



디지털 트랜스포메이션을 위한 IT 인프라의 클라우드 전환과 최적화 전략

F5 NETWORKS KOREA

신기욱 상무

디지털 Transformation

장점을 실현하기 위해서는 사람, 프로세스, 그리고 시스템 전환이 필요합니다.

사람

최신화수용, DevOps능력,
Cross-functional Team

프로세스

Application의 개발, 배포 및
전달 방법 변경

시스템

Application 자체가 새로운
디지털 경제의 토대가 됨

69%의 기관은 Digital Transformation을 실현하고 있습니다.

디지털 Transformation

장점을 실현하기 위해서는 사람, 프로세스, 그리고 시스템 전환이 필요합니다.

사람

최신화수용, DevOps능력, Cross-functional Team

48% 는 다양한 업무를 수행

43% 는 자동화된 app services 보유

35% 는 Self-Service Provisioning 수행

프로세스

Application의 개발, 배포 및 전달 방법 변경

52% 앱을 개발하는 방법의 변경

62% 앱을 배포하는 방법의 변경

48% 앱을 전달하는 방법의 변경

시스템

Application 자체가 새로운 디지털 경제의 토대가 됨

42% 새로운 아키텍처 탐구

56% Container를 지금 사용 중

58% Cloud 결정은 LOB (Line Of Business)와 함께, 또는 LOB에 의해.

69% 의 기관은 Digital Transformation을 실현하고 있습니다.

멀티 클라우드 전략이 필요합니다.

APP-CENTRIC 비즈니스는 MULTI-CLOUD 전략을 요구하고 있습니다.

전략 필요성

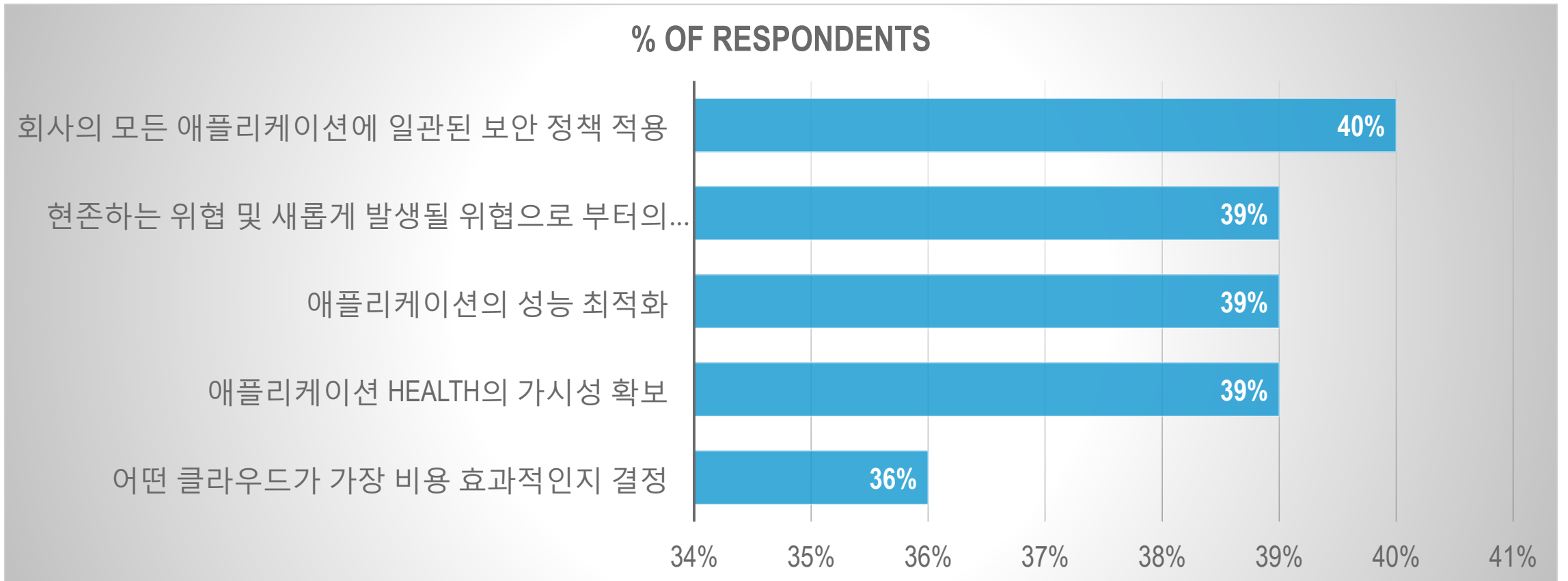
- Multi-cloud 는 실험적인 상황에서 전략적 모델로 변화했습니다.
- 디지털 경제는 Applications의 환경 전체를 변화시키고 있습니다.
- Automation과 Orchestrating은 디지털 트랜스포메이션을 수행하기 위한 핵심 요소입니다.
- 현재 설치 운영되는 상위 Top 5개의 Application services는 작년과 비슷하지만, 새로운 Application Service는 계속 발생 되고 있습니다.

Source: State of Application Services

Release date | 2019.01.15

멀티 클라우드 도전과제입니다.

남아 있는 TOP 도전 과제들: 모든 애플리케이션에 일관된 보안 정책 적용 및 수행



Source: State of Application Services - Release date | 2019.01.15



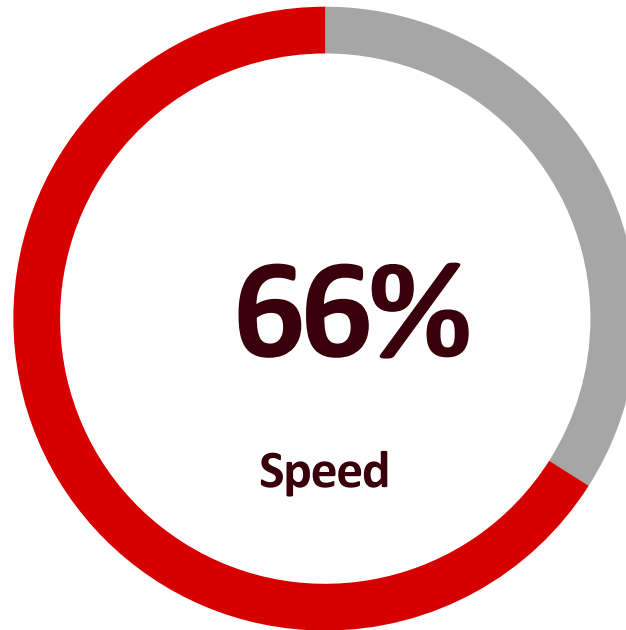
첫 번째: Automation and Orchestration

NetOps! DevOps를 만나면...

DEVOPS의 AUTOMATION을 통한 최대의 이점



IT부서의 서비스
생산성 향상



서비스 환경으로의 전환
속도가 개선



휴먼 에러의 비율을
줄임

Source: NetOps Meets DevOps: The State of Network Automation – F5 & Red Hat, July 2018

DevOps Tools 입니다.

예) <https://xebialabs.com/periodic-table-of-devops-tools>

PERIODIC TABLE OF DEVOPS TOOLS (V3) EMBED DOWNLOAD ADD

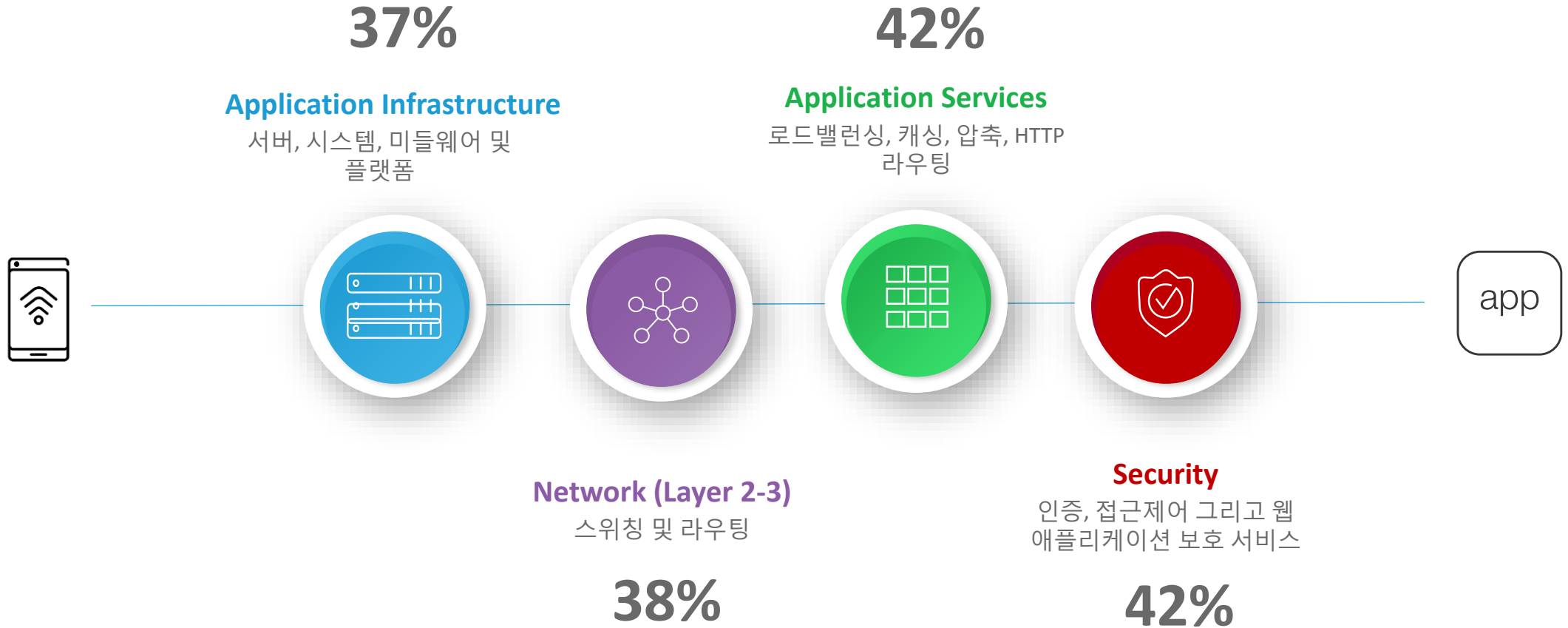
| | | | | | | | | | | | | | | | | | |
|---|---|---------------------------------|--|--------------------------------|--|--|--|---------------------------------|----------------------------------|---|---|--|---|---|---|---|----------------------------------|
| 1 Os Gl GitLab | | | | | | | | | | | | | | | 2 En Sp Splunk | | |
| 3 Fm Gh GitHub | 4 En Dt Datical | | | | | | | | | | | | | | | | |
| 11 Os Sv Subversion | 12 En Db DBMaestro | | | | | | | | | | | | | | | | |
| 19 En Cw ISPW | 20 En Dp Delphix | 21 Os Jn Jenkins | 22 Fm Cs Codeship | 23 Os Fn FitNesse | 24 Fr Ju JUnit | 25 Fr Ka Karma | 26 Os Su SoapUI | 27 En Ch Chef | 28 Fr Tf Terraform | 29 En Xld Xebialabs XL Deploy | 30 En Ud UrbanCode Deploy | 31 Os Ku Kubernetes | 32 Fm Cc CA CD Director | 33 En Pr Plutora Release | 34 Pd Al Alibaba Cloud | 35 Os Os OpenStack | 36 Os Ps Prometheus |
| 37 Os At Artifactory | 38 En Rg Redgate | 39 Pd Ba Bamboo | 40 Fm Vs VSTS | 41 Fr Se Selenium | 42 Fr Jm JMeter | 43 Os Ja Jasmine | 44 Pd Sl Sauce Labs | 45 Os An Ansible | 46 Os Ru Rudder | 47 En Oc Octopus Deploy | 48 Os Go GoCD | 49 Os Ms Mesos | 50 Pd Gke GKE | 51 Fm Om OpenMake | 52 Pd Cp AWS CodePipeline | 53 Os Cy Cloud Foundry | 54 En It ITRS |
| 55 Os Nx Nexus | 56 Os Fw Flyway | 57 Os Tr Travis CI | 58 Fm Tc TeamCity | 59 Os Ga Gatling | 60 Fr Tn TestNG | 61 Fm Tt Tricentis Tosca | 62 Pd Pe Perfecto | 63 En Pu Puppet | 64 Os Pa Packer | 65 Fm Cd AWS CodeDeploy | 66 En Ec ElectricCloud | 67 Os Ra Rancher | 68 Pd Aks AKS | 69 Os Rk Rkt | 70 Os Sp Spinnaker | 71 Pd Ir Iron.io | 72 Pd Mg Moogsoft |
| 73 Fm Bb BitBucket | 74 En Pf Perforce HelixCore | 75 Fm Cr Circle CI | 76 Pd Cb AWS CodeBuild | 77 Fr Cu Cucumber | 78 Os Mc Mocha | 79 Os Lo Locust.io | 80 En Mf Micro Focus UFT | 81 Os Sl Salt | 82 Os Ce CFEngine | 83 En Eb ElasticBox | 84 En Ca CA Automic | 85 En De Docker Enterprise | 86 Pd Ae AWS ECS | 87 Fm Cf Codefresh | 88 Os Hm Helm | 89 Os Aw Apache OpenWhisk | 90 Os Ls Logstash |
| 91 En Xli Xebialabs XL Impact | 92 Os Ki Kibana | 93 Fm Nr New Relic | 94 En Dt Dynatrace | 95 En Dd Datadog | 96 Fm Ad AppDynamic | 97 Os Ei ElasticSearch | 98 Os Ni Nagios | 99 Os Zb Zabbix | 100 En Zn Zenoss | 101 En Cx Checkmarx SAST | 102 En Sg Signal Sciences | 103 En Bd BlackDuck | 104 Os Sr SonarQube | 105 Os Hv HashiCorp Vault | | | |
| 106 En Sw ServiceNow | 107 Pd Jr Jira | 108 Fm Tl Trello | 109 Fm Sl Slack | 110 Fm St Stride | 111 En Cn CollabNet VersionOne | 112 En Ry Remedy | 113 En Ac Agile Central | 114 Pd Og OpsGenie | 115 Pd Pd Pagerduty | 116 Os Sn Snort | 117 Os Tw Tripwire | 118 En Ck CyberArk Conjur | 119 En Vc Veracode | 120 En Ff Fortify SCA | | | |

Legend:

- Os: Open Source
- Fr: Free
- Fm: Freemium
- Pd: Paid
- En: Enterprise
- Source Control Mgmt.
- Database Automation
- Continuous Integration
- Testing
- Configuration
- Deployment
- Containers
- Release Orchestration
- Cloud
- AI/Ops
- Analytics
- Monitoring
- Security
- Collaboration

Xebialabs Enterprise DevOps
Follow @xebialabs
Publication Guidelines
Download

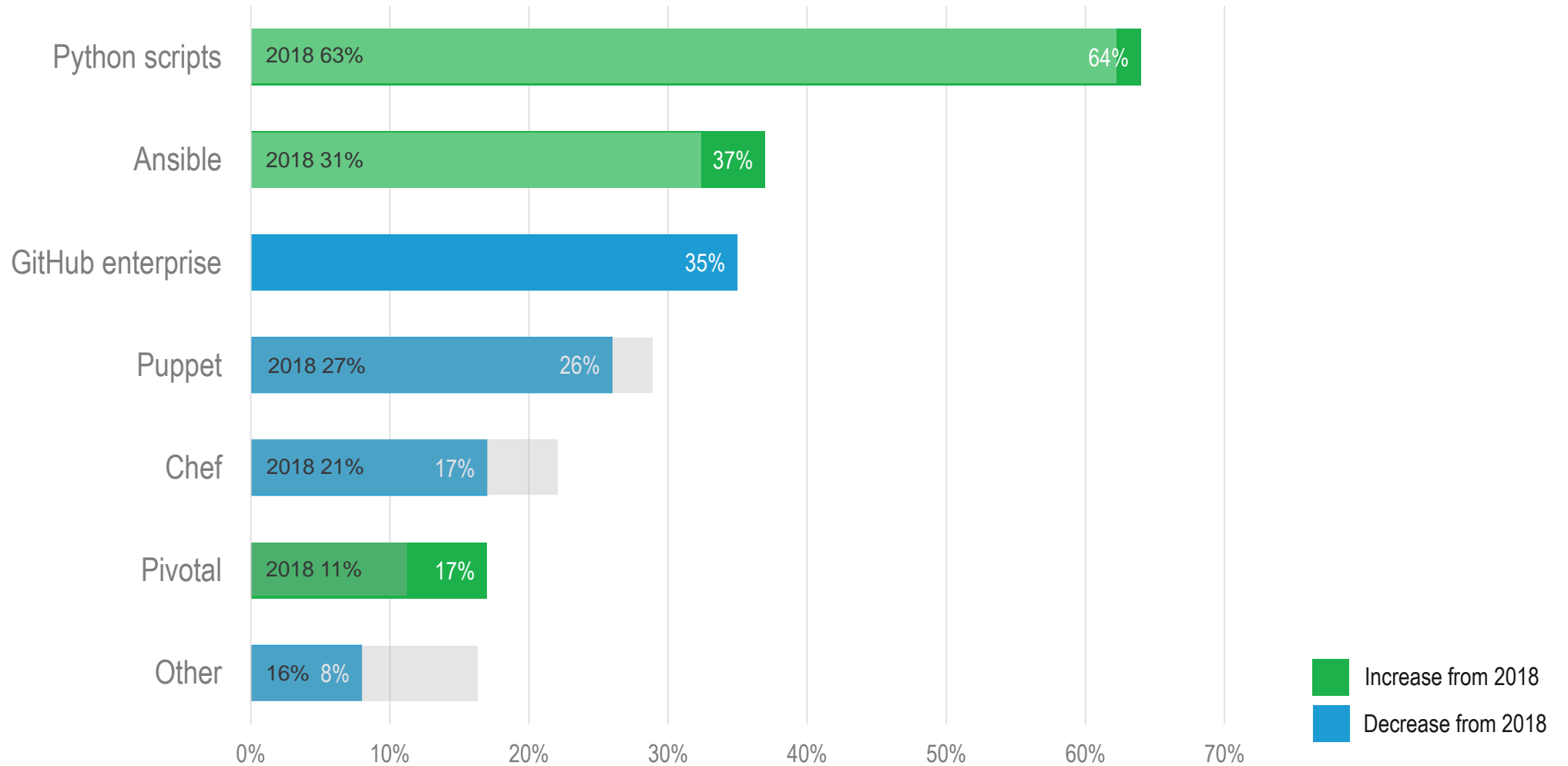
Automation의 현재입니다.



Source: State of Application Services - Release date | 2019.01.15

가장 많이 사용하는 Automation 방법입니다.

PYTHON과 ANSIBLE이 2018 VS 2019 가장 많이 사용하는 방법입니다.



Source: State of Application Services - Release date | 2019.01.15

Declarative Automation 솔루션이 필요합니다.

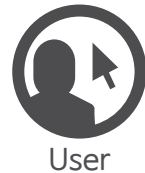


App Service 3 Extension (AS3)

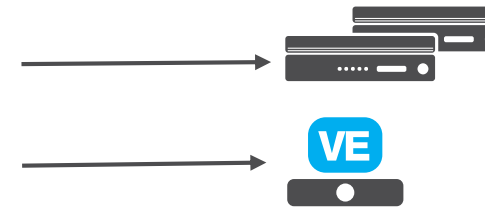
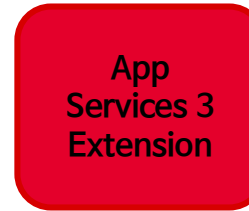
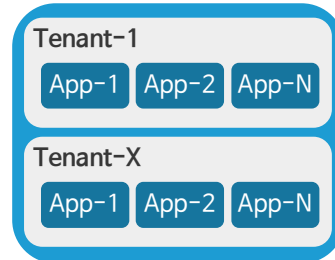
- Node.js 기반의 Single REST API Endpoint.

서비스들의 예제:

- HTTP
- SSL Offload
- URL Routing
- HTTP(s) w/ WAF
- Fast L4 TCP/UDP

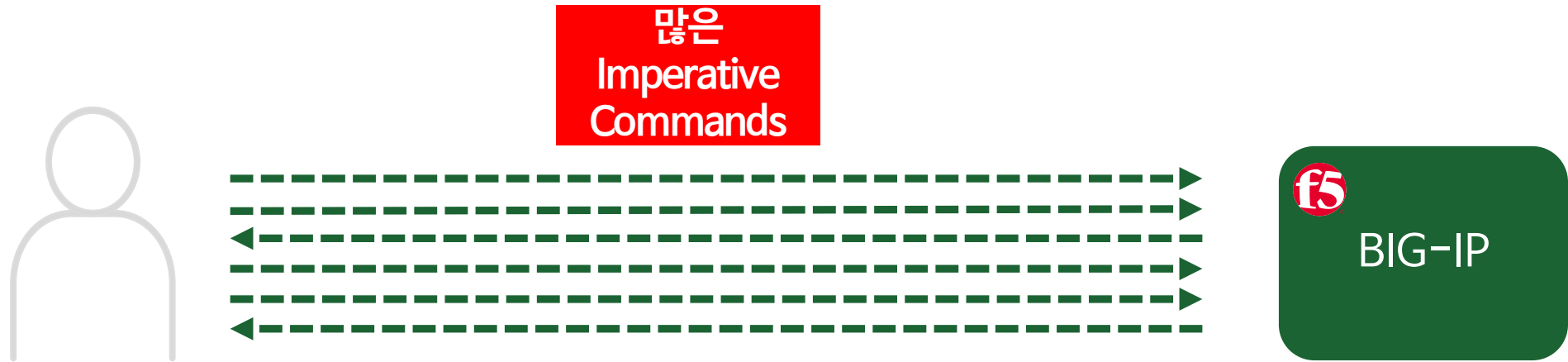


Declarative API



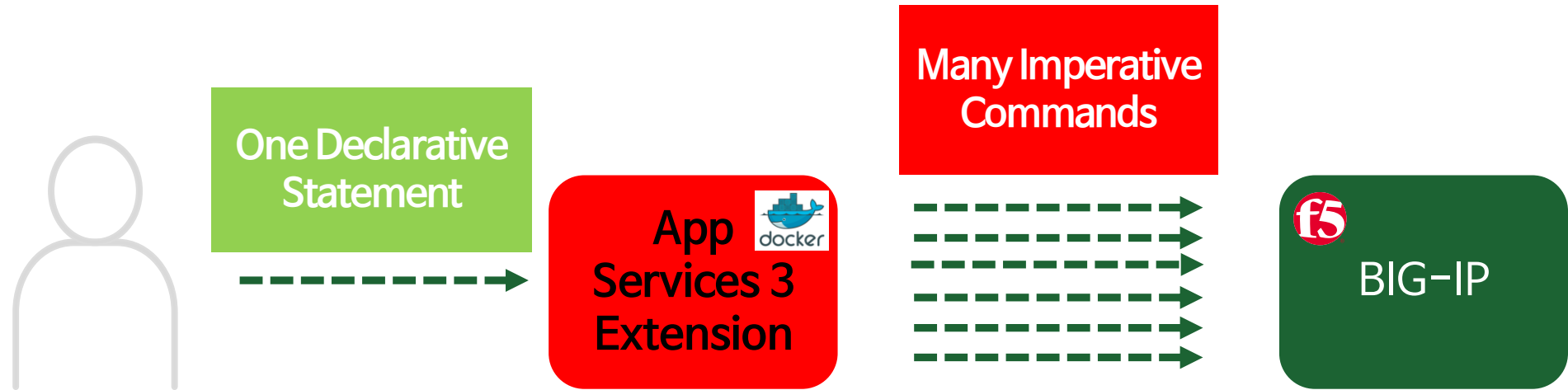
Available now: <https://github.com/F5Networks/f5-appsvcs-extension>

Imperative Automation vs Declarative Automation.



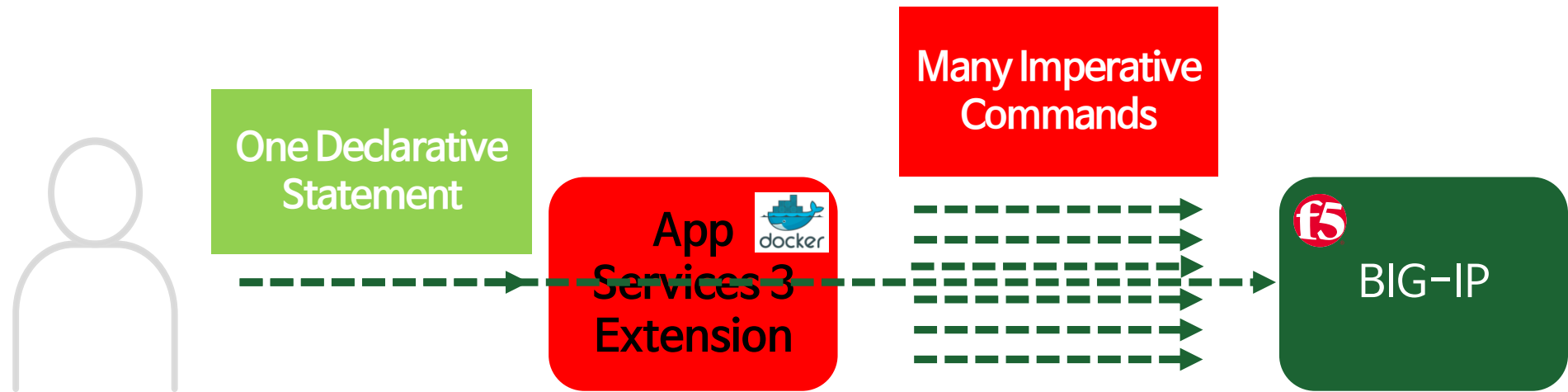
- 장비/솔루션에 대한 전문 지식 필요 및 많은 REST API 호출 필요
- 오케스트레이션 시스템에 통합하거나 자동화하는데 많은 관리 비용 필요
- 에러 발생 확률 상대적으로 높음

Imperative Automation vs Declarative Automation.



- F5 BIG-IP 장비의 설정에 대해서 전문 지식이 필요하지 않음
- 단일 REST-API 호출로 자동화 및 오케스트레이션 통합 단순화
- Declaration은 재사용 가능하고 일관성을 유지하며 오류를 줄일 수 있음

Imperative Automation vs Declarative Automation.

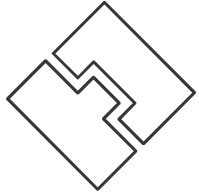


- F5 BIG-IP 장비의 설정에 대해서 전문 지식이 필요하지 않음
- 단일 REST-API 호출로 자동화 및 오케스트레이션 통합 단순화
- Declaration은 재사용 가능하고 일관성을 유지하며 오류를 줄일 수 있음

Ansible을 이용한 F5 자동화 솔루션입니다

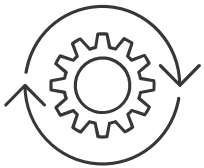


RED HAT[®]
ANSIBLE[®]



F5 제품군과 완벽하게 통합

- BIG-IP 애플리케이션 서비스 및 BIG-IQ 센트럴 매니저



F5 솔루션에 대한 Automation/Orchestration 제공

- 지속 가능한 자동화^{Automation} 및 설정 관리



No more hacky scripts

- Ad-hoc 셸 스크립트는 Automation 프레임워크를 대체할 수 없음

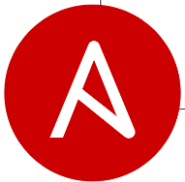
- 멀티클라우드 환경에 오케스트레이션 및 구성관리 간소화
- 애플리케이션 배포 프로세스의 모든 단계에 대한 자동화

F5가 제공하는 Programmability 입니다.

PROGRAMMABLE MANAGEMENT, CONTROL, AND DATA PLANES

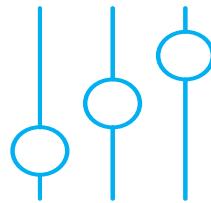
iControl

REST 와 SOAP.
User, Script, F5 devices간의
가볍고 빠른 상호작용



iRules

프로그램 방식의 실시간
애플리케이션 트래픽에
대한 액세스 제공

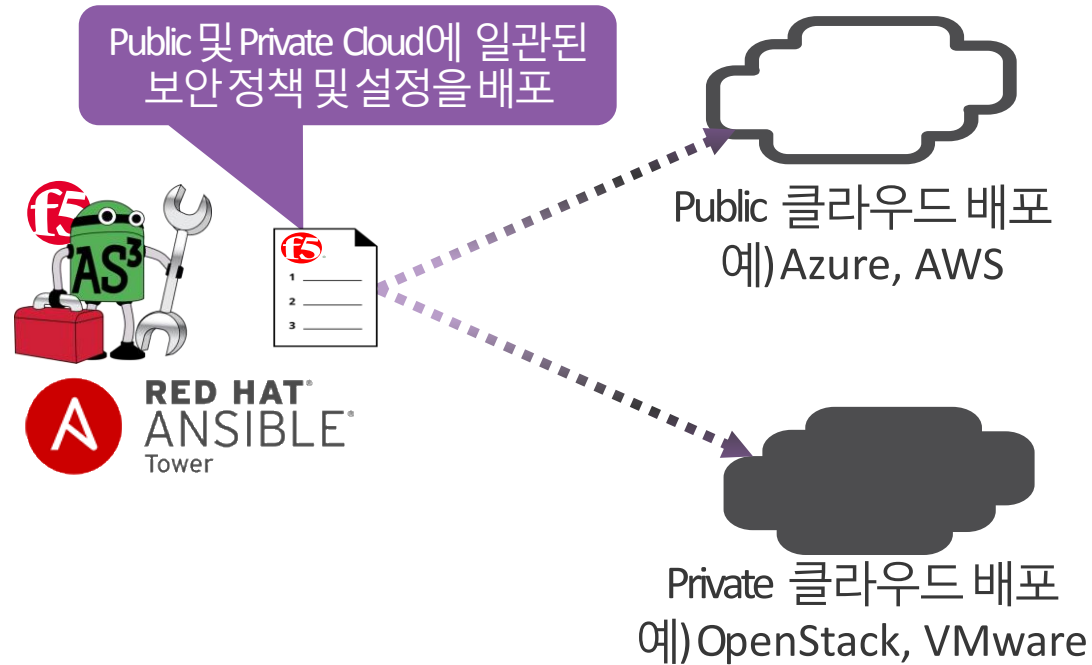


Templates

BIG-IP에서 Services 기반과
템플릿 기반의 설정 제공



AS3, Ansible을 통해 멀티 클라우드 환경에 일관된 보안 정책 적용 및 설정 관리가 가능합니다.

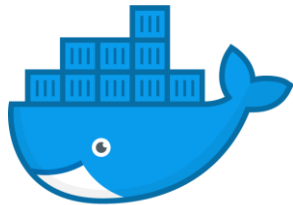


- 멀티 클라우드에서 모두 운영 가능한, 검증된 3rd party 보안 솔루션의 Automation 배포 적용을 통한 설정 및 운영 일원화. (한번 생성하여 멀티 클라우드에 적용가능)



두 번째: Container Architecture in Cloud

마이크로 서비스 문제를 해결하는 기술은 빠르게 발전하고 있습니다.



컨테이너는 Monolith →
Microservice 전환을 용의하게
함.

애플리케이션과 인프라간의
독립성 제공

예) Docker

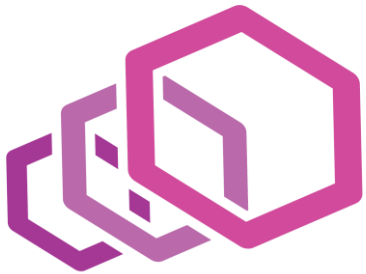


컨테이너 오케스트레이터 툴은
마이크로 서비스 생성(Build)과
구축(Deploy) 이슈를 해결함 →

로드분배(?), 라우팅(?),
관찰성(?)

예) Kubernetes

마이크로 서비스 문제를 해결하는 기술은 빠르게 발전하고 있습니다.



Data Plane Service Mesh는 서비스 Discovery, 로드분배, 라우팅 및 Observability을 포함한 Runtime 이슈들을 해결
예) Envoy

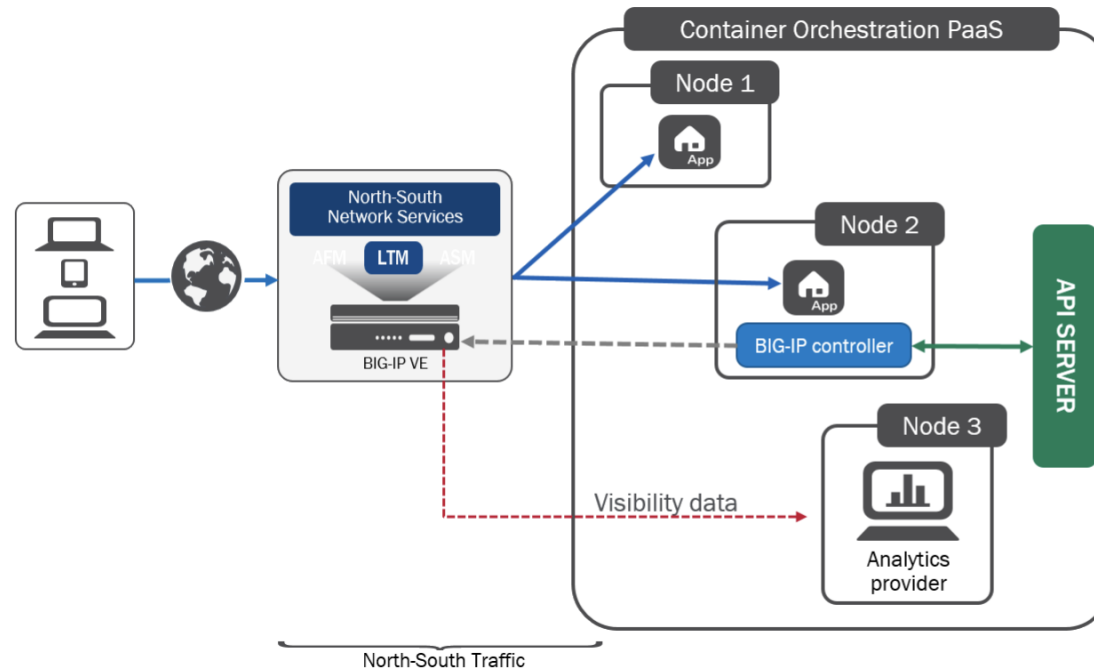


Control Plane Service Mesh는 모든 Data Plane들에, 정책과 설정을 제공 → 분산된 시스템으로의 전환 가능
예) Istio



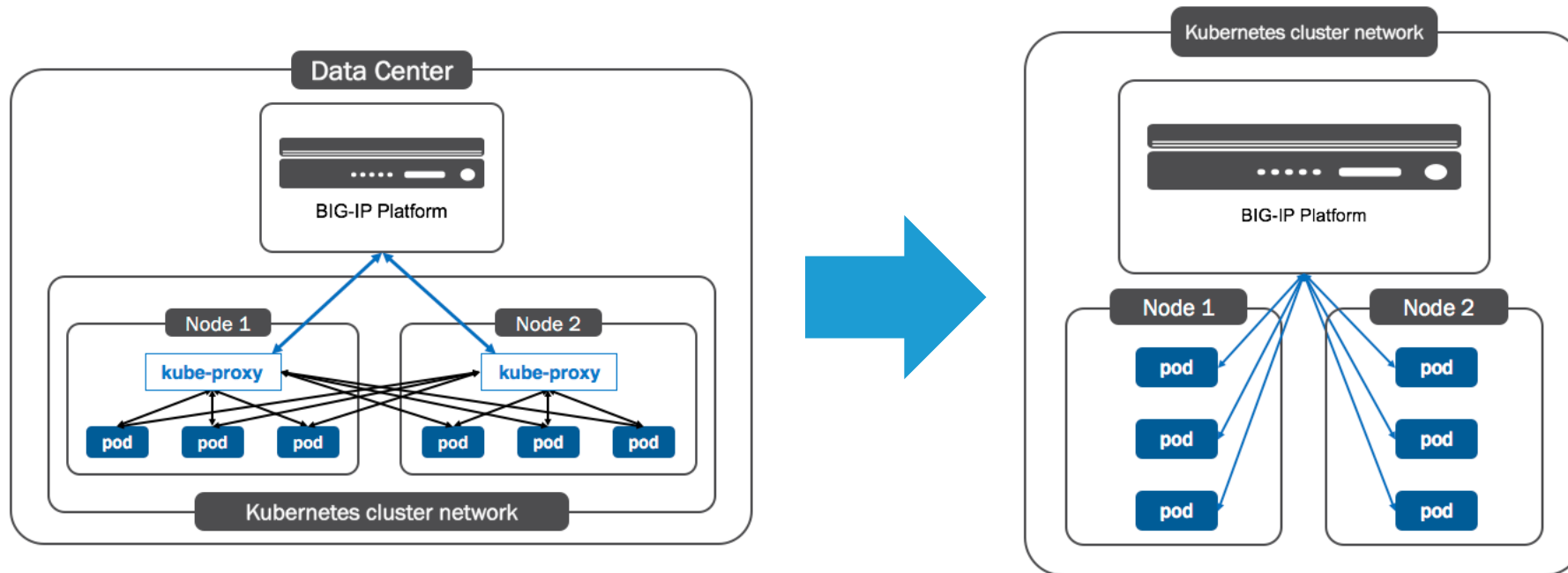
기술지원이 가능한 (Data Plan + Control Plan) Open Source Service Mesh 솔루션.
예) ASPEN MESH

유연한 North-South 동적 애플리케이션 서비스가 가능합니다



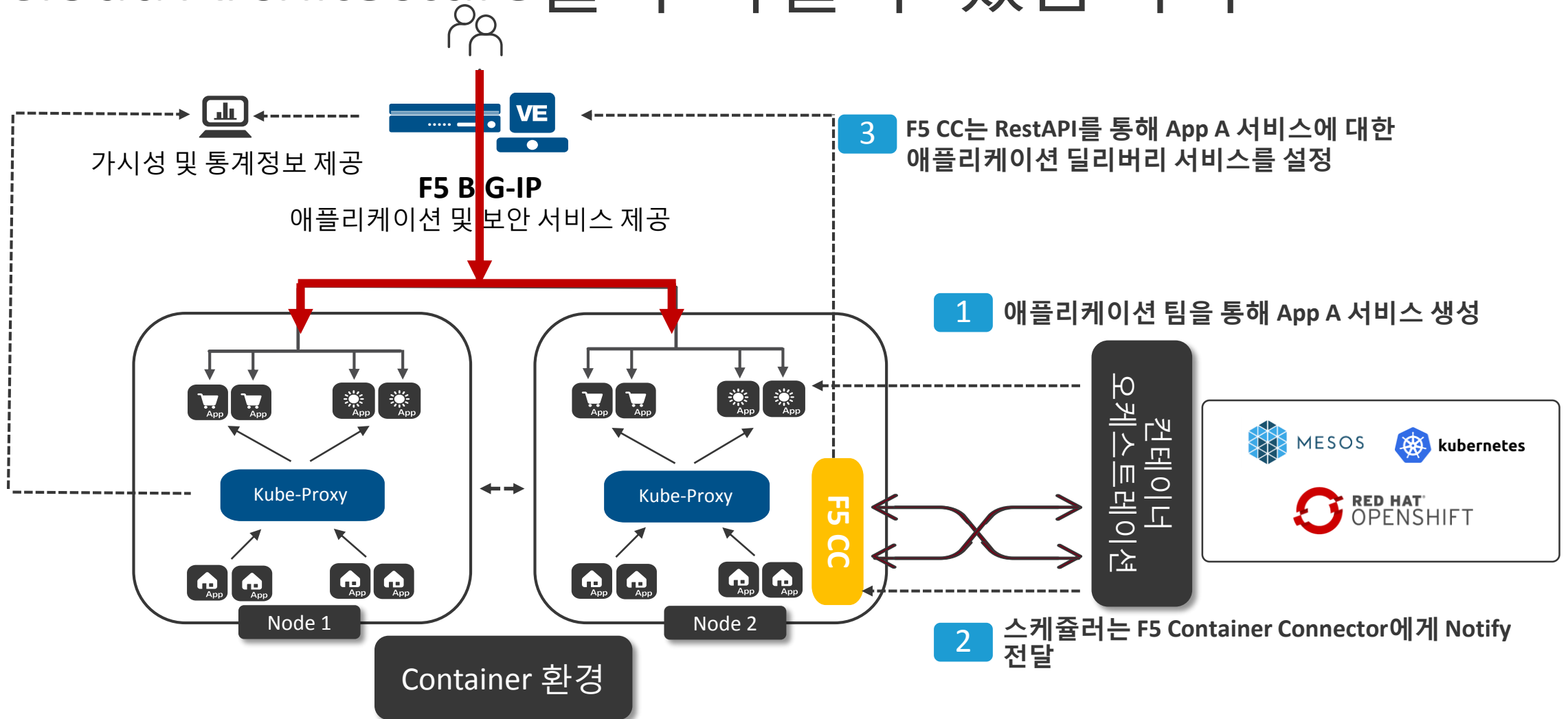
- 유연한 North-South 트래픽에 대한 동적 애플리케이션 서비스를 제공하는 Container Connector Architecture 입니다.

North-South 트래픽 처리를 위한 Kubernetes Nodeport mode 와 Cluster mode 의 차이점입니다.

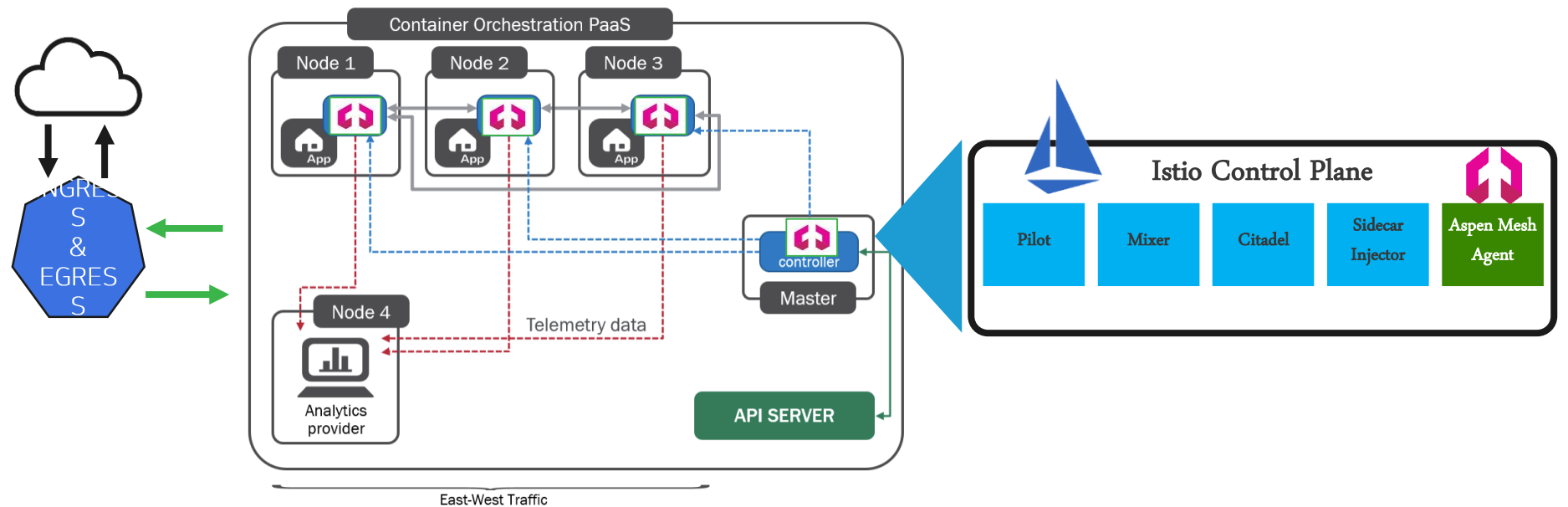


- Container Connector는 Container 환경에서의 North-South 유연한 트래픽 전달을 가능하게 합니다. (Kubernetes의 Nodeport Mode를 Cluster Mode로 전환 운영 가능)

North-South 트래픽 처리를 위한 유연한 자동화된 Cloud Architecture를 구축할 수 있습니다

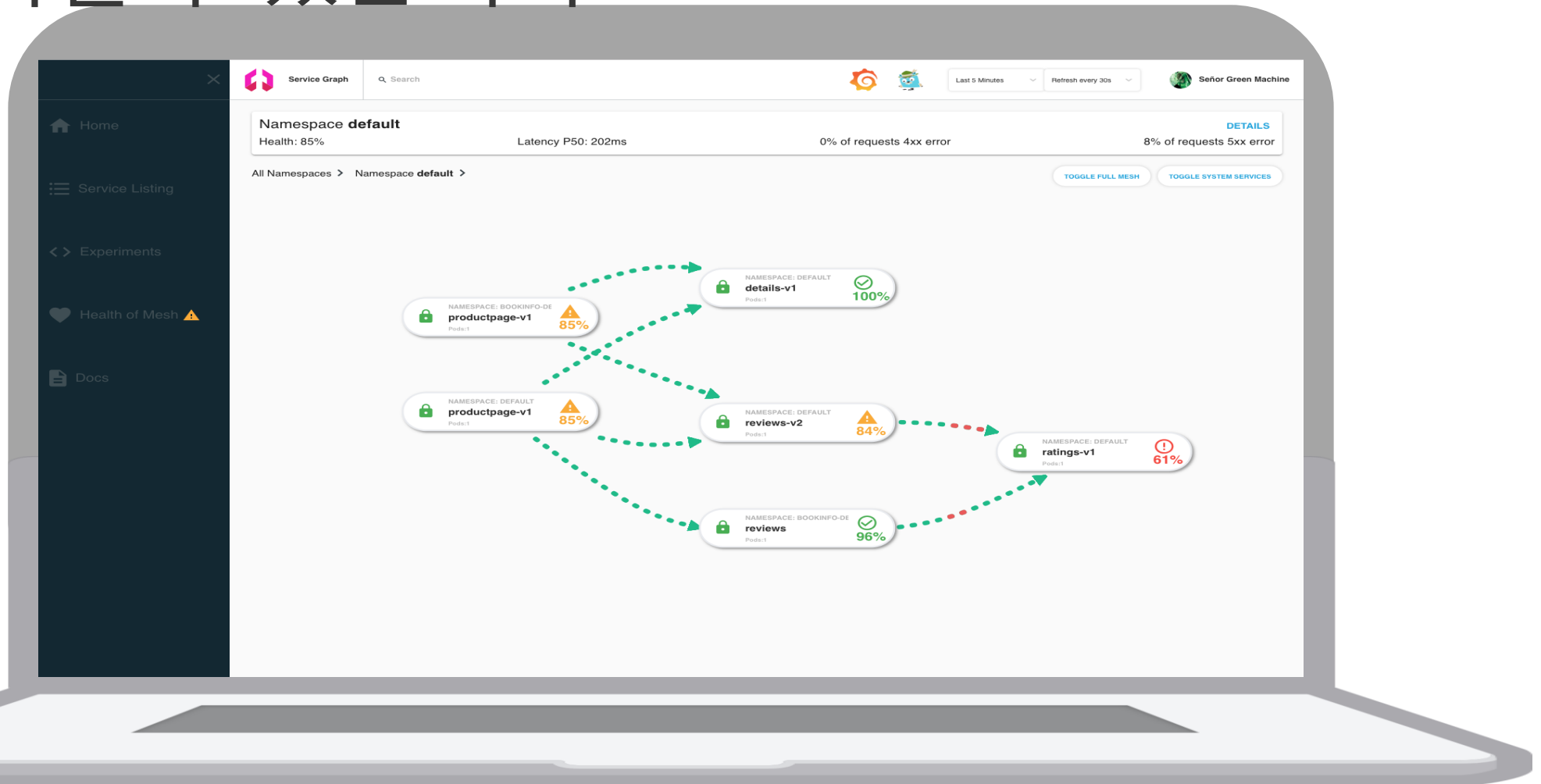


유연한 **East-West** 동적 애플리케이션 서비스가 가능합니다



- Aspen Mesh Agent는 Kubernetes Cluster Network에 대한 관찰성(Observability), 보안(Security) 및 전체적인 서비스 관리를 제공합니다.

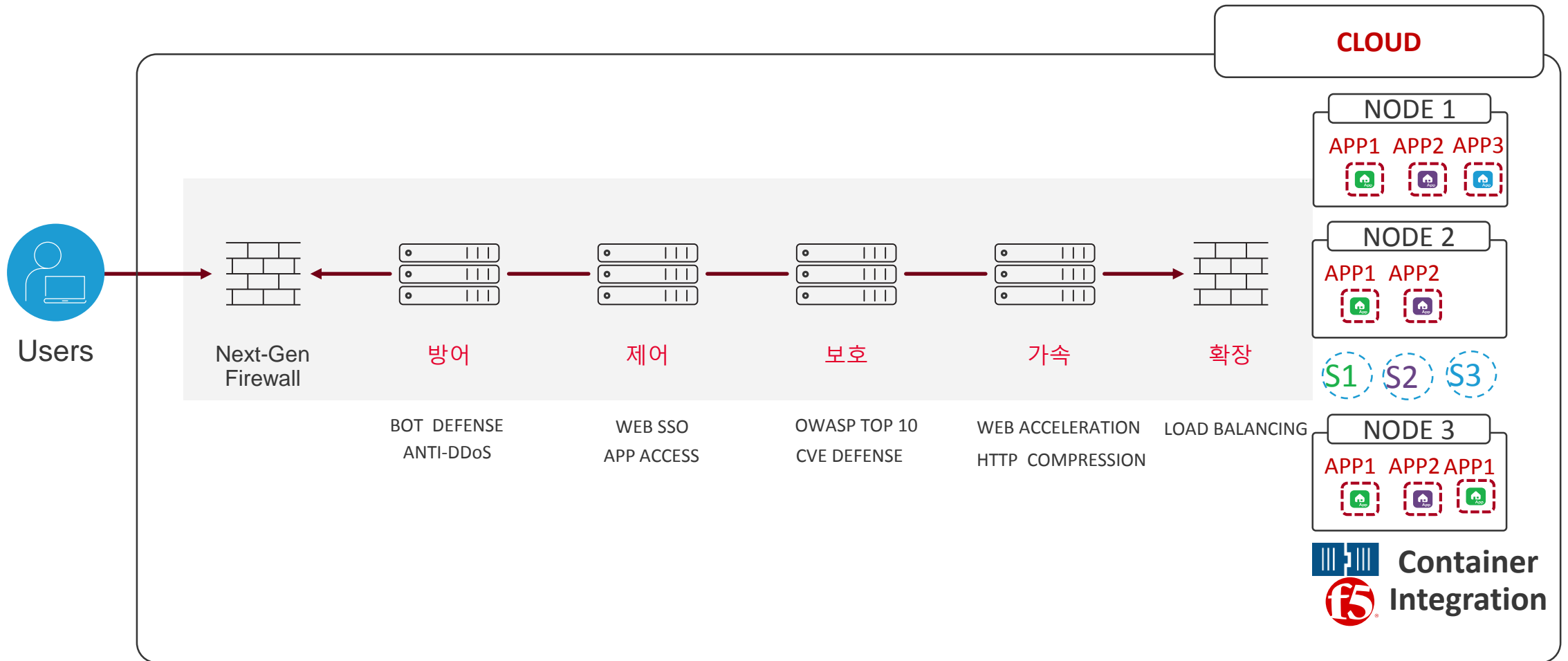
마이크로 서비스 관리 및 보안과 가시성을 보다 더 강화할 수 있습니다.



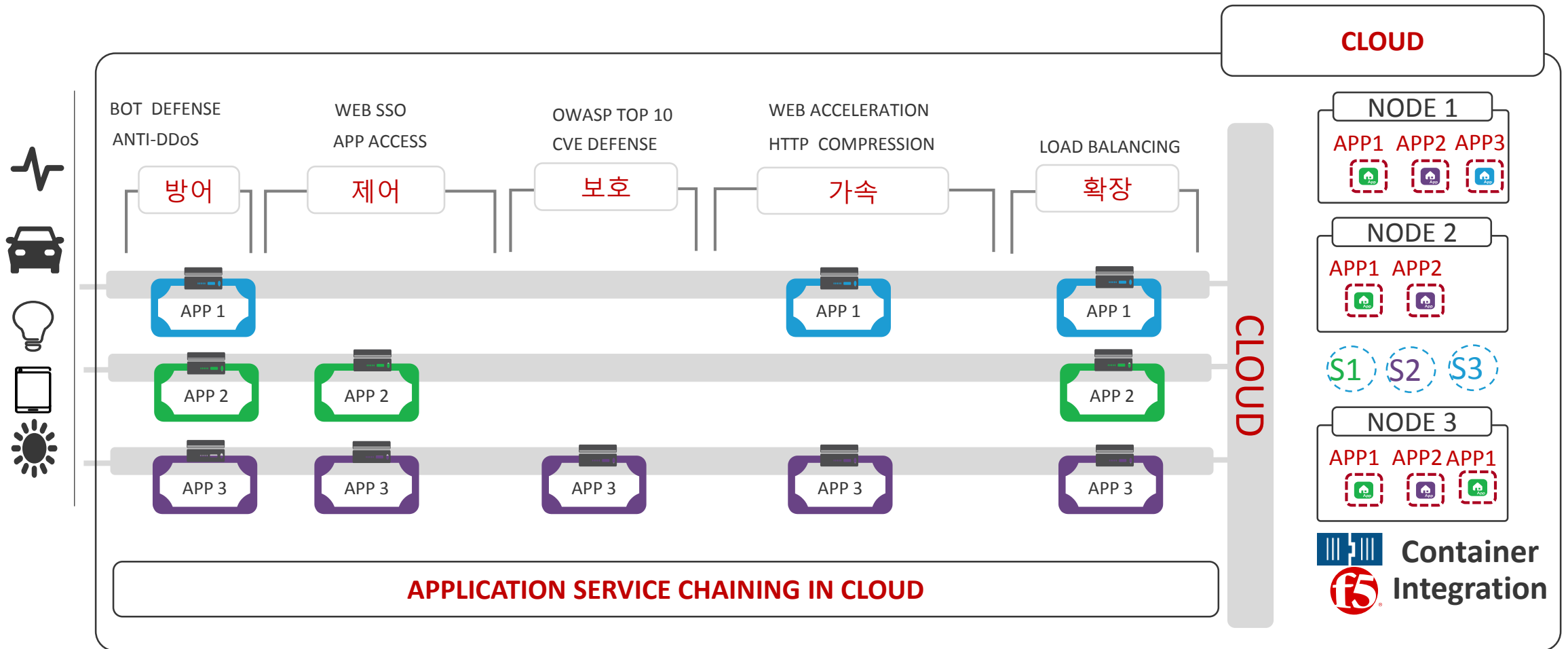


세 번째: Dynamic Service Chaining in Cloud

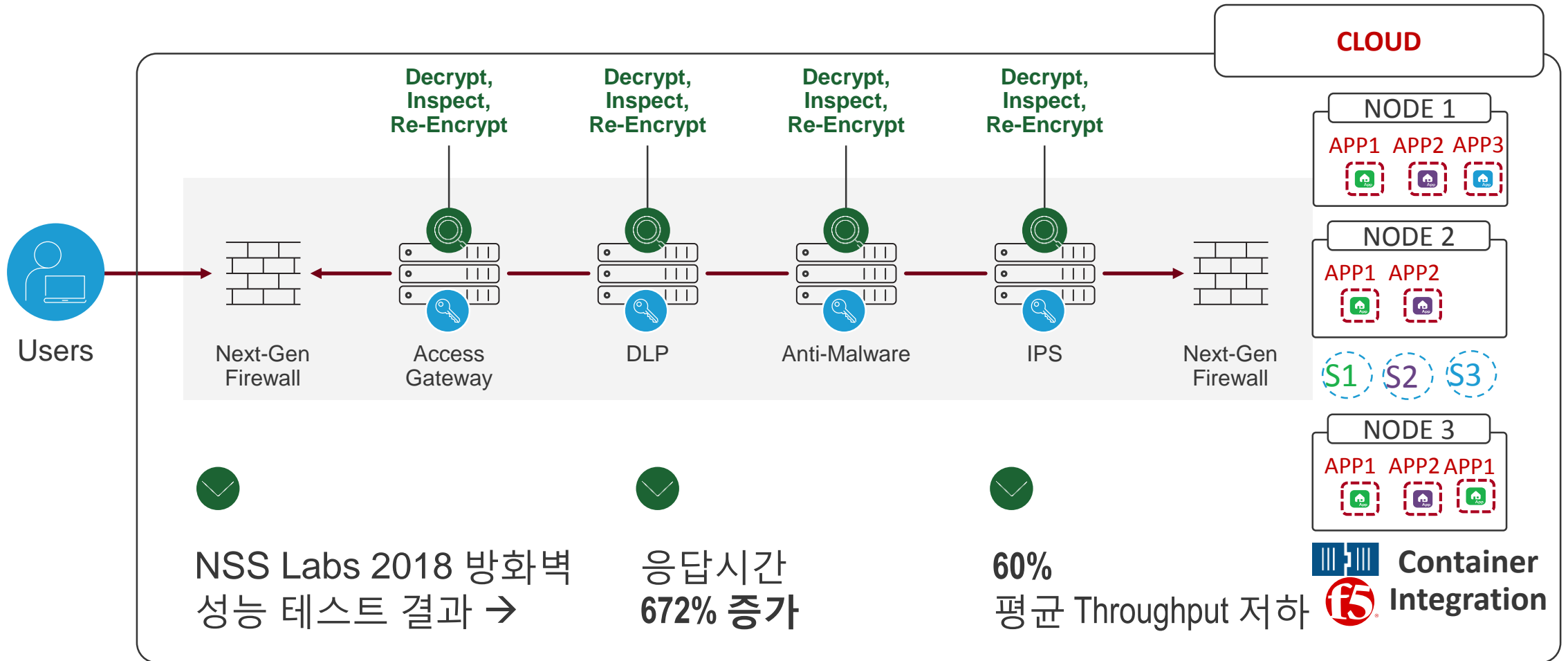
클라우드에서의 일반적인 애플리케이션에 대한 Daisy-Chaining Architecture 입니다.



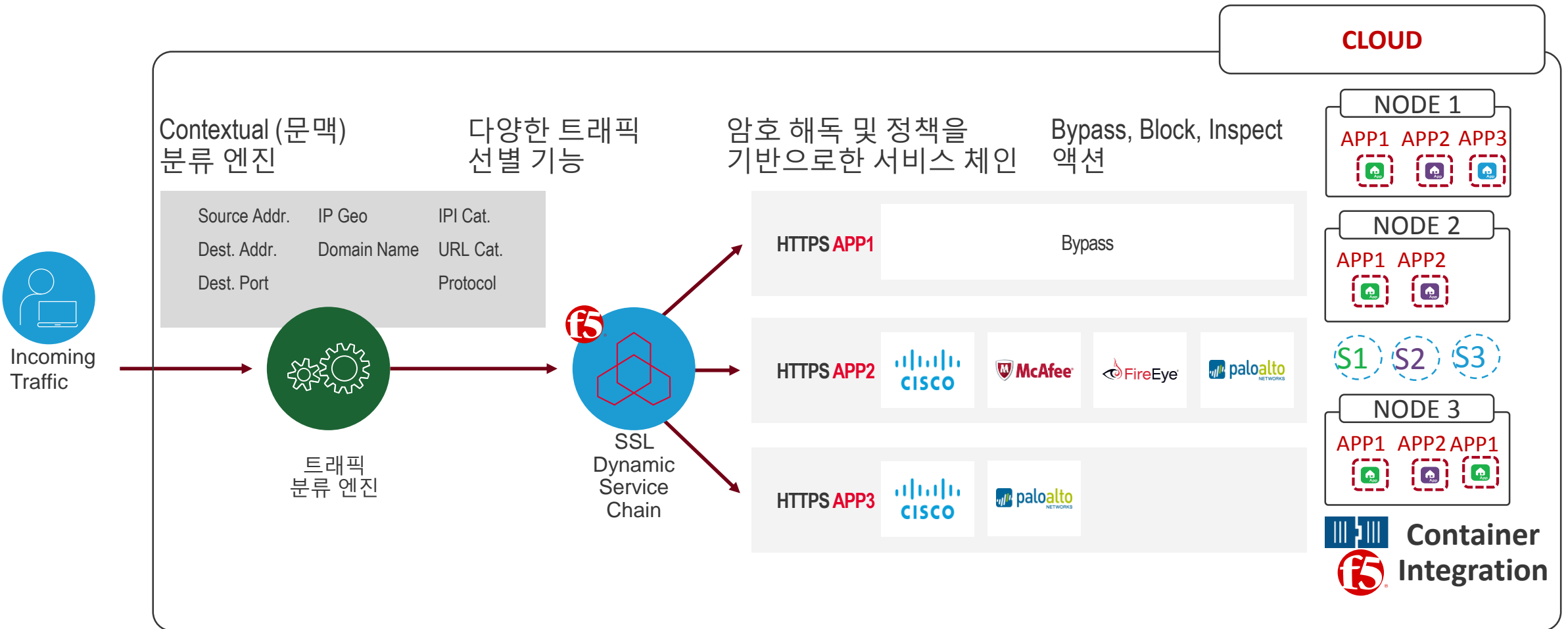
애플리케이션에 대한 Dynamic Service-Chaining 은 Computing Resource를 최적화 할 수 있습니다.



SSL에 대한 일반적인 Daisy-Chaining 방식입니다.



SSL에 대한 Dynamic -Service Chaining 은 Computing Resource를 보다 더 최적화 할 수 있습니다.

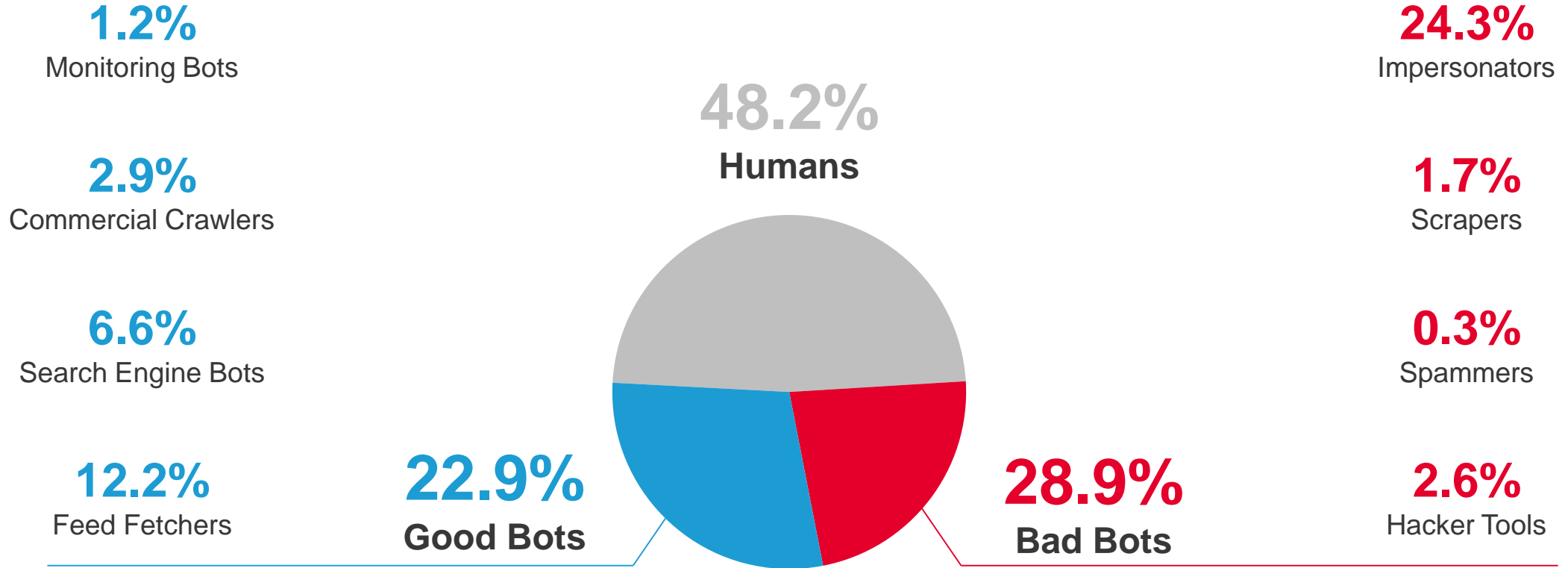




네 번째: Bad Bot Defense in Cloud

클라우드에서의 Bad Bots에 의한 트래픽

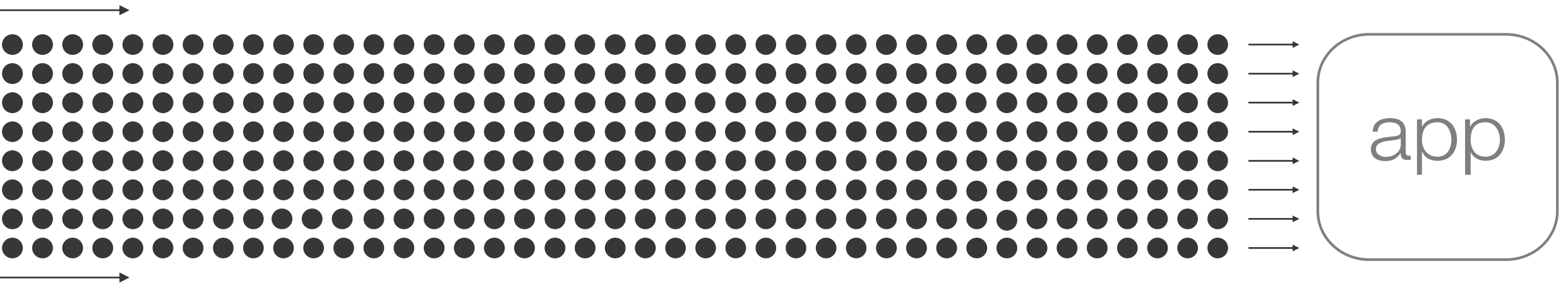
Bad Bot 트래픽에 의해 클라우드의 불필요한 Computing Resource 가 소모되고 있습니다.



Source: <https://www.globaldots.com/2018-bad-bot-report-the-year-bad-bots-went-mainstream/>

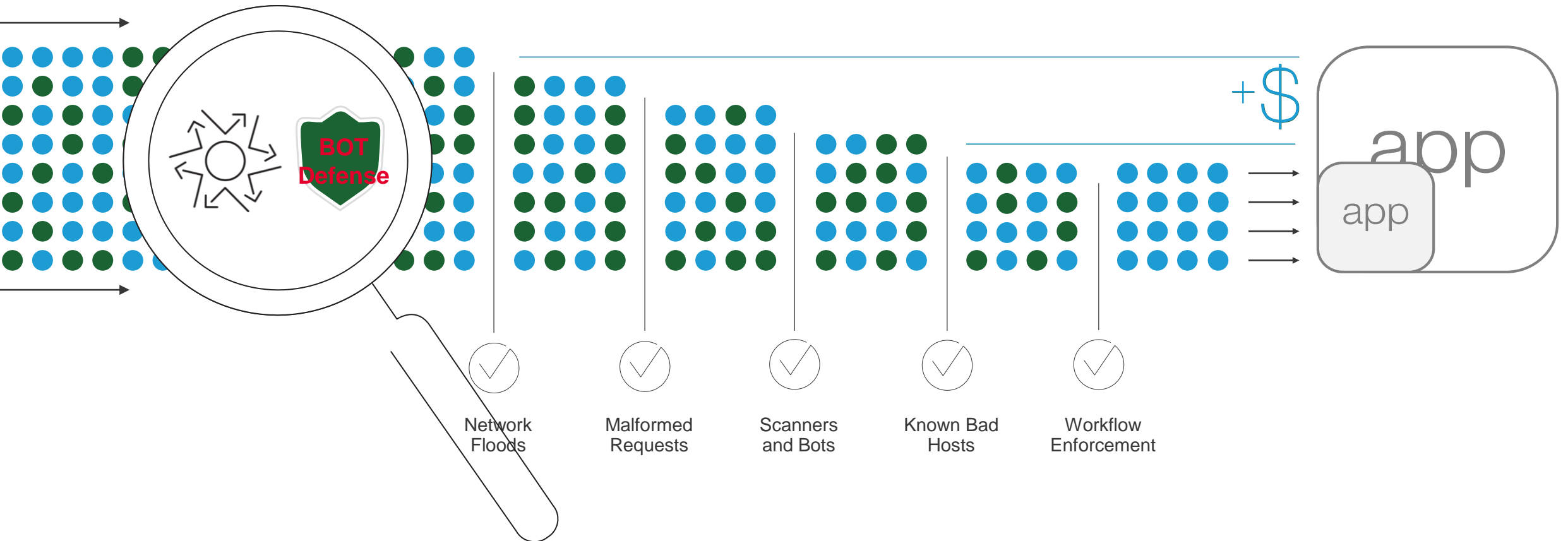
클라우드에서의 Bad Bot 트래픽 방어

지능화된 Bot Defense는 Bad Bot 트래픽에 의해 클라우드의 불필요한 Computing Resource 소모를 방어 할 수 있습니다.



클라우드에서의 Bad Bot 트래픽 방어

지능화된 Bot Defense는 Bad Bot 트래픽에 의해 클라우드의 불필요한 Computing Resource 소모를 방어 할 수 있습니다.



보다 지능적인 Bad Bots Defense 솔루션이 요구되고 있습니다.

Detect GET flood
attacks against
Heavy URIs

Identify non-human
surfing patterns



Fingerprint to
identify beyond
IP address

Operating System

Geolocation

Browser

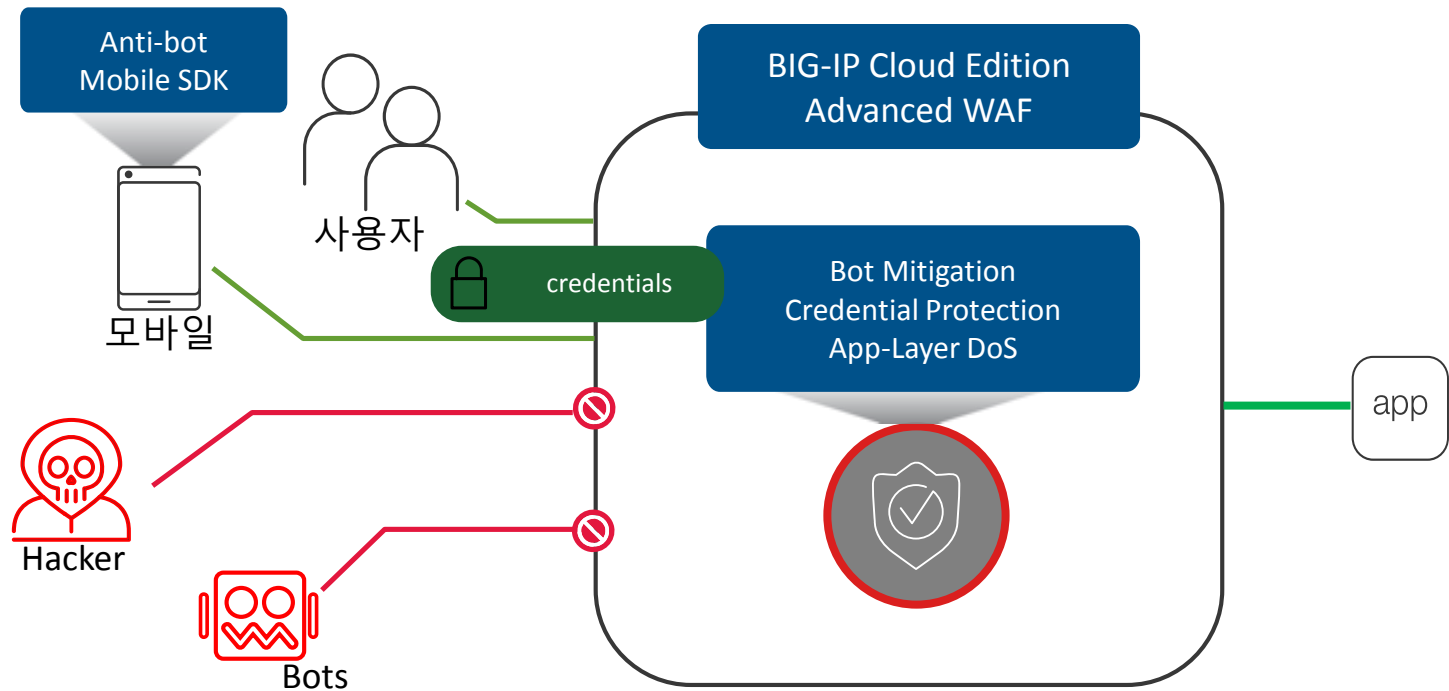
- Screen size and color depth
- Plugin details
- Time zone
- HTTP_ACCEPT Headers
- Language
- System Fonts
- Touch Support
- Extensions

보다 더 지능적인 Bad Bots Defense를 수행할 수 있습니다.



애플리케이션 계층의
차별화된 보안 기능

행위 분석 기반 차단
보안서비스 제공



Advanced WAF : Bad Bots 차단 및 새로운 공격 또는 진화하는 공격의 효율적인 보호가 가능합니다.



다섯 번째: Latency in Cloud

5G and IOT

5G는 LATENCY를 감소시킬 수 있는 기술을 필요로 하고 있습니다. (EDGE COMPUTING에서 고려해야 할 몇가지 기술들)

MQTT OVER TLS

암호화된 IOT 트래픽 적용

TCP 최적화

TCP 최적화를 통한
Latency 최소화

DNS CACHING

가장 손쉬운 Latency
최소화 방법중 한가지

Virtual Reality, Holography, Connected CAR, Smart Factory등이
실현되고 있습니다.

