

ACI 멀티 사이트 패브릭으로 사이트 간 L3out 구성

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[사이트 간 L3out 구성에 대해 지원되는 스키마](#)

[구성](#)

[네트워크 다이어그램](#)

[물리적 토폴로지](#)

[논리적 토폴로지](#)

[구성](#)

[스키마 구성1](#)

[패브릭 정책 구성](#)

[RTEP/ETEP 구성](#)

[스트레치 테넌트 구성](#)

[스키마 구성](#)

[스키마 생성](#)

[Site-A 템플릿 만들기](#)

[템플릿 구성](#)

[스트레치 템플릿 만들기](#)

[템플릿 첨부](#)

[고정 포트 바인딩 구성](#)

[BD 구성](#)

[호스트-A\(N9K\) 구성](#)

[Site-B 템플릿 만들기](#)

[사이트-B L3out 구성](#)

[외부 EPG 생성](#)

[외부 N9K\(Site-B\) 구성](#)

[사이트-B L3out을 사이트-A EPG에 연결\(BD\)](#)

[계약 구성](#)

[계약 생성](#)

[다음을 확인합니다.](#)

[엔드포인트 학습](#)

[ETEP/RTEP 확인](#)

[ICMP 연결성](#)

[경로 확인](#)

[문제 해결](#)

[사이트2 리프1](#)

[사이트2 스파인](#)

[사이트1 스파인](#)

[경로 구별자 항목 이해](#)

[사이트1 리프1](#)

[ELAM 확인\(Site1 Spine\)](#)

[Site1 Spine 경로 맵 확인](#)

소개

이 문서에서는 Cisco ACI(Application Centric Infrastructure) 멀티 사이트 패브릭을 사용한 사이트 간 L3out 컨피그레이션의 단계에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- 기능 ACI 멀티 사이트 패브릭 설정
- 외부 라우터/연결

사용되는 구성 요소

이 문서의 정보는 다음을 기반으로 합니다.

- MSO(Multi-Site Orchestrator) 버전 2.2(1) 이상
- ACI 버전 4.2(1) 이상
- MSO 노드
- ACI 패브릭
- Nexus 9000 Series Switch(N9K)(엔드 호스트 및 L3out 외부 장치 시뮬레이션)
- Nexus 9000 Series 스위치(N9K)(ISN(Inter-Site Network))

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

사이트 간 L3out 구성에 대해 지원되는 스키마

스키마-config1

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 사이트(A와 B) 간에 VRF(Virtual Routing and Forwarding)가 확장되었습니다.
- EPG(Endpoint Group)/BD(Bridge Domain)가 한 사이트(A)에 로컬입니다.
- L3out 로컬 사이트(B)
- L3out Local to Site(B)의 외부 EPG입니다.

- MSO에서 계약 생성 및 구성 완료

스키마-config2

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 사이트(A와 B) 간에 VRF가 확장되었습니다.
- 사이트(A와 B) 간에 EPG/BD 확장
- L3out을 로컬 사이트(B)로 설정합니다.
- L3out Local to Site(B)의 외부 EPG입니다.
- 계약 컨피그레이션은 MSO에서 수행할 수 있습니다. 또는 각 사이트에는 APIC(Application Policy Infrastructure Controller)에서 로컬 계약을 생성하고 확장된 EPG와 L3out 외부 EPG 간에 로컬로 연결됩니다. 이 경우 로컬 계약 관계 및 정책 구현에 Shadow External_EPG가 필요하므로 사이트-A에 나타납니다.

스키마-config3

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 사이트(A와 B) 간에 VRF가 확장되었습니다.
- 사이트(A와 B) 간에 EPG/BD 확장
- L3out을 로컬 사이트(B)로 설정합니다.
- L3out의 외부 EPG가 사이트(A와 B) 간에 확장되었습니다.
- 계약 컨피그레이션은 MSO에서 수행할 수 있습니다. 또는 각 사이트에는 APIC에서 로컬 계약을 생성하고 스트레치된 EPG와 스트레치된 외부 EPG 간에 로컬로 첨부됩니다.

스키마-config4

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 사이트(A와 B) 간에 VRF가 확장되었습니다.
- 각 사이트에 대해 한 사이트(A) 또는 EPG/BD 로컬(사이트 A의 EPG-A, 사이트 B의 EPG-B)
- L3out을 한 사이트에 로컬로(B) 또는 외부 연결에 대한 이중화를 위해 각 사이트에 로컬(로컬 사이트 A에서 로컬 사이트 B, 로컬 사이트 B)로 L3out을 설정할 수 있습니다.
- L3out의 외부 EPG가 사이트(A와 B) 간에 확장되었습니다.
- 계약 컨피그레이션은 MSO에서 수행할 수 있습니다. 또는 각 사이트에는 APIC에서 로컬 계약을 생성하고 확장 EPG와 확장 외부 EPG 간에 로컬로 첨부됩니다.

Schema-config5(전송 라우팅)

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 사이트(A와 B) 간에 VRF가 확장되었습니다.
- 각 사이트의 로컬 L3out(로컬-사이트 A 및 로컬-사이트 B)
- 각 사이트(A 및 B)에 대한 로컬 외부 EPG입니다.
- 계약 컨피그레이션은 MSO에서 수행할 수 있습니다. 또는 각 사업장은 APIC에서 로컬 계약을 생성하고 외부 EPG 로컬 및 새도우 외부 EPG 로컬 간에 로컬로 첨부됩니다.

Schema-config5(InterVRF 트랜짓 라우팅)

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 각 사이트(A 및 B)에 로컬인 VRF
- 각 사이트의 로컬 L3out(로컬-사이트 A 및 로컬-사이트 B)

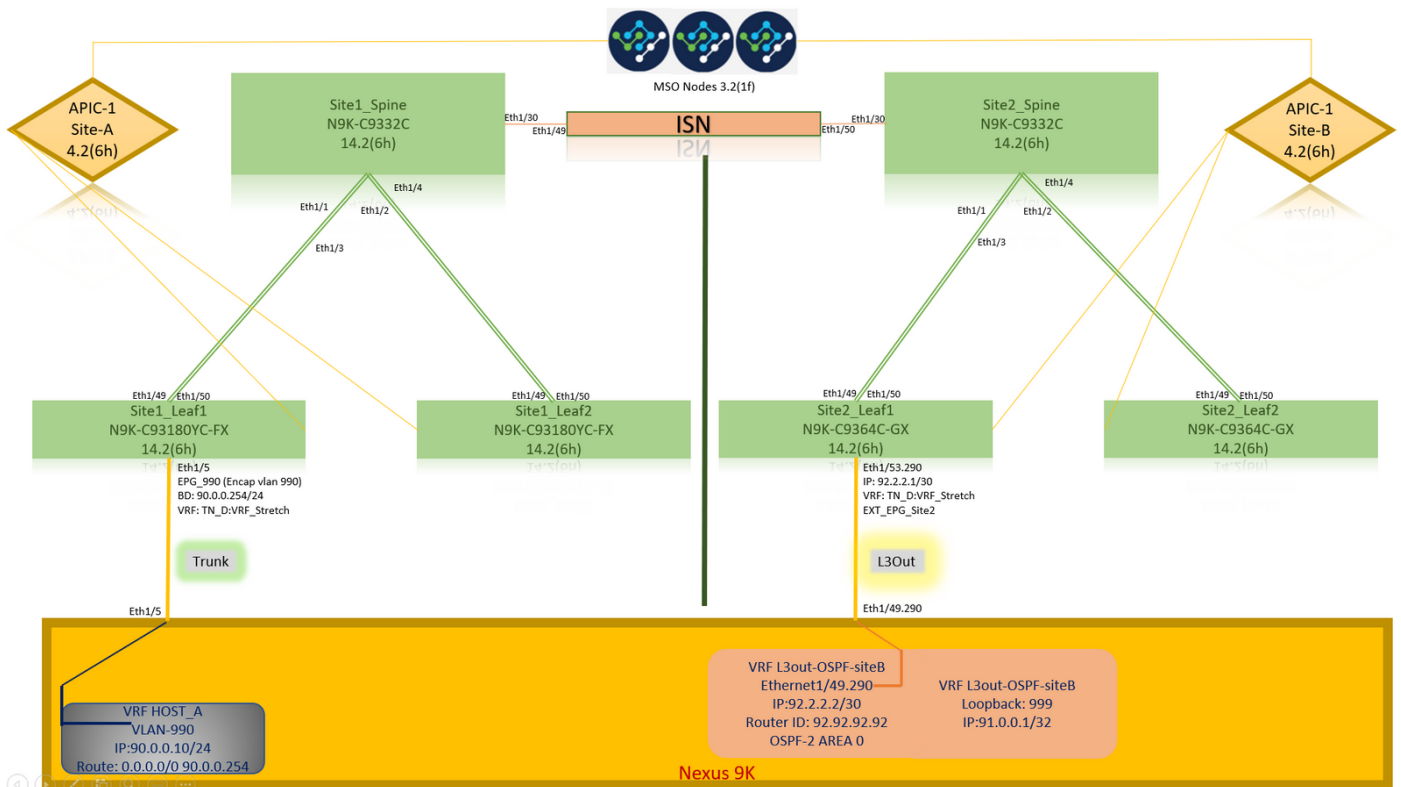
- 각 사이트(A 및 B)에 대한 로컬 외부 EPG입니다.
- 계약 컨피그레이션은 MSO에서 수행할 수 있습니다. 또는 각 사업장은 APIC에서 로컬 계약을 생성하고 외부 EPG 로컬 및 새도우 외부 EPG 로컬 간에 로컬로 첨부됩니다.

참고: 이 문서에서는 기본 사이트 간 L3out 구성 단계 및 확인을 제공합니다. 이 예에서는 Schema-config1이 사용됩니다.

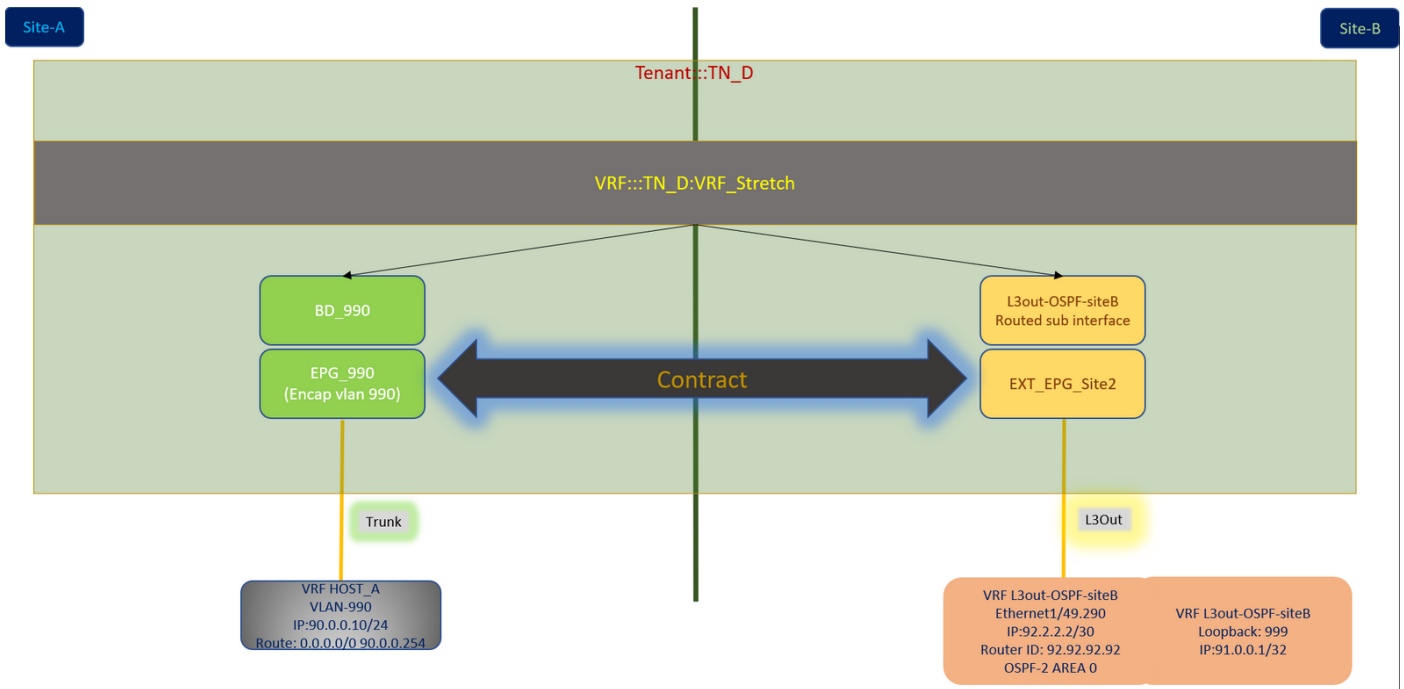
구성

네트워크 다이어그램

물리적 토폴로지



논리적 토폴로지



구성

이 예에서는 Schema-config1을 사용합니다. 그러나 스트레치된 개체가 특정 사이트 템플릿 대신 스트레치된 템플릿에 있어야 한다는 점을 제외하면 지원되는 다른 스키마 구성에 대해 유사한 방식으로(계약 관계식에 따라 약간 변경) 이 구성을 완료할 수 있습니다.

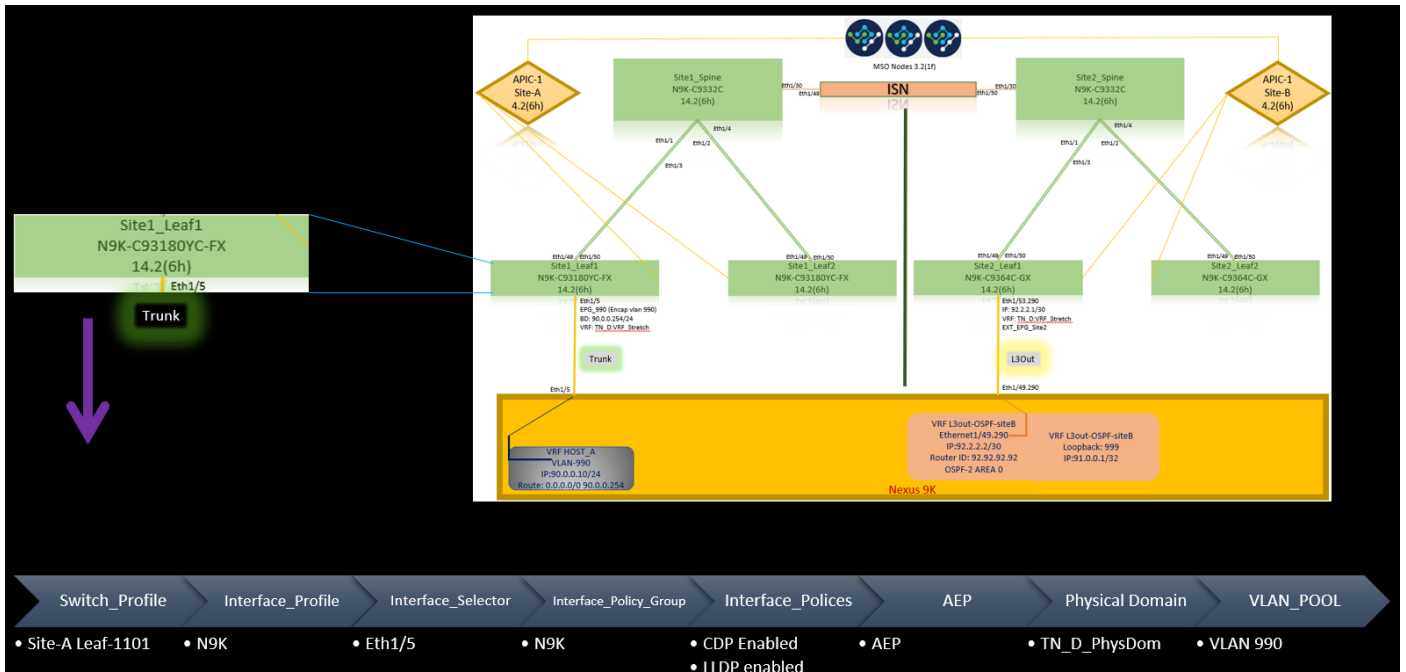
스키마 구성1

- 사이트 간에 테넌트가 확장되었습니다(A와 B).
- 사이트(A와 B) 간에 VRF가 확장되었습니다.
- EPG/BD에서 한 사이트(A)로 로컬.
- L3out 로컬 사이트(B)
- L3out Local to Site(B)의 외부 EPG입니다.
- MSO에서 계약 생성 및 구성 완료
사이트 간 [L3Out 지침 및 제한 사항을 검토합니다.](#)
- 사이트 간 L3out이 포함된 지원되지 않는 구성: 다른 사이트 L3out을 통해 외부 소스로부터 멀티캐스트를 수신하는 사이트의 멀티캐스트 수신기입니다. 외부 소스로부터 사이트에서 수신된 멀티캐스트는 다른 사이트로 전송되지 않습니다. 사이트의 수신자가 외부 소스로부터 멀티캐스트를 수신할 경우 로컬 L3out에서 수신해야 합니다. 내부 멀티캐스트 소스는 PIM-SM을 사용하여 ASM(Source Multicast)을 외부 수신자에게 전송합니다. 내부 멀티캐스트 소스는 로컬 L3out에서 외부 RP(Rendezvous Point)에 연결할 수 있어야 합니다. GOLF(Giant OverLay Fabric). 외부 EPG에 대한 기본 그룹

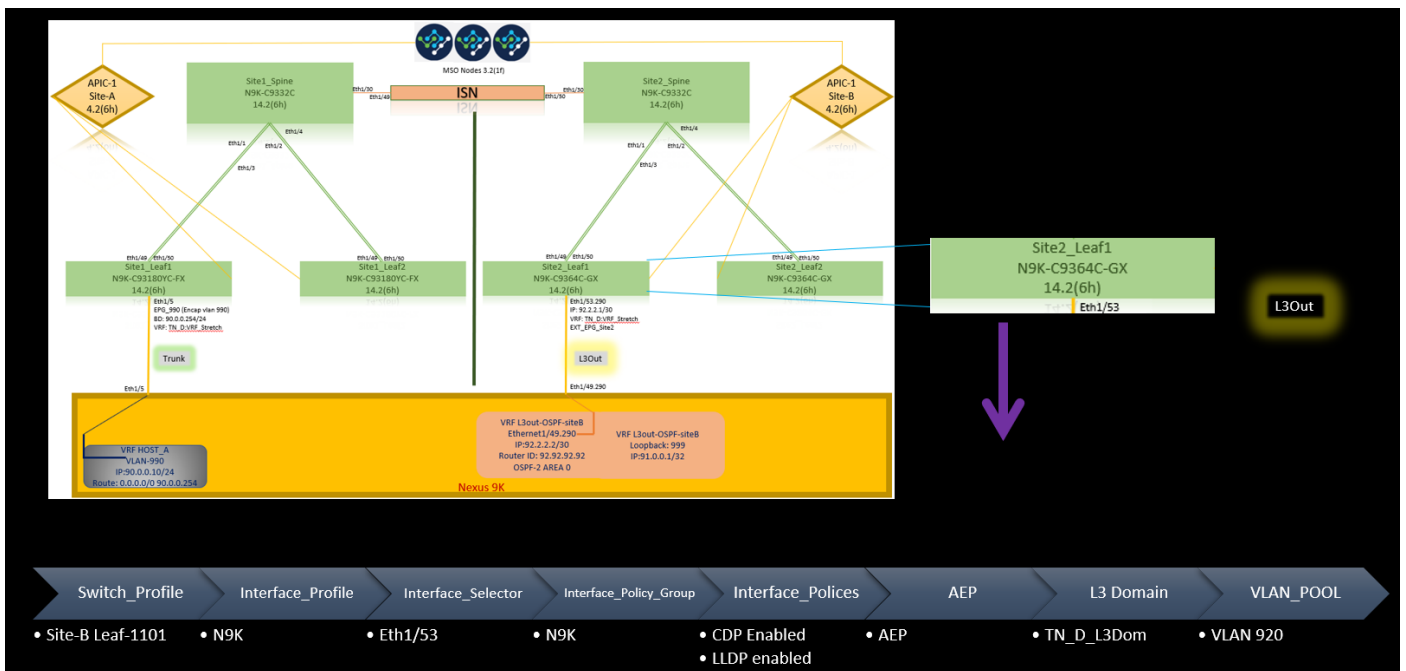
패브릭 정책 구성

각 사이트의 패브릭 정책은 특정 테넌트/EPG/고정 포트 바인딩 또는 L3out 물리적 연결에 연결되므로 필수적인 컨피그레이션입니다. 패브릭 정책이 잘못 구성되면 APIC 또는 MSO에서 논리적 컨피그레이션이 실패할 수 있으므로 랩 설정에서 사용된 패브릭 정책 컨피그레이션이 제공됩니다. MSO 또는 APIC에서 어떤 객체가 어떤 객체에 연결되어 있는지 이해하는 데 도움이 됩니다.

Site-A의 Host_A 연결 패브릭 정책



사이트-B의 L3out 연결 패브릭 정책



선택 단계

각 연결에 대한 패브릭 정책이 준비되면 모든 leaf/spine이 검색되고 해당 APIC 클러스터에서 연결할 수 있는지 확인할 수 있습니다. 그런 다음 두 사이트(APIC 클러스터)가 MSO에서 연결 가능하며 다중 사이트 설정이 작동(및 IPN 연결)되는지 확인할 수 있습니다.

RTEP/ETEP 구성

RTEP(Routable Tunnel Endpoint Pool) 또는 ETEP(External Tunnel Endpoint Pool)는 사이트 간 L3out에 필요한 컨피그레이션입니다. MSO의 이전 버전은 "Routable TEP Pools(라우팅 가능한 TEP 풀)"를, 새로운 버전의 MSO는 "External TEP Pools(외부 TEP 풀)"를 표시하지만 두 가지 모두 동의어입니다. 이러한 TEP 풀은 VRF "Overlay-1"을 통한 BGP(Border Gateway Protocol) 이더넷

VPN(EVPN)에 사용됩니다.

L3out의 외부 경로는 BGP EVPN을 통해 다른 사이트로 광고됩니다. 이 RTEP/ETEP는 원격 리프 컨피그레이션에 사용되므로 APIC에 이미 존재하는 ETEP/RTEP 컨피그레이션이 있는 경우 MSO에서 가져와야 합니다.

다음은 MSO GUI에서 ETEP를 구성하는 단계입니다. 버전은 3.X MSO이므로 ETEP를 표시합니다. ETEP 풀은 각 사이트에서 고유해야 하며 각 사이트의 내부 EPG/BD 서브넷과 중복되어서는 안 됩니다.

사이트-A

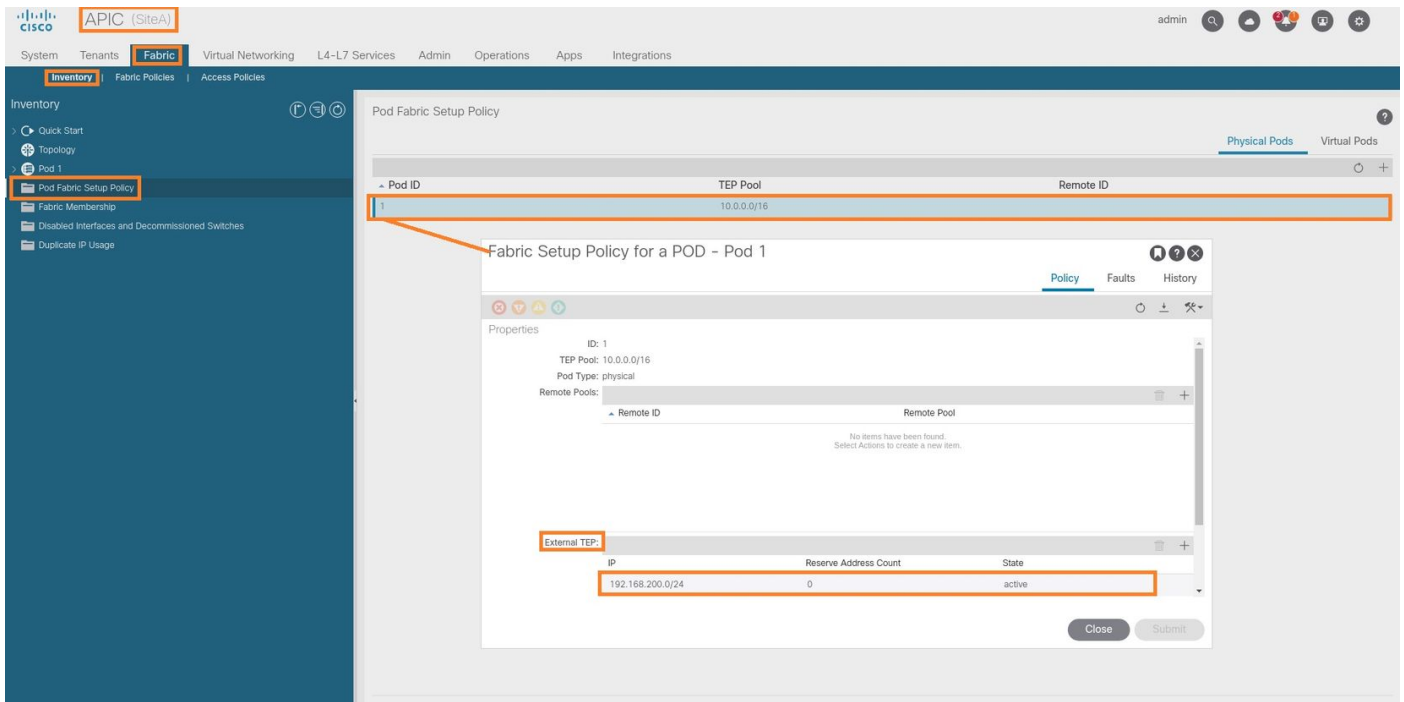
1단계. MSO GUI 페이지(웹 페이지에서 멀티 사이트 컨트롤러 열기)에서 Infrastructure(인프라) > Infra Configuration(인프라 컨피그레이션)을 선택합니다. Configure Infra를 클릭합니다.



2단계. Configure Infra(인프라 구성)에서 **Site-A**, Inside Site-A를 선택하고 **pod-1**을 선택한 다음 inside pod-1에서 **Site-A**의 외부 TEP IP 주소로 외부 TEP 풀을 구성합니다(이 예에서는 192.168.200.0/24). Site-A에 Multi-POD가 있는 경우 다른 Pod에 대해 이 단계를 반복합니다.



3단계. APIC GUI에서 ETEP 풀의 컨피그레이션을 확인하려면 Fabric(패브릭) > Inventory(인벤토리) > Pod Fabric Setup Policy(포드 패브릭 설정 정책) > Pod-ID(더블 클릭하여 [Fabric Setup Policy a POD-Pod-x]) > External TEP(외부 TEP)를 선택합니다.



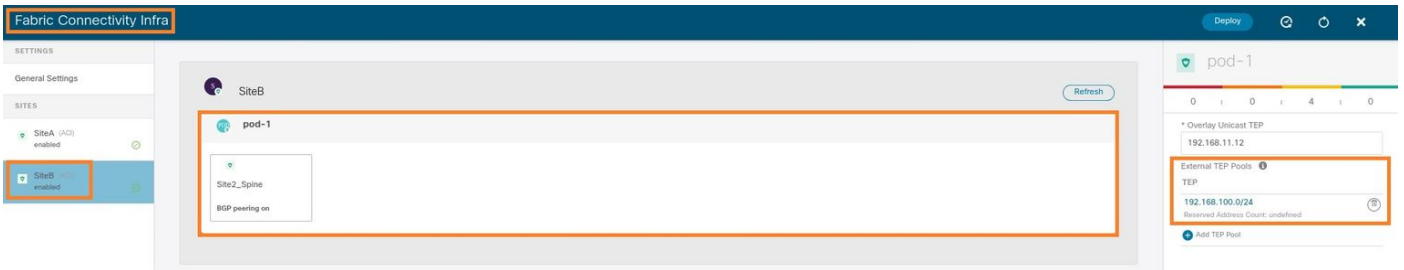
다음 명령을 사용하여 컨피그레이션을 확인할 수도 있습니다.

```
moquery -c fabricExtRoutablePodSubnet
moquery -c fabricExtRoutablePodSubnet -f 'fabric.ExtRoutablePodSubnet.pool=="192.168.200.0/24"'
```

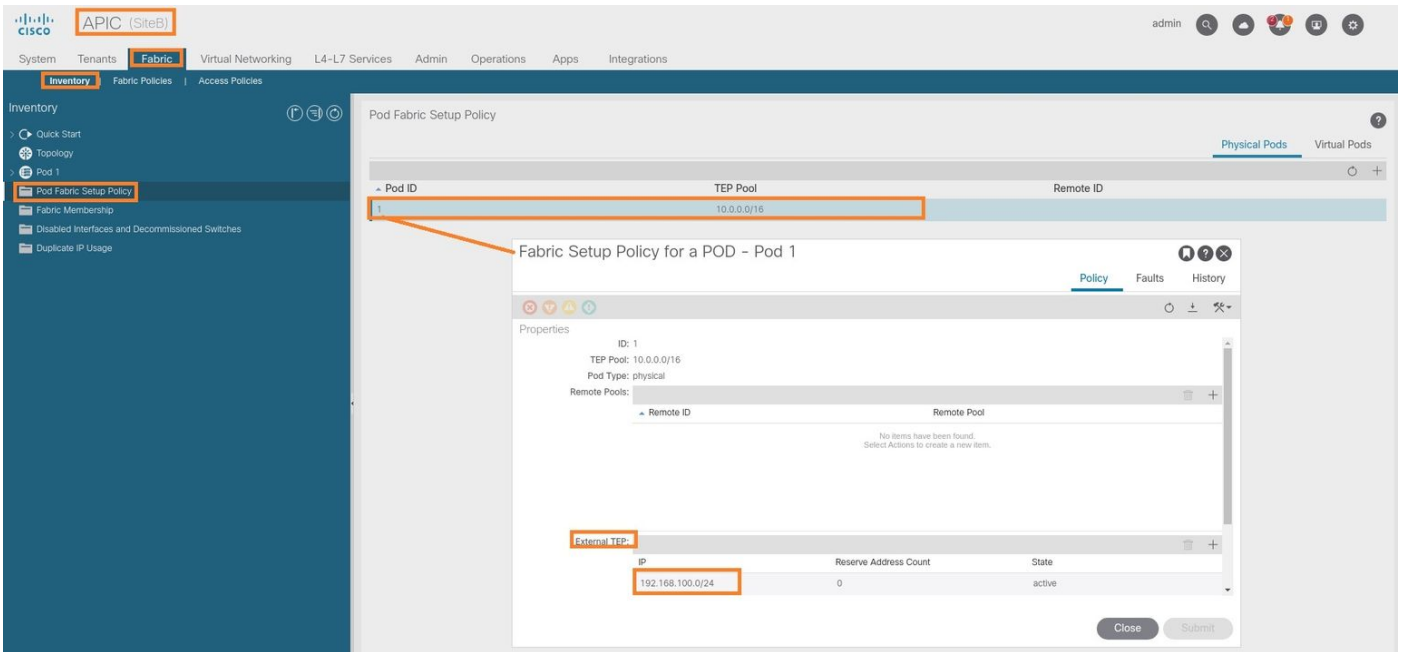
```
APIC1# moquery -c fabricExtRoutablePodSubnet
Total Objects shown: 1
# fabric.ExtRoutablePodSubnet
pool                : 192.168.200.0/24
annotation          : orchestrator:misc
childAction         :
descr               :
dn                  : uni/controller/setupp01/setupp-1/extrtpodsubnet-[192.168.200.0/24]
extMngdBy           :
lcOwn               : local
modTs               : 2021-07-19T14:45:22.387+00:00
name                :
nameAlias           :
reserveAddressCount : 0
rn                  : extrtpodsubnet-[192.168.200.0/24]
state               : active
status              :
uid                 : 0
```

사이트-B

1단계. Site-B용 외부 TEP 풀을 구성합니다(Site-A와 동일한 단계). MSO GUI 페이지(웹 페이지에서 멀티 사이트 컨트롤러 열기)에서 Infrastructure(인프라) > **Infra Configuration(인프라 컨피그레이션)**을 선택합니다. Configure **Infra**. Inside Configure Infra를 클릭하고 **Site-B**를 선택합니다. Site-B 내에서 **pod-1**을 선택한 다음 inside pod-1에서 Site-B의 외부 TEP IP 주소로 **외부 TEP 풀**을 구성합니다(이 예에서는 192.168.100.0/24). Site-B에 Multi-POD가 있는 경우 다른 Pod에 대해 이 단계를 반복합니다.



2단계. APIC GUI에서 ETEP 풀의 컨피그레이션을 확인하려면 **Fabric(패브릭) > Inventory(인벤토리) > Pod Fabric Setup Policy(포드 패브릭 설정 정책) > Pod-ID(더블 클릭하여 [Fabric Setup Policy a POD-Pod-x]) > External TEP(외부 TEP)**를 선택합니다.



Site-B APIC의 경우 ETEP 주소 풀을 확인하려면 이 명령을 입력합니다.

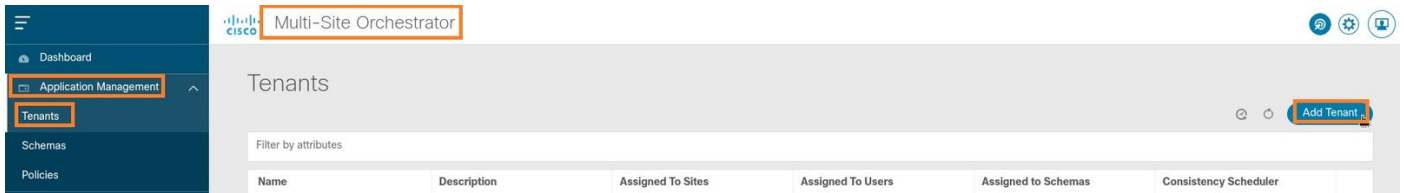
```

apic1# moquery -c fabricExtRoutablePodSubnet -f
'fabric.ExtRoutablePodSubnet.pool=="192.168.100.0/24"'
Total Objects shown: 1
# fabric.ExtRoutablePodSubnet
pool                : 192.168.100.0/24
annotation          : orchestrator:misc <<< This means, configuration pushed from MSO.
childAction         :
descr               :
dn                  : uni/controller/setupp01/setupp-1/extrtpodsubnet-[192.168.100.0/24]
extMngdBy           :
lcOwn               : local
modTs               : 2021-07-19T14:34:18.838+00:00
name                :
nameAlias           :
reserveAddressCount : 0
rn                  : extrtpodsubnet-[192.168.100.0/24]
state               : active
status              :
uid                 : 0
  
```

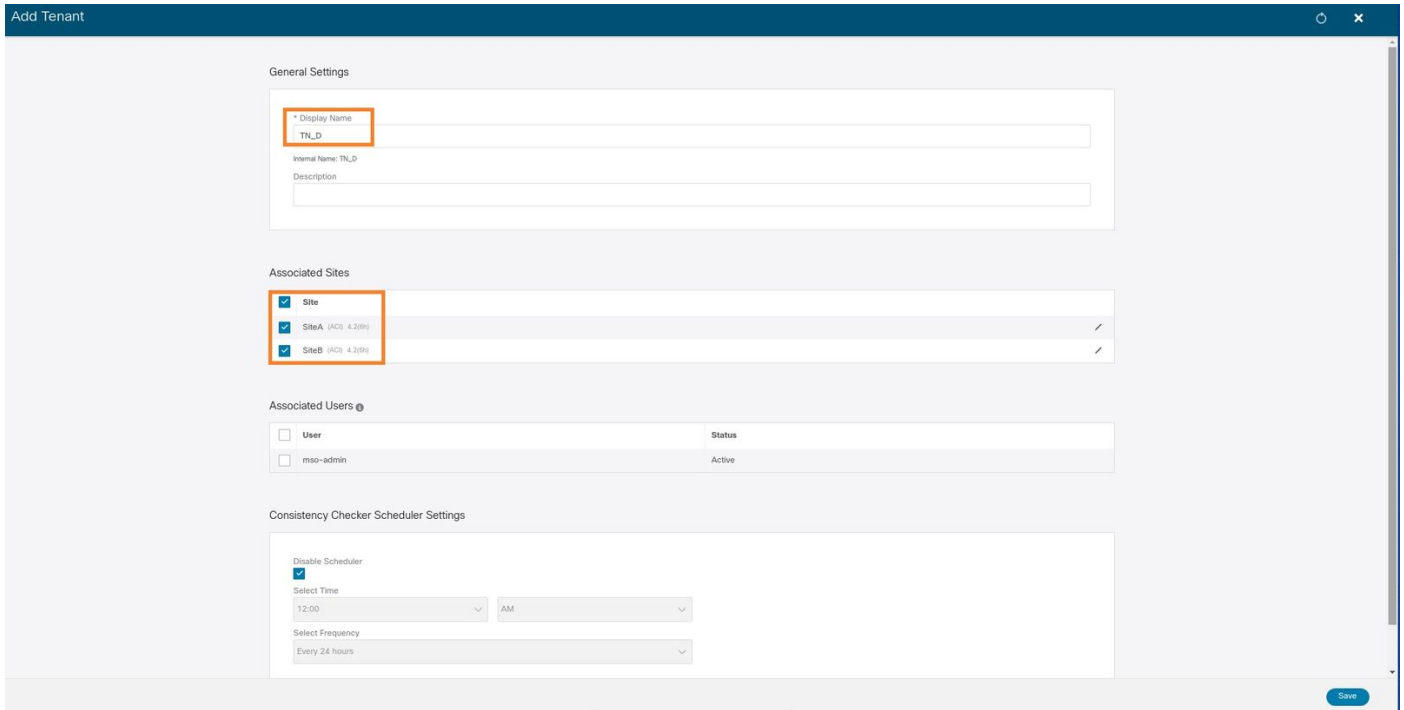
스트레치 테넌트 구성

1단계. MSO GUI에서 Application Management > Tenants를 선택합니다. Add Tenant를 클릭합니

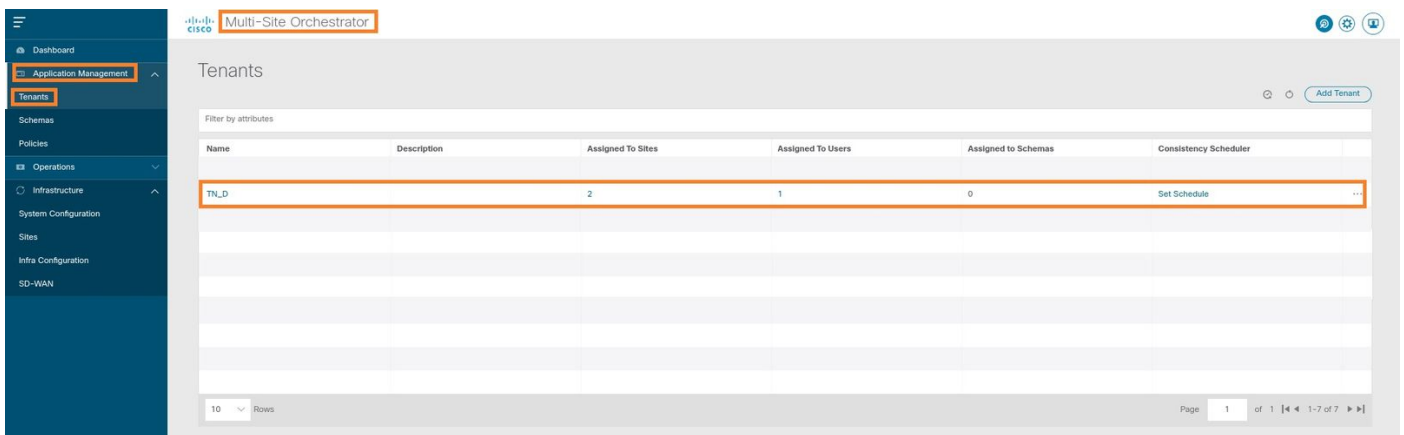
다. 이 예에서 테넌트 이름은 "TN_D"입니다.



2단계. **Display Name**(표시 이름) 필드에 테넌트의 이름을 입력합니다. **Associated Sites**(연결된 사이트) 섹션에서 **Site A** 및 **Site B** 확인란을 선택합니다.

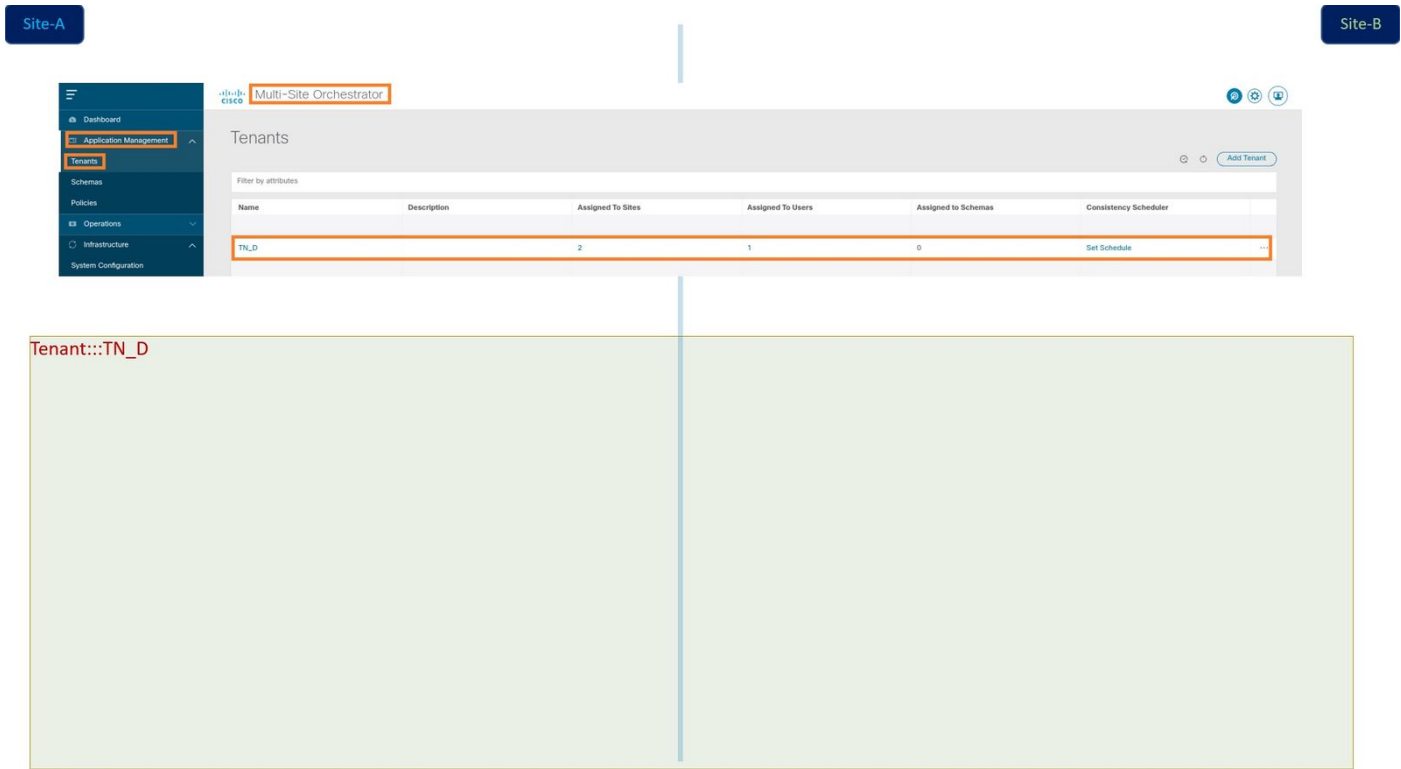


3단계. 새 테넌트 "Tn_D"가 생성되었는지 확인합니다.

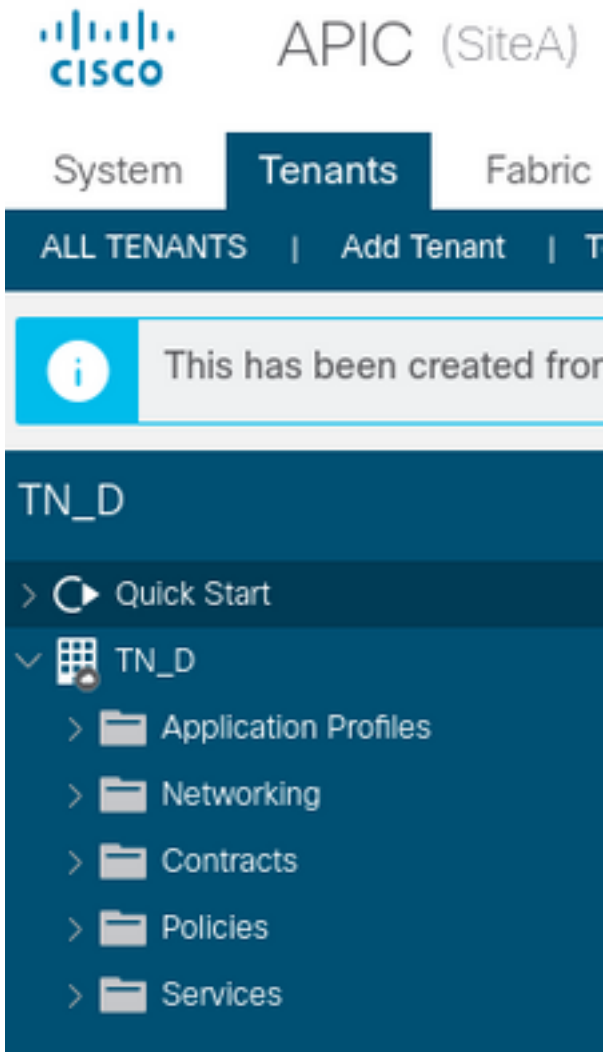


논리적 보기

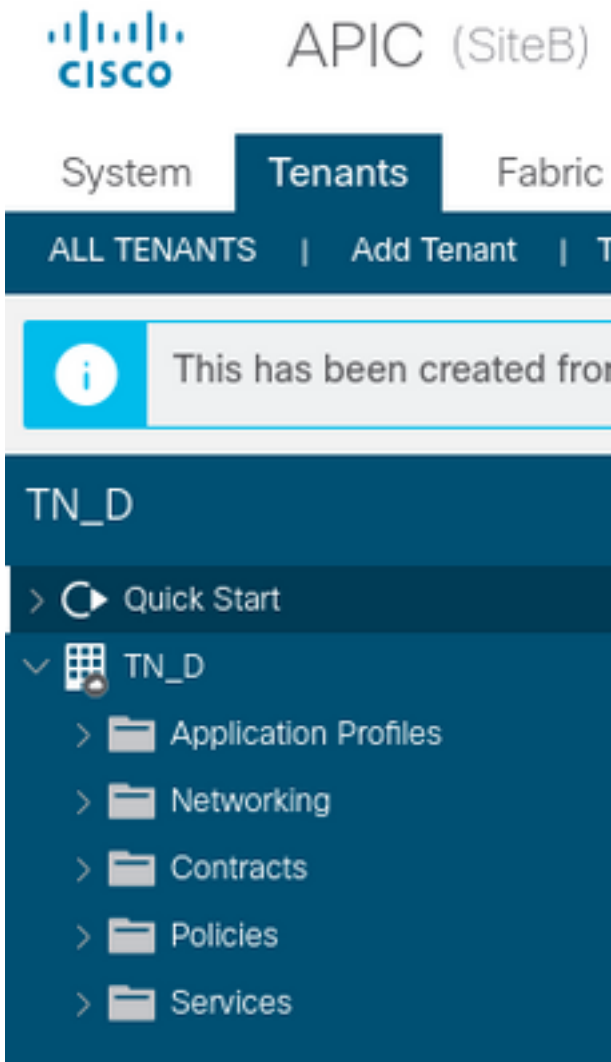
MSO에서 테넌트를 생성할 때 기본적으로 Site-A 및 Site-B에 테넌트를 생성합니다. 스트레치 테넌트입니다. 이 예에서는 이 테넌트의 논리적 보기가 표시됩니다. 이 논리적 보기는 테넌트 TN_D가 Site-A와 Site-B 간에 확장된 테넌트를 이해하는 데 도움이 됩니다.



각 사이트의 APIC에서 논리적 보기를 확인할 수 있습니다. Site-A와 Site-B 모두 "TN_D" 테넌트가 생성된 것을 확인할 수 있습니다.



동일한 스트레치된 테넌트 "TN_D"가 사이트-B에서도 생성됩니다.



이 명령은 MSO에서 푸시된 테넌트를 보여 주며 확인을 위해 사용할 수 있습니다. 두 사이트의 APIC에서 이 명령을 실행할 수 있습니다.

```
APIC1# moquery -c fvTenant -f 'fv.Tenant.name=="TN_D"'
Total Objects shown: 1
# fv.Tenant
name          : TN_D
annotation    : orchestrator:misc
childAction   :
descr         :
dn            : uni/tn-TN_D
extMngdBy     : misc
lcOwn         : local
modTs        : 2021-09-17T21:42:52.218+00:00
monPolDn      : uni/tn-common/monepg-default
nameAlias     :
ownerKey      :
ownerTag      :
rn            : tn-TN_D
status        :
uid           : 0
```

```
apic1# moquery -c fvTenant -f 'fv.Tenant.name=="TN_D"'
Total Objects shown: 1
```

```

# fv.Tenant
name : TN_D
annotation : orchestrator:msc
childAction :
descr :
dn : uni/tn-TN_D
extMngdBy : msc
lcOwn : local
modTs : 2021-09-17T21:43:04.195+00:00
monPolDn : uni/tn-common/monepg-default
nameAlias :
ownerKey :
ownerTag :
rn : tn-TN_D
status :
uid : 0

```

스키마 구성

다음으로, 총 세 개의 템플릿이 있는 스키마를 생성합니다.

1. 사이트-A 템플릿: Site-A용 템플릿은 Site-A에만 연결되므로 해당 템플릿의 논리적 객체 컨피그레이션은 Site-A의 APIC에만 푸시할 수 있습니다.
2. 사이트-B 템플릿: Site-B용 템플릿은 Site-B에만 연결되므로 해당 템플릿의 논리적 객체 컨피그레이션은 Site-B의 APIC에만 푸시할 수 있습니다.
3. 확장 템플릿: 확장된 템플릿은 두 사이트와 연결되며 확장된 템플릿의 논리적 컨피그레이션은 APIC의 두 사이트로 푸시할 수 있습니다.

스키마 생성

스키마는 MSO에서 로컬에서 중요하며 APIC에 어떤 객체도 생성하지 않습니다. 스키마 컨피그레이션은 각 컨피그레이션의 논리적 분리입니다. 동일한 테넌트에 대해 여러 스키마를 가질 수 있으며 각 스키마 내에 여러 템플릿을 포함할 수도 있습니다.

예를 들어 테넌트 X에 대한 데이터베이스 서버에 대한 스키마를 가질 수 있으며 응용 프로그램 서버는 동일한 테넌트-X에 대해 다른 스키마를 사용합니다. 이를 통해 각 특정 애플리케이션 관련 컨피그레이션을 분리할 수 있으며 문제를 디버깅해야 할 때 쉽게 분리할 수 있습니다. 또한 정보를 쉽게 찾을 수 있습니다.

테넌트의 이름으로 스키마를 생성합니다(예: TN_D_Schema). 그러나 스키마 이름이 테넌트 이름으로 시작되도록 할 필요는 없습니다. 어떤 이름으로도 스키마를 생성할 수 있습니다.

1단계. 응용 프로그램 관리 > 스키마를 선택합니다. 스키마 추가를 누릅니다.



2단계. 이름 필드에 스키마 이름을 입력합니다. 이 예에서는 "TN_D_Schema"이지만 환경에 적합한 이름을 유지할 수 있습니다. Add(추가)를 클릭합니다.

General ✕

* Name

TN_D_Schema

Description

Schema for Tenant TN_D

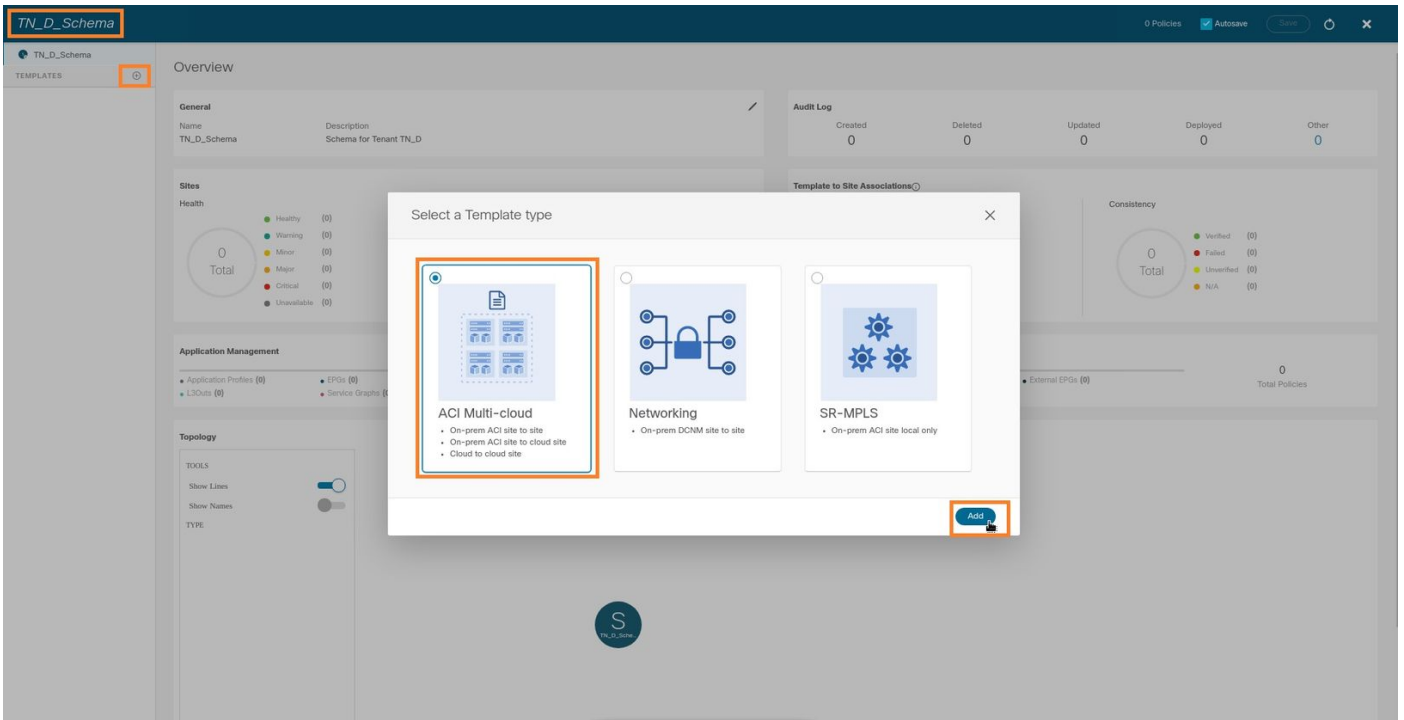
Add

3단계. "TN_D_Schema" 스키마가 생성되었는지 확인합니다.

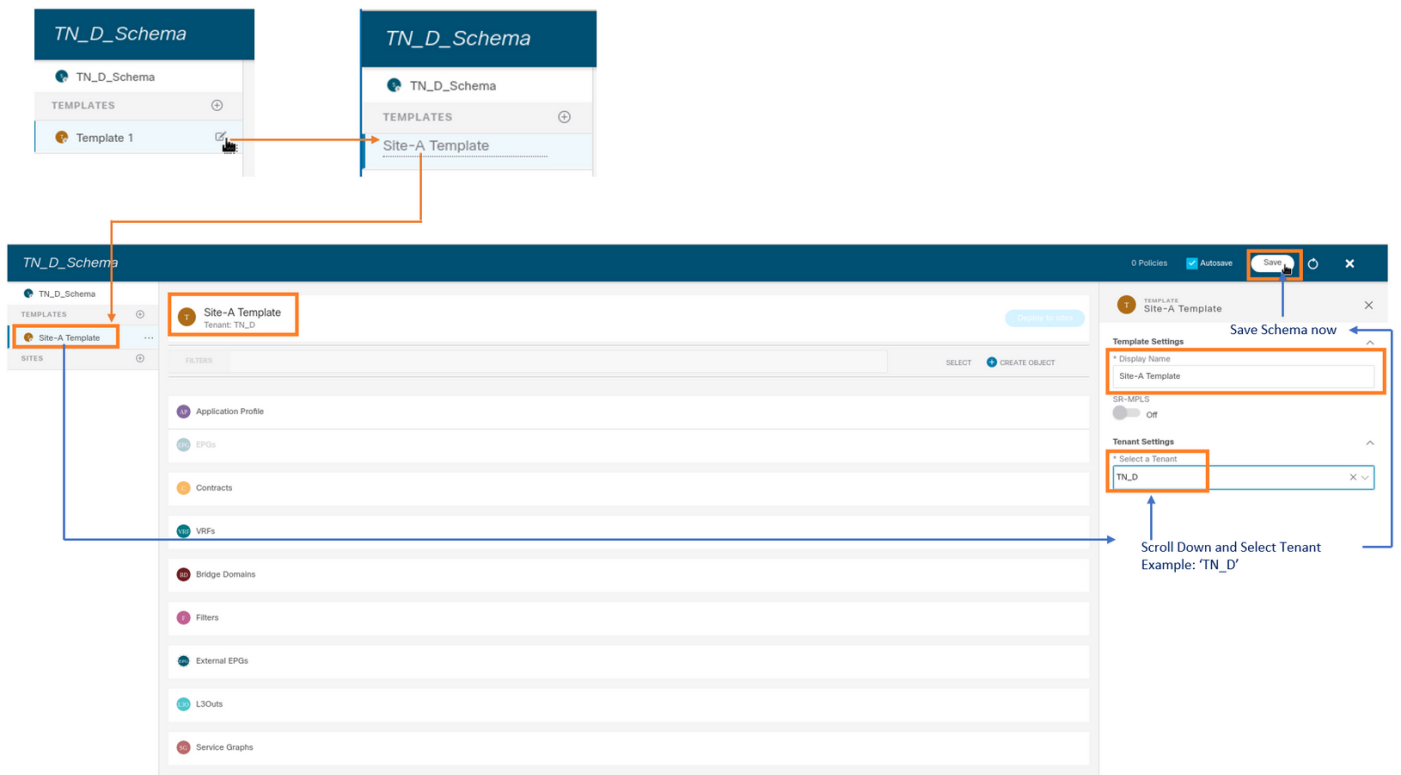
Site-A 템플릿 만들기

1단계. 스키마 내에 템플릿을 추가합니다.

1. 템플릿을 생성하려면 생성한 스키마 아래의 템플릿을 누릅니다. 템플릿 유형 선택 대화 상자가 표시됩니다.
2. **ACI Multi-cloud**를 선택합니다.
3. Add(추가)를 클릭합니다.



2단계. 템플릿의 이름을 입력합니다. 이 템플릿은 사이트-A에만 적용되므로 템플릿 이름은 "사이트-A 템플릿"입니다. 템플릿이 생성되면 특정 테넌트를 템플릿에 연결할 수 있습니다. 이 예에서는 테넌트 "TN_D"가 연결됩니다.



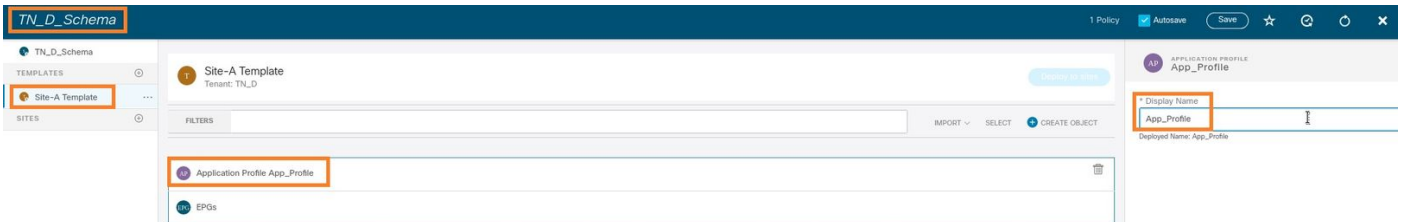
템플릿 구성

애플리케이션 프로파일 컨피그레이션

1단계. 생성한 스키마에서 **Site-A** 템플릿을 선택합니다. Add Application Profile을 클릭합니다.



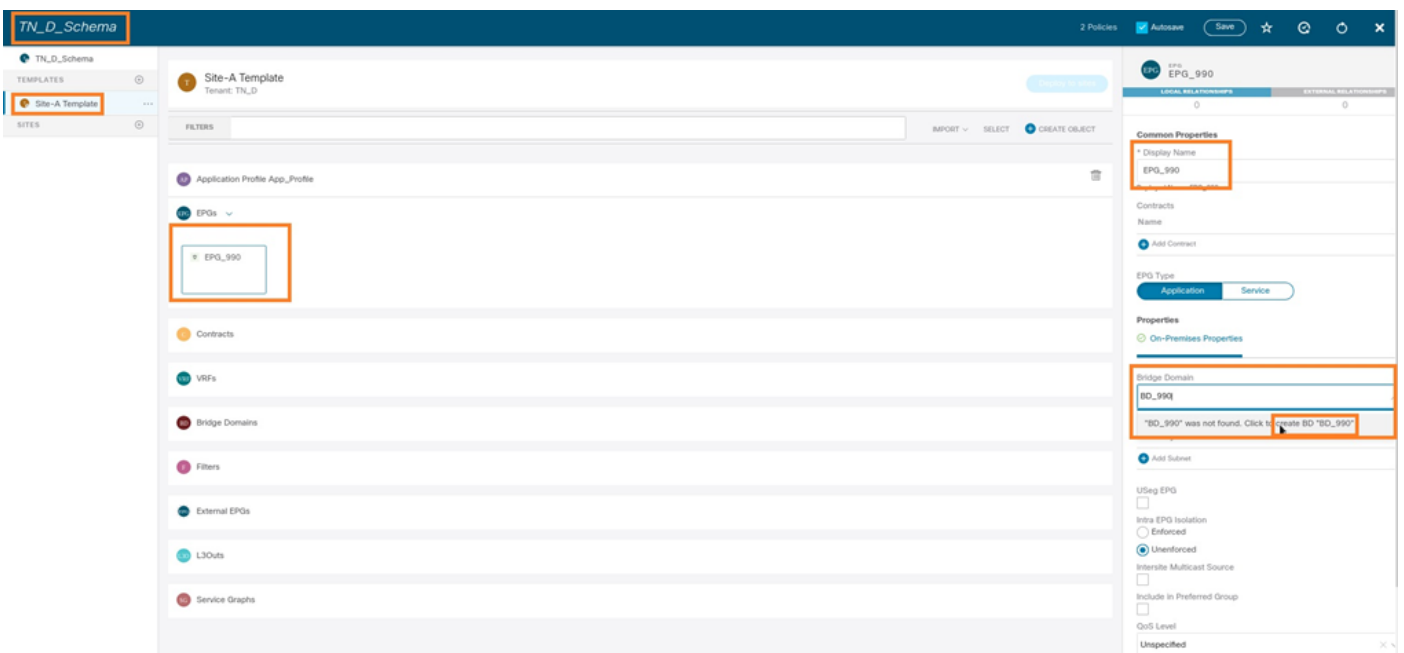
2단계. Display Name(표시 이름) 필드에 애플리케이션 프로파일 이름 App_Profile을 입력합니다



3단계. 다음 단계는 EPG를 생성하는 것입니다. 애플리케이션 프로파일 아래에 EPG를 추가하려면 Site-A 템플릿 아래에서 Add EPG(EPG 추가)를 클릭합니다. EPG 컨피그레이션 내에서 새 EPG가 생성되었음을 확인할 수 있습니다.



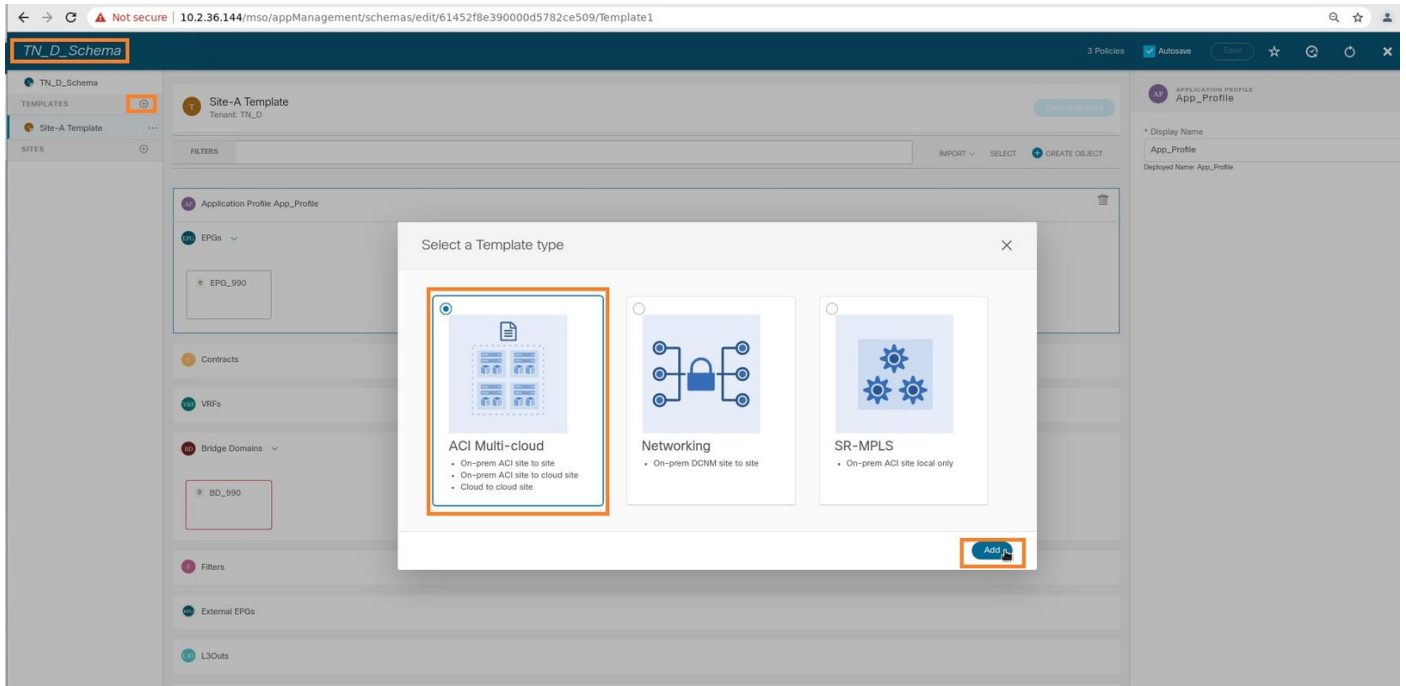
4단계. BD 및 VRF에 EPG를 연결하려면 EPG에서 BD 및 VRF를 추가해야 합니다. Site-A 템플릿을 선택합니다. Display Name(표시 이름) 필드에 EPG의 이름을 입력하고 새 BD를 연결합니다(새 BD를 생성하거나 기존 BD를 연결할 수 있음).



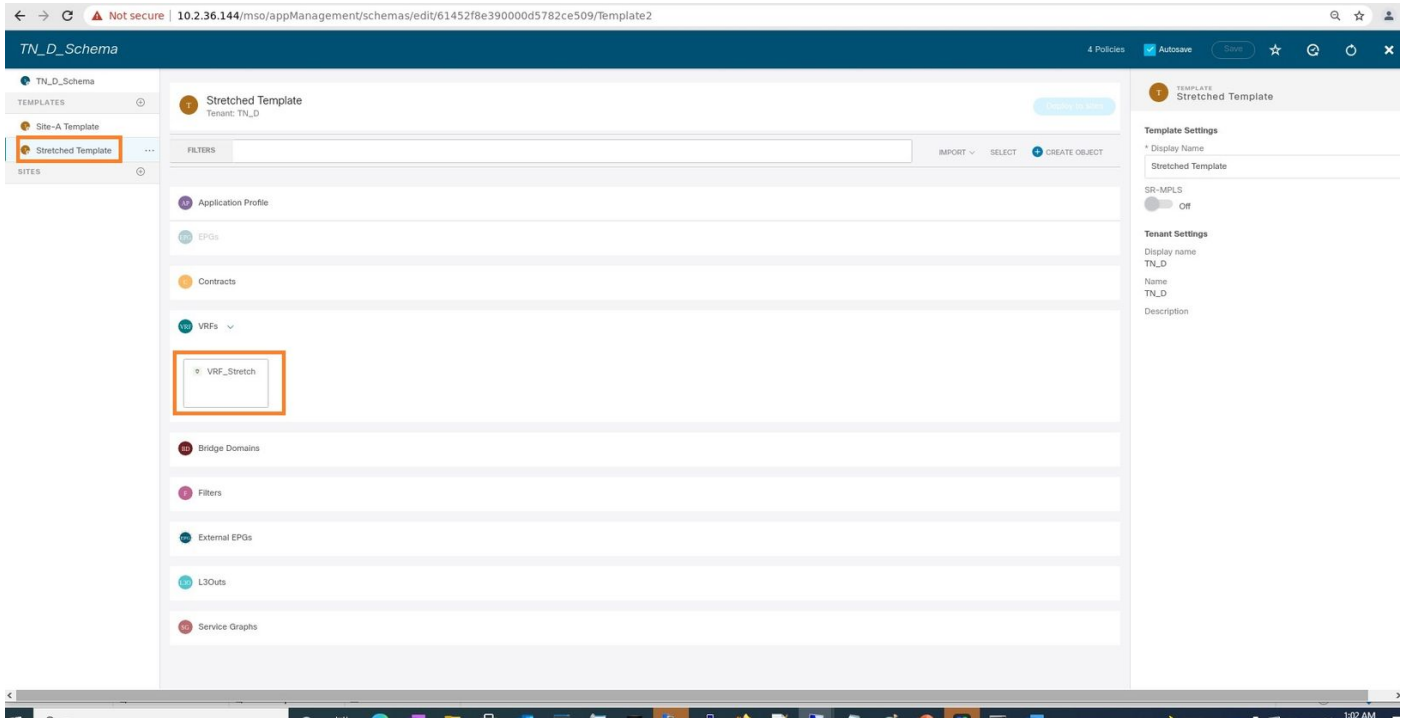
BD에 VRF를 연결해야 하지만 이 경우 VRF가 확장됩니다. 확장된 VRF로 스트레치된 템플릿을 생성한 다음 사이트별 템플릿(사이트 A 템플릿의 경우)에서 해당 VRF를 BD에 연결할 수 있습니다.

스트레치 템플릿 만들기

1단계. 스트레치 템플릿을 생성하려면 TN_D_Schema 아래에서 템플릿을 클릭합니다. 템플릿 유형 선택 대화 상자가 표시됩니다. ACI Multi-cloud를 선택합니다. Add(추가)를 클릭합니다. 템플릿의 확장 템플릿을 입력합니다. 스트레치된 템플릿의 이름을 입력할 수 있습니다.

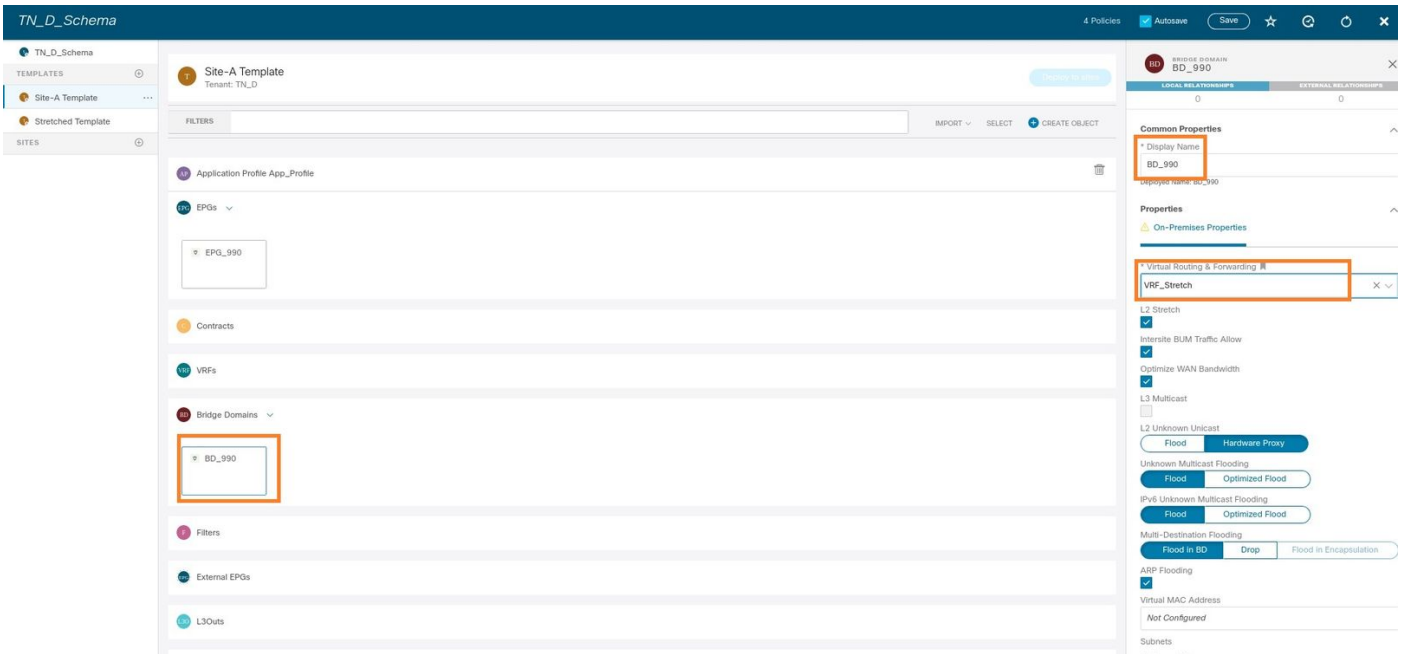


2단계. [스트레치된 템플릿]을 선택하고 이름이 VRF_Stretch인 VRF를 생성합니다. (VRF의 이름을 입력할 수 있습니다.)



BD는 Site-A 템플릿에서 EPG 생성으로 생성되었지만 VRF가 연결되어 있지 않으므로 확장 템플릿에 생성된 VRF를 연결해야 합니다.

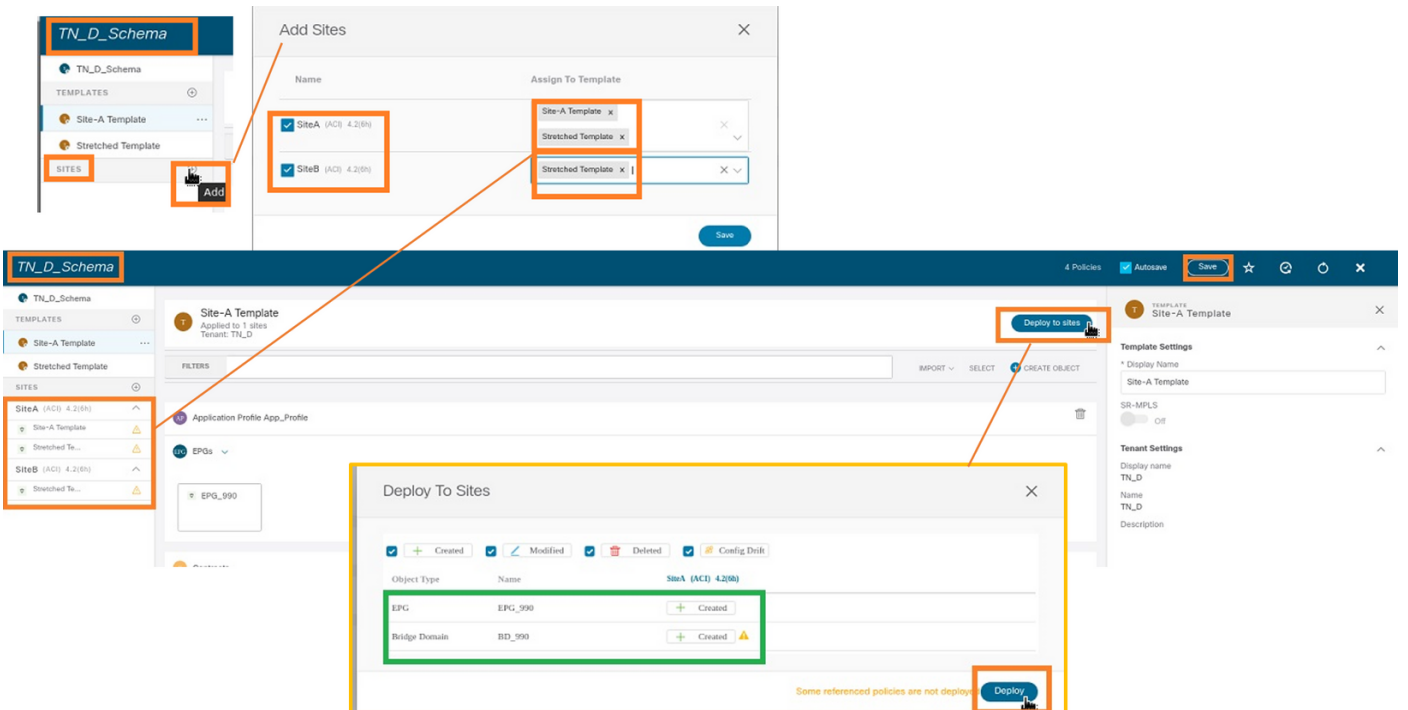
3단계. Site-A Template > BD_990을 선택합니다. Virtual Routing & Forwarding 드롭다운 목록에서 VRF_Stretch를 선택합니다. (이 섹션의 2단계에서 생성한 항목)



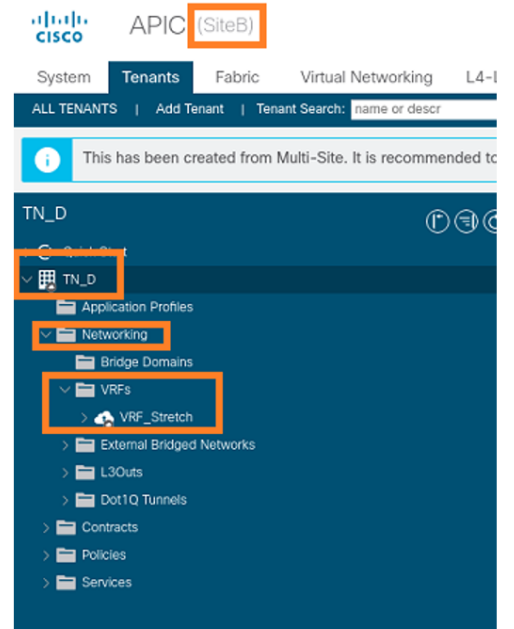
템플릿 첨부

다음 단계는 사이트-A 템플릿을 사이트-A만 첨부하고 확장 템플릿을 두 사이트에 모두 연결해야 합니다. 각 사이트에 템플릿을 배포하려면 스키마 내부의 사이트에 배포를 클릭합니다.

1단계. 템플릿에 사이트를 추가하려면 TN_D_Schema > SITES 아래의 + 기호를 클릭합니다. Assign to Template(템플릿에 할당) 드롭다운 목록에서 해당 사이트에 대한 각 템플릿을 선택합니다.



2단계. Site-A에 EPG가 있고 BD가 생성되지만 Site-B는 MSO의 Site-A에만 적용되므로 동일한 EPG/BD가 생성되지 않습니다. 그러나 VRF가 확장 템플릿에서 생성되어 두 사이트에서 생성된다는 것을 확인할 수 있습니다.



3단계. 다음 명령을 사용하여 컨피그레이션을 확인합니다.

```

APIC1# moquery -c fvAEPg -f 'fv.AEPg.name=="EPG_990"'
Total Objects shown: 1
# fv.AEPg
name                : EPG_990
annotation          : orchestrator:msc
childAction         :
configIssues       :
configSt           : applied
descr              :
dn                  : uni/tn-TN_D/ap-App_Profile/epg-EPG_990
exceptionTag       :
extMngdBy          :
floodOnEncap       : disabled
fwdCtrl            :
hasMcastSource     : no
isAttrBasedEPg    : no
isSharedSrvMsiteEPg : no
lcOwn              : local
matchT             : AtleastOne
modTs              : 2021-09-18T08:26:49.906+00:00
monPolDn           : uni/tn-common/monepg-default
nameAlias          :
pcEnfPref          : unenforced
pctag              : 32770
prefGrMemb         : exclude
prio               : unspecified
rn                 : epg-EPG_990
scope              : 2850817
shutdown           : no
status             :
triggerSt          : triggerable
txId               : 1152921504609182523
uid                : 0

```

```

APIC1# moquery -c fvBD -f 'fv.BD.name=="BD_990"'
Total Objects shown: 1
# fv.BD
name : BD_990
OptimizeWanBandwidth : yes
annotation : orchestrator:misc
arpFlood : yes
bcastP : 225.0.56.224
childAction :
configIssues :
descr :
dn : uni/tn-TN_D/BD-BD_990
epClear : no
epMoveDetectMode :
extMngdBy :
hostBasedRouting : no
intersiteBumTrafficAllow : yes
intersiteL2Stretch : yes
ipLearning : yes
ipv6McastAllow : no
lcOwn : local
limitIpLearnToSubnets : yes
llAddr : ::
mac : 00:22:BD:F8:19:FF
mcastAllow : no
modTs : 2021-09-18T08:26:49.906+00:00
monPolDn : uni/tn-common/monepg-default
mtu : inherit
multiDstPktAct : bd-flood
nameAlias :
ownerKey :
ownerTag :
pcTag : 16387
rn : BD-BD_990
scope : 2850817
seg : 16580488
status :
type : regular
uid : 0
unicastRoute : yes
unkMacUcastAct : proxy
unkMcastAct : flood
v6unkMcastAct : flood
vmac : not-applicable
: 0

```

```

APIC1# moquery -c fvCtx -f 'fv.Ctx.name=="VRF_Stretch"'
Total Objects shown: 1
# fv.Ctx
name : VRF_Stretch
annotation : orchestrator:misc
bdEnforcedEnable : no
childAction :
descr :
dn : uni/tn-TN_D/ctx-VRF_Stretch
extMngdBy :
ipDataPlaneLearning : enabled
knwMcastAct : permit
lcOwn : local
modTs : 2021-09-18T08:26:58.185+00:00

```

```

monPolDn          : uni/tn-common/monepg-default
nameAlias         :
ownerKey          :
ownerTag          :
pcEnfDir          : ingress
pcEnfDirUpdated  : yes
pcEnfPref        : enforced
pcTag             : 16386
rn                : ctx-VRF_Stretch
scope             : 2850817
seg               : 2850817
status            :
uid               : 0

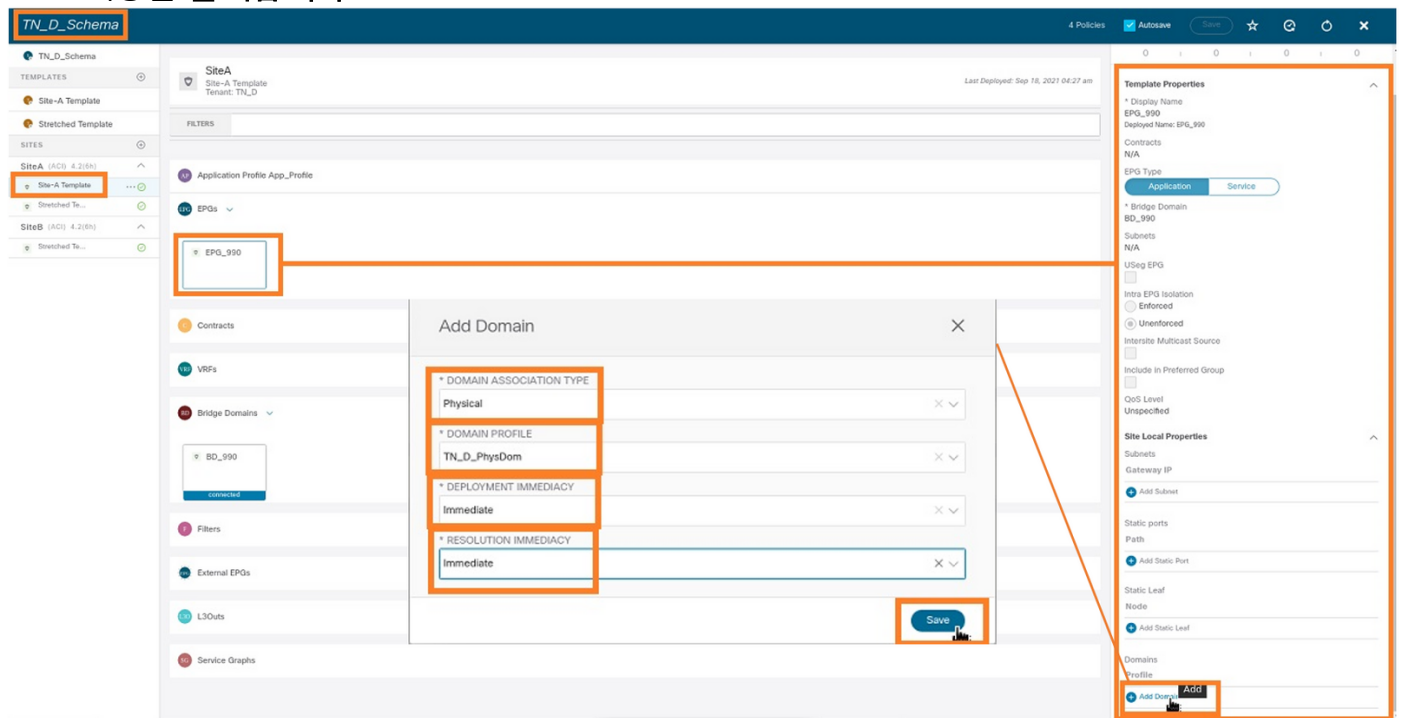
```

고정 포트 바인딩 구성

이제 EPG "EPG_990"에서 고정 포트 바인딩을 구성하고 VRF HOST_A를 사용하여 N9K를 구성할 수 있습니다(기본적으로 HOST_A를 시뮬레이션함). ACI 측 고정 포트 바인딩 컨피그레이션이 먼저 완료됩니다.

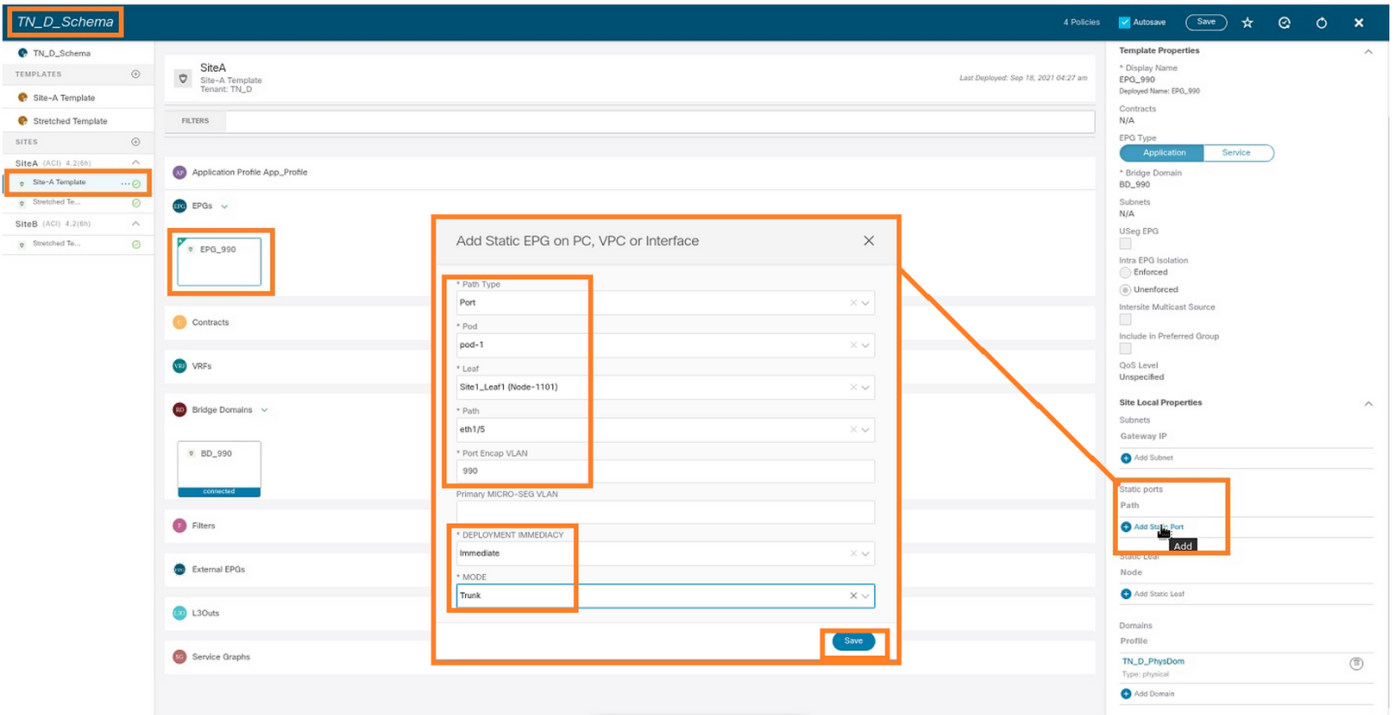
1단계. EPG_990에 물리적 도메인을 추가합니다.

1. 생성한 스키마에서 Site-A Template(사이트-A 템플릿) > EPG_990을 선택합니다.
2. 템플릿 속성 상자에서 도메인 추가를 클릭합니다.
3. Add Domain 대화 상자의 드롭다운 목록에서 다음 옵션을 선택합니다. 도메인 연결 유형 - 물리적도메인 프로파일 - TN_D_PhysDom구축 즉시 - 즉시즉각적 해결 - 즉각적
4. 저장을 클릭합니다.

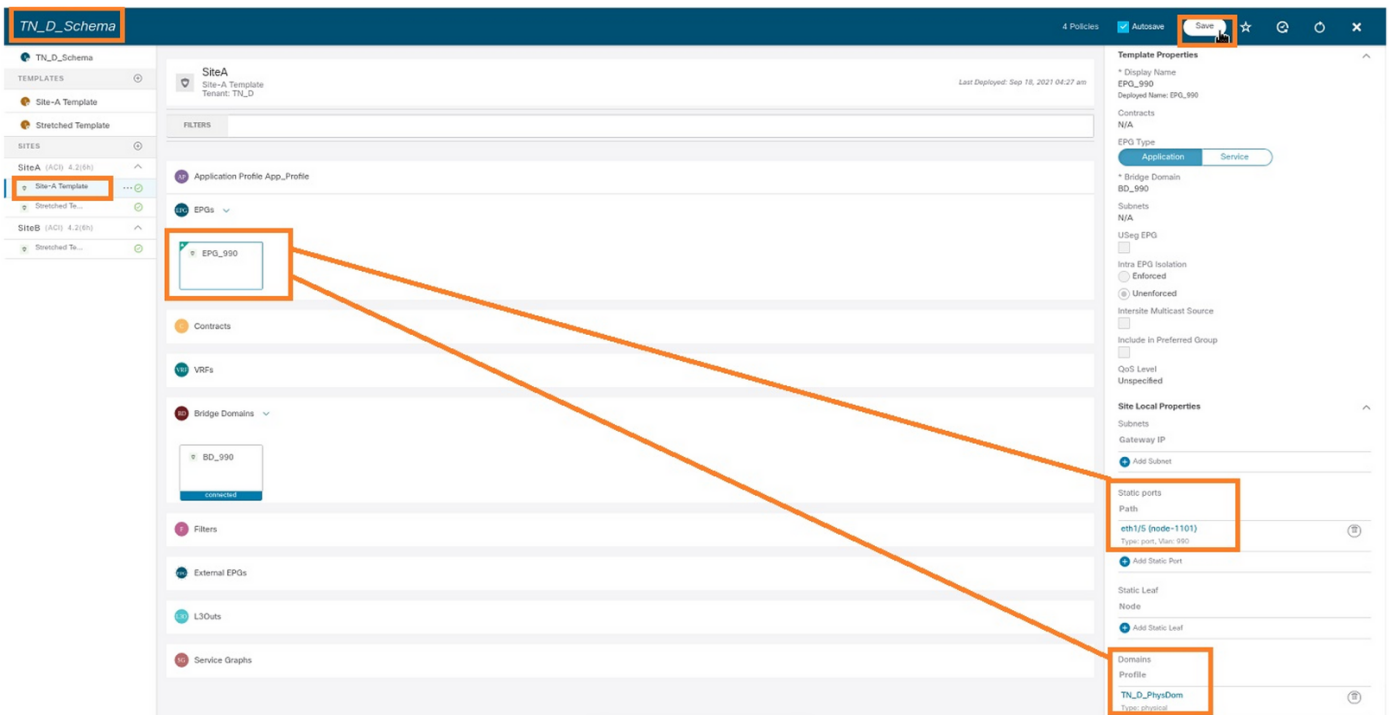


2단계. 고정 포트(Site1_Leaf1 eth1/5)를 추가합니다.

1. 생성한 스키마에서 Site-A Template(사이트-A 템플릿) > EPG_990을 선택합니다.
2. Template Properties(템플릿 속성) 상자에서 Add Static Port(고정 포트 추가)를 클릭합니다.
3. Add Static EPG on PC, VPC or Interface 대화 상자에서 Node-101 eth1/5를 선택하고 VLAN 990을 할당합니다.



3단계. 고정 포트 및 물리적 도메인이 EPG_990 아래에 추가되었는지 확인합니다.



다음 명령을 사용하여 고정 경로 바인딩을 확인합니다.

```

APIC1# moquery -c fvStPathAtt -f 'fv.StPathAtt.pathName=="eth1/5"' | grep EPG_990 -A 10 -B 5
# fv.StPathAtt
pathName      : eth1/5
childAction   :
descr         :
dn            : uni/epp/fv-[uni/tn-TN_D/ap-App_Profile/epg-EPG_990]/node-1101/stpathatt-[eth1/5]
lcOwn         : local
modTs         : 2021-09-19T06:16:46.226+00:00
monPolDn      : uni/tn-common/monepg-default
name          :
  
```

```

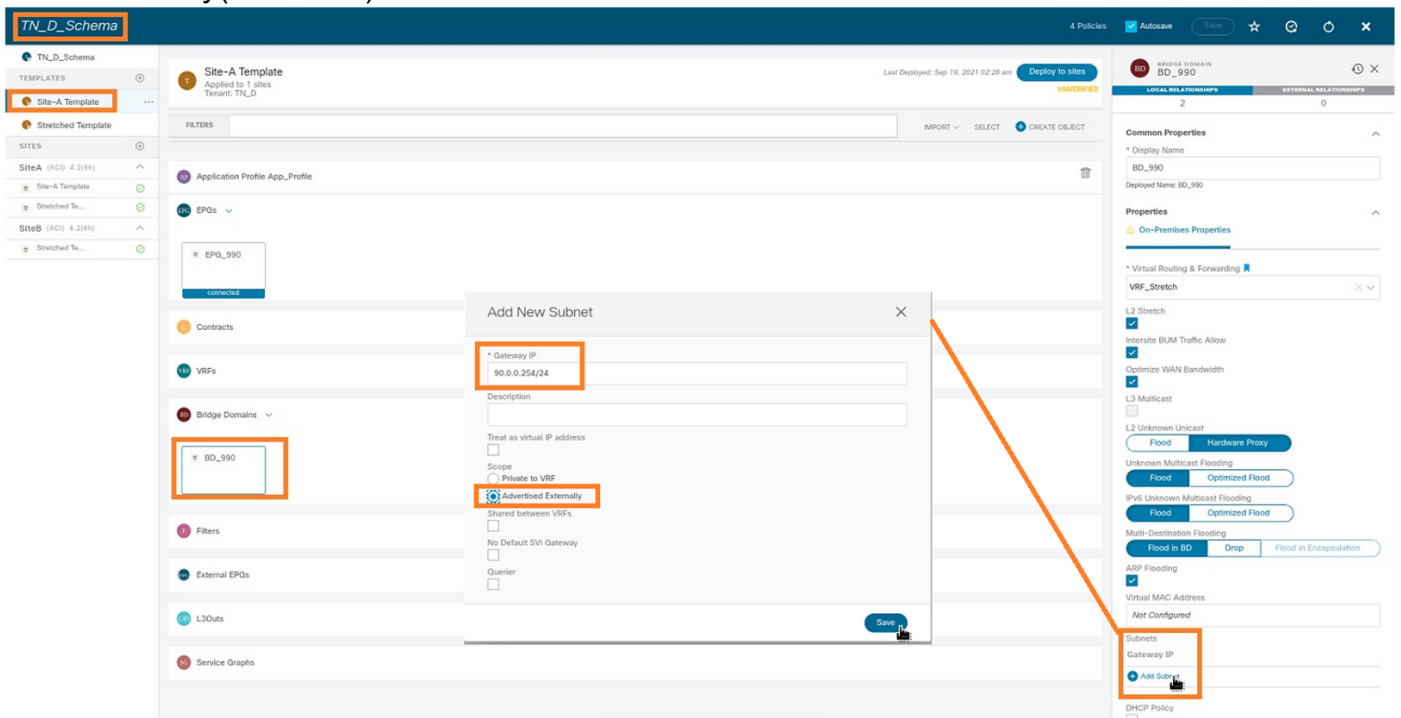
nameAlias      :
ownerKey       :
ownerTag       :
rn             : stpathatt-[eth1/5]
status         :

```

BD 구성

1단계. BD 아래에 서브넷/IP를 추가합니다(HOST_A는 BD IP를 게이트웨이로 사용).

1. 생성한 스키마에서 Site-A Template(사이트-A 템플릿) > BD_990을 선택합니다.
2. 서브넷 추가를 클릭합니다.
3. Add New Subnet(새 서브넷 추가) 대화 상자에서 Gateway IP 주소를 입력하고 Advertised Externally(외부 알림) 라디오 버튼을 클릭합니다.



2단계. 이 명령을 사용하여 서브넷이 APIC1 Site-A에 추가되었는지 확인합니다.

```

APIC1# moquery -c fvSubnet -f 'fv.Subnet.ip=="90.0.0.254/24"'
Total Objects shown: 1

```

```

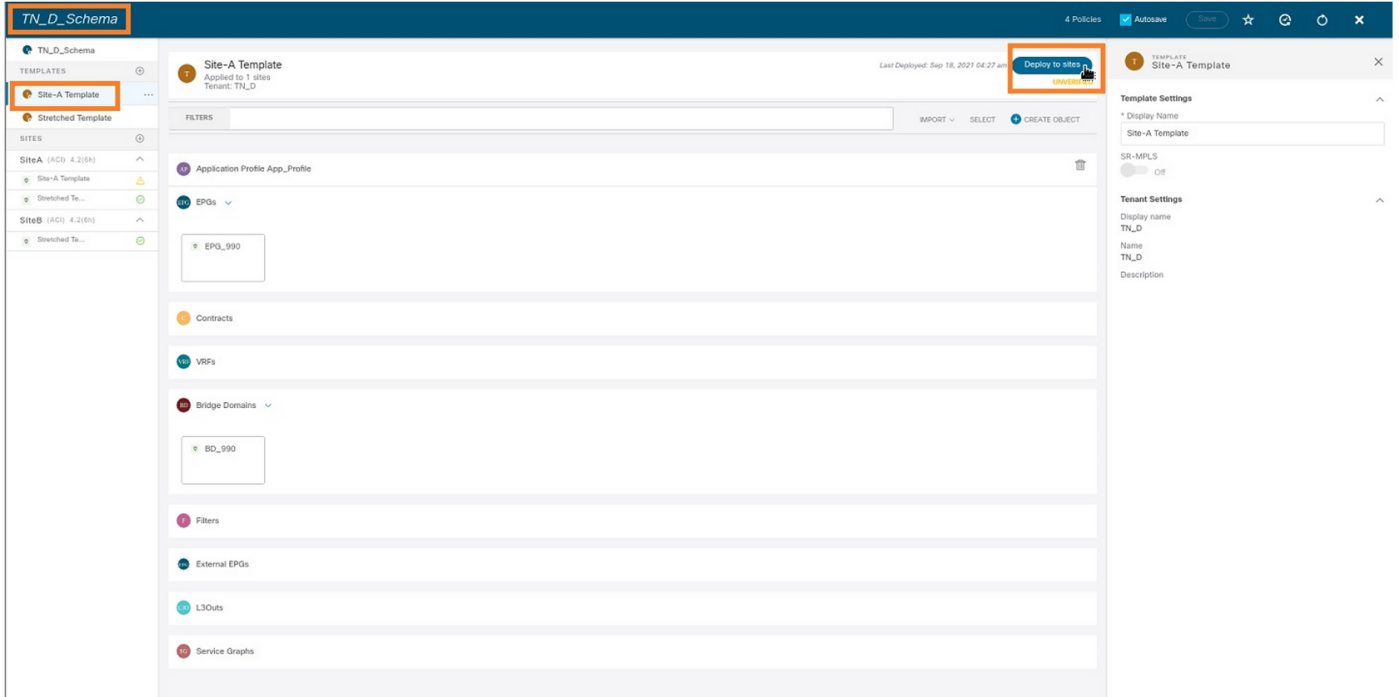
# fv.Subnet
ip           : 90.0.0.254/24
annotation   : orchestrator:misc
childAction  :
ctrl         : nd
descr       :
dn          : uni/tn-TN_D/BD-BD_990/subnet-[90.0.0.254/24]
extMngdBy   :
lcOwn       : local
modTs       : 2021-09-19T06:33:19.943+00:00
monPolDn    : uni/tn-common/monepg-default
name        :
nameAlias   :
preferred   : no
rn          : subnet-[90.0.0.254/24]
scope       : public

```

status :
uid : 0
virtual : no

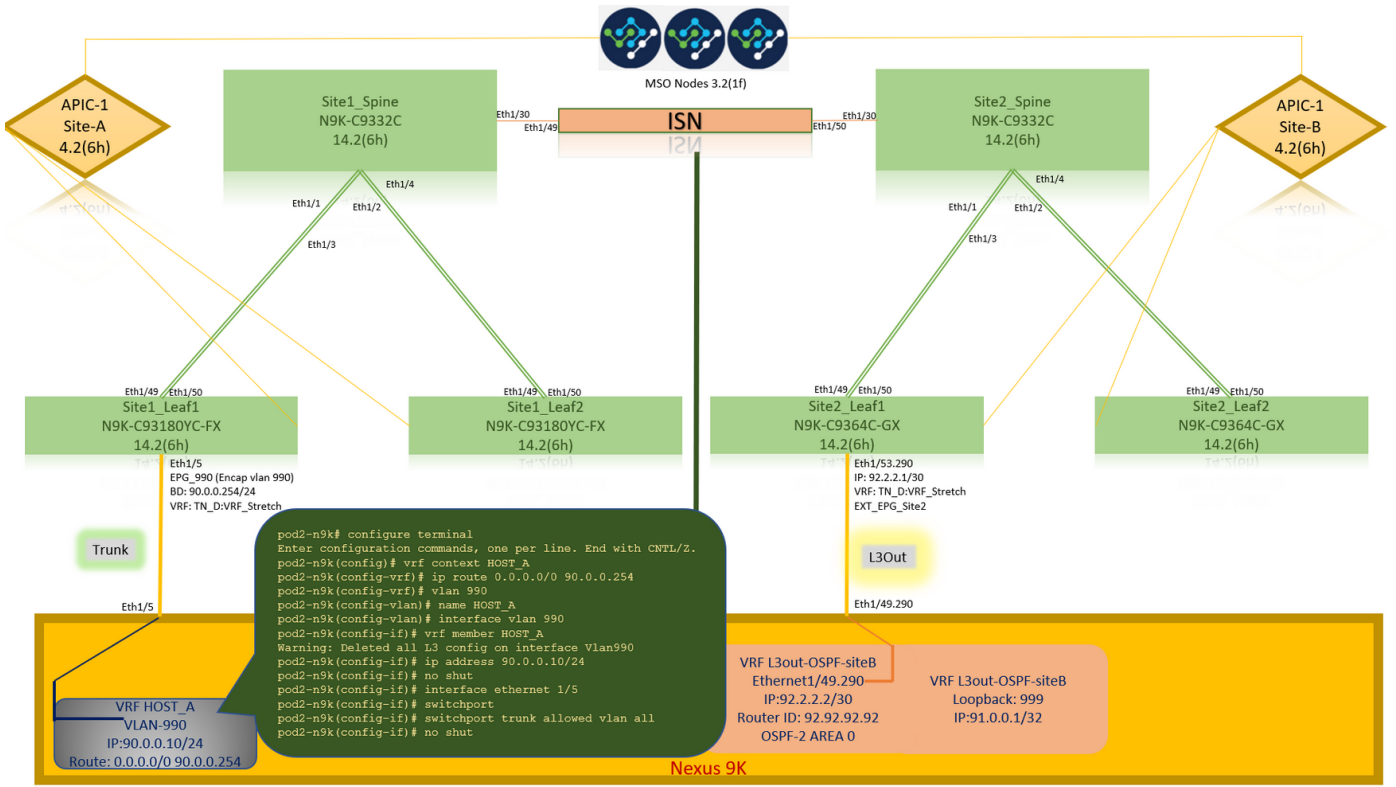
3단계. Site-A 템플릿을 구축합니다.

1. 생성한 스키마에서 **Site-A** 템플릿을 선택합니다.
2. **Deploy to sites**를 클릭합니다.



호스트-A(N9K) 구성

VRF HOST_A를 사용하여 N9K 디바이스를 구성합니다. N9K 컨피그레이션이 완료되면 이제 ICMP(ping)를 통해 ACI Leaf BD 애니캐스트 주소(HOST_A의 게이트웨이)에 연결할 수 있음을 확인할 수 있습니다.



ACI operational(ACI 운영) 탭에서는 90.0.0.10(HOST_A IP 주소)가 학습된 것을 확인할 수 있습니다.

APIC [SiteA]

System | Tenants | Fabric | Virtual Networking | L4-L7 Services | Admin | Operations | Apps | Integrations

ALL TENANTS | Add Tenant | Tenant Search: name or descr | common | TN_D | mgmt | infra | dosykes-in1

This has been created from Multi-Site. It is recommended to only make changes from Multi-Site. Please review the documentation before making any changes here.

TN_D

EPG - EPG_990

Summary | Policy | **Operational** | Stats | Health | Faults | History

Client End-Points

End Point	MAC	IP	Learning Source	Hosting Server	Reporting Controller Name	Interface	Multicast Address	Encap
EP-C0:14:FE5E:1...	C0:14:FE5E:14:07	90.0.0.10	learned	---	---	Pod-1/Node-1101/eth1/5 (learned)	---	vlan-990

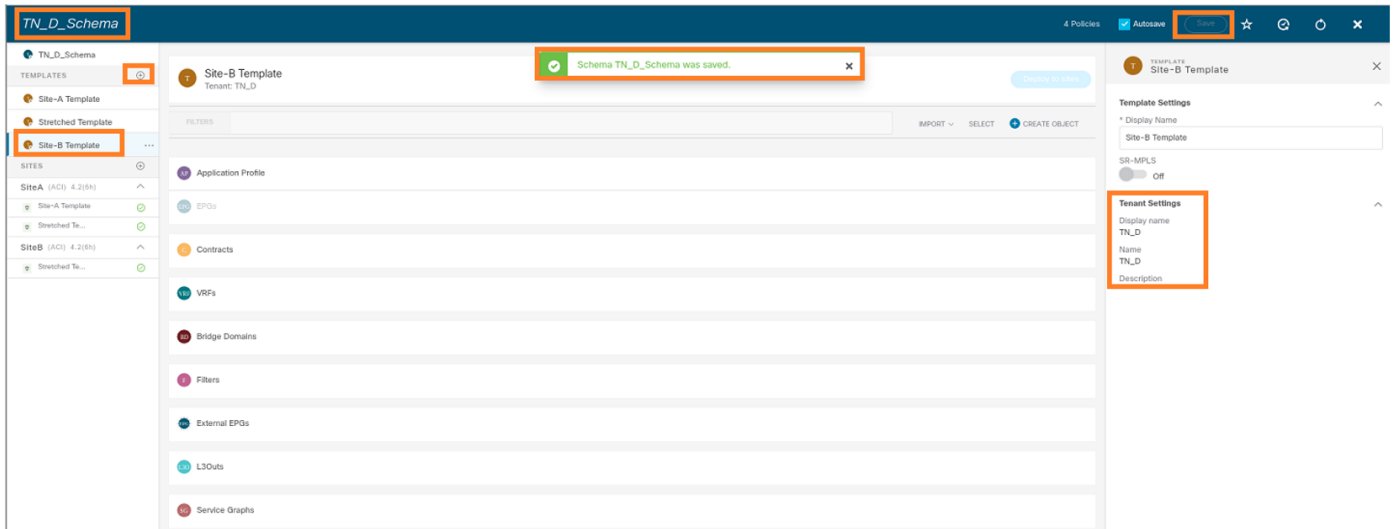
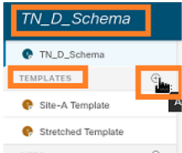
```

pod2-n9k# ping 90.0.0.254 vrf HOST_A
PING 90.0.0.254 (90.0.0.254): 56 data bytes
36 bytes from 90.0.0.10: Destination Host Unreachable
Request 0 timed out
64 bytes from 90.0.0.254: icmp_seq=1 ttl=63 time=0.902 ms
64 bytes from 90.0.0.254: icmp_seq=2 ttl=63 time=0.576 ms
64 bytes from 90.0.0.254: icmp_seq=3 ttl=63 time=0.708 ms
64 bytes from 90.0.0.254: icmp_seq=4 ttl=63 time=0.659 ms

--- 90.0.0.254 ping statistics ---
5 packets transmitted, 4 packets received, 20.00% packet loss
round-trip min/avg/max = 0.576/0.711/0.902 ms
pod2-n9k#
  
```

Site-B 템플릿 만들기

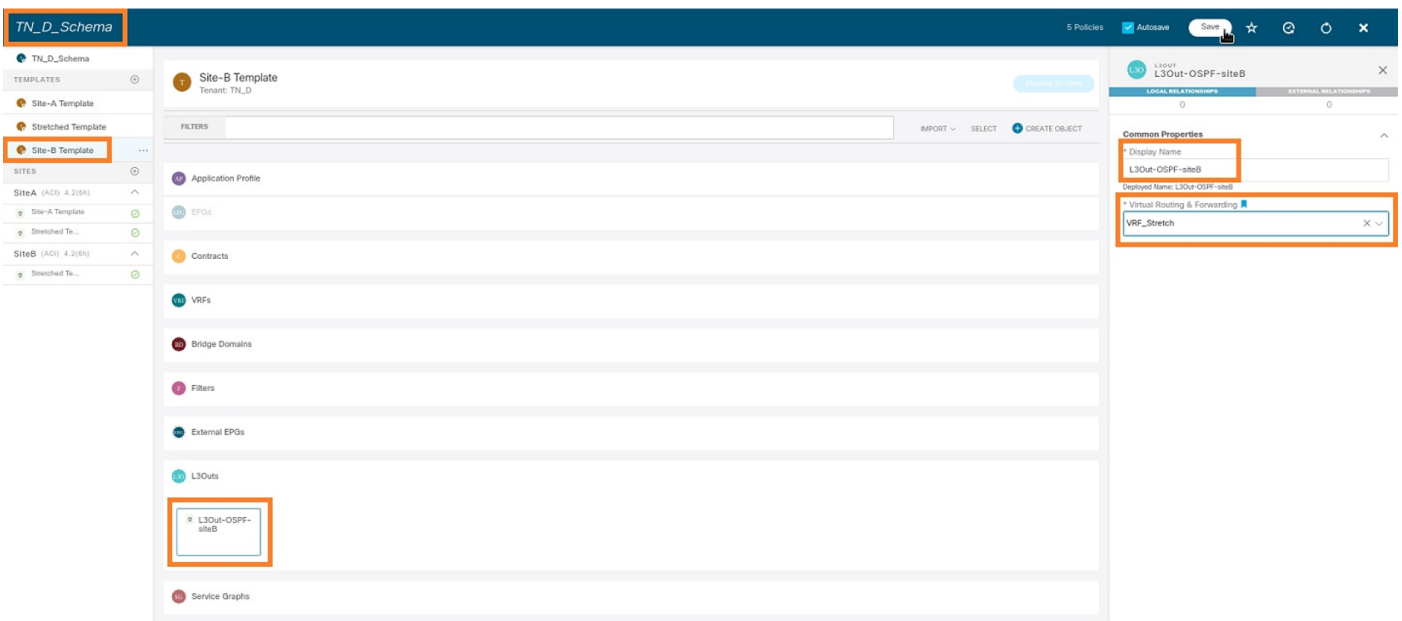
1단계. 생성한 스키마에서 TEMPLATES를 선택합니다. +기호를 클릭하고 Site-B 템플릿이라는 이름의 템플릿을 만듭니다.



사이트-B L3out 구성

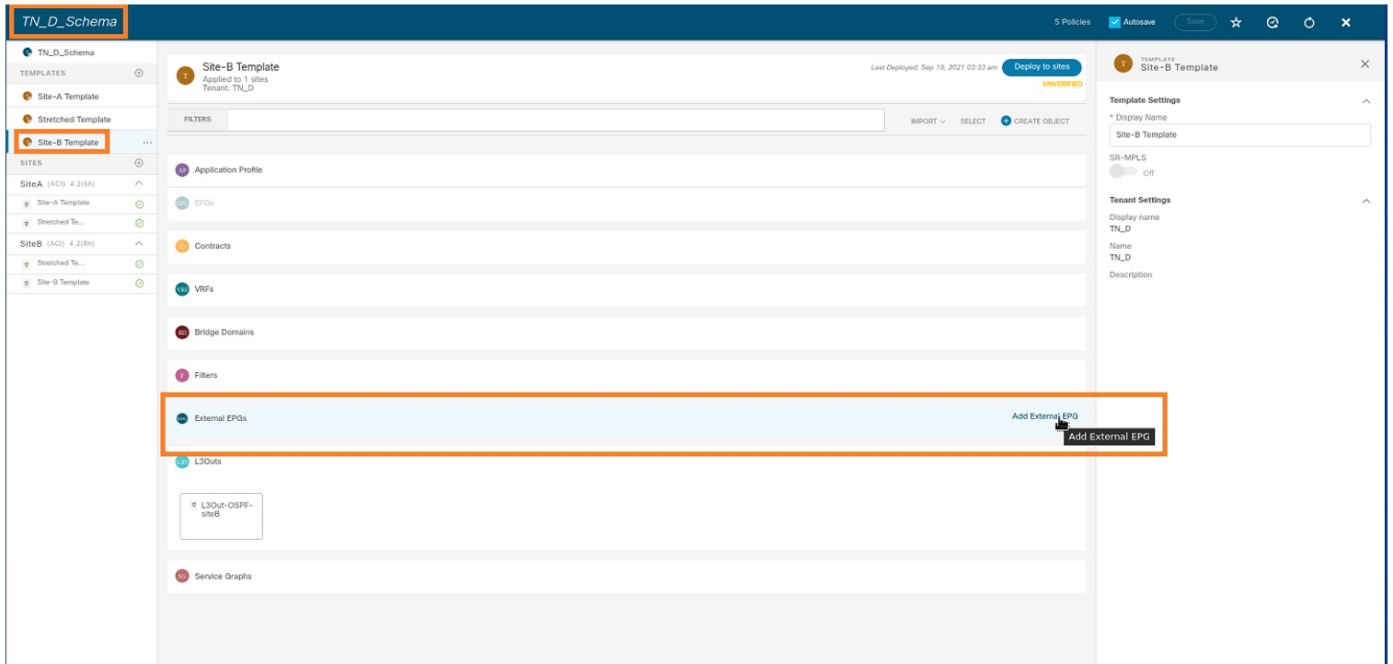
L3out을 생성하고 VRF_Stretch를 연결합니다. MSO에서 L3out 객체를 생성해야 하며 L3out 컨피그레이션의 나머지 부분은 APIC에서 수행해야 합니다(L3out 매개변수는 MSO에서 사용할 수 없음). 또한 MSO에서 외부 EPG를 생성합니다(Site-B 템플릿에서만 외부 EPG가 스트레치되지 않으므로).

1단계. 생성한 스키마에서 Site-B 템플릿을 선택합니다. Display Name(표시 이름) 필드에 L3out_OSPF_siteB를 입력합니다. Virtual Routing & Forwarding(가상 라우팅 및 포워딩) 드롭다운 목록에서 VRF_Stretch를 선택합니다.



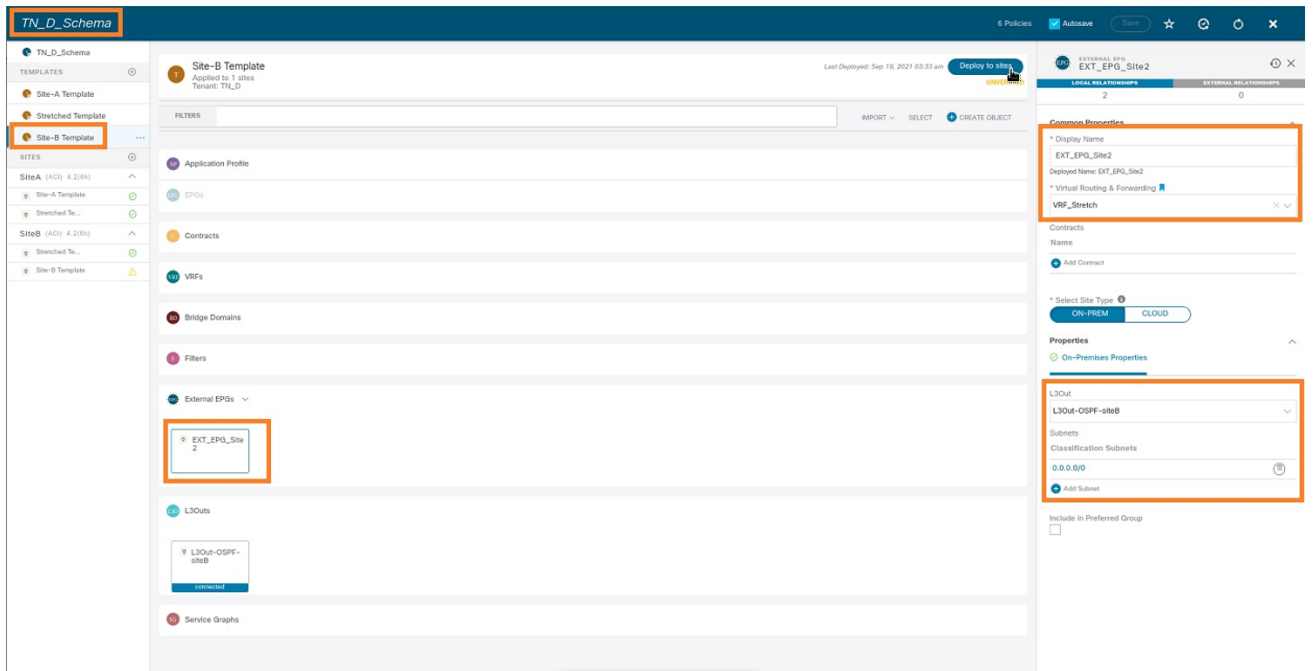
외부 EPG 생성

1단계. 생성한 스키마에서 Site-B 템플릿을 선택합니다. Add External EPG를 클릭합니다.



2단계. 외부 EPG를 사용하여 L3out을 연결합니다.

1. 생성한 스키마에서 Site-B 템플릿을 선택합니다.
2. Display Name(표시 이름) 필드에 EXT_EPG_Site2를 입력합니다.
3. Classification Subnets(분류 서브넷) 필드에 외부 EPG의 외부 서브넷에 0.0.0.0/0을 입력합니다.

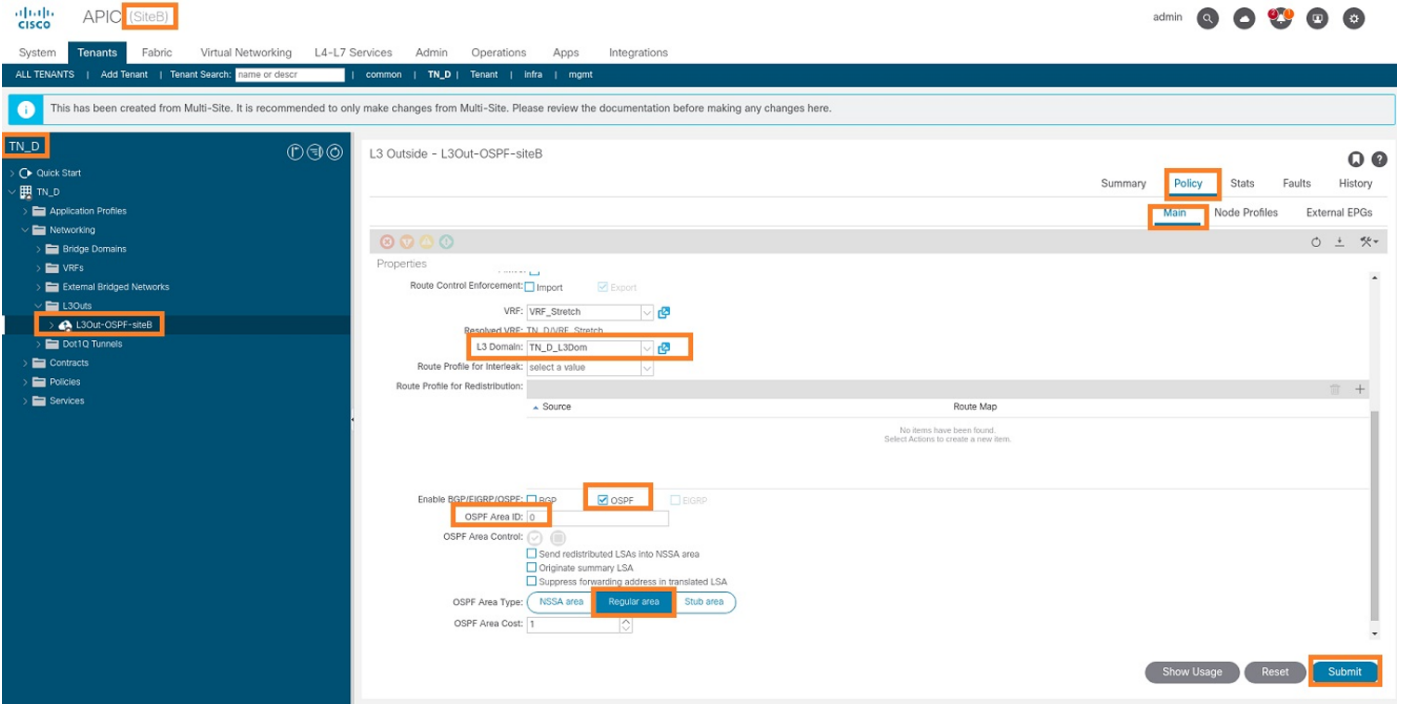


나머지 L3out 컨피그레이션은 APIC(Site-B)에서 완료됩니다.

3단계. L3 도메인을 추가하고, OSPF 프로토콜을 활성화하고, OSPF를 일반 영역 0으로 구성합니다

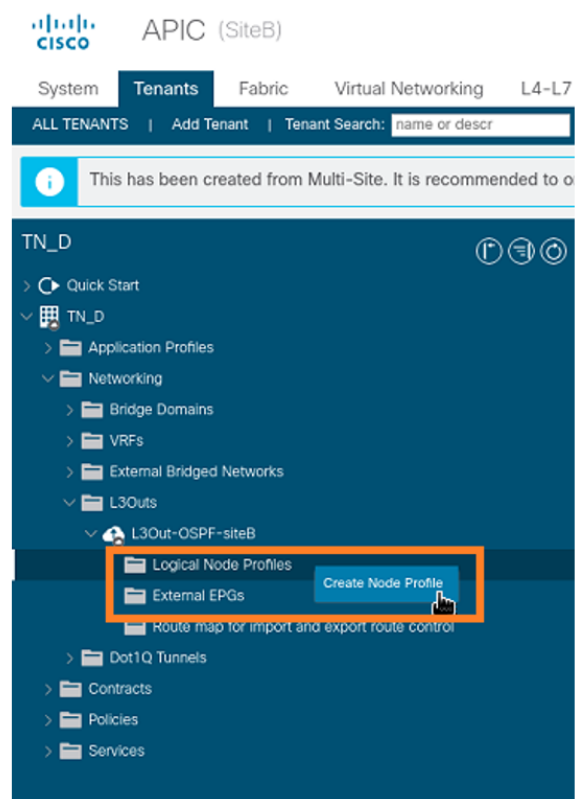
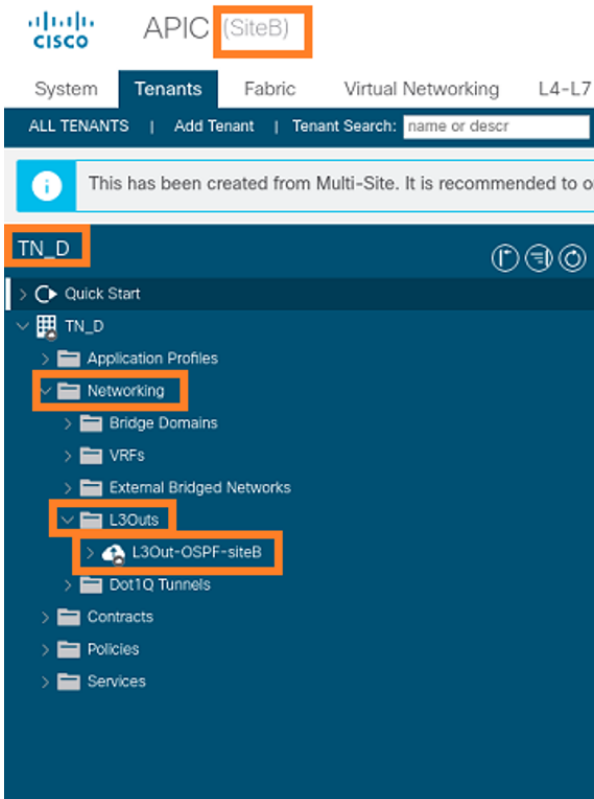
1. Site-B의 APIC-1에서 TN_D > Networking > L3out-OSPF-siteB > Policy > Main을 선택합니다.
2. L3 도메인 드롭다운 목록에서 TN_D_L3Dom을 선택합니다.
3. Enable BGP/EIGRP/OSPF에 대한 OSPF 확인란을 선택합니다.
4. OSPF Area ID 필드에 0을 입력합니다.
5. OSPF Area Type(OSPF 영역 유형)에서 Regular(일반) 영역을 선택합니다.

6. Submit(제출)을 클릭합니다.



4단계. 노드 프로파일을 생성합니다.

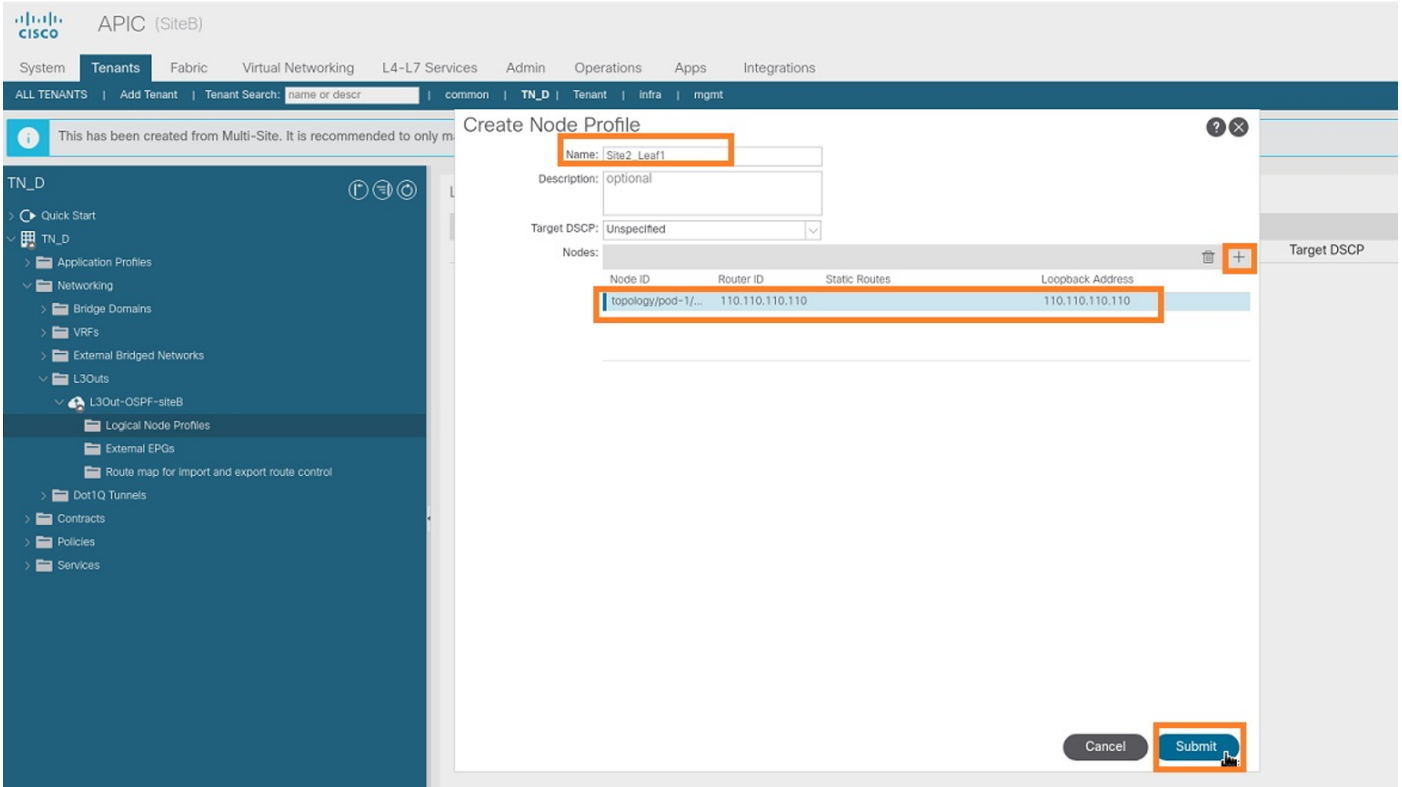
1. Site-B의 APIC-1에서 TN_D > Networking > L3Outs > L3Out-OSPF-siteB > Logical Node Profiles를 선택합니다.
2. Create Node Profile을 클릭합니다.



5단계. site-B의 노드로 Site2_Leaf1 스위치를 선택합니다.

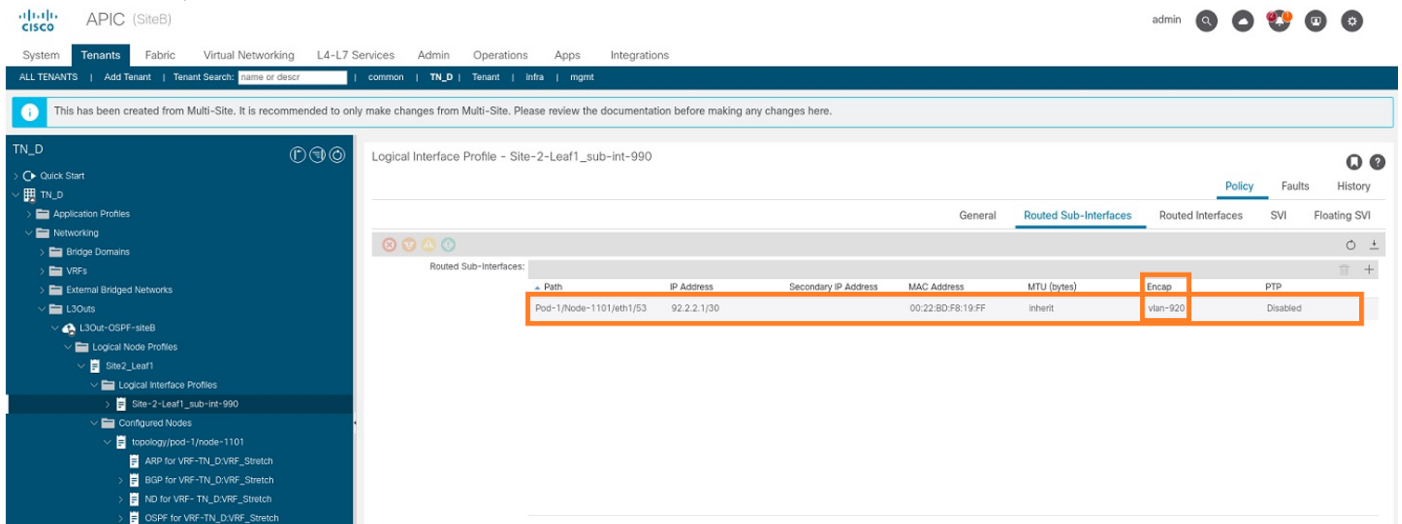
1. Site-B의 APIC-1에서 TN_D > Networking > L3Outs > L3Out-OSPF-siteB > Logical Node Profiles > Create Node Profile을 선택합니다.
2. Name(이름) 필드에 Site2_Leaf1을 입력합니다.

3. 노드를 추가하려면 + 기호를 클릭합니다.
4. 라우터 ID IP 주소로 pod-2 node-101을 추가합니다.



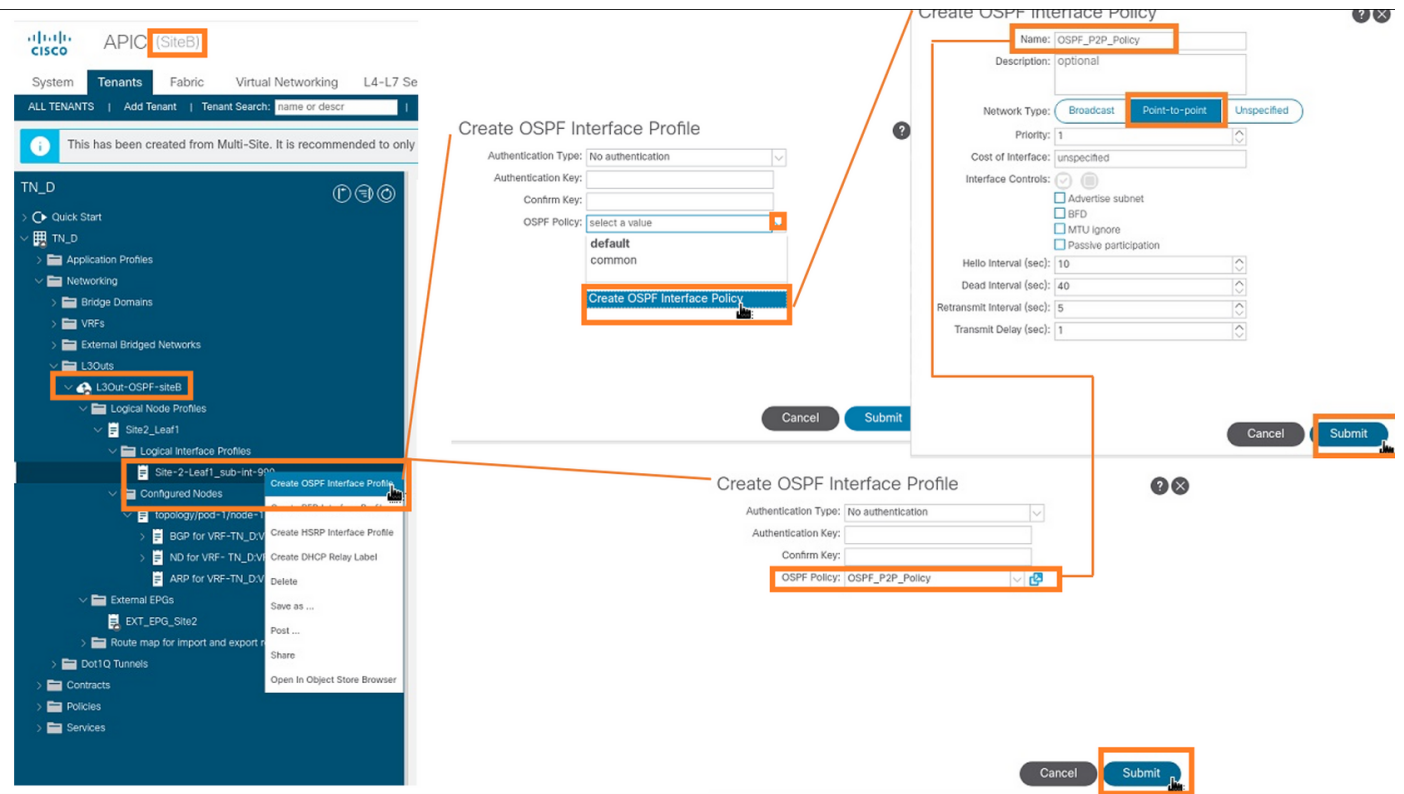
6단계. 인터페이스 프로파일을 추가합니다(외부 VLAN은 920(SVI 생성)).

1. Site-B의 APIC-1에서 TN_D > Networking > L3Outs > L3out-OSPF-SiteB > Logical Interface Profiles를 선택합니다.
2. 마우스 오른쪽 버튼을 클릭하고 인터페이스 프로파일을 추가합니다.
3. Routed Sub-Interfaces를 선택합니다.
4. IP 주소, MTU 및 VLAN-920을 구성합니다.

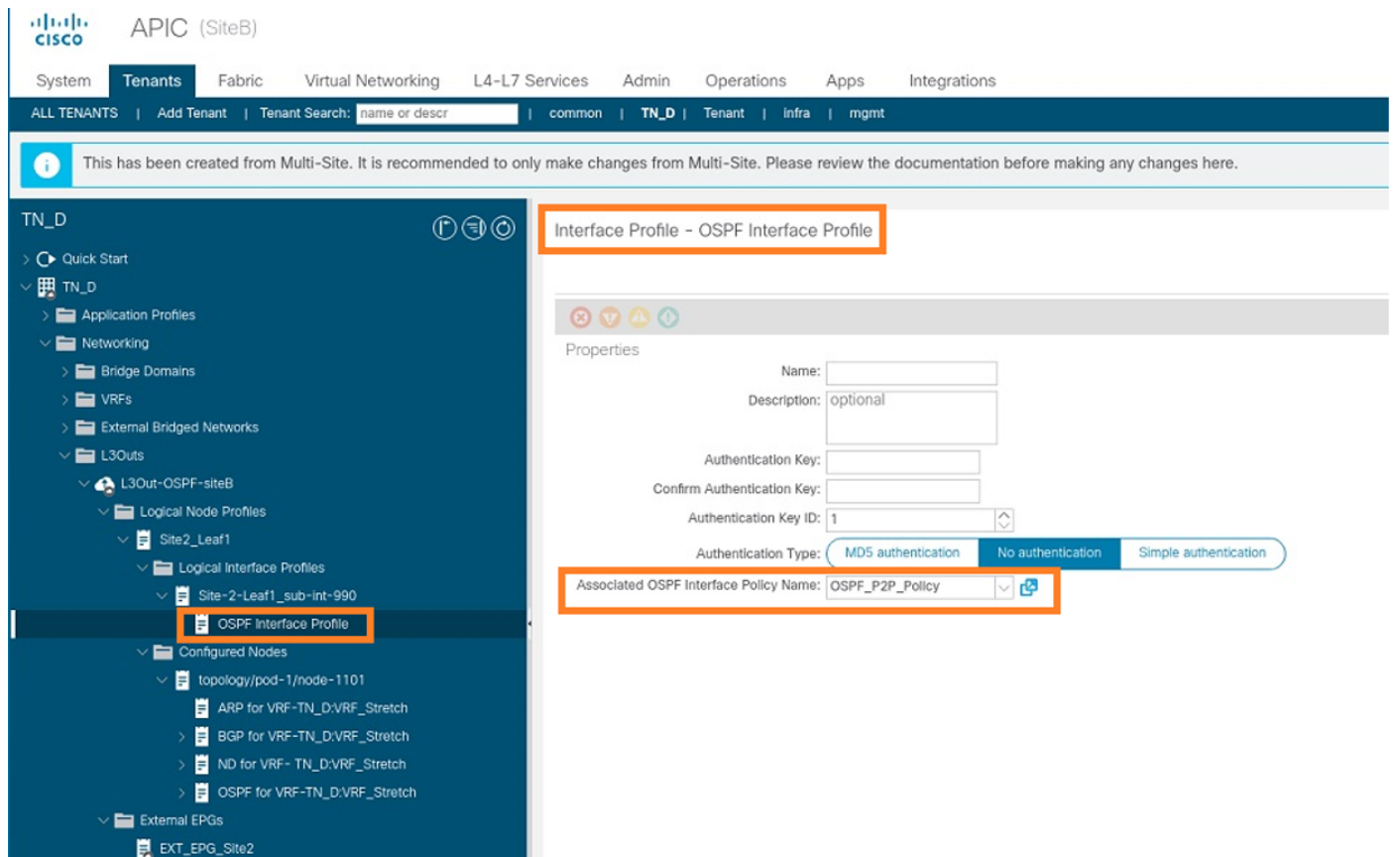


7단계. OSPF 정책(Point to Point Network)을 생성합니다.

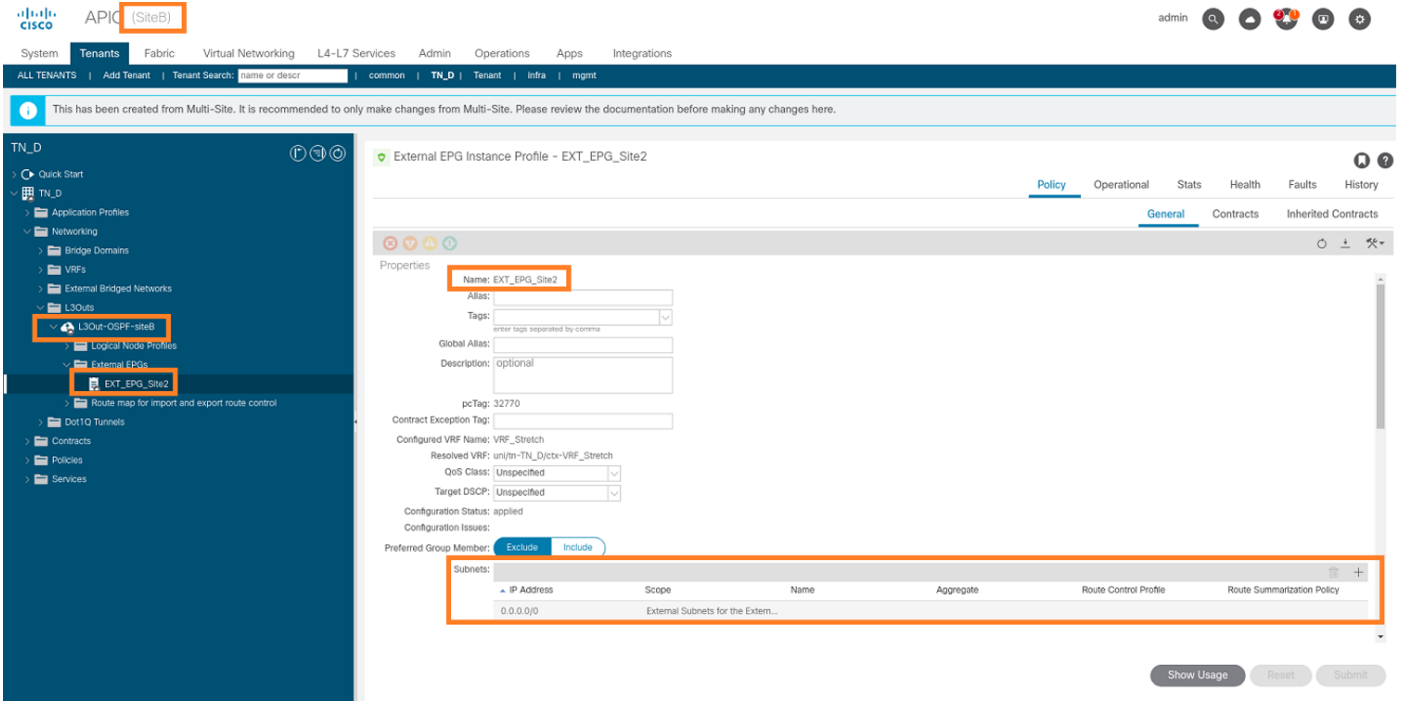
1. Site-B의 APIC-1에서 TN_D > Networking > L3Outs > L3Out-OSPF-siteB > Logical Interface Profiles를 선택합니다.
2. 마우스 오른쪽 버튼을 클릭하고 Create OSPF Interface Profile을 선택합니다.
3. 스크린샷에 표시된 옵션을 선택하고 Submit(제출)을 클릭합니다.



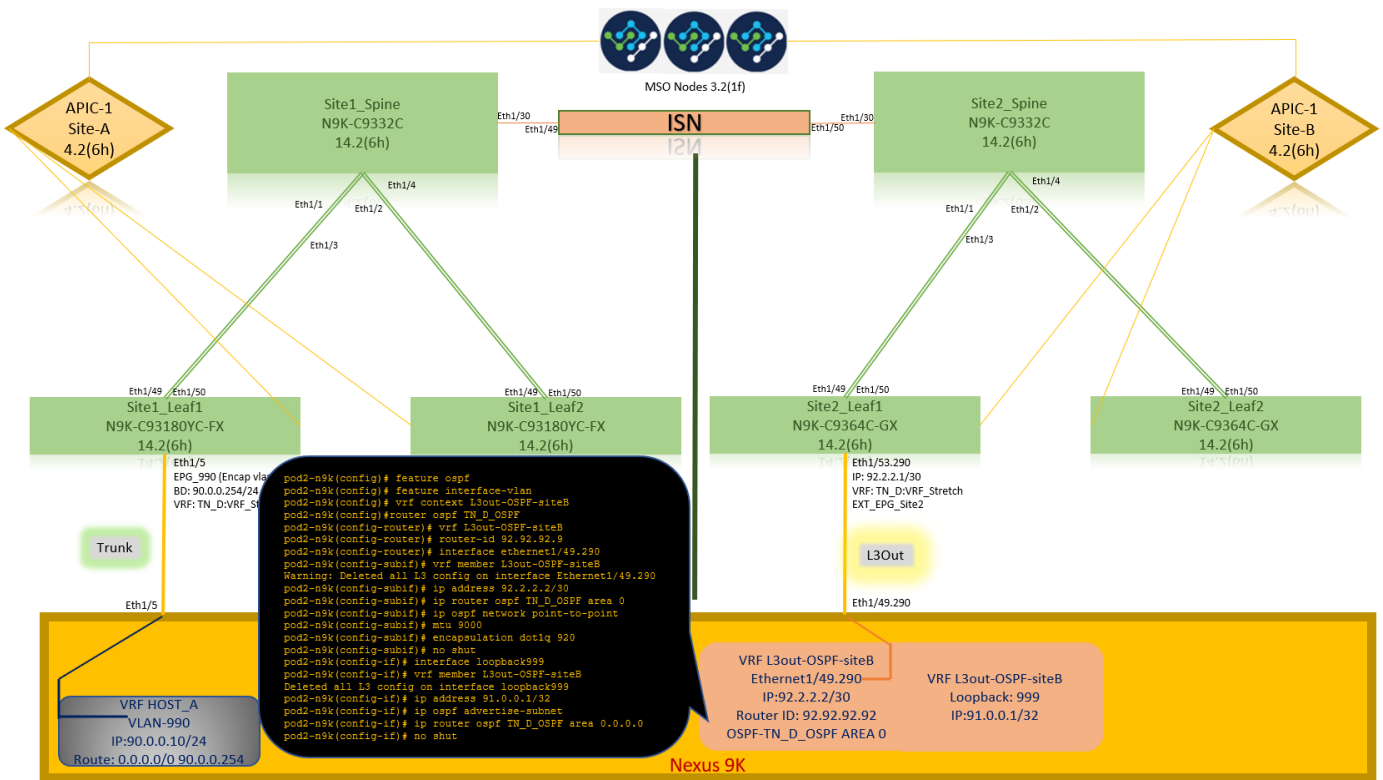
8단계. TN_D > Networking > L3Outs > L3Out-OSPF-siteB > Logical Interface Profiles > (interface profile) > OSPF Interface Profile 아래에 연결된 OSPF 인터페이스 프로파일 정책을 확인합니다.



9단계. 외부 EPG "EXT_EPG_Site2"가 MSO에 의해 생성되었는지 확인합니다. Site-B의 APIC-1에서 TN_D > L3Outs > L3Out-OSPF-siteB > External EPGs > EXT_EPG_Site2를 선택합니다.



외부 N9K(Site-B) 구성



N9K 컨피그레이션(VRF L3out-OSPF-siteB) 이후 N9K와 ACI Leaf(Site-B) 간에 OSPF 인접 디바이스가 설정되었음을 확인할 수 있습니다.

OSPF 인접 디바이스가 설정되고 UP(Full State)인지 확인합니다.

Site-B의 APIC-1에서 TN_D > Networking > L3Outs > L3Out-OSPF-siteB > Logical Node Profiles > Logical Interface Profiles > Configured Nodes > topology/pod01/node-11101 > OSPF for VRF-TN_RF_RF_Switch > Full ID > Neighbor state를 선택합니다.

The screenshot displays the Cisco APIC configuration page for OSPF - TN_D_VRF_Stretch. The left sidebar shows the navigation tree with 'L3Out-OSPF-siteB' and 'OSPF for VRF-TN_D_VRF_Stretch' highlighted. The main content area shows the configuration details for the OSPF process, including properties and a neighbors table.

Neighbor ID	State	Peer IP	Interface
92.92.92.92	Full	92.2.2.2	eth1/53.25

Below the neighbors table, there is a table for 'Inter Protocol Route Leak Into OSPF':

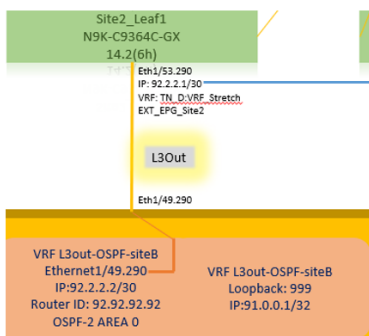
Name	Redistribution Protocol	Route Map	Scope
TN_D_VRF_Stretch	BGP	exp-ctx-prot0-2686978	Inter-protocol lea
TN_D_VRF_Stretch	ODOP	exp-ctx-st-2686978	Inter-protocol lea
TN_D_VRF_Stretch	Direct	exp-ctx-st-2686978	Inter-protocol lea
TN_D_VRF_Stretch	EIGRP	exp-ctx-prot0-2686978	Inter-protocol lea
TN_D_VRF_Stretch	Static	exp-ctx-st-2686978	Inter-protocol lea

The diagram on the right shows the physical network topology with Site2_Leaf1 (N9K-C9364C-GX) and L3Out. It also shows the VRF L3out-OSPF-siteB configuration with Ethernet1/49.290 and Loopback: 999.

N9K에서 OSPF 인접 디바이스를 확인할 수도 있습니다. 또한 ACI Leaf IP(Site-B)에 ping을 수행할 수 있습니다.

```
pod2-n9k(config-if)# ping 92.2.2.1 vrf L3out-OSPF-siteB
PING 92.2.2.1 (92.2.2.1): 56 data bytes
64 bytes from 92.2.2.1: icmp_seq=0 ttl=63 time=0.734 ms
64 bytes from 92.2.2.1: icmp_seq=1 ttl=63 time=0.591 ms
64 bytes from 92.2.2.1: icmp_seq=2 ttl=63 time=0.631 ms
64 bytes from 92.2.2.1: icmp_seq=3 ttl=63 time=0.588 ms
64 bytes from 92.2.2.1: icmp_seq=4 ttl=63 time=0.654 ms

--- 92.2.2.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.588/0.639/0.734 ms
```

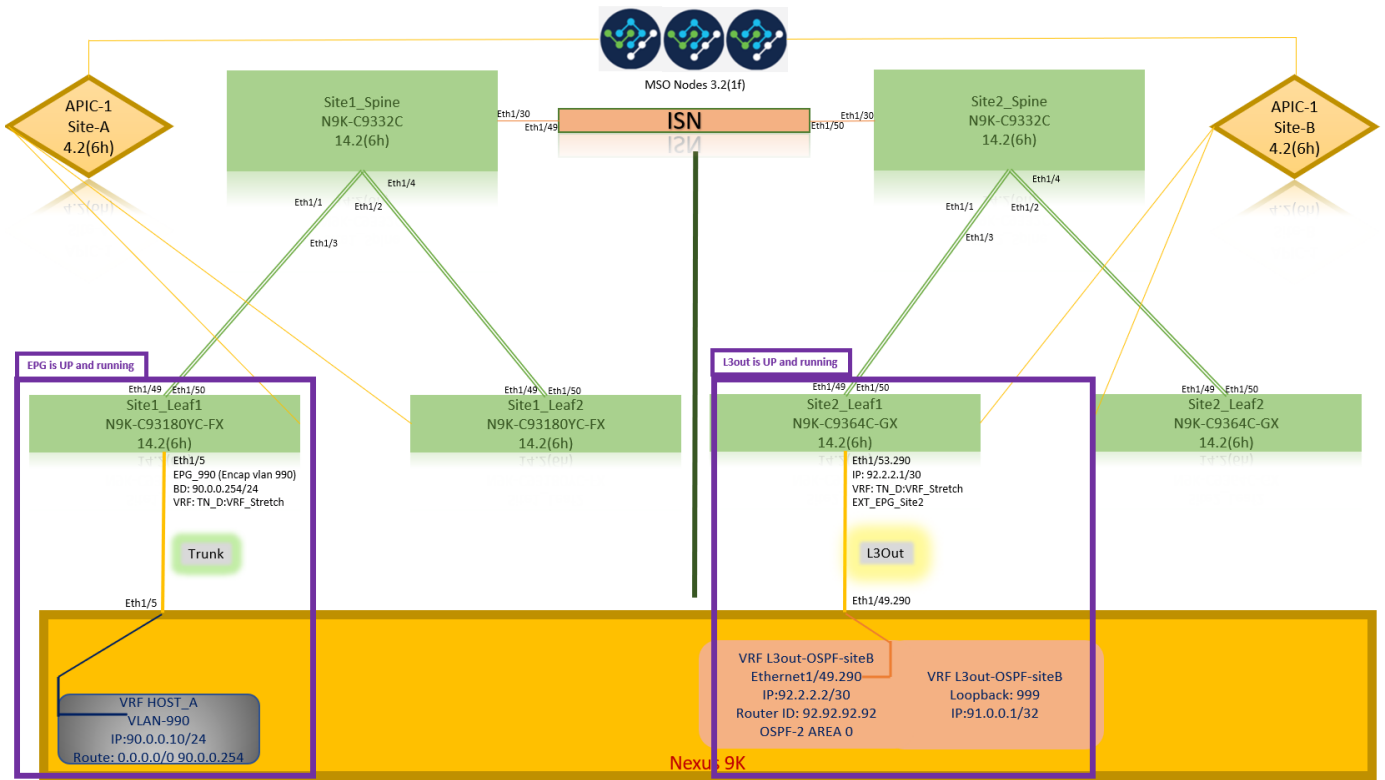


```
pod2-n9k(config-if)# show ip ospf neighbors vrf L3out-OSPF-siteB
OSPF Process ID TN_D_OSPF VRF L3out-OSPF-siteB
Total number of neighbors: 1
Neighbor ID Pri State Up Time Address Interface
110.110.110.110 1 FULL/ - 00:06:47 92.2.2.1 Eth1/49.290

pod2-n9k(config-if)# show ip route vrf L3out-OSPF-siteB
IP Route Table for VRF "L3out-OSPF-siteB"
'*' denotes best ucast next-hop
'***' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

92.2.2.0/30, ubest/mbest: 1/0, attached
*via 92.2.2.2, Eth1/49.290, [0/0], 00:19:38, direct
92.2.2.2/32, ubest/mbest: 1/0, attached
*via 92.2.2.2, Eth1/49.290, [0/0], 00:19:38, local
110.110.110.110/32, ubest/mbest: 1/0
*via 92.2.2.1, Eth1/49.290, [110/2], 00:06:48, ospf-TN_D_OSPF, intra
```

이 시점에서 사이트-A 및 사이트-B의 L3out 컨피그레이션의 Host_A 컨피그레이션이 완료되었습니다.

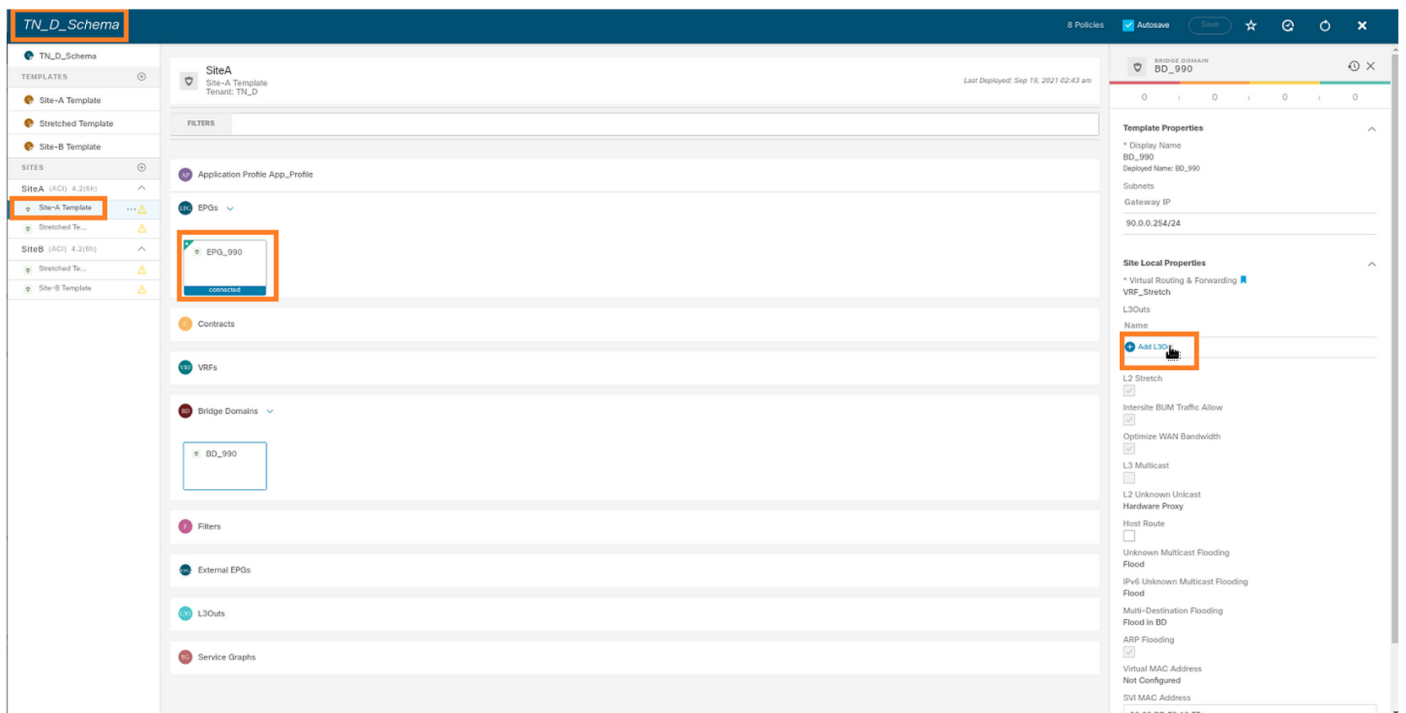


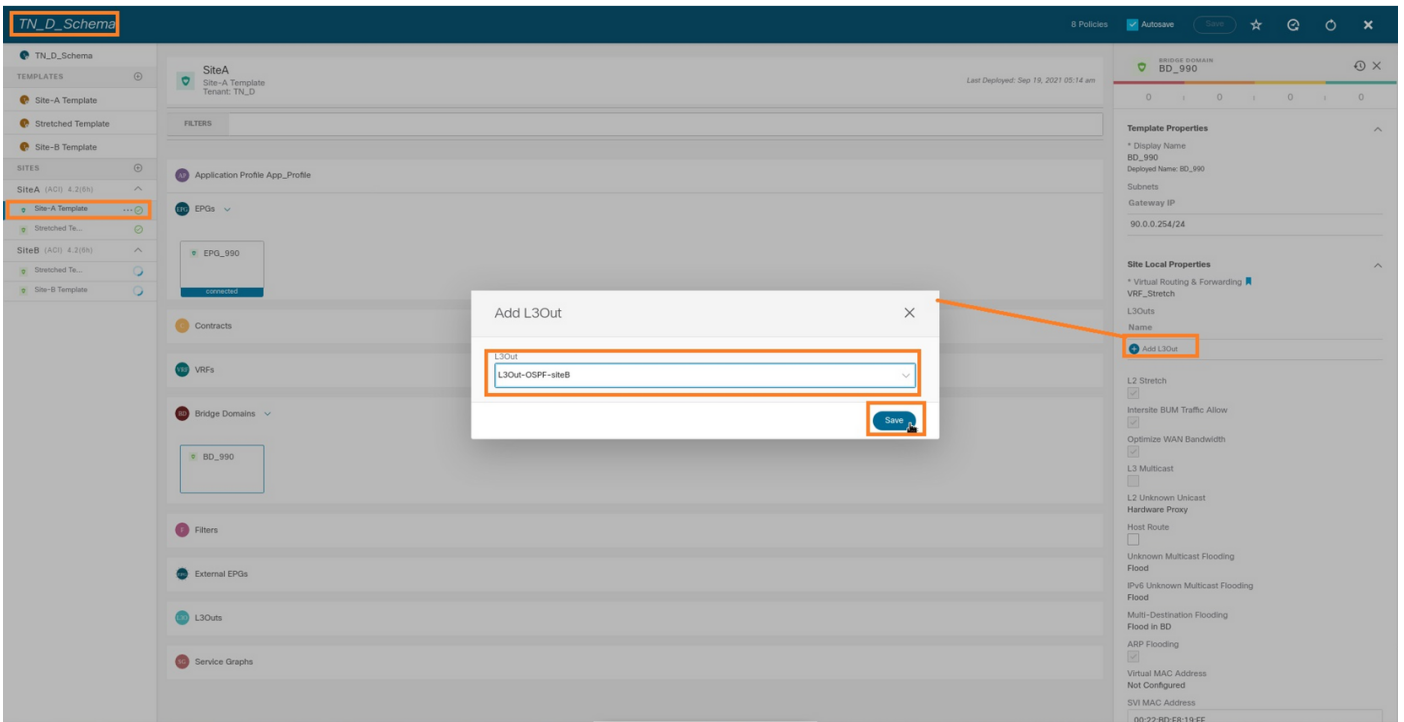
사이트-B L3out을 사이트-A EPG에 연결(BD)

그런 다음 MSO에서 Site-B L3out을 Site-A BD-990에 연결할 수 있습니다. 왼쪽 열에는 두 개의 섹션이 있습니다. 1) 템플릿 및 2) 사이트

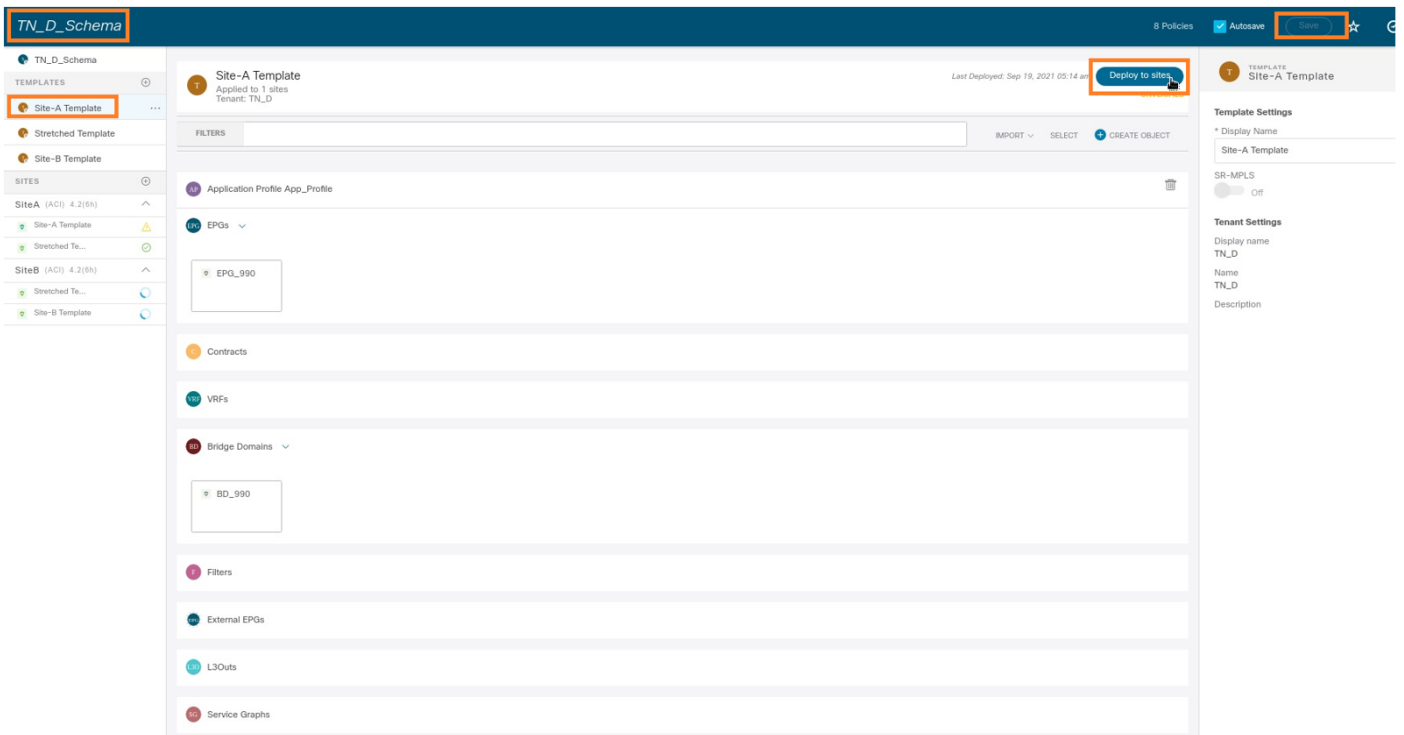
1단계. 두 번째 사이트 섹션에서 각 사이트에 첨부된 템플릿을 볼 수 있습니다. L3out을 "Site-A Template(사이트-A 템플릿)"에 연결하면 기본적으로 Sites(사이트) 섹션 내에 이미 연결된 템플릿에서 연결됩니다.

그러나 템플릿을 배포할 때 Templates(템플릿) > Site-A Template(사이트-A 템플릿) 섹션에서 배포하고 사이트에 저장/배포를 선택합니다.





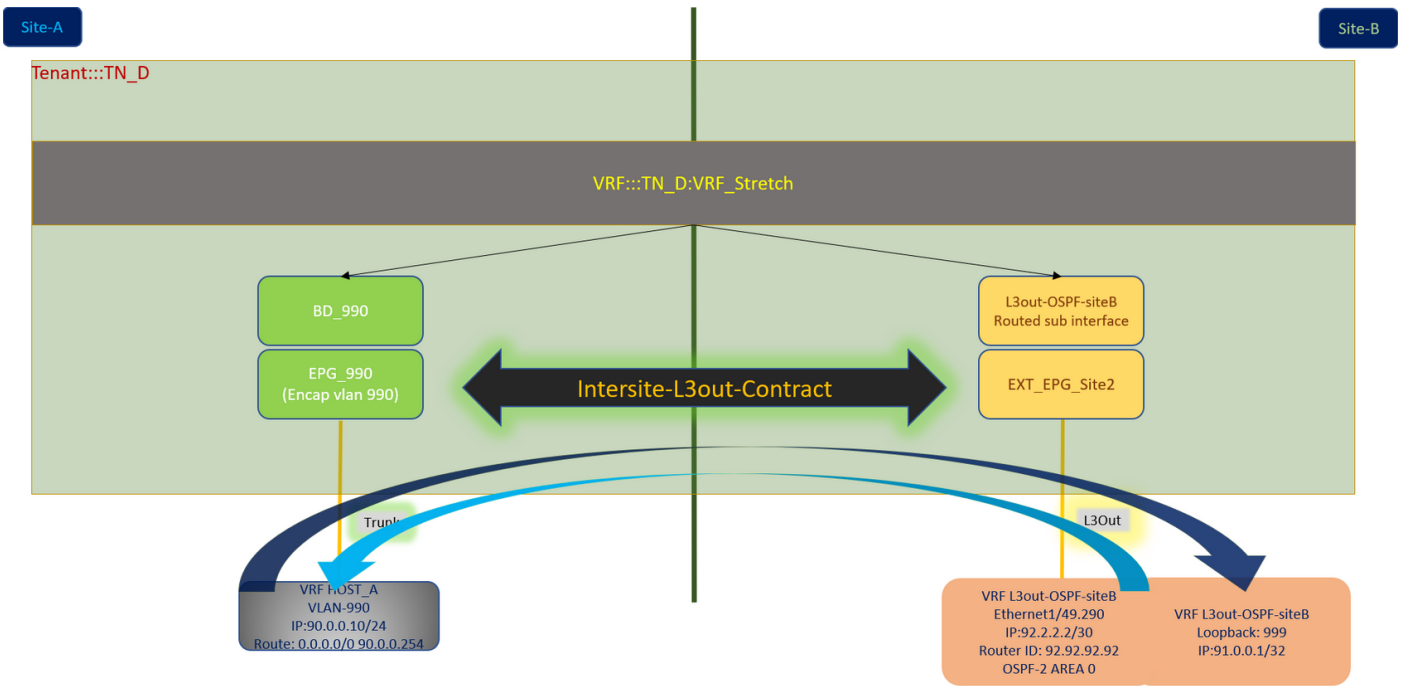
2단계. 기본 템플릿 "Site-A Template"의 첫 번째 섹션 "Templates"에서 구축합니다.



계약 구성

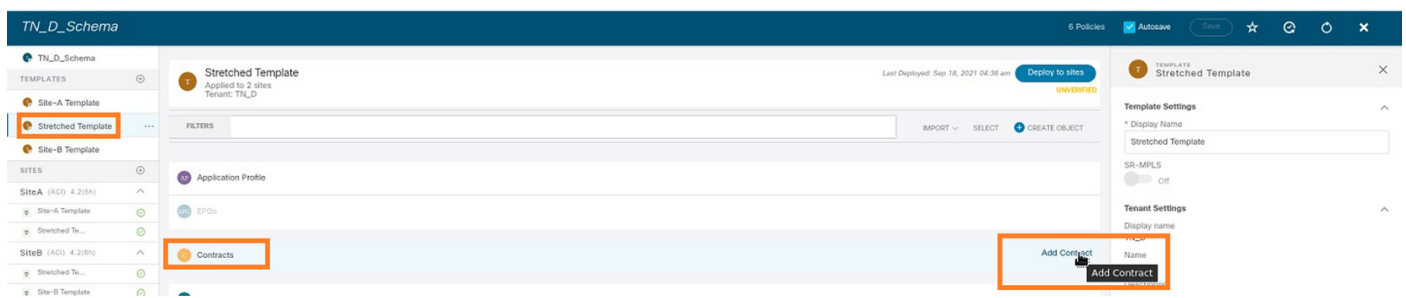
사이트-B의 외부 EPG와 사이트-A의 내부 EPG_990 간에 계약이 필요합니다. 따라서 먼저 MSO에서 계약을 생성하고 두 EPG에 연결할 수 있습니다.

[Cisco Application Centric Infrastructure - Cisco ACI Contract Guide](#)를 통해 계약을 이해할 수 있습니다. 일반적으로 내부 EPG는 공급자로 구성되고 외부 EPG는 소비자로 구성됩니다.



계약 생성

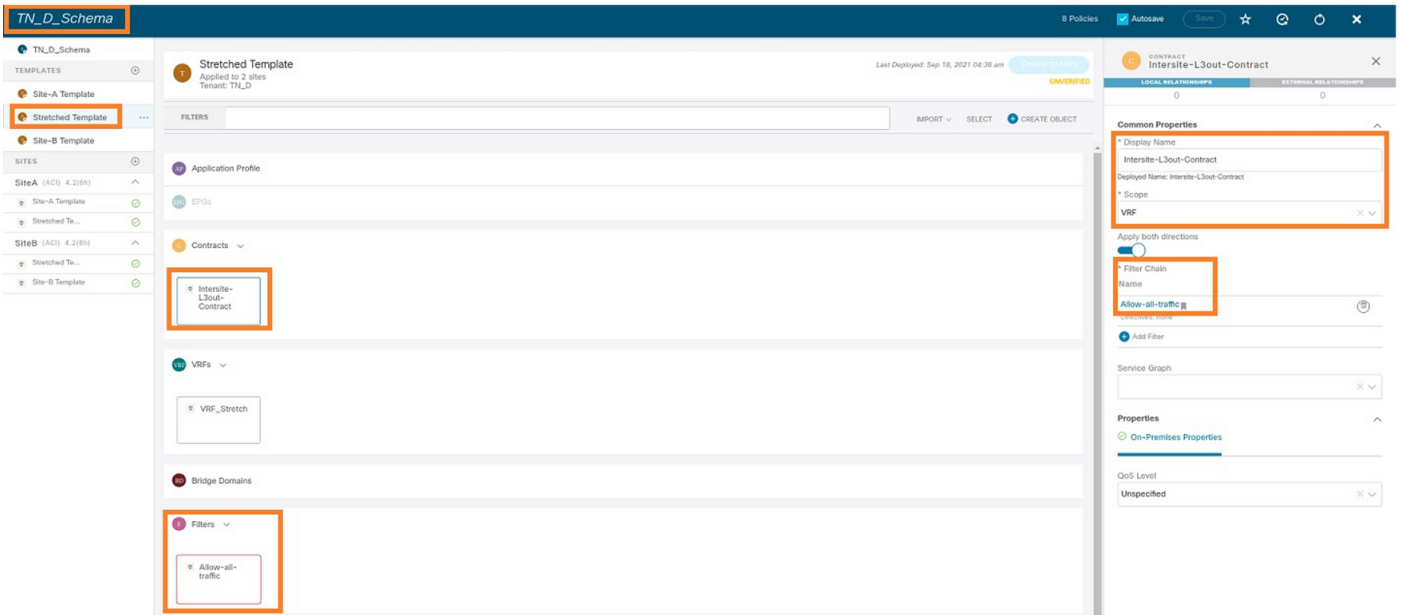
1단계. TN_D_Schema에서 스트레치된 템플릿 > 계약을 선택합니다. 클릭 계약을 추가합니다.



2단계. 필터를 추가하여 모든 트래픽을 허용합니다.

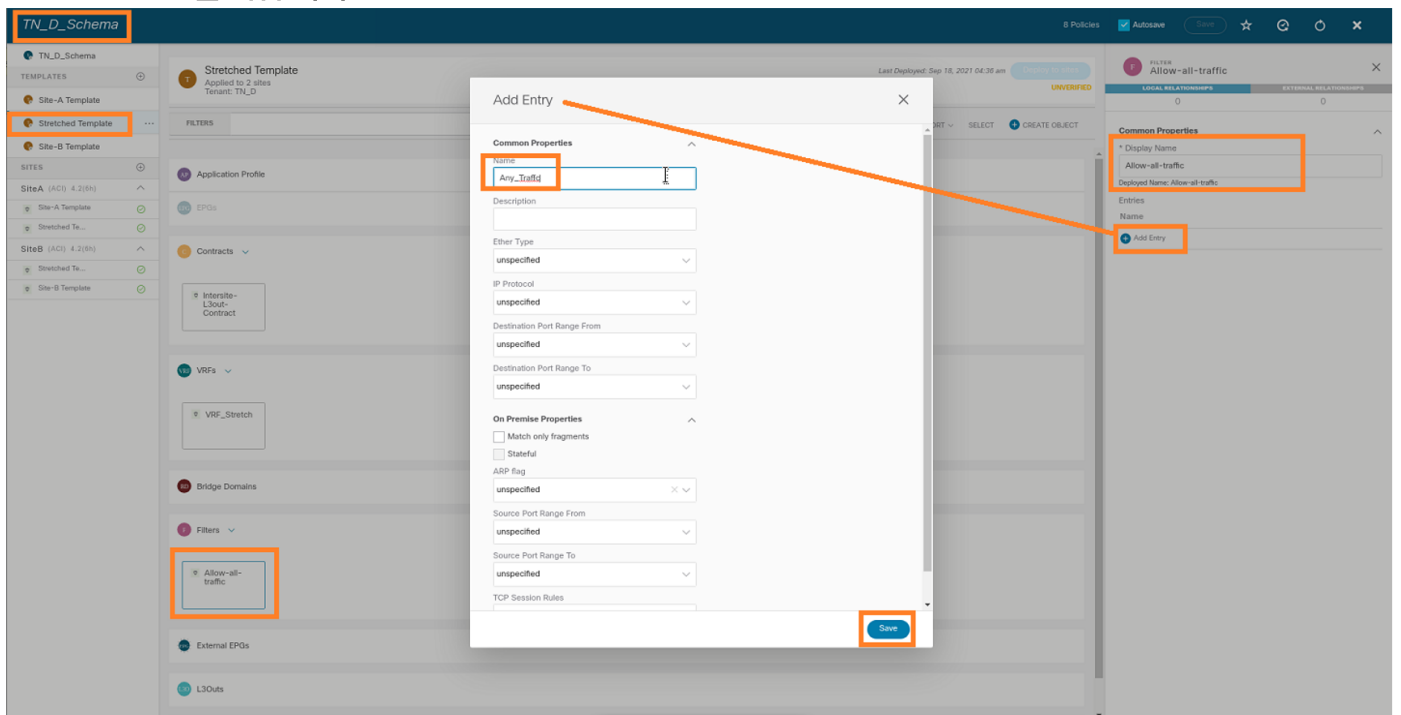
1. TN_D_Schema에서 스트레치된 템플릿 > 계약을 선택합니다.
2. 다음을 사용하여 계약 추가:

- 표시 이름: **Intersite-L3out-Contract**
- 범위: **VRF**



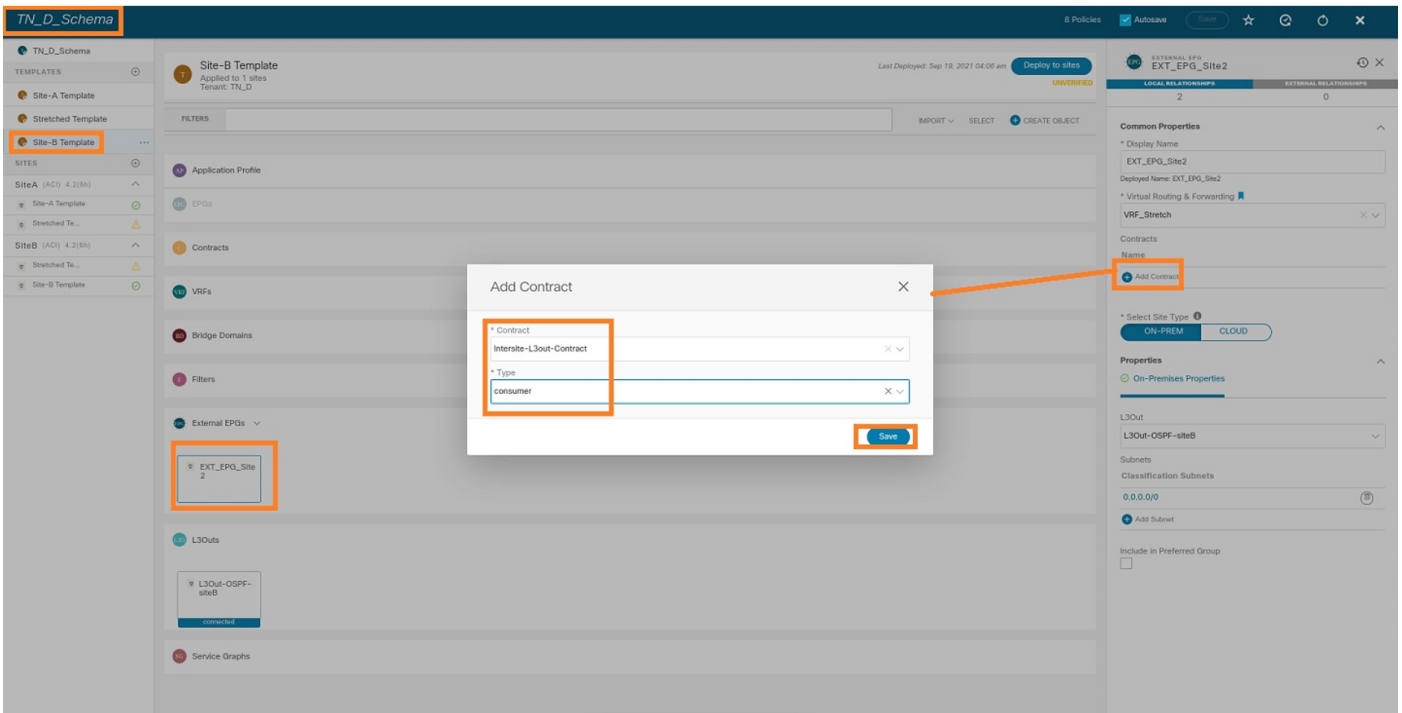
3단계.

1. TN_D_Schema에서 스트레치된 템플릿 > 필터를 선택합니다.
2. Display Name 필드에 Allow-all-traffic을 입력합니다.
3. 항목 추가를 클릭합니다. 항목 추가 대화 상자가 표시됩니다.
4. Name(이름) 필드에 Any_Traffic을 입력합니다.
5. Ether Type 드롭다운 목록에서 모든 트래픽을 허용하려면 미지정을 선택합니다.
6. Save를 클릭합니다.



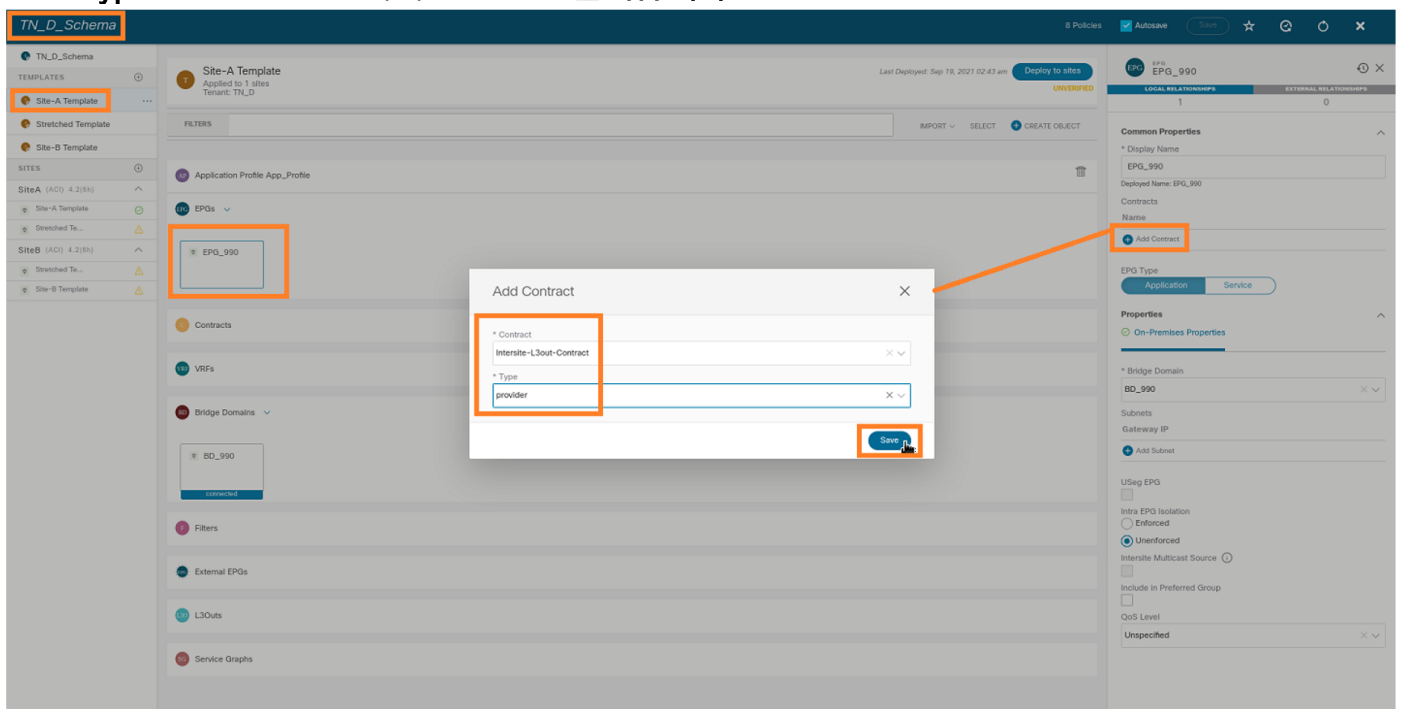
4단계. 외부 EPG에 "소비자"로 계약을 추가합니다(Site-B 템플릿에서)(사이트에 구축).

1. TN_D_Schema에서 Site-B Template > EXT_EPG_Site2를 선택합니다.
2. Add Contract를 클릭합니다. 계약 추가 대화 상자가 표시됩니다.
3. Contract(계약) 필드에 Intersite-L3out-Contract를 입력합니다.
4. Type 드롭다운 목록에서 소비자를 선택합니다.



5단계. 내부 EPG "EPG_990"에 "공급자"(Site-A 템플릿)(Deploy to site)로 계약을 추가합니다.

1. TN_D_Schema에서 Site-A Template > EPG_990을 선택합니다.
2. Add Contract를 클릭합니다. 계약 추가 대화 상자가 표시됩니다.
3. Contract(계약) 필드에 Intersite-L3out-Contract를 입력합니다.
4. Type 드롭다운 목록에서 제공자를 선택합니다.



계약이 추가되면 Site-A에서 생성된 "Shadow L3out / External EPG"를 확인할 수 있습니다.



APIC (SiteA)

System

Tenants

Fabric

Virtual Networking

L4-L7

ALL TENANTS

Add Tenant

Tenant Search:

name or descr



This has been created from Multi-Site. It is recommended to or

TN_D



Quick Start

TN_D

Application Profiles

Networking

Bridge Domains

VRFs

External Bridged Networks

L3Outs

L3Out-OSPF-siteB

Shadow L3out site-B

Logical Node Profiles

External EPGs

EXT_EPG_Site2

Shadow Ext EPG

Route map for import and export route control

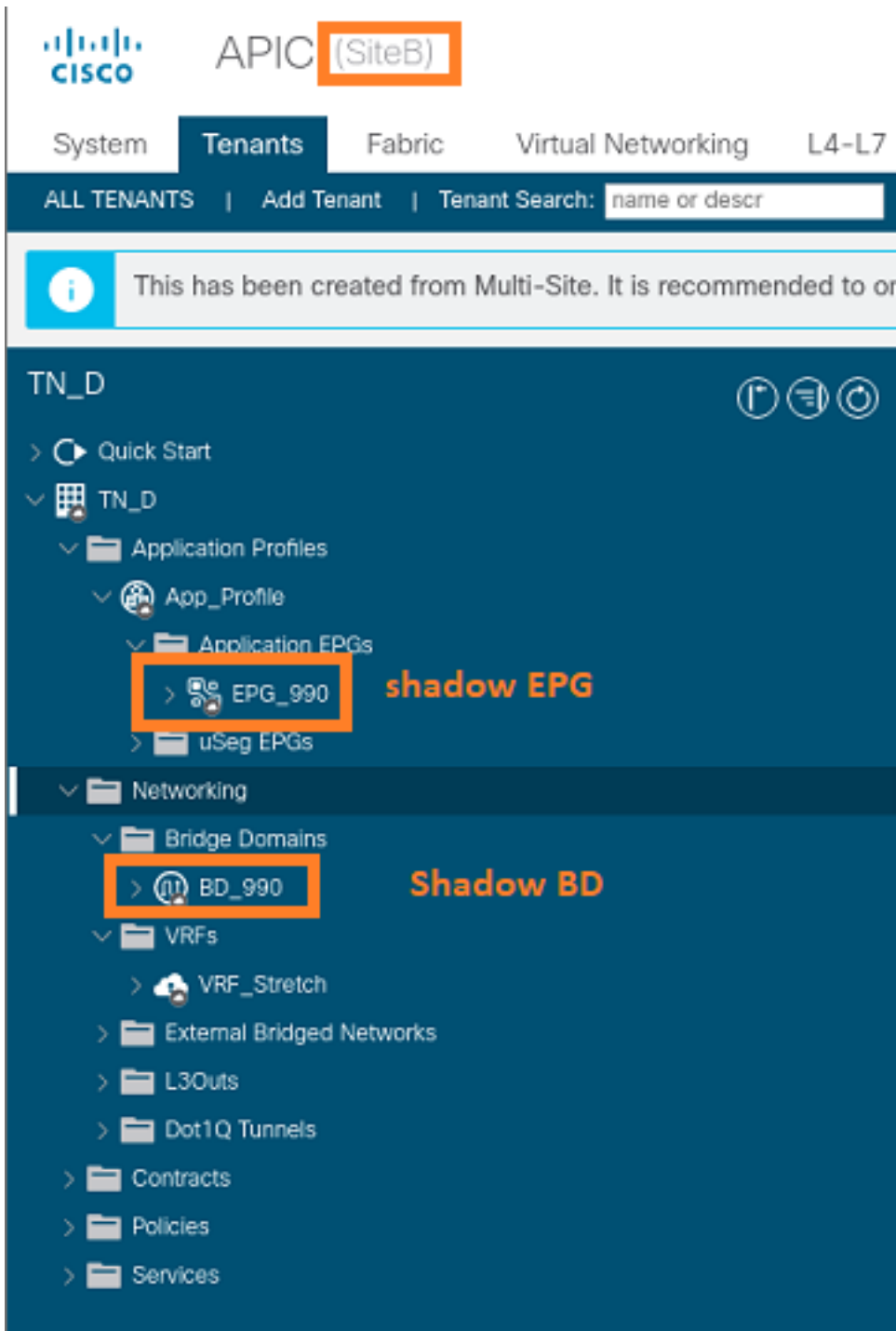
Dot1Q Tunnels

Contracts

Policies

Services

또한 Site-B에서 "Shadow EPG_990 및 BD_990"이 생성되었음을 확인할 수 있습니다.



6단계. Site-B APIC를 확인하려면 다음 명령을 입력합니다.

```

apic1# moquery -c fvAEPg -f 'fv.AEPg.name=="EPG_990"'
Total Objects shown: 1
# fv.AEPg
name : EPG_990
annotation : orchestrator:msc
childAction :
configIssues :
configSt : applied
descr :
dn : uni/tn-TN_D/ap-App_Profile/epg-EPG_990
exceptionTag :
extMngdBy :
floodOnEncap : disabled
fwdCtrl :

```

```

hasMcastSource      : no
isAttrBasedEPg     : no
isSharedSrvMsiteEPg : no
lcOwn               : local
matchT              : AtleastOne
modTs               : 2021-09-19T18:47:53.374+00:00
monPolDn            : uni/tn-common/monepg-default
nameAlias           :
pcEnfPref           : unenforced
pcTag              : 49153          <<< Note that pcTag is different for shadow EPG.
prefGrMemb          : exclude
prio                : unspecified
rn                  : epg-EPG_990
scope               : 2686978
shutdown            : no
status              :
triggerSt           : triggerable
txId                : 1152921504609244629
uid                 : 0

```

```

apic1# moquery -c fvBD -f 'fv.BD.name=="BD_990\"'

```

```

Total Objects shown: 1

```

```

# fv.BD

```

```

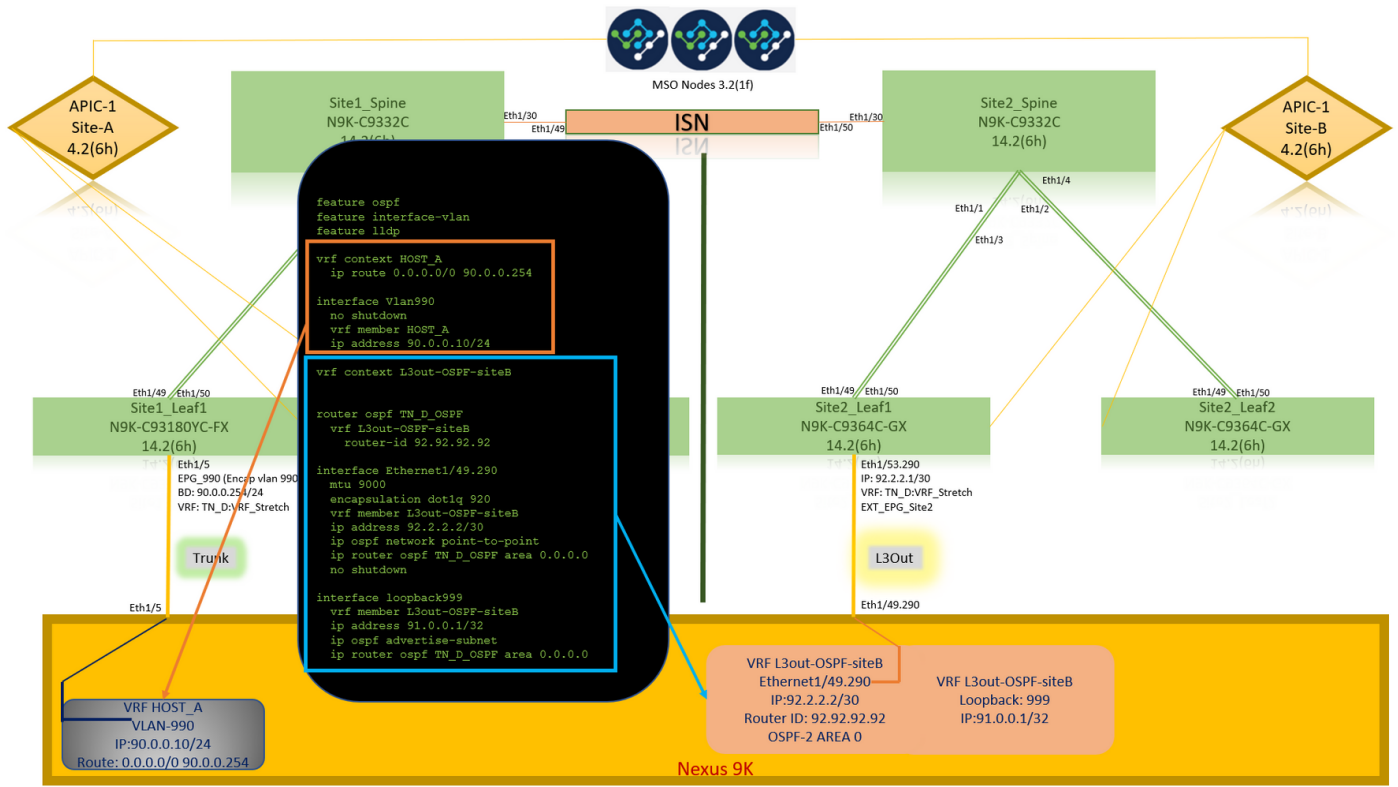
name              : BD_990
OptimizeWanBandwidth : yes
annotation            : orchestrator:msc
arpFlood              : yes
bcastP                : 225.0.181.192
childAction           :
configIssues          :
descr                 :
dn                : uni/tn-TN_D/BD-BD_990
epClear               : no
epMoveDetectMode     :
extMngdBy             :
hostBasedRouting      : no
intersiteBumTrafficAllow : yes
intersiteL2Stretch    : yes
ipLearning            : yes
ipv6McastAllow        : no
lcOwn                 : local
limitIpLearnToSubnets : yes
llAddr                : ::
mac                   : 00:22:BD:F8:19:FF
mcastAllow            : no
modTs                 : 2021-09-19T18:47:53.374+00:00
monPolDn              : uni/tn-common/monepg-default
mtu                   : inherit
multiDstPktAct        : bd-flood
nameAlias             :
ownerKey              :
ownerTag              :
pcTag                 : 32771
rn                    : BD-BD_990
scope                 : 2686978
seg                   : 15957972
status                :
type                  : regular
uid                   : 0
unicastRoute          : yes
unkMacUcastAct    : proxy
unkMcastAct      : flood

```



```
v6unkMcastAct      : flood
vmac                : not-applicable
```

7단계. 외부 디바이스 N9K 컨피그레이션을 검토하고 확인합니다.



다음을 확인합니다.

이 섹션을 사용하여 컨피그레이션이 제대로 작동하는지 확인합니다.

엔드포인트 학습

Site-A 엔드포인트가 Site1_Leaf1에서 엔드포인트로 학습되었는지 확인합니다.

```
Site1_Leaf1# show endpoint interface ethernet 1/5
```

Legend:

```

s - arp          H - vtep          V - vpc-attached    p - peer-aged
R - peer-attached-rl B - bounce      S - static          M - span
D - bounce-to-proxy O - peer-attached a - local-aged    m - svc-mgr
L - local        E - shared-service
  
```

VLAN/ Interface Domain	Encap VLAN	MAC Address IP Address	MAC Info/ IP Info
18	vlan-990	c014.fe5e.1407	L
eth1/5			
TN_D:VRF_Stretch	vlan-990	90.0.0.10	L
			eth1/5

ETEP/RTEP 확인

Site_A 리프

Site1_Leaf1# show ip interface brief vrf overlay-1

IP Interface Status for VRF "overlay-1"(4)

Interface	Address	Interface Status
eth1/49	unassigned	protocol-up/link-up/admin-up
eth1/49.7	unnumbered (lo0)	protocol-up/link-up/admin-up
eth1/50	unassigned	protocol-up/link-up/admin-up
eth1/50.8	unnumbered (lo0)	protocol-up/link-up/admin-up
eth1/51	unassigned	protocol-down/link-down/admin-up
eth1/52	unassigned	protocol-down/link-down/admin-up
eth1/53	unassigned	protocol-down/link-down/admin-up
eth1/54	unassigned	protocol-down/link-down/admin-up
vlan9	10.0.0.30/27	protocol-up/link-up/admin-up
lo0	10.0.80.64/32	protocol-up/link-up/admin-up
lo1	10.0.8.67/32	protocol-up/link-up/admin-up
lo8	192.168.200.225/32	protocol-up/link-up/admin-up <<<<< IP from ETEP site-A
lo1023	10.0.0.32/32	protocol-up/link-up/admin-up

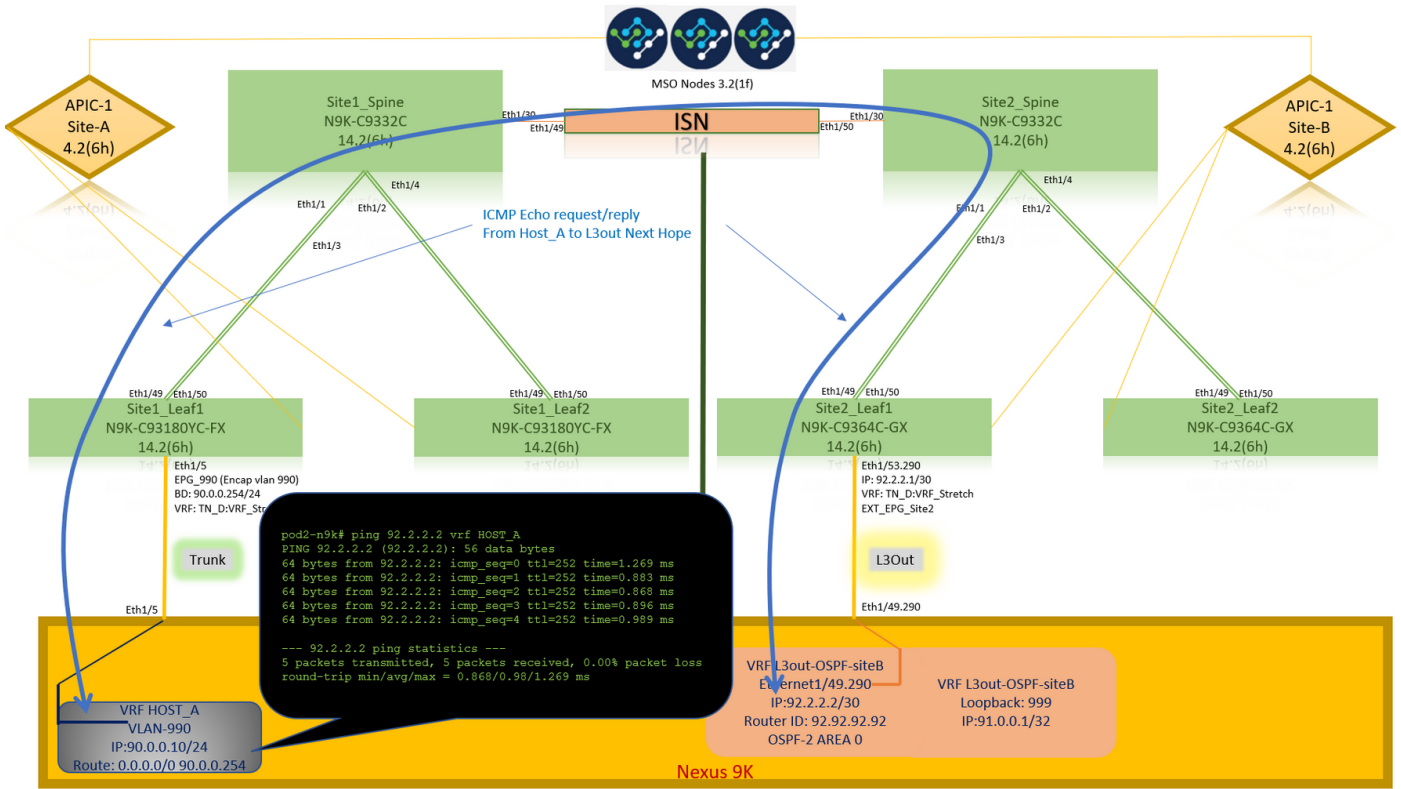
Site2_Leaf1# show ip interface brief vrf overlay-1

IP Interface Status for VRF "overlay-1"(4)

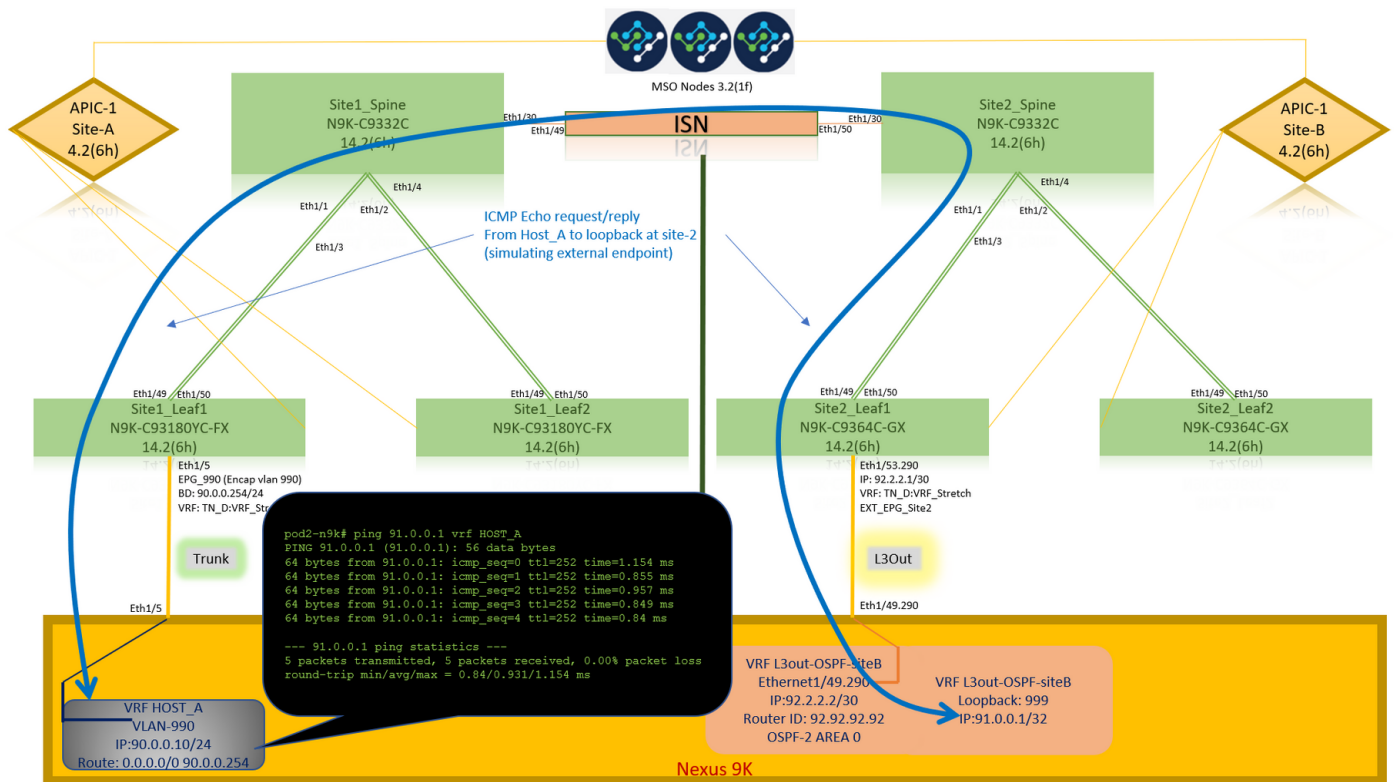
Interface	Address	Interface Status
eth1/49	unassigned	protocol-up/link-up/admin-up
eth1/49.16	unnumbered (lo0)	protocol-up/link-up/admin-up
eth1/50	unassigned	protocol-up/link-up/admin-up
eth1/50.17	unnumbered (lo0)	protocol-up/link-up/admin-up
eth1/51	unassigned	protocol-down/link-down/admin-up
eth1/52	unassigned	protocol-down/link-down/admin-up
eth1/54	unassigned	protocol-down/link-down/admin-up
eth1/55	unassigned	protocol-down/link-down/admin-up
eth1/56	unassigned	protocol-down/link-down/admin-up
eth1/57	unassigned	protocol-down/link-down/admin-up
eth1/58	unassigned	protocol-down/link-down/admin-up
eth1/59	unassigned	protocol-down/link-down/admin-up
eth1/60	unassigned	protocol-down/link-down/admin-up
eth1/61	unassigned	protocol-down/link-down/admin-up
eth1/62	unassigned	protocol-down/link-down/admin-up
eth1/63	unassigned	protocol-down/link-down/admin-up
eth1/64	unassigned	protocol-down/link-down/admin-up
vlan18	10.0.0.30/27	protocol-up/link-up/admin-up
lo0	10.0.72.64/32	protocol-up/link-up/admin-up
lo1	10.0.80.67/32	protocol-up/link-up/admin-up
lo6	192.168.100.225/32	protocol-up/link-up/admin-up <<<<< IP from ETEP site-B
lo1023	10.0.0.32/32	protocol-up/link-up/admin-up

ICMP 연결성

HOST_A에서 외부 디바이스 WAN IP 주소를 ping합니다.

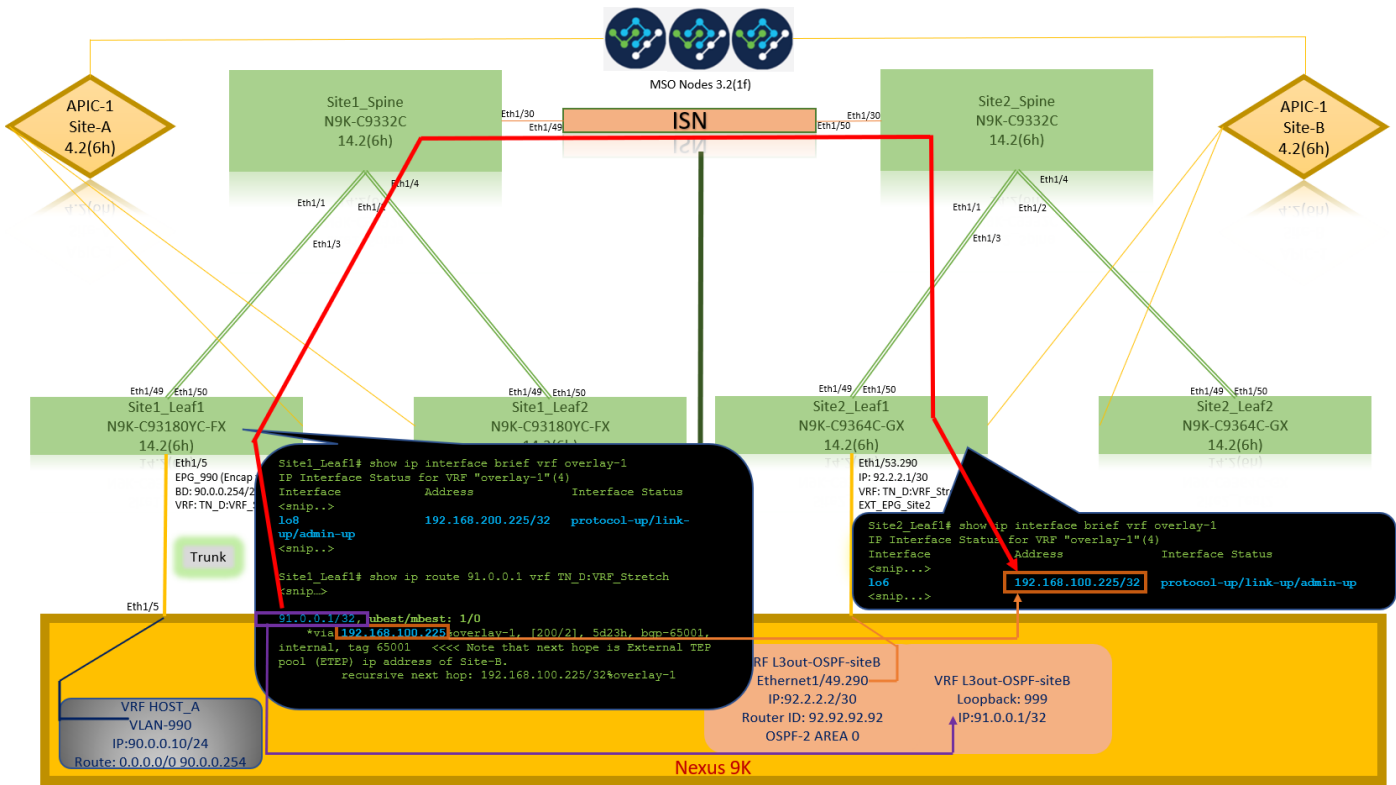


외부 디바이스 루프백 주소를 ping합니다.



경로 확인

외부 디바이스 WAN IP 주소 또는 루프백 서브넷 경로가 라우팅 테이블에 있는지 확인합니다. "Site1_Leaf1"에서 외부 디바이스 서브넷에 대한 다음 홉을 선택하면 "Site2-Leaf1"의 외부 TEP IP가 됩니다.



```

Site1_Leaf1# show ip route 92.2.2.2 vrf TN_D:VRF_Stretch
IP Route Table for VRF "TN_D:VRF_Stretch"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'% ' in via output denotes VRF
92.2.2.0/30, ubest/mbest: 1/0
  *via 192.168.100.225%overlay-1, [200/0], 5d23h, bgp-65001, internal, tag 65001  <<<< Note
that next hope is External TEP pool (ETEP) ip address of Site-B.
    recursive next hop: 192.168.100.225/32%overlay-1
Site1_Leaf1# show ip route 91.0.0.1 vrf TN_D:VRF_Stretch
IP Route Table for VRF "TN_D:VRF_Stretch"
'*' denotes best ucast next-hop
*** denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'% ' in via output denotes VRF
91.0.0.1/32, ubest/mbest: 1/0
  *via 192.168.100.225%overlay-1, [200/2], 5d23h, bgp-65001, internal, tag 65001  <<<< Note
that next hope is External TEP pool (ETEP) ip address of Site-B.
    recursive next hop: 192.168.100.225/32%overlay-1

```

문제 해결

이 섹션에서는 컨피그레이션 문제를 해결하는 데 사용할 수 있는 정보를 제공합니다.

사이트2_리프1

TN_D:VRF_stretch와 Overlay-1 간의 BGP 주소균 경로 가져오기/내보내기

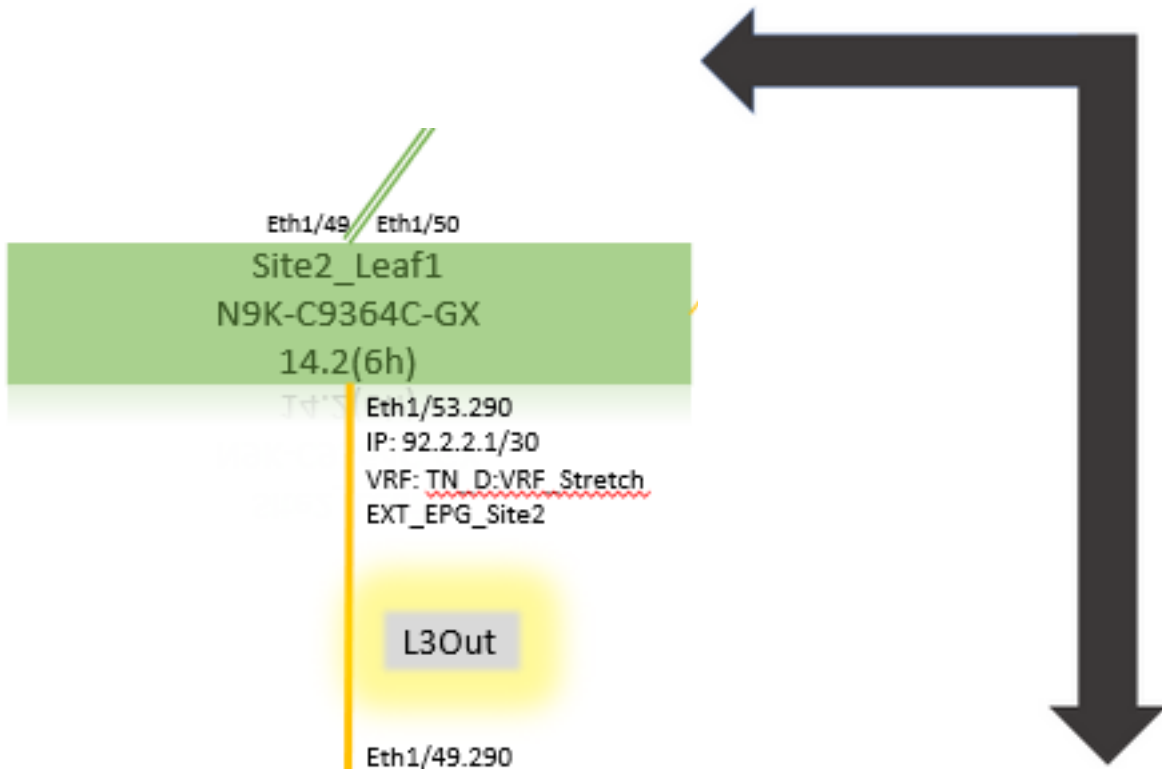
```
Site2_Leaf1# show system internal epn vrf TN_D:VRF_Stretch
```

```
-----
```

VRF	Type	VRF vnid	Context ID	Status	Endpoint Count
TN_D:VRF_Stretch	Tenant	2686978	46	Up	1

Site2_Leaf1# show vrf TN_D:VRF_Stretch detail

VRF-Name: TN_D:VRF_Stretch, VRF-ID: 46, State: Up
 VPNID: unknown
RD: 1101:2686978
 Max Routes: 0 Mid-Threshold: 0
 Table-ID: 0x8000002e, AF: IPv6, Fwd-ID: 0x8000002e, State: Up
 Table-ID: 0x0000002e, AF: IPv4, Fwd-ID: 0x0000002e, State: Up



Site2_Leaf1# vsh

Site2_Leaf1# show bgp vpnv4 unicast 91.0.0.1 vrf TN_D:VRF_Stretch

BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
Route Distinguisher: 1101:2686978 (VRF TN_D:VRF_Stretch)
BGP routing table entry for 91.0.0.1/32, version 12 dest ptr 0xae6da350
 Paths: (1 available, best #1)
 Flags: (0x80c0002 00000000) on xmit-list, is not in urib, exported
 vpn: version 346, (0x100002) on xmit-list
 Multipath: eBGP iBGP

Advertised path-id 1, VPN AF advertised path-id 1
 Path type: redistrib 0x408 0x1 ref 0 adv path ref 2, path is valid, is best path
 AS-Path: NONE, path locally originated

0.0.0.0 (metric 0) from 0.0.0.0 (10.0.72.64)
 Origin incomplete, MED 2, localpref 100, weight 32768
 Extcommunity:

RT:65001:2686978
VNID:2686978
 COST:pre-bestpath:162:110

VRF advertise information:
 Path-id 1 not advertised to any peer
 VPN AF advertise information:
 Path-id 1 advertised to peers:

10.0.72.65

```

apic1# acidiag fmvread ID Pod ID Name Serial Number IP Address Role State LastUpdMsgId -----
----- 101 1

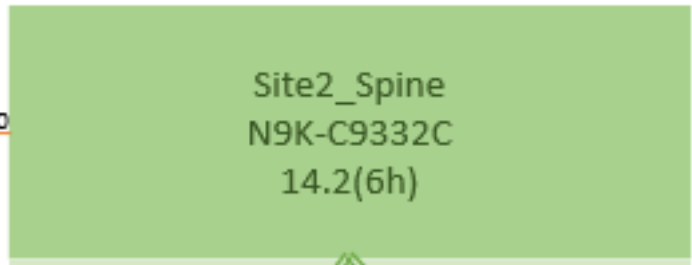
```

Site2_Spine FDO243207JH

```

10.0.72.65/32 spine active 0 102 1 Site2_Leaf2 FDO24260FCH 10.0.72.66/32 leaf active 0 1101
1 Site2_Leaf1 FDO24260ECW 10.0.72.64/32 leaf active 0

```

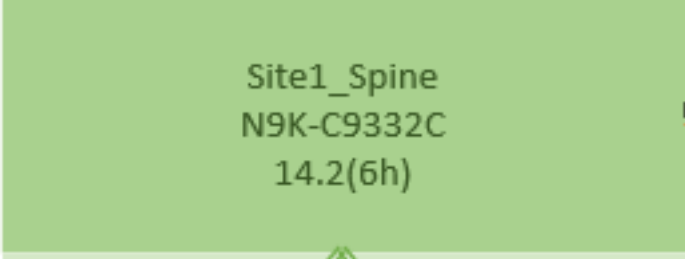


사이트2_스파인

```

Site2_Spine# vsh
Site2_Spine# show bgp vpnv4 unicast 91.0.0.1 vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
<-----26bits----->
Route Distinguisher: 1101:2686978 <<<<<2686978 <--
Binary--> 000010100100000000000000010
BGP routing table entry for 91.0.0.1/32, version 717 dest ptr 0xae643d0c
Paths: (1 available, best #1)
Flags: (0x000002 00000000) on xmit-list, is not in urib, is not in HW
Multipath: eBGP iBGP
  Advertised path-id 1
  Path type: internal 0x40000018 0x800040 ref 0 adv path ref 1, path is valid, is best path
  AS-Path: NONE, path sourced internal to AS
  10.0.72.64 (metric 2) from 10.0.72.64 (10.0.72.64) <<< Site2_leaf1 IP
  Origin incomplete, MED 2, localpref 100, weight 0
  Received label 0
  Received path-id 1
  Extcommunity:
    RT:65001:2686978
    COST:pre-bestpath:168:3221225472
    VNID:2686978
    COST:pre-bestpath:162:110
  Path-id 1 advertised to peers:
    192.168.10.13 <<<< Site1_Spine mscp-etest IP.
Site1_Spine# show ip interface vrf overlay-1
<snip...>
lo12, Interface status: protocol-up/link-up/admin-up, iod: 89, mode: mscp-etest
IP address: 192.168.10.13, IP subnet: 192.168.10.13/32 <<

```



사이트1_스파인

```

Site1_Spine# vsh
Site1_Spine# show bgp vpnv4 unicast 91.0.0.1 vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
<-----26Bits----->
Route Distinguisher: 1101:36241410
<<<<<36241410<--binary-->100010100100000000000000010
BGP routing table entry for 91.0.0.1/32, version 533 dest ptr 0xae643dd4
Paths: (1 available, best #1)
Flags: (0x000002 00000000) on xmit-list, is not in urib, is not in HW
Multipath: eBGP iBGP
  Advertised path-id 1
    Path type: internal 0x40000018 0x880000 ref 0 adv path ref 1, path is valid, is best path,
remote site path
  AS-Path: NONE, path sourced internal to AS
    192.168.100.225 (metric 20) from 192.168.11.13 (192.168.11.13) <<< Site2_Leaf1 ETEP IP
learn via Site2_Spine mcsp-etest address.
  Origin incomplete, MED 2, localpref 100, weight 0
  Received label 0
  Extcommunity:
    RT:65001:36241410
    SOO:65001:50331631
    COST:pre-bestpath:166:2684354560
    COST:pre-bestpath:168:3221225472
    VNID:2686978
    COST:pre-bestpath:162:110
  Originator: 10.0.72.64 Cluster list: 192.168.11.13 <<< Originator Site2_Leaf1 and
Site2_Spine ips are listed here...
  Path-id 1 advertised to peers:
    10.0.80.64 <<<< Site1_Leaf1 ip

```

```

Site2_Spine# show ip interface vrf overlay-1
<snip..>
lo13, Interface status: protocol-up/link-up/admin-up, iod: 92, mode: mcsp-etest IP address:
192.168.11.13, IP subnet: 192.168.11.13/32
  IP broadcast address: 255.255.255.255
  IP primary address route-preference: 0, tag: 0
<snip..>

```

```

Site-B apic1# acidiag fvnread

```

ID	Pod ID	Name	Serial Number	IP Address	Role	State
101	1	Site2_Spine	FDO243207JH	10.0.72.65/32	spine	active 0
102	1	Site2_Leaf2	FDO24260FCH	10.0.72.66/32	leaf	active 0
1101	1	Site2_Leaf1	FDO24260ECW	10.0.72.64/32	leaf	active 0

사이트 간 플래그를 확인합니다.

```

Site1_Spine# moquery -c bgpPeer -f 'bgp.Peer.addr*"192.168.11.13"'

```

```

Total Objects shown: 1
# bgp.Peer
addr           : 192.168.11.13/32
activePfxPeers : 0
adminSt        : enabled
asn            : 65001
bgpCfgFailedBmp :
bgpCfgFailedTs : 00:00:00:00.000
bgpCfgState    : 0
childAction    :
ctrl           :
curPfxPeers    : 0
dn             : sys/bgp/inst/dom-overlay-1/peer-[192.168.11.13/32]
lcOwn          : local
maxCurPeers   : 0
maxPfxPeers    : 0
modTs          : 2021-09-13T11:58:26.395+00:00
monPolDn      :
name           :
passwdSet      : disabled
password       :
peerRole       : msite-speaker
privateASctrl  :
rn             : peer-[192.168.11.13/32] <<

```

<<

경로 구별자 항목 이해사이트 간 플래그가 설정되면 로컬 사이트 스파인은 25번째 비트부터 경로 대상에서 로컬 사이트 ID를 설정할 수 있습니다. Site1은 RT에 이 비트가 설정된 BGP 경로를 가져오면 원격 사이트 경로임을 알게 됩니다.

```

Site2_Leaf1# vsh
Site2_Leaf1# show bgp vpnv4 unicast 91.0.0.1 vrf TN_D:VRF_Stretch
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
<-----26Bits----->
Route Distinguisher: 1101:2686978      (VRF TN_D:VRF_Stretch)                <<<<<2686978
<--Binary--> 00001010010000000000000010
BGP routing table entry for 91.0.0.1/32, version 12 dest ptr 0xae6da350

```

```

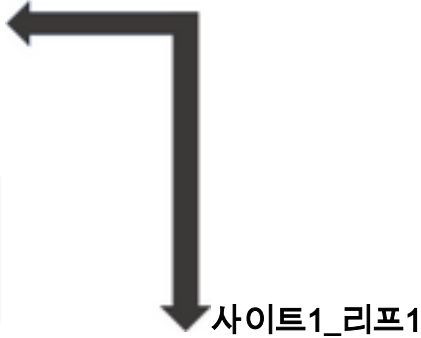
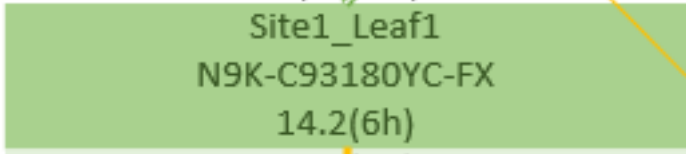
Site1_Spine# vsh
Site1_Spine# show bgp vpnv4 unicast 91.0.0.1 vrf overlay-1

<-----26Bits----->
Route Distinguisher: 1101:36241410
<<<<<36241410<--Binary-->10001010010000000000000010

```

^^---26th bit set to 1 and with 25th bit value it become 10.

26번째 비트가 1로 설정된 경우를 제외하고 Site1의 RT 이진 값은 정확히 동일합니다. 10진수 값 (파란색으로 표시됨)이 있습니다. 1101:36241410은 Site1에서 볼 수 있는 내용과 Site1의 내부 leaf를 가져와야 하는 내용을 나타냅니다



```
Site1_Leaf1# show vrf TN_D:VRF_Stretch detail
VRF-Name: TN_D:VRF_Stretch, VRF-ID: 46, State: Up
VPNID: unknown
RD: 1101:2850817
Max Routes: 0 Mid-Threshold: 0
Table-ID: 0x8000002e, AF: IPv6, Fwd-ID: 0x8000002e, State: Up
Table-ID: 0x0000002e, AF: IPv4, Fwd-ID: 0x0000002e, State: Up
```

```
Site1_Leaf1# show bgp vpnv4 unicast 91.0.0.1 vrf overlay-1
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
Route Distinguisher: 1101:2850817 (VRF TN_D:VRF_Stretch)
BGP routing table entry for 91.0.0.1/32, version 17 dest ptr 0xadeda550
Paths: (1 available, best #1)
Flags: (0x08001a 00000000) on xmit-list, is in urib, is best urib route, is in HW
vpn: version 357, (0x100002) on xmit-list
Multipath: eBGP iBGP
  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: internal 0xc0000018 0x80040 ref 56506 adv path ref 2, path is valid, is best path,
remote site path
    Imported from 1101:36241410:91.0.0.1/32
AS-Path: NONE, path sourced internal to AS
  192.168.100.225 (metric 64) from 10.0.80.65 (192.168.10.13)
  Origin incomplete, MED 2, localpref 100, weight 0
  Received label 0
  Received path-id 1
  Extcommunity:
    RT:65001:36241410
    SOO:65001:50331631
    COST:pre-bestpath:166:2684354560
    COST:pre-bestpath:168:3221225472
    VNID:2686978
    COST:pre-bestpath:162:110
  Originator: 10.0.72.64 Cluster list: 192.168.10.13 192.168.11.13 <<<<
'10.0.72.64'='Site2_Leaf1' , '192.168.10.13'='Site1_Spine' , '192.168.11.13'='Site2_Spine'
  VRF advertise information:
  Path-id 1 not advertised to any peer
  VPN AF advertise information:
  Path-id 1 not advertised to any peer
```

```
<snip..>
Site1_Leaf1# show bgp vpnv4 unicast 91.0.0.1 vrf TN_D:VRF_Stretch
BGP routing table information for VRF overlay-1, address family VPNv4 Unicast
Route Distinguisher: 1101:2850817 (VRF TN_D:VRF_Stretch)
BGP routing table entry for 91.0.0.1/32, version 17 dest ptr 0xadeda550
Paths: (1 available, best #1)
Flags: (0x08001a 00000000) on xmit-list, is in urib, is best urib route, is in HW
vpn: version 357, (0x100002) on xmit-list
Multipath: eBGP iBGP
  Advertised path-id 1, VPN AF advertised path-id 1
  Path type: internal 0xc0000018 0x80040 ref 56506 adv path ref 2, path is valid, is best path,
remote site path
    Imported from 1101:36241410:91.0.0.1/32
AS-Path: NONE, path sourced internal to AS
  192.168.100.225 (metric 64) from 10.0.80.65 (192.168.10.13)
  Origin incomplete, MED 2, localpref 100, weight 0
```

```

Received label 0
Received path-id 1
Extcommunity:
  RT:65001:36241410
  SOO:65001:50331631
  COST:pre-bestpath:166:2684354560
  COST:pre-bestpath:168:3221225472
  VNID:2686978
  COST:pre-bestpath:162:110
Originator: 10.0.72.64 Cluster list: 192.168.10.13 192.168.11.13
VRF advertise information:
Path-id 1 not advertised to any peer
VPN AF advertise information:
Path-id 1 not advertised to any peer

```

따라서 "Site1_Leaf1"에는 next-hop "Site2_Leaf1" ETEP 주소가 192.168.100.225인 서브넷 91.0.0.1/32에 대한 경로 항목이 있습니다.

```

Site1_Leaf1# show ip route 91.0.0.1 vrf TN_D:VRF_Stretch
IP Route Table for VRF "TN_D:VRF_Stretch"
 '*' denotes best ucast next-hop
 '**' denotes best mcast next-hop
 '[x/y]' denotes [preference/metric]
 '%' in via output denotes VRF
91.0.0.1/32, ubest/mbest: 1/0
  *via 192.168.100.225%overlay-1, [200/2], 5d23h, bgp-65001, internal, tag 65001 <<<< Note
that next hope is External TEP pool (ETEP) ip address of Site-B.
  recursive next hop: 192.168.100.225/32%overlay-1

```

Site-A Spine은 "Site2_Spine" mosp-ETEP의 BGP 네이버 IP 주소에 대한 경로 맵을 추가합니다. 따라서 트래픽 플로우에 대해 생각할 때 Site-A 엔드포인트가 외부 IP 주소와 통신할 때 패킷은 소스와 함께 "Site1_Leaf1" TEP 주소로 캡슐화할 수 있으며 대상은 "Site2_Leaf" IP 주소 192.168.100.225의 ETEP 주소입니다. ELAM 확인(Site1_Spine)

```

Site1_Spine# vsh_lc
module-1# debug platform internal roc elam asic 0
module-1(DBG-elam)# trigger reset
module-1(DBG-elam)# trigger init in-select 14 out-select 1
module-1(DBG-elam-insel14)# set inner ipv4 src_ip 90.0.0.10 dst_ip 91.0.0.1 next-protocol 1
module-1(DBG-elam-insel14)# start
module-1(DBG-elam-insel14)# status
  ELAM STATUS
  =====
  Asic 0 Slice 0 Status Armed
  Asic 0 Slice 1 Status Armed
  Asic 0 Slice 2 Status Armed
  Asic 0 Slice 3 Status Armed

```

```

pod2-n9k# ping 91.0.0.1 vrf HOST_A source 90.0.0.10
PING 91.0.0.1 (91.0.0.1) from 90.0.0.10: 56 data bytes
64 bytes from 91.0.0.1: icmp_seq=0 ttl=252 time=1.015 ms
64 bytes from 91.0.0.1: icmp_seq=1 ttl=252 time=0.852 ms
64 bytes from 91.0.0.1: icmp_seq=2 ttl=252 time=0.859 ms
64 bytes from 91.0.0.1: icmp_seq=3 ttl=252 time=0.818 ms
64 bytes from 91.0.0.1: icmp_seq=4 ttl=252 time=0.778 ms
--- 91.0.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 0.778/0.864/1.015 ms

```

Site1_Spine ELAM0이 트리거됩니다. Ereport는 패킷이 Site-A Leaf TEP IP 주소의 TEP 주소와 Site2_Leaf1 ETEP 주소를 향하는 대상으로 캡슐화되는지 확인합니다.

```

module-1(DBG-elam-insel14)# status
  ELAM STATUS

```

```
=====
Asic 0 Slice 0 Status Armed
Asic 0 Slice 1 Status Armed
Asic 0 Slice 2 Status Triggered
Asic 0 Slice 3 Status Armed
module-1(DBG-elam-insell14)# ereport
Python available. Continue ELAM decode with LC Pkg
ELAM REPORT
```

```
-----
Outer L3 Header
-----
```

```
-----
L3 Type           : IPv4
DSCP              : 0
Don't Fragment Bit : 0x0
TTL              : 32
IP Protocol Number : UDP
Destination IP    : 192.168.100.225    <<<'Site2_Leaf1' ETEP address
Source IP         : 10.0.80.64         <<<'Site1_Leaf1' TEP address
-----
```

```
-----
Inner L3 Header
-----
```

```
-----
L3 Type           : IPv4
DSCP              : 0
Don't Fragment Bit : 0x0
TTL              : 254
IP Protocol Number : ICMP
Destination IP    : 91.0.0.1
Source IP         : 90.0.0.10
-----
```

Site1_Spine 경로 맵 확인 Site-A 스파인이 패킷을 수신하면 coop 또는 경로 항목을 보는 대신 "Site2_Leaf1" ETEP 주소로 리디렉션할 수 있습니다. (사이트-B에 사이트-L3out이 있는 경우 사이트-A 스파인은 "infra-intersite-l3out"이라는 경로 맵을 만들어 Site2_Leaf1의 ETEP로 트래픽을 리디렉션하고 L3out에서 나갑니다.)

```
Site1_Spine# show bgp vpnv4 unicast neighbors 192.168.11.13 vrf overlay-1
BGP neighbor is 192.168.11.13, remote AS 65001, ibgp link, Peer index 4
  BGP version 4, remote router ID 192.168.11.13
  BGP state = Established, up for 10w4d
  Using loopback12 as update source for this peer
  Last read 00:00:03, hold time = 180, keepalive interval is 60 seconds
  Last written 00:00:03, keepalive timer expiry due 00:00:56
  Received 109631 messages, 0 notifications, 0 bytes in queue
  Sent 109278 messages, 0 notifications, 0 bytes in queue
  Connections established 1, dropped 0
  Last reset by us never, due to No error
  Last reset by peer never, due to No error
  Neighbor capabilities:
  Dynamic capability: advertised (mp, refresh, gr) received (mp, refresh, gr)
  Dynamic capability (old): advertised received
  Route refresh capability (new): advertised received
  Route refresh capability (old): advertised received
  4-Byte AS capability: advertised received
  Address family VPNv4 Unicast: advertised received
  Address family VPNv6 Unicast: advertised received
  Address family L2VPN EVPN: advertised received
  Graceful Restart capability: advertised (GR helper) received (GR helper)
  Graceful Restart Parameters:
```

Address families advertised to peer:
 Address families received from peer:
 Forwarding state preserved by peer for:
 Restart time advertised by peer: 0 seconds
 Additional Paths capability: advertised received
 Additional Paths Capability Parameters:
 Send capability advertised to Peer for AF:
 L2VPN EVPN
 Receive capability advertised to Peer for AF:
 L2VPN EVPN
 Send capability received from Peer for AF:
 L2VPN EVPN
 Receive capability received from Peer for AF:
 L2VPN EVPN
 Additional Paths Capability Parameters for next session:
 [E] - Enable [D] - Disable
 Send Capability state for AF:
 VPNv4 Unicast[E] VPNv6 Unicast[E]
 Receive Capability state for AF:
 VPNv4 Unicast[E] VPNv6 Unicast[E]
 Extended Next Hop Encoding Capability: advertised received
 Receive IPv6 next hop encoding Capability for AF:
 IPv4 Unicast
 Message statistics:

	Sent	Rcvd
Opens:	1	1
Notifications:	0	0
Updates:	1960	2317
Keepalives:	107108	107088
Route Refresh:	105	123
Capability:	104	102
Total:	109278	109631
Total bytes:	2230365	2260031
Bytes in queue:	0	0

For address family: VPNv4 Unicast
 BGP table version 533, neighbor version 533
 3 accepted paths consume 360 bytes of memory
 3 sent paths
 0 denied paths

Community attribute sent to this neighbor
 Extended community attribute sent to this neighbor
 Third-party Nexthop will not be computed.

Outbound route-map configured is infra-intersite-l3out, handle obtained <<<< route-map to redirect traffic from Site-A to Site-B 'Site2_Leaf1' L3out

For address family: VPNv6 Unicast
 BGP table version 241, neighbor version 241
 0 accepted paths consume 0 bytes of memory
 0 sent paths
 0 denied paths

Community attribute sent to this neighbor
 Extended community attribute sent to this neighbor
 Third-party Nexthop will not be computed.

Outbound route-map configured is infra-intersite-l3out, handle obtained
 <snip...> Site1_Spine# show route-map infra-intersite-l3out

route-map infra-intersite-l3out, permit, sequence 1

Match clauses:

ip next-hop prefix-lists: IPv4-Node-entry-102
 ipv6 next-hop prefix-lists: IPv6-Node-entry-102

Set clauses:

ip next-hop 192.168.200.226

route-map infra-intersite-l3out, permit, sequence 2 <<<< This route-map match if destination IP of packet 'Site1_Spine' TEP address then send to 'Site2_Leaf1' TEP address.

Match clauses:

ip next-hop prefix-lists: IPv4-Node-entry-1101

```
    ipv6 next-hop prefix-lists: IPv6-Node-entry-1101
Set clauses:
    ip next-hop 192.168.200.225
route-map infra-intersite-l3out, deny, sequence 999
Match clauses:
    ip next-hop prefix-lists: infra_prefix_local_pteps_inexact
Set clauses:
route-map infra-intersite-l3out, permit, sequence 1000
Match clauses:
Set clauses:
    ip next-hop unchanged
```

```
Site1_Spine# show ip prefix-list IPv4-Node-entry-1101
ip prefix-list IPv4-Node-entry-1101: 1 entries
seq 1 permit 10.0.80.64/32 <<
```