Capstone Project Runbook

Purpose

Deploy and manage a minimal production-style web app on Azure using Terraform and GitHub Actions.

1. Deployment Steps

```
Infrastructure (Terraform):
terraform init        # Initialize working directory
terraform validate    # Validate configuration
terraform apply -auto-approve  # Deploy resources

Resources:
- Resource Group, VNet (pre-existing), App Service (Linux) or Storage Account
- Tags: environment=prod, owner=shejal, project=capstone

Application (CI/CD):
- Push changes to main branch → triggers GitHub Actions workflow:
  - Build Angular/static app
  - Deploy to Azure App Service using azure/webapps-deploy@v2
- Ensure AZUREAPPSECRET publish profile is stored in GitHub Secrets.
```

2. Rollback

```
terraform destroy -auto-approve   # Remove all deployed resources
Or revert to previous commit and push to main (pipeline redeploys previous version).
```

3. Key Operations

```
Blob Storage: Use short-expiry SAS tokens for upload/download.
Key Vault: Retrieve secrets via Azure CLI:
az keyvault secret show --vault-name <vault-name> --name <secret-name>
```

4. Observability

- Log Analytics workspace linked to App Service. - Alerts: CPU > 80% or Availability test failure. - Action Group: Email notification.

5. Governance & Cost

- Tags applied during resource creation. - Monthly budget alert (or alternative cost tracking via tags).

6. Troubleshooting

- Pipeline fails: Check GitHub Actions logs; verify secrets. - App not reachable: Confirm DNS and App Service status in Azure Portal. - Terraform errors: Run terraform plan for diagnostics. - Logs: View in Log Analytics or App Service diagnostics.