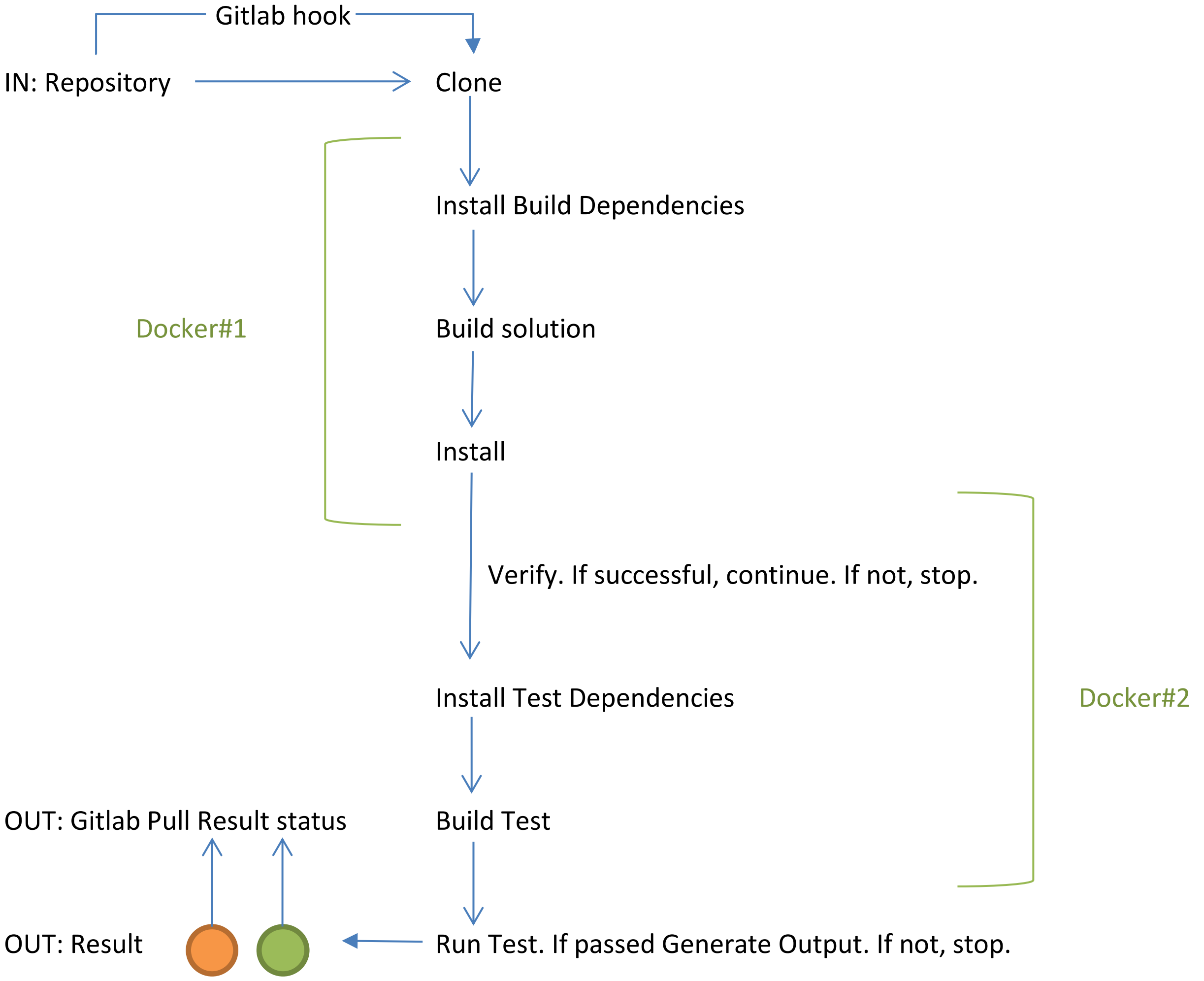
Development Jobs


Gitlab hook
IN: Repository
Clone
Install Build Dependencies
Docker#1
Build solution
Install
Verify. If successful, continue. If not, stop.
Install Test Dependencies
Docker#2
OUT: Gitlab Pull Result status
Build Test
OUT: Result
Run Test. If passed Generate Output. If not, stop.

Building

- Install all dependencies using binary packages
- Start each job with a clean state since the all have different dependencies
- Setup inside a Docker Container so the job is isolated. Use for this catkin_make_isolated

Testing

- Exercising code paths by executing test cases
- Automated unit tests
- .bag files
- Environment simulations:

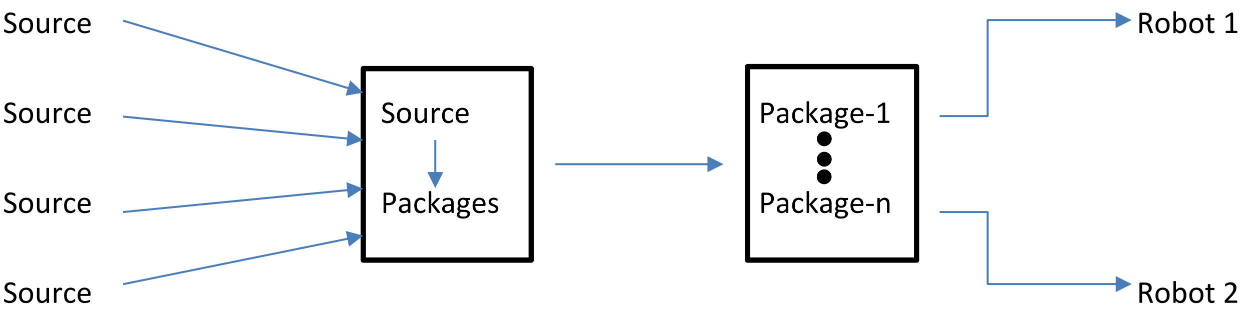
When making changes, always test before continuing. Once passed, continuous Integration

Deployment

- Methods installing robot software in “production” systems
- Deploying from source
  - Build everything on your developer machine using catkin
  - install it on your developer machine
  - Synch that over to the vehicle.
  - → Easy to keep track of your changes and to coordinate between multiple people.

- Deployment from single debian package
  - Take all your software
  - Put it on a build server or just on your own machine
  - Put it in to a single debian
  - → easy to track released versions

- Deployment from Apt Repository



- Deployment with Docker for build environments
  - It’s good for running a service and minimally exposing it
  - You have to run your entire ROS application, every node inside the same Docker container, otherwise setting up your network configuration will defeat the purpose of Docker

To make sure your system is doing fine you need to build, test and deploy. When everything ready, move to production.