مقدمة في نشر Ansible في Omnitouch

نظرة عامة

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

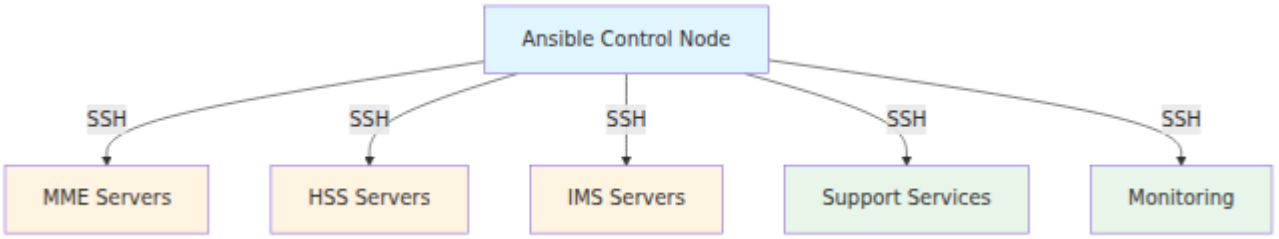
ما هو Ansible؟

Ansible هو أداة أتمتة مفتوحة المصدر تتيح لك:

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

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

كيف تستخدم Omnitouch Ansible



المفاهيم الرئيسية

1. الجرد (ملفات المضيفين)

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

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

ملفات المضيفين هي ما ستعمل به لتعريف شبكتك.

انظر: تكوين ملف المضيفين

2. الأدوار

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

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

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

يتم تعريف هذه من قبل فريق ONS، أي بينما يمكنك تعديلها، هناك عمومًا طرق أنظف لإجراء 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
```

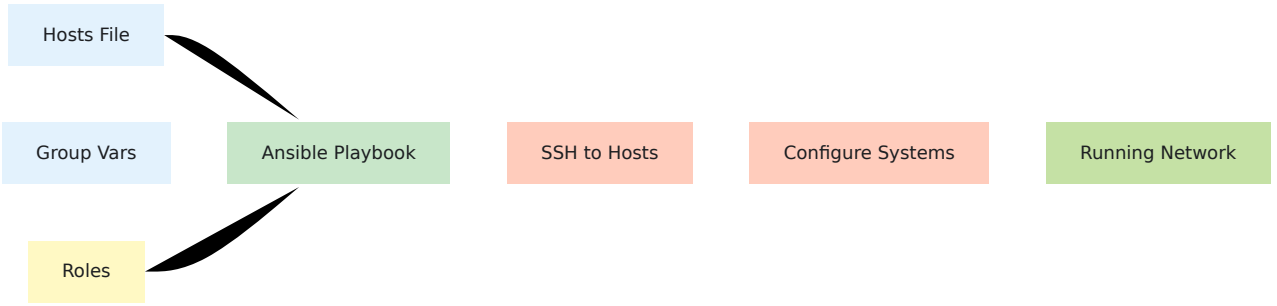
نستخدم هذه أساسًا كمجموعات للأدوار.

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:

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

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

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 بالعميل في `group_vars`:

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

#ToDo - أضف رابطًا هنا إلى مرجع التكوين لقائمة كاملة

3. تشغيل دفاتر التشغيل

نشر الشبكة:

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

4. النشر الآلي

سيقوم Ansible بـ:

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

المكونات الرئيسية التي نقوم بنشرها

OmniCore (منصة النواة الحزمية 4G/5G)

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

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** - خادم تطبيق IMS (خدمات VoLTE/VoNR)
- **OmniMessage** - مركز الرسائل القصيرة (SMS-C)
- **OmniMessage SMPP** - دعم بروتوكول SMPP
- **OmniSS7** - مكونات الإشارة SS7 (STP، HLR، CAMEL)
- **VisualVoicemail** - وظيفة البريد الصوتي

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

OmniCharge/OmniCRM

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

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

خدمات الدعم

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

- **المراقبة** - Prometheus، Grafana

انظر: نظرة عامة على بنية النشر

إدارة الحزم

نستخدم نموذج توزيع حزم هجين:

حزم APT مسبقة التجميع

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

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

نظام ذاكرة التخزين المؤقت APT

يمكن للعملاء الاختيار بين:

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

انظر: نظام ذاكرة التخزين المؤقت APT

إدارة الترخيص

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

- 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

يمكن أن تنشر نفس دفاتر التشغيل الخاصة بـ Ansible:

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

الاتساق

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

التحكم في الإصدار

يتم تعريف البنية التحتية ككود في Git:

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

التخصيص دون تعقيد

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

النشر السريع

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

البدء

المتطلبات المسبقة

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

1. إنشاء بيئة افتراضية Python

قم بإنشاء بيئة Python معزولة لنشر Ansible:

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

2. تفعيل البيئة الافتراضية

قم بتفعيل البيئة الافتراضية:

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

على Windows، استخدم:

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

3. تثبيت الحزم المطلوبة

قم بتثبيت جميع التبعيات من ملف requirements.txt:

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

سيقوم هذا بتثبيت Ansible وجميع حزم Python اللازمة لأتمتة نشر Omnitouch.

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

خطوات النشر

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

الخطوات التالية

- معيار تخطيط IP - **خطط بنية شبكتك وتخصيص IP**
- تكوين ملف المضيفين - تعلم كيفية تعريف طوبولوجيا شبكتك
- نظام ذاكرة التخزين المؤقت APT - افهم توزيع الحزم
- 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

مستودع APT وتوزيع الحزم

نظرة عامة

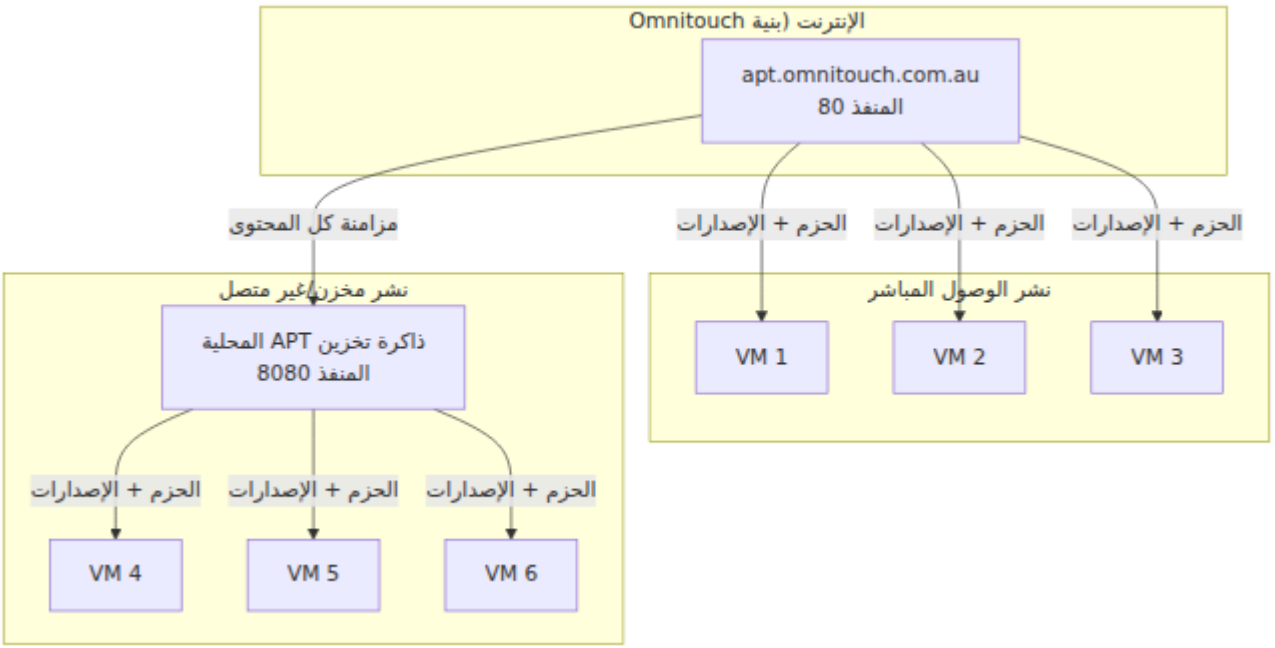
يوفر نظام APT من Omnitouch توزيع الحزم لجميع النشر. يتم تقديم نوعين من المحتوى:

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

يدعم النظام نموذجين للنشر:

1. **الوصول المباشر** — تقوم الآلات الافتراضية بسحب الحزم مباشرة من `apt.omnitouch.com.au`
2. **مرآة التخزين المحلية** Omnitouch — يقوم خادم محلي بمزامنة المحتوى من 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	حزم `.deb` مبنية خصيصًا (omnihss، omnimme، إلخ.)	`/dists/<distro>/`
حزم Ubuntu	حزم Ubuntu المخزنة مع جميع التبعيات	`/<distro>/pool/main/`
إصدارات GitHub	ثنائيات مُعدة مسبقًا (Prometheus، Grafana، Homer، إلخ.)	`/releases/<org>/<repo>/`
أرشيفات المصدر	أرشيفات المصدر لتطبيقات الويب (CGrateS_UI، speedtest)	`/repos/`
حزم الطرف الثالث	Galera، FRR، InfluxDB، KeyDB، إلخ.	`/releases/<vendor>/`

متغيرات التكوين

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

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 عناوين URL لتنزيلات ثنائية	تنزيل الملفات من مسار `/releases/` (Node Exporter، Zabbix، Nagios، إلخ.)

متى يتم استخدام كل مجموعة متغيرات

السيناريو	مصادر APT (`apt_repo`)	تنز��لات ثنائية (`remote_apt_*`)
`use_apt_cache: true`	يستخدم `apt_repo.apt_server`	يستخدم `apt_repo.apt_server`
`use_apt_cache: false`	يستخدم `apt_repo.*` مع بيانات الاعتماد	يستخدم `remote_apt_*` مع بيانات الاعتماد

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

الخيار 1: الوصول المباشر

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

متطلبات الشبكة

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

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

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

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

الخدمات التي تتطلب الوصول إلى بنية Omnitouch التحتية:

الخدمة	المنفذ	البروتوكول	الغرض
خادم APT	80	TCP	تنزيل الحزم
خادم APT	53	TCP/UDP	حل DNS لـ `apt.omnitouch.com.au`
خادم الترخيص	123	UDP	مزامنة الوقت NTP للتحقق من صحة الترخيص
خادم الترخيص	53	TCP/UDP	حل DNS للتحقق من صحة الترخيص

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

التكوين

```yaml
all:
  vars:
    use_apt_cache: false

    # تكوين مصادر حزم APT
    # يقوم بتكوين /etc/apt/sources.list لأوامر apt install
    apt_repo:
      apt_server: "apt.omnitouch.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.omnitouch.com.au"
    remote_apt_port: 80
    remote_apt_protocol: "http"
    remote_apt_user: "your-username"
    remote_apt_password: "your-password"
```

المعلمات

مصادر حزم APT (`apt_repo`)

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`apt_repo.apt_server`	سلسلة	نعم	-	اسم مضيف أو عنوان IP لخادم APT
`apt_repo.apt_repo_username`	سلسلة	نعم	-	اسم مستخدم مصادقة HTTP الأساسية لمصادر APT
`apt_repo.apt_repo_password`	سلسلة	نعم	-	كلمة مرور مصادقة HTTP الأساسية لمصادر APT

التنزيلات الثنائية (`remote_apt_*`)

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

عام

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`use_apt_cache`	Boolean	نعم	-	يجب أن تكون `false` للوصول المباشر

أنماط URL (الوصول المباشر)

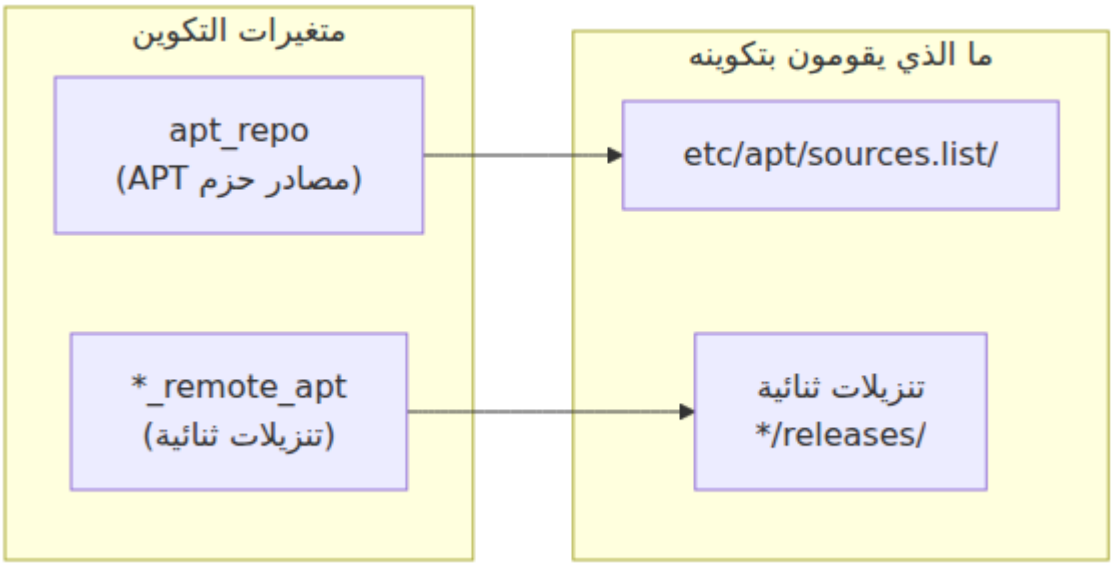
مصادر حزم APT (المكونة في `/etc/apt/sources.list`):

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

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

```
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



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

الخيار 2: مرآة التخزين المحلية

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

الهيكلية

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:
    # use_apt_cache: true  # يتم تعيينه تلقائيًا عند وجود مجموعة خوادم apt_cache
    # apt_repo.apt_server: 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 الحزم من `apt.omnitouch.com.au:80`

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

المعلمات

مصادر حزم APT (`apt_repo`)

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`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 خدام التخزين (لا حاجة للمصادقة)
`apt_repo.apt_repo_password`	سلسلة	لا	-	غير مطلوب عند خدام التخزين (لا حاجة للمصادقة)

مزامنة خادم التخزين (`remote_apt_*`)

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

المعلمة	النوع	مطلوب	الافتراضي	الوصف
`remote_apt_server`	سلسلة	نعم	-	خادم APT من Omnitouch للمزامنة منه
`remote_apt_port`	عدد صحيح	لا	`80`	منفذ APT من خادم Omnitouch
`remote_apt_protocol`	سلسلة	لا	`http`	ا��بروتوكول لمزامنة الاتصال
`remote_apt_user`	سلسلة	نعم	-	بيانات الاعتماد للمزامنة من Omnitouch
`remote_apt_password`	سلسلة	نعم	-	بيانات الاعتماد للمزامنة من Omnitouch

عام

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

أنماط URL (وضع التخزين)

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

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

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

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

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

نشر التخزين

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

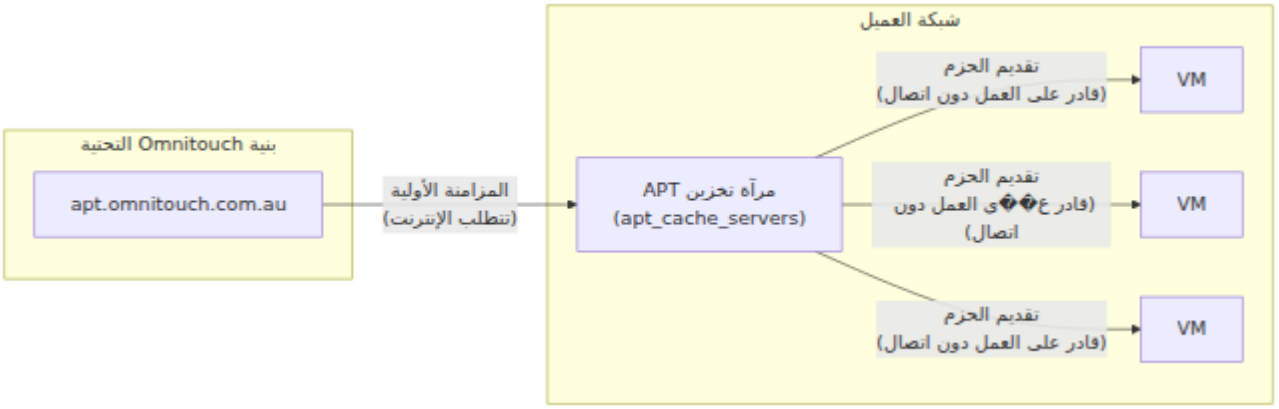
تقوم مرآة التخزين بمزامنة **كل المحتوى** من APT من Omnitouch باستخدام تنزيل wget التكراري:

مجلدات المحتوى التي تم مزامنتها:

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

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

كيف يعمل



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

التكوين التلقائي

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

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

مثال على التكوين الأدنى

```yaml
apt_cache_servers:
  hosts:
    apt-cache-01:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # يقوم خادم التخزين بمزامنة المحتوى من مستودع Omnitouch
    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"`
- يسحب جميع المضيفين من `http://192.168.1.100:8080/` دون بيانات اعتماد
- 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:

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

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

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

ملخص التكوين

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

تقوم جميع المضيفين بسحب الحزم مباشرة من خادم مستودع APT.

```yaml
all:
  vars:
    use_apt_cache: false

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

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

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

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

خادم التخزين موجود في جردك وسيتم نشره/مزامنته بواسطة Ansible.

```
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.omnitouch.com.au"
    remote_apt_port: 80
    remote_apt_protocol: "http"
    remote_apt_user: "user"
    remote_apt_password: "pass"

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

النتيجة:

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

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

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

```yaml
all:
  vars:
    use_apt_cache: true

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

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

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

مثال كامل

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

```yaml
# مجموعة خادم تخ??زين APT
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 (مفعل 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`):**
   - يستخدم `remote_apt_*` من قسم `vars:` الخاص به
   - يقوم بتنزيل جميع الحزم من `http://customer-username:customer-secure-token@apt.omnitouch.com.au:80/`
   - يقدم الحزم على المنفذ 8080 عبر nginx
2. **مضيفو التطبيقات (`customer-hss01`، `customer-mme01`، `customer-dns01`):**
   - يكتشفون تلقائيًا وجود مجموعة `apt_cache_servers`
   - يتم تعيين `use_apt_cache: true` تلقائيًا
   - يتم اشتقاق `apt_repo.apt_server: "10.179.1.114"` تلقائيًا
   - يتم إنشاء: `deb [trusted=yes] http://10.179.1.114:8080/noble noble main`
   - تسحب جميع الحزم من خادم التخزين (بدون حاجة لبيانات اعتماد)

تحديث التخزين

لمزامنة حزم جديدة أو تحديثات:

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

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

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

- أرشيفات مصدر محدثة

لذا يتم تخطي الملفات غير المتغيرة الموجودة، مما `wget --timestamping` تستخدم المزامنة يجعل إعادة ال��زامنة سريعة.

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

استكشاف الأخطاء وإصلاحها

فشل تحديث APT مع 401 غير مصرح به

الأعراض:

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

الأسباب المحتملة:

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

الحل:

1. **تحقق من نطاق التكوين**: تأكد من تعريف `apt_repo` في مع بيانات الاعتم��د `apt_cache_servers: vars:`، وليس في `all: 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/omnitouch.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. تأكد من أن عنوان IP العام الخاص بك مدرج في القائمة البيضاء مع Omnitouch (إذا كنت تستخدم الوصول المباشر)

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

الأعراض: فشل برنامج Ansible في تنزيل الملفات من مسار `/releases/`

الأسباب المحتملة:

- لم يتم تكوين متغيرات `remote_apt_*`
- اسم المستخدم `remote_apt_user` أو كلمة المرور `remote_apt_password` غير صح��حة
- عدم وجود `remote_apt_server` عند `use_apt_cache: false`

الحل:

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

فشل خادم التخزين في المزامنة

الأعراض: فشل برنامج خادم التخزين في تنزيل الحزم

الأسباب المحتملة:

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

- جدار الحماية يمنع الاتصالات الخارجية إلى Omnitouch

الحل:

1. تحقق من أن خادم التخزين يمكنه الوصول إلى `apt.omnitouch.com.au` على 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 للمعلمات
- بنية النشر — بنية النظام العامة
- نشر Proxmox — نشر خادم التخزين كحاوية LXC

مرجع التكوين

نظرة عامة

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

تستخدم نشرات OmniCore الحديثة نموذج تكوين مبسط:

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

تخطيط الشبكة

للحصول على إرشادات حول IP **قبل تكوين المضيفين**، راجع معيار تخطيط:

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

معلمات المضيف الشائعة

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

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               # ملء CRM بالبيانات الأولية
```

المتغيرات العالمية (all:vars)

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

المتغيرات العالمية الأساسية

المصادقة والوصول

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

بديل: استخدم مفاتيح SSH بدلاً من كلمات المرور:

```
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'       # MNC (دائمًا 3 أرقام) مع صفر بادئ

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

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

أسماء الشبكة

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

معروض: أسماء الشبكة المعروضة على أجهزة UE الشبكة المحمولة في الإعدادات >.

تكوين DNS

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

تكوين مستودع APT

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

- `use_apt_cache` افتراضيًا إلى `True` (إلى `False` ما لم يتم تعيينه صراحة)

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

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

apt_repo:
  apt_server: "10.10.1.114"      # خادم تخزين APT أو خادم المستودع
  # بيانات الاعتماد مطلوبة فقط عند use_apt_cache: False
  # 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
#           (2) عندما تكون Omnitouch مزامنة خادم التخزين المؤقت من
use_apt_cache: true
remote_apt_server: "apt.omnitouch.com.au"
remote_apt_user: "omni"
remote_apt_password: "omni"
```

انظر: نظام تخزين APT

خادم الترخيص

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

انظر: خادم الترخيص

إعدادات MME

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

إعدادات SAEGW

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

إعدادات IMS

```
ims_dra_support: False          # توجيه IMS عبر DRA
enable_homer: False             # تمكين التقاط SIP من Homer
```

تكوين مراقبة RAN

```
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'
```

خوادم DNS للتجوال

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

المستخدمون المحليون (مفاتيح SSH)

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

تكوين Hypervisor

Proxmox

```
proxmoxServers:
  customer-prxmx01:
    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

# إعدادات Proxmox الافتراضية
proxmoxServerAddress: 10.10.0.100
proxmoxServerPort: 8006
proxmoxNodeName: 'pve01'
proxmoxLxcOsTemplate: '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

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

وثائق المنتج

للحصول على أدلة تشغيل مفصلة وتكوين متقدم:

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

- **مكونات OmniCall**: https://docs.omnitouch.com.au/docs/repos/OmniCall
- **OmniCharge/OmniCRM**: https://docs.omnitouch.com.au/docs/repos/OmniCharge

نظرة عامة على بنية النشر

نظرة عامة

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

للحصول على تفاصيل حول وضع المكونات، وإرشادات تخصيص عناوين IP، راجع معيار تخطيط IP العامة والتعامل مع عناوين IP.

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

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

راجع: نشر VM/LXC على Proxmox

1. تعر��ف البنية التحتية (ملف المضيفين)

```yaml
# 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 مجلد `group_vars` المضيف أو الموقع أو الشبكة.

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

راجع: تكوين المتغيرات الجماعية

3. توزيع الحزم (APT Cache)

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

راجع: نظام APT Cache

4. تكوين الترخيص

```yaml
# 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` ستتعامل مع كل شيء، ويمكنك استخدام `--limit=myhost` أو `--limit=mmee,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 - مقدمة في نشر Ansible
- تعريف البنية التحتية - تكوين ملف المضيفين
- معيار تخطيط IP - **بنية الشبكة IP تخصيص**
- التخصيص - تكوين المتغيرات الجماعية
- إدارة الحزم - نظام APT Cache
- إدارة الترخيص - خادم 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

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

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

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

كيف يعمل

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

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

المثال 1: قالب تكوين مخصص (OmniMessage)

بعض المكونات تقبل قوالب تكوين Jinja2 مخصصة.

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 القالب الخاص بك في group_vars
```

ماذا يحدث:

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

بدون `smsc_template_config`، يتم استخدام القالب الافتراضي من الدور.

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

---

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

بعض المكونات تستخدم أدلة من ملفات التكوين.

هيكل الملف

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

المرجع في ملف المضيفين

```yaml
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/Customer/group_vars/gateways_prod/` إلى `/etc/freeswitch/sip_profiles/`
3. ينسخ جميع الملفات من `hosts/Customer/group_vars/dialplan/` إلى دليل قوالب OmniTAS
4. تقوم الخدمات بتحميل ا��تكوينات

بيئات مختلفة: استخدم مجلدات مختلفة لكل بيئة:

- `gateways_folder: "gateways_lab"`

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

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

---

المثال 3: قالب تكوين مخصص (OmniHSS)

يقبل خادم المشتركين المنزلي قوالب تكوين وقت التشغيل المخصصة.

هيكل الملف

```
hosts/Customer/
└── group_vars/
    └── hss_runtime.exs.j2        # قالب تكوين HSS المخصص الخاص بك
```

المرجع في ملف المضيفين

```yaml
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 القالب الخاص بك في group_vars
```

ماذا يحدث:

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

5. يعيد تشغيل الخدمة

بدون `hss_template_config`، يتم استخدام القالب الافتراضي من الدور.

تفاصيل التكوين: انظر https://docs.omnitouch.com.au/docs/repos/OmniCore

---

المثال 4: قالب تكوين مخصص (OmniMME)

تقبل وحدة إدارة الحركة قوالب تكوين وقت التشغيل المخصصة.

هيكل الملف

```
hosts/Customer/
└── group_vars/
    └── mme_runtime.exs.j2        # قالب تكوين MME الخاص بك المخصص
```

المرجع في ملف المضيفين

```yaml
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 القالب الخاص بك في group_vars
```

ماذا يحدث:

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

4. ينشر إلى `/etc/omnimme/runtime.exs`
5. يعيد تشغيل الخدمة

بدون `mme_template_config`، يتم استخدام القالب الافتراضي من الدور.

تفاصيل التكوين: انظر https://docs.omnitouch.com.au/docs/repos/OmniCore

---

مثال هيكل الدليل في العالم الحقيقي

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

المعلمات الشائعة التي تشير إلى

group_vars

المعلمة	المكون	المراجع
`smsc_template_config`	omnimessage	ملف قالب Jinja2 (مثل `smsc_controller.exs`)
`smsc_smpp_template_config`	omnimessage_smpp	ملف قالب Jinja2 (مثل `smsc_smpp.exs`)
`gateways_folder`	applicationserver	اسم المجلد (مثل `sip_profiles`)
خطط الاتصال (تلقائي)	applicationserver	مجلد `dialplan/` لملفات التوجيه XML
`tas_template_config`	applicationserver	ملف قالب Jinja2 (مثل `tas_runtime.exs.j2`)
`hss_template_config`	omnihss	ملف قالب Jinja2 (مثل `hss_runtime.exs.j2`)
`mme_template_config`	omnimme	ملف قالب Jinja2 (مثل `mme_runtime.exs.j2`)
`dra_template_config`	dra	ملف قالب Jinja2 (مثل `dra_runtime.exs.j2`)
`pgwc_template_config`	pgwc	ملف قالب Jinja2 (مثل `pgwc_runtime.exs.j2`)
`frr_template_config`	omniupf	ملف قالب Jinja2 (مثل `frr.conf.j2`)

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

النقاط الرئيسية

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

متى تستخدم group_vars

✅ استخدم group_vars لـ:

- قوالب تكوين مخصصة للمكو��ات
- تعريفات بوابة SIP
- خطط توجيه المكالمات
- قواعد توجيه Diameter
- إعدادات خاصة بالعميل تتجاوز الافتراضات

❌ لا تستخدم group_vars لـ:

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

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

الوثائق ذات الصلة

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

كتيبات الأدوات

نظرة عامة

يتضمن هذا المستودع العديد من كتيبات الأدوات للصيانة والمراقبة والمهام التشغيلية. تكمل هذه الكتيبات كتيبات النشر الرئيسية مع قدرات الإدارة اليومية.

أداة فحص الصحة

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

تعمل تلقائيًا كجزء من كتيب `services/all.yml`.

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

يعرض تقرير HTML:

معلومات المضيف

- **اسم المضيف وعنوان IP**
- **الشبكة/الشبكة الف��عية** (من متغير `host_vm_network`، أو غير متاح إذا لم 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** (التوزيعة والإصدار)

حالة الخدمة

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

أقران HSS Diameter

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

أدوات شائعة أخرى

تكوين النظام الأساسي

الدور الشائع (`services/common.yml`)

- يطبق تكوين النظام الأساسي على جميع المضيفين
- يقوم بإعداد المستودعات، مفاتيح SSH، المنطقة الزمنية، NTP
- يقوم بتكوين الشبكات وتقوية النظام
- قم بتشغيل هذا قبل ��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
- يدير مفاتيح SSH وامتيازات sudo
- 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

مولد خطة IP (`util_playbooks/ip_plan_generator.yml`)

- يولد تقرير HTML لتعيينات عناوين IP
- 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
```

نسخة احتياطية لـ HSS (`util_playbooks/hss_backup.yml`)

- يقوم بعمل نسخة احتياطية من جداول قاعدة بيانات HSS

- ينسخ تفريغ MySQL إلى آلة Ansible المحلية
- 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
- يسترجع ملفات pcap من `/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`)

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

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

الوثائق ذات الصلة

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

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
- أماكن تشغيلها (عناوين IP، شرائح الشبكة)
- كيفية تكوينها (معلمات محددة للخدمة)
- إعدادات خاصة بالعميل (PLMN، الميزات، بيانات الاعتماد)

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:

```yaml
# 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'
```

معلمات المضيفين الشائعة

تكوين الشبكة

عادةً ما يتضمن كل مضيف:

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

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

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

تخصيص موارد VM

بالنسبة للخدمات التي تحتاج إلى موارد محددة:

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

معلمات محددة للخدمة

لكل وظيفة شبكة معلماتها الخاصة. أمثلة:

MME:

```
mme_code: 1                       # معرف MME (1-255)
mme_gid: 1                        # معرف مجموعة MME
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:                         # تجمعات IP للمشتركين
  - '100.64.16.0/24'
  - '100.64.17.0/24'
combined_CP_UP: false             # فصل التحكم / خطة المستخدم
```

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

خادم التطبيق:

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

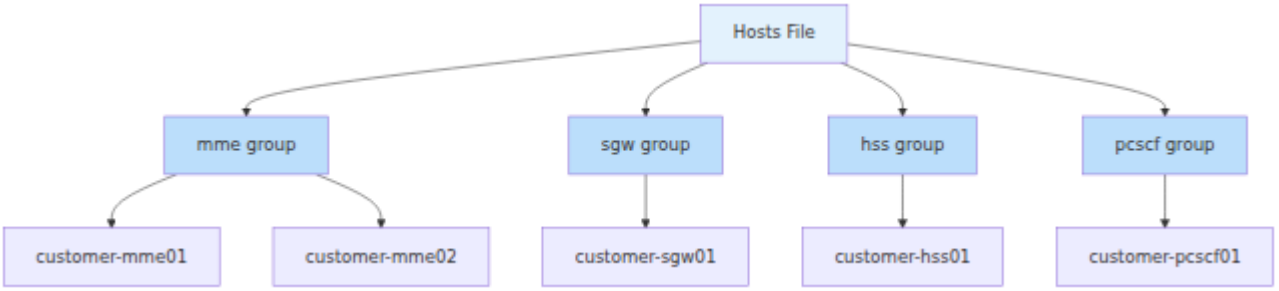
قسم المتغيرات العالمية

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

```yaml
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

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

    # أسماء الشبكات
    network_name_short: Customer
    network_name_long: Customer Network

    # مستودع APT
    # ملاحظة: إذا تم تعريف مجموعة apt_cache_servers للمضيفين، مع
    # apt_repo.apt_server و use_apt_cache تكون افتراضية على true 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

تنظم Ansible المضيفين في مجموعات تتوافق مع الأدوار:

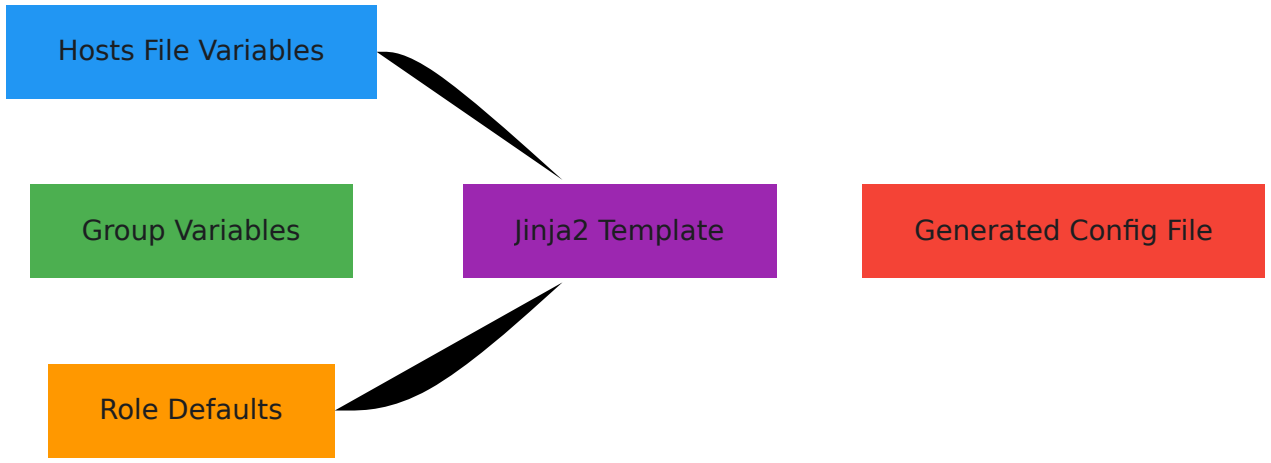


فإنه ينطبق على جميع المضيفين في قسم ، عند تشغيل كتاب اللعب المستهدف `mme` `mme:hosts:`.

التكوين باستخدام قوالب Jinja2

لإنشاء ملفات التكوين من المتغيرات المحددة في ملف Ansible تستخدم **قوالب Jinja2** المضيفين وgroup_vars.

كيفية عمل Jinja2



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
```

أنماط Jinja2 الشائعة

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 اكتمالًا (مع بيانات حساسة مخفية):

```yaml
# 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"
      cdrs_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-prxmx01:
        proxmoxServerAddress: 10.10.0.100
        proxmoxServerPort: 8006
        proxmoxApiTokenName: Customer
        proxmoxApiTokenSecret: "token-secret"
        proxmoxTemplateName: ubuntu-24.04-cloud-init-template
        proxmoxNodeName: pve01
```

راجع نشر Proxmox VM/LXC للحصول على تفاصيل إعداد وتكوين Proxmox الكاملة.

مراجع الوثائق الخاصة بالمنتج

للحصول على تكوين مفصل لكل مكون، راجع الوثائق الرسمية للمنتج:

مكونات OmniCore:

- **وثائق OmniCore**: 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** - وكيل توجيه Diameter
- **OmniTWAG** - بوابة الوصول WLAN الموثوقة

مكونات OmniCall:

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

OmniCharge/OmniCRM:

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

الوثائق ذات الصلة

- مقدمة في نشر Ansible - عملية النشر العامة
- مرجع التكوين - **دليل كامل لجميع متغيرات التكوين**
- تكوين متغيرات المجموعة - تجاوز التكوينات الافتراضية
- معيار تخطيط IP - **إرشادات بنية الشبكة وتخصيص IP**
- تكوين Netplan - **IP ثانوي وتكوين شبكة متقدم**
- نظام تخزين APT - توزيع الحزم
- 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. حدد PLMN وهويتك الشبكية
3. قم بتكوين الوصول إلى مستودع APT
4. إعداد خادم الترخيص
5. تخصيص باستخدام group_vars حسب الحاجة
6. نشر باستخدام كتب لعب Ansible

معيار تخطيط IP لـ OmniCore

نظرة عامة

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

متطلبات تخصيص IP

التخصيص القياسي: أربع شبكات فرعية /24

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

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

مهم: هذه توصيات، ليست متطلبات

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

- **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 لكل نوع من المكونات (واحدة لـ HSS، واحدة لـ MMEs، إلخ.)
- **تجميعات مخصصة**: يمكنك تنظيم المضيفين في أي هيكل شبكة فرعية يتناسب مع متطلباتك الخاصة

- **مزيج من IPs الداخلية والعامة**: يمكن لبعض المضيفين استخدام عناوين داخلية 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، ووضوح العمليات** IP، 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 ��)

الغرض: عناصر خطة المستخدم وخطة التحكم الأساسية

المكونات:

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

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)

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
- رابط التحذير Omnitouch CBC (مركز البث الخلوي) - إذا تم نشره
- خوادم ذاكرة التخزين المؤقت APT - إذا تم نشرها

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. شبكة IMS (24/ ثالث) الداخلية

الغرض: إشارات وخدمات IMS الأساسية (إشارات SIP الداخلية)

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 (وحدة التحكم في الرسائل القصيرة، SMPP، IMS)
- OmniSS7 STP (نقطة نقل إشارات SS7)
- OmniSS7 HLR (سجل الموقع المنزلي) - 2G/3G
- OmniSS7 IP-SM-GW (MAP SMSc)

- 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. شبكة UE (24/ رابع) العامة

الغرض: خدمات موجهة للمستخدم مثل IMS و DNS

المكونات:

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

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

يدعم OmniCore طريقتين رئيسيتين لتنفيذ هذا التقسيم الشبكي:

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

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

مثال:

```yaml
# 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"
```

الطريقة 2: تقسيم قائم على VLAN

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

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:

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

مثال على خريطة VLAN:

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

العامة IP العمل مع عناوين

نظرة عامة

عامة للاتصال IP أن تحتوي بعض المكونات على عناوين OmniCore تتطلب العد��د من نشرات الخارجي، مثل:

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

العامة IPs كيفية تخصيص

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

مثال: IPs العامة للتجوال مع SGW/PGW:

```yaml
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'
```

مثال: DRA مع IP عام

```yaml
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"
```

مثال: ePDG مع IP عام

```yaml
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"
```

مزيج من IPs الداخلية والعامة

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

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

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

خادم الترخيص

نظرة عامة

يدير خادم الترخيص تفعيل الميزات لجميع مكونات Omnitouch. يتحقق كل مكون من ترخيصه 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` (المقدم من Omnitouch) في `hosts/Customer/group_vars/`

3. النشر

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

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

متطلبات الشبكة

تكوين جدار الحماية

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

الغرض	عنوان IP	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: 160.22.43.18, 160.22.43.66, 160.22.43.114
- الاتجاه: خارجي

متطلبات DNS

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

تكوين DNS المطلوب:

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

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

الوثائق ذات الصلة

- مرجع التكوين
- تكوين ملف المضيفين

تكوين Netplan

نظرة عامة

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

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

تمكين تكوين Netplan

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

```
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

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

```
network:
  version: 2
  ethernets:
    # تستخدم - الواجهة الرئيسية ansible_host و gateway من الجرد
    eth0:
      addresses:
      - "{{ ansible_host }}/{{ mask_cidr | default(24) }}"
      nameservers:
        addresses:
{% if 'dns' in group_names %}
        # إذا كان هذا المضيف هو خادم DNS، استخدم DNS خارجي لتجنب
الاعتماد الدائري
        - 8.8.8.8
{% else %}
        # خلاف ذلك، استخدم خوادم DNS من مجموعة 'dns' في الجرد
{% 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 %}
    # حلقة عبر - الواجهات الثانوية dict secondary_ips من الجرد
    # 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 %}
{% endif %}
```

نقاط رئيسية:

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

تكوين الواجهة الرئيسية

يتم تكوين الواجهة الرئيسية (eth0) تلقائيًا باستخدام:

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

يتم تعيين خوادم DNS تلقائيًا إلى:

- المضيفين في مجموعة `dns` (تستخدم عناوين IP الخاصة بهم `ansible_host`)
- تتراجع إلى DNS `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 تكوين `secondary_ips`.

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

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

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

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

مثال على التكوين

```yaml
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: "vmbr0"
          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:

```yaml
network:
  version: 2
  ethernets:
    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"
```

تكامل Proxmox

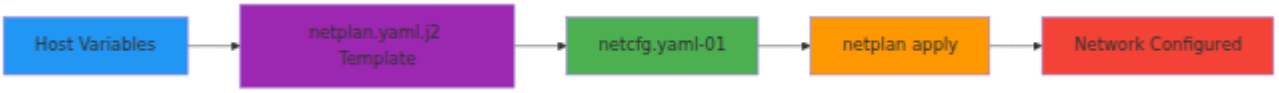
عند استخدام playbook `proxmox.yml`، يتم إنشاء NICs على الثانوية تلقائيًا VM:

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

يستخدم تكوين Proxmox:

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

كيف يعمل



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

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 عام لشركاء التجوال
      gateway: 192.0.2.1
      host_vm_network: "vmbr0"
      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"
```

الإشارة إلى IPs الثانوية في القوالب

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

على نفس المضيف

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

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

# أو بشكل صريح عبر inventory_hostname (نفس النتيجة)
{{ 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

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

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

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

مثال: تكوين نظير DRA

قم بتكوين نظير قطر لربطه بعنوان IP العام الخاص به:

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

التحقق مما إذا كانت IPs الثانوية موجودة

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

قم بتسجيل الدخول إلى VM وتحقق من أسماء الواجهات:

```
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> ...
```

تحقق من تكوين Netplan

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

تطبيق Netplan يدويًا

```
netplan apply
```

تصحيح Netplan

```
netplan --debug apply
```

تحقق من الطرق

```
ip route show
```

الوثائق ذات ا��صلة

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

نشر VM/LXC في Proxmox

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

نواصل دعم VMware وHyperV والسحابة (حالياً Vultr / AWS / GCP) للنشر.

انظر أيضًا:

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

VM مقابل LXC

حاويات LXC:

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

الآلات الافتراضية (KVM):

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

- **مطلوبة لنشر UPF**

حالات الاستخدام:

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

إعداد Proxmox

1. إنشاء رمز API

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

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

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

```
#!/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 المقتطفات ==="
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 "=== تم إنشاء القالب $TEMPLATE_ID بنجاح ==="
```

ملاحظات:

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

بديل: قالب يدوي من ISO

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

الخطوة 1: إنشاء VM عبر واجهة الويب

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

الخطوة 2: داخل VM - تنظيف وإعداد

```
# 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

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

الخطوة 3: إضافة Cloud-Init وتحويل إلى قالب

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

3. تنزيل قالب LXC (لـ LXC فقط)

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

تكوين ملف المضيف

لنشر VM (proxmox.yml)

```yaml
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

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

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

لنشر LXC (proxmox_lxc.yml)

```yaml
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

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

نشر حاويات LXC

```
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

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

proxmox_lxc.yml

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

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

proxmox_delete.yml

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

ووضع العلامات VM / LXC توزيع

Round-Robin توزيع

يتم توزيع VMs وLXCs تلقائيًا عبر عقد Proxmox باستخدام منطق round-robin (modulo):

مثال مع 3 محاكيات و5 MME:

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

التكوين:

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

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

وضع العلامات التلقائي

يتم وضع علامات تلقائية على VMs وLXCs بـ:

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

مثال:

```yaml
site_name: "melbourne-prod"

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

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

تكون العلامات مرئية في واجهة Proxmox ومفيدة للتصفية / التنظيم.

تجاوزات لكل مضيف

تجاوز الافتراضات على مضيفين محددين:

```yaml
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

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

مرجع سريع

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

فحص الصحة

الملف: `util_playbooks/health_check.yml`

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

```
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	حالة قاعدة البيانات، اتصالات نظير Diameter
OmniDRA	اتصالات نظير Diameter والحالة
OmniTAS	المكالمات النشطة، الجلسات، التسجيلات، استخدام وحدة المعالجة المركزية
OCS	حالة تكرار KeyDB

استعادة HSS

الملف: `util_playbooks/restore_hss.yml`

يستعيد OmniHSS من ملفات النسخ الاحتياطي. يدعم استعادة قاعدة البيانات فقط، أو التكوين 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`	تفريغ MySQL لقاعدة بيانات `omnihss`
التكوين	`hss_<hostname>_<timestamp>.tar.gz`	أرشيف لمجلد `/etc/omnihss`

مولد خطة IP

الملف: `util_playbooks/ip_plan_generator.yml`

ينشئ وثائق الشبكة من الجرد، بما في ذلك:

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

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

ملفات الإخراج

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

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

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

البيانات المجمعة

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

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.

تكوين MTU

الملف: `util_playbooks/updateMtu.yml`

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

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

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

تشغيل كتيبات الأدوات

الصيغة الأساسية

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

الخيارات الشائعة

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

أمثلة

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

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

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