# Asterisk Performance Analysis Document

## Introduction

The purpose of this document is to provide a comprehensive guide to monitor Asterisk's performance in real-time by analyzing resource usage. The tools used in this project include Grafana, Process-Exporter, and Prometheus for monitoring and visualization. SIPp is employed to generate SIP traffic, and support tools are used for additional traffic management. This document covers the setup process, tool configurations, expected outputs, and a final conclusion.

## Tools Overview

### Asterisk

A powerful open-source VoIP PBX system used as the core server for handling SIP/VoIP communications. Acts as the primary server whose performance is analyzed in this project.

### Prometheus

An open-source monitoring system that collects metrics from configured targets at given intervals. Provides a robust query language (PromQL) for retrieving and analyzing time-series data.

### Process-Exporter

A Prometheus exporter for exposing process-level resource usage metrics. Used to specifically track the resource usage of the Asterisk process.

### Grafana

A visualization tool that connects to Prometheus to display resource usage in the form of dashboards. Enables real-time analysis and monitoring of Asterisk's performance.

### SIPp

A traffic generator tool for SIP testing. Simulates various SIP call scenarios to stress-test Asterisk and generate measurable traffic.

### Support Tools

Additional utilities (e.g., network analyzers and SIP debugging tools) to validate and fine-tune the setup.

## Setup Instructions

This section details the step-by-step process to set up Prometheus, Process-Exporter, Grafana, and SIPp to monitor Asterisk's performance.

### Prerequisites

- Ubuntu 22.04 or a similar Linux distribution.  
- Basic knowledge of Linux commands.  
- Asterisk installed and configured.  
- Administrative privileges.

### 1. Install Prometheus

Update the system:

sudo apt update && sudo apt upgrade -y

Install Prometheus:

wget https://github.com/prometheus/prometheus/releases/download/v2.50.0/prometheus-2.50.0.linux-amd64.tar.gz  
tar -xvzf prometheus-2.50.0.linux-amd64.tar.gz  
sudo mv prometheus-2.50.0.linux-amd64 /usr/local/bin/prometheus

Create a Prometheus configuration file:

sudo nano /usr/local/bin/prometheus/prometheus.yml

Add the following content to the file:  
global:  
 scrape\_interval: 15s  
scrape\_configs:  
 - job\_name: 'process-exporter'  
 static\_configs:  
 - targets: ['localhost:9256']

Start Prometheus:

cd /usr/local/bin/prometheus  
./prometheus --config.file=prometheus.yml

### 2. Install Process-Exporter

Download and install Process-Exporter:

wget https://github.com/ncabatoff/process-exporter/releases/download/v0.7.8/process-exporter-0.7.8.linux-amd64.tar.gz  
tar -xvzf process-exporter-0.7.8.linux-amd64.tar.gz  
sudo mv process-exporter /usr/local/bin/

Create a configuration file for Process-Exporter:

sudo nano /etc/process-exporter.yml

Add the following content to the file:  
process\_names:  
 - name: "asterisk"  
 cmdline:  
 - "asterisk"

Run the Process-Exporter:

/usr/local/bin/process-exporter --config.path /etc/process-exporter.yml

### 3. Install Grafana

Add the Grafana repository:

sudo apt-get install -y software-properties-common  
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -  
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list  
sudo apt-get update

Install Grafana:

sudo apt-get install grafana -y

Start the Grafana service:

sudo systemctl start grafana-server  
sudo systemctl enable grafana-server

Access Grafana at http://localhost:3000 with default credentials (admin/admin).

### 4. Generate Traffic Using SIPp

Install SIPp:

sudo apt install sipp -y

Run a sample test to generate traffic:

sipp -sn uac <Asterisk\_IP> -r 10 -d 10000 -l 100  
-r: Call rate  
-d: Call duration  
-l: Concurrent calls

### 5. Configure Grafana Dashboard

1. Log in to Grafana.  
2. Add Prometheus as a data source:  
 - Navigate to Configuration > Data Sources > Add Data Source.  
 - Select Prometheus and set the URL to http://localhost:9090.  
3. Import or create a dashboard:  
 - Navigate to Dashboards > Import > Paste a dashboard JSON for process monitoring.  
 - Add panels to visualize metrics such as CPU, memory, and network usage.

## Outputs

- \*\*Prometheus\*\*: Time-series data for Asterisk’s process metrics such as CPU, memory, and network usage.  
- \*\*Grafana\*\*: Visual dashboards with real-time graphs and charts displaying resource utilization.  
- \*\*SIPp\*\*: Logs showing traffic generation statistics (call success rates, delays, and failures).

## Conclusion

This setup enables efficient, real-time monitoring of Asterisk's resource usage using Grafana, Prometheus, and Process-Exporter. By generating traffic with SIPp, the system is stress-tested, allowing users to analyze its performance under load. This provides valuable insights into optimizing Asterisk’s configuration and ensuring the stability of SIP/VoIP services.