How to automate configuration in Linux using Ansible

In the world of systems administration, automating configuration tasks is essential for managing large-scale environments efficiently. Automation helps reduce human error, saves time, and ensures consistency across systems. In Linux, one popular tool for automating configuration is Ansible.

Ansible is an open-source automation engine that simplifies the process of automating tasks such as configuration management, application deployment, and orchestration. It uses a simple and human-readable language called YAML (Yet Another Markup Language) to define tasks and playbooks.

To align the topic of automation with the Linux environment, we will focus on using Ansible for configuration management in Linux systems. This will enable system administrators to automate repetitive tasks, enforce desired configurations, and ensure consistency across multiple Linux servers.

Examples:

1. Installing Ansible on Linux:

   - Open a terminal and run the following command to install Ansible:

     ```
     sudo apt-get install ansible
     ```

2. Creating a simple Ansible playbook:

   - Create a file named configure-webserver.yml and open it in a text editor.

   - Add the following content to the file:

     ```
     ---
     - name: Configure web server
     hosts: webserver
     become: true
     tasks:
      - name: Install Apache web server
        apt:
          name: apache2
          state: present
        notify:
          - Start Apache

      - name: Enable Apache service
        service:
     ```

```
     name: apache2
     enabled: true
     state: started

handlers:
 - name: Start Apache
   service:
     name: apache2
     state: restarted
```

- Save the file.

3. Running the Ansible playbook:

   - Open a terminal and navigate to the directory where the playbook is saved.
   - Run the following command to execute the playbook:

     ```
     ansible-playbook configure-webserver.yml
     ```

Note: Ansible is not exclusive to Linux and can also be used with other operating systems such as Windows. However, the examples provided here are specific to the Linux environment. For Windows systems, alternatives like PowerShell Desired State Configuration (DSC) or Chocolatey can be used for configuration automation.