**CSCL1030 Applied Lab 1 - Managing AWS Infrastructure with Terraform**

**A black and red text

AI-generated content may be incorrect.**

**Student Name: Vinod Kumar Dhanavath  
Course: CSCL1030 –CloudOps Tools and Techniques  
Instructor: Joba Hassan  
Date: 26-03-2025**

**Lab Overview:**

The purpose of this lab was to use Terraform to create an AWS EC2 instance that hosts a basic web server. The goal was to ensure the infrastructure was correctly provisioned and accessible through a web browser.

**Tools and Technologies:**

* **Terraform (**Terraform v1.5.7) for infrastructure automation
* **AWS CLI (**aws-cli/2.24.23) for managing AWS resources
* **Git & GitHub**for version control (Used to clone the Repository)
* **SSH Keys**for secure server access

1. **Steps to Deploy the Web Server**
2. **Cloning the Repository**

To begin, the Terraform configuration files were retrieved from GitHub. This repository contained the necessary code (.tf files) to provision the AWS infrastructure. The following command was used:

**Commands:**

git clone https://github.com/SMannionYorkU/CloudOps\_Module4\_Class4\_Repo2\_WebServer\_Modularized.git

cd CloudOps\_Module4\_Class4\_Repo2\_WebServer\_Modularized

**A screenshot of a computer

AI-generated content may be incorrect.**

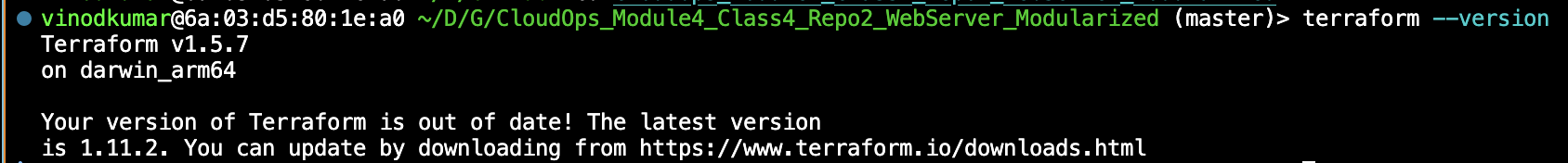
* **Note:** I have cloned the repository and opened the folder in VS Code for easier modifications to the `.tf` files. Now, I will initialize Terraform and proceed with provisioning the AWS infrastructure. Following is the screenshot for the reference.

A computer screen with text and images

AI-generated content may be incorrect.

1. **Initializing Terraform**

**Terraform Version Command and Output** to demonstrate that Terraform was installed and configured.

****

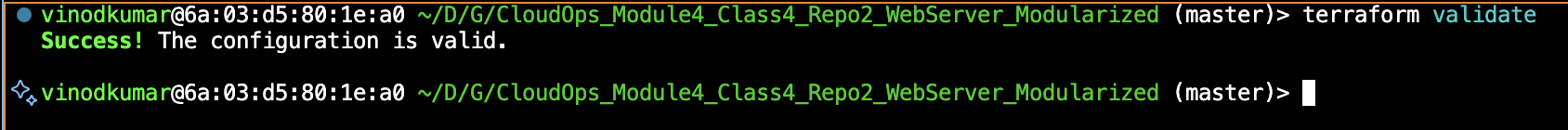
Terraform was initialized to download the required provider plugins and set up the working directory. This step was executed with: **Terraform init .**

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Validating the Configuration**

Before applying the Terraform configuration, the syntax and structure of the files were checked to ensure there were no errors. The validation was performed using: **Terraform validate.**

****

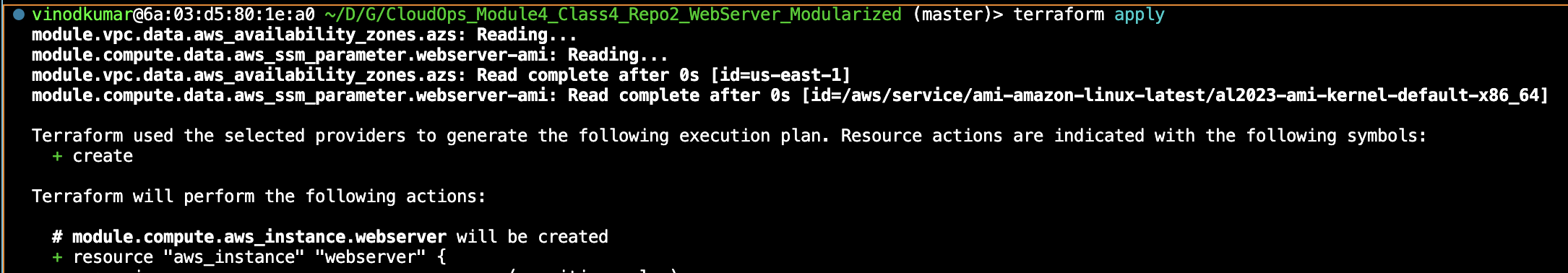
Generated a Terraform execution plan to preview the infrastructure changes before deployment using: **Terraform plan.**

A screenshot of a computer

AI-generated content may be incorrect.

1. **Deploying the Infrastructure**

Terraform was executed to create the required AWS resources, including the EC2 instance using the command: **Terraform apply.**

****

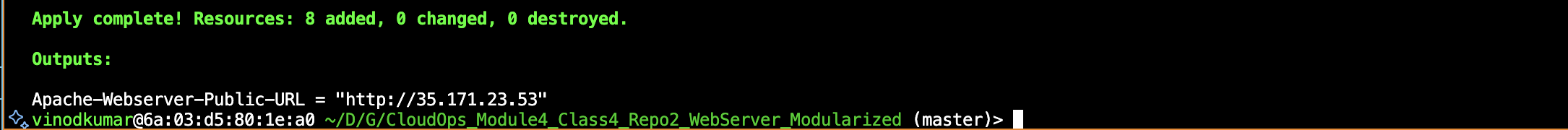
Once, apply command executed, terraform requires confirmation as follows.

Enter the value as **“yes”.**

**A black screen with white text

AI-generated content may be incorrect.**

Upon successful completion, Terraform provided a **Public URL for** the web server:



1. **Verifying the Web Server**
2. **Browser Verification**

The provided Public URL was opened in a web browser to confirm that the web server was running. Following is the output a webpage displaying **"It Works!"**

A screenshot of a computer

AI-generated content may be incorrect.

1. **AWS CLI Verification**

* The following screenshot demonstrates that I have successfully installed and configured the AWS Command Line Interface (CLI) on my system by generating **Access key and ID** from the AWS Console. The output confirms the following:
* AWS CLI is Installed – The version check (aws --version) confirms that AWS CLI version **2.24.23** is installed on a system running macOS (**Darwin/x86\_64).**
* AWS CLI is Configured – The aws configure list command verifies that the CLI is set up with valid AWS credentials, including the access key, secret key, and default region (**us-east-1**).
* AWS CLI Can Query EC2 Instances – The aws ec2 describe-instances command successfully retrieves and displays information about my EC2 instances, showing their Instance IDs, Public IPs, and States. Required Instance id (**i-047dda929ed3a4837)** is running.

**A screen shot of a computer

AI-generated content may be incorrect.**

Executed the command: **ssh -i "webserver.pem" ec2-user@ec2-35-171-23-53.compute-1.amazonaws.com,** to connect the webserver using SSH key from AWS CLI. Following is the screenshot for the reference.

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Confirming AWS Deployment**

The AWS Management Console was accessed to verify that the EC2 instance was running. The instance details, including its public IP address, were reviewed to ensure they matched the Terraform output.

Following the screenshot that displays the instance name: **webserver\_tf** created and running.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Conclusion**

Terraform successfully provisioned an EC2 instance, deployed a web server, and assigned a public IP. The server was accessible via a web browser and verified through AWS CLI. This demonstrated how Infrastructure as Code (IaC) simplifies cloud resource management.

1. **Submitted Deliverables**

* Screenshot of Terraform and AWS CLI versions
* Screenshot of Terraform apply output (Public URL only)
* Screenshot of the web browser displaying "It Works!"
* Screenshot of AWS CLI output confirming the instance is running
* Screenshot of AWS EC2 instance details in the AWS Console

1. **References:**

https://registry.terraform.io/providers/hashicorp/aws/latest

https://learn.continue.yorku.ca/mod/lesson/view.php?id=603345

**Thank you.**