GlobalLogic®
A Hitachi Group Company

Distributed Cloud Monitoring and Logging

Authored by Vladyslav Branytskyi & Volodymyr Vyshko,
CEE Technology Practice

Contents

Introduction

Distributed Cloud is a type of hybrid cloud that lets you run cloud infrastructure in multiple locations including on-premise, in cloud data centers, or within third-party data centers. You can manage everything from a single control plane.

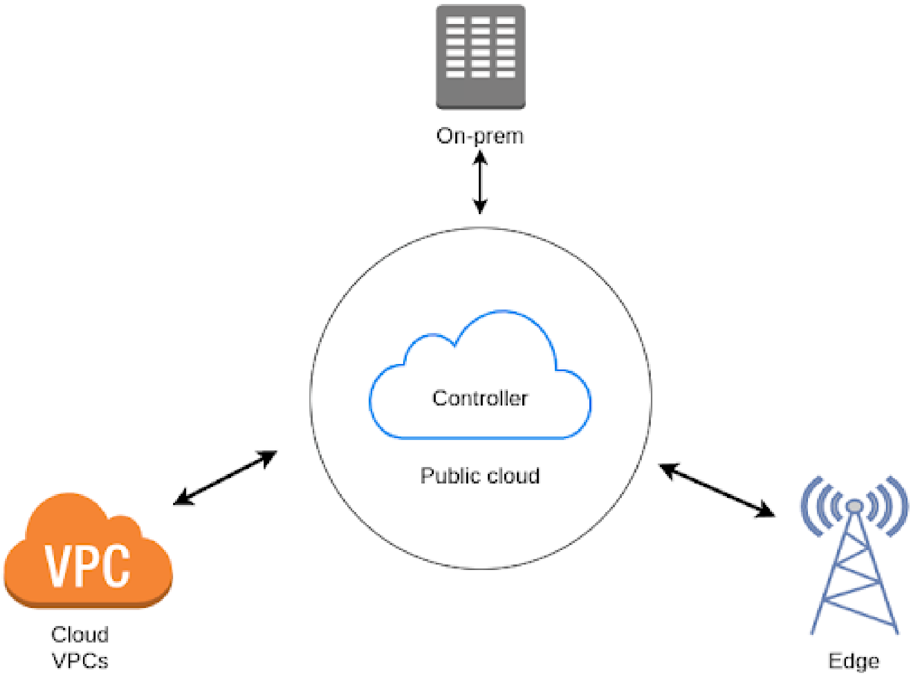


Image 1 - Distributed cloud high-level diagram

The visibility of the hybrid cloud infrastructure is one of the biggest concerns, as the combination of one or many clouds with local infrastructure can dramatically increase the complexity and risk. Organizations need deep visibility and control to handle this complexity and prevent security gaps. After all, if you cannot see it and evaluate it, you certainly cannot properly secure it.

This paper discusses monitoring and logging architectures for distributed cloud deployments, providing best practices and an overview of tools.

Logging vs. Monitoring: What's the Difference?

There are two important techniques that can help you in case of application failure when deploying an application: logging and monitoring.

Logging uses event messages to track and report the data produced by the application in a centralized way. Log events provide information about application execution state, failures and messages.

Application logging consists of the following parts:

- Log aggregation collects logs from different locations and saves them to a central location.
- Log storage involves establishing and implementing the right strategy for storing log files and controlling the deletion of the files after a retention period has ended.
- Log analysis makes sense of log data using log analyzers. It helps to mitigate risks, comply with security policies, audits, and regulations. Log analysis should not be confused with monitoring. Log analysis is post-incident work, while monitoring is permanent work.

Monitoring applies application instrumentation to logging data to provide metrics. These metrics can aggregate log data in a dashboard that provides a view of application health, memory or CPU utilization, network bandwidth, error count, performance (service response time) etc. Monitoring also makes it possible to create alerts based on needs so the system can notify us in case of failure.

Monitoring and logging are often considered to be the same. It's important to note that while they are strongly related, monitoring uses log data as the source. Unified monitoring and logging teams can gain deep insights from the application and be sure that the application remains stable, reliable, resilient and secure.

Monitoring Challenges for Distributed Cloud

Inconsistencies Due to Different Log Formats

Different clouds or on-premise environments can vary in terms of log format. If we need to monitor all environments from one point (dashboard), you must gather and interpret all log formats to "understand" and raise different events from the logs from various places. It is also important that there is a possibility to extend our Hybrid Distributed Cloud monitoring system with new places which have a new log format.

The Need to Centralize Monitoring Systems

As we know from the above text, many particular clouds included in Distributed and Hybrid Clouds have their own monitoring systems which gather and interpret the events inside themselves.

The challenge here is to put them together in one centralized monitoring system. Each custom monitoring system has its own methods for gathering information, kinds of KPIs, representations, and visualizations.

The challenge here is to interpret all of these differences between the monitoring systems and bring them into one centralized format.

Connectivity

Our Centralized Monitoring System must be able to deal with a situation where connectivity with the particular cloud or on-premise system is lost. We need to ensure that we:

- do not break the monitoring and KPI visualizations of other parts.
- restore the connection once connectivity returns.
- continue to gather information from that particular piece of the system once connectivity is restored.

Security

Communication with the monitoring system should be secure. No data should be overtaken by inappropriate persons.

Centralized Monitoring Architecture

The distributed cloud introduces some challenges to making monitoring centralized, consistent, secure, and reliable across multiple environments. There is no "silver bullet" architecture, and its structure must take into account company's constraints and requirements.

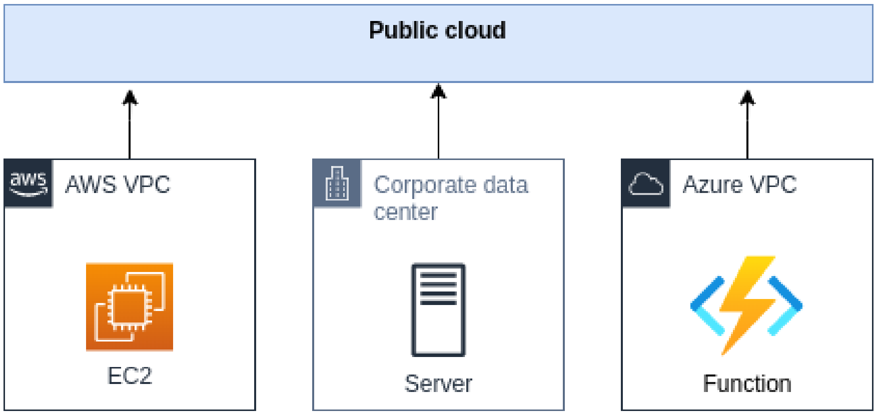


Image 2 - Distributed cloud centralized monitoring

Each part of the architecture shown in 'Image 2' may have a different approach to sending, storing, or visualizing monitoring data.

For example, the AWS component could contain a single EC2 instance with a custom Java application and an embedded cloud client for sending logs and monitoring data using public API. Or the application may be more complex and can be run in the EKS cluster with multiple Kubernetes nodes (Image 3). The Kubernetes API server exposes a number of metrics that are useful for monitoring and analysis.

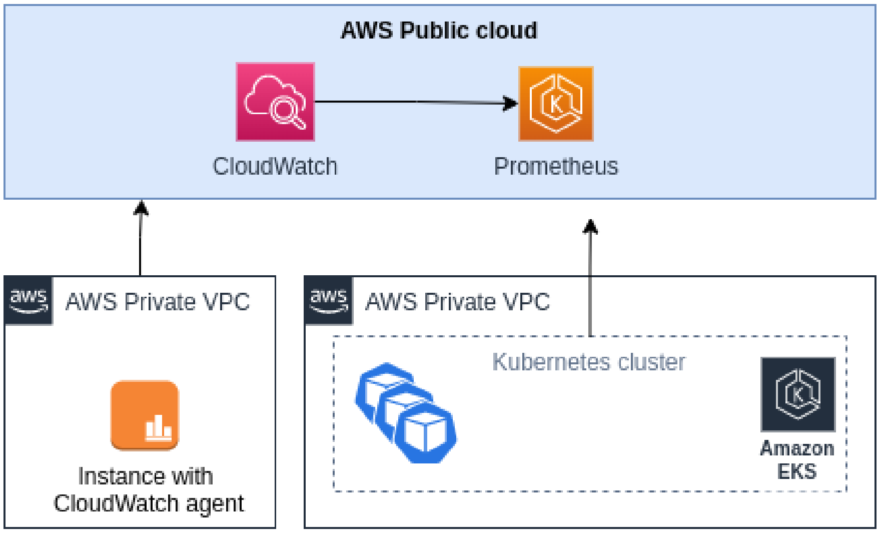


Image 3 - AWS Distributed cloud centralized monitoring

Another approach could be using a third-party management platform that acts as adapter middleware for a wide range of cloud providers.

One such management platform is Google's partner Blue Medora BindPlane. It enables users to import monitoring and logging data from popular cloud providers including AWS, GCP, IBM, and Alibaba Cloud, and from local on-premises VMs.

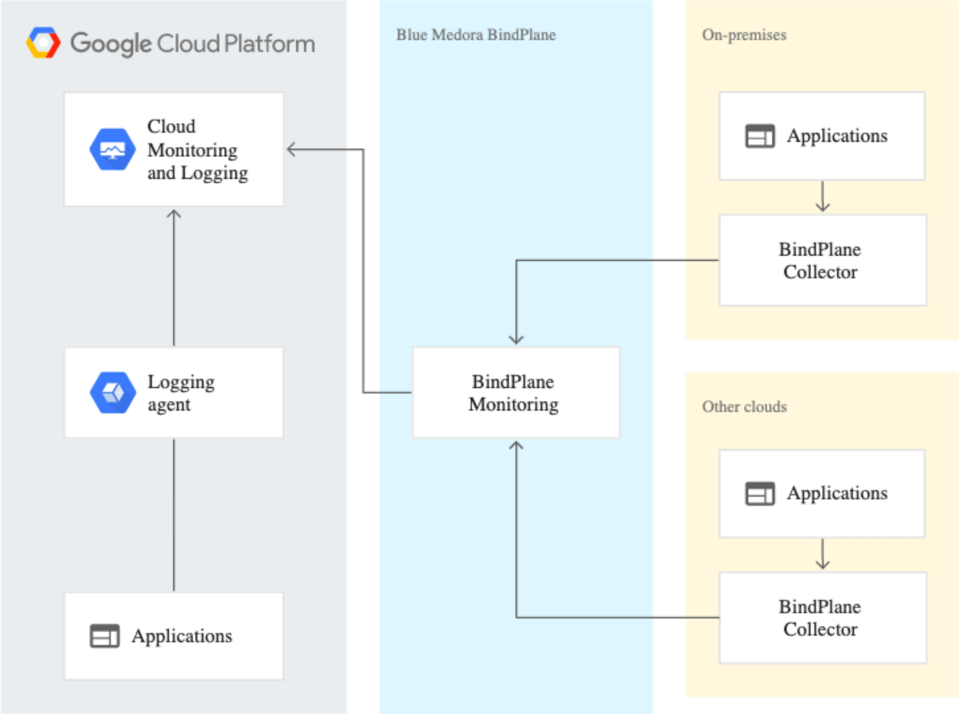


Image 4 - BindPlane monitoring and logging with Monitoring and BlueMedora BindPlane [Source]

Using such solutions gives you a wide range of features, data sources, integrated dashboards, and other things out of the box. On the other hand, there is also additional architectural complexity and cost.

In some cases, logs are important for different reasons. Some should be routed to a centralized place and others omitted. There are several solutions on the market that offer such functionality, the most popular being Fluent Bit and Fluentd. They can read logs and metrics from local files, network devices, docker containers, and more.

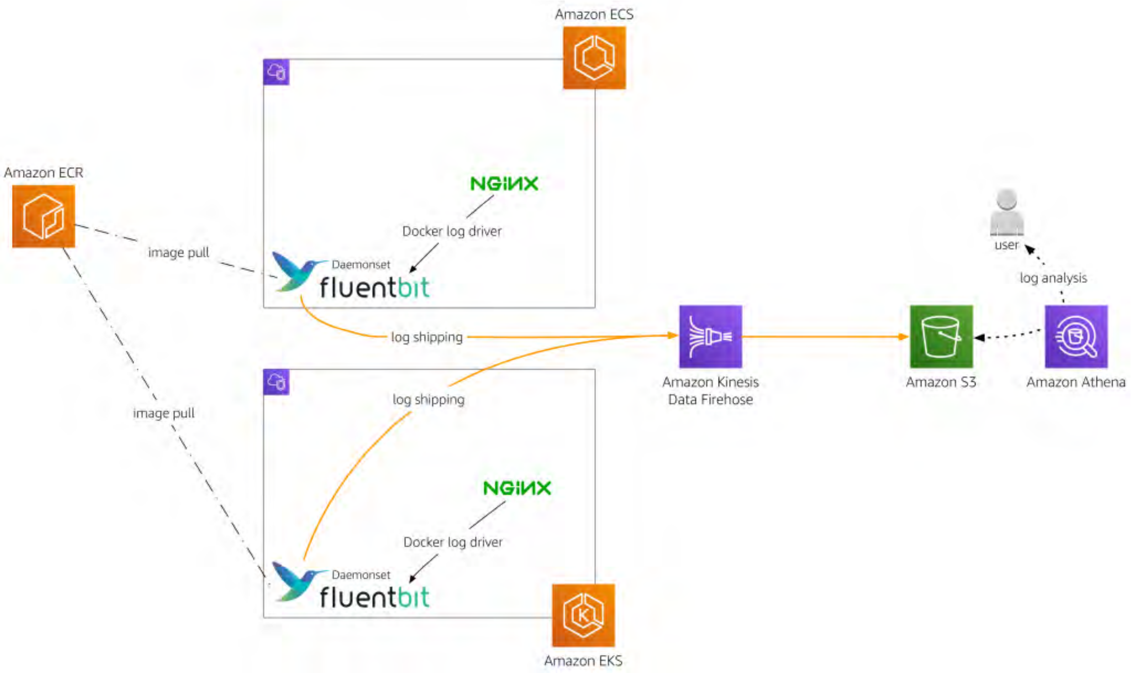


Image 5 - AWS Fluentbit logs routing [Source]

'Image 5' shows an example of the Fluent Bit in action with a hybrid EKS + ECS setup and centralized logs aggregation to the S3 bucket with Firehose ETL. These logs can be analyzed with AWS Athena later.

This architecture is a good choice if you:

- want to customize logs and remove sensitive data from there.
- have consistent Kubernetes/ECS logging across cloud and on-premises environments.
- don't want to waste additional licensing costs on third-party services.

Another approach would be introducing additional intermediate services such as Datadog, which will sit between your VPC and on-premises data centers. It ingests all of the metrics available to Logging so Datadog can function as a single pane of glass for monitoring.

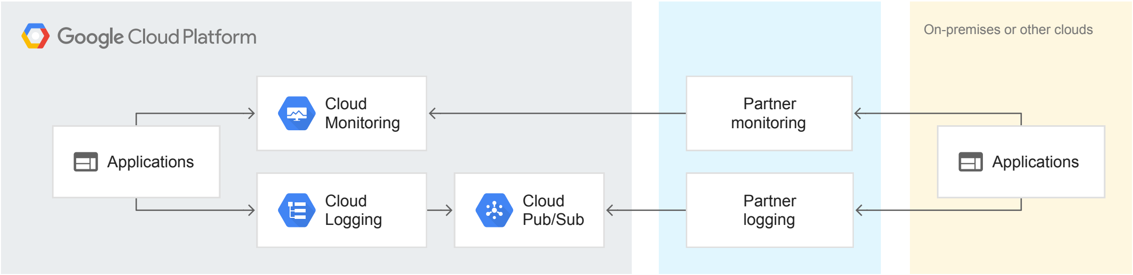


Image 6 - Exporting logs and metrics with external services [Source]

As you can see, there are many different ways to organize centralized logs and metrics aggregation. We showed only a few but will highlight more architectures and services in the next sections.

Hybrid Cloud Log Aggregation & Monitoring on the AWS Cloud

There are multiple distributed-cloud solutions available with centralized logging and monitoring features. However, sometimes we only need to gather logs and metrics to a centralized place in the public cloud without an overpay for some advanced services like AWS Outposts, Azure Arc, etc.

In this section, we will make a quick overview of possible solutions based on the AWS cloud.

CloudWatch Agent

The first thing that comes to mind when we think about logs and AWS cloud is the CloudWatch service. You can use Amazon CloudWatch Logs to monitor, store, and access log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.

But to use it together with private premises and other clouds, we need to install a CloudWatch agent on those locations.

The CloudWatch agent enables you to collect system-level metrics from on-premises servers that can include servers in a hybrid environment as well as servers not managed by AWS.

In 'Image 7', there is one possible architecture shown with a Kafka broker and CloudWatch agent as a centralized node for logs aggregation. This solution doesn't require any changes to the application source code. All logs and metrics can be gathered with Kafka or other MQ and transferred to the AWS cloud.

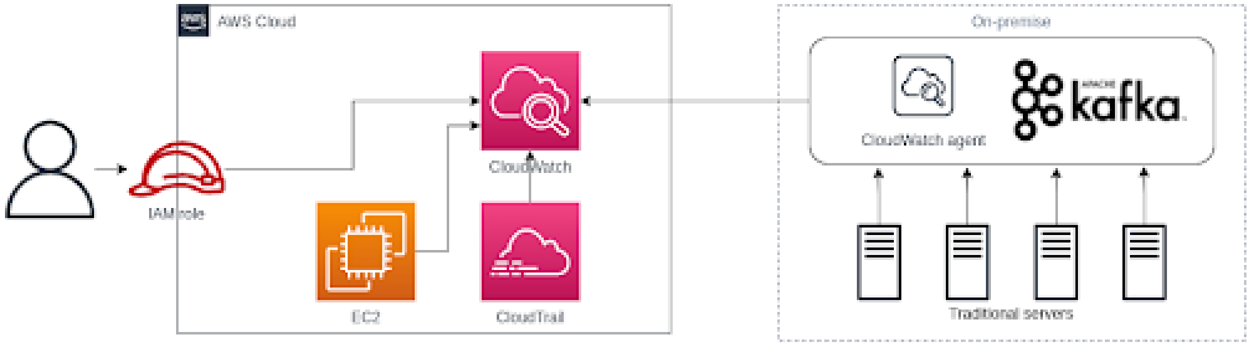


Image 7 - Distributed cloud centralized logging through the CloudWatch agent

You can store and view the metrics that you collect with the CloudWatch agent in CloudWatch just as you can with any other CloudWatch metrics.

The CloudWatch agent is open-source under the MIT license and is hosted on GitHub.

CloudWatch SDK

Sometimes it's easier to extend specific applications and use AWS CloudWatch SDK to send logs and metrics to the AWS cloud.

You may come up with this solution in several cases.

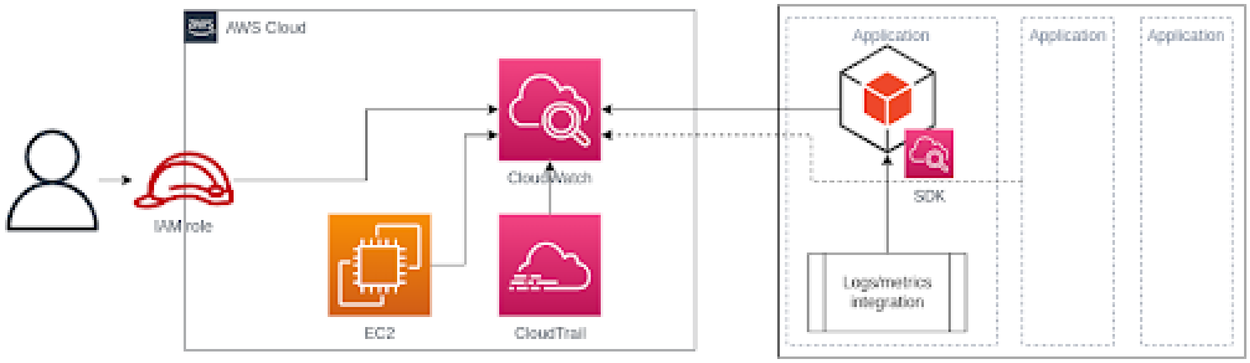


Image 8 - Distributed cloud centralized logging through CloudWatch SDK

Such a solution is not recommended for implementation on the backend applications as it has many downsides including:

- Modifying source code.
- Additional security risks.
- Problems with horizontal scaling.

However, sometimes it's just not possible to use CloudWatch Agent. In such cases, AWS SDK can help to solve the problem.

For example, you may want to perform logs aggregation for client-side applications – desktop, JS browser apps, and mobile applications. Or you may need to perform some log processing, such as the removal of sensitive data, before sending them to the cloud.

Kibana-based Solution

CloudWatch is good for small projects but as an application grows, you will have to pay more and more. Additionally, CloudWatch may not meet all your needs in operational intelligence and is better supported by Kibana.

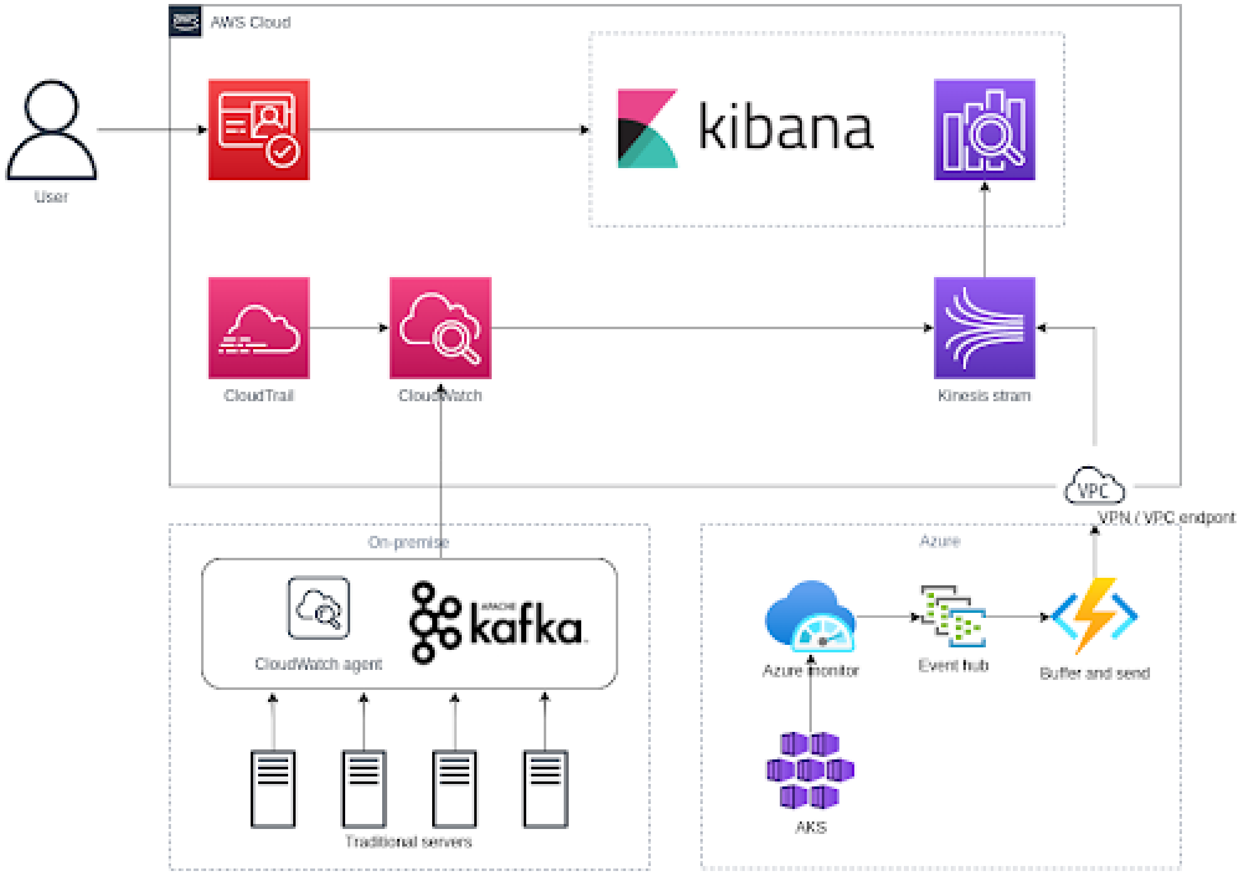


Image 9 - Distributed cloud Kibana-based centralized logging

You may use Kibana to visualize and report on a variety of KPIs; to drive business decisions for our Elastic SaaS business; and for business analytics use cases including special visualizations, indexing data, and presenting data to senior management.

The solution shown in 'Image 9' uses Amazon OpenSearch Service (managed Elasticsearch Service) and Kibana as a visualization platform. Kibana has advanced capabilities for searching and filtering logs. You can create reusable queries for filtering useful information in your specific case. It's fast and reliable as it uses Elasticsearch under the hood.

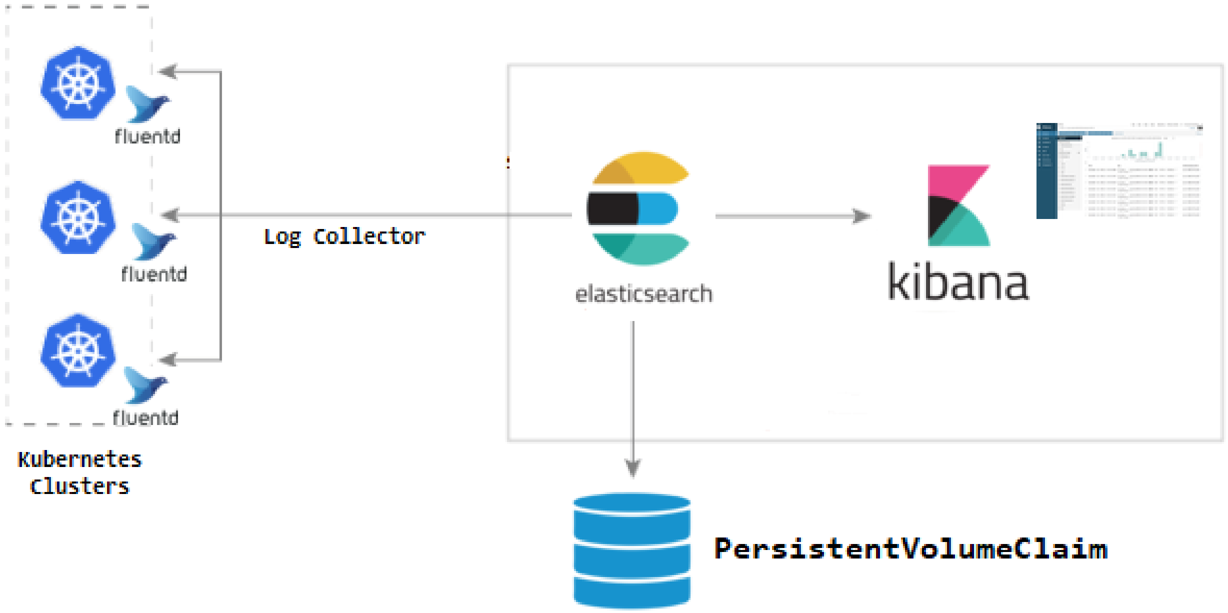


Image 10 - Kibana centralized K8S logging [Source]

Kibana can be tightly integrated with orchestrating systems like Kubernetes and even deployed to the same K8S cluster via Helm chart [link].

Generally, Kibana gives you a certain level of portability to other cloud providers in case you want to migrate from AWS to something else like Azure or GCP.

Monitoring the Distributed Kubernetes Cluster

Many cloud providers offer different solutions for centralized monitoring and logging of hybrid clouds. However, most of them are opinionated and vendor-specific.

Often, companies try to implement solutions that are as cloud-agnostic as possible by containerizing services and applying open-source container orchestrating systems.

Instead of using CloudWatch and cloud-specific solutions, companies tend to deploy monitoring and logging solutions directly to the Kubernetes cluster.

It's not always possible to run a Kubernetes cluster on a single cloud provider for multiple reasons (such as GDPR, HIPAA, etc), so some parts may be run on-premises or within other cloud providers.

In such an architecture, monitoring and visibility are crucial aspects of the stability of the entire system.

There is one possible architecture described in Image 11; a distributed Kubernetes cluster across EKS and GKE with a service mesh setup based on Istio.

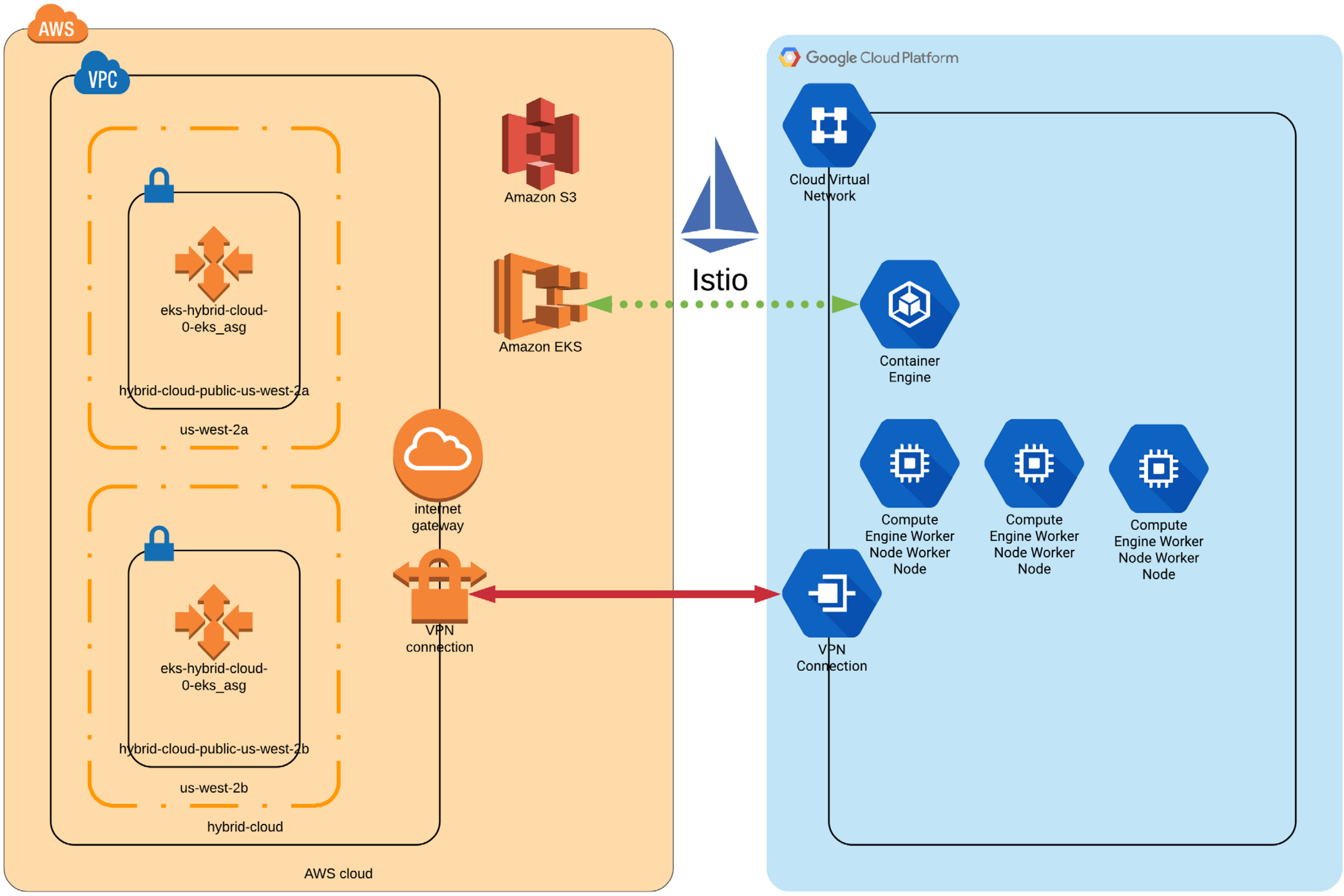

AWS
VPC
eks-hybrid-cloud-0-eks_asg
hybrid-cloud-public-us-west-2a
us-west-2a
eks-hybrid-cloud-0-eks_asg
hybrid-cloud-public-us-west-2b
us-west-2b
hybrid-cloud
AWS cloud
Amazon S3
Amazon EKS
internet gateway
VPN connection
Istio
Google Cloud Platform
Cloud Virtual Network
Container Engine
Compute Engine Worker Node Worker Node
Compute Engine Worker Node Worker Node
Compute Engine Worker Node Worker Node
VPN Connection

Image 11 - Distributed K8S monitoring with Istio [Source]

Istio provides advanced network features like load balancing, service-to-service authentication, monitoring, etc, without requiring any changes in service code.

With Istio, you gain monitoring of the traffic between microservices by default as well as the possibility to interconnect two separate K8S clusters and make them act like a single cluster.

Using Terraform (or any other IaaC tool), it's possible to set up the entire distributed environment on demand with a new VPC in each cloud, a VPN between, the Kubernetes master (EKS & GKE), and spin up the worker nodes in each cloud.

We can then install Istio as the "service mesh". This essentially allows two Kubernetes clusters to act as a single pool of resources, with telemetry and visibility across both at the same time.

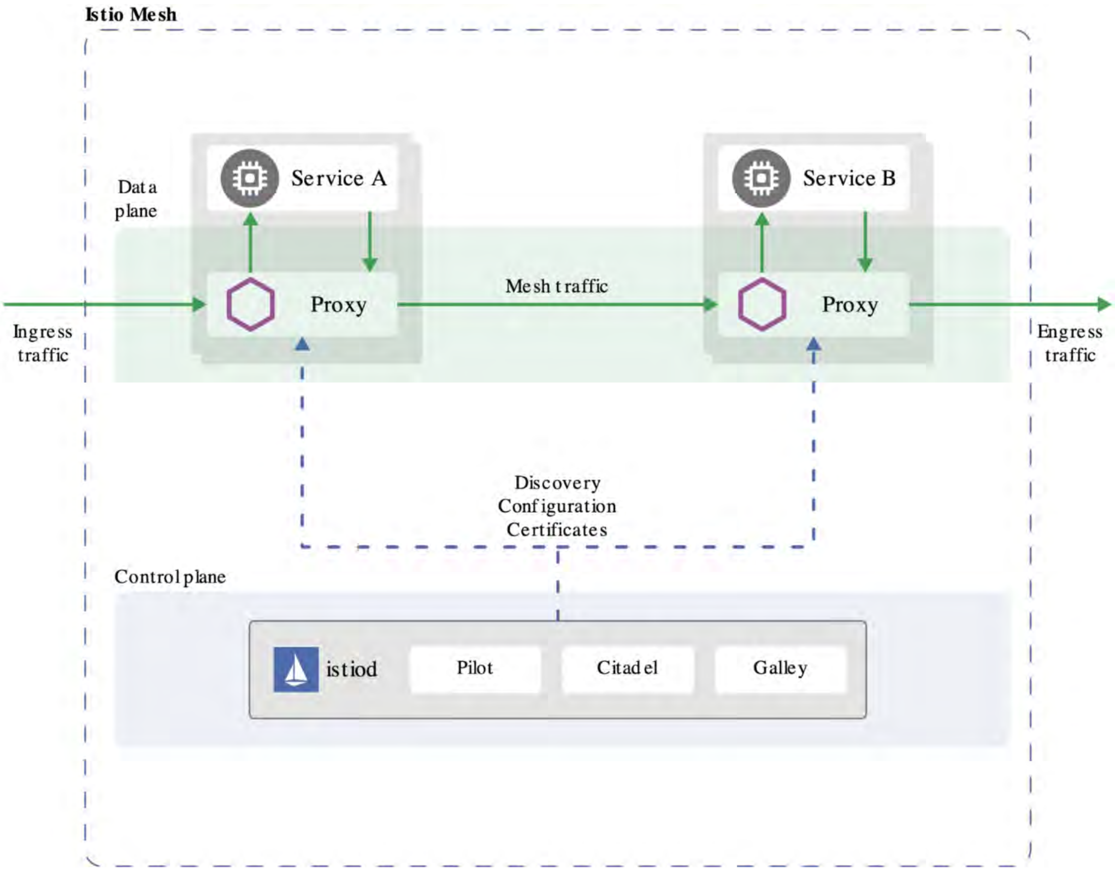
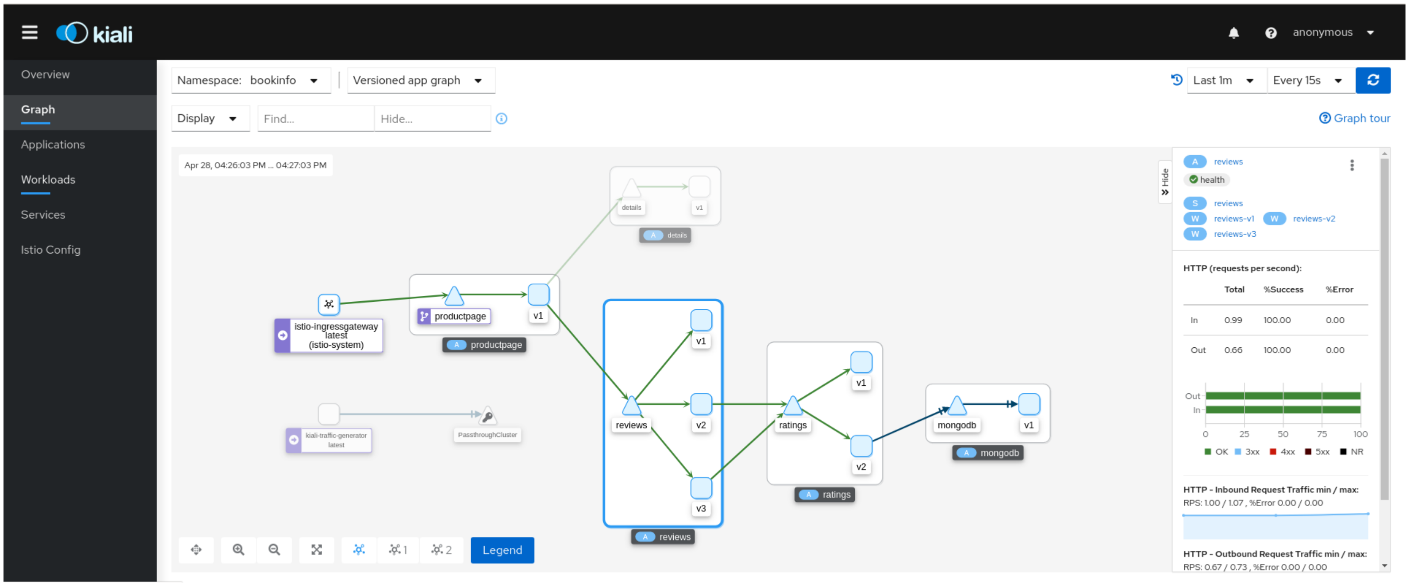


Image 12 - Istio components [Source]

Istio deploys an Envoy proxy as a sidecar container inside every pod that provides a service. This mediates every connection, and from that position, routes the incoming/outgoing traffic. In doing so, it brings a greater ability to monitor traffic.



Namespace: bookinfo

Versioned app graph

Last 1m

Every 15s

HTTP (requests per second):

	Total	%Success	%Error
In	0.99	100.00	0.00
Out	0.66	100.00	0.00

HTTP - Inbound Request Traffic min / max:
RPS: 1.00 / 1.07 , %Error 0.00 / 0.00

HTTP - Outbound Request Traffic min / max:
RPS: 0.67 / 0.73 , %Error 0.00 / 0.00

Image 13 - Istio Kiali addon and use the web-based graphical user interface [Source]

The sample source code can be found here: https://github.com/magic7s/k8s-hybrid-cloud

You can use the Istio Dashboard for monitoring your microservices in real-time using Kiali addon (Image 13). You are not limited to Istio and are therefore able to deploy any tool to the distributed Kubernetes cluster like Kibana or Grafana, as well.

Distributed Cloud Solutions Monitoring Tools: An Overview Cluster

Google Anthos

Google Anthos is a modern platform designed to give a consistent user experience across different platforms. It enables you to unify the management of infrastructure and applications across multiple cloud platforms, edges, and on-premises with the Anthos control plane. Developers can deploy applications to environments and manage them from Google Cloud.

The Anthos platform structure is complex, but we can split it into four high-level services: container management and orchestration, infrastructure management, policy enforcement, and service management.

Under the hood, Anthos uses Google Kubernetes Engine (GKE). Anthos GKE is the same Kubernetes distribution that is managed by Google Cloud. Anthos takes care of updates or security patches for GKE.

But a pure Kubernetes doesn't give you a possibility to join several clusters into one. To achieve this goal Anthos uses Google's Istio service mesh. It has multiple advantages, but most important is its flexibility and open-source nature.

It also provides a wide range of tools that help monitor and manage services that can be deployed on different cloud providers and on-prem data centers.

Anthos also brings the serverless experience to containers with Cloud Run for Anthos based on the open-source Knative, originally created by Google. Anthos Cloud Run is compatible with Anthos Service Mesh and provides built-in load balancing capabilities, autoscaling and monitoring.

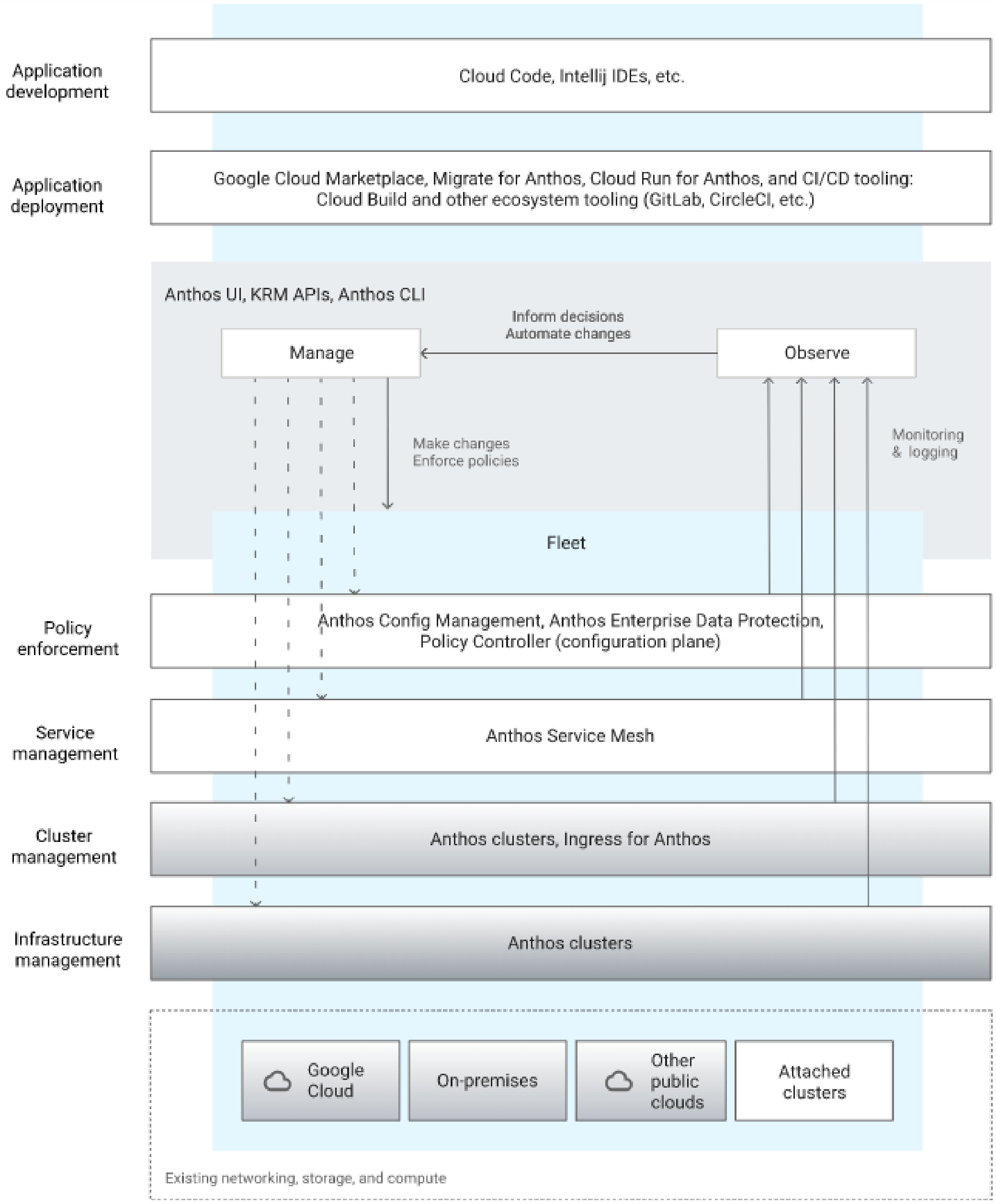

Application development
Cloud Code, Intellij IDEs, etc.
Application deployment
Google Cloud Marketplace, Migrate for Anthos, Cloud Run for Anthos, and CI/CD tooling: Cloud Build and other ecosystem tooling (GitLab, CircleCI, etc.)
Anthos UI, KRM APIs, Anthos CLI
Inform decisions
Automate changes
Manage
Observe
Make changes
Enforce policies
Monitoring & logging
Fleet
Policy enforcement
Anthos Config Management, Anthos Enterprise Data Protection, Policy Controller (configuration plane)
Service management
Anthos Service Mesh
Cluster management
Anthos clusters, Ingress for Anthos
Infrastructure management
Anthos clusters
Google Cloud
On-premises
Other public clouds
Attached clusters
Existing networking, storage, and compute

Image 14 - Anthos components [Source]

Logging and monitoring in Anthos is one of the key benefits for a distributed cloud, as we can manage Anthos with the same tools we use to manage applications in Google Cloud. Cloud Logging provides a unified place to store and analyze logs. Logs generated by the cluster are automatically sent to Cloud Logging. It works out-of-the-box, you don't need to waste additional time to set up metric collection dashboards for hybrid or multi-cloud environments.

Enabled by default, Kubernetes Engine Monitoring provides an integration that stores the application's critical metrics.

Taking advantage of these instruments, Anthos allows us to use these logs and metrics in debugging, alerting, and post-incident analysis.

Anthos Service Mash and Anthos Cloud Run service-level logs and metrics (latency, errors, requests per second etc.) are automatically collected and sent to Cloud Monitoring and Cloud Logging without any additional configuration.

Azure Stack and Azure Arc

Azure Stack is a hardware solution and an extension of Azure to run hybrid applications across on-premises (Azure Stack Hub and Azure Stack HCI) and edge (Azure Stack Edge).

Azure Stack HCI provides a hyper-converged infrastructure with servers with software-defined compute, storage, and networking. Azure Stack HCI cluster is Azure Arc-enabled out-of-the-box that provides deep integration with Azure, management tools, and built-in security. You can easily deploy a Kubernetes cluster, consistent with AKS.

Azure Stack Edge allows you to move the compute and storage to the edge.

Azure Arc is a software solution that brings Azure services and management to any infrastructure: on-premises, edge, and multi-cloud (Azure, AWS and Google Cloud).

Azure Arc allows you to use Azure Resource Manager and manage virtual machines, any CNCF-conformant Kubernetes clusters, and databases. However, unlike Google Anthos, Azure Arc is not tied to the infrastructure of Kubernetes.

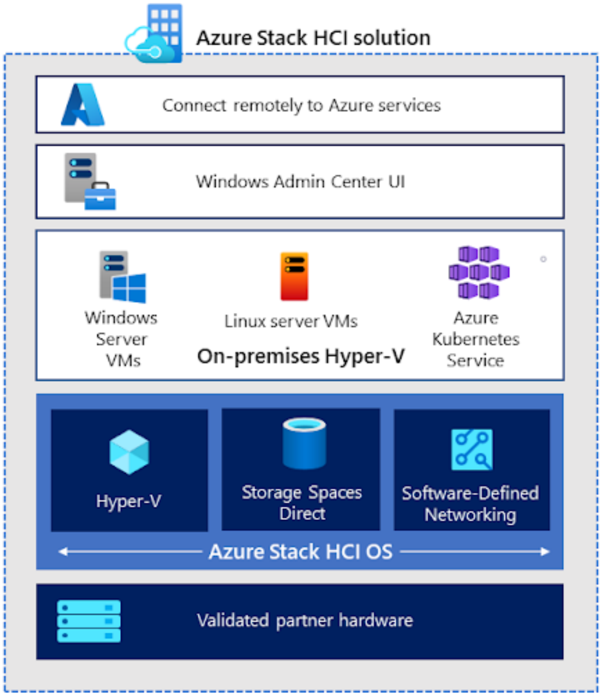


Image 15 - Azure Stack HCI solution [Source]

Azure Acr-enabled services such as Azure App Service, Azure Functions, Azure Logic Apps, Azure Event Grid, and Azure API Management allow you to run web applications, APIs, serverless applications, event-based applications and automated workflows.

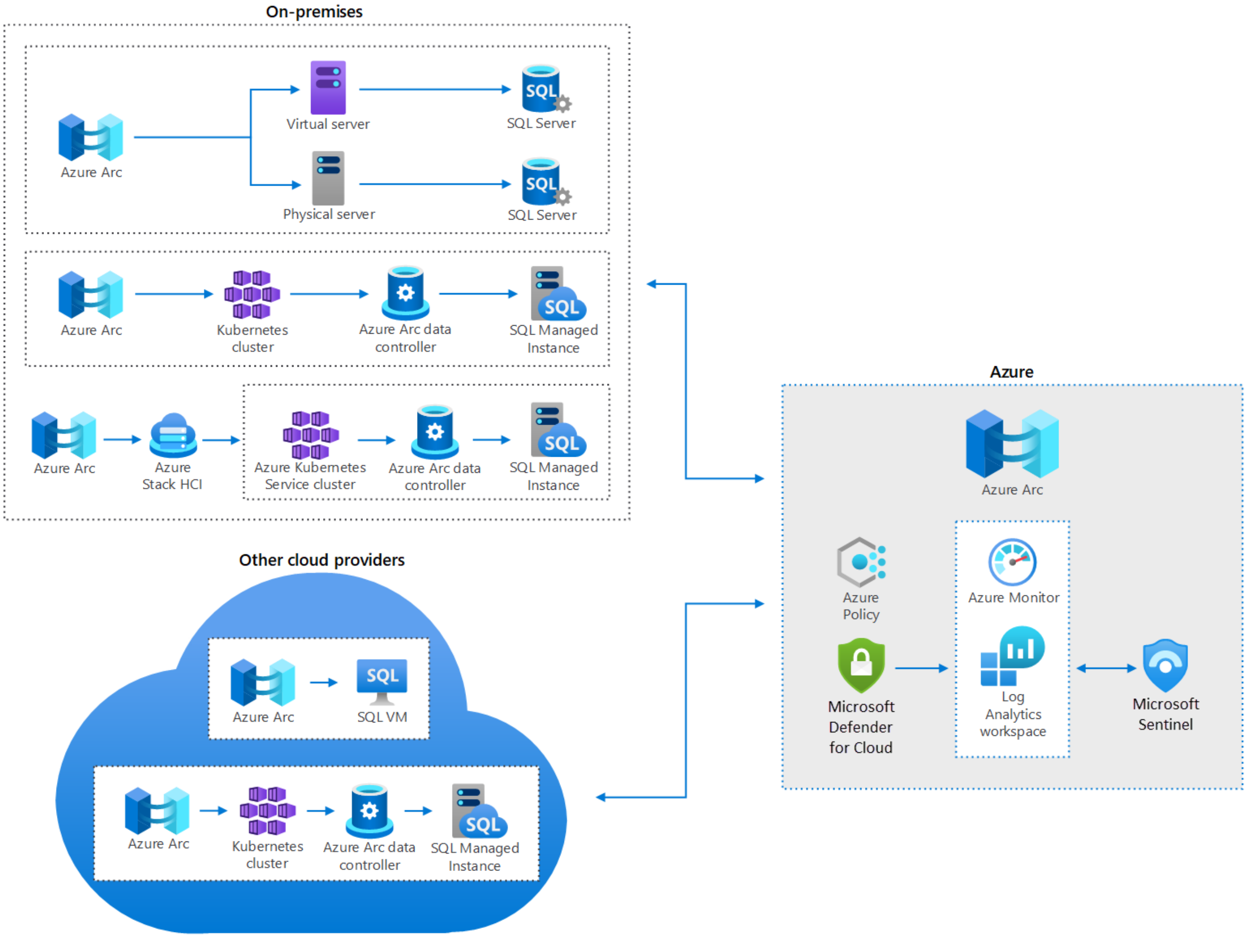


Image 16 - SQL Server instances optimization by using Azure Arc [Source]

As with Google Anthos, Azure Arc provides out-of-the-box logging and monitoring tools. The Log Analytic agent collects the log data such as performance data and events and stores them in a Log Analytics workspace. Azure Monitor uses VM insights to monitor application processes and dependencies with other resources.

AWS Outposts

AWS Outposts is a hardware solution with the same AWS infrastructure for on-premises data center environments. AWS Outposts include multiple AWS services:

1. Relational Database (RDS)
2. Amazon Elastic Compute Cloud (EC2)
3. Amazon Elastic Kubernetes Services (EKS)
4. Amazon S3
5. AWS App Mesh Envoy proxy
6. Amazon Elasticache
7. Elastic MapReduce (EMR)
8. Application Load Balancer (ALB)
9. AWS VPC

Despite this, you cannot run many AWS services on an Outpost. They will be available remotely.

To get started with AWS Outposts, you need to order from a range of fully integrated and pre-validated Outpost configurations. AWS certified personnel deliver and install the Outpost rack and connect it with the local network. You can then run your applications on Outpost using the AWS console, CLI or SDK.

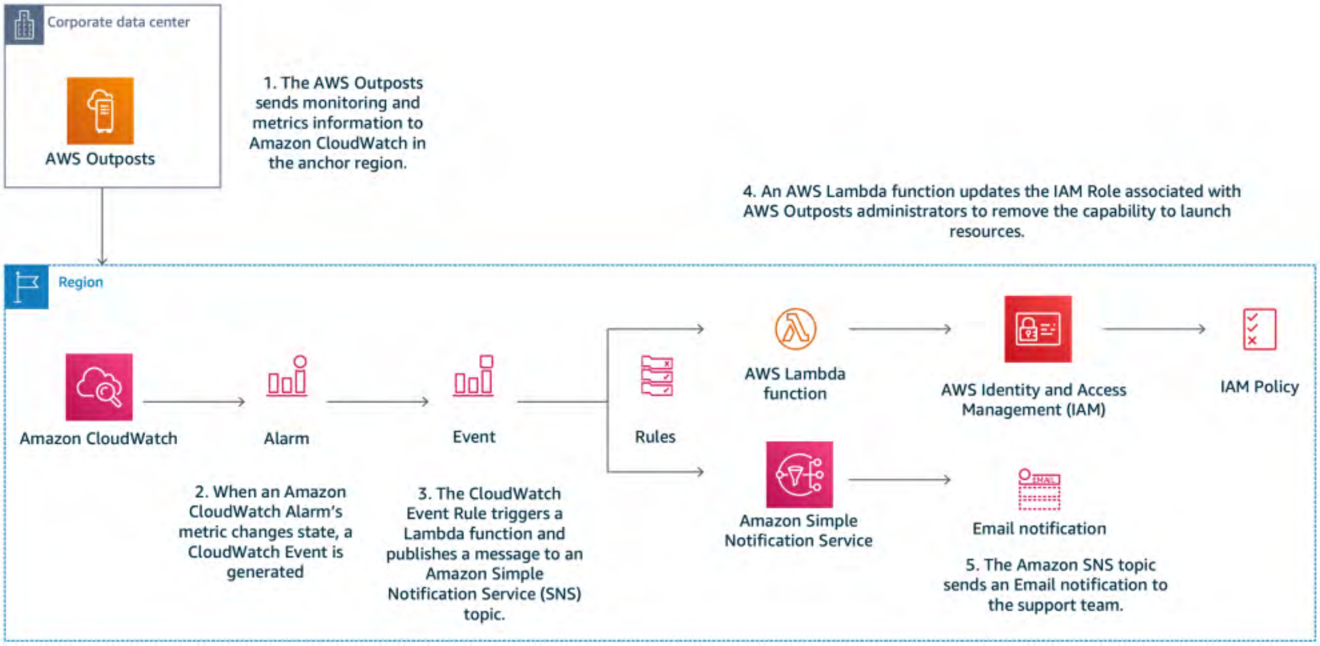

Corporate data center
AWS Outposts
1. The AWS Outposts sends monitoring and metrics information to Amazon CloudWatch in the anchor region.
4. An AWS Lambda function updates the IAM Role associated with AWS Outposts administrators to remove the capability to launch resources.
Region
Amazon CloudWatch
Alarm
Event
Rules
AWS Lambda function
AWS Identity and Access Management (IAM)
IAM Policy
Amazon Simple Notification Service
Email notification
2. When an Amazon CloudWatch Alarm's metric changes state, a CloudWatch Event is generated
3. The CloudWatch Event Rule triggers a Lambda function and publishes a message to an Amazon Simple Notification Service (SNS) topic.
5. The Amazon SNS topic sends an Email notification to the support team.

Image 17 - AWS Outposts monitoring [Source]

AWS Outposts are integrated with services that offer monitoring and logging capabilities, so you can view all logs in AWS Management Console.

AWS CloudWatch retrieves statistics about data points as an ordered set of time series data, known as metrics. AWS Outposts automatically publishes these metrics to AWS CloudWatch.

You can monitor any metrics from your Outpost over a specific period of time. AWS CloudTrail collects detailed information about API calls like IP address, a time when the call was made, and who made the call. VPC Flow Logs collect detailed information about the traffic going to and from your Outpost and within your Outpost.

Summary

The visibility of the distributed cloud infrastructure is an essential aspect of the infrastructure. Even the smallest downtime period for the specific service of the application may have a huge impact on the company's public image.

There are multiple ways to bring visibility to a distributed cloud infrastructure. You can use software given by the cloud provider (like CloudWatch agent). You can use cloud-agnostic tools (Kibana, ElasticSearch, Grafana) deployed to a dockerized environment with a container orchestrator (Kubernetes) and service mesh. And of course, you have a choice of multiple vendor-specific solutions with preconfigured services for monitoring and logging.

Microsoft Azure, Google Cloud, and AWS provide excellent solutions for the distributed cloud with built-in tools for deployment, management, logging and monitoring. These solutions make managing your distributed environment much easier.

Microsoft Azure provides both hardware and software solutions (Azure Stack and Azure Arc), while AWS provides hardware (Outposts) and Google Cloud provides software (Google Anthos). The advantage in Azure Arc is that, unlike with Google Anthos, it is not completely tied to the Kubernetes infrastructure. You can deploy virtual machines and Kubernetes clusters to your own environment.

Logging and monitoring are implemented in all platforms in a similar way. Logs are collected automatically and managed by built-in services such as Amazon CloudWatch Logs, Cloud Logging and Azure Log Analytic. Built-in monitoring services enable you to monitor your infrastructure and applications without any additional setup.

Links and References

Google Cloud, Hybrid and multi-cloud monitoring and logging patterns, accessed at
https://cloud.google.com/architecture/hybrid-and-multi-cloud-monitoring-and-logging-patterns,
May 2022.

MetricFire, Sample Approaches of Hybrid Cloud Monitoring Models, accessed at
https://www.metricfire.com/blog/examples-of-hybrid-cloud-deployment-models/,
May 2022.

Microsoft, Unified logging for microservices applications, accessed at
https://docs.microsoft.com/en-us/azure/architecture/example-scenario/logging/unified-logging,
May 2022.

Google Cloud, Anthos technical overview, accessed at
https://cloud.google.com/anthos/docs/concepts/overview,
May 2022.

Google Cloud, Hands-on with Anthos on bare metal, accessed at
https://cloud.google.com/blog/topics/developers-practitioners/hands-anthos-bare-metal,
May 2022.

Google Cloud, Anthos under the hood, accessed at
https://inthecloud.withgoogle.com/content-anthos/whitepaper_anthos_under_the_hood_2020.pdf,
May 2022.

Microsoft Azure, Build cloud-native applications that run anywhere, accessed at
https://azure.microsoft.com/en-gb/blog/build-cloudnative-applications-that-run-anywhere/,
May 2022.

Amazon AWS Blog, Managing your AWS Outposts capacity using Amazon CloudWatch and AWS Lambda, accessed at
https://aws.amazon.com/blogs/compute/managing-your-aws-outposts-capacity-using-amazon-cloudwatch-and-aws-lambda/,
May 2022.

Google Cloud, What is Istio? accessed at
https://cloud.google.com/learn/what-is-istio,
May 2022.

Istio, Visualizing Your Mesh, accessed at
https://istio.io/latest/docs/tasks/observability/kiali/,
May 2022.

Sysdig, How to monitor Istio, the Kubernetes service mesh, accessed at
https://sysdig.com/blog/monitor-istio/,
May 2022.

Github, magic7s/k8s-hybrid-cloud, accessed at
https://github.com/magic7s/k8s-hybrid-cloud,
May 2022.

ITNext.io, Building a Kubernetes Hybrid Cloud with Terraform, accessed at
https://itnext.io/building-a-kubernetes-hybrid-cloud-with-terraform-fe15164b35fb,
May 2022.

About the Authors

Vladyslav Branytskyi

I'm a software engineer with experience in Computer Science, Deep Learning and Big Data and a PhD candidate in Artificial Intelligence. I'm passionate about learning new things and innovative technologies.

Volodymyr Vyshko

Has been working in IT for about 10 years. During this time, managed to play different roles: java backend dev, js frontend, node.js backend, scrum master, lead, cloud architect. Trainer of different courses including "Algorithms and data structures" and "Distributed systems design."

GlobalLogic®
A Hitachi Group Company