Microsoft

WorkshopPLUS - Containers as Infrastructure

Containers Solutions Monitoring

Microsoft Services

Objectives

- Learn about Azure Monitor
- Understanding Azure Monitor Agents
- Learning to use Azure Monitor Search, Alerts and related services
- Using Azure Monitor with Containers, Kubernetes and Service Fabric
- Learning Kubernetes Security Best Practices

Microsoft

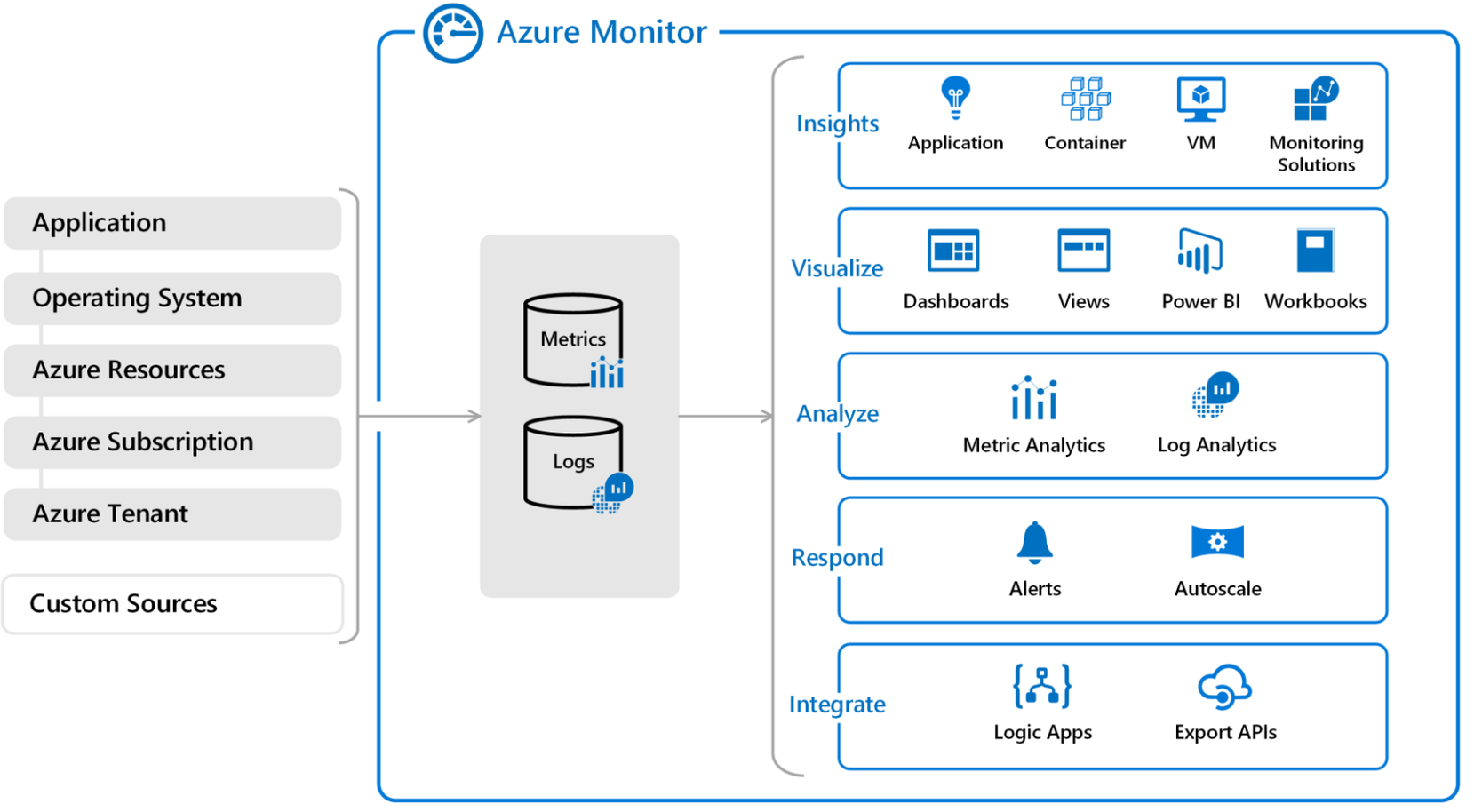
Containers Solutions Monitoring

Azure Monitoring High Level View

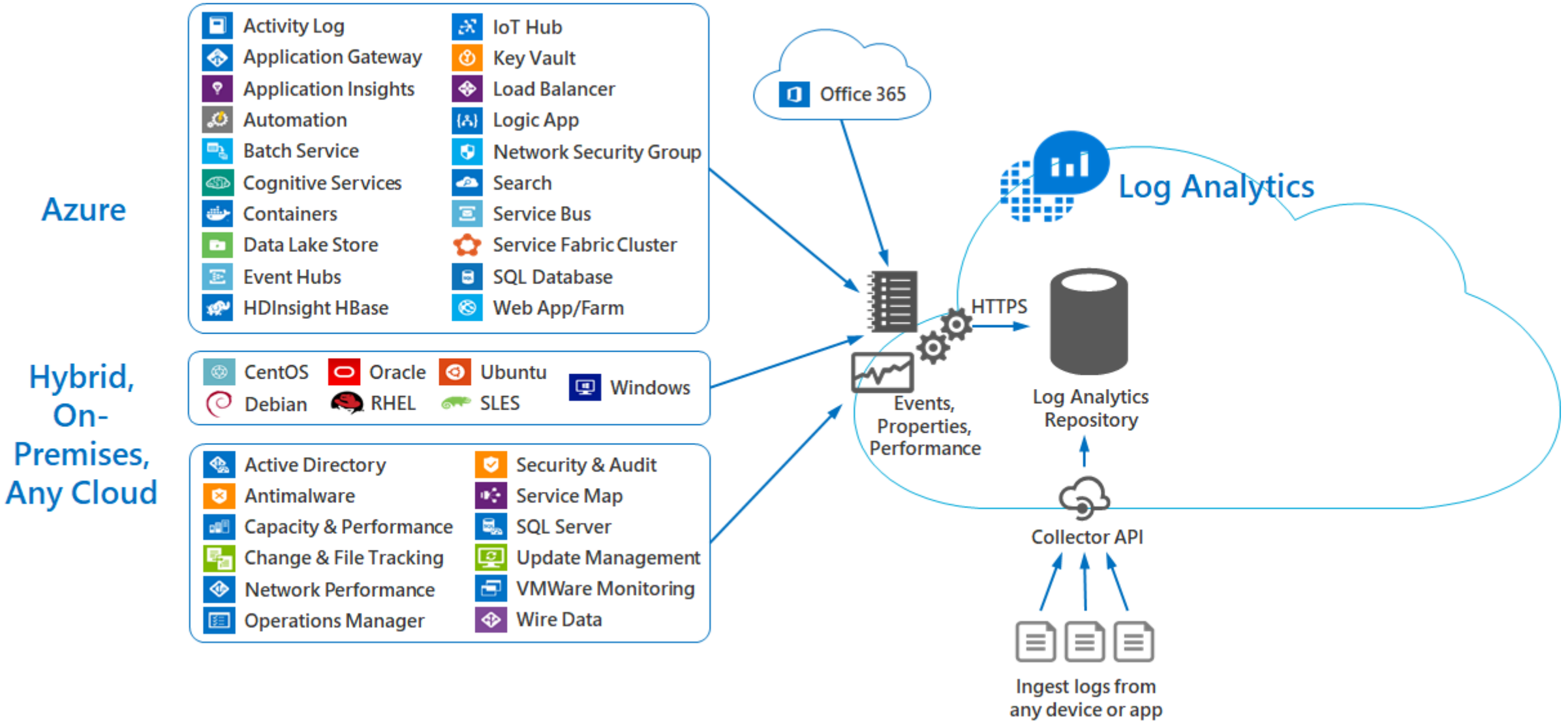
Microsoft Services

Azure Monitor

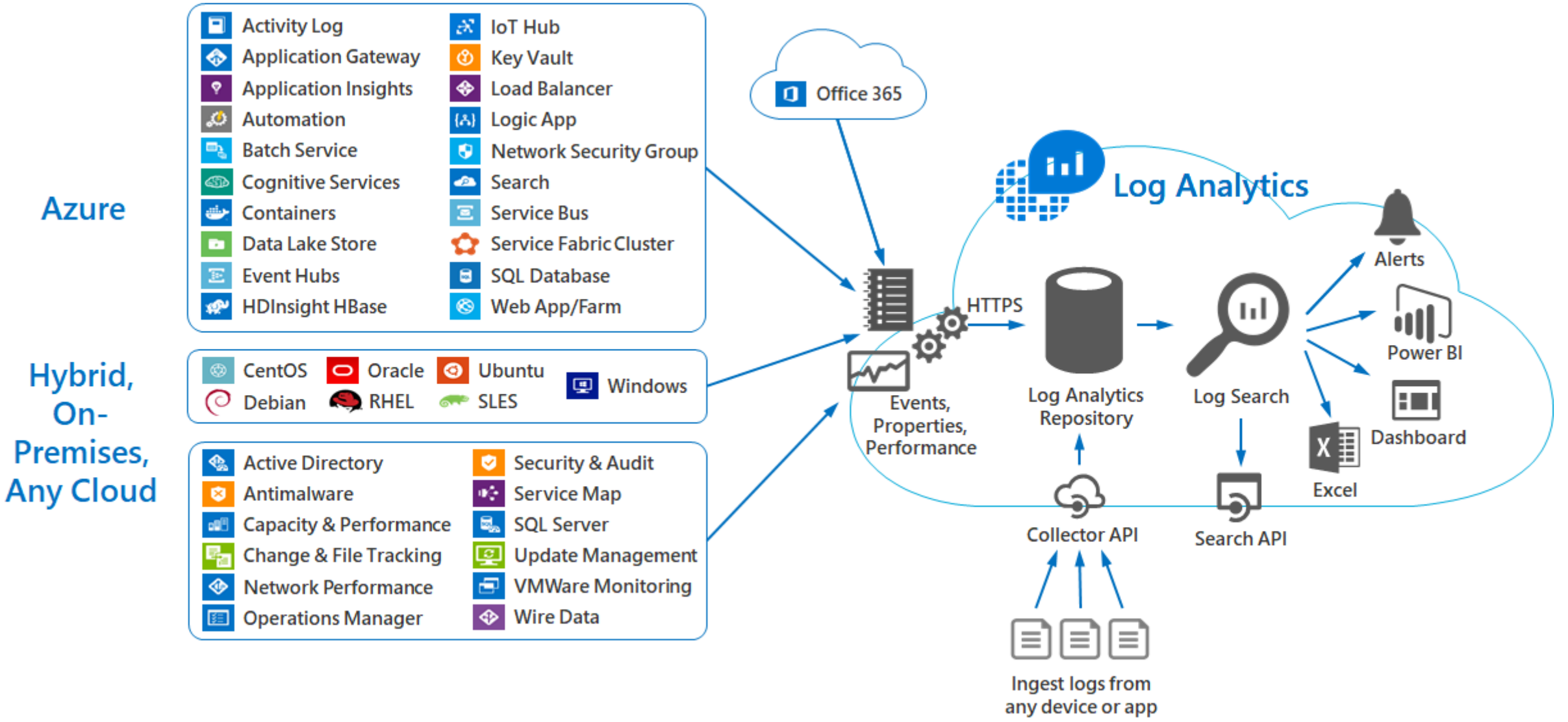
Log Analytics and Application Insights have been consolidated into Azure Monitor to provide a single integrated experience for monitoring Azure resources and hybrid environments. No functionality has been removed from these services, and you can perform the same scenarios as before with no loss or compromise of any features.



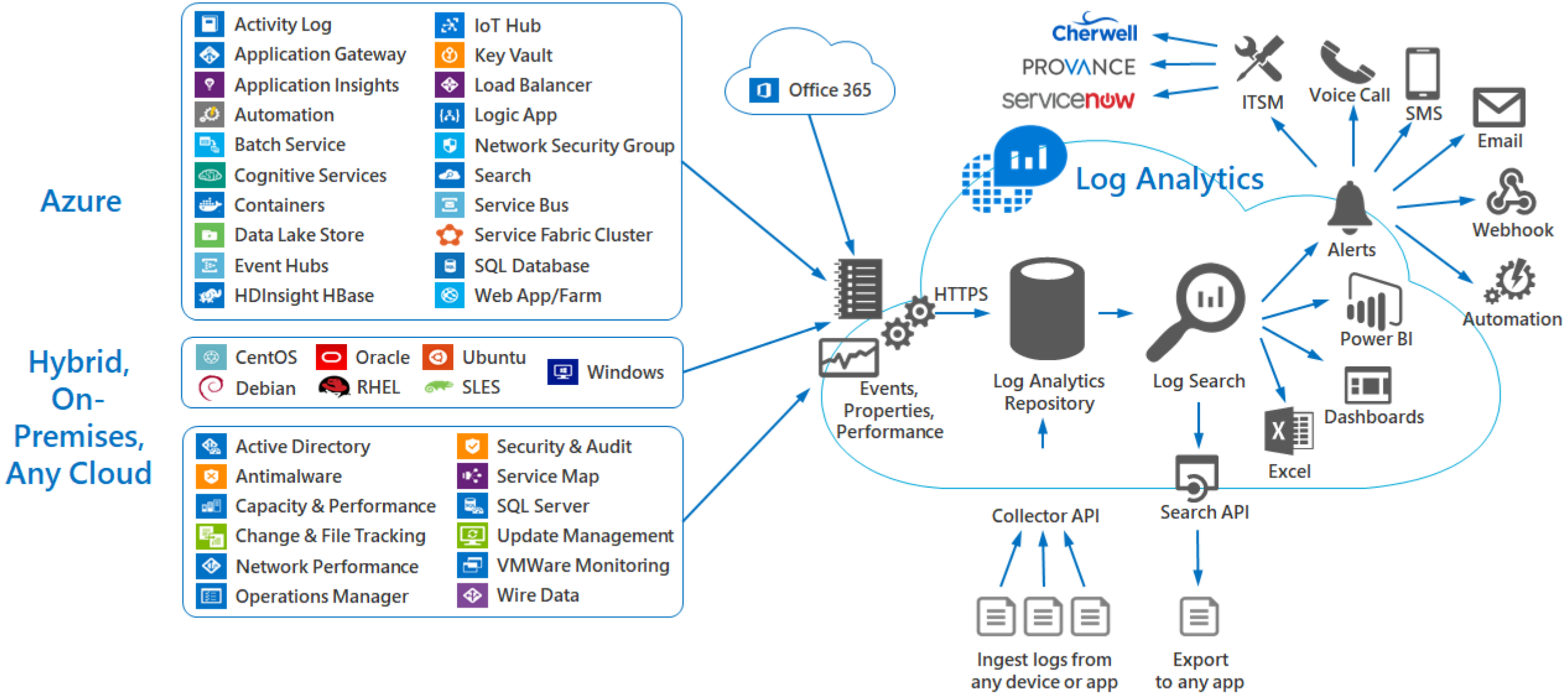
The view from above – custom logs



The view from above – extracting data



The view from above – alert remediation



Log Analytics – Data Retention

- Default is a 30 days retention (extendable for up to 2 years)
- Azure Activity Log is 90 days default (extendable up to 2 years)

Microsoft

Containers Solutions Monitoring

Azure Monitor Agents

Microsoft Services

Log Analytics Agent

- Developed for comprehensive management across on-premises machines, computers monitored by SCOM, and virtual machines in any cloud
- Windows and Linux agents attach to Azure Monitor, storing collected log data from different sources in your Log Analytics workspace
- Additionally stores unique logs or metrics as defined in a monitoring solution
- Both Linux and Windows agents communicate outbound to the Azure Monitor service over TCP port 443
- Agent is only available for Azure Kubernetes Service and Service Fabric; it can’t be used with Azure Container Instance

Agent Deployment

- Azure Security Center: when data collection is enabled, the Log Analytics Agent is automatically provisioned on all existing and any new supported virtual machines that are deployed in the subscription.

- Azure CLI: for Service Fabric and AKS cluster using Virtual Machine Scale Set you deploy the agent by targeting the VMSS.

- Azure Resource Manager: you can also use an ARM template to deploy the agent during the cluster creation.

Microsoft

Containers Solutions Monitoring

Azure Monitor Logs Search

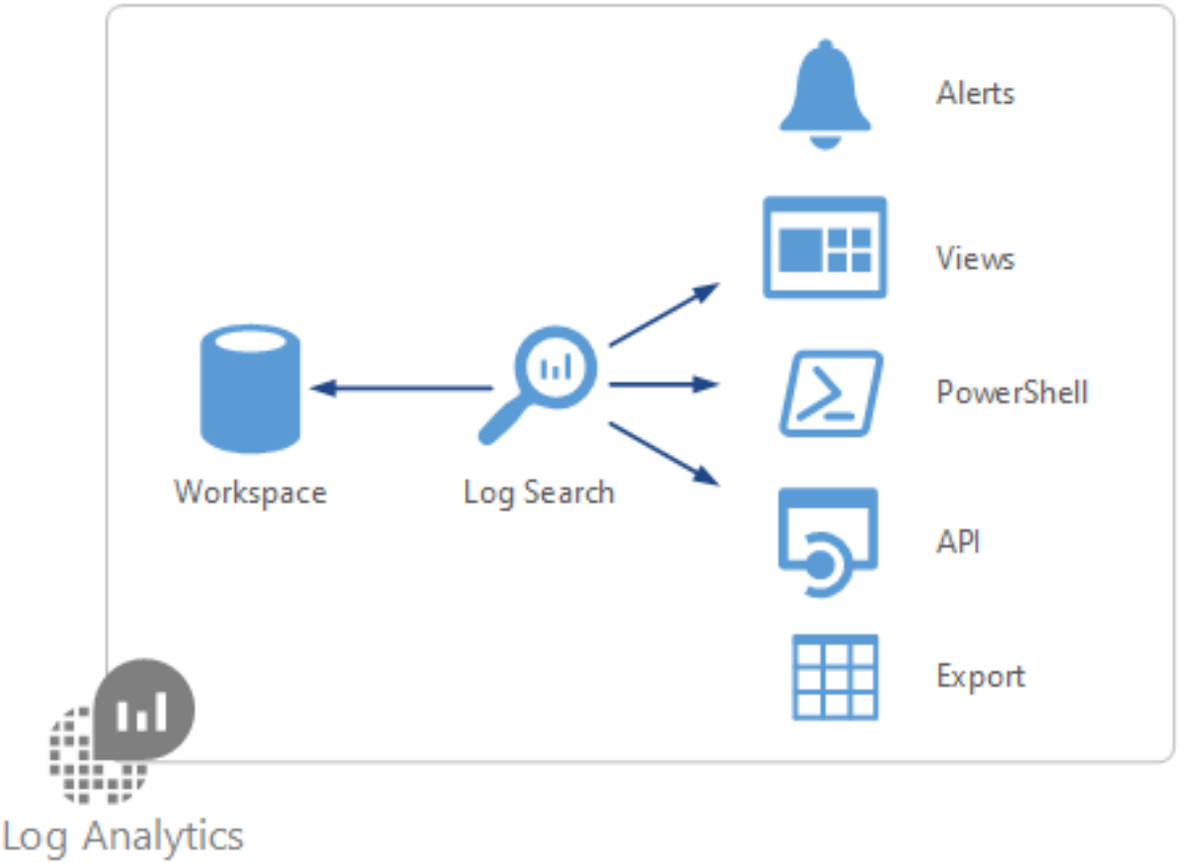
Microsoft Services

Log Search Overview

- Simple and powerful
- Full piping language
- Search-time field extractions
- Advanced queries (joins, multiline queries, advanced date/time functions)
- Smart Analytics (Advanced algorithms)
- Consistency with other applications (Application Insights)
- Integration with Power BI

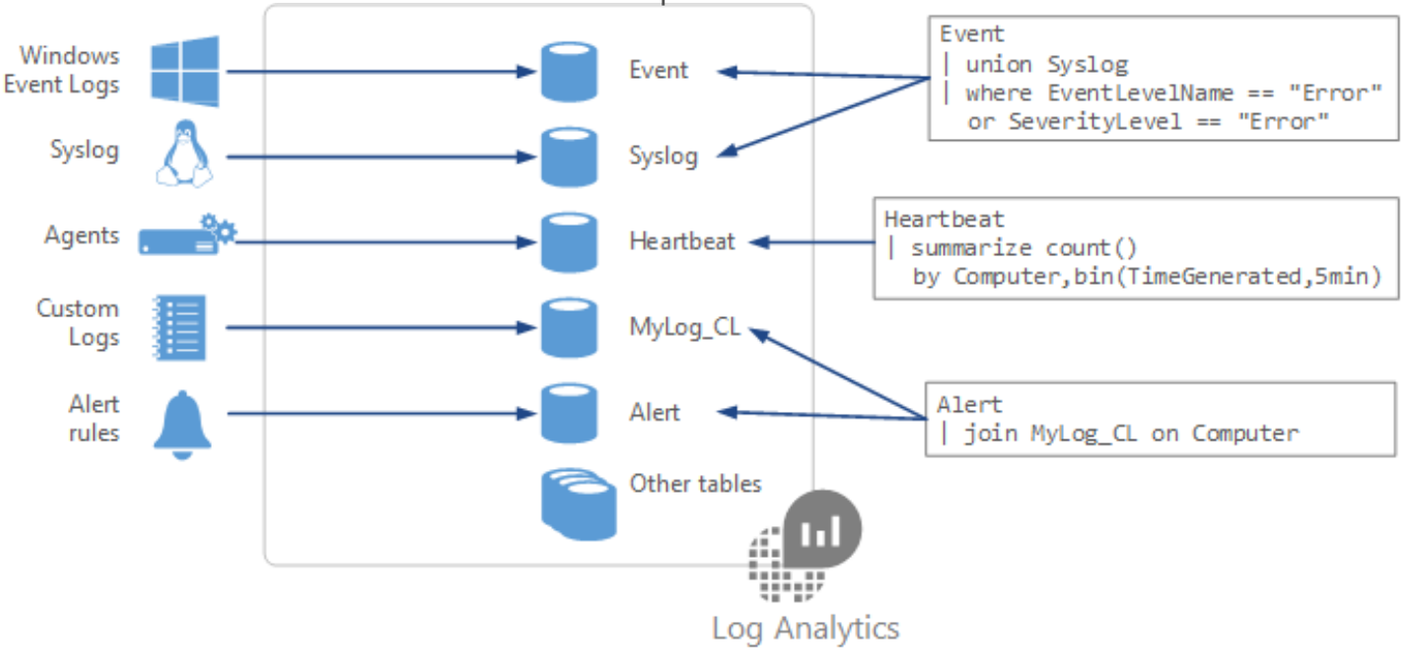
Where Log Searches are Used

- Portals
- Alert Rules
- Views
- Export
- PowerShell
- Log Analytics Search API



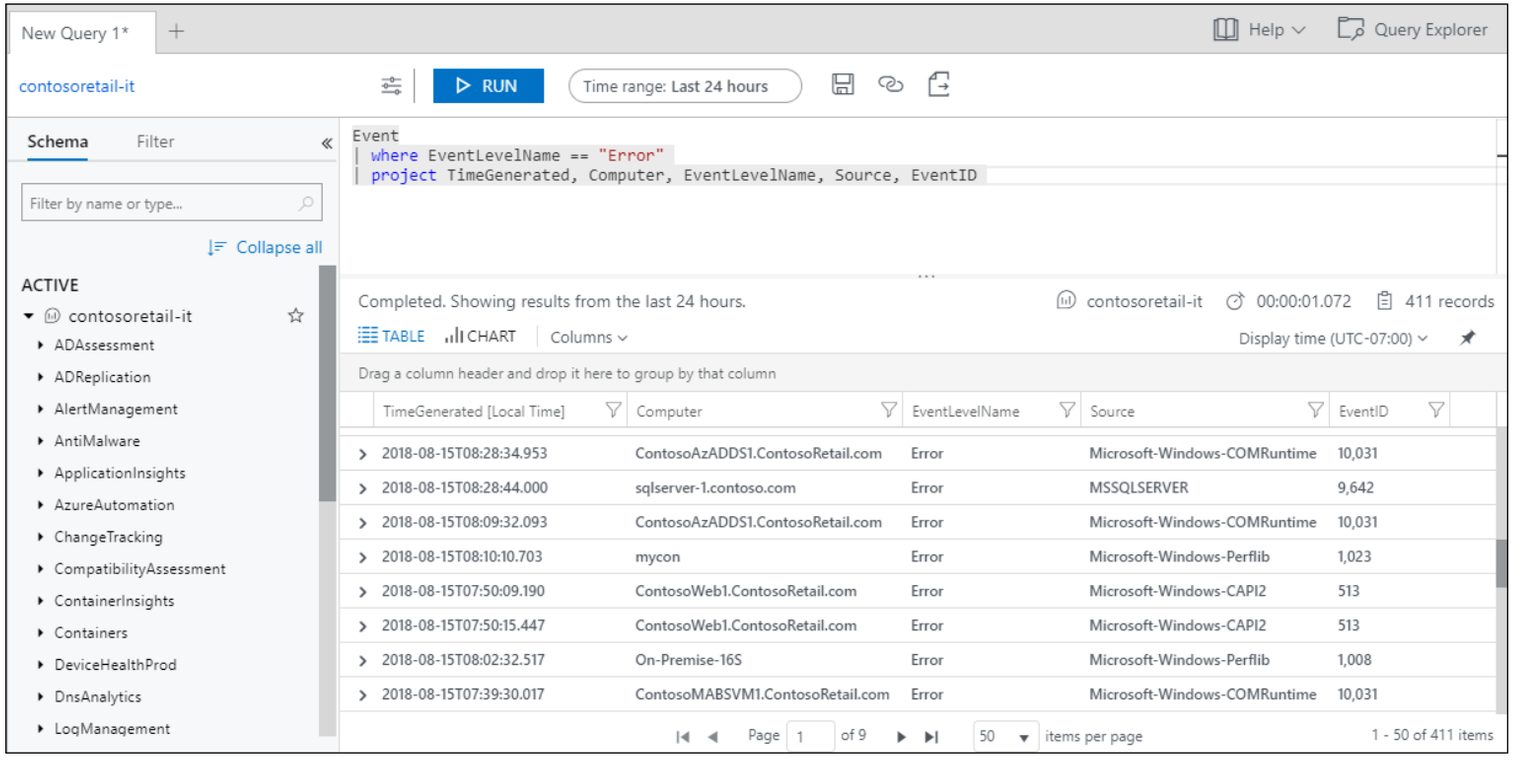
How Log Analytics Data is Organized

- When writing a query
  - Determine which tables have the data that you're looking for
  - Each data source and solution stores its data in dedicated tables
  - Each data source and solution includes the name of the data type and a description of each of its properties.
  - Each query can include data from multiple tables



Viewing and analyzing data - Log Analytics page

- Open the Log Analytics page from **Logs** in the Log Analytics menu
- The Log Analytics page provides:
  - Multiple tabs
  - Rich visualizations
  - Intellisense and language auto-completion
  - Syntax highlighting
  - Query explorer
  - Schema view
  - Share
  - Smart Analytics
  - Column selection



Microsoft

Containers Solutions Monitoring

Azure Monitor Alerts

Microsoft Services

Azure Monitor - Alerts

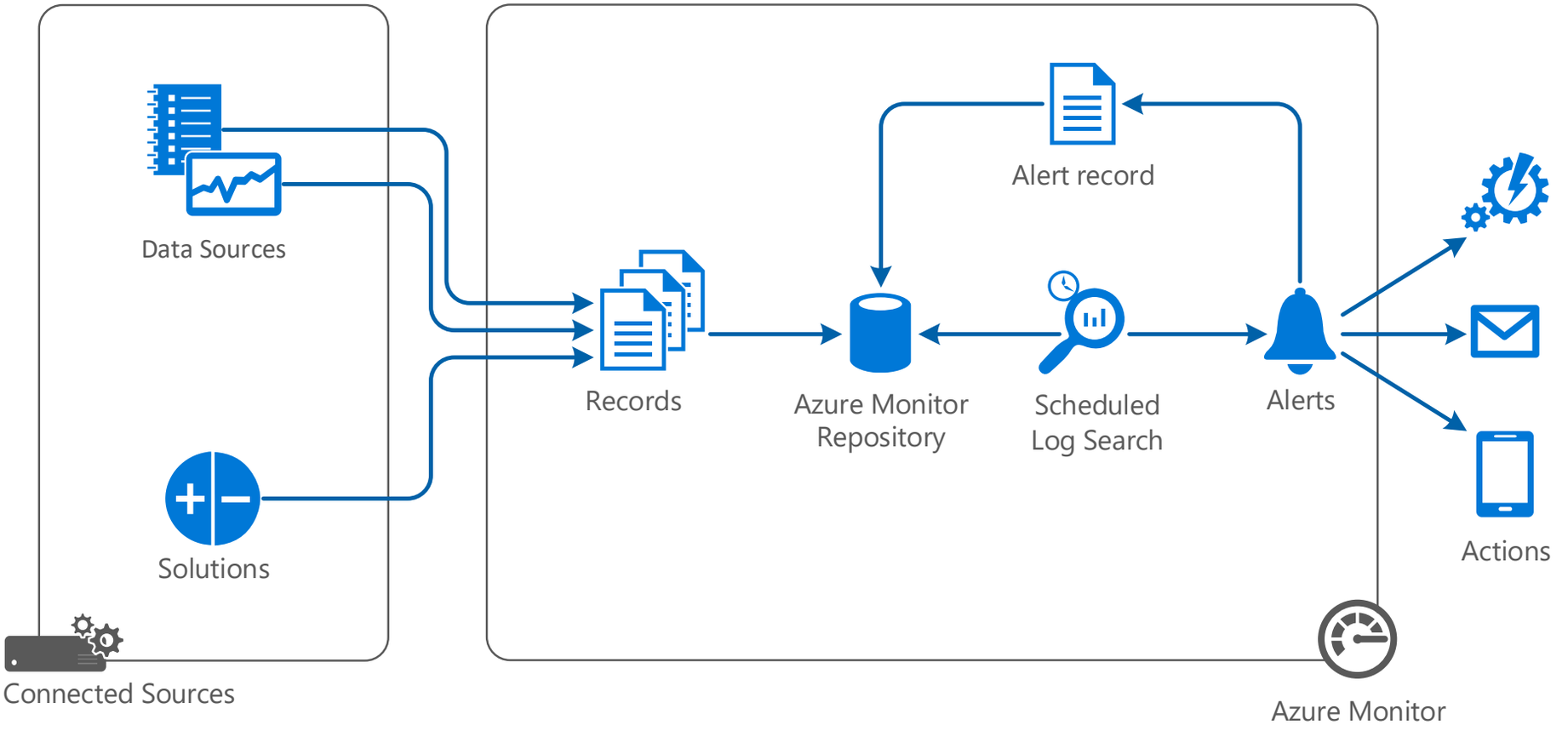
Alert on your Azure services:

- Metric values: Triggers when value of a specified metric crosses threshold
- Activity log events: Trigger on every event or only when certain events occurs

Alert based on an Azure Log Analytics Search:

- Custom Search
- Saved Search Query

Alerts - Flow


Data Sources
Solutions
Connected Sources
Records
Alert record
Azure Monitor
Repository
Scheduled
Log Search
Alerts
Actions
Azure Monitor

Alerts – General

Alert rule has three parts:

- Target Resource defines the scope and signals available for alerting. A target can be any Azure resource. Example targets: a virtual machine, a storage account, a virtual machine scale set, a Log Analytics workspace or an Application Insights resource.
  - Signals are emitted by the target resource and can be of several types. Metric, Activity log, Application Insights, and Log.

- Criteria is the combination of Signal and Logic applied on a Target resource, e.g.:
  - Percentage CPU > 70%
  - Server Response Time > 4 ms
  - Result count of a log query > 100

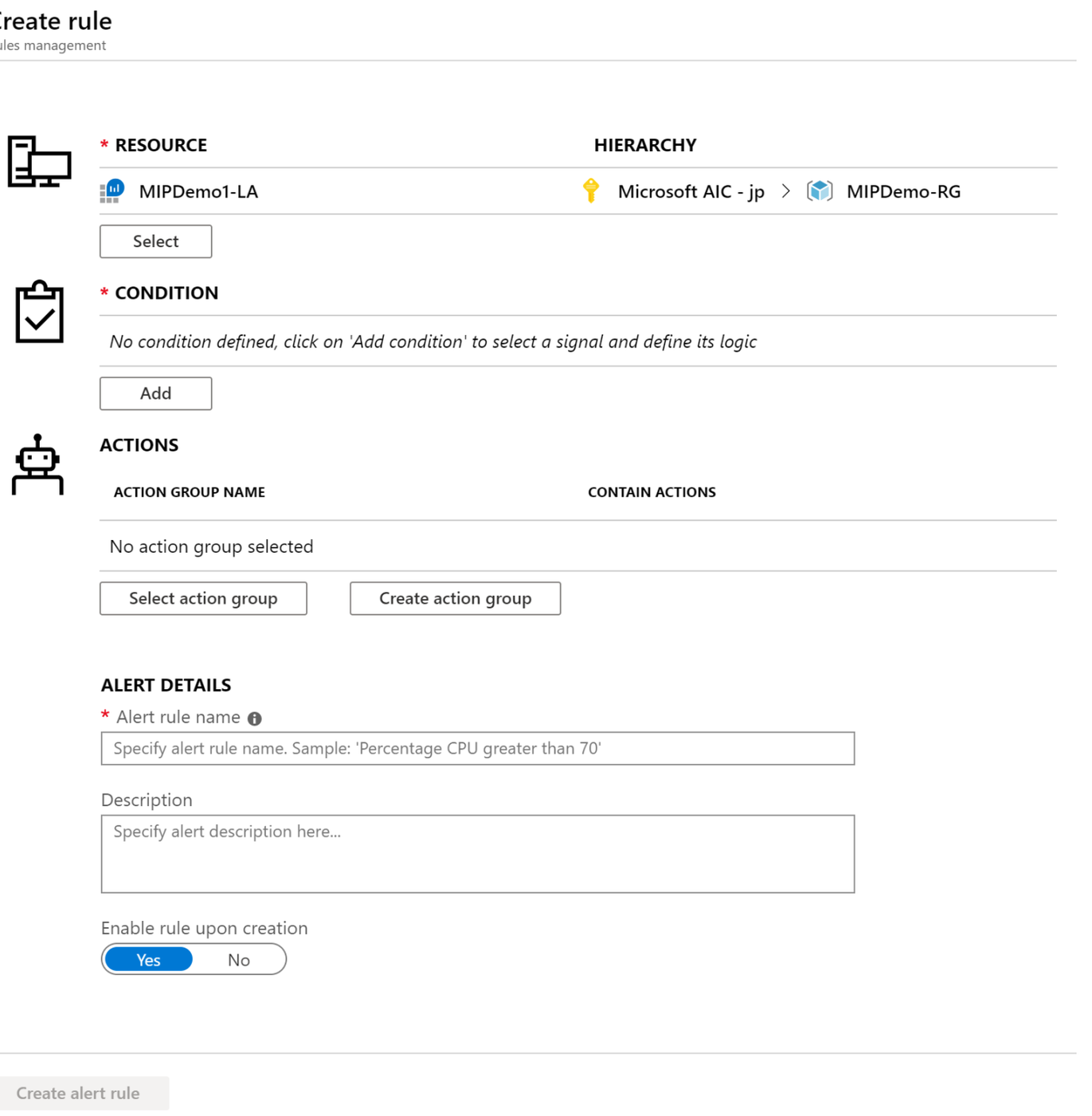
- Action is a specific call sent to a receiver of a notification - email, SMS, webhook etc.

Alerts – Create Rule

1. Define alert condition
   - Alert Target (Log Analytics WS)
   - Target Criteria

2. Define alert details
   - Alert rule name
   - Description
   - Severity

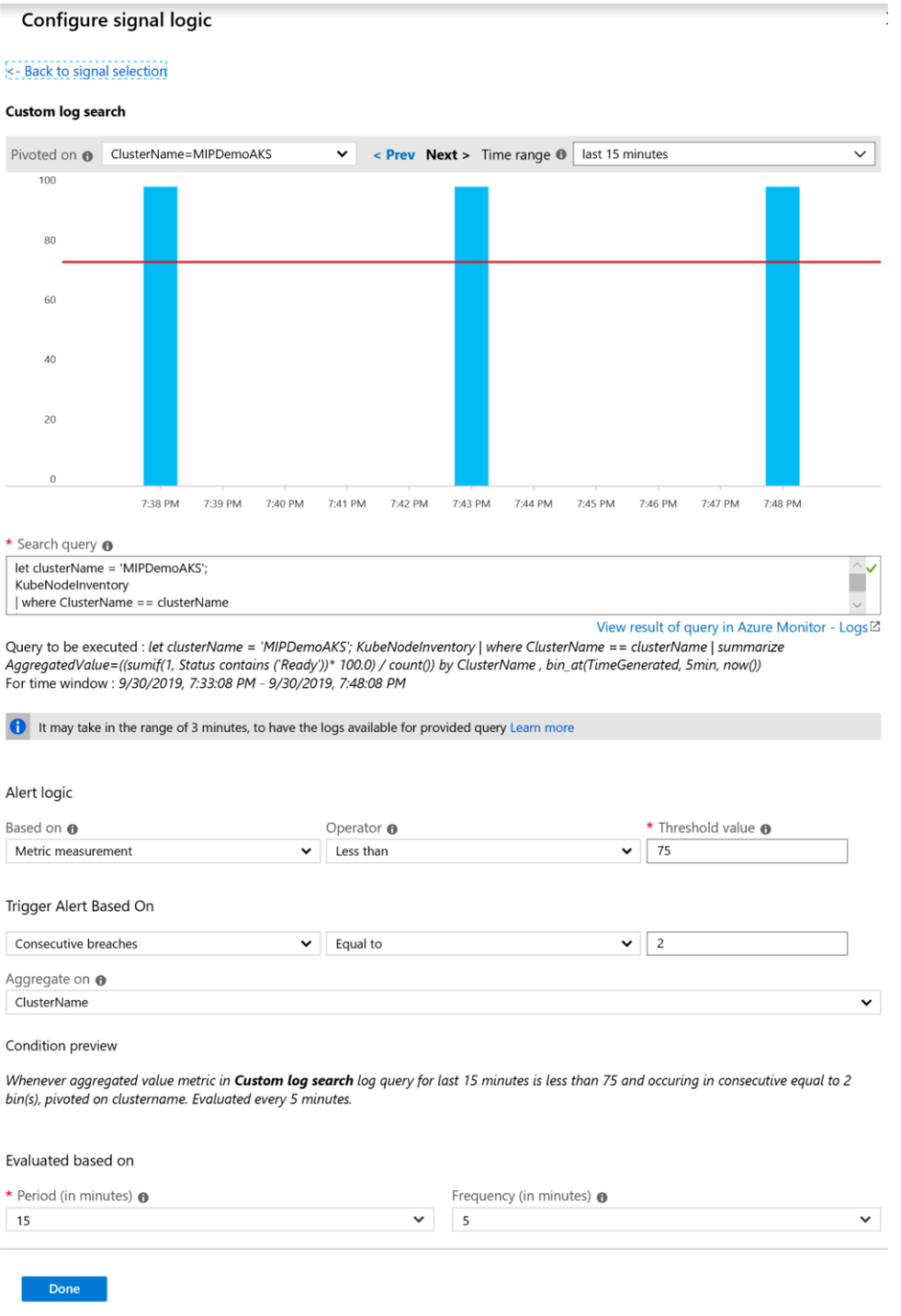
3. Define action group
   - Action Group
   - Customize Actions



Alerts – Target criteria

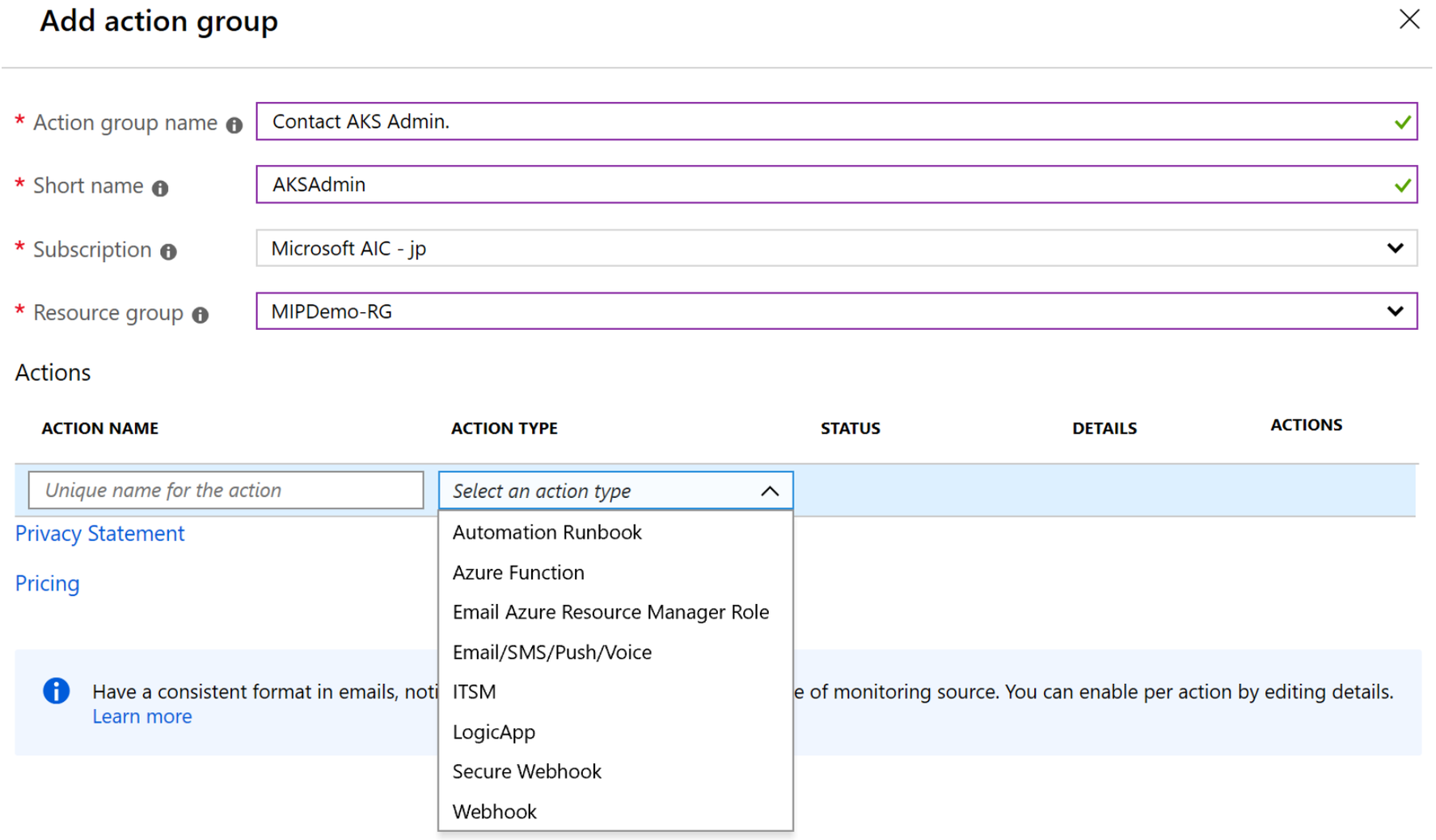
- Search query
- Alert logic
  - Number of Results
  - Metric Measurement

- Evaluated based on
  - Period (in minutes)
    - Value of 5 minutes to 24 hours
    - Should be greater than or equal to how often the query runs
  - Frequency



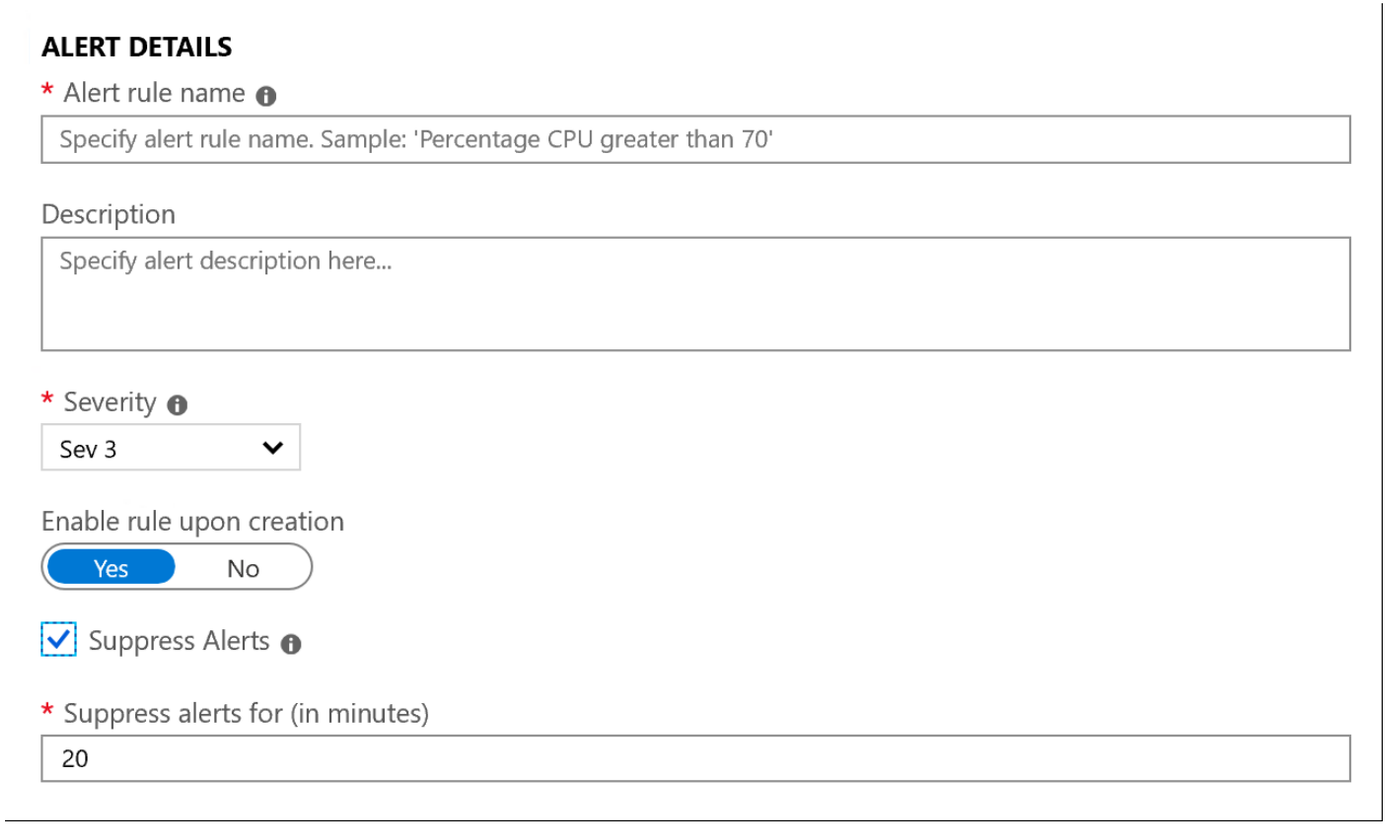
Alerts – Add action group

- Action group name (Unique in subscription)
- Short name (<12 char)
- Subscription
- Resource group
- Actions
  - Action Name
  - Action Type

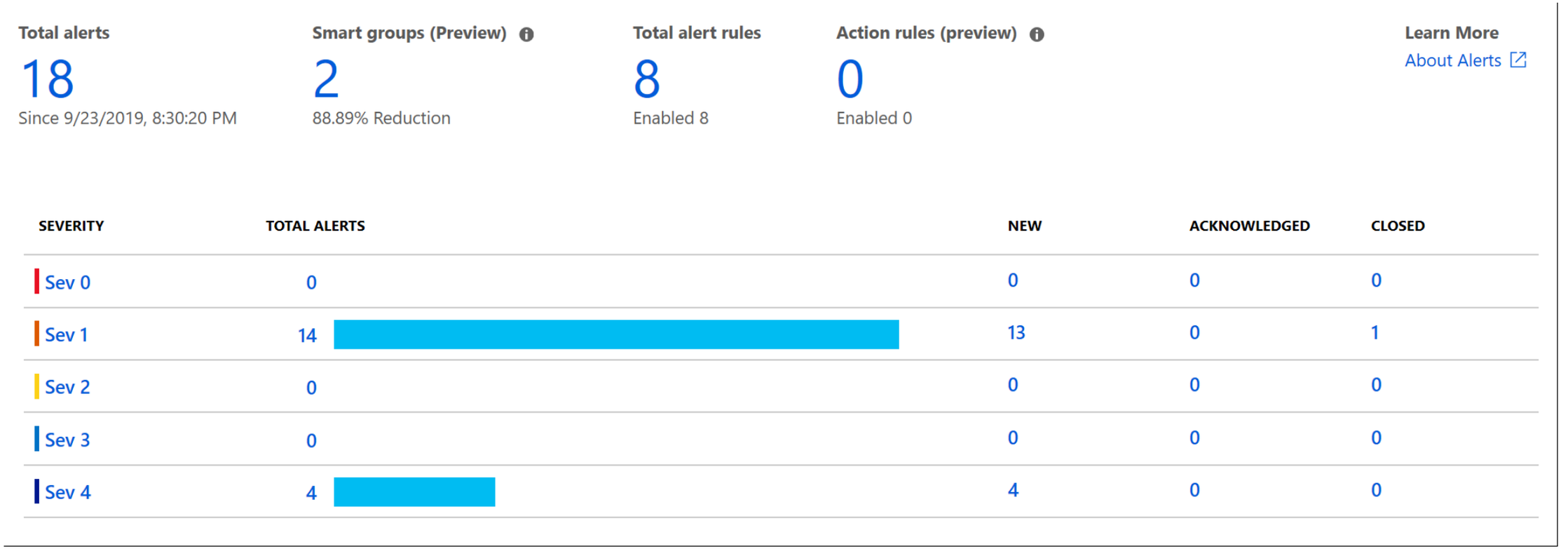


Alerts – Define alert details

- Alert rule name
- Alert Description
- Severity
  - Critical (Sev 0)
  - Warning (Sev 1)
  - Informational (Sev 2)
- Enable rule upon creation
- Suppress Alerts



Alerts – View Alerts



Microsoft

Containers Solutions Monitoring

Related Azure Services

Microsoft Services

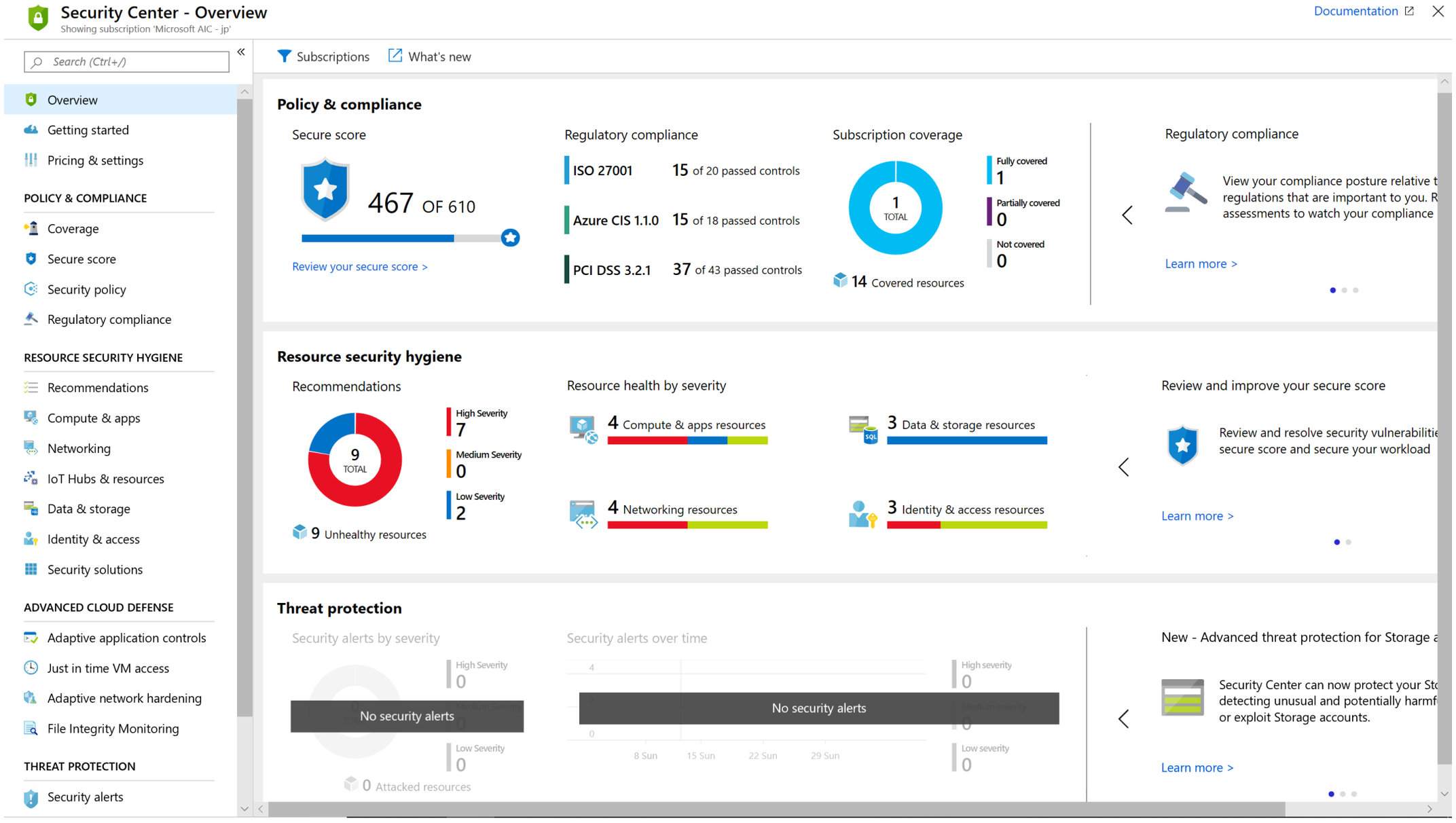
Azure Security Center

Azure Security Center helps you identify and perform the hardening tasks recommended as security best practices and implement them across your machines, data services and apps.

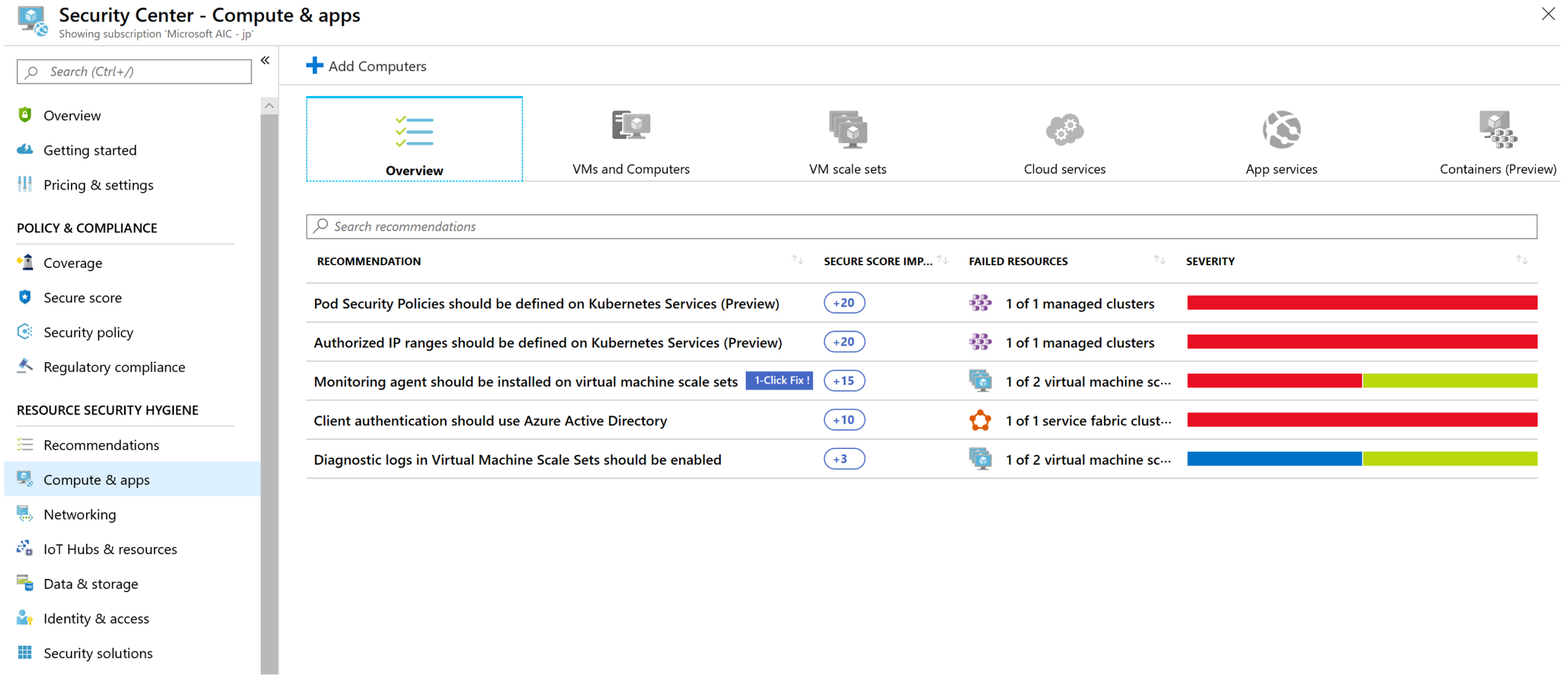
Security Center is natively part of Azure, PaaS services in Azure - including Service Fabric, SQL databases and storage accounts. These are monitored and protected by Security Center without necessitating any deployment.

The events collected from the agents and from Azure are correlated in the security analytics engine to provide you tailored recommendations (hardening tasks), that you should follow to make sure your workloads are secure, and gives you threat detection alerts.

Azure Security Center - Overview



Azure Security Center – Compute & Apps Overview



Azure Security Center - Recommendations

Pod Security Policies should be defined on Kubernetes Services (Preview)

Description

Define Pod Security Policies to reduce the attack vector by removing unnecessary application privileges. It is recommended to configure Pod Security Policies to only allow pods to access the resources which they have permissions to access.

General Information

Recommendation score	0/20
Recommendation impact	+20
User impact	High
Implementation effort	Moderate

LEARN MORE

Learn more about recommendations

Learn how to disable the recommendation

Threats

- Elevation of privilege
- Data exfiltration

Remediation steps

Manual remediation:

To configure Pod Security Policies, follow the steps described here Secure your cluster using pod security policies in Azure Kubernetes Service (AKS).

Effected resources

Unhealthy resources (1) | Healthy resources (0) | Unscanned resources (0)

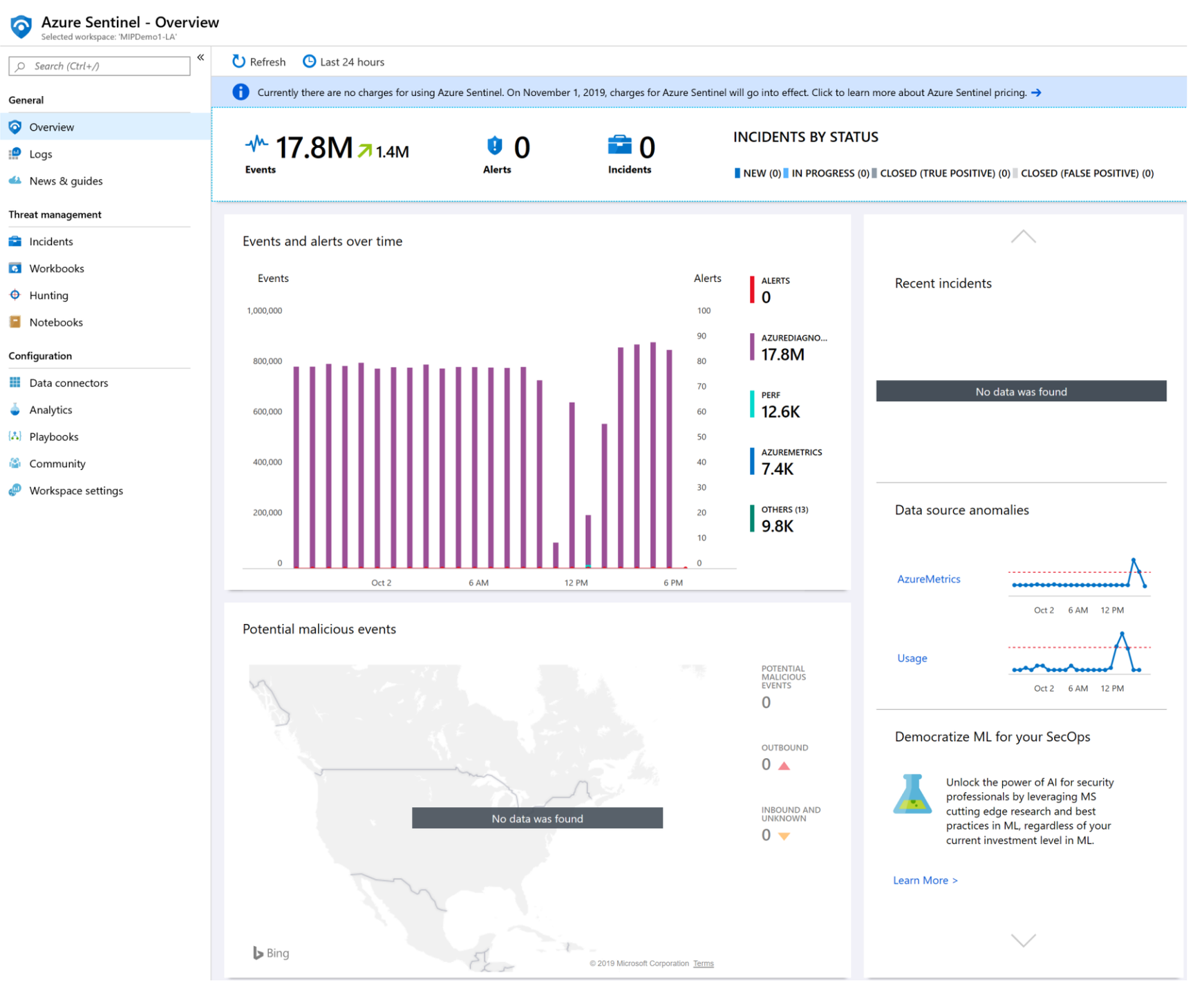
Search managed clusters

NAME	SUBSCRIPTION
mipdemoaks	Microsoft AIC - jp

Azure Sentinel - Overview

- Scalable, cloud-native security information event management (SIEM) and security orchestration automated response (SOAR) solution
- Delivers intelligent security analytics and threat intelligence across the enterprise
- Single solution for alert detection, threat visibility, proactive hunting, and threat response

Azure Sentinel



Microsoft

Containers Solutions Monitoring

Azure Monitoring for Containers

Microsoft Services

Monitoring Containers in Azure

Monitoring containers can be divided in two categories:

- Application
  - Application Insights monitors the availability, performance, and usage of your web applications
- Infrastructure
  - Log Analytics Agent
  - Cluster's Dashboard
  - Azure Monitor for containers for AKS platform
  - Service Fabric Analytics

Microsoft

Containers Solutions Monitoring

Azure Monitoring for Azure Container Instance

Microsoft Services

Monitor container instance resources

- Provides insight into the compute resources used by container instances
- Provides metrics that track network activity in your container instances

- Identifies processor and memory utilization of container groups and their containers hosted in Azure Container Instances

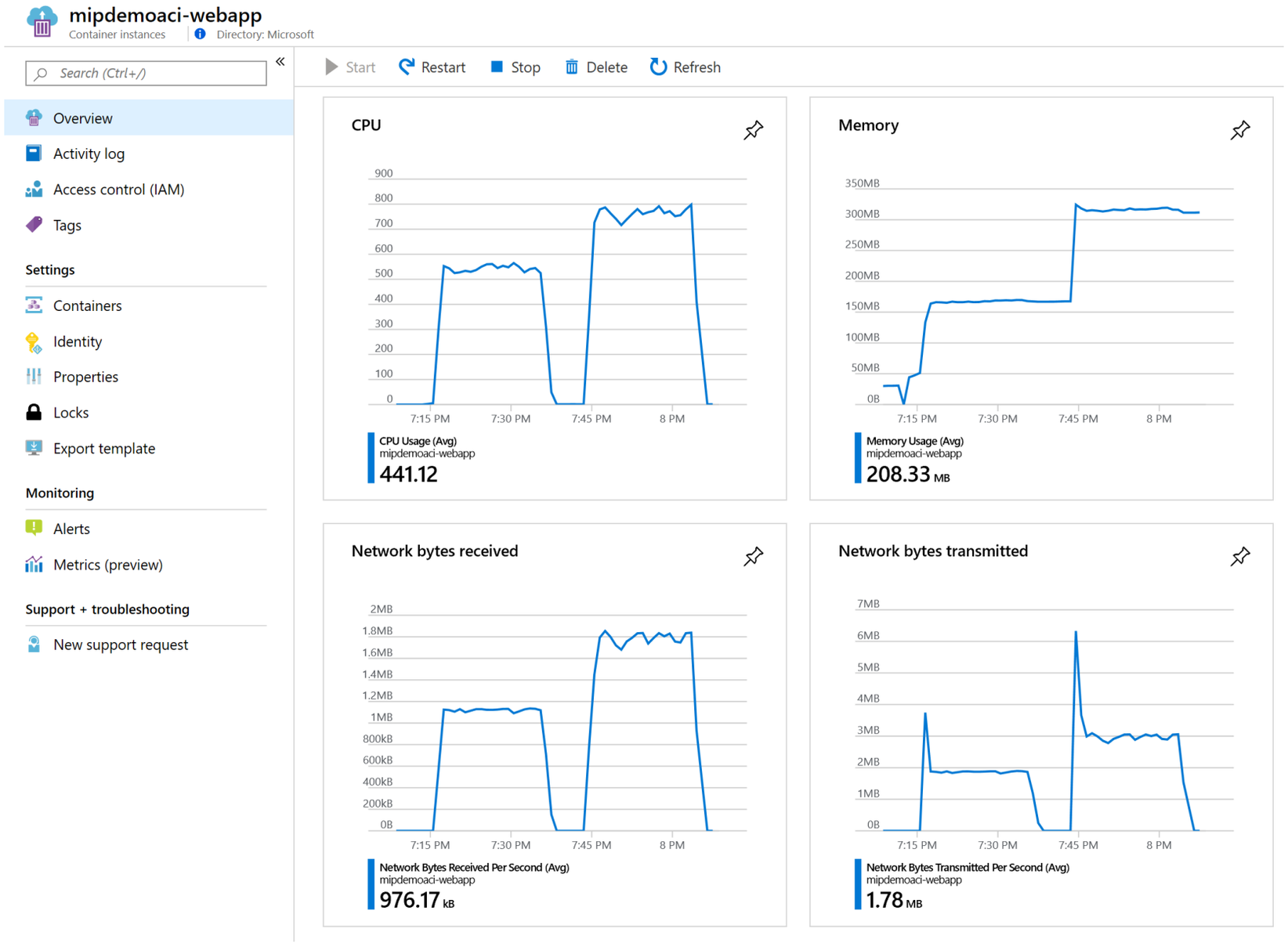
- Azure Monitor metrics in Azure Container Instances are currently in preview and only available for Linux containers

Monitor container instance - Available metrics

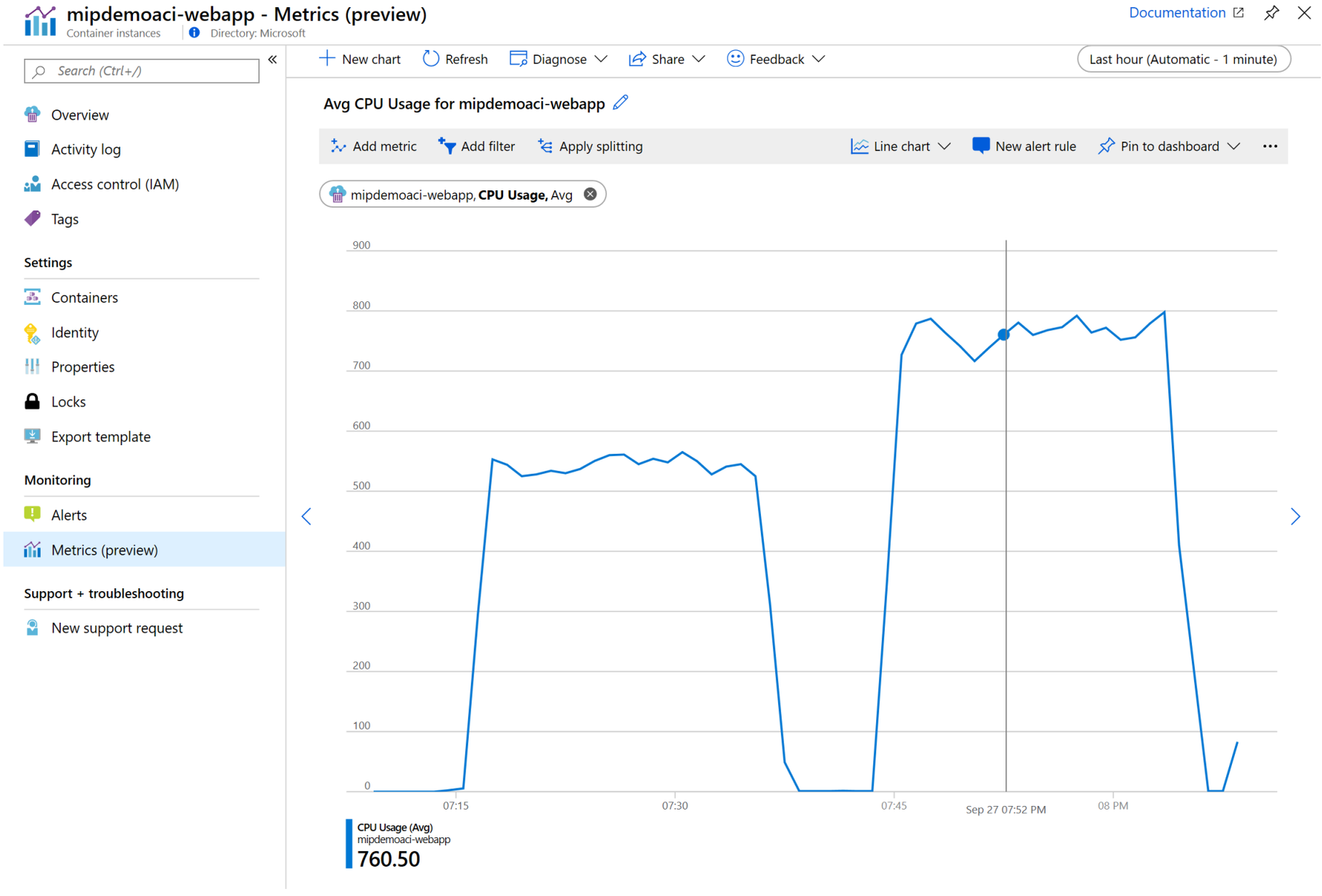
Azure Monitor provides the following metrics. These metrics are available for a container group and individual containers.

- **CPU Usage** - measured in millicores. One millicore is 1/1000th of a CPU core
  - Aggregated as average usage across all cores
- **Memory Usage**
  - Aggregated as average bytes
- **Network Bytes Received** Per Second and **Network Bytes Transmitted** Per Second
  - Aggregated as average bytes per second.

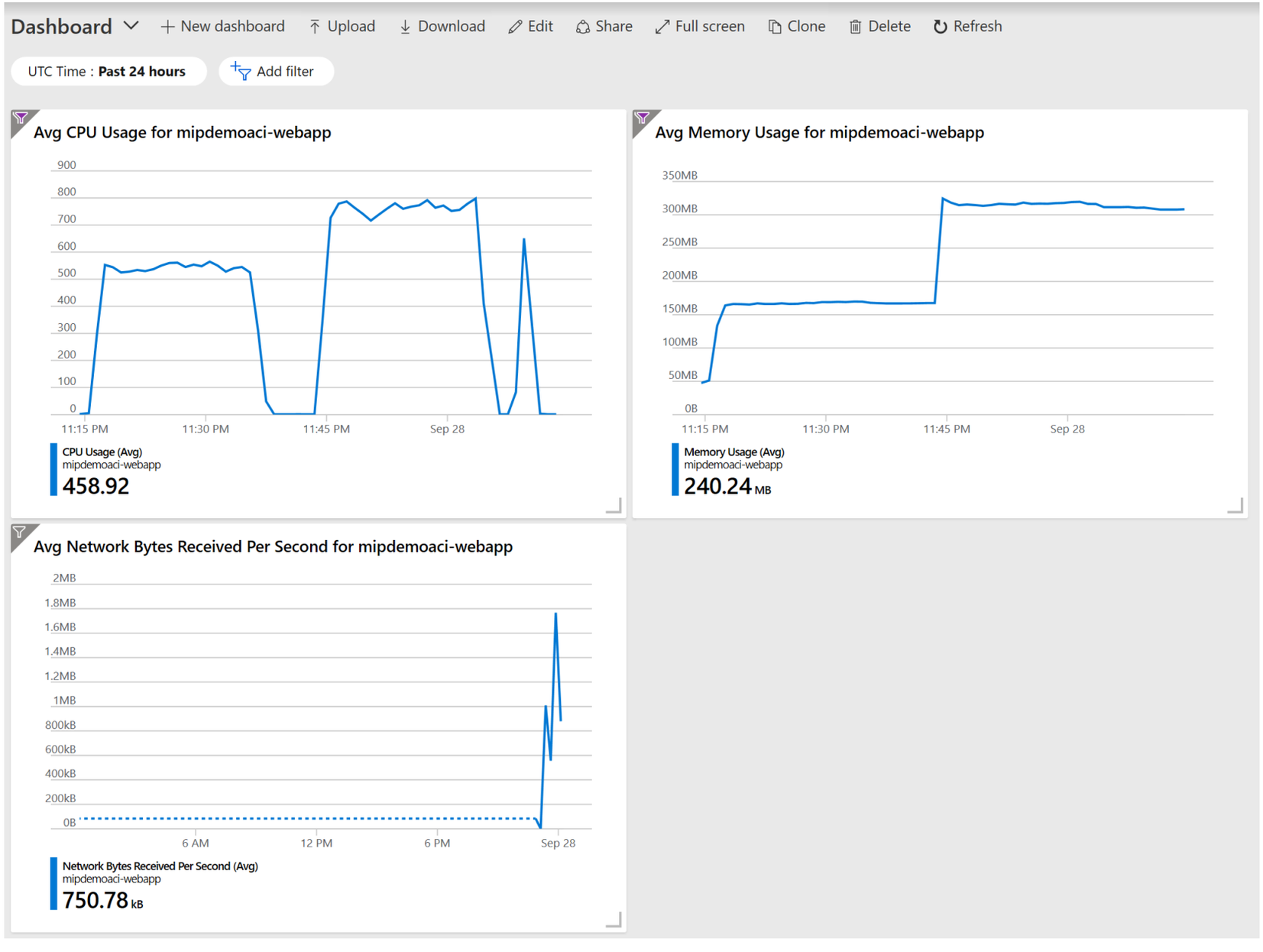
Monitor container instance – Metrics Overview via Portal



Monitor container instance – Metrics via Portal

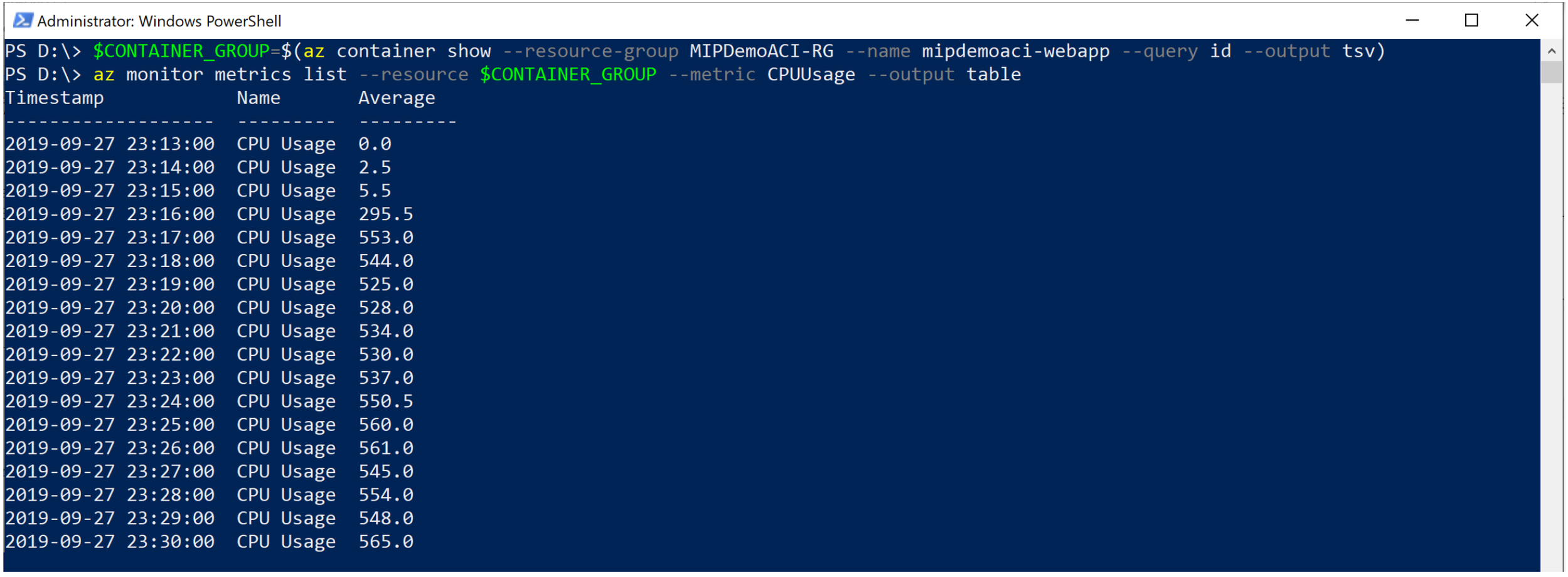


Monitor container instance – Metrics on Dashboard



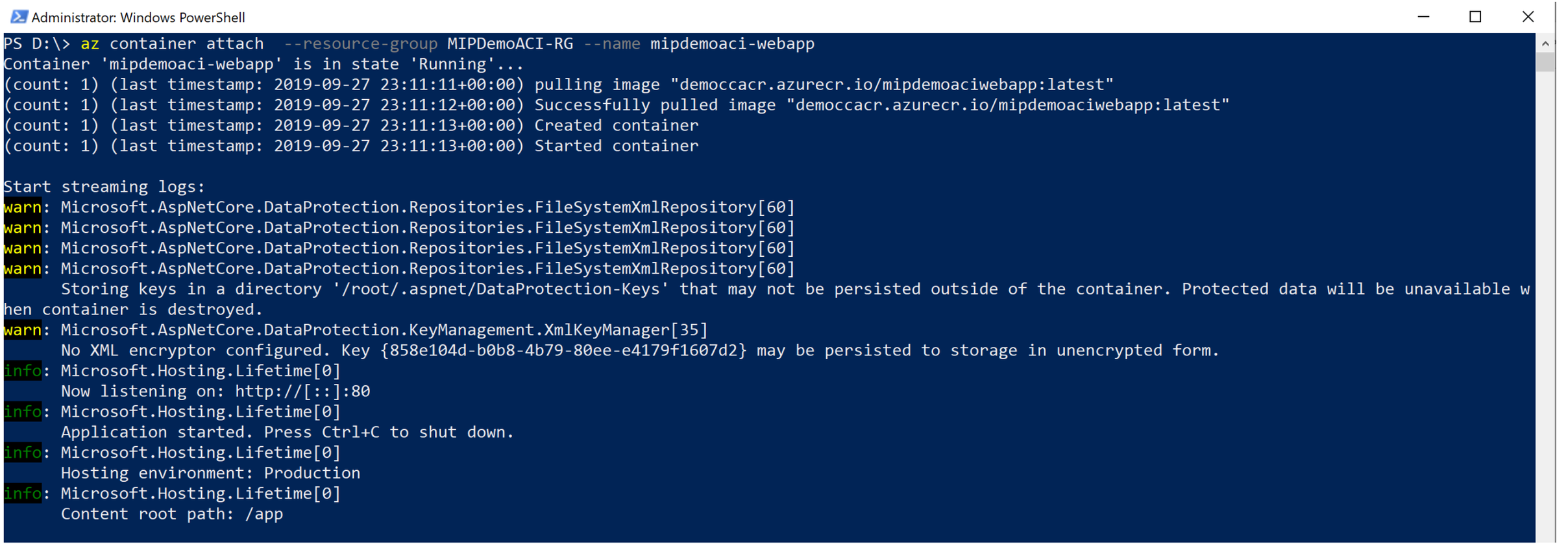
Monitor container instance – Metrics via Azure CLI

You can also access the metrics through the Azure CLI



Retrieve container logs and events

- View container logs with “az container logs”
- Stream container standard out and standard error with “az container attach”



Retrieve diagnostic events

If your container fails to deploy successfully, you need to review the diagnostic information provided by the Azure Container Instances resource provider.

The output includes the core properties of your container, along with deployment events (shown here truncated):

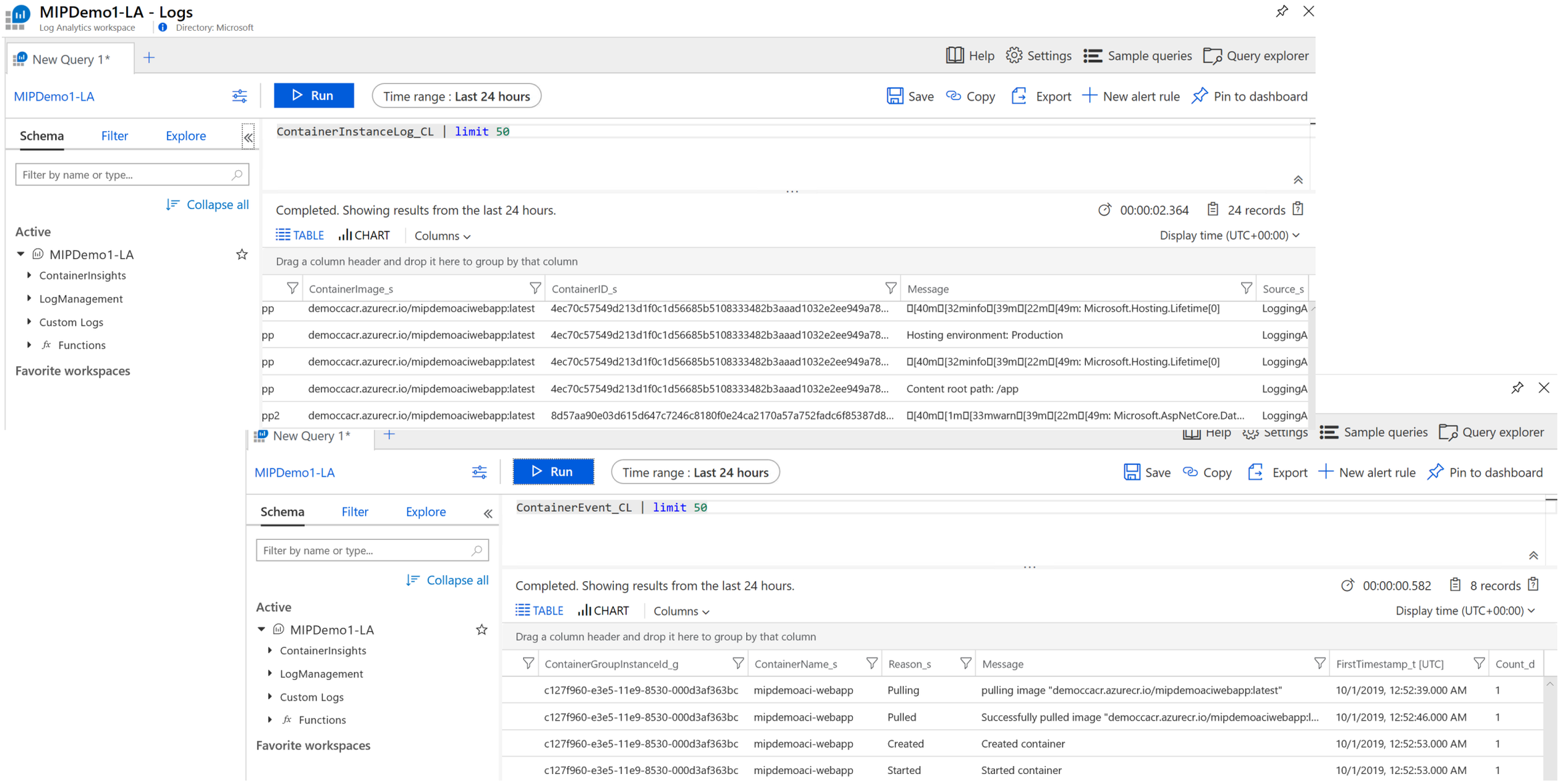
Administrator: Windows PowerShell

```
PS D:\> az container show  --resource-group MIPDemoACI-RG --name mipdemoaci-webapp
{
  "containers": [
    {
      "command": null,
      "environmentVariables": [],
      "image": "democcacr.azurecr.io/mipdemoaciwebapp:latest",
      "instanceView": {
        "currentState": {
          "detailStatus": "",
          "exitCode": null,
          "finishTime": null,
          "startTime": "2019-09-27T23:11:13+00:00",
          "state": "Running"
        },
        "events": [
          {
            "count": 1,
            "firstTimestamp": "2019-09-27T23:11:11+00:00",
            "lastTimestamp": "2019-09-27T23:11:11+00:00",
            "message": "pulling image \"democcacr.azurecr.io/mipdemoaciwebapp:latest\"",
            "name": "Pulling",
            "type": "Normal"
          },
```

Logging with Azure Monitor logs

- Centralized location for storing and querying log data from Azure resources
- Azure Container Instances include built-in support for sending logs and event data to Azure Monitor logs
- To send container group log and event data to Azure Monitor logs, you must specify a Log Analytics workspace ID and workspace key when creating a container group using the Azure CLI
- Event data can only be sent from Linux container instances to Log Analytics

View Logs and Events



Microsoft

Containers Solutions Monitoring

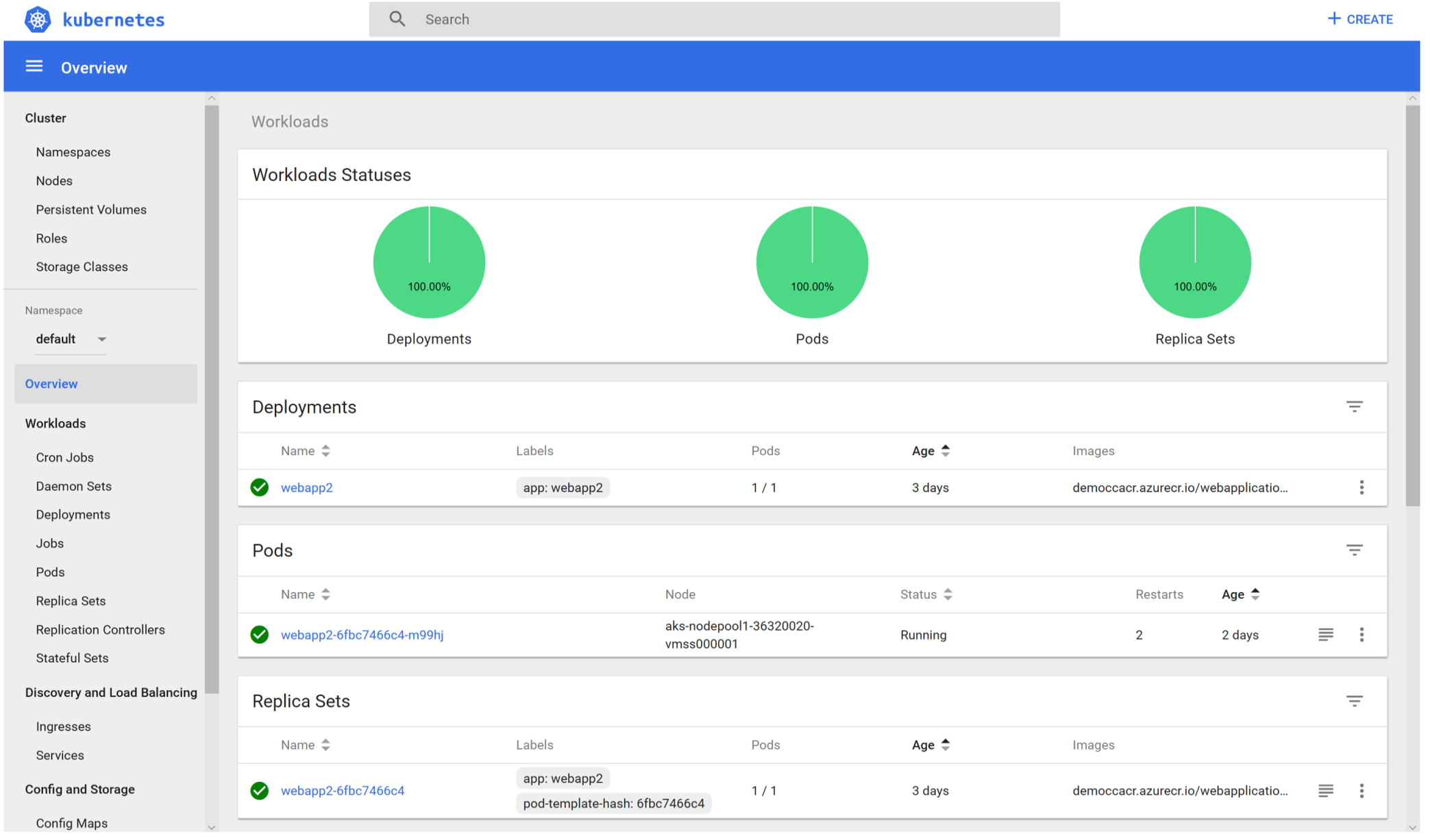
Azure Monitoring for Azure Kubernetes Service

Microsoft Services

Kubernetes Web Dashboard

- Kubernetes includes a web dashboard that can be used for basic management operations
- View basic health status and metrics for clusters and applications, create and deploy services, and edit existing applications
- Does not allow alerting, storage of metrics or querying data

Kubernetes Web Dashboard – Dashboard Overview



Azure Monitor for Containers

- Azure Monitor for containers is designed to monitor the performance of container workloads deployed to managed Kubernetes clusters hosted on Azure Kubernetes Service (AKS)

- Provides performance visibility by collecting container logs, memory and processor metrics from controllers, nodes and containers available in Kubernetes through the Metrics API

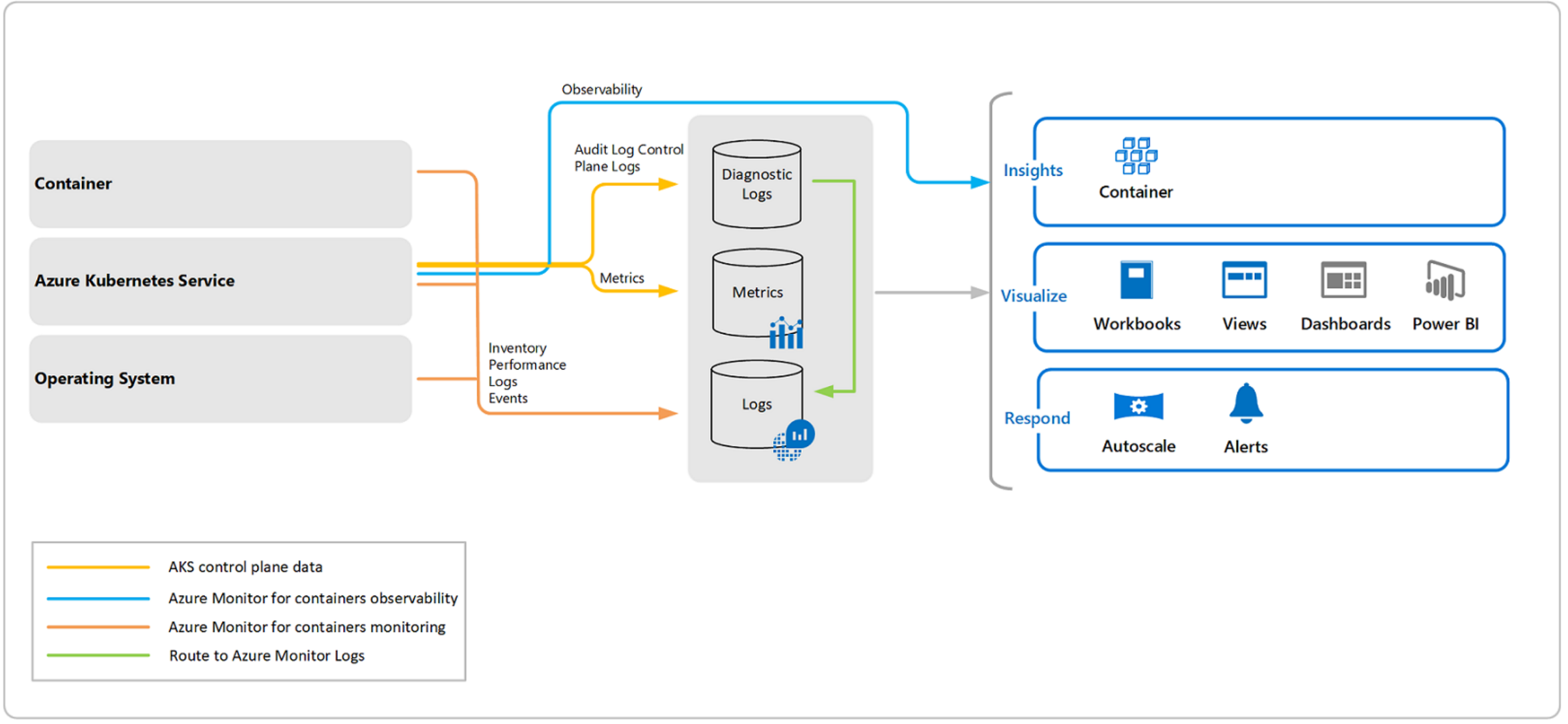
Azure Monitor for Containers

- Can be added during the creation of a cluster or to an existing cluster
- Once monitoring is enabled on Kubernetes clusters, metrics and logs are automatically collected through a containerized version of the Log Analytics agent for Linux
- Metrics are written to the metrics store and log data is written to the logs store associated with your Log Analytics workspace

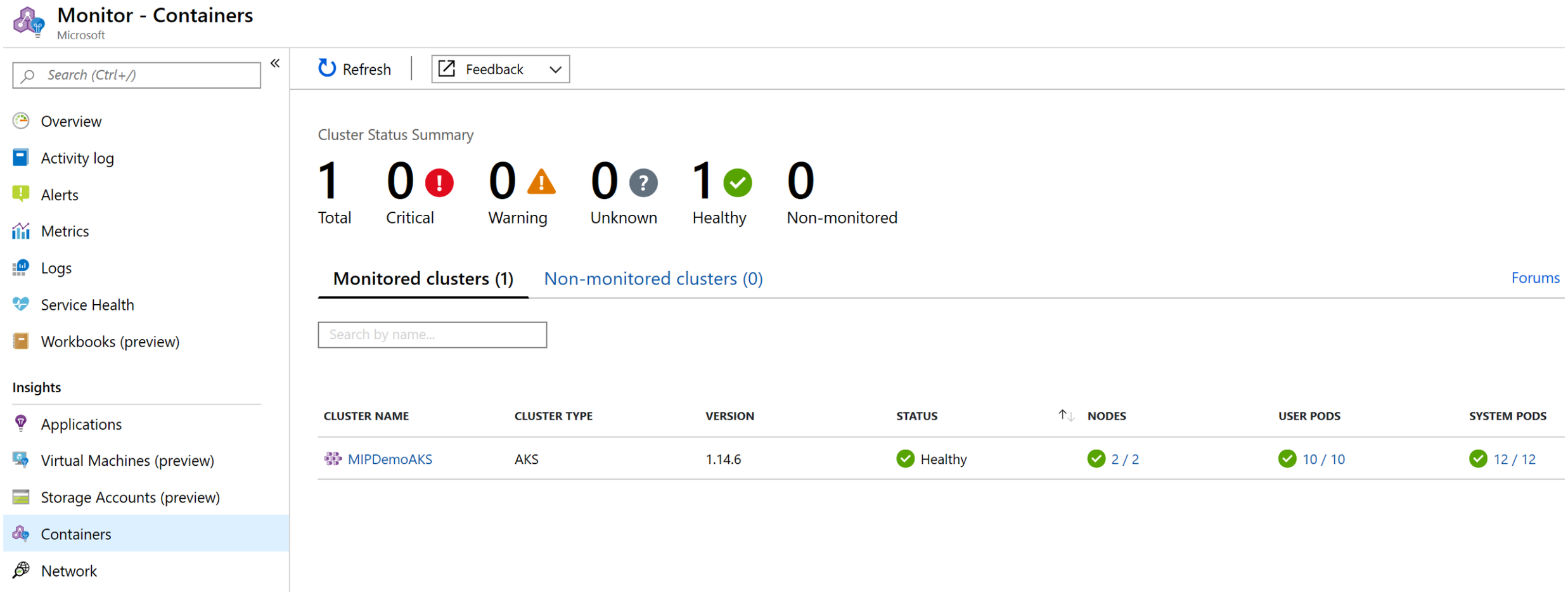
Azure Monitor for Containers

- Identify AKS containers that are running on the node and their average processor and memory utilization
- Identify where the container resides in a controller or a pod
- Review the resource utilization of workloads running on the host that are unrelated to the standard processes that support the pod
- Understand the behavior of the cluster under average and heaviest loads
- Configure alerts to proactively notify you or record them, e.g. when CPU and memory utilization on nodes or containers exceed your thresholds.

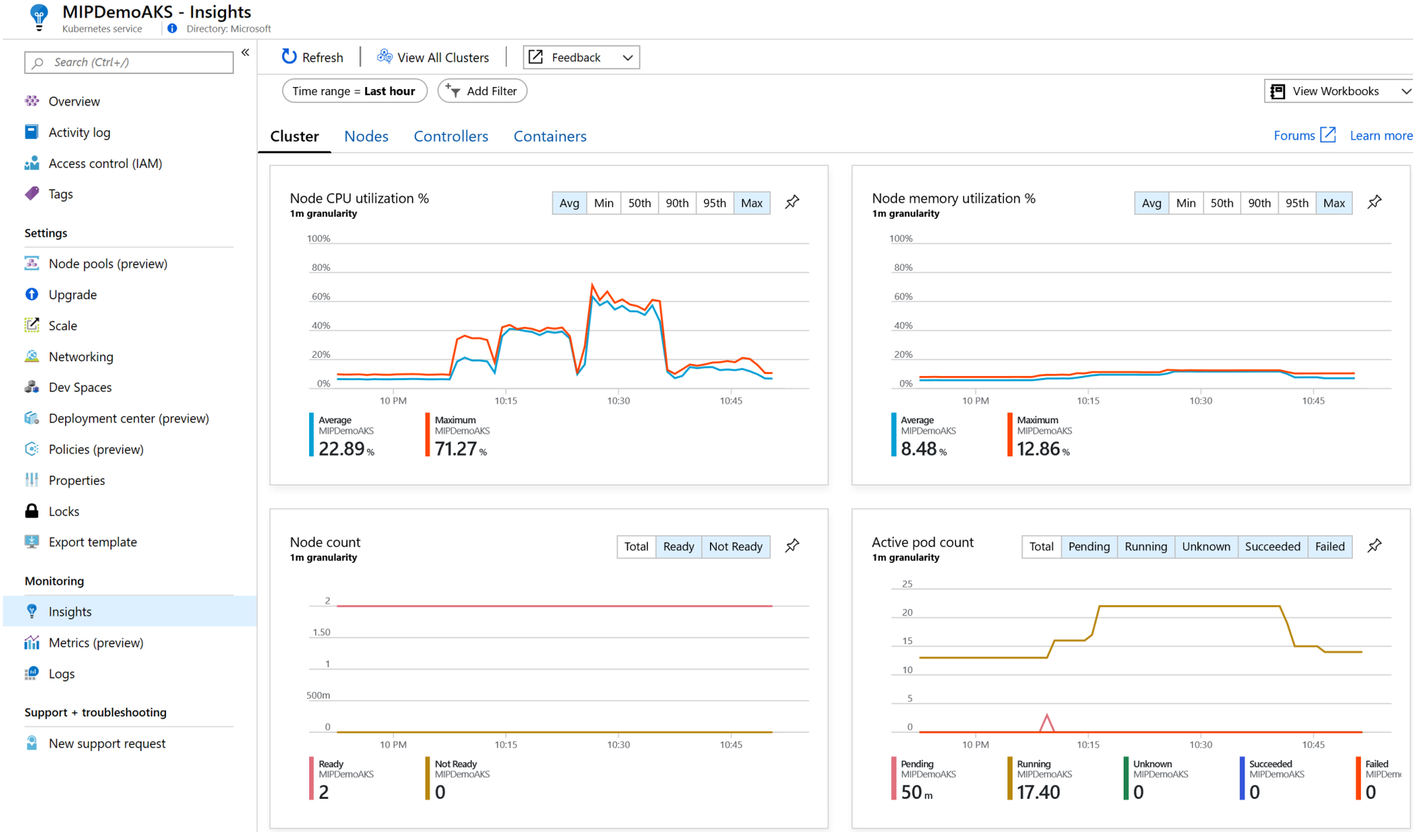
Azure Monitor for Containers - Flow



Azure Monitor for Containers – Clusters Overview

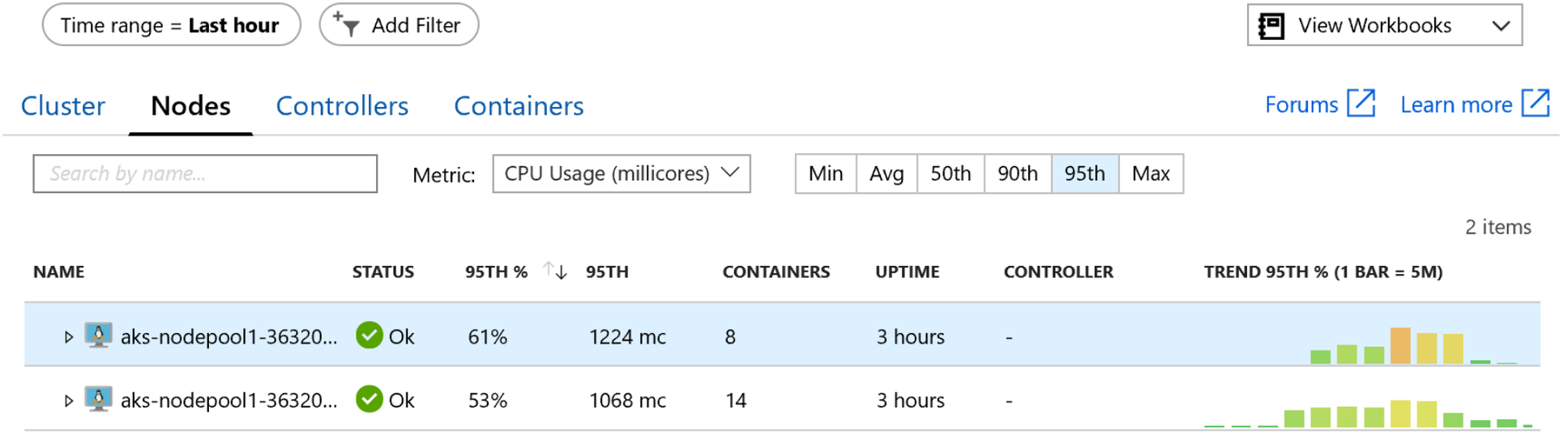


Azure Monitor for Containers – Cluster Insights

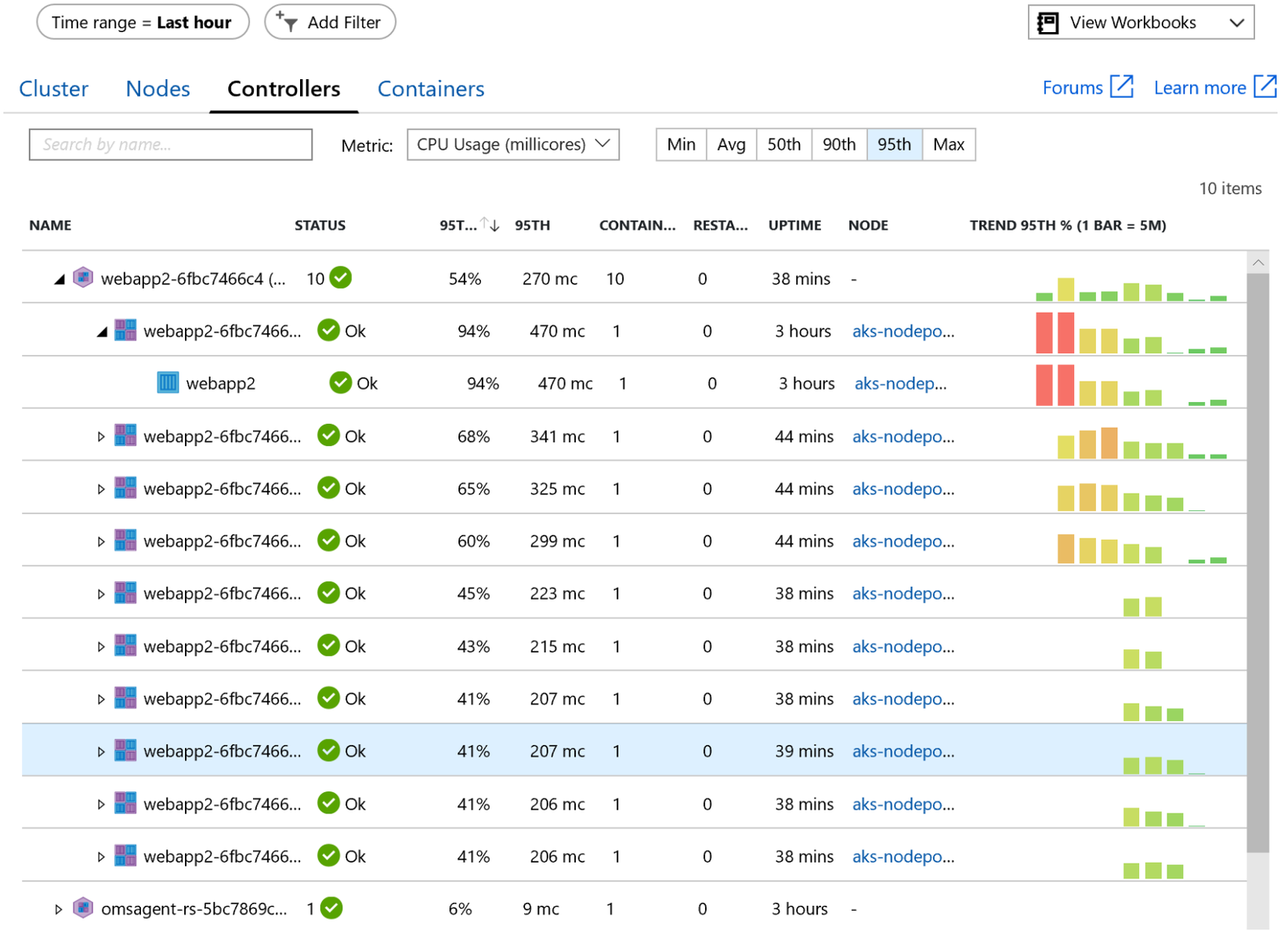


Azure Monitor for Containers – Nodes Insights

From the Insights blade, you can select and see directly the metrics for each of the nodes, controllers and containers.



Azure Monitor for Containers – Controllers and Containers Insights



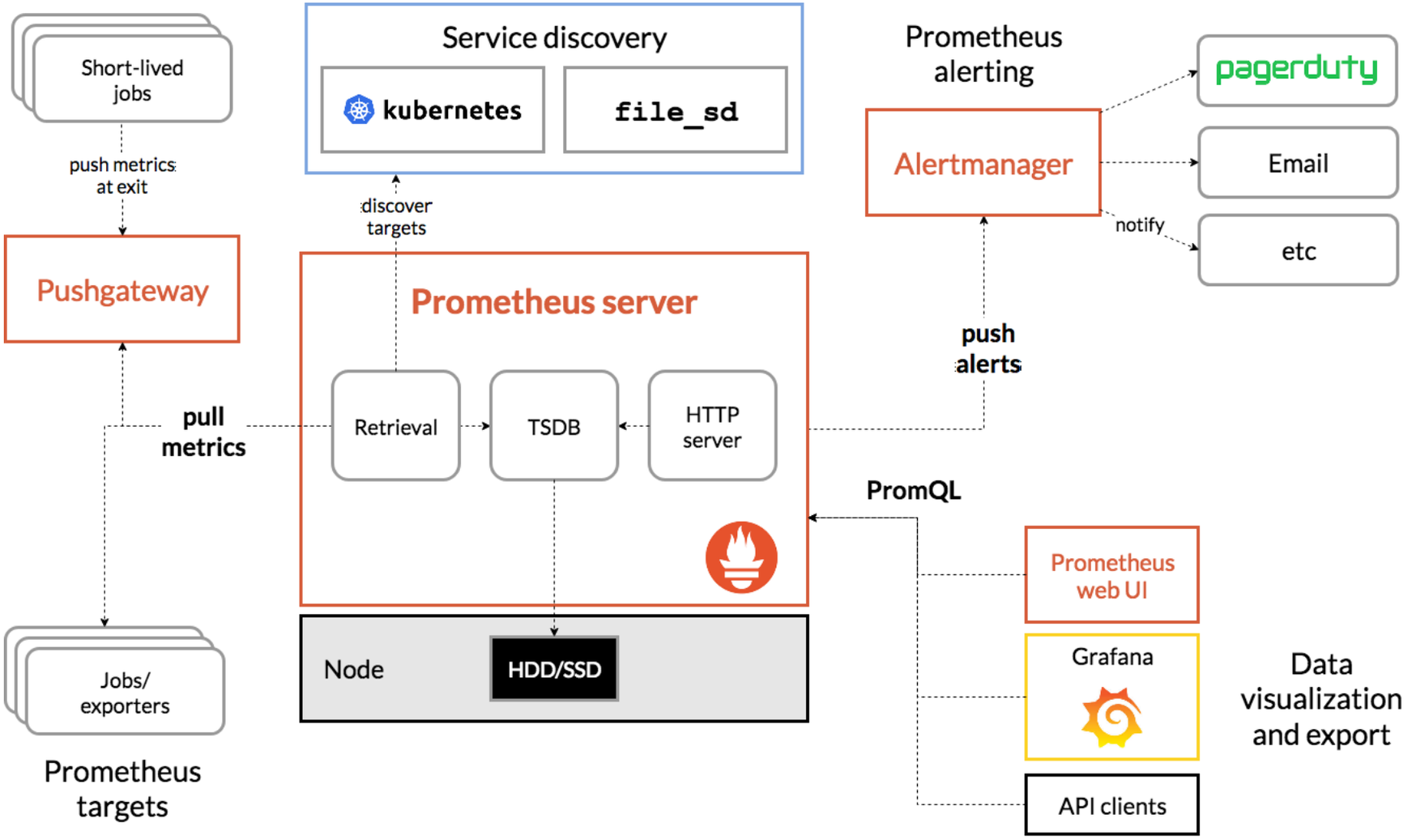
Kubernetes Master Node Logs

- Master components such as kube-apiserver and kube-controller-manager provided as a managed service
- Logs are enabled via the portal and can be sent to a Logs Analytics workspace
- Can be analyzed by a different platform by sending diagnostic logs to an Azure storage account or event hub

Prometheus & Grafana

Prometheus is a popular open source metric monitoring provides extensive metrics on Kubernetes.

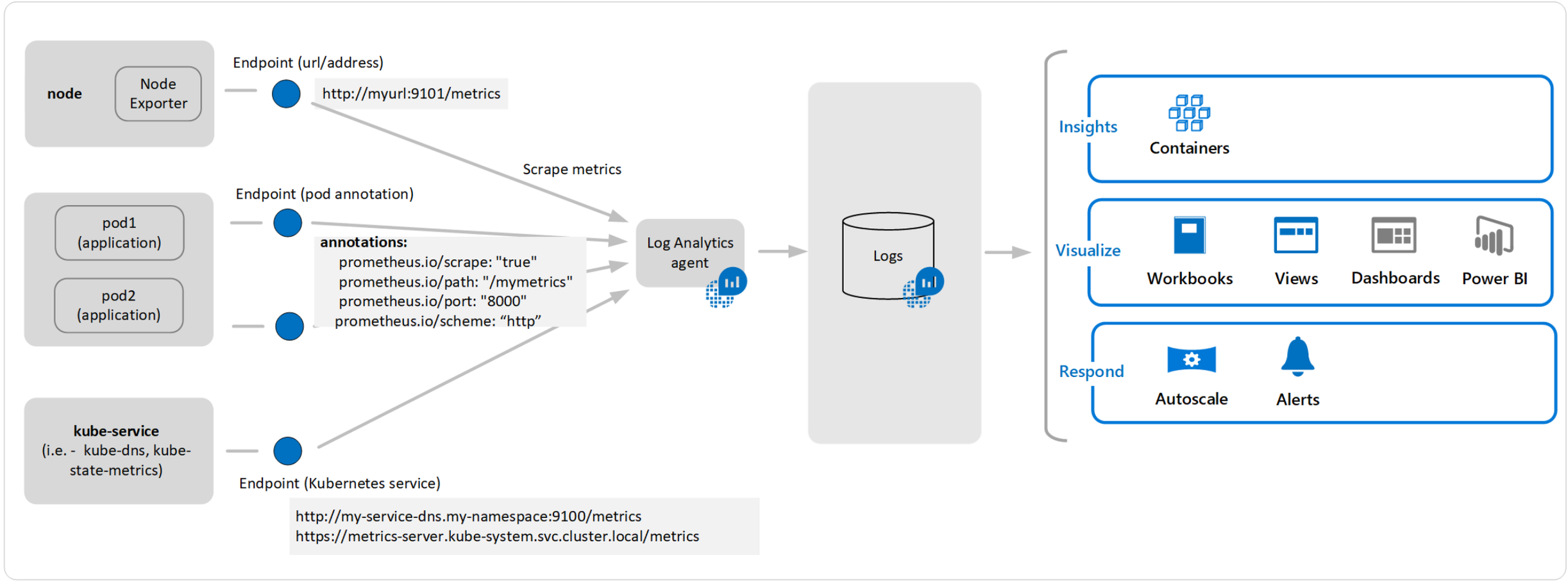
Grafana allows you to query and visualize alerts to help understand your metrics.



Azure Monitor for Containers with Prometheus

- Provides out of the box telemetry at the platform, container, orchestrator levels and to an extent the workload level
- Full stack, end to end monitoring view for Azure Kubernetes Services (AKS) in Azure Monitor for containers
- No Prometheus server is needed, just expose the Prometheus end-point through your exporters or pods (application), and the containerized agent for Azure Monitor for containers can scrape (collect) the metrics for you.

Azure Monitor for Containers with Prometheus



Microsoft

Containers Solutions Monitoring

Azure Monitoring for Azure Service Fabric

Microsoft Services

Service Fabric Dashboard

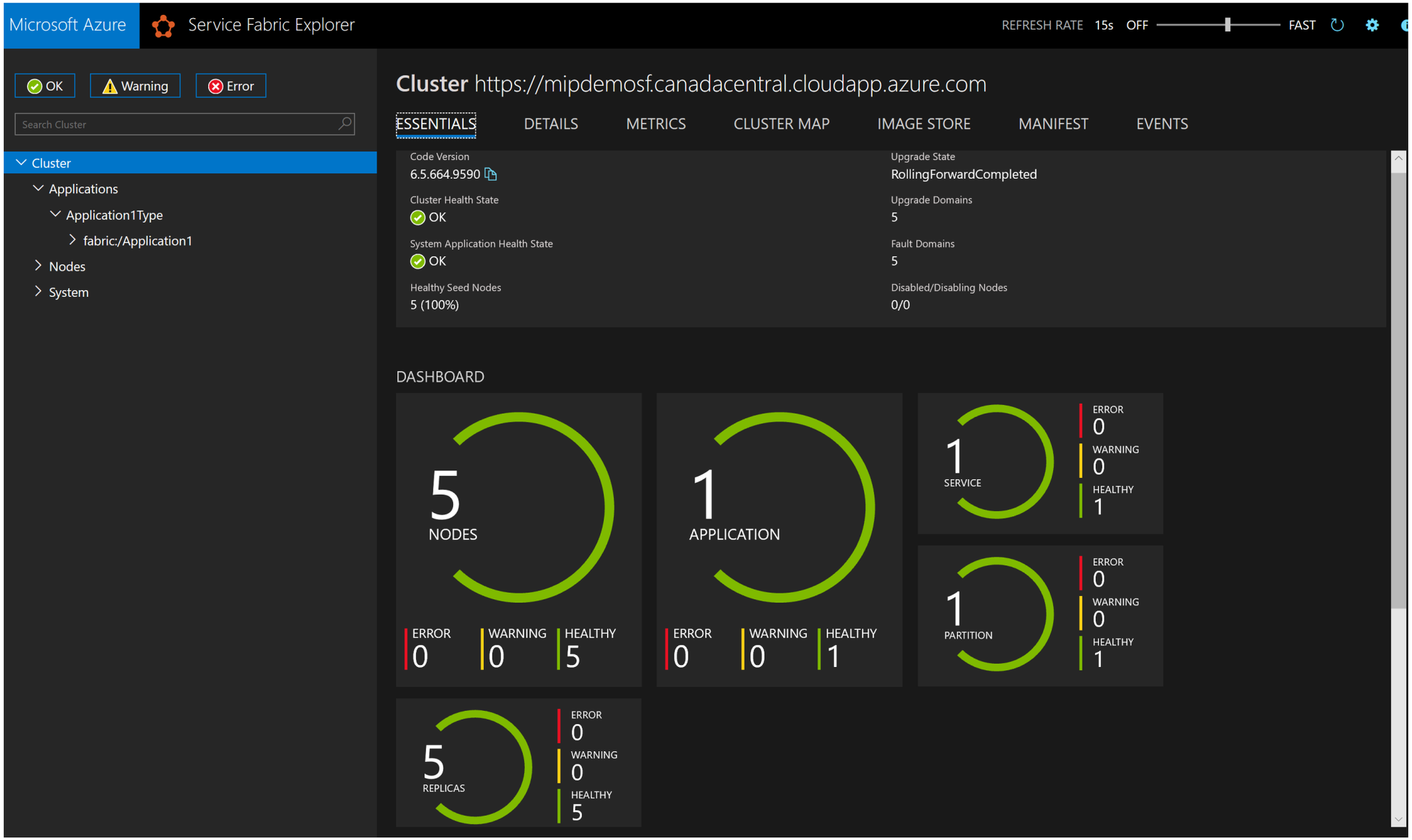
Like Kubernetes, Service Fabric provides a basic dashboard which gives you an overview of your clusters and applications health.

The dashboard, called Service Fabric Explorer, is available through web or via a standalone application.

You can:

- View the cluster's layout
- View applications and services
- View the cluster's nodes
- View events
- Create and delete applications

Service Fabric Dashboard - Overview



Azure Monitor - Service Fabric Analytics

- Microsoft recommends to monitor cluster level events through Azure Monitor logs
- Diagnostics must be enabled to view cluster-level or platform-level events, achieved by installing the Log Analytics agent on your nodes
- Configure the cluster to send data to a Log Analytics workspace, by creating a Service Fabric Analytics from the marketplace
- From the Service Fabric Analytics solution, you can change solution settings, workspace settings and access the Log Analytics workspace

Azure Monitor - Container Monitoring Solution

To monitor the containers on the cluster you need to add another solution called Container Monitoring Solution to the Log Analytics workspace.

The agent enables the collection of several container-specific logs that can be queried in Azure Monitor logs or used to visualize performance indicators.

The log types that are collected are:

- **ContainerInventory**: shows information about container location, name, and images
- **ContainerImageInventory**: information about deployed images, including IDs or sizes
- **ContainerLog**: specific error logs, docker logs (stdout, etc.), and other entries
- **ContainerServiceLog**: docker daemon commands that have been run
- **Perf**: performance counters including container cpu, memory, network traffic, disk i/o and custom metrics from the host machines

Service Fabric Analytics – Solution Overview

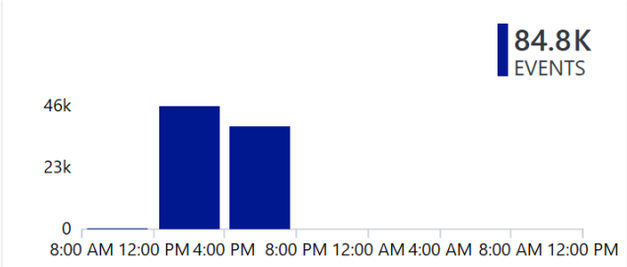
ServiceFabric(MIPDemoSFLA-LAW)

MIPDemoSFLA-LAW

Refresh | Solution Settings | Logs

10/5/19 10:57 - 10/6/19 10:57

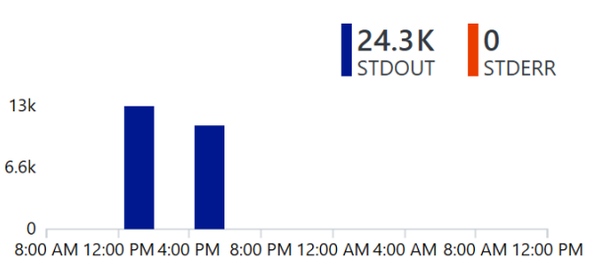
CLUSTER EVENTS



NODE	COUNT
_mipsfla_0	18.2K
_mipsfla_3	17.3K
_mipsfla_2	17.9K
_mipsfla_4	14.1K
_mipsfla_1	17.2K

See all...

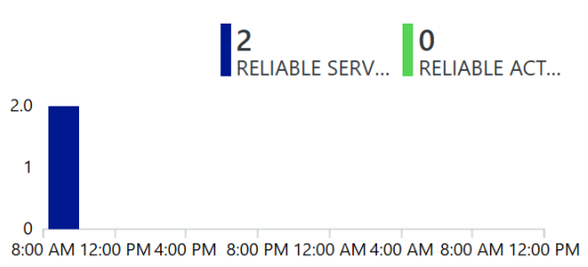
CONTAINER LOGS



NODE	NAME	EVENTS
mipsfla000001	sf-0-5722f9c5...	3
mipsfla000003	sf-0-e9803c3d...	6
mipsfla000003	sf-0-e83ad17...	3
mipsfla000003	sf-0-6569259...	3
mipsfla000003	sf-0-b045c94a...	3
mipsfla000000	sf-0-c2434cce...	3
mipsfla000000	sf-0-b9833ac2...	3
mipsfla000000	sf-0-bde04e9...	3
mipsfla000002	sf-0-ef7cab86...	3
mipsfla000002	sf-0-b1b2abc...	3

See all...

APPLICATION EVENTS

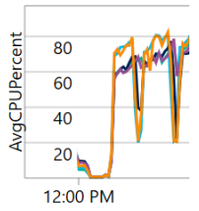


NODE	COUNT
fabric:/System	2

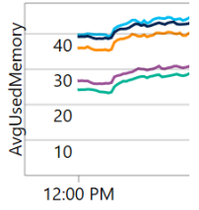
See all...

NODE METRICS

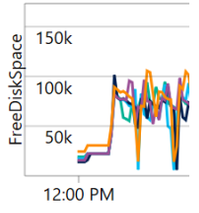
CPU (%)



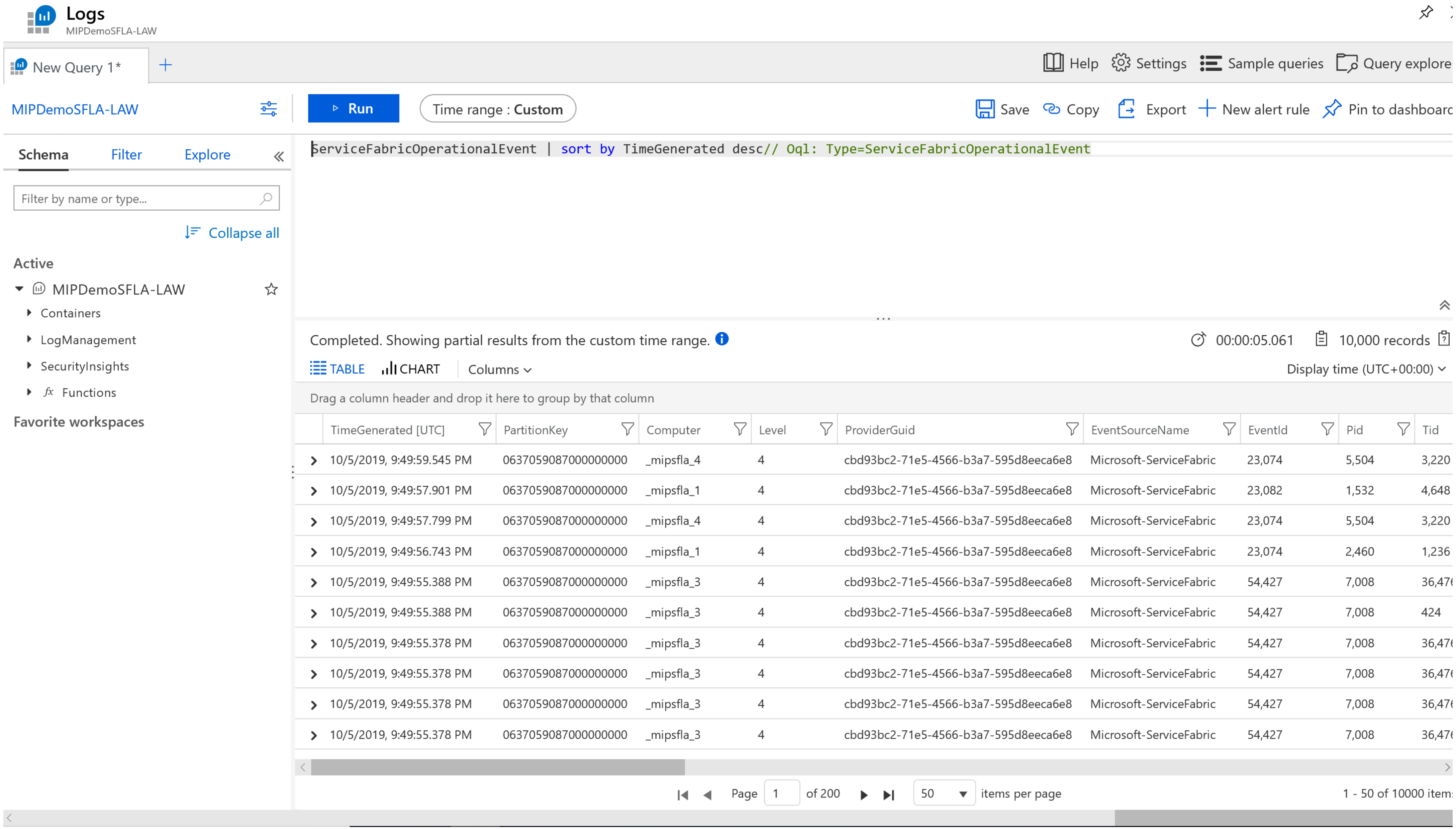
Memory (%)



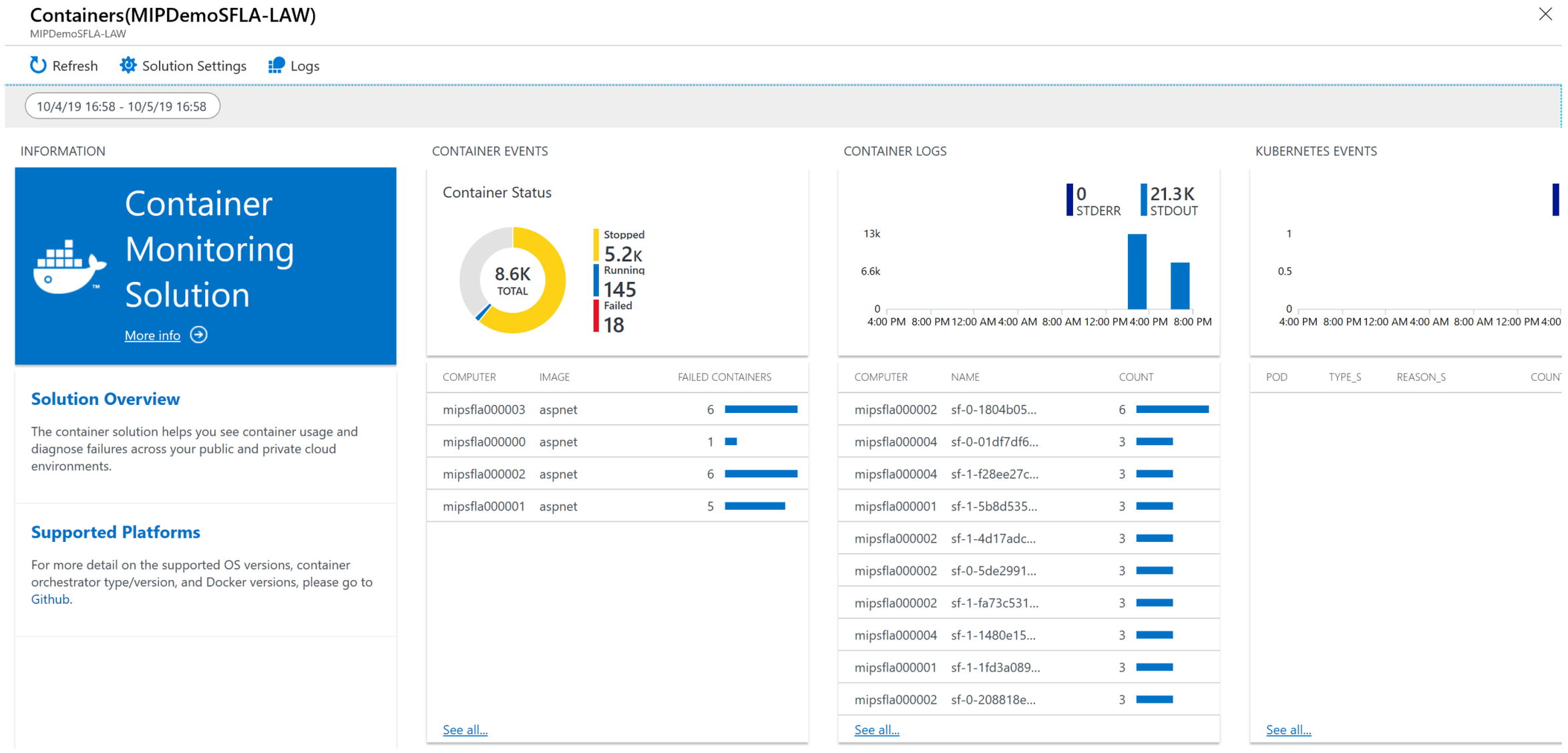
Disk usage (MB)



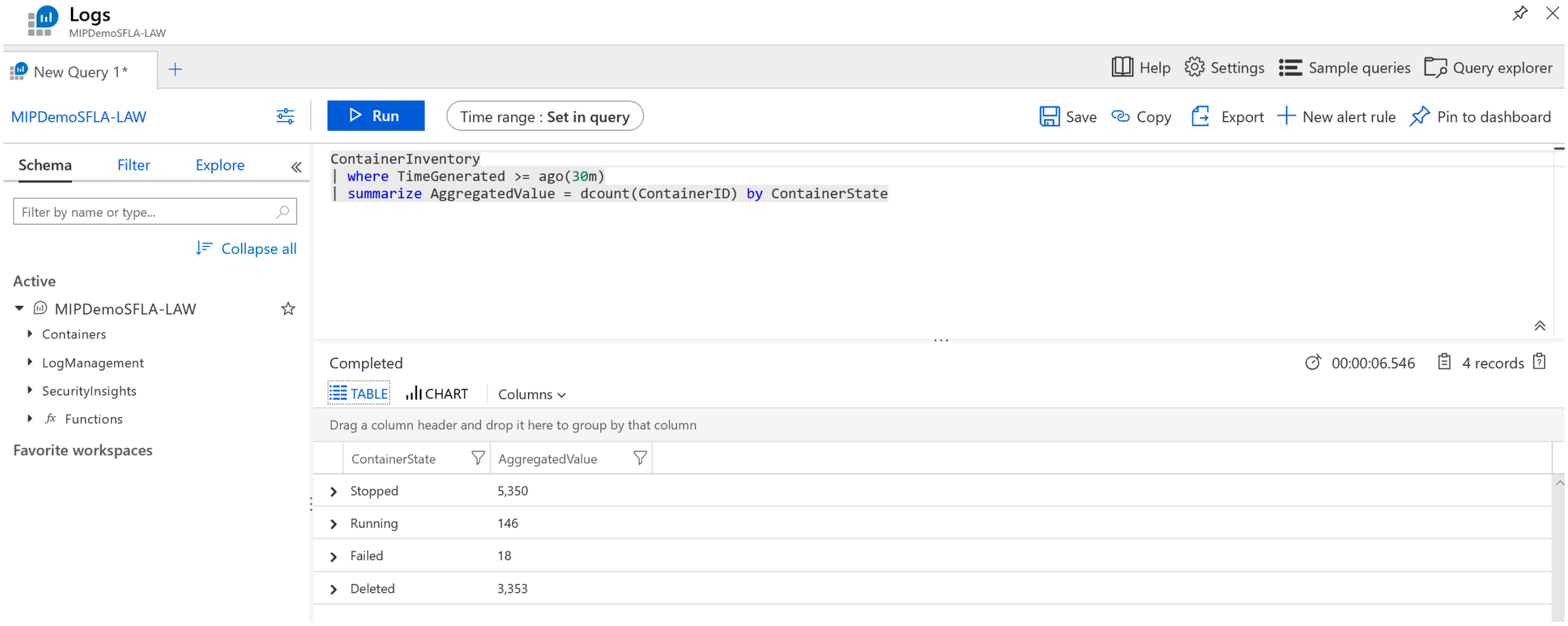
Service Fabric Analytics – Query



Container Monitoring – Solution Overview



Container Monitoring – Query



Microsoft

Containers Solutions Monitoring

Best Practices for Customers' Security and Management

Microsoft Services

Security – Common Best Practices

- **Use a private registry :** A publicly available container image does not guarantee security. You should store and retrieve images from a private registry, such as Azure Container Registry or Docker Trusted Registry.

- **Monitor and scan container images :** Security monitoring and scanning solutions such as Twistlock and Aqua Security are available through the Azure Marketplace.

- **Protect credentials :** Containers can spread across several clusters and Azure regions. Make sure to secure credentials required for logins or API access, such as passwords or tokens.

Security – Common Best Practices

Image and container level security

- AAD authenticated Container registry access
- ACR image scanning and content trust for image validation

Node and cluster level security

- Automatic security patching nightly
- Nodes deployed in private virtual network subnet w/o public addresses
- Network policy to secure communication paths between namespaces (and nodes)
- Pod Security Policies
- K8s RBAC and AAD for authentication

Security – Common Best Practices

Pod level security

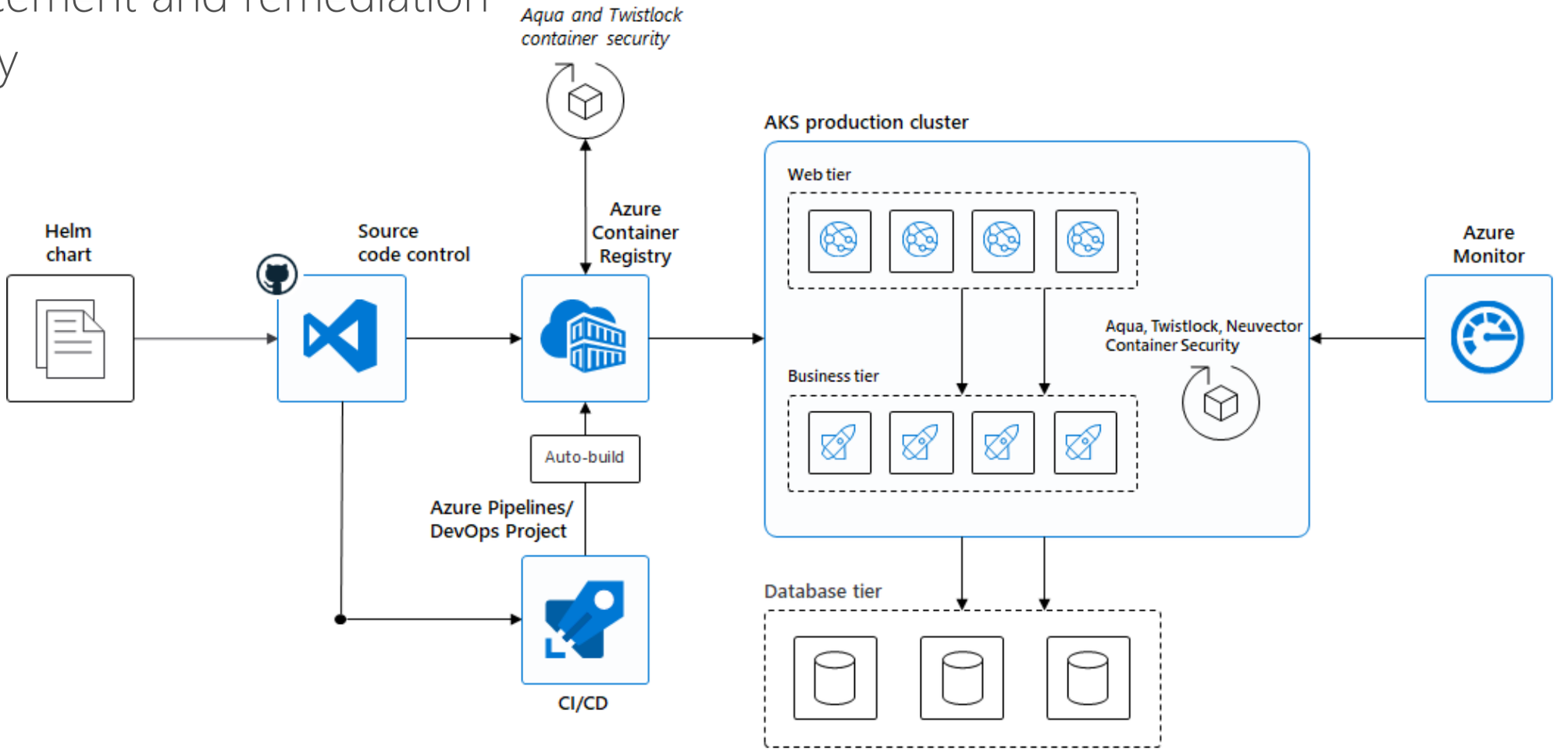
- Pod level control using AAD Pod Identity
- Pod Security Context

Workload level security

- Azure Role-based Access Control (RBAC) & security policy groups
- Secure access to resources & services (e.g. Azure Key Vault) via Pod Identity
- Storage Encryption
- App Gateway with WAF to protect against threats and intrusions

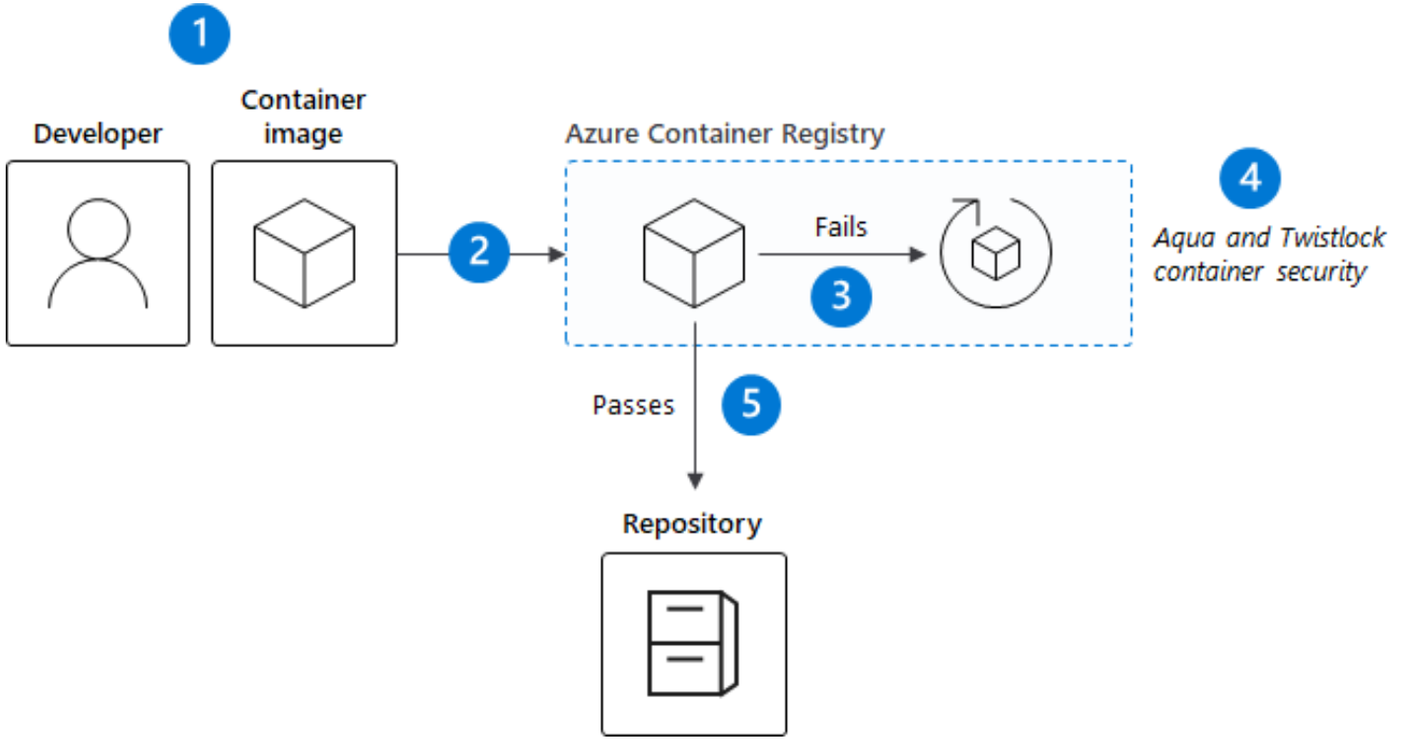
Container Level – Securing Images and Runtime

- Regularly apply security updates to the container images
- Scan your images, scan your containers
- Runtime enforcement and remediation
- Trusted Registry



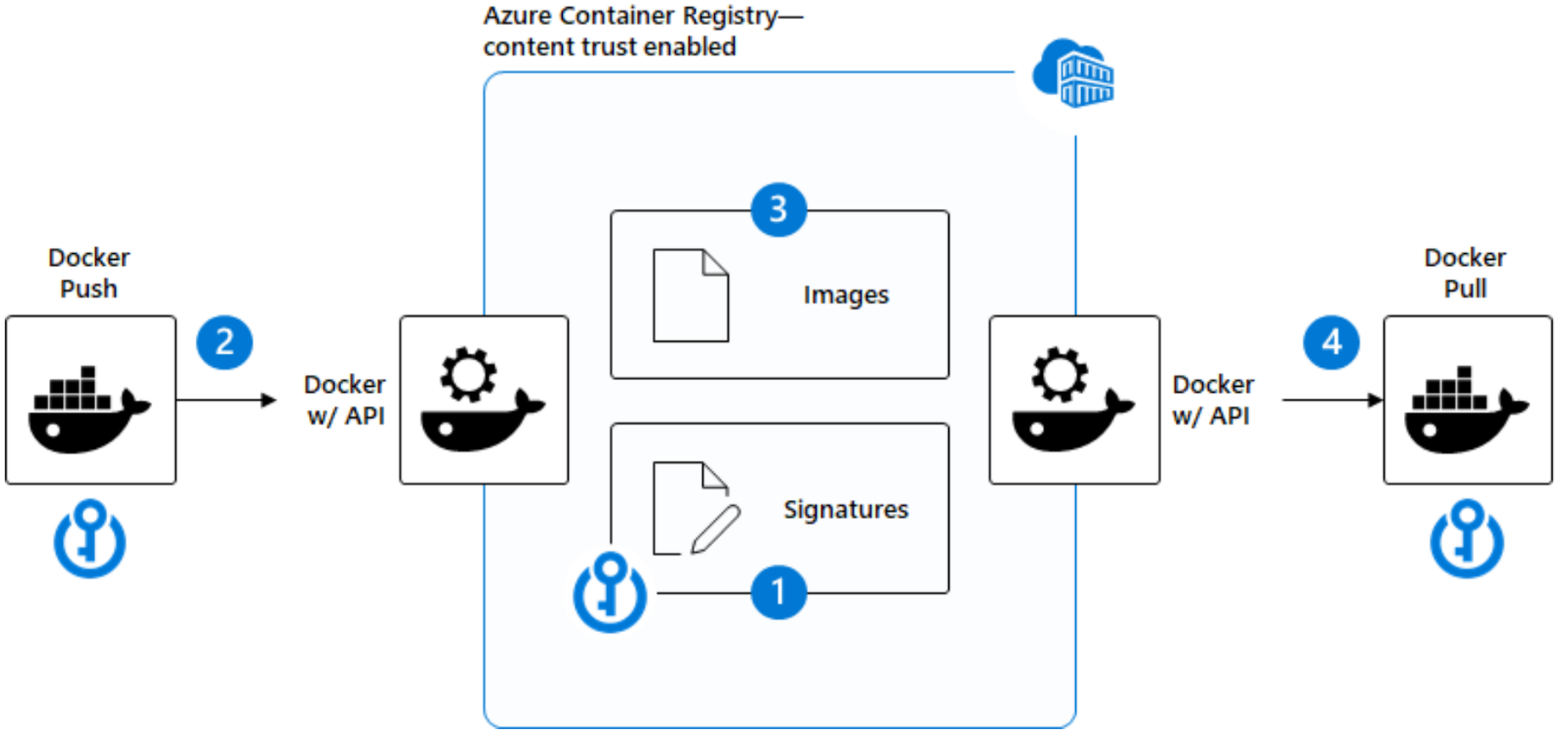
Azure Container Registry - vulnerability scanning

1. Developer/CI system builds container image
2. Image pushed to Azure Container Registry
3. Azure Container Registry quarantines image until scanning passes
4. Azure Container Registry scans content leveraging Aqua, Twistlock
5. Azure Container Registry publishes the image to the repository

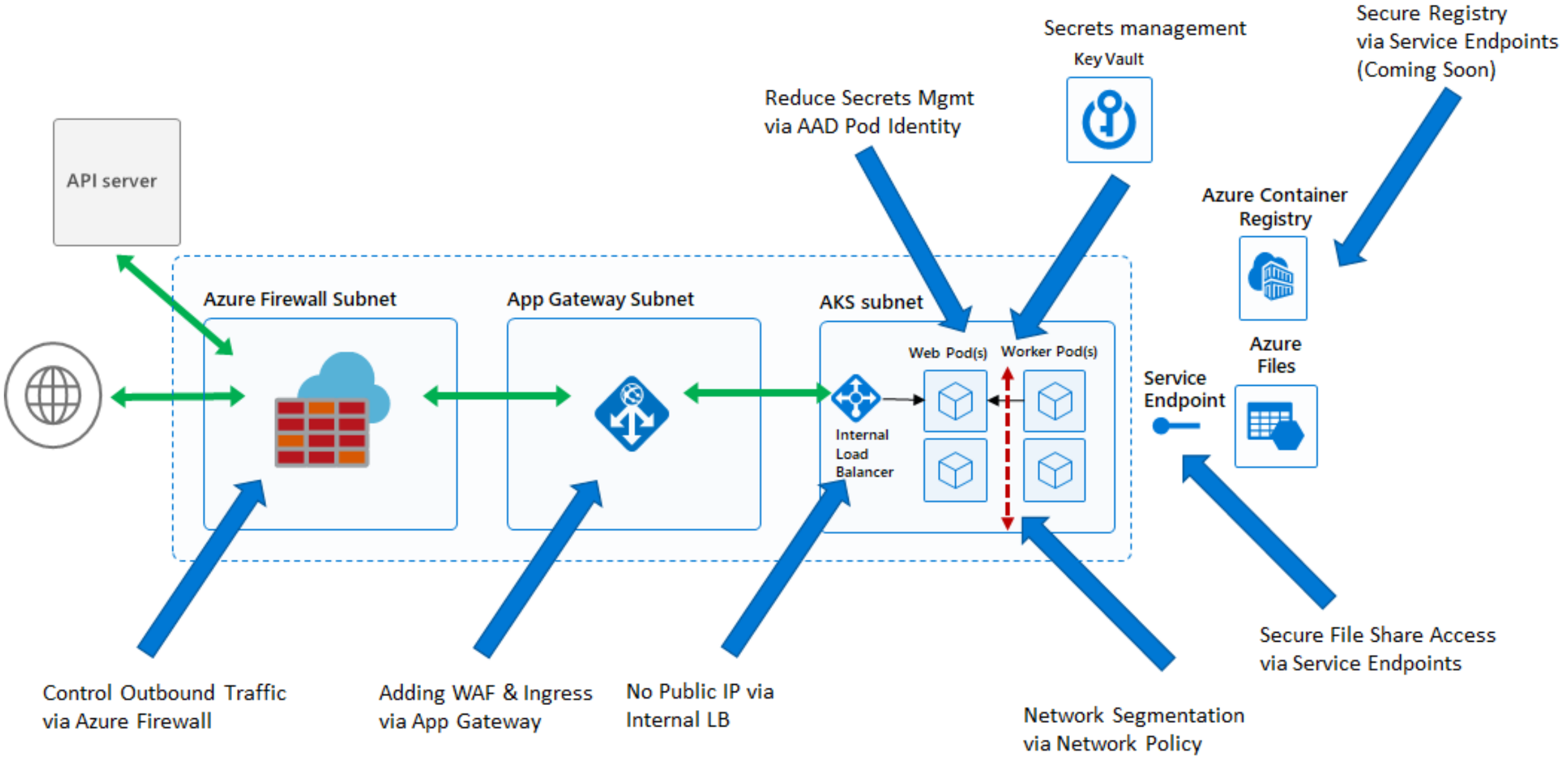


ACR - securing images from source to destination

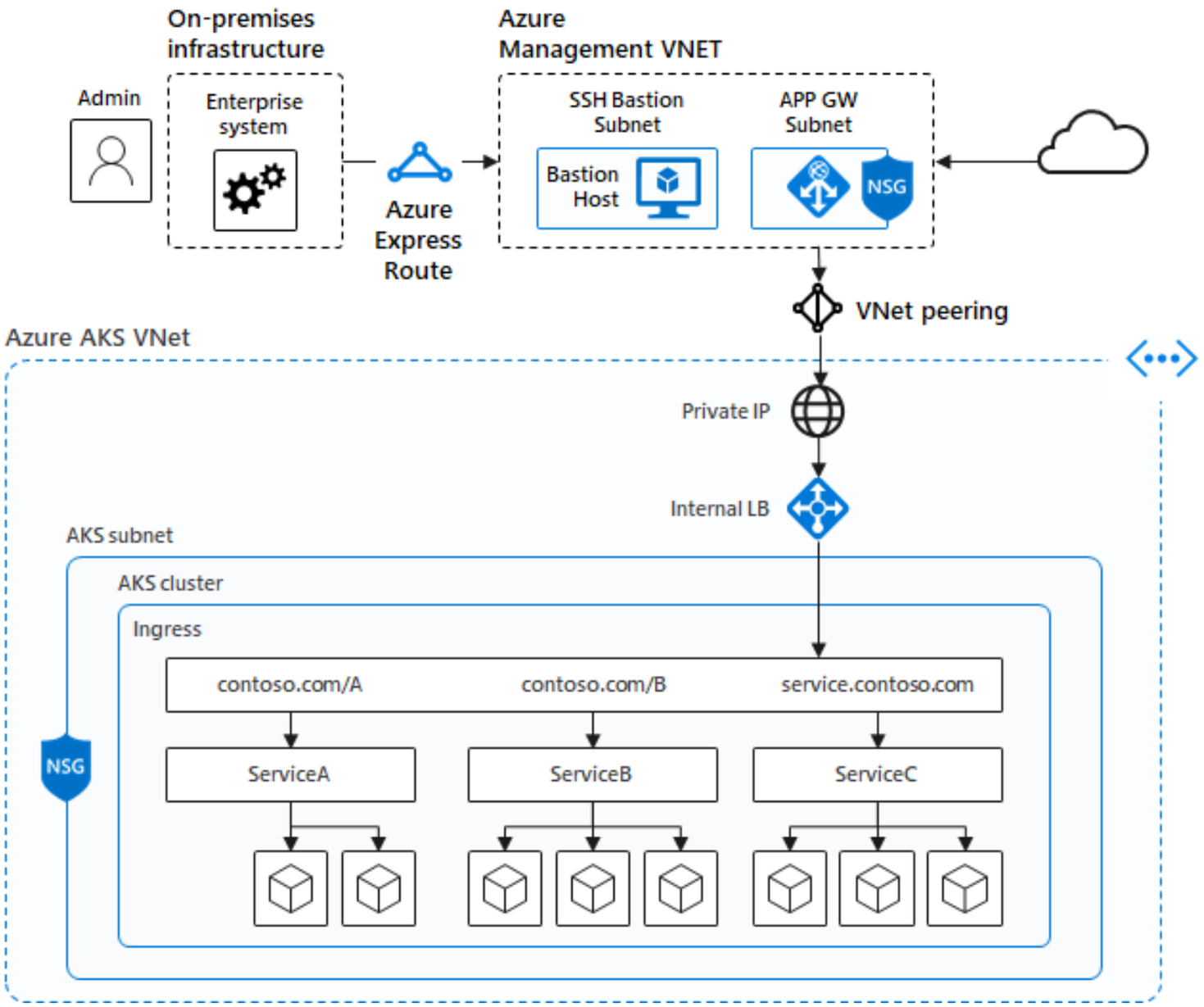
1. A set of cryptographic signing keys are associated with ACR and used for image signing
2. Image publisher signs the image and pushes to the ACR
3. Signed image is stored in Azure Container Registry
4. When an image consumer pulls a signed image, their Docker client verifies the integrity of the image



Basic example of AKS end to end security



Cluster Management Through Bastion Host



Microsoft

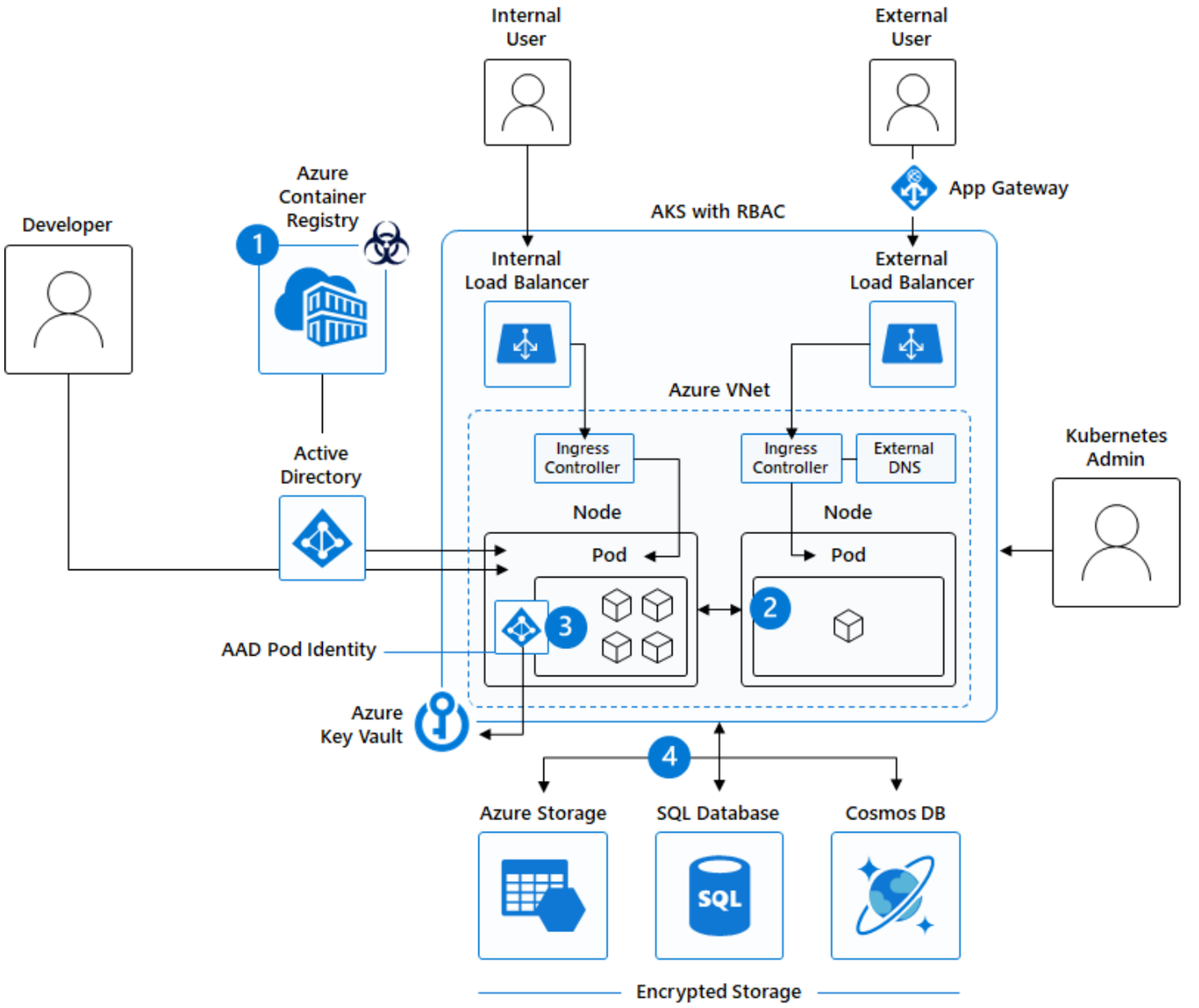
Containers Solutions Monitoring

Kubernetes Security Overview

Microsoft Services

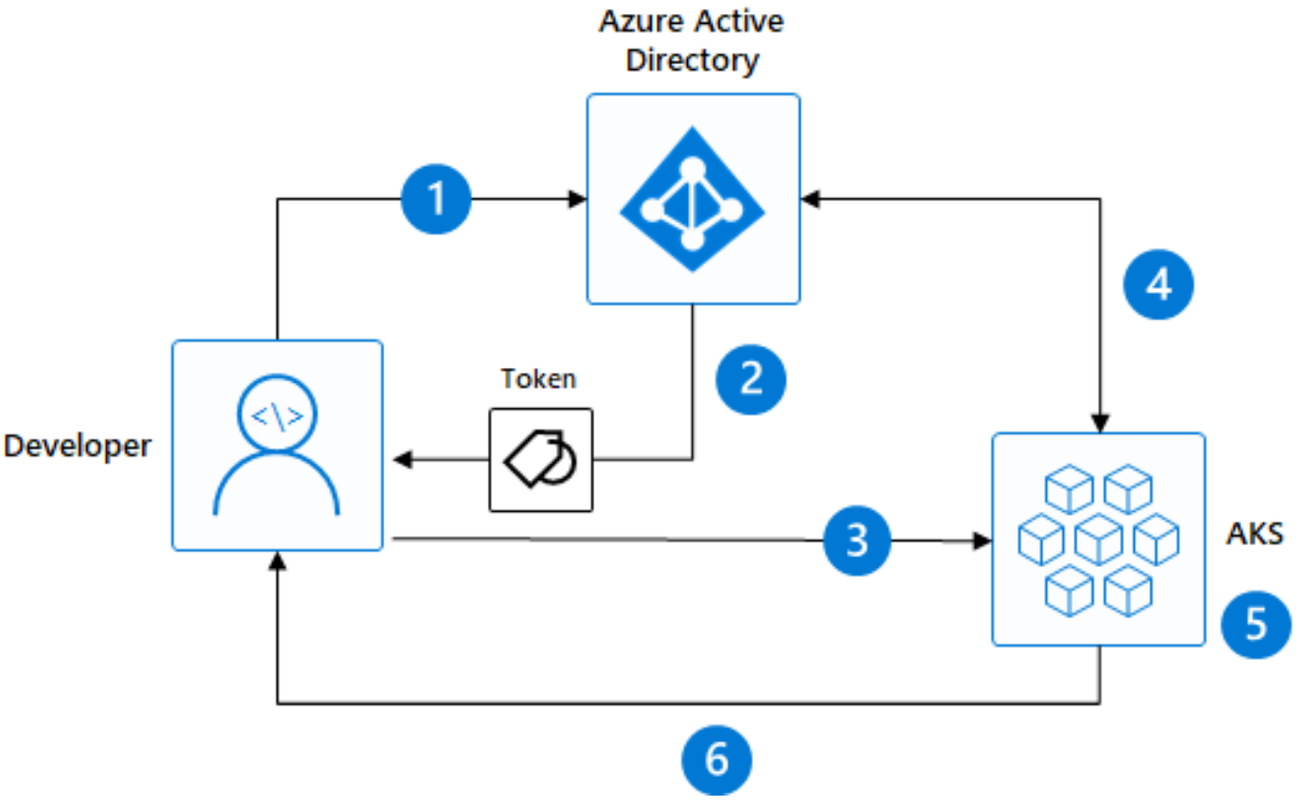
Security overview

1. Image and container level security
2. Node and cluster level security
3. Pod level security
4. Workload level security



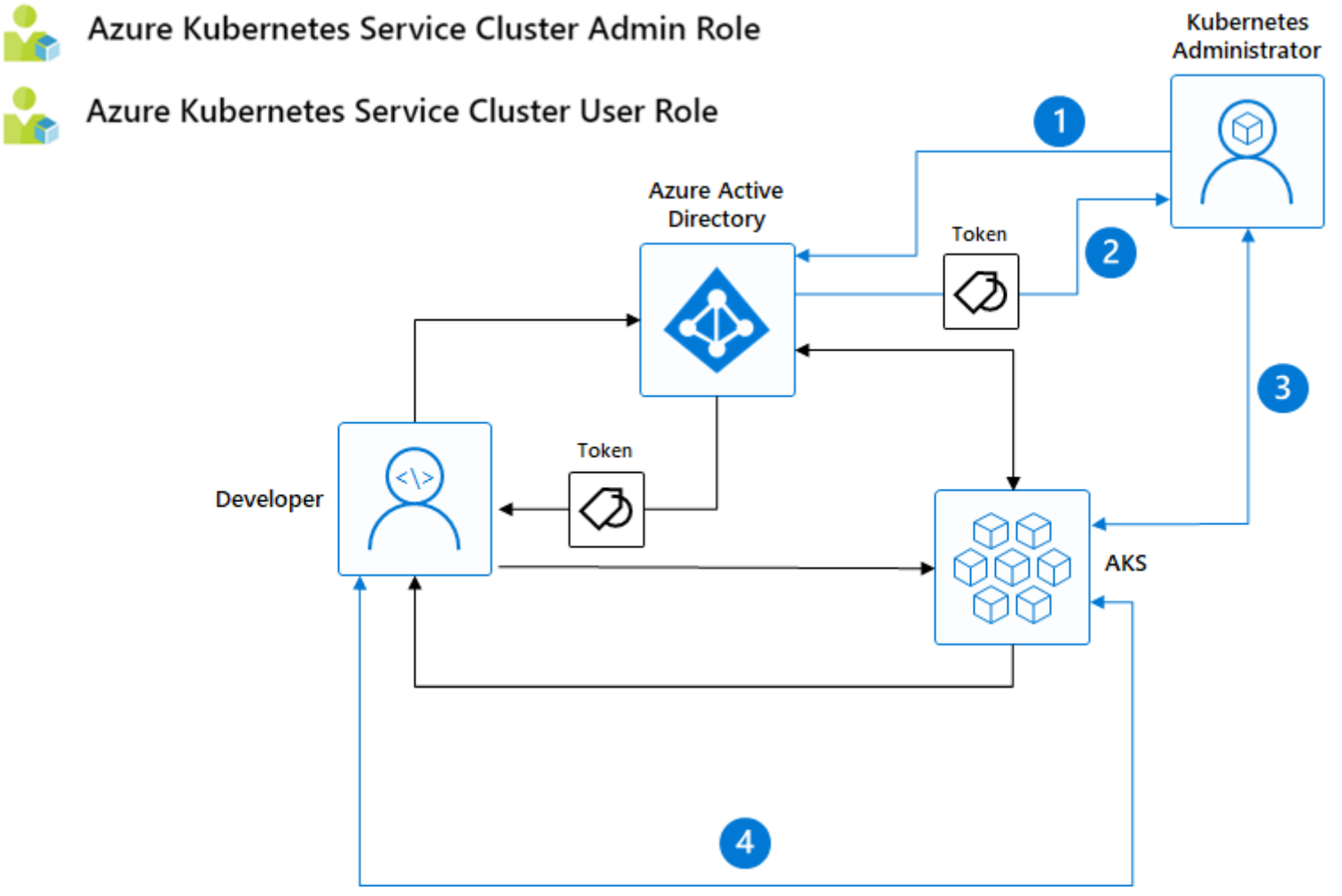
Cluster Level – RBAC and Azure AD integration

1. Kubernetes Developer authenticates with AAD
2. The AAD token issuance endpoint issues the access token
3. Developer performs action w/ AAD token.
   Eg. kubectl create pod
4. Kubernetes validates token with AAD and fetches the Developer's AAD Groups
   Eg. Dev Team A, App Group B
5. Kubernetes RBAC and cluster policies are applied
6. Request is successful or not based on the previous validation



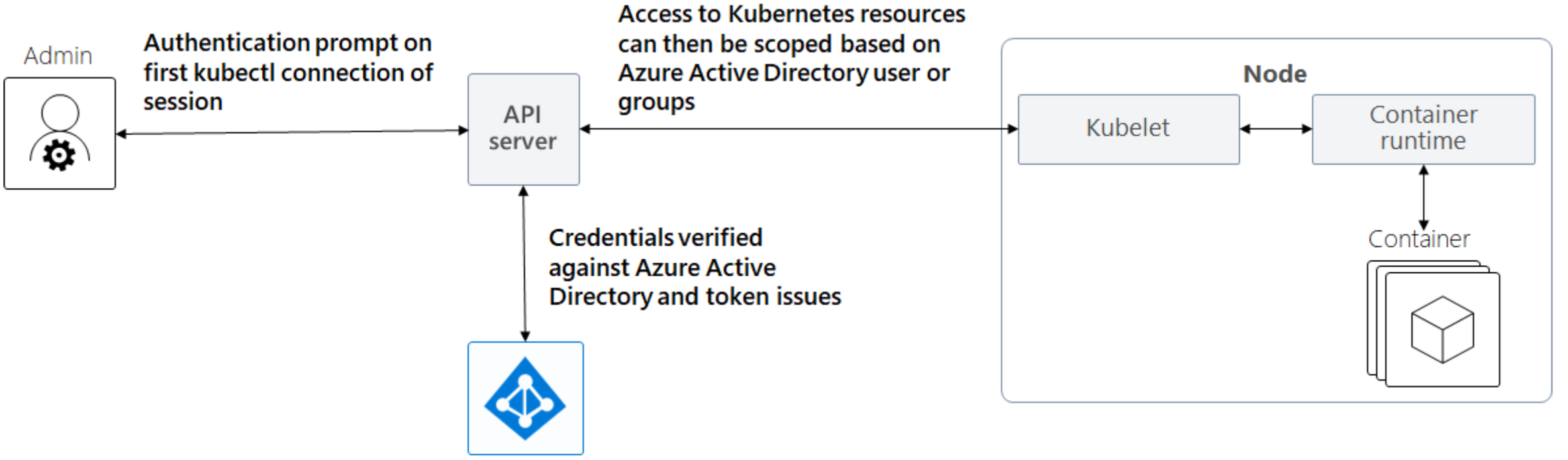
Azure Level - Kubernetes RBAC and Azure RBAC Roles

1. Kubernetes Administrator authenticates with AAD
2. The AAD token issuance endpoint issues the access token
3. Administrator fetches the admin kubeconfig and configures RBAC roles and bindings
4. Kubernetes Developer fetches the user kubeconfig



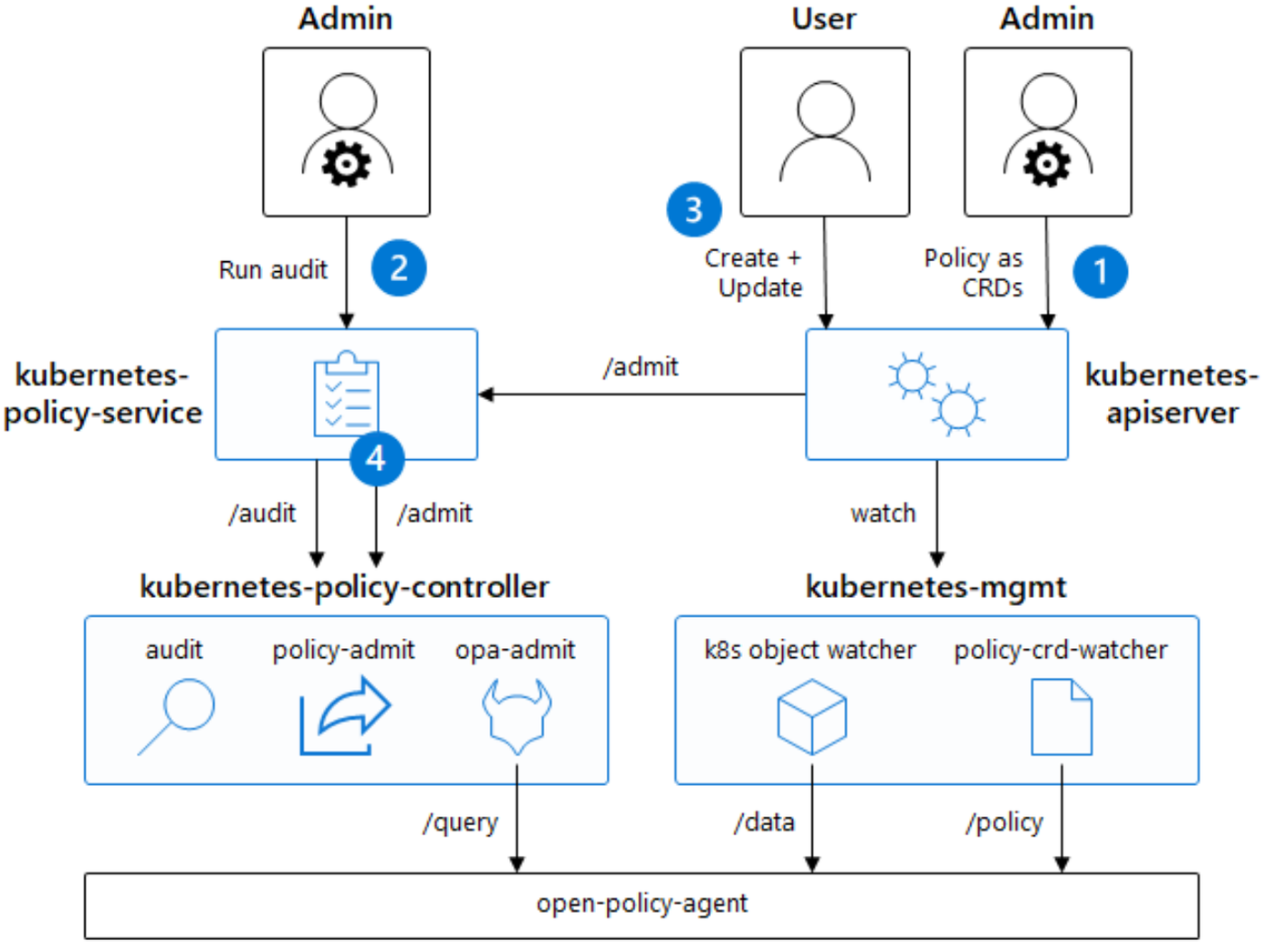
Cluster Level Security

- Securing endpoints for API server and cluster nodes
  - Integrate Kubernetes role-based access control (RBAC) with Azure Active Directory to control access to the API server. These controls let you secure AKS the same way that you secure access to your Azure subscriptions.



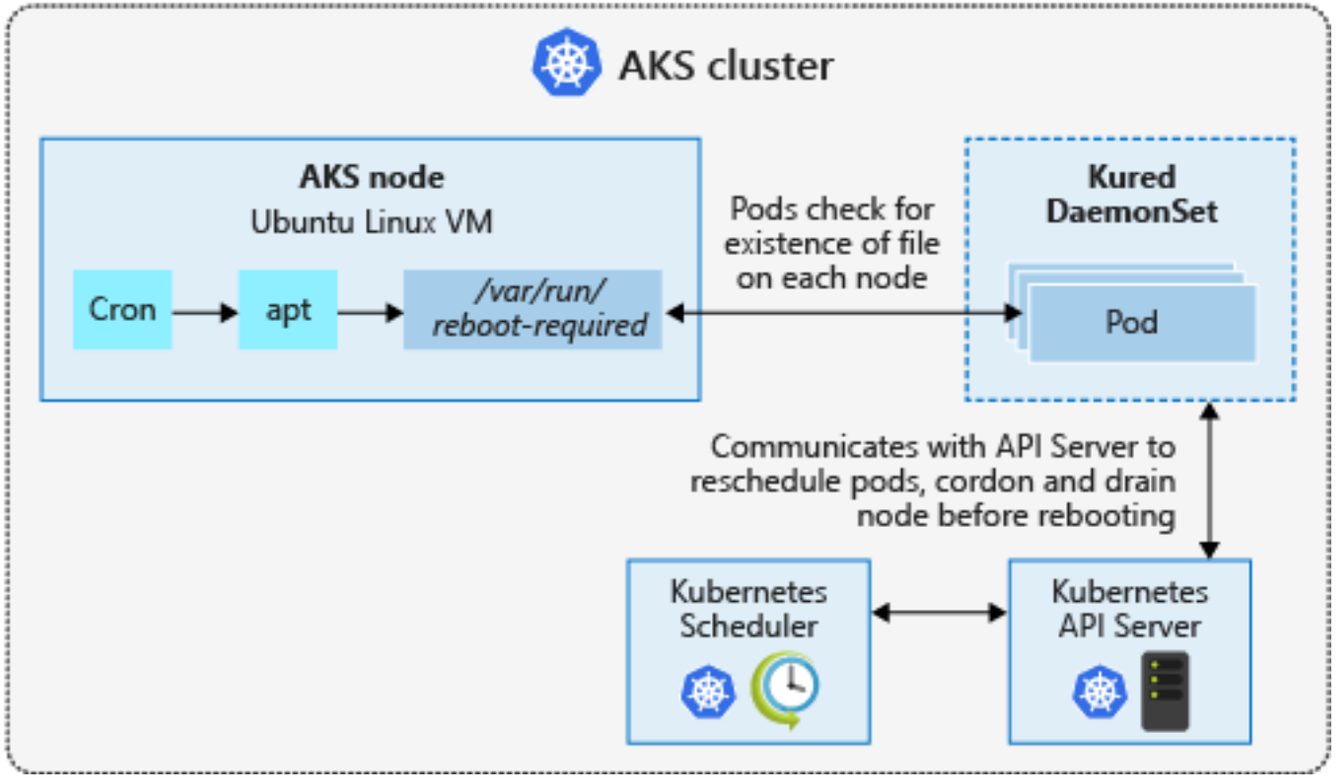
Policy controller

1. Admin adds policy for the cluster
2. Admin audits compliance of the cluster using /audit endpoint
3. User uses standard Kubernetes API to operate the cluster and the create actions are guarded by policy
4. Kubernetes-policy-controller provides an admission controller webhook that performs evaluations by calling open-policy-agent (OPA) service



Process Node OS update

- AKS automatically applies security patches to the nodes on a nightly schedule
- You're responsible to reboot as required
- Regular maintenance recommended
  - Monthly ideal, 3 months minimum
- Use Kured DaemonSet to watch for nodes requiring a reboot and handle the reschedule of the pods and reboot of the nodes.



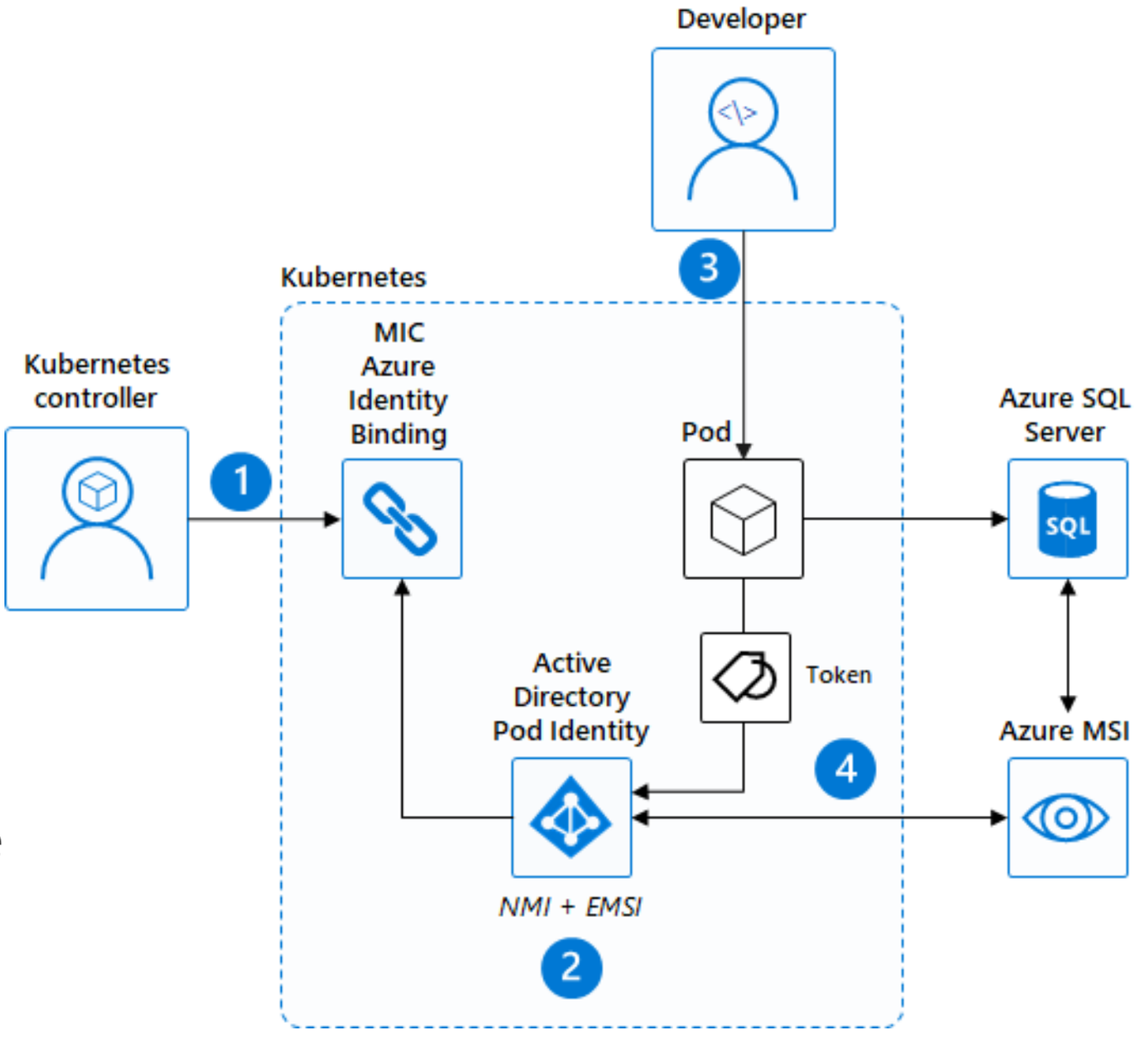
Securing Workloads - Pod Identity

1. Kubernetes operator defines an identity map for K8s service accounts

2. Node Managed Identity (NMI) watches for mapping reaction and syncs to Managed Service Identity (MSI)

3. Developer creates a pod with a service account. Pod uses standard Azure SDK to fetch a token bound to MSI

4. Pod uses access token to consume other Azure services; services validate token



Pod Security Policies

- Pod Security Policies (or PSPs) are objects that control security-sensitive aspects of pod specification (like root privileges)
- If a pod does not meet the conditions specified in the PSP, Kubernetes will not allow it to start
- You can assign PSPs at the cluster or project level
- PSPs work through inheritance
- Workloads that are already running in a cluster or project before a PSP is assigned will not be checked

Microsoft

Containers Solutions Monitoring

Azure Service Fabric Security Overview

Microsoft Services

Securing your Clusters

Always use a secure cluster:

- Implement cluster security by using certificates
- Provide client access (admin and read-only) by using Azure Active Directory (Azure AD)

Use automated deployments:

- Use scripts to generate, deploy and roll over secrets
- Store the secrets in Azure Key Vault and use Azure AD for all other client access
- Require authentication for human access to secrets

Security is only configured at cluster creation time. It's not possible to turn on security after the cluster is created.

Securing your Clusters - Scenarios

There are three scenarios for implementing cluster security:

- **Node-to-node security:** This scenario secures communication between the VMs and the computers in the cluster.
- **Client-to-node security**: This scenario secures communication between a Service Fabric client and the individual nodes in the cluster.
- **Role-Based Access Control (RBAC):** This scenario uses separate identities (certificates, Azure AD, and so on) for each administrator and user client role that accesses the cluster.

Node-to-node Security

- Helps secure communication between the VMs or computers in a cluster
- Ensures that only computers that are authorized to join the cluster can participate in hosting applications and services in the cluster
- Clusters running on Azure and standalone clusters running on Windows can use either certificate security or Windows security for Windows Server computers
- Service Fabric uses X.509 server certificates that are specified as part of the node-type configuration during cluster creation

Client-to-node Security

- Authenticates clients and helps secure communication between a client and individual nodes in the cluster
- Ensures that only authorized users can access the cluster and the applications can only be deployed on the cluster through either Windows or certificate security credentials
- For clusters running on Azure, can secure access to management endpoints by using Azure Active Directory
- This scenario also uses X.509 server certificates

Role-Based Access Control (RBAC)

Use access control to limit access to certain cluster operations for different groups of users.

Two access control types are supported for clients that connect to a cluster:

- **Administrator roles** have full access to management capabilities, including read and write capabilities.
- **User roles** have only read access to management capabilities (for example, query capabilities). They also can resolve applications and services

X.509 Certificates

Service Fabric uses X.509 certificates to secure a cluster and provide application security features.

Consider:

- For clusters that are running production workloads, only use a correctly configured Windows Server certificate service, or one from an approved certificate authority (CA).
- Only use self-signed certificates in test clusters, never production.
- When generating the certificate thumbprint, be sure to generate a SHA1 thumbprint. SHA1 is used when configuring the Client and Cluster certificate thumbprints.

X.509 Certificates Requirements

For cluster and server, the certificate must meet the following requirements:

- It must contain a private key (extensions .pfx or .pem)
- It must be created for key exchange, exportable to a Personal Information Exchange (.pfx) file
- The certificate's subject name must match the domain that you use to access the Service Fabric cluster. This matching is required to provide an SSL for the cluster's HTTPS management endpoint and Service Fabric Explorer. You must obtain a custom domain name for your cluster.

X.509 Certificates Rotation

Service fabric lets you specify two cluster certificates, a primary and a secondary, when you configure certificate security during cluster creation.

You will always need at least one valid (not revoked and not expired) cluster certificate deployed

If both primary and secondary certificates are installed, Service Fabric will pick the one with an expiring date further into the future and automatically rollover.

Other Considerations

Windows security baselines

- It is recommended to apply an industry standard baseline to the nodes.

Windows Defender

- Windows Defender antivirus is installed on Windows Server 2016. By default the Service Fabric processes and path are not excluded.

Platform Isolation

- Service Fabric applications are granted access to the Service Fabric runtime itself, in the eventuality that the service hosts itself untrusted code, it is advisable to disable this access to the SF runtime.

TLS 1.2

- It is highly recommended that TLS 1.2 is configured at the machine level or application level

Knowledge Check

1. What are the two categories of monitoring containers in Azure?
2. What are some of the benefits you get with Azure Monitor for containers with Prometheus?
3. What are the four levels of Kubernetes security?