

Cloud-native Drupal on AWS

Randstad Digital

Ahmet Nuri Uygun



Inhoud

- Probleemstelling
- Opdrachtoomschrijving
- Methode van Aanpak
 - Workflow
 - Lokaal Development
 - Multi-stage Build Methode
 - Docker Compose
 - AWS
 - Netwerk- en Beveiligingslaag
 - Databaselaag
 - Applicatielaag
 - Monitoring en Onderhoud
 - Terraform
 - CI / CD Pipelines
- Resultaat (Demo)
- Testing
 - Connectivity Testing
 - Load Testing(Demo)
- Besluit

Probleemstelling



Momenteel ontbreekt een gestandaardiseerde, cloud-native AWS-infrastructuur voor onze Docker-omgevingen. Nood aan een Proof of Concept en herbruikbare blauwdruk om efficiënt Drupal- of andere applicaties op AWS uit te rollen.

Opdrachtomschrijving

- Minimale Docker-vereisten bepalen (Drupal, data & caching).
- AWS-omgeving opzetten via Infrastructure as Code (Terraform).
- Productieklare Docker-container bouwen en deployen.
- Bestaande systemen integreren binnen de AWS POC-omgeving.
- Geautomatiseerd CI/CD-proces inrichten.
- Keuzes documenteren als herbruikbare blauwdruk.
- Functionele en load-testen uitvoeren.

Workflow

Methode van Aanpak



Workflow



Drupal is a CMS framework,
written by PHP

Package the app to into Docker containers



Containerization platform
for applications

Works with Bitbucket
pipelines



Bitbucket Pipelines

Docker Image
will be
deployed in
AWS



An Infrastructure as
Code(IaC) tool

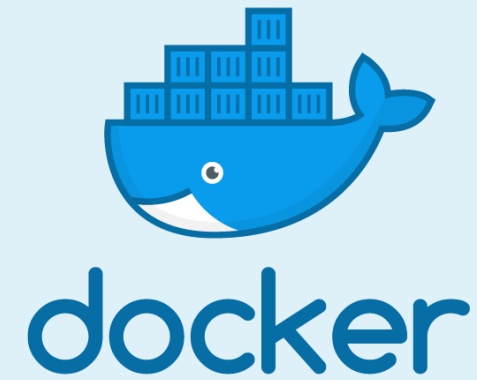
All the resources and cloud configurations will be
written with scripts



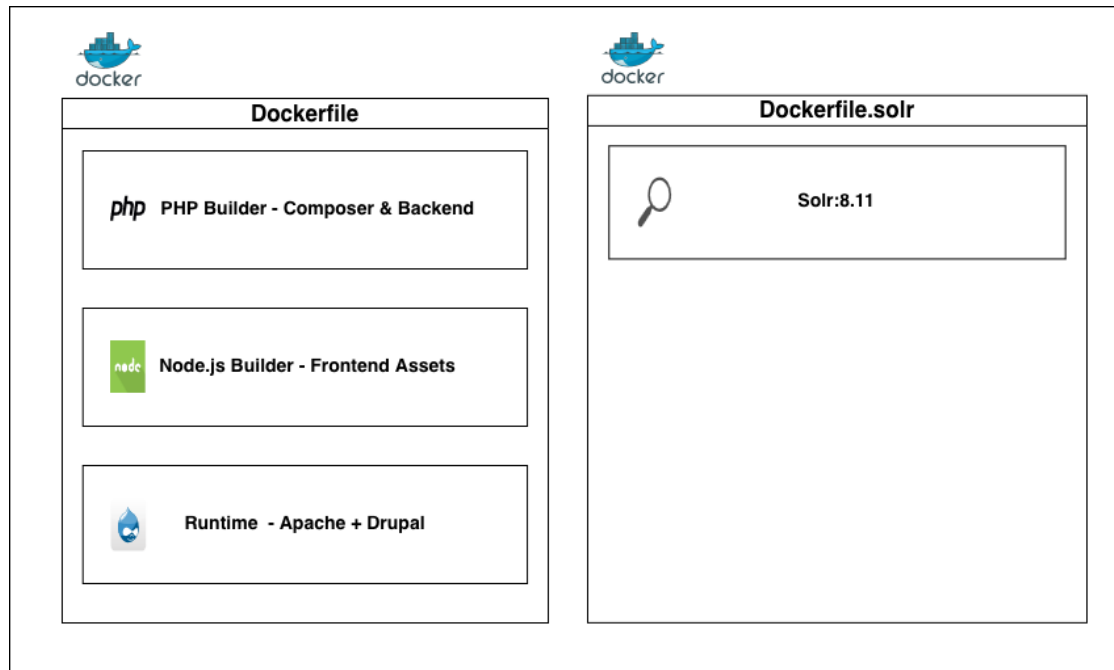
Cloud Computing Platform

Lokaal Development

Methode van Aanpak



Multi-stage Build Methode



Web Server

- Strikt scheiden van frontend, backend en runtime.
- Alleen noodzakelijke runtime-bestanden in de definitieve image.

Solr :

- Kopiëren van project-specifieke Solr-configuraties tijdens build.
- Geconfigureerd als sidecar-container binnen dezelfde ECS-task.

Docker compose

In de lokale omgeving gebruiken we docker-compose.yml om de Drupal-container te orkestreren, samen met een database en een webserver.

Hierdoor kunnen ontwikkelaars met één enkel commando snel de hele stack opstarten, zonder dat ze lokale afhankelijkheden (dependencies) hoeven te installeren.



docker-compose.yml



Web Server : Dockerfile



Database Server : mysql:8.0



Redis: redis:7-alpine



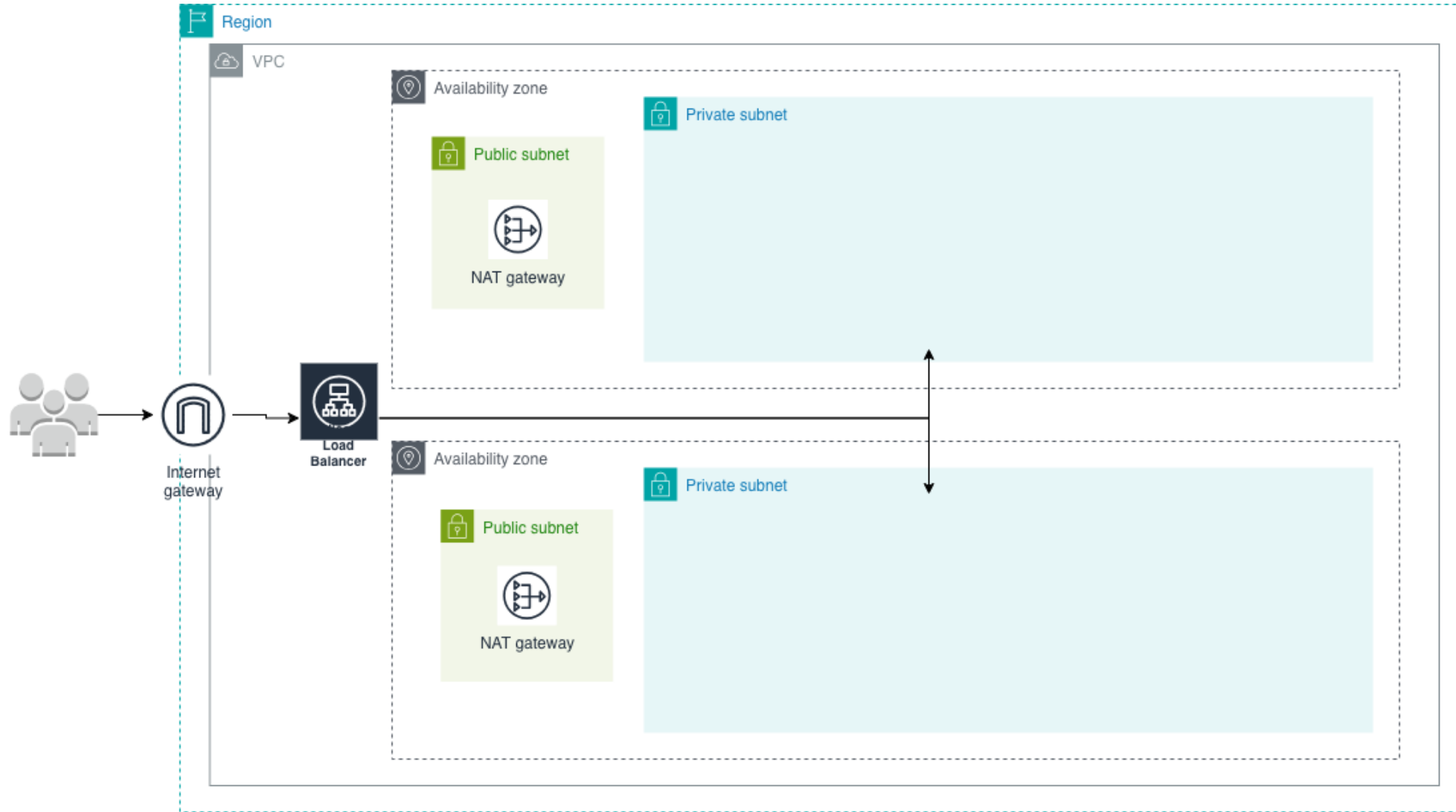
Solr : Dockerfile.solr

AWS

Methode van Aanpak

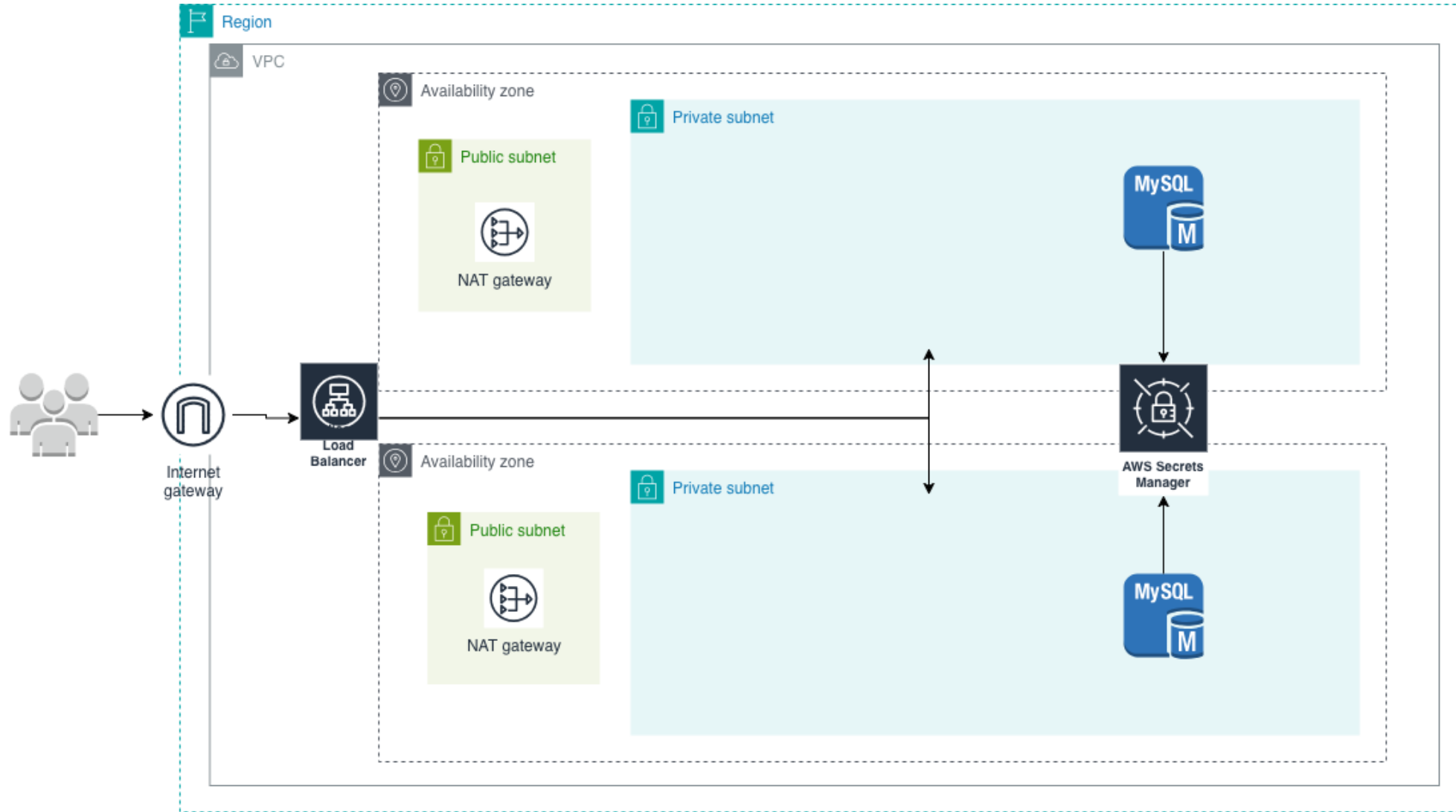


Netwerkniveau



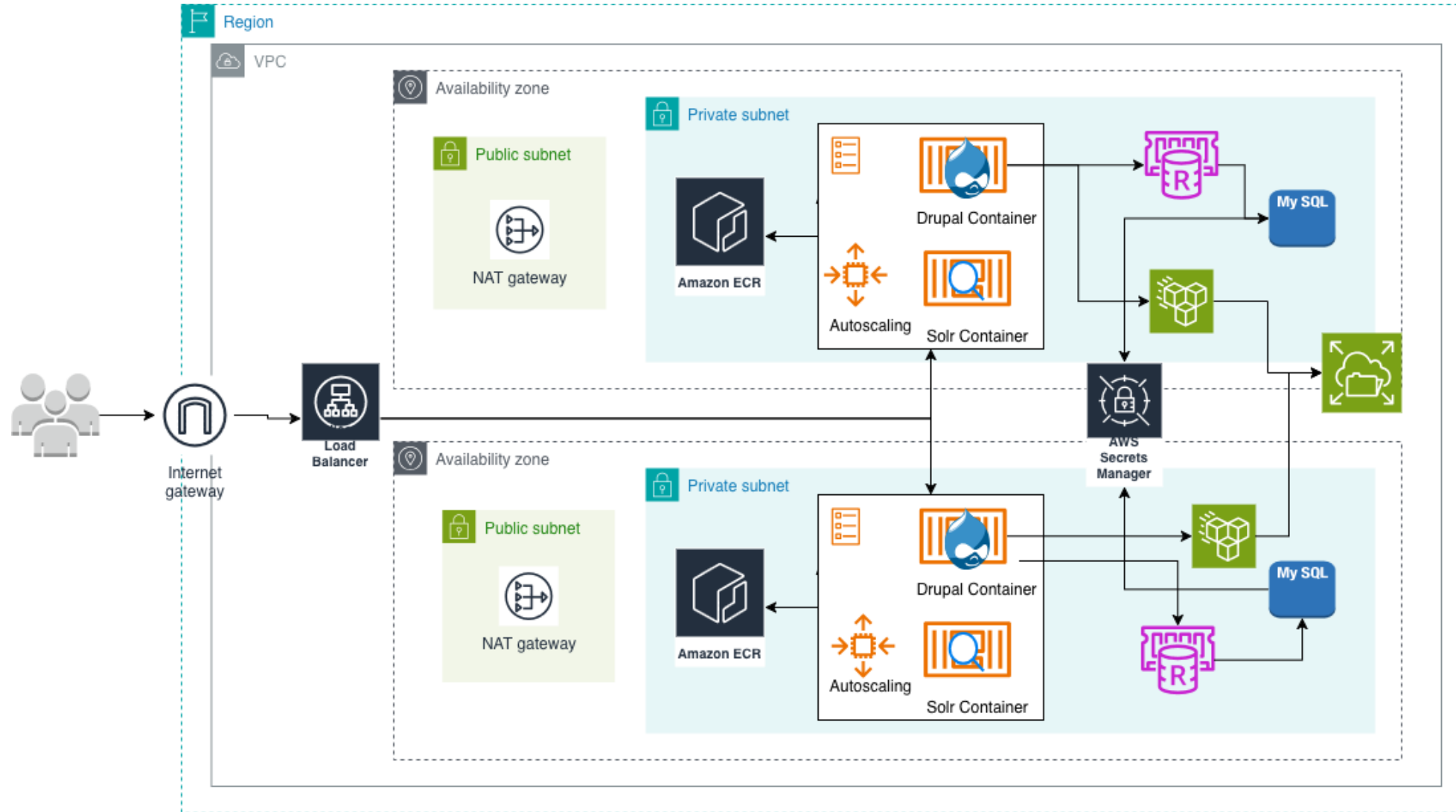
VPC, AZ, IG, NAT Gateway, Subnets, Load Balancer

Databaseniveau



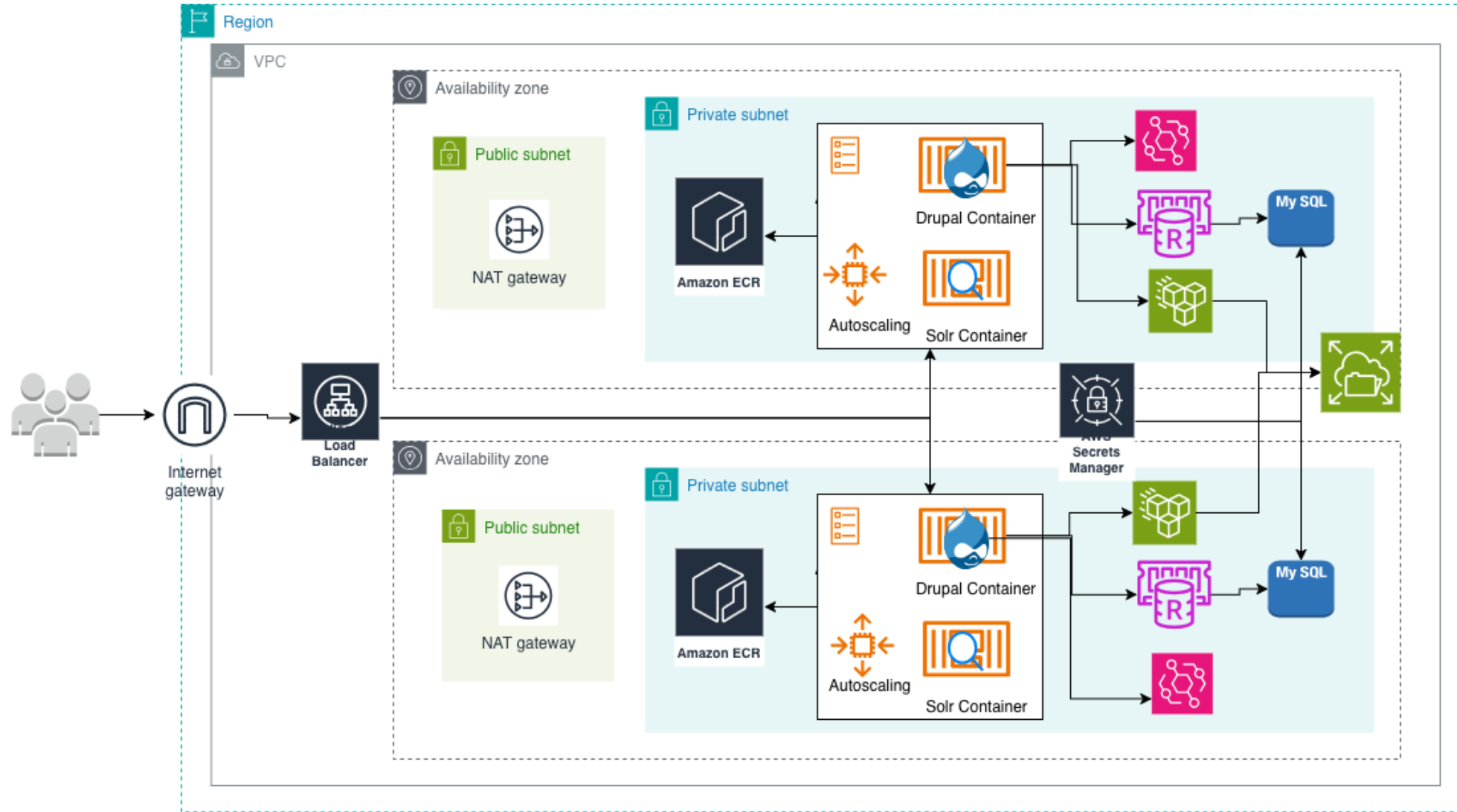
RDS, Secrets Manager

Applicationniveau



ECR, ECS, Task Definition, EFS, Redis,

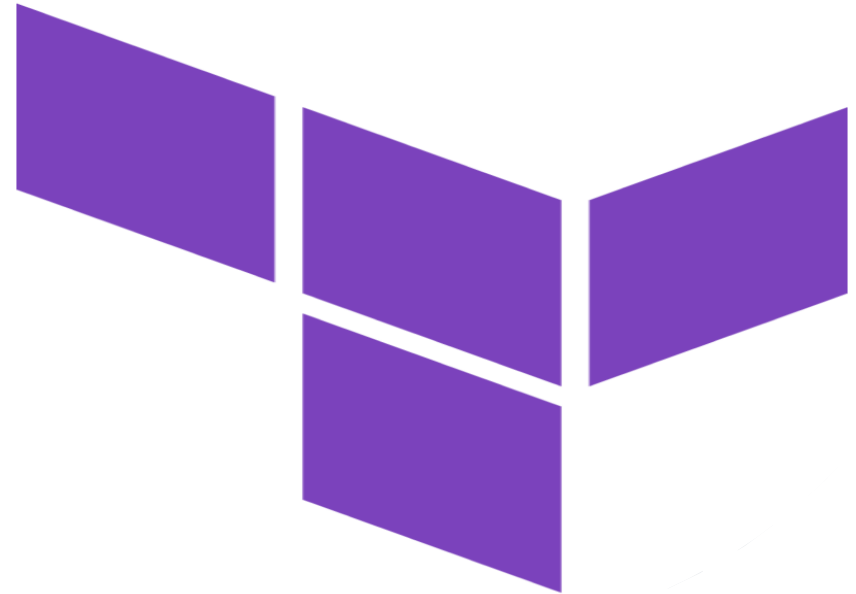
Monitoring en Onderhoud



Cloudwatch, EventBridge, SES

Terraform

Methode van Aanpak



Infrastructure as Code

- **Consistentie:** Identieke kopieën van ontwikkel-, staging- en productieomgevingen.
- **Samenwerking:** Gelijktijdig werken aan infrastructuur via Git-versiebeheer.)
- **Automatisering:** Pre-execution checks (Terraform Plan) om configuraties te valideren.

```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = ">= 5.46.0"
6     }
7   }
8
9   backend "s3" {
10    bucket     = "stage-drupal-terraform-state"
11    key        = "aws-infra/terraform.tfstate"
12    region     = "eu-north-1"
13    use_lockfile = true
14  }
15 }
16
17 provider "aws" {
18   region = var.aws_region
19
20   default_tags {
21     tags = merge(
22       {
23         Project      = var.project_name
24         Environment  = var.environment
25         ManagedBy    = "Terraform"
26         Owner        = var.owner
27       },
28       var.tags
29     )
30   }
31 }
```

Bitbucket Pipelines

Methode van Aanpak



✓ #22
 Rerun

023f13a fix: Update ECR scan_on_push formatting and add Redis and EFS port...

v1.0.0

[Learn more about reports](#)

37min 44 sec
 April 29, 2026

Pipeline

- ✓ **Snyk IaC Scan**
24s
- ✓ **Terraform Plan - Initialize and Plan ECR**
35s
- ✓ **Terraform Apply - Initialize and Apply ECR**
34s
- ✓ **Build and Push Drupal Image**
9m 38s
- ✓ **Build and Push Solr Image**
8m 7s
- ✓ **Container Scan - Drupal**
3m 44s
- ✓ **Container Scan - Solr**
3m 38s
- ✓ **Terraform Plan - Deploy Infrastructure**
36s
- ✓ **Terraform Apply - Deploy Infrastructure**
10m 23s

Build Metrics

Build setup

```
snyk auth $SNYK_TOKEN
```

```
cd terraform/production
```

```
snyk iac test --report --severity-threshold=crit
```

Build teardown

CI / CD Pipeline

Continuous Integration (CI): Ontwikkelaars pushen hun code regelmatig naar de main branch. De pipeline voert direct automatische tests en scans uit.

Continuous Delivery (CD): Na een succesvolle CI-fase wordt de code automatisch klaargezet voor deployment naar de doelomgeving

Continuous Deployment: Dit is de laatste stap waarbij gevalideerde wijzigingen zonder menselijke tussenkomst direct naar productie gaan.

Firefox File Edit View History Bookmarks Profiles Tools Window Help

19/05 (ECS Fargate Bestandsop x Team-Sylvie - ahmetnuri.uygun x Service deployments | Elastic C x Alles over kanker | Kom op tegen kanker x

aws-drupal-stage.rdbe.cloud

AusyBuilder Inhoud Structuur Uiterlijk Uitbreiding Personen Rapporten Instellingen Help

kom op tegen kanker Alles over kanker

Waar ben je naar op zoek?

Typ een trefwoord Zoeken

Meest voorkomende kankers ⓘ

Prostaatkanker Borstkanker Longkanker

Dikkedarmkanker Pancreaskanker Leukemie

Blaaskanker Nierkanker

Naar alle soorten kanker →

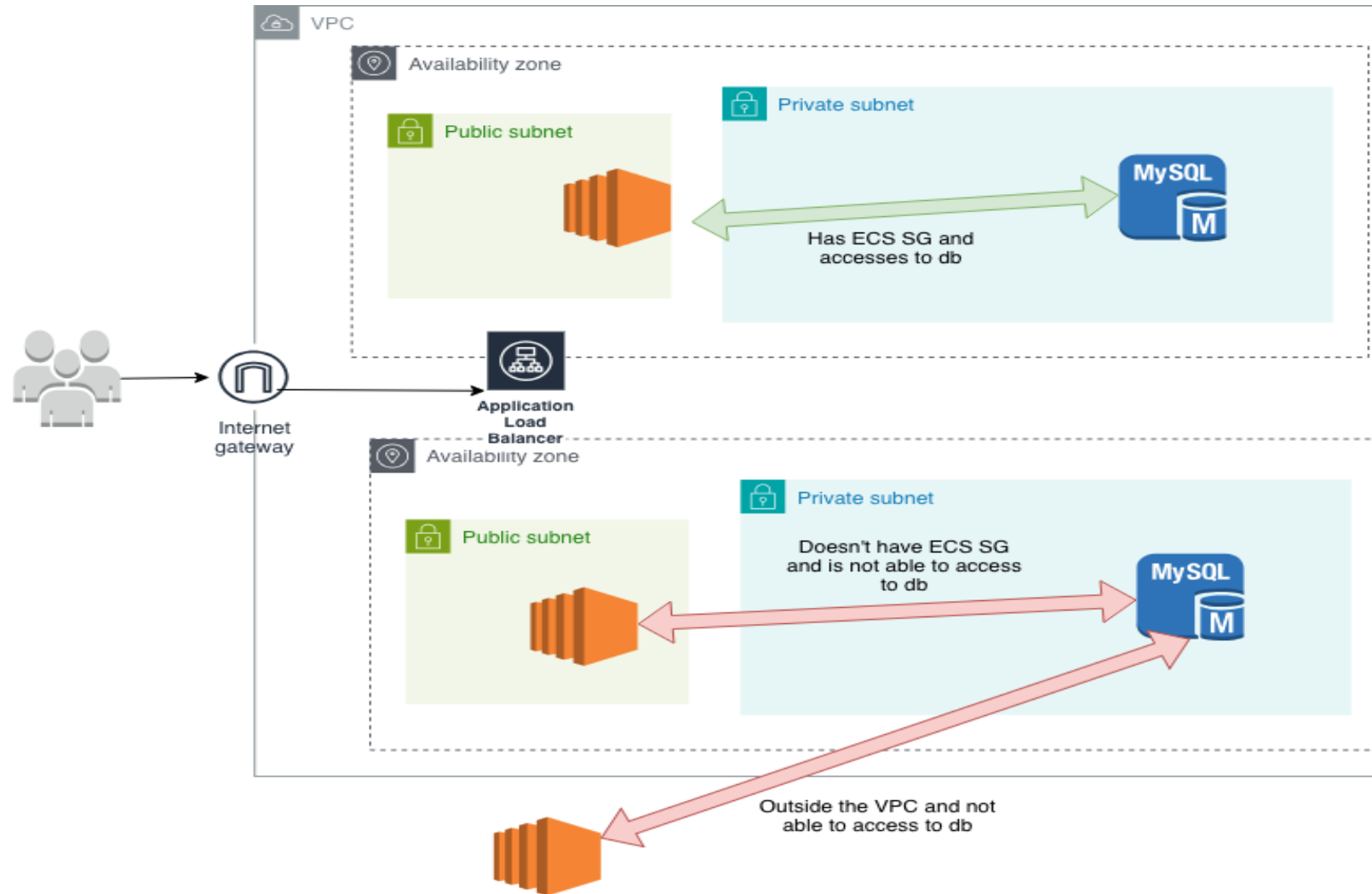
Alles over kanker

Resultaat

Connectivity Testing Testing



Beveiliging



Load Testing
Testing



Schaalberheid

Besluit

- **Succesvolle Validatie:** De KOTK-applicatie draait live en volledig geïntegreerd op AWS Fargate.
- **Onderzoeksconclusie:** Cloud-native Drupal op AWS is technisch haalbaar en lost on-premise limieten op.
- **Automatisering & Snelheid:** CI/CD en Terraform elimineren handmatige fouten en versnellen de time-to-market.
- **Schaalbaar & Veilig:** Automatische opschaling onder load en sterke netwerkisolatie via Security Groups

Bedankt!

Ahmet Nuri Uygun
r0937207@student.thomasmore.be

