LoomAI

An Augmented Interface for FABRIC and Cross-Facility Experimentation

Komal Thareja
Senior Research Software Engineer · RENCI, UNC Chapel Hill

USC ISI Cybersecurity Seminar Series · July 10, 2026

INTRODUCTION

Who I am & why this talk

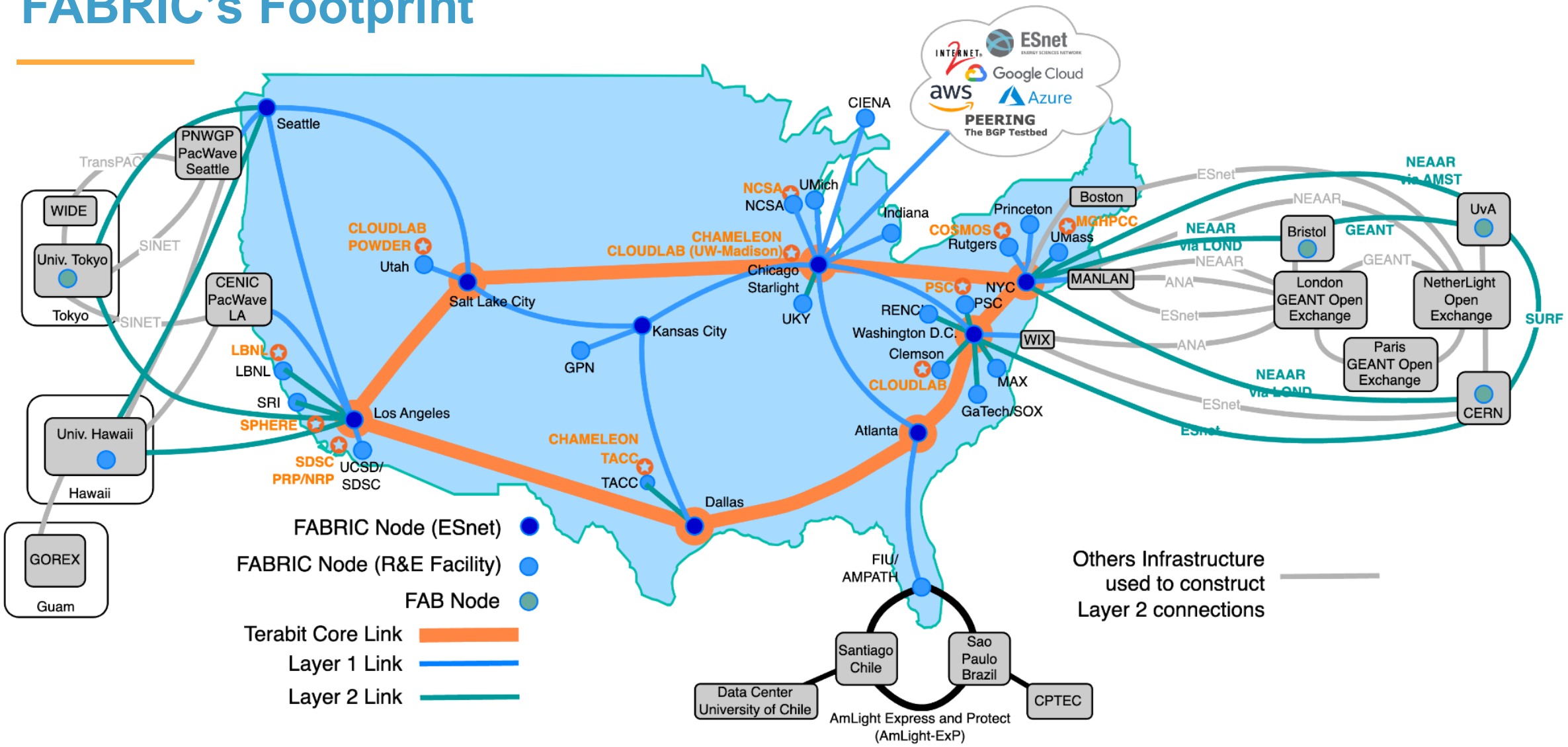
- Komal Thareja — Senior Research Software Engineer at RENCI (UNC Chapel Hill).
- Lead development of FABRIC's core platform: networking services, accelerators, distributed storage.
- Research: the convergence of distributed intelligence, multi-agent systems, programmable networks, and AI-driven scientific workflows.
- **LoomAI grew out of a simple observation:**
  - –Powerful testbeds are underused because the on-ramp is steep.
  - –Goal — make world-class infrastructure feel approachable, without hiding its power.

INTRODUCTION

FABRIC

an international NSF Mid-scale Research Infrastructure, empowers researchers to securely prototype and validate disruptive designs in wide-area networks. With its inclusion of advanced hardware uncommon in routers and switches, FABRIC facilitates experiments that closely emulate real-world production environments. By prioritizing connectivity with existing research and compute facilities, FABRIC enhances its usability and relevance.

FABRIC's Footprint



The problem we set out to solve

- **Today's On-Ramp: Powerful, but Code-First**
  - Experiments are defined as slices: nodes, components, and network topologies.
  - FABlib + Jupyter is flexible and scriptable, but assumes Python and API fluency.
  - Great for power users, but a steep wall for newcomers, classrooms, and cross-disciplinary teams.
  - Reproducing experiments means reading notebooks line by line, with no shared visual model.
  - Cross-facility work adds more SDKs, auth flows, and testbed-specific idioms.
  - Troubleshooting is fragmented across logs, tracebacks, notebooks, and docs.

- **Net effect:** time-to-first-experiment takes days, and every facility resets the learning curve.

THE IDEA

What is LoomAI?

An open-source, browser-based interface that turns experiment design into a visual, AI-assisted, cross-facility workflow — no notebook required.

Visual

Drag-and-drop topology editor, geographic map, live state — see your experiment.

AI-assisted

A built-in assistant designs, explains, and troubleshoots — on FABRIC & NRP GPUs.

Cross-facility

One interface across FABRIC and SPHERE — meta-slices span testbeds.

Open & shareable

Weaves and an Artifact Marketplace make experiments reusable assets.

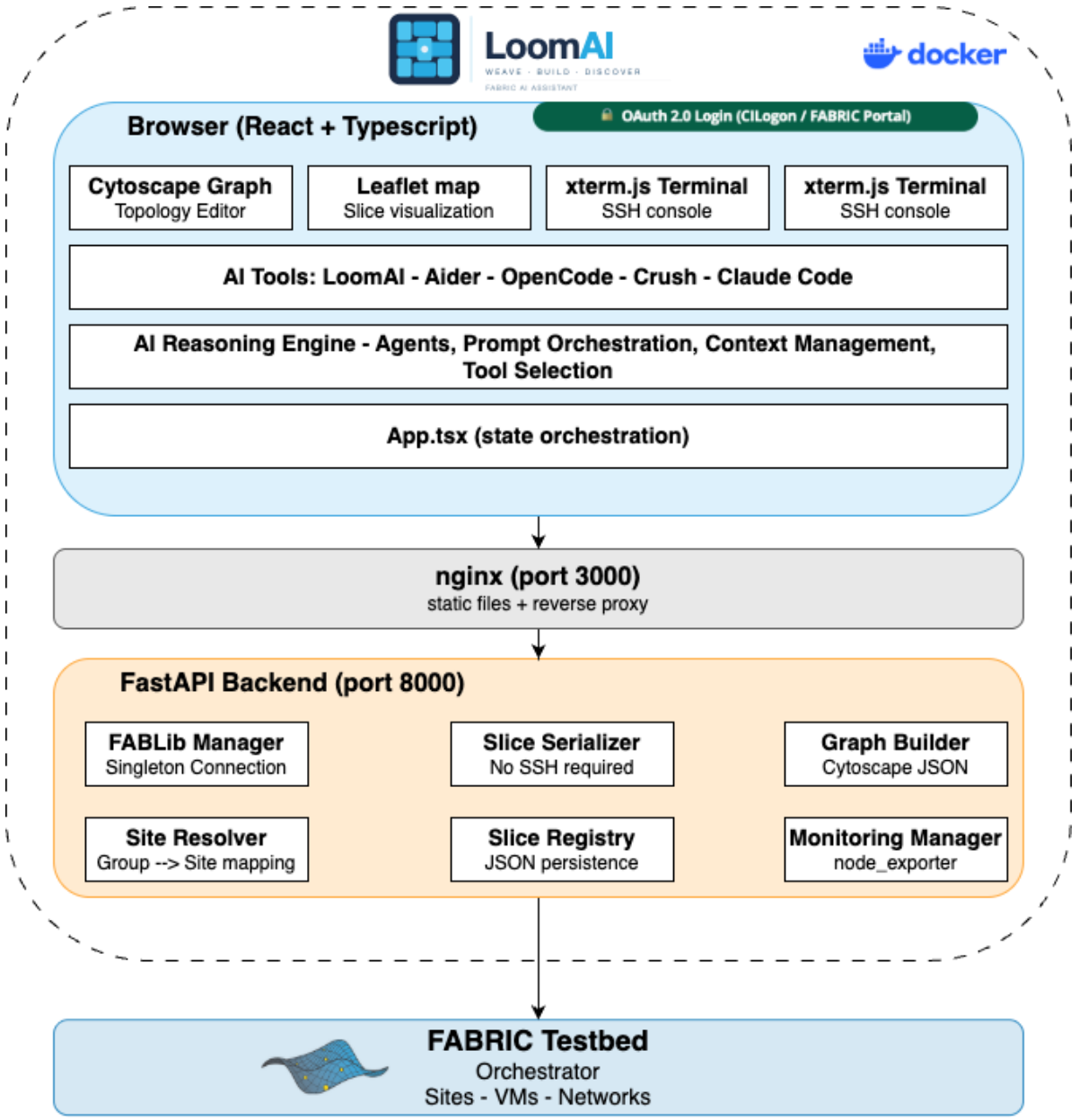
https://loomai.fabric-testbed.net/

HOW IT'S BUILT

Architecture at a glance

- **Frontend**
  - React 18 + TypeScript SPA
  - Cytoscape.js topology graphs
  - Leaflet.js geographic maps
  - xterm.js SSH terminals
  - CodeMirror 6 file editor

- **Backend**
  - FastAPI (Python, async)
  - FABlib Manager (direct API)
  - Graph Builder + Site Resolver
  - Monitoring Manager
  - LLM Integration (tool calling)



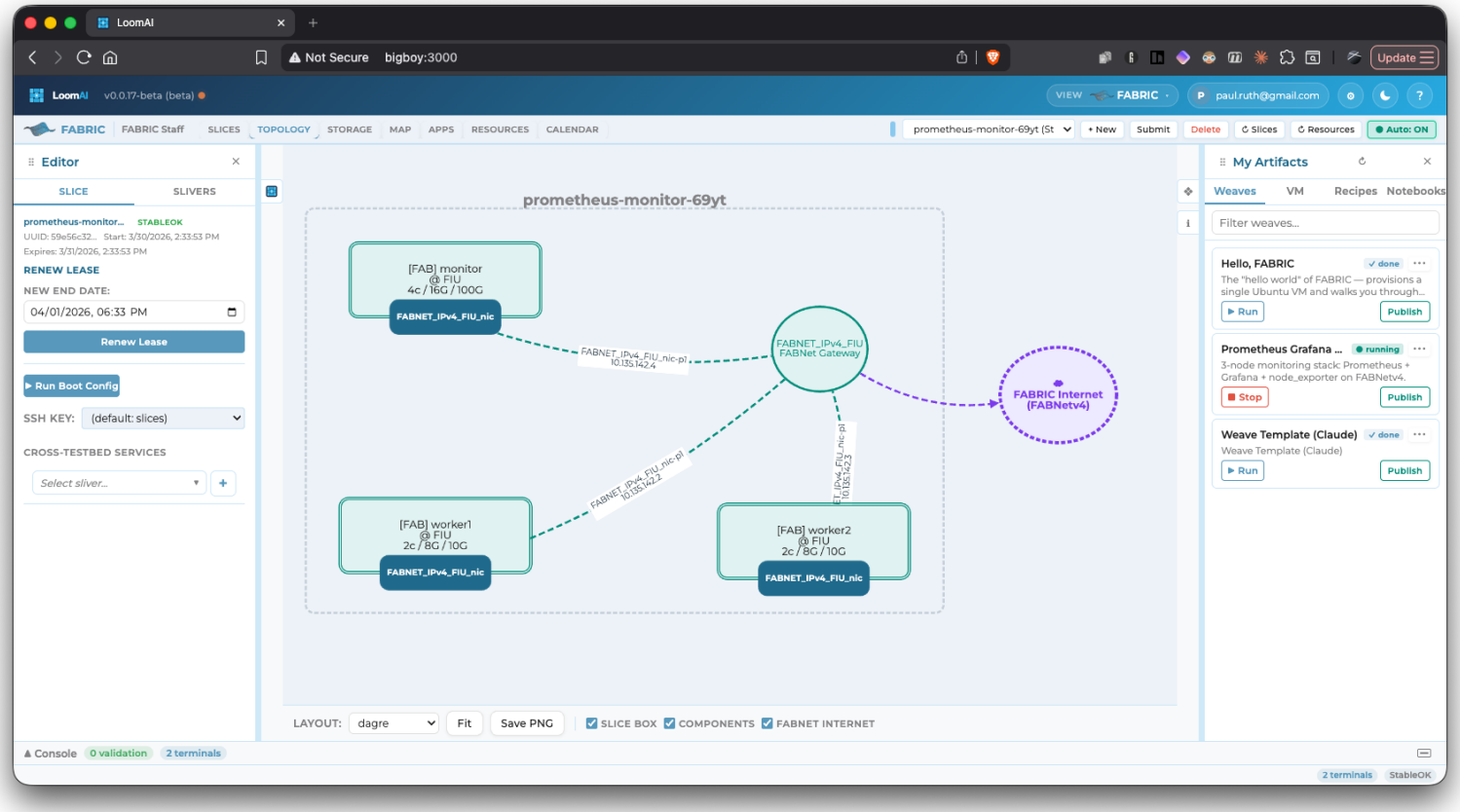
PART 1

The Visual Interface

See your experiment before you run it

Visual Topology Editor

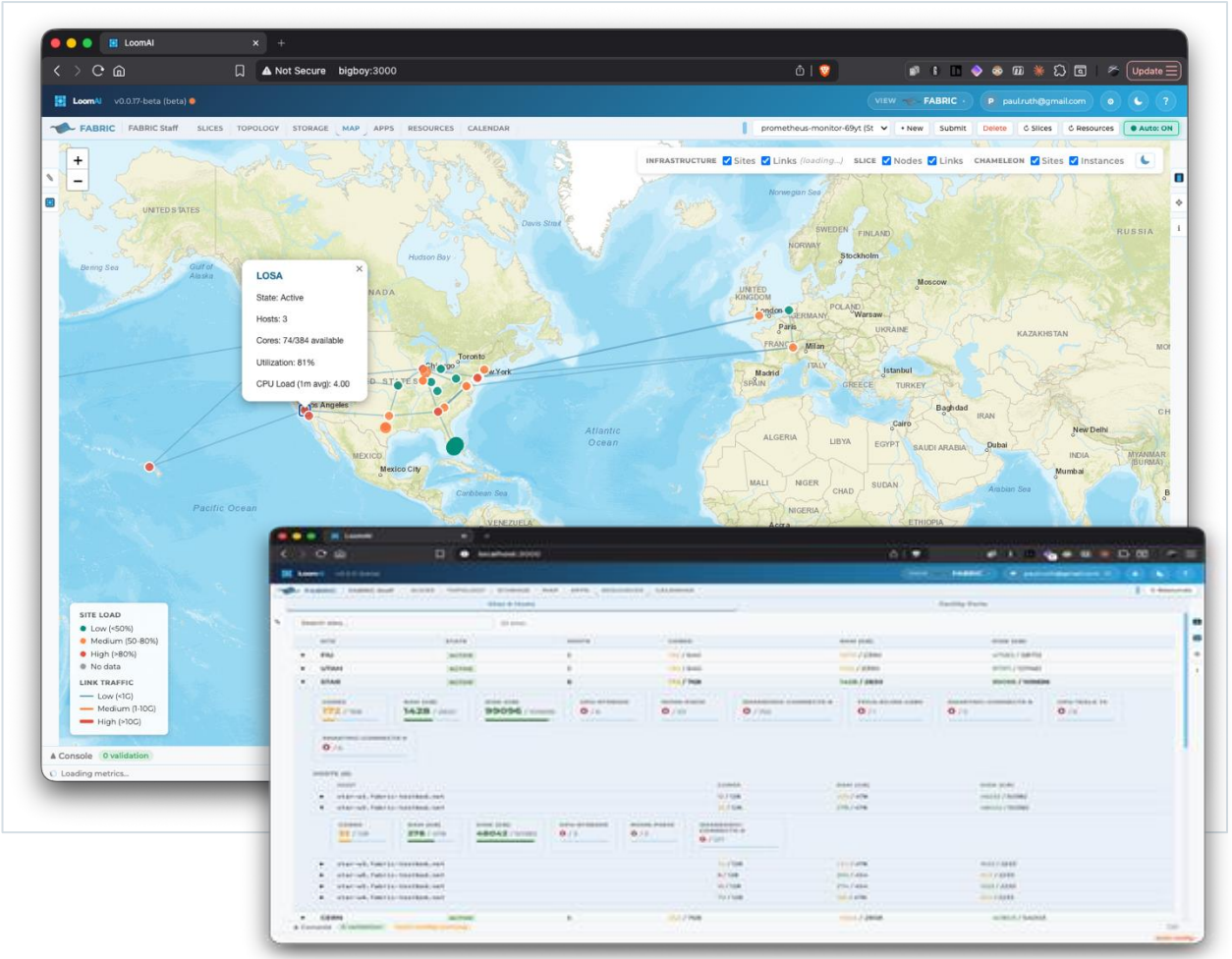
- **View Topologies**
  - Interactive Cytoscape.js topology canvas
  - Color-coded status indicators
- **Edit Slices**
  - Add/remove VMs
  - Add/remove network services
- **Manage Slices**
  - Submit, modify, delete slices



LoomAI topology editor — a 3-node monitoring slice

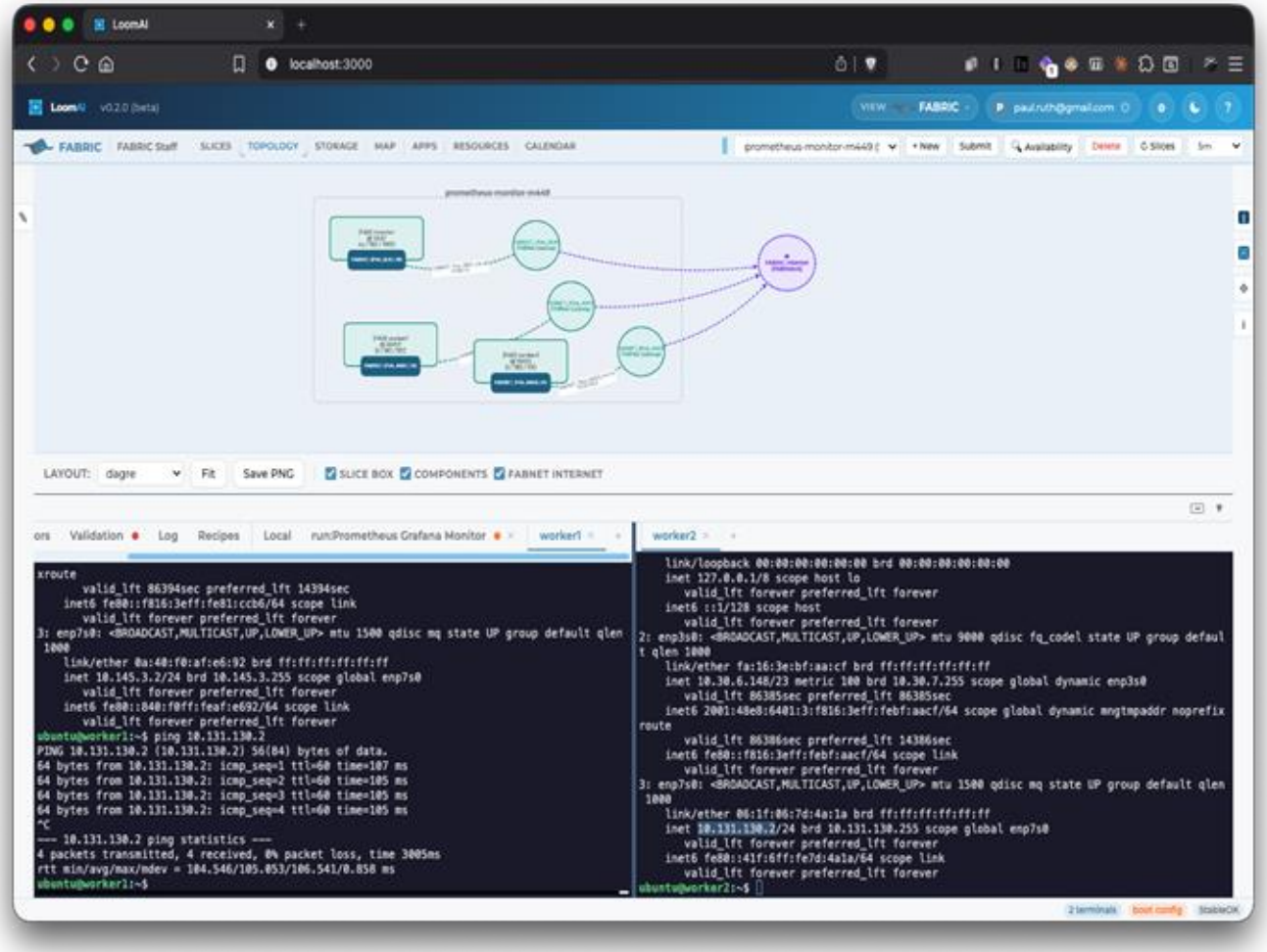
Geographic map & smart site resolution

- **Geographic Map**
  - Visually view sites
  - Colors indicate resource availability
- **Detailed Availability**
  - Browse and search detailed site and host resource availability
  - Data scraped from live infrastructure data



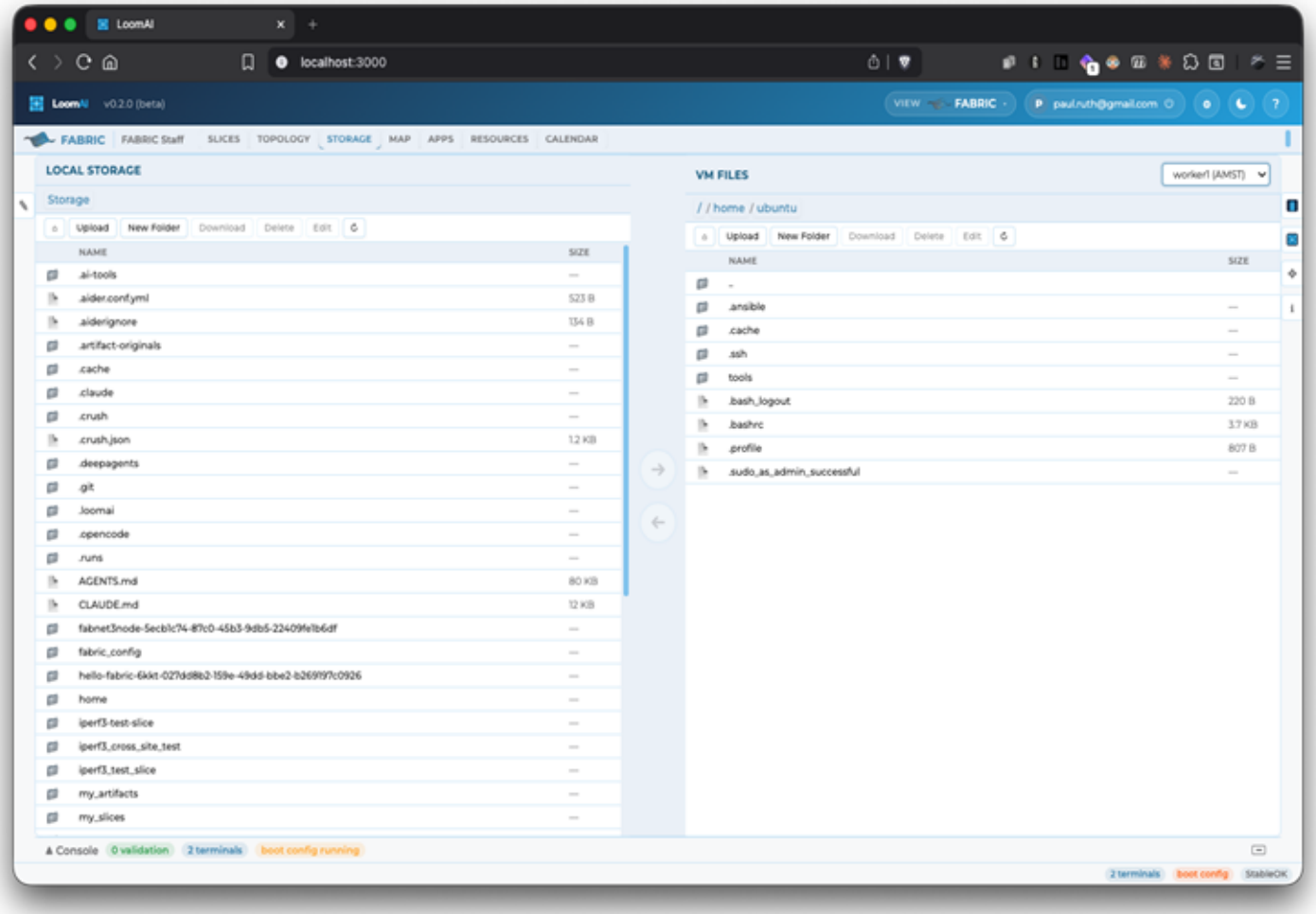
In Browser SSH Terminals

- **SSH Terminal**
  - Click to launch full terminals to any active VM
  - Multiple tabbed terminals possible
  - Automatically navigates bastion hosts



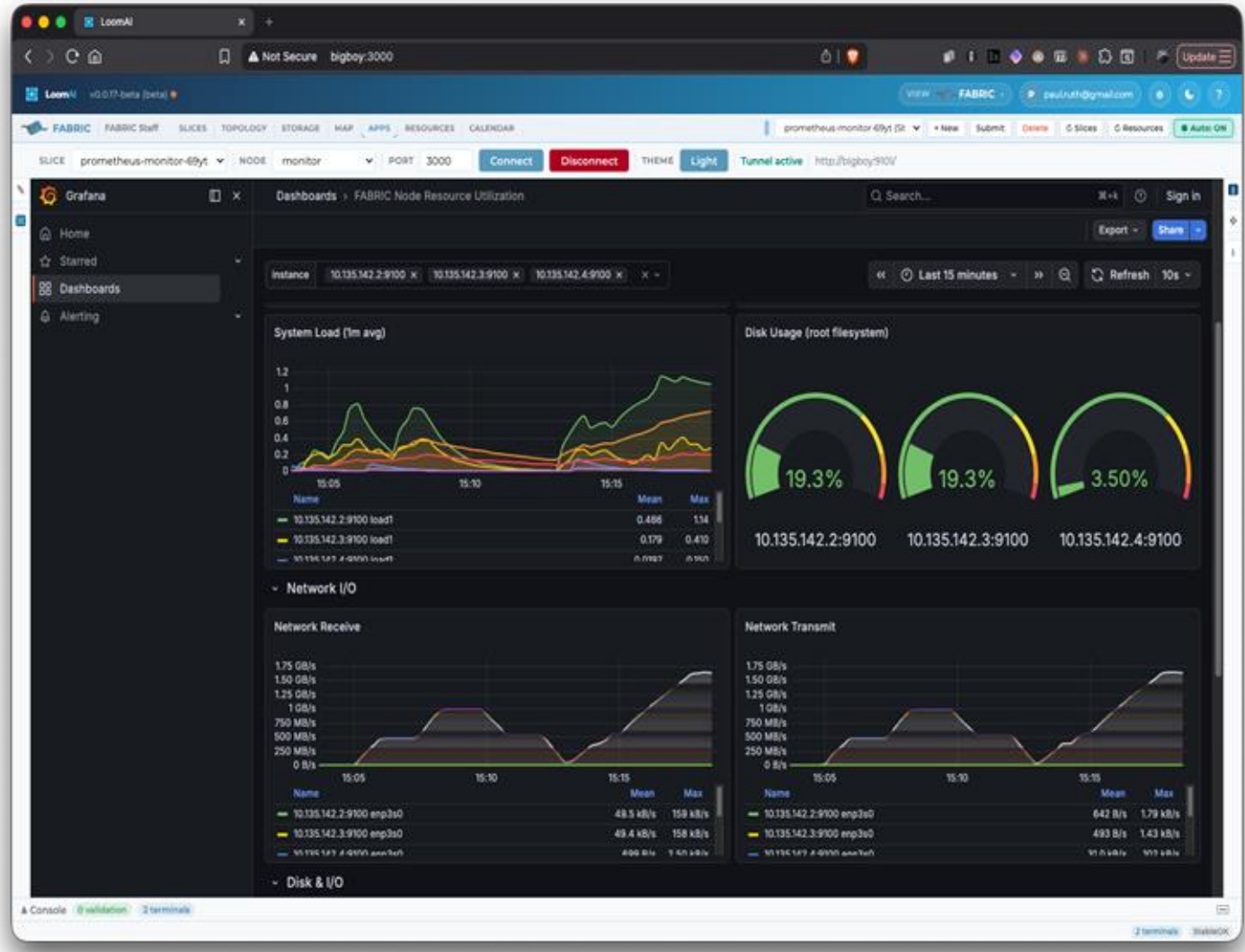
File Manager

- **Upload/download**
  - Simple drag/drop files to from active VMs
- **File editor**
  - Double click text files to edit in place.



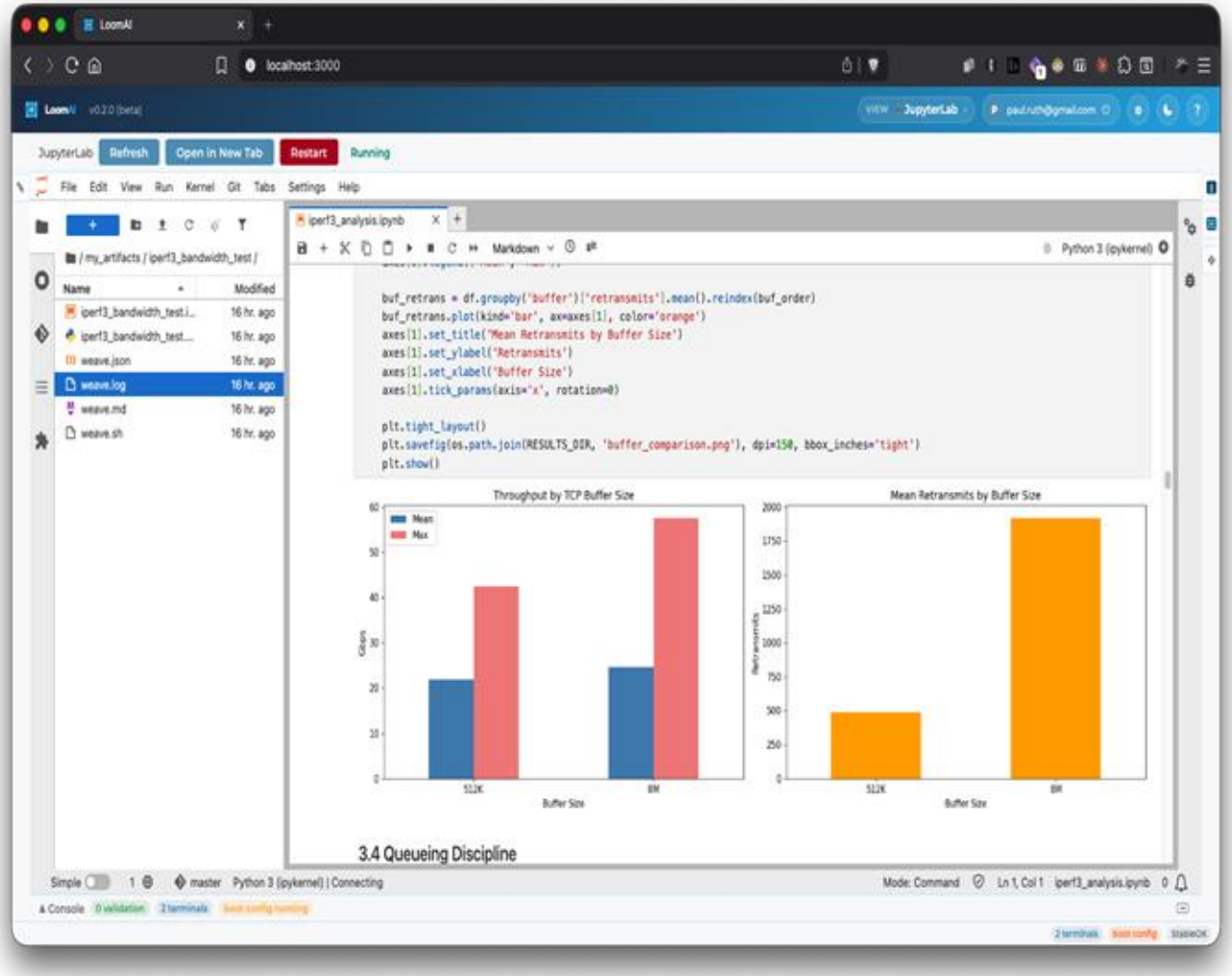
Web App Tunnels

- Expose web services from slice VMs
- Tunneled through bastion host automatically
- View web apps in ClientView panel
- No port forwarding or VPN required
- Supports Jupyter, Grafana, custom apps
- Dynamic port range: 9100–9199



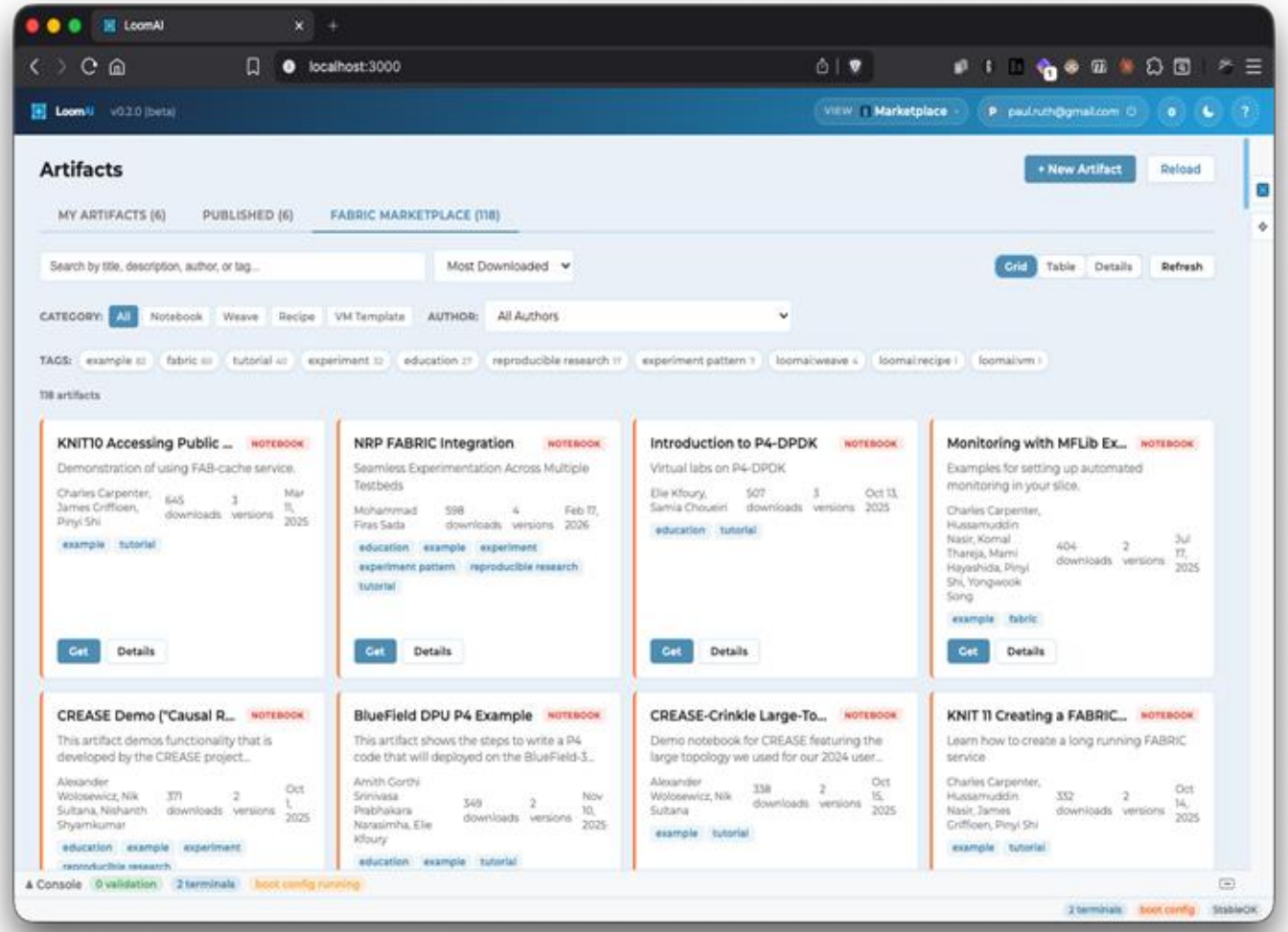
Embedded JupyterLab

- **Full FABRIC JupyterLab**
  - Includes FABRIC tools and config (fablib, etc.)
- **No Resource Limitations**
  - Runs on your laptop using your CPU, memory, storage



Artifacts & Marketplace

- **FABRIC Artifact Manager**
  - Integrated with FABRIC Artifact manager
  - Easily find, share, and publish artifacts
- **Existing Artifacts**
  - Run existing artifacts unmodified with the embedded JupyterLab
- **Weave Artifacts (New!)**
  - Weave: a FABRIC artifact with an executable entry point that can be used by LoomAI

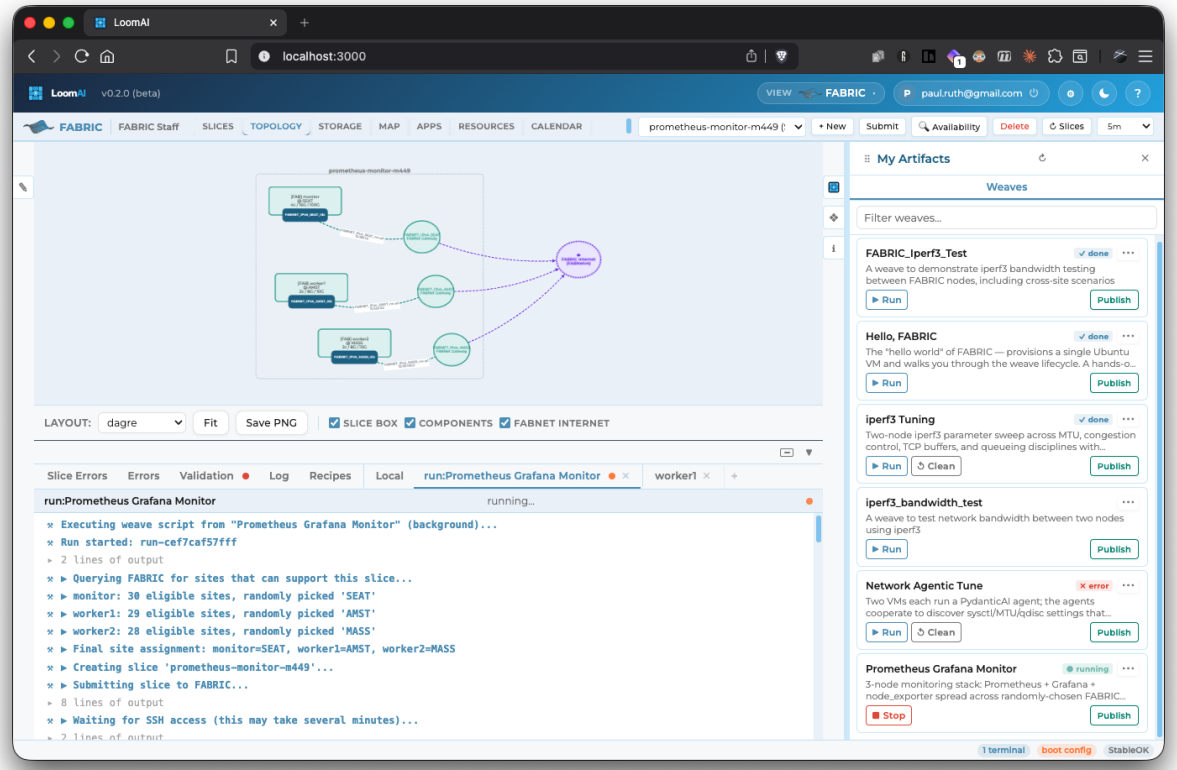


VISUAL · REUSE

Weaves — reusable experiment packages

- **Weaves**
  - A FABRIC artifact with an executable entry point that can be used by LoomAI
  - Parts: weave.json + entry script

- **Capabilities:**
  - Single slice despoilment
  - Multislice autonomous experiment management
  - In-slice AI agents
  - Jupyter notebook results analysis
  - And more!



Running a Weave: live log shows automatic site resolution

PART 2

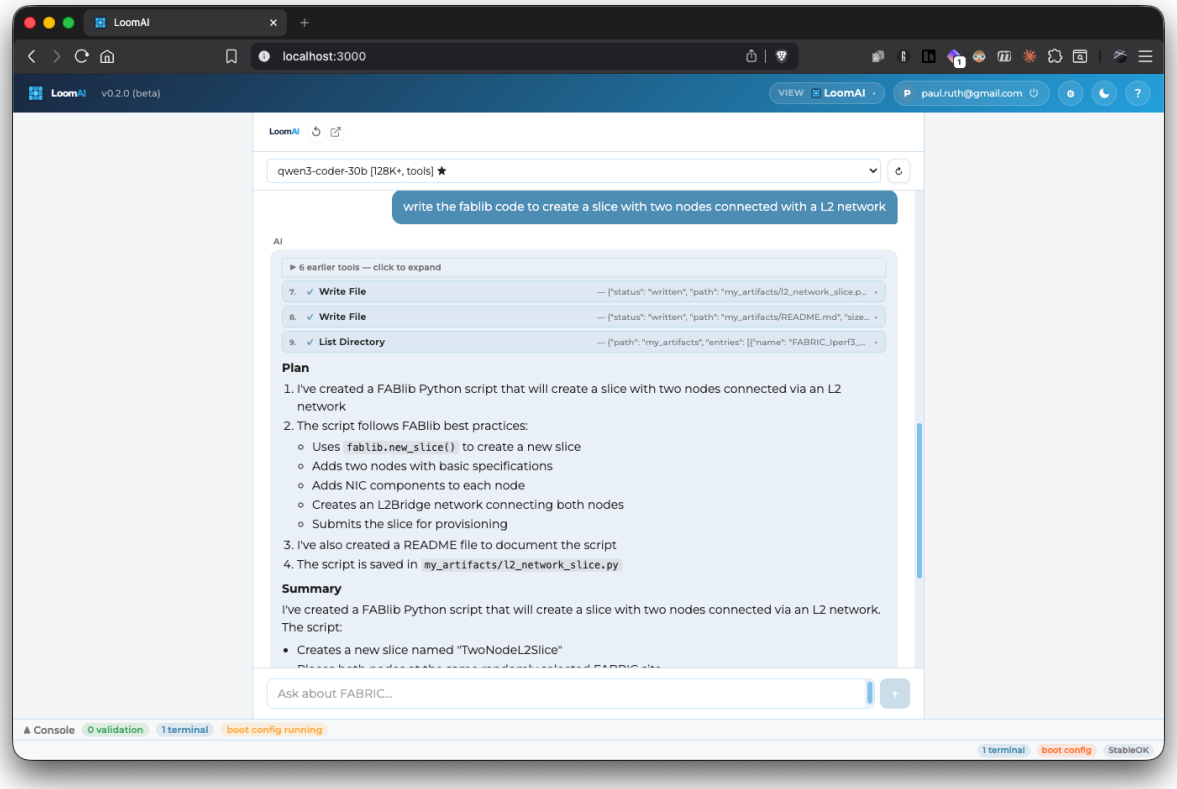
The AI Assistant

Design, explain, and troubleshoot — in plain language

LoomAI, the built-in assistant

- **Agentic interface**
  - 15+ FABRIC tools: create slices, query sites, manage resources
  - 25+ intent detection patterns for safe operation
  - Pre-loaded FABRIC documentation (RAG)

- **Capabilities**
  - Create/run slices
  - Create/run weaves
  - Find artifacts
  - Write fablib code
  - Search documentation
  - Configure active slices
  - And more!



Assistant generating FABlib code via real tool calls

Free LLM Access – No Paid API Keys Required

Use AI tools at zero cost with FABRIC-hosted models

FABRIC AI Gateway

ai.fabric-testbed.net

Qwen 3 Coder (free for FABRIC users)

OpenAI-compatible API

Auto-discovered on login

NRP LLM Proxy

ellm.nrp-nautilus.io

Additional model options

Fallback when FABRIC gateway busy

Also free for researchers

Custom Providers

Add your own: OpenAI, Anthropic, Ollama, vLLM, or any OpenAI-compatible API

Integrated Open Source AI Coding Tools

- **LoomAI** — Built-in chat with FABRIC tool calling
- **Aider** — Web IDE agent for AI pair programming
- **OpenCode** — Terminal agent with file operations
- **Claude Code** — Anthropic CLI, full codebase agent
- **Crush** — Lightweight TUI chat agent
- **Deep Agents** — LangChain multi-agent orchestrated workflows
- **Jupyter AI** — Notebook integration for data analysis

Suggested LoomAI Prompts – Create a slice

1. **Create the slice with LoomAI**

   "Create a slice that includes 3 nodes on 3 different sites, all connected to FABnetv4 with IP addresses automatically configured. The VMs should use Ubuntu 24."

2. **Submit the slice**

   Select the slice and click Submit, or ask LoomAI to submit it for you.

3. **Configure the slice with LoomAI**

   "Run apt update on all nodes in the FABnet three-node slice. Use apt to install common networking tools including iperf3 and iptables. Reboot the VMs if the Linux kernel was updated."

Suggested LoomAI Prompts – Prometheus/Grafana

- **Weave: Prometheus Grafana Monitor**
  - Get the weave from the Marketplace
  - Run the weave and wait until it is ready
  - Attach to the monitor node's Grafana app on port 3000
- **LoomAI Request**
  - *"Consider the existing Prometheus monitor slice. Do not create a new slice. Get the existing slice and the nodes called worker1 and worker2. Install iperf3 on the workers. Run an iperf3 bandwidth test for 60 seconds and report the results here."*

Network Tune Weave Creation

LoomAI Request:

"Create a new weave called 'network tune'. The weave should create a pair of VMs, install iperf3 on the VMs, and run iperf3 across a parameter sweep that includes MTU, congestion control algorithm, TCP buffer sizes, and queueing. It should run the sweep in both directions.

The results of each individual test should be transferred back to the weave folder. There should also be a Jupyter notebook that describes the experiments, analyzes the results with graphs, and suåggests the optimal host tuning for the pair of nodes and VM size.

The notebook should analyze partial data collected while the full run is still continuing. It should also provide a script that could be run on the hosts to tune them with the suggested parameters. The default settings should make the tests run for about 10 minutes."

Agentic Network Tune Weave Creation

LoomAI Request:

"Create a new weave called 'network agentic tune'. The weave should create a pair of VMs and run a PydanticAI agent on each one. The agents should work together to configure and tune the nodes to maximize bandwidth using iperf3 between them, then test to confirm their optimizations. In addition, a Jupyter notebook should be created and placed in the weave folder that explains the optimizations and how the agents reasoned to find the best configuration. The data confirming the results should be analyzed and included in the notebook."

Note: Currently, this one might need a more powerful AI tool like Claude.

PART 3

Cross-Facility Experimentation

One interface across FABRIC, SPHERE, and beyond

FEDERATION

Why federate — and what's live today

THE NEED

- Breakthrough research doesn't respect testbed boundaries — but today's infrastructure does.
- Security needs programmable networks; AI needs GPUs — most projects need several facilities at once.
- Each facility is a separate account, SDK, and auth flow: every boundary is another integration project that stalls the science.

LIVE TODAY — CHAMELEON + SPHERE

- LoomAI federates two facilities behind one interface — learn one workflow, reach both.
- Chameleon Cloud: 60+ endpoints, a Draft → Deploy workflow, an OpenStack browser.
- SPHERE (MergeTB): built into the same interface — no context switch.
- Each facility keeps its own auth and semantics; LoomAI harmonizes the experience on top.

Spotlight: SPHERE Edge IDS/IPS Lab

A reproducible intrusion-detection lab on the SPHERE testbed — a published LoomAI Weave, deployed and analyzed from the browser.

Compromised device
Runs the multi-stage attack

Benign device
Baseline 'good' traffic

Gateway + victim svc
Target services behind the edge

Suricata IDS sensor
Monitors & alerts on traffic

Attack chain: Recon (SYN/FIN/XMAS scans) → Web exploit (path traversal, SQLi) → SSH brute force → DNS-tunnel exfiltration

- A bundled Jupyter notebook automates deployment and analysis.
- Auto-generates topology graphs, alert categorization, severity, and attack timelines.
- Culminates in a closed-loop IPS that autonomously blocks the attacker — while benign traffic keeps flowing.
- Published on the Marketplace (tags: sphere, loomai:weave) — clone, deploy, reproduce.

Try it live: prompts for the SPHERE lab

Prompts to give the LoomAI assistant while demoing the SPHERE Edge IDS/IPS Lab:

- List all my SPHERE experiments and their realizations.
- Create a SPHERE experiment netsec1 in project fabricsphere. Push a mergexp model with a gateway and two edge nodes on a 10.10.1.0/24 LAN (facility moddeter), tag it v1, realize it as r0, and materialize it. Then attach a shared XDC for SSH access. Finally, package the experiment (realization r0) into a weave called netsec1-detector: add a detector.py setup script on the gateway that sniffs the LAN and flags anomalies (high SYN/ICMP/ARP rates, off-subnet source IPs) to alerts.jsonl, and author a notebook that launches the detector and then loads those alerts and plots them over time and by rule. Send the finished weave to the XDC.

Thank you

Questions & discussion

Komal Thareja · kthare10@renci.org

RENCI, UNC Chapel Hill · FABRIC Testbed

LoomAI is open source — try it, host it, contribute.

SPHERE Edge IDS/IPS Lab: artifacts.fabric-testbed.net