ITFlow

- Install ITFlow

Install ITFlow

https://www.youtube.com/embed/Ul10HXaPwNk

Setup a Server

In most self hosted applications these days, you'll need a machine to act as a server. This can be a machine in your home / business (such as an old laptop or desktop, a Raspberry PI or Single Board Computer (SBC), or even the computer you're reading this article from), a VM / Container hosted on one of your machines, or a VPS (Virtual Private Server) hosted by companies like RackNerd, Digital Ocean, Linode, Vultr, and so many more. Regardless of which option you choose, you'll want to do a few things to get the server setup properly.

Install updates to our Server

Ubuntu / Debian

`sudo apt update && sudo apt upgrade -y`

RedHat / CentOS / Fedora / Alma / Rocky

`sudo dnf update -y`

Add a non-root / sudo user on the server

Generally, when you setup a new server, the VPS (Virtual Private Server) service sets up a default "root" user for you. It's considered unsafe to do everything as "root", so let's setup a non-root user who has super user (sudo) privileges.

`adduser <username>` You'll be prompted to enter and confirm a password for this user. You'll also be asked for some user information like Name, etc, but this is not required information. At the end, confirm the entries, and you'll have your new user.

Next, we need to add the user to the super user group.

Ubuntu / Debian

`usermod -aG sudo <username>`

RedHat / CentOS / Fedora / Alma / Rocky

`usermod -aG wheel <username>`

Now, you can log out of the system, and log back in as your new non-root super user.

Install Docker and Docker Compose

Fortunately, there is a single line command that will install both Docker and Docker Compose for us on most Linux based distributions.

We need the 'curl' utility to get this to work, so if you don't have it, you'll want to install it first with

Ubuntu / Debian

```
sudo apt install curl -y
```

RedHat / CentOS / Fedora / Alma / Rocky

```
sudo dnf install curl -y
```

Next, we'll run the command to install Docker and Docker Compose:

```
curl https://get.docker.com | sh
```

You may be prompted to enter your super user password, so be ready for it. Once you do, the install should proceed.

Once complete. we want to add our user to the 'docker' group so we can do `docker` and `docker compose` commands without having to type in `sudo` each time.

```
sudo usermod -aG docker <username>
```

Now we'll logout / exit the session, and log right back in so the updated group will take effect.

Install ITFlow in Docker using Docker Compose

Let's create a new folder for our docker application.

```
mkdir -p docker/itflow
```

Now we'll move into the folder, and create two files. First, the 'compose.yaml' file, and then the '.env' file. The 'compose.yaml' file is where we define what services our application will need, as well as any storage, and internal networking we need for the services to communicate.

The '.env' file is where we define the environment variables we want our services to use.

```
cd docker/itflow
```

`nano compose.yaml`

NOTE - I had to build the docker image myself in order get the 'entrypoint.sh' file included in the image properly. The repo image was throwing an error about it missing.

Next, copy the yaml code block below, and paste it into the open editor.

```
networks:
  wan:
    name: wan
    driver: bridge
  itflow-db:
    name: itflow-db
    external: false

services:
  itflow:
    platform: linux/amd64
    hostname: itflow
    container_name: itflow
    image: itfloworg/itflow:latest
    restart: unless-stopped
    depends_on:
      - itflow-db
    networks:
      - wan
      - itflow-db
    ports:
      - "8090:8080"
    environment:
      - TZ=$TZ
      - ITFLOW_NAME=$ITFLOW_NAME
      - ITFLOW_URL=$ITFLOW_URL
      - ITFLOW_PORT=8080
      - ITFLOW_LOG_LEVEL=info
      - ITFLOW_DB_HOST=itflow-db
      - ITFLOW_DB_PASS=$ITFLOW_DB_PASS
    volumes:
      - ./itflow:/var/www/localhost/htdocs

  itflow-db:
    hostname: itflow-db
    container_name: itflow-db
    image: mariadb:latest # static 10.11.6
    restart: always
    networks:
      - itflow-db
    environment:
      - MARIADB_RANDOM_ROOT_PASSWORD=true
      - MARIADB_DATABASE=itflow
      - MARIADB_USER=itflow
      - MARIADB_PASSWORD=$ITFLOW_DB_PASS
    volumes:
      - ./itflow-db:/var/lib/mysql
```

> If you happen to have issues after running the above, you can instead use the build process. This is what I had to do, but I did report the issue I ran into, so hopefully it is fixed, and you can run the above and pull the pre-built image. If not, here's the compose.yaml for the build version.
>
> If you are running the build version below, you'll need to also download two files for the build to work properly. You can download them into the same directory where your 'compose.yaml' file is located with the commands:
>
> `wget https://raw.githubusercontent.com/itflow-org/itflow-docker/refs/heads/main/entrypoint.sh`
> `wget https://raw.githubusercontent.com/itflow-org/itflow-docker/refs/heads/main/Dockerfile`

```
networks:
  wan:
    name: wan
    driver: bridge
  itflow-db:
    name: itflow-db
    external: false

services:
  itflow:
    platform: linux/amd64
    hostname: itflow
    container_name: itflow
    build:
      context: .
      dockerfile: Dockerfile
    restart: unless-stopped
    depends_on:
      - itflow-db
    networks:
      - wan
      - itflow-db
    ports:
      - "8090:8080"
    environment:
      - TZ=$TZ
      - ITFLOW_NAME=$ITFLOW_NAME
      - ITFLOW_URL=$ITFLOW_URL
      - ITFLOW_PORT=8080
      - ITFLOW_LOG_LEVEL=info
      - ITFLOW_DB_HOST=itflow-db
      - ITFLOW_DB_PASS=$ITFLOW_DB_PASS
    volumes:
      - ./itflow:/var/www/localhost/htdocs

  itflow-db:
    hostname: itflow-db
    container_name: itflow-db
    image: mariadb:10.11.6
    restart: always
    networks:
      - itflow-db
    environment:
      - MARIADB_RANDOM_ROOT_PASSWORD=true
      - MARIADB_DATABASE=itflow
      - MARIADB_USER=itflow
      - MARIADB_PASSWORD=$ITFLOW_DB_PASS
    volumes:
      - ./itflow-db:/var/lib/mysql
```

Use CTRL + O to save your changes, press Enter to confirm, and exit the nano editor with CTRL + X.

Now let's create another file call '.env'. This file will hold the values for the environment variables used by our compose.yaml file. We do this so we can dfine a variable once, but it can be used multiple times throughout the compose.yaml file with less chance of a mistake, or mistyping.

`nano .env`

Now, copy the code block below, and paste it into the open editor.

```
# Set container timezone
TZ=America/Chicago


# Used within the docker-compose.yml template to provide easy configuration for your domain
ITFLOW_URL=it.yourgreatdomain.com


# Generate a random password using  openssl rand -base64 32
ITFLOW_DB_PASS=<openssl rand -base64 32>
```

Use CTRL + O to save your changes, press Enter to confirm, and exit the nano editor with CTRL + X.

For the file above, we need to replace the ITFLOW_URL with the actual subdomain and domain you want to use. You should own the domain, or at least have rights to create subdomains for the domain you intend to use, and the ability to create A / CNAME DNS records for the domain.

Next, you need to run the command `openssl rand -base64 32` in the command line, and let openssl generate a nice long random database password for you.

I like to remove the '=' equal sign at the end, and replace it with a random number or character as well.

Copy the generated password, and re-open the .env file with

`nano .env`

Move down, and replace the text surrounded by less than '<' and greater than '>' signs with the password you copied.

Once again use CTRL + O to save your changes, press Enter to confirm, and exit the nano editor with CTRL + X.

We are now ready to pull down the images we need in order to run our application.

`docker compose pull`

Once pulled, we'll tell docker to start our services in docker containers (tiny virtual machines) with the commands:

```
docker compose up -d && docker compose logs -f
```

These commands (concatenated by the &&) tell docker to bring up the services in detached (-d) mode, and once they are up and running to follow (-f) the logs as the applications start up. We can stop following the logs with the hotkey combination of CTRL + C.

As long as you don't see error messages of exit codes other than 0, the application should come up and running within 30 seconds or less.

You can test that you are able to reach the site by visiting the IPv4 address of the site and port 8090.

I visited http://192.168.10.20:8090

I was greeted by the ITFlow setup wizard, which lets me know everything appears to be working. Before I go through the wizard, however, I want to get my domain and subdomain setup to access the site securely over https (SSL encrypted) and by the desired domain / subdomain name.

Reverse Proxy Setup

A reverse proxy allows you to setup a domain / subdomain name for your service(s), and make sure those services can be reached, even when you run multiple different services on a single host machine with a single IP address.

I am using the Pangolin reverse proxy tunnel lately, and I really like it, but regardless of which reverse proxy you use, you want to setup the reverse proxy to be reachable by a wildcard DNS entry such as an A-record which points to the public IPv4 address where your reverse proxy is running.

Record Type	Domain / subdomain	IPv4	Running Here
A	*.yourgreatdomain.com	20.30.40.91	Reverse Proxy - Port 80 and 443
NA	NA	192.168.1.21	ITFlow - Port 8090

Your subdomain points to your reverse proxy server on ports 80 and 443. Your reverse proxy then points to your services and applications on their IPs and ports.

You type it.yourgreatdomain.com int he browser ->
-> this checks DNS and send you to 20.30.40.91 ->
-> the reverse proxy checks to see if the entry for it.yourgreatdomain.com exists in it's entries ->
-> and if so, it sends you to the service ip and port ->
-> 192.168.1.21:8090

I have videos on setting up NginX Proxy Manager, a really great reverse proxy with a clean Web User Interface. I also have a video on Pangolin, which helps tunnel traffic through to your LNA (local network) without having to open ports to the internet.

Once your reverse proxy is setup, you're ready to load your ITFlow site by URL. It should still show you the setup wizard. Let's go through that now.

ITFlow Setup Wizard

If you skipped to this step, you should make sure you have a reverse proxy setup, or have a subdomain setup that you are accessing this application by before you start the wizard.

Click the 'Bgin Setup' button. You'll be taken to the system check. In my case, because I'm using a reverse proxy for SSL, the SSL and https checks show x's, but I can verify in the URL bar that I am accessing it through https.

Click the 'Next (Database)' button. For the Database entries, we'll get those values from our 'compose.yaml' and '.env' files.

Field	value you should enter
Database Name	itflow
Database Host	itflow-db
Database User	itflow
Database Password	<password you generated in .env file>

Next, we'll create our first user. This user will be the administrator of the system. Make sure to give this user a long, strong password, and save it in a password manager.

After creating your first user, you'll want to setup your company in ITFlow. Enter all of the details for your IT / MSP company.

Next, fill out the details for your country, currency, and language.

After that, fill in any comments you have for the ITFlow team, and check the 'Share' box if you want to share data. If not, leave it unchecked, and click the button to continue forward with final install.

Once complete, you'll be redirected to the Login screen. Login with the admin user you created, and you are now ready to dive into ITFlow.

Settings

There are literally dozens of settings available, and you need to take the time to go through them, gather the information required, and get the system setup properly.

You can access the settings / administration area by clicking on your avatar at the top right, then selecting 'Administration' from the menu that comes up.

I recommend you check out the Mail, Security, Theme, Defaults, and if you use Microsoft Entra, the Identity Provider sections to start. But you should really check them all out.