Documentation
User Story 4: Setup a web server

Description:

This guide is for a Macbook user from Neue Fische.

Condition: User Story 1, 2 and 3 are completed.

Code: https://github.com/Rami2028/Team1/blob/main/EC2.tf

Task 1: Request latest AMI-Id for the respective region

We are requesting the AMI id that our EC2 instance should have. The AMI should be the latest version of the Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type. There are 2 ways to find this Id.
- Preferred: Request one in the code
- or manually look it up in the AWS Management Console

The code to request the AMI-Id is like this:

```
#Select newest AMI-id
data "aws_ami" "latest_amazon_linux" {
most_recent = true

filter {
name = "name"
values = ["amzn2-ami-hvm-*-x86_64-gp2"]
}

filter {
name = "virtualization-type"
values = ["hvm"]
}

owners = ["amazon"] # Amazon
}
```

The code will request the latest AMI (most recent). In the filter we request for an Amazon Linux 2 AMI (HVM).
If you want to search for the AMI ID in the AWS Management Console, follow these steps:

- AWS Management Console login
- Open the Service EC2
- Open in the left navigation bar: AMI Catalog

- Search for: ‘Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type’
- Copy the first ami-id (something like this): ’ami-0ea832bf7873542df’

Task 2: Create a web server using AWS EC2 service with the attributes like Instance Type, Keypair, Storage, etc., and use the script from User story 3 for the user data section.

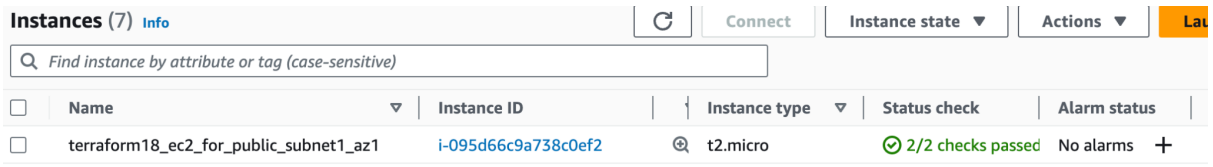
You can create an EC2 with this code. In the code you use the latest AMI-Id. When you copied the AMI Id in the AWS Management Console you can paste it in the code with “. So like this: ami =“’ami-0ea832bf7873542df”
Then you can choose the instance type (like t2.micro), the key:name (“vockey”), your subnet and your security group. When you want to use wordpress you should add user-data with the script of User Story 3.
You can add the provioner code.snippet. This will create a file allinstancedetails with Instance Type, Instance Id, Public DNS and AMI Id.

```
###Create EC2
resource "aws_instance" "deham6demo"{
ami = data.aws_ami.latest_amazon_linux.id
instance_type = "t2.micro"
key_name = "vockey"
vpc_security_group_ids = [aws_security_group.devVPC_sg_allow_ssh_http.id]
subnet_id = aws_subnet.devVPC_public_subnet1.id
tags = {
Name = "terraform18_ec2_for_public_subnet1_az1"
}
user_data = file("user-data.sh")
provisioner "local-exec"{
command = "echo Instance Type=${self.instance_type},Instance ID=${self.id},Public
DNS=${self.public_dns},AMI ID=${self.ami} >> allinstancedetails"
}
}
```

You need to customize the code to make it work for you:
- replace the security group with your own
- replace thesubnet with your own
- replace the security group with your own

Verify Solution:

When you run the code, you can see your instance in the EC2 service in the AWS Management Console.



If you select your instance, copy the public IP and paste it into the browser, you will need to see it: