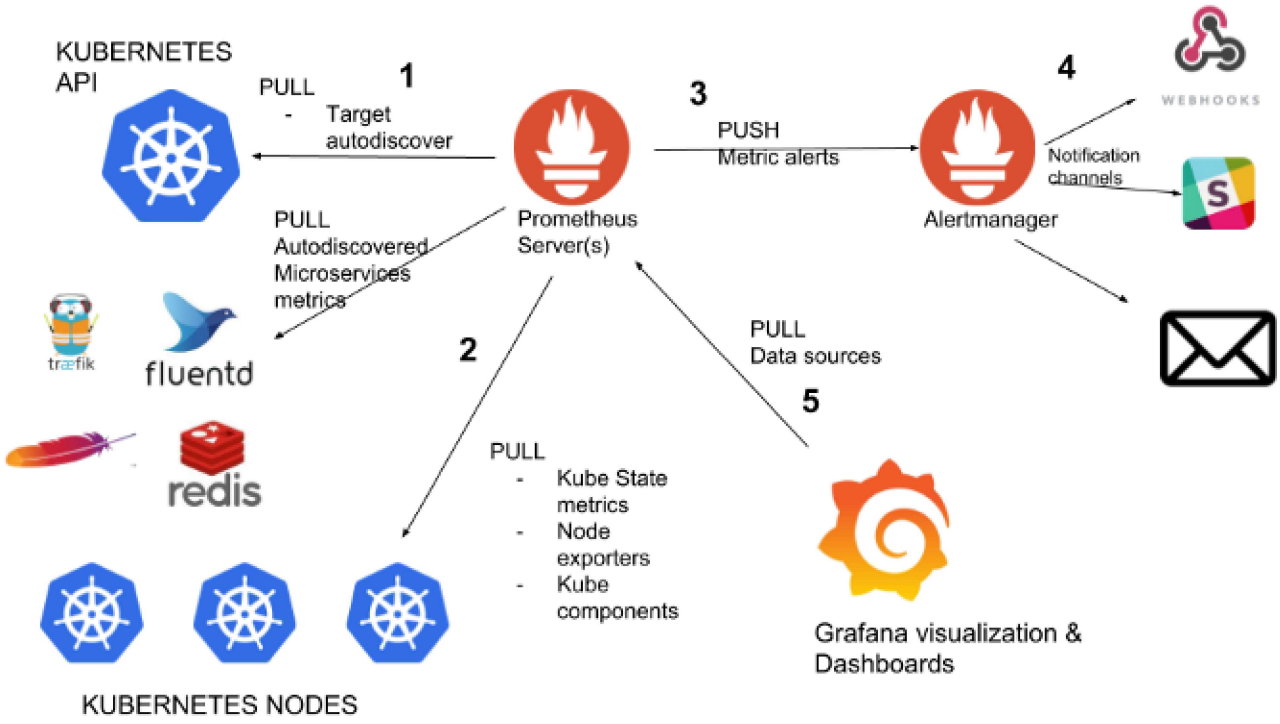
Prometheus monitoring with Elastic Stack in Kubernetes

Kubernetes Advocate
Mar 22 · 4 min read

Monitoring is one of the key components for managing large clusters. For this, we have a number of tools.



pic Credits https://mpraski.com/

Monitoring with Prometheus

It is a monitoring and alerting system. It was built at SoundCloud and was open-sourced in 2012. It handles the multi-dimensional data very well.

Prometheus has multiple components to participate in monitoring —

1. **Prometheus** — It is the core component that scraps and stores data.

2. **Prometheus node explores** — Gets the host level matrices and exposes them to Prometheus.
3. **Ranch-eye** — is a proxy and exposes cAdvisor stats to Prometheus.
4. **Grafana** — Visualization of data.
5. **InfuxDB** — Time series database specifically used to store data from ranchers.
6. **Prom-ranch-exporter** — It is a simple node.js application, which helps in querying the Rancher server for the status of a stack of services.

Description: Monitoring with Prometheus

Sematext Docker Agent

It is a modern Docker-aware metrics, events, and log collection agent. It runs as a tiny container on every Docker host and collects logs, metrics, and events for all cluster nodes and containers. It discovers all containers (one pod might contain multiple containers) including containers for Kubernetes core services if the core services are deployed in Docker containers. After its deployment, all logs and metrics are immediately available out of the box.

Deploying Agents to Nodes

Kubernetes provides DeamonSets which ensures pods are added to the cluster.

Configuring SemaText Docker Agent

It is configured via environment variables.

1. Get a free account at apps.sematext.com if you don't have one already.
2. Create an SPM App of type "Docker" to obtain the SPM App Token. SPM App will hold your Kubernetes performance metrics and event.
3. Create a Logsene App to obtain the Logsene App Token. Logsene App will hold your Kubernetes logs.
4. Edit values of LOGSENE_TOKEN and SPM_TOKEN in the DaemonSet definition as shown below.

- Grab the latest sematext-agent-daemonset.yml (raw plain-text) template (also shown below).
- Store it somewhere on the disk.
- Replace the SPM_TOKEN and LOGSENE_TOKEN placeholders with your SPM and Logsene App tokens.

```
Create DaemonSet Object
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
   name: sematext-agent
spec:
   template:
      metadata:
         labels:
            app: sematext-agent
      spec:
         selector: {}
         dnsPolicy: "ClusterFirst"
         restartPolicy: "Always"
         containers:
         - name: sematext-agent
            image: sematext/sematext-agent-docker:latest
            imagePullPolicy: "Always"
            env:
            - name: SPM_TOKEN
               value: "REPLACE THIS WITH YOUR SPM TOKEN"
            - name: LOGSENE_TOKEN
               value: "REPLACE THIS WITH YOUR LOGSENE TOKEN"
            - name: KUBERNETES
               value: "1"
            volumeMounts:
               - mountPath: /var/run/docker.sock
                  name: docker-sock
               - mountPath: /etc/localtime
                  name: localtime
            volumes:
               - name: docker-sock
                  hostPath:
                     path: /var/run/docker.sock
               - name: localtime
                  hostPath:
                     path: /etc/localtime
```

Running the Sematext Agent Docker with kubectl

```
$ kubectl create -f sematext-agent-daemonset.yml
```

```
daemonset "sematext-agent-daemonset" created
```

Kubernetes Log

Kubernetes container logs are not much different from Docker container logs. However, Kubernetes users need to view logs for the deployed pods. Hence, it is very useful to have Kubernetes-specific information available for log search, such as —

1. Kubernetes namespace
2. Kubernetes pod name
3. Kubernetes container name
4. Docker image name
5. Kubernetes UID

Using ELK Stack and LogSpout

ELK stack includes Elasticsearch, Logstash, and Kibana. To collect and forward the logs to the logging platform, we will use LogSpout (though there are other options such as FluentD).

The following code shows how to set up an ELK cluster on Kubernetes and create a service for ElasticSearch —

```
apiVersion: v1
kind: Service
metadata:
   name: elasticsearch
   namespace: elk
   labels:
      component: elasticsearch
spec:
   type: LoadBalancer
   selector:
      component: elasticsearch
   ports:
   - name: http
      port: 9200
      protocol: TCP
   - name: transport
      port: 9300
      protocol: TCP
Creating Replication Controller
```

```
apiVersion: v1
kind: ReplicationController
metadata:
   name: es
   namespace: elk
   labels:
      component: elasticsearch
spec:
   replicas: 1
   template:
      metadata:
         labels:
            component: elasticsearch
spec:
serviceAccount: elasticsearch
containers:
   - name: es
      securityContext:
      capabilities:
      add:
      - IPC_LOCK
   image: quay.io/pires/docker-elasticsearch-kubernetes:1.7.1-4
   env:
   - name: KUBERNETES_CA_CERTIFICATE_FILE
   value: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
   - name: NAMESPACE
   valueFrom:
      fieldRef:
         fieldPath: metadata.namespace
   - name: "CLUSTER_NAME"
      value: "myesdb"
   - name: "DISCOVERY_SERVICE"
      value: "elasticsearch"
   - name: NODE_MASTER
      value: "true"
   - name: NODE_DATA
      value: "true"
   - name: HTTP_ENABLE
      value: "true"
ports:
- containerPort: 9200
   name: http
   protocol: TCP
- containerPort: 9300
volumeMounts:
- mountPath: /data
   name: storage
volumes:
   - name: storage
      emptyDir: {}
```

Kibana URL

For Kibana, we provide the Elasticsearch URL as an environment variable.

```
- name: KIBANA_ES_URL
value: “http://elasticsearch.elk.svc.cluster.local:9200"
- name: KUBERNETES_TRUST_CERT
value: “true”
```

Kibana UI will be reachable at container port 5601 and the corresponding host/Node Port combination. When you begin, there won’t be any data in Kibana (which is expected as you have not pushed any data).

Startup Tech AWS DevOps Docker