Mail-In-A-Box Administration Manual

TBD

License

This manual is licensed under the Creative Commons Zero v1.0 Universal license. Any contributions to the repository will be under this license.

Here is the license:

```
CC0 1.0 Universal
Statement of Purpose
The laws of most jurisdictions throughout the world automatically conferexclusive Copyright
and Related Rights (defined below) upon the creator andsubsequent owner(s) (each and all, an
"owner") of an original work of
authorship and/or a database (each, a "Work").
Certain owners wish to permanently relinquish those rights to a Work for thepurpose of
contributing to a commons of creative, cultural and scientificworks ("Commons") that the
public can reliably and without fear of laterclaims of infringement build upon, modify,
incorporate in other works, reuseand redistribute as freely as possible in any form
whatsoever and for anypurposes, including without limitation commercial purposes. These
owners may
contribute to the Commons to promote the ideal of a free culture and thefurther production
of creative, cultural and scientific works, or to gainreputation or greater distribution for
their Work in part through the use and
efforts of others.
For these and/or other purposes and motivations, and without any expectationof additional
consideration or compensation, the person associating CC0 with aWork (the "Affirmer"), to
the extent that he or she is an owner of Copyrightand Related Rights in the Work,
voluntarily elects to apply CC0 to the Workand publicly distribute the Work under its terms,
with knowledge of his or herCopyright and Related Rights in the Work and the meaning and
intended legal
effect of CC0 on those rights.
1. Copyright and Related Rights. A Work made available under CC0 may beprotected by
```

```
copyright and related or neighboring rights ("Copyright andRelated Rights"). Copyright and
Related Rights include, but are not limited
to, the following:
  i. the right to reproduce, adapt, distribute, perform, display, communicate,  and
translate a Work;
  ii. moral rights retained by the original author(s) and/or performer(s);
  iii. publicity and privacy rights pertaining to a person's image or likeness  depicted in
a Work;
  iv. rights protecting against unfair competition in regards to a Work,  subject to the
limitations in paragraph 4(a), below;
  v. rights protecting the extraction, dissemination, use and reuse of data in
  a Work;
  vi. database rights (such as those arising under Directive 96/9/EC of the  European
Parliament and of the Council of 11 March 1996 on the legal  protection of databases, and
under any national implementation thereof,  including any amended or successor version of
such directive); and
  vii. other similar, equivalent or corresponding rights throughout the world  based on
applicable law or treaty, and any national implementations thereof.
2. Waiver. To the greatest extent permitted by, but not in contravention of,applicable law,
Affirmer hereby overtly, fully, permanently, irrevocably andunconditionally waives,
abandons, and surrenders all of Affirmer's Copyrightand Related Rights and associated claims
and causes of action, whether nowknown or unknown (including existing as well as future
claims and causes ofaction), in the Work (i) in all territories worldwide, (ii) for the
maximum
duration provided by applicable law or treaty (including future timeextensions), (iii) in
any current or future medium and for any number ofcopies, and (iv) for any purpose
whatsoever, including without limitationcommercial, advertising or promotional purposes (the
"Waiver"). Affirmer makesthe Waiver for the benefit of each member of the public at large
and to thedetriment of Affirmer's heirs and successors, fully intending that such Waiver
shall not be subject to revocation, rescission, cancellation, termination, orany other legal
or equitable action to disrupt the quiet enjoyment of the Workby the public as contemplated
by Affirmer's express Statement of Purpose.
3. Public License Fallback. Should any part of the Waiver for any reason bejudged legally
invalid or ineffective under applicable law, then the Waivershall be preserved to the
maximum extent permitted taking into accountAffirmer's express Statement of Purpose. In
addition, to the extent the Waiveris so judged Affirmer hereby grants to each affected
```

```
person a royalty-free,non transferable, non sublicensable, non exclusive, irrevocable and unconditional license to exercise Affirmer's Copyright and Related Rights inthe Work (i) in all territories worldwide, (ii) for the maximum durationprovided by applicable law or treaty (including future time extensions), (iii)in any current or future medium and for any number of copies, and (iv) for anypurpose whatsoever, including without limitation commercial, advertising orpromotional purposes (the "License"). The License shall be deemed effective as of the date CC0 was applied by Affirmer to the Work. Should any part of theLicense for any reason be judged legally invalid or ineffective underapplicable law, such partial invalidity or ineffectiveness shall notinvalidate the remainder of the License, and in such case Affirmer herebyaffirms that he or she will not (i) exercise any of his or her remaining Copyright and Related Rights in the Work or (ii) assert any associated claimsand causes of action with respect to the Work, in either case contrary toAffirmer's express Statement of Purpose.
4. Limitations and Disclaimers.
  a. No trademark or patent rights held by Affirmer are waived, abandoned,  surrendered, licensed or otherwise affected by this document.
  b. Affirmer offers the Work as-is and makes no representations or warranties  of any kind concerning the Work, express, implied, statutory or otherwise,  including without limitation warranties of title, merchantability, fitness  for a particular purpose, non infringement, or the absence of latent or  other defects, accuracy, or the present or absence of errors, whether or not
  discoverable, all to the greatest extent permissible under applicable law.
  c. Affirmer disclaims responsibility for clearing rights of other persons  that may apply to the Work or any use thereof, including without limitation  any person's Copyright and Related Rights in the Work. Further, Affirmer  disclaims responsibility for obtaining any necessary consents, permissions
  or other rights required for any use of the Work.
  d. Affirmer understands and acknowledges that Creative Commons is not a  party to this document and has no duty or obligation with respect to this
  CC0 or use of the Work.
For more information, please see<http://creativecommons.org/publicdomain/zero/1.0/>
```

About This Manual

Why?

The idea for this manual came about in conversation with @Eliter over on the Discourse Forums. Having an in-depth administration manual will help new-comers and maybe even veterans to Linux / MIAB to be able to setup and maintain their own mail server!

A lot of this manual was taken from mailinabox.email, and is used in accordance the the website license.

Attribution: https://mailinabox.email

What is Mail-In-A-Box?

About Mail-In-A-Box

Mail-in-a-Box lets you become your own mail service provider in a few easy steps. It’s sort of like making your own Gmail, but one you control from top to bottom.

Technically, Mail-in-a-Box turns a fresh cloud computer into a working mail server. But you don’t need to be a technology expert to set it up.

Development

Mail-in-a-Box is based on Ubuntu 18.04 LTS 64-bit and uses very-well-documented shell scripts and a Python management daemon to configure the system. Take a look at the system architecture diagram and security practices.

Development takes place on github at https://github.com/mail-in-a-box/mailinabox.

Note that the goals of this project are to . . .

- Make deploying a good mail server easy.
- Promote decentralization, innovation, and privacy on the web.
- Have automated, auditable, and idempotent system configuration.
- **Not** make a totally unhackable, NSA-proof server (but see our security practices).
- **Not** make something customizable by power users.

Additionally, this project has a Code of Conduct, which supersedes the goals above. Please review it when joining our community.

Joshua Tauberer (@JoshData) began this project in 2013 and is the primary maintainer.

Thank you to all of the contributors!

Why Build Mail-In-A-Box?

Mass electronic surveillance by governments revealed over the last several years has spurred a new movement to re-decentralize the web, a movement to empower individuals to be their own service providers again.

Although the core protocol of email, SMTP, is inherently decentralized, in practice email has become highly centralized because it is so damn difficult to implement the dozens of modern protocols that surround it. Mail-in-a-Box takes care of all of that, and no more.

This is important not just for privacy, but for the ability for the web to evolve and improve as it always has: through the ability of everyone to see how it works, tinker, and propose innovative changes.

Special Acknowledgements

This project was inspired in part by the "NSA-proof your email in 2 hours" blog post by Drew Crawford, Sovereign by Alex Payne, and conversations with @shevski, @konklone, and @GregElin.

Mail-in-a-Box is similar to iRedMail and Modoboa.

Mail-in-a-Box is based on Postfix, Dovecot, Z-Push, Roundcube, Nextcloud, Apache SpamAssassin, Postgrey, Nginx, @konklone’s nginx config, and more.

Mail-In-A-Box System Requirements

The following are a recommended minimum on system requirements.

- 25GB storage space or more
- 1GB available RAM
- Ubuntu 18.04 (18.04.1 is also compatible) *Note: MIAB will only install on 18.04, NOT 18.10.
- We recommend at least a dual core CPU, however it has been known to run on a single core.
- You will also need the ability to set a PTR or rDNS record with your ISP. This varies from company to company.
- The ability to *send and receive* on port 25. Some home ISP's will block this, but they might also allow it if you request it. Call them to find out.

Is Mail-In-A-Box right for me?

TBD

Finding a Hosting Provider & Setting Up a Domain

What is Hosting and a Domain?

What is a Server? Why do I need one?

MIAB requires a server to run. A server is a computer specially designed to run server applications, typically to be powered on all day, all month, all year, non-stop. However, when we say "server" in a hardware context, it does not necessarily have to be server-grade hardware, it can be any computer.

While you can run MIAB on your computer at home, there are some considerations you should take, as that may not be the best option for you. See the (WIP) section for more detail on choosing a hosting provider.

There is a business model, where companies host your servers for you, for a minutely, hourly, or monthly cost, depending on which provider. These companies put servers, on shelves, in aisles, in data centers (which are warehouses designed for servers). Of course, they provide power and Internet to your servers too. They also keep the building, server hardware, electricity and network maintained, with many backups and security measures in place to make sure that your server experiences the lowest possible downtime, at the most secure facility possible.

What is a Domain? Why do I need one?

In a computer network, every computer is assigned an IP address to communicate. These IP addresses look like the following: 8.8.8.8 (for an IPv4 address) and 2001:4860:4860::8888 (for an IPv6 address). Since those addresses are difficult to remember, DNS, or Domain Name System, is a system that converts words such as google.com (which is a domain) into an IP address.

Although you can technically setup a mail server without a domain, it will: make it difficult for users to remember your IP address, puts you in a position where you could be locked into a hosting provider, and MIAB, along with many other technologies, aren't really designed to work without a domain (or Fully Qualified Domain Name, or FQDN., it might sometimes be called). In short, you need a domain, or it will make it difficult to administrate your mailinabox server.

To get a domain, you must buy it from a domain name registrar. These

Terminology

Terminology within this manual may include "hosting company" or "hosting provider", which refers to the company that provides your server. These companies usually have an Internet Service Provider or "ISP", which provides access to the public network (commonly known as the "internet") to and from your server. However, for the most part, the hosting provider will usually work with their ISP to get what you need from their control panel.

Should I Self-Host?

There are many different hosting providers out there, with different kinds of services they offer, with many pros and cons to them, and things to consider. In this section, we'll discuss self-hosting, which may be beneficial to read, even if you plan to use a VPS (We will discuss VPS's further in the chapter), since it will also shed light on what hosting providers are responsible for, and what to look for.

Although there are many reasons not to, it is an option to host your own server at home. As stated in the *What is Hosting and a Domain?* section, a server is just a specially designed computer. In this section, we'll discuss self-hosting, in which you will be your own hosting provider. Please note, most residential Internet Service Providers frown upon hosting servers on their network, even writing things against doing so in their contracts at times.

Should I Self-Host?

Self-hosting should fit your needs. There are benefits and disadvantages of self-hosting. Self-hosting requires physical maintenance of servers, not just administrative maintenance, whereas if you pay a company to host your server, you would only have to worry about the administrative maintenance of the server.

Finding the Right Hardware

Do I Need Up Time?

Mail servers usually require a lot of up time, since mail is more likely to be lost if the server isn't online and waiting to accept it from another mail server. This could be detrimental to your personal life or business, if the server goes down when you or your employees need to send or receive emails. If users can wait while the server is down, and people emailing you have their servers setup to retry sending mail to you if it fails, then server down time may not be a problem for you. Keep in mind, that many servers sending mail to you are outside of your or mail-in-a-box's developer's control, and many servers are not configured well, so it is not safe to bet that they will retry sending mail to you.

Factors For Up Time

There are many factors that are involved that affect server up time. A server needs electricity, internet, powered on, and applications working to be considered "online". The physical environment you put the server in needs to stay at a temperature that is cool enough to operate, which may be a problem if you are hosting many servers in the same room, have poor ventilation, or it is hot outside. It is great to do further research on **best datacenter practices** to view more considerations, as this manual may not cover all of them.

Electricity

Electricity is also another concern. Despite the electricity being on most of the time (at least, in most of the United States), it sometimes goes out. It is always a good idea to have a plan for when the power goes out. Most datacenters have a power backup system that usually last between 24 and 48 hours, and probably longer if they have a backup generator and refuel it.

You can purchase an uninterruptible power supply (or UPS) to cope with these power outages. These UPS's plug into a wall and charge a battery, which you then plug a server into. When the power goes out, the UPS powers the server. However, the UPS only lasts a few minutes, so they are mostly intended to give the server an extra few minutes to properly shut down. When there is a power outage, a UPS will send a signal to the server, and the server decides what to do before the UPS battery drains. A problem to note with UPS's is that their batteries go bad after a while, so it is good to check on them every once in a while.

If having a UPS does not fit your needs, then you may need to find a solution that will power your server for longer, like a backup generator or bigger battery. We suggest that you do further research on having a backup power source, if up time and/or electrical outages are a concern.

It is also good to prepare for electrical storms, and have a surge protector. Please note that "power strip" and "surge protector" are two different things, as some power strips do not surge protect. If you have existing power strips, you can check if they surge protect by looking on the back. It is suggested that a good surge protector has the "UL" (Underwriter’s Laboratories) certification and at least meets the UL 1449 standards. Make sure that the energy absorption rating is at least 6-700 joules or higher (higher is better), and "clamping voltage" is 400V or lower (lower is better).[1] However, we are not electricians, so please do your own research. Warranties are also a good thing to have on your surge protector. If it fails to do its job, and you have a good warranty, you can use the money from the warranty to replace damaged equipment.

Having a good electrical ground is excellent too. However, that is an advanced electrical topic that you should cover with an electrician, as we are not experts in that field.

Network

Having a good network to have your server rely on is needed for server up time. As stated in the previous section, about electricity, you should also consider making sure your networking equipment stays powered on, has a power backup, and is surge protected.

You should consider both internal and external network (being your ISP) problems as well. Having a plan for when one ISP fails to provide is an excellent strategy. Most datacenters have redundant networks, so users don't have downtime when one network goes down.

There are companies that specialize in network redundancy, both internal and external. (WIP: more information needed) On your server, you can have multiple Network Interface Cards (NIC's), or multiple Ethernet ports and configure your server to fallback on one if the other fails.

It is also a good idea to make sure that your networking cables are properly installed. If you are hosting multiple physical servers, it is worthwhile to look into organizing the mess of cables. There are plenty of products you can find for **cable management** online. If your networking cables get into too much of a mess, you could experience server network downtime while you sift through the cable mess, if you ever need to unplug or rearrange things. Be sure to keep it neat, organized, and label which cable goes where. If your network is complex, it is a good idea to draw network diagrams of your network.

Be sure your network cables are reliable and won't fall out of port. Cables that do not have clamps are likely to fall out of the Ethernet port, which will of course lead to down time or high amounts of packets lost. High amounts of packets lost can lead to network disconnection or slower network speeds.

If you have network cables that are backed by metal that are difficult to bend and holds its shape, be sure to not bend them often. If you need to bend them often, purchase cables that are designed to be flexible and bend (WIP: what are these actually called?), as you may damage the cable.

Be sure your cables are sufficient for your needs. Various cables support different speeds at different lengths of cable. Below, is a table, which shows you the range (max length), and the max speed each cable supports. Please note, bits are not bytes, and 1 gbit/s (gigabit per second) is equal to 1,000 mbit/s (megabit per second). The chart below is from lowest quality cable to highest quality cable. CAT7 cable and better exists, but is expensive and higher quality than most users need. If you need longer distances, you can repeat the signal with a network router or network hub, or use a fiber optic cable; If you need higher speeds, use a fiber optic cable as well. However, fiber optic cables are expensive and go beyond what most users need.

Type:	range(ft)	range(meters)	100 mbit/s	1 gbit/s	10 gbit/s	MHz
CAT-5	328	100	X			100
CAT-5E	328	100	X	X		100
CAT-6	328	100	X	X		250
CAT-6	180	55	X	X	X	250
CAT-6A	328	100	X	X	X	500

Data Redundancy

It is a good idea to have data redundancy, because that is the most important part of a server--the data. While you can replace the software applications, any and all of the server hardware, you cannot replace lost data. While there are companies that specialize in data recovery, it costs a lot of money and there is no guarantee data recovery methods will work. The kind of data recovery we are referring to, is when you have a poor or failed data redundancy plan, and take your hard drives to a data recovery engineer, hoping that they can fix your issue.

Storage drives (both traditional mechanical and solid state drives) sometimes go bad. You can go the extra mile to by more expensive server-grade hard drives that have better manufacturer promises and guarantees, but they still can go bad. Sometimes drives just go bad, but drives can also be damaged from external forces. For mechanical drives, if there is vibration, the drive is dropped or smacked, heat, cold, or drastic changes in temperature, especially when powered on, can cause physical damage or corruption to the data. Both types of drives can be damaged by water, and other conditions that would damage a server as well, especially if the servers are exposed to people who might accidentally bump or mishandle equipment, or the building cannot protect the server from the elements.

Luckily, there are some technologies that, in a disaster, make it so you can throw away and replace equipment and still keep you servers and retain your data. Redundant Array of Independent Disks, or RAID, is a technology that spreads data across multiple drives, or, if you're more organized and complex, logical volumes. In this manual, we will use the term "volume", which can either means a physical drive, or another RAID setup, since you can layer RAID setups. However, if you prefer to have a simple setup, just mentally replace "volume" with "drive".

There are different types of RAID, which do different things. Below, we'll explain some of the RAID types. As a note, some of these RAID types have a minimum amount to start with. If the RAID type has redundancy, they also have a maximum fault tolerance, which is the maximum amount of volumes you can lose and still keep data; but if the RAID type does not have redundancy, the array must keep the same volumes at all times it is online, or will completely fail, and has zero fault tolerance. The minimum amount of volumes to start with is the amount you must have to initially configure the array. The minimum amount of volumes to start a RAID configuration does not mean that you cannot go below the minimum after the RAID configuration has started (WIP: find someone to explain this better!).

RAID 0, also called "disk striping", evenly splits writing and reading data across multiple volumes, without any redundancy. It requires at least two volumes. If you lose one volume in a 3 volume setup, you've lost all your data. However, this makes it a lot faster to read and write, since the workload is split evenly across volumes.

RAID 1 writes the same data to each volume, also called "mirroring", which means each volume is exactly the same. This has complete redundancy, and you can throw out and replace volumes without worry. This requires at least two volumes. This makes data reads faster, but costs more per-gigabyte.

RAID 5 (we didn't skip numbers), also called "Striping with parity", is a bit more complicated. It requires at least three volumes to run, and can lose up to one volume and still retain its data. It does this by spreading data across volumes, and adding a "parity" bit on the last volume, so the system can recalculate the data using math if one volume is lost. It does not matter what the "last" volume is, as the system automatically decides.

RAID 6, also called "Striping with double parity", is similar to RAID 5, in that it both spreads data across volumes and offers redundancy. However, RAID 6 offers more redundancy than RAID 5, as you can lose up to two volumes at once, and still retain data. It still uses the parity bit technology, but uses parity bits on the last two volumes.

RAID 10, also called "Striping + Mirroring", and is a combination of both RAID 0 and RAID 1, but we think you should do further research on the subject if you are interested in RAID 10.

The problem with this is that RAID 5 and RAID 6 is a theory that only works if all the volumes are online and synchronized properly. There is software available that automatically handles a lot of the hassle with dealing with RAID 5 and RAID 6. One of the problems we have not mentioned yet is that it takes a minimum of three drives to start either RAID configuration, and a maximum fault tolerance of one or two (depending on RAID type), but what happens when you lose a volume? The answer is that you should repair or replace the volume that went offline (because it was damaged or other reason) as soon as possible, while it is below the amount of volumes you started with.

Example configuration 1: Let us say that you have 5 drives in a RAID 5 configuration. Each drive holds a maximum of 100 gigabytes, but you are only using 200 gigabytes total from you RAID 5 configuration. As a note, you have 500 gigabytes of physical maximum storage, but only 400 gigabytes total are usable from the RAID 5 configuration, since one drive is taken up by the redundancy.

Example Scenario 1: Let's say you lose a drive, now you have a RAID 5 configuration that is missing a drive. How are you to know that you are missing a drive? You cannot replace a missing drive if you do not know about it. You need to find software that will notify you that a drive has failed in your RAID configuration. While the server is waiting for your to respond about your lost drive, you might want to have it configured to automatically make a few decisions. Firstly, how likely is it that another drive will fail by the time you repair or replace the drive? How important is it that there is redundancy between the time a drive goes offline and the time another drive goes online again to replace the lost drive? Is there enough storage space on the server to reconfigure the drives?

You can have the server make these decisions automatically, or manually, depending on the software you use to manage the array. Since the drives are in RAID 5, and one drive is lost, no more drives can be lost. The server has 400 gigabytes of maximum storage available, and 200 gigabytes of data it needs to keep, and 200 gigabytes of play room. It might be able to try to use the fact that there is 200 gigabytes of play room to move around data and reconfigure itself into a RAID 0 configuration or possibly RAID 1. When a drive comes online to the server, the software for managing the RAID setup also has some decisions to make or not make.

When a server is powered off, the RAID software needs to gracefully shutdown, or major data corruption may occur. RAID 5 and RAID 6 are very delicate in this matter, unlike RAID 1. If the server is in the middle of a write operation and is shut down, it is very possible that the entire array's data could be corrupted.

These are all things to anticipate when selecting software to manage your RAID setup (WIP: more on this!).

References

1. Henry, A., & Henry, A. (2013, September 27). How to Choose, Buy, and Safely Use a Good Surge Protector. Retrieved February 5, 2019, from https://lifehacker.com/how-to-choose-buy-and-safely-use-a-good-surge-protect-1405568999

Installing & Preparing Ubuntu for Mail-In-A-Box

Installing Ubuntu 18.04

> If you are using a VPS (Virtual Private Server) in the cloud, like Digital Ocean, OpenVZ, Nocix, Scaleway, etc then you do not need to follow this page. Please continue to the "Setting up Ubuntu 18.04 and Preparing for Mail-In-A-Box" page instead.

Before we do anything, we will need to download Ubuntu *Server* 18.04. Please do not use the Desktop version. It has packages that are not compatible with MIAB.

Here is a link to download Ubuntu Server 18.04 AMD64 Version

If you need arm64 version, please refer to the CDIMAGE server which has different architectures for different servers.

Once the download is completed, you will need to either burn the ISO to a DVD or USB stick. If you are using a Hypervisor to make a virtual machine (Xen, VMWare, VirtualBox, or Hyper-V to name a few) then you can continue without burning to a DVD or USB stick.

Lets boot up to the DVD, USB, or ISO and you will see a startup prompt. Select your preferred language and follow the on screen prompts. If this machine is not in the DMZ on your router or firewall please make sure to set the IP during the network setup as **manual** or **static**. This will allow you to port forward much easier later. Also make sure to select all drive space when prompted for partitions. The more space, the more emails and cloud hosting you can do.

During the installation you will be asked if you want to setup extra software. **DO NOT SELECT ANYTHING** and just continue. All software we need will be installed later. We do not want ubuntu automatically configuring anything that can conflict with MIAB.

Once the install is completed, remove any installation media and reboot the server. The next page will explain basic preparations and setup of ubuntu for MIAB.

Setting up Ubuntu 18.04 and Preparing for Mail-In-A-Box

In this part of the manual we will be setting up Ubuntu with SSH, secure passwords, networking (if not done during the install), and other stuff required for MIAB.

Configuring SSH

First thing we will need to do is install SSH, so let's get that out of the way right now.

```
sudo apt install openssh-server
```

After the installation is completed, please confirm you can connect by using PuTTY or some other SSH client to connect to the server.

Next, let's setup a public key authentication. If you are on Linux:

```
ssh-keygen
```

If you are on Windows, download and run puttygen (or install the PuTTY suite onto your computer.)

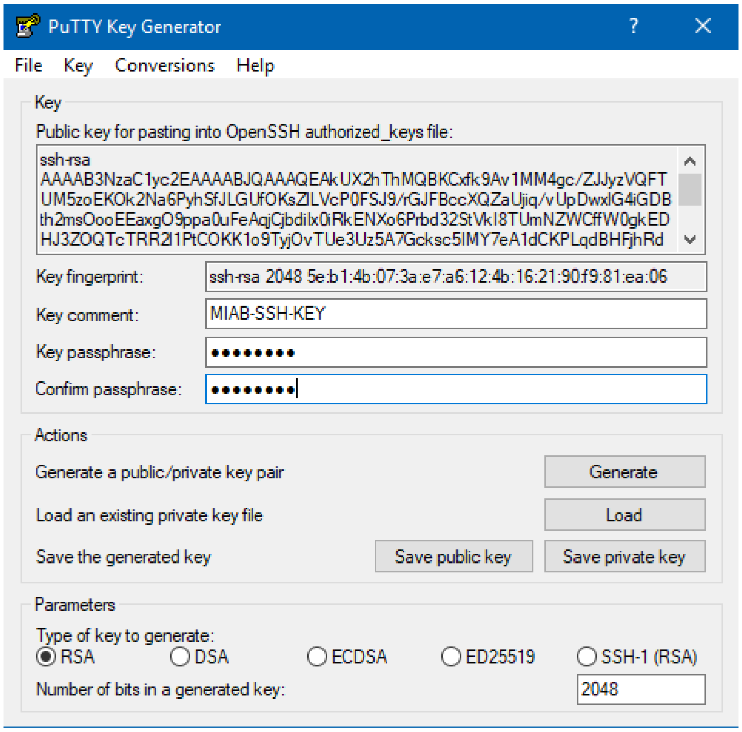
1. Go ahead and open puttygen and generate a new private / public key pair. You can password protect (recommended) this key pair for better security.
2. Press the "Generate" button in the application, follow the instructions and then fill out the form when it is done generating.
3. Give it a key comment, a password if desired (again, recommended!), and then select "Save public key" AND "Save private key".
4. If you plan on using the private key to authenticate on more than just your Windows machine, you can select "Conversions -> Export OpenSSH Key" from the menu bar. However if only on Windows, Saving the private key is enough.

On your server, do:

```
nano ~/.ssh/authorized_keys
```

And add the "Public key for pasting into OpenSSH authorized_keys file" and then press CTRL+X, then Y, then enter.

(Picture for reference)



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAkUX2hThMQBKCxfk9Av1MM4gc/ZJJyzVQFT
UM5zoEKOk2Na6PyhSfJLGUfOKsZlLVcP0FSJ9/rGJFBccXQZaUjiq/vUpDwxlG4iGDB
th2msOooEEaxgO9ppa0uFeAqjCjbdilx0iRkENXo6Prbd32StVkI8TUmNZWCffW0gkED
HJ3ZOQTcTRR2l1PtCOKK1o9TyjOvTUe3Uz5A7Gcksc5lMY7eA1dCKPLqdBHFjhRd
```

Key fingerprint: ssh-rsa 2048 5e:b1:4b:07:3a:e7:a6:12:4b:16:21:90:f9:81:ea:06

Key comment: MIAB-SSH-KEY

Key passphrase: ●●●●●●●●●

Confirm passphrase: ●●●●●●●●●

Generate a public/private key pair

Load an existing private key file

Save the generated key

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

Once the authorized_keys file is saved, we need to edit one more file:

```
sudo nano /etc/ssh/sshd_config
```

And modify the following line(s):

```
...
PasswordAuthentication no
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
...
```

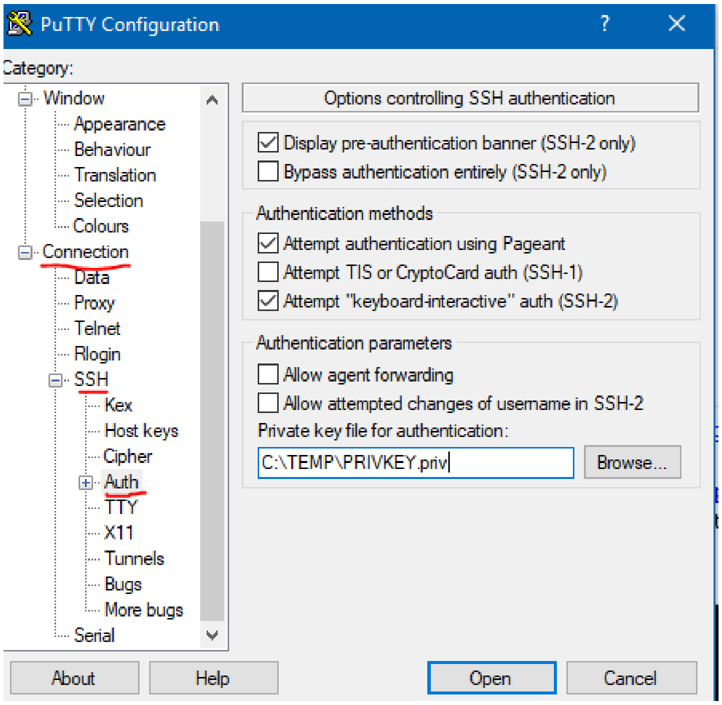
Save the file, and restart SSH:

```
sudo systemctl restart ssh
```

Once that is done, try to authenticate with the SSH private key, not your password. In Linux this is as simple as just referencing the new key in the SSH command line:

```
ssh -i ~/path/to/key username@ip_of_machine
```

In Windows, you will need to use PuTTY and then configure putty to use the private key:



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
□ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
□ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

□ Allow agent forwarding
□ Allow attempted changes of username in SSH-2

Private key file for authentication:
C:\TEMP\PRIVKEY.priv

If the connection was successful, then we can move on to the next step. However, if you have any issues, you will need to consult Google as troubleshooting Ubuntu issues is outside the scope of this Manual. (However, make sure you are using the right private key, password, and user. If anything else, in the open SSH session you should still have open, check the syslogs for any errors as well.)

Installing Mail-In-A-Box

Installing Mail-In-A-Box

> Before continuing, please make sure you followed the "Setting up Ubuntu 18.04 and Preparing for Mail-In-A-Box" page. There are some recommended steps to make the below go smoothly.

> **Warning!** By installing Mail-In-A-Box, you agree to the Let’s Encrypt Subscriber Agreement(s) & Terms of Services.

Once inside, you will now get the Mail-in-a-Box code onto your box and start its setup. Copy and paste this into your terminal and hit enter:

```
curl -s https://mailinabox.email/setup.sh | sudo -E bash
```

> **Advanced:** To change the default location where Mail-in-a-Box stores all of its data, you can set an environment variable named 'STORAGE_ROOT' *before* running the setup script.
>
> `export STORAGE_ROOT=/your/desired/path`

You will be asked to enter the email address you want and a few other configuration questions. At the end you will be asked for a password for your email address.

This password will be used to login to webmail, the administrative interface, and on your devices. It will **not** be used to log onto your Mail-in-a-Box server using SSH.

It is always safe to re-run the setup, either because something went wrong or you just want to see it again. You can do so by following two the steps above again or just running `sudo mailinabox` from the command line.

If the installation was successful, you should see:

```
Your Mail-in-a-Box is running.
Please log in to the control panel for further instructions at:
    https://ip_of_your_box/admin
You will be alerted that the website has an invalid certificate. Check that
the certificate fingerprint matches:
```

TLS / SSL Certificates

Go ahead and login to the admin panel. The first thing we should do now is get SSL certificates on this server. Head over to the TLS Certificates page of the admin panel and press "Provision" on the box.domain.tld you selected during install. This will take a few moments, so be patient and follow on-screen instructions. If you do not want to use Let's Encrypt (Though it is recommended that you do) you can manually import TLS/SSL certificates via this page as well.

Setting Up Your First Domain

Updating & Maintaining

Mail-In-A-Box

Troubleshooting & Diagnosing Issues with Ubuntu & Mail-In-A-Box

Getting Help

If you need **help**, please check the maintenance guide and then ask on the forum. If you think you have found a problem in Mail-in-a-Box or don't get a response on the forum, then open an issue on github.

We will post announcements and security advisories to our twitter account @mailinabox, the announcements section of the discussion forum, and our Slack chat (see above).

Do not tweet questions: Always start on the forum so others can benefit from seeing your question too.

Reporting spam/fraud/abuse: Mail-in-a-Box actually isn't a mail service at all. It is more like a cooking recipe for how to create a mail service — therefore, we have no way to know who is following the recipe and have no control over people baking our cake to hide a poison. Like cake, there are many recipes for creating email servers besides Mail-in-a-Box (Microsoft Exchange Server being one of the most popular), and we are just unlucky when sometimes someone with bad intentions choses ours. In other words, we don't control how people use Mail-in-a-Box and have no technical or legal means to disable other people's services. You may want to check out the Mail-in-a-Box discussion forum and coordinate with anyone that has reported a similar situation recently.