1. **Ad Hoc Commands** (Các lệnh Ad Hoc):
   * Các lệnh Ad hoc là những thao tác đơn giản, thực hiện trong một dòng lệnh mà không cần viết playbook.
   * Chúng hữu ích cho việc kiểm tra nhanh và thay đổi nhỏ.
   * Ví dụ: để khởi động một dịch vụ hoặc đảm bảo một dòng lệnh tồn tại trong file.
   * Tuy nhiên, các lệnh Ad hoc có giới hạn của nó.
2. **Ansible Modules** (Các mô-đun Ansible):
   * Ansible cung cấp các mô-đun, là mã lệnh được sử dụng để tự động hóa các tác vụ cụ thể.
   * Một số ứng dụng của mô-đun:
     + Đảm bảo người dùng tồn tại với các cài đặt nhất định.
     + Đảm bảo phiên bản mới nhất của một gói phần mềm được cài đặt.
     + Triển khai file cấu hình lên một máy chủ.
     + Kích hoạt một dịch vụ mạng và đảm bảo nó đang chạy.
   * Hầu hết các mô-đun đều là **idempotent**, nghĩa là chúng chỉ thực hiện thay đổi nếu cần thiết. Các mô-đun idempotent có thể chạy an toàn nhiều lần.
   * Một lệnh Ad hoc sẽ chạy một mô-đun trên các máy chủ được quản lý cụ thể.
3. **Running Ad Hoc Commands** (Chạy các lệnh Ad Hoc):
   * Lệnh ansible sẽ chạy một lệnh Ad hoc.
   * Tham số host-pattern xác định các máy chủ được quản lý để chạy lệnh.
   * Tùy chọn -m chỉ định mô-đun mà Ansible sẽ chạy.
   * Tùy chọn -a nhận danh sách các đối số cần thiết cho mô-đun.

Cú pháp:

ansible host-pattern -m module [-a 'module arguments'] [-i inventory]

**Running Ad Hoc Commands** (Chạy các lệnh Ad Hoc):

* Một trong những lệnh Ad hoc đơn giản nhất sử dụng mô-đun ping.
* Nó không gửi một ICMP ping đến máy chủ được quản lý.
* Nó kiểm tra xem các mô-đun Ansible được viết bằng Python có thể chạy trên máy chủ được quản lý hay không.

Ví dụ lệnh:  
  
ansible all -m ping

**Overriding Default Configuration Settings** (Ghi đè cài đặt cấu hình mặc định):

* Để ghi đè cài đặt cấu hình mặc định, có nhiều tùy chọn khác nhau.
* Các tùy chọn này ghi đè cấu hình trong file ansible.cfg.

Các tùy chọn phổ biến:

* -k hoặc --ask-pass: yêu cầu nhập mật khẩu kết nối.
* -u REMOTE\_USER: ghi đè cài đặt remote\_user trong ansible.cfg.
* -b: kích hoạt nâng quyền, chạy các thao tác với become: yes.
* -K hoặc --ask-become-pass: yêu cầu nhập mật khẩu nâng quyền.
* --become-method: ghi đè phương pháp nâng quyền mặc định, mặc định là sudo.

**Ansible Modules** (Các mô-đun Ansible):

* Hầu hết các mô-đun đều nhận các đối số để kiểm soát chúng.
* Sử dụng tùy chọn -a để truyền các đối số.

Ví dụ:

* Sử dụng mô-đun user để đảm bảo người dùng newbie tồn tại và có UID là 4000.

Ví dụ lệnh:  
  
ansible -m user -a 'name=newbie uid=4000 state=present' servera.lab.example.com

**Selected Ansible Command-line Options** (Các tùy chọn dòng lệnh Ansible được chọn):

* Một số chỉ thị dòng lệnh có thể được sử dụng để ghi đè các tùy chọn từ file cấu hình Ansible.

Bảng các tùy chọn:

* inventory: --inventory, --inventory-file, -i
* remote\_user: --user, -u
* become: --become, -b
* become-method: --become-method
* become\_user: --become-user
* become\_ask\_pass: --ask-become-pass, -K

**Finding Information about Ansible Modules** (Tìm thông tin về các mô-đun Ansible):

* Lệnh ansible-doc -l sẽ liệt kê tất cả các mô-đun được cài đặt trên hệ thống.
* Tên và mô tả của mô-đun sẽ được hiển thị.
* Có hàng ngàn mô-đun có sẵn: bạn có thể dùng lệnh grep để lọc kết quả.
* Thông tin tương tự có thể tìm thấy trên trang web Ansible: Link tới tài liệu mô-đun Ansible.

**Selected Ansible Modules** (Các mô-đun Ansible đã chọn):

* Có hàng ngàn mô-đun để thực hiện các tác vụ. Một số mô-đun được chọn bao gồm:

**File Modules** (Mô-đun file):

* copy: Sao chép một file cục bộ lên máy chủ được quản lý.
* file: Đặt quyền và các thuộc tính khác cho file.
* lineinfile: Đảm bảo một dòng cụ thể có hoặc không có trong file.
* synchronize: Đồng bộ hóa nội dung bằng rsync.

**Software package modules** (Mô-đun quản lý gói phần mềm):

* yum: Quản lý gói phần mềm sử dụng YUM.
* dnf: Quản lý gói phần mềm sử dụng DNF.
* gem: Quản lý các gem Ruby.

**System Modules** (Mô-đun hệ thống):

* firewalld: Quản lý các cổng và dịch vụ tùy ý bằng firewalld.
* reboot: Khởi động lại máy.
* service: Quản lý các dịch vụ.
* user: Thêm, xóa, và quản lý tài khoản người dùng.

**Net Tools modules** (Mô-đun công cụ mạng):

* get\_url: Tải file qua HTTP, HTTPS, hoặc FTP.
* nmcli: Quản lý mạng.
* uri: Tương tác với các dịch vụ web và giao tiếp với các API.

**Getting Information about an Ansible Module** (Lấy thông tin về một mô-đun Ansible):

* ansible-doc cũng có thể cung cấp thông tin về cách sử dụng một mô-đun.

Ví dụ lệnh:  
  
ansible-doc ping

**Ad Hoc Command Example: User Creation** (Ví dụ về lệnh Ad Hoc: Tạo người dùng):

* Bạn có thể sử dụng lệnh ansible-doc -l để khám phá mô-đun user.
* Sau đó, chạy lệnh ansible-doc user để tìm hiểu cách mô-đun hoạt động.

Ví dụ lệnh:  
  
ansible all -m user -a 'name=newbie uid=4000 state=present'

**Ad Hoc Command Example: Group Creation** (Ví dụ về lệnh Ad Hoc: Tạo nhóm):

* Bạn có thể sử dụng lệnh ansible-doc -l để khám phá mô-đun group.
* Sau đó, chạy lệnh ansible-doc group để tìm hiểu cách mô-đun hoạt động.

Ví dụ lệnh:  
  
ansible all -m group -a 'name=developers gid=2000 state=present'

**Ad Hoc Command Example: Group Management** (Ví dụ về lệnh Ad Hoc: Quản lý nhóm):

* Mô-đun user có thể được sử dụng để điều chỉnh thành viên của nhóm.

Ví dụ lệnh:  
  
ansible all -m user -a 'name=newbie groups=developers,wheel append=yes state=present'

**Ad Hoc Command Example: Software Package Installation** (Ví dụ về lệnh Ad Hoc: Cài đặt gói phần mềm):

* Bạn có thể sử dụng lệnh ansible-doc -l để khám phá mô-đun package.
* Sau đó, chạy lệnh ansible-doc package để tìm hiểu cách mô-đun hoạt động.

Ví dụ lệnh:  
  
ansible all -m package -a 'name=httpd state=present'

**Command Modules** (Mô-đun lệnh):

* Có một số mô-đun chạy lệnh trực tiếp trên máy chủ được quản lý.
* Bạn có thể sử dụng những mô-đun này nếu không có mô-đun nào khác làm được điều bạn cần.
* Chúng không phải là **idempotent**: bạn phải đảm bảo rằng chúng an toàn để chạy nhiều lần khi sử dụng chúng.

Các mô-đun bao gồm:

* command: Chạy một lệnh đơn trên hệ thống từ xa.
* shell: Chạy một lệnh trên shell của hệ thống từ xa.
* raw: Chạy một lệnh mà không qua xử lý (có thể nguy hiểm).

**Running Arbitrary Commands on Managed Hosts** (Chạy các lệnh tùy ý trên các máy chủ được quản lý):

* Mô-đun command cho phép quản trị viên chạy các lệnh tùy ý.
* Nó không thể truy cập các biến môi trường shell hoặc thực hiện các thao tác shell như chuyển hướng và piping.

Ví dụ lệnh:  
  
ansible mymanagedhosts -m command -a /usr/bin/hostname

**Running Arbitrary Commands on Managed Hosts** (Chạy các lệnh tùy ý trên các máy chủ được quản lý):

* Mô-đun shell được sử dụng khi các lệnh cần xử lý bởi shell.

Ví dụ:  
  
ansible localhost -m command -a set

* + Kết quả: FAILED | rc=2 >> [Errno 2] No such file or directory

ansible localhost -m shell -a set

* + Kết quả: CHANGED | rc=0 >> BASH=/bin/sh
* Cả mô-đun command và shell đều yêu cầu cài đặt Python hoạt động trên máy chủ được quản lý.
* Mô-đun raw có thể chạy lệnh trực tiếp sử dụng shell từ xa, bỏ qua hệ thống mô-đun.
* Điều này hữu ích khi quản lý các hệ thống không thể cài đặt Python (ví dụ như router mạng).

**When to Use Ad Hoc Commands** (Khi nào sử dụng các lệnh Ad Hoc):

* Các lệnh Ad hoc hữu ích khi bạn cần thực hiện một thay đổi nhanh chóng trên nhiều hệ thống.
* Điều này rất mạnh mẽ khi bạn cần thực hiện một thay đổi đơn giản nhanh chóng.

Tuy nhiên, chúng có một số nhược điểm:

* Bạn chỉ có thể gọi một mô-đun, giới hạn chúng trong các thay đổi đơn giản.
* Bạn phải nhập lại cùng một lệnh nếu muốn chạy lại, các tùy chọn có thể trở nên phức tạp.
* Chúng vẫn mang tính chất bán thủ công.

Trong nhiều trường hợp, một cách tiếp cận tốt hơn là sử dụng **Ansible Playbooks**:

* Playbooks có thể chạy nhiều mô-đun với các điều kiện và xử lý khác.
* Playbooks là các file văn bản có thể được theo dõi trong các hệ thống kiểm soát phiên bản.
* Playbooks có thể dễ dàng chạy lại với một lệnh đơn giản.