1. Arcane Docker Software

Arcane is a web-based Docker management UI that lets you manage containers, images, volumes, networks, and Docker Compose stacks. It connects to your Docker hosts (via the Docker socket) and gives a dashboard view plus tools like logs and resource monitoring.

- 1. Installation
- 2. Control remote Docker VM's
- 3. Container Updates
- 4. Resource Usage
- 5. Pruning of unused containers
- 6. Archiving Projects
- 7. Automatic Updates
- 8. Arcane Updates

1. Installation

Arcane will be the dashboard to use to install other docker applications with. Therefore, the only app we need to install via the command line is Arcane since after that we can use the Arcane dashboard for other apps.

1. The easiest way to install Arcane is by using their docker compose generator on their website. Just click on the link and then in the top left corner you will see docker compose generator.

1. Now, on the first page they will ask for the basic setup. Now, if you are going to run this locally you should change the APP URL to the IP address of your docker vm colon the port you want to run it on. The standard port is 3552 but I like to start with 1000. Then any other app you will increase the exposed ports with 1 i.e 1001, 1002,...

If you however want to use a domain name that is also possible by simply entering the domain name you want to access your arcane instance with. I would recommend to not expose your dashboard to the outside world and therefore it stays best to use a VPN to access it with a local domain name. However, the choice is up to you of course.

2. On the second step you can leave everything as is

3. On step 3 it will be asked where you want to store your projects. The default is at /opt/docker/projects/ which is completely fine. If you want something else then fill that in there

4. In step 4 you will define what the minimum warning level should be for it to be included into the logs. You can just leave this to info

5. Here you will generate you encryption key by just hitting Generate Key

6. Here you will choose a database. Normally SQLite is enough but it is recommended to use Postgres. Therefore, you can toggle on the Use External Postgres button and create your self a long ass complicated password.

7. The final step is Authentication and if you not plan to use an Single Sign on you can just leave it as is and generate the compose file

8. Then enter in your docker vm and create the following folders

```
mkdir arcane
mkdir /opt/docker
```

9. Then enter into the arcane folder to create the docker compose file with nano as follows

```
nano compose.yml
```

10. Then simply copy paste in the docker compose file that was generated

11. An example compose file can look like the one below

12. Now we will first pull all the images which I like to do before actually starting the container. If something is off or it cannot be pulled then you can first start to solve that problem. For that use the command

```
docker compose pull
```

13. Open DOCKERVMIP:3552 in your browser and follow the setup steps. The first time you sign in, you'll be asked to change the default admin password. Use these default credentials:

Username:

```
arcane
```

Password:

```
arcane-admin
```

14.

2. Control remote Docker VM's

Besides one controller, Arcane also offers the possibility to manage other docker vm's as well. In this way you can have a single glass that controls multiple docker vm's. For instance if you have a personal media, Arr or other important VM then you can manage all their resources and updates centrally. In this tutorial we will explain how to set this up.

1. The first thing we want to do as with the Arcane controller is to setup a project folder in which you will store all projects. This can be the same as for the Arcane controller e.g /opt/docker. Hence, make this folder and make sure it is owned by your user.

2. To make it owned by your user you can use the following command

```
sudo chown user:user -R /opt
sudo chmod 775 -R /opt
```

3. Once this is done go back into your Arcane controller.

4. Go to Environments > + Add Environment

5. Give it a name e.g "remote_dockervm_1"

6. Next add the Remote Docker VM IP address with a port you want to run it on e.g **3553** e.g 10.20.34.90:35537

7. Then click on generate agent configuration which will provide a docker compose file you can use for your remote docker vm.

8. Copy the generated compose file

9. Enter your remote docker vm with ssh

10. Create a new folder called arcane

```
mkdir arcane
```

11. Inside of the arcane folder create a new docker compose file and copy paste the generated compose file

```
nano compose.yml
```

> Note! here comes an important point to prevent Permission issues. It is important to include in this docker compose file your **User ID** and **Group ID e.g 1000,1000**. Otherwise you most likely will experience permission issues when you deploy applications from Arcane hat works on the UserID:65532 and GroupID:65532.

13. Then as an example it looks like this

compose.yml

```
services:
  arcane-agent:
    image: ghcr.io/getarcaneapp/agent:latest
    container_name: arcane-agent
    restart: unless-stopped
    environment:
      - PUID=1000
      - PGID=1000
      - AGENT_MODE=true
      - EDGE_TRANSPORT=poll
      - AGENT_TOKEN=arc_f18ea27f48b812813a6d6c1099b7ad0972beb22b6c532d1d253e6e89fb8f6fbe
      - MANAGER_API_URL=https://arcane.example.org
    ports:
      - "3553:3553"
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - ./arcane-data:/app/data
```

13. Now, for the volumes you need to give it access to your docker sock as with the arcane manager and also again the volume you want to store all your projects in.

14. Then simply start the arcane agent container with

```
docker compose up -d
```

15. Then go back to your Arcane manager and click on test connection. It should then show online if everything went well ;)

16. Sweet now you have a remote docker vm that can be managed from your main Arcan control plane.

3. Container Updates

4. Resource Usage

What Arcane also offers is resource usage and can even give you notifications when resources are hitting your set limits. This comes in handy especially

1. To trouble shoot which applications (containers) are using up most of your resources
2. Which applications are using up resources but that you don't use anymore

5. Pruning of unused containers

6. Archiving Projects

There might be a moment that you want to stop using an applications but want to keep the configurations for reference usage or because you want to work on it at a later stage. Then Arcane offers the option to archive these projects.

1. In that case it will be removed from your container and projects screen and you will only have green running projects!
2. Also when updates are executed it will not try to update the archived projects

To Archive a project

1. Go to projects > select the project you want to archive > click on Archive in the right top corner ;)
2. If you want to view your archived projects you go simply go to Projects > and then toggle on the Show Archived projects

7. Automatic Updates

As with any application updates are not only necessary for feature updates but also to remove security vulnerabilities. As with other docker management software Arcane offers a perfect solution to automate this process s.t you can just enjoy your apps without worrying about updates. Hence, this section will explain how you can do this yourself in minutes

1. First go to your Dashboard and select the Environment you want to monitor and have automatic updates for.
2. Then click on Settings > Automations
3. Then make sure you toggle on Auto Update and Image Update Watcher
4. Then simply hit save

Bamnnn and you are already done! now that was super simple wasn't it. Also note it is important to have a backup system in place in the case that an update crashes your application. If you have backups this all does not matter since you can just roll back to the backup before the update!

Sweet! Now updates happen in the background and you can enjoy new updates on the front end with it all being automated on the back end ;)

8. Arcane Updates

