Deploy with Docker

Deploy with Docker is a straightforward and highly recommended deployment method due to its simplicity and flexibility.

1. Preparation Steps

If you are using Windows,it’s reccomended that you download the **Cmder** that emulates a Linux prompt, allowing that you can execute your commands without problems.

Ensure that you have Docker and Docker Compose and operational. You can download on their website or execute the following command:

* curl -L https://get.docker.com | sh

If you want to run Docker without sudo, add the current user to the Docker group using the following commands:

* sudo usermod -aG docker $USER

sudo reboot

1. Deploy Rocket.Chat with Docker
   1. Fetching Compose File

Navigate to your preferred folder and create a composse.yaml following this [example](https://github.com/RocketChat/Docker.Official.Image/blob/master/compose.yml) or execute the following command:

curl-L https://raw.githubusercontent.com/RocketChat/Docker.Official.Image/master/compose.yml -O.

* 1. Editing Environment Variables

Next, we will create a .env file in your directory with the following command and paste the contentes of this [example](https://github.com/RocketChat/Docker.Official.Image/blob/master/env.example):

sudo nano .env

Next you will set the RELEASE variable in the .env to the Rocket.Chat version that you wish to use it and it’s available [here](https://github.com/RocketChat/Rocket.Chat/releases).

If you háve a domain or the IP address you will esit the ROOT\_URL and if you have a registration token to register the workspace automatically you can add it on REG\_TOKEN.

1. Iniciando o container

Now, save the .env fileand start up the container executing this command:

* docker compose up -d

This command will start a MongoDB servisse called **mongodb** and a service called **rocketchat**, which will wait for the **mongodb.**

You can access now Rocket.Chat workspace and invite others users.

1. Rocket.Chat Docker Images

Since the app is constantly being updated, it’s important that you keep your container with the latest files, and for that we can execute this command:

* docker pull registry.rocket.chat/rocketchat/rocket.chat

But, sometimes you will need to return to a previous if something happened with the one that you just updated and this can be done with the following command:

* docker pull registry.rocket.chat/rocketchat/rocket.chat:<release-tag>

It’s common that we can have a development environment so we can test the change and see if it’s working before it’s released to the production environment, and this can be done with this command:

* docker pull registry.rocket.chat/rocketchat/rocket.chat:develop