Module 7 OpenShift Monitoring

Cluster, Project and App Monitoring

Application Health

Liveness and Readiness

Liveness: Container is up and running

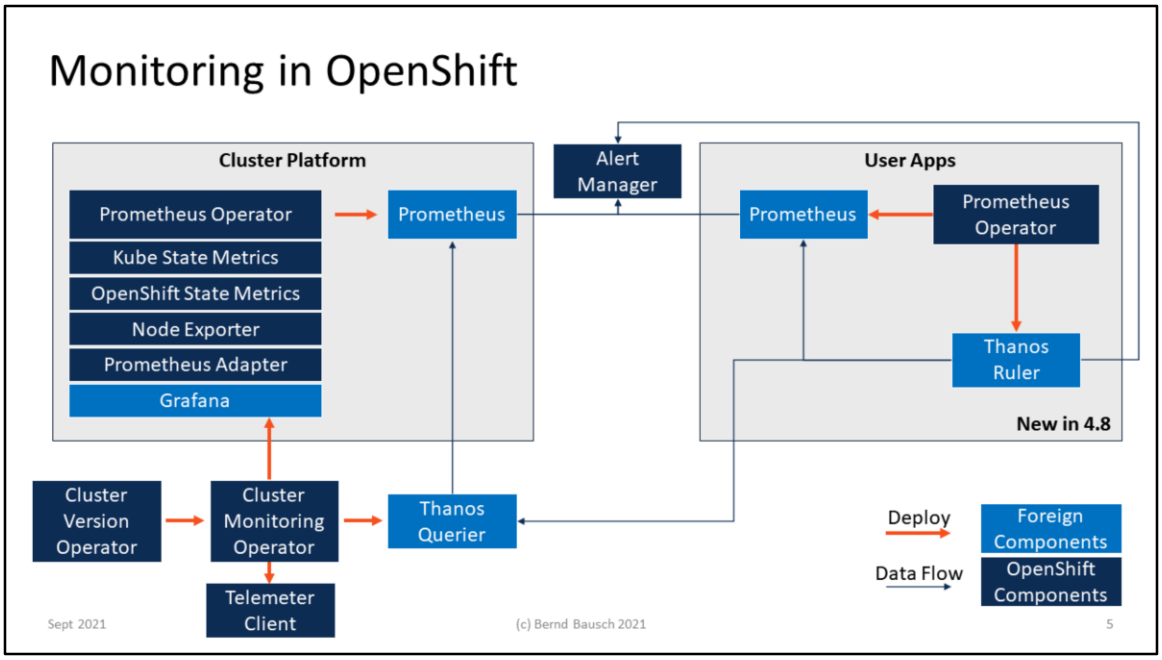
Readiness: Application is up and running

How to check?

- Run command in container
- HTTP GET
- TCP Connection request

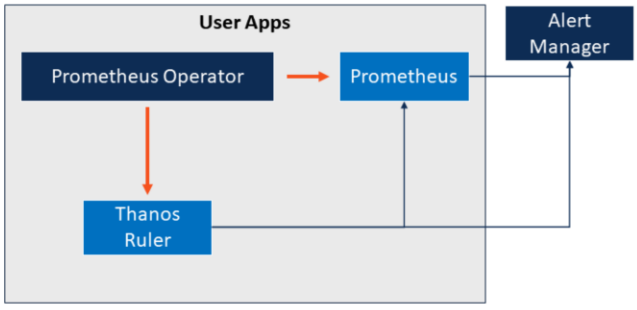
The meaning of Liveness and Readiness is ultimately defined by the application developer. How to determine liveness and readiness is also application specific.

OpenShift Monitoring



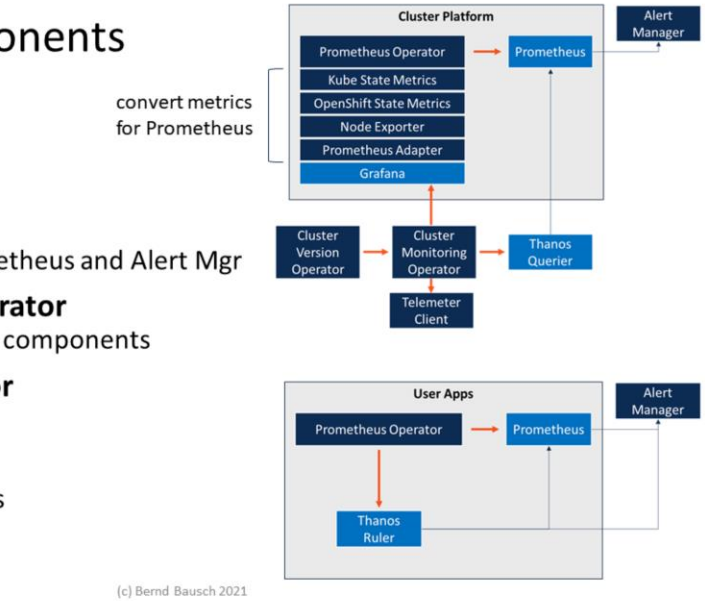
The OpenShift monitoring solution is based on open-source project Prometheus, which consists of a time-series store and a metrics evaluation and alerting component.
Prometheus uses HTTP requests to pull metrics data from the platform on which it runs. This includes data about OpenShift and Kubernetes objects but also platform data like nodes' CPU utilization and disk space.
To provide data in a format that suits Prometheus, OpenShift deploys a number of components that translate host, Kubernetes and OpenShift metrics. They run on the control plane; in addition, node exporters run on all hosts.
These components are deployed and managed by the Cluster Monitoring Operator.
On production installations, you can also have a telemetry client that sends data to Red Hat for product improvement purposes.
Starting with OpenShift 4.8, not only the cluster platform but also user workloads can be monitored with Prometheus.

Monitoring in OpenShift



Monitoring Components

- **Prometheus**
  Time-series database and metrics rule evaluation
- **Prometheus Operator**
  Deploys and manages Prometheus and Alert Mgr
- **Cluster Monitoring Operator**
  Deploys and manages other components
- **Cluster Version Operator**
  Deploys CMO
- **Alert Manager**
  Receives and manages alerts



Configuring Monitoring

Centralized in two config maps

Configuration examples:

- Retention interval
- Storage
- Log levels
- Labels for time series and alerts
- Sample limits

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cluster-monitoring-config
  namespace: openshift-monitoring
data:
  config.yaml: |
    enableUserWorkload: true
    prometheusK8s:
      retention: 12h
      volumeClaimTemplate:
       spec:
         volumeMode: Filesystem
         resources:
           requests:
             storage: 5Gi
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
    prometheus:
      retention: 12h
      resources:
        requests:
          cpu: 200m
          memory: 2Gi
```


Above, an example for the platform monitoring config map. It configures a data retention period of 12 hours and requests 5GB storage for the Prometheus component. The PVC does not request a particular storage class and will therefore be satisfied from the default storage class. In the case of Codeready Containers, this is the predefined storage class with 20 volumes.

Below, the configuration for monitoring user projects. This example also defines a retention period and requests CPU and memory resources for the Prometheus component.

Note that the two config maps reside in different namespaces (i.e., OpenShift projects). It is common for OpenShift infrastructure components to manage their resources in their own namespaces.

The same Prometheus component is named *prometheus* for user project configuration and *prometheusK8s* for cluster-level monitoring.

In both config maps, other components like the operators, the metrics components or the alert manager can be configured in similar ways.

RBAC for user workload monitoring

Role	Meaning
monitoring-rules-view	Read Prometheus monitoring rules in a project (*PrometheusRule* object)
monitoring-rules-edit	Modify Prometheus monitoring rules in a project
monitoring-edit	Like *monitoring-rules-edit*, plus • create scrape targets • handle ServiceMonitor and PodMonitor resources
user-workload-monitoring-config-edit	Edit the config map for user monitoring

```
$ oc policy add-role-to-user monitoring-rules-edit developer -n devproject
$ oc adm policy add-role-to-user user-workload-monitoring-config-edit developer \
  --role-namespace openshift-user-workload-monitoring                           \
  --namespace openshift-user-workload-monitoring
```

Sept 2021 (c) Bernd Bausch 2021 9

The table lists the roles that are relevant for monitoring user projects. The first three roles are about viewing, creating and modifying monitoring rules as well as adding metrics targets.
Users with the fourth role are able to configure the user project monitoring infrastructure.
The code below the table shows how a cluster administrator might bind those roles to users.

Retrieving Data