



ISTI Technical Reports

La caratterizzazione dei test funzionali per la valutazione delle performance delle funzionalità di condivisione informazione del "nodo casa"

Vittorio Miori, CNR-ISTI, Pisa, Italy

Dario Russo, CNR-ISTI, Pisa, Italy



La caratterizzazione dei test funzionali per la valutazione delle performance delle funzionalità di condivisione informazione del "nodo casa"

Miori V.; Russo D.
ISTI-TR-2020/024

Abstract

Il lavoro riguarda la caratterizzazione dei test funzionali per la valutazione delle performance delle funzionalità di condivisione informazione del "nodo casa". Il lavoro si è tradotto nell'individuazione sia del contesto nel quale andavano eseguiti tali test funzionali, sia degli strumenti idonei per valutare le procedure di condivisione relative alle prestazioni del software in termini di qualità, affidabilità e robustezza. A tal fine, il documento è organizzato come segue: presentazione dell'ambiente all'interno del quale andranno valutate le performance in questione e cioè l'infrastruttura Cloud Nuvola IT Self Data Center di Telecom Italia; analisi critica delle prestazioni del Cloud; analisi degli elementi e dei test da considerare in merito alla verifica della qualità del software; descrizione dei requisiti e delle funzionalità appropriati per lo specifico contesto, degli applicativi dedicati al monitoraggio; presentazione di due strumenti software di monitoraggio, individuati come possibile supporto alla successiva fase di vera e propria esecuzione sul campo dei test funzionali e verifiche mediante emulazione di interazione della casa con strutture gerarchicamente superiori.

Valutazione performance, Monitoraggio, Test funzionali, Nodo casa, Prestazioni software, Cloud computing, Qualità del software

Citation

Miori V.; Russo D. *La caratterizzazione dei test funzionali per la valutazione delle performance delle funzionalità di condivisione informazione del "nodo casa"* ISTI Technical Reports 2020/024. DOI: 10.32079/ISTI-TR-2020/024

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"

Area della Ricerca CNR di Pisa

Via G. Moruzzi 1

56124 Pisa Italy

<http://www.isti.cnr.it>

**La caratterizzazione dei test funzionali per la valutazione
delle performance delle funzionalità di condivisione
informazione del "nodo casa"**

Vittorio Miori (CNR) – Dario Russo (CNR)

Breve sommario

Il lavoro riguarda la caratterizzazione dei test funzionali per la valutazione delle performance delle funzionalità di condivisione informazione del "nodo casa". Il lavoro si è tradotto nell'individuazione sia del contesto nel quale andavano eseguiti tali test funzionali, sia degli strumenti idonei per valutare le procedure di condivisione relative alle prestazioni del software in termini di qualità, affidabilità e robustezza. A tal fine, il documento è organizzato come segue: presentazione dell'ambiente all'interno del quale andranno valutate le performance in questione e cioè l'infrastruttura Cloud Nuvola IT Self Data Center di Telecom Italia; analisi critica delle prestazioni del Cloud; analisi degli elementi e dei test da considerare in merito alla verifica della qualità del software; descrizione dei requisiti e delle funzionalità appropriati per lo specifico contesto, degli applicativi dedicati al monitoraggio; presentazione di due strumenti software di monitoraggio, individuati come possibile supporto alla successiva fase di vera e propria esecuzione sul campo dei test funzionali e verifiche mediante emulazione di interazione della casa con strutture gerarchicamente superiori.

Parole chiave

Valutazione performance, Monitoraggio, Test funzionali, Nodo casa, Prestazioni software, Cloud computing, Qualità del software

Indice

BREVE SOMMARIO	2
PAROLE CHIAVE	2
INDICE DELLE FIGURE	5
1. INTRODUZIONE	6
2. DEFINIZIONE DELL'INFRASTRUTTURA CLOUD UTILIZZATA PER L'IMPLEMENTAZIONE DELLA PIATTAFORMA "FRAMEWORK SHELL"	6
2.1 DESCRIZIONE DELLA PIATTAFORMA NUVOLA IT SELF DATA CENTER TELECOM ITALIA PER SHELL	6
2.1.1 <i>Accesso al portale di gestione del servizio</i>	12
2.1.2 <i>Rete di accesso e Indirizzi IP pubblici</i>	13
2.1.3 <i>Catalogo Cloud e vApp</i>	13
2.2 IL SERVIZIO DI BACKUP.....	14
2.3 PORTALE DI SELFRESTORE	14
2.4 IL SERVIZIO DI ANTIVIRUS	15
2.5 PROFILI DEGLI UTILIZZATORI DELLA PIATTAFORMA.....	16
2.5.1 <i>Strumenti a disposizione dell'Amministratore (Cliente)</i>	17
2.5.2 <i>Elementi del portale di gestione</i>	19
2.6 REPORTING	19
2.6.1 <i>Reporting TIM Self Data Center</i>	20
2.7 SICUREZZA.....	20
2.8 ASSISTENZA TECNICA E SLA DI DISPONIBILITÀ DEL SERVIZIO	21
2.8.1 <i>Dati di targa</i>	21
2.9 DIMENSIONAMENTO DELLA PIATTAFORMA CLOUD NUVOLA IT SELF DATA CENTER PER SHELL.....	21
2.10 DESCRIZIONE DELLE POLITICHE DI COSTO DELLA PIATTAFORMA CLOUD NUVOLA IT SELF DATA CENTER TELECOM ITALIA.....	23
2.10.1 <i>Pay-as-you-go</i>	23
2.10.2 <i>Allocation pool</i>	23
3. PRESTAZIONI	25
3.1 PRESTAZIONI TRAMITE UNA FASE DI MONITORAGGIO	25
3.1.1 <i>QoS per il cloud</i>	27
3.1.2 <i>Scalabilità</i>	28
4. LA QUALITÀ DEL SOFTWARE	29
4.1 CONTROLLI INTERNI ED ESTERNI	29
4.2 VERIFICA E VALIDAZIONE	30
4.2.1 <i>Controlli statici</i>	32
4.2.2 <i>Controlli dinamici</i>	32
4.2.3 <i>Progettazione dei test</i>	32
4.3 LIVELLI DI TEST	33
4.3.1 <i>Test sul sistema</i>	33
5. IL MONITORAGGIO DELLE PRESTAZIONI DELL'APPLICAZIONE PER LA VALUTAZIONE DELLE PERFORMANCE DELLE FUNZIONALITÀ DI CONDIVISIONE DELL'INFORMAZIONE	36
5.1 REQUISITI PRINCIPALI	36
5.1.1 <i>Requisiti hardware</i>	36
5.1.2 <i>Requisiti software</i>	36
5.2 FUNZIONALITÀ DI MONITORAGGIO DELLE PRESTAZIONI	37
5.2.1 <i>Tipologie di strumenti di monitoraggio delle prestazioni delle applicazioni</i>	37

6.	GLI STRUMENTI PER IL MONITORAGGIO DELLE PRESTAZIONI DELL'APPLICAZIONE.....	38
6.1	ZABBIX	38
6.1.1	<i>Caratteristiche</i>	42
6.1.2	<i>Monitoraggio</i>	42
6.1.3	<i>Enterprise Ready</i>	42
6.1.4	<i>Monitoraggio pro-attivo</i>	43
6.1.5	<i>Pianificazione delle capacità</i>	43
6.1.6	<i>Open Source</i>	43
6.1.7	<i>Raccolta dei dati</i>	43
6.1.8	<i>Visualizzazione</i>	45
6.1.9	<i>Sistema di notifica</i>	48
6.2	XYMON.....	50
6.2.1	<i>Xymon gestisce il monitoraggio di molti sistemi</i>	50
6.2.2	<i>Xymon ha una configurazione centralizzata</i>	50
6.2.3	<i>Configurazione e installazione di Xymon</i>	51
6.2.4	<i>Xymon è in fase di sviluppo</i>	51
6.2.5	<i>Monitoraggio di host e reti</i>	51
6.2.6	<i>Multiplatforma</i>	51
6.2.7	<i>Front-end basato sul web</i>	51
6.2.8	<i>Analisi integrata dei trend, dati storici e reporting SLA</i>	52
6.2.9	<i>Visualizzazioni basate sui ruoli</i>	52
6.2.10	<i>Adattamento alle esigenze</i>	52
6.2.11	<i>Test di servizio di rete</i>	52
6.2.12	<i>Avvisi</i>	53
6.2.13	<i>Sicurezza</i>	53
7.	RIFERIMENTI	55

Indice delle figure

Figura 1: Architettura NUVOLA	8
Figura 2: Opzioni di connettività	10
Figura 3: VMware vCloud Director	11
Figura 4: Catalogo Pubblico.....	14
Figura 5: Caratteristiche dei ruoli.....	16
Figura 6: La console.....	19
Figura 7: Modello per garantire le prestazioni di applicazioni Cloud.....	26
Figura 8: Errori, difetti e malfunzionamenti	30
Figura 9: Verifica e validazione rispetto al ciclo di vita.....	31
Figura 10: Architettura Zabbix.....	39
Figura 11: Scenario semplice con Zabbix Proxy.....	40
Figura 12: Estensione dell'architettura Zabbix con proxies	41
Figura 13: Scenario con Zabbix Proxy e Firewall.....	41
Figura 14: Alcune funzionalità di un agente	44
Figura 15: Funzionalità agentless.....	45
Figura 16: Dashboard	46
Figura 17: Grafico standard.....	46
Figura 18: Grafico personalizzato	47
Figura 19: Esempio di mappa.....	47
Figura 20: Raw Data	48
Figura 21: Eventi e dettagli di notifica.....	48
Figura 22: Sistema di notifiche	48
Figura 23: Test di rete.....	53
Figura 24: Server test	54

1. Introduzione

Le attività per la valutazione delle performance del software che svolge le funzionalità di condivisione dell'informazione, possono essere diverse a seconda dell'applicazione selezionata che è dedicata a svolgere tale lavoro.

Nella prassi i termini verifica, validazione e test sono usati, con abuso di notazione, come sinonimi di controllo.

Occorre analizzare:

- i concetti di verifica e validazione;
- la differenza tra controlli statici e dinamici (test);
- le principali tecniche di controllo statico;
- i metodi per impostare un piano di test;
- i livelli di test: dei moduli, integrazione, controllo di sistema;
- i controlli di accettazione.

Nel semestre di riferimento del VI SAL, TIM, ha analizzato le possibili architetture Cloud nell'ottica di trovare la soluzione più idonea alla progettazione e successiva implementazione della Piattaforma "Framework SHELL". Sono state studiate e analizzate le tipologie di erogazione dei servizi cloud: IaaS, SaaS, PaaS, con lo scopo di identificare e definire i modelli di distribuzione dei servizi dell'infrastruttura da realizzare, e le problematiche a cui si va incontro con l'utilizzo di ciascuno di essi.

Ogni attività e funzionalità del Cloud Computing viene distribuita all'utente come servizio, ciò rende inevitabile affrontare il tema della qualità dei servizi (QoS). In ogni sistema orientato ai servizi le principali proprietà percepibili che riguardano la QoS sono, l'affidabilità, sicurezza e performance.

Partendo da tale analisi e, tenendo conto delle necessità implementative dei vari OR per la realizzazione dei servizi applicativi a servizio del Framework SHELL, il gruppo di lavoro Telecom Italia, di concerto con i vari responsabili di OR ha definito l'architettura dell'ambiente Cloud per SHELL.

La soluzione scelta Italia a servizio del Progetto SHELL e messa in esercizio è basata sulla piattaforma cloud SaaS Self Data Center (1) di Telecom, e istanziata sull'infrastruttura di Public Cloud Multitenant di Telecom Italia.

2. Definizione dell'infrastruttura cloud utilizzata per l'implementazione della Piattaforma "Framework SHELL"

Nel presente paragrafo viene presentata l'infrastruttura CLOUD scelta per la piattaforma Framework "SHELL"

2.1 Descrizione della piattaforma Nuvola IT Self Data Center Telecom Italia per SHELL

L'infrastruttura Cloud analizzata e scelta per il deploy della Piattaforma SHELL è Nuvola IT Self Data Center di Telecom Italia [1].

Il servizio Nuvola It Self Data Center, che è una soluzione di public CLOUD (SaaS), nasce per fornire un accesso rapido e sicuro a risorse computazionali con la massima flessibilità di configurazione delle stesse. L'utilizzatore ha a disposizione una porzione di risorse della Nuvola Italiana, che può configurare in autonomia. Il servizio viene erogato su una piattaforma multitenant basata su suite di prodotti VMware. L'utilizzatore può in qualsiasi

momento usufruire dell'accesso alla server farm, utilizzare i servizi di supporto e gli strumenti di self management.

Il servizio può essere implementato in due modalità: Pay-As-You-Go e Allocation Pool.

- La modalità **Pay-as-you-go** fornisce risorse secondo un modello *full sharing* (non ci sono risorse riservate) per lo specifico Cliente. E' previsto un canone per il diritto di accesso al servizio e agli strumenti di supporto. Inoltre il Cliente, utilizzando in autonomia le risorse (CPU, Memoria e Storage), pagherà le stesse sulla base dell'effettivo utilizzo (durata temporale dell'allocazione della risorsa).
- La modalità **Allocation Pool** fornisce risorse riservate al Cliente, che vengono pagate con un canone mensile. Incluso nel canone mensile l'accesso ad un ulteriore 33% di risorse elaborative (vCore e vRam) in full sharing per garantire la gestione di picchi di elaborazione del proprio Data Center Virtuale. Sono inoltre compresi nel canone il servizio di backup, l'antivirus e le licenze di sistema operativo Windows.

Per il Progetto SHELL si è scelto di erogare il servizio in modalità Allocation Pool che prevede la disponibilità di risorse elaborative riservate, con la possibilità di superare la soglia prestabilita di GHz e RAM (vCore e vRam) per un ulteriore 33% in full sharing (risorse non riservate) per garantire la gestione di picchi di elaborazione del proprio Data Center Virtuale. Vengono, inoltre, assegnati un pool di 3 indirizzi IP pubblici.

Le caratteristiche principali della piattaforma Cloud sono:

- È una soluzione di Private Cloud Virtuale su infrastruttura fisica shared;
- È accessibile via web e permette di istanziare server virtuali, aggiungendo o diminuendo capacità elaborativa e storage;
- Si basa su una architettura completamente ridondata, sia in termini di infrastruttura fisica che di rete, oltre che basata su un hypervisor con servizi di alta affidabilità (HA) e Dynamic Resource Scheduling (DRS);
- È possibile riconfigurare le proprie Virtual Machine in maniera rapida e flessibile andando ad aggiungere o eliminare risorse (vcore, RAM e storage) da dedicare ai propri server;
- La piattaforma basata su VMware garantisce la compatibilità delle proprie applicazioni sia windows che linux evitando di dover modificare il software.

L'infrastruttura, vista come un unico pool logico indipendentemente dalla tipologia di architettura implementata, può essere gestita mediante un portale di servizio. L'utilizzo della suite software VMware combina l'agilità del cloud pubblico con la sicurezza, le prestazioni e la portabilità applicativa di cui le aziende hanno bisogno.

Il servizio Nuvola It Self Data Center è realizzato sulla base dell'infrastruttura cloud sicura di VMware attingendo da VMware vSphere, VMware vCloud Director, VMware vShield, VMware vCenter Chargeback e VMware vCenter Operation architettati e certificati da VMware.

Alcune caratteristiche dell'infrastruttura tecnica a supporto del servizio erogato all'interno della Nuvola Italiana di Telecom Italia:

- I Data Center si trovano sul territorio Italiano e offrono i più alti livelli di sicurezza fisica, sorveglianza presente 24 ore su 24, protezione perimetrale, locali e sale sistemi con alimentazione ridondata e sistemi di controllo all'avanguardia.

- L'utilizzo dell'hypervisor VMware garantisce completa ridondanza fisica e possibilità di failover, eliminando di fatto la possibilità di downtime per fault hardware.
- Il bilanciamento automatico dei carichi di lavoro (VMware DRS) garantisce le performance necessarie anche durante i picchi di utilizzo delle applicazioni.
- È previsto l'utilizzo di storage in SAN con meccaniche Fiber Channel e SATA.
- Per la connettività dall'esterno è possibile prevedere accessi internet e/o MPLS
- Utilizzo di server di classe enterprise.
- Tutte le risorse elaborative (server, SAN e storage) sono completamente ridondate abilitando così caratteristiche di prestazioni e affidabilità anche per le applicazioni più critiche.

L'architettura è riportata in figura 1:

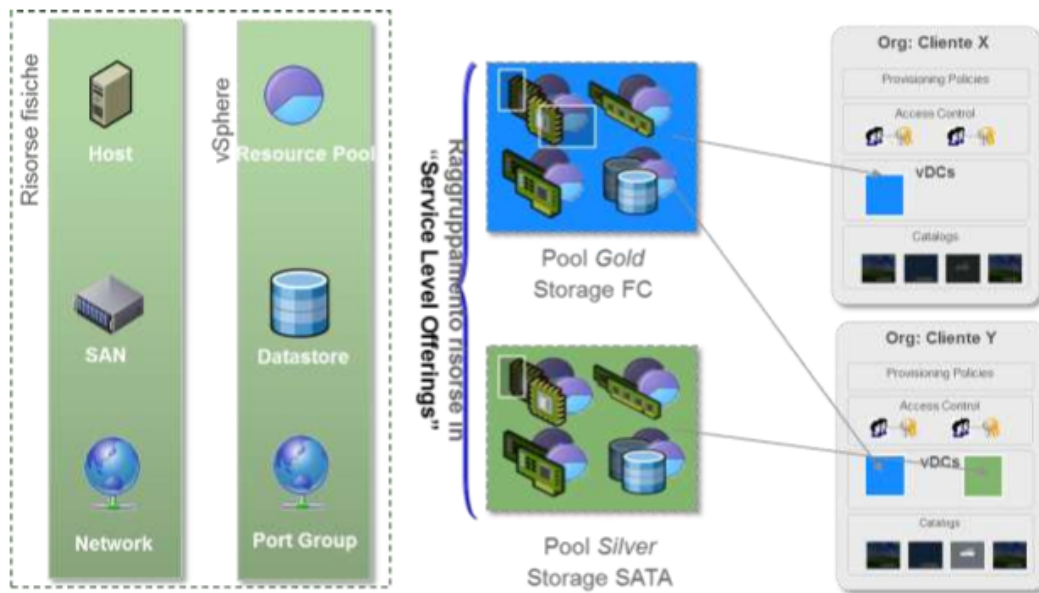


Figura 1: Architettura NUVOLA

Per il caso in esame è previsto l'utilizzo dello Storage FC.

Per quanto riguarda i servizi di accesso dall'esterno, sia la rete Internet che MPLS sono una "External Network" che corrisponde ad un Port Group su vSphere. L'accesso e l'erogazione del servizio avviene di default mediante connettività Internet su banda condivisa. Gli IP pubblici assegnati per l'accesso da internet hanno già una serie di porte/servizi abilitabili sui firewall perimetrali del Data Center come indicato in tabella 1:

Tabella 1: IP pubblici

source	port	type
internet	25/tcp	smtp
	465 /TCP	
	691 /TCP	
	53 / TCP	DNS
	53 / UDP	
	123 / TCP	
	123 / UDP	NTP
	123/tcp	
	123/udp	ssh/sftp ftp rdp
	22/tcp	
	21/tcp	
	3389/tcp	
	80 / TCP	http/https
8080 / TCP		
443 / TCP		
8443/tcp	snmp	
161/tcp		
161/UDP		
162/TCP		
162/UDP	LDAP LDAP/SSL	
389 /TCP		
636 /TCP		
379 /TCP		
390 /TCP		
3269 /TCP	IMAP4	
3268 /TCP		
143 /TCP		
993 /TCP	POP3	
110 /TCP		
995 /TCP		

Si possono gestire in autonomia l'apertura/chiusura delle porte indicate nella precedente tabella.

È disponibile il load balancer di piattaforma attraverso l'utilizzo di un virtual server che esegue il bilanciamento ad un pool di servers su uno specifico servizio. La configurazione di un pool inizia con la definizione dei servizi da bilanciare e delle service port utilizzate dai membri del pool. L'amministratore può selezionare tra servizi http, https e tcp. Ciascun servizio può utilizzare un diverso algoritmo di bilanciamento e gli algoritmi selezionabili sono: round-robin, URI e Least Connected. Possono inoltre essere configurati dei meccanismi di health check, si possono specificare pesi diversi per ciascun membro del pool ed inoltre ed è possibile specificare meccanismi di persistenza delle sessioni sulla base del protocollo utilizzato.

Per quanto riguarda i servizi di connettività interna è possibile connettere o isolare le VM all'interno del pool di risorse utilizzato. Sono disponibili 2 tipi di rete:

1. Diretta: Accessibile da più organizzazioni. Le macchine virtuali appartenenti a organizzazioni differenti possono connettersi a questa rete e visualizzarne il traffico. Tale rete fornisce una connettività diretta a livello del layer 2 alle macchine esterne all'organizzazione, che possono connettersi direttamente a quelle interne. Questo tipo di rete viene fornita in combinazione con l'accesso MPLS
2. Intradata: Accessibile solo da una organizzazione. Solo le macchine virtuali incluse nell'organizzazione possono connettersi a tale rete. Questo tipo di rete fornisce anche accesso controllato a una rete esterna. Gli amministratori di sistema e gli amministratori dell'organizzazione possono configurare le impostazioni NAT (Network Address Translation), del firewall e VPN in modo da rendere determinate macchine virtuali accessibili dalla rete esterna.

In figura 2, sono rappresentate le molteplici opzioni di connettività realizzabili all'interno dell'organizzazione e tra questa e il mondo esterno, che consentono la implementazione di landscape più o meno complessi.

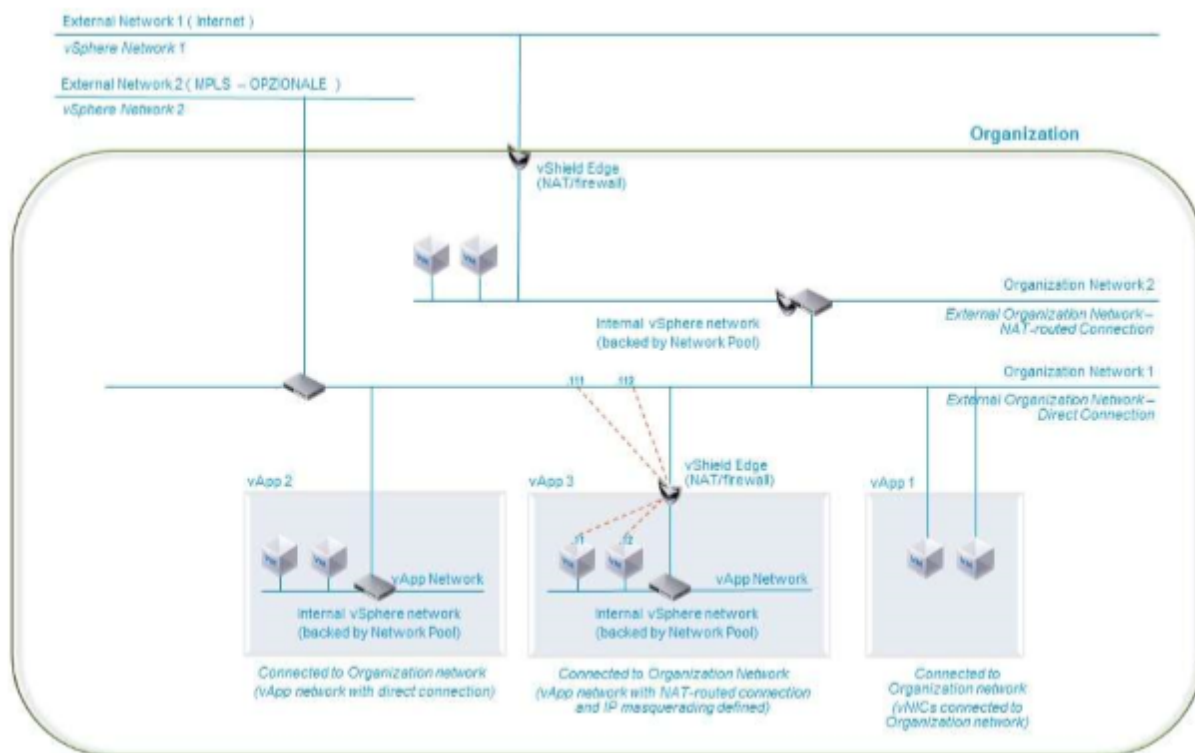


Figura 2: Opzioni di connettività

All'interno dell'organizzazione, infatti, è possibile avere:

- vApp con VM isolate e VM connesse su una rete interna isolata (vApp 1)
- vApp con VM connesse a una rete interna, ruotata (vApp2) sulla Organization Network
- vApp con VM connesse a una rete interna ruotata e con meccanismi di natting e firewalling (vApp3)

L'Organization Network, a sua volta, è connessa alla rete esterna direttamente, ovvero con meccanismi di natting/firewalling implementati dalla componente vShield Edge.

La soluzione tecnologica alla base del servizio cloud prevede l'utilizzo della soluzione VMware vCloud Director ed altri componenti della suite Cloud Infrastructure Management di VMware con la quale è possibile implementare contesti di sicurezza e rete isolati associando a ciascuno di essi risorse CPU, RAM e Storage con la possibilità di diverse modalità di allocazione e livelli di servizio.

La suite software utilizzata per erogare il servizio è la seguente:

Il servizio è implementato mediante la soluzione VMware vCloud Director ed altri componenti della suite VMware con la quale è possibile fornire, a diversi clienti, contesti di sicurezza e rete isolati associando a ciascuno di essi risorse CPU, RAM e Storage con la possibilità di diverse modalità di allocazione e livelli di servizio.

La suite software utilizzata per erogare il servizio è la seguente:

- VMware vCloud® Director™ 5.6.5

- VMware NSX™ 6.1.4
- VMware vCenter Chargeback™ 2.7
- VMware ESXi™ 5.5 (Update 3a)
- VMware vCenter Server™ 5.5 (Update 3°)
- VMware vRealize Operation Manager™ 6.1

VMware vCloud Director si basa su vSphere aggiungendo costrutti logici che facilitano l'utilizzo in multitenancy delle risorse (Figura 3).

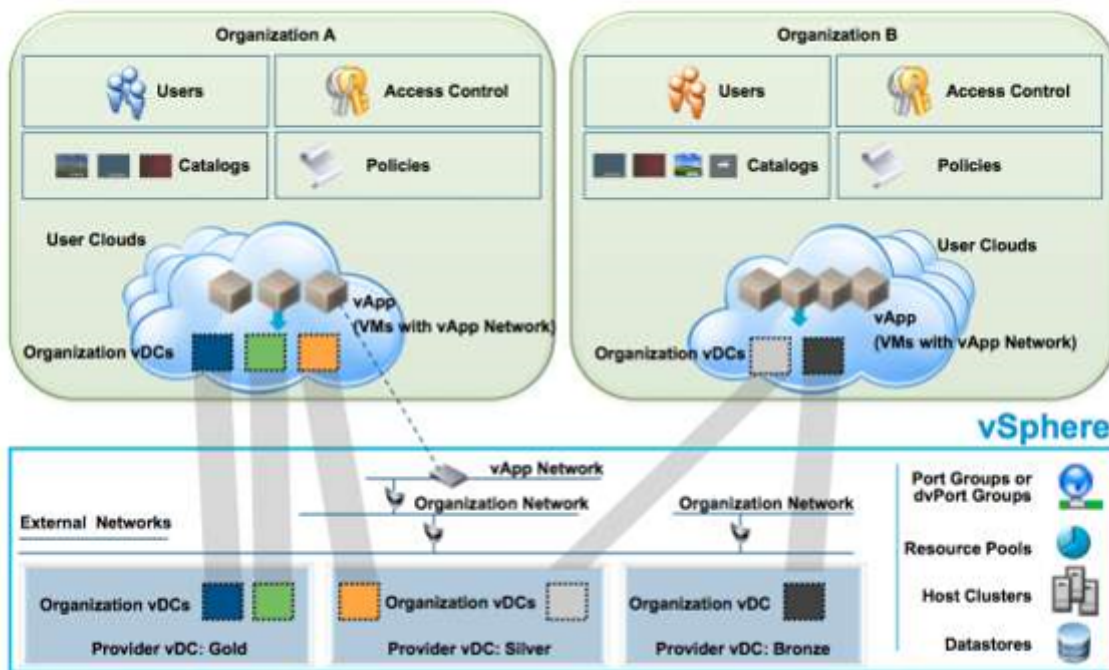


Figura 3: VMware vCloud Director

In Tab.2, un elenco dei costrutti di vCloud Director.

Tabella 2

Costrutto vCloud Director	Descrizione
<i>Organization</i>	Unità di amministrazione che rappresenta un raggruppamento logico di utenti, gruppi e risorse. Costituisce inoltre un confine di sicurezza entro il quale gli utenti possono attivare ed utilizzare Workload.
<i>Provider virtual Datacenter</i>	Insieme di risorse Server (Cluster vSphere o Resource Pool) e Storage fornite da un ambiente vSphere.
<i>Organization virtual datacenter</i>	Allocazione di un sottoinsieme di risorse fornite da un Provider virtual datacenter che sono assegnate ad un'Organization.
<i>vApp</i>	Contenitore di una o più macchine virtuali. In vCloud Director le VM sono sempre contenute in una vApp.
<i>vApp template and media catalogs</i>	Insieme di servizi disponibili per l'utilizzo. Un Catalog può contenere vApp Template e/o media (immagini ISO dei sistemi operativi).
<i>Internal and external organization networks</i>	Le reti virtuali che forniscono connettività di rete per vApp all'interno di un'Organization. Le Organization Network possono essere reti isolate utilizzate per la connettività tra vApp entro i confini dell'Organization (Organization internal network), o collegate a una rete esterna preconfigurata (VLAN – Port Group su vSphere), utilizzando una connessione diretta o NAT-Routed, per fornire la connettività al di fuori dell'Organization (External Organization Network). Alcuni tipi di Organization Network sono supportati da network pool.
<i>vApp network</i>	Rete virtuale contenuta all'interno di una vApp che consente la connettività di rete tra le macchine virtuali nella vApp. Una vApp Network può essere collegata ad una Organization Network utilizzando una connessione diretta o NAT-Routed per consentire la comunicazione con vApp nella organizzazione o all'esterno dell'organizzazione se l'Organization Network è connessa ad una rete esterna. Le vApp Network sono supportate da Network Pool. La maggior parte degli utenti con accesso ad un vApp può creare e gestire le proprie vApp Network.
<i>Network pool</i>	Un insieme di reti preallocate che vCloud Director può utilizzare per creare reti internal e NAT-Routed

2.1.1 Accesso al portale di gestione del servizio

All'attivazione del Servizio TIM comunicherà al Cliente tramite e-mail l'indirizzo web di accesso alla Console di Gestione dei Servizi <https://servizi.nuvolaitaliana.it/> e le Chiavi di Accesso. Le Chiavi di Accesso, limitatamente alla password, possono essere modificate a cura del Cliente già dal primo accesso al servizio. Il servizio mette a disposizione una serie di ruoli predefiniti con profilature diverse in base alle diverse esigenze di gestione.

Il servizio offre inoltre la possibilità di accedere alle funzionalità di gestione anche tramite **API (Application Programming Interface)** per eseguire ad esempio task automatici mediante l'accesso ad un servizio REST esposto sulla stessa interfaccia web.

La documentazione per le vCloud API è disponibile all'indirizzo riportato di seguito, TIM non fornisce supporto specifico sulle vCloud API:

<http://communities.VMware.com/community/vmtn/developer/forums/vcloudapi>

2.1.2 Rete di accesso e Indirizzi IP pubblici

Il servizio include connettività Internet con traffico illimitato. È possibile in caso di specifiche necessità di banda riservata (Internet o MPLS), ed in aggiunta alla connettività standard, la contrattualizzazione di un profilo di accesso tra quelli a catalogo previsti dall'offerta Data Center Solutions di TIM Per l'utilizzo del Servizio è inoltre sempre necessario l'acquisto di almeno un pool da 4 indirizzi IP pubblici di cui uno utilizzato per servizi di "routing" e quindi non direttamente utilizzabile dal Cliente. Qualora il Cliente abbia la necessità di ulteriori indirizzi pubblici può acquistarli opzionalmente. Gli indirizzi IP pubblici assegnati al Cliente hanno le seguenti porte abilitate sui firewall perimetrali del Data Center TIM:

- ❑ E' possibile accedere alla piattaforma tramite **Internet o MPLS**.
- ❑ E' possibile gestire in autonomia, per la propria organizzazione, la **apertura/chiusura** delle porte indicate nella tabella a lato
- ❑ In caso di necessità di aprire **altre porte/servizi**, il Cliente dovrà farne espressa richiesta a Telecom Italia per una verifica di fattibilità

source	port	type
internet	25/tcp	smtp
	465 /TCP	
	691 /TCP	
	53 / TCP	DNS
	53 / UDP	
	123 / TCP	NTP
	123 / UDP	
	123/tcp	ssh/sftp
	123/udp	
	22/tcp	ftp
	21/tcp	
	3389/tcp	rdp
	80 / TCP	
	8080 / TCP	http/https
	443 / TCP	
	8443/tcp	
	161/tcp	snmp
	161/UDP	
162/TCP		
162/UDP	LDAP LDAP/SSL	
389 /TCP		
636 /TCP		
379 /TCP		
390 /TCP		
3269 /TCP	IMAP4	
3268 /TCP		
143 /TCP	POP3	
993 /TCP		
110 /TCP		
995 /TCP		

L'apertura o chiusura di tali porte può essere gestita in autonomia dal Cliente sul proprio firewall virtuale tramite la Console di gestione del servizio. In caso di necessità di apertura di altre porte/servizi sui firewall perimetrali del Data Center, il Cliente deve farne espressa richiesta a TIM per una verifica di fattibilità.

2.1.3 Catalogo Cloud e vApp.

Un catalogo è un contenitore per i template di vApp e media files (es. immagini ISO).

La piattaforma cloud fornisce un catalogo standard di vApp Template utilizzabili da tutti gli utilizzatori (Catalogo Servizi Pubblico) ed è inoltre possibile per i clienti creare in autonomia cataloghi all'interno delle proprie Organization così come eseguire upload di proprie immagini .iso ed installare VM completamente nuove con sistemi operativi propri. Nel catalogo pubblico sono inseriti i seguenti template di vApp (Figura 4):

Catalogo Pubblico						
Modelli di vApp		File multimediale				
						Tutti i cataloghi
						Tutti
Nome	Descrizione	Stato	Gold Master	Publicati		
RHEL5x32vApp	Red Hat Enterprise Linux 5.5 32 bit	Pronti	👍	🔗	system	
RHEL5x64vApp	Red Hat Enterprise Linux 5.6 64 bit	Pronti	👍	🔗	system	
SLES10x32vApp	Suse Linux Enterprise Server 10 32 bit	Pronti	👍	🔗	system	
SLES10x64vApp	Suse Linux Enterprise Server 10 SP4 64 bit	Pronti	👍	🔗	system	
SQLWIN2K3x64vApp	Windows Server 2003 EE 64 bit SP2 con SQL	Pronti	👍	🔗	system	
SQLWIN2K8R2vApp	Windows Server 2008 EE R2 con SQL Server	Pronti	-	🔗	system	
WIN2K12-DatacenterE...		Pronti	👍	🔗	system	
WIN2K12-StandardED...		Pronti	👍	🔗	system	
WIN2K3x32vApp	Microsoft Windows Server 2003 Enterprise Ed	Pronti	👍	🔗	system	
WIN2K3x64vApp	Microsoft Windows Server 2003 Enterprise Ed	Pronti	👍	🔗	system	
WIN2K8R2vApp	Windows Server 2008 R2 Enterprise SP1	Pronti	-	🔗	system	

Figura 4: Catalogo Pubblico

Partendo da questi template si possono creare le vApp/VM e successivamente modificare le risorse assegnate. Si possono, comunque, istanziare VM con altri Sistemi Operativi, purché compatibili con VMware, utilizzando proprie immagini ISO in locale.

2.2 Il servizio di backup

Per le VM costituenti l'architettura del Cliente, TIM mette a disposizione adeguate funzionalità per il salvataggio ed il ripristino dei dati, basate sull'utilizzo di una infrastruttura tecnologica centralizzata basata sulla suite software Veeam Backup&Replication v9.

I servizi di Backup sono erogati e gestiti da TIM per quanto riguarda gli aspetti infrastrutturali. In linea con la filosofia di "self managing" dell'offerta Self Data Center vCloud, i Clienti avranno deleghe per eseguire in autonomia le azioni di restore di intere VM della propria infrastruttura così come di restore di file nelle VM.

L'accesso al portale di SelfRestore deve avvenire da una VM interna all'organization del cliente aprendo una finestra di un web browser al link <https://selfdcvcloud-restore.nuvolaitaliana.it:9443>

2.3 Portale di SelfRestore

Accedendo al portale di SelfRestore, l'utente (con profilo autorizzato) autonomamente potrà:

- visualizzare l'elenco delle VM di propria pertinenza sottoposte a backup con i relativi Restore Points.
- Effettuare il restore dell'intera VM nella farm SelfDC a partire da un dato restore point
- Effettuare il restore di uno o più file/cartelle di una VM

Per motivi di sicurezza qualunque restore (VM, file o cartelle) avverrà nella cartella originale (quella in cui il file era presente quando è stato effettuato il backup).

Il portale può essere attivato (solo se presente sul desktop di macchine windows) cliccando l'apposita icona (SelfDC backup restore) presente sul desktop di ogni VM. In caso di assenza dell'icona sul desktop il link per attivare l'applicazione è : <https://selfdcvcloud-restore.nuvolaitaliana.it:9443/>

La funzionalità di Image Level Backup consente di effettuare una copia di tutti i Server del proprio Data Center Virtuale, utilizzabile poi successivamente per ripristinare sia il singolo file che l'intera immagine del Server. La copia è schedulata in automatico dalla piattaforma in funzione della policy contrattualizzata. Il restore del Server Virtuale o del singolo file è richiedibile tramite un'apposita console web le cui Chiavi di Accesso vengono fornite via mail in fase di attivazione del servizio e, limitatamente alla password, possono essere cambiate a cura del Cliente già dal primo accesso al Servizio. La console è accessibile esclusivamente dall'interno, ossia da Server Virtuali interni al servizio mediante web browser indirizzando una specifica URL ed inserendo le proprie credenziali sulla login page.

Il costo della funzionalità di backup fatturato come consumo è funzione del profilo e della quantità di Spazio Disco allocato, indipendentemente dallo spazio disco occupato, secondo quanto riportato nel paragrafo 11. La policy base è inclusa sempre nel canone mensile del profilo Allocation Pool.

2.4 Il servizio di antivirus

La soluzione scelta per il servizio di antivirus è basata sulla piattaforma Deep Security di Trendmicro.

Il prodotto permette un'alta profilabilità degli Utenti, che pur avendo accesso alla console di gestione, possono essere autorizzati (vincolati) ad operare solo su un sottoinsieme dell'intera infrastruttura ospite, che nel nostro caso coincide con la Organization e con le VM del singolo Cliente.

Nell'ambito della singola Organization è possibile delegare al Cliente alcuni compiti specifici quali, ad esempio, la pianificazione di scansioni "spot" o la esclusione/inclusione di particolari file/database da modelli di scansione, ecc.

Nel servizio TIM Self Data Center sarà utilizzata la funzionalità di anti-malware che supporta:

- protezione anti malware agentless
- protezione di macchine virtuali attive
- possibilità di scan schedulati, manuali ed anche di protezione in tempo reale
- utilizzo del trend micro smartprotection network
- gestione dei files in quarantena, includendo il download e la cancellazione dal Deep Security Manager
- supporto anti malware integrato in dashboard widgets e reports

Nel dettaglio le azioni intraprese di default dal prodotto alla rilevazione dei diversi tipi di malware sono indicate nella seguente tabella:

Malware Type	Real-Time Scan		Manual Scan or Scheduled Scan	
	First Action	Second Action	First Action	Second Action
Joke	Quarantine	Delete	Quarantine	Delete
Trojan	Quarantine	Delete	Quarantine	Delete
Virus	Clean	Quarantine	Clean	Quarantine
Test Virus	Deny Access	N/A	Pass	N/A
Spyware	Quarantine	Delete	Quarantine	Delete
Packer	Quarantine	N/A	Quarantine	N/A
Possible malware	Pass	N/A	Pass	N/A
Others	Clean	Quarantine	Clean	Quarantine

La console è accessibile mediante web browser indirizzando una specifica URL ed inserendo le proprie credenziali sulla login page. Per il servizio di Self Data Center, la console sarà accessibile esclusivamente dall'interno, ossia da virtual machine interne all'Organization del Cliente.

2.5 Profili degli utilizzatori della piattaforma

Per quanto riguarda i profili degli utilizzatori è stato già detto che la soluzione si basa su una architettura multitenant. In Figura 5, le caratteristiche dei ruoli predefiniti. Il System Administrator può creare nuovi ruoli a partire dai permessi disponibili.

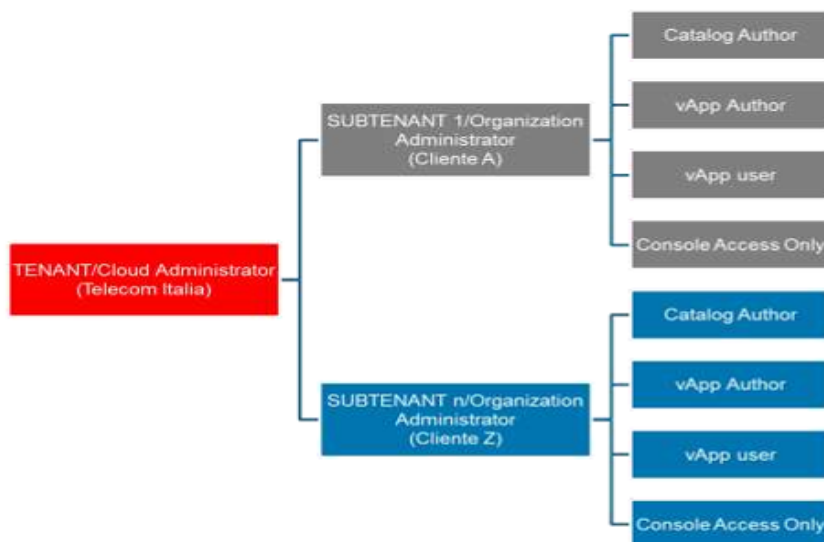


Figura 5: Caratteristiche dei ruoli

TENANT/Cloud Admin (System Administrator). È il ruolo svolto dal Service Provider (Telecom Italia) ed è il "root user" dell'infrastruttura Cloud. È l'unico ruolo predefinito che ha permessi validi su tutti i costrutti vCloud Director e su tutte le Organization:

- Fa il deployment dell'infrastruttura e la gestisce
- Associa il vcenter server
- Crea i virtual Data Center (associa le risorse computazionali), le External Networks (fornisce l'accesso dall'esterno al pool di risorse) e i Network Pools (crea un insieme di reti di livello 2)

- Crea le Organizations
- Crea gli Organization VDCs (es. pool di risorse con storage FC) e le Organization Networks (per la connettività delle vApp all'interno di una organization)
- Può creare e gestire ruoli ed utenti

SUBTENANT/Organization Administrator. Con questa utenza l'utilizzatore può creare nuovi utenti e assegnare dei ruoli. È il service administrator lato utente ed è il "root user" dell'organization:

- Fa il deployment delle utenze e le gestisce
- Crea i cataloghi specifici per l'organization
- Gestisce le policy dell'organization (lease, quotas e limits) e la configurazione delle notifiche via mail
- Può visualizzare e configurare impostazioni Firewall/NAT/DHCP/VPN sulla Organization Network NAT-Routed

SUBTENANT/Catalog Author. Ruolo assegnato dall'Organization Administrator nella propria Organization:

- Crea cataloghi nella propria organization
- Può creare e gestire vApp modificandone le impostazioni

SUBTENANT/vApp Author. Ruolo assegnato dall'Organization Administrator nella propria Organization:

- Può creare e gestire vApp modificando anche parametri virtual hw delle VM
- Ha inoltre gli stessi diritti di un vApp User
- Può cancellare le proprie vApp
- Può eseguire operazioni su vApp: Start/Stop/Suspend/Reset ed accesso VM Console
- Può eseguire Copia/Move vApp

SUBTENANT/vApp User. Ruolo assegnato dall'Organization Administrator nella propria Organization, è il ruolo tipico del System Administrator per uno o più workload:

- Può cancellare (ma non creare) le proprie vApp
- Può modificare le impostazioni delle proprie vApp
- Può eseguire operazioni su vApp: Start/Stop/Suspend/Reset ed accesso VM Console
- Può eseguire Copy/Move vApp
- Modifica impostazioni VM tranne parametri Virtual Hardware

SUBTENANT/Console Access Only. Ruolo assegnato dall'Organization Administrator nella propria Organization, è il ruolo tipico dell'utente che deve interagire solo con il sistema operativo guest delle VM contenute in una vApp. Può visualizzare ed interagire con la VM console delle VM contenute in una vApp.

2.5.1 Strumenti a disposizione dell'Amministratore (Cliente)

La piattaforma Cloud Self Data Center, mette a disposizione i seguenti strumenti di amministrazione:

- Web User Interface. È l'interfaccia grafica attraverso la quale si può gestire in autonomia il servizio.
- API (Application Programming Interface). È un insieme di procedure disponibili per eseguire task su vCloud Director mediante l'accesso ad un servizio REST esposto sulla stessa interfaccia web della UI di vCloud Director.

Metriche a disposizione dell'Amministratore

Le metriche disponibili sono relative ai seguenti parametri.

- **Disk Usage**
 - Spazio disco allocato alle VM
- **GHz**
 - Quantità di GHz utilizzati dalle VM
- **RAM**
 - Quantità di RAM utilizzata dalle VM
- **Sistema operativo windows**
 - Quantità di VM con SO Windows
- **DBMS SQL server**
 - Quantità di VM con DBMS Microsoft SQL
- **Backup**
 - E' la quantità di GB sottoposti a backup
- **Antivirus**
 - E' la quantità di VM che utilizzano il servizio di antivirus

2.5.2 Elementi del portale di gestione

Attraverso l'inserimento nel browser dell'indirizzo del portale TIM Self Data Center https://selfdcvcloud.nuvolaitaliana.it/cloud/org/<Org_name> e l'inserimento delle credenziali assegnate è possibile effettuare il login alla console di gestione

La console di gestione del servizio

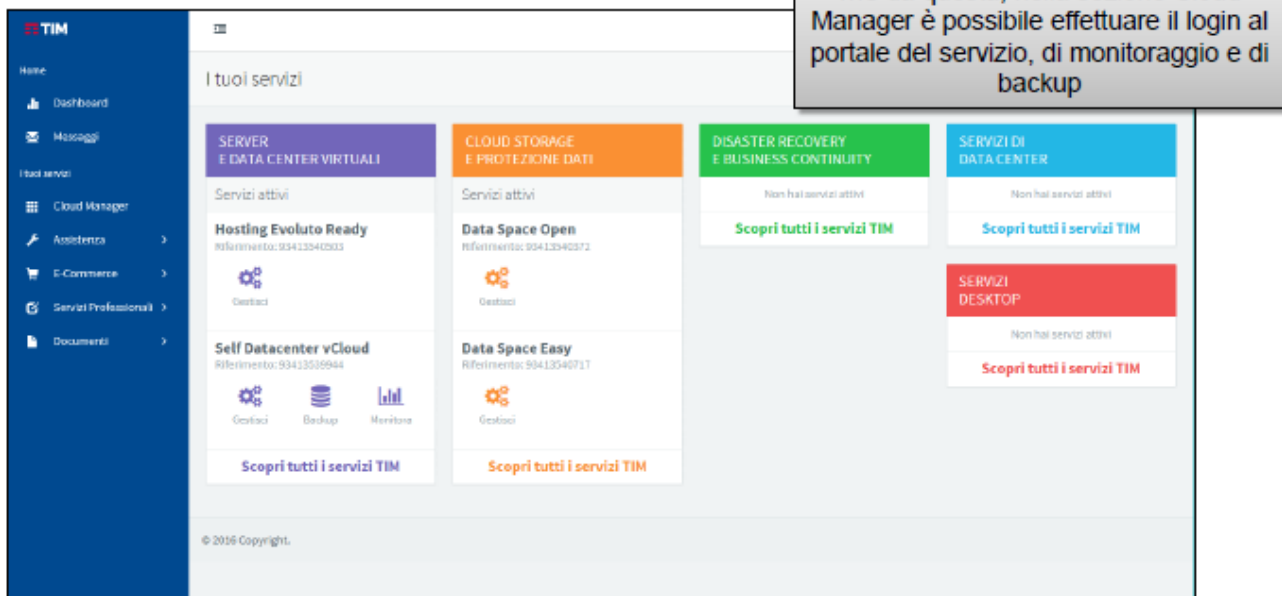


Figura 6: La console

2.6 Reporting

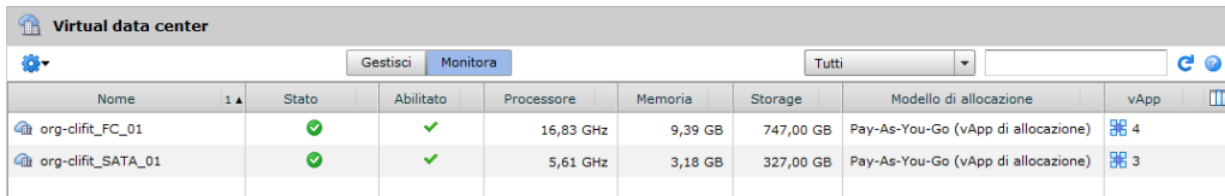
Il Cliente ha a disposizione una apposita reportistica consultabile sul portale **TUconTI 2.0**. Il Cliente può visualizzare il consumo delle risorse con lo stesso livello di dettaglio previsto per la fatturazione. Le metriche disponibili sono relative ai seguenti parametri: **Disk Usage** (Spazio disco allocato sui Server Virtuali), **GHz** (Cicli di CPU utilizzati dai Server Virtuali), **RAM** (Quantità di RAM allocata ai Server Virtuali), **Sistema operativo Windows** (Quantità di Server Virtuali con SO Windows), **DBMS SQL Server** (Quantità di Server Virtuali che utilizzano MS SQL Server).

Oltre ai report sul consumo delle risorse sono disponibili su TUconTI 2.0 anche i report relativi ad un summary degli SLA e ai ticket aperti dal Cliente.

All'interno della console di gestione del servizio inoltre sono disponibili anche ulteriori funzionalità che, consentono di visualizzare le risorse utilizzate al momento in termini di GHz, RAM e GB Storage.

2.6.1 Reporting TIM Self Data Center

All'interno della piattaforma TIM Self Data Center saranno inoltre disponibili delle funzionalità che, pur non avendo valenza contrattuale consentono di visualizzare le risorse utilizzate al momento in termini di Ghz, RAM e GB Storage (vedi immagine seguente)



The screenshot shows a web interface for a 'Virtual data center'. At the top, there are tabs for 'Gestisci' and 'Monitora', and a dropdown menu set to 'Tutti'. Below this is a table with the following columns: Nome, Stato, Abilitato, Processore, Memoria, Storage, Modello di allocazione, and vApp. Two rows of data are visible:

Nome	Stato	Abilitato	Processore	Memoria	Storage	Modello di allocazione	vApp
org-clifit_FC_01	✓	✓	16,83 GHz	9,39 GB	747,00 GB	Pay-As-You-Go (vApp di allocazione)	4
org-clifit_SATA_01	✓	✓	5,61 GHz	3,18 GB	327,00 GB	Pay-As-You-Go (vApp di allocazione)	3

Per ogni virtual Data Center è possibile inoltre avere il dettaglio a livello di vApp e VM

2.7 Sicurezza

La sicurezza della piattaforma di erogazione dei servizi cloud di Telecom Italia è articolata su più livelli:

- **Sicurezza fisica degli ambienti di erogazione del servizio.** Data center costruiti con impianti all'avanguardia, sorveglianza presente 24 ore su 24, protezione perimetrale, locali interni e sale sistemi.
- **Certificazione ISO/IEC 20000:2005** per la Gestione dei Servizi Informatici. Rappresenta uno strumento di riferimento per l'organizzazione dei servizi informatici che mira al miglioramento dell'erogazione/fruizione dei servizi IT, ponendosi come obiettivo il raggiungimento della massima qualità dei servizi erogati e un massimo contenimento di costi.
- **Certificazione UNI EN 9001:2008** per attività di progettazione, sviluppo e attivazione di componenti infrastrutturali per l'erogazione di servizi cloud in modalità IAAS
- **Certificazione ISO14000**
- **Certificazione ISO27001** per i servizi di delivery, operation e sicurezza fisica dei DC di Telecom Italia
- Applicazione dell'**Information Security Management System** per la struttura IT Service Management in conformità ai requisiti standard ISO/IEC 27001 relativamente a
 - processi di Delivery ed Esercizio per l'erogazione di soluzioni Housing e Hosting nei DC di Telecom Italia
 - servizi di PdL Management, LAN Management e System Management nella conduzione dei sistemi per il mercato e per le pubbliche amministrazioni
- La Nuvola Italiana ha ottenuto la certificazione **CSA STAR**. La Cloud Security Alliance (CSA) è un'associazione internazionale che opera con lo scopo di promuovere l'utilizzo di best practice per la sicurezza del cloud computing, insieme alla formazione e sensibilizzazione nell'utilizzo sicuro del cloud
- **Sicurezza nell'accesso ai servizi dall'esterno.** I server di front end del portale di servizio sono collocati in una DMZ per fornire servizi all'esterno senza compromettere la sicurezza della rete aziendale interna oltre alla definizione di opportune policies sui firewall del Data Center
- **Sicurezza della suite sw VMware.** Relativamente alla piattaforma informatica ci sono diversi aspetti da tenere in considerazione
 - Le funzionalità di sicurezza del software VMware partono dalla piattaforma vSphere che segue i criteri per la certificazione EAL4+.

- Anche per le funzionalità di rete, Firewall, VPN, Routing e NAT, la componente vShield segue i criteri per la certificazione EAL4+, fornendo funzionalità di firewall fra le varie vApp ed isolamento fra le Organization. I contesti di rete fra le diverse Organization sono isolati a livello L2.
- L'accesso all'interfaccia web di vCloud Director ed alla Console delle VM avviene in maniera cifrata mediante SSL

2.8 Assistenza tecnica e SLA di disponibilità del servizio

La disponibilità dell'infrastruttura virtuale è misurabile sulla base dei seguenti parametri di funzionalità operativa:

A) Connettività di Data Center: Uptime del 99,99% su base mensile, di accessibilità tramite rete Internet o MPLS alla Infrastruttura Virtuale del Cliente.

B) Nodi fisici: Uptime del 99,99% su base mensile, per la disponibilità dei nodi fisici (server) che ospitano l'Infrastruttura Virtuale del Cliente.

Nuovo modello di assistenza H24*365GG, basato su tempi di gestione delle segnalazioni!			
Profilo di servizio	Incluso nel servizio	PROMO -40%	
	Standard	Professional	Business
Disponibilità oraria Canali di Assistenza	H24 per 365gg		
Gestione priorità Trouble Ticket (TT)	P1 (ticket prioritari)/ P2 (ticket non prioritari)		
Tempo Max presa in carico TT	12h (P1)	2h (P1)	15 min (P1)
	24h (P2)	4h (P2)	1 h (P2)

È possibile richiedere assistenza tramite:

- La sezione assistenza della Console di Gestione Servizi
- Numero verde 800.199.477, pin 0082

Problem Resolution

2.8.1 Dati di targa

Di seguito i principali elementi di misurazione:

- L'attivazione di un Server Virtuale avviene entro 20 minuti (calcolo applicato solo per Server Virtuale istanziati a partire dai template del Catalogo Pubblico TIM)
- La variazione di un Server Virtuale avviene entro 20 minuti
- Disponibilità del portale di gestione del servizio al 99,8%. Tale valore viene calcolato sulla disponibilità di accesso alla home page del portale utilizzato dal Cliente

2.9 Dimensionamento della piattaforma Cloud Nuvola IT Self Data Center per SHELL

Il dimensionamento delle risorse allocate in Cloud, del Servizio Self Data Center di Telecom Italia, destinate al Progetto SHELL, sulle quali sono state predisposte e configurate le varie macchine virtuali e server virtuali, è il seguente:

Profilo: Allocation Pool

Il servizio configurato per il progetto SHELL definito **Allocation Pool** prevede:



Con il seguente dimensionamento:

Pool da 4 IP Pubblici	1
Assistenza	Standard
Portale di monitoraggio vRealize	Si
Numero Pool Base (6 GHz, 24 GB di RAM, 750 GB spazio disco FC)	0
Numero Pool Base SSD (6 GHz, 24 GB di RAM, 750 GB spazio disco SSD)	1
Blocchi di Upgrade di risorse elaborative (2 GHz, 8 GB di RAM)	1
Blocchi di Upgrade di CPU (2 GHz)	4
Blocchi di Upgrade di RAM (8 GB di RAM)	0
Blocchi di Upgrade di Storage FC (250 GB)	0
Blocchi di Upgrade di Storage SSD (250 GB)	1
Blocchi Storage NAS (500 GB)	0
Pacchetti licenze MS Remote Desktop Services (10 utenti)	0

All'interno del canone del profilo Allocation Pool sono compresi:

- Risorse oltre soglia per la gestione dei picchi;
- Licenze S.O. Windows;

- Antivirus;
- Backup Base.

Nella tabella successiva è riportato il valore complessivo di risorse previste.

GHz riservati	RAM riservata (GB)	Storage FC riservato (GB)	Storage SSD riservato (GB)
16	32	0	1000
GHz non riservati per gestione picchi	RAM non riservata (GB) per gestione picchi	# IP Pubblici *	
7	11	4	

* di cui 1 utilizzato per servizi di "routing" e quindi non direttamente utilizzabile dal Cliente

2.10 Descrizione delle politiche di costo della piattaforma Cloud Nuvola IT Self Data Center Telecom Italia

In generale le politiche di costo delle piattaforme Public Cloud SaS sono legate al consumo di risorse elaborative e storage ed alla loro disponibilità condivisa o dedicata.

L'offerta TIM Self Data Center fornisce risorse in due modalità **pay-as-you-go** e **allocation pool**.

- La modalità **Pay-As-You-Go** prevede un modello di risorse in full sharing (non ci sono risorse riservate). In questo caso si fa pagare GHz, vRAM e storage su base oraria.
- La modalità **Allocation Pool** prevede la disponibilità di risorse elaborative *riservate*, con la possibilità di superare la soglia prestabilita di GHz e RAM per un ulteriore 33%. In questa modalità il Cliente paga il pool di risorse riservate sulla base di un canone mensile e può utilizzare risorse aggiuntive (se disponibili) per gestire situazioni che necessitino di ulteriore capacità computazionale. Si rimanda all'Appendice 2 per il dettaglio della composizione degli stessi e sulla quantità di risorse riservate e quantità massima di risorse disponibili.

2.10.1 Pay-as-you-go

Quando si utilizza il modello Pay-As-You-Go il cliente paga in base alle risorse virtuali allocate alle singole VM solo quando queste sono utilizzate. Nel servizio TIM Self Data Center le risorse per tutte le VM che saranno create non avranno una percentuale di reservation impostata ma saranno in full sharing e non sarà posto un limite al numero massimo di VM.

2.10.2 Allocation pool

Con il modello Allocation Pool il cliente paga la preallocazione delle risorse ad un Organization vDC.

Il provider (TIM) ha il controllo dell'overcommitment e definisce una percentuale di risorse CPU e memoria garantite e può impostare un numero massimo di virtual machine. Il provider può aumentare le risorse allocate in ogni momento.

Tale modello fornisce la possibilità di acquistare una quantità di risorse garantita con la possibilità di avere dei picchi di utilizzo superiori se necessario. Poiché solo la percentuale riservata è garantita, la possibilità di contesa di risorse è eventualmente riscontrabile solo sulla capacità elaborativa oltre soglia utilizzabile.

Le licenze di Sistema Operativo, di backup dei dati e l'Antivirus su VM sono comprese nel canone per chi sottoscrive il servizio in modalità Allocation Pool e sono invece soggette a pagamento su base giornaliera per chi abbia sottoscritto il servizio in modalità Pay-as-you-Go.

I sistemi operativi Windows saranno messi a disposizione e fatturati (per il profilo Pay as You Go) da TIM mentre i sistemi operativi Linux saranno presenti nel catalogo ma sarà cura del Cliente sottoscrivere, se necessario, il servizio di supporto con il relativo vendor (es. Red Hat, Suse, ...).

Il cliente inoltre può attivare in autonomia da catalogo condiviso VM con SQL Server Std Edition preinstallata; per queste ultime è previsto un pricing su base giornaliera per entrambi i Profili.

Per una maggiore flessibilità, non sono previsti vincoli particolari al numero di VM che il cliente può creare. Il cliente ha a propria disposizione una console con un set di risorse che possono essere utilizzate sia con poche grosse VM che con tante piccole VM.

In prima istanza viene assegnato inizialmente un a pool di 3 ip pubblici disponibili n.1 utilizzato per il routing), con la possibilità poi di contrattualizzarne ulteriori con granularità pool da 4.

Le voci di costo, nel caso di Cloud in soluzione Allocation pool sono le seguenti:

VOCI DI COSTO
Contributo attivazione
Assistenza
Pool 4 Indirizzi IP Pubblici
Pool Base (6 GHz, 24 GB di RAM, 750 GB di spazio disco)
Upgrade Risorse di Calcolo
Blocchi Storage Aggiuntivi
Image Level Backup
Image Level Backup PREMIUM
Pacchetto di formazione
Portale di Monitoraggio
Pacchetto 10 Licenze RDP
MS SQL server Std. Edition

Ovvero oltre al contributo attivazione iniziale (Una Tantum) si paga un canone mensile fisso in funzione delle risorse cloud specificatamente allocate.

3. Prestazioni

Prima della nascita del cloud computing, la pianificazione delle strutture di calcolo era devoluta agli sviluppatori delle applicazioni. Ai fini di garantire adeguate prestazioni, occorreva determinare la quantità di risorse di calcolo (ad esempio, CPU, memoria) e le necessità che devono essere acquisite, in base ai requisiti per lo sviluppo del software e alla stima del carico futuro. I requisiti del software vengono individuati con operazioni di collaudo e monitoraggio, ma il carico è imprevedibile e variabile.

Con l'utilizzo del cloud computing, tale pianificazione delle risorse non è necessaria. Si acquistano via via le risorse da un provider di cloud in base al variare del carico di lavoro e si sottoscrivono, come abbiamo visto nei paragrafi precedenti, accordi di SLA (accordi bilaterali chiamati Service Level Agreement (SLA), che specificano la qualità del servizio richiesta e le penalità associate alle violazioni di tali accordi).

Tuttavia, a causa della complessa architettura del cloud e dell'interazione tra le varie applicazioni che condividono le risorse di calcolo, è molto difficile che la SLA di ogni applicazione venga rispettata. L'esperimento condotto in [2] mette a confronto le prestazioni di applicazioni distribuite in MapReduce, Amazon EC2 e in un cluster locale, e il risultato dimostra che MapReduce soffre di variazioni in prestazioni notevolmente più elevate che in EC2.

Un approccio comune del cloud, per il mantenimento della SLA per tutte le richieste degli utenti è di monitorare il funzionamento dell'intero cloud, raccogliere dati di runtime e conseguentemente adeguare il carico delle risorse [3].

Altre misurazioni ottenute dai test su alcuni cloud commerciali [*Qiang Duan, Modeling and Performance Analysis on Network Virtualization for Composite Network-Cloud Service Provisioning, 2011.*], come Amazon EC2, indicano che le prestazioni della rete hanno un impatto significativo sulla qualità di servizio del cloud e ciò rende la performance uno degli attributi fondamentali.

3.1 Prestazioni tramite una fase di monitoraggio

Le prestazioni di un'applicazione cloud sono determinate dall'interazione di diversi fattori.

Il modello [4] tenta di considerare tutti i fattori che influiscono sulle prestazioni. Tale modello è schematizzato in figura 7.

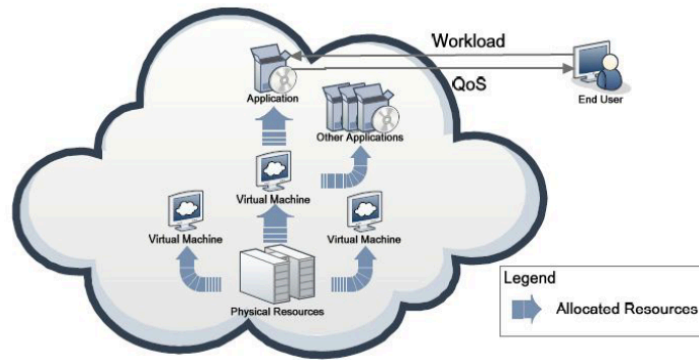


Figura 7: Modello per garantire le prestazioni di applicazioni Cloud

Questo modello parte da due attributi di QoS: l'affidabilità e dal response time (cioè il tempo di risposta che è il tempo trascorso dal momento nel quale l'utente invia una richiesta a quando la richiesta viene soddisfatta).

Si è soliti misurare la disponibilità di un servizio cloud sulla base della percentuale di volte in cui, in un periodo temporale prefissato, il servizio stesso è pubblicamente accessibile. Tale valore viene chiamato uptime e fa riferimento all'ammontare di tempo, su base percentuale, in cui un server web resta attivo e accessibile dall'esterno. Molti hosting provider pubblicizzano la percentuale di uptime per indicare la percentuale di disponibilità dei propri servizi su base annuale, mensile e/o settimanale.

L'unità di misura complementare è il downtime (o tempo di disservizio), che viene calcolato a partire dalla presenza di un malfunzionamento delle infrastrutture di pertinenza (o per la loro mancata disponibilità/accessibilità) e sino al momento in cui tali risorse risultano nuovamente disponibili.

Esso può essere causato da manutenzione delle macchine server, problemi di rete o configurazioni errate delle stesse.

Il downtime è quindi ottenuto (in percentuale), come $100 - \text{uptime}$. Tale unità di misura, per quanto indicativa e basata su modelli statistici, può essere utile a conoscere la disponibilità dei servizi che si stanno per acquistare.

La tabella di seguito mostra alcuni esempi di uptime e downtime in più unità di misura temporali: essa serve a comprendere, ad esempio, che un uptime del 90% (piuttosto scarso) equivale a circa 36 giorni di disservizio all'anno (non necessariamente consecutivi). Percentuali ottimali di uptime vanno dal 99,1% al 99,5% e oltre.

Uptime %	Downtime su base annuale	Downtime su base mensile*	Downtime su base settimanale
90%	36.5 giorni	72 ore	16.8 ore
95%	18.25 giorni	36 ore	8.4 ore
97%	10.96 giorni	21.6 ore	5.04 ore
98%	7.30 giorni	14.4 ore	3.36 ore
99%	3.65 giorni	7.20 ore	1.68 ore
99.5%	1.83 giorni	3.60 ore	50.4 minuti
99.8%	17.52 ore	86.23 minuti	20.16 minuti
99.9%	8.76 ore	43.2 minuti	10.1 minuti
99.95%	4.38 ore	21.56 minuti	5.04 minuti
99.99%	52.56 minuti	4.32 minuti	1.01 minuti

L'affidabilità è il rapporto tra "uptime / (uptime + downtime)", cioè "numero di richieste soddisfatte / (richieste soddisfatte + richieste non soddisfatte)"

Il response time è la devianza standard dal tempo medio di risposta e tale valore indica come variano le prestazioni.

Sulla base di questi due parametri si ottiene una metrica per le prestazioni. Infatti, le prestazioni vengono calcolate come rapporto tra l'affidabilità e tempi di risposta.

Un'applicazione sul cloud è distribuita su un server di applicazioni in una macchina virtuale. A causa della limitazione delle risorse però, nella stessa macchina virtuale altre applicazioni possono andare in concorrenza.

La modellazione delle prestazioni è un processo iterativo. Il primo passo è l'ottenimento dei dati di un training set, con il monitoraggio continuo dell'esecuzione sul cloud.

Tale raccolta dati viene eseguita a intervalli di 5 minuti sulle seguenti risorse: memoria totale, numero processori utilizzati (sia virtuali che fisici), numero di applicazioni residenti su macchina virtuale, utilizzo CPU, consumo risorse, numero di connessioni per minuto.

Un algoritmo di regressione lineare permette poi di determinare un valore da attribuire alle prestazioni.

Questo approccio deve tenere però tenere in considerazione che tutti i requisiti di SLA, devono essere monitorati continuamente. Per il mantenimento di performance accettabili, con l'aumentare del numero di servizi resi il provider del servizio dovrà aggiungere, via via, ulteriori risorse.

3.1.1 QoS per il cloud

Ogni attività e funzionalità del Cloud Computing viene distribuita all'utente come servizio, ciò rende inevitabile discutere la qualità dei servizi. In ogni sistema orientato ai servizi le principali proprietà non funzionali che sono percepibili dal cliente e riguardano la QoS sono l'affidabilità, sicurezza e performance.

Le prime due sono due aspetti del cloud strettamente collegate tra di loro mentre, i problemi riguardanti le prestazioni sono percepiti dall'utente come il tempo di risposta, il throughput e l'utilizzo della rete. Altro aspetto importante che riguarda la QoS è la disponibilità delle risorse. Un altro attributo di QoS che è difficile da soddisfare, soprattutto a causa della natura sconosciuta dei consumatori di servizi e dei volumi imprevedibili di chiamate di servizio, è la scalabilità del sistema.

3.1.2 Scalabilità

La scalabilità è fondamentale per il successo del servizio cloud poiché, in qualsiasi momento, potrebbero avvenire molte richieste contemporaneamente, in modo imprevedibile.

La scalabilità è definita in letteratura in modi diversi. Secondo [5] una definizione generalizzata è la seguente: La scalabilità di un servizio è una proprietà che fornisce una stima della capacità di gestire crescenti quantità di carichi di servizio senza subire degrado della qualità. Inoltre, la scalabilità ottenuta come risultato dell'applicazione di modelli o schemi (come modelli che adeguano le risorse in modo appropriato) deve essere proporzionale al costo per applicare il modello.

In questa definizione, troviamo quattro elementi:

- gestione dei crescenti carichi di lavoro,
- garantire la scalabilità attraverso modelli,
- costi accettabili,
- evitare la degradazione significativa della QoS.

I consumatori di servizi non sono generalmente noti in anticipo. Non è cioè possibile prevedere in maniera statistica il numero di consumatori di servizio e la quantità di invocazioni dei servizi.

In presenza di grandi crescite di richieste di un servizio, la QoS potrebbe degradare. Quindi, la scalabilità deve risolvere i problemi della gestione dei cambiamenti improvvisi dei carichi sui servizi.

Essa può essere assicurata mediante l'adozione di appositi modelli e lo schema più comune è quello di aggiungere risorse al variare della domanda. Questo ovviamente implica un costo di gestione (per esempio aggiunta di CPU, memoria, server, rete, etc.).

Poiché i servizi debbono fornire un certo livello di qualità, come specificato nel loro SLA, quelli che possiedono buona scalabilità riusciranno dunque a rispettare i minimi requisiti di QoS.

4. La qualità del software

Alcuni articoli e testi che trattano il tema del controllo del software sono: [6] [7] [8].

Lo standard, proposto da IEEE e recepito concordemente nella quasi totalità dei testi di ingegneria del software, oltre a precisare il significato dei termini, stabilisce fra essi un ordinamento basato sul rapporto di causa-effetto. Come tutti i software anche in questo caso siamo di fronte alla possibilità che ci possano essere errori. Un errore ha, nella maggior parte dei casi, radici umane: una distrazione in fase di codifica può introdurre un difetto nel programma che genera malfunzionamenti.

In alcuni casi, l'errore può derivare dagli strumenti di sviluppo o nel modo con cui essi sono utilizzati. Ad esempio, un compilatore difettoso o usato maldestramente può generare un codice malfunzionante pur partendo da un sorgente corretto.

4.1 Controlli interni ed esterni

Con l'attributo interno si identifica ogni attività di controllo che proviene dalla stessa organizzazione impegnata nel processo di sviluppo software. Chi è incaricato di eseguire i controlli appartiene cioè alla stessa azienda, o divisione, o squadra che lavora a un progetto software.

L'appartenenza al progetto di tale personale, rende i controlli interni estremamente mirati, con obiettivi precisi rispetto alle caratteristiche del prodotto in via di sviluppo e condotti a un livello di dettaglio molto fine. Caratteristica del tutto naturale, data la provenienza del personale incaricato e delle condizioni privilegiate in cui si trova ad agire: esso ha sicuramente a disposizione tutte le informazioni possibili, è investito dell'autorità necessaria per chiedere delucidazioni al personale più direttamente coinvolto nello sviluppo, ha, poiché spesso coinvolto nel progetto fin dal suo inizio, la competenza per dare la migliore direzione alla strategia dei controlli e, nei casi più fortunati, per suggerire soluzioni ai problemi che l'attività di controllo ha messo in evidenza.

Un'attività di controllo è invece detta esterna quando è condotta da personale estraneo all'organizzazione che ha in carico lo sviluppo di un prodotto software. I controlli esterni sono tipici delle situazioni in cui il prodotto è realizzato su commissione ed è evidente la contrapposizione delle due parti contrattuali del committente e del fornitore.

In questi casi i controlli esterni si sovrappongono ai controlli interni effettuati per conto proprio dal fornitore e hanno l'obiettivo di costituire, a vantaggio del committente, un'ulteriore garanzia circa la qualità del prodotto. E' nell'interesse del committente che sia lui stesso, o, come più spesso accade, una terza parte che agisce per suo conto, ad avere la responsabilità dell'organizzazione e della conduzione dei controlli.

D'altra parte il fornitore "subisce" l'attività di controllo ed è tenuto, nei limiti stabiliti dal contratto, a fornire tutto il supporto necessario affinché il personale esterno incaricato dei controlli sia in grado di eseguirli materialmente.

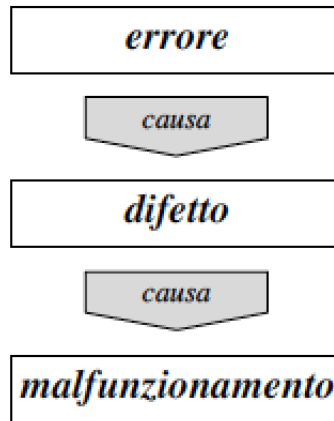


Figura 8: Errori, difetti e malfunzionamenti

Il limite dei controlli esterni sta appunto nelle ristrette possibilità di indagine del personale addetto alle operazioni di controllo: il fornitore, in sede di contratto, ha interesse a limitare il più possibile la visibilità del processo di sviluppo, anche per un suo diritto a mantenere riservate le tecnologie adottate e le proprie politiche organizzative. Per questo motivo, i controlli esterni assumono in generale il connotato di validazioni, essendo i requisiti il più comune elemento di riscontro in possesso del committente.

Un tipico controllo esterno è il collaudo.

Una terminologia molto diffusa divide i controlli finali in a-test e b-test. Fatte salve le considerazioni circa l'uso del termine test, i due termini distinguono i controlli rispetto al soggetto che li esegue.

Quando chi esegue i controlli appartiene alla stessa organizzazione che ha sviluppato il software si parla di a-test, altrimenti si deve parlare di b-test.

I controlli di sistema possono essere svolti sia come a-test che come b-test: nel secondo caso il prodotto, prima di essere rilasciato definitivamente, viene fatto circolare fra un insieme ristretto di gruppi di prova, "vicini" al fornitore, che si limitano a usare normalmente il software segnalando eventuali malfunzionamenti.

I controlli di accettazione, poiché eseguiti dal committente, rientrano nella definizione di b-test, ma non è del tutto corretto considerare i termini "controlli di accettazione" e "b-test" sinonimi.

A conferma della moderna tendenza verso cicli di vita evolutivi è da evidenziare l'uso intensivo del b-test come metodo di valutazione dei requisiti di un prodotto. Nel caso di software commerciale, è molto comune la distribuzione di versioni incomplete del software, dette b-release, al preciso scopo di valutarne il gradimento rispetto alle esigenze di un'utenza che, altrimenti, non sarebbe direttamente raggiungibile.

Per accrescere la base di valutazione, è normale distribuire le b-release gratuitamente a qualsiasi utente ne faccia richiesta. In questo caso, il mercato del prodotto finale è protetto introducendo nelle b-release dei meccanismi che ne consentono l'uso solo per un limitato periodo di tempo.

4.2 Verifica e Validazione

Con i termini di verifica e validazione si distinguono le attività di controllo in base alla loro portata rispetto ai diversi tempi del processo di sviluppo, individuando con il termine verifica le attività di controllo che sono caratterizzate dall'essere circoscritte a una sola fase.

Verifica e validazione rispondono rispettivamente alle domande:

- il prodotto viene realizzato correttamente?
- viene realizzato il prodotto corretto?

Obiettivo della verifica è il controllo di qualità delle attività svolte durante la fase dello sviluppo; obiettivo della validazione è il controllo di qualità del prodotto rispetto ai requisiti del committente.

Come illustrato in figura 9, è possibile interpretare il processo di sviluppo come una successione di lavorazioni di prodotti intermedi. In questa prospettiva, un'attività di verifica è il controllo che il prodotto ottenuto al termine di una fase sia congruente con il semilavorato avuto come punto di partenza di quella fase.

Per esempio, nella realizzazione di un modulo, è una tipica verifica il controllo che le specifiche del modulo siano state rispettate sia come interfaccia che come funzionalità.

La validazione è un'attività di controllo mirata a confrontare il risultato di una fase del processo di sviluppo con i requisiti del prodotto; tipicamente con quanto stabilito dal contratto o, meglio, dal documento di analisi dei requisiti. Un comune esempio di validazione è il controllo che il prodotto finito abbia funzionalità e prestazioni conformi con quelle stabilite all'inizio del processo di sviluppo.

La validazione è un'attività normalmente prevista sul prodotto finito. Tuttavia, limitandosi ad aspetti particolari, è comunque possibile effettuare delle operazioni di validazione anche durante il processo di sviluppo. Ad esempio, l'architettura può essere validata con i requisiti; in questo caso la validazione è una preventiva assicurazione contro possibili errori di interpretazione dei requisiti.

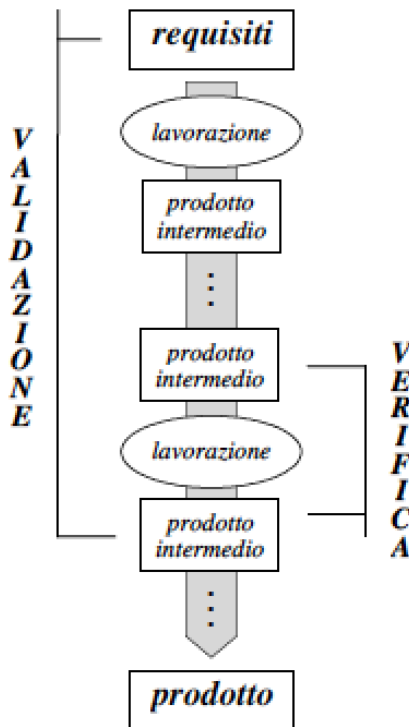


Figura 9: Verifica e validazione rispetto al ciclo di vita

Verifica e validazione sono attività che si sovrappongono a quelle tradizionali e concretamente produttive del processo di sviluppo e con le quali devono integrarsi nel più efficace possibile dei modi. La scoperta di un errore durante lo sviluppo è sicuramente un guadagno per la qualità del prodotto, ma è anche un evento la cui gestione va prevista.

L'organizzazione del processo di sviluppo deve prevedere e pianificare ogni attività di verifica e validazione, mediando fra la parallelizzazione e la serializzazione delle attività di sviluppo e di quelle di controllo: nel primo caso si tende a minimizzare i ritardi, nel secondo si cerca di risparmiare sulle risorse e di evitare gli sprechi.

4.2.1 *Controlli statici*

Il controllo statico di un programma, spesso chiamato anche verifica statica o analisi, è basato sulla non esecuzione del programma. Diversi sono i motivi per rinunciare alla naturale tendenza a lanciare un programma per sapere se esso funziona correttamente. Un primo motivo è che eseguire un programma, specialmente se questo è in realtà una procedura, cioè un pezzo di codice scollegato da ogni contesto eseguibile, ha un costo che, in molti casi può far preferire controlli basati sulla sola analisi del codice.

4.2.2 *Controlli dinamici*

Il controllo dinamico, o test, richiede l'esecuzione di un programma in un ambiente e con dati di ingresso controllati, la registrazione di tutti i dati riguardanti l'esecuzione e interessanti ai fini dei fattori di qualità che si vogliono valutare e, infine, l'analisi dei dati e il loro confronto con i requisiti.

Tecnicamente, il test consiste nell'esercitare un programma al fine di scoprire malfunzionamenti che denunciino l'esistenza di difetti.

Il termine test, assai usato anche in italiano, indica comunemente sia il controllo dinamico in generale, inteso come attività del processo di sviluppo e disciplina dell'ingegneria del software, che una singola sessione di prova di un programma identificata dall'insieme di condizioni (come dati di ingresso e contesto di esecuzione), in cui essa si svolge.

Il test come tecnica di controllo ha come principale obiettivo la verifica della correttezza funzionale di un programma o di un sistema, ma sempre più spesso viene usato come tecnica di base per la realizzazione di controlli mirati alla valutazione di altri fattori di qualità come ad esempio affidabilità, usabilità ed efficienza.

Il test è il controllo di qualità del software di gran lunga più praticato, anche se spesso, proprio per la sua concettuale semplicità (funziona?). Questa è un'attività condotta senza la necessaria pianificazione e di conseguenza, con scarsi risultati sul piano del raggiungimento di una migliore qualità del prodotto.

4.2.3 *Progettazione dei test*

Un test non si improvvisa, va studiato e preparato.

Obiettivo del test è eseguire un programma per provocare malfunzionamenti: per essere efficace un test deve essere realizzato in base alle caratteristiche del programma da esaminare. La progettazione è perciò mirata a definire il contesto di esecuzione, in particolar modo selezionando l'insieme dei dati di ingresso da fornire al programma durante il test.

4.2.3.1 *Ambiente di test*

Il test deve avvenire in un ambiente che permetta l'esecuzione controllata dei programmi in esame. Sono necessari strumenti per fornire nella giusta sequenza i dati di ingresso selezionati per il test e per registrare

tutti i dati del comportamento del programma utili per la successiva analisi; l'ambiente di esecuzione deve essere configurabile e controllabile in tutti i suoi aspetti (memoria disponibile, potenza di calcolo, esecuzione concorrente di altri programmi, etc.).

Se il software sottoposto a test non è un sistema compiuto, ma un suo componente o una sua parte, il test comporta anche la realizzazione di tutti i programmi di contorno necessari a simulare il comportamento delle parti mancanti del sistema.

4.2.3.2 *Analisi dei risultati*

I dati ottenuti dall'esecuzione di un test devono essere analizzati alla ricerca di eventuali malfunzionamenti. Sono perciò necessari dei dati di riscontro, che identifichino il risultato atteso, con cui confrontare i risultati di test.

4.2.3.3 *Debugging*

A differenza delle tecniche di controllo statico che, per definizione, mirano direttamente ai difetti e che spesso nel trovarli danno anche buone indicazioni sugli errori che li hanno generati, il test ha come obiettivo la scoperta di malfunzionamenti.

A un test positivo deve quindi seguire la necessaria attività per l'eliminazione dei difetti (detta debugging dall'inglese bug), vero fine di ogni controllo volto al miglioramento della qualità.

Il controllo dinamico si mostra quindi come un'attività particolarmente onerosa all'interno del processo di sviluppo, ma necessaria. I controlli statici, infatti, hanno grande importanza nel rivelare alcune categorie di difetti e sono un utile complemento al test nella ricerca dei difetti a fronte del manifestarsi di un funzionamento, ma non possono applicarsi in modo risolutivo a tutti i casi che si presentano nel corso della realizzazione di un prodotto software.

In particolare, i controlli statici, per le limitate capacità delle persone e degli strumenti, si possono applicare solo a porzioni di codice di dimensione ridotta (tipicamente i moduli), lasciando così scoperta la prova di un sistema nel suo insieme, per la quale il test rimane la forma di controllo più indicata. Ci sono inoltre requisiti, come l'efficienza, che possono essere controllati solo a fronte della messa in esecuzione del software.

4.3 *Livelli di test*

I prodotti software, anche quando si presentano all'utente finale come un tutto unico, sono in realtà dei sistemi complessi costituiti da più componenti integrati e cooperanti.

Le tecniche di testing si adeguano e, insieme, traggono vantaggio da questa organizzazione comune a tutti i sistemi software. Esistendo già una decomposizione in elementi più semplici, risulta naturale applicare il testing già a partire dai singoli componenti e quindi arrivare al test dell'intero sistema seguendo strategie di integrazione dettate dall'architettura.

4.3.1 *Test sul sistema*

Il test di sistema è la più canonica delle attività di validazione che valuta ogni caratteristica di qualità del prodotto software nella sua completezza, avendo come documento di riscontro i requisiti dell'utente.

Le tecniche più adottate per i test sul sistema sono basate su criteri funzionali. Gli obiettivi dei controlli di sistema sono normalmente mirati a ben determinati aspetti:

- Facility test (test delle funzionalità). È il più intuitivo dei controlli, quello cioè che mira a controllare che ogni funzionalità del prodotto stabilita nei requisiti sia stata realizzata correttamente.
- Security test. Cercando di accedere a dati o a funzionalità che dovrebbero essere riservate, si controlla l'efficacia dei meccanismi di sicurezza del sistema.
- Usability test. Con questo controllo si vuole valutare la facilità d'uso del prodotto da parte dell'utente finale. È una valutazione su una delle caratteristiche di un prodotto software fra le più soggettive; il controllo deve prendere in esame oltre al prodotto anche tutta la documentazione che lo accompagna e deve tener conto del livello di competenza dell'utenza e delle caratteristiche operative dell'ambiente d'uso del prodotto.
- Performance test. È un controllo mirato a valutare l'efficienza di un sistema soprattutto rispetto ai tempi di elaborazione e ai tempi di risposta. È un tipo di controllo critico per quelle categorie di prodotti, come ad esempio i sistemi in tempo reale, per le quali ai requisiti funzionali si aggiungono rigorosi vincoli temporali.
- Storage use test. È ancora un controllo legato all'efficienza di un sistema, ma mirato alla richiesta di risorse (in particolare la memoria) durante il funzionamento e ha implicazioni sull'ambiente operativo richiesto per l'installazione del sistema.
- Volume test (o load test, test di carico). Durante questo tipo di controllo il sistema è sottoposto al carico di lavoro massimo previsto dai requisiti e le sue funzionalità sono controllate in queste condizioni. Lo scopo è sia individuare malfunzionamenti che non si presentano in condizioni normali, quali difetti nella gestione della memoria, buffer overflows, etc., sia garantire un'efficienza base anche in condizioni di massimo carico.
- Le tecniche e gli strumenti del volume test sono di fatto usati anche per il performance test: vengono fissati alcuni livelli di carico, e su questi sono valutate le prestazioni del sistema. Però, i due tipi di test hanno scopi molto differenti, da un lato valutare le prestazioni a vari livelli di carico, non limite, dall'altro valutare il comportamento del sistema sui valori limite.
- Stress test. Il sistema è sottoposto a carichi di lavoro superiori a quelli previsti dai requisiti o è portato in condizioni operative eccezionali, in genere sottraendogli risorse di memoria e di calcolo. Non è da confondere con il volume test da cui differisce per l'esplicito superamento dei limiti operativi previsti dai requisiti. Lo scopo è quello di controllare la capacità di "recovery" (recupero) del sistema dopo un fallimento.
- Configuration test. Alcuni prodotti prevedono la possibilità di avere più configurazioni, per lo più in presenza di piattaforme di installazione diverse per sistema operativo o dispositivi hardware installati, in altri casi per soddisfare insieme di requisiti funzionali leggermente diversi. Questo tipo di controllo ha per obiettivo la prova del sistema in tutte le configurazioni previste.
- Compatibility test. È un controllo che ha l'obiettivo di valutare la compatibilità del sistema con altri prodotti software. Gli oggetti del confronto possono essere versioni precedenti dello stesso prodotto, sistemi diversi, ma funzionalmente equivalenti che il prodotto deve rimpiazzare, oppure altri sistemi software con cui il prodotto deve interagire nel suo ambiente operativo finale.

È interessante notare come i test descritti mirino a esercitare gli aspetti di un sistema software che corrispondano a quelle caratteristiche che sono normalmente percepite come fattori di qualità.

Troviamo ad esempio che il facility test e il security test sono direttamente legati con la funzionalità, volume test e stress test sono in rapporto con l'affidabilità, ovvia è la collocazione dell'usability test, performance test

e storage use test rispecchiano i due aspetti dell'efficienza, infine, configuration test e compability test sono collegabili alla portabilità del sistema.

5. Il monitoraggio delle prestazioni dell'applicazione per la valutazione delle performance delle funzionalità di condivisione dell'informazione

Al fine di effettuare le operazioni di monitoraggio delle prestazioni del sistema SHELL, nella fattispecie della condivisione dell'informazione del nodo "casa", occorre selezionare uno strumento idoneo a ottenere risultati il più possibile soddisfacenti a permettere una adeguata valutazione delle performance.

L'analisi di tale applicativo deve tener conto delle osservazioni esposte nei paragrafi precedenti e, in base ad esse, deve rispettare opportuni requisiti.

5.1 Requisiti principali

5.1.1 Requisiti hardware

I più importanti sono:

- la flessibilità: i componenti del sistema debbono potersi facilmente adattare alle molteplici necessità delle situazioni e alle condizioni ambientali in cui operano; nel sistema debbono poter essere integrate eventuali strumentazioni e reti già esistenti e comunque informazioni e dati provenienti da altre fonti;
- l'espandibilità: il sistema deve poter essere facilmente ampliato con componenti aggiuntive, oppure ridotto e riposizionato in relazione all'evoluzione dei fenomeni e dalle mutate necessità conoscitive;
- l'affidabilità: deve essere garantita la funzionalità dei componenti per lunghi periodi e in condizioni ambientali anche critiche, durante le quali spesso sono richieste le maggiori prestazioni;
- la disponibilità: l'applicativo deve disporre di risorse interne (ridondanza dei componenti critici, capacità di funzionamento indipendente, ecc.) per garantire adeguati livelli di funzionamento anche a fronte di eventi imprevisti, situazioni di crisi, ecc.;
- la manutenibilità: gli interventi di manutenzione, devono poter essere eseguiti in modo semplice grazie alla standardizzazione dei componenti e a una progettazione che tenga conto delle eventuali necessità.

5.1.2 Requisiti software

Le caratteristiche più importanti delle componenti strumentali software del sistema riguardano l'interfaccia uomo/macchina e le funzionalità di elaborazione. L'interfaccia generale del sistema deve consentire la massima semplicità e flessibilità di utilizzo dei diversi componenti funzionali, i cui requisiti principali sono:

1. la semplicità delle istruzioni;
2. la disponibilità di funzioni di aiuto (*help*) in linea per l'utente;
3. l'accessibilità agevole alle diverse possibilità offerte dal sistema;
4. il mantenimento di una traccia di tutto quanto avviene in una sessione di lavoro (storia dell'elaborazione).

5.2 Funzionalità di monitoraggio delle prestazioni

Il Monitoraggio delle prestazioni dell'applicazione (APM) è una sezione della tecnologia informatica (IT) che si occupa di garantire che i programmi di applicazione del codice del computer funzionino come previsto o meno, mentre gli strumenti di monitoraggio dell'applicazione sono dedicati all'esecuzione di tale monitoraggio o tracciamento. L'obiettivo finale del monitoraggio delle prestazioni è fornire agli utenti finali un'esperienza utente di alta qualità.

Tali strumenti dovranno fornire le informazioni per scoprire rapidamente e capire i problemi che incidono negativamente sulle prestazioni di un'applicazione. Tali strumenti possono essere specifici per un'applicazione selezionata o monitorare più applicazioni su una rete costante, raggruppando in generale i dati relativi all'utilizzo della CPU client, alle esigenze di memoria, all'output dei dati e alla larghezza di banda complessiva.

La gestione delle prestazioni delle applicazioni è fondamentalmente un termine utilizzato per tutto ciò che ha a che fare con la gestione o il monitoraggio delle prestazioni del codice, delle dipendenze delle applicazioni, dei tempi di negoziazione e dell'esperienza dell'utente nel suo complesso.

5.2.1 *Tipologie di strumenti di monitoraggio delle prestazioni delle applicazioni*

1. Basate su metriche di applicazione: molti strumenti utilizzano numerose metriche di server chiamate application performance management (APM) [9]. Esse servono per dare un'indicazione del tipo e del numero delle richieste che riceve l'applicazione, per aiutare a tenere sotto controllo le prestazioni del sistema, non eseguono però la profilazione a livello di codice.
2. Basate sulla profilazione del codice e sul tracciamento delle transazioni: sono sostanzialmente i prodotti per la gestione delle prestazioni delle applicazioni.
3. Basate su rete: sono gli strumenti di misura delle prestazioni delle applicazioni in base al traffico di rete.

5.2.1.1 *Utilizzo e prestazioni delle dipendenze delle applicazioni come database, servizi di rete, cache e altro*

Il motivo per cui le prestazioni di un'applicazione risultano degradate è, in genere, un picco del traffico o un malfunzionamento con una delle dipendenze dell'applicazione.

5.2.1.2 *Monitoraggio e metriche di base del server come CPU o memoria*

Ai fini dell'individuazione di eventuali malfunzionamenti che incidono sulle prestazioni dell'applicazione è importante monitorare la CPU del server e la memoria. Molte delle recenti applicazioni di rete a volte non sono vincolate alla CPU, tuttavia useranno ancora molta CPU ed è un indicatore utile per il ridimensionamento automatico dell'applicazione nel cloud.

6. Gli strumenti per il monitoraggio delle prestazioni dell'applicazione.

Le attività di ricerca e analisi eseguiti nell'OR, hanno permesso di individuare due applicativi che sono stati ritenuti idonei al caso specifico. Essi sono *Zabbix* e *Xymon*.

6.1 Zabbix

Zabbix è il sistema di monitoraggio open source (licenza GPL 2) della lituana *Zabbix SIA*, capace di monitorare contemporaneamente migliaia di dispositivi connessi in rete. Com'è possibile leggere dal sito ufficiale, le caratteristiche principali di *Zabbix* sono:

- la possibilità di accedere ai sorgenti del sistema di monitoraggio, essendo esso stesso open source e potendosi basare su piattaforme open source (Linux, Apache, MySQL/PostgreSQL, Php);
- la semplicità di configurazione;
- l'efficienza delle componenti software;
- un sistema di visualizzazione delle informazioni completo;
- un set di procedure di ottimizzazione dell'organizzazione dei dati raccolti.

Altri due punti di forza sono la possibilità di estensione delle componenti software e la possibilità di effettuare un monitoraggio distribuito, grazie all'architettura software dell'intero sistema. Attualmente è disponibile la versione 5.0.

Grazie al monitoraggio di tipo real-time, si possono monitorare decine di migliaia di server, macchine virtuali e dispositivi della rete simultaneamente.

Oltre alla gestione dei dati, sono disponibili anche caratteristiche di visualizzazione (overview, grafi, mappe, screen, ecc...) così come modalità di analisi dei dati flessibili per la gestione dei malfunzionamenti ed operazioni di avviso.

Zabbix effettua un monitoraggio di tipo distribuito grazie all'utilizzo di *Zabbix proxy*; esso inoltre è dotato di un'interfaccia web-based, un'autenticazione sicura e uno schema di permessi utente flessibile.

Sono supportati sia il polling che il trapping e c'è la possibilità di raccogliere informazioni relative ad un dispositivo (cpu, memoria e altro) grazie alla presenza di appositi agenti, ma è possibile anche operare anche in modalità agentless (ovvero senza agenti, ma con l'utilizzo di script esterni).

Sono supportati sia il monitoraggio Web che il monitoraggio di macchine virtuali VMware.

Zabbix, inoltre, può scoprire automaticamente server e dispositivi di rete ed seguire il rilevamento di basso livello con metodi di controllo di disponibilità e prestazioni dei componenti individuati.

Esso è nato come sistema di monitoraggio distribuito della rete con un'interfaccia web centrale attraverso la quale è possibile gestire l'intero monitoraggio.

A partire dalla versione 2.4, il numero di possibili architetture si è ridotto all'installazione di un solo server e di *Zabbix proxy* distribuiti.

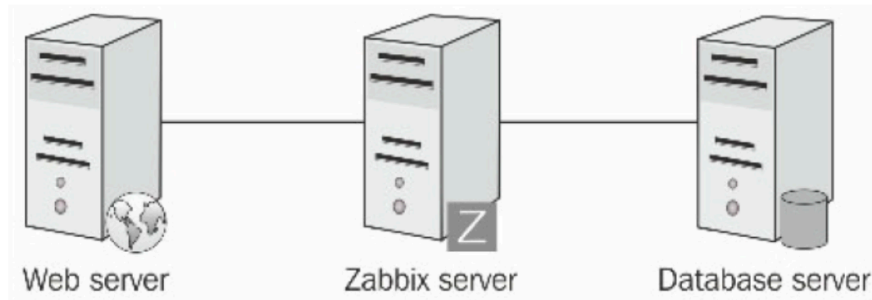


Figura 10: Architettura Zabbix

Le componenti principali di *Zabbix* sono le seguenti:

- **Zabbix Agent:** agente software che si occupa di raccogliere tutti i dati del dispositivo da monitorare (target); questo modulo va installato sul target e, una volta configurato e lanciato, aprirà una porta TCP (10050 di default) utilizzata dallo *Zabbix Server* per lo scambio delle misure sulle grandezze monitorate. *Zabbix Agent* può essere installato su molti sistemi operativi (tra cui Linux, Windows, MacOS, FreeBSD, OpenBSD e Solaris). In un sistema di monitoraggio *Zabbix*, l'uso dell'agente per collezionare dati è solitamente la via classica e maggiormente indicata, ma è possibile utilizzare anche altre interfacce di comunicazione. In caso di fallimento, come nel caso di hard disk pieno o crash di un processo di servizio, il server può avvisare gli amministratori della macchina che ha riscontrato tale problema. Gli agenti effettuano dunque chiamate native di sistema per raccogliere le informazioni desiderate e ognuno di essi può eseguire controlli passivi o attivi (passive or active check):
 - in un *passive check* (in quanto l'agente si mette in attesa di connessioni da parte del server, il quale è parte attiva della comunicazione), *Zabbix agent* risponde ad una richiesta di dati, ovvero, il server chiede un determinato controllo (ad esempio il carico della CPU) e l'agente gli risponde fornendo il risultato;
 - gli *active check* (in cui è l'agente che, periodicamente, invia le informazioni al server opportunamente configurato) richiedono un'elaborazione più complessa; infatti, l'agente deve prima recuperare una lista di item presenti sul *Zabbix server* per poterli poi successivamente processare in modo indipendente. Gli *active check* che un server può ottenere sono elencati nel parametro *ServerActive* del file di configurazione dell'agente. La frequenza di richiesta di tali controlli, viene definita nel parametro *RefreshActiveChecks* presente nello stesso file di configurazione. Infine, l'agente invia periodicamente i nuovi valori al server.
- **Zabbix Server:** modulo software installato solitamente (ma non necessariamente) su una macchina differente rispetto al target, e che si preoccupa di interrogare tutti i dispositivi configurati per il monitoraggio. Lo *Zabbix Server* può essere installato esclusivamente su un sistema Linux. Essendo possibile monitorare migliaia di dispositivi per volta, è solitamente necessario che l'hardware su cui viene eseguito lo *Zabbix Server* abbia capacità di elaborazione adeguata. Una volta raccolti i dati dai target, lo *Zabbix Server* si preoccupa di memorizzarli in un database opportunamente configurato, in modo da renderli accessibili ad una qualunque entità che si occupi della loro elaborazione/organizzazione/visualizzazione, così da essere resi fruibili da parte degli utenti preposti al

monitoraggio. Ad esempio, quando viene creato un item utilizzando l'interfaccia Web, questo viene aggiunto nella tabella degli item presente nel database. Ogni minuto circa, Zabbix server interroga tale tabella per ottenere una lista degli item attivi e li memorizza all'interno di una sua cache.

Zabbix server è il processo centrale del software Zabbix. Esegue il polling e il trapping dei dati, calcola i trigger ed invia le notifiche agli utenti. E' il componente a cui Zabbix agent e proxy inviano dati sulla disponibilità e integrità del sistema; esso inoltre, può controllare a distanza servizi di rete (come web server o mail server) utilizzando semplici check di servizio.

Nelle Figure 11 e 12, si vede che vengono utilizzati dei server dedicati con lo scopo di consentire il funzionamento del sistema di monitoraggio anche in ambienti molto grandi.

- **Zabbix Proxy:** modulo software opzionale, il cui compito è quello di collezionare dati dagli agent configurati ed inviarli allo *Zabbix Server*; solitamente il proxy si utilizza per due motivi:
 - 1 - organizzare meglio a livello logico il monitoraggio, facendo diminuire il carico computazionale del server; immaginiamo il caso in cui ci siano migliaia di dispositivi da monitorare e fosse possibile raggruppare logicamente tali dispositivi: il server interrogherebbe solo i proxies, diminuendo drasticamente il numero di connessioni necessarie (Figura 11).

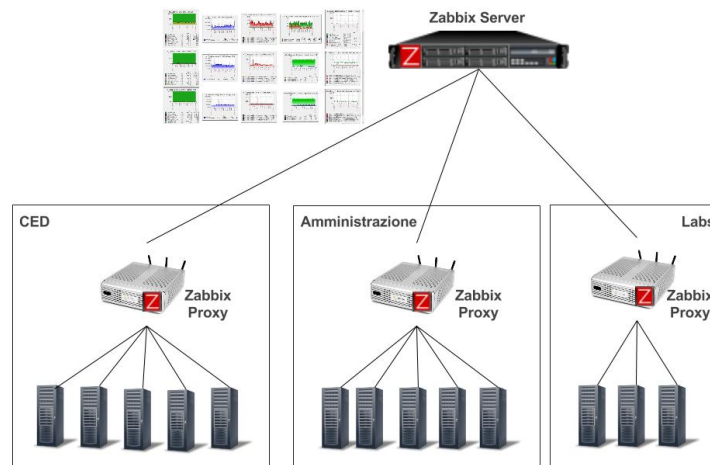


Figura 11: Scenario semplice con Zabbix Proxy

2 - rendere semplice la configurazione di rete in caso di monitoraggio di dispositivi protetti da firewall (Figure 12 e 13). In tal caso, grazie al proxy si rende semplice la comunicazione tra server e dispositivi monitorati. Occorre comunque dire che l'utilizzo di proxies si rende solitamente necessario solo per scenari complessi.

Zabbix proxy raccoglie i dati provenienti da un certo numero di host o dispositivi, ne acquisisce le metriche e agisce come un generico proxy. Esso ha la possibilità di mantenere memorizzati i dati su un database dedicato. *Zabbix proxy* non possiede un frontend, pertanto deve essere gestito direttamente dal server centrale (*Zabbix server*).

Zabbix proxy si limita a raccogliere i dati senza eseguire valutazioni o azioni di attivazione; per questo motivo, è utile utilizzare un server RDBMS efficiente e robusto al fine di prevenire la perdita di dati in caso di crash.

Poiché l'obiettivo è il controllo e l'ottimizzazione del flusso dei dati monitorati attraverso le reti, tale proxy consente di dividere ed isolare gli elementi e i dati su reti differenti. Le metriche acquisite vengono memorizzate nel database e in caso di interruzione accidentale, esse non vengono perse.

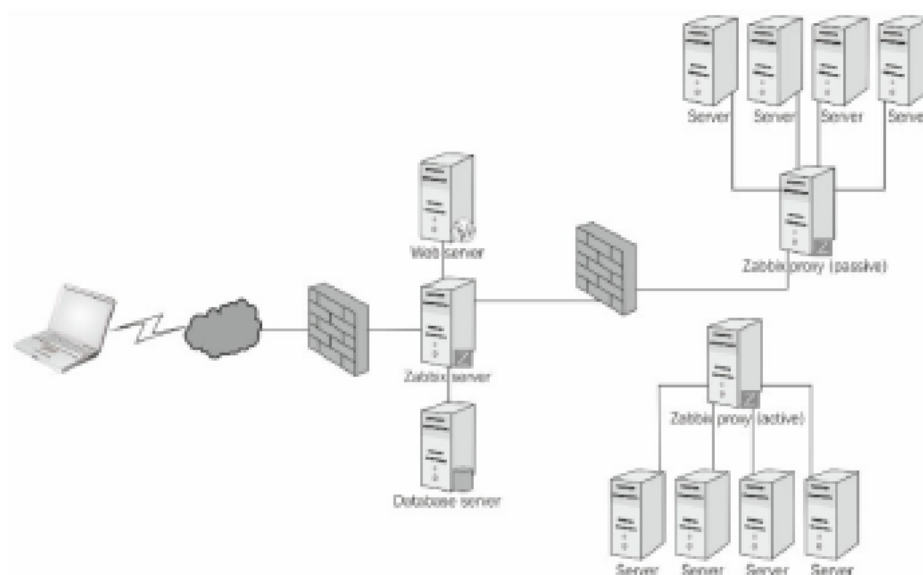


Figura 12: Estensione dell'architettura Zabbix con proxy

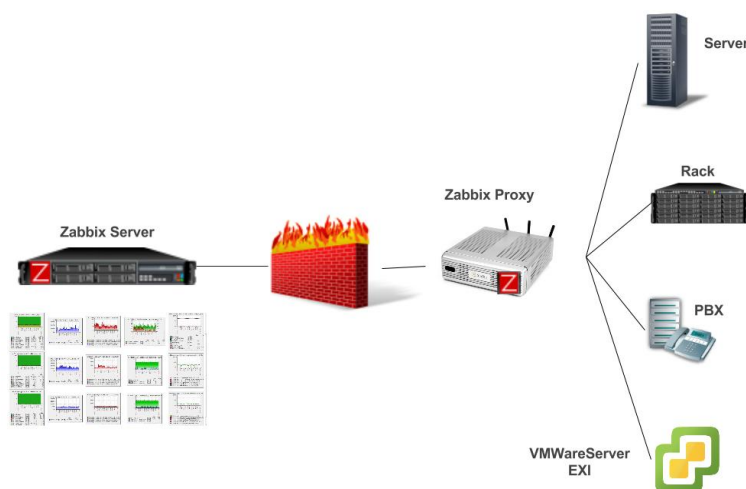


Figura 13: Scenario con Zabbix Proxy e Firewall

- **Zabbix Frontend:** GUI di configurazione e monitoraggio dell'intero sistema; è il modulo software con cui gli utenti si interfacciano verso il sistema di monitoraggio, dal quale è possibile rendersi conto dello stato del sistema. Al contrario degli altri componenti (scritti in linguaggio C), il frontend è scritto in Php.

L'accesso al sistema è basato su gruppi di utenti, i quali hanno dei privilegi prestabiliti (monitoraggio o configurazione del sistema, con granularità a livello di singolo dispositivo). In questo modo è possibile

configurare un gruppo di amministratori, i quali hanno pieno controllo sul sistema, e una serie di utenti con privilegi ristretti, che possono solo monitorare una serie di dispositivi prestabiliti.

6.1.1 *Caratteristiche*

In questo paragrafo, vengono esaminate le caratteristiche principali di Zabbix.

6.1.2 *Monitoraggio*

All'interno della rete, è possibile monitorare molti parametri: performance e disponibilità dei server, applicazioni Web, database, dispositivi di rete, applicazioni e molto altro.

Utilizzando *Zabbix*, è possibile raccogliere i dati relativi ai componenti monitorati e ricavare delle statistiche accurate.

Il monitoraggio di indicatori di performance come CPU, memoria, rete, spazio del disco e processi possono essere monitorati con l'utilizzo degli agenti che sono disponibili in sistemi UNIX, Linux e Windows.

Tali agenti sono processi nativi, pertanto non richiedono uno specifico ambiente di sviluppo come Java o .NET. *Zabbix* supporta gli SNMP agent presenti in tutti i dispositivi di rete come router o switch; e li utilizza per monitorare tali dispositivi.

Attraverso i gateway di Java, vengono monitorati i Java Application Server (JBoss, Tomcat, Oracle Application Server e altro) direttamente su JMX senza installazione di software di terze parti o integrazione di più livelli.

Per mezzo degli IPMI agent è possibile inoltre monitorare i parametri tipici del livello hardware come temperatura del dispositivo, voltaggio, la velocità delle ventole e stato del disco in modo da evitare i downtime. Quando non c'è la possibilità di installare agenti sul dispositivo da monitorare, si può ricorrere al monitoraggio in modalità agentless.

Zabbix può usare anche regole di discovery di basso livello per individuare automaticamente macchine virtuali e VMware hypervisor da monitorare e creare degli host basati su prototipi predefiniti.

L'applicativo permette di controllare, in modo dettagliato, qualsiasi database come MySQL, PostgreSQL, Oracle e Microsoft SQL Server. Spesso però, la distribuzione delle informazioni avviene attraverso siti web o sistemi IT web-based e *Zabbix* fornisce una soluzione built-in anche per il monitoraggio web. E' possibile infatti definire dei metodi che permettono di analizzare un determinato sito web, controllare la sua disponibilità e la velocità di download.

Infine, le possibilità di estensione e personalizzazione di *Zabbix* fanno sì che tale sistema possa essere integrato in molti ambienti, senza limitazioni dovute ai linguaggi di programmazione (SHELL, Perl, Python o altri linguaggi).

6.1.3 *Enterprise Ready*

Zabbix può scalare da ambienti piccoli con decine di dispositivi a quelli molto più grandi con centinaia di migliaia di componenti monitorati. Il sistema è in grado di monitorare 100 mila dispositivi, processando più di 3 milioni di check al minuto. Gli agenti utilizzano poche risorse di CPU e memoria e *Zabbix* server e proxy utilizzano varie soluzioni di data caching, riducendo il carico sul database.

Per quanto riguarda l'aspetto sicurezza, l'accesso al frontend può essere fatto attraverso una connessione protetta SSL, garantendo la sicurezza fra utenti e server. Il frontend inoltre possiede algoritmi di auto-protezione contro eventuali attacchi di forza bruta.

I componenti accettano solo connessioni provenienti da indirizzi IP autorizzati, tutte le altre richieste vengono rifiutate. E' inoltre possibile rendere sicure le comunicazioni utilizzando un protocollo TLS (Transport Layer Security).

6.1.4 *Monitoraggio pro-attivo*

Con lo scopo di ridurre i costi operativi, evitare i downtime e migliorare la qualità di servizio, *Zabbix* può inviare messaggi di notifica attraverso email, SMS, Jabber o altro metodo user-defined, inclusa la creazione di tickets automatici in Service Desk o Service Catalog System.

Esso è in grado di effettuare automaticamente semplici azioni come il riavvio di un servizio o l'indirizzamento verso un server alternativo. Inoltre, se una prima notifica o l'esecuzione di un task automatico non sono sufficienti per risolvere un problema, è previsto l'invio di notifiche a tecnici e amministratori di sistema.

6.1.5 *Pianificazione delle capacità*

Zabbix fornisce dati per effettuare analisi statistiche al fine della pianificazione; per esempio esaminando la crescita di utilizzo dello spazio disco è possibile stimare se lo spazio va esaurendosi.

E' possibile rilevare lo spreco di CPU, memoria, disco o larghezza di banda su un singolo dispositivo o su un intero gruppo di server; in questo modo, sarà possibile riallocare le applicazioni e le apparecchiature per utilizzare al meglio le risorse disponibili.

6.1.6 *Open Source*

Zabbix è rilasciato sotto la licenza GPL, quindi è gratuito sia per uso commerciale che non. Non ci sono limitazioni sul numero di dispositivi che *Zabbix* può monitorare. È possibile modificare il codice sorgente, nonché sviluppare strumenti personalizzati da integrare con *Zabbix* stesso. Tutte le impostazioni ed i valori raccolti vengono memorizzati in un formato aperto.

6.1.7 *Raccolta dei dati*

Zabbix raccoglie informazioni relative alla disponibilità e alle performance delle infrastrutture monitorate.

La sorgente di dati migliore è rappresentata dall'utilizzo di *Zabbix* agent. Essi sono processi nativi sviluppati in linguaggio C che possono essere eseguiti su varie piattaforme (come UNIX, Linux e Windows). Ogni agente può andare in esecuzione anche su un dispositivo con risorse limitate e le configurazioni di monitoraggio sono tutte residenti sul server.

Gli agenti supportano sia i check passivi (polling) che quelli attivi (trapping) che possono essere schedulati secondo intervalli di tempo personalizzati.

Alcune delle funzionalità offerte da un agente possono essere riassunte nella seguente tabella:

Network	Packets/bytes transfered Errors/dropped packets Collisions
CPU	Load average CPU idle/usage CPU utilization data per individual process
Memory	Free/used memory Swap/pagefile utilization
Disk	Space free/used Read and write I/O
Service	Process status Process memory usage Service status (ssh, ntp, ldap, smtp, ftp, http, pop, nntp, imap) Windows service status DNS resolution TCP connectivity TCP response time
File	File size/time File exists Checksum MD5 hash RegExp search
Log	Text log Windows eventlog
Other	System uptime System time Users connected Performance counter (Windows)

Figura 14: Alcune funzionalità di un agente

Un agente è in grado di creare file di log e questi ultimi vengono costantemente analizzati dall'agente stesso.

Il sistema supporta anche il Windows Management Instrumentation (WMI) che permette di monitorare sia workstation, sia server Windows.

L'agente inoltre esegue gli script userdefined; esso può essere così esteso aggiungendo alle sue funzionalità gli script utente, che possono essere scritti in diversi linguaggi di programmazione come SHELL, Perl, Python, Ruby, ecc....

Sono poi integrati anche gli SNMP agent (versioni supportate sono v1, v2 e v3) che permettono di raccogliere dati da tutti quei dispositivi su cui è installato SNMP, che possono essere sia componenti di rete, sia stampanti, NAS, UPS.

Per ottenere informazioni relative all'hardware di un dispositivo, vengono utilizzati gli IPMI agent, che sono presenti di default su server con architettura Intel come HP iLO e Dell DRAC. Attraverso questi agent, è possibile controllare la temperatura della CPU, la velocità di ventilazione, voltaggio del sistema e stato del disco fisico.

Attraverso Java gateway, è possibile monitorare applicazioni Java utilizzando la tecnologia JMX (Java Management Extensions); il server richiede al gateway uno specifico JMX counter il quale, attraverso le API JMX, può raccogliere informazioni su applicazioni Java senza la necessità di installare ulteriori componenti. Alcuni esempi di applicazioni che possono essere monitorati con JMX sono JBoss, Tomcat, GlassFish, WebSphere, ecc....

Zabbix supporta anche il monitoraggio VMware e di macchine virtuali utilizzando regole di discovery di basso livello e per automatizzare il processo, vengono creati dei prototipi di host.

Utilizzando la tecnologia ODBC, un server può memorizzare i dati in RDBMS database come MySQL, PostgreSQL, Oracle e Microsoft SQL Server. Il risultato delle query, viene memorizzato per consentire di creare grafi, allarmi o notifiche in caso di malfunzionamenti.

In *Zabbix* c'è anche la possibilità di agire anche in modalità agentless.

Network Services	TCP port availability TCP port response time Service check
ICMP Ping	Server availability ICMP response time Packet loss
Remote Check	Executing commands via SSH or Telnet

Figura 15: Funzionalità agentless

Zabbix server può controllare se un servizio è in ascolto su una determinata porta o se esso risponde in modo appropriato. Queste funzioni sono al momento supportati per FTP, IMAP, HTTP, HTTPS, LDAP, NNTP, POP3, SMTP, SSH, TCP e TELNET.

Zabbix può controllare se un host risponde a ICMP ping, per successivamente analizzare il tempo di risposta e la eventuale presenza di pacchetti persi e può eseguire comandi attraverso SSH e Telnet raccogliendo i risultati ottenuti.

6.1.8 Visualizzazione

Zabbix offre diverse soluzioni finalizzate alla visualizzazione dei dati da monitorare:

- Dashboard globale che rappresenta la pagina centrale del frontend di *Zabbix* e consente di personalizzare l'ambiente di monitoraggio. Le informazioni disponibili sono lo stato del *Zabbix* server, stato del sistema, stato delle macchine, ultimi 20 problemi, monitoraggio web, stato di discovery, grafici, screen e map preferite.

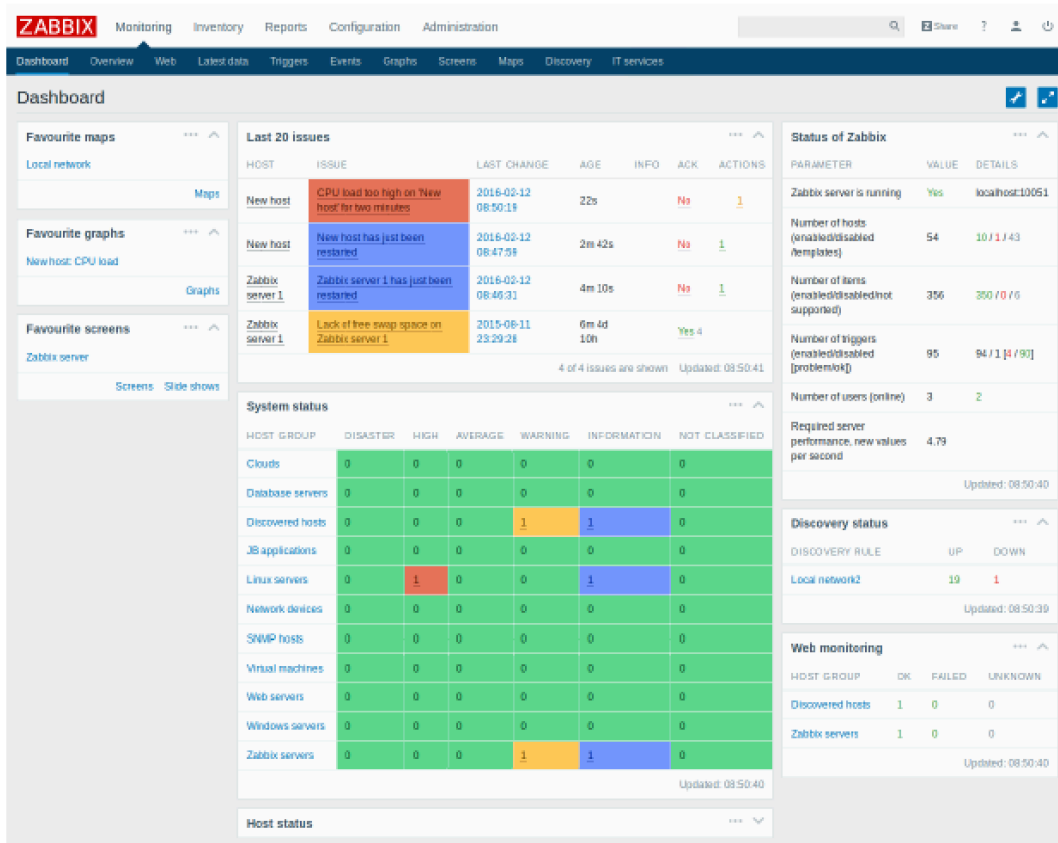


Figura 16: Dashboard

Grafici che rappresentano l'andamento dei dati raccolti e memorizzati. Grafici standard per valori numerici sono disponibili, a runtime, senza la necessità di configurazioni.

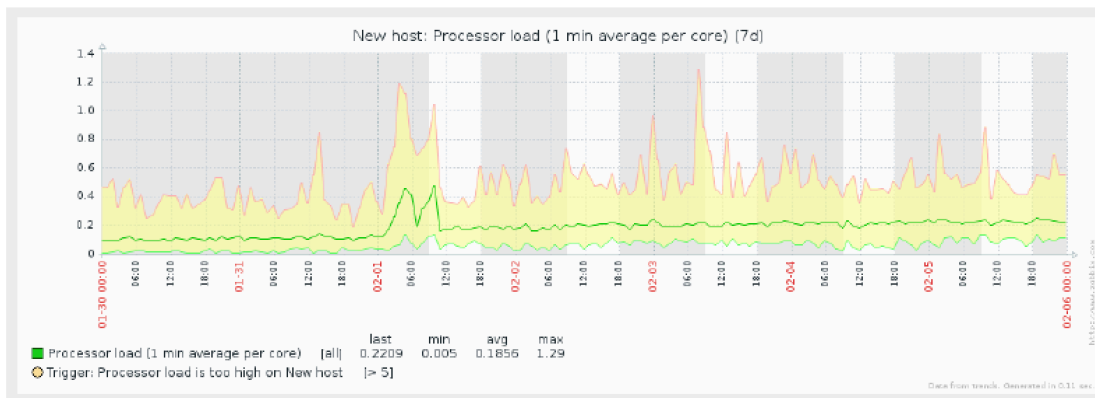


Figura 17: Grafico standard

C'è la possibilità di effettuare zoom di porzioni di grafico per analizzarlo in modo più dettagliato e individuare intervalli di tempo di interesse da visualizzare.

I grafici possono essere personalizzati per lo stile desiderato; si possono creare grafici che mostrano il confronto fra item differenti.

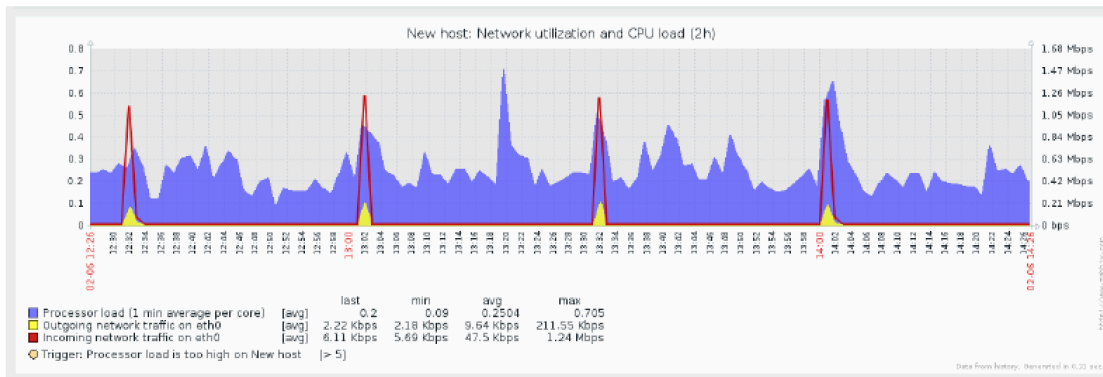


Figura 18: Grafico personalizzato

A titolo di esempio, nella Figura viene mostrato il carico del processore (in blu), il traffico di rete in uscita sull'interfaccia eth0 (in giallo) e il traffico di rete in entrata sull'interfaccia eth0 (in rosso);

- Mappe che offrono la possibilità di creare un ambiente monitorato userfriendly. Ogni elemento sulla mappa può rappresentare un host, un gruppo di host, un trigger, un'immagine o anche un'altra mappa. Quando gli elementi della mappa vengono collegati fra di loro e con uno o più trigger, essa si modifica in tempo reale. Quando si verifica un evento, le icone e il colore dei link cambiano automaticamente per consentire all'utente di osservare quello che succede nell'infrastruttura che si sta monitorando;

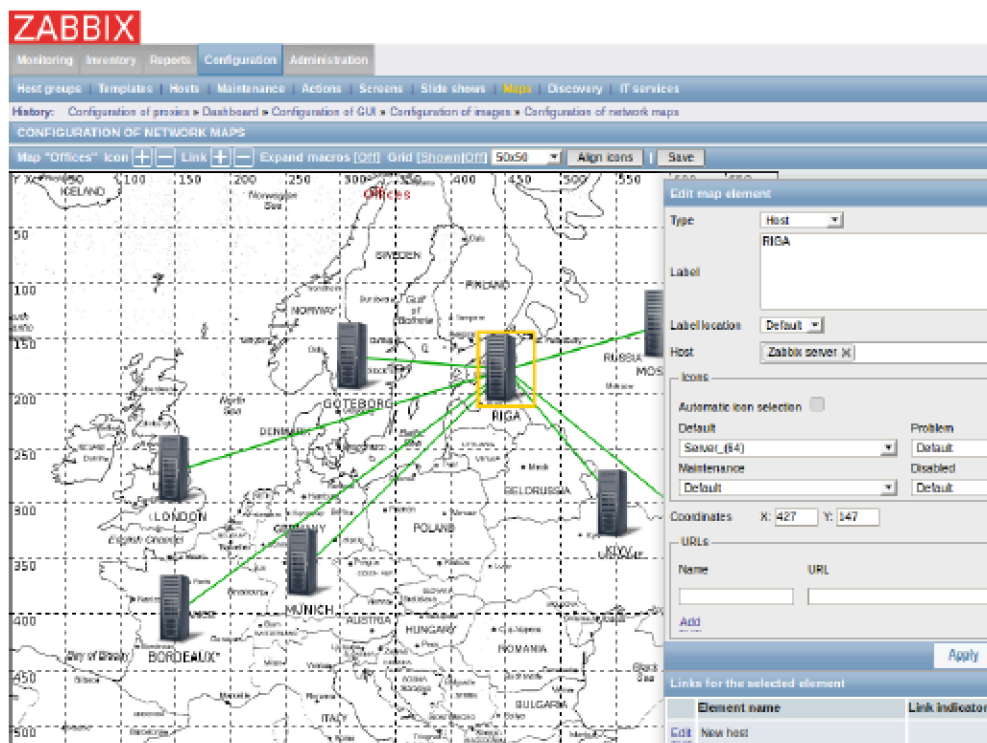


Figura 19: Esempio di mappa

- Tutti i dati raccolti e memorizzati possono essere visualizzati anche in formato tabellare;

TIMESTAMP	VALUE
2016-02-11 02:47:42	0.935
2016-02-11 02:46:42	0.885
2016-02-11 02:45:42	0.55

Figura 20: Raw Data

TIME	DESCRIPTION	STATUS	SEVERITY	DURATION	ACK	ACTIONS
2016-02-10 23:58:33	Zabbix agent on New host is unreachable for 5 minutes	OK	Average	2h 46m 14s	No	1
2016-02-10 23:56:00	Zabbix agent on New host is unreachable for 5 minutes	PROBLEM	Average	2m 33s	No	1
2016-02-09 22:55:45	Zabbix agent on New host is unreachable for 5 minutes	OK	Average	1d 1h	No	1
2016-02-09 02:45:00	Zabbix agent on New host is unreachable for 5 minutes	PROBLEM	Average	20h 10m 45s	No	1
2016-02-08 23:12:47	Disk IO is overloaded on New host	OK	Warning	2d 3h 32m	No	1

Figura 21: Eventi e dettagli di notifica

6.1.9 Sistema di notifica

Zabbix consente anche di informare del verificarsi di eventi importanti, usando diversi canali e opzioni. Esso fornisce un workflow per inviare notifiche, raccogliere gli acknowledge degli amministratori, scalare il problema ad altri operatori ed eseguire azioni in automatico sui sistemi coinvolti.

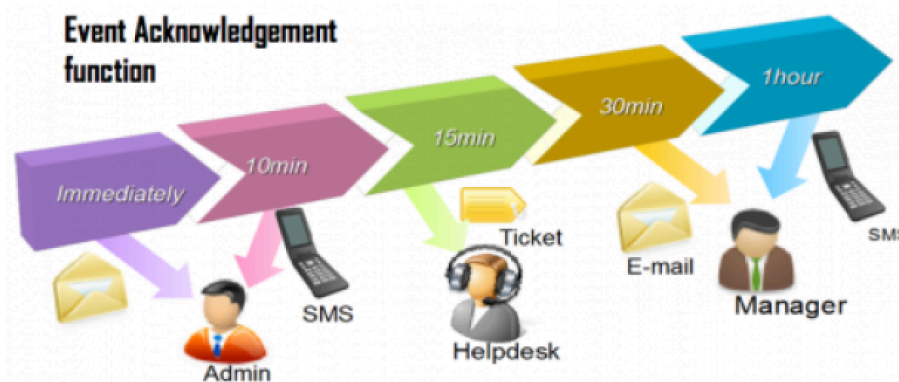


Figura 22: Sistema di notifiche

Zabbix dispone di diversi metodi predefiniti per l'invio delle notifiche:

- via e-mail;

- via SMS;
- via Jabber (messaging protocol);
- attraverso uno script personalizzato.

Inoltre gli alert, possono essere controllati da script. Il contenuto delle notifiche è personalizzabile in base al contesto; infatti, all'occorrenza, ogni contatto può essere configurato per ricevere notifiche solo per i livelli di severity desiderati, utilizzando i canali prescelti e negli orari e giorni specificati.

Ogni notifica può contenere le informazioni che consentono all'amministratore di comprendere il problema e sono: i dati identificativi del dispositivo, l'estratto dei file di log, un link verso l'interfaccia di management del dispositivo o alla documentazione online.

Il messaggio relativo ad un medesimo problema può essere personalizzato per uno specifico utente o gruppo di utenti. Esso può contenere informazioni differenti, in base al ruolo del destinatario (es.: operatore, amministratore, manager).

Zabbix offre anche l'opportunità di eseguire azioni automatiche; quando un trigger si attiva, comandi di SHELL possono essere eseguiti automaticamente sui sistemi remoti, ad esempio per rimediare a situazioni in cui un sistema sia sovraccarico o dei servizi abbiano smesso di funzionare. L'utilizzo tipico di questa funzione è per il riavvio di un servizio o eseguire il reboot di un server.

Tali comandi possono essere eseguiti:

- su *Zabbix* server;
- su *Zabbix* agent;
- utilizzando il protocollo IPMI;
- utilizzando il protocollo Telnet oppure SSH.

Quando si verifica un problema, *Zabbix* può notificare anche attraverso l'escalation; inizialmente viene inviata una notifica ad un solo destinatario ma, se il problema persiste e nessuno acknowledge viene ricevuto, verrà inviata una notifica ad altri destinatari. Infine, in assenza di ulteriori riscontri, il server eseguirà un comando automatico. Le regole supportate per definire uno scenario di escalation sono:

- notifica immediata dei nuovi problemi agli utenti;
- monitoraggio pro-attivo con *Zabbix* che esegue script predefiniti (comandi remoti);
- ripetizione della notifica finché il problema non viene risolto;
- notifiche ritardate;
- scalatura del problema ad altri gruppi di utenti;
- possibilità di scalare i problemi con o senza acknowledge;
- invio di messaggi di recovery a tutti i contatti coinvolti;
- possibilità di avere un numero illimitato di escalation.

Nelle notifiche può essere inserita l'intera cronologia di escalation in modo che il destinatario possa vedere cosa sta succedendo e perché ha ricevuto il messaggio.

6.2 Xymon

Xymon è un'applicazione per il monitoraggio degli host, degli applicativi e dei servizi di rete; esso permette anche monitoraggi specifici tramite estensioni. *Xymon* può generare periodicamente richieste ai servizi di rete (http, ftp, smtp e così via) e verificare se tali servizi rispondono come previsto. E' in grado di monitorare un vasto insieme di servizi di rete, ad esempio mail-server, web-server (sia semplici HTTP che HTTPS criptati), o utilizzo delle risorse. Permette anche di monitorare l'utilizzo del disco locale, i file di log e i processi attraverso l'uso di agenti installati sui server.

Tutti i risultati del monitoraggio sono raccolti da un server centrale e sono utilizzati per costruire un insieme di pagine web che mostrano lo stato della rete, con funzionalità di drill-down per controllare i problemi.

Xymon registra anche la cronologia di ogni elemento monitorato, in modo da poter generare rapporti di disponibilità e controllare eventuali malfunzionamenti. I dati vengono anche memorizzati per l'analisi dei trend e presentati sotto forma di grafici, in modo da poter tracciare facilmente, ad esempio, il tempo di risposta di un'applicazione web.

Se vengono riscontrati problemi vengono inviati appositi avvisi via e-mail, messaggi SMS o cercapersone, in modo che il personale tecnico possa rispondere tempestivamente senza essere costretto a sorvegliare i servizi.

Xymon è basato su un altro strumento di monitoraggio denominato Big Brother (BB), che è disponibile gratuitamente presso BB4 Technologies. Esso però è stato migliorato ed esteso sotto molti aspetti.

6.2.1 *Xymon gestisce il monitoraggio di molti sistemi*

Big Brother è implementato per lo più come script di SHELL e ciò influisce negativamente sulle prestazioni; per esempio in presenza di reti vaste, nelle quali è necessario monitorare centinaia o migliaia di host, l'elaborazione dei dati è carente. Un altro problema con BB è che esso memorizza tutte le informazioni di stato in file singoli; quando ci sono molti host, il numero degli I/O del disco sono tali da limitare fortemente il numero di sistemi che è possibile monitorare.

Xymon riesce ad evitare questi colli di bottiglia delle prestazioni principalmente sia perché la maggior parte dei dati è mantenuta in memoria anziché su disco, sia perché è implementato in C invece che in script SHELL.

6.2.2 *Xymon ha una configurazione centralizzata.*

Xymon mantiene tutti i dati di configurazione in un'unica locazione e cioè sul server *Xymon*. Big Brother invece ha molti file di configurazione memorizzati sui singoli server monitorati e ciò fa sì che per modificare un'impostazione può essere necessario effettuare il login su diversi server ed effettuare l'intervento su ciascuno di essi.

6.2.3 *Configurazione e installazione di Xymon*

Big Brother ha un numero molto elevato di componenti aggiuntivi, disponibili sul sito www.deadcat.net. Sebbene in questo modo è possibile trovare add-on per ciò che serve, molti di questi avrebbero dovuto essere inclusi del pacchetto base. E questi sono ad esempio la possibilità di tracciare i dati storici delle prestazioni, il monitoraggio dei servizi abilitati SSL e dei certificati SSL, o semplicemente una GUI per disabilitare temporaneamente il monitoraggio di un sistema.

In Xymon invece queste funzionalità sono incorporate e vengono fornite con il pacchetto base. I client inoltre, non richiedono modifiche di configurazione per l'installazione su più host.

6.2.4 *Xymon è in fase di sviluppo*

Le nuove versioni *Xymon* vengono rilasciate regolarmente, di solito ogni 4-6 mesi. Al contrario, lo sviluppo di BB sembra essersi fermato, almeno per quanto riguarda la versione non commerciale (BTF).

Xymon è sotto licenza Open Source, mentre BB non lo è.

Anche se la licenza BB "Better-than-Free" ne permette l'uso free per fini non commerciali, non è comunque un applicativo Open Source.

6.2.5 *Monitoraggio di host e reti*

Xymon raccoglie informazioni sui sistemi in due modi: da interrogazioni di servizi di rete (Web, LDAP, DNS, Mail, ecc.)(Figura 23), o da script che girano sul server *Xymon* o sui sistemi monitorati. Il pacchetto *Xymon* include un client *Xymon* che può essere installato sui server monitorati; esso raccoglie dati sul carico della CPU, sull'utilizzo del disco e della memoria, file di log, porte di rete in uso, informazioni su file e directory e altro ancora (Figura 24). Tutte le informazioni sono memorizzate all'interno di *Xymon*, possono generare condizioni che danno luogo ad avvisi, ad esempio se un servizio di rete smette di rispondere o un disco si riempie.

6.2.6 *Multiplatforma*

Il server *Xymon* gira su tutti i sistemi Unix-like, inclusi Linux, Solaris, FreeBSD, AIX, HP-UX e altri. Il client *Xymon* supporta tutte le principali piattaforme Unix, e ci sono altri progetti Open Source (ad esempio BBWin, vedi <http://bbwin.sourceforge.net/>), che forniscono supporto per sistemi basati su Microsoft Windows.

6.2.7 *Front-end basato sul web*

Con l'utilizzo del web gli host da monitorare possono essere raggruppati a piacimento e presentati in una struttura ad albero. All'interno delle pagine web vengono implementate tecniche per trasmettere informazioni sui sistemi monitorati, ad esempio vengono utilizzate icone diverse per gli stati cambiati di recente, i link alle sotto-pagine possono essere elencati in più colonne, si possono usare icone differenti per i dial-up-test o i reverse-test, le colonne selezionate possono essere eliminate per evitare la visualizzazione di informazioni

indesiderate, o possono includere determinate informazioni in modo permanente, si possono mostrare nomi a piacere per gli host indipendentemente dal loro vero hostname.

6.2.8 *Analisi integrata dei trend, dati storici e reporting SLA*

Tutte le misure vengono tracciate e rese disponibili in grafici basati sul tempo.

Quando è necessario approfondire gli eventi che si sono verificati, *Xymon* fornisce la possibilità di visualizzazione della cronologia degli eventi per ogni registro di stato, con il dettaglio di quando si sono verificati problemi in passato e uno zoom-in sull'evento specifico.

Per quanto riguarda il reporting SLA (service level agreement), è possibile pianificare i tempi di fermo macchina, il livello e il tempo di disponibilità del servizio e far generare dei report che mostrino la disponibilità effettiva rispetto allo SLA concordato. Tali report di disponibilità del servizio possono essere generati al volo, o pre-generati ad esempio per il reporting mensile.

6.2.9 *Visualizzazioni basate sui ruoli*

È possibile avere più viste diverse dello stesso host; ad esempio una vista per le esigenze del gruppo hardware e un'altra per i webmaster.

Si interviene inoltre con la configurazione di quali avvisi appariranno sui monitor. Per esempio, una semplice anomalia nel file di log del sistema non deve necessariamente attivare una chiamata al supporto di 3° livello alle 2 del mattino, ma se cade il trading on-line occorre che qualcuno intervenga immediatamente.

6.2.10 *Adattamento alle esigenze*

Sebbene nella versione base siano inclusi molti test, ci può essere qualcosa di specifico che sarebbe opportuno monitorare. C'è la possibilità di scrivere script di test nel linguaggio di scripting preferito e di far apparire i risultati come normali colonne di stato.

6.2.11 *Test di servizio di rete*

Lo strumento di test di rete agisce sui protocolli più comunemente usati, tra cui HTTP, SMTP (e-mail), DNS, LDAP (servizi di directory), e molti altri. Quando si testano i siti web, è possibile non solo controllare che il server web risponda, ma anche che la risposta sia corretta, confrontando la risposta con uno schema predefinito o un check-sum. In questo modo si può verificare non solo che il servizio di rete sia in funzione, ma anche che esso fornisca i dati attesi.

Sono supportati i protocolli che utilizzano la crittografia SSL come i siti web https; durante il controllo di tali servizi il tester di rete eseguirà automaticamente un controllo della validità del certificato del server SSL e avvertirà se quali certificati stanno per scadere.

6.2.12 Avvisi

Xymon consente di definire diversi criteri per l'invio di un alert, in modo da ricevere avvisi solo quando c'è davvero qualcosa per la quale si debba intervenire in modo urgente. Durante la gestione di un malfunzionamento, è possibile informare il sistema in modo esso interrompa l'invio di ulteriori avvisi.

6.2.13 Sicurezza

Tutti gli strumenti del server funzionano con un account utente non privilegiato ed è consigliato che venga dedicato uno specifico account per *Xymon*. Le pagine web vengono generate dinamicamente attraverso programmi CGI.

La possibilità di visualizzazione di tali pagine è dipendente dai controlli di accesso al server web, ad esempio è possibile richiedere un login attraverso una qualche forma di autenticazione HTTP

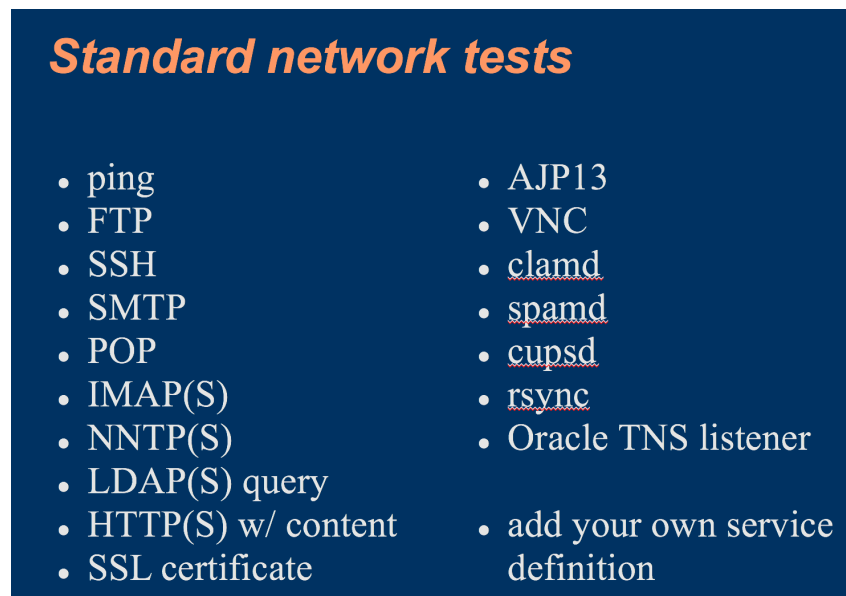


Figura 23: Test di rete

Standard server tests

- CPU load average
 - System uptime
 - System clock
 - Memory usage
 - Swap usage
 - File system usage
 - Process counts
 - Network ports
 - File attributes
 - File data
 - Directory sizes
 - Log file data
- Data can be graphed

Figura 24: Server test

7. Riferimenti

- [1] Tim Nuvola It Self Data Center <https://www.timbusiness.it/cloud-computing/data-center-e-housing/tim-self-data-center>
- [2] J. Schad, J. Dittrich, and J.A. Quiane-Ruiz, Runtime Measurements in the Cloud: Observing, Analyzing, and Reducing Variance, 2010.
- [3] Jin Shao, Qianxiang Wang, A Performance Guarantee Approach for Cloud Applications Based on Monitoring, 2011.
- [4] Jin Shao, Qianxiang Wang, A Performance Guarantee Approach for Cloud Applications Based on Monitoring, 2011.
- [5] Jae Yoo Lee, Soo Dong Kim, Software Approaches to Assuring High Scalability in Cloud Computing, 2010
- [6] G.A. Cignoni, P. De Risi, "Il test e la qualità del software", Ed. Il Sole 24 Ore, Milano, 1998.
- [7] Hong Zhu, Patrick A. V. Hall, John H. R. May, Software unit test coverage and adequacy, ACM Computing Surveys Volume 29, Issue 4, 1997.
- [8] Mauro Pezzè and Michael Young, Software Test and Analysis: Process, Principles, and Techniques John Wiley & Sons, 2007.
- [9] Dragich, Larry (4 April 2012). "The Anatomy of APM – 4 Foundational Elements to a Successful Strategy". APM Digest