VERSION 2.6
DATE Mai 2011

Table des matières

Chapitre 1

Deploying a DIET platform

Deployment is the process of launching a DIET platform including agents and servers. For DIET, this process includes writing configuration files for each element and launching the elements in the correct hierarchical order. There are three primary ways to deploy DIET.

Launching **by hand** is a reasonable way to deploy DIET for small-scale testing and verification. This chapter explains the necessary services, how to write DIET configuration files, and in what order DIET elements should be launched. See Section **??** for details.

GODIET is a Java-based tool for automatic DIET deployment that manages configuration file creation, staging of files, launch of elements, monitoring and reporting on launch success, and process cleanup when the DIET deployment is no longer needed. See Section 1.1 for details.

Writing your own scripts is a surprisingly popular approach. This approach often looks easy initially, but can sometimes take much, much longer than you predict as there are many complexities to manage. Learn GODIET– it will save you time !

1.1 GODIET

GODIET is an cross-platform tool that helps you automate ad-hoc deployment and management procedures for DIET infrastructure. It manages configuration file creation, staging of files, launch of software components, monitoring and reporting. GODIET is extremely useful for large deployments on a complex physical infrastructure. The mains features are :

- complete command line interface ;
- distributed command execution via SSH ;
- real time monitoring applications state ;
- complex physical infrastructure management with firewall and multiple network lan.

1.1.1 Installing GODIET

The following operating systems are known to support GoDiet :

- Linux : most recent distributions are likely to work ;
- Mac OS X 10.4 or later.

You need to have the Sun Java 6 or OpenJDK6 installed. Download GODIET on the project website[1]. Check that run.sh script works. You must have something on the figure1.3

1. http ://graal.ens-lyon.fr/DIET/godiet.html

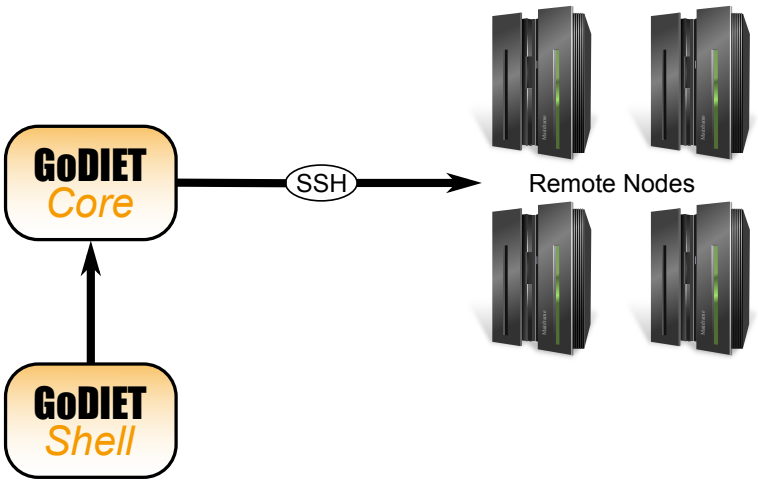

FIGURE 1.1 – Design principle of GoDIET.

1.1.2 Quickstart

The four steps for hurry people

- create your GoDIET configuration file in `${HOME}/.godiet/configuration.xml`1.1.3. It contains remote connection informations ;
- create your infrastructure description file1.1.3. It describes your computes nodes, gateways and storages ;
- create your DIET platform description file1.1.3. It describes all the DIET elements (agents and seds) that you want to deploy and manage ;
- Run GoDIET shell??.

That's all folks !

1.1.3 GoDIET setup

Before using GoDIET, you need to create three files : one to describe GoDIET configuration, one which describe your infrastructure and one which contains your DIET description. This files uses an XML-format that each conforms repectively to the Configuration.xsd, Infrastructure.xsd and Diet.xsd grammar (provide with GoDIET). Sample files are provided in the `examples` directory.

Configuration

This file aggregate informations about the local node from where GoDIET is launch. It contains information about user authentification too. GoDIET looks in the `${HOME}/.godiet/configuration.xml` directory by default.

The mains elements are :

- **localNode** : the node's name from where godiet is executed. This name must be present the infrastructure description ;
- **localscratch** : The working directory where GoDIET store his own temporary files ;

- **keys** : The paths of yours private ssh keys which are loaded at GoDiet startup. You can give the public key path too. GoDiet try to load a file with the same name than the private key ended by `.pub`. See `ssh initkeys` command1.1.4 to initialize passwords if your keys are encrypted (i.e need a passphrase).

General configuration description layout (some parts are omitted) :

```
<configuration schema="Configuration.xsd">
<goDietConfiguration localNode="local">
<localscratch dir="/tmp/scratch_godiet" />
  <user>
     <ssh>
       <key path="/home/.ssh/id_dsa"/>
       <key path="/home/.ssh/admin_cluster2"/>
     </ssh>
  </user>
</goDietConfiguration>
```

Infrastructure

GoDiet needs to have the description of infrastructure Diet will be running.

You can find full options list in `Infrastructure.xsd` grammar file. You can look in the `examples` directory too. The most important fields are :

- **Domain** : aggregate a set of infrastructure elements (nodes, gateways, storages) which are be able to communicate with Diet's exchange protocol (i.e CORBA). Typically elements separated with firewall and/or a router must be describe in separate domains.
- **Storage** : Describe a remote disk access.
- **Node** : Decribe a computing node where agents or seds will be running
- **Gateway** : Describe an access point for a domain. The link or Elle permette de créer les interconnections entre les différents domaines.
- **Link** : Définit un lien directionel par lequel deux passerelles pourront communiquer grâce au protocole SSH.

General infrastructure description layout (details are omitted) :

```
<infrastructure schema="Infrastructure.xsd">
 <domain id="idDomain1">
 ..
    <storage id="idStorage">..</storage>
    <node id="idNode" storageRefId="idStorage">..</node>
    <gateway id="idGateway1">..</gateway>

 </domain id="idDomain2">
```

```
 ..
   <gateway id="idGateway2"/>
 ..
 </domain>
 <link from="idGateway1" to="idGateway2"/>
</infrastructure>
```

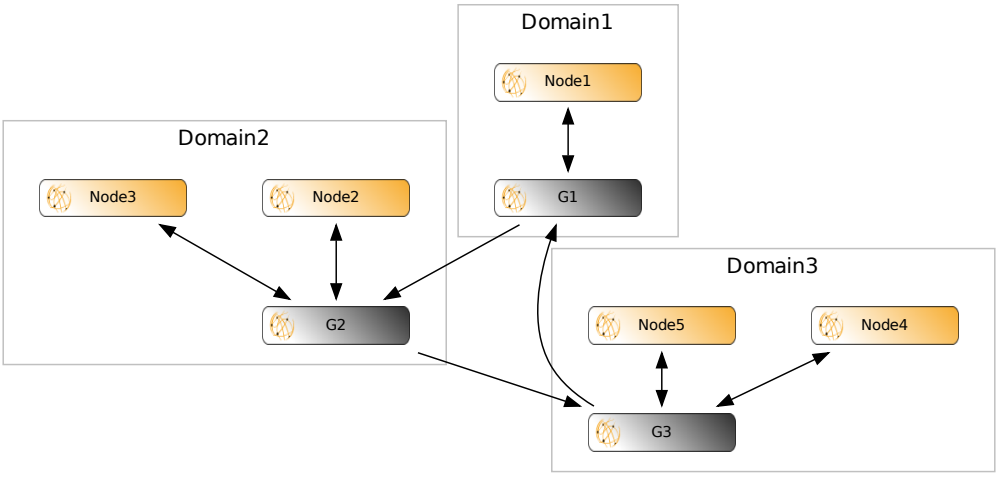

FIGURE 1.2 – Three domains infrastructure representation example

Diet platform

The user of GoDiet could describes the desired deployment in an XML file including all needed external services (*e.g.*, omniNames and LogService) ; the desired hierarchical organization of agents and servers is expressed directly using the hierarchical organization of XML.

1.1.4 Godiet shell

GoDiet shell is the interface to manage your Diet platform. It includes facilities features like syntax highlighting, command completions and history commands. After executing run script you will have a prompt like on the figure 1.3

Help command

SSH command

- **initpasswords** : Allow you to intialize ssh key's passphrase loaded from configuration file ;
- **addkey** : Register a new key ;

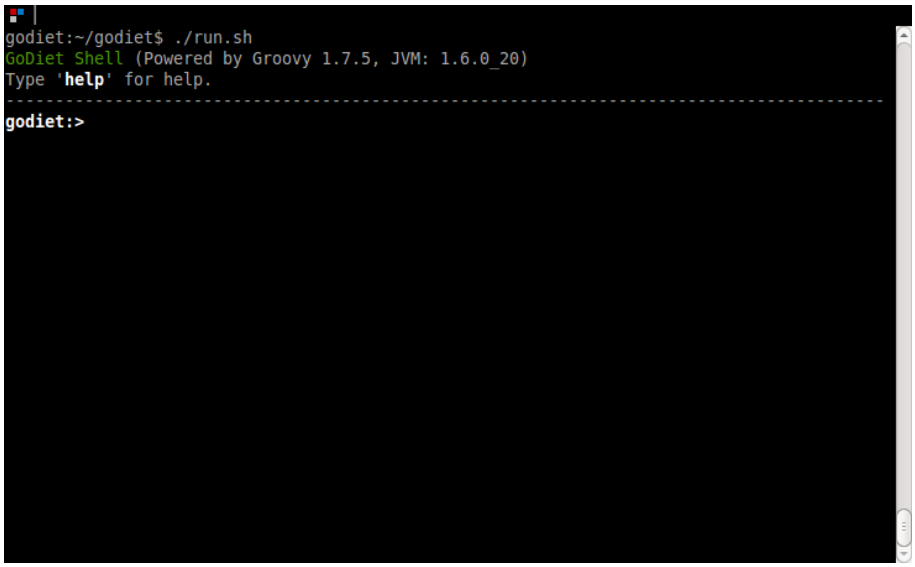

FIGURE 1.3 – GODIET shell startup.

- **modifykey n** : Modify a key that already registred ;
- **status** : Display the keys status. Could be PASSWORDNOTSET,PRIVATEKEYERROR,PUBKEYERR
  or LOADED.

Load commands

Use to load an infrastructure description1.1.3 LoadInfrastructure permet de charger un fichier de description d'infrastructure

calcule et création automatique des Forwarders

Start & Stop commands

`start` command Start command options :

- **software** name : Start one software given his name.
- **services** : Start all services
- **agents** : Start all agents
- **seds** : Start all seds
- **all** : Start all softwares components

`stop` command have same parameters.

Status command

Status command options :

- **ma** : Display the masters agents status
- **la** : Display the locals agents status
- **seds** : Display the servers daemons status
- **all** : Display all diet softwares status

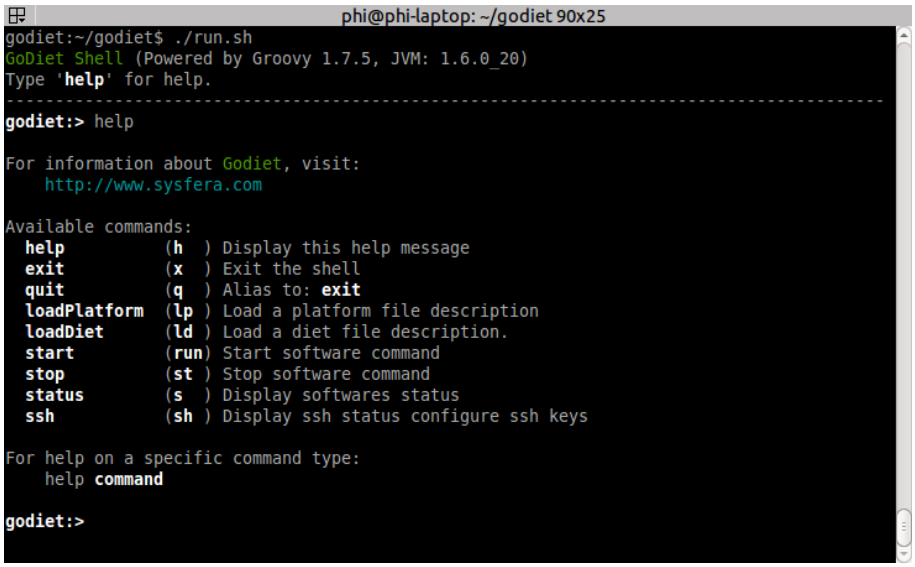

FIGURE 1.4 – The help command.

La commande status affiche sous forme de tableau l'état des éléments gérés pas GoDiet. Un exemple d'execution est affiché sur la figure 1.5. De gauche à droite les informations affichés sont :

- Le nom du logiciel comme décrit dans le fichier de description d'infrastructure.
- Son état et depuis quand il s'y trouve.
- La machine sur laquelle le logiciel va être executé ou est actuellement executé (selon son état).
- Le message d'information dans le cas où la ressource est dans l'état erreur.

Les différents états possibles sont

- **Incubate** : The component is correctly loaded in GoDiet.
- **Ready** : The configurations file are created in the local scratch directory and component is ready to start.
- **Up** : The component is Up. His state is periodically checked.
- **Down** : The componenent is Down. Typically if GoDiet user call stop on this component.
- **Error** : Could appears if there is an error in the description, if the

Debugging & Gestion des erreurs

parler des fichiers de log et des messages d'erreurs.

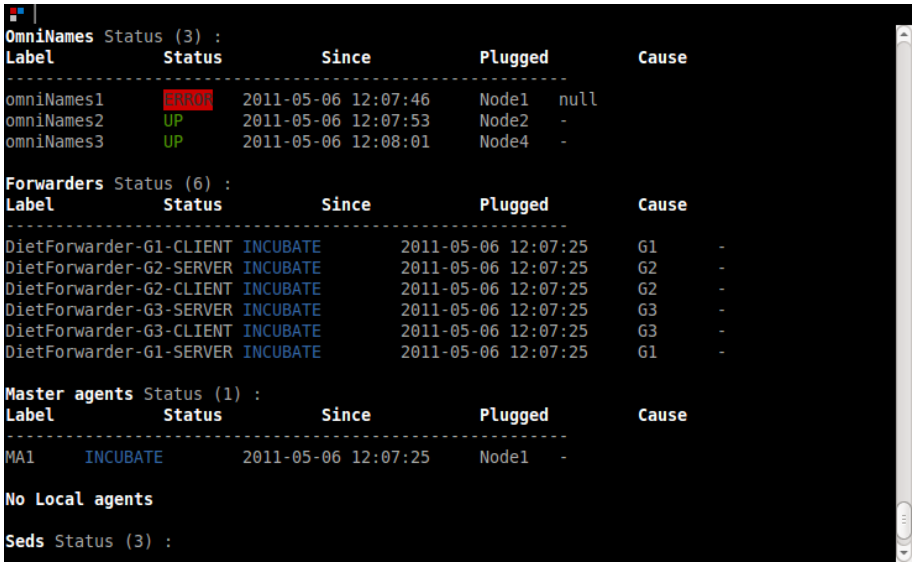

FIGURE 1.5 – The 'status all' command.