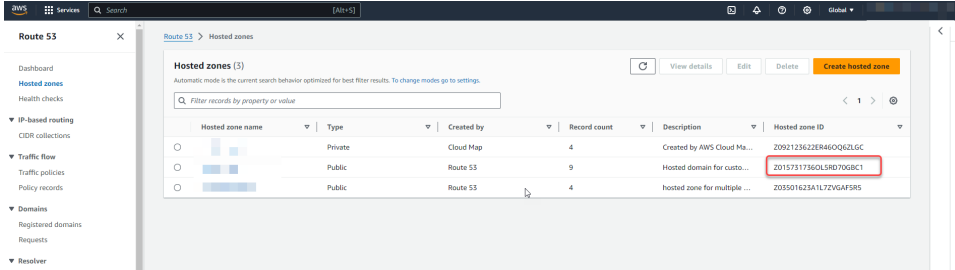


# Ed-Fi AWS ECS Terraform Scripts

This documentation will show you how to deploy an Ed-Fi ODS environment in Shared Instance mode (ODS/WebAPI, AdminApp, Swagger Documentation site) in Amazon Elastic Container Service.

## Requirements:

1. Install [Terraform](#)
2. Configure your [AWS credentials](#) in your console.
3. Visual Studio Code, or any Text Editor, of your choice. In this documentation, we use Visual Studio Code and also installed the HashiCorp Terraform extension (this is not required but it will help visualize the content of the files).
4. Docker Desktop installed and running.
5. Clone the [Ed-Fi-ECS-Starter](#) repository.
6. Valid AWS wildcard [certificate](#). For instance, \*.yourdomain.com. This needs to be a wildcard certificate because it will be used for multiple web applications. For instance: api-dev.yourdomain.com, adminapp-dev.yourdomain.com, and swagger-dev.yourdomain.com.
7. Valid AWS hosted zone identifier.



Open Visual Studio Code (VSC), and navigate to the folder "terraform" of the downloaded project **Ed-Fi-ECS-Starter**.

✓ ED-FI-ECS-STARTER [WSL: UBUNTU-20.04]

> .vscode

> src

✓ terraform

✓ DockerFiles

> DB-Admin

> DB-ODS

> Web-Ods-AdminApp

> Web-Ods-API

> Web-SwaggerUI

data.tf

local.tf

main.tf

provider.tf

terraform.tfvars

variables.tf

Open a New Terminal in the "terraform" folder and execute the following command (see below), this command will prepare your working directory for the commands we will execute later on.

```
terraform init
```

```
○ C:\temp\ECS-AWS>  
● C:\temp\ECS-AWS> terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/null...
- Finding latest version of hashicorp/time...
- Installing hashicorp/aws v5.23.1...
- Installed hashicorp/aws v5.23.1 (signed by HashiCorp)
- Installing hashicorp/null v3.2.1...
- Installed hashicorp/null v3.2.1 (signed by HashiCorp)
- Installing hashicorp/time v0.9.1...
- Installed hashicorp/time v0.9.1 (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.

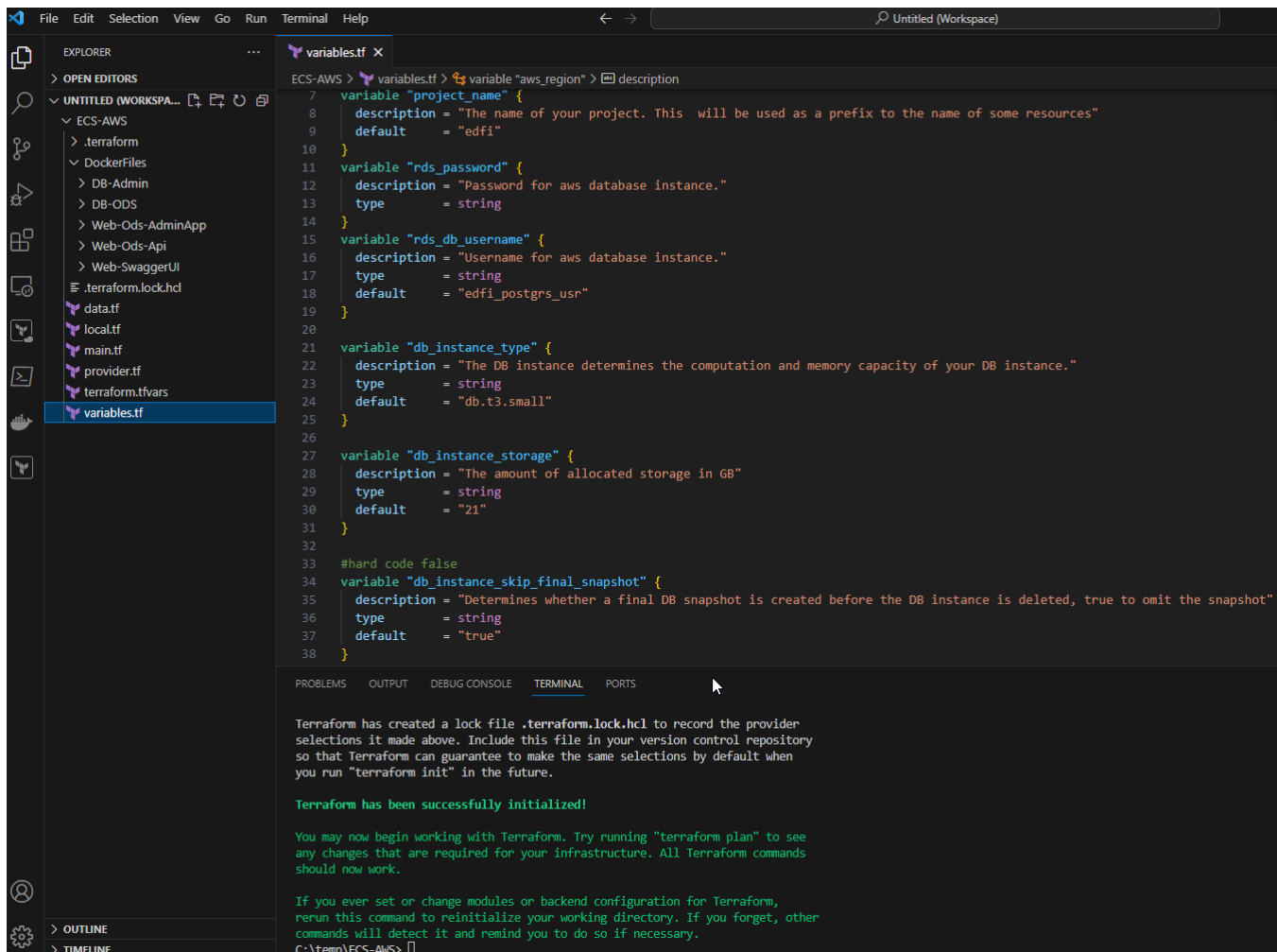
**Terraform has been successfully initialized!**

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.

```
○ C:\temp\ECS-AWS> █
```

Then modify the file "variables.tf" according to your needs.



Please find below all the variables and their description.

## Variables

**aws\_region:** The geographic [area](#) for the installation.

**aws\_maximun\_availability\_zones:** Total geographic areas where for the installation, by default 2.

**project\_name:** The name of your project. This will be used as a prefix to name some resources

**rds\_password:** Password for AWS RDS database instance.

**rds\_db\_username:** Username for AWS RDS database instance.

**db\_instance\_type:** The database instance type determines the capacity(CPU and memory) of your database instance. For instance: "db.t3.small", see list of instance types [here](#)

**db\_instance\_storage:** The amount of allocated storage in GB.

**db\_instance\_skip\_final\_snapshot:** Determines whether a final DB snapshot is created before the DB instance is deleted, true to omit the snapshot

**postgres\_password:** The password for the user postgres.

**cidr:** The IP address range of your AWS VPC.

**docker\_images:** To store a list of Repository names, docker file paths, and log groups that will be used when the docker images are created

**image\_tag:** The version of the docker image to identify it as unique.

**aws\_profile:** To get the credentials from the configured profile in order to run a command against the AWS cloud.

**cpu:** The limit for CPU resources. 1024 CPU units are the equivalent of 1 vCPU, 2048 CPU units is equal to 2 vCPU

**memory:** The limit of memory (in MiB) to present to the task, This value has to be compatible with the CPU.

**app\_environment:** stage example: development or dev, production or prod.

**certificate\_arn:** It is assumed that a certificate exists(in the same **aws\_region**), if not, you need to create a certificate: AWS Certificate Manager >> Certificates >> Request certificate

**ssl\_policy:** The predefined security policies for Load Balancers.

**hosted\_zone\_id:** To route internet traffic for your domain or to route traffic within your VPCs when you use Route 53. It is assumed that a certHosted zone exists, if not, create a Hosted zone: Route 53 >> Hosted zones >> Create hosted zone.

**favorite\_browser:** The browser common you use to open web applications.

**domain:** The name of your domain will be associated with the IP address of the server hosting your applications.

**encryption\_key:**

**white\_ips:** The list of these IPs will be allowed access to the Admin App, example 177.x.x.x/32,188.x.x.x/32

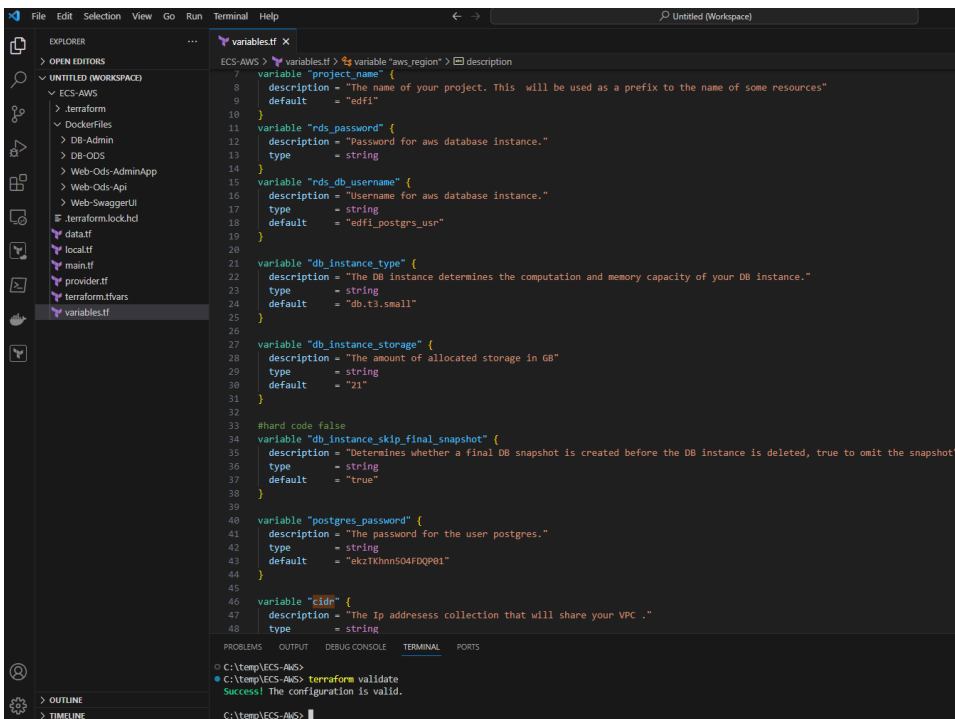
After finishing the variable configuration, run the terraform commands in the next order.

Now that terraform has been initialized and the variables are updated with values of your environment. We can run the following commands:

**\*Note:** to execute the commands, you should navigate to the location where the files were placed, example: ..\Ed-Fi-ECS-Starter\terraform

1. Validate configuration values.

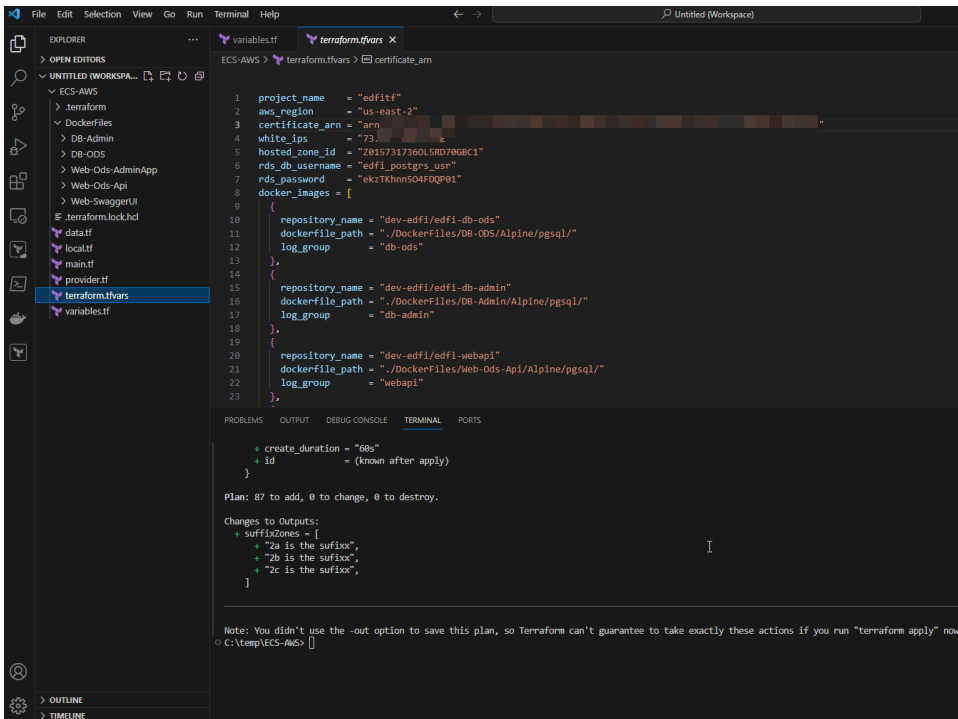
```
terraform validate
```



2. Show changes required by the current configuration.

```
terraform plan
```

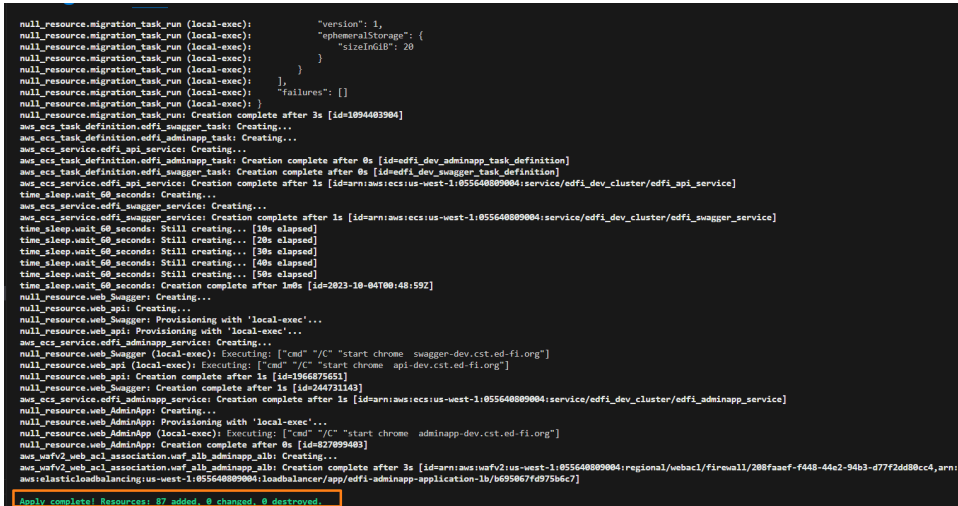
Please see the comment at the end that tells you how many resources are going to be added, changed, or destroyed. In this case, it says 77 resources are going to be added




3. Create or update infrastructure. Please note this command passes the extra option to approve the creation of resources.

```
terraform apply -auto-approve
```

To deploy the resources, you need to execute the "apply" command. This will execute the actions proposed in a Terraform plan. Please note that this process may take several minutes to complete.




If everything executed successfully, you will be able to access your Ed-Fi ODS/API, AdminApp, and Swagger applications.

← → ↻  api-dev.cst.ed-fi.org

```
{
  "version": "6.1",
  "informationalVersion": "6.1",
  "suite": "3",
  "build": "6.1.953.0",
  "apiMode": "Shared Instance",
  "dataModels": [
    {
      "name": "Ed-Fi",
      "version": "4.0.0",
      "informationalVersion": "The Ed-Fi Data Model 4.0"
    }
  ],
  "urls": {
    "dependencies": "https://api-dev.cst.ed-fi.org/metadata/data/v3/dependencies",
    "openApiMetadata": "https://api-dev.cst.ed-fi.org/metadata/",
    "oauth": "https://api-dev.cst.ed-fi.org/oauth/token",
    "dataManagementApi": "https://api-dev.cst.ed-fi.org/data/v3/",
    "xsdMetadata": "https://api-dev.cst.ed-fi.org/metadata/xsd",
    "changeQueries": "https://api-dev.cst.ed-fi.org/changeQueries/v1/",
    "composites": "https://api-dev.cst.ed-fi.org/composites/v1/"
  }
}
```

adminapp-dev.cst.ed-fi.org/identity/Login?ReturnUrl=%2FSetup%2FPostUpdateSetup

 **Ed-Fi ODS Admin App**

Log In



Email:


Password:

☐ Remember me?

[Sign In](#)

[Register as a new user](#)

→   swagger-dev.cst.ed-fi.org



## ODS / API Documentation v6.1

The Ed-Fi ODS / API is divided into several areas by function.

### Resources

Resources are the primary entities that most API client applications work with on a regular basis. Students, staff, education organizations, and their related entities are maintained using this area of the API

[Descriptors](#)  
[Resources](#)

### Composites

Once you have your environment up and running, If you want to remove all the created infrastructure, you just need to run the following command:

```
terraform destroy
```

## Known Issues:

After the script finishes deploying the environment, the AdminApp displays the error message "503 Service Temporarily Unavailable". This occurs because the AdminApp requires a little more time to run scripts and become fully operational. Simply wait a few minutes, then reload the page, and the AdminApp should be up and running.

<https://adminapp-dev.cst.ed-fi.org>

## 503 Service Temporarily Unavailable

## Conceptual diagram:

