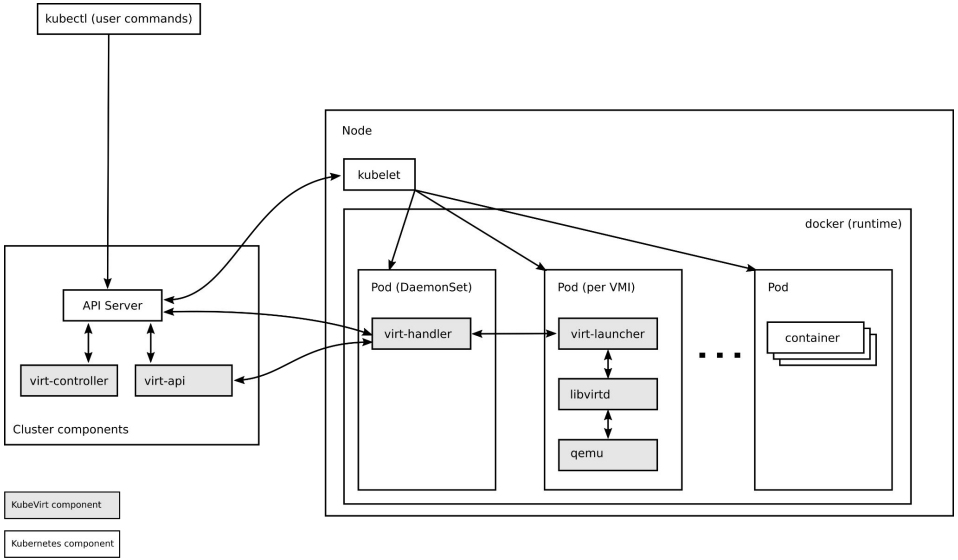
KubeVirt Observability with Prometheus and Grafana

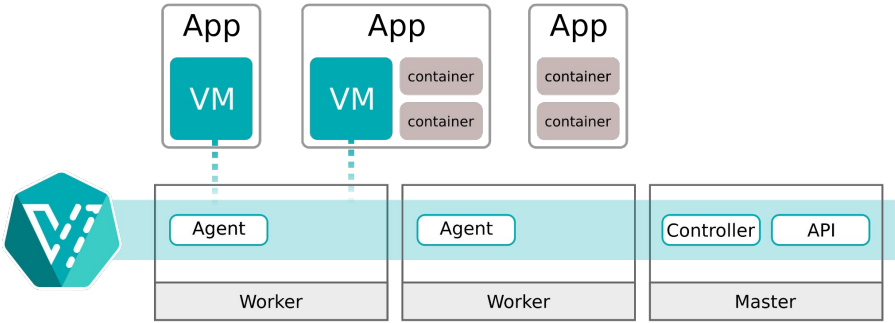
Ananda Dwi Rahmawati

Goals

- Provide leading indicators of an outage or service degradation.
- Help debug and detect outages, service degradations, bugs, and unauthorized activity.
- Identify long-term trends for capacity planning and business purposes.
- Expose unexpected side effects of changes

KubeVirt





KubeVirt Architecture

“In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs.”

KubeVirt Observability

- KubeVirt is an extension of Kubernetes, providing a way to run and manage virtual machines (VMs) alongside container workloads.
- Observability in KubeVirt involves monitoring, logging, and tracing to ensure the health, performance, and reliability of the VMs and the underlying infrastructure.

“Monitoring tells you whether a system is working; Observability lets you understand why isn't working.”

Goals

Key Metric to Monitor

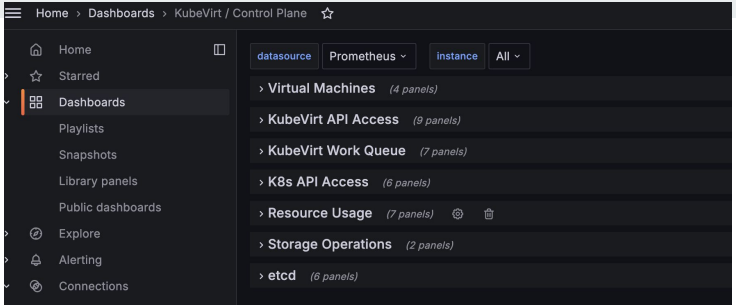
- Latency
- Error Rates
- Resource Utilization
- User Activity

Implementations

- Automated alerts
- Dashboard
- Regular review

KubeVirt

Monitoring



- **Prometheus**: A popular monitoring tool for Kubernetes, Prometheus can be configured to scrape metrics from KubeVirt components.
    - **KubeVirt Metrics**: KubeVirt exposes metrics that Prometheus can scrape.
    - All KubeVirt system-components expose Prometheus metrics at their /metrics REST endpoint.
- **Grafana**: For visualization, Grafana can be used to create dashboards based on metrics collected by Prometheus.
    - Dashboards: Pre-built dashboards can be imported to visualize KubeVirt-specific metrics.
- **KubeVirt Monitoring Operator**: Simplifies the setup of monitoring tools for KubeVirt by providing custom resource definitions (CRDs) to manage monitoring components.

```
root@master:/home/ubuntu# kubectl get endpoints -n kubevirt kubevirt-prometheus-metrics -o yaml
apiVersion: v1
kind: Endpoints
metadata:
  creationTimestamp: "2024-06-23T13:18:34Z"
  labels:
    app.kubernetes.io/component: kubevirt
    app.kubernetes.io/managed-by: virt-operator
    kubevirt.io: ""
    prometheus.kubevirt.io: "true"
    service.kubernetes.io/headless: ""
  name: kubevirt-prometheus-metrics
  namespace: kubevirt
  resourceVersion: "19080"
  uid: 0ff74c35-e089-46f1-841f-29dfa5d5f66c
subsets:
- addresses:
  - ip: 10.244.171.74
    nodeName: worker
    targetRef:
      kind: Pod
      name: virt-operator-865f487cf6-jtrgf
      namespace: kubevirt
      uid: be746582-7f82-40f8-8d51-456718245df3
  - ip: 10.244.171.75
```

KubeVirt

Strategies

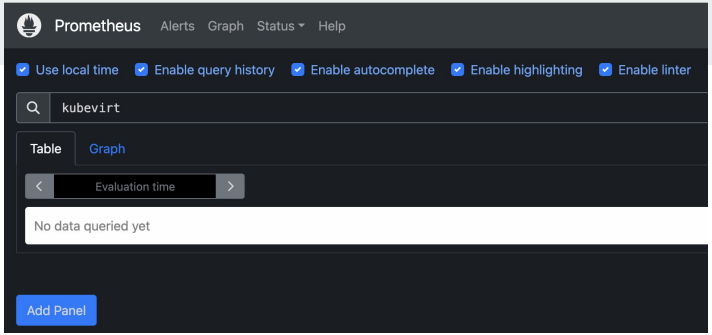
- Log Analysis
- Distributed Tracing
- Health Checks
- Security Monitoring

KubeVirt

Monitoring

Three settings are exposed in the KubeVirt custom resource to direct KubeVirt to create these resources correctly:

- **monitorNamespace**: The namespace that prometheus-operator runs in. Defaults to openshift-monitoring.
- **monitorAccount**: The serviceAccount that prometheus-operator runs with. Defaults to prometheus-k8s.
- **serviceMonitorNamespace**: The namespace that the serviceMonitor runs in. Defaults to be monitorNamespace



kubevirt_allocatable_nodes
The number of allocatable nodes in the cluster. Type: Gauge.
kubevirt_api_request_deprecated_total
The total number of requests to deprecated KubeVirt APIs. Type: Counter.
kubevirt_configuration_emulation_enabled
Indicates whether the Software Emulation is enabled in the configuration. Type: Gauge.

KubeVirt

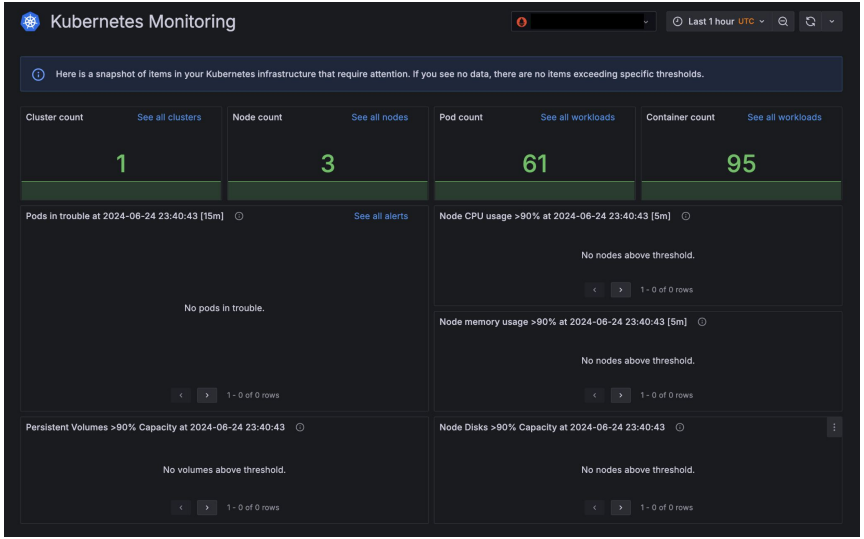
“Good Observability allows you to answer the questions you didn't know that you needed to ask”

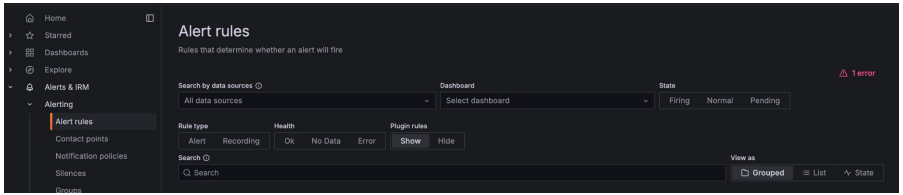
KubeVirt

Alerting

It allows you to create, manage, and execute alerts based on your monitoring data. We can receive notifications when your metrics or logs meet certain conditions, enabling you to respond quickly to potential issues.

- **Alert Rules:** Define rules that specify the conditions under which alerts are triggered.
- **Notification Channels:** Configure various methods for alert notifications, such as email, Slack, PagerDuty, and others.
- **Alert Groups**: Group related alerts for better organization and management.
- **Multi-dimensional Alerts**: Create alerts based on multiple conditions across different data sources.
- **Silencing and Inhibition**: Temporarily silence alerts or prevent alerts from triggering based on specific conditions.





KubeVirt

Tools and Challenges

- Want to be able to get a 360◦ view of a problem
- Need to correlate logs, metrics and traces to get deeper insights
- Repetitive troubleshooting process
- Data introspection

KubeVirt

References

- https://kubevirt.io/user-guide/architecture/
- https://github.com/kubevirt/monitoring
- https://github.com/prometheus-community/helm-charts
- https://github.com/kubevirt/monitoring/blob/main/dashboards/grafana/kubevirt-control-plane.json
- https://kubevirt.io/2019/An-overview-to-KubeVirt-metrics.html
- https://github.com/enp0s3/kubevirt-monitoring-operator
- https://github.com/kubevirt/kubevirt-tutorial/tree/main

KubeVirt