

Contents

Configuration Management.....	2
Which ansible command can display all ansible_ configuration for a host?	2
Logrotate-cronjob, ntpd configuration & nagios monitoring	3
Nagios Server setup:	3
Nagios Configuration Changes :.....	4
Nagios UI	5
Server1 Information.....	5
Server1 NTP service status.....	6
Ansible Structure	7
Docker -Kubernetes	9
Docker-compose file	10
Test and validation.....	10
Which Kubernetes command you will use to identify the reason for a pod restart in the project "internal" under namespace "production"	12
Helm chart.....	13
Kubernetes Cluster Details.....	13
Errors and problems faced during the helm chart deployment	13
Final output and validation	15
Metrics	16
Explain how Prometheus work	16
How do you create custom Prometheus alerts and alerting rules for Kubernetes monitoring? Provide an example alert rule and its configuration.....	16
Example Alert.....	16
What is the Prometheus query you can use in Granfana to properly show usage trend of an application metric that is a counter?	18
Database	19

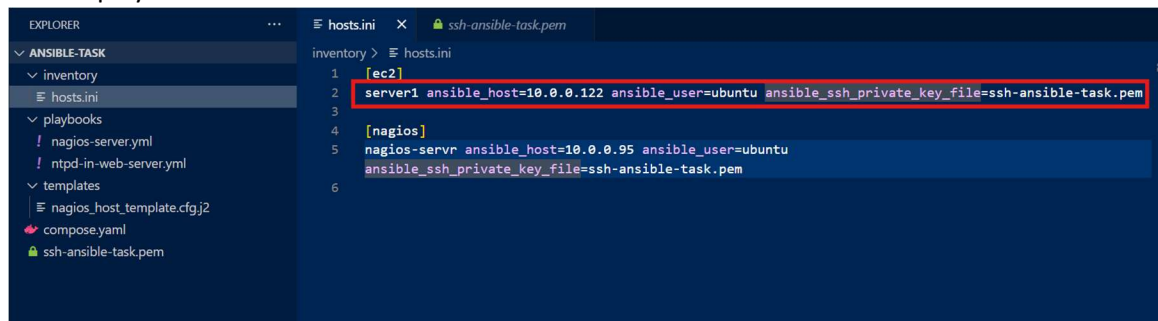
Configuration Management

Which ansible command can display all ansible_ configuration for a host?

'ansible -i <inventory-file> -m <server-name-for-which-we-want-to-check-ansible-configuration>'

The following command will show the ansible-configurations for the indicated server declared in hosts.ini file '**ansible -i ./inventory/hosts.ini -m setup server1** '

This displays all the facts of the host



```
EXPLORER
ANSIBLE-TASK
inventory
hosts.ini
playbooks
nagios-server.yml
ntpd-in-web-server.yml
templates
nagios_host_template.cfg.j2
compose.yaml
ssh-ansible-task.pem

inventory > hosts.ini
1 [ec2]
2 server1 ansible_host=10.0.0.122 ansible_user=ubuntu ansible_ssh_private_key_file=ssh-ansible-task.pem
3
4 [nagios]
5 nagios-servr ansible_host=10.0.0.95 ansible_user=ubuntu
6 ansible_ssh_private_key_file=ssh-ansible-task.pem
```

To have a reduced number of outputs we can only filter with ansible_related facts

ansible -i ./inventory/hosts.ini -a 'filter=ansible_*' -m setup server1

Output-of-the-command: Uploaded in github **server1-ansible-host-configurations.json** file

Logrotate-cronjob, ntpd configuration & nagios monitoring

3 EC2 instances has been created to complete this task.

1. 1 EC2 machine to run ansible playbook.
 - a. Ansible is installed
 - b. If Github-Actions/Gitlab-CI integrated, this EC2 machine will be able to run the playbook through automation
2. 1 nagios server
 - a. Security group rule
 - i. All ports are open for selected ip's (who will monitor the nagios server)
 - b. Public IP assigned
3. 1 web-server
 - a. No public IP assigned
 - b. SSH port is open for Ansible-host EC2
 - c. All ports are open for nagios server

Nagios Server setup:

The machine is **ubuntu 24.04** based.

Based on this set of instructions

Nagios Service status:

```
101 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

*** System restart required ***
Last login: Sun Mar 16 07:30:30 2025 from 10.0.0.74
ubuntu@ip-10-0-0-95:~$ sudo systemctl status nagios
Warning: The unit file, source configuration file or drop-ins of nagios.service changed on disk. Run 'systemctl daemon-reload'.
● nagios.service - Nagios Core 4.5.9
   Loaded: loaded (/usr/lib/systemd/system/nagios.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-12 17:56:50 UTC; 3 days ago
     Docs: https://www.nagios.org/documentation
   Main PID: 73767 (nagios)
    Tasks: 6 (limit: 2338)
  Memory: 10.4M (peak: 12.7M)
     CPU: 1min 55.915s
   CGroup: /system.slice/nagios.service
           └─73767 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
           └─73769 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─73770 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─73771 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─73772 /usr/local/nagios/bin/nagios --worker /usr/local/nagios/var/rw/nagios.qh
           └─73773 /usr/local/nagios/bin/nagios -d /usr/local/nagios/etc/nagios.cfg
```

Nagios Configuration Changes :

1. **Host directory:** Updated `/usr/local/nagios/etc/nagios.cfg` file to allow `/usr/local/nagios/etc/hosts` as valid config directory.
cfg_dir=/usr/local/nagios/etc/hosts
2. All hosts will have a separate configuration file under `/usr/local/nagios/etc/hosts` which will be **generated by ansible**.

```
root@ip-10-0-0-95:/usr/local/nagios/etc# tree
```

```
├── .
├── cgi.cfg
├── hosts
│   └── server1.cfg
├── htpasswd.users
├── nagios.cfg
├── objects
│   ├── commands.cfg
│   ├── contacts.cfg
│   ├── localhost.cfg
│   ├── printer.cfg
│   ├── switch.cfg
│   ├── templates.cfg
│   ├── timeperiods.cfg
│   └── windows.cfg
└── resource.cfg
```

a.

3. **Adding Commands:** Updated `objects/commands.cfg` file to add check-ping and `check_ntp` command

```
define command {
    command_name    notify-service-by-email
    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState: $SERVICESTATE$\nDate/Time: $LONGDATETIME$\n\nAdditional Info: $SERVICEOUTPUT$\n" | /bin/mail -s "*** $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/ $SERVICEDESC$ is $SERVICESTATE$" $CTEMAIL$
}

define command {
    command_name    check_ntp
    command_line    $USER1$/check_ntp_time -H $HOSTADDRESS$ -w 5 -c 10
}

define command {
    command_name    check-ping
    command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 100.0,20% -c 500.0,60% -p 5
}
```

Nagios UI

oa links

jira

Pite

confluence

Migration Diagram

AWS updated

IQVIA Links

Docs

Gitlab Ultimate links

build scripts

Nagios

General

Home

Documentation

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary

Grid

Service Groups

Summary

Grid

Problems

Services (Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Current Network Status

Last Updated: Sun Mar 16 08:07:15 UTC 2025

Updated every 90 seconds

Nagios® Core™ 4.5.9 - www.nagios.org

Logged in as nagiosadmin

View Service Status Detail For All Host Groups

View Status Overview For All Host Groups

View Status Summary For All Host Groups

View Status Grid For All Host Groups

Host Status Totals

Up Down Unreachable Pending

2 0 0 0

All Problems All Types

0 2

Service Status Totals

Ok Warning Unknown Critical Pending

8 0 0 1 0

All Problems All Types

1 9

Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
localhost	UP	03-16-2025 08:04:04	4d 20h 34m 54s	PING OK - Packet loss = 0%, RTA = 0.12 ms
server1	UP	03-16-2025 08:06:51	3d 14h 10m 24s	PING OK - Packet loss = 0%, RTA = 0.77 ms

Results 1 - 2 of 2 Matching Hosts

Server1 Information

Nagios

General

Home

Documentation

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary

Grid

Service Groups

Summary

Grid

Problems

Services (Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Availability

Trends

Alerts

History

Host Information

Last Updated: Sun Mar 16 08:09:19 UTC 2025

Updated every 90 seconds

Nagios® Core™ 4.5.9 - www.nagios.org

Logged in as nagiosadmin

View Status Detail For This Host

View Alert History For This Host

View Trends For This Host

View Alert Histogram For This Host

View Availability Report For This Host

View Notifications For This Host

Host

server1

(server1)

Member of

No hostgroups

10.0.0.122

Host State Information

Host Status: UP (for 3d 14h 12m 28s)

Status Information: PING OK - Packet loss = 0%, RTA = 0.77 ms

Performance Data: rta=0.769000ms;100.000000;500.000000;0.000000 pl=0%;20;60;0

Current Attempt: 1/5 (HARD state)

Last Check Time: 03-16-2025 08:06:51

Check Type: ACTIVE

Check Latency / Duration: 0.000 / 4.096 seconds

Next Scheduled Active Check: 03-16-2025 08:11:51

Last State Change: 03-12-2025 17:56:51

Last Notification: N/A (notification 0)

Is This Host Flapping? NO (0.00% state change)

In Scheduled Downtime? NO

Last Update: 03-16-2025 08:09:11 (0d 0h 0m 8s ago)

Active Checks: ENABLED

Passive Checks: ENABLED

Obsessing: ENABLED

Notifications: ENABLED

Event Handler: ENABLED

Flap Detection: ENABLED

Host Comments

Server1 NTP service status

Service Information

Last Updated: Sun Mar 16 08:10:25 UTC 2025
Updated every 90 seconds
Nagios® Core™ 4.5.9 - www.nagios.org
Logged in as nagiosadmin

[View Information For This Host](#)
[View Status Detail For This Host](#)
[View Alert History For This Service](#)
[View Trends For This Service](#)
[View Alert Histogram For This Service](#)
[View Availability Report For This Service](#)
[View Notifications For This Service](#)

Service

ntp_process

On Host

server1

(server1)

Member of

No servicegroups.

10.0.0.122

Service State Information

Current Status: OK (for 3d 14h 12m 20s)

Status Information: NTP OK: Offset -2.312660217e-05 secs, stratum best:3 worst:3

Performance Data: offset=-0.000023s;5;10; stratum_best=3 stratum_worst=3 num_warn_stratum=0 num_crit_stratum=0

Current Attempt: 1/3 (HARD state)

Last Check Time: 03-16-2025 08:08:05

Check Type: ACTIVE

Check Latency / Duration: 0.001 / 6.023 seconds

Next Scheduled Check: 03-16-2025 08:18:05

Last State Change: 03-12-2025 17:58:05

Last Notification: N/A (notification 0)

Is This Service Flapping? NO (0.00% state change)

In Scheduled Downtime? NO

Last Update: 03-16-2025 08:10:21 (0d 0h 0m 4s ago)

Active Checks: ENABLED

Passive Checks: ENABLED

Obsessing: ENABLED

Notifications: ENABLED

Event Handler: ENABLED

Flap Detection: ENABLED

Service Commands

✖ Disable active checks of this service

🕒 Re-schedule the next check of this service

❓ Submit passive check result for this service

✖ Stop accepting passive checks for this service

✖ Stop obsessing over this service

✖ Disable notifications for this service

📧 Send custom service notification

🕒 Schedule downtime for this service

✖ Disable event handler for this service

✖ Disable flap detection for this service

✖ Clear flapping state for this service

Page Tour

Ansible Structure

Directory

```
o tree
├── compose.yaml
├── inventory
│   └── hosts.ini
├── playbooks
│   ├── nagios-server.yml
│   └── ntpd-in-web-server.yml
├── server1-ansible-host-configurations.json
├── ssh-ansible-task.pem
├── templates
│   └── nagios_host_template.cfg.j2
```

- Inventory
 - o All hosts are defined
- Playbooks
 - o 1 for the web-server
 - o 1 for the nagios server
- Templates
 - o Nagios configuration jinja template

Ansible commands to execute

1. `ansible-playbook -i ./inventory/hosts.ini ./playbooks/web-servers.yml`
2. `ansible-playbook -i ./inventory/hosts.ini ./playbooks/nagios-server.yml`

Ansible Output:

```
➔ ansible-poc ansible-playbook -i ./inventory/hosts.ini ./playbooks/nagios-server.yml

PLAY [Deploy monitoring templates on Nagios server] *****
TASK [Gathering Facts] *****
ok: [nagios-srvr]

TASK [Create Nagios host configuration for EC2 instances] *****
ok: [nagios-srvr] => (item=server1)

TASK [Restart Nagios service] *****
changed: [nagios-srvr]

PLAY RECAP *****
nagios-srvr      : ok=3   changed=1   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

➔ ansible-poc ansible-playbook -i ./inventory/hosts.ini ./playbooks/web-servers.yml

PLAY [Configure cron job for logrotate] *****
TASK [Gathering Facts] *****
ok: [server1]

TASK [Run logrotate every 10 minutes between 2 AM and 4 AM] *****
ok: [server1]

PLAY [Deploy ntpd package on EC2 instance] *****
TASK [Gathering Facts] *****
ok: [server1]

TASK [Ensure ntpd is installed] *****
ok: [server1]

TASK [Configure ntp] *****
ok: [server1]

TASK [Ensure ntpd is started and enabled] *****
changed: [server1]

PLAY RECAP *****
server1         : ok=6   changed=1   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
```


Docker -Kubernetes

- Prepare a docker-compose for a nginx server.

Requirements:

- nginx logs need to survive between nginx container restarts
- docker should use network bridge subnet 172.20.8.0/24

Directory Preparation

For the nginx server a simple **nginx.conf** in root dir and **index.html** file has been created inside html folder .

The docker compose is prepared in **compose.yaml** file

```
o tree
├── commands.txt
├── compose.yaml
├── docker-inspect-output.json
├── html
│   └── index.html
├── logs
│   ├── access.log
│   └── error.log
└── nginx.conf

2 directories, 7 files
```

Docker-compose file

- **Volume mounting** : To persist logs between container restart a volume has been mounted from local machine
 - **logs** folder is the mounted path receiving logs
- **Custom Bridge Network**: To have the container an ip from the subnet 172.20.8.0/24, a custom bridge network for docker has been established.

Command to up docker-compose: docker-compose up

```
nginx-test > compose.yaml
1  version: '3.8'
2
3  > Run All Services
4  services:
5    > Run Service
6    nginx:
7      image: nginx:latest
8      container_name: nginx-server
9      ports:
10       - "8080:80"
11      volumes:
12       - ./html:/usr/share/nginx/html
13       - ./nginx.conf:/etc/nginx/nginx.conf
14       - ./logs:/var/log/nginx
15      restart: always
16      networks:
17       - custom_network
18
19 networks:
20   custom_network:
21     driver: bridge
22     ipam:
23       config:
24         - subnet: 172.20.8.0/24
```

Test and validation

Nginx logs:

- Multiple times **docker compose up** and **docker compose down** have been executed.
- The localhost server has been hit multiple times localhost:80/asr , localhost:80/asrfaaf, localhost:80 in this way. The error logs and access logs of the nginx are always there in the logs file **access.log** and **error.log**.

IP Check: A file has been attached docker-inspect-output.json. This file points out the ip of the container which is taken from the container.

```
commands.txt  {} docker-inspect-output.json X  nginx.conf
nginx-server > {} docker-inspect-output.json > {} 0 > {} State > FinishedAt
2      {
224      "NetworkSettings": {
249          "IPAddress": "",
250          "IPPrefixLen": 0,
251          "IPv6Gateway": "",
252          "MacAddress": "",
253          "Networks": {
254              "nginx-test_custom_network": {
255                  "IPAMConfig": null,
256                  "Links": null,
257                  "Aliases": [
258                      "nginx",
259                      "ed5a36768aed"
260                  ],
261                  "NetworkID": "76a1017e90fa9ca47089ab865ca7867e82b4eaa489",
262                  "EndpointID": "dcdc53bdca6ddbc21a60e6871138c429cbb6ee1b5",
263                  "Gateway": "172.20.8.1",
264                  "IPAddress": "172.20.8.2",
265                  "IPPrefixLen": 24,
266                  "IPv6Gateway": "",
267                  "GlobalIPv6Address": "",
268                  "GlobalIPv6PrefixLen": 0,
269                  "MacAddress": "02:42:ac:14:08:02",
270                  "DriverOpts": null
271              }
272          }
273      }
274  }
```

Which Kubernetes command you will use to identify the reason for a pod restart in the project "internal" under namespace "production".

```
kubectl describe pod <pod-name> -n production
```

This will provide the events. From events it is most likely few things

- Crashloopbackoff
 - i. Application error. Pod log checking afterwards.
- OOM (Out of memory)

```
kubectl logs <pod-name> -n production --all-containers=true
```

3) Java-app keep restarting at random. From Kubernetes configuration perspective, what are the possible reasons for the pod restarts?

The most likely scenario is JVM exceeding the memory limit of 1500. As it already has a heap memory of 1000M allocated it can impact the container runtime with memory spike for other java processes.

However standard debugging process needs to be performed side by side

- CPU usage check if CPU is getting throttled or not
- Application crashloopback is happening or not
- Livenessprobe/readinessprobe checks

Helm chart

Kubernetes Cluster Details

Kuberentes Cluster: AWS EKS 1.31 version

Cluster Node AMI : AmazonLinux2

PVC/Storageclass: EFS CSI driver

Errors and problems faced during the helm chart deployment

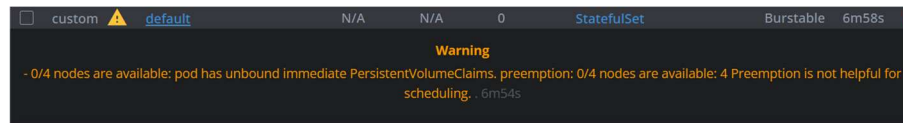
1. Deployed the helm chart with the following command

- `helm upgrade -i elasticsearch ./elasticsearch -f ./elasticsearch/envs/qa/customer-abc-elasticsearch/values.yaml`

• Error1:

- Error: parse error at (elasticsearch/templates/_helpers.tpl:20): unexpected <define> in command
- Resolution: Updating helpers.tpl function -> removing custom logic

• Error2:



○

○ Resolution:

- `helm uninstall elasticsearch`
- Updated the storage-class to aws-efs (Which was already present in the eks cluster)

○ PVC looks fine now

Persistent Volume Claims - k...							
default							
Aa * Search Persistent Volume Claims...							
3 items							
<input type="checkbox"/>	Nan	Namespace	Storage class	Size	Pods	Age	Status
<input type="checkbox"/>	custom	default	aws-efs	10Gi		106s	Bound
<input type="checkbox"/>	custom	default	aws-efs	10Gi		106s	Bound
<input type="checkbox"/>	custom	default	aws-efs	10Gi		106s	Bound

- Error 3:
 - ImagepullBackOff

Summary	Count	Age
Back-off pulling image "elasticsearch-exporter:1.5.0"	2	3m2s
Error: ImagePullBackOff	2	3m2s
Back-off pulling image "logrotate-1.3:1.0.1"	2	3m2s
Error: ImagePullBackOff	1	3m2s
Back-off pulling image "fluentd:v1.14.3-debian-1.0-0"	1	3m2s
Error: ImagePullBackOff	1	3m2s
Failed to pull image "fluentd:v1.14.3-debian-1.0-0": rpc error: code = NotFound	1	3m3s
Error: ErrImagePull	1	3m3s
Failed to pull image "logrotate-1.3:1.0.1": failed to pull and unpack image "dock	1	3m4s
Error: ErrImagePull	1	3m4s

Resolution:

- Updated the statefulset template with different images available in dockerhub
 - fluentd:v1.18.0-debian-1.0
 - blacklabelops/logrotate:1.3
 - bitnami/elasticsearch-exporter:1.5.0
 - Containers started afterwards
- Error4:
 - gelf is missing from the fluentd container.
 - Temporary resolution: Commented out gelf part from the fluentd config

```

elasticsearch > templates > ! configmap.yaml
7  data:
8    fluentd.conf: |
23    <filter graylog2.**>
25    <record>
29      tag ${tag}
30      namespace {{ .Values.global.namespace | quote }}
31    </record>
32    </filter>
33
34    # <match graylog2.**>
35    #   @type gelf
36    #   host graylog
37    #   port 12201
38    #   protocol udp
39    # </match>
40

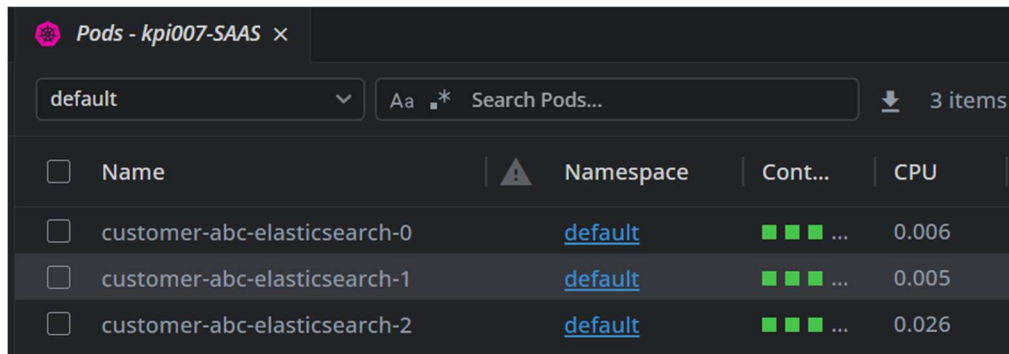
```

-
- **Actual resolution:** Install the gelf package in fluentd image

Final output and validation

Final Output

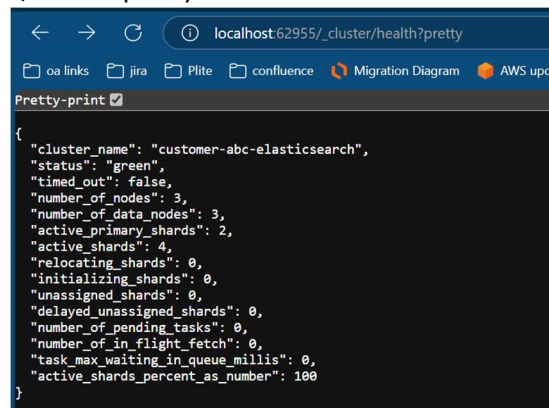
- **Deployed Resources**
 - 1 stateful with 3 replicas
 - 1 service: <http://customer-abc-headless:9200>
 - Configmaps and secrets



<input type="checkbox"/>	Name	Namespace	Cont...	CPU
<input type="checkbox"/>	customer-abc-elasticsearch-0	default	■ ■ ■ ...	0.006
<input type="checkbox"/>	customer-abc-elasticsearch-1	default	■ ■ ■ ...	0.005
<input type="checkbox"/>	customer-abc-elasticsearch-2	default	■ ■ ■ ...	0.026

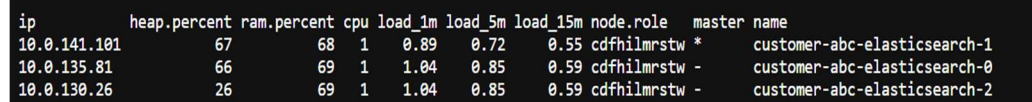
Validation

- After port-forwarding the service <http://customer-abc-headless:9200> to local-browser, triggered the following endpoints
 - `/_cluster/health?pretty`



```
localhost:62955/_cluster/health?pretty
oa links jira Plite confluence Migration Diagram AWS upd
Pretty-print [x]
{
  "cluster_name": "customer-abc-elasticsearch",
  "status": "green",
  "timed_out": false,
  "number_of_nodes": 3,
  "number_of_data_nodes": 3,
  "active_primary_shards": 2,
  "active_shards": 4,
  "relocating_shards": 0,
  "initializing_shards": 0,
  "unassigned_shards": 0,
  "delayed_unassigned_shards": 0,
  "number_of_pending_tasks": 0,
  "number_of_in_flight_fetch": 0,
  "task_max_waiting_in_queue_millis": 0,
  "active_shards_percent_as_number": 100
}
```

- `_cat/nodes?v`



ip	heap.percent	ram.percent	cpu	load_1m	load_5m	load_15m	node.role	master	name
10.0.141.101	67	68	1	0.89	0.72	0.55	cdfhilmrstw *		customer-abc-elasticsearch-1
10.0.135.81	66	69	1	1.04	0.85	0.59	cdfhilmrstw -		customer-abc-elasticsearch-0
10.0.130.26	26	69	1	1.04	0.85	0.59	cdfhilmrstw -		customer-abc-elasticsearch-2

Metrics

Explain how Prometheus work

As a complete monitoring toolkit, with Prometheus, sophisticated and ideal monitoring system can be implemented.

To simply put, how Prometheus works in a Kubernetes cluster, understanding 3 flows will be sufficient

- Scraping metrics with Service discovery: k8s services need to expose metrics for Prometheus. Prometheus service discovery will point these as target endpoints to scrape metrics from
- Time-Series Database: Prometheus aggregates all recorded metrics in a time-series database through Prometheus server
- Leverage promql & alertmanager: With flexible query language like promql, metrics can be parsed and multiple tools are needed to be integrated with Prometheus like alertmanager, Grafana.

How do you create custom Prometheus alerts and alerting rules for Kubernetes monitoring? Provide an example alert rule and its configuration.

Explaining the answer with the context of most popular and community standard Prometheus helm chart Kube-prometheus-stack. helm-charts/charts/kube-prometheus-stack at main · prometheus-community/helm-charts

Ideally to configure an alert, a promql needs to be placed against a specific service. The promql's responsibility will be to monitor scraped metrics against a specific time frame. The alert needs to be placed in helm chart from automation perspective.

Example Alert

In the previous task, elasticsearch statefulset had 3 replicas created. So, a dummy alert has been created where if the statefulset replica count is 4, an alert will be triggered.

Alert syntax added in the values.yaml file of the helm chart

```
! values.yaml M X
platform > pl-environment-prometheus > src > kube-prometheus-stack-65.5.0 > ! values.yaml
193 # - record: my_record
194 #   expr: 100 * my_record
195
196 ## Provide custom recording or alerting rules to be deployed into the cluster.
197 ##
198 additionalPrometheusRulesMap:
199   statefulset-replica-monitoring:
200     groups:
201     - name: statefulset_replica_count
202       rules:
203       - alert: StatefulSetReplicaCount
204         expr: kube_statefulset_status_replicas_ready{namespace="default", statefulset=~".+"} != 4
205         for: 2m
206         labels:
207           severity: warning
208         annotations:
209           summary: "StatefulSet {{ $labels.statefulset }} has replica count mismatch"
210           description: "StatefulSet {{ $labels.statefulset }} in namespace {{ $labels.namespace }} has {{ $value }} replicas ready,
211             which does not match the desired replica count."
```

Created alert in alertmanager UI

/etc/prometheus/rules/prometheus-prometheus-grafana-prometheus-rulefiles-0/monitoring-kube-prometheus-stack-statefulset-replica-monitoring-a92dd602-25d9-4c18-800a-5635aa41126a.yaml > statefulset_replica_count firing (1)

▼ StatefulSetReplicaCount (1 active)

name: StatefulSetReplicaCount

expr: kube_statefulset_status_replicas_ready{namespace="default",statefulset=~".+"} != 4

for: 2m

labels:

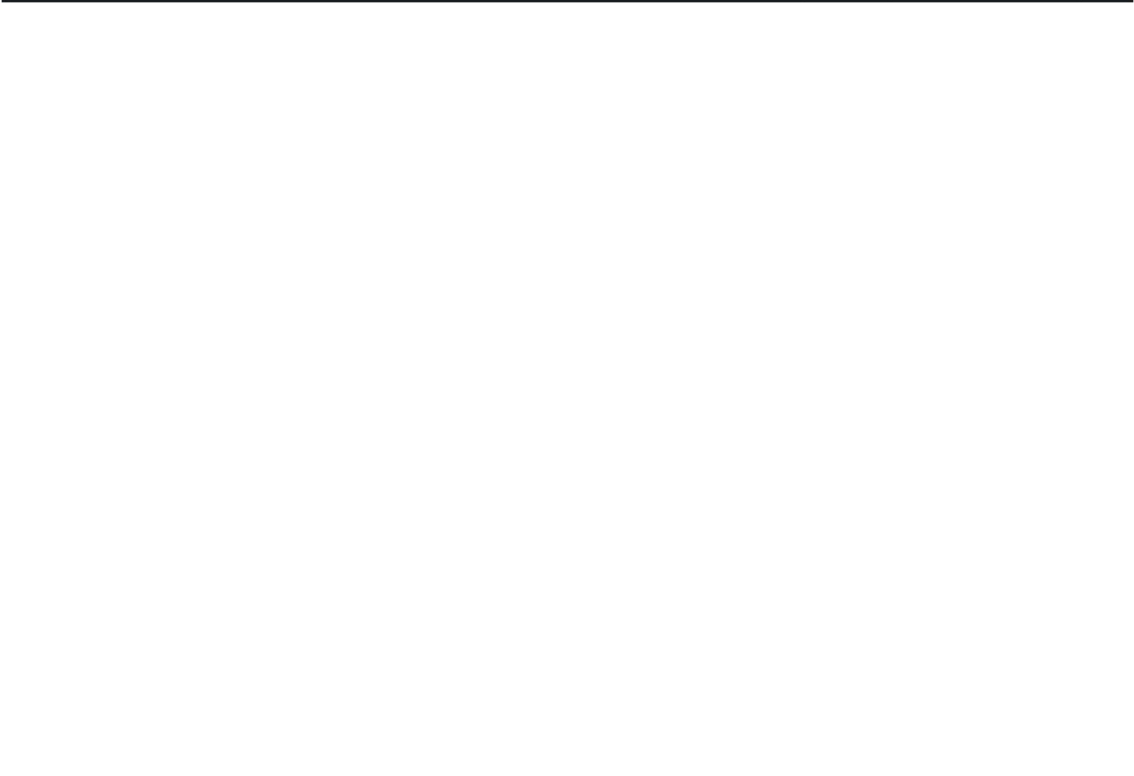
severity: warning

annotations:

description: StatefulSet {{ \$labels.statefulset }} in namespace {{ \$labels.namespace }} has {{ \$value }} replicas ready, which does not match the desired replica count

summary: StatefulSet {{ \$labels.statefulset }} has replica count mismatch

Labels	State	Active Since	Value
alertname=StatefulSetReplicaCount container=kube-state-metrics endpoint=http instances=10.0.144.102:8080 job=kube-state-metrics namespace=default pod=monitoring-kube-state-metrics-78f676469c-rwhdm service=monitoring-kube-state-metrics severity=warning statefulset=customer-abc-elasticsearch	FIRING	2025-03-15T20:22:45.059164355Z	3



What is the Prometheus query you can use in Granfana to properly show usage trend of an application metric that is a counter?

For a counter type metric, we need to use `rate()` type promql. Let's say for a metric name `app_hits` is a counter whose value gets scraped through Prometheus-operator. For the trend of this metric, `rate(app_hits)[10m]` will be an ideal query.

As example,

One promql is prepared dedicated to monitor a pod of the statefulset where the metric `container_cpu_usage_seconds_total` is a counter

```
rate(container_cpu_usage_seconds_total{container="elasticsearch",namespace="default",pod="customer-abc-elasticsearch-0"}[5m])
```



Database

Cassandra - > Not familiar with Cassandra

Mongo -> Replicaset and Sharding

In the problem statement, due to performance issue, a new replicaset has been provisioned.

Multiple replicaset for mongo:

In a mongo cluster we can have multiple replicaset. The primary node and other replicas. However only the primary node is responsible for the write operations whereas other nodes are only present to handle the replica data and will be used only for the read operations.

Sharding

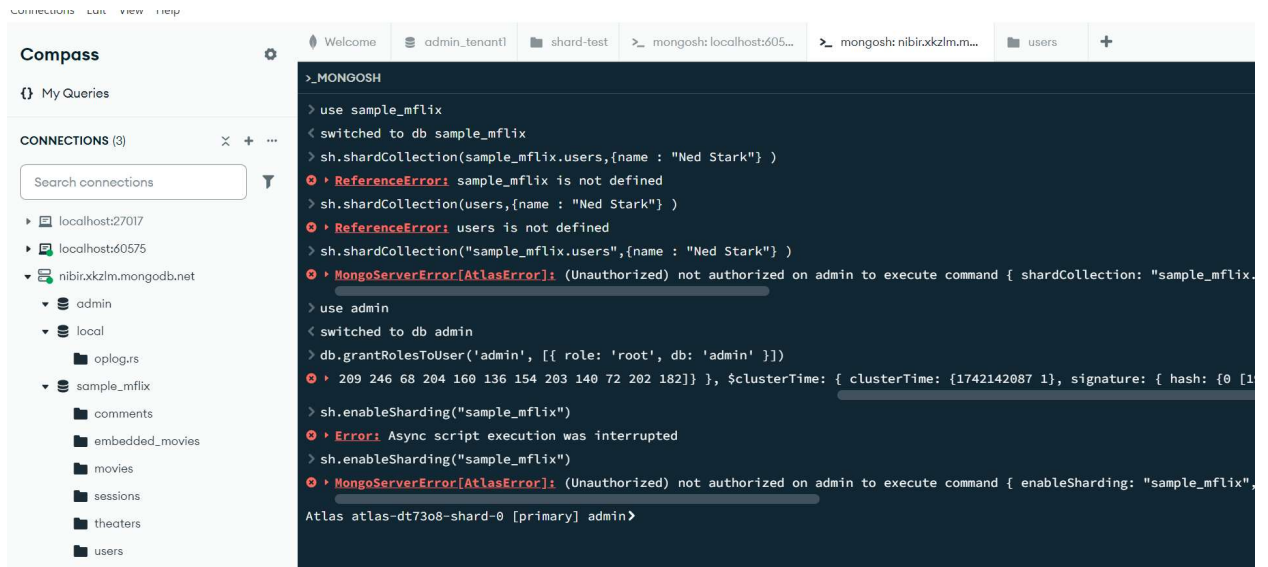
By sharding, storage will be distributed from the primary node. That means, let's say previously primary node had a total amount of 500 GB of a db. Now it will be in two parts. Depends on the key, let's assume 300gb will stay in one node and the rest will be moved to other replicaset.

How to perform sharding

- **Sharding keys:** Index needs to be prepared on the sharding key. As based on the index, sharding modes will be applied (hashed/ranged).
- **Config server and mongos service:** Config servers hold metadata of the shards and **mongos service** routes queries to appropriate shard.
- Executables
 - a. Enable sharding on the db :
`sh.enableSharding("sample_db")`
 - b. Shard collection based on the key:
`sh.shardCollection("sample_db.sample_collection ", { _id: "hashed_key" })`

Problems faced for simulating the scenario

1. **Mongo Kubernetes operator:** Deployed mongo as a Kubernetes operator. However, the community mongo operator doesn't support sharding. [mongodb/mongodb-kubernetes-operator: MongoDB Community Kubernetes Operator](#)
Mongos service doesn't get deployed with this operator and hence sharding isn't possible.
2. **MongoDB Atlas cluster free tier:** An mongodb atlas cluster was prepared to simulate sharding. In the free tier, it doesn't support sharding.



AWS DocumentDB(Mongo Compatible) : Not tested, but seems like possible [Choose shard keys to optimize Amazon DocumentDB Elastic Clusters | AWS Database Blog](#)