Scaling Search Clusters with Apache Solr & Kubernetes

Amrit Sarkar

Consultant, Seamadic Pvt Ltd

Engineering Manager, Search, Tata Unistore Ltd

TATA CLiQ

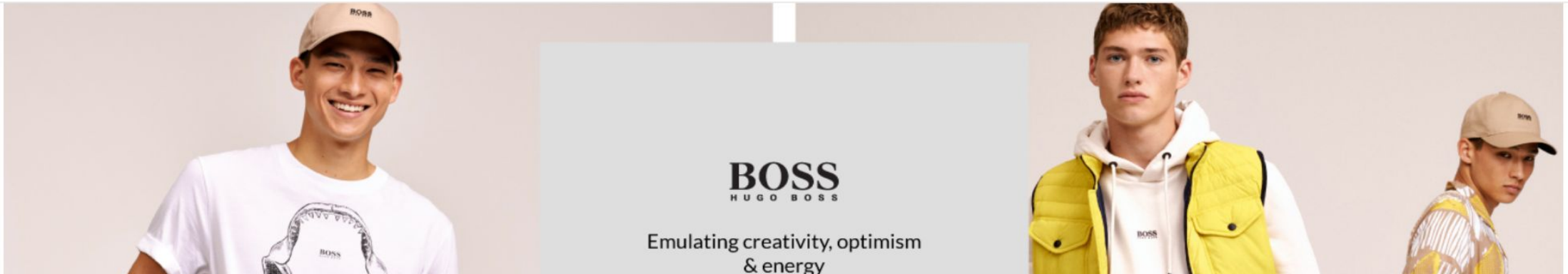
Tata CLiQ Luxury | CLiQ Cash | Gift Card | CLiQ Care | Track Orders | Hello

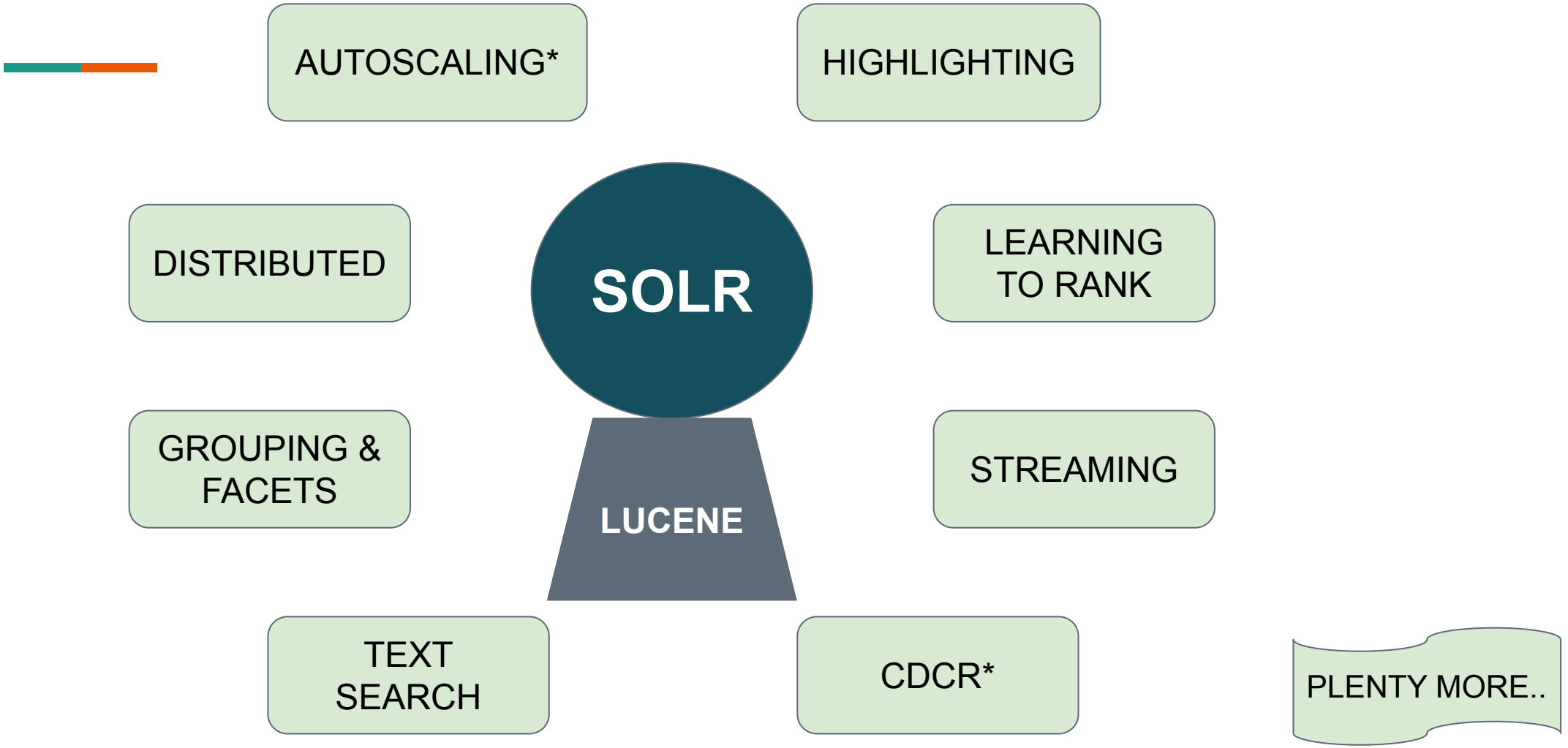
Categories | Brands | Search

LUXURY | INDILUXE

TATA CLiQ LUXURY
WHERE QUALITY IS NURTURED

New In | Brands | Men | Women | Kids | Beauty & Grooming | Home | Eyewear | Featured Stores

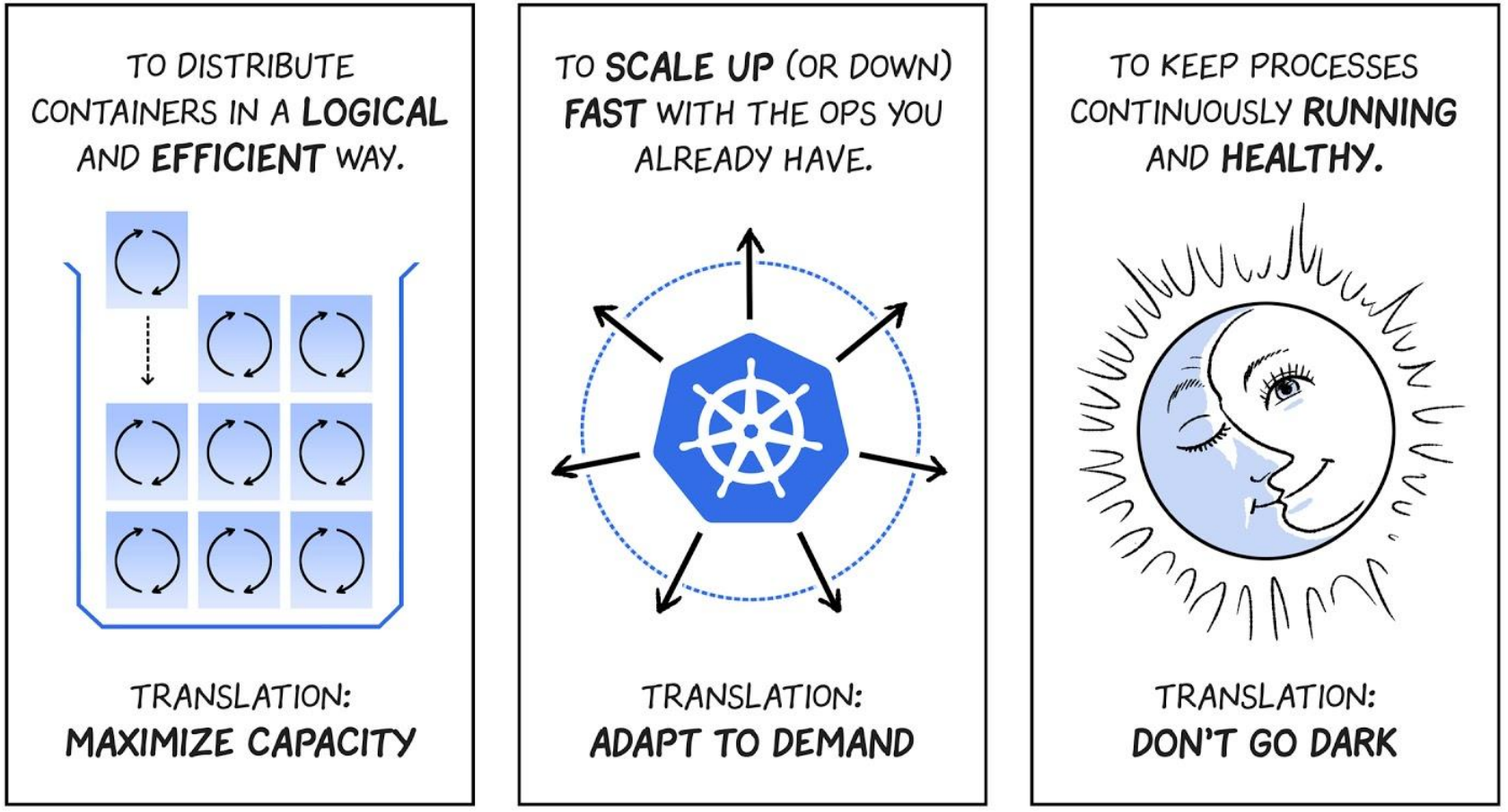



AUTOSCALING*
HIGHLIGHTING
DISTRIBUTED
SOLR
LEARNING TO RANK
GROUPING & FACETS
LUCENE
STREAMING
TEXT SEARCH
CDCR*
PLENTY MORE..

referred from: https://cloud.google.com/kubernetes-engine/kubernetes-comic/

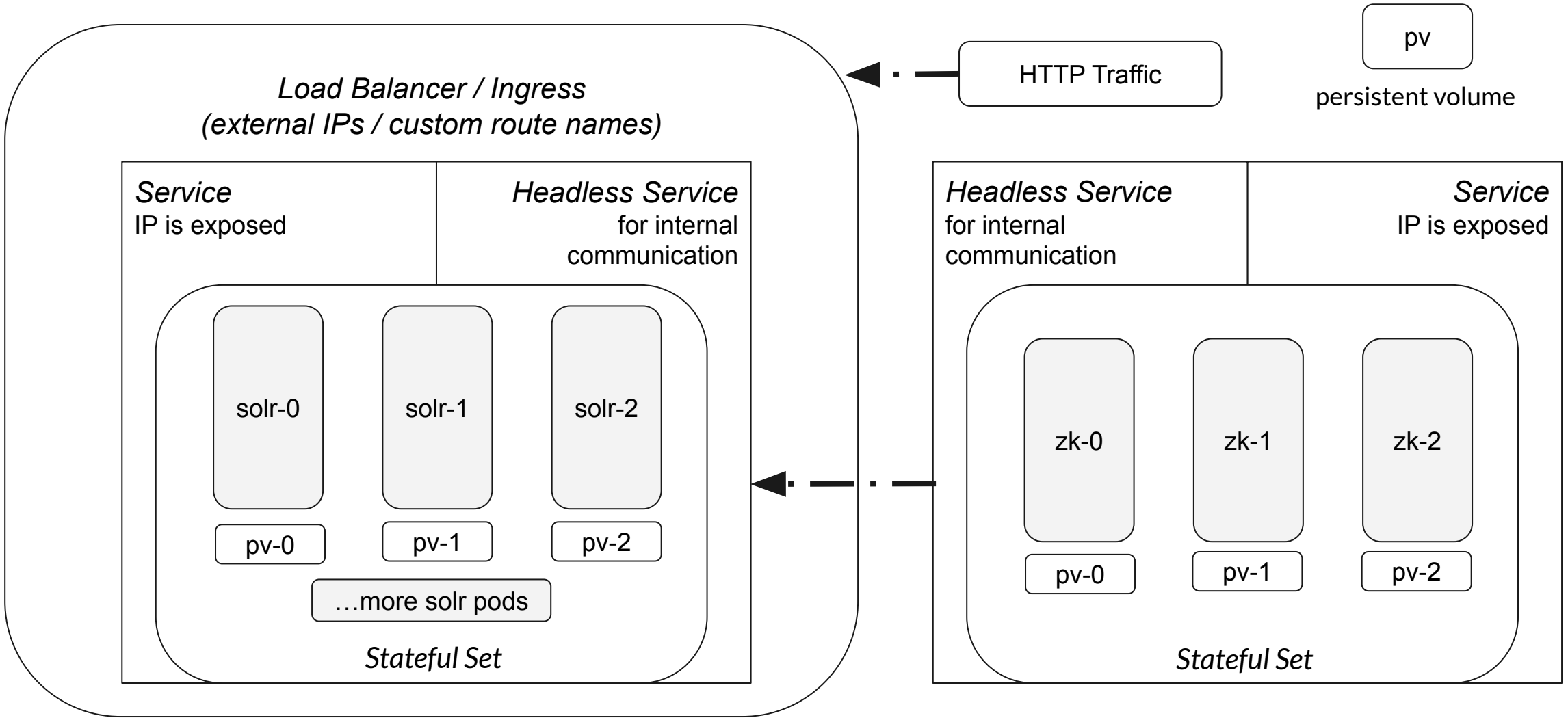
Kubernetes (K8s)

System for running and coordinating containerized applications across a cluster of machines

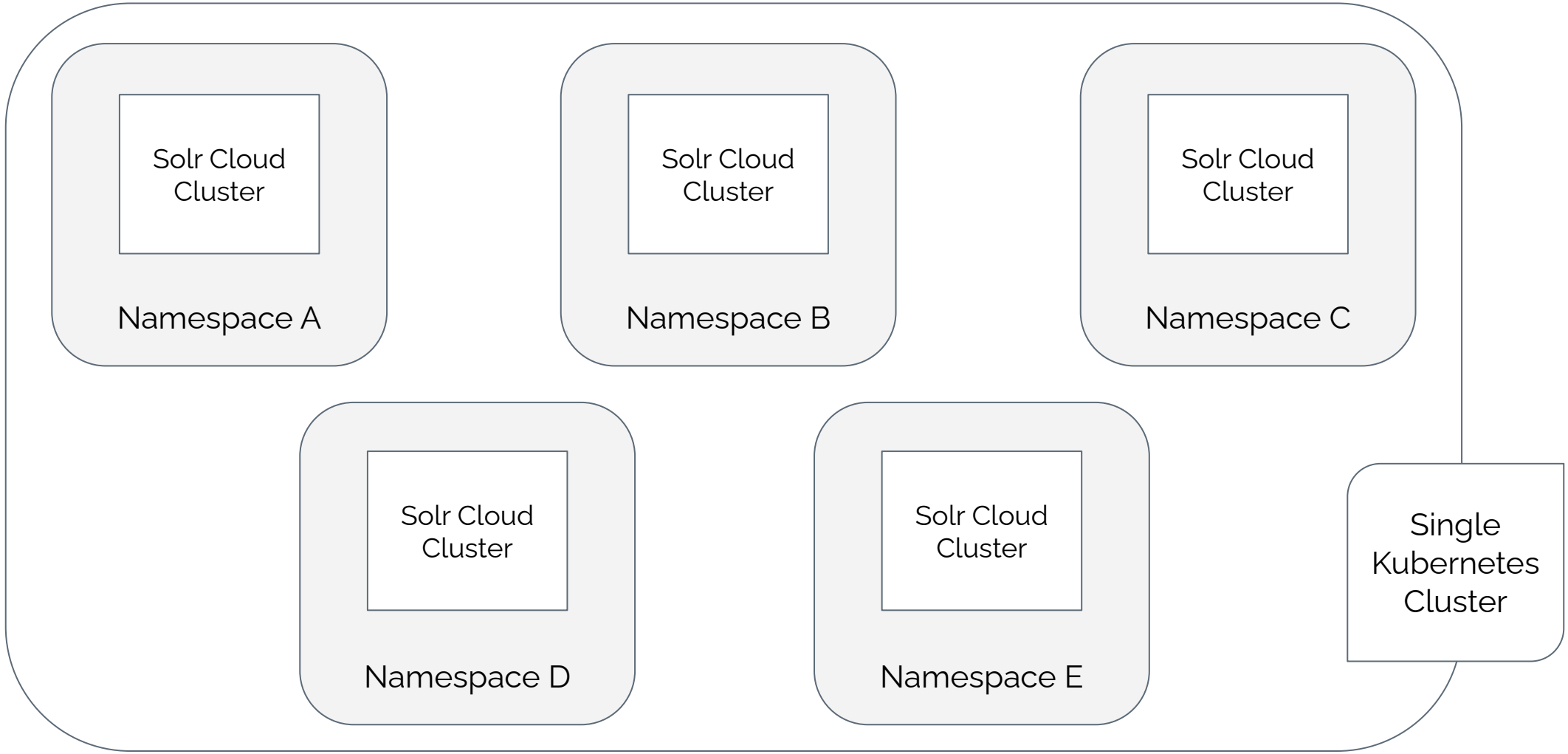


referred from: https://cloud.google.com/kubernetes-engine/kubernetes-comic/

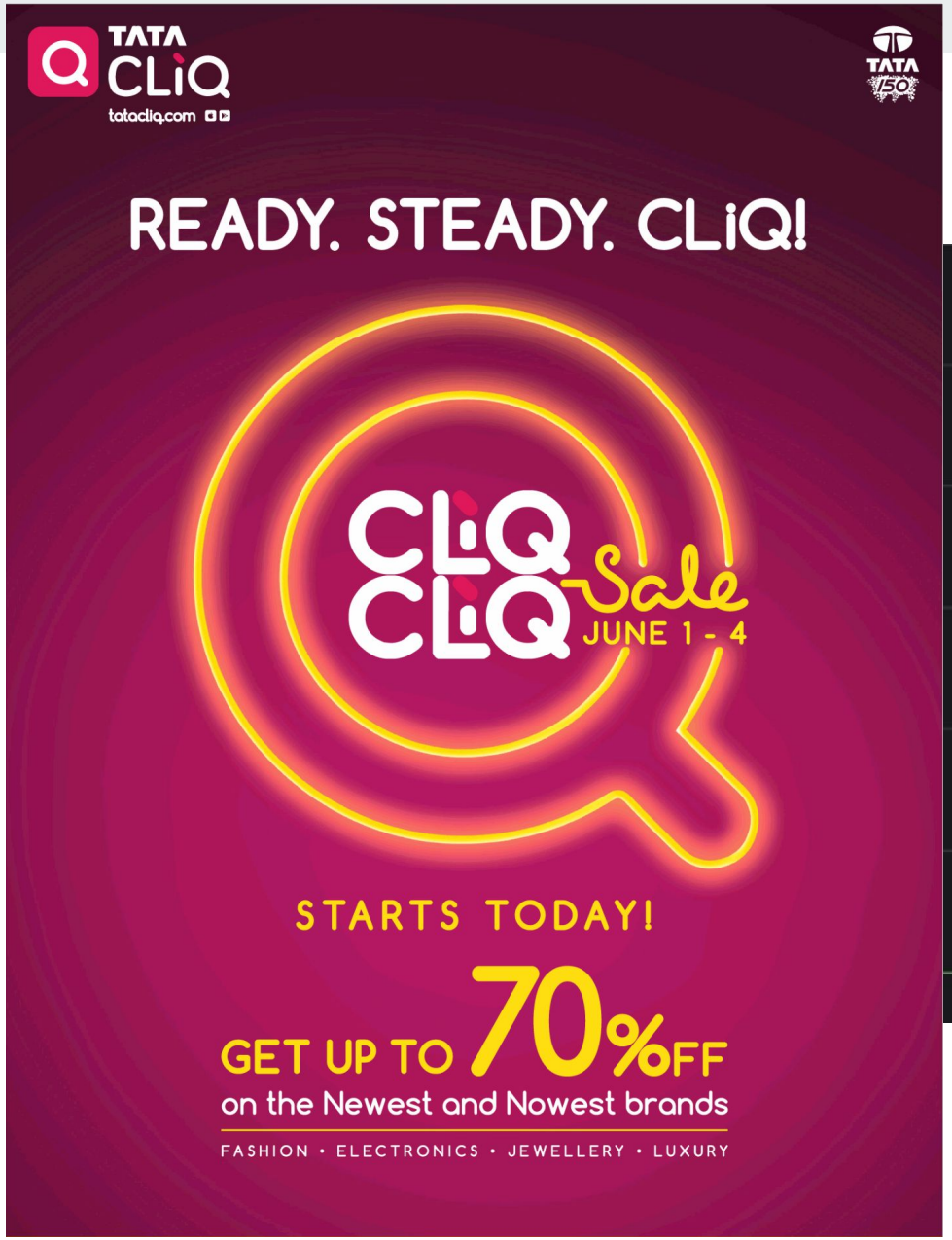
Run Solr Cloud on Kubernetes

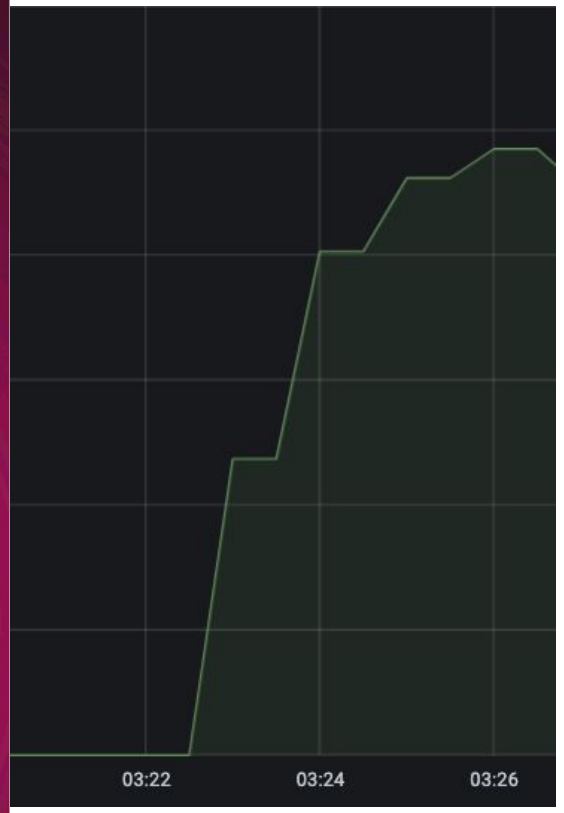


Solr Cloud Clusters within Kubernetes

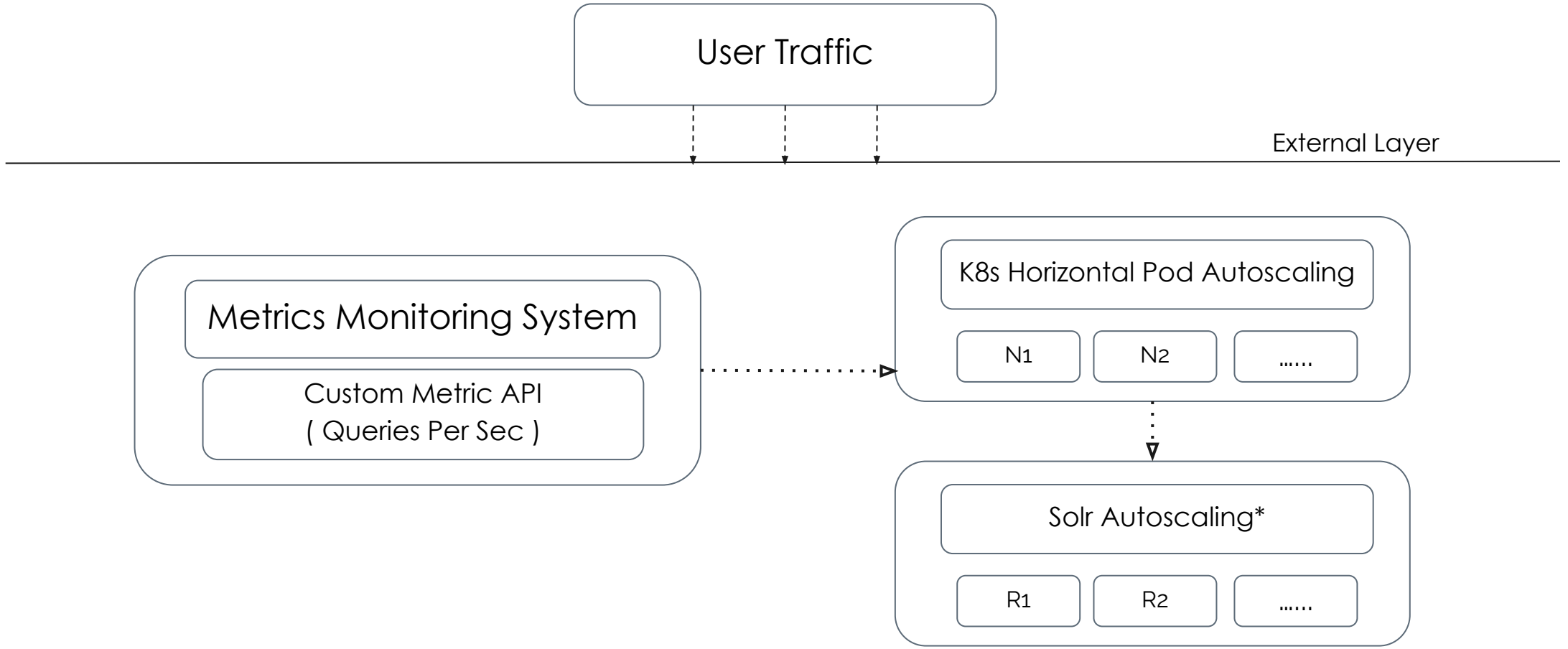



OUR AWS MONTH BILL
IS TOO DAMN HIGH


TATA CLiQ
tatacliq.com
TATA 150
READY. STEADY. CLiQ!
CLiQ CLiQ Sale
JUNE 1 - 4
STARTS TODAY!
GET UP TO 70%FF
on the Newest and Nowest brands
FASHION • ELECTRONICS • JEWELLERY • LUXURY


03:22
03:24
03:26

Scaling Solr Clusters w/o Manual Intervention



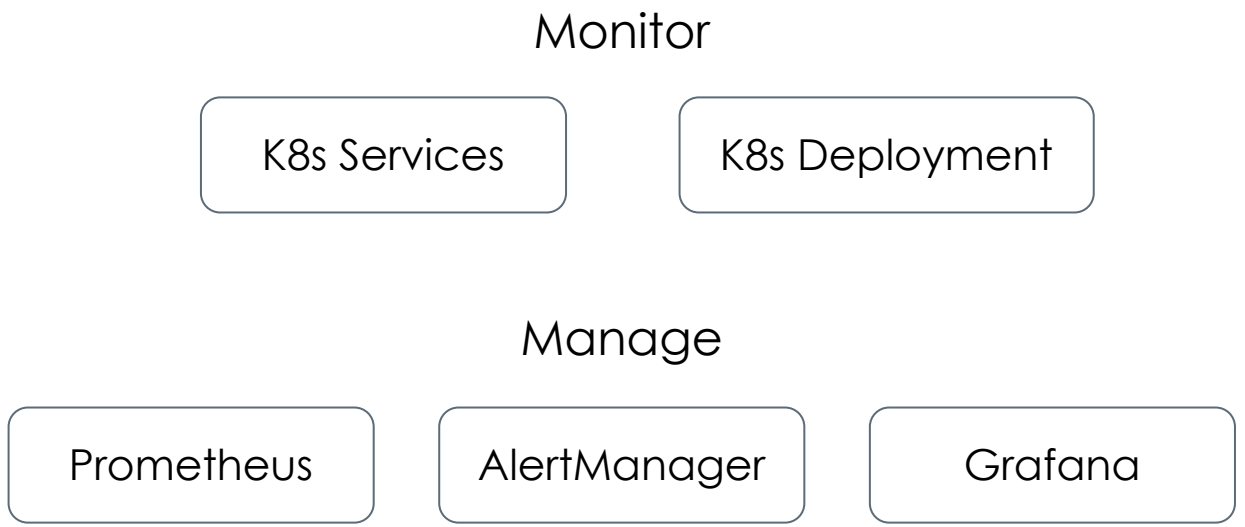
*deprecated in Apache Solr version 9.0

Monitoring with Prometheus & Grafana

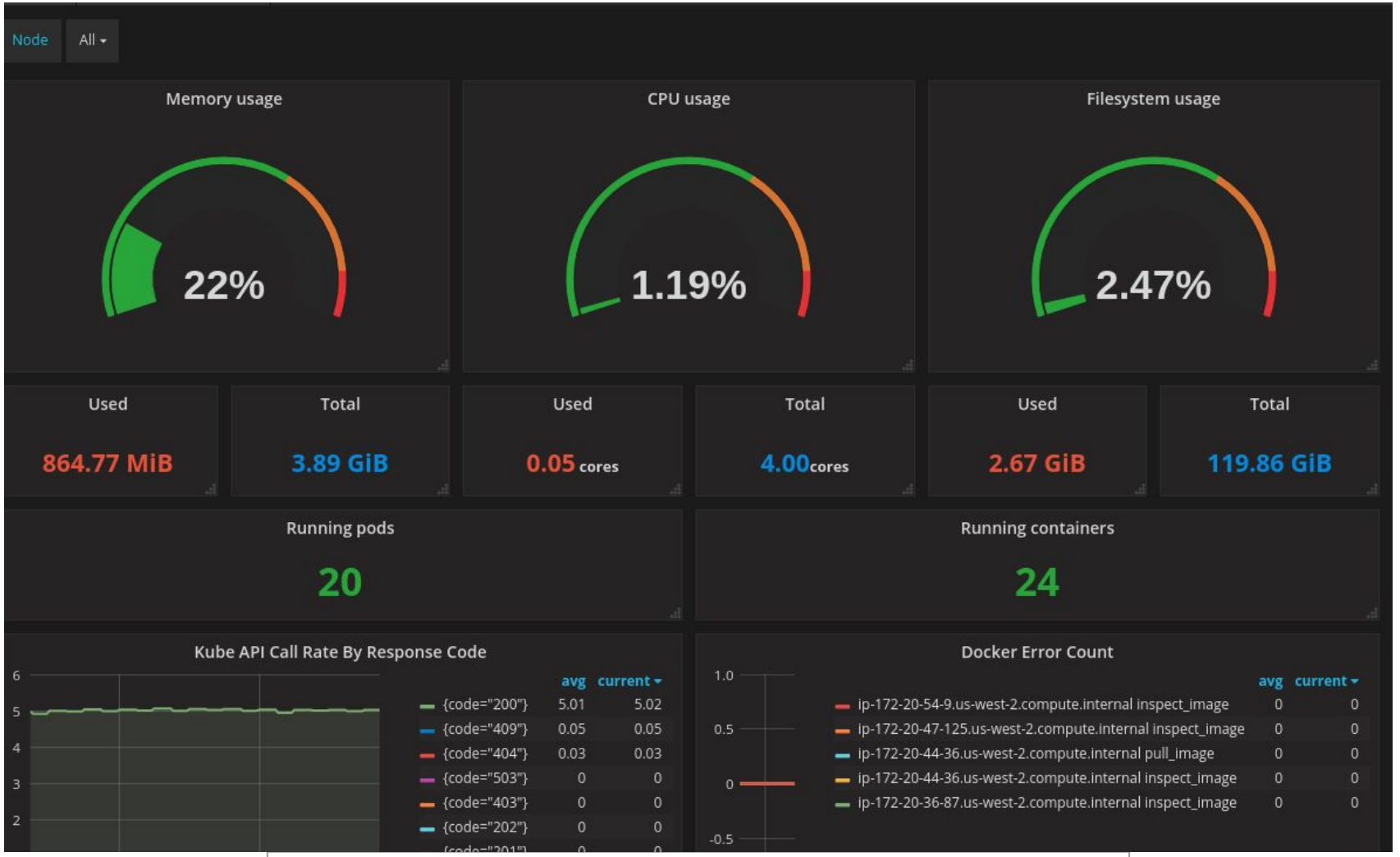
Prometheus helm chart: https://github.com/helm/charts/tree/master/stable/prometheus

- **Prometheus Operator:**
  Uses Custom Resource Definition (CRD), named **ServiceMonitor**,
  to abstract configuration to target.

```
apiVersion:
monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: solr
spec:
  selector:
    matchLabels:
      app: solr
```

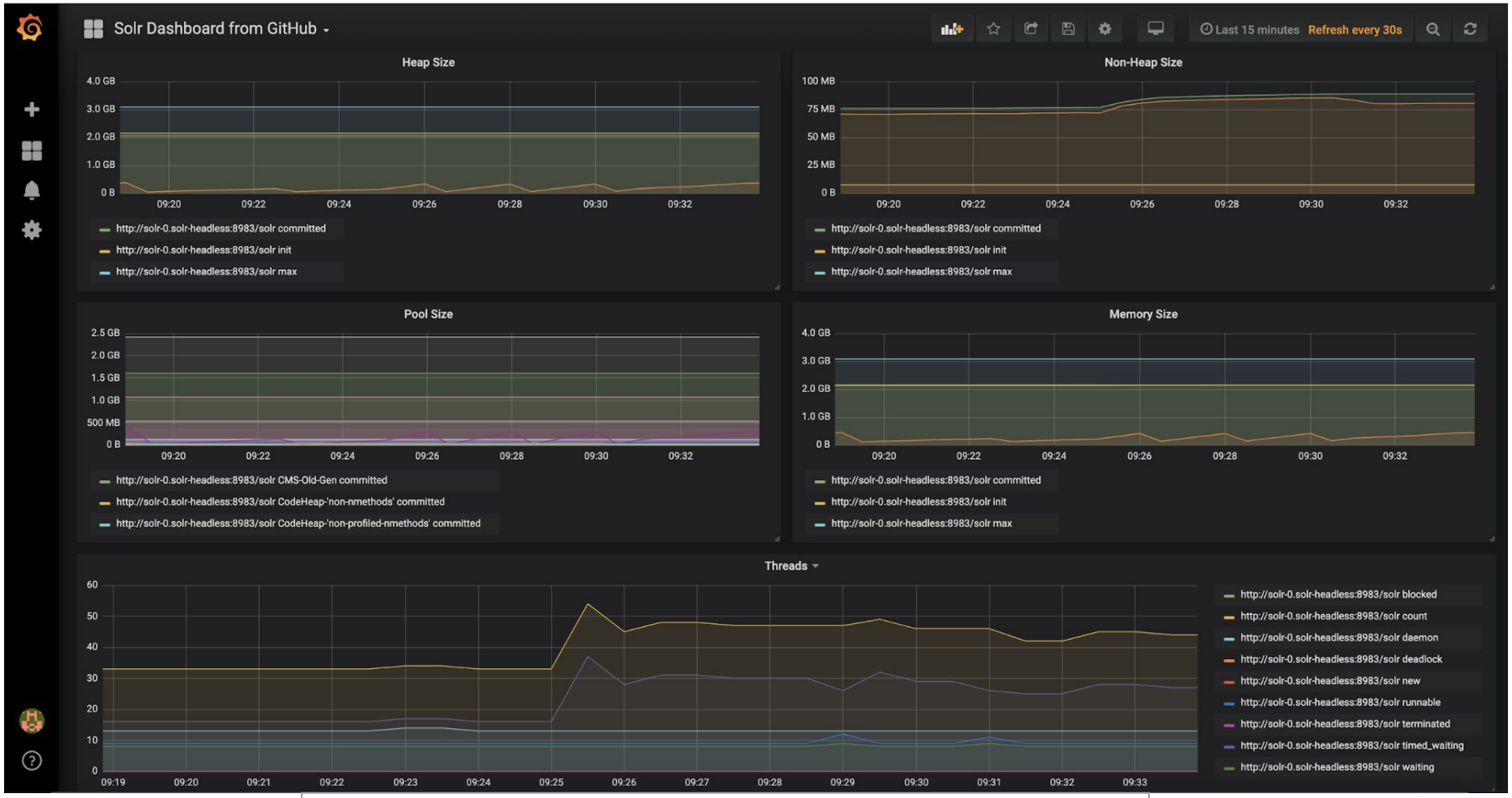


Monitoring with Prometheus & Grafana



Kubernetes State Metrics

Monitoring with Prometheus & Grafana



Solr Metrics

Custom Metrics with Prometheus Adapter

https://github.com/helm/charts/tree/master/stable/prometheus
https://github.com/kubernetes-sigs/custom-metrics-apiserver

- **Prometheus Adapter and Custom Metrics API Server:**
  Pull custom metrics from Prometheus installation and register with K8s.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: adapter-config
  namespace: monitoring
data:
  config.yaml: |
    rules:
    - seriesQuery: solr_requests
      resources:
        overrides:
          kubernetes_namespace:
            resource: namespace
          kubernetes_pod_name:
            resource: pod
      name:
        matches: "^(.*)_requests"
        as: "${1}_requests"
      metricsQuery: (<<.Series>>{<<.LabelMatchers>>})
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
.....
  name: custom-metrics-apiserver
  namespace: monitoring
spec:
  replicas: 1
...
  template:
    metadata:
...
      - name: custom-metrics-apiserver
        image: quay.io/coreos/k8s-prometheus-adapter-amd64:v0.4.1
        args:
        - /adapter
        - --prometheus-url=http://prometheus-k8s.monitoring.svc:9090/
        - --config=/etc/adapter/config.yaml
        ........
........
........
```

Custom Metrics with Prometheus Adapter

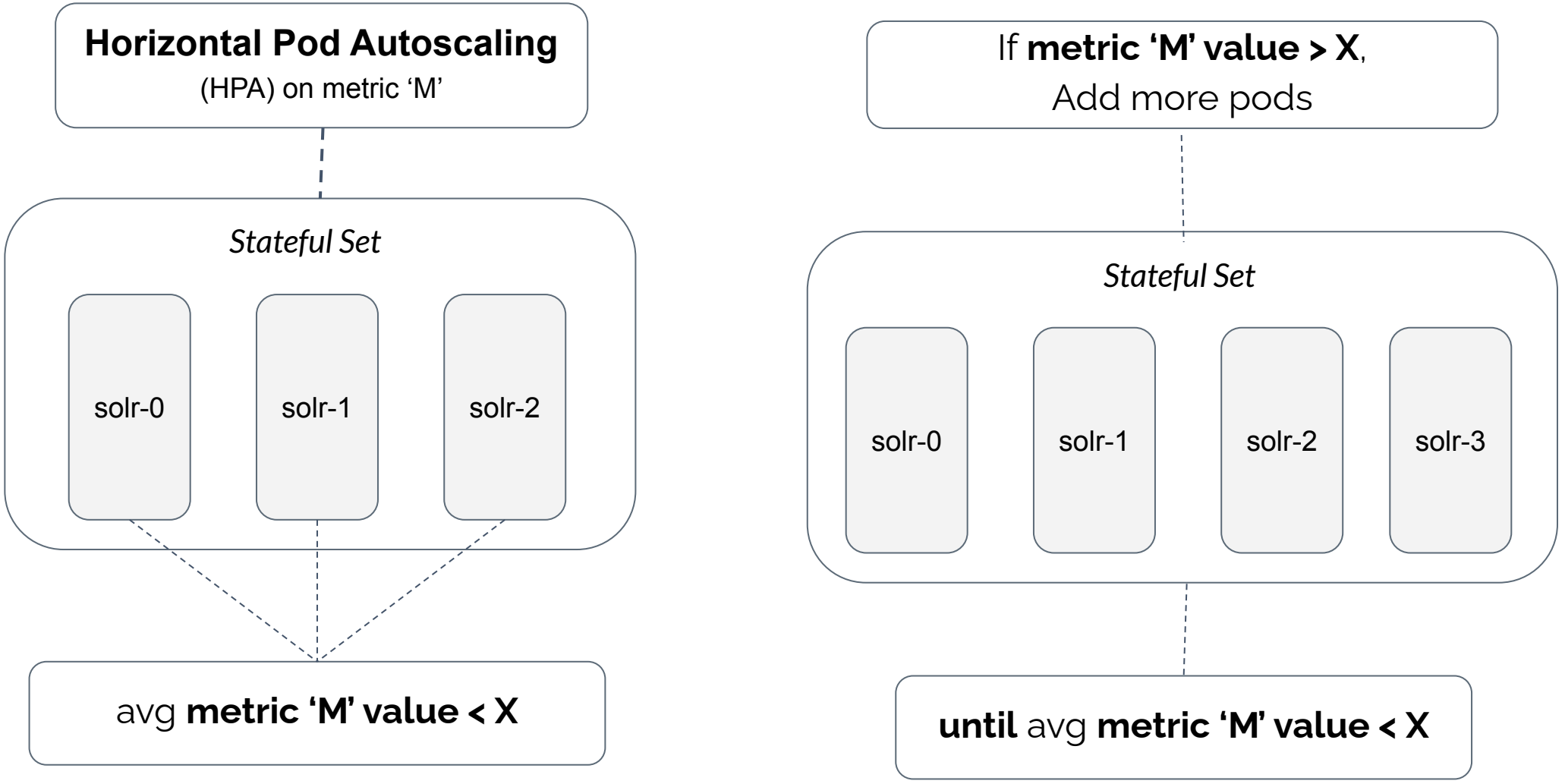
- **Prometheus Adapter and Custom Metrics API Server:**
  Pull custom metrics from Prometheus installation and register with K8s.

```
apiVersion: v1
kind: Service
metadata:
  name: custom-metrics-apiserver
  namespace: monitoring
spec:
  ports:
  - port: 443
    targetPort: 6443
  selector:
    app: custom-metrics-apiserver
---
apiVersion: apiregistration.k8s.io/v1beta1
kind: APIService
metadata:
  name: v1beta1.custom.metrics.k8s.io
spec:
  service:
    name: custom-metrics-apiserver
    namespace: monitoring
```

```
kubectl get --raw
"/apis/custom.metrics.k8s.io/v1beta1/namespaces/default/pods/*/
solr_metrics_core_distrib_select_one_minute_rate" | jq .

{
  "kind": "MetricValueList",
  "apiVersion": "custom.metrics.k8s.io/v1beta1",
  "metadata": {
    "selfLink":
"/apis/custom.metrics.k8s.io/v1beta1/namespaces/default/pods/%2A/so
lr_metrics_core_distrib_select_one_minute_rate"
  },
  "items": [
    {
      "describedObject": {
        "kind": "Pod",
        "namespace": "default",
        "name": "solr-test-0",
        "apiVersion": "/v1"
      },
      "metricName": "solr_metrics_core_distrib_select_one_minute_rate",
      "timestamp": "2021-06-13T18:55:21Z",
      "value": "0"
    },
  ]
}
```

K8s Horizontal Pod Autoscaling



K8s Horizontal Pod Autoscaling

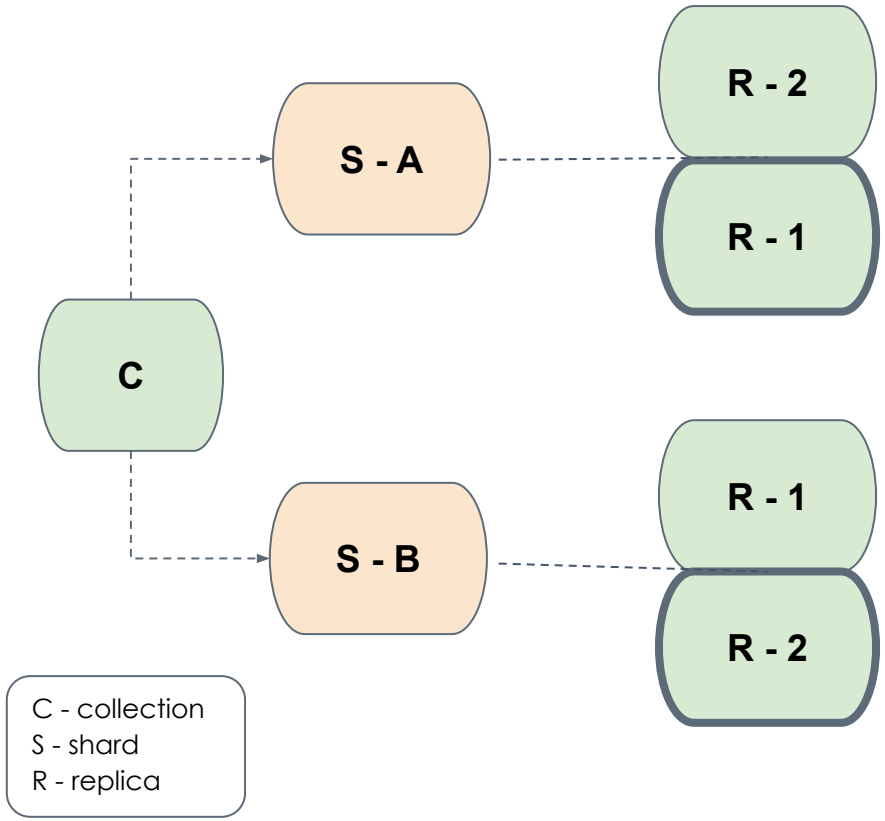
```
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
  name: solr-statefulset
  namespace: default
spec:
  scaleTargetRef:
    apiVersion: extensions/v1beta1
    kind: Statefulset
    name: solr-statefulset
  minReplicas: 3
  maxReplicas: 10
  metrics:
  - type: Pods
    pods:
      metricName:
solr_metrics_core_distrib_select_one_minute_rate
      targetAverageValue: 100
```

```
root$ kubectl describe hpa

Name:                                                  solr-hpa
Namespace:                                             default
Labels:                                                <none>
Annotations:                                           <none>
CreationTimestamp:                                     Sun, 13 Jun 2021 20:04:35 +0530
Reference:                                             StatefulSet/solr-statefulset
Metrics:                                               ( current / target )
  "solr_metrics_core_distrib_select_one_minute_rate" on pods:  50 / 100
Min replicas:                                          3
Max replicas:                                          10
StatefulSet pods:                                      3 current / 10 desired
Conditions:
  Type          Status  Reason          Message
  ----          ------  ------          -------
  AbleToScale     True    ReadyForNewScale  recommended size matches current size
  ScalingActive   True    ValidMetricFound  the HPA was able to successfully calculate a
replica count from pods metric solr_metrics_core_distrib_select_one_minute_rate
  ScalingLimited  True    TooFewReplicas    the desired replica count is less than the
minimum replica count
Events:         <none>
```

Solr Autoscaling*

- **Autoscaling in Solr** provide balanced and stable cluster for various cluster change events.
- Achieved by satisfying a **set of rules** and **sorting preferences** to select the target of cluster management operations.



Set solr-autoscaling policy and trigger such that -

Add one replica to each shard of collection

- if new Node is added

*deprecated in Apache Solr version 9.0

Solr Autoscaling*

Set solr-autoscaling policy and trigger such that - Add one replica to each shard of collection

- if new Node is added

```
{
   "cluster-policy":[
   {
    "node":"#ANY",
    "replica":"<2",
    "shard":"#EACH"}
   ]
}
```

Cluster Policy

```
"node_added_trigger":{
 "event":"nodeAdded",
 "waitFor":60,
 "preferredOperation":"ADDREPLICA",
 "actions":[
  {
   "name":"compute_plan",
   "class":"solr.ComputePlanAction"},
  {
   "name":"execute_plan",
   "class":"solr.ExecutePlanAction"}],
}
```

Trigger Policy

*deprecated in Apache Solr version 9.0

Production Readiness

- What happens when nodes are removed due to scaling? Is there a potential of losing data? What if all replicas of collection gets hosted on scaled nodes?
  - Cluster Policy to host at least one replica of each shard in safe nodes.
- Do we need to scale on each collection on the entire cluster? How to configure scaling on specific collection or set of collections with common attribute?
  - Prometheus adapter configuration based on business logic.
- How does the Node Added Trigger behaves when a node gets restarted? Replicas will be added to each shard of each collection where it was previously not.

Roadmap

- Explore Managing Metrics via Solr Operator
  (https://apache.github.io/solr-operator/docs/solr-prometheus-exporter/#prometheus-stack)

- Retire Solr Autoscaling* & look for better stable alternative
  - Integrate own Control Plane
  - Extend capabilities of Solr Operator
    (https://solr.apache.org/operator/)

References

- **Presentation Github Link** - https://github.com/sarkaramrit2/scaling-search-clusters-solr-k8s
- **Docker** - https://www.docker.com/
- **Kubernetes** - https://kubernetes.io/
- **Run Solr on Kubernetes** - https://lucidworks.com/post/running-solr-on-kubernetes-part-1/
- **Monitoring Apache Solr Ecosystem on K8s** - https://www.youtube.com/watch?v=UI9mh4gSIsw
- **Custom Metrics via Prometheus on K8s** - https://towardsdatascience.com/kubernetes-hpa-with-custom-metrics-from-prometheus-9ffc201991e
- **K8s HPA** -https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/
- **Solr Autoscaling** - https://lucene.apache.org/solr/guide/8_5/solrcloud-autoscaling-policy-preferences.html

Thank You!

@sarkaramrit2

>BLN
BZZ/
WRDS