

ACI-APIC Interfaces

Muhammad Isa Dadi, Aulia Amirullah

Abstrak

Studi ini mengenai Application Centric Infrastructure(ACI), khususnya Cisco APIC (Application Policy Infrastructure Controller) Interfaces dimana kita dapat membuktikan identifikasi antarmuka utama APIC Controller, eksekusi command dasar APIC Controller dengan menggunakan Command Line Interface, eksekusi command dasar APIC Controller dengan menggunakan Application Program Interface, cara komunikasi Graphic User Interface dengan controller.

Keywords : APIC interfaces, ACI, REST APIs, Cisco, Graphic User Interface, APIC Controller

1. Pendahuluan

Pembangunan *network* ditujukan agar satu aplikasi dapat berkomunikasi dengan aplikasi lainnya. Terminologi yang dikenal dalam jaringan saat ini adalah VLAN, subnet, protocol dan port . Terminologi dalam aplikasi biasanya berupa tier, relasi provider/konsumer. Misal, aplikasi A memiliki 3 tier, front end web, app, dan db. Ketiga tier tersebut memiliki relasi satu sama lain dengan kebutuhan spesifik. Relasi tersebutlah yang akan diberikan oleh infrastruktur jaringan yang menjadi pondasi dari sebuah aplikasi.

Infrastruktur yang mendasari sebuah aplikasi adalah network, compute, dan storage. Network berguna sebagai penghubung antar aplikasi. Compute merupakan processing power yang menjalankan aplikasi sedangkan storage menjadi tempat penyimpanan data hasil pemrosesan aplikasi atau data yang menjadi bahan baku pemrosesan. Ketiga komponen ini merupakan resource pool bagi sebuah aplikasi.

Namun, untuk menjalankan infrastruktur tersebut harus ada arsitektur yang dapat menyederhanakan, mengoptimasi dan mempercepat life-cycle dari pembangunan aplikasi. Harus ada platform dimana kebutuhan aplikasi dapat menentukan konfigurasi *network*.

Untuk mengatasi masalah di atas, Cisco menyediakan APIC Controller yang dapat mengendalikan resource pool dari sebuah aplikasi. Cisco Application Centric Infrastructure dapat menjadi platform dimana kebutuhan aplikasi dapat menentukan konfigurasi network. APIC menyediakan akses terpusat ke semua *fabric information*, mengoptimasi dan

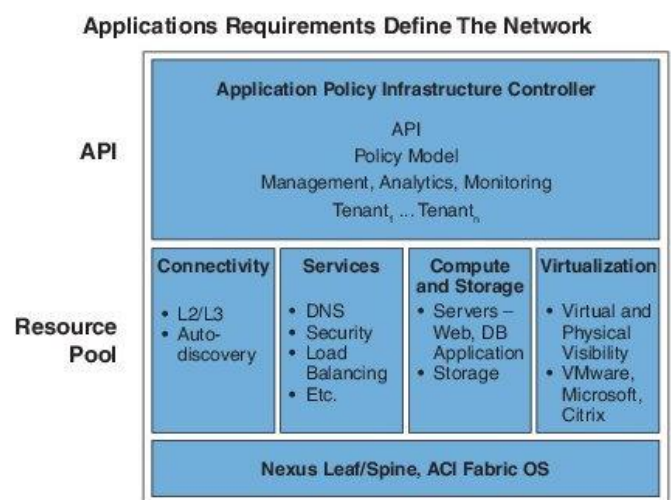
mempercepat *life-cycle* aplikasi dalam pembangunan aplikasi, dan mendukung fleksibilitas penyediaan aplikasi melalui *physical resource* maupun *virtual resource*.

2. Metodologi Riset

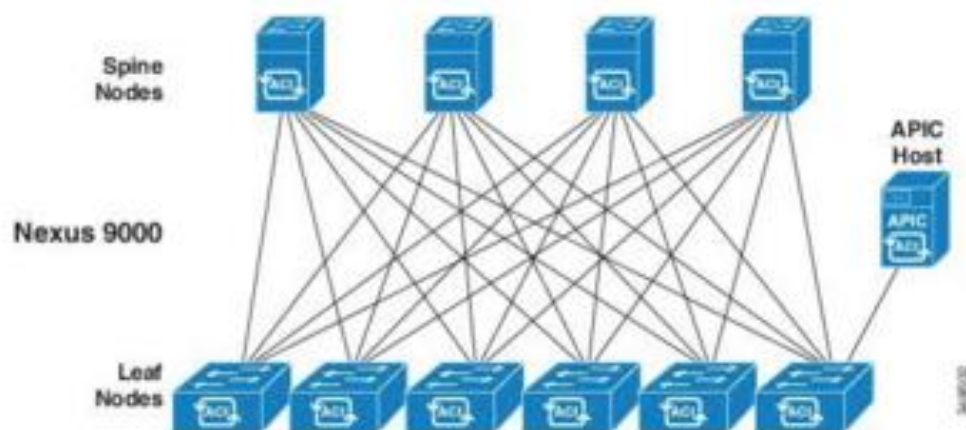
Pada bagian ini, akan dibahas mengenai metode APIC Interface CISCO dalam pembangunan jaringannya. Cisco APIC sendiri memiliki fitur dan kapabilitas khusus dibandingkan software lainnya dalam bidang *data center* dan SDN. Berikut ini fitur dan kapabilitasnya :

- Mesin application-level policy terpusat untuk infratraktur fisik, virtual maupun cloud.
- Visibilitas, telemetry, dan skor kesehatan yang detail dari aplikasi dan tenant.
- Didesain untuk open standard dan open APIs.
- Mempercepat implementasi keamanan multi-tenant, Quality of Service (QoS), dan ketersediaan yang tinggi (*high availability*).
- Integrasi dengan *management system* seperti VMware, Microsoft, dan OpenStack.

Didesain untuk otomasi, *programmability*, dan manajemen terpusat, Cisco APIC sendiri mengekspos northbound APIs melalui XML dan JSON. Cisco APIC menyediakan command-line interface (CLI) dan GUI yang memanfaatkan APIs untuk mengatur fabric secara holistic. Cisco APIC juga menyediakan *single pane of glass* dari *application-centric network policy*, manajemen dan inventori fabric image, aplikasi, tenant dan monitoring topologi, serta *troubleshooting*.

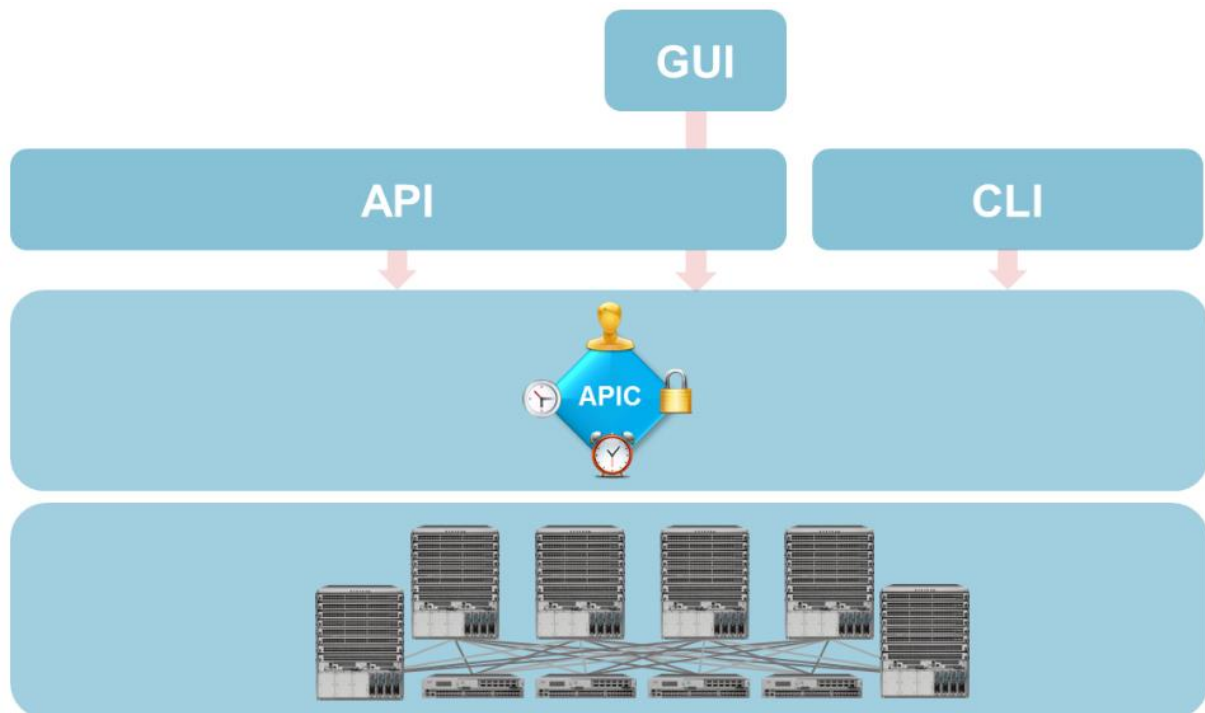


Beginilah application requirements define The Network APIC dari Cisco dengan resource pool dan API serta OS integration. Semua komponen fabric – spine switch dan leaf switch – seperti *network service* dan fitur VM management tertentu dapat disediakan dan diatur hanya dari APIC. Ketika operator dapat terhubung dengan node fabric, salah satu dari *key feature* ACI adalah menyediakan single access point via APIC Controller untuk status, penyediaan, konfigurasi dan troubleshooting, penyederhanaan network management.



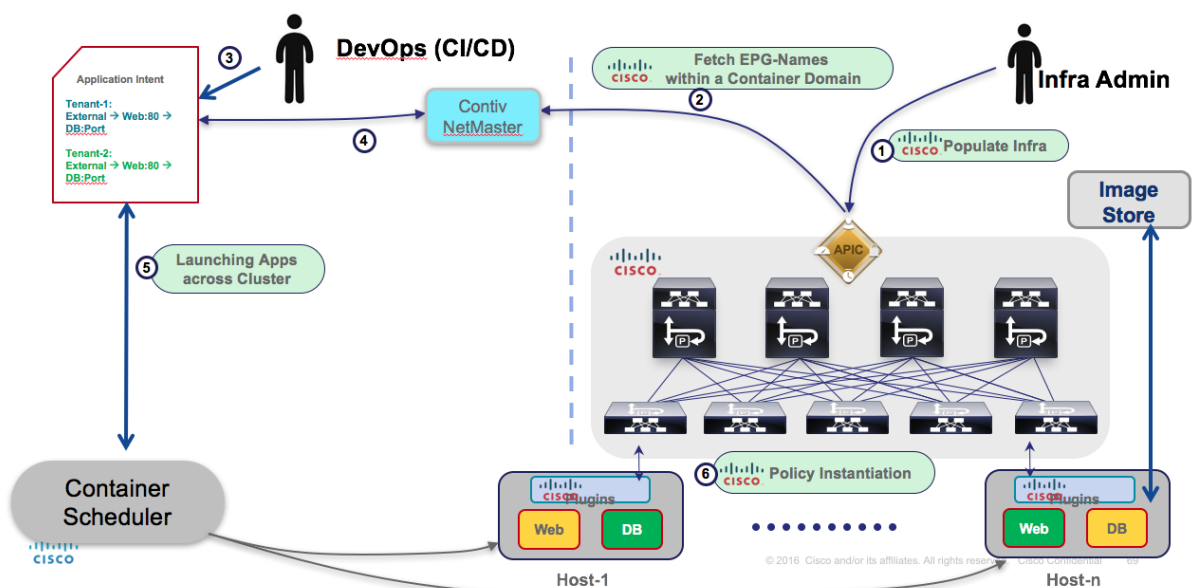
Ada tiga cara utama bagi pengguna untuk mengakses APIC controller yakni Command Line Interface (CLI), Application Program Interface (API), dan Graphic User Interface (GUI). Walaupun ada tiga cara untuk mengakses APIC, hanya ada dua interface yakni CLI dan API sedangkan GUI sebenarnya aplikasi Web-based yang dihosting pada APIC yang mempengaruhi API ke interface dengan internal data model yang terletak di controller.

Berikut ini skema CLI dan API module :



Ini adalah sekma physical network integration dengan ACI.

ACI Integration – Infrastructure Driven



2.1. Command Line Interface

Command dan fitur ACI dibedakan dari switch OS tradisional, seperti IOS atau NX-OS, namun metoda interaksinya sama. Untuk menyoroiti interface ini, kita masuk ke dalam APIC dan membuat contoh VRF yang baru. Dalam ACI, VRF dikenal sebagai Private Network dan berada di dalam sebuah tenant. Tenant dalam ACI diartikan secara logis sebagai ruang terpisah untuk network dan network administration. Tiap tenant memiliki jaringannya

masing-masing dalam contoh Virtual Routing Forwarding terpisah. Pada saat yang bersamaan perlu dibuat sebuah VRF kosong dalam sebuah tenant untuk menunjukkan penggunaan CLI.

CLI versi terbaru memungkinkan network engineer untuk troubleshooting dan menyediakan ACI dalam penggunaan interface familiar dan well-understood.

2.2. Application Program Interface

Sama seperti CLI, API mengekspos full access ke APIC dalam pengkonfigurasi, management, operasional, troubleshooting dan lainnya. APIC API sendiri merupakan REST-based yang menawarkan sejumlah keuntungan dibandingkan CLI ketika membuat script dan program agar dapat berinteraksi dengan ACI.

Kita dapat menunjukkan interaksi APIC dan API dengan cara menciptakan tenant lain dan VRF lain dalam fabric. Ada beberapa tool yang dapat digunakan untuk mengakses command line. Interface device, ada beragam pilihan untuk akses API. API sendiri dapat diakses dari script atau program.

Dalam studi ini, digunakan REST client untuk melihat secara eksplisit API call dan respon API serta menggunakan Postman, Google Chrome browser plugin. REST yang digunakan bukan sebagai protocol tapi sebagai framework, tidak ada dua RESTful APIs yang identic. Walaupun begitu, ada banyak kesamaan diantara dua APIs ini dan sedikit perbedaan yang akan diarahkan dalam penggunaan Postman dalam studi ini.

3. Result and Discussion

3.1. Command Line Interface

Setelah melakukan konfigurasi dengan tahap-tahap berikut :

- Pembangunan SSH session dengan salah satu IP Address yang ada dalam APIC Cluster.
- Pembuatan tenant baru sebagai managed object baru dan pengkonfigurasi.
- Pengubahan Direktori dalam APIC menjadi private-network folder(VRF) serta pengkonfigurasi ke controller.

Maka didapatkan hasil bahwa tenant dan private network telah terbuat.

```
ls /aci/tenants/ExampleCliTenant/networking/private-networks/
```

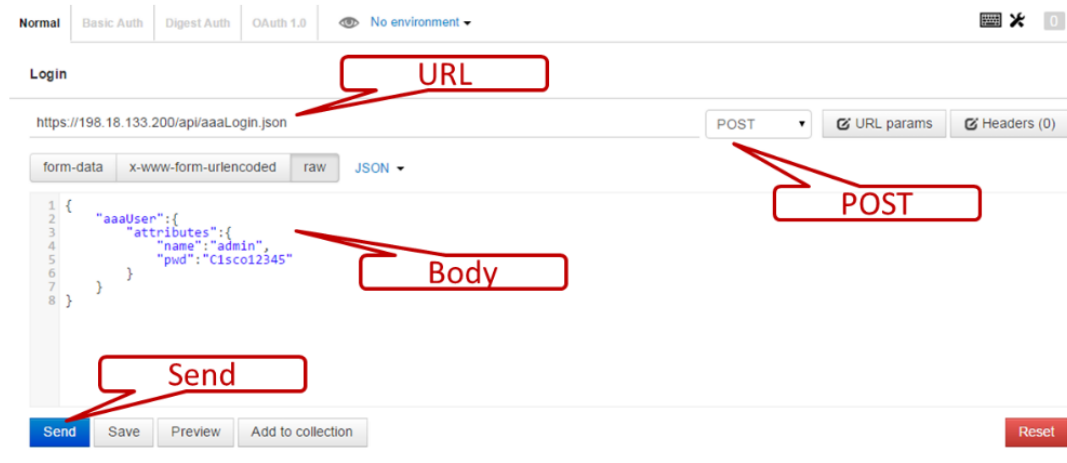
```
admin@apic1:/> show tenant
# Executing command: ls /aci/tenants/
ExamplecliTenant mgmt common infra
admin@apic1:/>
```

3.2. Application Program Interface

Setelah melakukan percobaan dengan tahap berikut :

- Login ke APIC (membuka Postman browser add-on dalam pengeksekusian RESTful API call)
- REST API mendukung format XML (Extensible Markup Language) maupun JSON dalam pembuatan tenant baru dan VRF.
- Membuat tenant dan VRF dengan API
- Pengkonfirmasi Tenant baru dan VRF dengan API

```
POST https://<IP-of-APIC>/api/aaaLogin.json
{
  "aaaUser" : {
    "attributes" : {
      "name" : "username-for-APIC",
      "pwd" : "password-for-APIC"
    }
  }
}
```



Gambar di atas mengindikasikan Postman Json dalam login ke APIC serta diagram yang menunjukkan bagaimana memasukkan REST call ke tool yang tersedia serta diagram tersebut mengindikasikan proses eksekusi program dan status code.

Ada sejumlah status Codes yang ditunjukkan ketika menjalankan REST framework yakni :

- Bad Request (400): Typically indicating a problem with the initial call.
- Not Found(404): The server cannot be found.
- Internal Server Error(500) : The server encountered an internal error.
- Service Unavailable (503) : The server is busy and/or the operation timed out.

```
POST https://<IP-of-APIC>/api/node/mo.json

{
  "polUni": {
    "attributes": {}
    "children": [{
      "fvTenant": {
        "attributes": {
          "name": "ExampleApiTenant"
        }
        "children": [{
          "fvCtx": {
            "attributes": {
              "name": "myVRF"
            }
          }
        ]
      }
    ]
  }
}
```

Code ini menciptakan tenant baru dan VRF dengan API sehingga tercipta private network baru.

Nantinya, setelah melakukan pengecekan itu maka akan didapatn hasil telah terbentuk tenant dan private network baru seperti berikut :

```
1 {
2   "totalCount": "1",
3   "imdata": [
4     {
5       "fvCtx": {
6         "attributes": {
7           "childAction": "",
8           "descr": "",
9           "dn": "uni/tn-ExampleApiTenant/ctx-myVRF",
10          "knwMcastAct": "permit",
11          "lcOwn": "local",
12          "modTs": "2015-10-13T00:57:20.843+00:00",
13          "monPolDn": "uni/tn-common/monepg-default",
14          "name": "myVRF",
15          "ownerKey": "",
16          "ownerTag": "",
17          "pcEnfPref": "enforced",
18          "pcTag": "16386",
19          "scope": "2359296",
20          "seg": "2359296",
21          "status": "",
22          "uid": "15374"
23        }
24      }
25    ]
26  }
27 }
```

Distinguished Name

3.3. Graphic User Interface

Interface terakhir dari APIC adalah GUI. GUI merupakan web-based program yang memanfaatkan API untuk mengakses APIC.

4. Kesimpulan

Berdasarkan studi ini dapat diambil berbagai kesimpulan mengenai APIC Interfaces yang dibangun oleh Cisco dalam data center dan SDN. Pertama, APIC Controller dapat menyediakan resource pool dari sebuah aplikasi. Cisco Application Centric Infrastructure dapat menjadi platform dimana kebutuhan aplikasi dapat menentukan konfigurasi network. APIC menyediakan akses terpusat ke semua *fabric information*, mengoptimasi dan mempercepat *life-cycle* aplikasi dalam pembangunan aplikasi, dan mendukung fleksibilitas penyediaan aplikasi melalui *physical resource* maupun *virtual resource*.

5. Referensi

- [1]<https://learninglabs.cisco.com/lab/aci-101-apic-interfaces/step/3>
- [2]<http://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-apic/index.html>
- [3]http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/aci-fundamentals/b_ACI-Fundamentals.html