

성공하는 데이터센터 네트워크 운영자의 7가지 습관



PACKET PUSHERS 백서

목차

소개	3
7가지 습관	4
1. 비즈니스 결과를 고려한 설계	4
2. 안정적이고 반복 가능한 자동화	4
3. 한 번 습득 후 자주 사용	5
4. 적절한 장비 선택	5
5. 지속적인 검증	6
6. 사후 대응이 아닌 사전 예방적 조치	6
7. 작업 문서화	7
Apstra 및 IBN 개요	7
핵심 원칙	8
Apstra 구성 요소	9

소개

데이터센터 네트워크 엔지니어가 된다는 것은 복잡성과 불확실성을 안고 살아가는 것입니다. 한 곳의 변경이 다른 곳에 의도하지 않은 영향을 미칠 수 있습니다. 성능이 저하되거나 워크로드가 노출되거나 네트워크 작동이 중단될 수 있습니다.

네트워크 요구 사항이 끊임없이 변하는 불확실한 환경에서 네트워크를 성공적으로 운영하기 위해서는 네트워크 엔지니어가 좋은 습관을 들여야 합니다. 숙련된 네트워크 엔지니어는 신중하고 분별력 있고 꼼꼼합니다. 대개 이들은 무언가를 변경하기 전에 SNMP 트랩, Syslog, 디바이스 구성, 패킷 캡처 및 스트리밍 텔레메트리를 통해 네트워크에 대한 정보를 확인합니다. 그리고 문제가 발생할 경우 문제를 찾고 진단하고 해결하는 프로세스를 갖추고 있습니다.

이 같은 일반적인 관행이 엔지니어에게 도움이 될 수 있지만, 본 백서에서는 네트워크 엔지니어가 보다 효과적으로 데이터센터 관리와 운영상의 불확실성에 대처하도록 돕는 7가지 구체적인 습관을 아래와 같이 제안합니다.

이러한 습관을 일련의 모범 사례로 단독 적용할 수도 있으나, IBN(Intent-Based Networking: 인텐트 기반 네트워킹) 기반 소프트웨어 플랫폼인 주니퍼 Apstra를 통해 보다 쉽게 적용할 수 있습니다. IBN은 데이터센터 네트워크에서 안정적인 자동화 및 오케스트레이션을 구현합니다. 멀티벤더 환경에서 작동하는 Apstra 소프트웨어는 변경에 대한 자동화, 오케스트레이션, 검증을 통해 결과가 비즈니스 인텐트와 일치하도록 합니다.

7가지 습관은 다음과 같습니다.

1. 비즈니스 결과를 고려한 설계
2. 안정적이고 반복 가능한 자동화
3. 한 번 습득 후 자주 사용
4. 적절한 장비 선택
5. 지속적인 검증
6. 사후 대응이 아닌 사전 예방적 조치
7. 작업 문서화

이 백서에서는 성공하는 데이터센터 네트워크 운영자의 7가지 습관을 살펴보고, Apstra가 이러한 습관을 적용하는 데 어떻게 도움을 주는지 알아봅니다.

7가지 습관

1. 비즈니스 결과를 고려한 설계

데이터센터 네트워크가 존재하는 이유는 비즈니스를 이끄는 애플리케이션과 서비스를 지원하기 위해서입니다. 그러나 일반적으로 데이터센터를 설계할 때 원하는 결과를 고려하지 않고 먼저 벤더를 선택하는 경향이 있습니다. 벤더 선택은 종종 비즈니스 목표 이외의 요인에 의해 영향을 받습니다. 영업 담당자와 근사한 식사를 많이 한 임원이 원하는 제품과 벤더 인증에 시간과 비용을 투입한 엔지니어링 직원이 원하는 제품은 서로 다를 수 있습니다.

제품마다 작지만 본질적인 차이가 있을 수 있습니다. 어떤 벤더의 코드에는 조직이 원하거나 필요로 하지 않지만 여전히 유지 보수하고 업데이트해야 하는 부가 기능이 많을 수 있습니다. 어떤 벤더의 하드웨어 사양은 적합하지만 네트워크 OS에 버그가 있을 수 있습니다. 소프트웨어에 핵심 프로토콜이 구현된 방식에 따라 소프트웨어를 작동하는 데 엔지니어의 추가 작업이 필요할 수도 있습니다.

이러한 경우 제품의 특성과 제약 조건에 맞추어 실제 비즈니스 애플리케이션의 요구 사항을 조정해야 합니다. 따라서 운영상의 복잡성과 문제 발생 위험이 높아집니다. 그리고 네트워크가 새로운 애플리케이션과 서비스를 지원하는 속도가 느려져 네트워크 병목 현상이 발생합니다.

더 나은 습관은 비즈니스 결과를 먼저 고려하여 그러한 결과를 중심으로 네트워크를 설계하는 것입니다. Apstra는 인텐트에 집중함으로써 결과를 우선시합니다. 디바이스 수준에서 작동하여 결과를 선언적으로 구현하며, 네트워크 상태를 지속적으로 모니터링하고 검증하여 결과가 달성되도록 합니다.

또한 Apstra는 광범위한 네트워크 하드웨어와 소프트웨어를 지원하기 때문에 조직에서 벤더의 특성에 따라 설계를 결정하는 대신 비즈니스에 가장 적합한 설계를 선택할 수 있습니다.

2. 안정적이고 반복 가능한 자동화

네트워킹 업계는 수십 년간 자동화에 대해 이야기해 왔지만, 대부분의 기업은 엔지니어가 마치 다용도 주머니칼처럼 부분적인 작업에 간편하게 사용하고 있는 자체 제작 스크립트에서 아직 벗어나지 못하고 있습니다.

사실 대부분의 엔터프라이즈 데이터센터는 여러 디바이스와 서비스에 걸쳐 조정되는 광범위하고 안정적인 자동화를 지원하기에 역부족입니다. 한 가지 이유는 자동화된 프로세스가 일련의 의도하지 않은 이벤트를 촉발시켜 전체 네트워크를 중단시킬 수 있기 때문입니다. 게다가 설정 후에 구성이 기준선에서 벗어나는 경우도 드문 일이 아닙니다. 스크립트와 플레이북을 작성하는 엔지니어는 네트워크 디바이스에서 실행되는 다양한 소프트웨어 버전의 미세한 차이에 따라 지속적으로 코드를 조정해야 합니다. |

또한 네트워크 엔지니어는 광범위한 자동화를 지원하는 데 핵심적인 요소를 이용하지 못하고 있습니다. 여기에는 네트워크 상태에 대한 심층적인 가시성, 필요한 디바이스 구성에 대한 신뢰할 수 있는 정보 소스, 운영 환경에 적용하기 전에 변경 사항을 테스트하는 기능, 프로세스 결과를 검증하는 메커니즘이 포함됩니다.

Apstra는 의도하지 않은 문제를 방지하고 변경 사항이 운영자의 인텐트와 일치하도록 보장하는 보호책을 통해 안정적이고 반복 가능한 자동화를 지원합니다. 예를 들어, 명령의 오타이든 기존 정책을 위반하는 구성 변경이든 엔지니어가 인텐트와 일치하지 않게 구성을 변경하려고 할 경우 Apstra에서 엔지니어에게 이를 알리고 변경을 막습니다.

Apstra는 전체 네트워크 상태를 그래프 데이터베이스(아래에 설명됨)에 저장하여 변경 사항을 운영 환경에 적용하기 전에 잠재적인 문제를 식별할 수 있기 때문에 이것이 가능합니다. 이처럼 Apstra는 의도하지 않은 결과를 방지하며 예측 가능하고 검증된 방식으로 일반적인 변경이 수행되도록 함으로써 변경 관리를 효율화합니다.

또한 변경으로 인해 문제가 발생할 경우 디바이스를 알려진 양호한 상태로 롤백하는 기능도 Apstra에 포함되어 있습니다. 이를 통해 반복 가능하고 안정적인 자동화 프로세스가 구현됩니다.

3. 한 번 습득 후 자주 사용

가끔 사용하는 비즈니스 애플리케이션의 경우 특정 작업을 위해 다시 사용할 때마다 사용 방법을 재차 익히느라 시간을 낭비하게 됩니다.

관리, 모니터링 및 자동화 도구도 마찬가지로 일상적인 워크플로우의 일부로 사용하지 않는다면 인터페이스를 여기저기 찾아다니다가 작업이 지연됩니다. 도구를 잘 익히고 최대한 활용하는 것이 좋은 습관입니다.

Apstra는 며칠 만에 간단히 익혀서 사용할 수 있습니다. 아키텍처 및 운영상의 큰 변화로 인해 익히는 데 몇 개월이 걸리는 다른 솔루션에 비해 유리합니다. 또한 모니터링, 구성, 분석을 포함한 다양한 기능을 제공하기 때문에 네트워크 엔지니어가 소프트웨어를 빠르게 익힌 다음 일상적으로 사용하여 유용한 도구를 숙달할 수 있습니다.

아울러 Apstra는 멀티벤더 환경에서 작동할 수 있기 때문에 각 NOS 및 벤더 프로토콜 구현의 개별적인 특성에 구애를 받지 않습니다. 즉, 네트워크 엔지니어가 각 NOS의 미세한 차이를 잘 알지 못하더라도 작업을 완료할 수 있습니다.

4. 적절한 장비 선택

특정 장비의 작동 방식에 맞추어 작업을 조정하기보다 필요한 작업에 적합한 장비를 선택하는 것이 현명한 습관입니다.

Apstra의 멀티벤더 전략을 통해 네트워크 팀은 기업의 목표를 지원하는 데 적합한 하드웨어와 소프트웨어를 유연하게 구축할 수 있습니다. 또한 기업에서 유리한 구매 조건을 이용하고 벤더 종속성을 피할 수 있습니다. 다양한 공급업체의 장비를 수급할 수 있기 때문에 공급망 제약이 있는 현재와 같은 상황에서 곤란을 피할 수 있습니다.

5. 지속적인 검증

테크 소셜 미디어에는 부주의한 CLI 명령, 잘못된 구성, 오타 등으로 인해 온갖 흥미롭고 힘든 결과가 발생한 일화가 넘쳐납니다. 사람은 실수를 하기 마련이므로 성공하는 엔지니어는 운영 환경에서 작업할 때 작업물을 철저히 점검합니다.

이러한 점검은 다양한 형태로 이루어질 수 있습니다. 일부 네트워크 OS는 입력되는 명령 문자열의 각 단어를 점검하여 오타가 있을 경우 엔지니어에게 경고해 줍니다. 또한 엔지니어는 구성에 문제가 없는지 확인해 주도록 동료에게 요청하거나 스크립트를 리포지토리에 업로드하여 다른 이들의 검토를 받을 수 있습니다. ITIL 기반 변경 관리 프로세스를 사용하는 조직도 있습니다.

Apstra는 디바이스 변경 사항을 지속적으로 검증하여 변경 사항이 인텐트와 일치하는지 확인함으로써 이 같은 좋은 습관을 적용합니다. 예를 들어, 새 VLAN을 생성한 후 태그가 지정되지 않은 기존 VLAN이 있는 인터페이스에 실수로 추가할 경우 Apstra가 오류를 반송할 뿐만 아니라 그 이유도 보여줍니다.

지속적인 검증 기능은 사람의 실수를 줄이는 동시에 새로운 변경 사항이 원하는 인텐트에 맞는지, 즉 연결 가능성, 성능, 보안 및 규정 준수와 관련된 비즈니스 정의 파라미터에 부합되는지 확인합니다.

6. 사후 대응이 아닌 사전 예방적 조치

작은 문제가 큰 문제가 되기 전에 해결하는 것이 좋은 습관입니다. 예를 들어 스위치 포트가 간헐적으로 작동 중지될 경우 완전히 못 쓰게 되기 전에 근본 원인을 찾아 해결하는 것이 좋습니다.

문제 해결을 미룰 경우 무엇이 잘못되었는지 알아내려고 할 때 수많은 경고와 로그(그리고 사용자 불만 텍스트 메시지)를 확인해야 합니다. 케이블 접속 불량이나 문제라면 다행이지만, 네트워크 카드나 옵티컬 모듈에 문제가 있을 경우 예비 부품이 있어야 합니다.

Apstra는 분석 기능을 통해 사전 예방적 운영을 지원합니다. 물리적 디바이스 상태를 모니터링하여 즉시 해결하지 않으면 더 큰 문제가 될 수 있는 이상 징후를 경고합니다. 또한 대역폭 문제, 사용률 및 전체 용량을 모니터링하여 미리 알려줌으로써 엔지니어가 난처한 입장에 처하지 않고 증가하는 요구 사항을 원활하고 효율적으로 충족할 수 있도록 합니다.

7. 작업 문서화

야채를 섭취하고 규칙적으로 운동하는 것과 같이 작업을 문서화하는 것(예: 네트워크 다이어그램 업데이트, 구성 변경 사항에 주석 추가)이 좋은 습관이라는 점은 누구나 알고 있습니다. 문서는 무엇을 왜 했는지 보여주며, 일어난 일에 대한 기록을 제공합니다. 이 기록은 이후의 새로운 변경, 문제 해결, 감사 등에 유용할 수 있습니다.

엔지니어가 다른 역할을 맡거나 다른 곳에서 일하게 될 수 있으므로 문서화는 중요합니다. 떠나는 엔지니어가 가지고 있던 많은 운영 및 관리 지식을 문서로 정리하면 조직에서 그러한 지식을 보유하고 다른 운영자에게 전해줄 수 있습니다.

문제는 작업을 문서화하는 것이 지루하고 많은 시간을 소모한다는 것입니다. 엔지니어가 까다로운 신규 구축이나 위기 대응을 위해 애쓰고 있을 때 문서화는 가장 뒷전으로 밀리는 경우가 많습니다. 이 목록의 모든 습관 중에서 정기적으로 신뢰성 있게 문서화하는 습관이 가장 들이기 어려울 수 있습니다.

Apstra는 기본적으로 자체 문서화를 수행하도록 설계되어 이 습관을 지원할 수 있습니다. 지속적으로 개별 디바이스에서 텔레메트리를 수집하고 전체 네트워크 상태를 유지하며, 변경 사항과 버전 기록을 저장하여 이전에 어떤 변경 사항 또는 업데이트가 이루어졌고 결과가 어떤지를 확인할 수 있도록 합니다. 과거를 들여다볼 수 있는 일종의 타임머신 같은 역할을 하는 것입니다.

또한 앞서 언급했듯이 Apstra는 필요한 경우 네트워크를 알려진 양호한 상태로 롤백할 수 있습니다.

Apstra 및 IBN 개요

Apstra는 IBN(Intent-Based Networking: 인텐트 기반 네트워킹) 개념을 중심으로 설계되었습니다. Apstra는 비즈니스 레벨의 인텐트(즉 기대 결과)로 시작하여 이러한 인텐트를 기반으로 해당 결과를 달성하는 데 필요한 디바이스 레벨의 구성을 생성합니다.

비즈니스 인텐트에는 일련의 애플리케이션에 특정 서비스 수준을 보장하는 것과 같은 일반적인 결과와 패브릭 구축, VLAN 설정, 올바른 포트 연결 또는 액세스 정책 적용과 같은 특정 결과가 포함됩니다.

Apstra는 전반적인 네트워크 상태와 개별 네트워크 디바이스를 지속적으로 모니터링하여 네트워크가 항상 조직의 인텐트를 충족하도록 보장합니다. 패킷 손실, 정체 또는 인터페이스 문제와 같이 인텐트를 위반하는 이벤트가 발생할 경우 Apstra에서 엔지니어에게 알리고 문제에 대한 세부 정보를 제공할 수 있습니다. 때때로 Apstra는 다음 권장 단계를 제시하거나 적절한 경우 문제를 자동으로 해결할 수 있습니다.

엔지니어가 이전 인텐트와 충돌하는 구성 변경을 시도할 경우 Apstra에서 사전 변경 분석을 제공하여 엔지니어에게 경고함으로써 의도하지 않은 결과를 방지합니다. 새로운 네트워크 서비스가 데이터센터에 추가될 때 Apstra는 해당 서비스를 지원하는 데 필요한 네트워크 구성이 기존 인텐트와 충돌하지 않도록 보장합니다.

핵심 원칙

IBN을 구현하기 위해 Apstra는 레퍼런스 설계, 멀티벤더 지원, 지속적인 검증, 통합을 포함한 핵심 원칙에 기반을 두고 있습니다.

레퍼런스 설계: 데이터센터가 복잡해지는 주요 이유는 데이터센터 네트워크가 종종 장기적 전략 없이 구성되기 때문입니다. 즉각적인 비즈니스 요구 사항을 해결하기 위해 맞춤 구성된 특수한 구현이 적용되며, 시간이 지남에 따라 복잡한 구성이 뒤섞여 자동화하기 어려워지고 많은 시간을 소모하는 수동 관리 및 유지 보수가 필요하게 됩니다. 엔지니어 간에 일부 지식이 구두로 전달될 뿐, 이러한 네트워크에 대한 문서화가 거의 또는 전혀 이루어지지 않는 경우가 많습니다. 기술적인 부담이 누적되면서 엔지니어가 단지 비즈니스 운영을 유지하기 위해 임기응변 방식을 사용해야 할 수 있습니다.

Apstra는 한정된 레퍼런스 설계를 사용하여 무결한 상태에서 시작합니다. 레퍼런스 설계는 스위칭 하드웨어와 케이블 연결을 포함한 물리적 인프라에 대한 템플릿을 나타냅니다. 대표적인 레퍼런스 설계는 리프 스파인 또는 Clos 네트워크입니다.

일부 엔지니어는 레퍼런스 설계 또는 템플릿이 제약을 준다고 느낄 수 있지만 이 접근 방식은 업계 모범 사례를 따라 일반적인 데이터센터 네트워크에 만연한 일회성, 임시방편 및 Snowflake 구현에서 발생하는 문제를 크게 줄입니다. 단일 정보 소스(SSOT)와 변경 제어를 위해 이러한 수준의 규율과 엄격한 지침을 따라야 합니다.

Apstra는 레퍼런스 청사진을 기반으로 전체 설계-구축-운영 수명주기에 걸친 자동화를 지원하며, 코드로 구성을 구현하고 버전 제어를 적용하며 종속성을 검증하는 기능을 제공합니다. 접근 방식을 통해 기업에서는 자체 전문 인력이 많지 않아도 클라우드 프로바이더의 검증된 자동화 기술을 활용할 수 있습니다.

멀티벤더 지원: Apstra는 Cisco, Artista, Dell 및 주니퍼를 비롯한 점점 더 많은 하드웨어 벤더와 화이트박스 하드웨어 제조업체를 지원합니다. 또한 상용 NOS에서 SONiC과 같은 오픈 소스 옵션에 이르기까지 다양한 네트워크 OS를 지원합니다.

이러한 멀티벤더 지원을 통해 조직에서는 비즈니스 및 기술 요구 사항을 충족하는 동시에 엔지니어의 지식과 기술 수준에 맞는 디바이스를 갖출 수 있습니다. 또한 유리한 가격 조건, 공급망 다양성, 운영 요구 충족을 위해 랙이나 포드 전반에서 여러 벤더와 NOS를 혼합하여 사용하고자 하는 고객에게 더 많은 옵션과 유연성을 제공합니다.

폐루프 검증(Closed-Loop Validation): Apstra는 단일 정보 소스를 기반으로 폐루프 검증을 통해 인텐트가 달성되도록 보장합니다. 폐루프 검증은 변경 또는 업데이트가 실제로 이루어졌는지 그리고 변경 또는 업데이트 결과가 인텐트와 일치하는지 확인합니다. 이에 반해, 간단한 자동화 스크립트를 사용하면 일련의 명령을 자동화할 수는 있지만 단지 스크립트가 실행된 후 중지될 뿐이며 변경이 실제로 이루어졌는지 또는 변경으로 인해 원하는 결과가 나왔는지를 파악할 수 없습니다.

폐루프 검증은 자동화된 프로세스가 실제로 비즈니스 요구 사항에 맞는 결과를 생성했는지 그리고 네트워크가 계속 원하는 상태로 운영되는지 확인함으로써 안정적인 자동화를 지원하며, 이러한 안정성을 바탕으로 자동화된 시스템에 대한 네트워크 팀의 신뢰를 불러일으킵니다.

통합: Apstra는 VMware NSX-T 및 VMware vSphere와 긴밀하게 통합되어 서버 팀과 네트워킹 팀 간의 운영을 효율화합니다. Apstra는 물리적 언더레이와 가상 오버레이 모두에서 엔드 투 엔드 가시성을 제공할 뿐만 아니라, 오버레이의 변경 사항으로 인해 언더레이의 조정이 필요한 경우(예: 패브릭에 구성된 MTU 크기가 오버레이 요구 사항에 맞지 않는 경우)에 이를 식별합니다. 또한 이상 징후와 잘못된 구성을 감지하고 문제를 해결합니다.

이러한 방식으로 언더레이가 지속적인 워크로드 연결 및 볼륨 변화에 대응하여 최적화됩니다. 또한 문제가 발생할 경우 Apstra 소프트웨어는 근본 원인이 언더레이에 있는지 아니면 가상 오버레이에 있는지를 빠르게 식별할 수 있습니다. Apstra는 문제 해결에 소요되는 시간을 단축함으로써 조직이 평균 문제 해결 시간과 운영 비용을 크게 줄일 수 있도록 돕습니다.

아울러 개방형 API가 ServiceNow, 챗봇, Slack과 같은 일반적인 워크플로우 도구와의 통합을 지원합니다.

Apstra 구성 요소

Apstra 제품은 세 가지 소프트웨어 요소, 즉 디바이스 에이전트, 데이터 저장소, 그래프 데이터베이스로 이루어집니다. 이러한 요소가 함께 작동하여 인텐트 기반 네트워킹을 제공합니다. 각 요소를 간략히 소개하면 다음과 같습니다.

에이전트(Agents): Apstra는 물리적 및 가상 네트워크 디바이스의 디바이스 에이전트를 사용하여 디바이스를 구성하고 텔레메트리 데이터를 Apstra의 데이터 저장소로 보냅니다. 타사 에이전트를 실행할 수 없는 네트워크 디바이스의 경우 오프박스 에이전트를 Apstra 서버에서 Linux 컨테이너로 실행할 수 있습니다. 이 오프박스 에이전트는 SSH 또는 API를 통해 네트워크 디바이스에서 상태 정보를 가져올 수 있습니다.

데이터 저장소(Data Store): 데이터 저장소는 서버에서 실행되며 에이전트 텔레메트리, 네트워크 설계 세부 정보, 네트워크 이상 징후 및 기타 데이터를 수집합니다. 사용자 인텐트도 그래프 데이터베이스와 함께 데이터 저장소에 유지됩니다.

그래프 데이터베이스(Graph Database): Apstra는 그래프 데이터베이스에 데이터센터 네트워크의 모든 요소 또는 객체와 모든 네트워크 구성을 나타냅니다. 이 방법으로 Apstra는 네트워크 상태를 "파악"합니다. 그래프 데이터베이스는 네트워크의 가상 모델 역할을 하며 여러 팀을 위한 단일 정보 소스입니다. 이 모델은 네트워크 및 디바이스 텔레메트리가 지속적으로 제공되어 실시간 최신 상태로 유지됩니다. 사용자 인텐트와 이 모델을 비교함으로써 네트워크 및 디바이스 구성이 적절한 결과를 내는지 확인됩니다.