Private Server

INSTALLATION GUIDE – 2025/10

R T

Contents

Introduction

If you want to install the Ubikron server **we assume a small amount technically know-how.** You don't need to be a dev-ops expert, but you should not be intimidated by a terminal.

The Ubikron private server is a Docker Stack of 12 containers. We provide installation instructions for different scenarios. The Ubikron server can run on:

1. same machine as client (localhost). Support for Windows, Mac and Linux.
2. on-premises host in a private network
3. VPC (for instance EC2 instance, Linode or other)

Important things for all installations:

Minimum hardware requirements

Ubikron loves memory. It will run with 16GB of RAM, but **it really wants 32GB** or more. CPU cores - all you can spare, but memory is much more important. A fast, low latency and stable connection to the server also makes for a much better experience.

DNS/SSL

- **If the server is running on your own machine, you don't need to worry about DNS or SSL certificates at all as communication is internal to your computer. You can skip this section.**
- The communication between the Ubikron extension and the Ubikron server needs to happen with a real (not self-signed) TLS/SSL connection. This is a restriction on Chromium browsers, not something we enforce.
- To facilitate this, we have created an easy, hands-free way to create and distribute certificates using the DNS4SSL.net domain. This happens in the installation, you only need to choose a name, this name could be anything. We also handle renewal of certificates.
- As a very basic level of obfuscation, we will add a dash, two letters and two numbers after the name you have chosen. So if you use 'test' your name might become test-ab12. We do this because a specific user have chosen obvious names and tied to an IP address that were on their public ranges.
- If, for some reason, you need to reinstall and you need to keep the DNS name you chose before, you can do so. Enter the full DNS name – e.g. 'test-ab12' and you can get the name back.
- You can **use your own DNS** of course. In that case, you will need your own certificates too, but you will need to generate and sign them yourself. And you need to renew them yourself. Please follow the section entitled 'Advanced install notes for DevOps ninjas' for more information re ninja-ing.

Install directory

- This is the directory where Ubikron will save data to and where the server "lives". Everything about the server will be contained in this directory.

- If you delete and remove the Docker containers/images/volumes, this directory will still contain all Ubikron data. **This is important to know when you want to safely remove all the data from a Ubikron server.**
- This also means that you can re-install your Ubikron server completely without losing your data. If you want to do this simply ensure that you specify the same install directory in the installation process.

Multi-user machine / Security considerations

It is assumed that the machine where you install Ubikron on will either be your own computer (for local installs) or it will be dedicated server for Ubikron. In other words, we don't specifically protect the server from other users on the same machine (think Unix/Linux based machines). Access to the server would be considered access to Ubikron.

We also assume that you will take security measures to protect the machine. This includes strong passwords, closing unused ports, on-device firewalls, and removal of unwanted services. You are responsible for securing the server.

In terms of network level filtering, we need ports 80,443 and 7001 to be open to the server.

Mac installer drama-rama

Both the Mac and Windows installers that you can download from the website are cryptographically signed. You may check the signatures yourself.

To run the binary (from the terminal) on Mac your computer needs to be able to verify this signature with a computer at Apple – this requires an Internet connection. This is not something we enforce, this is something Apple does. If your Internet connection is not working or weak sauce, this check may fail and you'll get a response that says 'Operation not permitted'. In this case you need to download the PKG from the website and install it like a normal PKG. The installer will drop the file 'ubikron-server-macos' in /usr/local/bin and you would be able to just run the command from the terminal.

PKGs can be stapled, meaning that your machine does not need to contact Apple to run it, so when you don't have Internet access, this is the way to get the binary securely to you.

Resetting admin password

You might get into a situation where you forget the password to the administration interface. There is way to reset the password from the host.

In the server installation directory you see a directory called '**reset-password**'. Create a file (could be empty) called '**reset-admin.txt**'. The server checks for this file every few minutes, so you would have to perhaps wait a few seconds. When the server sees this file on the file system it will reset the password to 'admin'.

On login, you will be forced/prompted to choose a new password.

Local install

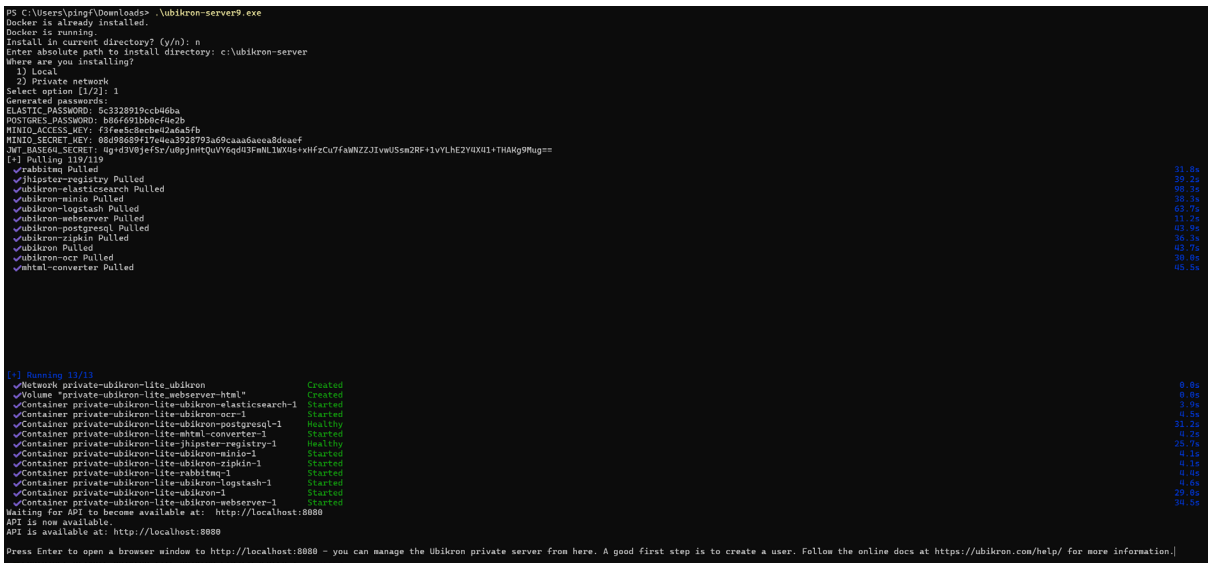
In other words - Install Ubikron server on same machine as Chrome Extension. This is a typical install for someone that wants to test the server out, but that does not have access to a VPC or "own server" infrastructure.

1. Ensure you have Docker installed.
    a. We use Docker Desktop for Windows or Mac.
2. Start Docker, ensure it's running.
3. Download the Ubikron Server installer for your operating system.
4. Open a terminal.
5. Run the installer
    a. For Linux and Mac users – you might need to run
        '*chmod +x ubikron-server-install*' beforehand
    b. For Linux users – you will need to run the installer as root – e.g.:
    sudo ./ubikron-server-install
6. Determine where to install the server– e.g. *d:\data\ubikron-server\ or /data/Ubikron-server* or whatever. You don't need to create these beforehand, we will create them during the installation.
7. In 'Where are you installing', pick '**Single user - install on Localhost only'**.

The following will now happen:

- The installer will generate random passwords for all the services.
- Docker images will be fetched to your Docker Desktop. Be patient.
- The containers will be started.
- We will wait for server to become available (**be patient, the start can take a while**)

The entire process looks like this (on Windows):



Ubikron private server is now installed, and you can navigate to the administrator interface (see later).

Dedicated server installation

In other words - install Ubikron server on a separate machine (in a private network).

This is a typical installation when you don't want to keep the server on your own machine, when it's in your private network, on your premise on a dedicated VM or host and you have multiple users connecting to it.

Of course, you could also install Ubikron server on a server with a public IP address like an EC2 instance or VPC.

1. Ensure you have Docker installed.
2. Start Docker, ensure it's running.
3. Download the Ubikron Server installer for the operating system.
4. Open a terminal and run the installer
    a. For Linux and Mac users – you might need to set the file as executable: '*chmod +x ubikron-server-install*' first
    b. For Linux users – you will need to run the installer as root – e.g.: *sudo ./ubikron-server-install*
5. Determine where to install the server – e.g. *d:\data\ubikron-server\ or /data/Ubikron-server* or whatever you choose. You don't need to create these beforehand, we will create them during the installation.
6. In 'Where are you installing', pick '**Multi-user - VPC /private network**'
7. Choose a name for your server. It could be any name that you will remember. A DNS4SSL.net address will be given to you – e.g. demo99.dns4ssl.net. This is what you'll use in the client and for your users.
    a. If the name is already taken you will be asked if you want to use the name/IP combo. Useful when re-installing. Not so useful if you can't reach the machine or if it's not yours.
    b. If you want to use your own DNS/SSL configuration you can read the 'Advanced install notes for DevOp ninjas' section.
8. If your machine has more than one interface, select the interface that the DNS name will be assigned to.
    a. We show ALL addresses. We also show the public IP – in some cases (for instance if you plan to do port forwarding) this is useful to use – but:
    b. **NB** – this should be IP address that you use to reach the server from the client machines. If you cannot connect to this IP address from your machine, then pick something else.

The following will now happen:

- The DNS name will be created for you; the SSL certificate will be obtained.
- The installer will generate random passwords for all the services.
- Docker images will be fetched. Be patient.
- The containers will be started.
- We will wait for server to become available (**be patient, the start can take a while**)

The entire process looks like this:

```
Docker is already installed - good!
Docker is running - good!
Install in current directory? (y/n): n
Enter absolute path to install directory: c:\ubikron-s1.0
Select your install type:
  1) Single user - install on Localhost only
  2) Multi-user - VPC /private network
Select option [1/2]: 2
Enter the DNS name you want to use (e.g. 'myserver' for myserver.dns4ssl.net): test
Generated DNS name: test-pb56.dns4ssl.net
Multiple IP addresses found:
  1) 105.184.27.123  (public)
  2) 10.0.0.170  (local)
  3) 192.168.56.1  (local)
  4) 169.254.188.61  (local)
  5) 172.20.48.1  (local)
  6) 172.24.240.1  (local)
Select interface / IP address to be used for DNS: 2
Requesting certificate from provider...
Certificate and key installed in c:\ubikron-s1.0\certs
Generated passwords:
ELASTIC_PASSWORD: 3bf1ed6348674e50
POSTGRES_PASSWORD: ced3fb98d3d14bbf
MINIO_ACCESS_KEY: bf597f0cc35f4217833f
MINIO_SECRET_KEY: 3ed36bf450de4b69931ddfa02d10911863a4f892
JWT_BASE64_SECRET: zS8np6pPS97zLXYGhwcBRmD6Y3HFBguWckHma2qO5t4VPsMhct0yn+7Xn6PTcGO9Us3UjPlPwGcKSyajaifBmA==
[+] Pulling 12/12
 ✔ubikron-webserver Pulled
 ✔rabbitmq Pulled
 ✔ubikron-ocr Pulled
 ✔ubikron-wud Pulled
 ✔ubikron-zipkin Pulled
 ✔mhtml-converter Pulled
 ✔ubikron-elasticsearch Pulled
 ✔ubikron-minio Pulled
 ✔jhipster-registry Pulled
 ✔ubikron-postgresql Pulled
 ✔ubikron-logstash Pulled
 ✔ubikron Pulled
[+] Running 12/12
 ✔Container private-ubikron-lite-ubikron-zipkin-1         Running
 ✔Container private-ubikron-lite-ubikron-ocr-1            Running
 ✔Container private-ubikron-lite-ubikron-elasticsearch-1  Running
 ✔Container private-ubikron-lite-ubikron-wud-1            Running
 ✔Container private-ubikron-lite-ubikron-postgresql-1     Healthy
 ✔Container private-ubikron-lite-ubikron-logstash-1       Running
 ✔Container private-ubikron-lite-mhtml-converter-1        Running
 ✔Container private-ubikron-lite-rabbitmq-1               Started
 ✔Container private-ubikron-lite-ubikron-minio-1          Started
 ✔Container private-ubikron-lite-jhipster-registry-1      Healthy
 ✔Container private-ubikron-lite-ubikron-1                Started
 ✔Container private-ubikron-lite-ubikron-webserver-1      Running
Waiting for server to become available at:  https://test-pb56.dns4ssl.net
Server is up!
You can access the server's admin interface at https://test-pb56.dns4ssl.net
A good first step is to create a user. Follow the online docs at https://ubikron.com/help/ for more information.
Press Enter to close the installer.
```

The installation is now complete.

Advanced install notes for DevOp ninjas

Note – Ubikron needs real SSL/TLS certificates. It cannot run on self-signed certificates (this is a requirement from Chrome, not us). We assume you have PKI internal to your organization or using real certificates signed by a real CA that is installed in your organization's browser. You need these real certificates in `server.crt` and `server.key` – how you get these is for the ninja in you.

We assume you are in full control of your DNS too.

Here is the recipe:

- Decide on a DNS name for your server and assign it to the IP where the Ubikron server will run
- Make the certificates – yay. This is for yourself, your PKI and/or your CA.

- Make a Ubikron directory on the host:
    a. *mkdir /Ubikron-server*
- Copy docker-compose.yaml file to this directory. You will receive this file from us.
- Make a 'certs' (sub)directory in the server directory
    a. *cd /Ubikron-server*
    b. *mkdir certs*
- Copy your `server.crt` and `server.key` files into this directory.
    a. These are PEM encoded in ASCII armor, like any other certificate
- In the server root (*/Ubikron-server* in our case), create an .env file (note the dot in the filename) with config and component passwords. Obviously change these passwords:

```
S3_PUBLIC_URL=https://ubik-75.dns4ssl.net:7001
JHIPSTER_REGISTRY_PASSWORD=83409c0a88bd4711
ELASTIC_PASSWORD=3a03b2ec15d8463b
POSTGRES_PASSWORD=3b2957540c94468e
RABBITMQ_PASSWORD=91a15d66de5d4263
MINIO_ROOT_PASSWORD=0a19ac8d286842b2
MINIO_ACCESS_KEY=32565b229c3148faa3f2
MINIO_SECRET_KEY=1cf09c08e4454c55a9c8bd44c57a46098fef76a4
JWT_BASE64_SECRET=STRkkQNQ2zjW2CCLLKFMx76rPFdnh+UNiw1k216hnk
2BSlxqpRo3lXBW4KTR0fKRGr4TNmpqDRt9JufBlmFzRA==
```

- Set the `S3_PUBLIC_URL` in this file to the DNS name you've set up.
- The '`JWT_BASE64_SECRET`' token must be encoded using Base64 and be at least 256 bits long (you can type *openssl rand -base64 64* on your command line to generate a 512 bits one, like in the example)
- You are now ready. Run '`docker compose up -d`' to start the server.
-

```
r@vm15306887:/data/ubikron-s1$ ls -al
total 28
drwxr-xr-x 4 root root 4096 Aug  6 16:33 .
drwxr-xr-x 3 root root 4096 Aug  6 16:32 ..
drwxr-xr-x 2 root root 4096 Aug  6 16:33 certs
-rw-r--r-- 1 root root 7170 Aug  6 16:32 docker-compose.yml
-rw-r--r-- 1 root root  476 Aug  6 16:33 .env
drwxr-xr-x 3 root root 4096 Aug  6 16:32 volumes
r@vm15306887:/data/ubikron-s1$ ls -la certs/
total 16
drwxr-xr-x 2 root root 4096 Aug  6 16:33 .
drwxr-xr-x 4 root root 4096 Aug  6 16:33 ..
-rw-r--r-- 1 root root 2851 Aug  6 16:33 server.crt
-rw-r--r-- 1 root root  241 Aug  6 16:33 server.key
r@vm15306887:/data/ubikron-s1$ more .env
DNS_PROVIDER_URL=https://api.dns4ssl.net
S3_PUBLIC_URL=https://ubik-75.dns4ssl.net:7001
JHIPSTER_REGISTRY_PASSWORD=83409c0a88bd4711
ELASTIC_PASSWORD=3a03b2ec15d8463b
POSTGRES_PASSWORD=3b2957540c94468e
RABBITMQ_PASSWORD=91a15d66de5d4263
MINIO_ROOT_PASSWORD=0a19ac8d286842b2
MINIO_ACCESS_KEY=32565b229c3148faa3f2
MINIO_SECRET_KEY=1cf09c08e4454c55a9c8bd44c57a46098fef76a4
JWT_BASE64_SECRET=STRkkQNQ2zjW2CCLLKFMx76rPFdnh+UNiw1k216hnk2BSlxqpRo3lXBW4KTR0fKRGr4TNmpqDRt9JufBlmFzRA==
r@vm15306887:/data/ubikron-s1$
```

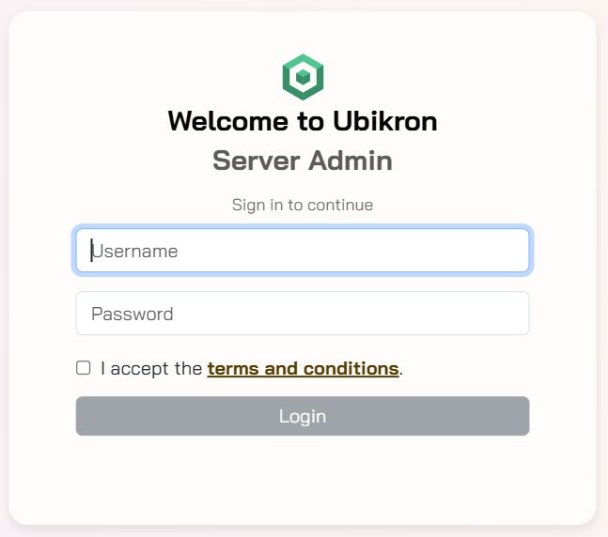
Feel free to look at the logs of the containers, how they all buzz together etc. Note – starting takes a bit of time. All the services will settle and you should see warnings and errors as they are waiting for each other to boot up. This is normal.

You should be able to browse to the admin interface – and there should be NO SSL/TLS warnings. **If there are – then Ubikron extension won't work.**

Administration guide

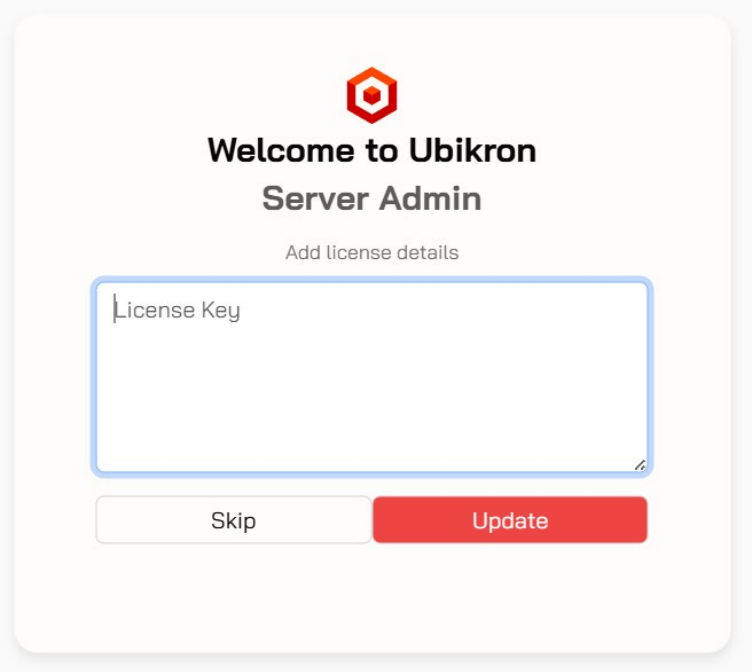
First time use

Now that your Ubikron server is installed you can browse the admin interface. The default username and password is **admin/admin.**



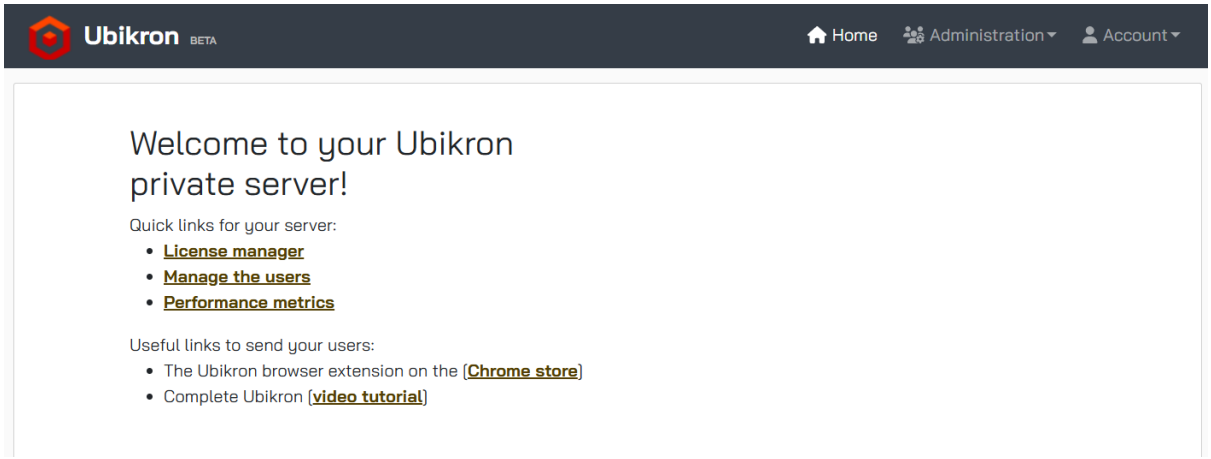
Once you are logged in you will be prompted to change the password. Choose a strong password! We don't need to baby you with password strength enforcement, you are an administrator!

The next step is to set the license for the server – if you have one. If you don't have a license, don't worry about it – click on 'Skip'. The free, default server comes with a single user and two projects. This is so you can test it and play with the server and see if you want to purchase one.

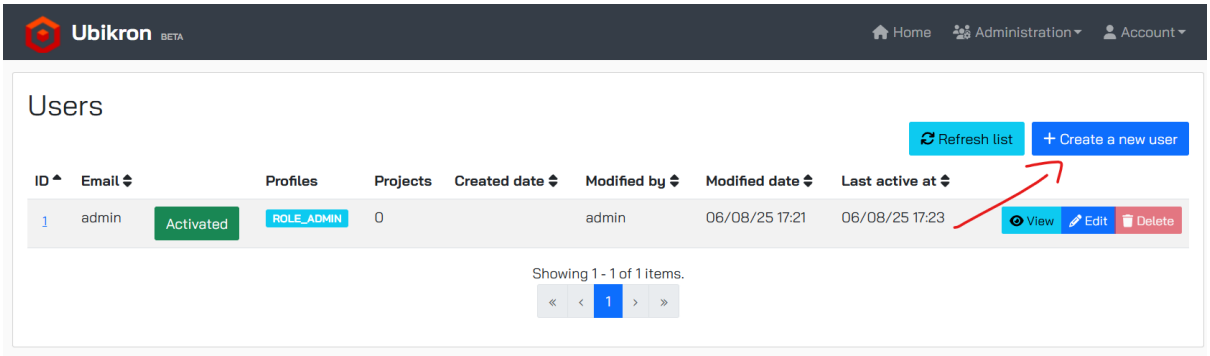


Adding new users

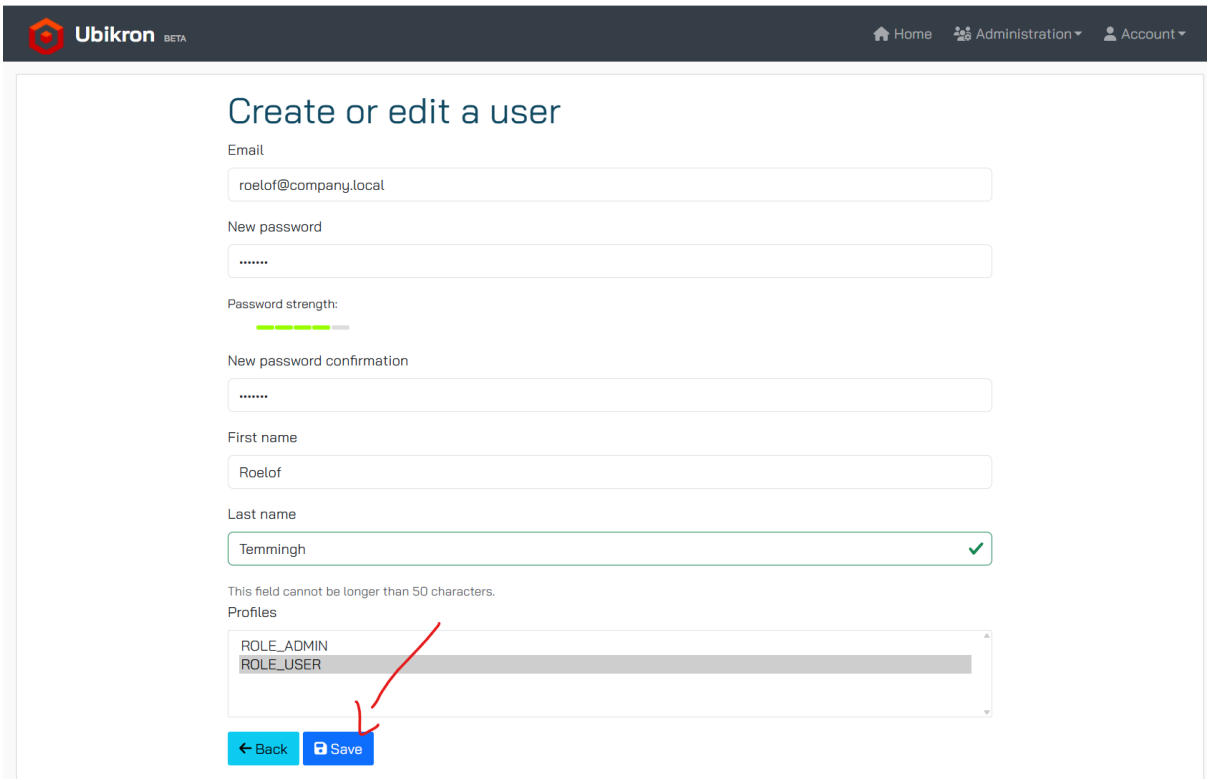
You are now on the Server administrator home page:



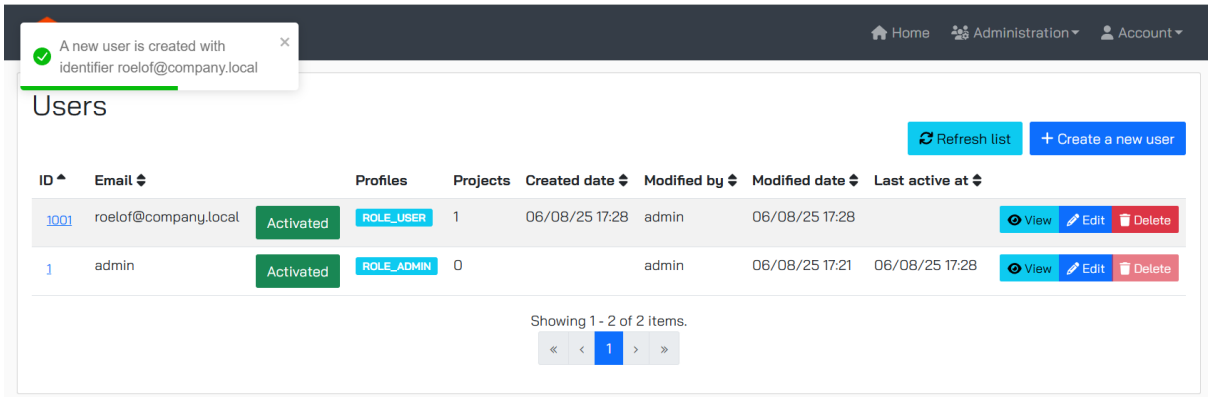
The first step is to create a new user account – this is user you'll use to log into the server with. Click on 'Manage the users'. Since there is only the admin user you should click on 'Create a new user':



Ubikron users always have an email address. Don't worry, the server will not email users – this is just a remnant of the public server. Feel free to use any dummy domain. Fill in the rest of the details and click on 'Save':



When you click on 'Save' a default project is created for the user, so the save operation takes a few seconds. You'll see the user appear in the interface:



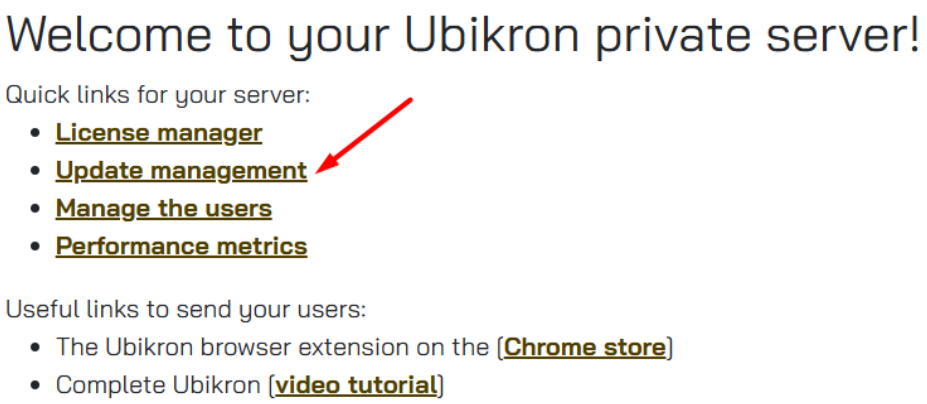
Updating the server

The Ubikron private server has a container that specifically checks on the Docker Hub for updates to any of images that make up the Ubikron private server. This check happens every hour.

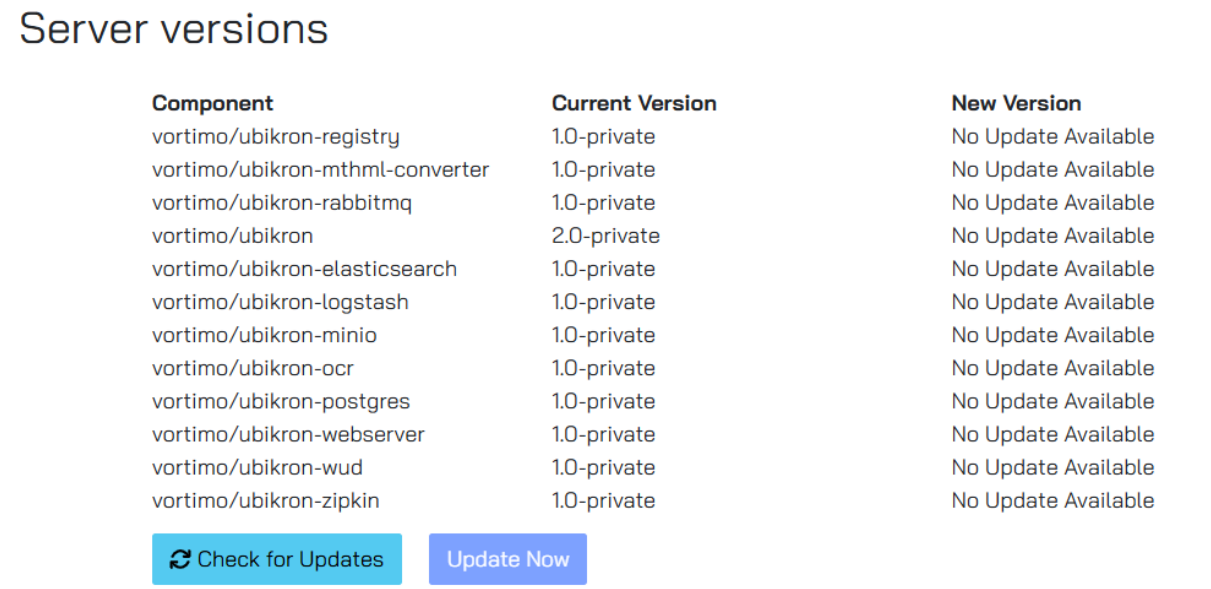
When an update is detected, a message is sent to all users on the system via the extension prompting them to ask the admin to update the server. We do this because we foresee that the admin will have no reason to log into the server on a regular basis (and will therefor not see the update). Upon logging into the server, the admin will also be prompted to update the system if they choose to do so.

Version information and deltas will be shown in the administration interface.

From the main interface, select ‘Update Management’:



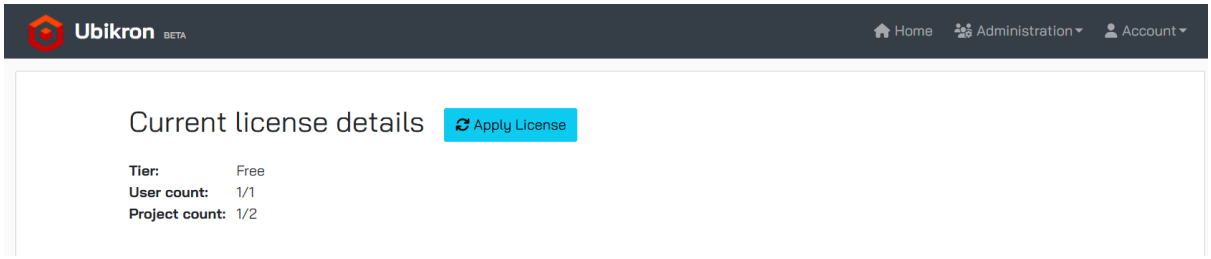
From here you will be taken to screen where you can manage the updates:



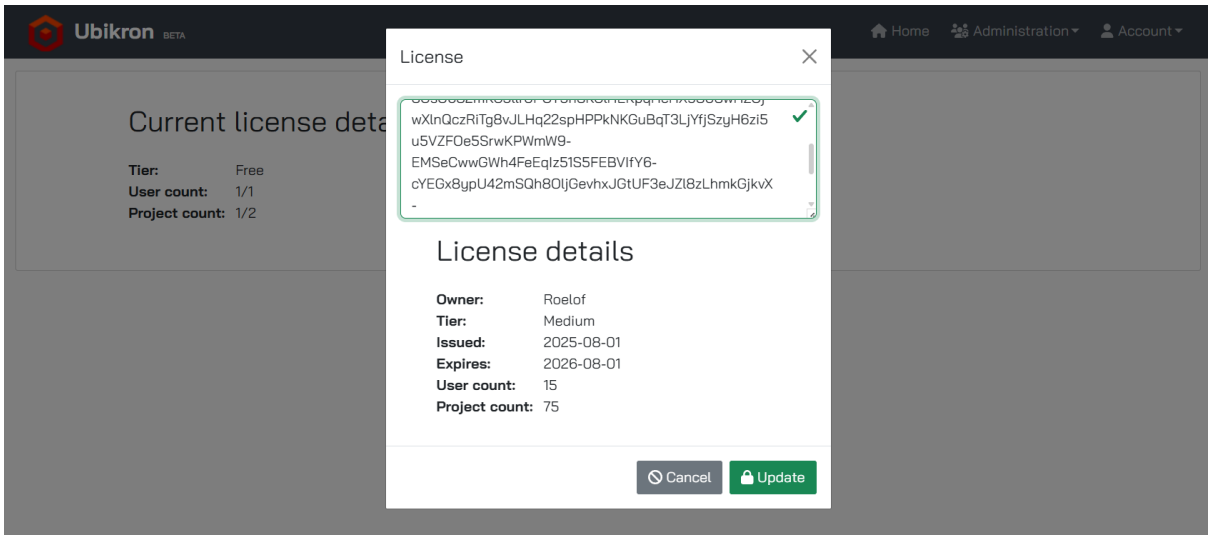
Clicking on the 'Check for Updates' will update the 'New Version' column. If there are updates available, you can click on the 'Update Now' button to apply them. Keep in mind that this could take a while to run, and during that time the server will be unavailable to users. You might even see errors appear on the management interface. This is normal. Just reload the interface after a while if it does not automatically update.

Entering a new license key

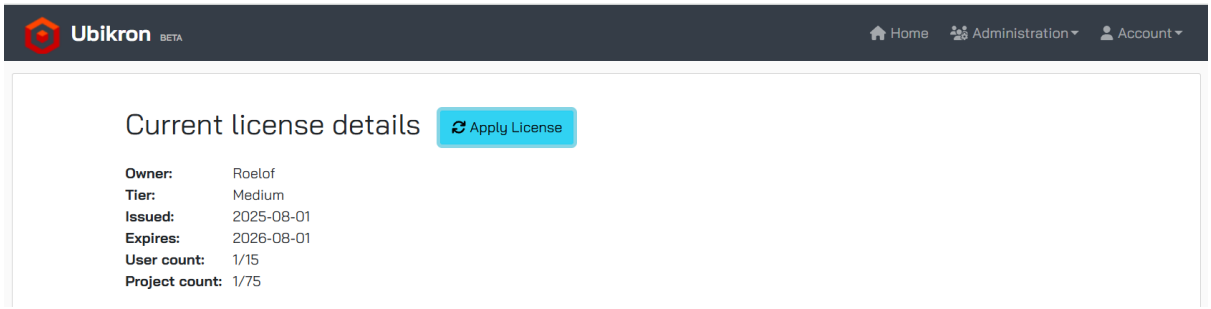
In the main menu, go to the 'License Manager'. You'll see the current license details:



Click on 'Apply License' and enter (mostly paste) the license given to you:



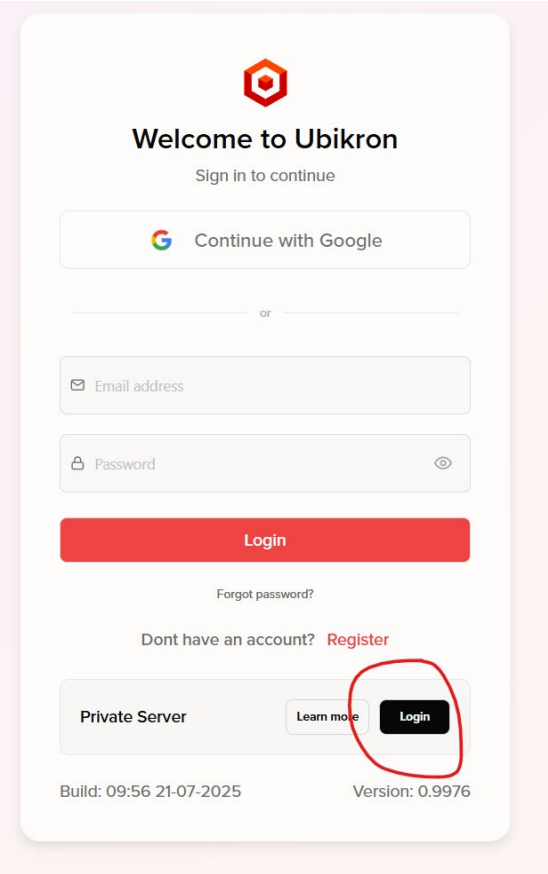
You'll see what the license details are. If this acceptable to you – enter 'Update' and the license will be applied:



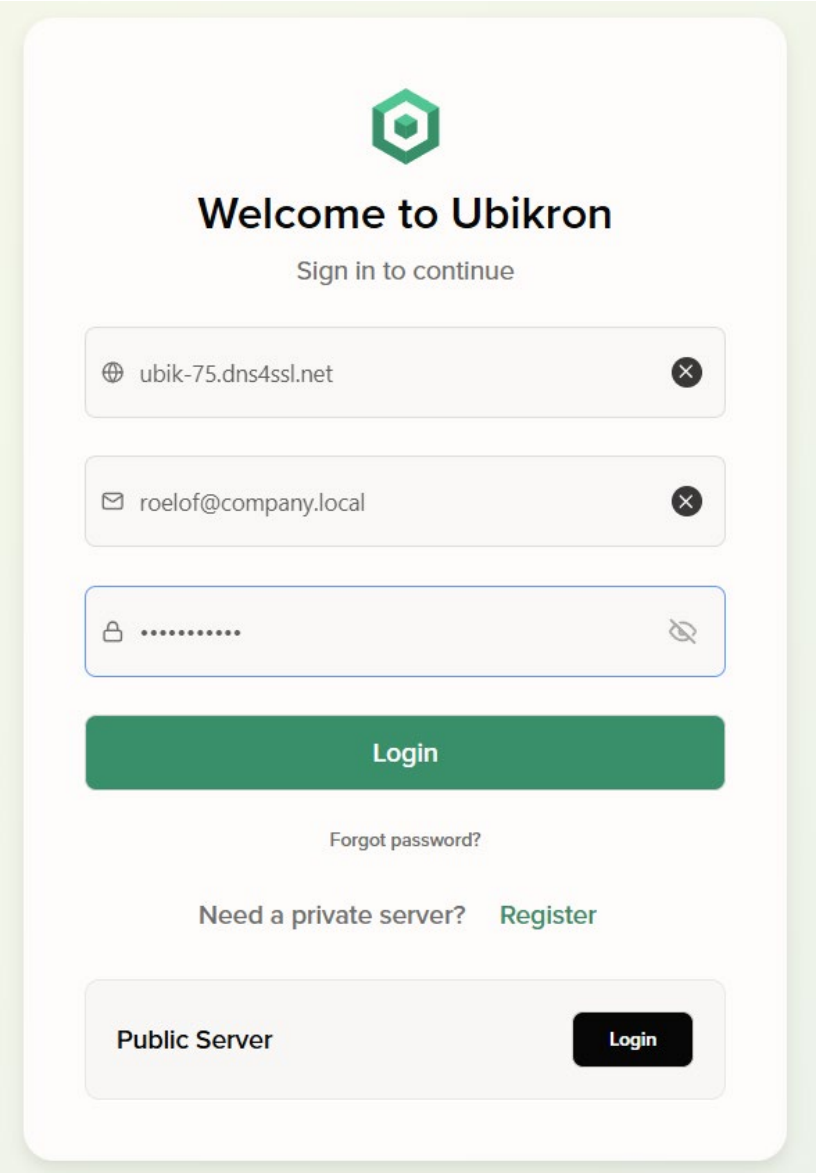
Using the Ubikron extension with your private server

When you've finally set up the server and you've added a license and you've added a user – you're ready to let your user use the standard Ubikron extension but connect to your own private server. Tada!

In the extension, at the log-in screen, click on the Private Server login button:



You will see that the interface will change from red to green and additional fields will appear:



In the IP Address / DNS name enter the DNS name of your server. Of course, when you have the server installed locally, you can just put in 127.0.0.1 or localhost.

In the Email field, enter the details of the user you've created – same with the password.

And – there you do, you're logged in: