



DevOps Shack

Ultimate DevOps Monitoring Project

[Click Here To Enrol To Batch-5 | DevOps & Cloud DevOps](#)

Prerequisites

- Ensure you have `wget` and `tar` installed on both VMs.
- Ensure you have appropriate permissions to download, extract, and run these binaries.
- Replace `<version>` with the appropriate version number you wish to download.

VM-1 (Node Exporter)

1. Download Node Exporter

```
wget  
https://github.com/prometheus/node_exporter/releases/download/v1.8.1/  
node_exporter-1.8.1.linux-amd64.tar.gz
```

2. Extract Node Exporter

```
tar xvfz node_exporter-1.8.1.linux-amd64.tar.gz
```

3. Start Node Exporter

```
4. cd node_exporter-1.8.1.linux-amd64  
./node_exporter &
```

VM-2 (Prometheus, Alertmanager, Blackbox Exporter)

Prometheus

1. Download Prometheus

```
wget
https://github.com/prometheus/prometheus/releases/download/v2.52.0/prometheus-2.52.0.linux-amd64.tar.gz
```

2. Extract Prometheus

```
tar xvfz prometheus-2.52.0.linux-amd64.tar.gz
```

3. Start Prometheus

4. `cd prometheus-2.52.0.linux-amd64`
`./prometheus --config.file=prometheus.yml &`

Alertmanager

1. Download Alertmanager

```
wget
https://github.com/prometheus/alertmanager/releases/download/v0.27.0/alertmanager-0.27.0.linux-amd64.tar.gz
```

2. Extract Alertmanager

```
tar xvfz alertmanager-0.27.0.linux-amd64.tar.gz
```

3. Start Alertmanager

4. `cd alertmanager-0.27.0.linux-amd64`
`./alertmanager --config.file=alertmanager.yml &`

Blackbox Exporter

1. Download Blackbox Exporter

```
wget
https://github.com/prometheus/blackbox_exporter/releases/download/v0.25.0/blackbox_exporter-0.25.0.linux-amd64.tar.gz
```

2. Extract Blackbox Exporter

```
tar xvfz blackbox_exporter-0.25.0.linux-amd64.tar.gz
```

3. Start Blackbox Exporter

4. `cd blackbox_exporter-0.25.0.linux-amd64`
`./blackbox_exporter &`

Notes:

- The `&` at the end of each command ensures the process runs in the background.

- Ensure that you have configured the `prometheus.yml` and `alertmanager.yml` configuration files correctly before starting the services.
 - Adjust the firewall and security settings to allow the necessary ports (typically 9090 for Prometheus, 9093 for Alertmanager, 9115 for Blackbox Exporter, and 9100 for Node Exporter) to be accessible.
-

Prometheus and Alertmanager Configuration

Prometheus Configuration (`prometheus.yml`)

Global Configuration

```
global:
  scrape_interval: 15s           # Set the scrape interval to every 15
seconds. Default is every 1 minute.
  evaluation_interval: 15s       # Evaluate rules every 15 seconds.
The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).
```

Alertmanager Configuration

```
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - 'localhost:9093'       # Alertmanager endpoint
```

Rule Files

```
rule_files:
  - "alert_rules.yml"           # Path to alert rules file
  # - "second_rules.yml"       # Additional rule files can be added
here
```

Scrape Configuration

Prometheus Itself

```
scrape_configs:
  - job_name: "prometheus"      # Job name for Prometheus

  # metrics_path defaults to '/metrics'
```

```

# scheme defaults to 'http'.

static_configs:
  - targets: ["localhost:9090"]    # Target to scrape (Prometheus
itself)

```

Node Exporter

```

- job_name: "node_exporter"        # Job name for node exporter

# metrics_path defaults to '/metrics'
# scheme defaults to 'http'.

static_configs:
  - targets: ["3.110.195.114:9100"] # Target node exporter endpoint

```

Blackbox Exporter

```

- job_name: 'blackbox'              # Job name for blackbox exporter
  metrics_path: /probe              # Path for blackbox probe
  params:
    module: [http_2xx]             # Module to look for HTTP 200
  response
  static_configs:
    - targets:
      - http://prometheus.io        # HTTP target
      - https://prometheus.io       # HTTPS target
      - http://3.110.195.114:8080/   # HTTP target with port 8080
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: 13.235.248.225:9115 # Blackbox exporter address

```

Alert Rules Configuration (alert_rules.yml)

Alert Rules Group

```

groups:
- name: alert_rules                # Name of the alert rules group
  rules:
    - alert: InstanceDown
      expr: up == 0                 # Expression to detect instance down
      for: 1m
      labels:
        severity: "critical"
      annotations:
        summary: "Endpoint {{ $labels.instance }} down"
        description: "{{ $labels.instance }} of job {{ $labels.job }} has
been down for more than 1 minute."

    - alert: WebsiteDown
      expr: probe_success == 0      # Expression to detect website down
      for: 1m
      labels:

```

```

        severity: critical
    annotations:
        description: The website at {{ $labels.instance }} is down.
        summary: Website down

- alert: HostOutOfMemory
  expr: node_memory_MemAvailable / node_memory_MemTotal * 100 < 25 #
Expression to detect low memory
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Host out of memory (instance {{ $labels.instance }})"
    description: "Node memory is filling up (< 25% left)\n VALUE = {{
$value }}\n LABELS: {{ $labels }}"

- alert: HostOutOfDiskSpace
  expr: (node_filesystem_avail{mountpoint="/" } * 100) /
node_filesystem_size{mountpoint="/" } < 50 # Expression to detect low disk
space
  for: 1s
  labels:
    severity: warning
  annotations:
    summary: "Host out of disk space (instance {{ $labels.instance }})"
    description: "Disk is almost full (< 50% left)\n VALUE = {{ $value
}}\n LABELS: {{ $labels }}"

- alert: HostHighCpuLoad
  expr: (sum by (instance)
(irate(node_cpu{job="node_exporter_metrics",mode="idle"}[5m]))) > 80 #
Expression to detect high CPU load
  for: 5m
  labels:
    severity: warning
  annotations:
    summary: "Host high CPU load (instance {{ $labels.instance }})"
    description: "CPU load is > 80%\n VALUE = {{ $value }}\n LABELS:
{{ $labels }}"

- alert: ServiceUnavailable
  expr: up{job="node_exporter"} == 0 # Expression to detect service
unavailability
  for: 2m
  labels:
    severity: critical
  annotations:
    summary: "Service Unavailable (instance {{ $labels.instance }})"
    description: "The service {{ $labels.job }} is not available\n
VALUE = {{ $value }}\n LABELS: {{ $labels }}"

- alert: HighMemoryUsage
  expr: (node_memory_Active / node_memory_MemTotal) * 100 > 90 #
Expression to detect high memory usage
  for: 10m
  labels:
    severity: critical
  annotations:
    summary: "High Memory Usage (instance {{ $labels.instance }})"
    description: "Memory usage is > 90%\n VALUE = {{ $value }}\n
LABELS: {{ $labels }}"

```

```

    - alert: FileSystemFull
      expr: (node_filesystem_avail / node_filesystem_size) * 100 < 10 #
Expression to detect file system almost full
      for: 5m
      labels:
        severity: critical
      annotations:
        summary: "File System Almost Full (instance {{ $labels.instance
    }}})"
        description: "File system has < 10% free space\n  VALUE = {{ $value
    }}\n  LABELS: {{ $labels }}"

```

Alertmanager Configuration (alertmanager.yml)

Routing Configuration

```

route:
  group_by: ['alertname']           # Group by alert name
  group_wait: 30s                  # Wait time before sending the first
notification
  group_interval: 5m               # Interval between notifications
  repeat_interval: 1h              # Interval to resend notifications
  receiver: 'email-notifications'  # Default receiver

receivers:
- name: 'email-notifications'      # Receiver name
  email_configs:
  - to: jaiswaladi246@gmail.com     # Email recipient
    from: test@gmail.com            # Email sender
    smarthost: smtp.gmail.com:587   # SMTP server
    auth_username: your_email       # SMTP auth username
    auth_identity: your_email       # SMTP auth identity
    auth_password: "bdmq omqh vvkz zoqx" # SMTP auth password
    send_resolved: true             # Send notifications for resolved
alerts

```

Inhibition Rules

```

inhibit_rules:
- source_match:
  severity: 'critical'             # Source alert severity
  target_match:
  severity: 'warning'              # Target alert severity
  equal: ['alertname', 'dev', 'instance'] # Fields to match

```