These instructions detail how to create two S3 buckets: one for your website's static assets and another for temporarily storing uploaded resumes. Security best practices are emphasized.

I. Creating the S3 Bucket for Website Static Assets:

1. **Open the S3 Console:** Log in to the AWS Management Console and navigate to the S3 service.
2. **Create Bucket:** Click "Create bucket."
3. **Bucket Name:** Enter a globally unique bucket name (e.g., yanga-portfolio-website). Use lowercase alphanumeric characters.
4. **Region:** Select an AWS region.
5. **Object Ownership:** Choose "Bucket owner enforced." This is a security best practice.
6. **Block Public Access (Optional):** For this bucket, you'll likely want to enable public access to serve the website. While the console provides a "Block all public access" setting, you should instead carefully configure a bucket policy (described in the next steps) to explicitly grant public read access *only* to the necessary objects (your website's files). This is much more secure than using the "Block all public access" setting which would prevent you from serving your static website.
7. **Create Bucket:** Click "Create bucket."
8. **Enable Static Website Hosting:** Navigate to your bucket's properties. Under the "Properties" tab, find "Static website hosting." Enable it, specifying index.html as the index document and error.html (or a suitable error page) as the error document. This configures the bucket to serve as a simple website host. Note the website endpoint URL; this will be needed for your CloudFront configuration.

II. Creating the S3 Bucket for Resume Uploads:

1. **Create Bucket:** Create a new S3 bucket specifically for storing uploaded resumes (e.g., resume-uploads).
2. **Region:** Select the same AWS region as your website bucket or one geographically closer to your expected users.
3. **Object Ownership:** Choose "Bucket owner enforced."
4. **Block Public Access: Crucially**, check "Block all public access." This prevents direct public uploads to the bucket.
5. **Server-Side Encryption:** Enable server-side encryption (SSE-S3 is simpler, but SSE-KMS is more secure if you have a KMS key set up).
6. **Create Bucket:** Click "Create bucket."
7. **Configure Bucket Policy:** This is where you restrict access to only your API Gateway:
    - Go to your bucket's "Permissions" tab.
    - Click "Bucket policy."
    - Paste a bucket policy that grants your API Gateway's execution role the permission to upload objects (PutObject) to this bucket. Replace placeholders with your actual ARN values:

```
 {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUploadsFromAPIGateway",
      "Effect": "Allow",
      "Principal": {
        "AWS":
"arn:aws:iam::YOUR_ACCOUNT_ID:role/YOUR_API_GATEWAY_EXECUTION_ROLE_ARN"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::resume-uploads/*"
    }
  ]
}
```

8.
   content_copy Use code with caution.Json

This setup ensures that only your API Gateway can upload files to the resume-uploads bucket, enhancing security. Always double-check your bucket policies before deploying your application. Remember to replace the placeholder ARNs with your actual values.