

S3 Bucket Details

The screenshot shows the Amazon S3 console interface. On the left is a navigation pane with 'Amazon S3' selected. The main content area is titled 's3-website-host' and has tabs for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The 'Properties' tab is active, showing a 'Bucket overview' section with details: Region (US East (N. Virginia) us-east-1), Amazon resource name (ARN) (arn:aws:s3::s3-website-host), and Creation date (January 29, 2021, 13:55:31 (UTC+05:30)). Below this is the 'Bucket Versioning' section, which states 'Versioning is a means of keeping multiple variants of an object in the same bucket...' and includes an 'Edit' button. At the bottom, it shows 'Bucket Versioning' is 'Disabled'.

Bucket overview		
Region	Amazon resource name (ARN)	Creation date
US East (N. Virginia) us-east-1	arn:aws:s3::s3-website-host	January 29, 2021, 13:55:31 (UTC+05:30)

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

[Edit](#)

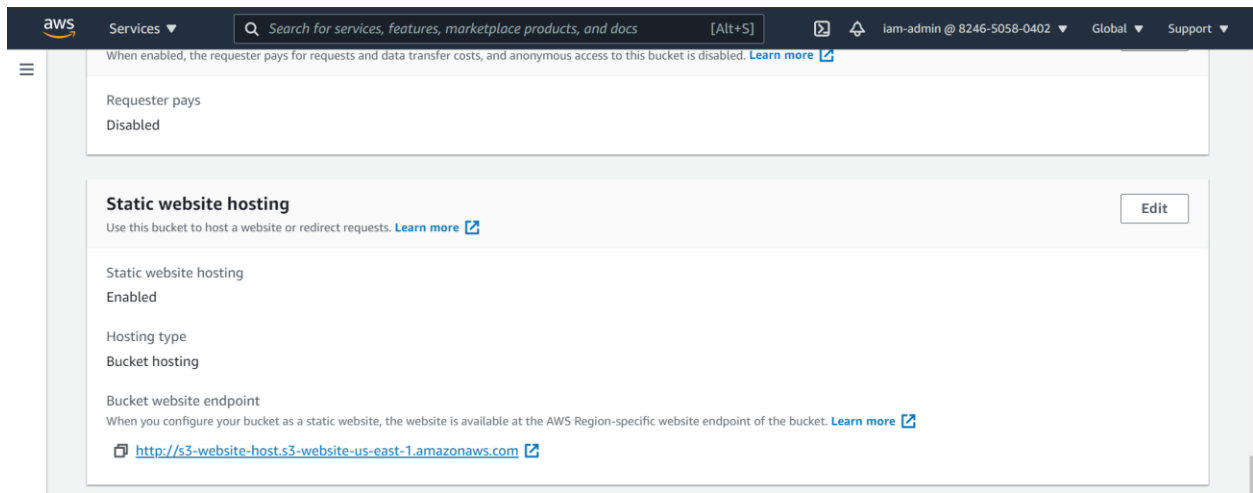
Bucket Versioning
Disabled

S3 Objects

The screenshot shows the 'Files and folders' tab of the 's3-website-host' bucket. It displays a list of 6 files and folders with a total size of 40.8 KB. The list includes a search bar and a table with columns for Name, Folder, Type, Size, Status, and Error.

Name	Folder	Type	Size	Status	Error
background.png	-	image/png	31.5 KB	✓ Succeeded	-
error.html	-	text/html	399.0 B	✓ Succeeded	-
index.html	-	text/html	383.0 B	✓ Succeeded	-
lu-cloud-small-sad.png	-	image/png	3.8 KB	✓ Succeeded	-
lu-cloud-small.png	-	image/png	4.3 KB	✓ Succeeded	-
style.css	-	text/css	454.0 B	✓ Succeeded	-

S3 Static Website Hosting



S3 Website Home Page



DynamoDB Table Details

The screenshot shows the AWS Management Console interface for a DynamoDB table named 'Vehicle'. The left sidebar contains a search bar and a list of tables, with 'Vehicle' selected. The main panel displays the 'Overview' tab for the 'Vehicle' table. The table details are as follows:

Property	Value
Table name	Vehicle
Primary partition key	Brand (String)
Primary sort key	Model (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	DEFAULT Manage Encryption
KMS Master Key ARN	Not Applicable
Encryption Status	
CloudWatch Contributor Insights	DISABLED Manage Contributor Insights NEW
Time to live attribute	DISABLED Manage TTL
Table status	Active
Creation date	January 29, 2021 at 2:07:50 PM UTC+5:30
Read/write capacity mode	Provisioned
Last change to on-demand mode	-
Provisioned read capacity units	5 (Auto Scaling Disabled)
Provisioned write capacity units	5 (Auto Scaling Disabled)
Last decrease time	-
Last increase time	-

DynamoDB Table Items

The screenshot shows the AWS Management Console interface for the 'Vehicle' table, specifically the 'Items' tab. The left sidebar is the same as in the previous screenshot. The main panel displays the 'Items' tab for the 'Vehicle' table. The table items are as follows:

Brand	Model
<input type="checkbox"/> Honda	Unicorn
<input type="checkbox"/> Honda	City
<input type="checkbox"/> Honda	Amaze
<input type="checkbox"/> Hyundai	I10

DynamoDB Table Querying

The screenshot displays the AWS Management Console interface for a DynamoDB table named "Vehicle". The top navigation bar includes the AWS logo, "Services", a search bar, and user information. The left sidebar shows the "Create table" and "Delete table" buttons, along with a search filter for table names. The main content area is titled "Vehicle" and includes tabs for "Overview", "Items", "Metrics", "Alarms", "Capacity", "Indexes", "Global Tables", and "Backups". The "Items" tab is selected, showing a query for "[Table] Vehicle: Brand, Model". The query interface includes a "Query" dropdown, a "Partition key" field set to "Brand" with a value of "TVS", and a "Sort key" field set to "Model" with a value of "Enter value". Below the query interface, a table lists the items returned by the query:

Brand	Model
TVS	Apache
TVS	Jupiter

EC2 Cluster Dashboard

The screenshot displays the 'sample-app-service' details page in the Amazon ECS console. The left sidebar shows navigation options for Amazon ECS, including Clusters, Task Definitions, and Account Settings. The main content area shows the service details for 'mycluster', which is in an 'ACTIVE' state. The service is a 'REPLICA' type, launched using 'FARGATE' launch type, and uses the 'AWSServiceRoleForECS' service role. The 'Created By' field shows the IAM role 'arn:aws:iam::824650580402:root'. The 'Desired count' is 1, 'Pending count' is 1, and 'Running count' is 0. Below the details, there are tabs for 'Details', 'Tasks', 'Events', 'Auto Scaling', 'Deployments', 'Metrics', 'Tags', and 'Logs'. The 'Load Balancing' section shows 'No load balancers'. The 'Network Access' section is partially visible.

Service : sample-app-service

Cluster: mycluster

Status: ACTIVE

Task definition: first-run-task-definition:1

Service type: REPLICA

Launch type: FARGATE

Service role: AWSServiceRoleForECS

Created By: arn:aws:iam::824650580402:root

Desired count: 1

Pending count: 1

Running count: 0

Details | Tasks | Events | Auto Scaling | Deployments | Metrics | Tags | Logs

Load Balancing

Load Balancer Name	Container Name	Container Port
No load balancers		

Network Access

This screenshot shows the 'Network Access' section of the 'sample-app-service' details page. It lists the 'Allowed VPC' as 'vpc-0f7ee673e5b5fa376', 'Allowed subnets' as 'subnet-0fb8171a4511ee697, subnet-0c85652de731c23d9', and 'Security groups*' as 'sg-085395a1379555141'. The 'Auto-assign public IP' is set to 'ENABLED'. The 'Load Balancing' section above it shows 'No load balancers'. The 'Details' tab is selected, and the 'Network Access' section is expanded.

Service type: REPLICA

Launch type: FARGATE

Service role: AWSServiceRoleForECS

Created By: arn:aws:iam::824650580402:root

Details | Tasks | Events | Auto Scaling | Deployments | Metrics | Tags | Logs

Load Balancing

Load Balancer Name	Container Name	Container Port
No load balancers		

Network Access

Allowed VPC: vpc-0f7ee673e5b5fa376

Allowed subnets: subnet-0fb8171a4511ee697, subnet-0c85652de731c23d9

Security groups*: sg-085395a1379555141

Auto-assign public IP: ENABLED

VPC Dashboard

The screenshot shows the 'Your VPCs (1)' dashboard in the AWS Management Console. The left sidebar shows the 'VIRTUAL PRIVATE CLOUD' section with options for 'Your VPCs', 'Subnets', and 'Route Tables'. The main content area shows a list of VPCs. A search filter is applied, showing 'search: vpc-0f7ee673e5b5fa376'. The table lists one VPC: 'ECS mycluster - VPC' with VPC ID 'vpc-0f7ee673e5b5fa376', State 'Available', and IPv4 CIDR '10.0.0.0/16'. The 'Create VPC' button is visible in the top right corner.

New VPC Experience

Tell us what you think

VPC Dashboard New

Filter by VPC:

Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs New

Subnets New

Route Tables

Your VPCs (1) Info

Filter VPCs

search: vpc-0f7ee673e5b5fa376 X Clear filters

	Name	VPC ID	State	IPv4 CIDR
<input type="checkbox"/>	ECS mycluster - VPC	vpc-0f7ee673e5b5fa376	Available	10.0.0.0/16

Create VPC

EBS Dashboard

The screenshot displays the AWS Elastic Beanstalk console. The left sidebar shows the 'Elastic Beanstalk' menu with options for Environments, Applications, and Change history. Under 'sample-webapp', there are links for Application versions and Saved configurations. The 'Samplewebapp-env' environment is selected, showing its details. The main content area for 'Samplewebapp-env' includes a 'Health' section with a green checkmark and 'Ok' status, a 'Running version' section with 'Sample Application' and an 'Upload and deploy' button, and a 'Platform' section with the Python logo and 'Python 3.7 running on 64bit Amazon Linux 2/3.1.4'. A 'Recent events' section is at the bottom with a 'Show all' button. The top navigation bar includes the AWS logo, a search bar, and user information.

Elastic Beanstalk ×

Environments
Applications
Change history

▼ sample-webapp
Application versions
Saved configurations

▼ **Samplewebapp-env**
Go to environment [🔗](#)
Configuration
Logs
Health
Monitoring

Samplewebapp-env
[Samplewebapp-env.eba-rmjg7yyv-us-east-2.elasticbeanstalk.com](#) (e-az2bki6xba)
Application name: **sample-webapp**

Refresh Actions

Health

Ok
Causes

Running version
Sample Application
Upload and deploy

Platform

Python 3.7 running on 64bit
Amazon Linux 2/3.1.4
Change

Recent events
Show all
< 1 >

EBS Sample Application

The screenshot shows the 'Congratulations' page in the AWS Elastic Beanstalk console. The left side has a green gradient background with the text 'Congratulations' and a message: 'Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud'. Below this, it says 'This environment is launched with Elastic Beanstalk Python Platform'. The right side has a light gray background with the heading 'What's Next?' and a list of links: 'AWS Elastic Beanstalk overview', 'AWS Elastic Beanstalk concepts', 'Deploy a Django Application to AWS Elastic Beanstalk', 'Deploy a Flask Application to AWS Elastic Beanstalk', 'Customizing and Configuring a Python Container', and 'Working with Logs'.

Congratulations

Your first AWS Elastic Beanstalk Python Application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Python Platform

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploy a Django Application to AWS Elastic Beanstalk](#)
- [Deploy a Flask Application to AWS Elastic Beanstalk](#)
- [Customizing and Configuring a Python Container](#)
- [Working with Logs](#)

