THE LINUX FOUNDATION MEMBER SUMMIT

PyTorch

Collaborative Infrastructure at Scale: PyTorch's Multi-Cloud CI Model

Andrea Frittoli
Developer Advocate
andrea.frittoli@uk.ibm.com

Andrea Frittoli

afrittoli | andreafrittoli | @blackchip76

- Open Source Advocate @ IBM
- Lives in Wales, enjoys the wind
- Multi-Cloud CI Working Group Lead
- Member of PyTorch Infra Working Group
- Tekton, CDEvents maintainer

Photo by Calum Lewis, CC0

Contents

The Infrastructure Challenge

PyTorch Foundation

Technical Architecture

Governance Model

Results & Next Steps

Q&A

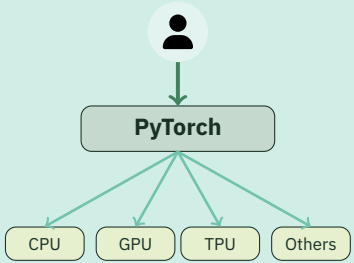
The Infrastructure Challenge

Photo by Denys Nevozhai, CC0

What is PyTorch?

A Leading Open Source ML Framework:

- Researchers, data scientists, ML engineers
- Research and production ML workloads
- (Distributed) Training, LLM, RL, Inference
- Python API, C++ Core (libtorch)
- Eager and Graph Mode (Inductor)
- Rapid growth and adoption across industry



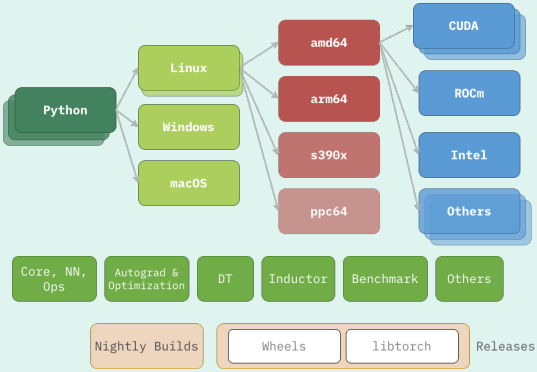
CI/CD Infrastructure Today

- ~1M$ monthly infrastructure costs
- ~1M hours of compute per month
- ~100 distinct PRs/day

- ~1.3k Compute-intensive Workflows/day
- Growing year over year

`pytorch/pytorch` repo only, numbers are estimated

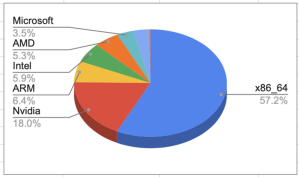
An Expanding Test Matrix



Combined Hardware Runtime by Hardware Vendor September 2025

Combined Runtime By Hardware Vendor

Vendor	Total
x86_64	834,085 hours
Nvidia	262,878 hours
ARM	93,000 hours
Intel	86,090 hours
AMD	77,517 hours
Microsoft	51,515 hours
Apple	46,812 hours
IBM	6,245 hours
N/A	0 hours
	1,458,142 hours



* Data from PT Foundation AWS Account + PyTorch HUD. This does not include self-hosted runner costs by member provided runners.

The Challenges

Scale Challenges:

- **Engineering:** Managing a Diverse Fleet, Access to Platforms
- **Financial:** Reconcile Community Wishes with Budget Constraints

Security Challenges:

- Infrastructure from public clouds and private pool
- Central Administration
- Trusted Builds

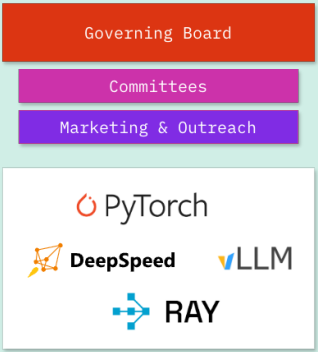
Experience Challenges:

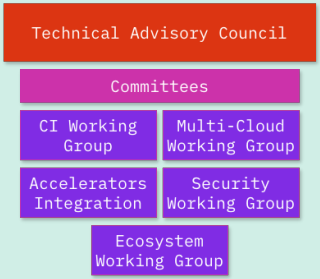
- Maintain high-quality end-user experience
- Preserve contributor workflow
- Provide clear path for vendor engagement

PyTorch Foundation & Working Groups

The PyTorch Foundation

- Part of the Linux Foundation
- Neutral home for PyTorch project
- Hosts multiple projects
- Thought Leadership (GB)
- Technical Leadership (TAC)
- Marketing & Outreach





Working Groups

Multi-Cloud CI

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Accelerator Integration

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

CI Infra

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Security

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Vendor Contribution Models

Three Ways to Contribute:

- **Compute Resources:** Multiple integration levels
- **Engineering Effort:** Integration, maintenance, support
- **Cloud Credits:** Financial contribution for shared infrastructure

Flexible & Proportional:

- Vendors choose contribution model that fits them
- Contribution proportional to their investment/stake

Technical Architecture

Photo by Nadi Whatisdelirium, CC0

Cloud-Agnostic CI Design

Key Principles:

- No vendor lock-in
- Standardized runner interfaces
- Portable CI definitions
- Multi-cloud orchestration

Security & Isolation

Security Considerations:

- Vendor-managed runners in isolated environments
- No access to project secrets
- Network isolation and egress controls
- Audit logging for all runner activity
- Regular security reviews

Monitoring & Observability

Visibility Across Vendors:

- Centralized metrics collection
- Performance monitoring per vendor
- Cost tracking and attribution
- SLA monitoring and alerting
- Public dashboards for transparency

Vendor-Managed Runners

Operational Model:

- Vendors provision and maintain runners
- Standard configuration templates
- Automated scaling based on demand
- Health checks and auto-remediation
- Vendor-specific optimizations allowed

Governance Model

Photo by Thom Milkovic, CC0

Governance Principles

Balancing Act:

- Project autonomy preserved
- Vendor participation encouraged
- Community control maintained
- Transparent decision-making
- Fair representation

Working Group Structure

Organization:

- Regular meetings (bi-weekly/monthly)
- Vendor representatives + maintainers
- Technical subcommittees
- Clear escalation paths
- Public meeting notes

Vendor Onboarding

Standardized Process:

- Technical requirements review
- Security assessment
- Pilot phase with limited workloads
- Performance validation
- Full integration after approval

Handling Conflicts

When Interests Diverge:

- Clear decision-making authority
- Project maintainers have final say
- Vendor concerns heard but not binding
- Documented rationale for decisions
- Exit strategy for vendors

Results & Lessons Learned

Platform Coverage Expansion

Achievements:

- X% increase in platform coverage
- New accelerator support (specific examples)
- Reduced time-to-market for new platforms
- Improved test reliability

Financial Impact

Sustainability Achieved:

- Distributed infrastructure costs
- Reduced burden on primary sponsor
- Predictable scaling model
- Long-term financial sustainability

Vendor Onboarding Experience

Feedback from Vendors:

- Clear expectations and requirements
- Reasonable onboarding timeline
- Good technical support from maintainers
- Fair governance model
- Challenges: [specific examples]

Key Lessons Learned

What Worked:

- Clear governance from day one
- Security-first approach
- Flexible contribution models
- Transparent communication

What Was Hard:

- Balancing vendor needs with project needs
- Standardization across diverse platforms

Applicability to Other LF Projects

Shared Challenges:

- Accepting vendor infrastructure
- Maintaining project neutrality
- Ensuring transparency
- Achieving financial sustainability

This Model Can Help:

- Proven governance framework
- Technical architecture patterns
- Onboarding playbook

Future Directions

What's Next:

- Expand to more vendors
- Improve automation and tooling
- Share model with other projects
- Refine governance based on experience
- Build community of practice

Key Takeaways

- Infrastructure costs can be distributed sustainably
- Vendor contributions work with right governance
- Project autonomy and vendor participation can coexist
- Security and transparency are non-negotiable
- Model is applicable to other LF projects

Questions?

Brecon Beacons, Wales, Photo by Carl Jorgensen, CC0

References & Contact

Resources:

- PyTorch Multi-Cloud CI Working Group: [URL]
- Documentation: [URL]
- GitHub: [URL]

Contact:

- Andrea Frittoli
- andrea.frittoli@uk.ibm.com
- @blackchip76 | afrittoli