

EKS 모니터링 꿀팁

여러분의 클러스터는 안녕하신가요?

Myungsup Kwak, Changjong Hyun, Sujeong Lee
AWS Support Engineering - Deployment



Series 소개



Series 소개

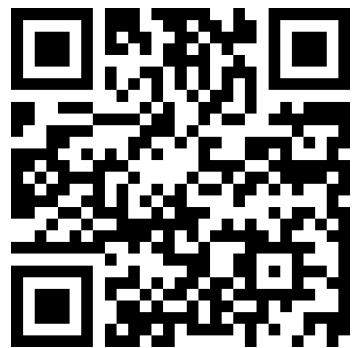
Date	Title	Speaker
9월 4일(목) 19:00	EKS 모니터링 꿀팁 : 여러분의 클러스터는 안녕하신가요?	곽명섭 이수정 현창종
9월 25일(목) 19:00	EKS에서 Karpenter 아무지게 활용하기	박동혁 이호성 황은빛
10월 말 예정 추후 밋업 공지	EKS 신규 기능으로 달달하게 운영하기 (Auto Mode, Auto Repair, Monitoring, ...)	김보영 박진희 하영



Agenda

- Why do we monitor EKS Cluster?
- How do we monitor EKS Cluster?
- Demo
- Advanced

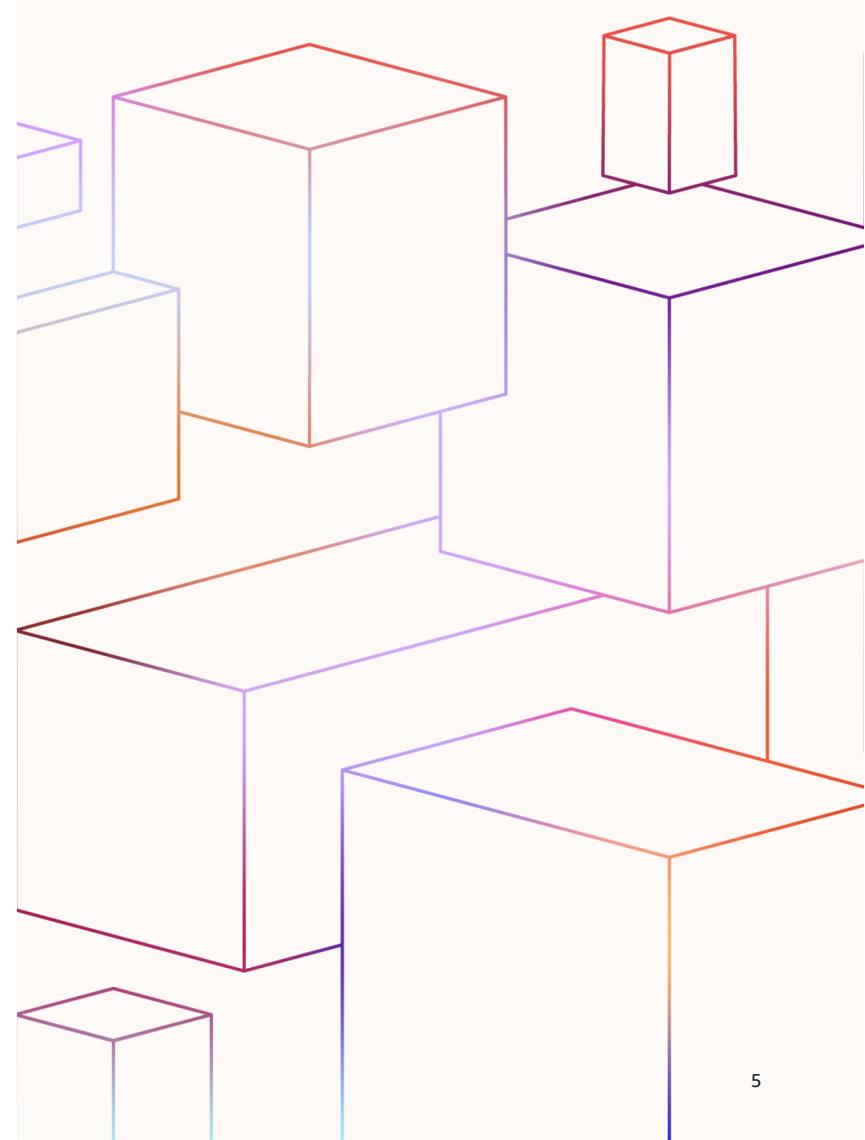
**Have a Question?
- Ask Anonymously**



slido.com #1490 347



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.



Why do we Monitor EKS Cluster?



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Why? - Unexpected termination of Pods

NAME	READY	STATUS	RESTARTS	AGE
amazon-cloudwatch-observability-controller-manager-5d58ccdv5gf	1/1	Running	0	2d3h
cloudwatch-agent-4c7j8	1/1	Running	0	9d
cloudwatch-agent-4dmdh	1/1	Running	0	20h
cloudwatch-agent-97h5d	1/1	Running	0	12h
cloudwatch-agent-n9567	1/1	Running	0	2d8h
cloudwatch-agent-t56qq	1/1	Running	0	8d
cloudwatch-agent-t6s2s	1/1	Running	0	2d3h
cloudwatch-nlb-exporter-7697df59f6-d4sdn	1/1	Running	0	22h
cwagent-prometheus-5ccc7bbc6f-hrcj4	1/1	Running	0	11h
fluent-bit-62cdh	1/1	Running	0	8d
fluent-bit-72wwt	1/1	Running	0	2d8h
fluent-bit-7cx9w	1/1	Running	0	2d3h
fluent-bit-8q6gq	1/1	Running	0	12h
fluent-bit-gx9zf	1/1	Running	0	9d
fluent-bit-hszpd	1/1	Running	0	20h
fluent-bit-tv5sn	1/1	Terminating	0	2d8h
fluent-bit-tv5sn	0/1	Completed	0	2d8h
fluent-bit-9m6lz	0/1	Pending	0	0s
fluent-bit-9m6lz	0/1	ContainerCreating	0	0s
fluent-bit-9m6lz	1/1	Running	0	2s



Why? – Lacking resource on Worker node

```
mskwak@c889f3d898ef yamlfiles % kubectl get pods -A -w
NAMESPACE      NAME                               READY   STATUS    RESTARTS   AGE
amazon-cloudwatch  amazon-cloudwatch-observability-controller-manager-8586fdctvc2g  1/1     Running   0          2d
amazon-cloudwatch  cloudwatch-agent-qslpq        1/1     Running   0          100m
amazon-cloudwatch  cloudwatch-agent-s6kkw        1/1     Running   0          100m
amazon-cloudwatch  fluent-bit-8jwlk           1/1     Running   0          100m
amazon-cloudwatch  fluent-bit-fpw4h           1/1     Running   0          100m
default          nginx-deployment-5485cb4898-ks9q5  0/1     Pending   0          115s
default          nginx-deployment-5485cb4898-rk2sf  1/1     Running   0          2m23s
default          nginx-deployment-64c4bbc7c6-7tpxb  1/1     Running   0          111s
kube-system      aws-node-lzlrn                  2/2     Running   0          100m
kube-system      aws-node-qvwl9                  2/2     Running   0          100m
kube-system      coredns-c844dd74d-cqnfb       1/1     Running   0          2d
kube-system      coredns-c844dd74d-zh2ck       1/1     Running   0          2d
kube-system      eks-node-monitoring-agent-dvtz5  1/1     Running   0          100m
kube-system      eks-node-monitoring-agent-jrtx6  1/1     Running   0          100m
kube-system      eks-pod-identity-agent-b69bp    1/1     Running   0          100m

Events:
Type      Reason     Age            From            Message
----      ----      --            ----            -----
Warning   FailedScheduling  42s           default-scheduler  0/2 nodes are available: 2 Insufficient cpu
No preemption victims found for incoming pod.
Warning   FailedScheduling  37s (x2 over 41s)  default-scheduler  0/2 nodes are available: 2 Insufficient cpu. preemption: 0/2 nodes are available: 2
No preemption victims found for incoming pod.
```



Why? – Degraded performance on Containerized workload

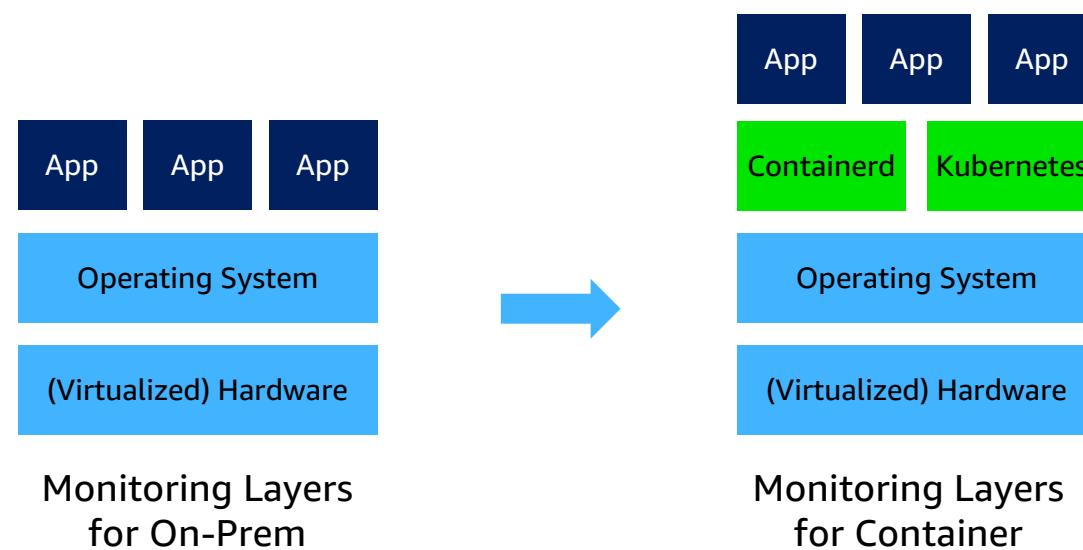


On-Prem vs Container

	On-Prem	Container
Characteristic	Static	Dynamic
Layers to monitor	HW OS Application	HW OS Orchestration ContainerRuntime Container Application
Complexity	Host Application	Host/Node Namespace Pod/Container
Alarm	Host base Report	Service base Auto Repair



On-Prem vs Container – Increased layers to monitor

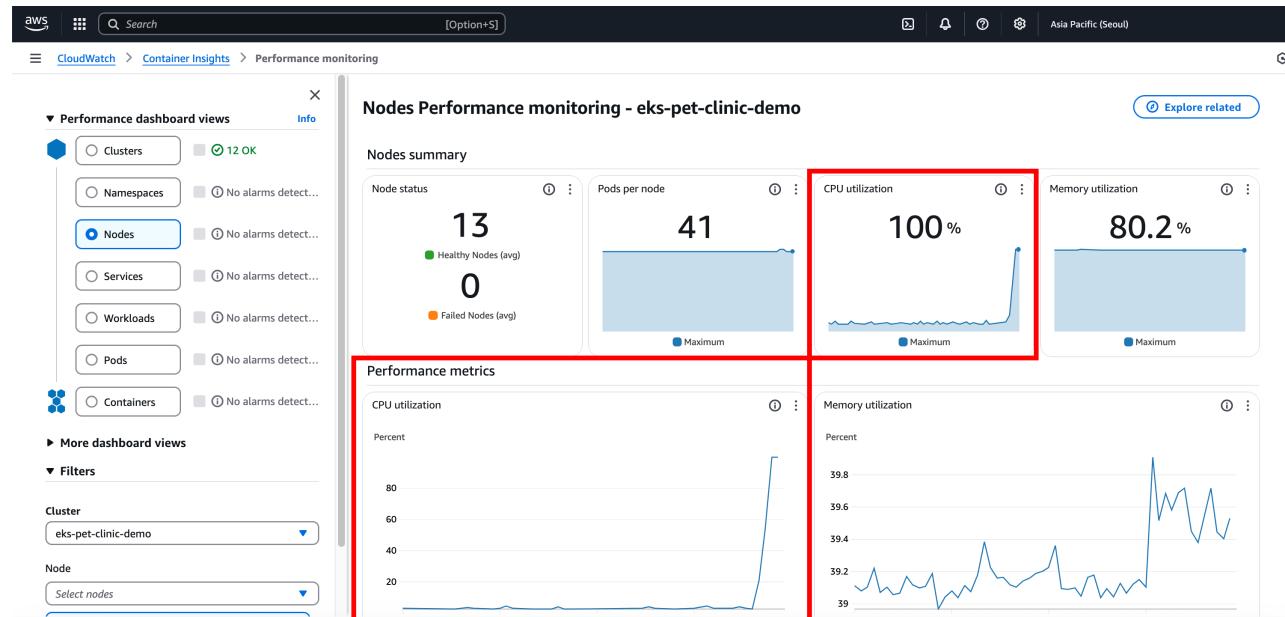


Purpose of EKS Monitoring

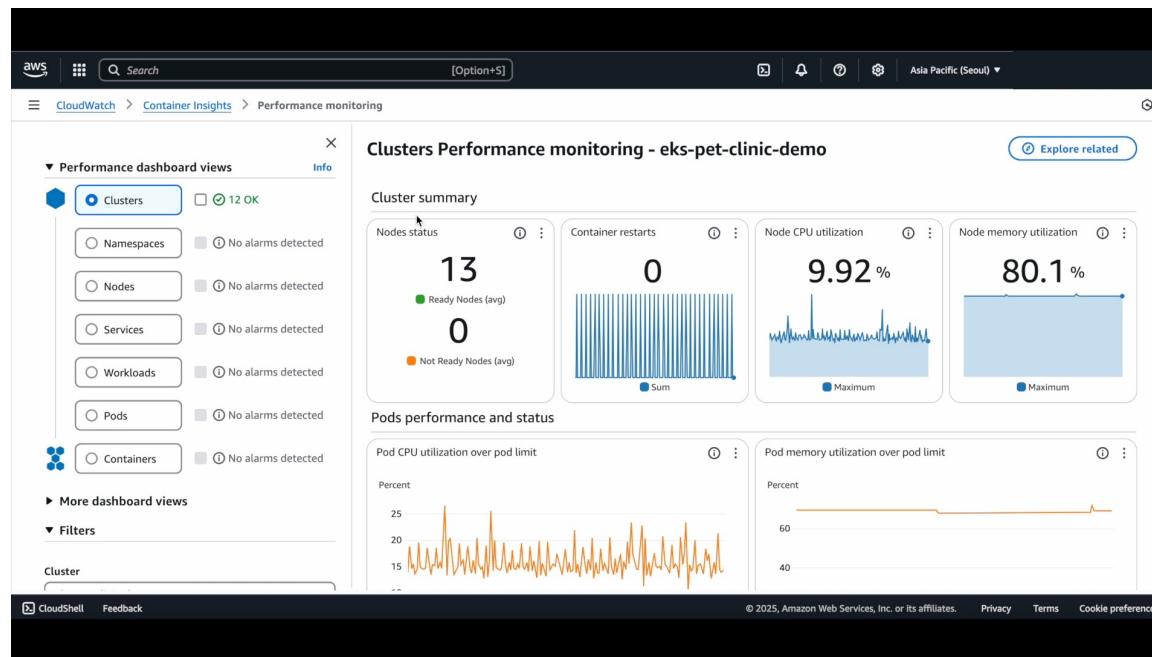
- Proactive Operation
- Convenient Troubleshooting
- Business Insight
- Security & Compliance



Purpose of EKS Monitoring – Proactive Operation



Purpose of EKS Monitoring – Convenient Troubleshooting



Purpose of EKS Monitoring – Business Insight

- Cost Optimization
- SLO & SLI Monitoring
- Collect service usage pattern / UX data
- Decision making – whether to use RI or Spot instance... etc.



Purpose of EKS Monitoring – Security & Compliance

- Compliance & Governance
- Security Monitoring & Auditing
- Access Control & Authentication

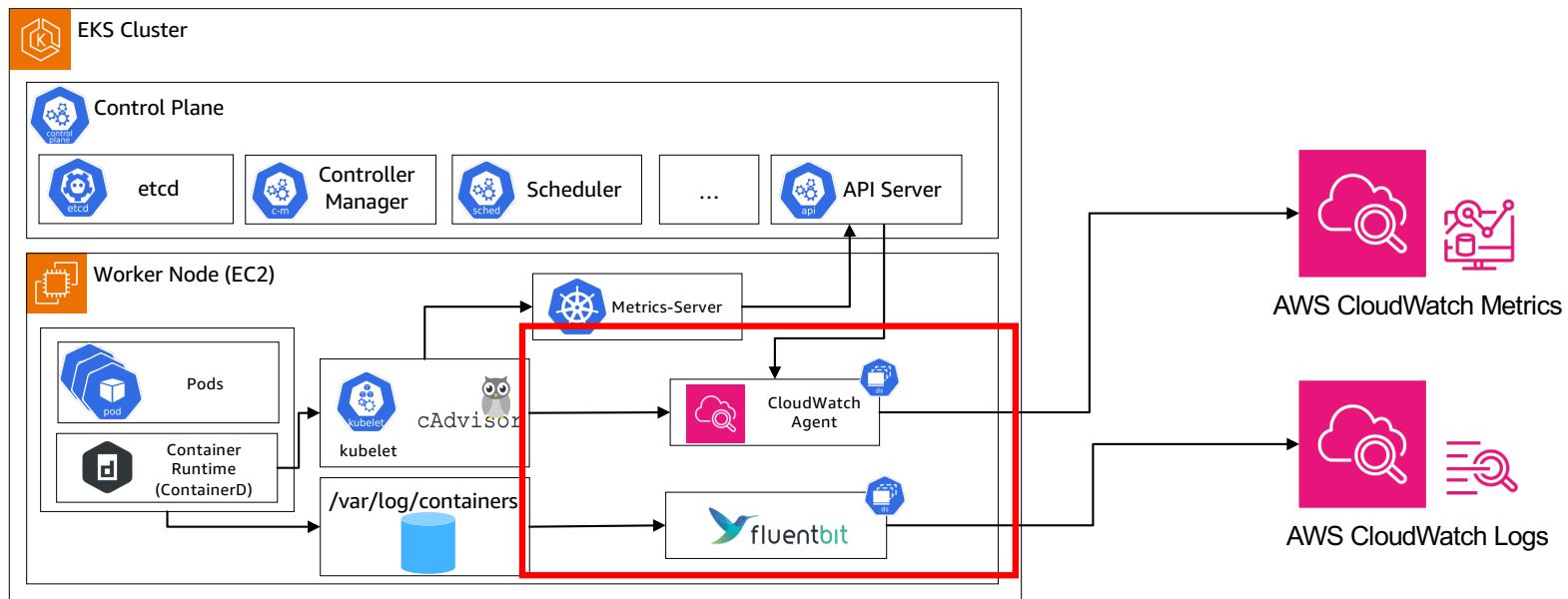


How do we Monitor EKS Cluster?



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Container insight - Architecture



<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ContainerInsights.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Container insight – Customized ConfigMap for CW Agent

```
% kubectl get cm -n amazon-cloudwatch cloudwatch-agent -o json \
| jq -r '.data."cwagentconfig.json"' \
| jq
{
  "agent": {
    "region": "ap-northeast-2"
  },
  "logs": {
    "metrics_collected": {
      "application_signals": {
        "hosted_in": "eks-pet-clinic-demo"
      },
      "kubernetes": {
        "cluster_name": "eks-pet-clinic-demo",
        "enhanced_container_insights": true
      }
    }
  },
  "traces": {
    "traces_collected": {
      "application_signals": {}
    }
  }
}
```

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Embedded_Metric_Format_Generation_CloudWatch_Agent.html#CloudWatch_Embedded_Metric_Format_Generation_Install_Agent_EKS



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Container insight – Customized Parser for Fluent Bit

```
% kubectl get cm -n amazon-cloudwatch fluent-bit-config -o yaml
apiVersion: v1
data:
  application-log.conf: |
    [INPUT]
      Name          tail
      Tag           application.*
      Exclude_Path ...
      Path          /var/log/containers/*.log
      multiline.parser
      DB            /var/fluent-bit/state/flb_container.db
      Read_From_Head ${READ_FROM_HEAD}

    [FILTER]
      Name          aws
      Match         application.*
      az            false
      ec2_instance_id  false
      Enable_Entity true

    [OUTPUT]
      Name          cloudwatch_logs
      Match         application.*
      region        ${AWS_REGION}
      log_group_name /aws/containerinsights/${CLUSTER_NAME}/application
      log_stream_prefix ${HOST_NAME}-
      auto_create_group true
      extra_user_agent container-insights
      add_entity     true
```

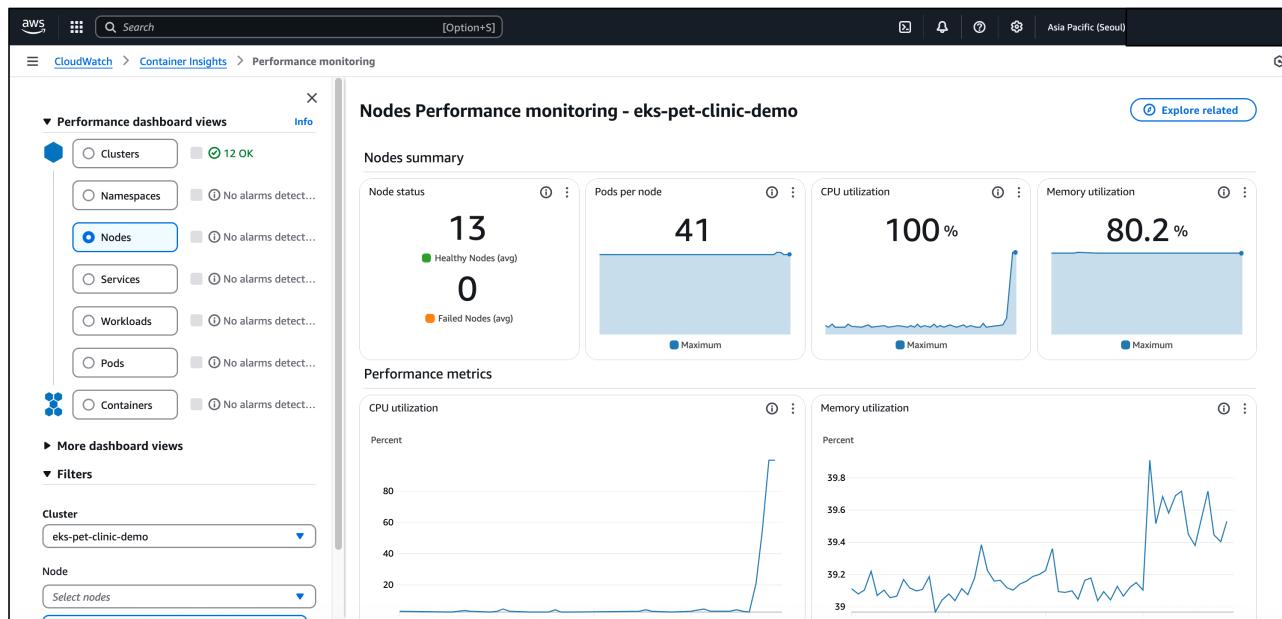


The Three Pillars of Observability

- **Metrics**: Numerical measurements of a system's performance collected over time.
- **Logs**: Timestamped, text-based records of discrete events within a system.
- **Traces**: Track the end-to-end journey of a single request or a transaction through a distributed system.



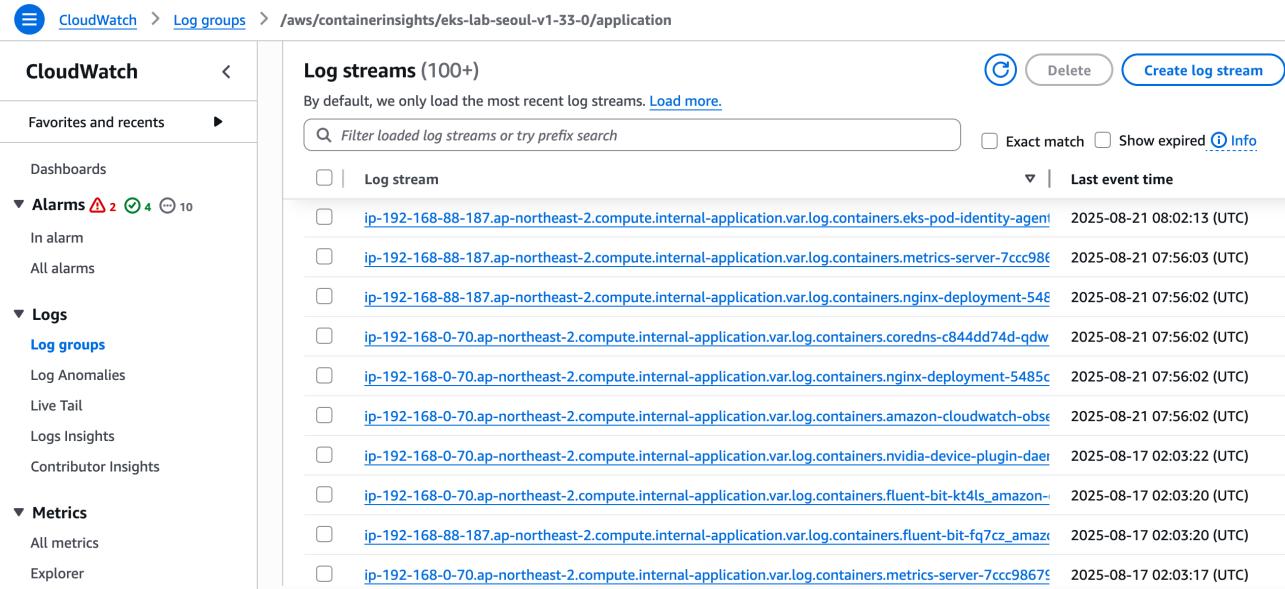
The Three Pillars of Observability – Metrics



AWS example: CloudWatch Metrics



The Three Pillars of Observability – Logs



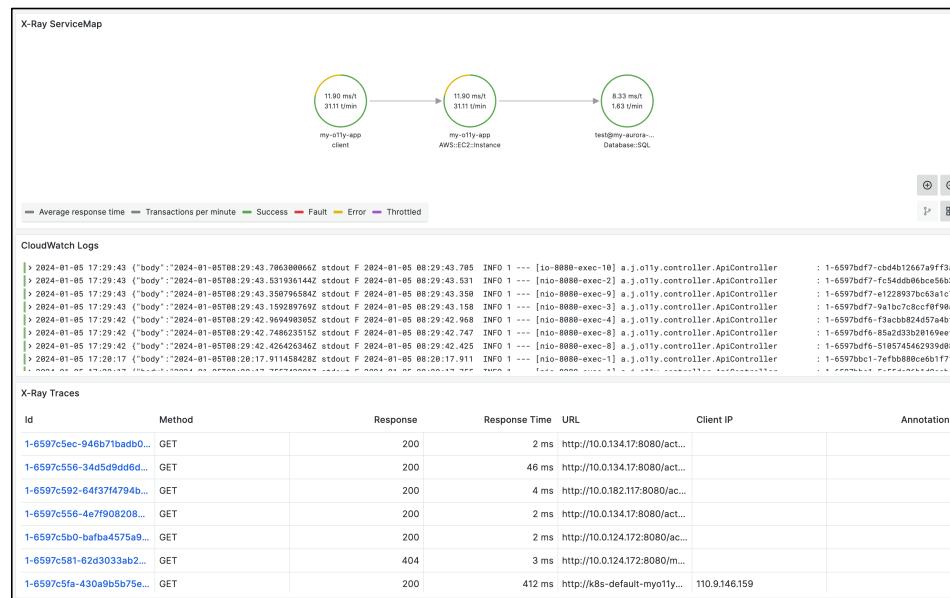
The screenshot shows the AWS CloudWatch Logs interface. The left sidebar navigation includes CloudWatch, Favorites and recents, Dashboards, Alarms (with 2 active, 4 total, and 10 pending), Logs (selected, showing Log groups, Log Anomalies, Live Tail, Logs Insights, and Contributor Insights), and Metrics (All metrics, Explorer). The main content area is titled "Log streams (100+)" and displays a list of log streams with their creation dates. A search bar at the top allows filtering by prefix search, exact match, or showing expired logs.

Log Stream	Last event time
ip-192-168-88-187.ap-northeast-2.compute.internal-application.var.log.containers.eks-pod-identity-agent	2025-08-21 08:02:13 (UTC)
ip-192-168-88-187.ap-northeast-2.compute.internal-application.var.log.containers.metrics-server-7ccc98e	2025-08-21 07:56:03 (UTC)
ip-192-168-88-187.ap-northeast-2.compute.internal-application.var.log.containers.nginx-deployment-54f	2025-08-21 07:56:02 (UTC)
ip-192-168-0-70.ap-northeast-2.compute.internal-application.var.log.containers.coredns-c844dd74d-qdw	2025-08-21 07:56:02 (UTC)
ip-192-168-0-70.ap-northeast-2.compute.internal-application.var.log.containers.nginx-deployment-5485c	2025-08-21 07:56:02 (UTC)
ip-192-168-0-70.ap-northeast-2.compute.internal-application.var.log.containers.amazon-cloudwatch-obs	2025-08-21 07:56:02 (UTC)
ip-192-168-0-70.ap-northeast-2.compute.internal-application.var.log.containers.nvidia-device-plugin-dae	2025-08-17 02:03:22 (UTC)
ip-192-168-0-70.ap-northeast-2.compute.internal-application.var.log.containers.fluent-bit-kt4ls_amazon-	2025-08-17 02:03:20 (UTC)
ip-192-168-88-187.ap-northeast-2.compute.internal-application.var.log.containers.fluent-bit-fq7cz_amaz	2025-08-17 02:03:20 (UTC)
ip-192-168-0-70.ap-northeast-2.compute.internal-application.var.log.containers.metrics-server-7ccc98e7	2025-08-17 02:03:17 (UTC)

AWS example: CloudWatch Logs



The Three Pillars of Observability – Traces

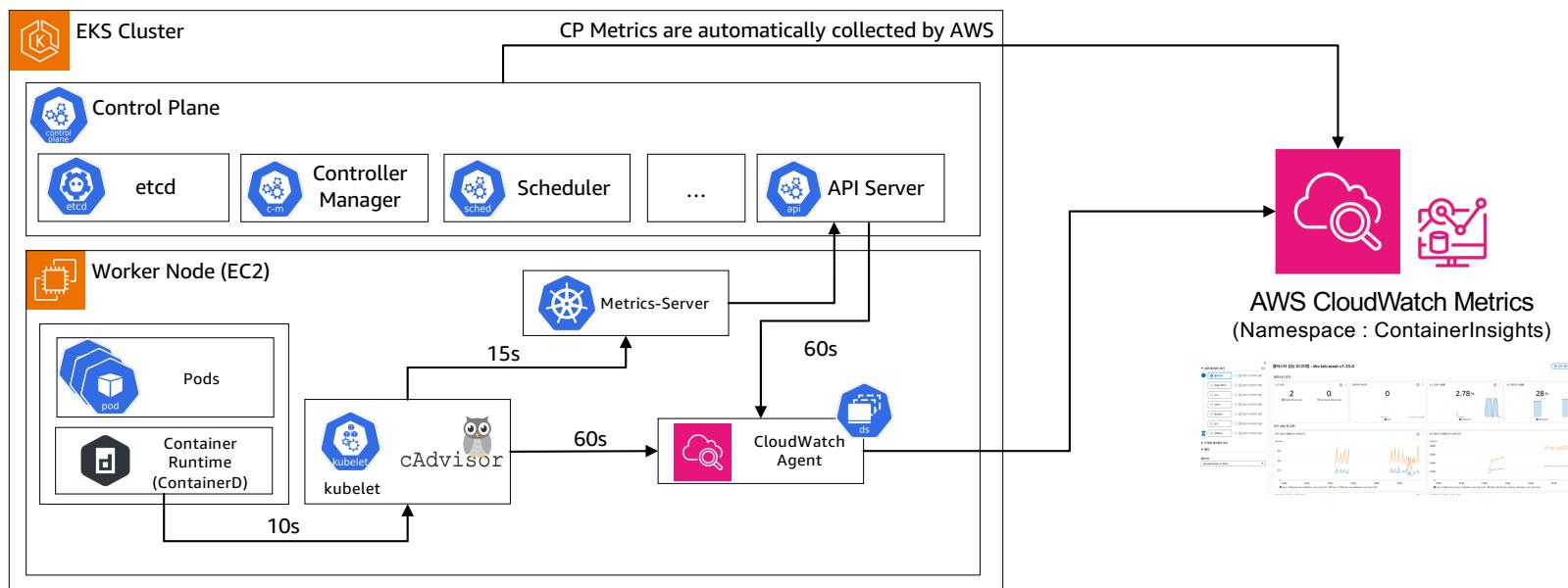


AWS example: X-Ray

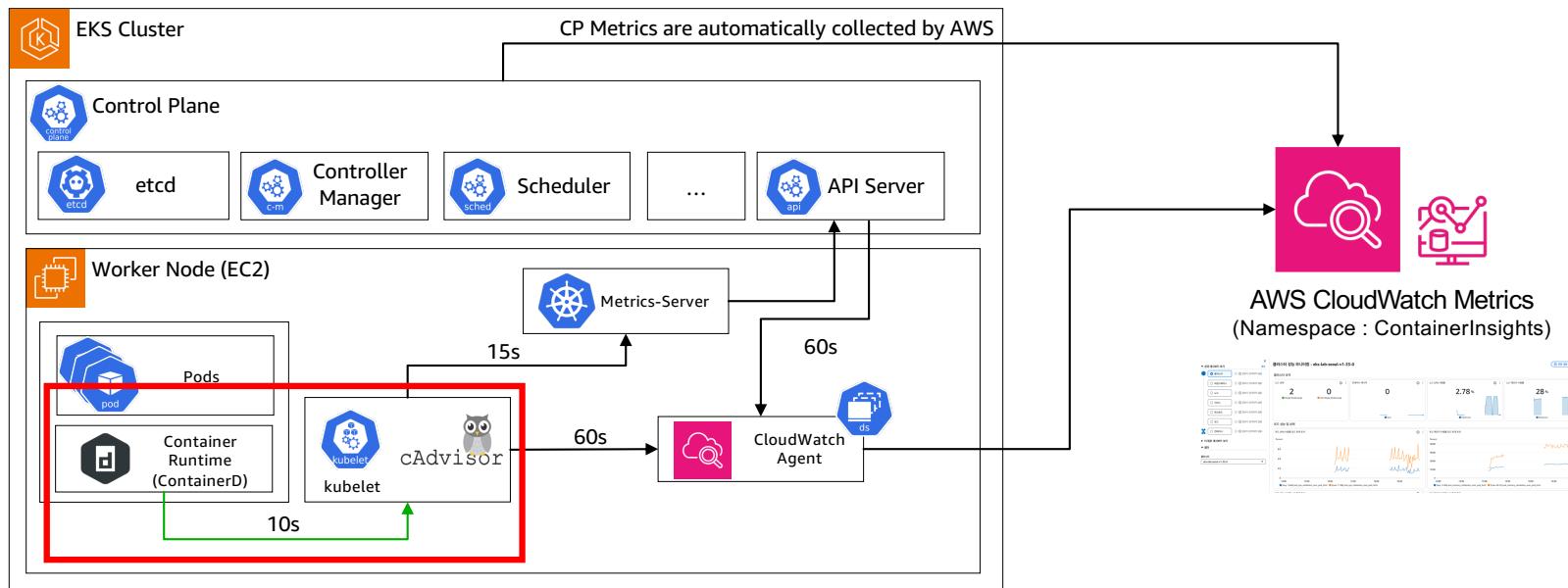


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

EKS Monitoring Architecture (Metrics)

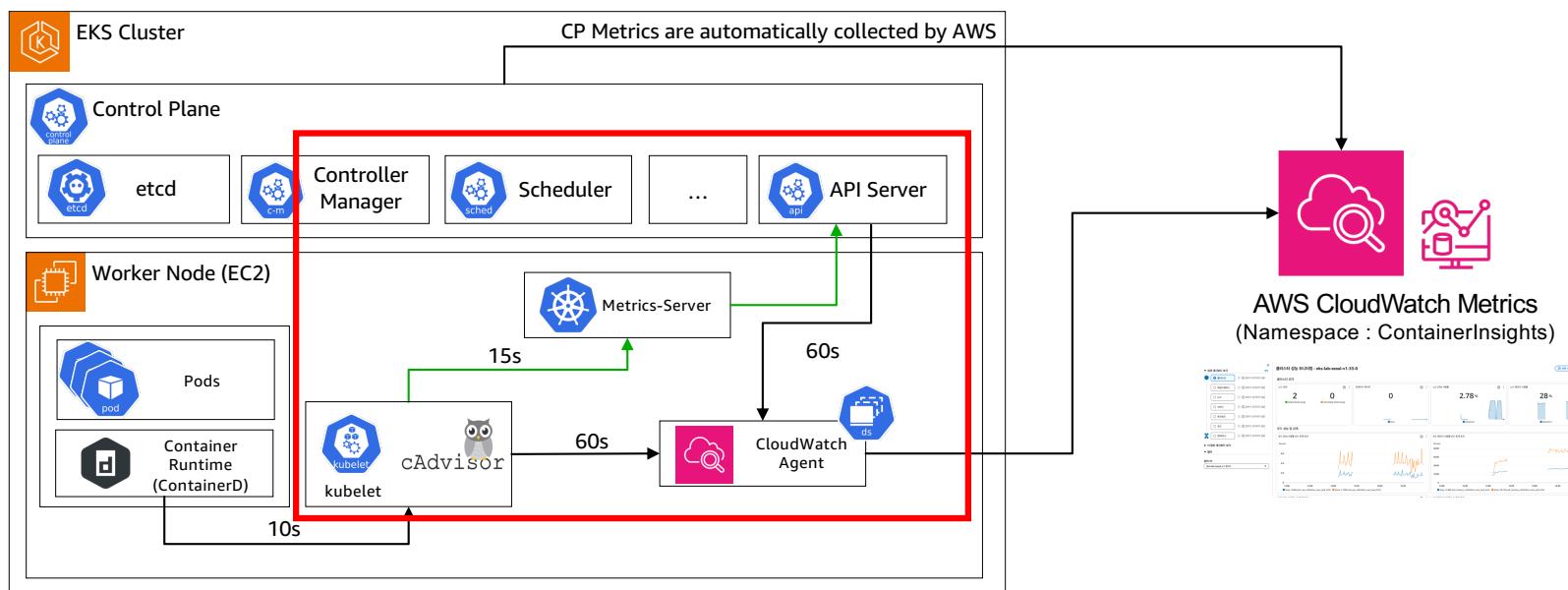


EKS Monitoring Architecture (Metrics) – cAdvisor & kubelet

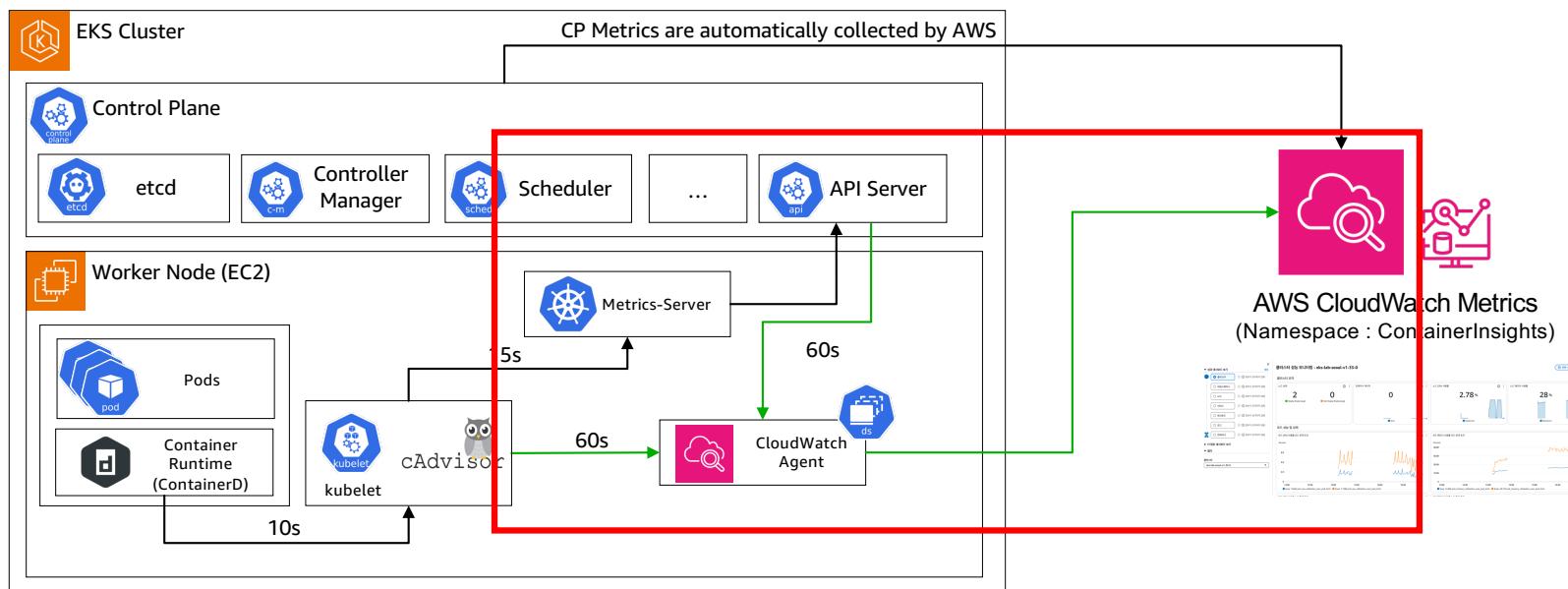


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

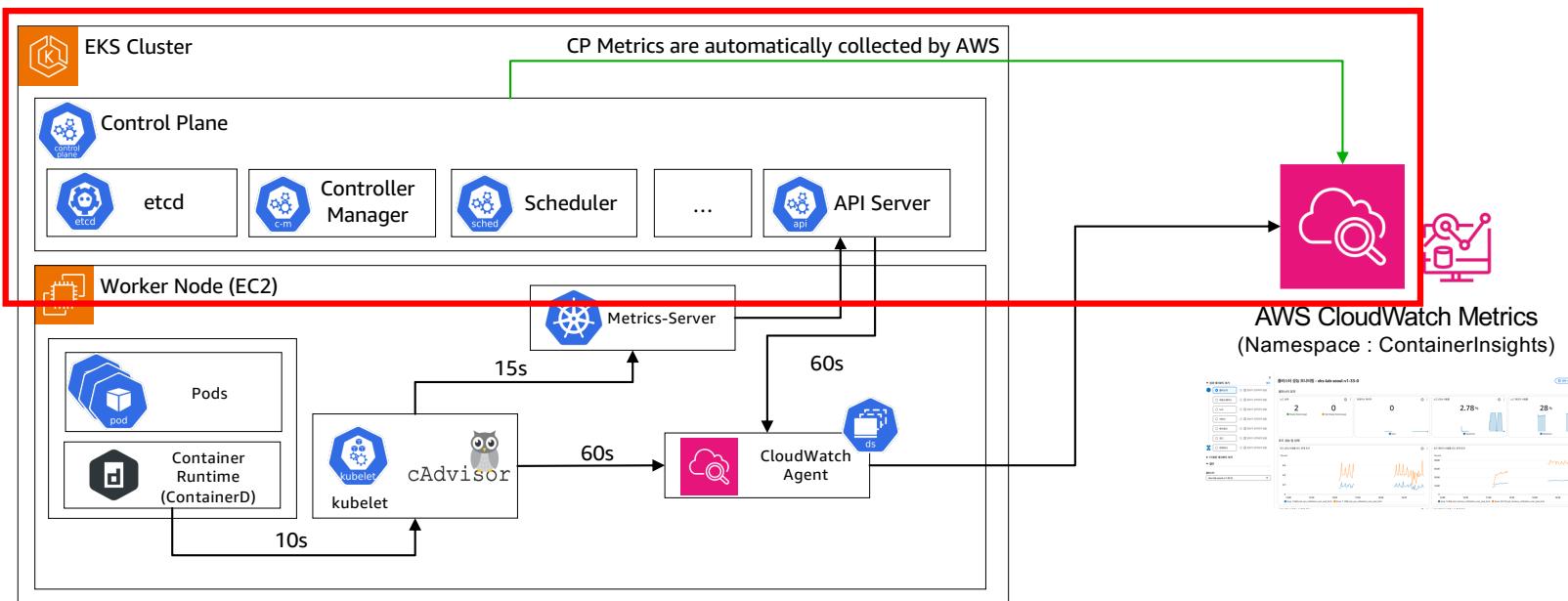
EKS Monitoring Architecture (Metrics) – Metrics server



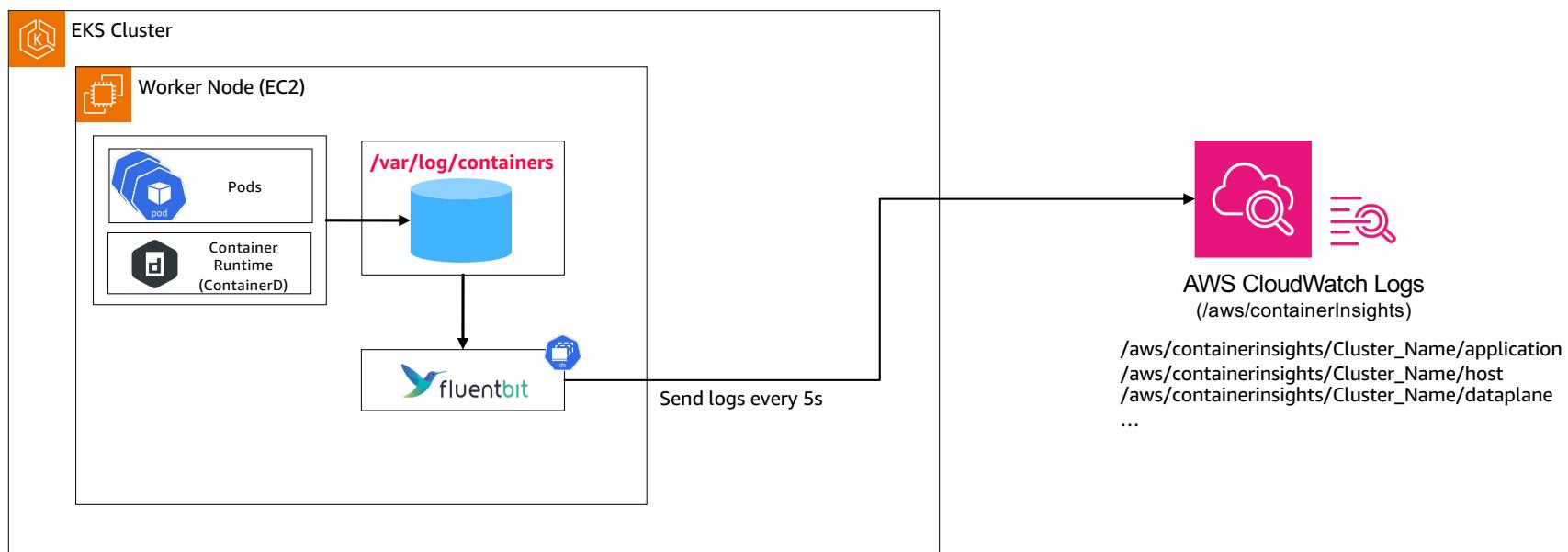
EKS Monitoring Architecture (Metrics) – CW Agent



EKS Monitoring Architecture (Metrics) – CP Metrics



EKS Monitoring Architecture (Logs)



Collected Logs by Container Insights

The screenshot shows the AWS CloudWatch Log groups interface. The left sidebar navigation includes CloudWatch, Favorites and recents, Dashboards, Alarms (with 1 in alarm and 42 total), and Logs (with Log groups, Log Anomalies, Live Tail, Logs Insights, and Contributor Insights). The main content area displays a search bar with the query "eks-pet-clinic-demo", a "Log groups (71)" heading, and a table of log groups. The table columns are Log group, Log class, Anomaly d..., and Data protection. The log groups listed are:

Log group	Log class	Anomaly d...	Data protection
/aws/containerinsights/eks-pet-clinic-demo/application	Standard	Configure	-
/aws/containerinsights/eks-pet-clinic-demo/dataplane	Standard	Configure	-
/aws/containerinsights/eks-pet-clinic-demo/host	Standard	Configure	-
/aws/containerinsights/eks-pet-clinic-demo/performance	Standard	Configure	-



Types of Collected Logs



AWS CloudWatch Logs
(/aws/containerInsights/[Cluster Name]/[Path])

application

pods
containers
(stdout, stderr)

dataplane

kubelet
kube-proxy
containerd
aws-node (vpc-cni)

host

/var/log/dmesg
/var/log/secure
/var/log/messages

performance

CPU
Memory
Network
Disk I/O

<https://docs.aws.amazon.com/eks/latest/userguide/control-plane-logs.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Types of Collected Logs - Application

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar navigation includes:

- CloudWatch
- Favorites and recents
- Dashboards
- Alarms (1 active, 42 recent)
- Logs
 - Log groups
 - Log Anomalies
 - Live Tail
 - Logs Insights
 - Contributor Insights
- Metrics
 - All metrics
 - Explorer
 - Streams

The main content area is titled "Log events" and displays log entries. A search bar at the top says "Filter events - press enter to search". Below it are time range controls (1m, 1h, UTC timezone) and a "Display" dropdown. The log table has columns for "Timestamp" and "Message". The first two entries are collapsed, while the third entry is expanded, showing a JSON log message. The expanded message content is as follows:

```
{  
  "time": "2025-08-27T16:25:14.253Z",  
  "stream": "stdout",  
  "_p": "F",  
  "log": "\t\tat io.netty.channel.AbstractChannelHandlerContext.invokeChannelRead(AbstractChannelHandlerContext.java:379)",  
  "kubernetes": {  
    "pod_name": "pet-clinic-frontend-java-84787ff964-5fjf4",  
    "namespace_name": "pet-clinic",  
    "pod_id": "b2cb12e0-7633-4e88-aaa0-257744ecfb0f",  
    "host": "ip-10-0-2-207.ap-northeast-2.compute.internal",  
    "container_name": "api-gateway-java",  
    "docker_id": "45e104ee8469af8ef883b0624092ee304422a40b1a4b8101374cd09aae2276c",  
    "container_hash": "404227808920.dkr.ecr.ap-northeast-2.amazonaws.com/springcommunity/spring-petclinic-api-gateway@sha256:19578c6d6972fb21d875214d0033097e00f4e6d0dd7c9be30abb41a916648022",  
    "container_image": "404227808920.dkr.ecr.ap-northeast-2.amazonaws.com/springcommunity/spring-petclinic-api-gateway:latest"  
  }  
}
```



Types of Collected Logs – Data Plane

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar includes navigation links for CloudWatch, Log groups, Alarms (with 1 active), Logs (with Log groups, Log Anomalies, Live Tail, Logs Insights, and Contributor Insights), and Metrics (with All metrics and Explorer). The main content area displays log events for the log group `/aws/containerinsights/eks-pet-clinic-demo/dataplane`. A search bar at the top allows filtering by terms, phrases, or values. Below the search bar are time range controls (1m, 1h, UTC timezone) and a display dropdown. The log events table has columns for Timestamp and Message. One event is expanded to show its full JSON content, which includes fields like time, stream, _p, log, numEndpoints, numFilterChains, numFilterRules, numNATChains, numNATRules, az, and ec2_instance_id.

Timestamp	Message								
2025-08-26T00:13:03.278Z	{"time": "2025-08-26T00:13:03.278850949Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:03.278600 1 ..."} {"time": "2025-08-26T00:13:04.278Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.278827 1 p..."} <tr><td>2025-08-26T00:13:04.287Z</td><td>{"time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 1 ..."}<td>{ "time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 [proxier.go:1547] \\\"Reloading service iptables data\\\" ipFamily=\\\"IPv4\\\" numServices=64 numEndpoints=131 numFilterChains=6 numFilterRules=14 numNATChains=6 numNATRules=67", "az": "ap-northeast-2b", "ec2_instance_id": "i-0b0f281a7c65e0f56" }</td></td></tr> <tr><td>2025-08-26T00:13:04.344Z</td><td>{"time": "2025-08-26T00:13:04.344973041Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.344857 1 ..."} {"time": "2025-08-26T00:13:26.372Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.372681 1 ..."}<tr><td>2025-08-26T00:13:26.382Z</td><td>{"time": "2025-08-26T00:13:26.382536808Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.382345 1 ..."}<td></td></td></tr></td></tr>	2025-08-26T00:13:04.287Z	{"time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 1 ..."} <td>{ "time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 [proxier.go:1547] \\\"Reloading service iptables data\\\" ipFamily=\\\"IPv4\\\" numServices=64 numEndpoints=131 numFilterChains=6 numFilterRules=14 numNATChains=6 numNATRules=67", "az": "ap-northeast-2b", "ec2_instance_id": "i-0b0f281a7c65e0f56" }</td>	{ "time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 [proxier.go:1547] \\\"Reloading service iptables data\\\" ipFamily=\\\"IPv4\\\" numServices=64 numEndpoints=131 numFilterChains=6 numFilterRules=14 numNATChains=6 numNATRules=67", "az": "ap-northeast-2b", "ec2_instance_id": "i-0b0f281a7c65e0f56" }	2025-08-26T00:13:04.344Z	{"time": "2025-08-26T00:13:04.344973041Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.344857 1 ..."} {"time": "2025-08-26T00:13:26.372Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.372681 1 ..."} <tr><td>2025-08-26T00:13:26.382Z</td><td>{"time": "2025-08-26T00:13:26.382536808Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.382345 1 ..."}<td></td></td></tr>	2025-08-26T00:13:26.382Z	{"time": "2025-08-26T00:13:26.382536808Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.382345 1 ..."} <td></td>	
2025-08-26T00:13:04.287Z	{"time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 1 ..."} <td>{ "time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 [proxier.go:1547] \\\"Reloading service iptables data\\\" ipFamily=\\\"IPv4\\\" numServices=64 numEndpoints=131 numFilterChains=6 numFilterRules=14 numNATChains=6 numNATRules=67", "az": "ap-northeast-2b", "ec2_instance_id": "i-0b0f281a7c65e0f56" }</td>	{ "time": "2025-08-26T00:13:04.287986913Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.287838 [proxier.go:1547] \\\"Reloading service iptables data\\\" ipFamily=\\\"IPv4\\\" numServices=64 numEndpoints=131 numFilterChains=6 numFilterRules=14 numNATChains=6 numNATRules=67", "az": "ap-northeast-2b", "ec2_instance_id": "i-0b0f281a7c65e0f56" }							
2025-08-26T00:13:04.344Z	{"time": "2025-08-26T00:13:04.344973041Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:04.344857 1 ..."} {"time": "2025-08-26T00:13:26.372Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.372681 1 ..."} <tr><td>2025-08-26T00:13:26.382Z</td><td>{"time": "2025-08-26T00:13:26.382536808Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.382345 1 ..."}<td></td></td></tr>	2025-08-26T00:13:26.382Z	{"time": "2025-08-26T00:13:26.382536808Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.382345 1 ..."} <td></td>						
2025-08-26T00:13:26.382Z	{"time": "2025-08-26T00:13:26.382536808Z", "stream": "stderr", "_p": "F", "log": "I0826 00:13:26.382345 1 ..."} <td></td>								

Types of Collected Logs – Host

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar contains navigation links for CloudWatch, Log groups, Metrics, and Streams. The main area displays log events for the path /aws/containerinsights/eks-pet-clinic-demo/host. The log events table has columns for Timestamp and Message. The first two log entries are expanded to show detailed JSON logs from containerd and kubelet. The third log entry is collapsed.

Timestamp	Message
2025-08-18T06:33:58.000Z	{ "host": "ip-10-0-3-183", "ident": "containerd", "message": "time=\"2025-08-18T06:33:58.561381380Z\" level=info msg=\"StopPodSandbox for \\"48bd716c20112c8c97c07a168cc68fa31ced73e4e22846f7b975a42e6ed2485d\\\" returns successfully\"", "az": "ap-northeast-2b", "ec2_instance_id": "i-058d24aa0f670c7a9" }
2025-08-18T06:33:58.000Z	{ "host": "ip-10-0-3-183", "ident": "kubelet", "message": "I0818 06:33:58.567170 2893 util.go:48] \"No ready sandbox for pod can be found. Need to start a new one\" pod=\"pet-clinic/billing-service-python-846f8cbd96-gfz7c\"", "az": "ap-northeast-2b", "ec2_instance_id": "i-058d24aa0f670c7a9" }
2025-08-18T06:33:58.000Z	{"host": "ip-10-0-3-183", "ident": "kubelet", "message": "I0818 06:33:58.610264 2893 reconciler_common.g..."}

Types of Collected Logs – Performance

The screenshot shows the AWS CloudWatch Log Events interface. The left sidebar navigation includes:

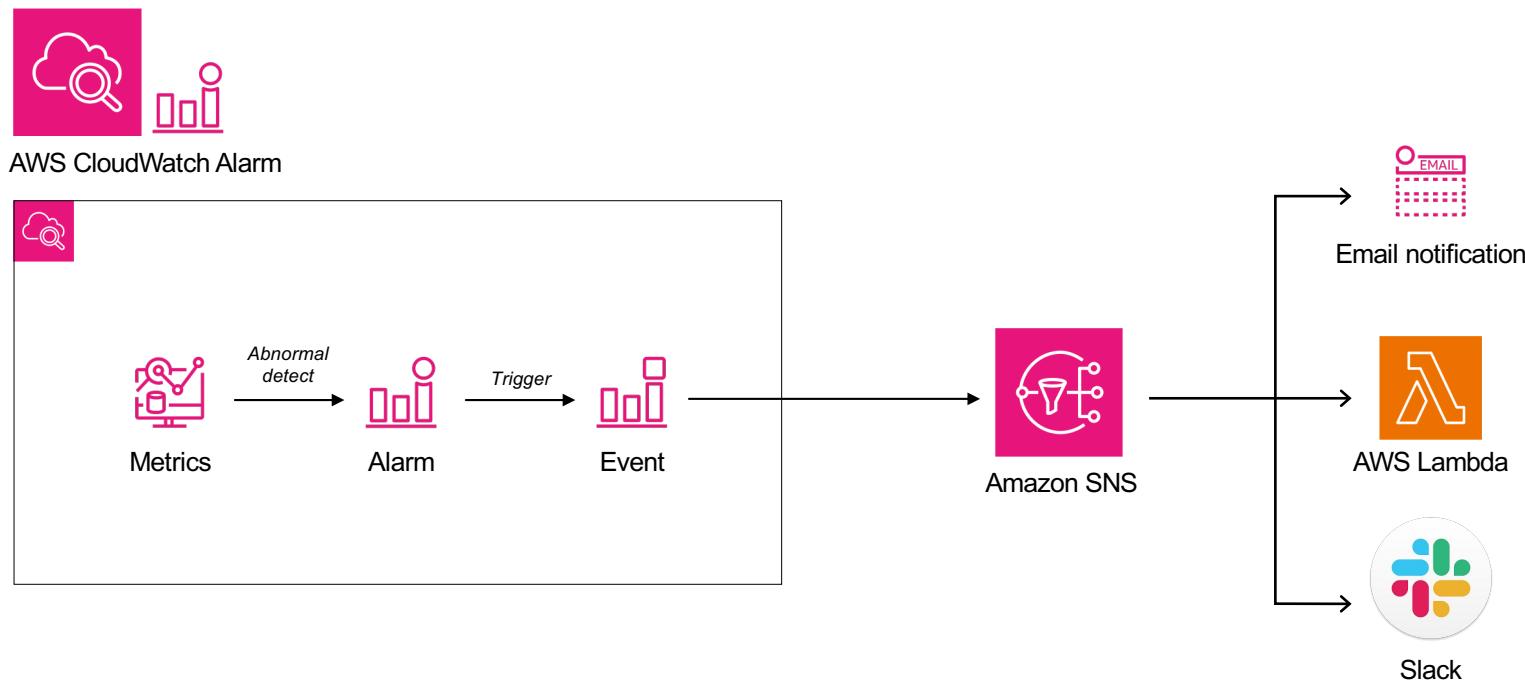
- CloudWatch
- Favorites and recents
- Dashboards
- Alarms (1)
- Logs (selected)
- Log groups
- Log Anomalies
- Live Tail
- Logs Insights
- Contributor Insights
- Metrics
- All metrics
- Explorer
- Streams
- Application Signals (APM)
- Network Monitoring
- Insights
- Container Insights
- Database Insights
- Lambda Insights
- Application Insights (1)
- EC2 Resource Health
- Settings
- Telemetry config

The main content area displays log events for a container named "kube-proxy". The log message is as follows:

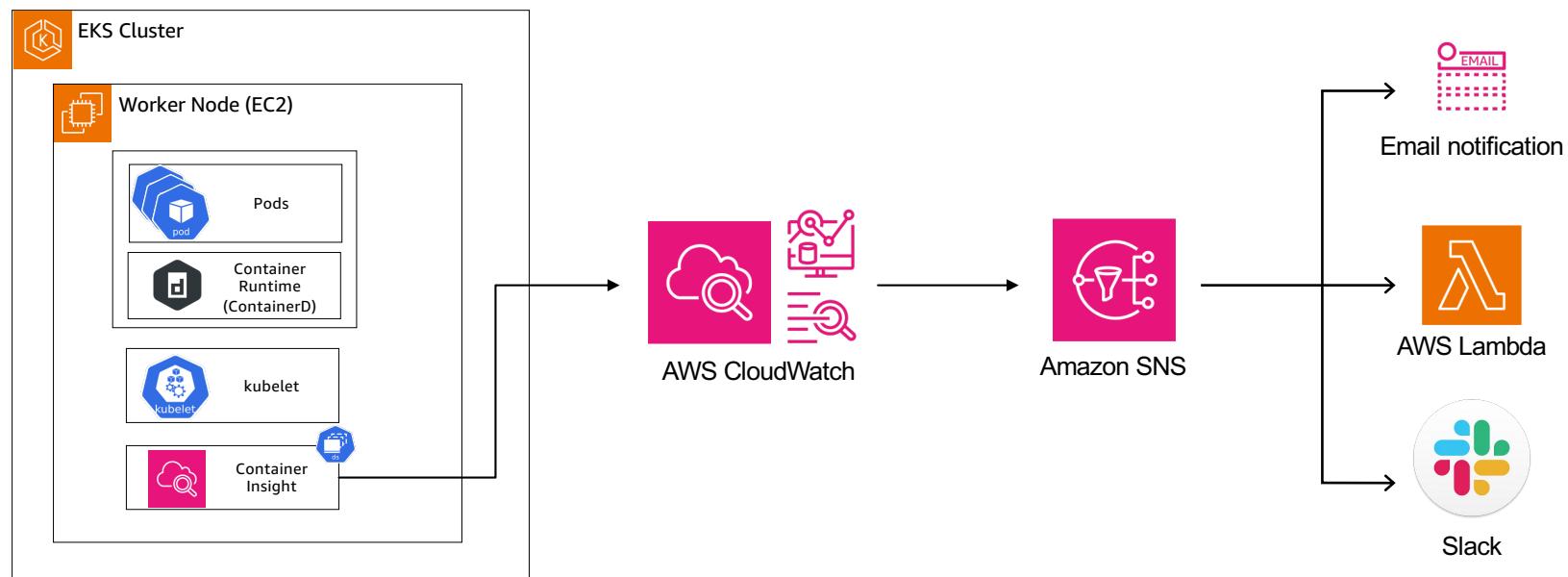
```
2025-08-27T18:57:53.698Z {"AutoScalingGroupName": "", "CloudWatchMetrics": [{"Namespace": "ContainerInsights", "Dimensions": [{"ClusterName": "..."}]}], "ContainerID": "e03e971b91e7dedee0fa4a2b89b837e94cabed28761263bb8ec658673628e353", "ContainerName": "kube-proxy", "ContainerType": "Container", "ContainerStatus": "Running", "ContainerVersion": "0", "ContainerUsage": {"System": 0.2197779838794033, "User": 0.292831708557714, "Total": 0.5126197}, "ContainerMemory": {"Cache": 1290240, "Failcnt": 0, "HierarchicalPgfault": 9.64951187702932, "HierarchicalPgmajfault": 0, "MappedFile": 327680, "MaxUsage": 20484096, "Pgfault": 9.64951187702932, "Pgmajfault": 0, "Rss": 14282752, "Swap": 0, "Usage": 16703488, "WorkingSet": 16187392}, "Host": "ip-10-0-2-175.ap-northeast-2.compute.internal", "Labels": {"controller-revision-hash": "7ffcf48564", "k8s-app": "kube-proxy", "pod-template-generation": "1"}, "NamespaceName": "kube-system", "PodID": "fdf2cd5-c116-4f84-a958-38bf0d3a80df", "PodName": "kube-proxy-fcscd", "PodOwners": [{"OwnerKind": "DaemonSet", "OwnerName": "kube-proxy"}], "NumberOfContainerRestarts": 0, "ContainerCPURequest": 100, "ContainerCPUUtilization": 0.014641565854278858, "ContainerMemoryFailuresTotal": 9.64951187702932, "ContainerMemoryUtilization": 0.40206158297505223}
```



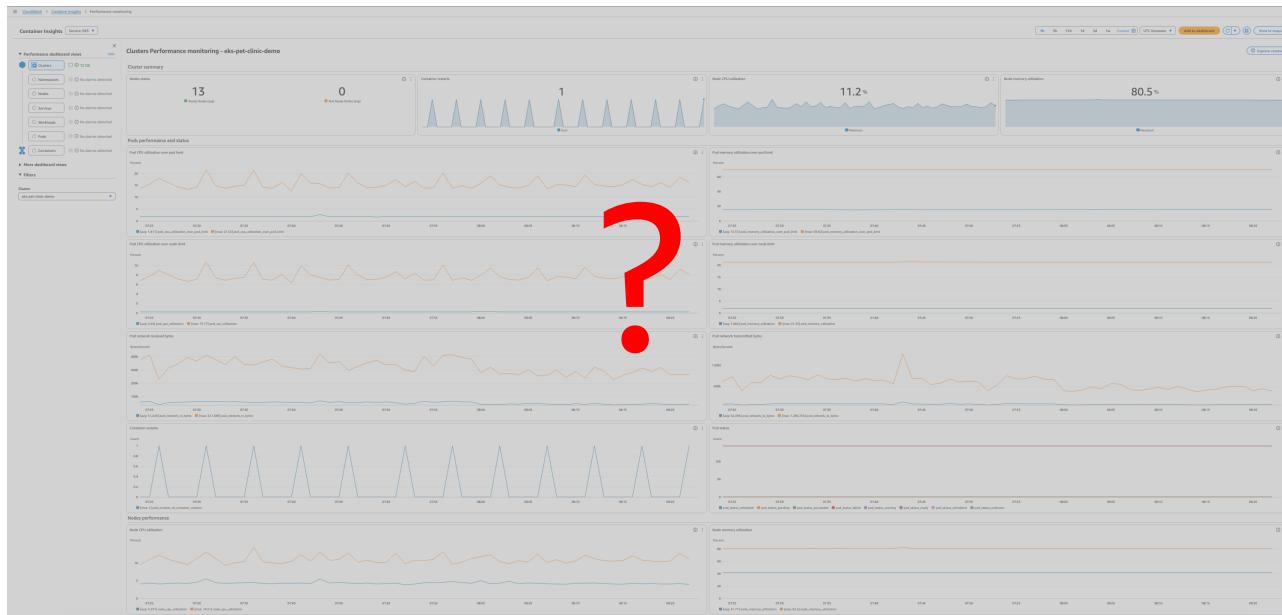
EKS Monitoring Architecture (Notification)



Overall EKS Monitoring Architecture



So, what can we do with these data collected?



APM (Application Performance Monitoring)

Monitor the performance of business-critical applications

- Agent/Sidecar or code-level modification
- Service Mesh (e.g., Istio)

Benefits of integrating with Cluster & Application Performance monitoring

- Performance issue in application & Abnormally found from Infra metrics
--> Faster root cause detection / mitigation for the issue
- Unified monitoring for cluster & Application
--> Optimizing performance by tuning worker node type / # of nodes

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Application-Monitoring-Intro.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

SLI & SLO?

SLI (Service Level Indicators) – Quantitative measure of service reliability

- Availability / Error Rate
- Latency
- Throughput

SLO (Service Level Objectives) – Target goals for SLI

- 99.9% of service availability
- 99.9% of request complete with 100ms

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Application-Monitoring-Intro.html>



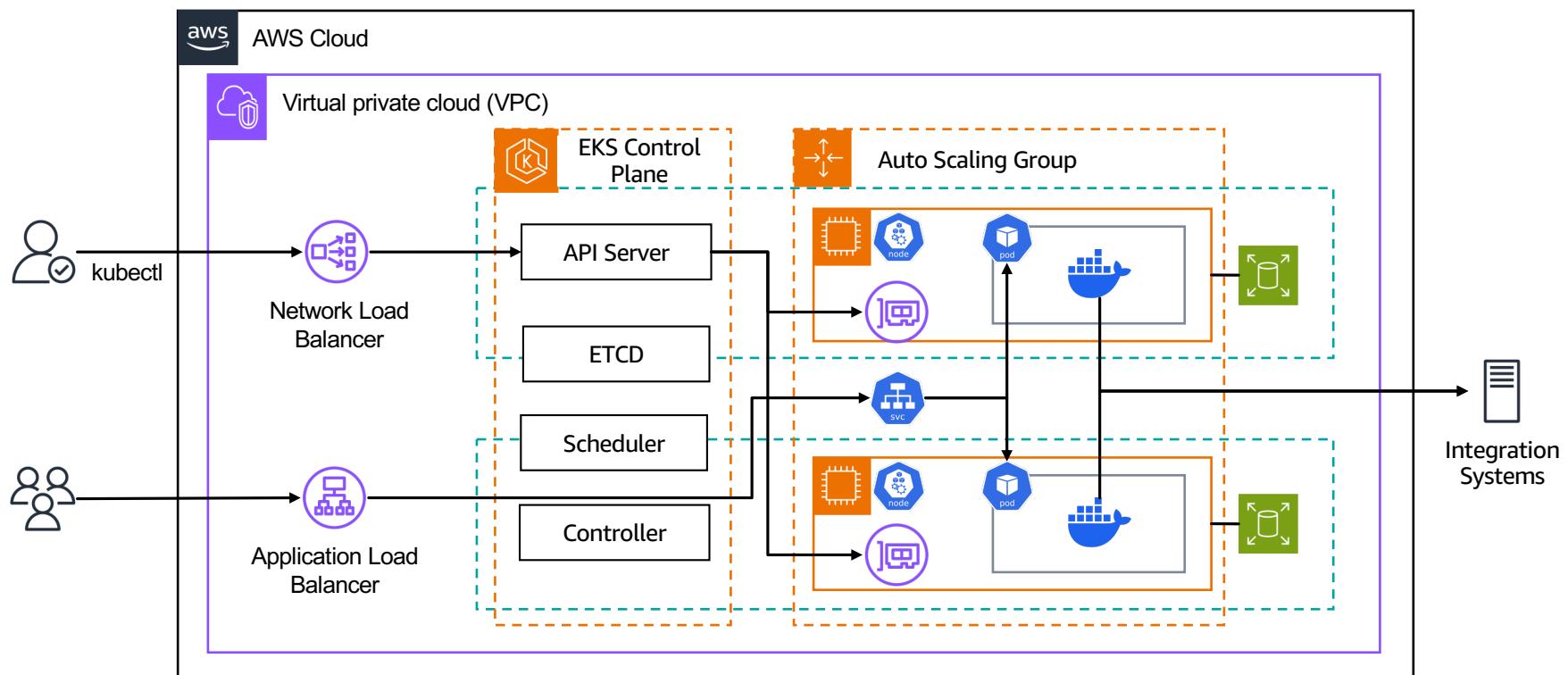
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Let's do!

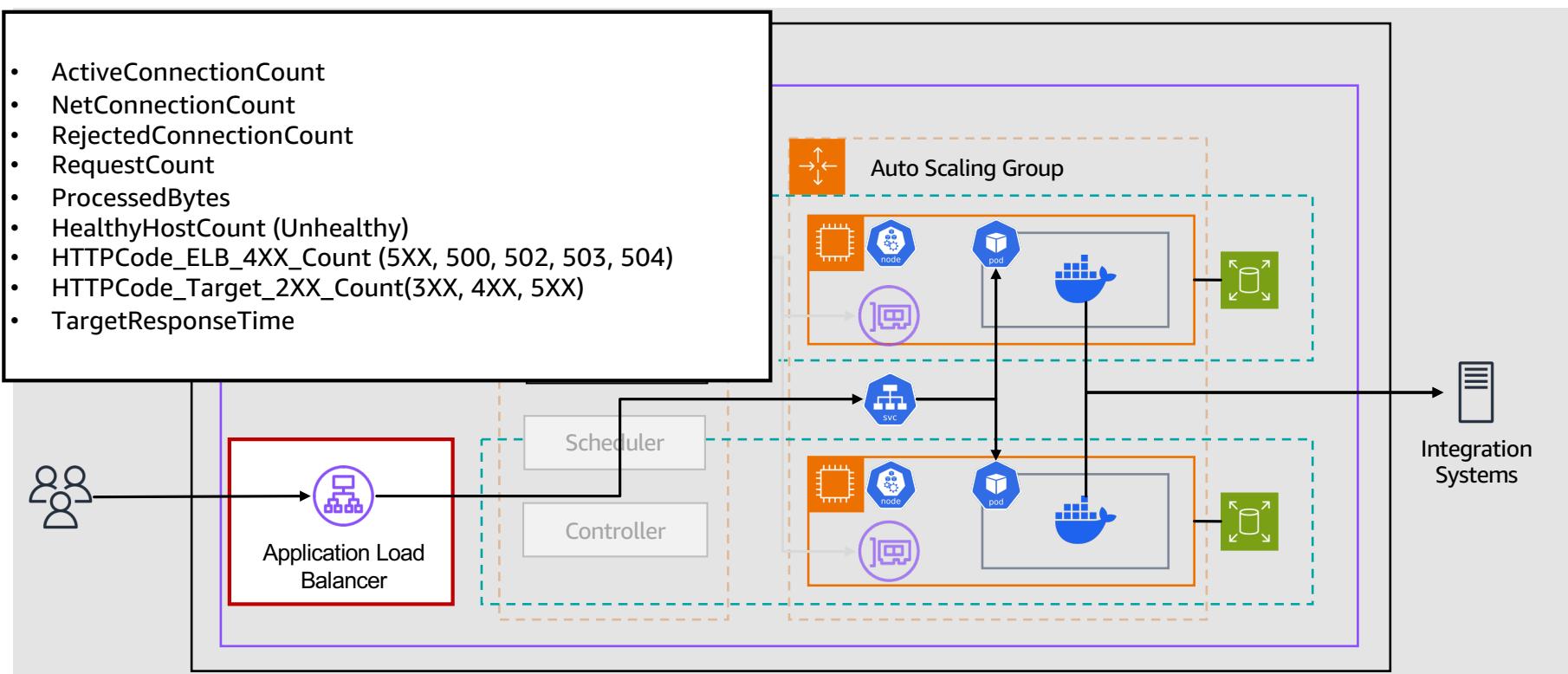


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Useful Metrics for EKS Cluster



Useful Metrics for EKS Cluster - ALB



<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-cloudwatch-metrics.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Useful Metrics for EKS Cluster - Node

Node

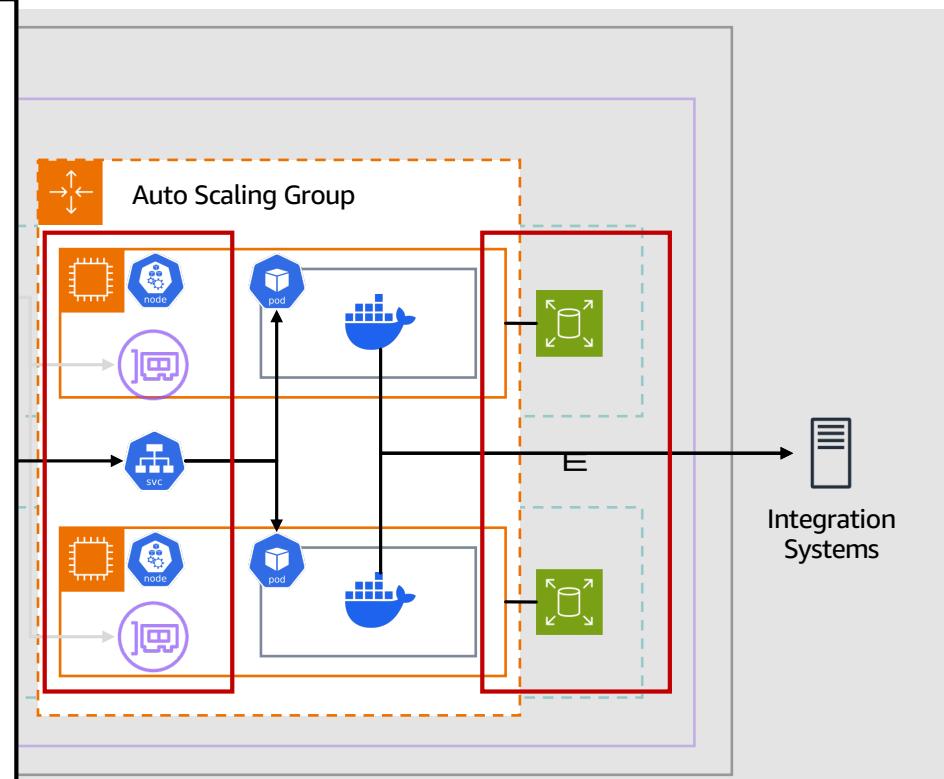
- node_cpu_utilization (limit, usage, reserved)
- node_filesystem_utilization (limit, usage, reserved)
- node_memory_utilization (limit, usage, reserved)
- node_network_total_bytes
- node_interface_network_rx/tx_dropped
- node_status_condition_ready (memory_pressure, pid_pressure, dis_pressure, unknown)

Instance

- CPUUtilization
- DiskReadOps/Bytes, DiskWriteOps/Bytes, EBSIOBalance%, VolumeldleTime, Throughput
- NetworkIn/Out, NetworkPacketsIn/Out
- StatusCheckFailed
- bw_in_allowance_exceeded/bw_out_allowance_exceeded
- linklocal_allowance_exceeded
- pps_allowance_exceeded

Logs

- kubelet, containerd, kube-proxy, vpc-cni
- /var/log/containers, /var/log/*



<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-metrics-enhanced-EKS.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

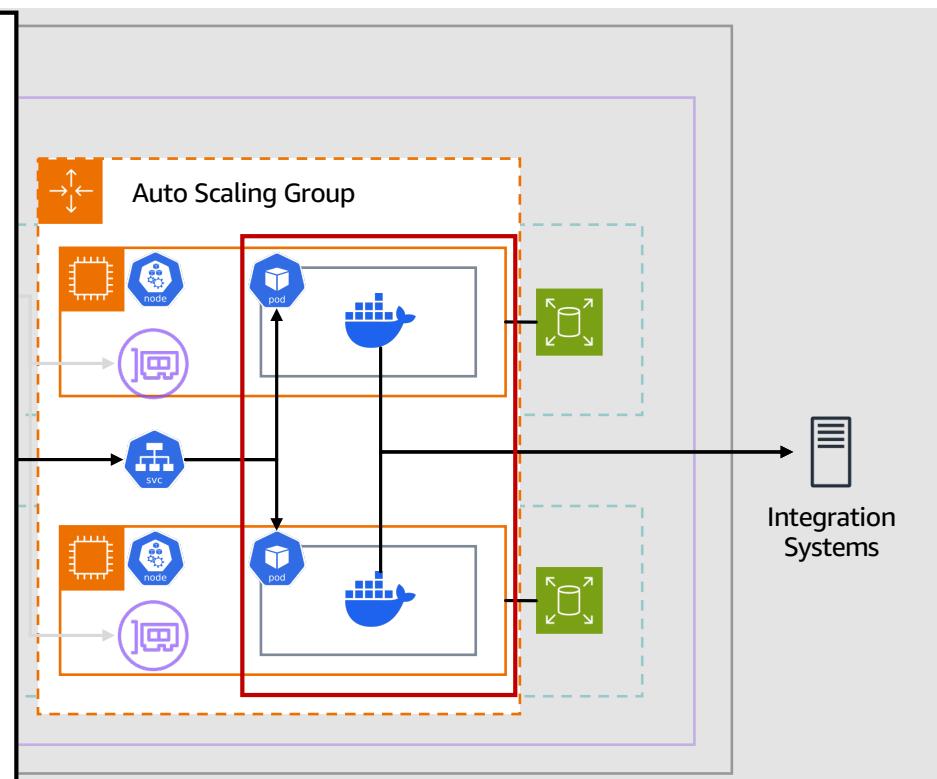
Useful Metrics for EKS Cluster – Pod/Container

Pod

- pod_cpu_utilization (reserved_capacity, over_pod_limit)
- pod_memory_utilization (reserved_capacity, over_pod_limit)
- pod_network_rx/tx_bytes
- pod_status_running (failed, ready, scheduled, unknown, pending, succeeded)
- pod_interface_network_rx/tx_dropped

Container

- pod_container_status_running (terminated, waiting, waiting_reason_crash_loop_back, waiting_reason_create_container_config_error, waiting_reason_create_container_error, waiting_reason_image_pull_error, waiting_reason_oom_killed, waiting_reason_start_error, terminated_reason_oom_killed)
- container_cpu_utilization (over_container_limit)
- container_memory_utilization (over_container_limit)

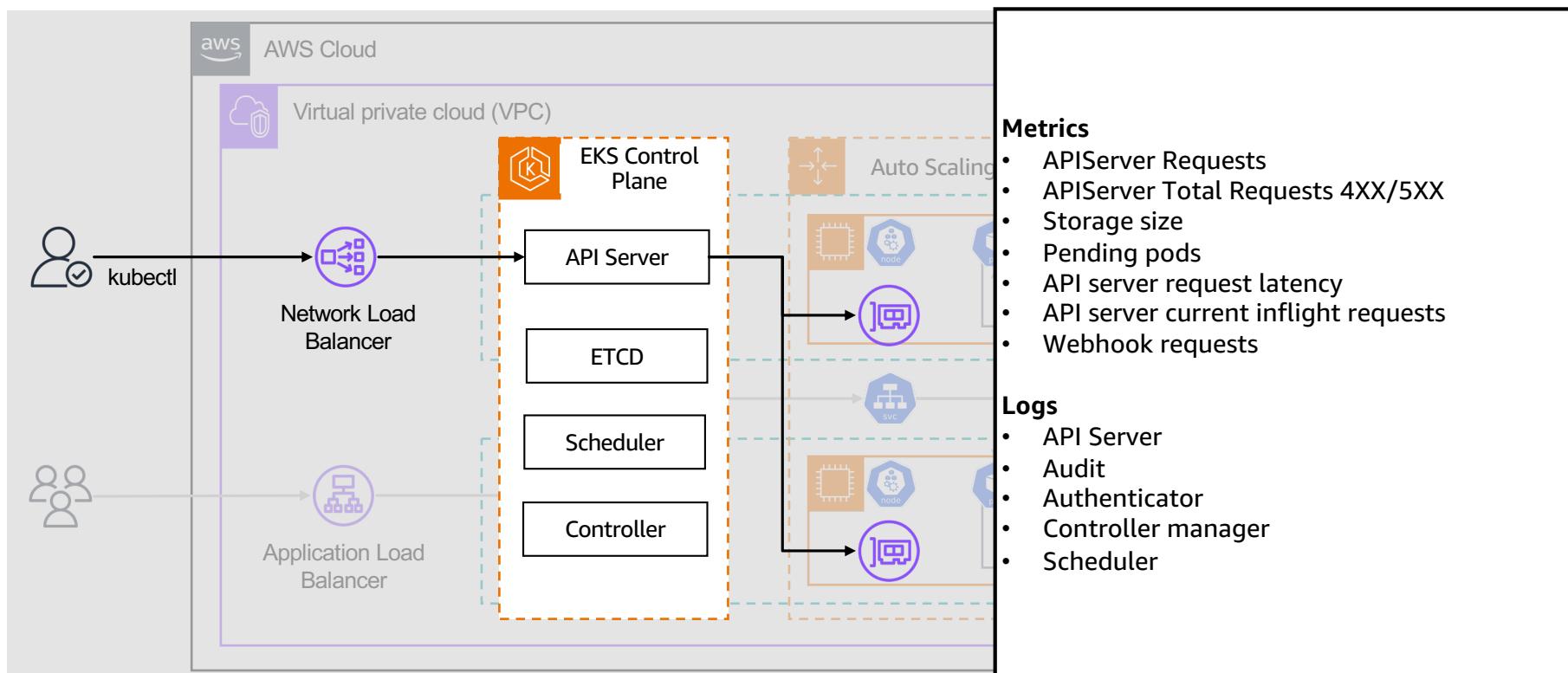


<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-metrics-enhanced-EKS.html>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Useful Metrics for EKS Cluster – Control Plane



<https://docs.aws.amazon.com/eks/latest/userguide/observability-dashboard.html#observability-control-plane>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Troubleshooting & Monitoring scenario

Scenario 1

- Disk Pressure issue

Scenario 2

- Control-plane monitoring

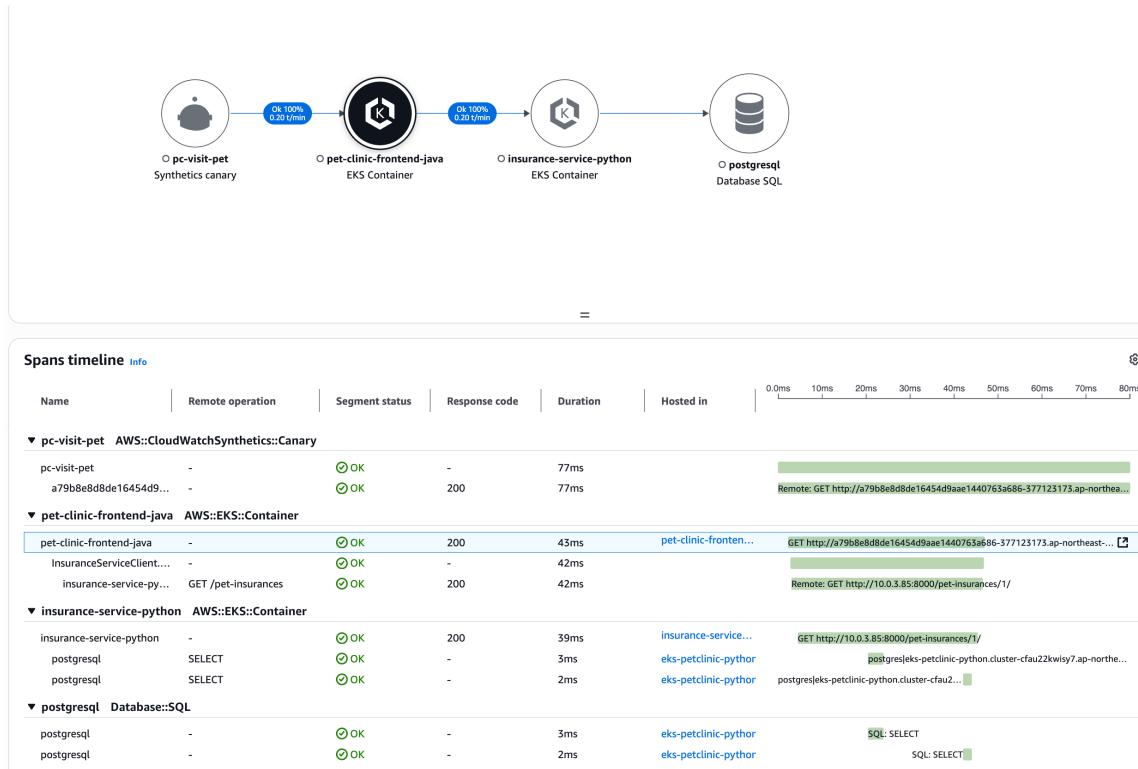


Scenario 1

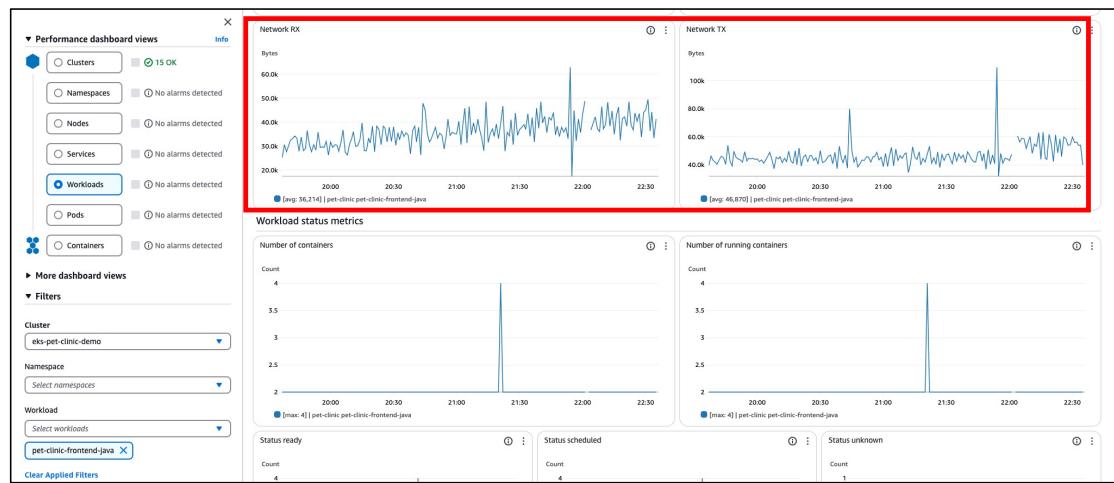
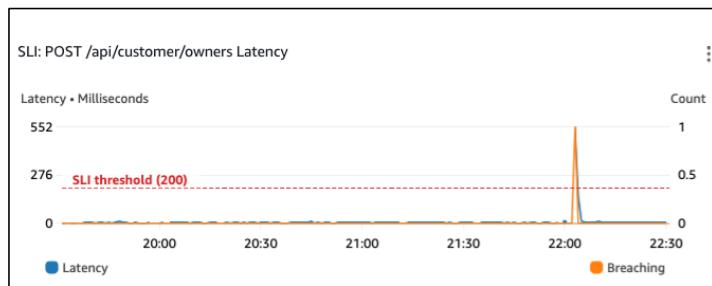
Service Level Objectives (SLO) (9) Info											Action
Filter SLOs by text, property or value											
SLO name	Goal	SLI status	Latest attainment	Attainment delta	Latest error budget	Error budget delta	Operation	Dependency			
Latency for Registering an Owner	99.9% 1 day rolling	🔴 Unhealthy	🟡 99.9%	-0.1%	⚠️ 30.2% 26 seconds remainin	-69.7%	POST /api/customer/owners	-			
Availability for fronted with payment ...	99.9% 1 day rolling	🔴 Unhealthy	🔴 0%	-	🔴 -99,900% 31968 requests ove	-	All operations	payment-service-dotnet			
Availability for Retrieving Payments	99.9% 1 day rolling	🔴 Unhealthy	🔴 98.6%	-0.1%	🔴 -1,219.7% 17 minutes over bur	0%	GET /owners/{ownerId:int}/pets/{pet...}	-			
Availability for Registering an Owner	99.9% 1 day rolling	🟢 Healthy	🟡 100%	0%	🟡 100% 1 minute remainin	0%	POST /api/customer/owners	-			
Latency for Searching an Owner	99.9% 1 day rolling	🟢 Healthy	🟡 99.9%	0%	🟡 100% 1 minute remaining	0%	GET /api/customer/owners	-			
Availability for Searching an Owner	99.9% 1 day rolling	🟢 Healthy	🟡 100%	0%	🟡 100% 1 minute remaining	0%	GET /api/customer/owners	-			
Latency for fronted with payment dep...	99.9% 1 day rolling	🟢 Healthy	🟡 100%	-	🟡 100% 1 minute remaining	-	All operations	payment-service-dotnet			
Availability for Retrieving Insurances	99.9% 1 day rolling	🟢 Healthy	🟡 100%	-	🟡 100% 48 requests remaini	-	GET /pet-insurances	-			
Latency of billing activities	99.9% 1 day rolling	🟢 Healthy	🟡 99.9%	0%	⚠️ 30.2% 26 seconds remainin	68.6%	GET ^billings/\$	-			



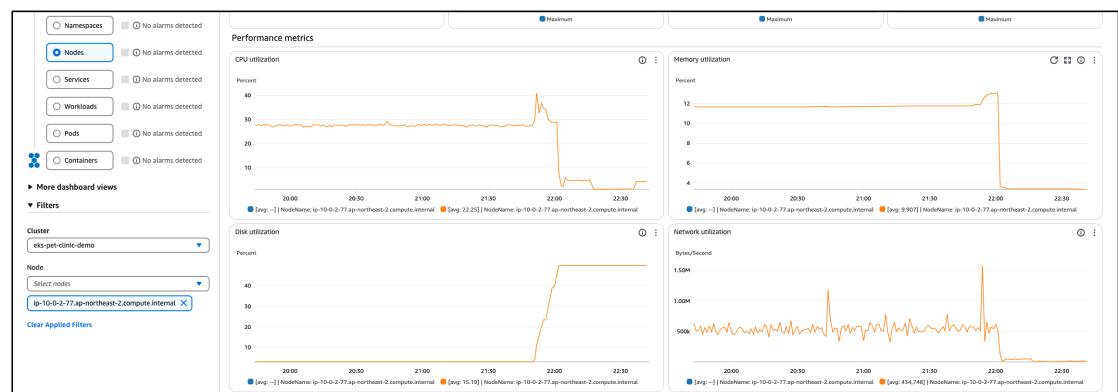
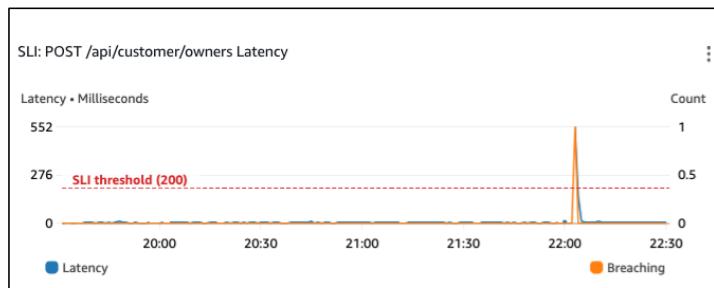
Scenario 1



Scenario 1

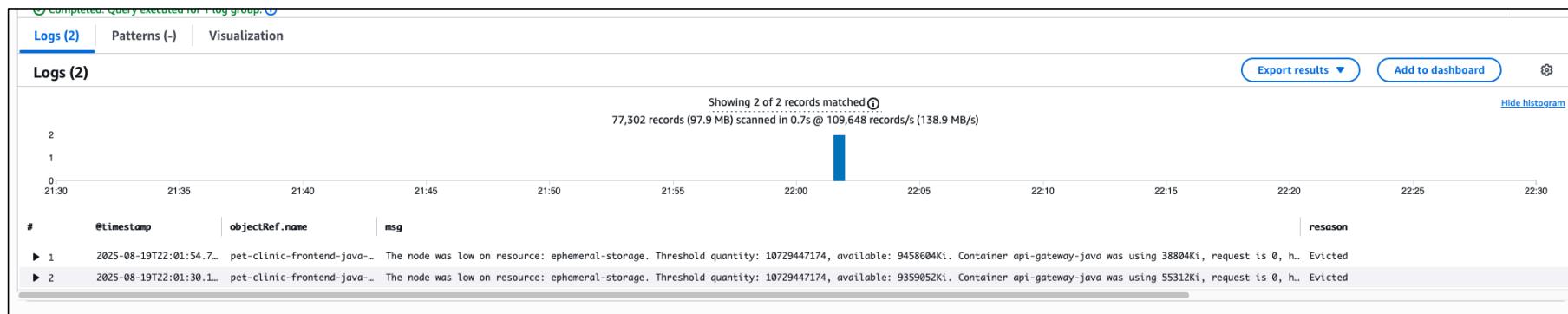


Scenario 1

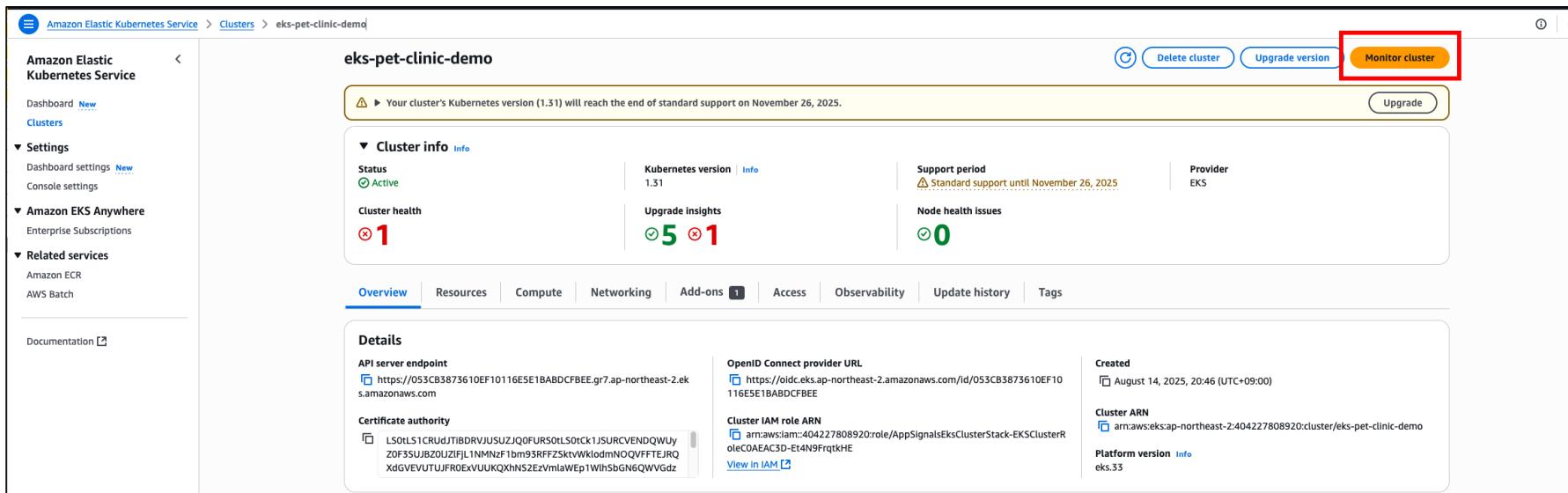


Scenario 1

```
fields @timestamp, objectRef.name, responseObject.message as msg, responseObject.reason as resason
| filter requestObject.reason == 'Evicted'
| filter objectRef.name like 'pet-clinic-frontend-java'
| sort @timestamp desc
| limit 10000
```



Scenario 2



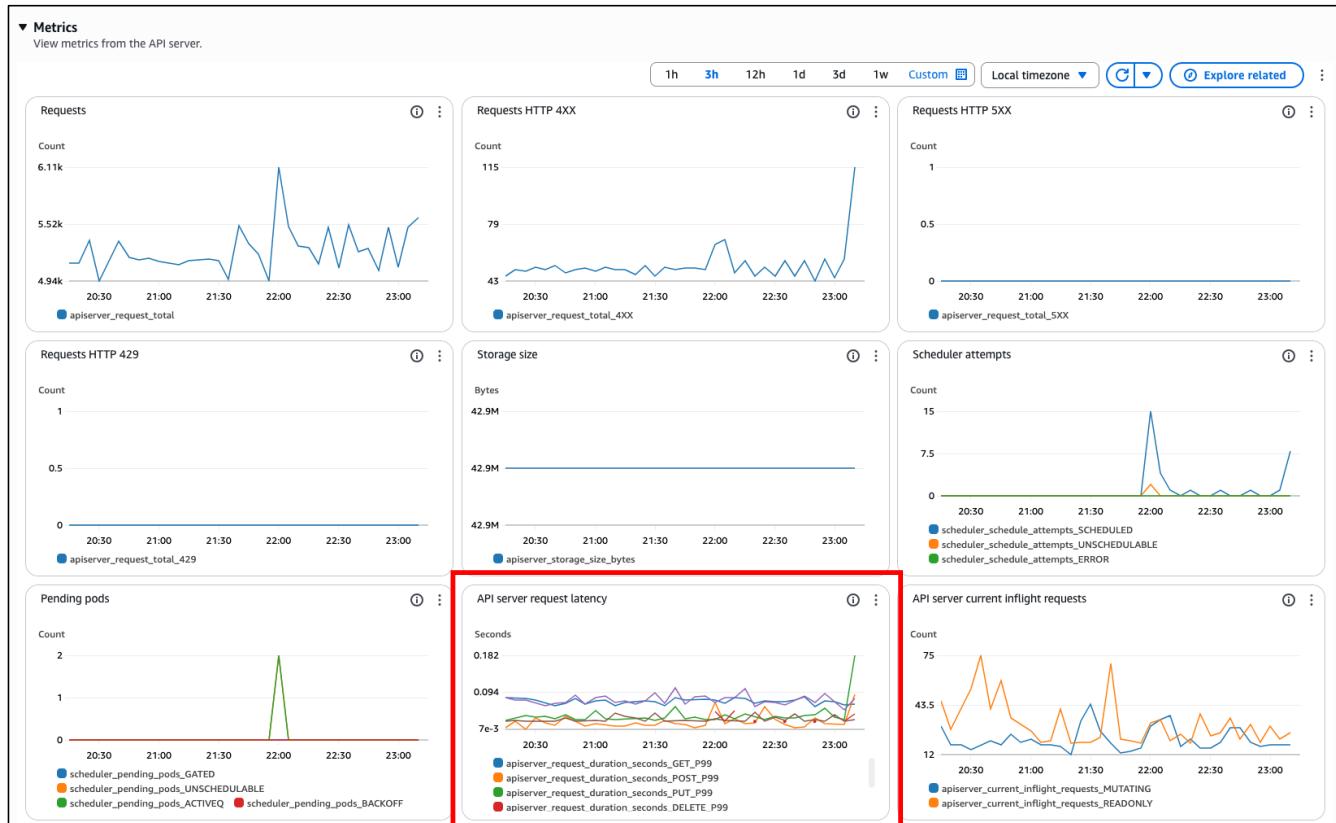
The screenshot shows the AWS EKS Cluster Overview page for the cluster `eks-pet-clinic-demo`. The left sidebar includes links for Amazon Elastic Kubernetes Service, Dashboard, Clusters, Settings (Dashboard settings, Console settings), Amazon EKS Anywhere (Enterprise Subscriptions), and Related services (Amazon ECR, AWS Batch). The main content area displays cluster details:

- Cluster info:** Status is Active (green), Kubernetes version is 1.31, Support period is Standard support until November 26, 2025, and Provider is EKS.
- Cluster health:** Shows 1 critical issue (red).
- Upgrade insights:** Shows 5 green icons and 1 red icon.
- Node health issues:** Shows 0 issues.

The navigation bar at the bottom includes tabs for Overview, Resources, Compute, Networking, Add-ons (1), Access, Observability, Update history, and Tags. The Details section provides API server endpoint (<https://053CB3873610EF10116E5E1BABDCFBE.gr7.ap-northeast-2.eks.amazonaws.com>), OpenID Connect provider URL (<https://oidc.eks.ap-northeast-2.amazonaws.com/id/053CB3873610EF10116E5E1BABDCFBE>), Certificate authority (long hex string), Cluster IAM role ARN (<arn:aws:iam::404227808920:role/AppSignalsEksClusterStack-EKSClusterRoleCOAEAC3D-Et4n9FrqtkHE>), Created (August 14, 2025, 20:46 UTC+09:00), Cluster ARN (<arn:aws:eks:ap-northeast-2:404227808920:cluster/eks-pet-clinic-demo>), and Platform version (Info) (eks.33).



Scenario 2



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Scenario 2

▼ CloudWatch Logs Insights
Each of these lists offers the option to run queries directly. To customize query parameters or change the query content, choose the applicable "View in CloudWatch" button.

ⓘ • Your audit logs are turned on. Choose any of the run queries options that you want to view.
• For future reference, you can view and manage the control plane Kubernetes audit logs with the [Manage logging](#) page.
• An additional cost will incur from CloudWatch if you run queries to see results from CloudWatch Logs Insights. [View CloudWatch pricing](#)

Run all queries 5m 30m 1h 3h 12h Custom Local timezone

► Top talkers
Show the most frequent callers to the API server, sorted by the number of requests made. Run query View in CloudWatch

► HTTP 4XX generators (97)
Show API server requests that resulted in HTTP 4XX status codes (client-side errors), sorted by count. Query last run: August 19, 2025, 23:24 (UTC+09:00) Run query View in CloudWatch

► HTTP 5XX generators
Show API server requests that resulted in HTTP 5XX status codes (server-side errors), sorted by count. Run query View in CloudWatch

► Broken Webhooks
Show instances of webhook errors. Run query View in CloudWatch

► Throttled clients
Show API server clients that are being rate-limited due to exceeding usage limits. Run query View in CloudWatch

► Request duration
Show the API server requests that took the longest time to process (sorted in descending order of duration). Run query View in CloudWatch

► Top talkers (1000)
Show the most frequent callers to the API server, sorted by the number of requests made. Query last run: August 19, 2025, 23:27 (UTC+09:00) Run query View in CloudWatch

Request URI	Verb	Response status code	User agent	Count
/api/v1/namespaces/amazon-cloudwatch/configmaps/cwagent-clusterleader	get	200	amazon-cloudwatch-agent/v0.0.0 (linux/amd64) kubernetes/\$Format	6294
/apis/coordination.k8s.io/v1/namespaces/chaos-mesh/leases/chaos-mesh	get	200	chaos-controller-manager/v0.0.0 (linux/amd64) kubernetes/\$Format/leader-election	3999



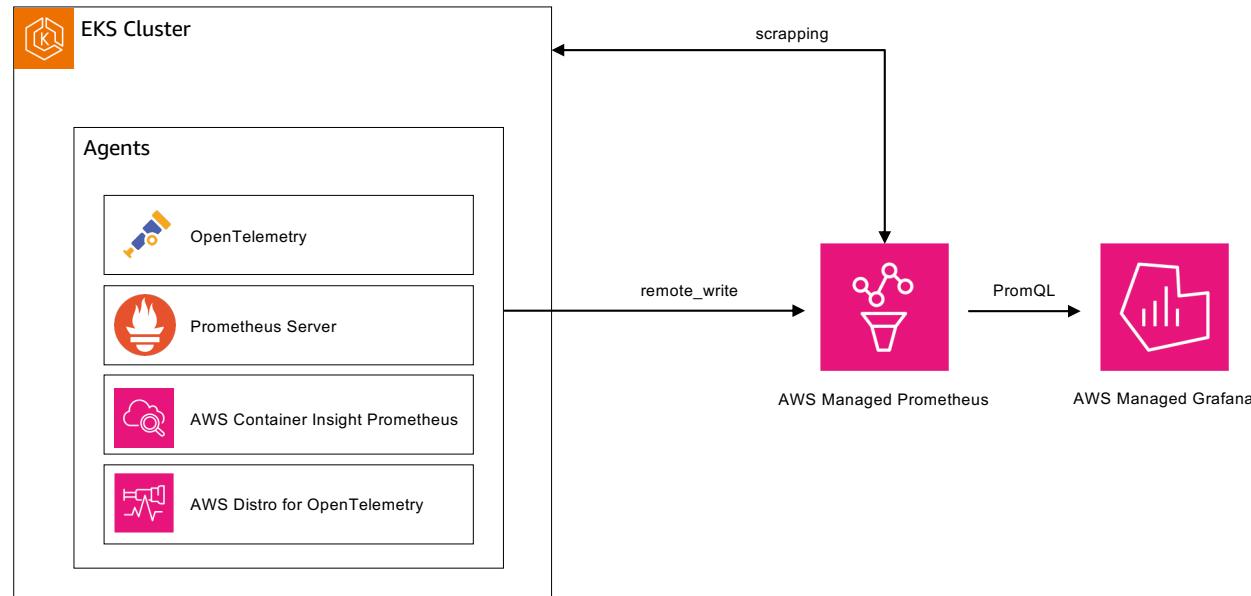
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Advanced Monitoring

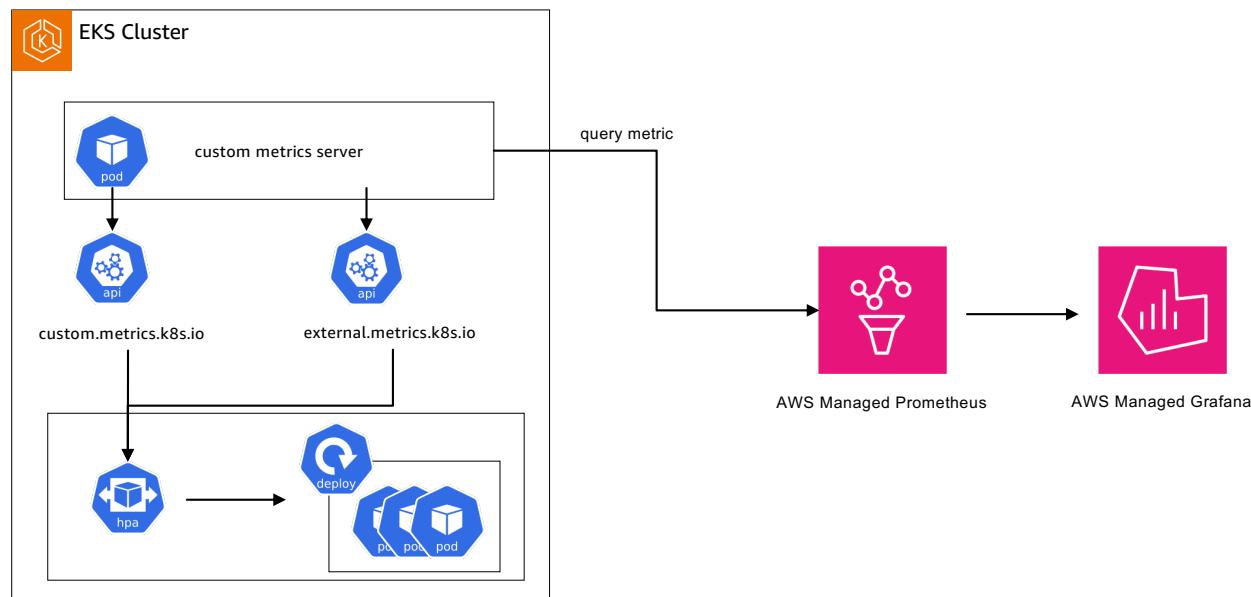


© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Prometheus



Prometheus



Example - Karpenter

Karpenter makes several metrics available in Prometheus format to allow monitoring cluster provisioning status.

(<https://karpenter.sh/docs/reference/metrics>)

NodeClaims

Pods

Nodes

Termination

Voluntary Disruption

Scheduler

NodePools

EC2NodeClass

Interruption

Cluster

Cluster State

CloudProvider

CloudProvider Batcher

Controller Runtime

Workqueue

Status Condition

Client Go

AWS SDK Go

Leader Election



Troubleshooting with Amazon Q

- [CloudWatch MCP Server](#)
- [CloudWatch Appsignals MCP Server](#)
- [Amazon EKS MCP Server](#)
- [Prometheus MCP Server](#)



Troubleshooting with Amazon Q

```
> 두 SLO의 상세 정보를 확인했습니다. attainment goal에 문제가 있는 이유를 분석해보겠습니다:

## 1. Availability for Retrieving Payments

문제점:
• **Attainment Goal: 99.9%** (매우 높은 목표)
• **실제 임계값: 99.0%** (가용성이 99% 이상이어야 함)
• **서비스:** payment-service-dotnet의 GET /owners/{ownerId:int}/pets/{petId:int}/payments 작업

문제 원인: 99.9% attainment goal은 하루 중 99.9%의 시간 동안 가용성이 99% 이상을 유지해야 한다는 뜻입니다. 즉, 하루 1440분 중 단 1.44분(약 86초)만 99% 미만으로 떨어져도 목표 달성을 실패합니다.

## 2. Latency of billing activities

문제점:
• **Attainment Goal: 99.9%** (매우 높은 목표)
• **실제 임계값: 5000ms** (P99 레이턴시가 5초 이하여야 함)
• **서비스:** billing-service-python의 GET /billings/$ 작업

문제 원인: 99.9% attainment goal로 설정되어 있어서 하루 중 거의 모든 시간 동안 P99 레이턴시가 5초 이하를 유지해야 합니다.

권장 해결책:
1. Attainment Goal을 현실적으로 조정 (예: 95% 또는 98%)
2. Warning Threshold 상향 조정 (현재 60%에서 80-90%로)
3. 임계값 재검토 (특히 레이턴시 5초는 너무 관대할 수 있음)
```



Troubleshooting with Amazon Q

```
## 결제 서비스 실패 요청 분석 결과

제공해 주신 X-Ray 트레이스 데이터를 분석한 결과, "Availability for Retrieving Payments" SLO 실패의 근본 원인을 확인했습니다:

### 🔍 핵심 문제점

DynamoDB ResourceNotFoundException 에러가 모든 실패 요청의 근본 원인입니다:

Amazon.DynamoDBv2.Model.ResourceNotFoundException: "Requested resource not found"

### 📈 실패 패턴 분석

1. 일관된 실패 경로: 모든 50개 트레이스가 동일한 패턴을 보입니다
   • pet-clinic-frontend-java + payment-service-dotnet + DynamoDB
   • 모든 요청이 HTTP 500 상태로 실패

2. DynamoDB 테이블 문제: PetClinicPayment 테이블에서 리소스를 찾을 수 없음
   • 테이블이 존재하지 않거나
   • 권한 문제이거나
   • 잘못된 테이블 이름/리전 설정

3. 서비스 체인 영향:
   • payment-service-dotnet에서 DynamoDB 호출 실패
   • 이로 인해 전체 요청이 500 에러로 실패
   • 프론트엔드까지 에러가 전파됨

### ⚡ 해결 방안

#### 즉시 조치 (Critical)
1. DynamoDB 테이블 확인:
   bash
   aws dynamodb describe-table --table-name PetClinicPayment --region ap-northeast-2

2. IAM 권한 검증:
   • payment-service-dotnet의 서비스 역할이 DynamoDB 테이블에 접근 권한이 있는지 확인
   • 필요한 권한: dynamodb:GetItem, dynamodb:Query, dynamodb:Scan

3. 애플리케이션 설정 검토:
   • DynamoDB 테이블 이름 설정 확인
   • AWS 리전 설정 확인
   • 연결 설정 검증

#### 중요 기록 (Important)
1. SLO 목표 조정: 99.9% → 95-98%로 현실적으로 조정
2. 에러 처리 개선: DynamoDB 실패 시 적절한 fallback 로직 구현
3. 모니터링 강화: DynamoDB 레벨 메트릭 추가

### 📈 비즈니스 영향

• **100% 실패율**: 결제 주최 기능이 완전히 작동하지 않음
• **사용자 경험**: 모든 결제 관련 요청이 실패
• **SLO 달성을 불가능**: 인프라 문제로 인한 구조적 실패

이 문제는 인프라 레벨의 구성 오류로, DynamoDB 테이블 설정을 수정하면 즉시 해결될 것으로 예상됩니다.
```



QnA?



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

64

Date	Title	Speaker
9월 4일(목) 19:00	EKS 모니터링 꿀팁 : 여러분의 클러스터는 안녕하신가요?	곽명섭 이수정 현창종
9월 25일(목) 19:00	EKS에서 Karpenter 아무지게 활용하기	박동혁 이호성 황은빛
10월 말 예정 추후 맛업 공지	EKS 신규 기능으로 달달하게 운영하기 (Auto Mode, Auto Repair, Monitoring, ...)	김보영 박진희 하영

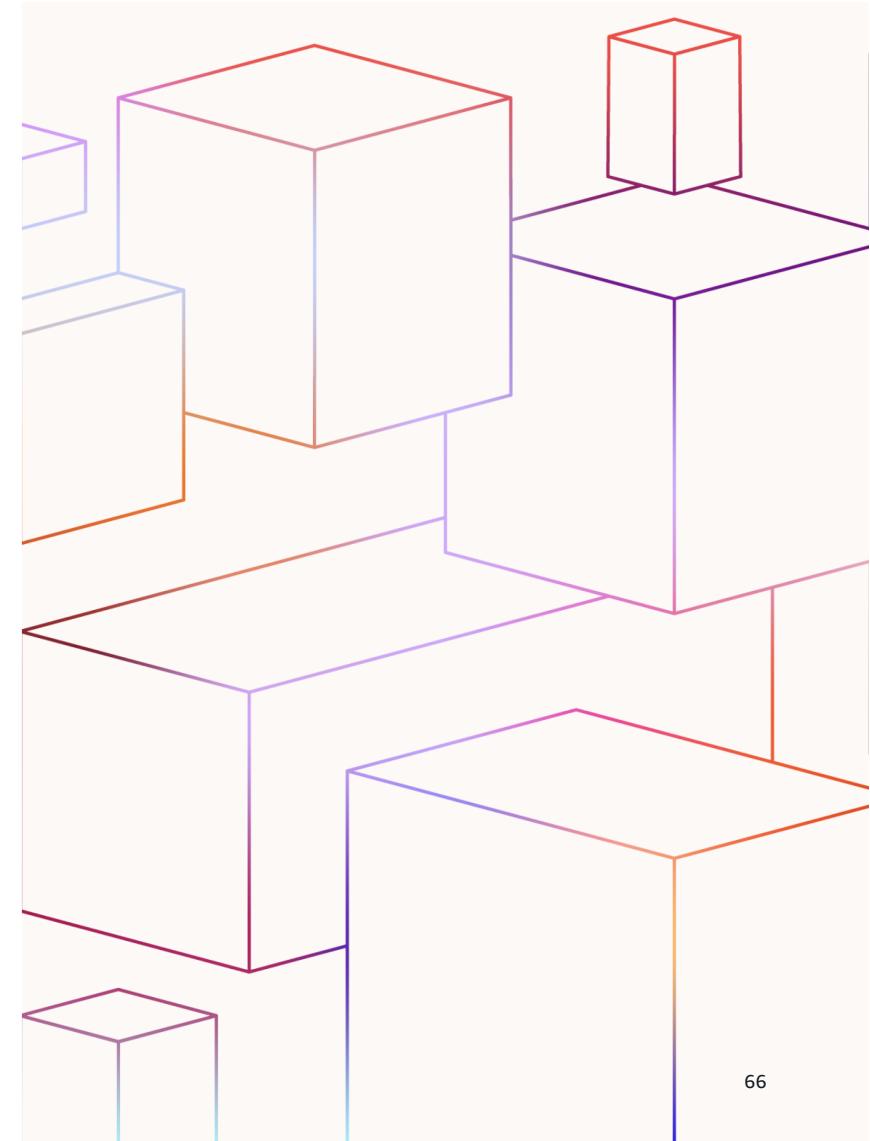
Your Feedback Matters



<https://pulse.amazon/survey/RJDHJBES>



© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.



Thank you!

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

