

## Doc history

Rev	Date	Author	Description
00	01/15/2025	Marco Selva	Create project template
01	02/05/2025	Marco Selva	Add meeting update
02	02/26/2025	Marco Selva	Update project description and link architecture doc
03	03/27/2025	Marco Selva	Add MoM 2 meeting

## Linked doc

Cod	Link	Name	Description
A2	<a href="#">link</a>	[A2] - Architecture	Architecture and developer doc
A3	<a href="#">link</a>	[A3] - Energy and cost simulation	Excel with energy and cost estimation and PC selection

## 20250327 – 2 meeting

Iniziare a formattare

- Ubutnu server
  - o Dobbiamo formattare i PC
  - o Preparare chiavetta
- Use case
  - o Tool interni per BM o staff
  - o Siti static
  - o Per il moment non serve scalabilita
- Possiamo optare per soluzione più leggere con solo Docker, per partire

Per accesso da remoto:

- Cloudfare Tunnel
  - o Supporta OAuth
  - o Esporre porte direttamente, tramite sottodominio di rete
- Valutazione di router
- Dominio per il momento uno che ci da [@Flavio Renzi](#)
  - o [@Marco Selva](#) to buy router

GitLab

- Perchè usare questo?
- Si può semplificare con GitHub,
  - o usare questo per pushare il codice e non fare il deploy interno

STEP 0:

- Installazione minima
  - o Ubutnu server CHIAVETTA [@Marco Selva](#)
    - 1 partizione minima da 50/100 Gb
    - Lasciamo spazio vuoto
  - o Router e rete online con 1 PC
    - Router fare
    - Costo del dominio
  - o Docker
  - o Accesso da remoto
    - Cloudfare Tunnel
    - [@Flavio Renzi](#) preparare breve doc per configurazione
- Stima con costi mantenimento

- Con presa di consumi
  - Presa moitor o shelly per stima consumi medi effettivi

## 2 Aprile

- Ubutnu server chiavetta
- Router

### Storage Server - STANDBY

- DB storage persistente
- Con [MicroCeph](#): partizione con partizione ridondata
- 3 worker node:
  - Partizione di sistema
  - Partizione di sati, gestita da MicroCeph, tutti i dati sono replicati.
    - Viene visto come un disco montato

# Hosting and Cloud project

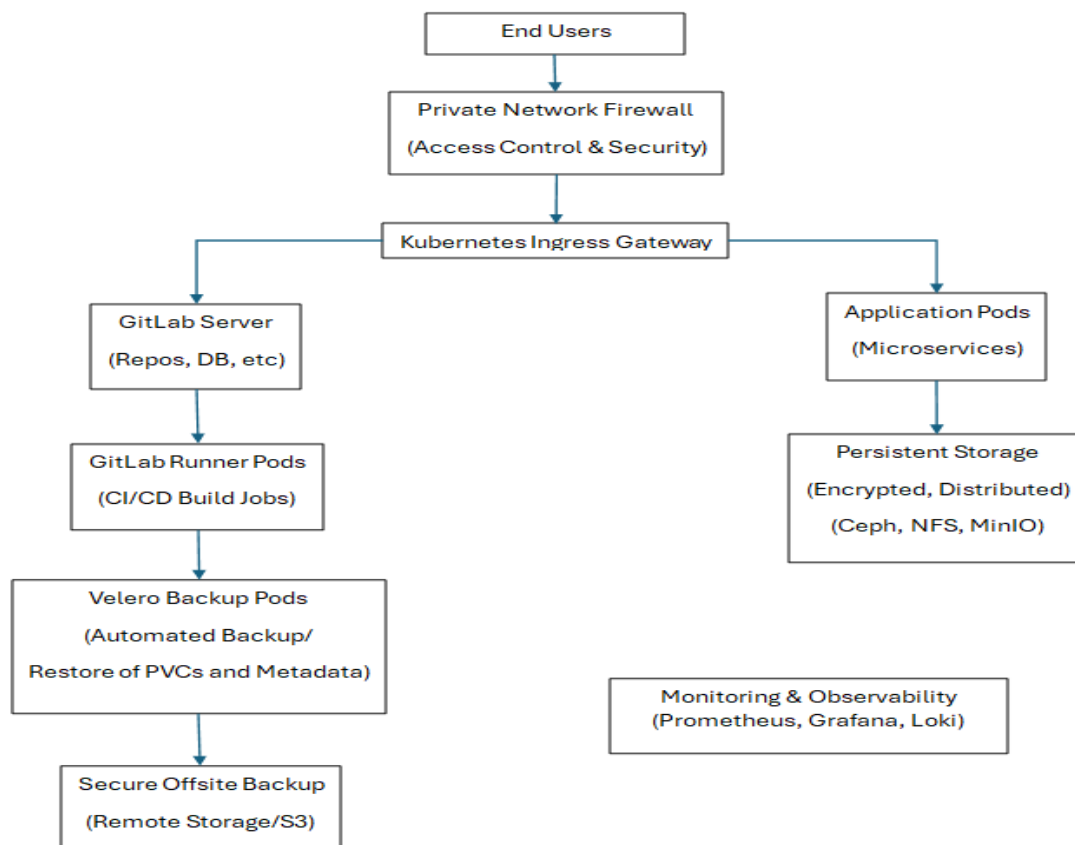
## Project Description:

This initiative gives new life to legacy company PCs by transforming them into a powerful **on-premises private cloud infrastructure**, powered by Kubernetes and GitLab.

By leveraging existing hardware, we create a scalable, fault-tolerant environment that supports essential applications and continuous integration processes, optimizing scalability and cost-efficiency.

More than just a technical solution, this infrastructure will **drive the company's internal projects**, centralizing development, testing, and operational workflows. It also opens the door for hosting client demos, **showcasing our capability to deliver secure, sustainable custom solutions**.

Through the reuse of company assets, we **reduce environmental impacts** and operational costs, paving the way for innovation, sustainability, and stronger data security initiatives.



## Roles and Responsibilities:

### Group Leader: Silvio Mario Pastori

- *Main Responsibilities:*
  - Coordinate group activities.
  - Ensure the achievement of set objectives.
  - Facilitate communication among team members.

### Developers: Flavio Renzi, Marco Selva

- *Main Responsibilities:*
  - Develop code according to provided specifications.
  - Collaborate in drafting technical documentation.
  - Participate in design reviews and provide feedback.
  - Test and validate implemented solutions.

## Specific Objective (End)

- Develop and release the **Cloud Management Module by end of Q2 2025**.
- Serve other internal projects as soon as possible, 2-3 month

## Priority Goal

- Ensure seamless integration of the Cloud Management Module with existing internal systems—supporting continuous integration pipelines and critical applications—without introducing regressions.

## Tasks and Planning

### Task 1: Requirement Analysis

- **Objective:** Gather and analyze requirements for the Cloud Management Module.
- **Activities:**
  - Interview stakeholders and end-users
  - Document functional and non-functional requirements
- **Deliverable:** Detailed requirements specification document
- **Estimation:** X weeks

### Task 2: Architecture & Design

- **Objective:** Define the system architecture for scalability and fault tolerance.
- **Activities:**
  - Create system diagrams and interface designs
  - Select appropriate hardware and software configurations
- **Deliverable:** Architectural design document and prototype diagrams
- **Estimation:** X weeks

### Task 3: Implementation & Integration

- **Objective:** Develop Module X and integrate it with the internal cloud environment.
- **Activities:**
  - Code development and unit testing
  - Continuous integration via GitLab CI/CD
- **Responsible Person:** Development Team
- **Deliverable:** Source code and integration test reports
- **Estimation:** X weeks

### Task 4: Testing & Quality Assurance

- **Objective:** Validate the module's functionality and performance.
- **Activities:**
  - Execute functional, regression, and performance tests
  - Compile and resolve any identified issues
- **Deliverable:** Comprehensive test report and bug resolution logs
- **Estimation:** X weeks

## Task 5: Deployment & Training

- **Objective:** Deploy Module to production and train relevant teams.
- **Activities:**
  - Final deployment and system integration
  - Conduct training sessions and prepare user manuals
- **Deliverable:** Live deployment, training materials, and session feedback
- **Estimation:** X weeks

## Expected Outputs

- **Source Code:** Fully functional code for the Cloud Management Module
- **Technical Documentation:** Detailed design, architecture, and user guides
- **UI Designs:** Prototypes and final designs for user interfaces
- **Test Reports:** Comprehensive test documentation with performance and regression details

## Necessary Resources

### Tools:

- **IDE:** Visual Studio
- **Repository:** GitHub for version control and collaboration
- **Other Tools:** Kubernetes management tools, GitLab CI/CD, Docker, and monitoring tools (e.g., Prometheus)

### Budget:

- Anticipated costs include hardware upgrades, software licenses, and possible cloud testing expenses.

### Training:

- Targeted sessions for Kubernetes, GitLab CI/CD, and cloud infrastructure management will be required.

## Feedback and Evaluation

### Success Metrics:

- Project completion within the planned timeline
- High stakeholder satisfaction
- Absence of critical bugs in production

### Evaluation Process:

- Final review meeting with all project team members
- Collection of both individual and collective feedback through a retrospective analysis



# Update

## 20250205 – First Meeting

- Docker compose
- Kubernetes + HELM (for environment variables)
- K9s
- Grafana for monitoring
- On the hardware side: 4 PCs

Steps to do:

- Hardware request
  - Flavio already ON
  - Our switch for ...
- Waiting for Silvio's documentation
  - Ubuntu (Mate) setup
- Network setup
  - Determine whether to be on the corporate network or not
  - Also evaluate the connection
- Tools:
  - Docker swarm (<https://github.com/docker-archive/classic-swarm>)
  - Microceph (<https://github.com/canonical/microceph>)

