

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione



POLO REGIONALE DI COMO
Corso di Laurea di Primo Livello in Ingegneria Informatica

GRU CONFIGURATION MANAGER

Sviluppo di una web application per la generazione di configurazioni per il tool Gru

Customer: Ing. Luca Florio

Thesys by:

Angelo Claudio Re
Student Id n. 758172

Academic Year 2015/2016

INDICE

1. Specifica dei requisiti.....	pag.3
2. Definizione dell'architettura del sistema.....	pag.18
3. Piano ed effettuazione dei Test.....	pag.25

1. SPECIFICA DEI REQUISITI

Gru è un tool parte di una PhD research al Politecnico di Milano: "Decentralized Self-Adaptation in Large-Scale Distributed Systems" ed è correntemente in fase di sviluppo da parte del committente della web application.

Essenzialmente il suo scopo è quello di gestire in modo automatico l'infrastruttura che caratterizza un sistema distribuito su larga scala

Per una completa documentazione, si rimanda al [repository di gru](#).

Tale infrastruttura sfrutta la recente tecnologia dei [docker container](#) per eseguire le varie macchine di cui è composto il sistema, per poi gestirne le risorse.

Le API del tool sono interamente scritte tramite il linguaggio di programmazione [GO](#).

Per funzionare Gru ha bisogno di alcuni parametri, che sono stati gestiti tramite delle configurazioni in formato Json.

Tali configurazioni si suddividono in 4 tipi:

- Configurazione dell'agente legato ad un cluster
- Configurazione dei servizi legati al cluster
- Configurazione della politica adottata dal container del cluster
- Configurazione delle analitiche usate dal container del cluster

Le varie configurazioni dovranno mantenere le seguenti strutture:

Configurazione per l'agente

La configurazione dell'agente deve essere corrispondente ad un file JSON così strutturato:

```
{
  "Docker": {
    "DaemonUrl": "unix:///var/run/docker.sock",
    "DaemonTimeout": 10(1)
  },
  "Autonomic": {
    "LoopTimeInterval": 30(2),
    "PlannerStrategy": "probabilistic"(3),
    "EnableLogReading": true(4)
  },
  "Communication": {
    "LoopTimeInterval": 28(5),
    "MaxFriends": 5(6)
  },
  "Storage": {
    "StorageService": "internal"(7)
  },
  "Metric": {
    "MetricService": "influxdb"(8),
    "Configuration": {
      "Url": "http://localhost:8086",
      "DbName": "gruDB",
      "Username": "luca",
      "Password": "influx_pwd"
    }
  }
}

"Discovery": {
  "AppRoot": "testApp",
  "TTL": 5(9)
}
```

NOTE AGENTE

- (1) DaemonTimeout: intero > 0
- (2) Autonomic LoopTimeInterval: intero > 0
- (3) Menù a tendina con scelte possibili (probdelta, prodcumulative e dummy)
- (4) Autonomic EnableLogReading: Flag
- (5) Communication LoopTimeInterval: intero > 0
- (6) Communication MaxFriends: intero > 0
- (7) Storage StorageService: Menù a tendina, scelte possibili (internal)

- (8) Metric MetricService: Menù a tendina, scelte possibili (influxdb)
- (9) Discovery TTL: intero > 0

Configurazione per i servizi

La configurazione dei servizi deve essere corrispondente ad un file JSON così strutturato:

```
{
  "Name": "service1a",
  "Type": "service1a",
  "Image": "elleflorio/service1a",
  "Remote": "/gru/pippo/services/service1a",(1)
  "DiscoveryPort": "50100",
  "Analytics": [(2)
    "resp_time_ratio"
  ],
  "Constraints": {(3)
    "MAX_RESP_TIME": 1500,
  },
  "Configuration": {
    "Env": {(4)
      "ETCD_ADDR": "",
      "HostIP": "",
      "INFLUX_USER": "",
      "INFLUX_PWD": "",
      "INFLUX_ADDR": ""
    },
    "Cmd": [(5)
      "start",
      "service1a",
      "-p",
      "50100",
      "-w",
      "low"
    ],
    "cpunumber": 1,(6)
    "Ports": {
      "50100": "50100-50104"(7)
    },
    "StopTimeout": 30(8)
  }
}
```

NOTE SERVIZI

- (1) Remote: Questo non viene scritto dall'utente ma direttamente dall'applicazione quando salva il tutto. La struttura è gru/<cluster>/services/<servizio>
- (2) Analytics: menù a tendina che va a pescare le analitiche già configurate dall'utente
- (3) Constraints: mappa chiave-valore di stringhe:float64
- (4) Configuration Env: qui l'utente ha la possibilità di specificare il nome della variabile di ambiente ed opzionalmente il valore. Il tutto quindi andrebbe fatto come "tabella" alla quale l'utente può aggiungere righe ognuna contenente una variabile di ambiente ed opzionalmente un valore (se è vuoto scriverò "" nella configurazione)
- (5) Configuration Cmd: stringa intervallata da spazi che poi verrà convertita in array nella configurazione. In questo caso l'utente avrebbe scritto: "start service1a -p 50100 -w low"
- (6) Configuration cpunumber: intero > 0
- (7) Configuration Ports: Il tutto deve essere rappresentato come una tabella con la prima colonna che rappresenta la porta del guest e la seconda che rappresenta l'intervallo di porte dell'host. La stringa che indica l'intervallo può essere composta anche da una sola porta, ma se si vuole indicare un intervallo, questo deve essere rappresentato come "porta_inizio-porta_fine"
- (8) Configuration StopTimeout: intero > 0

Configurazione politiche

La configurazione delle politiche deve essere corrispondente ad un file JSON così strutturato:

```
{
  "Scalein": {
    "Enable": true,(1)
    "Threshold": 0.3,(2)
    "Metrics": [
      "cpu_avg"
    ],
    "Analytics": [
      "resp_time_ratio"(3)
    ]
  },
  "Scaleout": {
    "Enable": true,(1)
    "Threshold": 0.8,(2)
    "Metrics": [
      "cpu_avg"
    ],
    "Analytics": [
      "resp_time_ratio"(3)
    ]
  },
  "Swap": {
    "Enable": true,(1)
    "Threshold": 0.6,(2)
    "Metrics": [
      "cpu_avg"
    ],
    "Analytics": [
      "resp_time_ratio"(3)
    ]
  }
}
```

NOTE POLITICA

(1) Scalein Enable, Scaleout Enable, Swap Enable: flag

(2) Scalein Threshold, Scaleout Threshold, Swap Threshold: valore float tra 0 e 1

(3) Scalein Analytics, Scaleout Analytics, Swap Analytics: menù a tendina che va a pescare le analitiche già configurate dall'utente

Configurazione analitiche

La configurazione delle analitiche deve essere corrispondente ad un file JSON strutturato nel seguente modo. (come per i servizi, ci sarà una diversa chiave/diverso file per ogni analitica)

```
{
  "Name": "resp_time_ratio",
  "Expr": "(execution_time - MIN_RESP_TIME) / (MAX_RESP_TIME - MIN_RESP_TIME)",
  "Metrics": [
    "execution_time"
  ],
  "Constraints": [
    "MAX_RESP_TIME",
    "MIN_RESP_TIME"
  ]
}
```

NOTE ANALITICHE

Name e Expr sono parametri obbligatori e in particolare nel secondo vanno inseriti i valori contenuti in Metrics o Configuration.

Il committente ha deciso di usare come store di tali configurazioni il server [Etd](#), distribuito e creato dalla [CoreOS](#).

Come per gru anche le [API del client nativo](#) per comunicare e interagire con Etd sono sviluppate in GO.

La struttura delle chiavi su etcd è la seguente:

```
/gru/
  |--- <cluster>/
    |--- uuid:string (id del cluster)
    |--- nodes/
      |--- <node>
        |--- configuration:string
        |--- constraints:string
        |--- resources:string
        |--- active:string
    |--- config (configurazione agente)
    |--- services/
      | --- <service>:string (configurazione servizio, salvata con il nome)
    | -- policy (configurazione politiche)
    | -- analytics/
      | -- <analytic>:string (configurazione analitica, salvata con il nome)
```

Quindi avremo ad esempio

```
/gru/
  |--- pippo/
    |--- uuid
    |--- nodes/
    |--- config (configurazione agente)
    |--- services/
      | --- servizio1
      | --- servizio2
      | --- servizio3
    | -- policy (configurazione politiche)
    | -- analytics/
      | -- analitica1
      | -- analitica2
```

Tutte le configurazioni sono salvate come stringhe.

Fino ad ora il salvataggio di tali configurazioni avviene direttamente dalla command line usando i comandi nativi del client di Etcd.

Da qui nasce l'esigenza di avere a disposizione una web application per consentire agli utenti (che implementeranno gru per gestire il proprio sistema) un facile e semplice accesso ad etcd per generare le configurazioni.

Una volta avviato il servizio, l'utente si collegherà tramite il browser ed avrà a disposizione un'interfaccia grafica per creare la configurazione.

In fase di avvio il configuration manager deve collegarsi ad una istanza di etcd e verificare la presenza di cluster gru. Se non sono presenti cluster l'utente non può generare la configurazione.

L'utente può selezionare dalla lista uno dei cluster disponibili e può quindi creare la configurazione per:

- agente
- servizi
- politica
- analitiche

Al salvataggio della prima configurazione di un cluster dovrà essere generato il suo ID univoco.

REQUISITI UTENTE

Nome Requisito	Gestione configurazione agente
Id Requisito	1
Tipo	Funzionale
Priorità	Alta
Criticità	Alta
Rischio	Basso
Data	20/04/2016
Descrizione	Salvare, aggiornare ed eliminare in etcd la configurazione dell'agente di un cluster
Fonte	Verbale, email
Implementazione	01/06/2016
Requisito Padre	/
Requisiti Figli	/

Nome Requisito	Gestione configurazione servizi
Id Requisito	2
Tipo	Funzionale
Priorità	Alta
Criticità	Alta
Rischio	Basso
Data	20/04/2016
Descrizione	Salvare, aggiornare ed eliminare in etcd la configurazione dei servizi di un cluster
Fonte	/
Implementazione	01/06/2016
Requisito Padre	/
Requisiti Figli	/

Nome Requisito	Gestione configurazione politica
Id Requisito	3
Tipo	Funzionale
Priorità	Alta
Criticità	Alta
Rischio	Basso
Data	20/06/2016
Descrizione	Salvare, aggiornare ed eliminare in etcd la configurazione di una politica di un cluster
Fonte	Verbale, email
Implementazione	01/07/2016
Requisito Padre	/
Requisiti Figli	/

Nome Requisito	Gestione configurazione analitiche
Id Requisito	4
Tipo	Funzionale
Priorità	Alta
Criticità	Alta
Rischio	Basso
Data	20/06/2016
Descrizione	Salvare, aggiornare ed eliminare in etcd la configurazione delle analitiche di un cluster
Fonte	Verbale, email
Implementazione	20/08/2016
Requisito Padre	/
Requisiti Figli	/

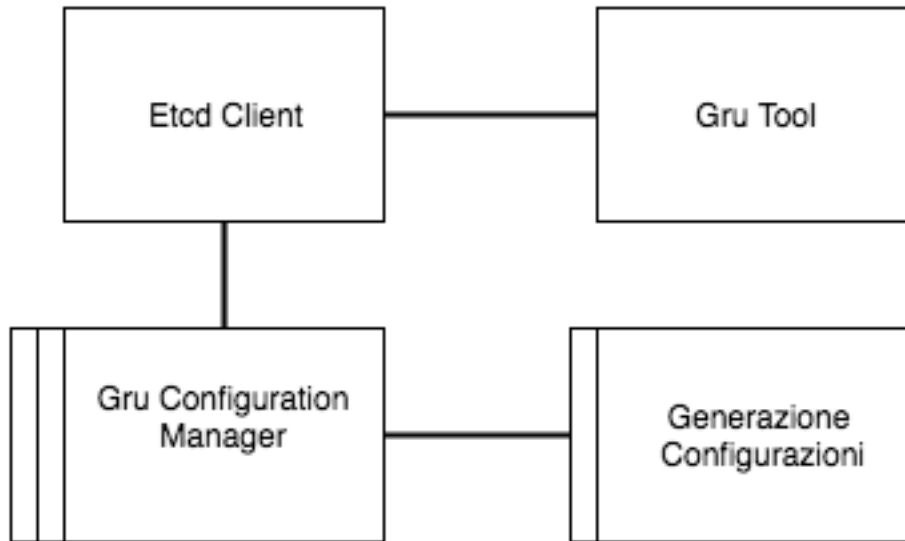
Nome Requisito	Autenticazione ad etcd
Id Requisito	5
Tipo	Funzionale
Priorità	Alta
Criticità	Alta
Rischio	Basso
Data	20/04/2016
Descrizione	L'utente potrà accedere ad etcd autenticandosi in fase preliminare
Fonte	Verbale, email
Implementazione	20/08/2016
Requisito Padre	/
Requisiti Figli	/

Nome Requisito	Creazione Cluster ID
Id Requisito	5
Tipo	Funzionale
Priorità	Alta
Criticità	Media
Rischio	Basso
Data	20/04/2016
Descrizione	In fase di salvataggio di un cluster, andrebbe creato anche il suo ID univoco
Fonte	Verbale, email
Implementazione	20/08/2016
Requisito Padre	/
Requisiti Figli	/

Nome Requisito	Creazione Utente Admin Etc
Id Requisito	5
Tipo	Funzionale
Priorità	Alta
Criticità	Alta
Rischio	Basso
Data	20/04/2016
Descrizione	Creazione utente Admin in fase di prima autenticazione con etc
Fonte	Verbale, email
Implementazione	20/08/2016
Requisito Padre	/
Requisiti Figli	/

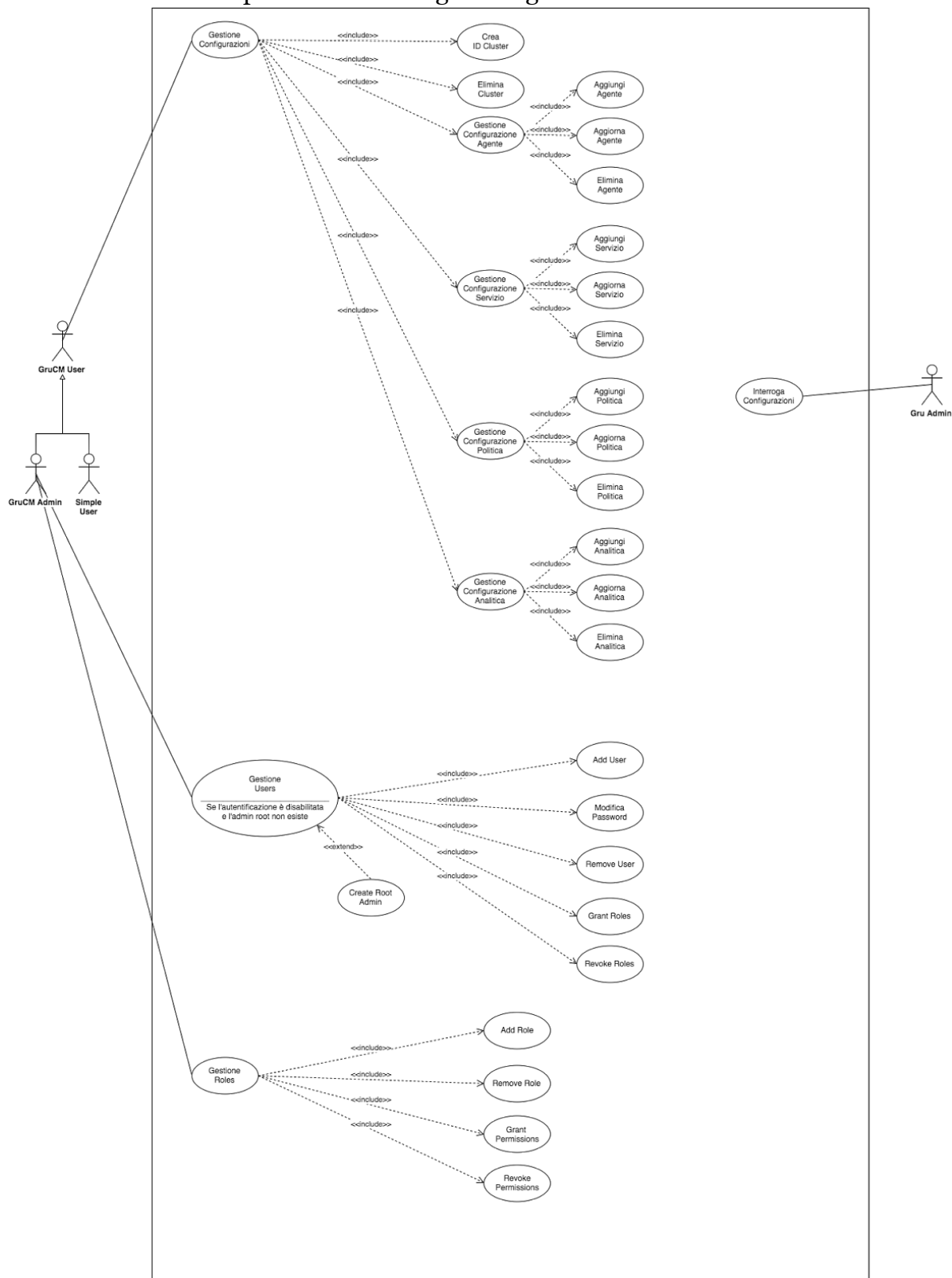
CONTEXT DIAGRAM

Possiamo rappresentare il dominio applicativo del sistema con il seguente diagramma di contesto.



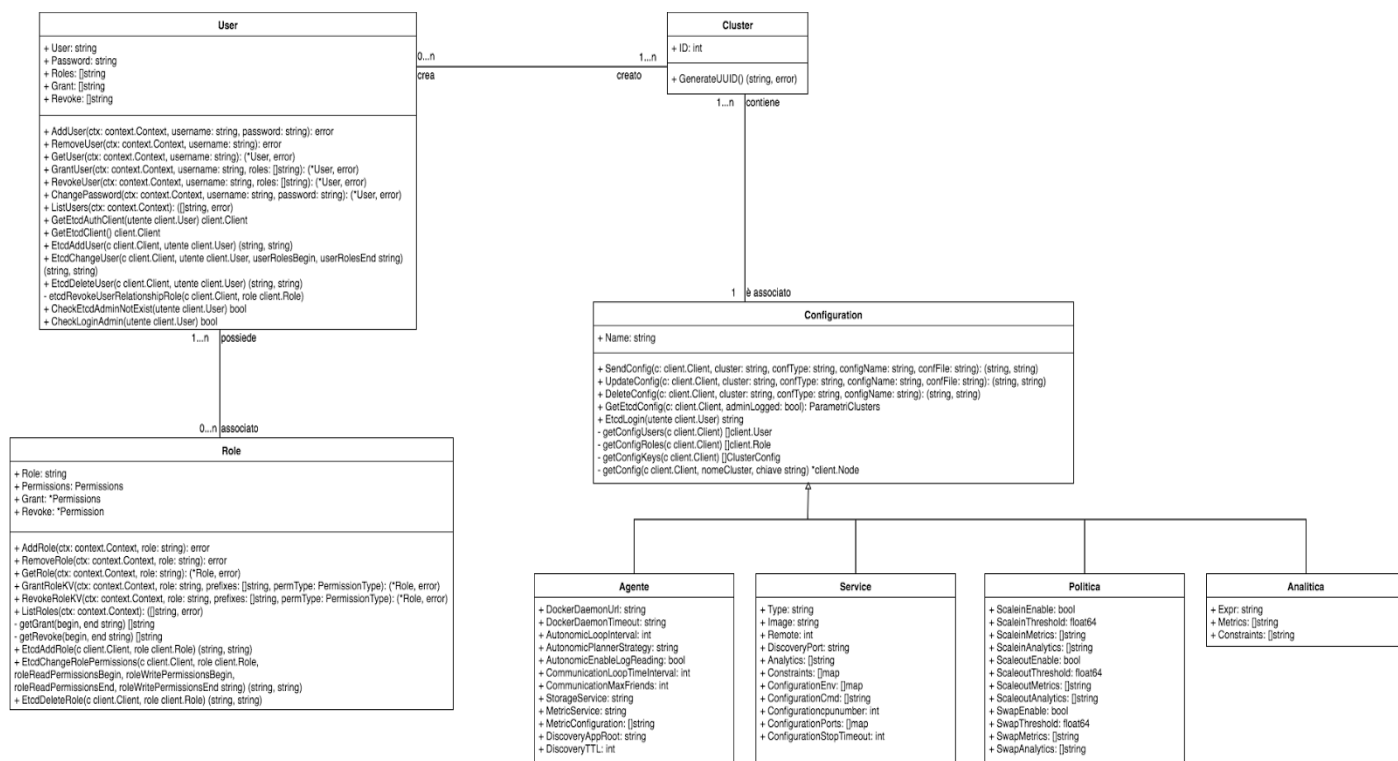
USE CASE DIAGRAM

Gli Utenti avranno la possibilità di svolgere i seguenti tasks:



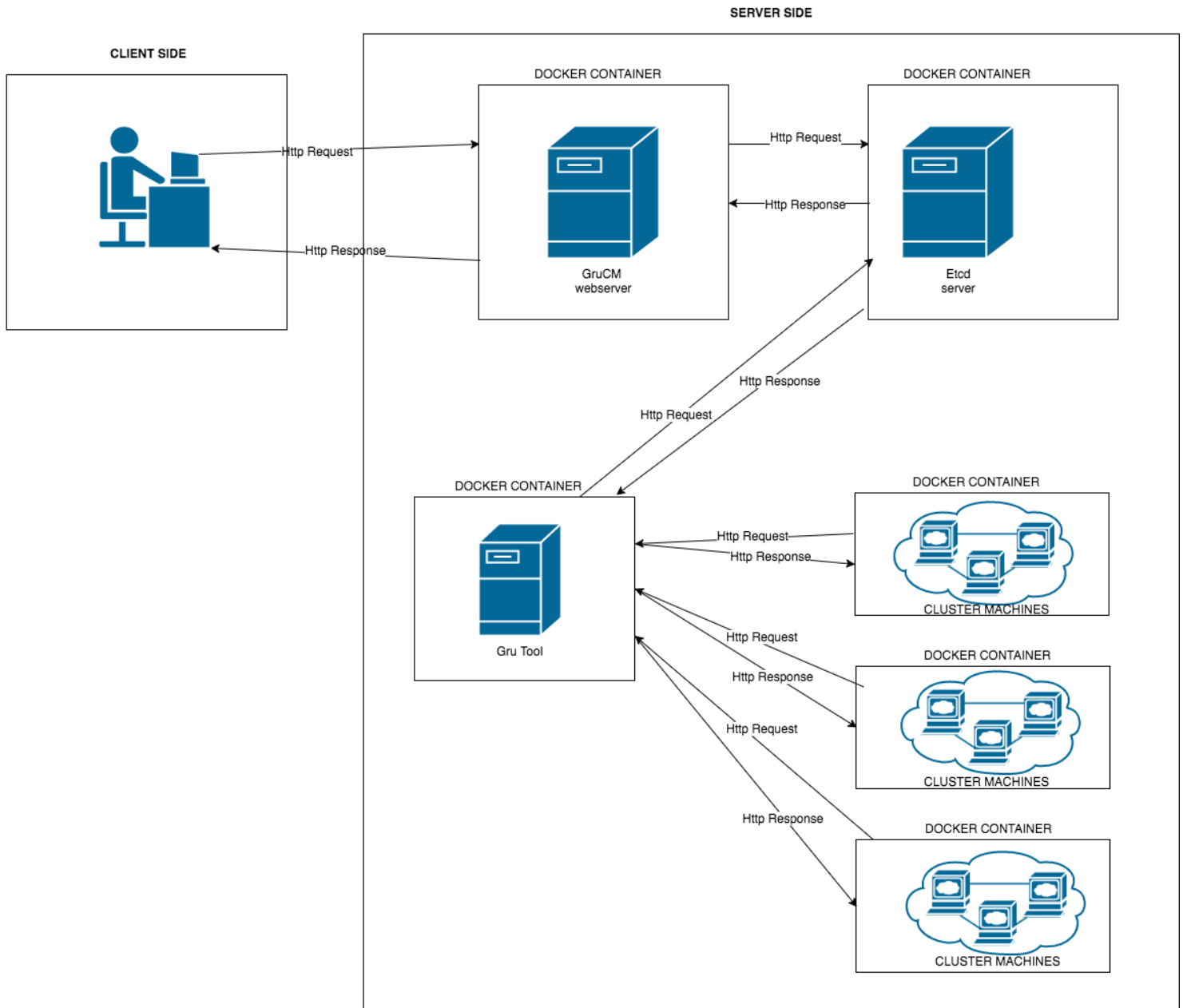
CLASS DIAGRAM

Possiamo riassumere le principali API utilizzate dal web server per generare le pagine HTML del GruCM



2. ARCHITETTURA DEL SISTEMA

Il sistema può essere schematizzato con la struttura sottostante



Il sistema creato per il configuration manager è costituito da un'architettura client-server suddivisa in 3 parti

- FRONTEND: HTML5 + CSS + JS
- BACKEND: web server in GO
- DB: Etcd database non relazionale distribuito chiave – valore (da quest ultimo il tool gru recupererà le configurazioni dei cluster necessarie)

Per l'installazione del web server si rimanda alla pagina github dello sviluppatore: [GruCM Repo](#).

Per completezza tale procedura verrà successivamente riepilogata.

Per la parte di backend si è scelto di utilizzare Go come linguaggio di gestione delle pagine html richieste al web server per comodità, dato che sia Etcd che Gru sono nativamente sviluppati con tale linguaggio.

Per creare il web server si è scelto di usare la [libreria Martini](#), molto affidabile e già consolidata per la creazione di web application in linguaggio GO.

Tramite librerie collegate a Martini, si sono anche implementate le classiche funzioni di gestione della [sessione utente](#) (che di default ha una durata di 30 minuti) e la renderizzazione tramite il browser delle pagine HTML che consentono di interagire con Etcd ([martini-contrib/render](#)).

Per quanto riguarda l'ultima libreria nominata, è stata creata un'estensione che permette di creare degli handles che caricano correttamente i fogli CSS e gli script JS richiamati esternamente, dato che non erano gestiti a standard.

Per l'autenticazione al sistema sono state utilizzate delle API native di Etcd per la gestione degli utenti.

La parte di frontend è stata sviluppata tramite la classica terna formata da HTML, CSS e JS.

Il web server riceve dal browser solo delle stringhe e le salva su etcd, la parte di creazione dei Json e il lavoro spaccettamento nelle form HTML in fase di lettura dati è totalmente gestito lato client alleggerendo il lavoro del web-server

Tale scelta è stata attuata a prescindere da eventuali usi futuri e dal workload possibile gestito dal web-server, quindi si è preso in considerazione il caso peggiore in assenza di specifiche direttive.

Per il database come specificato nel capitolo precedente verrà usato il client nativo di etcd eseguito in un container docker.

Questa ultima tecnologia è stata sfruttata anche per dare la possibilità agli utenti di eseguire la web application da un container docker, così da risparmiarsi tediose installazioni di script engine per interpretare le istruzioni del codice in GO e di eseguire meno istruzioni per far funzionare il tutto a prescindere dal sistema di utilizzo.

INSTALLAZIONE DEL WEB SERVER PER ESEGUIRE LA WEB APPLICATION

Per l'installazione ci sono due procedure:

- 1) Tramite Docker Container
- 2) Tramite i comandi di Go

1) PROCEDURA CON DOCKER CONTAINER

Per poter interagire con la web application bisogna seguire le seguenti fasi.

1.1) Scaricare e installare [docker](#).

1.2) Esportare la variabile d'ambiente da cmd che identifica l'IP dell'Host su cui sarà eseguito Etcd e a titolo d'esempio inseriamo quello del localhost.

"Export HostIP = 127.0.0.1"

es.

```
Last login: Mon Sep 12 17:02:49 on ttys000
MacBook-Pro-di-angelo:~ angelo$ export HostIP=127.0.0.1
MacBook-Pro-di-angelo:~ angelo$ █
```

1.3) Esportiamo la variabile d'ambiente da cmd che identifica la porta a cui il web server sarà in ascolto di richieste http.

"Export HostPort= 8080"

es.

```
MacBook-Pro-di-angelo:~ angelo$ export HostPort=8080
MacBook-Pro-di-angelo:~ angelo$ █
```

In questo caso, l'importante è non utilizzare le seguenti 3 porte già occupate da etcd nel caso sia in esecuzione insieme al web server sulla stessa macchina (2379, 2380, 4001).

1.4) A questo punto attiviamo docker attraverso l'app precedentemente installata

1.5) Eseguiamo il comando per lanciare in esecuzione etcd dentro un container

```
docker run -d -v /usr/share/ca-certificates:/etc/ssl/certs -p 4001:4001 -p 2380:2380 -p
2379:2379 \
--name etcd quay.io/coreos/etcd \
/usr/local/bin/etcd \
-name etcd \
-advertise-client-urls http://${HostIP}:2379,http://${HostIP}:4001 \
-listen-client-urls http://0.0.0.0:2379,http://0.0.0.0:4001 \
-initial-advertise-peer-urls http://${HostIP}:2380 \
-listen-peer-urls http://0.0.0.0:2380 \
-initial-cluster-token etcd-cluster-1 \
-initial-cluster etcd=http://${HostIP}:2380 \
-initial-cluster-state new
```

Il valore del parametro --name e -name è configurabile a proprio piacimento.

Nel caso state eseguendo questo comando per la prima volta o avete cancellato le docker images relative ad etcd, prima della sua esecuzione verranno scaricate .

1.6) Eseguiamo il comando per lanciare in esecuzione il gruCM dentro un container

```
docker run --name=grucom -p ${HostPort}:${HostPort} -e "HostIP=${HostIP}" -e
"HostPort=${HostPort}" --rm gexkill/grucom:webapp
```

Nel caso state eseguendo questo comando per la prima volta o avete cancellato le docker images relative al GruCM, prima della sua esecuzione verranno scaricate.

1.7) A questo punto possiamo collegarci all'indirizzo della macchina sulla quale il web server è in esecuzione

A titolo d'esempio se stiamo eseguendo dei test in locale useremo

<http://localhost:8080/gruCM>

- 1.8) (Opzionale) Nel caso vogliamo gestire la parte di creazione degli utenti e dei ruoli non tramite web server e l'interfaccia grafica della pagine richiamate, c'è la possibilità di utilizzare un comando da eseguire dentro un docker container seguito delle opzioni che permettono di gestirli.

```
docker run --rm gexkill/grucm:cmd  
--endpoint="http://${HostIP}:2379,http://${HostIP}:4001" + comandi di gestione .
```

2) PROCEDURA CON LO SCRIPT ENGINE DI GO

- 2.1) Installare lo [script engine di Go](#) sulle macchine che eseguiranno etcd e il web server (se diverse)

- 2.2) Esportare le seguenti variabili d'ambiente

```
- Gopath Location  
export GOPATH=$HOME/{{GoDirectory}}
```

dove per {{GoDirectory}}, bisogna specificare la cartella dove volete installare i sorgenti e gli eseguibili dei file che scaricherete.

```
- Workspace's bin subdirectory  
export PATH=$PATH:$GOPATH/bin
```

```
- Etcd HostIP  
export HostIP={{IpMacchina}}
```

dove l'IpMacchina è l'indirizzo dove è in esecuzione etcd facilmente individuabile eseguendo nella shell dei comandi `ifconfig`

```
- GruCM Web Server Port  
export HostPort=8080
```

- 2.3) a questo punto sulla macchina dove volete eseguire etcd scaricheremo i suoi sorgenti attraverso il comando

```
go get github.com/coreos/etcd
```

2.4) e installiamo etcd

```
go install github.com/coreos/etcd
```

2.5) per lanciare etcd ci sono due alternative:

da cmd

```
cd {{GoDirectory}}/src/github.com/coreos/etcd  
go run main.go
```

oppure

```
cd {{GoDirectory}}/bin  
./etcd for unix like system  
etcd for MS-DOS like system
```

2.6) Passiamo ora al web server

Scarichiamo i suoi sorgenti attraverso il comando

```
go get github.com/gexill/gruConfigurationManager
```

2.7) e lo installiamo

```
go install github.com/gexill/gruConfigurationManager
```

2.8) per lanciare GruCM ci sono due alternative:

da cmd

```
cd {{GoDirectory}}/src/github.com/gexill/gruConfigurationManager  
go run main.go
```

oppure

```
cd {{GoDirectory}}/bin  
./gruConfigurationManager for unix like system  
gruConfigurationManager for MS-DOS like system
```

2.9) A questo punto possiamo collegarci all'indirizzo della macchina sulla quale il web server è in esecuzione

A titolo d'esempio se stiamo eseguendo dei test in locale useremo

<http://localhost:8080/gruCM>

2.10) (Opzionale) Nel caso vogliamo gestire la parte di creazione degli utenti e dei ruoli non tramite web server e l'interfaccia grafica della pagine richiamate, c'è la possibilità di utilizzare un comando "etcdctl" seguito delle opzioni che permettono gestirli.

Prima lo installiamo

```
go get github.com/coreos/etcd/etcdctl  
go install github.com/coreos/etcd/etcdctl
```

ora possiamo eseguirlo con la seguente sintassi
etcdctl + [comandi di gestione](#) .

3. PIANO ED EFFETTUAZIONE DEI TEST

Sul sistema sono stati verificati i seguenti casi di test

Test name	Creazione admin, se non esiste, dalla pagina index.html al primo accesso ad etcd
Entry condition	Inserimento password e pressione tasto Crea Admin
Expected result	Correct: L'admin deve essere creato in etcd se il controllo iniziale sulla sua esistenza ha esito negativo, e deve essere restituito un messaggio di avvenuta creazione dopo essere stati reindirizzati alla pagina iniziale di inserimento di autenticazione utente. Viene abilitata l'autenticazione al sistema Wrong: compare una pagina diversa da quella di creazione dell'admin nonostante l'admin non esiste. La creazione dell'admin fallisce. Non viene abilitata l'autenticazione al sistema nonostante l'admin sia stato correttamente creato
Result	Verified
Observations	L'unico vincolo in questa situazione è il nome dell'utente admin che è stato impostato dagli sviluppatori della CoreOS fisso a "root" e non è modificabile per il corretto funzionamento di etcd.

Test name	Creazione admin, se non esiste, da linea di comando attraverso etcdctl
Entry condition	Inserimento password tramite da i comandi del cmd e abilitazione dell'autenticazione al sistema.
Expected result	Correct: L'admin deve essere creato in etcd Wrong: La creazione dell'admin fallisce. Non viene abilitata l'autenticazione al sistema nonostante l'admin sia stato correttamente creato
Result	Verified
Observations	Stessa considerazione fatta per il test precedente.

Test name	Creazione password utente admin da login.html
Entry condition	Inserimento nome utente con i privilegi di admin anche diverso da utente root (che è l'admin di riferimento), la vecchia password e ripetiamo la nuova password due volte
Expected result	Correct: La password viene cambiata, si viene reindirizzati alla pagina login.html iniziale per autenticarsi in etcd e appare un messaggio di operazione eseguita Wrong: La password non viene cambiata. Viene ritornato un errore di cluster non disponibile o configurato erroneamente.
Result	Verified
Observations	Questo flusso funziona solo per gli utenti con i privilegi del ruolo "root" e non per i semplici utenti. Tale scelta non è stata presa ancora in considerazione da chi rilascia etcd, permettendo solo agli admin di poter cambiare le password.

Test name	Autenticazione di un utente senza nessun ruolo associato
Entry condition	Inserimento nome utente e password
Expected result	Correct: L'utente riesce ad autenticarsi ma non può effettuare nessuna operazione Wrong: L'utente non riesce ad autenticarsi, e viene restituito un errore di cluster etcd non disponibile o configurato male
Result	Not Verified
Observations	Etcd a standard blocca l'accesso per gli utenti che non hanno associato almeno un ruolo con i permessi in lettura. Se si prova successivamente ad entrare in etcd con un utente Con ruolo "root" non vengono più caricati gli utenti presenti nel sistema impedendone possibile modifiche. Questo a causa dell'utente privo di ruoli che blocca la gestione utenti. L'unica soluzione è cancellarlo tramite i comandi da linea di comando e riaccedere al sistema. Comunque si tratta di un bug non gestito in questo momento da chi ha creato etcd.

Test name	Autenticazione di un utente con associato un ruolo diverso da root con permessi in lettura
Entry condition	Inserimento nome utente e password
Expected result	Correct: L'utente riesce ad autenticarsi ma non può modificare i dati presenti sull'istanza di etcd. Per verificare quest'ultima condizione verrà restituito un messaggio di errore all'utente. Wrong: L'utente non riesce ad autenticarsi L'utente riesce a scrivere/cancellare dati in etcd
Result	Verified
Observations	

Test name	Autenticazione di un utente con associato un ruolo diverso da root con permessi write
Entry condition	Inserimento nome utente e password
Expected result	Correct: L'utente riesce ad autenticarsi e viene restituito un messaggio di errore Wrong: L'utente riesce ad autenticarsi e riesce a visualizzare i dati delle configurazioni dei cluster presenti su etcd.
Result	Verified
Observations	

Test name	Autenticazione di un utente con associato un ruolo diverso da root con permessi readwrite
Entry condition	Inserimento nome utente e password
Expected result	Correct: L'utente riesce ad autenticarsi e può visualizzare e modificare i dati presenti sull'istanza di etcd. Wrong: L'utente non riesce ad autenticarsi L'utente non riesce a scrivere/cancellare dati in etcd
Result	Verified
Observations	

Test name	Gestione ruoli e utenti del sistema da parte dell'utente con ruolo di root
Entry condition	Visualizzazione dei due bottoni a menu
Expected result	Correct: L'utente riesce ad autenticarsi e può visualizzare e modificare i dati relativi a ruoli ed utenti presenti sull'istanza di etcd. Wrong: L'utente non riesce ad autenticarsi L'utente non riesce a scrivere/cancellare dati in etcd
Result	Verified
Observations	

Test name	Gestione ruoli e utenti del sistema da parte dell'utente senza ruolo di root
Entry condition	L'utente esegue il login
Expected result	Correct: L'utente non visualizza i due bottoni di gestione Wrong: L'utente vede i due bottoni e può modificare utenti e ruoli
Result	Verified
Observations	

Test name	Gestione ruoli e utenti del sistema da parte dell'utente senza ruolo di root
Entry condition	L'utente esegue il login
Expected result	Correct: L'utente non visualizza i due bottoni di gestione Wrong: L'utente vede i due bottoni e può modificare utenti e ruoli
Result	Verified
Observations	

Test name	Inserimento configurazione di un utente generico con ruolo readwrite
Entry condition	L'utente esegue il login e inserisce un nuovo cluster e salva la configurazione del suo agente
Expected result	<p>Correct: l'utente crea il nuovo cluster inserendo un nuovo nome (univoco) per il cluster , inserisce i dati del suo agente che dopo essere stati validati vengono salvati su etcd. Viene restituito un messaggio di successo e si viene reindirizzati alla pagina relativa all'ultima operazione eseguita, quindi non dovremo rileselzionare il cluster e il tipo configurazione agente per vederne i dati.</p> <p>Wrong: Viene restituito un messaggio d'errore, e la configurazione e il cluster non vengono salvati.</p>
Result	Verified
Observations	Stesso tipo di test è stessa eseguito con le altre 3 configurazioni (politica, servizio e analitica) oppure partendo da un cluster esistente con i medesimi risultati

Test name	Creazione ClusterID
Entry condition	L'utente salva una configurazione in etcd di un nuovo cluster
Expected result	<p>Correct: dopo il salvataggio accanto al nome del cluster appare il cluster ID ad esso associato</p> <p>Wrong: L'ID non viene visualizzato o non viene creato</p>
Result	Verified
Observations	

Test name	Esportazione di una configurazione in locale
Entry condition	L'utente seleziona il bottone "Esporta" dopo aver selezionato la configurazione già esistente su etcd di uno tra servizi, policy, analytics o agent config
Expected result	Correct: viene salvato un file json nella cartella download puntata dal browser Wrong: Il file non viene scaricato
Result	Verified
Observations	L'unica variante qua è quella del browser safari che non salva direttamente il file ma lo apre in un'altra pagina web per essere salvato

Test name	Visualizzazione identica dell'interfaccia grafica in vari browser
Entry condition	L'utente prova ad eseguire dei task in diversi browser
Expected result	Correct: in tutti i browser provati l'interfaccia è sempre la stessa Wrong: Ci sono delle incongruenze tra i vari browser sulla visualizzazione di alcuni elementi della pagina
Result	Verified
Observations	I Test sono stati condotti sui 4 browser di maggiore utilizzo commerciale di seguito elencati: (Chrome, Firefox, Safari, IE)

Test name	Visualizzazione della sola directory dei servizi di un determinato cluster in base al ruolo
Entry condition	L'utente prova a visualizzare i servizi associati a un cluster
Expected result	Correct: l'utente riesce ad autenticarsi e riesce a visualizzare solo la configurazione dei servizi associati ad un cluster e non potendo ne visualizzare ne modificare il resto. Wrong: L'utente non riesce ad autenticarsi
Result	Not Verified
Observations	Anche questa è una mancanza di etcd dato che riesce a gestire in maniera corretta le sicurezze dei ruoli solo inserimento le cartelle radici (nel nostro caso abbiamo solo /gru/ come radice, quindi è la sola.), invece inserimento la sotto cartelle blocca l'accesso a tutta la radice.

Test name	Blocco nell'inserimento di cluster con nomi duplicati
Entry condition	L'utente prova il nome di un nuovo cluster identico ad un altro già esistente
Expected result	Correct: viene restituito un messaggio d'errore Wrong: il nome viene inserito
Result	Verified
Observations	

Test name	Blocco nell'inserimento di un servizio per lo stesso cluster con un nome di un altro servizio preesistente
Entry condition	L'utente prova a caricare la configurazione di un servizio il cui nome è già presente per lo stesso cluster
Expected result	Correct: viene restituito un messaggio d'errore e non viene caricato nulla su etcd Wrong: l'inserimento viene effettuato con successo
Result	Verified
Observations	

Test name	Blocco nell'inserimento di un'analitica per lo stesso cluster con un nome di un'altra analitica preesistente
Entry condition	L'utente prova a caricare la configurazione di un' analitica il cui nome è già presente per lo stesso cluster
Expected result	Correct: viene restituito un messaggio d'errore e non viene caricato nulla su etcd Wrong: l'inserimento viene effettuato con successo
Result	Verified
Observations	

Test name	Blocco nell'inserimento di un servizio per lo stesso cluster con la discoveryPort già occupata da un altro servizio
Entry condition	L'utente prova a caricare la configurazione di un servizio la cui discovery port è già usata da un altro servizio
Expected result	Correct: viene restituito un messaggio d'errore e non viene caricato nulla su etcd Wrong: l'inserimento viene effettuato con successo
Result	Verified
Observations	

Test name	Blocco nell'inserimento di un servizio per lo stesso cluster con le porte host già occupate da un altro servizio
Entry condition	L'utente prova a caricare la configurazione di un servizio le cui portehost sono già usate da un altro servizio
Expected result	Correct: viene restituito un messaggio d'errore e non viene caricato nulla su etcd Wrong: l'inserimento viene effettuato con successo
Result	Verified
Observations	

Test name	Blocco nell'inserimento di un servizio per lo stesso cluster con la portaHostFrom minore della portaHostTo
Entry condition	L'utente prova a caricare la configurazione di un servizio la cui portaHostFrom è minore della portaHostTo
Expected result	Correct: viene restituito un messaggio d'errore e non viene caricato nulla su etcd Wrong: l'inserimento viene effettuato con successo
Result	Verified
Observations	