sudo yum update -y

sudo amazon-linux-extras install ansible2 –y

ansible ––version

cd /etc/ansible

ls

sudo vim hosts

[webservers]

node1 ansible\_host=<node2\_ip> ansible\_user=ec2-user

node2 ansible\_host=<node2\_ip> ansible\_user=ec2-user

[all:vars]

ansible\_ssh\_private\_key\_file=/home/ec2-user/<pem file>

scp -i <pem file> <pem file> ec2-user@<public DNS name of Control Node>:/home/ec2-user #key name kod ile taşımak

#scp -i /sizin/anahtar.pem /sizin/yerel/dosyanız ec2-user@<Control Node'un genel DNS adı>:/home/ec2-user

chmod 400 devops.pem

ansible all --list-hosts

ansible webservers --list-hosts

ansible all -m ping

ansible webservers -m ping

ansible node1 -m ping

ansible all -m ping -o

-o: Bu, komutun çıktısını düzgün bir şekilde formatlamak için kullanılır. Bu seçenek, çıktıyı "one line" (bir satırda) bir şekilde gösterir.

sudo vim /etc/ansible/ansible.cfg

[defaults](config dosyasının içindeki defaults altında olacak)

interpreter\_python=auto\_silent   #Çıkan uyarı mor yazıların çıkmamasını sağlar(Ptyhon ile ilgili)

interpreter\_python=auto\_silent" ifadesi, bir betik veya kod dosyasının çalıştırılmasında kullanılan Python yorumlayıcısının otomatik olarak belirlenmesini ve sessiz modda çalıştırılmasını ifade eder. Bu ifade, genellikle betik veya kod dosyasının başında bulunur ve yorumlayıcının otomatik olarak seçilmesine ve çıktının görüntülenmemesine neden olur.

host\_key\_checking = False

SSH, sunucu ile istemci (veya sunucu ile sunucu) arasında güvenli bir iletişim kurabilmek için anahtar tabanlı bir doğrulama sistemini kullanır. Bu doğrulama, istemcinin sunucunun kimliğini doğrulamasını sağlar. "host\_key\_checking" ayarı, bu anahtar doğrulama işlemini kontrol eder.

Eğer "host\_key\_checking = False" olarak ayarlanırsa, bu, SSH anahtar doğrulama işleminin devre dışı bırakıldığı anlamına gelir. Bu, SSH ile bağlantı kurarken sunucunun kimliğini doğrulamadan bağlanmanıza izin verir. Bu, bağlantı kurmada hızlı bir yol olabilir, ancak güvenlik riskleri taşır. Çünkü doğrulama devre dışı bırakıldığında, sunucunun gerçekten güvenilir olduğundan emin olmadan bağlantı kurabilirsiniz.

ansible webservers -a "uptime"

ansible: Bu komut, Ansible'in başlatılmasını belirtir.

webservers: Bu, hedef makinelerin bir grup adını temsil eder. "webservers" adlı bir grup tanımlandığı varsayılır ve bu grup içinde belirtilen hedef makineler bulunur. Önceden tanımlanmış bu grup, Ansible yapılandırma dosyalarında veya komut satırında belirtilmiş olmalıdır.

-a "uptime": Bu, belirli bir modül ve argümanları çalıştırmak için kullanılır. "uptime" komutu, hedef makinelerdeki sistemlerin ne kadar süredir çalıştığını gösterir. Bu komut "uptime" komutunu hedef makinelerde çalıştırmak için kullanılır.

Sonuç olarak, bu Ansible komutu, "webservers" adlı bir grup içindeki tüm hedef makinelerde "uptime" komutunu çalıştırır ve bu komutun çıktılarını toplar, böylece bu makinelerin ne kadar süredir çalıştığını görüntülersiniz. Bu, hedef makinelerin sistem durumunu kontrol etmek için kullanışlı bir komuttur.

Formun Üstü

Çıkan sonuçtaki load average:

"Load average" veya yük ortalaması, bir bilgisayar sisteminin belirli bir zaman diliminde işlemci kullanımını ifade eden bir metriktir. Yük ortalaması, sistemde çalışan işlem sayısı veya işlem sırasında bekleyen işlerin sayısı gibi faktörlere dayanarak hesaplanır.

ansible webservers -m shell -a "systemctl status sshd"

Bu komutun açıklaması şu şekildedir:

ansible: Bu komut, Ansible'in başlatılmasını belirtir.

webservers: Bu, hedef makinelerin bir grup adını temsil eder. "webservers" adlı bir grup tanımlandığı varsayılır ve bu grup içinde belirtilen hedef makineler bulunur. Önceden tanımlanmış bu grup, Ansible yapılandırma dosyalarında veya komut satırında belirtilmiş olmalıdır.

-m shell: Bu, kullanılan Ansible modülünü belirtir. "shell" modülü, komut satırında çalıştırılacak bir kabuk komutunu çalıştırmak için kullanılır.

-a "systemctl status sshd": Bu, çalıştırılacak kabuk komutunu belirtir. Bu durumda, "systemctl status sshd" komutu, hedef makinelerde SSH sunucusunun (sshd) durumunu sorgulayan bir komuttur.

Bu Ansible komutu, "webservers" grup içindeki tüm hedef makinelerde "systemctl status sshd" komutunu çalıştırır ve SSH sunucusunun (sshd) mevcut durumunu ve çalışma bilgilerini döndürür. Bu, SSH sunucusunun durumunu kontrol etmek ve hedef makinelerdeki hizmetleri izlemek için kullanışlı bir komuttur.

Formun Üstü

ansible webservers -m command -a 'df -h'

Bu komutun açıklaması şu şekildedir:

ansible: Bu komut, Ansible'in başlatılmasını belirtir.

webservers: Bu, hedef makinelerin bir grup adını temsil eder. "webservers" adlı bir grup tanımlandığı varsayılır ve bu grup içinde belirtilen hedef makineler bulunur. Önceden tanımlanmış bu grup, Ansible yapılandırma dosyalarında veya komut satırında belirtilmiş olmalıdır.

-m command: Bu, kullanılan Ansible modülünü belirtir. "command" modülü, belirtilen komutu hedef makinelerde çalıştırmak için kullanılır.

-a 'df -h': Bu, çalıştırılacak komutu belirtir. Bu durumda, 'df -h' komutu, hedef makinelerde disk kullanımını gösteren bir komuttur. Bu komut, dosya sistemi alanlarının ne kadarının kullanıldığını, boş olduğunu ve monte edildiğini gösteren çıktıyı üretir.

Bu Ansible komutu, "webservers" grup içindeki tüm hedef makinelerde 'df -h' komutunu çalıştırır ve her bir hedef makinenin dosya sistemi kullanımı hakkında bilgi toplar. Bu, hedef makinelerin disk kullanımını izlemek ve yönetmek için kullanışlı bir komuttur.

Yukarıda komutta -m shell yazsaydık ne farklılık olurdu

ChatGPT

Eğer ansible komutunuzda -m shell kullanırsanız, komutlarınızı çalıştırmak için bir kabuk (örneğin, Bash) kullanırsınız. Bu, komutunuzun hedef makinelerde bir kabukta çalıştırılmasına neden olur ve kabuk özelliklerini kullanabilmenizi sağlar. Örneğin, kabuk değişkenleri, boru hatları ve diğer kabuk özelliklerini kullanabilirsiniz.

İşte -m shell kullanmanın bazı farklılıkları:

Kabuk Özellikleri Kullanımı: -m shell kullanıyorsanız, komutlarınızı çalıştırmak için bir kabukta olduğunuz için, kabuk özelliklerini kullanabilirsiniz. Örneğin, değişkenler veya döngüler gibi.

Çalıştırma Bağımlılığı: Bazı komutlar, belirli bir kabukta çalışırken diğerlerinde çalışmayabilir. Bu nedenle, -m shell kullanarak komut çalıştırırken kabuk bağımlılıklarını ele almalısınız.

Çıktı Biçimi: -m shell kullanırsanız, komut çıktısının daha geniş veya karmaşık bir biçimde olabileceğini unutmayın. Bu, sonuçları daha fazla işlem yapmadan hemen anlamayı zorlaştırabilir.

Özetle, -m command ve -m shell kullanımı arasındaki temel fark, komutların nasıl çalıştırıldığıdır. -m command, komutları doğrudan çalıştırırken, -m shell komutları bir kabuk içinde çalıştırır ve kabuk özelliklerini kullanabilir. Hangi seçeneği kullanmanız gerektiği, yapmak istediğiniz işlemlere ve hedef makinelerinizin yapısına bağlı olacaktır. Genellikle -m command kullanmak, daha basit ve güvenli bir yaklaşım olabilir, çünkü işlemler daha az karmaşıklığa sahiptir.

ansible webservers -a 'df -h'

Yukarıda komutta –m command yazmadık ama komut yine çalışır.Varsayılan olarak –m command varmış gibi davranır.

ansible all -m copy -a "src=/home/ec2-user/testfile dest=/home/ec2-user/testfile" # node oluşturulan testfile bütün hostlara göndermek için

ansible node1 -m shell -a "echo Hello techproed > /home/ec2-user/testfile2 ; cat testfile2"

ansible all -m shell -a "ls ; cat test\*"

sudo vim /etc/ansible/hosts

[ubuntuserver]

node3 ansible\_host=<node3\_ip> ansible\_user=ubuntu #hostlara ubuntu makine ekleme

ansible all -m shell -a "echo Hello techproed > /home/ubuntu/testfile3" #sadece ubuntu makinaya yazdırır.Çünkü yolu ubuntu diğerlerinin yolu ec2-user

Yukardaki komutu ayrı ayrı yazarsak hem ec2 user hem de ubuntuya aynı anda gönderir

ansible node3 -m shell -a "echo Hello techproed > /home/ubuntu/testfile3"

ansible node1:node2 -m shell -a "echo Hello techproed > /home/ec2-user/testfile3"

Amazon Linux makinalar için Nginx kurulumu:

ansible webservers -b -m shell -a "amazon-linux-extras install -y nginx1 ; systemctl start nginx ; systemctl enable nginx" #-b sudo komutu demek

Ubuntu Makinalar için:

ansible node3 -b -m shell -a "apt update -y ; apt-get install -y nginx ; systemctl start nginx; systemctl enable nginx"

Amazon Linux makinalar için Nginx kaldırılması:

ansible webservers -b -m shell -a "yum -y remove nginx"

İki komut arası farklılıklar:

1.ansible webservers -b -m yum -a "name=nginx state=present"

2. ansible webservers -b -m shell -a "amazon-linux-extras install -y nginx1 ; systemctl start nginx ; systemctl enable nginx"

**ansible webservers -b -m yum -a "name=nginx state=present":**

-b: Bu seçenek, komutun "become" (yani süper kullanıcı veya yönetici haklarıyla çalışma) modunda çalıştırılmasını belirtir. Yani, komutun sudo veya root haklarıyla çalıştırılmasını sağlar.

-m yum: Bu, kullanılan Ansible modülünü belirtir. "yum" modülü, paket yöneticisi olarak "yum" kullanarak paketlerin yüklenmesini, kaldırılmasını veya güncellenmesini sağlar.

-a "name=nginx state=present": Bu, modüle iletilen argümanları belirtir. Bu komut, "nginx" adlı bir paketin "present" (var) durumda olmasını sağlar, yani eğer Nginx kurulu değilse, kurar.

**ansible webservers -b -m shell -a "amazon-linux-extras install -y nginx1 ; systemctl start nginx ; systemctl enable nginx":**

-b: Bu seçenek yine komutun "become" modunda çalıştırılmasını belirtir.

-m shell: Bu, kullanılan Ansible modülünü belirtir. "shell" modülü, belirtilen kabuk komutlarını hedef makinelerde çalıştırmak için kullanılır.

-a "amazon-linux-extras install -y nginx1 ; systemctl start nginx ; systemctl enable nginx": Bu, "shell" modülüne iletilen kabuk komutlarını belirtir. Bu komutlar, önce "amazon-linux-extras" aracılığıyla "nginx1" paketini yükler, ardından Nginx servisini başlatır ve başlangıçta otomatik olarak başlatılması için ayarlar.

Temel farklar:

İlk komut (yum modülü kullanarak) Ansible'ın paket yönetimini kullanarak Nginx'i kurar ve varsa günceller.

İkinci komut (shell modülü kullanarak) kabuk komutları kullanarak önce ekstra bir paket yönetimi aracını (amazon-linux-extras) kullanır ve ardından Nginx'i kurar ve ayarlar.

İkinci komut daha fazla işlem yapar ve daha fazla kontrol sağlar, ancak paket yönetimi aracı (amazon-linux-extras) kullanmaya ihtiyaç duyar. İlk komut, sadece Nginx paketini kurmak ve mevcutsa güncellemek için daha basit bir yaklaşım kullanır. Hangi yaklaşımın kullanılacağı, hedef makinelerin gereksinimlerine ve uygulama senaryonuza bağlıdır.

Ubuntu veya Amazon Linux farketmeksizin Nginx kurulumu:

ansible -b -m package -a "name=nginx state=present" all

İnventory otomatik elle oluşturulması:

vim inventory

[webservers]

web1 ansible\_host=<node1\_ip> ansible\_user=ec2-user #oluşturduğumuz inventory dosyasının içine yazılıdır

[webservers:vars]

ansible\_ssh\_private\_key\_file=/home/ec2-user/<YOUR-PEM-FILE-NAME>.pem

ansible -i inventory -b -m yum -a "name=httpd state=present" web1

ansible -i inventory -b -m yum -a "name=httpd state=absent" web1

Yukarıda nginx bizim oluşturduğumuz inventory dosyasının içine yazar –i inventory dediğimiz için. Demeseydik bu ifadeyi default olarak /etc/ansible/hosts bu dosyanın içindekilere yükler