Chat link: <https://chatgpt.com/c/aba0d3f6-9838-425a-bd60-da4e7f9fa917>

Here are the key points extracted from the provided text, organized for clarity:

### Overview

- Objective: Understanding how to log into and automate the creation of virtual machines (VMs) using various tools and methods.

### Key Methods for Logging into a Virtual Machine

1. AWS Console UI

- Steps:

1. Navigate to the EC2 Dashboard in the AWS Console.

2. Select the running instance.

3. Click on the instance ID and then on "Connect."

4. Follow prompts to establish a connection.

- Limitation: This method is less efficient for frequent access and does not retain sessions if you are away from the console.

2. Terminal Access

- Setup: Use a terminal emulator on your local machine to connect to the VM.

- Tools:

- Mac: `iTerm` is recommended for its advanced features and user experience.

- Windows: Options include:

- PuTTY: Widely used but has a basic interface.

- MobaXterm: Offers a better interface and additional features.

- NoMachine: Useful for various remote access needs.

- Linux/Ubuntu: Common options include:

- GNOME Terminal

- Konsole

- Terminator

- Alacritty

- Kitty

- xterm

- rxvt

- tmux and screen for multiplexing.

- Guake and Yakuake for drop-down terminals.

- Termius for modern SSH access.

- Connection Process:

1. Obtain the public IP address of the VM from the AWS Console.

2. Use SSH with the command `ssh -i [path-to-key] ubuntu@[public-ip]`.

3. If connection fails due to permissions, adjust key file permissions with `chmod 600 [path-to-key]`.

### Automating VM Creation

1. AWS API

- Description: Use AWS API calls to automate VM creation by scripting interactions with AWS services.

2. AWS CLI

- Description: Command-line interface for interacting with AWS services, allowing for VM creation and management through terminal commands.

3. AWS CDK (Cloud Development Kit)

- Description: Define infrastructure using code (e.g., TypeScript, Python) to automate resource management and deployment.

4. AWS CloudFormation Templates

- Description: Define and provision AWS infrastructure using declarative templates in JSON or YAML format.

5. Terraform

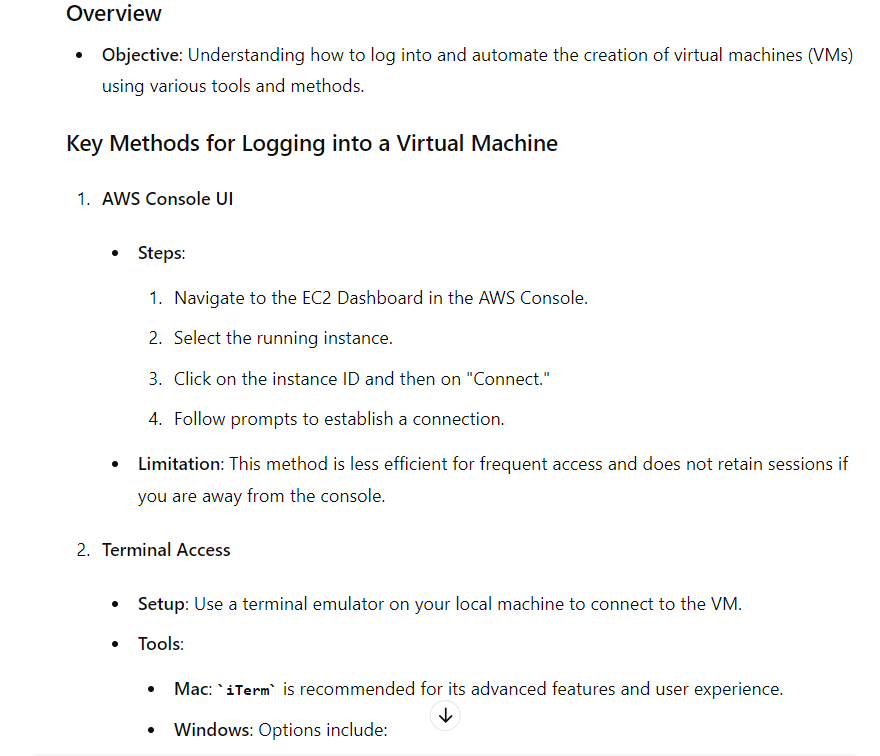
- Description: An open-source tool for provisioning and managing infrastructure across multiple cloud providers with a consistent configuration language.

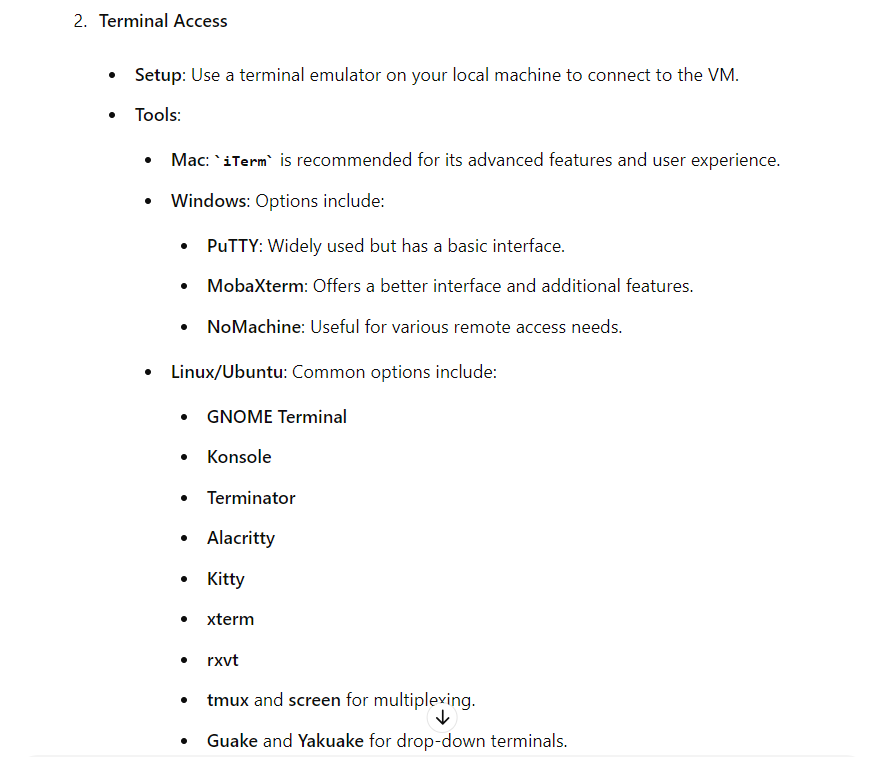
### Summary

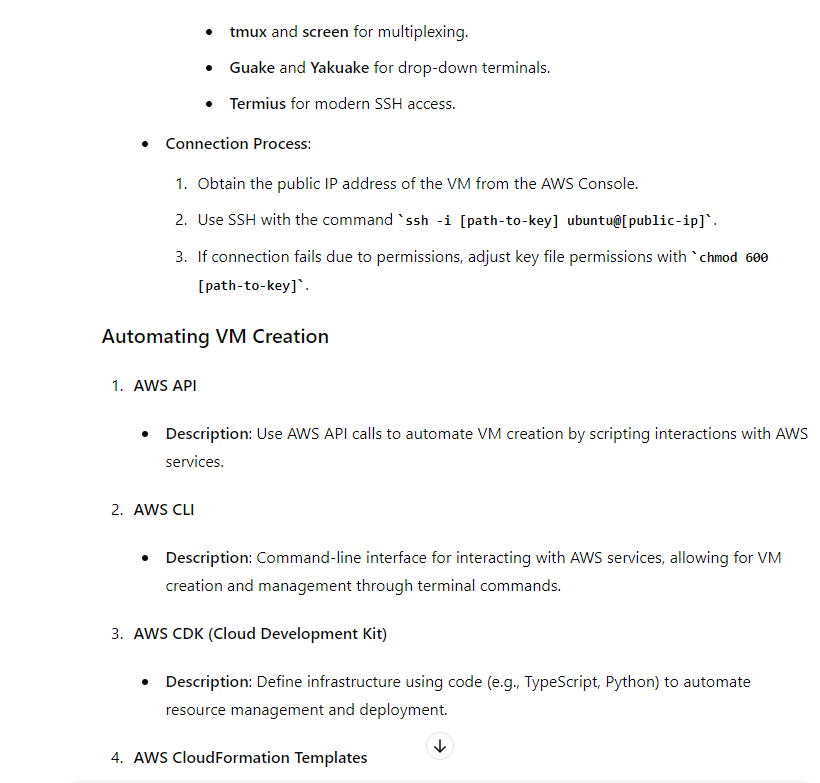
- Login Methods: AWS Console UI is straightforward but less efficient. Terminal access (with tools like `iTerm`, `PuTTY`, or `MobaXterm`) is preferred for frequent tasks.

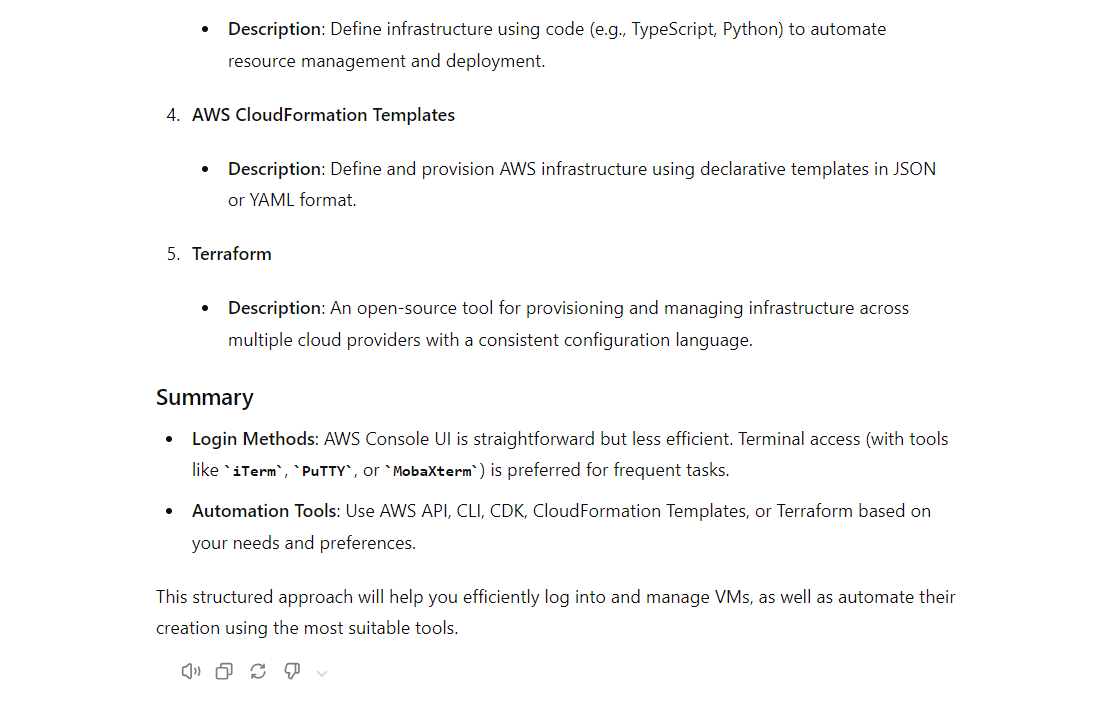
- Automation Tools: Use AWS API, CLI, CDK, CloudFormation Templates, or Terraform based on your needs and preferences.

This structured approach will help you efficiently log into and manage VMs, as well as automate their creation using the most suitable tools.

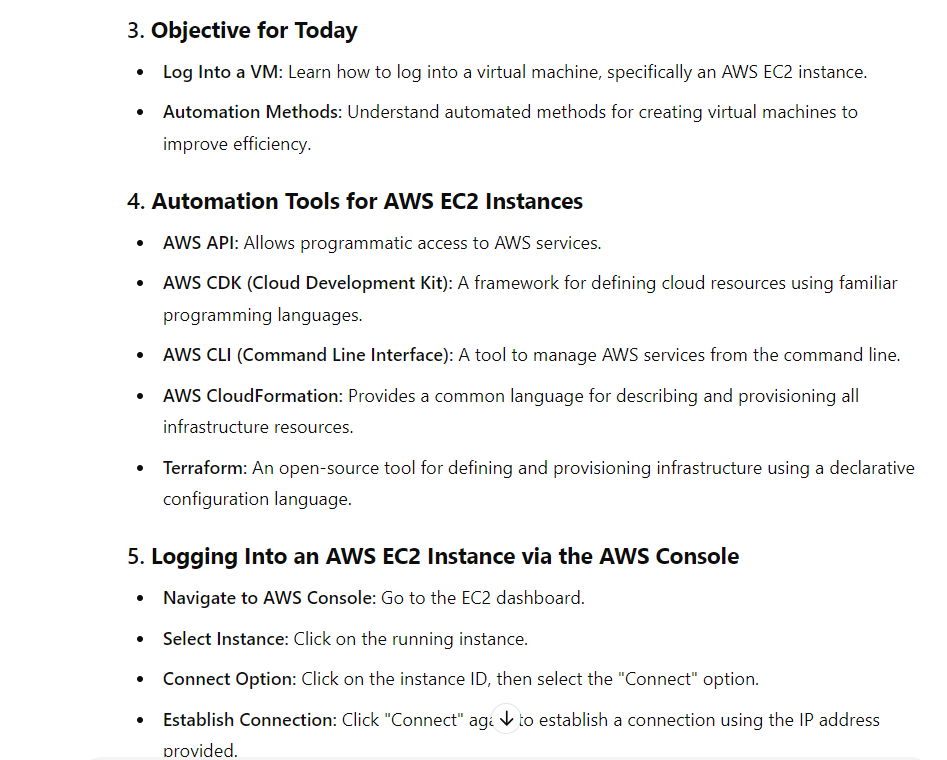


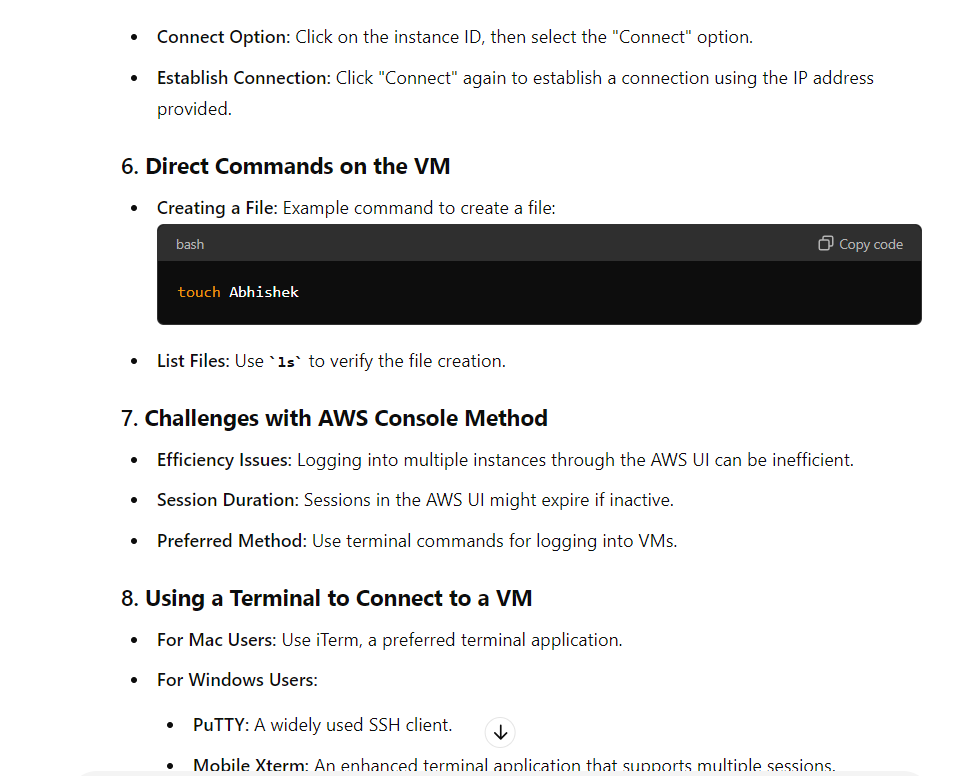


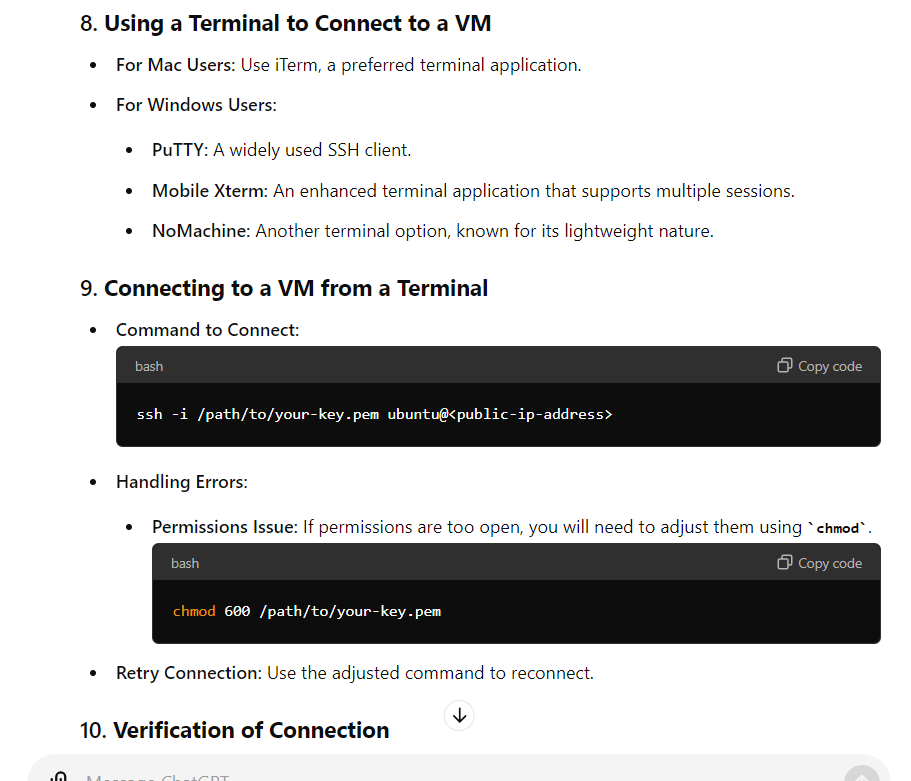


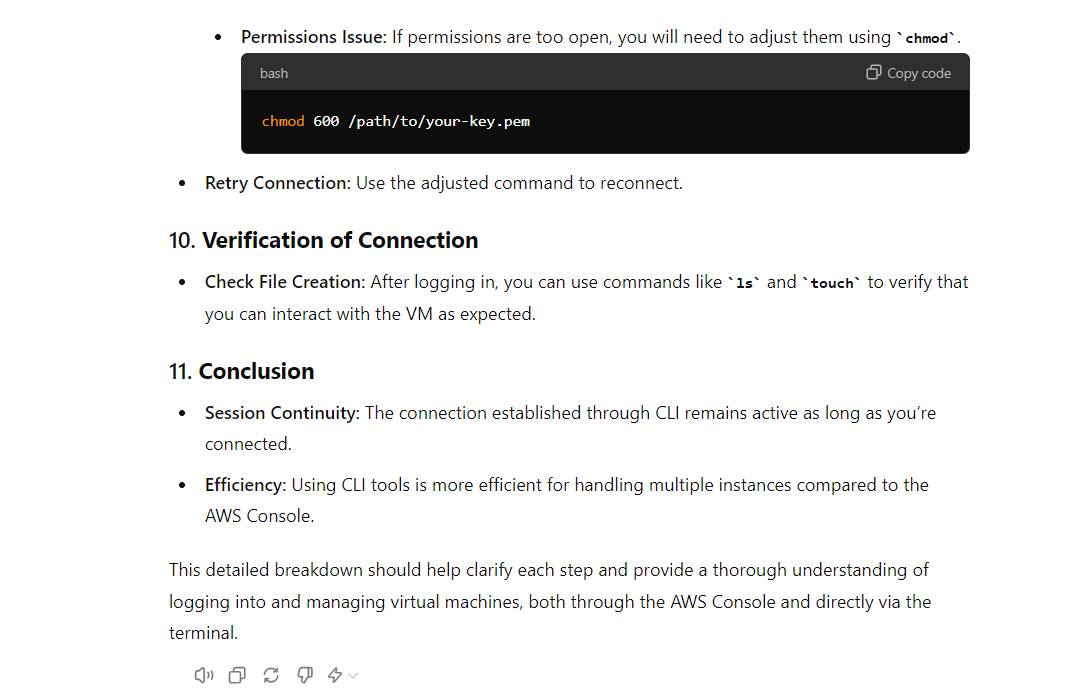


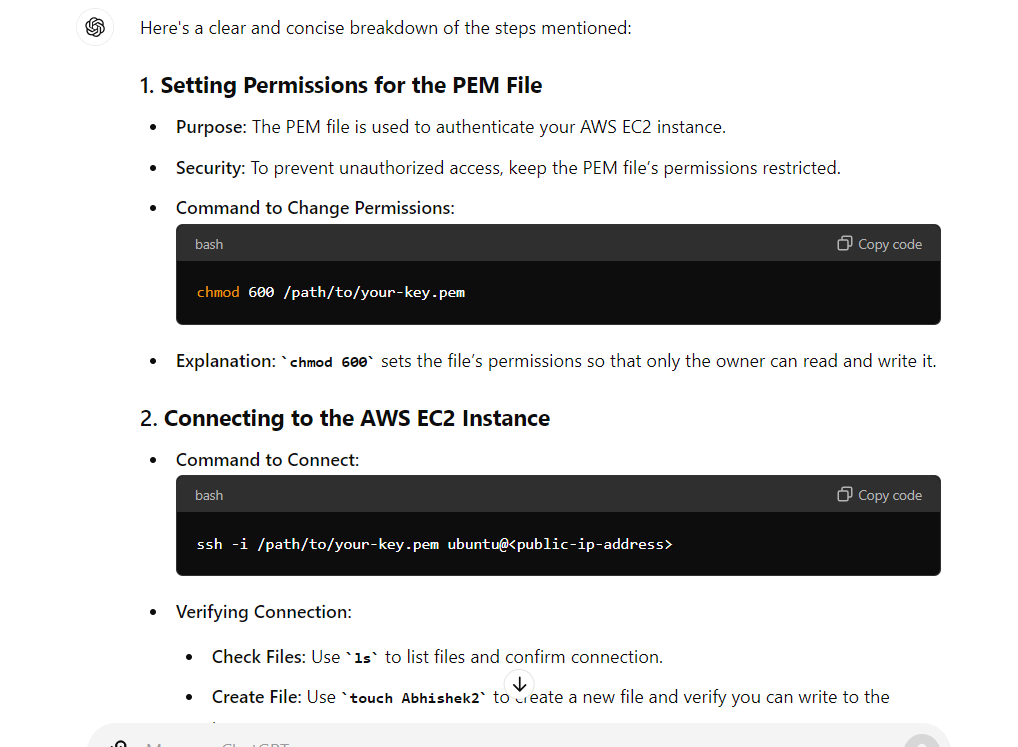
More in detail:

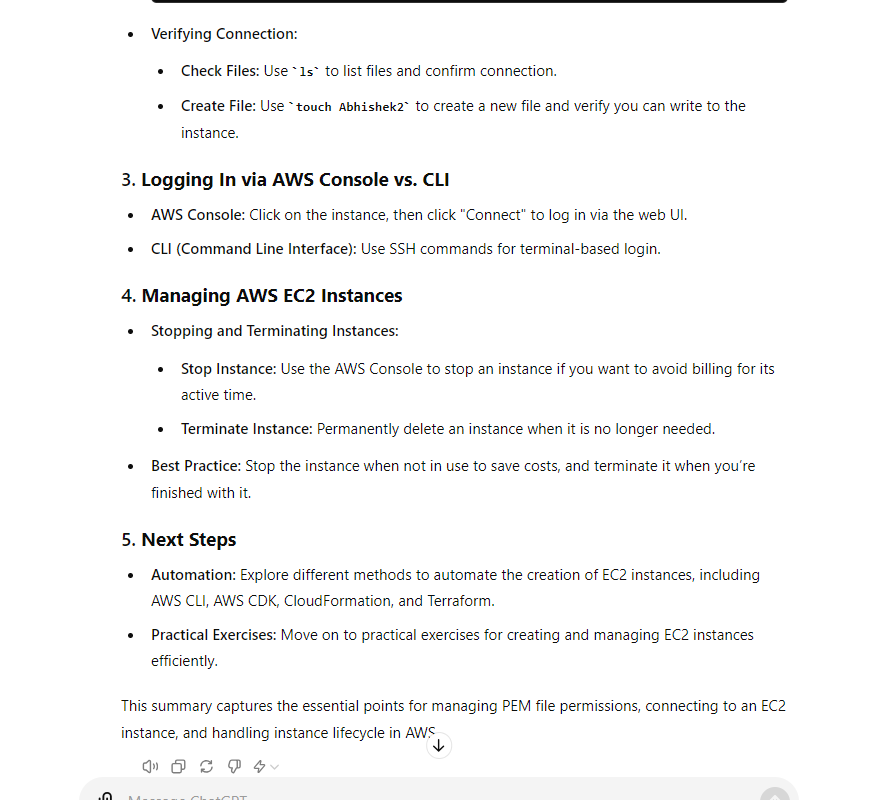




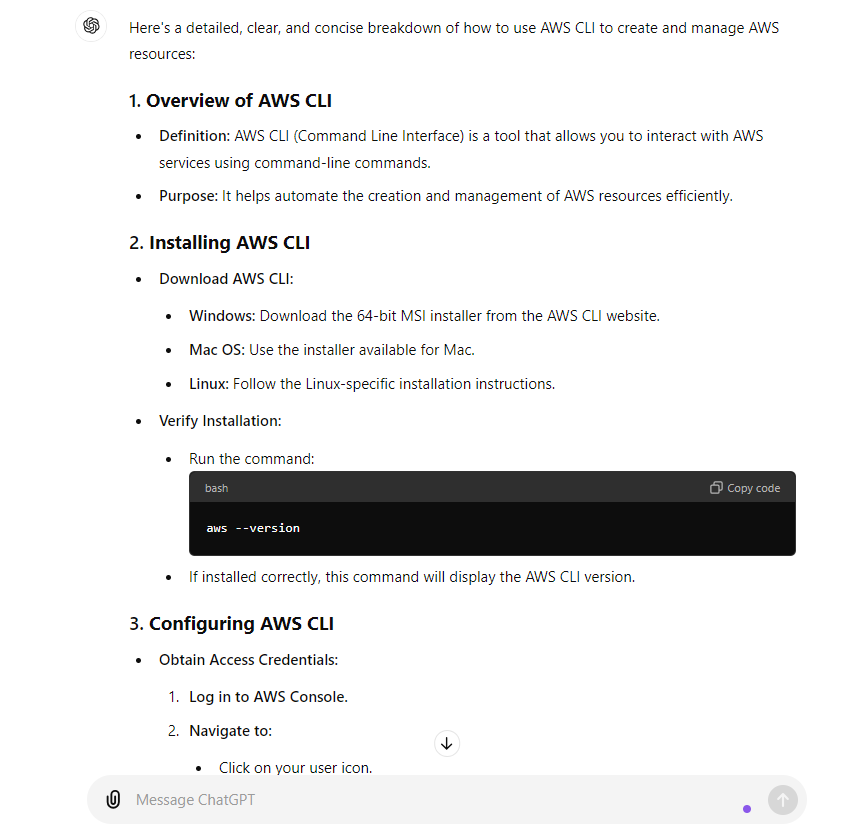


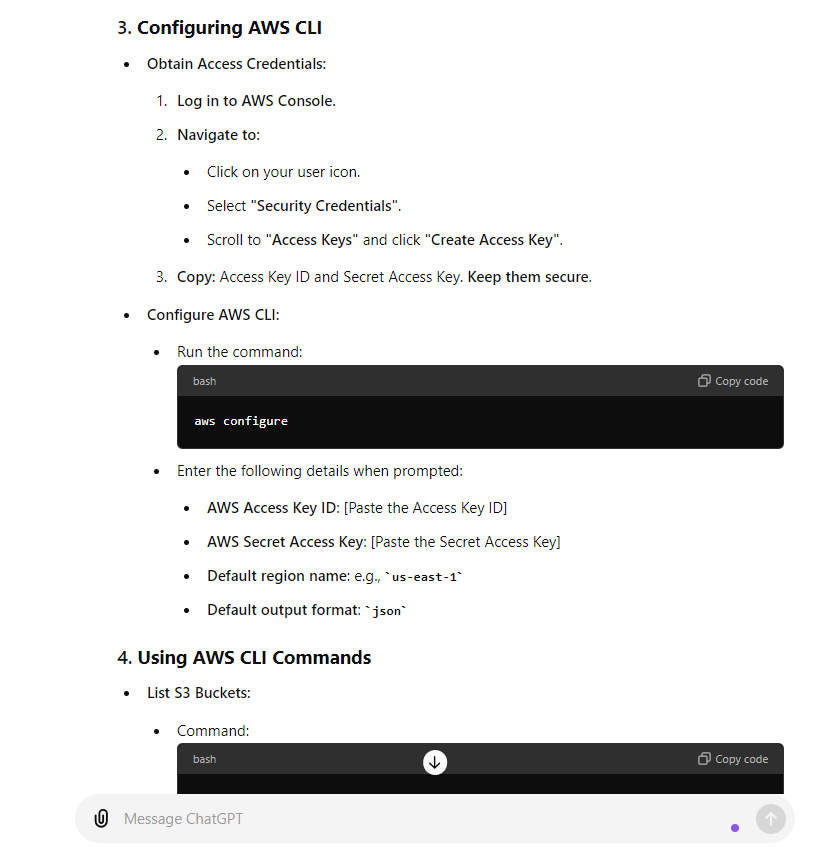


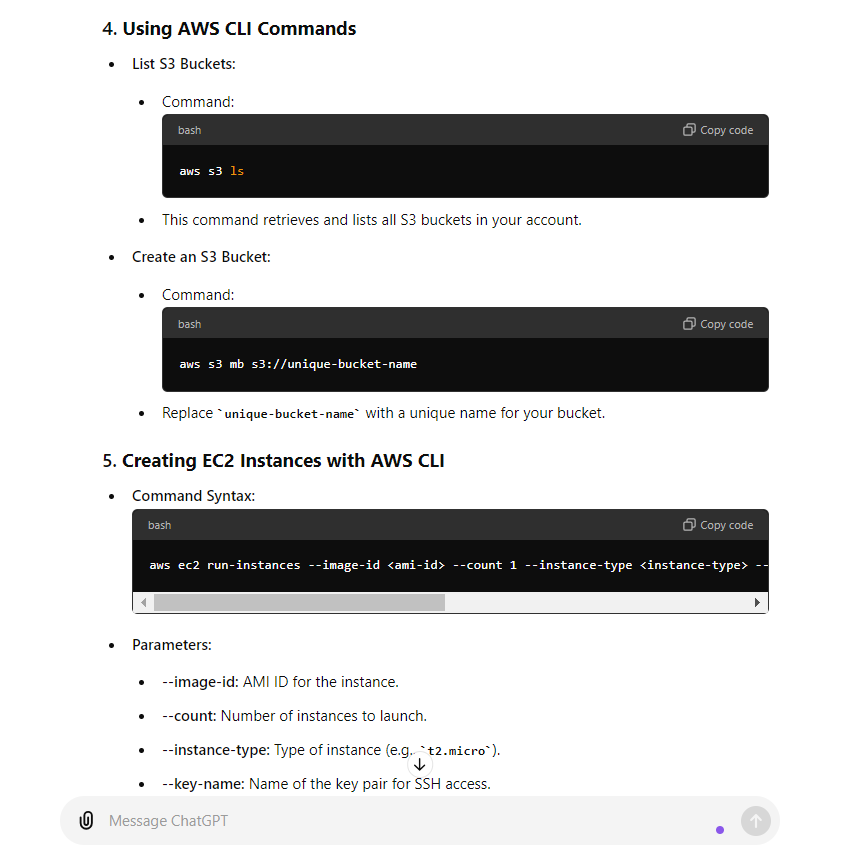


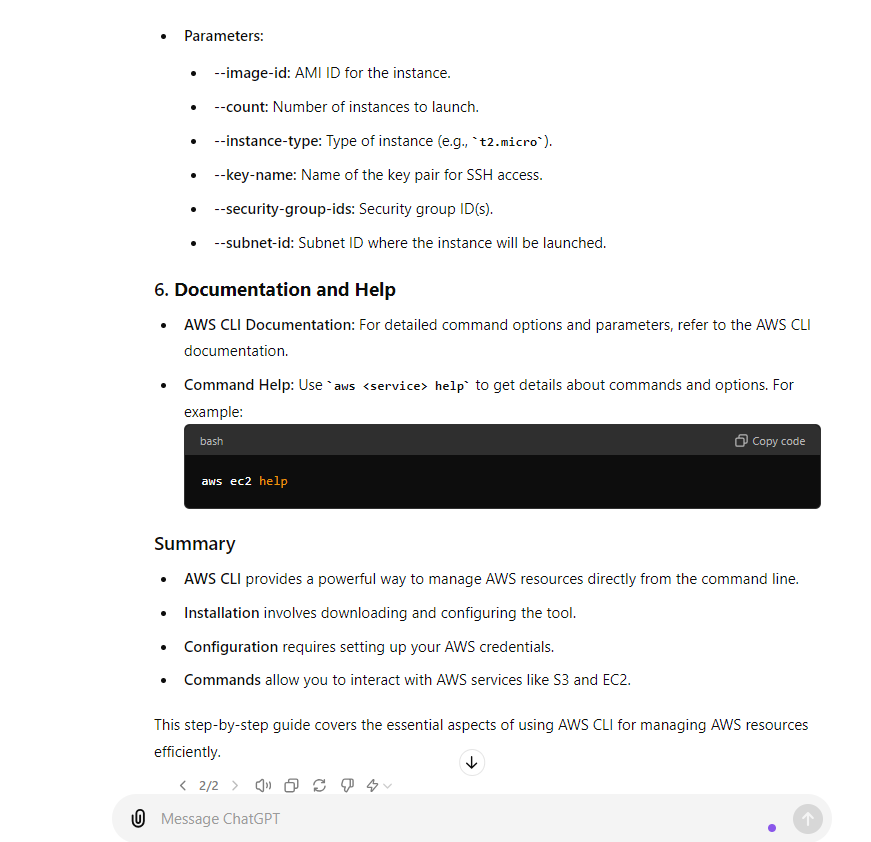


**13-20 mins video duration:**

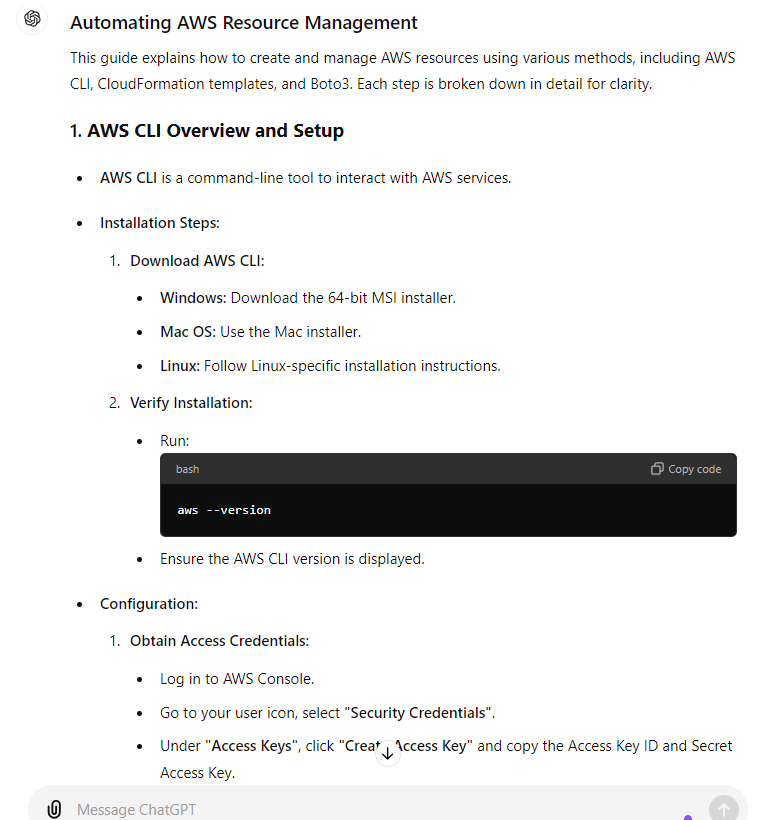
****

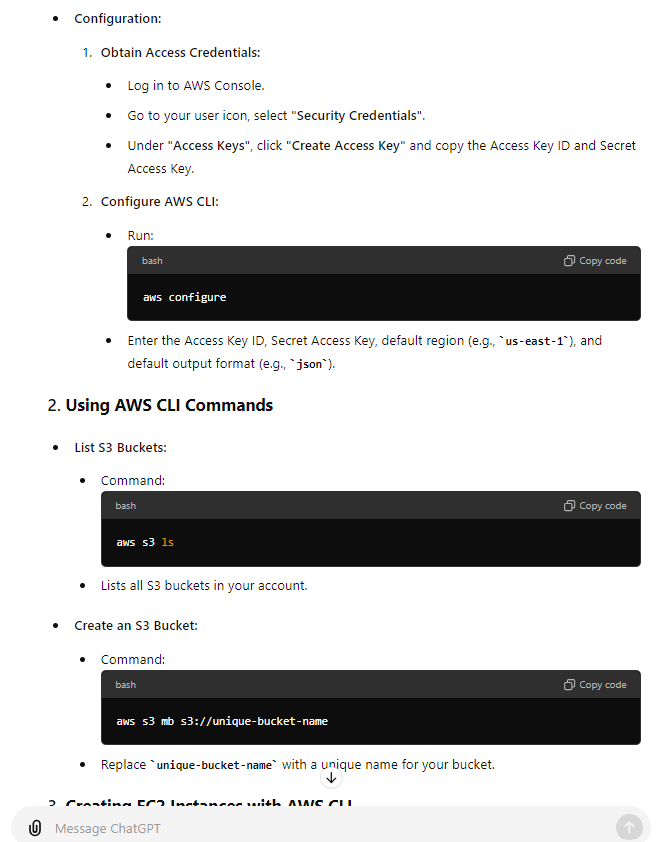
****

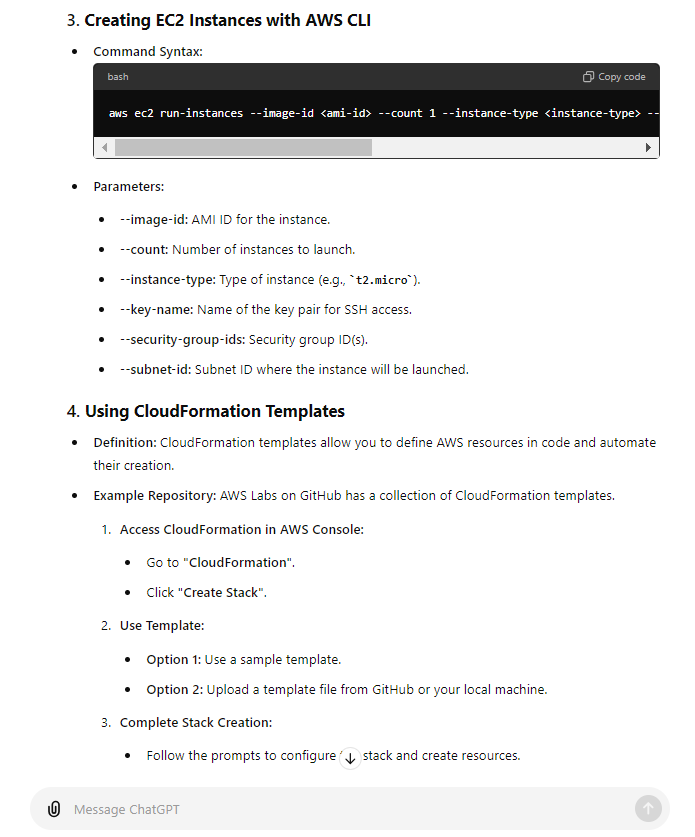
****

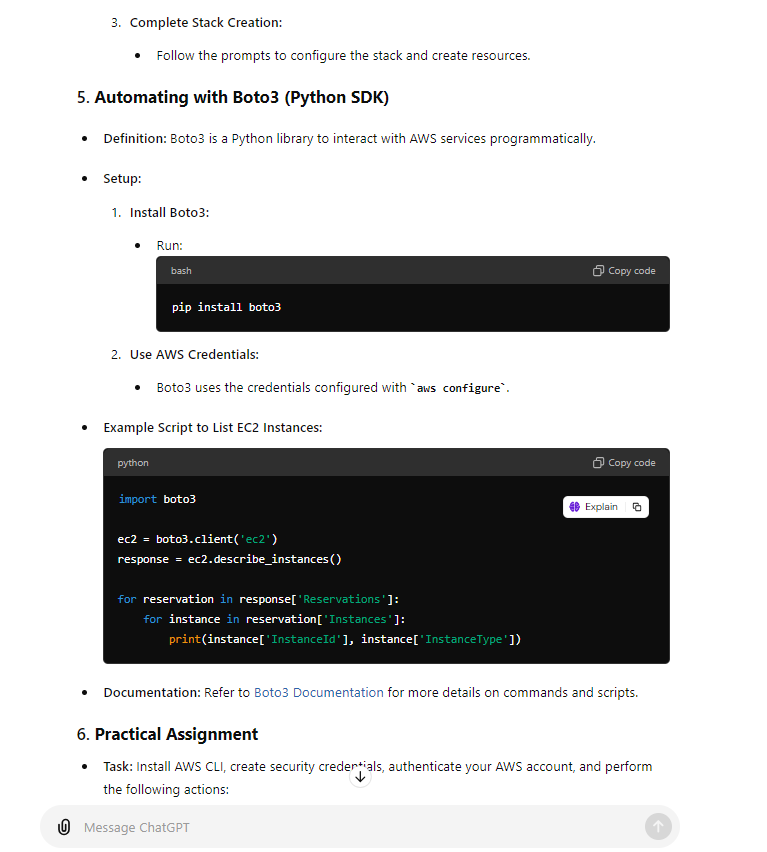
****

**20-30 mints video:**

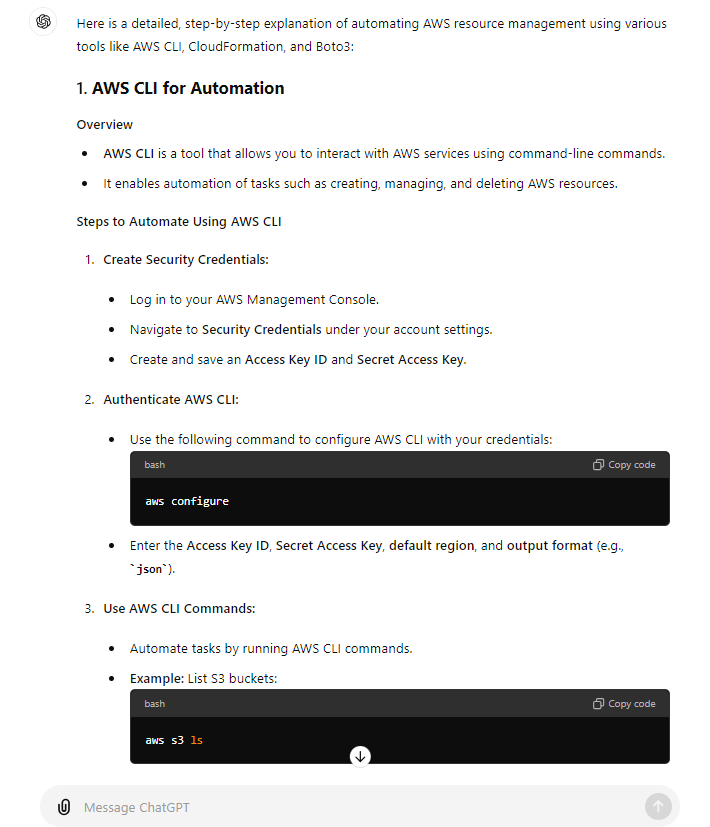
****

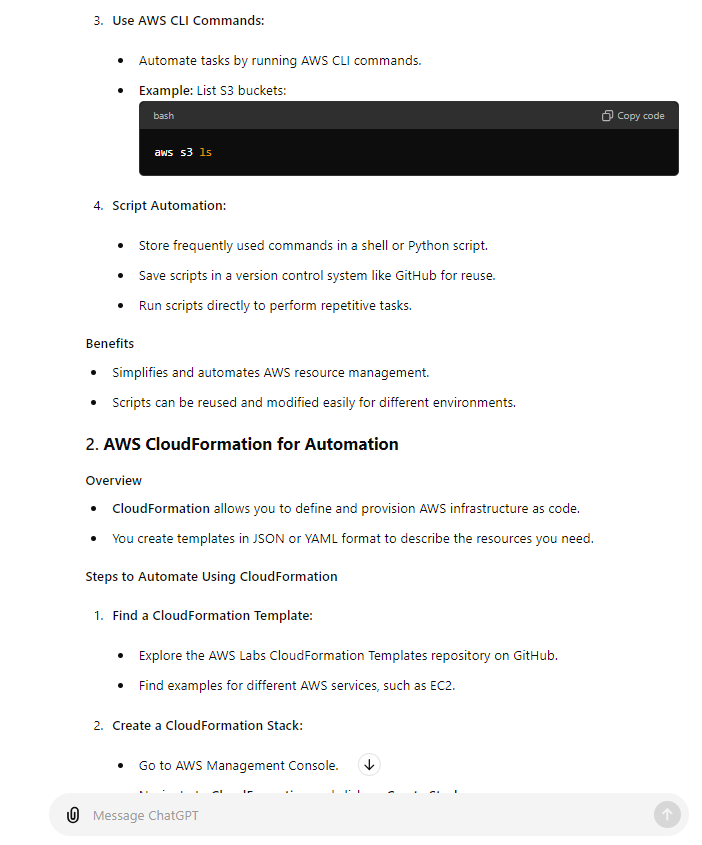
****

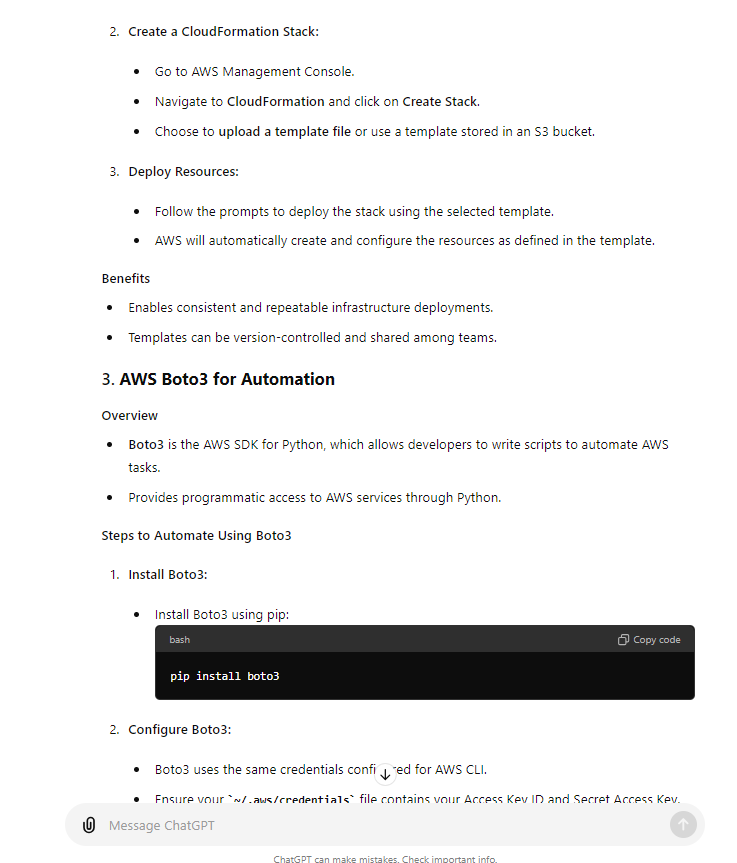
****

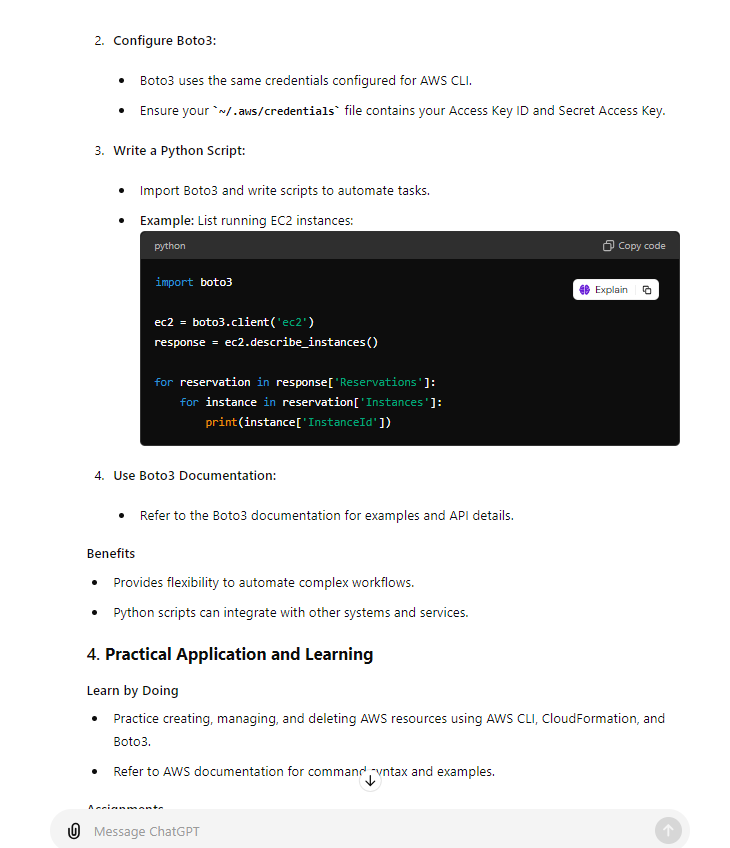
****

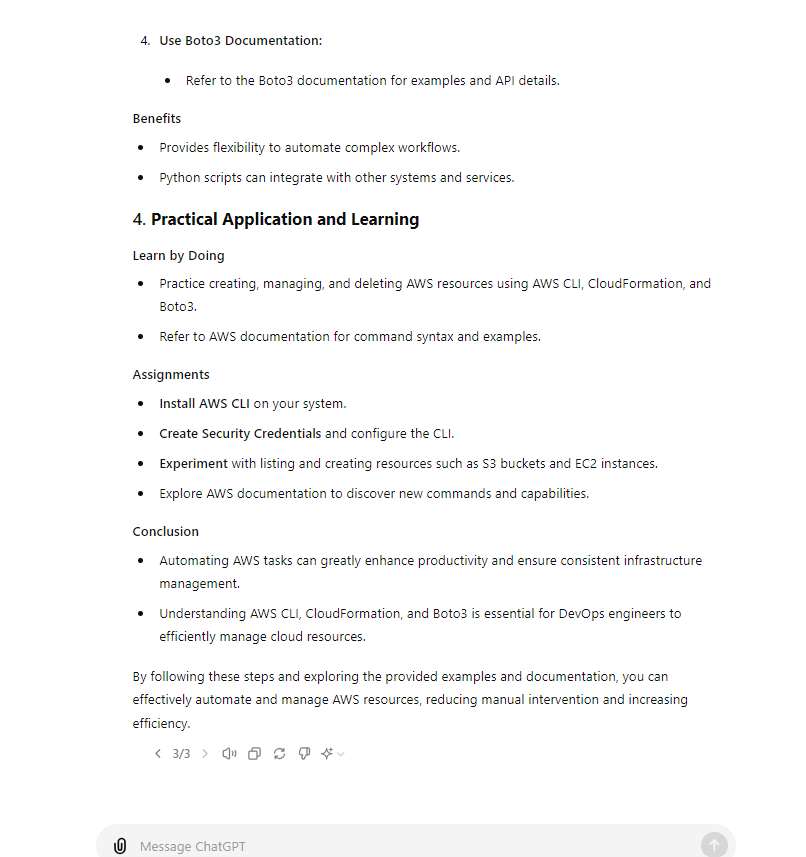
**Another Response for better understanding:**

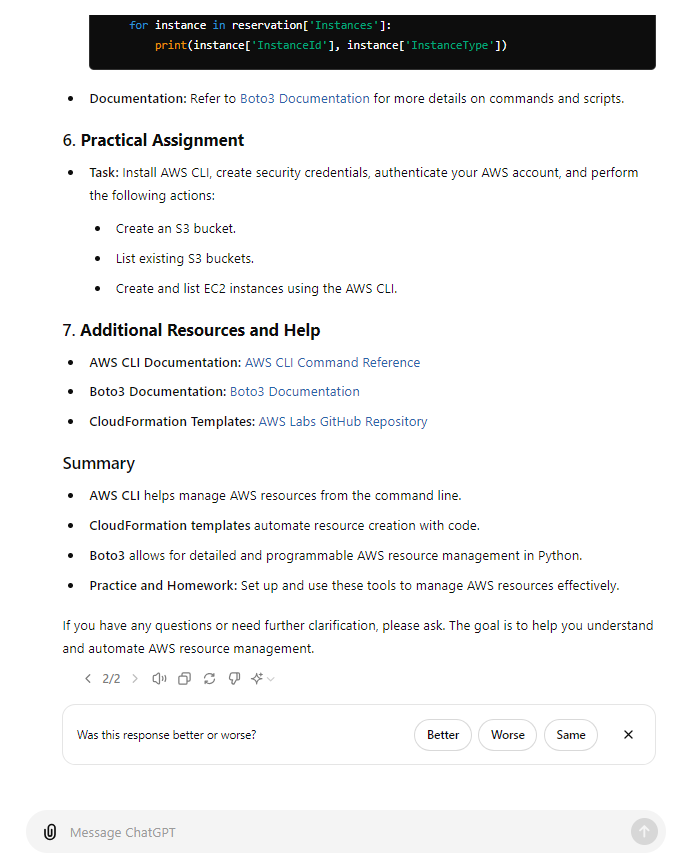
****

****

****

****

****

****