

# **FINAL PROJECT REPORT**

---

**IS 698 – Special Topics in Information Systems**

**CLOUD COMPUTING**

---

**“ BOOKVERSE ”**

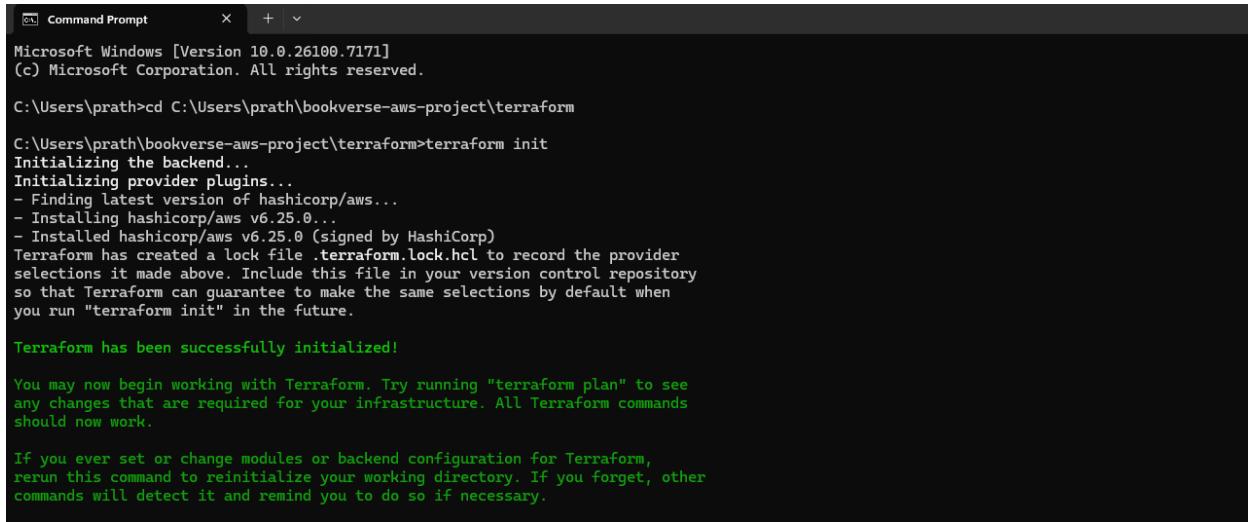
**SCREENSHOT DOCUMENT**

Submitted By

**Prathyusha Harish Kumar  
JB24771**

## TERRAFORM SCREENSHOTS

### terraform init



```
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

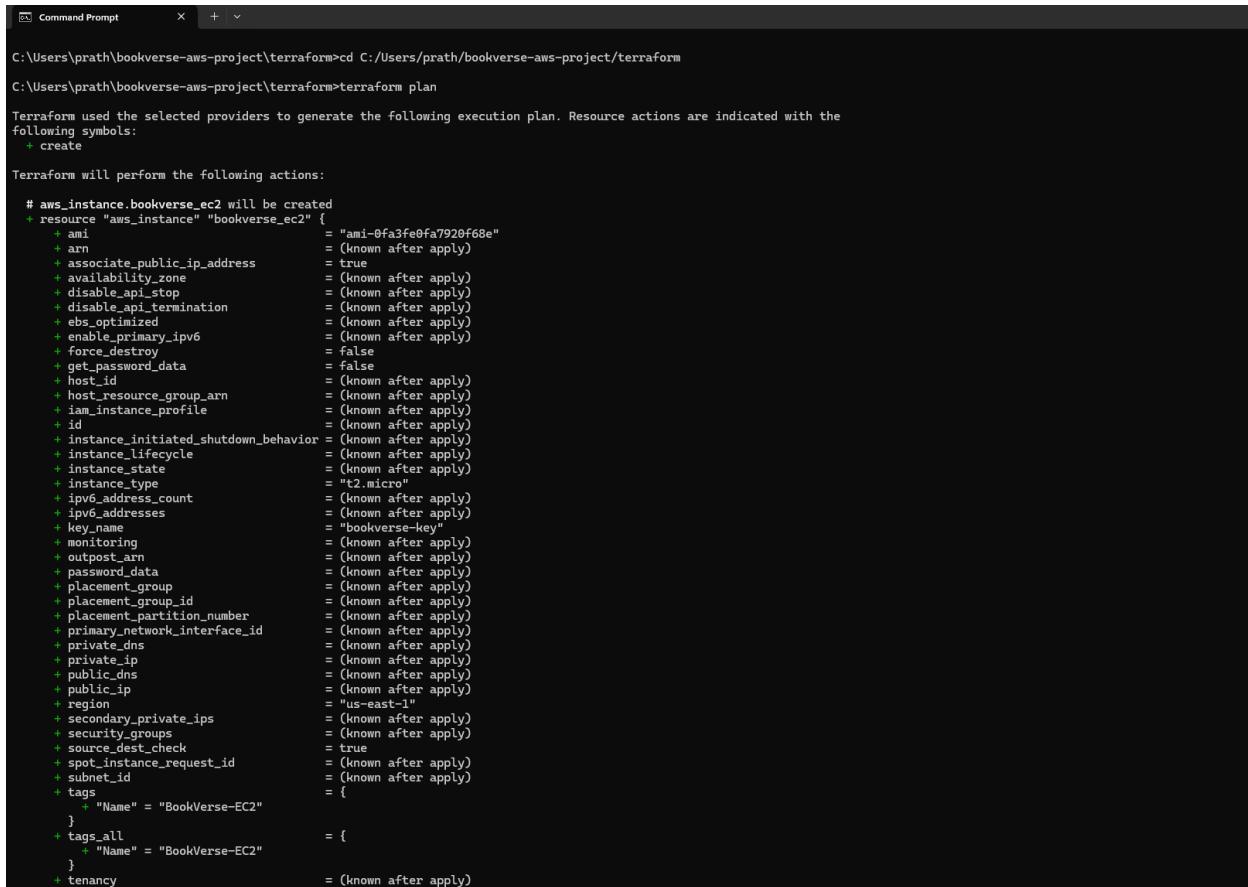
C:\Users\prath>cd C:\Users\prath\bookverse-aws-project>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.25.0...
- Installed hashicorp/aws v6.25.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

### terraform plan



```
C:\Users\prath\bookverse-aws-project>cd C:/Users/prath/bookverse-aws-project>terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.bookverse_ec2 will be created
+ resource "aws_instance" "bookverse_ec2" {
    + ami                                = "ami-0fa3fe0fa7920f68e"
    + arn                                = "(known after apply)"
    + associate_public_ip_address        = true
    + availability_zone                  = "(known after apply)"
    + disable_api_stop                  = "(known after apply)"
    + disable_api_termination           = "(known after apply)"
    + ebs_optimized                     = "(known after apply)"
    + enable_primary_ipv6               = "(known after apply)"
    + force_destroy                      = false
    + get_password_data                = false
    + host_id                            = "(known after apply)"
    + host_resource_group_arn          = "(known after apply)"
    + iam_instance_profile              = "(known after apply)"
    + id                                 = "(known after apply)"
    + instance_initiated_shutdown_behavior = "(known after apply)"
    + instance_lifecycle                = "(known after apply)"
    + instance_state                    = "(known after apply)"
    + instance_type                     = "t2.micro"
    + ipv6_address_count                = "(known after apply)"
    + ipv6_addresses                   = "(known after apply)"
    + key_name                           = "bookverse-key"
    + monitoring                         = "(known after apply)"
    + outpost_arn                       = "(known after apply)"
    + password_data                     = "(known after apply)"
    + placement_group                  = "(known after apply)"
    + placement_group_id                = "(known after apply)"
    + placement_partition_number       = "(known after apply)"
    + primary_network_interface_id     = "(known after apply)"
    + private_dns                       = "(known after apply)"
    + private_ip                        = "(known after apply)"
    + public_dns                         = "(known after apply)"
    + public_ip                          = "(known after apply)"
    + region                             = "us-east-1"
    + secondary_private_ips            = "(known after apply)"
    + security_groups                   = "(known after apply)"
    + source_dest_check                = true
    + spot_instance_request_id         = "(known after apply)"
    + subnet_id                          = "(known after apply)"
    + tags
        + "Name" = "BookVerse-EC2"
    }
    + tags_all                          = {
        + "Name" = "BookVerse-EC2"
    }
    + tenancy                           = "(known after apply)"
```

```
  Command Prompt + - x
+ tags           = {
+   * "Name" = "BookVerse-EC2"
}
+ tags_all       = {
+   * "Name" = "BookVerse-EC2"
}
+ tenancy        = (known after apply)
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ primary_network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

# aws_internet_gateway.igw will be created
resource "aws_internet_gateway" "igw" {
+ arn          = (known after apply)
+ id          = (known after apply)
+ owner_id    = (known after apply)
+ region      = "us-east-1"
+ tags         = {
+   * "Name" = "BookVerse-IGW"
}
+ tags_all     = {
+   * "Name" = "BookVerse-IGW"
}
+ vpc_id       = (known after apply)
}

# aws_key_pair.bookverse_key will be created
resource "aws_key_pair" "bookverse_key" {
+ arn          = (known after apply)
+ fingerprint  = (known after apply)
+ id          = (known after apply)
+ key_name    = "bookverse-key"
+ key_name_prefix = (known after apply)
}
```

```
  Command Prompt + - x
+ public_key     = "AAAAB3NzC1yc2EAAADAQABAAQOWUm/qG6GTzrI+jpms3G4xs6bzmlQ/K9KBbXgefMS/4Cq5de09q90AFDkQyPtvwjnabcVV/FvO9VGvL5nhx9EL0jQd9384+JsrErHyGenLV/RcvMmOrjChO5/c8Kw2/bpR1CqXF3eJPGY9MyBNax9sWFBKL2HvrSMkvBLfJgfh823mlMc5Ft2+g3T1ZddMYy95xzV7UEepYFyamdu4md2PQ/N/zkr04Mt10qbZ4qcjp445D6koQ3HCAK53PEisTk28815sBVFYCCy2FdIH4/izxUTZ6y7uGjRyUNxSk21b1QBz/vdhf6QoC+xquqLcSBPmbWxL9"
+ region        = "us-east-1"
+ tags_all      = (known after apply)

# aws_route_table.public_rt will be created
resource "aws_route_table" "public_rt" {
+ arn          = (known after apply)
+ id          = (known after apply)
+ owner_id    = (known after apply)
+ propagating_vgws = (known after apply)
+ region      = "us-east-1"
+ route       = [
+   {
+     + cidr_block      = "0.0.0.0/0"
+     + gateway_id     = (known after apply)
+     # (11 unchanged attributes hidden)
+   },
+   {
+     + tags          = {
+       * "Name" = "BookVerse-Public-RT"
+     }
+     + tags_all      = {
+       * "Name" = "BookVerse-Public-RT"
+     }
+   }
]
+ vpc_id       = (known after apply)
}

# aws_route_table_association.public_assoc will be created
resource "aws_route_table_association" "public_assoc" {
+ id          = (known after apply)
+ region      = "us-east-1"
+ route_table_id = (known after apply)
+ subnet_id   = (known after apply)
}

# aws_security_group.alb_sg will be created
resource "aws_security_group" "alb_sg" {
+ arn          = (known after apply)
+ description  = "Allow HTTP/HTTPS"
+ egress       = [
+   {
+     + cidr_blocks  = [
+       "+0.0.0.0/0",
+     ]
+     + from_port   = 0
+     + ipv6_cidr_blocks = []
+     + prefix_list_ids = []
+     + protocol    = "-1"
+     + security_groups = []
+     + self        = false
+     + to_port    = 0
+     # (1 unchanged attribute hidden)
+   },
+   {
+     + id          = (known after apply)
+     + ingress     = [

```

```

  Command Prompt x + 
}

# aws_subnet.public_subnet will be created
+ resource "aws_subnet" "public_subnet" {
  + arn = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone = "us-east-1a"
  + availability_zone_id = (known after apply)
  + cidr_block = "10.0.1.0/24"
  + enable_dns64 = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id = (known after apply)
  + ipv6_cidr_block_association_id = (known after apply)
  + ipv6_native = false
  + map_public_ip_on_launch = true
  + owner_id = (known after apply)
  + private_dns_hostname_type_on_launch = (known after apply)
  + region = "us-east-1"
  + tags = {
    + "Name" = "BookVerse-Public-Subnet"
  }
  + tags_all = {
    + "Name" = "BookVerse-Public-Subnet"
  }
  + vpc_id = (known after apply)
}

# aws_vpc.bookverse_vpc will be created
+ resource "aws_vpc" "bookverse_vpc" {
  + arn = (known after apply)
  + cidr_block = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id = (known after apply)
  + enable_dns_hostnames = true
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + id = (known after apply)
  + instance_tenancy = "default"
  + ipv6_association_id = (known after apply)
  + ipv6_cidr_block = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id = (known after apply)
  + region = "us-east-1"
  + tags = {
    + "Name" = "BookVerse-VPC"
  }
  + tags_all = {
    + "Name" = "BookVerse-VPC"
  }
}

Plan: 10 to add, 0 to change, 0 to destroy.

```

## terraform apply -auto-approve

```

  Command Prompt x + 
you run "terraform apply" now.

C:\Users\prath\bookverse-aws-project\terraform>terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.bookverse_ec2 will be created
+ resource "aws_instance" "bookverse_ec2" {
  + ami = "ami-0fa3fe0fa7920f68e"
  + arn = (known after apply)
  + associate_public_ip_address = true
  + availability_zone = (known after apply)
  + disable_api_stop = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized = (known after apply)
  + enable_primary_ipv6 = (known after apply)
  + force_destroy = false
  + get_password_data = false
  + host_id = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile = (known after apply)
  + id = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle = (known after apply)
  + instance_state = "running"
  + instance_type = (known after apply)
  + ipv6_addresses_count = (known after apply)
  + ipv6_addresses = (known after apply)
  + key_name = "bookverse-key"
  + monitoring = (known after apply)
  + outpost_arn = (known after apply)
  + password_data = (known after apply)
  + placement_group = (known after apply)
  + placement_group_id = (known after apply)
  + placement_partition_number = (known after apply)
  + private_ipnetwork_interface_id = (known after apply)
  + private_dns = (known after apply)
  + private_ip = (known after apply)
  + public_dns = (known after apply)
  + public_ip = (known after apply)
  + region = "us-east-1"
  + secondary_private_ips = (known after apply)
  + security_groups = (known after apply)
  + source_dest_check = true
  + spot_instance_request_id = (known after apply)
  + subnet_id = (known after apply)
  + tags = {
    + "Name" = "BookVerse-EC2"
  }
  + tags_all = {
    + "Name" = "BookVerse-EC2"
  }
  + tenancy = (known after apply)
  + user_data_base64 = (known after apply)
}

```

```

[1] Command Prompt x + v
    + "Name" = "BookVerse-EC2"
  + tags_all = {
    + "Name" = "BookVerse-EC2"
  }
  + tenancy = (known after apply)
  + user_data_base64 = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids = (known after apply)
  + capacity_reservation_specification (known after apply)
  + cpu_options (known after apply)
  + ebs_block_device (known after apply)
  + enclave_options (known after apply)
  + ephemeral_block_device (known after apply)
  + instance_market_options (known after apply)
  + maintenance_options (known after apply)
  + metadata_options (known after apply)
  + network_interface (known after apply)
  + primary_network_interface (known after apply)
  + private_dns_name_options (known after apply)
  + root_block_device (known after apply)
}

# aws_internet_gateway.igw will be created
+ resource "aws_internet_gateway" "igw" {
  + arn = (known after apply)
  + id = (known after apply)
  + owner_id = (known after apply)
  + region = "us-east-1"
  + tags = {
    + "Name" = "BookVerse-IGW"
  }
  + tags_all = {
    + "Name" = "BookVerse-IGW"
  }
  + vpc_id = (known after apply)
}

# aws_key_pair.bookverse_key will be created
+ resource "aws_key_pair" "bookverse_key" {
  + arn = (known after apply)
  + fingerprint = (known after apply)
  + id = (known after apply)
  + key_name = "bookverse-key"
  + key_name_prefix = (known after apply)
  + key_pair_id = (known after apply)
}

[2] Command Prompt x + v
  + placement_group_id = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns = (known after apply)
  + private_ip = (known after apply)
  + public_dns = (known after apply)
  + public_ip = (known after apply)
  + region = "us-east-1"
  + secondary_private_ips = (known after apply)
  + security_groups = (known after apply)
  + source_dest_check = true
  + spot_instance_request_id = (known after apply)
  + subnet_id = "subnet-0354126c9c502e0c1"
  + tags = {
    + "Name" = "BookVerse-EC2"
  }
  + tags_all = {
    + "Name" = "BookVerse-EC2"
  }
  + tenancy = (known after apply)
  + user_data_base64 = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids = [
    + "sg-0dc3cc80e43d4339",
  ]
  + capacity_reservation_specification (known after apply)
  + cpu_options (known after apply)
  + ebs_block_device (known after apply)
  + enclave_options (known after apply)
  + ephemeral_block_device (known after apply)
  + instance_market_options (known after apply)
  + maintenance_options (known after apply)
  + metadata_options (known after apply)
  + network_interface (known after apply)
  + primary_network_interface (known after apply)
  + private_dns_name_options (known after apply)
  + root_block_device (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.
aws_instance.bookverse_ec2: Creating...
aws_instance.bookverse_ec2: Still creating... [00m10s elapsed]
aws_instance.bookverse_ec2: Creation complete after 13s [id=i-09ea8075004f8685e]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

```

## Terraform output

```
C:\Users\prath\bookverse-aws-project>terraform output
alb_sg_id = "sg-05bebfb5bda8b8cebb"
ami_id = "ami-0fa3fe0fa7920f68e"
ec2_sg_id = "sg-0dc33cc80e43d4339"
key_name = "bookverse-key"
private_subnet_ids = [
  "subnet-0b50328c1f88045c5",
]
public_subnet_ids = [
  "subnet-0354126c9c502e0c1",
]
vpc_id = "vpc-0911a60255e733ec6"
```

## CLOUDFORMATION

```
C:\Users\prath\bookverse-aws-project\cloudformation>cd C:\Users\prath\bookverse-aws-project\cloudformation  
  
C:\Users\prath\bookverse-aws-project\cloudformation>dir  
 Volume in drive C is Windows  
 Volume Serial Number is EED3-89E9  
  
Directory of C:\Users\prath\bookverse-aws-project\cloudformation  
  
12/05/2025  10:30 AM    <DIR>          .  
12/02/2025  04:17 PM    <DIR>          ..  
12/02/2025  03:33 PM           0 .gitkeep  
12/05/2025  10:30 AM           3,250 webapp.yaml  
                  2 File(s)      3,250 bytes  
                  2 Dir(s)  812,376,682,496 bytes free
```

```
C:\Users\prath\bookverse-aws-project\cloudformation>aws cloudformation validate-template --template-body file://webapp.yaml --region us-east-1
{
  "Parameters": [
    {
      "ParameterKey": "KeyName",
      "NoEcho": false,
      "Description": "Name of an existing EC2 KeyPair"
    },
    {
      "ParameterKey": "AMIID",
      "NoEcho": false,
      "Description": "AMI ID for EC2 instances (Amazon Linux 2 recommended)"
    },
    {
      "ParameterKey": "GitRepoUrl",
      "DefaultValue": "https://github.com/PrathyushaHarishKumar-JB24771/bookverse-aws-project.git",
      "NoEcho": false,
      "Description": "GitHub repo URL to clone on instance"
    },
    {
      "ParameterKey": "VPCId",
      "NoEcho": false,
      "Description": "VPC ID (from Terraform)"
    },
    {
      "ParameterKey": "PublicSubnetId",
      "NoEcho": false,
      "Description": "Public subnet ID for ALB"
    },
    {
      "ParameterKey": "ALBSecurityGroup",
      "NoEcho": false,
      "Description": "Security Group ID for ALB"
    },
    {
      "ParameterKey": "EC2SecurityGroup",
      "NoEcho": false,
      "Description": "Security Group ID for EC2 instances"
    },
    {
      "ParameterKey": "InstanceType",
      "DefaultValue": "t3.micro",
      "NoEcho": false,
      "Description": "EC2 instance type"
    },
    {
      "ParameterKey": "PrivateSubnetId",
      "NoEcho": false,
      "Description": "Private subnet ID for EC2 instances (ASG)"
    }
  ],
  "Description": "BookVerse EC2 + ALB + AutoScaling (parameterized for Terraform-managed VPC/subnets/SGs)"
}
```

## Stack Creation

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The title bar includes the AWS logo, a search bar, and navigation links for CloudFormation > Stacks > Create stack. The top right corner shows 'United States (N. Virginia)'.

**Step 1: Create stack**

**Prerequisite - Prepare template**

You can also create a template by scanning your existing resources in the [CloudFormation generator](#).

**Prepare template**

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

**Choose an existing template** (selected) Upload or choose an existing template.

**Build from Infrastructure Composer** Create a template using a visual builder.

**Specify template info**

This [GitHub repository](#) contains sample CloudFormation templates that can help you get started on new infrastructure projects. [Learn more](#)

**Template source**

Selecting a template generates an Amazon S3 URL where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

**Amazon S3 URL** Provide an Amazon S3 URL to your template.

**Upload a template file** Upload your template directly to the console.

**Sync from Git** Sync a template from your Git repository.

**Upload a template file**

webapp.yaml

JSON or YAML formatted file

S3 URL: <https://s3.us-east-1.amazonaws.com/cf-templates-u6khzbupbz-us-east-1/2025-12-05T155019.4392ejv-webapp.yaml>

[View in Infrastructure Composer](#)

[Cancel](#) [Next](#)

AWS CloudFormation Step 1 Create stack Step 2 Specify stack details Step 3 Configure stack options Step 4 Review and create

## Specify stack details

**Provide a stack name**

**Stack name**

bookverse-web-stack

Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 19/128.

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**ALBSecurityGroup**  
Security Group ID for ALB

Enter String

**AMId**  
AMI ID for EC2 instances (Amazon Linux 2 recommended)

Enter String

**EC2SecurityGroup**  
Security Group ID for EC2 instances

Enter String

**GitRepoUrl**  
GitHub repo URL to clone on instance

https://github.com/PrathyushaHarishKumar-JB24771/bookverse-aws-project.git

**InstanceType**  
EC2 instance type

t3.micro

**KeyName**  
Name of an existing EC2 KeyPair

bookverse-key

**PrivateSubnetId**  
Private subnet ID for EC2 instances (ASG)

Enter String

**PublicSubnetId**  
Public subnet ID for ALB

The screenshot shows the AWS CloudFormation console with the following details:

- Region:** United States (N. Virginia)
- Stacks (0):** No stacks found.
- bookverse-web-stack:** A stack currently in progress, indicated by the "CREATE\_IN\_PROGRESS" status in the table below.
- Stack Info:** The active tab, showing basic information about the stack.
- Events:** One event listed: "1821924b-c2a8-476b-b9fc-e382580caed5" from 2025-12-05 10:52:49 UTC-0500.
- Resources:** No resources displayed.
- Outputs:** No outputs displayed.
- Parameters:** No parameters displayed.
- Template:** No template displayed.
- Change sets:** No change sets displayed.
- Git sync:** No git sync displayed.
- Actions:** Buttons for Delete, Update stack, Stack actions, and Create stack.

Stacks (1)			
<input type="text" value="Search by stack name"/> Filter status			
Stack name	Status	Created time	Description
<a href="#">BookVerse-StackWeb</a>	<span>CREATE_COMPLETE</span>	2025-12-05 20:35:05 UTC-0500	BookVerse Application Stack - EC2 (t3.micro), ALB, RDS, S3, Lambda

## Stack Outputs Verification

```
C:\Users\prath\bookverse-aws-project\cf>aws cloudformation describe-stacks --stack-name BookVerse-StackWeb --query "Stacks[0].Outputs" --output table --region us-east-1
|-----+-----+-----|
| Description | OutputKey | OutputValue |
|-----+-----+-----|
| RDS endpoint address | ROSEndpoint | bookverse-db.cala26qu66fk.us-east-1.rds.amazonaws.com |
| Application Load Balancer DNS name (URL) | ALBDNS | http://bookverse-elb-2131284506.us-east-1.elb.amazonaws.com |
| Lambda function name | LambdaName | bookverse-log-s3-uploads |
| S3 bucket name for uploads | UploadBucketName | bookverse-uploads-649418801823 |

C:\Users\prath\bookverse-aws-project\cf>aws cloudformation describe-stacks --stack-name BookVerse-StackWeb --query "Stacks[0].Outputs[?OutputKey=='UploadBucketName'].OutputValue" --output text --region us-east-1
bookverse-uploads-649418801823

C:\Users\prath\bookverse-aws-project\cf>aws cloudformation describe-stacks --stack-name BookVerse-StackWeb --query "Stacks[0].Outputs[?OutputKey=='LambdaName'].OutputValue" --output text --region us-east-1
bookverse-log-s3-uploads

C:\Users\prath\bookverse-aws-project\cf>aws lambda get-function --function-name bookverse-log-s3-uploads --query "Configuration.FunctionArn" --output text --region us-east-1
arn:aws:lambda:us-east-1:649418801823:function:bookverse-log-s3-uploads

C:\Users\prath\bookverse-aws-project\cf>
```

## ALB Website Check



## S3 Bucket Notification

The screenshot shows the AWS Amazon S3 console for the 'bookverse-uploads' bucket. Key sections include:

- AWS CloudTrail data events:** You can view and configure CloudTrail data events for Amazon S3 bucket object-level operations.
- Event notifications (1):** Send a notification when specific events occur in your bucket. One entry is listed: "bookverse-s3-to-lambda" (Event type: All object create events, Destination type: Lambda function, Destination: bookverse-log-s3-uploads).
- Amazon EventBridge:** For additional capabilities, use Amazon EventBridge to build event-driven applications at scale using S3 event notifications.
- Transfer acceleration:** Use an accelerated endpoint for faster data transfers.
- Object Lock:** Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely.
- Requester pays:** When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled.

## Log Verification

The screenshot shows the AWS CloudWatch Log Management interface. On the left, there's a navigation sidebar with sections like CloudWatch, Favorites and recents, Dashboards, Alarms, AI Operations, General Observability, Application Signals (APM), Infrastructure Monitoring, Logs, Metrics, Network Monitoring, and Setup. The main area is titled 'Log events' and contains a search bar with the placeholder 'Filter events - press enter to search'. Below the search bar is a table with two columns: 'Timestamp' and 'Message'. The table lists several log entries from December 12, 2025, at 06:48:21.9332. The messages include various error and warning logs related to Lambda runtime issues, such as unhandled exceptions and import module errors. At the top right of the main area, there are buttons for Actions, Start tailing, Create metric filter, Clear, and time range selection (1m, 30m, 1h, 12h, Custom, UTC timezone, Display).

## Create Bucket and Upload

```
C:\Users\prath\bookverse-aws-project\boto3>python create_bucket_and_upload.py
Creating bucket: bookverse-boto3-0fc94320
Uploading boto3-test-file.txt to s3://bookverse-boto3-0fc94320/
✓ Done.
✓ Bucket: bookverse-boto3-0fc94320
✓ Object: boto3-test-file.txt
```

## Boto3 script to list running EC2 instances

## Boto3 script to invoke Lambda

```
C:\Users\prath\bookverse-aws-project\boto3>python create_bucket_and_upload.py
Creating bucket: bookverse-boto3-0fc94320
Uploading boto3-test-file.txt to s3://bookverse-boto3-0fc94320/
✓ Done.
✓ Bucket: bookverse-boto3-0fc94320
✓ Object: boto3-test-file.txt

C:\Users\prath\bookverse-aws-project\boto3>cd C:\Users\prath\bookverse-aws-project\boto3

C:\Users\prath\bookverse-aws-project\boto3>python list_running_ec2.py
Running EC2 instances in us-east-1:
- i-00c0e3ae4aa049b21 | t3.micro | running | 10.0.3.88

C:\Users\prath\bookverse-aws-project\boto3>python invoke_lambda.py
Invoking Lambda: bookverse-log-s3-uploads
StatusCode: 200
Response payload: {"statusCode": 200, "body": "logged"}
```

## Connecting bastion EC2 instance to the RDS database

```

[ec2-user@ip-10-0-2-197:~] x + ~
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prath>ssh -i bookverse-key.pem ec2-user@50.17.100.62
Warning: Identity file bookverse-key.pem not accessible: No such file or directory.
The authenticity of host '50.17.100.62 (50.17.100.62)' can't be established.
ED25519 key fingerprint is SHA256:Gfgv+VZf+iwAMb6E7dHNdsPipIdR00jn9BbYe20gtw8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '50.17.100.62' (ED25519) to the list of known hosts.
ec2-user@50.17.100.62: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

C:\Users\prath>ssh -i C:\Users\prath\bookverse-aws-project\bookverse-key.pem ec2-user@50.17.100.62
#_
#_ _ _ _ Amazon Linux 2023
#_ \_ _ _ _ 
#_ \_ _ _ _ https://aws.amazon.com/linux/amazon-linux-2023
#_ \_ _ _ _ / 
#_ \_ _ _ _ /m/
#_ \_ _ _ _ /m/ 

[ec2-user@ip-10-0-2-197:~] x + ~
[ec2-user@ip-10-0-2-197 ~]$ sudo dnf install -y mariadb105
Last metadata expiration check: 0:04:02 ago on Sat Dec  6 10:58:33 2025.
Dependencies resolved.
=====
| Package           | Architecture | Version      | Repository | Size |
|=====|
| Installing:      |             |             |            |       |
| mariadb105        | x86_64       | 3:10.5.29-1.amzn2023.0.1 | amazonlinux | 1.5 M |
| Installing dependencies: |             |             |            |       |
| mariadb-connector-c | x86_64       | 3:3.10-1.amzn2023.0.1 | amazonlinux | 211 k |
| mariadb-connector-c-configure | noarch     | 3:3.10-1.amzn2023.0.1 | amazonlinux | 9.9 k |
| mariadb105-common | x86_64       | 3:10.5.29-1.amzn2023.0.1 | amazonlinux | 28 k |
| perl-Sys-Hostname | x86_64       | 1.23-477.amzn2023.0.7 | amazonlinux | 16 k |
|=====|
Transaction Summary
=====
Install 5 Packages

Total download size: 1.8 M
Installed size: 19 M
Downloading Packages:
(1/5): mariadb-connector-c-configure-3.3.10-1.amzn2023.0.1.noarch.rpm 292 kB/s | 9.9 kB  00:00
(2/5): mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64.rpm 5.0 MB/s | 211 kB  00:00
(3/5): mariadb105-10.5.29-1.amzn2023.0.1.x86_64.rpm 28 MB/s | 1.5 MB  00:00
(4/5): mariadb105-common-10.5.29-1.amzn2023.0.1.x86_64.rpm 1.2 MB/s | 28 kB  00:00
(5/5): perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64.rpm 797 kB/s | 16 kB  00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing
  Installing : mariadb-connector-c-configure-3.3.10-1.amzn2023.0.1.noarch 1/5
  Installing : mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64 1/5
  Installing : mariadb105-common-3:10.5.29-1.amzn2023.0.1.x86_64 2/5
  Installing : perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64 3/5
  Installing : mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64 4/5
  Running scriptlet: mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64 5/5
  Verifying   : mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64 5/5
  Verifying   : mariadb-connector-c-configure-3.3.10-1.amzn2023.0.1.noarch 1/5
  Verifying   : mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64 2/5
  Verifying   : mariadb105-common-3:10.5.29-1.amzn2023.0.1.x86_64 3/5
  Verifying   : perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64 4/5
  Verifying   : mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64 5/5
=====
Installed:
  mariadb-connector-c-3.3.10-1.amzn2023.0.1.x86_64      mariadb-connector-c-configure-3.3.10-1.amzn2023.0.1.noarch
  mariadb105-3:10.5.29-1.amzn2023.0.1.x86_64          mariadb105-common-3:10.5.29-1.amzn2023.0.1.x86_64
  perl-Sys-Hostname-1.23-477.amzn2023.0.7.x86_64

Complete!
[ec2-user@ip-10-0-2-197 ~]$ mysql --version
mysql Ver 15.1 Distrib 10.5.29-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-10-0-2-197 ~]$ mysql -h bookverse-db.cala26qu66fk.us-east-1.rds.amazonaws.com \
-u admin -p
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'10.0.2.197' (using password: YES)

[ec2-user@ip-10-0-2-197:~] x + ~
[ec2-user@ip-10-0-2-197 ~]$ mysql -h bookverse-db.cala26qu66fk.us-east-1.rds.amazonaws.com \
-p
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'10.0.2.197' (using password: YES)
[ec2-user@ip-10-0-2-197 ~]$ mysql -h bookverse-db.cala26qu66fk.us-east-1.rds.amazonaws.com \
-u admin -p
Enter password:
Mysql [(none)]> MariaDB monitor.  Commands end with ; or \q.
Your MySQL connection id is 89
Server version: 8.0.43 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> \h

General information about MariaDB can be found at
http://mariadb.org

List of all client commands:
Note that all text commands must be first on line and end with ';'
?          (\?) Synonyms for 'help'.
charsets (\c) List available character sets. Might be needed for processing binlog with multi-byte charsets.
clear      (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set delimiter for statements.
edit      (\e) Edit command with EDITOR.
ego       (\g) Send command to MariaDB server, display result vertically.
exit      (\q) Exit from MariaDB.
go        (\g) Send command to MariaDB server.
help      (\h) Display this help.
interactive (\i) Read input from stdin.
notes    (\n) Don't write into outfile.
nowarning (\w) Don't show warnings after every statement.
pager    (\P) Set pager to a program. Print the query results via PAGER.
print    (\p) Print current command.
prompt   (\r) Change your mysql prompt.
quit    (\q) Exit from MySQL.
rehash   (\R) Recreate completion hash.
sandbox  (\S) Disallow commands that access the file system (except \P without an argument and \e).
source   (\.) Execute a file. Takes a file name as an argument.
status   (\s) Get status information from the server.
system   (\!) Execute a system shell command.
tee      (\t) Write output to given outfile.
use     (\u) Use another database. Takes database name as argument.
warnings (\W) Show warnings after every statement.

For server side help, type 'help contents'.

```

## Creation and Functional Database Working

```
ec2-user@ip-10-0-2-197:~ + v

MySQL [bookverse]>
MySQL [bookverse]> CREATE TABLE books (
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> title VARCHAR(256),
    -> author VARCHAR(255),
    -> genre VARCHAR(50)
    -> );
ERROR 1060 (42S01): Table 'books' already exists
MySQL [bookverse]>
MySQL [bookverse]> INSERT INTO books (title, author, genre) VALUES
    -> ('Pride and Prejudice', 'Jane Austen', 'Romance'),
    -> ('1984', 'George Orwell', 'Fiction'),
    -> ('Sapiens', 'Yuval Noah Harari', 'Non-Fiction');
Query OK, 3 rows affected (0.005 sec)
Records: 3  Duplicates: 0  Warnings: 0

MySQL [bookverse]> CREATE DATABASE IF NOT EXISTS bookversedb;
Query OK, 1 row affected (0.010 sec)

MySQL [bookverse]> USE bookversedb;
Database changed
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(256),
    author VARCHAR(255),
    genre VARCHAR(50)
);
MySQL [bookversedb]>
MySQL [bookversedb]> CREATE TABLE IF NOT EXISTS books (
    -> id INT AUTO_INCREMENT PRIMARY KEY,
    -> title VARCHAR(256),
    -> author VARCHAR(255),
    -> genre VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.029 sec)

MySQL [bookversedb]> INSERT INTO books (title, author, genre) VALUES
    -> ('Pride and Prejudice', 'Jane Austen', 'Romance'),
    -> ('Outlander', 'Diana Gabaldon', 'Romance'),
    -> ('Wuthering Heights', 'Emily Brontë', 'Romance'),
    -> ('The Notebook', 'Nicholas Sparks', 'Romance'),
    -> ('Jane Eyre', 'Charlotte Brontë', 'Romance'),
    -> ('To Kill a Mockingbird', 'Harper Lee', 'Fiction'),
    -> ('The Hobbit', 'J.R.R. Tolkien', 'Fiction'),
    -> ('Project Hail Mary', 'Andy Weir', 'Fiction'),
    -> ('1984', 'George Orwell', 'Fiction'),
    -> ('The Alchemist', 'Paulo Coelho', 'Fiction'),
    -> ('The Diary of a Young Girl', 'Anne Frank', 'Non-Fiction'),
    -> ('Sapiens', 'Yuval Noah Harari', 'Non-Fiction'),
    -> ('The Glass Castle', 'Jeannette Walls', 'Non-Fiction'),
    -> ('Atomic Habits', 'James Clear', 'Non-Fiction'),
    -> ('Man's Search for Meaning', 'Viktor E. Frankl', 'Non-Fiction');
Query OK, 15 rows affected (0.006 sec)
Records: 15  Duplicates: 0  Warnings: 0

MySQL [bookversedb]> SELECT * FROM books;
+----+-----+-----+-----+
| id | title          | author        | genre       |
+----+-----+-----+-----+
|  1 | Pride and Prejudice | Jane Austen   | Romance     |
|  2 | Outlander      | Diana Gabaldon | Romance     |
|  3 | Wuthering Heights | Emily Brontë  | Romance     |
|  4 | The Notebook    | Nicholas Sparks | Romance     |
|  5 | Jane Eyre      | Charlotte Brontë | Romance     |
|  6 | To Kill a Mockingbird | Harper Lee   | Fiction     |
|  7 | The Hobbit      | J.R.R. Tolkien | Fiction     |
|  8 | Project Hail Mary | Andy Weir    | Fiction     |
|  9 | 1984           | George Orwell | Fiction     |
| 10 | The Alchemist   | Paulo Coelho | Fiction     |
| 11 | The Diary of a Young Girl | Anne Frank | Non-Fiction |
| 12 | Sapiens         | Yuval Noah Harari | Non-Fiction |
| 13 | The Glass Castle | Jeannette Walls | Non-Fiction |
| 14 | Atomic Habits   | James Clear   | Non-Fiction |
| 15 | Man's Search for Meaning | Viktor E. Frankl | Non-Fiction |
+----+-----+-----+-----+
15 rows in set (0.002 sec)
```

## Proof that EC2 Instance can reach VPC

```
[ec2-user@ip-10-0-2-197 ~]$ nano db_check.py
[ec2-user@ip-10-0-2-197 ~]$ python3 db_check.py
Traceback (most recent call last):
  File "/home/ec2-user/db_check.py", line 1, in <module>
    import mysql.connector
ModuleNotFoundError: No module named 'mysql'
[ec2-user@ip-10-0-2-197 ~]$ pip3 install --user mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.4.0-cp39-cp39-manylinux_2_28_x86_64.whl (33.9 MB)
[ec2-user@ip-10-0-2-197 ~]$ pip3 show mysql-connector-python
Name: mysql-connector-python
Version: 9.4.0
Summary: A self-contained Python driver for communicating with MySQL servers, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249).
Home-page: UNKNOWN
Author: Oracle and/or its affiliates
Author-email:
License: GNU GPLv2 (with FOSS License Exception)
Location: /home/ec2-user/.local/lib/python3.9/site-packages
Requires:
Required-by:
[ec2-user@ip-10-0-2-197 ~]$ python3 db_check.py
Connecting to RDS...
Connected. Sample rows:
- [Romance] Pride and Prejudice by Jane Austen (id=1)
- [Romance] Outlander by Diana Gabaldon (id=2)
- [Romance] Wuthering Heights by Emily Brontë (id=3)
- [Romance] The Notebook by Nicholas Sparks (id=4)
- [Romance] Jane Eyre by Charlotte Brontë (id=5)
- [Fiction] To Kill a Mockingbird by Harper Lee (id=6)
- [Fiction] The Hobbit by J.R.R. Tolkien (id=7)
- [Fiction] Project Hail Mary by Andy Weir (id=8)
- [Fiction] 1984 by George Orwell (id=9)
- [Fiction] The Alchemist by Paulo Coelho (id=10)
[ec2-user@ip-10-0-2-197 ~]$
```

## EC2 MetaData Retrieval

```
[ec2-user@ip-10-0-2-197 ~] Microsoft Windows [Version 10.0_26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prath>ssh -i C:\Users\prath\bookverse-aws-project\bookverse-key.pem ec2-user@50.17.100.62
#_
#_ _###_ Amazon Linux 2023
#_ \###_ 
#_ \##_ 
#_ \#_ https://aws.amazon.com/linux/amazon-linux-2023
#_ V-'-->
#_ / 
#_ / 
#_ / 
#_ / 
#_ / 
#_ / 
Last login: Sat Dec  6 10:58:04 2025 from 71.121.211.119
[ec2-user@ip-10-0-2-197 ~]$ sudo dnf install -y python3-pip
Last metadata expiration check: 0:18:29 ago on Sat Dec  6 10:58:33 2025.
Dependencies resolved.

=====
Package           Architecture   Version      Repository  Size
=====
Installing:
  python3-pip          noarch    21.3.1-2.amzn2023.0.14  amazonlinux 1.8 M
Installing weak dependencies:
  libcrypt-compat       x86_64    4.4.33-7.amzn2023            amazonlinux 92 K

Transaction Summary
=====
Install 2 Packages

Total download size: 1.9 M
Installed size: 11 M
Downloading Packages:
(1/2): libcrypt-compat-4.4.33-7.amzn2023.x86_64.rpm  2.2 MB/s |  92 kB   00:00
(2/2): python3-pip-21.3.1-2.amzn2023.0.14.noarch.rpm  27 MB/s | 1.8 MB   00:00

Total                                         19 MB/s | 1.9 MB   00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Installing : libcrypt-compat-4.4.33-7.amzn2023.x86_64 1/1
  Installing : python3-pip-21.3.1-2.amzn2023.0.14.noarch 2/2
  Running scriptlet: python3-pip-21.3.1-2.amzn2023.0.14.noarch 2/2
  Verifying  : libcrypt-compat-4.4.33-7.amzn2023.x86_64 1/1
  Verifying  : python3-pip-21.3.1-2.amzn2023.0.14.noarch 2/2

Installed:
  libcrypt-compat-4.4.33-7.amzn2023.x86_64                                python3-pip-21.3.1-2.amzn2023.0.14.noarch

Complete!
[ec2-user@ip-10-0-2-197 ~]$ pip3 install --user boto3 requests
Collecting boto3
  Downloading boto3-1.42.4-py3-none-any.whl (140 kB)
     140 kB 15.7 MB/s
Requirement already satisfied: requests in /usr/lib/python3.9/site-packages (2.25.1)
```

```

[ec2-user@ip-10-0-2-197: ~] x + 
Collecting boto3
  Downloading boto3-1.42.4-py3-none-any.whl (140 kB)
[ec2-user@ip-10-0-2-197: ~] x + 
Requirement already satisfied: requests in /usr/lib/python3.9/site-packages (2.25.1)
Collecting botocore<1.43.0,>=1.42.4
  Downloading botocore-1.42.4-py3-none-any.whl (14.5 MB)
[ec2-user@ip-10-0-2-197: ~] x + 
Collecting s3transfer<0.17.0,>=0.16.0
  Downloading s3transfer-0.16.0-py3-none-any.whl (86 kB)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/lib/python3.9/site-packages (from boto3) (0.10.0)
Requirement already satisfied: charset<5,>=3.0.2 in /usr/lib/python3.9/site-packages (from requests) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/lib/python3.9/site-packages (from requests) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/lib/python3.9/site-packages (from requests) (1.25.10)
Requirement already satisfied: s3transfer<0.17.0,>=0.16.0 in /usr/lib/python3.9/site-packages (from botocore<1.43.0,>=1.42.4->boto3) (2.8.1)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1->botocore<1.43.0,>=1.42.4->boto3) (1.15.0)
Installing collected packages: botocore, s3transfer, boto3
Successfully installed boto3-1.42.4 botocore-1.42.4 s3transfer-0.16.0
[ec2-user@ip-10-0-2-197: ~] x + 
[ec2-user@ip-10-0-2-197: ~]$ nano get_instance_metadata.py
[ec2-user@ip-10-0-2-197: ~]$ python3 get_instance_metadata.py
EC2 Instance Metadata (IMDSv2):
- Instance ID: i-0a5c231b4ed93b72e
- Instance Type: t3.micro
- Local IPv4: 10.0.2.197
- Availability Zone: us-east-1b

[ec2-user@ip-10-0-2-197: ~] x + 
[ec2-user@ip-10-0-2-197: ~]$ ssh -l C:\Users\prathibookverse-aws-project\bookverse-key.pem ec2-user@50.17.100.62
Warning: Identity file C:\Users\prathibookverse-aws-project\bookverse-key.pem not accessible: No such file or directory.
The authenticity of host '50.17.100.62' (ED25519) can't be established.
ED25519 key fingerprint is SHA256:Gfgv+VZFIwMbe67dNdsPipIdR00jn9BbYe20gtw8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '50.17.100.62' (ED25519) to the list of known hosts.
[ec2-user@50.17.100.62: ~]$ Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-0-2-197: ~]$ python3 get_instance_metadata.py
EC2 Instance Metadata (IMDSv2):
- Instance ID: i-0a5c231b4ed93b72e
- Instance Type: t3.micro
- Local IPv4: 10.0.2.197
- Availability Zone: us-east-1b
/home/ec2-user/.local/lib/python3.9/site-packages/boto3/compat.py:89: PythonDeprecationWarning: Boto3 will no longer support Python 3.9 starting April 29, 2026. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.10 or later. More information can be found here: https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/
warnings.warn(warning, PythonDeprecationWarning)

Additional details via Boto3 describe_instances:
- State: running
- Launch Time: 2025-12-06 10:48:28+00:00
- Tags:
  * Name: bookverse-bastion
EC2 Instance Metadata (IMDSv2):
- Instance ID: i-0a5c231b4ed93b72e
- Instance Type: t3.micro
- Local IPv4: 10.0.2.197
- Availability Zone: us-east-1b

Additional details via Boto3 describe_instances:
- State: running
- Launch Time: 2025-12-06 10:48:28+00:00
- Tags:
  * Name: bookverse-bastion
[ec2-user@ip-10-0-2-197: ~]$ |

```

## Verification Screenshots:

### RDS

DB identifier	Status	Role	Engine	Upgrade rollout order	Region ...	Size	Recommendations	CPU	Current activity	Ma...
bookverse-db	Available	Instance	MySQL Co...	SECOND	us-east-1a	db.t3.micro			3.18%	0 Connections

## EC2

The screenshot shows the AWS EC2 Instances page. A single instance, "BookVerse" with ID i-00c0e3aa049b21, is listed as "Running". It is an t3.micro instance type, located in us-east-1a, with 5/3 checks passed. The instance has no public IP and is using an elastic IP. The status bar at the bottom indicates "less than a minute ago".

## Security Groups

The screenshot shows the AWS Security Groups page. Five security groups are listed: "launch-wizard-1", "default", "bookverse-rds-sg", "bookverse-ec2-sg", and "bookverse-alb-sg". Each group has a specific VPC ID, description, owner, and inbound rules. For example, "bookverse-ec2-sg" allows DB access from EC2 instances.

## Key Pair

The screenshot shows the AWS Key pairs page. One key pair, "bookverse-key", is listed. It is an RSA type key pair created on 2025/12/05 10:10 GMT-5. The fingerprint is 1a:8f:6c:a9:f6:7e:33:dd:03:87:d5:8a:1c:d4:81:b7, and its ID is key-0662339f5160deb45.

## Network Interface

The screenshot shows the AWS Network interfaces page. Four network interfaces are listed, each associated with a specific subnet, VPC ID, availability zone, security group, interface type, and description. For example, "eni-0515a65c7ea6f5e7" is an elastic network interface for the ELB app "bookverse-alb".

## Load Balancer

The screenshot shows the AWS Load balancers page. One load balancer, "bookverse-alb", is listed as active. It is an application load balancer (ALB) with an Internet-facing IP address (vpc-0a6bf1bc3448c6d3b). It has two availability zones and is associated with the security group sg-05a81eba686b49ad and the DNS name bookverse-alb-2131284506. The ARN is arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/bookverse-alb/2131284506.

## Target Group

The screenshot shows the AWS EC2 Target groups page. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, and Cloud Watch Metrics. The main content area displays a table for 'Target groups (1)'. The table has columns for Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. One entry is listed: bookverse-tg, arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/bookverse-tg, 80, HTTP, Instance, bookverse-alb, vpc-0a6bf1bc3448c6d3b.

## Auto Scaling Groups

The screenshot shows the AWS Auto Scaling Groups page. The left sidebar includes links for Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, and AMIs. The main content area displays a table for 'Auto Scaling groups (1)'. The table has columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, Availability Zones, and Creation time. One entry is listed: bookverse-ssg, bookverse-launch-template | Version 1, 1, - (status), 1, 1, 3, 2 Availability Zones, Fri Dec 05 2025 20:35:29 GMT-0500 (Eastern Standard Time).

## VPC

The screenshot shows the AWS VPC Your VPCs page. The left sidebar includes links for AWS Global View, Filter by VPC, Virtual private cloud, Your VPCs (Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways), and Regional. The main content area displays a table for 'Your VPCs (2)'. The table has columns for Name, VPC ID, State, Encryption controls, Block Public, IPv4 CIDR, IPv6 CIDR, and DHCP option set. Two entries are listed: bookverse-vpc, vpc-0a6bf1bc3448c6d3b, Available, -, Off, 10.0.0.0/16, -, dopt-05c6a27f9be674176; bookverse-public, vpc-024633ef0fa801d2, Available, -, Off, 172.31.0.0/16, -, dopt-05c6a27f9be674176.

## Subnets

The screenshot shows the AWS VPC Subnets page. The left sidebar includes links for AWS Global View, Filter by VPC, Virtual private cloud, Your VPCs (Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections, Route servers), and Security. The main content area displays a table for 'Subnets (10)'. The table has columns for Name, Subnet ID, State, VPC, Block Public, IPv4 CIDR, IPv6 CIDR, IPv6 CIDR association ID, and Available. Ten entries are listed across three VPCs: bookverse-private-a, bookverse-public-b, bookverse-public-a, bookverse-private-b, bookverse-public-b, bookverse-public-a, bookverse-private-b, bookverse-public-b, bookverse-public-a, and bookverse-private-b.

## Route Table

VPC dashboard > Route tables

Route tables (3) info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
-	rtb-0f55a594c7ea09f	-	-	Yes	vpc-0a6bf1bc3448c6d3b   book...	649418801823
bookverse-public-rt	rtb-0b31297974f6354	2 subnets	-	No	vpc-0a6bf1bc3448c6d3b   book...	649418801823
-	rtb-0f1397cb9e19ed44	-	-	Yes	vpc-024633ef0fa801d2	649418801823

## Internet Gateway

VPC dashboard > Internet gateways

Internet gateways (2) info

Name	Internet gateway ID	State	VPC ID	Owner
bookverse-igw	igw-0d591bc85e38584c0	Attached	vpc-0a6bf1bc3448c6d3b   bookverse-vpc	649418801823
-	igw-0d73a9b882ecc1ed4	Attached	vpc-024633ef0fa801d2	649418801823

## Security Groups

VPC dashboard > Security Groups

Security Groups (6) info

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count
-	sg-0b6014380451b6322	launch-wizard-1	vpc-024633ef0fa801d2	launch-wizard-1 created 2025-10-11T0...	649418801823	2 Permission entry
-	sg-09f1745ebf8e03d0f	default	vpc-024633ef0fa801d2	default VPC security group	649418801823	2 Permission entry
-	sg-08142f771b0317996a	default	vpc-0a6bf1bc3448c6d3b	default VPC security group	649418801823	1 Permission entry
bookverse-rds-sg	sg-00c130326950cf95	bookverse-rds-sg	vpc-0a6bf1bc3448c6d3b	Allow DB access from EC2 instances	649418801823	1 Permission entry
bookverse-elb-sg	sg-038117fe7640ca21f	bookverse-elb-sg	vpc-0a6bf1bc3448c6d3b	Allow HTTP from ALB and SSH from my...	649418801823	2 Permission entry
bookverse-alb-sg	sg-03a81eba68649a2	bookverse-alb-sg	vpc-0a6bf1bc3448c6d3b	Allow HTTP/HTTPS from the internet	649418801823	2 Permission entry

## CloudWatch

The screenshot shows the AWS CloudWatch Log Management interface. On the left, there's a sidebar with navigation links like Dashboards, Alarms, AI Operations, General Observability, Application Signals (APM), Infrastructure Monitoring, Logs (selected), Log Management (New), Metrics, Network Monitoring, and Setup. The main content area is titled '/aws/lambda/bookverse-log-s3-uploads' under 'Log group details'. It shows basic information such as Log class (Info, Standard), ARN (arn:aws:log:us-east-1:649418801823:log-group:/aws/lambda/bookverse-log-s3-uploads), Creation time (2 hours ago), Retention (Never expire), and Stored bytes (-). On the right, there are sections for Metric filters (0), Subscription filters (0), Contributor Insights rules (-), KMS key ID (-), and Deletion protection (Off). Below this, there are tabs for Actions, View in Logs Insights, Start tailing, and Search log group. A large section below is titled 'Log streams (3)' with a table showing three log streams: '2025/12/06/[LATEST]ff7ee6061db43fb75fc4f527cc02d' (Last event time: 2025-12-06 04:07:20 (UTC)), '2025/12/06/[LATEST]b7b6728f2a54fa3b3b30ed0930175bb5' (2025-12-06 02:53:30 (UTC)), and '2025/12/06/[LATEST]0410b6ba03e5942a5bfac389961aa9fe' (2025-12-06 02:51:36 (UTC)). There are buttons for Delete, Create log stream, and Search all log streams.

## Lambda

The screenshot shows the AWS Lambda Functions page. The left sidebar includes Lambda, Dashboard, Applications, Functions (selected), and Related AWS resources (Capacity providers, Code signing configurations, Event source mappings, Layers, Replicas). The main content area is titled 'Functions (2)' and lists two functions: 'log\_s3\_uploads' and 'bookverse-log-s3-uploads'. Both are in Zip package type, Python 3.11 runtime, and were last modified 7 hours ago. To the right, there's a 'Create function' button, an 'Info' tab (selected), a 'Tutorials' tab (with a 'Create a simple web app' section), and a 'Learn how to implement common use cases in AWS Lambda' section. A 'Create a simple web app' tutorial is shown with steps: Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage; Invoke your function through its function URL. There are 'Learn more' and 'Start tutorial' buttons.

## S3 Bucket

The screenshot shows the AWS S3 Bucket objects page for 'bookverse-uploads-649418801823'. The left sidebar includes Amazon S3, Buckets (selected), General purpose buckets (Directory buckets, Table buckets, Vector buckets), Access management and security (Access Points, Access Points for FSx, Access Grants, IAM Access Analyzer), Storage management and insights (Storage Lens, Batch Operations), Account and organization settings, and AWS Marketplace for S3. The main content area shows one object named 'test-file.txt' with a size of 26.0 B and a storage class of Standard. There are buttons for Actions (Copy S3 URI, Copy URL, Download, Open, Delete, Upload, Create folder), and a 'Find objects by prefix' search bar. A note at the top says '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.' There are also 'Actions' and 'Create folder' buttons.

## Cloud Formation

The screenshot shows the AWS CloudFormation console with the following details:

- Stacks (1)**: A list of stacks, with **BookVerse-StackWeb** selected.
- Stack info**: Overview of the stack.
  - Stack ID**: arn:aws:cloudformation:us-east-1:649418801823:stack/bookVerse-StackWeb/b009460-d245-11f0-8d32-0ef3ae46d93
  - Description**: BookVerse Application Stack - EC2 (t3.micro), ALB, RDS, S3, Lambda
  - Status**: CREATE\_COMPLETE
  - Status reason**: -
  - Parent stack**: -
  - Created time**: 2025-12-05 20:35:05 UTC-0500
  - Updated time**: -
  - Deleted time**: -
  - Drift status**: NOT\_CHECKED
  - Last drift check time**: -
  - Termination protection**: Deactivated
  - IAM role**: -
- Notifications**: 0 errors, 0 warnings, 0 informational, 5 notices.
- Actions**: Delete, Update stack, Stack actions, Create stack.

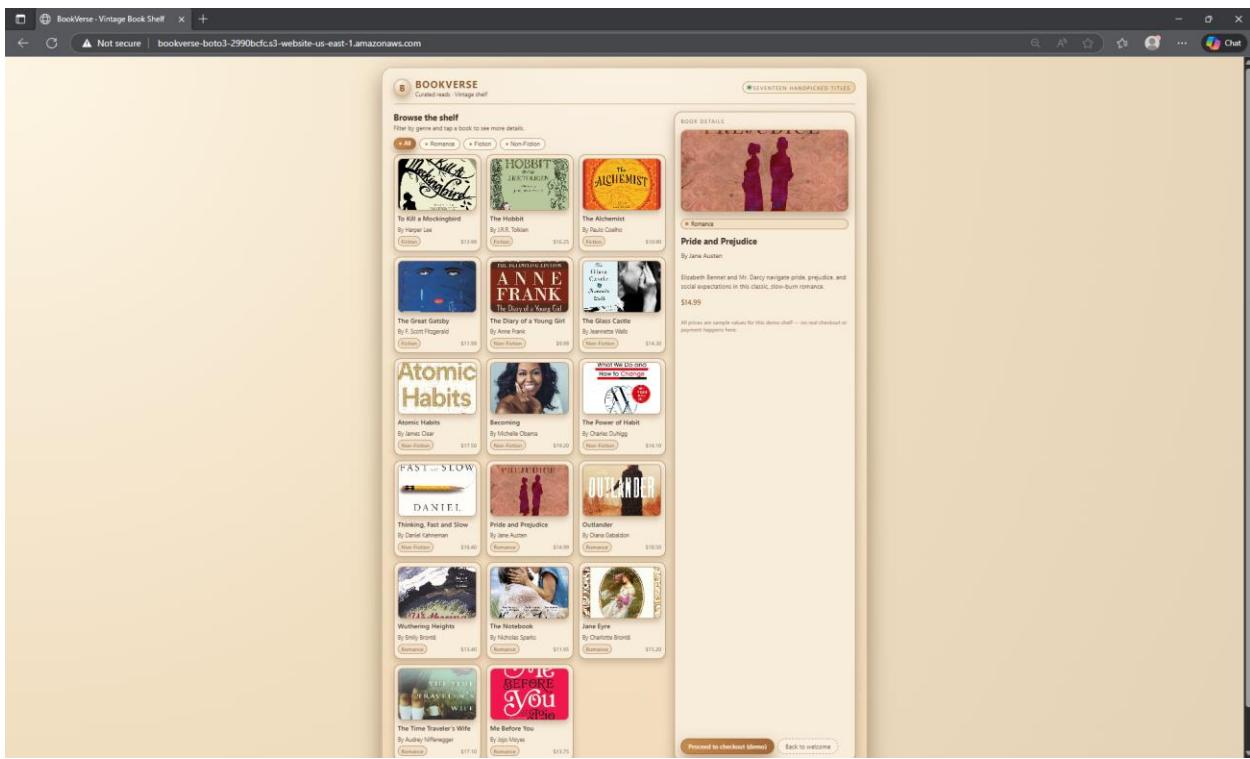
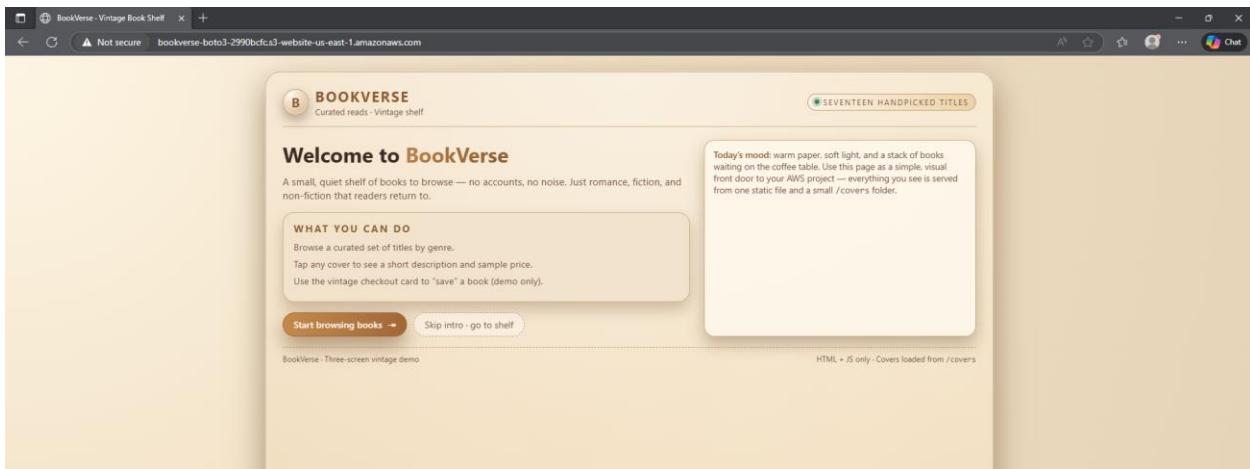
## Static Website Hosting

The screenshot shows the AWS Amazon S3 console with the following details:

- Buckets**: General purpose buckets, including **bookverse-boto3-2990bcfc**.
- bookverse-boto3-2990bcfc** (Info): Properties tab.
- Bucket overview**:
  - AWS Region: US East (N. Virginia) us-east-1
  - Amazon Resource Name (ARN): arn:aws:s3:::bookverse-boto3-2990bcfc
  - Creation date: December 6, 2025, 02:52:26 (UTC-05:00)
- 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](#)
  - Bucket Versioning: Disabled
- Multi-factor authentication (MFA) delete**: An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)
- Bucket ABAC**: Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. With ABAC, you can attach tags to your general purpose buckets and AWS Identity and Access Management (IAM) entities (users or roles).
  - ABAC status: Disabled
- Tags**: You can use tags to track storage costs, organize general purpose buckets, and specify permissions for a general purpose bucket. AWS-generated tags are created by AWS and are read-only. [Learn more](#)
  - S3 Console now uses s3>ListTagsForResource, s3:TagResource, and s3:UntagResource APIs to manage tags on S3 general purpose buckets by default. To use these APIs for tagging, please provide permissions to s3>ListTagsForResource, s3:TagResource, and s3:UntagResource actions. [Learn more](#)
- User-defined tags**: AWS-generated tags

# Home

Link: <http://bookverse-boto3-2990bcfc.s3-website-us-east-1.amazonaws.com>



## Genre Filtering

BookVerse - Vintage Book Shelf

Not secure | bookverse-boto3-2990bcfc3-website-us-east-1.amazonaws.com

BOOKVERSE  
Curated reads - Vintage shelf

Browse the shelf  
Filter by genre and tap a book to see more details.

All • Romance • Fiction • Non-Fiction

PREJUDICE  
Pride and Prejudice  
By Jane Austen  
Romance \$14.99

OUTLANDER  
Outlander  
By Diana Gabaldon  
Romance \$18.50

WUTHERING HEIGHTS  
Wuthering Heights  
By Emily Brontë  
Romance \$13.40

THE NOTEBOOK  
The Notebook  
By Nicholas Sparks  
Romance \$11.95

JANE EYRE  
Jane Eyre  
By Charlotte Brontë  
Romance \$15.20

THE TIME TRAVELER'S WIFE  
The Time Traveler's Wife  
By Audrey Niffenegger  
Romance \$17.10

ME BEFORE YOU  
Me Before You  
By Jojo Moyes  
Romance \$13.75

SEVENTEEN HANDPICKED TITLES

BOOK DETAILS

Pride and Prejudice  
By Jane Austen

Elizabeth Bennet and Mr. Darcy navigate pride, prejudice, and social expectations in this classic slow-burn romance.

\$14.99

All prices are sample values for this demo shelf — no real checkout or payment happens here.

Proceed to checkout (demo) Back to welcome

BookVerse - Three-screen vintage demo

HTML + JS only - Covers loaded from /covers

BookVerse - Vintage Book Shelf

Not secure | bookverse-boto3-2990bcfc3-website-us-east-1.amazonaws.com

BOOKVERSE  
Curated reads - Vintage shelf

Browse the shelf  
Filter by genre and tap a book to see more details.

All • Romance • Fiction • Non-Fiction

TO KILL A MOCKINGBIRD  
To Kill a Mockingbird  
By Harper Lee  
Fiction \$13.99

THE HOBBIT  
The Hobbit  
By J.R.R. Tolkien  
Fiction \$16.25

THE ALCHEMIST  
The Alchemist  
By Paulo Coelho  
Fiction \$10.90

THE GREAT GATSBY  
The Great Gatsby  
By F. Scott Fitzgerald  
Fiction \$11.99

SEVENTEEN HANDPICKED TITLES

BOOK DETAILS

To Kill a Mockingbird  
By Harper Lee

Scout Finch grows up in a small Southern town and watches her father stand up for justice in a powerful story about empathy and moral courage.

\$13.99

All prices are sample values for this demo shelf — no real checkout or payment happens here.

Proceed to checkout (demo) Back to welcome

BookVerse - Three-screen vintage demo

HTML + JS only - Covers loaded from /covers

BookVerse - Vintage Book Shelf Not secure | bookverse-bot03-2990bcfc.s3-website-us-east-1.amazonaws.com

**BOOKVERSE**  
Curated reads - Vintage shelf

Browse the shelf  
Filter by genre and tap a book to see more details.

All Romance Fiction Non-Fiction

The Diary of a Young Girl by Anne Frank \$9.99  
The Glass Castle by Jeannette Walls \$14.30  
Atomic Habits by James Clear \$17.50  
Becoming by Michelle Obama \$19.20  
What We Do and How to Change It by Charles Duhigg \$14.10  
Thinking, Fast and Slow by Daniel Kahneman \$18.40  
FAST AND SLOW by Daniel Kahneman \$14.30

SEVENTEEN HANDPICKED TITLES

BOOK DETAILS

Glass Castle by Jeannette Walls Non-Fiction

The Glass Castle by Jeannette Walls

Walls recounts her unconventional upbringing with charismatic but deeply flawed parents, and how she rebuilt her life on her own terms.

All prices are sample values for this demo shelf — no real checkout or payment happens here.

Proceed to checkout (demo) Back to welcome

BookVerse - Three-screen vintage demo HTML + JS only - Covers loaded from /covers

## Purchasing book

BookVerse - Vintage Book Shelf Not secure | bookverse-bot03-2990bcfc.s3-website-us-east-1.amazonaws.com

**BOOKVERSE**  
Curated reads - Vintage shelf

← Back to shelf

**Checkout - Pride and Prejudice**

Fill in this card to simulate saving a book to your reading list. This is a front-end only demo — no orders are stored and no real payment is processed.

FULL NAME  
Ada Lovelace

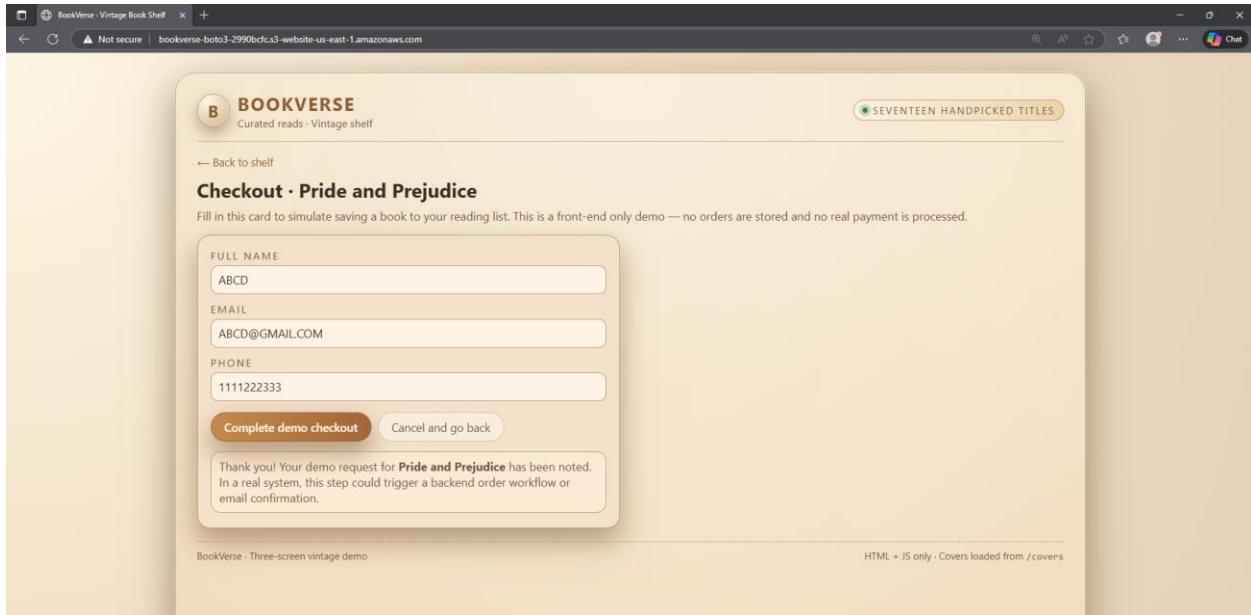
EMAIL  
you@example.com

PHONE  
+1 (555) 123-4567

Complete demo checkout Cancel and go back

SEVENTEEN HANDPICKED TITLES

BookVerse - Three-screen vintage demo HTML + JS only - Covers loaded from /covers



## BONUS

### 1. Deploying API Gateway to invoke Lambda via HTTP requests.

Successfully updated the function bookverse-api-hello.

The test event "EventTest" was successfully saved.

```

7     "headers": {
8         "Content-Type": "application/json"
9     },
10    "body": json.dumps({
11        "message": "Hello From BookVerse API!",
12        "path": event.get("rawPath", "/"),
13        "method": event.get("requestContext", {}).get("http", {}).get("method", "GET")
14    })
15}
16
17

```

DEPLOY (Ctrl+Shift+U)

Test (Ctrl+Shift+B)

TEST EVENTS (SELECTED: EVENTTEST)

- + Create new test event
- Private saved events

EventTest

ENVIRONMENT VARIABLES

Lambda Deployed

Code properties

Runtime settings

Layers

Info Tutorials

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Start tutorial

The screenshot shows the AWS API Gateway console. On the left, a sidebar menu includes sections for API Gateway, APIs, Domains, VPC links, Develop (Routes, Authorizations, Integrations, CORS, Reimport, Export), Deploy (Stages), Monitor (Metrics, Logging), and Protect (Throttling). The main area displays a green success message: "Successfully created API BookverseAPI (sknufv1aqk)". Below this, the "Routes" section for the "BookverseAPI" is shown, featuring a search bar and a list of routes. A single route, "/hello", is listed under the "GET" method. A "Create" button is available at the top right of the route list.



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The title bar also shows "Microsoft Windows [Version 10.0.26100.7171]" and "(c) Microsoft Corporation. All rights reserved.". The command entered is "curl \"https://sknufv1aqk.execute-api.us-east-1.amazonaws.com/prod/hello\"". The output of the command is displayed below the command line: {"message": "Hello from BookVerse API!", "path": "/prod/hello", "method": "GET"}

```
Microsoft Windows [Version 10.0.26100.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\prath>curl "https://sknufv1aqk.execute-api.us-east-1.amazonaws.com/prod/hello"
{"message": "Hello from BookVerse API!", "path": "/prod/hello", "method": "GET"}
C:\Users\prath>
```

## 2. STEP FUNCTION

Step Functions

Dashboard

State machines

Activities

Developer resources

Execution inspector

Online learning workshop

Local Development

Data flow simulator

Feature spotlight

Documentation

Join our feedback panel

Search [Alt+S]

Design Code Config

Undo Redo Format Copy Commands View docs

Comment: "BookVerse - simple book purchase workflow",  
StartAt: "ValidateOrder",  
States: {  
 "ValidateOrder": {  
 "Type": "Task",  
 "Resource": "arn:aws:lambda:us-east-1:123456789012:function:validate-order",  
 "Parameters": {  
 "step": "ValidateOrder",  
 "book.\$": "\$.book",  
 "user.\$": "\$.user"  
 },  
 "Next": "ProcessPayment"  
 },  
 "ProcessPayment": {  
 "Type": "Task",  
 "Resource": "arn:aws:lambda:us-east-1:123456789012:function:process-payment",  
 "Parameters": {  
 "step": "ProcessPayment",  
 "book.\$": "\$.book",  
 "user.\$": "\$.user"  
 },  
 "Next": "CompleteOrder"  
 },  
 "CompleteOrder": {  
 "Type": "Task",  
 "Resource": "arn:aws:lambda:us-east-1:123456789012:function:complete-order",  
 "Parameters": {  
 "step": "CompleteOrder",  
 "book.\$": "\$.book",  
 "user.\$": "\$.user"  
 },  
 "End": true  
 }  
}

Zoom in Zoom out Center

Feedback

Start

ValidateOrder

ProcessPayment

CompleteOrder

End

CloudShell Feedback Console Mobile App Search [Alt+S]

United States (N. Virginia) PrathyushaHarishKumar

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

The screenshot shows the AWS Lambda Function Editor interface. At the top, a green banner indicates that a test event named 'EventTest2' was successfully saved. The main area displays the Lambda function code:

```
19 result = "Unknown step"
20
21 return {
22     "statusCode": 200,
23     "body": json.dumps(
24         {
25             "step": step,
26             "result": result,
27             "book": book,
28             "user": user
29         })
30 }
```

Below the code, there are several configuration sections:

- Event**:
  - Synchronous**: Executes the Lambda function and blocks until receiving the function's response, with a maximum timeout of 15 minutes. Returns function output or error details directly to the calling application.
  - Asynchronous**: Dequeues the Lambda function for execution and returns immediately with a request ID. Functions processes independently, with results optionally sent to a configured destination like SQS, SNS or EventBridge.
- Event sharing settings**:
  - Private**: This event is only available in the Lambda Console and to the event creator. You can configure a total of ten.
  - Shareable**: This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more
- Template - optional**: `EventTest2`

The bottom section shows the execution log:

```
Function Log:
START RequestId: Ba32a2eb-077e-4eb2-8e3e-becd2196a203 Version: $LATEST
[1] Bookverse workflow step: unknown
User: N/A, Book: N/A
END RequestId: Ba32a2eb-077e-4eb2-8e3e-becd2196a203 Duration: 1.39 ms Billed Duration: 86 ms Memory Size: 128 MB Max Memory Used: 34 MB Init Duration: 83.72 ms
```

On the left sidebar, there are tabs for **Code properties**, **Runtime settings**, **Handler**, **Architecture**, and **Runtime management configuration**. The **Code properties** tab is currently selected, showing the package size (492 byte), SHA256 hash (mCeK5zze2Odpl3iuWVipnYeJdxuTEkM1flncSeHp=), and an encryption key (SHA256 hash: mCeK5zze2Odpl3iuWVipnYeJdxuTEkM1flncSeHp=).

The right sidebar includes links to **Info** and **Tutorials**, and a section titled 'Create a simple web app' with a 'Start tutorial' button.

Screenshot of the AWS Step Functions console showing the creation of a new state machine.

**Workflow Definition:**

```

graph TD
    Start((Start)) --> Lambda1[Lambda Invoke ValidateOrder]
    Lambda1 --> Lambda2[Lambda Invoke ProcessPayment]
    Lambda2 --> Lambda3[Lambda Invoke CompleteOrder]
    Lambda3 --> End((End))

```

**Workflow Properties:**

- Start at:** ValidateOrder
- Comment - optional:** BookVerse - simple book purchase workflow
- TimeoutSeconds - optional:** 600

Screenshot of the AWS Step Functions console showing the execution of a specific workflow.

**Execution Details:**

**Execution ID:** 2eb1ed36-aef-460c-bba5-4469c71b34e7

**Execution Input and Output:**

**State Input:**

```

1: {
2:   "book": "Pride and Prejudice",
3:   "user": "student@example.com"
4: }

```

**State Output:**

```

1: [
2:   {
3:     "step": "CompleteOrder",
4:     "result": "Order completed",
5:     "book": "Pride and Prejudice",
6:     "user": "student@example.com"
7:   }
8: ]

```

**Graph View:**

**Step Details:**

Choose a step to view its details.

**Event View:**

Screenshot of the AWS Step Functions console showing the execution history of a state machine named "BookVerseOrderWorkflow".

The navigation path is: AWS Console Home > Step Functions > State machines > BookVerseOrderWorkflow > Execution: 2eb1ed56-aeaf-460c-bba5-4469c71b34e7

The left sidebar includes links for Step Functions, Dashboard, State machines, Activities, Developer resources (Execution inspector, Online learning workshop, Local Development, Data flow simulator, Feature spotlight, Documentation), and Join our feedback panel.

The main area displays the "Event view" tab, showing 17 events. The table columns are: ID, Type, Step, Resource, Started After, and Timestamp.

ID	Type	Step	Resource	Started After	Timestamp
1	ExecutionStarted			0	Dec 6, 2025, 10:51:45.645 (UTC-05:00)
2	TaskStateEntered	ValidateOrder		00:00:00.032	Dec 6, 2025, 10:51:45.677 (UTC-05:00)
3	LambdaFunctionScheduled	ValidateOrder	Lambda   Log group	00:00:00.032	Dec 6, 2025, 10:51:45.677 (UTC-05:00)
4	LambdaFunctionStarted	ValidateOrder		00:00:00.117	Dec 6, 2025, 10:51:45.762 (UTC-05:00)
5	LambdaFunctionSucceeded	ValidateOrder		00:00:00.189	Dec 6, 2025, 10:51:45.834 (UTC-05:00)
6	TaskStateExited	ValidateOrder		00:00:00.212	Dec 6, 2025, 10:51:45.857 (UTC-05:00)
7	TaskStateEntered	ProcessPayment		00:00:00.212	Dec 6, 2025, 10:51:45.857 (UTC-05:00)
8	LambdaFunctionScheduled	ProcessPayment	Lambda   Log group	00:00:00.212	Dec 6, 2025, 10:51:45.857 (UTC-05:00)
9	LambdaFunctionStarted	ProcessPayment		00:00:00.299	Dec 6, 2025, 10:51:45.944 (UTC-05:00)
10	LambdaFunctionSucceeded	ProcessPayment		00:00:00.346	Dec 6, 2025, 10:51:45.991 (UTC-05:00)
11	TaskStateExited	ProcessPayment		00:00:00.368	Dec 6, 2025, 10:51:46.013 (UTC-05:00)
12	TaskStateEntered	CompleteOrder		00:00:00.368	Dec 6, 2025, 10:51:46.013 (UTC-05:00)
13	LambdaFunctionScheduled	CompleteOrder	Lambda   Log group	00:00:00.368	Dec 6, 2025, 10:51:46.013 (UTC-05:00)
14	LambdaFunctionStarted	CompleteOrder		00:00:00.445	Dec 6, 2025, 10:51:46.090 (UTC-05:00)
15	LambdaFunctionSucceeded	CompleteOrder		00:00:00.505	Dec 6, 2025, 10:51:46.150 (UTC-05:00)
16	TaskStateExited	CompleteOrder		00:00:00.530	Dec 6, 2025, 10:51:46.175 (UTC-05:00)
17	ExecutionSucceeded			00:00:00.562	Dec 6, 2025, 10:51:46.207 (UTC-05:00)

<https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1>

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