



# 次世代アプリケーション基盤「BIG-IP Next」が実現する、運用管理の自動化による進化・改善・改革

2024年9月4日

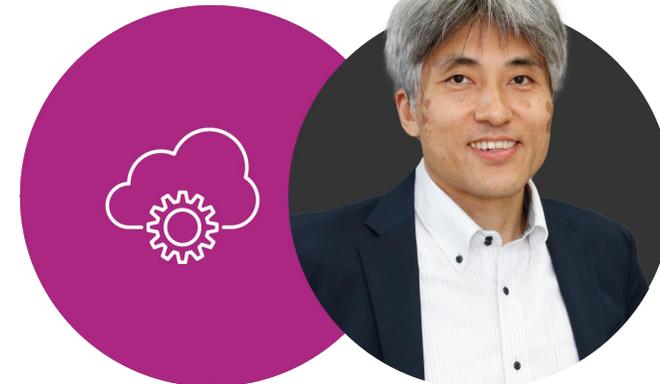
F5ネットワークスジャパン合同会社

シニアソリューションエンジニア

辻 哲也

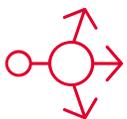
# 自己紹介

- [名前] 辻 哲也 (つじ てつや)
- [所属・肩書]  
F5ネットワークスジャパン合同会社 シニアソリューションエンジニア
- [担当業務]  
パートナー様向けの提案支援、新製品情報のご提供 etc.
- [経歴]  
国内通信大手企業から、外資系ネットワーク/セキュリティベンダーを経て、2019年にF5入社
- [得意分野] ネットワーク、セキュリティ、サーバ/ストレージ周辺技術
- [SNS (LinkedIn)] [www.linkedin.com/in/tetsuyatsuji](https://www.linkedin.com/in/tetsuyatsuji)



# 再掲: BIG-IPとは？

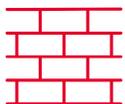
LOCAL LOAD  
BALANCING



GLOBAL LOAD  
BALANCING



FIREWALL



WEB APP  
FIREWALL



DDoS + BOT  
PROTECTION



ACCESS  
MANAGEMENT



SSL DECRYPTION  
and  
ORCHESTRATION



WEB FRAUD  
PROTECTION



APPLICATION  
PERFORMANCE  
MANAGEMENT



幅広いアプリケーション基盤のポートフォリオを提供



BIG-IPソフトウェア製品モジュールを稼働させる基盤



アプライアンス型HW  
シャーシ&ブレード型HW



仮想アプライアンス  
(VE)

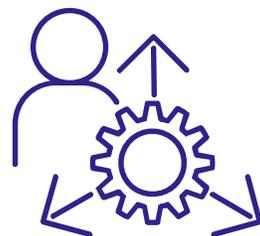
アプリを安心安全に公開

# BIG-IPが守る“アプリ”の進化



## 環境の多様化

- **87%**の顧客がマルチクラウドを利用<sup>(\*1)</sup>
- 平均で、**2.6個**のパブリッククラウドと**2.7個**のプライベートクラウドを利用<sup>(\*2)</sup>
- **58%**が、セキュリティとコンプライアンスの確保が最大の課題であると回答<sup>(\*1)</sup>



## Automation & DevOps

- **83%**の組織がDevOpsプロセスとツールを導入<sup>(\*3)</sup>
- **73%**の組織が効率を高めるためにネットワークの自動化を実施<sup>(\*1)</sup>
- **25%**の開発者とDevOpsがアプリの運用に注力<sup>(\*1)</sup>



## モダナイゼーション

- **75%**の組織がアプリのモダナイズを実施<sup>(\*1)</sup>
- 2024年までに、エンタープライズアプリの**75%**が、コンテナがデフォルトの技術に<sup>(\*4)</sup>
- インターネットトラフィックの**83%**がAPIであり、HTMLは**17%**のみ<sup>(\*5)</sup>

<sup>1</sup> F5 State of Application Services Report 2020

<sup>2</sup> Flexera State of the Cloud Report 2021

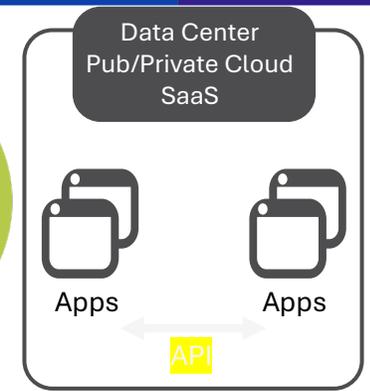
<sup>3</sup> State of DevOps Report 2021

<sup>4</sup> Gartner Forecast Analysis: container management

<sup>5</sup> Akamai State of the Internet Report 2019

# 昨今のアプリケーションのトレンド

- アプリケーションへアクセスすることを想定した場合、



最近だと、アプリケーションがパブリッククラウド (AWS/Azure/GCPなど)の環境で開発・提供、そのパブリッククラウドの上で利用されることが主流になってきています。

経済産業省が2018年に出した標準ガイドライン、クラウド・バイ・デフォルトは「政府情報システム（各府省がサービスや業務を実施する際に利用する情報システム）」を構築する際、クラウドの活用を第一に検討するという方針です。

## 企画的、方針的な概念

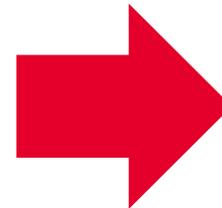
クラウド・バイ・デフォルト  
クラウドファースト  
クラウドシフト、クラウドリフト

## より技術的に使っていこう！という概念

クラウドネイティブ

## クラウドが使われる理由 (総務省[通信利用動向調査])

1. 資産、保守体制を社内に持たないから
2. 場所、機器を選ばずに利用できるから
3. 安定運用、可用性が高くなるから
4. 災害時のバックアップとして利用できるから
5. サービスの信頼性が高いから



## どんな技術なアプローチなの??

1. コンテナ
2. マイクロサービス
3. サービスメッシュ
4. イミュータブル・インフラストラクチャ
5. 宣言型API

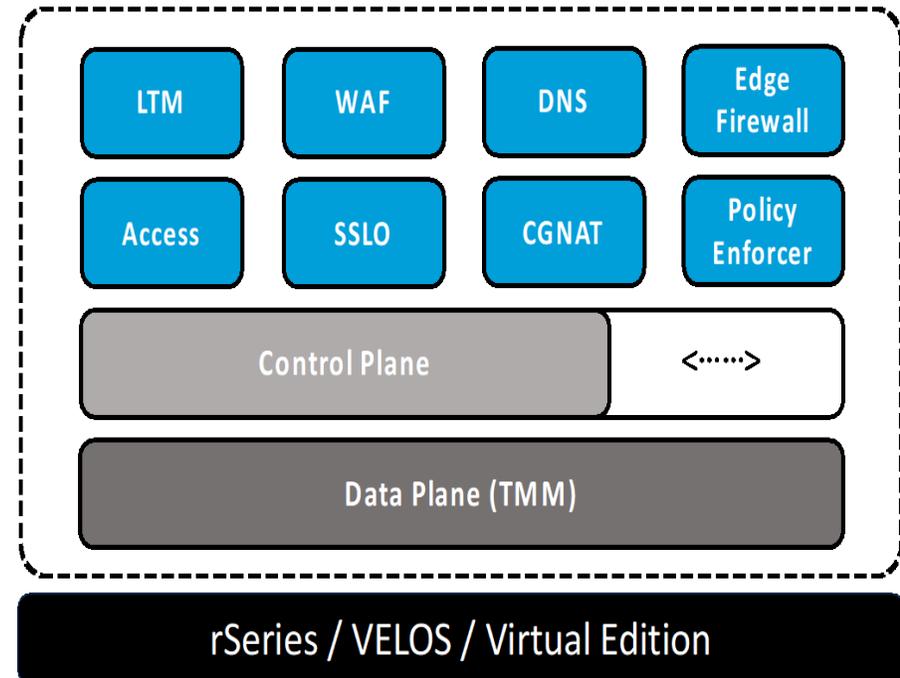


# なぜBIG-IP “Next”が必要なのか？

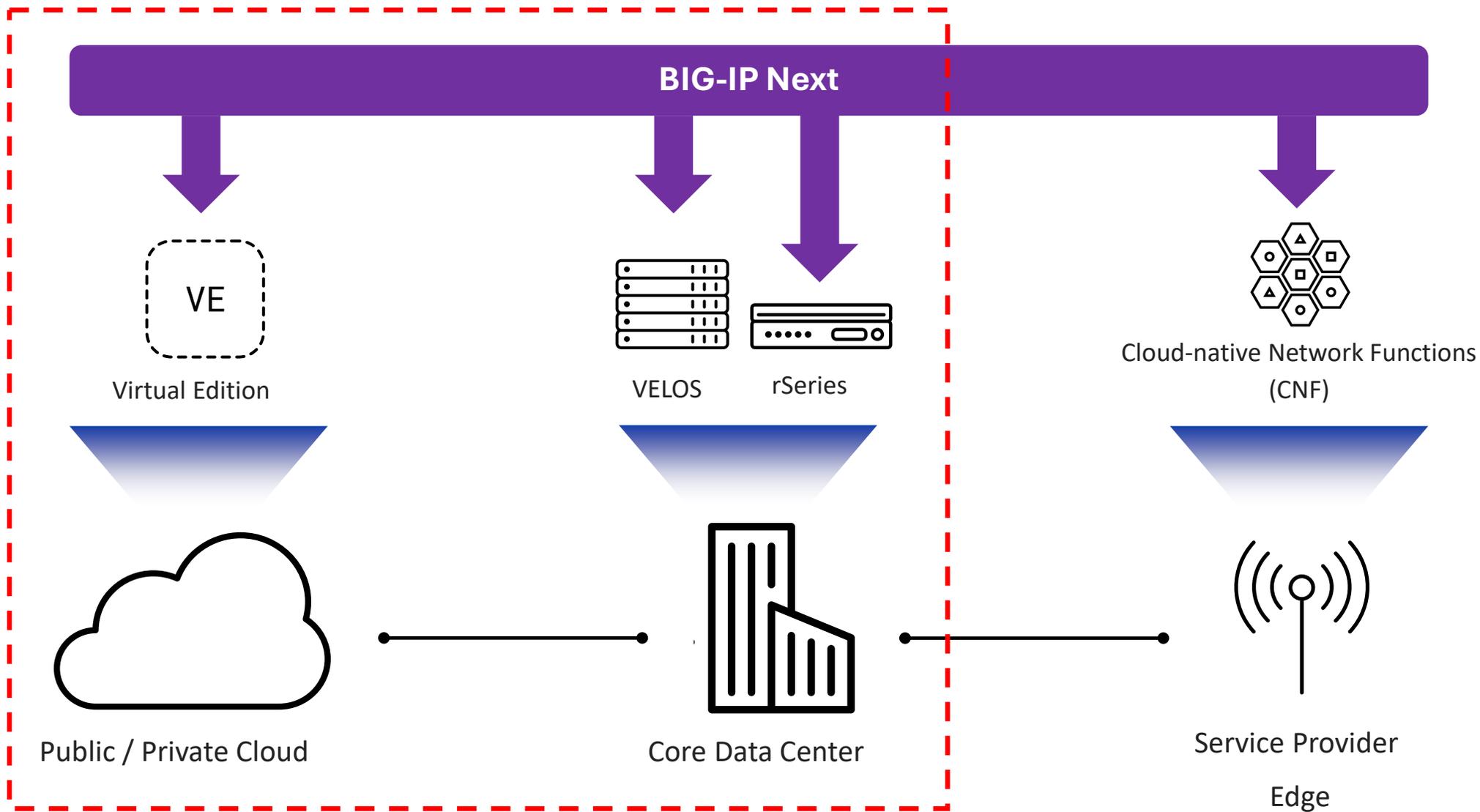
- 現行のBIG-IP (TMOS)はコントロールプレーンがモノリシックであり、パフォーマンスや拡張性に限界
  - 大量のコンフィグの読み込み、OSアップグレード、多量API操作時の冪等性担保に時間がかかる



- コンテナ・テクノロジーを利用して、コントロールプレーンとデータプレーンを分離
- 各機能 (=プロセス)単位でコンテナ化することにより、全体のパフォーマンスや管理性を向上

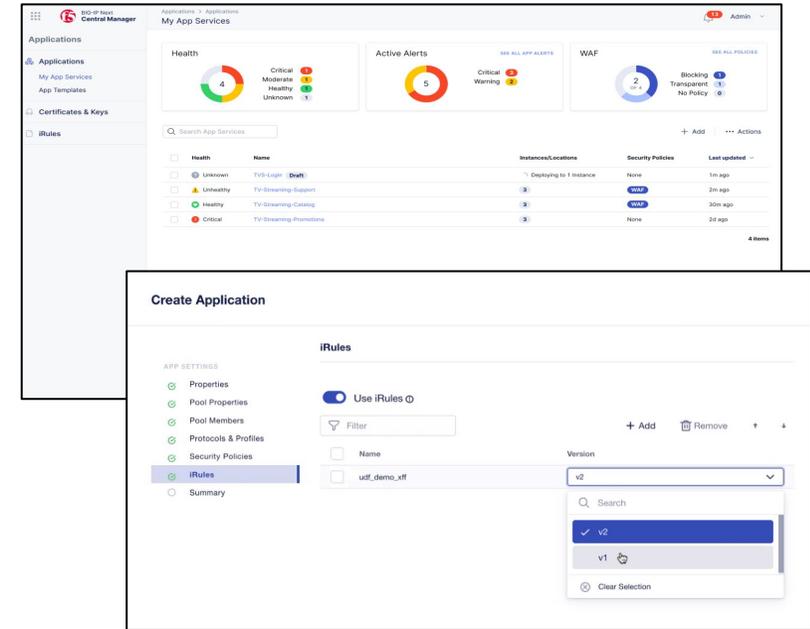
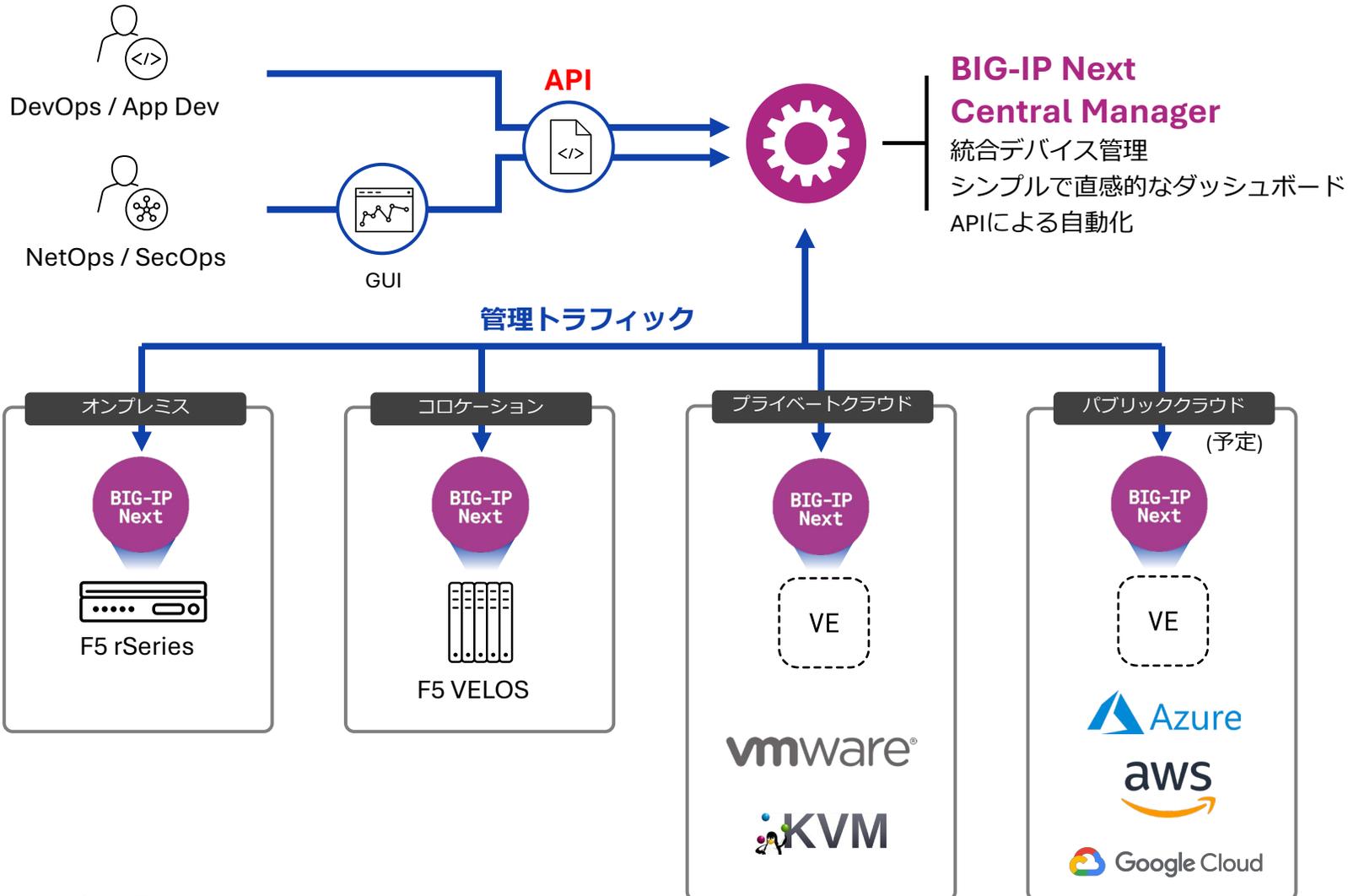


# BIG-IP Nextはあらゆるインフラ環境で利用可能



# BIG-IP Nextの利用形態

BIG-IP Next Central Managerによる統合管理



今まで

個々のBIG-IPそれぞれに対して設定を実行



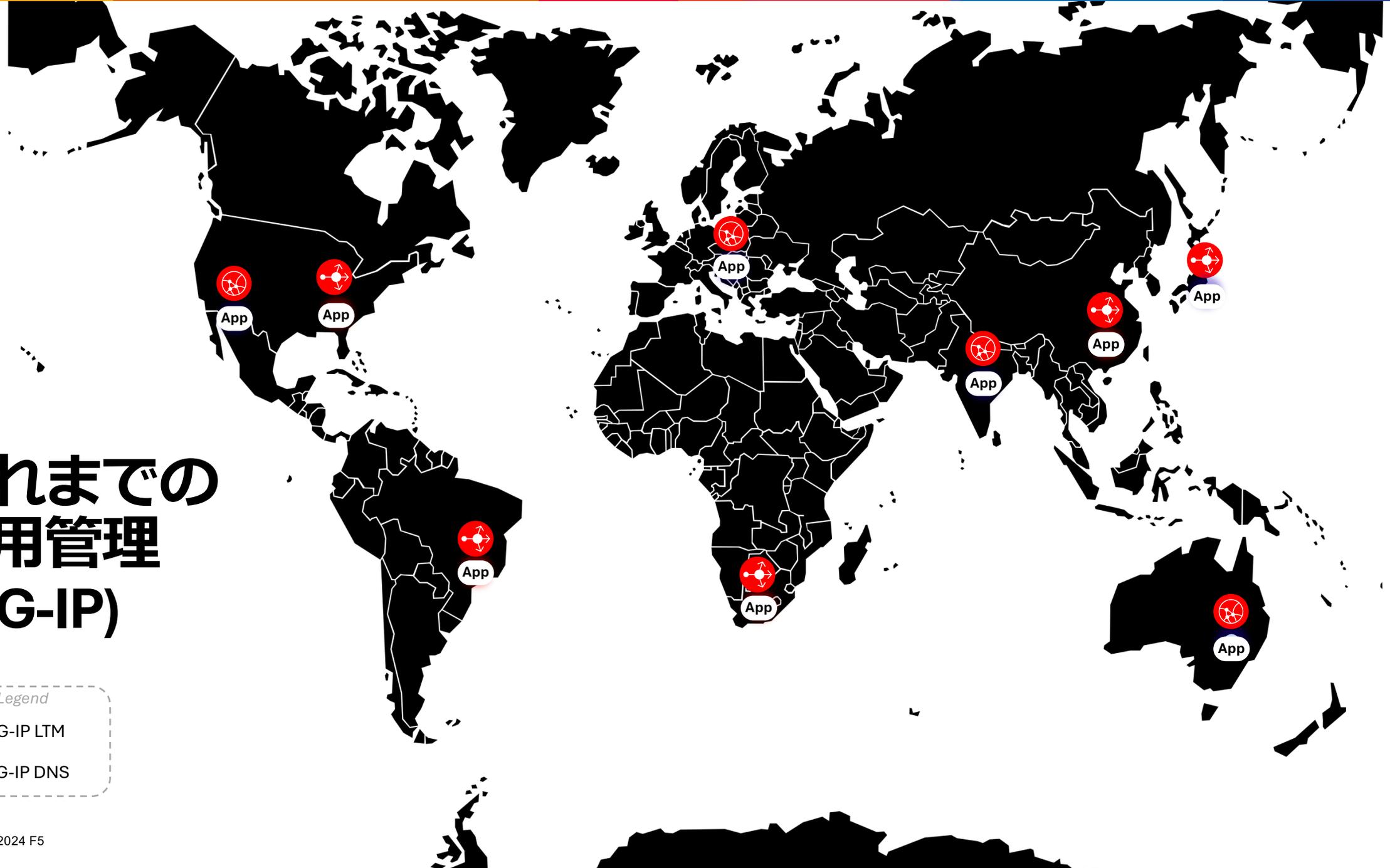
これから

設定を”テンプレート”化して、指定した複数のBIG-IP Nextインスタンス (テナント)に適用

# これまでの 運用管理 (BIG-IP)

Legend

-  BIG-IP LTM
-  BIG-IP DNS



# これからの 運用管理 (BIG-IP Next)

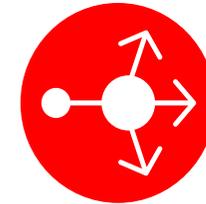
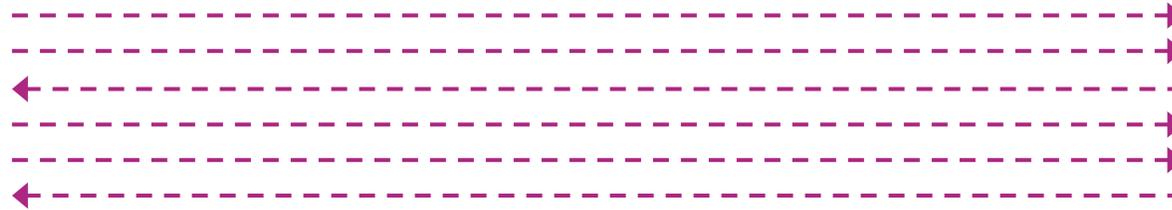
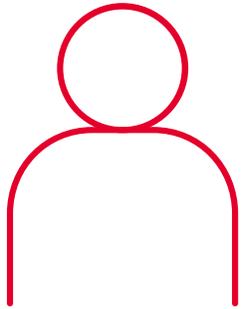
Legend

-  BIG-IP Next LTM
-  BIG-IP Next DNS
-  BIG-IP Next CM

# 命令型API/CLIによるオペレーション

- BIG-IPを設定するためには、数多くのREST API Callが必要

多数の命令型コマンド



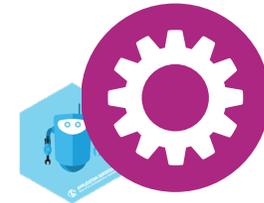
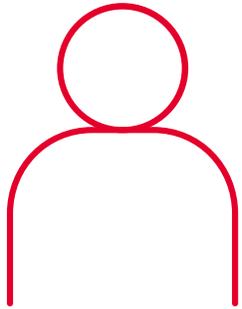
**BIG-IP LTM**  
(w/o AS3)

- BIG-IPに関する知識が必要 (Virtual Server, Pool, Profile etc.)
- 多くのREST API、もしくははCLIコマンドを実行
- 自動化やオーケストレーション・ツールとの連携が困難
- 時間がかかるアプローチ
- エラーが発生しやすいアプローチ

# BIG-IP Next Central Manager = 宣言型APIによるオペレーション

- BIG-IPを設定するために必要なのは、単一のREST API Callのみ

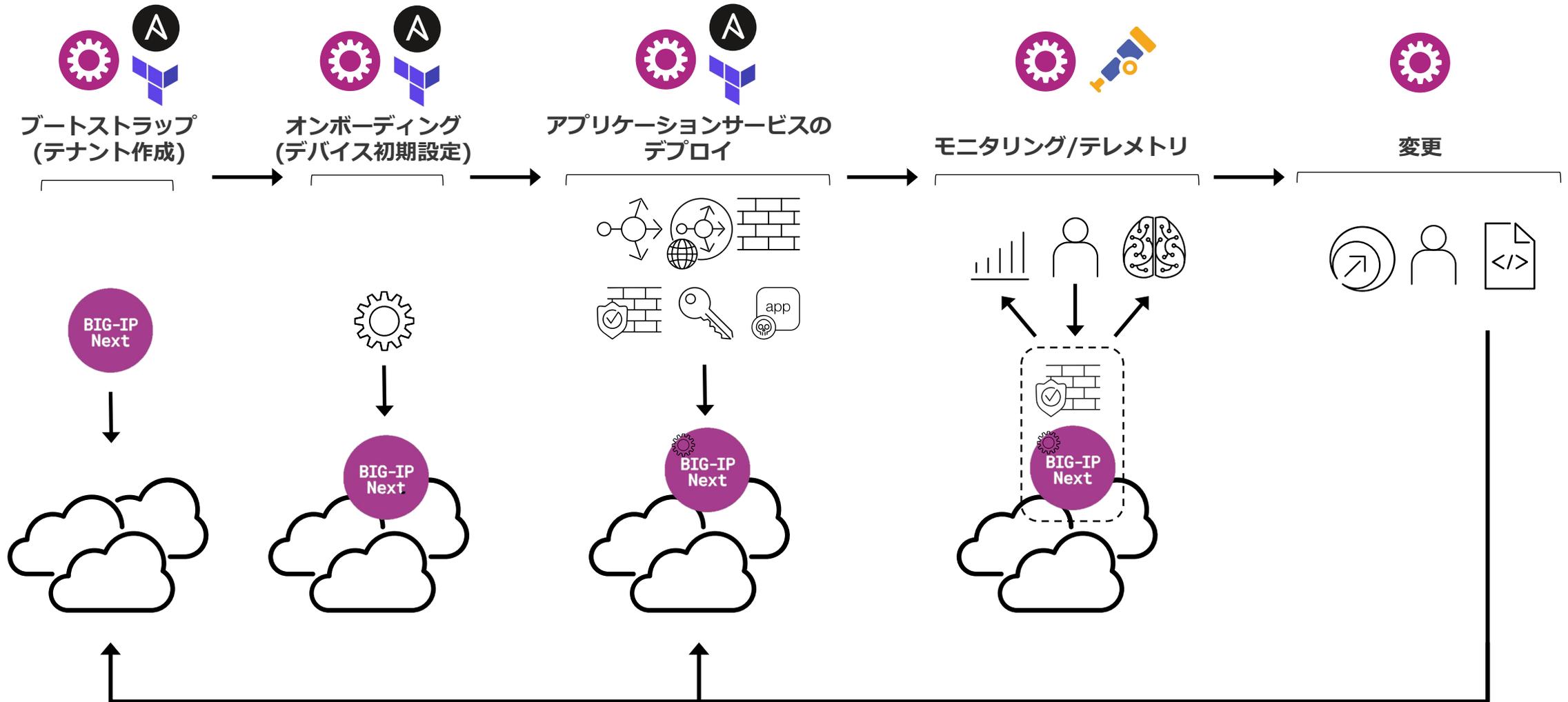
単一の宣言定義



BIG-IP Next Central Manager  
(powered by AS3)

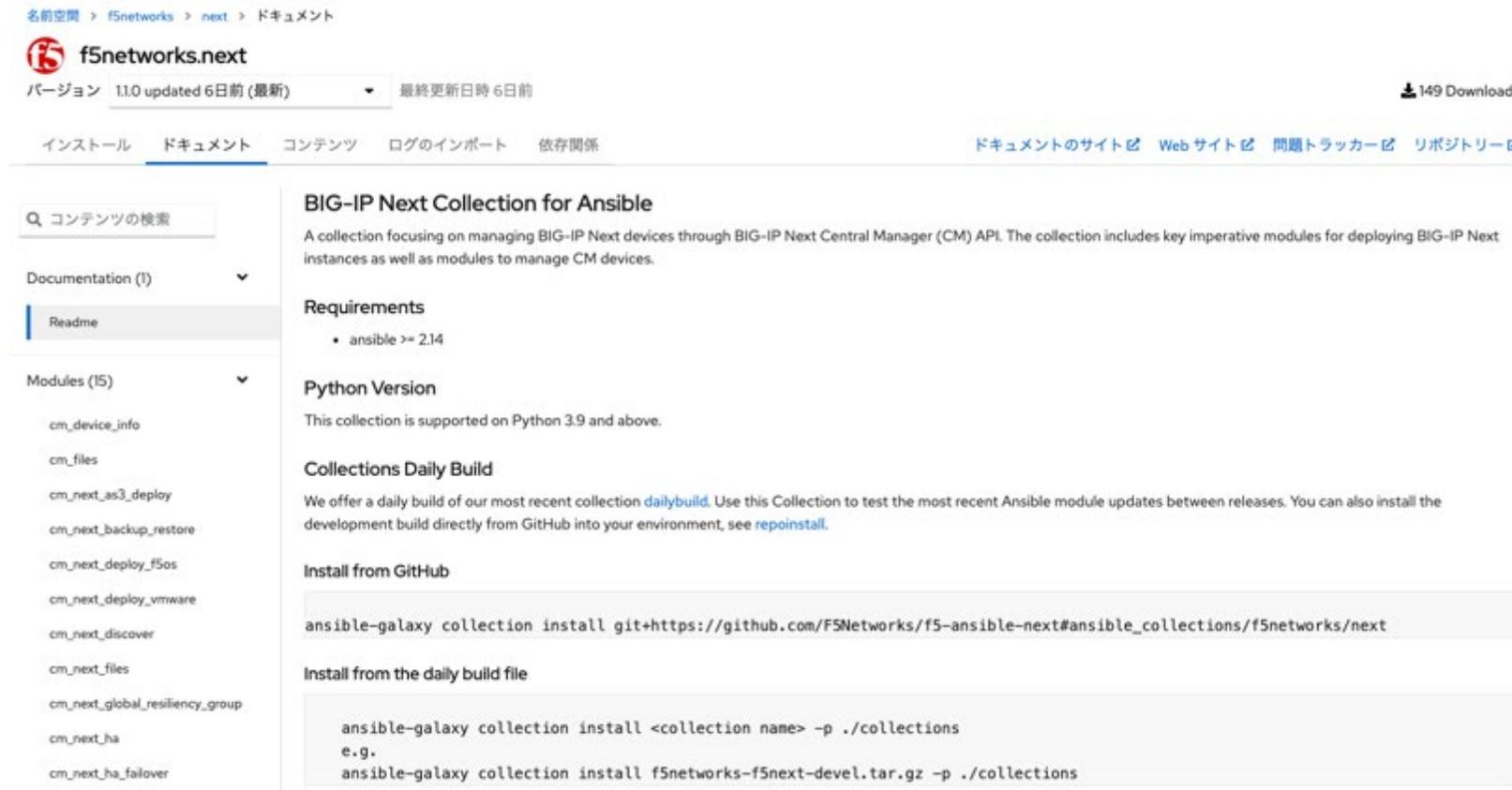
- AS3がBIG-IP設定の複雑さを抽象化
- 希望する構成の最終状態を定義するのみ
- 単一のREST API Callにより、自動化とオペレーションを簡素化
- アプリケーションサービスのデプロイを加速
- 宣言定義は再利用可能で、設定の一貫性を保証し、エラーを削減

# 自動化によるBIG-IP Nextのライフサイクル



# BIG-IP Next Collection for Ansible

- BIG-IP Nextに対応したAnsible Collectionをリリース
  - 詳細は以下を参照
    - <https://galaxy.ansible.com/ui/repo/published/f5networks/next/docs/>
    - <https://clouddocs.f5.com/products/orchestration/ansible/devel/next/next-index.html>



The screenshot shows the Ansible Galaxy page for the 'f5networks.next' collection. The page header includes the collection name, version (1.1.0), and update date (6 days ago). The main content area is titled 'BIG-IP Next Collection for Ansible' and provides a description, requirements (ansible >= 2.14), and Python version (3.9 and above). It also includes instructions for installing from GitHub and the daily build file.

名前空間 > f5networks > next > ドキュメント

**f5networks.next**  
バージョン 1.1.0 updated 6日前 (最新) 最終更新日時 6日前 149 Downloads

インストール ドキュメント コンテンツ ログのインポート 依存関係

ドキュメントのサイト Web サイト 問題トラッカー リポジトリ

コンテンツの検索

Documentation (1) **Readme**

Modules (15)

- cm\_device\_info
- cm\_files
- cm\_next\_as3\_deploy
- cm\_next\_backup\_restore
- cm\_next\_deploy\_f5os
- cm\_next\_deploy\_vmware
- cm\_next\_discover
- cm\_next\_files
- cm\_next\_global\_resiliency\_group
- cm\_next\_ha
- cm\_next\_ha\_failover

### BIG-IP Next Collection for Ansible

A collection focusing on managing BIG-IP Next devices through BIG-IP Next Central Manager (CM) API. The collection includes key imperative modules for deploying BIG-IP Next instances as well as modules to manage CM devices.

#### Requirements

- ansible >= 2.14

#### Python Version

This collection is supported on Python 3.9 and above.

#### Collections Daily Build

We offer a daily build of our most recent collection [dailybuild](#). Use this Collection to test the most recent Ansible module updates between releases. You can also install the development build directly from GitHub into your environment, see [reinstall](#).

#### Install from GitHub

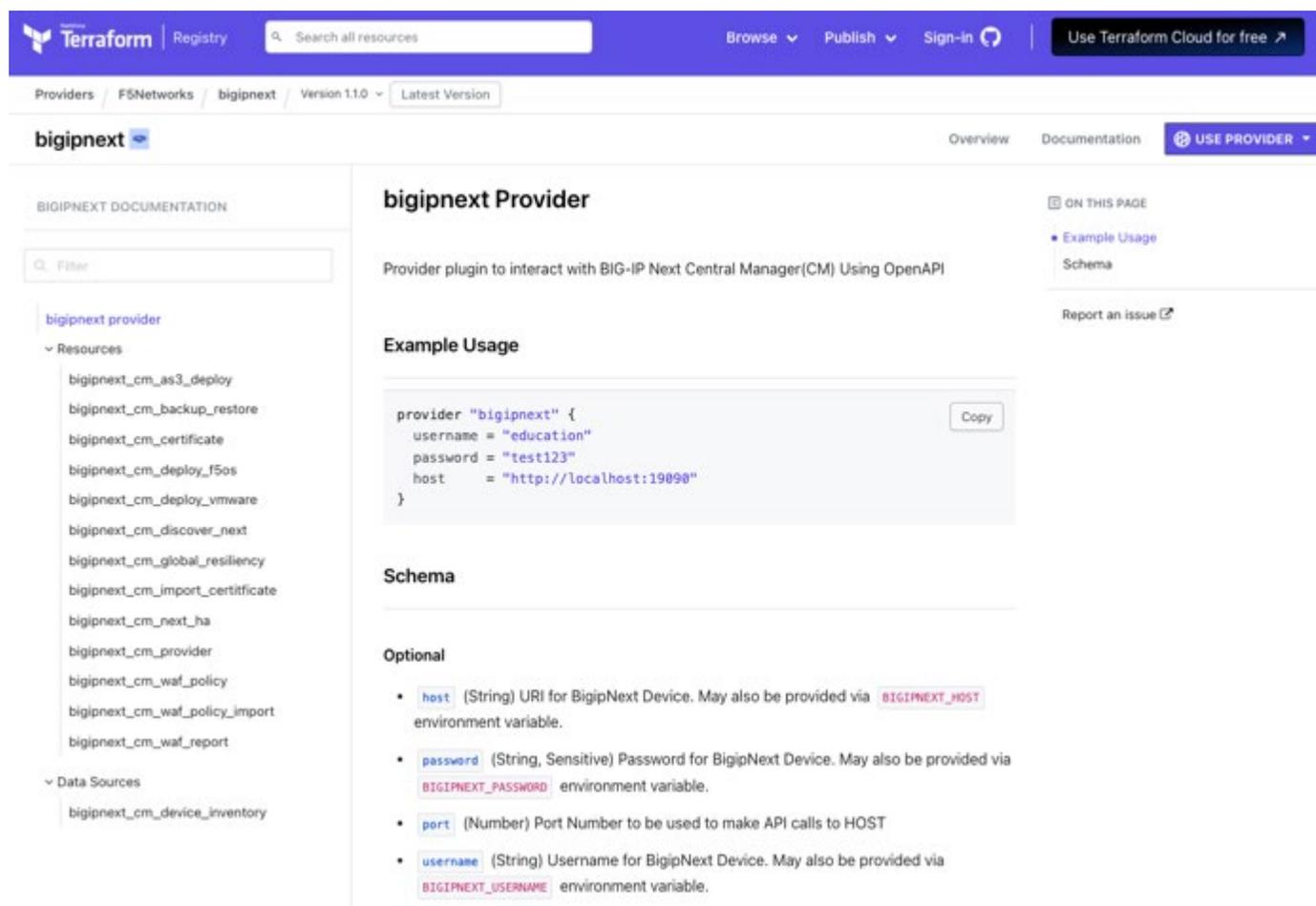
```
ansible-galaxy collection install git+https://github.com/F5Networks/f5-ansible-next#ansible_collections/f5networks/next
```

#### Install from the daily build file

```
ansible-galaxy collection install <collection name> -p ./collections  
e.g.  
ansible-galaxy collection install f5networks-f5next-devel.tar.gz -p ./collections
```

# BIG-IP Next Terraform Provider

- BIG-IP Nextに対応したTerraform Providerをリリース
- 詳細は以下を参照
- <https://registry.terraform.io/providers/F5Networks/bigipnext/latest/docs>



The screenshot shows the Terraform Registry page for the `bigipnext` provider. The page is titled "bigipnext Provider" and includes a search bar, navigation links, and a list of resources. The main content area displays the provider's description, example usage, and schema.

**bigipnext Provider**

Provider plugin to interact with BIG-IP Next Central Manager(CM) Using OpenAPI

**Example Usage**

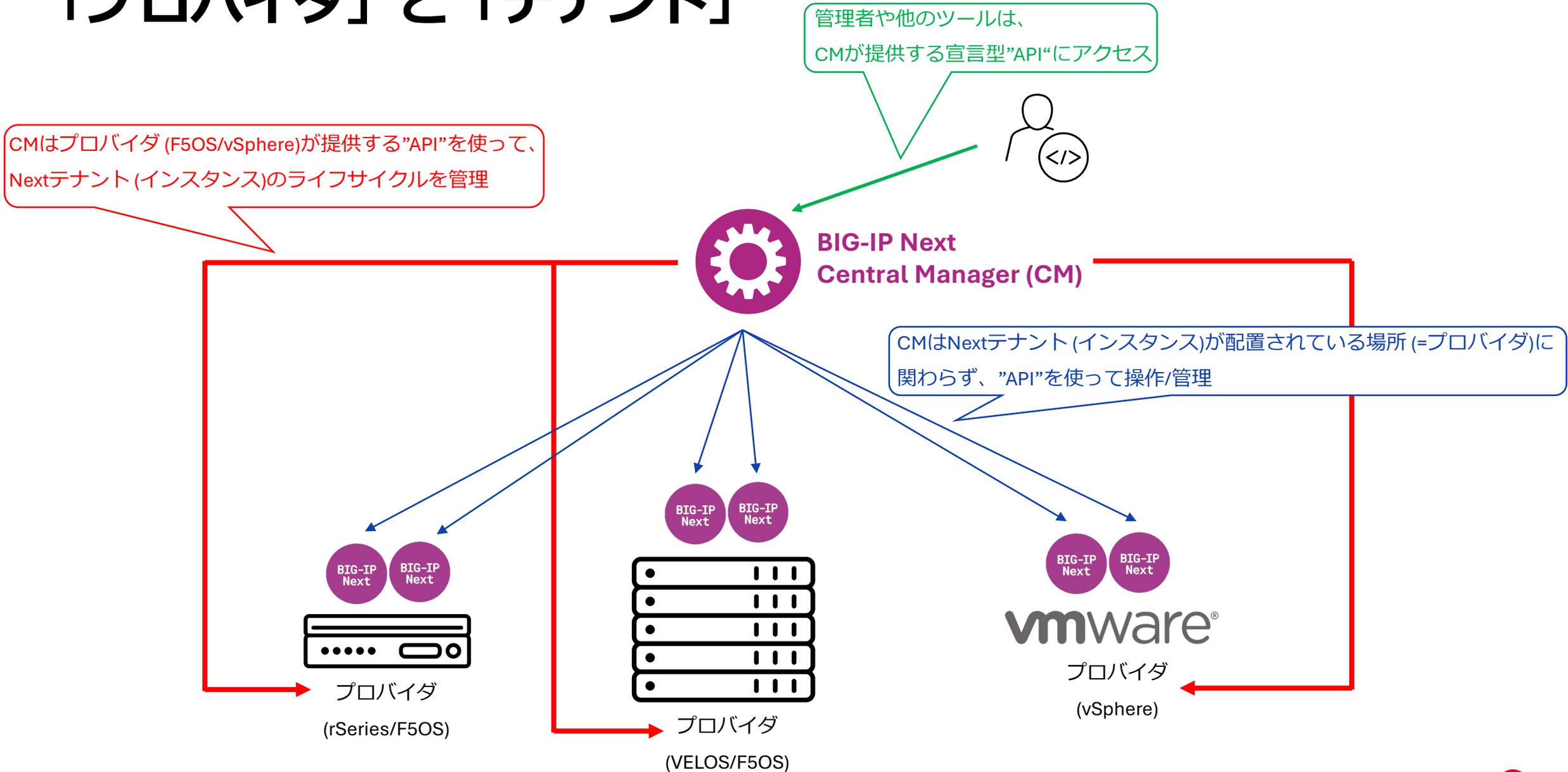
```
provider "bigipnext" {
  username = "education"
  password = "test123"
  host     = "http://localhost:19090"
}
```

**Schema**

**Optional**

- `host` (String) URI for BigipNext Device. May also be provided via `BIGIPNEXT_HOST` environment variable.
- `password` (String, Sensitive) Password for BigipNext Device. May also be provided via `BIGIPNEXT_PASSWORD` environment variable.
- `port` (Number) Port Number to be used to make API calls to HOST
- `username` (String) Username for BigipNext Device. May also be provided via `BIGIPNEXT_USERNAME` environment variable.

# 「プロバイダ」と「テナント」



# BIG-IP Nextテナント(インスタンス)の作成と初期設定

例: BIG-IP Next Ansible Collections

- **cm\_next\_deploy\_f5osモジュールにより、F5OSプロバイダ (rSeries/VELOS)上にBIG-IP Nextテナントをデプロイ**

[https://clouddocs.f5.com/products/orchestration/ansible/devel/next/modules\\_4\\_0/cm\\_next\\_deploy\\_f5os\\_module.html#examples](https://clouddocs.f5.com/products/orchestration/ansible/devel/next/modules_4_0/cm_next_deploy_f5os_module.html#examples)

## Examples

```
- name: Deploy BIG-IP Next instance on rSeries
cm_next_deploy_f5os:
  state: "deploy"
  provider:
    provider_name: "vCenterDemo"
    type: "rseries"
    image_name: "BIG-IP-Next-20.0.1-2.139.10-0.0.136-VM"
  instance:
    instance_hostname: "demoVM01.lab.local"
    vlans:
      - 21
      - 22
      - 23
      - 24
    mgmt_address: "172.16.1.1/24"
    mgmt_gateway: "172.16.1.254"
    mgmt_user: "admin"
    mgmt_password: "s3KrI!T"
    ha_dp_vlan_name: "dp-demo-ha"
    ha_cp_vlan_name: "cp-demo-ha"
    ha_dp_vlan_tag: 23
    ha_cp_vlan_tag: 24
    external_network_name: "DemoNet123"
    external_vlan_name: "ExtDemoVlan"
    external_vlan_tag: 21
    external_ip_address: "192.168.1.1/24"
    internal_vlan_name: "IntDemoVlan"
    internal_vlan_tag: 22
    internal_ip_address: "192.168.2.1/24"
  register: task

- name: Check the deployment status
cm_next_deploy_f5os:
  state: "present"
  task_id: "{{ task.task_id }}"
  timeout: 1000
```

# アプリケーションサービスのデプロイ

例: REST API (AS3)

- 宣言的なAPI (AS3)によるアプリケーション作成

```
POST https://{{bigip_next_cm_mgmt_ip}}/mgmt/shared/appsvcs/declare
Params Authorization Headers (9) Body Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON
1 [{"class": "ADC",
2  "schemaVersion": "3.0.0",
3  "id": "openapi-spec",
4  "ecommerce": {
5    "class": "Tenant",
6    "http_demo_app_as3": {
7      "class": "Application",
8      "http_demo_app-virtual": {
9        "class": "Service_HTTP",
10       "virtualPort": "{{vip_port1}}",
11       "virtualAddresses": [
12         "{{vip1}}"
13       ],
14     },
15     "pool": "http_demo_app-pool"
16   },
17   "http_demo_app-pool": {
18     "class": "Pool",
19     "members": [
20       {
21         "serverAddresses": [
22           "{{app_server1_ip}}",
23           "{{app_server2_ip}}",
24         ],
25         "servicePort": "{{app_server_port}}",
26       }
27     ],
28     "monitors": [
29       "http"
30     ]
31   }
32 }
33 ]
34 ]
```

Central Managerに対してAPI  
コールを実行

アプリケーションの設定を単一の  
JSON内で宣言的に定義

- Virtual Server
- Pool
- Pool Member
- Monitor

Etc.

# アプリケーションサービスのデプロイ

例: REST API (テンプレート:FAST)

```
POST https://{{bigip_next_cm_mgmt_ip}}/mgmt/shared/fast/appsvcs

Params Authorization Headers (9) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2   "allowOverwrite": false,
3   "name": "postman_app",
4   "set_name": "Examples",
5   "template_name": "http",
6   "parameters": {
7     "application_description": "",
8     "application_name": "postman_app",
9     "pools": [
10      {
11        "loadBalancingMode": "round-robin",
12        "monitorType": [
13          "http"
14        ],
15        "poolName": "postman_pool",
16        "servicePort": {{app_server_port}}
17      }
18    ],
19    "virtuals": [
20      {
21        "pool": "postman_pool",
22        "virtualName": "postman_vs",
23        "virtualPort": {{vip_port1}}
24      }
25    ]
26  }
27 }
```

選択するテンプレート

アプリケーション作成時に、デフォルトで指定

# アプリケーションサービスのデプロイ

例: Terraform “bigipnext” Provider

- **bigipnext\_cm\_as3\_deploy**リソースにより、AS3定義 (JSONフォーマット)のアプリケーションををデプロイ

[https://registry.terraform.io/providers/F5Networks/bigipnext/latest/docs/resources/cm\\_as3\\_deploy](https://registry.terraform.io/providers/F5Networks/bigipnext/latest/docs/resources/cm_as3_deploy)

## Example Usage

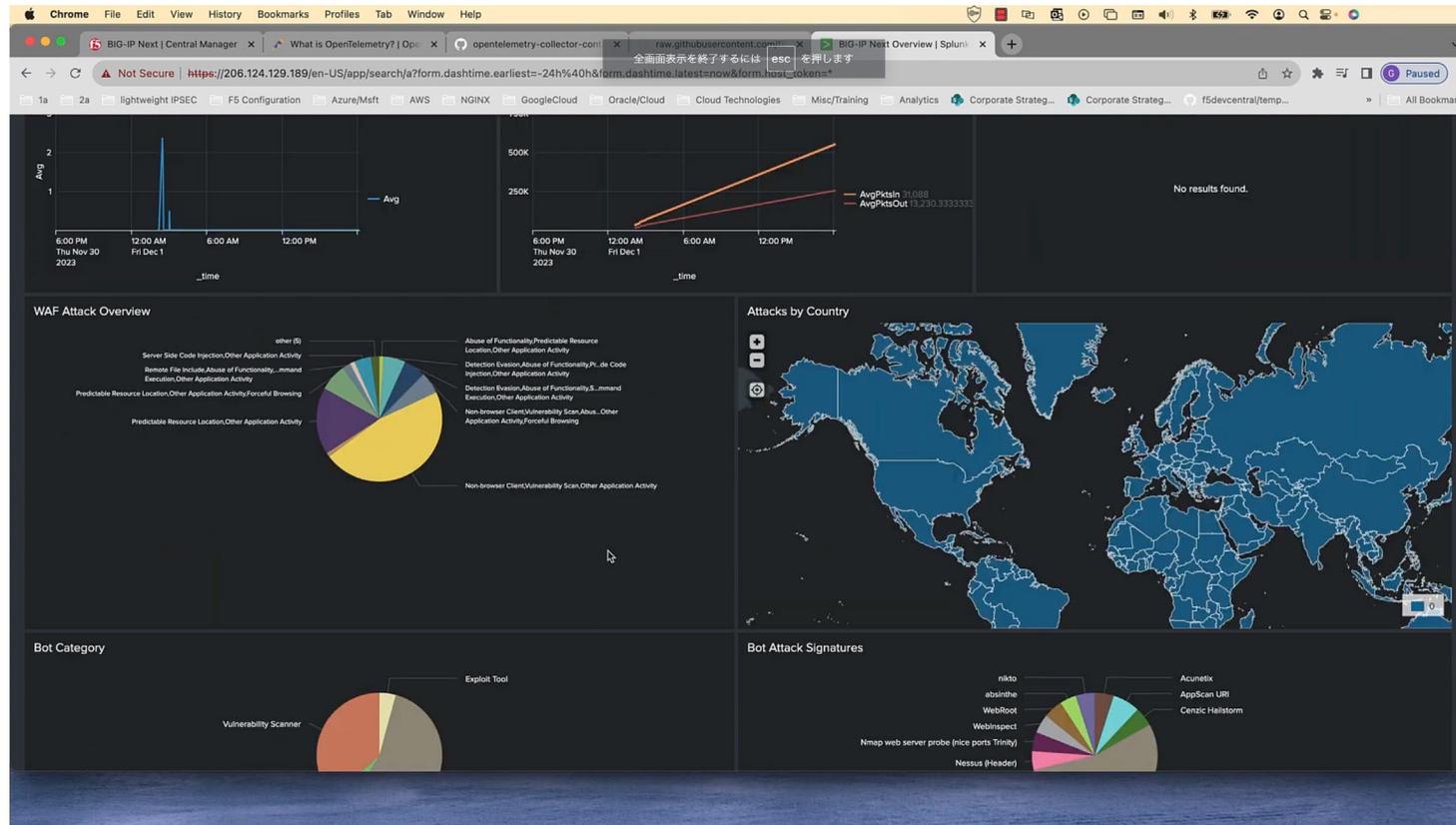
```
resource "bigipnext_cm_as3_deploy" "test" {  
  target_address = "10.xxx.xxx.xxx"  
  as3_json       = <<EOT  
{  
  "class": "ADC",  
  "schemaVersion": "3.45.0",  
  "id": "example-declaration-01",  
  "label": "Sample 1",  
  "remark": "Simple HTTP application with round robin pool",  
  "next-cm-tenant01": {  
    "class": "Tenant",  
    "next-cm-app01": {  
      "class": "Application",  
      "template": "http",  
      "serviceMain": {  
        "class": "Service_HTTP",  
        "virtualAddresses": [  
          "10.0.12.10"  
        ],  
        "pool": "next-cm-pool01"  
      },  
      "next-cm-pool01": {  
        "class": "Pool",  
        "monitors": [  
          "http"  
        ],  
        "members": [  
          {  
            "servicePort": 80,  
            "serverAddresses": [  
              "192.0.2.100",  
              "192.0.2.110"  
            ]  
          }  
        ]  
      }  
    }  
  }  
}  
EOT  
}
```

# BIG-IP Nextのモニタリング

例: OpenTelemetry (Splunk)

- BIG-IP Next Central ManagerでCollector (Log Consumer)の設定を行い、Splunkで可視化
- 参考: “Remote logging with F5 BIG-IP Next and OpenTelemetry”

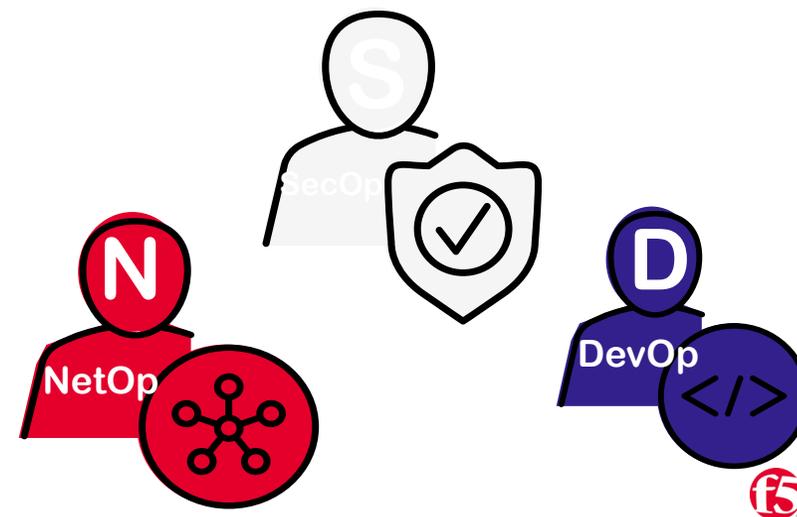
<https://www.youtube.com/watch?v=94nEx2DQpWk>



# BIG-IP Next: まとめ

## • BIG-IP Nextとは・・・

- ✓ 一新したソフトウェア・アーキテクチャによる、次世代に向けて進化したBIG-IP
- ✓ 設定/管理をシンプルにしつつ、パフォーマンス/セキュリティを向上
- ✓ NetOps/SecOps/DevOpsそれぞれをサポートする「ワークフロー中心」「APIフレンドリー(宣言型API)」のアプローチ
- ✓ 複雑なマルチクラウド環境への対応
- ✓ 長年培ってきた、中核となるBIG-IPの機能を保持





**App  
World  
2024**