WORKSHOP GARR 2026

NET MAKERS

Governo della sicurezza

Un’esperienza di evoluzione dal Cloud al Locale

Enrico Venuto
Davide Stevano
Politecnico di Torino

Riassunto delle puntate precedenti
Collaborazione con Università GDA

Progetto SIEM x IdP

Setup del sistema di log ingestion con sistema multi-coda modulare a microservizi.

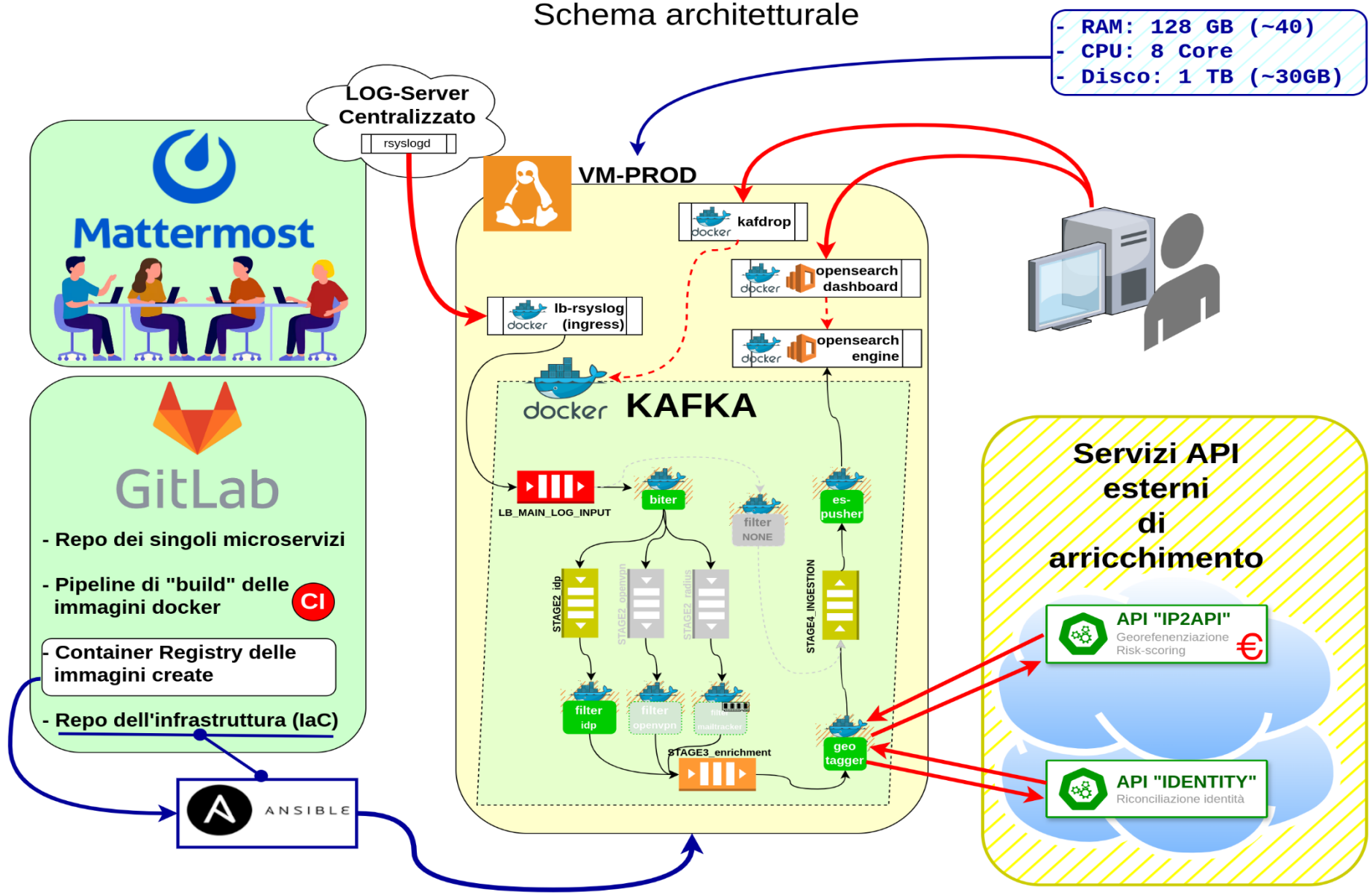
Automazione della creazione dei microservizi (CI) e del relativo deploy con ansible (CD)

Setup Infrastruttura di ingestion dei log dell'IdP provenienti dal syslog

- filtrare gli eventi
- arricchire i log con altri dati
- inserire dati in OpenSearch
- realizzare cruscotti per la visualizzazione dati.

LOG Biter @ PoliTO

Schema architetturale



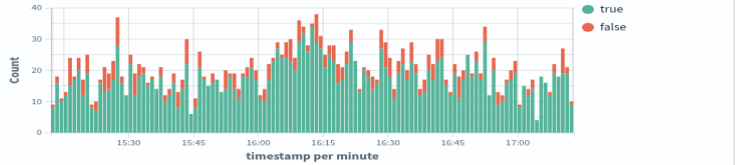
[AUTH-*] [numeri] auth TOT

2,471

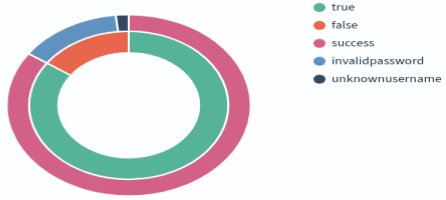
[AUTH-*] [numeri] auth_success / auth_fail

2,092	379
true - Count	false - Count

[AUTH-*] [Istogramma] auth success/fail > Time



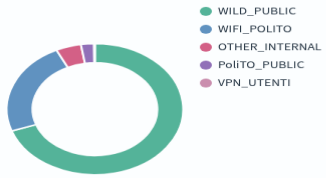
[AUTH-*] [Torta] auth_success >> auth_result



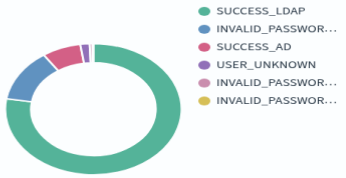
[AUTH-*] [Torta] auth_success >> sp



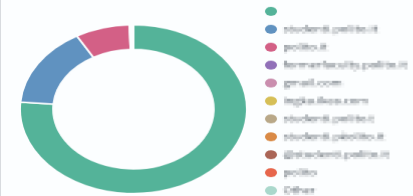
[AUTH-*] [Torta] ip_class



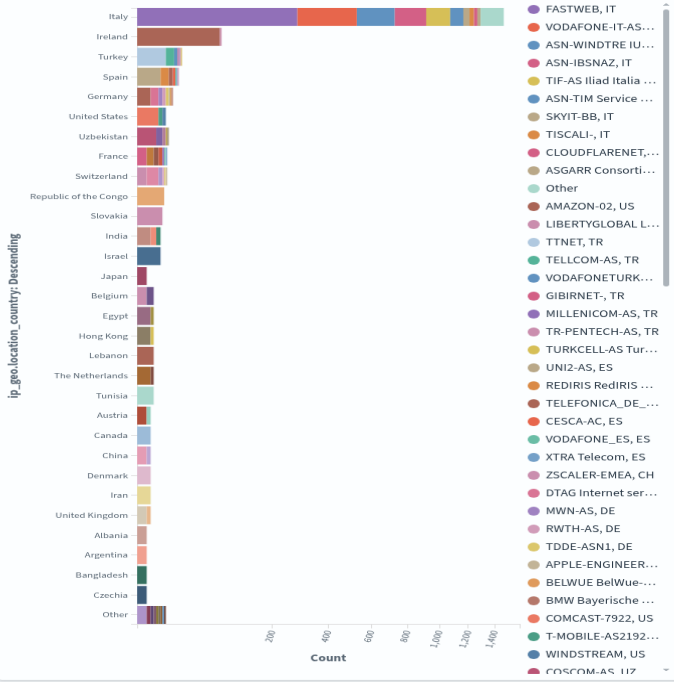
[AUTH-*] [Torta] Label



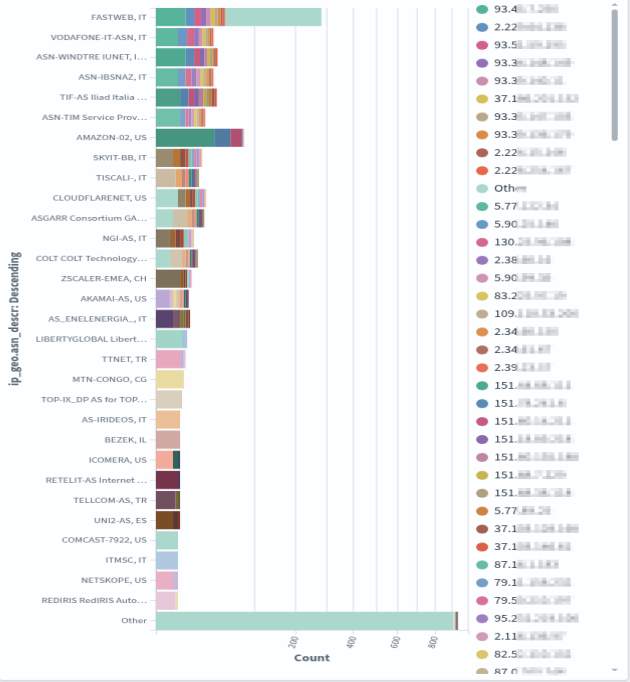
[AUTH-*] [Torta] realm



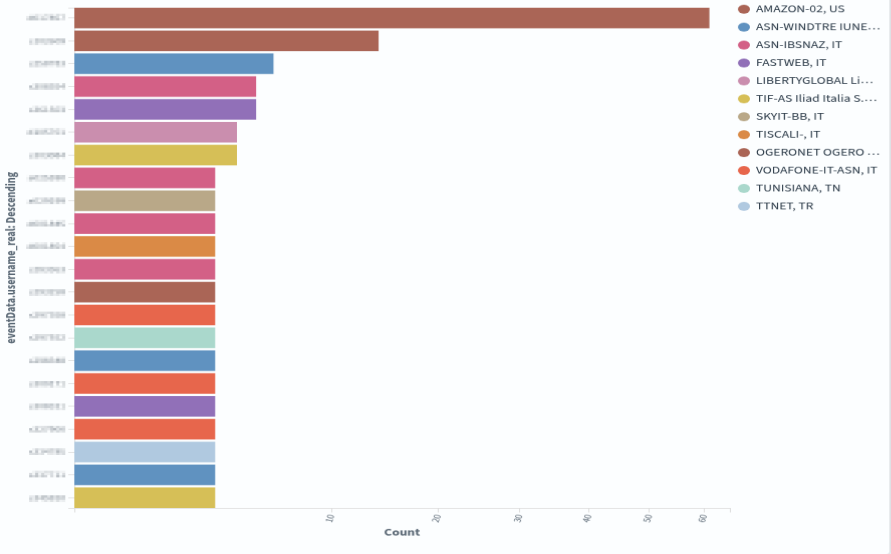
[AUTH-*][Istogramma] Country >> ASN



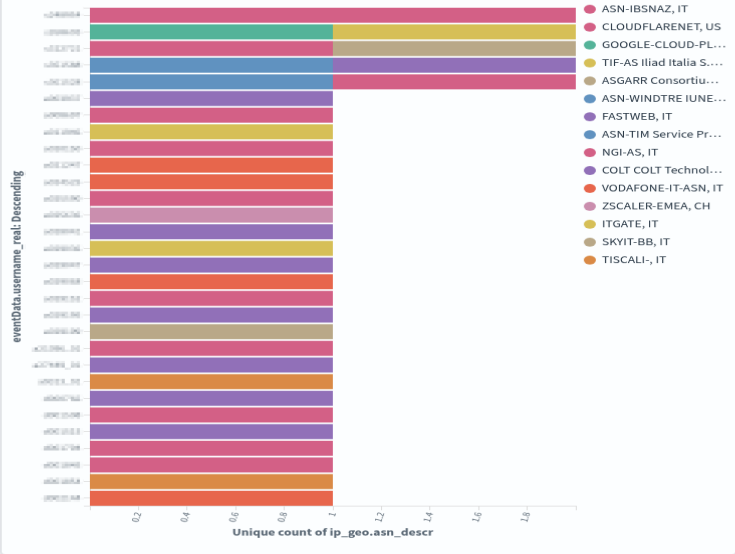
[AUTH-*][Istogramma] ASN >> IP



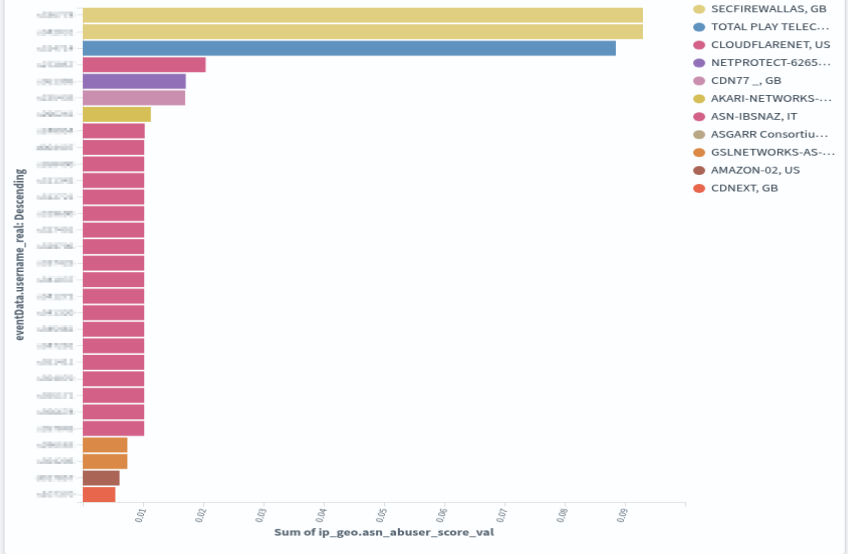
[AUTH-*][Istogramma] Utente >> num_login(asn)



[AUTH-*][Istogramma] Utente >>count(distinct(asn))



[AUTH-*][Istogramma] Utente >>sum(asn_score)



Dal prototipo all'artefatto

- Necessità di Ingestion da tante altre fonti
  - Firewall
  - VPN
  - Radius
  - Active Directory
  - DNS
  - Google Suite
  - MS365
  - Mail
  - Endpoint protection
- Correlazione degli eventi
- Scarsità di risorse umane

  Adozione di un sistema di SIEM/SOAR basato su OpenSearch analogo all'esistente

SIEM italiano
SaaS → Locale

- Selezioniamo un SIEM basato su OpenSearch, molto simile a quanto sviluppato
- Iniziamo con un contratto SaaS con dati Cloud (si paga in €/TB)
- Lavoro proattivo per arrivare ad un sistema con dati on-premises
- Mettiamo a disposizione VM sui nostri sistemi di virtualizzazione in business continuity
- Grandissima collaborazione: in pochi mesi abbiamo un sistema con log locali

  → si incominciano a caricare i log
- Soluzione scalabile: con la retention inizia subito la crescita

PT

POLITECNICO DI TORINO
Managed Security Services · Elastic / Kibana SIEM

Mappa Log Sources & Policy ILM

Giugno 2026 · Documento riservato

11 sorgenti monitorate · ~1.4 TB storage attuale · **11 policy confermate** · **ILM concordata — nessuna policy pendente**

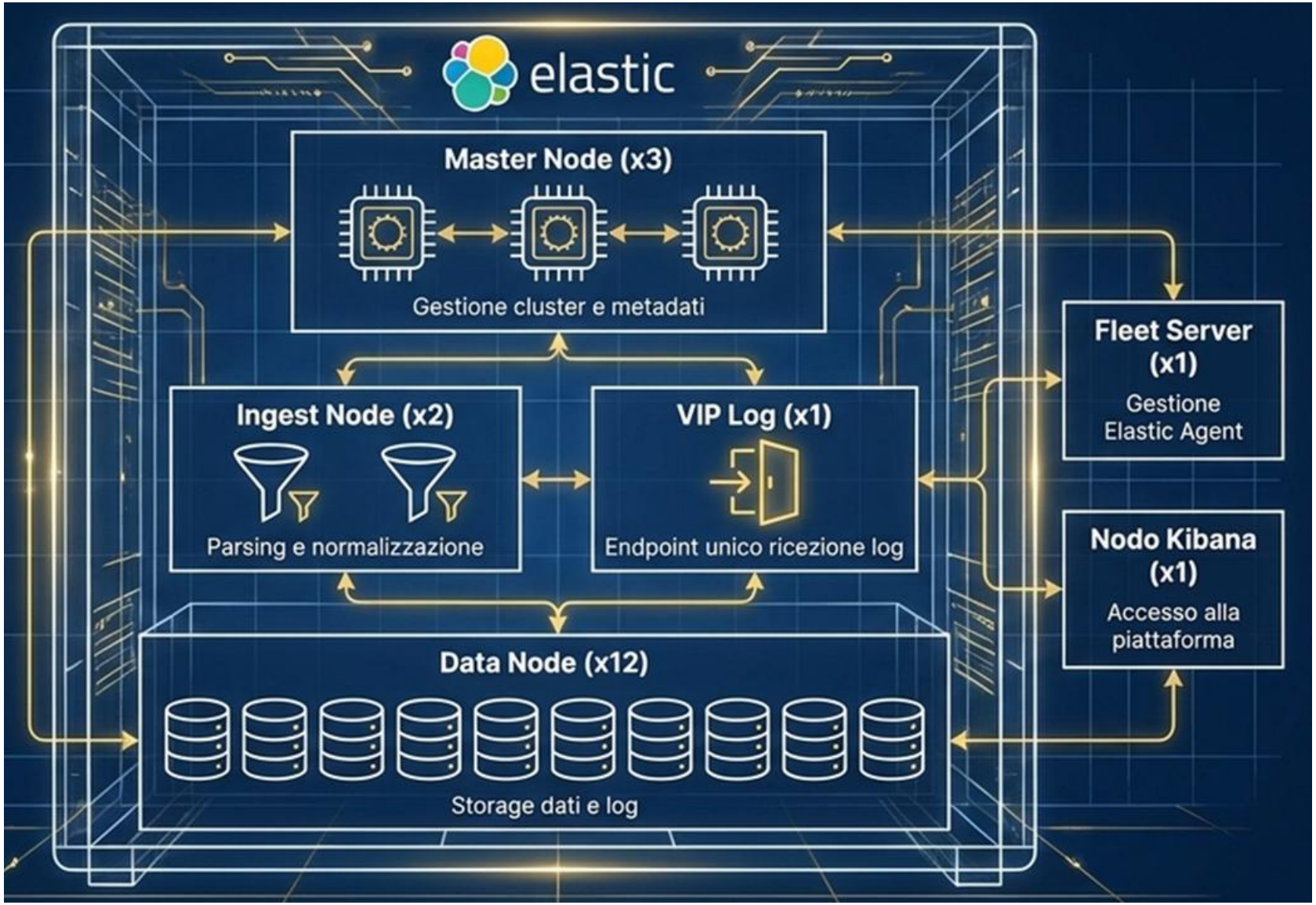
Sorgenti log e policy ILM — concordate con il cliente al 19 giugno 2026

Sorgente log	ILM attuale	ILM proposta	Storage	Stato
Checkpoint Firewall Audit log · Identity Awareness	365 giorni	365 giorni	~40 GB ~14 GB/me · n→g 202	Confermato
Checkpoint Firewall Short Log non-audit	7 giorni	7 giorni	ra rabil -day	Confermato
Shibboleth Identity & access management	365 giorni	365 gi	ese · mar→giu 20	Confermato
Microsoft 365 Audit & activity logs	365 giorni	5 giorni	~530 GB ~133 GB ese · giu 2026	Confermato
DNS Bind9 Query & resolution logs	3 me	3 mesi	GB/mese · apr→giu 2026	Confermato
Google Workspace Admin Audit & activity logs · nuovo	3 gi i	36 g	In raccolta dal 4 giu 2026	Confermato
Google Workspace Login Audit & activity logs · nuovo	365 giorni	365 giorni	In raccolta dal 4 giu 2026	Confermato
Active Directory Domain Controller — 4 sorgenti				
Application l s logs-system.a on-p	365 giorni	365 giorni	Minimo	Confermato
Sy tem logs log m.syste ito	365 giorni	365 giorni	Minimo	Confermato
Secu y logs logs-system.security-polito *Audit/compliance · l eline*	12 mesi	12 mesi	~17 GB/mese (stima) 1.8M ev/7gg · compliance only (-94%)	Confermato
Security s Sho s-s .security-polito-short *Vo levato · log critici SIEM*	1 mese	1 mese	~263 GB/mese (stima) ~27M ev/7gg · volume bulk (94%)	Confermato

+100 milioni di log/giorno
Decine di miliardi di log complessivamente gestiti

NET MAKERS

Architettura del sistema



ELASTICSEARCH MONITORING

Capacita cluster & namespace

Panoramica del cluster, dello storage disponibile e della distribuzione dei namespace.

Aggiorna dati dal cluster

SPAZIO TOTALE (CLUSTER)	SPAZIO OCCUPATO (CLUSTER)	SPAZIO DISPONIBILE (CLUSTER)
5862,94 GB	3424,37 GB · 58,41%	2438,57 GB · 41,59%

SHARDS / DATA STREAM	STATO CLUSTER
524 shards / 80 DS	GREEN · 17 nodi

Distribuzione storage per namespace

2 namespace monitorati

La home mostra la sintesi operativa. Il dettaglio resta nelle pagine dedicate dei namespace.



polito	3064,09 GB · 53 DS	99,95%
polito-short	925,74 GB · 2 DS	30,20%

Allocazione nodi

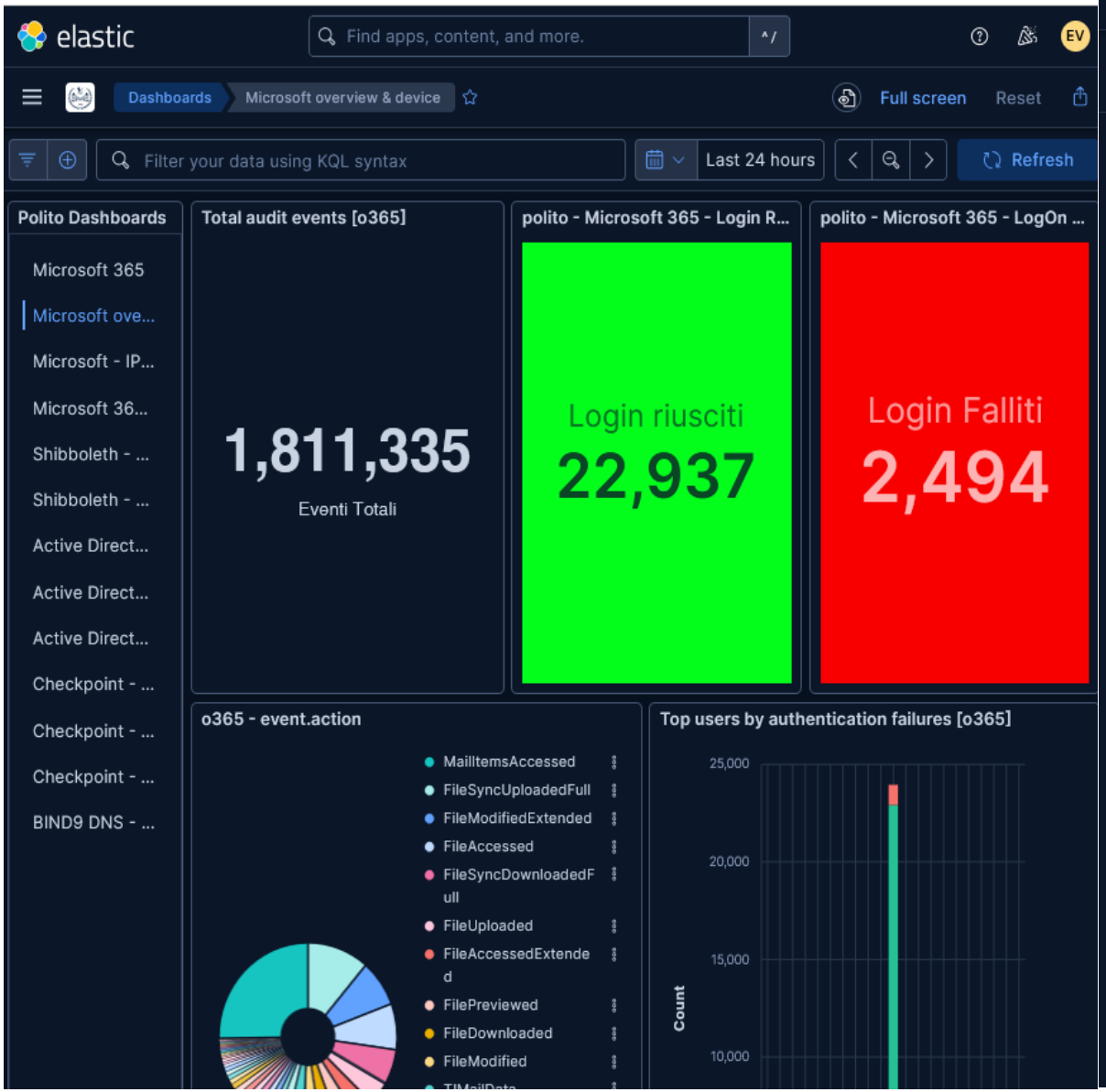
Disponibile: 2438,57 GB / Totale: 5862,94 GB / Usato: 58,41%

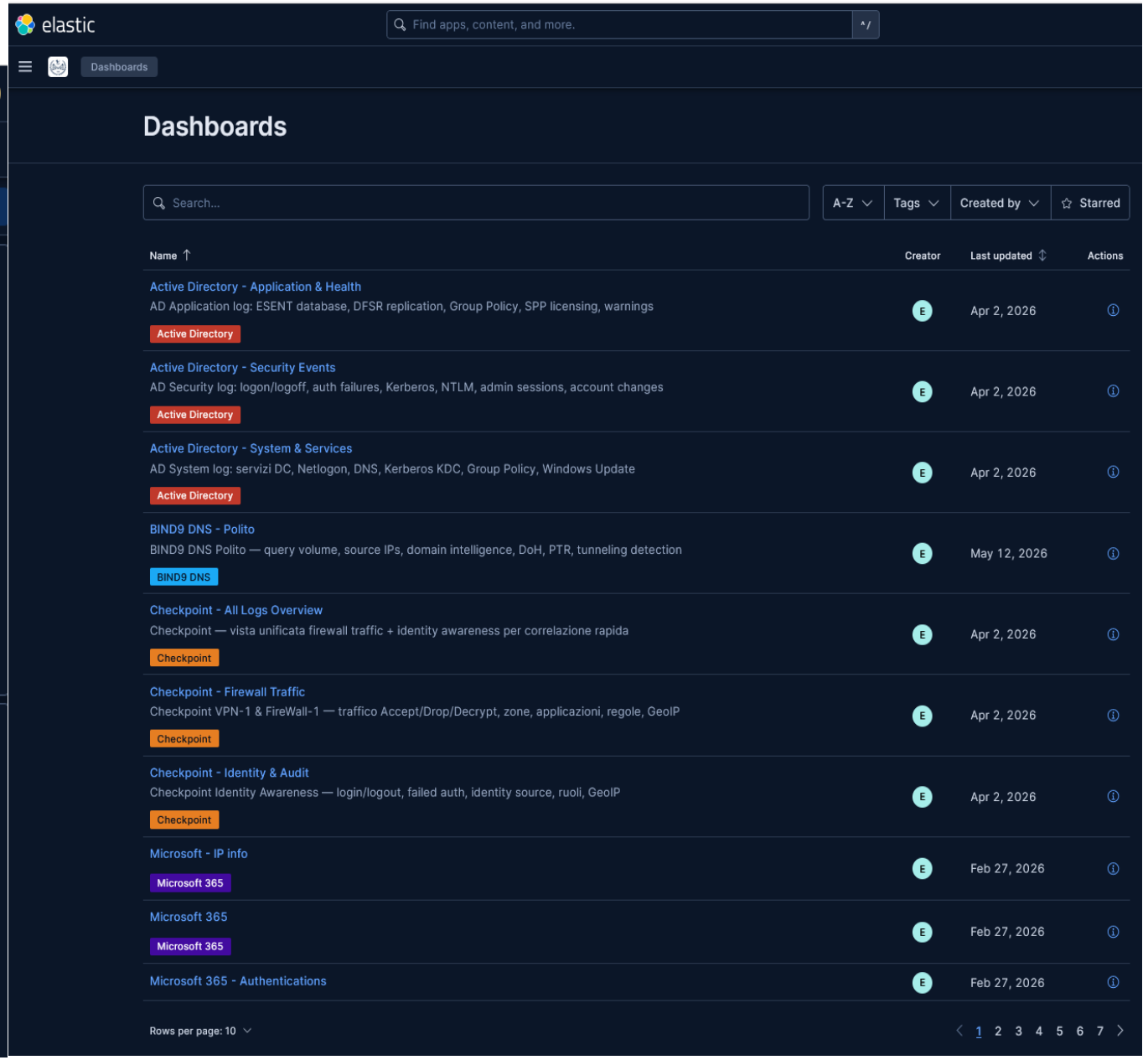
NODO	SHARDS	USATO (GB)	USATO (%)	DISPONIBILE (GB)	TOTALE (GB)
ch-data-1	39	389,98	79,00%	98,60	488,58
ch-data-4	42	341,49	69,00%	147,09	488,58
ch-data-6	41	331,50	67,00%	157,08	488,58
ch-data-2	42	318,52	65,00%	170,06	488,58
ch-data-11	45	294,81	60,00%	193,77	488,58
ch-data-8	45	272,08	55,00%	216,50	488,58
ch-data-7	45	271,69	55,00%	216,89	488,58
ch-data-3	43	266,25	54,00%	222,32	488,58
ch-data-9	42	256,68	52,00%	231,90	488,58
ch-data-5	47	237,87	48,00%	250,70	488,58
ch-data-12	46	223,42	45,00%	265,16	488,58
ch-data-10	47	220,07	45,00%	268,50	488,58

Discovery Level



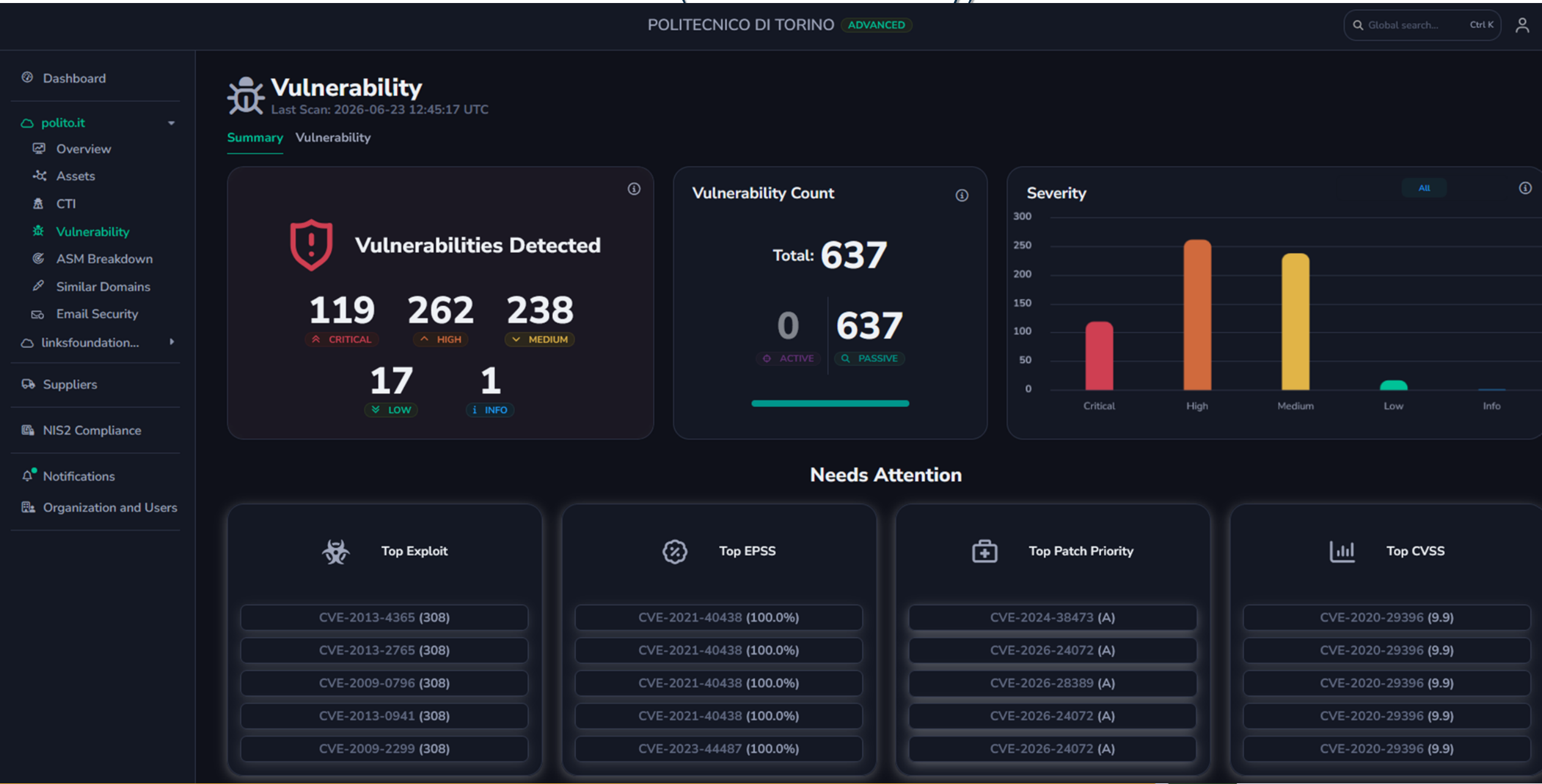
Dashboard Level





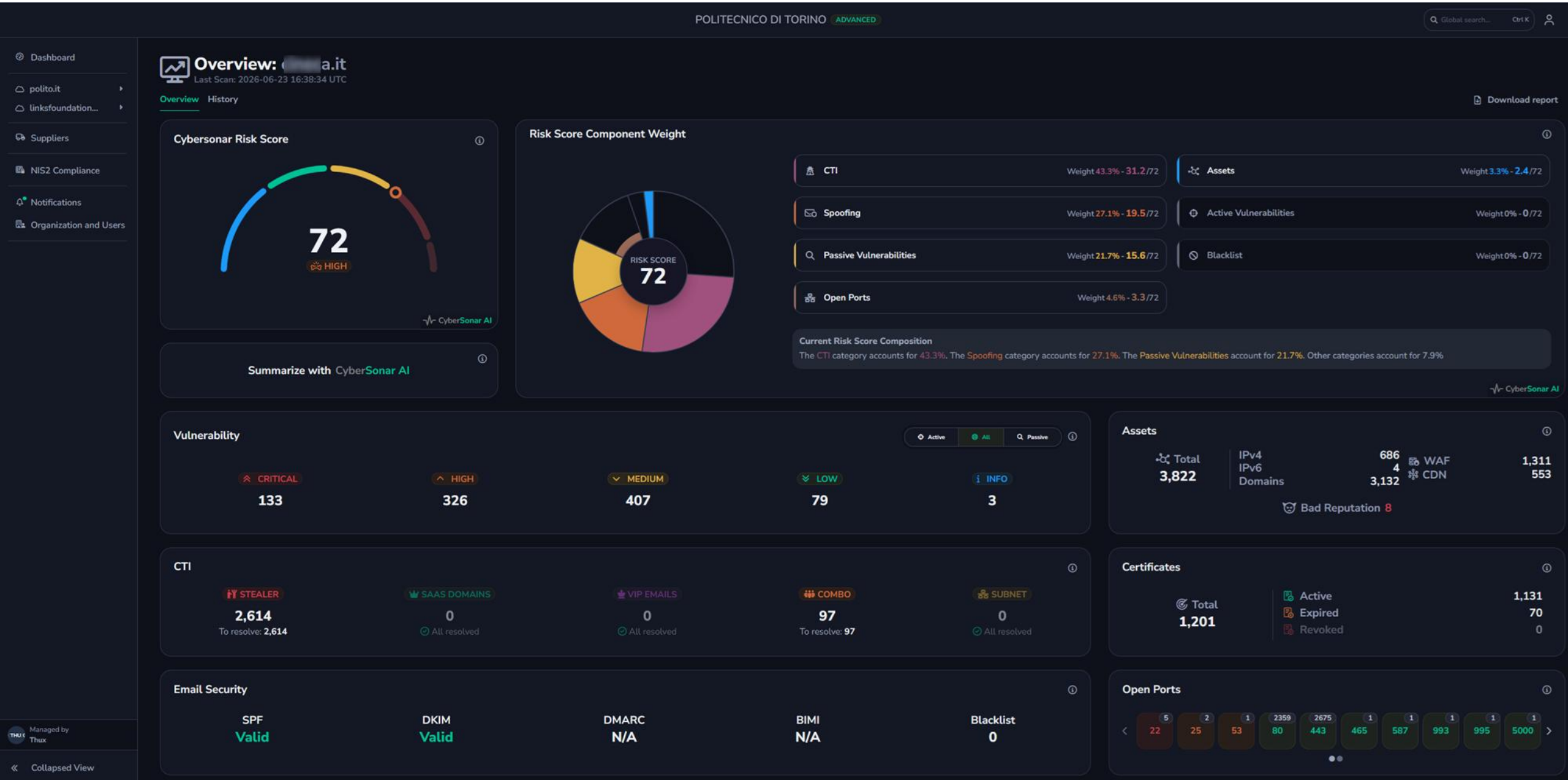
ASM Level

Affiancato da piattaforme di Automated Red Teaming (Offensive Security)

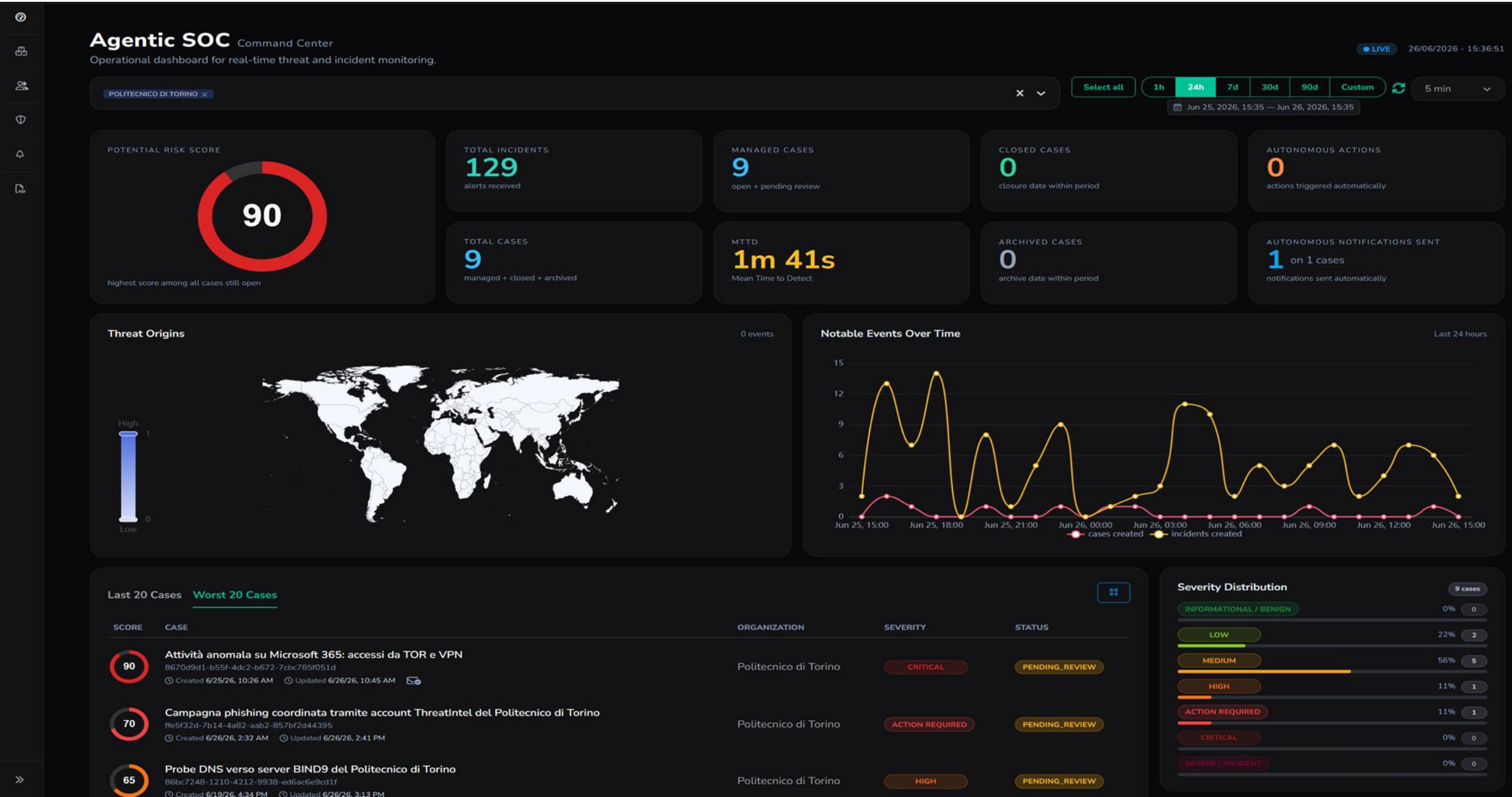


Supplier Level

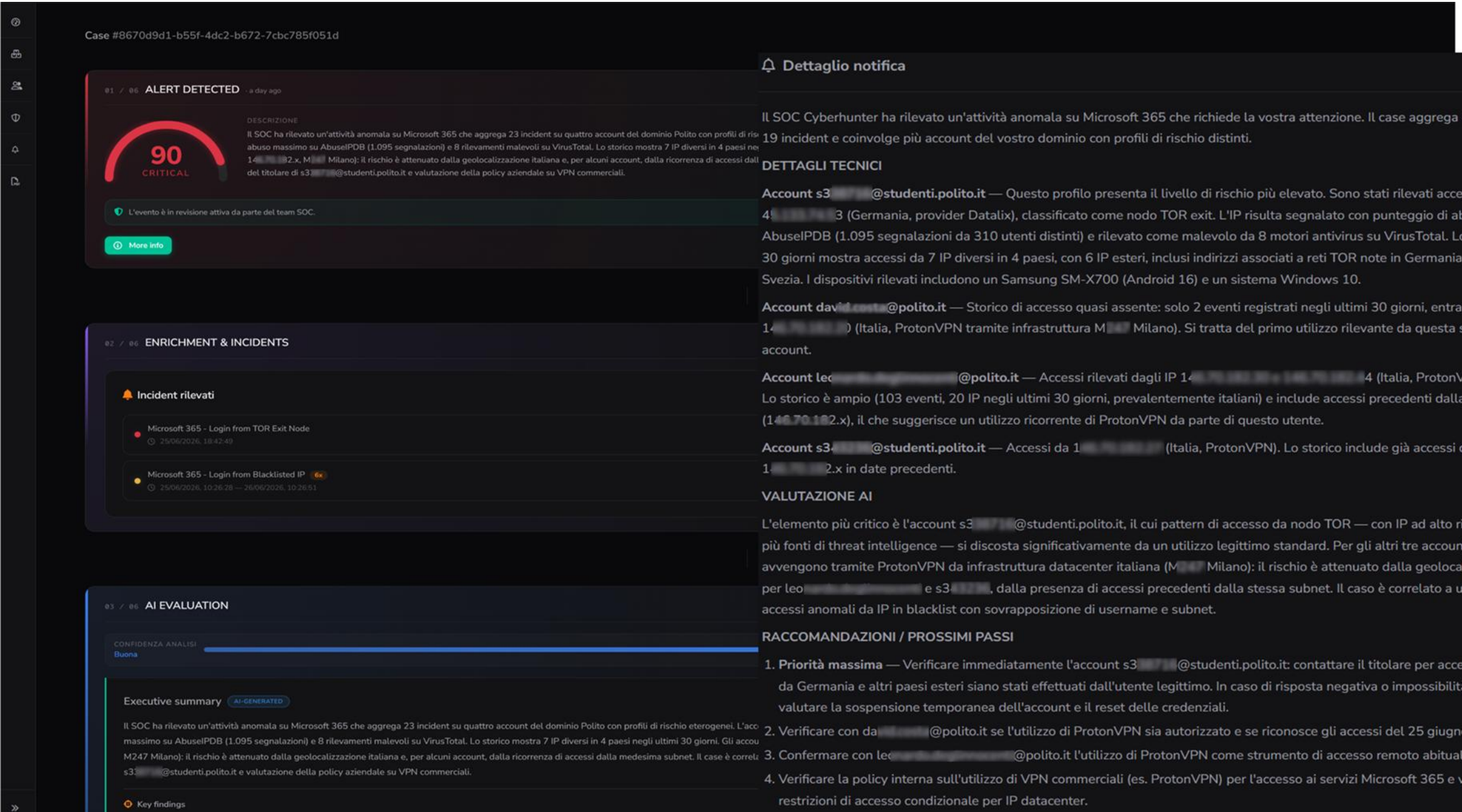
Nessuna scansione attiva



Agentic SOC Level



Agentic SOC Level



Il SOC agentico

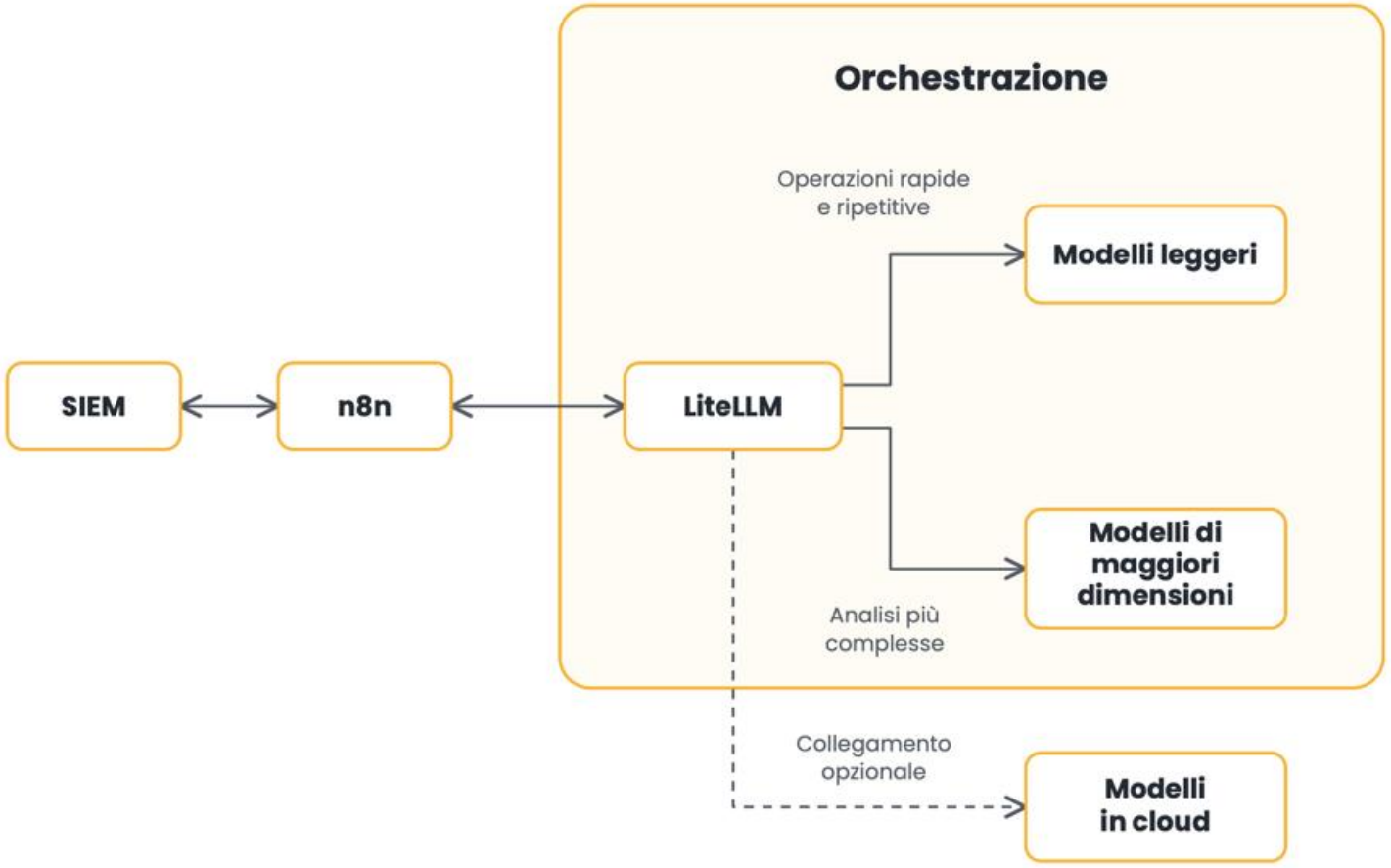
- Per fare un SOC agentico occorre l'IA
- Partiamo con le lambda function su AWS
- Correlare grandi moli di dati costa diversi token
- I dati si sono decuplicati dall'inizio del progetto
  → Si inizia a parlare di incrementi di costi
- All'inizio usare modelli di IA locale sembrava impossibile
- Insistenza, proattività, competenza, disponibilità di infrastrutture.....
  .....inizia lo studio e le prove per portare l'IA on-premises

Descrizione architettura

Deploy dei modelli LLM

Ansible prepara i server GPU e pubblica gli endpoint di inferenza usati da LiteLLM.

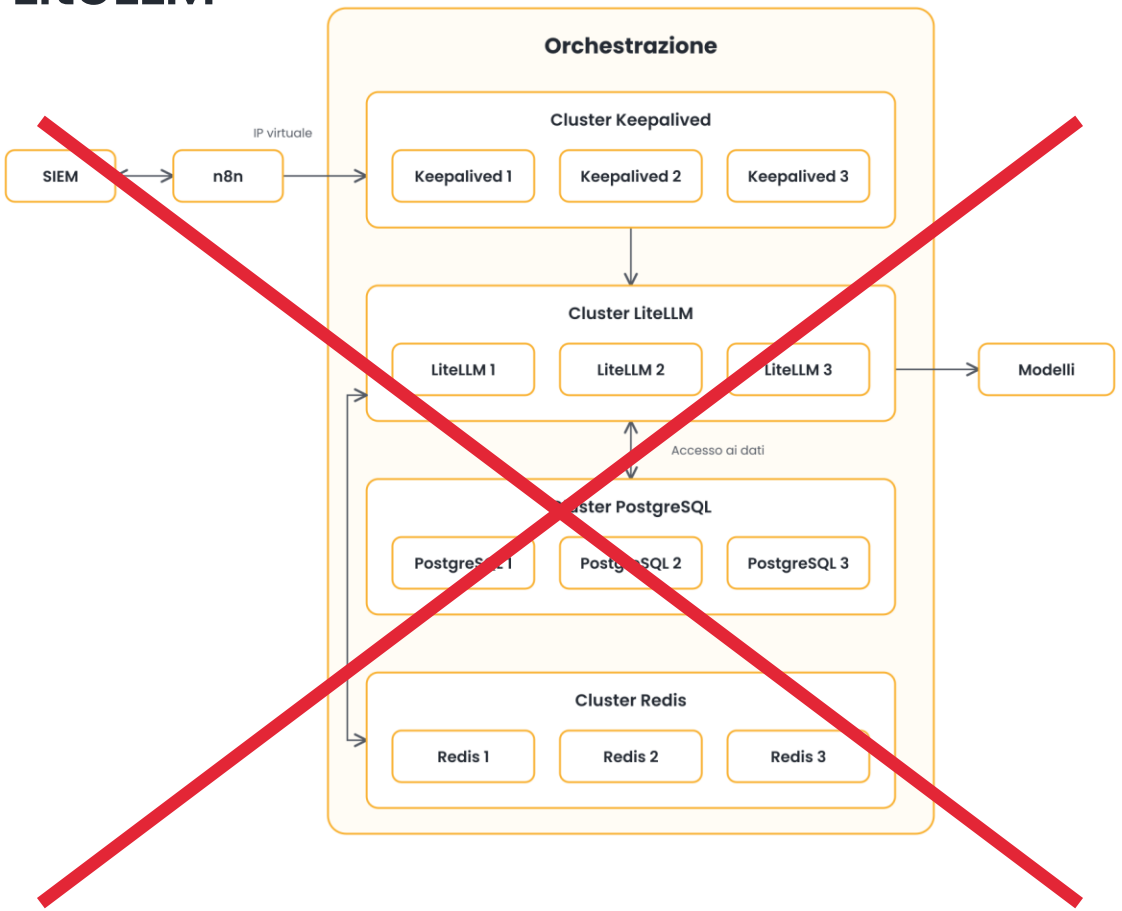
Il SIEM usa n8n per preparare il workflow di analisi tramite uno o più model. E' possibile utilizzare anche model in cloud



Orchestrazione e continuità operativa

Uno schema snello, con una sola istanza LiteLLM

- **HA vSphere** LiteLLM protetto dall'alta disponibilità del cluster.
- **Due datacenter.** Business continuity sul cluster vSphere.
- **SIEM.** Più VM nello stesso cluster.
- Se LiteLLM non è disponibile o non lo sono i model locali , l'applicativo può inoltrare il flusso a un'altra istanza LiteLLM.



Deploy con Docker

LiteLLM (e le sue dipendenze postgres e redis) vengono installati e configurati con docker

Configurazione dichiarativa

Servizio modello = catalogo + assegnazione host

File	Responsabilità
inventory.yml	Host, indirizzi e utenti di connessione
host_vars/<host>.yml	OS, GPU, runtime, engine e modelli assegnati
vars/model_catalog.yml	Sorgente, revisione e parametri del modello
vars/platform_compatibility.yml	Combinazioni ammesse e capacità della piattaforma

Un cambio di modello o parametri richiede solo il deploy dei modelli.

Configurazione dichiarativa

Un cambio di modello o parametri richiede solo l'esecuzione di un playbook

E' possibile assegnare allo stesso server piu' model (1 per GPU)

E' possibile disattivare un model , testarne un'altro e in seguito riattivarlo (i pesi rimangono sul server per evitare un nuovo download)

```
server_profile:
  os_family: Ubuntu
  os_release: "24.04"
  gpu_vendor: nvidia
  gpu_model: rtx6000_blackwell
  gpu_count: 2
  runtime: cuda
  engine: llamacpp
  nvidia:
    driver_branch: "615"
    driver_package: nvidia-driver-open
    driver_version: null
    cuda_version: "13.4"

llamacpp_version: 74bca3950054eed1795f525cb9cf06bfe796492d

model_assignments:
  - profile: qwen3_8_flash_next_q8_0_llamacpp
    port: 8080
    enabled: false
    device:
      cuda_visible_devices: "0,1"
      devices: "CUDA0,CUDA1"

  - profile: deepseek_v4_1_flash_1017_ud_iq1_s_llamacpp
    port: 8081
    enabled: true
    device:
      cuda_visible_devices: "0,1"
```

I playbook del deploy

Fase	Playbook	Risultato
Driver e runtime	prepare-platform.yml	GPU pronta
Motore LLM	install-engine.yml	Engine installato
Download e configurazione	deploy-models.yml	Modello e servizio attivi
Benchmark	benchmark-models.yml	Report JSON e Markdown
Controllo di salute	check-model-health.yml	Stato servizi e alias

Driver e runtime GPU

ansible-playbook prepare-platform.yml --limit <host>

Intel Pro Max 1100

Ubuntu 24.04
Runtime SYCL / oneAPI
Driver e strumenti Intel
Verifica con xpu-smi e sycl-ls

NVIDIA RTX 6000 Blackwell / L40S

Ubuntu 24.04 o 26.04
Driver NVIDIA e runtime CUDA
Verifica con nvidia-smi e nvcc
Controllo del numero di GPU

Vengono bloccate le combinazioni non ammesse prima delle modifiche.

Installazione del motore LLM

ansible-playbook install-engine.yml --limit <host>

llama.cpp

Build con backend SYCL o CUDA

Modelli in formato GGUF

Verifica del binario e dei backend

vLLM

Ambiente Python su NVIDIA / CUDA

Checkpoint Hugging Face compatibili

Test prima di attivare il nuovo venv

Le versioni dell'engine sono indipendenti dai driver.

Tag o commit collaudati rendono riproducibile l'installazione.

E' possibile installare branch diversi e utilizzare motori ottimizzati come Strata

Download e configurazione del modello

ansible-playbook deploy-models.yml --limit <host>

01 **Gate VRAM**

Stima di pesi, KV cache e margine engine prima del download.

02 **Artefatti**

Repo e SHA fissati. Modello, chat template e file opzionali in /opt/models.

03 **Parametri di serving**

Catalogo e override host definiscono contesto, batch, cache, MTP/Dflash e gpu utilizzate.

04 **Servizio systemd**

Launcher e unità dedicati. Avvio del modello e verifica di health.

Controllo di salute e pubblicazione

ansible-playbook check-model-health.yml --limit <host>

Verifica dell'endpoint

Servizio systemd attivo

/health risponde correttamente

/v1/models espone l'alias atteso

Ingresso in LiteLLM

Il generatore include gli endpoint con health e alias corretti.

Produce il frammento model_list o aggiorna i deployment via API.

python3 scripts/generate-litellm-config.py pubblica su litellm i model "sani"

Il test tramite chat verifica la risposta effettiva del modello.

Procedura di test

Collaudo proposto, insieme ai controlli automatici del repository

Livello	Criterio di superamento
Preflight	OS, GPU, runtime ed engine ammessi. Memoria entro il budget.
Test API	Health corretto, alias atteso e risposta valida a /v1/chat/completions.
Qualità applicativa	Prompt SIEM rappresentativi, con criteri di correttezza prestabiliti.
Stabilità	Richieste ripetute e riavvio del servizio senza crash, OOM o timeout.

Le configurazioni che superano il collaudo passano al benchmark.

Benchmark riproducibile

ansible-playbook benchmark-models.yml --limit <host>

Warm-up, poi 10 prompt sequenziali sui modelli abilitati

Misura	Interpretazione
Risposte e token	Conserva gli output e il volume effettivamente generato
Durata e token/s	Misura end-to-end, inclusi prefill e overhead HTTP
VRAM	Confronta stima, richiesta misurata e scostamento

Report JSON e Markdown in benchmark-results/<host>/

Il confronto richiede la stessa suite e gli stessi parametri di test.

Scelta della migliore configurazione

Sweep controllato: stessa base, una leva per confronto

Contesto, KV cache, parallelismo, batch e speculative decoding

01 **Filtro iniziale**
Compatibilità e fit VRAM eliminano le candidate non eseguibili.

02 **Confronto esteso**
Prime prove sulle varianti, poi ripetizioni su base e finaliste. Mediana e p95.

03 **Scelta**
Qualità e stabilità rispettano le soglie. Poi si confrontano velocità e VRAM.

04 **Promozione**
Proposta, approvazione umana, aggiornamento catalogo, deploy e nuovo health check.

Lab MTP: base senza speculative e k=1…4. Proposta solo con gain mediano ≥10%.

Modelli a catalogo e risultati dei test

11 famiglie, 30 profili attivi: piu' di 400 benchmark effettuati

#	Modello	Profili testati	
1	Qwen3.8-27B	7	Q4_K_XL+MTP
2	Qwen3.8 Flash Next	7	Q8
3	Qwen3.5-122B A10B	5	Q4_K_M
4	Flint 40B-A2.5B	4	Q8_0
5	Gemma 4 31B IT	6	UD-Q4_K_XL
6	Muse Glimmer 30B	3	Q4_K_XL+dflash
7	DeepSeek V4 Flash 0731	5	UD-IQ3_XXS
8	DeepSeek V4.1 Flash 1017	2	IQ1_M
9	Nemotron 3 Super 120B	3	IQ4-XS
10	gpt-oss-120b	3	MXFP4
11	GLM-5.3-Flash	1	IQ4_XS

Test con esito positivo

Qwen3.5-122B: viene usato comeriferimento
DeepSeek V4 Flash 0731: eccellente
Muse Glimmer: buoni risultati

Non adatti

DeepSeek V3
Command R+
Mistral Large 2/3.5
Codestral

Superati da modelli più recenti

Qwen3.6 27B e 35B: testati
Qwen3.8 27B: confronti
Flash Next: test successivi

Vivi = profili attivi nel catalogo. L'archivio riguarda i profili. Storico completo nelle note.

Ambiente di Deploy

Come è stato realizzato in breve tempo?

Soluzione iniziale all-in-one

Un server con gpu , ollama e un imbracatura agentica per avviare il primo ambiente.

Strumenti agentici testati

Codex CLI, Claude Code e Hermes.

Configurazione realizzata

LiteLLM, Redis e PostgreSQL in docker
Configurati subagenti (planning,coding,doc..) per usare il model giusto per ogni operazione
Utilizzo di Git per isolare l'ambiente di deploy dalla produzione

Considerazioni su prestazioni e memoria

Configurazione custom e Ollama

Fino al +40%

di token/s sulle stesse GPU

Il deploy con configurazioni custom può superare Ollama anche con finestre di contesto più ampie.

Quantizzazione e offload

Quantizzazioni diverse e offload degli esperti in RAM o su NVMe consentono di utilizzare modelli inizialmente fuori portata.

Esempi citati

DeepSeek V4.1 Flash 1017
GLM-5.3-Flash

Conclusioni

- Privacy & Sovranità sui dati
- Economicità e sostenibilità
- Scalabilità → possibilità ampliare le fonti
  - es. log https
    in alternativa o complementazione di firewall https inspection/IDS
- Accountability → occorrono benchmark x valutare affidabilità
  - Si provano più modelli locali e si valutano e raffrontano i risultati
  - Si raffrontano risultati ottenuti da modelli locali rispetto a modelli cloud

Ci potrebbero essere tutte le pre-condizioni per un SOC H24 a sola reperibilità

Grazie

enrico.venuto@polito.it

davide.stevano@polito.it

WORKSHOP GARR 2026

NET MAKERS