

USER GUIDE

**Testgrid User Documentation**

Arkena  
27, bld Hippolyte Marquès  
94200 Ivry sur Seine  
FRANCE

Last Update: 08.2014

Table of Contents

[What is Testgrid 1](#_Toc396426155)

[Schema & Use Case 1](#_Toc396426156)

[Configuration File 3](#_Toc396426157)

[Command line tool explanation 3](#_Toc396426158)

[Ansible Integration 6](#_Toc396426159)

[External Inventory Script 6](#_Toc396426160)

[Launch ansible-playbook through Testgrid 7](#_Toc396426161)

[Nodes type specification 9](#_Toc396426162)

[Vagrant 9](#_Toc396426163)

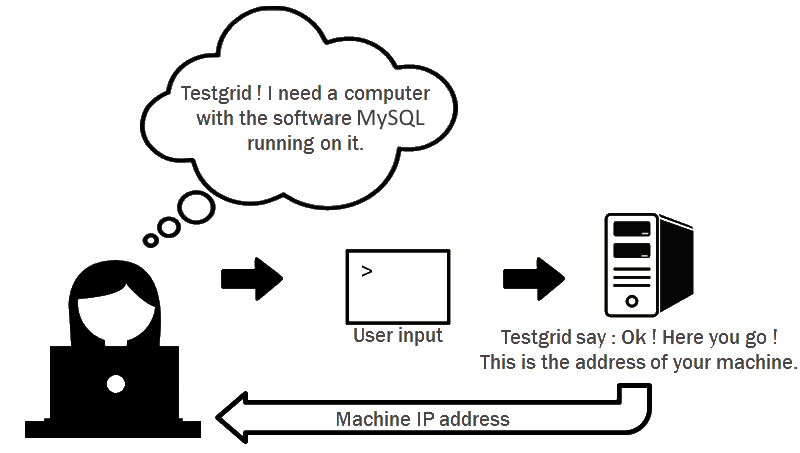
[*Installsystems* 9](#_Toc396426164)

## What is Testgrid

Testgrid is a software that will give the ability to deploy an indefinite number of on-demand and programmable test environment quickly, with customizable requirements. The Testgrid has to provide flexibility.

## Schema & Use Case

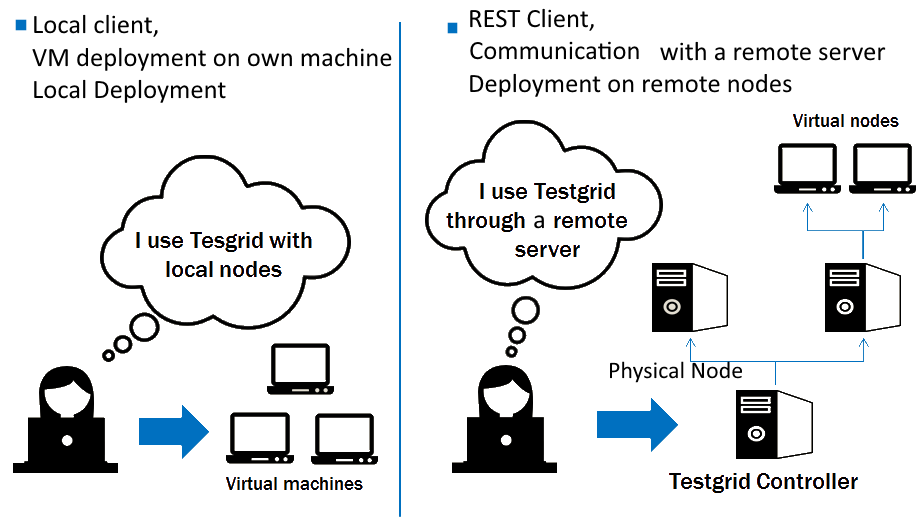
#### Basic Use Case



There is two ways for the user to interact with Testgrid:

* Directly launch command from a terminal.
* Use the Python API to interact with Testgrid.

#### Two interactions possible



## Configuration File

There are two ways to configure a file: Local & Rest mode

Rest client:

|  |
| --- |
| [controller]  host=testgrid.controller.com  port=80 |

Local client grid sample using Vagrant:

|  |
| --- |
| [grid]  type = vagrant grid  path = /Users/user/Documents/work/boxes |

Those files could be used to manage virtual machine using Vagrant on your own computer.

## Command line tool explanation

Following, the list command that provides all the information stored in Testgrid:

#### Basic Commands

|  |
| --- |
| > testgrid --list-nodes  name   type     status  allocation user  session -----  ---------------------  ----------- ---------- ----- ------- node1  remote node/wheezy64  up allocated root  test  node2  remote node/wheezy64  up allocated root  test  node3  remote node/wheezy64  unreachable allocated root test  node4  remote node/wheezy64  unreachable quarantine node5  remote node/wheezy64  up available |

|  |
| --- |
| > tg --open-session test  > tg --list-sessions  username  name  --------  ----  root   test |

#### Deploy a node Sample

|  |
| --- |
| > tg --list-nodes  name type status  allocation username session  ---- -------------------- ------  ---------- -------- -------  node1  remote node/wheezy64  up   available   -      -  node2  remote node/wheezy64  up   available   -      -  node3  remote node/wheezy64  up   available   -      - |

|  |
| --- |
| > tg –s test –deploy –deb sqlite3 fleche mysql  package sqlite3 installed on node node1  package fleche installed on node node2  package mysql installed on node node3  > tg --list-nodes  name type status  allocation username session  ---- -------------------- ------  ---------- -------- -------  node1  remote node/wheezy64  up allocated root     test  node2  remote node/wheezy64  up allocated root     test  node3  remote node/wheezy64  up allocated root     test |

The “release-node” command release the specified nodes.

|  |
| --- |
| > tg –s test –release-node node1  > tg --list-nodes  name type status  allocation username session  ---- -------------------- ------  ---------- -------- -------  node1  remote node/wheezy64 up available   - -  node2  remote node/wheezy64 up allocated root     test  node3  remote node/wheezy64 up allocated root     test |

The “Undeploy” command release all the nodes for a specific session.

|  |
| --- |
| > tg –s test –undeploy  > tg --list-nodes  name type status  allocation username session  ---- -------------------- ------  ---------- -------- -------  node1  remote node/wheezy64  up available   - -  node2  remote node/wheezy64  up available   -      -  node3  remote node/wheezy64  up available   -  - |
|  |

#### Node Allocation

Add node specification in a .ini file:

|  |
| --- |
| > cat /etc/tg/rest-client.ini  [controller]  host=testgrid.controller.com  port=80  [alloc]  sysname = wheezy64  > tg –s test-alloc --allocate-node alloc  > tg –s test-alloc –list-nodes  name type status  allocation username session  ---- -------------------- ------  ---------- -------- ----------  node1  remote node/wheezy64  up allocated root test-alloc |

|  |
| --- |
| > tg –s test-alloc –n node1 –install –deb sqlite3 |

## Ansible Integration

This part requires to be familiar with Ansible.

Testgrid can be used to generate a group of host as an inventory for Ansible.

### External Inventory Script

|  |
| --- |
| > cat node-description.ini  [ansible-role1]  Sysname= wheezy64  [ansible-role2]  Sysname = wheezy64  [session-inventory]  nodes = ansible-role1, ansible-role2  > cat inventory-file  ansible-role1  ansible-role2    [group1]  ansible-role1  [group2]  ansible-role2 |

|  |
| --- |
| > tg -s session-inventory --inventory path-to-inventory-file --session-manifest node-description.ini  Session-inventory.py will be generated  > ansible-playbook -i session-inventory.py /path/to/yml/site.yml  > tg –s session-inventory –list-nodes  name type status allocation username session  ------------- -------- ------  ---------- -------- --------  ansible-role1 wheezy64 up allocated root session-inventory  ansible-role2 wheezy64 up allocated root session-inventory |

## Launch ansible-playbook through Testgrid

Store Ansible deployment in web server. Package with yml and inventory file must be installed on the machine that runs the Testgrid controller or the machine that run Testgrid in local mode.

|  |
| --- |
| > cat /etc/tg/web-server.ini  [tg-web-server]  host = 10.22.1.1  port = 80  ssh-key = /home/tg-user/id-rsa  > cat my\_pkg.ini  [ansible-role-1]  image\_name = squeeze  [ansible-role-2]  image\_name = wheezy  [ansible-role-3]  image\_name = squeeze  [ansible-role-4]  image\_name = wheezy  [my-pkg]  inventory\_path=/path/to/inventory/file  playbook\_path=/path/to/yml  nodes= ansible-role-1 ansible-role-2 ansible-role-3 ansible-role-4  > cat inventory-file  ansible-role1  ansible-role2  ansible-role3  ansible-role4    [group1]  ansible-role1  ansible-role2  [group2]  ansible-role2  ansible-role4 |

|  |
| --- |
| > tg -s test --ans-playbook my\_pkg  > tg –s test –list-nodes  name type status allocation username session  ------------- -------- ------  ---------- -------- --------  ansible-role1 **squeeze** up allocated root test  ansible-role2 wheezy64 up allocated root test  ansible-role3 **squeeze** up allocated root test  ansible-role4 wheezy64 up allocated root test |

## Nodes type specification

### Vagrant

|  |
| --- |
| [vagrant-centos] box = "centos63-64"  [vagrant-squeeze] box = "squeeze64"  [vagrant-wheezy] box = "wheezy64" |

## Installsystems

Testgrid can deploy virtual machines using installsystems.

Link to installsystem project:

<https://github.com/seblu/installsystems>