

## Using the deploy.sh script

The "deploy.sh" script has been added to allow for rapid and easy development for physical teams and judges alike! This script is found in the SwarmBaseCode-ROS/misc folder and automates several key tasks for the user! Now connecting to swarmies takes seconds.

The script is run from your workstation and not the swarmies themselves. Keeping work on the workstation has many benefits with one being a fast and reliable way to transfer and run code. You will need to ensure the GUI is running before running this script!

Before running the code, navigate to the misc folder:

```
cd ~/SwarmBaseCode-ROS/misc
```

Give permission to the script to be executable:

```
chmod +x deploy.sh
```

You are now set for rapid deployment and development!

"deploy.sh" has 3 built-in options to be used:

```
./deploy.sh -R
```

- -R will ask the user for which rovers they wish to connect with and start sending information back to the workstation GUI

```
./deploy.sh -L
```

- -L will compile and compress the local repository from which the script was run. Then the user is prompted for a list of rovers that will receive the repository. The code is unpacked on the specified rovers and the nodes started.

- ex.) `./deploy.sh -L` means the local repo is compiled and compressed
- Rover Name/IP To Start: `R17 R18 R19` means the packaged local repo is sent to R17 R18 R19 and then those nodes are started
- If changes are made to the local repo they will not take affect as part of the transfer until the script is made aware of the changes. This is done by using the `-RC` command in place of a target rover id:
  - Typing `'-RC'` recompiles the code base the user is currently using to deploy to a swarmie and repackages it for transfer

`deploy.sh -G {branch}` (where branch is the desired branch you wish to pull)

- `-G` requires the branch users wish to pull from. This allows users to choose different branches for testing. This will then follow a similar logic to `-L` and begin sending information back to the workstation GUI. Like `-L` this has unique built-in options
  - Typing `'-NB'` will allow users to get a new branch at anytime
  - Typing `'-RP'` will allow users to re-pull from your current selected github branch

If you want to use the script, and don't want to use the interface, you are able to do so. Running any option (`-R`, `-L`, `-G`) followed by a `-S` will trigger a silent mode for the command, running your commands and returning to the terminal that the script was called from immediately after all operations are performed.

EX: `./deploy.sh -R -S robot1` will attempt to connect and run robot1 and then return to the current terminal without opening the interface.

Feature:

Typing "REBOOT {hostname}" in any option will allow you to reboot the selected rover.

- If changes been made to the password for a swarmie, users will need to change the password in the script file as well in the variable "roverPass" to allow it to work!

Running multiple commands at once is allowed. So typing in a line such as "rover1 rover2 REBOOT rover3" will work.

## NOTES FOR DEPLOY.SH SCRIPT: Applying Keys

This script runs better when using ssh-keys. Keys allow you to SSH without requiring the user to type in a password every time.

- Follow this guide to learn about using keys: <https://www.ssh.com/ssh/copy-id>

If unfamiliar or have not setup an SSH-Key on a current machine, users can type: `ssh-keygen` and follow the prompt

Once the key has been setup, copy the key from the users machine to each rover you wish to add it to with `ssh-copy-id swarmie@{hostname}` where hostname is the rover's hostname

That's it! You should now have a seamless way to SSH without having to type in passwords each time!