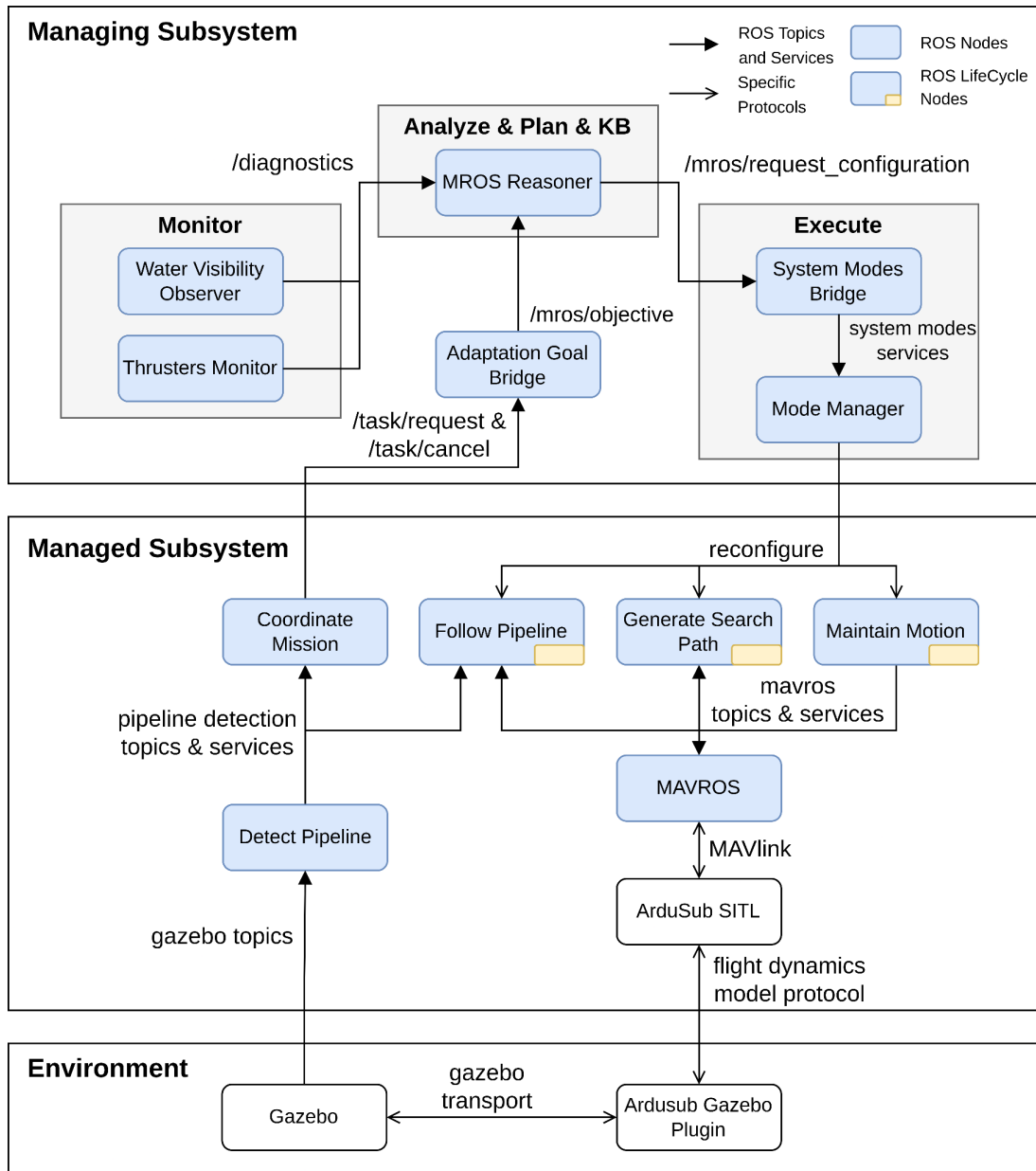


Quick Links

1. [System Overview](#)
2. [Use SUAVE with Docker](#)
3. [Install SUAVE locally](#)
4. [Run SUAVE](#)
5. [Extending SUAVE and connecting managing subsystems](#)

1. System overview



For more details, feel free to have a look over the [Github repository](#).

The exemplar can either be used with Docker or installed locally. The exemplar can be executed following these instructions.

1. Use Suave with Docker

You can pull and run the exemplar as a Docker container using the following command. Keep in mind you need to have [Docker](#) installed on your computer and running.

In a terminal on your computer run:

```
docker run -it --shm-size=512m -p 6901:6901 -e VNC_PW=password  
--security-opt seccomp=unconfined ghcr.io/kas-lab/suave:main
```

Optionally you can add the parameter:

`-v <absolute_path_host_compute>:/home/kasm-user/suave/results` to save the results into your computer, replace `<absolute_path_host_compute>` with the absolute path of where you want the data to be saved in your computer, e.g:

```
docker run -it --shm-size=512m -v  
$HOME/suave_results:/home/kasm-user/suave/results -p 6901:6901 -e  
VNC_PW=password --security-opt seccomp=unconfined  
ghcr.io/kas-lab/suave:main
```

Once the container is up and running, you can interface with it through your web browser. The container will be hosted locally at the port specified, in this case 6901. So in your browser, go to <http://localhost:6901>.

A dialog will request a username and password, these are shown below, with the password being specifiable in the run command.

- User : **kasm_user**
- Password: **password**

Now you can proceed to [run the exemplar](#).

Build Docker images locally

To build the docker images locally, run:

```
./build_docker_images.sh
```

2. Install Suave locally

To install the exemplar locally, you have to [install Gazebo Garden](#), [install ROS2 Humble](#), [install ArduSub](#), [install the ArduSub plugin](#), and finally [install the SUAVE workspace](#).

Install Gazebo Garden

Follow the [official instructions](#) for installing Gazebo Garden.

Install ROS2 Humble

Follow the [official instructions](#) for installing ROS2 Humble.

Install ArduSub

ArduSub is a subproject within ArduPilot for piloting underwater vehicles.

Disclaimer: Problems may occur with different combinations of ArduPilot and MavROS versions. The SUAVE repo was tested with [this ArduPilot commit](#) and [mavros 2.4.0](#). Unfortunately, at least at the time of writing this instruction manual, the releases available in Ubuntu 22.04 do not match.

```
cd ~/
git clone https://github.com/ArduPilot/ardupilot.git
cd ardupilot
git checkout e9f46b9
git submodule update --init --recursive
```

Note that the script used to install prerequisites available for this version of ArduSub does not work in Ubuntu 22.04. Therefore, you need to replace them before running ArduSub. To install the ArduPilot prerequisites, do the following.

```
cd ~/ardupilot
cd Tools/environment_install/
rm install-prereqs-ubuntu.sh
wget
http://raw.githubusercontent.com/ArduPilot/ardupilot/master/Tools/environment_
install/install-prereqs-ubuntu.sh
cd ~/ardupilot
chmod +x Tools/environment_install/install-prereqs-ubuntu.sh
```

```
Tools/environment_install/install-prereqs-ubuntu.sh -y
. ~/.profile
```

To test if the installation worked, run:

```
sim_vehicle.py -v ArduSub -L RATBeach --console --map
```

ArduPilot SITL should open and a console plus a map should appear.

Install the ArduSub plugin

Install the dependencies:

```
sudo apt install libgz-sim7-dev rapidjson-dev
```

Clone and build the repository:

```
cd ~/
git clone https://github.com/ArduPilot/ardupilot_gazebo
cd ardupilot_gazebo
mkdir build && cd build
cmake .. -DCMAKE_BUILD_TYPE=RelWithDebInfo
make -j4
```

Add the required paths:

Assuming that you have cloned the repository in `$HOME/ardupilot_gazebo`, run:

```
echo 'export
GZ_SIM_SYSTEM_PLUGIN_PATH=$HOME/ardupilot_gazebo/build:${GZ_SIM_SYSTEM_PLUGIN_PATH}' >> ~/.bashrc
echo 'export
GZ_SIM_RESOURCE_PATH=$HOME/ardupilot_gazebo/models:$HOME/ardupilot_gazebo/worlds:${GZ_SIM_RESOURCE_PATH}' >> ~/.bashrc
```

Now that new environmental variables have been added to your terminal, you need to reload it with:

```
source ~/.bashrc
```

More info about the plugin can be found in the corresponding [repository](#).

Install the SUAVE workspace

Create the workspace and download the required repositories:

```
mkdir -p ~/suave_ws/src/
cd ~/suave_ws/
```

If you want to get the most updated version of the repo:

```
wget https://raw.githubusercontent.com/kas-lab/suave/main/suave.rosinstall
vcs import src < suave.rosinstall --recursive
```

SEAMS2023: If you want to get the version submitted to SEAMS 2023 instead of the most updated version get the following dependencies instead:

```
wget
https://raw.githubusercontent.com/kas-lab/suave/9e6468896ce766376557ca9522d
84f92b70129f1/suave.rosinstall
vcs import src < suave.rosinstall --recursive
```

Before building the ros_gz package (one of the dependencies), you need to export the gazebo version:

```
export GZ_VERSION="garden"
```

You can also add this to your `~/.bashrc` to make this process easier.

Install the dependencies:

```
source /opt/ros/humble/setup.bash
cd ~/suave_ws/
rosdep install --from-paths src --ignore-src -r -y
```

Build the project:

```
cd ~/suave_ws/
colcon build --symlink-install
```

If you have memory problems while building the package, run the following command instead, it is slower but uses less memory:

```
colcon build --symlink-install --executor sequential --parallel-workers 1
```

Install a MAVROS dependency:

```
wget
https://raw.githubusercontent.com/mavlink/mavros/master/mavros/scripts/install_geographiclib_datasets.sh
sudo bash ./install_geographiclib_datasets.sh
```

Now you can proceed to [run the exemplar](#).

3.Run Suave