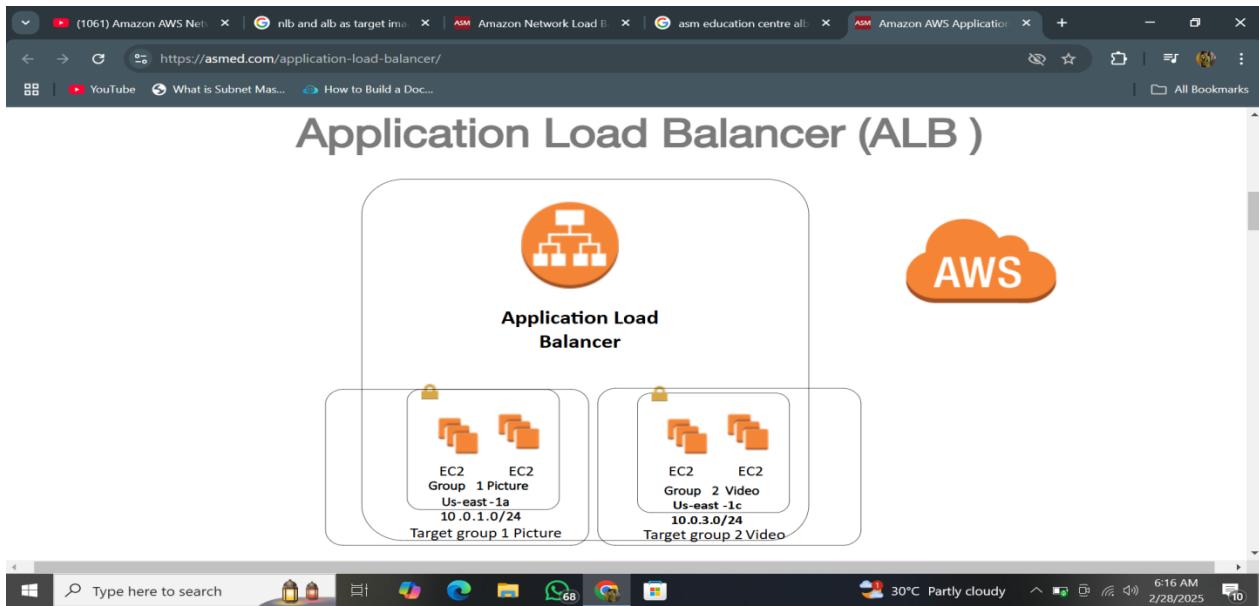


## ALB = PATH -BASE- ROTING PROJECT



## Lab Practice

Before we do the lab, we need to setup the infrastructure with these 4 instances as follows:

You need 4 Virtual Linux Machines

PC1 should be called LinuxPic1: us-east-1a

PC2 should be called LinuxPic2 : us-east-1a

PC3 should be called LinuxVideo3: us-east-1c

PC4 should be called LinuxVideo4: us-east-1c

**1)** Linux PC that is called LinuxPic1 , created two files :

**a)** index.html —> content will be “This is Main site of Picture1 “

**b)** picture.html —> content will be ” This is pic1 ” or

**1a)** Linux PC that is called LinuxPic2 , created two files :

**a)** index.html —> content will be “This is Main site of Picture2”

**b)** picture.html —> content will be ” This is pic2 “

---

**2) Linux PC that is called Linuxvideo3 , created two files :**

**a)** index.html —> content will be “This is Main site of Video3”

**b)** video.html —> content will be ” this is Video3”

**2a) Linux PC that is called Linuxvideo4 , created two files :**

**a)** index.html —> content will be “This is Main site of Video4”

**b)** video.html —> content will be ” this is Video4 “

so when I type x.x.x.x/index.html I will see content with main websites ( for Video)

when you type x.x.x.x/video.html I will see content with Video 3 or 4

so when I type x.x.x.x/index.html I will see content with main websites ( for Picture)

when I type x.x.x.x/picture.html I will see content with picture 1 or 2

---

This are my 4 Ec2 instance

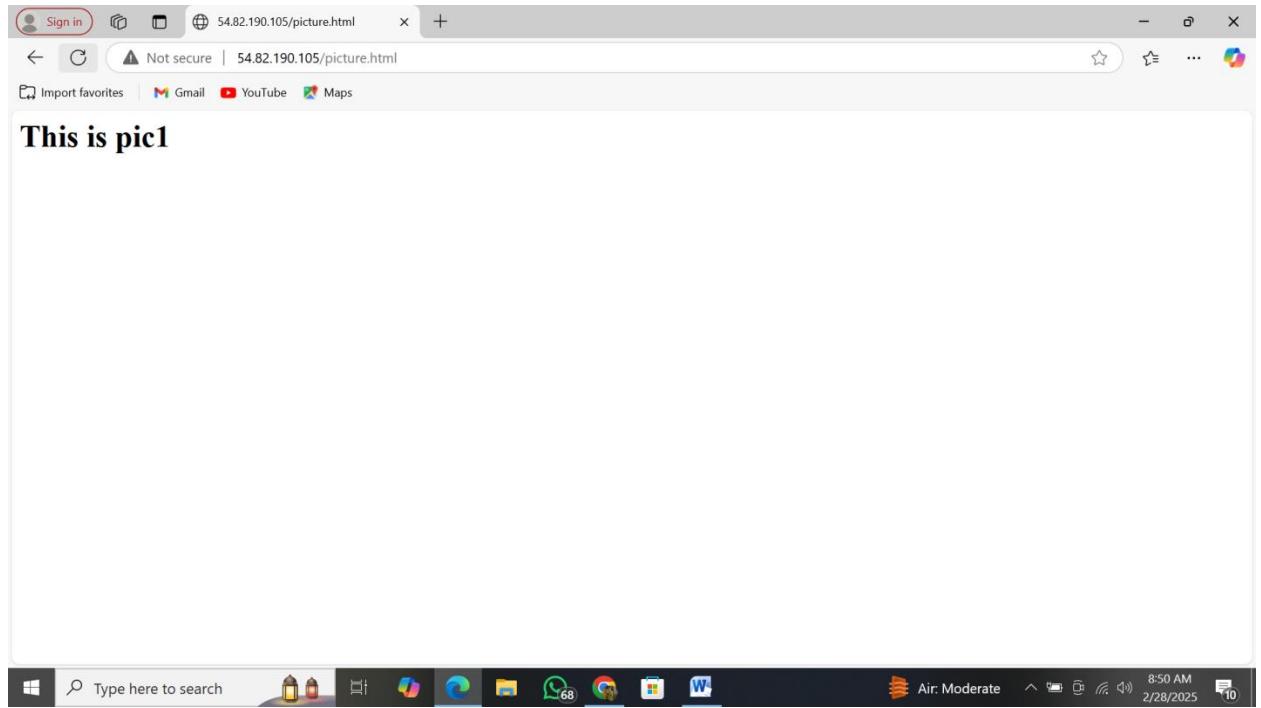
The screenshot shows the AWS Management Console with the EC2 service selected. The main pane displays a list of four EC2 instances, all of which are currently running. The instances are named LinuxVideo3, LinuxPic1, LinuxPic2, and LinuxVideo4. Each instance is associated with an instance ID, its state (Running), instance type (t2.micro), and a status check indicating 2/2 checks passed. The interface includes a search bar, sorting options, and buttons for connecting, launching new instances, and managing alarms.

Name	Instance ID	Instance state	Instance type	Status check
LinuxVideo3	i-046757a9388766f10	Running	t2.micro	2/2 checks passed
LinuxPic1	i-0b7c9f0d125833e8a	Running	t2.micro	2/2 checks passed
LinuxPic2	i-0d4e04d822ee081d2	Running	t2.micro	2/2 checks passed
LinuxVideo4:	i-0f9ce80d8880fc6	Running	t2.micro	2/2 checks passed

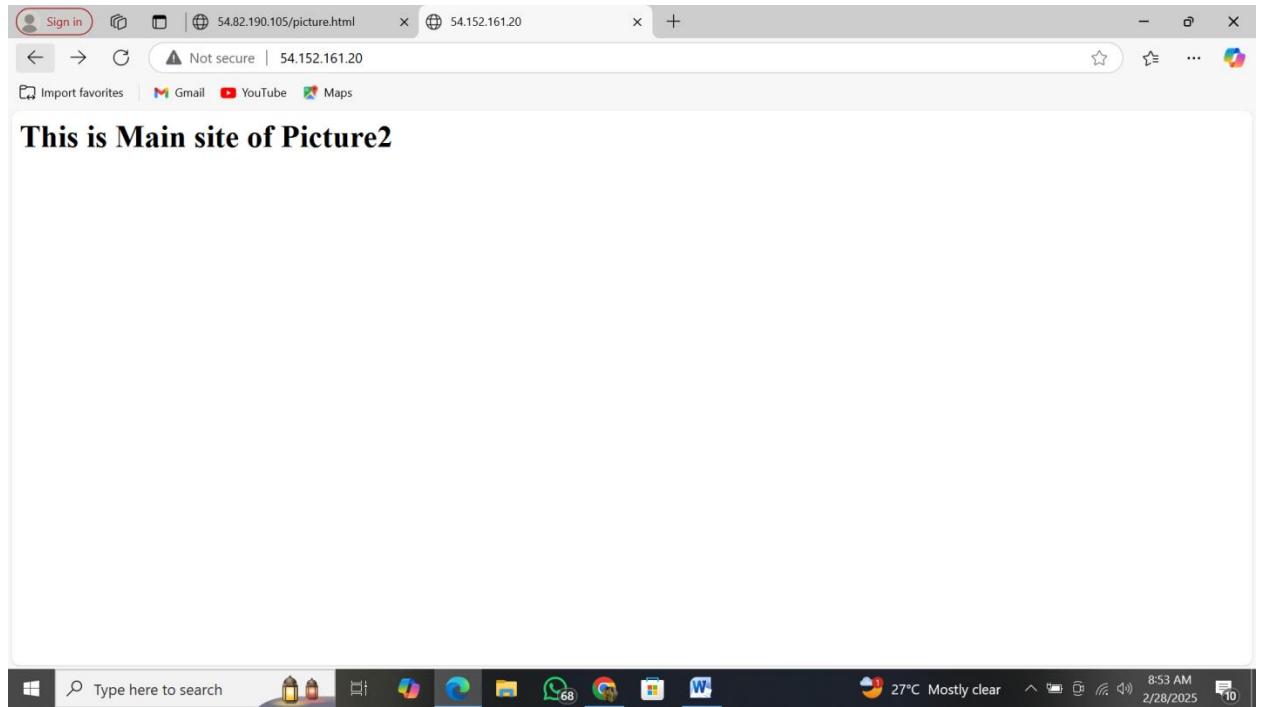
This are my 4 ec2 instance output

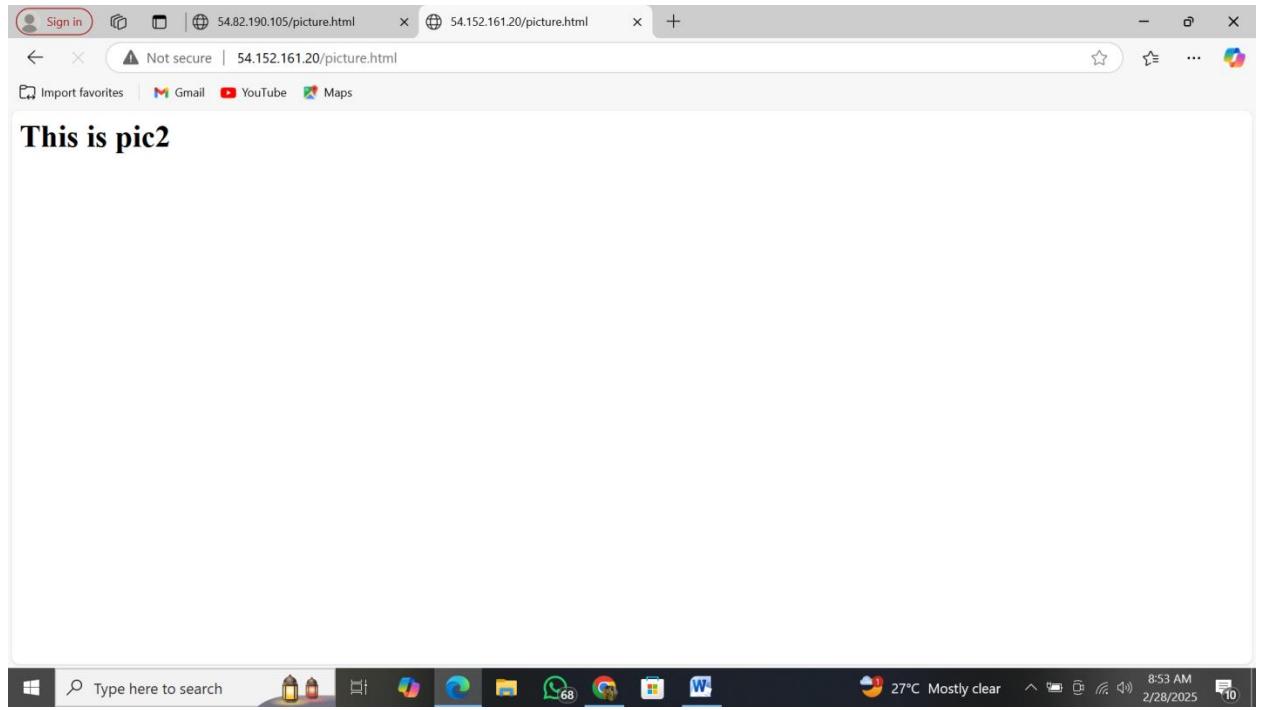
## LinuxPic1

The screenshot shows a Microsoft Edge browser window displaying a blank white page. The address bar indicates the URL is 54.82.190.105, which is likely the public IP address of one of the EC2 instances. The browser interface includes standard navigation buttons, a search bar, and a taskbar at the bottom showing other open applications like File Explorer, Task View, and Microsoft Word.

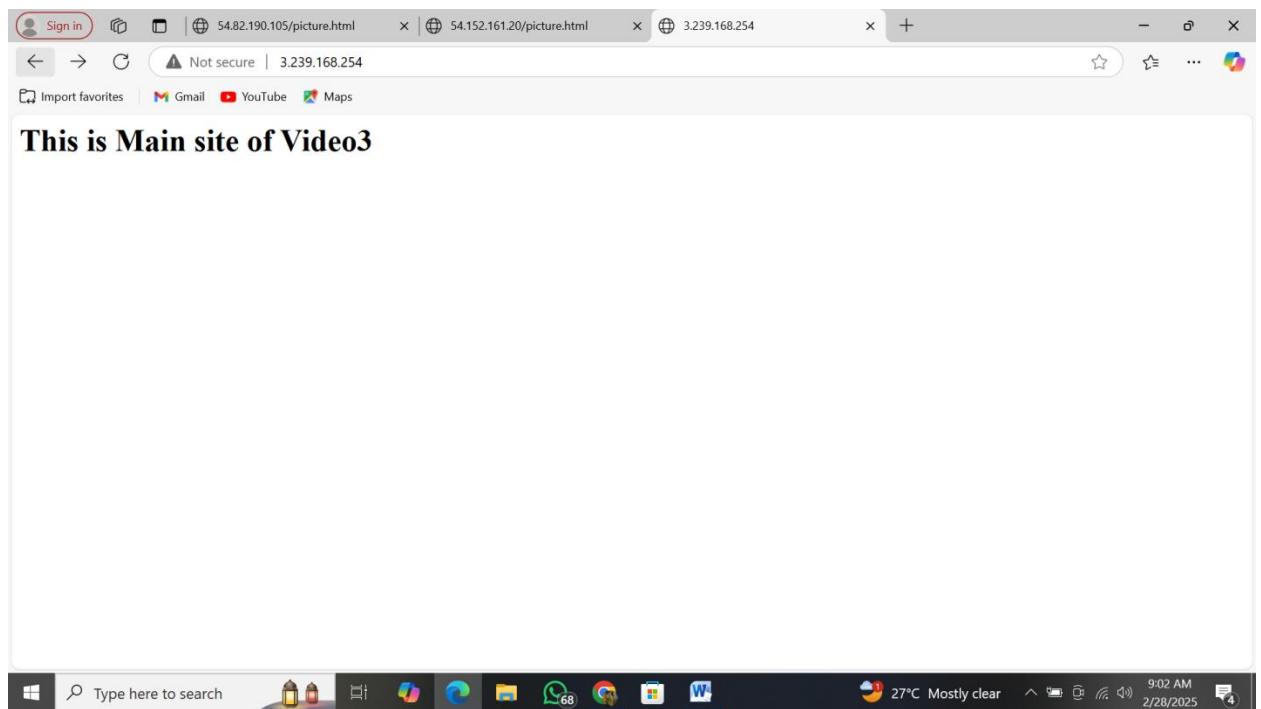


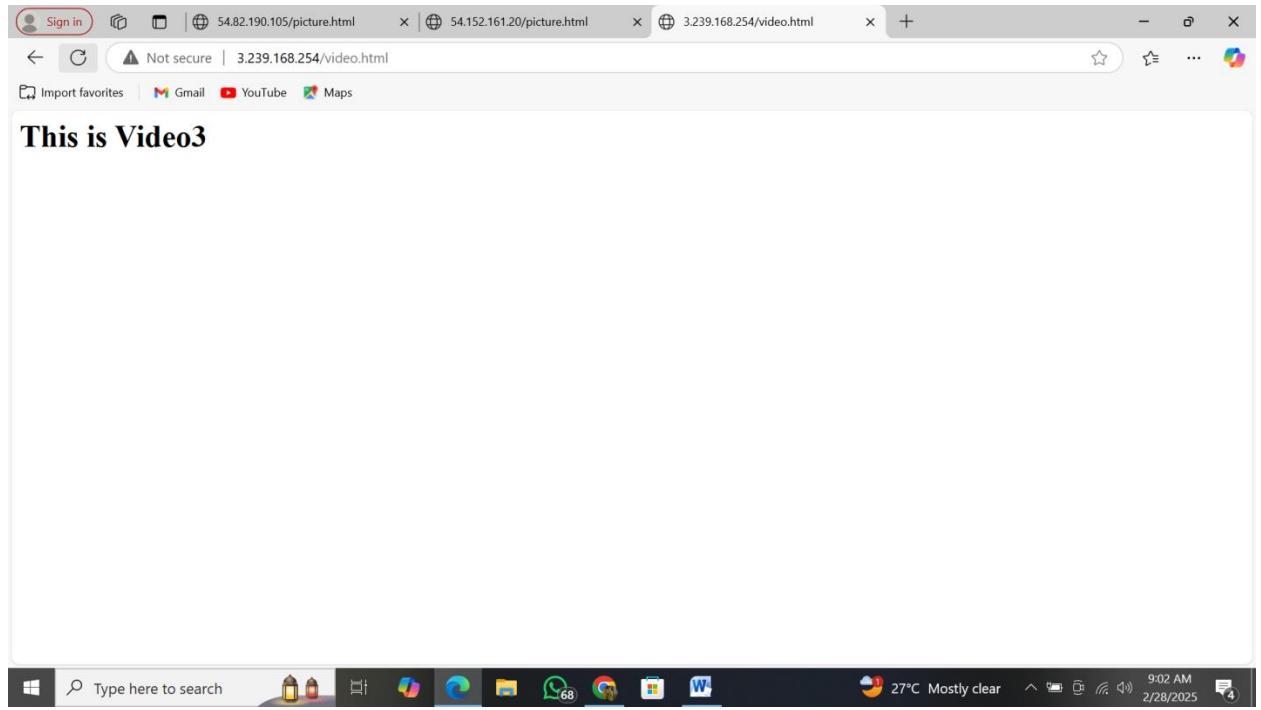
## LinuxPic2



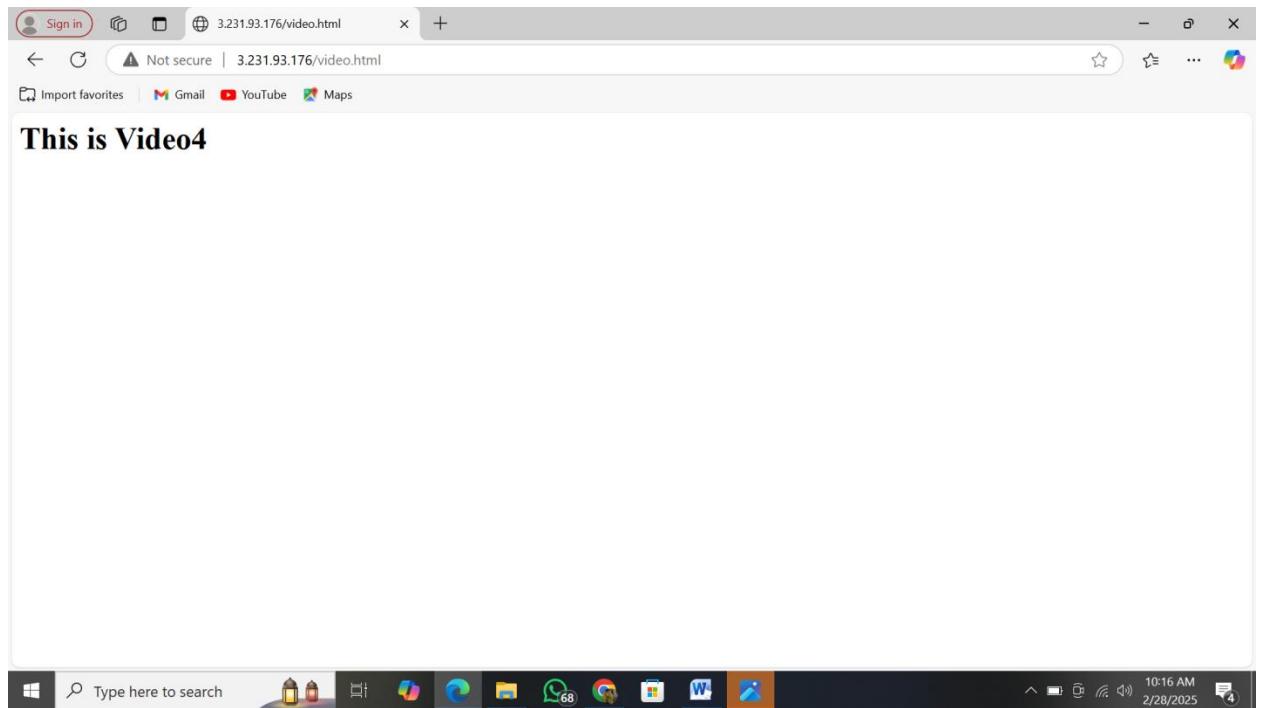


## Linuxvideo3





## LinuxVideo4



Make sure check and see all the 4 Linux Machine is working correctly with Browser

Now we will do the Lab

**Step 1)** before you do anything you need to create two target group

- a. PictureTG
- b. VideoTG

**Step 2)** Choose create a new target group

Target group name=PictureTG

Pick = Instance

Protocol = http

Choose VA VPC

The rest pick default value

Health check = http

Path=/index.html ( We need to make sure your instance has this file name )

**Step 3)** When you get to register targets; make sure you pick LinuxPic1 and LinxPic2 Instance ;

**Step 4)** click include as pending below

**Step 5)** Then create target group; after it finish ; click on it and you will see it is “unused” ; later on it will be changed .

**Step 6)** Do same steps in above but called it

Target group name=VideoTG

Pick = Instance

Protocol=http

Choose VA VPC

The rest pick default value

Health check=http

Path=/ind(We need to make sure your instance has this file name )

Screenshot of the AWS Lambda console showing the creation of a new function named "Step 1 Create". The function is configured with the "AWS Lambda" runtime and the "HelloWorld" template. The "Handler" is set to "index.handler". The "Role" is "Lambda execution role". The "Memory Size" is 128 MB and the "Timeout" is 3 seconds. The "Environment Variables" section contains the variable "NAME" with the value "Step 1 Create". The "Code" tab shows the Lambda function code in Python.

Step 1  
Specify group details

Step 2  
Register targets

## Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

### Basic configuration

Settings in this section can't be changed after the target group is created.

#### Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

CloudShell Feedback Type here to search © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Partly cloudy 10:22 AM 2/28/2025

Screenshot of the AWS Lambda console showing the creation of a new function named "Step 1 Create". The function is configured with the "AWS Lambda" runtime and the "HelloWorld" template. The "Handler" is set to "index.handler". The "Role" is "Lambda execution role". The "Memory Size" is 128 MB and the "Timeout" is 3 seconds. The "Environment Variables" section contains the variable "NAME" with the value "Step 1 Create". The "Code" tab shows the Lambda function code in Python.

PictureTG

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

#### Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP 80 1-65535

#### IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

#### VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

my-vpc-01

CloudShell Feedback Type here to search © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Partly cloudy 10:23 AM 2/28/2025



The screenshot shows the AWS CloudShell interface. At the top, there are several browser tabs open, including "Step 1 Create" which is currently active. Below the tabs, the URL is https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTargetGroup;. The main content area displays the "Create target group" wizard. Step 1 is titled "Specify group details". It includes fields for "Health check protocol" (set to HTTP), "Health check path" (set to /index.html), and "Attributes" (with a note: "Certain default attributes will be applied to your target group. You can view and edit them after creating the target group."). The status bar at the bottom indicates "CloudShell Feedback" and shows system icons like battery level and signal strength.



The screenshot shows the AWS CloudShell interface. At the top, there are several browser tabs open, including "Step 2 Create" which is currently active. Below the tabs, the URL is https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTargetGroup;. The main content area displays the "Create target group" wizard. Step 2 is titled "Register targets". It shows a step-by-step process: "Step 1 Specify group details" (done) and "Step 2 Register targets" (in progress). A note states: "This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets." Below this, a table titled "Available instances (2/4)" lists two instances: "LinuxVideo4:" and "LinuxVideo3", both running and associated with "web-sg". The status bar at the bottom indicates "CloudShell Feedback" and shows system icons like battery level and signal strength.

The screenshot shows the AWS Lambda console interface. A new function named "Step 2 Create" is being created. The "Code" tab is active, displaying the following code:

```
function handler(event) {
  return {
    statusCode: 200,
    body: JSON.stringify({
      message: 'Hello from Lambda'
    })
  }
}
```

The "Test" tab shows a successful execution with the output "Hello from Lambda".

click include as pending below

The screenshot shows the AWS Lambda console interface. A new function named "Step 2 Create" is being created. The "Code" tab is active, displaying the following code:

```
function handler(event) {
  return {
    statusCode: 200,
    body: JSON.stringify({
      message: 'Hello from Lambda'
    })
  }
}
```

The "Test" tab shows a successful execution with the output "Hello from Lambda".

Click create target group

Do same steps in above but called it

Target group name=VideoTG

Pick = Instance

Protocol=http

Chose VA VPC

The rest pick default value

Health check=http

Path=/index.html (We need to make sure your instance has this file name )

The screenshot shows the AWS CloudShell interface with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#TargetGroups>. The left sidebar is collapsed. The main content area displays the 'Target groups' section with two entries:

Name	ARN	Port	Protocol	Target type
VideoTG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/VideoTG/1234567890123456	80	HTTP	Instance
PictureTG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/PictureTG/1234567890123456	80	HTTP	Instance

Below the table, a message says "0 target groups selected". The status bar at the bottom right shows "27°C Partly cloudy" and the date "2/28/2025".

Alright we have done it

The screenshot shows the AWS EC2 Target groups page. On the left, there's a navigation sidebar with sections for Volumes, Snapshots, Lifecycle Manager, Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The 'Target Groups' section is currently selected and highlighted in blue. The main content area is titled 'Target groups (2) Info'. It features a search bar labeled 'Filter target groups' and a table with the following data:

Name	Protocol	Target type	Load balancer	VPC ID
VideoTG	HTTP	Instance	None associated	vpc-009aef3f1a7cc3f81
PictureTG	HTTP	Instance	None associated	vpc-009aef3f1a7cc3f81

Below the table, a message says '0 target groups selected' and 'Select a target group above.' At the bottom of the page, there's a footer with links for CloudShell, Feedback, and various system icons like battery level, signal strength, and date/time (10:34 AM, 2/28/2025).

**Step 6)** As we see when we are at target page; we do not see any load Balancer in front of it ; since we have not configured it; but later on we will see it

**Step 7)** Now we will go and create Application Load Balancer

**Step 8)** Before you creatin ALB ; make sure you will know what is your security group for you Instance and have it ready

**Step 9)** Create ALB ;

Name =VAALB

Pick your VPC

Pick both AZ=1a and AZ=1c

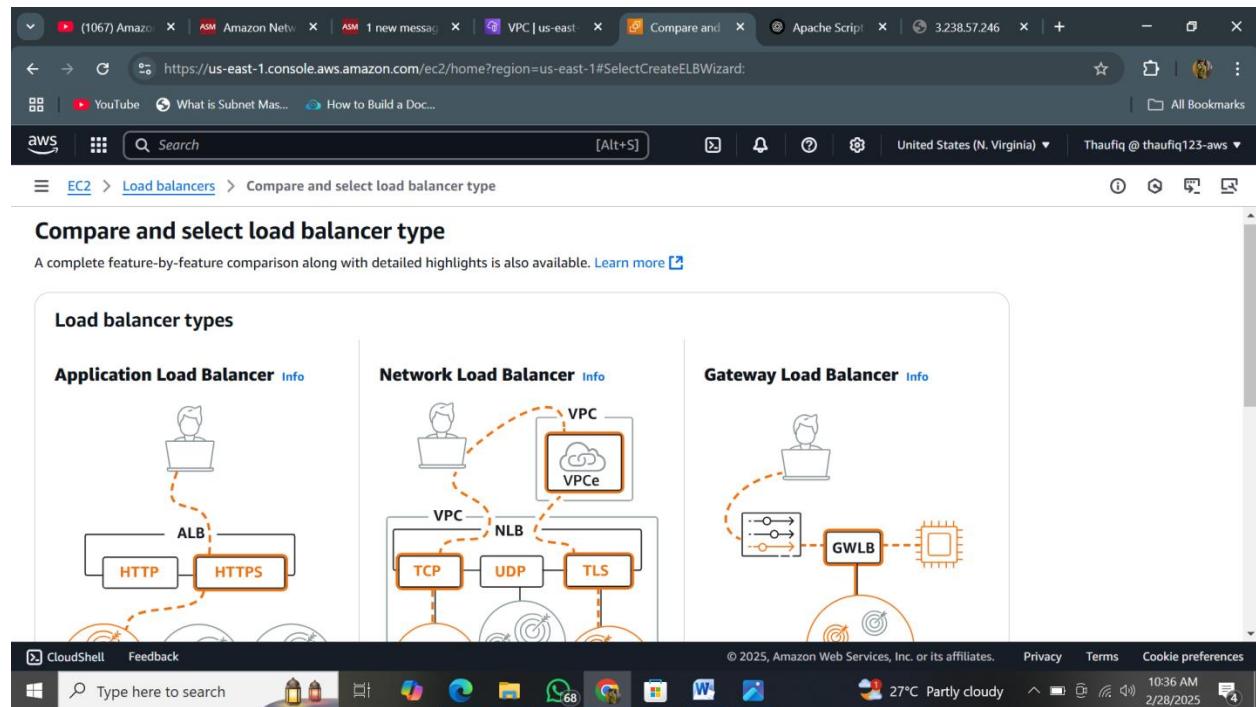
Pick Same Security group as above

Listener = 80

**Step 9)** In front of Listener; you need to pick your Target Group ; in here does not matter which one you will pick ; but I will pick PictureTG

**Step 10)** Create ALB ; then back to your target group ; click on “PictureTG” you will see statues as initial ; then it will become “Healthy”

**Step 11)** Now after two mins it become Healthy and when you look at Target Group page we see in column under Load Balancer; it shows name of VAALB with PictureTG ( since we chose it as default) but VideoTG has nothing in it.



The screenshot shows the AWS Management Console with multiple tabs open at the top, including 'Compare and select load balancer type'. The main content area displays three cards for different types of load balancers:

- Application Load Balancer (ALB):** Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.
- Network Load Balancer (NLB):** Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.
- Gateway Load Balancer (GLB):** Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Each card includes a 'Create' button at the bottom.

https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateALBWizard:

YouTube What is Subnet Mas... How to Build a Doc...

aws Search [Alt+S] United States (N. Virginia) Thauqiq @ thauqiq123-aws

EC2 > Load balancers > Create Application Load Balancer

## Basic configuration

**Load balancer name**  
Name must be unique within your AWS account and can't be changed after the load balancer is created.

VAAALB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** | **Info**  
Scheme can't be changed after the load balancer is created.

**Internet-facing**

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name is publicly resolvable.
- Requires a public subnet.

**Internal**

- Serves internal traffic.
- Has private IP addresses.
- DNS name is publicly resolvable.
- Compatible with the IPv4 and Dualstack IP address types.

**Load balancer IP address type** | **Info**  
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

**IPv4**  
Includes only IPv4 addresses.

**Dualstack**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 10:37 AM 27°C Partly cloudy 2/28/2025

The screenshot shows the AWS CloudShell interface with several tabs open at the top: (1067) Amazon, Amazon Network, 1 new message, VPC | us-east, Create application, Apache Script, and 3.238.57.246. The main content area displays the 'Create Application Load Balancer' wizard. The 'Subnets' step is shown, where the user has selected two subnets from a dropdown menu. The selected subnets are highlighted with a blue border: 'my-vpc-01' (IPv4 CIDR: 10.0.0.0/16) and 'my-vpc-01' (IPv4 CIDR: 10.0.0.0/16). Below the subnets, the 'Availability Zones and subnets' section is visible, showing the checked boxes for 'us-east-1a (use1-az4)' and 'us-east-1a (use1-az4)'. The bottom of the screen shows the standard AWS navigation bar with CloudShell, Feedback, and various icons.

This screenshot continues the AWS CloudShell session. The 'Availability Zones and subnets' step is still active, showing the checked box for 'us-east-1a (use1-az4)'. A new subnet, 'subnet-0d2dafd6bdbba3a04d' (IPv4 subnet CIDR: 10.0.1.0/24), has been added to the list and is highlighted with a blue border. The dropdown menu for this subnet is set to 'pub-1a'. Below this, the 'Subnet' section is visible, showing the checked box for 'us-east-1c (use1-az1)'. A note indicates that only CIDR blocks corresponding to the load balancer IP address type are used, and at least 8 available IP addresses are required. A second subnet, 'subnet-0fb215aa40ace432' (IPv4 subnet CIDR: 10.0.3.0/24), is listed and highlighted with a blue border. The dropdown menu for this subnet is set to 'pub-1c'. The bottom of the screen shows the standard AWS navigation bar with CloudShell, Feedback, and various icons.

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#CreateFunctionWizard>. The page is titled "Create Function". A dropdown menu is open, showing options like "Lambda function", "AWS Lambda function", and "AWS Lambda function with VPC". Below the dropdown, there's a search bar and a "Create function" button.

The screenshot shows the AWS Lambda console with the URL <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#CreateFunctionWizard>. The page is titled "Create Function". A dropdown menu is open, showing options like "Lambda function", "AWS Lambda function", and "AWS Lambda function with VPC". Below the dropdown, there's a search bar and a "Create function" button.

The screenshot shows the AWS EC2 Load Balancers console. On the left, a navigation sidebar lists Network & Security, Load Balancing (Load Balancers selected), and Auto Scaling. The main content area displays a table titled "Load balancers (1)". The table has columns for Name, DNS name, State, VPC ID, and Availability Zones. One row is shown, labeled "VAALB", with the DNS name "VAALB-1832223006.us-east...". The "State" column shows "Active", and the "VPC ID" column shows "vpc-009aef3f1a7cc3f81". The "Availability Zones" column indicates "2 Availability Zones". Below the table, a message says "0 load balancers selected" and "Select a load balancer above." The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating "27°C Partly cloudy" and the date "2/28/2025".

The screenshot shows the AWS EC2 Target Groups console. On the left, a navigation sidebar lists Network & Security, Load Balancing (Target Groups selected), and Auto Scaling. The main content area displays a table titled "Target groups (1/2)". The table has columns for Name, Protocol, Target type, Load balancer, and VPC ID. Two rows are listed: "PictureTG" (Protocol: HTTP, Target type: Instance, Load balancer: VAALB, VPC ID: vpc-009aef3f1a7cc3f81) and "VideoTG" (Protocol: HTTP, Target type: Instance, Load balancer: None associated, VPC ID: vpc-009aef3f1a7cc3f81). Below the table, a section titled "Target group: PictureTG" shows tabs for Details, Targets, Monitoring, Health checks, Attributes, and Tags. The "Details" tab is selected, displaying the ARN: arn:aws:elasticloadbalancing:us-east-1:688567302802:targetgroup/PictureTG/23755325c35c2a9d. The bottom of the screen shows a Windows taskbar with various icons and a system tray indicating "27°C Partly cloudy" and the date "2/28/2025".

## Step 12) Now go to VideoTG and click it and you will see “unused”

The screenshot shows the AWS EC2 Target Groups page for the 'VideoTG' target group. The left sidebar navigation includes Network & Security, Load Balancing, Auto Scaling, and Settings. The main content area displays the 'Details' section for the target group, which lists the ARN, Target type (Instance), Protocol (HTTP: 80), Protocol version (HTTP1), and VPC (vpc-009aeef3f1a7cc3f81). Below this, a summary table shows 2 total targets, 0 healthy, 0 unhealthy, 2 unused, 0 initial, and 0 draining. A section titled 'Distribution of targets by Availability Zone (AZ)' is also present. The bottom of the page includes a CloudShell search bar and a system tray with weather information.

The screenshot shows the AWS EC2 Target Groups page with the 'Targets' tab selected. The left sidebar navigation is identical to the previous screenshot. The main content area displays the 'Registered targets' section, which lists two targets: 'LinuxVideo4:' and 'LinuxVideo3'. Both targets are in the 'us-east-1c' zone, port 80, and are marked as 'Unused'. The status for both is 'Target group is not healthy'. The bottom of the page includes a CloudShell search bar and a system tray with weather information.

**Step 13)** Now we want to make sure we use VideoTG

**Step 14)** Go to Load Balancer on left ; go to Listener tab

**Step 15)** Click View/Edit Rule

**Step 16)** Pick path then type /video.html then action Forward to target Group VideoTG

**Step 17)** Make sure click Save and do same steps ; but add rule for /picture.html and FW to pictureTG

**Step 18)** Now Click Back ; go back to VAALB and click on description tab and copy and paste the Log DNS name then try with

/picture.html and

/video.html

You should be able to see all the instance is working

The screenshot shows the AWS CloudShell interface with the following details:

- URL:** https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancer:loadBalancerArn=arn:aws:elasticloadbalancing:us-east-1:68856...
- Region:** United States (N. Virginia)
- User:** Thauqiq @ thaufiq123-aws
- EC2 > Load balancers > VAALB**
- Listeners and rules** tab is selected.
- Listeners and rules (1) info:** A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
- Filter listeners:** HTTP:80
- Protocol:Port:** HTTP:80
- Default action:** Forward to target group
- Target group:** PictureTG (1) (100%)
- Rules:** 1 rule
- ARN:** ARN
- Security policy:** Not applicable

The screenshot shows the AWS CloudWatch Metrics interface. At the top, there are tabs for 'Metrics' (selected), 'Logs', 'CloudWatch Metrics Insights', and 'CloudWatch Metrics Metrics Insights'. Below the tabs, there's a search bar and a 'Create metric filter' button. The main area displays a table of metrics with columns: Metric name, ARN, Last value, Last timestamp, and Last period. One row is highlighted in blue, showing 'AWS/CloudWatch Metrics' with a value of 1000. The bottom of the screen shows the Windows taskbar with various pinned icons.

click on rule

The screenshot shows the AWS CloudWatch Metrics Insights interface. At the top, there are tabs for 'Metrics' (selected), 'Logs', 'CloudWatch Metrics Insights', and 'CloudWatch Metrics Metrics Insights'. Below the tabs, there's a search bar and a 'Create metric filter' button. The main area displays a table of metrics with columns: Metric name, ARN, Last value, Last timestamp, and Last period. One row is highlighted in blue, showing 'AWS/CloudWatch Metrics' with a value of 1000. The bottom of the screen shows the Windows taskbar with various pinned icons.



The screenshot shows the AWS VAAALB Step 2 Add rule conditions page. On the left, a vertical navigation bar lists steps: Step 1 (Add rule), Step 2 (Define rule conditions, currently selected), Step 3 (Define rule actions), Step 4 (Set rule priority), and Step 5 (Review and create). The main content area is titled "Define rule conditions" with an info link. It states: "Requests reaching this rule must match all specified conditions for the rule to apply. At least 1 condition is required." Below this is a section titled "Listener details: HTTP:80" and a "Conditions (0)" section. A "Rule limits" button is located in the top right of the conditions section. The "Conditions (0)" section displays "No conditions" and "No conditions to display." A "Add condition" button is centered in this section. At the bottom right are "Cancel", "Previous", and "Next" buttons.

Click on add condition and choose path

The screenshot shows the same AWS VAAALB Step 2 Add rule conditions page as above, but with a modal window open over the "Conditions (0)" section. The modal title is "Define rule conditions" with an info link. It says: "Include one of each" and "Just match all specified conditions for the rule to apply. At least 1 condition is required." Inside the modal, there is a list of condition types: "Host header", "Path", "HTTP request method", and "Source IP". Below this is a "Choose condition" button. At the bottom right of the modal are "Cancel" and "Confirm" buttons. The background of the main page is dimmed.

The screenshot shows the AWS VPC console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AddListenerRule:listenerArn=arn:aws:elasticloadbalancing:us-east-1:68856730...>. The page title is "Step 2 Add Listener Rule". The left sidebar shows steps: Step 1 (Add rule), Step 2 (Define rule conditions) which is selected, Step 3 (Define rule actions), Step 4 (Set rule priority), Step 5 (Review and create). The main content area is titled "Add condition" and "Rule limits". It says "Rule condition types: Route traffic based on the condition type of each request. Each rule can include one of each of the following conditions: host-header, path, http-request-method and source-ip. Each rule can include one or more of each of the following conditions: http-header and query-string." A dropdown menu shows "Path" is selected. Below it, a text input field contains "/video.html". A note says "Valid characters are a-z, A-Z, 0-9 and special characters. Path must be 1-128 characters." A "Cancel" button and an orange "Confirm" button are at the bottom right. The status bar at the bottom shows "CloudShell Feedback" and system icons.

This screenshot continues from the previous one, showing the "Conditions (1)" section expanded. It displays a single condition: "Path (1)" with the value "/video.html". The "Edit", "Delete", and "Add condition" buttons are visible next to the conditions list. The status bar at the bottom shows "CloudShell Feedback" and system icons.

The screenshot shows the AWS CloudShell interface with the AWS Lambda function configuration page open. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#function:arn:aws:lambda:us-east-1:68856730...>. The Lambda function name is 'VideoTG'. The configuration tab is selected, showing the code editor with the following Python code:

```
def lambda_handler(event, context):  
    # Add your code here to process the event  
    # Return a response.  
    # You can return a dict or a string.  
    # Example:  
    # return {"text": "Hello world"}  
    pass
```

The Lambda function ARN is `arn:aws:lambda:us-east-1:688567301234:function:VideoTG`. The Region is set to US East (N. Virginia). The Handler is `lambda.lambda_handler`. The Role is `lambdaBasicExecutionRole`.

The screenshot shows the AWS CloudShell interface with the AWS Lambda function configuration page open. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#function:arn:aws:lambda:us-east-1:688567301234:function:VideoTG>. The Lambda function name is 'VideoTG'. The Configuration tab is selected, showing the configuration details:

- Function name: VideoTG
- Runtime: Python 3.9
- Memory: 128 MB
- Timeout: 3 seconds
- Environment variables: None
- Code size: 10 KB
- Layers: None
- Tracing: Off
- Logs: View logs
- Metrics: View metrics
- Logs configuration: Log group: /aws/lambda/VideoTG, Log stream: \$LogStream
- Deployment package: zip file
- Deployment: Deploy

Now go to your target group and see

The screenshot shows the AWS CloudShell interface with the AWS Management Console open. The user is navigating through the EC2 service to view Target Groups. The 'Target groups' table shows two entries:

Name	Protocol	Target type	Load balancer	VPC ID
PictureTG	HTTP	Instance	VAALB	vpc-009aef3f1a7cc3f81
VideoTG	HTTP	Instance	VAALB	vpc-009aef3f1a7cc3f81

Below the table, the 'Details' tab of the 'Target group: VideoTG' page is active, showing the ARN: arn:aws:elasticloadbalancing:us-east-1:688567302802:targetgroup/VideoTG/cd74e73daa508de8.

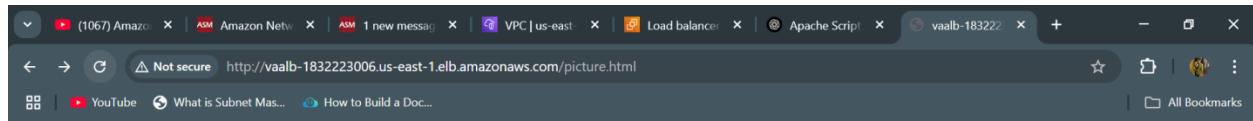
**Step 19)** Now Click Back ; go back to VAALB and click on description tab and copy and paste the Log DNS name then try with

/picture.html and

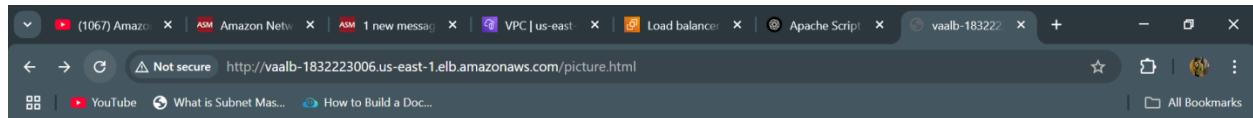
/video.html

VAALB-1832223006.us-east-1.elb.amazonaws.com/picture.html

VAALB-1832223006.us-east-1.elb.amazonaws.com/picture.html

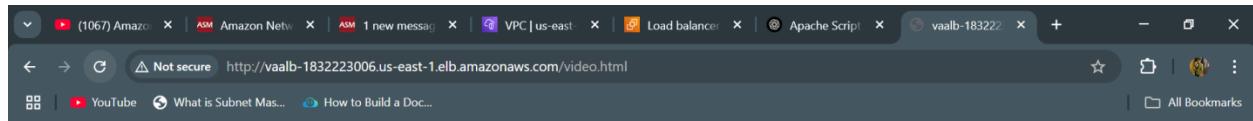


This is pic1

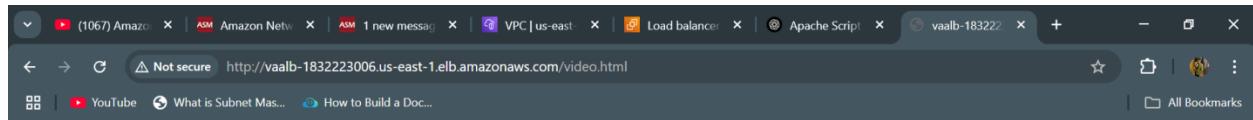


This is pic2





**This is Video3**



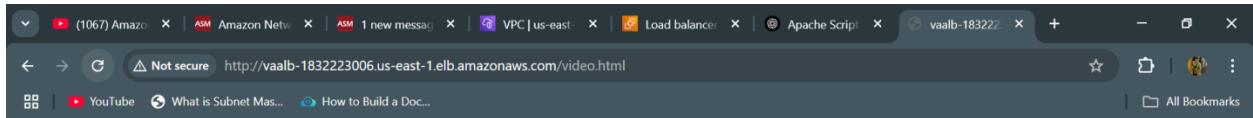
**This is Video4**



Has you one you hit the url [VAALB-1832223006.us-east-1.elb.amazonaws.com/picture.html](http://VAALB-1832223006.us-east-1.elb.amazonaws.com/picture.html)

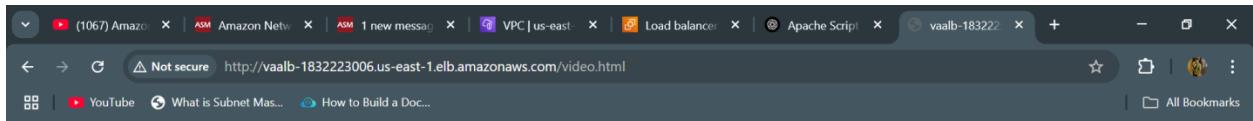
It show this is pic1 and you hit multiple time the traffic shift to pic2

Same we try for /video.html



This is Video3





**Step 20)** Remember you can go back to your Route53 and create an “A” record and point to the above link

**Step 21)** Now If I go back to Target Group page; now under ALB column I will see both PictureTG and VideoTG

**Step 22)** Go and click on VideoTG and now you will see “healthy” Remember ; it as “unused”we had