SOLUTION OVERVIEW

How to Quickly Deploy Apps across Multiple Clouds

Multi-cluster app mesh is a key element of the F5 Distributed Cloud Services multi-cloud networking solution, dramatically reducing operational complexity and cross-cloud application deployment times.

KEY BENEFITS

Faster deployments and simplified ops
Developers can simplify deployment of one or more Kubernetes ingress and egress controllers with SaaS-based lifecycle management and a multi-cluster control plane.

Maximized security
Implement multi-layer security in and across clusters, including ingress and egress, WAF, and DDoS mitigation.

Increased uptime and reliability
Deliver highly available services across clusters or clouds. Use the F5 Global Network to connect across clusters and securely expose services to the Internet.

Improved developer experience
Increase productivity by delivering APIs without VPNs or complex firewall configurations, allowing for faster changes and troubleshooting.

Apps and APIs are protected outside your perimeter
Distributed Cloud Services (including WAF, API security, and JavaScript Challenge) offer protection to wherever the app or API is advertised, even outside your perimeter.

Connecting Clusters Between Clouds Using Conventional Tools

As organizations expand their cloud strategies to multi-cloud, they seek solutions to not just cross-connect virtual networks, but to interconnect applications and services hosted in different clusters.

Applications may need to connect to specialized services such as analytics, or to external identity and access management sources, or to in-house data or services that have historically been hosted in a different cloud. Thus, organizations are discovering that the task is more challenging than originally thought.

App-to-app connections have different requirements than human machine interactions. As application architectures are changed from “scale up” to “scale out” and network traffic increases between machines, the new east-west traffic paradigm complicates network design and security enforcement.

Within a single cluster, apps can reach each other because the app-to-app connections are typically within the same IP address scope, which is similar to being in the same local network. In other words, apps and services can connect to each other across different nodes within the same cluster. However, each cluster usually has its own private IP address scope in its internal network, so simple connectivity between clusters won’t work. Inter-service connections between clusters therefore need some method of exchanging metadata like service advertisements. Traditional VPNs that just provide Layer 3 IP connectivity are insufficient for app-to-app communication.

Application speed is heavily impacted by east-west network latency, with delays directly correlating to the product of the latency between clusters and the number of request/response pairs or “app turns.” A single API call in a core loop can create a large delay as each iteration waits for the response from the remote service. While the Internet is reliable for the general delivery of traffic, most providers optimize their networks for capacity, not latency. Plus, each additional “hop” between providers just makes the trip slower. Therefore, it is important to have visibility and control of the path of east-west connections between different regions or clouds.

Security is potentially the trickiest aspect of app-to-app connections. Within a cluster, the connections and permissions are typically orchestrated by a service mesh. In the service mesh, each node communicates with a central controller, which informs both endpoints about the expected connection and metadata, such as identity. However, once an outbound connection leaves a cluster, its situational context and metadata is lost. Even within secure transport, such as a VPN, the remote cluster will have to rebuild all security context for access to the service.

Using conventional tools, complications like these may explain why enterprises today may require weeks or months to deliver an application across multiple clouds.

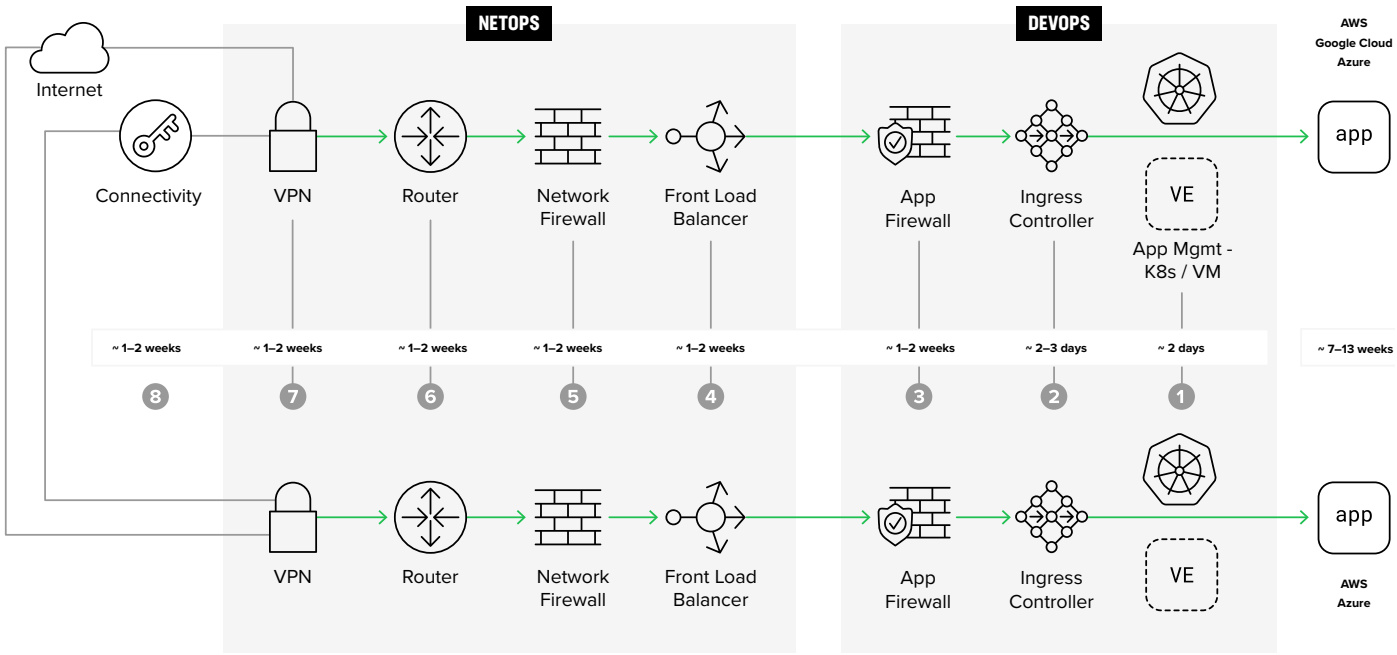

Figure 1: A graphical depiction of the steps involved in delivering an app across multiple clouds using conventional tools and cloud providers.

How the F5 Distributed Cloud Platform Reduces Cross-Cloud App Deployment Time

APP-TO-APP CONNECTIONS HAVE DIFFERENT REQUIREMENTS THAN HUMAN MACHINE INTERACTIONS. AS APPLICATION ARCHITECTURES ARE CHANGED FROM "SCALE UP" TO "SCALE OUT" AND NETWORK TRAFFIC INCREASES BETWEEN MACHINES, THE NEW EAST-WEST TRAFFIC PARADIGM CREATES CHALLENGES IN NETWORK DESIGN AND SECURITY ENFORCEMENT.

The F5® Distributed Cloud Platform, with multi-cluster app mesh, securely delivers an application across clouds in minutes rather than days or weeks—a fraction of the time it takes using conventional tools. More on that below. First, let's explore why conventional tools take more time and resources to connect apps across clouds.

To provide an example here, let's talk about Company A, which wants to offer private access to an application across different clouds. This includes not only clouds the company owns but also the cloud for a partner site or branch, where they don't control the configuration or the end user.

Company A wants fine-grained controls over which services it delivers to which partners and branches. For API security, it needs to control the HTTP methods, the routes, and the specific APIs involved. It also requires protection as well as visibility for what is happening related to the service being advertised.

KEY FEATURES

Secure and reliable connectivity
Secure and redundant tunnels between sites are automatically created via any specified transit, including the F5 Global Network.

Service discovery and remote advertisement
Service discovery is enabled at every cluster via DNS, Kubernetes, etc., with policy-driven service export to other sites and clusters.

End-to-end observability
A single, unified dashboard provides widespread visibility across all company and partner sites.

API gateway auto-discovery and control
Work smarter and faster with API auto-discovery, granular cross-site policy controls, and machine learning-driven anomaly detection.

Secure ingress and egress Kubernetes gateway
Leverage an ingress-egress controller with integrated network and security services for multiple app clusters across the WAN.

Three IT teams would traditionally be responsible for enforcing these controls internally: (1) the app team building the application; (2) DevOps, which is responsible for the app deployment and infrastructure like the firewall to control access to the app across clouds and sites, and (3) NetOps, which will use a service-ticket rule in the firewall to allow certain traffic in and block out all malicious traffic.

In delivering the app to the partner branch, a partner team must commit to taking similar steps to secure the service as it is delivered to its end users. The partner may also have to cope with IP address overlap, set up a VPN connection, and establish their own routing to the original company's site.

There are other time-consuming steps involved and challenges related to this multi-cloud app deployment:

- **Changes to routes and firewalls.** With routes being changed, the entire network potentially could be exposed—a major concern if you are advertising only one service. These may be required by both the company's IT team and the partner team, due to variations in cloud access rules and other complications.
- **Rules regarding service and API access.** Services that have traditionally been delivered together may be difficult to serve individually, especially if those services have been delivered as API trees on the same site. A traditional firewall 5-tuple lacks the granularity to control access only to portions of a site.
- **Safeguards on the company site.** The site of origin must protect against threats and potential attacks coming from unexpected sources if the partner unexpectedly exposes the service more broadly than intended.
- **Visibility across the company's network.** Logs and dashboards are needed to track metrics across the company network, both for troubleshooting and for performance monitoring.
- **Troubleshooting complexity.** If there's a problem between the partner and an end user, the company may not have access to enough information to isolate the problem: Is it a problem with the service, the partner, or the end user?

By comparison, multi-cluster app mesh from F5 Distributed Cloud Services offers an easy solution to connect clusters, regardless of cloud provider or region.

This F5 solution provides orchestrated awareness for the API gateways on all connected clusters, allowing cross-cluster service discovery and advertisement for seamless app-to-app communication with fine-grained API control. Connections between sites are self-maintaining, redundant, and fully automated, relieving administrators of tasks such as establishing VPNs and routing.

F5 DISTRIBUTED CLOUD MESH WORKS WITH A VARIETY OF CLUSTERS, INCLUDING MOST KUBERNETES ENVIRONMENTS.

Customers have end-to-end visibility. They can choose their underlying transport, including the option of the F5 Global Network, which is purpose-built for reliable high-speed app-to-app connections.

F5® Distributed Cloud Mesh works with a variety of clusters, including most Kubernetes environments. Distributed Cloud Mesh can connect natively to Kubernetes to discover services, advertise specific services to remote clusters across clouds, and distribute security policies across clouds to protect the advertised services.

In other cluster environments, Distributed Cloud Mesh has multiple options for remote service advertisement and delivery, including presenting the remote service with an IP address local to its consumers for the ultimate in no-routing-required reachability.

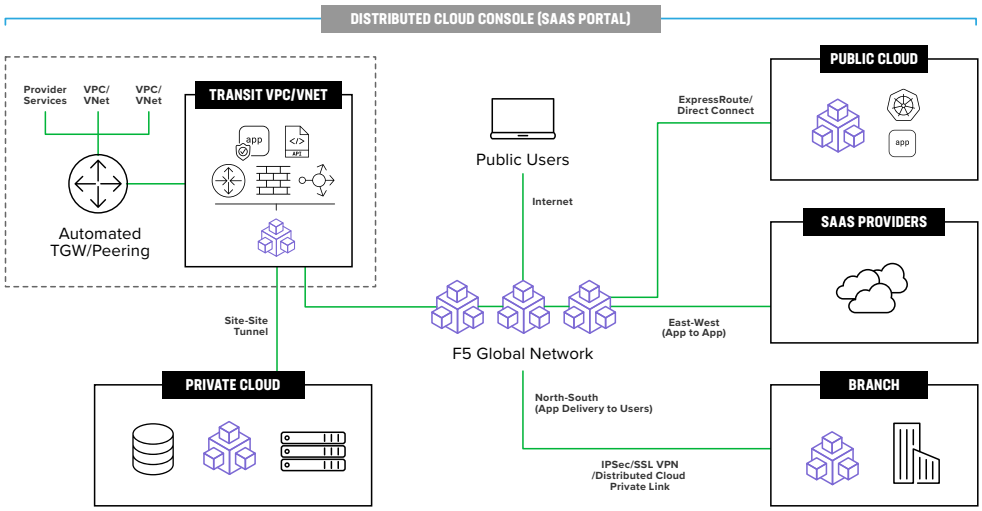

Figure 2: An architectural view of multi-cluster app mesh in F5's multi-cloud networking solution, which reduces the time and complexity in connecting apps across public and private clouds.

Conclusion

Here are three important ways that multi-cluster app mesh reduces operational complexity and time involved in cross-cloud app deployment:

1. **Multi-cluster app mesh enables app-to-app connectivity at L7 without exposing the underlying L3 network.** This helps lower security risks to speed deployment. IP overlap concerns are minimal, and most security risks are avoided because of the protection of the L3 network.

2. **It provides fine-grained control of all APIs and methods.** Intent-driven policies can be distributed to any or all sites to add load balancing, API security, and other enhancements to all required locations. There are also protections that can be injected above the network layer, such as TLS fingerprinting for high assurance of client identity, and a JavaScript challenge to create barriers to automated site scraping. You have complete control over what's being advertised and exposed.
3. **It delivers single-pane-of-glass cross-functional end-to-end control and observability.** In the unified SaaS console, the configuration for delivery of each app is aware of the settings for its network and security, which simplifies setup and maintenance. The integrated stack architecture also provides a single source of truth for troubleshooting, combining visibility for every site and function with application context and instrumentation for modern cross-cloud observability.

Remember how Company A, discussed earlier, wanted to perform a controlled offering of one of its APIs? Distributed Cloud Mesh quickly discovers all the relevant apps and services available using the cluster's native method, such as Kubernetes or DNS. Exporting and re-advertising is controlled by policy, with fine-grained controls, integrated firewall and WAF, and multiple delivery options per remote site. App reachability and health status are continually delivered between sites across the control plane, with continuous verification from each site to detect issues proactively.

With each of those remote sites, there's no IP overlap issues, no complex routing, and no one-off security settings to configure. As the number of remote sites gets larger, management is still straightforward, scaling from a single cluster up to hundreds of thousands of remote sites managed as a fleet. All aspects of all sites are centrally managed for true end-to-end control and observability.

Company A is now satisfied with stability and agility and is starting to explore self-service multi-cluster networking for dev teams for its next generation of APIs, which will be delivered using F5 Distributed Cloud Services.

Sign up for a free trial or contact your local F5 representative for a solution demo and more details.