

# QTL as a Service

---

Project webpage: <http://www.it.uu.se/research/project/ctrait/QLaaS>

## Summary

---

We have developed QTL as a Service (QLaaS) using PruneDIRECT algorithm. QLaaS automatically deploys an R cluster for using PruneDIRECT, or any statistical analysis in R, over your desired infrastructure.

Three files are required for this method: `ansible_install.sh`, `setup_var.yml`, `spark_deployment.yml`

Note: Following commands have been tested on Ubuntu 16.04.

0. Step 0: `python` command is required to be available on each node. If it is not available install with `# apt install python-minimal`
1. Step 1: Install Ansible using the bash script, `ansible_install.sh`.
2. Step 2: Modify the environment variables available in the file: `setup_var.yml`, if needed.
3. Step 3: For setup deployment, execute: `spark_deployment.yml` as root which is the actual file that contains the installation setups for all the components of QLaaS platform. Command: `# ansible-playbook -s spark_deployment.yml`, where `-s` is the sudo flag.

We will soon provide a demo through our project webpage using the SNIC cloud resources. The users can try QLaaS over a few nodes in our cloud setting. For larger computation, one can download QLaaS from the github repository and deploy the desired number of nodes over an infrastructure.

## Setup details

---

1. Setup at least 3 nodes, one for the Ansible Master, one for the Spark Master, and at least one for the Spark Worker.
2. Install Ansible on Ansible Master node by executing the script: `ansible_install.sh`. The script requires super-user privilege.
3. Add the IP-address and hostname of the Ansible Master, Spark Master and Spark Worker to `/etc/hosts` file in Ansible Master node.
4. Generate a SSH-key pair in Ansible Master node and copy its public part to `~/.ssh/authorized_keys` in all the Spark nodes. This step allows Ansible Master to communicate with Spark nodes.

For more information on ansible communication setup, visit: <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-16-04> (Step-1 - 4)

5. Edit `/etc/ansible/hosts` using `example-hosts-file` available in the repository. (Add `[sparkmaster]` followed by the name of sparkmaster node in the next line. Add `[sparkworker]` followed by the names of sparkworkers in the next lines, one per line).
6. Modify the environment variables available in the file `setup_var.yml`, if needed.
7. Run `ansible-playbook -s spark_deployment.yml`, where `-s` is the sudo flag.
8. Make sure the following ports are open on Spark Master node, `60060` for Jupyter Hub (external access), `7077` Spark Context (internal access), `8080` Spark Web UI (internal access).
9. Jupyter server tokens will be visible in ansible log messages.

10. Now you can access following services: `http://<sparkmaster-external-IP>:60060`

11. Execute the steps mentioned in `example-sparkR` file to make sure your setup is working.

After all the steps above, Jupiter, Spark Master and R will be installed in Spark Master, and Spark Worker and R is installed in all Spark Workers.

## (Optional) Add more node(s)

---

Here are the steps to add new nodes to your already configured cluster:

1. New node(s) should be accessible from the Ansible Master node (repeat steps (3,4 and 5) mentioned in the "Setup details" section).
2. Run the ansible playbook again. `ansible-playbook -s spark_deployment.yml .`