



Общество с ограниченной ответственностью «Ребреин»

ИНН 7727409582, ОГРН 1197746106161

Адрес: 123056, город Москва, Большая Грузинская ул, д. 36а стр. 5а, офис 13

Утверждено

Приказом № ПР-1 от 17.06.2025 г.

Генеральный директор

 Фролкина Е.А.
«17» июня 2025 г.

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
– ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«ИНФРАСТРУКТУРА КАК КОД: АВТОМАТИЗАЦИЯ С TERRAFORM»

Срок реализации: 1 месяц

Количество часов: 48 акад. ч.

Форма обучения: заочная форма

Формат обучения: с применением
исключительно дистанционных технологий

Возраст обучающихся: для лиц старше 17
лет, имеющих или получающих среднее
профессиональное и (или) высшее
образование

Москва, 2025 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая дополнительная профессиональная программа – программа повышения квалификации «Инфраструктура как код: автоматизация с Terraform» (далее – Программа) разработана в соответствии с:

- Федеральным законом от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- Постановлением Правительства РФ от 11.10.2023 № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Профессиональным стандартом 06.026 «Системный администратор информационно-коммуникационных систем», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 26.10.2020 года № 60580;
- ФГОС высшего образования – бакалавриат по направлению 09.03.02 Информационные системы и технологии, утв. приказом Минобрнауки России от 19.09.2017 №926;
- Локальными нормативными актами ООО «Ребреин».

В данной программе учтены основные идеи формирования универсальных учебных действий учащихся и соблюдена преемственность с программами высшего и/или среднего профессионального образования.

Направленность программы: Программа имеет техническую направленность.

Адресат:

- **Сетевые инженеры и системные администраторы**, которые хотят автоматизировать управление инфраструктурой и уйти от ручных операций.
- **Разработчики, технические и релизные менеджеры**, стремящиеся лучше понимать процессы развертывания и сопровождения приложений.
- **Специалисты по тестированию (QA)**, которым важно уметь быстро поднимать и воспроизводить тестовые окружения.
- **SRE-инженеры и DevOps-инженеры**, использующие практики IaC для обеспечения стабильности и масштабируемости систем.

Требования к входным знаниям обучающегося:

Для успешного освоения программы слушателям рекомендуется:

- базовое знание Linux и командной строки Bash;
- понимание принципов работы сетевых сервисов и клиент-серверных приложений;
- опыт работы с системами контроля версий (например, Git);
- общее представление о виртуализации или облачных сервисах (будет плюсом, но не обязательно).

Актуальность реализации:

Современная IT-инфраструктура всё чаще строится по принципу Infrastructure as Code (IaC), где управление серверами, сетями и облачными ресурсами происходит через код, а не вручную. Такой подход позволяет:

- снизить количество ошибок, связанных с человеческим фактором;
- ускорить развертывание и обновление инфраструктуры;
- обеспечить воспроизводимость и прозрачность процессов;

- повысить надежность и отказоустойчивость систем.

Terraform — один из самых популярных инструментов в этой области. Его используют компании любого масштаба — от стартапов до корпораций и государственных структур. Понимание принципов работы Terraform становится обязательным навыком для системных администраторов, DevOps-инженеров и разработчиков.

Отличительные особенности программы:

- Программа сочетает теорию и практику: слушатели осваивают базовые концепции IaC и тут же закрепляют знания на практике.
- Рассматриваются не только базовые команды Terraform, но и **углубленные возможности**: работа с переменными, циклами, условиями, backend-хранилищами и модулями.
- Включены современные инструменты-обёртки (Terragrunt, Terraspace), которые позволяют упростить работу в больших проектах.
- Программа завершается обзором альтернативных решений, что помогает сформировать у слушателей широкое понимание инструментов IaC и выбрать оптимальный стек под задачи своей организации.

Объем и срок освоения программы: 48 академ. ч. в течение 1 мес.

Доступ к материалам Программы у обучающихся остаётся и после окончания периода обучения. Это позволяет повторять изученный материал в удобное время, восполнять пробелы в знаниях, а также возвращаться к практическим заданиям при решении рабочих задач. Такой формат способствует более глубокому закреплению навыков и поддерживает профессиональное развитие выпускников даже после завершения обучения.

Выдаваемый документ о квалификации: удостоверение о повышении квалификации и/или сертификат об успешном освоении программы.

Цели и задачи программы:

Сформировать у слушателей системное понимание концепции «Инфраструктура как код» и навыки практического использования Terraform и сопутствующих инструментов для автоматизации развертывания, настройки и сопровождения ИТ-инфраструктуры.

Программа направлена на решение следующих основных задач:

Обучающие:

- дать представление о принципах IaC и роли Terraform в современном DevOps-процессе;
- обучить установке, настройке и работе с базовыми и продвинутыми возможностями Terraform;
- освоить использование переменных, модулей, провайдеров, backends и обёрток (Terragrunt, Terraspace);
- научить применять Terraform для автоматизации развертывания облачной и локальной инфраструктуры.

Развивающие:

- сформировать навыки анализа и оптимизации рабочего процесса через IaC;
- развить умение работать с инфраструктурой в команде, используя best-practices Git и Terraform workflow;
- укрепить способность к решению нетривиальных инженерных задач за счёт использования динамических блоков, условий, циклов и модульного подхода.

Воспитательные:

- формировать культуру ответственного отношения к изменениям в инфраструктуре и понимание последствий автоматизации;
- воспитывать аккуратность в работе с кодом и данными состояния (state), как основой стабильности систем;
- развивать уважение к командной работе, обмену опытом и использованию общедоступных практик и модулей (Terraform Registry).

Планируемые результаты:

Знания:

- основные принципы подхода Infrastructure as Code и место Terraform в DevOps-экосистеме;
 - структуру и синтаксис конфигурационных файлов Terraform;
 - работу с провайдерами, ресурсами, переменными и модулями;
 - возможности обёрток Terragrunt и Terraspace, а также альтернативные инструменты IaC;
 - механизмы хранения состояния (state) и их влияние на стабильность инфраструктуры.

Умения:

- устанавливать и настраивать Terraform, организовывать рабочее окружение;
- использовать базовые команды init, plan, apply, destroy, а также команды валидации конфигураций;
- работать с циклами, условиями и динамическими блоками;
- применять backend-хранилища для state-файлов, обеспечивая совместную работу в команде;
- строить модульную инфраструктуру и использовать готовые решения из Terraform Registry.

Навыки:

- автоматизация развертывания и управления ресурсами в облачных и локальных средах с помощью Terraform;
- организация командной работы с инфраструктурным кодом через Git и Terraform workflow;
- использование Terragrunt и Terraspace для упрощения и стандартизации IaC-подходов;
- интеграция IaC в процессы CI/CD и повышение отказоустойчивости инфраструктуры;
- безопасное и ответственное сопровождение изменений в инфраструктуре.

Перечень профессиональных компетенций, на получение которых направлено обучение:

На основе профстандарта 06.026 «Системный администратор информационно-коммуникационных систем»:

- В/02.5 Обеспечение работы технических и программных средств информационно-коммуникационных систем;
- С/05.6 Выполнение обновления программного обеспечения сетевых устройств информационно-коммуникационных систем;
- С/08.6 Планирование и проведение работ по распределению нагрузки между имеющимися ресурсами, снятию нагрузки на сетевые устройства информационно-коммуникационных систем перед проведением регламентных работ, восстановлению штатной схемы работы в случае сбоев.

Таким образом, в результате освоения программы у обучающихся формируются следующие профессиональные компетенции:

- ОПК-5. Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;
- ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;
- ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.

Организационно-педагогические условия реализации программы дополнительного профессионального образования

Язык реализации образовательной программы: обучение проводится на русском языке.

Форма обучения: заочная форма.

Особенности реализации программы: программа реализуется с использованием электронного обучения и исключительно дистанционных образовательных технологий.

Условия набора: на обучение принимаются все желающие лица, оплатившие обучение и заключившие договор об образовании. Обучение проходит в индивидуальном формате без формирования учебных групп. Обучающийся самостоятельно определяет время освоения Программы.

Формы проведения занятий:

- занятия в текстовом формате;
- практическая работа;
- самостоятельная работа с литературой;
- индивидуальные вопросы.

Материально-техническое оснащение

Материальное обеспечение программы

Занятия проводятся в системе дистанционного обучения «Rebrain». Каждый обучающийся и педагог оснащены доступом к системе дистанционного обучения: <https://rebrainme.com/>. У педагога дополнительного профессионального образования имеется необходимое оборудование средства для реализации программы: ноутбук с подключением к интернету, программное обеспечение.

Методическое обеспечение программы

Программа обеспечена:

- учебно-методическими материалами (текстовые занятия, полезными материалами);
- практическими заданиями.

Кадровое обеспечение:

К реализации программы в качестве педагогов дополнительного образования допускаются лица:

- 1) отвечающее одному из требований:
 - а) имеющее высшее образование или среднее профессиональное образование в рамках укрупненных групп специальностей и направлений подготовки высшего

образования и специальностей среднего профессионального образования «Образование и педагогические науки»;

б) имеющее высшее образование либо среднее профессиональное образование в рамках иных укрупненных групп специальностей и направлений подготовки высшего образования и специальностей среднего профессионального образования при условии его соответствия дополнительной общеобразовательной общеразвивающей программе, реализуемой ООО «Ребреин», и получение при необходимости дополнительного профессионального образования педагогической направленности;

в) успешно прошедшее промежуточной аттестации не менее чем за два года обучения по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующей направленности дополнительной общеобразовательной общеразвивающей программе;

2) не имеющее ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации;

3) прошедшее обязательный предварительный (при поступлении на работу) и периодические медицинские осмотры (обследования), а также внеочередные медицинские осмотры (обследования) в порядке, установленном законодательством Российской Федерации.

Реализация Программы также возможна лицами, привлекаемыми на условиях гражданско-правового договора в соответствии с действующим законодательством РФ.

2. УЧЕБНЫЙ ПЛАН

№ п/ п	Наименование модуля	Количество часов			Формы контроля / аттестация
		Всего	Теория	Практика	
1	Модуль 1 “Онбординг”	1	0,5	0,5	Входное тестирование
2	Модуль 2 “Terraform”	40	15	25	Практическое задание
3	Итоговая аттестация	7		7	Итоговое практическое задание

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование модуля	1 неделя	2 неделя	3 неделя	4 неделя
1	Модуль 1 “Онбординг”	1			
2	Модуль 2 “Terraform”	12	13	13	2
3	Итоговая аттестация				7 А

4. РАБОЧАЯ ПРОГРАММА

Модуль 1. Онбординг

Теория 0,5 академ. ч. Практика 0,5 академ. ч.

Модуль состоит из следующих тем:

Тема 01: Онбординг

В модуле обучающемуся предоставляется вводный конспект, содержащий общую информацию о программе, структуре курса, форматах взаимодействия с материалами и ожидаемых результатах обучения.

Предусмотрено прохождение входного тестирования, включающего 7 вопросов, направленных на закрепление информации из онбординга. В рамках темы обучающийся выполняет задание по целеполаганию: формулирует свою цель прохождения программы, указывает желаемые навыки по окончании обучения, а также оценивает текущий уровень своих знаний по DevOps, выбрав один из предложенных вариантов.

Модуль 2. Terraform

Теория 15 академ. ч. Практика 25 академ. ч.

Модуль состоит из следующих тем:

Тема 1: Инфраструктура как код

Содержание: Понимание парадигмы «инфраструктура как код». Ключевые принципы IaC. Преимущества использования IaC. Подходы к управлению конфигурациями инфраструктуры. Terraform как инструмент для управления IaC. Тестирование.

Тема 2: Введение в Terraform и его установка

Содержание: Краткий обзор Terraform. Под капотом Terraform (файлы конфигурации, компоненты и т.д.). Установка Terraform: пошаговое руководство. Практическое задание.

Тема 3: Воркфлоу Terraform и первый запуск

Содержание: Рабочий процесс (workflow) Terraform. Синтаксис HCL в Terraform на примере Yandex Cloud. Форматирование кода Terraform с помощью terraform fmt. Первый запуск Terraform на примере Yandex Cloud. Практическое задание.

Тема 4: Terraform settings (настройки ядра Terraform). Команды «terraform init» и «terraform validate»

Содержание: Возможности Terraform: настройка основных параметров. Инициализация в Terraform. Обеспечение достоверности и согласованности кода Terraform. Практическое задание.

Тема 5: Оттачиваем workflow: plan, apply, destroy

Содержание: Команды «terraform plan», «terraform apply», «terraform destroy» (суть команд, лучшие практики, примеры использования). Практическое задание.

Тема 6: Углубленная работа с провайдерами. Разбор блоков «resource» и «data»

Содержание: Работа с провайдерами (что делают, откуда берутся, установка провайдера). Блок «resource». Блок «data» (источники данных). Пример использования блоков resource и data вместе. Практическое задание.

Тема 7: Переменные (variables)

Содержание: типы переменных (локальные переменные; входные переменные и поддерживаемые типы данных). Как подставить переменные с помощью командной строки. Как определяется приоритет переменных. Как использовать переменные в манифестах. Практическое задание.

Тема 8: Выходные значения (outputs)

Содержание: где и как применяются выходные значения. Как объявить выходное значение и получить доступ к выводам дочернего модуля. Какие необязательные аргументы можно указать для выходного значения. Практическое задание.

Тема 9: Работа с циклами в Terraform: count / for_each

Содержание: Мета-аргумент count (счётчик). Мета-аргумент for_each. Когда использовать for_each, а когда count. Практическое задание.

Тема 10: Работа с условиями и динамическими блоками: lookup, dynamic

Содержание: Условные выражения. Пользовательские проверки условий. Проверки на наличие значений (lookup). Динамические блоки. Практическое задание.

Тема 11: Backends: удалённое хранение файла состояния terraform.tfstate

Содержание: что такое файлы состояния и бэкенд (backend) в Terraform. Пример бэкенда. Передача данных с помощью output и backend. Практическое задание.

Тема 12: Модули в Terraform

Содержание: что будет, если писать файлы конфигурации в одном каталоге. Как модули помогут избежать проблем с управлением конфигурациями. Что такое корневой и дочерний модули. Как использовать модули наиболее эффективно. Как работать с модулями на практике. Практическое задание.

Тема 13: Обёртка Terragrunt

Содержание: зачем нужен Terragrunt. Удалённые конфигурации Terraform. Как установить Terragrunt и использовать его. Практическое задание.

Тема 14: Обёртка Terraspace

Содержание: Концепции Terraspace. Отличия Terraspace от Terragrunt. Практический пример работы с Terraspace. Практическое задание.

Тема 15: Альтернативы Terraform

Содержание: OpenTofu. Ansible. CloudFormation. AWS CDK (Cloud Development Kit). Pulumi. Тестирование.

Модуль направлен на формирование у обучающихся базовых и продвинутых навыков работы с Terraform. Каждая тема модуля включает текстовое занятие с теоретическим материалом и пошаговыми инструкциями, после изучения которого предлагается практическое задание. Практические задания рассчитаны на 1-2 академических часа. Выполнение заданий предполагает отправку решения на проверку через личный кабинет обучающегося. Критерии оценки прописаны в описании к каждому заданию. В случае корректного выполнения выставляется зачёт. Если работа содержит ошибки, задание возвращается на доработку. При повторной неудачной попытке (после двух доработок) обучающийся получает «незачёт».

Итоговая аттестация.

Блок посвящён выполнению финального практического задания без предварительного теоретического блока.

5. МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Программа обеспечена системой дистанционного обучения <https://rebrainme.com/>.

Педагогические технологии:

- технология дифференцированного обучения;
- технология разноуровневого обучения;
- технология развивающего обучения;
- технология проблемного обучения;
- технология дистанционного обучения.

Методы обучения:

- словесный, наглядный практический;
- объяснительно – иллюстративный;
- частично-поисковый, исследовательский проблемный;
- игровой, дискуссионный.

Электронно-библиотечные ресурсы и системы, информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.RU.
2. Собственные учебные материалы: <https://rebrainme.com/terraform/>
3. Официальная документация по Terraform [Электронный ресурс]: <https://developer.hashicorp.com/terraform>
4. Официальная документация по Terraform Yandex [Электронный ресурс]: <https://cloud.yandex.ru/docs/tutorials/infrastructure-management/terraform-quickstart>

6. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Оценочные материалы:

Для отслеживания результатов освоения программы среди слушателей проводится текущий контроль и итоговое оценивание.

Текущий контроль

Осуществление текущего контроля проводится после занятий в виде написания практических заданий или тестирований. Тематика и условия выполнения практических заданий расписаны в личном кабинете обучающегося в СДО. Педагог проверяет решение и принимает решение о принятии решения (зачет), о необходимости доработать решение или о незачете. Если промежуточный контроль представлен в виде тестирования, подсчет верных ответов и выставление оценки «зачёт» и «незачёт» происходят в автоматическим решиме в СДО.

Тема 3: Воркфлоу Terraform и первый запуск

Задание.

Примечание: Обратите внимание, что создаваемая для процесса обучения виртуальная машина в целях повышения удобства, уже имеет преднастроенную конфигурацию для работы с Yandex cloud и установленным Terraform

Для отработки практических навыков, а также понимания того, как будет проходить работа с практическими заданиями на платформе, вам потребуется воспроизвести процесс создания ресурсов и их удаления. После запуска и подключения к созданной виртуальной машине перейдите в каталог /opt, там должен всегда будет находиться ваш код.

Подробности:

1. Создайте файл main.tf:
! Обратите внимание: файл с описанием инфраструктуры в формате JSON уже находится в каталоге /opt. Используйте его при необходимости в вашем main.tf.

```

# блок конфигурации, которая определяет необходимого провайдера (Yandex Cloud) и его
источник
terraform {
  required_providers {
    yandex = {
      source = "yandex-cloud/yandex"
    }
  }
}

# создаёт провайдера Yandex Cloud и устанавливает параметры для подключения к Yandex Cloud
(зону размещения)
provider "yandex" {
  zone = "ru-central1-a"
  service_account_key_file = "authorized_key.json"
}
[...]

```

2. Не меняйте название ресурса output "vm_name" — здесь и далее эти значения будут проверяться системой автоматической проверки.
3. Для обеспечения уникальности вместо <vmname> впишите вашу фамилию, например: aalovelace
4. Исполните рабочий процесс Терраформ, выполнив команды в папке с файлом:
 - terraform init
 - terraform plan
 - terraform apply
 - terraform destroy
5. Перед отправкой на проверку задания ресурсы не должны существовать в облаке. Файлы с кодом проекта необходимо скопировать на созданную в платформе виртуальную машину в папку /opt, если вы создавали их где-то в другом месте.

Тема 8: Выходные значения (outputs)

Практическое задание

Напишите манифест Terraform, в котором необходимо описать создание виртуальной машины (регион не важен, в качестве ОС используйте Ubuntu 22.04). Ресурсы ЦПУ = 2, оперативная память = 2 Гб. Выведете в outputs публичный IP-адрес, размер диска и идентификатор образа.

- Outputs должны быть названы следующим образом
- ```

output "vm_public_ip" {} # публичный IP-адрес
output "vm_disk_size" {} # размер диска
output "vm_image_id" {} # идентификатор образа

```
- Не меняйте название ресурса output — здесь и далее эти значения будут проверяться системой автоматической проверки.
  - Для обеспечения уникальности виртуальной машины впишите вашу фамилию, инициалы и год рождения, например: vm-aalovelace1815.
  - Перед отправкой на проверку задания ресурсы не должны существовать в облаке. Файлы с кодом проекта необходимо скопировать на созданную в платформе виртуальную машину в папку /opt, если вы создавали их где-то в другом месте.

## **Тема 12: Модули в Terraform**

## **Задание**

Создайте модуль, в котором необходимо описать группу безопасности, открывающую доступ из интернета к виртуальной машине по портам, заданным через переменную.

Используйте ваш модуль и модуль из примера в тексте, чтобы создать виртуальную машину.

Подробности:

1. Создание модуля группы безопасности:
  - Создайте новую директорию для вашего модуля группы безопасности, например `security_group`.
  - В этой директории создайте файл `main.tf`, в котором опишите ресурсы для группы безопасности.
  - Определите переменную для портов, которые должны быть открыты, в файле `variables.tf`. Пример:

```
variable "allowed_ports" {
 type = list(number)
 default = [22, 80, 443]
 description = "Список портов, которые должны быть открыты для доступа из интернета"
}
```

## **Итоговое оценивание**

В конце программы обучающиеся сдают итоговую аттестацию.

## **Задание.**

Представьте, что вы работаете DevOps-инженером в группе, отвечающей за помощь командам разработчиков веб-сайтов вашей компании. Компания занимается разработкой сайтов на WordPress. Однако его развёртывание не автоматизировано.

Ваш руководитель хотел бы от вас решения проблемы с автоматизацией развёртывания сайта в Яндекс Облаке. Это необходимо для ускорения процесса создания новых окружений для различных проектных команд, использующих WordPress как основу своей разработки.

В текущем задании вам потребуется развернуть в облаке реальное приложение WordPress, которое будет использовать базу данных. Для успешного выполнения задания требуется создать следующие ресурсы для двух окружений (`dev` и `prod`):

1. Создайте VPC и две подсети: одну для БД (без доступа в интернет), вторую для виртуальной машины (с доступом в интернет).
2. В качестве операционной системы используйте Ubuntu 22.04.
3. Создайте Группы безопасности, которые будут обеспечивать доступ для виртуальной машины из интернета по портам 80 и 443. Внутри VPC должна быть обеспечена доступность СУБД только для подключений их подсети с виртуальной машины на порт СУБД.
4. Создайте необходимые ресурсы кластера СУБД для WordPress.
5. Создайте виртуальную машину, где должно быть развернуто приложение, установите приложение и настройте подключение к базе данных, используя раздел метаданных.

[...]

Результаты текущего контроля и итогового оценивания отображаются в личном кабинете слушателя в системе дистанционного обучения <https://rebrainme.com/>.

По результатам сдачи текущего контроля и итогового оценивания педагог даёт обратную связь слушателям, отмечает их сильные стороны и обращает внимание на зоны для развития. При необходимости педагог может повторить пройденные темы со слушателями, если установлен факт плохого закрепления и усвоения темы у слушателей.