

Ncore Software Prometheus Copilot

Getting Start Guide

Product guide | September 2026

Ask questions about your metrics in natural language, visualize Prometheus results, and reuse saved queries to reduce AI calls.

Table of Contents

1. Requirements 3
2. Getting started 4
3. Welcome to Ncore Prometheus Copilot 5
4. Ask and explore your metrics 6
5. Data and PromQL lineage 7
6. Monitor multiple instances with auto-refreshing charts 8
7. Saved Queries 9
8. History 10
9. Alert configuration 11
10. Create and publish an alert 12
11. Sample Questions 13
12. Search metrics optional 14
13. AI Usage 15
14. License and TLS settings 16
15. How to write good questions 17
16. Security notes 17
17. Troubleshooting 18
18. First Query in 60 Seconds 19
    What you should see 19

1. Requirements

Requirement	Example or check
Docker on Ubuntu 24.04	sudo apt update sudo apt install docker.io docker-compose-v2 wget
Docker on RHEL	sudo dnf install dnf-plugins-core wget sudo dnf config-manager --add-repo https://download.docker.com/linux/rhel/docker-ce.repo sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Start and verify Docker	sudo systemctl enable --now docker sudo docker info sudo docker compose version
Network access	Allow Copilot to reach Prometheus (typically TCP 9090) and Alertmanager (typically TCP 9093). Allow Prometheus to reach Linux node_exporter (typically TCP 9100) and Windows windows_exporter (typically TCP 9182). Allow outbound HTTPS access to the configured managed AI service.

2. Getting started

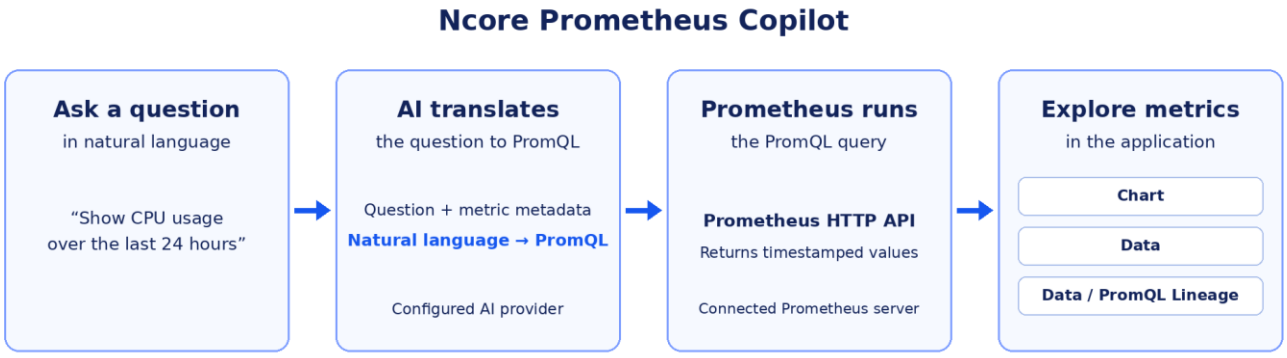
Download file: **wget https://ncoreblobstore.blob.core.windows.net/prometheus-web/deploy-prometheus-copilot.sh**

Run on linux host-→ **bash ./deploy-prometheus-copilot.sh**

- **Open Ncore Prometheus Copilot→** Go to http://192.168.10.4:5188. Replace the example IP with your host address. The URL will open a login page use the following credentials [**username**: Administrator, **password**: admin].
- **Go to License →** to obtain trial license from Ncore software, see page **Error! Bookmark not defined.**6 for illustrated steps
- **After Activation go to → User accounts** to add user entitlements application access.
- Review illustrated "How To" steps below for queries, alert notification, and other features.

3. Welcome to Ncore Prometheus Copilot

Ncore Prometheus Copilot brings AI-assisted analysis, visualization, and alert management to your Prometheus environment. Ask questions in everyday language, investigate performance across Windows and Linux systems, and monitor multiple instances through automatically refreshing charts. With tools for creating alert rules, inspecting query results, and reusing analyses, it helps teams move from metrics to informed action.



Ncore Prometheus Copilot overview

Your metric data stays between Prometheus and the application. Ncore uses your question and selected metric metadata to generate PromQL. Metric values and query results are not sent to AI.

4. Ask and explore your metrics

Enter a question and choose Ask Prometheus. The AI generates PromQL, Prometheus executes it, and the chart displays the returned values. The numbered markers identify the main controls, including the two detail buttons beneath the chart.

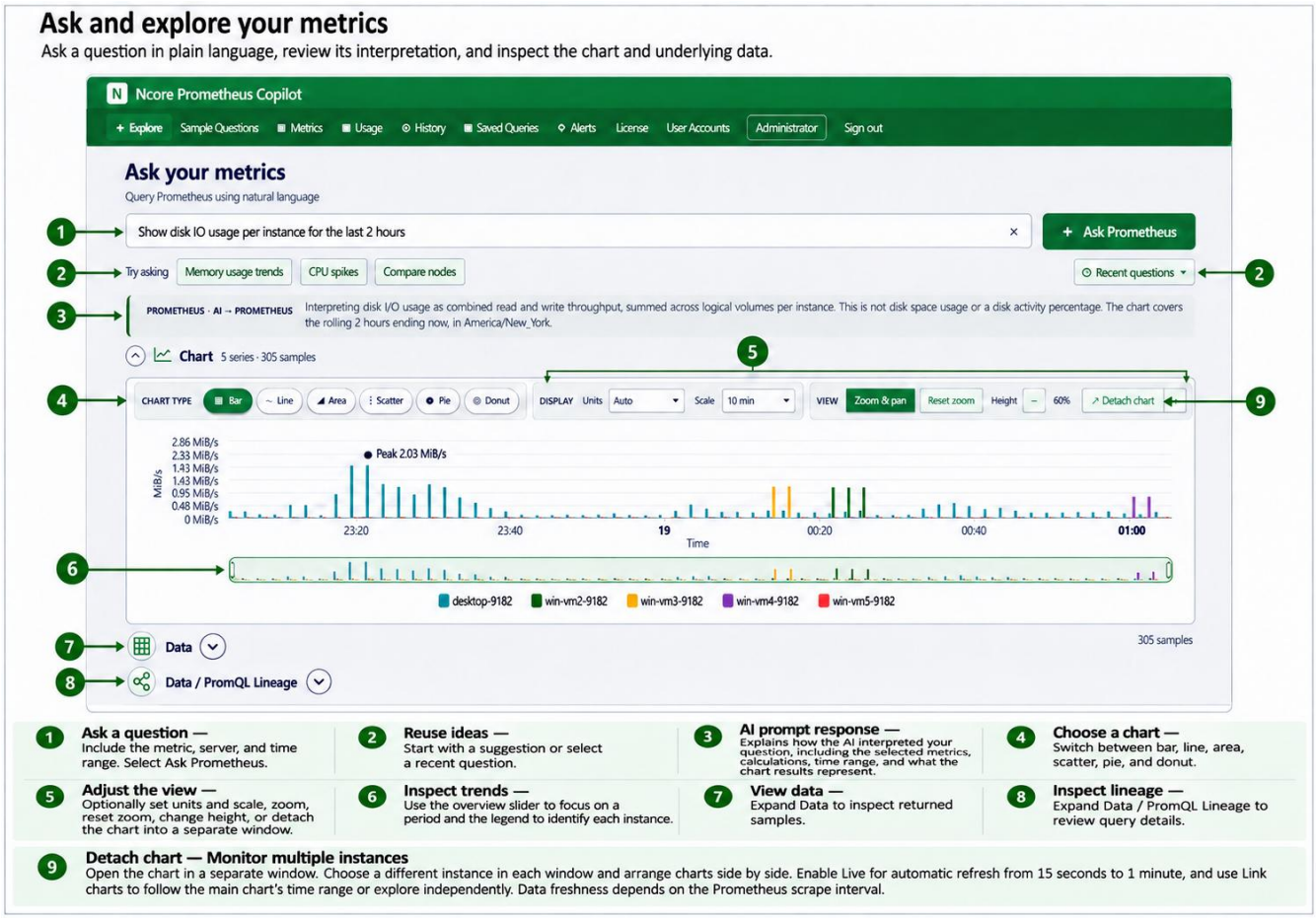


Figure 2 Main window feature markers

5. Data and PromQL lineage

Choose Data beneath the main chart (marker 7 in Figure 2). Each row shows a timestamp, a series, and its value. Open Data / PromQL Lineage (marker 8 in Figure 2) to inspect the source metrics and transformations behind the query.

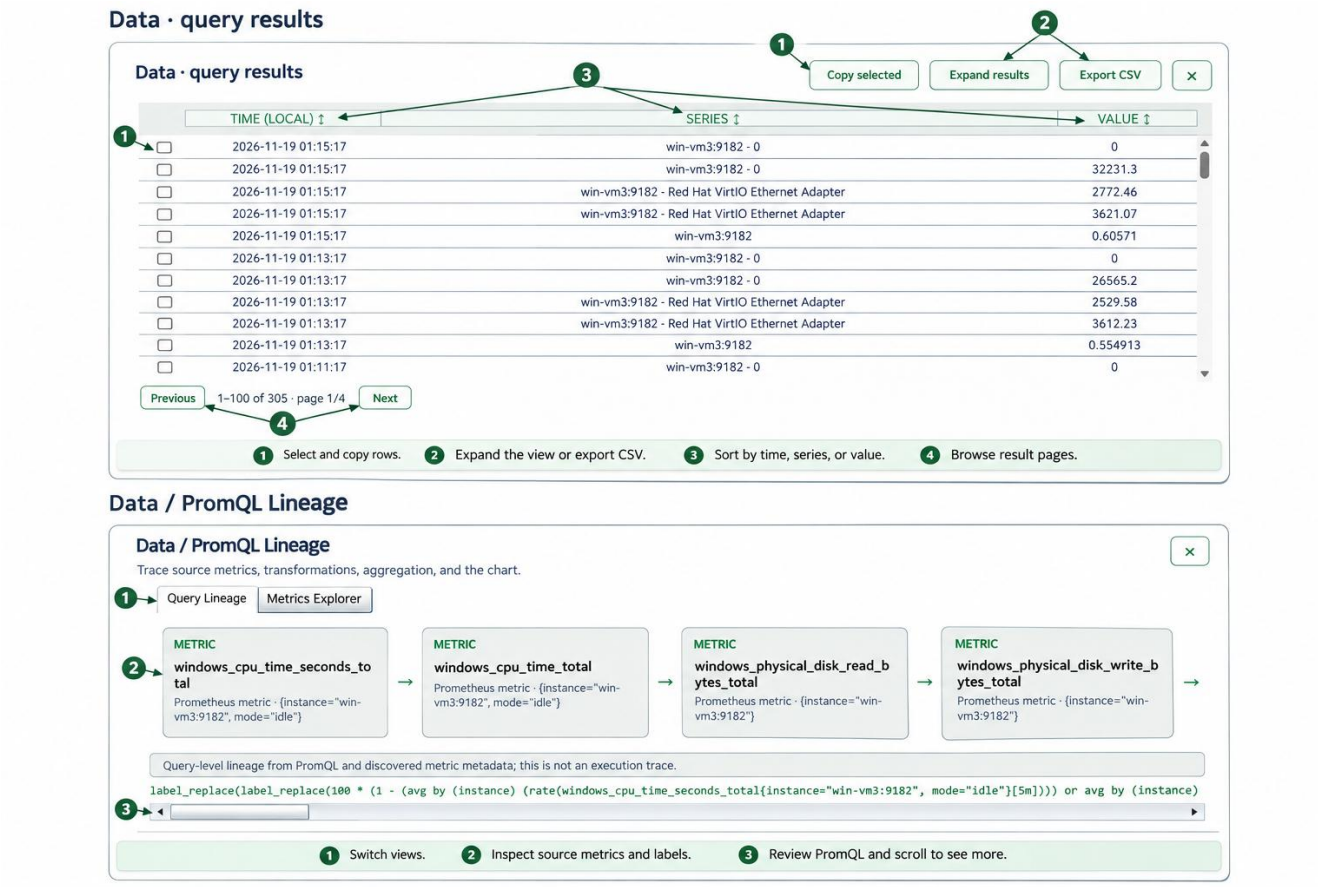
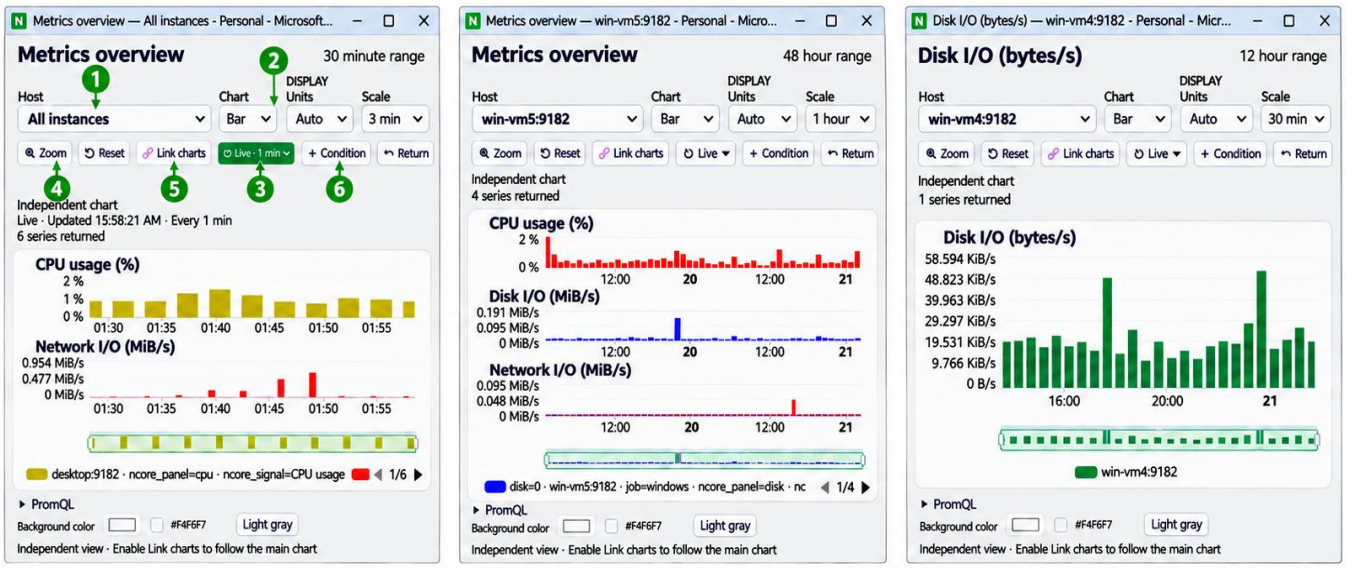


Figure 3 Data card opened from the main window Data button

6. Monitor multiple instances with auto-refreshing charts

Open separate chart windows for different instances and track metrics with automatic refresh intervals from 15 seconds to 1 minute. Data freshness depends on the configured Prometheus scrape interval.



1. **Select an instance** — Choose a host for each independent window.
2. **Display settings** — Optional. Keep the defaults or change chart type, units, and scale.
3. **Live** — Refresh automatically every 15 seconds to 1 minute.
4. **Explore the data** — Zoom, reset, or drag the time navigator.
5. **Link charts** — Follow the main chart's instance and time range.
6. **Condition** — Highlight a threshold such as CPU above 80%. Use Live for ongoing checks.

Chart conditions are separate from published alert rules.

7. Saved Queries

Saved Queries reduce repeat AI calls. Once AI has generated a useful PromQL query, save it and reuse that query when you need the same analysis again. Running saved PromQL sends it directly to Prometheus; it does not ask AI to translate the original question again.

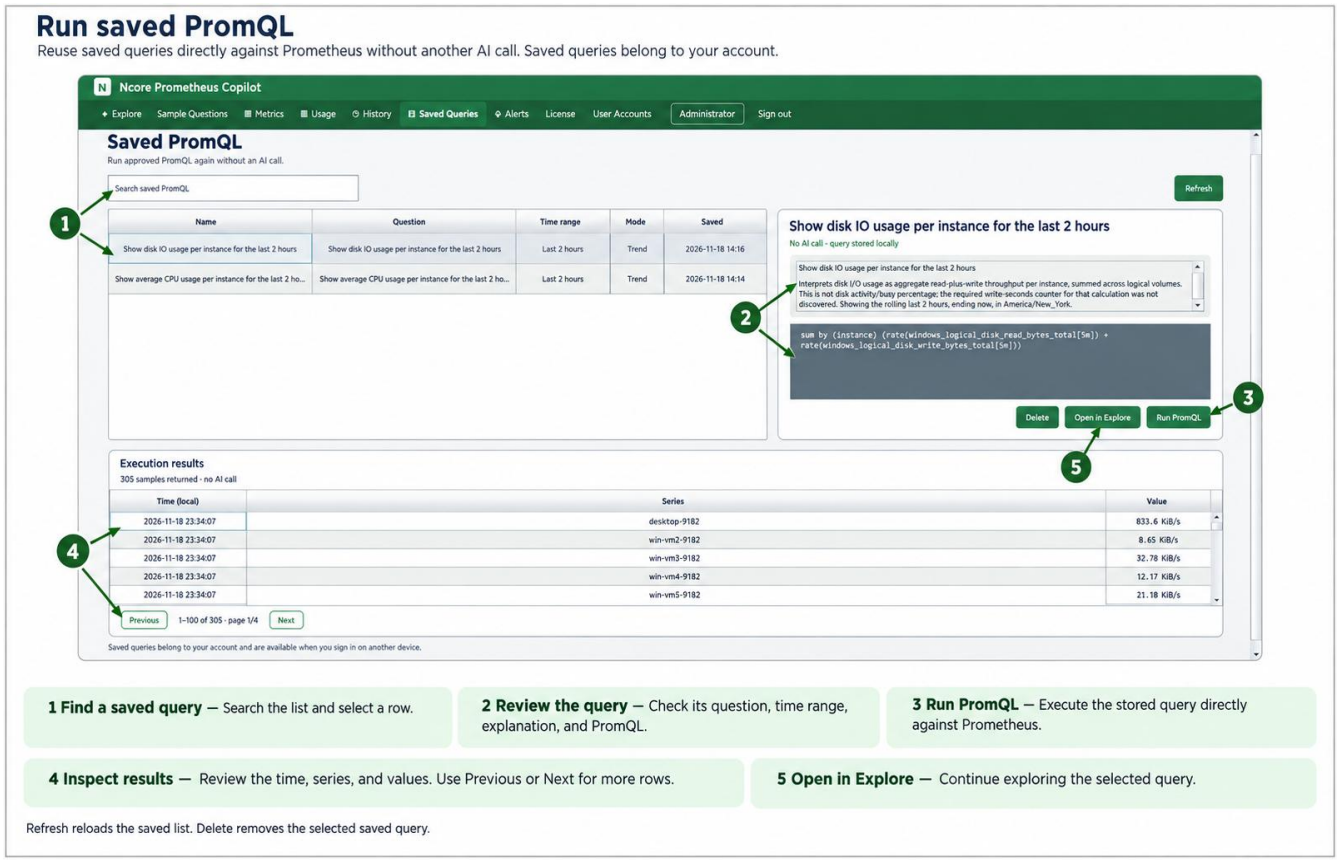


Figure 5 Saved Queries and execution results

8. History

Use History to review and reopen recent queries. The latest 20 distinct queries are retained in your account. Choose Details to inspect a query or Open in Explorer to run it again against Prometheus without another AI translation.

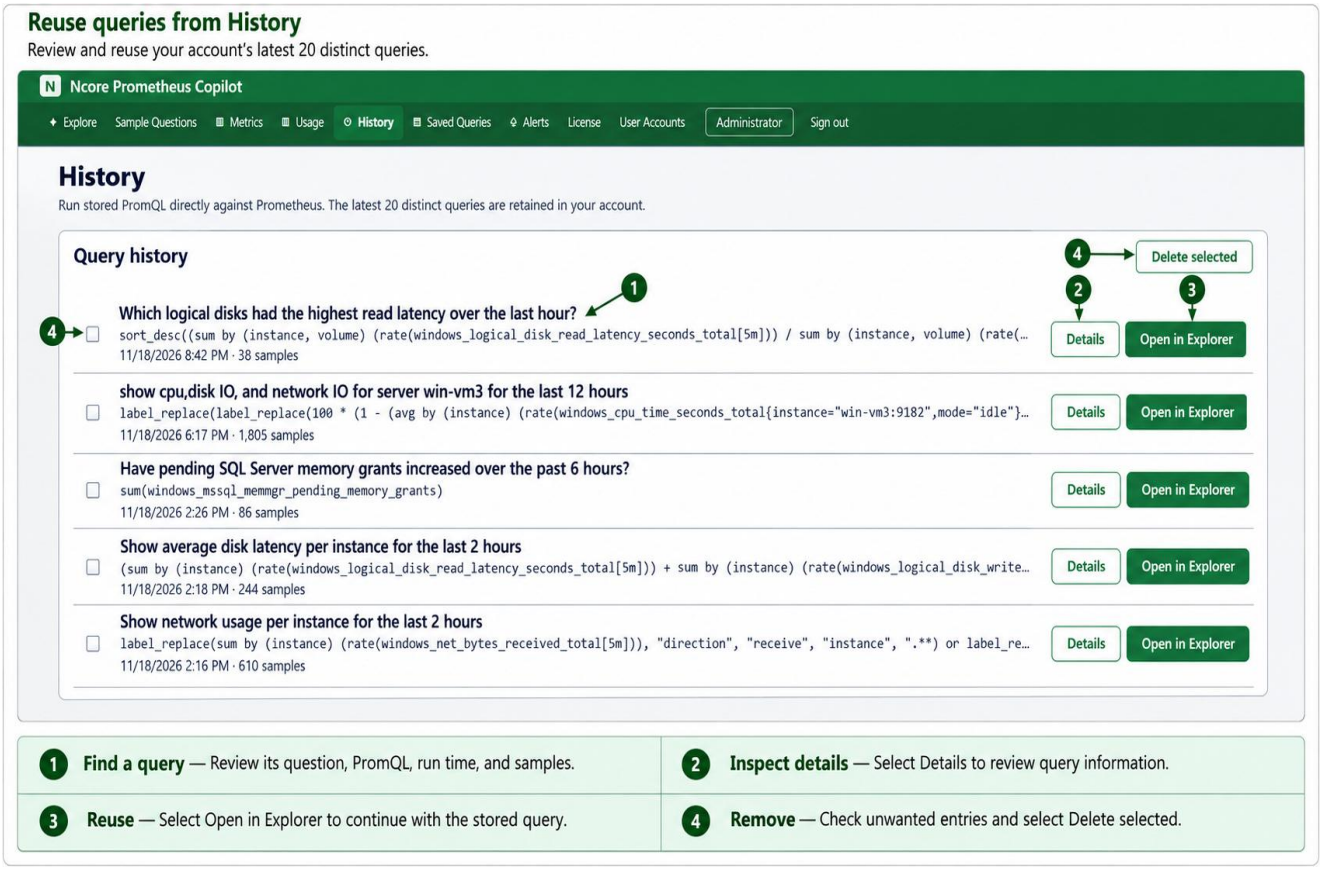
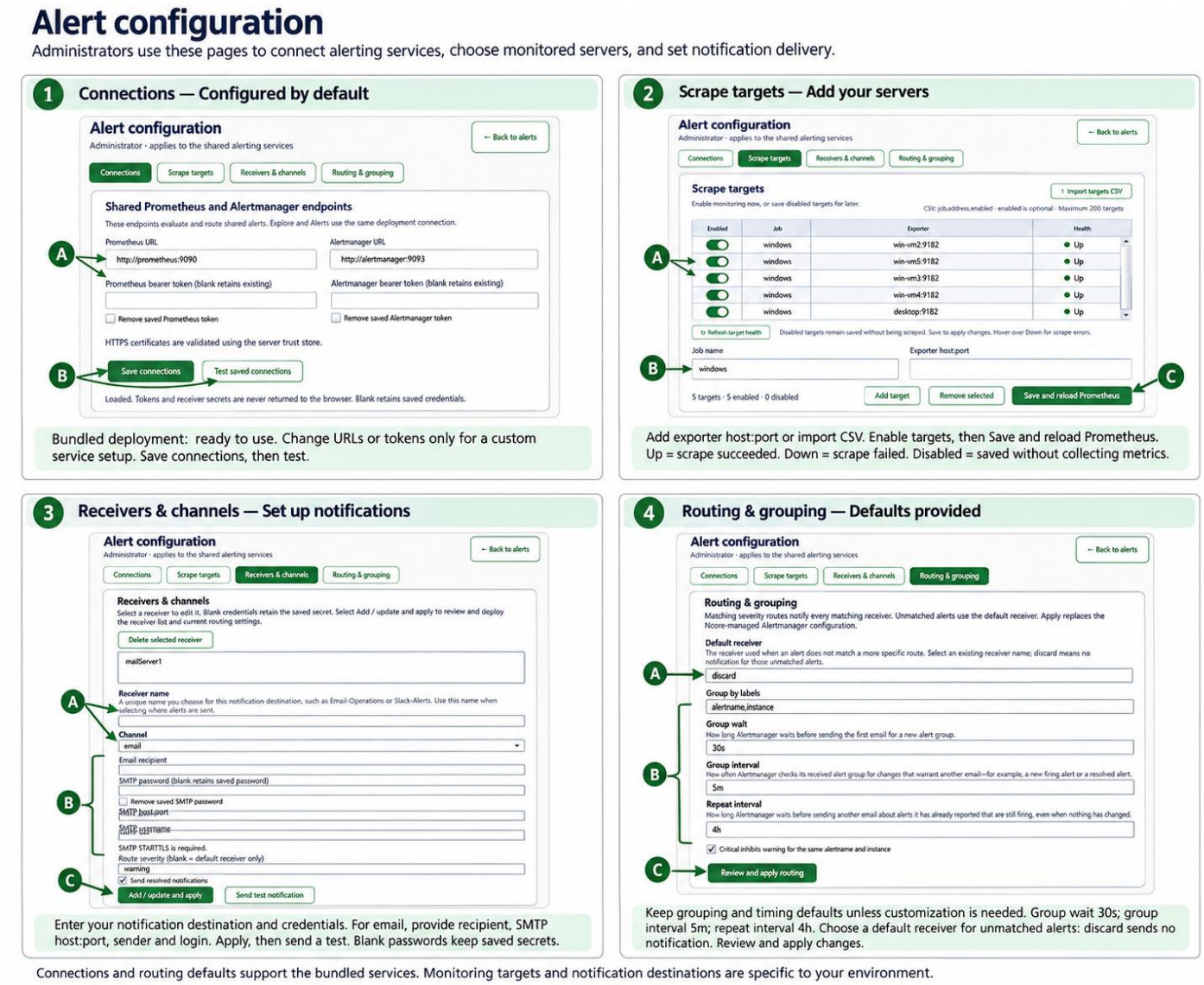


Figure 6 Query history and available actions

Select the checkboxes and choose Delete selected to remove entries you no longer need. Use Saved Queries to retain queries for repeated use.

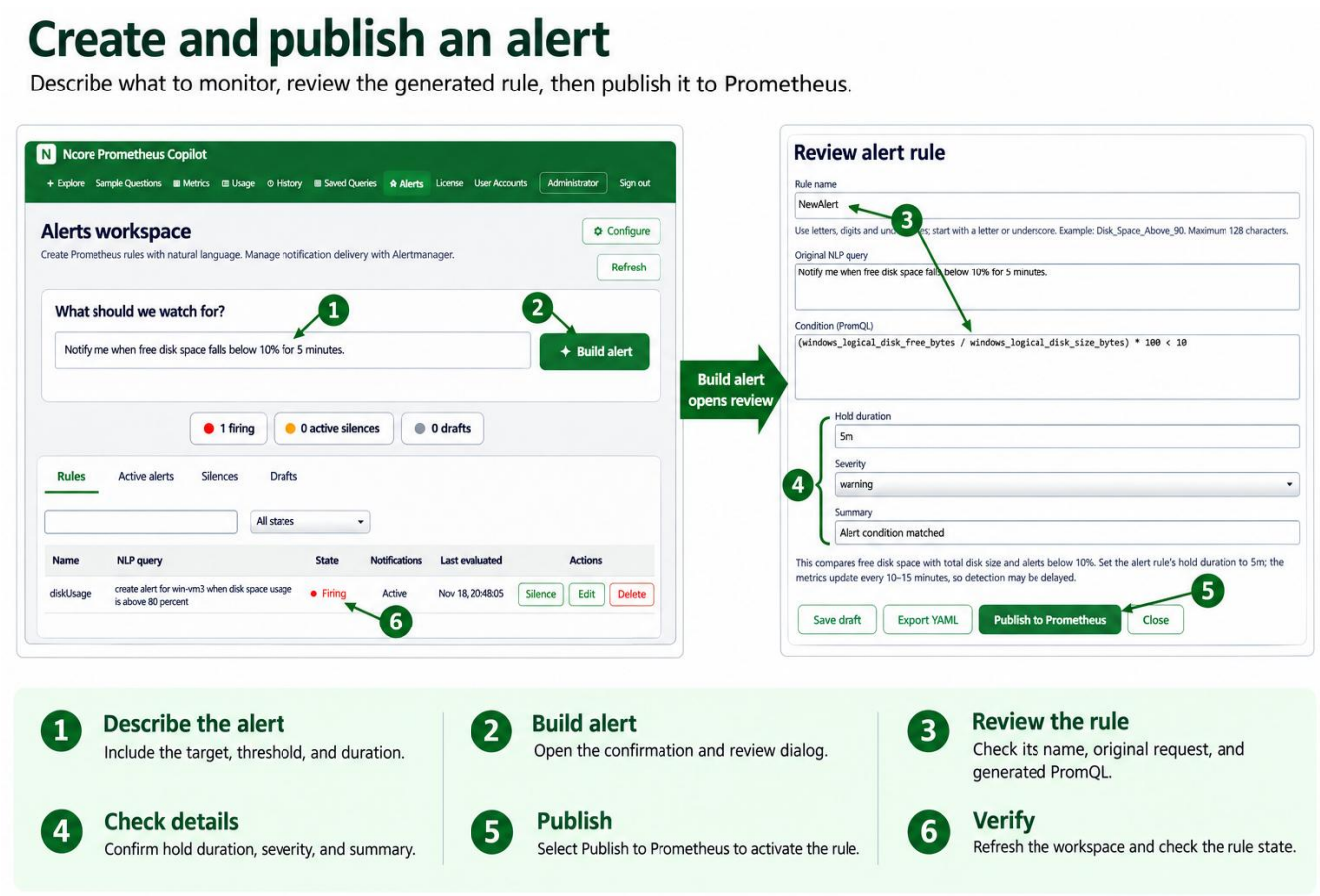
9. Alert configuration

Connections and standard routing values are preconfigured for the bundled deployment; change them only when customization is needed. Add scrape targets for your environment and configure a receiver to deliver notifications. A discard default receiver does not send notifications for unmatched alerts.



10. Create and publish an alert

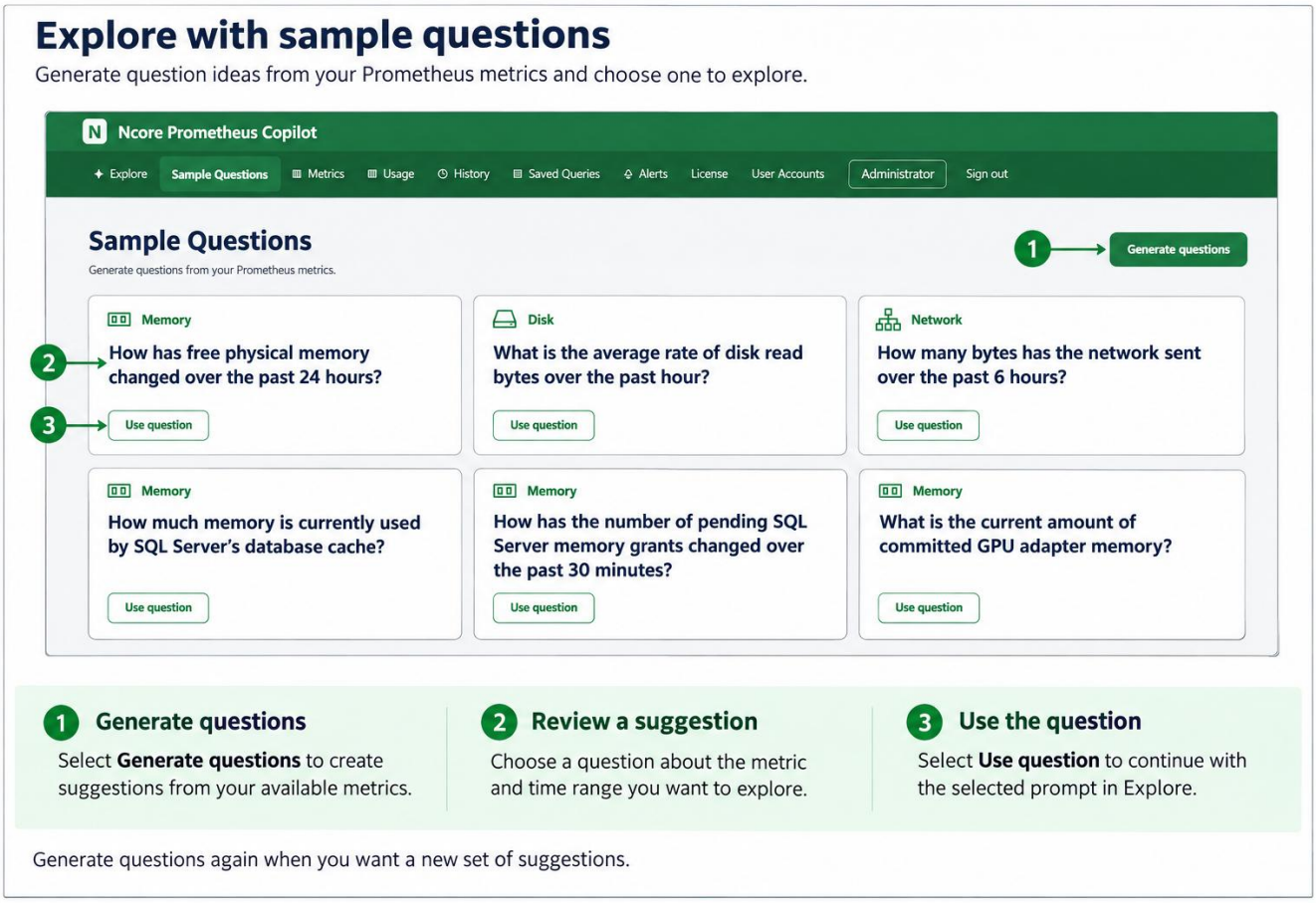
Enter an alert request, select Build alert, then review the generated rule, condition, hold duration, severity, and summary. Publish to Prometheus when ready, or save a draft for later.



Other review actions: Save draft keeps the rule for later. Export YAML downloads the rule. Close dismisses the dialog.

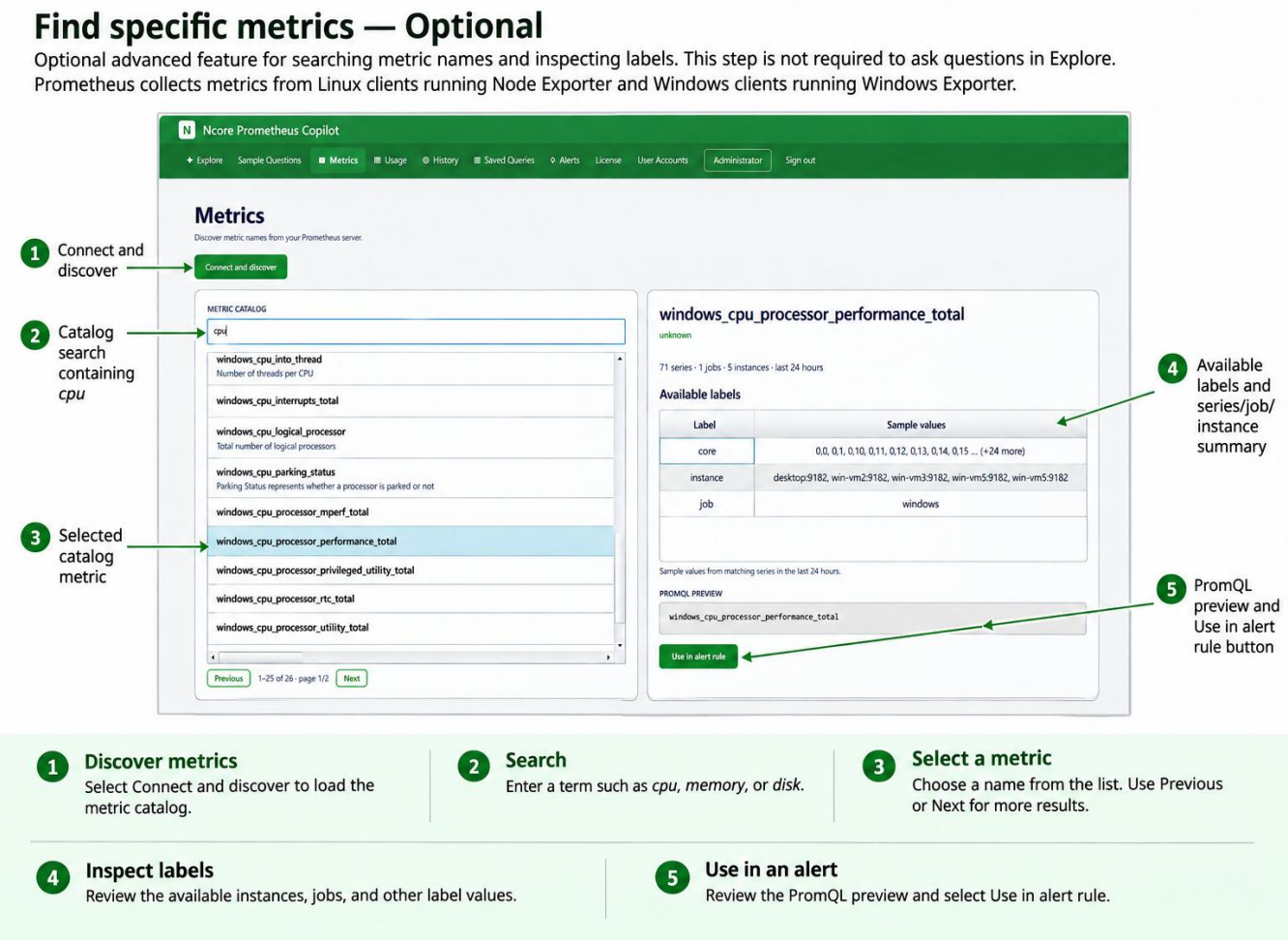
11. Sample Questions

Use this feature for AI generate example questions based on the metrics available in Prometheus. Select Use question to continue in Explore and review or refine the request.



12. Search metrics optional

Optional advanced feature. Discover and search specific metric names, inspect labels, and use a metric in an alert rule. Linux clients typically expose host metrics through node_exporter; Windows clients use windows_exporter. This page is not required for natural-language queries.



13. AI Usage

Open Usage to review AI activity, including total requests, input tokens, output tokens, average latency, and the details of individual operations. Choose a period or specific day where available, then refresh the view.

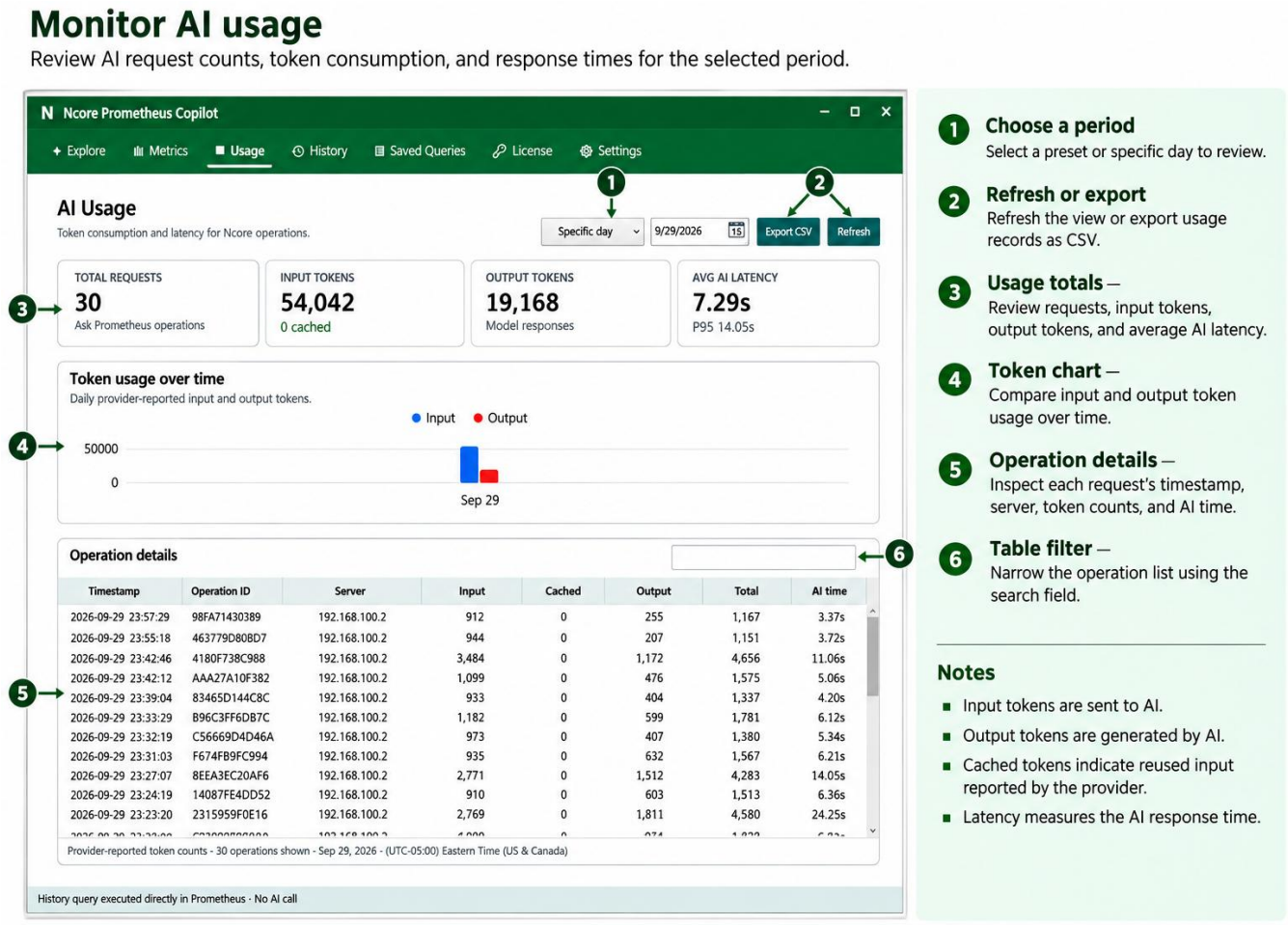


Figure 8 AI Usage and operation details

Input tokens include the question, application instructions, and metric context needed to generate PromQL. Output tokens reflect the provider's returned response. The size of the final PromQL expression alone does not represent the full request.

Reuse Saved Queries or History when you already have the PromQL you need. Direct Prometheus execution avoids an additional AI generation request.

14. License and TLS settings

Open License to review activation status. Select Export request, obtain a signed license for this deployment, then choose Choose signed license and import. Select Refresh license to verify the updated status. The TLS certificate section separately enables HTTPS.

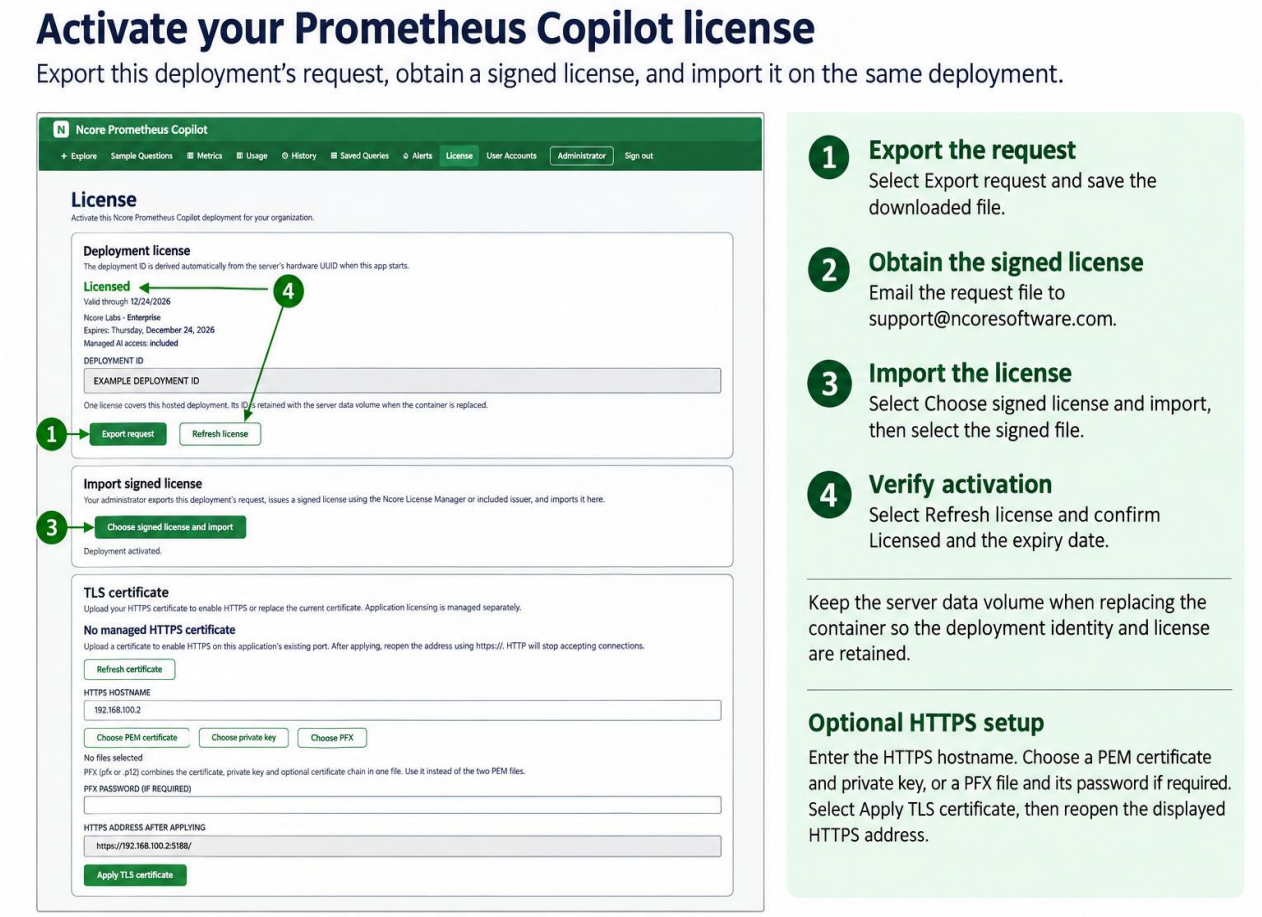


Figure 9 Product License and request actions

15. How to write good questions

Describe the metric, target, time range, and comparison you want in ordinary language. You do not need to write PromQL.

Specify the measure. “Show average non-idle CPU usage per instance for the last 24 hours”

Name the target. “Show memory usage for host1 over the last two days.” Use instance labels that actually exist in your Prometheus server.

Describe the comparison. “Show the five CPU cores with the highest average usage over the last 24 hours.” Specify whether you mean an average or a peak.

State the interval. “Plot CPU usage at five-minute intervals for the last six hours.” The chart interval differs from the PromQL rate window.

Check the chart units, time range, and returned series. If the result is unexpected, refine the wording or inspect Data and Query details. Requests outside the server’s retention cannot retrieve missing historical samples.

16. Security notes

AI prompts and data privacy. Ncore Prometheus Copilot does not send metric data or query results to AI. Only your question and selected metric metadata needed to generate PromQL are shared with the AI provider. The generated query runs against your Prometheus server, and its results are returned directly to the application for display. Avoid including secrets or sensitive information in your question.

Data access. Use only authorized Prometheus endpoints and credentials. Restrict access to the application and exported results according to your organization’s policies.

17. Troubleshooting

Prometheus cannot connect. Check the server URL, port, authentication settings, and network access. Reconnect and run metric discovery.

No data is returned. Check that the metric and instance labels exist and that samples fall within the requested time range. Try a recent period within retention.

The answer shows unexpected series. Inspect the labels and grouping in Query details. Specify the instance, CPU mode, or aggregation more clearly in the question.

Prometheus contains both Windows and Linux metrics. When a Prometheus server collects metrics from both platforms, explicitly mention Windows or Linux in your question so the AI can select the appropriate metrics. For example: “For Windows, show CPU usage over the last 24 hours” or “For Linux, show memory usage over the last two hours.” Add the instance name when targeting a specific server.

AI generation fails. Check configured AI access and network connectivity. If a quota or authorization message appears, contact the administrator responsible for the deployment.

The chart appears crowded. Choose a more suitable chart type, narrow the series or time range, adjust Scale, or use Zoom & pan. Choose Reset zoom to restore the view, or Detach chart to open a separate chart window.

A saved query returns different values. Saved PromQL runs against the current Prometheus data for its configured period. Relative time ranges move forward; changing values do not mean the query was regenerated by AI.

License activation is unavailable. Check the status on License and confirm that the signed license matches this computer and product. Export a new request if your administrator needs to issue a replacement.

18. First Query in 60 Seconds

Start here once the application is installed, your license and Ncore managed AI access are active, and you have a working Prometheus URL. Connection time depends on your environment.

1. Open Settings. Choose the Prometheus connection page.

2. Connect to Prometheus. Enter the server URL and any required authentication settings.

3. Discover metrics. Choose Connect and discover, confirm the connected status, then choose Save connection.

4. Enter a question in Explore. Try “For Linux, show CPU usage over the last 24 hours.” Use Windows instead if you want Windows metrics.

5. Click Ask Prometheus. Wait for the generated query to execute and the chart to appear.

What you should see

- A chart of the returned CPU usage values.
- Timestamped results through Data beneath the chart.
- The generated PromQL through Query details · View PromQL.

If the query returns no data, try a shorter recent time range and confirm that the selected platform has CPU metrics. See Troubleshooting for connection, retention, and mixed-platform guidance.