

# Введение

---

## О проекте

Целью текущего проекта является развертывание приложения в отказоустойчивой масштабируемой инфраструктуре на платформе Amazon Web Services. В вашем распоряжении имеется ограниченный набор сервисов, который включается в себя:

- ECS Fargate
- ECR
- VPC
- ELB
- RDS
- CloudFront
- AWS WAFv2

Вышеуказанный перечень сервисов является достаточным для выполнения задания, однако нет необходимости использовать все сервисы одновременно. Для оптимизации проектов в организации используется CloudFormation.

## Описание приложения

Приложение представляется из себя небольшой веб-сервер (<https://nsalab-rnd-share.s3.eu-central-1.amazonaws.com/web-53.zip>). При необходимости переменные можете задать в файле `config.yml`, такие как имя таблицы, путь до кластера Redis. Для проверки работоспособности приложения можете использовать страницу `/status`.

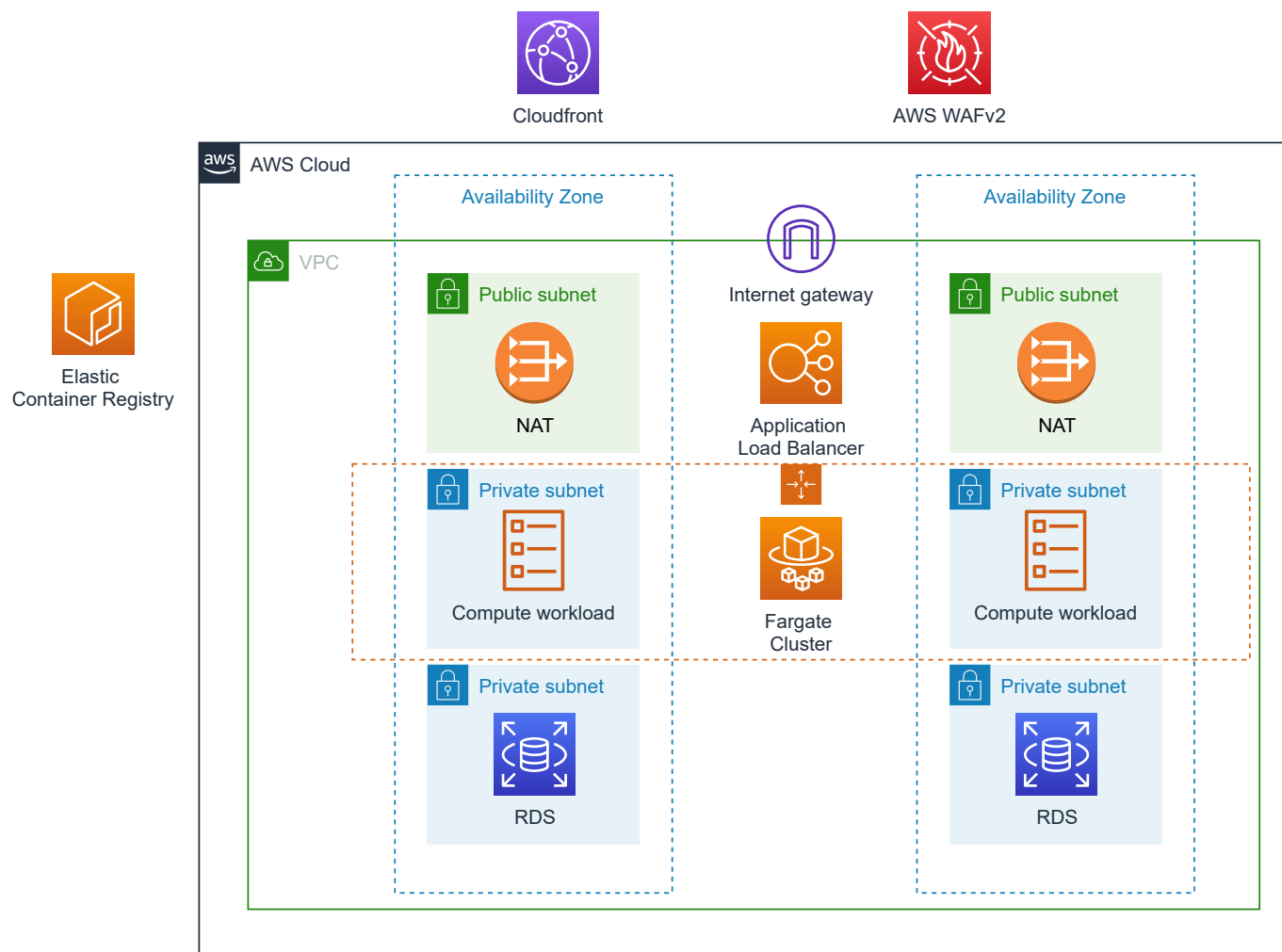
Для оценки здоровья сервера другими сервисами можете использовать страницу `/health`.

## Базовое состояние

Вам предоставлена учетная запись IAM для доступа AWS Management Console. Так же вам предоставлен доступ к Request Dashboard, в котором вам будет необходимо указать точку входа в развернутое приложение в формате `http://[IP адрес]:[порт]` или `http://[FQDN]:[порт]`. Вам необходимо будет залогиниться через Azure. Задание и учетные данные для AWS console расположены в дашборде.

## Референсная архитектура

---



## Scaling policy

Сервис в кластере Fargate должен иметь следующие параметры горизонтального масштабирования:

- Min instances: 1
- Max: 4
- Desired: 1
- Scaling Policy: target
  - Responses per instance: 20
  - Warm: 120sec
  - Cooldown: 120 sec

## Ограничения и допущения

Внимание, несоблюдение следующих условий, может помешать корректной оценке проделанной вами работы.

- При конфигурации сервиса используйте **Platform Version 1.4.0**. Это связано с особенностями реализации сбора метаданных из контейнера.
- Не используйте контейнеры более 2GB RAM, но не менее 512MB RAM.
- В конфигурации WAFv2 достаточно защитить приложение от SQL инъекций.
- Собирайте контейнер на основе **Alpine Linux v3.12**.

## База данных

База данных MySQL проходит две функциональные проверки:

- Ping до базы данных по указанным в конфге Host, Port, Username, Password.
- Попытка создать и прочитать запись в указанной в конфиге таблице. Таблица должна иметь следующую структуру (без автоинкремента):

Field	Type	Null	Key	Default	Extra
recordId	varchar(256)	YES		NULL	