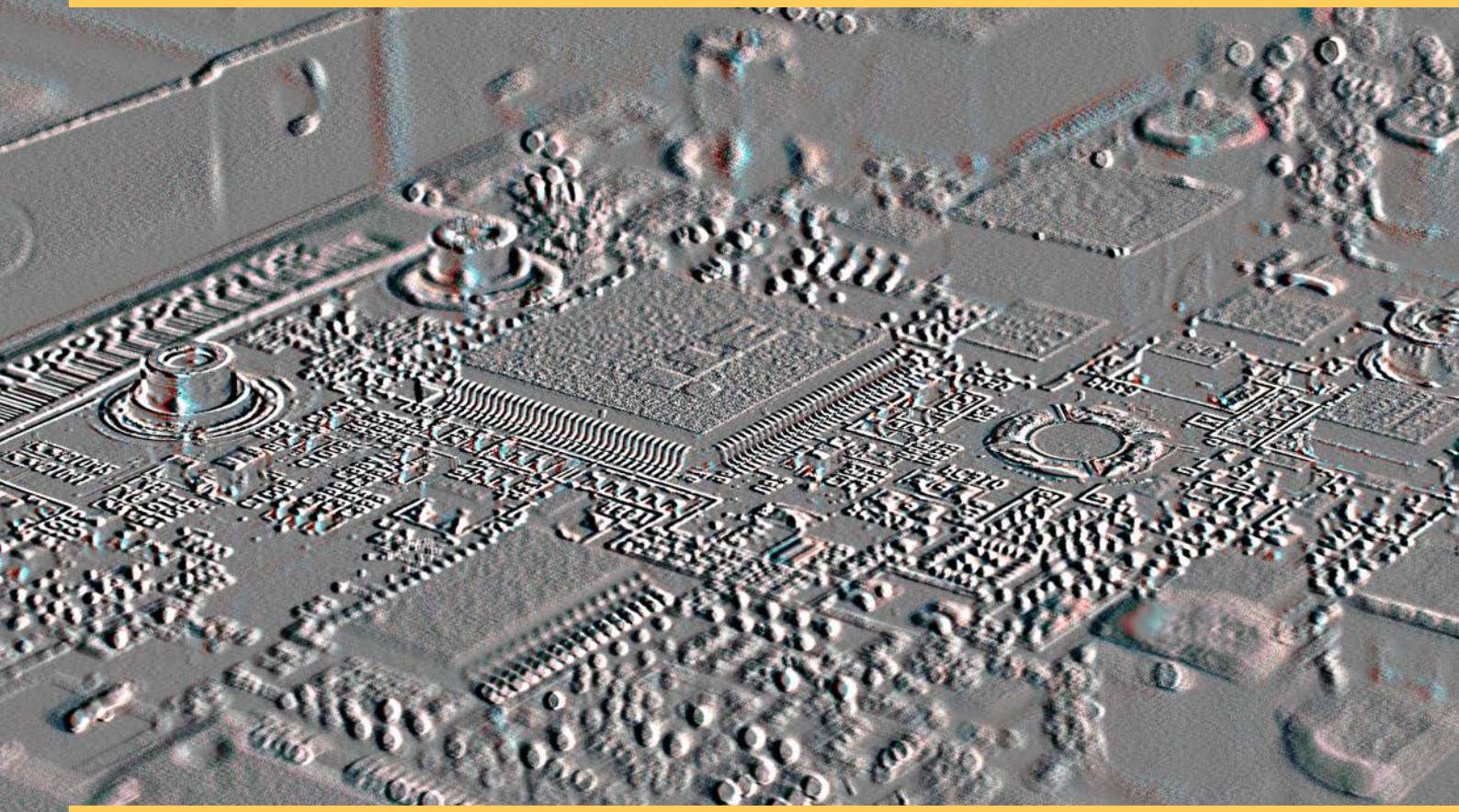
ISSN 2039-7941
anno 2021 numero 427



RAPPORTI TECNICI INGV

HSIT over cloud. Migrazione dell'infrastruttura HSIT su cloud Microsoft Azure

ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

427

REGISTRAZIONE AL TRIBUNALE DI ROMA N.174 | 2014, 23 LUGLIO

ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

ISSN 2039-7941
anno **2021** numero **427**

RAPPORTI TECNICI INGV

HSIT over cloud. Migrazione dell'infrastruttura HSIT su cloud Microsoft Azure

HSIT over cloud. Migration of the HSIT infrastructure to the Microsoft Azure cloud

Diego Sorrentino[1], Valerio De Rubeis[2], Patrizia Tosi[2], Paola Sbarra[2]

[1] INGV | Istituto Nazionale di Geofisica e Vulcanologia, Amministrazione Centrale
[2] INGV | Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Sismologia e Tettonofisica

Accettato 19 ottobre 2020 | *Accepted 19 October 2020*
Come citare | *How to cite* Sorrentino D., De Rubeis V., Tosi P., Sbarra P., (2021). HSIT over cloud. Migrazione dell'infrastruttura HSIT su cloud Microsoft Azure. Rapp. Tec. INGV, 427: 1-46, https://doi.org/10.13127/rpt/427

In copertina Elaborazione di B. Angioni da una fotografia di Alexandre Debiève, Unsplash | *Cover Processed by B. Angioni from a picture by Alexandre Debiève, Unsplash*

427

INDICE

Riassunto

L'attuale infrastruttura informatica di *Hai sentito il terremoto?*, di seguito *HSIT*[1], seppur perfezionata e ottimizzata negli anni per riuscire a sopportare un carico sempre maggiore di utenti, risente dell'obsolescenza dei dispositivi in uso e della difficoltà di aggiornamento dell'hardware, sia in termini burocratici, in quanto la procedura di acquisto richiede tempo, che tecnici, in quanto non è presente personale tecnico dedicato che, oltre alla prima installazione e configurazione, può costantemente monitorare il corretto funzionamento dell'intero sistema. Inoltre, in caso sia necessario incrementare risorse *on-the-fly*, non sono attualmente presenti dispositivi informatici in grado di *adattarsi dinamicamente* alle necessità del momento.
Si è deciso, quindi, di migrare l'intera infrastruttura informatica su piattaforma cloud, in quanto il servizio è costantemente aggiornato e mantenuto, permette di *scalare*[2], manualmente o automaticamente, il numero di dispositivi a seconda delle necessità e, non ultimo, in caso di problemi lato software (un aggiornamento del sistema fallito, un bug nella web application, ecc..) è sempre possibile tornare a uno stato funzionante, semplicemente ripristinando uno *snapshot*[3] precedente.
Il presente documento descrive l'architettura informatica realizzata su piattaforma *cloud* Microsoft Azure e la procedura per poter replicare l'intero sistema.

Abstract

Current Hai sentito il terremoto?, below HSIT[1], informatic infrastructure is stable and optimized over the years to be able to withstand an increasing load of users but is based on obsolete devices and is very hard to update, due to bureaucratic reasons, because the buy procedure takes a long time, and due to technical reason, because there is a lack of dedicated technical specialist, in addition to the initial installation and configuration, that can constantly monitor the entire system. Furthermore, if more resources are needed on-the-fly, there are no IT devices capable of dynamically adapting to the needs of the moment.
So we decide to migrate the entire informatic infrastructure over cloud platform, because the service is costanlty updated and manteined, permit to scale[2], manually or automatically, the device numbers based on real-time need, and the last but not the least during software issue (opeating system update failed, bug in web application, etc.) is always possible rollback to a previous working status, simply restoring a previuos snapshot[3].
This document describes the IT architecture built on the Microsoft Azure cloud platform and the procedure for replicating the entire system.

Keywords HSIT; Infrastruttura; Cloud | Dyfi; Infrastructure; Cloud

[1] Sito web ufficiale: https://www.hsit.it/

[2] Aumentare o diminuire

[3] Letteralmente una "istantanea", è generalmente la cattura di stato di un oggetto in un determinato momento nel tempo. Rappresenta uno stato del sistema in uno specifico momento, una fase di lavoro che si vuole "fermare" nel caso che le variazioni che si stanno per compiere non ci soddisfino. Le snapshot consentono quindi per esempio di vedere versioni alternative di una stessa immagine, per poter scegliere la migliore. Cit. https://it.wikipedia.org/wiki/Snapshot

Introduzione

HSIT è un portale web nato per monitorare in tempo reale gli effetti dei terremoti italiani e per informare la popolazione sull'attività sismica[4]. La sua realizzazione è resa possibile grazie al contributo di ogni persona che, compilando il questionario macrosismico descrive la propria esperienza.
Il portale riceve diversi questionari macrosismici immediatamente dopo l'occorrenza di un terremoto, anche prima della pubblicazione della localizzazione ufficiale, è quindi necessario che l'infrastruttura sia il più possibile stabile e flessibile per poter affrontare un carico di lavoro leggero durante "i momenti di pace" ma che sia capace di sostenere i picchi di accessi durante un'emergenza sismica.
Purtroppo, durante gli ultimi anni di attività, sono emerse diverse problematiche che non hanno permesso il corretto funzionamento del sistema, soprattutto nei momenti di maggior necessità, a volte arrivando fino al blocco totale del servizio. I disservizi verificatisi si possono raggruppare in tre macro aree: Software, Hardware e Connettività.
Nel tempo sono stati corretti i bug del software sviluppato internamente ed è stato ottimizzato per essere il più possibile performante, inoltre sono stati superati i problemi derivati da un'errata configurazione dei software di terze parti utilizzati.
Anche i problemi di connettività sono stati superati, in quanto il collegamento a Internet della sede romana nel tempo è passato da una banda di 60Mb/s simmetrica garantita, nel 2007, a 1Gb/s simmetrico garantito.
I problemi legati all'hardware, invece, sono rimasti e nel tempo si sono accentuati, in quanto il parco macchine è diventato obsoleto, richiede un supporto tecnico sempre più presente e risente della mancanza di infrastrutture informatiche capaci di *adattasi automaticamente e velocemente* in caso di necessità.
Il passaggio ad un servizio in *cloud* permette di superare i problemi di hardware, in quanto il servizio è sempre aggiornato e mantenuto da personale dedicato e specializzato, l'aggiunta di nuovi dispositivi è un'operazione semplice e veloce e, in caso di necessità, è possibile *automatizzare* l'attivazione di nuovi dispositivi, precedentemente configurati, a supporto del sistema. Inoltre non ci sono limitazioni di banda, quindi anche il problema della connettività è risolto a monte.

1. Stato dell'arte

Attualmente l'intera infrastruttura informatica di *HSIT* è ospitata all'interno della *webfarm* della sede Romana, una *subnet*[5] dedicata a tutti i servizi web, esclusi quelli dedicati al servizio di sorveglianza sismica.

Il sistema si compone di:

- 1x Firewall con funzione di bilanciatore di carico.
- 3x Reverse Proxy con servizio cache.
- 1x Server Front-End (Web).
- 1x Server Back-End (Mappe).
- 1x Server DBMS.

[4] Per maggiori informazioni si rimanda al Rapporto Tecnico INGV 128/2010, "Realizzazione ed Evoluzione della versione 1.0 del Questionario Macrosismico online dell'INGV", consultabile all'indirizzo http://istituto.ingv.it/images/collane-editoriali/rapporti%20tecnici/rapporti-tecnici-2010/rapporto128.pdf

[5] Sottorete informatica, ripartizione logica di una rete informatica

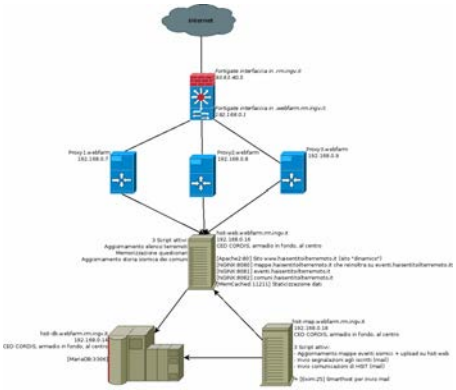


Figura 1 Infrastruttura informatica HSIT.

Figure 1 *HSIT informatic infrastructure.*

1.1 Bilanciatore di carico

Il bilanciatore di carico è un dispositivo, hardware o software, che distribuisce il carico di lavoro su differenti dispositivi.
Per la *subnet* della sede Romana è stato realizzato attraverso una funzione del firewall *Fortigate*[6]. Come da figura 1, la sua configurazione necessita un indirizzo IP su cui *rimanere in ascolto* per ricevere le richieste, un elenco di server a cui inoltrare le richieste, detto *BackEnd Pool*, di seguito *pool*, e un indirizzo IP nella *subnet* in cui risiede questo *pool* di server, per poter *comunicare* con essi.
Di conseguenza sono stati attivati:

- un'interfaccia ethernet virtuale a cui è stato assegnato un indirizzo IP pubblico, l'unico raggiungibile attraverso internet, a cui sono stati assegnati i vari nomi DNS di tutti i siti presenti nella *subnet webfarm*;
- un'interfaccia ethernet virtuale a cui è stato assegnato un indirizzo IP privato nella *subnet webfarm* che, oltre a permettere di comunicare con il *pool* di server da bilanciare, funge da *gateway* per tutti i dispositivi nella *subnet* per permettere la comunicazione verso l'esterno di quest'ultima (aggiornamento sistema operativo, sincronizzazione orario, invio mail, controllo remoto, ecc.);
- un *pool* di server a cui inoltrare tutto il traffico web HTTP (vedi *Reverse Proxy*, di seguito).

È stata attivata anche la funzionalità di *Health Check*, che permette al bilanciatore di carico di controllare, a intervalli regolari, il corretto funzionamento dei server presenti nel *pool* e, in caso di malfunzionamento, sospenderli temporaneamente dal *pool* fino al ripristino del servizio, in maniera completamente automatica.

1.2 Reverse proxy con servizio cache

È il *pool* di server a cui inoltrare le richieste.
Attualmente sono presenti 3 server configurati per funzionare come *Reverse Proxy* con *cache*[7].
Per realizzare il servizio è stato utilizzato il web server NGINX[8], configurato con 4 tipi differenti di *cache*, da scegliere a seconda del sito ospitato:

- *LongTime*, cache della durata di 30 giorni, per i siti aggiornati molto raramente;

[6] https://help.fortinet.com/fos50hlp/54/Content/FortiOS/fortigate-load-balancing-52/ldb.htm

[7] È una parte di disco dedicata da un server proxy http a salvare le pagine man mano caricate durante la navigazione, che potranno pertanto essere riproposte senza doverle chiedere di nuovo al sistema remoto. Cit. https://it.wikipedia.org/wiki/Cache#Web_cache

[8] NGINX è un web server/reverse proxy leggero ad alte prestazioni; fornisce rapidamente i contenuti statici con un utilizzo efficiente delle risorse di sistema. È possibile distribuire contenuti dinamici HTTP su una rete che utilizza i gestori FastCGI per gli script, e può servire come bilanciatore di carico

- *MediumTime*, cache della durata di 5 giorni, per i siti aggiornati saltuariamente;
- *ShortTime*, cache della durata di 1 minuto, per i siti aggiornati continuamente ma che necessitano del servizio di caching per ridurre il traffico in ingresso;
- *No cache*, per i siti aggiornati continuamente che non necessitano del servizio di caching.

Il loro funzionamento è semplice, dopo aver ricevuto la richiesta da parte del bilanciatore di carico, effettuano a loro volta una richiesta al server in cui è presente la risorsa richiesta. Dopo aver memorizzato la risposta del server nella propria *cache*, questa viene trasmessa al bilanciatore di carico che provvederà a inoltrarla al richiedente.
Da questo momento il proxy può soddisfare direttamente le successive richieste che richiedono la medesima risorsa, per tutto il tempo che questa è considerata *valida*.
HSIT sfrutta le potenzialità del servizio di caching *ShortTime* per fornire contenuti costantemente aggiornati ma rimanendo *difeso*, in termini di saturazione delle risorse, in caso di emergenza sismica in quanto la maggior parte delle richieste dei visitatori verrebbero soddisfatte dai *reverse proxy* che presenterebbero i dati presenti nella propria *cache*.
Si evince la loro utilità soprattutto durante un evento sismico in quanto migliaia di visitatori vogliono accedere alla medesima pagina web del terremoto appena avvenuto. Una pagina web può contenere anche 700Kb di testo e immagini. I reverse proxy effettuano la prima richiesta al server contenente la risorsa e per il minuto successivo (in quanto HSIT sfrutta il caching *ShortTime*) tutte le richieste vengono soddisfatte da loro. Ad esempio, contando 1000 richieste al minuto per ogni server proxy, solo 3 arriverebbero effettivamente al server web, una per proxy, con un trasferimento di dati di poco più di 2Mb, mentre ogni proxy si preoccuperebbe di soddisfare le altre 999, con un trasferimento di dati di circa 2Gb per proxy.

1.3 Server Web

Server di front-end che fornisce agli utenti i contenuti del progetto *HSIT*.
Nel corso degli anni il servizio web è stato suddiviso in tre servizi tematici, ripartendoli in altrettanti *Virtual Host*[9]:

- sito **www**, in cui i visitatori possono interagire, inviando segnalazioni, ricercando eventi, consultare pubblicazioni, notizie, ecc.;
- sito **eventi**, in cui si possono consultare gli elaborati relativi ad ogni evento sismico per cui sono presenti sufficienti dati;
- sito **comuni**, in cui si possono consultare gli elaborati dei risentimenti sismici a livello comunale.

Il primo sito, raggiungibile all'indirizzo *http://www.haisentitoilterremoto.it/*, è distribuito attraverso il web server Apache2, scelto sulla base della sua semplicità nell'utilizzo di linguaggi interpretati, necessari per fornire contenuti dinamici, semplicemente attivando il relativo modulo interno del servizio. Di contro si paga la sua versatilità con un minor numero di richieste simultanee gestibili, quindi un minor numero di visitatori simultanei.
Gli altri due servizi, raggiungibili agli indirizzi *http://eventi.haisentitoilterremoto.it/* e *http://comuni.haisentitoilterremoto.it/* rispettivamente, sono distribuiti dal web server NGINX, configurato con due *virtual host*.
È stato scelto questo web server in quanto permette di soddisfare un maggior numero di

[9] L'Hosting virtuale è un metodo usato sui server web per ospitare più siti web con differenti nomi di dominio sullo stesso server fisico condividendo, se necessario, anche stesso indirizzo IP. Cit.
https://it.wikipedia.org/wiki/Virtual_hosting

richieste simultanee, soprattutto per i contenuti statici[10], quali sono le pagine degli elaborati, sia degli eventi che dei comuni.
Recentemente, per alleggerire il carico del server di back-end,sono stati spostati i servizi di sincronizzazione del database degli eventi sismici tramite chiamate, a intervalli regolari, al webservices dell'ONT[11], e l'elaborazione e produzione dei contenuti statici relativi alle pagine comunali.

1.4 Server Mappe

È un server di back-end, non offre alcun servizio e non è raggiungibile dai visitatori. Si occupa di:

- elaborare i dati e produrre i contenuti statici relativi agli eventi sismici con cui poi aggiornare il server di front-end;
- inviare agli iscritti le notifiche relative ad un nuovo eventi sismico;
- pubblicare su Twitter la presenza di nuovi contenuti (solo la prima volta che viene realizzata la pagina di un evento sismico);
- inviare comunicazioni, da parte del gruppo *HSIT*, agli iscritti o parte di essi.

1.5 Server Database

Server di back-end in cui vengono memorizzati tutti i dati in formato strutturato. Durante l'attività di migrazione si è passati dal DBMS Mysql[12] a MariaDB[13], l'attività era già programmata da tempo ma con il sistema *HSIT* sempre operativo risultava difficile effettuare il cambio di DBMS senza dare lunghi disservizi; inoltre l'operazione doveva poter essere interrotta in caso di evento sismico.
Si è deciso, quindi, di sfruttare la migrazione verso il cloud per ultimare l'operazione in quanto il nuovo sistema è stato messo in funzione in parallelo, creando il minimo disservizio.
Le motivazioni per il cambio sono sia tecniche che *politiche*.
Dal punto di vista tecnico il DBMS MariaDB offre differenti tipi di motori a seconda delle necessità che, se correttamente utilizzati, permettono di ottenere ottime prestazioni su grandi set di dati, anche di tipo spaziale. Inoltre, pur aumentando la sicurezza del sistema, è stata mantenuta la sua semplicità di configurazione e utilizzo[14].
Dal punto di vista *politico*, a seguito dell'acquisizione della Sun Microsystems da parte della Oracle Corporation, si è sempre temuta la possibilità di un conflitto di interessi, da parte dell'azienda, in merito all'evoluzione di entrambi i DBMS *Mysql* e *Oracle*[15], il primo *free* e *open source*, il secondo a pagamento e *closed source*.

[10] Le pagine web statiche non coinvolgono, per definizione, alcuna attività di programmazione lato server, con linguaggi come ASP, .NET, Perl o PHP, ma necessita comunque della predisposizione del server web e della codifica delle pagine nei linguaggi interpretati dai browser: HTML, CSS e JavaScript. Ospitare un sito statico è tipicamente meno costoso, perché richiede minimi carichi di CPU, non necessita della presenza di database, inoltre e operazioni di messa in opera del sito consistono nella mera copia dei file su quest'ultimo.

[11] http://cnt.rm.ingv.it/webservices_and_software

[12] MySQL o Oracle MySQL, è un relational database management system (RDBMS) composto da un client a riga di comando e un server. https://www.mysql.com/it/

[13] MariaDB è un DBMS nato da un fork di MySQL, creato dal suo programmatore originale.
Per informazioni più dettagliate si rimanda a https://it.wikipedia.org/wiki/MariaDB

[14] Per un elenco aggiornato delle differenze tra i due DBMS si rimanda alle pagine ufficiali:
https://mariadb.com/kb/it/mariadb-versus-mysql-features/
https://mariadb.com/kb/en/optimizer-feature-comparison-matrix/
e ad un articolo di Digital Guide IONOS: https://www.ionos.it/digitalguide/hosting/tecniche-hosting/mariadb-vs-mysql/

[15] Oracle Database è uno tra i più famosi software di database management system sviluppato da Oracle Corporation. https://www.oracle.com

2. Migrazione verso il cloud Azure

Sfruttando le convenzioni già attive dell'Ente, si è deciso di utilizzare il servizio cloud *Microsoft Azure* in quanto già presente nella convenzione *CRUI*.
A seguito dell'attivazione di un account nella *subscription INGV* ed uno spazio di lavoro, chiamato *Gruppo di Risorse*, è possibile iniziare a creare la propria infrastruttura.
Previa autenticazione al portale è possibile iniziare a creare e gestire risorse attraverso interfaccia web o *Azure Cloud Shell*[16] (utilizzando la *Powershell*[17] o la *Bash*[18]). L'interfaccia web permette di configurare dispositivo per dispositivo in maniera semplice e intuitiva con un buon controllo su ciò che viene attivato. L'Azure Cloud Shell, di contro, permette di avere un controllo completo su ogni singolo dispositivo attivato, oltre a permettere l'utilizzo di script riutilizzabili per la creazione e configurazione in automatico dell'intera infrastruttura. Si è deciso, quindi, di utilizzare l'Azure Cloud Shell per la realizzazione e configurazione dell'infrastruttura e, successivamente, effettuare il *fine-tuning* della scalabilità via interfaccia web, dopo un primo periodo di funzionamento.
Nota. Per la stesura del presente documento è stato utilizzato quasi esclusivamente il dominio *hsit.it*, in quanto ancora non divulgato, quindi tutti i test sono stati effettuati senza arrecare alcun disservizio alla piattaforma ufficiale, raggiungibile al dominio *haisentitoilterremoto.it*.

2.1 La nuova infrastruttura informatica

Per effettuare una migrazione *indolore* si è deciso di ricalcare la struttura attuale ma adattandola ai servizi offerti per sfruttare al meglio le peculiarità del cloud.
Inoltre, si è deciso di sfruttare l'operazione di migrazione per passare tutta la comunicazione su protocollo cifrato HTTPS che, come vedremo più avanti, oltre ad aumentare la sicurezza della web application porta con sé numerose migliorie.
Occorre quindi sapere quali sono i dispositivi virtuali disponibili nell'offerta MS Azure e i loro costi per valutare cosa attivare.
È possibile stimare preventivamente costi e dimensioni delle *virtual machine*, di seguito *VM*, utilizzando il *Calcolatore dei costi*[19] oppure, sempre via web, direttamente durante l'attivazione dei vari dispositivi virtuali.
Dopo un'analisi dei dispositivi virtuali offerti e delle loro funzionalità, si è deciso di mantenere invariata la struttura ad eccezione del *bilanciatore di carico* da sostituire con un *application-gateway* in quanto il primo opera a livello 4 della pila ISO/OSI e non permette un corretto utilizzo dei certificati SSL (necessari per attivare il protocollo HTTPS), come descritto successivamente.
In figura 2 lo schema della nuova infrastruttura informatica.
Sono quindi necessari:

- Certificati SSL, da preparare precedentemente e richiederne la *certificazione*
- Accesso al portale Azure
- Un *Gruppo di risorse*
- Una rete virtuale divisa in almeno 2 subnet e un indirizzo IP pubblico
- Un application gateway, in seguito *AG*
- Una coppia di chiavi SSH da dedicare all'infrastruttura
- Un'immagine ISO da utilizzare come modello (*Template*) per le *VM* del set di scalabilità

[16] Azure Cloud Shell è una shell interattiva, autenticata e accessibile tramite browser per la gestione delle risorse di Azure.

[17] https://it.wikipedia.org/wiki/Windows_PowerShell

[18] https://it.wikipedia.org/wiki/Bash

[19] Raggiungibile all'indirizzo web: https://azure.microsoft.com/it-it/pricing/calculator/

- Un set di scalabilità, in seguito *VMSS*
- Regole di redirect da protocollo HTTP a HTTPS
- 3x Server HSIT

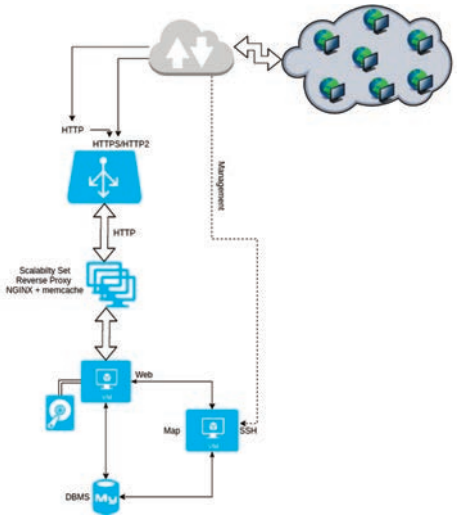


Figura 2 Infrastruttura informatica HSIT da realizzare in cloud.

Figure 2 *HSIT informatic infrastructure to realize in cloud platform.*

2.2 Certificati SSL

Citando il sito DigiCertCA[20] che ha firmato i certificati[21] di *HSIT*:

"*Le credenziali SSL (Secure Protocol) (a volte chiamate credenziali digitali) stabiliscono un collegamento crittografato tra un browser o un computer e un server o una rete. Ad ogni sessione, il collegamento SSL protegge le informazioni della carta di credito e altre informazioni che non vengono intercettate da soggetti non autorizzati.*
Un programma invisibile agli utenti finali si chiama "stretta di mano SSL", che crea una connessione sicura tra il server Web e il browser. Utilizza una chiave tripla per creare una chiave di sessione simmetrica, che a sua volta crittografa tutti i dati trasmessi.

1. *Il server invia la propria chiave pubblica simmetrica al browser.*
2. *Il browser crea una chiave di sessione simmetrica e la crittografa con la chiave pubblica simmetrica del server e la invia al server.*
3. *Il server decodifica la chiave di sessione crittografata con la propria chiave privata simmetrica per ottenere una chiave di sessione simmetrica.*
4. *Il server e il browser ora crittografano e decrittografano tutti i dati trasmessi con una chiave di sessione simmetrica. Questo programma garantisce la sicurezza del canale perché solo il browser e il server conoscono la propria chiave di sessione simmetrica e questo set di chiavi di sessione può essere utilizzato solo per questa sessione. Se il browser si collega allo stesso server a giorni alterni, dovrà generare una nuova chiave di sessione.*"

Figura 3 Esempio di comunicazione SSL.

Figure 3 *Exampe of SSL communication.*

[20] https://www.digicert.com/it/ssl-certificate/#What_is_an_SSL_Certificate
[21] https://it.wikipedia.org/wiki/Certificato_digitale

Aprendo un terminale sul server sono stati creati due certificati da sottomettere a *DigiCertCA*, uno per il dominio *haisentitoilterremoto.it* e uno per il suo acronimo *hsit.it*.
Nota. La coppia di simboli " \↲ " indica di inserire il carattere *backslash* seguito da *Invio*, questo permette di *spezzare* il comando su più righe.

```
$ openssl req \↲
      -new \↲
      -newkey rsa:2048 \↲
      -nodes \↲
      -out www_haisentitoilterremoto_it.csr \↲
      -keyout www_haisentitoilterremoto_it.key \↲
       -subj "/C=IT/ST=RM/L=Roma/O=Istituto Nazionale di Geofisica e Vulcanologia/=Hai
Sentito il Terremoto?/CN=www.haisentitoilterremoto.it"
```

```
$ openssl req \↲
      -new \↲
      -newkey rsa:2048 \↲
      -nodes \↲
      -out www_hsit_it.csr \↲
      -keyout www_hsit_it.key \↲
       -subj "/C=IT/ST=RM/L=Roma/O=Istituto Nazionale di Geofisica e Vulcanologia/=Hai
Sentito il Terremoto?/CN=www.hsit.it"
```

Ognuno di questi comandi crea 2 file, il *Certificate Signing Request*, solitamente con estensione *.csr*, e la *Chiave Privata*, solitamente estensione *.key*, da sottomettere al certificatore che, dopo aver controllato l'effettivo possesso del dominio indicato nella richiesta, *firmerà* il certificato inviando al richiedente un ulteriore file contenente il certificato *firmato* dalla *Certification Authority*, la Chiave Pubblica e altre informazioni sul dominio (solitamente un file con estensione *.crt o .cert o .cer*).
Purtroppo l'*AG* di Azure non supporta il certificato in questo formato, quindi deve essere preventivamente convertito in formato *PKCS#12*[22] (solitamente con estensione *.pfx*), contenente sia il certificato che la chiave privata, protetti da password.
Si procede quindi con la conversione dei due certificati in formato *PKCS#12*, durante la conversione verrà richiesto di inserire la password di protezione:

```
$ openssl pkcs12 \↲
      -export \↲
      -out cert-hsit-azure.pfx \↲
      -inkey www_hsit_it.key \↲
      -in www_hsit_it.crt
```

```
$ openssl pkcs12 \↲
      -export \↲
      -out cert-haisentitoilterremoto-azure.pfx \↲
      -inkey www_haisentitoilterremoto_it.key \↲
      -in www_haisentitoilterremoto_it.crt
```

[22] https://en.wikipedia.org/wiki/PKCS_12

Abbiamo, adesso, anche i seguenti file contenenti i rispettivi certificati:

1. cert-haisentitoilterremoto-azure.pfx;
2. cert-hsit-azure.pfx.

in formato *PKCS#12*.

2.3 Accesso al portale Azure

Come detto in precedenza si è preferito utilizzare la potenza di *Azure Cloud Shell*, sfruttando la shell *Bash*, per realizzare e configurare l'intera infrastruttura virtuale.
Per accedere al terminale occorre prima autenticarsi al portale Azure all'indirizzo https://portal.azure.com e cliccare sull'icona cerchiata in rosso, come da figura 4, che attiva la funzione Azure Cloud Shell.

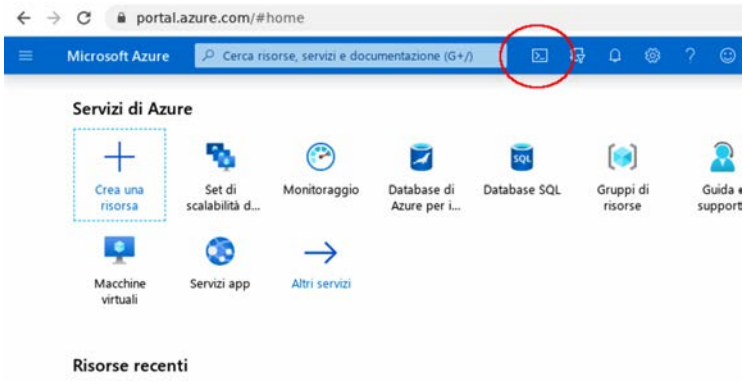


Figura 4 Icona per accedere alla Azure Cloud Shell.

Figure 4 *Azure Cloud Shell icon.*

Al primo accesso alla Azure Cloud Shell viene richiesto di creare una *risorsa di archiviazione*, come mostrato in figura 5.
Cliccando su *Mostra impostazioni avanzate* vengono mostrate le opzioni per personalizzare il dispositivo da attivare, come mostrato in figura 6.
Nel dettaglio viene richiesto:

1. Il *Gruppo di risorse* in cui inserirlo.
2. Il nome dell'*account di archiviazione* da creare (sono consentiti solo caratteri alfanumerici minuscoli e trattini).
3. Il nome dell'*account di condivisione file* da creare (sono consentiti solo caratteri alfanumerici minuscoli e trattini).

In caso non si disponga di queste tre risorse è possibile crearle sul momento.

Tempistiche
L'operazione di creazione richiede tra i 2 e i 5 minuti.

Una volta cliccata viene aperta la finestra della shell, come in figura 7.
In caso si preferisca utilizzare la *PowerShell*, come da figura 8, basta selezionarla direttamente nel menù della shell.

Figura 5 Richiesta di attivazione di un account di archiviazione.

Figure 5 *Request for archive account activation.*

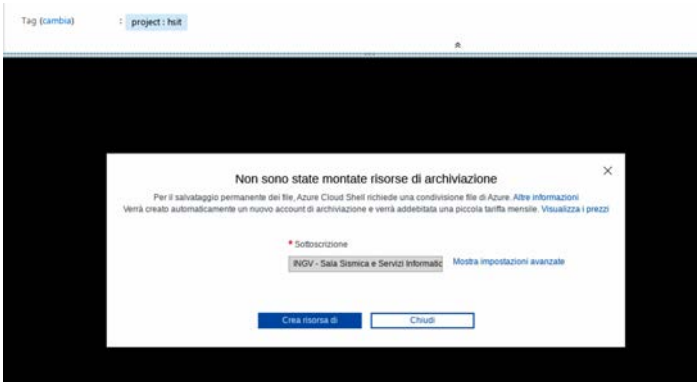


Figura 6 Personalizzazione dell'account di archiviazione.

Figure 6 *Customizing archive account.*

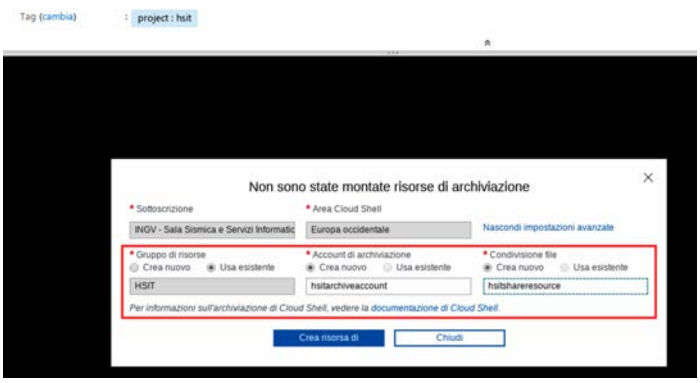


Figura 7 Azure Cloud Shell.

Figure 7 *Azure Cloud Shell.*

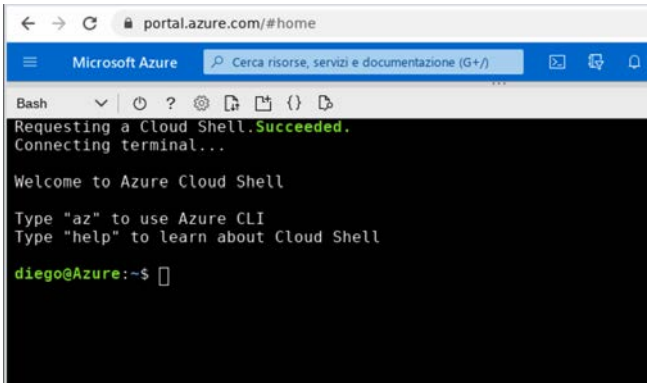
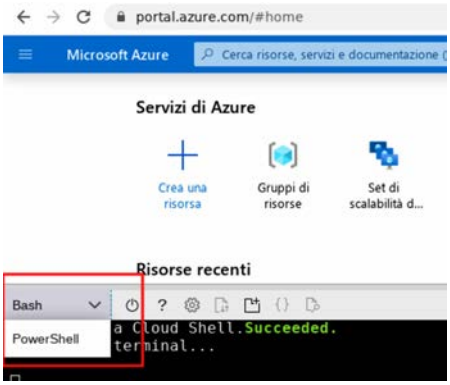


Figura 8 Icona per passare tra Bash e Powershell.

Figure 8 *Bash and Powershell switch icon.*



2.4 Creazione di un Gruppo di Risorse

Per poter creare dispositivi virtuali nel cloud Azure è indispensabile un contenitore, detto *Gruppo di Risorse*.
Se non fosse già stato fornito dal proprio Amministratore IT è necessario crearne uno, ricordando che è un *privilegio* che deve essere assegnato al proprio account Azure.

Come da figura 9 il *Gruppo di Risorse* assegnato al progetto *Hai Sentito il Terremoto?* è chiamato *HSIT*.

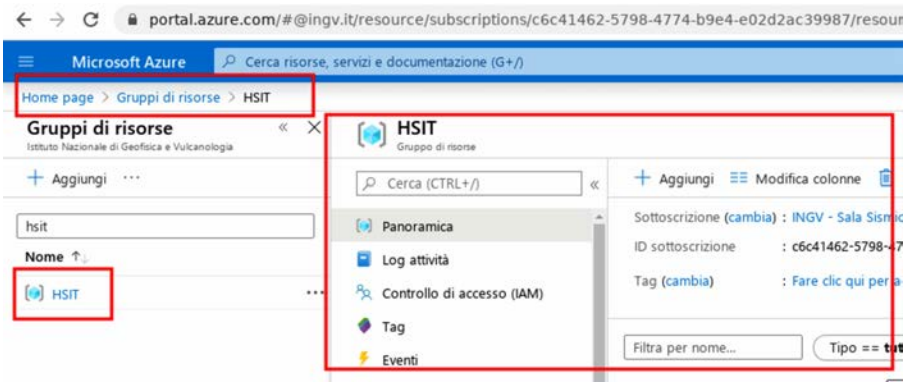


Figura 9 Gruppo di Risorse.

Figure 9 *Resouce Group.*

Altrimenti, per creare un *Gruppo di Risorse*, aprire la console Azure cliccando sull'icona cerchiata in rosso per accedere alla console e digitare:

```
$ az group create \↲
        —name HSIT \↲
        —location westeurope
```

il comando crea un *Gruppo di Risorse*, nel nostro caso chiamato *HSIT*, nella località *Europa Occidentale*.
Nota. Se si stanno effettuando prove, il modo più veloce per eliminare con sicurezza tutti i dispositivi virtuali creati è la rimozione completa del *Gruppo di Risorse*:

```
$ az group delete \↲
        —name HSIT \↲
```

2.5 Creazione di una rete virtuale

Per poter creare un'infrastruttura informatica è necessario attivare una rete informatica.
Per un sistema informativo basilare basta creare una rete con una sola *subnet*, anche con lo stesso indirizzamento IP ma, nel caso dell'infrastruttura di *HSIT* è necessario realizzare due differenti *subnet* in quanto l'*AG*, per impostazioni Microsoft, deve necessariamente risiedere su una *subnet* separata e dedicata.
Dal terminale Azure, digitare i seguenti comandi:

```
$ az network vnet create \↲
      —name HSITVNet  \↲
      —resource-group HSIT  \↲
      —address-prefix 10.0.0.0/22 \↲
      —subnet-name HSITAgSubnet \↲
      —subnet-prefix 10.0.0.0/24
```

Per una rete con nome *HSITVNet* con indirizzamento 10.0.0.0/22 con indirizzamento disponibile da 10.0.0.0 a 10.0.3.255 e al suo interno crea la sottorete *HSITAgSubnet* dedicata all'*AG* con indirizzamento 10.0.0.0/24:

```
$ az network vnet subnet create \↲
      —name HSITServersSubnet \↲
      —resource-group HSIT  \↲
      —vnet-name HSITVNet  \↲
      —address-prefix 10.0.1.0/24
```

Per creare la sottorete *HSITServersSubnet* dedicata ai server, con indirizzamento 10.0.1.0/24.

```
$ az network public-ip create \↲
      —resource-group HSIT  \↲
      —name HSITAgPublicIP \↲
      —dns-name hsit
```

Per creare un indirizzo IP pubblico, con nome *HSITAgPublicIP* da assegnare all'*AG*, a cui viene assegnato il nome di dominio *hsit*, raggiungibile via internet attraverso il *FQDN*[23] hsit.*westeurope.cloudapp.azure.com* (.westeurope.cloudapp.azure.com è il dominio dedicato per l'*Europa Occidentale* di Azure).
Assegnare un nome di dominio all'indirizzo IP è necessario in quanto, in alcuni casi particolari, quest'ultimo potrebbe essere modificato a nostra insaputa, quindi la risoluzione del nome assicura sempre la raggiungibilità della web-application.

Tempistiche
La creazione della rete e il suo *subnetting* ha richiesto meno di un minuto.

2.6 Creazione dell'Application Gateway

Citando il manuale di Azure[24]:

"*Il gateway applicazione di Azure è un servizio di bilanciamento del carico del traffico Web che consente di gestire il traffico verso le applicazioni Web. I servizi di bilanciamento del carico tradizionali operano a livello di trasporto (OSI livello 4 - TCP e UDP) ed eseguono il routing del traffico da un indirizzo IP e una porta di origine verso un indirizzo IP e una porta di destinazione. Il gateway applicazione consente di prendere decisioni relative al routing basate su altri attributi di una richiesta HTTP, ad esempio il percorso dell'URI o le intestazioni host. Ad esempio, è possibile eseguire il routing del traffico in base all'URL in ingresso.*
Questo tipo di routing è detto bilanciamento del carico a livello di applicazione (OSI livello 7). Il gateway applicazione di Azure può eseguire il routing basato su URL e molto altro."

L'*AG*, come detto precedentemente, è l'unico dispositivo raggiungibile via web dai visitatori, in quanto è l'unico servizio web ad avere un indirizzo IP pubblico.

[23] *Fully Qualified Domain Name* è un nome di dominio non ambiguo che specifica la posizione assoluta di un nodo all'interno della gerarchia dell'albero DNS. Per distinguere un FQDN da un nome di dominio standard si aggiunge il nome dell'host alla stringa del dominio, in modo da renderla assoluta.

[24] https://docs.microsoft.com/it-it/azure/application-gateway/overview

Il certificato SSL deve, quindi, essere installato su questo dispositivo.
Internamente il gateway inoltra le richieste ai vari servizi web nella *subnet* privata ma è l'*AG* che instaura la comunicazione con il browser del visitatore.
Se avessimo adottato, nella configurazione attuale, un bilanciatore di carico, l'intera procedura di cifratura sarebbe passata in carico al server interno alla *subnet* privata ma, visto che la comunicazione è sempre tra bilanciatore e browser, sarebbe risultata comunque una connessione cifrata ma il browser avrebbe segnalato un'anomalia nella cifratura della sessione, definendo la comunicazione con il sito non sicura. Inoltre, il bilanciatore di carico, per tipologia di dispositivo, non assicura il mantenimento della sessione[25].

Tra le varie funzionalità dell'*AG* troviamo la possibilità di utilizzare il protocollo HTTP/2[26] che, tra le migliorie di nostro interesse troviamo:

- l'invio delle risorse in formato binario che, rispetto al formato testuale di HTTP/1.1, è più compatto ed è in grado di incrementare la velocità di caricamento delle pagine;
- i dati da caricare sono suddivisi in più parti ed inviati al browser in un'unica sessione, sarà così compito del browser assemblare i vari frammenti;
- non è più necessario attendere l'analisi del codice HTML da parte del browser, il server *anticipa* le richieste del client, inviando direttamente al browser gli elementi nella cache del client.

Prima di creare il dispositivo è necessario caricare la chiave nella nostra *home* su Azure, cliccando l'icona di scambio file, come da figura 10.
Si procede con la creazione dell'*AG* da terminale:

```
$ az network application-gateway create \↲
    —name HSITAg \↲
    —resource-group HSIT \↲
    —capacity 2 \↲
    —cert-file cert-hsit-azure.pfx \↲
    —cert-password PFX_PASSWORD \↲
    —frontend-port 443 \↲
    —http-settings-cookie-based-affinity Disabled \↲
    —http-settings-port 80 \↲
    —http-settings-protocol Http \↲
    —http2 Enabled \↲
    —public-ip-address HSITAgPublicIP\↲
    —sku Standard_Medium \↲
    —subnet HSITAgSubnet \↲
    —vnet-name HSITVNet
```

L'*AG* supporta un solo certificato quindi, per attivare entrambi i domini *HSIT*: *haisentitoilterremoto.it* e il suo acronimo *hsit.it* sarà necessario creare un'ulteriore *AG* con indirizzo IP pubblico e *subnet* dedicata, come precedentemente illustrato, modificando i nomi delle risorse.

Tempistiche
La creazione dell'AG ha richiesto fino a 20 minuti.

[25] Le sessioni permettono di mantenere informazioni arbitrarie durante la navigazione tra le varie pagine navigate.

[26] HTTP/2 è la nuova versione del protocollo di rete HTTP usato dal World Wide Web, basato su SPDY/3 di Google.

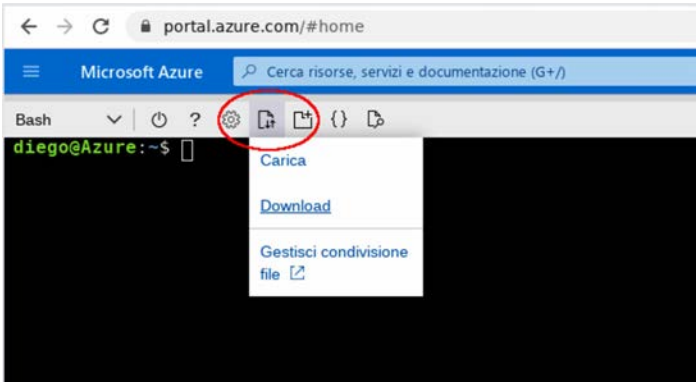


Figura 10 Opzioni per scambiare file con la console Azure.

Figure 10 *Azure file sharing options.*

2.7 Opzione Carica, Download e Gestisci condivisione file

Le due opzioni *Carica* e *Download* sono comode per caricare o scaricare velocemente un singolo file alla volta.
Se è necessario aggiungere numerosi file contemporaneamente risulta più comodo l'utilizzo della terza opzione.
Cliccando su *Gestisci condivisione file* si apre una finestra di condivisione file, come riportato in figura 11.

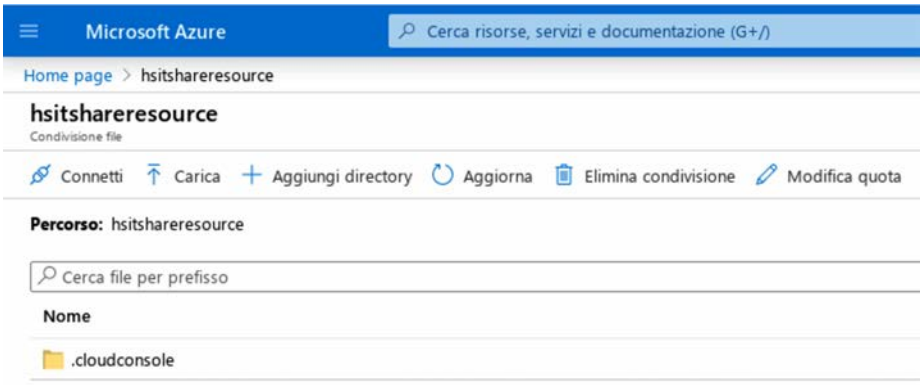


Figura 11 Interfaccia web per la Gestione della condivisione dei file.

Figure 11 *Share file web interface.*

Cliccando su *Connetti* è possibile *montare* il disco di rete Azure direttamente sul proprio PC per scambiare file velocemente. Si aprirà, a destra, un frame contenente le istruzioni multipiattaforma per montare il disco. Selezionando la piattaforma preferita, come in figura 12, viene aggiornato il riquadro in grigio con i passaggi da compiere.
Se, invece, si vuole caricare direttamente alcuni file, basta cliccare sull'opzione *Carica* che, come da figura 13, aprirà un riquadro a destra contenente il menù di *upload* dei files. Cliccando sull'icona in blu viene aperto il gestore di file del proprio PC da cui poter selezionare i dati da caricare sul sistema.

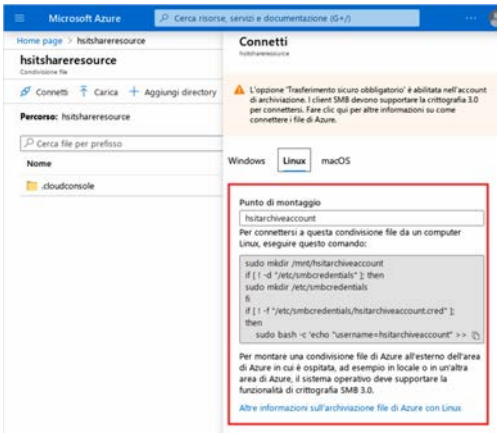


Figura 12 Gestione condivisione file, istruzioni per il collegamento del disco di rete.

***Figure 12** Share file, mount net drive instructions.*

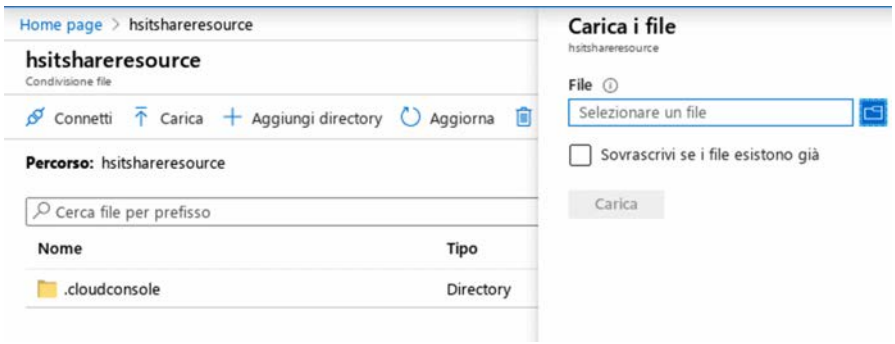


Figura 13 Gestione condivisione file, interfaccia di Upload files.

***Figure 10** Share file, Upload file interface.*

Una volta caricati i file è possibile ritrovarli nella propria Azure Cloud Shell nella directory *clouddrive/* come mostrato in figura 14.

Figura 14 Collegamento alla directory di Gestione condivisione file.

***Figure 14** Azure Cloud Shell, symbolic link to Share file directory.*

2.8 Chiavi SSH per collegarsi alle VM

Per tutte le operazioni di configurazione e manutenzione delle *VM* è stato deciso di creare una coppia di chiavi SSH dedicate.
Per creare le chiavi SSH abbiamo utilizzato l'algoritmo *RSA* in quando l'unico, al momento, supportato dalla piattaforma Azure[27].
Quindi, dal terminale di Azure Cloud Shell, eseguire il comando:

```
$ ssh-keygen -m PEM -t rsa -b 4096 -f azure-rsa.key
```

che creerà due files:

1. azure-rsa.key, la chiave privata che deve rimanere segreta;
2. azure-rsa.key.pub, la chiave pubblica da aggiungere in ogni *VM* in cui vogliamo poterci collegare.

Si può scaricare la chiave sul proprio PC dal pannello di controllo di Azure Cloud Shell, cliccando sull'icona per lo scambio file, come da figura 10, e digitare il nome del file da scaricare.

Tempistiche
La creazione delle chiavi SSH ha richiesto pochi secondi.

2.9 Creazione di un modello per il set di scalabilità

Seguendo il nostro progetto, dopo aver creato il punto di accesso al sistema occorre creare una *VMSS* con una serie di istanze di macchine virtuali.
Prima, però, abbiamo creato un *template* di *VM (Virtual Machine)*, preconfigurata, che verrà utilizzata dal *VMSS* come immagine di base per creare le istanze necessarie.
La procedura per creare un *template* di macchina virtuale occorre seguire i seguenti passaggi:

1. creare una macchina virtuale base e/o preconfigurata;
2. effettuare il deprovisioning, rimuovendo gli account utente e la configurazione di rete;
3. deallocare la macchina, arrestando la *VM* e rilasciando le risorse di calcolo;
4. generalizzare la *VM*, impostando lo stato del sistema operativo della macchina virtuale su *generalizzato*;
5. creare l'immagine personalizzata della *VM*.

[27] https://docs.microsoft.com/en-us/azure/virtual-machines/linux/mac-create-ssh-keys

2.10 Creazione di una macchina virtuale base e/o preconfigurata

Dalla console di Azure Cloud Shell lanciare il comando:

```
$ az vm create \↲
    —verbose \↲
    —resource-group HSIT \↲
    —name TemplateScalingVM \↲
    —size Standard_B2s\↲
    —image UbuntuLTS\↲
    —vnet-name HSITVNet\↲
    —subnet HSITServersSubnet\↲
    —private-ip-address 10.0.0.10/24↲
   —public-ip-address TemplateScalingVM-public-ip \↲
    —custom-data ./init_image_to_generalize.yml \↲
    —admin-username diego.sorrentino\↲
    —ssh-key-value @azure-rsa.key.pub
```

per creare una *VM* nel Gruppo di risorse *HSIT* con nome *TemplateScalingVM*, di tipo *UbuntuLTS*[28,29], con dimensione *Standard_B2s*[30], nella *subnet HSITServersSubnet* della rete *HSITVNet*, con indizzo IP *10.0.0.10* e netmask *255.255.255.0*, utente *diego.sorrentino* che potrà accedere alla *VM* esclusicamente tramite scambio di chiavi.
Il flag *— custom-data* permette di configurare automaticamente la *VM* attraverso una serie di comandi descritti nel file che viene passato come argomento, in questo caso *init_image_to_generalize.yml*, nel formato *YAML*[31].
Nel caso in cui sia stato impostato il comando per configurare automaticamente la *VM*, durante l'installazione del sistema operativo viene installato il pacchetto cloud-init che, al primo avvio della macchina, interpreta ed esegue la procedura di configurazione.
Trattandosi del *template* delle *VM* che svolgeranno il compito di *reverse proxy* la configurazione consiste in:

1. Aggiornare il sistema operativo.
2. Installare i pacchetti NGINX e NtpDate.
3. Installare una configurazione personalizzata di NGINX e aggiungere i *Virtual Hosts* necessari.
4. Riavviare il daemon di NGINX con la nuova configurazione.
5. Aggiungere alle procedure orarie automatiche la sincronizzazione dell'orario.

Di seguito la configurazione, epurata dai file di configurazione customizzati:

[28] Per un elenco delle immagini disponibili eseguire il comando: $ az vm image list - output table

[29] È stata scelta l'immagine UbuntuLTS in quanto l'immagine Debian ancora non supporta la configurazione automatica. https://docs.microsoft.com/en-us/azure/virtual-machines/linux/using-cloud-init

[30] Per un elenco delle dimensioni disponibili eseguire il comando: az vm list-sizes - location westeurope - output table

[31] https://it.wikipedia.org/wiki/YAML

```
#cloud-config
package_upgrade: true
packages:
  - nginx
  - ntpdate
write_files:
-   content: |
      CONFIGURAZIONE DI NGINX
    owner: root:root
    permission: '0644'
    path: /etc/nginx/nginx.conf
-   content: |
       server {
      CONFIGURAZIONE DA APPLICARE PER OGNI VIRTUAL HOST
    owner: root:root
    permission: '0644'
    path: /etc/nginx/sites-enabled/www.hsit.it.conf
runcmd:
  - service nginx restart
```

Tempistiche
La creazione della *VM* ha richiesto appena 2 minuti. L'aggiornamento del sistema e la sua configurazione al primo avvio ha richiesto quasi 10 minuti.

2.11 Deprovisioning, deallocazione, generalizzazione della VM e creazione dell'immagine personalizzata

Come anticipato, il deprovisioning effettua la rimozione degli utenti e delle impostazioni di rete per rendere la *VM* riutilizzabile in maniera automatica.
L'operazione può essere inserita sia nel file di configurazione *YAML* che eseguita manualmente dopo aver controllato che la *VM* sia stata correttamente configurata.
Nel nostro caso il deprovisioning è stato effettuato manualmente, previo collegamento alla console traminte SSH.
Per conoscere l'indirizzo IP pubblico della *VM* appena creata basta *interrogare*[32] Azure:

```
$ VM_VIRTUAL_IP=`az network public-ip show \↲
      —resource-group HSIT \↲
      —name TemplateScalingVM-public-ip  \↲
      —query [ipAddress] \↲
      —output tsv`

$ ssh -i azure-rsa.key diego.sorrentino@${VM_VIRTUAL_IP}
$ sudo waagent -deprovision+user -force && export HISTSIZE=0 && sync && shutdown -h now
```

[32] Le risposte vengono date in formato JSON, per *interrogare* Azure si può fare riferimento al *query language JMESPath*, https://jmespath.org/

Completato il deprovisioning della *VM* si continua con la deallocazione delle risorse:

```
$ az vm deallocate \↲
      —resource-group HSIT \↲
      —name TemplateScalingVM
```

si passa a generalizzare la *VM*:

```
$ az vm generalize \↲
      —resource-group HSIT \↲
      —name TemplateScalingVM
```

e, finalmente, si può creare l'immagine base per il *VMSS*:

```
$ az image create \↲
      —resource-group  HSIT \↲
      —name HSIT_ReverseProxyTPL \↲
      —source TemplateScalingVM
```

Prima di continuare, per motivi economici, conviene *fare pulizia* eliminando il dispositivo virtuale appena creato *TemplateScalingVM* e tutte le sue risorse correlate.

Rimuovere la *VM:*

```
$ az vm delete \↲
        —resource-group HSIT \↲
        —name TemplateScalingVM \↲
        —yes
```

Rimuovere la scheda di rete, solitamente creato con il nome della *VM* con suffisso *VMNIC:*

```
$ az network nic delete \↲
        —resource-group HSIT \↲
        —name TemplateScalingVMVMNic
```

Rimuovere l'indirizzo IP pubblico, solitamente creato con il nome della *VM* con suffisso *-public-ip:*

```
$ az network public-ip delete \↲
        —resource-group HSIT \↲
        —name TemplateScalingVM-public-ip
```

Per rimuovere il disco del sistema operativo occorre prima conoscerne il nome in quanto questo viene creato in maniera non predicibile.
Quindi, sempre dalla console, interroghiamo Azure sul nome assegnato al disco e lo assegniamo alla variabile *DiskName*:

```
$ DiskName=`az vm show \↲
        —resource-group HSIT \↲
        —name TemplateScalingVM \↲
        —query storageProfile.osDisk.name \↲
        —output tsv`

$ az disk delete \↲
        —resource-group HSIT \ ↲
        —name $DiskName
```

Infine eliminiamo il Network Security Group, solitamente creato con il nome della *VM* con suffisso *NSG*:

```
$ az network nsg delete \↲
        —resource-group HSIT \↲
        —name TemplateScalingVMNSG
```

Tempistiche
Tutte le operazioni necessarie alla realizzazione dell'immagine e la pulizia delle risorse hanno richiesto appena 5 minuti.

2.11.1 Creazione di un Network Security Group (NSG)

Durante la creazione di una *VM*, in automatico, viene creato un *Gruppo di sicurezza di rete* che consente di limitare l'accesso al dispositivo, una sorta di firewall, con nome *<NOME-VM>-nsg*
Le regole applicate di base consistono in:

- *INPUT*, permettere il totale accesso alla risorsa all'interno rete Azure di appartenenza;
- *INPUT*, permettere l'accesso SSH via Internet **SE** la *VM* è stata creata con un indirizzo IP pubblico;
- *OUTPUT*, permettere la totale navigazione della *VM* sia all'interno della propria rete Azure che su Internet.

All'occorrenza è possibile configurare, in maniera estremamente dettagliata, le restrizioni della VM in rete.

2.12 Creazione del Set di Scalabilità

A questo punto abbiamo tutte le risorse create e *nominate* per poter procedere con la creazione del *VMSS*.
La configurazione richiede di specificare il *Gruppo di risorse* in cui creare la nuova risorsa, la *rete* e la *subnet* in cui posizionarla, l'*AG* di riferimento, il *pool* di backend in cui essere inserito, il tipo di *VM-SKU* da creare (*Stock-Keeping Unit*), l'immagine del sistema operativo da utilizzare, il numero di istanze da creare e l'utente amministratore da creare e in quale modo potrà autenticarsi alle varie istanze delle *VM*.

```
$ az vmss create \↲
  —name HSITScaleSet \↲
  —resource-group HSIT \↲
  —image HSIT_ReverseProxyTPL \↲
  —admin-username diego.sorrentino \↲
  —ssh-key-value @azure-rsa.key.pub \↲
  —instance-count 2 \↲
  —vnet-name HSITVNet\↲
  —subnet HSITServersSubnet\↲
  —vm-sku Standard_DS2\↲
  —upgrade-policy-mode Automatic \↲
  —app-gateway HSITAg\↲
  —backend-pool-name appGatewayBackendPool
```

Tempistiche
La creazione del *VMSS* ha richiesto poco meno di 10 minuti.

2.13 Redirect da protocollo HTTP a HTTPS

Come anticipato, si è deciso di mettere in sicurezza la comunicazione passando da protocollo HTTP, per standard raggiungibile su porta 80, a HTTPS, per standard raggiungibile su porta 443, ma il portale è attualmente *conosciuto* come raggiungibile esclusivamente tramite protocollo HTTP, porta 80.
Occorre quindi redirezionare tutte le richieste dalla porta 80 alla porta 443, indicando anche il cambio di protocollo.
Si procede creando una nuova porta sull'*AG* che riceva le richieste sulla porta 80, metterla *in ascolto* e redirezionare il traffico in ingresso.

Dal terminale di Azure Cloud Shell:

```
$ az network application-gateway frontend-port create \↲
      —port 80 \↲
      —gateway-name HSITAg \↲
      —resource-group HSIT \↲
      —name port_80
```

```
$ az network application-gateway http-listener create \↲
      —name ListenerHttpPort \↲
      —frontend-ip appGatewayFrontendIP \↲
      —frontend-port port_80\↲
      —gateway-name HSITAg \↲
      —resource-group HSIT
```

```
$ az network application-gateway redirect-config create \↲
    —name httpToHttps \↲
    —gateway-name HSITAg \↲
    —resource-group HSIT \↲
    —type Permanent \↲
    —target-listener appGatewayHttpListener \↲
    —include-path true \↲
    —include-query-string true
```


```
$ az network application-gateway redirect-config create \↲
    —name httpToHttps \↲
    —gateway-name HSITAg \↲
    —resource-group HSIT \↲
    —type Permanent \↲
    —target-listener appGatewayHttpListener \↲
    —include-path true \↲
    —include-query-string true
```


Tempistiche
La creazione delle regole di routing ha richiesto fino a 25 minuti.

2.14 Creazione server HSIT

Finita la preparazione dell'infrastruttura si può finalmente passare alla creazione e configurazione dei server del progetto.
Come da figura 2, il progetto si compone di 3 server, tutti con configurazioni molto differenti. Anche in questo caso si è preferito automatizzare la creazione e configurazione, così da poter aggiornare, sostituire o replicare le *VM* senza dover effettuare ogni volta la configurazione manualmente.

Dal terminale di Azure Cloud Shell:

```
# Creazione DB Server
$ az vm create \↲
    —verbose \↲
    —resource-group HSIT \↲
    —name hsit-db \↲
    —size Standard_D2_v3 \↲
    —image UbuntuLTS \↲
    —vnet-name HSITVNet \↲
    —subnet HSITServersSubnet \↲
    —private-ip-address 10.0.1.20 \↲
    —custom-data ./init_hsit_db.yml  \↲
    —admin-username diego.sorrentino \↲
    —ssh-key-value @azure-rsa.key.pub
```

```
# Creazione Map server + 1 IP pubblico per Mail Server e operazioni di management
$ az vm create \↲
      —verbose \↲
      —resource-group HSIT \↲
      —name hsit-map\↲
      —size Standard_D2_v2 \↲
      —image UbuntuLTS \↲
      —vnet-name HSITVNet \↲
      —subnet HSITServersSubnet \↲
      —private-ip-address 10.0.1.21 \↲
      —custom-data ./init_hsit_map.yml \↲
      —admin-username diego.sorrentino \↲
      —ssh-key-value @azure-rsa.key.pub \↲
      —public-ip-address hsit-map-ip \↲
      —public-ip-address-dns-name management
```

```
# Creazione Web server + 1 disco di storage da 1024G (1Tb)
$ az vm create \↲
      —verbose \↲
      —resource-group HSIT \↲
      —name hsit-web \↲
      —size Standard_B4ms \↲
      —image UbuntuLTS \↲
      —vnet-name HSITVNet  \↲
      —subnet  HSITServersSubnet \↲
      —private-ip-address 10.0.1.22 \↲
      —custom-data ./init_hsit_web.yml \↲
      —admin-username diego.sorrentino \↲
      —ssh-key-value @azure-rsa.key.pub \↲
      —data-disk-sizes-gb 1024
```

Come per la creazione del *template* per le *VM* del *VMSS*, abbiamo passato una configurazione, redatta in formato *YAML*, personalizzata per ogni singola *VM*, a seconda dei software che dovranno essere presenti.
Notare le personalizzazioni della *VM* di front-end che quella di back-end.
Alla *VM* di front-end è stato agganciato un ulteriore disco in cui vengono memorizzati tutti gli elaborati.
Alla *VM* di back-end è stato aggiunto un indirizzo IP pubblico, anch'esso agganciato a un nome di dominio management.*westeurope.cloudapp.azure.com*, per rendere la *VM* raggiungibile via internet, sia per installarvi il mail server del progetto che per le operazioni di management di tutta l'infrastruttura.
Terminata la realizzazione dell'infrastruttura è possibile passare all'installazione del portale *HSIT*.

Tempistiche
La creazione delle 3x *VM* ha richiesto fino a 4 minuti, mentre il loro aggiornamento e configurazione al primo avvio ha richiesto, in tutto, circa 15 minuti, in quanto la creazione di una nuova *VM* non *bloccava* le operazioni di aggiornamento/configurazione di quella precedentemente creata.

2.15 Riconfigurazione DNS hsit.it

Come accennato in *Creazione di una macchina virtuale* e in *Creazione server HSIT*, adesso abbiamo due nomi di dominio che puntano, rispettivamente, all'*AG* e alla *VM* che useremo come *mail server* e manutenzione dell'intera infrastruttura.
Quindi, l'ultimo step per ultimare la configurazione, è la riconfigurazione della zona di dominio interessata.
Tutti i record necessari, sia quelli indicati nel certificato che quelli di management, verranno configurati per essere di tipo *CNAME*[33] del nome di dominio impostato su Azure, quindi per il dominio hsit.it avremo:

- www.hsit.it
- e.hsit.it
- c.hsit.it

che dovranno puntare a hsit.*westeurope.cloudapp.azure.com*

- management.hsit.it

che dovrà puntare a management.*westeurope.cloudapp.azure.com*.

```
diego@bigbang:~$ dig hsit.westeurope.cloudapp.azure.com

; <<>> DiG 9.10.3-P4-Debian <<>> hsit.westeurope.cloudapp.azure.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31119
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;hsit.westeurope.cloudapp.azure.com. IN A

;; ANSWER SECTION:
hsit.westeurope.cloudapp.azure.com. 10 IN A     52.178.119.254

;; AUTHORITY SECTION:
westeurope.cloudapp.azure.com. 165 IN   NS      ns1-201.azure-dns.com.
westeurope.cloudapp.azure.com. 165 IN   NS      ns1prod.6619.azuredns-prd.org.
westeurope.cloudapp.azure.com. 165 IN   NS      ns2prod.6619.azuredns-prd.org.
westeurope.cloudapp.azure.com. 165 IN   NS      ns2prod.6619.azuredns-prd.info.

;; ADDITIONAL SECTION:
ns1-201.azure-dns.com.  10166   IN      A       40.90.4.201
ns1prod.6619.azuredns-prd.org. 318 IN   A       40.90.4.201
ns2prod.6619.azuredns-prd.org. 318 IN   A       64.4.48.201
ns2prod.6619.azuredns-prd.info. 318 IN  A       13.107.160.201

;; Query time: 37 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sat May 09 15:00:25 CEST 2020
;; MSG SIZE  rcvd: 284
```

Figura 15 Risoluzione del nome di dominio hsit.westeurope.cloudapp.azure.com

Figure 15 *hsit.westeurope.cloudapp.azure.com name resolution.*

Analogamente mentre per il dominio *haisentitoilterremoto.it* avremo tutti i *CNAME* che punteranno al rispettivo *AG* con nome di dominio *haisentitoilterremoto.it*:

- www.haisentitoilterremoto.it
- eventi.haisentitoilterremoto.it
- comuni.haisentitoilterremoto.it

[33] Un CNAME, o Canonical Name record, è un tipo di record presente nel DNS che permette agli utenti di specificare un alias per un nome a dominio. Ad esempio, è possibile creare un alias di domain1.com utilizzando domain2.com.

che dovranno puntare a haisentitoilterremoto.*westeurope.cloudapp.azure.com*

- management.haisentitoilterremoto.it

che dovrà puntare a management.*westeurope.cloudapp.azure.com*.

Dopo aver aggiornato i dati della zona e aggiornato il server DNS in uso, la nuova infrastruttura è finalmente raggiungibile e pubblicamente usabile.

```
; Created by: diego.sorrentino@ingv.it on date: 05-05-2020 15:34.41
$TTL 604800     ; 1 week
@               IN SOA  dns1.ingv.it. csi.ingv.it. (
                1588685681; serial
                86400; refresh
                3600; retry
                604800; expire
                86400; minimum
                )

; NS section
@                               IN      NS      dns1.ingv.it.
@                               IN      NS      dns2.ingv.it.

; NS-GlueRecord section
dns1.ingv.it.                   IN      A       93.63.40.4
dns2.ingv.it.                   IN      A       93.63.40.2

; TXT section
@                               IN      TXT     "HSIT.IT"

; OutSourcing CNAME section
c                               IN      CNAME   hsit.westeurope.cloudapp.azure.com.
e                               IN      CNAME   hsit.westeurope.cloudapp.azure.com.
management                      IN      CNAME   management.westeurope.cloudapp.azure.com.
www                             IN      CNAME   hsit.westeurope.cloudapp.azure.com.
```

Figura 16 Configurazione della zona DNS di hsit.it.

Figure 16 *hsit.it DNS zone configuration.*

3. Controlli di funzionamento

Per controllare che l'infrastruttura stia funzionando correttamente basta eseguire dei semplici controlli, su ogni passaggio necessario per ricevere una risorsa.

3.1 Controllo di redirezione da HTTP a HTTPS

Tramite il comando wget possiamo richiedere da terminale una risorsa web.
Proviamo richiedendo la pagina iniziale del sito http://www.hsit.it con il comando:

```
$ wget -O - /dev/null http://www.hsit.it/
```

Il comando scarica e *scarta* la risorsa, inviandola *a /dev/null*, ma dalla figura 17 possiamo vedere che l'*AG* inoltra il traffico da verso il protocollo sicuro, restituendo il codice **301 Moved Permanently** così da indicare a browser e motori di ricerca che la risorsa richiesta non è più disponibile al vecchio indirizzo e indica il nuovo.
Intrinsecamente abbiamo controllato anche il funzionamento del download della richiesta su protocollo cifrato in quanto la risorsa richiesta è stata trovata e scaricata.

```
diego@bigbang:~$ wget -O /dev/null http://www.hsit.it/
--2020-05-09 18:49:52--  http://www.hsit.it/
Resolving www.hsit.it (www.hsit.it)... 52.178.119.254
Connecting to www.hsit.it (www.hsit.it)|52.178.119.254|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://www.hsit.it/ [following]
--2020-05-09 18:49:53--  https://www.hsit.it/
Connecting to www.hsit.it (www.hsit.it)|52.178.119.254|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 80 [text/html]
Saving to: `/dev/null'

/dev/null           100%[===================>]      80  --.-KB/s    in 0s

2020-05-09 18:49:53 (38.2 MB/s) - `/dev/null' saved [80/80]

diego@bigbang:~$
```

Figura 17 Redirezione da protocollo HTTP a HTTPS.

Figure 17 *Redirect from protocol HTTP to HTTPS.*

3.2 Controllo del certificato

Per controllare velocemente la validità del certificato basta consultare, tramite browser, una qualsiasi pagina web del sito e vedere se compare l'icona di un lucchetto vicino all'indirizzo richiesto, come in figura 18.
In caso di errore nel certificato o nella pagina, non completamente sicura, si otterrebbe una segnalazione di sito *Non sicuro*, come in figura 19.
Analizzando la pagina si può notare che il certificato è valido ma la pagina contiene un mix di riferimenti a risorse HTTP e HTTPS. Nel caso particolare l'errore è dovuto alla non completa migrazione del sito.

Figura 18 Controllo validità certificato.

Figure 18 *Certificate validity check.*

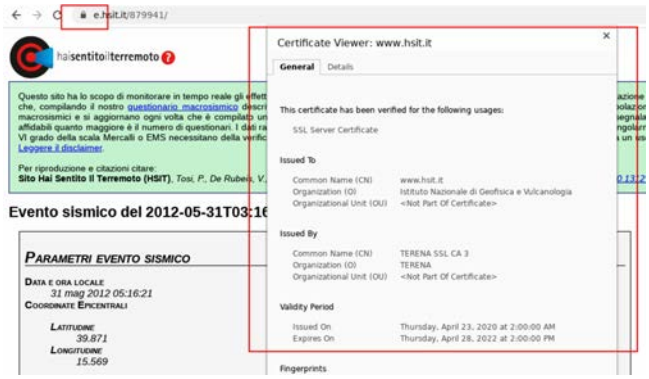
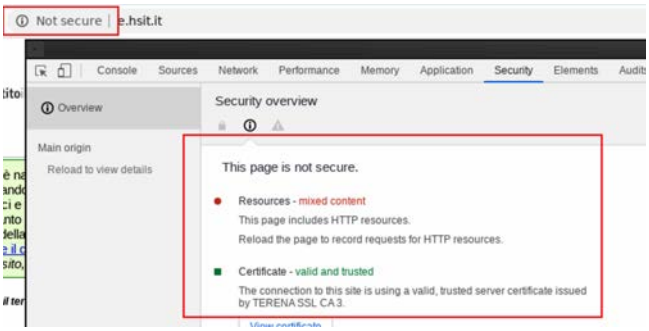


Figura 19 Segnalazione di errore.

Figure 19 Alert error.



3.3 Utilizzo di HTTP/2

Per controllare se si sta utilizzando HTTP/2 abbiamo utilizzato come browser *Google Chrome*. Attivando la funzione *Strumenti di sviluppo*, selezionare il *tab Network* e ricaricare la pagina. Si otterrà un grafico simile a quello in figura 20.

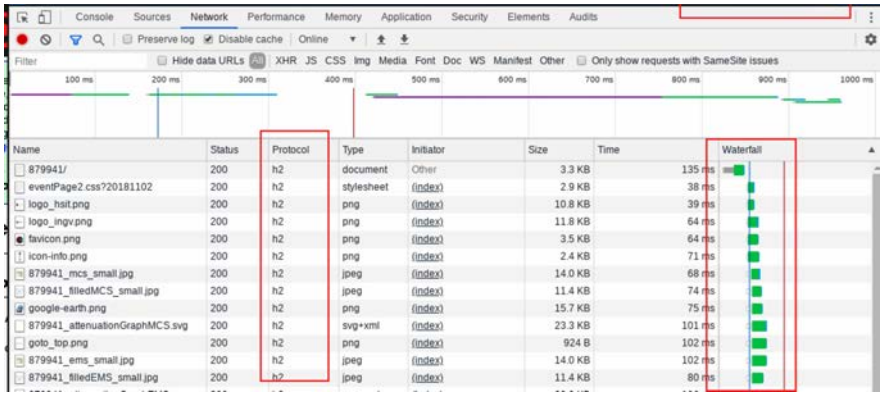


Figura 20 Utilizzo di HTTP/2.

Figure 20 HTTP/2 connection.

Il grafico in alto mostra il numero di chiamate effettuate e i tempi di download delle risorse. Nella tabella sottostante le due colonne di interesse sono la *Protocol* che indica il protocollo in uso e la *Waterfall*, che indica come vengono *spedite* le risorse al client, nel caso *tutte insieme* riducendo il numero di richieste che il browser effettua al server e, di conseguenza, i tempi di attesa.
Per poter apprezzare il miglioramento si riporta in figura 21 la stessa pagina richiesta all'infrastruttura sita nella *webfarm* della sede Romana.

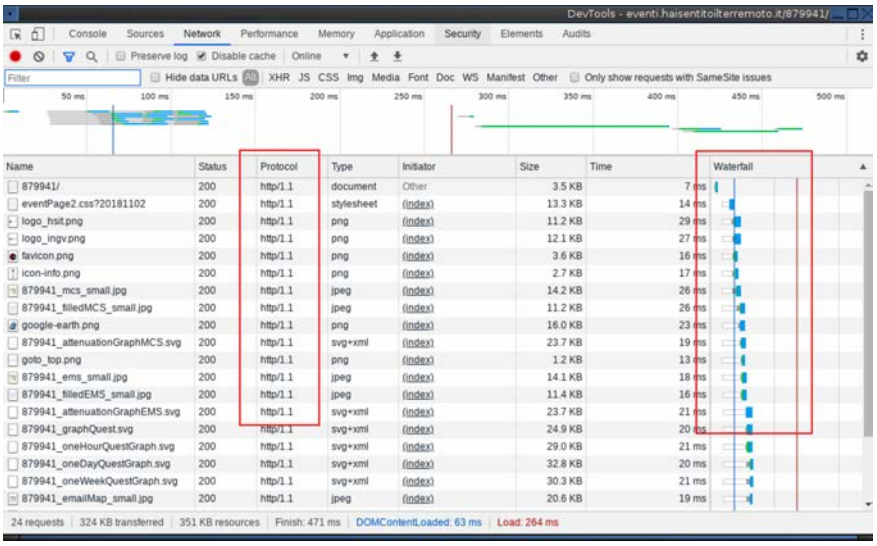


Figura 21 Utilizzo di http/1.1.

Figure 21 http/1.1 connection.

Il grafico mostra la quantità di richieste effettuate al web server mentre in tabella, nella colonna *Waterfall* troviamo le fasi di download delle varie risorse (richiesta, inizio e fine download).
I tempi sembrano essere simili ma va ricordato che nel caso HTTP/2 tutte le informazioni devono essere preventivamente cifrate.

4. Riuso degli script di realizzazione dell'infrastruttura

Per velocizzare le operazioni di creazione e distruzione dell'infrastruttura durante la prima fase di apprendimento e per poter supportare altri colleghi nella realizzazione di una infrastruttura simile, sono stati realizzati una serie di script ad-hoc per ogni passaggio e un file di configurazione generale, riportati di seguito.

4.1 File 00_settings.sh

Nota. Questo file contiene tutte le variabili usati negli script successivi che, quindi, lo *includono* prima di far partire i comandi Azure (direttiva "**. ./00_settings.sh**").

```
#!/bin/sh

# GroupName
GROUP_NAME=HSIT

# DNS Name (ip may change...)  it will resolved in
${DNS_NAME}.westeurope.cloudapp.azure.com.
DNS_NAME=hsit
MANAGEMENT_DNS_NAME=management
# Virtual devices tags
TAG_PROJECT=hsit
TAG_SERVER=server
TAG_AG=AppGateway
TAG_SCALABILITY=ScaleSet

# User
USER_NAME=diego.sorrentino
USER_SSH_KEY=azure-rsa.key.pub
USER_SSH_PRIVATE_KEY=azure-rsa.key

# Network and subnetting
VNET_NAME=HSITVNet
SUBNET_NAME_SERVER=HSITServersSubnet
SUBNET_NAME_AG=HSITAgSubnet

# Application Gateway
AG_PUBLIC_IP_NAME=HSITAgPublicIP
AG_NAME=HSITAg
AG_BACKEND_POOL_DEFAULT_NAME=appGatewayBackendPool
AG_FRONTEND_IP_DEFAULT_NAME=appGatewayFrontendIP
AG_HTTP_LISTENER_DEFAULT_NAME=appGatewayHttpListener
```

```
# Cert file
CERT_HSIT_CSR=www_hsit_it.csr
CERT_HSIT_CRT=www_hsit_it.crt
CERT_HSIT_KEY=www_hsit_it.key
CERT_HAISENTITOILTERREMOTO_CSR=www_haisentitoilterremoto_it.csr
CERT_HAISENTITOILTERREMOTO_CRT=www_haisentitoilterremoto_it.crt
CERT_HAISENTITOILTERREMOTO_KEY=www_haisentitoilterremoto_it.key

PFX_HSIT_FILE=cert-hsit-azure.pfx
PFX_HAISENTITOILTERREMOTO_FILE=cert-haisentitoilterremoto-azure.pfx
PFX_PWD=TUA_PASSWORD

# VM base to generalize
VM_TO_GENERALIZE_NAME=TemplateScalingVM
VM_TO_GENERALIZE_NIC=TemplateScalingVMNic
VM_TO_GENERALIZE_NSG=TemplateScalingVMNSG
VM_TO_GENERALIZE_PUBLIC_IP=TemplateScalingVM-public-ip
VM_TO_GENERALIZE_YML=init_image_to_generalize.yml
VM_TO_GENERALIZE_SIZE=Standard_B2s
VM_TO_GENERALIZE_IMAGE=UbuntuLTS
VM_TO_GENERALIZE_IP=”10.0.1.10”
VM_TO_GENERALIZE_PUBLIC_IP_NAME=TemplateScalingVM-public-ip

# Scale set
SCALESET_NAME=HSITScaleSet
SCALESET_IMAGE=HSIT_ReverseProxyTPL
SCALESET_VM_SKU=Standard_DS2
# Redirect Http to Https rules
PORT80_NAME=port_80
LISTENER_NAME=ListenerHttpPort
REDIRECT_NAME=httpToHttps

### Servers
# DB
SERVER_DB_NAME=hsit-db
SERVER_DB_SIZE=Standard_D2_v3
SERVER_DB_IMAGE=UbuntuLTS
SERVER_DB_IP=”10.0.1.20”
SERVER_DB_YML=init_hsit_db.yml

# Map
SERVER_MAP_NAME=hsit-map
SERVER_MAP_SIZE=Standard_D2_v2
SERVER_MAP_IMAGE=UbuntuLTS
SERVER_MAP_IP=”10.0.1.21”
SERVER_MAP_YML=init_hsit_map.yml
SERVER_MAP_IP_NAME=hsit-map-ip

# Web
SERVER_WEB_NAME=hsit-web
```

```
SERVER_WEB_SIZE=Standard_B4ms
SERVER_WEB_IMAGE=UbuntuLTS
SERVER_WEB_IP=”10.0.1.22”
SERVER_WEB_YML=init_hsit_web.yml
```

4.2 File 01_create_net.sh

```
#!/bin/sh
. ./00_settings.sh

# net 10.0.0.0/22 => 10.0.0.0 / 255.255.252.0
# ip range 10.0.0.0 -> 10.0.3.254

az network vnet create \
        —name ${VNET_NAME} \
        —resource-group ${GROUP_NAME} \
        —address-prefix 10.0.0.0/22 \
        —subnet-name ${SUBNET_NAME_AG} \
        —subnet-prefix 10.0.0.0/24

az network vnet subnet create \
        —name ${SUBNET_NAME_SERVER} \
        —resource-group ${GROUP_NAME} \
        —vnet-name ${VNET_NAME} \
        —address-prefix 10.0.1.0/24
az network public-ip create \
        —resource-group ${GROUP_NAME} \
        —name ${AG_PUBLIC_IP_NAME} \
        —dns-name ${DNS_NAME}
```

4.3 File 02_convert_certificate.sh

Nota. La prima parte è la creazione del certificato da inviare al certificatore. Nello script è commentata in quanto si assume che sia già stato creato e restituito firmato al richiedente.

```
#!/bin/sh
. ./00_settings.sh

## Create hsit.it certificate
#openssl req \
#       -new \
#       -newkey rsa:2048 \
#       -nodes \
#       -out ${CERT_HSIT_CSR} \
#       -keyout ${CERT_HSIT_KEY} \
#      -subj “/C=IT/ST=RM/L=Roma/O=Istituto Nazionale di Geofisica e Vulcanologia/OU=Hai
Sentito il Terremoto?/CN=www.hsit.it”
```

```
## Create haisentitoilterremoto.it certificate
#openssl req \
#       -new \
#       -newkey rsa:2048 \
#       -nodes \

#       -out ${CERT_HAISENTITOILTERREMOTO_CSR} \
#       -keyout ${CERT_HAISENTITOILTERREMOTO_KEY} \
#      -subj "/C=IT/ST=RM/L=Roma/O=Istituto Nazionale di Geofisica e Vulcanologia/OU=Hai
Sentito il Terremoto?/CN=www.haisentitoilterremoto.it"

# Convert certificate HSIT
openssl pkcs12 \
        -export \
        -out ${PFX_HSIT_FILE} \
        -inkey ${CERT_HSIT_KEY} \
        -in ${CERT_HSIT_CRT}

# Convert certificate HSIT
openssl pkcs12 \
        -export \
        -out ${PFX_HAISENTITOILTERREMOTO_FILE} \
        -inkey ${CERT_HAISENTITOILTERREMOTO_KEY} \
        -in ${CERT_HAISENTITOILTERREMOTO_CRT}
```

4.4 File 03_create_ssh_keys.sh

```
#!/bin/sh
. ./00_settings.sh

# Create a dedicated SSH key-pair
ssh-keygen -m PEM -t rsa -b 4096 -f azure-rsa.key
```

4.5 File 04_app_gateway.sh

```
#!/bin/sh
. ./00_settings.sh

az network application-gateway create \
        —name ${AG_NAME} \
        —resource-group ${GROUP_NAME} \
        —capacity 2 \
        —cert-file ${PFX_HSIT_FILE} \
        —cert-password ${PFX_PWD} \
        —frontend-port 443 \
        —http-settings-cookie-based-affinity Disabled \
        —http-settings-port 80 \
        —http-settings-protocol Http \
```

```
    —http2 Enabled \
    —public-ip-address ${AG_PUBLIC_IP_NAME} \
    —sku Standard_Medium \
    —subnet ${SUBNET_NAME_AG} \
    —vnet-name ${VNET_NAME} \
    —tags Project=${TAG_PROJECT} Type=${TAG_AG} Job=sslbalancer
```

4.6 File 05_image_for_scale-set.sh

```
#!/bin/sh
. ./00_settings.sh

# Create image
az vm create \
    —verbose \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_NAME} \
    —size ${VM_TO_GENERALIZE_SIZE} \
    —image ${VM_TO_GENERALIZE_IMAGE} \
    —vnet-name ${VNET_NAME} \
    —subnet ${SUBNET_NAME_SERVER} \
    —private-ip-address ${VM_TO_GENERALIZE_IP} \
    —public-ip-address ${VM_TO_GENERALIZE_PUBLIC_IP_NAME} \
    —nsg ${VM_TO_GENERALIZE_NSG} \
    —custom-data ./${VM_TO_GENERALIZE_YML} \
    —admin-username ${USER_NAME} \
    —ssh-key-value ${USER_SSH_KEY}

VM_PUBLIC_IP=`az network public-ip show \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_PUBLIC_IP_NAME} \
    —query [ipAddress] \
    —output tsv`

echo “Now, from Azure console login in this VM with the command:”
echo “$ ssh -i ${USER_SSH_PRIVATE_KEY} ${USER_NAME}@${VM_PUBLIC_IP}”
echo “and check if the configuration is right OR configure the VM”
echo “when everything is working, deprovisionate the VM:”
echo “$ sudo waagent -deprovision+user -force && export HISTSIZE=0 && sync && sudo
shutdown -h now”
```

4.7 File 06_generalize_image.sh

```
#!/bin/sh
. ./00_settings.sh

# Deallocate
az vm deallocate \
```

```
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_NAME}

# Generalize
az vm generalize \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_NAME}


# Make image
az image create \
    —resource-group ${GROUP_NAME} \
    —name ${SCALESET_IMAGE} \
    —source ${VM_TO_GENERALIZE_NAME}

#delete temporary vm
DiskName=`az vm show \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_NAME} \
    —query storageProfile.osDisk.name \
    —output tsv`

az vm delete \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_NAME} \
    —yes
az network nic delete \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_NIC}

az network public-ip delete \
    —resource-group ${GROUP_NAME} \
    —name ${VM_TO_GENERALIZE_PUBLIC_IP}

az disk delete \
    —resource-group ${GROUP_NAME} \
    —name ${DiskName}

az network nsg delete \
    —resource-group ${GROUP_NAME} \
    —name ${TemplateScalingVMNSG}
```

4.8 File 07_scale_set.sh

```
#!/bin/sh
. ./00_settings.sh

az vmss create \
  —name ${SCALESET_NAME} \
```

```
  --resource-group ${GROUP_NAME} \
  --image ${SCALESET_IMAGE} \
  --admin-username ${USER_NAME} \
  --ssh-key-value @${USER_SSH_KEY} \
  --instance-count 2 \
  --vnet-name ${VNET_NAME} \
  --subnet ${SUBNET_NAME_SERVER} \
  --vm-sku ${SCALESET_VM_SKU} \

  --upgrade-policy-mode Automatic \
  --app-gateway ${AG_NAME} \
  --backend-pool-name ${AG_BACKEND_POOL_DEFAULT_NAME} \
  --tags Project=${TAG_PROJECT} Type=${TAG_SCALABILITY} Job=nginxcache
```

4.9 File 08_redirect_http_to_https.sh

```
#!/bin/sh
. ./00_settings.sh

# “Create” a named port
az network application-gateway frontend-port create \
        --port 80 \
        --gateway-name ${AG_NAME} \
        --resource-group ${GROUP_NAME} \
        --name ${PORT80_NAME}
# Create a listener on the named port (port_80)
az network application-gateway http-listener create \
        --name ${LISTENER_NAME} \
        --frontend-ip ${AG_FRONTEND_IP_DEFAULT_NAME} \
        --frontend-port ${PORT80_NAME} \
        --gateway-name ${AG_NAME} \
        --resource-group ${GROUP_NAME}

# Redirect traffic
az network application-gateway redirect-config create \
        --name ${REDIRECT_NAME} \
        --gateway-name ${AG_NAME} \
        --resource-group ${GROUP_NAME} \
        --type Permanent \
        --target-listener ${AG_HTTP_LISTENER_DEFAULT_NAME} \
        --include-path true \
        --include-query-string true

az network application-gateway rule create \
        --gateway-name ${AG_NAME} \
        --name rule2 \
        --resource-group ${GROUP_NAME} \
        --http-listener ${LISTENER_NAME} \
        --rule-type Basic \
```

```
    —redirect-config ${REDIRECT_NAME}
```

4.10 File 09_create-servers.sh

```
#!/bin/sh –x
. ./00_settings.sh

# Create DB server
az vm create \
    —verbose \
    —resource-group ${GROUP_NAME}\
    —name ${SERVER_DB_NAME} \
    —size ${SERVER_DB_SIZE} \
    —image ${SERVER_DB_IMAGE} \
    —vnet-name ${VNET_NAME} \
    —subnet ${SUBNET_NAME_SERVER} \
    —private-ip-address ${SERVER_DB_IP} \
    —public-ip-address “” \
    —custom-data ./${SERVER_DB_YML} \
    —admin-username ${USER_NAME} \
    —ssh-key-value @${USER_SSH_KEY} \
    —tags Project=${TAG_PROJECT} Type=${TAG_SERVER} Job=dbms

# Create Map server + 1public ip for MAIL SERVER and management
az vm create \
    —verbose \
    —resource-group ${GROUP_NAME} \
    —name ${SERVER_MAP_NAME} \
    —size ${SERVER_MAP_SIZE} \
    —image ${SERVER_MAP_IMAGE} \
    —vnet-name ${VNET_NAME} \
    —subnet ${SUBNET_NAME_SERVER} \
    —private-ip-address ${SERVER_MAP_IP} \
    —custom-data ./${SERVER_MAP_YML} \
    —admin-username ${USER_NAME} \
    —ssh-key-value @${USER_SSH_KEY} \
    —tags Project=${TAG_PROJECT} Type=${TAG_SERVER} Job=map \
    —public-ip-address ${SERVER_MAP_IP_NAME} \
    —public-ip-address-dns-name ${MANAGEMENT_DNS_NAME}

# Create Web server + one storage disk of 1024G (1Tb)
az vm create \
    —verbose \
    —resource-group ${GROUP_NAME} \
    —name ${SERVER_WEB_NAME} \
    —size ${SERVER_WEB_SIZE} \
    —image ${SERVER_WEB_IMAGE} \
    —vnet-name ${VNET_NAME} \
    —subnet ${SUBNET_NAME_SERVER} \
```

```
—private-ip-address ${SERVER_WEB_IP} \
—public-ip-address “” \
—custom-data ./${SERVER_WEB_YML} \
—admin-username ${USER_NAME} \
—ssh-key-value @${USER_SSH_KEY} \
—tags Project=${TAG_PROJECT} Type=${TAG_SERVER} Job=web \
—data-disk-sizes-gb 1024
```

5. Conclusioni

Inizialmente sarà necessario monitorare il funzionamento del sistema, effettuare numerosi test di carico per controllare il funzionamento dell'*Application Gateway* e del *Set di Scalabilità*, in quanto si dovranno decidere i parametri ottimali per la configurazione del servizio.
In una fase successiva, per avere un'ulteriore affidabilità, si prenderà in considerazione la possibilità di sfruttare al massimo le potenzialità dei servizi *Batch* di Azure, per l'aggiornamento degli elaborati e della sincronizzazione del sistema, e dei servizi *SaaS* su cui migrare il server DBMS.

Bibliografia

Aivaliotis D., (2016). *Mastering NGINX - Second Edition*. Packt publishing.

Albitz P., Liu C., (2006). *DNS and BIND (5th Edition)*. O'Reilly.

Apache Software Foundation, (2016). *Apache HTTP Server Documentation Version 2.5*. 12th Media Services.

Barrett D.J., Silverman R.E., Byrnes R. G., (2005). *SSH, The Secure Shell: The Definitive Guide*. O'Reilly.

Lindsey M., (2020). *Azure:Microsoft Azure: Build, manage, and scale cloud applications using the Azure Infrastructure*. Independently published.

Newham C., (2005). *Learning the bash Shell: Unix Shell Programming*. O'Reilly.

Sorrentino D., Sbarra P., De Rubeis V., Tosi P., (2010). *Realizzazione ed Evoluzione della versione 1.0 del Questionario Macrosismico online dell'INGV*, Rapporti Tecnici INGV.

Viega J., Messier M., Chandra P., (2002). *Network Security with OpenSSL: Cryptography for Secure Communications*. O'Reilly.

QUADERNI di GEOFISICA

ISSN 1590-2595 http://istituto.ingv.it/it/le-collane-editoriali-ingv/quaderni-di-geofisica.html/

I QUADERNI DI GEOFISICA (QUAD. GEOFIS.) accolgono lavori, sia in italiano che in inglese, che diano particolare risalto alla pubblicazione di dati, misure, osservazioni e loro elaborazioni anche preliminari che necessitano di rapida diffusione nella comunità scientifica nazionale ed internazionale. Per questo scopo la pubblicazione on-line è particolarmente utile e fornisce accesso immediato a tutti i possibili utenti. Un Editorial Board multidisciplinare ed un accurato processo di peer-review garantiscono i requisiti di qualità per la pubblicazione dei contributi. I QUADERNI DI GEOFISICA sono presenti in "Emerging Sources Citation Index" di Clarivate Analytics, e in "Open Access Journals" di Scopus.

QUADERNI DI GEOFISICA (QUAD. GEOFIS.) welcome contributions, in Italian and/or in English, with special emphasis on preliminary elaborations of data, measures, and observations that need rapid and widespread diffusion in the scientific community. The on-line publication is particularly useful for this purpose, and a multidisciplinary Editorial Board with an accurate peer-review process provides the quality standard for the publication of the manuscripts. QUADERNI DI GEOFISICA are present in "Emerging Sources Citation Index" of Clarivate Analytics, and in "Open Access Journals" of Scopus.

RAPPORTI TECNICI INGV

ISSN 2039-7941 http://istituto.ingv.it/it/le-collane-editoriali-ingv/rapporti-tecnici-ingv.html/

I RAPPORTI TECNICI INGV (RAPP. TEC. INGV) pubblicano contributi, sia in italiano che in inglese, di tipo tecnologico come manuali, software, applicazioni ed innovazioni di strumentazioni, tecniche di raccolta dati di rilevante interesse tecnico-scientifico. I RAPPORTI TECNICI INGV sono pubblicati esclusivamente on-line per garantire agli autori rapidità di diffusione e agli utenti accesso immediato ai dati pubblicati. Un Editorial Board multidisciplinare ed un accurato processo di peer-review garantiscono i requisiti di qualità per la pubblicazione dei contributi.

RAPPORTI TECNICI INGV (RAPP. TEC. INGV) publish technological contributions (in Italian and/or in English) such as manuals, software, applications and implementations of instruments, and techniques of data collection. RAPPORTI TECNICI INGV are published online to guarantee celerity of diffusion and a prompt access to published data. A multidisciplinary Editorial Board and an accurate peer-review process provide the quality standard for the publication of the contributions.

MISCELLANEA INGV

ISSN 2039-6651 http://istituto.ingv.it/it/le-collane-editoriali-ingv/miscellanea-ingv.html

MISCELLANEA INGV (MISC. INGV) favorisce la pubblicazione di contributi scientifici riguardanti le attività svolte dall'INGV. In particolare, MISCELLANEA INGV raccoglie reports di progetti scientifici, proceedings di convegni, manuali, monografie di rilevante interesse, raccolte di articoli, ecc. La pubblicazione è esclusivamente on-line, completamente gratuita e garantisce tempi rapidi e grande diffusione sul web. L'Editorial Board INGV, grazie al suo carattere multidisciplinare, assicura i requisiti di qualità per la pubblicazione dei contributi sottomessi.

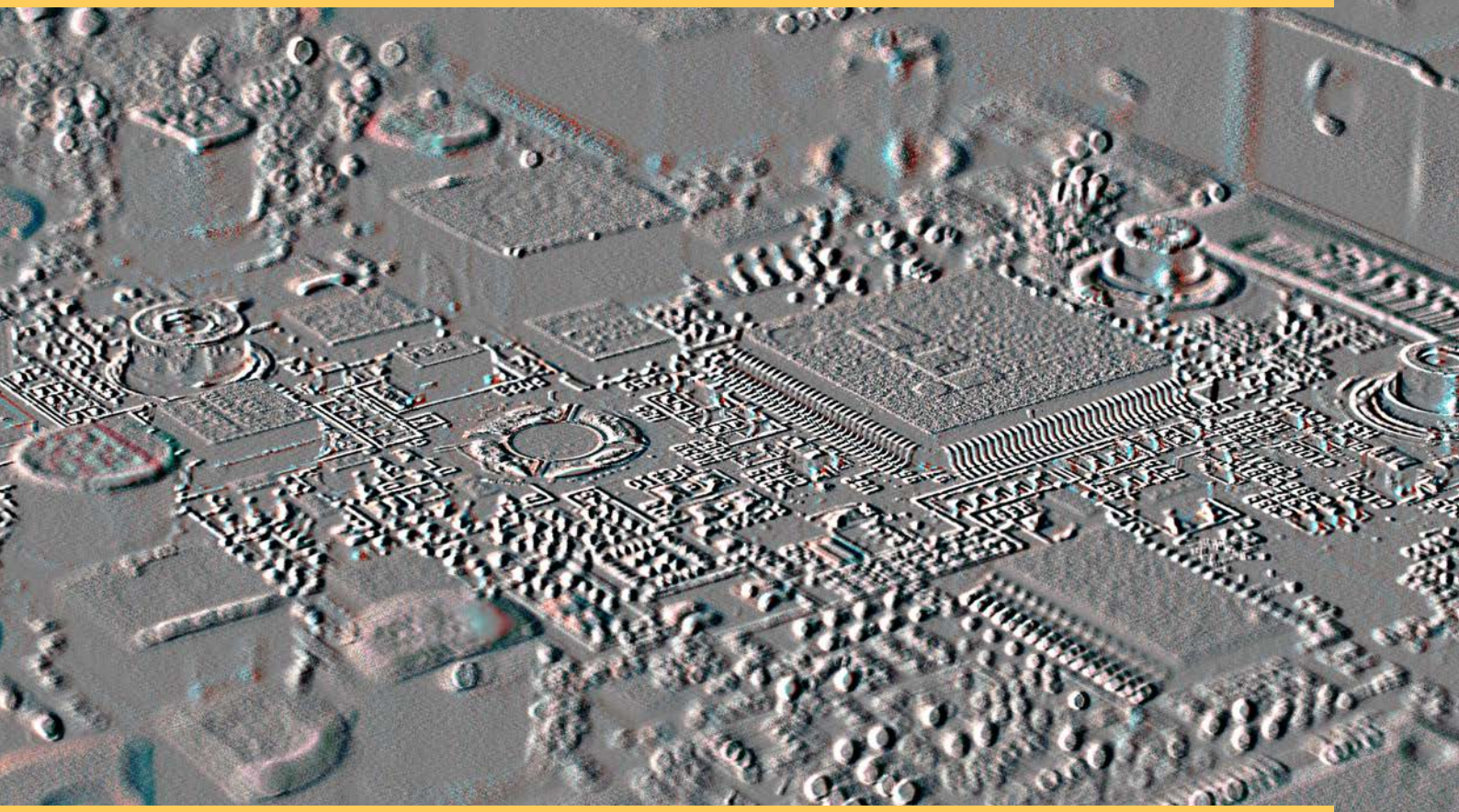
MISCELLANEA INGV (MISC. INGV) favours the publication of scientific contributions regarding the main activities carried out at INGV. In particular, MISCELLANEA INGV gathers reports of scientific projects, proceedings of meetings, manuals, relevant monographs, collections of articles etc. The journal is published online to guarantee celerity of diffusion on the internet. A multidisciplinary Editorial Board and an accurate peer-review process provide the quality standard for the publication of the contributions.

Coordinamento editoriale e impaginazione
Francesca DI STEFANO, Rossella CELI
Istituto Nazionale di Geofisica e Vulcanologia

Progetto grafico e impaginazione
Barbara ANGIONI
Istituto Nazionale di Geofisica e Vulcanologia

Istituto Nazionale di Geofisica e Vulcanologia
Via di Vigna Murata, 605
00143 Roma
tel. +39 06518601

www.ingv.it



ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA