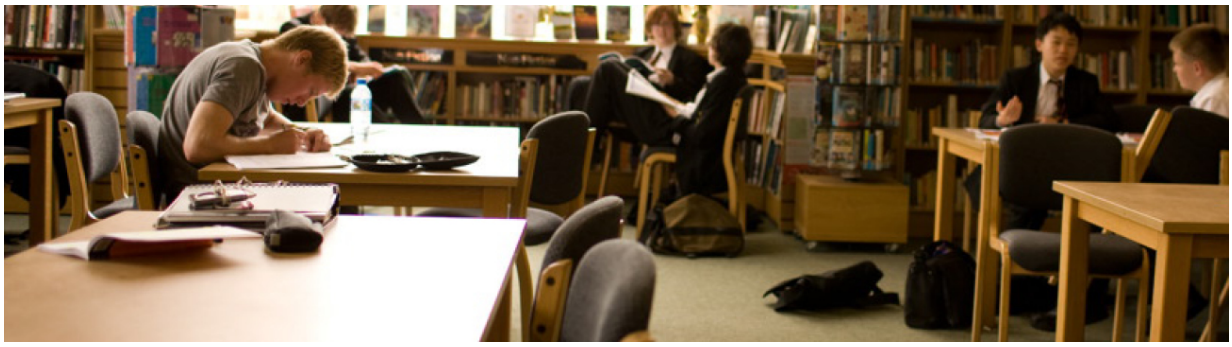
Cisco Intercloud Fabric Provider Platform Architecture



2015-11-13

Contents

Intended Audience and Disclaimer

This technical architectural document is primarily intended for service provider decision-makers, architects, engineers, and application owners involved in integration with Cisco Intercloud Fabric Provider Platform (ICFPP) for offering hybrid cloud as a service for their customers.

Many external API interfaces and integration frameworks referred to in the document are intended to help explain the overall architecture.

Cisco Intercloud Fabric Overview

The Cisco Intercloud Fabric (ICF) solution provides a faster and flexible response to business needs and addresses the potential challenges that are incumbent with hybrid cloud. Cisco ICF is an **open** solution that supports multi-hypervisor and multi-cloud with the freedom to place workloads across heterogeneous environments in both private and provider clouds. To protect critical business assets and meet compliance, Cisco ICF delivers **secure** and scalable connectivity that extends private clouds to provider clouds and provides workload **security** throughout hybrid cloud. In addition, Cisco ICF enforces consistent network and workload policy throughout the hybrid cloud. To provide consistent operations and workload **portability** across clouds, Cisco ICF delivers unified hybrid cloud management for end users and IT administrators. It also enables workload mobility to and from provider clouds for both physical and virtual workloads.

Cisco Intercloud Fabric Solution Overview

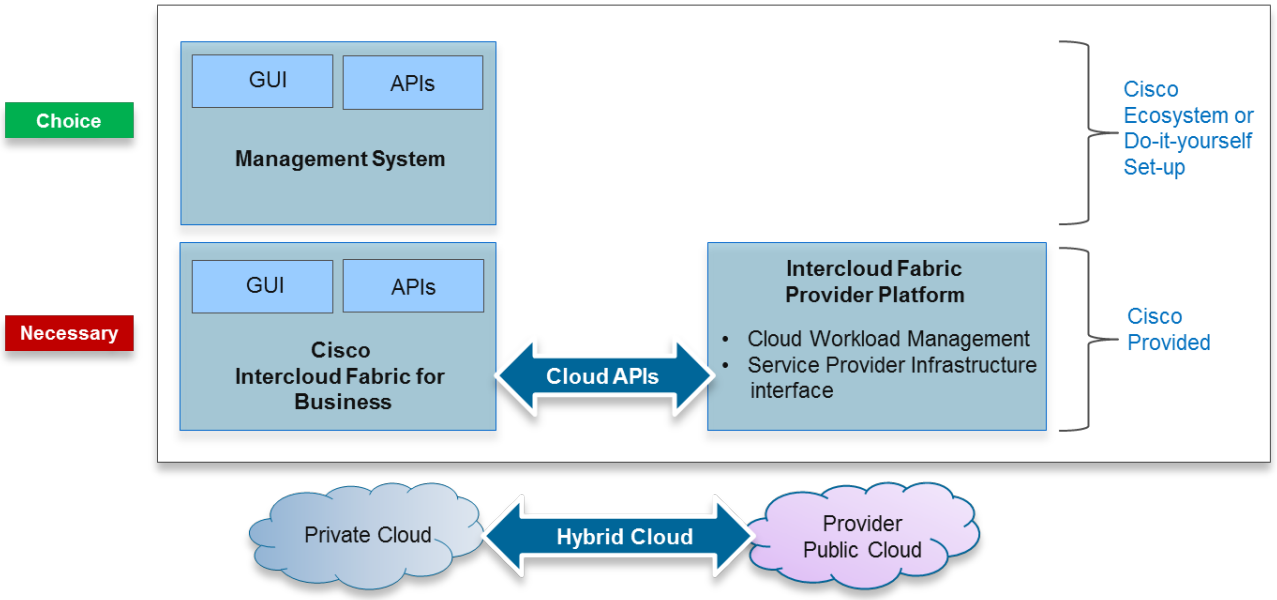


Figure 1 - Cisco Intercloud Fabric Solution Overview

Cisco Intercloud Fabric for Provider

Cisco Intercloud Fabric for Provider (ICFP) is intended for provider-managed cloud environments, allowing their enterprise customers to seamlessly extend their private cloud environments into the provider's public cloud, while keeping the same level of security and policy across cloud environments. Cisco ICFP comprises the following components:

- Cisco Intercloud Fabric Director (ICFD)
- Cisco Intercloud Secure Extender (ICSE)
- Cisco ICFPP

Cisco Intercloud Fabric Architectural Components

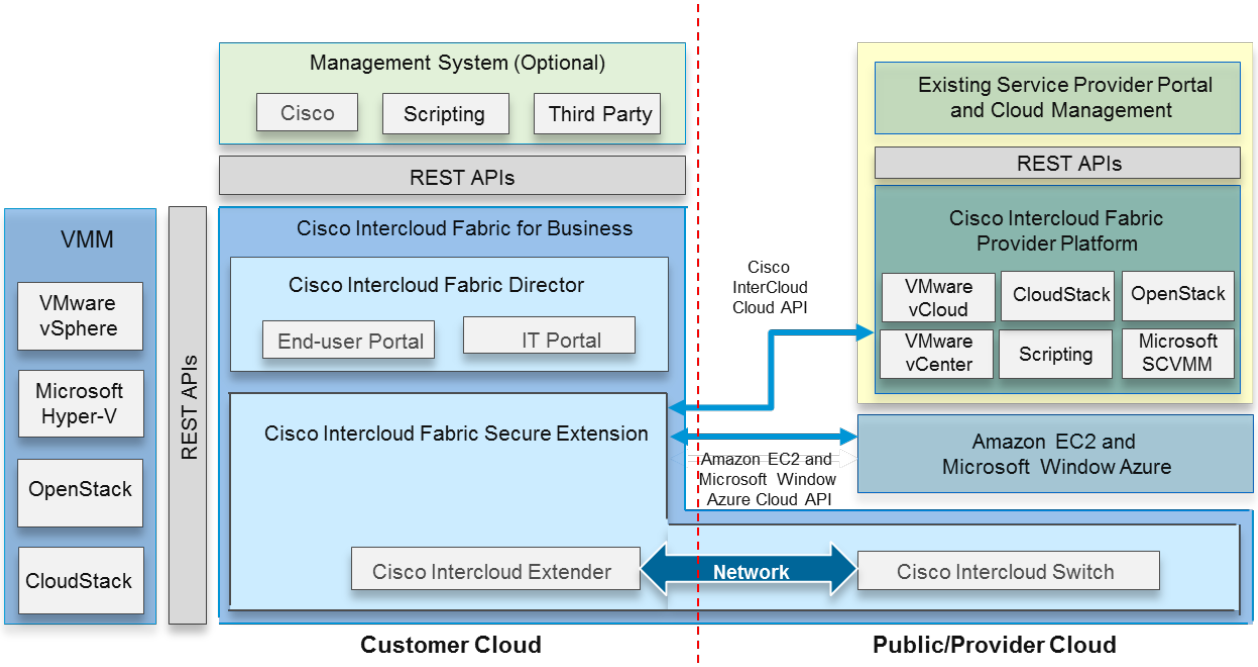


Figure 2 - Cisco Intercloud Fabric Architectural Components

Cisco Intercloud Fabric Director

Workload management in a hybrid environment goes beyond the ability to create and manage a virtual machine (VM) in the private or public/provider cloud, and network extension. Both capabilities are part of the overall hybrid cloud solution that must also provide different types of services, such as policy capabilities (placement, quotas, and so on), the ability to manage workloads in a heterogeneous environment, and other capabilities discussed below.

Cisco ICFD provides end users and IT administrators with a seamless experience and access to the private and provider clouds, enabling workloads to be placed where it benefits the most and according to a technical (capacity, security, etc.) and business (compliance, etc.) needs. Cisco ICFD is the single point of management and consumption of hybrid cloud offerings for end users and IT administrators.

Heterogeneous cloud platforms will be supported by Cisco ICFD in the private cloud, which operationally unifies workload management in a cloud composed of different cloud infrastructure platforms, such as VMware (vSphere, vCloud), Microsoft (Hyper-V, SCVMM), OpenStack, and CloudStack. This provides a holistic workload management experience and multiple options in terms of cloud infrastructure platforms for our customers.

Cisco ICFD exposes northbound APIs that allow customers to programmatically manage their workloads in a hybrid cloud environment or to integrate with their management system of choice, thereby allowing a granular application management capability that includes policy and governance, application blueprint, and other features.

Future releases of Cisco ICFD will include enhanced services that will differentiate the Cisco ICF solution, such as a bare-metal workload deployment in a hybrid cloud environment and an enhanced IT administrator portal for options to configure data redundancy (DR), backup, Virtual Desktop Interface (VDI), and other services.

Cisco Intercloud Secure Extender

The Cisco Intercloud Secure Extender (ICSE) forms the basis for the core switching and services infrastructure in the Intercloud solution. The functionality provided by Cisco ICF includes:

- A secure Layer 2 network extension from a private data center network to a provider cloud.
- Advanced switching features, such as access control lists (ACLs) and Internet Group Management Protocol (IGMP) for applications running in the public cloud.
- Intercloud services including zone-based firewalls, VPN, and routing capabilities in the cloud.

The Cisco ICSE is composed of several components working together to provide this functionality. The enterprise data center is connected to the provider data center through a secure tunnel established between a pair of virtual appliances - the Intercloud Extender (ICX) running in the enterprise, and the Intercloud Switch (ICS) running in the provider cloud. These virtual appliances can be deployed in a high-availability pair to provide redundancy. Virtual services are then deployed within this environment to provide firewall and routing services within the cloud.

Cisco Intercloud Fabric Provider Platform

Cisco ICFPP simplifies and abstracts the complexity of involved in working with a variety of public cloud APIs, and enables cloud API support for service providers who currently do not have API support. Cisco ICFPP provides an extensible adapter framework that allows integration with different provider cloud infrastructure management platforms, such as VMware vCloud, OpenStack, Microsoft System Center, scripts, and other cloud APIs.

Currently, service providers have their own proprietary cloud APIs—such as Amazon EC2 and Windows Azure—that give customers limited choices and provide no easy method for moving from one provider to another. Cisco ICFPP abstracts this complexity and translates Cisco Intercloud Fabric cloud API calls to cloud platform APIs of different provider infrastructure platforms, giving customers the option of moving their workloads regardless of the cloud API exposed by the service provider.

Many service providers do not provide cloud APIs that Cisco Intercloud Fabric can use to deploy customers' workloads. One option for these providers is to provide direct access to their virtual machine manager's SDK or API, such as vCenter or System Center, which exposes the provider environment and is not a preferred option for service providers due to security concerns. Cisco ICFPP, as the first point of authentication for the customer cloud when requesting cloud resources, enforces highly secure access to the provider environment. In addition, Cisco ICFPP provides the cloud APIs that are required for service providers to be part of the provider ecosystem for Cisco Intercloud Fabric.

As the interface between the Cisco Intercloud Fabric from customer cloud environments and provider clouds (public and virtual private clouds), Cisco ICFPP provides the following benefits:

- Standardizes and brings uniformity to cloud APIs, making it easier for Cisco Intercloud Fabric to consume cloud services from service providers that are part of the Cisco Intercloud Fabric ecosystem.
- Helps secure access to a service provider's underlying cloud platform.
- Limits the utilization rate per customer or tenant environment.
- Provides northbound APIs for service providers for integration with existing management platforms.
- Supports multitenancy.
- Monitors resource usage per tenant.
- Meters resource usage per tenant.

In the future, Cisco ICFPP will help build Cisco infrastructure-specific differentiation. For example, Cisco ICFPP could allow the support for enterprises to deploy bare-metal workloads in the provider clouds.

Cisco Intercloud Management System

The seamless environment created by Cisco ICF between private and public resources can be a boon for functionality, but needs to go a step further to avoid requiring placement decisions by consumers. The Cisco ICF solution is augmented with a management system that makes placement decisions that comply with business needs in areas such as:

- **Access Control—**A set of policies that enforce role-based access control to the different VMs.
- **Compliance—**Support for defining different policies aligned with existing compliances, such as SOX, PCI, HIPAA, or SAS 70.
- **Capacity Utilization—**The ability to define policies that monitor capacity utilization and take subsequent action, such as notification or restricting the environment usage. Eventually this policy will trigger an environment resizing.
- **Network—**The ability to enforce ACL or firewall rules based on the workload requirement, on the appliance/hardware (Intercloud Fabric Firewall) or operating system (Windows Firewall or iptables).
- **Performance—**Policy definition for performance characteristics of the workload (such as memory, CPU, or disk utilization) and the ability to take actions, such as resizing a VM based on the utilization.
- **Personalization—**VM operating system personalization that adheres to corporate standards for naming conventions, installed software, and so on.

- **Placement—**Restricting VM placement in accordance with the business requirement; for example, a policy for VMs that have a sensitive workload and therefore cannot run in the public cloud.
- **Provisioning—**The **a**bility to establish the number of VMs per user or project.

With a Cisco ICF management system in place, these kinds of decisions, implemented from policies set by the enterprise, allow for functionality within multiple clouds as a contiguous environment, while implementing consistent, business-relevant placement decisions.

The Cisco ICF management system connects to Cisco ICFD via the available northbound API integrating upstream portal and orchestration systems to the resources that Cisco ICF provides.

Cisco Intercloud Fabric Provider Platform Architecture

When to Deploy Cisco ICFPP?

Cisco ICFPP should be deployed by all service providers that interface with Cisco ICFD platforms. The only exceptions to this are Amazon EC2 and Windows Azure, which are available to Cisco ICF through their native public Cloud APIs.

Cisco ICFPP Deployment Network Topology

To access a service provider's cloud resources, Cisco ICFD must access the Cisco ICFPP virtual appliance from the public network. Therefore, the public network interface of the virtual appliance must be deployed on a provider network that is exposed to the service provider edge router. The private network interface of the virtual appliance can connect to the private provider network that accesses the service provider cloud platform (such as vCloud Director).

The Cisco ICFPP deployment topology can vary for different service providers and cloud platforms. The Cisco ICFPP virtual appliance uses HTTPS connections to communicate with Cisco ICSE and the service provider cloud platform. As a result, no additional firewall rules need to be deployed in the network path between Cisco ICSE and the Cisco ICFPP virtual appliance, or the Cisco ICFPP virtual appliance and cloud platform end-points.

Single-Node Deployment Topology

The following diagram illustrates a single-node deployment with a VMware vCloud Director platform in a service provider environment:

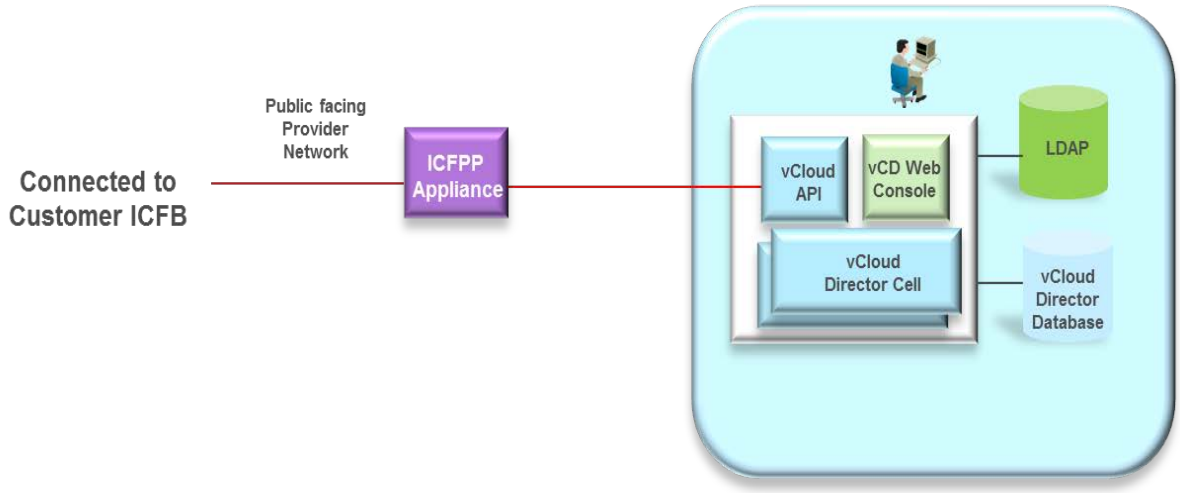


Figure 3 - Cisco ICFPP Deployment Network Topology – Single Node

The VMware high availability (HA) feature ensures that the VMs from a failed ESX/ESXi host can be restarted on another ESX/ESXi host. Cisco ICFPP HA requirements are met by this VMware HA feature. To support HA, the following configuration constraints are assumed:

- The Cisco ICFPP database is stored as an external shared storage so that it can be shared with multiple ESX/ESXi hosts.
- Affinity rules must be set for Cisco ICFPP instances to ensure that the standby Cisco ICFPP instance will not be migrated to the same ESX/ESXi host on which the primary Cisco ICFPP instance resides.

Multi-Nodes Deployment Topology

The following diagram illustrates a multi-node node deployment with a vCloud Director platform in a service provider environment:

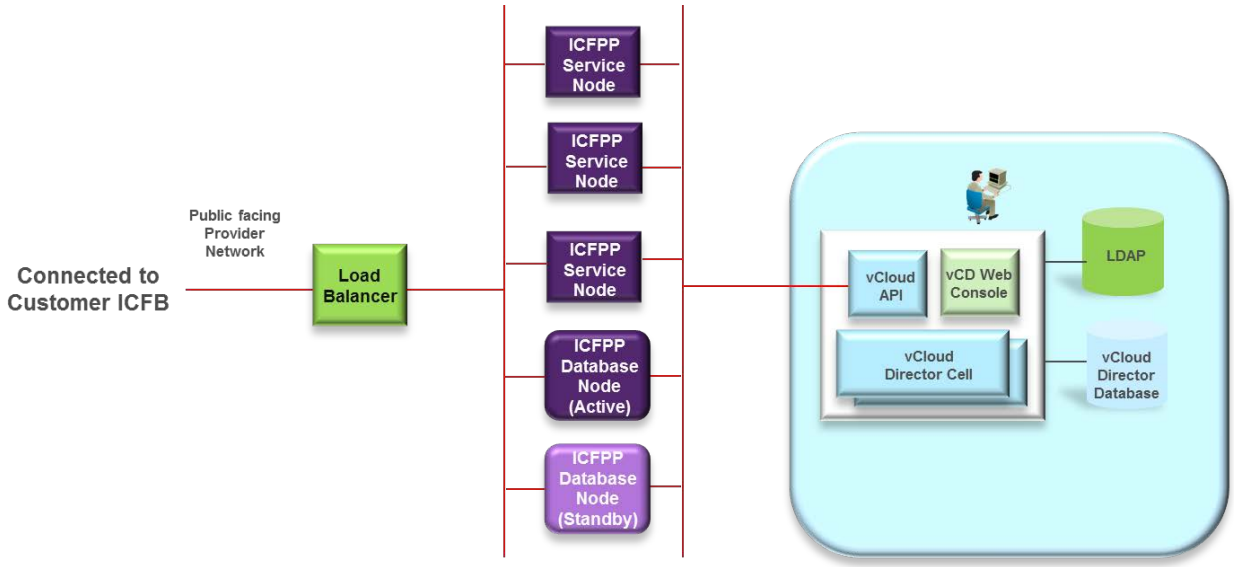


Figure 4 - Cisco ICFPP Deployment Network Topology – Multi-Node

A multi-node deployment topology consists of a load-balancer, multiple Cisco ICFPP service nodes, and two Cisco ICFPP database nodes. The load-balancer distributes customers' Cisco ICFD request workloads across multiple Cisco ICFPP service nodes, thereby scaling the Cisco ICFPP processing capacity while providing high availability.

The Cisco ICFPP database nodes serve as the database backend servers for the Cisco ICFPP service nodes. The Cisco ICFPP database nodes use a standard Active-Standby scheme to provide high availability of database services for the service nodes. A virtual IP address is shared between active and standby database nodes to simplify the Cisco ICFPP database service failover process.

Cisco ICFPP Operational Model

The Cisco ICFPP operational model contains the following operational stages:

- *SP Operations*—Operations performed on the service provider data center floor by service provider administrator. This set of operations is involved with installing Cisco ICFPP, configuring Cisco ICFPP, and provisioning tenant-related information (such as the cloud platform instance) to Cisco ICFPP.
- *Business Operations*—Operations performed in the enterprise by a customer administrator and end-users of the Cisco ICF solution. This set of operations is usually conducted after Cisco ICFPP has been deployed and activated on the service provider data center floor.

Figure 5 illustrates the Cisco ICFPP operational domains.

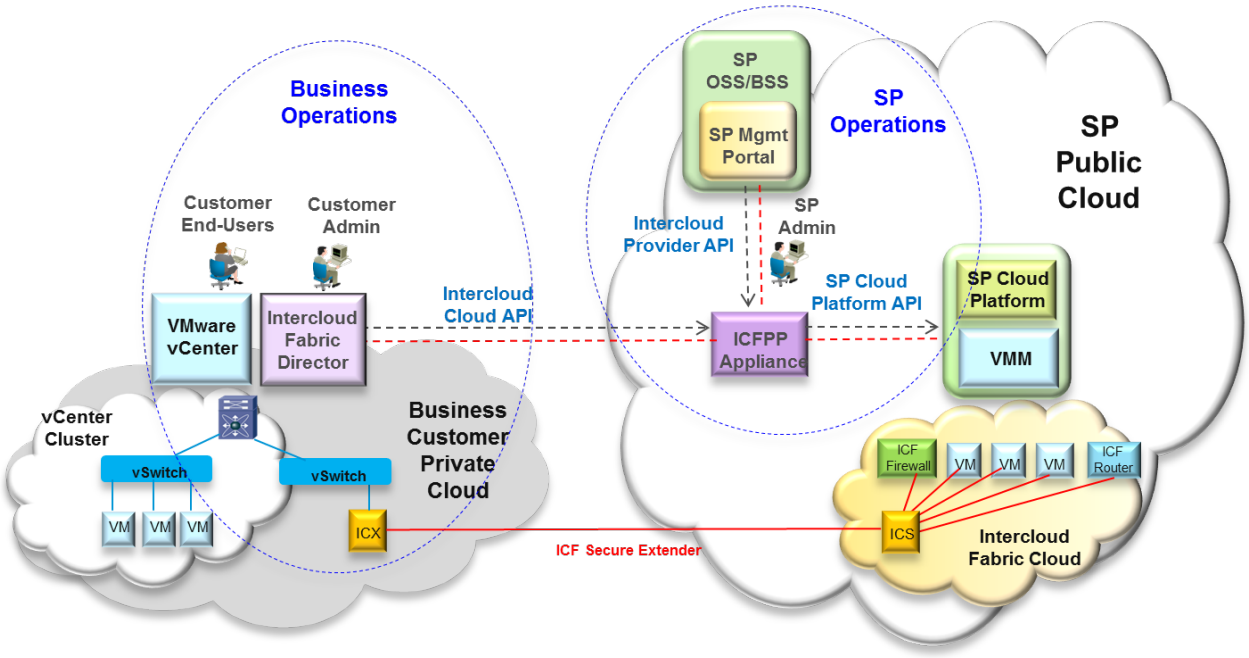


Figure 5 - ICFPP Operational Models

The operations involved in these two domains are summarized in the following sections.

Service Provider Operations – Deployment and Initialization

A Cisco ICFPP virtual appliance is deployed in the service provider data center as part of the service provider cloud platform. The service provider administrator deploys the virtual appliance with the following configurations:

- Appliance IP addresses
- SSL server and client configurations
- Initial admin user credential and privileges
- LDAP configuration for service provider administrator authentication purposes.

The service provider administrator then adds one or more cloud platform instances that the Cisco ICFPP virtual appliance can interface with and provide front-end function. These cloud platforms instances can be assigned to tenants when tenants are added. The configuration relevant to each cloud platform instance usually includes the following:

- Cloud platform type (such as, vCloud, CloudStack, or Cisco Intercloud Services – V)
- Cloud platform end-point address and port number (if default port 443 is not used)
- Service provider administrator credentials for signing on with the cloud platform (this is optional if tenant credentials are used for signing on)

Service Provider Operation – Tenant On-Boarding

Cisco ICFPP is designed to support multiple tenants. To enable a tenant on Cisco ICFPP, the service provider administrator needs to provision the following tenant-specific information to the virtual appliance:

- The cloud platform instance that has been assigned to the tenant.
- The *resources domain* (a predefined set of resources) that is assigned to the tenant (such as orgVDCs in a vCloud Director environment).

- Tenant credentials, such as an API key, which are used by Cisco ICFPP to sign a tenant onto the service provider cloud platform. Tenant credentials can be generated by the service provider management portal when the tenant is registered to a cloud account.
- The tenant account username (optional), which is used to identify the tenant-specific record.

The same process is used for adding and updating existing tenants in Cisco ICFPP. After the Cisco ICFPP virtual appliance is deployed and tenants are provisioned, the service provider administrator must ensure that the public addresses (URLs or IP addresses) of Cisco ICFPP appliances are published to the enterprise customer's portal so that the tenants can reach Cisco ICFPP through the Internet.

Business Operation – Cisco ICFD Sign-On with Cisco ICFPP

By using either the URLs or public IP addresses of the Cisco ICFPP virtual appliance instances and tenant credentials, an IT administrator can sign on with Cisco ICFPP to establish an ICFD-ICFPP management session. Before customer end users can use the Cisco ICFD self-service portal, the IT administrator must set up the Cisco ICSE to extend tenant on-premise networks to the service provider cloud.

Business Operation – Setting up Intercloud Fabric Secure Extender

With an established ICFD-ICFPP management session, IT administrators can issue Intercloud Fabric Cloud APIs to set up the Cisco ICSE to extend the tenant's enterprise network and demanded service appliances such as ICF firewall and ICF routing services. The Cisco ICSE provides Cisco ICFD end-users with a hybrid infrastructure, which preserves workload network addresses and ensures that the workload security policy is persisted across private and public clouds.

Cisco ICFSE has a several virtual appliance components that run in the provider's cloud: Intercloud Switch (ICS), ICF Router, and ICF Firewall. As a part of a Cisco ICSE deployment, Cisco ICFD works with Cisco ICFPP to upload the appliance images to public cloud, instantiate appliance instances one-by-one, and eventually bring up the entire Cisco ICF infrastructure in the public cloud.

Business Operation – Cloud Provisioning and VM Life-Cycle Management

When a Cisco ICFSE instance is established, Cisco ICFD provides different portals for customer administrators and end-users. Both customer administrators and users can use their portals to provision and/or migrate workloads to the public clouds and continue to manage the workloads with VM life-cycle management API interfaces provided by the portals.

In addition to supporting cloud orchestration API requests that are issued by Cisco ICFD, Cisco ICFPP meters tenant resource usage and monitors tenant resources, thereby enabling service providers to manage and to monetize the hybrid cloud services.

Cisco ICFPP Architecture

Cisco ICFPP is a virtual appliance that is deployed on the service provider cloud data center to enable hybrid cloud services (such as application workload migration) for service provider customers. Logically, the Cisco ICFPP virtual appliance provides two virtual network interfaces:

- One interface enables a customer's Cisco ICFD to reach the Cisco ICFPP appliance instance from public networks.
- One interface is used for Cisco ICFPP appliance connections with the service provider cloud platforms.

The following figure illustrates the Cisco ICFPP appliance architecture.

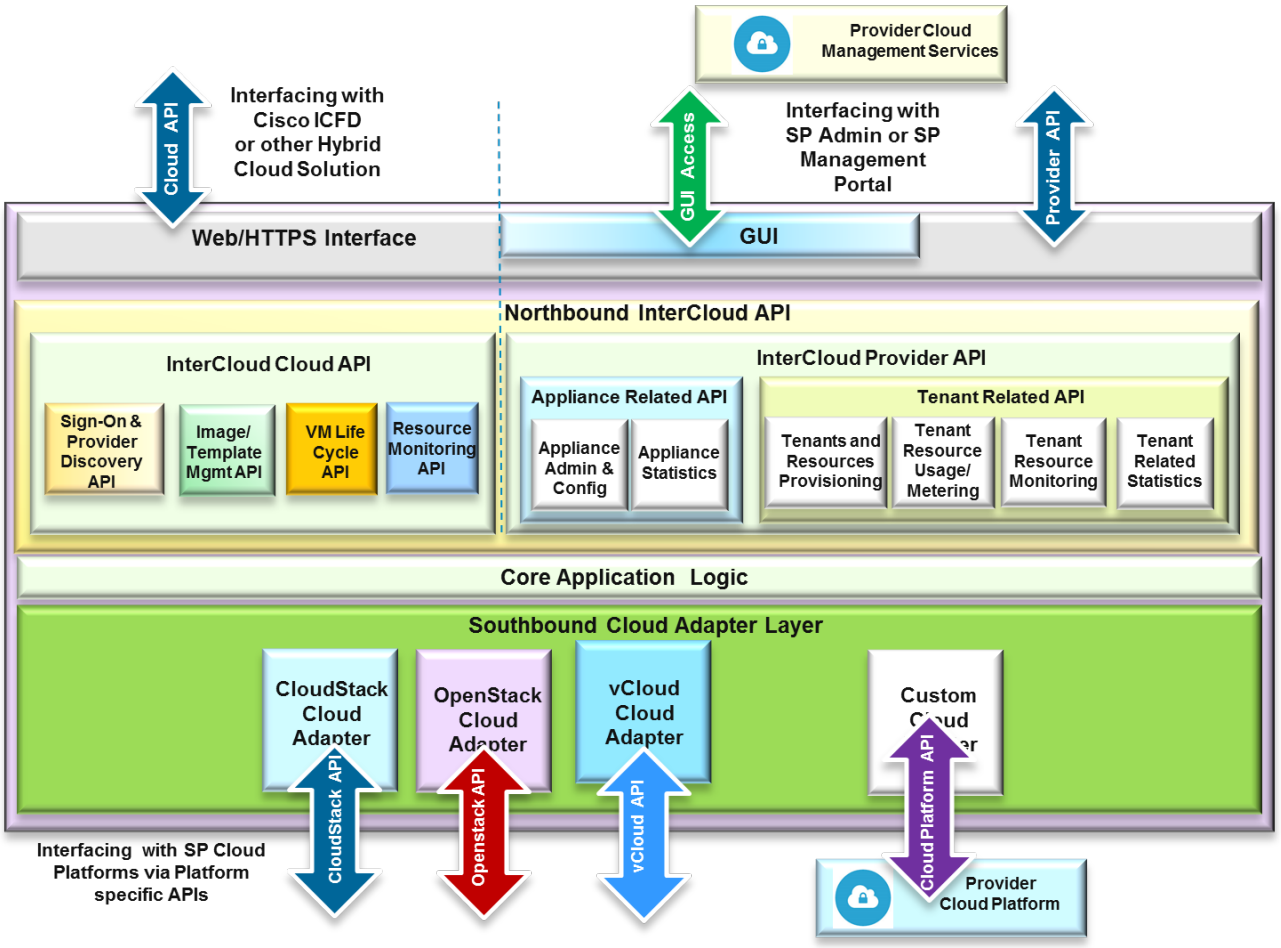


Figure 6 - Cisco ICFPP Architecture

Cisco ICFPP architecture includes the following four major interfacing modules:

1. **Northbound Intercloud Cloud API**—This module implements the Intercloud Cloud API, which is consumed by Cisco ICFD in the customer private cloud for workload-provisioning purposes.
2. **Northbound Intercloud Provider API**—This module implements two sets of APIs that enable the service provider administrator to configure the virtual appliance, provision tenants and resources assigned to the tenant, monitor tenant operations, and retrieve statistics for tenants and the virtual appliance.
3. **Core Application Logic**—This module implements the main application logic of Cisco ICFPP, such as tenant configuration in Cisco ICFPP and resource usage metering.
4. **Southbound Cloud Adapter Interface**—This module implements the various cloud platform-interfacing adapters, each of which is responsible for interfacing with a specific cloud platform, such as VMware vCloud Director, CloudStack, OpenStack, or Cisco Intercloud Services – V cloud platform.

Northbound Cisco Intercloud Cloud API

This module implements a set of RESTful Intercloud Cloud APIs, which are consumed by Cisco ICFD in the customer private cloud for workload provisioning and workload image and template management purposes.

Northbound Cisco Intercloud Provider API

This interfacing module implements a set of APIs that the service provider administrator can use to configure and manage the Cisco ICFPP virtual appliance. These APIs are categorized as follows:

- Appliance Login API
- Cloud Instance Management APIs
- Tenant Management APIs

- Remote Syslog Configuration APIs
- Logging APIs

Appliance Login API

The Appliance Login API enables an admin user to log in to the Cisco ICFPP virtual appliance.

Cloud Instance Management APIs

The Cloud Instance Management APIs are:

- *Cloud Instance Provision API* – Provisions a cloud platform instance to Cisco ICFPP.
- *Cloud Instance Update API* – Updates a specified cloud platform instance on Cisco ICFPP.
- *Cloud Instance Delete API* – Deletes a specified cloud platform instance on Cisco ICFPP.
- *Get Cloud Instance by ID* – Retrieves details of a cloud platform instance.
- *Get All Cloud Instances* – Retrieves all cloud platform instances.

Tenant Management APIs

The Tenant Management APIs are:

- *Tenant Provisioning API* – Provisions tenants and user information.
- *Tenant Update API* – Updates tenant and user information.
- *List a Tenant API* – Retrieves detailed tenant information.
- *List all Tenants API* – Retrieves all tenant records.
- *Tenant Delete API* – Deletes the specified tenant.

Remote Syslog Configuration APIs

The Remote Syslog Configuration APIs are:

- *Configure Syslog Server API* – Configures remote syslog servers.
- *Get Syslog Configuration API* – Retrieves remote syslog configuration.

Logging APIs

The Logging APIs are:

- *Download Current Logs API* – Downloads the current logs in a zipped file.
- *Download All Logs API* – Downloads all Cisco ICFPP logs in a zipped file.

Core Application Logic

The Core Application Logic module handles the following functions:

- **Intercloud Cloud API Back-End Processing**—This function primarily serves as the back end for Intercloud Cloud API processing. Depending on the type of cloud platform configured for the tenant, this function calls the appropriate cloud adapter functions for fulfilling the cloud orchestration requests issued by Cisco ICFD.
- **Cloud Instance and Tenant Provisioning**—This function creates and manages cloud platform instances and tenant records.

Southbound Cloud Adapter Interface

This module implements a number of cloud adapters for communications with cloud platforms in provisioning workloads and for the orchestration of cloud infrastructures. The Cisco ICFPP Cloud Adapter Interface enables Cisco ICFPP to support a cloud platform by implementing a set of Java-based cloud adapter plug-in classes as the back-end module for processing Intercloud Cloud API requests issued by Cisco ICFD.

As depicted in Figure 6, the Cisco ICFPP platform plans to support a number of built-in cloud adapters for facilitating the integration with the following cloud platforms in service provider environments:

- Apache CloudStack
- OpenStack

- Cisco Intercloud Services – V
- VMware vCloud Director

The service providers who use the above cloud platforms can take advantage of these built-in cloud adapters, whereas service providers who use other cloud platforms must build platform-specific adapters in order for Cisco ICFPP to work the targeted cloud platforms. To facilitate cloud adapter development, Cisco has developed a Custom Cloud Adapter Integration framework, which simplifies the cloud adapter development for service provider customers.

The Custom Cloud Adapter Integration framework implements a set of Java-based plug-in abstract classes as the back-end processing functions of Intercloud Cloud API requests received from Cisco ICFD. Cloud adapters that inherit these abstract classes must implement the actual methods of the subclasses – which usually involves issuing one or more API requests to the targeted cloud platform and expecting asynchronous corresponding API responses from the cloud platforms. Figure 7 illustrates how the cloud adapter infrastructure is logically shared between built-in and custom adapters.

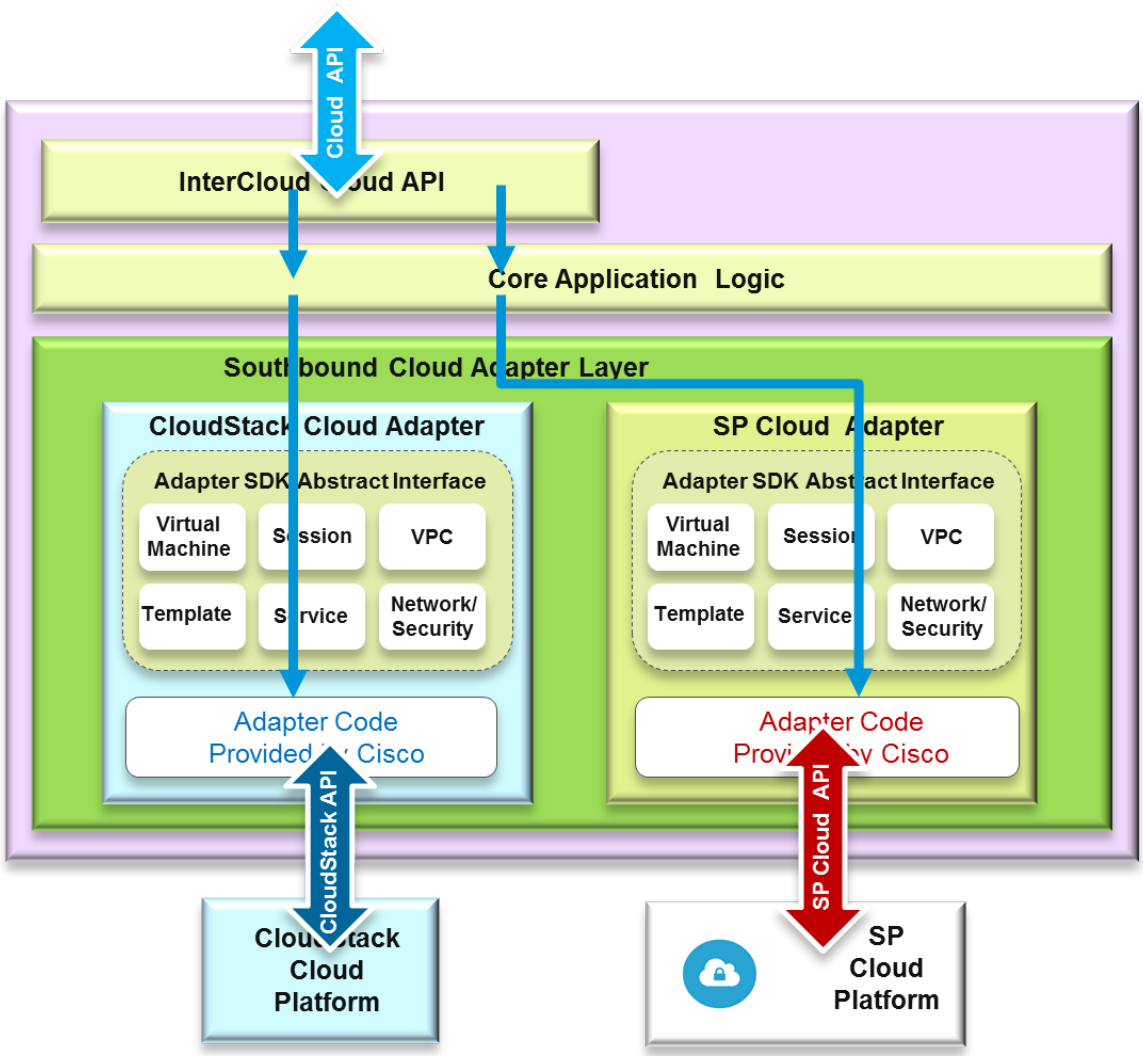


Figure 7 - Cisco ICFPP Cloud Adapter Integration

Custom Cloud Adapter Programming Model

After a custom cloud adapter is developed, the customer can use the following steps to load the adapter plug-in code on to the Cisco ICFPP platform and enable the cloud adapter functions for the targeted tenants:

1. Service provider developers download the cloud adapter plug-in SDK from the Cisco Connection Online (CCO) site for developing a custom cloud adapter.

2. After the custom cloud adapter plug-in code is ready to use, the developer uses standard Linux tools to load the jar file (such as *custom1.jar*) to the file system on the targeted Cisco ICFPP virtual appliance.
3. The service provider administrator uses the Cloud Instance Provision API to add an instance on to the Cisco ICFPP platform. In the Cloud Instance Provision API request, the service provider administrator uses the *cloud module* field to specify the name of the jar file (in this example, it is "*custom1*"). This binds the plug-in code with the cloud instance to be added.
4. The service provider administrator provisions tenants on the Cisco ICFPP platform using the Tenant Provision API and binds the tenants with the newly added cloud instance.
5. When the provisioned tenant issues Intercloud Cloud API requests from a Cisco ICFD platform, the API requests are handled by the newly added cloud adapter plug-on code.

Figure 8 depicts the logical flow for loading custom cloud adapter plug-in code on to a Cisco ICFPP platform and the subsequent processing of incoming Intercloud Cloud API requests issued by a tenant.

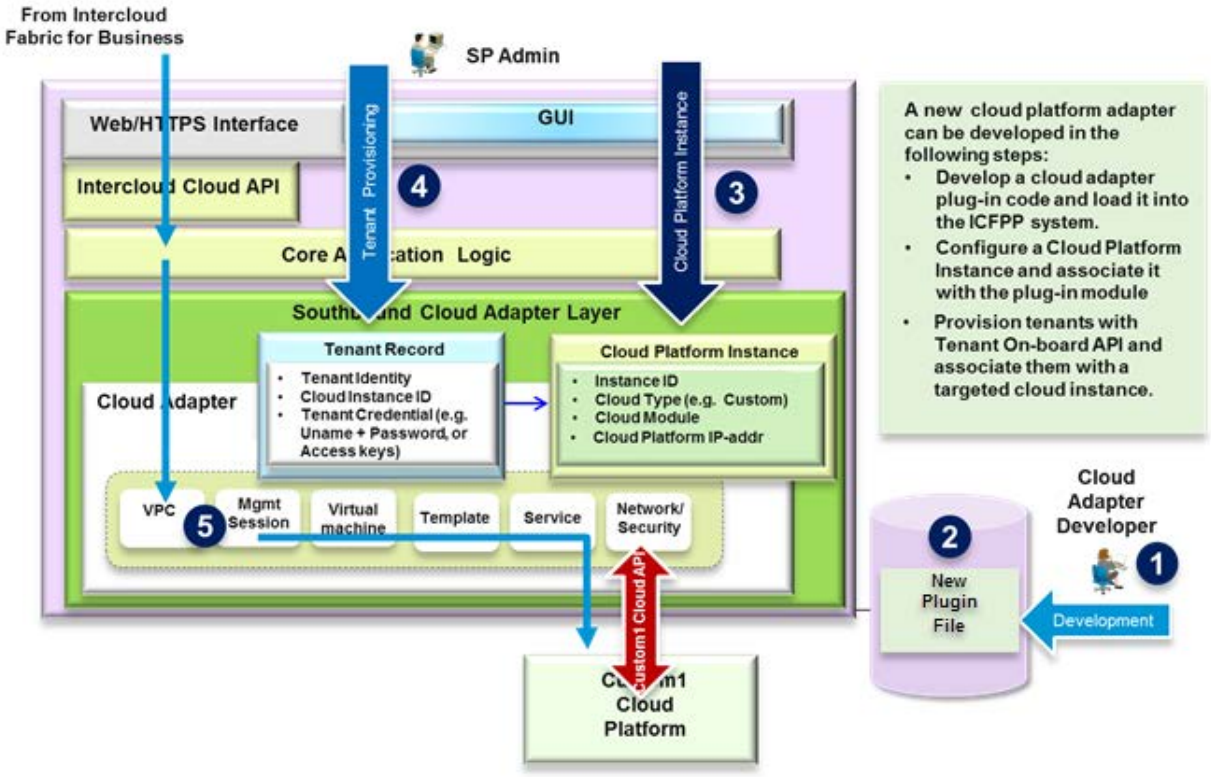


Figure 8 - Cisco ICFPP Programming Model Overview

The following sections summarize the current Southbound API stub functions supported in the Cloud Adapter Plug-in Classes. For detailed information about these APIs, contact your Cisco representative.

Management Session Interface APIs

The Management Session Interface APIs are:

- *createClientSession* – Creates a management session with a specified cloud platform instance.
- *deleteClientSession* – Deletes a management session.
- *validateClientSession* – Validates the current management session.

Network Management Interface API

The Network Management Interface API *listPublicIpAddress* lists the public IP addresses assigned for a tenant user.

Security Management Interface APIs

The Security Management Interface APIs are:

- *createSecurityGroup* – Creates a security group.
- *deleteSecurityGroup* – Deletes the specified security group.
- *updateSecurityGroup* – Updates the specified security group.
- *addSecurityRule* – Adds a security rule.
- *updateSecurityRule* – Updates the specified security rule.
- *deleteSecurityRule* – Deletes the specified security rule.

Service Management Interface APIs

The Service Management Interface APIS are:

- *listCapability* – Lists the cloud platform capabilities.
- *listLocations* – Lists the locations and sites that are supported by the provider.
- *getName* – Provides the name of an adapter.
- *getDescription* – Returns the description of an adapter.
- *getVersion* – Returns the adapter version.
- *getNetworkService* – Returns the adapter class object associated with the network service.
- *getSecurityService* – Returns the adapter class object associated with the security service.
- *getSessionService* – Returns the adapter class object associated with the session service.
- *getStorageService* – Returns the adapter class object associated with the storage service.
- *getTemplateService* – Returns the adapter class object associated with the template service.
- *getVmService* – Returns the adapter class object associated with the VM service.
- *getVpcService* – Returns the adapter class object associated with the VPC service.

Template Management Interface APIs

The Template Management Interface APIs are:

- *createTemplate* – Creates a template based on the specified image.
- *deleteTemplate* – Deletes the specified template.
- *discoverTemplate* – Discovers the Intercloud-ready templates available in the cloud.

VM Management Interface APIs

The VM Management Interface APIs are:

- *deployVirtualMachine* – Deploys a VM using a specified template.
- *destroyVirtualMachine* – Destroys a VM.
- *downloadVirtualMachine* – Downloads the VM disk from the cloud provider catalog to Cisco ICFPP.
- *getVirtualMachines* – Periodically fetches the VM state.
- *listVirtualMachine* – Lists all VMs instantiated by the tenant.
- *rebootVirtualMachine* – Reboots a VM.
- *startVirtualMachine* – Starts a VM that was previously stopped.
- *stopVirtualMachine* – Stops a VM.
- *updateVirtualMachine* – Updates the attributes of a VM.

VPC Management Interface APIs

The VPC Management Interface APIs are:

- *createVpc* – Creates a provider VPC.
- *createVpcNetwork* – Creates a VPC network.
- *deleteVpc* – Deletes the specified VPC.

- *deleteVpcNetwork* – Deletes a specified network from the specified VPC.
- *listProviderVpc* – Lists all VPCs for a specified tenant.
- *listVpcById* – Lists the specified VPC of a specified tenant.
- *listVpcNetworkById* – Lists the specified network for the specified VPC of a specified tenant.

Service Provider Integration Requirements

Cisco ICFPP is designed to provide a simple programmable cloud adapter interface to ease integration with service provider cloud platforms. As described in previous sections, most of the southbound adapter interfaces are straight-forward and adapter developers can develop the interfaces quickly. However, developers must pay attention to the following five areas:

- Deployment Network Topology
- Provider Platform Capability
- Provider Network Models
- Cloud VM Deployment Functions
- Public Network Address Assignment
- Multi-Site Support

Deployment Network Topology

As discussed in other sections, the Cisco ICFPP deployment network topology is relatively straight-forward, but it can vary for different service providers and different cloud platforms.

The guidelines for setting up Cisco ICFPP deployment networks are:

- Cisco ICFPP must be accessible from the Internet and/or service provider-managed Inter-data center networks.
- Service providers can usually expose Cisco ICFPP through a NAT rule configured at a service provider edge router or a web-proxy.
- In general, Cisco ICFPP and cloud platforms should be able to communicate to each other so both of them can initiate management connections with each other.
- If the ability to upload and download images is required and Cisco ICFPP and the cloud platform use shared storage, Cisco ICFPP should be able to reach and access the shared storage networks.

Figure 9 depicts an example of a Cisco ICFPP platform deployed in a typical cloud platform network environment.

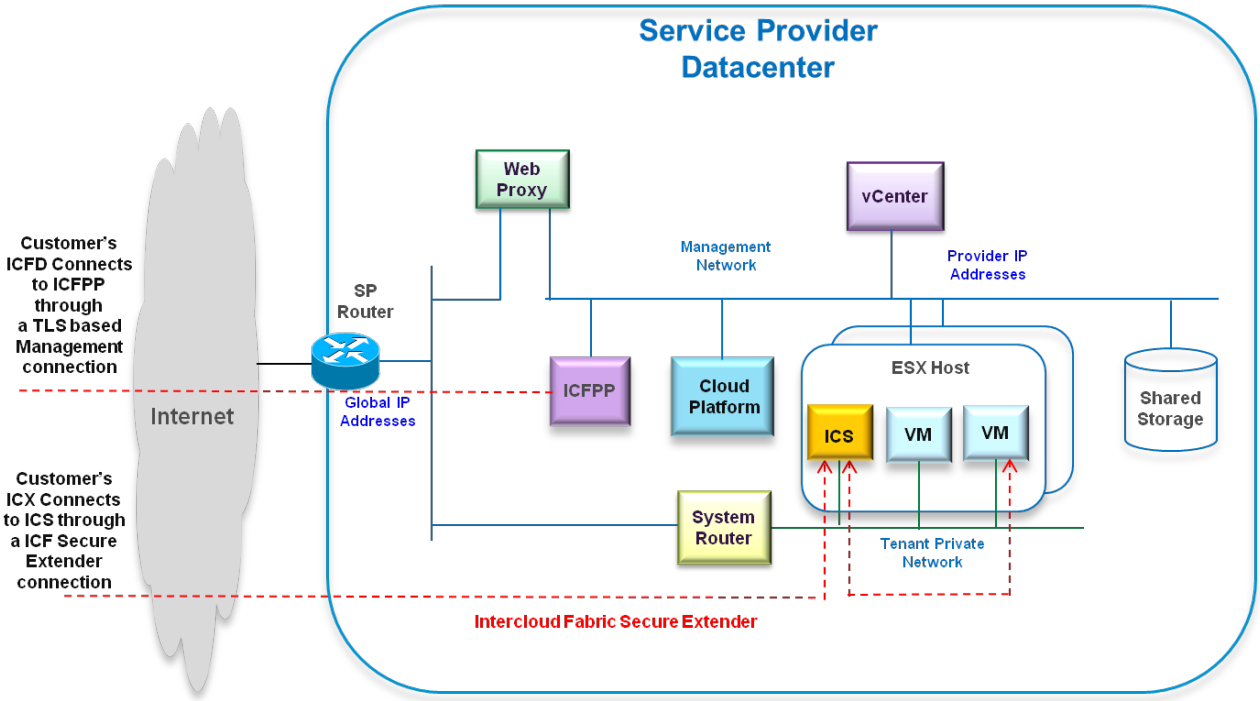


Figure 9 - Cisco ICFPP Deployment Network Topology

Provider Platform Capability

The provider platform must describe the capabilities supported by the platform.

Property	Description	Available Options
cloudAgent	Array of VM Agent types allowed	AWS-getkey, Azure-WAA, Open-Vmtools, Vmware-Vmtools
cloudStyle	Tenant network model	VPC, InstanceBase
containerType	Image container type	OVA
fractionalMemorySupport	Supports fractional memory	True, False
hypervisorType	Hypervisor type	ESXi, Xen-PV, Xen-HVM, KVM
imageFormat	Image format	Raw, VMDK, VHD, QCow2
initIcsOnFirstBoot	Automatically provisions ICS on first boot	True, False
locationHierarchy	Location hierarchy based on location objects	Region, Availability-Zone, Datacenter
partitionType	Disk partition type	{Flat, Multi}
providerName	Provider name	<actual provider name>, such as “Cisco Intercloud Service”
providerNetworkManagement	Provider supports create/update/delete of provider networks	READ, {CREATE, DELETE}
providerVpcCidr	Provider VPC supports CIDR	True, False
providerVpcContainment	Provider VPC containment	Location-Region, Location-Zone, Location-Datacenter
providerVpcManagement	Provider supports create/update/delete of ProviderVpc	READ, {CREATE, DELETE}

Property	Description	Available Options
securityGroupContainment	Security group containment can be based on Location, VM, ProviderVpcNetwork, and ProviderVpc	Location, ProviderVPC, VM
serverCpuRange	Server CPU supported range	1-64
serverMemoryRange	Server memory supported range	1-64 GB
supportedOsTypes	Supported Guest OS type	{RHEL_6, CentOS_6, Windows_2008.R2 (64bit)}
supportedServicesType	Supported service type	{VSG, CSR, CSR-Discover}
templateContainment	Template containment can be based on Location-<locationType>	Location, Location-Datacenter, Location-Zone, ProviderVPC

Based on the platform capability values returned, Cisco ICFD determines:

- How to transform workload image to an appropriate format
- How to allocate required network resources (such as public IP addresses, network segments, subnets, and so on) in order to build the secure network extender
- The platform type to use for deploying infrastructure VMs (such as ICF Router and ICF Firewall)
- How to create template containments based on the provider's cloud site hierarchy (for example, by region or availability zone)
- How to create a security group within a cloud site containment

Provider Network Models

Cisco ICF extends enterprise network segments to public cloud data centers with a secure extender that facilitates enterprise workload migration. The secure network extension is achieved through a Layer 2 overlay network that is built on top of provider networks. As a result, overlay network functions are transparent to the underlying provider network architecture. Nevertheless, the provider network architecture might impact how Cisco ICFD deploys infrastructure VMs and builds the secure extender operationally.

Cloud service providers commonly offer networks two types of network architectures:

- Shared Network—All tenants share the same network infrastructure; isolation is achieved through security-groups.
- Isolated Network—A tenant can dynamically create one or more tenant-specific isolated networks, which protect and isolate the tenant from others.

Depending on the values of *cloudStyle* and *providerVpcManagement* returned from provider platform, Cisco ICFD must proceed through different operations to build the overlay networks and deploy infrastructure VMs such as ICF Router and ICF Firewall. Figure 10 depicts how ICF Secure Extender can be built on a shared network architecture. Because all overlay network functions are provided by the infrastructure VMs such as ICS, ICF Router, and ICF Firewall, the only infrastructure requirements needed from provider are:

- Cloud VM deployment
- Provider private IP address assignments
- Provider public IP address assignment to ICS

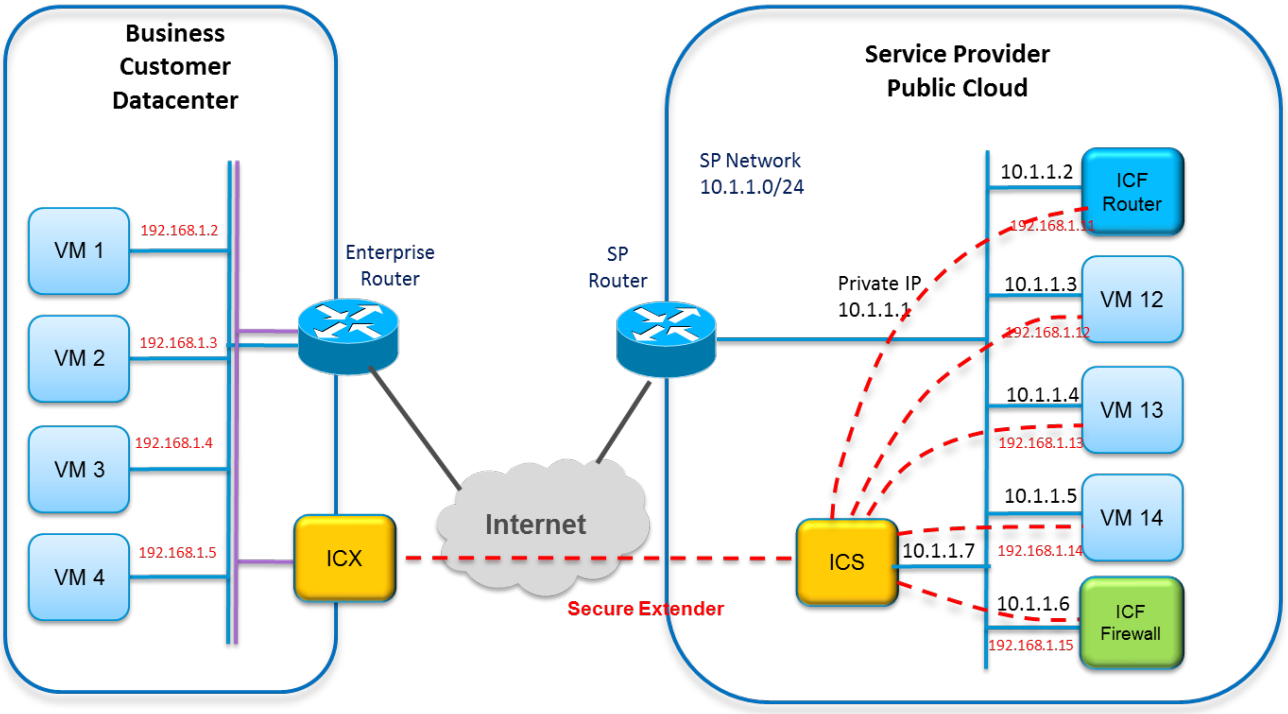


Figure 10 - Secure Extender built on a Shared Provider Networks

Figure 11 depicts how ICF Secure Extender can be built on an Isolated Network architecture. In addition to the infrastructure requirements mentioned previously, Cisco ICFD must be able to access provider Network Management APIs for dynamically allocated networks. In many cases, provider-isolated network functions are built on top of a VPC-based tenant provisioning architecture, which not only isolates the network, but also compute and storage resources from other tenants.

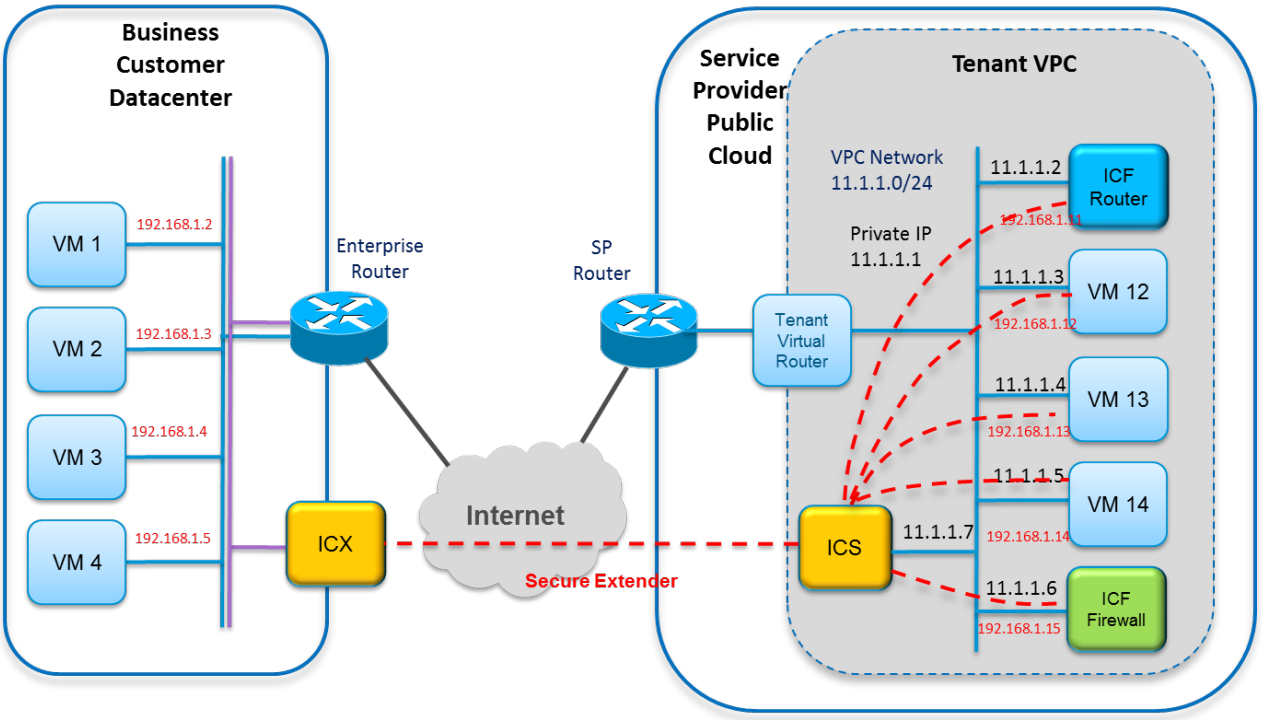


Figure 11 - Secure Extender built on an Isolated Per-Tenant Network

Cloud VM Deployment

In current Cisco Intercloud Fabric use cases, four types of cloud VMs must be deployed in public cloud provider data centers:

- Application VMs—VMs running business customers' applications.
- ICF Switch (Infra-ICS)—VMs providing Cisco ICF switched overlay functions.
- ICF Router (Infra-CSR)—VMs providing routing, NAT, and VPN functions.
- ICF Firewall (Infra-VSG)—VMs offering firewall and isolation functions.

Some of the infrastructure VMs impose certain deployment requirements, which are summarized in the following table:

Provider VM Deployment Requirements	**Cloud VM Types**			
	ICF Router (Infra-CSR)	**ICF Firewall (Ingra-VSG)**	**ICF Switch (Infra-ICS)**	**Application VM**
O/S Platform Supported	64-bit Monta Vista Linux with 2.6.32 Kernel	32-bit NXOS Linux with 2.6.27 Kernel	64-bit Ubuntu Linux with 3.10 Kernel	RHEL, CentOS, Windows 2008 R2
OVF Customization	Required	N/A	N/A	N/A

The OVF customization required for ICF Router (Infra-CSR) and Infra-VSG enables the cloud orchestrator (in this case, Cisco ICFPP) to pass a set of OVF configuration parameters to the ICF Router programmatically during VM instantiation. Depending on the approach that the service provider cloud platform takes to support the OVF parameter-passing feature, the OVF customization process can be different for different service providers.

Using the CloudStack platform as an example, the OVF parameter-passing requirement in a CloudStack environment can be achieved by completing the following steps:

1. Obtain all necessary OVF parameters (such as VM IP address, VM netmask address, gateway IP address, DNS, domain name, hostname, and so on) from the service provider. Cisco ICFPP uses this information to create an OVF configuration XML file, named "ovf-env.xml".
2. Convert the "ovf-env.xml" file to an ISO image and, using the API *registerIso* ( ), register the image so that it can be attached to a specific user.

   http://cloudstack.apache.org/docs/api/apidocs-4.1/domain_admin/registerIso.html
3. Deploy the targeted VM in a Stopped state (that is, with the *startvm* parameter set to False) with the API *deployVirtualMachine* ( ).

   http://cloudstack.apache.org/docs/api/apidocs-4.1/domain_admin/deployVirtualMachine.html
4. Attach the OVF configuration xml file, "ovf-env.xml" to the VM that was deployed with the API *attachIso* ( ).

   http://cloudstack.apache.org/docs/api/apidocs-4.1/domain_admin/attachIso.html
5. Start the VM with the API *startVirtualMachine* ( ); the VM will pick up the parameters passed through the configuration file.

   http://cloudstack.apache.org/docs/api/apidocs-4.1/domain_admin/startVirtualMachine.html

For more details of VM deployment, see the VM Management Interface APIs.

Public Network Address Assignment

In certain Cisco ICF use cases, cloud VMs might need to communicate with systems on external networks such as the Internet. These uses cases include the following:

- An application VM hosting a web server
- An application VM that must communicate with external web servers
- An application VM offers VPN services for branch offices
- An infrastructure VM (such as Infra-ICS) that provides secure network extension for an enterprise.

In Cisco ICSE architecture, the majority of external connectivity is handled by the ICF Router (CSR). The only exception is the ICF Switch (ICS), which is responsible for establishing Cisco ICSE by interconnecting with the Intercloud Extender (ICX) running in the enterprise. The following table summarizes the public IP address assignment requirements and how relevant NAT rules are used in different systems.

ICF Use Cases	Infrastructure Configuration Required			
	Application VM	**ICF Router (Infra-CSR)**	**ICF Switch (Infra-ICS)**	**Provider Router**
Hosting a Web Server	Set Default Gateway to CSR	NAT rules for translating Enterprise Private IP Address to Provider Private IP address	N/A	NAT rules for translating Provider Private IP Address to Provider Public IP address
Connecting to Internet Servers				
Offering VPN Services				
Offering Secure Extension	N/A	N/A	Set Default Gateway to Provider Router	

For more details of VM public network address assignment, see the VM Management Interface APIs.

Multi-Site Support

It is very common for a provider cloud platform to support multiple regions and/or multiple availability zones for scaling and high availability.

A *region* has a separate API endpoint per cloud platform deployment, allowing for a more discrete separation. Users who want to run instances across sites must explicitly select a region. An a*vailability zone* is a logical separation within a cloud platform deployment for physical isolation or redundancy. When users provision resources, they can specify the availability zone from which they want their instance to be built. This allows cloud consumers to ensure that their application resources are spread across disparate machines to achieve high availability in the event of hardware failure.

To support this multi-site requirement, Cisco ICF architecture is designed to honor the following guidelines:

- Cisco ICF assumes that a service provider will deploy one Cisco ICFPP virtual appliance per region and provide separate URLs (or IP addresses) for each Cisco ICFPP virtual appliance.
- After the Cisco ICFPP cloud adapter has signed on with the cloud platform, Cisco ICFPP first uses the southbound adapter API *listLocations* to determine how many availability zones are supported by the platform and then reports the realized availability zones to Cisco ICFD.
- Upon receiving the availability zone information, Cisco ICFD reflects the information to IT administrator and end-user portals, where users can specify the availability zone from which they want their instance to be built.

Figure 12 depicts how multiple sites are supported in Cisco ICF. For more details of multi-site support, see the Service Management Interface APIs (such as *listLocations*).

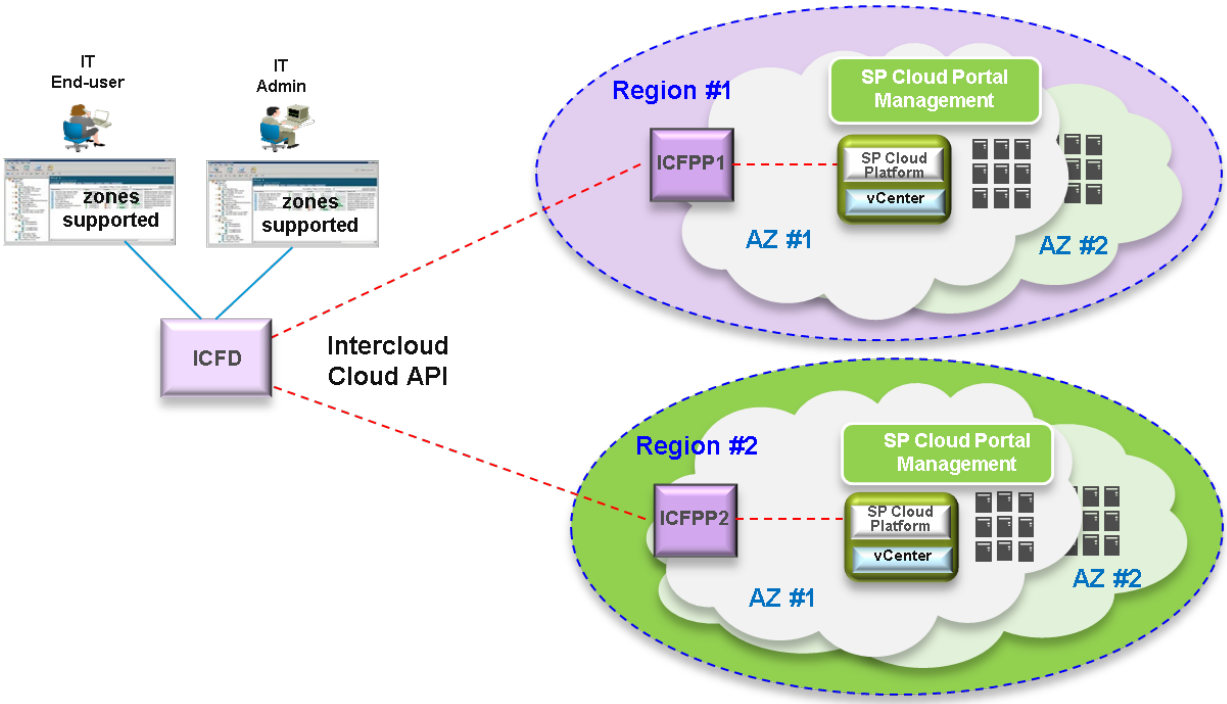


Figure 12 - Intercloud Fabric Multi-Site Support Overview

Conclusion

Cisco ICF addresses the most common needs surrounding hybrid cloud adoption. It creates a seamless environment for enterprise customers within hybrid cloud situations, and enables service providers with the capability to present their public cloud offerings for consumption to these enterprise customers.

Cisco ICFPP simplifies and abstracts the complexity of communicating with different cloud platforms, and enables API support for service providers that currently do not offer API support. Cisco provides a set of built-in cloud adapters for interfacing cloud platforms such as VMware vCloud, CloudStack, Cisco Intercloud Services – V, and OpenStack.

For other cloud platforms supported by service provider customers, Cisco offers an extensible Cisco ICFPP Cloud Adapter SDK framework, which allows customers to develop their own cloud adapter seamlessly. This approach is built from the ground up, leveraging and offering APIs to offer greater flexibility of implementation and to ensure a wider range of independent integration.