



*Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte
Prestazioni*

Monitoraggio dell'infrastrutture di rete e di calcolo dell'ICAR-CNR Sede di Napoli

Angelo Esposito, Gennaro Oliva

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio
Nazionale delle Ricerche (ICAR-CNR)
Via Pietro Castellino, 111 – 80131 Napoli

angelo.esposito@icar.cnr.it

gennaro.oliva@icar.cnr.it

RT-ICAR-NA-2021-04

Dicembre 2021



Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-0816139531, e-mail: napoli@icar.cnr.it, URL: www.icar.cnr.it



**Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte
Prestazioni**

Monitoraggio dell'infrastrutture di rete e di calcolo dell'ICAR-CNR Sede di Napoli

Angelo Esposito, Gennaro Oliva

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio
Nazionale delle Ricerche (ICAR-CNR)
Via Pietro Castellino, 111 – 80131 Napoli

angelo.esposito@icar.cnr.it

gennaro.oliva@icar.cnr.it

RT-ICAR-NA-2021-04

Dicembre 2021

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità degli autori, descrivono l'attività del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

Monitoraggio dell'infrastrutture di rete e di calcolo dell'ICAR-CNR Sede di Napoli

Angelo Esposito, Gennaro Oliva

Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche (ICAR-CNR) Via Pietro Castellino, 111 – 80131 Napoli

angelo.esposito@icar.cnr.it, gennaro.oliva@icar.cnr.it

Abstract

Questo lavoro nasce in continuità rispetto all'attività di ricognizione svolta e descritta in [1]. Dalla attività di ricognizione sono emerse diverse criticità, che sono state risolte dagli autori attraverso interventi tecnici e di riconfigurazione degli apparati di rete. Considerando l'esperienza pregressa, la grande quantità di risorse di rete e di calcolo da gestire, gli autori del presente RT hanno deciso di predisporre un sistema di monitoraggio centralizzato, allo scopo di prevenire eventuali malfunzionamenti ed intervenire in maniera efficace ed efficiente nel caso questi avvengano. In questo RT è presentato il sistema di monitoraggio delle risorse di rete e di calcolo ad alte prestazioni configurato dagli autori del presente RT ed in uso presso la sede di Napoli dell'ICAR.

Keywords: Infrastruttura di rete e calcolo, Monitoraggio, Backup

1. Introduzione

L'Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) [2], è un Istituto del Consiglio Nazionale delle Ricerche (CNR) [3] che afferisce al Dipartimento di Ingegneria, ICT e Tecnologie per l'Energia e i Trasporti (DIITET) [4]. L'Istituto è presente sul territorio nazionale con tre sedi, Rende, Napoli e Palermo. La sua missione è quella di sviluppare ricerca, trasferimento tecnologico e alta formazione nell'area dei sistemi intelligenti a funzionalità complessa. L'Istituto sviluppa applicazioni significative nel campo dell'e-health, energia, sicurezza, bioinformatica, beni culturali, High Performance Computing e città intelligenti.

L'infrastruttura di rete, per la sede di Napoli, è stata descritta in lavoro [1], a valle di una attività di ricognizione. Tale attività ha portato alla luce diverse criticità, che sono state risolte mediante interventi tecnici e di riconfigurazione degli apparati di rete dell'ICAR. Considerando l'esperienza pregressa e la grande quantità di risorse di rete e di calcolo da gestire, gli autori del presente RT hanno deciso di predisporre un sistema di monitoraggio centralizzato degli apparati di rete e di calcolo, grazie al quale prevenire eventuali malfunzionamenti ed intervenire in maniera efficace ed efficiente nel caso questi avvengano.

Il sistema di monitoraggio nel suo complesso è costituito da un insieme di software installati e opportunamente configurati su un server da cui è possibile verificare lo stato di tutti gli apparati di rete e di calcolo ad alte prestazioni dell'ICAR. Il sistema consente una visione globale dello stato dei servizi e delle risorse in tempo reale, ed in caso di malfunzionamenti invia opportune notifiche agli amministratori di sistema.

2. Configurazione Server di Monitoring

Il server di monitoraggio offre una serie di servizi per la gestione dell'intera infrastruttura di calcolo e reti ad alte prestazioni dell'ICAR. Il server è collegato alla rete campus mediante due interfacce di rete, in modo da raggiungere sulle varie subnet gli apparati di rete, i server e i servizi da monitorare.

In Figura 1 è presentato graficamente il server di monitoring, dove sono riportati i servizi attivi e le infrastrutture di rete e calcolo monitorate. Nei successivi paragrafi sono descritti nel dettaglio i servizi predisposti e la loro configurazione sul server di monitoring.

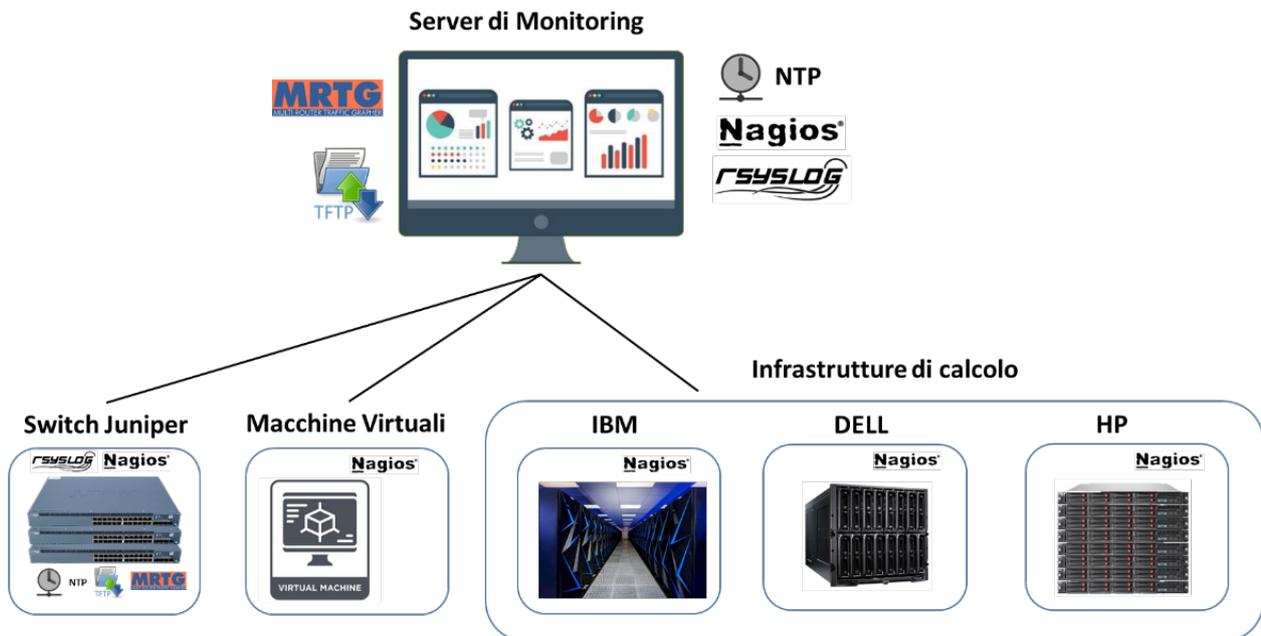


Figura 1 - Server di Monitoraggio

2.1. Network Time Protocol Server

La sincronizzazione oraria degli apparati di rete è di fondamentale importanza per comprendere le dinamiche temporali degli eventi che occorrono sui diversi dispositivi. Lo standard di riferimento per la sincronizzazione oraria è il protocollo Network Time Protocol [5]. Tale protocollo, client-server appartenente al livello applicativo, in ascolto sulla porta UDP 123, consente la sincronizzazione degli orologi di diversi apparati tecnologici collegati ad una rete. Il server di monitoring espone un servizio "NTP Server" grazie al quale i diversi dispositivi sulla rete sono sincronizzati. In Figura 2 i client collegati e sincronizzati con "NTP Server" sono tutti gli switch della rete ICAR. **Gli indirizzi IP dei server contenuti in tutte le immagini di questo rapporto vengono oscurati per ragioni di sicurezza.**

```

root@Monitoring:~# chronyc clients

```

Hostname	NTP	Drop	Int	IntL	Last	Cmd	Drop	Int	Last
.192	14	0	7	-	19	0	0	-	-
.237	11	0	8	-	121	0	0	-	-
.247	2	0	10	-	589	0	0	-	-
.236	2	0	10	-	561	0	0	-	-
.234	2	0	10	-	422	0	0	-	-
.216	2	0	10	-	300	0	0	-	-
.233	2	0	10	-	89	0	0	-	-
.cnr.it	1	0	-	-	990	0	0	-	-
.250	1	0	-	-	905	0	0	-	-
.238	1	0	-	-	820	0	0	-	-
.17	1	0	-	-	722	0	0	-	-

Figura 2 - Clients del servizio Chrony

Il server di monitoring espone il servizio “NTP Server” mediante l’applicativo Chrony [6] configurato per consentire la sincronizzazione oraria da parte dei diversi dispositivi collegati alla rete. Nella Figura 3 è riportato un estratto del file di configurazione “/etc/chrony.cfg”, dove sono specificate le subnet/apparati di rete ai quali è consentita la sincronizzazione oraria.

```
# Allow NTP client access from local network.
allow .0/8
allow .0/8
allow .0/8
allow .0/8
```

Figura 3 - Estratto file chrony.cfg

Agli switch Juniper è stato necessario aggiungere alla configurazione l’indirizzo ip dell’NTP Server (Figura 4).

```
ntp {
  server .232 prefer;
}
```

Figura 4 - Estratto file di configurazione switch

2.2. Nagios

Nagios [7] è una applicazione open source per il monitoraggio di computer e risorse di rete. La sua funzione base è quella di controllare host, reti e servizi, notificando malfunzionamenti mediante opportuni alert. Allo scopo di monitorare i diversi dispositivi collegati alla rete ICAR-CNR è stato installato e configurato l’applicativo Nagios su una vm in esecuzione sul server di monitoring. Oltre agli apparati di rete sono monitorati, mediante Nagios, anche i server e le virtual machines attive sulla rete ed i servizi da essi forniti. Ad oggi Nagios monitora 835 servizi e 116 host (Figura 5).

La configurazione di Nagios ha previsto, per ogni host da monitorare, la creazione di un file di memorizzato nella directory /usr/local/nagios/etc/objects, richiamato nel file di configurazione generale “/usr/local/nagios/etc/nagios.cfg”.

The screenshot displays the Nagios web interface. On the left is a navigation menu with sections: General, Current Status, Host Groups, Problems, Reports, and System. The main content area is divided into several sections:

- Current Network Status:** Last Updated: Mon Oct 19 15:46:27 CEST 2021, Updated every 30 seconds. Nagios® Core™ 4.4.6 - www.nagios.org. Logged in as nagiosadmin.
- Host Status Totals:**

Up	Down	Unreachable	Pending
116	1	0	0
- Service Status Totals:**

Ok	Warning	Unknown	Critical	Pending
835	47	1	8	0
- Service Overview For All Host Groups:**
 - DELL Alienware Aurora R4 (HyperV-90):**

Host	Status	Services	Actions
Backup.na.icar.cnr.it-14-190	UP	2 OK	[Icons]
GestioneProgetti-14.73	UP	2 OK	[Icons]
Intranet-14.215	UP	4 OK	[Icons]
Na.icar.cnr.it-14.40	UP	4 OK	[Icons]
StolCAR-14.72	UP	2 OK	[Icons]
 - IBM-Infrastructure (IBM):**

Host	Status	Services	Actions
Ben-14.82	UP	2 OK	[Icons]
R1-ibsw01.20.175	UP	1 OK	[Icons]
R3-ibsw02.20.176	UP	1 OK	[Icons]
R3-ibsw03.20.172	UP	1 OK	[Icons]
R3-ibsw04.20.173	UP	1 OK	[Icons]
ben00-20.180	UP	2 OK	[Icons]
ben01-20.181	UP	2 OK	[Icons]
ben02-20.182	UP	2 OK	[Icons]
ben03-20.183	UP	2 OK	[Icons]
ben04-20.184	UP	2 OK	[Icons]
ben05-20.185	UP	2 OK	[Icons]
ben06-20.186	UP	2 OK	[Icons]
ben07-20.187	UP	2 OK	[Icons]
ben08-20.188	UP	2 OK	[Icons]
ben09-20.189	UP	2 OK	[Icons]
ben10-20.190	UP	2 OK	[Icons]
ben11-20.191	UP	2 OK	[Icons]
ben12-20.192	UP	2 OK	[Icons]

Figura 5 - Schermata Nagios

Nella Figura 6 è riportato un estratto del file “nagios.cfg” dove sono specificati i path relativi ai file di configurazione degli host da monitorare.

```
#loadctl_options=jobs_max=100;backoff_limit=10;rampup_change=5
cfg_file=/usr/local/nagios/etc/objects/ConferenceSites-14.181.cfg
cfg_file=/usr/local/nagios/etc/objects/Millennium-14.209.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.17Ed4_p2.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.216Ed4_p-1.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.233Ed4_p1.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.234Ed4_p2.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.236Ed4_p-1.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.237Ed4_p1.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.238Ed4_p2.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.247Ed4_p0.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.250Ed4_p2.cfg
cfg_file=/usr/local/nagios/etc/objects/ji.192Ed4_p0.cfg
cfg_file=/usr/local/nagios/etc/objects/Intranet-14.215.cfg
cfg_file=/usr/local/nagios/etc/objects/Nalilab-14.67.cfg
cfg_file=/usr/local/nagios/etc/objects/WindowsPresenze-14.177.cfg
cfg_file=/usr/local/nagios/etc/objects/Fabric-FHIR-14.185.cfg
cfg_file=/usr/local/nagios/etc/objects/GestioneProgetti-14.73.cfg
cfg_file=/usr/local/nagios/etc/objects/SitoICAR-14.72.cfg
cfg_file=/usr/local/nagios/etc/objects/Covid-14.14.75.cfg
cfg_file=/usr/local/nagios/etc/objects/Na.icar.cnr.it-14.40.cfg
cfg_file=/usr/local/nagios/etc/objects/DNS1.ICAR-14-3.cfg
cfg_file=/usr/local/nagios/etc/objects/DNS2.ICAR-14-4.cfg
cfg_file=/usr/local/nagios/etc/objects/DNSAREA.NA-7.8.cfg
cfg_file=/usr/local/nagios/etc/objects/VMMinutolo-14.216.cfg
cfg_file=/usr/local/nagios/etc/objects/ServerAngelo-14.212.cfg
```

Figura 6 - Estratto del file nagios.cfg

2.2.1. Monitoraggio Switch Juniper

Utilizzando l'applicativo Nagios sono monitorati costantemente gli switch Juniper della rete ICAR sede di Napoli (Figura 7). Per ogni switch viene controllata la raggiungibilità e la disponibilità del servizio SSH.

Juniper EX2200 (Switch)			
Host	Status	Services	Actions
ji.17Ed4_p2	UP	2 OK	  
ji.192Ed4_p0	UP	2 OK	  
ji.216Ed4_p-1	UP	2 OK	  
ji.233Ed4_p1	UP	2 OK	  
ji.234Ed4_p2	UP	2 OK	  
ji.236Ed4_p-1	UP	2 OK	  
ji.237Ed4_p1	UP	2 OK	  
ji.238Ed4_p2	UP	2 OK	  
ji.247Ed4_p0	UP	2 OK	  
ji.250Ed4_p2	UP	2 OK	  

Figura 7 - Schermata Nagios – monitoraggio Switch Juniper

Nella Figura 8 è mostrato il dettaglio dei servizi controllati, come si evince dalla figura per ogni servizio monitorato sono riportati una serie di dettagli, come ad esempio il tipo di protocollo SSH oppure RTA (Round Trip Average in millisecondi) di risposta al ping.

Limit Results: 100						
Host	Service	Status	Last Check	Duration	Attempt	Status Information
ji.17Ed4_p2	PING	OK	10-20-2021 16:26:28	49d 2h 24m 33s	1/4	PING OK - Packet loss = 0%, RTA = 2.65 ms
	SSH	OK	10-20-2021 16:27:06	49d 2h 24m 46s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.192Ed4_p0	PING	OK	10-20-2021 16:27:44	49d 2h 29m 30s	1/4	PING OK - Packet loss = 0%, RTA = 2.59 ms
	SSH	OK	10-20-2021 16:28:21	49d 2h 26m 58s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.216Ed4_p-1	PING	OK	10-20-2021 16:23:59	15d 10h 29m 8s	1/4	PING OK - Packet loss = 0%, RTA = 2.58 ms
	SSH	OK	10-20-2021 16:24:09	49d 3h 8m 7s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.233Ed4_p1	PING	OK	10-20-2021 16:25:13	18d 21h 2m 6s	1/4	PING OK - Packet loss = 0%, RTA = 3.17 ms
	SSH	OK	10-20-2021 16:25:51	49d 2h 49m 11s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.234Ed4_p2	PING	OK	10-20-2021 16:28:37	0d 3h 16m 30s	1/4	PING OK - Packet loss = 0%, RTA = 2.65 ms
	SSH	OK	10-20-2021 16:27:06	49d 2h 27m 8s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.236Ed4_p-1	PING	OK	10-20-2021 16:27:44	49d 3h 6m 28s	1/4	PING OK - Packet loss = 0%, RTA = 3.11 ms
	SSH	OK	10-20-2021 16:28:22	49d 2h 27m 3s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.237Ed4_p1	PING	OK	10-20-2021 16:23:59	4d 23h 29m 7s	1/4	PING OK - Packet loss = 0%, RTA = 2.96 ms
	SSH	OK	10-20-2021 16:24:09	49d 2h 54m 21s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.238Ed4_p2	PING	OK	10-20-2021 16:25:13	1d 3h 37m 49s	1/4	PING OK - Packet loss = 0%, RTA = 2.29 ms
	SSH	OK	10-20-2021 16:25:51	49d 2h 31m 31s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)
ji.247Ed4_p0	PING	OK	10-20-2021 16:26:29	11d 5h 36m 38s	1/4	PING OK - Packet loss = 0%, RTA = 2.72 ms
	SSH	OK	10-20-2021 16:27:07	49d 2h 29m 21s	1/4	SSH OK - OpenSSH_6.0 (protocol 2.0)

Figura 8 - Monitoraggio servizi Switch

A titolo esemplificativo nella Figura 9 è riportato il file di configurazione per il monitoraggio dello switch ji.17Ed4_p2. Come si può notare sono specificati i due servizi da controllare, in questo caso PING e SSH.

```
define host {
    use linux-server
    host_name ji.17Ed4_p2
    address [REDACTED].17
}
define service {
    use local-service
    host_name ji.17Ed4_p2
    service_description PING
    check_command check_ping!100.0,20%!500.0,60%
    notifications_enabled 0
}
define service {
    use local-service
    host_name ji.17Ed4_p2
    service_description SSH
    check_command check_ssh
    notifications_enabled 0
}
```

Figura 9 - Estratto file di configurazione monitoraggio switch di Nagios

2.2.2. Monitoraggio delle Macchine Virtuali

L'infrastruttura tecnologica dell'ICAR ospita un ingente numero di VM opportunamente configurate per offrire una serie di servizi (interni ed esterni all'Istituto). Le VM monitorate (Figura 10) sono divise in "Host Groups", che rappresentano i server fisici su cui sono in esecuzione. Il monitoraggio di ogni VM è in funzione dei servizi che la stessa eroga.

DELL Alienware Aurora R4 (HyperV-80)			
Host	Status	Services	Actions
Backup.na.icar.cnr.it-14.190	UP	2 OK	  
GestioneProgetti-14.73	UP	2 OK	  
Intranet-14.215	UP	4 OK	  
Na.icar.cnr.it-14.40	UP	4 OK	  
SitoICAR-14.72	UP	2 OK	  

Yoda (Yoda-202)			
Host	Status	Services	Actions
Benvpn-14.189	UP	2 OK	  
DNS1.ICAR-14.3	UP	3 OK	  
Dbremiam.na.icar.cnr.it-14.94	UP	3 OK	  
Icarvpn-14.183	UP	2 OK	  
Porbec.it-14.25	UP	3 OK	  
Vpn-databencart-14.84	UP	2 OK	  
Yoda-14.202	UP	2 OK	  

HP Proliant ML350p Gen8 (Millennium-209)			
Host	Status	Services	Actions
DNSAREA.NA-7.8	UP	2 OK	  
Millennium-78.209	UP	1 OK	  
ServerAngelo-14.212	UP	2 OK	  
ServerOpenVPN-14.183	UP	2 OK	  
VMMinutolo-14.216	UP	2 OK	  

EMC-Solo1 (Solo1-217)			
Host	Status	Services	Actions
AI4healthsec-14.211	UP	3 OK	  
Bracs.icar.cnr.it-14.210	UP	3 OK	  
Cat.Covid-14.102	UP	1 OK	  
ConferenceSites-14.181	UP	15 OK	  
Covid-14.14.75	UP	2 OK	  
DNS2.ICAR-14.4	UP	3 OK	  
Fabric-FHIR-14.185	UP	2 OK	  
Nailiab-14.67	UP	3 OK	  
SitiWeb-14.214	UP	5 OK	  
Solo1-WmWare-78.217	UP	2 OK	  
WindowsManagment-14.68	UP	1 OK	  
WindowsPresenze-14.177	UP	1 OK	  
WindowsServer2019Molim-14.222	UP	1 OK	  
WindowsServer2019Remiam-14.223	UP	1 OK	  
WindowsServerRemiam2-14.224	UP	1 OK	  
vivara-14.221	UP	2 OK	  

Figura 10 – Estratto schermata Nagios Monitoraggio VM

Service	Status	Last Check	Duration	Attempt	Status Information
CANS2018	OK	10-18-2021 13:27:53	0d 10h 10m 56s	1/4	SSL OK - Certificate 'cans2018.na.icar.cnr.it' will expire in 70 days on 2021-12-27 22:20 +0100/CET.
PING	OK	10-18-2021 13:28:31	0d 10h 13m 9s	1/4	PING OK - Packet loss = 0%, RTA = 0.45 ms
SSH	OK	10-18-2021 13:24:07	0d 10h 9m 42s	1/4	SSH OK - OpenSSH_8.2p1 Ubuntu-4ubuntu0.3 (protocol 2.0)
Sellie21	OK	10-18-2021 13:24:45	0d 10h 9m 4s	1/4	SSL OK - Certificate 'sellie21.na.icar.cnr.it' will expire in 73 days on 2021-12-31 06:46 +0100/CET.
cans2018.na.icar.cnr.it	OK	10-18-2021 13:25:23	0d 10h 13m 26s	1/4	SSL OK - Certificate 'cans2018.na.icar.cnr.it' will expire in 70 days on 2021-12-27 22:20 +0100/CET.
sellie21.na.icar.cnr.it	OK	10-18-2021 13:26:00	0d 10h 12m 49s	1/4	SSL OK - Certificate 'sellie21.na.icar.cnr.it' will expire in 73 days on 2021-12-31 06:46 +0100/CET.
www.ai4hb2e.icar.cnr.it	OK	10-18-2021 13:26:38	0d 10h 12m 12s	1/4	SSL OK - Certificate 'www.ai4hb2e.icar.cnr.it' will expire in 73 days on 2021-12-31 06:46 +0100/CET.
www.ai4health.icar.cnr.it	OK	10-18-2021 13:27:16	0d 10h 10m 23s	1/4	SSL OK - Certificate 'www.ai4health.icar.cnr.it' will expire in 73 days on 2021-12-31 06:46 +0100/CET.
www.icts4ehealth.icar.cnr.it	OK	10-18-2021 13:27:53	0d 10h 10m 56s	1/4	SSL OK - Certificate 'www.icts4ehealth.icar.cnr.it' will expire in 73 days on 2021-12-31 06:46 +0100/CET.
www.mhaec.icar.cnr.it	OK	10-18-2021 13:28:31	0d 10h 10m 18s	1/4	SSL OK - Certificate 'www.mhaec.icar.cnr.it' will expire in 73 days on 2021-12-31 06:47 +0100/CET.
www.newk2020.icar.cnr.it	OK	10-18-2021 13:24:07	0d 10h 9m 42s	1/4	SSL OK - Certificate 'www.newk2020.icar.cnr.it' will expire in 73 days on 2021-12-31 06:47 +0100/CET.
www.newk2021.icar.cnr.it	OK	10-18-2021 13:24:45	0d 10h 9m 4s	1/4	SSL OK - Certificate 'www.newk2021.icar.cnr.it' will expire in 73 days on 2021-12-31 06:47 +0100/CET.
www.pdeim.icar.cnr.it	OK	10-18-2021 13:25:23	0d 10h 13m 26s	1/4	SSL OK - Certificate 'www.pdeim.icar.cnr.it' will expire in 73 days on 2021-12-31 06:47 +0100/CET.
www.pdeim2020.icar.cnr.it	OK	10-18-2021 13:26:01	0d 10h 12m 48s	1/4	SSL OK - Certificate 'www.pdeim2020.icar.cnr.it' will expire in 73 days on 2021-12-31 06:47 +0100/CET.
www.pdeim2021.icar.cnr.it	OK	10-18-2021 13:26:38	0d 10h 12m 11s	1/4	SSL OK - Certificate 'www.pdeim2021.icar.cnr.it' will expire in 73 days on 2021-12-31 06:47 +0100/CET.

Figura 11 – Estratto schermata Nagios Monitoraggio VM servizi

In Figura 11 è riportata una schermata di Nagios che visualizza lo stato dei servizi attivi sulla della VM “ConferenceSites-14.181”. In particolare, i controlli effettuati riguardano oltre alla raggiungibilità della VM e il servizio SSH anche la scadenza dei certificati digitale SSL dei vari siti web.

A titolo esemplificativo in Figura 12 è riportato un estratto del file di configurazione per la VM “ConferenceSites-14.181” su Nagios.

```

define host {
    use                linux-server
    host_name          ConferenceSites-14.181
    address             .181
    hostgroups         Solo1-217
}
define service {
    use                local-service
    host_name          ConferenceSites-14.181
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          ConferenceSites-14.181
    service_description SSH
    check_command      check_ssh
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          ConferenceSites-14.181
    service_description CANS2018
    check_command      check_http! -H cans2018.na.icar.cnr.it --ssl -p443 --sni -C 30,10
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          ConferenceSites-14.181
    service_description Sellie21
    check_command      check_http! -H sellie21.na.icar.cnr.it --ssl -p443 --sni -C 30,10
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          ConferenceSites-14.181
    service_description www.mhaec.icar.cnr.it
    check_command      check_http! -H www.mhaec.icar.cnr.it --ssl -p443 --sni -C 30,10
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          ConferenceSites-14.181
    service_description www.icts4ehealth.icar.cnr.it
    check_command      check_http! -H www.icts4ehealth.icar.cnr.it --ssl -p443 --sni -C 30,10
    notifications_enabled 0
}

```

Figura 12 – Estratto del file di configurazione “ConferenceSites-14.181.cfg”

2.2.3. Monitoraggio della Infrastruttura di calcolo IBM

L'ICAR dispone di una infrastruttura di calcolo ad alte prestazioni costituita da 25 nodi di calcolo IBM Power System (AC922), 2 nodi IBM Elastic Storage Server Management Server (5148-21L) e quattro nodi IBM Elastic Storage Server Data Server (5148-22L). I nodi di calcolo e storage sono connessi tra

loro mediante 4 switch infiniband 8828 Model E37. L'intera infrastruttura di calcolo è monitorata mediante Nagios come si evince in Figura 13.

IBM-Infrastructure (IBM)				ben07-20.187				ben21-20.201			
Host	Status	Services	Actions	Host	Status	Services	Actions	Host	Status	Services	Actions
Ben-14.82	UP	2 OK	[Icons]	ben08-20.188	UP	2 OK	[Icons]	ben21-20.201	UP	2 OK	[Icons]
R1-ibsw01.20.175	UP	1 OK	[Icons]	ben09-20.189	UP	2 OK	[Icons]	ben22-20.202	UP	2 OK	[Icons]
R3-ibsw02.20.176	UP	1 OK	[Icons]	ben10-20.190	UP	2 OK	[Icons]	ben23-20.203	UP	2 OK	[Icons]
R3-ibsw03.20.172	UP	1 OK	[Icons]	ben11-20.191	UP	2 OK	[Icons]	ben24-20.204	UP	2 OK	[Icons]
R3-ibsw04.20.173	UP	1 OK	[Icons]	ben12-20.192	UP	2 OK	[Icons]	benio03-20.168	UP	2 OK	[Icons]
ben00-20.180	UP	2 OK	[Icons]	ben13-20.193	UP	2 OK	[Icons]	benio04-20.169	UP	1 OK 1 WARNING	[Icons]
ben01-20.181	UP	2 OK	[Icons]	ben14-20.194	UP	2 OK	[Icons]	ems02-04-20.253	UP	2 OK	[Icons]
ben02-20.182	UP	2 OK	[Icons]	ben15-20.195	UP	2 OK	[Icons]	gssio01-20.177	UP	1 OK 1 WARNING	[Icons]
ben03-20.183	UP	2 OK	[Icons]	ben16-20.196	UP	2 OK	[Icons]	gssio02-20.178	UP	1 OK 1 WARNING	[Icons]
ben04-20.184	UP	2 OK	[Icons]	ben17-20.197	UP	2 OK	[Icons]	test14.212	UP	3 OK	[Icons]
ben05-20.185	UP	2 OK	[Icons]	ben18-20.198	UP	2 OK	[Icons]				
ben06-20.186	UP	2 OK	[Icons]	ben19-20.199	UP	2 OK	[Icons]				
				ben20-20.200	UP	2 OK	[Icons]				

Figura 13 - Schermata Nagios monitoraggio infrastruttura IBM

La configurazione di Nagios per il monitoraggio dell'Infrastruttura di calcolo IBM ha previsto una serie di attività descritte di seguito nel dettaglio:

1. È stato realizzato uno script “/usr/local/nagios/libexec/check_ipmi.sh” (Figura 14 – Appendice A) per verificare lo stato dei nodi utilizzando le schede di management out-of-band BMC (Baseboard Management Controller) mediante protocollo IPMI (Intelligent Platform Management Interface).

```
#!/bin/bash
# check_ipmi.sh
# Description: check bmc ibm via ipmi
# @Autor: Angelo Esposito CNR, angelo.esposito@icar.cnr.it

USER=root
HOST=$1
RACK=$2

COMANDOTEMP="tempfile"
FILENAME=$(COMANDOTEMP)

if [ $RACK = 1 ]; then
PASSWORD=XXXXXXX
COMANDO="ipmitool -I lanplus -H $HOST -P $PASSWORD sdr >> $FILENAME"
eval $COMANDO
COMANDOPOWER="ipmitool -I lanplus -H $HOST -P $PASSWORD chassis status"
fi

if [ $RACK = 3 ]; then
PASSWORD=YYYYYYY
COMANDO="ipmitool -I lanplus -H $HOST -U $USER -P $PASSWORD sdr >> $FILENAME"
eval $COMANDO
COMANDOPOWER="ipmitool -I lanplus -U $USER -H $HOST -P $PASSWORD chassis status"
fi

CPU2_STATUS=$(cat $FILENAME | grep -w "cpu1" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
CPU2_TEMP=$(cat $FILENAME | grep "p0_core0 temp" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
CPU1_STATUS=$(cat $FILENAME | grep -w "cpu0" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
CPU1_TEMP=$(cat $FILENAME | grep "p0_core0 temp" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN1_RPM=$(cat $FILENAME | grep -w "fan0_0" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN1_STATUS=$(cat $FILENAME | grep -w "fan0_0" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN2_RPM=$(cat $FILENAME | grep -w "fan0_1" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN2_STATUS=$(cat $FILENAME | grep -w "fan0_1" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN3_RPM=$(cat $FILENAME | grep -w "fan1_0" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN3_STATUS=$(cat $FILENAME | grep -w "fan1_0" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN4_RPM=$(cat $FILENAME | grep -w "fan1_1" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN4_STATUS=$(cat $FILENAME | grep -w "fan1_1" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN5_RPM=$(cat $FILENAME | grep -w "fan2_0" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN5_STATUS=$(cat $FILENAME | grep -w "fan2_0" | cut -d "|" -f 3 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
FAN6_RPM=$(cat $FILENAME | grep -w "fan2_1" | cut -d "|" -f 2 | sed 's/[[:space:]]*$/' | sed 's/^[:space:]]*$/')
```

```

STATUS=""
CRITICAL=0
WARNING=0
UNKNOWN=0

if [ "$SYSTEM_POWER" == "on" ]
then
STATUS="Power On"

if [ $CPU1_STATUS == "ok" ]
then
STATUS="$STATUS \nCPU1 Temp OK ( $CPU1_TEMP )"
else
STATUS="$STATUS \nCPU1 Temp CRITICAL ( $CPU1_TEMP )"
CRITICAL=1
fi

if [ $CPU2_STATUS == "ok" ]
then
STATUS="$STATUS \nCPU2 Temp OK ( $CPU2_TEMP )"
else
STATUS="$STATUS \nCPU2 Temp CRITICAL ( $CPU2_TEMP )"
CRITICAL=1
fi

if [ $FAN1_STATUS == "ok" ]
then
STATUS="$STATUS \nFAN1 OK ( $FAN1_RPM )"
else
STATUS="$STATUS \nFAN1 WARNING ( $FAN1_RPM )"
WARNING=1
fi

```

Figura 14 – Estratto del file “/usr/local/nagios/libexec/check_ipmi.sh”

2. È stato modificato il file “/usr/local/nagios/etc/objects/commands.cfg” aggiungendo il comando check-ipmi (Figura 15).

```

define command{
command_name    check-ipmi
command_line    bash $USER1$/check_ipmi.sh $HOSTADDRESS$ $ARG1$
}

```

Figura 15 – Estratto file “/usr/local/nagios/etc/objects/commands.cfg”

3. Per ogni nodo da monitorare è stato creato un file di configurazione, come ad esempio quello mostrato in Figura 16.

```

define host {
use                linux-server
host_name          ben00-20.180
address            .180
hostgroups         IBM
}
define service {
use                local-service
host_name          ben00-20.180
service_description PING
check_command      check_ping!100.0,20%!500.0,60%
notifications_enabled 0
}
define service {
use                local-service
host_name          ben00-20.180
service_description BMC
check_command      check-ipmi!1
notifications_enabled 0
}

```

Figura 16 – Estratto file di configurazione “/usr/local/nagios/etc/objects/ICAR/ ben00-20.180.cfg”

4. Al file nagios.cfg sono stati aggiunti i path relativi ai file di configurazione dei diversi nodi da monitorare.

Il risultato ottenuto su Nagios è riportato in Figura 17. Per ogni nodo dell’infrastruttura sono monitorati una serie di parametri, come ad esempio le temperature delle diverse componenti hardware, lo stato dei moduli di memoria DIMM, la potenza erogata degli alimentatori, lo stato delle schede video, la velocità di rotazione delle ventole di raffreddamento, etc.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ben00-20.180	BMC	OK	10-18-2021 15:17:20	18d 23h 58m 4s	1/4	OK Power On CPU1 Temp OK (34 degrees C) CPU2 Temp OK (34 degrees C) FAN1 OK (3200 revolution) FAN2 OK (3400 revolution) FAN3 OK (3200 revolution) FAN4 OK (3500 revolution) FAN5 OK (3100 revolution) FAN6 OK (3500 revolution) FAN7 OK (3100 revolution) FAN8 OK (3500 revolution) POWER0 OK (230 Watts) POWER1 OK (240 Watts) Temperature OK (22.20 degrees C) DIMM0 OK (ok) DIMM1 OK (ok) DIMM2 OK (ok) DIMM3 OK (ok) DIMM4 OK (ok) DIMM5 OK (ok) DIMM6 OK (ok) DIMM7 OK (ok) DIMM8 OK (ok) DIMM9 OK (ok) DIMM10 OK (ok) DIMM11 OK (ok) DIMM12 OK (ok) DIMM13 OK (ok) DIMM14 OK (ok) DIMM15 OK (ok) GV0_STATUS OK (ok) GV1_STATUS OK (ok) GV3_STATUS OK (ok) GV4_STATUS OK (ok)
	PING	OK	10-18-2021 15:17:57	17d 22h 48m 1s	1/4	PING OK - Packet loss = 0%, RTA = 0.66 ms

Figura 17 – Schermata Nagios controllo parametri nodi di calcolo dell’infrastruttura IBM

2.2.4. Monitoraggio Server DELL

L’infrastruttura di calcolo dell’ICAR comprende inoltre 2 chassis DELL PowerEdge M1000e con 10 server blade PowerEdge M620 ciascuno e uno chassis PowerEdge MX7000 con 3 server blade PowerEdge MX740c , un server Rack PowerEdge R720xd ed un server rack PowerEdge R620. Tutti i server e gli chassis sono monitorati con Nagios utilizzando il sistema DELL di management out-of-band Integrated Dell Remote Access Controller (IDRAC). Le attività compiute per il monitoraggio dei server DELL sono analoghe a quelle per i server IBM. In particolare, è stato aggiunto lo script idrac_2.2rc4 [8] alla directory “/usr/local/nagios/libexec/”, è stato modificato il file un “/usr/local/nagios/etc/objects/commands.cfg” aggiungendo il comando check-idrac (Figura 18), è stato creato per ogni server da monitorare un file di configurazione, come quello riportato in Figura 19. Infine, al file nagios.cfg sono stati aggiunti i path relativi ai file di configurazione dei diversi server DELL da monitorare. In Figura 20 è mostrata una schermata di Nagios che riporta i parametri monitorati per uno dei server DELL.

```
define command{
command_name    check-idrac
command_line    python $USER1$/idrac_2.2rc4 -f $USER1$/idrac_2.1.conf -H $HOSTADDRESS$ -v2c -c public -m $USER1$/idrac-smiv2.mib -w $ARG1$
}
```

Figura 18 – Estratto file usr/local/nagios/etc/objects/commands.cfg

```

define host {
    use                linux-server
    host_name          idrac-78.42-SOL02
    address            [REDACTED].42
    hostgroups        IDRAC
}
define service {
    use                local-service
    host_name          idrac-78.42-SOL02
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          idrac-78.42-SOL02
    service_description GLOBAL
    check_command      check-idrac!GLOBAL
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          idrac-78.42-SOL02
    service_description MEMORY-1
    check_command      check-idrac!MEM#1
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          idrac-78.42-SOL02
    service_description MEMORY-2
    check_command      check-idrac!MEM#2
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          idrac-78.42-SOL02
    service_description MEMORY-3
    check_command      check-idrac!MEM#3
    notifications_enabled 0
}
define service {
    use                local-service
    host_name          idrac-78.42-SOL02
    service_description MEMORY-4
    check_command      check-idrac!MEM#4
    notifications_enabled 0
}

```

Figura 19 – Estratto file di configurazione “/usr/local/nagios/etc/objects/ICAR/idrac78.42.cfg”

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
blade00-20.100	BATTERY1	OK	10-19-2021 11:46:05	84d 19h 59m 10s	1/4	OK - System Board CMOS Battery: ENABLED/OK [PRESENCEDETECTED]
	BATTERY2	OK	10-19-2021 11:41:43	84d 19h 55m 39s	1/4	OK - PERC1 ROMB Battery: ENABLED/OK [PRESENCEDETECTED]
	CPU1	OK	10-19-2021 11:42:20	47d 23h 50m 5s	1/4	OK - CPU1 (8 cores/16 threads): ENABLED/OK [Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz]
	CPU2	OK	10-19-2021 11:42:58	19d 20h 12m 46s	1/4	OK - CPU2 (8 cores/16 threads): ENABLED/OK [Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz]
	DISK1	WARNING	10-19-2021 11:43:36	112d 19h 28m 53s	4/4	WARN - PDisk 1 (0.1.0) 558.38 GB: FOREIGN(!), PowerStat: 0, HotSpare: no [SEAGATE, SPUNUP, S/N: S0M05QAC] isFailing: HDD
	DISK2	OK	10-19-2021 11:44:06	84d 19h 54m 30s	1/4	OK - PDisk 2 (0.1.1) 558.38 GB: ONLINE, PowerStat: 0, HotSpare: no [SEAGATE, SPUNUP, S/N: S0M055QJ] isFailing: HDD
	GLOBAL	WARNING	10-19-2021 11:44:50	84d 22h 35m 57s	4/4	systemStateGlobalSystemStatus NONCRITICAL(!)
	PING	OK	10-19-2021 11:45:28	47d 23h 50m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.83 ms
	SENSOR1	OK	10-19-2021 11:46:06	84d 19h 57m 50s	1/4	OK - System Board Inlet Temp: 21.0 C ENABLED/OK
	SENSOR2	OK	10-19-2021 11:41:43	84d 19h 59m 10s	1/4	OK - CPU1 Temp: 38.0 C ENABLED/OK
	SENSOR3	OK	10-19-2021 11:42:21	84d 19h 55m 39s	1/4	OK - CPU2 Temp: 33.0 C ENABLED/OK
	VDISK	WARNING	10-19-2021 11:42:58	84d 19h 50m 8s	4/4	VDisk 1 (Virtual Disk 0): NONCRITICAL(!)/DEGRADED(!), RAID-1 (558.38 GB), BadBlock: 0 [Virtual Disk 0 on Integrated RAID Controller 1]
	aa-MEMORY1	OK	10-19-2021 11:43:36	19d 20h 8m 14s	1/4	OK - Memory 1 (DIMM Socket A1) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 36802577]
	ab-MEMORY2	OK	10-19-2021 11:44:06	84d 19h 58m 57s	1/4	OK - Memory 2 (DIMM Socket A2) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 3630255E]
	ac-MEMORY3	OK	10-19-2021 11:44:50	84d 19h 59m 30s	1/4	OK - Memory 3 (DIMM Socket A3) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 36502595]
	ad-MEMORY4	OK	10-19-2021 11:45:28	84d 19h 58m 58s	1/4	OK - Memory 4 (DIMM Socket A4) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 36802588]
	ae-MEMORY5	OK	10-19-2021 11:46:06	84d 19h 58m 24s	1/4	OK - Memory 5 (DIMM Socket A5) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 157D2A35]
	af-MEMORY6	OK	10-19-2021 11:41:43	84d 19h 58m 58s	1/4	OK - Memory 6 (DIMM Socket A6) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 157D2A3D]
	ag-MEMORY7	OK	10-19-2021 11:42:21	84d 19h 58m 10s	1/4	OK - Memory 7 (DIMM Socket A7) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 154D2A38]
	ah-MEMORY8	OK	10-19-2021 11:42:59	19d 20h 12m 45s	1/4	OK - Memory 8 (DIMM Socket A8) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 151D2A36]
	ai-MEMORY9	OK	10-19-2021 11:43:37	19d 20h 8m 13s	1/4	OK - Memory 9 (DIMM Socket B1) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 3610257F]
	aj-MEMORY10	OK	10-19-2021 11:44:06	84d 19h 55m 37s	1/4	OK - Memory 10 (DIMM Socket B2) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 36202553]
	am-MEMORY11	OK	10-19-2021 11:44:52	84d 19h 58m 58s	1/4	OK - Memory 11 (DIMM Socket B3) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 36A0257E]
	an-MEMORY12	OK	10-19-2021 11:45:28	84d 19h 59m 31s	1/4	OK - Memory 12 (DIMM Socket B4) 16.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 36902569]
	ao-MEMORY13	OK	10-19-2021 11:46:06	84d 19h 59m 30s	1/4	OK - Memory 13 (DIMM Socket B5) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 158D2A3D]
	ap-MEMORY14	OK	10-19-2021 11:41:44	84d 19h 58m 23s	1/4	OK - Memory 14 (DIMM Socket B6) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 154D2A36]
	aq-MEMORY15	OK	10-19-2021 11:42:21	84d 19h 59m 30s	1/4	OK - Memory 15 (DIMM Socket B7) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 155D2A3D]
	ar-MEMORY16	OK	10-19-2021 11:43:00	19d 20h 12m 44s	1/4	OK - Memory 16 (DIMM Socket B8) 8.0 GB/1600 MHz: ENABLED/OK [DDR3, Hynix Semiconductor, S/N: 156D2A37]

Figura 20 – Schermata Nagios DELL mediante Nagios

2.2.5. Monitoraggio dei server HP

L'infrastruttura di calcolo dell'ICAR comprende per finire alcuni server HP, anch'essi monitorati con Nagios utilizzando il sistema di management out-of-band HP Integrated Lights Out (ILO). Come descritto

e schematizzato nei precedenti paragrafi, le attività compiute sono le medesime di seguito descritte brevemente, è stato aggiunto lo script `check_hp` [9] alla directory `"/usr/local/nagios/libexec/"`, è stato modificato il file `"/usr/local/nagios/etc/objects/commands.cfg"` aggiungendo il comando `check-hp` (Figura 21), è stato creato per ogni server da monitorare un file di configurazione (come quello mostrato in Figura 22). Infine, al file `nagios.cfg` sono stati aggiunti i path relativi ai file di configurazione dei diversi server HP da monitorare. In Figura 23 è mostrata una schermata di Nagios che riporta i parametri monitorati per uno dei server HP.

```
define command{
command_name    check-hp
command_line    perl $USER1$/check_hp/check_hp -H $HOSTADDRESS$
}
```

Figura 21 - Estratto file `usr/local/nagios/etc/objects/commands.cfg`

```
define host {
use                linux-server
host_name          iloInteromics20.84
address           [REDACTED].84
hostgroups        ILO
}

define service {
use                local-service
host_name          iloInteromics20.84
service_description PING
check_command      check_ping!100.0,20%!500.0,60%
notifications_enabled 0
}

define service {
use                local-service
host_name          iloInteromics20.84
service_description STATUS
check_command      check-hp
notifications_enabled 0
}
```

Figura 22 – Estratto file di configurazione `"/usr/local/nagios/etc/objects/ICAR/ilo20.84.cfg"`

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ilo78.23	PING	OK	10-19-2021 09:57:40	47d 22h 3m 39s	1/4	PING OK - Packet loss = 0%, RTA = 0.47 ms
	STATUS	CRITICAL	10-19-2021 09:58:18	47d 19h 56m 12s	4/4	Compaq/HP Agent Check: ProLiant ML350p Gen8 (tower) S/N CZ225102DC physical drive S.M.A.R.T status (0.4: replaceDrive) logical drive and associated physical state (0.1: degraded) array accelerator status (degraded) controller status (degraded) condition of the power supply (0.1: failed) overall condition of power supply subsystem (degraded) physical drive status (0.4: predictiveFailure) physical drive condition (0.4: degraded)
iloInteromics20.80	PING	OK	10-19-2021 09:58:55	66d 2h 57m 33s	1/4	PING OK - Packet loss = 0%, RTA = 0.49 ms
	STATUS	CRITICAL	10-19-2021 09:59:06	68d 15h 20m 21s	4/4	Compaq/HP Agent Check: ProLiant DL380e Gen8 (rack-mount) S/N CZ3320EA6A logical drive and associated physical state (2.1: degraded) controller status (degraded) physical drive status (2.2: failed) condition of the power supply (0.2: failed) array accelerator status (failed) overall condition of power supply subsystem (degraded) logical drive status (2.1: recovering) physical drive condition (2.2: failed) overall IML entries (failed)
iloInteromics20.81	PING	OK	10-19-2021 09:55:09	30d 23h 58m 52s	1/4	PING OK - Packet loss = 0%, RTA = 0.51 ms
	STATUS	CRITICAL	10-19-2021 09:55:47	96d 14h 44m 14s	4/4	Compaq/HP Agent Check: ProLiant DL360e Gen8 (rack-mount) S/N CZ3320E96D overall IML entries (failed) status of the processor fan(s) (failed) condition of the fan (0.7: degraded) condition of the fan (0.6: degraded)
iloInteromics20.82	PING	CRITICAL	10-19-2021 09:56:25	96d 21h 17m 53s	4/4	CRITICAL - Host Unreachable (192.168.20.82)
	STATUS	UNKNOWN	10-19-2021 09:57:03	96d 21h 18m 23s	4/4	Compaq/HP Agent Check: ERROR: No snmp response from 192.168.20.82 (alarm)
iloInteromics20.83	PING	OK	10-19-2021 09:57:40	82d 8h 31m 23s	1/4	PING OK - Packet loss = 0%, RTA = 0.59 ms
	STATUS	CRITICAL	10-19-2021 09:58:18	13d 17h 43m 44s	4/4	Compaq/HP Agent Check: ProLiant DL360e Gen8 (rack-mount) S/N CZ3320E969 condition of the power supply (0.1: failed) overall condition of power supply subsystem (degraded)
iloInteromics20.84	PING	OK	10-19-2021 09:58:56	37d 1h 33m 20s	1/4	PING OK - Packet loss = 0%, RTA = 0.57 ms
	STATUS	CRITICAL	10-19-2021 09:59:06	68d 15h 14m 42s	4/4	Compaq/HP Agent Check: ProLiant DL360e Gen8 (rack-mount) S/N CZ3320E967 status of the processor fan(s) (degraded) condition of the fan (0.1: degraded) overall condition of power supply subsystem (degraded) condition of the power supply (0.1: failed)
iloInteromics20.85	PING	OK	10-19-2021 09:55:10	83d 1h 32m 22s	1/4	PING OK - Packet loss = 0%, RTA = 0.54 ms
	STATUS	OK	10-19-2021 09:55:48	84d 20h 59m 48s	1/4	Compaq/HP Agent Check: ProLiant DL360e Gen8 (rack-mount) S/N CZJ41405QW overall system state OK

Figura 23 – Schermata Nagios monitoraggio server HP

2.2.6. Nagios Remote Plugin Executor

Per avere un controllo puntuale sullo stato dei servizi e l'utilizzo delle risorse di alcune VM di particolare rilevanza, è stato installato il plugin NRPE [10], che consente di eseguire comandi su macchine Linux/Unix remote e verificarne l'output direttamente dall'interfaccia di Nagios. Nella Figura 24 è riportato uno schema di funzionamento del plugin NRPE e l'interazione con l'applicazione Nagios installata sul server di monitoraggio.

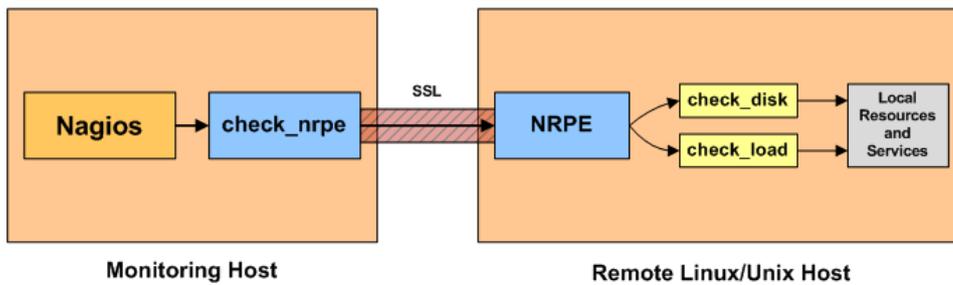


Figura 24 - Schema di funzionamento NRPE

Il plugin è stato configurato sia lato monitoring host che lato remote linux host. Sulla macchina remota sono stati installati i pacchetti software nagios-nrpe-server e nagios-plugins-basic, è stato modificato il file /etc/nagios/nrpe.d/op5_commands.cfg così come riportato in Figura 25. In particolare, considerando la necessità di controllare il corretto avvio dei servizi del sistema operativo, sulla macchina remota, è stato aggiunto un comando “check_processlistfailed”, che una volta eseguito, fornisce in output la lista di processi non avviati correttamente.

```
#####
#
# op5-nrpe command configuration file
#
# COMMAND DEFINITIONS
# Syntax:
# command[<command_name>]=<command_line>
#
command[users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[swap]=/usr/lib/nagios/plugins/check_swap -w 20% -c 10%
command[root_disk]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p / -m
command[usr_disk]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /usr -m
command[var_disk]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /var -m
command[zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[total_procs]=/usr/lib/nagios/plugins/check_procs -w 190 -c 200
command[proc_named]=/usr/lib/nagios/plugins/check_procs -w 1: -c 1:2 -C named
command[proc_cron]=/usr/lib/nagios/plugins/check_procs -w 1: -c 1:5 -C cron
command[proc_syslogd]=/usr/lib/nagios/plugins/check_procs -w 1: -c 1:2 -C syslog-ng
command[proc_rsyslogd]=/usr/lib/nagios/plugins/check_procs -w 1: -c 1:2 -C rsyslogd
command[check_processlistfailed]=systemctl list-units --state=failed
```

Figura 25 - Estratto file di configurazione /etc/nagios/nrpe.d/op5_commands.cfg

L’ultimo step, per quanto riguarda la configurazione della macchina remote linux host, ha previsto la modifica del file /etc/nagios/nrpe.cfg sulla parte relativa agli indirizzi ip delle macchine che possono collegarsi mediante nrpe (come mostrato in Figura 26).

```
## you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1, .232
```

Figura 26 - Estratto file di configurazione /etc/nagios/nrpe.cfg

Sul server di monitoraggio è stato installato il pacchetto software nrpe-nrpe-4.0.3 e aggiornato il file /etc/services aggiungendo il servizio nagios nrpe così come riportato in Figura 27.

```
noclog 5354/tcp # noclogd with TCP (nocol)
noclog 5354/udp # noclogd with UDP (nocol)
hostmon 5355/tcp # hostmon uses TCP (nocol)
hostmon 5355/udp # hostmon uses UDP (nocol)
rplay 5555/udp # RPlay audio service
nrpe 5666/tcp # Nagios Remote Plugin Executor
nsca 5667/tcp # Nagios Agent - NSCA
mrttd 5674/tcp # MRT Routing Daemon
bgpsim 5675/tcp # MRT Routing Simulator
canna 5680/tcp # cannaserver
syslog-tls 6514/tcp # Syslog over TLS [RFC5425]
```

Figura 27 - Estratto file di configurazione /etc/services

Al file /usr/local/nagios/etc/objects/command.cfg (Figura 28) sono stati aggiunti due comandi. Il primo comando riguarda la verifica del corretto avvio dei servizi del sistema operativo, mentre il secondo riguarda lo spazio disponibile sul disco del remote linux host.

```

define command {
command_name check-process
command_line bash $USER1$/check_process.sh $HOSTADDRESS$
}

define command {
command_name check-disk
command_line /usr/local/nagios/libexec/check_nrpe -H $HOSTADDRESS$ -c root_disk
}

```

Figura 28 – Estratto file di configurazione /usr/local/nagios/etc/objects/command.cfg

A titolo esemplificativo si riporta un estratto del file di configurazione /usr/local/nagios/etc/objects/ICAR/test.cfg (Figura 29) dove sono richiamati i comandi “check-disk” e “check-process”.

```

define service {
use local-service
host_name test14.212
service_description checkProcess
check_command check-process
notifications_enabled 0
}

define service {
use local-service
host_name test14.212
service_description checkDisk
check_command check-disk
notifications_enabled 0
}

```

Figura 29 – Estratto file di configurazione /usr/local/nagios/etc/objects/ICAR/test.cfg

Il file check_process.sh (Appendice B) è stato definito dagli autori per il controllo del corretto avvio dei servizi del sistema operativo Linux. In Figura 30 è riportato un estratto dello script, che richiama il comando check_processlistfailed sul remote linux host, mediante il plugin nrpe, analizza l’output restituito dal comando e fornisce il risultato a nagios.

```

#!/bin/bash
# check_process.sh
# Description: check systemctl list-units --state=failed
# @Autor: Angelo Esposito CNR, angelo.esposito@icar.cnr.it
HOST=$1
STATUS=""
SYSTEM=""
LISTA=""

COMANDOTEMP="tempfile"
FILENAME=${COMANDOTEMP}
FILENAME2=${COMANDOTEMP}
COMANDO="/usr/local/nagios/libexec/check_nrpe -H $HOST -c check_processlistfailed | grep '^0' >> $FILENAME"
COMANDOLIST="/usr/local/nagios/libexec/check_nrpe -H $HOST -c check_processlistfailed >> $FILENAME2"
eval $COMANDO
eval $COMANDOLIST

while IFS=' ' read -r line || [[ -n "$line" ]]; do
    if [ "$line" == "0 loaded units listed." ] || [ "$line" == "0 loaded units listed. Pass --all to see loaded but inactive units, too." ]
    then
        SYSTEM="OK"
    fi
done < "$FILENAME"

LISTA=""

while IFS=' ' read -r a b c d e || [[ -n "$line" ]]; do
    LISTA="$LISTA\n$a"
done < "$FILENAME2"

CRITICAL=0
WARNING=0
UNKNOWN=0

if [ [ "$SYSTEM" == "OK" ] ]
then
    STATUS="OK"
else
    STATUS="$STATUS PROCESS FAILED $LISTA"
    WARNING=1
fi

if [ $CRITICAL == 1 ]
then
    echo "CRITICAL \n$STATUS"
    exit 2
fi

if [ $WARNING == 1 ]
then
    echo "WARNING \n$STATUS"
    exit 1
fi

```

Figura 30 – Estratto file script “check_process.sh”

In Figura 31 è mostrata la schermata di Nagios che riporta le metriche relative ad uno dei server remoti controllati mediante plugin NRPE. In questo caso specifico è possibile notare che il servizio unattended-upgrades non è avviato correttamente.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
test14.212	PING	OK	11-18-2021 15:59:40	3d 4h 36m 37s	1/4	PING OK - Packet loss = 0%, RTA = 0.64 ms
	checkDisk	OK	11-18-2021 16:00:17	71d 4h 43m 7s	1/4	DISK OK - free space: / 49435 MB (82% inode=95%):
	checkProcess	WARNING	11-18-2021 16:02:54	0d 0h 0m 8s	1/4	WARNING PROCESS FAILED UNIT LOAD ACTIVE SUB DESCRIPTION • unattended-upgrades service loaded failed failed Unattended Upgrades Shutdown LOAD = Reflects whether the unit definition was properly loaded. ACTIVE = The high-level unit activation state, i.e. generalization of SUB. SUB = The low-level unit activation state, values depend on unit type. 1 loaded units listed

Figura 31 – Schermata Nagios visualizzazione metriche remote linux host

2.3. Rsyslog

Rsyslog [11] è un software open source per la gestione dei messaggi di log via rete per sistemi UNIX che implementa il protocollo syslog e lo estende con funzionalità avanzate. Sul server di monitoring è stato configurato un servizio Rsyslog che raccoglie i log inviati dalle apparecchiature della rete dell'ICAR. Questa configurazione consente agli amministratori di rete ICAR di avere a disposizione in un unico punto centralizzato tutti i log degli apparati di rete e nel caso di malfunzionamenti di poter mettere in relazione gli avvenimenti occorsi sui diversi dispositivi.

L'installazione del servizio sul server di monitoring ha previsto la modifica del file di configurazione "/etc/rsyslog.conf" per consentire la ricezione dei log inviati dagli apparati di rete. In Figura 32 è riportato un estratto del file di configurazione del servizio rsyslog dove è stato specificato da quali subnet accettare i log.

```

14
15 $template Incoming-logs, "/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
16 *.* ?Incoming-logs
17
18
19
20 #####
21 #### RULES ####
22 #####
23
24
25 $AllowedSender TCP, 127.0.0.1, .0/8, .0/8, .0/8
26
27 #

```

Figura 32 - Estratto file di configurazione Rsyslog

Gli switch Juniper della rete ICAR sono stati opportunamente configurati per inviare i log al server di monitoring. In Figura 33 è riportato un estratto del file di configurazione degli switch, dove è specificato l'indirizzo ip del server rsyslog e le tipologie di messaggi log da inviare.

```

syslog {
  user * {
    any emergency;
  }
  host .232 {
    any any;
  }
  file messages {
    any notice;
    authorization info;
  }
  file interactive-commands {
    interactive-commands any;
  }
}

```

Figura 33 - Estratto file di configurazione switch juniper

Allo scopo di ottimizzare la memorizzazione dei log è stato opportunamente configurato l'utility "logrotate" che facilita la gestione dell'elevato numero dei file di log inviati dagli switch, consentendo la

loro rotazione, compressione, rimozione e invio tramite email (in Figura 34 è riportato un estratto del file di configurazione dell'utility logrotate).

```
/var/log/ji.17Ed4_p2/*.log
/var/log/ji.192Ed4_p0/*.log
/var/log/ji.216Ed4_p-1/*.log
/var/log/ji.233Ed4_p1/*.log
/var/log/ji.234Ed4_p2/*.log
/var/log/ji.236Ed4_p-1/*.log
/var/log/ji.237Ed4_p1/*.log
/var/log/ji.238Ed4_p2/*.log
/var/log/ji.247Ed4_p0/*.log
/var/log/ji.250Ed4_p2/*.log
{
daily
delaycompress
rotate 1
compress
missingok
}
```

Figura 34 - Estratto file di configurazione /etc/logrotate.d/juniper

2.4. MRTG

Multi Router Traffic Grapher (MRTG) [12] è un software libero disponibile con licenza GPL per il monitoraggio e la misurazione del carico di collegamenti di rete. Consente di visualizzare il traffico di rete nel tempo in forma grafica. Questo applicativo viene utilizzato per la sua duttilità e per la facilità di utilizzo nel controllo degli apparati di rete come, ad esempio, i router, inoltre è in grado di operare sia su sistemi operativi Unix-like che Microsoft.

Sul server di monitoraggio è stato installato il software MGRT per monitorare il traffico della rete ICAR-CNR, in particolare è stato modificato il file di configurazione "/etc/mrtg.cfg" andando a specificare tutti gli apparati di rete da monitorare, in Figura 35 è mostrato un estratto del file di configurazione per il servizio MRTG.

```

\uvv/
### Interface 509 >> Descr: 'ge-0/0/8' | Name: 'ge-0/0/8' | Ip: 'No Ip' | Eth: 'No Ethernet Id' ###

Target[      .1_ge-0_0_8]: \ge-0/0/8:a486053@      .1::::2
SetEnv[      .1_ge-0_0_8]: MRTG_INT_IP="No Ip" MRTG_INT_DESCR="ge-0/0/8"
MaxBytes[    .1_ge-0_0_8]: 125000000
Title[       .1_ge-0_0_8]: ge-0/0/8 --
PageTop[     .1_ge-0_0_8]: <h1>ge-0/0/8 ip=      jicar-1-basso</h1>
              <div id="sysdetails">
                <table>
                  <tr>
                    <td>System:</td>
                    <td>CNR-Juniper-CentroStella in </td>
                  </tr>
                  <tr>
                    <td>Maintainer:</td>
                    <td></td>
                  </tr>
                  <tr>
                    <td>Description:</td>
                    <td>ge-0/0/8 Icar Piano -1 </td>
                  </tr>
                  <tr>
                    <td>ifType:</td>
                    <td>ethernetCsmacd (6)</td>
                  </tr>
                  <tr>
                    <td>ifName:</td>
                    <td>ge-0/0/8</td>
                  </tr>
                  <tr>
                    <td>Max Speed:</td>
                    <td>125.0 MBytes/s</td>
                  </tr>
                  <tr>
                    <td>Ip:</td>
                    <td>No Ip (No DNS name)</td>
                  </tr>
                </table>
              </div>

```

Figura 35 - Estratto file di configurazione servizio MGRT

In Figura 36 è riportata una parte della pagina web MRTG con i grafici che rappresentano il traffico di rete nel tempo differenziando lo stesso sulle diverse interfacce di rete da monitorare.

MRTG

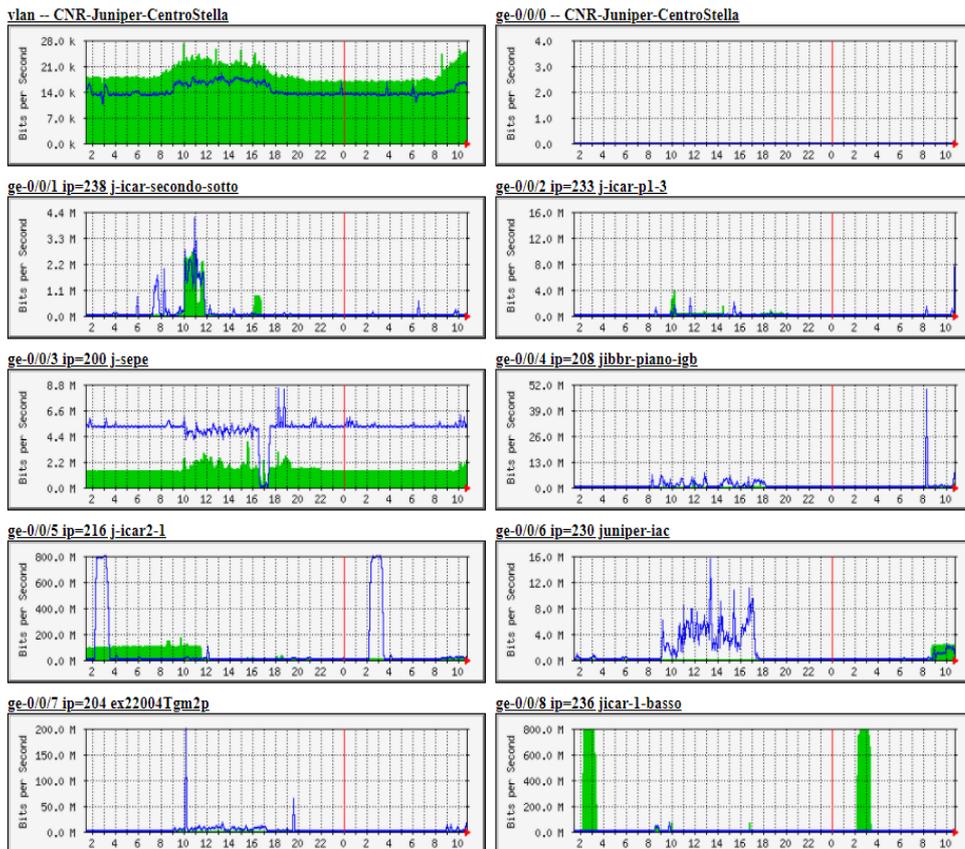


Figura 36 - Pagina web MGRT

2.5. Trivial File Transfer Protocol

Il Trivial File Transfer Protocol (TFTP) [13] è un protocollo di trasferimento file di livello applicativo molto semplice, che fornisce le funzionalità di base del protocollo FTP. Il TFTP viene utilizzato principalmente nelle fasi di boot via rete ed è supportato dalla maggior parte dei dispositivi di rete per l'upload e il download di file per la sua semplicità. Sul server di monitoraggio è stato configurato per la ricezione dei file di configurazione degli switch Juniper della rete ICAR per effettuare il backup degli stessi. In Figura 37 è riportato un estratto del file di configurazione del servizio "/etc/default/tftpd-hpa".

```
TFTP_USERNAME="tftp"  
TFTP_DIRECTORY="/backup/network/"  
TFTP_ADDRESS="0.0.0.0:69"  
TFTP_OPTIONS="--secure -c -U 0022 -p"
```

Figura 37 - Estratto file di configurazione del servizio tftpd-hpa

Lo script riportato in Figura 38 è avviato una volta al giorno mediante crontab sugli switch della rete ICAR. Lo script è programmato per effettuare il collegamento con il server di monitoraggio via TFTP ed inviare gli ultimi quattro file di configurazione dello switch. Il server di monitoraggio, una volta ricevuti i file di configurazione da parte degli switch, li memorizza in una cartella specifica sincronizzata in cloud.

```
tftp .232 <<'EOF'  
put /config/juniper.conf.gz juniper. .17.gz  
put /config/juniper.conf.1.gz juniper. .17.gz  
put /config/juniper.conf.2.gz juniper. .17.gz  
put /config/juniper.conf.3.gz juniper. .17.gz  
quit  
EOF
```

Figura 38 - Script backup.sh

3. Primi risultati

Il server di monitoraggio, attivo da circa tre mesi, ha permesso l'individuazione di una errata configurazione del protocollo Spanning-Tree [14] sugli switch della rete Area di Ricerca NA 1. In particolare, avendo la possibilità di valutare nel complesso i log di tutti gli apparati di rete in un unico punto centralizzato, grazie all'utilizzo del software Rsyslog, è stato possibile rilevare un comportamento anomalo nella gestione del protocollo spanning-tree, da parte degli switch. Il comportamento anomalo riguardava il continuo cambio di root-bridge da parte di tutti gli switch periferici, che causava rallentamenti ed irraggiungibilità degli switch dell'infrastruttura di rete. Dopo un'analisi accurata della configurazione degli switch del centro stella, gli autori del presente RT hanno notato la presenza di interfacce su cui erano attivi simultaneamente i protocolli di spanning tree VSTP and RSTP. La problematica riscontrata è stata risolta attivando solo uno dei due protocolli di spanning-tree sugli switch del centro stella.

4. Conclusioni

L'attività svolta e descritta in questo RT ha riguardato la predisposizione e configurazione di un sistema di monitoraggio centralizzato degli apparati di rete e di calcolo dell'ICAR sede di Napoli, con l'obiettivo di prevenire eventuali malfunzionamenti ed intervenire in maniera efficace ed efficiente nel caso questi avvengano. Il sistema di monitoraggio nel suo complesso è costituito da un insieme di software installati e opportunamente configurati su una macchina server per il monitoraggio centralizzato di tutti gli apparati di rete e di calcolo ad alte prestazioni dell'ICAR. Il sistema consente una visione globale dello stato dei servizi e delle risorse in tempo reale, ed in caso di malfunzionamenti invia opportune notifiche agli amministratori di sistema. Il sistema di monitoraggio ha già permesso l'individuazione di una problematica relativa agli switch di rete, che gli autori del presente RT hanno risolto aggiornando la configurazione degli switch del centro stella.

Le attività future di maggiore interesse, in continuità con il presente RT, riguardano la configurazione di sistemi in grado di processare i log automaticamente allo scopo di anticipare temporalmente eventuali malfunzionamenti degli apparati di rete/server/servizi.

Un tool molto interessante, che potrebbe essere configurato sul server di monitoraggio, è Elasticsearch [15], che è un motore di ricerca e analisi distribuito e open source per tutti i tipi di dati, inclusi testuali, numerici, geospaziali, strutturati e non strutturati. In particolare, potrebbe essere configurato Logstash, che è un modulo di Elastic Stack per aggregare ed elaborare i log degli apparati di rete/server/servizi ICAR ed inviarli a Elasticsearch per l'analisi automatica.

5. Appendice A – check_ipmi.sh

```
#!/bin/bash
# check_ipmi.sh
# Descripcion: check bmc ibm via ipmi
# @Author: Miguel Andrés Caballero mac_121@hotmail.com
# @originalscriptlink:
#https://exchange.nagios.org/directory/Addons/Active-Checks/check_ipmi-2Esh/
details
# @Modified by Authors: Angelo Esposito e Gennaro Oliva CNR,
angelo.esposito@icar.cnr.it, gennaro.oliva@icar.cnr.it

USER=root
HOST=$1
RACK=$2

COMANDOTEMP="tempfile"
FILENAME=$(COMANDOTEMP)

if [ $RACK = 1 ]; then
PASSWORD=XXXXXXXX
COMANDO="ipmitool -I lanplus -H $HOST -P $PASSWORD sdr >> $FILENAME"
eval $COMANDO
COMANDOPOWER="ipmitool -I lanplus -H $HOST -P $PASSWORD chassis status"
fi

if [ $RACK = 3 ]; then
PASSWORD=XXXXXXXX
COMANDO="ipmitool -I lanplus -H $HOST -U $USER -P $PASSWORD sdr >> $FILENAME"
eval $COMANDO
COMANDOPOWER="ipmitool -I lanplus -U $USER -H $HOST -P $PASSWORD chassis
status"
fi

CPU2_STATUS=$(cat $FILENAME | grep -w "cpu1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
CPU2_TEMP=$(cat $FILENAME | grep "^p0_core0_temp" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
CPU1_STATUS=$(cat $FILENAME | grep -w "cpu0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
CPU1_TEMP=$(cat $FILENAME | grep "^p0_core0_temp" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN1_RPM=$(cat $FILENAME | grep -w "fan0_0" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN1_STATUS=$(cat $FILENAME | grep -w "fan0_0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN2_RPM=$(cat $FILENAME | grep -w "fan0_1" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN2_STATUS=$(cat $FILENAME | grep -w "fan0_1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN3_RPM=$(cat $FILENAME | grep -w "fan1_0" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN3_STATUS=$(cat $FILENAME | grep -w "fan1_0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN4_RPM=$(cat $FILENAME | grep -w "fan1_1" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN4_STATUS=$(cat $FILENAME | grep -w "fan1_1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN5_RPM=$(cat $FILENAME | grep -w "fan2_0" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
FAN5_STATUS=$(cat $FILENAME | grep -w "fan2_0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*$//')
```

```

FAN6_RPM=$(cat $FILENAME | grep -w "fan2_1" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
FAN6_STATUS=$(cat $FILENAME | grep -w "fan2_1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
FAN7_RPM=$(cat $FILENAME | grep -w "fan3_0" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
FAN7_STATUS=$(cat $FILENAME | grep -w "fan3_0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
FAN8_RPM=$(cat $FILENAME | grep -w "fan3_1" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
FAN8_STATUS=$(cat $FILENAME | grep -w "fan3_1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
POWER0_WATT=$(cat $FILENAME | grep -w "ps0_input_power" | cut -d "|" -f 2 |
sed 's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
POWER0_STATUS=$(cat $FILENAME | grep -w "ps0_input_power" | cut -d "|" -f 3 |
sed 's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
POWER1_WATT=$(cat $FILENAME | grep -w "ps1_input_power" | cut -d "|" -f 2 |
sed 's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
POWER1_STATUS=$(cat $FILENAME | grep -w "ps1_input_power" | cut -d "|" -f 3 |
sed 's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
TEMP_GRA=$(cat $FILENAME | grep -w "ambient" | cut -d "|" -f 2 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
TEMP_STATUS=$(cat $FILENAME | grep -w "ambient" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM0_STATUS=$(cat $FILENAME | grep -w "dimm0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM1_STATUS=$(cat $FILENAME | grep -w "dimm1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM2_STATUS=$(cat $FILENAME | grep -w "dimm2" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM3_STATUS=$(cat $FILENAME | grep -w "dimm3" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM4_STATUS=$(cat $FILENAME | grep -w "dimm4" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM5_STATUS=$(cat $FILENAME | grep -w "dimm5" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM6_STATUS=$(cat $FILENAME | grep -w "dimm6" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM7_STATUS=$(cat $FILENAME | grep -w "dimm7" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM8_STATUS=$(cat $FILENAME | grep -w "dimm8" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM9_STATUS=$(cat $FILENAME | grep -w "dimm9" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM10_STATUS=$(cat $FILENAME | grep -w "dimm10" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM11_STATUS=$(cat $FILENAME | grep -w "dimm11" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM12_STATUS=$(cat $FILENAME | grep -w "dimm12" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM13_STATUS=$(cat $FILENAME | grep -w "dimm14" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM14_STATUS=$(cat $FILENAME | grep -w "dimm14" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
DIMM15_STATUS=$(cat $FILENAME | grep -w "dimm15" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
GV0_STATUS=$(cat $FILENAME | grep -w "gv100card0" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
GV1_STATUS=$(cat $FILENAME | grep -w "gv100card1" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
GV3_STATUS=$(cat $FILENAME | grep -w "gv100card3" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')
GV4_STATUS=$(cat $FILENAME | grep -w "gv100card4" | cut -d "|" -f 3 | sed
's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')

```

```

SYSTEM_POWER=$(($COMANDOPOWER chassis status | grep "System Power" | cut -d
":" -f 2 | sed 's/[[:space:]]*$//') | sed 's/^[[:space:]]*//')

if [ -f "$FILENAME" ] ; then
    rm "$FILENAME"
fi

STATUS=""
CRITICAL=0
WARNING=0
UNKNOWN=0

if [[ "$SYSTEM_POWER" == "on" ]]
then
    STATUS="Power On"

    if [ $CPU1_STATUS == "ok" ]
    then
        STATUS="$STATUS \nCPU1 Temp OK ( $CPU1_TEMP )"
    else
        STATUS="$STATUS \nCPU1 Temp CRITICAL ( $CPU1_TEMP )"
        CRITICAL=1
    fi

    if [ $CPU2_STATUS == "ok" ]
    then
        STATUS="$STATUS \nCPU2 Temp OK ( $CPU2_TEMP )"
    else
        STATUS="$STATUS \nCPU2 Temp CRITICAL ( $CPU2_TEMP )"
        CRITICAL=1
    fi

    if [ $FAN1_STATUS == "ok" ]
    then
        STATUS="$STATUS \nFAN1 OK ( $FAN1_RPM )"
    else
        STATUS="$STATUS \nFAN1 WARNING ( $FAN1_RPM )"
        WARNING=1
    fi

    if [ $FAN2_STATUS == "ok" ]
    then
        STATUS="$STATUS \nFAN2 OK ( $FAN2_RPM )"
    else
        STATUS="$STATUS \nFAN2 WARNING ( $FAN2_RPM )"
        WARNING=1
    fi

    if [ $FAN3_STATUS == "ok" ]
    then
        STATUS="$STATUS \nFAN3 OK ( $FAN3_RPM )"
    else
        STATUS="$STATUS \nFAN3 WARNING ( $FAN3_RPM )"
        WARNING=1
    fi

    if [ $FAN4_STATUS == "ok" ]
    then
        STATUS="$STATUS \nFAN4 OK ( $FAN4_RPM )"
    else
        STATUS="$STATUS \nFAN4 WARNING ( $FAN4_RPM )"
        WARNING=1
    fi

    if [ $FAN5_STATUS == "ok" ]
    then
        STATUS="$STATUS \nFAN5 OK ( $FAN5_RPM )"

```

```

else
    STATUS="$STATUS \nFAN5 WARNING ( $FAN5_RPM )"
    WARNING=1
fi
if [ $FAN6_STATUS == "ok" ]
then
    STATUS="$STATUS \nFAN6 OK ( $FAN6_RPM )"
else
    STATUS="$STATUS \nFAN6 WARNING ( $FAN6_RPM )"
    WARNING=1
fi
if [ $FAN7_STATUS == "ok" ]
then
    STATUS="$STATUS \nFAN7 OK ( $FAN7_RPM )"
else
    STATUS="$STATUS \nFAN7 WARNING ( $FAN7_RPM )"
    WARNING=1
fi
if [ $FAN8_STATUS == "ok" ]
then
    STATUS="$STATUS \nFAN8 OK ( $FAN8_RPM )"
else
    STATUS="$STATUS \nFAN8 WARNING ( $FAN8_RPM )"
    WARNING=1
fi
if [ $POWER0_STATUS == "ok" ]
then
    STATUS="$STATUS \nPOWER0 OK ( $POWER0_WATT )"
else
    STATUS="$STATUS \nPOWER0 WARNING ( $POWER0_WATT )"
    WARNING=1
fi
if [ $POWER1_STATUS == "ok" ]
then
    STATUS="$STATUS \nPOWER1 OK ( $POWER1_WATT )"
else
    STATUS="$STATUS \nPOWER1 WARNING ( $POWER1_WATT )"
    WARNING=1
fi
if [ $TEMP_STATUS == "ok" ]
then
    STATUS="$STATUS \nTemperature OK ( $TEMP_GRA )"
else
    STATUS="$STATUS \nTemperature WARNING ( $TEMP_GRA )"
    WARNING=1
fi
if [ $DIMM0_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM0 OK ( $DIMM0_STATUS )"
else
    STATUS="$STATUS \nDIMM0 WARNING ( $DIMM0_STATUS )"
    WARNING=1
fi
if [ $DIMM1_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM1 OK ( $DIMM1_STATUS )"
else
    STATUS="$STATUS \nDIMM1 WARNING ( $DIMM1_STATUS )"
    WARNING=1
fi
if [ $DIMM2_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM2 OK ( $DIMM2_STATUS )"

```

```

else
    STATUS="$STATUS \nDIMM2 WARNING ( $DIMM2_STATUS )"
    WARNING=1
fi

    if [ $DIMM3_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM3 OK ( $DIMM3_STATUS )"
else
    STATUS="$STATUS \nDIMM3 WARNING ( $DIMM3_STATUS )"
    WARNING=1
fi

    if [ $DIMM4_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM4 OK ( $DIMM4_STATUS )"
else
    STATUS="$STATUS \nDIMM4 WARNING ( $DIMM4_STATUS )"
    WARNING=1
fi

    if [ $DIMM5_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM5 OK ( $DIMM5_STATUS )"
else
    STATUS="$STATUS \nDIMM5 WARNING ( $DIMM5_STATUS )"
    WARNING=1
fi

    if [ $DIMM6_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM6 OK ( $DIMM6_STATUS )"
else
    STATUS="$STATUS \nDIMM6 WARNING ( $DIMM6_STATUS )"
    WARNING=1
fi
if [ $DIMM7_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM7 OK ( $DIMM7_STATUS )"
else
    STATUS="$STATUS \nDIMM7 WARNING ( $DIMM7_STATUS )"
    WARNING=1
fi
    if [ $DIMM8_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM8 OK ( $DIMM8_STATUS )"
else
    STATUS="$STATUS \nDIMM8 WARNING ( $DIMM8_STATUS )"
    WARNING=1
fi
    if [ $DIMM9_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM9 OK ( $DIMM9_STATUS )"
else
    STATUS="$STATUS \nDIMM9 WARNING ( $DIMM9_STATUS )"
    WARNING=1
fi

    if [ $DIMM10_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM10 OK ( $DIMM10_STATUS )"
else
    STATUS="$STATUS \nDIMM10 WARNING ( $DIMM10_STATUS )"
    WARNING=1
fi

    if [ $DIMM11_STATUS == "ok" ]

```

```

then
    STATUS="$STATUS \nDIMM11 OK ( $DIMM11_STATUS )"
else
    STATUS="$STATUS \nDIMM11 WARNING ( $DIMM11_STATUS )"
    WARNING=1
fi

    if [ $DIMM12_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM12 OK ( $DIMM12_STATUS )"
else
    STATUS="$STATUS \nDIMM12 WARNING ( $DIMM12_STATUS )"
    WARNING=1
fi

    if [ $DIMM13_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM13 OK ( $DIMM13_STATUS )"
else
    STATUS="$STATUS \nDIMM13 WARNING ( $DIMM13_STATUS )"
    WARNING=1
fi

    if [ $DIMM14_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM14 OK ( $DIMM14_STATUS )"
else
    STATUS="$STATUS \nDIMM14 WARNING ( $DIMM14_STATUS )"
    WARNING=1
fi

    if [ $DIMM15_STATUS == "ok" ]
then
    STATUS="$STATUS \nDIMM15 OK ( $DIMM15_STATUS )"
else
    STATUS="$STATUS \nDIMM15 WARNING ( $DIMM15_STATUS )"
    WARNING=1
fi

    if [ $GV0_STATUS == "ok" ]
then
    STATUS="$STATUS \nGV0_STATUS OK ( $GV0_STATUS )"
else
    STATUS="$STATUS \nGV0_STATUS WARNING ( $GV0_STATUS )"
    WARNING=1
fi

    if [ $GV1_STATUS == "ok" ]
then
    STATUS="$STATUS \nGV1_STATUS OK ( $GV1_STATUS )"
else
    STATUS="$STATUS \nGV1_STATUS WARNING ( $GV1_STATUS )"
    WARNING=1
fi

    if [ $GV3_STATUS == "ok" ]
then
    STATUS="$STATUS \nGV3_STATUS OK ( $GV3_STATUS )"
else
    STATUS="$STATUS \nGV3_STATUS WARNING ( $GV3_STATUS )"
    WARNING=1
fi

    if [ $GV4_STATUS == "ok" ]
then
    STATUS="$STATUS \nGV4_STATUS OK ( $GV4_STATUS )"
else

```

```
                STATUS="$STATUS \nGV4_STATUS WARNING ( $GV4_STATUS )"
                WARNING=1
            fi

else
    STATUS="SYSTEM POWER OFF"
    CRITICAL=1
fi

if [ $CRITICAL == 1 ]
    then
        echo "CRITICAL \n$STATUS"
        exit 2
fi

if [ $WARNING == 1 ]
    then
        echo "WARNING \n$STATUS"
        exit 1
fi

if [ $UNKNOWN == 1 ]
    then
        echo "UNKNOWN \n$STATUS"
        exit 3
fi

echo "OK \n$STATUS"
exit 0
```

6. Appendice B – check_process.sh

```
#!/bin/bash
# check_process.sh
# Descrpcion: check systemctl list-units --state=failed
# @Authors: Angelo Esposito e Gennaro Oliva CNR, angelo.esposito@icar.cnr.it
gennaro.oliva@icar.cnr.it

HOST=$1
STATUS=""
SYSTEM=""
LISTA=""

COMANDOTEMP="tempfile"

FILENAME=$(($COMANDOTEMP)
FILENAME2=$(($COMANDOTEMP)

COMANDO="/usr/local/nagios/libexec/check_nrpe -H $HOST -c
check_processlistfailed | grep '^0' >> $FILENAME"
COMANDOLIST="/usr/local/nagios/libexec/check_nrpe -H $HOST -c
check_processlistfailed >> $FILENAME2"

eval $COMANDO
eval $COMANDOLIST

while IFS=' ' read -r line || [[ -n "$line" ]]; do
    if [ "$line" == "0 loaded units listed." ] || [ "$line" == "0 loaded
units listed. Pass --all to see loaded but inactive units, too." ]
    then
        SYSTEM="OK"
    fi
done < "$FILENAME"

LISTA=""

while IFS=' ' read -r a b c d e || [[ -n "$line" ]]; do

LISTA="$LISTA\n $a"

done < "$FILENAME2"

CRITICAL=0
WARNING=0
UNKNOWN=0

if [[ "$SYSTEM" == "OK" ]]
then
    STATUS="OK"
else
    STATUS="$STATUS PROCESS FAILED $LISTA"
    WARNING=1
fi

if [ $CRITICAL == 1 ]
then
    echo "CRITICAL \n$STATUS"
    exit 2
fi

if [ $WARNING == 1 ]
then
    echo "WARNING \n$STATUS"
```

```

        exit 1
fi

if [ $UNKNOWN == 1 ]
then
    echo "UNKNOWN \n$STATUS"
    exit 3
fi

if [ -f "$FILENAME" ] ; then
    rm "$FILENAME"
fi

if [ -f "$FILENAME2" ] ; then
    rm "$FILENAME2"
fi

echo "$STATUS"

exit 0

```

7. Riferimenti

- [1] “Infrastruttura di Rete dell’ICAR-CNR Sede di Napoli” - Angelo Esposito, Gennaro Oliva - RT-ICAR-NA-2020-05
- [2] Sito web: www.icar.cnr.it
- [3] Sito web: www.cnr.it
- [4] Sito web: www.diiit.cnr.it
- [5] Standard - Network Time Protocol: <https://datatracker.ietf.org/doc/html/rfc5905>
- [6] Applicativo Chrony: <https://chrony.tuxfamily.org>
- [7] Applicativo Nagios: <https://www.nagios.com/>
- [8] Script Nagios Idrac: https://github.com/dangmocrang/check_idrac
- [9] Script Nagios ILO: https://github.com/lairsdragon/check_hp/blob/master/check_hp
- [10] Plugin Nagios NRPE: <https://exchange.nagios.org/directory/Addons/Monitoring-Agents/NRPE--2D-Nagios-Remote-Plugin-Executor/details>
- [11] Applicativo Rsyslog: <https://www.rsyslog.com/>
- [12] Applicativo MRTG: <https://oss.oetiker.ch/mrtg/>
- [13] Standard - Trivial File Transfer Protocol : <https://datatracker.ietf.org/doc/html/rfc1350>
- [14] Standard - Spanning-Tree: https://standards.ieee.org/standard/802_1D-2004.html
- [15] Applicazione - Elasticsearch: <https://www.elastic.co/elasticsearch/>