

Ansible مقدمة في نشر في Omnitouch

نظرة عامة

كمنصة لأتمتة البنية التحتية لنشر حلول الشبكة Omnitouch Ansible تستخدم خدمات شبكة بطريقة متسقة وقابلة للتكرار وآلية. توفر هذه الوثيقة نظرة عامة حول (4G/5G) الخلوية الكاملة لتنظيم عمليات نشر الاتصالات المعقدة Ansible كيفية استغلالنا لـ

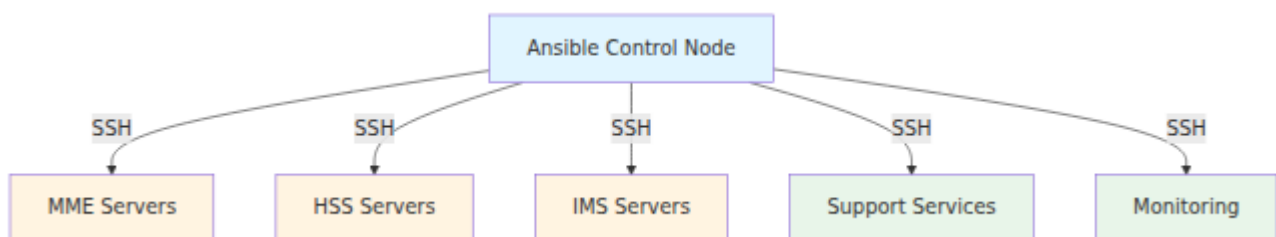
ما هو Ansible؟

هو أداة أتمتة مفتوحة المصدر تتيح لك Ansible

- تكوين الأنظمة
- نشر البرمجيات
- تنظيم سير العمل المعقد
- إدارة البنية التحتية ككود

من أنها Ansible نهجًا تصريحيًا - حيث تصف **الحالة المرغوبة** لأنظمتك، ويتأكد Ansible يستخدم. تصل إلى تلك الحالة

OmniTouch Ansible كيف تستخدم



المفاهيم الرئيسية

1. الجرد (ملفات المضيفين)

يحدد **ما** الأنظمة التي يجب إدارتها. يحتوي كل نشر للعميل على ملف مضيفين يصف:

- جميع الآلات الافتراضية في الشبكة
- الخاصة بها IP عناوين
- تكوين الشبكة
- معلومات محددة للخدمة

ملفات المضيفين هي ما ستعمل به لتعريف شبكتك.

انظر: **تكوين ملف المضيفين**

2. الأدوار

يحدد **كيف** يتم تكوين كل مكون. الأدوار هي وحدات قابلة لإعادة الاستخدام تحتوي على:

- المهام (خطوات التنفيذ)
- القوالب (قوالب ملفات التكوين)
- المعالجات (الإجراءات التي يتم تفعيلها بواسطة التغييرات)
- المتغيرات (قيم التكوين الافتراضية)

إلخ, `omnidra`, `omnipgwc`, `omnisgwc`, `omnihss` OmniCore: أمثلة على الأدوار لمكونات

بينما يمكنك تعديلها، هناك عمومًا طرق أنظف لإجراء أي ONS، يتم تعريف هذه من قبل فريق. تعديلات قد تحتاجها من داخل ملف المضيفين الخاص بك.

3. دفاتر التشغيل

تنظم **متى** و **أين** يتم تطبيق الأدوار

```
- name: Deploy EPC Core
hosts: mme
roles:
  - common
  - omnimme
```

نستخدم هذه أساسًا كمجموعات للأدوار.

4. متغيرات المجموعة

توفر تكوينًا خاصًا بالعميل يتجاوز الافتراضات الخاصة بالدور. هنا يحدث تخصيص العميل دون تعديل الأدوار الأساسية.

انظر: متغيرات المجموعة والتكوين

بنية النشر



عملية النشر

1. تعريف البنية التحتية

:قم بإنشاء ملف مضيفين يصف طوبولوجيا شبكتك

للحصول على إرشادات حول IP **ملاحظة تخطيط**: قبل تعريف البنية التحتية، راجع **معيّار تخطيط** وتنظيم الشبكات الفرعية، IP تقسيم الشبكة، وتخصيص عنوان.

Proxmox على VM/LXC انظر **نشر** Proxmox، إذا كنت تنشر على **Proxmox مستخدمو** حاويات تلقائيًا/VMS لتوفير.

انظر: تكوين ملف المضيفين و مرجع التكوين

```
mme:
  hosts:
    customer-mme01:
      ansible_host: 10.10.1.15
      mme_code: 1
```

2. تخصيص التكوين

قم بتعيين المتغيرات الخاصة بالعمل في `group_vars`:

```
plmn_id:
  mcc: '001'
  mnc: '01'
customer_name_short: customer
```

أضف رابطاً هنا إلى مرجع التكوين لقائمة كاملة - #ToDo

3. تشغيل دفاتر التشغيل

نشر الشبكة:

```
ansible-playbook -i hosts/customer/host_files/production.yml
services/epc.yml
```

4. النشر الآلي

بـ Ansible سيقوم:

- Proxmox/VMware إذا كنت تستخدم تكامل VMs إنشاء/توفير
- تكوين الشبكات
- APT تثبيت حزم البرمجيات من ذاكرة التخزين المؤقت
- نشر كود التطبيق
- تكوين الخدمات بإعدادات العمل
- بدء الخدمات
- التحقق من النشر

المكونات الرئيسية التي نقوم بنشرها

OmniCore (G/5G منصة النواة الحزمية 4)

- **OmniHSS** - خادم المشتركين المنزلي
- **OmniSGW** - بوابة الخدمة (الطبقة التحكمية)
- **OmniPGW** - بوابة الحزم (الطبقة التحكمية)
- **OmniUPF** - وظيفة الطائرة المستخدمة
- **OmniDRA** - Diameter وكيل توجيه
- **OmniTWAG** - الموثوق بها WLAN بوابة الوصول

انظر: <https://docs.omnitouch.com.au/docs/repos/OmniCore>

OmniCall (منصة الصوت والرسائل)

- **OmniCall CSCF** - وظيفة التحكم في جلسة المكالمات (P-CSCF، I-CSCF، S-CSCF)
- **OmniTAS** - IMS خادم تطبيق (VoLTE/VoNR خدمات)
- **OmniMessage** - مركز الرسائل القصيرة (SMS-C)
- **OmniMessage SMPP** - SMPP دعم بروتوكول
- **OmniSS7** - SS7 مكونات الإشارة (STP، HLR، CAMEL)
- **VisualVoicemail** - وظيفة البريد الصوتي

انظر: <https://docs.omnitouch.com.au/docs/repos/OmniCall>

OmniCharge/OmniCRM

- إدارة علاقات العملاء، التسجيل الذاتي، الفوترة - **CRM منصة**

انظر: <https://docs.omnitouch.com.au/docs/repos/OmniCharge>

خدمات الدعم

- **DNS** - الشبكة DNS حل
- **خادم الترخيص** - إدارة الترخيص

- **المراقبة** - Prometheus, Grafana

انظر: **نظرة عامة على بنية النشر**

إدارة الحزم

:نستخدم نموذج توزيع حزم هجين

مسبقة التجميع APT حزم

:Debian كحزم Omniture يتم توزيع جميع برمجيات (.deb files)

- الخاص بنا CI/CD تم بناؤها من المصدر في خط أنابيب
- تم إصدارها واختبارها
- مستضافة على مستودعات الحزم

APT نظام ذاكرة التخزين المؤقت

:يمكن للعملاء الاختيار بين

1. **المحلية** - مرآة للحزم المطلوبة في الموقع للنشر **APT ذاكرة التخزين المؤقت** غير المتصل
2. **المستودع العام** - الوصول المباشر إلى مستودع الحزم المستضاف من قبل Omniture

APT انظر: **نظام ذاكرة التخزين المؤقت**

إدارة الترخيص

:تراخيص صالحة تُدار من خلال خادم ترخيص مركزي Omniture تتطلب جميع مكونات برمجيات

- تتحقق المكونات من صلاحية الترخيص عند بدء التشغيل
- يتم تمكين/تعطيل الميزات بناءً على الترخيص
- يمكن أن يكون خادم الترخيص محليًا أو مستضافًا في السحابة

انظر: **خادم الترخيص**

فوائد هذا النهج

القابلية للتكرار

Ansible يمكن أن تنشر نفس دفاتر التشغيل الخاصة بـ

- مختبرات التطوير
- بيئات الاختبار
- الشبكات الإنتاجية
- مواقع العملاء

الاتساق

.يستخدم كل نشر نفس التكوينات المختبرة، مما يقلل من الأخطاء البشرية

التحكم في الإصدار

Git: يتم تعريف البنية التحتية ككود في

- تتبع جميع التغييرات
- مراجعة قبل النشر
- التراجع إذا لزم الأمر

التخصيص دون تعقيد

.دون تعديل الأدوار الأساسية `group_vars` يمكن للعملاء تخصيص نشرهم من خلال

النشر السريع

.نشر شبكة خلوية كاملة في ساعات بدلاً من أيام أو أسابيع

البدء

المتطلبات المسبقة

وتشيت Python تحتاج إلى إعداد بيئة افتراضية، Ansible قبل تشغيل دفاتر التشغيل الخاصة بـ التبعيات المطلوبة.

1. إنشاء بيئة افتراضية Python

Ansible معزولة لنشر Python قم بإنشاء بيئة:

```
python3 -m venv .venv
```

2. تفعيل البيئة الافتراضية

قم بتفعيل البيئة الافتراضية:

```
source .venv/bin/activate
```

استخدم، Windows على:

```
.venv\Scripts\activate
```

3. تثبيت الحزم المطلوبة

requirements.txt قم بتثبيت جميع التبعيات من ملف:

```
pip install -r requirements.txt
```

Omnitouch اللازمة لأتمتة نشر Python وجميع حزم Ansible سيقوم هذا بتثبيت

يمكنك إلغاء تفعيلها. Ansible **ملاحظة:** احتفظ بالبيئة الافتراضية مفعلة كلما قمت بتشغيل أوامر deactivate عند الانتهاء عن طريق تشغيل

خطوات النشر

1. راجع تكوين ملف المضيفين لفهم كيفية تعريف شبكتك
2. تعرف على متغيرات المجموعة للتخصيص
3. لإدارة الحزم APT افهم نظام ذاكرة التخزين المؤقت
4. راجع بنية النشر لترى كيف تتناسب كل شيء معًا
5. انشر!

الخطوات التالية

- IP خطط بنية شبكتك وتخصيص - IP معيار تخطيط
- تكوين ملف المضيفين - تعلم كيفية تعريف طوبولوجيا شبكتك
- افهم توزيع الحزم - APT نظام ذاكرة التخزين المؤقت
- خادم الترخيص - تعرف على إدارة الترخيص
- نظرة عامة على بنية النشر - انظر الصورة الكاملة
- تكوين متغيرات المجموعة - خصص نشرك
- دفاتر التشغيل المساعدة - أدوات تشغيلية لفحوصات الصحة، النسخ الاحتياطي، والصيانة

وتوزيع الحزم APT مستودع

نظرة عامة

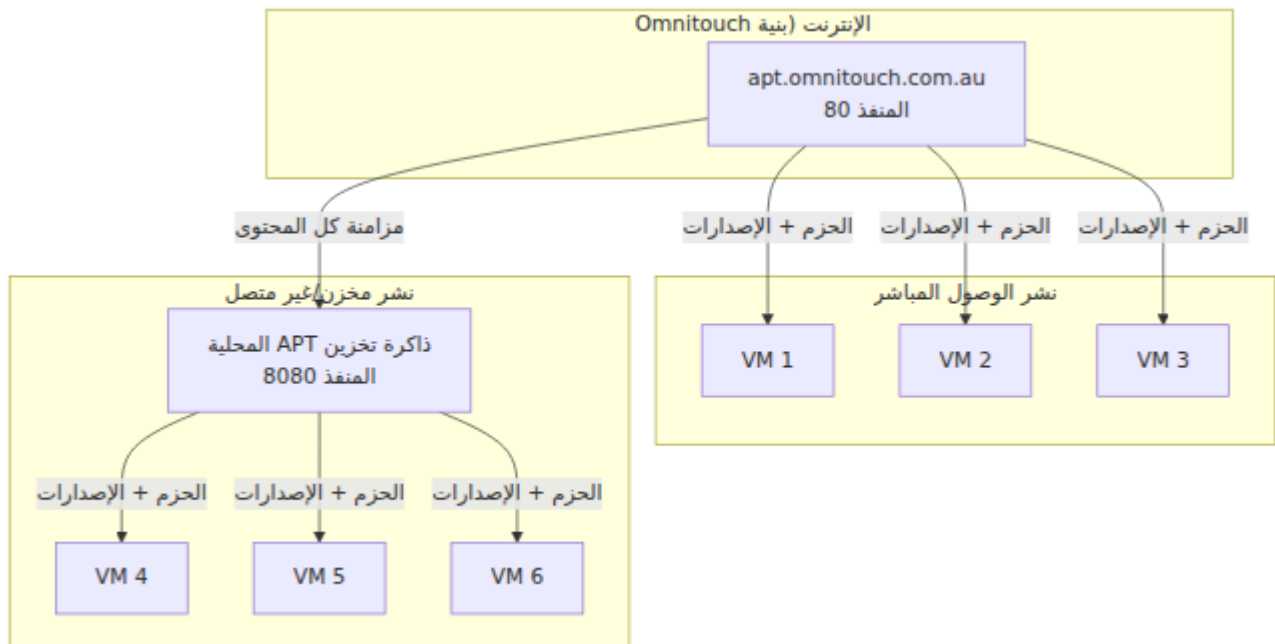
:توزيع الحزم لجميع النشر. يتم تقديم نوعين من المحتوى Omnitouch من APT يوفر نظام

1. **حزم APT** المثبتة عبر Debian حزم `apt install`
2. (مصدري Prometheus، إصدارات ثنائية — ثنائيات مُعدة مسبقًا يتم تنزيلها مباشرة، إلخ.)

:يدعم النظام نموذجين للنشر

1. **الوصول المباشر** — تقوم الآلات الافتراضية بسحب الحزم مباشرة من `apt.omnitouch.com.au`
2. **مرآة التخزين المحلية** — يقوم خادم محلي بمزامنة المحتوى من Omnitouch ويقدم الحزم إلى الآلات الافتراضية (للنشر غير المتصل بالإنترنت/المعزول)

الهيكلية



المحتوى المقدم

:جميع المحتويات المطلوبة للنشر APT يستضيف خادم

نوع المحتوى	الوصف	المسار
حزم Omnitouch	مبنية خصيصًا <code>.deb</code> حزم (omnihss, omnimme, إلخ.)	<code>/dists/<distro>/</code>
حزم Ubuntu	المخزنة مع جميع حزم Ubuntu التبعيات	<code>/<distro>/pool/main/</code>
إصدارات GitHub	(Prometheus, Grafana, Homer, إلخ.) ثنائيات مُعدة مسبقًا	<code>/releases/<org>/<repo>/</code>
أرشفيات المصدر	أرشفيات المصدر لتطبيقات الويب (CGrateS_UI, speedtest)	<code>/repos/</code>
حزم الطرف الثالث	Galera, FRR, InfluxDB, KeyDB, إلخ.	<code>/releases/<vendor>/</code>

متغيرات التكوين

تتحكم مجموعتان منفصلتان من المتغيرات في توزيع الحزم. فهم أغراضها أمر ضروري للتكوين الصحيح.

```
متغيرات التكوين

apt_repo
(APT مصادر حزم)

remote_apt_*
(تنزيلات ثنائية)
```

```
ما الذي يقومون بتكوينه

/etc/apt/sources.list

تنزيلات ثنائية
/releases/*
```

أغراض المتغيرات

مجموعة المتغيرات	الغرض	الاستخدام
apt_repo	يقوم بتكوين مصادر حزم APT	/etc/apt/sources.list و /etc/apt/sources.list.d/*.list
remote_apt_*	يقوم بتكوين URL عناوين لتنزيلات ثنائية	/releases/ (Node Exporter, Zabbix, Nagios, إلخ.) تنزيل الملفات من مسار

متى يتم استخدام كل مجموعة متغيرات

السيناريو	APT مصادر (apt_repo)	تنزيلات ثنائية (remote_apt_*)
use_apt_cache: true	يستخدم apt_repo.apt_server	يستخدم apt_repo.apt_server
use_apt_cache: false	يستخدم مع apt_repo.* بيانات الاعتماد	يستخدم مع remote_apt_* بيانات الاعتماد

تكون كلا مجموعتي المتغيرات مطلوبة ، `use_apt_cache: false` عند

الخيار 1: الوصول المباشر

بالنسبة للنشر الذي يتطلب اتصالاً بالإنترنت، تقوم الآلات الافتراضية بسحب الحزم مباشرة من خادم APT من Omnitouch.

متطلبات الشبكة

العام الخاص بك مدرجًا في القائمة IP **المصدر**: يجب أن يكون عنوان IP **قائمة بيضاء لعنوان** أثناء الإعداد، قدم نطاقات المصدر الخاصة بك إلى Omnitouch. APT البيضاء على خادم Omnitouch. في المقابل، ستتلقى.

- الأساسية HTTP **اسم المستخدم** و **كلمة المرور** لمصادقة
- APT لخادم **FQDN**

التالية من IP **متطلبات جدار الحماية**: يجب السماح بالوصول الخارجي إلى نطاقات Omnitouch:

النطاق	الشبكة
144.79.167.0/24	IPv4
160.22.43.0/24	IPv4
2001:df3:dec0::/48	IPv6
AS152894	ASN

التحتية Omnitouch الخدمات التي تتطلب الوصول إلى بنية:

العرض	البروتوكول	المنفذ	الخدمة
تنزيل الحزم	TCP	80	APT خادم
ل DNS حل apt.omnitech.com.au	TCP/UDP	53	APT خادم
للتحقق من صحة الترخيص NTP مزامنة الوقت	UDP	123	خادم الترخيص
للتحقق من صحة الترخيص DNS حل	TCP/UDP	53	خادم الترخيص

إلى (DNS (TCP+UDP/53 و، NTP (UDP/123)، HTTP (TCP/80) تأكد من السماح بحركة مرور Omnitech من IP نطاقات.

التكوين

```
all:
vars:
  use_apt_cache: false

  # تكوين مصادر حزم
  # apt install لأوامر /etc/apt/sources.list يقوم بتكوين
  apt_repo:
    apt_server: "apt.omnitech.com.au"
    apt_repo_username: "your-username"
    apt_repo_password: "your-password"

  # تكوين التنزيلات الثنائية
  # يستخدم لتنزيل الملفات من مسار /releases/
  remote_apt_server: "apt.omnitech.com.au"
  remote_apt_port: 80
  remote_apt_protocol: "http"
  remote_apt_user: "your-username"
  remote_apt_password: "your-password"
```

المعلومات

APT (apt_repo) مصادر حزم

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>apt_repo.apt_server</code>	سلسلة	نعم	-	اسم مضيف أو APT خادم IP عنوان
<code>apt_repo.apt_repo_username</code>	سلسلة	نعم	-	اسم مستخدم مصادقة HTTP الأساسية APT لمصادر
<code>apt_repo.apt_repo_password</code>	سلسلة	نعم	-	كلمة مرور مصادقة HTTP الأساسية APT لمصادر

(remote_apt_*) التنريلات الثنائية

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>remote_apt_server</code>	سلسلة	نعم	-	اسم المضيف أو لتنزيلات IP عنوان الثانية
<code>remote_apt_port</code>	عدد صحيح	لا	80	منفذ الخادم لتنزيلات الثانية
<code>remote_apt_protocol</code>	سلسلة	لا	http	أو (http) البروتوكول (https)
<code>remote_apt_user</code>	سلسلة	نعم	-	اسم مستخدم HTTP مصادقة الأساسية للتنزيلات
<code>remote_apt_password</code>	سلسلة	نعم	-	كلمة مرور مصادقة الأساسية HTTP للتنزيلات

عام

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>use_apt_cache</code>	Boolean	نعم	-	false يجب أن تكون للوصل المباشر

(الوصول المباشر) URL أنماط

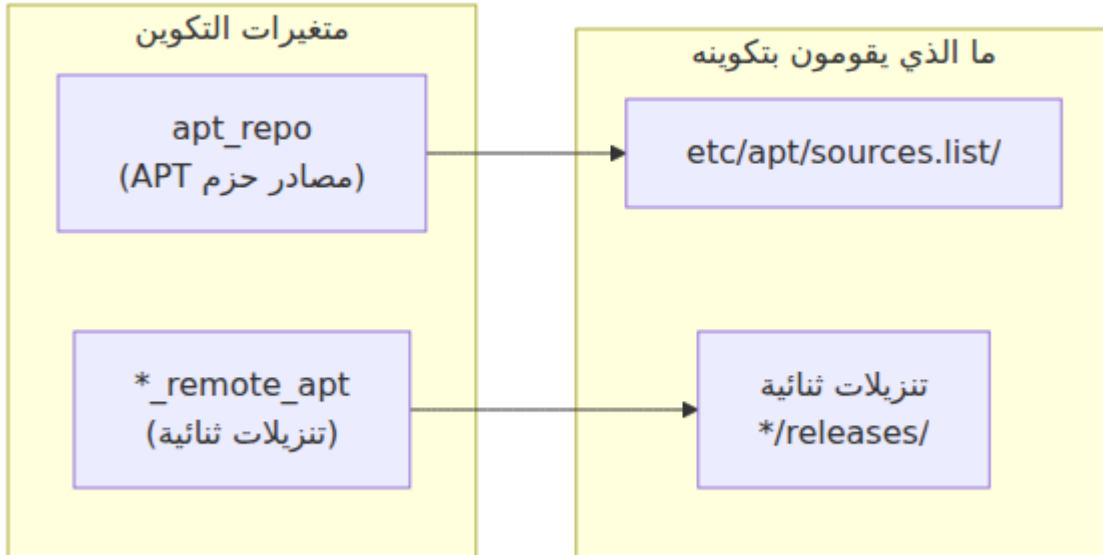
(`/etc/apt/sources.list` المكونة في) **APT مصادر حزم**:

```
deb [trusted=yes] http://{apt_repo_username}:
{apt_repo_password}@{apt_server}/ noble main
```

(`get_url` Ansible المستخدمة بواسطة مهام) **التنزيلات الثانية**:

```
http://{remote_apt_user}:  
{remote_apt_password}@{remote_apt_server}:  
{remote_apt_port}/releases/prometheus/node_exporter/node_exporter-  
1.8.1.linux-amd64.tar.gz
```

كيف يعمل



والتنزيلات APT الأساسية لكل من حزم HTTP تقوم الآلات الافتراضية بالمصادقة باستخدام مصادقة لذا لا تحتاج، (مخزنة مسبقًا) Omnitech من خادم Ubuntu الثنائية. يتم أيضًا تقديم حزم نظام Ubuntu. الآلات الافتراضية إلى الوصول إلى مرايا.

الخيار 2: مرآة التخزين المحلية

APT بالنسبة للنشر غير المتصل، المعزول، أو المحدود بالنطاق الترددي، قم بنشر ذاكرة تخزين Omnitech محلية تقوم بمزامنة كل المحتوى من

الهيكلة



التكوين

:حدد خادم التخزين في ملف المضيفين الخاص بك مع تكوين المستودع الخاص به

```
apt_cache_servers:
  hosts:
    customer-apt-cache:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # يقوم خادم التخزين بمزامنة الحزم من المستودع المعتمد
    remote_apt_server: "apt.omnitouch.com.au"
    remote_apt_port: 80
    remote_apt_protocol: "http"
    remote_apt_user: "your-username"
    remote_apt_password: "your-password"

all:
  vars:
    # يتم تعيينه تلقائيًا عند وجود مجموعة خوادم
    use_apt_cache: true
  apt_cache:
    # apt_repo.apt_server: 192.168.1.100 (أول خادم)
    # يتم اشتقاقه تلقائيًا إلى 192.168.1.100 (أول خادم)
    # تخزين
```

كيف يعمل:

- لمزامنة `remote_apt_*` **خادم التخزين** (192.168.1.100): يستخدم بيانات اعتماد `apt.omnitouch.com.au:80` الحزم من

- **apt_repo.apt_server:** جميع المضيفين الآخرين: يتم اشتقاق تلقائيًا وسحب الحزم من التخزين على المنفذ 8080 دون بيانات "192.168.1.100" اعتماد

المعلومات

(apt_repo) مصادر حزم APT

المعلومة	النوع	مطلوب	الافتراضي	لوصف
apt_repo.apt_server	سلسلة	نعم	مشتق تلقائيًا	تخزين IP عنوان ي. يتم اشتقاقه من أول مضيف في apt_cache_s ا. لم يتم تحديده
apt_repo.apt_repo_username	سلسلة	لا	-	غير مطلوب عند ندام التخزين (لا حاجة للمصادقة)
apt_repo.apt_repo_password	سلسلة	لا	-	غير مطلوب عند ندام التخزين (لا حاجة للمصادقة)

(remote_apt_*) مزامنة خادم التخزين

Omnitouch: تقوم هذه المتغيرات بتكوين كيفية مزامنة خادم التخزين للمحتوى من

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>remote_apt_server</code>	سلسلة	نعم	-	من APT خادم Omnitouch للمزامنة منه
<code>remote_apt_port</code>	عدد صحيح	لا	80	من APT منفذ خادم Omnitouch
<code>remote_apt_protocol</code>	سلسلة	لا	http	البروتوكول للمزامنة الاتصال
<code>remote_apt_user</code>	سلسلة	نعم	-	بيانات الاعتماد للمزامنة من Omnitouch
<code>remote_apt_password</code>	سلسلة	نعم	-	بيانات الاعتماد للمزامنة من Omnitouch

عام

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>use_apt_cache</code>	Boolean	لا	true	true يتم تعيينه تلقائيًا إلى عند وجود مجموعة <code>apt_cache_servers</code>
<code>apt_cache_port</code>	عدد صحيح	لا	8080	المنفذ الذي يستمع عليه خادم التخزين المحلي

وضع التخزين (URL أنماط)

(`/etc/apt/sources.list` المكونة في) **APT مصادر حزم**:

```
deb [trusted=yes] http://192.168.1.100:8080/noble noble main
```

التنزيلات الثنائية (Ansible `get_url` المستخدمة بواسطة مهام):

```
http://192.168.1.100:8080/releases/prometheus/node_exporter/node_exporter-1.8.1.linux-amd64.tar.gz
```

APT `[trusted=yes]` لا حاجة لبيانات اعتماد للوصول إلى التخزين - يستخدم تكوين

نشر التخزين

1. (بسرعة قرص +50 جيجابايت LXC آلة افتراضية أو حاوية) قم بتوفير خادم التخزين

2. :قم بتشغيل برنامج إعداد التخزين

```
ansible-playbook -i hosts/customer/production.yml  
services/apt_cache.yml
```

3. تحقق من التخزين عن طريق تصفح `http://192.168.1.100:8080/`

ما الذي يتم مزامنته

wget باستخدام تنزيل Omnitouch من APT تقوم مرآة التخزين بمزامنة كل المحتوى من خادم التكراري:

apt.omnitouch.com.au				
حزم Omnitouch .deb /pool/main/	التبعيات + حزم Ubuntu /noble/pool/main/	إصدارات GitHub /releases/	أرشيفات المصدر /repos/	بيانات التعريف APT /dists/

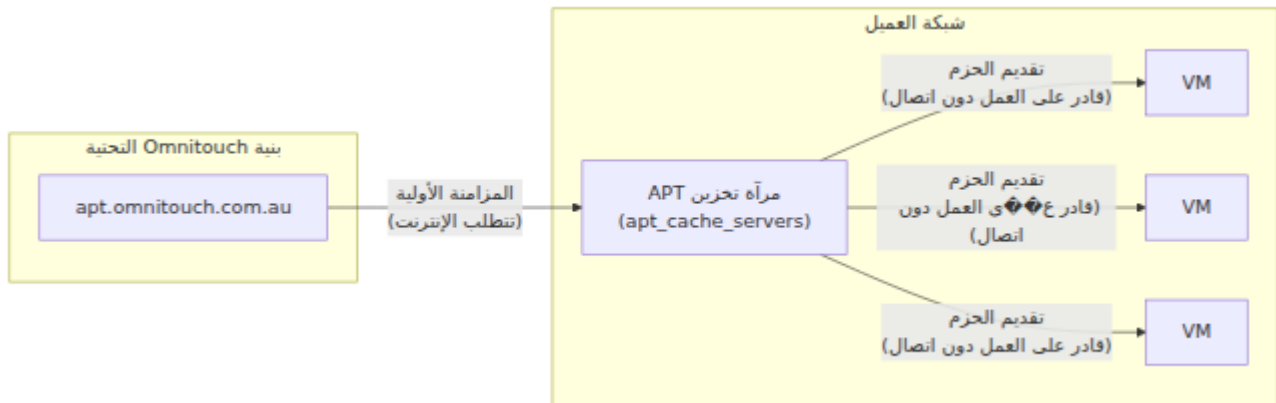
مرآة التخزين المحلية				
حزم Omnitouch .deb	التبعيات + حزم Ubuntu	إصدارات GitHub	أرشيفات المصدر	بيانات التعريف APT

:مجلدات المحتوى التي تم مزامنتها

المسار	المحتوى
<code>/dists/<distro>/</code>	(ملفات الحزم، ملفات الإصدار) APT بيانات تعريف مستودع
<code>/pool/main/</code>	Omnitouch المخصصة من deb. حزم
<code>/<distro>/pool/main/</code>	وجميع التبعيات Ubuntu حزم
<code>/releases/</code>	إصدارات GitHub (Prometheus, Grafana, Zabbix, إلخ.)
<code>/repos/</code>	(Erlang, Elixir, CGrateS_UI, إلخ.) أرشيفات المصدر

.بعد المزامنة الأولية، يمكن أن يقدم التخزين جميع الحزم دون اتصال بالإنترنت

كيف يعمل



الأساسية لتنزيل كل المحتوى من HTTP مع مصادقة `wget --recursive` تستخدم مرآة التخزين تقوم المزامنة اللاحقة بتنزيل الملفات الجديدة/المعدلة فقط. من Omnitech. APT خادم (توقيت).

التكوين التلقائي

في جردك، يقوم النظام تلقائيًا `apt_cache_servers` عندما توجد مجموعة

- لجميع المضيفين (ما لم يتم تجاوز ذلك صراحة) `use_apt_cache: true` بتعيين
- الخاص بأول خادم تخزين IP من عنوان `apt_repo.apt_server` باشتقاق

مثال على التكوين الأدنى

```
apt_cache_servers:
  hosts:
    apt-cache-01:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # Omnitouch يقوم خادم التخزين بمزامنة المحتوى من مستودع
    remote_apt_server: "apt.omnitouch.com.au"
    remote_apt_user: "your-username"
    remote_apt_password: "your-password"
```

ما يحدث تلقائيًا:

- يحصل جميع المضيفين (باستثناء خادم التخزين) على `use_apt_cache: true`
- يحصل جميع المضيفين (باستثناء خادم التخزين) على `apt_repo.apt_server: "192.168.1.100"`
- دون بيانات اعتماد `http://192.168.1.100:8080/` يسحب جميع المضيفين من
- يقوم خادم التخزين بمزامنة الحزم من `http://your-username:your-password@apt.omnitouch.com.au/`

تجاوز السلوك التلقائي

:لإجبار الوصول المباشر حتى مع تعريف خوادم التخزين

```
all:
  vars:
    use_apt_cache: false # إجبار الوصول المباشر حتى مع تعريف خوادم
    التخزين

    apt_repo:
      apt_server: "apt.omnitech.com.au"
      apt_repo_username: "user"
      apt_repo_password: "pass"

    remote_apt_server: "apt.omnitech.com.au"
    remote_apt_user: "user"
    remote_apt_password: "pass"
```

ملخص التكوين

بدون APT السيناريو 1: الوصول المباشر إلى خادم (تخزين)

APT. تقوم جميع المضيفين بسحب الحزم مباشرة من خادم مستودع

```
all:
  vars:
    use_apt_cache: false

    # تستخدم من قبل جميع المضيفين - APT مصادر حزم
    apt_repo:
      apt_server: "apt.omnitech.com.au"
      apt_repo_username: "user"
      apt_repo_password: "pass"

    # التنزيلات الثنائية - تستخدم من قبل جميع المضيفين
    remote_apt_server: "apt.omnitech.com.au"
    remote_apt_port: 80
    remote_apt_protocol: "http"
    remote_apt_user: "user"
    remote_apt_password: "pass"
```

deb [trusted=yes] تقوم جميع المضيفين بإنشاء
http://user:pass@apt.omnitech.com.au/ noble main

معرف في ملف APT السيناريو 2: خادم تخزين المضيفين (تلقائي)

Ansible خادم التخزين موجود في جردك وسيتم نشره/مزامنته بواسطة

```
apt_cache_servers:
  hosts:
    cache-server:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # يقوم خادم التخزين بمزامنة الحزم من المستودع المعتمد
    remote_apt_server: "apt.omnitech.com.au"
    remote_apt_port: 80
    remote_apt_protocol: "http"
    remote_apt_user: "user"
    remote_apt_password: "pass"

# لا حاجة لتكوين في all: vars:
# apt_cache_servers كل شيء يتم اشتقاقه تلقائيًا من مجموعة
```

النتيجة:

- خادم التخزين: يقوم بالمزامنة من
http://user:pass@apt.omnitech.com.au:80/
- جميع المضيفين الآخرين: يقومون بإنشاء deb [trusted=yes]
http://192.168.1.100:8080/noble noble main (بدون بيانات اعتماد)

المخزن غير موجود في ملف APT السيناريو 3: خادم المضيفين (يدوي)

(الخاص بك Ansible غير مُدار بواسطة) يوجد خادم التخزين في مكان آخر وتم إعداداه بالفعل.

```
all:
  vars:
    use_apt_cache: true

    # توجيه جميع المضيفين إلى خادم التخزين الخارجي
    apt_repo:
      apt_server: "192.168.1.100" # لخدم التخزين الخارجي IP عنوان
      apt_repo_port: 8080         # عادةً ما يعمل التخزين على المنفذ
8080

# apt_cache_servers لا حاجة لمجموعة
# remote_apt_* (التخزين تم إعداداه بالفعل خارجيًا) لا حاجة لـ
```

النتيجة: تقوم جميع المضيفين بإنشاء `deb [trusted=yes]`
(بدون بيانات اعتماد) `http://192.168.1.100:8080/noble noble main`

مثال كامل

إليك مثال كامل يعمل يوضح تكوين خادم التخزين مع عدة مضيفين للتطبيق:

APT مجموعة خادمات 00زين

apt_cache_servers:

hosts:

customer-apt-cache:

ansible_host: 10.179.1.114

gateway: 10.179.1.1

host_vm_network: "vmbr0"

num_cpus: 4

memory_mb: 16384

proxmoxLxcDiskSizeGb: 120

vars:

يقوم خادم التخزين بمزامنة الحزم من المستودع المعتمد

remote_apt_server: "apt.omnitouch.com.au"

remote_apt_port: 80

remote_apt_protocol: "http"

remote_apt_user: "customer-username"

remote_apt_password: "customer-secure-token"

خوادم التطبيقات

hss:

hosts:

customer-hss01:

ansible_host: 10.179.2.140

gateway: 10.179.2.1

mme:

hosts:

customer-mme01:

ansible_host: 10.179.1.15

gateway: 10.179.1.1

dns:

hosts:

customer-dns01:

ansible_host: 10.179.2.177

gateway: 10.179.2.1

التكوين العالمي

all:

vars:

التكوين التلقائي (لا حاجة لتكوين يدوي)

- use_apt_cache: true (مفعّل تلقائيًا عند وجود apt_cache_servers)

```
# - apt_repo.apt_server: "10.179.1.114" (مشتق تلقائيًا من خادم  
(التخزين)
```

ما يحدث أثناء النشر:

1. خادم التخزين (10.179.1.114):

- الخاص به vars: من قسم remote_apt_* يستخدم
- يقوم بتنزيل جميع الحزم من `http://customer-username:customer-secure-token@apt.omnitouch.com.au:80/`
- nginx يقدم الحزم على المنفذ 8080 عبر

2. مضيفو التطبيقات (customer-hss01، customer-mme01، customer-dns01):

- يكتشفون تلقائيًا وجود مجموعة apt_cache_servers
- تلقائيًا use_apt_cache: true يتم تعيين
- تلقائيًا apt_repo.apt_server: "10.179.1.114" يتم اشتقاق
- deb [trusted=yes]: يتم إنشاء `http://10.179.1.114:8080/noble noble main`
- تسحب جميع الحزم من خادم التخزين (بدون حاجة لبيانات اعتماد)

تحديث التخزين

للمزامنة حزم جديدة أو تحديثات:

```
ansible-playbook -i hosts/customer/production.yml  
services/apt_cache.yml
```

بشكل تدريجي Omnitouch من APT يقوم هذا بمزامنة كل المحتوى من خادم

- Omnitouch إصدارات جديدة من حزم
- جديدة وتبعياتها Ubuntu حزم
- GitHub إصدارات جديدة من

- أرشيفات مصدر محدثة

لذا يتم تخطي الملفات غير المتغيرة الموجودة، مما `wget --timestamping` تستخدم المزامنة `services/apt.yml` لجعل إعادة ال `services/apt_cache.yml` زامنة سريعة.

هو المصدر الوحيد للحقيقة (`apt.omnitouch.com.au`) من APT **ملاحظة:** خادم أولاً لبناء/تحديث الحزم، ثم قم APT على خادم `services/apt.yml` لجميع الحزم. قم بتشغيل `services/apt_cache.yml` على مرايا التخزين لمزامنتها بتشغيل

استكشاف الأخطاء وإصلاحها

مع 401 غير مصرح به APT فشل تحديث

الأعراض:

```
http://10.179.1.115:80/noble/dists/noble/main/binary-  
amd64/Packages 401 غير مصرح به فشل في جلب
```

الأسباب المحتملة:

- `apt_cache_servers:` بدلاً من `all: vars:` في `apt_repo` تم تعريف تكوين `vars:`
- تحاول المضيفون الوصول إلى المستودع المعتمد مباشرة بدلاً من التخزين
- غير `apt_repo_password` أو كلمة المرور `apt_repo_username` اسم المستخدم صحيحة
- Omnitech من APT المصدر غير مدرج في القائمة البيضاء على خادم IP عنوان
- استخدام بيانات اعتماد التخزين للوصول المباشر أو العكس

الحل:

1. مع بيانات الاعتماد `apt_repo` في `vars:` تحقق من نطاق التكوين: تأكد من تعريف `all: vars:` وليس في `apt_cache_servers: vars:`
2. تحقق من وضع التخزين: عند استخدام التخزين، يجب أن تتصل المضيفون بخادم التخزين (المنفذ 8080)، وليس بالمستودع (المنفذ 80)

3. تحقق من المصادر المولدة: على المضيف الفاشل، تحقق من

```
/etc/apt/sources.list.d/omnitech.list
```

◦ صحيح (وضع التخزين): deb [trusted=yes]

```
http://10.179.1.114:8080/noble noble main
```

◦ غير صحيح (لديه بيانات اعتماد في المكان الخطأ): deb

```
[trusted=yes] http://user:pass@10.179.1.115:80/noble  
noble main
```

4. تحقق من أن بيانات الاعتماد صحيحة لوضع النشر الخاص بك

5. إذا Omnitech العام الخاص بك مدرج في القائمة البيضاء مع IP تأكد من أن عنوان (كنت تستخدم الوصول المباشر)

فشل التنزيلات الثنائية (Node Exporter، Zabbix، إلخ.)

في تنزيل الملفات من مسار Ansible الأعراس: فشل برنامج /releases/

الأسباب المحتملة:

- remote_apt_* لم يتم تكوين متغيرات
- remote_apt_password أو كلمة المرور remote_apt_user اسم المستخدم صححة
- use_apt_cache: false عند remote_apt_server عدم وجود

الحل:

1. remote_apt_* تأكد من تعريف جميع متغيرات
2. Omnitech تحقق من أن بيانات الاعتماد تتطابق مع تلك المقدمة من
3. يشير إلى المضيف الصحيح remote_apt_server تحقق من أن

فشل خادم التخزين في المزامنة

الأعراس: فشل برنامج خادم التخزين في تنزيل الحزم

الأسباب المحتملة:

- لا يوجد اتصال بالإنترنت لخادم التخزين
- غير صحيحة remote_apt_* بيانات اعتماد

- جدار الحماية يمنع الاتصالات الخارجية إلى Omnitouch

الحل:

1. على `apt.omnitouch.com.au` تحقق من أن خادم التخزين يمكنه الوصول إلى المنفذ 80
 2. تحقق من بيانات اعتماد `remote_apt_*`
 3. راجع قواعد جدار الحماية للوصول الخارجي.
-

الوثائق ذات الصلة

- تكوين ملف المضيفين — تكوين الجرد والمتغيرات
- مرجع التكوين — مرجع كامل للمعلومات
- بنية النشر — بنية النظام العامة
- LXC نشر خادم التخزين كحاوية — Proxmox نشر

مرجع التكوين

نظرة عامة

من خلال ملفات المضيفين. يتم تعريف OmniCore يوفر هذا المستند مرجعًا شاملاً لتكوين نشرات group_vars التكوين بشكل أساسي في ملفات جرد المضيفين مع الحد الأدنى من تجاوزات المطلوبة للنشرات الحديثة.

للحصول على وثائق محددة للمنتج، انظر:

- **OmniCore:** <https://docs.omnitouch.com.au/docs/repos/OmniCore>
- **OmniCall:** <https://docs.omnitouch.com.au/docs/repos/OmniCall>
- **OmniCharge:** <https://docs.omnitouch.com.au/docs/repos/OmniCharge>

نهج التكوين

:الحديثة نموذج تكوين مبسط OmniCore تستخدم نشرات



المبدأ الأساسي: يتم تعريف معظم التكوين مباشرة في ملف المضيفين. تتعامل القيم الافتراضية فقط للتخصيصات المحددة group_vars للدور مع معظم الإعدادات، مع استخدام

تخطيط الشبكة

:للحصول على إرشادات حول IP قبل تكوين المضيفين، راجع [معيّار تخطيط](#)

- استراتيجيات تقسيم الشبكة
- IP تخصيص عنوان
- تنظيم الشبكة الفرعية
- العام IP التعامل مع

معلومات المضيف الشائعة

لهذا hosts-file-configuration.md فقط قل للتحقق من - #ToDo

علامات محددة للخدمة

```
cdrs_enabled: True          # تمكين توليد CDR
in_pool: False              # استبعاد من مجموعة التوازن
online_charging_enabled: False # تمكين تكامل OCS
recording: True             # تمكين تسجيل المكالمات (AS)
populate_crm: False         # بالبيانات الأولية CRM ملء
```

(all:vars) المتغيرات العالمية

على إعدادات على مستوى النشر. تستخدم النشرات الحديثة متغيرات `all:vars` يحتوي قسم عالمية قليلة مع معظم التكوين في القيم الافتراضية `all:vars`ية للدور.

المتغيرات العالمية الأساسية

المصادقة والوصول

```
ansible_connection: ssh
ansible_user: root
ansible_password: password
ansible_become_password: password
```

بدلاً من كلمات المرور SSH بديل: استخدم مفاتيح

```
ansible_ssh_private_key_file: '/path/to/key.pem'
```

هوية العميل

```
customer_name_short: omnitouch
customer_legal_name: "YKTN Lab"
site_name: YKTN
region: AU
TZ: Australia/Melbourne
```

تكوين PLMN

```
plmn_id:
  mcc: '001'          # رمز الدولة المحمول (3 أرقام)
  mnc: '01'           # رمز الشبكة المحمولة (2-3 أرقام)
  mnc_longform: '001' # مع صفر بادئ (دائمًا 3 أرقام) MNC

diameter_realm: epc.mnc{{ plmn_id.mnc_longform }}.mcc{{
plmn_id.mcc }}.3gppnetwork.org
```

Diameter **الغرض**: يحدد شبكة الهاتف المحمول الخاصة بك بشكل فريد. يستخدم لبناء مجال

أسماء الشبكة

```
network_name_short: Omni
network_name_long: Omnitouch
tac_list: [10100,100]      # يمكن تجاوزها لكل الافتراضية TAC قائمة
MME)
```

في الإعدادات < الشبكة المحمولة UE **معروض**: أسماء الشبكة المعروضة على أجهزة

DNS تكوين

```
netplan_DNS: False      # بدلاً من systemd-resolved استخدام
DNS netplan
```

APT تكوين مستودع

مع `apt_cache_servers` الإعدادات الافتراضية التلقائية: عندما يتم تعريف مجموعة المضيفين:

- `False` ما لم يتم تعيينه صراحة إلى `True` افتراضياً إلى `use_apt_cache`

- `apt_repo.apt_server` لأحد خوادم التخزين المؤقت الأولى IP افتراضيًا إلى عنوان

```
# (موجودة apt_cache_servers اختياري إذا كانت مجموعة) التكوين اليدوي
use_apt_cache: True # APT استخدام التخزين المؤقت المحلي لـ
# مقابل الوصول المباشر إلى المستودع

apt_repo:
  apt_server: "10.10.1.114" # أو خادم المستودع APT خادم تخزين
  # use_apt_cache: False بيانات الاعتماد مطلوبة فقط عند
  # apt_repo_username: "omni"
  # apt_repo_password: "omni"

# تكوين تنزيلات ثنائية ومزامنة التخزين المؤقت
# عندما تكون /releases/ يستخدم لـ: (1) تنزيل الملفات الثنائية من
use_apt_cache: false
# عندما تكون Omnitouch مزامنة خادم التخزين المؤقت من (2)
use_apt_cache: true
remote_apt_server: "apt.omnitouch.com.au"
remote_apt_user: "omni"
remote_apt_password: "omni"
```

APT انظر: نظام تخزين

خادم الترخيص

```
license_server_api_urls: ["https://10.10.2.150:8443/api"]
license_enforced: true
```

انظر: خادم الترخيص

MME إعدادات

```
mme_dns: False # MME لـ DNS تمكين حل
```

SAEGW إعدادات

```
mtu: 1400 # الحد الأقصى لوحدة النقل
```

IMS إعدادات

```
ims_dra_support: False  
enable_homer: False
```

```
# DRA عبر IMS توجيه  
# Homer من SIP تمكين التقاط
```

RAN تكوين مراقبة

```
use_nokia_monitor: True
use_casa_monitor: True
install_influxdb: True

influxdb_user: monitor
influxdb_password: "secure-password"
influxdb_organisation_name: omnitouch
influxdb_nokia_bucket_name: nokia-monitor
influxdb_casa_bucket_name: casa-monitor
influxdb_operator_token: "generated-token"
influxdb_url: http://127.0.0.1:8086

enable_pm_collection: False
enable_alarm_collection: False
enable_location_collection: False
enable_ran_status_collection: True
enable_nokia_rectifier_collection: False
collection_interval_in_seconds: 120

ran_monitor:
  sql:
    user: ran_monitor
    password: "secure-password"
    database_host: 127.0.0.1
    database_name: ran_monitor
  influxdb:
    address: 10.10.2.135
    port: 8086
  nokia:
    airscales:
      - address: 10.7.15.66
        name: site-Lab-Airscale
        port: 8080
        web_password: nemuuser
        web_username: Nemuadmin
```

تكوين جدار الحماية

```
firewall:
  allowed_ssh_subnets:
    - '10.0.1.0/24'
    - '10.0.0.0/24'
  allowed_ue_voice_subnets:
    - '10.0.1.0/24'
  allowed_carrier_voice_subnets:
    - '10.0.1.0/24'
  allowed_signaling_subnets:
    - '10.0.1.0/24'
```

للتجوال DNS خوادم

```
roaming_dns_servers:
  wildcard: ['10.0.99.1']
  # DNS استنادًا إلى (محدد بواسطة الناقل PLMN)
  123456: # مثال على الناقل 1
    - '10.10.2.197'
  654321: # مثال على الناقل 2
    - '10.10.0.4'
```

(SSH مفاتيح) المستخدمين المحليون

```
local_users:
  usera:
    name: Example User A
    public_key: "ssh-rsa AAAAB3Nza..."
  userb:
    name: Example User B
    public_key: "ssh-ed25519 AAAAC3..."
```

Hypervisor تكوين

Proxmox

```
proxmoxServers:  
  customer-prxm01:  
    proxmoxServerAddress: 10.10.0.100  
    proxmoxServerPort: 8006  
    proxmoxRootPassword: password  
    proxmoxApiTokenName: AnsibleToken  
    proxmoxApiTokenSecret: "token-secret"  
    proxmoxTemplateName: ubuntu-24.04-cloud-init-template  
    proxmoxTemplateId: 9000  
    proxmoxNodeName: pve01
```

الافتراضية Proxmox إعدادات

```
proxmoxServerAddress: 10.10.0.100  
proxmoxServerPort: 8006  
proxmoxNodeName: 'pve01'  
proxmoxLxc0sTemplate: 'local:vztmpl/ubuntu-24.04-standard_24.04-  
2_amd64.tar.zst'  
proxmoxApiTokenName: DocsTest  
proxmoxLxcCores: 8  
proxmoxLxcDiskSizeGb: 20  
proxmoxLxcMemoryMb: 64000  
proxmoxLxcRootFsStorageName: SSD_RAID0  
proxmoxLxcBridgeName: vmbr0  
proxmoxTemplateName: "ubuntu-24.04-cloud-init-template"  
proxmoxStorage: SSD_RAID0  
vLabNetmask: 24  
PROXMOX_API_TOKEN: "token-secret"  
vlabRootPassword: password  
vLabPublicKey: "ssh-rsa AAAAB3..."  
mask_cidr: 24
```

VMware vCenter

```
vcenter_ip: "vcenter.example.com"
vcenter_username: "administrator@vsphere.local"
vcenter_password: "password"
vcenter_datacenter: "DC1"
vcenter_vm_template: ubuntu-24.04-model
vcenter_vm_disk_size: 50
vcenter_folder: "Omnicore"
host_vm_network: "Management"

vhosts:
  "10.0.0.23":
    vcenter_cluster_ip: 10.0.0.23
    vcenter_datastore: "datastore1 (3)"

netmask: 255.255.255.0
```

الوثائق ذات الصلة

- IP إرشادات حول بنية الشبكة وتخصيص - IP معيار تخطيط
- تكوين ملف المضيفين - كيفية هيكلة ملفات المضيفين
- group_vars تكوين المتغيرات الجماعية - متى وكيف تستخدم
- multi-NIC الثانوية وإعدادات IPs - Netplan تكوين
- بنية النشر - كيفية تفاعل المكونات
- إدارة الحزم - APT نظام تخزين
- خادم الترخيص - تكوين الترخيص

وثائق المنتج

للحصول على أدلة تشغيل مفصلة وتكوين متقدم:

- **OmniCore مكونات:**
<https://docs.omnitouch.com.au/docs/repos/OmniCore>

- **مكونات OmniCall:**

<https://docs.omnitech.com.au/docs/repos/OmniCall>

- **OmniCharge/OmniCRM:**

<https://docs.omnitech.com.au/docs/repos/OmniCharge>

نظرة عامة على بنية النشر

نظرة عامة

OmniTouch يوفر هذا المستند رؤية كاملة حول كيفية نشر برنامج الشبكة الخلوية لخدمات عاملة G/5G موضحة كيف تتناسب جميع المكونات معًا لإنشاء شبكة 4، Ansible باستخدام

IP، للحصول على تفاصيل حول وضع المكونات، وإرشادات تخصيص عناوين IP راجع [معيّار تخطيط](#) العامة IP والتعامل مع عناوين

مثال كامل على النشر

0. توفير البنية التحتية (اختياري)

قبل التكوين VMs/LXCs قم بتوفير Proxmox، للنشر على

```
# نشر VMs على Proxmox
ansible-playbook -i hosts/Customer/hosts.yml services/proxmox.yml

# (للمختبر/الاختبار فقط) LXC أو نشر حاويات
ansible-playbook -i hosts/Customer/hosts.yml
services/proxmox_lxc.yml
```

[Proxmox على VM/LXC](#) راجع: [نشر](#)

تعريف البنية التحتية (ملف المضيفين). 1.

```
# تعريف ما يجب نشره وأين  
mme:  
  hosts:  
    customer-mme01:  
      ansible_host: 10.10.1.15  
  
hss:  
  hosts:  
    customer-hss01:  
      ansible_host: 10.10.2.140  
  
# جميع المكونات الأخرى ...
```

راجع: تكوين ملف المضيفين

2. التخصيص (group_vars)

هو المكان الذي يمكننا فيه تخزين أي تجاوزات للتكوين المطلوبة على مستوى `group_vars` مجلد المضيف أو الموقع أو الشبكة.

الخاص بك، وستعيش هنا OmniMessage SMS Sc على سبيل المثال، سيكون لديك مجلد مع تكوين الخاص بك، وما إلى Diameter الخاص بك، وكل منطق توجيه TAS التي يتصل بها SIP خطوط ذلك.

راجع: تكوين المتغيرات الجماعية

3. توزيع الحزم (APT Cache)

```
# تكوين مكان الحصول على الحزم  
apt_repo:  
  apt_server: "10.254.10.223" # عنوان IP خادم أو خادم التخزين المؤقت أو خادم IP عنوان  
المستودع المباشر  
use_apt_cache: false # true = استخدام التخزين المؤقت المحلي، false =  
الوصول المباشر إلى المستودع
```

APT Cache راجع: نظام

4. تكوين الترخيص

```
# توجيه المكونات إلى خادم الترخيص
license_server_api_urls: ["https://10.10.2.150:8443/api"]
license_enforced: true
```

راجع: [خادم الترخيص](#)

5. تنفيذ النشر

على سبيل المثال، ولكن `services/twag.yml` يمكن نشر المكونات الفردية عن طريق تشغيل `services/all.yml` أو `--limit=myhost` ستتعامل مع كل شيء، ويمكنك استخدام `limit=mee,sgw` إلخ، لتحديد المضيفين الذين نعمل عليهم.

```
# نشر الشبكة الكاملة
ansible-playbook -i hosts/customer/host_files/production.yml
services/all.yml

# أو نشر مكونات محددة
ansible-playbook -i hosts/customer/host_files/production.yml
services/epc.yml
ansible-playbook -i hosts/customer/host_files/production.yml
services/ims.yml
```

الوثائق ذات الصلة

- البدء - [Ansible مقدمة في نشر](#)
- تكوين ملف المضيفين - تعريف البنية التحتية
- IP بنية الشبكة ➡ ➡ تخصيص - [IP معيار تخطيط](#)
- تكوين المتغيرات الجماعية - [التخصيص](#)
- إدارة الحزم - [APT Cache نظام](#)
- [خادم الترخيص](#) - إدارة الترخيص

وثائق المنتج

للحصول على معلومات مفصلة حول تكوين كل مكون:

- **OmniCore** (4G/5G Packet Core):
<https://docs.omnitouch.com.au/docs/repos/OmniCore>
 - OmniHSS, OmniSGW, OmniPGW, OmniUPF, OmniDRA, OmniTWAG
- **OmniCall** (الصوت والرسائل):
<https://docs.omnitouch.com.au/docs/repos/OmniCall>
 - OmniTAS, OmniCall CSCF, OmniMessage, OmniSS7, VisualVoicemail
- **OmniCharge/OmniCRM** (الفوترة):
<https://docs.omnitouch.com.au/docs/repos/OmniCharge>
- **الوثائق الرئيسية:** <https://docs.omnitouch.com.au/>

تكوين متغيرات المجموعة

نظرة عامة

هو المكان الذي تخزن فيه ملفات التكوين المخصصة التي تتجاوز القوالب `group_vars` دليل الافتراضية.

SMS، منطق توجيه، Diameter قواعد توجيه، SIP هنا تعيش تكويناتك الخاصة بالعملاء - خطوط خطط الاتصال، وأي تخصيصات أخرى حيث لا تريد استخدام التكوين الافتراضي - إنها تعيش في `group_vars`.

الموقع: `hosts/{Customer}/group_vars/`

كيف يعمل

على قوالب تكوين افتراضية. لتخصيص نشر معين، ضع ملفاتك المخصصة Ansible تحتوي أدوار وارجع إليها في ملف المضيفين الخاص بك `group_vars` في

التكوين → (إذا تم تحديده) `group_vars` قالب الدور الافتراضي → تجاوز المنشر

المثال 1: قالب تكوين مخصص (OmniMessage)

مخصصة Jinja2 بعض المكونات تقبل قوالب تكوين.

هيكل الملف

```
hosts/Customer/  
└─ group_vars/  
    └─ smsc_controller.exs
```

قالب التكوين المخصص الخاص بك #

المرجع في ملف المضيفين

```
omnimessage:
  hosts:
    customer-smsc-controller01:
      ansible_host: 10.10.3.219
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      smsc_template_config: smsc_controller.exs # ارجع إلى اسم ملف
      group_vars: قالب الخاص بك في
```

ماذا يحدث:

1. يجد Ansible `smc_template_config: smsc_controller.exs`
2. يبحث في `hosts/Customer/group_vars/smc_controller.exs`
3. يمكن استخدام Jinja2 يقوم بتهيئته باستخدام `{{ inventory_hostname }}`، `{{ plmn_id.mcc }}` إلخ،
4. ينشر إلى `/etc/omnimessage/runtime.exs`
5. يعيد تشغيل الخدمة

يتم استخدام قالب الافتراضي من الدور، `smc_template_config` بدون

انظر <https://docs.omnitech.com.au/docs/repos/OmniCall> تفاصيل التكوين:

المثال 2: مجموعات ملفات التكوين (وخطط الاتصال OmniTAS بوابات)

بعض المكونات تستخدم أدلة من ملفات التكوين.

هيكل الملف

```
hosts/Customr/
├── group_vars/
│   ├── gateways_prod/                # SIP تكوينات بوابة
│   │   ├── gateway_carrier1.xml
│   │   ├── gateway_carrier2.xml
│   │   └── gateway_emergency.xml
│   ├── gateways_lab/                 # بوابات المختبر
│   │   └── gateway_test.xml
│   └── dialplan/                     # قواعد توجيه المكالمات
│       ├── mo_dialplan.xml           # منشأ الهاتف المحمول (صادر)
│       ├── mt_dialplan.xml           # منتهى الهاتف المحمول (وارد)
│       └── emergency.xml
```

المرجع في ملف المضيفين

```
applicationserver:
  hosts:
    customer-tas01:
      ansible_host: 10.10.3.60
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      gateways_folder: "gateways_prod"  # ارجع إلى مجلد البوابة الخاص
      بك لاستخدامه على هذا المضيف
```

ماذا يحدث:

1. يجد Ansible `gateways_folder: "gateways_prod"`
2. ينسخ جميع الملفات من `hosts/Customr/group_vars/gateways_prod/` إلى `/etc/freeswitch/sip_profiles/`
3. ينسخ جميع الملفات من `hosts/Customr/group_vars/dialplan/` إلى دليل OmniTAS
4. تقوم الخدمات بتحميل التكوينات

بيئات مختلفة: استخدم مجلدات مختلفة لكل بيئة

- `gateways_folder: "gateways_lab"`

- `gateways_folder: "gateways_prod"`
- `gateways_folder: "gateways_customer_specific"`

انظر <https://docs.omnitouch.com.au/docs/repos/OmniCall> تفاصيل التكوين:

المثال 3: قالب تكوين مخصص (OmniHSS)

يقبل خادم المشتركين المنزلي قوالب تكوين وقت التشغيل المخصصة.

هيكل الملف

```
hosts/Custom/  
└─ group_vars/  
    └─ hss_runtime.exs.j2
```

المخصص الخاص بك HSS قالب تكوين #

المرجع في ملف المضيفين

```
omnihss:  
  hosts:  
    customer-hss01:  
      ansible_host: 10.10.3.50  
      gateway: 10.10.3.1  
      host_vm_network: "vmbr3"  
      hss_template_config: hss_runtime.exs.j2 # ارجع إلى اسم ملف  
# ارجع إلى اسم ملف group_vars
```

ماذا يحدث:

1. يجد Ansible `hss_template_config: hss_runtime.exs.j2`
2. يبحث في `hosts/Custom/group_vars/hss_runtime.exs.j2`
3. يمكن استخدام Jinja2 يقوم بتهيئته باستخدام `{{ inventory_hostname }}`، `{{ plmn_id.mcc }}` إلخ.
4. ينشر إلى `/etc/omnihss/runtime.exs`

يعيد تشغيل الخدمة 5.

يتم استخدام القالب الافتراضي من الدور `hss_template_config` بدون

<https://docs.omnitech.com.au/docs/repos/OmniCore> تفاصيل التكوين: انظر

المثال 4: قالب تكوين مخصص (OmniMME)

تقبل وحدة إدارة الحركة قوالب تكوين وقت التشغيل المخصصة

هيكل الملف

```
hosts/Custom/  
└─ group_vars/  
    └─ mme_runtime.exs.j2
```

المخصص الخاص بك MME قالب تكوين #

المرجع في ملف المضيفين

```
omnimme:  
  hosts:  
    customer-mme01:  
      ansible_host: 10.10.3.51  
      gateway: 10.10.3.1  
      host_vm_network: "vmbr3"  
      mme_template_config: mme_runtime.exs.j2
```

ارجع إلى اسم ملف #
group_vars القالب الخاص بك في

ماذا يحدث:

1. يجد Ansible `mme_template_config: mme_runtime.exs.j2`
2. يبحث في `hosts/Custom/group_vars/mme_runtime.exs.j2`
3. يمكن استخدام Jinja2 يقوم بتهيئته باستخدام `{{ inventory_hostname }}`, `{{ plmn_id.mcc }}` إلخ.

4. ينشر إلى `/etc/omnimme/runtime.exs`

5. يعيد تشغيل الخدمة

يتم استخدام القالب الافتراضي من الدور، `mme_template_config` بدون

<https://docs.omnitech.com.au/docs/repos/OmniCore> تفاصيل التكوين: انظر

مثال هيكل الدليل في العالم الحقيقي

```
hosts/Custom/
├─ host_files/
│   └─ production.yml          # ملف المضيفين يشير إلى ملفات
group_vars
├─ group_vars/
│   └─ smsc_controller.exs     # OmniMessage قالب مخصص لـ
│   └─ smsc_smpp.exs          # OmniMessage SMPP قالب مخصص لـ
│   └─ tas_runtime.exs.j2      # TAS قالب مخصص لـ
│   └─ hss_runtime.exs.j2      # HSS قالب مخصص لـ
│   └─ mme_runtime.exs.j2      # MME قالب مخصص لـ
│   └─ dra_runtime.exs.j2      # DRA قالب مخصص لـ
│   └─ pgwc_runtime.exs.j2     # PGW قالب مخصص لـ
│   └─ dea_runtime.exs.j2      # DEA قالب مخصص لـ
│   └─ upf_config.yaml         # UPF تكوين
│   └─ crm_config.yaml         # CRM تكوين
│   └─ stp.j2                  # SS7 STP قالب
│   └─ hlr.j2                  # SS7 HLR قالب
│   └─ camel.j2                # SS7 CAMEL قالب
│   └─ ipsmgw.j2               # IP-SM-GW قالب
│   └─ omnicore_smsc_ims.yaml.j2 # SMSC IMS تكوين
│   └─ pytap.yaml              # TAP3 تكوين
│   └─ sip_profiles/           # SIP (مجلد) بوابات
│       └─ gateway_otw.xml
│   └─ dialplan/               # قواعد توجيه المكالمات (مجلد)
│       └─ mo_dialplan.xml      # منشأ الهاتف المحمول
│       └─ mt_dialplan.xml      # منتهى الهاتف المحمول
│       └─ mo_emergency.xml     # توجيه الطوارئ
```

المعلومات الشائعة التي تشير إلى

group_vars

المعلمة	المكون	المراجع
<code>smc_template_config</code>	omnimessage	مثل Jinja2 ملف قالب <code>smc_controller.exs</code>)
<code>smc_smpp_template_config</code>	omnimessage_smpp	مثل Jinja2 ملف قالب <code>smc_smpp.exs</code>)
<code>gateways_folder</code>	applicationserver	مثل اسم المجلد <code>sip_profiles</code>)
خطط الاتصال (تلقائي)	applicationserver	ملفات <code>dialplan/</code> مجلد التوجيه XML
<code>tas_template_config</code>	applicationserver	مثل Jinja2 ملف قالب <code>tas_runtime.exs.j2</code>)
<code>hss_template_config</code>	omnihss	مثل Jinja2 ملف قالب <code>hss_runtime.exs.j2</code>)
<code>mme_template_config</code>	omnimme	مثل Jinja2 ملف قالب <code>mme_runtime.exs.j2</code>)
<code>dra_template_config</code>	dra	مثل Jinja2 ملف قالب <code>dra_runtime.exs.j2</code>)
<code>pgwc_template_config</code>	pgwc	مثل Jinja2 ملف قالب <code>pgwc_runtime.exs.j2</code>)
<code>frr_template_config</code>	omniupf	مثل Jinja2 ملف قالب <code>frr.conf.j2</code>)

المراجع	المكون	المعلمة
مثل Jinja2 ملفات قالب stp.j2، hlr.j2، camel.j2)	ss7 (أدوار متنوعة)	SS7 قوالب
ملفات التكوين المباشرة (مثل upf_config.yaml، crm_config.yaml)	مكونات متنوعة	التكوين YAML ملفات

النقاط الرئيسية

1. **على تخصيصات** - تجاوزات للتكوينات الافتراضية **group_vars** تحتوي
2. أو `smc_template_config` **المرجع بالاسم** - استخدم معلمات مثل `gateways_folder`
3. `{{ variable_name }}` باستخدام Ansible الوصول إلى أي متغير - **Jinja2 تدعم القوالب**
4. **تقوم المجلدات بنشر كل شيء** - يتم نسخ جميع الملفات في المجلدات المرجعية
5. Git إلى `group_vars` **تحكم في النسخة لكل شيء** - قم بالتزام جميع

group_vars متى تستخدم

❑ group_vars استخدم

- قوالب تكوين مخصصة للمكونات
- SIP تعريفات بوابة
- خطط توجيه المكالمات
- Diameter قواعد توجيه
- إعدادات خاصة بالعمل تتجاوز الافتراضات

❑ group_vars لا تستخدم

- استخدم ملف المضيفين - (أسماء المضيفين، IP عناوين) تكوين المضيف الأساسي

- الاختبارات لمرة واحدة - استخدم متغيرات محددة للمضيف في ملف المضيفين
 - إذا كانت قائمة group_vars التغيرات المؤقتة - قم بالتعديل على الهدف، والتزم إلى
-

الوثائق ذات الصلة

- مرجع التكوين - جميع معلومات المضيفين وما تفعله
- تكوين ملف المضيفين - كيفية هيكلة ملفات المضيفين
- **OmniCall** تكوين: <https://docs.omnitouch.com.au/docs/repos/OmniCall>
ما الذي يجب أن يكون في ملفات التكوين -
- **OmniCore** تكوين:
<https://docs.omnitouch.com.au/docs/repos/OmniCore> - تفاصيل تكوين المكونات

كُتِيبَات الأَدَوَات

نظرة عامة

يتضمن هذا المستودع العديد من كُتِيبَات الأَدَوَات للصيانة والمراقبة والمهام التشغيلية. تكمل هذه الكُتِيبَات كُتِيبَات النشر الرئيسية مع قدرات الإدارة اليومية.

أداة فحص الصحة

يظهر صحة النظام، حالة الخدمة، وقت التشغيل، ومعلومات HTML تولد أداة فحص الصحة تقرير OmniCore الإصدار عبر جميع مكونات

. `services/all.yml` **تعمل تلقائيًا** كجزء من كُتِيب

الاستخدام

التشغيل اليدوي

```
ansible-playbook -i hosts/customer/host_files/production.yml  
util_playbooks/health_check.yml
```

المخرجات

يتم إنشاء التقرير في `/tmp/health_check_YYYY-MM-DD HH:MM:SS.html`

.افتحه في أي متصفح ويب لعرضه

محتويات التقرير

HTML: يعرض تقرير

معلومات المضيف

- IP اسم المضيف وعنوان
- أو غير متاح إذا لم `host_vm_network` (من متغير) الشبكة/الشبكة الف `host_vm_network` (يتم تعيينه)
- وحدة المعالجة المركزية (عدد وحدات المعالجة الافتراضية)
- الذاكرة العشوائية (إجمالي وذاكرة مجانية)
- القرص (إجمالي وحر المساحة في قسم الجذر مع النسبة المئوية)
- نظام التشغيل (التوزيع والإصدار)

حالة الخدمة

- حالة الخدمة (نشطة/غير نشطة مع مؤشرات لونية)
- وقت التشغيل
- معلومات الإصدار/الإصدار

HSS Diameter أقران

- حالة اتصال قاعدة البيانات (متصل/غير متصل)
- مضيف الأصل، الحالة، IP، Diameter اتصالات أقران
- (المنفذ 9568) HSS تم جلبها من نقطة نهاية مقاييس

أدوات شائعة أخرى

تكوين النظام الأساسي

(`services/common.yml`) الدور الشائع

- يطبق تكوين النظام الأساسي على جميع المضيفين
- NTP، المنطقة الزمنية، SSH، يقوم بإعداد المستودعات، مفاتيح
- يقوم بتكوين الشبكات وتقوية النظام
- قم بتشغيل هذا قبل `services/common.yml` شر الخدمات

```
ansible-playbook -i hosts/customer/host_files/production.yml
services/common.yml
```

إعداد المستخدمين (services/setup_users.yml)

- ينشئ ويقوم بتكوين حسابات المستخدمين عبر جميع المضيفين
- sudo وامتيازات SSH يدير مفاتيح
- يضمن إعداد مستخدم متسق

```
ansible-playbook -i hosts/customer/host_files/production.yml
services/setup_users.yml
```

إعادة التشغيل (services/reboot.yml)

- يعيد تشغيل جميع المضيفين المستهدفين بشكل لطيف
- ينتظر حتى تعود الأنظمة على الإنترنت (مهلة 5 دقائق)
- مفيد بعد تحديثات النواة أو تغييرات التكوين

```
ansible-playbook -i hosts/customer/host_files/production.yml
services/reboot.yml
```

أدوات تشغيلية

IP مولد خطة (util_playbooks/ip_plan_generator.yml)

- IP لتعيينات عناوين HTML يولد تقرير
- يظهر الطوبولوجيا الكاملة للشبكة من ملف المضيفين
- مفيد للتوثيق واستكشاف الأخطاء وإصلاحها

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/ip_plan_generator.yml
```

HSS نسخة احتياطية لـ (util_playbooks/hss_backup.yml)

- HSS يقوم بعمل نسخة احتياطية من جداول قاعدة بيانات

- المحلية Ansible إلى آلة MySQL ينسخ تفرغ
- مطالبات تفاعلية لمسار النسخة الاحتياطية

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/hss_backup.yml
```

الحصول على الالتقاط المحلي (util_playbooks/getLocalCapture.yml)

- يجلب أحدث ملفي التقاط حزم من جميع المضيفين
- من pcap يسترجع ملفات /etc/localcapture/
- مفيد لاستكشاف مشكلات الاتصال

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/getLocalCapture.yml
```

تحديث MTU (util_playbooks/updateMtu.yml)

- لواجهة الشبكة MTU يحدث إعدادات
- netplan يطبق التغييرات عبر
- jumbo مفيد لتكوين إطارات

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/updateMtu.yml
```

الوثائق ذات الصلة

- **الملف التعريفي الرئيسي** - نظرة عامة وكيفية البدء
- تشغيل الكتيبات - **Ansible** مقدمة في نشر
- تكوين ملف المضيفين - تكوين جردك
- معمارية النشر - نظرة عامة كاملة على النظام
- إدارة الحزم - **APT** نظام ذاكرة التخزين المؤقت

تكوين ملف المضيفين

نظرة عامة

ملف المضيفين (المعروف أيضًا بملف الجرد) هو الوثيقة المركزية للتكوين التي تحدد نشر شبكة الهاتف المحمول بالكامل. يحدد:

- وظائف الشبكة التي يجب نشرها
- (شرائح الشبكة، IP عناوين) أماكن تشغيلها
- كيفية تكوينها (معلومات محددة للخدمة)
- (بيانات الاعتماد، الميزات، PLMN) إعدادات خاصة بالعميل

موقع الملف

تُنظم ملفات المضيفين حسب العميل والبيئة:

```
services/hosts/  
└─ Customer_Name/  
    └─ host_files/  
        ├── production.yml  
        ├── staging.yml  
        └─ lab.yml
```

مثال على هيكل ملف المضيفين

إليك مثال مبسط يوضح الأقسام الرئيسية:

مكونات EPC

```
mme:
  hosts:
    customer-mme01:
      ansible_host: 10.10.1.15
      gateway: 10.10.1.1
      host_vm_network: "vmbr1"
      mme_code: 1
      network_name_short: Customer
      tac_list: [600, 601, 602]
```

```
sgw:
  hosts:
    customer-sgw01:
      ansible_host: 10.10.1.25
      gateway: 10.10.1.1
      cdrs_enabled: true
```

```
pgwc:
  hosts:
    customer-pgw01:
      ansible_host: 10.10.1.21
      gateway: 10.10.1.1
      ip_pools:
        - '100.64.16.0/24'
```

مكونات IMS

```
pcscf:
  hosts:
    customer-pcscf01:
      ansible_host: 10.10.4.165
```

خدمات الدعم

```
license_server:
  hosts:
    customer-licenseserver:
      ansible_host: 10.10.2.150
```

المتغيرات العالمية

```
all:
  vars:
    ansible_connection: ssh
    ansible_password: password
```

```
customer_name_short: customer
plmn_id:
  mcc: '001'
  mnc: '01'
```

معلومات المضيفين الشائعة

تكوين الشبكة

عادةً ما يتضمن كل مضيف

```
pcscf:
  hosts:
    customer-pcscf01:
      ansible_host: 10.10.1.15      # SSH للوصول عبر IP عنوان
      gateway: 10.10.1.1           # البوابة الافتراضية
      host_vm_network: "vmbr1"     # للاستخدام على NIC اسم
Hypervisor
```

واستراتيجيات تقسيم الشبكة، راجع **معيّار IP ملاحظة**: للحصول على إرشادات حول تخطيط عنوان OmniCore الذي يحدد بنية الشبكة الموصى بها ذات الأربعة شرائح لنشر **IP تخطيط**.

الجسر الذي يجب استخدامه. راجع `host_vm_network` تحدد معلمة **Proxmox مستخدمو** للتوفير التلقائي **Proxmox VM/LXC نشر**.

VM تخصيص موارد

بالنسبة للخدمات التي تحتاج إلى موارد محددة:

```
num_cpus: 4      # نوى المعالج
memory_mb: 8192  # الذاكرة العشوائية بالميغابايت
proxmoxLxcDiskSizeGb: 50 # حجم القرص بالجيجابايت
```

معلومات محددة للخدمة

لكل وظيفة شبكة معلوماتها الخاصة. أمثلة:

MME:

```
mme_code: 1 # MME معرف (1-255)
mme_gid: 1 # MME معرف مجموعة
network_name_short: Customer # اسم الشبكة (المعروض على الهواتف)
network_name_long: Customer Network
tac_list: [600, 601, 602] # رموز منطقة التتبع
```

PGW:

```
ip_pools: # للمشاركين IP تجمعات
  - '100.64.16.0/24'
  - '100.64.17.0/24'
combined_CP_UP: false # فصل التحكم / خطة المستخدم
```

للحصول على شرح مفصل لما يتحكم فيه كل متغير، راجع: مرجع التكوين

خادم التطبيق:

```
online_charging_enabled: true # OCS تمكين تكامل
tas_branch: "main" # فرع البرمجيات للنشر
gateways_folder: "gateways_prod" # SIP تكوين بوابة
```

قسم المتغيرات العالمية

على إعدادات تنطبق على النشر بالكامل `all:vars` يحتوي قسم

```

all:
  vars:
    # المصادقة
    ansible_connection: ssh
    ansible_password: password
    ansible_become_password: password

    # هوية العميل
    customer_name_short: customer
    customer_legal_name: "Customer Inc."
    site_name: "Chicago DC1"
    region: US

    # (شبكة الهاتف المحمول) PLMN معرف
    plmn_id:
      mcc: '001'          # رمز الدولة المحمول
      mnc: '01'           # رمز شبكة الهاتف المحمول
      mnc_longform: '001' # مع صفر مضاف MNC

    # أسماء الشبكات
    network_name_short: Customer
    network_name_long: Customer Network

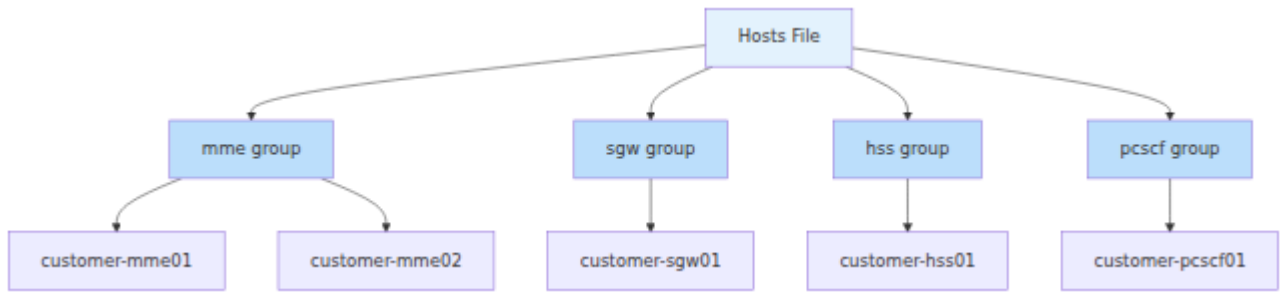
    # مستودع APT
    # مع 000 لمضيفين apt_cache_servers ملاحظة: إذا تم تعريف مجموعة
    # apt_repo.apt_server و true تكون افتراضية على use_apt_cache فإن
    # لأحد خوادم التخزين المؤقت تلقائيًا IP تكون افتراضية على عنوان
    apt_repo:
      apt_server: "10.254.10.223"
      apt_repo_username: "customer"
      apt_repo_password: "secure-password"
    use_apt_cache: false

    # المنطقة الزمنية
    TZ: America/Chicago

```

فهم مجموعات المضيفين

المضيفين في مجموعات تتوافق مع الأدوار Ansible تنظم

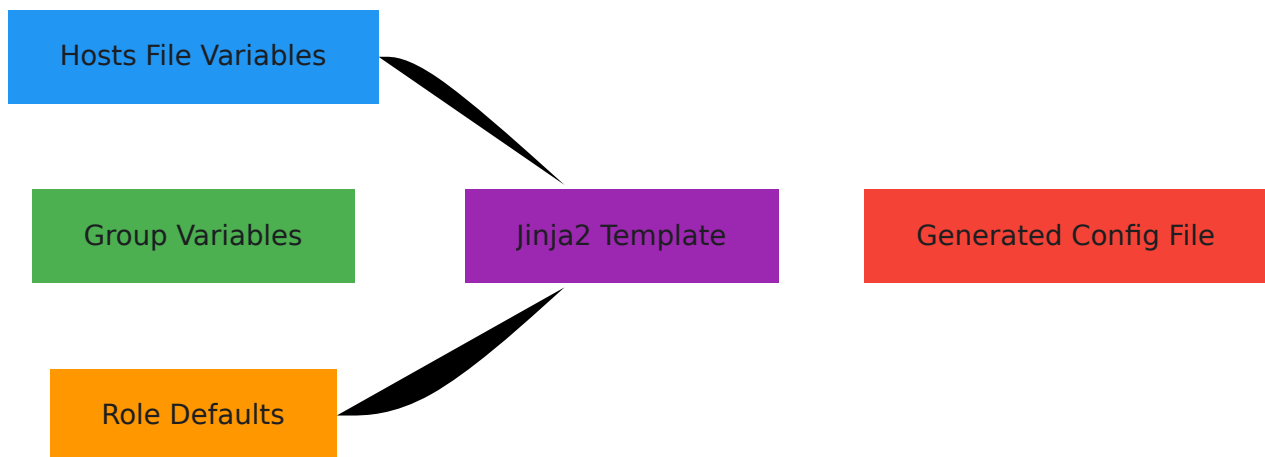


فإنه ينطبق على جميع المضيفين في قسم `mme`، عند تشغيل كتاب اللعب المستهدف `mme:hosts:`.

Jinja2 التكوين باستخدام قوالب

لإنشاء ملفات التكوين من المتغيرات المحددة في ملف `Jinja2` قوالب Ansible تستخدم `group_vars` والمضيفين و.

Jinja2 كيفية عمل



مثال على استخدام القالب

يحدد ملف المضيفين:

```
plmn_id:
  mcc: '001'
  mnc: '01'
customer_name_short: acme
```

قالب Jinja2 (في الدور):

```
# mme_config.yml.j2
network:
  plmn:
    mcc: {{ plmn_id.mcc }}
    mnc: {{ plmn_id.mnc }}
  operator: {{ customer_name_short }}
  realm: epc.mnc{{ plmn_id.mnc_longform }}.mcc{{ plmn_id.mcc
}}.3gppnetwork.org
```

ملف التكوين الناتج:

```
network:
  plmn:
    mcc: 001
    mnc: 01
  operator: acme
  realm: epc.mnc001.mcc001.3gppnetwork.org
```

الشائعة Jinja2 أنماط

الوصول إلى المتغيرات المتداخلة:

```
{{ plmn_id.mcc }}
{{ apt_repo.apt_server }}
```

المنطق الشرطي:

```
{% if online_charging_enabled %}
  charging:
    enabled: true
    ocs_ip: {{ ocs_ip }}
{% endif %}
```

الحلقات:

```
tracking_areas:
{% for tac in tac_list %}
  - {{ tac }}
{% endfor %}
```

التنسيق:

```
# إضافة صفر إلى 3 أرقام
mnc{{ '%03d' | format(plmn_id.mnc|int) }}
```

group_vars تجاوز المتغيرات باستخدام

بينما يحدد ملف المضيفين البنية التحتية وإعدادات المضيف المحددة، يمكن أن تتجاوز `group_vars` القيم الافتراضية لمجموعات المضيفين.

راجع: تكوين متغيرات المجموعة

مثال كامل على ملف المضيفين

إليك مثال أكثر اكتمالاً (مع بيانات حساسة مخفية)

EPC Core

mme:

hosts:

customer-mme01:

ansible_host: 10.10.1.15

gateway: 10.10.1.1

host_vm_network: "vmbr1"

mme_code: 1

mme_gid: 1

network_name_short: Customer

network_name_long: Customer Network

tac_list: [600, 601, 602, 603]

omnimme:

sgw_selection_method: "random_peer"

pgw_selection_method: "random_peer"

sgw:

hosts:

customer-sgw01:

ansible_host: 10.10.1.25

gateway: 10.10.1.1

host_vm_network: "vmbr1"

cdns_enabled: true

pgwc:

hosts:

customer-pgw01:

ansible_host: 10.10.1.21

gateway: 10.10.1.1

host_vm_network: "vmbr1"

ip_pools:

- '100.64.16.0/24'

combined_CP_UP: false

hss:

hosts:

customer-hss01:

ansible_host: 10.10.2.140

gateway: 10.10.2.1

host_vm_network: "vmbr2"

IMS Core

pcscf:

```
hosts:
  customer-pcscf01:
    ansible_host: 10.10.4.165
    gateway: 10.10.4.1
    host_vm_network: "vmbr4"

icscf:
  hosts:
    customer-icscf01:
      ansible_host: 10.10.3.55
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"

scscf:
  hosts:
    customer-scscf01:
      ansible_host: 10.10.3.45
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"

applicationserver:
  hosts:
    customer-as01:
      ansible_host: 10.10.3.60
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      online_charging_enabled: false
      gateways_folder: "gateways_prod"

# خدمات الدعم
license_server:
  hosts:
    customer-licenseserver:
      ansible_host: 10.10.2.150
      gateway: 10.10.2.1
      host_vm_network: "vmbr2"

monitoring:
  hosts:
    customer-oam01:
      ansible_host: 10.10.2.135
      gateway: 10.10.2.1
      host_vm_network: "vmbr2"
      num_cpus: 4
```

```

memory_mb: 8192

dns:
  hosts:
    customer-dns01:
      ansible_host: 10.10.2.177
      gateway: 10.10.2.1
      host_vm_network: "vmbr2"

# المتغيرات العالمية
all:
  vars:
    ansible_connection: ssh
    ansible_password: password
    ansible_become_password: password

    customer_name_short: customer
    customer_legal_name: "Customer Network Inc."
    site_name: "Primary DC"
    region: US
    TZ: America/Chicago

# تكوين PLMN
plmn_id:
  mcc: '001'
  mnc: '01'
  mnc_longform: '001'
  diameter_realm: epc.mnc{{ plmn_id.mnc_longform }}.mcc{{
plmn_id.mcc }}.3gppnetwork.org

# أسماء الشبكات
network_name_short: Customer
network_name_long: Customer Network
tac_list: [600, 601]

# تكوين APT
apt_repo:
  apt_server: "10.254.10.223"
  apt_repo_username: "customer"
  apt_repo_password: "secure-password"
  use_apt_cache: false

# تكوين الشحن
charging:

```

```

data:
  online_charging:
    enabled: false
  voice:
    online_charging:
      enabled: true
    domain: "mnc{{ plmn_id.mnc_longform }}.mcc{{ plmn_id.mcc
}}.3gppnetwork.org"

# قواعد جدار الحماية
firewall:
  allowed_ssh_subnets:
    - '10.0.0.0/8'
    - '192.168.0.0/16'
  allowed_ue_voice_subnets:
    - '10.0.0.0/8'
  allowed_signaling_subnets:
    - '10.0.0.0/8'

# تكوين Hypervisor (مثال Proxmox)
proxmoxServers:
  customer-prxm01:
    proxmoxServerAddress: 10.10.0.100
    proxmoxServerPort: 8006
    proxmoxApiTokenName: Customer
    proxmoxApiTokenSecret: "token-secret"
    proxmoxTemplateName: ubuntu-24.04-cloud-init-template
    proxmoxNodeName: pve01

```

الكاملة Proxmox للحصول على تفاصيل إعدادات وتكوين Proxmox VM/LXC راجع [نشر](#)

مراجع الوثائق الخاصة بالمنتج

للحصول على تكوين مفصل لكل مكون، راجع الوثائق الرسمية للمنتج

OmniCore مكونات:

- **OmniCore وثائق:**
<https://docs.omnitouch.com.au/docs/repos/OmniCore>
- **OmniHSS** - خادم المشتركين المنزلي

- **OmniSGW** - بوابة الخدمة (خطة التحكم)
- **OmniPGW** - بوابة الحزمة (خطة التحكم)
- **OmniUPF** - وظيفة خطة المستخدم
- **OmniDRA** - Diameter وكيل توجيه
- **OmniTWAG** - الموثوقة WLAN بوابة الوصول

OmniCall مكونات:

- **OmniCall وثائق:** <https://docs.omnitouch.com.au/docs/repos/OmniCall>
- **OmniTAS** - IMS (VoLTE/VoNR) خادم تطبيق
- **OmniCall CSCF** - وظائف التحكم في جلسات المكالمات
- **OmniMessage** - مرسلات الرسائل القصيرة
- **OmniMessage SMPP** - SMPP دعم بروتوكول
- **OmniSS7** - SS7 كومة الإشارات
- **VisualVoicemail** - البريد الصوتي

OmniCharge/OmniCRM:

- **OmniCharge وثائق:**
<https://docs.omnitouch.com.au/docs/repos/OmniCharge>

الوثائق ذات الصلة

- **عملية النشر العامة - Ansible مقدمة في نشر**
- **مرجع التكوين - دليل كامل لجميع متغيرات التكوين**
- **تكوين متغيرات المجموعة - تجاوز التكوين الافتراضي**
- **IP إرشادات بنية الشبكة وتخصيص - IP معيار تخطيط**
- **ثانوي وتكوين شبكة متقدم IP - Netplan تكوين**
- **توزيع الحزم - APT نظام تخزين**
- **خادم الترخيص - إدارة التراخيص**
- **نظرة عامة على بنية النشر - عرض كامل للنظام**

الخطوات التالية

1. أنشئ ملف المضيفين الخاص بك بناءً على هذا القالب
2. وهويتك الشبكية PLMN حدد
3. APT قم بتكوين الوصول إلى مستودع
4. إعداد خادم الترخيص
5. حسب الحاجة `group_vars` تخصيص باستخدام
6. Ansible نشر باستخدام كتب لعب

لـ IP معيار تخطيط OmniCore

نظرة عامة

تتطلب البنية المعمارية **أربعة** OmniCore القياسي لنشر IP توضح هذه الوثيقة نهج تخطيط **شبكات فرعية داخلية** لتقسيم وظائف الشبكة بشكل صحيح من أجل الأمان والأداء ووضوح العمليات.

IP متطلبات تخصيص

التخصيص القياسي: أربع شبكات فرعية /24

:أربع شبكات فرعية متميزة للتواصل الداخلي OmniCore يتطلب كل نشر لـ

1. **شبكة النواة الحزمية** - أول /24
2. **شبكة الإشارة** - ثاني /24
3. **الداخلية** - ثالث /24 **IMS شبكة**
4. **العامة** - رابع /24 **UE شبكة**

مهم: هذه توصيات، ليست متطلبات

تخصيص الشبكات الفرعية الموصوف في هذه الوثيقة هو **أفضل ممارسة موصى بها** لتنظيم :ومع ذلك، فإن البنية المعمارية **مرنة تمامًا**. OmniCore نشرات

- **جميع المضيفين في شبكة فرعية واحدة**: يمكنك وضع جميع المكونات في شبكة فرعية واحدة إذا كان ذلك يناسب احتياجات نشراتك
- **كل نوع مضيف في شبكته الفرعية الخاصة**: يمكنك إنشاء شبكات فرعية منفصلة (إلخ، HSS واحدة لـ MMEs واحدة لـ) لكل نوع من المكونات
- **تجميعات مخصصة**: يمكنك تنظيم المضيفين في أي هيكل شبكة فرعية يتناسب مع متطلباتك الخاصة

- **الداخلية والعامة:** يمكن لبعض المضيفين استخدام عناوين داخلية **IPs مزيج من** عامة، كل ذلك ضمن نفس النشر IPS بينما يستخدم الآخرون (RFC 1918)

يوفر نهج الشبكات الفرعية الأربعة الموصى به **عزل أمني مثالي، إدارة حركة المرور، IP ووضوح العمليات**، ولهذا السبب نقترح ذلك لنشر الإنتاج. ومع ذلك، يجب عليك تعديل خطة لتناسب طوبولوجيا الشبكة الخاصة بك، ومساحة العناوين المتاحة، ومتطلبات التشغيل.

تحليل مقاطع الشبكة

1. شبكة النواة الحزمية (أو 24/ 24)

الغرض: عناصر خطة المستخدم وخطة التحكم الأساسية

المكونات:

- OmniMME (كيان إدارة الحركة)
- OmniSGW (البوابة الخدمية)
- OmniPGW-C (PDN خطة التحكم في بوابة)
- OmniUPF/PGW-U (PDN وظيفة خطة المستخدم / بوابة)

مثال: 10.179.1.0/24

```
mme:
  hosts:
    omni-site-mme01:
      ansible_host: 10.179.1.15
      gateway: 10.179.1.1
      host_vm_network: "vmbr1"
```

2. شبكة الإشارة (ثاني 24/ 24)

الغرض: إشارات قطرية، سياسة، شحن، ووظائف إدارة

المكونات:

- OmniHSS (خادم المشتركين المنزليين)

- OmniCharge OCS (نظام الشحن عبر الإنترنت)
- OminiHSS PCRF (وظيفة قواعد السياسة والشحن)
- OmniDRA DRA (وكيل توجيه قطرية)
- خوادم DNS
- خوادم TAP3/CDR
- المراقبة/OAM
- التقاط SIP
- خادم الترخيص
- مراقب RAN
- إذا تم نشره - (مركز البث الخلوي) Omnitouch CBC رابط التحذير
- إذا تم نشرها - APT خوادم ذاكرة التخزين المؤقت

مثال: 10.179.2.0/24

```
hss:
  hosts:
    omni-site-hss01:
      ansible_host: 10.179.2.140
      gateway: 10.179.2.1
      host_vm_network: "vmbr2"
```

3. الداخلية (ثالث / 24) IMS شبكة

(الداخلية SIP إشارات) الأساسية IMS **الغرض:** إشارات وخدمات

المكونات:

- OmniCSCF S-CSCF (وظيفة التحكم في جلسات المكالمات الخدمية)
- OmniCSCF I-CSCF (وظيفة التحكم في جلسات المكالمات الاستقصائية)
- OmniTAS (خادم تطبيقات الهاتف / خادم التطبيقات)
- OmniMessage (SMPP، IMS، وحدة التحكم في الرسائل القصيرة)
- OmniSS7 STP (SS7 نقطة نقل إشارات)
- OmniSS7 HLR (سجل الموقع المنزلي) 2 - JG/3G
- OmniSS7 IP-SM-GW (MAP SMS Sc)

- بوابة OmniSS7 CAMEL

مثال: 10.179.3.0/24

```
scscf:
  hosts:
    omni-site-scscf01:
      ansible_host: 10.179.3.45
      gateway: 10.179.3.1
      host_vm_network: "vmbr3"
```

4. العامة (رابع /24) شبكة UE

DNS و IMS الغرض: خدمات موجهة للمستخدم مثل

المكونات:

- OmniCSCF P-CSCF (الوكيلة) جلسات المكالمات الوكيله
- خوادم XCAP
- خوادم البريد الصوتي المرئي
- العملاء DNS

مثال: 10.179.4.0/24

```
pcscf:
  hosts:
    omni-site-pcscf01:
      ansible_host: 10.179.4.165
      gateway: 10.179.4.1
      host_vm_network: "vmbr4"
```

طرق التنفيذ

:طريقتين رئيسيتين لتنفيذ هذا التقسيم الشبكي OmniCore يدعم

الطريقة 1: واجهات الشبكة الفيزيائية/الافتراضية (موصى بها للإنتاج)

منفصلة أو جسور افتراضية لكل مقطع شبكة. يوفر ذلك أقوى عزل وهو النهج NICs استخدم الموصى به لنشر الإنتاج.

مثال:

```
# النواة الحزمية - vmbr1
mme:
  hosts:
    omni-lab07-mme01:
      ansible_host: 10.179.1.15
      gateway: 10.179.1.1
      host_vm_network: "vmbr1"

# الإشارة - vmbr2
hss:
  hosts:
    omni-lab07-hss01:
      ansible_host: 10.179.2.140
      gateway: 10.179.2.1
      host_vm_network: "vmbr2"

# IMS الداخلية - vmbr3
icscf:
  hosts:
    omni-lab07-icscf01:
      ansible_host: 10.179.3.55
      gateway: 10.179.3.1
      host_vm_network: "vmbr3"

# UE العامة - vmbr4
pcscf:
  hosts:
    omni-lab07-pcscf01:
      ansible_host: 10.179.4.165
      gateway: 10.179.4.1
      host_vm_network: "vmbr4"
```

VLAN الطريقة 2: تقسيم قائم على

لفصل الشبكات. هذا مناسب للنشر الأصغر أو عندما VLAN استخدم واجهة فيزيائية واحدة مع وسم الفيزيائية محدودة NICs تكون.

مثال:

```
# مختلفة VLANs مع vmbr12 جميع المكونات تستخدم
```

```
applicationserver:
```

```
hosts:
```

```
ons-lab08sbc01:
```

```
ansible_host: 10.178.2.213
```

```
gateway: 10.178.2.1
```

```
host_vm_network: "ovsbr1"
```

```
vlanid: "402"
```

```
dra:
```

```
hosts:
```

```
ons-lab08dra01:
```

```
ansible_host: 10.178.2.211
```

```
gateway: 10.178.2.1
```

```
host_vm_network: "ovsbr1"
```

```
vlanid: "402"
```

```
dns:
```

```
hosts:
```

```
ons-lab08dns01:
```

```
ansible_host: 10.178.2.178
```

```
gateway: 10.178.2.1
```

```
host_vm_network: "ovsbr1"
```

```
vlanid: "402"
```

تكوين الشبكة:

- على المفتاح الفيزيائي VLANs قم بتكوين
- وسم حركة المرور بشكل مناسب على مستوى المحاكى
- عند البوابة/الجدار الناري VLANs قم بتوجيه بين

VLAN: مثال على خريطة

VLAN 10: 10.x.1.0/24 (النواة الحزمية)
VLAN 20: 10.x.2.0/24 (الإشارة)
VLAN 30: 10.x.3.0/24 (الداخلية IMS)
VLAN 40: 10.x.4.0/24 (العام UE)

العام IP العمل مع عناوين

نظرة عامة

عام للاتصال IP أن تحتوي بعض المكونات على عناوين OmniCore تتطلب العددين من نشرات الخارجي، مثل:

- لإشارات قطرية التجوال مع شركات النقل الخارجية - **DRA**
- من الشركاء المتجولين GTP **للتجوال** - لحركة مرور **SGW/PGW**
- (UE من IPsec أنفاق) WiFi للمكالمات عبر - **ePDG**
- مع مجموعات الرسائل القصيرة الخارجية SMPP للاتصالات - **بوابة SMSC**
- UE SIP للتسجيل المباشر لـ - (في بعض النشرات) **P-CSCF**

العام IPs كيفية تخصيص

الداخلية في ملفات جرد **IPs** العامة **بالطريقة نفسها تمامًا كما يتم معالجة** IPs تتم معالجة مع البوابة المناسبة وقناع `ansible_host` العام في حقل IP المضيفين لديك. ببساطة حدد عنوان الشبكة.

العام IPs للتجوال مع SGW/PGW :مثال

```

sgw:
  hosts:
    # SGWs الداخلية على الشبكة الخاصة
    opt-site-sgw01:
      ansible_host: 10.4.1.25
      gateway: 10.4.1.1
      host_vm_network: "v400-omni-packet-core"

    # SGWs العامة IPs للتجوال مع
    opt-site-roaming-sgw01:
      ansible_host: 203.0.113.10
      gateway: 203.0.113.9
      netmask: 255.255.255.248          # /29 subnet
      host_vm_network: "498-public-servers"
      in_pool: False
      cdrs_enabled: True

smf: # PGWs
  hosts:
    # PGW عام IP للتجوال مع
    opt-site-roaming-pgw01:
      ansible_host: 203.0.113.20
      gateway: 203.0.113.17
      netmask: 255.255.255.240          # /28 subnet
      host_vm_network: "497-public-services-LTE"
      in_pool: False
      ip_pools:
        - '100.64.24.0/22'

```

عام IP مع DRA :مثال

```

dra:
  hosts:
    opt-site-dra01:
      ansible_host: 198.51.100.50
      gateway: 198.51.100.49
      netmask: 255.255.255.240          # /28 subnet
      host_vm_network: "497-public-services-LTE"

```

عام IP مع ePDG :مثال

```
epdg:
  hosts:
    opt-site-epdg01:
      ansible_host: 198.51.100.51
      gateway: 198.51.100.49
      netmask: 255.255.255.240          # /28 subnet
      host_vm_network: "497-public-services-LTE"
```

الداخلية والعامة IPs مزيج من

الداخلية والعامة ضمن نفس مجموعة المكونات. على IPs من الشائع أن يكون هناك مزيج من سبيل المثال:

- GTP الداخلية لمواقع محلية تستخدم SGWs
- العامة خصيصًا لحركة مرور التجوّل من شركات النقل الخارجية SGWs
- الداخلية والخارجية SGWs كل من PGW-C يمكن أن تدير نفس

المناسبة له IP مع ذلك بسلاسة - فقط قم بتكوين كل مضيف مع عناوين OmniCore تتعامل بنية

خادم الترخيص

نظرة عامة

يتحقق كل مكون من ترخيصه Omnitouch. يدير خادم الترخيص تفعيل الميزات لجميع مكونات عند بدء التشغيل وبشكل دوري أثناء التشغيل.

الإعداد

1. التعريف في ملف المضيفين

```
license_server:
  hosts:
    customer-licenseserver:
      ansible_host: 10.10.2.150
      gateway: 10.10.2.1
      host_vm_network: "vmbr2"

all:
  vars:
    customer_legal_name: "Customer Name"
    license_server_api_urls: ["https://10.10.2.150:8443/api"]
    license_enforced: true
```

2. توفير ملف الترخيص

ضع `license.json` في `hosts/Customer/group_vars/` (Omnitouch المقدم من)

3. النشر

```
ansible-playbook -i hosts/customer/host_files/production.yml
services/license_server.yml
```

. https://license_server يمكنك التحقق من حالة جميع التراخيص عن طريق التصفح إلى

متطلبات الشبكة

تكوين جدار الحماية

إلى (المنفذ 443) HTTPS يجب تكوين جدران الحماية في موقع العميل للسماح بحركة مرور التالية Omnitouch خوادم التحقق من الترخيص الخاصة بـ:

الغرض	IP عنوان	اسم المضيف
خادم التحقق من الترخيص 1	160.22.43.18	time.omnitouch.com.au
خادم التحقق من الترخيص 2	160.22.43.66	time.omnitouch.com.au
خادم التحقق من الترخيص 3	160.22.43.114	time.omnitouch.com.au

القواعد المطلوبة للخروج:

- البروتوكول: HTTPS (TCP/443)
- الوجهة: 160.22.43.114 , 160.22.43.66 , 160.22.43.18
- الاتجاه: خارجي

DNS متطلبات

وظيفي للتواصل مع بنية التحقق من الترخيص الخاصة بـ DNS يتطلب خادم الترخيص وجود حل Omnitouch.

المطلوب DNS تكوين:

- العامة DNS يجب أن يكون لخادم الترخيص وصول إلى خوادم
- لاستخدام واحد من الخيارات التالية DNS قم بتكوين:
 - 1.1.1.1 (Cloudflare - يدعم)
 - 8.8.8.8 (Google Public DNS)
- الداخلية/الشركات لخادم الترخيص DNS لا تستخدم خوادم

يضمن استخدام Omnitouch DNS الآمن (DoH/DoT) ملائمة: تستخدم خوادم ترخيص DNS ويمنع المشاكل الناتجة عن اعتراض DNSSEC العامة التحقق الصحيح من DNS خوادم بواسطة أجهزة الأمان.

الوثائق ذات الصلة

- مرجع التكوين
- تكوين ملف المضيفين

Netplan تكوين

نظرة عامة

هذا netplan المنشورة باستخدام VMs تكوين واجهات الشبكة تلقائيًا على OmniCore يمكن لـ مفيد لـ:

- (eth0) إعداد واجهة الإدارة الرئيسية
- العامة، أو اتصالات التبادل، أو حركة المرور المخصصة IP إضافة واجهات ثانوية لعناوين
- تكوين الطرق الثابتة لوجهات معينة

Netplan تمكين تكوين

الذي يشير إلى netplan_config لمضيف، أضف متغير netplan لتمكين التكوين التلقائي لـ الخاص بك group_vars في مجلد Jinja2 قالب:

```
dra:
  hosts:
    <hostname>:
      ansible_host: 10.0.1.100
      gateway: 10.0.1.1
      netplan_config: netplan.yaml.j2
```

سيتم الحصول على القالب من hosts/<customer>/group_vars/netplan.yaml.j2.

مرجع القالب

مع تعليقات تشرح كل قسم netplan.yaml.j2 إليك القالب الكامل:

```

network:
  version: 2
  ethernet:
    # من الجرد gateway و ansible_host الواجهة الرئيسية - تستخدم
    eth0:
      addresses:
        - "{{ ansible_host }}/{{ mask_cidr | default(24) }}"
      nameservers:
        addresses:
{% if 'dns' in group_names %}
          # خارجي لتجنب DNS استخدم، DNS إذا كان هذا المضيف هو خادم
          الاعتماد الدائري
          - 8.8.8.8
{% else %}
          # في الجرد 'dns' من مجموعة DNS خلاف ذلك، استخدم خوادم
{% for dns_host in groups['dns'] | default([]) %}
          - {{ hostvars[dns_host]['ansible_host'] }}
{% endfor %}
{% endif %}
      search:
        - slice
      routes:
        - to: "default"
          via: "{{ gateway }}"

{% if secondary_ips is defined %}
  # من الجرد dict secondary_ips الواجهات الثانوية - حلقة عبر
  # تسمية الواجهة: ens19, ens20, ens21... (18 + loop.index)
  {% for nic_name, nic_config in secondary_ips.items() %}
    ens{{ 18 + loop.index }}:
      addresses:
        - "{{ nic_config.ip_address }}/{{ mask_cidr | default(24) }}"
      {% if nic_config.routes is defined %}
        # طرق ثابتة لهذه الواجهة - تستخدم كل طريق بوابة هذه الواجهة
        routes:
          {% for route in nic_config.routes %}
            - to: "{{ route }}"
              via: "{{ nic_config.gateway }}"
          {% endfor %}
      {% endif %}
  {% endfor %}

```

```
{% endfor %}  
{% endif %}
```

نقاط رئيسية:

- تأتي من إدخال الجرد الخاص بالمضيف `gateway` و `ansible_host`
- `dns` ديناميكيًا من المضيفين في مجموعة DNS يتم سحب خوادم
- في NIC إلخ. لتتناسب مع تسمية ، `ens20` ، `ens19` يتم تسمية الواجهات الثانوية Proxmox
- ثانوي على بوابة وطرق ثابتة خاصة بها IP يمكن أن تحتوي كل

تكوين الواجهة الرئيسية

تلقائيًا باستخدام (eth0) يتم تكوين الواجهة الرئيسية:

- `ansible_host` - عنوان IP
- `gateway` - البوابة الافتراضية
- `mask_cidr` - قناع الشبكة (يكون افتراضيًا 24)

تلقائيًا إلى DNS يتم تعيين خوادم

- (`ansible_host` الخاصة بهم IP تستخدم عناوين) `dns` المضيفين في مجموعة
- DNS تتراجع إلى `8.8.8.8` إذا كان المضيف هو نفسه خادم

الواجهات الثانوية

استخدم ، (العامة، التبادل، إلخ IP عناوين) للمضيفين الذين يحتاجون إلى واجهات شبكة إضافية `secondary_ips`. تكوين

المخطط

```
secondary_ips:
  <logical_name>:
    ip_address: <ip_address>
    gateway: <gateway_ip>
    host_vm_network: <proxmox_bridge>
    vlanid: <vlan_id>
    routes: اختياري - طرق ثابتة عبر هذه الواجهة #
      - '<destination_cidr>'
      - '<destination_cidr>'
```

تسمية الواجهة

Ubuntu: يتم تسمية الواجهات الثانوية تلقائيًا باستخدام نظام التسمية القابل للتنبؤ في

- أول واجهة ثانوية: ens19
- ثاني واجهة ثانوية: ens20
- ثالث واجهة ثانوية: ens21
- وهكذا...

VM. إضافية إلى NICs عند إضافة Proxmox هذا يتطابق مع أسماء الواجهات المخصصة من قبل

مثال على التكوين

```
dra:
  hosts:
    <hostname>:
      ansible_host: 10.0.1.100
      gateway: 10.0.1.1
      host_vm_network: "ovsbr1"
      vlanid: "100"
      netplan_config: netplan.yaml.j2
      secondary_ips:
        public_ip:
          ip_address: 192.0.2.50
          gateway: 192.0.2.1
          host_vm_network: "vibr0"
          vlanid: "200"
          routes:
            - '198.51.100.0/24'
            - '203.0.113.0/24'
        peering_ip:
          ip_address: 172.16.50.10
          gateway: 172.16.50.1
          host_vm_network: "ovsbr2"
          vlanid: "300"
          routes:
            - '172.17.0.0/16'
```

الناتج Netplan ناتج

:يولد التكوين أعلاه

```
network:
  version: 2
  ethernet:
    eth0:
      addresses:
        - "10.0.1.100/24"
      nameservers:
        addresses:
          - 10.0.1.53
        search:
          - slice
      routes:
        - to: "default"
          via: "10.0.1.1"
    ens19:
      addresses:
        - "192.0.2.50/24"
      routes:
        - to: "198.51.100.0/24"
          via: "192.0.2.1"
        - to: "203.0.113.0/24"
          via: "192.0.2.1"
    ens20:
      addresses:
        - "172.16.50.10/24"
      routes:
        - to: "172.17.0.0/16"
          via: "172.16.50.1"
```

Proxmox تكامل

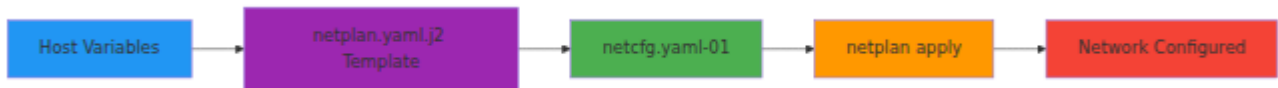
VM: الثانوية تلقائيًا على NICs يتم إنشاء ، proxmox.yml playbook عند استخدام

1. VMs الثانوية أثناء التكوين الأولي NICs جديدة: يتم إضافة VMs
2. VMs لتطبيق التغييرات VM الثانوية ويتم إعادة تشغيل NICs موجودة: يتم إضافة VMs

Proxmox: يستخدم تكوين

- به NIC الجسر الذي يتم توصيل - host_vm_network
- للواجهة VLAN علامة - vlanid

كيف يعمل



1. Jinja2 يتم تمرير المتغيرات من ملف المضيف إلى قالب
2. يتم عرض القالب إلى `/etc/netplan/01-netcfg.yaml`
3. موجودة لتجنب التعارضات netplan تتم إزالة أي تكوينات
4. بتنشيط التكوين `netplan apply` يقوم
5. `ip addr show` باستخدام IP يتم التحقق من عناوين

حالات الاستخدام الشائعة

عام IP مع Diameter Edge Agent (DEA)

```
<hostname>:
  ansible_host: 10.0.1.100          # IP إدارة داخلي
  gateway: 10.0.1.1
  netplan_config: netplan.yaml.j2
  secondary_ips:
    diameter_roaming:
      ip_address: 192.0.2.50        # IP لشركاء التجوال
      gateway: 192.0.2.1
      host_vm_network: "vibr0"
      vlanid: "200"
      routes:
        - '198.51.100.0/24'        # شبكة شريك التجوال
```

PGW مع واجهة S5/S8

```
<hostname>:
  ansible_host: 10.0.2.20          # IP داخلي
  gateway: 10.0.2.1
  netplan_config: netplan.yaml.j2
  secondary_ips:
    s5s8_interface:
      ip_address: 203.0.113.17     # IP عام S5/S8
      gateway: 203.0.113.1
      host_vm_network: "vmbr0"
      vlanid: "50"
```

خادم متعدد المنافذ مع شبكات إدارة وبيانات منفصلة

```
<hostname>:
  ansible_host: 10.0.1.100         # شبكة الإدارة
  gateway: 10.0.1.1
  netplan_config: netplan.yaml.j2
  secondary_ips:
    data_network:
      ip_address: 10.0.2.100       # شبكة البيانات
      gateway: 10.0.2.1
      host_vm_network: "ovsbr2"
      vlanid: "200"
    backup_network:
      ip_address: 10.0.3.100       # شبكة النسخ الاحتياطي
      gateway: 10.0.3.1
      host_vm_network: "ovsbr3"
      vlanid: "300"
```

الثانوية في القوالب IPs الإشارة إلى

الأخرى وملفات التكوين Jinja2 الثانوية في قوالب IP يمكنك الإشارة إلى عناوين

على نفس المضيف

ثانوية، يمكنك الإشارة مباشرة أو استخدام IPS عند تكوين خدمة على نفس المضيف الذي لديه `inventory_hostname`:

```
# الإشارة مباشرة (الأبسط)
{{ secondary_ips.diameter_public_ip.ip_address }}

# نفس النتيجة inventory_hostname أو بشكل صريح عبر
{{ hostvars[inventory_hostname]['secondary_ips']
  ['diameter_public_ip']['ip_address'] }}

# الوصول إلى خصائص أخرى
{{ secondary_ips.diameter_public_ip.gateway }}
{{ secondary_ips.diameter_public_ip.vlanid }}
```

من مضيف آخر

ثانوي لمضيف مختلف (على سبيل المثال، تكوين اتصال نظير)، IP عند الحاجة إلى الإشارة إلى مع اسم المضيف المستهدف `hostvars` استخدم:

```
# dra الإشارة إلى المضيف الأول في مجموعة
{{ hostvars[groups['dra'][0]]['secondary_ips']
  ['diameter_public_ip']['ip_address'] }}

# العامة الخاصة بهم IP والحصول على عناوين DRA حلقة عبر جميع مضيفي
{% for host in groups['dra'] %}
{% if hostvars[host]['secondary_ips'] is defined %}
  - {{ hostvars[host]['secondary_ips']['diameter_public_ip']
    ['ip_address'] }}
{% endif %}
{% endfor %}
```

DRA مثال: تكوين نظير

:العام الخاص به IP قم بتكوين نظير قطر لربطه بعنوان

```
# للمضيف الحالي inventory_hostname استخدم - dra_config.yaml.j2 في
peers:
  - name: external_peer
    # العام الخاص بهذا المضيف IP ربط بعنوان
    local_ip: {{ hostvars[inventory_hostname]['secondary_ips']
['diameter_public_ip']['ip_address'] }}
    remote_ip: 198.51.100.50
    port: 3868
```

الثانوية موجودة IPs التحقق مما إذا كانت

تحقق دائمًا مما إذا كان المتغير موجودًا قبل استخدامه

```
{% if secondary_ips is defined and
secondary_ips.diameter_public_ip is defined%}
public_ip: {{ secondary_ips.diameter_public_ip.ip_address }}
{% else %}
public_ip: {{ ansible_host }}
{% endif %}
```

استكشاف الأخطاء وإصلاحها

تحقق من أسماء الواجهات

وتحقق من أسماء الواجهات VM قم بتسجيل الدخول إلى

```
ip link show
```

مع واجهتين ثانويتين VM الناتج المتوقع لـ

```
1: lo: <LOOPBACK,UP,LOWER_UP> ...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> ...
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> ...
4: ens20: <BROADCAST,MULTICAST,UP,LOWER_UP> ...
```

Netplan تحقق من تكوين

```
cat /etc/netplan/01-netcfg.yaml
```

يدويًا Netplan تطبيق

```
netplan apply
```

Netplan تصحيح

```
netplan --debug apply
```

تحقق من الطرق

```
ip route show
```

الوثائق ذات الصلة

- تكوين ملف المضيفين - إعداد جرد المضيف
- VM تكوين - Proxmox VM/LXC نشر
- مرجع التكوين - جميع متغيرات التكوين

في VM/LXC نشر Proxmox

وهذه الدليل، Proxmox، على OmniCore تقوم الغالبية العظمى من عملائنا بتشغيل مجموعة Proxmox. لإعداد بيئتهم باستخدام proxmox يشرح بالتفصيل كيفية استخدام عمليات

.لنشر (Vultr / AWS / GCP حالياً) والسحابة HyperV وVMware نواصل دعم

انظر أيضًا:

- للنشر VMS **تكوين ملف المضيف** - تعريف
- IP إرشادات تخصيص عنوان - **IP معيار تخطيط**
- NIC الثانوية وإعدادات متعددة IP عناوين - **Netplan تكوين**
- **معمارية النشر** - سير العمل الكامل للنشر

VM مقابل LXC

LXC حاويات:

- خفيفة الوزن، تشارك نواة المضيف
- بدء تشغيل سريع، تكلفة منخفضة
- عزل محدود
- لا يمكن تشغيل نوى مخصصة أو وحدات نواة
- **غير مدمجة** **أسبة للنشر في الإنتاج**
- TUN يتطلب وحدات نواة / أجهزة) **UPF لا يمكن تشغيل**

(KVM) الآلات الافتراضية:

- افتراضية كاملة مع نواة مخصصة
- عزل كامل
- يمكن تشغيل وحدات النواة والشبكات المخصصة
- تكلفة موارد أعلى
- **موصى بها للإنتاج**

- **UPF مطلوبة لنشر**

:حالات الاستخدام

- جميع وظائف الشبكة، UPF، **الآلات الافتراضية**: مواقع الإنتاج
- **LXC** (مراقبة، apt-cache) بيئات مختبر / اختبار، خدمات خفيفة الوزن

Proxmox إعداد

1. API إنشاء رمز

```
# API مركز البيانات → الأذونات → رموز Proxmox: في واجهة #  
# إنشاء رمز: root@pam!<TokenName>  
# انسخ سر الرمز (يظهر مرة واحدة)
```

2. (فقط VMs ل) VM Cloud-Init إنشاء قالب

ويقوم Ubuntu يقوم بتنزيل صورة سحابية من Proxmox. قم بتشغيل هذا السكريبت على مضيف cloud-init بإنشاء قالب مع بيانات اعتماد مستخدم.

```
#!/bin/bash
set -e

TEMPLATE_ID=9000
IMAGE_URL="https://cloud-images.ubuntu.com/noble/current/noble-server-cloudimg-amd64.img"
IMAGE="noble-server-cloudimg-amd64.img"

echo "=== تنزيل صورة سحابية من Ubuntu ==="
cd /var/lib/vz/template/iso
wget -N "$IMAGE_URL"

echo "=== تنظيف القالب القديم ==="
qm destroy $TEMPLATE_ID --purge 2>/dev/null || true

echo "=== تمكين تخزين المقطعات ==="
pvesm set local --content images,vztmpl,iso,backup,snippets

echo "=== إنشاء بيانات مستخدم cloud-init ==="
mkdir -p /var/lib/vz/snippets
cat > /var/lib/vz/snippets/user-data.yml << 'USERDATA'
#cloud-config
ssh_pwauth: true
users:
  - name: omnitouch
    plain_text_passwd: password
    lock_passwd: false
    shell: /bin/bash
    sudo: ALL=(ALL) NOPASSWD:ALL
    groups: sudo
USERDATA

echo "=== إنشاء قالب VM ==="
qm create $TEMPLATE_ID --name ubuntu-2404-template --memory 2048 -
-cores 2 --net0 virtio,bridge=vmbr0
qm importdisk $TEMPLATE_ID $IMAGE local-lvm
qm set $TEMPLATE_ID --scsihw virtio-scsi-pci --scsi0 local-
lvm:vm-${TEMPLATE_ID}-disk-0
qm set $TEMPLATE_ID --ide2 local-lvm:cloudinit
qm set $TEMPLATE_ID --boot c --bootdisk scsi0
qm set $TEMPLATE_ID --vga std
qm set $TEMPLATE_ID --agent enabled=1
qm set $TEMPLATE_ID --cicustom user=local:snippets/user-data.yml
```

```
qm template $TEMPLATE_ID
```

```
echo "=== بنجاح $TEMPLATE_ID تم إنشاء القالب ==="
```

ملاحظات:

- للوصول إلى وحدة) omnitouch / password :يوفر القالب تسجيل دخول احتياطي (cloud-init التحكم إذا فشل
- في ملف local_users يتم تجاوز بيانات الاعتماد من Ansible عند الاستنساخ عبر المضيفين الخاص بك:
 - local_users اسم المستخدم: مفتاح أول مستخدم من
 - يكون الافتراضي) لأول مستخدم password كلمة المرور: حقل (إذا لم يتم تعيينه 'password')
 - لأول مستخدم public_key حقل SSH: مفتاح
- Proxmox يضمن عمل وحدة التحكم على الويب في -vga std -
- تقوم بالتنزيل فقط إذا كانت أحدث من النسخة المحلية wget على -N علامة

ISO بديل: قالب يدوي من

:إذا لم تكن صور السحاب متاحة أو كنت بحاجة إلى تثبيت مخصص

عبر واجهة الويب VM الخطوة 1: إنشاء

- ubuntu-2404-template: الاسم، VM 9000 جديدة → معرف VM إنشاء
- موجود ISO أو استخدام Ubuntu خادم ISO نظام التشغيل: تحميل
- (SCSI: VirtIO SCSI وحدة تحكم) النظام: الافتراضي
- SCSI: الأقراص: 32 جيجابايت، الناقل
- الم❖❖الج: 2 نوى
- الذاكرة: 2048 ميجابايت
- vmbr0 جسر، VirtIO: الشبكة
- Ubuntu وتثبيت خادم VM بدء

تنظيف وإعداد - VM الخطوة 2: داخل

```
# تثبيت cloud-init
sudo apt update
sudo apt install cloud-init qemu-guest-agent -y

# تنظيف بيانات محددة للجهاز
sudo cloud-init clean
sudo rm -f /etc/machine-id /var/lib/dbus/machine-id
sudo rm -f /etc/ssh/ssh_host_*
sudo truncate -s 0 /etc/hostname
sudo truncate -s 0 /etc/hosts

# وإيقاف التشغيل bash مسح تاريخ
history -c
sudo poweroff
```

وتحويل إلى قالب Cloud-Init الخطوة 3: إضافة

- local-lvm (اختار التخزين مثل) CloudInit الأجهزة → إضافة → محرك → VM اختار
- Cloud-Init → المستخدم: omnitouch، كلمة المرور: password
- تمكين → QEMU الأجهزة → الخيارات → وكيل
- تحويل إلى قالب → VM انقر بزر الماوس الأيمن على

3. (فقط LXC لـ) LXC تنزيل قالب

```
# في Proxmox قشرة عقدة:
pveam update
pveam download local ubuntu-24.04-standard_24.04-2_amd64.tar.zst
```

تكوين ملف المضيف

نشر VM (proxmox.yml)

```
all:
  vars:
    proxmoxServers:
      pve-node-01:
        proxmoxServerAddress: 192.168.1.100
        proxmoxServerPort: 8006
        proxmoxRootPassword: YourPassword
        proxmoxApiTokenName: ansible
        proxmoxApiTokenSecret: "your-token-secret-uuid"
        proxmoxTemplateName: ubuntu-2404-template
        proxmoxTemplateId: 9000
        proxmoxNodeName: pve-node-01
        storage: local-lvm # اختياري
      pve-node-02:
        # ... تكوين العقدة الثانية ...

# VM cloud-init بيانات اعتماد المستخدم - يتم استخدام أول مستخدم لـ
local_users:
  admin_user:
    name: Admin User
    public_key: "ssh-rsa AAAA..."
    password: "optional-password" # إذا 'password' الافتراضي هو
    # لم يتم تعيينه

mme:
  hosts:
    site-mme01:
      ansible_host: 192.168.1.10
      gateway: 192.168.1.1
      vlanid: "100" # اختياري
```

LXC (proxmox_lxc.yml) لنشر

```
all:
  vars:
    proxmoxServerAddress: 192.168.1.100
    proxmoxServerPort: 8006
    proxmoxNodeName: ['pve-node-01', 'pve-node-02'] # فردي أو قائمة
    proxmoxApiTokenName: ansible
    PROXMOX_API_TOKEN: "your-token-secret-uuid"
    proxmoxLxcOsTemplate: 'local:vztmpl/ubuntu-24.04-
standard_24.04-2_amd64.tar.zst'
    proxmoxLxcCores: 2
    proxmoxLxcMemoryMb: 4096
    proxmoxLxcDiskSizeGb: 30
    proxmoxLxcRootFsStorageName: local-lvm
    mask_cidr: 24
    host_vm_network: vmbr0

# VM / بيانات اعتماد المستخدم - يتم استخدام أول مستخدم للوصول الأولي إلى
LXC
  local_users:
    admin_user:
      name: Admin User
      public_key: "ssh-rsa AAAA..."
      password: "optional-password" # إذا 'password' الافتراضي هو
      لم يتم تعيينه

  apt_cache_servers:
    hosts:
      site-cache:
        ansible_host: 192.168.1.20
        gateway: 192.168.1.1
        vlanid: "100" # اختياري
        proxmoxLxcDiskSizeGb: 120 # تجاوز لكل مضيف
```

الاستخدام

نشر VMs

```
ansible-playbook -i hosts/Customer/hosts.yml services/proxmox.yml
```

LXC نشر حاويات

```
ansible-playbook -i hosts/Customer/hosts.yml  
services/proxmox_lxc.yml
```

حذف VMs / LXCs

```
ansible-playbook -i hosts/Customer/hosts.yml  
services/proxmox_delete.yml
```

السلوك

proxmox.yml

- Proxmox بنفس الاسم موجودة بالفعل في VM يتحقق مما إذا كانت
- round-robin عبر العقد باستخدام طريقة VMs يوزع
- يستنسخ من القالب
- cloud-init ثابت، العلامات، و IP يقوم بتكوين
- **local_users** من أول إدخال في cloud-init يحدد بيانات اعتماد مستخدم
- VLAN يدعم وضع العلامات

proxmox_lxc.yml

- IP يتحقق من عدم وجود الحاوية بالاسم أو
- round-robin عبر العقد باستخدام طريقة LXCs يوزع

- ثابت IP ينشئ حاوية مع
- SSH ومفتاح sudo مع وصول `local_users` ينشئ تلقائيًا أول حساب
- للشبكات netplan يقوم بتكوين
- يبدأ الحاويات تلقائيًا
- UPF يستثني مضيفي

proxmox_delete.yml

- المطابقة لاسم المضيف في الجرد VM / LXC يوقف ويحذف
- يبحث عبر جميع العقد المكونة
- يتوقف بالقوة بعد 20 ثانية

وضع العلامات VM / LXC توزيع

Round-Robin توزيع

round-robin (modulo) باستخدام منطق Proxmox تلقائيًا عبر عقد LXCs و VMs يتم توزيع

MME: مثال مع 3 محاكيات و5

```
mme01 → pve-node-01 (index 0 % 3 = 0)
mme02 → pve-node-02 (index 1 % 3 = 1)
mme03 → pve-node-03 (index 2 % 3 = 2)
mme04 → pve-node-01 (index 3 % 3 = 0)
mme05 → pve-node-02 (index 4 % 3 = 1)
```

كيف يعمل:

1. مجموعة دور المضيف playbook يحدد (مثل `mme`، `sgw`، `hss`)
2. يحسب فهرس المضيف داخل تلك المجموعة (بدءًا من 0)
3. modulo يستخدم عملية: `host_index % number_of_nodes`
4. يختار المحاكي بناءً على النتيجة

التكوين:

```
# تعريف خوادم متعددة - VMs (proxmox.yml) بالنسبة لـ
proxmoxServers:
  pve-node-01: { ... }
  pve-node-02: { ... }
  pve-node-03: { ... }

# قائمة بعقد متعددة - LXCs (proxmox_lxc.yml) بالنسبة لـ
proxmoxNodeName: ['pve-node-01', 'pve-node-02', 'pve-node-03']
```

وضع العلامات التلقائي

بـ VMs و LXCs يتم وضع علامات تلقائية على:

- التي ينتمي إليها المضيف Ansible **أسماء الدور / المجموعة**: جميع مجموعات
- **اسم الموقع**: متغير `site_name`

مثال:

```
site_name: "melbourne-prod"

mme:
  hosts:
    melbourne-mme01: { ... }
```

بـ VM / LXC **النتيجة**: تم وضع علامة على `mme`، `melbourne-prod`

.ومفيدة للتصفية / التنظيم Proxmox تكون العلامات مرئية في واجهة

تجاوزات لكل مضيف

:تجاوز الافتراضات على مضيفين محددين

hosts:

high-spec-host:

ansible_host: 192.168.1.50

gateway: 192.168.1.1

proxmoxLxcCores: 8 # تجاوز النوى

proxmoxLxcMemoryMb: 16384 # تجاوز الذاكرة

proxmoxLxcDiskSizeGb: 100 # تجاوز القرص

كتيبات الأدوات

التحتية المنفذة. تقع هذه الكتيبات في OmniCore توفر كتيبات الأدوات أدوات تشغيلية لإدارة بنية ويمكن تشغيلها بشكل مستقل لأداء مهام الصيانة واستكشاف الأخطاء `util_playbooks/` دليل الشائعة.

مرجع سريع

الغرض	كتيب الأدوات
إنشاء تقرير شامل عن صحة جميع الخدمات	<code>health_check.yml</code>
و/أو التكوين من النسخة HSS استعادة قاعدة بيانات الاحتياطية	<code>restore_hss.yml</code>
Mermaid إنشاء وثائق الشبكة مع مخططات	<code>ip_plan_generator.yml</code>
تدقيق المنافذ المفتوحة والخدمات المستمعة عبر جميع المضيفين	<code>get_ports.yml</code>
استرجاع ملفات التقاط الحزم من المضيفين	<code>getLocalCapture.yml</code>
إزالة حساب مستخدم محلي من جميع المضيفين	<code>delete_local_user.yml</code>
إلى 9000 (إطارات ضخمة) على واجهات MTU تعيين الشبكة	<code>updateMtu.yml</code>
EPC التحقق من حالة الخدمة على مكونات	<code>systemctl status.yml</code>

فحص الصحة

الملف: `util_playbooks/health_check.yml`

المنفذة OmniCall و OmniCore يغطي جميع خدمات HTML ينشئ تقرير صحة شامل بتنسيق

```
ansible-playbook -i hosts/customer/host_files/production.yml  
util_playbooks/health_check.yml
```

الإخراج: /tmp/health_check_YYYY-MM-DD HH:MM:SS.html

المعلومات التي تم جمعها

المكون	البيانات المجمعة
جميع الخدمات	حالة الخدمة، الإصدار، مدة التشغيل
OmniHSS	Diameter حالة قاعدة البيانات، اتصالات نظير
OmniDRA	والحالة Diameter اتصالات نظير
OmniTAS	المكالمات النشطة، الجلسات، التسجيلات، استخدام وحدة المعالجة المركزية
OCS	KeyDB حالة تكرار

HSS استعادة

الملف: util_playbooks/restore_hss.yml

من ملفات النسخ الاحتياطي. يدعم استعادة قاعدة البيانات فقط، أو التكوين OmniHSS يستعيد فقط، أو كليهما.

```
ansible-playbook -i hosts/customer/host_files/production.yml  
util_playbooks/restore_hss.yml
```

تنسيقات ملفات النسخ الاحتياطي

النوع	نمط اسم الملف	المحتويات
قاعدة البيانات	hss_dump_<hostname>_<timestamp>.sql	لقاعدة MySQL تفرغ بيانات omnihss
التكوين	hss_<hostname>_<timestamp>.tar.gz	أرشفة لمجلد /etc/omnihss

IP مولد خطة

الملف: util_playbooks/ip_plan_generator.yml

ينشئ وثائق الشبكة من الجرد، بما في ذلك:

- (الأساسية والثانوية NICs) للمضيفين IP تعيينات
- نظرة عامة على شريحة الشبكة
- (Diameter, GTP, PFCP, SIP, SS7) مخططات الاتصال بالواجهة

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/ip_plan_generator.yml
```

ملفات الإخراج

الوصف	التنسيق	الملف
وثائق نصية	Markdown	/tmp/ip_plan_<customer>_<site>.md
مخطط تفاعلي مع طبقات قابلة للتصفية	HTML	/tmp/ip_plan_<customer>_<site>.html

تدقيق المنافذ

الملف: `util_playbooks/get_ports.yml`

.يدقق جميع المنافذ المستمعة عبر النشر وينشئ وثائق

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/get_ports.yml
```

ملفات الإخراج

الوصف	الملف
البروتوكول، المنفذ، الخدمة، IP، مع اسم المضيف CSV	<code>/tmp/all_ports.csv</code>
Sphinx لوثائق reStructuredText جدول	<code>./open_ports.rst</code>

البيانات المجمعة

الوصف	الحقل
اسم المضيف في الجرد	اسم المضيف
للمضيف <code>ansible_host</code> الخاص بـ IP عنوان	IP
IPv4 أو IPv6	IP إصدار
TCP أو UDP	النقل
رقم المنفذ المستمع	المنفذ
اسم العملية	الخدمة

استرجاع الالتقاط المحلي

الملف: `util_playbooks/getLocalCapture.yml`

يسترجع أحدث ملفي التقاط حزم من كل مضيف من دليل `/etc/localcapture`.

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/getLocalCapture.yml
```

الإخراج: `./localCapturePcaps/<hostname>/*.pcap`

إدارة المستخدمين

الملف: `util_playbooks/delete_local_user.yml`

يزيل حساب مستخدم محلي من جميع المضيفين في الجرد.

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/delete_local_user.yml
```

المطالبة: أدخل اسم المستخدم للحذف عند المطالبة.

MTU تكوين

الملف: `util_playbooks/updateMtu.yml`

عبر جميع المضيفين `ens160` إلى 9000 (إطارات ضخمة) على واجهة MTU يحدد.

```
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/updateMtu.yml
```

قم بتعديل الكتيب إذا كانت بيئتك تستخدم `ens160`. **ملاحظة:** هذا الكتيب محدد مسبقًا لواجهة `ens160`.
أسماء واجهات مختلفة.

تشغيل كتيبات الأدوات

الصيغة الأساسية

```
ansible-playbook -i <inventory_file> util_playbooks/<playbook>.yaml
```

الخيارات الشائعة

الخيار	الوصف
<code>-i <inventory></code>	تحديد ملف الجرد
<code>--limit <hosts></code>	تقييد إلى مضيفين أو مجموعات محددة
<code>-v</code> / <code>-vv</code> / <code>-vvv</code>	زيادة verbosity
<code>--check</code>	تشغيل جاف (عرض ما سيتغير)
<code>--diff</code>	عرض اختلافات الملفات

أمثلة

```
# تشغيل فحص الصحة على الإنتاج
ansible-playbook -i hosts/acme/host_files/production.yaml
util_playbooks/health_check.yaml
```

```
# على مضيف محدد HSS استعادة
ansible-playbook -i hosts/acme/host_files/production.yaml
util_playbooks/restore_hss.yaml --limit hss01
```

```
# مع إخراج مفصل IP إنشاء خطة
ansible-playbook -i hosts/acme/host_files/production.yaml
util_playbooks/ip_plan_generator.yaml -v
```

