Consul Extension for ACI

About Consul

Consul is a highly distributed service mesh solution by HashiCorp for providing a full-featured control plane with service discovery, configuration, and segmentation functionality at L4-L7. (https://www.consul.io/)

About Consul Extension for ACI

Consul Extension for ACI runs as an application on APIC to enable following use cases with the objective to have service mesh visibility into the network and drive network middleware automation based on service intentions.

Service visibility and faster Mean-time-to-Resolution (MTTR)

- Real-time visibility into dynamic L4-L7 services, service health and service-to-service communication on virtual, container and bare-metal workloads connected by the ACI multi-cloud network.

- Faster identification of issues based on service health and network data correlation.

Network Infrastructure Automation

- Consistent L4-L7 service mesh driven network policy (contracts and filters) automation for virtual, bare-metal and container workloads across private and public cloud for your ACI multi-cloud network.

- Easier transition to a secure service mesh-based deployments for Applications teams and DevOps operators with the ACI multi-cloud network.

Installing Consul

Before you start using Consul Extension for ACI, you must install consul agents and bring up a Consul cluster on the application workloads connected to your ACI multi-cloud network. See the Getting Started Guide (https://www.consul.io/intro/getting-started.html) for details.

Onboarding a Consul Seed Agent

Use this procedure to onboard one or more Consul agents on the application Consul Extension for ACI using the UI.
It is recommended to onboard more than one Consul Servers per Consul Datacenter for HA purposes.
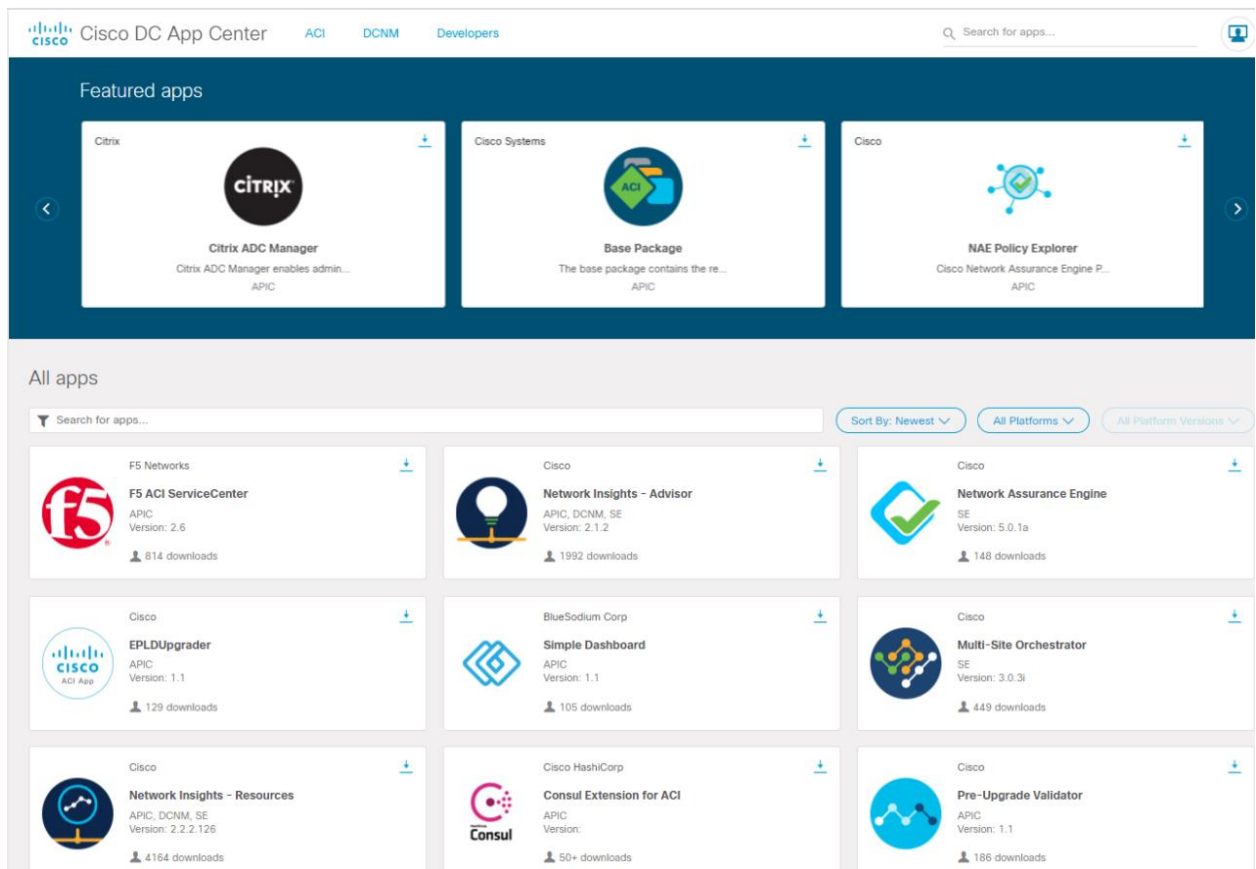
Before you begin

- You should have "admin read-only" privileges associated with the ACI user to install and view the complete functionality of Consul Extension for ACI application.

- There is a non-admin instance of Consul Extension for ACI available with limited functionality, in case you want to install and enable this app for non-admin users. Please reach out to your Cisco or HashiCorp rep for this version or make a "non-admin version request" on https://github.com/ciscoecosystem/consul-aci/issues.

- You must have in-band or out-of-band connectivity from APIC to the Consul agents on TCP ports 8500 and 8501.

- You must have installed Consul agents and brought up a Consul Datacenter.

- It is recommended to add **more than one Consul Server** as the seed agent per Consul Datacenter.

- You should have Consul ACL Token supporting "read" for following resources on the Consul agent to be onboarded. (master token with read privileges for Global Management policy would be ideal)
  - service
  - node
  - agent
  - query

Note:

- Consul Extension for ACI **ONLY READs / GETs** data from Consul. It **DOES NOT WRITE / POST** any data to the Consul cluster.
- Consul Extension for ACI **ONLY READs / GETs** data from APIC. It **DOES NOT WRITE / POST** any data to the APIC.

Procedure

1. Download the **Consul** app from DC App Center.

2. On the APIC select **Apps** tab and click **Upload** icon to upload the Consul app.

3. Once the app is uploaded select the **security domain** in which you want to make the app available.

4. **Enable** the app.

5. Once **Enabled,** the app is installed in the **Tenant** tab on all the Tenants on the APIC

6. Click **Agent** on the left navigation pane.

7. Click **Add Agent** to connect to a new Consul agent. Refer to **Before you begin** for guidelines.

8. Input **Protocol**, unique **Address**, **Port** number, **VRF** and ACL **Token** for the Consul agent.

There is a possibility of having the same IP address across multiple VRFs in a Tenant. Input the appropriate VRF to map it with the seed agent. If no VRF is selected, then Consul Extension for ACI app shall iterate through all the endpoints in a Tenant including endpoints with overlapping IP addresses (across multiple VRFs) and come up with the mappings.

9. Click **Add**

10. Click 'X' on the far right side of the work pane.

About Agents

The **Connected** status indicates that the agent is reachable and active to fetch data. The **Disconnected** status indicates that the agent is not reachable and will not fetch data. You can delete an agent on the **Agent** work pane from **Action**.

You can add multiple Agents for the same Consul Datacenter. Consul Datacenter will be deleted from the application when all the agents in that particular Consul Datacenter are deleted from the application.

Consul Extension for ACI: Dashboard
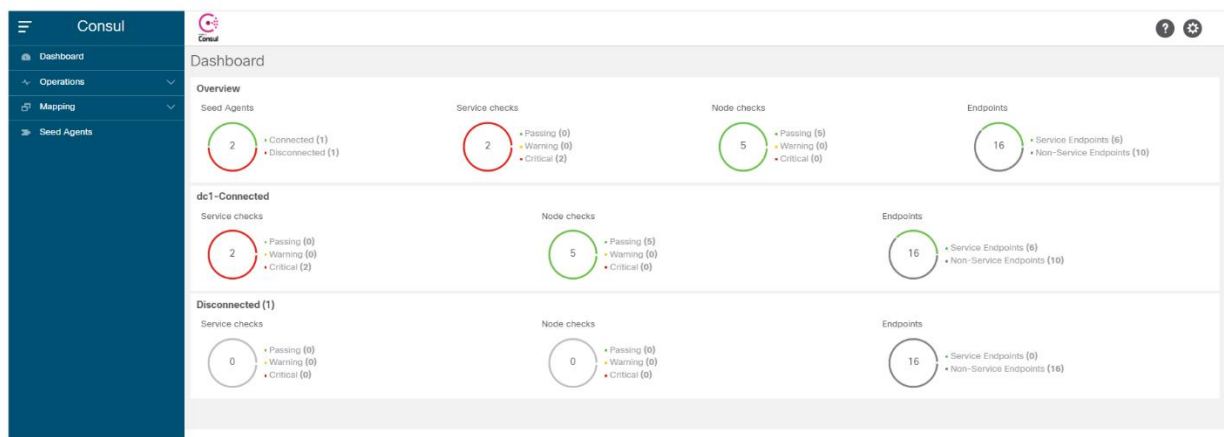
Agents -- Consul Extension for ACI dashboard displays the Consul agents added to the application on APIC. The **Up** represents the number of added agents that are connected to the Consul cluster. The

Down represents the number of added agents that are disconnected from the Consul cluster. On click of Connected/Disconnected will navigate the user to Agents screen showing the Connected/Disconnected agents based on the selection on Dashboard screen.

Services -- Services discovered by the added agents in the Consul Datacenter. The **Passing** represents the number of services which are passing the DevOps defined health checks. The **Warning** represents the number of services which are in the warning state in the health checks defined by the DevOps team. The **Critical** represents the number of services which are failing the DevOps defined health checks. On click of Passed/Warning/Critical it will navigate the user to Service checks screen displaying service with passed/warning/critical check status based on the selection on Dashboard screen.

Nodes -- Nodes discovered by the added agents in the Consul Datacenter. The **Passing** represents the number of Consul nodes which are passing the DevOps defined health checks. The **Warning** represents the number of Consul nodes which are in the warning state in the health checks defined by the DevOps team. The **Critical** represents the number of Consul nodes which are failing the DevOps defined health checks. On click of Passed/Warning/Critical it will navigate the user to Node checks screen displaying Nodes with passed/warning/critical check status based on the selection on Dashboard screen.

Node Checks

IP	Pod	Consul Node	Node Checks	Data Center
192.168.166.151	container_name0	nodename3	⚠ 1	dc1
192.168.108.219	container_name0	nodename4	⊗ 1	dc1

10 ⌄ Rows   Page [1] of 1 |◀ ◀ 1-2 of 2 ▶ ▶|

Endpoints -- ACI endpoints detected in the Tenant. The **Service Endpoints** represents the number of ACI endpoints within the Tenant which have one or more services detected by Consul. The **Non-Service Endpoints** represents the number of endpoints within the Tenant that do not have any services running on them as detected by Consul. On click of Service Endpoints/ Non-service Endpoints it will navigate the user to Service Endpoint screen and Non-service endpoint screens.

Service Endpoints:

Service Endpoints

Endpoint	Pod	IP	Application Profile	EPG	EPG Health	Consul Node	Node Checks	Service	Service Instance	Port	Service Kind	Service Tags	Service Chec
name6-test6	container_nar	192.168.166.	ap0-testap	epg1-testepg	53	nodename3	⚠ 1	servicename2	serviceid2	1030	servicekind2	cache global	⚠ 1
name9-test9	container_nar	192.168.108.	ap0-testap	epg1-testepg	53	nodename4	⊗ 1	servicename9	serviceid9	1031	servicekind9	cache global	⊗ 2

Non-Service Endpoints:

IP	MAC	Pod	Application Profile	EPG
172.22.16.4	72:a6:4b:5c:e6:3e		ap1-testap	epg0-testepg
10.134.225.194	8e:6e:d3:15:57:0a	container_name0	ap0-testap	epg1-testepg
172.25.46.10	26:09:ff:4a:f8:a7		ap0-testap	epg1-testepg
172.17.166.84	53:e6:4c:17:3a:6d	container_name0	ap0-testap	epg1-testepg
10.142.215.193	99:b4:9d:f8:dc:2c	container_name2	ap1-testap	epg1-testepg
10.5.7.83	e8:19:fe:2e:cc:77	container_name4	ap1-testap	epg1-testepg
172.27.159.109	9b:70:5c:a6:af:01		ap1-testap	epg0-testepg

Consul Extension for ACI: Operations

Operations allow users to explore and visualize L4-L7 service mapping ACI policy and fabric constructs. Along with service mapping, Operations allow users to also get operational including service health mapped to associated ACI constructs. This enables users to triage and resolve network related issues impacting the application services, in turn reducing Mean-Time-To-Resolution (MTTR).

Operations supports two visualization options. **List** and **Topology.**

List -- Mapping of L4-L7 Consul service details to ACI logical and fabric constructs in a tabular format with filters.

Select desired Consul Datacenter in the left pane under Operations.

Click the **List** icon on the top right in the work pane.

Endpoint -- **Name** of the **ACI endpoint** supporting a **service instance**.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Pod -- **Name** of the **ACI Pod** to which the endpoint belongs to

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

IP -- **IP address** of the ACI endpoint supporting a **service instance**

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Application Profile -- ACI Application Profile Name of the endpoint running the service instance is a member.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

EPG -- ACI Endpoint Group name of the endpoint running the service instance is a member.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

EPG Health -- Health of the EPG as detected by ACI to which the endpoint running the service instance is a member.

Supported Filters: "**==**", "**!=**", "**>**", "**<**", "**>=**", "**<=**"

Consul Node -- Node name on which the service instance is running on. Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Node Checks -- Health checks of the node running the service instance. Defined by application developer/DevOps. Discovered by Consul.

Supported Filters: "**==**", "**!=**", "**>**", "**<**", "**>=**", "**<=**"

Accepted Values: "**passing**", "**failing**", "**critical**"

Service -- Name of the service to which service instance belongs to. Defined by application developer/DevOps. Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Service Instances -- Name of the service instance. Defined by application developer/DevOps. Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Port -- Port number on which service instance is open to support other application components. This would map to a filter entry in a provided contract. Defined by application developer/DevOps. Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Service Kind -- Type of Service. Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Service Tags -- Tags associated with this service. Example of tags: "Web","Dev", "Test", "Prod" **.** Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

Service Checks -- Health checks associated with the service. Defined by application developer/DevOps. Discovered by Consul.

Supported Filters: "**==**", "**!=**", "**>**", "**<**", "**>=**", "**<=**"

Accepted Values: "**passing**", "**warning**", "**critical**"

Namespace -- Namespace to which the service instance belongs. Defined by application developer/DevOps. Discovered by Consul.

Supported Filters: "**!contains**", "**contains**", "**==**", "**!=**"

List: Details:

Click on any row on the list to view service instances to ACI fabric correlation in the right work pane.

Interface -- Pod, Node and Interfaces on which the service instance is connected to the ACI fabric

MAC -- MAC address of the ACI endpoint on which the service instance is running.

VRF -- Virtual Routing and Forwarding (VRF) Context of the ACI endpoint running the service instance.

BD -- Bridge Domain (BD) of the ACI endpoint running the service instance.

Learning Source -- Method through which the endpoint running the service instance is learned by ACI.

Reporting Controller -- Physical or Virtualization controller supporting the endpoint running the service instance.

Hosting Server -- Compute server hosting the endpoint running the service instance.

Topology -- Mapping of L4-L7 Consul service details to ACI logical and fabric constructs in a relationship tree.

Select desired Consul Datacenter in the left pane under Operations.

Click the **Topology** icon on the top right in the work pane.

Each node in the tree represents one of the following:

Service (S) -- Representation of a service instance discovered by Consul. Includes **service instance name** in **grey** colored text and service health checks. The **check-mark** symbol in **green** represents passing **service health checks**. The **warning** symbol in **amber** represents the number of **service health checks** in a warning state. The **X** symbol in **red** represents the number of failing **service health checks**.

Parent relationship: **Runs on an ACI Endpoint**.

End Point (EP) -- Representation of an ACI endpoint running a service instance. Includes **endpoint name** in **black** colored text, all **service instance names** running on the endpoint in **grey** colored text and service health checks. The **check-mark** symbol in **green** represents the passing **node and service health checks** contained in the endpoint. The **warning** symbol in **amber** represents the number of **node and service health checks** in a warning state contained in the endpoint. The **X** symbol in **red** represents the number of failing **node and service health checks** contained in the endpoint.

Parent relationship: **Member of an ACI Endpoint Group (EPG)**.

Child relationship: **Runs one or more service instances discovered by Consul.**

Non-Service Endpoints (EP) -- Representation of all the Endpoints within an ACI Endpoint Group (EPG) which **do not have** Consul discovered **service instances** running on them.

Parent relationship: **Member of an ACI Endpoint Group (EPG)**.

End Point Group (EPG) -- Representation of an ACI Endpoint Group (EPG). Includes **endpoint group (EPG) name** in **black** colored text, all **service instance names** running within the EPG in **grey** colored text and service health checks. The **check-mark** symbol in **green** represents the passing **node and service health checks** contained in the endpoint. The **warning** symbol in **amber** represents the number of **node and service health checks** in a warning state contained in the endpoint. The **X** symbol in **red** represents the number of failing **node and service health checks** contained in the endpoint.

Parent relationship: **Member of an Application Profile** .

Child relationship: **Contains one or more Endpoints.**

Application Profile (AP) -- Representation of an Application Profile in ACI. Includes **Application Profile (AP) name** in **black** colored text, all **service instance names** running within the application profile in **grey** colored text and service health checks. The **check-mark** symbol in **green** represents the passing **node and service health checks** contained in the endpoint. The **warning** symbol in **amber** represents

the number of **node and service health checks** in a warning state contained in the endpoint. The **X** symbol in **red** represents the number of failing **node and service health checks** contained in the endpoint.

Child relationship: **Contains one or more Endpoint Groups.**

Click any node on the topology to view details

Details will be displayed in the **right work pane**.

Topology: Service:
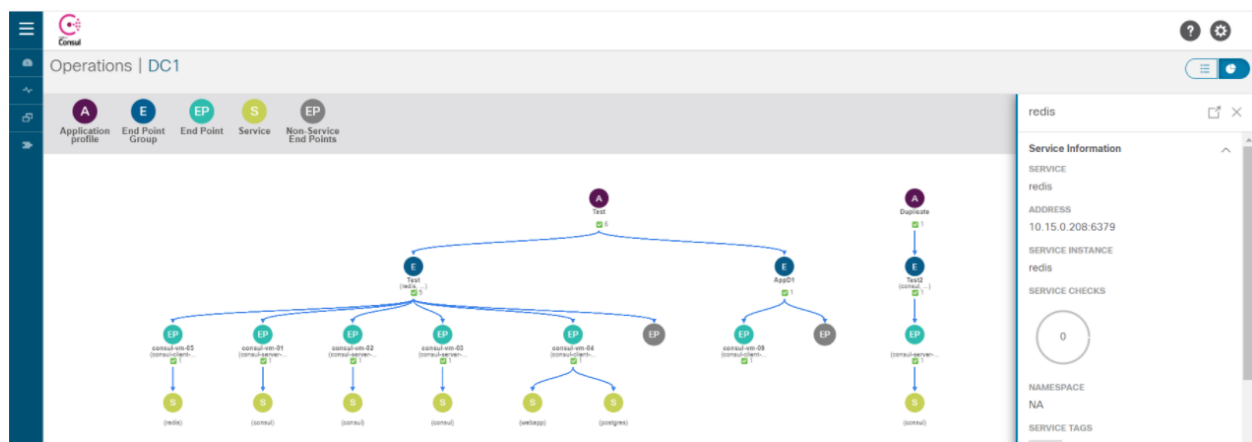
Service -- Name of the service instance to which the service instance is a member. Discovered by Consul

Service Instance -- Service instance name. Discovered by Consul.

Service Checks -- Health checks associated with the service. Defined by application developer/DevOps. Discovered by Consul.

Namespace -- Consul namespace to which the service instance belongs. Discovered by Consul

Service Tags -- Tags associated with a service instance. E.g. "Web", "Prod", Dev", etc. Discovered by Consul.

Service Kind -- Type of Service. E.g. "envoy-proxy". Discovered by Consul.

Topology: Service: Details:

Click the **pop-out** icon in the right work pane to view details.

Details will be displayed in the new **work pane**.

Details include:

Service Checks

       Name

       Service Name

       CheckID

       Type

       Notes

Topology: Endpoint:

Endpoint -- **Name** of the **ACI endpoint** supporting a **service instance**.

IP -- **IP address** of the ACI endpoint supporting a **service instance**

Interface -- Pod, Node and Interfaces on which the service instance is connected to the ACI fabric

MAC -- MAC address of the ACI endpoint on which the service instance is running.

VRF -- Virtual Routing and Forwarding (VRF) Context of the ACI endpoint running the service instance.

BD -- Bridge Domain (BD) of the ACI endpoint running the service instance.

Learning Source -- Method through which the endpoint running the service instance is learned by ACI.

Reporting Controller -- Physical or Virtualization controller supporting the endpoint running the service instance.

Hosting Server -- Compute server hosting the endpoint running the service instance.

Consul Node -- Node name on which the service instance is running on. Discovered by Consul.

Node Checks -- Health checks of the node running the service instance. Defined by application developer/DevOps. Discovered by Consul.

Service -- Name of the service to which service instance belongs to. Defined by application developer/DevOps. Discovered by Consul.
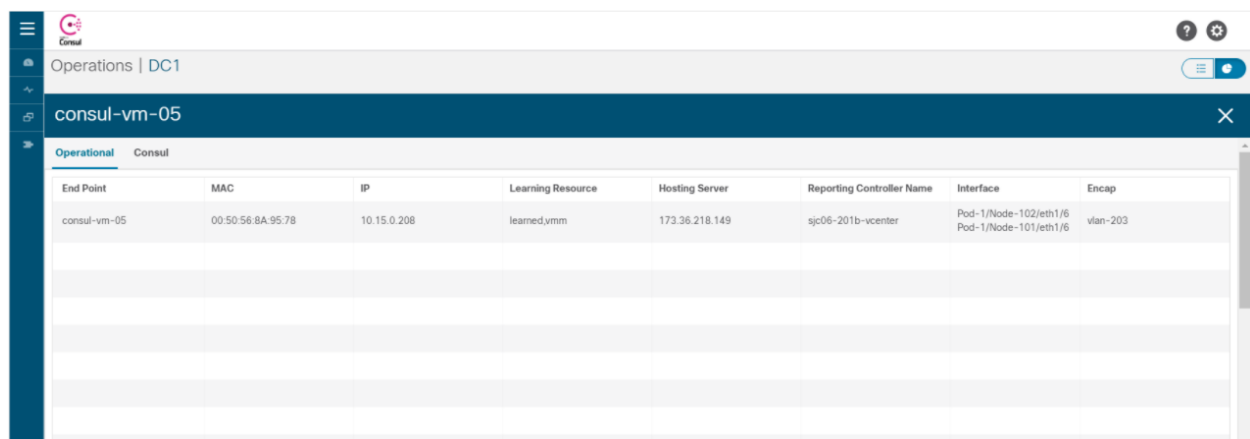
Service Checks -- Health checks associated with the service. Defined by application developer/DevOps. Discovered by Consul.

Service IP -- **IP** address and **Port** number of the service. Discovered by Consul.

Topology: Endpoint: Details:

Click the **pop-out** icon in the right work pane to view details.

Details will be displayed in the new **work pane**.

Details include:

Operational

        Endpoint

        IP

        MAC

        Learning Resource

Hosting Server

Reporting Controller Name

Interface

Multicast Address

Encap

Consul

Node Checks

Service Checks

Services

Topology: Endpoint Group:

EPG Name -- ACI Endpoint Group name of the endpoint running the service instance is a member.

VRF -- Virtual Routing and Forwarding (VRF) Context of the ACI endpoint running the service instance.

BD -- Bridge Domain (BD) of the ACI endpoint running the service instance.

Consumer Contract -- ACI consumed contract applied to the EPG.

Provider Contract -- ACI provided contract applied to the EPG.

Consul Node -- Node name on which the service instance is running on. Discovered by Consul.

Node Checks -- Health checks of the node running the service instance. Defined by application developer/DevOps. Discovered by Consul.

Service -- Name of the service to which service instance belongs to. Defined by application developer/DevOps. Discovered by Consul.
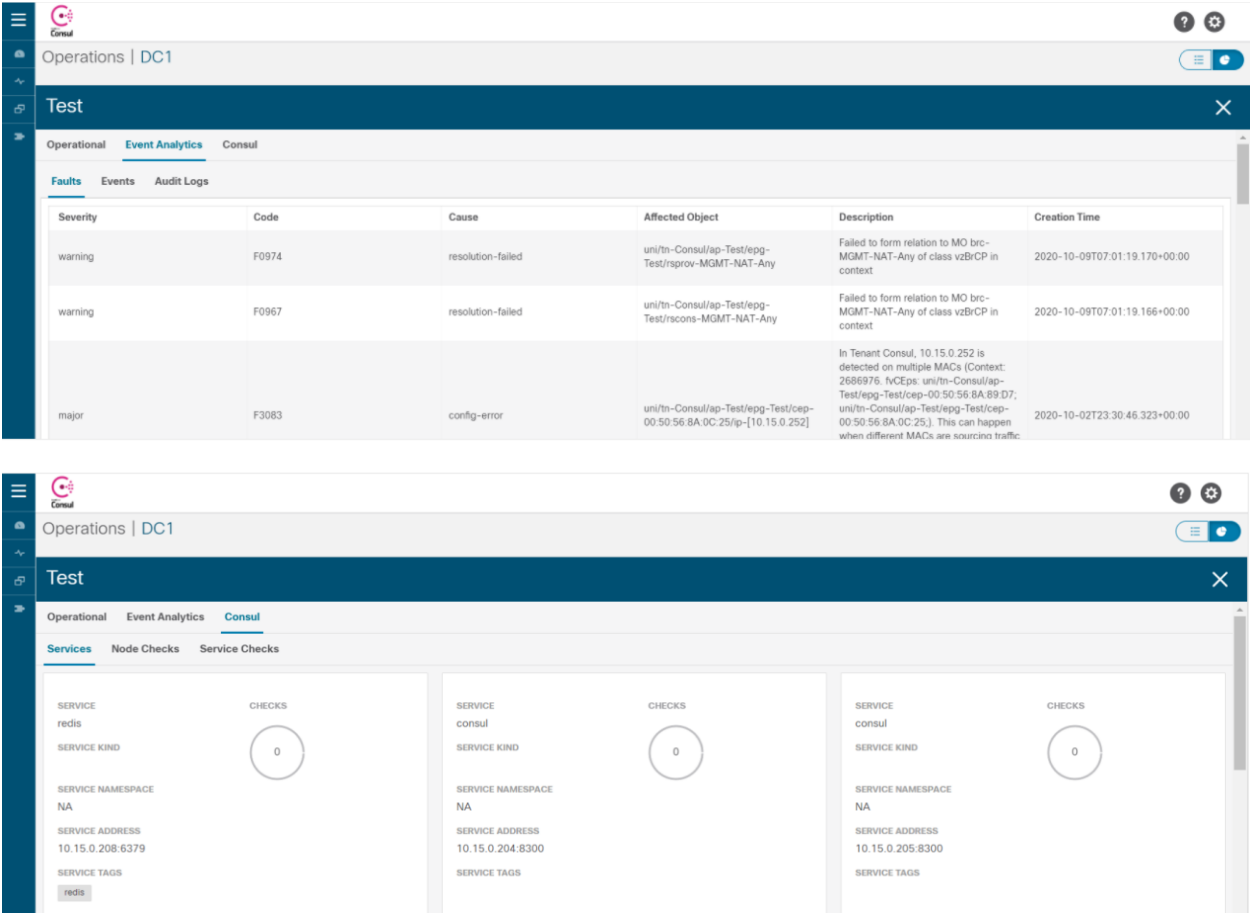
Service Checks -- Health checks associated with the service. Defined by application developer/DevOps. Discovered by Consul.

Service IP -- **IP** address and **Port** number of the service. Discovered by Consul.

Topology: Endpoint Group: Details:

Click the **pop-out** icon in the right work pane to view details.

Details will be displayed in the new **work pane**.

Details include:

Event Analytics

Faults

Events

Audit Logs

Operational

Endpoint

IP

MAC

Learning Resource

Hosting Server

Reporting Controller Name

Interface

Multicast Address

Encap

Consul

Node Checks

Service Checks

Services

Consul Extension for ACI: Mapping

There is a possibility of having the same IP address across multiple VRFs in a Tenant. Based on the VRF input selected while adding the seed agent, Consul Extension for ACI app shall iterate through all the endpoints in a Tenant and come up with the mapped list.
If no VRF is selected while adding seed agent, all the endpoints shall be iterated by app in a tenant and mappings shall include endpoints with overlapping IP addresses too (across multiple VRFs).

Note: Application will only support endpoints in an ACI Bridge Domain where **Unicast routing is enabled**. This is the only method through which APIC learns the IP address of the endpoint.

Mapping | dc1						✕
Selected 7 out of 7						
IP	Mapped	VRF	BD	EPG	Application Profile	Tenant
10.15.0.204	⊘	Consul/Consul_VRF	10.15.0.0_24_BD	Test	Test	Consul
10.15.0.204	⊘	Consul/Consul_VRF_2	10.15.0.0_24_BD_2	Test2	Duplicate	Consul
10.15.0.205	⊘	Consul/Consul_VRF	10.15.0.0_24_BD	Test	Test	Consul
10.15.0.206	⊘	Consul/Consul_VRF	10.15.0.0_24_BD	Test	Test	Consul
10.15.0.207	⊘	Consul/Consul_VRF	10.15.0.0_24_BD	Test	Test	Consul
10.15.0.208	⊘	Consul/Consul_VRF	10.15.0.0_24_BD	Test	Test	Consul
10.15.0.213	⊘	Consul/Consul_VRF	10.15.0.0_24_BD	AppD1	Test	Consul
10 ∨ Rows					Page 1 of 1 ⏮ ◀ 1-7 of 7 ▶ ⏭	

There are separate entries in the Mapping dropdown in the left pane for each Consul Datacenter.

IP -- IP address of the endpoint discovered by recommendation engine.

Recommended -- Endpoints recommendation to include in the service correlation. Recommended endpoints are considered by default.

VRF -- Name of the VRF for the endpoint.

BD -- Name of the Bridge Domain (BD) of which the endpoint is a part of.

EPG -- Endpoint Group (EPG) name to which the endpoint has a membership to.

Application Profile -- Application Profile (AP) name that the endpoint belongs to
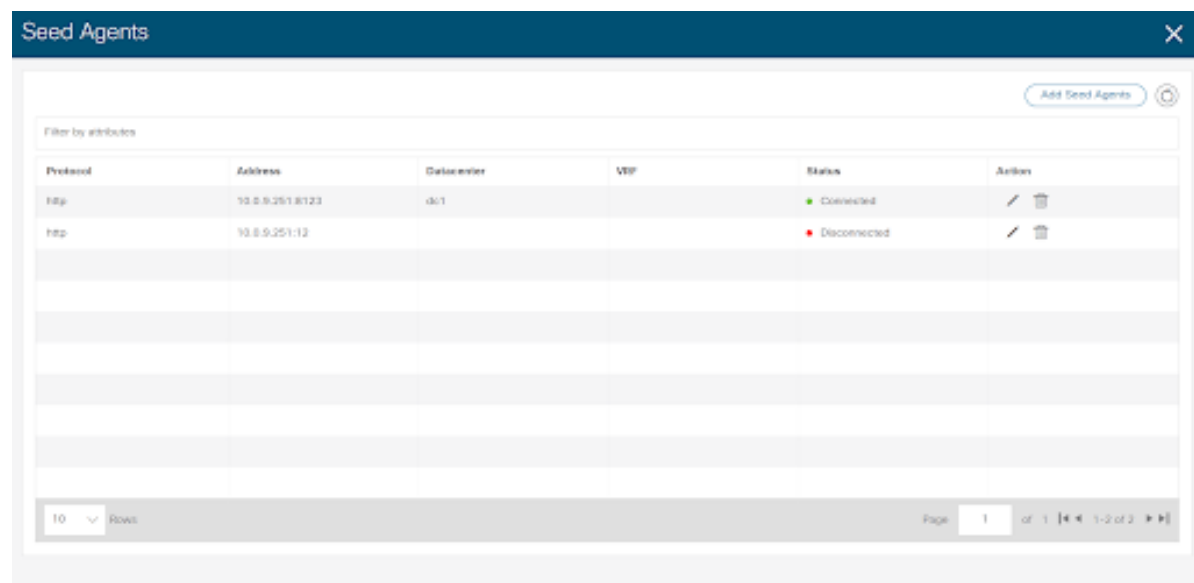
**Tenant** -- Name of the Tenant of which the endpoint is a part of.

Map -- Consider the endpoint for the automated service correlation. The **Toggle Right** adds the endpoint to the automated service correlation. The **Toggle Left** removes the endpoint from automated service correlation.

Note: User needs to save the mapping by clicking **Save** on the far-right corner of the work pane.

Consul Extension for ACI: Agent

Please refer to **Onboarding a Consul Agent** to get started with Consul Extension for ACI.

Protocol -- Protocol over which API calls are made to the Consul agent. Supports **http** and **https**.

Address -- Address/URL and port number for the added Consul agent.

Token -- Consul ACL token with privileges to read Consul data.

Datacenter -- Consul private networking environment to which the agent belongs.

Status -- The **Connected** represents the application is able to gather data from the Consul agent. The **Disconnected** represents the inability of the application to gather data from the Consul agent

Actions -- Actions include **Update** the agent configuration on this application and **Delete** the agent from this application.

Consul Extension for ACI: Service Intentions (Network Middleware Automation)

This feature is under construction and will have following functionality when complete:

- Service mesh defined service-to-service communication topology overlay with ACI logical and fabric constructs.

- Verification of ACI policy (contract and filter) to support Consul service mesh intentions.

- ACI policy (contract and filter) recommendation and creation based on Consul service mesh intentions.

Additional Resources

Download the application
<dcappcenter>

FAQs
<faq github link>

Learn more about Consul
https://learn.hashicorp.com/consul

Requests Feature and File Bugs
https://github.com/ciscoecosystem/consul-aci/issues

Glossary

Consul Terminology	What it mean and how it relates to ACI
Agent	The Consul agent is the core process of Consul. The agent maintains membership information, registers services, runs checks, responds to queries, and more. The agent must run on every node that is part of a Consul cluster. Any agent may run in one of two modes: client or server. A server node takes on the additional responsibility of being part of the consensus quorum. No approximation to APIC terminology
Connect	Consul Connect provides service-to-service connection authorization and encryption using mutual Transport Layer Security (TLS).

	Applications can use sidecar proxies in a service mesh configuration to establish TLS connections for inbound and outbound connections without being aware of Connect at all. Applications may also natively integrate with Connect for optimal performance and security. Connect can help you secure your services and provide data about service-to-service communications. L4-L7 approximation of ACI policy
Datacenter	Consul datacenter is a networking environment that is private, low latency, and high bandwidth. This excludes communication that would traverse the public internet, but for our purposes multiple availability zones Approximation of a Pod or a site in ACI
Namespace (Enterprise feature)	Namespaces allow multiple teams within the same organization to share the same Consul datacenter(s) by separating services, Consul KV data, and other Consul data per team. This provides operators with the ability to more easily provide Consul as a service. Namespaces also enable operators to delegate ACL management. Approximation of a Tenant in ACI
Network Segments (Enterprise feature)	Consul Network Segments enables operators to create separate LAN gossip segments in one Consul cluster. Agents in a segment are only able to join and communicate with other agents in it's network segment. This functionality is useful for clusters that have multiple tenants that should not be able to communicate with each other. Approximation of a VRF in ACI.
Node	Node is a representation of a virtual or physical machine which runs your workload. Approximation of an Endpoint in ACI
Service	Service is a well defined component of functional behaviour that provides a logical grouping of application functions. A service is defined in a configuration file or added at runtime over the HTTP interface. Approximation of an End Point Group (EPG) in ACI
Access Control (ACLs)	Consul uses Access Control Lists (ACLs) to secure the UI, API, CLI, service communications, and agent communications. At the core, ACLs

	operate by grouping rules into policies, then associating one or more policies with a token.
	Consul ACL can be used to control access to data and APIs.
	Approximation of Role-based Access Control (RBAC) in ACI
Intentions	Intentions define access control for services via Connect and are used to control which services may establish connections.
	Intentions are enforced by the proxy or natively integrated application on inbound connections. After verifying the TLS client certificate, the authorize API endpoint is called which verifies the connection is allowed by testing the intentions. If authorize returns false the connection must be terminated.
	L4-L7 approximation of ACI Filters, Filter Entries and Contracts