Network and Application Observability for Multi-cloud Ops Teams

kentik®

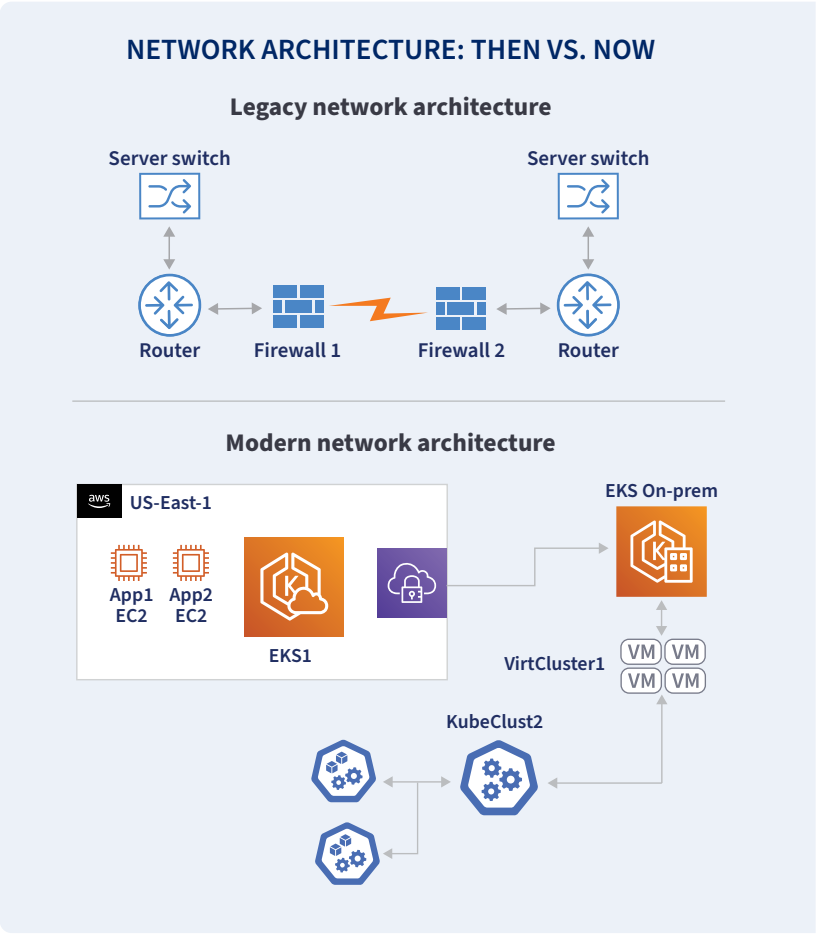
kentik

An increasingly complex future for multi-cloud ops teams

Public cloud and cloud-native application infrastructure have introduced incredible new ways to build and deploy applications. The wonders of on-demand access to a diverse range of compute, storage, and networking also add more challenges with monitoring and managing infrastructure and applications.

One of the most challenging aspects is that these new application patterns come with increasingly complex networks. Networks no longer service just rack-level or localized traffic from the application down to the host and core network.

Now you have to understand and operate complex virtual networks, service mesh architectures, hybrid, multi-region, and multi-cloud networking. This is already difficult in small to medium environments and becomes untenable as the environment and applications scale.



Multi-cloud is the new cloud

It's likely you have more than one cloud in use today. It could have come from a design choice, or maybe you acquired something that forced a multi-cloud operations challenge on you. Multi-cloud can include SaaS connectivity, hybrid data center and cloud mix, two or more major cloud providers, and sometimes managed hosting.

Multi-cloud adoption generally maps to these top reasons:

- **Best-of-breed cloud services**: Getting access to purpose-built services that are application and workload-specific (e.g., machine learning, AI, data services, specialized hardware/software, serverless offerings).
- **Acquisition**: You acquire a company, team, or tool with a footprint in a specific cloud that was not your primary cloud provider.
- **Enterprise agreement and vendor relationship**: Gaining access to discounts in one or more clouds to maximize value and minimize operational costs.

- **Technology preferences and requirements**: This can include developer preference, tooling integration, and adjacent and dependent services (e.g., identity and access management, networking, and data services).
- **Avoiding "vendor lock-in"**: Maintaining flexibility and cloud diversity to move workloads without being beholden to a single cloud platform provider.
- **Migration in progress**: Moving and porting workloads to a new provider will require multi-cloud configuration and operations during migration which rarely goes as quickly as planned.

These are all valid and important reasons why multi-cloud and hybrid cloud deployments are common. Even single cloud environments have operational challenges. Multi-region and multi-cloud infrastructure come with an unavoidable level of complexity, especially with networking, security, application performance, and cloud costs.

What you have is a multi-dimensional challenge that requires us to rethink the operational processes and tools.

The diversity of a multi-cloud platform is great for developer, application, and infrastructure flexibility. That comes at a cost, both figuratively and literally.

The multi-cloud operational complexity problem

Complexity increases as we continue to build new applications and cloud infrastructure. That's just an unfortunate fact. The diversity of a multi-cloud platform is great for developer, application, and infrastructure flexibility. However, that comes at a cost, both figuratively and literally.

The top reasons often cited for complexity in multi-cloud operations are:

- **Data gravity and data movement**: Data access and data transfer introduce latency, risk, and unexpected costs due to egress charges and other dependent services needed to protect data in transit and at rest.
- **Cloud cost management**: Each cloud provider has a complex matrix of services and costs that are very difficult to understand and control as the environment grows.
- **Security and compliance**: Diverse and distributed cloud applications introduce complexity with security and compliance.
- **Operational visibility and control**: Monitoring and controlling diverse, distributed cloud environments is complex and creates operational risk.
- **Performance monitoring and management**: Application performance in single-cloud environments is already challenging and increases exponentially in multi-cloud environments.
- **Cloud-specific tooling and APIs**: Each cloud service provider had proprietary methods and tools to configure and operate your applications, making it very difficult for both operator skill set and creating common operational processes.

Innovation continues at a furious pace everywhere, from the front-end code all the way down to the bare metal on which it runs. Those same innovations also bring new operational process challenges, especially with traditional methods and tools.

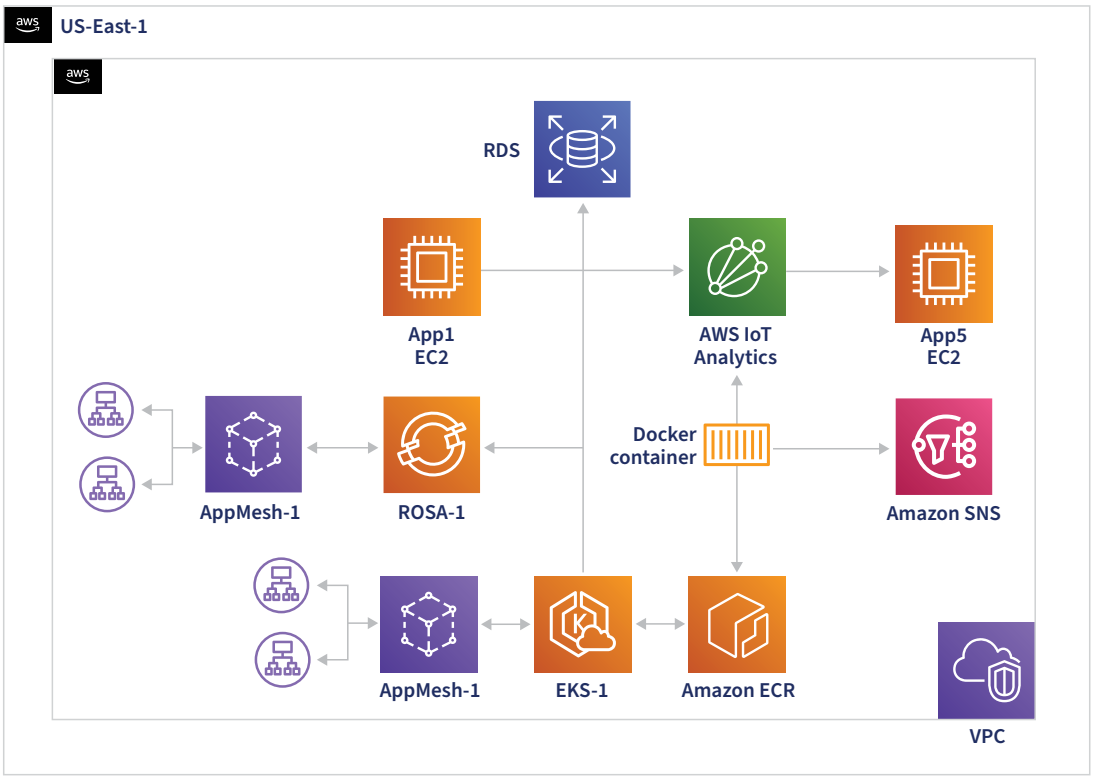
There's one more infrastructure pattern that has also extended the complexity while solving some critical application operations challenges, especially in multi-cloud deployments.

Kubernetes adds new capabilities and new complexity

One of the most widely used platforms in multi-cloud environments is Kubernetes. As a de facto standard in cloud-native application hosting, Kubernetes offers an automation-friendly, API-accessible compute platform for containerized workloads. Kubernetes also presents a standard model across your on-premises and cloud environments.

Many, if not most, new applications are being developed to take advantage of Kubernetes as a hosting platform, but that does not rid us of complexity. The simplicity you gain in application deployment comes with new complexity in monitoring and operating the applications and infrastructure.

Before, we had just IaaS workloads and data platforms with relatively simple network designs. Now we have new microservices architectures that require entirely new and dynamic network topologies. Many teams are also adding service mesh architectures to handle application layer security and networking, adding lots of capabilities but further increasing complexity.



An example of a modern service mesh architecture

Monitoring these environments is increasingly complex. It has led to a growing array of products and methods to pull analytics and instrumentation from all of the different layers to try to give operations, network, and application teams the information they need to ensure availability, resiliency, and application performance.

The challenge is that each of these monitoring tools is narrow in what it can instrument and lacks the ability to correlate diverse data points to understand total system behavior.

Observability: Going beyond monitoring

The concept of observability is to be able to infer the state of the environment based on the outputs from the overall system. Observability aims to ask questions of the system rather than just sifting through monitoring data and attempting to correlate it.

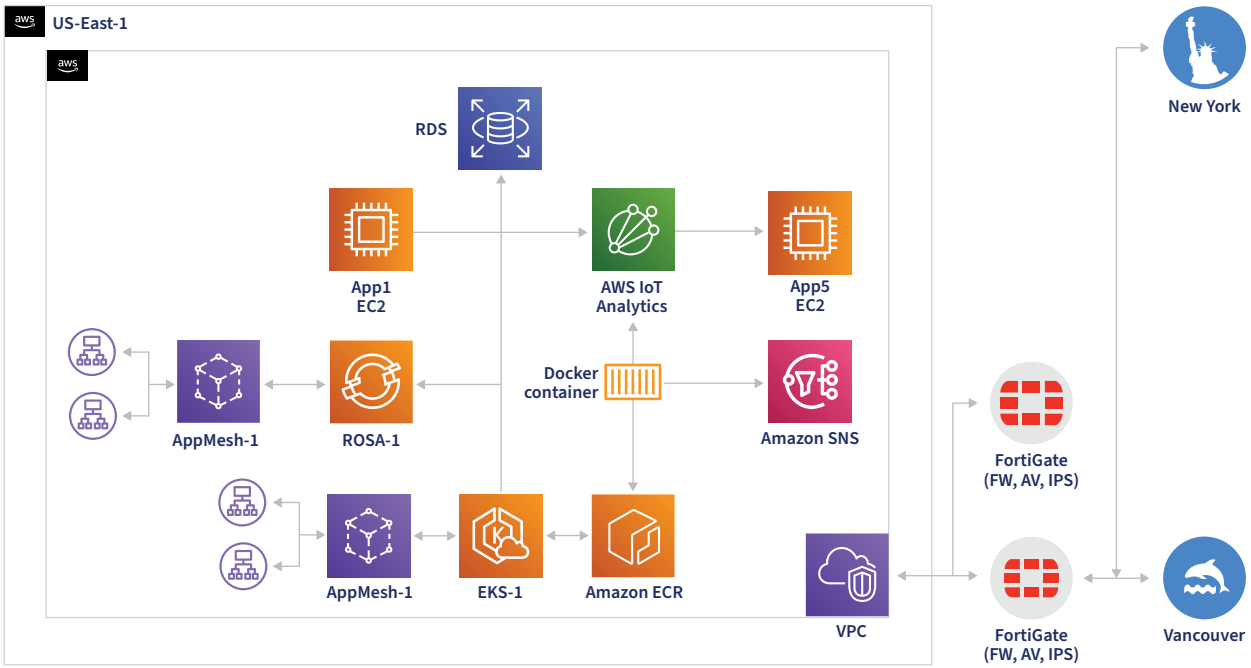
Examples of monitoring include analytics such as:

- **Infrastructure**: CPU, memory, network, and storage utilization, including capacity and performance of all these resources
- **Application**: Response time, latency, database memory utilization, garbage collection, transaction throughput
- **Network utilization**: End-to-end network path latency, route availability, and round trip time for application transactions
- **Application response time**: User experience measurements such as page load, checkout transaction time, and general client-side usability metrics

Monitoring lets us ask questions like “What is the current application response time?” whereas observability would be more about asking, “What is causing application response time to go above 25ms?”

The practice of observability requires gathering and correlating data and analytics from across the system that can inform the operator or another system about behavioral metrics rather than just utilization metrics.

Observability can also correlate availability data (e.g., sudden loss of access to a region, node, or network) so we can ask the system, “Why is the application slower from New York than Vancouver?” which may surface that a network outage is happening in an eastern region that is causing latency and application failures.

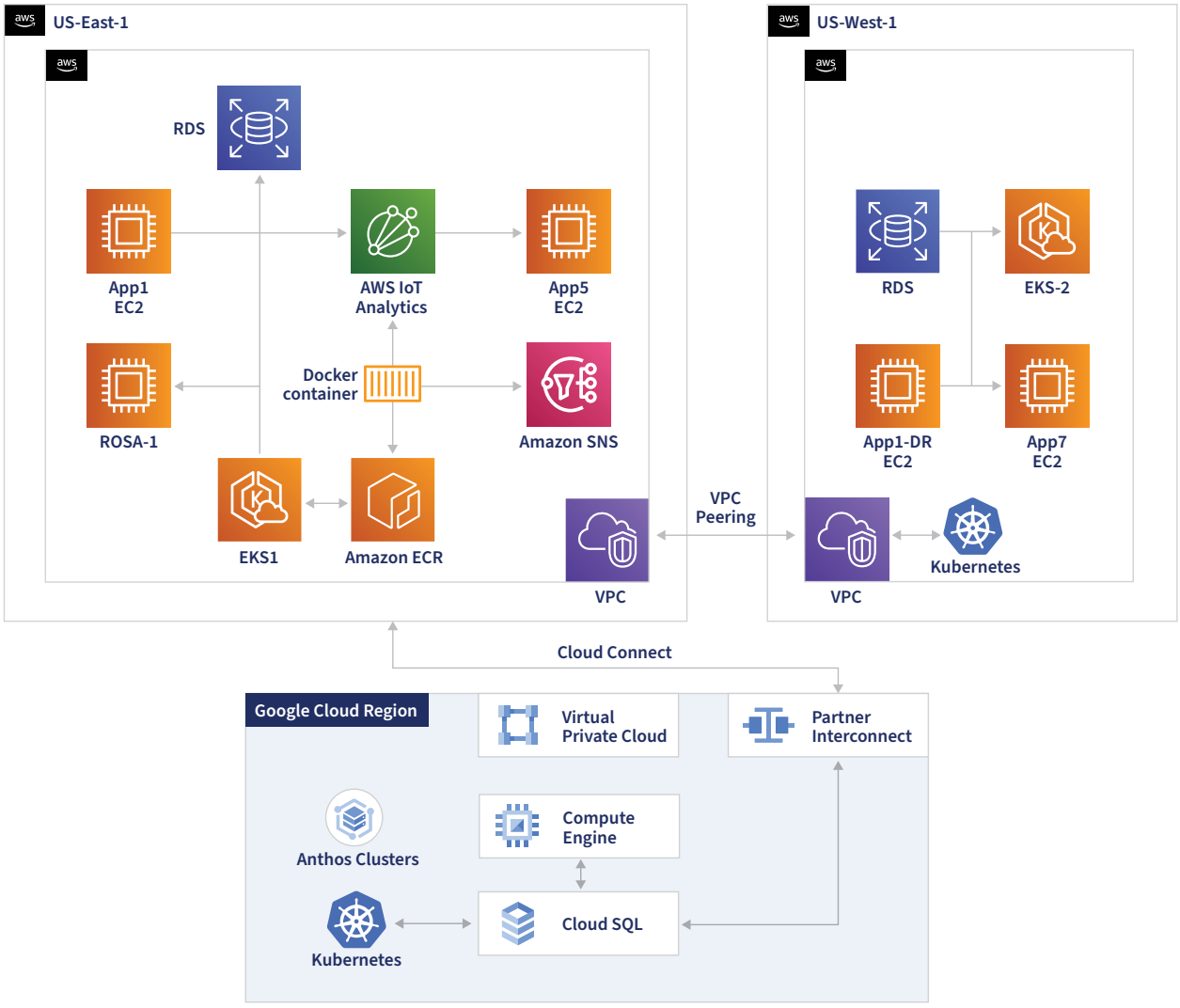


Observability is critical to the success of multi-cloud operations because your scalable applications will behave fundamentally differently from the traditional monolithic and single-platform applications we are used to.

Let’s walk through two scenarios you will likely encounter as you modernize your application with scalable and multi-cloud architectures.

USE CASE 1: From monolith to microservices to major operations issues

Your dev team has been tasked with modernizing a monolithic application, and they've chosen to turn their current 3-tier application running on public cloud IaaS into a micro-services architecture. The dev team lead is responsible for making the application available on two different cloud providers to provide resiliency and also reduce vendor lock-in.



Using Kubernetes as a hosting platform on multiple cloud providers.

They've standardized on using Kubernetes services rather than building their own container platform hosting. Using Kubernetes as a hosting platform protects the application from the platform-specific idiosyncrasies of each cloud provider.

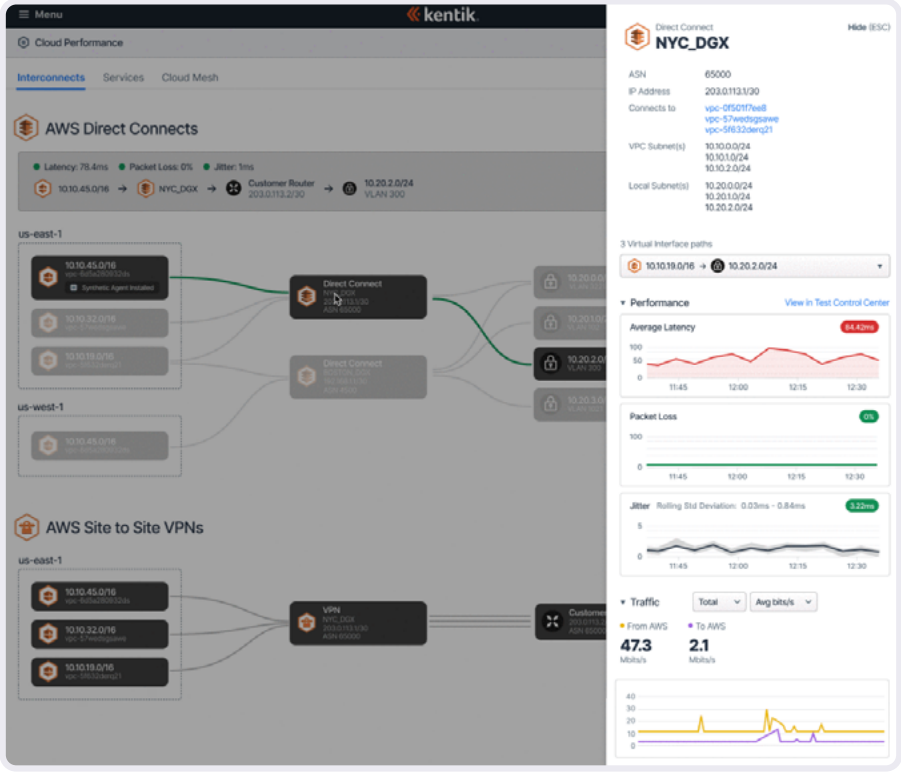
Moving from a simple application with no east-west network traffic and straightforward access control suddenly became complex in order to take advantage of the scalable hosting platform. We've added scalability and resiliency but also introduced new risks.

Now our application has new points of failure and potential risks, including:

- **Service-to-service connectivity**: Managing layer 4-7 connectivity and access control between services and traversing multiple clusters for the multi-cloud application topology.
- **Front-end-to-data connectivity**: Data access by the application now crosses network boundaries, introducing latency.
- **Distributed data requirements**: Spanning the application across regions and clouds requires data to be distributed, complicating data protection and synchronization.
- **Security model change**: Application services and data now have new attack surfaces and additional complexity to protect data in transit and at rest.

None of the issues are unavoidable. It requires changing how we choose to observe and operate the environment. We need to be able to understand the end-to-end performance and availability using practices of observability to ask questions like:

- What is causing slowness during the checkout process?
- Why are sign-ups suddenly failing in Europe?
- Why is the western US region slower than the eastern US region for client-side page loads?



Monitoring availability, resiliency, and performance in your cloud environment using Kentik.

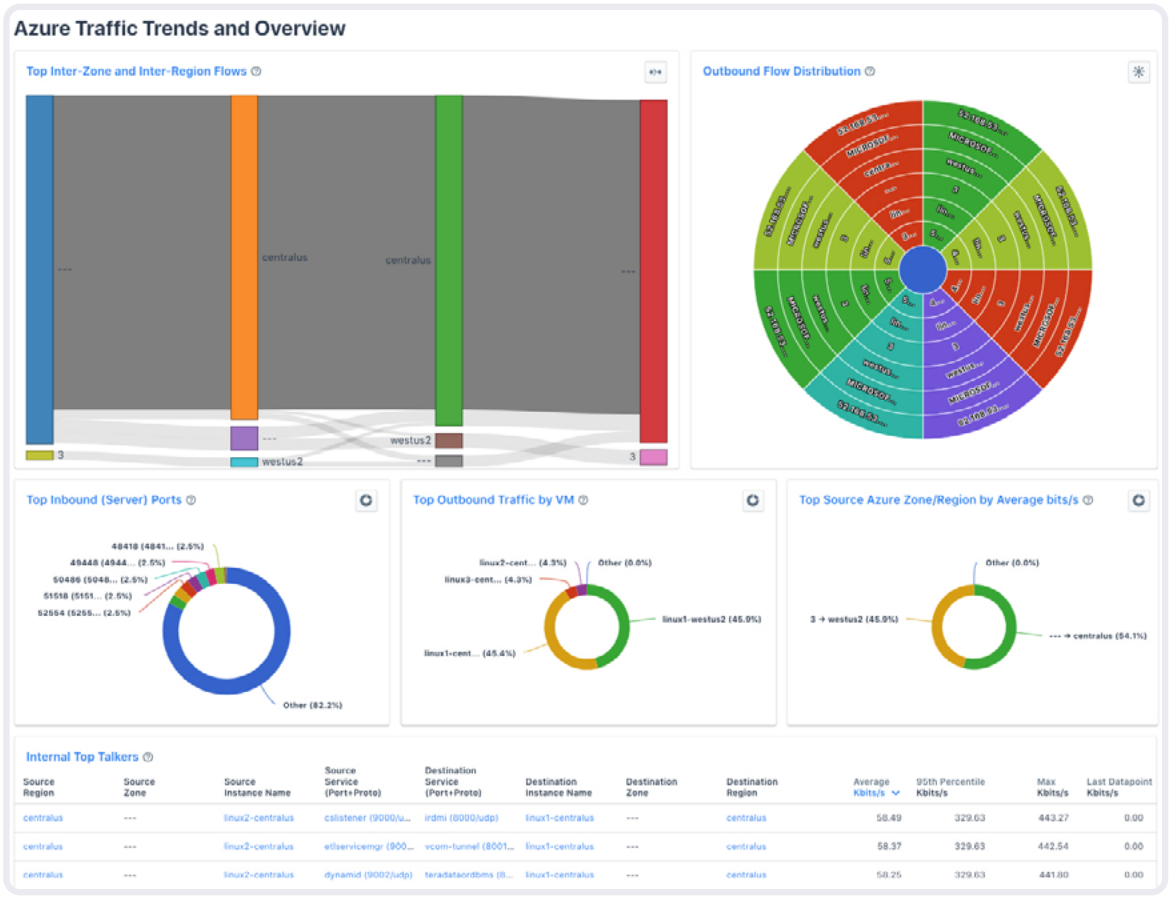
Scalable architectures do not always create linear increases in performance. You need to understand the real end-to-end path, including the application to database to container and host resources, plus the network topology, real-time availability, and performance metrics of the cloud provider itself.

Availability, resiliency, and performance each have their own success measurements, and each scale independently. This is why network observability becomes critical as we adopt and scale-out applications and multi-cloud hosting. But even in a single cloud environment, you can have region-to-region latency that could profoundly affect your application performance.

USE CASE 2: Why is my egress charge exploding this month?!

Your team has designed a scalable application architecture, and it's in response to two fundamental requirements that your organization has, including:

- **Resiliency**: Ensuring that if localized or regional outages occur, overall availability is not affected during rolling application updates or unexpected regional service outages.
- **Client-side performance**: Putting data, application endpoints, and CDN resources to use to increase performance, reduce global audiences' latency, and keep client-side application response time below a desired 25ms transaction time.

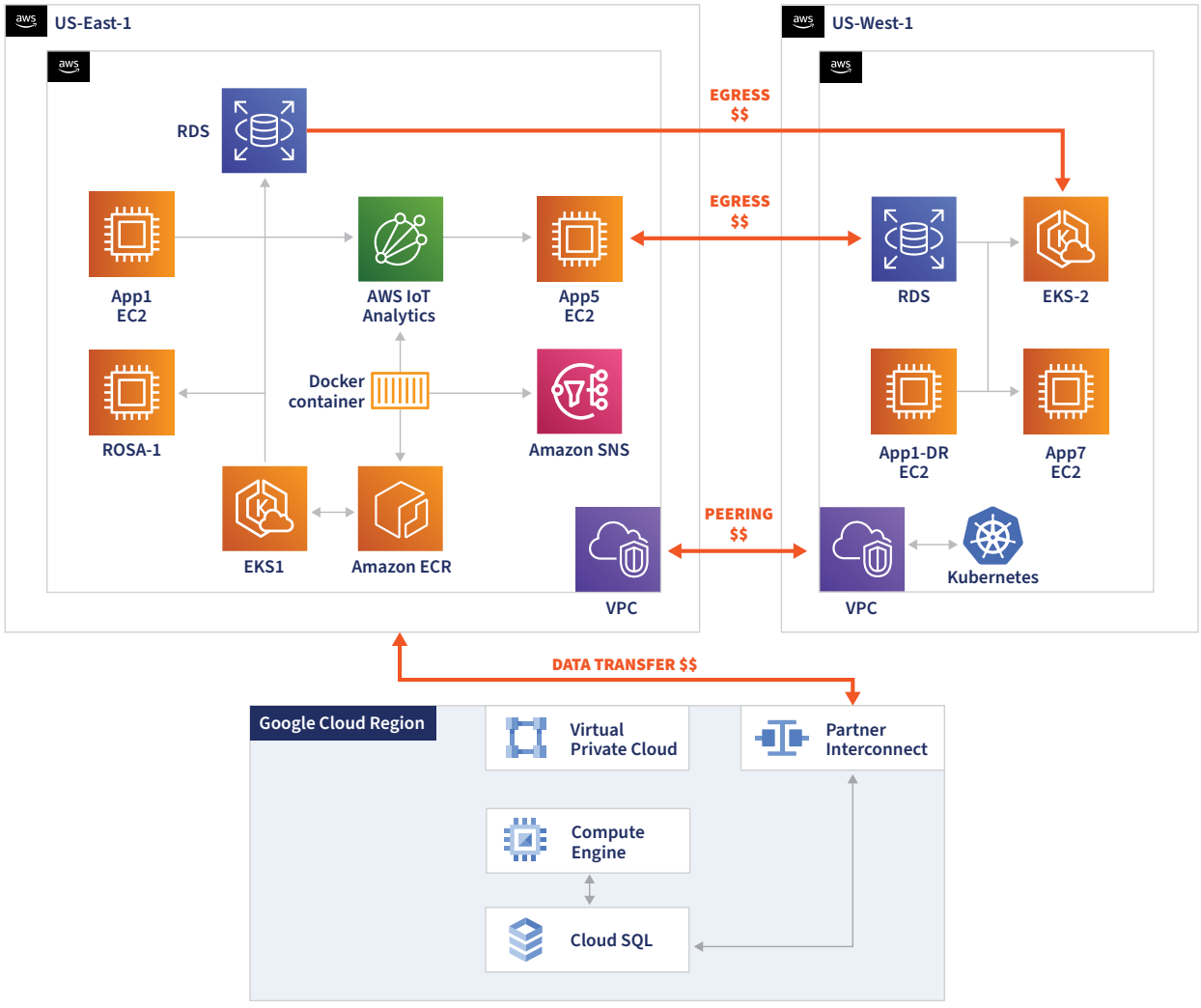


Scaling out an application across multiple regions can incur unexpected egress data costs, as shown here by Kentik.

You've scaled your application across multiple regions and even added data resiliency and availability by extending your data into a second cloud as you build out your multi-cloud application design.

But what's this? You've suddenly found some unexpected charges showing up in your cloud bill?!

These charges can surprise many cloud operators and network administrators because cloud providers have egress usage changes that can be complex to understand.



Observability insights inform the design mapping of application to fight hidden costs.

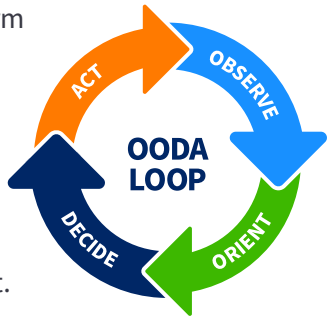
Now that we've scaled out our application across multiple regions in one cloud and are replicating data to a second cloud provider, we have unexpectedly started incurring egress data costs which we may not have accounted for.

The team has accepted that the application needs to be scalable, but we must manage costs while balancing scalability and resilience. We need the right observability for both performance and topology costs to make the right decisions about tradeoffs between resiliency and data transfer costs.

The right observability insights can inform the application owners so they can choose the right design to map to their cost profile. We can also use these cost KPIs to help network and cloud operators by mapping these costs with real-time observability.

Moving from visibility to preventative action

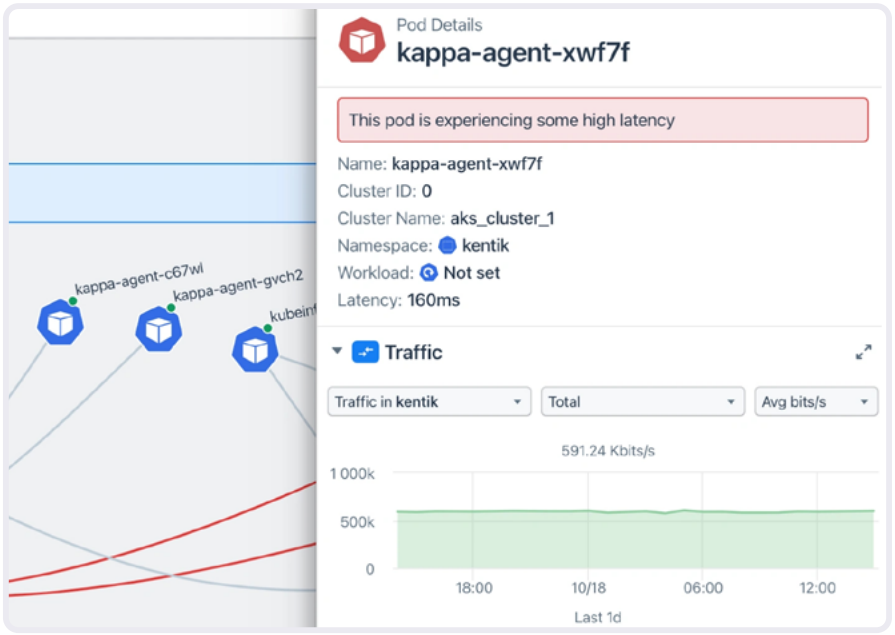
The goal of a well-architected system is to have resilient services that perform well and maintain a set of KPIs (key performance indicators) that match the desired business objectives. You've seen how important observability is in identifying behaviors and issues that affect performance and resiliency.



The next step is creating truly responsive and automation-focused systems that can be self-healing and adaptive. This is a continuous process loop similar to what is known as the OODA loop: observe, orient, decide, act.

To automate the system, we need to take observability data and generate insights that map to business objectives set as a service level objective (SLO), such as application response time or total system cost.

These insights mapped against your SLOs will generate actionable decisions that can develop systematized automation or some human intervention to ensure the right actions are taken to adjust the system to restore application health and get within the range of your SLO.



Properly monitoring latency can help to avoid SLO violations.

You can use observability to answer a question like "What is the end-to-end network path latency for my application?" which can surface that you have a particular network path in use that suddenly has very high latency.

The latency is causing your application response time to violate your SLO of 25ms for a page load transaction. You can use this data to decide whether to route application traffic through the lower latency path to bring application response time back within your SLO.

This could be an operator decision to adjust the application deployment. Your ideal environment should be self-healing and use the insight about the system to automate an application or infrastructure change (e.g., scale out/up, deploy additional network paths, flush CDN data) using any of several automation-friendly systems.

Intelligent actions must be driven by thoughtful, real-time insights derived from observability that has more than just availability as a metric.

Conclusion

Multi-cloud environments offer incredible advantages, but only with increased complexity and a set of risk tradeoffs, including design decisions, application performance impacts, and operational costs.

Using network observability is critical to understanding issues in the environment, which can be both on the cloud service provider side and the application deployment side. Scalable applications offer enormous gains in availability, resiliency, and portability. The cost tradeoff is with the added complexity of infrastructure latency and potential costs such as the hidden risk of egress and data transfer costs.

Making the right business and operational decisions requires intelligent insights that can inform people and automated processes using real-time data that maps resource metrics and application metrics to your business objectives and SLOs.

Managing network observability using an intelligent platform such as the Kentik Network Observability Cloud can empower cloud and network operations teams to plan, run, and fix any network in your multi-cloud environment.

For more information about the Kentik platform, download our hybrid cloud monitoring solution brief and book a demo today with our solutions engineers.

ABOUT KENTIK

Kentik is the network observability company. Our platform is a must-have for the network front line, whether digital business, corporate IT, or service provider. Network professionals turn to the Kentik Network Observability Cloud to plan, run, and fix any network, relying on our infinite granularity, AI-driven insights, and ridiculously fast search. Kentik makes sense of network, cloud, host, and container flow, internet routing, performance tests, and network metrics. We show network pros what they need to know about their network performance, health, and security to make their business-critical services shine. Networks power the world's most valuable companies, and those companies trust Kentik. Market leaders like IBM, Box, and Zoom rely on Kentik for network observability. Visit us at kentik.com and follow us at @kentikinc.

kentik®

Revised 20221130