Use vzAny to Automatically Apply Communication Rules to all EPGs in a VRF

Revised: April 7, 2022

New and Changed Information

Table 1: New Features and Changed Behavior in Cisco APIC

Cisco APIC Release Version	Feature	Description	What Changed
Release 2.2(1n)	Preferred Group Member	Support for `vzAny` marked as a Preferred Group Member, allowing communications between EPGs in the group without a contract.	--
Release 1.0(1e)	This feature was introduced.	--	--

What vzAny Is

The `vzAny` managed object provides a convenient way of associating all endpoint groups (EPGs) in a Virtual Routing and Forwarding (VRF) instance to one or more contracts (`vzBrCP`), instead of creating a separate contract relation for each EPG.

In the Cisco ACI fabric, EPGs can only communicate with other EPGs according to contract rules. A relationship between an EPG and a contract specifies whether the EPG provides the communications defined by the contract rules, consumes them, or both. By dynamically applying contract rules to all EPGs in a VRF, `vzAny` automates the process of configuring EPG contract relationships. Whenever a new EPG is added to a VRF, vzAny contract rules automatically apply. The `vzAny` one-to-all EPG relationship is the most efficient way of applying contract rules to all EPGs in a VRF.

Note In the APIC GUI under tenants, a VRF is also known as a private network (a network within a tenant) or a context.

In the case of shared services, you *must* define the provider EPG shared subnet under the EPG in order to properly derive the pcTag (classification) of the destination from the consumer (vzAny) side. If you are migrating from a BD-to-BD shared services configuration, where both the consumer and provider subnets are defined under bridge domains, to vzAny acting as a shared service consumer, you must take an extra configuration step where you add the provider subnet to the EPG with the shared flags at minimum.

Note If you add the EPG subnet as a duplicate of the defined BD subnet, ensure that both definitions of the subnet always have the same flags defined. Failure to do so can result in unexpected fabric forwarding behavior.

To use vzAny, navigate to **Tenants** > ***tenant-name*** > **Networking** > **VRFs** > ***vrf-name*** > **EPG Collection for VRF**.

How vzAny Works

The following scenarios illustrate how vzAny works:

- EPG contract provider and consumer relations without using vzAny
- `vzAny` is the consumer and one EPG is the provider

- `vzAny` is the provider and one EPG is the consumer
- `vzAny` is the provider and the consumer

Figure 1: EPG1 provides FTP to EPG2 and EPG3 without using vzAny

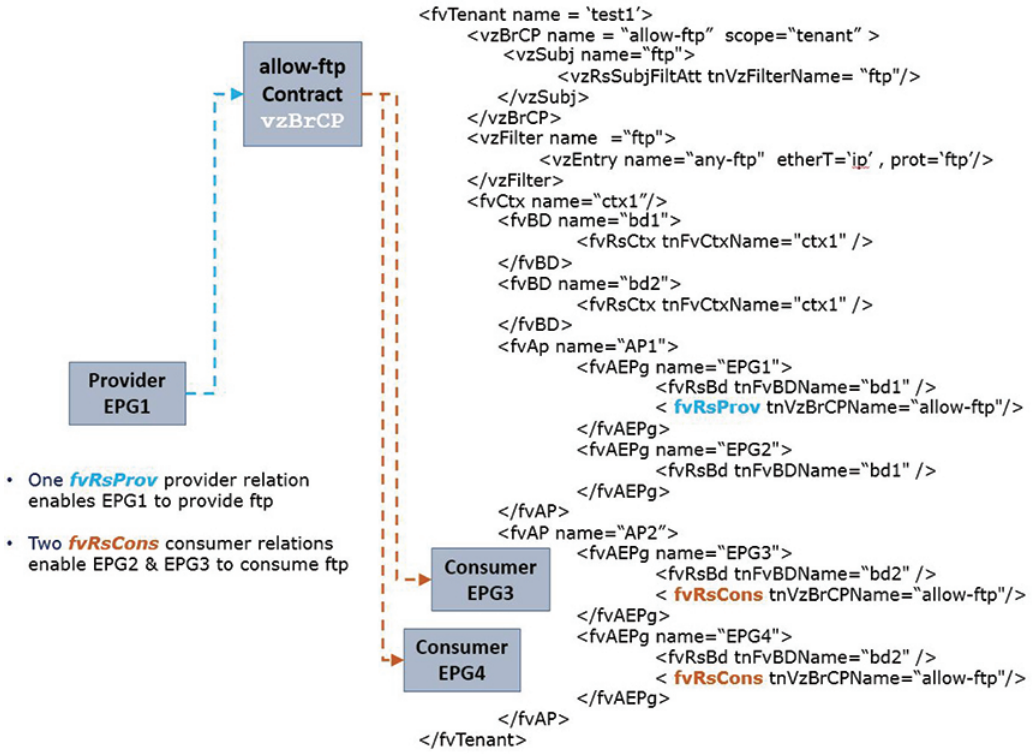
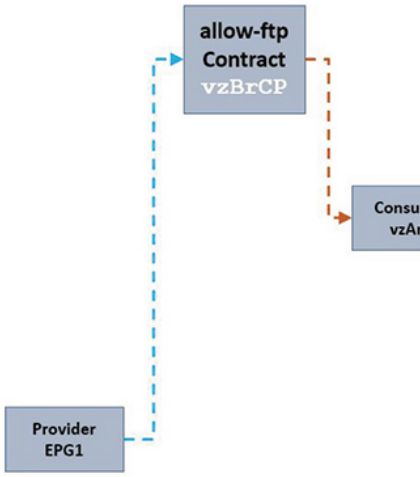


Figure 2: Using vzAny, Any EPG in the Context Consumes the FTP that EPG1 Provides

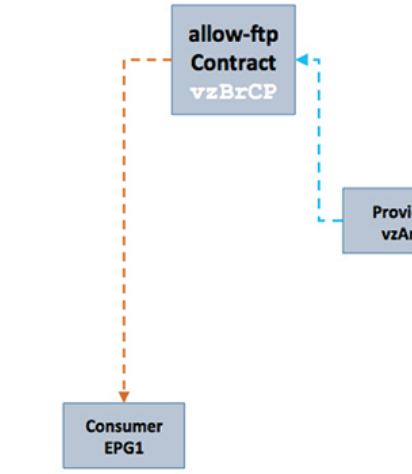



```
<fvTenant name = 'test1'>
    <vzBrCP name = "Allow-FTP"  scope="tenant" >
        <vzSubj name="FTP">
            <vzRsSubjFiltAtt tnVzFilterName= "FTP"/>
        </vzSubj>
    </vzBrCP>
    <vzFilter name  ="FTP ">
        <vzEntry name="any-ftp"  etherT='ip' , prot='FTP'/>
    </vzFilter>
    <fvCtx name="ctx1"/>
        <vzAny>
            <vzRsAnyToCons tnVzBrCPName="Allow-FTP "/>
        </vzAny>

        <fvBD name="bd1">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvBD name="bd2">
            <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name="AP1">
            <fvAEPg name="EPG1">
                <fvRsBd tnFvBDName="bd1" />
                < fvRsProv tnVzBrCPName="Allow-FTP "/>
            </fvAEPg>
            <fvAEPg name="EPG2">
                <fvRsBd tnFvBDName="bd1" />
            </fvAEPg>
        </fvAP>
        <fvAP name="AP2">
            <fvAEPg name="EPG3">
                <fvRsBd tnFvBDName="bd2" />
            </fvAEPg>
            <fvAEPg name="EPG4">
                <fvRsBd tnFvBDName="bd2" />
            </fvAEPg>
        </fvAP>
</fvTenant>
```

- One ***fvRsProv*** relation enables EPG1 to provide FTP
- One ***vsRsAnyToCons*** relation enables **any** EPG in context ctx1 to consume FTP

Figure 3: Using vzAny, Any EPG in the Context Provides the FTP that EPG1 Consumes



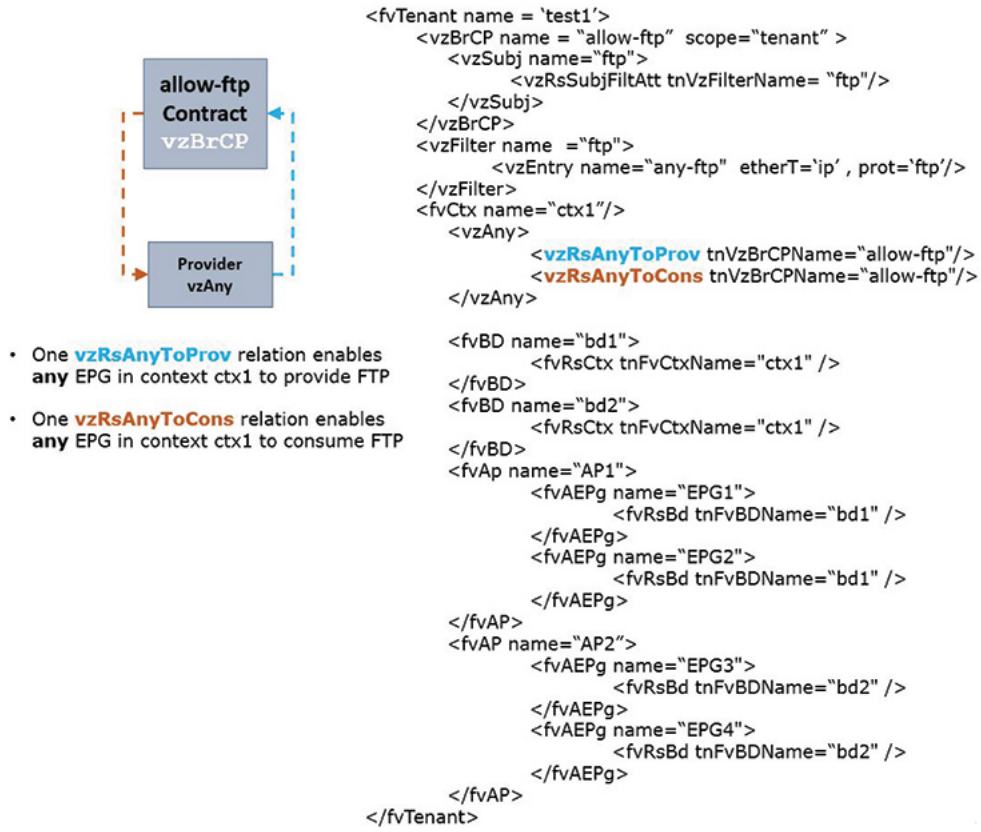

```
<fvTenant name = 'test1'>
    <vzBrCP name = "allow-ftp"  scope="tenant" >
        <vzSubj name="ftp">
            <vzRsSubjFiltAtt tnVzFilterName= "ftp"/>
        </vzSubj>
    </vzBrCP>
    <vzFilter name  ="ftp">
        <vzEntry name="any-ftp"  etherT='ip' , prot='ftp'/>
    </vzFilter>
    <fvCtx name="ctx1"/>
        <vzAny>
        <vzRsAnyToProv tnVzBrCPName="allow-ftp"/>
        </vzAny>

        <fvBD name="bd1">
        <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvBD name="bd2">
        <fvRsCtx tnFvCtxName="ctx1" />
        </fvBD>
        <fvAp name="AP1">
        <fvAEPg name="EPG1">
                <fvRsBd tnFvBDName="bd1" />
                < fvRsCons tnVzBrCPName="allow-ftp"/>
        </fvAEPg>
        <fvAEPg name="EPG2">
                <fvRsBd tnFvBDName="bd1" />
        </fvAEPg>
        </fvAP>
        <fvAP name="AP2">
        <fvAEPg name="EPG3">
                <fvRsBd tnFvBDName="bd2" />
        </fvAEPg>
        <fvAEPg name="EPG4">
                <fvRsBd tnFvBDName="bd2" />              </fvAEPg>

        </fvAP>
</fvTenant>
```

- One **vzRsAnyToProv** relation enables **any** EPG in context ctx1 to provide FTP
- One **fvRsCons** relation enables EPG1 to consume FTP

Figure 4: Using vzAny, Any EPG in the Context can Consumer and Provide FTP



While these scenarios illustrate very simple configurations, large production systems can easily have thousands of contract relationships in a single VRF. In such cases, vzAny could eliminate half or more of the contract relationships. This simplification not only makes the configuration much easier to maintain but can also save on switch TCAM buffer consumption.

vzAny Guidelines and Limitations

Observe the following guidelines when using `vzAny`:

- All EPGs in a VRF can take the role of a provider, a consumer, or both depending on the configuration.
- In addition to application profile EPGs, `vzAny` providers and consumers include external EPGs such as `l2extOut` and `l3extOut`, and endpoint groups for out-of-band (`mgmtOoB`) or in-band (`mgmtInB`) access.
- `vzAny` is supported as a consumer of a shared service but is not supported as a provider of a shared service.
- `vzAny` implicitly permits all subnets. When `vzAny` is in use for a VRF, it also includes the L3out, hence it is equivalent to having created a L3external classification that includes the subnets specified in the VRF itself.
- If an EPG, within a VRF, is consuming a shared service contract from an EPG of a different VRF (that we call provider VRF), the traffic from the EPG of the provider VRF is filtered within the consumer VRF. `vzAny` is equivalent to a wildcard for the source or destination EPG. Be careful when you configure a contract with a `vzAny` in the consumer VRF because the vzAny contracts may also apply to the traffic between the EPG of the provider VRF and the EPG of the consumer VRF.

Note Failure to observe this guideline could allow unintended traffic between EPGs across VRFs.

- Configuring a VRF with `vzAny` as both the provider and the consumer, and with the default `allow all` filter, is the same as configuring an `unenforced` VRF. The `unenforced` VRF is a simpler configuration that produces the same result. In addition, all EPGs within that VRF are free to communicate to each other without a contract. If `vzAny` is marked as a Preferred Group Member, only those EPGs marked as a Preferred Group Member will be able to communicate with each other without a contract. Alternatively, any EPG, within that VRF that is not marked as a Preferred Group Member, will be unable to communicate without a contract

  Note It is not recommended to enable **Preferred Group Member** (marked as **Include**) for vzAny when configuring a VRF with `vzAny` as both the provider and the consumer, and with the default `allow all` filter, due to it then disables the ability for all EPGs within that VRF to communicate without a contract.

- If the contract scope is `application-profile`, the `vzAny` configuration is ignored and filter rules are expanded; CAM utilization is the same as if specific consumers and providers are deployed. In this case, there is no saving in CAM usage.
- Since `vzAny` dynamically enables its contract rules for any current or future EPG in a VRF, good practice is to document such configurations.
- In the case of shared services using inter-VRF contracts, you *must* define the provider EPG shared subnet under the EPG in order to properly derive the pcTag (classification) of the destination from the consumer (vzAny) side. If you are migrating from a BD-to-BD shared services configuration, where both the consumer and provider subnets are defined under bridge domains, to vzAny acting as a shared service consumer, you must take an extra configuration step where you add the provider subnet to the EPG with the shared flags at minimum.

  Note For inter-VRF contracts, if you add the EPG subnet as a duplicate of the defined BD subnet, ensure that both definitions of the subnet always have the same flags defined. Failure to do so can result in unexpected fabric forwarding behavior.

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.