

1. Architecture Evidence

API Gateway

The screenshot shows the AWS API Gateway Resources page. On the left sidebar, under the API: final-project-vvs-url-shortener section, the 'Resources' option is selected. The main panel displays a 'Resources' section with a 'Create resource' button and a tree view of the current resources. The root resource '/' has three methods: GET, OPTIONS, and POST. To the right, there is a 'Resource details' section showing the path '/' and Resource ID '2bx1uhq1dl', along with buttons for 'Update documentation' and 'Enable CORS'. Below it is a 'Methods (0)' section with a 'Create method' button.

Lambda Functions

The screenshot shows the AWS Lambda Functions page. On the left sidebar, under the 'Additional resources' section, the 'Functions' option is selected. The main panel displays a 'Functions (2)' table with two entries: 'final-project-vvs-create-url' and 'final-project-vvs-redirect-url'. Both functions are listed as 'Zip' type, 'Python 3.9' runtime, 'Standard' type, and were last modified 6 days ago and 7 days ago respectively. A 'Create function' button is located at the top right of the table.

DynamoDB Table

The screenshot shows the AWS DynamoDB Tables page. On the left sidebar, under the 'Tables' section, the 'final-project-user-urls' table is selected. The main panel displays the 'Settings' tab for this table. In the 'General information' section, the table's partition key is 'short_code (String)', capacity mode is 'On-demand', and item count is 15. The table status is 'Active' and Point-in-time recovery (PITR) is 'Off'. The table size is 1.6 kilobytes and the resource-based policy is 'Not active'. There are also sections for 'Indexes', 'Monitor', 'Global tables', and 'Backups'.

S3 Static Website

The screenshot shows the AWS S3 console with the 'Static website hosting' configuration for the bucket 'final-project-vvs'. The left sidebar includes sections for 'Amazon S3', 'Buckets', 'Access management and security', and 'Storage management and insights'. The main panel displays a summary of the static website hosting setup, including a recommendation to use AWS Amplify Hosting, the status of 'S3 static website hosting' (Enabled), the 'Hosting type' (Bucket hosting), and the 'Bucket website endpoint' (http://final-project-vvs.s3-website-us-east-1.amazonaws.com). A 'Create Amplify app' button is also present.

2. Security & IAM Proof

Lambda IAM Role

- With Least-privilege access

The screenshot shows the AWS IAM console with the 'Permissions' tab selected for the policy 'final-project-vvs-dynamodb-policy'. The left sidebar lists 'Identity and Access Management (IAM)' options like 'Dashboard', 'Access management', 'Policies', and 'Root access management'. The main panel displays the 'Permissions defined in this policy' section, which shows a single permission entry for 'DynamoDB' allowing 'Read, Write' access with specific resource and request conditions.

Service	Access level	Resource	Request condition
DynamoDB	Limited: Read, Write	region string like us-east-1, TableName string like final-project-user-urls	None

Lambda

- No secrets in code or environment variables

The screenshot shows the AWS Lambda configuration interface for a function named 'final-project-vvs-create-url'. The 'Configuration' tab is selected. On the left, a sidebar lists 'General configuration', 'Triggers', 'Permissions', 'Destinations', 'Function URL', 'Environment variables' (which is currently selected), and 'Tags'. The main panel is titled 'Environment variables (0)' and contains a search bar with placeholder text 'Find environment variables'. Below the search bar is a table with columns 'Key' and 'Value'. A message at the bottom states 'No environment variables associated with this function.' There is an 'Edit' button at the bottom right of the table.

3. Application Working Proof

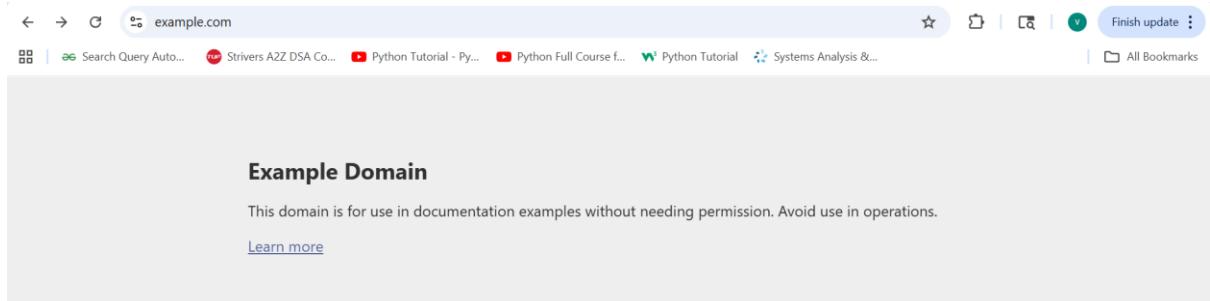
Write Path (POST) – DB write working

```
C:\Users\Admin\final-project-submission>curl -X POST "https://th1omwipo7.execute-api.us-east-1.amazonaws.com/prod/shorten" ^
More? -H "Content-Type: application/json" ^
More? -d "{\"url\":\"https://example.com\"}"
{"short_code": "duoyhB", "short_url": "https://th1omwipo7.execute-api.us-east-1.amazonaws.com/prod/duoyhB", "original_url": "https://example.com"}
C:\Users\Admin\final-project-submission>curl -X POST "https://th1omwipo7.execute-api.us-east-1.amazonaws.com/prod/shorten" ^
More? -H "Content-Type: application/json" ^
More? -d "{\"url\":\"https://example.com\"}"
{"short_code": "RNkg1Y", "short_url": "https://th1omwipo7.execute-api.us-east-1.amazonaws.com/prod/RNkg1Y", "original_url": "https://example.com"}
C:\Users\Admin\final-project-submission>
```

DB read working / Redirect verified

The screenshot shows the AWS API Gateway test interface. The left sidebar shows the 'APIs' section with 'Custom domain names', 'Domain name access associations', 'VPC links', and 'AgentCore targets'. Below that, under 'API: final-project-vvs-url-shortener', there are sections for 'Resources', 'Stages', 'Authorizers', and 'Gateway responses'. The main panel has a 'Create resource' button and a tree view of resources: '/' (with '/(short_code)', 'GET', '/shorten', 'OPTIONS', 'POST'). To the right, a 'Test' tab is open, showing a successful GET request to '/(short_code)'. The request details show a 'Request' of '/4gutVz' and a 'Latency ms' of 956. The response details show a 'Status' of 302 and a 'Response body' of 'No data'. The response headers include a Location header pointing to 'https://example.com' and X-Amzn-Trace-Id headers.

Redirect verified



4. Monitoring Evidence

CloudWatch Metrics - Lambda

