

CloudStack Infrastructure Monitoring using Grafana Stack



Grafana Labs

By Kiran Chavala

About Me

- CloudStack Committer
- QA Engineer – ShapeBlue
- Involved with CloudStack/Forks since 2013 by providing technical support to various customers
- Open-source enthusiast
- Addicted to various Podcasts/Audiobooks
- Email : kiran.chavala@shapeblue.com / kirachavala@apache.org
- Blog: kiranchavala.in
- [Linkedin : https://www.linkedin.com/in/kiranchavala/](https://www.linkedin.com/in/kiranchavala/)
- Twitter : @kiranchavala



Agenda

- Why Infrastructure Monitoring Matters
- CloudStack Monitoring Challenges
- Various Infrastructure Monitoring tools overview
- Grafana Stack Overview
- Architecture & Data Flow
- Grafana Dashboards and Demo
- Lessons Learned & Best Practices
- Conclusion
- QA/ References



Grafana Labs

Why Infrastructure Monitoring Matters

Dynamic and Distributed Infrastructure

Modern environments are multi-zone and multi-hypervisor.

Workloads scale up and down continuously.

Failures can propagate quickly without early detection.

Visibility Enables Reliability



MONITORING PROVIDES REAL-TIME INSIGHT INTO PERFORMANCE AND UTILIZATION.



EARLY DETECTION REDUCES MTTR AND IMPROVES SERVICE UPTIME.



CORRELATED METRICS AND LOGS HELP IDENTIFY ISSUES FASTER.

CloudStack Monitoring Challenges

Fragmented Metrics

Metrics
exposed only
through
multiple API
calls.

Hypervisor
layers (KVM,
Xen, VMware)
report data
separately.

Storage,
Compute, and
Network
appear as
isolated
domains.

No unified
time-series
view across
the stack.

Scalability & Retention



PER-ZONE AND PER-CLUSTER
MONITORING NEEDS CUSTOM
EXPORTERS.



CLOUDSTACK FOCUSES ON
REAL-TIME OPERATIONS, NOT
LONG-TERM TRENDS.



NO NATIVE MECHANISM FOR
HISTORICAL PERFORMANCE
STORAGE OR ANALYSIS.

Hard to Visualize



LOGS AND DB STATS ARE
TEXT-BASED.



NO BUILT-IN CORRELATION
BETWEEN LOGS AND
METRICS.



TROUBLESHOOTING
REQUIRES MULTIPLE TOOLS
AND MANUAL
INTERPRETATION.

Various Infrastructure Monitoring tools overview



 OpenSearch

 ZABBIX

 Nagios

 new relic.

 splunk>

 DATADOG

 logz.io

Why I chose the Grafana Stack



**Lightweight &
Modular
Architecture**



**Massive Plugin &
Datasource
Ecosystem**



**Unified
Observability**

Grafana Stack Overview

- Open-source platform for metrics, logs, and traces.
- Unifies observability with a single pane of glass.
- Correlates data across layers, from hypervisor to API calls.



Grafana Labs

Key Capabilities

- Integrates with major open-source monitoring systems and external data sources.
- Supports MySQL, Elastic, Kubernetes.
- Provides alerting based on SLOs and SLIs.
- Sends alerts to Slack, PagerDuty, and Email.
- Offers RBAC with Ops/Admin/Viewer roles.
- Includes HTTPS, API keys, and SSO/LDAP support.

Grafana Integrations



Prometheus

Collects CloudStack and hypervisor metrics.



Alloy

Gathers logs and telemetry from hosts and system VMs.



Loki

Centralizes logs and correlates them with metrics.

Architecture & Data Flow



Grafana Dashboard CloudStack Prometheus Metrics



CloudStack Global Settings

```
prometheus.exporter.enable = true  
prometheus.exporter.allowed ips = <Prometheus-IP>
```



Prometheus Configuration

```
- job_name: 'cloudstack'  
  scrape_interval: 60s  
  scrape_timeout: 15s  
  metrics_path: /metrics  
  static_configs:  
    - targets: ['<Cloudstack-IP>:9595']
```



Sample metric

```
cloudstack_hosts_total{zone="ref-trl-4857-k-M7-kiran-chavala",filter="online"}  
1
```

Grafana Dashboard CloudStack Prometheus Metrics



Grafana Dashboard KVM Prometheus Metrics



Install the libvirt plugin on the kvm hosts

<https://github.com/inovex/prometheus-libvirt-exporter/releases>



Prometheus Configuration

```
- job_name: 'kvm'
  scrape_interval: 60s
  scrape_timeout:
  metrics_path: /metrics
  static_configs:
    - targets:
      ['<kvmhost1>:9177','<kvmhost2>:9177']
```



Sample metric

```
libvirt_domain_vcpu_current{domain="i-2-12-VM"} 2
```

Grafana Dashboard KVM Prometheus Metrics



Grafana Dashboard VMware Prometheus Metrics



vmware

Install the VMware Prometheus exporter

https://github.com/pryorda/vmware_exporter

https://github.com/grafana/vmware_exporter



Prometheus

Prometheus Configuration

- job_name: 'vmware_vcenter'
- metrics_path: '/metrics'
- scrape_timeout: 15s
- static_configs:
 - targets:
 - 'localhost:9272'



Grafana

Sample metric

vmware_host_cpu_max{host_name="esxihost1"} 28728.0

Grafana Dashboard VMware Prometheus Metrics



Grafana Dashboard XCP-ng Prometheus Metrics



Install the XCP-ng Prometheus exporter

<https://github.com/MikeDombo/xen-exporter>



Prometheus Configuration

```
- job_name: 'xcpng'
  scrape_interval: 60s
  scrape_timeout: 15s
  metrics_path: /metrics
  static_configs:
    - targets: ['localhost:9100']
```



Sample metric

```
xen_host_cpu{cpu="2", host_uuid="5c4b0287-7d2f-4740-897f-55e65630fc74", host="ref-trl-9870-k-mol8-kiran-chavala-xs1"} 0.17158
```

Grafana Dashboard XCP-ng Prometheus Metrics



Grafana Dashboard for CloudStack Logs



Install Grafana Alloy in CloudStack Management Server and configure it

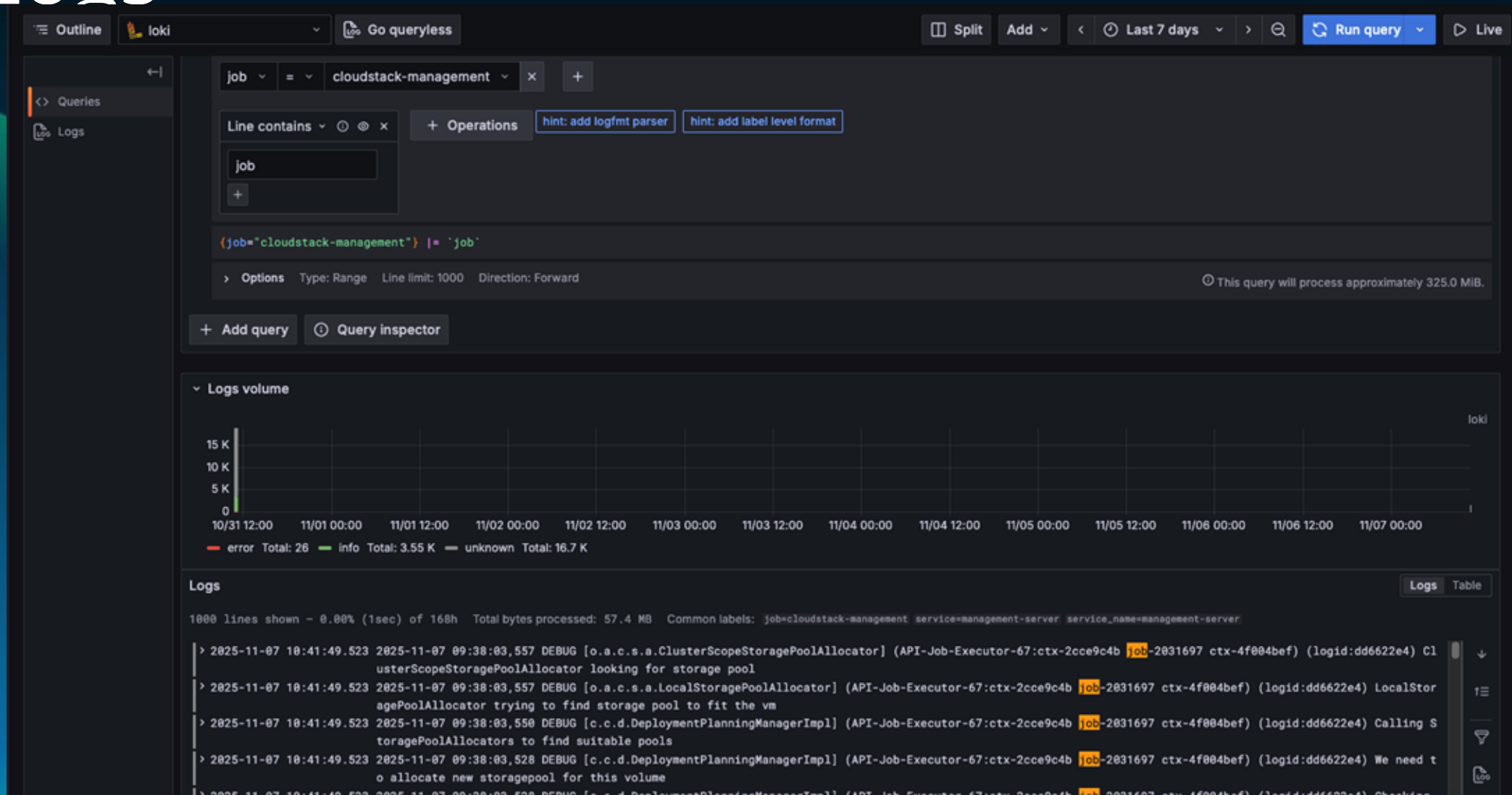
```
- local.file_match "cloudstack_management_logs" {
  path_targets = [
    {
      __path__ = "/var/log/cloudstack/management/management-server.log",
      job      = "cloudstack-management",
      service  = "management-server",
    },
  ]

  loki.source.file "management_logs" {
    targets      = local.file_match.cloudstack_management_logs.targets
    tail_from_end = true
    forward_to   = [loki.write.default.receiver]
  }

  loki.write "default" {
    endpoint {
      url = "http://loki-server:3100/loki/api/v1/push"
    }
  }
}
```

Add LOKI as a data source in Grafana Query the CloudStack Logs

Grafana Dashboard for CloudStack Logs



Grafana Dashboard for MySQL



```
CREATE USER 'grafana' IDENTIFIED BY 'password';  
GRANT SELECT ON cloud.* TO 'grafana';
```



Connection

Host URL *

Database name

Authentication

Username *

Password

Grafana Dashboard for CloudStack MySQL

Home > Dashboards > Sofia-lab-mysql-dashboard > Edit panel

Q Search... ⌘+k + ⓘ

← Back to dashboard Discard panel changes Save dashboard

account_name ⓘ kiran.chavala

Table view ⓘ Last 6 hours ⓘ Refresh ⓘ

Sofia-lab-mysql-dashboard ⓘ

account_id	name	display_text	vm_count	type
10423	kiran.chavala	kiran.chavala@shapeblue.com	16	Account
23606	ref-tri-9433-k-Mol8-kiran-chavala-	kiran-upstream-kiran.chavala	5	Project
23779	ref-tri-9490-k-Mol8-kiran-chavala-	kiran-proxmox-vnc-kiran.chavala	5	Project
24293	ref-tri-9690-k-Mol8-kiran-chavala-	kiran-upstream-kiran.chavala	3	Project
24513	ref-tri-9870-k-Mol8-kiran-chavala-	kiran-ccc-poc-kiran.chavala	9	Project
24659	ref-tri-9967-k-Mu24-kiran-chavala-	kiran-ubutnu24-422rc3-kiran.chavala	3	Project
24698	ref-tri-9998-k-Mol8-kiran-chavala-	kiran-upstream-hatest-kiran.chavala	5	Project
24703	ref-tri-10001-k-Mol8-kiran-chavala-	kiran-pr-11962-kiran.chavala	2	Project
TOTAL			48	

Queries 1 Transformations 0

Format: Table

```
4 display_text,
5 vm_count,
6 type
7 FROM (
8   SELECT
9     a.id AS account_id,
10    a.account_name AS name,
11    u.email AS display_text,
12    COUNT(v.id) AS vm_count,
13    'Account' AS type
14  FROM account a
15  JOIN user u
16    ON u.account_id = a.id
17  LEFT JOIN vm_instance v
```

Run query Builder Code

Visualization

Table

Panel options

Title
Sofia-lab-mysql-dashboard

Description
Sofia-lab-mysql-dashboard

Transparent background

Panel links

Repeat options

Table

Show table header

Frozen columns
Columns are frozen from the left side of the table
none

Cell height
Small Medium Large

Max row height
none

Enable pagination

Grafana Alerts

- Send alerts to various touch points. E.g.: Slack, Pagerduty, e-mail

Home > Alerting > Contact points

Contact points

Choose how to notify your contact points when an alert instance fires

Create contact point

Name *

Integration

OpsGenie
PagerDuty
Pushover
Sensu Go
Slack
Telegram
Threema Gateway
VictorOps

Save contact point Cancel

Home > Alerting > Alert rules > Edit alert rule

Edit alert rule

datastore alert

Alert evaluation currently paused
Notifications for this rule will not fire and no alert instances will be created until the rule is un-paused.

1. Enter alert rule name

Enter a name to identify your alert rule.

Name

2. Define query and alert condition

Define query and alert condition [Need help?](#)

prometheus Options 10 minutes, MD = 43200, Min. Interval = 1s

Click start your query Explain

Run queries Builder Code

Metrics browser > `vmware_datastore_freespace_size(ds_name="04f088a4eb8d3e5e929308436a5efdb9") / vmware_datastore_capacity_size {ds_name="04f088a4eb8d3e5e929308436a5efdb9"}`

Options Legend: Auto Format: Time series Step: auto Type: Instant

Alert condition

WHEN QUERY IS ABOVE 60

Preview alert rule condition

3. Add folder and labels

Organize your alert rules with a folder and set of labels. [Need help?](#)

Lessons Learned & Best Practices

Start Small

1

Begin with a minimal Prometheus + Grafana setup.

2

Add hypervisor exporters gradually (KVM, Xen, VMware).

3

Integrate Alloy and Loki only after validating core metrics.

Use Consistent Naming

1

Define a naming convention early.

2

Apply labels consistently (zone, cluster, host, vm_id).

3

Consistent labels simplify dashboards and alerting.

Centralize Dashboards

1

Keep shared dashboards in a central Grafana organization.

2

Standardize templates for prod/test using folders and variables.

3

Reuse the same dashboard logic across multiple sites.

Conclusion

CloudStack is a Rich Data Source

- Generates valuable metrics across all layers.
- The challenge is visibility, not availability.
- With exporters and pipelines, it becomes a strong observability platform.



Grafana Stack Delivers Unified Observability

- Grafana + Prometheus + Loki + Alloy form a cohesive ecosystem.
- Vendor-neutral and open-source.
- Enables cross-layer insight: management, hypervisor, tenant workloads.



Grafana Labs

Unified Monitoring Improves Reliability



DASHBOARDS ACCELERATE
ROOT-CAUSE ANALYSIS.



PROACTIVE ALERTS REDUCE
DOWNTIME AND BLIND
SPOTS.



TEAMS CAN DETECT
ANOMALIES AND FORECAST
RESOURCE NEEDS IN REAL
TIME.

References

- **Official Resources**

- Grafana Dashboards Library: community-built dashboards for CloudStack, VMware, and Xen.

- **Community Exporters**

- CloudStack Monitoring GitHub: exporter configs, dashboards, setup guides.

<https://github.com/kiranchavala/cloudstack-monitoring>

- VMware Monitoring Blog: collecting VMware cluster metrics with Prometheus.

<https://michaelkotelnikov.medium.com/monitoring-vmware-clusters-using-prometheus-and-grafana-6223bb7936f9>

- **Prebuilt Dashboards**

- CloudStack & Libvirt Dashboard (Grafana ID: 23228).
- Xen Hypervisor Dashboard (Grafana ID: 16588).



Q&A

