

ANSIBLE Intro

طيب اول حاجه كدا احنا كنا بنعمل اى اصلا قبل ال Ansible ؟

تمام، قبل ما نستخدم Ansible كنا بنعمل الآتي:

-كنا بنتصل ب remote servers عن طريق SSH protocol.
-بنكتب الأوامر يدويًا على كل جهاز server لتنفيذ ال task المطلوب.
-بنعمل update لل packages المطلوبة على كل جهاز server عن طريق انا بكتب الكومند بايدي

«تخيل معايا كميه التعب والارهاق وكمان دا كان بيحتاج وقت كبير اوى عشان نقدر ننفذه ونختبره على عدد كبير من السيرفرات

👉 طب اى بقا هو Ansible ؟

ال Ansible اول ما اتذكرت الكلمه دى اتزكرت في كتاب خيال علمي وكان في الروايه دى عباره عن جهاز بيتواصل مع اى جهاز في الكون ودا فيه شبه من ال Ansible automation tool الى هنعرفها دلوقت انها بتقدر تتواصل مع اى Host في اى مكان
يكفي بس Control-Node يعني Host اعمل Install ل Ansible عليه
ومن خلاله الاقدر اشتغل مع Linux,,,Unix,,,Windows,,,Network-devices
وهو يعتبر من اقوى DevOps Tools
استحوذت عليه شركه Red-Hat سنة 2015
-فالآ Ansible هي أداة هنتحكم من خلالها في عدة سيرفرات من خلال كتابة تعليمات في ملفات YAML بسيطة بنسميها Playbooks

-بديل ما ادخل على كل جهاز بايدي وادخل الاوامر على كل جهاز لا انا هستخدم Ansible واشغل الامر الى انا عاوزه على كل السيرفرات عندي في وقت واحد

- يعني من خلال كتابة Playbook بسيط، بقدر احدد المهام الى عاوز انفذها على السيرفرات الى انا عاوزها وبعدين يقوم Ansible بتنفيذ المهام دى ويرجعي نتائجها

-باستخدام Ansible ، هنفذ الأمور بشكل أكثر كفاءة وسهولة ومرونة.

🖥️ طبعا لو عندي Infrastructure كبيره اوى اوى ومعقده ممكن نحتاج اكر من Control-Node عشان اقدر ادير ال Infrastructure دى وفي الحاله دى Red-Hat

بتديك وتقدمك ال Red-Hat Ansible Tower

«بيتيح Ansible Tower واجهة بسيطة وسهلة الاستخدام لإدارة كل أوامر Ansible بسهولة

✎ دلوقتي عندك خلفيه كويسه عن Ansible تقدر تعمله install وتشغل عليه ولو محتاج حاجه نقدر نتواصل سوا

بس قبل ماتزاكره ياريت تكون عارف YAML كويس.

15 Ansible Interview Questions - Part One

1. What is Ansible?


برنامج مفيد جداً في إدارة الأنظمة والتكوينات. يسمح لنا بكتابة سكريبتات بسيطة نقدر نستخدمها عشان نتحكم في الأجهزة والخوادم بشكل سهل ومرتب.


الحلو في الأنسبيل إنه بيستخدم لغة بسيطة ومفهومة، مش زي البرامج التانية اللي بتحتاج للغات برمجة متقدمة. فبتقدر تكتب السكريبتات بشكل سهل وتعدلها براحتك.


في الخلاصة، الأنسبيل هو أداة بتساعدك تدير وتتحكم في الأنظمة والتكوينات بشكل مرتب وبسيط.


2. What are the component of Ansible?


فيه عناصر أساسية في الأنسبيل. العناصر دي بتتكون من:

1.  ال (Control Machine): دا الجهاز اللي بتشتغل عليه أوامر الأنسبيل. ممكن تكون على جهازك الخاص أو على سيرفر مخصص للغرض ده.

2.  المتحكم (Controller): ده الجزء اللي بيشتغل على ويتحكم في العمليات. بيدير وينظم الأوامر والتكوينات ويتواصل مع العناصر الأخرى في البنية التحتية.

3.  ال (Managed Nodes): دي الأجهزة اللي بتكون مستهدفة بأوامر الأنسبيل. ممكن تكون خوادم أو أجهزة شبكة أو غيرها. الأنسبيل بيتواصل مع العناصر دي عن طريق الشبكة.

4.  ال (Inventory): ده ملف أو قاعدة بيانات فيها معلومات عن Managed Nodes. بيتم استخدامه لتعريف وتنظيم العناصر وتحديد الاتصال بهم.

5.  ال (Modules): دي الأدوات اللي بيستخدمها الأنسبيل لتنفيذ المهام على Managed Nodes. بتوفر مجموعة متنوعة من الوظائف، زي تثبيت البرامج وتكوين الإعدادات وإدارة الملفات وغيرها.

العناصر دي هي اللي بتشتغل مع بعضها في الأنسبيل وتساعد في إدارة التكوينات والتشغيل الآلي وغيرها من

3. What is Inventory in Ansible?

ال Inventory في الأنسبل هو قائمة بالأجهزة المستهدفة التي بتشتغل عليها. يعني بتحدد فيها الأجهزة التي الأنسبل هيتعامل معاها.

الفائدة من ال Inventory إنها بتسهل عليك إدارة الأجهزة المستهدفة (Managed Nodes) لو عاوز تعمل أوامر الأنسبل على مجموعة محددة من الأجهزة، بتقدر تحدد الأجهزة دي في ال Inventory وتشتغل عليهم بسهولة.

4. What are the types of Inventories in ansible?

1- Static Inventory:

ده ببساطة قائمة بأسماء الأجهزة المستهدفة بتاعتك، ممكن تكون في ملف نصي زي "inventory.txt" افتحها وأكتب فيها أسماء الأجهزة وعناوينهم زي الأمثلة دي:

```
server1 ansible_host=192.168.1.10
```

```
server2 ansible_host=192.168.1.11
```

كدا أنت عملت Static Inventory التي فيها جهازين اسمهم server1 و server2 وعناوينهم IP. بعددين تقدر تستخدم الأسماء دي في ال Playbooks

2- Dynamic Inventory:

ده لما تكون عندك بيئة متغيرة وعاوز توليد قائمة الأجهزة تلقائياً. مثلاً، لو مستخدم Cloud AWS وعاوز تشتغل على ال EC2، هتستخدم ال Dynamic Inventory.

في المثال ده، هنستخدم ال EC2 Inventory Plugin التي بيولد قائمة الأجهزة ال EC2 تلقائياً.

```
ansible -i aws_ec2.yml all -m ping
```

ده مثال بسيط على استخدام ال Dynamic Inventory في تشغيل أمر ping على كل الأجهزة ال EC2.

3- External Inventory:

ده لما تستخدم مصادر خارجية لجلب قائمة الأجهزة. ممكن تستخدم أدوات مثل أدوات الشبكات مثل ناجيوس (Nagios) أو أدوات الاكتشاف التلقائي. فبالطريقة دي، بتقدر تحدث قائمة الأجهزة تلقائيًا بناءً على المعلومات اللي بتلقاها من الأداة الخارجية.

ده مثال على استخدام ال External Inventory بناءً على أداة الشبكة: nagios

```
ansible -i nagios\_inventory.py all -m ping
```

ده بيستخدم الأداة الخارجية [nagios_inventory.py](#) لجلب قائمة الأجهزة وبعدين يعمل ping على كل الأجهزة.

5. What is Configuration Management?

هي عبارة عن طريقة لإدارة الإعدادات والتغييرات على الأنظمة وال hosts والشبكات. بتساعدنا على إدارة كل الحاجات اللي بتخص التكوين، زي التنصيب والإعدادات وتحديث البرامج وتوزيع الملفات.

فالأنسبيل هو برنامج بيسهل علينا إدارة التكوين. بنقدر نكتب سكريبتات صغيرة وسهلة بتساعدنا نتعامل مع الأنظمة بشكل سهل ومنظم. السكريبتات دي بتكون بتحدد المهام اللي عايزين نعملها، زي التنصيب والإعدادات والتحديثات، وبعدين نشغل السكريبتات وهي بتنفذ التغييرات دي على الأنظمة.

6. Difference between hosts & groups?

في `Ansible`، تستخدم `hosts` لتحديد ال hosts الى على `Ansible` عشان نتواصل معاها وننفذ الأوامر عليها. بنعرف ال `hosts` باستخدام `IP` أو أسماء ال host أو عناوين. `URL`

على الجانب الآخر، بنستخدم `groups` لتحديد مجموعه من ال hosts التي على `Ansible` عشان نتواصل معاها وننفذ الأوامر عليها. بنعرف ال group of hosts باستخدام أسماء بتمثل وظيفة محددة لل hosts مثل `web_servers` أو `database_servers` أو `load_balancers` وما إلى ذلك.

باستخدام `groups`، هنعهد الأوامر الى هنفذها مرة واحدة على جميع ال hosts في ال group، بدل ما انفذ الاوامر بإيدى على كل host على سبيل المثال، هنعهد `web_servers` ك group ونستخدمها عشان ننفذ الاوامر مره واحده بدل ما ننفذها على كل host لوحده.

7. What is Roles ?

في `Ansible` ، ال `Roles` هيا ملفات هتسهل عليك كتابة الأكود وتنظيمها على شكل مجموعة وظيفية معينة .

يعني ال `Role` بتحتوي على مجموعة من المهام والمتغيرات والملفات والقوالب بشكل منظم وهدفها تقديم وظيفة محددة مثل `Install hosts or programs or services on hosts`

باستخدام ال `Roles` ، هتقدر تعيد استخدام الأكود اللي كتبته بشكل سهل ومنظم وبدون تكرار كتابة الأكود المشابهة في كل ملف، وكمان هتساعدك في الحفاظ على ترتيب الأكود وسهولة فهمها وتعديلها في المستقبل.

8. How to install a Role?

طريقة تثبيت ال Role في Ansible بسيطة وسهلة، وهي عبارة عن إضافة اسم ال Role إلى ملف `requirements.yml` اللي بتنشئه في مجلد ال `playbook`.

أول حاجة لازم تعرفها هي اسم ال Role اللي عايز تثبته. بعد كدا، هتحتاج تنشئ ملف `requirements.yml` في مجلد ال `playbook` اللي بتشتغل عليه.

في الملف ده، هتضيف اسم ال Role اللي عايز تثبته، ولو فيه متغيرات خاصة بال Role بتقدر تحدددها في الملف كمان.

مثال بسيط على محتوى ملف `requirements.yml`:

```
name: webserver -  
src: git@github.com:username/webserver-role.git
```

الأول عدد مرات تكرار ال Role والثاني اسم ال Role اللي عايز تثبته، والثالث مسار ال Role يا من ال git أو المسار اللي عندك.

بعد كدا، هتشتغل الأمر التالي في مجلد ال `playbook` اللي عليها ال Role:
`ansible-galaxy install -r requirements.yml`

الأمر ده هيثبت ال Role اللي موجودة في الملف `requirements.yml` وهتلاقى فيها في مجلد ال `roles` اللي بجانب مجلد ال `playbook`.

9. How to install multiple roles?

يمكن تثبيت العديد من الـ `Roles` في `Ansible` بكل سهولة باستخدام ملف `requirements.yml` والأمر `ansible-galaxy`.

بنكتب اسم كل `Role` عاوزين نثبته في ملف `requirements.yml` في سطر منفصل، مع تحديد أي متغيرات إضافية لو فيه.

مثال بسيط على محتوى ملف `requirements.yml` اللي بيحتوي على تثبيت عدة `Roles`:

```
...
name: webserver -
src: git@github.com:username/webserver-role.git

name: database -
src: git@github.com:username/database-role.git
...
```

بعد إنشاء ملف `requirements.yml` الخاص بالـ `Roles` المطلوبة، بيمنك تنفيذ الأمر التالي لتثبيتها:

```
...
ansible-galaxy install -r requirements.yml
...
```

سيتم تنزيل وتثبيت جميع الـ `Roles` المذكورة في ملف `requirements.yml` بشكل تلقائي، ويمكن العثور على الملفات المثبتة في مجلد `roles` في نفس المجلد الذي يحتوي على ملف الـ `playbook`.

10. Where is the Ansible Configuration file located ?

ملف تهيئة أنسيبل هو ملف يستخدم لتكوين وتخصيص أنسيبل ويوجد هذا الملف في مسار محدد على النظام.

عادةً ما يكون مسار ملف تهيئة أنسيبل هو `/etc/ansible/ansible.cfg`. ويمكن العثور على الملف باستخدام الأمر `ls` أو `cat`، ويتم تحديد المسار الكامل للملف في النتيجة. على سبيل المثال، يمكن استخدام الأمر التالي للعثور على Ansible Configuration file

```
...
ls /etc/ansible/ansible.cfg
...
```

وتقدر تعدل في الملف دا باستخدام مثلاً vim أو nano عشان نخصص تكوين أنسيبل وفقًا للاحتياجات الخاصة بيينا.

11. How to create roles ?

يمكن إنشاء الـ `Roles` في `Ansible` باستخدام الأمر `ansible-galaxy`، الذي يقوم بإنشاء هيكلية الملفات الأساسية للـ `Role` بشكل تلقائي.

لإنشاء `Role` جديد، يتم تنفيذ الأمر التالي:

```
ansible-galaxy init <role_name>
```

حيث ``<role_name>`` هو اسم الـ `Role` إلى عاوزين نثبته.

بعد تنفيذ الأمر، يتم إنشاء الملفات الأساسية للـ `Role` في مجلد جديد باسم الـ `Role` المحدد. يشمل هذا المجلد الملفات الأساسية التالية:

- ``tasks/main.yml``: ملف يحتوي على المهام الأساسية التي يقوم الـ `Role` بها.
- ``defaults/main.yml``: ملف يحتوي على القيم الافتراضية للمتغيرات التي يستخدمها الـ `Role`.
- ``meta/main.yml``: ملف يحتوي على معلومات الـ `Role` مثل اسم المؤلف والإصدار.
- ``files``: مجلد يحتوي على ملفات يستخدمها الـ `Role`.
- ``templates``: مجلد يحتوي على قوالب (templates) التي يستخدمها الـ `Role`.

يمكن للمستخدم إضافة ملفات إضافية أو مجلدات إلى هيكلية الـ `Role` حسب الحاجة.

بعد إنشاء الـ `Role`، يمكن استخدامه في أي مجموعة من المهام (playbook) لتنفيذ المهام المطلوبة.

12. What are the different ways other than SSH by which Ansible?

الطرق المختلفة عن SSH التي يمكن تشغيل بيها أنسيبل

أنسيبل يستخدم عشان الأتمتة عن بُعد وعادة بتشغل من خلال SSH. بس فيه طرق تانية ممكن تستخدمها بدل من SSH عشان تشغل أنسيبل.

من بين الطرق اللي ممكن تستخدمها دي:

1. تشغيل Ansible على نفس الجهاز اللي عايز تتحكم فيه: ممكن تشغل Ansible على نفس الجهاز اللي عايز تتحكم فيه وتستخدمه عشان تدير الجهاز ده نفسه. وفي الحالة دي مش بتحتاج SSH.

2. استخدام API: ممكن تستخدم API عشان تتحكم في الخوادم والأنظمة بدل من استخدام SSH. وباستخدام API، ممكن تشغل الأوامر وتتحكم في الخوادم والأنظمة عن بعد.

3. استخدام بروتوكولات ثانية: غير SSH، ممكن تستخدم بروتوكولات ثانية زي Telnet و Rlogin وغيرها لتشغيل Ansible على Remote hosts

4. استخدام البرمجيات الإضافية: ممكن تستخدم البرمجيات الإضافية زي Paramiko و PyCrypto وغيرها عشان تشغل Ansible بدون ما تحتاج ل SSH.

ملحوظة: لازم تاخذ بالك من الأمان لو استخدمت طرق ثانية غير SSH عشان تشغل Ansible. لازم تستخدم بروتوكولات وبرامج آمنة وموثوقة عشان تضمن سلامة البيانات والأجهزة اللي هتتحكم فيها.

13. What is Dynamic Inventory & when we use it & for what?

هو ببساطة بيعني قائمة بال hosts بتاعتنا في ملف بيتولد بشكل أوتوماتيكي عشان نستخدمها في التحكم والتعامل مع السيرفرات والأجهزة.

في حالة ال "Dynamic Inventory"، مش لازم نكتب ال hosts يدويًا في ملف "hosts" التقليدي، بس بنعتمد على برامج أو سكربتات بتنشئ ال inventory بشكل أوتوماتيكي بناءً على البنية التحتية الموجودة. ممكن نستخدمها لما بنشغل على بيئة سحابية أو بيئة تحتوي على عدد كبير من السيرفرات الي بيتغير عنوانهم مثلاً باستمرار.

بيكون ال "Dynamic Inventory" في شكل ملف بيتولد بتنفيذ سكربت أو أمر، وبيحتوي على بيانات ال hosts الي احتاجناها. وبعدين لما نحتاج نشغل مع ال Ansible (أو أي أداة ثانية)، بنقول للأداة ابحث عن ال hosts دي في الملف ده، وهيتعامل معاهم بناءً على البيانات الي فيه.

بالتالي، بنستخدم ال "Dynamic Inventory" عشان نجعل عملية إدارة وتحكم السيرفرات والبنية التحتية أكثر سهولة ومرونة، خصوصاً لو البنية التحتية بتتغير بشكل مستمر وبيحتوي على عدد كبير من السيرفرات. بيعتبر ال "Dynamic Inventory" أداة قوية في الأتمتة والتشغيل الآلي لعمليات الإدارة.

مثال على استخدام Dynamic Inventory في Ansible: لو عندنا مؤسسة بتستخدم AWS لإنشاء السحابة الخاصة بها، فهنستخدم Dynamic Inventory عشان نجتمع معلومات عن الموارد الموجودة في AWS مثل ال EC2 instances وال VPCs وال Subnets. يمكن ل Ansible استخدام هذه المعلومات لتحديث قائمة الموارد الخاصة بها وتنفيذ السياسات والأوامر المختلفة عليها.

يمكن جمع المعلومات باستخدام Dynamic Inventory في Ansible من مصادر مختلفة. يمكن أن تشمل هذه المصادر:

1- مزودي السحابة مثل AWS، Azure، Google Cloud Platform

2- أنظمة إدارة التكوين مثل Puppet، Chef، Salt

3- أنظمة إدارة ال Virtualization مثل VMware، OpenStack

- 4-أنظمة إدارة ال Network مثل Cisco ACI ، Juniper Contrail
- 5-أي مصادر أخرى توفر API للحصول على معلومات الموارد.

14. Can connect to remote hosts?

أيوا، ممكن نوصل للخوادم البعيدة باستخدام أنسيل. أنسيل بيقدر يتصل بالخوادم البعيدة عن طريق بروتوكول SSH.

15. What is variable in Ansible?

في أنسيل، المتغيرات (Variables) هي قيم بنستخدمها عشان نخزن فيها البيانات والمعلومات اللي نحتاجها في سكربتات الأنسيل اللي عاوزين ننفذها. المعلومات دي بتبقى مثلاً أسماء ال hosts اللي عاوزين نشتغل عليها، أو ال ports اللي بنفتحها، أو أي معلومات ثانية بنحتاجها في السكربت .

بشكل عام، بنستخدم المتغيرات عشان نجعل السكربتات أكثر تكيفاً وإعادة استخداماً، لأن بنقدر نغير القيم اللي في المتغيرات بشكل سريع ونشغل السكربت بنفس الطريقة على بيئات مختلفة .

مثال على استخدام المتغيرات في أنسيل: لو عندنا مجموعة من ال servers وعاوزين ننفذ سكربت يقوم بتحميل ملفات معينة عليهم ، بنستخدم متغير لتحديد أسماء ال servers اللي عاوزين نشتغل عليها. مثلاً، بنستخدم المتغير "servers" لتحديد ال servers اللي عاوزين ننفذ السكربت عليهم:

```
---
- hosts: "{{ servers }}"
  tasks:
  - name: Copy files to remote servers
    copy:
      src: /path/to/files
      dest: /path/on/remote/server
  ...
```

في المثال ده، بنستخدم المتغير "{{ servers }}" عشان نحدد ال servers اللي عاوزين ننفذ السكربت عليهم. وبنستخدم Ansible لإرسال الأوامر اللازمة لتنفيذ السكربت على ال servers اللي اتحددت.