



**Universitatea Tehnică a Moldovei**

# **Automatizarea implementării arhitecturilor in sistemele cloud computing**

**Student:**

**Muntean Egor**

**Conducător:**

**Conf.Univ.**

**Victor Moraru**

**Chişinău – 2020**

# АННОТАЦИЯ

Для дипломной работы на тему: „Автоматизация развёртывания облачной инфраструктуры”

Разработанную студентом Мунтян Егор, Кишинэу 2020

Ключевые слова: облако, облачная инфраструктура, инфраструктура как код, Terraform, AWS, облачный провайдер

Цель дипломной работы изучение процесса автоматического развёртывания, облака. Разворачивается инфраструктура при помощи технологии Terraform на базе облачного провайдера Amazon Web Services. Используется аккаунт с доступами уровня free tier.

Данный проект представляет собой пример того, как разворачивают инфраструктуру DevOps специалисты в крупных компаниях.

Используемые технологии: AWS, Terraform, Bash, Amazon Linux.

Отчёт содержит: Введение, 3 главы, библиографию, 30 Рисунков, 3 Приложения.

Глава 1: Описывает концепции, плюсы и минусы облачных технологий, так-же информацию о использованных приложениях.

Глава 2: Описывает из чего состоит AWS, как автоматизировать развёртывание какие технологии при этом стоит использовать.

Глава 3: Описывает ход работы, методы развёртывания инфраструктуры демонстрирует на практике процесс автоматизации развёртывания

# ADNOTARE

Pentru teza pe tema: „Automatizarea implementării arhitecturilor în sistemele cloud computing”

Proiectat de studentul Muntian Egor, Chișinău 2020

Cuvinte cheie: cloud, infrastructură cloud, infrastructură ca cod, Terraform, AWS, furnizor de cloud

Scopul tezei este de a studia procesul de implementare automată, cloud. Infrastructura este implementată utilizând tehnologia Terraform bazată pe furnizorul de cloud Amazon Web Services. Este utilizat un cont cu acces gratuit la niveluri.

Acest proiect este un exemplu al modului în care specialiștii din companiile mari implementează infrastructura DevOps.

Tehnologii utilizate: AWS, Terraform, Bash, Amazon Linux.

Raportul conține: Introducere, 3 capitole, bibliografie, 30 de figuri, 3 anexe.

Capitolul 1: Descrie conceptele, avantajele și dezavantajele tehnologiilor cloud, precum și informații despre aplicațiile utilizate.

Capitolul 2: Descrie din ce este format AWS, cum se automatizează implementarea și ce tehnologii merită folosite.

Capitolul 3: Descrie progresul muncii, metodele de implementare a infrastructurii, demonstrează în practică procesul de automatizare a implementării

# ANNOTATION

For the thesis on the topic: "Cloud computing architecture deployment automatisaton"

Designed by student Muntean Egor, Chisinau 2020

Keywords: cloud, cloud infrastructure, infrastructure as code, Terraform, AWS, cloud provider

The purpose of the thesis is to study the process of automatic deployment, the cloud. The infrastructure is being deployed using Terraform technology based on the cloud provider Amazon Web Services. An account with free tier access is used.

This project is an example of how specialists in large companies deploy DevOps infrastructure.

Technologies used: AWS, Terraform, Bash, Amazon Linux.

The report contains: Introduction, 3 chapters, bibliography, 30 Figures, 3 Appendices.

Chapter 1: Describes the concepts, pros and cons of cloud technologies, as well as information about the applications used.

Chapter 2: Describes what AWS is made up of, how to automate deployment, and which technologies to use.

Chapter 3: Describes the course of work, methods of deploying infrastructure, demonstrates in practice the process of automation of deployment

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1.АНАЛИЗ СИТУАЦИИ В ОБЛАСТИ.....	10
1.1 Типы облачных вычислений.....	10
1.2 Типы облачных служб.....	11
1.3 Использование облачных вычислений.....	12
1.4 Преимущества облачных технологий.....	13
1.5 Минусы облачных технологий.....	14
1.6 Свойства облачных вычислений.....	15
1.7 Безопасность.....	17
1.8 Провайдеры облачных услуг .....	17
1.9 Технологии для развёртывания.....	24
2.АВТОМАТИЗАЦИЯ РАЗВЁРТЫВАНИЯ.....	28
2.1 Сервисы AWS .....	28
2.2 Использование Terraform.....	35
2.3 Среда разработки.....	40
3. СОЗДАНИЕ ИНФРАСТРУКТУРЫ.....	42
3.1 Подготовка лабораторной среды.....	42
3.2 Планирование задач.....	42
3.3 Поэтапная реализация.....	42
ВЫВОДЫ.....	54
БИБЛИОГРАФИЯ.....	55
ПРИЛОЖЕНИЕ.....	56

## Введение

Мы являемся современники революций в технологическом плане, каждый год появляются новые технологии которые двигаю мир вперёд. Технологический прогресс во многом обусловлен бешеным темпом развития ИТ- индустрии, где каждый день появляется что новое в технологическом плане, а также в коцепутальном, меняются методы разработки, подходы к хранению и криптации данных и т.д.

Ещё лет 15 назад все разработки, базы данных и веб приложения хранились на локальных серверах и редко, когда брали в аренду какой либо сервер в дата-центре. Сейчас-же очень мало крупных компаний которые держат свои приложения на локальных серверах. Мы пришли к тому, что это просто стало не выгодно с точки зрения обслуживания и обновления серверов. Да и зачем это нужно, когда есть возможность расширить своё дисковое хранилище одним кликом или двумя командами в терминале, вместо того, чтобы заказывать или ехать куда-то за новыми HDD или SSD. Теперь все наши ресурсы мы можем видеть лишь в виртуальном пространстве, либо на фотографиях из дата-центров. Но прогресс толкает нас вперёд и лет 10 назад и люди, работающие с облачными вычислениями захотели ещё большей автоматизации процессов развёртывания серверов и появились технологии предоставляющие им эту возможность, такие как Terraform, Cloud Foundation, Ansible и многие другие.

Так-же с появлением новых технологий стала появляться необходимость в людях, которые смогут с ними работать. Системный администратору как профессии пришлось двигаться вперёд и уже где залезать на поле программиста. Так-как разворачивания инфраструктуры стало уже не выполнением ряда команд или кликаньем по необходимым ресурсам, а стало полноценным кодом и тут мы знакомимся с понятием `infrastructure-as-code` и появлние новой ИТ дисциплины и коцепции DevOps.

Мы вступили в эру когда облачные инфруструктуры декларируется в коде и создание таких скриптов стало обыденным занятием для определённых специалистов. В данной работе я опишу процесс создания такого кода и разберу некоторые концепции `infrastructure as code`.

## **Выводы**

В ходе данной дипломной работы я продолжил концепцию предыдущей работы, по познанию процессов осуществляемых в облачной экосистеме и по взаимодействию с ними. Если тогда это всё происходило локально, то теперь вся инфраструктура находится где в датацентре в Франкфурте.

Данная работа сфокусирована на концепции `infrastructure-as-code`, я сделал упор на демонстрацию автоматизации процессов по созданию облачных инфраструктур. Хотя я и выбрал как провайдер Amazon Web Services, но данная концепция распространяется и на другие облачные решения, грубо говоря поднять тоже самое на условном Microsoft Azure не является проблемой, так-как Terraform умеет работать с ним так-же.

В принципе Terraform и есть причина по которой я и выбрал данную тему, так-как развиваюсь в дисциплине DevOps основой концепция которого состоит в автоматизации всех процессов которые возможно.

Благодаря данному проекту я ближе познакомился с AWS, открыл для себя новые понятия и принципы работы некоторых сервисов. Весь проект построен на Free Tier сервисах и в принципе любой человек может запустить данный проект и получить тот-же результат, что так-же следует идеологии DevOps о одинаковом результате при исполнении какого либо развёртывания на разных устройствах и тд.

В целом я доволен своей работой и буду продолжать двигаться в этом же русле.

## Библиография

1. Облачные вычисления IBM - IBM cloud computing Облачные вычисления IBM. Доступно по: [https://ru.qaz.wiki/wiki/Cloud\\_computing](https://ru.qaz.wiki/wiki/Cloud_computing). Заходил: 06.10.2020
2. Типы облачных вычислений. Доступно по: <https://aws.amazon.com/ru/types-of-cloud-computing/>. Заходил: 07.10.2020
3. Преимущества и недостатки облачных технологий. Доступно по: <https://sistyle.ru/blog/item/30-clouds-advantages/>. Заходил 07.10.2020
4. Плюсы и минусы облачных вычислений. Доступно по: <https://oncloud.ru/blog/2019/03/19/plyusy-i-minusy-oblachnyh-vychisleniy>. Заходил 08.10.2020.
5. Безопасность облачных ресурсов. Доступно по: <https://www.pwc.ru/ru/services/technology/cyber-security/bezopasnost-oblachnykh-resursov.html>. Заходил 08.10.2020
6. Top 10 Cloud Service Providers In 2020. Доступно по: <https://www.c-sharpcorner.com/article/top-10-cloud-service-providers/>. Заходил 09.10.2020
7. Что такое Azure? Доступно по: <https://azure.microsoft.com/ru-ru/overview/what-is-azure/>. Заходил 10.10.2020
8. Amazon Web Services. Доступно по: [https://ru.wikipedia.org/wiki/Amazon\\_Web\\_Services](https://ru.wikipedia.org/wiki/Amazon_Web_Services). Заходил 11.10.2020
9. Google Cloud Platform. Доступно по: [https://ru.wikipedia.org/wiki/Google\\_Cloud\\_Platform](https://ru.wikipedia.org/wiki/Google_Cloud_Platform). Заходил 11.10.2020
10. IBM cloud computing. Доступно по: [https://en.wikipedia.org/wiki/IBM\\_cloud\\_computing](https://en.wikipedia.org/wiki/IBM_cloud_computing). Заходил 11.10.2020
11. Alibaba Cloud. Доступно по: [https://en.wikipedia.org/wiki/Alibaba\\_Cloud](https://en.wikipedia.org/wiki/Alibaba_Cloud). Заходил 11.10.2020
12. Oracle Cloud. Доступно по: [https://ru.wikipedia.org/wiki/Oracle\\_Cloud](https://ru.wikipedia.org/wiki/Oracle_Cloud). Заходил 11.10.2020
13. Terraform: новый подход к Infrastructure as code. Доступно по: <https://habr.com/ru/company/piter/blog/351878/>. Заходил 13.10.2020
14. AWS CloudFormation: Concepts, Templates, and Use Case. Доступно по: <https://www.simplilearn.com/tutorials/aws-tutorial/aws-cloudformation>. Заходил 14.10.2020
15. Ansible Documentation. Доступно по: <https://docs.ansible.com/ansible/latest/index.html>. Заходил 15.10.2020
16. ХОШТЕЙН Л., МОЗЕР Р. Запускаем Ansible. Washington D.C., 2019.
17. MIKE RYAN, FEDERICO LUCIFREDI. AWS System Administration Best Practices for Sysadmins in the Amazon Cloud. New Jersey 2019.
18. NATHANIEL FELSEN. Effective DevOps with AWS. Chicago 2019