Sending Mail or Text Messages via the Linux Command Line

To setup your Allstar server to send mail and SMS messages you will need to establish an MTA 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 an additional non two step password.

Go to this site -

https://wiki.archlinux.org/index.php/SSMTP

You will need to download the SSMTP package.

pacman -Sy ssmtp

Follow the instructions on this page to setup the config file located in **/etc/ssmtp/ssmtp.conf**. You will need an email server. GMAIL is a good choice.

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

Example send messages -

- Testing:

```
# echo -n 'Subject: test message' | sendmail -v destination_user@destination_domain
```

- Send to voip.ms to SMS number setup on their site:

```
# echo -n 'Subject:2155551212' | sendmail -v sms@voip.ms
```

These test methods will show if the process completed or resulted in an error.

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

```
# 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
```

Each SMS provider has a different email address. Here is a list for most -

SMS provider emails -

https://20somethingfinance.com/how-to-send-text-messages-sms-via-email-for-free/

NOTE – Some providers block out going email so this may not work in all cases!

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. If 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 ssmtp 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
```