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. ومع ذلك، فإن البنية المعمارية **مرنة تمامًا**:

- **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 (HSS واحدة لـ، MMEs واحدة لـ، 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
- عملاء 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

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)

كيفية تخصيص IPs العامة

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

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

مقدمة في نشر Ansible في Omnitouch

نظرة عامة

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

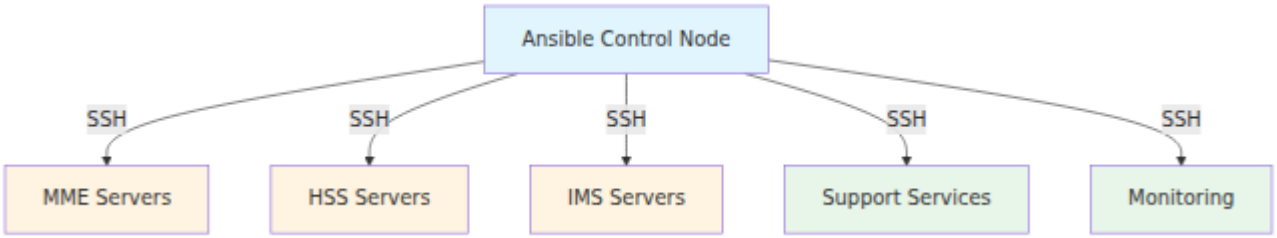
ما هو Ansible؟

Ansible هو أداة أتمتة مفتوحة المصدر تتيح لك:

- 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.

كيف تستخدم Omnitouch Ansible



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
- عناوين IP الخاصة بها
- تكوين الشبكة
- معلمات 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)

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

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

3. دفاتر التشغيل

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

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

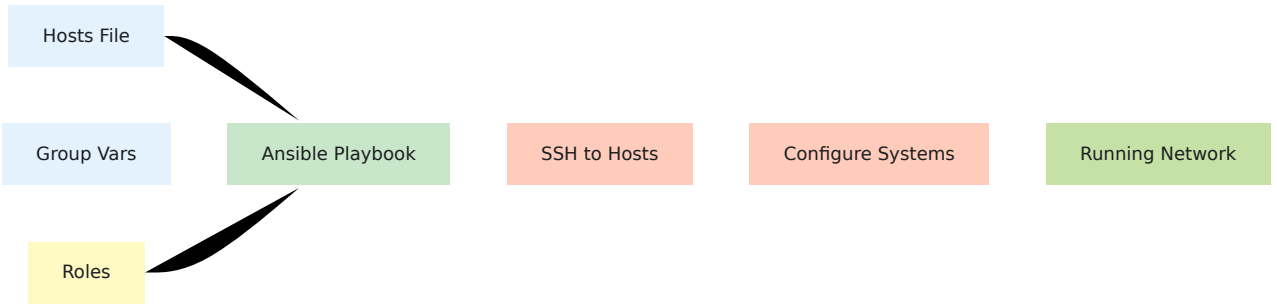
نستخدم هذه أساسًا كمجموعات للأدوار.

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 VMs/حاويات تلقائيًا.

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

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

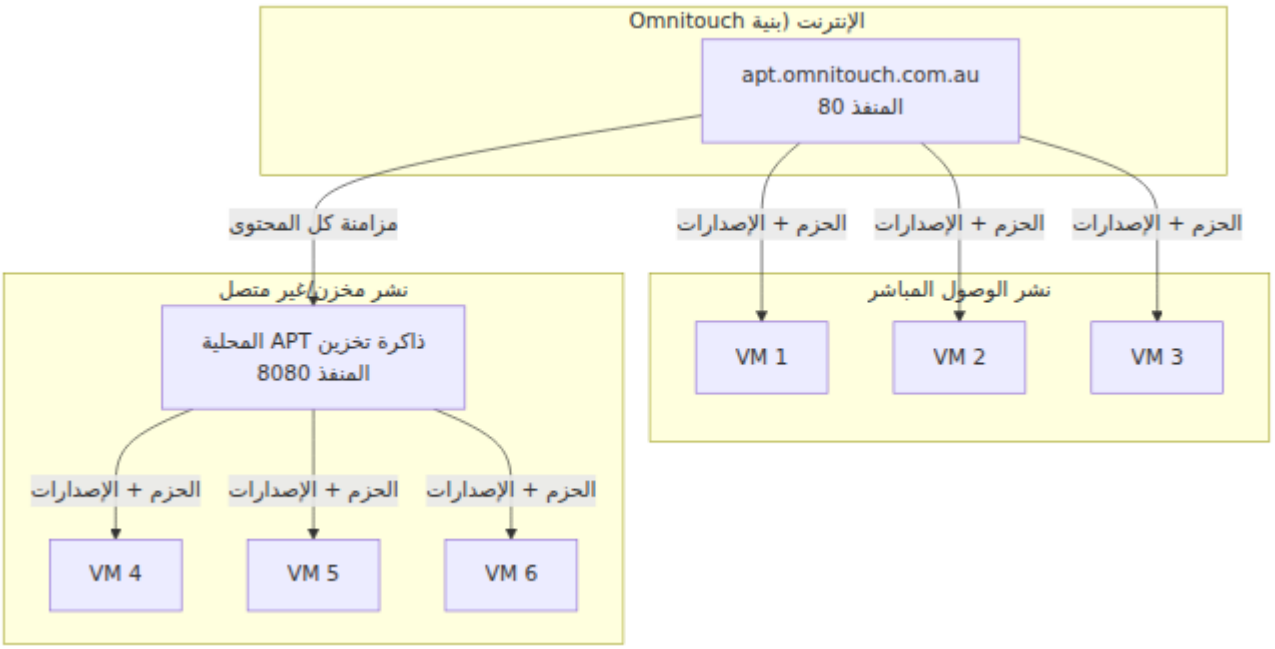
يوفر نظام APT من Omnitouch توزيع الحزم لجميع النشر. يتم تقديم نوعين من المحتوى:

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 IP: 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)

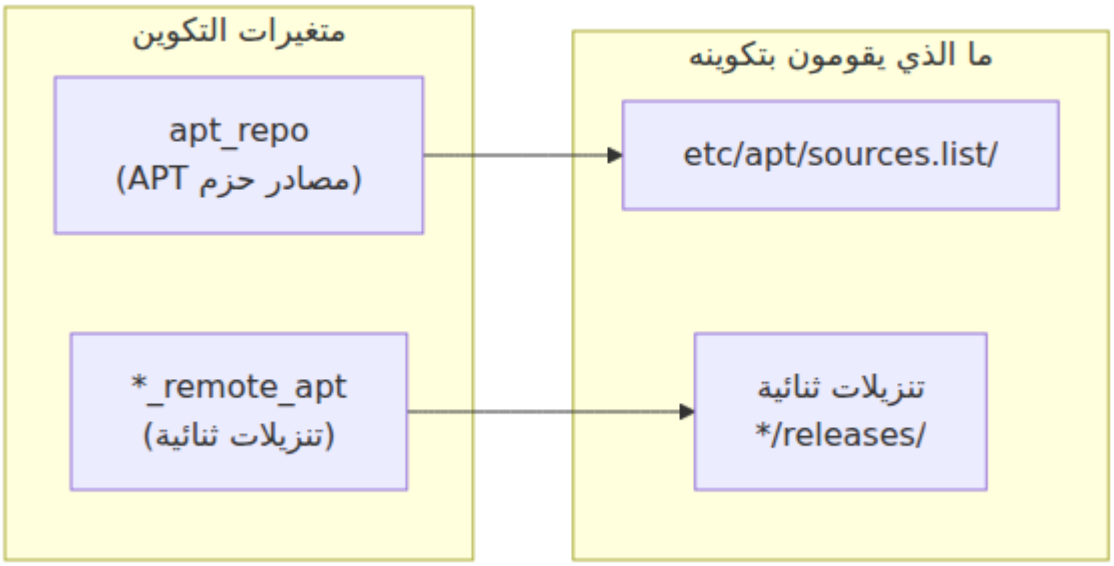
مصادر حزم APT (المكونة في `/etc/apt/sources.list`):

```
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: 192.168.1.100 يتم اشتقاقه تلقائيًا إلى (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	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 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:

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

عام

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

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

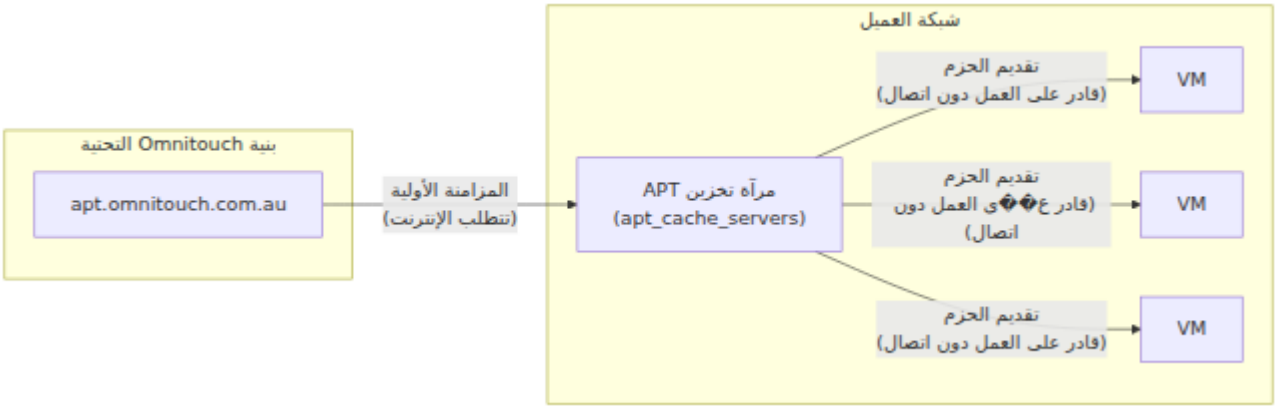
تقوم مرآة التخزين بمزامنة **كل المحتوى** من APT من Omnitouch باستخدام تنزيل wget التكراري:

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

التكوين التلقائي

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

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

# apt_cache_servers لا حاجة لمجموعة
# remote_apt_* لا حاجة لـ (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 `services/apt_cache.yml` بتشغيل على مرايا التخزين لمزامنتها.

استكشاف الأخطاء وإصلاحها

فشل تحديث APT به مصرح غير 401 مع

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

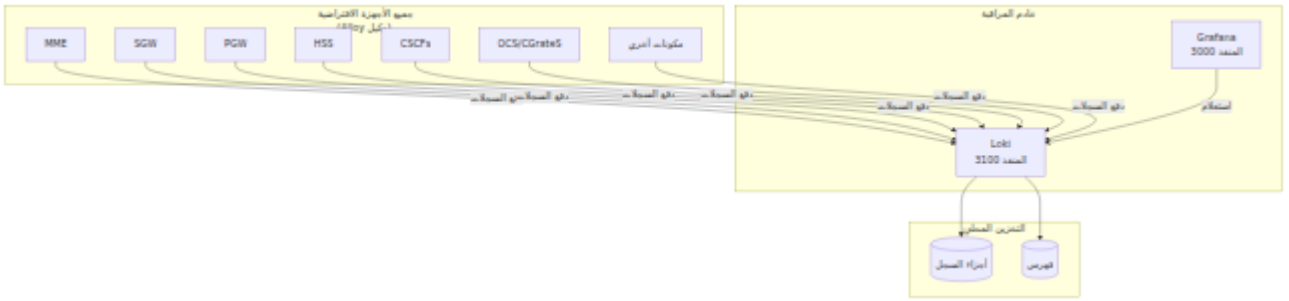
تسجيل مركزي

نظرة عامة

نظام تسجيل مركزي يجمع السجلات من جميع المكونات ويجعلها قابلة للبحث OmniCore يتضمن من خلال Grafana. يستخدم النظام:

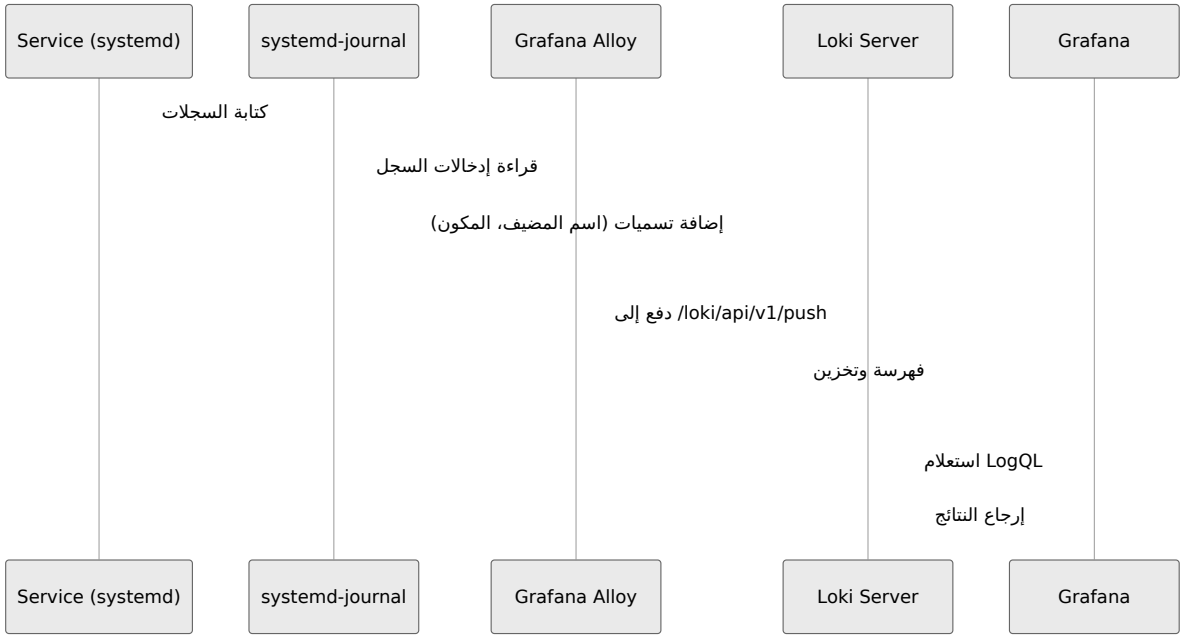
- **Grafana Alloy** — وكيل جمع السجلات يتم نشره على جميع الأجهزة الافتراضية
- **Grafana Loki** — خادم تجميع وتخزين السجلات على مضيفي المراقبة
- **لوحات معلومات Grafana** — لوحات معلومات مسبقة البناء لكل نوع من المكونات

الهيكلية



كيفية العمل

تدفق جمع السجلات



تسميات المكونات

يتم تلقائيًا تسمية كل إدخال سجل بناءً على عضويته في مجموعة جرد المضيف:

مجموعة الجرد	تسمية المكون	الخدمات المجمعة
`mme`	`mme`	omnimme, open5gs
`sgw`	`sgw`	omnisgwc, open5gs
`pgw`	`pgw`	omnipgwc, open5gs
`upf`	`upf`	omniupf, frr
`hss`	`hss`	omnihss, pyhss, mysql
`dra`	`dra`	omnidra, freediam
`pcscf`, `scscf`, `icscf`	`cscf`	kamailio, rtpengine
`applicationserver`	`applicationserver`	freeswitch, omnitas
`ocs`	`ocs`	cgrates, ocs, pcef, keydb
`cgrates`	`cgrates`	cgrates, mysql
`omnimessage`	`omnimessage`	omnimessage, kamailio
`dns`	`dns`	named, bind
`omnicrm`	`omnicrm`	omnicrm, nginx, mysql
`monitoring`	`monitoring`	prometheus, grafana, loki, nginx

تسميات السجلات

يتضمن كل إدخال سجل هذه التسميات للتصفية والاستعلام:

التسمية	الوصف	المثال
`job`	اسم المضيف للجهاز الافتراضي المصدر	`customer-mme01`
`hostname`	نفس الشيء مثل الوظيفة (للتوافق)	`customer-mme01`
`component`	نوع المكون من مجموعة الجرد	`mme`, `cscf`, `ocs`
`unit`	اسم وحدة systemd	`omnimme.service`
`level`	شدة السجل	`info`, `warning`, `error`
`service`	معرف Syslog	`omnimme`

التكوين

النشر التلقائي

يتم تكوين التسجيل تلقائيًا عندما:

1. توجد مجموعة `monitoring` في جردك
2. يتم تشغيل الدور المشترك على المضيفين المستهدفين

لا يتطلب أي تكوين إضافي للوظائف الأساسية.

متطلبات الجرد

```
monitoring:
  hosts:
    customer-monitoring01:
      ansible_host: 10.10.2.200
      gateway: 10.10.2.1
```

عندما يتم تعريف مجموعة `monitoring`:

- Loki (يستقبل ويخزن السجلات) **مضيفو المراقبة**: تشغيل خادم
- Alloy (يجمع ويرسل السجلات) **جميع المضيفين الآخرين**: تشغيل وكيل

تكوين الاحتفاظ

يتم تكوين Loki مع حدود احتفاظ مزدوجة:

الوصف	الافتراضي	الإعداد
يتم حذف السجلات الأقدم من 7 أيام	7 أيام	الاحتفاظ القائم على الوقت
يتم حذف أقدم السجلات عندما يتجاوز التخزين 50 جيجابايت	50 جيجابايت	الاحتفاظ القائم على الحجم

أي حد يتم الوصول إليه أول�� يؤدي إلى حذف السجلات.

لتخصيص الاحتفاظ، قم بتجاوز هذه المتغيرات في جردك:

```
all:
  vars:
    loki_retention_period: "168h"  # 7 أيام بالساعات
```

تكوين ذاكرة تخزين Alloy

يقوم Alloy بتخزين السجلات محليًا عندما يكون Loki غير متاح. يتم تحديد سعة الذاكرة لتجنب استنفاد القرص:

الوصف	الافتراضي	الإعداد
الحد الأقصى لمساحة القرص للسجلات المخزنة	500 ميجابايت	الحد الأقصى لحجم WAL
يتم إسقاط أقدم السجلات المخزنة بعد 1 ساعة	1 ساعة	الحد الأقصى لعمر الجزء في WAL

عندما تمتلئ الذاكرة، يتم إسقاط أقدم السجلات لإفساح المجال لإدخالات جديدة.

لوحات معلومات Grafana

يتم توفير لوحات معلومات مسبقة البناء تلقائيًا لكل نوع من المكونات.

لوحات المعلومات المتاحة

لوحة المعلومات	الموقع	الوصف
سجلات CSCF	السجلات CSCF ← سجلات	سجلات I-CSCF و S-CSCF و P-CSCF (Kamailio)
سجلات MME	السجلات MME ← سجلات	أحداث MME للاتصال والفصل والأخطاء
سجلات SGW	السجلات SGW ← سجلات	أحداث جلسة SGW وحامل البيانات
سجلات PGW	السجلات PGW ← سجلات	أحداث PDN وجلسة PGW
سجلات HSS	السجلات HSS ← سجلات	رسائل Diameter HSS وأحداث المصادقة
سجلات OmniMessage	السجلات OmniMessage ← سجلات	SMPP وأحداث SMS تسليم
سجلات OCS/CGrateS	السجلات OCS/CGrateS ← سجلات	أحداث الشحن وحركة مرور JSONRPC
استدعاءات CGrateS RPC	السجلات استدعاءات CGrateS RPC ←	البحث والتوافق بين طلبات/استجابات RPC مع التأخير

ميزات لوحة المعلومات

تتضمن كل لوحة معلومات:

- **مربع البحث** — بحث نصي حر عبر جميع السجلات
- **رسوم بيانية لمعدل السجل** — الحجم بمرور الوقت حسب المضيف والشدة
- **أقسام المكونات** — عرض مجمع (مثل I-CSCF و S-CSCF و P-CSCF)
- **تصفية الأخطاء** — عرض مسبق التصفية للأخطاء والإخفاقات
- **التدفق المباشر** — بث السجلات في الوقت الحقيقي (تحديث كل 10 ثوانٍ)

استخدام لوحات المعلومات

1. انتقل إلى Grafana (عادةً `http://<monitoring-host>:3000`)
2. اذهب إلى **لوحات المعلومات** → **السجلات** → اختر المكون
3. استخدم متغير **البحث** لتصفية السجلات
4. اضبط نطاق الوقت حسب الحاجة

استعلام السجلات

أساسيات LogQL

يستخدم Loki LogQL للاستعلام. الصيغة الأساسية:

```
{label="value"} |= "search string"
```

الاستعلامات الشائعة

جميع السجلات من مضيف معين:

```
{hostname="customer-mme01"}
```

جميع سجلات مكون MME:

```
{component="mme"}
```

البحث عن الأخطاء عبر جميع CSCFs:

```
{component="cscf"} |~ "(?i)error"
```

تصفية حسب وحدة systemd:

```
{unit="kamailio.service"}
```

دمج الفلاتر:

```
{component="hss", hostname="customer-hss01"} |= "ULR" |=
"DIAMETER_SUCCESS"
```

استعلامات معدل السجل

حجم السجل لكل مكون:

```
sum by (component) (rate({job=~".+"} [5m]))
```

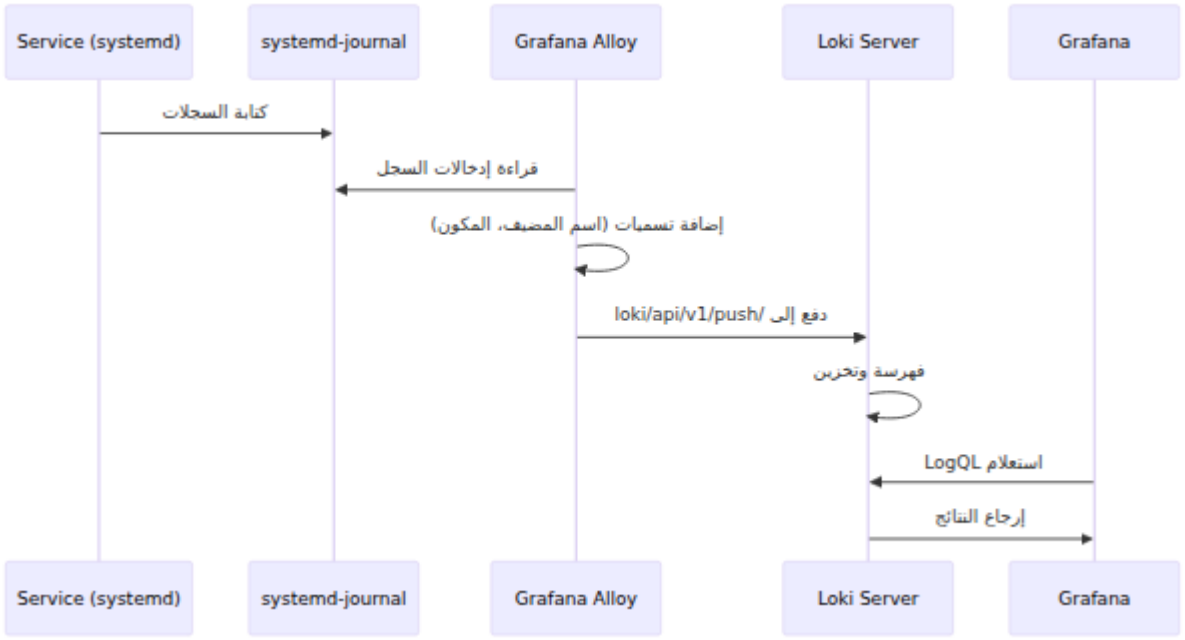
معدل الأخطاء لـ CSCF:

```
sum(rate({component="cscf"} |~ "(?i)error" [5m]))
```

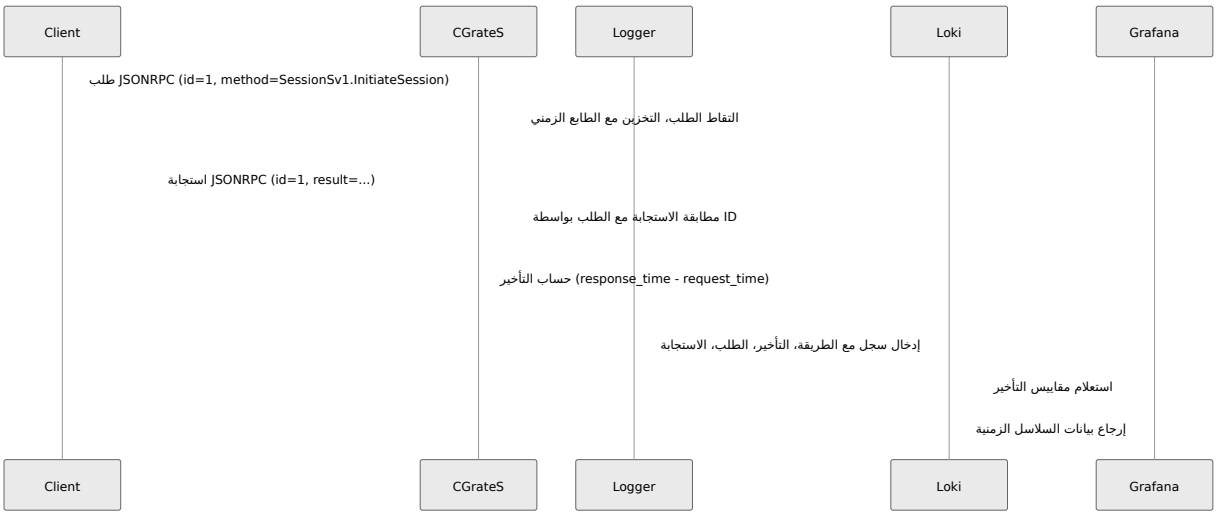
تسجيل JSONRPC لـ CGrateS

يتم التقاط حركة مرور JSONRPC /الطلب وتسجيلها مع توافق OCS/CGrateS، بالنسبة لنشر الاستجابة وتتبع التأخير لتحليل الأداء واستكشاف الأخطاء وإصلاحها.

الهيكلية



تدفق الطلب/الاستجابة



ما الذي يتم التقاطه

يتم التقاط حركة المرور على هذه المنافذ لـ CGrateS:

المنفذ	اسم الخدمة	البر��توكول	الوصف
2012	`rpc_json`	TCP	JSONRPC الخام عبر TCP
2080	`http`	HTTP	واجهة برمجة التطبيقات HTTP (تصفية `/health` و `/metrics`)

يقوم المسجل تلقائيًا بتصفية:

- عمليات سحب مقاييس Prometheus (`GET /metrics`)
- طلبات التحقق من الصحة (`GET /health`)
- الاستجابات المضغوطة باستخدام Gzip (بيانات ثنائية)

تنسيق السجل

تكتب سجلات JSONRPC إلى `/var/log/cgrates/jsonrpc.log`. يمثل كل إدخال سجل **زوج الطلب/الاستجابة المكتمل** مع قياس التأخير:

```
2026-03-01T10:30:45.123456 port=2012 service=rpc_json request_id=1
method=SessionSv1.InitiateSession latency_ms=2.45 status=ok error=
src=10.10.1.50:45678 dst=10.10.2.100:2012 request=
{"id":1,"method":"SessionSv1.InitiateSession",...} response=
{"id":1,"result":{...}}
```

حقول السجل

الحقل	الوصف	المثال
`timestamp`	الطابع الزمني ISO 8601 عند استلام الاستجابة	`2026-03-01T10:30:45.123456`
`port`	منفذ CGrateS الذي تم تقديم الطلب إليه	`2012`, `2080`
`service`	اسم الخدمة للمنفذ	`rpc_json`, `http`
`request_id`	معرف طلب JSONRPC للتوافق	`1`, `42`, (فارغ للعدم)
`method`	اسم طريقة JSONRPC	`SessionSv1.InitiateSession`
`latency_ms`	التأخير في جولة الرح��ة بالميلي ثانية	`2.45`
`status`	حالة النتيجة	`ok`, `error`
`error`	رسالة الخطأ إذا كانت الحالة خطأ	`INSUFFICIENT_CREDIT`
`src`	عنوان IP:المنفذ (العميل)	`10.10.1.50:45678`
`dst`	عنوان IP:المنفذ (CGrateS)	`10.10.2.100:2012`
`request`	الحمولة الكاملة لطلب JSONRPC (JSON مضغوط)	`{"id":1,"method":"..."}`
`response`	الحمولة الكاملة لاستجابة JSONRPC (JSON مضغوط)	`{"id":1,"result":{...}}`

لوحة معلومات Grafana

تتضمن لوحة **سجلات OCS / CGrateS** لوحات مخصصة لتحليل JSONRPC.

حسب الطريقة JSONRPC التأخير لـ

رسم بياني زمني يظهر التأخير لكل طريقة JSONRPC. مفيد لتحديد بمرور الوقت:

- الطرق البطيئة التي قد تحتاج إلى تحسين
- ارتفاعات التأخير التي تشير إلى مشاكل في النظام
- اتجاهات الأداء بمرور الوقت

يعرض الأسطورة متوسط وأقصى تأخير لكل طريقة.

حركة مرور JSONRPC

لوحة سجلات تعرض جميع أزواج الطلب/الاستجابة JSONRPC مع:

- الطابع الزمني
- اسم الطريقة
- التأخير
- الحمولة الكاملة للطلب والاستجابة (بشكل منسق)

تحليل الطريقة

تصفية حركة مرور JSONRPC حسب طريقة معينة باستخدام متغير **الطريقة** في أعلى لوحة لرؤية حركة المرور (مثل `SessionSv1.InitiateSession`) المعلومات. أدخل اسم الطريقة الخاصة بتلك الطريقة فقط.

لوحة استدعاءات CGrateS RPC

أدوات مركزة لاستكشاف أخطاء استدعاءات API تقدم لوحة **CGrateS RPC** **استدعاءات** والتوافق بين الطلبات والاستجابات.

حالة الاستخدام: العثور على استدعاء فاشل

عندما يبلغ مستخدم "حاولت الاتصال بـ 1234 وفشل":

1. افتح **CGrateS RPC** **لوحات المعلومات** ← **السجلات** ← **استدعاءات**
2. في حقل **البحث**، أدخل رقم الهاتف أو معرف الحساب: `1234`
3. اضبط **الحالة** على **خطأ** لرؤية الاستدعاءات الفاشلة فقط
4. تظهر لوحة **RPC** **بحث استدعاءات** جميع الاستدعاءات المطابقة مع الطلب/الاستجابة الكاملة

ما أعادته) يعرض كل إدخال سجل ا��حمولة الكاملة للطلب (ما تم إرساله) والحمولة الاستجابة CGrateS)، مما يسهل تحديد الخطأ الدقيق.

متغيرات لوحة المعلومات

المتغير	الغرض	المثال
البحث	بحث نصي حر في الحمولة الطلب/الاستجابة	`1234`, `Account123`, `INSUFFICIENT_CREDIT`
الطريقة	تصفية حسب اسم طريقة RPC	`SessionSv1.InitiateSession`, `CDRsV1.ProcessEvent`
الحالة	تصفية حسب حالة النتيجة	الكل، النجاح، الخطأ

لوحات المعلومات

إحصائيات ملخصة (الصف العلوي):

- **إجمالي استدعاءات RPC** — عدد جميع الاستدعاءات في نطاق الوقت
- **الأخطاء** — عدد الاستدعاءات الفاشلة (ملونة: أخضر=0، أصفر=1-9، أحمر=10+)
- **متوسط التأخير** — متوسط الوقت المستغرق (حدود ملونة)
- **أقصى تأخير** — أعلى استدعاء بتأخير

حسب الطريقة: سلسلة زمنية تظهر التأخير لكل طريقة. مرر فوق لرؤية القيم **تأخير RPC** المحددة.

بحث استدعاءات RPC: لوحة السجل الرئيسية تعرض الاستدعاءات المطابقة. انقر على أي إدخال لتوسيعه ورؤية الطلب/الاستجابة الكامل JSON.

الفاشلة: عرض مسبق التصفية يظهر فقط الاستدعاءات ذات **استدعاءات RPC** `status=error`.

تحليل الطريقة: مخططات دائرية تظهر توزيع الاستدعاءات وتوزيع الأخطاء حسب الطريقة.

استعلام سجلات JSONRPC

استعلامات أساسية

جميع حركة مرور JSONRPC:

```
{job="cgrates-jsonrpc"}
```

تصفية حسب الطريقة:

```
{job="cgrates-jsonrpc"} |= "method=SessionSv1.InitiateSession"
```

البحث في الحمولة الطلب/الاستجابة:

```
{job="cgrates-jsonrpc"} |= "Account\":\"12345"
```

الأخطاء فقط:

```
{job="cgrates-jsonrpc"} |= "status=error"
```

تحليل التأخير

استخراج التأخير كمقياس (للرسوم البيانية):

```
max_over_time(
  {job="cgrates-jsonrpc"}
  |= "method="
  | regexp "method=(?P<method>[^ ]+) latency_ms=(?P<latency>[0-
9.]+)"
  | __error__=""
  | unwrap latency [$__auto]
) by (method)
```

البحث ن الطلبات البطيئة (التأخير > 100 مللي ثانية):

```
{job="cgrates-jsonrpc"}
| regexp "latency_ms=(?P<latency>[0-9.]+)"
| latency > 100
```

استعلامات خاصة بالطريقة

معالجة CDR:

```
{job="cgrates-jsonrpc"} |= "method=CDRsV1"
```

إدارة الجلسات:

```
{job="cgrates-jsonrpc"} |~ "method=SessionSv1"
```

عمليات الحساب:

```
{job="cgrates-jsonrpc"} |~ "method=ApierV"
```

إدارة الخدمة

يعمل مسجل JSONRPC كخدمة systemd على مضيفي OCS/CGrateS.

تحقق من حالة الخدمة:

```
systemctl status cgrates-jsonrpc-logger
```

عرض سجلات الخدمة:

```
journalctl -u cgrates-jsonrpc-logger -f
```

إعادة تشغيل الخدمة:

```
systemctl restart cgrates-jsonrpc-logger
```

استكشاف الأخطاء وإصلاحها

عدم ظهور سجلات JSONRPC

تظهر عدم وجود بيانات JSONRPC **الأعراض**: لوحة حركة مرور

الأسباب المحتملة:

- خدمة المسجل غير قيد التشغيل
- عدم تثبيت ngrep
- عدم تطابق واجهة الشبكة
- لا يجمع ملف السجل Alloy

الحل:

1. تحقق من حالة خدمة المسجل:

```
systemctl status cgrates-jsonrpc-logger
journalctl -u cgrates-jsonrpc-logger -n 50
```

2. تحقق من تثبيت ngrep:

```
which ngrep
```

3. تحقق من كتابة ملف السجل:

```
tail -f /var/log/cgrates/jsonrpc.log
```

4. تحقق من أن Alloy يجمع الملف:

```
journalctl -u alloy | grep jsonrpc
```

عدم ظهور بيانات في رسم التأخير

الأعراض: لوحة التأخير لـ JSONRPC تظهر "لا توجد بيانات حسب الطريقة"

الأسباب المحتملة:

- عدم وجود حركة مرور JSONRPC في نطاق الوقت المحدد
- عدم تطابق تنسيق السجل (لا يتطابق regex)

الحل:

1. تحقق من وجود سجلات في نطاق الوقت:

```
{job="cgrates-jsonrpc"} |= "method="
```

2. تحقق من أن حقل `latency_ms` موجود في السجلات
3. قم بتوسيع نطاق الوقت

عدد الطلبات المعلقة مرتفع

الأعراض: تظهر سجلات التصحيح العديد من الطلبات المخزنة ولكن القليل من الإكمال المسجل

الأسباب المحتملة:

- الطلبات بدون استجابات (م قطع اتصال العميل)
- تم التقاط الاستجابة قبل الطلب (ترتيب الحزم)
- عدم تطابق معرف الطلب بين الطلب والاستجابة

الحل:

1. يقوم المسجل تلقائيًا بتنظيف الطلبات المعلقة التي تزيد عن 60 ثانية
2. تحقق من وجود مشاكل في الشبكة بين العميل و CGrateS
3. تحقق من أن CGrateS تستجيب للطلبات

سجلات الملفات

بالإضافة إلى سجلات نظام journal، يقوم Alloy بجمع سجلات ملفات محددة:

سجلات FreeSWITCH (خوادم التطبيقات)

تسمية الوظيفة	المسار
`freeswitch`	`/var/log/freeswitch/*.log`

سجلات Nginx (OmniCRM)

تسمية النوع	تسمية الوظيفة	المسار
`access`	`nginx`	`/var/log/nginx/access.log`
`error`	`nginx`	`/var/log/nginx/error.log`

JSONRPC لـ CGrateS (OCS)

تسمية الوظيفة	المسار
`cgrates-jsonrpc`	`/var/log/cgrates/jsonrpc.log`

استكشاف الأخطاء وإصلاحها

عدم ظهور السجلات في Grafana

الأ��راض: لا توجد سجلات مرئية في لوحات معلومات Loki

الأسباب المحتملة:

- خدمة Alloy غير قيد التشغيل على المضيف المصدر
- خدمة Loki غير قيد التشغيل على مضيف المراقبة
- حظر الاتصال الشبكي بين المضيفين
- لا يمكن لـ Alloy الوصول إلى Loki 3100 على المنفذ

الحل:

1. تحقق من حالة Alloy على المضيف المصدر:

```
systemctl status alloy
journalctl -u alloy -f
```

2. تحقق من حالة Loki على مضيف المراقبة:

```
systemctl status loki
journalctl -u loki -f
```

3. اختبار الاتصال من المصدر إلى المراقبة:

```
curl http://<monitoring-ip>:3100/ready
```

4. تحقق من أن جدار الحماية يسمح بـ TCP 3100 من جميع المضيفين إلى المراقبة

استخدام Alloy العالي للقرص

الأعراض: تستهلك مساحة قرص كبيرة `/var/lib/alloy`

الأسباب المحتملة:

- عدم إمكانية الوصول إلى Loki لفترة طويلة
- امتلاء ذاكرة WAL

الحل:

1. تحقق من الاتصال بـ Loki (انظر أعلاه)
2. إذا كان Loki غير متاح، يتم تخزين السجلات محليًا حتى 500 ميجابايت
3. بمجرد أن يصبح Loki متاحًا، يتم إرسال السجلات المخزنة تلقائيًا
4. إذا كانت الذاكرة ممتلئة، يتم إسقاط أقدم السجلات (حسب التصميم)

امتلاء تخزين Loki

الأعراض: يتوقف Loki عن قبول السجلات، القرص ممتلئ على خادم المراقبة

الأسباب المحتملة:

- حجم السجل يتجاوز حد الاحتفاظ 50 جيجابايت
- عدم تشغيل ضغط الاحتفاظ

الحل:

1. تحقق من استخدام تخزين Loki:

```
du -sh /var/lib/loki/
```

2. تحقق من أن الضاغط يعمل (تحقق من سجلات Loki)
3. قم بتشغيل الضغط يدويًا إذا لزم الأمر عن طريق إعادة تشغيل Loki
4. ضع في اعتبارك زيادة تخصيص القرص أو تقليل فترة الاحتفاظ

عدم وجود تسميات مكون

الأعراض: تظهر السجلات ولكن التسمية `component` هي `infrastructure` بدلاً من القيمة المتوقعة

الأسباب المحتملة:

- المضيف ليس في مجموعة الجرد المتوقعة
- بعد تغيير الجرد Alloy لم يتم تج��يد تكوين

الحل:

1. تحقق من أن المضيف في مجموعة الجرد الصحيحة
2. أعد تشغيل Ansible لتجديد تكوين Alloy:

```
ansible-playbook -i hosts/customer/hosts.yml
services/all.yml --limit <hostname>
```

3. أعد تشغيل Alloy:

```
systemctl restart alloy
```

منافذ الخدمة

الخدمة	المنفذ	البروتوكول	الوصف
Loki	3100	HTTP	واجهة إدخال السجل واستعلام API
Loki	9096	gRPC	gRPC داخلي (غير مستخدم خارجيًا)
Alloy	12345	HTTP	واجهة المستخدم Alloy ومقاييس
Grafana	3000	HTTP	الوصول إلى لوحة المعلومات

الوثائق ذات الصلة

- المراقبة والملاحظة — Grafana و Prometheus ولوحات المعلومات والتنبيهات
- هيكلية النشر — الهيكلية العامة للنظام
- تكوين ملف المضيفين — تكوين الجرد
- مرجع التكوين — مرجع كامل للمعلمات

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 للنشر الحديث.

للحصول على الوثائق الخاصة بالمنتج، 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 نموذج تكوين مبسط الحديثة:

المبدأ الأساسي: يتم تعريف مع��م التكوين مباشرة في ملف المضيفين. تتعامل إعدادات group_vars الدور الافتراضية مع معظم الإعدادات، مع استخدام 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 العامة

معلمات المضيف الشائعة

للحصول على هذا hosts-file-configuration.md فقط قل للتحقق من - #ToDo

علامات الخدمة المحددة

```
cdrs_enabled: True                # تمكين إنشاء 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 ق��يلة مع معظم التكوين في إعدادات الدور الافتراضية.

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
```

إعدادات PLMN

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

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 الشبكات

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

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

إعدادات DNS

```
netplan_DNS: False                # استخدام systemd-resolved بدلاً من
netplan DNS
manage_resolv_conf: True          # تعيين إلى False لمنع Ansible من
إدارة /etc/resolv.conf
```

ملاحظة: عندما يتم تعيين `manage_resolv_conf` إلى `False`، لن يقوم Ansible بكتابة `/etc/resolv.conf` على ذلك المضيف. هذا مفيد للمضيفين الذين يحتاجون إلى تكوين DNS

مخصص أو يتم إدارتهم بواسطة أنظمة خارجية. يمكن تعيينه لكل مضيف في الجرد أو عالميًا في `all:vars`.

إعدادات مستودع APT

مع المضيفين `apt_cache_servers` عند تعريف مجموعة **الإعدادات الافتراضية التلقائية:**

- يتم تعيين `use_apt_cache` تلقائيًا إلى `True` (إلى `False` ما لم يتم تعيينه صراحة)
- يتم تعيين `apt_repo.apt_server` تلقائيًا إلى عنوان IP الخاص بأول خادم ذاكرة التخزين المؤقت

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

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

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

انظر: نظام ذاكرة التخزين المؤقت 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 من قبل الناقل (استنادًا إلى PLMN)
  123456:                          # مثال الناقل 1
    - '10.10.2.197'
  654321:                          # مثال الناقل 2
    - '10.10.0.4'
```

المستخدمون المحليون (مفاتيح SSH)

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

تكوين Hypervisor

Proxmox

يتم نشر الموقع كـ VMs أو حاويات LXC، يتم اختيارها لكل إدخال `proxmoxServers` عبر `deployment_type` (الافتراضي هو `vm`). يجب أن تتفق جميع الإدخالات في الموقع؛ الخلط يفشل في التحقق. انظر نشر Proxmox للحصول على الإرشادات الكاملة.

موقع VM

```
proxmoxServers:
  customer-prxmx01:
    # تم حذف deployment_type → هو الافتراضي "vm"
    proxmoxServerAddress: 10.10.0.100
    proxmoxServerPort: 8006
    proxmoxApiTokenName: AnsibleToken
    proxmoxApiTokenSecret: "token-secret"
    proxmoxNodeName: pve01
    proxmoxTemplateName: ubuntu-24.04-cloud-init-template
    proxmoxTemplateId: 9000
    proxmoxTemplateUser: omnitouch        # اسم مستخدم cloud-init اختياري؛ الافتراضي هو أول مفتاح local_users
    proxmoxTemplatePassword: omnitouch    # كلمة مرور cloud-init اختيارية؛ الافتراضي هو اسم مستخدم cloud-init
    storage: SSD_RAID0            # تخزين VM اختياري، الافتراضي
```

موقع LXC

```
proxmox_lxc_nameserver: "1.1.1.1"  # عند LXCs اختياري، يتم حقنه في الإنشاء

proxmoxServers:
  customer-prxmx01:
    deployment_type: lxc
    proxmoxServerAddress: 10.10.0.100
    proxmoxServerPort: 8006
    proxmoxApiTokenName: AnsibleToken
    proxmoxApiTokenSecret: "token-secret"
    proxmoxNodeName: pve01
    proxmoxLxcOsTemplate: "local:vztmpl/ubuntu-24.04-standard_24.04-2_amd64.tar.zst"
    proxmoxLxcDefaultStorage: SSD_RAID0   # اختياري، النسخة الاحتياطية لجذر النظام
```

تجاوزات على مستوى المجموعة (كلا النوعين)

```
dns:
  vars:
    proxmox_interface: vmbr0              # مطلوب: الجسر
    gateway: 10.10.0.1                    # مطلوب
    netmask: 255.255.255.0                # مطلوب
    vlanid: 100                           # اختياري
    proxmoxLxcCores: 2                    # اختياري (فقط LXC)
    proxmoxLxcMemoryMb: 4096              # اختياري (فقط LXC)
    proxmoxLxcDiskSizeGb: 30              # اختياري (فقط LXC)
    proxmoxLxcRootFsStorageName: SSD_RAID0 # اختياري (فقط LXC)
    host_vm_network: vmbr1                # تجاوز الجسر الاختياري
```

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 هيكل الشبكة وتخصيص IP
- تكوين ملف المضيفين - كيفية هيكلة ملفات المضيفين
- تكوين المتغي��ات الجماعية group_vars - متى وكيفية استخدام
- تكوين Netplan - عناوين IP متعدد وإعدادات NIC الثانوية
- هندسة النشر - كيفية تفاعل المكونات
- نظام ذاكرة التخزين المؤقت 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 الخاص بخدمات Omnitouch موضحًا كيف تتناسب جميع المكونات معًا لإنشاء شبكة 4G/5G تعمل باستخدام Ansible.

للحصول على إرشادات تفصيلية حول وضع المكونات، وإرشادات تخصيص IP راجع معيار تخطيط عنوان IP، والتعامل مع IP العامة.

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
util_playbooks/proxmox.yml

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

راجع: نشر VM/LXC على Proxmox

1. تعريف 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 الخاص بك، وستكون خطوط SIP التي يتصل بها TAS الخاص بك، وكل منطق توجيه Diameter الخاص بك هنا، 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 المباشر 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`، وما إلى ذلك، 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
- دفاتر الخدمة - **مرجع دفاتر التشغيل والتسلسل الهرمي**
- تكوين ملف المضيف - تعريف البنية التحتية
- معيار تخطيط IP - **بنية الشبكة وتخصيص IP**
- التخصيص - تكوين المتغيرات الجماعية
- إدارة الحزم - نظام APT Cache
- إدارة الترخيص - خادم الترخيص
- المراقبة والرصد - Grafana، Prometheus، ولوحات المعلومات، التنبيهات
- جمع سجلات - تسجيل مركزي Loki وAlloy

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.

خطوط - Diameter، قواعد توجيه SIP، هذا هو المكان الذي تعيش فيه تكوينات العملاء المحددة 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

تمتلك أدوار Ansible قوالب تكوين افتراضية. لتخصيصها لنشر محدد، ضع ملفاتك المخصصة في `group_vars` واستشهد بها في ملف المضيفين الخاص بك.

```
التكوين ← (إذا تم تحديده) group_vars ← 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 تكوينك المخصص
```

الإشارة 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    # الإشارة 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 بقالبته 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
(افتراضي)
    │   ├── mo_dialplan.xml         # (صادر) منشأ الهاتف المحمول
    │   ├── mt_dialplan.xml         # (وارد) منتهي الهاتف المحمول
    │   └── emergency.xml
    └── dialplan_lab/               # خطط الاتصال في المختبر
        └── mo_dialplan.xml
```

الإشارة 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"   # الإشارة إلى مجلد البوابة الخاص بك لاستخدامه على هذا المضيف
      dialplan_folder: "dialplan"         # اختياري - الافتراضي هو "dialplan" إذا لم يتم تعيينه
```

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/` (أو المجلد المحدد بواسطة `dialplan_folder`) إلى دليل قوالب 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"`
- `dialplan_folder: "dialplan_lab"`
- `dialplan_folder: "dialplan_prod"`

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
```

الإشارة 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   # الإشارة 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 بقالبته 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 المخصص الخاص بك
```

الإشارة 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   # الإشارة 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 بقالبته 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 SMPP مخصص لـ OmniMessage
    ├── 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     # منتهي الهاتف المحمول
        └── mo_emergency.xml    # توجيه الطوارئ
```

المعلمات الشائعة التي تشير إلى

group_vars

المعلمة	المكون	الإشارات
`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`)
`dialplan_folder`	applicationserver	اسم المجلد (مثل `dialplan`) - الافتراضي هو `dialplan` إذا لم يتم تعيينه
`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`)

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

النقاط الرئيسية

1. **تحتوي group_vars على تخصيصات** - تجاوزات للتكوينات الافتراضية
2. **الإشارة 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 المشار إليها
5. **تحكم في الإصدار لكل شيء** - التزام بجميع group_vars إلى Git

متى تستخدم group_vars

✅ استخدم group_vars لـ:

- قوالب تكوين مخصصة للمكونات
- تعريفات بوابة SIP
- خطط توجيه المكالمات
- قواعد توجيه Diameter
- إعدادات محددة للعملاء التي تتجاوز الافتراضات

❌ لا تستخدم group_vars لـ:

- استخدم ملف المضيفين - (أسماء المضيفين، IP عناوين) تكوين المضيفين الأساسي
- الاختبار لمرة واحدة - استخدم متغيرات محددة للمضيفين في ملف المضيفين
- إذا كانت دائمة group_vars التغييرات المؤقتة - قم بالتعديل على الهدف، والتزم بـ

الوثائق ذات الصلة

- مرجع التكوين - جميع معلمات المضيفين وما تقوم به
- 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
- أين تعمل (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 المضيف الشائعة

تكوين الشبكة

كل مضيف يتضمن عادةً:

```
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 لاستخدامه
```

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

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

تخصيص موارد VM

بالنسبة للخدمات التي تحتاج إلى موارد محددة:

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

معلمات محددة للخدمة

كل وظيفة شبكة لها معلماتها الخاصة. أمثلة:

MME:

```
mme_code: 1                      # معرف MME (1-255)
mme_gid: 1                       # معرف مجموعة MME
network_name_short: Customer     # اسم الشبكة (يظهر 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
```

للحصول 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
```

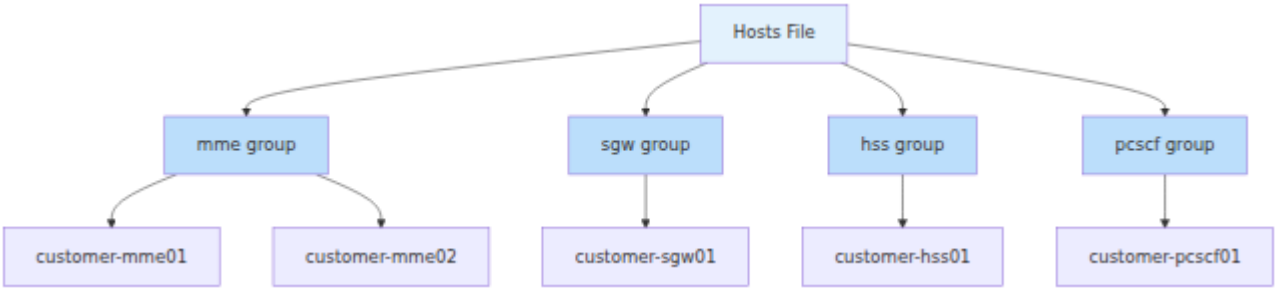
قسم المتغيرات العالمية

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 مع المضيفين،
    # فإن use_apt_cache و افتراضيًا يكون صحيحًا apt_repo.apt_server
    # لأحد خوادم التخزين المؤقت IP افتراضيًا يكون 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:

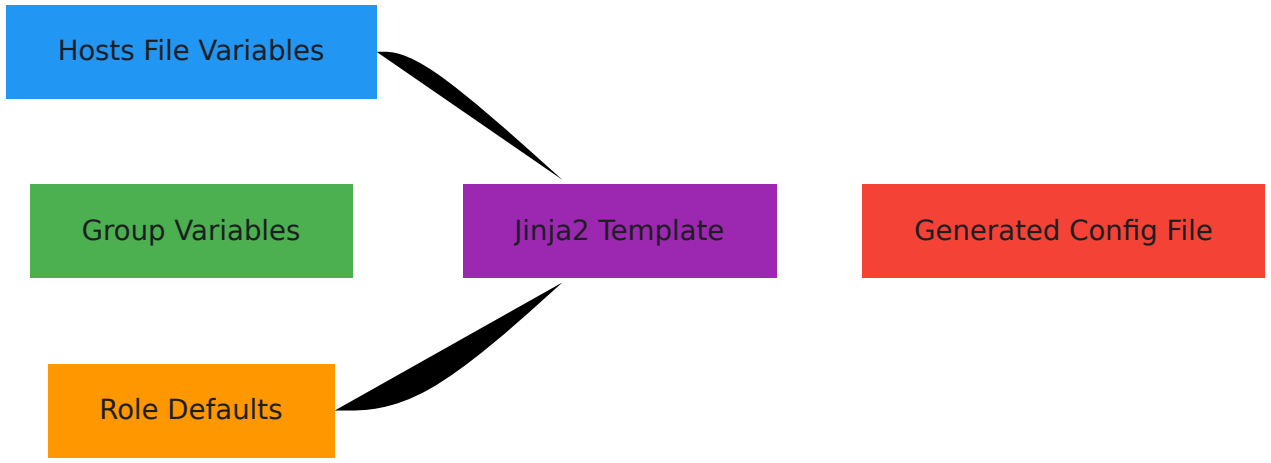


فإنه ينطبق على جميع المضيفين في قسم ، عند تشغيل كتاب اللعب الذي يستهدف `mme` `mme:hosts:`.

التكوين باستخدام قوالب Jinja2

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

كيفية عمل Jinja2



مثال على استخدام القالب

ملف المضيفين يحدد:

```
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) }}
```

تجاوز المتغيرات باستخدام group_vars

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

انظر: تكوين المتغيرات الجماعية

مثال كامل على ملف المضيفين

إليك مثال أكثر اكتمالاً (مع إخفاء البيانات الحساسة):

```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 VM)
    proxmoxServers:
      customer-prxmx01:
        # الافتراضي هو ← تم حذف نوع النشر "vm"
        proxmoxServerAddress: 10.10.0.100
        proxmoxServerPort: 8006
        proxmoxApiTokenName: Customer
        proxmoxApiTokenSecret: "token-secret"
        proxmoxNodeName: pve01
        proxmoxTemplateName: ubuntu-24.04-cloud-init-template
        proxmoxTemplateId: 9000
        proxmoxTemplateUser: omnitouch        # اسم المستخدم الاختياري
cloud-initالافتراضي هو أول مفتاح ؛ local_users
        proxmoxTemplatePassword: omnitouch    # كلمة المرور الاختيارية
cloud-initالافتراضي هو اسم المستخدم ؛ cloud-init

    # لموقع LXC، قم بتعيين deployment_type: lxc و
    # proxmoxLxcOsTemplate في كل إدخال proxmoxServers بدلاً من ذلك.
```

انظر نشر Proxmox VM/LXC للحصول على تفاصيل إعداد وتكوين Proxmox بالكامل.

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 جلسة المكالمات
- **OmniMessage** - مركز 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 المتغيرات الجماعية - تجاوز التكوينات الافتراضية

- معيار تخطيط IP - **إرشادات بنية الشبكة وتخصيص IP**
- تكوين Netplan - **عناوين IP المتقدم وتكوين الشبكة الثانوية**
- نظام تخزين APT - توزيع الحزم
- خادم الترخيص - إدارة الترخيص
- 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 اللعب Ansible

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

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

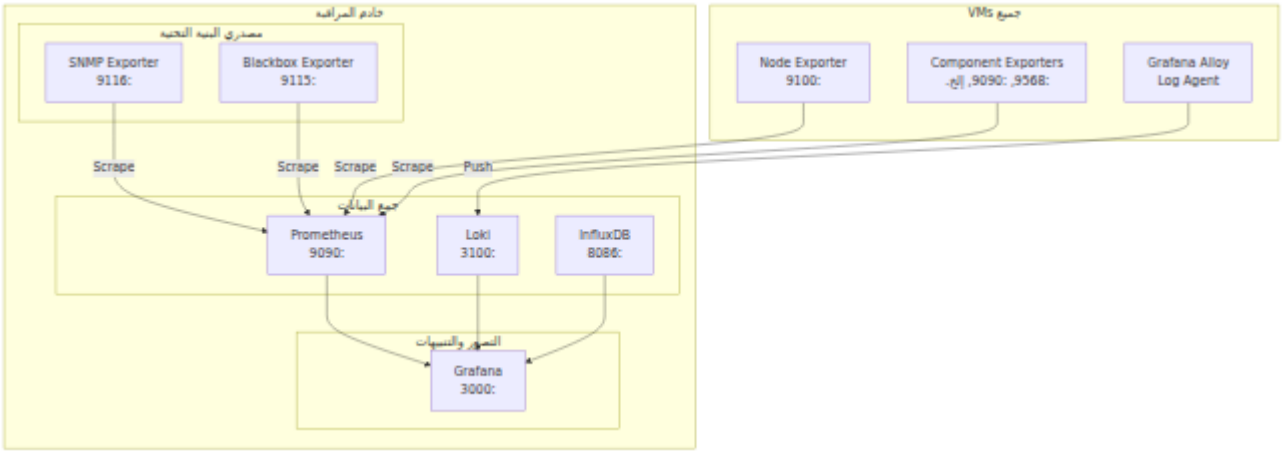
الوثائق ذات الصلة

- مرجع التكوين
- تكوين ملف المضيفين

المراقبة والملاحظة

نظرة عامة

مجموعة شاملة من أدوات المراقبة والملاحظة التي توفر جمع المقاييس، OmniCore يتضمن وتجميع السجلات، والتصور، والتنبيهات. يتم نشر النظام تلقائيًا بواسطة دور المراقبة.



المكونات

المكون	الغرض	المنفذ	الاحتفاظ بالبيانات
Prometheus	تخزين مقاييس السلاسل الزمنية	9090	15 يومًا (افتراضي)
Loki	تجميع السجلات	3100	7 أيام أو 50 جيجابايت
InfluxDB	مقاييس RAN (نوكيا)	8086	30 يومًا
Grafana	التصور والتنبيهات	3000	غير متاح
Node Exporter	مقاييس المضيف	9100	غير متاح
SNMP Exporter	مقاييس أجهزة الشبكة	9116	غير متاح
Blackbox Exporter	اختبارات ICMP/HTTP	9115	غير متاح

مصادر البيانات

Prometheus

يقوم Prometheus بجمع المقاييس من جميع مكونات OmniCore كل دقيقة واحدة.

معرف مصدر البيانات: `omnicore_prometheus`

الأهداف المجمعة

اسم الوظيفة	مجمو��ة الجرد	المنفذ	المقاييس
`Node Exporter`	`all`	9100	وحدة المعالجة المركزية للمضيف، الذاكرة، القرص، الشبكة
`MMEs`	`mme`	9568	عدد الجلسات، معدلات الإرفاق/الفصل
`HSS`	`hss`	9568	طلبات المصادقة، عمليات البحث عن المشتركين
`SGW-C`	`sgw`	9568	عدد الحاملات، رسائل GTP-C
`PGW-C`	`pgwc`	9090	اتصالات PDN، تخصيصات IP
`UPF Standalone`	`upf`	9090	عدد الحزم، معدل النقل
`Kamailio CSCF`	`pcscf`, `scscf`, `icscf`	9090	التسجيلات، المعاملات
`ApplicationServer FreeSWITCH`	`applicationserver`	9090	عدد المكالمات، استخدام القناة
`DRA`	`dra`	9568	قرارات التوجيه، حالة النظراء
`OmniMessage`	`omnimessage`	9568	تسليم الرسائل القصيرة، جلسات SMPP

المقاييس	المنفذ	مجمو��ة الجرد	اسم الوظيفة
مقاييس بوابة SS7/SIGTRAN	8080	`omniss7`	`OmniSS7`
استخدام الذاكرة، العمليات/ثانية	9121	`ocs`	`KeyDB`
التصنيف، الشحن، إحصائيات CDR	2080	`ocs`	`CGrateS`

مصدري البنية التحتية

متغير التكوين	الأهداف	الم��در
`mikrotik.hosts`	أجهزة التوجيه/المفاتيح	SNMP (MikroTik)
`idrac.hosts`	خوادم Dell	SNMP (iDRAC)
`synology.hosts`	أجهزة NAS	SNMP (Synology)
`saf.hosts`	روابط الميكروويف	SNMP (SAF)
`cisco.hosts`	المفاتيح	SNMP (Cisco)
تلقائي	جميع المضيفين + 8.8.8.8	Blackbox ICMP
`vcenter_ip`, `vcenter_password`	مجموعات vCenter	VMware
`proxmoxServers`	مجموعات PVE	Proxmox

Loki

يتلقى Loki السجلات من وكلاء Grafana Alloy الذين يعملون على جميع VMs.

معرف مصدر البيانات: `omnicore_loki`

انظر Loki/Alloy بالتفصيل للحصول على تكوين التسجيل المركزي.

تسميات السجلات

التسمية	الوصف	المثال
`hostname`	اسم مضيف VM المصدر	`customer-mme01`
`component`	نوع المكون	`mme`, `cscf`, `ocs`
`unit`	اسم وحدة systemd	`omnimme.service`
`level`	شدة السجل	`info`, `error`

InfluxDB

تخزن InfluxDB بيانات السلاسل الزمنية لحالات الاستخدام المحددة التي تتطلب احتفاظًا أطول أو أنماط استعلام مختلفة.

معرف مصدر البيانات: `omnicore_influxdb`

قواعد البيانات

قاعدة البيانات	الغرض	الاحتفاظ
`nokia-monitor`	مقاييس أداء RAN من نوكيا	30 يومًا
`dra`	إحصائيات توجيه DRA	365 يومًا
`Omnicharge_TAP3`	مقاييس ملف TAP للتجوال	90 يومًا

Grafana لوحات معلومات

تنظيم لوحات المعلومات

تُنظم لوحات المعلومات في مجلدات حسب المنطقة الوظيفية:

المجلد	المحتويات
BSS	CGrateS، KeyDB، لوحات معلومات الشحن
EPC	لوحات معلومات MME، SGW، PGW، UPF، HSS، DRA
IMS	لوحات معلومات CSCF، خادم التطبيقات، OmniMessage
Infrastructure	Node Exporter، SNMP، مراقبة الشبكة
RAN	لوحات معلومات أداء Nokia/OmniRAN
Logs	عارضات السجلات الخاصة بالمكونات

تحميل لوحات المعلومات (استنادًا إلى API)

يتم تحميل لوحات المعلومات عبر واجهة برمجة تطبيقات Grafana. بدلاً من توفيرها عبر الملفات. وهذا يسمح بأن تكون لوحات المعلومات:

- قابلة للتعديل من خلال واجهة مستخدم Grafana
- قابلة للتعديل عبر API
- خاضعة للتحكم في الإصدار في مستودع Ansible

��حميل لوحات المعلومات

يتم تحميل لوحات المعلومات تلقائيًا أثناء نشر Ansible. لإعادة التحميل يدويًا:

```
# من وحدة التحكم Ansible
python3 roles/monitoring/files/load_grafana_dashboards.py \
    --url http://<monitoring-ip>:3000 \
    --user admin \
    --password <grafana_admin_password> \
    --dashboards-dir roles/monitoring/templates/grafana/dashboards

# تحديث اللوحات الموجودة بالقوة
python3 roles/monitoring/files/load_grafana_dashboards.py \
    --url http://<monitoring-ip>:3000 \
    --user admin \
    --password <grafana_admin_password> \
    --dashboards-dir roles/monitoring/templates/grafana/dashboards
\
    --force
```

ملفات مصدر لوحة المعلومات

تُخزن ملفات JSON الخاصة بلوحات المعلومات في مستودع Ansible:

```
roles/monitoring/templates/grafana/dashboards/
├── BSS/
│   ├── KeyDB_Cluster.json
│   ├── cgrates_mysql.json
│   └── cgrates_stats.json
├── EPC/
│   ├── MME_Dashboard.json
│   ├── OmniHSS.json
│   ├── OmniDRA.json
│   ├── SGW.json
│   ├── PGWs.json
│   └── ...
├── IMS/
│   ├── SMSc.json
│   ├── MMSc.json
│   └── ...
├── Infrastructure/
│   ├── Node_Exporter_Full.json
│   ├── MikroTik_Dashboard.json
│   └── ...
├── RAN/
│   ├── Nokia_Overview.json
│   ├── Nokia_Detailed.json
│   └── ...
└── Logs/
    ├── CSCF_Logs.json
    ├── MME_Logs.json
    └── ...
```

النسخ الاحتياطي للوحا

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، أو اتصالات التبادل العامة
- 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

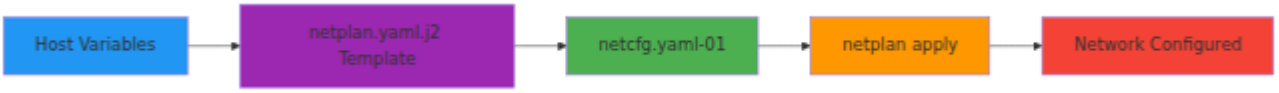
عند استخدام playbook `proxmox.yml`، يتم إنشاء NICs على الثانوية تلقائيًا VM:

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 وملفات التكوين الأخرى.

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. يقوم كتاب التشغيل الواحد — `services/proxmox.yml` — بتوفير كل من VMs QEMU وحاويات LXC بناءً على تبديل `deployment_type` على مستوى الموقع.

نستمر أيضًا في دعم VMware وHyperV والسحابة (Vultr / AWS / GCP) للنشر.

انظر أيضًا:

- تعريف VMs/LXCs للنشر — تكوين ملف المضيفين
- معيار تخطيط IP
- تكوين Netplan
- معمارية النشر

VM مقابل LXC

حاويات LXC:

- خفيفة الوزن، تشارك نواة المضيف
- بدء سريع، تكلفة منخفضة
- عزل محدود
- لا يمكن تشغيل نوى مخصصة أو وحدات نواة
- **غير مناسبة لنشر EPC/IMS في الإنتاج**
- **لا يمكن تشغيل UPF** (يتطلب وحدات نواة / أجهزة TUN)

VMs (KVM):

- افتراضية كاملة مع نواة مخصصة
- عزل كامل، تشغيل وحدات نواة / شبكات مخصصة
- تكلفة موارد أعلى
- **موصى بها للإنتاج**

- **مطلوبة لنشر UPF**

حالات الاستخدام:

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

موقع واحد = نوع واحد

يجب أن تتفق جميع إدخالات `proxmoxServers.*.deployment_type` في الموقع **معًا**. لا يُدعم خلط LXC وVM في نفس ملف المضيفين ويفشل في التحقق. اختر واحدًا لكل موقع.

إعداد Proxmox

1. إنشاء رمز API

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

2a. إنشاء قالب VM Cloud-Init (مواقع VM فقط)

يقوم بتنزيل صورة سحابة Ubuntu قم بتشغيل هذا البرنامج النصي على مضيف Proxmox. في وقت النسخ SSH ومفاتيح cloud-init وينشئ قالبًا نظيفًا — يتم حقن ��يانات اعتماد بواسطة `proxmox.yml` (انظر الملاحظات أدناه).

```
#!/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"

cd /var/lib/vz/template/iso
wget -N "$IMAGE_URL"

qm destroy $TEMPLATE_ID --purge 2>/dev/null || true

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 template $TEMPLATE_ID
```

ملاحظات:

- `proxmox.yml` القالب ليس به مستخدمون أو كلمات مرور محددة مسبقًا — يقوم بتعيين `ciuser` و`cipassword` و`sshkeys` في وقت النسخ.
- أولوية مستخدم cloud-init: `proxmoxTemplateUser` (على إدخال `proxmoxServers` المطابق) → أول مفتاح `local_users`.
- أولوية كلمة مرور cloud-init: `proxmoxTemplatePassword` (على إدخال `proxmoxServers` المطابق) → مستخدم cloud-init (أي القيمة المختارة أعلاه).
- يتم حقن مفاتيح SSH من كل إدخال `local_users.*.public_key` (المصادقة المعتمدة على المفتاح هي مسار تسجيل الدخول المقصود).
- قم بتعيين `proxmoxTemplateUser` (ويمكن اختيارياً `proxmoxTemplatePassword`) على كل إدخال `proxmoxServers` كلما تم إعداد القالب بمستخدم cloud-init ليس هو أول إدخال `local_users`، بحيث يتم اختيار الحساب الصحيح.
- `--vga std` يضمن أن وحدة التحكم على الويب في Proxmox تعمل.

- علامة `-N` على wget تقوم بالتنزيل فقط إذا كانت أحدث من النسخة المحلية.

بديل: قالب يدوي من ISO

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

الخطوة 1: إنشاء VM عبر واجهة الويب

- إنشاء VM → 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: 32GB، الناقل: SCSI
- وحدة المعالجة المركزية: 2 نواة
- الذاكرة: 2048 ميجابايت
- الشبكة: VirtIO، الجسر vmbr0
- بدء VM وتثبيت خادم Ubuntu

الخطوة 2: داخل VM - تنظيف وتحضير

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

# تنظيف بيانات الجهاز المحددة
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 وتحويله إلى قالب

- اختر VM → محرك → إضافة → الأجهزة CloudInit (اختر التخزين مثل local-lvm)
- وكيل → الخيارات → الأجهزة QEMU → تمكين
- تحويل إلى قالب → انقر بزر الماوس الأيمن على VM

- لا تقم بتعيين مستخدم / كلمة مرور cloud-init يقوم — على القالب `proxmox.yml` بحقن هذه في وقت النسخ من `proxmoxTemplateUser` / `proxmoxTemplatePassword` (مع استخدام `local_users` كخيار احتياطي لاسم المستخدم).

2b. تنزيل قالب نظام تشغيل LXC (مواقع LXC فقط)

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

مسار الحجم الناتج (مثل `local:vztmpl/ubuntu-24.04-standard_24.04-2_amd64.tar.zst`) هو ما تقوم بتكوينه كـ `proxmoxLxcOsTemplate` أدناه.

تكوين ملف المضيفين

نشر VM

```
all:
  vars:
    # تم حذف deployment_type → هو الافتراضي "vm"
    proxmoxServers:
      pve-node-01:
        proxmoxServerAddress: 192.168.1.100
        proxmoxServerPort: 8006
        proxmoxApiTokenName: ansible
        proxmoxApiTokenSecret: "your-token-secret-uuid"
        proxmoxNodeName: pve-node-01
        proxmoxTemplateName: ubuntu-2404-template
        proxmoxTemplateId: 9000
        proxmoxTemplateUser: omnitouch        # اسم مستخدم cloud-init اختياري؛ الافتراضي هو أول مفتاح local_users
        proxmoxTemplatePassword: omnitouch    # كلمة مرور cloud-init اختياري؛ الافتراضي هو اسم مستخدم cloud-init
        storage: local-lvm  # اختياري
      pve-node-02:
        # ... تكوين العقدة الثانية

    local_users:
      admin_user:
        name: Admin User
        public_key: "ssh-rsa AAAA..."

mme:
  hosts:
    site-mme01:
      ansible_host: 192.168.1.10
  vars:
    proxmox_interface: vmbr0
    gateway: 192.168.1.1
    netmask: 255.255.255.0
    vlanid: 100  # اختياري
```

نشر LXC

```yaml
all:
  vars:
    proxmox_lxc_nameserver: "1.1.1.1"   # 1.1.1.1 اختياري، الافتراضي
    proxmoxServers:
      pve-node-01:
        deployment_type: lxc
        proxmoxServerAddress: 192.168.1.100
        proxmoxServerPort: 8006
        proxmoxApiTokenName: ansible
        proxmoxApiTokenSecret: "your-token-secret-uuid"
        proxmoxNodeName: pve-node-01
        proxmoxLxcOsTemplate: "local:vztmpl/ubuntu-24.04-
standard_24.04-2_amd64.tar.zst"
        proxmoxLxcDefaultStorage: local-lvm  # احتياطي، اختياري rootfs
      pve-node-02:
        deployment_type: lxc
        # ... نفس الشكل

    local_users:
      admin_user:
        name: Admin User
        public_key: "ssh-rsa AAAA..."

dns:
  hosts:
    site-dns01:
      ansible_host: 192.168.1.20
  vars:
    proxmox_interface: vmbr0
    gateway: 192.168.1.1
    netmask: 255.255.255.0
    vlanid: 100                          # اختياري
    proxmoxLxcCores: 2                   # تجاوز اختياري
    proxmoxLxcMemoryMb: 4096             # تجاوز اختياري
    proxmoxLxcDiskSizeGb: 30             # تجاوز اختياري
    proxmoxLxcRootFsStorageName: local-lvm  # اختياري، يتجاوز التخزين
الافتراضي proxmoxLxcDefaultStorage
```

الاستخدام

توفير (VM أو LXC)

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

حذف

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

يتعامل كتاب التشغيل للحذف مع كل من VMsوLXCs تلقائيًا، بغض النظر عن `deployment_type`.

سلوك كتاب التشغيل

حراس النوع

- إذا كان هناك VM باسم المضيف المستهدف موجود ولكن `deployment_type: lxc` → يفشل كتاب التشغيل مع خطأ واضح.
- الحالة العكسية (LXC موجود، الموقع مُكون كـ VM) → نفس الشيء.
- مجموعة UPF على موقع LXC → تحذير فقط (لن يعمل UPF ليس فشلًا حادًا، لكن).

مورد جديد

- يقوم بتوزيع الموارد عبر إدخالات `proxmoxServers` حسب فهرس المضيف داخل المجموعة.
- يستعلم عن واجهة برمجة تطبيقات Proxmox للحصول على VMID التالي المتاح.
- **VM**: يتم نسخه من `proxmoxTemplateId`، ويقوم بتعيين مستخدم cloud-init / من `proxmoxTemplateUser` (مستخدم cloud-init = كلمة المرور / مفاتيح SSH ؛ كلمة مرور `local_users` إدخال الخادم إذا تم تعيينه، وإلا أول مفتاح cloud-init = `proxmoxTemplatePassword` إذا تم تعيينها، وإلا اسم مستخدم cloud-init)، يقوم بتغيير حجم `scsi0`، ويبدأ.

- **LXC**: يتم إنشاؤه من `proxmoxLxcOsTemplate`، rootfs على `proxmoxLxcRootFsStorageName` (المجموعة) ← `proxmoxLxcDefaultStorage` (الخادم) ← `storage` ← `local-lvm` (احتياطي نهائي)، ويحقن جميع `local_users.*.public_key` عبر `ssh-public-keys`، ويبدأ. كلمة مرور الجذر محددة مسبقًا إلى `mn1Touch@`. جميع LXCs لها امتيازات `0` (`unprivileged: 0`) مع تمكين التعشيش.

مورد موجود

تقوم كل من VM وLXC بتحديث المسارات بمقارنة الحالية مقابل المرغوبة وتطبيق التغييرات:

- تكوين الشبكة (الجسر، علامة VLAN، IP لـ LXC)
- النوى / الذاكرة
- حجم القرص (VM `scsi0` / LXC `rootfs`)
- NICs الثانوية من `secondary_ips` يعيد تشغيل المورد إذا تم تغيير الشبكة أو الموارد.

الوسم

يتم وسم كل ورد بأسماء مجموعته في Ansible + `site_name`.

التوزيع عبر العقد

نفس منطق التوزيع الدائري لـ VMs وLXCs:

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

تجاوزات لكل مضيف

يمكن تعيين أي من مفاتيح LXC على مستوى المجموعة (`proxmoxLxcCores`، `proxmoxLxcMemoryMb`، `proxmoxLxcDiskSizeGb`، `proxmoxLxcRootFsStorageName`، `proxmoxLxcOsTemplate`، `host_vm_network`) أيضًا لكل مضيف تحت اسم المضيف.

```yaml
dns:
  hosts:
    site-dns01:
      ansible_host: 192.168.1.20
      proxmoxLxcDiskSizeGb: 120   # تجاوز لكل مضيف
      host_vm_network: vmbr1      # تجاوز للجسر لكل مضيف
  vars:
    proxmox_interface: vmbr0
    gateway: 192.168.1.1
    netmask: 255.255.255.0
```

`util_playbooks/proxmox_lxc.yml` القديمة

مُهملة. تستخدم شكل المتغيرات القديم `proxmoxServerAddress` / `PROXMOX_API_TOKEN` وليست متصلة بتدفق النشر. يجب على جميع المواقع الجديدة استخدام `services/proxmox.yml` مع `deployment_type: lxc`.

كتيبات الخدمة

تقوم كتيبات الخدمة بنشر وتكوين بنية OmniCore التحتية. تقع في دليل `services/`.

تسلسل كتيبات اللعب

تم هيكلة كتيبات اللعب بشكل هرمي لتمكين عمليات النشر السريعة والمستهدفة:

```
all.yml
├── setup_users.yml
├── apt_cache.yml
├── dns.yml
├── common.yml
├── license_server.yml
├── monitoring.yml
│   └── grafana.yml
├── epc.yml
│   ├── common.yml
│   ├── omnimme.yml
│   ├── omnisgwc.yml
│   ├── omnipgwc.yml
│   ├── upf.yml
│   ├── omnihss.yml
│   └── omnidra.yml
├── ims.yml
│   ├── pcscf.yml
│   ├── icscf.yml
│   ├── scscf.yml
│   ├── as.yml
│   ├── omnimessage.yml
│   ├── smsc.yml
│   └── omnisep.yml
├── omniepdg.yml
├── omniss7.yml
├── ocs.yml
├── crm.yml
├── ran_monitor.yml
└── health_check.yml
```

ثوانٍ. يستغرق (`omnimme.yml` مثل) **قم بتشغيل فقط ما تحتاجه.** يستغرق نشر مكون واحد `--limit` عبر 50 مضيفًا وقتًا أطول ولكنه يتعامل مع كل شيء. استخدم `all.yml` تشغيل لتضييق النطاق أكثر.

مرجع سريع

كتيبات اللعب على المستوى الأعلى

كتيب اللعب	النطاق	الوصف
`all.yml`	الشبكة الكاملة	نشر كامل: المستخدمون، DNS، المراقبة، EPC، IMS، OCS، CRM
`epc.yml`	النواة الحزمية	MME، SGW-C، PGW-C، UPF، HSS، DRA
`ims.yml`	الصوت/IMS	P/I/S-CSCF، خادم التطبيق، XCAP، الرسائل
`ocs.yml`	الشحن	CGrateS، مجموعة KeyDB، مزامنة OCS
`monitoring.yml`	القابلية للمراقبة	Prometheus، Grafana، المصدّرين، HOMER

كتيبات بنية تحتية

كتيب اللعب	الوصف
`common.yml`	وكلاء التسجيل ،NTP ،تكوين نظام التشغيل الأساسي، الحزم
`setup_users.yml`	حسابات المستخدمين المحليين ومفاتيح SSH
`dns.yml`	نشر خادم DNS
`license_server.yml`	خادم ترخي�� OmniCore
`netplan.yml`	تكوين واجهة الشبكة
`firewall.yml`	قواعد iptables/nftables
`apt_cache.yml`	مرآة APT (إذا كانت مفعلة) المحلية

مكونات EPC

الوصف	المكون	كتيب اللعب
كيان إدارة التنقل (4G)	OmniMME	`omnimme.yml`
خطة التحكم في بوابة الخدمة	OmniSGW-C	`omnisgwc.yml`
خطة التحكم في بوابة PDN	OmniPGW-C	`omnipgwc.yml`
وظيفة مستوى المستخدم (SGW-U/PGW-U مجتمعة)	OmniUPF	`upf.yml` / `omniupf.yml`
خادم المشتركين المنزلي	OmniHSS	`omnihss.yml` / `hss.yml`
وكيل توجيه Diameter	OmniDRA	`omnidra.yml`
بوابة الوصول اللاسلكي الموثوق	OmniTWAG	`omnitwag.yml`
بوابة بيانات الحزم المتطورة (الاتصال عبر WiFi)	OmniEPDG	`omniepdg.yml`
وكيل حركة مرور GTP	GTP Proxy	`gtp_proxy.yml`

مكونات IMS

الوصف	المكون	كتيب اللعب
وظيفة التحكم في جلسة المكالمات الوكيلة	P-CSCF	`pcscf.yml`
CSCF الاستجواب	I-CSCF	`icscf.yml`
CSCF الخدمة	S-CSCF	`scscf.yml`
خادم تطبيق الهاتف	OmniTAS	`as.yml`
البريد الصوتي، BSF، خادم الاستحقاق، XCAP المرئي	OmniSEP	`omnisep.yml`
وحدة تحكم الرسائل القصيرة عبر IP	OmniMessage	`omnimessage.yml`
مركز خدمة الرسائل القصيرة	SMSC	`smsc.yml`

الخدمات الداعمة

الوصف	كتيب اللعب
نظام الشحن عبر الإنترنت (CGrateS + KeyDB)	`ocs.yml`
بوابة إدارة العملاء	`crm.yml`
بوابة SS7/SIGTRAN	`omniss7.yml`
التقاط حزم SIP/Diameter	`homer.yml`
لوحات معلومات Grafana والتنبيهات	`grafana.yml`
شحن السجلات إلى Loki	`promtail.yml`
تكامل مراقبة RAN	`ran_monitor.yml`

توفير VM

الوصف	كتيب اللعب
إنشاء VMs على Proxmox VE	`proxmox.yml`
إنشاء حاويات LXC (معمل/اختبار)	`proxmox_lxc.yml`
حذف VMs/LXCs من Proxmox	`proxmox_delete.yml`

العمليات

الوصف	كتيب اللعب
نسخ احتياطي لقواعد البيانات والتكوينات	`backup.yml`
إعادة تشغيل مضبوطة للمضيفين	`reboot.yml`
إيقاف تشغيل سلس	`shutdown.yml`
تحديث الحزم	`apt_update.yml`
تحديث بيانات التعريف لمرآة APT	`apt_refresh_metadata.yml`
اختبار عرض النطاق الترددي للشبكة	`speedtest.yml`

كتيبات الاستعادة

الوصف	كتيب اللعب
استعادة OmniTAS من النسخة الاحتياطية	`restore_applicationserver.yml`
استعادة OmniMessage من النسخة الاحتياطية	`restore_omnimessage_controller.yml`
استعادة SMSC من النسخة الاحتياطية	`restore_smsc.yml`

الاستخدام

نشر كل شيء

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

نشر نظام فرعي محدد

```
# فقط النواة الحزمية
ansible-playbook -i hosts/customer/host_files/production.yml
services/epc.yml

# فقط IMS/الصوت
ansible-playbook -i hosts/customer/host_files/production.yml
services/ims.yml
```

نشر مكون واحد

```
# تحديث فقط MME
ansible-playbook -i hosts/customer/host_files/production.yml
services/omnimme.yml

# تحديث فقط المراقبة
ansible-playbook -i hosts/customer/host_files/production.yml
services/monitoring.yml
```

تحديد مضيفين محددين

```
# تشغيل all.yml ولكن فقط على مضيف واحد
ansible-playbook -i hosts/customer/host_files/production.yml
services/all.yml --limit mme01

# التشغيل على عدة مضيفين محددين
ansible-playbook -i hosts/customer/host_files/production.yml
services/all.yml --limit "mme01,hss01"

# التشغيل على مجموعة
ansible-playbook -i hosts/customer/host_files/production.yml
services/all.yml --limit mme
```

الوثائق ذات الصلة

- سير العمل العام للنشر - بنية النشر
- تعريف البنية التحتية - تكوين ملف المضيفين
- التخصيص - تكوين متغيرات المجموعة
- أدوات تشغيلية (فحص الصحة، استعادة، إلخ.) - كتيبات الأدوات المساعدة
- التنبيهات، Grafana، Prometheus - المراقبة والقابلية للمراقبة

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

الغرض	كتيب الأدوات
توفير VMs على Proxmox	`proxmox.yml`
حذف VMs/LXCs على Proxmox	`proxmox_delete.yml`
توفير حاويات LXC على Proxmox	`proxmox_lxc.yml`
توفير VMs على VMware vSphere	`vmware.yml`
حذف VMs على VMware vSphere	`vmware_delete.yml`
إنشاء تقرير صحة شامل لجميع الخدمات	`health_check.yml`
استعادة قاعدة بيانات HSS و/أو التكوين من النسخة الاحتياطية	`restore_hss.yml`
استعادة OCS KeyDB وMySQL والتكوين من النسخة الاحتياطية	`restore_ocs.yml`
إنشا�� وثائق الشبكة مع مخططات Mermaid	`ip_plan_generator.yml`
تدقيق المنافذ المفتوحة والخدمات المستمعة عبر جميع المضيفين	`get_ports.yml`
استرجاع ملفات التقاط الحزم من المضيفين	`getLocalCapture.yml`
إزالة حساب مستخدم محلي من جميع المضيفين	`delete_local_user.yml`

توفير VMs

تقوم كتيبات توفير VMs بإنشاء وإدارة الآلات الافتراضية على منصات المحاكاة الافتراضية. تدعم جميع الكتيبات `--limit` لاستهداف مضيفين أو مجموعات معينة.

راجع نشر Proxmox للحصول على تكوين مفصل.

Proxmox

الملفات: `util_playbooks/proxmox.yml`, `util_playbooks/proxmox_lxc.yml`, `util_playbooks/proxmox_delete.yml`

```
# توفير VMs
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/proxmox.yml

# توفير مجموعة معينة فقط
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/proxmox.yml --limit mme

# توفير حاويات LXC
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/proxmox_lxc.yml

# حذف VMs/LXCs
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/proxmox_delete.yml --limit old-host
```

VMware vSphere

الملفات: `util_playbooks/vmware.yml`, `util_playbooks/vmware_delete.yml`

المتطلبات المسبقة: يتطلب تثبيت التبعيات من `requirements.txt`.

الاستخدام:

```
# توفير VMs
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/vmware.yml

# توفير مجموعة معينة فقط
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/vmware.yml --limit mme

# حذف VMs
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/vmware_delete.yml --limit old-host
```

المتغيرات المطلوبة:

```yaml
all:
  vars:
    vcenter_ip: "vcenter.example.com"
    vcenter_username: "administrator@vsphere.local"
    vcenter_password: "password"
    vcenter_datacenter: "Datacenter"
    vcenter_folder: "OmniCore"
    vcenter_vm_template: "ubuntu-2404-template"
    vhosts:
      esxi-01:
        vcenter_cluster_ip: "192.168.1.10"
        vcenter_datastore: "datastore1"
```

فحص الصحة

الملف: `util_playbooks/health_check.yml`

ينشئ تقرير صحة شامل بتنسيق HTML يغطي جميع خدمات OmniCore وOmniCall 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 المجمعة

المكون	البيانات المجمعة
جميع الخدمات	حالة الخدمة، الإصدار، وقت التشغيل
OmniHSS	حالة قاعدة البيانات، اتصالات نظير Diameter
OmniDRA	اتصالات نظير والحالة Diameter
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 لدليل `/etc/omnihss`

استعادة OCS

الملف: `util_playbooks/restore_ocs.yml`

يستعيد OCS (CGrateS) متعدد الماستر KeyDB من ملفات النسخ الاحتياطي. يتعامل مع تكرار عن طريق الاستعادة إلى عقدة واحدة والسماح بالتكرار لمزامنة الآخرين.

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

عملية الاستعادة

1. إيقاف CGrateS وKeyDB على جميع عقد OCS
2. مسح ملفات AOF لمنع التعارض مع البيانات المستعادة
3. استعادة KeyDB RDB إلى العقدة الأولى، بدء KeyDB، والسماح بالتكرار للمزامنة
4. استعادة MySQL StoreDB (اختياري)
5. استعادة تكوين `/etc/cgrates` (اختياري)
6. بدء CGrateS على جميع العقد

تنسيقات ملفات النسخ الاحتياطي

النوع	نمط اسم الملف	المحتويات
بيانات KeyDB	`keydb_dump_<hostname>_<timestamp>.rdb`	لقطة RDB لـ KeyDB
MySQL StoreDB	`cgrates_dump_<hostname>_<timestamp>.sql`	تفريغ MySQL لقاعدة بيانات `cgrates`
التكوين	`cgrates_<hostname>_<timestamp>.tar.gz`	أرشيف لدليل `/etc/cgrates`

المطالبات

المطالبة	مطلوب	الوصف
مسار تفريغ KeyDB RDB	نعم	المسار الكامل لملف النسخ الاحتياطي KeyDB RDB
مسار تفريغ MySQL SQL	لا	المسار الكامل لتفريغ SQL لـ CGrateS (تخطي للحفاظ على StoreDB الحالي)
مسار التكوين tar.gz	لا	المسار الكامل لنسخة التكوين الاحتياطية (تخطي للحفاظ على التكوين الحالي)

مولد خطة IP

الملف: `util_playbooks/ip_plan_generator.yml`

ينشئ وثائق الشبكة من الجرد، بما في ذلك:

- تخصيصات IP للمضيف (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	ثائق 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، البروتوكول، المنفذ
`./open_ports.rst`	جدول reStructuredText لوثائق Sphinx

البيانات المجمعة

الحقل	الوصف
اسم المضيف	اسم المضيف في الجرد
IP	عنوان IP للمضيف `ansible_host`
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 ملفين لالتقاط الحزم من دليل `/etc/localcapture` لكل مضيف.

```
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.

تشغيل كتيبات الأدوات

الصياغة الأساسية

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

الخيارات الشائعة

الخيار	الوصف
`-i <inventory>`	تحديد ملف الجرد
`--limit <hosts>`	الحد من المضيفين أو المجموعات المحددة
`-v` / `-vv` / `-vvv`	زيادة مستوى التفاصيل
`--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 معين
ansible-playbook -i hosts/acme/host_files/production.yml
util_playbooks/restore_hss.yml --limit hss01

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