

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

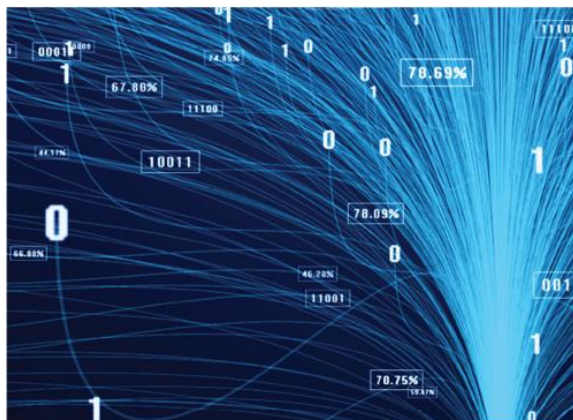
**DIPARTIMENTO REGIONALE DELL'AUTORITA' DI BACINO
DEL DISTRETTO IDROGRAFICO DELLA SICILIA
SERVIZIO 5 - ATTUAZIONE INTERVENTI A TITOLARITA' O A REGIA**

Programma Operativo Ambiente FSC 2014-2020 – Asse 2 – Linea di Azione 2.3.1 Progetto: “Autorità di Bacino del Distretto Idrografico della Sicilia – Interventi per il miglioramento della qualità dei corpi idrici”- Linea d'intervento L3- Aggiornamento quadro conoscitivo in materia di derivazioni - Attività “Servizio informatico per la creazione banca dati documentale dei titoli di derivazione” e “Servizio tecnico di reperimento dati presso uffici competenti al rilascio di titoli concessori e autorizzatori (Genio Civile e Dipartimento dell'acqua e dei rifiuti) verifica e caricamento in banca dati” - CUP: F62G16000000001	Accordo Quadro Consip denominato “Applicativi di Data management e servizi di PMO - Ordine Diretto - Lotto 4 - Servizi Applicativi di Data Management per le PAL Centro/Sud” CIG A.Q. 8184365FA4 CIG DERIVATO 9264301708
--	---

Piattaforma IDROCAP

Servizio LA.DW.1

- Manuale di gestione dell'applicazione -



Autore del documento: **RTI**

 Fondo per lo Sviluppo e la Coesione	 Piano Operativo Ambiente	 MINISTERO DELLA TRANSIZIONE ECOLOGICA
Programmazione FSC 2014-2020 Piano Operativo Ambiente		

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

1	SEZIONE DI CONTROLLO DEL DOCUMENTO	4
1.1	LISTA DEI CAMBIAMENTI	4
1.2	LISTA DISTRIBUZIONE DEL DOCUMENTO	4
2	DOCUMENTI DI RIFERIMENTO	5
3	ACRONIMI E DEFINIZIONI	5
4	INTRODUZIONE.....	7
5	GESTIONE E MANUTENZIONE DELLE WEB APPLICATION	8
6	GESTIONE E AMMINISTRAZIONE DELLO STORAGE DISTRIBUITO	9
6.1	FALLIMENTO DI PIÙ DI DUE NODI	11
7	GESTIONE E MANUTENZIONE DELLE BASI DI DATI	12
7.1	DATABASE PRINCIPALE	12
7.1.1	<i>Configurazione del DB</i>	<i>12</i>
7.1.2	<i>Manutenzione preventiva del DB</i>	<i>13</i>
7.1.3	<i>Monitoraggio delle performance del DB</i>	<i>13</i>
7.1.4	<i>Gestione utenti, privilegi e sicurezza</i>	<i>14</i>
7.1.5	<i>Ripristino del DB</i>	<i>15</i>
7.1.6	<i>Migrazione del disco con consistenza dei dati e riduzione del downtime</i>	<i>15</i>
8	GESTIONE E MANUTENZIONE DELLE NOTIFICHE	16
8.1	PANORAMICA DEI SERVIZI	16
8.2	CONFIGURAZIONE DEI SERVIZI DI MESSAGGISTICA/NOTIFICA	17
8.3	GESTIONE E MANUTENZIONE	17
8.3.1	<i>Monitoraggio del sistema</i>	<i>18</i>
8.3.2	<i>Aggiornamento delle immagini</i>	<i>18</i>
8.3.3	<i>Gestione delle configurazioni SMTP</i>	<i>18</i>
8.3.4	<i>Backup e ripristino</i>	<i>18</i>
9	GESTIONE E MONITORAGGIO DELL'INFRASTRUTTURA	18
9.1	PROMETHEUS	19

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

9.2	ORCHESTRATOR CON DOCKER SWARM	21
9.3	ANSIBLE	21
9.4	DOCKER REGISTRY & LABORATORY	22
9.5	RIFERIMENTI UTILI	25

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

1 SEZIONE DI CONTROLLO DEL DOCUMENTO

1.1 LISTA DEI CAMBIAMENTI

Riepilogo delle versioni			
Versione	Data	Nome	Motivo aggiornamento
01.00	31/03/2023	RTI	Prima emissione (Lotto 1)
02.00	26/06/2023	RTI	Estensione funzionalità (Lotto 2)
03.00	26/09/2023	RTI	Estensione funzionalità (Lotto 3)
04.00	14/06/2024	RTI	Estensione funzionalità (Lotto 4)

1.2 LISTA DISTRIBUZIONE DEL DOCUMENTO

Lista di Distribuzione

Società	Nome	Cognome	Ruolo
Regione Siciliana - Dipartimento Regionale dell'Acqua e dei Rifiuti	Vito	Cutrone	RUP
	Raffaele	Di Salvo	DEC
Regione Siciliana – Dipartimento Regionale dell'Autorità di bacino	Giuseppe	Sciacca	Punto Ordinante
	Enzo Marco	Castrogiovanni	Collaboratore RUP/DEC
LEONARDO S.p.A.	Antonio	Rumiz	RUAC
	Ennio	Girimonte	Referente Tecnico
	Pasquale	Fierro	PM

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

2 DOCUMENTI DI RIFERIMENTO

Rif.	Nome Documento	Titolo
DOC1	AQDML3IRCP01SR01 Rev 04.00 Autorità di Bacino del Distretto Idrografico della Sicilia-SdR	LA.DW.1 - Specifica dei Requisiti
DOC2	AQDML3IRCP01SDD01 Rev.04.00 - IDROCAP - Specifiche di progettazione	Piattaforma IDROCAP Servizio LA.DW.1 - Specifiche di progettazione -
DOC3	AQDML3IRCP01SR02 Rev.04.00 - IDROCAP - Specifiche Funzionali	Piattaforma IDROCAP Servizio LA.DW.1 - Specifiche funzionali -

3 ACRONIMI E DEFINIZIONI

Acronimi/Definizione	Descrizione
DRAR	Dipartimento regionale dell'Acqua e dei Rifiuti
GC	Genio Civile
AdB	Autorità di Bacino
IDROCAP	Sistema di gestione delle captazioni idrauliche
N/A	Non applicabile
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
LTS	Transport Layer Security
IP	Internet Protocol
CLI	Command Line Interface
API	Application Programming Interface
HTTPS	HyperText Transfer Protocol Secure
SDK	Software Development Kit
CRUD	Create, Read, Update, Delete
PHP	Hypertext Preprocessor
AWS	Amazon Web Services
RAM	Random Access Memory

Programmazione FSC 2014-2020 Piano Operativo Ambiente	  
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione	Pag. 5 di 27 Stato: Rev.04.00

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

Acronimi/Definizione	Descrizione
vCPU	Virtual Central Processing Unit
GB	GigaByte
DB	DataBase
SQL	Structured Query Language
DKIM	DomainKeys Identified Mail
SPF	Sender Policy Framework
DMARC	Domain-based Message Authentication, Reporting & Conformance
DNS	Domain Name System
YAML	Ain't Markup Language
VM	Virtual Machine
SSH	Secure SHell

 	Titolo	
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS

4 INTRODUZIONE

Questo è il manuale dedicato alla gestione dell'applicazione IDROCAP relativa all'infrastruttura e ai microservizi che la compongono. Questo manuale è stato creato con l'obiettivo di fornire agli amministratori di sistema e ai responsabili della gestione delle applicazioni, una guida pratica per la gestione e la manutenzione di un'infrastruttura basata su microservizi.

Il manuale copre una vasta gamma di argomenti, dalla gestione e manutenzione delle Web Application allo storage distribuito, dalle basi di dati alle notifiche, fino alla gestione e monitoraggio dell'infrastruttura. Ogni sezione fornisce una panoramica dettagliata delle operazioni che devono essere eseguite per mantenere l'infrastruttura funzionante in modo efficiente ed affidabile.

La sezione 2 fornisce informazioni sulla gestione e la manutenzione delle Web Application, mentre la sezione 3 si concentra sulla gestione e l'amministrazione dello storage distribuito per mezzo di MinIO. La sezione 4 tratta in modo specifico la gestione e la manutenzione delle basi di dati, con una sezione dedicata alla configurazione del DB, alla manutenzione preventiva, al monitoraggio delle performance, al ripristino e alla migrazione del disco.

La sezione 5 fornisce informazioni sulla gestione e la manutenzione delle notifiche, con una panoramica dei servizi di messaggistica e notifica, la configurazione dei servizi e la gestione delle configurazioni SMTP. La sezione 6 è dedicata alla gestione e al monitoraggio dell'infrastruttura, con informazioni su Prometheus, Orchestrator con Docker Swarm, Ansible, e Docker registry & laboratory.

In sintesi, questo manuale offre una guida dettagliata alla gestione e alla manutenzione di un'infrastruttura a microservizi. Questo manuale costituisce uno strumento utile alla gestione della infrastruttura e al mantenimento dell'applicazione, supportandone il corretto funzionamento e garantendone efficienza ed affidabilità.

Disclaimer per ciclo di sviluppo completo a lotti

L'approccio è caratterizzato da fasi di realizzazione e prevede l'utilizzo di cicli di: pianificazione e raccolta dei requisiti, progettazione, sviluppo, testing e controllo qualità, rilascio a lotti, monitoraggio e feedback, rilascio completo, sviluppi iterativi e incrementali, in cui il team di sviluppo si concentra sulla realizzazione di un insieme di funzionalità specifiche. Ogni rilascio incrementale contiene un insieme di funzionalità autoconsistenti da sottoporre a validazione/collauda; il team lavora insieme all'utente per definire, progettare, sviluppare, testare e consegnare una porzione del prodotto finale. In questo modo, il prodotto nel suo complesso viene sviluppato in modo graduale ed incrementale; il team ha la possibilità di ricevere feedback costante dal cliente e di adattarsi rapidamente alle esigenze in evoluzione.

Per ogni lotto, il team di sviluppo si concentra sulla identificazione dei requisiti e delle specifiche di progettazione che sono stati definiti per quel particolare lotto. Se necessario, il team può apportare modifiche ad elementi dei lotti precedenti per riflettere le nuove esigenze raccolte dal cliente od integrare nuove informazioni acquisite durante la fase di monitoraggio e feedback. Tuttavia, queste modifiche vengono

Programmazione FSC 2014-2020 Piano Operativo Ambiente		  
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione		Pag. 7 di 27 Stato: Rev.04.00

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

gestite in modo strutturato e controllato, in modo da minimizzare l'impatto sullo sviluppo complessivo del prodotto.

In sintesi, dopo aver eseguito diversi cicli di rilascio a lotti ed aver affrontato eventuali problemi significativi, l'applicazione sarà pronta per un rilascio completo a tutti gli utenti, ciò consente di sviluppare il prodotto in modo graduale, adattarsi ai feedback ricevuti dal cliente e gestire le modifiche alle specifiche di progettazione in modo efficiente.

Per ulteriori informazioni sulla metodologia di progettazione fare riferimento al documento “Appendice 2 al Capitolato tecnico speciale Lotti 1-2-3 - Cicli e Prodotti”.

5 GESTIONE E MANUTENZIONE DELLE WEB APPLICATION

Le web application sono progettate e realizzate in modo distribuito per affrontare diversi tipi di carico e garantire elevata affidabilità. Esse si basano su immagini Docker personalizzate, che rispondono alle esigenze specifiche dell'applicazione. Il sistema è progettato per essere flessibile e adattabile a diverse infrastrutture, sia on-premise che in cloud.

Il traffico verso le web application viene gestito tramite una logica di load balancing e floating IP, implementata attraverso l'utilizzo di load balancer dedicati. Questi possono essere installati su macchine virtuali on-premise che fungono da bastion host, oppure essere forniti direttamente dall'infrastruttura cloud di riferimento. La scelta tra le due opzioni dipenderà dalle risorse messe a disposizione dal cliente, poiché entrambe le soluzioni sono applicabili.

L'obiettivo principale è quello di garantire un tempo di inattività nullo durante le operazioni di manutenzione o gestione delle web application. Per raggiungere tale obiettivo, è fondamentale disporre di diversi servizi che espongono la stessa web application, in modo da mantenere i servizi sempre attivi e funzionanti, senza compromettere le funzionalità del progetto.

Le principali attività di gestione e manutenzione delle web application sono state identificate come segue:

- Aggiornamento dei certificati SSL/TLS per garantire connessioni sicure.
- Aggiornamento dell'applicativo per risolvere bug e malfunzionamenti vari.
- Aggiornamento dell'applicativo per implementare nuove funzionalità (importante in un contesto Agile che coinvolge l'intero progetto).
- Aggiornamento del file di configurazione dell'applicativo (custom.php).
- Ispezione dei log per monitorare il funzionamento delle web application.
- Lock down del codice sorgente per proteggersi contro potenziali attacchi informatici.

 	Titolo	
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS

La maggior parte di queste operazioni viene effettuata utilizzando script Ansible, ad eccezione dell'analisi dei log. Per esaminare i log, è necessario accedere al percorso /var/www/jixel/logs/. È importante tenere presente che, a causa della logica di alta disponibilità implementata nelle web application, potrebbe essere necessario ispezionare i log di ogni singola macchina per identificare e risolvere eventuali problemi.

In conclusione, l'architettura distribuita delle web application consente di garantire elevate prestazioni, scalabilità e affidabilità. Grazie all'utilizzo di tecnologie come Docker, load balancing e floating IP, è possibile adattare il sistema alle diverse esigenze infrastrutturali dei clienti e mantenere i servizi sempre attivi e funzionanti, anche durante le attività di manutenzione e gestione.

6 GESTIONE E AMMINISTRAZIONE DELLO STORAGE DISTRIBUITO

La gestione dello storage distribuito è un'attività essenziale per l'infrastruttura di IDROCAP in quanto è strutturata per la gestione di grandi quantità di dati, in particolare per la gestione degli allegati (principalmente la storicizzazione delle pratiche di concessione). Minio è una delle soluzioni più popolari per lo storage distribuito e rappresenta una tecnologia leggera, semplice da gestire e altamente scalabile.

La corretta gestione di un cluster Minio richiede una serie di attività, la configurazione dei nodi, la definizione dei bucket per l'organizzazione dei dati, la gestione dei permessi di accesso, la ridondanza dei dati e la sicurezza dei dati.

La guida che segue fornisce una panoramica completa sulla gestione di un cluster Minio, con dettagli su ciascuna di queste attività.

Una corretta gestione dello storage distribuito consente di garantire la disponibilità, la ridondanza e la sicurezza dei dati, fornendo una base solida per l'elaborazione distribuita e l'analisi dei dati. Questa guida può essere utile per una migliore comprensione su come gestire un cluster Minio e sfruttare al meglio questa tecnologia per le esigenze di storage distribuito.

Concetti principali

- Server MinIO:** I server MinIO possono essere configurati come singoli nodi o come cluster distribuito per fornire storage scalabile e ad alta disponibilità. Per garantire un'alta ridondanza e disponibilità in caso di failover in un cluster MinIO, è consigliabile utilizzare almeno 4 nodi. Questa configurazione permette di implementare l'Erasure Coding, una tecnica di protezione dei dati che suddivide e distribuisce i dati e le informazioni di parità su diversi dispositivi di archiviazione. L'Erasure Coding di MinIO consente una tolleranza ai guasti configurabile, che può essere definita in base al rapporto dati/parità. Per tanto su IDROCAP sono previsti inizialmente 4 nodi. In un cluster di 4 nodi, è possibile utilizzare un rapporto di 2 dati e 2 parità (2+2), il che significa che il sistema può tollerare la perdita di 2 nodi contemporaneamente senza compromettere l'integrità e la disponibilità dei dati.

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

- **Client MinIO (mc):** Il client MinIO, chiamato mc, è uno strumento CLI che fornisce un'interfaccia utente per interagire con il server MinIO e gestire i dati.
- **Console MinIO:** La console MinIO è un'interfaccia web che permette la gestione e il monitoraggio dell'infrastruttura MinIO. Su IDROCAP è esposta alla porta 9000, ma accessibile solo per mezzo di autenticazione/autorizzazione e confinata con regole di firewall ad un ristretto numero di IP sorgente.
- **Bucket:** I dati in MinIO sono organizzati in bucket. Un bucket è un contenitore per gli oggetti e può contenere un numero illimitato di oggetti.
- **Oggetti:** Gli oggetti sono i file effettivi memorizzati in un bucket. Ogni oggetto ha un nome univoco all'interno del bucket.

Gestione del cluster MinIO

Essendo che l'ecosistema è realizzato per mezzo di containers Docker, il nodo/server di MinIO risiederà su uno dei nodi gestiti dall'orchestratore/balancer dei servizi. Pertanto la maggior parte delle operazioni qui elencate potrà essere effettuata direttamente accedendo all'interno del container con i comandi ufficiali. Inoltre è stata esposta la porta della dashboard web presso indirizzi federati, così da poter gestire alcune semplici operazioni di monitoring/gestione da interfaccia utente grafica per mezzo di autenticazione.

- **Avvio/Riavvio/Spegnimento:** Il cluster MinIO può essere avviato, riavviato e spento utilizzando i comandi appropriati forniti dalla documentazione ufficiale di MinIO.
- **Backup:** Per eseguire il backup dei dati in MinIO, è possibile utilizzare il comando mc mirror per copiare gli oggetti tra bucket o tra diversi server MinIO. Il backup viene effettuato regolarmente secondo quanto stabilito dal documento **“Documentazione delle procedure batch/DTS”**.
- **Monitoraggio:** MinIO supporta Prometheus per il monitoraggio delle metriche del cluster. Inoltre, MinIO fornisce la propria console di amministrazione per monitorare lo stato del cluster e gestire i bucket e gli oggetti. Tenere presente che il monitoraggio è sempre a disposizione per ciascuno nodo all'interno del servizio di Prometheus che risiede su uno dei container dell'infrastruttura.
- **Gestione degli accessi:** MinIO supporta la gestione degli accessi basata su ruoli (RBAC) e la gestione delle policy per controllare l'accesso ai bucket e agli oggetti. Le credenziali di accesso e le chiavi segrete possono essere gestite tramite il client MinIO o la console di amministrazione. Ciascun servizio che consuma le API di MinIO ha la propria coppia di credenziali (secret).

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

- **Sicurezza:** MinIO supporta la crittografia server-side e client-side per proteggere i dati memorizzati. Inoltre, MinIO può essere configurato per utilizzare HTTPS per garantire la sicurezza delle comunicazioni tra client e server.
- **Aggiunta/rimozione di nodi:** Nel caso di un cluster distribuito, è possibile aggiungere o rimuovere nodi dall'infrastruttura MinIO. È importante notare che l'aggiunta o la rimozione di nodi può richiedere una modifica della configurazione del cluster e potrebbe influire sulla ridondanza dei dati. Effettuare questa operazione solo se si evidenziano gravi criticità nell'ordinario funzionamento del nodo. Tenere in mente che lo storage di IDROCAP è progettato per supportare la perdita di due nodi contemporaneamente, quindi 2 failover consecutivi.
- **Ridondanza:** MinIO supporta la ridondanza tramite il meccanismo di Erasure Coding. Ciò garantisce la disponibilità e la durabilità dei dati, anche in caso di guasti hardware o perdita di nodi.
- **Interazione con applicazioni:** Le applicazioni possono interagire con l'infrastruttura MinIO utilizzando l'SDK MinIO fornito, disponibile in diversi linguaggi di programmazione, come Python, Java, JavaScript, .NET, e altri. L'SDK permette alle applicazioni di eseguire operazioni CRUD (Create, Read, Update, Delete) sugli oggetti e di gestire i bucket e le policy di accesso. L'interazione con le Web Application di IDROCAP sono effettuate per mezzo della libreria PHP AWS SDK for PHP – Version 3.
- **Scalabilità:** MinIO è progettato per essere altamente scalabile, sia in termini di capacità di storage che di prestazioni. Il cluster distribuito di MinIO può essere facilmente espanso aggiungendo nodi aggiuntivi o aumentando la capacità di storage dei nodi esistenti. Inoltre, MinIO è ottimizzato per garantire elevate prestazioni in lettura e scrittura, adattandosi alle esigenze delle applicazioni moderne. Questo si adatta bene alle esigenze di progetto in quanto si presta in modo assoluto alle variazioni in termini di storage necessario, che può essere crescente o decrescente. Tuttavia si tiene bene presente che il minimo numero di nodi che deve essere sempre disponibile è di 4, per i motivi descritti sopra.
- **Migrazione dei dati:** MinIO offre strumenti e funzionalità per facilitare la migrazione dei dati da e verso altre piattaforme di storage, come Amazon S3, Google Cloud Storage, Azure Blob Storage e altri sistemi compatibili con S3. Il comando `mc cp` e il comando `mc mirror` possono essere utilizzati per copiare e sincronizzare i dati tra diverse piattaforme di storage.

6.1 FALLIMENTO DI PIÙ DI DUE NODI

Questa procedura permette di garantire la consistenza dei dati e il corretto funzionamento del file system distribuito MinIO anche in caso di guasti che coinvolgono più nodi. Il backup infrastrutturale tramite snapshot dei volumi delle macchine virtuali rappresenta, quindi, un'importante misura di protezione per preservare le

Programmazione FSC 2014-2020 Piano Operativo Ambiente	 	MINISTERO DELLA TRANSIZIONE ECOLOGICA
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione		Pag. 11 di 27 Stato: Rev.04.00

 	Titolo	
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS

pratiche digitalizzate e assicurare la disponibilità e l'affidabilità del sistema. Nel caso accadesse seguire attentamente la seguente procedura:

- Identificazione degli snapshot corrispondenti ai volumi dei nodi interessati, basandosi sulla data e sull'orario di creazione.
- Creazione di nuove macchine virtuali o sostituzione dei volumi danneggiati con quelli recuperati dagli snapshot, assicurando che le configurazioni e le impostazioni siano coerenti con la struttura originale del cluster MinIO.
- Verifica dell'integrità dei dati ripristinati e della corretta funzionalità del cluster MinIO, effettuando test e confrontando i risultati con quelli ottenuti prima del fallimento.
- Monitoraggio delle prestazioni del cluster MinIO ripristinato e delle eventuali anomalie o problemi che potrebbero verificarsi dopo il ripristino degli snapshot.

7 GESTIONE E MANUTENZIONE DELLE BASI DI DATI

7.1 DATABASE PRINCIPALE

Questa sezione descrive la gestione e la manutenzione del database principale del backend MySQL. Le informazioni fornite coprono diversi aspetti, tra cui la configurazione del database, il ripristino, il clustering, il backup e le operazioni di migrazione.

7.1.1 CONFIGURAZIONE DEL DB

Per garantire prestazioni ottimali, il server MySQL deve essere configurato correttamente. In particolare, il file di configurazione (/etc/my.cnf) deve essere impostato in modo appropriato. Di seguito sono riportate alcune raccomandazioni generali per la configurazione del file:

- Assegnare molta RAM e vCPU al server. Nel caso di IDROCAP, si utilizzano 16 vCPU e 32 GB di RAM.
- Impostare la dimensione del buffer pool di InnoDB in base alla quantità di RAM disponibile. Una buona regola generale è utilizzare il 70% della RAM totale per un server dedicato, altrimenti il 10%.
- Abilitare il logging delle modifiche al binary log tra i backup, rimuovendo il commento dalla riga log_bin.
- Regolare le dimensioni dei buffer per join, ordinamento e lettura casuale in base alle esigenze, sperimentando per trovare i valori ottimali.

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

7.1.2 MANUTENZIONE PREVENTIVA DEL DB

Per prevenire problemi futuri e garantire le prestazioni ottimali del database, è importante seguire alcune pratiche di manutenzione preventiva:

- Backup regolari - effettuare backup regolari del database per garantire la sicurezza dei dati e facilitare il ripristino in caso di problemi. È consigliabile programmare backup giornalieri, settimanali e mensili, a seconda delle esigenze del sistema.
- Monitoraggio delle prestazioni - utilizzare strumenti di monitoraggio per verificare le prestazioni del database e identificare eventuali problemi prima che diventino critici. Tenere d'occhio le metriche come il tempo di risposta delle query, l'utilizzo della CPU e della memoria.
- Ottimizzazione delle query - rivedere periodicamente le query per assicurarsi che siano efficienti e non creino problemi di prestazioni. Valutare l'aggiunta di indici alle tabelle per velocizzare le operazioni di ricerca e ridurre il tempo di esecuzione delle query.
- Pulizia del database - eliminare i dati obsoleti o non necessari per ridurre la dimensione del database e migliorare le prestazioni. Utilizzare strumenti come mysqlcheck o mysqldumpslow per analizzare il database e identificare le aree che possono essere ottimizzate.
- Aggiornamenti del software - mantenere il software MySQL aggiornato all'ultima versione stabile per beneficiare delle correzioni di bug, delle patch di sicurezza e delle nuove funzionalità. Seguire le linee guida del fornitore per eseguire gli aggiornamenti in modo sicuro e senza interruzioni del servizio.

Seguire queste pratiche di manutenzione preventiva aiuterà a garantire un funzionamento stabile e sicuro del database, prevenendo problemi e riducendo al minimo i tempi di inattività del sistema.

7.1.3 MONITORAGGIO DELLE PERFORMANCE DEL DB

Per monitorare le performance del database MySQL, è possibile utilizzare alcuni strumenti e metodi. Di seguito sono elencati alcuni strumenti utili:

MySQL Workbench - questo strumento grafico offre una panoramica delle performance del database, permettendo di monitorare le query lente, l'utilizzo della CPU e della memoria, e altre informazioni importanti. Può essere scaricato dal sito web ufficiale di MySQL.

SHOW PROCESSLIST - questo comando SQL mostra informazioni sulle connessioni al database e sulle query in esecuzione. È utile per identificare eventuali problemi di performance.

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

Slow Query Log - MySQL registra le query lente nel "Slow Query Log". È possibile abilitare questo registro modificando il file di configurazione /etc/my.cnf. Una volta abilitato, il registro conterrà tutte le query che impiegano più tempo del valore specificato nella direttiva long_query_time.

Performance Schema - MySQL offre uno schema di performance che fornisce informazioni dettagliate sulle attività interne del server. Per abilitare lo schema di performance, modificare il file di configurazione /etc/my.cnf:

7.1.4 GESTIONE UTENTI, PRIVILEGI E SICUREZZA

Una gestione adeguata degli utenti e dei privilegi è fondamentale per la sicurezza e l'integrità del database MySQL. Seguire queste linee guida per gestire gli utenti e i loro privilegi in modo efficace:

1. Creare utenti specifici - evitare di utilizzare l'account amministratore root per le operazioni quotidiane. Creare utenti specifici con privilegi limitati per le diverse attività, come la gestione delle tabelle o l'esecuzione di query.
2. Assegnare privilegi in base al principio del minimo privilegio - concedere agli utenti solo i privilegi necessari per svolgere le loro funzioni. Ad esempio, se un utente deve solo leggere i dati, concedere solo il privilegio SELECT.
3. Utilizzare password sicure - imporre l'uso di password complesse e uniche per gli account utente e aggiornarle regolarmente. Utilizzare funzioni di hashing come PASSWORD() o SHA2() per memorizzare le password in modo sicuro nel database.
4. Monitorare e registrare le attività degli utenti - tenere traccia delle attività degli utenti e delle modifiche apportate al database. Utilizzare funzioni di registrazione come il log delle query lente o il log degli errori per identificare problemi o comportamenti sospetti.
5. Revocare i privilegi quando necessario - se un utente non ha più bisogno di un determinato privilegio o se un account utente non è più in uso, revocare i privilegi appropriati o eliminare l'account per ridurre i rischi di sicurezza.

Implementare una solida gestione degli utenti e dei privilegi è essenziale per proteggere il database MySQL e assicurare che solo gli utenti autorizzati possano accedere e modificare i dati.

Conclusione

La gestione di un database MySQL richiede attenzione a numerosi aspetti, tra cui la progettazione del database, l'ottimizzazione delle prestazioni, la manutenzione preventiva e la gestione degli utenti e dei privilegi. Seguendo le linee guida e le pratiche consigliate in questi ambiti, sarà possibile garantire un

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

funzionamento efficiente, sicuro e stabile del database, supportando al meglio le esigenze del sistema e degli utenti.

7.1.5 RIPRISTINO DEL DB

Se è necessario ripristinare il database, seguire la procedura riportata di seguito:

- Fermare il servizio MySQL: `systemctl stop mysqld`
- Modificare il file `my.cnf` (`/etc/my.cnf`) per abilitare i parametri `max_allowed_packet` e `wait_timeout` (rimuovere il commento dalle righe corrispondenti). Questi parametri consentono di gestire pacchetti di grandi dimensioni e di aumentare il tempo di attesa durante il processo di ripristino.
- Avviare il servizio MySQL: `systemctl start mysqld`
- Accedere a MySQL con l'utente root: `mysql -u root -p`
- Eliminare lo schema esistente: `drop schema jixel;`
- Ricreare lo schema vuoto: `create schema jixel;`
- Uscire dall'interfaccia MySQL: `exit`
- Eseguire il comando MySQL di ripristino: `mysql -u root -p jixel < mybigdump.sql`
- Attendere il completamento del dump. Il tempo richiesto dipenderà dalla dimensione del database e delle risorse del sistema.
- Fermare nuovamente il servizio MySQL: `systemctl stop mysqld`
- Commentare i parametri `max_allowed_packet` e `wait_timeout` nel file `my.cnf` per ripristinare le impostazioni precedenti.
- Riavviare il servizio MySQL: `systemctl start mysqld`

Dopo aver completato questi passaggi, il database dovrebbe essere ripristinato con successo. Ricordarsi di monitorare attentamente il sistema per assicurarsi che tutto funzioni correttamente dopo il ripristino.

7.1.6 MIGRAZIONE DEL DISCO CON CONSISTENZA DEI DATI E RIDUZIONE DEL DOWNTIME

Per migrare il disco di un database MySQL, mantenendo la consistenza dei dati e riducendo al minimo il downtime, è importante seguire una serie di passaggi accuratamente pianificati e testati.

1. Backup completo del database: Prima di iniziare, effettuare un backup completo del database per proteggere i dati in caso di problemi durante la migrazione. Verificare il successo del backup ripristinandolo su un ambiente di test separato.
2. Pianificazione della migrazione: Pianificare la migrazione durante un periodo di bassa attività per ridurre l'impatto sugli utenti.

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

3. Monitoraggio del database: Durante la migrazione, monitorare i processi e le prestazioni del database per individuare eventuali problemi o rallentamenti. Utilizzare gli strumenti di monitoraggio e diagnostica visti sopra per garantire la massima visibilità sul processo.
4. Sincronizzazione incrementale dei dati: Per ridurre il downtime e garantire la consistenza dei dati, utilizzare un approccio incrementale per sincronizzare i dati tra il disco originale e quello di destinazione. Questo processo implica il trasferimento delle modifiche apportate al disco originale al disco di destinazione in tempo reale, consentendo al database di rimanere operativo durante la migrazione.
5. Interruzione temporanea delle scritture: Quando i dischi sono sincronizzati, interrompere temporaneamente le scritture sul database e avviare un'ultima sincronizzazione per garantire che tutti i dati siano coerenti tra i dischi.
6. Switch al nuovo disco: Dopo aver verificato che i dischi siano sincronizzati, modificare la configurazione del database per puntare al nuovo disco e riavviare il servizio.
7. Verifica della correttezza e della funzionalità: Una volta completata la migrazione, testare il nuovo disco per assicurarsi che sia corretto e funzionante. Eseguire test di integrità e funzionalità per garantire che il sistema sia pronto per riprendere le normali operazioni.
8. Comunicazione del completamento: Informare gli utenti interessati che la migrazione è stata completata con successo e che il servizio è stato ripristinato.

Seguendo attentamente questi passaggi, è possibile eseguire una migrazione del disco con successo, mantenendo la consistenza dei dati e riducendo al minimo il downtime.

8 GESTIONE E MANUTENZIONE DELLE NOTIFICHE

Questa guida illustra la gestione e la manutenzione dei diversi microservizi che si occupano del dispatching di notifiche di vario titolo e tipologia in un ambiente basato su container Docker orchestrati da swarm.

8.1 PANORAMICA DEI SERVIZI

I servizi di messaggistica includono l'invio di Email, Telegram, e Push Notifications tramite chiamate a WebServices o chiamate SMTP. Le code sono organizzate come segue:

- `telegram`: Telegram
- `push_notifications`: Push Notifications
- `emails`: Emails
- `prod_messages_queue`: coda per inviare messaggi in ambiente di produzione

Programmazione FSC 2014-2020 Piano Operativo Ambiente			
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione			Pag. 16 di 27 Stato: Rev.04.00

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

8.2 CONFIGURAZIONE DEI SERVIZI DI MESSAGGISTICA/NOTIFICA

La configurazione fornita definisce diversi servizi per gestire l'invio di notifiche tramite email e push notifications. Ogni servizio ha un'immagine "idrocap-messaging:latest" e si connette a diverse reti per comunicare con altre componenti del sistema, come RabbitMQ, Redis e Postfix.

- Servizio di messaggistica email: Il servizio "messaging-email" gestisce l'invio delle email. Utilizza la coda "emails" di RabbitMQ e si connette alle reti "messaging_email", "backend" e "postfix". Anche in questo caso, le variabili di ambiente configurano la connessione a RabbitMQ, Redis.
- Servizio di messaggistica push: Il servizio "messaging-push" si occupa delle push notifications. Sfrutta la coda "push_notifications" di RabbitMQ e si connette alle reti "messaging_email", "backend" e "postfix". Le variabili di ambiente configurano la connessione a RabbitMQ, Redis e l'endpoint Hermes. Inoltre, il servizio monta un volume contenente il file "serviceAccountKey.json" necessario per l'autenticazione con le API delle push notifications.
- Servizio di monitoraggio della messaggistica: Il servizio "messaging-monitoring" espone un'interfaccia per il monitoraggio delle code e delle prestazioni del sistema di messaggistica. Ascolta sulla porta 8888 e si connette alle reti "messaging_monitoring" e "backend". Le variabili di ambiente configurano la connessione a Redis.
- Servizio Redis Monitoring: Il servizio "redis-monitoring" si connette alla rete "backend". Redis viene utilizzato per memorizzare le informazioni sul monitoraggio e le statistiche delle prestazioni del sistema di messaggistica.
- Servizio Postfix: Il servizio "postfix" si occupa dell'invio di email tramite il protocollo SMTP. Ascolta sulla porta 25 e si connette alla rete "postfix". Monta un volume contenente la configurazione DKIM per garantire la corretta firma delle email e la loro consegna nella casella di posta in arrivo dei destinatari. Le variabili di ambiente impostano l'utente SMTP e il dominio di posta.

Reti: Sono definite diverse reti per permettere ai servizi di comunicare tra loro. Le reti includono "postfix", "backend", "messaging_email", "messaging" e "messaging_monitoring".

Con questa configurazione, i servizi di messaggistica saranno in grado di gestire l'invio di notifiche tramite, email e push notifications, monitorare le prestazioni e utilizzare un server SMTP per l'invio di email.

8.3 GESTIONE E MANUTENZIONE

Per garantire il corretto funzionamento del sistema di messaggistica e monitoraggio, è importante eseguire regolarmente operazioni di manutenzione e controllare lo stato dei servizi. Di seguito sono riportate alcune linee guida per la gestione e la manutenzione del sistema.

Programmazione FSC 2014-2020 Piano Operativo Ambiente	 Fondo per lo Sviluppo e la Coesione	 PIANO OPERATIVO AMBIENTE	 MINISTERO DELLA TRANSIZIONE ECOLOGICA
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione			Pag. 17 di 27 Stato: Rev.04.00

 	Titolo	
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS

8.3.1 MONITORAGGIO DEL SISTEMA

Monitorare costantemente lo stato dei servizi e delle code per rilevare eventuali problemi o malfunzionamenti. Controllare i log dei servizi utilizzando il comando `docker logs -f <<nomedelcontainer>>` per verificare se il dispatching sta procedendo correttamente o se si verificano errori.

8.3.2 AGGIORNAMENTO DELLE IMMAGINI

Nel caso in cui sia necessario aggiornare le immagini dei servizi, modificare il file di configurazione (ad esempio, `docker-stack.yml`) con le nuove immagini e riavviare il sistema di messaggistica.

8.3.3 GESTIONE DELLE CONFIGURAZIONI SMTP

Per garantire che le email inviate non vengano considerate spam, è importante configurare correttamente le impostazioni di SPF, DKIM, DMARC e Reverse DNS. Verificare periodicamente che queste configurazioni siano corrette e aggiornate. Utilizzare strumenti come <https://www.mail-tester.com> per verificare la qualità delle email inviate.

8.3.4 BACKUP E RIPRISTINO

Poiché i dati di gestione non sono persistenti e i dischi dei container possono essere cancellati, è importante eseguire regolarmente il backup dei dati e delle configurazioni importanti, come le chiavi di autenticazione per le API delle push notifications e la configurazione DKIM per Postfix.

Seguendo queste linee guida, il sistema di messaggistica sarà in grado di gestire e inviare notifiche in modo efficiente e affidabile. Prestare attenzione ai problemi o malfunzionamenti rilevati durante il monitoraggio e risolverli tempestivamente per garantire un servizio di qualità agli utenti.

9 GESTIONE E MONITORAGGIO DELL'INFRASTRUTTURA

In un'infrastruttura basata su macchine virtuali, si possono gestire e monitorare i servizi utilizzando l'orchestratore Docker Swarm. Docker Swarm consente di creare e gestire un cluster di container Docker, semplificando il processo di distribuzione, scalabilità e gestione dei servizi. Ogni servizio all'interno dell'infrastruttura viene eseguito all'interno di un container Docker e viene monitorato tramite un agente Prometheus specifico per quel servizio.

Nell'ambito della gestione dell'infrastruttura basata su macchine virtuali, Docker Swarm e Prometheus, è importante prestare attenzione a determinati aspetti per garantire la stabilità, la scalabilità e la sicurezza del sistema nel suo complesso. Si suggerisce di considerare i seguenti punti chiave:

1. Assicurarsi che il monitoraggio dei servizi e delle risorse sia sempre attivo e funzionante, controllando regolarmente le dashboard di Prometheus e Grafana per identificare eventuali anomalie o tendenze preoccupanti.

 	Titolo	
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS

2. Prestare particolare attenzione alla gestione delle risorse e delle configurazioni, in modo da garantire che le macchine virtuali e i servizi distribuiti siano configurati correttamente e ottimizzati per le loro specifiche funzioni. Ciò include l'aggiornamento regolare dei parametri di configurazione e l'implementazione di best practice per la sicurezza e la performance.
3. Monitorare e gestire le notifiche di allarme generate da Prometheus, che possono includere avvisi relativi a situazioni critiche come il sovraccarico delle risorse, l'indisponibilità dei servizi o potenziali vulnerabilità di sicurezza. Quando si riceve un allarme, è fondamentale analizzare la causa alla radice del problema e risolvere tempestivamente per evitare impatti negativi sul sistema.
4. Valutare periodicamente le metriche di monitoraggio raccolte da Prometheus per identificare eventuali tendenze, anomalie o modelli emergenti. Questo aiuta a prevenire potenziali problemi prima che si verifichino e a ottimizzare proattivamente le risorse del sistema per garantire prestazioni ottimali e affidabilità.
5. Effettuare revisioni regolari del sistema, incluso l'orchestratore Docker Swarm, gli agent di Prometheus e le macchine virtuali, per garantire che le versioni del software e le configurazioni siano aggiornate e in linea con le ultime raccomandazioni e best practice del settore. Questo aiuta a mantenere l'infrastruttura resiliente alle minacce emergenti e a garantire che le prestazioni siano ottimizzate per le esigenze del progetto.

9.1 PROMETHEUS

Prometheus è un sistema di monitoraggio e allarme open-source che consente di avere una visione completa e dettagliata dei servizi in esecuzione nell'infrastruttura. Attraverso l'utilizzo di agenti specifici per ogni servizio, Prometheus raccoglie e memorizza metriche riguardanti le prestazioni, l'utilizzo delle risorse e la salute dei servizi. Queste metriche vengono poi esposte tramite un'interfaccia web o API per la visualizzazione e l'analisi.

Uno degli aspetti più importanti del monitoraggio dell'infrastruttura è l'abilità di ricevere notifiche tempestive in caso di situazioni critiche o problemi con i servizi. Prometheus integra un componente chiamato Alertmanager, che permette di configurare e gestire gli allarmi basati sulle metriche raccolte. Quando una condizione di allarme viene soddisfatta, Alertmanager invia notifiche ai sysadmin o agli operatori responsabili, tramite vari canali come email, Slack o altri sistemi di messaggistica opportunamente preconfigurati.

Per configurare gli allarmi in Prometheus, è necessario definire le regole di allarme nel file di configurazione di Prometheus e configurare l'Alertmanager con i dettagli sui destinatari delle notifiche e i canali di comunicazione. In questo modo, il team di sysadmin può essere tempestivamente informato su eventuali

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

problemi e agire rapidamente per risolverli, garantendo la continuità dei servizi e la qualità del servizio offerto.

Alcune informazioni a contorno della gestione stessa di Prometheus sono le seguenti:

1. Installare e configurare l'agente Prometheus su ogni nodo del cluster Docker Swarm. Per farlo, è necessario aggiungere il seguente blocco di configurazione nel file di configurazione di Prometheus (prometheus.yml):

```
scrape_configs:
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['<NODE-IP>:9100']
```

2. Utilizzare il comando docker service create per distribuire Prometheus su Docker Swarm. Assicurarsi di montare il file di configurazione prometheus.yml e di esporre la porta 9090 per accedere all'interfaccia web di Prometheus.

Esempio di comando:

```
docker service create --name prometheus --replicas 1 --network my-network -p 9090:9090 -v /path/to/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus
```

3. Configurare Prometheus per inviare notifiche al sysadmin di riferimento in caso di situazioni critiche. Per farlo, è necessario configurare Alertmanager e definire le regole di allarme nel file di configurazione di Prometheus.

4. Esempio di configurazione Alertmanager:

```
global:
  resolve_timeout: 5m
```

```
route:
  group_by: ['job']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 12h
  receiver: 'email-notification'
```

```
receivers:
  - name: 'email-notification'
```

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

email_configs:

```
- to: 'sysadmin@example.com'
  send_resolved: true
  from: 'alertmanager@example.com'
  smarthost: 'smtp.example.com:587'
  auth_username: 'username'
  auth_password: 'password'
```

9.2 ORCHESTRATOR CON DOCKER SWARM

Per creare un cluster Docker Swarm, è necessario inizializzare un nodo manager con il comando `docker swarm init`. Questo nodo manager sarà responsabile della gestione dell'intero cluster e delle decisioni di orchestrazione.

1. Esempio di comando: `docker swarm init --advertise-addr <MANAGER-IP>`
2. Per aggiungere nuovi nodi al cluster, utilizzare il comando `docker swarm join` sui nodi worker. Il comando generato durante l'inizializzazione del nodo manager può essere utilizzato per unire i nodi worker al cluster.
3. Esempio di comando: `docker swarm join --token <TOKEN> <MANAGER-IP>:<PORT>`
4. Per distribuire un servizio su Docker Swarm, è possibile utilizzare il comando `docker service create`. È importante specificare i parametri di configurazione, come il numero di repliche e le reti a cui il servizio deve appartenere.
5. Esempio di comando: `docker service create --name my-service --replicas 3 --network my-network my-image`.

9.3 ANSIBLE

Ansible è uno strumento di automazione open source che semplifica la gestione di configurazioni complesse, la distribuzione di applicazioni e l'esecuzione di attività. Utilizza un linguaggio dichiarativo basato su YAML (YAML Ain't Markup Language) per definire le attività di automazione, che sono organizzate in "playbook". Questi playbook consentono agli utenti di automatizzare le attività ripetitive, garantendo una gestione coerente e affidabile dei servizi applicativi.

La maggior parte delle operazioni di gestione e manutenzione dei servizi applicativi viene eseguita con i playbook Ansible grazie ai seguenti vantaggi:

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

- Consistenza - i playbook Ansible aiutano a mantenere la coerenza tra i diversi ambienti, riducendo le possibilità di errore umano e garantendo che i servizi applicativi siano configurati e gestiti in modo uniforme.
- Scalabilità - con l'aumento del numero di servizi applicativi, i playbook di Ansible facilitano la gestione e la manutenzione di un gran numero di servizi con un intervento manuale minimo, rendendola una soluzione ideale per la scalabilità delle operazioni.
- Controllo delle versioni - i playbook di Ansible possono essere archiviati in sistemi di controllo delle versioni, come Git, consentendo un facile monitoraggio, condivisione e collaborazione tra i membri del team. Questo facilita un approccio semplificato alla gestione degli aggiornamenti e delle modifiche dei servizi applicativi.
- Flessibilità - ansible supporta un'ampia gamma di moduli e plugin, che gli consentono di integrarsi con diverse piattaforme e strumenti. Questa flessibilità lo rende adatto alla gestione e alla manutenzione di diversi servizi applicativi.
- Riduzione dei tempi di inattività - automatizzando le attività, i playbook di Ansible aiutano a ridurre al minimo il rischio di downtime causato da errori manuali o configurazioni errate. Ciò garantisce che i servizi applicativi rimangano disponibili e funzionino in modo ottimale.

I playbook di Ansible svolgono un ruolo significativo nella gestione e manutenzione dei servizi applicativi, offrendo coerenza, scalabilità, controllo delle versioni, flessibilità e riduzione dei tempi di inattività. Sfruttando Ansible, le si può ottenere una gestione efficiente e affidabile dei servizi applicativi, garantendo un funzionamento continuo e ottimale.

Alla conclusione del progetto, saranno forniti tutti i playbook e i task Ansible indispensabili, poiché, trattandosi di script software autentici, devono essere sviluppati progressivamente mentre il sistema IDROCAP si espande verso la soluzione finale. Questo permetterà al team di gestione di integrare in modo impeccabile la soluzione di automazione all'interno dell'infrastruttura preesistente e di gestire e mantenere con efficacia i servizi applicativi. La disponibilità di queste risorse garantisce una transizione senza intoppi e mette a disposizione del team gli strumenti essenziali per la gestione a lungo termine dei servizi applicativi attraverso l'impiego del framework Ansible.

9.4 DOCKER REGISTRY & LABORATORY

In questa sezione verranno illustrate le procedure e le pratiche di gestione e manutenzione del Docker Registry e del Docker Laboratory. Lo scopo della sezione è di fornire le informazioni necessarie per lavorare con l'infrastruttura Docker e Harbor in modo efficiente ed efficace.

1. Struttura e organizzazione

Programmazione FSC 2014-2020 Piano Operativo Ambiente	  
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione	Pag. 22 di 27 Stato: Rev.04.00

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

Il Docker Laboratory e il Docker Registry sono due componenti chiave dell'infrastruttura. Il Docker Laboratory è una macchina virtuale utilizzata per creare e testare immagini Docker, mentre il Docker Registry è un'implementazione di Harbor che funge da repository per immagini Docker.

Le immagini Docker vengono create e testate nel Docker Laboratory. I file di configurazione e i sorgenti per la creazione delle immagini sono conservati nella directory `/usr/local/docker/`, che è un repository Git. È importante effettuare commit e push delle modifiche apportate ai file di configurazione.

Il Docker Registry è il componente responsabile della conservazione e distribuzione delle immagini Docker. Utilizza Harbor come piattaforma per la gestione del repository delle immagini. Harbor supporta anche la creazione di regole per la cancellazione automatica di immagini obsolete.

2. Creazione e gestione delle immagini Docker

Per creare un'immagine Docker nel Docker Laboratory, è necessario utilizzare uno script di build, solitamente chiamato `build.sh`, presente nella directory dell'immagine. Lo script può essere eseguito senza parametri o con parametri specifici, a seconda delle esigenze del progetto. Dopo aver creato un'immagine, è necessario assegnare un tag all'immagine, in modo da poterla caricare nel Docker Registry.

Per caricare un'immagine nel Docker Registry, è necessario utilizzare il comando `"docker push"` con il tag dell'immagine. Se si desidera caricare l'immagine su un altro registry, è necessario creare un tag specifico per quel registro e utilizzare il comando `"docker tag"` per assegnare il nuovo tag all'immagine.

3. Manutenzione del Docker Registry e del Docker Laboratory

Entrambi i componenti, Docker Registry e Docker Laboratory, dispongono di script per la cancellazione automatica delle immagini Docker obsolete. Tuttavia, è importante monitorare l'utilizzo dello spazio su disco e cancellare manualmente le immagini obsolete, se necessario.

Nel caso in cui il Docker Registry sia saturo e non sia possibile caricare nuove immagini o accedere all'interfaccia di Harbor, è necessario seguire la procedura descritta nella sezione "Docker Registry Saturo" del documento di input.

Se invece lo spazio su disco nel Docker Laboratory è esaurito, seguire la procedura descritta nella sezione "Docker Laboratory Saturo" del documento di input.

Seguono le procedure più comuni per la gestione del Docker Registry e del Docker Laboratory.

Programmazione FSC 2014-2020 Piano Operativo Ambiente	 	MINISTERO DELLA TRANSIZIONE ECOLOGICA
QUALSIASI COPIA CARTACEA DI QUESTO DOCUMENTO È SOLO PER RIFERIMENTO Riservato RTI - Amministrazione		Pag. 23 di 27 Stato: Rev.04.00

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

1. Docker Laboratory: questa VM serve per effettuare il building delle immagini Docker e per procedere al push verso Harbor. È raggiungibile in DNS come `docker-laboratory.dev.ies.solutions`.
2. Docker Registry: fare riferimento alla macchina virtuale dell'infrastruttura in cui è presente il servizio. Fa da repository per le immagini "fatte e finite".
3. Utilizzo di Docker Laboratory: tutti i file per la containerizzazione si trovano sotto `/usr/local/docker/`. Questa directory è un repository GIT, quindi è consigliabile committare e pushare dopo aver effettuato modifiche.
4. Building delle immagini: per eseguire il build, solitamente è presente un file `build.sh` all'interno delle cartelle di ogni immagine. Esso può essere semplice (senza parametri) o più articolato per consentire l'immissione di parametri. Per esempio, è possibile utilizzare il comando `./build.sh` oppure `./build.sh 20211014 master` (con data e branch come parametri).
5. Tagging delle immagini: gli script di build solitamente utilizzano come immagine di default un indirizzo configurato al suo interno. Taggando le immagini in questo modo, è possibile fare push verso il registro container (Harbor) associato al DNS. Se è necessario pushare verso un altro container, è possibile utilizzare un tag apposito.
6. Gestione delle immagini vecchie: sia su `docker-registry` che `docker laboratory`, ci sono script che cancellano automaticamente container più vecchi. Su `docker-registry`, le regole di cancellazione sono impostate in Harbor attraverso Policy > Tag retention. Tuttavia, alcuni container non hanno regole di cancellazione, quindi è importante ricordarsi di cancellare i container vecchi manualmente quando si fa push di nuovi container.
7. Garbage Collector: quando si cancella un container dal registry, lo spazio non si riduce immediatamente perché è necessario attendere che il Garbage Collector entri in gioco. Il Garbage Collector è impostato per attivarsi automaticamente ogni ora.

Gestione dello spazio su Docker Registry:

8. Se si satura lo spazio su Harbor, non sarà possibile eseguire il push o accedere a Harbor. In tal caso, seguire questi passaggi:
 - a. Accedere in SSH
 - b. Andare nella directory `/data` e cancellare il file `to_be_deleted_if_disk_is_full`.
 - c. Navigare nella directory `/root/harbor`.
 - d. Fermare il compose di Harbor con il comando `docker-compose down`.
 - e. Riavviare Harbor con il comando `docker-compose up -d`.
 - f. Accedere ad Harbor e cancellare alcune immagini.

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

- g. Forzare l'esecuzione del Garbage Collector premendo "GC NOW".
- h. Ricreare il file con il comando `fallocate -l 1G ./to_be_deleted_if_disk_is_full` nella directory `/data`.

Gestione dello spazio su Docker Laboratory:

9. Se si satura Docker Laboratory, il procedimento è simile a quello descritto per Docker Registry:
 - a. Andare nella directory `/home` e cancellare il file `to_be_deleted_if_disk_is_full` usando `sudo`.
 - b. Se necessario, riavviare il servizio Docker con il comando `sudo systemctl restart docker`.
 - c. Usare i comandi `docker images` e `docker rmi <<tagimmagine o nomeimmagine>>` per cancellare immagini vecchie.
 - d. Ricreare il file con il comando `sudo fallocate -l 1G ./to_be_deleted_if_disk_is_full` nella directory `/home`.

9.5 RIFERIMENTI UTILI

1. Docker: il sito ufficiale di Docker offre documentazione, tutorial e informazioni sulle immagini Docker e la loro gestione. (<https://www.docker.com/>)
2. Kubernetes: Kubernetes è un sistema di orchestrazione di contenitori open-source per l'automazione della distribuzione, la scalabilità e la gestione delle applicazioni containerizzate. (<https://kubernetes.io/>)
3. Ansible: il sito ufficiale di Ansible offre documentazione, tutorial e informazioni su come utilizzare Ansible per automatizzare la gestione delle infrastrutture. (<https://www.ansible.com/>)
4. Nginx: Nginx è un server web e proxy inverso di alta performance e bassa latenza. La documentazione può aiutare nella configurazione di load balancing e floating IP. (<https://www.nginx.com/>)
5. Let's Encrypt: Let's Encrypt è una Certification Authority (CA) gratuita e aperta che fornisce certificati SSL/TLS per garantire connessioni sicure. (<https://letsencrypt.org/>)
6. ELK Stack (Elasticsearch, Logstash, Kibana): ELK Stack è una soluzione di monitoraggio e analisi di log che consente di aggregare e visualizzare i log in modo centralizzato. (<https://www.elastic.co/what-is/elk-stack>)

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

7. OWASP: l'Open Web Application Security Project (OWASP) è un'organizzazione no-profit che si dedica all'identificazione e alla mitigazione delle vulnerabilità delle applicazioni web. (<https://owasp.org/>)
8. Agile Alliance: un'organizzazione no-profit che promuove l'adozione di pratiche di sviluppo Agile e fornisce risorse per comprendere e applicare i principi Agile. (<https://www.agilealliance.org/>)
9. MinIO: il sito ufficiale di MinIO offre documentazione, tutorial e informazioni su come configurare e gestire un cluster MinIO. (<https://min.io/>)
10. Kubernetes: Kubernetes è un sistema di orchestrazione di contenitori open-source per l'automazione della distribuzione, la scalabilità e la gestione delle applicazioni containerizzate. (<https://kubernetes.io/>)
11. Prometheus: Prometheus è un sistema di monitoraggio e allerta open-source per applicazioni e infrastrutture. (<https://prometheus.io/>)
12. AWS SDK for PHP: la documentazione ufficiale del SDK AWS per PHP offre informazioni su come utilizzare il SDK per interagire con i servizi AWS, tra cui MinIO. (<https://aws.amazon.com/sdk-for-php/>)
13. Erasure Coding: questa pagina di Wikipedia fornisce una panoramica sull'Erasure Coding e come viene utilizzato nei sistemi di storage distribuiti. (https://en.wikipedia.org/wiki/Erasure_code)
14. MinIO GitHub Repository: il repository GitHub di MinIO offre il codice sorgente del progetto e ulteriori risorse sulla configurazione e l'utilizzo di MinIO. (<https://github.com/minio/minio>)
15. Console MinIO: la documentazione della console MinIO fornisce informazioni su come utilizzare l'interfaccia web per la gestione e il monitoraggio dell'infrastruttura MinIO. (<https://docs.min.io/minio/console/>)
16. MinIO SDKs: la documentazione degli SDK MinIO per diversi linguaggi di programmazione, come Python, Java, JavaScript, .NET e altri. (<https://docs.min.io/minio/index.html>)
17. MySQL Documentation: <https://dev.mysql.com/doc/>
18. MySQL Workbench: <https://www.mysql.com/products/workbench/>
19. MySQL Performance Schema: <https://dev.mysql.com/doc/refman/8.0/en/performance-schema.html>
20. MySQL Slow Query Log: <https://dev.mysql.com/doc/refman/8.0/en/slow-query-log.html>

 	Titolo	RTI: Leonardo - IBM – Sistemi Informativi – Onit – Prisma – Deas - Reply PS
	Proprietario: Autorità di Bacino del Distretto Idrografico della Sicilia – RTI Autore: RTI Data rilascio: 14/06/2024 Codice: AQDML3IRCP01SUM02 Revisione: 04.00	

21. MySQL Backup and Recovery: <https://dev.mysql.com/doc/refman/8.0/en/backup-and-recovery.html>
22. MySQL User Management: <https://dev.mysql.com/doc/refman/8.0/en/user-account-management.html>
23. <https://www.mail-tester.com> - uno strumento per verificare la qualità delle email inviate e testare le configurazioni SPF, DKIM, DMARC e Reverse DNS.
24. RabbitMQ: è un message broker open source che implementa il protocollo Advanced Message Queuing Protocol (AMQP) (<https://www.rabbitmq.com/>).
25. Redis (REmote DIctionary Server): è un datastore in-memory open source che può essere utilizzato come database, cache o message broker (<https://redis.io/>).
26. Postfix: è un Mail Transfer Agent (MTA) open source che consente di inviare e ricevere email utilizzando il protocollo SMTP (Simple Mail Transfer Protocol) (<http://www.postfix.org/>).
27. Docker Swarm: Docker - Docker Swarm Documentation. (n.d.). Retrieved from <https://docs.docker.com/engine/swarm/>
28. Prometheus: Prometheus - Monitoring system & time series database. <https://prometheus.io/>
29. Grafana: Grafana - The open observability platform. <https://grafana.com/>
30. Alertmanager: Prometheus Alertmanager. <https://prometheus.io/docs/alerting/latest/alertmanager/>
31. Node Exporter: Prometheus - Node Exporter. https://github.com/prometheus/node_exporter
32. Ansible: Ansible is Simple IT Automation. <https://www.ansible.com/>
33. Harbor: Harbor - An open-source trusted cloud-native registry project. <https://goharbor.io/>