

OpenLaaS Project Charter

1. Essentials

Project Name

OpenLaaS (Open Lab as a Service)

Team Members

- Evan Parker – Bachelors in Computer Science - (System Architecture & Backend)
- Dan McCarthy – Bachelors in Information Technology - (Frontend Development)
- Matt Gee - Bachelors in Computer Science - (Full Stack Development)
- Kestutis Biskis – Bachelors in Computer Science - (Frontend Development & Database Systems)
- Alex Houle – Bachelors in Computer Science - (Backend API Development & Frontend Integration)

Target Audience

Students, faculty, and researchers at the University of New Hampshire who need rapid access to lab infrastructure (bare metal servers, VMs, containers) for coursework, experimentation, and cybersecurity training.

2. Long-Term Vision

Vision

Empower students and administrators with an open, self-service platform to deploy and manage infrastructure resources in a unified and automated way, lowering technical barriers for hands-on learning and research.

Value

OpenLaaS streamlines provisioning for both users and admins:

- **For students:** enables quick, self-service VM and server deployments.
- **For admins:** centralizes management, reduces manual setup, and integrates with existing systems (FreeIPA, Proxmox, etc.).

3. Personas

Persona 1 - Evan - System Architect

- **Bio:** The System Architect responsible for designing and maintaining the overall infrastructure. Primary system administrator for the cybersecurity lab, oversees

FreeIPA, Proxmox, Bare-Metal servers and their BMCs. Interested in automation and security while providing robust solutions.

- **Motivation:** Wants to automate resource allocation to save time on repetitive setup, additionally wants to enable students and faculty to independently manage their lab resources while promoting projects.
- **Goals:** Streamline user onboarding, reduce provisioning errors, and ensure uptime across lab infrastructure. Create a safe and effective application.

Persona 2 - Dan - Student

- **Bio:** A student who wants to deploy their personal projects on real hardware. Isn't as focused on the technical details or computation power of the systems, is more focused on being able to easily get their projects running and accessible to the public. Doesn't want to deal with complicated UI or deployment process for their projects, and would prefer if software was intuitive to use.
- **Motivation:** Wants to use a service that makes interacting with the underlying systems as simple as possible without needing a lot of technical skills. Wants to focus on their projects, not dealing with infrastructure.
- **Goals:** Setup their projects on the provided systems, and share the things they've made publicly.

Persona 3 - Matt - Researcher

- **Bio:** A researcher in need of a reserved system in which they can utilize computation power they do not have access to or are unable to procure. Interested in being able to reserve / book resources from the service with the ability to manage booked resources as if the resource was on their network.
- **Motivation:** Wants to be able to book resources to be used for approved personalized projects, additionally may need additional resources on-demand with control over the systems.
- **Goals:** Streamline ability for onboarded users to book hosts and provide interfaces in order to execute Redfish or IPMI commands on the allocated systems.

Persona 4 - Kestutis - Instructor

- **Bio:** Student interested in networking programming looking to create a network communication protocol that will require multiple hosts on a network.
- **Motivation:** wants to have control of multiple hosts on a network for testing with minimal setup.
- **Goals:** enable users to easily create multiple hosts on a network that can communicate with each other.

Persona 5 - Alex - Coach

- **Bio:** A chess coach who host their own chess engine to tests students. The engine is run locally to have full control and no one else can mess with it.

- **Motivation:** Want to monitor long term progress of their athletes.
- **Goals:** Provide compute to Coach to record statistics about each student their progress.

4. Epics and Features

Epic: Dashboard & Proxmox Integration

Feature 1 – User & Resource Dashboard

- Unified dashboard for users to log in, view resources, and submit booking requests.
- Admins can view and approve bookings, manage infrastructure, and monitor server states.

Feature 2 – Virtualization Integration

- Connects OpenLaaS to Proxmox API for automated VM/CT creation.
- Includes control actions: start, stop, reboot, and delete.

Epic: Bare Metal Provisioning

Feature 3 – Bare Metal Provisioning

- Integrates with IPMI and BMC interfaces for bare metal server management.
- Supports automated ISO repository for automated OS installation.
- Supports automated OS deployment and hardware monitoring.

Feature 4 – Monitoring and Logging

- Provides real-time monitoring of hardware and virtual resources.
- Implements centralized logging for system events and user activities.
- Supports alerting and notification for critical events.

5. Stories

Feature 1 – User & Resource Dashboard

- As the System Architect, I want to provide a way to show and manage resources to facilitate user bookings and resource allocation so that users can easily access the resources they need.
- As a Student, I want to view and book available resources so that I can get my projects deployed easily.
- As a Researcher, I want to be able to reserve and manage computational resources for my projects so I can focus on other aspects of my work instead of spending time manually setting up hosts.
- As an Instructor, I want to monitor health and status of my students hosts so that it can be used for grading.

- As a Chess Coach, I want to monitor the health and status of the bare metal servers I am using, so that I can ensure uptime during ongoing training sessions.

Feature 2 – Virtualization Integration

- As a System Architect, I want to integrate with Proxmox API to automate VM and container lifecycle management so that users can quickly deploy virtualized environments.
- As a Student, I want to create, start, stop, and delete virtual machines so that I have a dedicated system to run my project on.
- As a Researcher, I want to manage virtualized environments for my computational tasks so me and my team are able to access identical resources we have individual control over.
- As an Instructor, I want to be able to easily create several virtual machines for low compute tasks so that I can make several lightweight hosts for classes.
- As a Chess Coach, I want to deploy my chess analysis tools on virtual machines with minimal setup, so that I can quickly spin up testing environments for matches.

Feature 3 – Bare Metal Provisioning

- As a System Architect, I want to integrate with IPMI and BMC interfaces for bare metal server management so that users can deploy projects on physical hardware.
- As a Student, I want to use bare metal servers with minimal setup so that I don't have to deal with the technical parts of the underlying system.
- As a Researcher, I want to be able to provision bare metal system so I am able to work on low level components of systems (such as network interface functions) that may be not be accessible in a virtualized environment.
- As an Instructor, I want to easily create bare metal hosts so that I can allow students to use them for assignments.
- As a Chess Coach, I want to access detailed hardware metrics and logs for troubleshooting, so that I can identify performance issues affecting simulations or training workloads.

Feature 4 – Monitoring and Logging

- As a System Architect, I want to be able to safely monitor and log system activities so that I can ensure system integrity and troubleshoot issues.
- As a Student, I want to receive alerts for critical system events so that I know if anything happens to my project.
- As a Researcher, I want centralized logs for troubleshooting and analysis so my team can more efficiently create reports on experiments and ensure the systems running them are working as expected.
- As an Instructor, I want to access logs and alerts related to my resource usage so that I can grade my students.
- As a Chess Coach, I want to configure monitoring thresholds and alerts, so that I am notified when system resources are under heavy use during peak training times.

6. Product Backlog (GitLab)

Please See: <https://github.com/orgs/OpnLaaS/projects/1>

7. Notes / Future Work

- Add Bare Metal Provisioning System and Booking System integration.
- Expand test coverage and monitoring.