

# ARTH Task 6 : Creating a high availability architecture with AWS CLI



First we need to install and login to the AWS cli console using the **Access ID** and **Secret ID** provided by **IAM** user

To check AWS cli Version : **aws — version**

To login to IAM user using AWS cli : **aws configure**

#### Command Prompt

```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

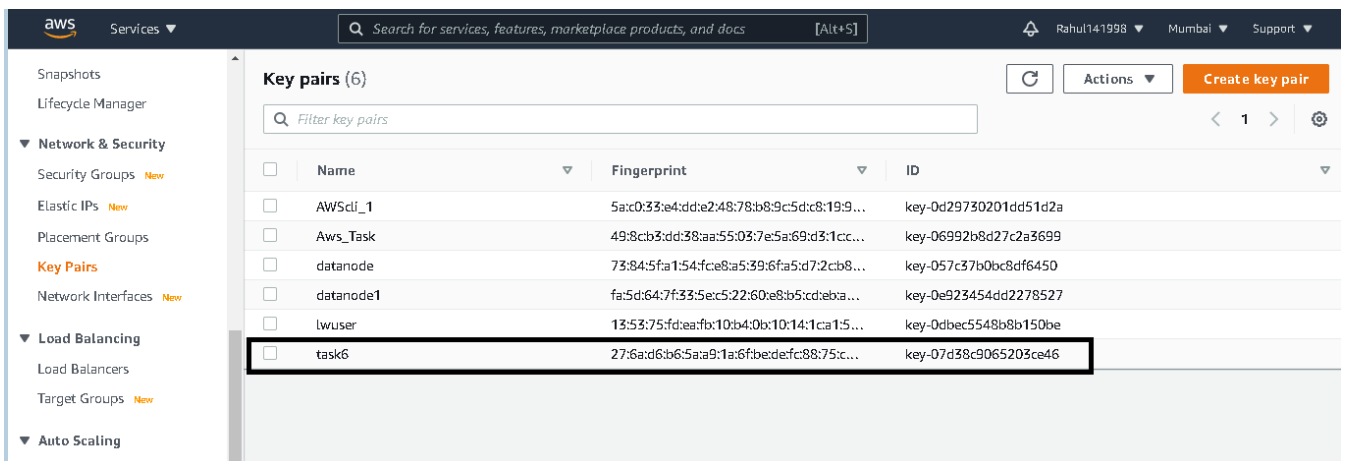
C:\Users\tanmo>aws --version
aws-cli/2.0.30 Python/3.7.7 Windows/10 botocore/2.0.0dev34

C:\Users\tanmo>aws configure
AWS Access Key ID [None]: AKIAVPRFFJDZTTMASXGQ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1a
Default output format [None]:
```

Now we need a key for EC2 instance Authentication

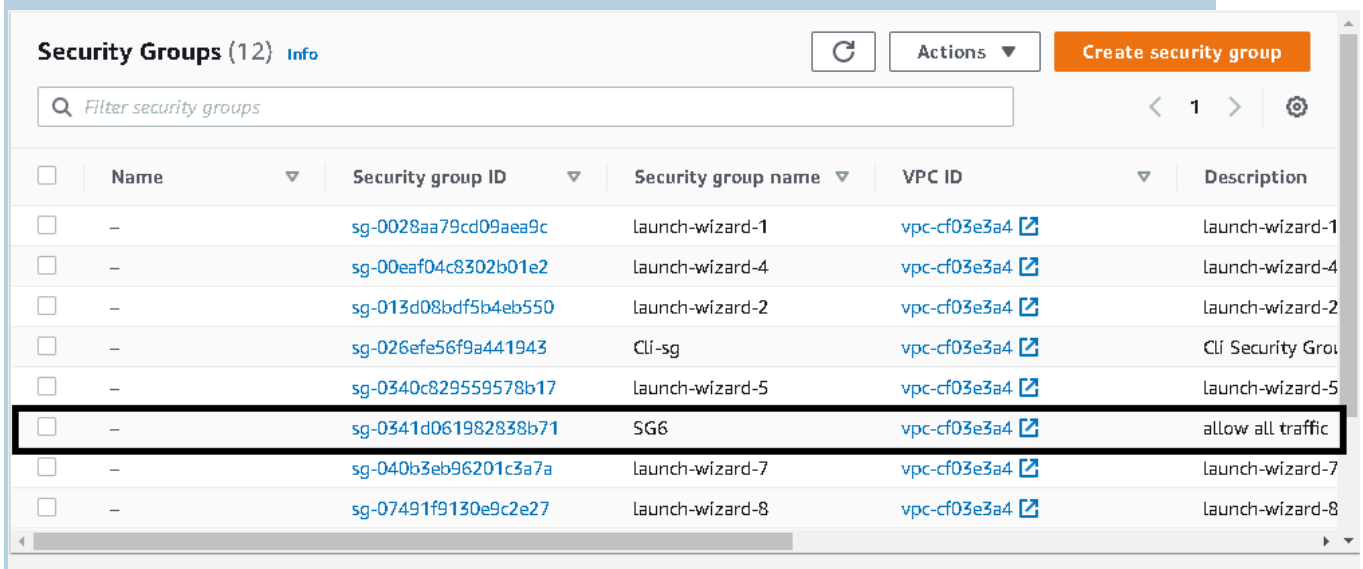
### aws ec2 create-key-pair — keyname task 6

```
C:\Users\tanmo>aws ec2 create-key-pair --key-name task6
{
  "KeyFingerprint": "27:6a:d6:b6:5a:a9:1a:6f:be:de:fc:88:75:cd:11:25:b1:29:17:d4",
  "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEAjTBsy+/i0Dw6q+lnQn\n0BNG5kxwFPackVCvKfp30I0Cj4dbEcrgXL9cn5RXl20rWDJ/ctQ0fr3KIxUZrKm\nnSnZYwTz2xVYdREAeDuRAiM\nMzxLqhLZx\nnEiOoOZOx+mB3wuyL0IntDWQ4+RJwDRBVRmbhOKLHypGjP5cIEU1MGFum1YxX88PD\nngdImHLn8NH\nrrdHdkig53hGeVrZJYiX2\nnMB9wp6AgkwmtxdKFJz/P4/IajxTikUkfI25Q0wIDAQABAoIBADsjjH7DxVKJ192I\nQYf/l+CZWPCocnAR/Rfv0lH8ioN+lcveW\nnu/9l5gleJcEMKRdyWMLIr1qyNnMyjGvT6tmHxtdyLL8tueaLoMuG\nfIt1IrmRlIC/1aap+NYUXxHM6Nv900UB6aygwsGtU3JD\nnDIhxWtz7p9neg1dDgcw0VgFWPEjVHs2oIMTFpfKv\nZHxVbk4q3Ye1egHOnqQ4LCKbqlNiKVvk4jKSmx4S8I1YA9hJTrA+VX/IJ\nneDSPSHECgYEA7gobp8tuIhRoq89s\n5eY\nnv+GBGvvaDnoQwwS7+rOK3ygENiKrXOUSjvcs1WZ+rpoeKYog8n/2ezaV9rxeHD8l\nnilu00MsN08iXBlAE\nPrHH8MCgYEA19ew\nnjtYwSG++a170QJrMJppMz8E+TwgM6mnJu/XoIq1PfWmfsPQl/+V9/ps7JFjEGMq+\nnpYp8\nAm4Hn8LtDqUmC3gx1ZdMK8KgXaA\nnbXpuLxfAPcoq+W6b+dd+I0VwvI/X2/9uHCQzPikCgYEAiVkpYfj2haMQsS\nc3uKxwIso3/iDd9yXhtaGI9NVLZQwvj2Y9IKMX8\nnncc1P7nFQD7uWSYxsiKPTsRBrXmwICgl0y88y1+RRRIZSj\nPqlKe/Xjw0CgYAEU/TKpXhC2EyJI606o67l+sfttAheoFAd7l39\nnqYiwVFhLrtMKbCsZHCv30F17L7yHKGa+v6\nD18Bhr1Y6GlxXwu0L/zbpBbAHvYLNSESlURqo+kMzr+WRTBDk9eGgCYKpFwh9tP\nnmgiL4QKBgQCEp6Azi0QekL\nGHSdhIYE3\nnCDqHbN02LCOB5o1wyCVUX2JvxXMJX704aHDFgkVYsWxW2J4rG3tEg21+0VHG1ln8\nnAT1hyww7rL\nCHxi9OSHImQ==\n-----END RSA PRIVATE KEY-----",
  "KeyName": "task6",
  "KeyPairId": "key-07d38c9065203ce46"
}
```



Now we need to create a security group : **aws ec2 create-security-group — description “Allow all traffic” — group-name SG6**

```
C:\Users\tanmo>aws ec2 create-security-group --description "allow all traffic" --group-
{
  "GroupId": "sg-0341d061982838b71"
}
```



Also we can verify our security group using command : **aws ec2 authorize-security-group — ingress — group-id**

[security-group id] — protocol [protocolname] — port [Number] — cidr o.o.o.o/o

```
C:\Users\tanmo>aws ec2 authorize-security-group-ingress --group-id sg-0341d061982838b71 --protocol tcp --port 80 --cidr 0.0.0.0/0

C:\Users\tanmo>
```

sg-0341d061982838b71 - SG6

Actions

Details

Security group name SG6	Security group ID sg-0341d061982838b71	Description allow all traffic	VPC ID vpc-cf03e3a4
Owner 376961321203	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules

Outbound rules

Tags

Inbound rules (1)

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-

To view the security group details : **aws ec2 describe-security-group — group-id [security group id]**

```
C:\Users\tanmo>aws ec2 describe-security-groups --group-id sg-0341d061982838b71
{
  "SecurityGroups": [
    {
      "Description": "allow all traffic",
      "GroupName": "SG6",
      "IpPermissions": [
        {
          "FromPort": 80,
          "IpProtocol": "tcp",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "ToPort": 80,
          "UserIdGroupPairs": []
        }
      ],
      "OwnerId": "376961321203",
      "GroupId": "sg-0341d061982838b71",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": [],
          "UserIdGroupPairs": []
        }
      ],
      "VpcId": "vpc-cf03e3a4"
    }
  ]
}
```

Now launching instance : **aws ec2 run-instances — security-group[sg id] — instance-type [type] — count [number] — image-id [Image id] — key-name [name]**

```

C:\Users\tanmo>aws ec2 run-instances --security-group-ids sg-0341d061982838b71 --instance-type t2.micro --count 1 --image-id ami-0a9d27a9f4f5c0efc --key-name task6
C:\Users\tanmo>aws ec2 run-instances --security-group-ids sg-0341d061982838b71 --instance-type t2.micro --count 1 --image-id ami-0a9d27a9f4f5c0efc --key-name task6
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0a9d27a9f4f5c0efc",
      "InstanceId": "i-0fd62c91313c2e733",
      "InstanceType": "t2.micro",
      "KeyName": "task6",
      "LaunchTime": "2021-03-15T07:49:42+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "ap-south-1b",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-4-48.ap-south-1.compute.internal",
      "PrivateIpAddress": "172.31.4.48",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-57c4bb1b",
      "VpcId": "vpc-cf03e3a4",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "5d081603-53a3-4d30-9833-42af79786af2",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2021-03-15T07:49:42+00:00",
            "AttachmentId": "eni-attach-0a1b9fac3b2175839",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching"
          }
        }
      ]
    }
  ]
}

```

aws Services Search for services, features, marketplace products, and docs [Alt+S] Rahul141998 Mumbai Support

New EC2 Experience Tell us what you think

EC2 Dashboard Events Tags Limits

Instances

Instances (1) Info

Filter instances

Instance state: running Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	-	i-0fd62c91313c2e733	Running	t2.micro	Initializing	No alarms	ap-south-1b

Now creating EBS volume : **aws ec2 create-volume — availability-zone [AZ of instance] — volume-type [Type] — size [value] -tag-specifications**

```

C:\Users\tanmo>aws ec2 create-volume --availability-zone ap-south-1b --volume-type gp2 --size 5 --tag-specifications
{
  "AvailabilityZone": "ap-south-1b",
  "CreateTime": "2021-03-15T07:51:50+00:00",
  "Encrypted": false,
  "Size": 5,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0978d99fc18f9b8f8",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}

```

Create VolumeActions

Filter by tags and attributes or search by keyword

1 to 10 of 10

<input type="checkbox"/>	Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status	Attachment Information
<input type="checkbox"/>		vol-0978d99f...	5 GiB	gp2	100	-		March 15, 2021 at ...	ap-south-1b	available	None	
<input type="checkbox"/>		vol-0eb12b1...	10 GiB	gp2	100	-	snap-0aa33e4...	March 15, 2021 at ...	ap-south-1b	in-use	None	i-0fd62c91313c2e73...
<input type="checkbox"/>		vol-0eeba98...	10 GiB	gp2	100	-		March 13, 2021 at ...	ap-south-1b	in-use	None	i-01e5a4323b71a0dd...
<input type="checkbox"/>		vol-090dd73...	10 GiB	gp2	100	-	snap-0aa33e4...	March 12, 2021 at ...	ap-south-1b	in-use	None	i-09c0229cc067e78c5...
<input type="checkbox"/>		vol-0a3790c...	10 GiB	gp2	100	-	snap-0aa33e4...	March 12, 2021 at ...	ap-south-1b	in-use	None	i-01e5a4323b71a0dd...
<input type="checkbox"/>		vol-09e0ad6...	10 GiB	gp2	100	-	snap-0aa33e4...	January 28, 2021 at ...	ap-south-1a	in-use	None	i-001804790d8a0c4f0...

Now we need to attach the EBS volume to the running instance : **aws ec2 attach-volume — instance-id [Instance] — volume-id [Volume] — device /attaching directory**

```
C:\Users\tanmo>aws ec2 attach-volume --instance-id i-0fd62c91313c2e733 --volume-id vol-0978d99fc18f9b8f8 --device /dev/sdh
{
  "AttachTime": "2021-03-15T07:55:24.261000+00:00",
  "Device": "/dev/sdh",
  "InstanceId": "i-0fd62c91313c2e733",
  "State": "attaching",
  "VolumeId": "vol-0978d99fc18f9b8f8"
}
```

<input type="checkbox"/>	Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot
<input checked="" type="checkbox"/>		vol-0978d99...	5 GiB	gp2	100	-	

To confirm the attaching of the instance : **aws ec2 describe-volumes**

```
C:\Users\tanmo>aws ec2 describe-volumes

{
  "Volumes": [
    {
      "Attachments": [
        {
          "AttachTime": "2021-01-28T09:42:37+00:00",
          "Device": "/dev/sda1",
          "InstanceId": "i-01c50cde1ba8d0c19",
          "State": "attached",
          "VolumeId": "vol-0d455b2ee08e270c9",
          "DeleteOnTermination": true
        }
      ],
      "AvailabilityZone": "ap-south-1b",
      "CreateTime": "2021-01-28T09:42:37.856000+00:00",
      "Encrypted": false,
      "Size": 10,
      "SnapshotId": "snap-0aa33e40a60b0122d",
      "State": "in-use",
      "VolumeId": "vol-0d455b2ee08e270c9",
      "Iops": 100,
      "VolumeType": "gp2",
      "MultiAttachEnabled": false
    },
    {
      "Attachments": [
        {
          "AttachTime": "2021-01-28T09:50:34+00:00",
          "Device": "/dev/sda1",
          "InstanceId": "i-06a5a429ac1941862",
          "State": "attached",
          "VolumeId": "vol-037ff041c5a5b0db9",
          "DeleteOnTermination": true
        }
      ],
      "AvailabilityZone": "ap-south-1b",
      "CreateTime": "2021-01-28T09:50:34.850000+00:00",
      "Encrypted": false,

```

Now we need to create an S3 bucket : **aws s3 mb s3://[bucket-name] — region [region]**

```
C:\Users\tanmo>aws s3 mb s3://mybucket-awstask --region ap-south-1
make_bucket: mybucket-awstask
```

Then we upload a file that we need there



## mybucket-awstask


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

### Objects (1)

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](#)

[Refresh](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	 bird-banner3-L.png	png	March 15, 2021, 13:37:22 (UTC+05:30)	692.2 KB	Standard

We need to make the file in the S3 bucket public to make it accessible to everyone.

#### ☐ Block ~~all~~ public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

##### ☐ Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

##### ☐ Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

##### ☐ Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

##### ☐ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Cancel

Save changes

Now we created cloudfront web distribution for faster content delivery across the world from any of the location using origin as s3

```

C:\Users\tanmo>aws cloudfront create-distribution --origin-domain-name mybucket-awstask.s3.amazonaws.com
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/E18S2654N4OR3",
  "ETag": "E2DUU67VHMHLC",
  "Distribution": {
    "Id": "E18S2654N4OR3",
    "ARN": "arn:aws:cloudfront::376961321203:distribution/E18S2654N4OR3",
    "Status": "InProgress",
    "LastModifiedTime": "2021-03-15T08:10:11.674000+00:00",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1u8l160theczz.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "DistributionConfig": {
      "CallerReference": "cli-1615795810-385647",
      "Aliases": {
        "Quantity": 0
      },
      "DefaultRootObject": "",
      "Origins": {
        "Quantity": 1,
        "Items": [
          {
            "Id": "mybucket-awstask.s3.amazonaws.com-1615795810-101415",
            "DomainName": "mybucket-awstask.s3.amazonaws.com",
            "OriginPath": "",
            "CustomHeaders": {
              "Quantity": 0
            },
            "S3OriginConfig": {
              "OriginAccessIdentity": ""
            },
            "ConnectionAttempts": 3,
            "ConnectionTimeout": 10
          }
        ]
      },
      "OriginGroups": {
        "Quantity": 0
      },
      "DefaultCacheBehavior": {

```

Now we connect to the Instance and check the partitions using **fdisk -l**

```
[root@ip-172-31-14-224 ec2-user]# fdisk -l
Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: A3A5FBF3-1027-4AF7-96EE-E46C819E3D1F

Device            Start      End  Sectors  Size Type
/dev/xvda1        4096 16777182 16773087   8G Linux filesystem
/dev/xvda128      2048    4095    2048    1M BIOS boot

Partition table entries are not in disk order.

Disk /dev/xvdh: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@ip-172-31-14-224 ec2-user]#
```

We create a partition to use the attached EBS volume : **fdisk** /**directory**

```
[root@ip-172-31-14-224 ec2-user]# fdisk /dev/xvdh

Welcome to fdisk (util-linux 2.30.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x0b535bb4.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-10485759, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-10485759, default 10485759):

Created a new partition 1 of type 'Linux' and of size 5 GiB.

Command (m for help):
```

Now we need to format the partition pre usage : **mkfs.ext4** /**directory**

```
[root@ip-172-31-14-224 ec2-user]# mkfs.ext4 /dev/xvdb
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310720 blocks
65536 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Before Mounting we need to install the webserver : **yum install httpd -y**

```
[root@ip-172-31-14-224 etc]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.46-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.46-1.amzn2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.46-1.amzn2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.6.3-5.amzn2.0.2 will be installed
---> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.46-1.amzn2 will be installed
---> Package httpd-tools.x86_64 0:2.4.46-1.amzn2 will be installed
---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
---> Package mod_http2.x86_64 0:1.15.14-2.amzn2 will be installed
--> Running transaction check
---> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

now changing the directory to the main webserver directory : **cd /var/www/html** and mount the partition there : **mount /partitiondirectory /mounting directory**

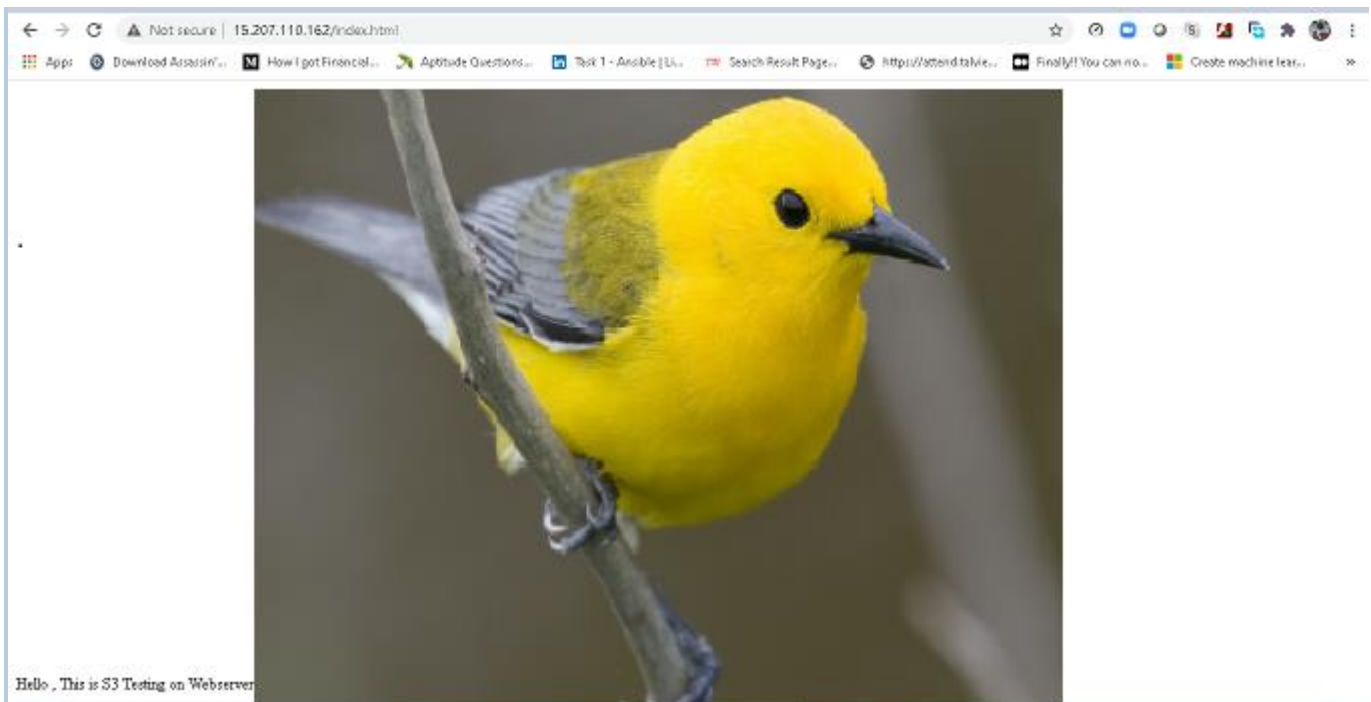
```
[root@ip-172-31-14-224 etc]# cd /var/www/html
[root@ip-172-31-14-224 html]# mkdir /web
[root@ip-172-31-14-224 html]# mount /dev/xvdd /var/www/html/web/
mount: /var/www/html/web/: mount point does not exist.
[root@ip-172-31-14-224 html]# mount /dev/xvdd /var/www/html
[root@ip-172-31-14-224 html]#
```

## Making a file index.html and entering the code there

[illegible]

## Starting the webserver : **systemctl start httpd**

And the going to the browser and entering in the address bar :  
**[public ip of the instance/filename]**



Thus the Page runs successfully and 4 different services of AWS are used: **EC2:** for Running OS

**EBS:** for Attaching Extra storage

**S3:** for storing the file

**Cloudfront :** For providing high availability