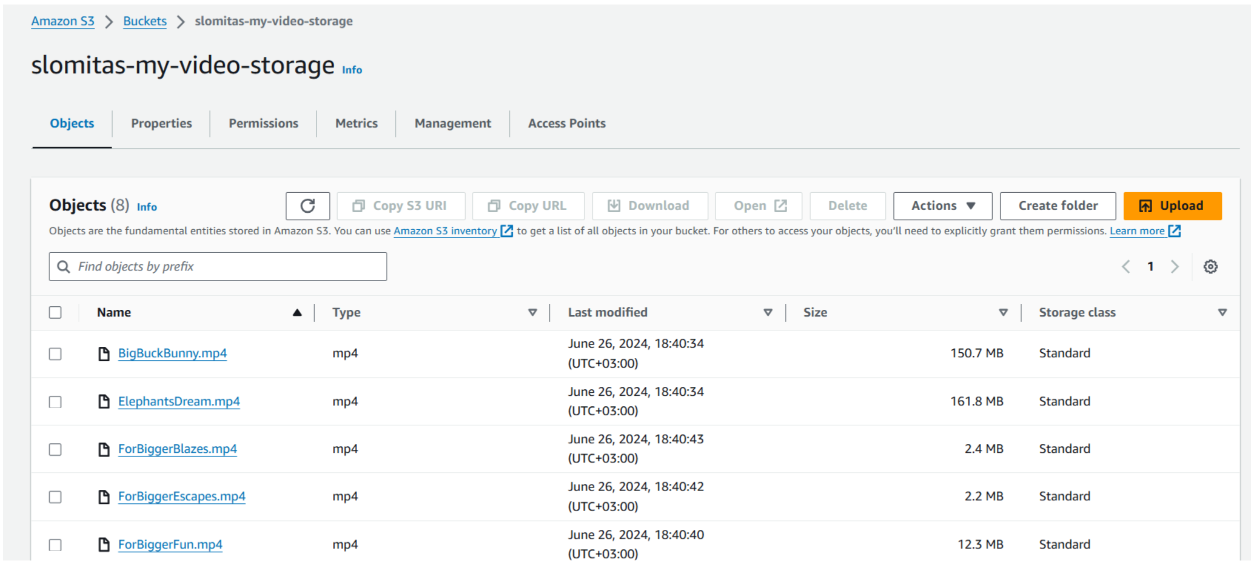
Account ID: 8973-5340-5477

Federated user: voclabs/user3229217=Shlomit_Ashkenaz

Git repository:
https://github.com/slomit1234/cloud_computing_ex1

1st Task - Build the first backend API for your application

I created an S3 Bucket ("slomitas-my-video-storage") and store the videos:



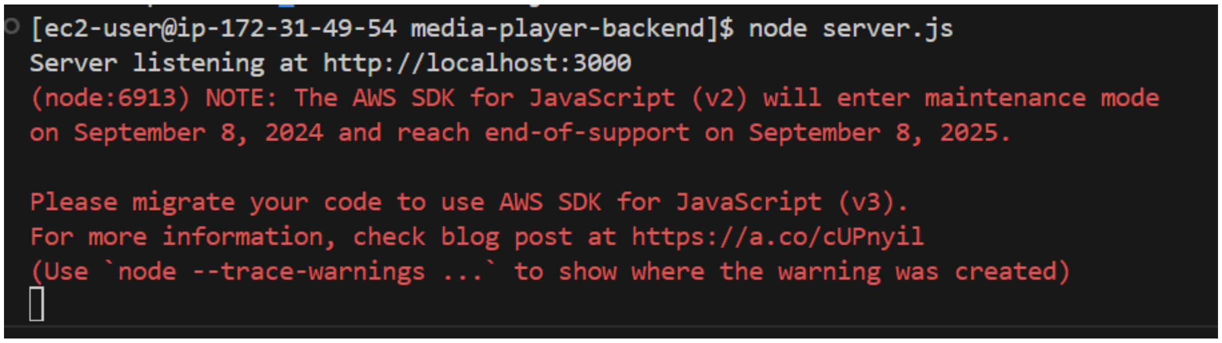
Amazon S3 > Buckets > slomitas-my-video-storage
slomitas-my-video-storage Info
Objects | Properties | Permissions | Metrics | Management | Access Points
Objects (8) Info
Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix

Name	Type	Last modified	Size	Storage class
BigBuckBunny.mp4	mp4	June 26, 2024, 18:40:34 (UTC+03:00)	150.7 MB	Standard
ElephantsDream.mp4	mp4	June 26, 2024, 18:40:34 (UTC+03:00)	161.8 MB	Standard
ForBiggerBlazes.mp4	mp4	June 26, 2024, 18:40:43 (UTC+03:00)	2.4 MB	Standard
ForBiggerEscapes.mp4	mp4	June 26, 2024, 18:40:42 (UTC+03:00)	2.2 MB	Standard
ForBiggerFun.mp4	mp4	June 26, 2024, 18:40:40 (UTC+03:00)	12.3 MB	Standard

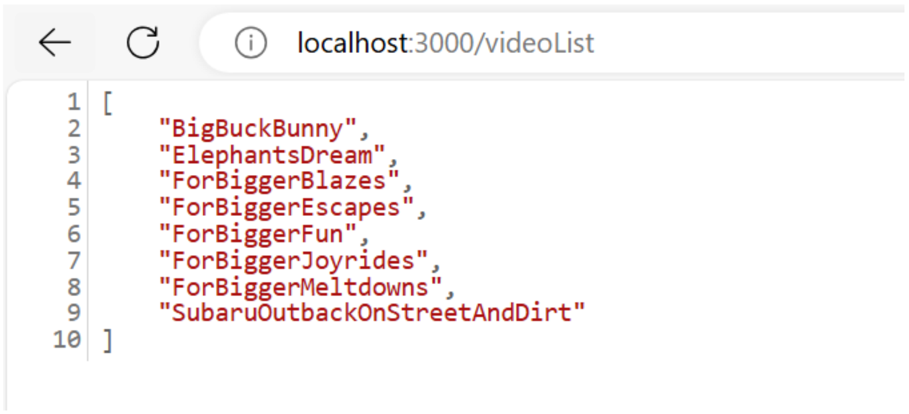
The next stage was creating the API:

I created "server.js" file (called also start_server.js)
then I downloaded node.ls and other dependencies

I used node to run the server:



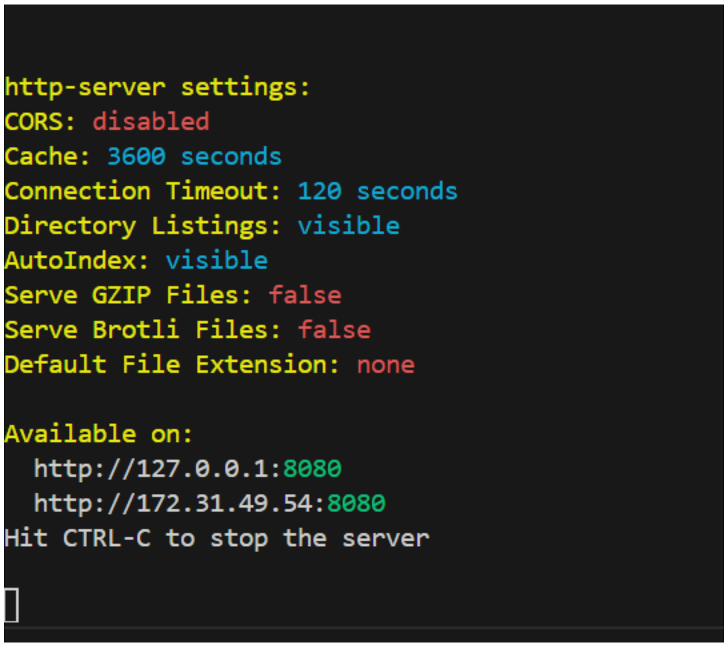
```
[ec2-user@ip-172-31-49-54 media-player-backend]$ node server.js
Server listening at http://localhost:3000
(node:6913) NOTE: The AWS SDK for JavaScript (v2) will enter maintenance mode
on September 8, 2024 and reach end-of-support on September 8, 2025.

Please migrate your code to use AWS SDK for JavaScript (v3).
For more information, check blog post at https://a.co/cUPnyil
(Use `node --trace-warnings ...` to show where the warning was created)
```



```
[
    "BigBuckBunny",
    "ElephantsDream",
    "ForBiggerBlazes",
    "ForBiggerEscapes",
    "ForBiggerFun",
    "ForBiggerJoyrides",
    "ForBiggerMeltdowns",
    "SubaruOutbackOnStreetAndDirt"
]
```

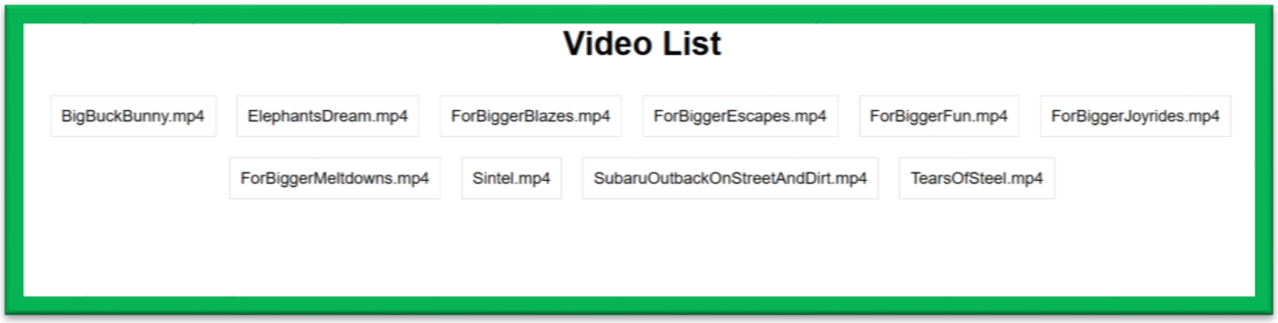
2nd Task - Use the first backend API for your application

I created 3 files (index.html, styles.css, script.js)
and used http-server to run everything:



```
http-server settings:
CORS: disabled
Cache: 3600 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:
  http://127.0.0.1:8080
  http://172.31.49.54:8080
Hit CTRL-C to stop the server
```

On 127.0.0.1:8080:



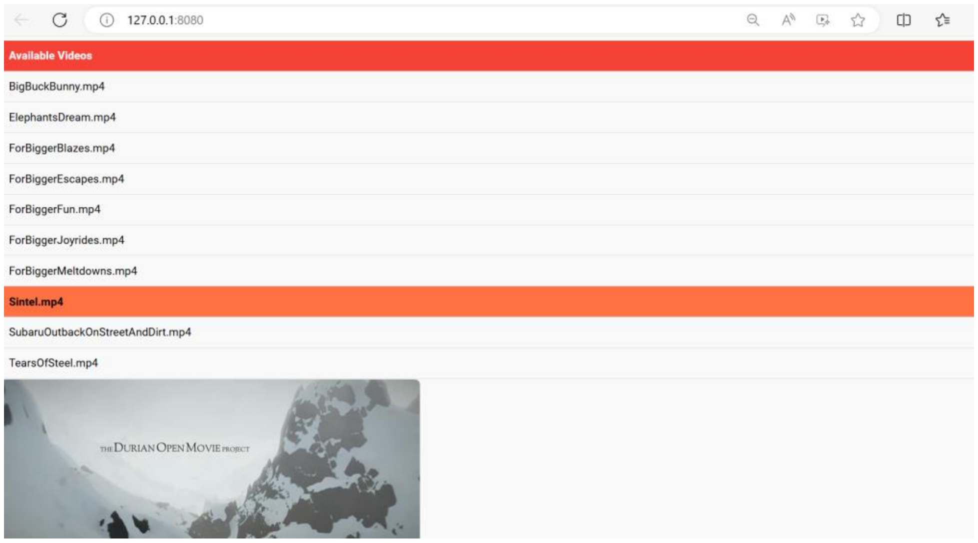
Video List

BigBuckBunny.mp4 ElephantsDream.mp4 ForBiggerBlazes.mp4 ForBiggerEscapes.mp4 ForBiggerFun.mp4 ForBiggerJoyrides.mp4
ForBiggerMeltdowns.mp4 Sintel.mp4 SubaruOutbackOnStreetAndDirt.mp4 TearsOfSteel.mp4

3rd Task - Build the Entire App

Here are the major changes of this step:

1. Most of the server logic the same, getVideoUrl (generating a URL so the frontend will stream the video).
2. I tried to upgrade it style wise (css)
3. On the HTML I added the media player
4. On the javascript I added the handling of the video playing functionality when a video name is clicked



Available Videos

BigBuckBunny.mp4

ElephantsDream.mp4

ForBiggerBlazes.mp4

ForBiggerEscapes.mp4

ForBiggerFun.mp4

ForBiggerJoyrides.mp4

ForBiggerMeltdowns.mp4

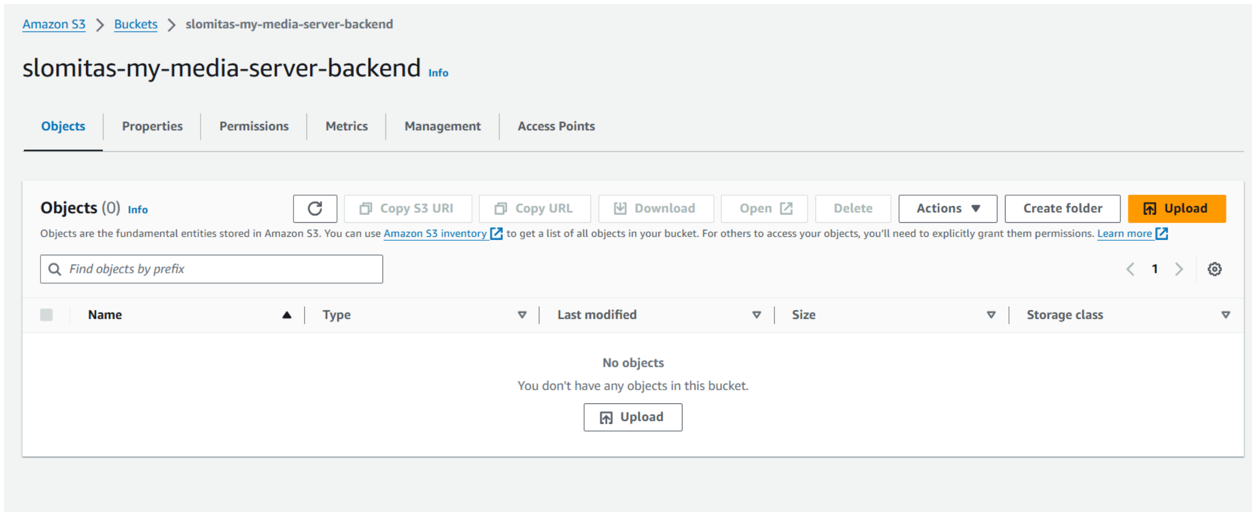
Sintel.mp4

SubaruOutbackOnStreetAndDirt.mp4

TearsOfSteel.mp4

3rd Task - Build the Entire App

I created a new bucket “slomitas-my-media-server-backend”:



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

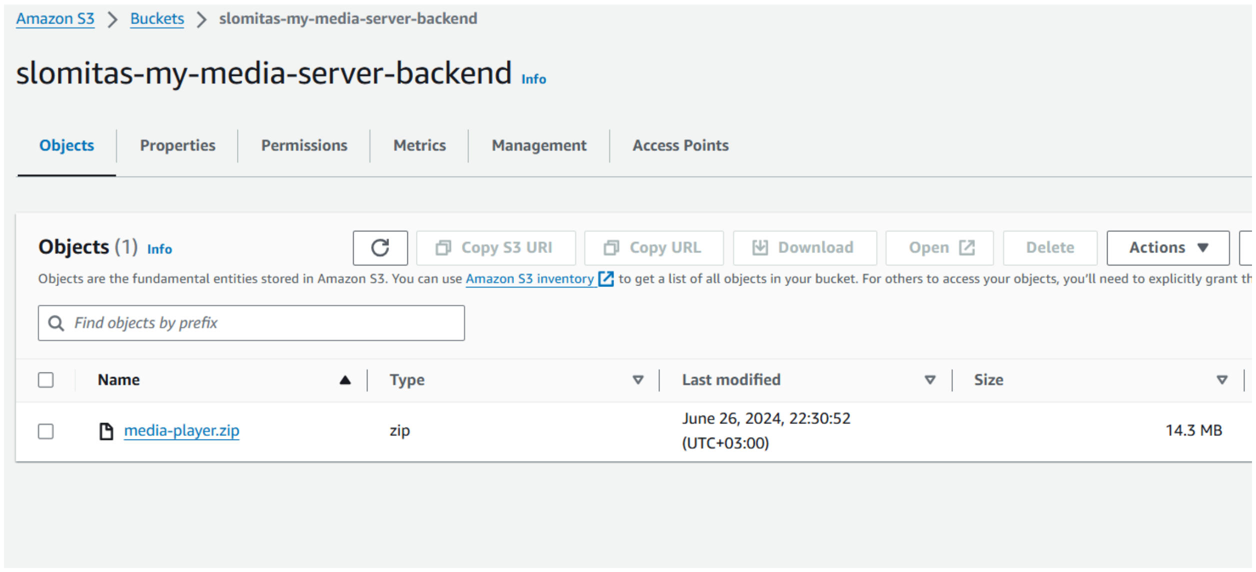
Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.

Upload

Than uploaded the zip:



Amazon S3 > Buckets > slomitas-my-media-server-backend

slomitas-my-media-server-backend Info

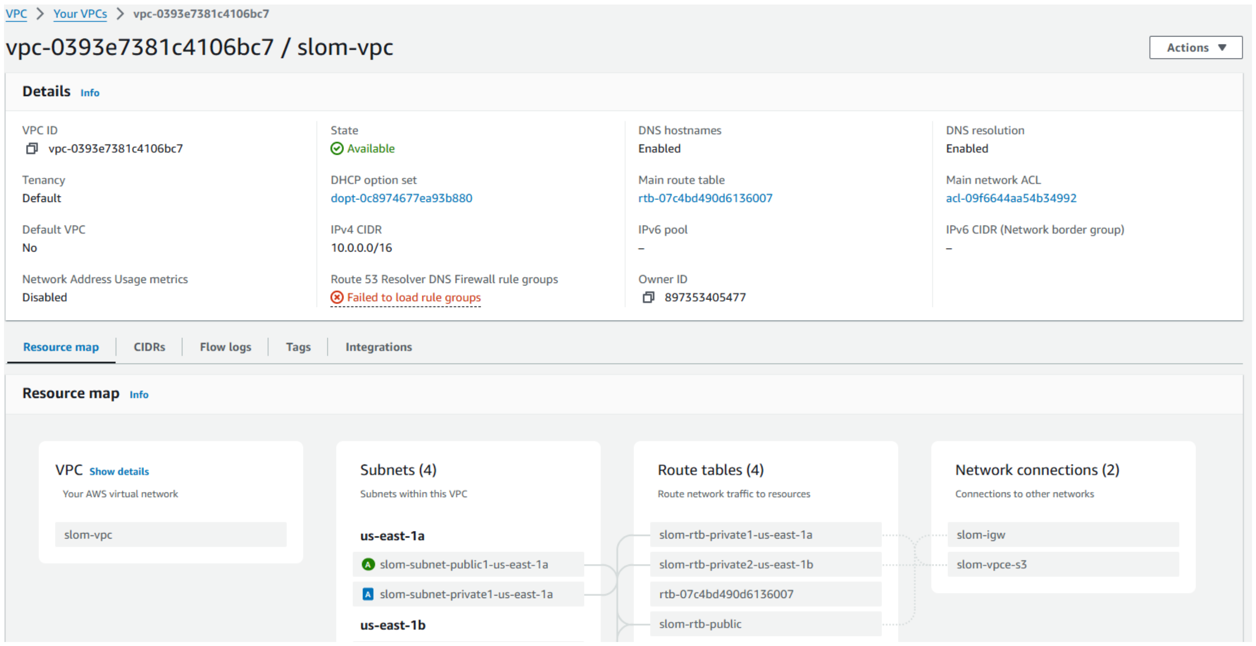
Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant th

Name	Type	Last modified	Size
media-player.zip	zip	June 26, 2024, 22:30:52 (UTC+03:00)	14.3 MB

I created the VPC:



VPC > Your VPCs > vpc-0393e7381c4106bc7

vpc-0393e7381c4106bc7 / slom-vpc

Details Info

VPC ID: vpc-0393e7381c4106bc7	State: Available	DNS hostnames: Enabled	DNS resolution: Enabled
Tenancy: Default	DHCP option set: dopt-0c8974677ea93b880	Main route table: rtb-07c4bd490d6136007	Main network ACL: acl-09f6644aa54b34992
Default VPC: No	IPv4 CIDR: 10.0.0.0/16	IPv6 pool: –	IPv6 CIDR (Network border group): –
Network Address Usage metrics: Disabled	Route 53 Resolver DNS Firewall rule groups: Failed to load rule groups	Owner ID: 897353405477	

Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map Info

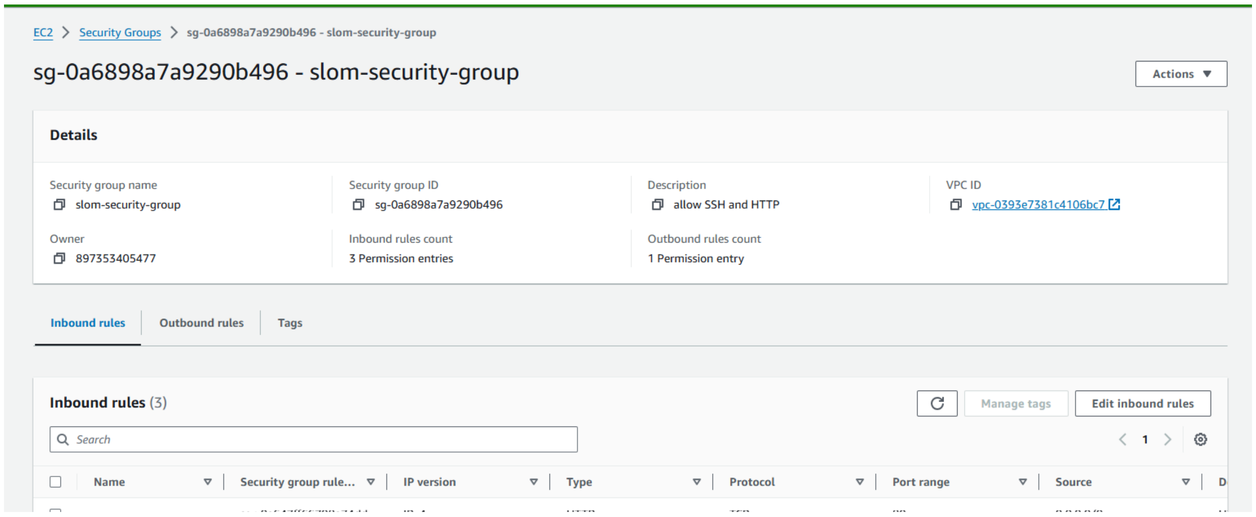
VPC Show details — Your AWS virtual network
- slom-vpc

Subnets (4) — Subnets within this VPC
- us-east-1a
  - slom-subnet-public1-us-east-1a
  - slom-subnet-private1-us-east-1a
- us-east-1b

Route tables (4) — Route network traffic to resources
- slom-rtb-private1-us-east-1a
- slom-rtb-private2-us-east-1b
- rtb-07c4bd490d6136007
- slom-rtb-public

Network connections (2) — Connections to other networks
- slom-igw
- slom-vpce-s3

I created a security group:



EC2 > Security Groups > sg-0a6898a7a9290b496 - slom-security-group

sg-0a6898a7a9290b496 - slom-security-group

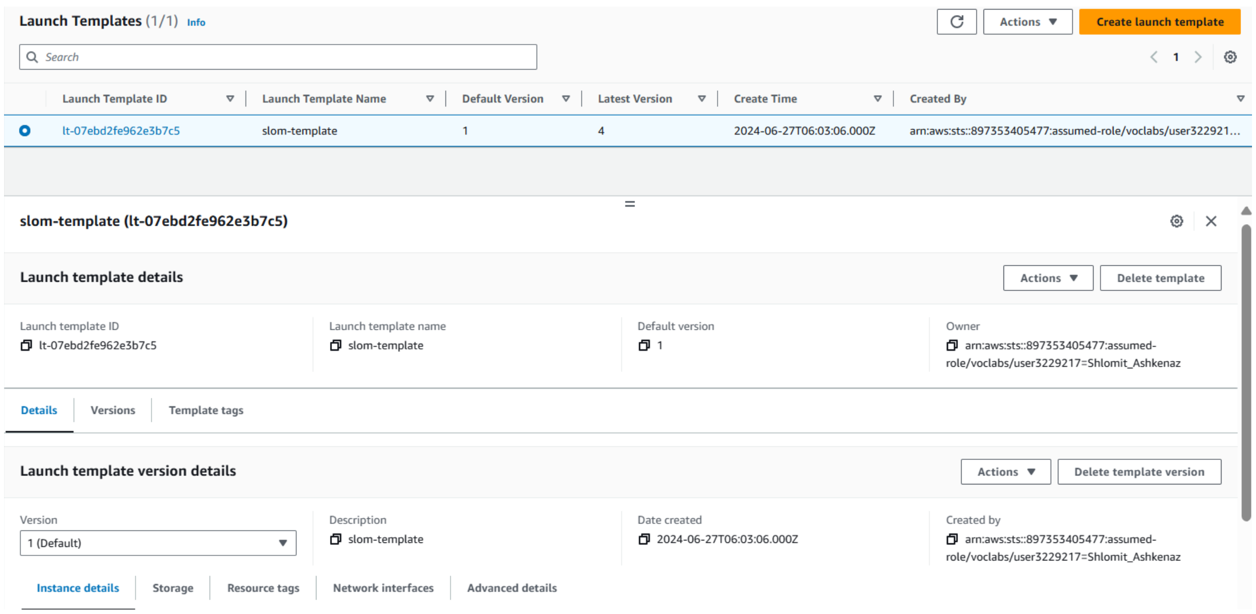
Details

Security group name: slom-security-group	Security group ID: sg-0a6898a7a9290b496	Description: allow SSH and HTTP	VPC ID: vpc-0393e7381c4106bc7
Owner: 897353405477	Inbound rules count: 3 Permission entries	Outbound rules count: 1 Permission entry	

Inbound rules (3)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source

The launch template:



Launch Templates (1/1)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-07ebd2fe962e3b7c5	slom-template	1	4	2024-06-27T06:03:06.000Z	arn:aws:sts::897353405477:assumed-role/voclabs/user322921…

slom-template (lt-07ebd2fe962e3b7c5)

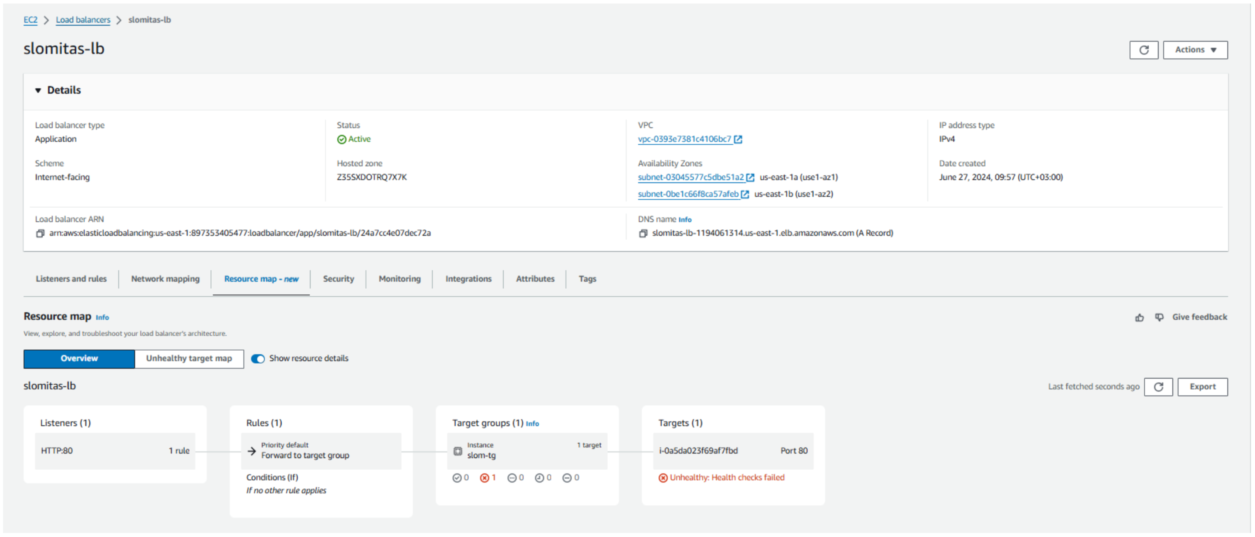
Launch template details

Launch template ID: lt-07ebd2fe962e3b7c5	Launch template name: slom-template	Default version: 1	Owner: arn:aws:sts::897353405477:assumed-role/voclabs/user3229217=Shlomit_Ashkenaz

Launch template version details

Version: 1 (Default)	Description: slom-template	Date created: 2024-06-27T06:03:06.000Z	Created by: arn:aws:sts::897353405477:assumed-role/voclabs/user3229217=Shlomit_Ashkenaz

Load balancer:



EC2 > Load balancers > slomitas-lb

slomitas-lb

Details

Load balancer type Application	Status Active	VPC vpc-0393e7381c4106bc7	IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-03045577c5dbe51a2 us-east-1a (use1-az1) subnet-0be1c66f8ca57afeb us-east-1b (use1-az2)	Date created June 27, 2024, 09:57 (UTC+03:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:897353405477:loadbalancer/app/slomitas-lb/24a7cc4e07dec72a

DNS name Info
slomitas-lb-1194061314.us-east-1.elb.amazonaws.com (A Record)

Resource map Info
View, explore, and troubleshoot your load balancer's architecture.

Show resource details

slomitas-lb

Last fetched seconds ago

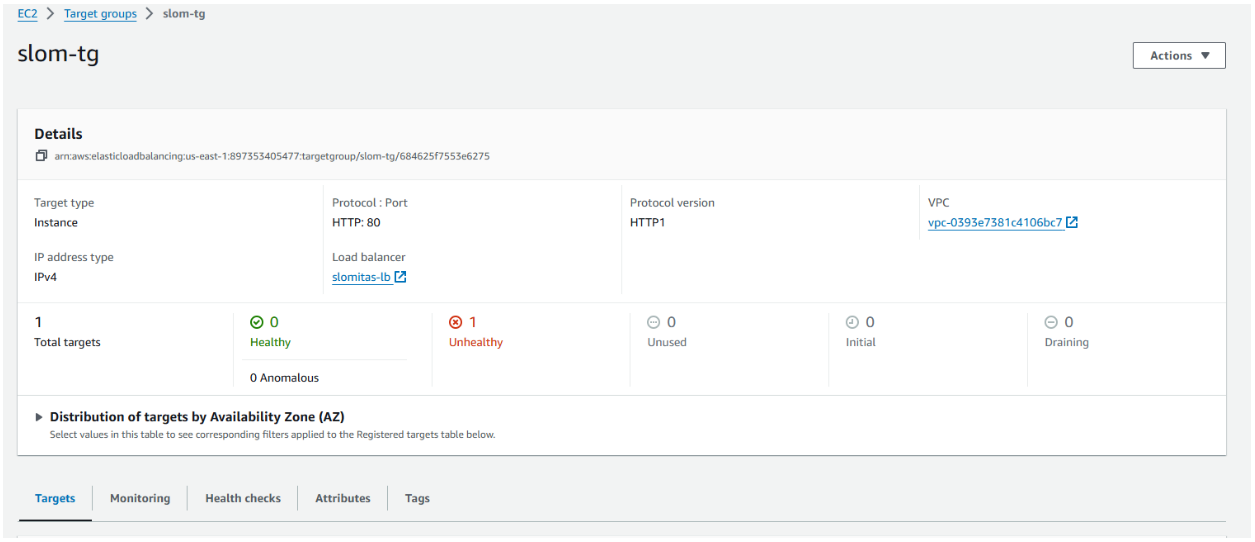
Listeners (1): HTTP:80 — 1 rule

Rules (1): Priority default — Forward to target group; Conditions (If) — *If no other rule applies*

Target groups (1) Info: Instance slom-tg — 1 target; 0 ✓, 1 ✗, 0, 0, 0

Targets (1): i-0a5da023f69af7fbd — Port 80; Unhealthy: Health checks failed

Target group:



EC2 > Target groups > slom-tg

slom-tg

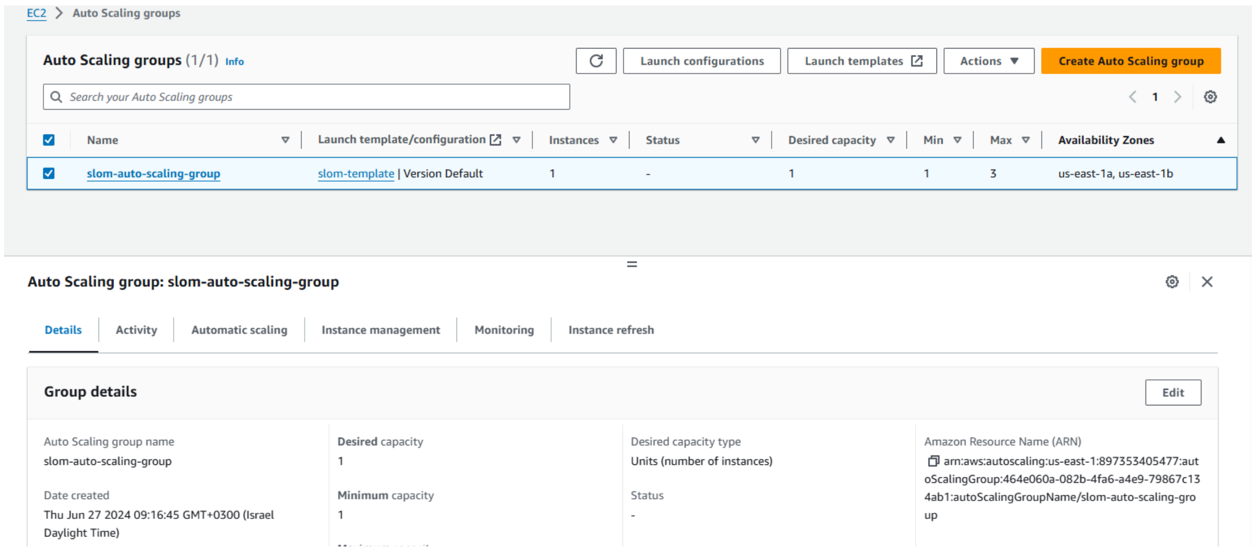
Details
arn:aws:elasticloadbalancing:us-east-1:897353405477:targetgroup/slom-tg/684625f7553e6275

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0393e7381c4106bc7
IP address type IPv4	Load balancer slomitas-lb		

1 Total targets	0 Healthy 0 Anomalous	1 Unhealthy	0 Unused	0 Initial	0 Draining

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Auto-scaling:



EC2 > Auto Scaling groups

Auto Scaling groups (1/1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
slom-auto-scaling-group	slom-template | Version Default	1	-	1	1	3	us-east-1a, us-east-1b

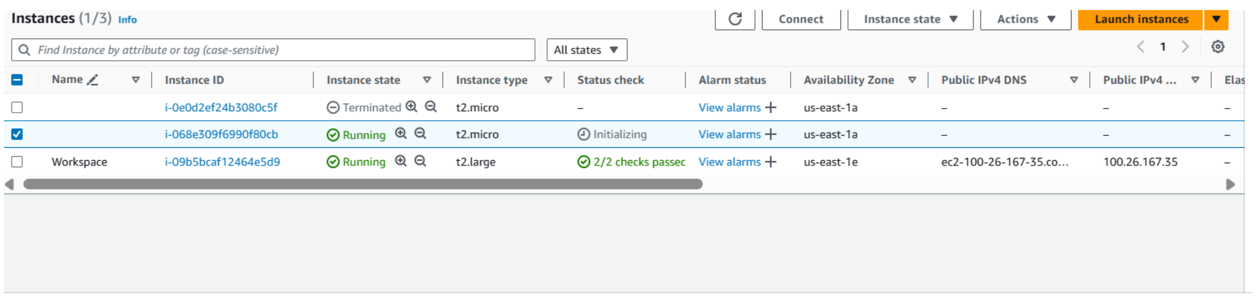
Auto Scaling group: slom-auto-scaling-group

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details

Auto Scaling group name: slom-auto-scaling-group	Desired capacity: 1	Desired capacity type: Units (number of instances)	Amazon Resource Name (ARN): arn:aws:autoscaling:us-east-1:897353405477:autoScalingGroup:464e060a-082b-4fa6-a4e9-79867c134ab1:autoScalingGroupName/slom-auto-scaling-group
Date created: Thu Jun 27 2024 09:16:45 GMT+0300 (Israel Daylight Time)	Minimum capacity: 1	Status: -	

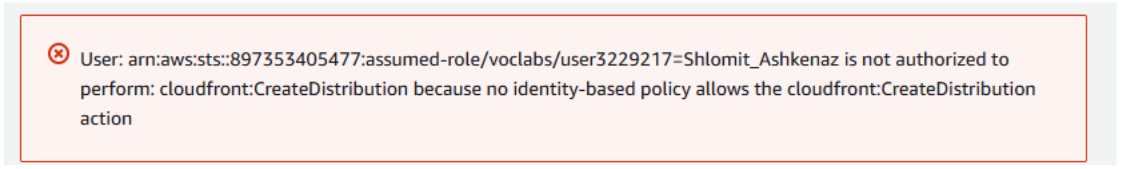
We can see that a new instance was created:



Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elas
	i-0e0d2ef24b3080c5f	Terminated	t2.micro	–	View alarms +	us-east-1a	–	–	–
	i-068e309f6990f80cb	Running	t2.micro	Initializing	View alarms +	us-east-1a	–	–	–
Workspace	i-09b5bcaf12464e5d9	Running	t2.large	2/2 checks passec	View alarms +	us-east-1e	ec2-100-26-167-35.co...	100.26.167.35	–

All of those screenshots made after I did 5th and 6th part.
the updated server.js needs to contain the key-pair-id – which I could not create because of a problem with cloudfront premmisions:

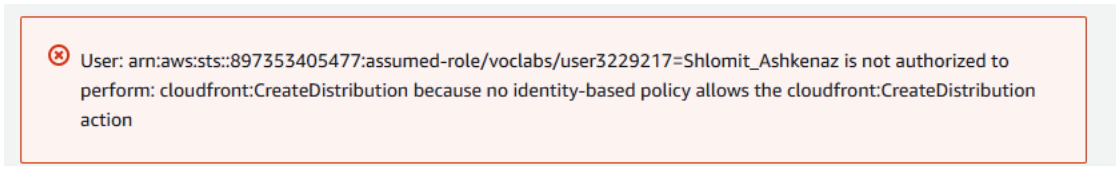


User: arn:aws:sts::897353405477:assumed-role/voclabs/user3229217=Shlomit_Ashkenaz is not authorized to perform: cloudfront:CreateDistribution because no identity-based policy allows the cloudfront:CreateDistribution action

(I did a lot of research on this)
on the AWS API – I also couldn't do since the key-pair-id needs to be created manually and it's creation is not supported by the API.
so I just left it as is.

5th Task - Using CDN for Media File Efficient Distribution

I had a problem with the cloudfront premissions while using the GUI, so I did this part on with the CLI.


User: arn:aws:sts::897353405477:assumed-role/voclabs/user3229217=Shlomit_Ashkenaz is not authorized to perform: cloudfront:CreateDistribution because no identity-based policy allows the cloudfront:CreateDistribution action

So first of all I created a distribution for the “slomitas-my-video-storage”

```
  [ec2-user@ip-172-31-49-54 server-media-player]$ aws cloudfront create-
distribution --origin-domain-name slomitas-my-video-storage.s3.amazonaws.com --
default-root-object index.html
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-
31/distribution/E3DQX1LITSQI5O",
    "ETag": "E3U2HFSHI0ZSV4",
    "Distribution": {
        "Id": "E3DQX1LITSQI5O",
        "ARN": "arn:aws:cloudfront::897353405477:distribution/E3DQX1LITSQI5O",
        "Status": "InProgress",
        "LastModifiedTime": "2024-06-26T23:18:00.795000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d1pkskx0yceldq.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1719443880-262435",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "slomitas-my-video-storage.s3.amazonaws.com-
1719443880-915322",
                        "DomainName": "slomitas-my-video-
storage.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "slomitas-my-video-storage.s3.amazonaws.com-
1719443880-915322",
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "MinTTL": 0,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "SSLSupportMethod": "vip",
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true,
            "ContinuousDeploymentPolicyId": "",
            "Staging": false
        }
    }
}
(END)
```

Next stage is to creat an OAI:

```
  [ec2-user@ip-172-31-49-54 server-media-player]$ aws cloudfront create-cloud-
front-origin-access-identity --cloud-front-origin-access-identity-config
CallerReference=unique-string,Comment="OAI for CloudFront"
{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/origin-access-
identity/cloudfront/E106KVZIMZV0IL",
    "ETag": "E27JH8FY73JS3S",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E106KVZIMZV0IL",
        "S3CanonicalUserId":
"dc20fefe8facdefcc0cb7657afd17cf8d49d4fb4440c78423b492fc056146ce6f29f08cd41233927
ab9cd8e9469cc6bc",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "unique-string",
            "Comment": "OAI for CloudFront"
        }
    }
}
[ec2-user@ip-172-31-49-54 server-media-player]$
```

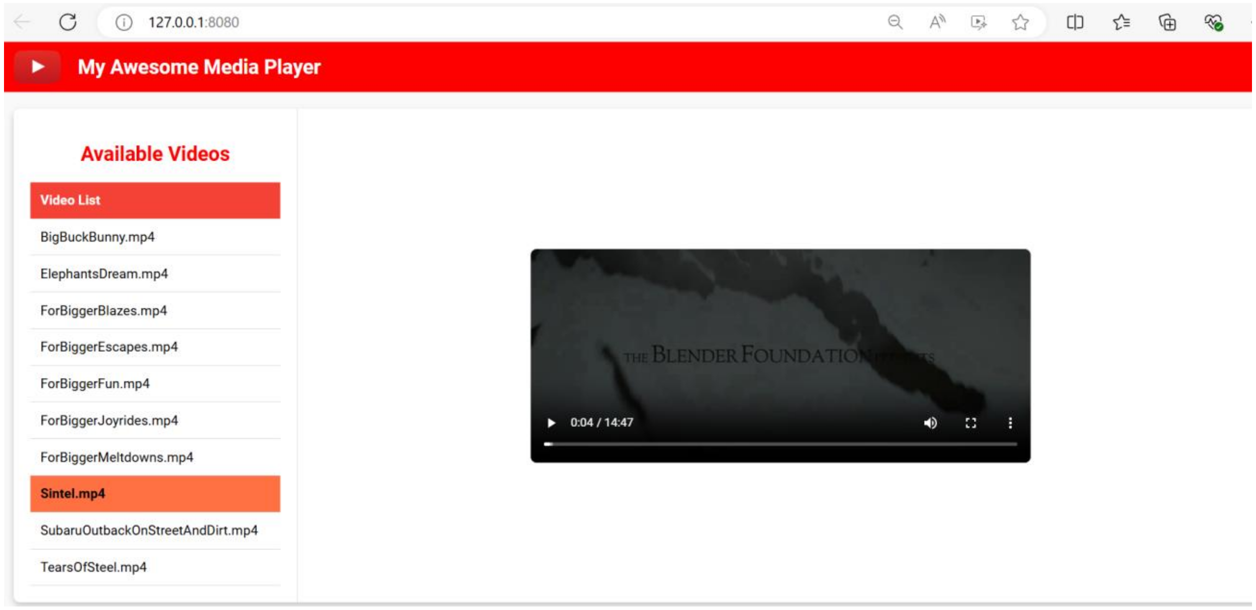
Than update the bucket policy:

```
  [ec2-user@ip-172-31-49-54 server-media-player]$ cat > bucket-policy.json <<EOL
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GrantCloudFrontAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
Identity E106KVZIMZV0IL"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::slomitas-my-video-storage/*"
        }
    ]
}
EOL
```

```
  [ec2-user@ip-172-31-49-54 server-media-player]$ aws s3api put-bucket-policy --
bucket slomitas-my-video-storage --policy file://bucket-policy.json
```

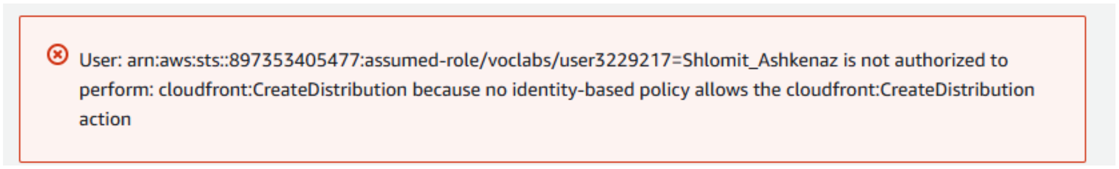
Lastly we modified server.js to use the distribution.



We can see that the client is working (only in this part the videos come from the cloudfront CDN)

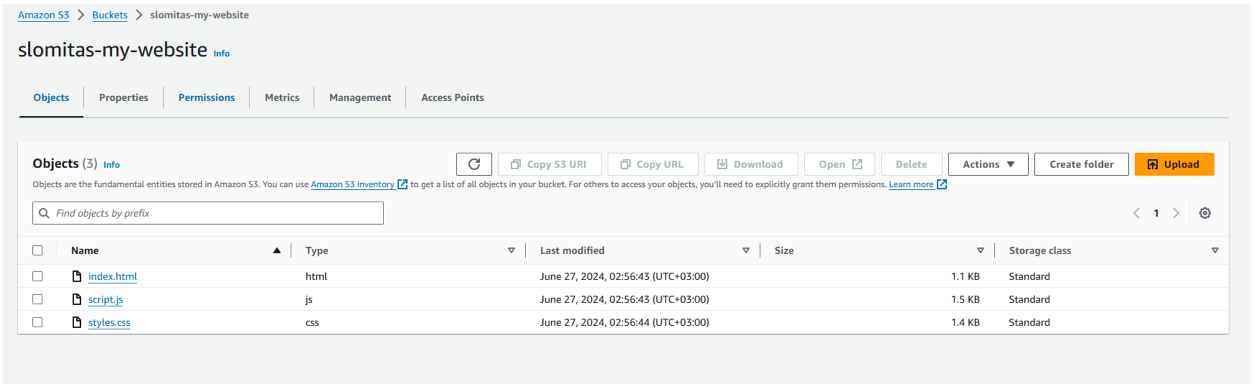
6th Task - Deploy your static website to the internet

since this part also include “CloudFront” I will continue with the CLI



User: arn:aws:sts::897353405477:assumed-role/voclabs/user3229217=Shlomit_Ashkenaz is not authorized to perform: cloudfront:CreateDistribution because no identity-based policy allows the cloudfront:CreateDistribution action

The first part is to create a new bucket and enter all of the website relevant data to it



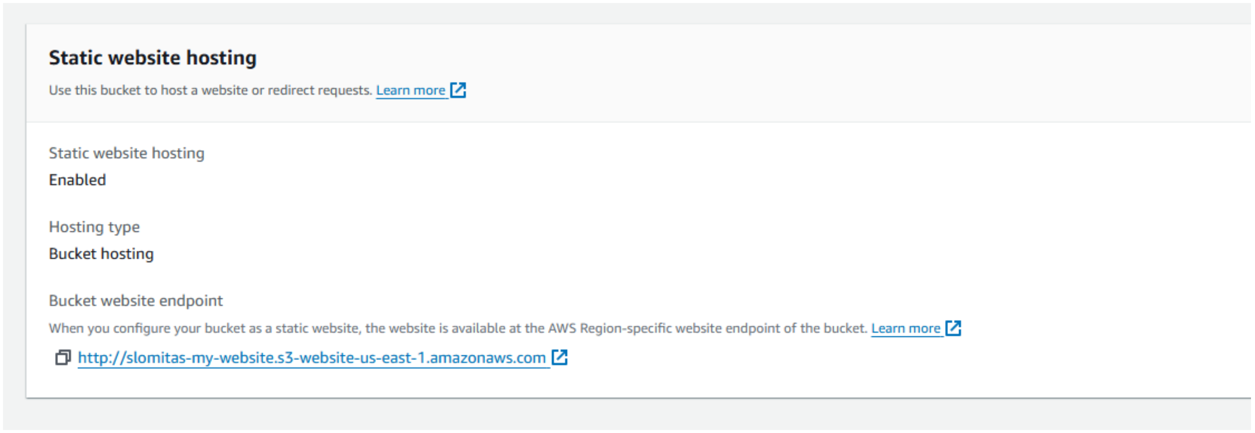
slomitas-my-website

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
index.html	html	June 27, 2024, 02:56:43 (UTC+03:00)	1.1 KB	Standard
script.js	js	June 27, 2024, 02:56:43 (UTC+03:00)	1.5 KB	Standard
styles.css	css	June 27, 2024, 02:56:44 (UTC+03:00)	1.4 KB	Standard

We will configure S3 Bucket for Static Website Hosting:



Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://slomitas-my-website.s3-website-us-east-1.amazonaws.com

Now we will deploy a distribution with the AWS CLI:

```
[ec2-user@ip-172-31-49-54 client-media-player]$ aws cloudfront create-
distribution --origin-domain-name slomitas-my-website.s3.amazonaws.com

{
    "Location": "https://cloudfront.amazonaws.com/2020-05-
31/distribution/E2MMGDLDH778M6",
    "ETag": "E2P5Y8HLY67S7G",
    "Distribution": {
```

```
ion/E2MMGDLDH778M6",
        "Status": "InProgress",
        "LastModifiedTime": "2024-06-27T00:02:15.358000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d36tr60kmj96gq.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1719446535-261996",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "slomitas-my-website.s3.amazonaws.com-1719446535-
598536",
                        "DomainName": "slomitas-my-website.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "slomitas-my-website.s3.amazonaws.com-
1719446535-59853
6",
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": "",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "MinTTL": 0,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": true,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "SSLSupportMethod": "vip",
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true,
            "ContinuousDeploymentPolicyId": "",
            "Staging": false
        }
    }
}
(END)
```

Now we will create the OAI:

```
[ec2-user@ip-172-31-49-54 client-media-player]$ aws cloudfront create-cloud-
front-origin-access-identity --cloud-front-origin-access-identity-config
CallerReference=unique-string-2,Comment="OAI for Static Website"

{
    "Location": "https://cloudfront.amazonaws.com/2020-05-31/origin-access-
identity/cloudfront/E3Q0FT0MKQMZ8J",
    "ETag": "E2VS66GZS81CWV",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E3Q0FT0MKQMZ8J",
        "S3CanonicalUserId":
"ff342a2b65b2974f06d2a96e5abb51a9256bd3394302351c9198bca3da1d2b45a4b7c15fa6f8c158
eb54bdee8928dffe",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "unique-string-2",
            "Comment": "OAI for Static Website"
        }
    }
}
```

Take the bucket policy:

```
{

    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity
E3Q0FT0MKQMZ8J"
        },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::slomitas-my-website/*"
      }
    ]
  }
```

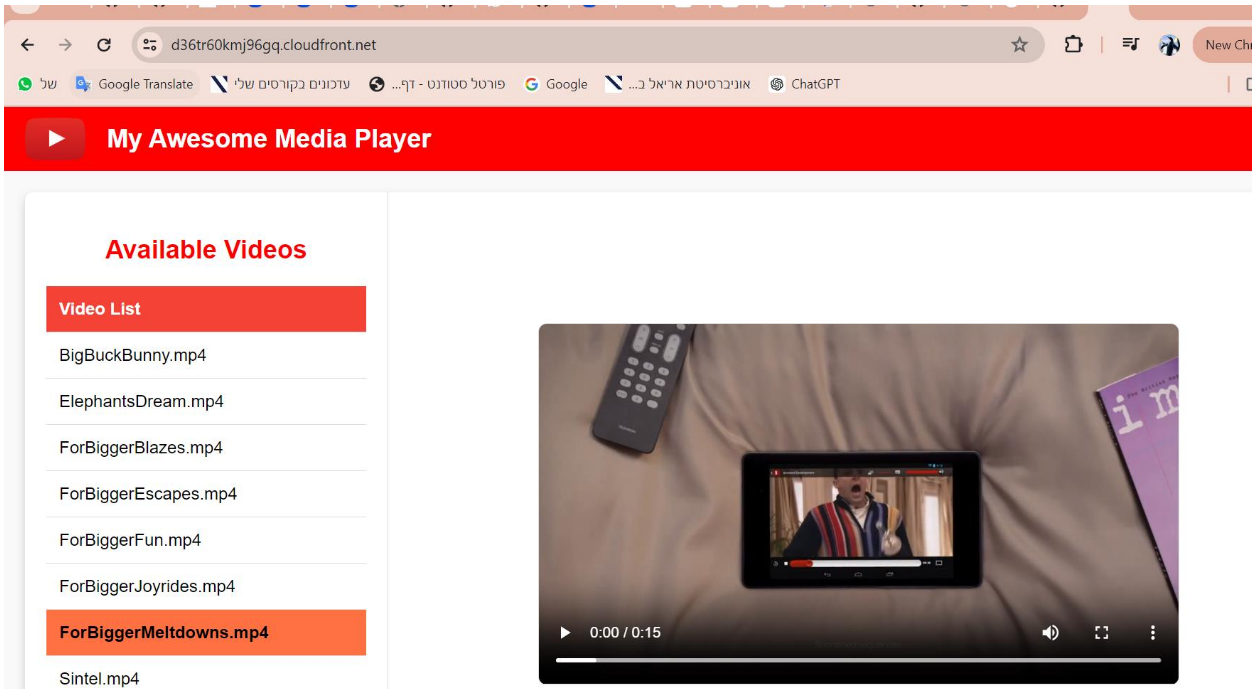
And deploy:

```
  [ec2-user@ip-172-31-49-54 client-media-player]$ aws s3api put-bucket-policy --
bucket slomitas-my-website --policy file://bucket-policy.json
```

```
After config:


  [ec2-user@ip-172-31-49-54 client-media-player]$ aws cloudfront update-
distribution --id E2MMGDLDH778M6 --if-match E2P5Y8HLY67S7G --distribution-config
file://new-config.json
{
    "ETag": "E16S2S9ZB0MFWG",
    "Distribution": {
        "Id": "E2MMGDLDH778M6",
        "ARN": "arn:aws:cloudfront::897353405477:distribution/E2MMGDLDH778M6",
        "Status": "InProgress",
        "LastModifiedTime": "2024-06-27T00:21:28.110000+00:00",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d36tr60kmj96gq.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "ActiveTrustedKeyGroups": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1719446535-261996",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "slomitas-my-website.s3.amazonaws.com-1719446535-
598536",
                        "DomainName": "slomitas-my-website.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": "origin-access-
identity/cloudfront/E
3Q0FT0MKQMZ8J"
                        },
                        "ConnectionAttempts": 3,
                        "ConnectionTimeout": 10,
                        "OriginShield": {
                            "Enabled": false
                        },
                        "OriginAccessControlId": ""
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId": "slomitas-my-website.s3.amazonaws.com-
1719446535-59853
6",
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "TrustedKeyGroups": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FunctionAssociations": {
```

```
                "Quantity": 0
            },
            "FieldLevelEncryptionId": "",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "MinTTL": 0,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "SSLSupportMethod": "vip",
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true,
            "ContinuousDeploymentPolicyId": "",
            "Staging": false
        }
    }
}
(END)
```

now the frontend side will work from the CDN:



d36tr60kmj96gq.cloudfront.net

My Awesome Media Player

Available Videos

Video List

BigBuckBunny.mp4

ElephantsDream.mp4

ForBiggerBlazes.mp4

ForBiggerEscapes.mp4

ForBiggerFun.mp4

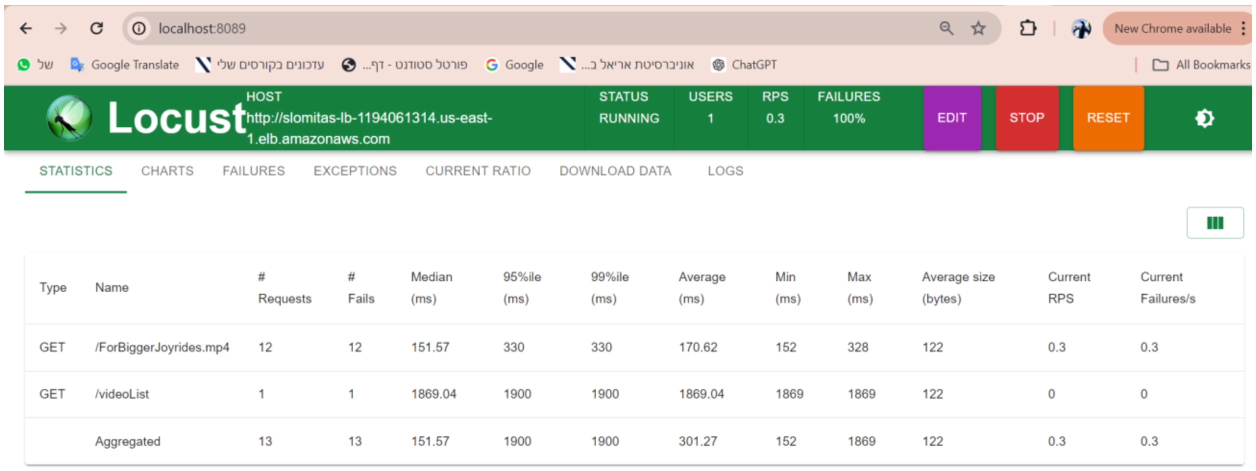
ForBiggerJoyrides.mp4

ForBiggerMeltdowns.mp4

Sintel.mp4

7th Task - Test your backend scalability

I used locust to crate the test (you can install with pip install locust)
than run: locust -f locustfile.py --host http://slomitas-lb-1194061314.us-east-1.elb.amazonaws.com



Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/ForBiggerJoyrides.mp4	12	12	151.57	330	330	170.62	152	328	122	0.3	0.3
GET	/videoList	1	1	1869.04	1900	1900	1869.04	1869	1869	122	0	0
	Aggregated	13	13	151.57	1900	1900	301.27	152	1869	122	0.3	0.3

as you can see because of the problem I had, I could not test everything properly. So there was no point to attach screenshots of everything.

On a personal note ☺ ..

This task was really fun and interesting.

Although I didn't perform everything on the "right way" I hope you would appreciate my creativity on using the API (which was so much harder btw).

I wanted to write to you on it, but as you know I submitted this already in late. Also, the fact that I worked on this task mostly at night (when no one was up) made me to try and face it myself.