Certainly! Here’s a detailed breakdown of the 5th step: \*\*Verify and Configure Prometheus and Grafana\*\*.

### \*\*5. Verify and Configure Prometheus and Grafana\*\*

#### \*\*a. Verify Prometheus\*\*

1. \*\*Access the Prometheus UI:\*\*

- Open your browser and navigate to `http://localhost:9090`.

2. \*\*Check Target Status:\*\*

- Go to \*\*Status\*\* > \*\*Targets\*\* in the Prometheus UI.

- You should see entries for `rabbitmq-master`, `rabbitmq-slave1`, and `rabbitmq-slave2` listed as targets.

- Ensure that their `last scrape` times are recent and their `last scrape` results show as `up`. This indicates that Prometheus is successfully scraping metrics from the RabbitMQ instances.

3. \*\*Query Metrics:\*\*

- Go to \*\*Graph\*\* or \*\*Console\*\* in the Prometheus UI.

- Use Prometheus queries to check the data being collected. For example, to see the number of messages in a queue, you can use a query like:

```promql

rabbitmq\_queue\_messages{queue="your\_queue\_name"}

```

- Replace `your\_queue\_name` with the actual name of the queue you want to monitor. Adjust queries based on the metrics you are interested in.

#### \*\*b. Verify Grafana\*\*

1. \*\*Access the Grafana UI:\*\*

- Open your browser and go to `http://localhost:3000`. The default username is `admin` and the default password is `admin`.

2. \*\*Add Prometheus as a Data Source:\*\*

- Log in to Grafana.

- Click on the gear icon (⚙️) in the left-hand menu to go to \*\*Configuration\*\*.

- Click on \*\*Data Sources\*\*.

- Click \*\*Add data source\*\*.

- Select \*\*Prometheus\*\* from the list of available data sources.

3. \*\*Configure Prometheus Data Source:\*\*

- \*\*URL:\*\* Set this to `http://prometheus:9090`. This URL is how Grafana will access Prometheus within the Docker network.

- \*\*Access:\*\* Choose \*\*Server (Default)\*\*.

- Click \*\*Save & Test\*\* to verify the connection. Grafana should indicate that it successfully connected to Prometheus.

4. \*\*Import or Create Dashboards:\*\*

- \*\*Import Dashboards:\*\*

- Go to the \*\*+\*\* icon in the left-hand menu and select \*\*Import\*\*.

- You can import dashboards from JSON files or by using a dashboard ID from the Grafana dashboard repository. For RabbitMQ, you can search for relevant dashboards on the Grafana website (e.g., search for "RabbitMQ").

- Enter the dashboard ID or upload the JSON file and click \*\*Import\*\*.

- Follow the prompts to configure the

imported dashboard, selecting the Prometheus data source you configured.

- \*\*Create Custom Dashboards:\*\*

- Go to the \*\*+\*\* icon in the left-hand menu and select \*\*Dashboard\*\*.

- Click \*\*Add new panel\*\*.

- In the panel editor, select \*\*Prometheus\*\* as the data source.

- Write Prometheus queries to display the metrics you want to visualize. For example, to create a graph showing the number of messages in a queue:

```promql

rabbitmq\_queue\_messages{queue="your\_queue\_name"}

```

- Customize the visualization options as needed (e.g., graph type, legend, etc.).

- Click \*\*Apply\*\* to save the panel to your dashboard.

5. \*\*Set Up Alerts (Optional):\*\*

- If you want to receive notifications based on specific metrics:

- Go to the panel where you want to set up alerts.

- Click on the \*\*Alert\*\* tab in the panel editor.

- Click \*\*Create Alert\*\*.

- Define alert conditions based on Prometheus queries (e.g., trigger an alert if the queue depth exceeds a certain threshold).

- Configure alert notifications (e.g., email, Slack) under \*\*Notification channels\*\*.

### \*\*Summary\*\*

- \*\*Prometheus:\*\* Ensure it is scraping metrics from RabbitMQ and that the data is accessible.

- \*\*Grafana:\*\* Add Prometheus as a data source, import or create dashboards to visualize RabbitMQ metrics, and optionally set up alerts for critical conditions.

With these steps, you should be able to effectively monitor RabbitMQ using Prometheus and Grafana. If you have further questions or need additional assistance, let me know!