

Sending Mail or Text Messages via the Linux Command Line

IMPORTANT CHANGES, UPDATED 1/10/2023 !!!

To setup your Allstar server to send mail and SMS messages you will need to establish an MTA (Mail Transfer Agent) to send your mail to. The easiest is GMAIL. If you use Google two step verification you will need to login to your GMAIL account security settings to setup a device password.

First you will need to download the SSMTP package.

pacman -Sy ssmtp

We will setup ssmtp below but for info on ssmtp go to - <https://wiki.archlinux.org/index.php/SSMTP>

SETUP Gmail Account

This document assumes you are using gmail as your MTA. If you don't already have an account go to gmail.com and set one up. The username could be your ham call but it must be at least 6 characters in length so if you have a 5 letter ham call or less you will need to add characters, like your name or whatever. Usernames can contain letters (a-z), numbers (0-9), and periods (.). So w3xx.joe or joe.w3xx would be valid.

Now setup 2 factor authentication. This is important for the security of your account. It will text your phone with a code or use some other verification method. This is only done one time on a trusted device.

Now add a new device to your gmail account and get a password for it. Go to -



<https://myaccount.google.com/apppasswords>

Sign in to your account using username and password. You will then get this screen -

← App passwords

App passwords let you sign in to your Google Account from apps on devices that don't support 2-Step Verification. You'll only need to enter it once so you don't need to remember it. [Learn more](#)

Your app passwords

Name	Created	Last used	
Allstar	Jan 10	12:26 AM	
ESP8266	Jan 9	Jan 10	

Select the app and device you want to generate the app password for.

Select app ▼ Select device ▼

GENERATE

This is just an example of what the App Passwords screen will look like. In your case it will show your devices if you have setup any. Ignore the ones shown, they are mine.

Select the device drop-down and then select OTHER then give it a name – like Allstar or whatever you want. Then click the lower right GENERATE button once and you will get a password. You will get a security alert email from Gmail which warns you that a password was setup. since it is you doing it, you can ignore it.

The password will be four groups of four lowercase characters like -

xxxx xxxx xxxx xxxx

Write this down for later reference!

You can close the page in your browser.

Now we will configure the ssmtp config file. It would be easier to rename the existing file and then create a new one copy pasted from the text below.

```
cd /etc/ssmtp
```

```
mv ssmtp.conf orig_ssmtp.conf
```

nano /etc/ssmtp/ssmtp.conf.

Here is an example file that works as of 1/10/2023. Copy and paste this into the new blank ssmtp.conf file you have opened.

```
#
# /etc/ssmtp.conf -- a config file for sSMTP sendmail.
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=USERNAME@gmail.com

# The place and port where the mail goes. The actual machine name is required
mailhub=smtp.gmail.com:587

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
# if you do not have a domain name for your IP connection use the following with NODE set to a
# valid Allstar node on your public IP. This will return the correct sender IP address assuming you have an
# active #node at the message origination site. You could also use an assigned DynDNS name here.
hostname=NODE.ip.hamvoip.org

AuthUser=USERNAME@gmail.com
AuthPass=xxxxxxxxxxxxxxxxxxxx
AuthMethod=LOGIN
UseSTARTTLS=yes
~
# end config
```

You must then change **USERNAME** to YOUR gmail username and the x's in AuthPass, four groups of four letters to the device generated password. You can enter it as the four separated groups but after checking you **MUST** remove the spaces between the groups as shown to make it one contiguous password.

Save the updated config file.

Once it is setup use these tests to see if it works by sending a message or text to yourself.

Example send messages -

At the Linux prompt type the following -

```
echo -n "Subject: test message\n\nTesting SSMTP" | sendmail -v user@domain
```

You could send this to your gmail account or anywhere else you get mail. This test method will show if the process completed or resulted in an error. The -v gives verbose output.

Here is what the output looks like on the command line when it is working using my config. Yours would obviously have your config info -

```
# echo -e "Subject: test\n\nTesting SSMTP" | sendmail -v wa3dsp@gmail.com
[<-] 220 smtp.gmail.com ESMTP br31-20020a05620a461f00b006fa2cc1b0fbsm8437334qkb.11 - gsmt
[->] EHLO 27225.ip.hamvoip.org
[<-] 250 SMTPUTF8
[->] STARTTLS
[<-] 220 2.0.0 Ready to start TLS
[->] EHLO 27225.ip.hamvoip.org
[<-] 250 SMTPUTF8
[->] AUTH LOGIN
[<-] 334 VXNlcm5hbWU6
[->] d2EzZHNwQGdtYWlsLmNvbQ==
[<-] 334 UGFzc3dvcmQ6
[<-] 235 2.7.0 Accepted
[->] MAIL FROM:<root@27225.ip.hamvoip.org>
[<-] 250 2.1.0 OK br31-20020a05620a461f00b006fa2cc1b0fbsm8437334qkb.11 - gsmt
[->] RCPT TO:<wa3dsp@gmail.com>
[<-] 250 2.1.5 OK br31-20020a05620a461f00b006fa2cc1b0fbsm8437334qkb.11 - gsmt
[->] DATA
[<-] 354 Go ahead br31-20020a05620a461f00b006fa2cc1b0fbsm8437334qkb.11 - gsmt
[->] Received: by 27225.ip.hamvoip.org (sSMTP sendmail emulation); Wed, 11 Jan 2023 01:09:05 -0500
[->] From: "root" <root@27225.ip.hamvoip.org>
[->] Date: Wed, 11 Jan 2023 01:09:05 -0500
[->] Subject: test
[->]
[->] Testing SSMTP
[->] .
[<-] 250 2.0.0 OK 1673417348 br31-20020a05620a461f00b006fa2cc1b0fbsm8437334qkb.11 - gsmt
[->] QUIT
[<-] 221 2.0.0 closing connection br31-20020a05620a461f00b006fa2cc1b0fbsm8437334qkb.11 - gsmt
```

Sending SMS text messages via email

Each provider has an email address to send text to their service. Here is a partial list of the most common carriers.

Carrier	SMS Gateway	MMS Gateway
AT&T	<u>@txt.att.net</u>	<u>@mms.att.net</u>
Boost Mobile	<u>@sms.myboostmobile.com</u>	<u>@myboostmobile.com</u>
Cricket Wireless	<u>@sms.cricketwireless.net</u>	<u>@mms.cricketwireless.net</u>
T-Mobile	<u>@tmomail.net</u>	<u>@tmomail.net</u>
UScellular	<u>@email.uscc.net</u>	<u>@mms.uscc.net</u>
Verizon	<u>@vtext.com</u>	<u>@vzwpix.com</u>

Many third party carriers use the main carriers like consumer cellular which uses AT&T.

Method to send complete message to SMS to number on the command line.

Here is an example sending to a fictitious number at AT&T using a different command line method to send.

```
# sendmail 2155551212@txt.att.net
Subject: This is a test
This is a test message
More lines if needed
Control D on a blank line to end
```

IMPOTANT! When you turn on mail on your server be aware that cron jobs send email to root when they are executed. This means that if you have something like AutoSky in a cron at a 4 minute interval you will get email every 4 minutes. Not good! The answer is to put -

MAILTO=""

as the first line in your cron file. This will stop all cron jobs from sending mail and littering up your inbox.

crontab -e

As the first line of file enter MAILTO=""

Save and exit.

Mail or SMS sending examples

Here is a real world example script to test the CPU temperature on a particular server and an SMS text message being sent if it is over temperature. Set the MAXTEMP, RECEIVER, and ALERT_INTERVAL variables and run this script in the background to check CPU temperature and send a text message at the interval specified if the maximum temperature is exceeded. This script could be adapted to other alerts or messages sent from Asterisk Allstar.

How this script works

The script checks the CPU temperature every 30 seconds and if it exceeds the maximum it sends a text message to the designated number. It then continues to check and if the condition still exists at the alert interval time (30 minutes in this example) it sends another text message and will continue to do so until the problem clears. When the temperature returns to the proper range it sends a temperature OK message and continues to check the CPU temperature every 30 seconds.

```
#!/bin/bash
#
# EXAMPLE Script to send email or SMS message if
# CPU temperature exceeds a maximum.
# The smtp package must be installed and configured

# WA3DSP 3/2019

# Maximum CPU Temperature in Degrees C
MAXTEMP="30"
# Destination SMS number and address – CHANGE THIS!
RECEIVER="2155551212@txt.att.net"
# How often to alert if temperature is exceeded - Default 30 Minutes
ALERT_INTERVAL="30m"

function send_mail_warning() {
echo -e "Subject: Warning CPU High Temp\nTo: $RECEIVER\n\nWarning CPU Temperature $CTEMP C at
server $HOSTNAME" | sendmail -t
}

function send_mail_OK() {
echo -e "Subject: CPU Temp OK\nTo: $RECEIVER\n\nCPU Temperature $CTEMP C at server
$HOSTNAME" | sendmail -t
}

while :
do
PTEMP=`cat /sys/class/thermal/thermal_zone0/temp`
CTEMP=`expr $PTEMP / 1000`
echo -n $CTEMP
echo -n "C / "
FTEMP=`expr 9 '*' $CTEMP / 5 + 32`
echo -n "$FTEMP"
echo "F"
if [ $CTEMP -gt $MAXTEMP ]
then
```

```

echo "Sending temperature warning message - CPU $CTEMP C"
send_mail_warning
while [ $CTEMP -gt $MAXTEMP ]
do
    sleep $ALERT_INTERVAL
    if [ $CTEMP -le $MAXTEMP ]
    then
        echo "CPU Temperature Normal $CTEMP C"
        send_mail_ok
        break
    else
        echo "Sending temperature warning message - CPU $CTEMP C"
        send_mail_warning
    fi
done
fi
sleep 30
done

```

Another useful script may be to advise you of an IP address change at a remote location. This script would be run at startup by the /etc/rc.local file and run continuously in the background.

```

#!/bin/bash

# Notify via email and/or SMS text if IP address changes
# WA3DSP 1/10/2023

# Destination SMS number and address ***** CHANGE THIS! *****
RECEIVER="2155551212@txt.att.net"

function send_mail_warning() {
echo -e "Subject: IP Address Change\nTo: $RECEIVER\n\nThe IP address has changed to $1 at server
$HOSTNAME" | sendmail -t
}

startIP=$(mypublicip.sh)

while :
do
    currentIP=$(mypublicip.sh)
    if [ "$currentIP" != "$startIP" ]
    then
        #send notification
        echo -e "\nSending notification\n"
        send_mail_warning $currentIP
        echo "Current IP = "$currentIP
        echo "Start IP = "$startIP
        echo -e "\nIP address changed"
        startIP=$currentIP
    fi
    sleep 10m
done

```

Multiple TO addresses can be sent in one notification. Just separate the addresses with a comma.

someone@domain1.com,someone@domain2.com,someone@domain3.com

You could send any information that is available to at your node like GPIO status, IP addresses, etc. This could be done at a specific time, interval, or based on an action happening..

I hope this helps to get you started in sending data from your Allstar node to your email and/or SMS text.

If you have any questions email me - wa3dsp@gmail.com