**🔧 Step 1: Enable EPEL Repo**

Run this:

bash

CopyEdit

**sudo yum install epel-release -y**

This adds the required repo for msmtp.

**🔄 Step 2: Clean and Rebuild Cache**

bash

CopyEdit

**sudo yum clean all**

**sudo yum makecache**

**📦 Step 3: Install msmtp**

bash

CopyEdit

**sudo yum install msmtp -y**

✅ This should now work.

Let me know once msmtp is installed — then I’ll walk you through email testing and provide the **automated health check script with alerts**.

**You said:**

done

**ChatGPT said:**

Excellent! Since msmtp is now installed, you're ready to **send emails using Gmail SMTP** without dealing with mailx TLS errors.

**✅ Step-by-Step: Configure & Send Test Mail with msmtp**

**🛠 Step 1: Create msmtp Config File**

bash

CopyEdit

**sudo vi /etc/msmtprc**

Paste this (replace your details):

**ini**

**CopyEdit**

**defaults**

**auth on**

**tls on**

**tls\_trust\_file /etc/ssl/certs/ca-bundle.crt**

**logfile /var/log/msmtp.log**

**account gmail**

**host smtp.gmail.com**

**port 587**

**from your-email@gmail.com**

**user your-email@gmail.com**

**password your-app-password**

**account default : gmail**

🔁 Replace:

* your-email@gmail.com → your Gmail address
* your-app-password → 16-digit **Gmail App Password** (no spaces)

**🔐 Step 2: Secure the Config File**

bash

CopyEdit

**sudo chmod 600 /etc/msmtprc**

**🧪 Step 3: Send a Test Email**

Run this:

bash

CopyEdit

**echo -e "Subject: MSMTP Test Mail\n\nThis is a test from CentOS using msmtp." | msmtp chandrakanth3103@gmail.com**

✅ You should receive the mail in your inbox **with no errors**.

**🎯 Optional: Use mail Command with msmtp**

Link msmtp as the default sendmail:

bash

CopyEdit

**sudo alternatives --install /usr/sbin/sendmail sendmail /usr/bin/msmtp 1**

Now mail command (e.g., echo "msg" | mail -s "subj" ...) will also work with msmtp.

**🚀 Ready for the Health Check Script?**

Now that mailing works, I’ll give you a **real-world Bash script** that:

**🔍 Linux Health Check (with Email Alerts):**

* ✅ CPU usage
* ✅ Memory usage
* ✅ Disk usage
* ✅ Services status (like nginx, docker, etc.)
* 📧 Sends daily/alert mail using msmtp

Would you like:

1. 📅 Daily full report at fixed time (via cron)
2. 🚨 Alert-only email (only when thresholds are exceeded)
3. 🔁 Both

Let me know your preference and I’ll give you the script with cron setup!

**You said:**

yes now its working. please provide script

**ChatGPT said:**

Awesome! Since email is working now via msmtp, here’s a **complete real-time Linux Health Check Bash Script** that:

* 📊 Checks CPU, memory, disk
* 🛡 Monitors key services
* 📧 Sends a mail **report every day at 8 AM**
* 🕐 Set up with cron

**✅ Step 1: Bash Script — health\_check.sh**

Create the script file:

bash

CopyEdit

**sudo vi /usr/local/bin/health\_check.sh**

Paste this:

bash

CopyEdit

**#!/bin/bash**

**# ====== Health Check Variables ======**

**HOSTNAME=$(hostname)**

**DATE=$(date +"%Y-%m-%d %H:%M:%S")**

**REPORT="/tmp/health\_report.txt"**

**EMAIL="chandrakanth3103@gmail.com"**

**SUBJECT="[$HOSTNAME] Health Report - $DATE"**

**# ====== Start Report ======**

**echo "📊 Health Report for $HOSTNAME - $DATE" > "$REPORT"**

**echo "=======================================" >> "$REPORT"**

**# ----- Uptime -----**

**echo -e "\n🕒 Uptime:" >> "$REPORT"**

**uptime >> "$REPORT"**

**# ----- CPU Usage -----**

**echo -e "\n🔥 CPU Load:" >> "$REPORT"**

**top -bn1 | grep "Cpu(s)" >> "$REPORT"**

**# ----- Memory -----**

**echo -e "\n💾 Memory Usage:" >> "$REPORT"**

**free -h >> "$REPORT"**

**# ----- Disk -----**

**echo -e "\n🗄 Disk Usage:" >> "$REPORT"**

**df -h --total >> "$REPORT"**

**# ----- Running Services -----**

**echo -e "\n🔧 Active Services:" >> "$REPORT"**

**for svc in sshd crond docker nginx; do**

**systemctl is-active --quiet "$svc" && echo "$svc is ✅ running" || echo "$svc is ❌ not running"**

**done >> "$REPORT"**

**# ----- Top Processes -----**

**echo -e "\n📈 Top 5 Memory-Hungry Processes:" >> "$REPORT"**

**ps aux --sort=-%mem | head -n 6 >> "$REPORT"**

**# ====== Send Email ======**

**cat "$REPORT" | msmtp "$EMAIL"**

**# Cleanup (optional)**

**# rm -f "$REPORT"**

Then make it executable:

bash

CopyEdit

**sudo chmod +x /usr/local/bin/health\_check.sh**

**✅ Step 2: Schedule via Cron (Daily at 8 AM)**

Edit cron:

bash

CopyEdit

**crontab -e**

Add this line at the bottom:

bash

CopyEdit

**0 8 \* \* \* /usr/local/bin/health\_check.sh**

This runs the health check every day at **8 AM IST** and sends the email.

**✅ Sample Output (Email you'll get)**

perl

CopyEdit

📊 Health Report for rhel-server - 2025-06-16 08:00:00

=======================================

🕒 Uptime:

14:55:23 up 12 days, 1:23, 1 user, load average: 0.05, 0.12, 0.09

🔥 CPU Load:

Cpu(s): 1.2%us, 0.4%sy, 0.0%ni, 98.0%id, 0.2%wa, 0.0%hi, 0.2%si, 0.0%st

💾 Memory Usage:

total used free shared buff/cache available

Mem: 1.9G 850M 520M 64M 530M 900M

🗄 Disk Usage:

Filesystem Size Used Avail Use% Mounted on

/dev/xvda1 20G 11G 8.5G 57% /

total 20G 11G 8.5G 57%

🔧 Active Services:

sshd is ✅ running

crond is ✅ running

docker is ❌ not running

nginx is ❌ not running

📈 Top 5 Memory-Hungry Processes:

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 1503 1.5 13.2 162848 26000 ? Ss 14:23 0:02 /usr/lib/firefox

...