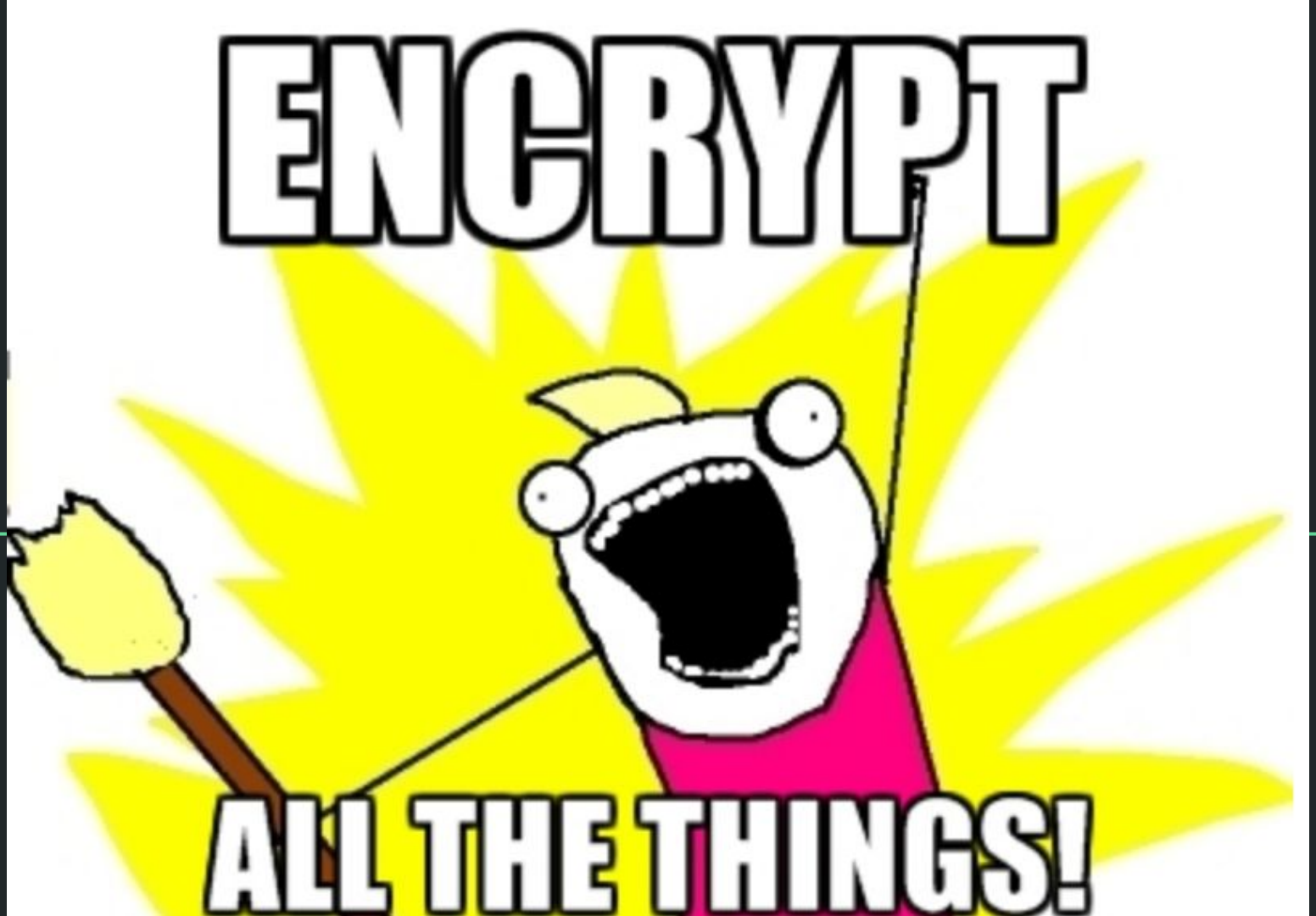
# Encrypt all the things

Andrew Denner
February 2018 Central Iowa Linux User's Group


ENCRYPT
ALL THE THINGS!

# Welcome to CIA LUG

Website: (http://cialug.org)

Email List: (see website)

IRC/Slack: (see website)

Video: Will be posted when it gets uploaded

Slides will be emailed after and at https://denner.co

# Little about me

Andrew Denner

Email: denner@gmail.com

Website: http://denner.co

Twitter: @adenner

Slides will be posted to https://denner.co

Even paranoid people have enemies

If you lose your private key (or password) you will loose data

# Encrypt your home directory (ubuntu)

# Why Encrypt?

- You have a laptop
- You deal with information that is sensitive (PII)
- Healthy Paranoia

# Why not encrypt?

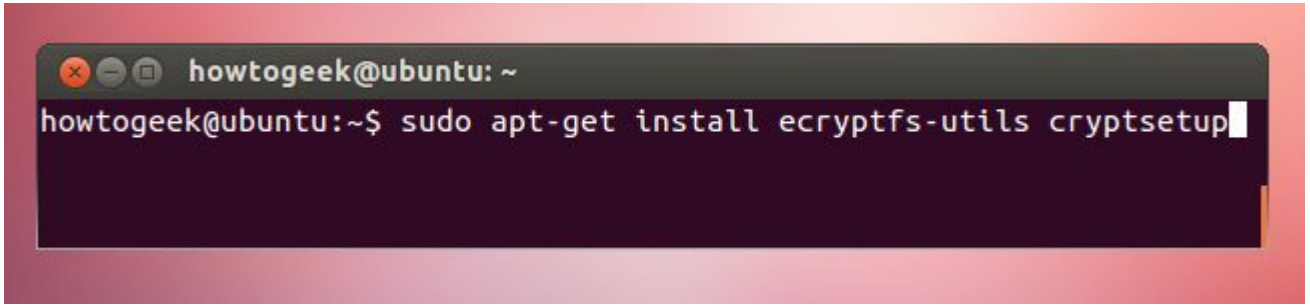
- You don’t care about your data
- Performance Hit
- Hard drive failure challenge
- Forget username and password you lose your data

# Easiest way to handle--From the start

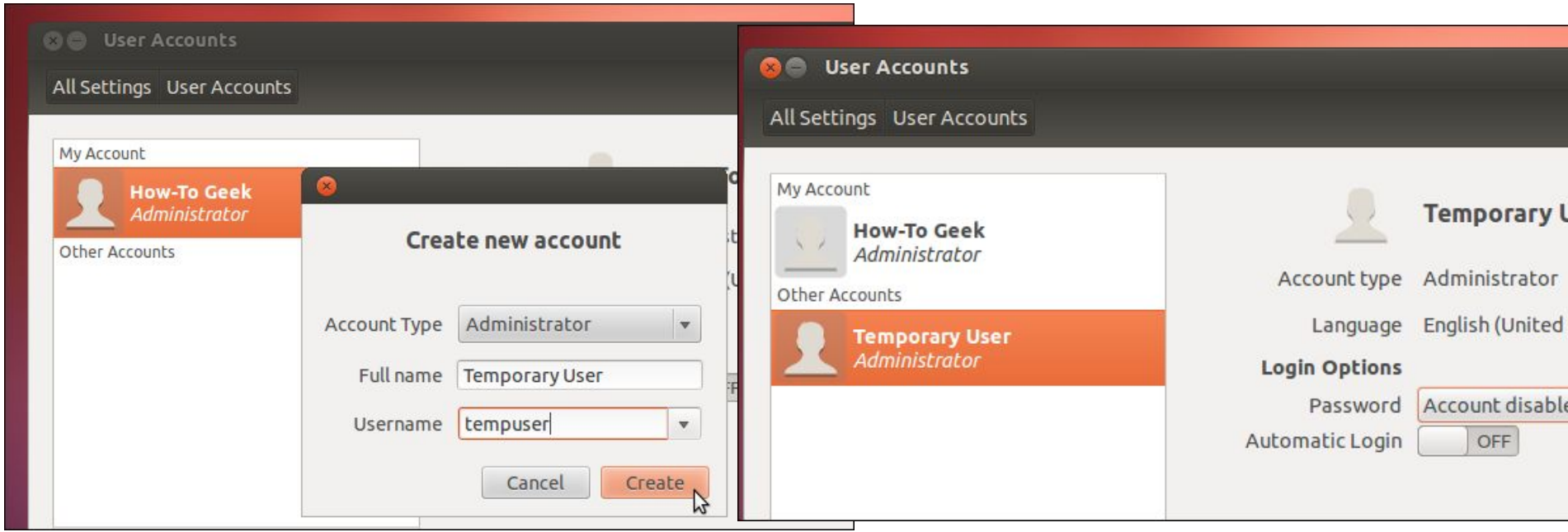
# Encrypt by hand...

Ubuntu uses eCryptfs (http://ecryptfs.org/about.html) think of it as PGP as a filesystem

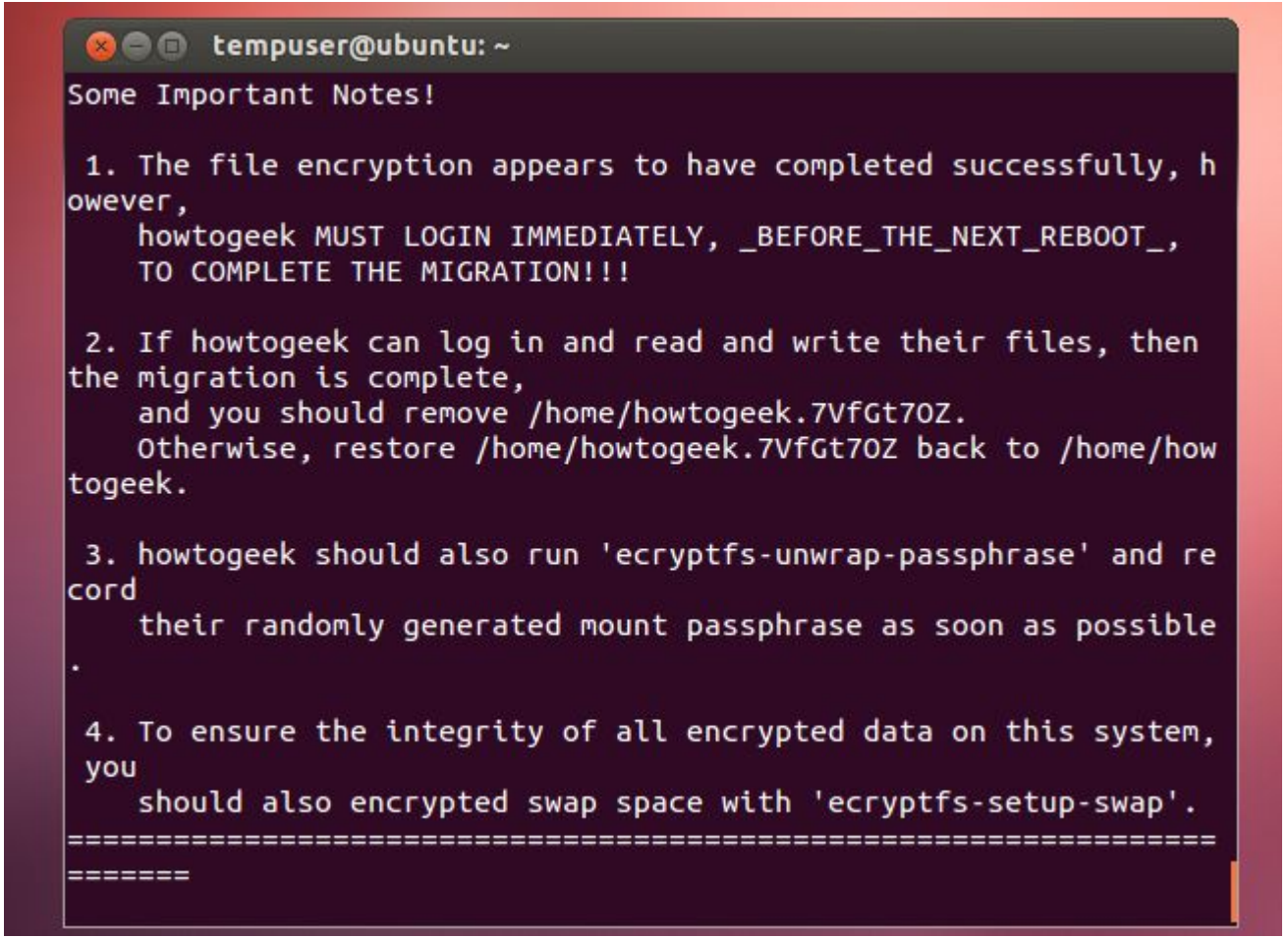
Screenshots are from https://www.howtogeek.com/116032/how-to-encrypt-your-home-folder-after-installing-ubuntu/

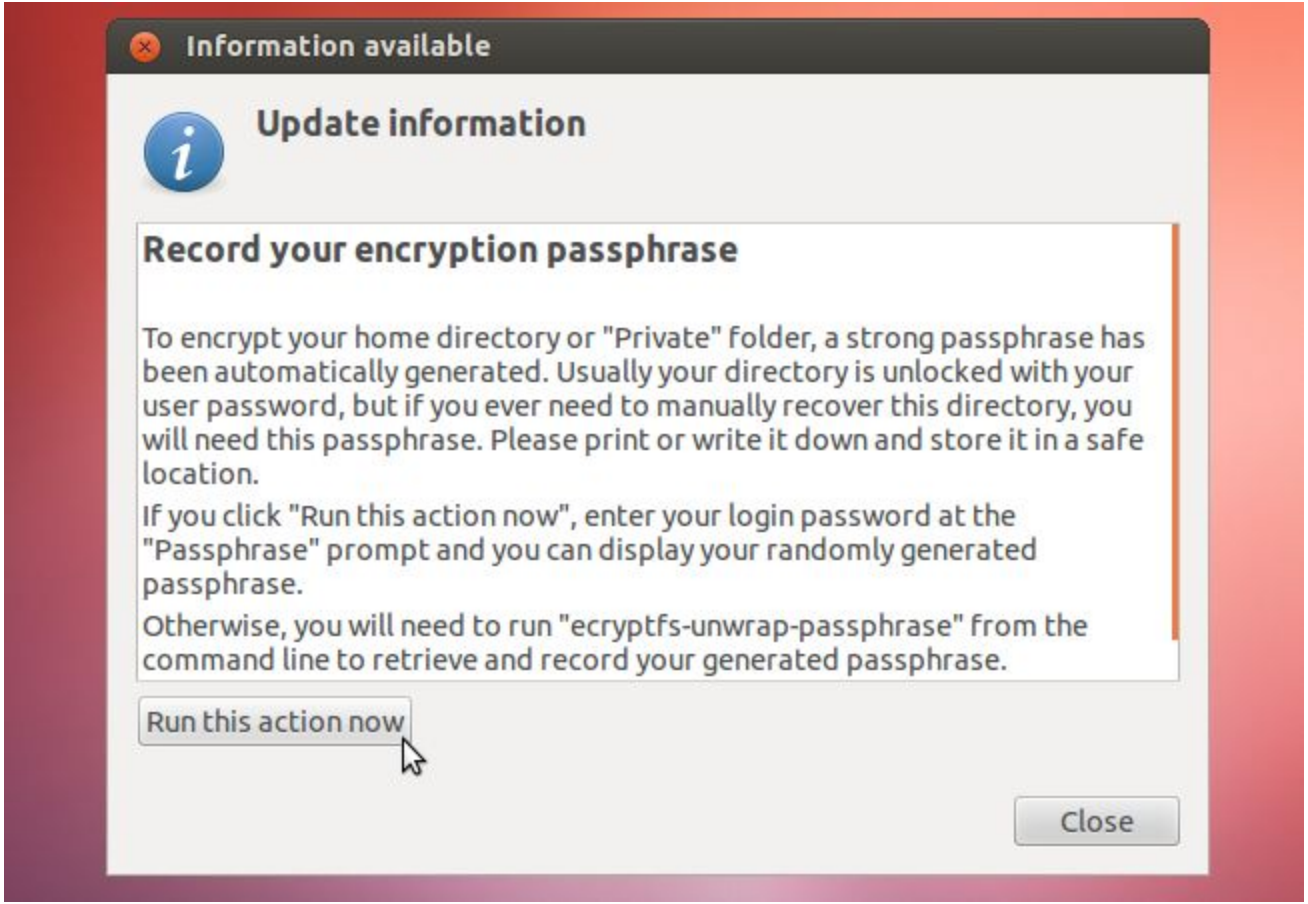


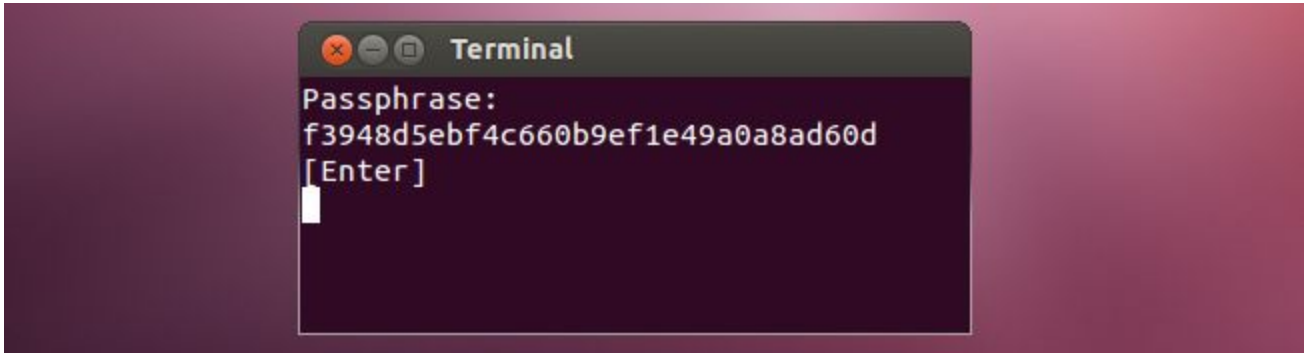
You cannot encrypt your account while you are logged in... you need to create a seperate account in the wheel group



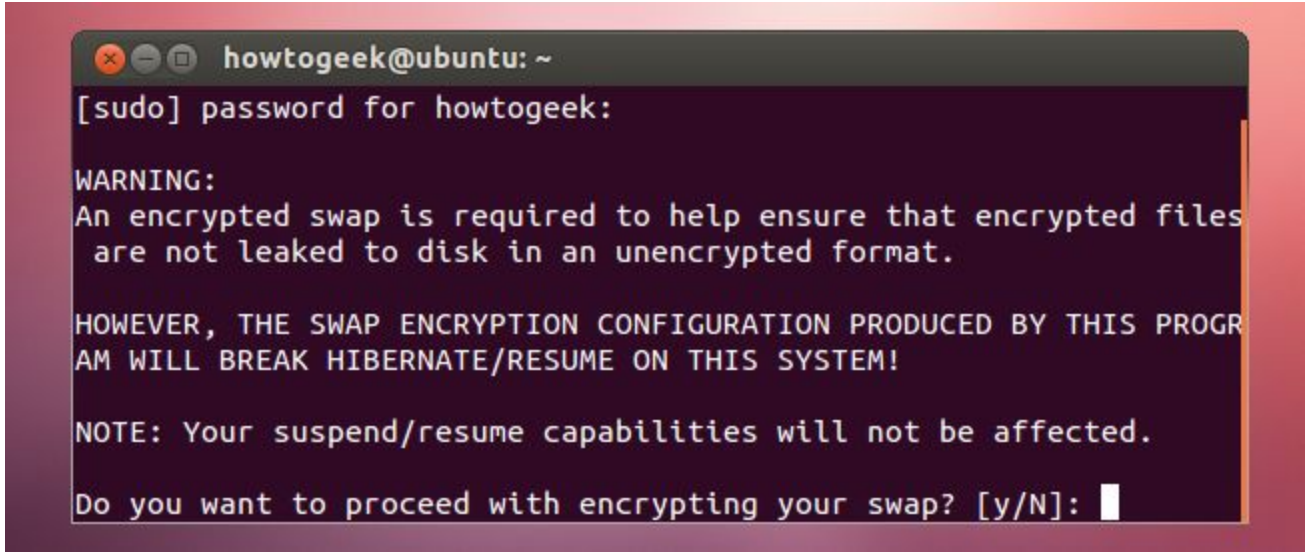
sudo ecryptfs-migrate-home -u user




Information available
Update information
Record your encryption passphrase
To encrypt your home directory or "Private" folder, a strong passphrase has been automatically generated. Usually your directory is unlocked with your user password, but if you ever need to manually recover this directory, you will need this passphrase. Please print or write it down and store it in a safe location.
If you click "Run this action now", enter your login password at the "Passphrase" prompt and you can display your randomly generated passphrase.
Otherwise, you will need to run "ecryptfs-unwrap-passphrase" from the command line to retrieve and record your generated passphrase.
Run this action now
Close


Terminal
Passphrase:
f3948d5ebf4c660b9ef1e49a0a8ad60d
[Enter]

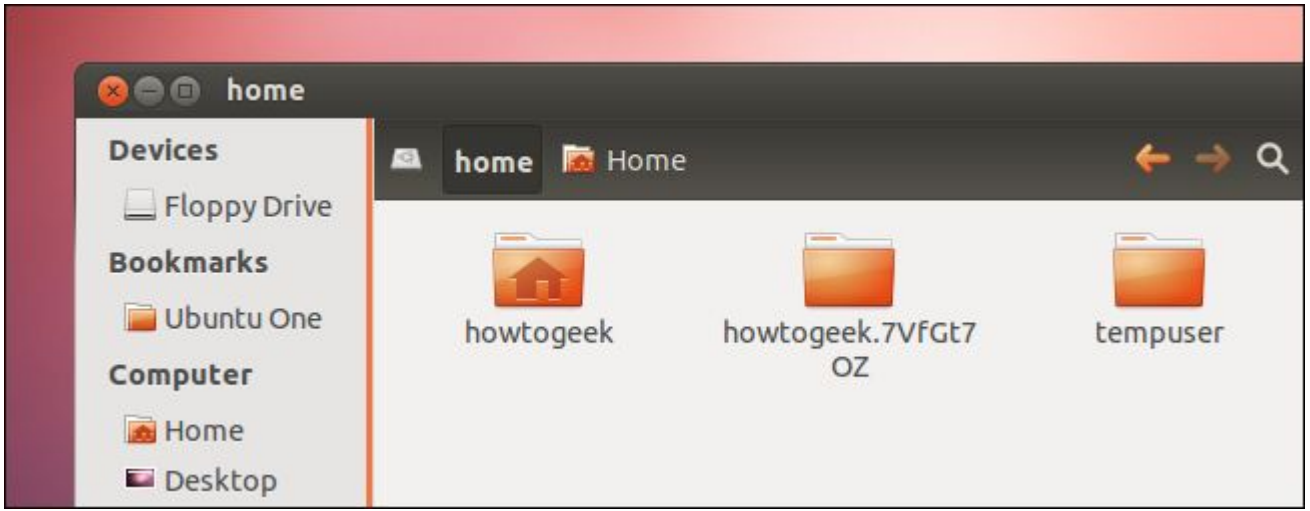
sudo ecryptfs-setup-swap



Note that an encrypted swap won’t work properly with Ubuntu’s hibernate feature

# cleanup

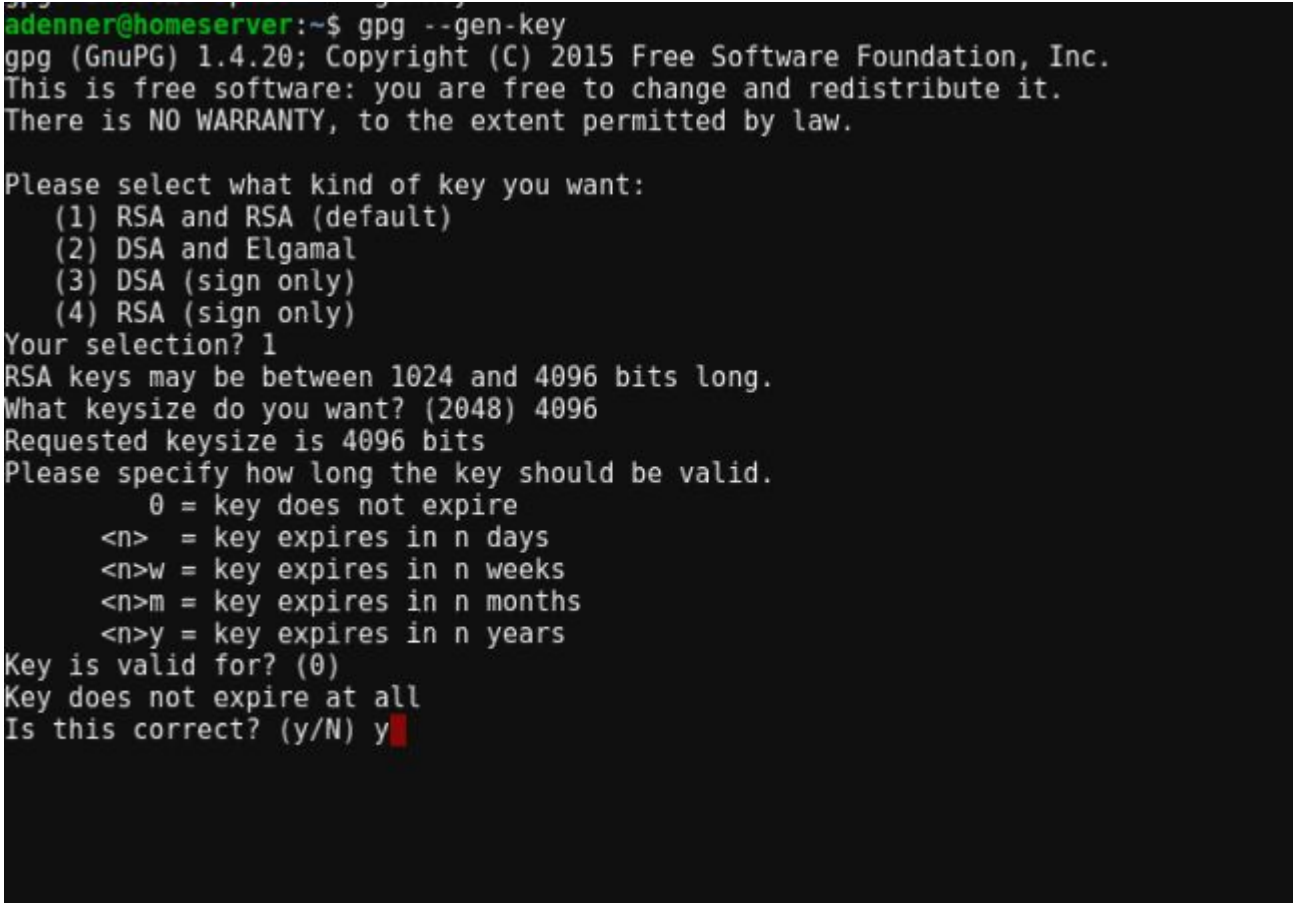
`sudo rm -rf /home/user.random`



# GNUPG

Swiss army knife of encryption... can do email, files, etc.

Dates back to Phil Zimmermann's PGP  (Prety Good Privacy)


adenner@homeserver:~$ gpg --gen-key
gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Andrew Denner
Email address: denner@gmail.com
Comment:
You selected this USER-ID:
    "Andrew Denner <denner@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

*We need to generate a lot of random bytes. It is a good idea to perform*
*some other action (type on the keyboard, move the mouse, utilize the*
*disks) during the prime generation; this gives the random number*
*generator a better chance to gain enough entropy.*
*++++++++++..+++++.+++++++++++++++.++++++++++...+++++++++++++++...++++++*
*+++++++++++++++++++++++++++.+++++..+++++.+++++.++++++++++++++++++++++++>.*
*++++++++++>+++++.....................................................+++++*
*Not enough random bytes available. Please do some other work to give*
*the OS a chance to collect more entropy! (Need 284 more bytes)*

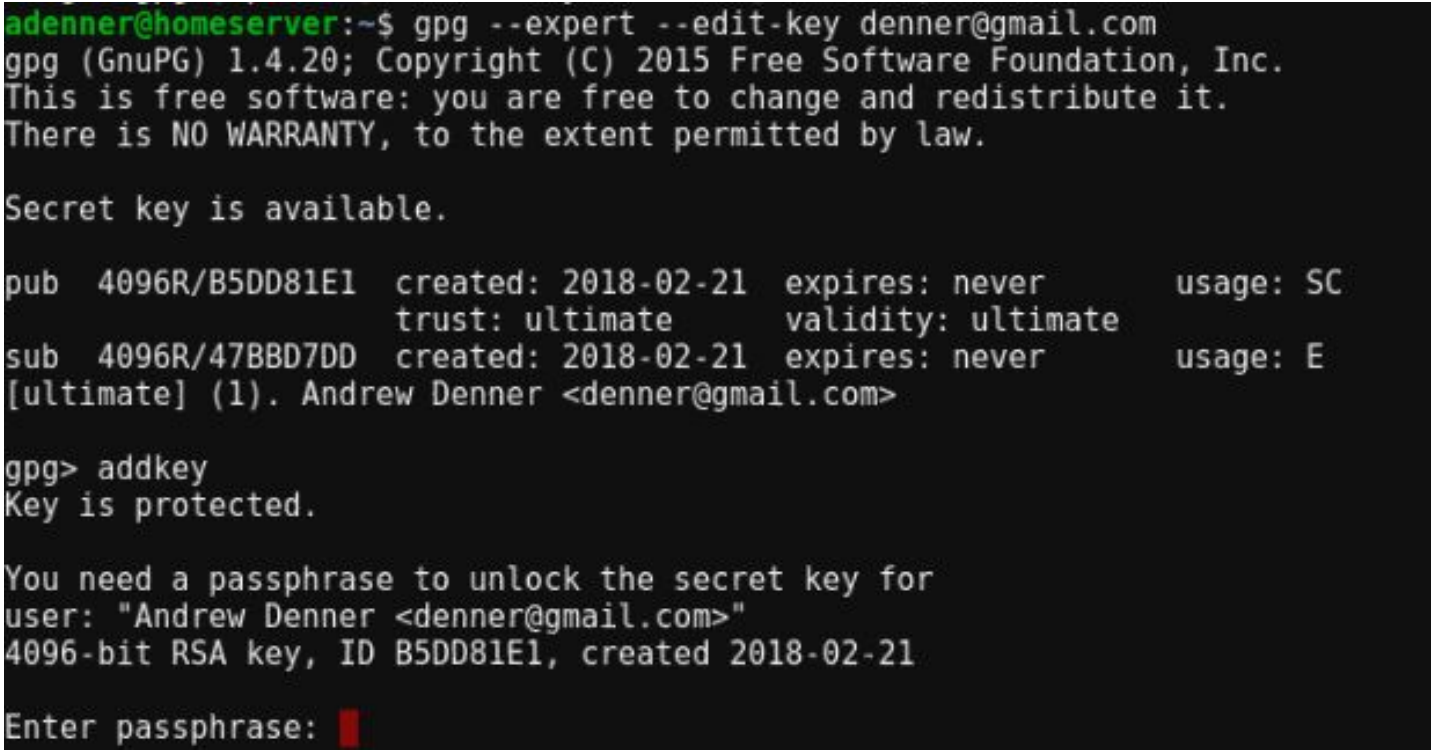
# Let’s make some random to fix the problem! (in a separate window)

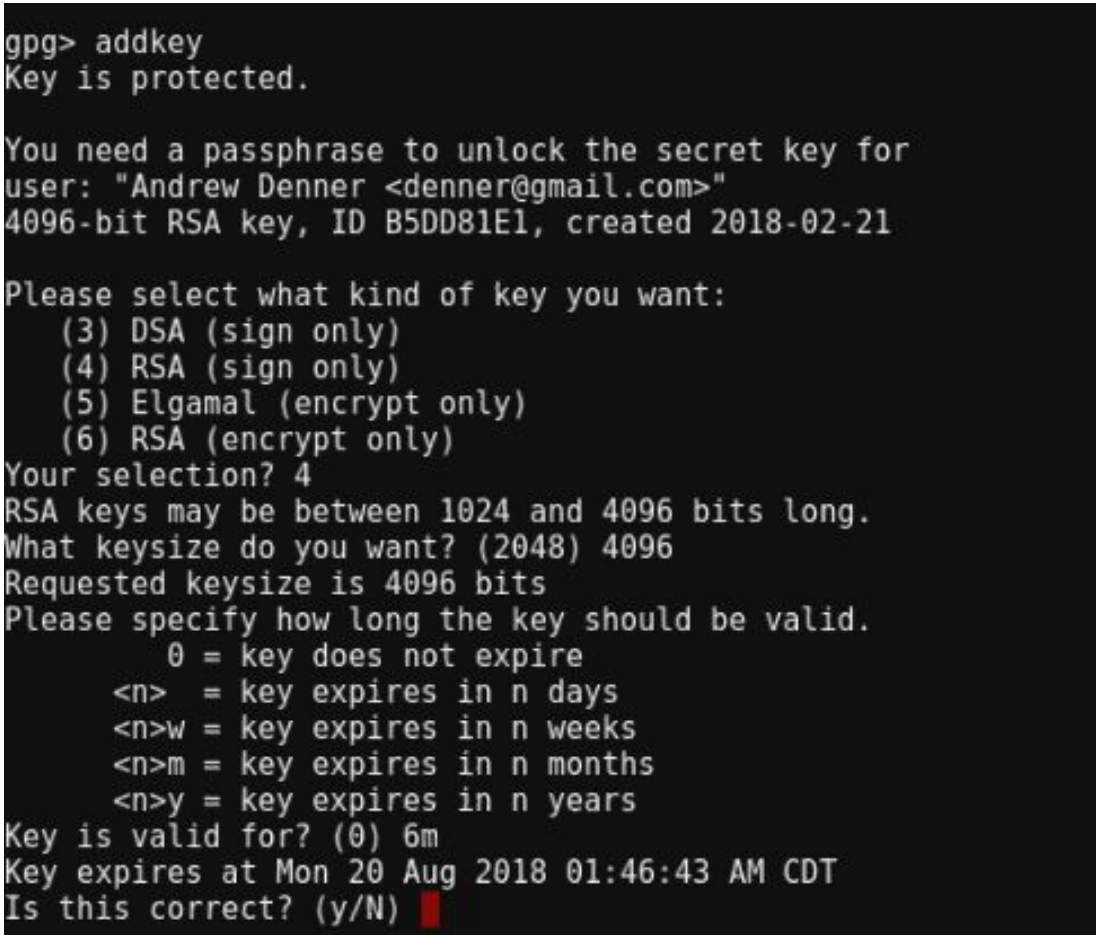
ls -l -R / would work as well

```
adenner@homeserver:~$ sudo ping -f 10.0.0.8
PING 10.0.0.8 (10.0.0.8) 56(84) bytes of data.
.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
.E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E....
..E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E...
....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E...
......E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E..
.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E
......E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....E.....
.....E.....^C
--- 10.0.0.8 ping statistics ---
2064 packets transmitted, 0 received, +1759 errors, 100% packet loss, time 504863ms
pipe 13
```

```
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.........................+++++
..+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
......+++++
..+++++
gpg: /home/adenner/.gnupg/trustdb.gpg: trustdb created
gpg: key B5DD81E1 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   4096R/B5DD81E1 2018-02-21
      Key fingerprint = ED31 5D8B 8EE5 AA9F 4505  1197 E981 8706 B5DD 81E1
uid                  Andrew Denner <denner@gmail.com>
sub   4096R/47BBD7DD 2018-02-21
```


adenner@homeserver:~$ gpg --expert --edit-key denner@gmail.com
gpg (GnuPG) 1.4.20; Copyright (C) 2015 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Secret key is available.
pub 4096R/B5DD81E1 created: 2018-02-21 expires: never usage: SC
trust: ultimate validity: ultimate
sub 4096R/47BBD7DD created: 2018-02-21 expires: never usage: E
[ultimate] (1). Andrew Denner <denner@gmail.com>
gpg> addkey
Key is protected.
You need a passphrase to unlock the secret key for
user: "Andrew Denner <denner@gmail.com>"
4096-bit RSA key, ID B5DD81E1, created 2018-02-21
Enter passphrase:


gpg> addkey
Key is protected.
You need a passphrase to unlock the secret key for
user: "Andrew Denner <denner@gmail.com>"
4096-bit RSA key, ID B5DD81E1, created 2018-02-21
Please select what kind of key you want:
(3) DSA (sign only)
(4) RSA (sign only)
(5) Elgamal (encrypt only)
(6) RSA (encrypt only)
Your selection? 4
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0) 6m
Key expires at Mon 20 Aug 2018 01:46:43 AM CDT
Is this correct? (y/N)

```
Is this correct? (y/N) y
Really create? (y/N) y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.............+++++
......+++++

pub  4096R/B5DD81E1  created: 2018-02-21  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  4096R/47BBD7DD  created: 2018-02-21  expires: never       usage: E
sub  4096R/0DFB2592  created: 2018-02-21  expires: 2018-08-20  usage: S
[ultimate] (1). Andrew Denner <denner@gmail.com>
```

`gpg -a --export-secret-key john.doe@example.com > secret_key`

# Generate revocation cert

```
gpg -a --gen-revoke john.doe@example.com > revocation_cert.gpg

sec  4096R/144A027B 2013-11-04 John Doe <john.doe@example.com>

Create a revocation certificate for this key? (y/N) y
Please select the reason for the revocation:
  0 = No reason specified
  1 = Key has been compromised
  2 = Key is superseded
  3 = Key is no longer used
  Q = Cancel
(Probably you want to select 1 here)
Your decision? 1
Enter an optional description; end it with an empty line:
>
Reason for revocation: Key has been compromised
(No description given)
Is this okay? (y/N) y

You need a passphrase to unlock the secret key for
user: "John Doe <john.doe@example.com>"
4096-bit RSA key, ID 144A027B, created 2013-11-04

Revocation certificate created.

Please move it to a medium which you can hide away; if Mallory gets
access to this certificate he can use it to make your key unusable.
It is smart to print this certificate and store it away, just in case
your media become unreadable.  But have some caution: The print system of
your machine might store the data and make it available to others!
$
```

gpg -a --export john.doe@example.com > public_key.gpg

```
denner@homeserver:~/example$ gpg --output doc.gpg --recipient denner@gmail.com --encrypt test_doc.txt
```

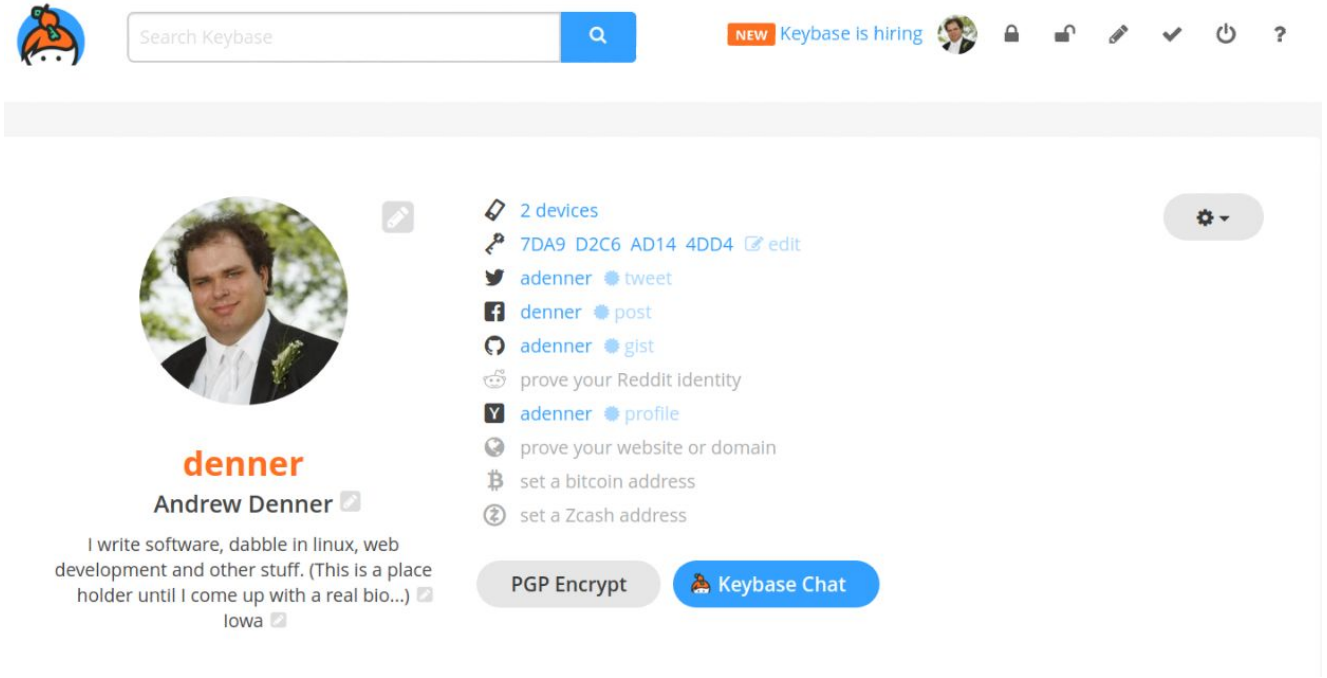
```
denner@homeserver:~/example$ gpg --output doc --decrypt doc.gpg

You need a passphrase to unlock the secret key for
user: "Andrew Denner <denner@gmail.com>"
4096-bit RSA key, ID 47BBD7DD, created 2018-02-21 (main key ID B5DD81E1)

gpg: encrypted with 4096-bit RSA key, ID 47BBD7DD, created 2018-02-21
      "Andrew Denner <denner@gmail.com>"
denner@homeserver:~/example$
```

# Web-based pgp

https://keybase.io/denner



# Let’s encrypt

```
Setting up python3-parsedatetime (2.4-3+ubuntu16.04.1+certbot+3) ...
Setting up python3-zope.hookable (4.0.4-4+ubuntu16.04.1+certbot+1) ...
Setting up python3-zope.interface (4.3.2-1+ubuntu16.04.1+certbot+1) ...
Setting up python3-zope.event (4.2.0-1) ...
Setting up python3-zope.component (4.3.0-1+ubuntu16.04.1+certbot+3) ...
Setting up python3-certbot (0.21.1-1+ubuntu16.04.1+certbot+0.2) ...
Setting up certbot (0.21.1-1+ubuntu16.04.1+certbot+0.2) ...
Setting up libaugeas0 (1.4.0-0ubuntu1.1) ...
Setting up python3-augeas (0.5.0-1+ubuntu16.04.1+certbot+1) ...
Setting up python3-certbot-apache (0.21.1-1+ubuntu16.04.1+certbot+1) ...
Setting up python-certbot-apache (0.21.1-1+ubuntu16.04.1+certbot+1) ...
Setting up python3-icu (1.9.2-2build1) ...
Setting up python3-pyasn1 (0.1.9-2+certbot~xenial+1) ...
Setting up rename (0.20-4) ...
update-alternatives: using /usr/bin/file-rename to provide /usr/bin/rename (rename) in auto mode
Setting up ssl-cert (1.0.37) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/D
ebconf/FrontEnd/Dialog.pm line 76.)
debconf: falling back to frontend: Readline
Processing triggers for libc-bin (2.23-0ubuntu10) ...
Processing triggers for systemd (229-4ubuntu21) ...
root@9b7b580195bf:/# certbot --aoache
usage:
  certbot [SUBCOMMAND] [options] [-d DOMAIN] [-d DOMAIN] ...

Certbot can obtain and install HTTPS/TLS/SSL certificates.  By default,
it will attempt to use a webserver both for obtaining and installing the
certificate.
certbot: error: unrecognized arguments: --aoache
root@9b7b580195bf:/# certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): denner+test@gmail.com

-------------------------------------------------------------------------------
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v01.api.letsencrypt.org/directory
-------------------------------------------------------------------------------
(A)gree/(C)ancel: A

-------------------------------------------------------------------------------
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about EFF and
our work to encrypt the web, protect its users and defend digital rights.
-------------------------------------------------------------------------------
(Y)es/(N)o: n
No names were found in your configuration files. Please enter in your domain
name(s) (comma and/or space separated)  (Enter 'c' to cancel): test.com
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for test.com
Enabled Apache rewrite module
Waiting for verification...
Cleaning up challenges
Failed authorization procedure. test.com (http-01): urn:acme:error:unauthorized :: The client lacks sufficient authorization :: Invalid response from http://test.com/.well-known/acme-challenge/xmg2fbDyPDSomOtDSlzLDwnJOUf0bwnlgjF03H6Foic: "<!DOCTYPE html PUBLIC "-//W3
TD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>"

IMPORTANT NOTES:
 - The following errors were reported by the server:

   Domain: test.com
   Type:   unauthorized
   Detail: Invalid response from
   http://test.com/.well-known/acme-challenge/xmg2fbDyPDSomOtDSlzLDwnJOUf0bwnlgjF03H6Foic:
   "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
   <html>"

   To fix these errors, please make sure that your domain name was
   entered correctly and the DNS A/AAAA record(s) for that domain
   contain(s) the right IP address.
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
root@9b7b580195bf:/#
```

# Resources

Lets encrypt https://letsencrypt.org/getting-started/

https://certbot.eff.org/

Gnupg https://www.gnupg.org/documentation/manuals/gnupg/

https://wiki.debian.org/Subkeys

https://encryptallthethings.net/