



Cisco ACI and Cloud Foundry Integration

[New and Changed Information](#) 2

[Trademarks](#) ?

Revised: June 21, 2022

New and Changed Information

The following table provides an overview of the significant changes up to this current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

Cisco APIC Release Version	Feature	Description
Release 3.1(2)	Support for open-source Cloud Foundry in Cisco ACI	This release enables the deployment of Cloud Foundry in the Cisco ACI fabric.

Cisco ACI and Cloud Foundry Integration

Cloud Foundry is a platform as a service (PAAS) that uses Linux containers to deploy and manage applications. It works as an overlay on various infrastructure systems like VMware vSphere and Amazon Web Services (AWS) and operates on the underlying network used by these systems.

Beginning with Cisco APIC Release 3.1(2), Cloud Foundry is integrated with Cisco Application Centric Infrastructure (ACI). This enables customers to use all Cisco ACI security and policy features with Cloud Foundry containers.

In the Cisco APIC Release 3.1(2), Cisco ACI integration applies to Cloud Foundry deployed on VMware vSphere where the Cisco ACI provides the network fabric for VMware vSphere.

This document is a guide to deploying open-source Cloud Foundry integrated with Cisco ACI and describes the use of Cisco ACI-specific extensions to Cloud Foundry.

Preparation

Cloud Foundry Compatibility

Cloud Foundry is compatible with the following software:

- Cisco APIC Release 3.1(2)
- Cloud Foundry cf-deployment 1.29.0 on Ubuntu Trusty
- Cisco ACI add-ons 0.2.0



Note This document does not include deployment of isolation segments.

Preparing for Cloud Foundry ACI Integration

The following tasks must be completed before you can integrate Cloud Foundry with the Cisco ACI.

Before you begin

It is assumed that you have completed the following tasks:

- Set up the Cisco ACI fabric to use with a VMware vCenter deployment.

See the [Cisco ACI and Cisco APIC documentation](#) on Cisco.com.

- Set up the node subnet as a private subnet and made sure that it has access to the Cisco APIC management address.
- Deployed BOSH Director and Cloud Foundry components to the same Cisco ACI endpoint group (EPG).
- Read and understood the guidelines in the knowledge base article [Cisco ACI and OpFlex Connectivity for Orchestrators](#).

Procedure

- Step 1** Create a VMware VMM domain in Cisco APIC that uses the desired VMware vCenter data center.
- Step 2** Ensure that you have an Attachable Entity Profile (AEP) in Cisco APIC that enables communication on the switch ports that are connected to ESXi hypervisors.
- Step 3** Create a VRF that to hold all your endpoints (BOSH Director, Cloud Foundry component VMs, and containers).
- Step 4** Create and provision L3Out in Cisco APIC for external communication and associate it with the VRF you created in the previous step.
- Step 5** Create an external network under the L3Out that allows traffic to all destinations (0.0.0.0/0).
- Step 6** Create a working directory and extract the Cisco ACI add-ons distribution file (`dist-generics-cloudfoundryxxxxxxx.tgz`).
- Step 7** Install the `acc_provision` Debian package.
- Step 8** Ensure that your machine is a development machine; you will use this machine for compiling source code during BOSH deployment.
- See "Before you Begin" in the section "Deploying BOSH Director" for the list of required packages.
- Step 9** Ensure that Ruby 2.4.1 or higher is installed on your machine. If you have older version installed, remove them (packages `ruby` and `rubydev`) and install a newer version of Ruby.
- For example: <https://cache.ruby-lang.org/pub/ruby/2.4/ruby-2.4.1.tar.gz>
- Step 10** Download and install [BOSH \(v2\) CLI](#).
- Step 11** Install the [cf CLI Debian package](#).
- Step 12** Get source code for `bosh-deployment` and `cf-deployment`:
- ```
git clone https://github.com/cloudfoundry/bosh-deployment
git clone https://github.com/cloudfoundry/cf-deployment
```
- Step 13** If your machine requires a proxy to reach the Internet, set the `no_proxy` environment variable for the following: VMware vCenter server, Cisco APIC server, the BOSH Director IP you will choose (in the section Provisioning ACI for CloudFoundry) and the system domain you will choose section Deploying Cloud Foundry with Cisco ACI Add-Ons).
- For the examples in this document this environmental variable is set as:
- ```
no_proxy=172.28.184.150,172.28.184.85,10.1.0.2,mycf0.fab15.local
```
-

Deployment

Provisioning Cisco ACI to Work with Cloud Foundry

Before you begin

Ensure that you have completed the tasks in the section "Preparation for Cisco ACI Deployment for Cloud Foundry" in this guide.

Procedure

Step 1 Create a provisioning config file, using the following example.

Change the values in the example to fit your environment.

Example:

```
## Configuration for ACI Fabric
aci_config:
system_id: mycf0 # Unique ID for this install
apic_hosts: # List of APIC hosts to connect to
172.28.184.150
apic_login:
username: admin
password: myadminpassword
vmm_domain: # CloudFoundry VMM domain config
encap_type: vxlan # Encap mode: vxlan or vlan
mcast_range: # Every VMM must use a distinct range
start: 225.20.1.1
end: 225.20.255.255
nested_inside:
type: vmware
name: fab15vds1 # Your VMware VMM domain name
# The following resources must already exist on the APIC,
# they are used, but not created by the provisioning tool.
aep: esxaep # The AEP for ports/VPCs
vrf: # VRF to place the endpoints in
name: l3out_1_vrf
tenant: common
l3out:
name: l3out1 # Used for external communication
external_networks:
- l3out_1_net # Used for external contracts
#
# Networks used by CloudFoundry
#
net_config:
node_subnet: 10.1.0.1/16 # Subnet for CloudFoundry nodes
pod_subnet: 10.2.0.1/16 # Subnet for container IPs
extern_dynamic: 150.3.0.1/24 # Subnet for dynamic external IPs
extern_static: 150.4.0.1/24 # Subnet for static external IPs
node_svc_subnet: 10.5.0.1/24 # Subnet for service graphs
service_vlan: 4002 # VLAN used by LoadBalancer services
infra_vlan: 4093 # VLAN used by ACI infra
```

In the preceding example, node subnet 10.1.0.0/16 will be used for the BOSH Director virtual machine (VM) and Cloud Foundry component VM. Reserve three IP addresses within this range for the following VMs:

- BOSH Director—one address; for example, 10.1.0.2
- GoRouter—two addresses; for example, 10.1.0.3 and 10.1.0.4

Step 2 Configure Cisco APIC and generate a configuration file for cf-deployment.

Example:

```
acc_provision -a -u <apic username> -p <apic password> -c mycf0-prov-config.yaml -o mycf0-vars.yaml  
-f cloudfoundry-1.0
```

This command configures Cisco APIC for Cloud Foundry and generates a file called `mycf0-vars.yaml`.

Step 3 Make a note of the following values in the file `mycf0-vars.yaml`: `apic_dvs` and `apic_node_portgroup`.

Deploying BOSH Director

Before you begin

You need to have several packages in order to deploy BOSH Director:

build-essential	libssl-dev
curl	libxslt-dev
git-core	libyaml-dev
libreadline6	openssl
libreadline6-dev	sqlite3
libsqlite3-dev	wget
libxml2-dev	zlib1g-dev
libxslt-dev	zlibc

Procedure

Step 1 Create a file containing BOSH Director deployment parameters, `boshvars.yaml`.

Ensure that `internal_cidr` and `internal_gw` match the values you picked earlier for the node subnet. Also, ensure that `internal_ipis` set to the address you reserved for BOSH Director.

Example:

```
director_name: fab15bosh # Pick a name  
internal_cidr: 10.1.0.0/16 # Must match node_subnet  
internal_gw: 10.1.0.1  
internal_ip: 10.1.0.2 # Reserved IP for BOSH Director  
network_name: "fab15vds1/  
mycf0|cloudfoundry|cfnode" # From mycf0vars.yaml, format is apic_dvs/apic_node_portgroup  
vcenter_dc: CF # vCenter datacenter to use  
vcenter_ds: datastore1 # vCenter datastores to use  
vcenter_ip: 172.28.184.85 # vCenter IP  
vcenter_user: root  
vcenter_password: vmware  
vcenter_templates: fab15boshtemplates # Pick a name  
vcenter_vms: fab15boshvms # Pick a name  
vcenter_disks: fab15boshdisks # Pick a name  
vcenter_cluster: cfcluster # vCenter cluster to use
```

```
# Following lines are required if your environment requires a proxy to reach the Internet
http_proxy: http://proxywsa.esl.cisco.com:80 # HTTP proxy
https_proxy: http://proxywsa.esl.cisco.com:80 # HTTPS proxy
no_proxy: 172.28.184.85,10.1.0.2 # vcenter_ip and internal_ip specified earlier

# Following line is required if your environment does not allow access to public DNS servers like
8.8.8.8
internal_dns: [171.70.168.183, 8.8.8.8] # List of DNS servers
```

Step 2 Deploy BOSH Director.

Example:

```
bosh create-env bosh-deployment/bosh.yml \
-o bosh-deployment/vsphere/cpi.yml \
-o bosh-deployment/jumpbox-user.yml \
-l boshvars.yml \
--state=state.json \
--vars-store=creds.yml
```

If your environment requires a proxy to access the Internet, also include this parameter: `-o bosh-deployment/misc/proxy.yml`.

If your environment doesn't allow name resolution using public DNS servers, also include this parameter: `-o bosh-deployment/misc/dns.yml`.

Step 3 Create an alias for the BOSH Director for ease of use using the following command: `bosh alias -env <aliasname> -e <Director IP> --ca cert "$(bosh int creds.yml --path /director_ssl/ca)"`.

Example:

```
bosh alias-env fab15bosh -e 10.1.0.2 --ca cert "$(bosh int creds.yml path /director_ssl/ca)"
```

Step 4 Set up a few environment variables to use with BOSH CLI.

Example:

```
export BOSH_ENVIRONMENT=<aliasname>
export BOSH_CLIENT=admin
export BOSH_CLIENT_SECRET=`bosh int creds.yml path /admin_password`
```

`bosh env`

You should

Step 5 Verify that you have a successful deployment by entering the following command: `bosh env`

Output appears similar to the following example:

```
Using environment '10.1.0.2' as client 'admin'
Name      fab15bosh
UUID      48cd341ed0be4d6da2aa7fa76b62d985
Version   262.3.0 (00000000)
CPI       vsphere_cpi
Features  compiled_package_cache: disabled
          config_server: disabled
          dns: disabled
          snapshots: disabled
User      admin
Succeeded
```

Deploying Cloud Foundry with Cisco ACI Add-Ons

Procedure

Step 1 Set up BOSH cloud configuration by creating a file, `mycf0-cloud-config.yml`.

Example:

```
azs:
- name: z1
  cloud_properties:
    datacenters:
      - clusters: [cfcluster: {}] # vCenter cluster to use
name: z2
  cloud_properties:
    datacenters:
      - clusters: [cfcluster: {}] # vCenter cluster to use
name: z3
  cloud_properties:
    datacenters:
      - clusters: [cfcluster: {}] # vCenter cluster to use

vm_types:
- name: default
  cloud_properties:
    cpu: 2
    ram: 1024
    disk: 3240
- name: large
  cloud_properties:
    cpu: 2
    ram: 4096
    disk: 20_240
- name: small
  cloud_properties:
    cpu: 1
    ram: 4096
    disk: 4096
- name: minimal
  cloud_properties:
    cpu: 2
    ram: 8192
    disk: 30_240
- name: small-highmem
  cloud_properties:
    cpu: 4
    ram: 10240
    disk: 51200
- name: sharedcpu
  cloud_properties:
    cpu: 1
    ram: 2048
    disk: 3240

disk_types:
- name: default
  disk_size: 3000
- name: large
  disk_size: 50_000
- disk_size: 1024
  name: 1GB
- disk_size: 5120
```

```

name: 5GB
-disk_size: 10240
name: 10GB
-disk_size: 100240
name: 100GB

vm_extensions:
- name: 5GB_ephemeral_disk
- name: 10GB_ephemeral_disk
- name: 50GB_ephemeral_disk
- name: 100GB_ephemeral_disk
- name: 500GB_ephemeral_disk
- name: 1TB_ephemeral_disk
- name: ssh-proxy-and-router-lb
name: diego-ssh-proxy-networkproperties
cloud_properties:
  ports:
    - host: 80
    - host: 443
    - host: 2222
name: cf-tcp-router-network-properties
name: cf-router-network-properties
cloud_properties:
  ports:
    - host: 10241123
    - host: 80
    - host: 443
    - host: 2222

networks:
- name: default
  type: manual
  subnets:
    - range: 10.1.0.0/16      # internal_cidr from boshvars.yml
      gateway: 10.1.0.1     # internal_gw from boshvars.yml
      azs: [z1, z2, z3]
      dns: [171.70.168.183, 8.8.8.8] # Use internal_dns from boshvars.yml, else [8.8.8.8]
      reserved: []
      static: [10.1.0.3, 10.1.0.4] # Reserved IPs for GoRouter

```

Step 2 Update the cloud configuration using the following command:

```

bosh updatecloudconfig \
mycf0-cloud-config.yml \
-o manifest-generation/cloud_config_ops.yml
-l mycf0-vars.yaml

```

Step 3 Upload the required stemcell to BOSH Director.

```

export STEMCELL_VERSION=$(bosh int cf-deployment/cf-deployment.yml
--path /stemcells/alias=default/version)

bosh upload-stemcell
https://bosh.io/d/stemcells/bosh-vsphere-esxi-ubuntu-trusty-go_agent?
v=$STEMCELL_VERSION

```

Step 4 Upload the Cisco ACI add-ons BOSH release file to BOSH Director.

```

bosh upload-release release/aci-containers-release0.1.0-beta1.tar.gz

```

Step 5 Choose a DNS name (system domain) for your deployment and ensure that this name resolves to the reserved IP addresses you chose for GoRouter. (For example, 10.1.0.3 and 10.1.0.4)

Step 6 Ensure that wildcard DNS resolution is allowed.

That is, if your system domain is `mycf0.fab15.local`, then all names like `*.mycf0.fab15.local` will resolve to the GoRouter's address.

Step 7 Create a cf-deployment operations file, `router-static-cf.yml` to assign a static address to the GoRouter.

```
---
- type: replace
  path:
  /instance_groups/name=router/networks/name=default/static_ips?
  # Replace the addresses below with IPs reserved for GoRouter
  value: ["10.1.0.3", "10.1.0.4"]
```

Step 8 Deploy Cloud Foundry.

Example:

```
bosh deploy -d cf cf-deployment/cf-deployment.yml \
-o manifest-generation/cf_ops.yml \
-o router-static-cf.yml \
--vars-store=mycf0-vars-store.yml \
-l mycf0-vars.yml \
-v system_domain=<your-system-domain>
```

Remember to replace *your-system-domain*. The deployment can take a while.

Step 9 Verify that CloudFoundry has been deployed successfully.

```
cf login --skip-ssl-validation -a https://api.your-system-domain-u admin -p $(bosh int
mycf0-vars-store.yml path /cf_admin_password)
```

Output appears similar to the following example:

```
API endpoint: https://api.your-system-domain
Authenticating...
OK

Targeted org system
API endpoint: https://api.<yoursystemdomain> (API version:2.99.0)
User: admin
Org: system
Space: No space targeted, use 'cf target -s SPACE'
```

Removing Cisco ACI Add-ons from Cloud Foundry

Installed Cisco ACI add-ons can be removed by running the command in this section. After you run the command, the Cloud Foundry deployment uses Cisco ACI as a pure underlay.

Procedure

Remove the Cisco ACI add-ons.

Example:

```
bosh deploy -d cf cf-deployment/cf-deployment.yml \
-o router-static-cf.yml \
--vars-store=mycf0-vars-store.yml \
-l mycf0-vars.yml \
-v system_domain=<your-system-domain>
```

Unprovisioning Cloud Foundry from the ACI Fabric

This section describes how to unprovision Cloud Foundry from the ACI fabric.

Before you begin

Before unprovisioning the resources allocated to your Cloud Foundry installation from your Cisco ACI fabric, ensure that Cloud Foundry and BOSH Director have been removed.

Procedure

Step 1 Delete Cloud Foundry

Example:

```
bosh delete-deployment -d cf
```

Step 2 Delete BOSH Director.

Example:

```
bosh delete-env bosh-deployment/bosh.yml \  
-o bosh-deployment/vsphere/cpi.yml \  
-o bosh-deployment/jumpbox-user.yml \  
-l bosh-vars.yml \  
--state=state.json \  
--vars-store=creds.yml
```

Step 3 Unprovision the fabric.

Example:

```
acc_provision -a -d -u <apic username> -p <apic password> -c mycf0-prov-config.yaml -o mycf0-vars.yaml  
-f cloudfoundry-1.0
```

Note This command also deletes the Cisco ACI tenant. If you are using a shared tenant, this command is dangerous.

Operations

Using Cisco ACI-Specific Extensions

You can access Cisco ACI-specific Cloud Foundry extensions through a Python CLI script, `cf-aci.py`. Extensions features include EPG annotations and external IP address.

The Python CLI script is in the `scripts/` directory of the distribution files (`dist-generics-cloudfoundryxxxxxx.tgz`). Most commands are self-explanatory and take one or two arguments.

Procedure

Run the Python CLI script to access the Cisco ACI-specific Cloud Foundry extensions, using the following example:

```
./scripts/cf-aci.py --help  
Usage: cf-aci.py <command> <command-arguments>  
Available commands:  
  app-ext-ip <app-name> Get external IP of an app  
  app-vip <app-name> Get virtual IP of an app  
  epg-app <app-name> Get EPG annotation of an app  
  epg-org <org-name> Get EPG annotation of an org
```

```
epg-space <space-name> Get EPG annotation of a space
set-app-ext-ip <app-name> <IP-address> Set external IP on an app
set-epg-app <app-name> <EPG-name> Set EPG annotation on an app
set-epg-org <app-name> <EPG-name> Set EPG annotation on an org
set-epg-space <app-name> <EPG-name> Set EPG annotation on a space
unset-app-ext-ip <app-name> Remove external IP of an app
unset-epg-app <app-name> Remove EPG annotation of an app
unset-epg-org <org-name> Remove EPG annotation of an org
unset-epg-space <space-name> Remove EPG annotation of a space
```

Collecting Log Files for Support Requests

If problems arise, Cisco support may ask that you submit log files to help them troubleshoot the problems. Follow the steps in this section to collect the log files for Cloud Foundry.

Procedure

Step 1 Get the list of VM instances in your deployment.

Example:

```
bosh vms -d cf
```

Step 2 Generate report on the desired VM instance (diego-api or diego-cell).

Example:

```
bosh ssh -d cf <instance> -c 'sudo/var/vcap/packages/apic-host-report/apic-host-report.sh'
```

Step 3 Note the report file mentioned in the output.

Step 4 Copy over the report file.

Example:

```
bosh scp -d cf <instance>:<path/to/report-file> .
```

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: <http://www.cisco.com/go/softwareterms>. Cisco product warranty information is available at <http://www.cisco.com/go/warranty>. US Federal Communications Commission Notices are found here <http://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com go trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2018 Cisco Systems, Inc. All rights reserved.



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.