IAMONDEMAND

IOD

[tech.content]

MULTICLOUD NETWORKING

THE DEFINITIVE GUIDE

By:
Milos Dodic
IOD Expert

{ INTRODUCTION }

In 2019, Gartner predicted that by 2021 up to 75% of large and midsize organizations would be running some variation of multicloud environments. This IT trend is due in part to the modern necessity of working on multiple cloud platforms, and in part because it's easy to do—though your cloud architects and DevOps engineers may not agree.

Possibly the most painful part about adopting a multicloud strategy is connecting your public cloud providers to each other. If that newly designed multicloud setup also connects to your on-premise workloads, it can be even harder to implement and maintain. But these challenges don't have to stand in your way.

In this comprehensive guide on multicloud networking, I'll demystify the process of developing a multicloud strategy. It will start by going over theories and concepts, later there will be practical examples of connecting AWS and GCP.

Pros and Cons of a Multicloud Strategy

The following list of benefits—and even a few drawbacks—are important elements to consider before implementing a multicloud strategy.

Benefits

The most obvious benefits of multicloud adoption are increased fault tolerance and the elimination of vendor lock-in, but there are a number of other reasons as well.

Eliminating Vendor Lock-In

By running workloads on at least two public cloud platforms, you aren't limiting yourself to the services that one provider offers. This means that as your cloud architects develop their environments they have the freedom to consider:

- A given provider's features
- The complexity of using services from multiple providers
- The costs associated with using multiple providers

High Availability

Because it's unlikely that two major public cloud providers would have outages at the same time, deploying across providers can ensure your enterprise maintains high availability.

For example, in 2017, the AWS S3 service went down in the us-east-1 region. The companies running only in AWS would have been able to maintain service if they'd had failover deployments on Azure or GCP, since both Google and Microsoft have their data centers in North Virginia where the AWS us-east-1 region is located.

Managed Services

Enterprises often overlook managed services and decide to provision their own, self-hosted solutions instead of opting in for the SaaS workloads. This may create a burden for their engineering teams, and might even result in added costs. If a public cloud provider doesn't offer a managed service tailored to your needs, going multicloud will allow you to utilize a competitor's services instead of trying to go it alone.

Amazon, for example, has a managed MongoDB database service called DocumentDB, managed Prometheus and Grafana monitoring offerings that require no work to set up, and managed messaging services like Kafka, ActiveMQ, or RabbitMQ.

Azure and GCP don't have those listed services, other than Azure Service Bus which has support for ActiveMQ. This means that if your primary cloud provider is Azure or GCP, you may turn to AWS for these kinds of features.

Drawbacks

Despite their many advantages, multicloud strategies are not without drawbacks.

The primary disadvantage of working across cloud providers is the complexity of designing, deploying, and maintaining effective multicloud environments. Public cloud providers make an effort to ease this burden by sharing some concepts—like the shared responsibility model, assisting with the abstraction of hardware, and offering similar pricing models. But the setup of core multicloud functionalities, such as networking, can be cumbersome even for experienced IT veterans and network engineers.

Before I dive into multicloud networking examples, I'll illustrate the fundamentals of this pillar of multicloud adoption and highlight the main services available in all three major public cloud providers.

{ What is Multicloud Networking? }

Multicloud networking connects on-premise workloads with at least two public cloud providers to create a mesh network topology.

In hybrid cloud topologies, your cloud architects and network engineers have the goal of connecting your on-premise IT infrastructure with a public cloud provider. That infrastructure can range from a group of servers and network firewall or router devices in your office to a full-fledged datacenter environment. With multicloud networking, this becomes exponentially more complex.

In essence, multicloud networking is a level of complexity above hybrid-cloud networking. This is because it requires you to connect two or more public cloud networks to your on-premise network in addition to connecting your public cloud networks to each other.

A Multicloud Use Case

How does a multi-cloud networking setup look in the environments of big enterprises? Let's take a look at a real-world example.

Let's say that you've been running on-premise workloads in your datacenter. You've connected it to AWS to extend your on-premise workloads with services such as S3, RDS, DynamoDB, or other managed offerings. Now your team of system administrators want to extend the internal Active Directory environment with Azure AD, and to migrate internal Exchange servers to Office 365. Their goal is to streamline email services and offload operational and management tasks to Microsoft.

To successfully introduce Azure into your global enterprise networking topology, you would need to include another networking site for Azure, and two more connected network endpoints: AWS to Azure, and your on-premise datacenter network to Azure.

In the end, you will have three isolated network segments (AWS, Azure, and your datacenter) each of these will require a /16 subnet mask, or smaller, if you have the need for more network hosts. You will also have three links connecting these segments (datacenter to AWS and Azure, respectively, and from AWS to Azure).

This change would involve even more complexity and configuration, but luckily you can reuse the networking concepts and protocols you've already been using in your hybrid deployments. For encryption of the network tunnels between public clouds and your datacenter, you will probably rely on a IPSec/IKEv2 comb, with your own or PKI infrastructure from the public cloud providers. When it comes to routing, the most logical choices are either to use static routes or BGP for dynamic propagation. However, EIGRP and OSPF are also supported if you deploy cloud-hosted routers that support these vendor-specific routing protocols.

Additional options, beyond IPSec tunneling, include:

- MPLS
- VPLS
- VXLAN
- L2TPv3

Other options are available, but the best fit for your needs will depend on what equipment you have on-premise, how you decide to connect your clouds and datacenter, and if you use collocation peering. All the major public cloud providers offer collocation through their services and partners and we'll explore those options in greater detail in the next section.

{ Necessary Services for Multicloud Networking }

When it comes to cloud networking concepts, AWS paved the way for other public cloud providers. Public cloud networks introduced the idea of abstracting firewalls and routing networking devices into a unified configuration that can be performed through a variety of methods, including:

- Web console
- SDK
- Command line interface
- A variety of third-party, open-source, or commercial tools

Regardless of which provider you use, you'll need to provision many of the same services:

- **VPC:** VPC, or virtual private cloud, is an isolated network segment in which you provision subnets and access control lists.
- **Subnets:** VCPs are segmented into subnetworks which you decide to make either public (available outside of your VPC, from public internet) or private (only visible inside the VPC).
- **Gateways:** This includes an internet gateway for internet access, a network address translation (NAT) gateway to allow hosts in private subnets to have outbound connections to internet (for example, if you need to patch your servers), or VPN gateways (for establishing VPN site-to-site connections with other public clouds or your on-premise environments).
- **ACLs / firewall rules / security groups:** The cloud is already abstracting your firewall devices, but you still need to specify some rules or access control lists with which to control traffic in your VPC. These rules can be defined on the subnet or host level, depending on the service and public cloud provider you use.
- **DNS:** You will need to rely on cloud-managed DNS services to make internal DNS queries resolve between all interconnected networking sites.
- **Private (direct connectivity):** This is the only network segment that is not standardized between cloud providers. AWS, Azure, and GCP all have separate services, vendors, and partners that directly connect your collocated on-premise network environment to their cloud via private, dedicated network connections.

The following table shows the main network services from each of the major public cloud providers:

Service	Amazon Web Services	Microsoft Azure	Google Cloud Platform
Virtual Networking	AWS VPC	Azure VNET	GCP VPC
VPN	AWS VPN	Azure VPN Gateway	GCP Cloud VPN
Private connectivity	AWS Direct Connect AWS PrivateLink	Azure ExpressRoute	GCP Cloud InterConnect
DNS	AWS Route 53	Azure DNS	GCP Cloud DNS

Table 1: Common network services offered by the Big 3 cloud providers

Now that we've established some of the fundamentals of multicloud networking, let's dive into some practical examples.

{ ACME Corp Goes Multicloud }

For the purpose of this demonstration, I'll focus on the famous (and fictional) ACME Corporation and their cloud journey.

Let's say ACME established its data center decades ago, but when AWS launched in 2006 they decided to join forces with Amazon, starting with popular AWS services like EC2, S3, and RDS. A couple of years later, they utilized AWS Site-to-Site VPN to establish a private cloud network and eliminate the need to use AWS services over the public internet.

Cloud engineers and technical management were happy with what AWS added to ACME Corp's IT needs, but started to fear vendor lock-in. What if Amazon decided to stop offering hosting services to companies they deemed competition? If they did, ACME would have to scramble to find another provider.

ACME's tech personnel wanted anyway to improve costs and try services that AWS doesn't offer by working with multiple cloud providers.

Kubernetes with the managed GKE service was especially interesting to ACME. They reasoned that, since Google invented Kubernetes, GKE will offer them the most features out of the box, and would be the first among public cloud providers to introduce newer versions and major updates.

With this in mind, the decision was made to implement a multicloud network with Google Cloud Platform and to connect it to their existing AWS network and on-premises data center.

But where to start?

Designing a Blueprint for Multicloud Networking

ACME Corp's business focuses on the U.S. market; so they've organized their AWS presence in three regions: **us-east-1**, **us-east-2**, and **us-west-2**, with **us-east-1** as their primary region. Their compute workloads are based on EC2 instances deployed across all three regions, with connectivity between regions maintained by VPC peering. VPN connection to the data center is established with redundant VPN tunnels terminating in the **us-east-1** region.

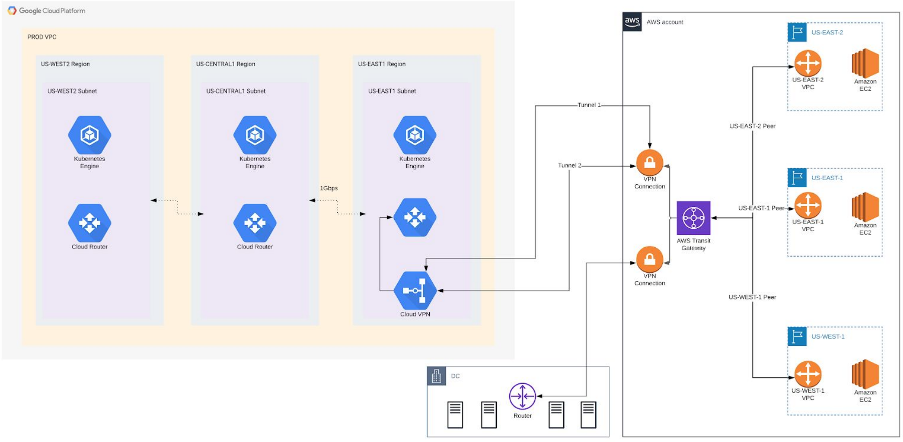

Figure 1: AWS, GCP and on-premises data center connected in a multicloud network

Naturally, they want a similar setup on GCP, including three regions, a K8s cluster in each, and VPN termination in the **us-east1** GCP region. This should provide the least latency between their primary regions in both public clouds.

The diagram in Figure 1 makes the plan look simple. Deploy a GCP environment, peer GCP networks, set up VPN connection with two tunnels, connect to AWS, and voila! They've gone multicloud.

But in order to implement this simple plan, they must first make other changes based on Amazon and Google recommended best practices.

(Re)Configuring the AWS Side

Before introducing another public cloud provider into its network topology, ACME Corp has to follow Amazon's recommended best practice for establishing VPN connections: one VPN connection to the on-premises data center with two redundant VPN tunnels. For that setup, they'll need VPN Gateway (to define the IPSec settings on the AWS side), and Customer Gateway (to define IP settings from their data center environment).

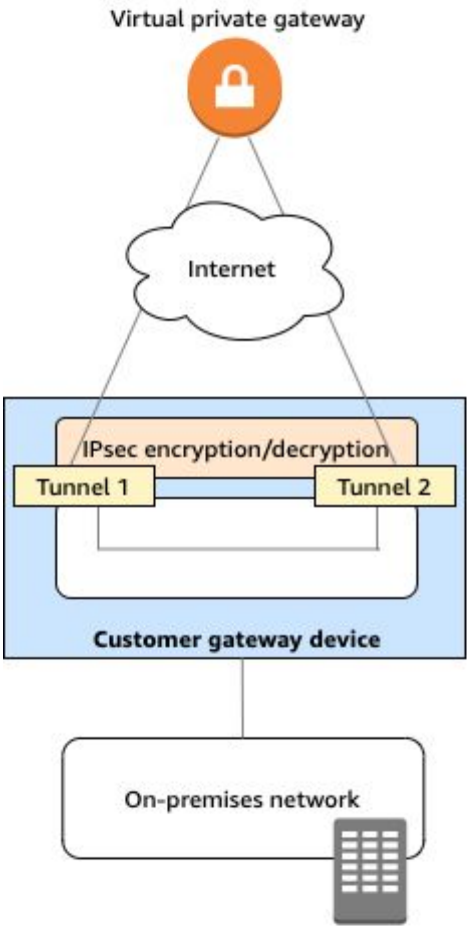

Figure 2: AWS Site-to-Site VPN connection (Source: aws.amazon.com)

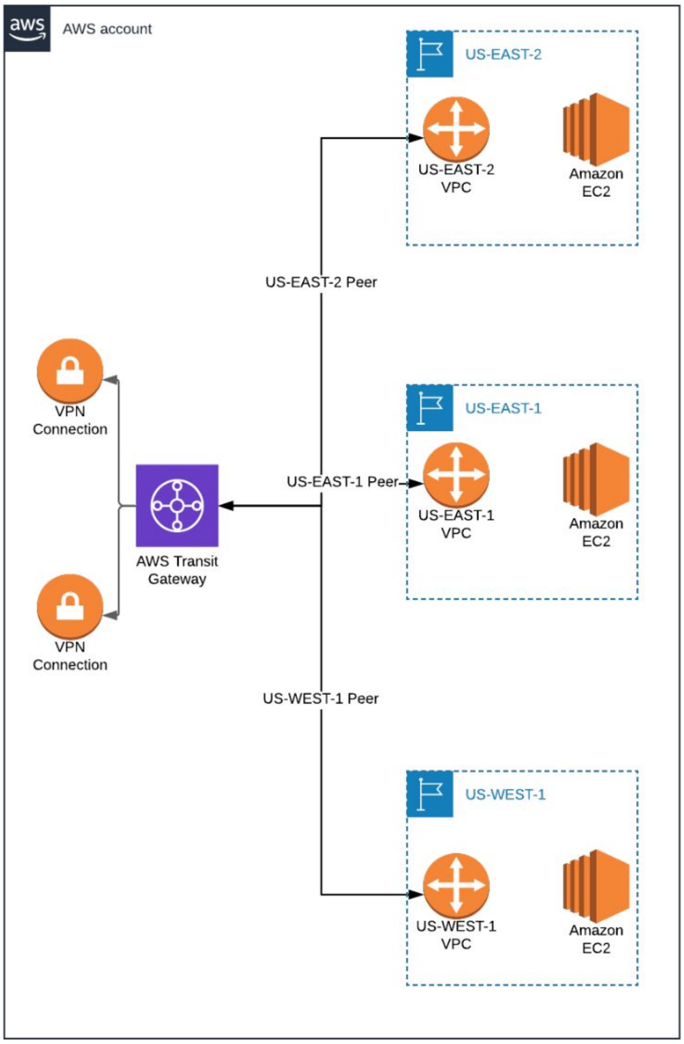

Figure 3: Connecting AWS VPCs with Transit Gateway

In order to maintain connectivity between all of its virtual private clouds in U.S. regions, ACME Corp has to establish multiple VPC peerings and maintain multiple route tables—even with BGP dynamic routing protocol.

The solution to this situation was released in 2018, with AWS Transit Gateway. This managed network service allows enterprises to connect thousands of VPCs and on-premises networks using a single gateway.

With the introduction of Transit Gateway on ACME Corp's AWS networking side, they'll follow the required configuration:

- Set up three Transit Gateway Attachments to connect all three VPCs to the Transit Gateway)
- Reconfigure the current VPN to on-premises to use Transit Gateway instead of VPN Gateway
- Set up new VPN connection to GCP on Transit Gateway

Instead of maintaining two Customer Gateways, two VPN Gateways, three VPC peerings, and multiple route tables (for each VPC and VPN), AWS makes it possible for ACME to consolidate all of their networking configurations on one Transit Gateway with one Transit Gateway Route Table.

Much to their relief.

{ Configuring the GCP Side }

ACME Corp wants to replicate their region placement on the GCP side: three regions distributed throughout the U.S. to minimize latency for customers, regardless of where they're located in the country.

Since Google has central-U.S. cloud regions (compared to AWS' East and West Coast regions), ACME has made the smart choice to deploy some workloads in the **us-central1** GCP region.

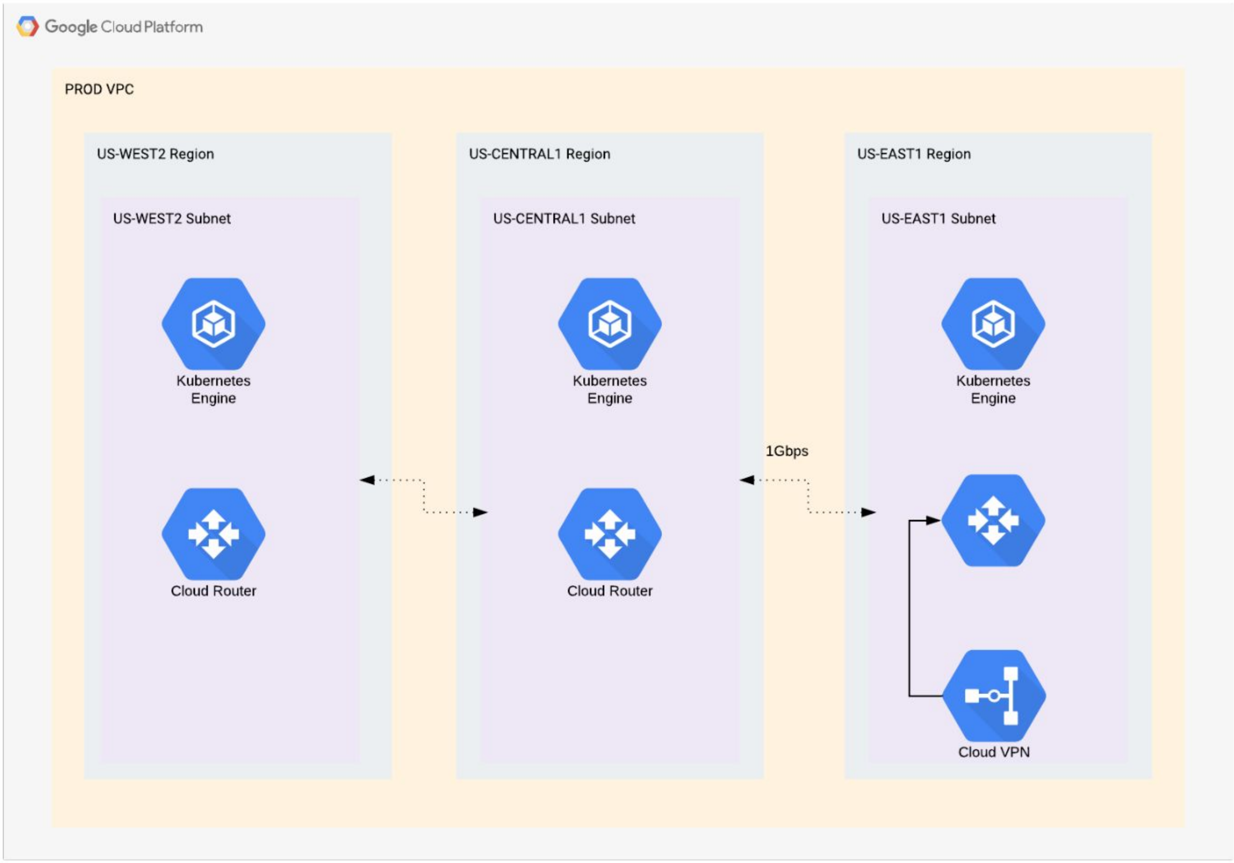

Figure 4: GCP networking with multiple VPCs/projects

Since ACME Corp is establishing a multicloud VPN in the us-east1 region, their Cloud VPN managed service needs to be deployed in the VPC tied to this region.

With this service, ACME can establish two redundant tunnels to AWS and propagate routing tables from VPCs on both ends by utilizing dynamic BGP routing between AWS and GCP. This means that they don't need to set up any static routes. And if they're utilizing BGP, all routes from the GPC side will be advertised to AWS and vice versa, eliminating the need to constantly maintain and verify static route tables.

If ACME's on-premises physical routers (such as Cisco, Juniper, Palo Alto, Check Point, Fortinet, or others) support BGP routing, they can eliminate the need for static routes across their entire network. This will vastly improve their overall network performance and ease the day-to-day activities of their network specialists, cloud architects, and DevOps engineers.

Unfortunately, GCP doesn't have an equivalent service to Amazon's Transit Gateway. In order to connect their GCP VPCs and allow all GCP internal networks to reach AWS and on-premises data centers via VPN, ACME would need to utilize GCP VPC peering.

{ AWS-GCP Multicloud Networking Key Takeaways & Best Practices }

Before I wrap up the second article in this two-part series, here are a couple of notes regarding the practical example of AWS-GCP multicloud networking we presented here:

- **Avoid network ranges overlapping**. You cannot have any duplicated segments, such as VPC or subnet CIDR, with the same address.
- **Avoid overlapping for BGP ASNs if you use dynamic routing**. You cannot have the same BGP ASN two times anywhere in your entire network.
- **Create Transit Gateway attachments for each VPC and VPN connection**. In the setup we presented, you would have five of them (three VPCs plus two VPN attachments).
- If you have multiple accounts on the AWS side, instead of one account with multiple VPCs, **you'll need to use AWS Resource Access Manager to share one Transit Gateway with multiple AWS accounts**.

Conclusion

I've covered a lot of cloud networking terms and concepts in this guide, as well as one extensive practical example on how to connect AWS and GCP with your on-premises network. Even so, if you're designing such a network, you should know that this is just the tip of the iceberg.

Before embarking on your multicloud networking journey, arm yourself with as much networking documentation from public cloud providers as you can. Consult the internet for tutorials, Reddit posts, or even third-party mailing lists and discussion boards. If it's an option and within your budget, sign up for professional support from your cloud vendor. It's expensive, but can be a life saver in scenarios like the one I've covered here.

Remember that productional networks cannot tolerate outages, and if you already have sensitive workloads in your public cloud, and you want to connect it to a data center or some other public cloud provider, you simply cannot make mistakes, because they will cost you dearly.

IAMONDEMAND

IOD

[tech.content]