

Web Scraping and Visualization with Prometheus and Grafana

Configuration of blackbox in YAML file

```
blackbox.yml - Notepad
File Edit Format View Help

tcp:
  query_response:
    - expect: "^+OK"
    - send: "PING"
  tls: true
  tls_config:
    insecure_skip_verify: false
grpc:
  proben: grpc
  grpc:
    tls: true
    preferred_ip_protocol: "ip4"
grpc_plain:
  proben: grpc
  grpc:
    tls: false
    service: "service1"
ssh_banner:
  proben: tcp
  tcp:
    query_response:
      - expect: "^SSH-2.0-"
      - send: "SSH-2.0-blackbox-ssh-check"
irc_banner:
  proben: tcp
  tcp:
    query_response:
      - send: "NICK proben"
      - send: "USER proben proben :proben"
      - expect: "PING :([^\s]+)"
      - send: "PONG $1"
      - expect: "^[^ ]+ 001"
icmp:
  proben: icmp
```

Ln 32, Col 9 100% Unix (LF) UTF-8

Activate Windows
Go to Settings to activate Windows.

Configuring prometheus in YAML file. Setting regres.in as target.

```
prometheus.yml - Notepad
File Edit Format View Help

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

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

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "google-blackbox"

    metrics_path: /probe
    params:
      module: [http_2xx] # Use the module defined in your blackbox.yml
    static_configs:
      - targets: ["localhost:9115"]
  - job_name: "regres-blackbox"

    metrics_path: /probe
    params:
      module: [http_2xx] # Use the module defined in your blackbox.yml
      target: ["http://regres.in"]
    static_configs:
      - targets: ["localhost:9115"]
```

Ln 38, Col 1 100% Unix (LF) UTF-8

Activate Windows
Go to Settings to activate Windows.

State is normal

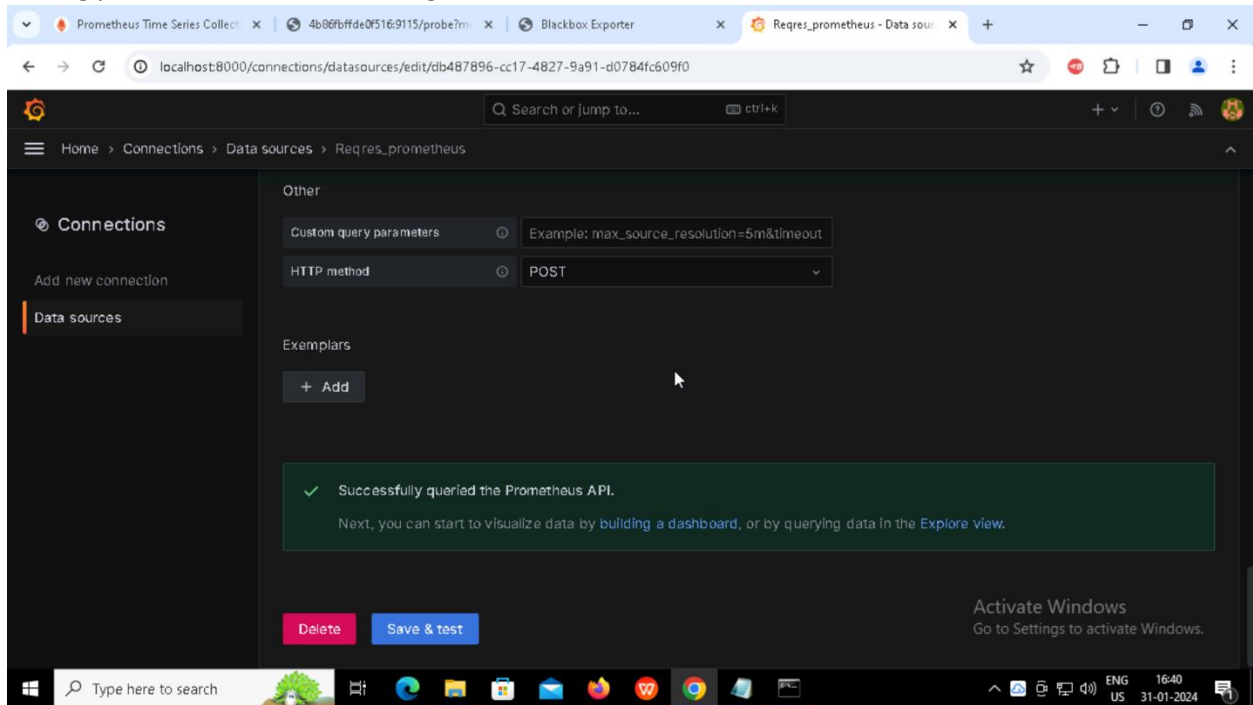
The screenshot shows the Prometheus web interface at localhost:9090. The 'Targets' page displays a list of scrape pools. The 'reqres-blackbox' pool has 1/1 targets up. The table below shows the details of the target.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9115/probe module="http_2xx" target="http://reqres.in"	UP	instance="localhost:9115" job="reqres-blackbox"	28.915s ago	796.212ms	

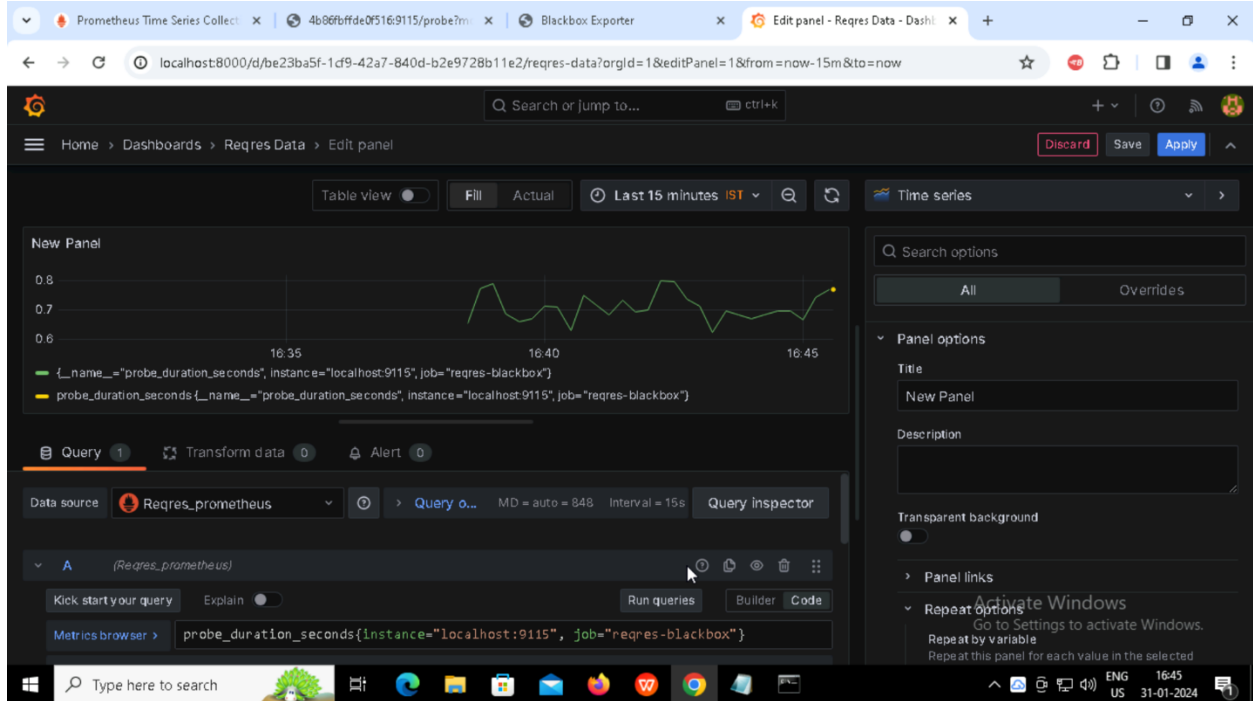
This screenshot shows the same Prometheus interface, but the 'reqres-blackbox' target is expanded. It reveals a list of discovered labels for the target.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9115/probe module="http_2xx" target="http://reqres.in"	UP	instance="localhost:9115" job="reqres-blackbox" Discovered labels: __address__="localhost:9115" __metrics_path__="/probe" __param_module="http_2xx" __param_target="http://reqres.in" __scheme__="http" __scrape_interval__="15s" __scrape_timeout__="10s" job="reqres-blackbox"	28.915s ago	796.212ms	

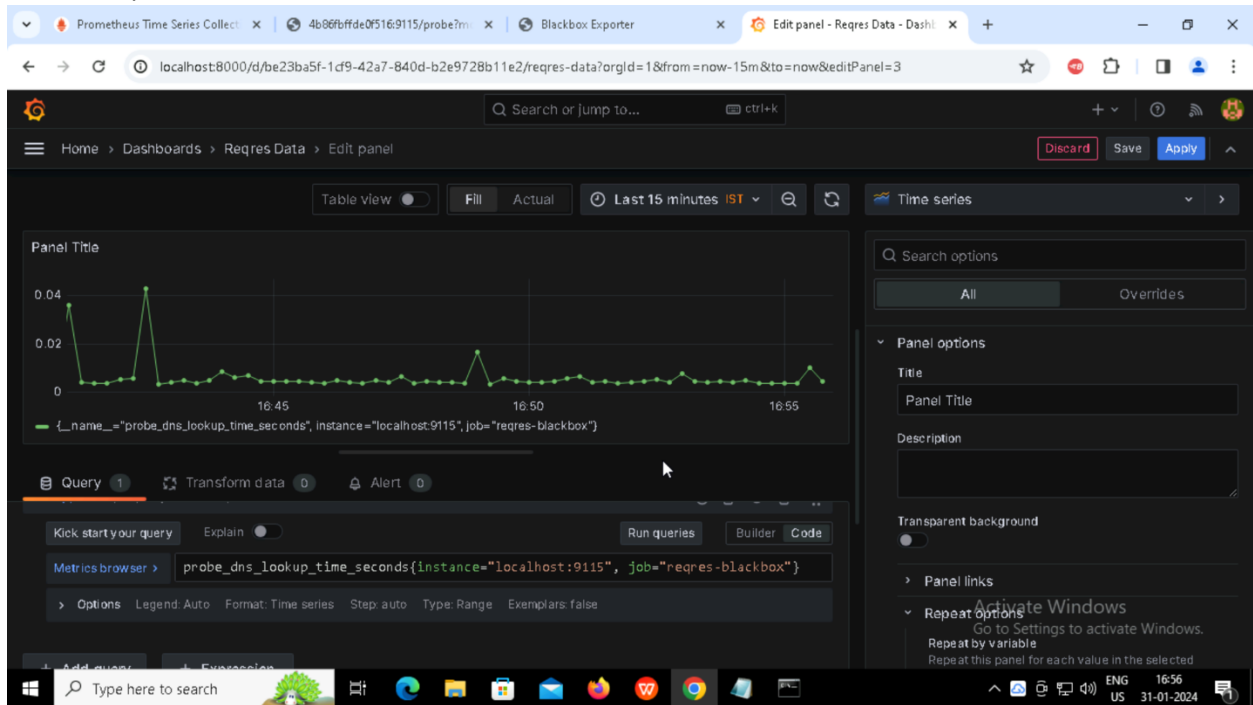
Adding prometheus data source in grafana



Adding query and running it to check the duration seconds



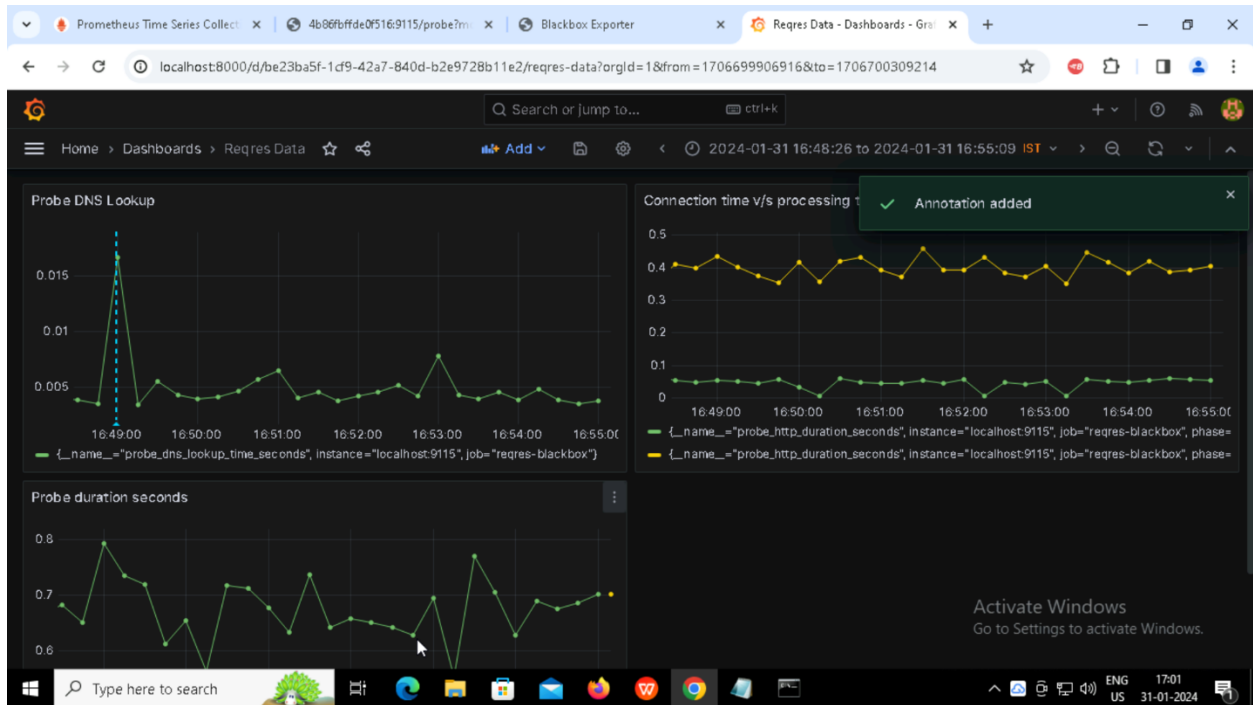
DNS lookup time



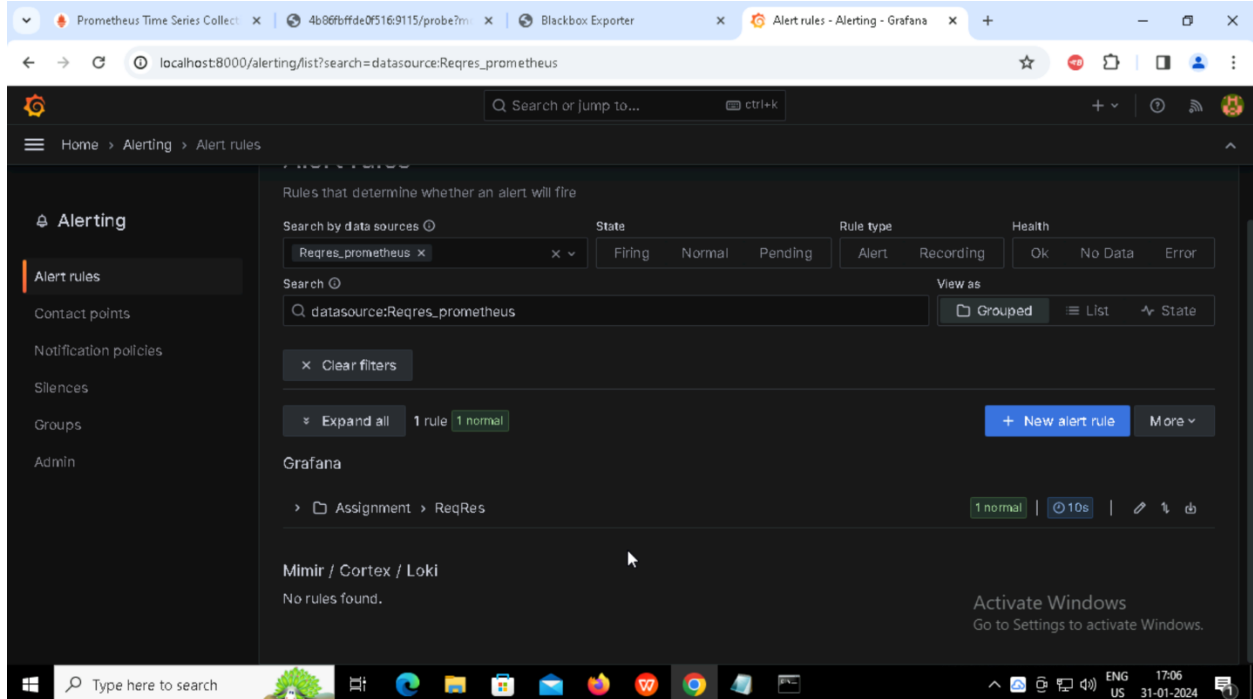
Adding annotation



Final dashboard



Adding alert



After adding alert, checked and found no abnormality

The screenshot displays the Grafana Alerting interface in a web browser. The browser's address bar shows the URL `localhost:8000/alerting/list`. The Grafana interface has a dark theme. On the left, a sidebar contains navigation links: 'Alerting', 'Alert rules' (selected), 'Contact points', 'Notification policies', 'Silences', 'Groups', and 'Admin'. The main content area shows a list of alert rules. At the top, a summary bar indicates '4 rules' with '1 firing', '1 errors', and '2 normal' status. Below this, the 'Grafana' section lists three alert rule groups: 'Alerts > Evaluation group1' (1 firing, 10s), 'Alerts > ITDepotEvaluation' (1 errors, 1 normal, 5m), and 'Assignment > ReqRes' (1 normal, 10s). The 'ReqRes' rule is expanded, showing a table with columns: State, Name, Health, Summary, Next evaluation, and Actions. The table contains one entry with State 'Normal', Name 'Threshold value exceeded', Health 'ok', Summary 'Limit exceeded', and Next evaluation 'within 10 seconds'. A context menu is open over the 'ReqRes' rule, showing options: 'Duplicate', 'Modify export', and 'Delete'. At the bottom of the main content area, there is a section for 'Mimir / Cortex / Loki' with the text 'No rules found.' and an 'Activate Windows' watermark. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system clock in the bottom right corner shows '17:10' and '31-01-2024'.

State	Name	Health	Summary	Next evaluation	Actions
Normal	Threshold value exceeded	ok	Limit exceeded	within 10 seconds	Duplicate, Modify export, Delete