

Implementation guide

Contents

1. Introduction	2
2. Requirements	2
3. Download terraform files from my GitHub	2
4. Set up AWS credentials for Terraform	3
5. Terraform initiation and deployment	4
6. Terraform deletion	6
7. Conclusion	6

1. Introduction

This document will showcase a **step-by-step guide** on how to set up this Terraform project and apply it to your own AWS account. **The** Target group of this document are **IT developers or engineers with some basic experience**. The goal of this document is to make the whole deployment **more clear and simple** for future clients and ensure that everyone can deploy it.

2. Requirements

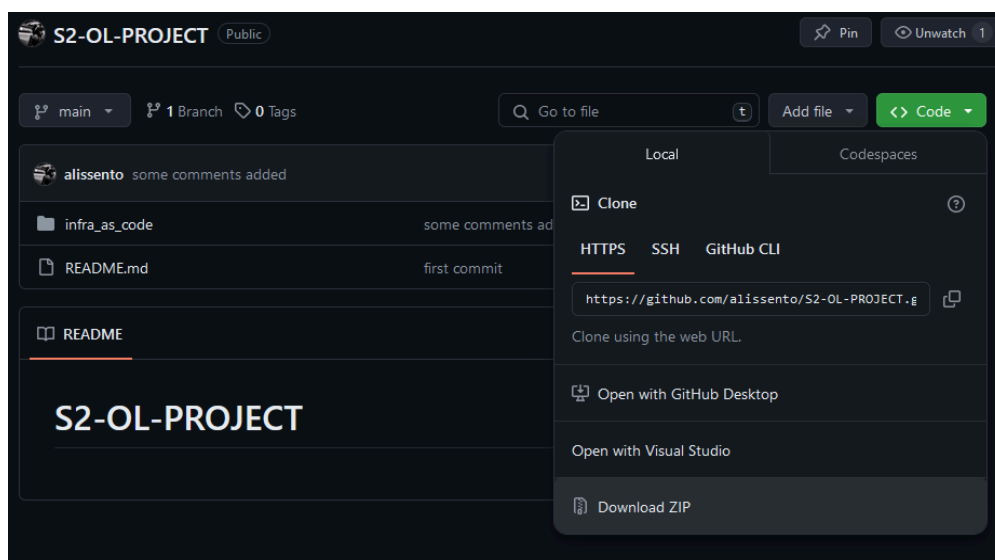
- [Terraform installed](#) - necessary as the whole infrastructure code is in this language - we can deploy everything in a few commands instead of provisioning everything manually through the AWS console.
- [AWS account created](#) - targeted cloud environment of the infrastructure as whole code is written to run it in the AWS
- [Basic terminal knowledge](#) - perform basic commands to navigate to the terraform folder using “cd” and activating all listed terraform commands

3. Download terraform files from my GitHub

- Type “git clone <https://github.com/alissento/S2-OL-PROJECT.git>” in the command line when in the desired folder (git must be already installed)

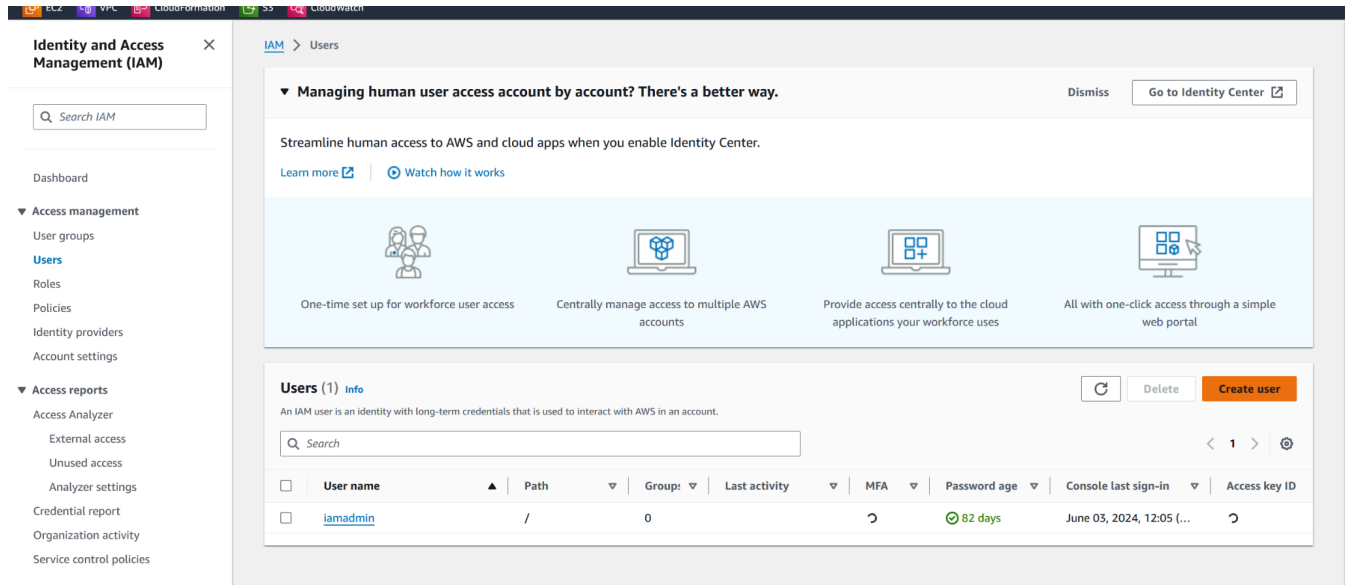
```
PS C:\Users\knezn\TerraformGuide> git clone https://github.com/alissento/S2-OL-PROJECT.git
Cloning into 'S2-OL-PROJECT'...
remote: Enumerating objects: 83, done.
remote: Counting objects: 100% (83/83), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 83 (delta 27), reused 75 (delta 19), pack-reused 0
Receiving objects: 100% (83/83), 14.02 KiB | 7.01 MiB/s, done.
Resolving deltas: 100% (27/27), done.
```

- Or go to <https://github.com/alissento/S2-OL-PROJECT>, click the green button “Code” select “Download zip” and unpack it in the desired location

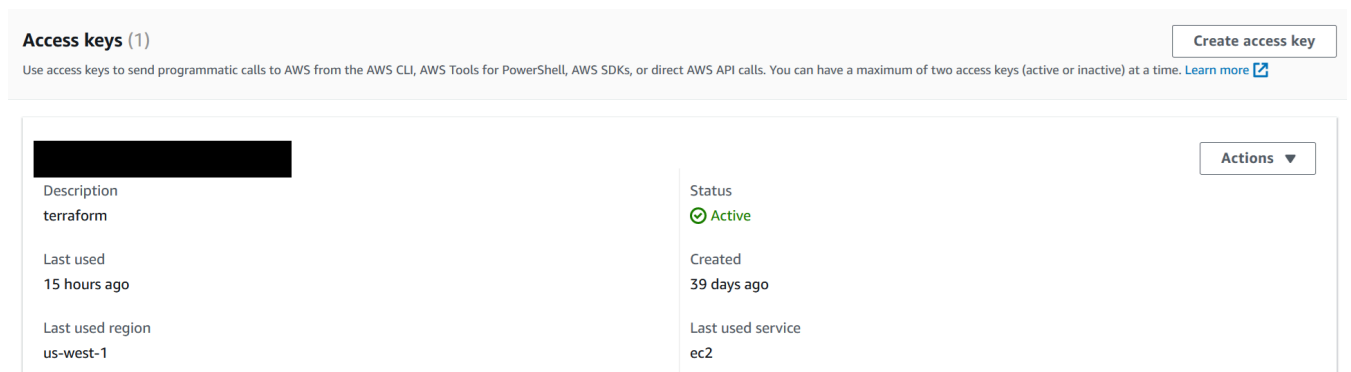


4. Set up AWS credentials for Terraform

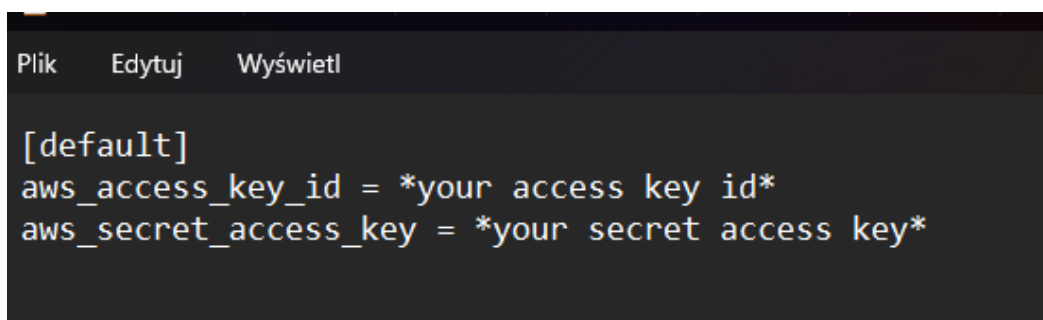
- Go to your AWS account, go to IAM then go to users and click on the already existing user or create a new one and give it the necessary permissions



- Go to your user and click the “Security credentials” tab and in the access keys section create a new access key for Terraform



- Copy these credentials and insert them into `C:\Users\youraccountname\.aws\credentials` file (for Linux and Mac users path is `~/.aws/credentials`) in this format:
[default]
aws_access_key_id = *your access key id*
aws_secret_access_key = *your secret access key*



- Or download AWS CLI (<https://aws.amazon.com/cli/>) and configure it through the “aws configure” command

```
PS C:\Users\knezn> aws configure
AWS Access Key ID [*****]: *your access key id*
AWS Secret Access Key [*****]: *your secret access key*
Default region name [None]: *leave blank*
Default output format [None]: *leave blank*
PS C:\Users\knezn> |
```

5. Terraform initiation and deployment

- Through the command line go to the folder “infra_as_code” that contains terraform files and run the “terraform init” command

```
PS C:\Users\knezn\Desktop\TerraformTutorial\S2-OL-PROJECT\infra_as_code> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/random...
- Installing hashicorp/aws v5.52.0...
- Installed hashicorp/aws v5.52.0 (signed by HashiCorp)
- Installing hashicorp/random v3.6.2...
- Installed hashicorp/random v3.6.2 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

- After all necessary downloads run “terraform plan” and type the desired region to test if everything will deploy successfully

```
PS C:\Users\knezn\Desktop\TerraformTutorial\S2-OL-PROJECT\infra_as_code> terraform plan
var.target_region
  Type a desired AWS region to deploy this project

Enter a value: eu-central-1
```

- If you don't see any errors and you have displayed information “Plan: 42 to add, 0 to change, 0 to destroy.” then it is ready to get deployed!
- Run the command “terraform apply”, type the desired aws region and then type “yes” and terraform will handle the whole infrastructure for you. It will take around 15 minutes to deploy and configure everything

- After successful deployment (Apply complete information in the command line), you should have displayed various outputs in the command line. Copy the "lb_dns" value and paste it into the browser. You should have displayed the WordPress configuration menu. If not give it a couple of minutes and try again.
- The infrastructure is ready so you start configuring your WordPress environment!

Apply complete! Resources: 42 added, 0 changed, 0 destroyed.

Outputs:

```
EC2_AMI = "ami-09e647bf7a368e505"  
first_availability_zone = "eu-central-1a"  
lb_dns = "wordpressALB-2061377929.eu-central-1.elb.amazonaws.com"  
rds_password = <sensitive>  
rds_username = "admin"  
region_name = "eu-central-1"  
second_availability_zone = "eu-central-1b"  
vpc_cidr = "10.10.0.0/21"
```

wordpressalb-2061377929.eu-central-1.elb.amazonaws.com/wp-admin/install.php



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password



Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine
visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

6. Terraform deletion

- If you need to delete the infrastructure from AWS you can type in the command line command “terraform destroy”, then type the region infrastructure is in and then “yes” and terraform will handle everything for you. This also takes around 15 minutes.

```
PS C:\Users\knezn\Desktop\TerraformTutorial\S2-OL-PROJECT\infra_as_code> terraform destroy
var.target_region
Type a desired AWS region to deploy this project

Enter a value: eu-central-1
```

```
Destroy complete! Resources: 42 destroyed.
```

7. Conclusion

Now, you have completed the deployment of the Terraform project on your AWS account. By following this guide, you have experienced how to:

- Set up and configure AWS credentials for Terraform
- Clone the necessary Terraform files from a GitHub repository
- Initialize and apply Terraform configurations to deploy infrastructure
- Destroy the deployed infrastructure when it is no longer needed

This project demonstrates the power and flexibility of Terraform in managing cloud infrastructure as code. It is a very powerful tool as it saves a lot of manual work thanks to its automation