

# **CSM ASSIGNMENT - 2**

**Name :- Maanav Singh**

**SAP ID :- 500108304**

**Batch :- CCVT (B-3)**

**Enrollment no. :- R2142220977**

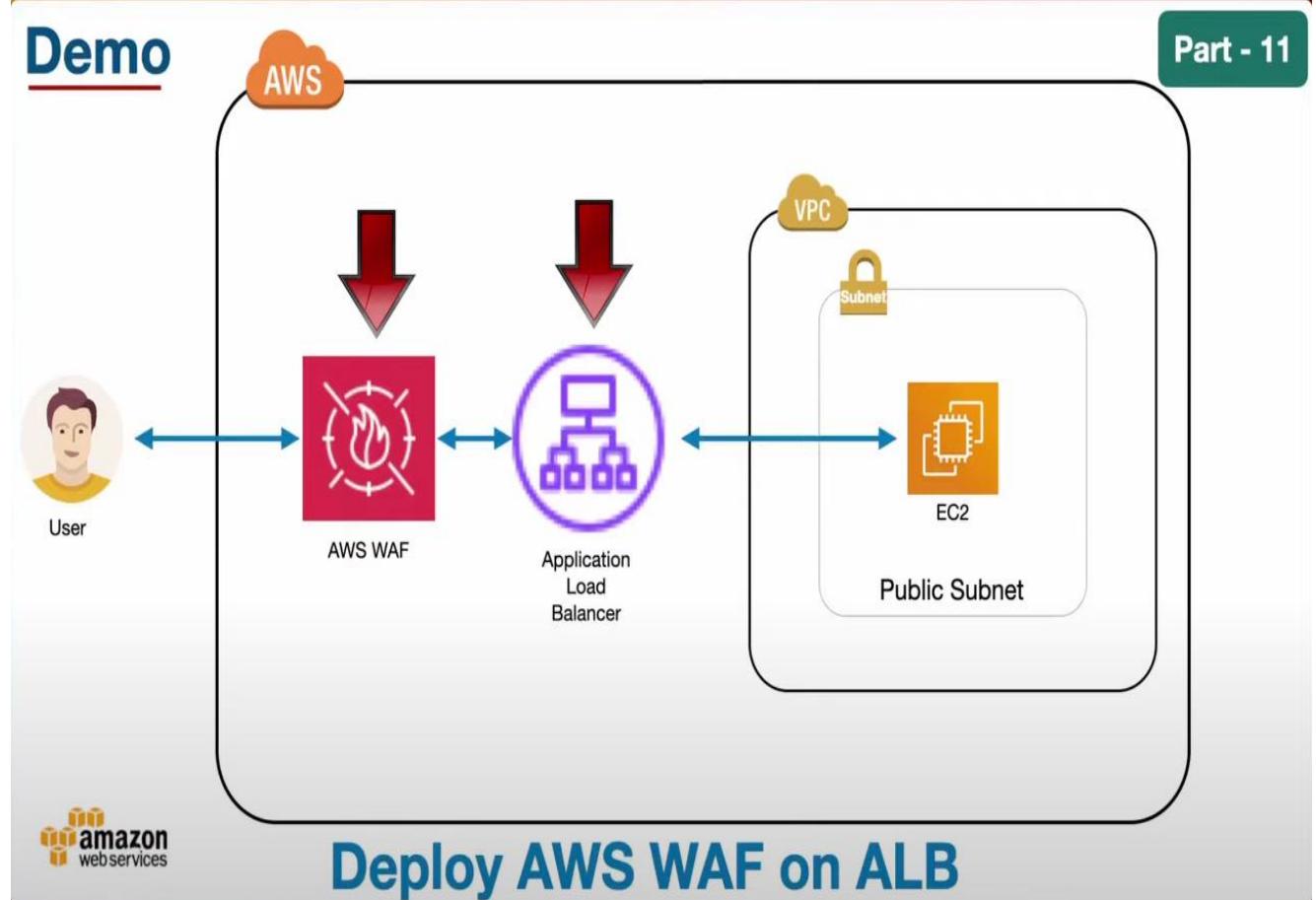
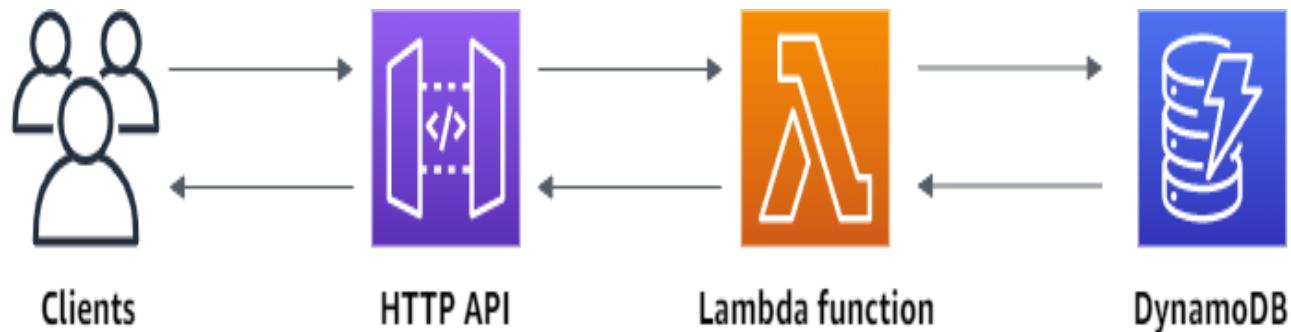
**Faculty :- Avita Katal**

**Q-1 To gain hands-on experience in deploying a three-tier architecture using AWS services: AWS Lambda for the application layer, Amazon RDS or DynamoDB for the data layer, Amazon EC2 for the presentation layer, and**

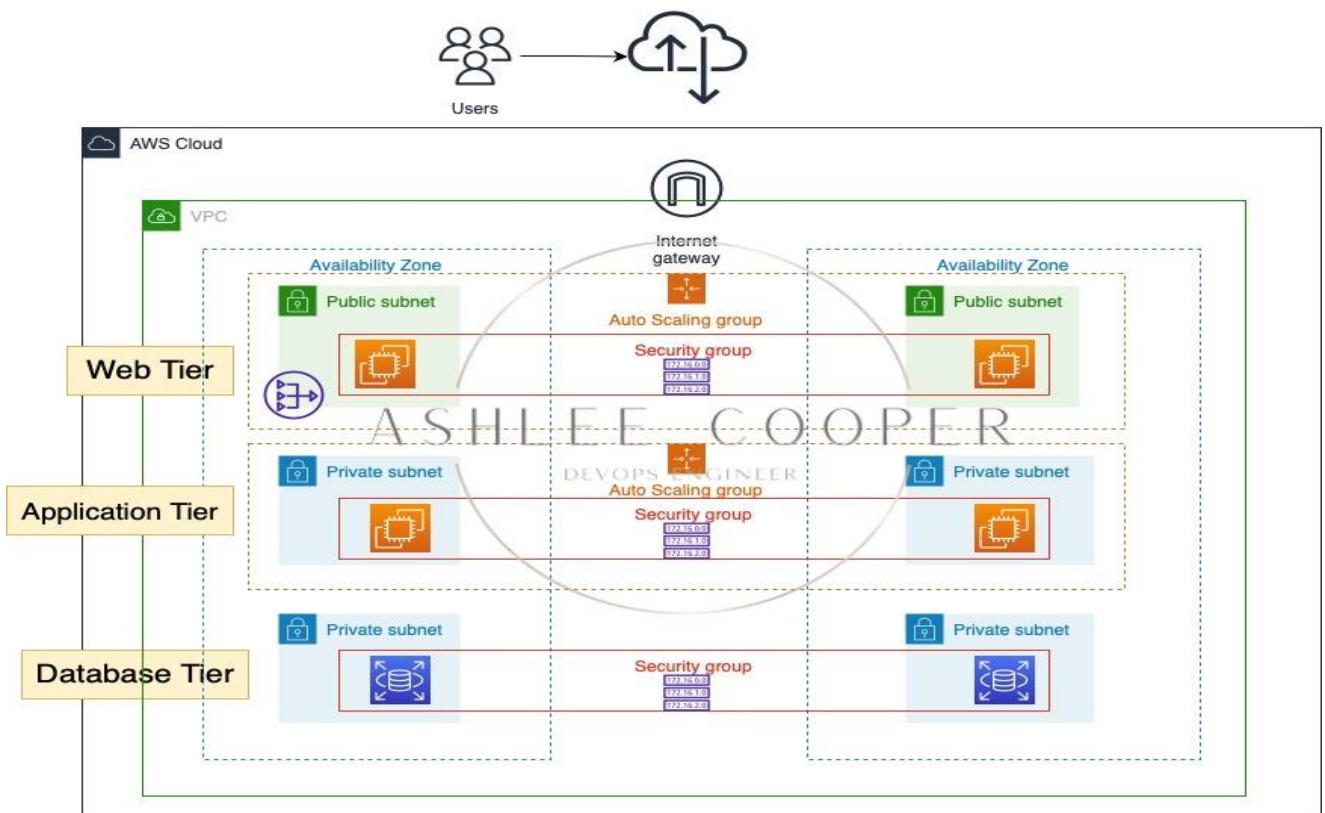
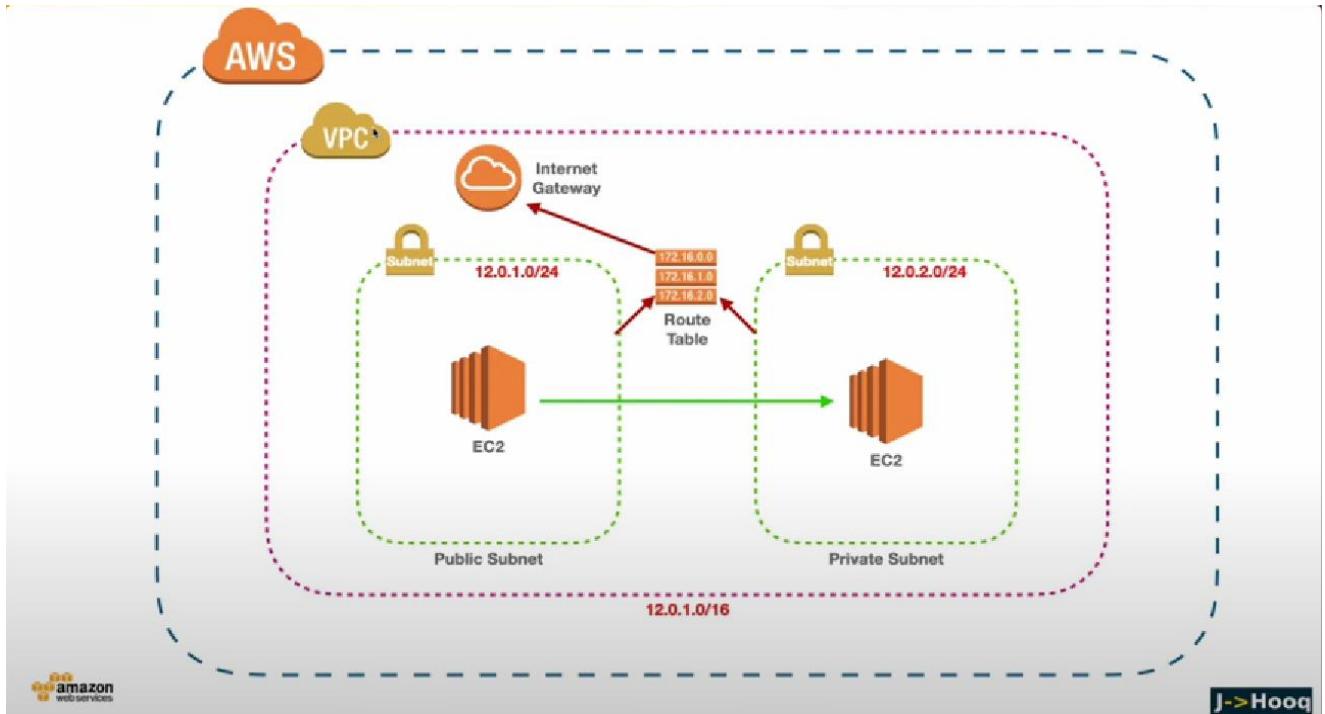
# setting up a Virtual Private Cloud (VPC) for network isolation and security.

## Solution :-

- 1) Architectures Used in this Project







2) Create table in dynamoDB

The screenshot shows the AWS DynamoDB console with the URL [ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#tables](https://ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#tables). The left sidebar has 'Tables' selected under 'DynamoDB'. The main area shows a table titled 'Tables (1) Info' with one row for 'db-signup-table'. The table columns include Name, Status, Partition key, Sort key, Indexes, Deletion protection, Read capacity mode, and Write capacity mode. The table row shows 'db-signup-table' as Active, with 'uid (S)' as the partition key and '0' as the sort key. The 'Read capacity mode' and 'Write capacity mode' are both set to 'Provisioned (1)'. The status bar at the bottom right indicates '© 2024, Amazon Web Services, Inc. or its affiliates.'

The screenshot shows the AWS DynamoDB console with the URL [ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#table?name=db-signup-table](https://ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#table?name=db-signup-table). The left sidebar has 'Tables' selected under 'DynamoDB'. The main area shows the 'db-signup-table' details. The 'General information' section includes fields for Partition key (uid String), Sort key (-), Capacity mode (Provisioned), and Table status (Active). It also shows Alarms (No active alarms) and Point-in-time recovery (PITR) (Off). The 'Items summary' section indicates that DynamoDB updates information approximately every six hours. The status bar at the bottom right indicates '© 2024, Amazon Web Services, Inc. or its affiliates.'

3) Create Roles in IAM and select amazonDynamoDBFullAccess also.

The screenshot shows the AWS IAM Roles page. On the left, the navigation menu includes 'Identity and Access Management (IAM)', 'Access management' (with 'User groups' and 'Users' sub-options), 'Roles' (selected), 'Policies', 'Identity providers', and 'Account settings'. Under 'Access reports', there are 'Access Analyzer', 'External access', 'Unused access', and 'Analyzer settings'. A 'Credential report' option is also present. The main content area displays 'Roles (11) Info' with a search bar and a table showing three roles:

Role name	Trusted entities	Last activity
Powerfunction2-role-3wn5oe6u	AWS Service: lambda	-
Powerfunction2-role-dw9okjcv	AWS Service: lambda	81 days ago
role-signup	AWS Service: lambda	4 days ago

Below this, the 'Roles Anywhere' section provides information on how to access non-AWS workloads:

- Access AWS from your non AWS workloads**: Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS.
- X.509 Standard**: Use your own existing PKI infrastructure or use AWS Certificate Manager Private Certificate Authority to authenticate identities.
- Temporary credentials**: Use temporary credentials with ease and benefit from the enhanced security they provide.

At the bottom right, there are links for 'Manage', 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

## 4) Now Create Lambda function

The screenshot shows the AWS Lambda Functions page. The left sidebar includes 'AWS Lambda' (selected), 'Dashboard', 'Applications', 'Functions' (selected), 'Additional resources' (with 'Code signing configurations', 'Event source mappings', 'Layers', and 'Replicas' sub-options), and 'Related AWS resources' (with 'Step Functions state machines' sub-option). The main content area displays 'Lambda > Functions' with a table showing one function:

Function name	Description	Package type	Runtime	Last modified
lambda-signup	-	Zip	Python 3.12	5 days ago

At the bottom right, there are links for 'CloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates.' and 'Privacy Terms Cookie preferences'.

lambda-signup

Description

Last modified  
5 days ago

Function ARN  
arn:aws:lambda:ap-south-1:654654341426:function:lambda-signup

Function URL

After creating lambda function, create environment variables.

Code Test Monitor Configuration Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

RDS databases

Monitoring and operations tools

Concurrency

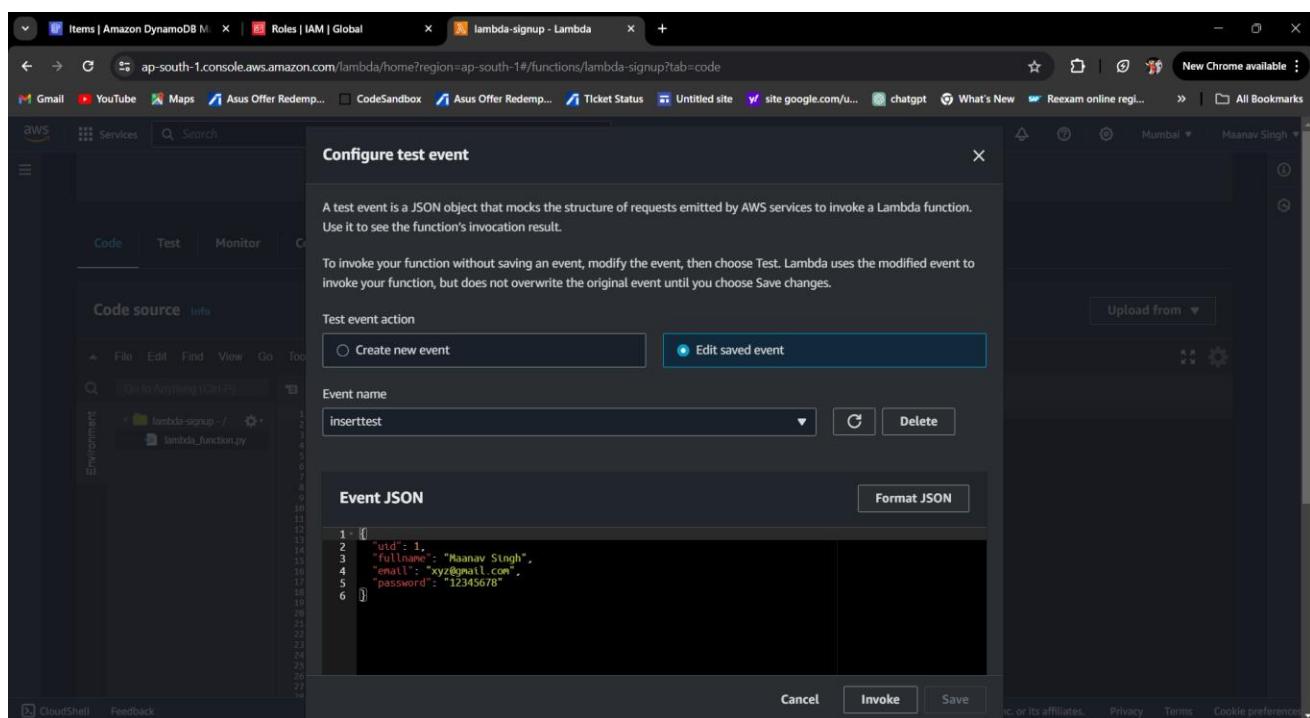
Environment variables (1)

Key	Value
DYNAMODB_TABLE_NAME	db-signup-table

After creating environment variable, now write your python script and test it.

The screenshot shows the AWS Lambda function code editor. The file being edited is `lambda_function.py`. The code implements a Lambda function named `lambda_handler` that takes an event and context as input. It generates a unique UUID for the user ID, extracts fullname, email, and password from the event, and inserts a new item into a DynamoDB table named `db-signup-table`. If successful, it returns a response with status code 200 and a message indicating success. If an exception occurs, it returns a response with status code 500 and an error message.

```
1 import json
2 import uuid
3 import boto3
4
5 dynamodb = boto3.resource('dynamodb')
6 table = dynamodb.Table('db-signup-table')
7
8 def lambda_handler(event, context):
9     try:
10         # Extract values from the event
11         fullname = event.get('fullname', '')
12         email = event.get('email', '')
13         password = event.get('password', '')
14
15         # Generate unique UUID
16         uid = str(uuid.uuid4())
17
18         # Insert record into DynamoDB
19         table.put_item(
20             Item={
21                 'uid': uid,
22                 'fullname': fullname,
23                 'email': email,
24                 'password': password
25             }
26         )
27
28         return {
29             'statusCode': 200,
30             'body': json.dumps({'message': 'Data saved successfully'})
31         }
32     except Exception as e:
33         return {
34             'statusCode': 500,
35             'body': json.dumps({'message': f'Error saving data to DynamoDB: {str(e)}'})
36         }
37
```



## 5) Create API Gateway

Screenshot of the AWS API Gateway - APIs page showing a single API named "api-signup".

The left sidebar shows the following navigation:

- APIs
- Custom domain names
- VPC links
- Usage plans
- API keys
- Client certificates
- Settings

The main content area displays the "APIs (1/1)" table with the following data:

Name	Description	ID	Protocol	API endpoint type	Created
api-signup	olunuzjrcce	REST	Regional	2024-04-22	

Buttons at the top right include "Create API", "Delete", and a refresh icon.

Screenshot of the AWS API Gateway - Resources page for the "api-signup" API.

The left sidebar shows the following navigation under "API: api-signup":

- Resources
- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

The main content area shows the "Resources" section for the root path "/".

Resource details:

- Path: /
- Resource ID: r5u1us5n2e

Methods:

Method type	Integration type	Authorization	API key
OPTIONS	Mock	None	Not required
POST	Lambda	None	Not required

Buttons at the top right include "API actions", "Deploy API", "Update documentation", and "Enable CORS".

The screenshot shows the AWS API Gateway Resources page for the 'api-signup' API. The left sidebar shows the API structure. The main panel displays the 'OPTIONS' method execution flow. The ARN is listed as arn:aws:execute-api:ap-south-1:654654341426:olunuzjrce/\*OPTIONS/. The flow diagram shows a 'Client' sending a 'Method request' to an 'Integration request', which then leads to a 'Mock integration' (indicated by a circular arrow icon). Below the flow diagram, tabs for Method request, Integration request, Integration response, Method response, and Test are visible. The 'Test' tab is selected.

The screenshot shows the AWS API Gateway Resources page for the 'api-signup' API. The left sidebar shows the API structure. The main panel displays the 'POST' method execution flow. The ARN is listed as arn:aws:execute-api:ap-south-1:654654341426:olunuzjrce/\*POST/. The flow diagram shows a 'Client' sending a 'Method request' to an 'Integration request', which then leads to a 'Lambda integration' (indicated by a Lambda icon). Below the flow diagram, tabs for Method request, Integration request, Integration response, Method response, and Test are visible. The 'Test' tab is selected.

## 6) Create VPC

The screenshot shows the AWS VPC console interface. On the left, there is a navigation sidebar with various options like EC2 Global View, Filter by VPC, and a list of Virtual private cloud components such as Subnets, Route tables, Internet gateways, Egress-only Internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main panel displays a table titled "Your VPCs" with two entries:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP o...
aws_vpc	vpc-009f1e3c295410f59	Available	172.31.0.0/16	-	dopt-02
test-vpc	vpc-01da18d1ffb09900c	Available	12.0.0.0/16	-	dopt-02

At the bottom of the main panel, there is a message: "Select a VPC above".

The screenshot shows the "VpcDetails | VPC Console" page for the VPC "vpc-01da18d1ffb09900c / test-vpc". The top navigation bar includes links for Items, Amazon DynamoDB, Roles, IAM, Global, lambda-signup - Lambda, API Gateway - Resources, and vpcs | VPC Console. The left sidebar is identical to the one in the first screenshot.

The main content area is titled "Details" and shows the following information for the VPC:

VPC ID	State	DNS hostnames	DNS resolution
vpc-01da18d1ffb09900c	Available	Disabled	Enabled
Tenancy	DHCP option set	Main route table	Main network ACL
Default	dopt-02a9bcc8d46ac5164	rtb-043a121b4e4a939a	acl-0e2f13210a8a3bfa5
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	12.0.0.0/16	-	-
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID	
Disabled	-	654654341426	

Below the details table, there are tabs for "Resource map", "CIDRs", "Flow logs", "Tags", and "Integrations". Under the "Resource map" tab, there are four cards: "VPC Show details", "Subnets (2)", "Route tables (2)", and "Netw".

## 7) Create subnets

Screenshot of the AWS VPC Subnets list page:

**Subnets (6) Info**

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-0a17bf4d35e8147e5	Available	vpc-009f1e3c295410f59   aws...	172.31.32.0/20
my-subnet-1	subnet-0835592c85ad6de3d	Available	vpc-009f1e3c295410f59   aws...	172.31.48.0/20
-	subnet-0153ffb3ac0854dd	Available	vpc-009f1e3c295410f59   aws...	172.31.16.0/20
-	subnet-0db309ce8e28fea5e	Available	vpc-009f1e3c295410f59   aws...	172.31.0.0/20
test-public-subnet-1b	subnet-0cc0a398c794912b1	Available	vpc-01da18d1ff0b09900c   test...	12.0.2.0/24
test-public-subnet-1a	subnet-0167065bfc6890117	Available	vpc-01da18d1ff0b09900c   test...	12.0.1.0/24

**Actions** Create subnet

Screenshot of the AWS VPC Subnet details page for subnet-0167065bfc6890117:

**subnet-0167065bfc6890117 / test-public-subnet-1a**

**Details**

Subnet ID subnet-0167065bfc6890117	Subnet ARN arn:aws:ec2:ap-south-1:654654341426:subnet/subnet-0167065bfc6890117	State Available	IPv4 CIDR 12.0.1.0/24
Available IPv4 addresses 250		Availability Zone ap-south-1a	Availability Zone ID aps1-az1
Network border group ap-south-1	IPv6 CIDR -	Route table rtb-0de52e2b22cd63df   test-public-RT	Network ACL acl-0e2f13210a8a3bfa5
Default subnet No	VPC vpc-01da18d1ff0b09900c   test-vpc	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No
Customer-owned IPv4 pool -	Auto-assign public IPv4 address No	IPv4 CIDR reservations -	IPv6 CIDR reservations -
IPv6-only No	Outpost ID -	Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled
DNS64 Disabled	Hostname type IP name	Owner 654654341426	

The screenshot shows the AWS VPC Subnet Details page for a subnet named 'subnet-0cc0a398c794912b1' within a VPC. The subnet has an ARN of arn:aws:ec2:ap-south-1:1654654341426:subnet/subnet-0cc0a398c794912b1, is in the 'Available' state, and is associated with the availability zone ap-south-1b. It is part of a route table rtb-0de52e2b22cd63df and is connected to a default subnet vpc-01da18d1ff09900c. The subnet has 250 available IPv4 addresses and no IPv6 CIDR assigned. Auto-assign IPv6 address is disabled, and IPv4 CIDR reservations are also disabled. The subnet is owned by user 654654341426.

## 8) Create Route Table

The screenshot shows the AWS Route Tables page. It lists three existing route tables: rtb-043a121be4e4a939a, rtb-0426fab6f79133c09, and test-public-RT. The test-public-RT route table is selected, showing it has 2 subnets associated with it. A modal window titled 'Select a route table' is open at the bottom, listing the same three route tables for selection.

The screenshot shows the AWS VPC Route Table Details page. The route table ID is rtb-0de52e2b22cdb63df, and its name is test-public-RT. It has two explicit subnet associations: Subnet 1 and Subnet 2. There are two routes defined:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0d28d36e03a423d17	Active	No
12.0.0.0/16	local	Active	No

## 9) Create Internet Gateway

The screenshot shows the AWS Internet Gateways page. There are two Internet Gateways listed:

Name	Internet gateway ID	State	VPC ID	Owner
igw-test	igw-0d28d36e03a423d17	Attached	vpc-01da18d1ffb09900c   test-vpc	654654341426
igw-aws-default	igw-0fcf74ae980da063	Attached	vpc-009f1e3c295410f59   aws_vpc	654654341426

The screenshot shows the AWS VPC console interface. The left sidebar is expanded, showing the 'Internet gateways' section under 'Virtual private cloud'. The main content area displays the details of an Internet gateway named 'igw-0d28d36e03a423d17'. The 'Details' tab is selected, showing the Internet gateway ID (igw-0d28d36e03a423d17), state (Attached), VPC ID (vpc-01da18d1ff0990c), and owner (654654341426). A 'Tags' section shows a single tag 'Name: igw-test'. The top navigation bar shows multiple tabs like 'Items | Amazon DynamoDB M...', 'Roles | IAM | Global...', 'Lambda-signup - Lambda...', 'API Gateway - Resources...', and 'InternetGateway | VPC Console...'. The browser address bar shows the URL: ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#InternetGateway:internetGatewayId=igw-0d28d36e03a423d17.

## 10) Create Endpoints to connect DynamoDB to VPC.

The screenshot shows the AWS VPC console interface. The left sidebar is expanded, showing the 'Endpoints' section under 'Virtual private cloud'. The main content area displays a table titled 'Endpoints (1) Info' with one entry: 'test-endpoint' (VPC endpoint ID: vpce-09bab96afbe5e9a5e, Status: Available, Service name: com.amazonaws.ap-south-1.dynamodb). The top navigation bar shows multiple tabs like 'Endpoints | VPC Management...', 'subnets | VPC Console...', and 'InternetGateway | VPC Console...'. The browser address bar shows the URL: ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#Endpoints:vpceEndpointId=vpce-09bab96afbe5e9a5e. The bottom status bar shows the date and time: 08-11-2024 15:40.

The screenshot shows the AWS VPC console with the URL [ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EndpointDetailsVpcEndpointId=vpce-09bab96afbe5e9a5e](https://ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EndpointDetailsVpcEndpointId=vpce-09bab96afbe5e9a5e). The main content area displays the details of a newly created endpoint named 'vpce-09bab96afbe5e9a5e / test-endpoint'. The endpoint ID is 'vpce-09bab96afbe5e9a5e', it is in an 'Available' status, and it was created on Friday 8 November 2024 at 15:29:58 GMT+5:30. The service name is 'com.amazonaws.ap-south-1.dynamodb'. The endpoint type is 'Gateway' and private DNS names are not enabled. A 'Route tables' tab is selected, showing one route table named 'private-route-table' associated with the endpoint.

## 11) Create S3 Bucket and upload your website code (HTML,CSS,JS etc.)

The screenshot shows the AWS S3 console with the URL [ap-south-1.console.aws.amazon.com/s3/home?region=ap-south-1#](https://ap-south-1.console.aws.amazon.com/s3/home?region=ap-south-1#). The main content area displays the 'Amazon S3' dashboard. Under the 'General purpose buckets' section, there is one bucket named 'buckypajl'. The bucket was created on April 15, 2024, at 15:54:30 (UTC+05:30). The bucket is located in the 'Asia Pacific (Mumbai) ap-south-1' region. The 'Create bucket' button is visible in the top right of the bucket list.

The screenshot shows the AWS S3 console interface. On the left, a sidebar lists various services like Amazon DynamoDB, Roles | IAM | Global, Lambda-signup - Lambda, API Gateway - Resources, vpcs | VPC Console, and buckypaji - S3 bucket. The main area displays the 'buckypaji' bucket. The 'Objects' tab is selected, showing one object: 'AWS\_Website\_Project.zip' (Type: zip, Last modified: April 22, 2024, 15:14:37 (UTC+05:30), Size: 856.3 KB, Storage class: Standard). There are buttons for Actions (Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, Upload), a search bar for 'Find objects by prefix', and a table header for Name, Type, Last modified, Size, and Storage class.

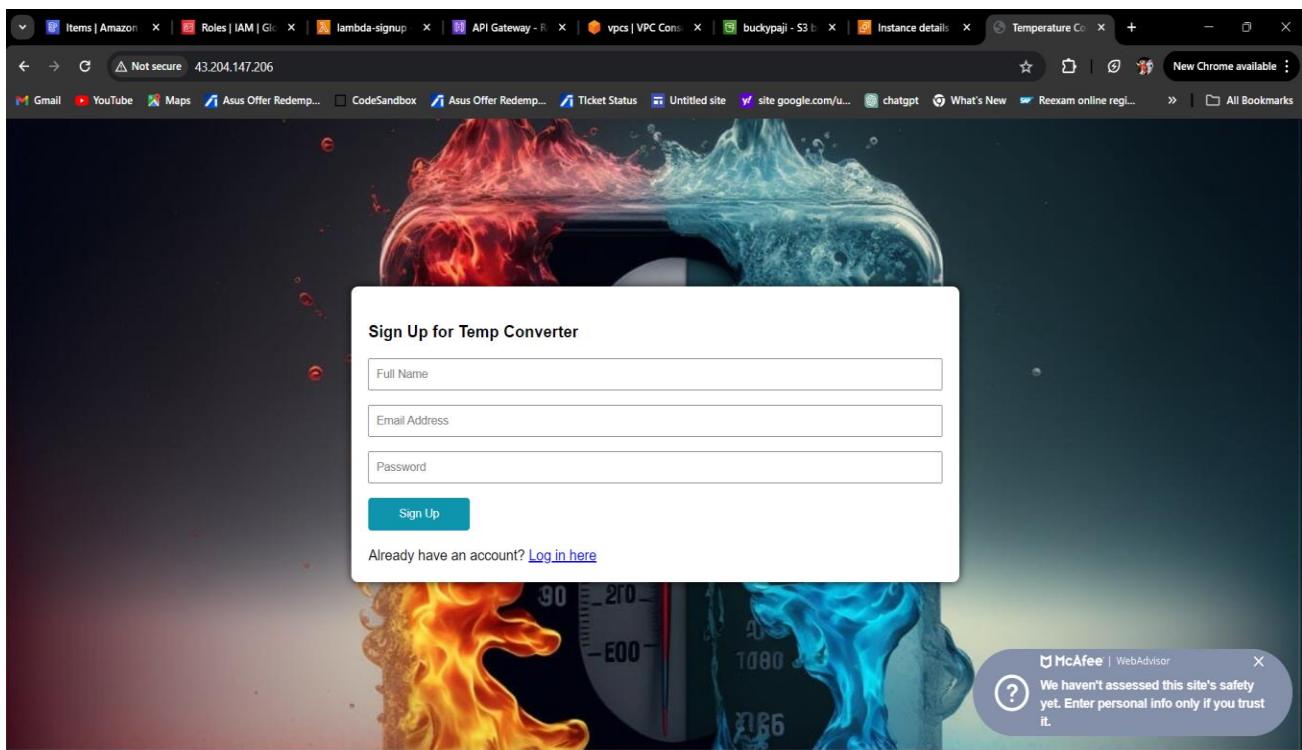
## 12) Now create EC2 instance and Host your website on it.

The screenshot shows the AWS EC2 Instances page. The sidebar includes options like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations. The main content area shows a table of instances with one entry: 'temp-website' (Instance ID: i-00f09a0392960c982, Instance state: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Availability Zone: ap-south-1b). A modal window titled 'Select an instance' is open at the bottom.

The screenshot shows the AWS EC2 Instances details page for instance `i-00f09a0392960c982`. The instance is running and has a public IPv4 address of `43.204.147.206`. The instance type is `t2.micro`, and it is associated with a VPC ID `vpc-01da18d1ffb09900c`. The subnet ID is `subnet-0cc0a398c794912b1`. The instance summary table includes fields for Instance ID, Public IPv4 address, Private IPv4 addresses, Instance state, Private IP DNS name, Instance type, VPC ID, Subnet ID, IAM Role, and Auto Scaling Group name.

After creating EC2 instance, copy the public ip and paste it on another tab to see your website properly hosted or not.

The screenshot shows the same AWS EC2 Instances details page as before, but with a tooltip overlaying the public IP address field. The tooltip contains the text "Copied" and the copied value `43.204.147.206`. This indicates that the public IP address has been successfully copied from the instance details page.



### 13) Create target group for load balancer.

The screenshot shows the AWS CloudFormation console with a stack named 'Temperature' in the 'Outputs' tab. The output 'Temperature API URL' is displayed as <https://43.204.147.206>. The 'Resources' tab shows a single resource: 'tg-temp' (AWS::ElasticLoadBalancingV2::TargetGroup). The 'Logs' tab shows deployment logs for the stack.

The screenshot shows the AWS EC2 Target Groups page. On the left, a sidebar menu includes options like AMIs, Services, Network & Security, Load Balancing, Auto Scaling, and more. The main content area displays a target group named 'tg-temp'. The 'Details' section shows the ARN: arn:aws:elasticloadbalancing:ap-south-1:654654341426:targetgroup/tg-temp/3607df65efef65a. It lists the Target type as 'Instance', Protocol as 'HTTP: 80', and Protocol version as 'HTTP1'. The VPC is set to 'vpc-01da18d1ffb0990c'. Below this, a table shows 1 Total targets, with 1 Healthy, 0 Unhealthy, 0 Unused, 0 Initial, and 0 Draining. A note indicates a distribution of targets by Availability Zone (AZ). At the bottom, tabs for Targets, Monitoring, Health checks, Attributes, and Tags are visible.

14) After creating target group, it's time to create load balancer.

The screenshot shows the AWS EC2 Load Balancers page. The sidebar menu includes options like AMIs, Services, Network & Security, Load Balancing, Auto Scaling, and more. The main content area displays a load balancer named 'lb-temp'. The 'Load balancers (1)' section shows the following details: Name is 'lb-temp', DNS name is 'lb-temp-597939375.ap-so...', State is 'Active', VPC ID is 'vpc-01da18d1ffb0990c', and there are '2 Availability Zones'. The Type is listed as 'application'. A note states that Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic. At the bottom, a message says '0 load balancers selected'.

The screenshot shows the AWS EC2 Load Balancers console. On the left, a sidebar lists various services and regions. The main panel displays the 'lb-temp' load balancer under the 'Load balancers' section. The 'Details' tab is selected, showing the following configuration:

Load balancer type	Status	VPC	IP address type
Application	Active	vpc-01da18d1ffb0990c	IPv4
Scheme	Internet-facing	Hosted zone ZP97RAFLXTNZK	Availability Zones subnet-0cc0a398c794912b1 (ap-south-1b (aps1-az3)) subnet-0167065bfc6890117 (ap-south-1a (aps1-az1))
Load balancer ARN	arn:aws:elasticloadbalancing:ap-south-1:654654341426:loadbalancer/app/lb-temp/ead83fd9e27b7653	DNS name lb-temp-597939375.ap-south-1.elb.amazonaws.com (A Record)	Date created April 22, 2024, 15:37 (UTC+05:30)

Below the details, there are tabs for 'Listeners and rules', 'Network mapping', 'Resource map - new', 'Security', 'Monitoring', 'Integrations', 'Attributes', and 'Tags'. The 'Listeners and rules' tab is currently active, showing '(1) info'. At the bottom, there are buttons for 'Manage rules', 'Manage listener', 'Add listener', and 'CloudShell Feedback'.

## 15) Now we are going to create Web ACLs in WAF.

The screenshot shows the AWS WAF & Shield console. The left sidebar has sections for 'AWS WAF' (Getting started, Web ACLs, Bot control dashboard, Application integration, IP sets, Regex pattern sets, Rule groups, AWS Marketplace managed rules) and 'AWS Shield' (Getting started, Overview). The main panel shows the 'Web ACLs' page for the 'Web ACLs (1)' section. The table displays one entry:

Name	Description	ID
waf-temp	-	b243e716-bb35-4481-80d7-5379625a1e1b

At the top right of the table, there are buttons for 'Asia Pacific (Mumbai)', 'Copy ARN', 'Delete', and 'Create web ACL'. Below the table, there is a search bar 'Find web ACLs' and navigation controls. The bottom of the screen includes standard AWS footer links: 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS WAF & Shield console. The left sidebar under 'AWS WAF' has 'Web ACLs' selected. The main content area shows the 'waf-temp' Web ACL. It includes sections for 'Data filters' (with options for 'Blocked', 'Allowed', 'Captcha', and 'Challenge'), 'Action totals for the specified time range - all traffic' (with tabs for 'Total', 'Blocked', 'Allowed', and 'Captcha'), and a summary table.

## 16) Create IP sets

The screenshot shows the AWS WAF & Shield console. The left sidebar under 'AWS WAF' has 'IP sets' selected. The main content area shows the 'IP sets (1)' page. It includes a search bar, a table with columns for 'Name', 'Description', and 'ID', and a single entry for 'ip-temp'.

The screenshot shows the AWS WAF IP sets configuration page. On the left, the navigation menu includes 'AWS WAF' and 'IP sets'. The main content area displays the details for the 'ip-temp' IP set, which has a name of 'ip-temp', is in the 'Asia Pacific (Mumbai)' region, and is using IPv4. It contains one IP address entry: '122.161.72.120/32'.

The screenshot shows the AWS WAF Web ACLs configuration page for 'waf-temp'. The navigation menu includes 'AWS WAF' and 'Web ACLs'. The main content area displays the 'Rules' tab, which lists a single rule named 'block-my-ip-temp' with an action of 'CAPTCHA' and priority 0. Below the rules, it shows '1/5000 WCUs' available.

17) After establishing WAF successfully now we will go back to EC2 instance and run the website and enter the required details and we're able to see the filled details in DynamoDB.

Screenshot of the AWS Cloud Console showing the EC2 Instances details for an instance named "temp-website".

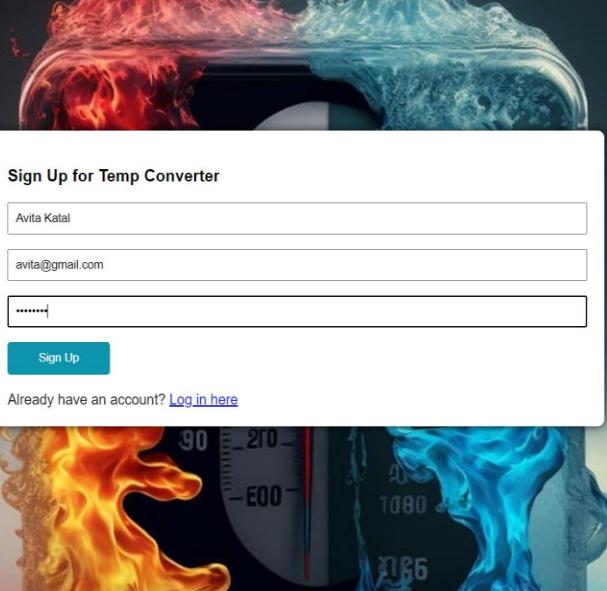
**Instance summary for i-00f09a0392960c982 (temp-website)**

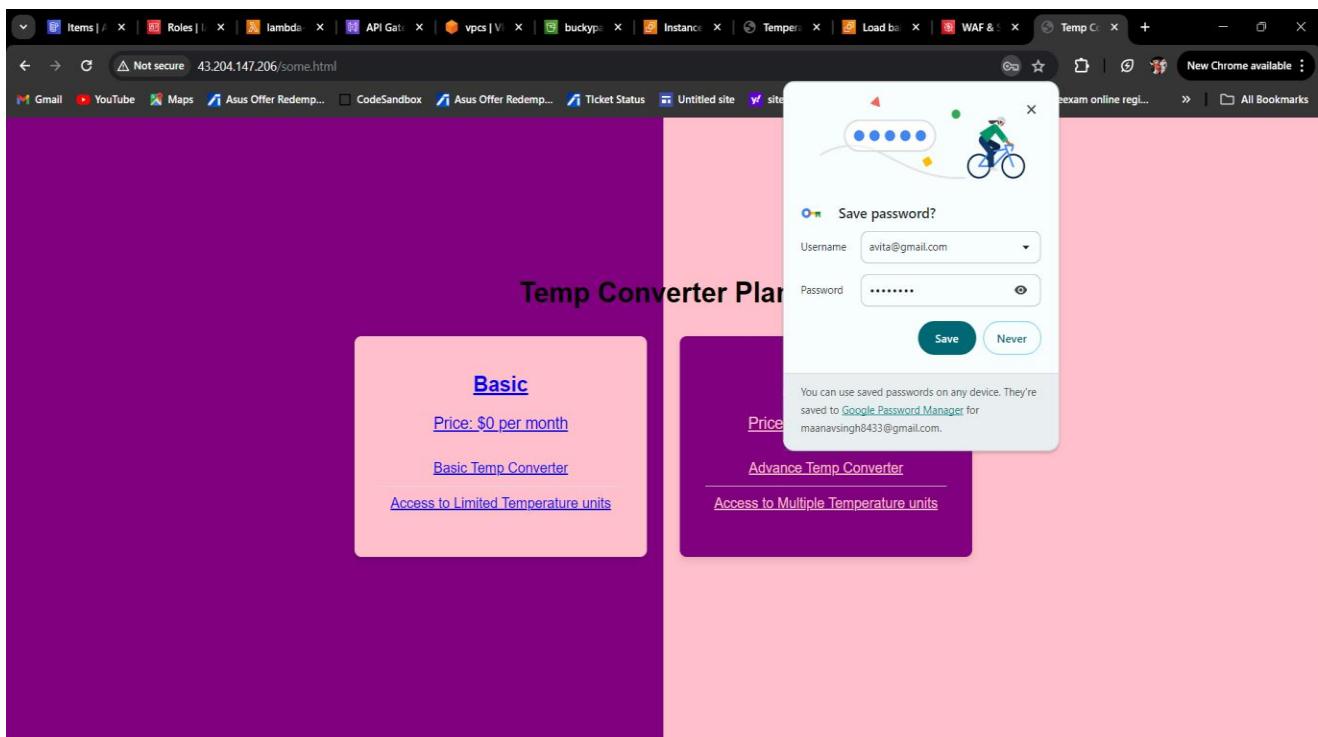
Attribute	Value
Instance ID	i-00f09a0392960c982 (temp-website)
Public IPv4 address	43.204.147.206 [open address]
Private IPv4 addresses	12.0.2.226
IPv6 address	-
Instance state	Running
Private IP DNS name (IPv4 only)	ip-12-0-2-226.ap-south-1.compute.internal
Public IPv4 DNS	-
Hostname type	IP name: ip-12-0-2-226.ap-south-1.compute.internal
Instance type	t2.micro
Elastic IP addresses	-
Spot Requests	-
VPC ID	vpc-01da18d1ffb0990c (test-vpc)
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations.
Reservations	New
Subnet ID	subnet-0cc0a398c794912b1 (test-public-subnet-1b)
Auto Scaling Group name	-
IMDSv2	Required
IAM Role	-

Screenshot of a web browser displaying a sign-up form for a service named "Temp Converter".

**Sign Up for Temp Converter**

Already have an account? [Log in here](#)





A screenshot of the AWS DynamoDB console. The left sidebar shows navigation options like Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, Settings, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main area shows a success message: "Selected items have been deleted successfully." Below this, the "db-signup-table" is selected. The interface includes tabs for "Scan or query items" (with "Scan" selected), "Select a table or index" (set to "Table - db-signup-table"), "Select attribute projection" (set to "All attributes"), and "Run" and "Reset" buttons. At the bottom, there's a section for "Items returned (1/5)" with "Actions" and "Create item" buttons.

The screenshot shows the AWS DynamoDB console interface. On the left, there's a navigation sidebar with options like Dashboard, Tables, Explore Items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. Below that is a section for DAX with Clusters, Subnet groups, Parameter groups, and Events. The main area has tabs for 'Items' (selected), 'Roles', 'Lambda', 'API Gateways', 'vpc', 'buckypes', 'Instances', 'Temperatures', 'Load balancers', 'WAF & S3', and 'Temporary'. A search bar at the top right includes a placeholder '[Alt+S]'. The main content area has a 'Filters' section with 'Run' and 'Reset' buttons. A green success message box says 'Completed. Read capacity units consumed: 0.5'. Below it, a table titled 'Items returned (1/5)' shows five rows of data:

	uid (String)	email	fullname	password
<input type="checkbox"/>	2fa9f1bf-fb12-4fa6-8...	xyz@gmail....	Maanav Singh	12345678
<input type="checkbox"/>	e96a3203-0bbe-4de3...	abc@gmail....	Raghav Aga...	87654321
<input type="checkbox"/>	ea7ff136c-e01b-4b4b...	maanavbea...	Maanav Singh	kdnfngos
<input checked="" type="checkbox"/>	0050e66e-b2eb-48aa...	avita@gmai...	Avita Katal	12345678
<input type="checkbox"/>	4a106be7-50dd-46bf...	tarushi010...	Tarushi cha...	iosuiafb

At the bottom, there are links for CloudShell, Feedback, and a footer with copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

# THANK YOU !!