

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_f18ea27f48b812813a6d6c1099b7ad0972beb22b6c532d1d253e6e89fb8f6f6be
      - 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.

Revision #7

Created 2026-08-18 22:25:32 UTC by dpasn

Updated 2026-09-28 10:14:10 UTC by dpasn