Table of Contents

Setting Up Prometheus Monitoring – K8S

1. Create Prometheus cluster using kops addon:
    a. https://github.com/kubernetes/kops/tree/master/addons/prometheus-operator
2. This will create Prometheus deployments using CoreOS Prometheus Operator project
3. By default, the setup is created under 'monitoring' namespace
4. Once the setup is created, use LoadBalancer Configuration to create an externally accessible service, for:
    a. prometheus-k8s
    b. alertmanager-main
    c. grafana

Monitoring applications

1. Prometheus Operator works on custom resource definitions (CRDs) to specify which k8s services need to be monitored.
2. It is as per these CRDs, that the scrape configurations under Prometheus deployments are created.
3. In order to make Prometheus to scrape your applications, we must create our additional config, to define the rules for scraping
4. Below is an example of additional config created for scraping one spring based application (prometheus-additional.yaml)

```
- job_name: default/monitor-tenant-resource-provision
  scrape_interval: 30s
  scrape_timeout: 10s
  metrics_path: "/prometheus"
  scheme: http
  kubernetes_sd_configs:
    - role: endpoints
      namespaces:
        names:
          - default
  relabel_configs:
    - source_labels: [__meta_kubernetes_endpoint_port_name]
      separator: ;
      regex: web
      replacement: $1
      action: keep
    - source_labels: [__meta_kubernetes_service_name]
      separator: ;
      regex: tenant-resource-provision-service
      replacement: $1
      action: keep
```

5. The above file can have as many configurations as required
6. Please note that 'relabel_configs' are required in an environment, where you have dynamic targets, based on their metadata keys
7. Use "Service Discovery" under Prometheus to understand the correct values for this

8. After creating the required addition configuration, we must store this file as “Secret” resource under k8s

Storing Scrape Config File as Secret

1. Generate a secret config file using the following command:
    a. kubectl create secret generic additional-scrape-configs --from-file=prometheus-additional.yaml --dry-run -oyaml > additional-scrape-configs.yaml
2. This creates a new file “additional-scrape-configs.yaml”, which should something like:

```
apiVersion: v1
data:
  prometheus-additional.yaml:
        LSBqb2JfbmFtZTogZGVmYXVsdC9tb25pdG9yLXRlbmFudC1yZXNvdXJjZS1wcm92
        aXNpb24KICBzY3JhcGVfaW50ZXJ2YWw6IDMwcwogIHNjcmFwZV90aW1lb3V0OiA
        xMHMKICBtZXRyaWNzX3BhdGg6ICIvcHJvbWV0aGV1cyIKICBzY2hlbWU6IGh0dH
        AKICBrdWJlcm5ldGVzX3NkX2NvbmZpZ3M6CiAgICAtIHJvbGU6IGVuZHBvaW50cw
        ogICAgICBuYW1lc3BhY2VzOgogICAgICAgIG5hbWVzOgogICAgICAgLSBkZW
        ZhdWx0CiAgcmVsYWJlbF9jb25maWdzOgogICAgLSBzb3VyY2VfbGFiZWxzOiBbX1
        9tZXRhX2t1YmVybmV0ZXNfZW5kcG9pbnRfcG9ydF9uYW1lXQogICAgICBzZXBhc
        mF0b3I6IDsKICAgICAgcmVnZXg6IHdlYgogICAgICByZXBsYWNlbWVudDogJDEKIC
        AgICAgYWN0aW9uOiBrZWVwCiAgICAtIHNvdXJjZV9sYWJlbHM6IFtfX21ldGFfa3ViZ
        XJuZXRlc19zZXJ2aWNlX25hbWVdCiAgICAgIHNlcGFyYXRvcjogOwogICAgICByZW
        dleDogdGVuYW50LXJlc291cmNlLXByb3Zpc2lvbi1zZXJ2aWNlCiAgICAgIHJlcGxxhY2
        VtZW50OiAkMQogICAgICBhY3Rpb246IGtlZXAKCg==
kind: Secret
metadata:
  creationTimestamp: null
  name: additional-scrape-configs
  namespace: monitoring
```

3. Create this resource under k8s using:

```
kubectl apply -f additional-scrape-configs.yaml
```

Specify Usage of Additional Configuration under ‘Prometheus’ resource

1. Under Prometheus resource yaml (v0.26.yaml in our case), search for resource of “Kind: Prometheus”
2. Add the following configuration to use the secret created above:

```
serviceMonitorNamespaceSelector: {}
  serviceMonitorSelector: {}
  additionalScrapeConfigs:
```

```yaml
  name: additional-scrape-configs
  key: prometheus-additional.yaml
```

3. Update the resource group for Prometheus using:

kubectl apply -f v0.26.0.yaml

4. This should have your monitoring of application setup

If you want to edit your prometheus-additional.yaml for entering more configuration then edit it and after that execute command:-

< kubectl create secret generic additional-scrape-configs
--from-file=prometheus-additional.yaml --dry-run -oyaml > additional-scrape-configs.yaml>

after that kubectl apply -f v0.26.0.yaml

Now check in your prometheus that weather the configuration is updated or not, If it is not updated then copy the prometheus-additional.yaml data from addition-scrape-config that somewhat looks like this:-

LSBqb2JfbmFtZTogZGVmYXVsdC9tb25pdG9yLXRlbmFudC1yZXNvdXJjZS1wcm92aXNpb2
4KICBzY3JhcGVfaW50ZXJ2YWw6IDMwcwogIHNjcmFwZV90aW1lb3V0OiAxMHMKICBtZX
RyaWNzX3BhdGg6ICIvcHJvbWV0aGV1cyIKICBzY2hlbWU6IGh0dHAKICBrdWJlcm5ldGVz
X3NkX2NvbmZpZ3M6CiAgICAtIHJvbGU6IGVuZHBvaW50cwogICAgICBuYW1lc3BhY2VzO
gogICAgICAgIG5hbWVzOgogICAgICAgICAgLSBkZWZhdWx0CiAgcmVsYWJlbF9jb25maW
dzOgogICAgLSBzb3VyY2VfbGFiZWxzOiBbX19tZXRhX2t1YmVybmV0ZXNfZW5kcG9pbnRf
cG9ydF9uYW1lXQogICAgICBzZXBhcmF0b3I6IDsKICAgICAgcmVnZXg6IHdlYgogICAgICBy
ZXBsYWNlbWVudDogJDEKICAgICAgYWN0aW9uOiBrZWVwCiAgICAtIHNvdXJjZV9sYWJlb
HM6IFtfX21ldGFfa3ViZXJuZXRlc19zZXJ2aWNlX25hbWVdCiAgICAgIHNlcGFyYXRvcjogOw
ogICAgICByZWdleDogdGVuYW50LXJlc291cmNlLXByb3Zpc2lvbi1zZXJ2aWNlCiAgICAgIHJl
cGxhY2VtZW50OiAkMQogICAgICBhY3Rpb246IGtlZXAKCi0gam9iX25hbWU6IGRlZmF1bH
QvbW9uaXRvci1zZWN1cml0eS1mcmFtZXdvcmsKICBzY3JhcGVfaW50ZXJ2YWw6IDMwcw
ogIHNjcmFwZV90aW1lb3V0OiAxMHMKICBtZXRyaWNzX3BhdGg6ICIvcHJvbWV0aGV1cyI
KICBzY2hlbWU6IGh0dHAKICBrdWJlcm5ldGVzX3NkX2NvbmZpZ3M6CiAgICAtIHJvbGU6IG
VuZHBvaW50cwogICAgICBuYW1lc3BhY2VzOgogICAgICAgIG5hbWVzOgogICAgICAgICAg
LSBkZWZhdWx0CiAgcmVsYWJlbF9jb25maWdzOgogICAgLSBzb3VyY2VfbGFiZWxzOiBbX
19tZXRhX2t1YmVybmV0ZXNfZW5kcG9pbnRfcG9ydF9uYW1lXQogICAgICBzZXBhcmF0b3I
6IDsKICAgICAgcmVnZXg6IHdlYgogICAgICByZXBsYWNlbWVudDogJDEKICAgICAgYWN0
aW9uOiBrZWVwCiAgICAtIHNvdXJjZV9sYWJlbHM6IFtfX21ldGFfa3ViZXJuZXRlc19zZXJ2a
WNlX25hbWVdCiAgICAgIHNlcGFyYXRvcjogOwogICAgICByZWdleDogc2VjdXJpdHktZnJhb
WV3b3JrLXNlcnZpY2UKICAgICAgcmVwbGFjZW1lbnQ6ICQxCiAgICAgIGFjdGlvbjoga2VlcA
oKLSBqb2JfbmFtZTogZGVmYXVsdC9tb25pdG9yLWV4cGVyaWVuY2UtbWFuYWdlcgogIH
NjcmFwZV9pbnRlcnZhbDogMzBzCiAgc2NyYXBlX3RpbWVvdXQ6IDEwcwogIG1ldHJpY3Nf
cGF0aDogIi9wcm9tZXRoZXVzIgogIHNjaGVtZTogaHR0cAogIGt1YmVybmV0ZXNfc2RfY29u
ZmlnczoKICAgIC0gcm9sZTogZW5kcG9pbnRzCiAgICAgIG5hbWVzcGFjZXM6CiAgICAgICA
gbmFtZXM6CiAgICAgICAgICAtIGRlZmF1bHQKICByZWxhYmVsX2NvbmZpZ3M6CiAgICAtI
HNvdXJjZV9sYWJlbHM6IFtfX21ldGFfa3ViZXJuZXRlc19lbmRwb2ludF9wb3J0X25hbWVdCi
AgICAgIHNlcGFyYXRvcjogOwogICAgICByZWdleDogd2ViCiAgICAgIHJlcGxhY2VtZW50OiA
kMQogICAgICBhY3Rpb246IGtlZXAKICAgIC0gc291cmNlX2xhYmVsczogW19fbWV0YV9rd
WJlcm5ldGVzX3NlcnZpY2VfbmFtZV0KICAgICAgc2VwYXJhdG9yOiA7CiAgICAgIHJlZ2V4Oi

BleHBlcmllbmNlLW1hbmFnZXItc2VydmljZQogICAgICByZXBsYWNlbWVudDogJDEKICAgICAgICAgYWN0aW9uOiBrZWVwCg==

now go to your kubernetes dashboard and select your name space for ex:- default or monitoring and edit your additional-scrape-config and copy the data below prometheus-additional.yaml and update it

Now kubectl apply -f v0.26.0.yaml

update your prometheus dashboard 2 0r 3 times…..you wil get your configurations updated.