**Ping Atmak için:**Playbook1

- name: Test Connectivity

  hosts: all

  tasks:

   - name: Ping test

     ansible.builtin.ping:

ansible-playbook playbook1.yml **#Terminalde Kodu çalıştırmak için**

Control Node olan bir dosyayı webservers hostlarına göndermek:Playbook2

- name: Copy for linux

  hosts: webservers

  tasks:

   - name: Copy your file to the webservers

     ansible.builtin.copy:

       src: /home/ec2-user/testfile1

       dest: /home/ec2-user/testfile1

- name: Copy for ubuntu

  hosts: ubuntuservers

  tasks:

   - name: Copy your file to the ubuntuservers

     ansible.builtin.copy:

       src: /home/ec2-user/testfile1

       dest: /home/ubuntu/testfile1

ansible-playbook playbook.yml **#Terminalde Kodu çalıştırmak için**

ansible all -m shell -a "ls" (bütün nodelarda ls komutu ile varolan dosyalar gözükebilir)

ansible node1 –m commant –a “cat text\*” yada (node1 adı verilen node text ön adlı bütün dosyaları getirir)

ansible node1 –m commant –a “cat testfile1”

testfile1 isimli dosyanın kullanıcı izinleri root okuma yazma,gruba sadece okuma yetkisi verilmesi ,diğerlerine hiçbir yetki verilmez:Playbook3

- name: Copy for linux

  hosts: webservers

  tasks:

   - name: Copy your file to the webservers

     ansible.builtin.copy:

       src: /home/ec2-user/testfile1

       dest: /home/ec2-user/testfile1

       mode: u+rw,g-wx,o-rwx

- name: Copy for ubuntu

  hosts: ubuntuservers

  tasks:

   - name: Copy your file to the ubuntuservers

     ansible.builtin.copy:

       src: /home/ec2-user/testfile1

       dest: /home/ubuntu/testfile1

       mode: u+rw,g-wx,o-rwx

- name: Copy for node1

  hosts: node1

  tasks:

   - name: Copy using inline content

     ansible.builtin.copy:

       content: 'This is content of file2'

       dest: /home/ec2-user/testfile2

   - name: Create a new text file

     ansible.builtin.shell: "echo Hello World > /home/ec2-user/testfile3"

#Yukarıda ikinci kodda node1 adlı noda test2 adında klasör oluşturup içeriğine 'This is content of file2' yazılması ve echo komutunun kullanılması(Bu yorumu kod yazarken silin)

Amazon Linux makinalara yum paketi ile httpd adıyla apache kurulması.Ubuntu makinaya ise apache2 adıyla ve apt komutuyla apache kurulması:Playbook4

- name: Apache installation for webservers

  hosts: webservers

  tasks:

   - name: install the latest version of Apache

     ansible.builtin.yum:

       name: httpd

       state: latest

   - name: start Apache

     ansible.builtin.shell: "service httpd start"

- name: Apache installation for ubuntuservers

  hosts: ubuntuservers

  tasks:

   - name: update

     ansible.builtin.shell: "apt update -y"

   - name: install the latest version of Apache

     ansible.builtin.apt:

       name: apache2

       state: latest

ansible-playbook -b playbook4.yml (Terminalde sudo yetkisi ile bu komutların çaılştırılması gerekiyor)

Apache’nin kaldırılması state:absent kaldırır.autoremove:yes uzantılarını kaldırır.Purge:yes köken kaldırır:Playbook5

- name: Remove Apache from webservers

  hosts: webservers

  tasks:

   - name: Remove Apache

     ansible.builtin.yum:

       name: httpd

       state: absent

       autoremove: yes

- name: Remove Apache from ubuntuservers

  hosts: ubuntuservers

  tasks:

   - name: Remove Apache

     ansible.builtin.apt:

       name: apache2

       state: absent

       autoremove: yes

       purge: yes

ansible-playbook -b playbook5.yml(Terminalde kodun çalıştırılması için)

Ubuntu makinada apachenin yüklenmesi html kodlarının olduğu yere gidip(/var/www/html/index.html) oranın içeriğini content: "<h1>Hello Techproed</h1>" bu komutla değiştirip ana sayfıyı Hello Techproed yazması ve apache nin çalışması içinde başlatıp enable yapması:

Linux makinalarda ise httpd yüklemesi ve wget indirme programının yüklenmesi:Playbook6

- name: Apache installation and configuration for ubuntuservers

  hosts: ubuntuservers

  tasks:

   - name: installing apache

     ansible.builtin.apt:

       name: apache2

       state: latest

   - name: index.html

     ansible.builtin.copy:

       content: "<h1>Hello Techproed</h1>"

       dest: /var/www/html/index.html

   - name: restart apache2

     ansible.builtin.service:

       name: apache2

       state: restarted

       enabled: yes

- name: Apache and wget installation for webservers

  hosts: webservers

  tasks:

    - name: installing httpd and wget

      ansible.builtin.yum:

        pkg: "{{ item }}" #kodun açıklamsı aşağıda

        state: present

      loop:

        - httpd

        - wget

ansible-playbook -b playbook6.yml

- name: Apache and wget installation for webservers

  hosts: webservers

  tasks:

    - name: installing httpd and wget

      ansible.builtin.yum:

        pkg: "{{ item }}"

        state: present

      loop:

        - httpd

        - wget

Yukarıdaki kodun açıklaması:

1. **name** (ad): Bu, bu görevin açıklamasını belirtir. Görevin adı, betik çalıştırıldığında görüntülenir ve betiği daha okunabilir hale getirmek için kullanılır.
2. **ansible.builtin.yum**: Bu, Ansible'ın yerel modülü olan **yum** modülünü kullanarak işlemler yapacağınızı belirtir. **ansible.builtin** öneki, Ansible'ın yerleşik modüllerini işaret eder.
3. **pkg** (paket): Bu, kurulacak veya kaldırılacak paketin adını belirtir. İşte bu alanda **{{ item }}** kullanılarak bir döngü oluşturulmuştur. **item**, her dönüşte bir listedeki öğeyi temsil eder. Bu betikte **item**, **loop** ile belirtilen liste içindeki her bir paketi temsil eder.
4. **state** (durum): Bu, paketin durumunu belirtir. Bu durum, **present** olarak ayarlanmış, yani paketin yüklenmesini istediğinizi gösterir. Ayrıca **absent** olarak ayarlanabilirse, paketin kaldırılmasını sağlar.
5. **loop**, belirtilen paket listesini dolaşarak her bir paketi kurmayı sağlar. Bu sayede aynı işlemi tekrar tekrar yazmak yerine, liste içindeki her bir öğeyi işlemek için bir döngü kullanabilirsiniz. Bu, birçok benzer görevi kısaltmak ve daha temiz ve okunabilir Ansible betikleri oluşturmanıza olanak tanır.

Apache’nin kadırılması:Playbook7

- name: Remove Apache from ubuntuservers

  hosts: ubuntuservers

  tasks:

   - name: Uninstalling Apache

     ansible.builtin.apt:

       name: apache2

       state: absent

       update\_cache: yes         # update et sonra kaldır.Açıklaması aşağıda

       autoremove: yes

       purge: yes

- name: Remove Apache and wget from webservers

  hosts: webservers

  tasks:

   - name: removing apache and wget

     ansible.builtin.yum:

       pkg: "{{ item }}"

       state: absent

     loop:

       - httpd

       - wget

Bu komutta niye güncelleyip kaldırıyor:

- name: Uninstalling Apache

     ansible.builtin.apt:

       name: apache2

       state: absent

       update\_cache: yes         # update et sonra kaldır

       autoremove: yes

       purge: yes

Paket yöneticileri, paketlerin kurulu olduğu sistemdeki bağımlılıkları ve paket sürümlerini yönetirler. Bir paketi kaldırırken veya değiştirirken, bu paketin bağımlı olduğu diğer paketlerle ilişkileri ve uyumluluğu göz önünde bulundurmak önemlidir.

İşte neden paket kaldırma işlemi sırasında güncelleme gerekebilir:

1. **Bağımlılıklar**: Bir paket, diğer paketlere veya kütüphanelere bağımlı olabilir. Bu bağımlılıkların güncel ve uyumlu olması gerekebilir. Eğer bağımlı paketler güncellenmişse veya kaldırılmışsa, bu durumda paket yöneticisi bu değişiklikleri yönetmelidir. Bu nedenle güncel paket listeleriyle çalışmak önemlidir.
2. **Güncelleme Politikaları**: Paket yöneticisi, güncelleme politikalarını takip eder. Bazı paket yöneticileri, bir paketi kaldırmadan önce önceden güncellenmiş olmasını veya güncelleme politikalarına uygun bir şekilde işlem yapılmasını gerektirebilir.
3. **Uyum Sorunları Önleme**: Paket yöneticileri, uyum sorunlarını önlemek için işlem sırasında güncel bilgilere ihtiyaç duyar. Eğer paket yöneticisi güncel değilse, uyumsuzluklar ve hatalar ortaya çıkabilir.

Sonuç olarak, bir paketi kaldırırken veya değiştirirken, paket yöneticisinin güncel paket listelerine erişmesi ve bu listeleri güncellemesi önemlidir. Bu nedenle, paket yöneticisi güncel değilse veya güncel değilse, işlem sırasında **update\_cache** veya benzeri bir komut kullanılması gerekebilir. Bu, işlemin düzgün ve güncel bilgilere dayalı bir şekilde gerçekleşmesini sağlar.

Nodeların İşletim sistemine göre farklı kullanıcılar oluşturur:Playbook8

- name: Create users

  hosts: "\*"

  tasks:

    - name: Create user for REDHAT OS FAMILY

      ansible.builtin.user:

        name: "{{ item }}"

        state: present

      loop:

        - joe

        - matt

        - james

        - oliver

      when: ansible\_os\_family == "RedHat"     # if

    - name: Create user for SUSE OS FAMILY

      ansible.builtin.user:

        name: "{{ item }}"

        state: present

      loop:

        - david

        - tyler

      when: ansible\_os\_family == "SUSE"

    - name: Create user for DEBIAN OS FAMILY

      ansible.builtin.user:

        name: "{{ item }}"

        state: present

      loop:

        - john

        - aaron

      when: ansible\_os\_family == "Debian" or ansible\_distribution\_version == "20.04"

ansible-playbook -b playbook8.yml

ansible all -m shell -a "tail -4 /etc/passwd | cut -d: -f1" #kullanıcıları görmek için

ansible all -m shell -a "tail -4 /etc/passwd | cut -d: -f1" #kullanıcıları görmek için

Yukardaki kodun açıklaması

* **ansible all**: Bu komut, tüm hedef sunucuları hedef olarak seçer. **all** anahtar kelimesi, tüm sunucuları temsil eder.
* **-m shell**: Bu, Ansible'ın kullanılacak modülü belirtir. **shell** modülü, bir kabuk komutunu hedef sunucularda çalıştırmak için kullanılır.
* **-a "tail -4 /etc/passwd | cut -d: -f1"**: Bu, çalıştırılacak kabuk komutunu belirtir. İşte bu komutun parçaları:
  + **tail -4 /etc/passwd**: Bu komut, **/etc/passwd** dosyasının son dört satırını görüntüler. **/etc/passwd**, Linux ve Unix tabanlı işletim sistemlerinde kullanıcı hesap bilgilerini içeren bir dosyadır. Son dört satır genellikle sistem kullanıcılarını temsil eder.
  + **|** (boru işareti): Bu, **tail** komutundan gelen çıktının **cut** komutuna iletilmesini sağlar.
  + **cut -d: -f1**: Bu komut, **:** karakteriyle ayrılmış sütunları (kullanıcı bilgileri) alır ve ilk sütunu (kullanıcı adlarını) gösterir. **cut** komutu, belirli bir sütunu veya alanı ayıklamak için kullanılır.

Sonuç olarak, bu Ansible komutu tüm hedef sunucularda **/etc/passwd** dosyasının son dört satırını alır, bu satırlardaki kullanıcı adlarını ayırır ve bu kullanıcı adlarını görüntüler. Bu, hedef sunuculardaki kullanıcıları listelemek için kullanışlı bir komuttur.

cut -d: -f1" tam olarak ne yapıyor

ChatGPT

**cut -d: -f1** komutu, metin işleme işlemleri yapmak için kullanılan bir Unix/Linux komutudur. Bu komut, belirli bir ayırıcı karakterle ayrılmış bir metin dosyasından belirli sütunları (alanları) çıkarmak veya almak için kullanılır. İşte bu komutun ayrıntılı açıklaması:

* **cut**: **cut** komutu, metin işleme ve metin manipülasyonu işlemleri için kullanılır.
* **-d:**: Bu, ayırıcı karakteri belirtir. **-d** (delimiter) seçeneği ile birlikte kullanılır ve ayırıcı karakteri belirtir. Burada **:** karakteri, ayırıcı olarak kullanılır. Yani, **:** karakteri ile ayrılmış sütunları işleyeceğimizi belirtir. Örneğin, **/etc/passwd** dosyasındaki kullanıcı girişleri **:** karakteri ile ayrılmıştır.
* **-f1**: Bu, çıkarılacak sütunu belirtir. **-f** (field) seçeneği ile birlikte kullanılır ve hangi sütunu alacağınızı belirtir. **1**, ilk sütunu ifade eder. Yani, bu komut, belirtilen ayırıcı karakterle ayrılmış metin dosyasındaki her satırın ilk sütununu çıkarır ve görüntüler.

Örneğin, **/etc/passwd** dosyasındaki kullanıcı girişleri şu şekildedir:

root:x:0:0:root:/root:/bin/bash

user1:x:1001:1001:User 1:/home/user1:/bin/bash

user2:x:1002:1002:User 2:/home/user2:/bin/bash

Eğer **cut -d: -f1** komutunu bu dosyaya uygularsanız, aşağıdaki çıktıyı alırsınız:

root

user1

user2

Yani, bu komut, **:** karakteri ile ayrılmış olan bu satırlardaki ilk sütunu (kullanıcı adlarını) alır ve görüntüler. Bu, belirli bir metin dosyasındaki belirli bir sütunu almak için oldukça kullanışlı bir komuttur.

Formun Üstü