

5G Core (5GC)

يقوم دور مشترك `services/5gc.yml` playbook بواسطة (SA) G يتم نشر النواة المستقلة 5. تشرح هذه الصفحة هذا NF. يتم تهيئة الدور لكل (NF) بتثبيت كل وظيفة شبكة `gc_nf` واحد، 5 التي يبينها، والتكوين الذي يقرأه كل واحد منها NFs الدور، و.

نظرة عامة

ال NFs لجميع (`5gc_nf`) دور واحد OmniCore تستخدم NF، بدلاً من وجود دور واحد لكل الذي `nf_package` ويمرر متغير NF الدور مرة واحدة لكل playbook يستدعي. **11GC في 5** هذا يحافظ على تثبيت، إصدار. Jinja2 وقالب تكوين، systemd، خدمة، Debian يحدد حزمة NF. التكوين، الترخيص، ومنطق إعادة التشغيل متطابقاً عبر كل

```
- name: Setup AMF
  hosts: amf
  roles:
    - {role: 5gc_nf, nf_package: omniamf, become: yes}
```

وظائف الشبكة

وقالب تكوين خاص به، systemd له مجموعة جرد، حزمة، خدمة NF كل (`roles/5gc_nf/templates/<nf_package>.j2`).

الدور	قالب التكوين	مجموعة الجرد	nf_package	NF
اكتشاف الخدمة / NF تسجيل	omninrf.j2	nrf	omninrf	وظيفة مستودع الشبكة
N1/N2، إشارات التسجيل، إدارة الاتصال	omniamf.j2	amf	omniamf	إدارة الوصول والتنقل
G-AKA / مصادقة 5 EAP-AKA'	omniausf.j2	ausf	omniausf	خادم المصادقة
جلسة PCF ربط PDU	omnibsf.j2	bsf	omnibsf	وظيفة دعم الربط
شحن متقارب عبر الإنترنت / غير متصل	omnichf.j2	chf	omnichf	وظيفة الشحن
اختيار / NSSAI الشريحة	omnissf.j2	nssf	omnissf	اختيار شريحة الشبكة
قواعد السياسة وجودة الخدمة	omnipcf.j2	pcf	omnipcf	وظيفة التحكم في السياسة
SBI / توجيه رسائل الاتصالات غير المباشرة	omniscp.j2	scp	omniscp	وكيل الاتصال بالخدمة
والتحكم PDU جلسة UPF (N4) في	omnismf.j2	smf	omnismf	وظيفة إدارة الجلسات
بيانات الاشتراك، فك SUCI تشفير	omniudm.j2	udm	omniudm	إدارة البيانات الموحدة

الدور	قالب التكوين	مجموعة الجرد	nf_package	NF
تخزين بيانات داعم لـ UDM/PCF	omniudr.j2	udr	omniudr	مستودع البيانات الموحد

5 gc_nf كيفية عمل دور

:يقوم الدور بنفس التسلسل NF، لكل

1. حل `group_vars_omnicore`: يتم تعيينه إلى `<inventory>/group_vars/`، بحيث يتم حل عمليات البحث عن التكوين بالنسبة إلى الجرد النشط.
2. يتم `license_server_api_urls` **اكتشاف خوادم الترخيص**: إذا لم يتم تعيين `license_server` (`https://<host>:8443/api`) اشتقاؤه من مجموعة جرد.
3. **تثبيت المتطلبات الأساسية**: `lsb-release`، `curl`، `gpg`، `unzip`، `libglu1-mesa`، `libsctp1`.
4. الأحدث بشكل افتراضي؛ قم بتثبيته Omnitouch APT. من مستودع **NF تثبيت حزمة** (`nf_version` **انظر تثبيت الإصدار**).
5. إلى `templates/<nf_package>.j2` **تكوين القالب (المُعَلَّم)**: يقوم بتجسيد `/etc/<nf_package>/runtime-<deb_version>.exs`.
6. تكوين الإصدار النشط، ونسخ ذلك الرابط `runtime.exs` **إنشاء رابط رمزي لـ** `/opt/<nf_package>/releases/*` الرمزي إلى كل دليل.
7. كلما تغير التكوين النشط (المعالج) **NF إعادة تشغيل**.

تثبيت الإصدار والعودة

هو `runtime.exs` و `(runtime-<deb_version>.exs)` Debian يتم كتابة التكوين لكل إصدار رابط رمزي للإصدار النشط. يتم الاحتفاظ بالتكوينات السابقة على القرص، لذا فإن العودة هي مجرد إعادة تثبيت إصدار الحزمة الأقدم:

```
- {role: 5gc_nf, nf_package: omniamf, nf_version: "1.2.2-1", become: yes}
```

لتتبع أحدث حزمة `nf_version` تجاهل

(SUCI فك تشفير) UDM لـ HNET مفاتيح

SUCI (TS 33.501) خطوات إضافية لدعم فك تشفير (nf_package: omniudm) UDM يمتلك عند التسجيل. يحتاج إلى UE المخفية التي يقدمها SUCI من SUPI يستعيد فك التشفير. (§6.12) USIMs مفاتيح الشبكة المنزلية **السرية** التي تتراوح مع المفاتيح العامة للشبكة المنزلية على

- ك Ansible على وحدة التحكم `group_vars/hnet_udm/` تعيش المفاتيح في `<curve>-<id>.key` (PEM)، حيث `<curve>` هو `curve25519` (Profile A، X25519) أو `secp256r1` (Profile B، prime256v1)، و `<id>` هو معرف المفتاح العام للشبكة المنزلية.
- افتراضيًا بحيث يعمل فك `curve25519-1.key` إذا كان الدليل **فارغًا**، يتم إنشاء التشفير من الصندوق. هذه الخطوة غير متغيرة. لا يتم الكتابة فوق مفتاح موجود.
- (hex) يتم تصدير القيمة العامة الخام المطابقة 32 بايت، `curve25519-*.key` لكل يتم `*.key` في الجرد؛ فقط ملفات `.pub`. تبقى ملفات USIM. لتوفير `.pub`. إلى ملف (الوضع 0600، الجذر) UDM على `/etc/omniudm/hnet/` نشرها إلى

المبرمجة USIMs من SUCIs المفتاح الذي تم إنشاؤه تلقائيًا عشوائي ويقوم فقط بفك تشفير المصدرة وقدمها كمفتاح عام للشبكة المنزلية على `.pub`. بمفتاحها العام **المطابق**. خذ ويمكنها ترك HNET التي تستخدم المخطط الفارغ إلى مفتاح SIM لا تحتاج بطاقات USIM. الدليل فارغًا.

تفاصيل كاملة وأمثلة تخطيط الملفات: **تكوين المتغيرات الجماعية: المثال 8**

الوثائق ذات الصلة

- الخدمة وهيكل النشر playbooks جميع - **Service Playbooks**
- (BSC، MSC، SS7) القديم CS **النواة الدائرية** - مجال
- HNET ومفاتيح NF **تكوين المتغيرات الجماعية** - قوالب تكوين لكل
- **معمارية النشر** - كيف تتناسب المجالات معًا
- **OmniCore: تكوين**
ما يتم تضمينه في - <https://docs.omnitouch.com.au/docs/repos/OmniCore>
NF ملفات تكوين

لـ IP معيار تخطيط OmniCore

نظرة عامة

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

IP متطلبات تخصيص

التخصيص القياسي: أربع شبكات فرعية /24

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

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

مهم: هذه توصيات، وليست متطلبات

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

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

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

يوفر نهج الشبكات الفرعية الأربعة الموصى به **عزل أمني، إدارة حركة المرور، ووضوح IP عمليات** مثالي، ولهذا السبب نقترحه للنشر الإنتاجية. ومع ذلك، يجب عليك تكييف خطة لتناسب طوبولوجيا الشبكة الخاصة بك، ومساحة العناوين المتاحة، ومتطلبات التشغيل.

تحليل تقسيم الشبكة

1. شبكة النواة الحزمية (أول /24)

الغرض: عناصر خطة المستخدم وخطة التحكم الأساسية

المكونات:

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

مثال: 10.179.1.0/24

```
mme:
  hosts:
    customer-mme01:
      ansible_host: 10.179.1.15
      gateway: 10.179.1.1
      host_vm_network: "v400-omni-packet-core"
```

2. شبكة الإشارة (ثاني /24)

الغرض: إشارات قطرية، سياسة، وظائف شحن وإدارة

المكونات:

- OmniHSS (خادم المشتركين المنزلي)

- OmniCharge OCS (نظام الشحن عبر الإنترنت)
- OmniHSS PCRF (وظيفة قواعد السياسة والشحن)
- OmniDRA DRA (وكيل توجيه قطرية)
- IMS الموجه للعملاء في قسم DNS داخلي/بنية تحتية - متميز عن DNS خوادم (العامة؛ كلاهما مقصود)
- خوادم TAP3/CDR
- OAM/المراقبة
- SIP التقاط
- خادم الترخيص
- مراقبة RAN
- إذا تم نشره - (مركز بث الخلايا) Omnitouch CBC رابط تحذير

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: "v401-omni-signalling"
```

3. الداخلية (ثالث / 24) IMS شبكة

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

المكونات:

- OmniCSCF S-CSCF (وظيفة التحكم في جلسة المكالمات الخدمية)
- OmniCSCF I-CSCF (وظيفة التحكم في جلسة المكالمات الاستجوابية)
- OmniTAS (خادم تطبيقات الهاتف / خادم التطبيقات)
- OmniMessage (SMS، SMPP، IMS وحدة التحكم في)

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: "v402-omni-ims-internal"
```

4. العامة (رابع /24) IMS شبكة

القديمة/SS7 الموجه للعملاء، والتوصيل DNS، الموجهة للمستخدم IMS **الغرض:** خدمات

المكونات:

- وظيفة التحكم في جلسة المكالمات الوكيل (OmniCSCF P-CSCF)
- خوادم XCAP
- خوادم البريد الصوتي المرئي
- العملاء DNS
- SS7 نقطة نقل إشارات (OmniSS7 STP)
- JG/3G - 2 (سجل الموقع المنزلي) (OmniSS7 HLR)
- IP-SM-GW (MAP SMS) (OmniSS7 IP-SM-GW)
- بوابة CAMEL (OmniSS7)
- (مركز التحويل المحمول) MSC

والتوصيل القديمة ذات الصلة في القسم العام، SS7/HLR **ملاحظة:** عادةً ما توجد وظائف الداخلية SIP IMS بدلاً من SIGTRAN/M3UA الخارجية عبر G/3G حيث تتصل بشبكات 2.

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: "v403-omni-ims-public"
```

طرق التنفيذ

:طريقتين رئيسيتين لتنفيذ هذا التقسيم الشبكي OmniCore يدعم

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

منفصلة أو جسور افتراضية لكل قسم شبكة. يوفر هذا أقوى عزل وهو النهج NICs استخدم
(v400-omni-packet- core, v401-omni-signalling, v402-omni-ims-internal, v403-omni-ims-
public) الموصى به للنشرات الإنتاجية. القاعدة هي تسمية كل جسر بعد قسمه
صالح أيضًا حيث يتم التعامل مع (v400-omni-packet-core) مثل (v401-omni-signalling) على الرغم من أن جسر مشترك واحد،
التقسيم في مكان آخر.

مثال:

```
# النواة الحزمية - v400-omni-packet-core
mme:
  hosts:
    omni-site-mme01:
      ansible_host: 10.179.1.15
      gateway: 10.179.1.1
      host_vm_network: "v400-omni-packet-core"

# الإشارة - v401-omni-signalling
hss:
  hosts:
    omni-site-hss01:
      ansible_host: 10.179.2.140
      gateway: 10.179.2.1
      host_vm_network: "v401-omni-signalling"

# IMS الداخلية - v402-omni-ims-internal
icscf:
  hosts:
    omni-site-icscf01:
      ansible_host: 10.179.3.55
      gateway: 10.179.3.1
      host_vm_network: "v402-omni-ims-internal"

# IMS العامة - v403-omni-ims-public
pcscf:
  hosts:
    omni-site-pcscf01:
      ansible_host: 10.179.4.165
      gateway: 10.179.4.1
      host_vm_network: "v403-omni-ims-public"
```

VLAN الطريقة 2: تقسيم قائم على

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

مثال:

```

# VLAN : كل قسم يستخدم وسم(ovsbr1) جميع المكونات تشترك في جسر واحد
مختلف.
# applicationserver (TAS) هو مكون IMS داخلي -> VLAN 402 / 10.178.3.x
applicationserver:
  hosts:
    omni-site-sbc01:
      ansible_host: 10.178.3.213
      gateway: 10.178.3.1
      host_vm_network: "ovsbr1"
      vlanid: "402"

# dra و dns هما مكونات الإشارة -> VLAN 401 / 10.178.2.x
dra:
  hosts:
    omni-site-dra01:
      ansible_host: 10.178.2.211
      gateway: 10.178.2.1
      host_vm_network: "ovsbr1"
      vlanid: "401"

dns:
  hosts:
    omni-site-dns01:
      ansible_host: 10.178.2.178
      gateway: 10.178.2.1
      host_vm_network: "ovsbr1"
      vlanid: "401"

```

تكوين الشبكة:

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

VLAN مثال على تخطيط:

```

VLAN 400: 10.x.1.0/24 (النواة الحزمية)
VLAN 401: 10.x.2.0/24 (الإشارة)
VLAN 402: 10.x.3.0/24 (الداخلية IMS)
VLAN 403: 10.x.4.0/24 (IMS العامة)

```

العامّة IP العمل مع عناوين

نظرة عامة

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

- **DRA** - لشارات قطرية التجوال مع شركات النقل الخارجية
- **SGW/PGW** من الشركاء المتجولين GTP **للتجوال** - لحركة مرور
- **ePDG** (من UEs من IPsec نفق) WiFi لمكالمات
- الخارجية SMS مع مجمعات SMPP لاتصالات - **SMSC بوابة**
- **P-CSCF** UE SIP للتسجيل المباشر ل- (في بعض النشرات)

العامّة IPs كيفية تخصيص

الداخلية في ملفات جرد **IPs** العامّة بنفس الطريقة تمامًا كما يتم التعامل مع **IPs** تُعالج مع البوابة المناسبة وقناع `ansible_host` العام في حقل IP المضيفين لديك. ببساطة حدد عنوان الشبكة.

العامّة IPs للتجوال مع SGW/PGW :مثال

```

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

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

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

```

عام IP مع DRA :مثال

```

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

```

عام IP مع ePDG :مثال

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

الداخلية والعامة IPs مزج

الداخلية والعامة ضمن نفس مجموعة المكونات. على سبيل المثال IPs من الشائع وجود مزيج من

- GTP الداخلية للمواقع المحلية باستخدام SGWs
- العامة خصيصًا لحركة مرور التجوال من شركات النقل الخارجية SGWs
- الداخلية والخارجية SGWs كل من PGW-C يمكن أن تدير نفس

المناسبة IP مع هذا بـ   لاسة - فقط قم بتكوين كل مضيف مع عناوين OmniCore تتعامل بنية له.

Ansible مقدمة في نشر في Omnitouch

نظرة عامة

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

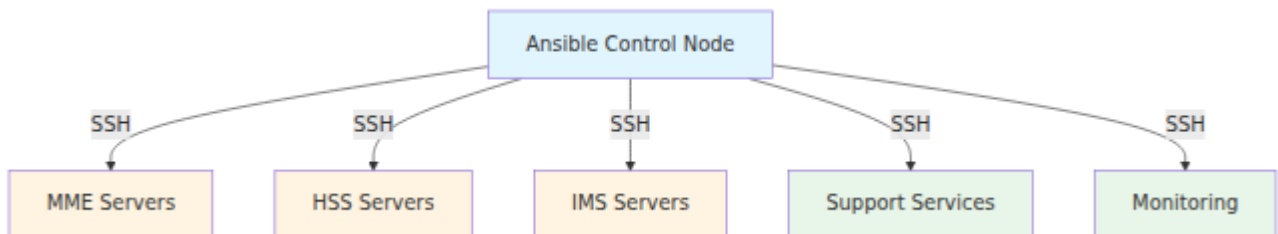
ما هو Ansible؟

هو أداة أتمتة مفتوحة المصدر تتيح لك Ansible:

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

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

Omnitouch Ansible كيف تستخدم



المفاهيم الأساسية

1. الجرد (ملفات المضيفين)

:يحدد **ما** الأنظمة التي يجب إدارتها. تحتوي كل نشرة عميل على ملف مضيف يصف

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

.ملفات المضيفين هي ما ستعمل معه لتعريف شبكتك

انظر: **تكوين ملف المضيفين**

2. الأدوار

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

- مهام (خطوات للتنفيذ)
- قوالب (قوالب ملفات التكوين)
- معالجات (إجراءات يتم تشغيلها بواسطة التغييرات)
- متغيرات (قيم التكوين الافتراضية)

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

هذه الأدوار. يمكنك تعديلها. ومع ذلك، هناك عمومًا طرق أنظف لإجراء أي تعديلات ONS تحدد فريق تحتاجها من داخل ملف المضيفين الخاص بك.

3. دفاتر التشغيل

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

```
- name: Setup Omnicore MME
  hosts: mme
  roles:
    - omnimme
```

.نستخدم هذه بشكل أساسي كمجموعات للأدوار

في الممارسة العملية، يطبق كل دفتر تشغيل خدمة دورًا واحدًا، ويتم سحب الإعداد المشترك

لتشغيل `import_playbook` تستخدم `services/epc.yml` بشكل منفصل. حزمة الخدمة مثل `common.yml` أولاً ثم كل دفتر تشغيل خاص بالخدمة

- name: Install Common
import_playbook: common.yml
- name: Run MME Playbook
import_playbook: omnimme.yml

4. متغيرات المجموعة

يوفر تكوينًا محددًا للعميل يتجاوز القيم الافتراضية للدور. هنا يحدث تخصيص العميل دون تعديل الأدوار الأساسية.

انظر: متغيرات المجموعة والتكوين

هندسة النشر



عملية النشر

1. تعريف البنية التحتية

إنشاء ملف مضيف يصف طوبولوجيا شبكتك:

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

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

انظر: تكوين ملف المضيفين و مرجع التكوين

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

2. تخصيص التكوين

group_vars: تعيين المتغيرات المحددة للعمليات في

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

إضافة رابط هنا إلى مرجع التكوين للقائمة الكاملة - #ToDo

3. تشغيل دفاتر التشغيل

نشر الشبكة:

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

4. النشر الآلي

بـ Ansible سوف يقوم

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

المكونات الرئيسية التي نقوم بنشرها

OmniCore (G/5G منصة النواة الحزمية 4)

- **OmniHSS** - خادم المشتركين المنزلي
- **OmniSGW** - بوابة الخدمة (طبقة التحكم)
- **OmniPGW** - بوابة الحزم (طبقة التحكم)
- **OmniUPF** - وظيفة طبقة المستخدم
- **OmniDRA** - Diameter وكيل توجيه
- **OmniTWAG** - بوابة الوصول اللاسلكي الموثوق

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

OmniCall (منصة الصوت والرسائل)

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

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

OmniCharge/OmniCRM

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

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

خدمات الدعم

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

- **المراقبة** - Prometheus, Grafana

انظر: [نظرة عامة على هندسة النشر](#)

إدارة الحزم

:نستخدم نموذج توزيع حزم هجين

مسبقة التجميع APT حزم

: (deb files) Debian كحزم Omnitech توزع جميع برامج

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

APT نظام ذاك؟؟ ة التخزين المؤقت

:يمكن للعملاء الاختيار بين

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

الذي يجمع دفاتر التشغيل والأدوات (packaging/) مبني من) deb. موحد omnichore يوجد أيضًا لتوفير نشر ذاتي.

APT انظر: [نظام ذاكرة التخزين المؤقت](#)

إدارة الترخيص

تراخيص صالحة يتم إدارتها من خلال خادم ترخيص Omnitech تتطلب جميع مكونات برامج 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

:بنشر Ansible يمكن أن تقوم نفس دفاتر تشغيل

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

الاتساق

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

التحكم في الإصدار

Git: يتم تعريف البنية التحتية ككود في

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

التخصيص بدون تعقيد

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

النشر السريع

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

البدء

المتطلبات المسبقة

uv تحتاج إلى تثبيت التبعيات المطلوبة. يستخدم هذا المشروع Ansible قبل تشغيل دفاتر تشغيل **uv**. لذا يتطلب Python لإدارة بيئة

1. تثبيت uv

مُثَبَّتًا بالفعل، قم بتثبيته باستخدام أي طريقة تناسب عقد التحكم الخاص بك. انظر uv إذا لم يكن: **uv** وثائق تثبيت

```
pipx install uv # عبر pipx
curl -Lsf https://astral.sh/uv/install.sh | sh # مثبت مستقل
```

2. تثبيت التبعيات

من جذر المستودع، قم بتشغيل

```
uv sync
```

المثبتة في) اللازمة Python بالإضافة إلى جميع حزم Ansible معزولة ويثبت `.venv/` هذا ينشئ `uv.lock` Omnitouch. لأتمتة نشر

3. تفعيل البيئة

Ansible: قم بتفعيل البيئة الافتراضية، واحتفظ بها مفعلة لجميع أوامر

```
source .venv/bin/activate # Windows: .venv\Scripts\activate
```

4. تثبيت مجموعات Ansible

`requirements.yml` (من) التي تعتمد عليها دفاتر التشغيل Ansible قم بتثبيت مجموعات

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

5. Ansible تشغيل أوامر

:مباشرة Ansible مع تفعيل البيئة، قم بتشغيل أوامر

```
ansible --version
```

.deactivate **ملاحظة:** قم بإلغاء تفعيل البيئة عند الانتهاء عن طريق تشغيل

خطوات النشر

1. مراجعة **تكوين ملف المضيفين** لفهم كيفية تعريف شبكتك
2. تعلم عن **متغيرات المجموعة** للتخصيص
3. لإدارة الحزم APT فهم **نظام ذاكرة التخزين المؤقت**
4. مراجعة **هندسة النشر** لرؤية كيفية تناسق كل شيء
5. انشر!

الخطوات التالية

- IP **خطة لهيكل شبكتك وتخصيص** - IP **معياري تخطيط**
- **تكوين ملف المضيفين** - تعلم كيفية تعريف طوبولوجيا شبكتك
- فهم توزيع الحزم - APT **نظام ذاكرة التخزين المؤقت**
- **خادم الترخيص** - تعلم عن إدارة الترخيص
- **نظرة عامة على هندسة النشر** - رؤية الصورة الكاملة
- **تكوين متغيرات المجموعة** - تخصيص نشر الخاص بك
- **دفاتر التشغيل المساعدة** - أدوات تشغيلية لفحص الصحة، النسخ الاحتياطي، والصيانة

وتوزيع الحزم APT مستودع

نظرة عامة

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

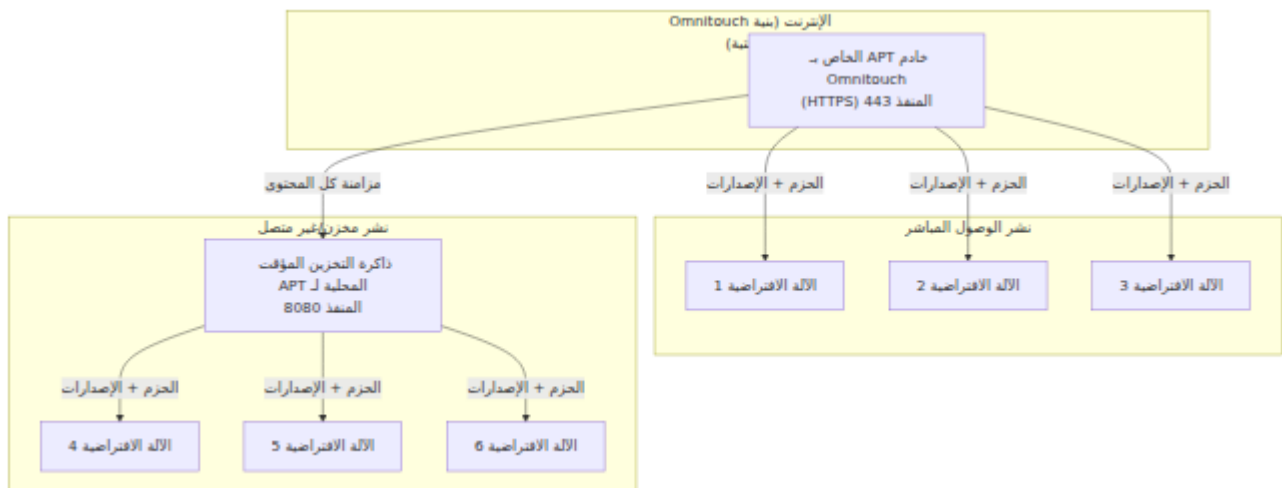
1. **حزم APT:** المثبتة عبر Debian حزم `apt install`
2. (مصدري Prometheus، **الإصدارات الثنائية:** ثنائيات مسبقة البناء يتم تنزيلها مباشرة. إلخ (الوكلاء، إلخ)

والتي توفر، **(OmniCore)** أيضًا علامة **الإعداد الذاتي** APT تتضمن صفحة الحزم على خادم التي تحتوي على مجموعة أدوات (/pool/main/o/omnicore/) المقدمة في) `omnicore` حزمة للعملاء الذين يرغبون في نشر وإدارة شبكتهم بأنفسهم Ansible الأتمتة الكاملة باستخدام

:يدعم النظام نموذجين للنشر:

1. الخاص بـ APT **الوصول المباشر:** تسحب الآلات الافتراضية الحزم مباشرة من خادم Omnitouch (مثل `apt.<region>.omnitou.ch`)
2. ويقدم Omnitouch **مرآة التخزين المحلية:** يقوم خادم محلي بمزامنة المحتوى من الحزم للآلات الافتراضية (لنشر غير المتصل/المعزول)

الهيكلية



المحتوى المقدم

كل المحتوى المطلوب للنشر APT يستضيف خادم

المسار	الوصف	نوع المحتوى
/dists/<distro>/	omnihss, omnimme, إلخ) مخصصة .deb حزم	حزم OmniTouch
/pool/main/o/omnicore/	omnicore حزمة Ansible أتمتة (الأدوار، دفاتر اللعب، التبعيات) للنشر الذاتي	مجموعة أدوات OmniCore
/<distro>/pool/main/	المخزنة مع جميع Ubuntu حزم التبعيات	حزم Ubuntu
/releases/<org>/<repo>/	ثنائيات مسبقة البناء (Prometheus, Grafana, Homer, إلخ)	إصدارات GitHub
/repos/	أرشفات المصدر لتطبيقات الويب (CGRateS_UI, speedtest)	أرشفات المصدر
/releases/<vendor>/	Galera, FRR, InfluxDB, KeyDB, إلخ	حزم الطرف الثالث

متغيرات التكوين

هو المصدر الوحيد للحقيقة لتوزيع الحزم. قم بتعريفه مرة واحدة في جردك. `apt_repo` لتنزيل الثنائيات، أو المخطط، أو المنفذ، أو URL كل شيء آخر مشتق منه. لا تقوم بتكوين عنوان بيانات الاعتماد بشكل منفصل.

apt_repo
(المصدر الوحيد للحقيقة,
المعروف في الجرد)

مشتق تلقائياً

remote_apt_*
(roles/common/defaults/main.yml)

apt_download_base
(roles/common/tasks/facts.yml)

ما يقومون بتكوينه

/etc/apt/sources.list.d/omnitouch.list

تزيلات ثنائية

/releases/, /repos/

كيف يعمل الاشتقاق

الدور	المكان الذي تم تعيينه	المتغير
الكتلة الوحيدة التي تقوم بتكوينها. يتم تضمينها مباشرة في سطر مصدر APT.	جردك	apt_repo
مشتق من الملاذ. apt_repo. النهائي عندما يكون apt_repo غائبًا: apt.us-fl.omnitou.ch / 443 / https.	roles/common/defaults/main.yml	remote_apt_*
URL عنوان الأساسي المحسوب الوحيد الذي تستخدمه كل دور لتنزيلات الثنائيات، بحيث يتفق سطر مصدر APT وتنزيلات الثنائيات دائمًا على المضيف، والمخطط، والمنفذ، وبيانات الاعتماد.	roles/common/tasks/facts.yml	apt_download_base

فإن جردًا عاديًا يحتاج فقط إلى ، apt_repo نظرًا لأن كلا القيم المشتقة مبنية من نفس كتلة بنفسك remote_apt_* لا تقوم بتعيين . apt_repo

هو المكان apt_cache_servers **استثناء: خادم مرآة التخزين.** المضيف في مجموعة الخاصة apt_repo بشكل صريح. تشير remote_apt_* الوحيد الذي لا يزال يتم فيه تعيين

Omnitouch خادم `remote_apt_*` به إلى ذاكرة التخزين المؤقت المحلية. ثم تصف قيم العلوي الذي يقوم بمزامنة منه (انظر الخيار 2)

المخطط والمنفذ

عند حذف `http` → أي منفذ آخر ، `https` → يتبع المخطط المنفذ تلقائيًا: **المنفذ 443**
`apt_repo.apt_repo_port`:

- **(HTTPS) الوصول المباشر الافتراضي هو 443**
- **(HTTP) وضع التخزين المؤقت الافتراضي هو 8080**

متى يتم استخدام كل وضع

السيناريو	APT سطر مصدر	تنزيلات ثنائية (<code>apt_download_base</code>)
<code>use_apt_cache:</code> <code>true</code> (التخزين المؤقت)	<code>http://<cache>:8080/...</code> (بدون مصادقة)	<code>http://<cache>:8080/...</code> (بدون مصادقة)
<code>use_apt_cache:</code> <code>false</code> (مباشر)	<code>https://<user>:<pass>@<server>/...</code>	<code>https://<user>:<pass>@<server>:443/...</code>

الخيار 1: الوصول المباشر

بالنسبة لعمليات النشر التي تحتوي على اتصال بالإنترنت، تسحب الآلات الافتراضية الحزم مباشرة Omnitouch الخاص بـ APT من خادم.

متطلبات الشبكة

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

- الأساسية HTTP اسم المستخدم و كلمة المرور لمصادقة

- **FQDN** لـ APT (مثل apt.<region>.omnitou.ch)

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

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

التحتية Omnitouch الخدمات التي تتطلب الوصول إلى بنية

الغرض	البروتوكول	المنفذ	الخدمة
(HTTPS) تنزيل الحزم	TCP	443	APT خادم
APT الخاص بـ FQDN لـ DNS حل	TCP/UDP	53	APT خادم
للتحقق من الترخيص NTP مزامنة الوقت	UDP	123	خادم الترخيص
للتحقق من الترخيص DNS حل	TCP/UDP	53	خادم الترخيص

(DNS (TCP+UDP/53 و، (NTP (UDP/123 و، (HTTPS (TCP/443 تأكد من السماح بحركة مرور Omnitouch الخاصة بـ IP إلى نطاقات.

التكوين

```
all:
  vars:
    use_apt_cache: false

    # و (remote_apt_*) المصدر الوحيد للحقيقة. تكوين تنزيل الثنائيات
    # مشتق من هذا تلقائيًا - لا تقوم بتعيينها apt_download_base لـ URL عنوان
    # بشكل منفصل.
    apt_repo:
      apt_server: "apt.<region>.omnitou.ch" # FQDN من المقدم
Omnitouch
      apt_repo_username: "your-username"
      apt_repo_password: "your-password"
      # apt_repo_port: 443 # اختياري؛ الافتراضي هو
443 (HTTPS)
```

المعلومات

APT (apt_repo) مصادر حزم

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>apt_repo.apt_server</code>	سلسلة	نعم	-	اسم مضيف أو APT خادم IP عنوان
<code>apt_repo.apt_repo_username</code>	سلسلة	نعم	-	اسم مستخدم مصادقة HTTP الأساسية
<code>apt_repo.apt_repo_password</code>	سلسلة	نعم	-	كلمة مرور مصادقة HTTP الأساسية
<code>apt_repo.apt_repo_port</code>	عدد صحيح	لا	443	منفذ الخادم. يتبع المخطط (443) المنفذ → HTTPS, HTTP). وإلا

(remote_apt_*) تنزيلات الشائيات

لا تقم بتعيينها في جرد الوصول المباشر. توجد `apt_repo` هذه مشتقة تلقائيًا من فقط حتى يمكن لخادم مرآة التخزين الإشارة إلى مصدر علوي مختلف (انظر الخيار 2). كمرجع، الافتراضي لها كما يلي:

المعلمة	مشتق	الملاذ
remote_apt_server	apt_repo.apt_server	apt.us-fl.omnitou.ch
remote_apt_port	apt_repo.apt_repo_port	443
remote_apt_protocol	المنفذ (443 → https)	https
remote_apt_user	apt_repo.apt_repo_username	" "
remote_apt_password	apt_repo.apt_repo_password	" "

عام

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

(الوصول المباشر) URL أنماط

APT مصادر حزم (المعرفة في `/etc/apt/sources.list.d/omnitouch.list`):

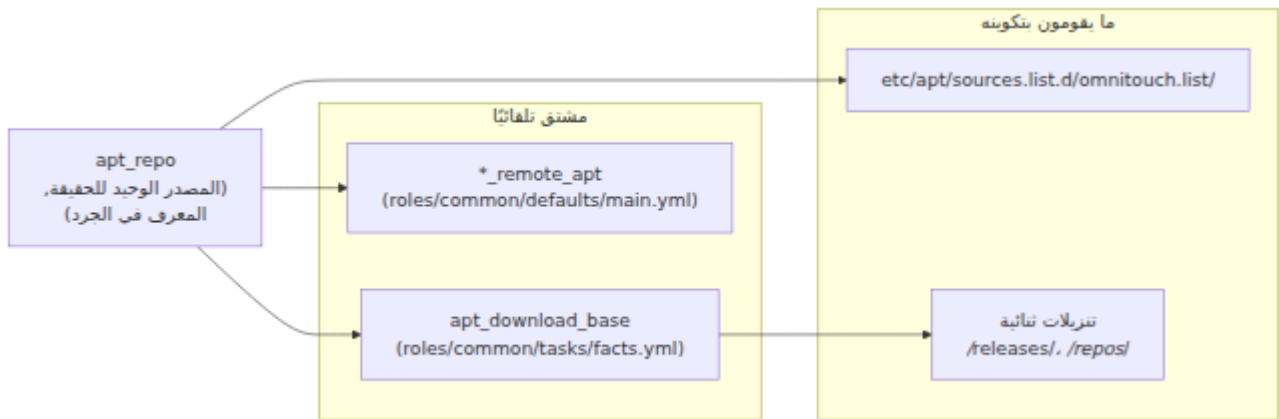
```
deb [trusted=yes] https://{username}:{password}@{apt_server}/noble main
```

Ansible `get_url` بواسطة مهام `apt_download_base` المبنية من تنزيلات الثنائيات:

```
https://{username}:{password}@{apt_server}/releases/prometheus/node_exporter/node_exporter-1.8.1.linux-amd64.tar.gz
```

يتم إضافة منفذ غير (HTTPS إنه الافتراضي لـ) URL يتم حذف المنفذ 443 من عنوان. ما لم يكن 443 HTTP ويحول المخطط إلى `<port>`: قياس ك

كيف يعمل



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

(GPG) توقيع الحزمة

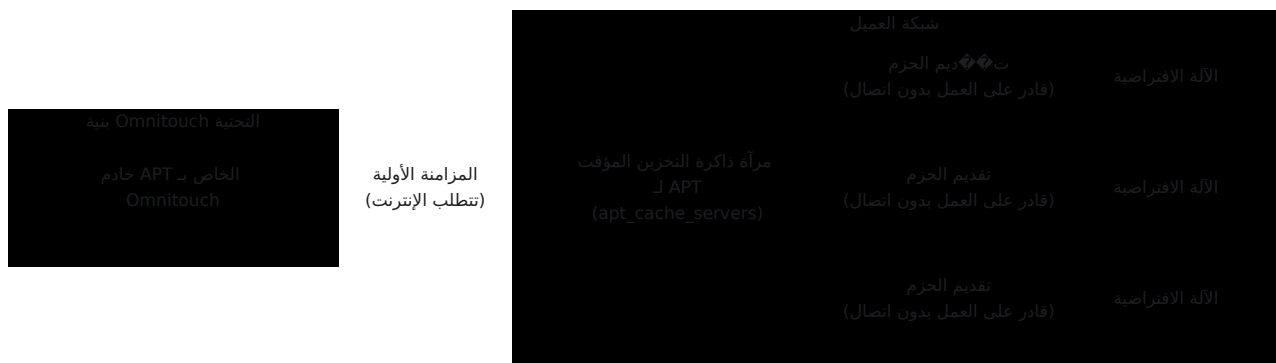
مفتاح توقيع المستودع ويقدم المفتاح العام المحمي في Omnitech الخاص بـ APT ينشئ خادم تلقائيًا بجلب `common` بالنسبة للمضيفات ذات الوصول المباشر، تقوم الدور `/omnitech.gpg` `apt-get` بحيث يتحقق `/etc/apt/trusted.gpg.d/omnitech.gpg` هذا المفتاح وتثبيته في `update` `NO_PUBKEY` ن بيانات التعريف الخاصة بالمستودع دون تحذيرات `update` لا حاجة لإدارة المفاتيح يدويًا.

تستخدم المضيفات التي تسحب من ذاكرة GPG. لا يستخدم وضع مرآة التخزين التحقق من بدلاً من ذلك (انظر الخيار 2) `[trusted=yes]` التخزين المؤقت المحلية

الخيار 2: مرآة التخزين المحلية

لعمليات النشر غير المتصلة، المعزولة، أو التي تعاني من قيود النطاق الترددي، قم بنشر ذاكرة Omnitech. تقوم بمزامنة كل المحتوى من APT تخزين مؤقت محلية لـ

الهيكلة



التكوين

قم بتعريف خادم التخزين المؤقت في ملف المضيفين الخاص بك. إنه المكان الوحيد الذي يتم فيه **العلوي** الذي تقوم Omnitouch بشكل صريح. تصف تلك المتغيرات خادم `remote_apt_*` تعيين ذاكرة التخزين المؤقت بمزامنة منه:

```
apt_cache_servers:
  hosts:
    example-apt-cache:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # العلوي Omnitouch يقوم خادم التخزين المؤقت بمزامنة المحتوى من مستودع
    remote_apt_server: "apt.<region>.omnitou.ch"
    remote_apt_port: 443
    remote_apt_protocol: "https"
    remote_apt_user: "your-username"
    remote_apt_password: "your-password"

all:
  vars:
    # use_apt_cache: true # يتم تعيينه تلقائيًا عند وجود مجموعة
  apt_cache_servers
    # apt_repo.apt_server: 192.168.1.100 (أول خادم
    تخزين مؤقتات)
```

كيف يعمل:

- **خادم التخزين المؤقت (192.168.1.100):** يستخدم بيانات اعتماد HTTPS العلوي عبر Omnitouch لمزامنة المحتوى من خادم `remote_apt_*`
- **جميع المضيفات الأخرى:** `apt_repo.apt_server:` تشق تلقائيًا وتسحب من ذاكرة التخزين المؤقت على المنفذ 8080 دون "192.168.1.100" بيانات اعتماد.

المعلومات

APT (apt_repo) مصادر حزم

المعلمة	النوع	مطلوب	الافتراضي	الوصف
<code>apt_repo.apt_server</code>	سلسلة	نعم	تلقائيًا	تخزين IP عنوان ت المحلي. يتم ، تلقائيًا من أول مضيف <code>apt_cache_s</code> لا لم يتم تحديده
<code>apt_repo.apt_repo_username</code>	سلسلة	لا	-	غير مطلوب عند استخدام التخزين مؤقت (لا حاجة للمصادقة)
<code>apt_repo.apt_repo_password</code>	سلسلة	لا	-	غير مطلوب عند استخدام التخزين مؤقت (لا حاجة للمصادقة)
<code>apt_repo.apt_repo_port</code>	عدد صحيح	لا	8080	الذي تستخدمه ات للوصول ن خادم التخزين المؤقت المحلي محل <code>apt_cache_p</code> (الذي تم إهماله)

(remote_apt_*) مزامنة خادم التخزين المؤقت

هذا Omnitouch تقوم هذه المتغيرات بتكوين كيفية مزامنة خادم التخزين المؤقت للمحتوى من يدويًا remote_apt_* هو السيناريو الوحيد الذي يتم فيه تعيين

الوصف	الافتراضي	مطلوب	النوع	المعلمة
العلوي APT خادم الخاص بـ Omnitouch للمزامنة منه	-	نعم	سلسلة	remote_apt_server
APT منفذ خادم العلوي الخاص بـ Omnitouch	443	لا	عدد صحيح	remote_apt_port
البروتوكول لاتصال المزامنة	https	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

يتم الآن تعيين المنفذ الذي تتصل به apt_cache_port تم إهماله: لم يعد يستخدم (الافتراضي هو 8080 عند حذفه) apt_repo.apt_repo_port المضيفات عبر

(وضع التخزين المؤقت) URL أنماط

APT مصادر حزم (المعرفة في `/etc/apt/sources.list.d/omnitouch.list`):

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

apt_download_base (المبنية من) تنزيلات الثنائيات:

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

APT `[trusted=yes]` لا تحتاج الوصول إلى التخزين المؤقت إلى بيانات اعتماد. يستخدم تكوين

نشر التخزين المؤقت

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

2. قم بتشغيل برنامج إعداد التخزين المؤقت:

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

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

ما الذي يتم مزامنته

تختلف الآلية. Omnitouch الخاص بـ APT تقوم مرآة التخزين بمزامنة كل المحتوى من خادم حسب نوع المحتوى:

- تتم مزامنتها (Ubuntu و Omnitouch لكل من `/dists/` + `/pool/`) APT حزم باستخدام `curl` Debian مدفوعة بفهرس مستودع `curl` للحصول على اسم `Packages.gz` العلوي (وتتجاوز إذا لم يتغير)، وتحلل `Release` حجم كل حزمة، وتقوم بتنزيل الملفات التي تفتقر إليها فقط أو التي لا `SHA256`/الملف مع النسخة المحلية. يتم التحقق من كل ملف تم تنزيله باستخدام `SHA256` تتطابق قبل الالتزام به `SHA256`.

- ليس لديها (/releases/, /repos/) الإصدارات الثنائية وأرشفات المصدر
wget --recursive --timestamping، لذا تتراجع إلى Debian فهرس بأسلوب
أحدث Last-Modified الذي يعيد تنزيل الملفات فقط التي تحتوي على

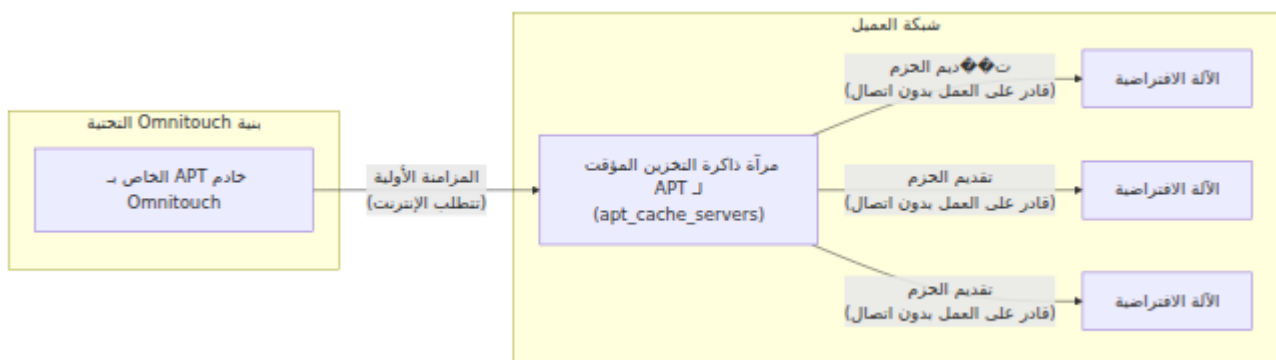
الخامس بـ APT خادم	أرشفات المصدر	إصدارات GitHub	التبنيات + حزم Ubuntu	حزم Omnitouch .deb
/dists/	/repos/	/releases/	/noble/pool/main/	/pool/main/
curl + index/SHA256	wget --recursive --timestamping	wget --recursive --timestamping	curl + index/SHA256	curl + index/SHA256
بيانات التعريف الخاصة بـ APT	أرشفات المصدر	إصدارات GitHub	التبنيات + حزم Ubuntu	حزم Omnitouch .deb
/dists/	/repos/	/releases/	/noble/pool/main/	/pool/main/
curl + index/SHA256	wget --recursive --timestamping	wget --recursive --timestamping	curl + index/SHA256	curl + index/SHA256

دلائل المحتوى المزامنة:

المسار	المحتوى
/dists/<distro>/	APT بيانات التعريف الخاصة بمستودع Packages, Release)
/pool/main/	Omnitouch المخصصة لـ .deb حزم
/<distro>/pool/main/	وجميع التبنيات Ubuntu حزم
/releases/	(إلخ، Zabbix، Grafana، Prometheus) إصدارات GitHub
/repos/	(إلخ، Erlang، Elixir، CGrateS_UI) أرشفات المصدر

بعد المزامنة الأولية، يمكن أن تقدم الذاكرة التخزينية جميع الحزم دون اتصال بالإنترنت.

كيف يعمل



الأساسية لكل المحتوى. تتم مزامنة HTTP تقوم مرآة ❖❖ لتخزين بالمصادقة باستخدام مصادقة وتنزيل، (Release + Packages.gz) Debian ضد فهرس مستودع curl باستخدام APT حزم عن النسخة المحلية والتحقق من كل منها بعد التنزيل؛ SHA256 الملفات فقط التي تختلف في كلتا. wget --recursive --timestamping تتراجع إلى /repos/ و /releases/ الحالتين، تقوم المزامنات اللاحقة بجلب الملفات الجديدة أو المتغيرة فقط.

التكوين التلقائي

في جردك، يقوم النظام تلقائيًا apt_cache_servers عند وجود مجموعة

1. لجميع المضيفات (ما لم يتم تجاوزها صراحة) use_apache: true بتعيين
2. الخاص بأول خادم ansible_host IP من عنوان apt_repo.apt_server باشتقاق تخزين مؤقت

مثال على التكوين الأدنى

```
apt_cache_servers:
  hosts:
    apt-cache-01:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # العلوي Omnitouch يقوم خادم التخزين المؤقت بمزامنة المحتوى من مستودع
    remote_apt_server: "apt.<region>.omnitou.ch"
    remote_apt_user: "your-username"
    remote_apt_password: "your-password"
```

ما يحدث تلقائيًا:

- use_apache: جميع المضيفات (باستثناء خادم التخزين المؤقت) تحصل على true
- جميع المضيفات (باستثناء خادم التخزين المؤقت) تحصل على apt_repo.apt_server: "192.168.1.100"
- دون بيانات اعتماد http://192.168.1.100:8080/ جميع المضيفات تسحب من
- خادم التخزين المؤقت يقوم بمزامنة الحزم من https://your-username:your-password@apt.<region>.omnitou.ch/

تجاوز السلوك التلقائي

لإجبار الوصول المباشر حتى مع تعريف خوادم التخزين المؤقت

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

    apt_repo:
      apt_server: "apt.<region>.omnitou.ch"
      apt_repo_username: "user"
      apt_repo_password: "pass"
```

ملخص التكوين

بدون APT السيناريو 1: الوصول المباشر إلى خادم تخزين مؤقت

مطلوب. `apt_repo` فقط. APT. تقوم جميع المضيفات بسحب الحزم مباشرة من خادم مستودع منها `remote_apt_*` يتم اشتد❖❖ اق قيم

```
all:
  vars:
    use_apt_cache: false

    # وتنزيلات الثنائيات APT المصدر الوحيد للحقيقة لكل من مصادر
    apt_repo:
      apt_server: "apt.<region>.omnitou.ch"
      apt_repo_username: "user"
      apt_repo_password: "pass"
```

النتيجة: جميع المضيفات تولد `deb [trusted=yes] https://user:pass@apt.<region>.omnitou.ch/ noble main`

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

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

```
apt_cache_servers:
  hosts:
    cache-server:
      ansible_host: 192.168.1.100
      gateway: 192.168.1.1
  vars:
    # يقوم خادم التخزين المؤقت بمزامنة الحزم من المستودع المعتمد العلوي
    remote_apt_server: "apt.<region>.omnitou.ch"
    remote_apt_port: 443
    remote_apt_protocol: "https"
    remote_apt_user: "user"
    remote_apt_password: "pass"

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

النتيجة:

- **خادم التخزين المؤقت:** `https://user:pass@apt.<region>.omnitou.ch/` يقوم بالمزامنة من
- **جميع المضيفات الأخرى:** تولد `deb [trusted=yes]`
`http://192.168.1.100:8080/noble noble main` (بدون بيانات اعتماد)

غير موجودة APT السيناريو 3: ذاكرة التخزين المؤقت في ملف المضيفين (يدوي)

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

```
all:
  vars:
    use_apt_cache: true

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

    # apt_cache_servers لا حاجة لمجموعة
    # (التخزين مؤقت مُعد بالفعل خارجيًا) remote_apt_* لا حاجة لـ
```

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

مثال كامل

هذا مثال كامل يعمل. يظهر تكوين خادم التخزين المؤقت   مع عدة مضيفات تطبيق

مجموعة خادم التخزين المؤقت APT

apt_cache_servers:

hosts:

example-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 المؤقت بمزامنة الحزم من المستودع المعتمد العلوي

remote_apt_server: "apt.<region>.omnitou.ch"

remote_apt_port: 443

remote_apt_protocol: "https"

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 (ممكن تلقائيًا عند وجود apt_cache_servers)

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

ما يحدث أثناء النشر:

1. خادم التخزين المؤقت (10.179.1.114):

- `vars:` من قسم `remote_apt_*` يستخدم
- يقوم بتنزيل جميع الحزم من `https://customer-username:customer-secure-token@apt.<region>.omnitou.ch/`
- `nginx` يقدم الحزم على المنفذ 8080 عبر

2. مضيفات التطبيقات (`customer-hss01`, `customer-mme01`, `customer-dns01`):

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

تحديث التخزين المؤقت

للمزامنة مع حزم جديدة أو تحديثات:

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

بشكل تدريجي Omnitouch الخاص بـ APT تقوم هذه العملية بمزامنة كل المحتوى من خادم:

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

- أرشيفات المصدر المحدثة

العلوي بواسطة Debian بفهرس APT تكون المزامنات اللاحقة تدريجية: تتم مقارنة حزم `/releases/` بينما تستخدم `Release` (العلوي) وتتخطى تمامًا إذا لم يتغير ملف (SHA256) يتم تخطي الملفات غير المتغيرة الموجودة، مما يجعل `wget --timestamping` `/repos/` وإعادة المزامنة سريعة.

هو المصدر الوحيد للحقيقة لجميع الحزم. تقوم Omnitouch الخاص بـ APT **ملاحظة:** خادم على مرآة التخزين `services/apt_cache.yml` بإنشاء ونشر الحزم عليه؛ تشغيل Omnitouch. المؤقت بعد ذلك يقوم بمزامنة ما هو منشور حاليًا.

استكشاف الأخطاء وإصلاحها

مع 401 غير مصرح به APT فشل تحديث

الأعراض:

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

الأسباب المحتملة:

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

الحل:

1. مع بيانات الاعتماد في `apt_repo` تحقق من نطاق التكوين: تأكد من تعريف `all: vars:` وليس في `apt_cache_servers: vars:`

2. **تحقق من وضع التخزين المؤقت:** عند استخدام التخزين المؤقت، يجب أن تتصل المضيفات بخادم التخزين المؤقت (المنفذ 8080)، وليس بالمستودع العلوي
3. **تحقق من المصادر المولدة:** على المضيف الذي فشل، تحقق من `/etc/apt/sources.list.d/omnitouch.list`
 - **الصحيح (وضع التخزين المؤقت):** `deb [trusted=yes] http://10.179.1.114:8080/noble noble main`
 - **غير صحيح (لديه بيانات الاعتماد في المكان الخطأ):** `deb [trusted=yes] https://user:pass@10.179.1.115/noble noble main`
4. تحقق من أن بيانات الاعتماد صحيحة لوضع النشر الخاص بك
5. Omnitouch العام الخاص بك مدر ◆◆ في القائمة البيضاء مع IP تأكد من أن عنوان (إذا كنت تستخدم الوصول المباشر)

فشل تنزيلات الثنائيات (Node Exporter، Zabbix، إلخ.)

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

الأسباب المحتملة:

- تنزيلات الثنائيات تستمد مصادقتها من) غير صحيحة `apt_repo` بيانات اعتماد `apt_repo` (في الوضع المباشر)
- `remote_apt_user` / على خادم مرآة التخزين، بيانات اعتماد `remote_apt_password` غير صحيحة
- وأي منفذ آخر يعني HTTPS المخطط أو المنفذ خاطئ. تذكر أن المنفذ 443 يعني HTTP.

الحل:

1. `apt_repo.apt_repo_username` / في الوضع المباشر، تحقق من أن `apt_repo.apt_repo_password` Omnitouch تتطابق مع تلك المقدمة من
2. العلوية `remote_apt_*` على خادم مرآة التخزين، تحقق من بيانات اعتماد
3. المشتقة تستخدم المضيف/المخطط/المنفذ `apt_download_base` تأكد من أن `ansible-playbook ... -vvv` تحقق باستخدام) المتوقع (الفاشلة `get_url`)

خادم التخزين المؤقت لا يمكنه المزامنة

الأعراض: فشل برنامج خادم التخزين المؤقت في تنزيل الحزم

الأسباب المحتملة:

- خادم التخزين المؤقت ليس لديه وصول إلى الإنترنت
- غير صحيحة `remote_apt_*` بيانات اعتماد
- جدار الحماية يمنع الاتصالات الخارجية إلى Omnitech

الحل:

1. العلوي الخاص بـ APT تحقق من أن خادم التخزين المؤقت يمكنه الوصول إلى خادم Omnitech 443 على المنفذ
2. `remote_apt_*` تحقق من بيانات اعتماد
3. راجع قواعد جدار الحماية للوصول الخارجي

الوثائق ذات الصلة

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

تسجيل مركزي

نظرة عامة

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

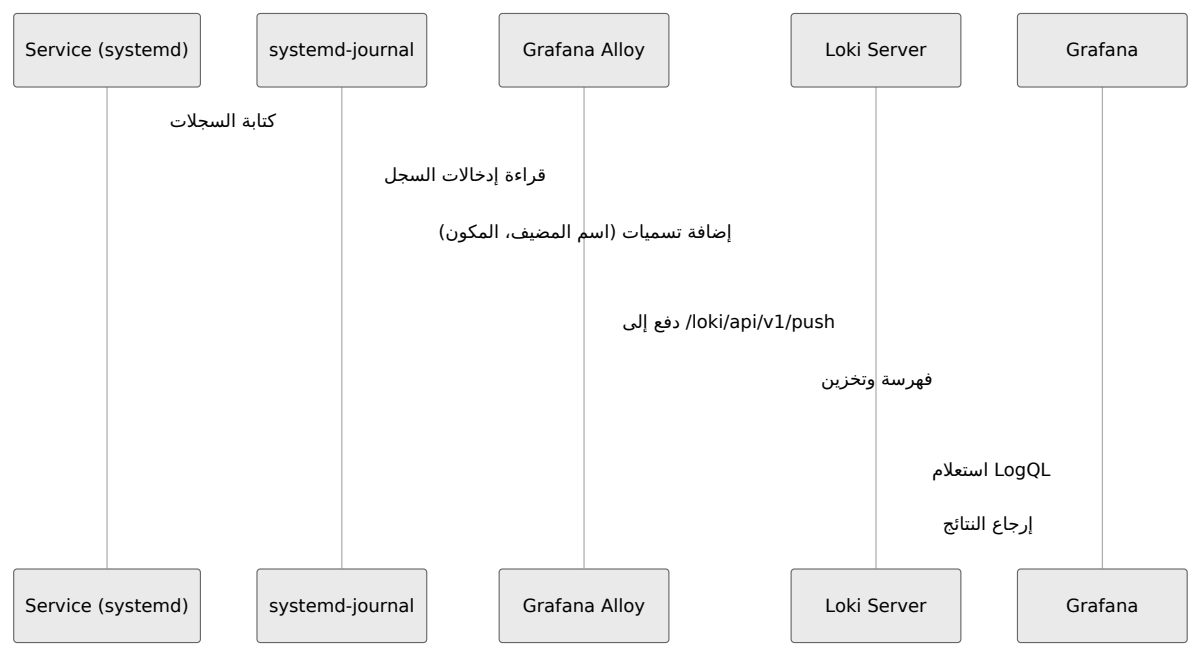
- **Grafana Alloy**: وكيل جمع السجلات يتم نشره على جميع الآلات الافتراضية
- **Grafana Loki**: خادم تجميع وتخزين السجلات على مضيفي المراقبة
- **Grafana**: لوحات مسبقة البناء لكل نوع من المكونات

الهيكلية



كيف يعمل

تدفق جمع السجلات



تسميات السجل

تتضمن كل إدخال سجل هذه التسميات للتصفية والاستعلام:

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

التكوين

النشر التلقائي

:يتم تكوين التسجيل تلقائيًا عندما

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

. لا يتطلب الأمر أي تكوين إضافي لوظائف أساسية

متطلبات الجرد

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

:`monitoring` عند تعريف مجموعة

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

تكوين الاحتفاظ

مع الاحتفاظ القائم على الوقت. يقوم الضاغط بحذف السجلات الأقدم من فترة Loki يتم تكوين الاحتفاظ:

الوصف	الافتراضي	الإعداد
يتم حذف السجلات الأقدم من فترة الاحتفاظ	ساعة (3 72 أيام)	الاحتفاظ القائم على الوقت

:لتخصيص فترة الاحتفاظ، قم بتجاوز هذه المتغير في جردك

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

Alloy تكوين ذاكرة التخزين المؤقت لـ

غير متاح. Loki عندما يكون (WAL) بتخزين السجلات محليًا في سجل الكتابة المسبق Alloy يقوم
:يتم التخلص من القطع المخزنة لمنع نمو القرص غير المحدود

الوصف	الافتراضي	الإعداد
يتم إسقاط قطع السجل المخزنة التي تزيد عن 1 ساعة	ساعة 1	الحد الأقصى لعمر القطعة في WAL

.غير متاح لفترة أطول من عمر القطعة، يتم إسقاط أقدم السجلات المخزنة Loki عندما يكون

Grafana لوحات

.يتم توفير لوحات مسبقة البناء تلقائيًا لكل نوع من المكونات

اللوحات المتاحة

الوصف	الموقع	اللوحة
P-CSCF, S-CSCF, I-CSCF (OmniCSCF) سجلات	CSCF السجلات → سجلات	CSCF سجلات
للتوصيل/الفصل MME أحداث والأخطاء	MME السجلات → سجلات	MME سجلات
وحامل SGW أحداث جلسة البيانات	SGW السجلات → سجلات	SGW سجلات
PGW وجلسة PDN أحداث	PGW السجلات → سجلات	PGW سجلات
وأحداث HSS Diameter رسائل المصادقة	HSS السجلات → سجلات	HSS سجلات
SMPP وأحداث SMS تسليم	السجلات → سجلات OmniMessage	سجلات OmniMessage
أحداث الشحن وحركة مرور JSONRPC	السجلات → سجلات OCS/CGrateS	سجلات OCS/CGrateS
RPC البحث ومطابقة طلبات/ردود مع الكمون	RPC السجلات → مكالمات CGrateS	RPC مكالمات CGrateS

مميزات اللوحة

تتضمن كل لوحة:

- **صندوق البحث:** بحث نصي حر عبر جميع السجلات
- **رسوم بيانية لمعدل السجل:** الحجم بمرور الوقت حسب المضيف والشدة
- **أقسام المكونات:** عرض مجمع (على سبيل المثال) P-CSCF, S-CSCF, I-CSCF
- **تصفية الأخطاء:** عرض مسبق التصفية للأخطاء والإخفاقات
- **ذيل حي:** بث سجلات في الوقت الحقيقي (تحديث كل 10 ثواني)

استخدام اللوحات

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

استعلام السجلات

LogQL أساسيات

:للاستعلام. الصيغة الأساسية LogQL لـ Loki يستخدم

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

الاستعلامات الشائعة

:جميع السجلات من مضيف معين

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

MME: جميع سجلات مكون

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

CSCFs: البحث عن الأخطاء عبر جميع

```
{component=~"pcscf|scscf|icscf"} |~ "(?i)error"
```

systemd: تصفية حسب وحدة

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

دمج الفلاتر:

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

استعلامات معدل السجل

حجم السجل لكل مكون:

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

CSCF معدل الأخطاء لـ

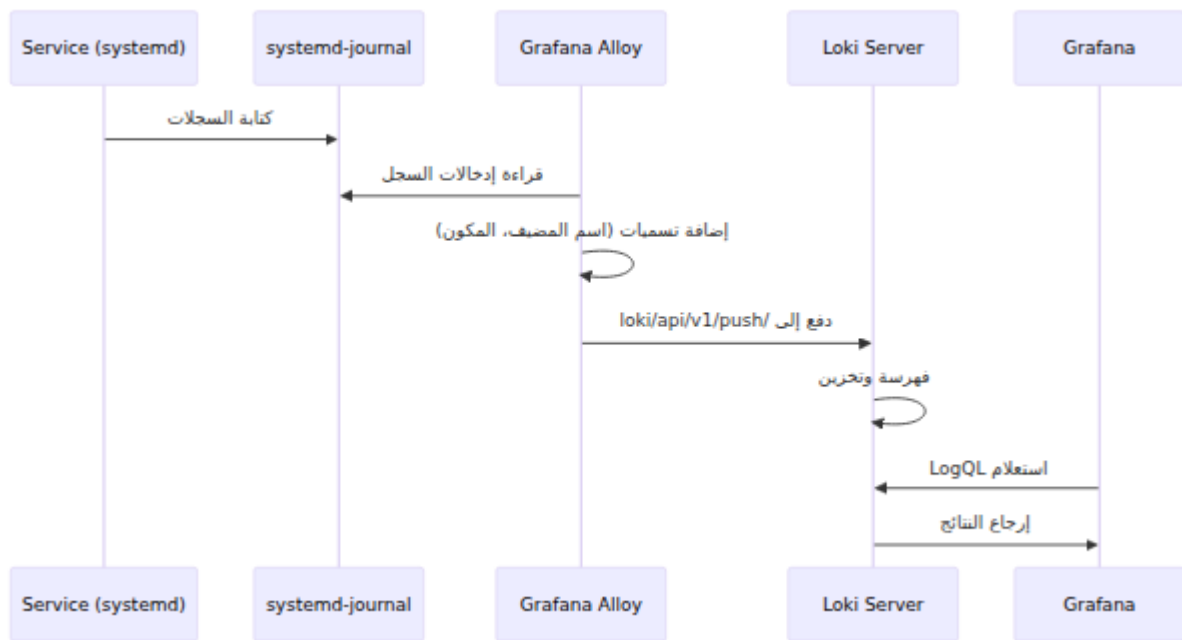
```
sum(rate({component=~"pcscf|scscf|icscf"} |~ "(?i)error" [5m]))
```

JSONRPC لـ CGrates تسجيل

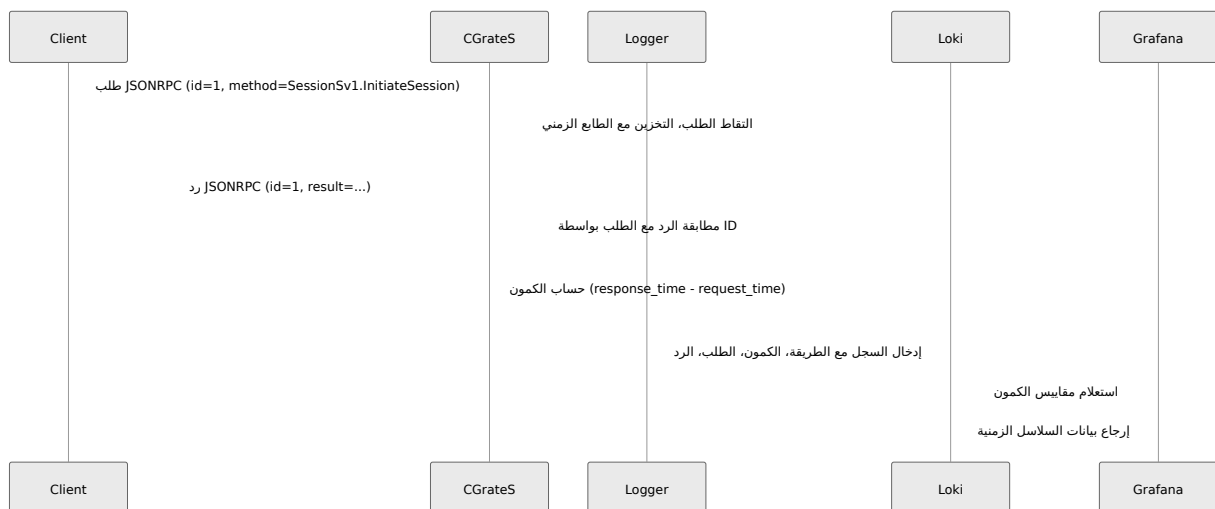
وتسجيلها مع مطابقة الطلب / JSONRPC يتم التقاط حركة مرور OCS/CGrates بالنسبة لنشر الرد وتتبع الكمون لتحليل الأداء واستكشاف الأخطاء وإصلاحها.

ويتم جمع ملف السجل الخاص به `cgrates` في دور JSONRPC يتم تعريف مسجل و `ocs` على كل من مجموعات مضيفي Alloy بواسطة `(/var/log/cgrates/jsonrpc.log)` `cgrates`.

الهيكلة



تدفق الطلب/الرد



ما يتم التقاطه

CGrateS: يتم التقاط حركة المرور على هذه المنافذ لـ

الوصف	البروتوكول	اسم الخدمة	المنفذ
TCP الخام عبر JSONRPC	TCP	rpc_json	2012
تستبعد HTTP واجهة برمجة التطبيقات (/metrics و /health)	HTTP	http	2080

يقوم المسجل تلقائيًا بتصفية:

- Prometheus عمليات سحب مقاييس (GET /metrics)
- طلبات فحص الصحة (GET /health)
- (بيانات ثنائية) Gzip الردود المضغوطة باستخدام

تنسيق السجل

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

```
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	عند الطابع الزمني ISO 8601 استلام الرد	2026-03-01T10:30:45.123456
port	الذي تم الطلب CGrateS منفذ إليه	2012, 2080
service	اسم الخدمة للمنفذ	rpc_json, http
request_id	للمطابقة JSONRPC معرف طلب	1, 42, (فارغ للفراغ)
method	JSONRPC اسم طريقة	SessionSv1.InitiateSession
latency_ms	الكمون في جولة كاملة بالمللي ثانية	2.45
status	حالة النتيجة	ok, error
error	رسالة الخطأ إذا كانت الحالة خطأ	INSUFFICIENT_CREDIT
src	المنفذ (العميل): IP عنوان	10.10.1.50:45678
dst	(CGrateS) المنفذ: IP عنوان	10.10.2.100:2012
request	JSONRPC الحمولة الكاملة لطلب (مضغوط JSON)	{"id":1,"method":"..."}
response	JSONRPC الحمولة الكاملة لرد (مضغوط JSON)	{"id":1,"result":{"..."}}

لوحة Grafana

JSONRPC لوحات مخصصة لتحليل OCS / CGrateS تتضمن لوحة سجلات

حسب الطريقة JSONRPC الكمون لـ

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

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

.يعرض الشريط الأسطوري متوسط وأقصى كمون لكل طريقة

JSONRPC حركة مرور

:مع JSONRPC لوحة سجلات تعرض جميع أزواج الطلب/الرد لـ

- الطابع الزمني
- اسم الطريقة
- الكمون
- الحمولة الكاملة للطلب والرد (منسقة)

تعمق في الطريقة

بواسطة طريقة معينة باستخدام متغير **الطريقة** في أعلى JSONRPC قم بتصفية حركة مرور لرؤية حركة (`SessionSv1.InitiateSession`) ، على سبيل المثال) اللوحة. أدخل اسم الطريقة المرور الخاصة بتلك الطريقة فقط.

CGraTeS لـ RPC لوحة مكالمات

ومطابقة API أدوات مركزة لاستكشاف أخطاء مكالمات CGraTeS لـ RPC تقدم لوحة مكالمات الطلبات مع الردود.

حالة الاستخدام: العثور على مكالمات فاشلة

: "عندما يبلغ المستخدم "حاولت الاتصال بـ 1234 وفشل

1. **CGraTeS لـ RPC** افتح لوحات الم❖❖لومات → السجلات → **مكالمات**
2. في حقل **البحث**، أدخل رقم الهاتف أو معرف الحساب: **1234**
3. اضبط **الحالة** على **خطأ** لرؤية المكالمات الفاشلة فقط.
4. جميع المكالمات المطابقة مع الطلب/الرد الكامل **RPC** تعرض لوحة **بحث مكالمات**.

ما أعاده) يعرض كل إدخال سجل الحمولة الكاملة للطلب (ما تم إرساله) والحمولة الرد
CGrates)، مما يسهل تحديد الخطأ بالضبط.

متغيرات اللوحة

المتغير	الغرض	المثال
البحث	بحث نصي حرفي الحمولة الطلب/الرد	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)

- عدد جميع المكالمات في نطاق الوقت: **RPC إجمالي مكالمات**
- الأخطاء:** عدد المكالمات الفاشلة (ملونة: أخضر=0، أصفر=9-1، أحمر=10+))
- متوسط الكمون:** متوسط الوقت المستغرق (ملون وفقاً للعتبات)
- أقصى كمون:** أعلى مكالمات كمون

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

اللوحة الرئيسية للسجلات تعرض المكالمات المطابقة. انقر على أي إدخال **RPC بحث مكالمات**
الطلب/الرد الكامل JSON للتوسع ورؤية.

الفاشلة: عرض مسبق التصفية يظهر فقط المكالمات **RPC مكالمات** `status=error`.

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

JSONRPC استعلام سجلات

استعلامات أساسية

JSONRPC: جميع حركة مرور

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

تصفية حسب الطريقة:

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

البحث في الحمولة الطلب/الرد:

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

الأخطاء فقط:

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

تحليل الكمون

استخراج الكمون كمعيار (للمرسوم 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 عن الطلبات البطيئة (الكمون < 100 مللي ثانية)

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

استعلامات محددة للطريقة

معالجة CDR:

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

إدارة الجلسات:

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

عمليات الحساب:

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

إدارة الخدمة

OCS/CGRateS. على مضيفي systemd كخدمة JSONRPC يعمل مسجل

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

JSONRPC عدم ظهور سجلات

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

الأسباب المحتملة:

- خدمة المسجل غير قيد التشغيل
- ngrep عدم تثبيت
- 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. ngrep تحقق من تثبيت:

```
which ngrep
```

3. تحقق من كتابة ملف السجل:

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

4. يجمع الملف Alloy تحقق من أن:

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

تظهر لوحة الكمون عدم وجود بيانات

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

الأسباب المحتملة:

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

الحل:

1. تحقق من وجود السجلات لنطاق الوقت:

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

2. موجود في السجلات `latency_ms` تحقق من أن حقل

3. قم بتوسيع نطاق الوقت

عدد الطلبات المعلقة مرتفع

الأعراض: تظهر سجلات التصحيح العديد من الطلبات المخزنة ولكن القليل من الاكتمالات المسجلة

الأسباب المحتملة:

- طلبات بدون ردود (تم فصل العميل)
- تم التقاط الرد قبل الطلب (ترتيب الحزم)
- عدم تطابق معرف الطلب بين الطلب والرد

الحل:

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

سجلات الملفات

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

(خوادم التطبيقات) FreeSWITCH سجلات

المسار	تسمية الوظيفة
<code>/var/log/freeswitch/freeswitch.log</code>	<code>freeswitch</code>
<code>/var/log/omnitas-freeswitch.log</code>	<code>freeswitch-structured</code>

سجلات Nginx (OmniCRM)

المسار	تسمية الوظيفة	تسمية النوع
<code>/var/log/nginx/access.log</code>	nginx	access
<code>/var/log/nginx/error.log</code>	nginx	error

JSONRPC لـ CGrates (OCS)

المسار	تسمية الوظيفة
<code>/var/log/cgrates/jsonrpc.log</code>	cgrates-jsonrpc

استكشاف الأخطاء وإصلاحها

Grafana عدم ظهور السجلات في

Loki الأعراض: لا توجد سجلات مرئية في لوحات

الأسباب المحتملة:

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

الحل:

1. على المضيف المصدر Alloy تحقق من حالة:

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

2. على مضيف المراقبة Loki تحقق من حالة:

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

3. اختبر الاتصال من المصدر إلى المراقبة:

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

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

العالى للقرص Alloy استخدام

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

الأسباب المحتملة:

- غير متاح لفترة طويلة Loki
- امتلاء ذاكرة التخزين المؤقت لـ WAL

الحل:

1. انظر أعلاه) Loki تحقق من اتصال
2. WAL معطلاً، يتم تخزين السجلات محلياً في Loki إذا كان
3. متاحاً، يتم إرسال السجلات المخزنة تلقائياً Loki بمجرد أن يصبح
4. يتم إسقاط القطع المخزنة التي تزيد عن 1 ساعة (حسب التصميم)

Loki امتلاء تخزين

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

الأسباب المحتملة:

- حجم السجل يتجاوز القرص المتاح قبل أن تحذف فترة الاحتفاظ بالسجلات القديمة
- عدم تشغيل ضغط الاحتفاظ

الحل:

1. Loki تحقق من استخدام تخزين:

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

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

فقدان تسميات المكونات

بدلاً من القيمة `infrastructure` هي `component` الأعراس: تظهر السجلات ولكن تسمية المتوقعة

الأسباب المحتملة:

- المضيف ليس في مجموعة الجرد المتوقعة
- بعد تغيير الجرد Alloy لم يتم تجديد تكوين

الحل:

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

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

3. Alloy أعد تشغيل:

```
systemctl restart alloy
```

منافذ الخدمة

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

الوثائق ذات الصلة

- لوحات المعلومات، والتنبيهات، Grafana، Prometheus: **المراقبة والملاحظة**
- معمارية النشر**: الهيكلية العامة للنظام
- تكوين ملف المضيفين**: تكوين الجرد
- مرجع التكوين**: مرجع كامل للمعلومات

النواة المعتمدة على الدوائر (CS)

بواسطة (الصوت والرسائل القصيرة 2G/3G) يتم نشر مجال الدوائر المعتمدة على الدوائر القديمة بواسطة OmniMSC، الموجهة للراديو، ومفتاح BSC يقوم بتوفير `services/cs.yml` playbook. SS7/SIGTRAN الإشارات stack و

نظرة عامة

ثلاثة أدوار، كل منها ضد مجموعة جرد خاصة بها `cs.yml` تشغل

الوصف	الحزم	مجموعة الجرد	الدور	المكون
وحدة التحكم في المحطة الأساسية + بوابة الوسائط للاختبار والتحقق من 2G/3G)	BSC, MGW	<code>bsc</code>	<code>circuit_switched/bsc</code>	BSC
مفتاح مركز التحويل المحمول	<code>omnimsc</code>	<code>msc</code>	<code>circuit_switched/omnimsc</code>	MSC
stack SS7/SIGTRAN (STP, HLR, SMSC, CAMEL, مختبر)	<code>omniss7</code>	<code>ss7</code>	<code>omniss7</code>	SS7

- `name`: Setup MSC
`hosts`: msc
`roles`:
 - {`role`: circuit_switched/omnimsc, `become`: yes}

BSC (circuit_switched/bsc)

للاختبار والتحقق من Omnitouch APT من خادم (MGW) وبوابة الوسائط BSC يقوم بتثبيت الدوائر المعتمدة. يقوم الدور بتكوين كلا الخدمتين.

- من قوالب الدور MGW و BSC يقوم بإنشاء ملفات تكوين.
- التي تعطل **جدولة وحدة المعالجة** MGW لخدمة systemd يقوم بتثبيت تجاوز `CPUSchedulingPolicy` **المركزية في الوقت الحقيقي**. يقوم بإزالة `Priority` و `AmbientCapabilities` في الجدولة الافتراضية. الوقت الحقيقي.
- يقوم المعالجات بإعادة تشغيل كل خدمة. ثم **يتحقق** كل معالج من أن الخدمة تصل إلى `active` (6 ثواني 5 فاصل، محاولات، باستمرار).

MSC (circuit_switched/omnimsc)

NF 5GC: ويقوم بتكوينها مثل `omnimsc` يقوم بتثبيت حزمة

- من `license_server_api_urls` ويستخرج `group_vars_omnicore` يحل `license_server` مجموعة.
- إذا تم تعيين `/etc/omnimsc/` المدمج إلى `runtime.exs.j2` يقوم بإنشاء `group_vars/omnimsc_template_config` فإنه يقوم بإنشاء قالب مخصص من `omnimsc_template_config` بدلاً من ذلك.
- الافتراضي، `omnimsc_cdr_output_dir` CDR يضمن وجود دليل إخراج `/var/lib/omnimsc/cdr`.
- API / موقعة ذاتياً لواجهة الويب TLS يقوم بإنشاء شهادة.
- مع إعادة التشغيل عند التغيير، `runtime.exs` + يستخدم نمط التكوين النسخي.

SS7 (omniss7)

يتم، SS7، ويقوم بإنشاء قالب تكوين واحد لكل دور omniss7 دور واحد يقوم بتثبيت حزمة ss7_role: اختياره بواسطة متغير المضيف:

ss7_role	القالب	الوظيفة
stp	stp.j2	SS7 توجيه) نقطة نقل الإشارة
hlr	hlr.j2	سجل الموقع المنزلي
smsc	ipsmgw.j2	SMSC / IP-SM-GW
camel	camel.j2	CAMEL التحكم في خدمة
tester	tester.j2	وكيل اختبار / تحميل الإشارات

محددات إضافية:

- بدلاً من (M2PA نقطة نهاية) m2pa.j2 يقوم بإنشاء ss7_protocol: m2pa تعيين قالب tester.
- يتجاوز جميع القوالب (group_vars/) مسار تحت) ss7_template_config تعيين المدمجة بتكوين محدد للمضيف.

runtime.exs + نمط التكوين النسخي omniss7 يستخدم Elixir، مثل المكونات الأخرى في /opt/omniss7/releases/*. ويرتبط بالرمز في

الوثائق ذات الصلة

- الخدمة وهيكل النشر playbooks جميع - Service Playbooks
- gc_nf والدور المشترك 5 G النواة المستقلة 5 - 5G Core
- مثل) قوالب التكوين المخصصة - Group Variables Configuration (ss7_template_config, omnimsc_template_config)
- OmniCore تكوين:**
ما يجب أن يتواجد - <https://docs.omnitouch.com.au/docs/repos/OmniCore>
في ملفات تكوين المكونات

مرجع التكوين

نظرة عامة

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

للحصول على وثائق محددة للمنتج، انظر:

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

نهج التكوين

الحديثة نموذج تكوين مبسط OmniCore تستخدم نشرات



المبدأ الرئيسي: يتم تعريف معط  التكوين مباشرة في ملف المضيفين. تتعامل القيم الافتراضية للدور مع العديد من الإعدادات. ومع ذلك، يتم تكوين بعض المكونات حصريًا من هو المثال الملحوظ (انظر أدناه). OmniCRM. وليس لديها قيمة افتراضية للدور group_vars في تلك الحالات، فإن القيمة المفقودة تفشل التشغيل بصوت عالٍ بدلاً من العودة إلى قيمة افتراضية.

ملاحظة أمان: العديد من القيم الافتراضية للدور هي عناصر غير آمنة مخصصة للاستخدام في المختبر فقط ويجب تجاوزها لأي نشر حقيقي. تشمل الأمثلة كلمات مرور قاعدة البيانات APT وبيانات اعتماد مستودع (omnitech2024, sql_pass: password, مثل) قبل group_vars تأكد دائمًا من تعيين أسرار حقيقية وفريدة في جردك أو (omni/omni). النشر.

تخطيط الشبكة

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

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

معلومات المضيف الشائعة

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

علامات الخدمة المحددة

```
cdrs_enabled: True          # تمكين توليد CDR
in_pool: False              # استبعاد من مجموعة التوازن
populate_crm: False         # بالبيانات الأولية CRM ملء
```

(all:vars) المتغيرات العالمية

على إعدادات على مستوى النشر. تستخدم النشرات الحديثة متغيرات `all:vars` يحتوي قسم عالمية قليلة، مع تغطية العديد من الإعدادات بواسطة القيم الافتراضية للدور. لاحظ الاستثناءات حيث يجب تقديم التكوين بشكل صريح لأنه لا توجد قيمة (OmniCRM، Grafana SMTP) أدناه افتراضية للدور.

المتغيرات العالمية الأساسية

المصادقة والوصول

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

بدلاً من كلمات المرور SSH **بديل**: استخدم مفاتيح

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

هوية العميل

```
customer_name_short: omnitouch
customer_legal_name: "Customer Lab"
site_name: Customer-Site
region: AU
TZ: Australia/Melbourne
```

PLMN تكوين

```
plmn_id:
  mcc: '001'          # رمز الدولة المتنقلة (3 أرقام)
  mnc: '01'           # رمز الشبكة المتنقلة (2-3 أرقام)
  mnc_longform: '001' # مزود بالصفير (دائماً 3 أرقام) MNC

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

Diameter **الغرض**: يحدد شبكتك المتنقلة بشكل فريد. يستخدم لبناء مجال

أسماء الشبكات

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

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

DNS تكوين

```
manage_resolv_conf: True # من Ansible لمنع False تعيين إلى إدارة /etc/resolv.conf
```

بكتابة Ansible لن يقوم، `False` إلى `manage_resolv_conf` **ملاحظة**: عندما يتم تعيين DNS على هذا المضيف. هذا مفيد للمضيفين الذين يحتاجون إلى تكوين `/etc/resolv.conf` مخصص أو يتم إدارتهم بواسطة أنظمة خارجية. يمكن تعيينه لكل مضيف في الجرد أو عالميًا في `all:vars`.

APT تكوين مستودع

هو المصدر الوحيد للحقيقة. قم بتعريفه مرة واحدة. يتم اشتقاق متغيرات تحميل `apt_repo` منه تلقائيًا. لا تقوم بتعيينها بشكل `apt_download_base` URL و `remote_apt_*` الثنائيات منفصل في جرد عادي.

مع المضيفين `apt_cache_servers` **القيم الافتراضية التلقائية**: عندما يتم تعريف مجموعة

- ما لم يتم تعيينه صراحة إلى `True` افتراضيًا إلى `use_apt_cache` يتم تعيين `False`)
- لأحد خوادم التخزين IP افتراضيًا إلى عنوان `apt_repo.apt_server` يتم تعيين المؤقت الأولى

```
use_apt_cache: False # True = الوصول المباشر إلى المستودع (التخزين المؤقت المحلي)

apt_repo:
  apt_server: "apt.<region>.omnitou.ch" # FQDN المقدمة من Omnitouch (للتخزين المؤقت IP أو عنوان)
  apt_repo_username: "your-username" # مطلوب للوصول المباشر
  apt_repo_password: "your-password" # مطلوب للوصول المباشر
  # apt_repo_port: 443 # اختياري؛ الافتراضي هو 443
  (HTTPS).
  # -> يتبع المخطط المنفذ: 443
  http, وإلا https.
```

في وضع **8080 / HTTP** للوصول المباشر و **443 / HTTPS** المخطط والمنفذ الافتراضيان هما يقوم بتعيين (`apt_cache_servers` group) التخزين المؤقت. فقط خادم التخزين المؤقت العلوي الذي يتزامن منه Omnitouch يدويًا، لوصف خادم `remote_apt_*`.

APT انظر: نظام تخزين

خادم الترخيص

```
# في الجرد عند عدم `license_server` اختياري: يتم اشتقاقه تلقائيًا من مجموعة #  
# تعيينه.  
# https://<ansible_host>:8443/api. يصبح كل مضيف في تلك المجموعة  
license_server_api_urls: ["https://10.10.2.150:8443/api"]  
license_enforced: false # القيمة الافتراضية الفعالة هي  
(roles/omnihss/defaults/main.yml)
```

موجودة في الجرد، يتم اشتقاق `license_server` **ملاحظة:** إذا كانت مجموعة تلقائيًا من أعضائها؛ تحتاج فقط إلى تعيينها يدويًا لتجاوز هذا `license_server_api_urls` السلوك.

انظر: خادم الترخيص

MME إعدادات

باستخدام طريقة قابلة للتكوين، يتم تعيينها لكل بوابة تحت P-GW و S-GW أقران MME يختار `home/roaming` يمكن أن تكون الطريقة قيمة واحدة أو مقسمة حسب حركة المرور. `omnimme`. اختيار من الأقران `random_peer` و (DNS حل الأقران عبر) `dns_lookup` القيم الصالحة هي (المكونة).

```
omnimme:  
  sgw:  
    selection_method: random_peer # ...قيمة واحدة، أو  
    # selection_method:  
    #   home: random_peer  
    #   roaming: dns_lookup  
  pgw:  
    selection_method:  
      home: random_peer  
      roaming: dns_lookup
```

AMF إعدادات

```
# هنا يتم TAC كل AMF. J supported_ta_list تقود أيضًا tac_list قائمة
# يتعين AMF إذا لم يتم تعيينه، فإن NG. المتصلة أثناء إعداد gNB الإعلان عنه للأجهزة
# TAC 1 لا يثبت gNB فقط، مما يفرض أي TAC 1 على
tac_list: [1, 3]

# omniamf لأن true الافتراضي هو (NEA0 فرض) NAS تعطيل تشفير
# المشفر في الاتجاه NAS ولكن لا يمكنه فك تشفير NEA2 يتفاوض على >=20260727
# الصاعد (الوضع الأمني)
# false تعيين إلى G. مكتمل والرسائل اللاحقة)، مما يتسبب في فشل كل تسجيل 5
# في الاتجاه الصاعد يعمل NEA2 بمجرد أن يؤكد البائع أن فك تشفير
# (NIA2) NAS الحقيقية. لا تتأثر سلامة UEs ضد
amf_force_null_cipherring: true
```

SAEGW إعدادات

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

OmniHSS إعدادات

mme و dra تلقائيًا من الجرد. تستخدم مجموعات Diameter بناء قائمة الأقران OmniHSS تقوم أي منها يتم استخدامها يعتمد على) applicationserver و pcscf و icscf و scscf و pgwc و packet_core_dra_support / ims_dra_support). تقوم exclude_diameter_peers بإسقاط المضيفين الفرديين من تلك القائمة الناتجة، للأقران الموجودة في الجرد ولكن لا ينبغي أن Diameter J HSS. تكون أقران

```
omnihss:
  exclude_diameter_peers: # (لا شيء)
    - customer-as01
    - customer-as02
```

- Diameter IP أو FQDN وليس `inventory_hostname` يتم مطابقة الإدخالات ضد
- ينطبق الاستبعاد على كل مجموعة تم اكتشافها تلقائيًا، لذا تغطي قائمة واحدة مضيقة. بغض النظر عن المجموعة التي يجلس فيها
- يجب أن توجد الأسماء في الجرد. تؤكد الدور ذلك قبل النشر، لذا فإن خطأ مطبعي `/etc/omnihss/runtime.exs` يفشل التشغيل بصوت عالٍ بدلاً من ترك النظر في

بصمت.

- `vars:` لكل مضيف، أو تحت `omnihss` قم بتعيينه حيث يتم تعيين باقي قاموس `vars:` يستبدل بدلاً من دمج القواميس، يجب أن تعيش في Ansible نظرًا لأن (لمجموعة مثل المفاتيح الأخرى `omnihss` نفس كتلة).
- تحت SIP S-CSCF فقط. قائمة أقران Diameter يؤثر ذلك على قائمة الأقران `config: hss, ims:` غير ذات صلة ولا يتم تصفيتها بواسطة هذه المفتاح.

على مضيف يزيله من `dra_peer_exclude: True` على آلية خاصة به: تعيين DRA يحتوي أو DRA لا يستبعده من HSS الاثنين مستقلان. استبعاد مضيف من OmniDRA قائمة أقران العكس.

إعدادات OmniCRM

لا توجد (`hosts/` تحت) للعميل `group_vars` حصريًا من OmniCRM يتم الحصول على تكوين قيم افتراضية مدمجة في الدور لأسرار الموقع.

تثبيت حزم: (القيمة الافتراضية للدور `omnicrm_deploy_mode: package`) المسار المفضل إلى `crm_config.yaml` قالب، `apt_repo` العامة من `omnicrm-ui` / `omnicrm-api` `deployment_assets/`، `systemctl` مزامنة، `/etc/omnicrm/crm_config.yaml` `restart omnicrm-api` من كتلة UI يأتي علامة. `ui: (runtime GET /crm/public/ui-config)` على المضيف ولا شيء من `yarn` يحتاج هذا المسار إلى لا شيء من بناء. `crm_env_override`.

`yarn:` بيئة المضيف + بناء + git أرشيف (`omnicrm_deploy_mode: source`) المسار القديم `crm_env_override` مطلوب لهذا الوضع فقط.

```
omnicrm_deploy_mode: package      # package (default) | source
omnicrm_package_version: ""       # apt لـ omnicrm-
api/ui                             # اختياري تثبيت إصدار
crm_config.yaml                   # API + ui: config (group_vars;
(مطلوب)                           # رموز اختيارية لـ
deployment_assets/                # شعارات / رموز اختيارية لـ
/etc/omnicrm/deployment_assets     # مطلوب فقط
crm_env_override: my-site.env      # وضع المصدر / النسخة
الاحتياطية المحلية للواجهة
cell_sites: cellSites.txt          # اختياري: بيانات موقع الخلية للواجهة
message_templates: constants.js    # اختياري: قوالب رسائل البث الخلوي
site_data: site_data.json          # اختياري: حدود جغرافية لبث الخلية
```

إعدادات Grafana SMTP

و `grafana_smtp_user` Grafana. تستخدمها دور المراقبة لإرسال بريد إلكتروني لتنبيه **مطلوبان** (لا توجد قيمة افتراضية: يجب تعيينهما): `grafana_smtp_password` انظر. `in-v3.mailjet.com:587` الافتراضي هو `grafana_smtp_host`. `roles/monitoring/templates/grafana/grafana.ini`.

```
grafana_smtp_user: "smtp-user"           # مطلوب، لا توجد قيمة افتراضية
grafana_smtp_password: "smtp-password"   # مطلوب، لا توجد قيمة افتراضية
grafana_smtp_host: "in-v3.mailjet.com:587" # اختياري، هذا هو الافتراضي
```

إعدادات IMS

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

RAN تكوين مراقب

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

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

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

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

تكوين جدار الحماية

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

للتجوال DNS خوادم

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

SSH مفاتيح المستخدمين المحليون

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

Hypervisor تكوين

Proxmox

عبر `proxmoxServers` يتم اختيارها لكل إدخال LXC، أو حاويات VMs يتم نشر الموقع كـ `deployment_type` (القيمة الافتراضية `vm`). يجب أن تتفق جميع الإدخالات في موقع واحد؛. للحصول على الشرح الكامل **Proxmox** الخلط يفشل التحقق. انظر **نشر**

VM موقع

```
proxmoxServers:
  customer-prxm01:
    # الافتراضي هو → deployment_type تم حذف "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 المستخدم الاختياري لـ
    local_users: cloud-init # الافتراضي هو أول مف 100 ح
    proxmoxTemplatePassword: omnitouch # كلمة المرور الاختيارية لـ
    cloud-init: cloud-init # الافتراضي هو اسم مستخدم cloud-init
    storage: SSD_RAID0 # VM اختياري، التخزين الافتراضي لـ
```

LXC موقع

```

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

proxmoxServers:
  customer-prxm01:
    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 لـ
rootfs

```

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 # فقط LXC اختياري
    proxmoxLxcMemoryMb: 4096 # فقط LXC اختياري
    proxmoxLxcDiskSizeGb: 30 # فقط LXC اختياري
    proxmoxLxcRootFsStorageName: SSD_RAID0 # فقط LXC اختياري
    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

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

وثائق المنتج

للحصول على أدلة تشغيل مفصلة وتكوين متقدم:

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

- **مكونات OmniCall:**

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

- **OmniCharge/OmniCRM:**

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

نظرة عامة على بنية النشر

نظرة عامة

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

للحصول على إرشادات تفصيلية حول موضع المكونات، وإرشادات تخصيص IP راجع [معيّار تخطيط](#) العام IP والتعامل مع IP، عنوان

مثال كامل على النشر

توفير البنية التحتية (اختياري) 0.

قبل التكوين VMS/LXCS قم بتوفير Proxmox، بالنسبة لنشر

```
# نوع النشر لكل مضيف يحدد Proxmox على LXC و/أو حاويات VMS نشر #  
(أيهما  
ansible-playbook -i hosts/Custommer/hosts.yml  
util_playbooks/proxmox.yml
```

[Proxmox على VM/LXC](#) راجع: نشر

1. تعريف البنية التحتية (ملف المضيفين)

```
# 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)

أي تجاوزات تكوين مطلوبة على مستوى المضيف أو الموقع أو الشبكة `group_vars` يخزن مجلد

الخاص بك. تعيش خطوط OMniMessage SMS Sc على سبيل المثال، يحتوي مجلد على تكوين الخاص بك يعيش هنا أيضًا Diameter هنا. كل منطق توجيه TAS التي يتصل بها SIP

فقط. يتم استنساخ مصدره من `group_vars` هنا أيضًا. إعداداته خاصة بـ OMniCRM يتم تكوين باستخدام بيانات اعتماد (Ansible (`delegate_to: localhost`) على وحدة التحكم GitHub وحدة التحكم الخاصة بها قبل مزامنتها مع الهدف، بحيث لا تهبط أي بيانات اعتماد على المضيف الهدف.

راجع: تكوين المتغيرات الجماعية

3. توزيع الحزم (APT Cache)

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

APT Cache راجع: نظام

4. تكوين الترخيص

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

راجع: خادم الترخيص

5. تنفيذ النشر

تتعامل `services/twag.yml` يمكنك نشر مكونات فردية. على سبيل المثال، قم بتشغيل `--limit=myhost` أو مع كل شيء. استخدم `services/all.yml` مجموعة التشغيل `limit=mmee,sgw` لتحديد المضيفين الذين تعمل 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
```

نماذج الن??ر

.على وحدة التحكم وتشغيلها OmniCore هناك طريقتان للحصول على مجموعة أدوات

استنساخ المستودع على وحدة التحكم A: النموذج

الخاصة بك، احتفظ Ansible النموذج التقليدي: استنساخ هذا المستودع على وحدة التحكم وقم بتشغيل مجموعات التشغيل مباشرة من النسخة المستنسخة. هذا، `hosts/` بمخزونك تحت هو النموذج المفترض في مثال النشر أعلاه.

ذاتيًا `.deb` omniconore تثبيت B: النموذج

بحيث يمكن لوحدة التحكم، `omniconore` ذاتية الاحتواء Debian أيضًا كحزمة OmniCore تم تعبئة بدلاً من استنساخ المستودع `apt` تثبيت مجموعة الأدوات من

- `packaging/nfpm.yaml` انظر، `(name: omniconore)` **ما الذي يتم تثبيته.** الحزمة ومترجم، `vendored` ومجموعات التشغيل، ومجموعات، `Ansible` تحتوي على أدوار `Python` مضمن `(/opt/omniconore/python)` تقوم خطوة ما بعد التثبيت. `/opt/omniconore` إلى بإنشاء بيئة افتراضية ذاتية الاحتواء `(packaging/postinstall.sh)` `ansible*` من المترجم المضمن + بيت العجلات وترتبط `(/opt/omniconore/venv)` بحيث تقوم وحدة التحكم بتشغيل النسخ، `/usr/local/bin` عبر `PATH` المضمنة بـ من `Ansible` الخاصة بها: لا `Python` ومكتبات `Ansible` من `uv.lock` الدقيقة من من النظام، لا شبكة. يتم سحب الأدوات النظامية الحقيقية فقط `Python` التوزيع، لا وتستهدف **amd64** الحزمة هي `apt`. بواسطة `(git, rsync, openssh-client)` **Ubuntu 22.04+** (`glibc ≥ 2.35`).
- بتهيئة المستودع إلى `packaging/build.sh` **كيف يتم بناؤها.** يقوم `(/opt/omniconore)`، `hosts/`، `.git/`، `requirements.yaml` من `Ansible` يورد مجموعات يقوم بتشغيل فحص سري مغلق، ثم يبني `Python` العجلات من الاعتماديات المقفلة لـ `nfpm` باستخدام `.deb`.
- بتشغيل `github/workflows/build-deb.yaml` **كيف يتم نشرها.** يقوم `_metadata.json` (بالإضافة إلى) `.deb` وينشر `main` في كل دفع إلى `build.sh` الخاص بخادم `discover_github_debs.py` يقوم برنامج `GitHub` كإصدار (كملحق لا حاجة `aptly`. ويضيفه إلى `Omnitouch` باكتشاف هذا الأصل الإصدار عبر منظمة `apt` `apt`. لتغييرات في دور

تدفق العمل:

```
# 1. تثبيت مجموعة الأدوات من apt
apt-get install omnicore

# 2. توفير مخزونك الخاص
# (لا تحتوي الحزمة على مخزون - إنه يعيش خارج)
vi /etc/omnicore/inventory

# 3. تشغيل مجموعات التشغيل من مجموعة الأدوات المثبتة
cd /opt/omnicore
ansible-playbook -i /etc/omnicore/inventory services/all.yml
```

منفصل عن مجموعة الأدوات (/etc/omnicore/inventory) نظرًا لأن المخزون يعيش في /opt/omnicore) المثبتة، فإن ترقية الحزمة لا تلمس أبدًا تعريف البنية التحتية أو التخصيص.

الوثائق ذات الصلة

- البدء - **Ansible** مقدمة في نشر
- **مجموعات تشغيل الخدمة** - مرجع مجموعة التشغيل والتسلسل الهرمي
- **تكوين ملف المضيفين** - تعريف البنية التحتية
- **IP بنية الشبكة وتخصيص** - IP معيار تخطيط
- **تكوين المتغيرات الجماعية** - التخصيص
- **إدارة الحزم** - **APT Cache** نظام
- **خادم الترخيص** - إدارة الترخيص
- **التنبيهات، ولوحات المعلومات، Prometheus، Grafana** - **المراقبة والرؤية**
- **Alloy وLoki التسجيل المركزي** - جمع سجلات

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.omnitech.com.au/docs/repos/OmniCall>
 - OmniTAS, OmniCall CSCF, OmniMessage, OmniSS7, VisualVoicemail
- **OmniCharge/OmniCRM** (الفوترة):
<https://docs.omnitech.com.au/docs/repos/OmniCharge>
- **الوثائق الرئيسية:** <https://docs.omnitech.com.au/>

جدار الحماية حسب الدور

مستمدة `iptables` و `ip6tables` يوفر جدار الحماية حسب الدور لكل مضيف مجموعة قواعد من نموذج إعلاني واحد لحركة مرور الشبكة، بدلاً من قواعد مكتوبة يدوياً لكل مضيف. يصف ملف واحد كل تدفق مسموح به في النواة (من يمكنه الوصول إلى أي دور، وعلى أي بروتوكول ومنفذ، ولماذا). من هذا النموذج، تنتج المنصة شيئين يتفقان دائماً مع بعضهما البعض:

- **مجموعة القواعد المفروضة** المطبقة على كل مضيف، و
- للمراجعة ولتسليمها إلى جدار حماية أعلى (CSV) **مصفوفة الاتصالات / المنافذ**

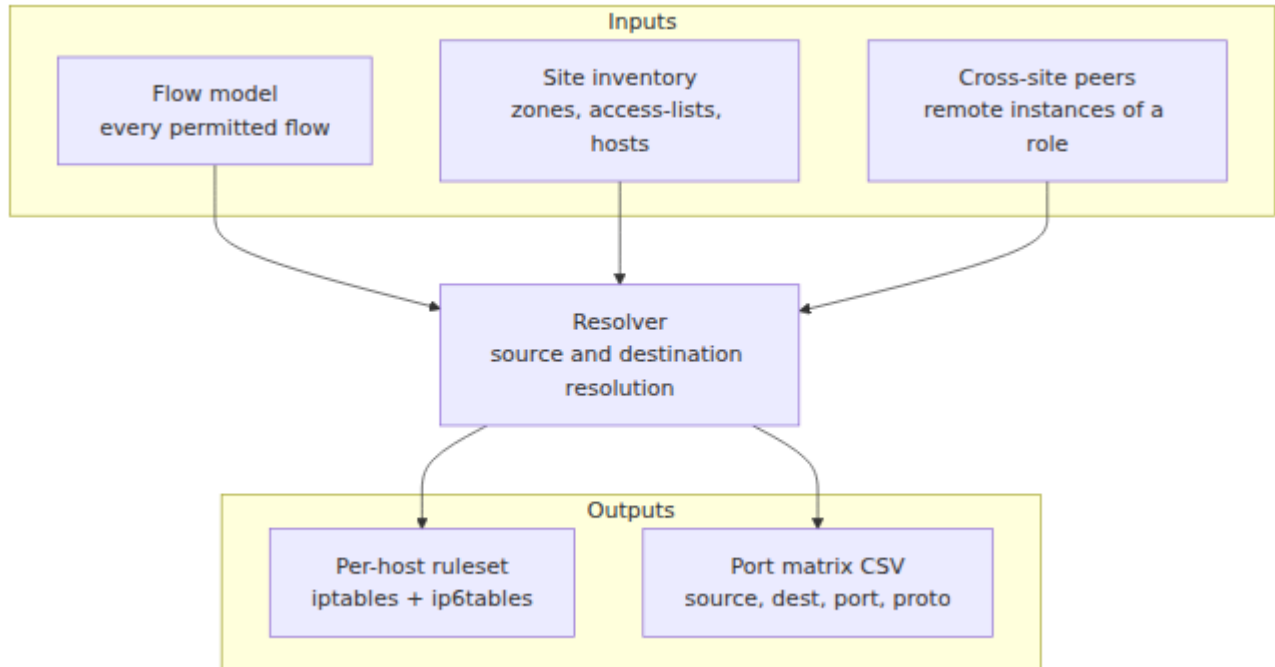
تعمل القواعد في أحد وضعين: **مراقبة** (تسجيل كل تدفق، عدم إسقاط أي شيء) أو **فرض** (افتراضي-الرفض). الحجب متعمد وآمن، وكل تطبيق محمي من خلال استعادة تلقائية على المضيف حتى لا يمكن لمجموعة قواعد خاطئة أن تقفل المشغلين.

جدول المحتويات

- نظرة عامة على المعمارية
- نموذج التدفق
- أنواع المصادر
- فهرس الخدمات
- تكوين الموقع
- أوضاع التشغيل
- الأقران عبر المواقع
- حل العناوين
- مصفوفة الاتصالات / المنافذ
- حماية من الإغلاق
- والدعم المزدوج IPv6
- التسجيل
- تحذيرات تشغيلية

نظرة عامة على المعمارية

نموذج واحد بالإضافة إلى تغذية جرد كل موقع تغذي محلاً واحداً. ينتج المحلل كل من القواعد على المضيف ومصفوفة المنافذ، لذا لا يمكن أن تصف الوثائق شيئاً مختلفاً عما يتم فرضه فعلياً.



:يتم تسجيل الحركة المروية التي تم إسقاطها وتدقيقها من خلال خط أنابيب الرؤية القياسي

iptables NFLOG

ulogd2

journald

Alloy

Loki

نموذج التدفق

نموذج التدفق هو المصدر الوحيد للحقيقة حول من يمكنه الوصول إلى ماذا. يحتوي على جزئين: مجموعة من الافتراضات الهيكلية (التي لا تحمل أي عنوان موقع)، وقائمة التدفقات. لا تقول التدفق شيئاً عن أرقام المنافذ مباشرة - بل تسمى خدمة مكشوفة من قبل الدور الوجهة (انظر [فهرس الخدمات](#))، ويتم البحث عن المنفذ من إعلان ذلك الدور

```
- id: '13'                                # stable label, appears in each
rule's comment
  dst: hss                                # inventory group the rule is
applied to
  service: diameter                       # named service exposed by the
hss role (-> 3868/tcp+sctp)
  purpose: 'Diameter'                     # description, shown in the
matrix
  sources: [{ zone: internal }]           # who may reach it (list of
source tokens)
```

الوصف	الافتراضي	مطلوب	النوع	المعامل
معرف ثابت للتدفق. يظهر في تعليق كل قاعدة تم إنشاؤها كـ <code>ans:<dst>:<id></code> بحيث يمكن تتبع قاعدة حية إلى النموذج.	-	نعم	سلسلة	<code>id</code>
مجموعة الجرد التي تقع عليها <code>all</code> القاعدة، أو الوجهات الزائفة <code>five_g_sbi</code> و (كل مضيف) <code>G</code> مجموعات وظائف الشبكة (5). ينتج التدفق الذي لا يتم نشر مجموعة وجهته في موقع ما لا قواعد ولا صفوف في المصفوفة هناك.	-	نعم	سلسلة	<code>dst</code>
خدمة مسماة تم إعلانها من قبل الدور الوجهة (انظر فهرس الخدمات). يتم البحث عن المنفذ (المنافذ) من إعلان الدور، لذا لا تعيد التدفق أبداً ذكر رقم منفذ.	-	*نعم	سلسلة	<code>service</code>
<code>service</code> ، البديل الحرفي لـ للتدفقات غير المرتبطة بدور واحد <code>all-host</code> على سبيل المثال تدفقات كل إدخال هو. <code>PORT/proto (tcp, udp, sctp</code> قد يكون المنفذ نطاق <code>LO-HI</code> أو بروتوكول عاري <code>(icmp, esp, proto 89)</code> .	-	*نعم	قائمة	<code>proto_ports</code>
وصف إنساني للتدفق. يظهر في من مصفوفة <code>service</code> عمود المنافذ.	-	نعم	سلسلة	<code>purpose</code>
المصادر المسموح بها، كقائمة من رموز المصدر. انظر أنواع	-	نعم	قائمة	<code>sources</code>

الوصف	الافتراضي	مطلوب	النوع	المعامل
المصادر.				

وليس كلاهما ، `proto_ports` أو `service` يوفر التدفق إما *

أنواع المصادر

للتدفق هو رمز ذو مفتاح واحد يتم حله ضد تكوين الموقع وجردها. `sources` كل إدخال في قائمة لا يساهم الرمز الذي لا يتم حله إلى أي شيء (على سبيل المثال قائمة وصول فارغة) في أي قاعدة: لا يتم إصداره أبداً كعنصر نائب

الرمز	يحل إلى
<code>{ zone: <name> }</code>	<code>(firewall.zones.<name>)</code> من تلك المنطقة الطائرة CIDRs.
<code>{ list: <name> }</code>	قائمة وصول المشغل، على سبيل المثال <code>allowed_oam_subnets</code> .
<code>{ group: <name> }</code>	عناوين مضيف مجموعة الجرد، بالإضافة إلى أقران ذلك الدور عبر المواقع. انظر الأقران عبر المواقع .
<code>{ ue_pools: true }</code>	<code>(firewall.ue_pools)</code> المشتركين في الموقع / UE مجموعات
<code>{ dra_peers: true }</code>	DRA. كل عنوان تم تكوينه في قائمة الأقران الخاصة بمضيف
<code>{ roaming: true }</code>	حالياً أي مصدر: انظر. (IR.21 / GRX-IPX) نطاقات التجوال. تحذيرات تشغيلية .
<code>{ literal: "..."} </code>	مصدر نصي حر غير مشتق من الجرد

فهرس الخدمات

أرقام المنافذ **تعود للدور الذي يفتحها**، وليس للنموذج المركزي. كل دور يكشف عن المنافذ في دليل دوره، مفاتيحاً بمجموعات الجرد التي يخدمها `firewall_services.yml` يشحن ملف هذا يحافظ على تعريف المنفذ بجوار. (يخدم عدد **❖❖** مجموعات، لذا يعلن كل منها `cscf` دور مثل) الكود الذي يفتحه:

```
# roles/omnihss/firewall_services.yml
firewall_services:
  hss:
    diameter: [3868/tcp, 3868/sctp] # S6a / Cx-Dx / Sh
    api_oam: 8080/tcp # HSS API (OAM)
    api_nf: 8443/tcp # HSS API (queried by other NFs)
    postgres: 5432/tcp # logical replication between HSS peers
```

لكل دور في فهرس واحد. يتم البحث عن `firewall_services.yml` يجمع المحلل كل لذلك التدفق، لذا يعبر النموذج فقط عن **العلاقة** `dst` للتدفق في الفهرس لمجموعة `service:` بينما **المنفذ** يعيش مع الدور. أضف أو (`hss`) الخاصة بـ `diameter` من يمكنه الوصول إلى خدمة الخاصة به؛ لا يتغير شيء في `firewall_services.yml` انقل مستمعاً في دور وقم بتحديث النموذج المركزي.

تحتوي هذه الانقسام على شبكة أمان تفتقر إليها النموذج المركزي فقط: **فحص الانحراف** يقارن ما تعلنه الأدوار ضد ما يستمع إليه المضيفون فعلياً.

```
ansible-playbook -i hosts/<Deployment>/host_files/<Site>.yaml \
  util_playbooks/firewall_services_audit.yaml
```

لكل مضيف، يتم الإبلاغ عن:

- إما إضافته إلى) **غير معلن** - مستمع حي على منفذ لم يعلنه أي دور و، (للدور، أو أنه مستمع غير متوقع `firewall_services.yml`
- **غير مستمع** - خدمة معلنه ليست فعلياً قيد التشغيل.

يتم تخطي المضيفون الذين لا تعلن مجموعاتهم عن أي خدمات، لذا فإن الفحص مفيد أثناء ترحيل الأدوار إلى الفهرس تدريجياً بدلاً من جميعها دفعة واحدة.

منافذه MME **المستمعون الشرطيون**. يتم فتح بعض المنافذ فقط عندما يتم نشر دور آخر: يفتح في `omnilcs` فقط عندما يكون مضيف (`9082/sctp` SLs، `29168/sctp` LCS (SBc-AP، `firewall_services.yml` الجرد. لا يمكن أن تعبر الإعلانات الثابتة عن هذا الشرط، لذا يتم الإشارة إلى مثل هذه المنافذ في الدور ولكن لا يتم إعلانها. في موقع يقوم بتشغيل الدور المعتمد، يتم `firewall_services.yml` الإبلاغ عنها كغير معلنة، وهو ما يعد تحذيراً صحيحاً للمراجعة والسماح بها.

تكوين الموقع

في جرده. لا توجد بدائل للعناوين: يجب `all.vars` تحت `firewall` يعلن كل موقع عن كتلة على الموقع وصف شبكته الخاصة. أي شيء لا يعلنه الموقع يكون ببساطة فارغاً، ولا يتم وراثته أي عنوان من بيئة أخرى.

المثال أدناه مكتمل تماماً بحيث يظهر كل حقل قيمة تمثيلية. تستخدم العناوين نطاقات توثيق كل طائفة هي شبكة فرعية خاصة بها ضمن الشبكة الفائقة) وتخطيط متماسك لكل طائفة وهو ما يجعل المناطق مفيدة: فهي تسمح لتدفق واحد بالسماح، على سبيل المثال، (`internal`)، بدلاً من النواة بالكامل IMS فقط لطائفة.

```

firewall:
  allowed_oam_subnets:      ['10.8.0.0/24']      #
management source (SSH, UI, scrape, mgmt APIs)
  allowed_ran_subnets:      ['10.20.0.0/16']      # RAN
signalling source (SIAP, NGAP, GTP-U, Abis)
  allowed_ue_voice_subnets:  ['100.64.0.0/16']     # external
UE SIP into the P-CSCF
  allowed_carrier_voice_subnets: ['203.0.113.0/24'] # carrier
interconnect SIP
  zones:                                # plane
zones: each plane's real subnet(s)
  internal:      ['10.8.0.0/16']      # trusted
core supernet
  mobile_core:   ['10.8.1.0/24']      # mobile
core NFs (EPC + 5GC + CS core)
  core_svc:      ['10.8.2.0/24']      # core
services plane (HSS, DRA, OCS, OAM)
  ims:           ['10.8.3.0/24']      # IMS
plane (CSCF, TAS)
  ue_svc:        ['10.8.4.0/24']      # UE-
reachable plane (P-CSCF, DNS, ePDG)
  core_v6:       ['2001:db8:0:1::/64'] # IPv6
addresses of core NFs (not UEs)
  ue_pools:      ['100.64.0.0/16', '2001:db8:64::/48'] # UE /
subscriber IP ranges (v4 and v6)
  extra_rules:                                # manual
accept rules (see below)
  - { proto: tcp, dport: '8443', source: '203.0.113.10/32',
comment: 'partner API' }

```

هو إلزامي. يمكن ترك كل قائمة أخرى فارغة ([]) في موقع `allowed_oam_subnets` فقط وموقع بدون نواة، `allowed_ran_subnets: []` يحدد RAN يفتقر إلى تلك الوظيفة: موقع بدون تنتج القائمة. `ue_pools: []` يحدد UE وموقع بدون مجموعات، `core_v6: []` يحدد IPv6 (أدناه Required انظر عمود) الفارغة ببساطة عدم وجود قواعد للتدفقات التي تستخدمها.

الوصف	الافتراضي	مطلوب	النوع	المعامل
الشبكات رعية لمصدر إدارة. تحكم واجهة، SSH، OmniWeb جمع Promethe وواجهات برمجة التطبيقات الإدارية. إذا كانت فارغة، وقف المولد يع خطأ، لأن قيمة فارغة تترك الإدارة بدون مصدر مسموح به سبب إغلاقاً تحت فرض	-	نعم	قائمة	allowed_oam_subnets
الشبكات رعية لمصدر الإشارة RA (S1AP، NGAP، GT U، Abis). كن أن تكون فارغة في موقع بدون RAN.	[]	لا	قائمة	allowed_ran_subnets
الشبكات رعية لمصدر رجي لـ SIP	[]	لا	قائمة	allowed_ue_voice_subnets

الوصف	الافتراضي	مطلوب	النوع	المعامل
P- إلى UE CSCF.				
الشبكات رعية لمصدر لتواصل SIP ن الشركات	[]	لا	قائمة	allowed_carrier_voice_subnets
مناطق لمئات التي بط كل اسم ثرة بشبكته الفرعية قيقية. انظر معاملات المنطقة.	-	نعم	خريطة	zones
IP نطاقات UE / المشاركين. باردة إذا لم يكن لدى الموقع أي منها.	[]	لا	قائمة	ue_pools
قواعد قبول دوية مطابقة على كل ضيف. انظر معاملات القاعدة اليدوية.	[]	لا	قائمة	extra_rules

معاملات المنطقة

قم بتعيين الطائرات التي يستخدمها الموقع فقط؛ CIDRs. تربط كل منطقة اسم طائرة بقائمة من اترك الباقي فارغاً.

الوصف	المنطقة
الشبكة الفائقة الأساسية الموثوقة. أوسع مصدر إشارة داخلي.	internal
EPC (MME، S/P-GW، UPF)، 5GC (AMF، SMF، وشبكة SBI)، وCS core (MSC، SS7/SIGTRAN).	mobile_core
طائرة IMS (I-CSCF، S-CSCF، TAS).	ims
طائرة UE (P-CSCF، DNS، ePDG).	ue_svc
طائرة خدمات النواة (HSS، DRA، OCS، OAM).	core_svc
IPv6 فارغة إذا لم يكن لدى الموقع نواة IPv6 عناوين خدمات النواة.	core_v6

معاملات القاعدة اليدوية

لكل مضيف firewall_extra_rules و (عالمية، لجميع المضيفين) extra_rules تضيف الذي يحتوي على : على source قواعد قبول خام للحالات التي لا يغطيها النموذج. يتم تطبيق IPv6.

الوصف	الافتراضي	مطلوب	النوع	المعامل
tcp، البروتوكول، على سبيل المثال udp، sctp.	-	لا	سلسلة	proto
منفذ الوجهة أو النطاق	-	لا	سلسلة	dport
المصدر. يتم تطبيق قاعدة على CIDR : في القيمة التي تحتوي على IPv6	-	لا	سلسلة	source
تسمية مسجلة في تعليق القاعدة	override	لا	سلسلة	comment

أوضاع التشغيل

firewall_mode. يتم التحكم في وضع التطبيق بواسطة

السلوك	الوضع
ويتم تسجيل ACCEPT على INPUT يتم تثبيت كل قاعدة قبول، تظل سياسة ثم السماح بها. لا يتم إسقاط أي (FW-AUDIT ك) حركة المرور المتساقطة شيء. استخدم هذا للتحقق من النموذج مقابل حركة المرور الحقيقية قبل فرضه	monitor
ثم إسقاطها، ويتم تعيين (FW-DROP ك) يتم تسجيل حركة المرور المتساقطة DROP إلى FORWARD و INPUT سياسات	enforce

تسقط حركة المرور. أي قيمة **enforce** الحجب هو اختيار آمن: فقط القيمة الدقيقة أخرى، بما في ذلك متغير غير مضبوط أو خطأ مطبعي، يتم التعامل معها ك مراقبة، لذا لا يمكن لمضيف أن يبدأ في إسقاط حركة المرور عن طريق الخطأ. هذا ينطبق على ثلاثة مستويات: في جردها، firewall_mode: monitor تعلن المواقع عن monitor الافتراضي للمنصة هو ك مراقبة. يتطلب فرض تغييراً متعمداً enforce ويعالج المولد أي شيء ليس

```
all:
  vars:
    firewall_mode: monitor # change to 'enforce' only after the
audit log confirms coverage
```

أو تجاوزها لكل مجموعة أو لكل `all.vars` لكل موقع تحت `firewall_mode` يمكن تعيين مضيف من خلال تسلسل المتغيرات العادي.

الأقران عبر المواقع

بإدراج (HSS، OCS، تكرار، CSCF، TAS، OmniMessage) تقوم الأدوار التي تتحد عبر المواقع على سبيل المثال) أقرانها البعيدة في ملف الأقران الخاص بالموقع `{ group: <role> }` يتم دمج تلك العناوين البعيدة في حل `prod_master_peers.yml`، لذا فإن التدفق الذي يسمح بدور يغطي تلقائياً كل من الحالات المحلية وعبر المواقع له تحت سياسة واحدة متطابقة. يتم استخدام نفس الحل للقواعد ولمصفوفة المنافذ، لذا تظهر الأقران في كلاهما.

حل العناوين

يتم حل المضيف إلى **كل عنوان تم الإعلان عنه معه**، وليس فقط إلى عنوانه الرئيسي. هذا في الجرد. `secondary_ips.<name>.ip_address` بالإضافة إلى كل `ansible_host` يعني على UPF الثانوية: يجب NIC تربط العقدة الخاصة بمسار البيانات وعقد الإشارة الخدمات ببطاقات على عنوانه العام. تغطي كل من SIGTRAN ويبدأ STP ويتلقى `n3` على عنوانه N3 GTP-U القواعد والمصفوفة كل تلك العناوين، كمصادر وكوجهات، لذا فإن نظيراً يسمح لعقدة بواسطة العنوان المقدم له هو السماح بالعنوان الذي تستخدمه الحركة فعلياً.

المضيف الذي ينتمي إلى **عدة مجموعات جرد** يتلقى **اتحاد** كل تدفقات كل مجموعة وخدماتها المعلنة. تكشف العقدة المتقاربة عن كل شيء لجميع الأدوار التي تعمل بها، ومجموعة المستمعين المتوقعة لها (المستخدمة في فحص الانحراف) هي الاتحاد عبر مجموعاتهما.

يمكن أن تسرد NIC، لأن الحل يستخدم كل عنوان معلن بدلاً من تتبع أي خدمة ترتبط بأي المعروف على N3 GTP-U على سبيل المثال) المصفوفة خدمة على عنوان لا ترتبط به فعلياً هذا هو تقريبي متعمد: مصفوفة كاملة ولكنها واسعة قليلاً آمنة لتسليمها. (UPF الخاص بـ N6 عنوان إلى كسر SIGTRAN إلى جدار حماية أعلى، بينما يؤدي حذف عنوان مسار بيانات حقيقي أو عنوان NIC الحركة بشكل صامت. إذا كانت الدقة حسب العنوان مطلوبة لاحقاً، يمكن أن تسمى الخدمة الذي ترتبط به؛ حتى ذلك الحين، يبقى الحل مبنياً على الحقائق وكاملاً.

مصفوفة الاتصالات / المنافذ

مصفوفة المنافذ هي نموذج التدفق المحلول ضد جرد الموقع   مسطحة إلى صف واحد لكل مصدر، مضيف وجهة، ومنفذ/بروتوكول واحد. إنها الشكل للمراجعة أو لتسليمها إلى جدار حماية أعلى. قم بإنشائها باستخدام

```
ansible-playbook -i hosts/<Deployment>/host_files/<Site>.yaml \
  util_playbooks/render_port_matrix.yaml
```

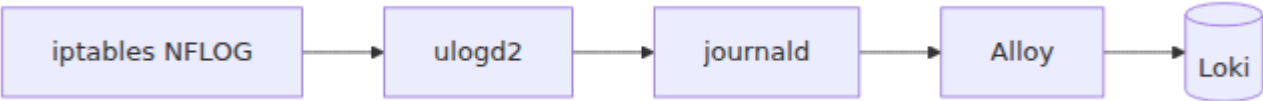
(للكل موقع، غير ملتزم) `generated/firewall-port-matrix-exploded.csv` هذا يكتب الأعمدة هي:

الوصف	العمود
معرف التدفق.	<code>id</code>
غرض التدفق.	<code>service</code>
أو عنوان مصدر تم حله CIDR.	<code>source</code>
مجموعة الجرد الوجهة.	<code>destination_role</code>
اسم المضيف الوجهة.	<code>destination_host</code>
عنوان المضيف الوجهة.	<code>destination_ip</code>
ICMP فارغ لإدخالات البروتوكول فقط (مثل) منفذ الوجهة أو النطاق أو ESP).	<code>port</code>
البروتوكول.	<code>protocol</code>

يظهر الصف فقط عندما يتم نشر وجهته في الموقع ويحل مصدره إلى عنوان حقيقي. الأدوار غير المنشورة والمصادر غير المحلولة لا تنتج أي صفوف.

حماية من الإغلاق

تتم حماية تطبيق القواعد بواسطة استعادة تلقائية، يتم التحكم فيها بواسطة `firewall_rollback_guard` و `firewall_rollback_timeout` (افتراضياً مفعل)



- 1. يتم التقاط مجموعة القواعد الحية.
- 2. يتم تسليح مؤقت على المضيف سيستعيد اللقطة. يبقى هذا حتى يتم قطع جلسة الإدارة.
- 3. يتم تطبيق القواعد الجديدة.
- 4. يعيد المتحكم إنشاء اتصال جديد ويجري فحصاً تافهاً. إذا كانت القواعد الجديدة قد أقفلت. عليه، فإن هذا يفشل ويتوقف التشغيل لذلك المضيف
- 5. إذا تم تأكيد الاتصال، يتم إلغاء تسليح المؤقت. إذا لم يكن كذلك، يتم تشغيل المؤقت. ويستعيد مجموعة القواعد العاملة دون أي إجراء من المشغل.

الوصف	الافتراضي	مطلوب	النوع	المعامل
يمكن اللقطة، المؤقت، وفحص إعادة الاتصال حول كل تطبيق.	true	لا	بولياني	firewall_rollback_guard
الثواني التي ينتظرها المؤقت قبل استعادة اللقطة تلقائياً.	120	لا	عدد صحيح	firewall_rollback_timeout

والدعم المزدوج IPv6

ويتم تطبيق القاعدة على IPv6، يعني : في القيمة CIDR: يتم استنتاج عائلة العنوان لكل IPv6 لا تنتج أي قواعد IPv4 CIDRs وبالتالي، فإن منطقة أو قائمة تحتوي فقط على `ip6tables`. المناطق والمجموعات الخاصة بـ IPv6 لذلك المصدر. تترك المواقع التي لا تحتوي على نواة إلى IPv6 لـ `INPUT` تحت فرض، يتم تعيين سياسة (`core_v6: []`) على سبيل المثال) فارغة في وضع المراقبة قبل فرض مضيف يصل إليه فقط من IPv6 لذا تحقق من حركة مرور، `DROP` فقط خلال IPv4 مصدر.

التسجيل

وتحمل عبر `NFLOG` يتم تسجيل الحركة المرورية التي تم تدقيقها وإسقاطها باستخدام `Loki` الأنابيب القياسي إلى:

البادئة	المعنى
<code>FW-FLOW</code> / <code>FW-FLOW6</code>	اتصال جديد تم رؤيته في وضع المراقبة. يبني خريطة التدفق الحية للحركة المرورية المستخدمة فعلياً.
<code>FW-AUDIT</code> / <code>FW-AUDIT6</code>	حركة المرور المتساقطة في وضع المراقبة: سيتم إسقاطها تحت الفرض، ولكنها مسموح بها هنا. راجع هذه قبل فرضها.
<code>FW-DROP</code> / <code>FW-DROP6</code>	حركة المرور المتساقطة التي تم إسقاطها فعلياً تحت الفرض.

المثال `Loki` استعلامات:

```
# الحركة المرورية التي سيتم حظرها بمجرد فرض هذا المضيف
{job="firewall"} |= "FW-AUDIT"

# الحركة المرورية التي يتم إسقاطها تحت الفرض، حسب المضيف
sum by (host) (count_over_time({job="firewall"} |= "FW-DROP"
[5m]))
```

تحذيرات تشغيلية

- **التجوال والمصادر النصية الحرة حالياً أي مصدر.** يتم إصدار التدفقات التي يكون بدون قيود على المصدر (0.0.0.0/0). اعتبرها `literal` أو `roaming` مصدرها. حقيقة CIDRs مفتوحة حتى يتم حل نطاقات التجوال إلى.
- **بدون ICMP من أي مصدر.** يسمح مجموعة القواعد المولدة بـ **ICMP يتم قبول** ping. شروط، وهو أوسع من نطاق الإدارة المعروض لتدفق.
- **لا تفرض مضيفاً في طبقة المستخدم دون قواعد توجيه.** تحت الفرض، يتم ولا يتم إنشاء أي قواعد توجيه. سيفشل فرض على `DROP` إلى `FORWARD` تعيين سياسة. دون إضافة قواعد توجيه في كسر طبقة المستخدم. تحقق أولاً SGW أو PGW أو UPF واجهة المستخدم، `SSH`، `allowed_oam_subnets` **قائمة الإدارة خشنة.** تحكم جمع، وواجهات برمجة التطبيقات الإدارية معاً. يحتاج الموقع الذي يجمع من شبكة لفصلها `extra_rules` مختلفة عن تلك التي يدير منها إلى إدخال.

تكوين متغيرات المجموعة

نظرة عامة

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

SMS، منطق توجيهه، Diameter قواعد توجيهه، SIP هنا تعيش تكويناتك الخاصة بالعملاء: خطوط خطط الاتصال، وأي تخصيصات أخرى حيث لا تريد استخدام التكوين الافتراضي. إنه موجود في `group_vars`.

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

`group_vars/` دليل - `group_vars_omnicore` تحل الأدوار هذا الدليل في وقت التشغيل كـ `{{ inventory_dir.rspllit('/', 1)[0] }}` الذي يقع بجوار دليل الجرد `{{ group_vars/ }}` في أي مكان يشير فيه مسار الملف أدناه إلى `group_vars_omnicore` `}}` فإنه يشير إلى هذا الدليل.

كيف يعمل

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

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

المثال 1: قالب تكوين مخصص (OmniMessage)

مخصصة Jinja2 بعض المكونات تقبل قوالب تكوين.

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 group_vars الخاص بك في
```

ماذا يحدث:

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

يتم استخدام القالب الافتراضي من الدور `smc_template_config` بدون

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

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

بعض المكونات تستخدم أدلة من ملفات التكوين.

هيكل الملف

```
hosts/Customer/
├── group_vars/
│   ├── gateways_prod/                # SIP تكوينات بوابة
│   │   ├── gateway_carrier1.xml
│   │   ├── gateway_carrier2.xml
│   │   └── gateway_emergency.xml
│   ├── gateways_lab/                # بوابات المختبر
│   │   └── gateway_test.xml
│   └── dialplan/                    # قواعد توجيه المكالمات
│       ├── mo_dialplan.xml          # منشأ من الهاتف المحمول
│       ├── mt_dialplan.xml          # منتهي من الهاتف المحمول
│       ├── emergency.xml
│       └── dialplan_lab/            # خطط الاتصال في المختبر
│           └── mo_dialplan.xml
```

الإشارة في ملف المضيفين

```
applicationserver:
  hosts:
    customer-tas01:
      ansible_host: 10.10.3.60
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      gateways_folder: "gateways_prod" # الإشارة إلى مجلد البوابة الخاص
      dialplan_folder: "dialplan"      # اختياري - الافتراضي هو
      "dialplan"                       # إذا لم يتم تعيينه
```

ماذا يحدث:

1. يجد Ansible `gateways_folder: "gateways_prod"`
2. إلى `hosts/Customer/group_vars/gateways_prod/` ينسخ جميع الملفات من `/etc/freeswitch/sip_profiles/`

3. أو المجلد) `hosts/Customer/group_vars/dialplan/` ينسخ جميع الملفات من `dialplan_folder` المحدد بواسطة OmniTAS إلى دليل قوالب
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"`

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

المثال 3: قالب تكوين مخصص (OmniHSS)

يقبل خادم المشتركين المنزلي قوالب تكوين وقت التشغيل المخصصة.

❖❖ هيكل المل

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

المخصص الخاص بك HSS قالب تكوين #

الإشارة في ملف المضيفين

```
omnihss:
  hosts:
    customer-hss01:
      ansible_host: 10.10.3.50
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      hss_template_config: hss_runtime.exs.j2  # الإشارة إلى اسم ملف
      group_vars: 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. Jinja2 يقوم بتطبيقه باستخدام `{{ inventory_hostname }}`، `{{ plmn_id.mcc }}`، إلخ
4. ينشر إلى `/etc/omnihss/runtime.exs`
5. يعيد تشغيل الخدمة

يتم استخدام القالب الافتراضي من الدور، `hss_template_config` بدون

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

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
# لمخصص الخامس MME قالب تكوين
بك
```

الإشارة في ملف المضيفين

```
omnimme:
  hosts:
    customer-mme01:
      ansible_host: 10.10.3.51
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      mme_template_config: mme_runtime.exs.j2  # الإشارة إلى اسم ملف
      group_vars: group_vars
```

ماذا يحدث:

1. يجد Ansible `mme_template_config: mme_runtime.exs.j2`
2. يبحث في `hosts/Customergroup_vars/mme_runtime.exs.j2`
3. Jinja2 يقوم بتطبيقه باستخدام `{{ inventory_hostname }}`، `{{ plmn_id.mcc }}`، إلخ
4. ينشر إلى `/etc/omnimme/runtime.exs`
5. يعيد تشغيل الخدمة

يتم استخدام القالب الافتراضي من الدور، `mme_template_config` بدون

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

المثال 5: ملفات التكوين المطلوبة (OmniCRM)

لا يوجد دور افتراضي للرجوع إليه. تم - `group_vars` حصريًا من OmniCRM يتم الحصول على ليكون غير مدمج في الدور بحيث يفشل التشغيل بصوت عالٍ بدلاً من شحن API تصميم تكوين. أسرار موقع آخر بصمت.

apt وضع النشر المفضل: حزم

بتثبيت حزم Ansible يقوم، (الإعداد الافتراضي للدور) `omnicrm_deploy_mode: package` مع `omnicrm-api` و `omnicrm-ui` العامة من `apt_repo`، تطبيق التكوين تحت،

لا يتم إعادة بناء واجهة systemctl عبر omnircm-api ويعيد تشغيل /etc/omnircm/، يتم) crm_config.yaml في ui: المستخدم على المضيف؛ يأتي العلامة التجارية من كتلة (GET /crm/public/ui-config تقديمها ك

هيكل الملف

```
hosts/Customr/
└─ group_vars/
    └─ crm_config.yaml          # مطلوب: API + ui تكوين
    └─ deployment_assets/      → شعارات / أيقونات اختيارية
/etc/omnircm/deployment_assets
├─ └─ logos/
├─ customer_crm.env           # اختياري؛ فقط لوضع
omnircm_deploy_mode: source   المحلية أو تطوير واجهة المستخدم
├─ cellSites.txt              # بيانات مواقع الخلايا الاختيارية
├─ constants.js               # قوالب رسائل البث الخلوي
الاختيارية
├─ site_data.json             # حدود جغرافية لبث الخلايا
الاختيارية
```

الإشارة في ملف ❓❓ لمضيفين

```
omnicrm:
  hosts:
    customer-crm01:
      ansible_host: 10.10.3.70
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      omnicrm_deploy_mode: package          # source الافتراضي؛ استخدم
      # القديم git+yarn
      # omnicrm_package_version: "20260810...." # تثبيت اختياري
      logo_light: logo-light.png           # يتم نسخه إلى
      deployment_assets/logos/
      logo_dark: logo-dark.png
      login_background: login-bg.jpg        # →
      deployment_assets/login/auth-one-bg.jpg (وقت التشغيل)
      login_splash: splash.jpg              # →
      deployment_assets/login/login-side.jpg (يحافظ على امتداد المصدر)
      ssl_fqdn: portal.example.com
      # crm_env_override: customer_crm.env   # مطلوب فقط لوضع المصدر
      cell_sites: cellSites.txt             # اختياري
      message_templates: constants.js       # اختياري
      site_data: site_data.json             # اختياري
```

ماذا يحدث (وضع الحزمة)

1. omnicrm-ui و omnicrm-api بتثبيت Apt يقوم.
2. إلى {{ group_vars_omnicore }}/crm_config.yaml يقوم بتطبيق (/etc/omnicrm/crm_config.yaml (مطلوب).
3. إلى (الاختياري) logo_* / login_* (و deployment_assets/ يقوم بمزامنة /etc/omnicrm/deployment_assets/).
4. بتقديم واجهة المستخدم من nginx ؛ يقوم omnicrm-api يعيد تشغيل /usr/share/omnicrm-ui.

انظر) GET /crm/public/assets/login/... يتم تقديم شاشة تسجيل الدخول / الخلفية عبر ui.branding.login_splash / login_background في crm_config.yaml). لا يتم splash.jpg → (يحفظ النسخ بامتداد المصدر .deb. دمجها في واجهة المستخدم موجود login-side.* بتقديم أي ملف API ثم يقوم (login/login-side.jpg).

ماذا يحدث (وضع المصدر، القديم)

1. يقوم بتطبيق `crm_config.yaml` إلى `/etc/omnicrm/OmniCRM-API/`.
2. **مطلوب** `.env` إلى واجهة المستخدم `{{ crm_env_override }}` يقوم بتطبيق (في وضع المصدر).
3. `yarn` يقوم ببناء واجهة المستخدم على المضيف باستخدام.

`ui:` موجودًا. بالنسبة لوضع الحزمة، قم بتضمين كتلة `crm_config.yaml` **مهم:** يجب أن يكون `ui.assets_dir: '/etc/omnicrm/deployment_assets'` واضبط.

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

المثال 6: دليل بيانات التزويد (OmniHSS)

بيانات المشتركين (`services/provision_omnihss.yaml`) OmniHSS تقرأ خدمة تزويد `group_vars` داخل `omnihss/` والملفات الشخصية من دليل مخصص.

هيكل الملف

```
hosts/Customr/  
└─ group_vars/  
    └─ omnihss/  
        ├── subscribers.csv           # المشتركين للتزويد  
        ├── apn_identifiers.json  
        ├── apn_qos_profiles.json  
        ├── apn_profiles.json  
        ├── eir_rules.json  
        ├── ims_profiles.json  
        ├── epc_profiles.json  
        ├── roaming_rules.json  
        ├── roaming_profiles.json  
        ├── flow_information_rules.json  
        ├── charging_rules.json  
        └─ pcrf_profiles.json
```

ماذا يحدث:

1. تقرأ الخدمة `{{ group_vars_omnicore }}/omnihss/subscribers.csv` لقائمة المشتركين للتزويد.
2. `omnihss/` المصاحبة من نفس دليل JSON تقرأ الملفات `(apn_identifiers.json, apn_qos_profiles.json, apn_profiles.json, eir_rules.json, ims_profiles.json, epc_profiles.json, roaming_rules.json, roaming_profiles.json, flow_information_rules.json, charging_rules.json, pcrf_profiles.json)` لتعريف بيانات APN و QoS و IMS و EPC و roaming و flow و charging و PCRF المشتركين المطبقة على هؤلاء المشتركين.

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

SMTP المثال 7: بيانات اعتماد Grafana (المراقبة)

لتنبيهات (SMTP) لدور المراقبة بتوصيل إعدادات البريد الإلكتروني `grafana.ini` تقوم قالب لا توجد قيمة افتراضية للدور لاسم المستخدم وكلمة المرور - تم `group_vars` من Grafana تصميمها عمداً لتكون غير مدمجة، بحيث يفشل التشغيل بصوت عالٍ بدلاً من شحن بيانات الاعتماد.

الإشارة في ملف المضيفين

```
monitoring:
  hosts:
    customer-oam01:
      ansible_host: 10.10.3.135
      gateway: 10.10.3.1
      host_vm_network: "vmbr3"
      grafana_smtp_user: "smtp-user" # مطلوب - لا يوجد
افتراضي
      grafana_smtp_password: "smtp-password" # مطلوب - لا يوجد
افتراضي
      grafana_smtp_host: "in-v3.mailjet.com:587" # اختياري -
in-v3.mailjet.com:587 الافتراضي هو
```

ماذا يحدث:

1. في قسم `grafana_smtp_password` و `grafana_smtp_user` يتم تطبيق `[smtp]` من `/etc/grafana/grafana.ini` لا يوجد بديل.
2. `in-grafana_smtp_host` اختياري ويفترض القيمة الافتراضية `v3.mailjet.com:587` إذا لم يتم تعيينه.

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

لفك تشفير HNET المثال 8: مفاتيح UDM في SUCI

المخفية التي SUCI من SUPI تستعيد: (TS 33.501 §6.12) SUCI بفك تشفير OmniUDM تقوم عند التسجيل. يتطلب ذلك مفاتيح خاصة للشبكة المنزلية تتراوح مع مفاتيح عامة UE يقدمها تعيش هذه المفاتيح الخاصة في دليل مخصص. USIMs للشبكة المنزلية المجهزة على UDM على مضيف `/etc/omniudm/hnet/` ويتم نشرها إلى `group_vars` داخل `hnet_udm/` `gc_nf` بواسطة دور 5.

هيكل الملف

```
hosts/Customer/
├── group_vars/
│   └── hnet_udm/
│       ├── curve25519-1.key          # معرف (PEM، X25519 مفتاح خاص
│       │                               (المفتاح العام للشبكة المنزلية 1
│       ├── curve25519-1.pub          # - (hex) قيمة عامة خام 32 بايت
│       │                               USIM لتجهيز
│       ├── secp256r1-2.key           # EC (prime256v1)، مفتاح خاص
│       │                               (HNET 2) معرف المفتاح العام (PEM
│       └── ...
```

هو معرف المفتاح العام للشبكة المنزلية `<id>` حيث `<curve>-<id>.key`، **تسمية الملفات** أو `curve25519` هو `<curve>` و `SUCI` مع `UDM` الذي يتطابق اسم الملف هو كيف يتم ربط المفتاح بمعرف. `secp256r1` (المعرف الشخصي `prime256v1`، B، الملف الشخصي) `USIM` الخاص به. يجب أن يتطابق مع المعرف المبرمج على `HNET` مفتاح.

ماذا يحدث:

1. موجود على وحدة `{{ group_vars_omnicore }}/hnet_udm/` يضمن الدور أن تبقى المفاتيح على وحدة `become: false` تم إنشاؤه باستخدام Ansible تحكم (التحكم ولا يتم إنشاؤها أبداً على الجهاز البعيد).
2. `*.key` يقوم بفحص الدليل بحثاً عن ملفات.
3. `curve25519-1.key` إذا لم يتم العثور على أي منها، يتم إنشاء مفتاح افتراضي `openssl genpkey` تلقائياً باستخدام `(X25519)` `1.key` لا يتم الكتابة فوق `(creates: guard)` idempotent من الصندوق. هذا هو SUCI مفتاح موجود.
4. يتم تصدير القيمة العامة الخام المطابقة 32 بايت إلى `curve25519-*.key` لكل ملف `.pub` (hex) لتجهيز USIM. تبقى ملفات UDM.
5. الوضع UDM على `/etc/omniudm/hnet/` إلى `*.key` يتم نسخ جميع ملفات `omniudm` ويتم إعادة تشغيل `(0600)`، مملوكة للجذر.

USIMs من SUCIs **مهم:** المفتاح الذي تم إنشاؤه تلقائياً عشوائياً وسيقوم فقط بفك تشفير `.pub` خذ، SIM المبرمجة بمفتاحها العام **المطابق**. بالنسبة لمختبر تتحكم فيه في برمجة SIMs إذا كانت USIM على (ID 1) المصدرة وقم بتجهيزها كمفتاح عام للشبكة المنزلية ويمكن أن يبقى هذا HNET فلا حاجة لمفتاح، (بدون إخفاء) `null` الخاصة بك تستخدم مخطط الدليل فارغاً.

:لتصدير مفتاح عام من مفتاح خاص موجود يدوياً

```
openssl pkey -in curve25519-1.key -pubout -outform DER | tail -c 32 | xxd -p -c 32
```

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

مثال هيكل الدليل في العالم الحقيقي

```
hosts/Customer/
├─ host_files/
│   └─ production.yml          # ملف المضيفين يشير إلى ملفات
group_vars
├─ group_vars/
│   └─ smsc_controller.exs     # قالب مخصص لـ OmniMessage
│   └─ smsc_smpp.exs          # قالب مخصص لـ OmniMessage SMPP
│   └─ tas_runtime.exs.j2      # قالب مخصص لـ TAS
│   └─ hss_runtime.exs.j2      # قالب مخصص لـ HSS
│   └─ mme_runtime.exs.j2      # قالب مخصص لـ MME
│   └─ dra_runtime.exs.j2      # قالب مخصص لـ DRA
│   └─ pgwc_runtime.exs.j2     # قالب مخصص لـ PGW
│   └─ dea_runtime.exs.j2      # قالب مخصص لـ DEA
│   └─ upf_config.yaml         # UPF تكوين
│   └─ crm_config.yaml         # CRM تكوين
│   └─ 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 # SMSC IMS تكوين
│   └─ pytap.yaml              # TAP3 تكوين
│   └─ sip_profiles/           # (مجلد) SIP بوابات
│       └─ gateway_otw.xml
│   └─ dialplan/               # قواعد توجيه المكالمات (مجلد)
│       └─ mo_dialplan.xml      # منشأ من الهاتف المحمول
│       └─ mt_dialplan.xml      # منتهي من الهاتف المحمول
│       └─ mo_emergency.xml     # توجيه الطوارئ
│   └─ omnihss/                # (مجلد) OmniHSS بيانات تزويد
│       └─ subscribers.csv      # المشتركين للتزويد
│       └─ apn_identifiers.json
│       └─ apn_qos_profiles.json
│       └─ apn_profiles.json
│       └─ eir_rules.json
│       └─ ims_profiles.json
│       └─ epc_profiles.json
│       └─ roaming_rules.json
│       └─ roaming_profiles.json
│       └─ flow_information_rules.json
│       └─ charging_rules.json
│       └─ pcrf_profiles.json
```

└─ hnet_udm/ # UDM في SUCI مفاتيح فك تشفير
(مجلد)
└─ curve25519-1.key # معرف (X25519 مفتاح خاص
(المفتاح العام للشبكة المنزلية 1
└─ curve25519-1.pub # لتجهيز (hex) قيمة عامة خام
USIM

المعلومات الشائعة التي تشير إلى

group_vars

المعلمة	المكون	الإشارات
<code>smc_template_config</code>	omnimessage	مثل (Jinja2 ملف قالب <code>smc_controller.exs</code>)
<code>smc_smpp_template_config</code>	omnimessage_smpp	مثل (Jinja2 ملف قالب <code>smc_smpp.exs</code>)
<code>gateways_folder</code>	applicationserver	مثل (اسم المجلد <code>gateways_prod</code>)
<code>dialplan_folder</code>	applicationserver	مثل (اسم المجلد <code>dialplan</code>) الافتراضي - إذا لم يتم <code>dialplan</code> هو تعيينه
<code>tas_runtime_template</code>	applicationserver	مثل (Jinja2 ملف قالب <code>tas_runtime.exs.j2</code>) الافتراضي هو - <code>tas_runtime.exs.j2</code> إذا لم يتم تعيينه
<code>hss_template_config</code>	omnihss	مثل (Jinja2 ملف قالب <code>hss_runtime.exs.j2</code>)
<code>mme_template_config</code>	omnimme	مثل (Jinja2 ملف قالب <code>mme_runtime.exs.j2</code>)
<code>dra_template_config</code>	dra	مثل (Jinja2 ملف قالب <code>dra_runtime.exs.j2</code>)
<code>pgwc_template_config</code>	pgwc	مثل (Jinja2 ملف قالب <code>pgwc_runtime.exs.j2</code>)

المعلمة	المكون	الإشارات
frr_template_config	omniupf	مثل Jinja2 ملف قالب frr.conf.j2)
SS7 قوالب	ss7 (أدوار متنوعة)	مثل Jinja2 ملفات قالب stp.j2, hlr.j2, camel.j2)
crm_config.yaml	omnicrm	مطلوب ، - API ملف تكوين يتم الحصول عليه فقط من {{ group_vars_omnicore }}/crm_config.yaml (لا يوجد دور افتراضي)
crm_env_override	omnicrm	اسم ملف واجهة المستخدم (مثل .env customer_crm.env) - مطلوب ، لا يوجد دور افتراضي
cell_sites	omnicrm	ملف بيانات مواقع الخلايا (مثل الاختيارية cellSites.txt)
message_templates	omnicrm	قوالب رسائل البث الخلوي (مثل الاختيارية constants.js)
site_data	omnicrm	حدود جغرافية لبث الخلايا (مثل الاختيارية site_data.json)
grafana_smtp_user	monitoring	- Grafana SMTP مستخدم مطلوب ، لا يوجد دور افتراضي

المعلمة	المكون	الإشارات
<code>grafana_smtp_password</code>	monitoring	Grafana كلمة مرور مطلوب ، لا يوجد - SMTP دور افتراضي
<code>grafana_smtp_host</code>	monitoring	Grafana SMTP مضيف اختياري - الافتراضي هو <code>in-v3.mailjet.com:587</code>
OmniHSS تزويد	omnihss	مع <code>omnihss/</code> دليل <code>subscribers.csv</code> + للملف JSON ملفات الشخصي
HNET مفاتيح	5gc_nf (omniudm)	لمفاتيح <code>hnet_udm/</code> دليل - <code><curve></code> خاصة لـ <code><id>.key</code> تشفير يتم إنشاء مفتاح - SUCI - <code>curve25519</code> افتراضي تلقائيًا إذا كان فارغًا <code>1.key</code>
للتكوين YAML ملفات	مكونات متنوعة	(مثل ملفات تكوين مباشرة <code>upf_config.yaml</code> ، <code>crm_config.yaml</code>)

النقاط الرئيسية

- على تخصيصات - تجاوزات للتكوينات الافتراضية `group_vars` تحتوي
- أو `smc_template_config` الإشارة بالاسم - استخدم معلمات مثل `gateways_folder`
- {{ باستخدام Ansible الوصول إلى أي متغير - Jinja2 تدعم القوالب `variable_name }}`

4. تقوم المجلدات بنشر كل شيء - يتم نسخ جميع الملفات في المجلدات المشار إليها
5. Git إلى group_vars تحكم في الإصدار لكل شيء - قم بالتزام جميع

group_vars متى تستخدم

ل: group_vars استخدم

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

لا group_vars لا تستخدم

- استخدم ملف المضيفين - (أسماء المضيفين، IP عناوين) تكوين المضيف الأساسي
- الاختبارات لمرة واحدة - استخدم متغيرات محددة للمضيف في ملف المضيفين
- إذا كانت قائمة group_vars التغييرات المؤقتة - قم بالتعديل على الهدف، والتزم إلى

الوثائق ذات الصلة

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

تكوين ملف المضيفين

نظرة عامة

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

- ما هي وظائف الشبكة التي يجب نشرها
- (شرائح الشبكة، IP عناوين) أين تعمل
- كيف يتم تكوينها (معلومات محددة للخدمة)
- (بيانات الاعتماد، الميزات، PLMN) إعدادات محددة للعميل

موقع الملف

في `inventory` (المحدد كـ `hosts/`) توجد ملفات المضيفين تحت الدليل الرئيسي `ansible.cfg`، منظمة حسب العميل:

```
hosts/
├── Customer_Name/
│   ├── host_files/
│   │   ├── Production.yml
│   │   ├── Staging.yml
│   │   └── Customer_Lab.yml
```

أسماء الملفات عشوائية. يوجد ملف واحد لكل نشر، وعادة ما يكون مسمى باسم النشر أو الموقع بدلاً من نظام ثابت (`Production.yml`، `Staging.yml` مثل) `production.yml/staging.yml/lab.yml`.

مثال على هيكل ملف المضيفين

:يوضح هذا المثال المبسط الأقسام الرئيسية

مكونات EPC

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

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

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

مكونات IMS

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

خدمات الدعم

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

المتغيرات العالمية

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

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

معلومات المضيفين الشائعة

تكوين الشبكة

يتضمن كل مضيف عادةً:

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

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

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

VM تخصيص موارد

للخدمات التي تحتاج إلى موارد محددة:

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

معلومات محددة للخدمة

لكل وظيفة شبكة معلوماتها الخاصة. أمثلة:

MME:

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

PGW:

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

للحصول على شرح مفصل لما يتحكم فيه كل متغير، راجع: [مرجع التكوين](#)

خادم التطبيقات:

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

قسم المتغيرات العالمية

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

```

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

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

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

    # أسماء الشبكة
    network_name_short: Customer
    network_name_long: Customer Network

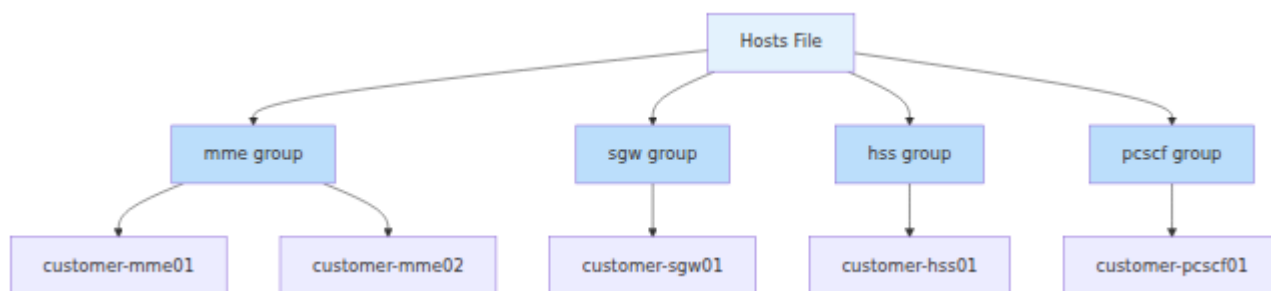
    # مستودع APT
    # مع المضيفين apt_cache_servers ملاحظة: إذا تم تعريف مجموعة
    # apt_repo.apt_server و true افتراضيًا يكون use_apt_cache فإن
    # لأحد خوادم التخزين المؤقت IP افتراضيًا يكون عنوان
    apt_repo:
      apt_server: "10.254.10.223"
      apt_repo_username: "customer"
      apt_repo_password: "secure-password"
    use_apt_cache: false

    # المنطقة الزمنية
    TZ: America/Chicago

```

فهم مجموعات المضيفين

المضيفين في مجموعات تتوافق مع الأدوار Ansible تنظم:



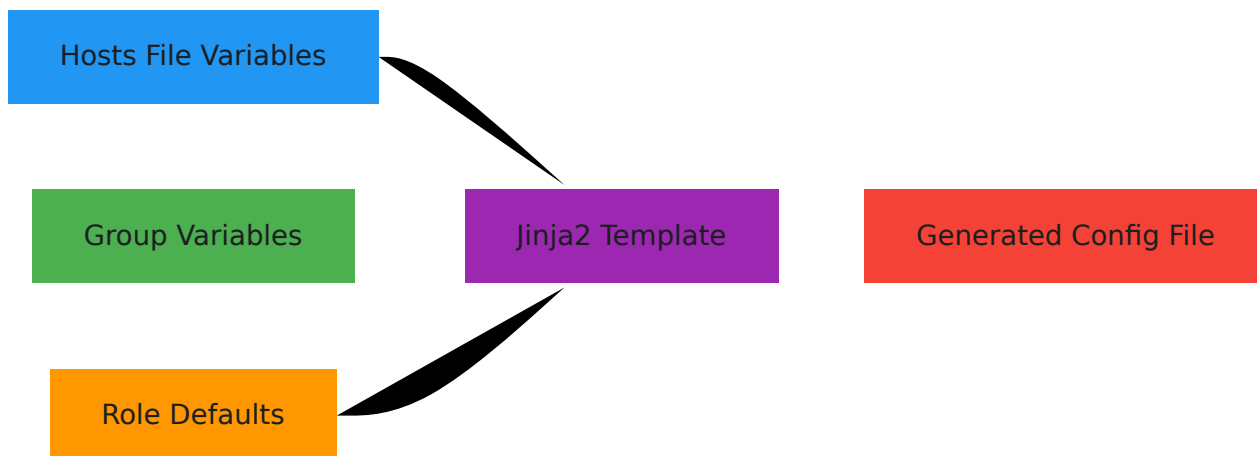
فإنه ينطبق على جميع المضيفين في قسم `mme`، عند تشغيل كتاب اللعب الذي يستهدف `mme:hosts:`.

يوضح الرسم البياني أعلاه عددًا قليلًا فقط من المجموعات. توجد العديد من المجموعات الأخرى، ومجموعات `ocs`، `crm`، `smsc`، `omnimessage`، `dra`، `xcap`، `homer`، `cbc`، من بين آخرين `upf`، و `5GC amf`، `smf`.

Jinja2 التكوين باستخدام قوالب

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

Jinja2 كيف تعمل



مثال على استخدام القالب

يحدد ملف المضيفين:

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

: قالب Jinja2 (في الدور):

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

: ملف التكوين الناتج:

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

الشائعة Jinja2 أنماط

: الوصول إلى المتغيرات المتداخلة:

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

: المنطق الشرطي:

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

الحلقات:

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

التنسيق:

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

group_vars تجاوز المتغيرات باستخدام

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

راجع: تكوين متغيرات المجموعة

مثال كامل على ملف المضيفين

هذا المثال الأكثر اكتمالاً يحتوي على بيانات حساسة مخفية:

EPC Core

mme:

hosts:

customer-mme01:

ansible_host: 10.10.1.15

gateway: 10.10.1.1

host_vm_network: "vmbr1"

mme_code: 1

mme_gid: 1

network_name_short: Customer

network_name_long: Customer Network

tac_list: [600, 601, 602, 603]

omnimme:

sgw_selection_method: "random_peer"

pgw_selection_method: "random_peer"

sgw:

hosts:

customer-sgw01:

ansible_host: 10.10.1.25

gateway: 10.10.1.1

host_vm_network: "vmbr1"

cdns_enabled: true

pgwc:

hosts:

customer-pgw01:

ansible_host: 10.10.1.21

gateway: 10.10.1.1

host_vm_network: "vmbr1"

ip_pools:

- '100.64.16.0/24'

combined_CP_UP: false

hss:

hosts:

customer-hss01:

ansible_host: 10.10.2.140

gateway: 10.10.2.1

host_vm_network: "vmbr2"

IMS Core

pcscf:

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

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

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

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

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

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

```
memory_mb: 8192

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

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

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

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

# أسماء الشبكة
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 الجدار الناري
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'

# Hypervisor (مثال Proxmox VM) تكوين
proxmoxServers:
  customer-prxm01:
    # "vm" تم حذف نوع النشر → الافتراضي هو
    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 # cloud-اسم مستخدم
    local_users اختياري؛ الافتراضي هو أول مفتاح init
    proxmoxTemplatePassword: omnitouch # cloud-init كلمة مرور
    cloud-init اختيارية؛ الافتراضي هو اسم مستخدم

# و LXC، لموقع deployment_type: lxc قم بتعيين
# بدلاً من ذلك proxmoxServers في كل إدخال proxmoxLxc0sTemplate

```

والتكوين الكاملة Proxmox للحصول على تفاصيل إعداد Proxmox VM/LXC راجع [نشر](#).

مراجع وثائق المنتج

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

OmniCore مكونات:

- **OmniCore وثائق:**
<https://docs.omnitouch.com.au/docs/repos/OmniCore>
- **OmniHSS** - خادم المشتركين المنزلي
- **OmniSGW** - بوابة الخدمة (خطة التحكم)
- **OmniPGW** - بوابة الحزمة (خطة التحكم)
- **OmniUPF** - وظيفة خطة المستخدم
- **OmniDRA** - Diameter وكيل توجيه
- **OmniTWAG** - الموثوقة WLAN بوابة الوصول

OmniCall مكونات:

- **OmniCall وثائق:** <https://docs.omnitouch.com.au/docs/repos/OmniCall>
- **OmniTAS** - IMS (VoLTE/VoNR) خادم تطبيق
- **OmniCall CSCF** - وظائف التحكم في جلسة المكالمات
- **OmniMessage** - مركز الرسائل   لقصيرة
- **OmniMessage SMPP** - SMPP دعم بروتوكول
- **OmniSS7** - SS7 مجموعة إشارات
- **VisualVoicemail** - البريد الصوتي

OmniCharge/OmniCRM:

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

الوثائق ذات الصلة

- **عملية النشر العامة** - Ansible مقدمة في نشر
- **مرجع التكوين** - دليل كامل لجميع متغيرات التكوين
- **تكوين متغيرات المجموعة** - تجاوز التكوينات الافتراضية

- IP إرشادات بنية الشبكة وتخصيص - IP معيار تخطيط
- الثانوية وتكوين الشبكة المتقدم IP عناوين - Netplan تكوين
- توزيع الحزم - APT نظام تخزين
- 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. وهويتك الشبكية PLMN حدد
3. APT قم  تكوين وصول مستودع
4. إعداد خادم الترخيص
5. حسب الحاجة `group_vars` تخصيص باستخدام
6. Ansible نشر باستخدام كتب لعب

خادم الترخيص

نظرة عامة

يتحقق كل مكون من ترخيصه. Omnitouch. يدير خادم الترخيص تفعيل الميزات لجميع مكوناته. 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: "اسم العميل"
    license_server_api_urls: ["https://10.10.2.150:8443/api"]
```

وليس تبديل تطبيق عالمي. يتم تعريفه، **HSS** هو نطاق `license_enforced` **ملاحظة:** حيث يكون افتراضيًا، `(roles/omnihss/defaults/main.yml)` `omnihss` فقط لدور `false`؛ لا يستهلكه أي دور آخر. قم بتعيينه تحت HLD و يتم قراءته حاليًا فقط بواسطة مولد `(false)`. إذا كنت بحاجة إليه `omnihss` متغيرات مجموعة.

2. توفير ملف الترخيص

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

النشر .3

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

مثل) تكون أدلة الجرد الحقيقية مكتوبة بحروف كبيرة ومسبوبة
(مثل) وملف المضيف يسمى باسم الموقع ،(hosts/Omnicores_Customer/
hosts/Omnicores_Customer/host_files/Production.yml).

. https://license_server للتحقق من حالة جميع التراخيص، انتقل إلى

إزالة ترخيص

.util_playbooks/delete_license.yml لإزالة ترخيص، استخدم

متطلبات الشبكة

تكوين جدار الحماية

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

الغرض	اسم المضيف
خادم التحقق من الترخيص 1	1.time.omnitouch.com.au
خادم التحقق من الترخيص 2	2.time.omnitouch.com.au
خادم التحقق من الترخيص 3	3.time.omnitouch.com.au

قواعد الخروج المطلوبة:

- البروتوكول: HTTPS (TCP/443)
- DNS الوجهة: أسماء المضيفين أعلاه، يتم حلها عبر
- الاتجاه: صادر

FQDN دائمًا اسمح لهذه المضيفين بواسطة **IP لا تقم بتشغيل عناوين DNS: حل عبر** يمكن أن تتغير. (أو بواسطة العناوين التي يقوم جدار الحماية الخاص بك بحلها ديناميكيًا) الثابتة غير مدعومة وستؤدي في IP الأساسية دون إشعار. القوائم البيضاء لعناوين IP عناوين. النهاية إلى كسر التحقق من الترخيص.

DNS متطلبات

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

المطلوب DNS تكوين

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

يض  ن استخدام (DoH/DoT) الآمن DNS Omnitouch **ملاحظة:** تستخدم خوادم ترخيص بواسطة DNS ويمنع المشكلات مع اعتراض DNSSEC العامة التحقق الصحيح من DNS خوادم الأجهزة الأمنية.

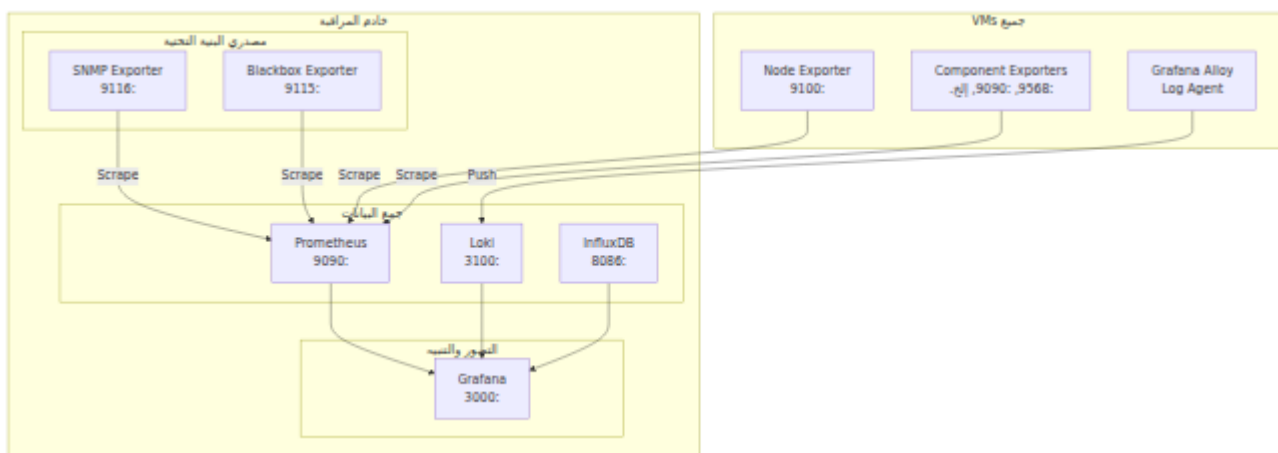
الوثائق ذات الصلة

- مرجع التكوين
- تكوين ملف المضيفين

المراقبة والمراقبة

نظرة عامة

مجموعة شاملة من أدوات المراقبة والمراقبة التي توفر جمع المقاييس، OmniCore يتضمن وتجميع السجلات، والتصوير، والتنبيه. يتم نشر النظام تلقائيًا بواسطة دور المراقبة



المكونات

المكون	الغرض	المنفذ	الاحتفاظ بالبيانات
Prometheus	تخزين مقاييس السلاسل الزمنية	9090	يومًا أو 5 30 جيجابايت
Loki	تجميع السجلات	3100	أيام أو 50 7 جيجابايت
InfluxDB	(نوكيا) RAN مقاييس	8086	يومًا 30
Grafana	التصور والتنبيه	3000	غير متاح
Node Exporter	مقاييس المضيف	9100	غير متاح
SNMP Exporter	مقاييس أجهزة الشبكة	9116	غير متاح
Blackbox Exporter	ICMP/HTTP اختبارات	9115	غير متاح

مصادر البيانات

Prometheus

كل دقيقة واحدة OmniCore بجمع المقاييس من جميع مكونات Prometheus يقوم

معرف مصدر البيانات: `omnicore_prometheus`

الأهداف المجمعة

اسم الوظيفة	مجموعة 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 المشاركين
SGW-C	sgw	9568	عدد الحاملات، رسائل GTP-C
PGW-C	pgwc	9090	PDN اتصالات، IP تخصيصات
UPF Standalone	upf	9090	عدد الحزم، عرض النطاق
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	جلسات SMS، تسليم SMPP
OmniSS7	omniss7	8080	مقاييس بوابة SS7/SIGTRAN

المقاييس	المنفذ	مجموعة الجرد	اسم الوظيفة
استخدام الذاكرة، العمليات/ثانية	9121	ocs	KeyDB
التصنيف، الشحن، CDR إحصائيات	2080	ocs	CGrateS

مصدري البنية التحتية

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

Loki

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

معرف مصدر البيانات: omnicores_loki

Loki/Alloy. انظر [التسجيل المركزي](#) للحصول على تفاصيل تكوين

تسميات السجلات

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

InfluxDB

بتخزين بيانات السلاسل الزمنية لحالات استخدام محددة تتطلب احتفاظًا أطول أو InfluxDB تقوم بأنماط استعلام مختلفة.

معرف مصدر البيانات: `omnicore_influxdb`

قواعد البيانات

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

Grafana لوحات معلومات

تنظيم لوحة المعلومات

تُنظم لوحات المعلومات في مجلدات حسب المنطقة الوظيفية

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

(API استنادًا إلى) تحميل لوحة المعلومات

بدلاً من توفيرها من خلال Grafana يتم تحميل لوحات المعلومات عبر واجهة برمجة تطبيقات الملفات. يتيح ذلك لوحات المعلومات أن تكون:

- Grafana معدلة من خلال واجهة مستخدم
- API معدلة عبر
- Ansible خاضعة للتحكم في الإصدارات في مستودع



تحميل لوحات المعلومات

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

```

# من وحدة التحكم
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

Ansible الخاصة بلوحة المعلومات في مستودع JSON يتم تخزين ملفات

```
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  
    └── ...
```

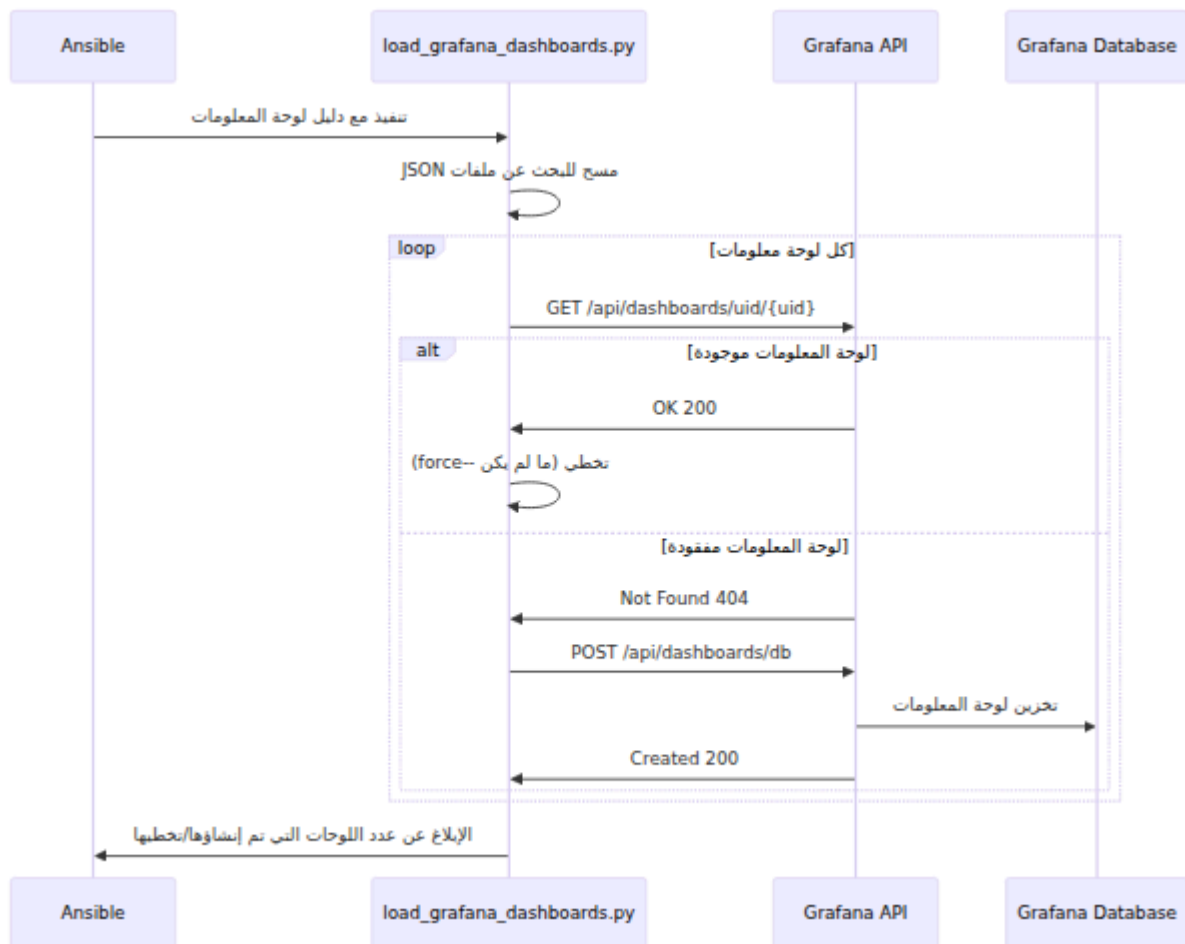
النسخ الاحتياطي للوحة المعلومات

قيد التشغيل إلى المستودع من أجل Grafana قم بتصدير لوحات المعلومات والتنبيهات من مثيل التحكم في الإصدارات:

```
# دليل من roles/monitoring  
python3 export_grafana.py --customer <CUSTOMER>
```

هذا يصدر:

- جميع لوحات المعلومات إلى
`roles/monitoring/templates/grafana/dashboards/{folder}/*.json`
(مشاركة عبر العملاء)
- قواعد التنبيه إلى
`hosts/Omnicores/CUSTOMER/group_vars/grafana/alerts/all_alert_rules.json`
- نقاط الاتصال إلى
`hosts/Omnicores/CUSTOMER/group_vars/grafana/alerts/contact_points.json`
- سياسات الإشعار إلى
`hosts/Omnicores/CUSTOMER/group_vars/grafana/alerts/notification_policies.json`



سلوك التصدير

- (`id` و `version`) يتجاهل الحقول المتقلبة مثل) يكتب فقط الملفات التي تغيرت

- Grafana يحافظ على هيكل المجلدات الذي يتطابق مع تنظيم
- يطابق عناوين لوحات المعلومات بأسماء ملفات متسقة

لوحات المعلومات المخصصة

يمكن وضع لوحات المعلومات الخاصة بالعمل في `group_vars/grafana/dashboards/` وسيتم تحميلها جنبًا إلى جنب مع لوحات المعلومات القياسية:

```
hosts/customer/group_vars/  
└─ grafana/  
    └─ dashboards/  
        ├── Custom_Dashboard_1.json  
        └── Custom_Dashboard_2.json
```

التنبيه

الهيكليّة

Grafana تنبيه

قواعد التنبيه
alerts.yaml

محرك التقييم
فترة 1 دقيقة

تنبيه يشير

وجه الإشعارات

مصادر البيانات

Prometheus

Loki

نقاط الاتصال

Slack Webhook

Google Chat

البريد الإلكتروني

Pushover

تحميل التنبيهات

من صادرات كل عميل بدلاً من ملف Grafana يتم تحميل التنبيهات عبر واجهة برمجة تطبيقات ثابت واحد. في وقت النشر، يتم إزالة ملف توفير التنبيه الثابت YAML بحيث يمكن تعديل التنبيهات (/etc/grafana/provisioning/alerting/alerts.yaml) باستخدام API عبر واجهة المستخدم.

المسار الرئيسي يحمل قواعد التنبيه الخاصة بكل عميل، ونقاط الاتصال، وسياسات الإشعار من باستخدام hosts/Omnicores/CUSTOMER/group_vars/grafana/alerts/load_grafana_from_export.py:

```
python3 roles/monitoring/files/load_grafana_from_export.py \
  --url http://<monitoring-ip>:3000 \
  --user admin \
  --password <grafana_admin_password> \
  --alerts-dir
hosts/Omnicores/CUSTOMER/group_vars/grafana/alerts \
  --dashboards-dir roles/monitoring/templates/grafana/dashboards
\
  --force
```

عبر alerts.yaml إذا لم توجد تنبيهات خاصة بالعميل، فإن الدور يتراجع إلى تحميل قالب الدور load_grafana_alerts.py:

```
python3 roles/monitoring/files/load_grafana_alerts.py \
  --url http://<monitoring-ip>:3000 \
  --user admin \
  --password <grafana_admin_password> \
  --alerts-file
roles/monitoring/templates/grafana/provisioning/alerting/alerts.yaml
```

قواعد التنبيه الافتراضية

المدة	الشرط	المجلد	التنبيه
5 دقائق	%نظام الملفات الجذر < 90	البنية التحتية	استخدام مساحة القرص 90%
5 دقائق	CPU > 90% استخدام	البنية التحتية	%فوق 90 CPU
5 دقائق	%متوسط الحمل < 90	البنية التحتية	%حمل النظام فوق 90
5 دقائق	غير Prometheus الهدف متاح	البنية التحتية	المضيف متوقف
30 ثانية	%انخفاض عدد الجلسات 40	EPC	انخفاض 40% في مشتركين MME
5 دقائق	لا يوجد مشتركين متصلين	EPC	MME مشتركين على 0
30 ثانية	P-CSCF انخفاض تسجيلات 40%	IMS	انخفاض 40% في مشتركين IMS
5 دقائق	جودة الصوت متدهورة	IMS	أقل من 4 MOS متوسط
1 دقيقة	المتصلة eNB انخفاض عدد	RAN	eNodeB انخفاض عدد
5 دقائق	P95 ارتفاع زمن الاستجابة	EPC	تأخير استجابة القطر مرتفع

هيكل قاعدة التنبيه

قواعد التنبيه في `alerts.yaml`:

```

apiVersion: 1
groups:
  - orgId: 1
    name: مجموعة التنبيهات
    folder: البنية التحتية
    interval: 1m
    rules:
      - uid: alert-lowDiskSpace
        title: استخدام مساحة القرص 90% على المضيف
        condition: C
        data:
          - refId: A
            datasourceUid: omnicores-prometheus
            model:
              expr: |
                100 - ((node_filesystem_avail_bytes{job="Node
Exporter",mountpoint="/" } * 100)
                  / node_filesystem_size_bytes{job="Node
Exporter",mountpoint="/" })
                # ... العتبة
            noDataState: OK
            execErrState: Error
            for: 5m
            annotations:
              summary: تجاوز استخدام مساحة القرص 90% على المضيف
            labels:
              type: resource

```

تكوين نقاط الاتصال

تُكوّن نقاط الاتصال عبر متغيرات الجرد

```
# أو متغيرات خاصة بالعمل في group_vars/all.yml
monitoring_data:
  contactPoints:
    - orgId: 1
      name: الافتراضي
      receivers:
        # Slack تكامل
        - uid: slack-alerts
          type: slack
          disableResolveMessage: false
          settings:
            url: "https://hooks.slack.com/services/xxx/yyy/zzz"

        # Google Chat تكامل
        - uid: gchat-alerts
          type: googlechat
          disableResolveMessage: false
          settings:
            url: "https://chat.googleapis.com/v1/spaces/xxx/messages?key=yyy"
```

انظر (Pushover والبريد الإلكتروني و Google Chat و Slack تشمل أنواع نقاط الاتصال المدعومة roles/monitoring/templates/grafana/contact-points.yaml.j2). يتم تكوين notifications.pushover عبر كتلة Pushover (مع enabled و api_token و user_key).

سياسات الإشعارات

توجه سياسات الإشعارات التنبيهات إلى نقاط الاتصال المناسبة:

```
# templates/grafana/notification-policies.yaml.j2
apiVersion: 1

policies:
  - orgId: 1
    receiver: الافتراضي
    group_by: ['grafana_folder', 'alertname']
    group_wait: 30s
    group_interval: 5m
    repeat_interval: 4h
```

مرجع التكوين

Prometheus تكوين

.على خادم المراقبة `/etc/prometheus/prometheus.yml` يقع في

المعلمة	الافتراضي	الوصف
<code>scrape_interval</code>	1m	كم مرة يتم جمع الأهداف
<code>evaluation_interval</code>	1m	كم مرة يتم تقييم قواعد التسجيل
<code>scrape_timeout</code>	50s	مهلة طلبات الجمع

Grafana تكوين

.على خادم المراقبة `/etc/grafana/grafana.ini` يقع في

:الإعدادات الرئيسية التي تم تكوينها بواسطة دور المراقبة

الوصف	القيمة	الإعداد
iframes تمكين التضمين في	true	<code>allow_embedding</code>
دليل التوفير	<code>/etc/grafana/provisioning</code>	<code>provisioning</code>

`admin_password` يتم ترك سطر) `grafana.ini` لم يتم تعيين كلمة مرور المسؤول في الذي يتم تمريره إلى `grafana_admin_password` بدلاً من ذلك، يتم تطبيقها عبر متغير. (معلقاً أثناء النشر (محملات لوحة المعلومات/التنبيه) API مهام محمل

(SMTP) تنبيه البريد الإلكتروني

انظر) `group_vars` في SMTP يتطلب تنبيه البريد الإلكتروني إعداد بيانات اعتماد `roles/monitoring/templates/grafana/grafana.ini`):

الوصف	الافتراضي	مطلوب	المتغير
اسم مستخدم لا يوجد SMTP. افتراضي. إذا لم يتم تعيينه، يفشل النشر بصوت عالٍ بدلاً من إرسال سر.	(لا شيء)	نعم	grafana_smtp_user
كلمة مرور لا يوجد SMTP. افتراضي. إذا لم يتم تعيينه، يفشل النشر.	(لا شيء)	نعم	grafana_smtp_password
مضيف المنفذ: SMTP.	in-v3.mailjet.com:587	لا	grafana_smtp_host
SMTP ما إذا كان مفعلاً.	true	لا	grafana_smtp_enabled

alerts+ (مثل customer_legal_name من from_name و from_address يتم اشتقاق <customer_legal_name>@omnitouch.com.au و Omnitouch Alerts - <customer_legal_name>).

Loki تكوين

Loki انظر [التسجيل المركزي](#) للحصول على إعدادات الاحتفاظ والتخزين الخاصة بـ

InfluxDB تكوين

.على خادم المراقبة /etc/influxdb/influxdb.conf يقع في

المدة	سياسة الاحتفاظ	قاعدة البيانات
يومًا 30	autogen	nokia-monitor
يومًا 365	autogen	dra
يومًا 90	autogen	Omnicharge_TAP3

منافذ الخدمة

الوصف	البروتوكول	المنفذ	الخدمة
API واجهة لوحة المعلومات و	HTTP	3000	Grafana
تخزين المقاييس والاستعلام	HTTP	9090	Prometheus
استيعاب السجلات والاستعلام	HTTP	3100	Loki
InfluxDB واجهة برمجة تطبيقات	HTTP	8086	InfluxDB
مقاييس المضيف	HTTP	9100	Node Exporter
SNMP وكيل مقاييس	HTTP	9116	SNMP Exporter
مقاييس الاختبار	HTTP	9115	Blackbox Exporter
والمقاييس Alloy واجهة	HTTP	12345	Alloy

العمليات الشائعة

عر؟؟ المقاييس

1. افتح Grafana على `http://<monitoring-ip>:3000`
2. انتقل إلى **لوحات المعلومات** واختر المجلد المناسب.
3. استخدم محدد نطاق الوقت لضبط نافذة العرض.

البحث في السجلات

1. افتح Grafana على `http://<monitoring-ip>:3000`
2. **Loki** انتقل إلى **استكشاف** واختر مصدر بيانات.
3. LogQL استخدم استعلامات:

```
# جميع السجلات من مضيف معين
{hostname="customer-mme01"}

# الأخطاء MME من جميع مكونات
{component="mme"} |~ "(?i)error"

# معين IMSI البحث عن
{component="hss"} |= "123456789012345"
```

إنشاء تنبيهات مخصصة

ثم تصديرها إلى دليل Grafana، تدفق العمل الموصى به هو إنشاء/تعديل التنبيهات في واجهة الخاص بالعميل (انظر **النسخ الاحتياطي للوحة المعلومات**) `group_vars/grafana/alerts/` بحيث تكون خاضعة للتحكم في الإصدارات وتُعاد تطبيقها في النشر التالي.

لتحميل التنبيهات الخاصة بالعميل يدويًا:

```
python3 roles/monitoring/files/load_grafana_from_export.py \
  --url http://<monitoring-ip>:3000 \
  --user admin --password <password> \
  --alerts-dir
hosts/Omnicores/<CUSTOMER>/group_vars/grafana/alerts \
  --dashboards-dir roles/monitoring/templates/grafana/dashboards
\
  --force
```

يتم استخدام قالب الدور

```
roles/monitoring/templates/grafana/provisioning/alerting/alerts.yaml
```

فقط كخيار احتياطي عندما لا يكون لدى العميل أي تنبيهات مصدرة.

النسخ الاحتياطي للوحات المعلومات

قم بتصديرها إلى المستودع، Grafana، بعد إجراء تغييرات في واجهة

```
cd roles/monitoring
python3 export_grafana.py --customer <CUSTOMER> --url
http://<monitoring-ip>:3000
git add ../../hosts/Omnicores/<CUSTOMER>/group_vars/grafana/
git commit -m "تحديث التنبيهات من الإنتاج"
```

مطلوبة. يقوم هذا البرنامج النصي بتصدير قواعد التنبيه، ونقاط الاتصال، `--customer` تكون حجة الخاص بالعميل؛ لم يعد يصدر `group_vars/grafana/alerts/` وسياسات الإشعار إلى دليل لوحات المعلومات، والتي تتم مشاركتها عبر العملاء في دليل الدور (`roles/monitoring/templates/grafana/dashboards/`).

استكشاف الأخطاء وإصلاحها

متوقف Prometheus الهدف

الأعراض: تظهر لوحة المعلومات "لا توجد بيانات" أو حالة الهدف تظهر متوقفة

الأسباب المحتملة:

- الخدمة غير قيد التشغيل على المضيف المستهدف
- جدار الحماية يحظر منفذ المصدر
- حل اسم المضيف غير صحيح

الحل:

1. تحقق مما إذا كانت الخدمة تعمل على الهدف:

```
systemctl status <service>  
curl http://localhost:<port>/metrics
```

2. تحقق من الاتصال من خادم المراقبة:

```
curl http://<target>:<port>/metrics
```

3. Prometheus تحقق من صفحة أهداف: `http://<monitoring-ip>:9090/targets`

لوحة المعلومات لا تُحمّل

الأعراض: تظهر لوحة المعلومات رمز تحميل أو خطأ

الأسباب المحتملة:

- فشل اتصال مصدر البيانات
- مهلة الاستعلام
- لوحة المعلومات JSON تلف

الحل:

1. **التكوين → مصادر البيانات → اختبار Grafana:** اختبار مصدر البيانات في
2. Grafana: `journalctl -u grafana-server -f` تحقق من سجلات
3. حاول إعادة تحميل لوحة المعلومات من المستودع

التنبيهات لا تُطلق

الأعراض: الشرط مستوفى ولكن لم يتم تلقي إشعار

الأسباب المحتملة:

- التنبيه متوقف
- تكوين نقطة الاتصال غير صحيح
- سياسة الإشعار لا تطابق

الحل:

1. **التنبيه → قواعد التنبيه:** Grafana تحقق من حالة التنبيه في
2. تحقق من اختبار نقطة الاتصال: **التنبيه → نقاط الاتصال → اختبار**
3. مراجعة توجيه سياسة الإشعار.

لا يمكنها الاتصال بمصدر البيانات Grafana

Grafana **الأعراض:** "بوابة سيئة" أو مهلة الاتصال في

الأسباب المحتملة:

- متوقفة Prometheus/Loki/InfluxDB خدمة
- localhost جدار الحماية يحظر اتصالات
- الخدمة مرتبطة بواجهة خاطئة

الحل:

1. تحقق من حالة الخدمة:

```
systemctl status prometheus loki influxdb
```

2. تحقق من أن الخدمة تستمع:

```
ss -tlnp | grep -E '9090|3100|8086'
```

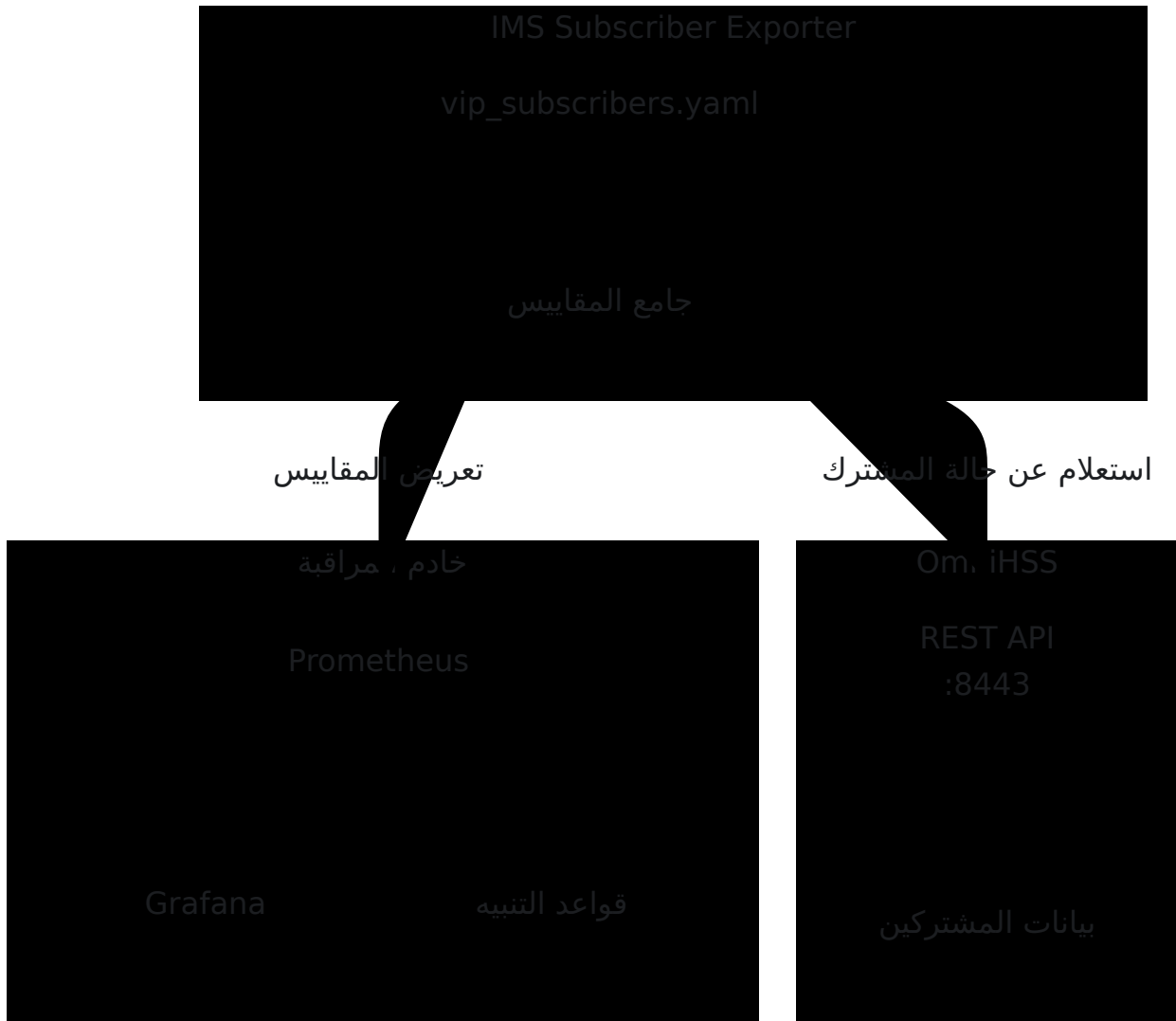
3. اختبار الاتصال المحلي:

```
curl http://localhost:9090/api/v1/status/config  
curl http://localhost:3100/ready
```

VIP مراقبة المشتركين

تتبع حالة في الوقت الفعلي للمشتركين الحرجين. يراقب النظام VIP تقدم مراقبة المشتركين للمشتركين ذوي الأولوية العالية مثل خدمات UE وقابلية الوصول إلى EPC، وإرفاق IMS، تسجيل الطوارئ، والمستشفيات، والبنية التحتية الرئيسية.

الهيكلية



أنواع الخدمات

يتم تصنيف المشتركين حسب نوع الخدمة المتوقع، مما يحدد تقييم الصحة

النوع	مطلوب IMS	مطلوب EPC	حالة الاستخدام
full	نعم	نعم	مستخدمون عاديون مع صوت وبيانات
voip_only	نعم	لا	هواتف، IP هواتف) فقط VoIP أجهزة (برمجية)
data_only	لا	نعم	أجهزة توجيه) أجهزة بيانات فقط (IoT، M2M)

تقييم الصحة:

- **full**: EPC و (ليس فارغًا assigned_scscf) مسجلًا IMS صحي عندما يكون كل من (ليس فارغًا last_seen_mme) متصلًا
- **voip_only**: (ليس فارغًا assigned_scscf) مسجلًا IMS صحي عندما يكون
- **data_only**: (ليس فارغًا last_seen_mme) متصلًا EPC صحي عندما يكون

التكوين

في متغيرات مجموعة المضيف VIP يتم تكوين المشتركين

الملف: hosts/<customer>/group_vars/vip_subscribers.yaml

```
vip_subscriber_monitoring:
  hss_url: "https://10.80.12.140:8443"

  # إعدادات المراقبة
  scrape_interval_seconds: 30
  ping_timeout_seconds: 2
  ping_count: 2

  # المشتركين الذين يجب مراقبتهم
  subscribers:
    # مشتركون بخدمة كام000 (IMS + EPC)
    - imsi: "0010100000000001"
      label: "خدمات الطوارئ"
      type: "full"

    - imsi: "0010100000000002"
      label: "البنية التحتية الحرجة"
      type: "full"

    # IMS لا يتوقع - IoT، أجهزة توجيه) خدمات بيانات فقط
    - imsi: "0010100000000003"
      label: "جهاز التوجيه في الموقع"
      type: "data_only"

    # IMS، بيانات) فقط VoIP خدمات (لا حامل بيانات
    - imsi: "0010100000000004"
      label: "IP هواتف"
      type: "voip_only"
```

معلومات التكوين

الوصف	الافتراضي	مطلوب	النوع	المعلمة
برمجة URL عنوان تطبيقات OmniHS (HTTPS)	-	نعم	سلسلة	hss_url
رة يتم استعلام حالة المشارك	30	لا	عدد صحيح	scrape_interval_seconds
IP مهلة لاختبارات UE	2	لا	عدد صحيح	ping_timeout_seconds
سلة ping عدد حزم	2	لا	عدد صحيح	ping_count
لمصدر URL عنوان blackbox اختبارات ping (مثل http://pcscf01 ICM يستخدم اختبار محلي ping بدلاً من	-	لا	سلسلة	blackbox_exporter_url
ة بالمشاركين الذين يجب مراقبتهم	-	نعم	قائمة	subscribers

معلومات المشترك

الوصف	الافتراضي	مطلوب	النوع	المعلمة
المشارك (15 رقمًا) IMSI	-	نعم	سلسلة	imsi
اسم قابل للقراءة البشرية للعرض	IMSI	لا	سلسلة	label
أو full, voip_only, نوع الخدمة data_only	full	لا	سلسلة	type

ولا تحتاج إلى HSS تلقائيًا من استجابة واجهة برمجة التطبيقات MSISDN **ملاحظة:** يتم استرداد تكوينها.

المقاييس

.`/metrics` يعرض المصدر المقاييس على المنفذ 9550 في

مقاييس الحالة

المقياس	النوع	الوصف
<code>vip_subscriber_service_healthy</code>	Gauge	إذا كان المشترك يلبي معايير 1 الصحة لنوع خدمته، 0 خلاف ذلك
<code>vip_subscriber_ims_registered</code>	Gauge	S-CSCF إذا كان لدى المشترك 1 المعين، 0 خلاف ذلك
<code>vip_subscriber_epc_registered</code>	Gauge	إذا كان لدى المشترك إرفاق 1 نشط، 0 خلاف ذلك MME
<code>vip_subscriber_ue_ip_reachable</code>	Gauge	ping، لاختبار IP UE إذا استجاب 1، خلاف ذلك 0
<code>vip_subscriber_hss_reachable</code>	Gauge	إذا أعادت واجهة برمجة 1 بيانات صالحة، 0 HSS التطبيقات خلاف ذلك
<code>vip_subscriber_enabled</code>	Gauge	إذا كان حساب المشترك مفعلاً 1 خلاف ذلك 0 HSS، في

مقاييس العمر

المقياس	النوع	الوصف
<code>vip_subscriber_ims_registration_age_seconds</code>	Gauge	ثوانٍ منذ آخر IMS (-1) تسجيل إذا لم يكن مسجلاً
<code>vip_subscriber_epc_registration_age_seconds</code>	Gauge	ثوانٍ منذ آخر تحديث موقع EPC (-1) إذا لم يكن متصلًا

مقياس المعلومات

المقياس	النوع	الوصف
<code>vip_subscriber_info</code>	Gauge	دائمًا 1، يحمل جميع الحالة والمعلومات كعلامات

التسميات على جميع المقاييس:

التسمية	الوصف	المثال
<code>imsi</code>	المشارك IMSI	<code>0010100000000002</code>
<code>label</code>	اسم قابل للقراءة البشرية	<code>البنية التحتية الحرجة</code>
<code>msisdn</code>	(من HSS) رقم الهاتف	<code>614000000002</code>
<code>type</code>	نوع الخدمة	<code>full</code> ، <code>voip_only</code> ، <code>data_only</code>

تسميات إضافية على `vip_subscriber_info`:

المثال	الوصف	التسمية
أو 0 1	حالة صحة الخدمة	healthy
أو 0 1	IMS حالة تسجيل	ims
أو 0 1	EPC حالة إرفاق	epc
أو 0 1	UE IP قابلية الوصول إلى	ping
scscf01.ims.example.com	S-CSCF اسم مضيف	assigned_scscf
mme02.epc.example.com	MME اسم مضيف	last_seen_mme
100.72.83.20	UE المعين لـ IP عنوان	ue_ip

استعلامات المثال

```
# الصحيحين VIP عدد المشتركين
count(vip_subscriber_service_healthy == 1)

# غير الصحيحين VIP عدد المشتركين
count(vip_subscriber_service_healthy == 0)

# VoIP لخدمات IMS حالة تسجيل
vip_subscriber_ims_registered{type=~"voip_only|full"}

# لخدمات البيانات EPC حالة إرفاق
vip_subscriber_epc_registered{type=~"data_only|full"}

# قديم (<1 ساعة) IMS المشتركين الذين لديهم تسجيل
vip_subscriber_ims_registration_age_seconds > 3600
```

لوحة المعلومات

VIP رؤية في الوقت الفعلي لحالة المشتركين **VIP IMS** توفر لوحة معلومات **مشتركي**

الموقع: Grafana → IMS → مشتركين VIP IMS

للوحة المعلومات URL عنوان: /d/ims-vip-subscribers/

الألواح

الوصف	اللوحة
عدد المشتركين الذين يليون معايير الصحة	الخدمات الصحية
عدد المشتركين الذين يفشلون في معايير الصحة	الخدمات غير الصحية
نشط IMS عدد المشتركين الذين لديهم تسجيل	IMS مسجل
نشط EPC عدد المشتركين الذين لديهم إرفاق	EPC مسجل
قابل للاختبار IP UE عدد المشتركين الذين لديهم	UE قابلية الوصول إلى
المكونين VIP العدد الإجمالي للمشاركين	إجمالي المراقبين
جدول يعرض جميع المشتركين مع أعمدة الحالة	VIP حالة المشترك
خط زمني للحالة يظهر تاريخ الصحة	صحة الخدمة مع مرور الوقت

التنبيه

غير صحية VIP تثير قواعد التنبيه عندما تصبح خدمات المشتركين

غير صحية VIP خدمة المشترك

الخطورة: حرجة لمدة: 1 دقيقة VIP Subscriber Service Unhealthy **التنبيه**

الشرط: vip_subscriber_service_healthy == 0

التعليقات التوضيحية:

- غير صحي VIP Subscriber {{ \$labels.label }} **الملخص:**
- الوصف:** يتضمن نوع الخدمة والمعرف المناسب
 - IMSISDN الاسم و VoIP خدمات
 - IMSI خدمات البيانات: الاسم و

- IMSI و MSISDN الخدمات الكاملة: الاسم و

أمثلة على أوصاف التنبهات

- متوقفة (voip_only) خدمة هواتف . MSISDN: 61400000001
- متوقفة (data_only) خدمة جهاز التوجيه في الموقع . IMSI: 001010000000003
- متوقفة (full) خدمة البنية التحتية الحرجة . MSISDN: 61400000002
IMSI: 001010000000002

جدد VIP إضافة مشتركين

1. تحرير ملف التكوين:

```
# hosts/<customer>/group_vars/vip_subscribers.yaml
subscribers:
- imsi: "123456789012345"
  label: "جديد VIP مشترك"
  type: "full" # أو voip_only, data_only
```

2. نشر التكوين المحدث:

```
scp vip_subscribers.yaml <monitoring-server>:/tmp/
ssh <monitoring-server> "sudo cp /tmp/vip_subscribers.yaml
/etc/prometheus/vip_subscribers.yaml && sudo systemctl
restart ims-subscriber-exporter"
```

3. تحقق من ظهور المشترك الجديد في المقاييس:

```
curl -s http://<monitoring-server>:9550/metrics | grep
"جديد VIP مشترك"
```

الوثائق ذات الصلة

- Alloy و Loki [التسجيل المركزي](#): تفاصيل تكوين

- **معمارية النشر:** المعمارية العامة للنظام
- **تكوين ملف المضيفين:** تكوين الجرد

Netplan تكوين

نظرة عامة

تكوين واجهات الشبكة تلقائيًا على الأجهزة الافتراضية الموزعة باستخدام OmniCore يمكن لـ netplan. هذا مفيد لـ:

- (eth0) إعداد واجهة الإدارة الأساسية
- العامة، أو اتصالات التبادل، أو حركة المرور المخصصة IP إضافة واجهات ثانوية لعناوين
- تكوين المسارات الثابتة لواجهات محددة

Netplan تمكين تكوين

الذي يشير إلى قالب `netplan_config` التلقائي لمضيف، أضف المتغير netplan لتمكين تكوين الخاص بك `group_vars` في مجلد Jinja2:

```
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

مشارك أو قياسي. يقوم كل عميل بإنشاء خاص به `netplan.yaml.j2` لا يوجد وتخلف هذه القوالب - ، `hosts/<customer>/group_vars/` في `netplan.yaml.j2` وتسمية الواجهة الثانوية تختلف حسب العميل. عند DNS اسم الواجهة الأساسية، مجال البحث إدخال عميل جديد، أنشئ قالبًا خاصًا بالعميل يتناسب مع تخطيط واجهته بدلاً من نسخ قالب عميل آخر حرفيًا.

فيما يلي قالب عميل واحد كمثال. إنه توضيحي، وليس مواصفة - العملاء الآخرون يختلفون. تم
إضافة تعليقات هنا للتوضيح:

```

network:
  version: 2
  ethernet:
    # من الجرد gateway و ansible_host الواجهة الأساسية - تستخدم
    eth0:
      addresses:
        - "{{ ansible_host }}/{{ mask_cidr | default(24) }}"
      nameservers:
        addresses:
{% if 'dns' in group_names %}
          # الخارجي لتجنب DNS استخدم، DNS إذا كان هذا المضيف هو خادم
          الاعتماد الدائري
          - 8.8.8.8
{% else %}
          # في الجرد 'dns' من مجموعة DNS خلاف ذلك، استخدم خوادم
{% for dns_host in groups['dns'] | default([]) %}
          - {{ hostvars[dns_host]['ansible_host'] }}
{% endfor %}
{% endif %}
          خاص بالعميل (يختلف حسب العميل؛ بعضهم لا - DNS مجال البحث
          يحددون أي شيء)
          search:
            - customer.example
          routes:
            - to: "default"
              via: "{{ gateway }}"

{% if secondary_ips is defined %}
  # من الجرد dict secondary_ips الواجهات الثانوية - حلقة عبر
  # هذا ens19، ens20، ens21... (18 + loop.index).
  هذا
  # خاص بال قالب؛ انظر "تسمية الواجهة" أدناه.
{% for nic_name, nic_config in secondary_ips.items() %}
  ens{{ 18 + loop.index }}:
    addresses:
      # (يكون افتراضيًا 24) NIC الخاص بكل mask_cidr القناع يأتي من
      - "{{ nic_config.ip_address }}/{{ nic_config.mask_cidr |
default(24) }}"
{% if nic_config.routes is defined %}
      # مسارات ثابتة لهذه الواجهة - كل مسار يستخدم بوابة هذه
      الواجهة
      routes:
{% for route in nic_config.routes %}

```

```
- to: "{{ route }}"
  via: "{{ nic_config.gateway }}"
{% endfor %}
{% endif %}
{% endfor %}
{% endif %}
```

النقاط الرئيسية:

- تأتي من إدخال الجرد الخاص بالمضيف `gateway` و `ansible_host`
- `dns` ديناميكيًا من المضيفين في مجموعة DNS يتم سحب خوادم
- خاص بالعمي ❖❖ (يختلف؛ بعض العملاء يتجاهلونه) DNS مجال البحث
- `ens19`، `ens20`، ...! بعضهم يستخدم) تختلف تسمية الواجهة الثانوية حسب العميل (انظر أدناه)
- وبوابة، ومسارات ثابتة، (`mask_cidr`) ثانوي قناع خاص به IP يمكن أن يكون لكل

تكوين الواجهة الأساسية

تلقائيًا باستخدام (eth0) يتم تكوين الواجهة الأساسية:

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

تلقائيًا إلى DNS يتم تعيين خوادم:

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

MTU (link_mtu) الرابط

netplan، **الرابط** للواجهة الأساسية بشكل مستقل عن قالب **MTU** تثبيت `common` يمكن لدور drop-in عند تعيينه، فإنه يقوم بإنشاء `link_mtu` عبر متغير الجرد على الواجهة الأساسية للمضيف ويطبقه. هذا موجود (`/etc/netplan/999-link-mtu.yaml`)
تم شحنه من مجموعة MTU مع أي VMware/Proxmox لأن الأجهزة الافتراضية تأتي من قالب

مما يؤدي إلى عدم تطابق مع بقية النسيج ويؤدي إلى فقدان (jumbo/9000 أحيانًا) المنفذ المصدر معطلاً PMTUD عندما يكون SCTP/SIP الحزم الكبيرة.

بدمج جميع netplan عمداً بـ 999- بحيث يتم فرزها أخيراً: يقوم drop-in تم تسمية بترتيب أبجدي مع فوز آخر ملف لكل مفتاح، ويمكن أن تترك عمليات `/etc/netplan/*.yaml` `-ansible-` ملف تكوين قديم (80) على نفس الواجهة `mtu` التهيئة ملفات متنافسة تحدد ذو drop-in سيتم تجاوز (VMware) لمحرك `netcfg-vmware.yaml` وملف 99 `config.yaml` الرقم الأقل بهدوء بواسطة تلك.

```
all:
  vars:
    link_mtu: 1500 # الأساسي لكل مضيف على 1500 NIC تثبيت
```

لتطبيقه على مستوى الموقع، أو لكل مضيف للتجاوز. إذا لم يتم تعريف `all.vars` قم بتعيينه في `link_mtu`، الواجهة دون تغيير MTU فإن الدور يترك.

GTP/tunnel الحمولة MTU على المستوى الأعلى هو `mtu` المتغير. `link_mtu` ليس `mtu` رابط. سيكون MTU ليس `packet-core` الذي تستهلكه أدوار (مثل 1400/1430/1300) الواجهة MTU المخصص لـ `link_mtu` الفيزيائي لتلك القيم خاطئاً. استخدم دائماً NIC تعيين

الواجهات الثانوية

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

المخطط

```
secondary_ips:
  <logical_name>:
    ip_address: <ip_address>
    mask_cidr: <prefix_length> # افتراضيًا 24، NIC اختياري - قناع لكل
    gateway: <gateway_ip>
    host_vm_network: <proxmox_bridge>
    vlanid: <vlan_id>
    nic_name: <interface_name> # اختياري - تجاوز اسم الواجهة المحدد
    # (مدعوم من القوالب التي تحترم ذلك)
    routes: # اختياري - مسارات ثابتة عبر هذه الواجهة
      - '<destination_cidr>'
      - '<destination_cidr>'
    table_routes: # اختياري - متقدم، خاص بالقالب
      - { to: '<cidr>', via: '<gateway>', table: <table_id> }
    routing_policy: # اختياري - متقدم، خاص بالقالب
      from: '<source_cidr>'
      via: '<gateway>'
      table: <table_id>
      priority: <priority> # اختياري، افتراضيًا 100
```

يحدد طول البادئة لتلك الواجهة الثانوية الواحدة. ي❖❖م قراءته من `mask_cidr` المفتاح على `mask_cidr` وليس من `(nic_config.mask_cidr)` الثانوي IP داخل إدخال المستوى الأعلى. إذا تم تجاهله، فإنه يكون افتراضيًا إلى 24/

تمكّن التوجيه حسب السياسة/المصدر ولكن يتم `routing_policy` و `table_routes` عرضها فقط بواسطة القوالب التي تنفذها. التوجيه المتقدم خاص بالقالب. تأكد من أن `netplan.yaml.j2` للتعديل يحترم هذه المفاتيح قبل استخدامها

تسمية الواجهة

تسمية الواجهة الثانوية خاصة بالقالب - ليست موحدة عبر العملاء

- `ens19`، `ens20`، `ens21`، ... (18 + `loop.index`): بعض العملاء يسميون الواجهات حسب الفهرس عند إضافة Proxmox يتطابق هذا مع أسماء الواجهات المعينة بواسطة VM. إضافية إلى NICs
- صراحة عند تعيينه، ويتراجعون إلى `nic_config.nic_name` يستخدم عملاء آخرون خلاف ذلك (`ens(34 + loop.index)` مثل) اسم قائم على الفهرس

يمكنك تثبيت واجهة ثانوية على اسم محدد عن طريق تعيين `nic_name`، إذا كان القالب يحترم `nic_name` الثاني؛ خلاف ذلك، يتم تطبيق مخطط الفهرس الخاص بالقالب IP على إدخال `nic_name`.

مثال على التكوين

```
dra:
  hosts:
    <hostname>:
      ansible_host: 10.0.1.100
      gateway: 10.0.1.1
      host_vm_network: "ovsbr1"
      vlanid: "100"
      netplan_config: netplan.yaml.j2
      secondary_ips:
        public_ip:
          ip_address: 192.0.2.50
          mask_cidr: 28
          gateway: 192.0.2.1
          host_vm_network: "vibr0"
          vlanid: "200"
          routes:
            - '198.51.100.0/24'
            - '203.0.113.0/24'
        peering_ip:
          ip_address: 172.16.50.10
          gateway: 172.16.50.1
          host_vm_network: "ovsbr2"
          vlanid: "300"
          routes:
            - '172.17.0.0/16'
```

Netplan الناتج عن

واسم DNS مجال البحث) تم عرض التكوين أعلاه باستخدام القالب المثال، مما ينتج عنه `ens19/ens20` خاص بالعميل):

```

network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - "10.0.1.100/24"
      nameservers:
        addresses:
          - 10.0.1.53
        search:
          - customer.example
      routes:
        - to: "default"
          via: "10.0.1.1"
    ens19:
      addresses:
        - "192.0.2.50/28"
      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"

```

بينما تتراجع ، mask_cidr: 28 ك 28/ لأن إدخالها حدد public_ip تظهر واجهة
إلى الافتراضي 24/ peering_ip

Proxmox تكامل

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

1. الثانية أثناء التهيئة الأولية NICs الأجهزة الافتراضية الجديدة: تتم إضافة
2. VM الثانية ويتم إعادة تشغيل NICs الأجهزة الافتراضية الموجودة: تتم إضافة
لتطبيق التغييرات

Proxmox تستخدم تكوين

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

كيف يعمل



1. `/etc/netplan` الموجودة في netplan يتم عمل نسخة احتياطية من تكوينات ثم يتم إزالتها لتجنب التعارضات (`*.bak`)
2. `/etc/netplan/01-netcfg.yaml` الخاص بالعميل إلى Jinja2 يتم عرض قالب (`وضع 0600`)
3. بتطبيق التكوين مع التراجع التلقائي إذا فقد `netplan try --timeout 30` يقوم الاتصال
4. أن المضيف ظل قابلاً للوصول `wait_for_connection` يؤكد
5. بتأكيد التكوين `netplan apply` يقوم

حالات الاستخدام الشائعة

عام IP مع (DEA) Diameter Edge Agent

```
<hostname>:
ansible_host: 10.0.1.100          # IP الإدارة الداخلية
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: "vmbro"
    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 الاتصال مع شبكات إدارة وبيانات منفصلة

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

الثانوية في القوالب IPs الإشارة إلى

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

على نفس المضيف

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

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

# نفس النتيجة (inventory_hostname أو صراحة عبر)
{{ hostvars[inventory_hostname]['secondary_ips']
  ['diameter_public_ip']['ip_address'] }}

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

من مضيف آخر

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

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

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

DRA مثال: تكوين نظير

العام الخاص به IP للربط إلى diameter تكوين نظير:

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

الثانوية موجودة IPs التحقق مما إذا كانت

تحقق دائمًا مما إذا كانت المتغيرات موجودة قبل استخدامها

```
{% 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

وتتحقق من أسماء الواجهات VM قم بتسجيل الدخول إلى

```
ip link show
```

النتائج المتوقعة لجهاز افتراضي به واجهتين ثانويتين

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

Netplan تحقق من تكوين

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

يدويًا Netplan تطبيق

```
netplan apply
```

Netplan تصحيح

```
netplan --debug apply
```

تحقق من المسارات

```
ip route show
```

الوثائق ذات الصلة

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

عمليات النسخ المتماثل و OmniHSS العقد في

في قاعدة بيانات IMS وبيانات (AuC) بيانات المشتركين، والمصادقة OmniHSS تخزن محلية على كل عقدة. **النسخ المتماثل المنطقي ثنائي الاتجاه في PostgreSQL** يشكل هذا **شبكة نشطة-نشطة**: كل OmniHSS يحافظ على تزامن عدة عقد **PostgreSQL** عقدة هي رئيسية كاملة للقراءة/الكتابة، والكتابة على أي عقدة تنتشر إلى جميع العقد الأخرى. لا وتقبل التهيئة في أي Cx/Dx و S6a يوجد رئيس/احتياطي. يمكن لأي عقدة أن تخدم حركة مرور وقت.

جديدة وتحميلها OmniHSS يغطي هذا الدليل نموذج النسخ المتماثل. يشرح كيفية إضافة عقدة بالبيانات. كما يشرح كيفية التحقق من الصحة، وإصلاح، وإزالة الأقران.

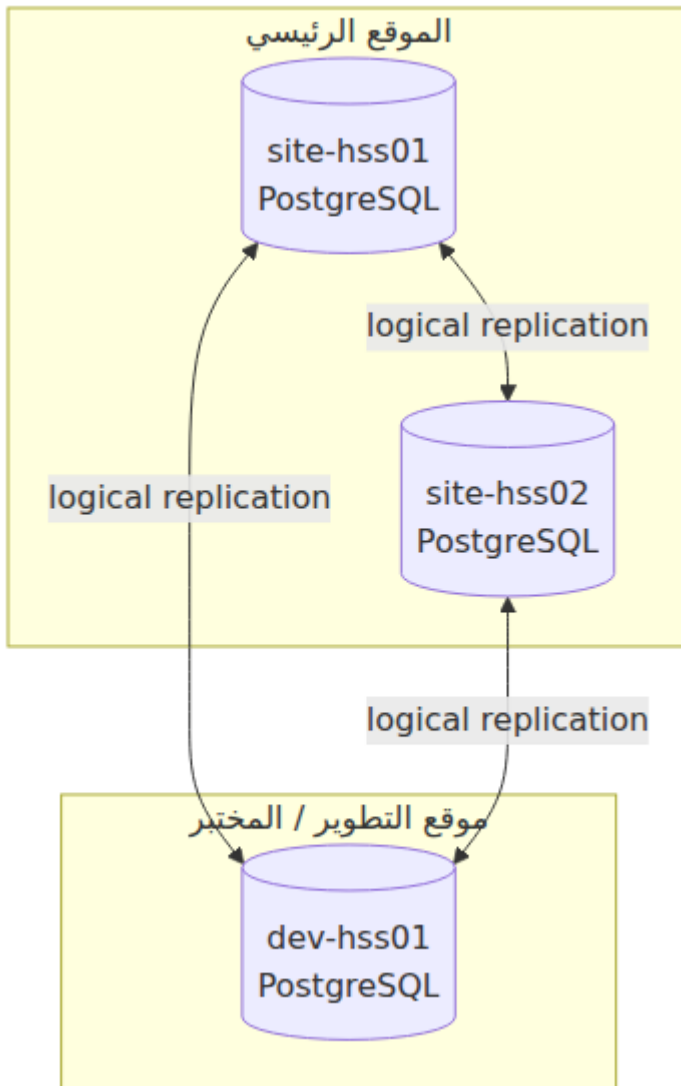
جدول المحتويات

- نظرة عامة على المعمارية
- كيفية اكتشاف الأقران
- التكوين
- جديدة OmniHSS إضافة عقدة
- تحميل عقدة جديدة بالبيانات
- التحقق من النسخ المتماثل
- سلامة البيانات
- إزالة نظير
- النسخ الاحتياطي والاستعادة
- المقاييس
- استكشاف الأخطاء وإصلاحها

نظرة عامة على المعمارية

خاص بها. كل عقدة **تنشر** جميع PostgreSQL على تشغيل مثل OmniHSS تعمل كل عقدة يمنع حلقات النسخ `origin = none`. جداول التطبيق الخاصة بها و **تشتترك** في نشر كل نظير

المتماثل. على جداول الحالة المتغيرة بسرعة، تحل الشبكة تعارضات الكتابة آخر كتابة تفوز updated_at من خلال مقارنة (LWW).

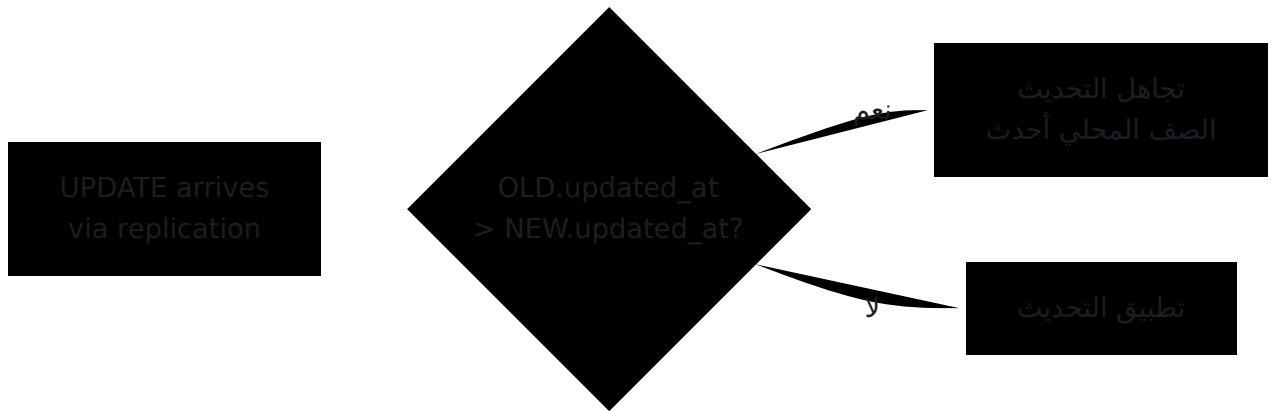


عقد، تحتفظ كل عقدة بـ N كل رابط أعلاه هو اشتراك أحادي الاتجاه (واحد لكل اتجاه). لشبكة من $N-1$ من فتحات النسخ المتماثل الخاصة بنشرها $N-1$ اشتراكات. تستهلك الأقران $N-1$.

ما الذي يتم نسخه متماثلاً

معالجة التعارض	أمثلة	فئة الجدول
لا حاجة: يتم تهيئتها من مكان واحد في كل مرة	<code>subscriber</code> , <code>key_set (AuC)</code> , <code>sim</code> , <code>msisdn</code>	البيانات الرئيسية / المهياة
آخر كتابة تفوز على <code>updated_at</code>	<code>subscriber_state</code> , <code>pdn_session</code> , <code>lte_call</code> , <code>ims_subscription</code>	الحالة الديناميكية
مستبعدة من النسخ المتماثل	<code>schema_migrations</code>	محاسبة الهجرة

تقوم كل عقدة (`schema_migrations` باستثناء `public` يغطي النشر كل جدول في مخطط عندما تقوم عقدتان بتحديث نفس صف الحالة في وقت واحد، (بتشغيل هجراتها بشكل مستقل بدلاً من أي تحديث وصل أخيراً `updated_at` على أحدث LWW تحافظ مشغلات



كيفية اكتشاف الأقران

:بناء قائمة الأقران تلقائياً من مصدرين OmniHSS عندما يتم تمكين النسخ المتماثل، يقوم دور

- المعرفة OmniHSS المحلية: عقد `hss` المضيفون الآخرون في مجموعة جرد 1. لا (`hss01` / `hss02` على سبيل المثال، زوج محلي) في نفس ملف الجرد/المضيف ؛ يكفي أن تكون عضواً في `connected_omnihss` تحتاج هذه إلى أن تكون مدرجة في المجموعة.

2. OmniHSS في ملف الأقران البعيد للموقع: عقد `connected_omnihss` قائمة المعرفة في ملف مضيف مختلف (مواقع أخرى). يمكن اكتشاف الأقران عبر الملفات الخاصة بآخر `hss` فقط هنا، لأن جرد واحد ليس لديه معرفة بمجموعة

يقوم الدور بتصفية  أقران من المجموعة المحلية إلى المضيفين الذين هم مشاركون في `omnihss.replication.enabled` أولئك الذين تكون قيمة OmniHSS: النسخ المتماثل لـ لا يتم اختيارها كقرين نسخ متماثل `true` العقدة التي ليست علامتها `true` لديهم

يستبعد الدور المضيف الحالي من قائمة أقرانه تلقائيًا، بحيث يمكن مشاركة نفس كتلة من قبل كل موقع دون محاولة عقدة النسخ من نفسها `connected_omnihss`

تسمية الكائنات

تسمي كل عقدة كائنات النسخ المتماثل الخاصة بها بحيث لا تتداخل شبكة بأي حجم

فريد لكل	تم إنشاؤه في	الاسم	كائن
العقدة	هذه العقدة	<code><self>_pub</code>	النشر
(المشترك)	هذه العقدة (المشترك)	<code>sub_from_<source></code>	الاشتراك
(المشترك، الزوج (المصدر	المصدر (الناشر)	<code><subscriber>_from_<source></code>	فتحة النسخ المتماثل

الإعداد) يجب أن يتضمن اسم الفتحة **المشترك**. اعتبر ما يحدث إذا كان يتضمن فقط المصدر ستقوم عقدتان تشتركان في (حيث ترث الفتحة اسم الاشتراك، PostgreSQL  افتراضي لـ نفس المصدر بمحاولة إنشاء فتحة بنفس الاسم على ذلك المصدر. ستفشل الاشتراك الثانية بعد ذلك مع "فتحة النسخ المتماثل موجودة بالفعل". إن تضمين المشترك يعطي كل رابط فتحة فريدة عالميًا.

السلامة `--limit` المصالحة و

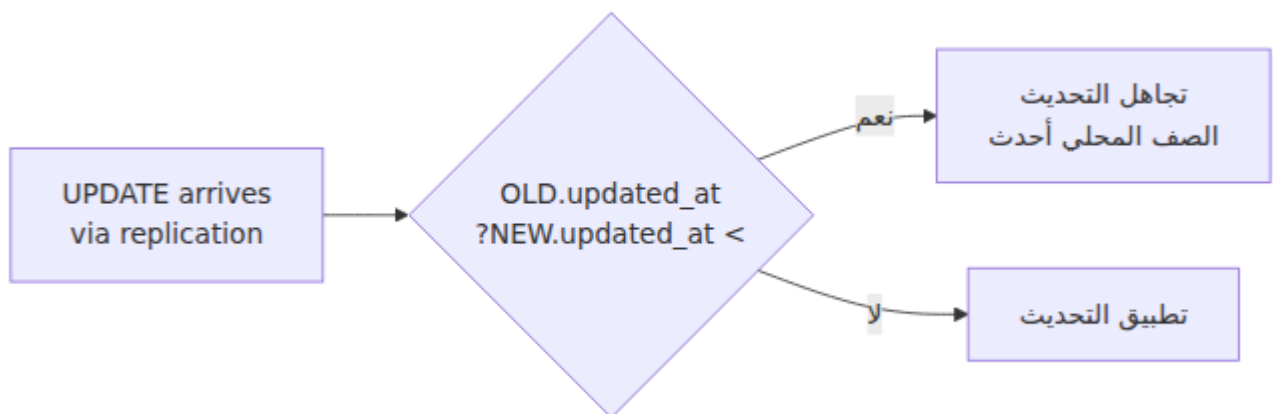
كل تشغيل للدور **يصلح** كائنات النسخ المتماثل للعقدة مقابل مجموعة الأقران المصالحة الحالية (مطروحًا منها الذات، `connected_omnihss` + المحلية `hss` مجموعة)

- تنظف المنشورات القديمة/المعاد `<self>_pub` يتم إسقاط المنشورات الأخرى غير (تسميتها تلقائيًا).
- يتم تعطيل الاشتراكات التي لم يعد نظيرها في المجموعة الصالحة، ويتم فصلها عن فتحها البعيدة، ويتم إسقاطها.
- يتم إسقاط الفتحات المنطقية غير النشطة التي لا تغذي نظيرًا صالحًا حاليًا. لا يتم لمس الفتحات النشطة أبدًا.

قائمة الشبكة **السلطوية**: يقوم التشغيل التالي بإزالة `connected_omnihss` هذا ♦♦♦ يجعل من كل عقدة. تمنع خاصيتان من الأمان هذا (ذات الصلة `hss` ومن مجموعات) قرين تم إزالته منها: من أن يكون مدمرًا

- و `groups['hss']` آمن. يتم اشتقاق المجموعة الصالحة من `--limit` والتي تعكس دائمًا عضوية الجرد الكاملة بغض النظر عن `connected_omnihss` ترى `hss01` ولكن لا تزال `hss01` يصلح فقط `--limit hss01` تشغيل `limit` أبدًا بإزالة `--limit` كصالحين وتحافظ على اشتراكاتهم. لا يقوم `hss02/hss03` الأقران المستبعدة.
- **المجموعات الفارغة لا تقوم بالإزالة.** يتم تخطي خطوات الإزالة تمامًا بدلاً من (على) مسح جميع النسخ المتماثل إذا كانت مجموعة الأقران الصالحة تحسب إلى فارغة (غير محملة `connected_omnihss` واحد و `HSS` سبيل المثال، جرد جزئي مع

تحذير القائمة السلطوية: نظرًا لأن الإزالة تلقائية، يقوم التشغيل بإزالة قرين عبر الملفات تأكد دائمًا من أن `connected_omnihss` يكون جزءًا حقيقيًا من الشبكة ولكنه مفقود من `connected_omnihss` مكتمل لكل قرين عبر الملفات.



التكوين

تمكين النسخ المتماثل

يتم التحكم في النسخ المتماثل لكل نشر بواسطة علامة `omnihss.replication.enabled`.
`group_vars` بشكل افتراضي. قم بتعيينها في جرد الموقع أو `false` وهي

```
omnihss:  
  database_type: "postgres"  
  database_host: "localhost"  
  database_port: 5432  
  database_name: "omnihss"  
  database_username: "hss"  
  database_password: "password"  
  replication:  
    enabled: true
```

المعلمة	النوع	مطلوب	الافتراضي	الوصف
database_type	سلسلة	لا	postgres	قاعدة البيانات الخلفية. ينطبق النسخ المتماثل فقط على postgres.
database_host	سلسلة	لا	localhost	مضيف قاعدة البيانات. يتم تكوين النسخ المتماثل فقط عندما تكون قاعدة البيانات محلية (localhost).
database_port	عدد صحيح	لا	5432	PostgreSQL، يستخدم أيضًا كمنفذ افتراضي للاتصالات بالأقران.
database_name	سلسلة	لا	omnihss	اسم قاعدة البيانات. يجب أن يتطابق عبر جميع الأقران.
database_username	سلسلة	نعم	hss	دور تسجيل الدخول. تم إنشاؤه مع خاصية REPLICATION حتى يتمكن من قيادة النسخ المتماثل المنطقي.
database_password	سلسلة	نعم	password	كلمة المرور لدور تسجيل الدخول. يجب أن تتطابق عبر جميع الأقران (تستخدم في سلاسل اتصال الأقران).

المعلمة	النوع	مطلوب	الافتراضي	الوصف
replication.enabled	منطقي	لا	false	مفتاح رئيسي. عندما يتم true تكون PostgreSQL تكوين للنسخ المتماثل المنطقي ويتم إنشاء المنشورات/الاشتراكات.

PostgreSQL يقوم الدور بتطبيق إعدادات true، هي replication.enabled عندما تكون التالية على المثل المحلي:

الإعداد	القيمة
wal_level	logical
max_wal_senders	10
max_replication_slots	10
listen_addresses	*
max_slot_wal_keep_size	10GB (يمكن تجاوزها عبر omnihss.replication.max_slot_wal_keep_size)

القيمة	الإعداد

و `all` لكل نظير (`32/`) لكل من اتصالات IP للسماح بعنوان `pg_hba.conf` يتم تحديث `replication`.

(`connected_omnihss`) إعلان الأقران عبر المواقع

البعيدة في ملف الأقران البعيد المشار إليه بواسطة متغير OmniHSS تُعلن عقد `remote_peers_file` (عادةً `prod_master_peers.yml`) الخاص بكل موقع:

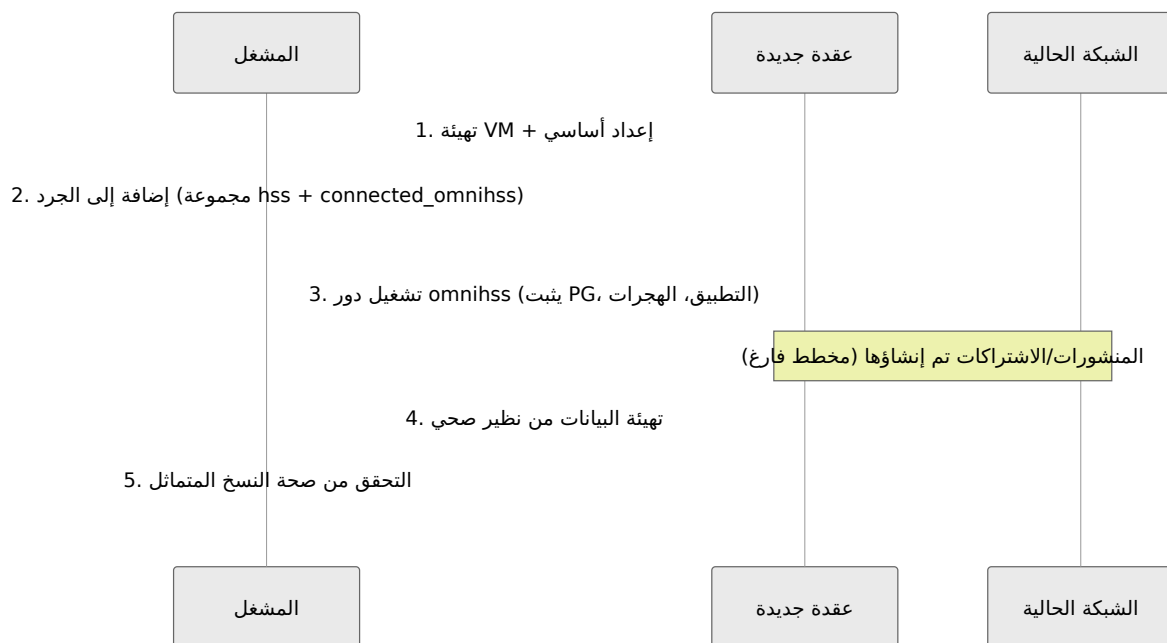
```
connected_omnihss:
- name: site-hss01
  host: 10.4.2.142
  port: 5432
- name: site-hss02
  host: 10.4.2.143
  port: 5432
- name: dev-hss01
  host: 10.4.10.142
  port: 5432
```

المعلمة	النوع	مطلوب	الافتراضي	الوصف
name	سلسلة	نعم	-	للنظير. هذا inventory_hostname (sub_from_<name>) يسمى الاشتراك لذا (<name>_pub) ويحل نشر النظير يجب أن يتطابق مع اسم المضيف الفعلي للنظير.
host	سلسلة	نعم	-	للنظير، يمكن الوصول إليه من IP عنوان هذه العقدة عبر شبكة الإشارة.
port	عدد صحيح	لا	5432	للنظير. يعود إلى PostgreSQL منفذ database_port.

نظرًا لأن ملف الأقران البعيد يتم تحميله بواسطة كل موقع، يجب أن تحتوي القائمة الكاملة لـ connected_omnihss في الشبكة. تتخطى كل موقع تلقائيًا عقدها OmniHSS على كل عقد hss لأنها مغطاة بالفعل بواسطة مجموعة) المحلية وتتخطى نفسها (المحلية hss لأنها مغطاة بالفعل بواسطة مجموعة) المحلية

جديدة OmniHSS إضافة عقدة

إضافة عقدة لها مرحلتان متميزتان: **الانضمام إلى الشبكة** (حتى تتدفق التغييرات في كلا الاتجاهين) و **تحميل البيانات الموجودة** على العقدة الجديدة. تهم المرحلة الثانية لأن العقدة الجديدة تمامًا تبدأ بقاعدة بيانات فارغة. انظر **تحميل عقدة جديدة بالبيانات**



الخطوة 1: تهيئة المضيف

(Proxmox انظر نشر) OmniCore والشبكة الأساسية كما هو الحال في أي عقدة VM قم بتهيئة لكل نظير (5432 بشكل افتراضي) عبر شبكة PostgreSQL يجب أن تصل العقدة إلى منفذ الإشارة، ويجب أن تتمكن الأقران من الوصول إليها.

الخطوة 2: إضافة العقدة إلى الجرد

hss: أضف المضيف الجديد إلى ملف مضيف الموقع تحت مجموعة

```
hss:
  hosts:
    site-hss01:
      ansible_host: 10.4.2.142
      gateway: 10.4.2.1
      host_vm_network: "v401-omni-signalling"
    site-hss03: # عقدة جديدة
      ansible_host: 10.4.2.144
      gateway: 10.4.2.1
      host_vm_network: "v401-omni-signalling"
```

في ملف الأقران البعيد حتى تتعلم المواقع الأخرى عنه connected_omnihss ثم أضفه إلى أيضًا:

```
connected_omnihss:
  - name: site-hss03
    host: 10.4.2.144
    port: 5432
# ... إدخلات موجودة ...
```

OmniHSS الخطوة 3: تشغيل دور

و PostgreSQL ضد المضيف الجديد. يقوم هذا بتثبيت OmniHSS قم بتشغيل كتاب خدمة ویش❖❖ ل هجرات قاعدة البيانات (إنشاء مخطط فارغ)، ويقوم بتكوين النسخ، OmniHSS، المتماثل المنطقي، ويقوم بإنشاء النشر بالإضافة إلى اشتراك لكل نظير:

```
ansible-playbook -i hosts/Customer_Name/host_files/Production.yml \
  services/omnihss.yml --limit site-hss03
```

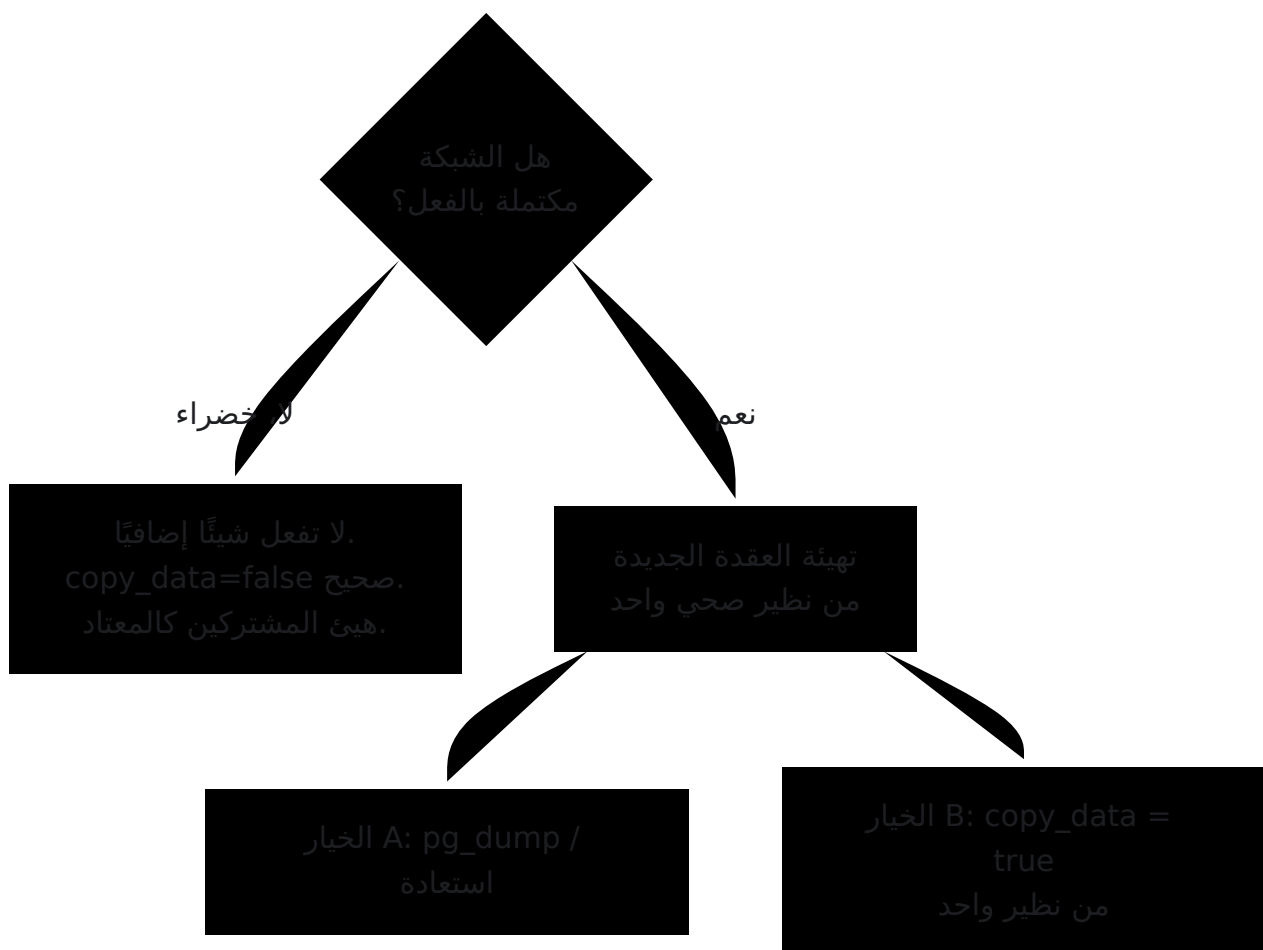
أعد تشغيل الدور على الأقران الحالية (أو دع التشغيل المجدول التالي يتعامل معها) حتى تفتح للعقدة الجديدة وتقوم بإنشاء اشتراكها مرة أخرى. الاشتراكات غير متغيرة: `pg_hba.conf` `ALTER SUBSCRIPTION ... REFRESH PUBLICATION` يكتشف الدور ويخطي الاشتراكات الموجودة، ويصدر حتى تلتقط أي جداول جديدة.

في هذه المرحلة، أصبحت العقدة الجديدة عضوًا في الشبكة وتلقى جميع الكتابات المستقبلية. لكنها لا تحتوي بعد على قاعدة المشتركين الموجودة. تابع لتحميل بياناتها.

تحميل عقدة جديدة بالبيانات

هذا صحيح لشبكة **خضراء** (تبدأ `copy_data = false`). يقوم الدور بإنشاء الاشتراكات الآلية مع كل عقدة فارغة في نفس الوقت: لا يوجد شيء للنسخ، والنسخ سيؤدي إلى تعارضات صفوف مكررة). ولكن هذا يعني أن العقدة التي تنضم إلى شبكة **مكتملة بالفعل** تبقى فارغة باستثناء التغييرات التي تحدث بعد انضمامها.

:اختر المسار الذي يتناسب مع حالتك



شبكة خضراء

معًا دون بيانات مسبقة، فلا تحتاج إلى التهيئة. قم OmniHSS إذا كنت تقوم بإنشاء جميع عقد
بتمكين النسخ المتماثل، وشغل الدور عبر جميع العقد، وقم بتهيئة المشتركين بشكل طبيعي. كل
كتابة تتكرر إلى جميع الأعضاء.

تهيئة عبر النسخ والاستعادة (موصى به): الخيار A

خذ لقطة من نظير صحي واستعدّها على العقدة الجديدة. ثم دع النسخ المتماثل يحمل التغييرات
المستمرة. يعيد هذا استخدام أدوات النسخ الاحتياطي/الاستعادة القياسية وهو النهج الأكثر توقعًا، بما
في ذلك لمجموعات البيانات الكبيرة.

1. لقاعدة بيانات `pg_dump` قم بعمل نسخة احتياطية من نظير صحي. ينتج هذا
بحيث لا، `--no-publications --no-subscriptions` تم أخذه مع OmniHSS
على آلة التحكم (يتم نقل كائنات النسخ المتماثل

```
ansible-playbook -i
hosts/Customer_Name/host_files/Production.yml \
services/backup.yml --limit site-hss01
```

يتم حفظ النسخة الاحتياطية كـ

```
backups/<Site_Name>/hss_dump_<hostname>_<timestamp>.sql.
```

2. ويسقط OmniHSS **استعادة على العقدة الجديدة**. يتوقف كتاب الاستعادة عن
ويعيد إنشاء قاعدة البيانات، ويستعيد النسخة الاحتياطية، ويعيد تشغيل الخدمة. يأخذ
نسخة احتياطية قبل الاستعادة من الهدف أولاً:

```
ansible-playbook -i
hosts/Customer_Name/host_files/Production.yml \
util_playbooks/restore_hss.yml --limit site-hss03
```

من الخطوة 1 وأترك مسار SQL عند المطالبة، قدم المسار إلى النسخة الاحتياطية
التكوين فارغاً (لأن العقدة الجديدة لديها تكوينها الخاص من تشغيل الدور).

3. **تأكيد تدفق النسخ المتماثل**. الآن تطبق الاشتراكات الموجودة للعقدة الجديدة.
التغييرات على القيمة من النسخة الاحتياطية المستعادة. قم (`copy_data = false`)
بتشغيل **التحقق** أدناه.

نافذة الصيانة: هناك نافذة صغيرة بين النسخة الاحتياطية واللحظة التي يلحق فيها النسخ
المتماثل. خلال ذلك، لم تعد تغييرات التهيئة على الأقران موجودة على العقدة الجديدة. تتغير
(`subscriber_state`) بيانات المشترك الرئيسية بشكل نادر، وتتعاوى الحالة الديناميكية
من حركة المرور الحية، لذا فإن (`pdn_session`, `lte_call`, `ims_subscription`)
نافذة النشاط المنخفضة كافية. لا تقم بتهيئة مشتركين جدد خلال التهيئة.

ت؟؟ يئة باستخدام كتاب التهيئة (موصى به): B الخيار

بالتهيئة من **مصدر** نظير مختار واحد دون `seed_hss_replication.yml` يقوم كتاب الخدمة
لا يوجد (`copy_data = false`) عيوب النسخ الخام. يعيد إنشاء الاشتراك كاشتراك فقط للتدفق
ثم يقوم بملء الصفوف الموجودة في **الجدول الرئيسي** بتحميل آمن من (جداول متزامنة
كل مشترك يشير) **هو** جزء من الملء: إنها بيانات هيكلية 1:1 `subscriber_state`. التعارض
تركه يعني أن كل مشترك تم ملؤه يصبح. (تم إنشاؤه مرة واحدة عند التهيئة، FK إلى واحد بواسطة
على الهدف. لا يمكن للتدفق إصلاح ذلك، لأن النسخ المتماثل المنطقي S6a و API يتيماً، مما يكسر
يسقط بهدوء التحديثات للصفوف التي لا توجد. لا يقوم كتاب الخدمة عمداً بنسخ الجداول المؤقتة

لتلك الجداول الساخنة (مفتاح COPY يمكن أن يتداخل). (pdn_session, lte_call) الحقيقية مكرر) مع الكتابات المتزامنة على نظام حي ويؤدي إلى حلقات تعطل لجداول التزامن. يتم إعادة ملئها من النسخ المتماثل الحي وحركة المرور الحية بدلاً من ذلك.

تفاعلي (موصى به): اترك المصدر. سيجد كتاب الخدمة بعد ذلك الأهداف الفارغة، ويستجوب الأقران عن الأقران التي تحمل البيانات، ويطلب منك الاختيار:

```
ansible-playbook -i hosts/Customer_Name/host_files/Production.yml \
  util_playbooks/seed_hss_replication.yml \
  --limit site-hss01,site-hss02
```

:الأقران التي تتوفر منها البيانات للتهيئة
مشترك 838 - (10.4.10.142) dev-hss01 (1)
أدخل رقم المصدر للتهيئة منه

صريح: قم بتسمية المصدر مباشرة. هذا يجبر على إعادة التهيئة حتى لو كانت الهدف تحتوي بالفعل على بيانات (استخدم للتعافي):

```
ansible-playbook -i hosts/Customer_Name/host_files/Production.yml \
  util_playbooks/seed_hss_replication.yml \
  --limit site-hss01 \
  -e "seed_source_name=dev-hss01 seed_source_host=10.4.10.142"
```

للحصول على مرجع كامل للمعلومات والسلوك **HSS Replication Seed** انظر

A للعقدة المستهدفة، وتهيئة من مصدر واحد فقط. يفضل الخيار `--limit` دائمًا استخدم لمجموعات البيانات الكبيرة جدًا.

لاحظ) تتم الخطوات اليدوية المعادلة على العقدة المستهدفة. أولاً، قم بإنشاء الاشتراك المتدفق ثم قم بملء الجداول الرئيسية بأمان من. (الصريح لكل رابط، انظر **تسمية الكائنات** slot_name):
التعارض مع تخطي الجداول الساخنة:

-- 1. تدفق جميع الجداول، لا نسخ جماعي

```
DROP SUBSCRIPTION IF EXISTS sub_from_dev_hss01;
CREATE SUBSCRIPTION sub_from_dev_hss01
  CONNECTION 'host=10.4.10.142 port=5432 dbname=omnihss user=hss
password=<omnihss.database_password>'
  PUBLICATION dev_hss01_pub
  WITH (copy_data = false, origin = none,
        slot_name = 'site_hss01_from_dev_hss01');
```

كل جدول -t ملء الجداول الرئيسية فقط (تشغيل على الهدف؛ 2. باستثناء
schema_migrations, pdn_session, lte_call - يجب تضمين subscriber_state,
إلى subscriber_state_id أو يصبح كل مشترك تم ملؤه يتيماً: يشير
صف لا يصل أبدًا، مما يكسر
ولا يمكن للنسخ المتماثل إصلاحه لأن النسخ المتماثل API/S6a، المنطقي
(يسقط بهدوء التحديثات للصفوف المفقودة).
origin=none: يجب أن يكون التحميل مختومًا بأصل النسخ المتماثل
تقوم الاشتراكات الموقعة بتصفية التغييرات الموقعة عند الناشر،
لذا فإن الملء يكون
مثل WAL غير مرئي للشبكة. يتم فك تشفير التحميل غير الموقعة من
أي كتابة محلية
،وتعيد صدى الإدخالات إلى الأقران الذين يحملون الصفوف بالفعل،
مما يؤدي إلى تأخير عمال تطبيقهم بسبب المفتاح المكرر.
sudo -u postgres psql -d omnihss -c \
"SELECT pg_replication_origin_create('omnihss_seed_load');"\
{ echo "SELECT
pg_replication_origin_session_setup('omnihss_seed_load');"\
PGPASSWORD=<pw> pg_dump -h 10.4.10.142 -U hss -d omnihss \
--data-only --inserts --on-conflict-do-nothing --disable-
triggers \
-t subscriber -t subscriber_state -t key_set -t sim -t msisdn
... ; } \
| sudo -u postgres psql -d omnihss
sudo -u postgres psql -d omnihss -c \
"SELECT pg_replication_origin_drop('omnihss_seed_load');"

التحقق من النسخ المتماثل

دور التشغيل يقوم بإجراء فحص صحي تلقائي بعد الإعداد، ويمكن أيضًا تشغيله بمفرده. تؤكد عملية التحقق أن:

- كل اشتراك لديه عملية عامل حية.
- كل فتحة تكرار منطقية نشطة.
- أقل من 100 ميجابايت لكل فتحة WAL تأخير.
- (ليس `lost`) أو `extended` أو `reserved` لكل فتحة هي WAL حالة.

أو، (تتم عملية التحقق في النهاية) OmniHSS للتحقق من الصحة عند الطلب، أعد تشغيل دور: تحقق مباشرة على عقدة

```
# Subscription workers - pid should be non-null for each
sudo -u postgres psql -d omnihss -c \
    "SELECT subname, pid IS NOT NULL AS worker_alive FROM
pg_stat_subscription;"

# Replication slots - active should be true, wal_status
reserved/extended
sudo -u postgres psql -d omnihss -c \
    "SELECT slot_name, active, wal_status, \
        pg_size_pretty(pg_wal_lsn_diff(pg_current_wal_lsn(),
confirmed_flush_lsn)) AS lag \
    FROM pg_replication_slots WHERE slot_type = 'logical';"
```

المعنى	العرض
عامل الاشتراك متوقف: النظير غير متاح أو بيانات الاعتماد/سلسلة الاتصال خاطئة.	<code>worker_alive = false</code>
لا يوجد مشترك يستهلك الفتحة: اشتراك النظير متوقف أو تم إسقاطه.	<code>active = false (slot)</code>
يتراكم ويستهلك القرص WAL. النظير بطيء أو الرابط متدهور.	<code>lag</code> تأخير متزايد
يجب PostgreSQL بواسطة WAL الفتحة تأخرت كثيرًا وتمت إزالة عليك إعادة تهيئة النظير (انظر تحميل عقدة جديدة بالبيانات)	<code>wal_status = lost</code>

للعقد **HSS Postgres** أيضًا لوحة **تكرار** (`health_check.yml`) تتضمن **تقرير الفحص الصحي** WAL التي تم تمكين التكرار فيها، مما يظهر حالة عامل كل اشتراك وحالة كل فتحة نشطة وحالة في لمحّة.

سلامة البيانات

تطبيق التكرار المنطقي لا يفرض المفاتيح الخارجية. يعمل عامل التطبيق مع التي تنفذ قيود RI مما يجمع المحفزات الداخلية، `session_replication_role = replica`، **بإعادة التحقق أبدًا** من صف يصل عبر التكرار PostgreSQL لذلك، لا يقوم FOREIGN KEY. لك سلامة لكل عقدة، وليس سلامة عبر الشبكة بأكملها FK مقابل والده. تعطي قيود

:في شبكة نشطة-نشطة تفتح ثغرة لا يمكن لأي عقدة واحدة أن تلتقطها

- ليس لديها A لأن `ON DELETE RESTRICT` تمر `epc_profile X` تحذف A العقدة في تلك اللحظة X مشترك محلي يشير إلى
- لا يزال X تمر هذه لأن X تشير إلى المشترك 203 إلى (أو خطوة توفير) B العقدة B. موجودًا على
- "X" و "X" → تتقارب كل عقدة على "203. FK تتكرر كلا التغيرين وتطبق بدون فحوصات الخاصة بأي عقدة محليًا FK محذوف": مرجع عالق لم تنتهك

غير قادر على بناء بيانات الاشتراك HSS يجعل `subscriber.epc_profile_id` مرجع عالق ULR/AIR، على `DIAMETER_ERROR_USER_UNKNOWN (5001)` بـ HSS لذلك المشترك. ثم يجيب

على الرغم من أن صف المشترك موجود ومفعّل. تجعل نافذة المخطط قبل البيانات (أدناه) الأمر أكثر احتمالاً، لأن المعاملات المتجاوزة يمكن أن تفصل إنشاء/حذف الوالد عن إعادة توجيه الطفل.

الكشف (ليس التنفيذ)

على مسار التطبيق إلى إيقاف الشبكة عند أي شذوذ، لذا بدلاً من ذلك نحن FKS سوف يؤدي فرض `SECURITY DEFINER`، **نكتشف وننبه**. تقوم كل عقدة بتثبيت دالة `omnihss_fk_orphan_count()`، ذات عمود واحد تشير قيمتها إلى والد FK التي تحسب كل `postgres_exporter` مفقود عبر جميع الجداول. يقوم عندما تكون Grafana بتحديثها، وتنبه `postgres_exporter` مفقود عبر جميع الجداول. يقوم **أكبر من صفر (انظر المقاييس)**.

قم بتشغيل عملية المسح يدوياً في أي وقت:

```
sudo -u postgres psql -d omnihss -tAc "SELECT
omnihss_fk_orphan_count();" # 0 = clean
```

غير EPC مثال: المشتركين الذين يشيرون إلى ملف) لتحديد الصفوف المخالفة لمرجع معين (موجود):

```
SELECT s.imsi, s.epc_profile_id
FROM subscriber s
LEFT JOIN epc_profile e ON e.id = s.epc_profile_id
WHERE s.epc_profile_id IS NOT NULL AND e.id IS NULL;
```

قم بإصلاح ذلك عن طريق إعادة توجيه الصف إلى والد صالح (تحديث على صف موجود يتكرر بشكل نظيف عبر الشبكة) أو إزالته.

منع انحراف المخطط

كان المحفز الجذري للحادث الذي دفع إلى هذا القسم هو **ترحيل المخطط الذي تم تطبيقه على عقدة واحدة بينما كانت نظيراتها لا تزال على المخطط القديم**، ثم كتابة بيانات جديدة ضد الحقول الجديدة. قامت المصدر بنشر صفوف لم تتمكن نظيرات المخطط القديم من تطبيقها، مما تسبب في أخطاء تطبيق ومعاملات متجاوزة، وبالتالي تباين.

- **OmniHSS قم بترحيل الشبكة بأكملها معاً.** قم بتطبيق الترحيلات على كل عقدة قبل التوفير ضد الحقول الجديدة. لا تكتب بيانات المخطط الجديد بينما أي نظير متأخر.

- **راقب الانحراف.** `omnihss_schema_max_version / omnihss_schema_migration_count` لكل عقدة. قارنها عبر الشبكة. العقدة `omnihss_schema_migration_count` المتأخرة عن نظيراتها هي ترحيل نصف مطبق.
- **راقب عدادات الأخطاء.** يعني ارتفاع `omnihss_subscription_error_apply_error_count / _sync_error_count` أن عامل التطبيق يفشل (وربما يتجاوز). يتم تنبيه "أخطاء" `_sync_error_count` على هذا "HSS" تطبيق/مزامنة.

إزالة نظير

تبقى `pg_hba.conf` تؤدي إزالة نظير إلى إسقاط الاشتراك من ذلك النظير وإزالة إدخاله في البيانات التي تم تكرارها بالفعل؛ فقط التغييرات المستقبلية تتوقف عن التدفق. يجب تأكيد الإزالة بشكل صريح.

على العقدة التي تقوم بإزالة النظير منها:

```
DROP SUBSCRIPTION IF EXISTS sub_from_<peer_name>;
```

PostgreSQL وأعد تحميل `pg_hba.conf` ثم قم بإزالة سطر (أسطر) النظير من

```
sudo -u postgres psql -c 'SELECT pg_reload_conf();'
```

لا يزال النظير الذي يتم إزالته يحتفظ باشتراكه إلى هذه العقدة. إذا كان يجب إزالة الرابط بالكامل، ومجموعة `connected_omnihss` قم بإزالة ذلك الاشتراك هناك أيضًا. ثم قم بإزالة العقدة من `hss` حتى لا يعيد تشغيل الدور التالي إنشائها.

النسخ الاحتياطي والاستعادة

القياسية PostgreSQL تفريغات OmniHSS تستخدم النسخ الاحتياطي والاستعادة في

العملية	الكتابة	الملاحظات
النسخ الاحتياطي	<code>services/backup.yml</code>	ينتج <code>hss_dump_<host>_<ts>.sql</code> و (قاعدة البيانات) <code>hss_<host>_<ts>.tar.gz</code> تحت (<code>/etc/omnihss</code> تكوين) <code>backups/<Site_Name>/</code> . يستثني التفريغ المنشورات/ الاشتراكات.
الاستعادة	<code>util_playbooks/restore_hss.yml</code>	و/أو أرشيف SQL يطلب تفريغ يأخذ، OmniHSS التكوين. يوقف نسخة احتياطية قبل الاستعادة، يسقط/يعيد إنشاء قاعدة البيانات، يستعيد، ويعيد التشغيل.

انظر [كتب التشغيل المساعدة](#) لعملية النسخ الاحتياطي/الاستعادة الأوسع.

المقاييس

بما في PostgreSQL مما يعرض مقاييس `postgres_exporter`، OmniHSS تشغل كل عقدة السلاسل المفيدة لصحة التكرار. Prometheus إلى (ذلك التكرار

الوصف: ما إذا كانت كل فتحة Gauge **النوع:** `pg_replication_slots_active` **المقياس:** تكرار منطقية لديها مستهلك نشط. **التسميات**

- فتحة التكرار (واحدة لكل نظير مشترك): `slot_name`

الوصف: تأخير موضع Gauge: تأخير الفتحة **النوع:** `pg_stat_replication_*` **المقياس:** لكل فتحة. يظهر مدى تأخر نظير مستهلك WAL

استعلامات المثال:

```
# أي فتحة تكرار منطقية غير نشطة (تنبيه إذا كانت < 0)
count(pg_replication_slots_active == 0)
```

```
# يمكن التحقق من الاشتراكات التي يجب أن توجد مقابل العمال النشطين
# ضد مخرجات كتاب التشغيل للتحقق أثناء النشر
```

المخصصة OmniHSS مقاييس

على كل عقدة أيضًا بتصدير مقاييس صحة التكرار والسلامة الخاصة postgres_exporter يقوم عبر ملف استعلام مخصص OmniHSS بـ Prometheus يقوم (roles/omnihss/files/postgres_exporter_queries.yaml). **HSS** تدعم قواعد التنبيه في مجموعة **تنبيهات** HSS Postgres. بجمع هذه تحت وظيفة (roles/monitoring/templates/grafana/rules/):

المقياس	النوع	ي
omnihss_subscription_worker_alive{subname}	Gauge	ن 1 امل ليبق عمل
omnihss_replication_slot_active{slot_name}	Gauge	ن 1 ظير هلك تحة
omnihss_replication_slot_wal_status_code{slot_name}	Gauge	0= ممتد غير جوز قود
omnihss_replication_slot_lag_bytes{slot_name}	Gauge	بتات WA تفظ تحة
omnihss_subscription_error_apply_error_count{subname}	Counter	طاء امل ليبق كمية
omnihss_subscription_error_sync_error_count{subname}	Counter	طاء امنة

المقياس	النوع	ملاحظات
		أول كمية
omnihss_integrity_fk_orphans	Gauge	وف ذات فتاح رجي بالق ، أن (0
omnihss_schema_max_version / omnihss_schema_migration_count	Gauge	لامة مياه عالية جيل بق / العد

انظر **المراقبة والرصد** لخط أنابيب المقاييس ولوحات المعلومات

استكشاف الأخطاء وإصلاحها

العقدة الجديدة تبقى فارغة بعد الانضمام

الأعراض: العمال التكراريون نشطون والفتحات نشطة، لكن العقدة الجديدة ليس لديها (أو لديها القليل من) المشتركين.

الأسباب المحتملة:

- انضمت العقدة إلى شبكة مكتظة بالفعل ولم يتم تهيئتها أبدًا. تستخدم الاشتراكات `copy_data = false`، لذا تصل فقط التغييرات بعد الانضمام.

الحل:

- قم بتهيئة العقدة من نظير صحي. انظر **تحميل عقدة جديدة بالبيانات**.

عامل الاشتراك غير نشط

الأعراض: `pg_stat_subscription` لا اشتراك في `pid IS NULL`.

الأسباب المحتملة:

- (جدار ناري / توجيه) PostgreSQL النظير غير متاح على منفذ.
- لهذه العقدة `pg_hba.conf` للنظير مفقود من IP عنوان
- (`database_username` / `database_password` بين العقد (تختلف بين العقد).

الحل:

1. إلى النظير على `TCP 5432` تأكد من إمكانية الوصول عبر.
2. وأعد (المحلية `hss` أو مجموعة) `connected_omnihss` تحقق من ظهور النظير في `pg_hba.conf`. تشغيل الدور حتى يتم تحديث
3. متطابقة عبر جميع `database_username/database_password` تأكد من أن العقد.

wal_status = lost فتحة التكرار تظهر

؛. النظير المقابل متأخر كثيرًا `lost` لفتحة ما هي `wal_status` الأعراض: حالة

الأسباب المحتملة:

- المطلوبة WAL كان النظير متوقفًا أو غير متاح لفترة طويلة بحيث تمت إزالة.

الحل:

1. قم بإعادة تهيئة النظير المتأثر من عقدة صحية. انظر **تحميل عقدة جديدة بالبيانات**.

أخطاء مفتاح مكرر أثناء النسخ الأولي

الأعراض: `copy_data = true` أخطاء في تطبيق التغييرات مباشرة بعد إنشاء اشتراك مع

الأسباب المحتملة:

- تم تشغيل النسخ الأولي من أكثر من نظير، مما أدى إلى نسخ نفس الصفوف مرتين.

الحل:

1. احتفظ بجميع `copy_data = true` قم بالتهيئة من نظير واحد فقط مع `copy_data = false` الاشتراكات الأخرى عند

تحديثات حالة متضاربة بين المواقع

يتأرجح " بين القيم المحددة في " (`subscriber_state` مثل) **الأعراض**: يبدو أن صف الحالة مواقع مختلفة.

الأسباب المحتملة:

- تحديثات متزامنة لنفس الصف في موقعين.

الحل: هذا متوقع ويتم التعامل معه. يحافظ المحفز الذي يفوز في آخر كتابة على التحديث مع عبر جميع المواقع حتى يكون ترتيب (NTP) تأكد من تزامن الساعات. `updated_at` أحدث `updated_at` إذا معنى.

Proxmox VM/LXC نشر

يقوم كتاب اللعب Proxmox على OmniCore تقوم الغالبية العظمى من عملائنا بتشغيل مجموعة بناءً على LXC وحاويات VMs QEMU بتوفير كل من `util_playbooks/proxmox.yml`، الواحد `proxmoxServers` (على إدخال) يمكنك تعيين التبديل لكل موقع `deployment_type` تبديل بينما تبقى المضيفات الفردية LXC وتجاوزها لكل مضيف. يتيح ذلك لموقع ما أن يعمل في الغالب كـ (UPF) مثل VMs كاملة.

للسنشر (Vultr / AWS / GCP) والسحابة HyperV وVMware نحن لا نزال ندعم.

انظر أيضًا:

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

VM مقابل LXC

LXC حاويات:

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

VMs (KVM):

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

حالات الاستخدام:

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

اختيار نوع النشر

لكل مضيف بهذا الترتيب `deployment_type` يتم حل

1. يفوز إذا كان موجودًا: (المعين تحت اسم المضيف) `deployment_type` **لكل مضيف**
2. الذي يختاره التوزيع `proxmoxServers` على إدخال `deployment_type` **لكل موقع** الدائري للمضيف.
3. إذا لم يتم تعيين أي منهما `vm` **افتراضي**.

مثل `proxmoxServers` النمط الشائع هو تعيين الافتراضي للموقع على إدخال `deployment_type: lxc` وتجاوز فقط المضيفات التي يجب أن تختلف. المثال القياسي هو `deployment_type: lxc` ويجب أن يكون TUN / الذي يحتاج إلى وحدات نواة، UPF، باستثناء LXC مختبر يعمل كل شيء ك VM:

```
all:
  vars:
    proxmoxServers:
      pve-node-01:
        deployment_type: lxc      # LXC الافتراضي للموقع → كل شيء هو
        # ...

upf:
  hosts:
    site-upf01:
      ansible_host: 192.168.1.24
      deployment_type: vm        # VM تجاوز لكل مضيف → هذا المضيف هو
```

VM كقاعدة، حافظ على الافتراضي للموقع متسقًا وتجاوز فقط الاستثناءات. عندما تقوم بتوزيع عبر العديد من المضيفات، يعتمد وضع العقد على فهرس المضيف ومن الصعب التفكير فيه LXC و

يؤدي كتاب اللعب فقط إلى الفشل الصارم في حالة حدوث تصادمات نوع مع الموارد الموجودة: إذا ولكن LXC مثل وجود) كانت هناك مورد مع اسم المضيف المستهدف موجود بالفعل كنوع خاطئ

فإنه يتوقف مع خطأ وا❖❖ح. انظر، (أو العكس، deployment_type: vm، المضيف يحل إلى حراس النوع.

Proxmox إعداد

1. API إنشاء رمز

```
# API مركز البيانات → الأذونات → رموز Proxmox: واجهة  
# إنشاء رمز: root@pam!<TokenName>  
# نسخ سر الرمز (يظهر مرة واحدة)  
# قم بإلغاء تحديد مربع `فصل الامتيازات`
```

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

يقوم بتنزيل صورة سحابية لأوبونتو ويخلق Proxmox. قم بتشغيل هذا البرنامج النصي على مضيف في وقت النسخ SSH ومفاتيح cloud-init بحقن بيانات اعتماد proxmox.yml قالبًا نظيفًا. يقوم (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=vbr0
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:

- بتعيين `proxmox.yml` قالب لا يحتوي على مستخدمين أو كلمات مرور مشفرة. يقوم `ciuser` و `cipassword` و `sshkeys` في وقت النسخ.
- على إدخال `proxmoxTemplateUser` cloud-init: أولوية مستخدم `local_users` أول مفتاح → (المطابق `proxmoxServers`).
- على إدخال `proxmoxTemplatePassword` cloud-init: أولوية كلمة مرور (أي القيمة المختارة أعلاه) cloud-init مستخدم → (المطابق `proxmoxServers`).
- المصادقة `local_users.*.public_key` من كل إدخال SSH يتم حقن مفاتيح (القائمة على المفاتيح هي مسار تسجيل الدخول المقصود).
- ويمكن اختيارياً `proxmoxTemplateUser` قم بتعيين `proxmoxTemplatePassword` على كل إدخال `proxmoxServers` كلما تم إعداد `local_users` ليس هو أول إدخال cloud-init قالب مع مستخدم الحساب الصحيح.
- تعمل Proxmox يضمن أن وحدة التحكم في الويب `--vga std`.

- تقوم بتنزيل فقط إذا كانت أحدث من النسخة المحلية wget على -N علامة

ISO بديل: قالب يدوي من

إذا لم تكن الصور السحابية متاحة أو كنت بحاجة إلى تثبيت مخصص

عبر واجهة الويب VM الخطوة 1: إنشاء

- ubuntu-2404-template الاسم، VM 9000 جديدة → معرف VM إنشاء
- موجود ISO خادم أوبونتو أو استخدام ISO نظام التشغيل: تحميل
- (SCSI: VirtIO SCSI وحدة التحكم) النظام: الافتراضي
- SCSI: الناقل، IGB الأفراس: 32
- وحدة المعالجة المركزية: 2 نوى
- الذاكرة: 2048 ميجابايت
- vmbr0 جسر، VirtIO: الشبكة
- وتثبيت خادم أوبونتو VM بدء

تنظيف وتحضير - VM الخطوة 2: داخل

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

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

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

وتحويل إلى قالب Cloud-Init الخطوة 3: إضافة

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

- `proxmox.yml` على القالب. يقوم cloud-init لا تقم بتعيين مستخدم / كلمة مرور `proxmoxTemplateUser` / `proxmoxTemplatePassword` (لاسم المستخدم `local_users` مع تراجع).

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

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

أدناه `proxmoxLxc0sTemplate` هو ما تقوم بتكوينه كـ (`local:vztmpl/ubuntu-24.04-standard_24.04-2_amd64.tar.zst` مثل) مسار الحجم الناتج

تكوين ملف المضيفين

نشر VM

```
all:
  vars:
    # الافتراضي هو → deployment_type تم حذف
    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 # cloud- اسم مستخدم
        local_users # افتراضي هو أول مفتاح init
        proxmoxTemplatePassword: omnitouch # cloud-init كلمة مرور
        cloud-init # افتراضي هو اسم مستخدم
        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
      mask_cidr: 24 # افتراضي 24 اختياري
      vlanid: 100 # اختياري
```

LXC نشر

```
all:
  vars:
    proxmox_lxc_nameserver: "1.1.1.1" # 1.1.1.1 الافتراضي، اختياري
    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 # rootfs اختياري، تراجع
      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
      mask_cidr: 24 # 24 الافتراضي، اختياري
      vlanid: 100 # اختياري
      proxmoxLxcCores: 2 # تجاوز اختياري
      proxmoxLxcMemoryMb: 4096 # تجاوز اختياري
      proxmoxLxcDiskSizeGb: 30 # تجاوز اختياري
      proxmoxLxcRootFsStorageName: local-lvm # يتجاوز التخزين
      proxmoxLxcDefaultStorage: proxmox الافتراضي للمضيف
```

الاستخدام

(VM أو LXC) توفير

```
ansible-playbook -i hosts/Custommer/site.yml
util_playbooks/proxmox.yml
```

حذف

```
ansible-playbook -i hosts/Custommer/site.yml
util_playbooks/proxmox_delete.yml
```

.`deployment_type` تلقائيًا، بغض النظر عن LXCs و VMs يتعامل كتاب اللعب للحذف مع كل من

سلوك كتاب اللعب

حراس النوع

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

مورد جديد

- حسب فهرس المضيف داخل `proxmoxServers` يقوم بالتوزيع الدائري عبر إدخالات لتثبيت إدخال معين (انظر `proxmox_server:` المجموعة، ما لم يتم المضيف بتعيين **التوزيع عبر العقد**).
- التالي المتاح VMID للحصول على Proxmox يستعلم عن واجهة برمجة تطبيقات.
- `cloud-init` ويقوم بتعيين مستخدم `proxmoxTemplateId` يتم استنساخه من **VM:** من `proxmoxTemplateUser` `cloud-init =` مستخدم SSH كلمة المرور / مفاتيح `cloud-init =` ! كلمة مرور `local_users` إدخال الخادم إذا تم تعيينه، وإلا أول مفتاح

يقوم cloud-init إذا تم تعيينها، وإلا اسم مستخدم proxmoxTemplatePassword وبيداً scsi0 بتغيير حجم.

- **LXC:** على proxmoxLxc0sTemplate ، rootfs يتم إنشاؤه من proxmoxLxcRootFsStorageName (المجموعة) → proxmoxLxcDefaultStorage (الخادم) → storage → local-lvm (التراجع) عبر ssh-public-keys عبر local_users.*.public_key يقوم بحقن جميع ،(النهائي غير مميزة LXCs جميع @mn1Touch@. كلمة مرور الجذر مشفرة ك 0 ، keys (unprivileged: 1) مع تمكين التعشيش (features: nesting=1).

مورد موجود

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

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

ك (data / gtp أو n3) المسمى secondary_ips تعالج الأدوات إدخال **OmniUPF:** به ويخصص N3/N9 + XDP بربط omniupf مخصص لطائرة المستخدم. يقوم دور NIC وحدة معالجة مركزية للواجهة الرئيسية، بحيث لا يمكن أن يؤدي الحمل الثقيل على مسار (الرئيسي IP التي تبقى على) PFCP/API/SSH البيانات إلى تجويع.

وضع العلامات

.site_name + Ansible يتم وضع علامة ع❖❖ على كل مورد بأسماء مجموعته في

التوزيع عبر العقد

بالتوزيع proxmoxServers بشكل افتراضي، تقوم الأدوات بتوزيع المضيفات عبر إدخالات الفهرس هو موضع المضيف **داخل مجموعة دوره**، LXCs و VMs الدائري، مع نفس المنطق لـ (بترتيب الإعلان) proxmoxServers مأخوذاً من باقي عدد إدخالات

```
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)
```

إضافة عقدة ثانية تعيد توزيع كل شيء. نظرًا لأن التوزيع يعتمد على الموضع، فإن واحد إلى اثنين يغير مكان وجود مجموعات متعددة `proxmoxServers` الانتقال من إدخال المضيفين الموجودة (المضيف-0 → الإدخال-0، المضيف-1 → الإدخال-1، ...). قم بتثبيت المضيفات التي لا تريد نقلها (أدناه) قبل إعادة التشغيل ضد أسطول مختلط

تثبيت مضيف على عقدة معينة

لفرض `proxmoxServers` إلى اسم إدخال `proxmox_server:` قم بتعيين المفتاح الاختياري مضيف على تلك العقدة بغض النظر عن فهرس مجموعته. عند عدم تعيينه، ينطبق التوزيع الدائري تفشل التشغيل مع خطأ يسرد الأسماء الصالحة (لذا `proxmoxServers` أعلاه. قيمة ليست مفتاح. لا يمكن أن يؤدي خطأ مطبعي إلى التوزيع الدائري في مكان آخر).

```
device_tester:
  hosts:
    site-device-tester01:
      ansible_host: 10.179.2.50
      proxmox_server: nicklab-prxm01 # تثبيت على هذا الإدخال من
proxmoxServers (→ عقده)
# إذا كان تخزين القرص لتلك العقدة يختلف عن اسم التخزين العالمي
proxmoxLxcRootFsStorageName,
# قم بتجاوزه هنا أيضًا (يتم قراءة التخزين عالميًا، وليس من إدخال الخادم)
proxmoxLxcRootFsStorageName: local-lvm
```

للمجموعة لتثبيت كل مضيف في `vars:` على مستوى `proxmox_server` يمكن أيضًا تعيين مجموعة على عقدة واحدة.

تجاوزات لكل مضيف

(تثبيت العقدة: انظر **التوزيع عبر العقد**) `proxmox_server` و `deployment_type` يمكن تعيين `proxmoxLxcCores`، `proxmoxLxcMemoryMb`، `proxmoxLxcDiskSizeGb`، `proxmoxLxcRootFsStorageName`، `proxmoxLxcOsTemplate`، `host_vm_network` تحت اسم المضيف لكل مضيف.

```
dns:
  hosts:
    site-dns01:
      ansible_host: 192.168.1.20
      deployment_type: vm          # تجاوز لكل مضيف الافتراضي للموقع
      proxmoxLxcDiskSizeGb: 120    # تجاوز لكل مضيف
      host_vm_network: vmbr1       # تجاوز جسر لكل مضيف
  vars:
    proxmox_interface: vmbr0
    gateway: 192.168.1.1
    mask_cidr: 24 # اختياري، الافتراضي 24
```

كتيبات الخدمة

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

تسلسل كتيبات اللعب

:تم هيكلة كتيبات اللعب بشكل هرمي لتمكين نشر سريع وموجه

```

all.yml
├─ setup_users.yml
├─ apt_cache.yml          # فقط عندما يتم تعريف مجموعة
apt_cache_servers
├─ dns.yml
├─ common.yml
├─ license_server.yml
├─ monitoring.yml
│   └─ grafana.yml
├─ device_tester.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
│   └─ omnisepp.yml
├─ omniepdg.yml
├─ omnitwag.yml
├─ omniss7.yml
├─ ocs.yml
├─ crm.yml
├─ ran_monitor.yml
├─ 5gc.yml
└─ ../util_playbooks/health_check.yml

```

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

مرجع سريع

كتيبات اللعب على المستوى الأعلى

الوصف	النطاق	كتيب اللعب
المراقبة، مختبر، DNS، نشر كامل: المستخدمون الأجهزة، EPC، IMS، ePDG، TWAG، SS7، OCS، CRM، RAN، 5G Core، مراقبة صحي نهائي (من <code>util_playbooks/</code>)	الشبكة الكاملة	<code>all.yml</code>
MME، SGW-C، PGW-C، UPF، HSS، DRA	Packet Core	<code>epc.yml</code>
P/I/S-CSCF، الرسائل، OmniSEP، خادم التطبيقات، (XCAP/الامتيازات/BSF)	الصوت/IMS	<code>ims.yml</code>
OCS مزامنة، KeyDB مجموعة، CGrateS	الشحن	<code>ocs.yml</code>
Prometheus، Grafana، المصدّرين، HOMER	المراقبة	<code>monitoring.yml</code>

كُتِيبَات بنية تحتية

الوصف	كُتِيب اللعب
وكلاء، NTP، تكوين نظام التشغيل الأساسي، الحزم التسجيل	<code>common.yml</code>
SSH حسابات المستخدمين المحليين ومفاتيح	<code>setup_users.yml</code>
DNS نشر خادم	<code>dns.yml</code>
OmniCore خادم ترخيص	<code>license_server.yml</code>
مثبتًا على كل مضيف netplan تحقق مما إذا كان	<code>does_netplan_exist.yml</code>
netplan الواجهة عبر MTU تعيين وتطبيق	<code>update_mtu.yml</code>
iptables/nftables قواعد	<code>firewall.yml</code>
محلية (إذا كانت مفعلة) APT مرآة	<code>apt_cache.yml</code>

EPC مكونات

الوصف	المكون	كتيب اللعب
(4G) كيان إدارة التنقل	OmniMME	omnimme.yml
بوابة الخدمة لطبقة التحكم	OmniSGW-C	omnisgwc.yml
لطبقة التحكم PDN بوابة	OmniPGW-C	omnipgwc.yml
(SGW-U/PGW-U) وظيفة طبقة المستخدم (مجموعة)	OmniUPF	upf.yml / omniupf.yml
خادم المشتركين المنزليين	OmniHSS	omnihss.yml / hss.yml
Diameter وكيل توجيه	OmniDRA	omnidra.yml
بوابة الوصول اللاسلكي الموثوق	OmniTWAG	omnitwag.yml
(WiFi Calling) بوابة بيانات الحزمة المتطورة	OmniEPDG	omniepdg.yml
GTP وكيل حركة مرور	GTP Proxy	gtp_proxy.yml

IMS مكونات

الوصف	المكون	كتيب اللعب
وظيفة التحكم في جلسة المكالمات الوكيل	P-CSCF	pcscf.yml
المستجوب CSCF	I-CSCF	icscf.yml
الخادم CSCF	S-CSCF	scscf.yml
خادم تطبيقات الهاتف	OmniTAS	as.yml
البريد ، BSF ، خادم الامتيازات ، XCAP الصوتي المرئي	OmniSEP	omnisep.yml
ينشر (omnisep.yml الاسم القديم لـ omnisep) نفس دور	OmniSEP	xcap.yml
IMS و SMS-over-IP ، SMPP ، وحدة تحكم مع cscf (يتم نشرها عبر دور) cscf_type: smsc_ims	OmniMessage / IMS SMSC	omnimessage.yml

5G Core

المشترك مع gc_nf يتم تثبيت كل وظيفة شبكة بواسطة دور 5 . gc.yml يتم نشره عبر 5 NF ، للحصول على تفاصيل الدور ، تكوين كل 5G Core راجع 5 NF لكل nf_package تثبيت UDM HNET/SUCI الإصدار ، ومفاتيح

الوصف	المكون	كتيب اللعب
Gينشر جميع وظائف الشبكة المستقلة 5	5G Core	5gc.yml
وظيفة مستودع الشبكة	OmniNRF	
وظيفة إدارة الوصول والتنقل	OmniAMF	
وظيفة خادم المصادقة	OmniAUSF	
وظيفة دعم الربط	OmniBSF	
وظيفة الشحن	OmniCHF	
وظيفة اختيار شريحة الشبكة	OmniNSSF	
وظيفة التحكم في السياسات	OmniPCF	
وكيل الاتصال بالخدمة	OmniSCP	
وظيفة إدارة الجلسات	OmniSMF	
إدارة البيانات الموحدة	OmniUDM	
مستودع البيانات الموحد	OmniUDR	

الدائرة المبدلة

Circuit-Switched Core راجع (`circuit_switched/` الأدوار تحت) `cs.yml` يتم نشره عبر SS7 و MSC و BSC للحصول على تفاصيل أدوار.

الوصف	المكون	كتيب اللعب
القديم CS ينشر مجال	الدائرة المبدلة	cs.yml
وحدة التحكم في المحطة الأساسية (circuit_switched/bsc)	BSC	
مركز التحويل المحمول (circuit_switched/omnimsc)	OmniMSC	
SS7/SIGTRAN (omniss7 role) مجموعة	SS7	

الخدمات الداعمة

الوصف	كتيب اللعب
نظام الشحن عبر الإنترنت (CGrateS + KeyDB)	<code>ocs.yml</code>
؛ التكوين الآن في OmniCRM (بوابة إدارة العملاء <code>group_vars/</code>)	<code>crm.yml</code>
بوابة SS7/SIGTRAN	<code>omniss7.yml</code>
SIP/Diameter التقاط حزم	<code>homer.yml</code>
والتنبيه Grafana لوحات معلومات	<code>grafana.yml</code>
RAN تكامل مراقبة	<code>ran_monitor.yml</code>
واجهة برمجة ، TAP3/RAP واجهة) OmniRoam مجموعة تجوال (مولد ، Elixir تطبيقات	<code>omniroam.yml</code>
وبث الخلايا OmniLCS خدمات موقع	<code>omnilcs.yml</code>
(أجهزة الاختبار) OmniWeb منصة اختبار وكيل هاتف	<code>device_tester.yml</code>

موجود تحت (`proxmox.yml` , `proxmox_delete.yml`) **Proxmox** توفير
`util_playbooks/`. راجع [Proxmox VM/LXC](#) و [نشر Utility Playbooks](#).

العمليات

الوصف	كتيب اللعب
نسخ احتياطي لقواعد البيانات والتكوينات	<code>backup.yml</code>
إعادة تشغيل مضبوطة للمضيفين	<code>reboot.yml</code>
إيقاف تشغيل سلس	<code>shutdown.yml</code>
تحديث الحزم	<code>apt_update.yml</code>
APT تحديث بيانات تعريف الخاصة بذاكرة	<code>apt_refresh_metadata.yml</code>
اختبار عرض النطاق الترددي للشبكة	<code>speedtest.yml</code>

كتيبات الاستعادة

الوصف	كتيب اللعب
من النسخة الاحتياطية OmniTAS استعادة	<code>restore_applicationserver.yml</code>
من النسخة الاحتياطية؛ وحدة SMSC استعادة IMS وعقد SMPP، بوابة OmniMessage، تحكم SMSc	<code>restore_smsc.yml</code>

ليست في `restore_ocs.yml`، و `restore_hss.yml`، `health_check.yml`: ملاحظة
يقوم **Utility Playbooks** راجع `util_playbooks/` إنها موجودة في `services/`.
`../util_playbooks/health_check.yml` بتشغيل الفحص الصحي ك `all.yml`.

الاستخدام

نشر كل شيء

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

نشر نظام فرعي محدد

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

نشر مكون واحد

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

الحد من مضيفين محددين

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

# التشغيل على مضيفين محددين متعددين
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

- **معمارية النشر** - سير العمل العام للنشر
- **تكوين ملف المضيف** - تعريف البنية التحتية
- **تكوين المتغيرات الجماعية** - التخصيص
- **كتيبات الأدوات** - أدوات التشغيل (فحص صحي، استعادة، إلخ.)
- **التنبيه، Grafana، Prometheus، المراقبة والرؤية**

كتيبات الأدوات

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

مرجع سريع

الغرض	كُتِيب الأدوات
على LXC توفير الآلات الافتراضية و/أو حاويات Proxmox (حسب <code>deployment_type</code>)	<code>proxmox.yml</code>
Proxmox على LXC حذف الآلات الافتراضية/حاويات	<code>proxmox_delete.yml</code>
عبر LXC فرض إعادة تشغيل الآلات الافتراضية/حاويات عندما لا تستجيب Proxmox واجهة برمجة تطبيقات SSH	<code>proxmox_restart_hung.yml</code>
VMware vSphere توفير الآلات الافتراضية على	<code>vmware.yml</code>
VMware vSphere حذف الآلات الافتراضية على	<code>vmware_delete.yml</code>
إنشاء تقرير شامل عن صحة جميع الخدمات	<code>health_check.yml</code>
و/أو التكوين من النسخة HSS استعادة قاعدة بيانات الاحتياطية	<code>restore_hss.yml</code>
فارغة من نظير مأهول (تحميل OmniHSS تهيئة عقدة بيانات النسخ الأولية)	<code>seed_hss_replication.yml</code>
لاكتشاف انحراف البيانات (عدد OmniHSS تدقيق شبكة الصفوف + تجزئة المحتوى لكل جدول مرجعي عبر جميع العقد)؛ قراءة فقط	<code>audit_hss_drift.yml</code>
عن طريق ملء OmniHSS تسوية انحراف بيانات شبكة الصفوف المفقودة عبر العقد (تشغيل جاف افتراضي)	<code>reconcile_hss_drift.yml</code>
والتكوين من النسخة MySQL و OCS KeyDB استعادة الاحتياطية	<code>restore_ocs.yml</code>
Mermaid إنشاء وثائق الشبكة مع مخططات	<code>ip_plan_generator.yml</code>

العرض	كُتِبَ الأدوات
تدقيق المنافذ المفتوحة والخدمات المستمعة عبر جميع المضيفين	<code>get_ports.yml</code>
استرجاع ملفات التقاط الحزم من المضيفين	<code>getLocalCapture.yml</code>
إزالة حساب مستخدم محلي من جميع المضيفين	<code>delete_local_user.yml</code>
Mnesia لـ OmniSS7 / OmniMessage (تدميري) إعادة تشغيلها مسح قاعدة بيانات	<code>clear_mnesia.yml</code>
حذف ملف الترخيص على خادم الترخيص وإعادة تشغيله	<code>delete_license.yml</code>

توفير الآلات الافتراضية

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

للحصول على تكوين مفصل [Proxmox](#) راجع [نشر](#).

Proxmox

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

؛ يتم تحديد ما إذا كان LXC بتوفير كل من الآلات الافتراضية وحاويات `proxmox.yml` يقوم تجاوز) المحلولة `deployment_type` بواسطة LXC المضيف المعطى يصبح آلة افتراضية أو حاوية [Proxmox](#) راجع [نشر](#). (vm الافتراضي → `proxmoxServers` لكل مضيف → إدخال

```
# توفير (deployment_type حسب LXC، آلات افتراضية و/أو حاويات)
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
```

```
# حذف الآلات الافتراضية/حاويات LXC
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/proxmox_delete.yml --limit old-host
```

Proxmox إعادة تشغيل المضيفين المعلقين على

الملف: `util_playbooks/proxmox_restart_hung.yml`

غير LXC لكل مضيف ويفرض إعادة تشغيل أي آلات افتراضية أو حاويات SSH يتحقق من استجابة بشكل قسري QEMU يتم إيقاف تشغيل آلات Proxmox. مستجيبة عبر واجهة برمجة تطبيقات يتم. (أو إعادة تشغيلها إذا كانت قيد التشغيل بالفعل) LXC وإعادة تشغيلها؛ يتم بدء تشغيل حاويات SSH. تخطي المضيفين الذين يستجيبون لـ

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

```
# تقييد إلى مضيف أو مجموعة واحدة
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/proxmox_restart_hung.yml --limit hss01
```

VMware vSphere

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

`util_playbooks/vm/vmware/` تتضمن كلاً من `create_individual_vm.yml` و `delete_individual_vm.yml`. `util_playbooks/vm/vmware/`

SSH ينتظر أيضًا `vmware.yml` هو غلاف رقيق حول مهمة الحذف؛ بينما `vmware_delete.yml` على الآلة الافتراضية الجديدة netplan القديمة ويطبق netplan بعد التوفير، ثم يزيل ملفات

`uv sync` Python **المتطلبات المسبقة:** يتطلب تثبيت تبعيات

الاستخدام:

```
# توفير الآلات الافتراضية
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 الآلات الافتراضية
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/vmware_delete.yml --limit old-host
```

المتغيرات المطلوبة:

```
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"
```

Vultr

الملفات: `util_playbooks/vm/vultr/vmProvisionVultr.yml`,
`util_playbooks/vm/vultr/vmTeardownVultr.yml`,
`util_playbooks/vm/vultr/vmCommonSetup.yml`

يقوم Vultr عبر واجهة برمجة تطبيقات Vultr يوفر وينزع الحالات على `vmProvisionVultr.yml` بإزالتها، `vmTeardownVultr.yml` بإنشاء الحالات، و `vmCommonSetup.yml` الثلاثة هي. (إعادة التشغيل، إلخ، DNS) بأداء إعداد قاعدة بعد التوفير `vmCommonSetup.yml` من متغير البيئة API كتيبات نقطة دخول على مستوى عالٍ بدلاً من تضمينات المهام. تقرأ رمز

من نفس الدليل، لذا قم `cloud-config.yaml` وبشير كتيب التوفير إلى `VULTR_API_TOKEN`، بتشغيلها من `util_playbooks/vm/vultr/`.

```
export VULTR_API_TOKEN=<token>
ansible-playbook -i hosts/customer/host_files/production.yaml
util_playbooks/vm/vultr/vmProvisionVultr.yaml
```

فحص الصحة

الملف: `util_playbooks/health_check.yaml`

المنفذة OmniCall وOmniCore يغطي جميع خدمات HTML ينشئ تقرير صحة شامل بتنسيق

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

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

المعلومات المجمعة

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

HSS استعادة

الملف: `util_playbooks/restore_hss.yml`

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

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

تنسيقات ملفات النسخ الاحتياطي

النوع	نمط اسم الملف	المحتويات
قاعدة البيانات	<code>hss_dump_<hostname>_<timestamp>.sql</code>	تفريغ قاعدة بيانات OmniHSS (الافتراضي) <code>omnihss</code>)
التكوين	<code>hss_<hostname>_<timestamp>.tar.gz</code>	أرشف من دليل <code>/etc/omnihss</code>

قبل تشغيل (قاعدة البيانات و/أو التكوين) HSS يتم عمل نسخة احتياطية من الحالة الحالية لـ الاستعادة.

المطالبات

المطالبة	مطلوب	الوصف
HSS مسار تفريغ SQL	لا	اتركه فارغًا (HSS SQL المسار الكامل إلى ملف تفريغ (لتخطي استعادة قاعدة البيانات)
مسار ملف تكوين HSS tar.gz	لا	اتركه (HSS tar.gz المسار الكامل إلى ملف تكوين (فارغًا لتخطي استعادة التكوين)
تأكيد	نعم	لتأكيد عملية الاستعادة على المضيفين ☐ yes اكتب المذكورين

HSS تهيئة تكرار

الملف: `util_playbooks/seed_hss_replication.yml`

تنضم إلى شبكة تكرار نشطة نشطة OmniHSS يقوم بتحميل البيانات الأولية لمرة واحدة لعقدة يعيد (`copy_data = false`) يتم إنشاء الاشتراكات مع) مأهولة بالفعل. تظهر عقدة جديدة فارغة بحيث `copy_data = true` هذا الكتيب إنشاء الاشتراك إلى نظير مصدر   واحد مختار مع بنسخ البيانات الموجودة ثم تواصل البث دون أي فجوة PostgreSQL تقوم

:يعمل في أحد وضعين

فارغة. يكتشف الكتيب أي هدف محدود `seed_source_*` **تفاعلي (افتراضي):** اترك متغيرات فارغ. ثم يستفسر عن الأقران الصالحين للعنود على الأقران التي تحتوي على بيانات، ويقوم بإدراجها مع عدد المشتركين، ويطلب منك اختيار مصدر. يقوم بتغذية الأهداف الفارغة فقط

```
ansible-playbook -i hosts/customer/host_files/production.yml \
  util_playbooks/seed_hss_replication.yml \
  --limit site-hss01,site-hss02
```

:الأقران التي تحتوي على بيانات متاحة للتغذية منها
مشترك 838 - (10.4.10.142) dev-hss01 (1)
أدخل رقم المصدر للتغذية منه

صريح: مرر المصدر مباشرة لتخطي القائمة. هذا **يجب** إعادة التغذية حتى لو كانت الهدف تحتوي بالفعل على بيانات (استخدم لاستعادة).

```
ansible-playbook -i hosts/customer/host_files/production.yml \
  util_playbooks/seed_hss_replication.yml \
  --limit site-hss01 \
  -e "seed_source_name=dev-hss01 seed_source_host=10.4.10.142"
```

المعلومات

المعلمة	مطلوب	الافتراضي	الوصف
seed_source_name	لا	(مطلوب)	اسم مضيف الجرد للنظير المصدر المأهول. يستخدم لتسمية الاشتراك (<sub_from_name>) وحل نشره تركه (<name_pub>). في وضع التفاعل.
seed_source_host	لا	(مطلوب)	للتظير IP عنوان المصدر، قابل للوصول على منفذ الخاص PostgreSQL به. اتركه في وضع التفاعل.
seed_source_port	لا	منفذ omnihss.database_port المستهدف (5432)	PostgreSQL منفذ على النظير المصدر.

معًا لاستخدام الوضع seed_source_host و seed_source_name يجب تقديم كل من الصريح؛ عدم تقديم أي منهما يؤدي إلى تفعيل وضع التفاعل.

السلوك

- يعد عدد المشتركين على كل هدف محدود لتحديد أي منها فارغ.
- + المحلية `hss` (مجموعة) **وضع التفاعل**: يستفسر عن الأقران الصالحين للعثور على الأقران التي تحمل بيانات، ويقول **يادراجها**، `connected_omnihss` **وضع الصريح**: يطلب تأكيد `yes`.
- الخاص بالهدف `pg_hba.conf` المصدر مسموح به في IP يضمن أن عنوان.
- `WITH (copy_data = false)` يقوم بإسقاط وإعادة إنشاء الاشتراك إلى المصدر يقوم ببث **جميع** (`<target>_from_<source>` لكل رابط هو `slot_name` اسم) الجداول ولكن لا يقوم بأداء مزامنة جداول جماعية.
- يقوم بملء الصفوف الموجودة في **جدول الماستر** من المصدر باستخدام تحميل آمن `pg_dump --data-only --inserts --on-conflict-do-nothing --disable-triggers` من حيث التعارض كل مشترك: `subscriber_state` يتم تضمين `FK` يشير إلى واحد بواسطة `pdn_session`, `lte_call` ملء جداول الجلسات المؤقتة.
- يبلغ عن عدد المشتركين على العقدة التي تم تغذيتها.

جماعي لها على `COPY` لماذا يتم تخطي جداول الجلسات: إنها جداول نشطة؛ قد يتصادم نظام حي (مفتاح مكرر) مع كتابات متزامنة ويؤدي إلى تعطل جداول المزامنة. يتم إعادة ملئها `subscriber_state`. من النسخ المتماثل المتدفق وحركة المرور الحية بدلاً من ذلك مختلف: يتم إدراج الصفوف فقط في وقت التوفير (1:1 مع المشترك) ويتم تحديثها فقط آمن. بدونها، تشير مشتركات الهدف إلى `--on-conflict-do-nothing` لاحقًا، لذا فإن صفوف الحالة التي لا تصل أبدًا، حيث تقوم النسخ المنطقية بإسقاط تحديثات الصفوف المفقودة بصمت.

إلى العقدة (أو العقد) المستهدفة. في وضع التفاعل، يتم تخطي `--limit` مهم: دائمًا **وعمليات العقد** للحصول على سير العمل **OmniHSS** الأهداف غير الفارغة تلقائيًا. راجع **تكرار** الكامل.

HSS تدقيق انحراف

و (مستقل) `util_playbooks/audit_hss_drift.yml` **الملفات**: `roles/omnihss/tasks/audit_drift.yml` (التنفيذ المشترك).

النشطة النشطة. يتم إنشاء اشتراكات النسخ **OmniHSS** يبلغ عن **انحراف البيانات** عبر شبكة لذا فإن البيانات التي انحرفت قب `copy_data = false` المنطقية مع (اختلافات وقت التغذية) أو عبر كتابة مباشرة/غير مكررة لا تتم تسويتها أبدًا ولا يتم تنبيهها. تظهر

لكل موقع رؤيتها لأن الشبكة تمتد Prometheus النسخ 0 تأخير بينما تختلف البيانات، ولا يمكن لـ عبر موقعين.

(connected_omnihss المحلية + أقران hss مجموعة) يتصل التدقيق بكل عضو في الشبكة غير المعتمد على الترتيب md5 وبالنسبة لكل جدول مرجعي/تكوين، يقارن عدد الصفوف الدقيق و لالتقاط انحراف (عدد + أقصى إصدار) schema_migrations للمحتويات الكاملة. كما يقارن (subscriber_state, pdn_session, lte_call) المخطط. يتم استبعاد جداول الحالة الساخنة فهي تختلف بشكل مشروع تحت حركة المرور الحية. يتم تضمين فحص صحي هيكلي: (lte_call) عدد المشتركين لكل عقدة، __orphaned_subscribers__ واحد فوق المقارنة عبر العقد يؤدي عدد غير صفري على أي عقدة إلى فشل subscriber_state الذين تفتقر صفوفهم إلى بدون subscriber التدقيق حتى عندما تتفق كل عقدة. قد يؤدي تغذية نسخ إلى يتيم كل عقدة بشكل متطابق، وهو ما لا يمكن أن تراه المقارنة النقية subscriber_state للأقران، لذا Postgres عبر العقد، يتم مركز المقارنة على عداء واحد يصل إلى كل قاعدة بيانات --limit. فإن تشغيل واحد يغطي الشبكة الكاملة عبر الموقعين من أي جرد، وهو آمن لـ

(عندما يتم تمكين النسخ) omnihss في نهاية دور HSS. يتم تشغيله تلقائيًا بعد كل نشر لـ يتم تشغيل نفس التدقيق كتحقق من التناسق بعد النشر، وبشكل افتراضي يفشل التشغيل إذا تم العثور على انحراف، لذا فإن الشبكة المعطلة من المستحيل تفويتها أثناء النشر. إنه غير تفاعلي (لا المجموع، services/all.yml لنشر free توقف): يجب أن يكون التدقيق آمنًا تحت استراتيجية حيث ستؤدي التوقف إلى حدوث خطأ. في ذلك النشر المجموع، يتم كتم التدقيق داخل الدور ويتم تشغيل التدقيق بدلاً من ذلك مرة واحدة في كتيب خطّي مخصص في نهاية: يتم التحكم في السلوك بواسطة متغيري دور services/deploy_nfs.yml.

المتغير	الافتراضي	التأثير
omnihss_audit_drift	true	تشغيل التدقيق في نهاية الدور
omnihss_audit_fail_on_drift	true	فشل التشغيل في حالة الانحراف؛ للتقرير فقط false قم بتعيين (cron/CI غير تفاعلي)

التدقيق هو قراءة فقط في كلتا الحالتين. الافتراضي لكتيب المستقل هو التقرير ثم الفشل في cron/CI: حتى يمكنه التحكم في) حالة الانحراف

```
# عند الطلب
ansible-playbook -i hosts/<inv>/host_files/<site>.yaml
util_playbooks/audit_hss_drift.yaml
# تقرير فقط، دائمًا ينجح
ansible-playbook -i ... util_playbooks/audit_hss_drift.yaml -e
audit_fail_on_drift=false
# تعطيل التدقيق التلقائي لنشر واحد
ansible-playbook -i ... services/omnihss.yaml -e
omnihss_audit_drift=false
```

HSS تسوية انحراف

الملف: util_playbooks/reconcile_hss_drift.yaml

يعيد جداول المرجع المتباينة إلى التزامن من خلال جعل كل عقدة تحمل **اتحاد** الصفوف: لكل جدول متباين يقوم بملء الصفوف التي تفتقر إليها العقدة من أقرانها، باستخدام نفس الأسلوب (pg_dump --data-only --inserts --on-conflict-do-nothing --disable-triggers).

- **غير تدميري:** يتم إدراج الصفوف فقط (تخطي التعارضات). لا يتم حذف أي شيء أو تحديثه. يحل "تفتقر العقدة إلى صفوف!"; لا يحل "نفس المفتاح الأساسي، قيم مختلفة" أو يزيل الصفوف التي يجب ألا توجد. تحتاج تلك إلى قرار بشري.
- قبل (للمتابعة yes) **تفاعلي بشكل افتراضي:** يقوم بالمسح، ويطيع الخطأ، ويسأل -e dry_run=true الكتابة؛ أي شيء آخر يتسبب في الإلغاء دون تغييرات. استخدم للتشغيلات غير التفاعلية -e assume_yes=true للنظر فقط، أو
- (--disable-triggers) ل postgres ل superuser تتم الكتابات محليًا على كل هدف كـ. يتم قراءة الأقران فقط، كمستخدم التطبيق).
- **اتحاد متماثل، وليس موثوقًا للإنتاج:** تشغيله ضد موقع يسحب صفوف ذلك ❖❖❖ وقع إلى الآخرين. لا يتعامل مع الإنتاج كالأستاذ، لذا فإن تشغيله ضد جرد المختبر سيؤدي إلى نشر صفوف المختبر نحو الإنتاج. إذا كان يجب أن يكون الإنتاج هو المصدر الحقيقي، قم بتشغيله من جرد الإنتاج (أو اطلب وضع الموثوقية)
- **عبر المواقع:** يمكن أن تكون العقدة هدفًا فقط عندما تكون في الكتيب، لذا قم بتشغيل هذا ضد كل جرد موقع لتغطية كل عقدة. يقوم إعادة المسح في النهاية بالإبلاغ عن أي شيء لا يزال يحتاج إلى تشغيل.

```
# تفاعلي (افتراضي): يسأل قبل الكتابة
ansible-playbook -i hosts/<inv>/host_files/<site>.yaml
util_playbooks/reconcile_hss_drift.yaml
# نظر فقط، لا يكتب أبدًا
ansible-playbook -i ... util_playbooks/reconcile_hss_drift.yaml -e
dry_run=true
# تطبيق غير تفاعلي (CI)
ansible-playbook -i ... util_playbooks/reconcile_hss_drift.yaml -e
assume_yes=true
```

أولاً لرؤية ما انحراف، ومرة أخرى بعد التسوية (ضد `audit_hss_drift.yaml` قم بتشغيل
وعمليات العقد OmniHSS كلا الجردين) لتأكيد التقارب. انظر [تكملة](#) [هنا](#).

OCS استعادة

الملف: `util_playbooks/restore_ocs.yaml`

متعدد الماستر KeyDB من ملفات النسخ الاحتياطي. يتعامل مع تكرار OCS (CGrateS) يستعيد
من خلال الاستعادة إلى عقدة واحدة والسماح للنسخ بالتزامن إلى الآخرين.

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

عملية الاستعادة

1. OCS على جميع عقد KeyDB وCGrateS يوقف
2. لمنع التعارض مع البيانات المستعادة AOF يسمح ملفات
3. ويترك النسخ تتزامن، KeyDB إلى العقدة الأولى، يبدأ KeyDB RDB يستعيد
4. (اختياري) MySQL StoreDB يستعيد
5. (اختياري) `/etc/cgrates` يستعيد تكوين
6. على جميع العقد CGrateS يبدأ

تنسيقات ملفات النسخ الاحتياطي

المحتويات	نمط اسم الملف	النوع
ل RDB لقطة KeyDB	<code>keydb_dump_<hostname>_<timestamp>.rdb</code>	قاعدة بيانات KeyDB
MySQL تفرغ لقاعدة بيانات <code>cgrates</code>	<code>cgrates_dump_<hostname>_<timestamp>.sql</code>	قاعدة بيانات MySQL StoreDB
أرشف من دليل <code>/etc/cgrates</code>	<code>cgrates_<hostname>_<timestamp>.tar.gz</code>	التكوين

المطالبات

الوصف	مطلوب	المطالبة
KeyDB RDB المسار الكامل إلى ملف النسخ الاحتياطي	نعم	KeyDB مسار تفرغ RDB
تخطي) CGrateS ل SQL المسار الكامل إلى تفرغ (الحالي StoreDB للحفاظ على	لا	مسار تفرغ MySQL SQL
المسار الكامل إلى النسخة الاحتياطية للتكوين (تخطي للحفاظ على التكوين الحالي)	لا	مسار ملف تكوين tar.gz

IP مولد خطة

الملف: `util_playbooks/ip_plan_generator.yml`

:ينشئ وثائق الشبكة من الجرد، بما في ذلك

- (الأساسية والثانوية NICs) للمضيف IP تعيينات
- نظرة عامة على مقطع 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

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

تدقيق المنافذ

الملف: util_playbooks/get_ports.yml

.يدقق جميع المنافذ المستمعة عبر النشر وينشئ وثائق

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

ملفات الإخراج

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

البيانات المجمعة

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

استرجاع الالتقاط المحلي

الملف: util_playbooks/getLocalCapture.yml

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

```
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.

Mnesia مسح قاعدة بيانات

الملف: `util_playbooks/clear_mnesia.yml`

الخاصة بها، ويعيد تشغيلها. يعمل ضد مجموعة Mnesia يوقف الخدمة، يحذف دليل قاعدة بيانات `ss7` (OmniSS7) ومجموعة `omnimessage` (OmniMessage).

بشكل دائم وإعادة بنائها فارغة عند Mnesia **تحذير:** هذا تدميري. يتم حذف قاعدة بيانات إعادة التشغيل.

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

```
# تقييد إلى خدمة أو مضيف واحد
ansible-playbook -i hosts/customer/host_files/production.yml
util_playbooks/clear_mnesia.yml --limit ss7
```

حذف الترخيص

الملف: `util_playbooks/delete_license.yml`

ويعيد `license_server` على مجموعة `/etc/license_server/license.json` يحذف تشغيل خادم الترخيص.

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

تشغيل كتيبات الأدوات

الصياغة الأساسية

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

الخيارات الشائعة

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

الأمثلة

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

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

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

