
SGAD



protech.unipd@gmail.com

MANUALE DI INSTALLAZIONE V1.00

Nome del documento	Manuale di Installazione
Versione del documento	1.00
Data redazione	2014/03/10
Redazione	Biancucci Maurizio
Verifica	Gallo Francesco
Approvazione	Segantin Fabio
Uso	Esterno
Distribuzione	ProTech

Sommario

Questo documento presenta l'elenco dei comandi che sarà necessario eseguire per installare correttamente il software Sgad.

Diario delle modifiche

Modifica	Autore	Ruolo	Data	Versione
<i>Approvazione del documento</i>	Segantin Fabio	Project Manager	2014/03/13	v 1.00
<i>Verifica del documento</i>	Gallo Francesco	Verificatore	2014/03/12	v 0.03
<i>Completamento della stesura del documento</i>	Biancucci Maurizio	Amministratore	2014/03/12	v 0.02
<i>Inizio stesura documento con impostazione generale dello scheletro del documento</i>	Biancucci Maurizio	Amministratore	2014/03/10	v 0.01

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Glossario	3
1.3	Riferimenti	3
1.3.1	Normativi	3
1.3.2	Informativi	3
2	Requisiti minimi	4
3	Installazione delle dipendenze	5
3.1	Java 7	5
3.2	MongoDB versione 2.4.9	5
3.3	SBT versione 0.13.1	5
4	Installazione di SGAD	7
4.1	Ottenere il prodotto	7
4.2	Configurare MongoDB con i dati di gioco	7
4.3	Configurare il cluster e Spray	9
4.4	Configurare il server HTTP	10
4.5	Configurare il database	11
5	Avvio	13

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di indicare e spiegare quali sono i comandi da eseguire per installare correttamente il software Sgad su un *sistema operativo*_{|g|} *Linux*_{|g|} *Ubuntu*_{|g|} 12.04.

1.2 Glossario

Per evitare ogni ambiguità relativa al linguaggio e ai termini utilizzati nei documenti formali, si allega il “*Glossario v2.00*”. In questo documento vengono definiti e descritti tutti i termini con un significato particolare. Per rendere più facile la lettura, i termini saranno posti in corsivo e accanto a questi ci sarà una ‘g’ corsiva, compresa tra barre verticali, a pedice (esempio: *Glossario*_{|g|}).

1.3 Riferimenti

1.3.1 Normativi

- **Norme di Progetto:** “*Norme di Progetto v5.00*”;

1.3.2 Informativi

- GNU Screen manuale utente: <http://www.gnu.org/software/screen/manual/screen.html>
- Git manuale utente: <https://www.kernel.org/pub/software/scm/git/docs/user-manual.html>
- MongoDB manuale installazione: <https://www.kernel.org/pub/software/scm/git/docs/user-manual.html>
- SBT manuale installazione: <http://www.scala-sbt.org/release/docs/Getting-Started/Setup.html>

2 Requisiti minimi

La parte del *prodotto_[g]* lato server richiede come requisiti minimi per funzionare:

- sistema operativo Linux Ubuntu x64 12.04 o superiori;
- spazio libero su disco rigido di almeno 1GB;
- 1GB di memoria RAM totale;
- 384MB di memoria RAM disponibile;
- processore core singolo da almeno 2.0GHz.

3 Installazione delle dipendenze

I seguenti comandi sono da utilizzare in una shell di Linux.

3.1 Java 7

Il software richiede *Java*_{|g|} 7 per il funzionamento. Si assicura il corretto funzionamento dalla sotto release 51. Di seguito è presente l'elenco di comandi per installare automaticamente l'ultima versione di Java 7:

1. `sudo apt-get install python-software-properties`
2. `sudo add-apt-repository ppa:webupd8team/java`
3. `sudo apt-get update`
4. `sudo apt-get install oracle-java7-installer`

3.2 MongoDB versione 2.4.9

Il prodotto software richiede, per funzionare, un'istanza attiva e configurata correttamente di *MongoDB*_{|g|} versione 2.4.9. Non è necessario installare tale software sulla stessa macchina in cui si intende eseguire l'applicativo lato server. Alla sezione 4.5 è spiegato come configurare l'indirizzo in cui è presente l'istanza di MongoDB che si desidera utilizzare. Di seguito è presente l'elenco di comandi per installare la versione corretta di MongoDB automaticamente:

1. `sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10`
2. `echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen'|
sudo tee /etc/apt/sources.list.d/mongodb.list`
3. `sudo apt-get update`
4. `sudo apt-get install mongodb-10gen`
5. Riavviare il sistema.

Assicurarsi a questo punto che MongoDB sia installato correttamente provando ad entrare nella shell di amministrazione dello stesso digitando il comando: `mongo`. Se la shell di MongoDB non si avvia correttamente eseguire il comando: `sudo mongod --dbpath /var/lib/mongodb --smallfiles` per risolvere il problema. Per uscire dalla shell di MongoDB basta digitare il comando `exit` oppure premere CTRL+C.

3.3 SBT versione 0.13.1

Il codice lato server necessita del build tool SBT (Simple Build Tool) versione 0.13.1, esso si occuperà della compilazione e dell'avvio dell'applicativo lato server. È possibile installare la versione 0.13.1 tramite i seguenti comandi:

1. `sudo apt-get install gdebi`

3 Installazione delle dipendenze

2. `wget http://repo.scala-sbt.org/scalasbt/sbt-native-packages/org/scala-sbt/sbt/0.13.1/sbt.deb`

3. `sudo gdebi sbt.deb`

È possibile avviare SBT tramite il comando `sbt`. Per uscire dalla shell di SBT basta digitare il comando `exit` oppure premere CTRL+C.

4 Installazione di SGAD

Dopo essersi assicurati di soddisfare tutti i prerequisiti è possibile procedere con l'installazione del prodotto SGAD.

4.1 Ottenere il prodotto

Il prodotto è disponibile gratuitamente sotto Licenza Apache v2. Esso è pubblicato su *GitHub*_[g] e disponibile al seguente indirizzo: <https://github.com/protechunipd/SGAD>. Un modo per ottenere il prodotto è il seguente:

1. navigare sul terminale fino alla cartella in cui si desidera installare il prodotto;
2. `git clone https://github.com/protechunipd/SGAD.git` - provvederà al download del prodotto nella cartella corrente.

Altrimenti sarà possibile scaricare il prodotto tramite *browser*_[g] web accedendo all'indirizzo fornito precedentemente.

4.2 Configurare MongoDB con i dati di gioco

Il prodotto richiede, per motivi di sicurezza, che MongoDB richieda l'autenticazione alla connessione. Per assicurare questo, bisogna aprire con permessi di root il file “*/etc/mongod.conf*” e de commentare rimuovendo il carattere ‘#’ dalla riga `#auth = true`. A questo punto, per completare l'operazione, va riavviato il sistema.

Attivata l'autenticazione, se non già presente¹, è necessario creare un utente con poteri sull'intera istanza di MongoDB. Dal momento in cui l'utente amministratore verrà creato, non sarà più possibile compiere operazioni senza autenticazione. Per creare l'utente è necessario avviare una shell di MongoDB con il comando `mongo` ed eseguire i seguenti comandi:

1. `use admin` - permette il passaggio all'utilizzo del *database*_[g] di amministrazione.
2. `db.addUser({ user: 'Admin',
pwd: 'passwordAScelta',
roles: ['userAdminAnyDatabase'] })` - crea un nuovo utente amministratore. Personalizzare il comando sostituendo la stringa “**passwordAScelta**” con la password che si intende utilizzare.

A questo punto è necessario configurare MongoDB con un utente che gestirà il database con i dati di gioco. È possibile farlo avviando una shell di MongoDB con il comando `mongo` ed eseguendo i seguenti comandi:

1. `use sgaddb` - permette il passaggio all'utilizzo del database del gioco. Essendo la prima volta che si passa all'utilizzo di questo database, esso verrà creato.

¹Se si sta lavorando con un'installazione pulita di MongoDB questo non è presente.

2. `db.addUser({ user: 'nomeUser',
pwd: 'passwordAScelta',
roles: ['readWrite', 'dbAdmin'] })` - crea un nuovo utente amministratore del database di gioco. Personalizzare il comando sostituendo la stringa “**nomeUser**” con il nome desiderato e la stringa “**passwordAScelta**” con la password che si intende utilizzare.

Adesso è necessario assicurare che gli username e le email di registrazione degli utenti rimangano univoche all'interno del database. È possibile farlo tramite i seguenti comandi eseguiti nella shell di Mongo:

1. `use sgaddb` - permette il passaggio all'utilizzo del database di del gioco.
2. `db.auth('user', 'password')` - permette di autenticarsi come l'utente amministratore del database di gioco. Personalizzare il comando sostituendo alle stringhe “**user**” e “**password**” l'username e la password dell'utente amministratore del database di gioco.
3. `db.userdata.ensureIndex({ 'authenticationData.user': 1 }, { unique: true })` - assicura l'unicità della username di registrazione degli utenti.
4. `db.userdata.ensureIndex({ 'authenticationData.email': 1 }, { unique: true })` - assicura l'unicità della email di registrazione degli utenti.

Ora bisogna inserire i dati generali di gioco, essi sono presenti nel file “*Codice/sgad/severtier/src/main/resources/gamedata/gameDataOfficial.js*” a partire dalla cartella radice del prodotto. Questo file contiene tutti i dati di gioco eventualmente personalizzabili. La procedura di inserimento di questi dati è la seguente:

1. `use sgaddb` - permette il passaggio all'utilizzo del database di gioco.
2. `db.auth('user', 'password')` - permette di autenticarsi come l'utente amministratore del database di gioco. Personalizzare il comando sostituendo alle stringhe “**user**” e “**password**” l'username e la password dell'utente amministratore del database di gioco.
3. `load('file.js')` - personalizzare il comando sostituendo a “**file.js**” il suddetto file inserendo il percorso assoluto oppure il percorso relativo alla posizione dalla quale si è fatta partire la shell di MongoDB.

Se si modificano i dati di gioco editando il file che li contiene, per aggiornarli nel database è sufficiente rieseguire la procedura appena descritta.

È presente anche il seguente file “*Codice/sgad/severtier/src/main/resources/game-data/gameDataTest.js*” contenente dati di gioco adatti a test, caratterizzati da tempi di attesa estremamente ridotti. Per inserirli è necessario eseguire la procedura appena descritta utilizzando questo file.

Ogni volta che si modificano i dati di gioco generali nel database è necessario riavviare l'applicativo lato server affinché le modifiche abbiano effetto nel gioco.

4.3 Configurare il cluster e Spray

All'avvio del `cluster|g|` i seed-node devono essere avviati per primi. I seed-node sono i server² che identificano il cluster e ai quali i server normali fanno riferimento per unirsi al cluster. Per questo motivo i server normali devono essere avviati necessariamente dopo almeno il primo seed-node. Si consiglia, per motivi di affidabilità, di avere almeno due o tre server seed-node per cluster.

Ogni applicativo server deve sapere se è un seed-node e quali sono i seed-node. Queste informazioni sono configurabili editando il file “*Codice/sgad/severtier/src/main/resources/application.conf*” con percorso indicato dalla cartella radice del prodotto.

Il file di default sarà configurato per funzionare con un cluster formato da un singolo server. Esso si presenterà così:

```
1 akka {
2   loglevel = INFO
3   log-dead-letters = off
4   actor {
5     provider = "akka.cluster.ClusterActorRefProvider"
6   }
7   remote {
8     log-remote-lifecycle-events = off
9     netty.tcp {
10      hostname = "localhost"
11      port = 2787
12    }
13  }
14
15  cluster {
16    seed-nodes = [
17      "akka.tcp://sgadSystem@localhost:2787"
18    ]
19
20    auto-down = on
21  }
22 }
23
24
25 spray.can.server {
26   request-timeout = 24s
27   pipelining-limit = 8
28   resgistration-timeout = 20s
29 }
```

Alla linea 10 è configurabile l'indirizzo IP esterno della macchina utilizzata. Se si vuole utilizzare una unica macchina si può lasciare “localhost”.

Alla riga 11 è necessario impostare la porta di comunicazione verso la quale il server si esporrà agli altri server del cluster. È obbligatorio impostarla per i server seed-node. Se

²Conviene usare i più stabili e performanti

il server non è un seed-node si può lasciare all'applicazione il compito di scegliere una porta libera automaticamente impostando il valore a 0.

Alla riga 17 sono configurabili gli indirizzi dei seed-node. Per la configurazione a singolo server bisogna impostare l'indirizzo dello stesso server e, in generale, se un server è un seed-node inserire comunque l'indirizzo dello stesso nella lista. Per configurare più seed-node basta aggiungere altre righe e separarle con virgole. La singola riga deve mantenere lo stile impostato, inserire l'indirizzo dopo la '@' e, separata da ':', la porta, senza mai inserire spazi.

Alla riga 26 è possibile impostare il timeout massimo di una richiesta *HTTP*_{|g|}, è importante che sia configurato più alto di tutti i timeout configurati alla sezione 4.4.

Alla linea 27 è possibile impostare il numero massimo di connessioni accettate da un singolo client.

Alla linea 28 è configurabile il tempo massimo per il quale si aspetta che un richiesta HTTP possa essere gestita.

4.4 Configurare il server HTTP

Nel file di configurazione "*Codice/sgad/severtier/src/main/resources/serverhttp.conf*" con percorso indicato dalla cartella radice del prodotto è possibile configurare i server HTTP. Esso si presenta nella configurazione standard nel seguente modo:

```

1 # Configurazione del server HTTP
2
3 hostname                = localhost
4 port                    = 8080
5 minimizeCode            = false
6 httpRequestTimeouts    = 22
7 httpRequestWorkerTimeouts = 21
8 verifyUserActorAliveTimeouts = 3
9 userActorLiveTimeouts  = 180
10 internalRequestTimeouts = 3
11
12 # Configurazione di default
13
14 #hostname                = localhost
15 #port                    = 80
16 #minimizeCode            = false
17 #httpRequestTimeouts    = 22
18 #httpRequestWorkerTimeouts = 21
19 #verifyUserActorAliveTimeouts = 3
20 #userActorLiveTimeouts  = 300

```

Dalla linea 12 in poi è possibile vedere i parametri di default utilizzati dall'applicazione, nel caso siano inseriti valori non validi nel file. Anche un solo inserimento sbagliato provocherà l'utilizzo di tutti i valori di default.

Alla linea 3 è configurabile l'indirizzo in cui il server riceverà le richieste HTTP. Impostare l'indirizzo IP esterno della macchina server.

Alla linea 4 è configurabile la porta sulla quale il server ascolterà le richieste HTTP.

Alla linea 5 è configurabile la minimizzazione del codice lato client impostandone il valore booleano desiderato. Se impostato a `true` il server all'avvio caricherà il codice lato client e lo minimizzerà andando a ridurre i tempi di invio, rimuovendo commenti e spazi fornendo un primo oscuramento del codice. Questo ritarderà l'avvio del server da 30 secondi a 3 minuti circa in base alla potenza dello stesso.

Alle linee 6 e 7 è configurabile il timeout per il quale il server HTTP aspetterà una risposta. È importante che il valore alla linea 6 sia strettamente maggiore del valore della linea 7. È ammesso l'utilizzo di valori decimali, avendo cura di separare le cifre decimali con il punto.

Alla linea 8 è configurabile il tempo massimo per cui si aspetta una risposta durante la verifica di presenza dell'attore associato ad un utente. Questo tempo influenza quindi la durata minima di una richiesta di login. È ammesso l'utilizzo di un valore decimale, avendo cura di separare le cifre decimali con il punto.

Alla linea 9 è configurabile la durata della sessione di un utente. È ammesso l'utilizzo di un valore decimale, avendo cura di separare le cifre decimali con il punto.

Alla linea 10 è configurabile l'attesa massima di una richiesta interna. Essa è utilizzata per le richieste interne fra gli attori degli utenti per le loro interazioni (ad esempio: l'attacco). È ammesso l'utilizzo di un valore decimale, avendo cura di separare le cifre decimali con il punto.

4.5 Configurare il database

Nel file di configurazione "*Codice/sgad/severtier/src/main/resources/database.conf*" con percorso indicato dalla cartella radice del prodotto è possibile configurare quale istanza di MongoDB l'applicativo andrà ad utilizzare. Esso si presenta nella configurazione standard nel seguente modo:

```
1 # Configurazione del Database Manager
2
3 username                =
4 password                =
5 database_name           = sgadddb
6 db_socket               = localhost:27017
7 connection_timeoutMS    = 6000
8 socket_timeoutMS        = 6000
9
10 # Configurazione di Default, viene impostata se la
    configurazione specifica non è valida
11
12 #default_username       =
13 #default_password       =
14 #default_database_name  = sgadddb
15 #default_db_socket      = localhost:27017
16 #default_connection_timeoutMS = 6000
17 #default_socket_timeoutMS = 6000
```

Dalla linea 10 in poi è possibile vedere i parametri di default utilizzati dall'applicazione nel caso siano inseriti valori non validi nel file. Anche un solo inserimento sbagliato provocherà l'utilizzo di tutti i valori di default.

Alla linea 3 è obbligatorio inserire l'user dell'utente amministratore del database di gioco configurato in precedenza.

Alla linea 4 è obbligatorio inserire la password dell'utente amministratore del database di gioco.

Alla linea 5 è configurabile un nome diverso per il database del gioco. Si consiglia di non modificare.

Alla linea 6 è configurabile l'indirizzo e la porta (separati da ':') del database scelto. Se il database è stato installato in locale si può lasciare il valore inserito.

Alle linee 7 e 8 sono configurabili i valori di attesa massima per ciascuna richiesta verso il database. Si consiglia di settare i due valori allo stesso modo.

5 Avvio

Eseguita la configurazione si è pronti per avviare il prodotto. È possibile farlo tramite i seguenti comandi di terminale:

1. Navigare via terminale, tramite il comando `cd`, nella *directory*_{|g|} “*Codice/sgad/sever-tier*” a partire dalla cartella radice del progetto;
2. `sbt` - avvia SBT. Se la macchina su cui si vuole eseguire dispone di poca RAM ed SBT non si avvia si può provare con il seguente comando alternativo: `java -jar /usr/share/sbt/bin/sbt-launch.jar run`. Se si è precedentemente settata una porta http inferiore alla 1024 è necessario avviare SBT con i permessi di root, semplicemente antepoendo la parola chiave `sudo` al comando di avvio.
3. `run`

È possibile in alternativa avviare contemporaneamente SBT e l’avvio dell’applicativo³ con il seguente ed unico comando: `sbt run`.

È possibile fermare l’esecuzione del prodotto chiudendo SBT con la combinazione di tasti: `CTRL+C`.

Per il deploy dell’applicativo si consiglia di usare il software “Gnu Screen” in modo da evitare che il processo venga terminato alla chiusura della connessione ssh.

³Corrisponde ad eseguire i passi 2 e 3 contemporaneamente