

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО  
Факультет программной инженерии и компьютерной техники



Облачные и туманные вычисления

Лабораторная работа № 2

Структурная схема с обоснованием

Выполнил

Гурин Евгений Иванович

Группа № Р34122

Преподаватель: Перл Ольга Вячеславовна

г. Санкт-Петербург

2021

## Оглавление

Концепт.....	3
Компоненты системы и выбор облака .....	3
Схема компонентов.....	4
Схема базы данных .....	5
Вывод.....	16

## Концепт

Сервис организованного сбора и статистики для отчетов по автотестам с возможностью анализа итоговых данных.

Описание возможностей системы:

- сохранение данных с различных автотестов
- агрегация данных
- вывод статистики
- анализ с помощью графиков

## Компоненты системы и выбор облака

Решено было использовать облачное решение от Amazon, так как это популярное и гибкое решение и в нём я нашёл все необходимые мне возможности.

- **БД (Mongo DB)** было найдено несколько возможных решений, либо вручную поднять экземпляр виртуальной машины и на нём запустить Mongo DB, либо воспользоваться аналогом от Amazon, под названием Amazon DocumentDB, которая совместима с MongoDB

- **Веб сервис апи (сбор статистики)** я решил использовать EC2 Environment, на котором будет настроено виртуальное окружение. Гибкость этого решения позволяет обеспечить бесшовный деплой, с помощью нескольких экземпляров виртуального окружения, из которых прошлая задеплоенная версия останавливается только после запуска новой и прохождения проверок.

- Также при необходимости есть возможность настроить **scaling** (AWS Auto Scaling) и load-balancer (Amazon Lightsail) для использования нескольких экземпляров веб сервиса и автоматической работе в случае пиковых нагрузок

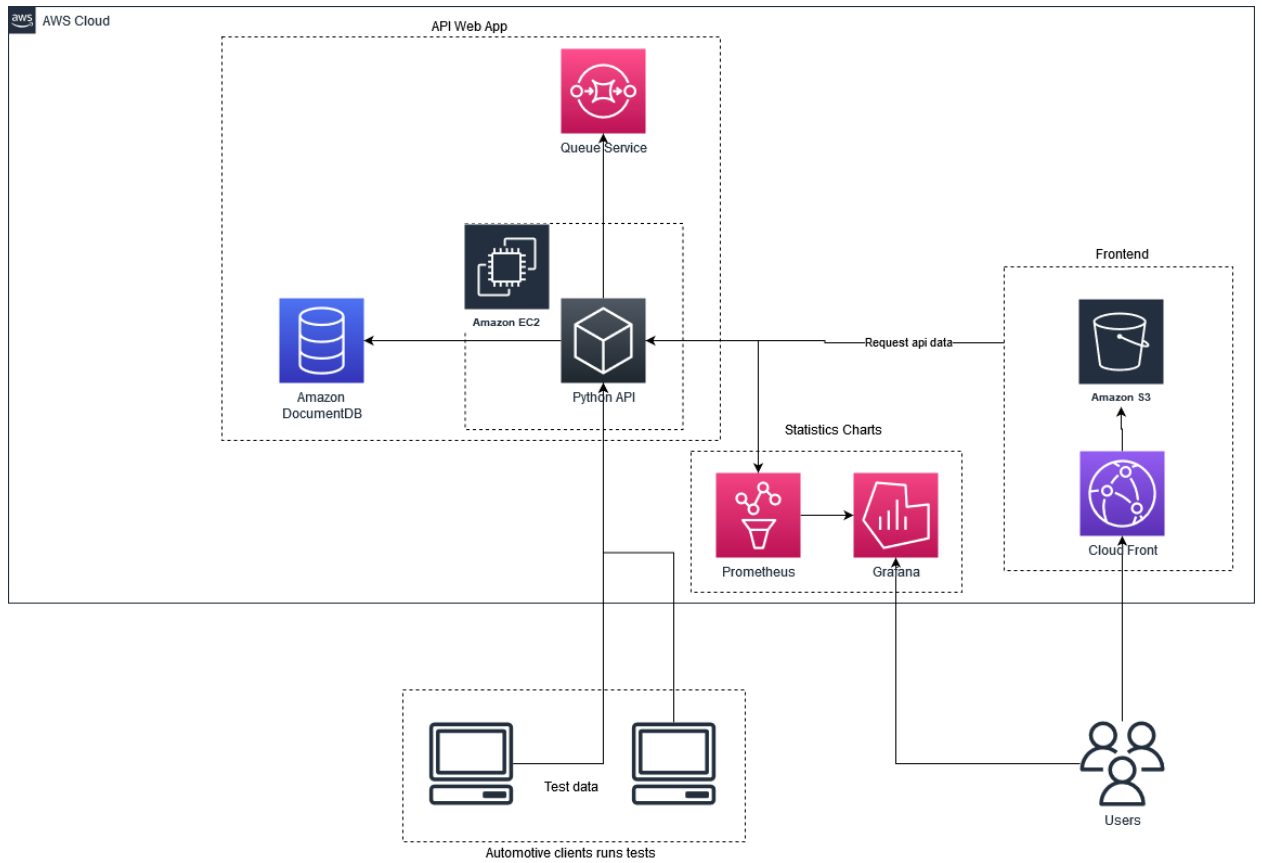
- **Prometheus** предполагается использовать Amazon Prometheus

- **Grafana** предполагается использовать Amazon Grafana

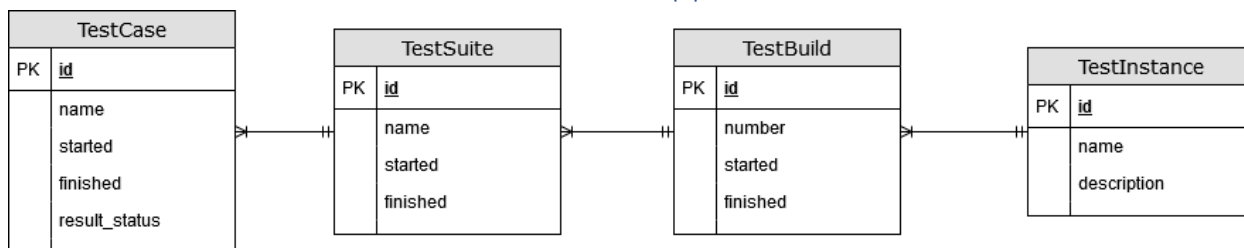
- В качестве **брокера очередей** я решил воспользоваться Amazon SQS решением. Изначально я предполагал использовать RabbitMQ, но решил что интереснее будет воспользоваться и настроить решение от Amazon, которое в том числе предоставляет тип очередей FIFO, который я и планировал использовать

- Для **фронтенда** я решил использовать S3 bucket в связке с CloudFront, который будет выступать в качестве cdn для быстрой работы

## Схема компонентов



## Схема базы данных



Представленная схема базы данных состоит из 4 таблиц

- Тестовый Инстанс – некий тестируемый продукт
- Тестовый билд – билд тестов для продукта (единица итерации)
- Тестовый сьют – набор тестовых кейсов одной группы
- Тестовый кейс – один из тестов группы

Для базы данных АПИ будет предоставлять CRUD интерфейс

## Схема апи

`/api/test-instance/`

`/api/test-instance/{instance-id}/`

`/api/test-instance/{instance-id}/test-build/`

`/api/test-instance/{instance-id}/test-build/{build-id}/`

`/api/test-instance/{instance-id}/test-build/{build-id}/start/`

`/api/test-instance/{instance-id}/test-build/{build-id}/finish/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/start/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/finish/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/test-case/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/test-case/{case-id}/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/test-case/{case-id}/start/`

`/api/test-instance/{instance-id}/test-build/{build-id}/test-suite/{suite-id}/test-case/{case-id}/finish/`

## Спецификация Open API

```
components:
  schemas:
    TestInstance:
      type: object
      properties:
        name:
          type: string
          required: true
        description:
          type: string
          required: false
    TestBuild:
      type: object
      properties:
        name:
          type: string
          required: true
        started:
          type: int
          required: false
        finished:
          type: int
          required: false
    TestSuite:
      type: object
      properties:
        name:
          type: string
          required: true
        started:
          type: int
          required: false
        finished:
          type: int
          required: false
    TestCase:
      type: object
      properties:
        name:
          type: string
          required: true
        started:
          type: int
          required: false
        finished:
          type: int
          required: false
        result_status:
          type: string
          enum:
            - SUCCESS
            - ERROR
            - FAILED

paths:
  /api/test-instance/:
    get:
```

```

summary: List all test instances
operationId: listTestInstaces
parameters:
  limit:
    name: limit
    in: query
    required: false
    schema:
      type: integer
      format: int32
  offset:
    name: offset
    in: query
    required: false
    schema:
      type: integer
      format: int32
  search:
    name: search
    in: query
    required: false
    schema:
      type: string
responses:
  '200':
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/TestInstance"
post:
summary: Create test instance
operationId: createTestInstace
requestBody:
  application/json:
    schema:
      $ref: "#/components/schemas/TestInstance"
responses:
  '200':
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/TestInstance"

api/test-instance/{test-instance-id}/:
get:
  operationId: recieveTestInstace
  parameters:
    instanceId:
      name: test-instance-id
      in: path
      required: true
      schema:
        type: string
  responses:
    '200':
      content:
        application/json:
          schema:

```

```

                                $ref: "#/components/schemas/TestInstance"

api/test-instance/{test-instance-id}/test-build/:
  get:
    operationId: listTestBuilds
    parameters:
      instanceId:
        name: test-instance-id
        in: path
        required: true
        schema:
          type: string
      limit:
        name: limit
        in: query
        required: false
        schema:
          type: integer
          format: int32
      offset:
        name: offset
        in: query
        required: false
        schema:
          type: integer
          format: int32
      search:
        name: search
        in: query
        required: false
        schema:
          type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/TestBuild"
  post:
    summary: Create test builds
    operationId: createTestBuild
    parameters:
      instanceId:
        name: test-instance-id
        in: path
        required: true
        schema:
          type: string
    requestBody:
      application/json:
        schema:
          $ref: "#/components/schemas/TestBuild"
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/TestBuild"
```



```
api/test-instance/{test-instance-id}/test-build/{test-build-id}/:
  get:
    operationId: recieveTestBuild
    parameters:
      instanceId:
        name: test-instance-id
        in: path
        required: true
        schema:
          type: string
      buildId:
        name: test-build-id
        in: path
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/TestBuild"
```

```
api/test-instance/{test-instance-id}/test-build/{test-build-id}/start/:
  post:
    summary: Start test build
    operationId: startTestBuild
    parameters:
      instanceId:
        name: test-instance-id
        in: path
        required: true
        schema:
          type: string
      buildId:
        name: test-build-id
        in: path
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/TestBuild"
```

```
api/test-instance/{test-instance-id}/test-build/{test-build-id}/stop/:
  post:
    summary: stop test build
    operationId: stopTestBuild
    parameters:
      instanceId:
        name: test-instance-id
        in: path
        required: true
```

```

        schema:
          type: string
      buildId:
        name: test-build-id
        in: path
        required: true
        schema:
          type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/TestBuild"

  api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-
suite/:
    get:
      operationId: listTestCases
      parameters:
        instanceId:
          name: test-instance-id
          in: path
          required: true
          schema:
            type: string
        buildId:
          name: test-build-id
          in: path
          required: true
          schema:
            type: string
        limit:
          name: limit
          in: query
          required: false
          schema:
            type: integer
            format: int32
        offset:
          name: offset
          in: query
          required: false
          schema:
            type: integer
            format: int32
        search:
          name: search
          in: query
          required: false
          schema:
            type: string
      responses:
        '200':
          content:
            application/json:
              schema:

```

```

                                $ref: "#/components/schemas/TestSuite"
  post:
    summary: Create test suite
    operationId: createTestSuite
    parameters:
      instanceId:
        name: test-instance-id
        in: path
        required: true
        schema:
          type: string
      buildId:
        name: test-build-id
        in: path
        required: true
        schema:
          type: string
    requestBody:
      application/json:
        schema:
          $ref: "#/components/schemas/TestSuite"
    responses:
      '200':
        content:
          application/json:
            schema:
              $ref: "#/components/schemas/TestSuite"

  api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-suite/{test-suite-id}/:
    get:
      operationId: recieveTestSuite
      parameters:
        instanceId:
          name: test-instance-id
          in: path
          required: true
          schema:
            type: string
        buildId:
          name: test-build-id
          in: path
          required: true
          schema:
            type: string
        suiteId:
          name: test-suite-id
          in: path
          required: true
          schema:
            type: string
      responses:
        '200':
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/TestSuite"

```

```
    api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-suite/{test-suite-id}/start/:
      post:
        summary: Start test suite
        operationId: startTestSuite
        parameters:
          instanceId:
            name: test-instance-id
            in: path
            required: true
            schema:
              type: string
          buildId:
            name: test-build-id
            in: path
            required: true
            schema:
              type: string
          suiteId:
            name: test-suite-id
            in: path
            required: true
            schema:
              type: string
        responses:
          '200':
            content:
              application/json:
                schema:
                  $ref: "#/components/schemas/TestSuite"

    api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-suite/{test-suite-id}/stop/:
      post:
        summary: stop test build
        operationId: stopTestCase
        parameters:
          instanceId:
            name: test-instance-id
            in: path
            required: true
            schema:
              type: string
          buildId:
            name: test-build-id
            in: path
            required: true
            schema:
              type: string
          suiteId:
            name: test-suite-id
            in: path
            required: true
            schema:
              type: string
        responses:
          '200':
```



```

    post:
      summary: Create test case
      operationId: createTestCase
      parameters:
        instanceId:
          name: test-instance-id
          in: path
          required: true
          schema:
            type: string
        buildId:
          name: test-build-id
          in: path
          required: true
          schema:
            type: string
        suiteId:
          name: test-suite-id
          in: path
          required: true
          schema:
            type: string
      requestBody:
        application/json:
          schema:
            $ref: "#/components/schemas/TestCase"
      responses:
        '200':
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/TestCase"

    api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-suite/{test-suite-id}/test-case/{test-case-id}/:
      get:
        operationId: recieveTestCase
        parameters:
          instanceId:
            name: test-instance-id
            in: path
            required: true
            schema:
              type: string
          buildId:
            name: test-build-id
            in: path
            required: true
            schema:
              type: string
          suiteId:
            name: test-suite-id
            in: path
            required: true
            schema:
              type: string
          caseId:
            name: test-case-id

```

```

        in: path
        required: true
        schema:
          type: string
      responses:
        '200':
          content:
            application/json:
              schema:
                $ref: "#/components/schemas/TestCase"

    api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-
suite/{test-suite-id}/test-case/{test-case-id}/start/:
      post:
        summary: Start test build
        operationId: startTestCase
        parameters:
          instanceId:
            name: test-instance-id
            in: path
            required: true
            schema:
              type: string
          buildId:
            name: test-build-id
            in: path
            required: true
            schema:
              type: string
          suiteId:
            name: test-suite-id
            in: path
            required: true
            schema:
              type: string
          caseId:
            name: test-case-id
            in: path
            required: true
            schema:
              type: string
        responses:
          '200':
            content:
              application/json:
                schema:
                  $ref: "#/components/schemas/TestCase"

    api/test-instance/{test-instance-id}/test-build/{test-build-id}/test-
suite/{test-suite-id}/test-case/{test-case-id}/stop/:
      post:
        summary: stop test build
        operationId: stopTestCase
        parameters:
          instanceId:
            name: test-instance-id
            in: path

```

```
      required: true
      schema:
        type: string
    buildId:
      name: test-build-id
      in: path
      required: true
      schema:
        type: string
    suiteId:
      name: test-suite-id
      in: path
      required: true
      schema:
        type: string
    caseId:
      name: test-case-id
      in: path
      required: true
      schema:
        type: string
  requestBody:
    application/json:
      schema:
        result_status:
          type: string
  responses:
    '200':
      content:
        application/json:
          schema:
            $ref: "#/components/schemas/TestCase"
```

## Вывод

В процессе выполнения данной лабораторной работы была разработана схема компонентов для проекта, учитывающая деплой на облачных технологиях Amazon Web Services, а также создана схема базы данных для представленной идеи и разработана схема api.