praxis.md

Praxis: Github Actions

Zielsetzung

- Statische Webiste mit React
- IaC mit AWS und Terraform
- Deployment in ein S3 Bucket
- CI/CD mit Github Actions

1. Vorbereitung

1.1 AWS CLI + Profile

- Stelle sicher, dass dein aws Profil richtig konfiguriert ist und logge dich gegebenenfalls neu ein

```
aws configure list-profiles
# Mein profil heißt techstarter also:
aws sso login --profile techstarter
```

1.2 Ordnerstruktur

- Erstelle einen neuen Projektordner

```
mkdir -p cicd-demo && cd cicd-demo
git init
code .
```

- **Alle weiteren Befehle sollten im Root des neuen Projektordners ausgeführt werden**

2. Frontend: React

2.1 Create-React-App

- Initialisiere eine neue React App

```
npx create-react-app .
```

- Teste die react app:

```
npm start
```

- Committe deine Änderungen

```
git add .
git commit -m "feat: init commit"
```

2.2 Build your Website!

- In diesem Part erstellen wir unsere Anpassungen für die erste Version unserer Website

src/App.js

```
import './App.css';

function App() {
  return (
      <h1>My Techstarter Webiste</h1>
  );
}

export default App;
```

2.3 Unit Tests

- Führe die includierten Unit Tests aus:

`bash npm run test`

- Aktuell failen alle tests, fixe sie also:

src/App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/Techstarter/i);
  expect(linkElement).toBeInTheDocument();
});
```

```
npm run test
```

3. Infrastruktur: Terraform

3.1 Terraform Vorbereitungen

- Erstelle den Ordner + Files für die Infrastruktur

```
mkdir -p infra/modules/s3-website
touch infra/{versions.tf,main.tf,outputs.tf,variables.tf}
touch infra/modules/s3-website/{versions.tf,main.tf,outputs.tf,variables.tf}
```

infra/versions.tf

```
terraform {
  required_version = ">= 1.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region  = var.region
  profile = var.aws_profile
}
```

infra/variables.tf

```
variable "region" {
  type    = string
  default = "eu-central-1"
}
variable "aws_profile" {
  type    = string
  default = "techstarter"
}
```

infra/modules/s3-website/versions.tf

```
terraform {
  required_version = ">= 1.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}
```

- Füge terraform dateien zur .gitignore hinzu

.gitignore

```
# Local .terraform directories
**/.terraform/*
.terraform.*

# .tfstate files
*.tfstate
*.tfstate.*
```

3.2 S3 Website Modul

infra/modules/s3-website/main.tf

```
resource "aws_s3_bucket" "this" {
  bucket = var.bucket_name

  tags = {
    Name        = "My lovely website"
    Terraform   = "True"
  }
}

resource "aws_s3_bucket_website_configuration" "this" {
  bucket = aws_s3_bucket.this.id

  index_document {
    suffix = "index.html"
  }

  error_document {
    key = "error.html"
  }

}

resource "aws_s3_bucket_public_access_block" "this" {
  bucket = aws_s3_bucket.this.id

  block_public_acls       = false
  block_public_policy     = false
```

```
  ignore_public_acls      = false
  restrict_public_buckets = false
}
```

infra/modules/s3-website/variables.tf

```
variable "bucket_name" {
  type = string
}
```

infra/modules/s3-website/outputs.tf

```
output "bucket_arn" {
  value = aws_s3_bucket.this.arn
}
```

3.3 Einbindung im Hauptmodul

infra/main.tf

```
module "s3-website" {
  source = "./modules/s3-website"
  bucket_name = "my-website-bucket-234808sfd"
}
```

3.4 Erster Test

```
terraform -chdir=infra init
terraform -chdir=infra plan
terraform -chdir=infra apply
```

4. Github

4.1 Vorbereitungen

- Erstelle ein neues Github Repository und pushe deinen bis jetzt erstellten Code.
- Stelle sicher, dass keine Terraform state Dateien includiert sind!
- Erstelle einen neuen Order

```
mkdir -p .github/workflows
```

4.2 Github Actions

- Erstelle einen neuen Workflow

.github/workflows/cicd.yaml

```
name: GitHub Actions Demo
run-name: ${{ github.actor }} is testing out GitHub Actions
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "   The job was automatically triggered by a ${{ github.event_name }} event."
      - run: echo "   This job is now running on a ${{ runner.os }} server hosted by GitHub!"
      - name: Check out repository code
```

```
      uses: actions/checkout@v4
    - run: echo "💡 The ${{ github.repository }} repository has been cloned to the runner."
    - run: echo "    The workflow is now ready to test your code on the runner."
    - name: List files in the repository
      run: |
        ls ${{ github.workspace }}
    - run: echo "   This job's status is ${{ job.status }}."
```

- Committe und Pushe deinen Code und stelle sicher, dass die Pipeline erstellt wird

4.3 Build

- Jetzt können wir auch npm Befehle in der Pipeline ausführen

.github/workflows/cicd.yaml

```
...
    - run: echo "   This job's status is ${{ job.status }}."
    - name: Use Node.js
      uses: actions/setup-node@v3
      with:
        node-version: '20.x'
    - run: npm ci
    - run: npm run build
```

4.4 Unit Tests

.github/workflows/cicd.yaml

```
...
    - run: npm run build
    - run: npm test
```

- Committe und Pushe deinen Code und stelle sicher, dass die npm jobs richtig erstellt werden

5. Github Actions - Terraform

5.1 AWS Keys

- Leider unterstützt die Techstarter Sandbox keine Erstellung von weiteren Usern.
- **Normalerweise würde man hier einen seperaten User erstellen mit bschränkten Rechten und die Keys als Secret hinterlegen**
- Deshalb muss in diesem Schritt für dieses Github Repo er Access Key und Secret Key als secret Env variable in Github hinterlegt werden

Im Github Repo -> Settings -> Security -> Secrets & Variables -> Actions -> New Repository Secret

Kopiere das secret von dem AWS SSO Login -> Command Line or programmatic access Name: AWS_CONFIG Secret: [techstarter] aws_access_key_id=abcdefg aws_secret_access_key=12345677 aws_session_token=GAAAANZVIELTEXT

5.2 Installation der aws cli

.github/workflows/cicd.yaml

```
    - run: npm test
    - id: install-aws-cli
      uses: unfor19/install-aws-cli-action@v1
    - run: mkdir -p ~/.aws/
    - run: echo "$super_secret" > ~/.aws/credentials
      env: # Or as an environment variable
        super_secret: ${{ secrets.AWS_CONFIG }}
```

```
      - run: cat ~/.aws/credentials
      - run: aws configure list-profiles
      - run: aws s3 ls --region eu-central-1 --profile techstarter
```

5.3 Installation von Terraform

.github/workflows/cicd.yaml

```
...
      - run: aws s3 ls --region eu-central-1 --profile techstarter
      - uses: hashicorp/setup-terraform@v2
      - run: terraform version
```

5.4 Terraform Apply

- Aktuell haben wir unseren Terraform state noch Lokal gehosted, das wird zu Problemen führen!
- Lösche deshalb deine lokal erstellten TF resourcen:

```
terraform -chdir=infra destroy
```

.github/workflows/cicd.yaml

```
...
     - run: terraform -chdir=./infra init
     - run: terraform -chdir=./infra validate
     - run: terraform -chdir=./infra apply -auto-approve
```

6. Deploy

6.1 Deployment in das S3 Bucket

.github/workflows/cicd.yaml

```
...
     - run: terraform -chdir=./infra apply -auto-approve
     - run: aws s3 sync ./build s3://DEIN_BUCKET_NAME/
```