Infrastructure as a code. Terraform. Lection 1.

INFRASTRUCTURE AS CODE. Basic definition.

Infrastructure as code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. The IT infrastructure managed by this process comprises both physical equipment, such as bare-metal servers, as well as virtual machines, and associated configuration resources. The definitions may be in a version control system. It can use either scripts or declarative definitions, rather than manual processes, but the term is more often used to promote declarative approaches.

INFRASTRUCTURE AS CODE. Frequently used tools

HashiCorp Terraform

CloudFormation

AWS CloudFormation

Azure Resource Manager

Google Cloud
Deployment Manager

INFRASTRUCTURE AS CODE. Terraform

HashiCorp
Terraform

Terraform is an infrastructure provisioning tool created by Hashicorp. It allows you to describe your infrastructure as code, creates “execution plans” that outline exactly what will happen when you run your code, builds a graph of your resources, and automates changes with minimal human interaction.

Terraform uses its own domain-specific language (DSL) called Hashicorp Configuration Language (HCL). HCL is JSON-compatible and is used to create these configuration files that describe the infrastructure resources to be deployed.

Terraform is cloud-agnostic and allows you to automate infrastructure stacks from multiple cloud service providers simultaneously and integrate other third-party services.

You even can write Terraform plugins to add new advanced functionality to the platform.

INFRASTRUCTURE AS CODE. AWS CloudFormation

Similar to Terraform, AWS CloudFormation is a configuration orchestration tool that allows you to code your infrastructure to automate your deployments.

Primary differences lie in that CloudFormation is deeply integrated into and can only be used with AWS, and CloudFormation templates can be created with YAML in addition to JSON.

CloudFormation allows you to preview proposed changes to your AWS infrastructure stack and see how they might impact your resources, and manages dependencies between these resources.

To ensure that deployment and updating of infrastructure is done in a controlled manner, CloudFormation uses Rollback Triggers to revert infrastructure stacks to a previous deployed state if errors are detected.

You can even deploy infrastructure stacks across multiple AWS accounts and regions with a single CloudFormation template. And much more.

INFRASTRUCTURE AS CODE.

Azure Resource Manager and Google Cloud Deployment Manager

If you're using Microsoft Azure or Google Cloud Platform, these cloud service providers offer their own IaC tools similar to AWS CloudFormation.

Azure Resource Manager allows you to define the infrastructure and dependencies for your app in templates, organize dependent resources into groups that can be deployed or deleted in a single action, control access to resources through user permissions, and more.

Google Cloud Deployment Manager offers many similar features to automate your GCP infrastructure stack. You can create templates using YAML or Python, preview what changes will be made before deploying, view your deployments in a console user interface, and much more.

INFRASTRUCTURE AS CODE. Terraform

Amazon Web Services

Microsoft Azure

Google Cloud Platform

Digital Ocean

AliCloud

Github

OR

You could develop "provider" for your own platform

Provider

INFRASTRUCTURE AS CODE. Terraform

Code syntax: Hashicorp Corporation Language (HCL)

Plain text

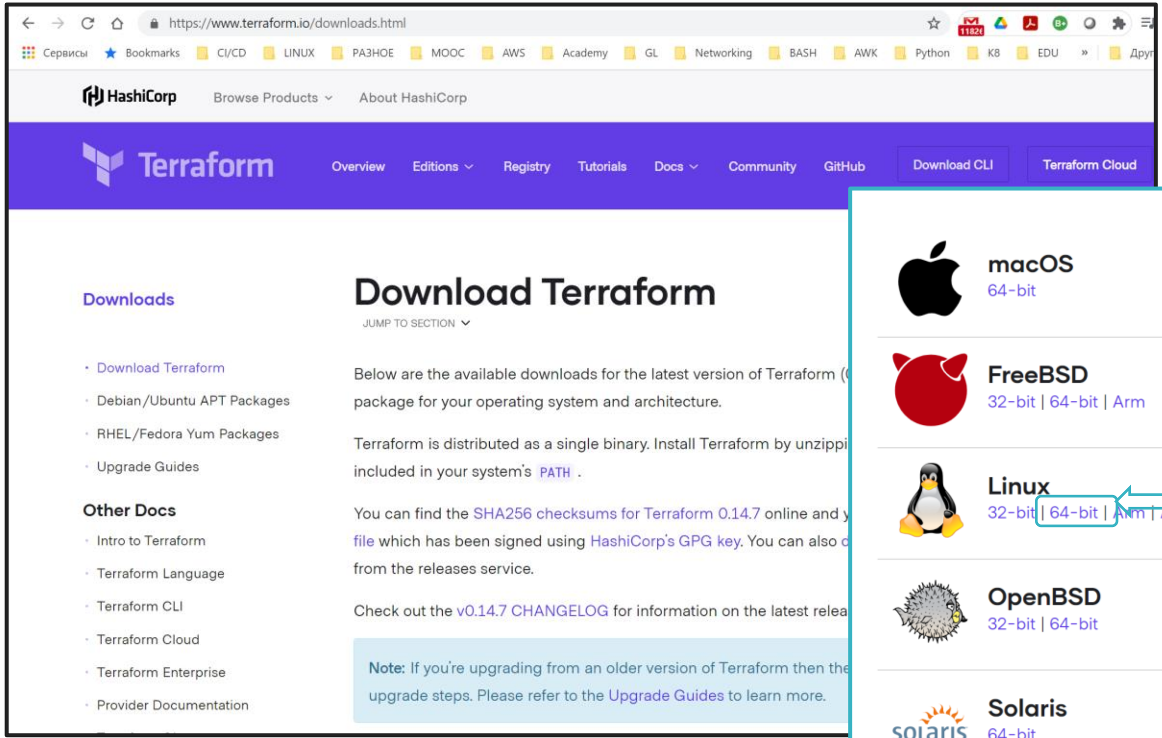
No IDE

Simple text editors

No compilation needed

Cross-platform: Linux, MacOS, MS Windows

INFRASTRUCTURE AS CODE. Terraform



https://www.terraform.io/downloads.html

Downloads

- Download Terraform
- Debian/Ubuntu APT Packages
- RHEL/Fedora Yum Packages
- Upgrade Guides

Other Docs

- Intro to Terraform
- Terraform Language
- Terraform CLI
- Terraform Cloud
- Terraform Enterprise
- Provider Documentation

Download Terraform

JUMP TO SECTION

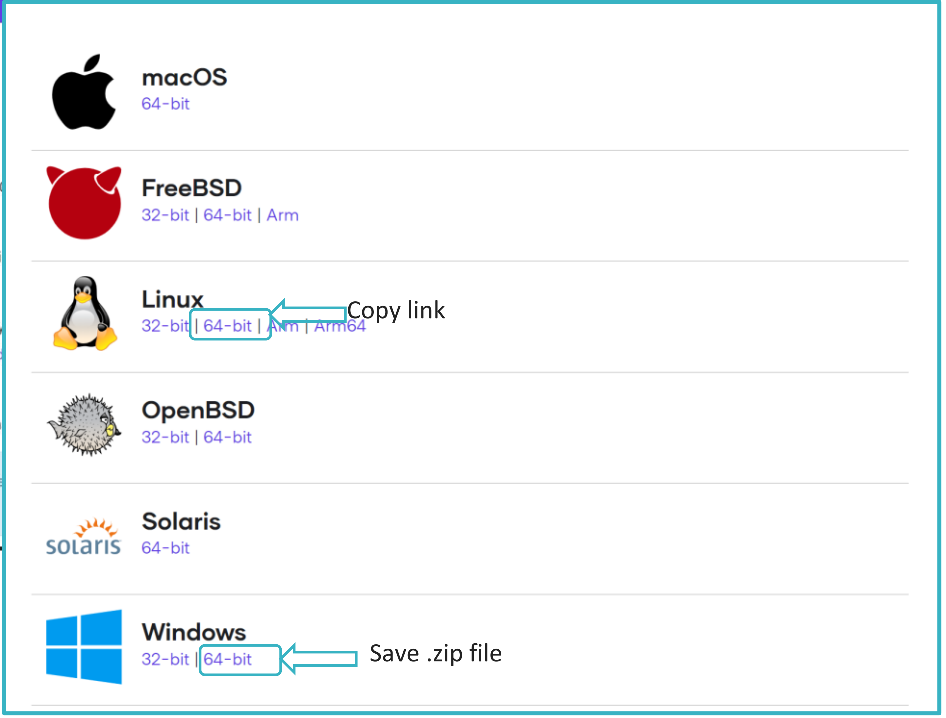
Below are the available downloads for the latest version of Terraform ((package for your operating system and architecture.

Terraform is distributed as a single binary. Install Terraform by unzippi included in your system's `PATH` .

You can find the SHA256 checksums for Terraform 0.14.7 online and y file which has been signed using HashiCorp's GPG key. You can also d from the releases service.

Check out the v0.14.7 CHANGELOG for information on the latest relea

Note: If you're upgrading from an older version of Terraform then the upgrade steps. Please refer to the Upgrade Guides to learn more.



macOS
64-bit

FreeBSD
32-bit | 64-bit | Arm

Linux
32-bit | 64-bit | Arm | Arm64

Copy link

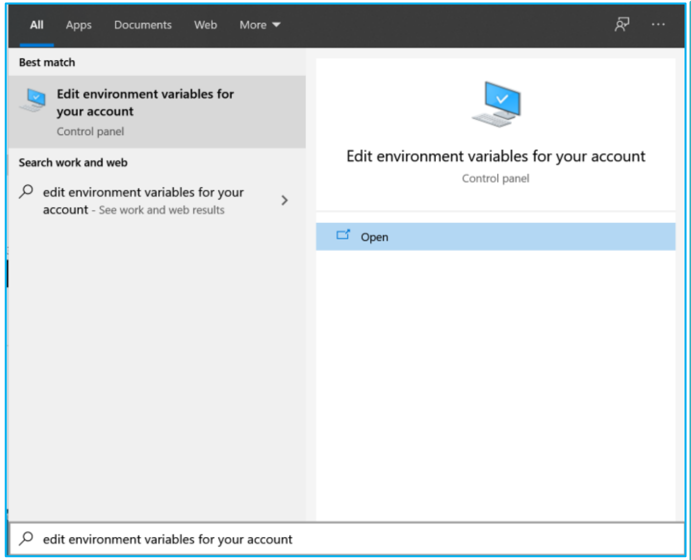
OpenBSD
32-bit | 64-bit

Solaris
64-bit

Windows
32-bit | 64-bit

Save .zip file

INFRASTRUCTURE AS CODE. Terraform. Windows

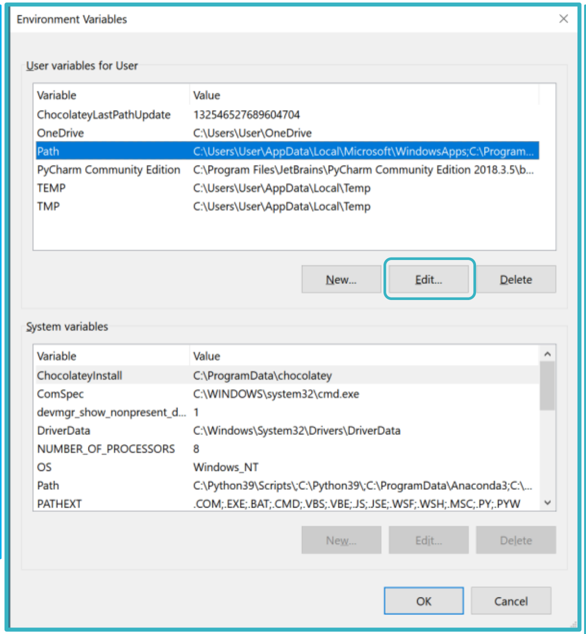


Edit environment variables for your account
Control panel

edit environment variables for your account - See work and web results

Edit environment variables for your account
Control panel

edit environment variables for your account

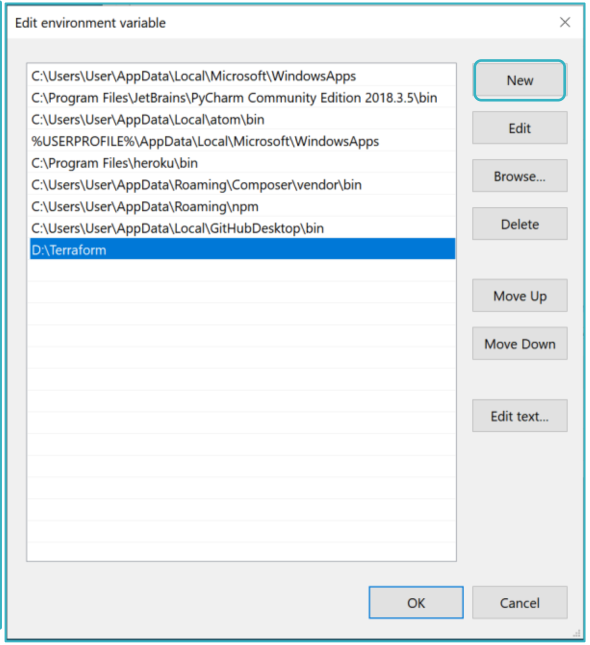


User variables for User

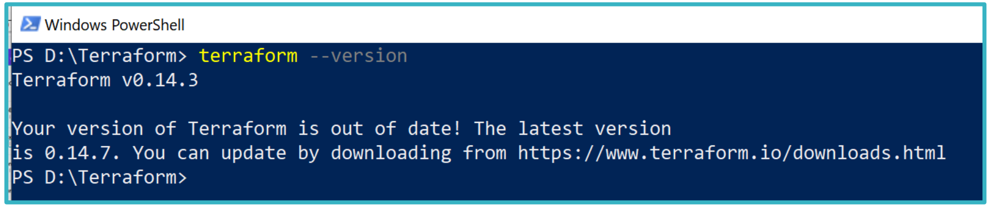
Variable	Value
ChocolateyLastPathUpdate	132546527689604704
OneDrive	C:\Users\User\OneDrive
Path	C:\Users\User\AppData\Local\Microsoft\WindowsApps;C:\Program...
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2018.3.5\b...
TEMP	C:\Users\User\AppData\Local\Temp
TMP	C:\Users\User\AppData\Local\Temp

System variables

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
devmgr_show_nonpresent_d...	1
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Python39\Scripts\;C:\Python39\;C:\ProgramData\Anaconda3;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW

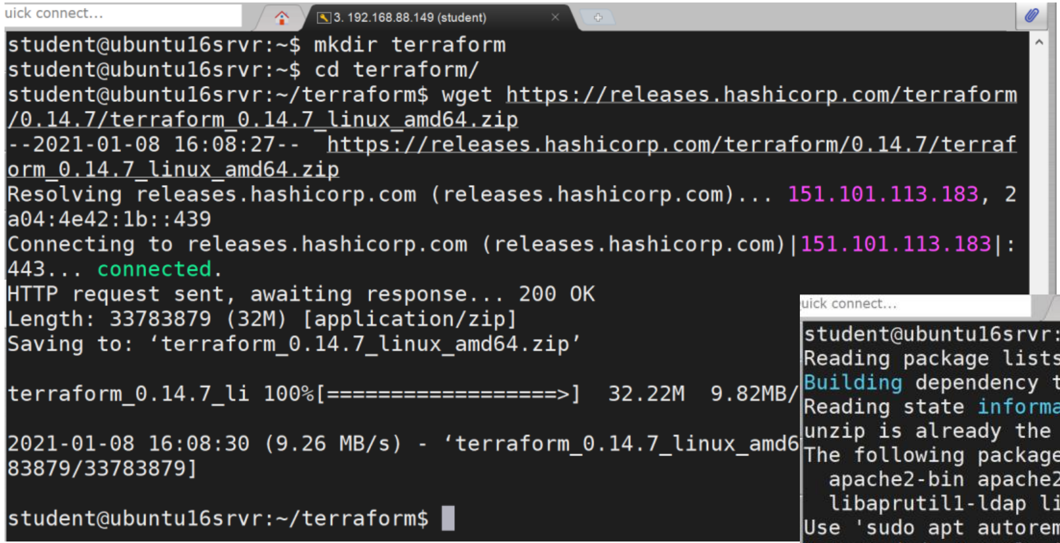


C:\Users\User\AppData\Local\Microsoft\WindowsApps
C:\Program Files\JetBrains\PyCharm Community Edition 2018.3.5\bin
C:\Users\User\AppData\Local\atom\bin
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\heroku\bin
C:\Users\User\AppData\Roaming\Composer\vendor\bin
C:\Users\User\AppData\Roaming\npm
C:\Users\User\AppData\Local\GitHubDesktop\bin
D:\Terraform

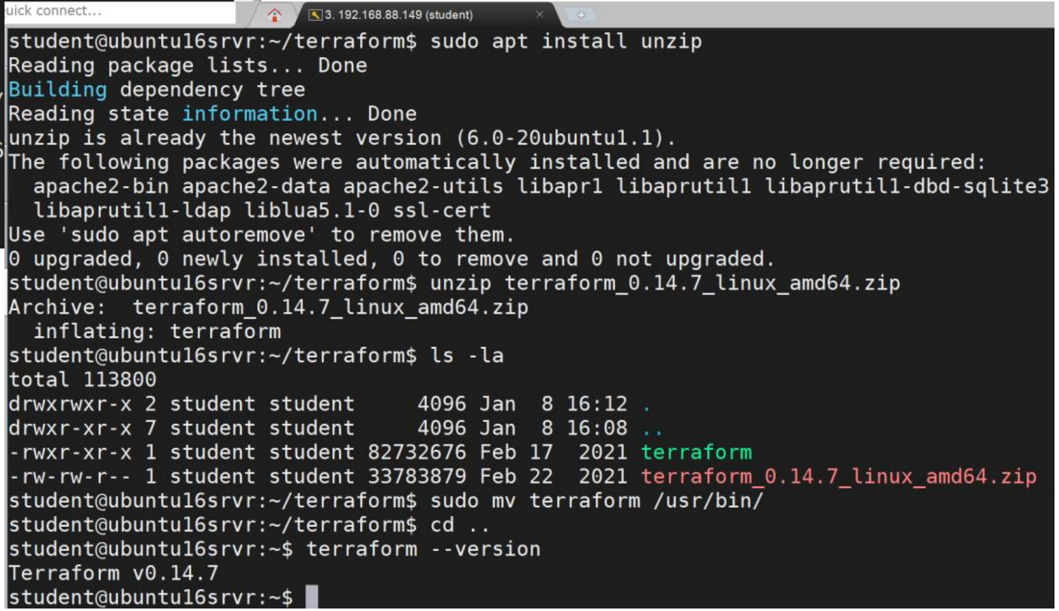


```
PS D:\Terraform> terraform --version
Terraform v0.14.3

Your version of Terraform is out of date! The latest version
is 0.14.7. You can update by downloading from https://www.terraform.io/downloads.html
PS D:\Terraform>
```

INFRASTRUCTURE AS CODE. Terraform. Linux



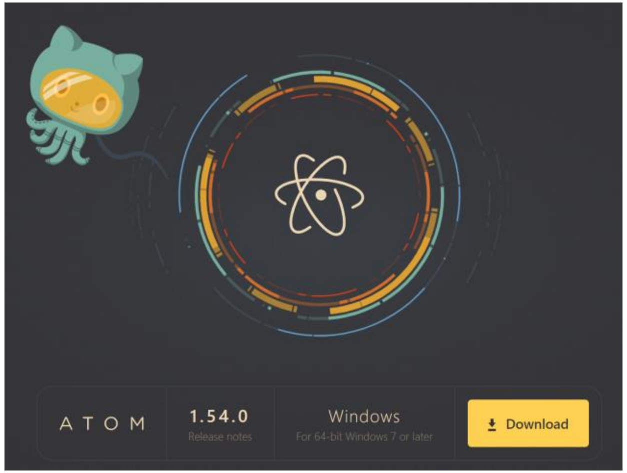
```
student@ubuntu16srvr:~$ mkdir terraform
student@ubuntu16srvr:~$ cd terraform/
student@ubuntu16srvr:~/terraform$ wget https://releases.hashicorp.com/terraform
/0.14.7/terraform_0.14.7_linux_amd64.zip
--2021-01-08 16:08:27--  https://releases.hashicorp.com/terraform/0.14.7/terraf
orm_0.14.7_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 151.101.113.183, 2
a04:4e42:1b::439
Connecting to releases.hashicorp.com (releases.hashicorp.com)|151.101.113.183|:
443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33783879 (32M) [application/zip]
Saving to: ‘terraform_0.14.7_linux_amd64.zip’

terraform_0.14.7_li 100%[==================>]  32.22M  9.82MB/

2021-01-08 16:08:30 (9.26 MB/s) - ‘terraform_0.14.7_linux_amd6
83879/33783879]

student@ubuntu16srvr:~/terraform$ 
```



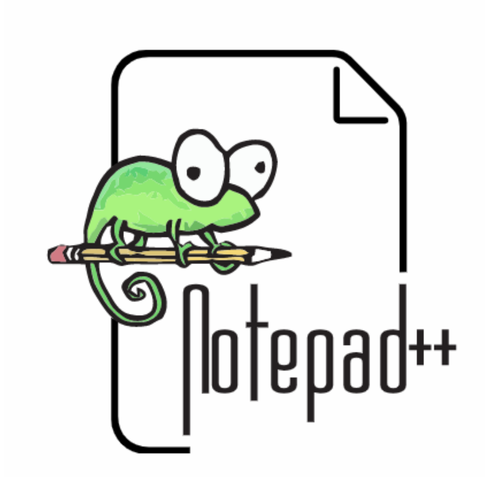
```
student@ubuntu16srvr:~/terraform$ sudo apt install unzip
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-20ubuntu1.1).
The following packages were automatically installed and are no longer required:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.1-0 ssl-cert
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
student@ubuntu16srvr:~/terraform$ unzip terraform_0.14.7_linux_amd64.zip
Archive:  terraform_0.14.7_linux_amd64.zip
  inflating: terraform
student@ubuntu16srvr:~/terraform$ ls -la
total 113800
drwxrwxr-x 2 student student     4096 Jan  8 16:12 .
drwxr-xr-x 7 student student     4096 Jan  8 16:08 ..
-rwxr-xr-x 1 student student 82732676 Feb 17  2021 terraform
-rw-rw-r-- 1 student student 33783879 Feb 22  2021 terraform_0.14.7_linux_amd64.zip
student@ubuntu16srvr:~/terraform$ sudo mv terraform /usr/bin/
student@ubuntu16srvr:~/terraform$ cd ..
student@ubuntu16srvr:~$ terraform --version
Terraform v0.14.7
student@ubuntu16srvr:~$ 
```

INFRASTRUCTURE AS CODE. Terraform. Code (Text) Editors

https://flight-manual.atom.io/getting-started/sections/installing-atom/



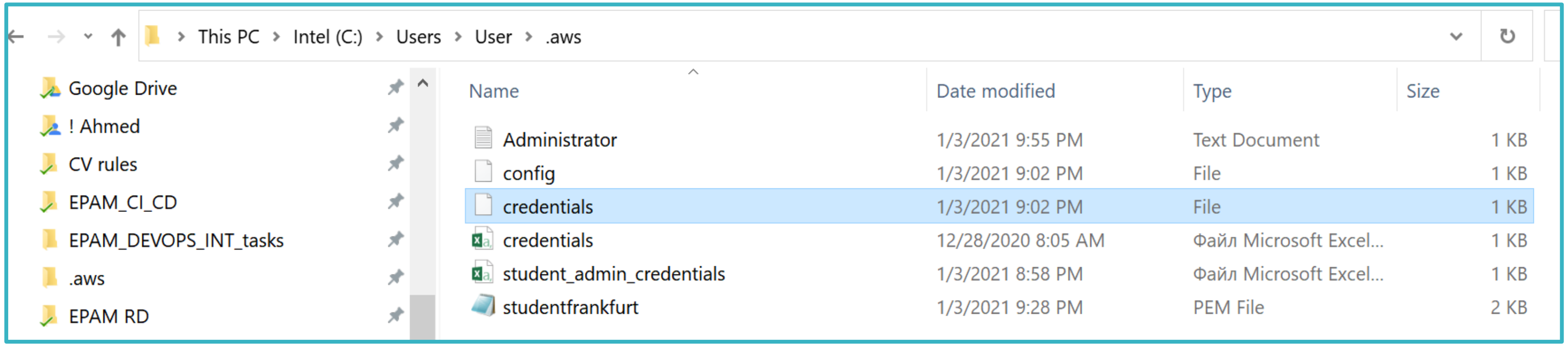
https://notepad-plus-plus.org/downloads/



+ Plugins (Frequently used)

INFRASTRUCTURE AS CODE. Terraform. First steps

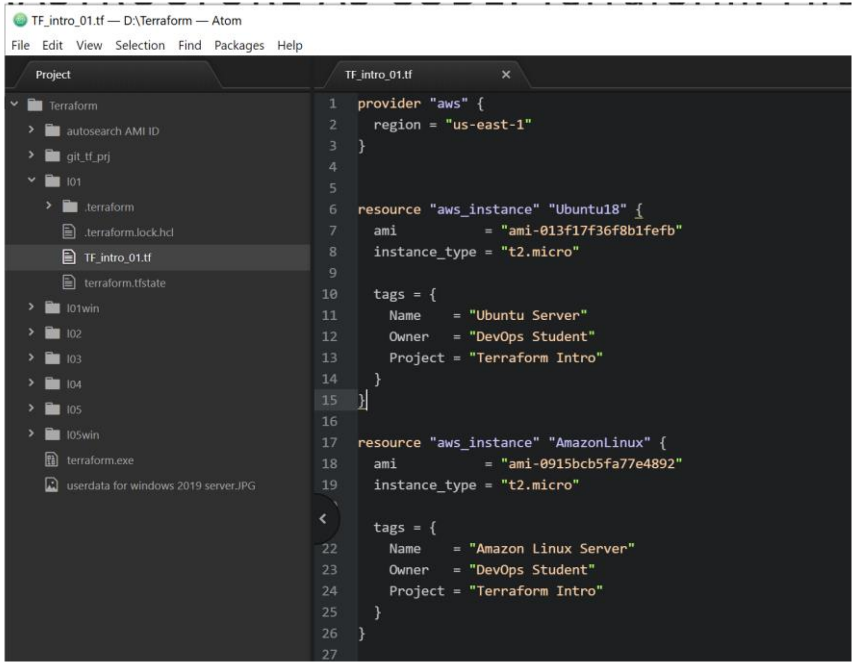
1. Create “terraform” user IAM in AWS console and give him Admin access, save credentials in following way



Name	Date modified	Type	Size
Administrator	1/3/2021 9:55 PM	Text Document	1 KB
config	1/3/2021 9:02 PM	File	1 KB
credentials	1/3/2021 9:02 PM	File	1 KB
credentials	12/28/2020 8:05 AM	Файл Microsoft Excel...	1 KB
student_admin_credentials	1/3/2021 8:58 PM	Файл Microsoft Excel...	1 KB
studentfrankfurt	1/3/2021 9:28 PM	PEM File	2 KB

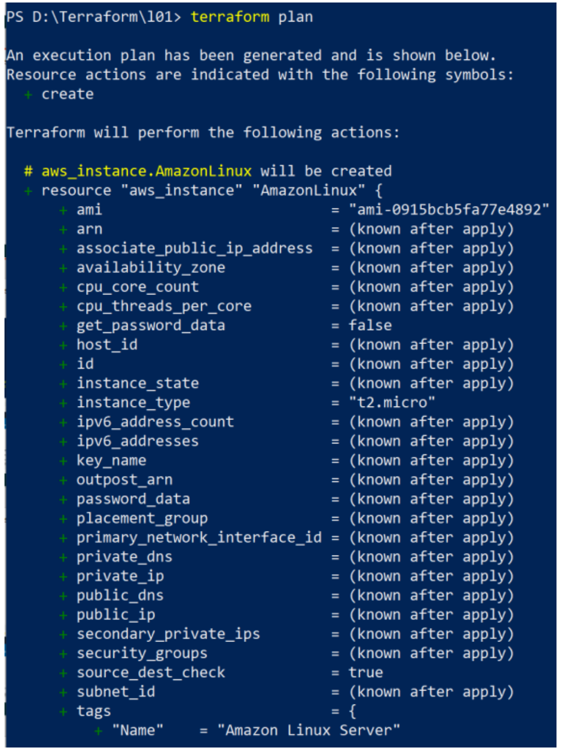
INFRASTRUCTURE AS CODE. Terraform. First steps

Create test.tf file in some home directory



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


resource "aws_instance" "Ubuntu18" {
  ami           = "ami-013f17f36f8b1fefb"
  instance_type = "t2.micro"

  tags = {
    Name    = "Ubuntu Server"
    Owner   = "DevOps Student"
    Project = "Terraform Intro"
  }
}

resource "aws_instance" "AmazonLinux" {
  ami           = "ami-0915bcb5fa77e4892"
  instance_type = "t2.micro"

  tags = {
    Name    = "Amazon Linux Server"
    Owner   = "DevOps Student"
    Project = "Terraform Intro"
  }
}
```

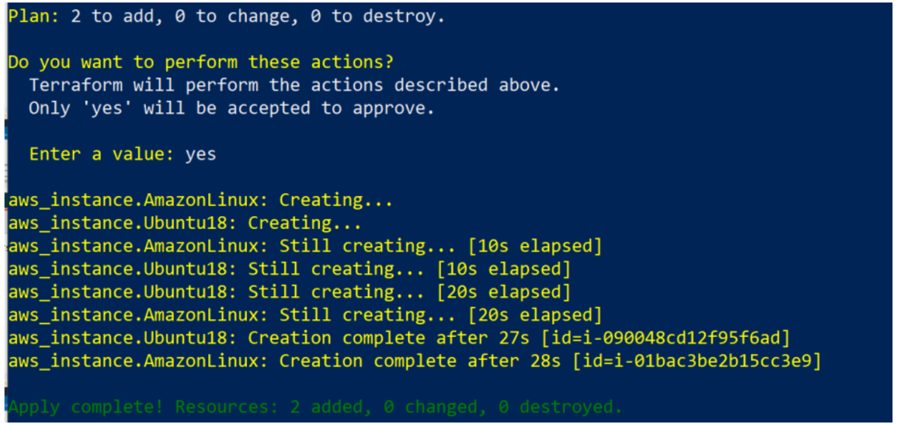
terraform plan



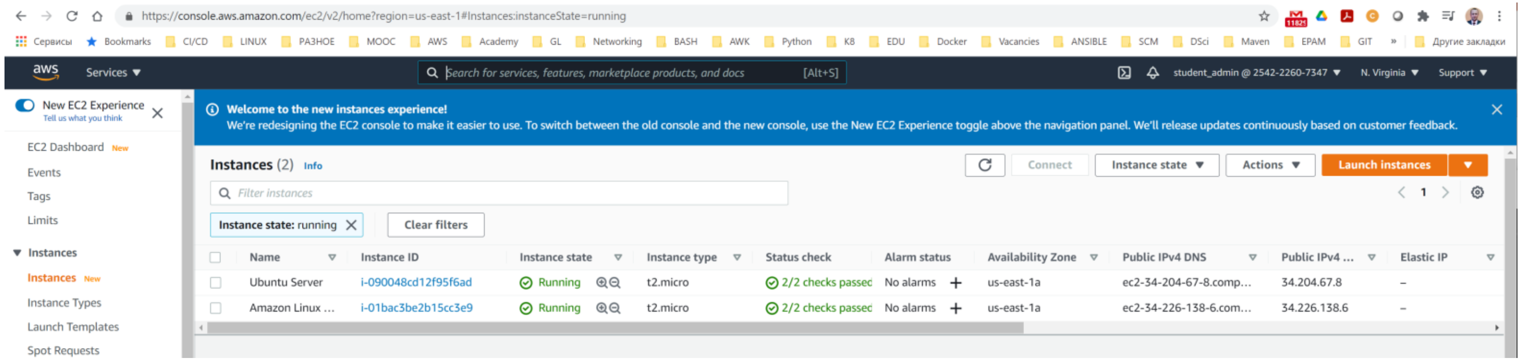
```
PS D:\Terraform\l01> terraform plan

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.AmazonLinux will be created
  + resource "aws_instance" "AmazonLinux" {
      + ami                          = "ami-0915bcb5fa77e4892"
      + arn                          = (known after apply)
      + associate_public_ip_address  = (known after apply)
      + availability_zone            = (known after apply)
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = (known after apply)
      + ipv6_addresses               = (known after apply)
      + key_name                     = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (known after apply)
      + primary_network_interface_id = (known after apply)
      + private_dns                  = (known after apply)
      + private_ip                   = (known after apply)
      + public_dns                   = (known after apply)
      + public_ip                    = (known after apply)
      + secondary_private_ips        = (known after apply)
      + security_groups              = (known after apply)
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name"    = "Amazon Linux Server"
```

INFRASTRUCTURE AS CODE. Terraform. First Steps

terraform apply



```
Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.AmazonLinux: Creating...
aws_instance.Ubuntu18: Creating...
aws_instance.AmazonLinux: Still creating... [10s elapsed]
aws_instance.Ubuntu18: Still creating... [10s elapsed]
aws_instance.Ubuntu18: Still creating... [20s elapsed]
aws_instance.AmazonLinux: Still creating... [20s elapsed]
aws_instance.Ubuntu18: Creation complete after 27s [id=i-090048cd12f95f6ad]
aws_instance.AmazonLinux: Creation complete after 28s [id=i-01bac3be2b15cc3e9]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```



https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:instanceState=running

Welcome to the new instances experience!
We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

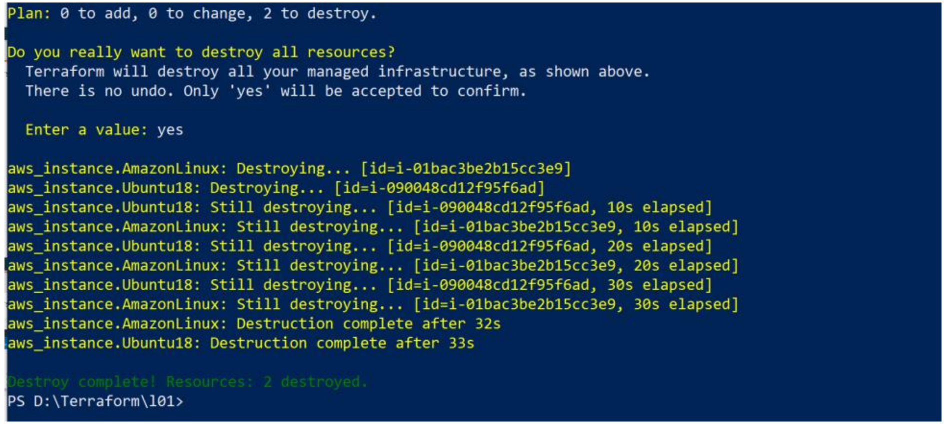
Instances (2) Info

Instance state: running

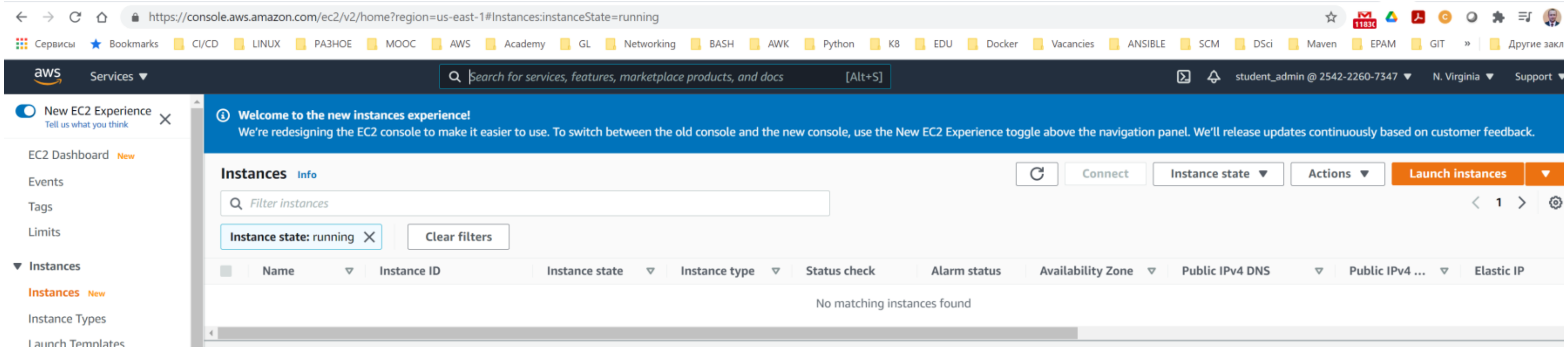
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☐	Ubuntu Server	i-090048cd12f95f6ad	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-204-67-8.comp...	34.204.67.8	–
☐	Amazon Linux ...	i-01bac3be2b15cc3e9	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-226-138-6.com...	34.226.138.6	–

INFRASTRUCTURE AS CODE. Terraform. First Steps

terraform destroy



```
Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.AmazonLinux: Destroying... [id=i-01bac3be2b15cc3e9]
aws_instance.Ubuntu18: Destroying... [id=i-090048cd12f95f6ad]
aws_instance.Ubuntu18: Still destroying... [id=i-090048cd12f95f6ad, 10s elapsed]
aws_instance.AmazonLinux: Still destroying... [id=i-01bac3be2b15cc3e9, 10s elapsed]
aws_instance.Ubuntu18: Still destroying... [id=i-090048cd12f95f6ad, 20s elapsed]
aws_instance.AmazonLinux: Still destroying... [id=i-01bac3be2b15cc3e9, 20s elapsed]
aws_instance.Ubuntu18: Still destroying... [id=i-090048cd12f95f6ad, 30s elapsed]
aws_instance.AmazonLinux: Still destroying... [id=i-01bac3be2b15cc3e9, 30s elapsed]
aws_instance.AmazonLinux: Destruction complete after 32s
aws_instance.Ubuntu18: Destruction complete after 33s

Destroy complete! Resources: 2 destroyed.
PS D:\Terraform\101>
```



https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:instanceState=running

Сервисы Bookmarks CI/CD LINUX PA3HOE MOOC AWS Academy GL Networking BASH AWK Python K8 EDU Docker Vacancies ANSIBLE SCM DSci Maven EPAM GIT » Другие закл

Welcome to the new instances experience!
We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP

No matching instances found

References

https://www.terraform.io/
https://www.terraform.io/docs/language/index.html
https://learn.hashicorp.com/terraform?utm_source=terraform_io
https://learn.hashicorp.com/tutorials/terraform/aws-build?in=terraform/aws-get-started

Q&A

Thank you!