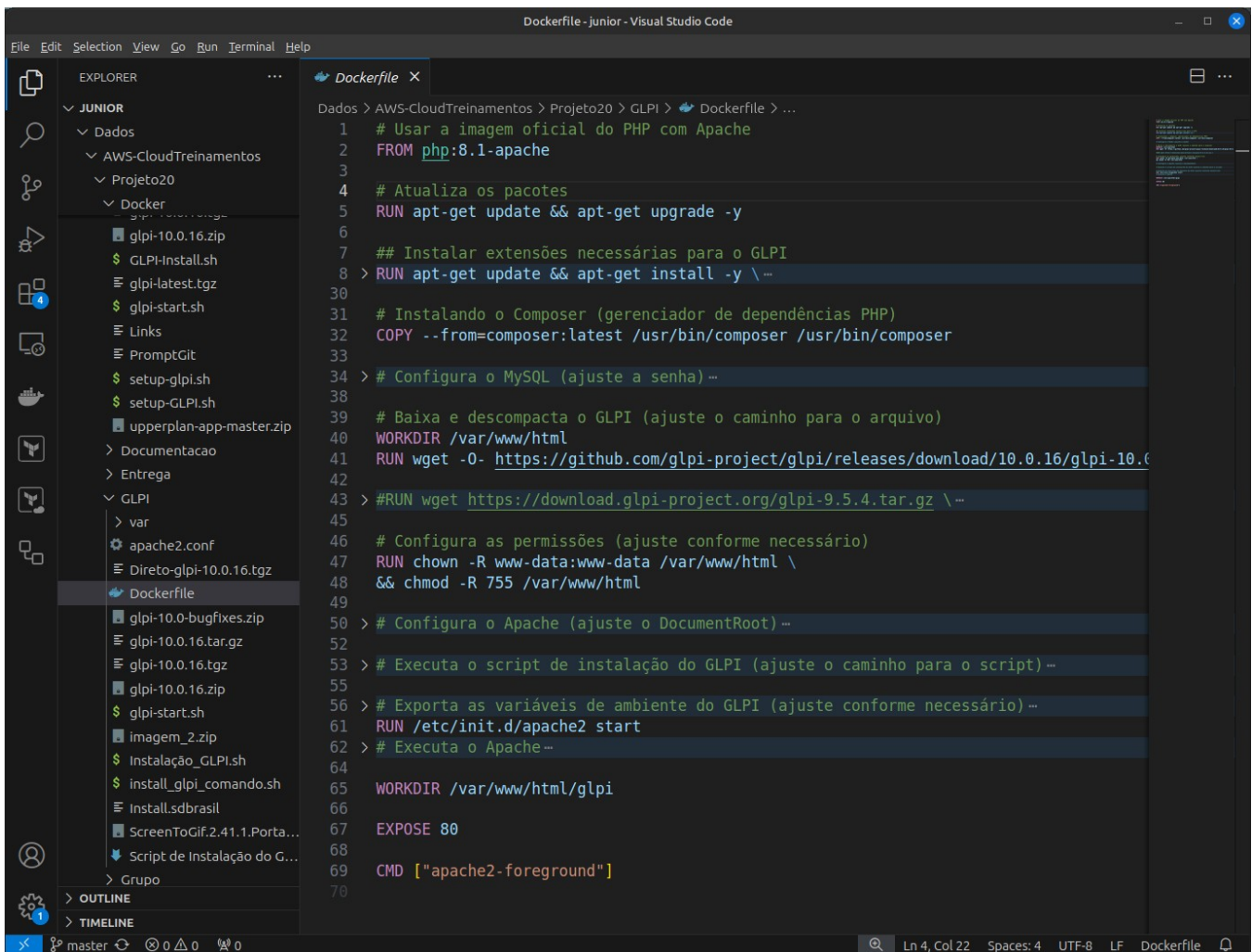


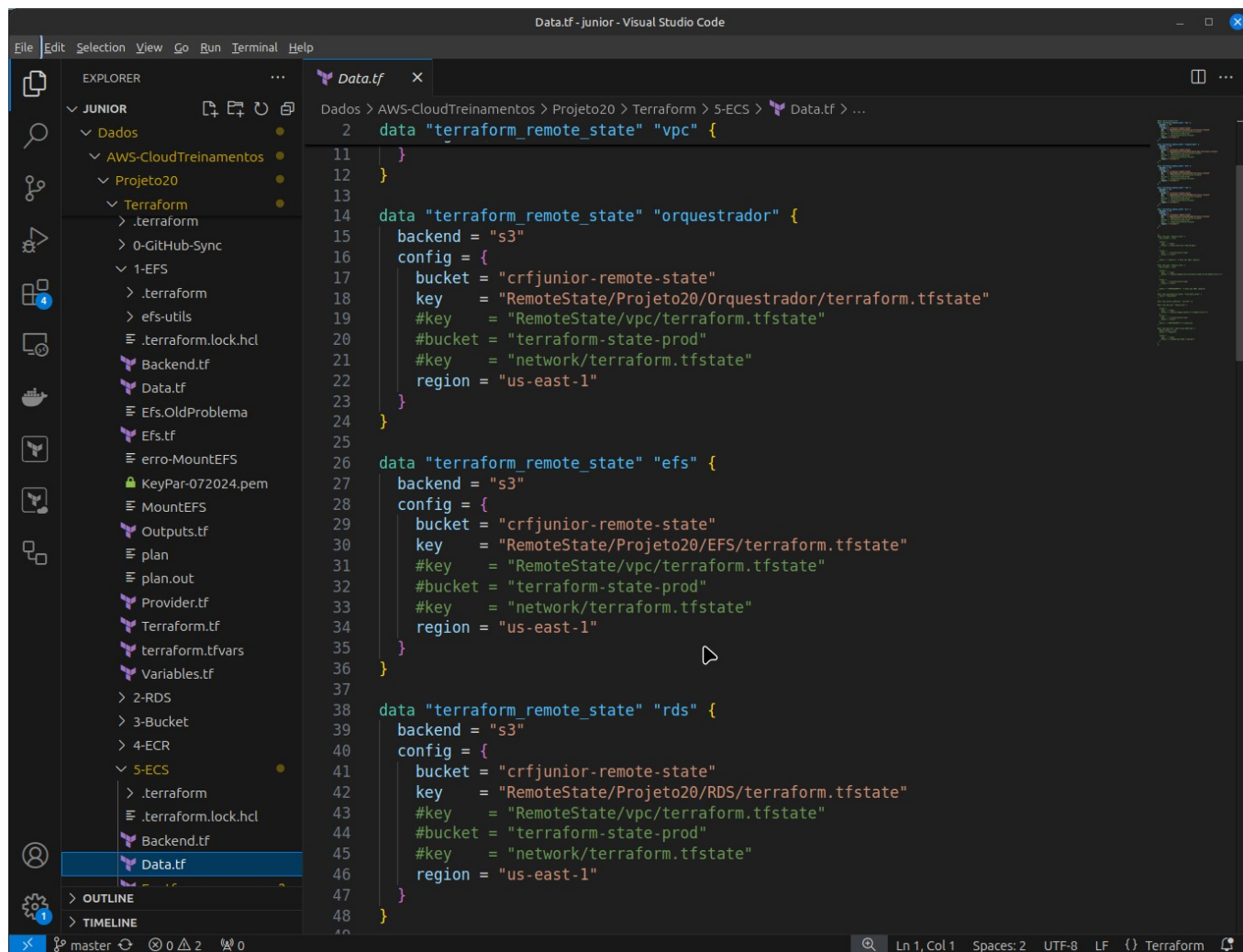
Projeto 20 – Subir GLPI, em uma estrutura em alta disponibilidade e escalavel de baixo Custo Usando a AWS



```
Dados > AWS-CloudTreinamentos > Projeto20 > GLPI > Dockerfile > ...
1  # Usar a imagem oficial do PHP com Apache
2  FROM php:8.1-apache
3
4  # Atualiza os pacotes
5  RUN apt-get update && apt-get upgrade -y
6
7  ## Instalar extensões necessárias para o GLPI
8  > RUN apt-get update && apt-get install -y \--
30
31 # Instalando o Composer (gerenciador de dependências PHP)
32 COPY --from=composer:latest /usr/bin/composer /usr/bin/composer
33
34 > # Configura o MySQL (ajuste a senha) --
38
39 # Baixa e descompacta o GLPI (ajuste o caminho para o arquivo)
40 WORKDIR /var/www/html
41 RUN wget -O- https://github.com/glpi-project/glpi/releases/download/10.0.16/glpi-10.0.16.tar.gz && tar -xzf -
42
43 > # RUN wget https://download.glpi-project.org/glpi-9.5.4.tar.gz \--
44
45 # Configura as permissões (ajuste conforme necessário)
46 RUN chown -R www-data:www-data /var/www/html \
47 && chmod -R 755 /var/www/html
48
49 > # Configura o Apache (ajuste o DocumentRoot) --
52
53 > # Executa o script de instalação do GLPI (ajuste o caminho para o script) --
54
55 > # Exporta as variáveis de ambiente do GLPI (ajuste conforme necessário) --
56
57 RUN /etc/init.d/apache2 start
58
59 > # Executa o Apache --
62
63 WORKDIR /var/www/html/glpi
64
65 EXPOSE 80
66
67 CMD ["apache2-foreground"]
70
```

---

Projeto 20 – Subir GLPI, em uma estrutura em alta disponibilidade e escalavel de baixo Custo Usando a AWS



The image shows a Visual Studio Code editor window titled "Data.tf - junior - Visual Studio Code". The Explorer sidebar on the left displays a project structure under "JUNIOR":

- Dados
  - AWS-CloudTreinamentos
    - Projeto20
      - Terraform
        - .terraform
        - 0-GitHub-Sync
        - 1-EFS
          - .terraform
          - efs-utils
          - .terraform.lock.hcl
          - Backend.tf
          - Data.tf
          - Efs.OldProblema
          - Efs.tf
          - erro-MountEFS
          - KeyPar-072024.pem
          - MountEFS
          - Outputs.tf
          - plan
          - plan.out
          - Provider.tf
          - Terraform.tf
          - terraform.tfvars
          - Variables.tf
        - 2-RDS
        - 3-Bucket
        - 4-ECR
        - 5-ECS
          - .terraform
          - .terraform.lock.hcl
          - Backend.tf
          - Data.tf

The main editor displays the content of "Data.tf":

```
2 data "terraform_remote_state" "vpc" {
11 }
12 }
13
14 data "terraform_remote_state" "orquestrador" {
15   backend = "s3"
16   config = {
17     bucket = "crfjunior-remote-state"
18     key     = "RemoteState/Projeto20/0rquestrador/terraform.tfstate"
19     #key    = "RemoteState/vpc/terraform.tfstate"
20     #bucket = "terraform-state-prod"
21     #key    = "network/terraform.tfstate"
22     region = "us-east-1"
23   }
24 }
25
26 data "terraform_remote_state" "efs" {
27   backend = "s3"
28   config = {
29     bucket = "crfjunior-remote-state"
30     key     = "RemoteState/Projeto20/EFS/terraform.tfstate"
31     #key    = "RemoteState/vpc/terraform.tfstate"
32     #bucket = "terraform-state-prod"
33     #key    = "network/terraform.tfstate"
34     region = "us-east-1"
35   }
36 }
37
38 data "terraform_remote_state" "rds" {
39   backend = "s3"
40   config = {
41     bucket = "crfjunior-remote-state"
42     key     = "RemoteState/Projeto20/RDS/terraform.tfstate"
43     #key    = "RemoteState/vpc/terraform.tfstate"
44     #bucket = "terraform-state-prod"
45     #key    = "network/terraform.tfstate"
46     region = "us-east-1"
47   }
48 }
49 }
```

The status bar at the bottom indicates "Ln 1, Col 1", "Spaces: 2", "UTF-8", "LF", and "Terraform".

---

Projeto 20 – Subir GLPI, em uma estrutura em alta disponibilidade e escalavel de baixo Custo  
Usando a AWS

The screenshot shows the Visual Studio Code interface with a Terraform configuration file named `Ecs.tf` open. The Explorer sidebar on the left shows a project structure with folders like `JUNIOR`, `Dados`, `AWS-CloudTreinamentos`, `Projeto20`, and `Terraform`. The `Ecs.tf` file is selected under the `5-ECS` folder. The main editor displays the Terraform code for an `aws_ecs_task_definition` resource named `glpi_task_definition`. The code includes properties for family, network mode, requires compatibility, CPU, memory, execution role, and task role. It also defines container definitions for a container named `container_glpi`, including image, essential status, port mappings, and environment variables for database connection details. The status bar at the bottom indicates the file is on line 1, column 1, with 2 spaces, UTF-8 encoding, and LF line endings.

```
1 #Task Definition
2 resource "aws_ecs_task_definition" "glpi_task_definition" {
3     family            = "TaskDefinition_glpi"
4     network_mode      = "awsvpc"
5     requires_compatibilities = ["FARGATE"]
6     cpu               = "2048"
7     memory            = "4096"
8     #essential = true
9     execution_role_arn = aws_iam_role.ecs_execution_role.arn
10    task_role_arn      = aws_iam_role.ecs_task_role.arn
11
12    container_definitions = jsonencode([
13        {
14            name = "container_glpi"
15            image = "${data.terraform_remote_state.ecr.outputs.ecr_repository_arn}:latest"
16            #654654346517.dkr.ecr.us-east-1.amazonaws.com/repository_glpi:latest # URL da s
17            essential = true
18            portMappings = [
19                {
20                    containerPort = 80
21                    hostPort      = 80
22                }
23            ]
24            environment = [
25                {
26                    name = "DB_HOST"
27                    value = "${data.terraform_remote_state.rds.outputs.db_end_point}" #aws_db_in
28                },
29                {
30                    name = "DB_NAME"
31                    value = "${data.terraform_remote_state.rds.outputs.db_db_name}"
32                },
33                {
34                    name = "DB_USER"
35                    value = "${data.terraform_remote_state.rds.outputs.db_db_username}"
36                },
37                {
38                    name = "DB_PASSWORD"
39                    value = "glpiglpi"
40                }
41            ]
42        }
43    ])
44    mountPoints = [
```

---

Projeto 20 – Subir GLPI, em uma estrutura em alta disponibilidade e escalavel de baixo Custo  
Usando a AWS