

AWS для разработчиков

Научитесь развертывать инфраструктуру в Amazon Web Services

Длительность курса: 108 академических часов

1 Cloud computing in theory

- 1 Введение в облачные сервисы**

что такое облако. Виды предоставляемых сервисов (виртуальные машины, контейнеры, сервисы). Обзор популярных облачных решений (AWS, GCP, Azure, IBM Bluemix) и их возможностей.

- 2 Подходы к выстраиванию инфраструктуры**

рассмотрим основные подходы к организации инфраструктуры в разрезе решаемых задач. Поговорим о жизненном цикле разработки ПО, архитектурных решениях и их влиянии на инфраструктуру. Затронем проблему “бесшовной” поставки ПО (CI / CD). Рассмотрим паттерн Infrastructure as code, и обсудим проблемы мониторинга.

3 **Docker**

обзор Docker. Docker CLI, Docker registry. Учимся создавать оптимальные images.

Домашние задания

1 Docker

Цель: Разобраться с основами docker, с образа, эко системой docker в целом.

Описание ДЗ в документе

4 **Docker практикум**

создание докер контейнера. Docker compose.

1 AWS IAM

роли, policies, виды доступа к API. Billing.

Домашние задания

1 AWS IAM

Цель: Получить первичные навыки работы с aws web console.

Получить первичные навыки работы с aws iam.

Получить первичные навыки работы с aws ec2.

Получить первичные навыки работы с aws s3.

Зайти в aws тестовый аккаунт <https://otus-test.signin.aws.amazon.com/console>,
перейти в регион Stockholm (eu-north-1),
создать себе keypair,
создать ec2 инстанс t3.micro с образа ami-1dab2163 (ubuntu 18.04) с тегом student, равный вашему имени пользователя и вашей Security group (otus-\$вашлогин*),
добавить к вашей security group доступ по ssh,
просмотреть файлы в s3 корзине otus-test,
убедиться, что доступа у вас нет,
создать aws iam политику с доступом к s3,
создать роль с установленным permissions boundary otus-students-permissions-boundary,
применить к полученной роли вашу политику,
применить полученную роль (а точнее instance profile с этой ролью),
просмотреть файлы в s3 корзине otus-test,
скачать файлы из корня корзины,
попытаться положить найденные файлы в корзину s3 в каталог \$вашлогин,
убедиться, что доступа у вас нет,
получить для своего пользователя access key,
положить найденные файлы (или любые другие) в корзину s3 в каталог \$вашлогин,
удалите ваш ec2 инстанс.

1 EC2 инстансы

типы инстансов, их возможности и конфигурирование. Работа с инстансами (включение / выключение, подключение по ssh, работа с дисками (EBS, EFS)). Развертка простого веб-сервера.

Домашние задания

- 1 Создать аккаунт на AWS. Создать инстанс на EC2, и развернуть на нем простое веб приложение.

Цель: Получить готовый шаблон запуска EC2 (EC2 Launch Template) с простейшим веб сервером.

1. Войдите в тестовый аккаунт <https://otus-test.signin.aws.amazon.com/> со своим логином.
2. Перейдите в AWS EC2 web console региона Stockholm.
3. Перейдите в раздел Launch Templates.
4. Нажмите Create launch template.
5. Укажите имя для шаблона, нажмите Show Tags и добавьте тег student со вашим логином (установите такой же тег и для Instance tags).
6. В качестве AMI ID укажите ami-1dab2163 (это ubuntu-bionic-18.04-amd64-server).
7. Для параметров launch template укажите те же настройки (Instance Type - t3.micro, Key Pair Name - ваш ключ, Security Groups - otus-\$вашлогин-*, Instance tags) что и в занятии "AWS IAM" (другие настройки лучше не меняйте).
8. Нажмите Advanced Details, установите ваш IAM instance profile (также, из занятия "AWS IAM") и в Userdata укажите строки из примера ниже (вы можете сделать тот сервер, который вам привычен, используя нужные пакеты и файлы).
9. Нажмите Create launch template.
10. Вернитесь в раздел Launch Templates.
11. Выберите ваш шаблон и нажмите Actions > Launch instance from template.
12. Выберите Source template version и нажмите Launch instance from template (другие настройки

лучше не меняйте).

13. Проверьте, правильно ли отработал ваш Userdata (например, вы можете зайти на инстанс и посмотреть файл /var/log/cloud-init-output.log или просто в браузере открыв адрес вашего инстанса по http).

14. Если всё хорошо, удалите ваш инстанс (НО НЕ ШАБЛОН).

Пример Userdata для простейшего веб сервера:

```
#!/bin/bash -xe
export DEBIAN_FRONTEND=noninteractive
apt-get update
apt-get -y upgrade
apt-get -y install apache2 php7.2 php7.2-xml php7.2-
mbstring unzip awscli
wget --output-document /tmp/aws.zip
https://docs.aws.amazon.com/aws-sdk-
php/v3/download/aws.zip
mkdir /var/www/aws-php-sdk
unzip /tmp/aws.zip -d /var/www/aws-php-sdk/
aws s3 cp s3://otus-test/s3-test.php
/var/www/html/index.php
rm -rvf /var/www/html/index.html
service apache2 restart
```

Содержание файла s3://otus-test/s3-test.php из примера Userdata

```
<?php
require '../aws-php-sdk/aws-autoloader.php';
use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;
$bucket = 'otus-test';
$keyname = 'NGC6543.jpg';
$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'eu-north-1'
]);
$s3object = $s3->getCommand('GetObject', [
    'Bucket' => $bucket,
    'Key' => $keyname
]);
$s3request = $s3-
>createPresignedRequest($s3object, '+10 minutes');
```

```
$s3signedUrl = (string) $s3request->getUri();  
echo "<img src='\".$s3signedUrl.\">";  
?>
```

2 **Load balancing. Auto Scaling groups**

обзор ELB. Виды балансировщиков.
Автоскалирование инстансов.

Домашние задания

1 Load balancing. Auto Scaling groups

Цель: 1. Создайте image (AMI) на основе ec2 инстанса с вашим приложением (из предыдущего ДЗ).
2. Создайте launch configuration для вашего инстанса
3. Создайте autoscaling group по созданному launch configuration
4. Создайте load balancer и target group
5. Свяжите target group с autoscaling group
6. Вручную созданный инстанс с приложением надо будет потушить

3 **Beanstalk. LightSail**

упрощенный деплой приложений. Обзор сервисов beanstalk и lightsail

4 **введение в Kubernetes.**

сущности. Основные понятия и архитектура.

5 S3

принцип работы хранилища и типовые кейсы использования. Интеграция с другими сервисами

Домашние задания

1 S3

Цель: 1. Создайте новый сервис, который будет выгружать на s3 bucket файл.
2. Задеплойте ваш сервис (либо в виде lambda функции, либо в виде docker контейнера)

6 serverless

концепция без серверных вычислений. Обзор lambda functions

Домашние задания

1 Домашняя работа

Цель: 1. Разбейте ваше приложение до lambda функций. Критерий, как разбивать – выбирайте сами
2. Задеплойте ваше приложение
3. Если у вас остались ресурсы на ECS / EKS – их нужно убрать

4 Storage

1 **Базы данных** ACID, BASE. Виды СУБД

2 **ECS, EKS, ECR.** обзор сервисов ECS, EKS, ECR. Отличия ECS и EKS

Домашние задания

1 Домашняя работа

Цель: 1. Удалите ваши ec2 инстансы, и всю сопутствующую конфигурацию (autoscaling group, ELB и т.д.)

2. Опишите докер файл с вашим приложением и задеплойте его на ECR

3. Разверните ваше приложение на ECS (или EKS)

3 **RDS, DynamoDB, Neptune.** обзор сервисов RDS, DynamoDB, Neptune. БД как сервис

Домашние задания

1 Домашняя работа

Цель: 1. Добавьте RDS / DynamoDB / Neptune инстанс (в зависимости от выбранной вами ранее СУБД)

2. Переведите ваше приложение на работу с созданной БД

3. Потушите вашу БД на ec2

Домашние задания

1 Домашняя работа

Цель:

Добавьте к вашему приложению возможность считать количество обращений к определенному endpoint'у. Результат счетчика – должна быть возможность посмотреть через REST API. Сам счетчик нужно сбрасывать раз в 24 часа.

1 **Очереди сообщений** обзор подходов к организации обмена сообщениями между сервисами. Обзор apache kafka, RabbitMQ, ActiveMQ

2 **SQS, ActiveMQ** понятие очереди как сервиса. Обзор сервисов SQS и ActiveMQ

Домашние задания

1 Домашняя работа

Цель: В вашем приложении, замените интерфейс взаимодействия между сервисом регистрации и сервисом отправки уведомлений, на очередь сообщений (SQS / ActiveMQ)

3 **VPC, VPN. Route 53** зачем нужна приватная сеть. Организация своей приватной сети на примере взаимодействия между контейнерами EC2. Создание своего домена и привязка к ресурсу.

- 1 **cloudwatch.**
Push / Pull log
systems.
CloudTrail зачем нужен мониторинг. Какие метрики. Обзор CloudWatch. Ревизия действий аккаунтов через cloudTrail

1 **CDN.** **CloudFront**

что из себя представляет сервис доставки контента в глобальной сети. Обзор технологии Cloudfront. Пример использования CDN для организации кеша веб-сервиса.

Домашние задания

- 1 Создать приватную сеть. Подключить VPN, чтобы можно было работать с ней удаленно.

Цель: Создать приватную сеть. Подключить VPN, чтобы можно было работать с ней удаленно. Далее расширить разработанное веб приложение таким образом, чтобы там появился статический контент (картинки, js, css). Далее настроить CDN таким образом, чтобы он кешировал данный статический контент.

- 1 Риски при проектировании (технические и экономические)**

технические риски при проектировании архитектуры разрабатываемой ИС (отказоустойчивость, масштабируемость, защищенность). Экономические риски при проектировании инфраструктуры в разрезе предложенной архитектуры (внеплановое увеличение стоимости, привязка к зависимым сервисам). Типы производств и их потребности (средняя компания, стартап и т.д.). Стадии развития компании и переоценка ценностей.

- 2 Cloudformation, TerraForm**

infrastructure as code. Обзор сервиса CloudFormation
Изучение terraform.

- 3 Планирование бюджета**

оценка бюджета и стоимости архитектуры. Подходы к сокращению бюджета и последующее влияние на архитектуру системы.

1 Итоговый проект

в качестве финального проекта будет дана бизнес задача (или согласована с преподавателем). Нужно разработать план архитектуры (физическую и логическую топологию), рассчитать примерный бюджет и риски. Далее, необходимо будет реализовать согласованную архитектуру. Задача должна включать в себя все аспекты: выбор и развертка CVS, настройка CI/CD, авто развертка приложений, авто скалирование, защита информации.

Домашние задания

1 Проект

Цель: выбрать тему проекта; закрепить тему в чат с преподавателем.