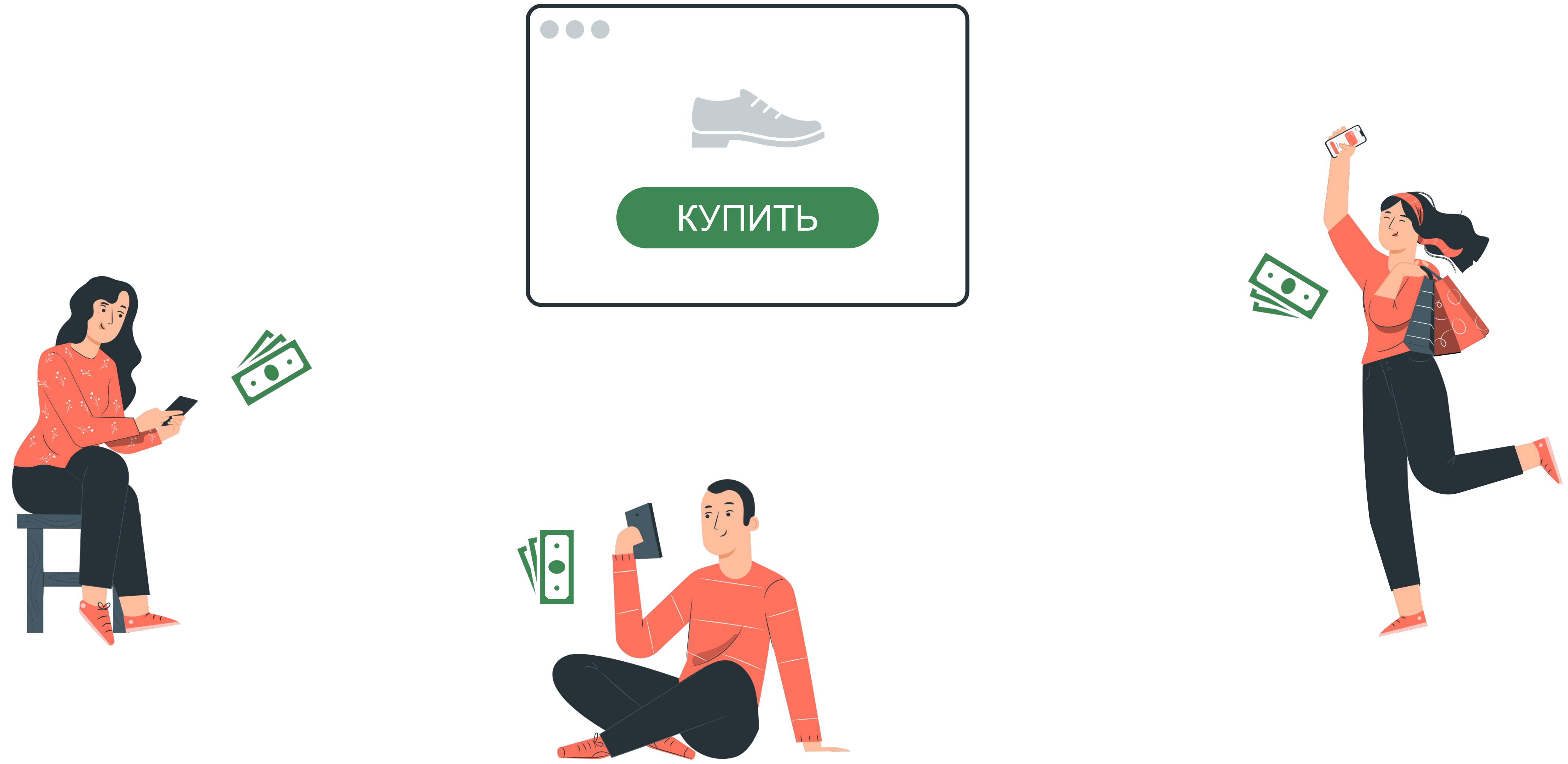
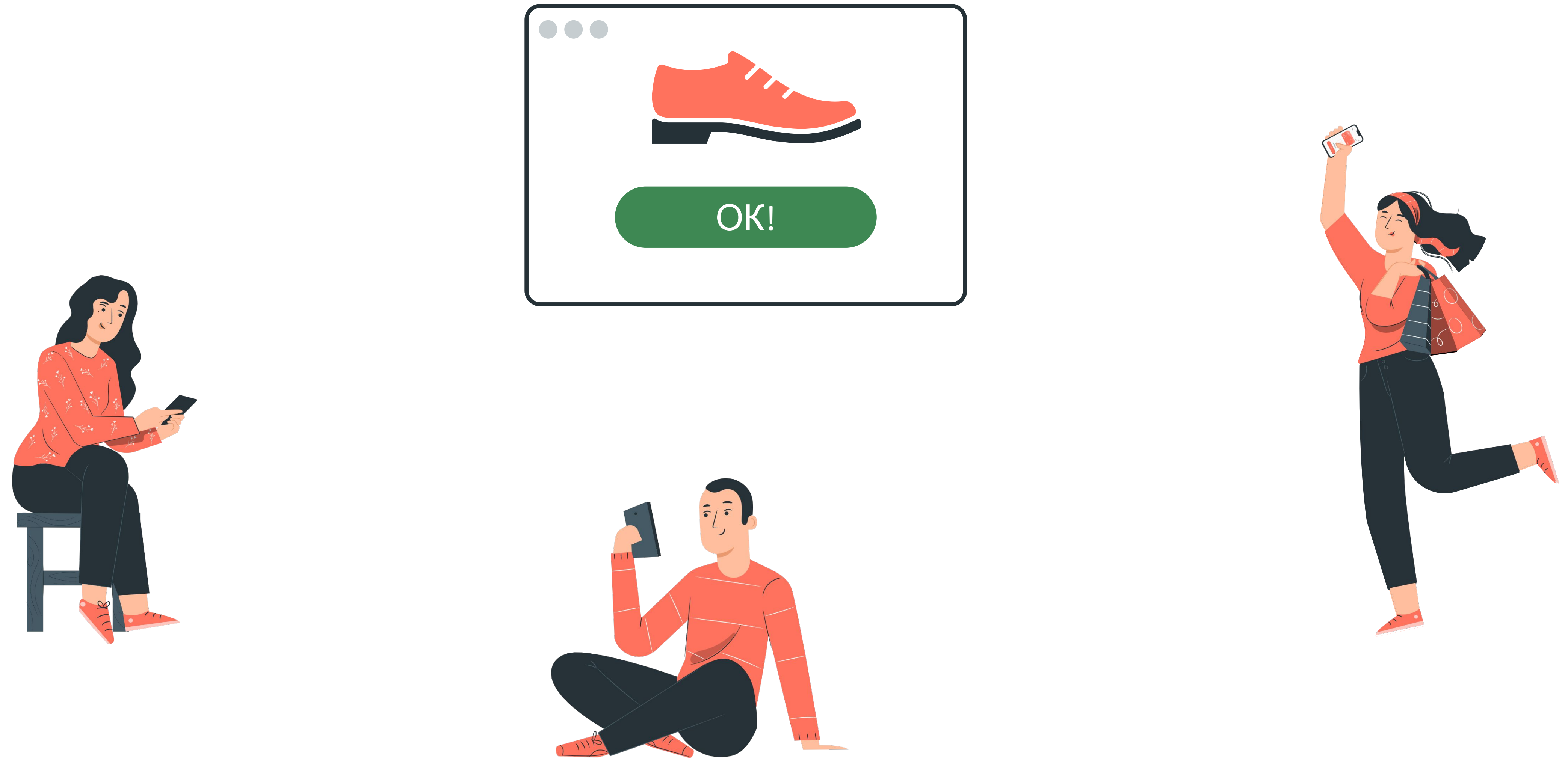
Skillbox

Знакомство с Terraform

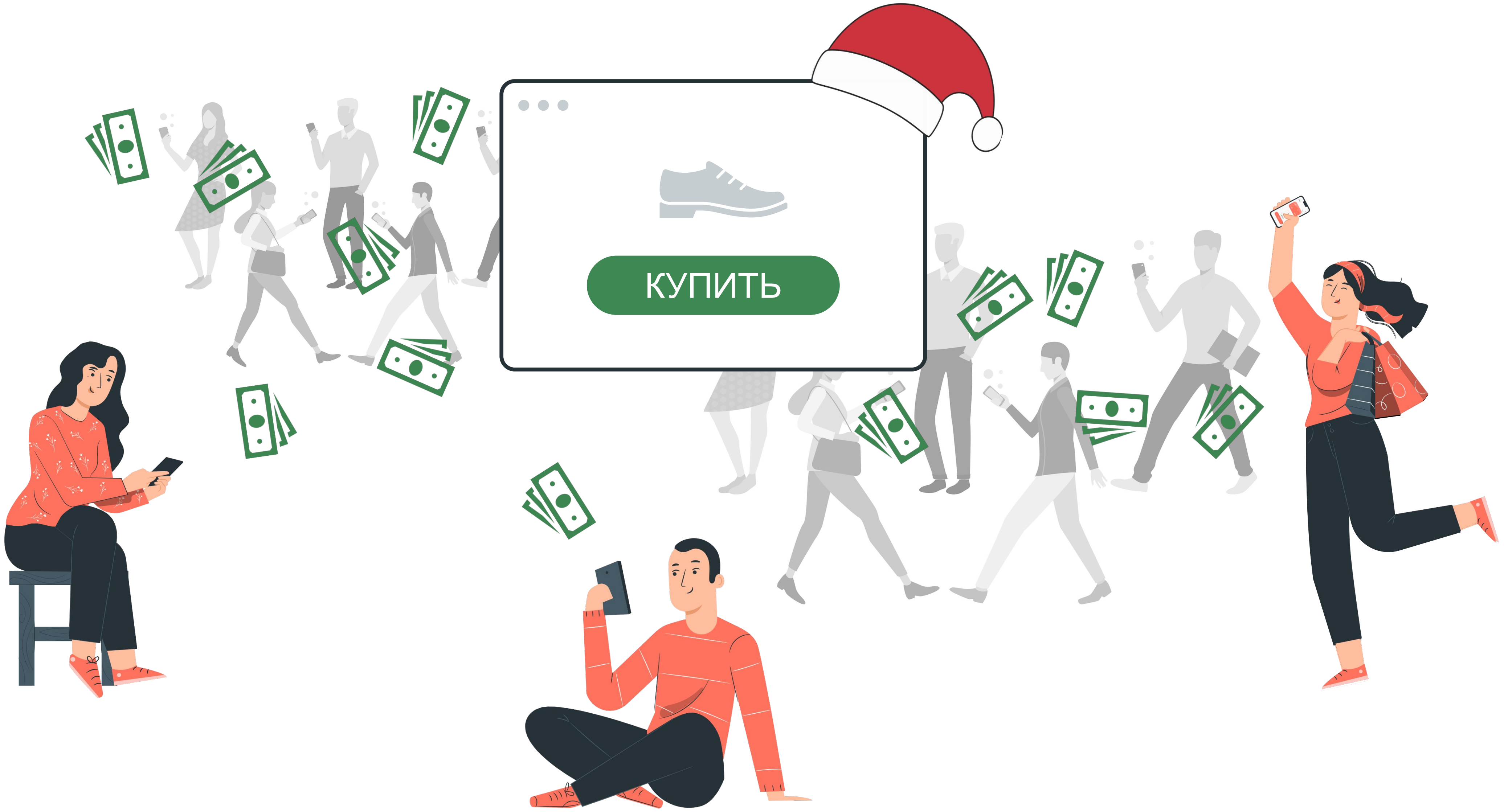
Skillbox



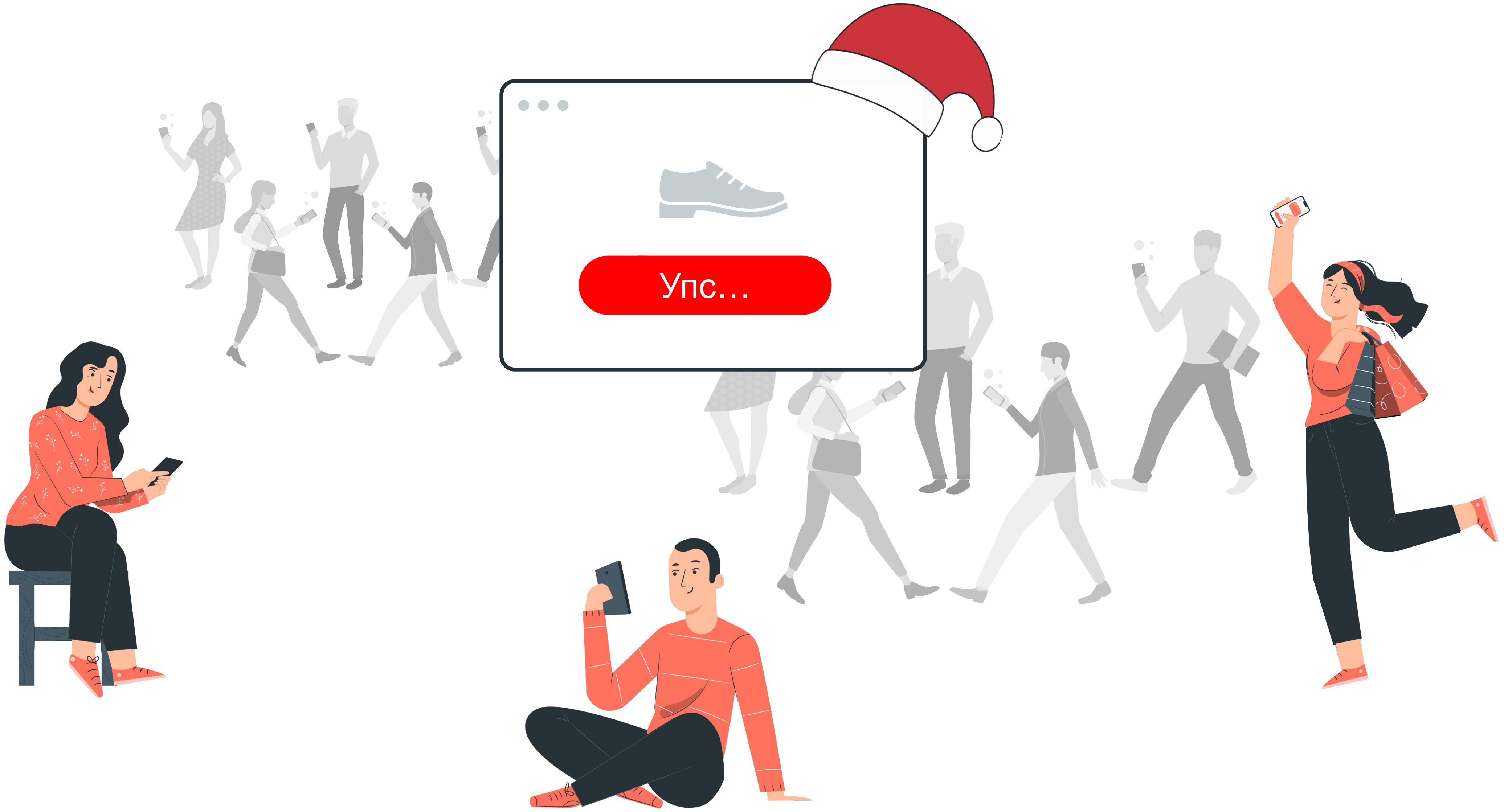
Skillbox



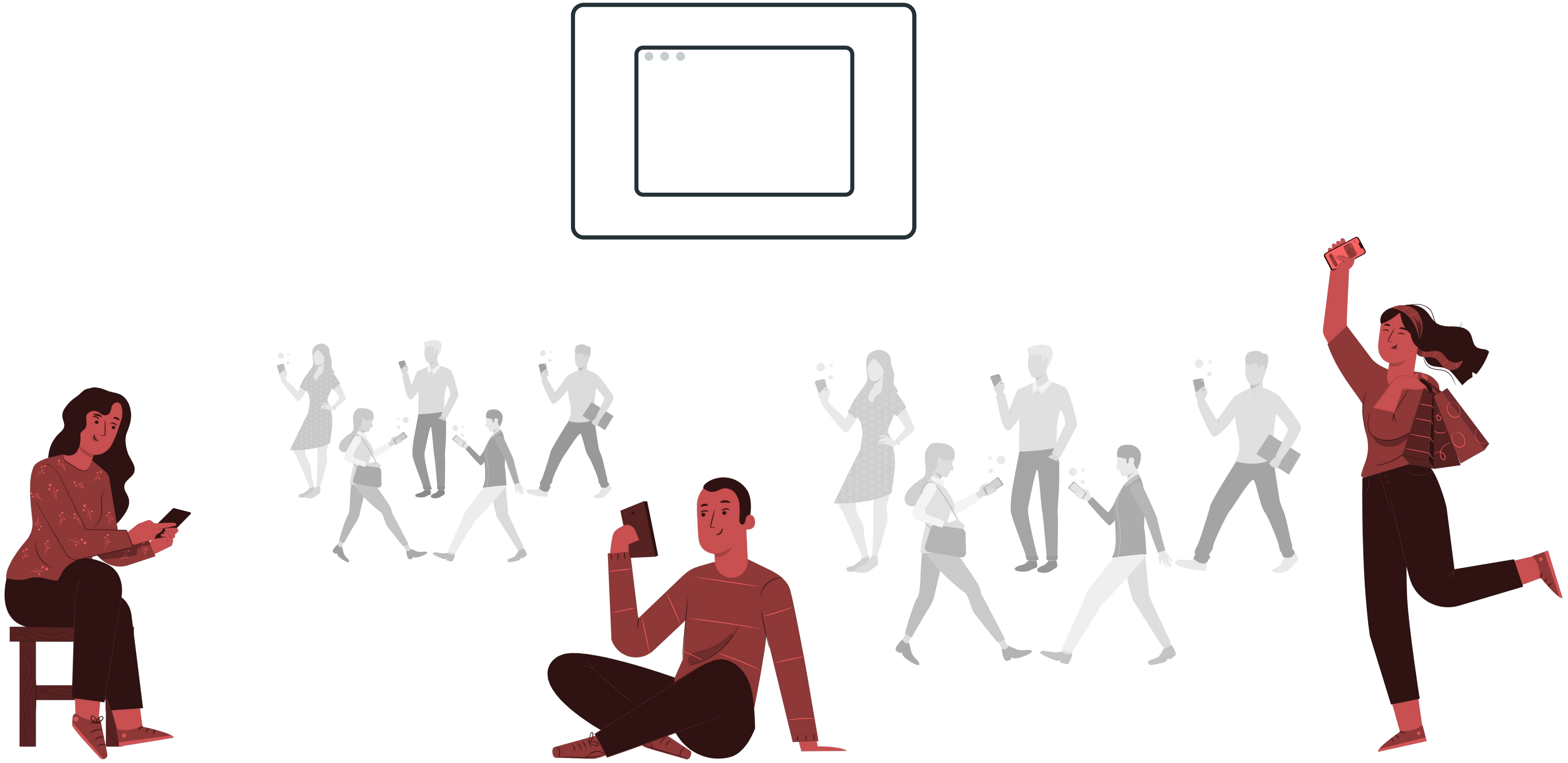
Skillbox



Skillbox



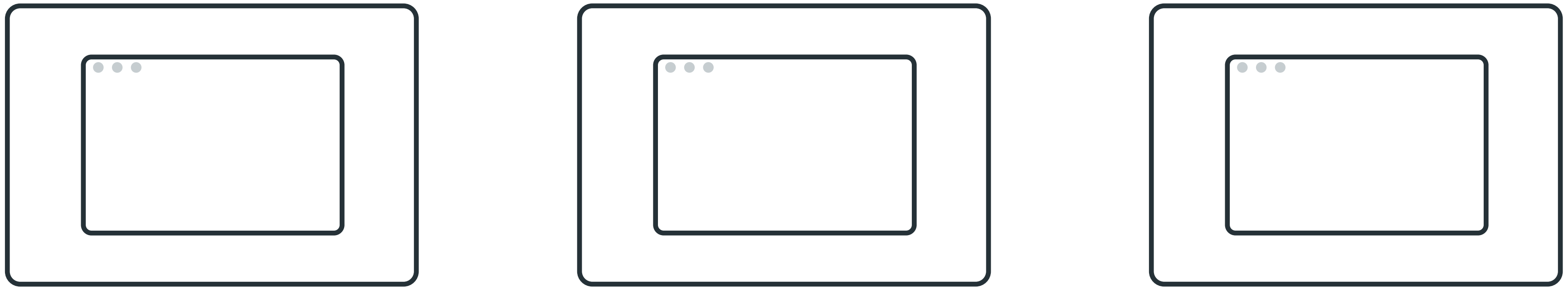
?!
error
?!
?!



Skillbox

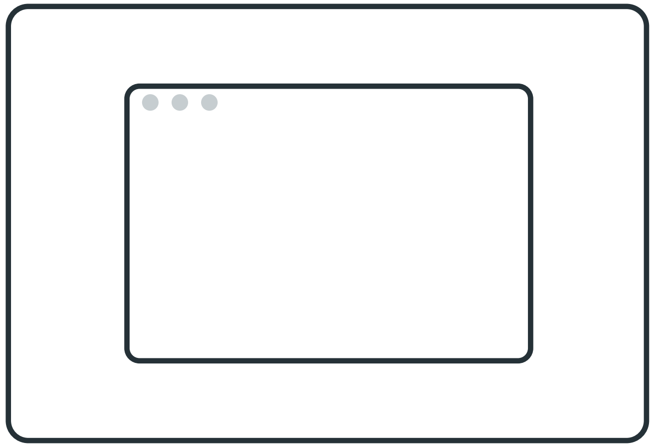
Skillbox

День



Skillbox

Ночь



Skillbox

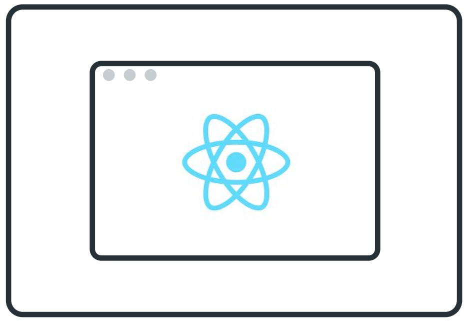
Запрос к API AWS для создания инстанса

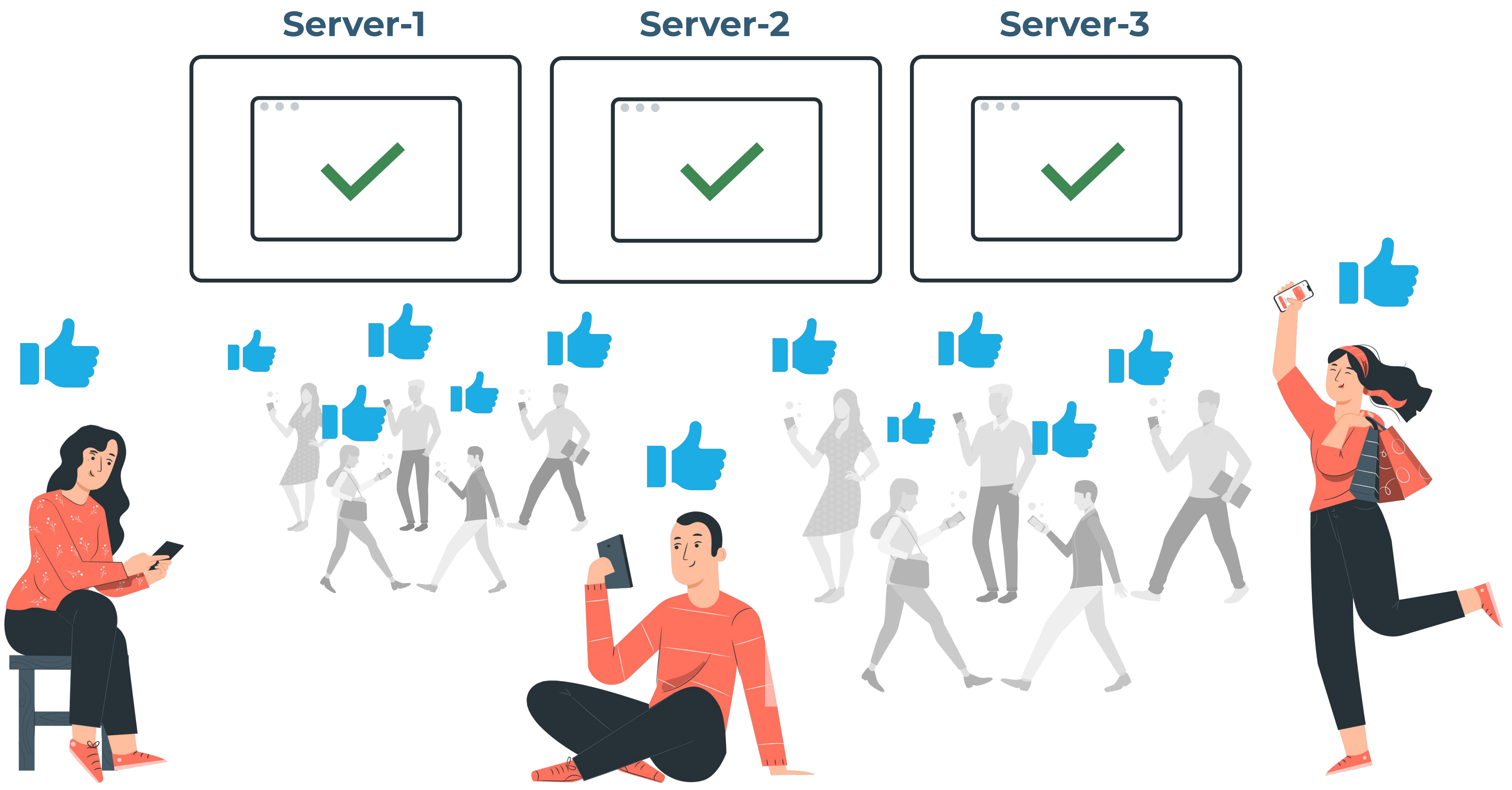
```
https://ec2.amazonaws.com/?Action=RunInstances
&ImageId=ami-2bb65342
&MaxCount=3
&MinCount=1
&Placement.AvailabilityZone=us-east-1a
&Monitoring.Enabled=true
&Version=2016-11-15
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLEus-east-1%2Fec2%2Faws4_request
&X-Amz-Date=20130813T150206Z
&X-Amz-SignedHeaders=content-type%3Bhost%3Bx-amz-date
&X-Amz-Signature=ced6826de92d2bdeed8f846f0bf508e8559e98e4b0194b84example54174deb456c
Content-type: application/json
host:ec2.amazonaws.com
```

HashiCorp

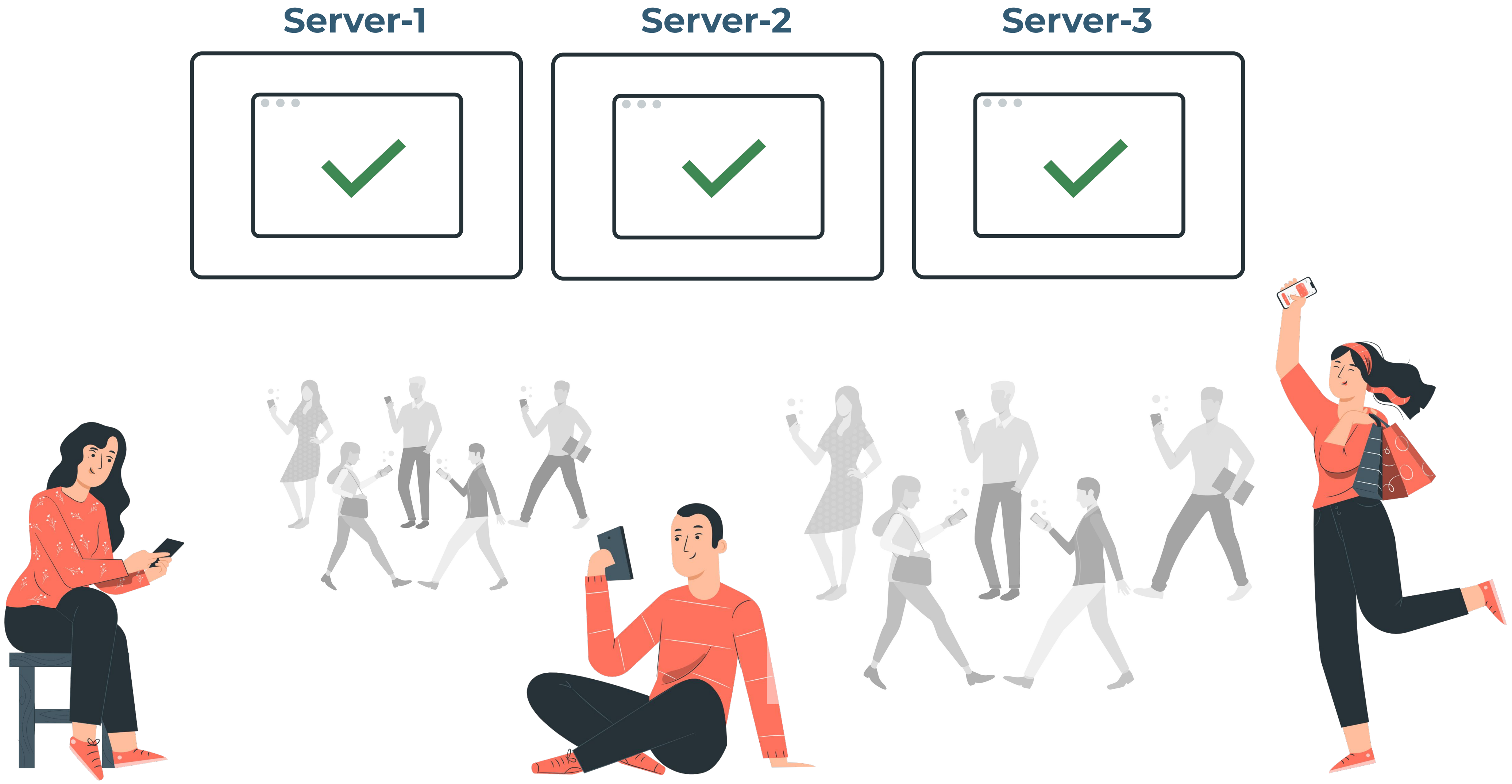
Terraform

Skillbox



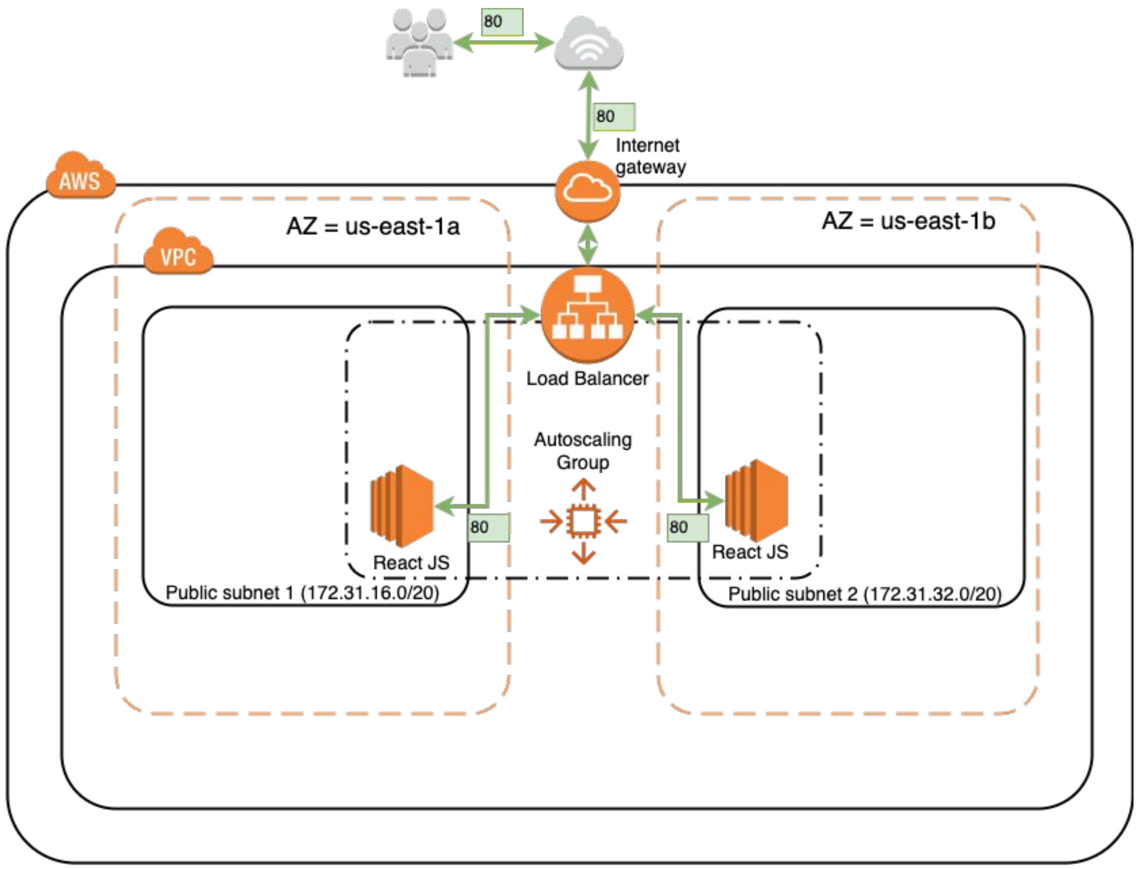

Server-1
Server-2
Server-3

Skillbox



Skillbox

Инфраструктура с балансировкой нагрузки



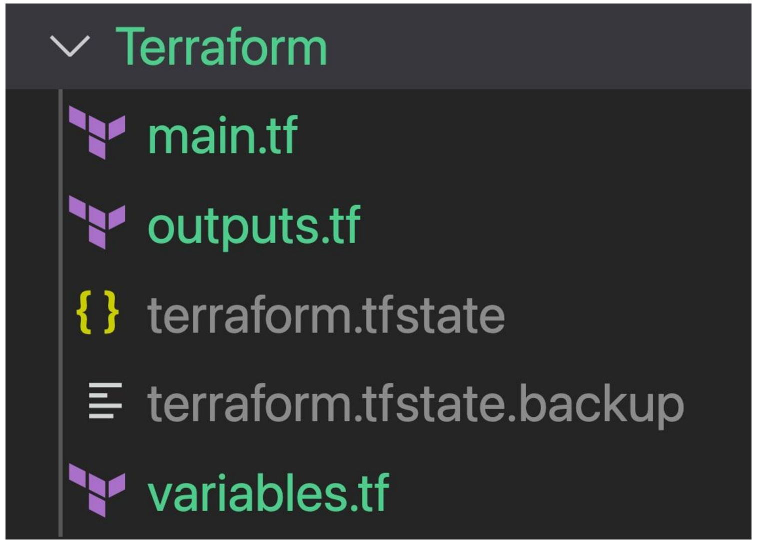
Skillbox

Концепция Terraform

- ▹ Написан на Go
- ▹ Декларативное описание инфраструктуры
- ▹ Не имеет привязки к сервису
- ▹ Использует подключаемые провайдеры

Skillbox

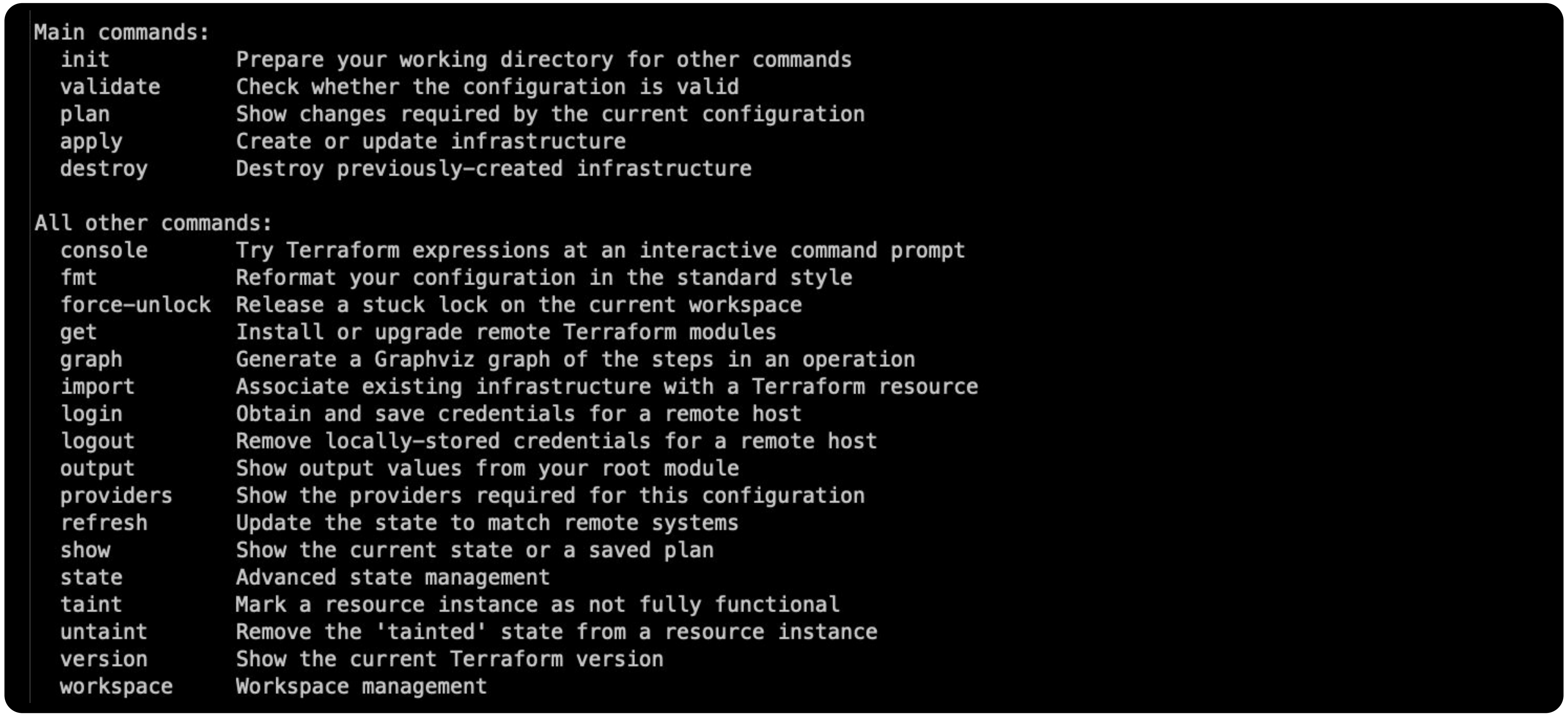
Структура файлов в директории Terraform



Terraform
- main.tf
- outputs.tf
- terraform.tfstate
- terraform.tfstate.backup
- variables.tf

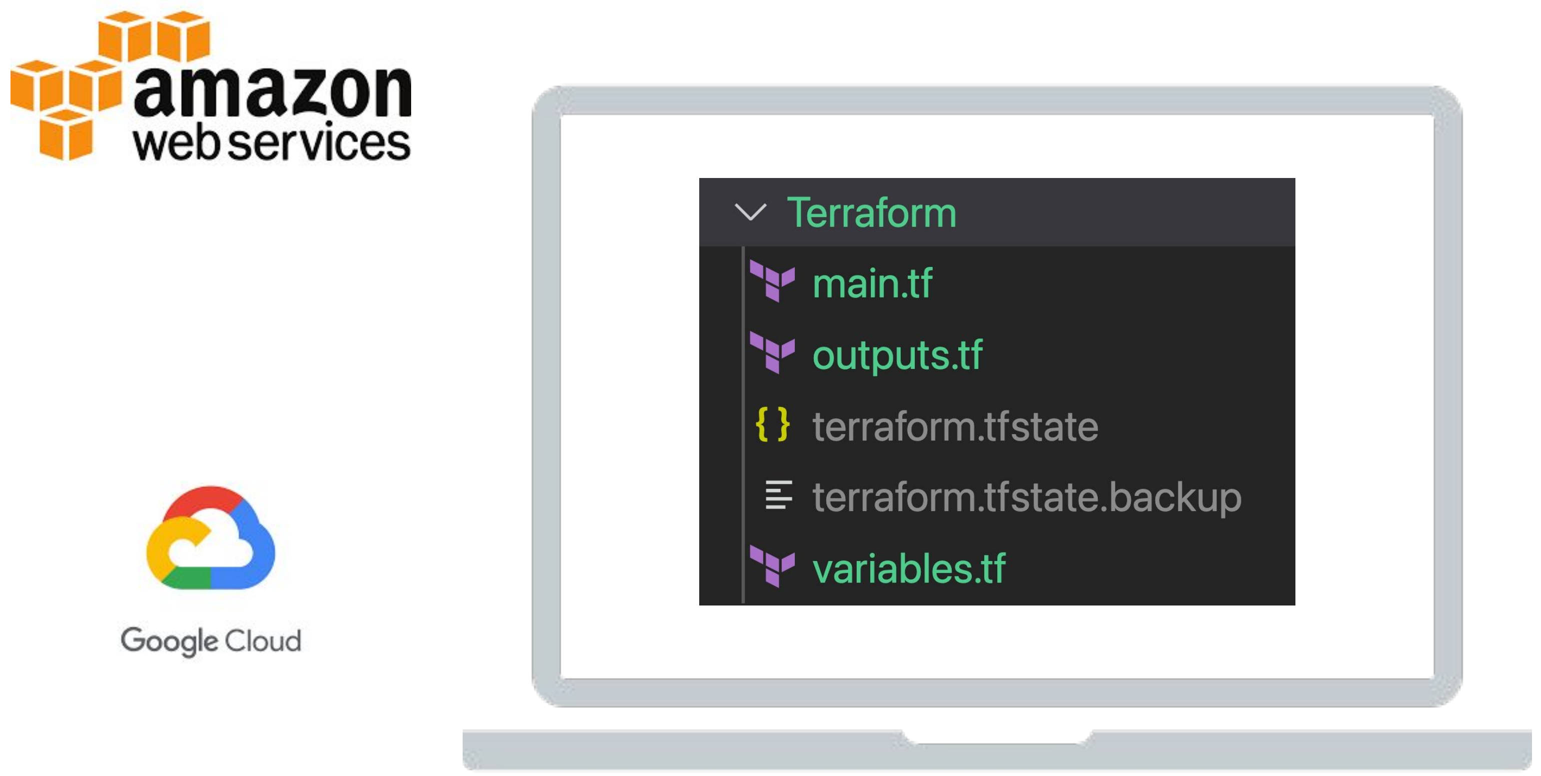
Skillbox

Основные команды Terraform



```
Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management
```



Terraform
main.tf
outputs.tf
terraform.tfstate
terraform.tfstate.backup
variables.tf

Skillbox

terraform init

- Создаёт папку .terraform
- Определяет используемые модули и провайдеры
- Загружает плагины в .terraform

terraform plan

- Создаёт план выполнения
- Выполняет обновления
- Определяет необходимые действия для желаемого состояния

Skillbox

Создание и удаление инфраструктуры

terraform apply

Skillbox

Создание и удаление инфраструктуры

terraform apply

terraform destroy

HCL Configuration Syntax

```
provider "aws" {
 region = "us-east-1"
}

resource "aws_eip" "my_static_ip" {
 instance = aws_instance.my_webserver.id
}

data "aws_ami" "ubuntu" {
 owners      = ["099720109477"]
 most_recent = true
 filter {
   name   = "name"
   values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
 }
}

variable "ssh_key_name" {
 type        = string
 default     = "id_rsa"
 description = "description"
}
```

Skillbox

Блок — это контейнер для другого контента

```
resource "aws_instance" "reactjs_server" {
   ami = data.aws_ami.ubuntu.id

   network_interface {
   # ...
 }
   tags = {
       Name  = "ReactJS Server IP"
   }
}
```

Skillbox

Variable

```
variable "ssh_key_name" {
 type        = string
 default     = "id_rsa"
 description = "description"
}
```

Skillbox

Functions

```
split (separator, string)
join (separator, list)
regex (pattern, string)
replace (string, substring, replacement)
contains (list, value)
file (path)
...
```

Skillbox

Provider

```
provider "aws" {
 region = "us-east-1"
}
```

Skillbox

Resource

```
resource "aws_eip" "my_static_ip" {
 instance = aws_instance.my_webserver.id
}
```

Skillbox

Data

```
data "aws_ami" "ubuntu" {
 owners      = ["099720109477"]
 most_recent = true
 filter {
   name   = "name"
   values = ["ubuntu-focal-20.04-amd64-server-*"]
 }
}
```

Variable

```
variable "ssh_key_name" {
 type        = string
 default     = "id_rsa"
 description = "description"
}
```

Переменные языка HCL

```
variables
  input
  output
  local

types
  simple
  construction

functions
  split
  join
  regex...
```

https://www.terraform.io/docs/configuration/functions.html

Skillbox

Обращение к объектам

```
<RESOURCE TYPE>.<NAME>

var.<NAME>

local.<NAME>

var.names[0] или var.names.0

"instance ${aws_instance.test.name} created"
```

Skillbox

Провайдер и регион

```
provider "aws" {
 region = "us-east-1"
}
```

Skillbox

Поиск Ubuntu

```
# Ищем образ с последней версией Ubuntu
data "aws_ami" "ubuntu" {
 owners      = ["099720109477"]
 most_recent = true
 filter {
   name   = "name"
   values =
["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd
64-server-*"]
 }
}
```

IP

```
resource "aws_eip" "my_static_ip" {
 instance = aws_instance.my_webserver.id
 tags = {
   Name  = "ReactJS Server IP"
 }
}
```

Skillbox

Политики безопасности

```
resource "aws_security_group" "my_webserver" {
  name        = "ReactJS Servers Security Group"
  description = "Security group for accessing traffic to our ReactJS Server"


  dynamic "ingress" {
    for_each = ["80"]
    content {
      from_port   = ingress.value
      to_port     = ingress.value
      protocol    = "tcp"
      cidr_blocks = ["0.0.0.0/0"]
    }
  }


  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name   = "ReactJS Server SecurityGroup"
  }
}
```

Skillbox

Конфигурация сервера

```
resource "aws_instance" "my_webserver" {
 ami                    = data.aws_ami.ubuntu.id
 instance_type          = "t3.micro"
 vpc_security_group_ids = [aws_security_group.my_webserver.id]
 user_data = file("user_data.sh")
 key_name = "id_rsa"
 tags = {
   Name  = "ReactJS Server IP"
   Env = "Production"
   Tier = "Frontend"
 }
}
```

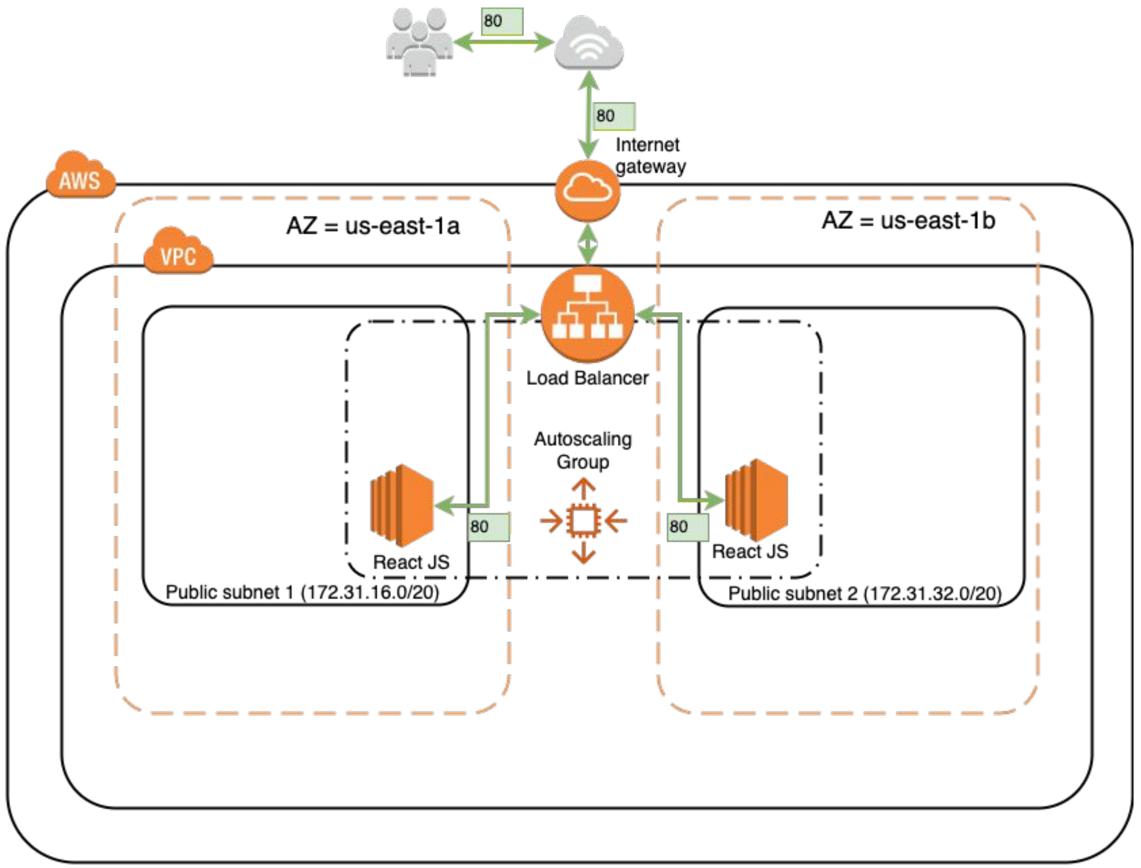
Output

```
output "my_web_site_ip" {
 description = "Elatic IP address assigned to our WebSite"
 value       = aws_eip.my_static_ip.public_ip
}
```

Skillbox

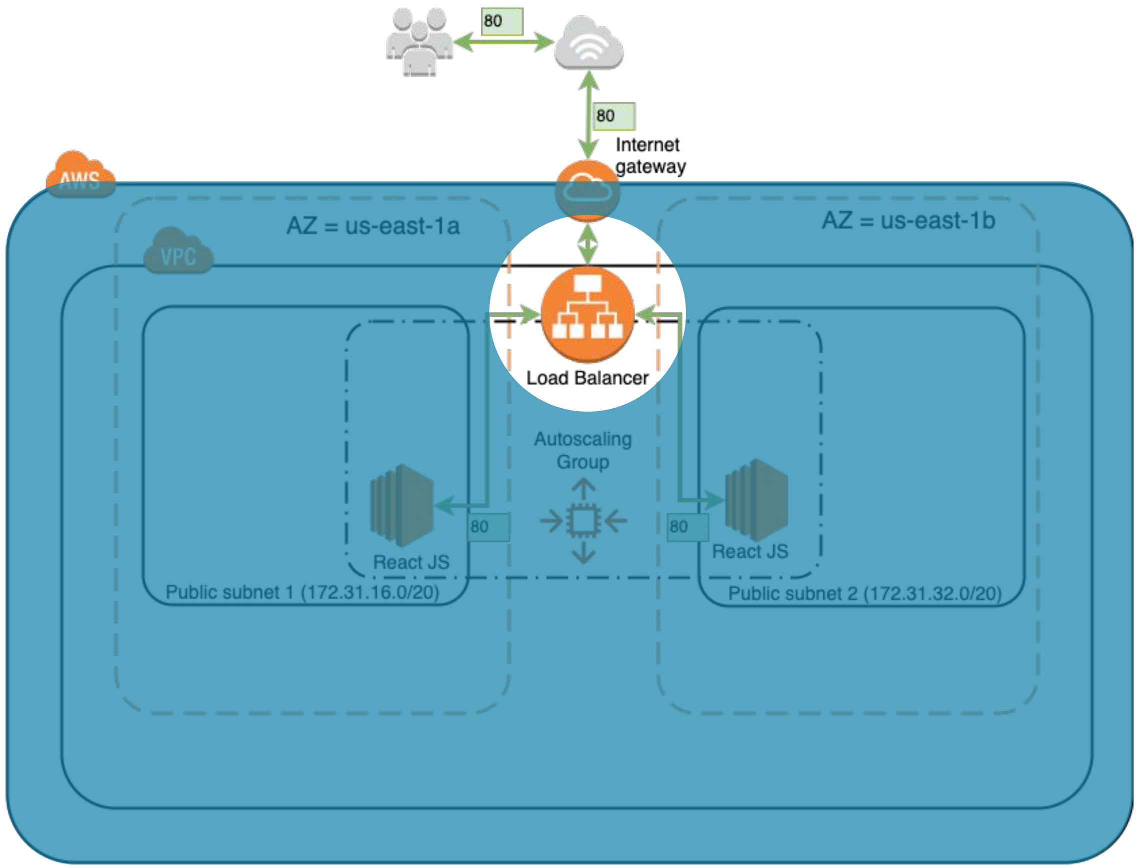
Практика

Инфраструктура с балансировкой нагрузки



Skillbox

Инфраструктура с балансировкой нагрузки



AWS Provider

```
# Указываем, что мы хотим разворачивать окружение в AWS
provider "aws" {
 region = "us-east-1"
}
```

Skillbox

Дата центры

```
# Узнаём, какие есть Дата центры в выбранном регионе
data "aws_availability_zones" "available" {}
```

Skillbox

Подсети

```
# Созаём подсети в разных Дата центрах
resource "aws_default_subnet" "availability_zone_1" {
 availability_zone =
data.aws_availability_zones.available.names[0]
}

resource "aws_default_subnet" "availability_zone_2" {
 availability_zone =
data.aws_availability_zones.available.names[1]
}
```

Skillbox

Последняя версия Ubuntu

```
# Ищем образ с последней версией Ubuntu
data "aws_ami" "ubuntu" {
 owners      = ["099720109477"]
 most_recent = true
 filter {
   name   = "name"
   values = ["ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-*"]
 }
}
```

Skillbox

Динамическое добавление правил разрешения трафика

```
 Созаём правило, которое будет разрешать трафик к нашим серверам
resource "aws_security_group" "web" {
 name = "Dynamic Security Group"

 dynamic "ingress" {
   # Зададим правило, по каким портам можно обращаться к нашим серверам
   for_each = ["22", "80"]
   content {
     from_port   = ingress.value
     to_port     = ingress.value
     protocol    = "tcp"
     cidr_blocks = ["0.0.0.0/0"]
   }
 }
 egress {
   from_port   = 0
   to_port     = 0
   protocol    = "-1"
   cidr_blocks = ["0.0.0.0/0"]
 }

 tags = {
   Name  = "Web access for Application"
 }
}
```

Skillbox

Launch Configuration

```
resource "aws_launch_configuration" "web" {
 name_prefix     = "Web-server-"
 image_id        = data.aws_ami.ubuntu.id
 instance_type   = "t3.micro"
 security_groups = [aws_security_group.web.id]
 user_data       = file("user_data.sh")
 iam_instance_profile = "AmazonEC2RoleForSSM"
 key_name = "id_rsa"
}
```

Skillbox

User Data

```
#!/bin/bash -xe
exec > >(tee /var/log/user-data.log |logger -t user-data -s 2>/dev/console) 2>&1
cd /home/ubuntu/
git clone https://gitlab.com/entsupml/skillbox-deploy-blue-green
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main"  | sudo tee
/etc/apt/sources.list.d/yarn.list
sudo apt update -y && sudo apt install yarn -y
cd /home/ubuntu/skillbox-deploy-blue-green/
sudo apt install nodejs -y
sudo apt install npm -y
npm install
# We can get the IP address of instance
myip=`curl http://169.254.169.254/latest/meta-data/local-ipv4`
npm install pm2 -g
export PORT=80
sed -i 's|Test of revert|'$myip'|g' src/App.js
yarn start &
```

Skillbox

Autoscaling Group

```
resource "aws_autoscaling_group" "web" {
 name                    = "ASG-${aws_launch_configuration.web.name }"
 launch_configuration = aws_launch_configuration.web.name
 min_size                = 2
 max_size                = 2
 min_elb_capacity        = 2
 health_check_type       = "ELB"
 vpc_zone_identifier    = [aws_default_subnet.availability_zone_1.id, aws_default_subnet.availability_zone_2.id]
 load_balancers          = [aws_elb.web.name]

 dynamic "tag" {
   for_each = {
     Name    = "WebServer in Auto Scalling Group"
   }
   content {
     key                   = tag.key
     value                 = tag.value
     propagate_at_launch =  true
   }
 }
  lifecycle {
   create_before_destroy =  true
 }
}
```

Skillbox

Балансировщик нагрузки

```
resource "aws_elb" "web" {
 name               = "WebServer-Highly-Available-ELB"
 availability_zones = [data.aws_availability_zones.available.names[0],
      data.aws_availability_zones.available.names[1]]
 security_groups    = [aws_security_group.web.id]
 listener {
   lb_port           = 80
   lb_protocol       = "http"
   instance_port     = 80
   instance_protocol = "http"
 }
 health_check {
   healthy_threshold   = 2
   unhealthy_threshold = 2
   timeout             = 3
   target              = "HTTP:80/"
   interval            = 10
 }
 tags = {
   Name = "WebServer-Highly-Available-ELB"
 }
}
```

Output

```
output "web_loadbalancer_url" {
 value = aws_elb.web.dns_name
}
```

Skillbox

Практика

Итог

- Чем нам помог Terraform
- Развернули сервер с ReactJS приложением
- Развернули инфраструктуру с балансировщиком нагрузки