Test Environment Provisioning Optimization - Project Proposal

Executive Summary

This proposal outlines an optimization to test environment provisioning infrastructure that will reduce wait times by approximately **10 minutes per request**, delivering **$32,100-$48,200** in annual productivity value and **$141,393-$215,131** in net present value over 5 years (mid-range: **$171,621**). If we achieve our stretch goal of **15 minutes improvement**, the net value increases to **$215,131-$325,509** over 5 years.

Financial projections assume 77 provisions/week after hiring initiative adds 11 engineers (current: 65 provisions/week with 60 engineers).

Strategic Context: AI Development Velocity

The company is pursuing an aggressive AI adoption strategy. AI-assisted coding practices are enabling developers to produce code significantly faster than traditional methods. However, this increased velocity puts stress on existing CI and testing infrastructure, creating bottlenecks that can throttle the benefits of AI adoption.

By optimizing test environment provisioning, we ensure that infrastructure keeps pace with AI-enhanced development speed. This work directly supports AI adoption efforts by removing bottlenecks that would otherwise limit developer velocity and hinder the company's AI transformation goals.

The Goal

Reduce test environment provisioning time by approximately **10 minutes** per request, with a stretch goal of **15 minutes** through additional optimizations. This enables developers to access test environments faster and ensures infrastructure keeps pace with AI-enhanced development velocity.

Financial Impact

Implementation Cost: **$5,625** (1 engineer, 1.5 weeks)

Annual Savings (10 min improvement): **$32,100 - $48,200** (mid-range: $38,700)

5-Year Net Present Value (10 min improvement): **$141,393 - $215,131** (mid-range: $171,621, 3% discount rate)

Payback Period: 1.7 months

Stretch Goal (15 min improvement): **$48,200 - $72,300** annually (**$215,131 - $325,509** net over 5 years)

Calculations assume 77 provisions/week (projected post-hiring rate) vs. current 65/week with 60 engineers.

See Detailed Financial Analysis section below for complete breakdown of scenarios and assumptions.

Success Metrics

We will measure success through the following key performance indicators:

Primary Metric: Provisioning Lead Time

- **Current**: ~18 minutes (mural-api bottleneck, confirmed by CI analysis)
- **Realistic Target**: ~8 minutes (10 min improvement)
- **Stretch Target**: ~5 minutes (15 min improvement)
- **Measurement**: P50, P90, P99 latency from request to environment ready

Supporting Metrics

- **Environment Usage**: Number of provisions per week/month
- **Image Build Time**: Duration for Docker images to be built and pushed (provisioning proceeds when images are available)
- **Manual Trigger Delay**: Time gap between images ready and provision request (eliminated by auto-provisioning)
- **Provisioning Success Rate**: Reliability of environment creation
- **Manual Re-provisioning Rate**: Frequency of environment issues requiring recreation

How Will We Achieve the 10-Minute Improvement?

This optimization focuses on three coordinated work streams:

1. Comprehensive Metrics Implementation

Instrument the entire provisioning flow with timing and success metrics to:

- Establish baseline performance
- Validate actual improvements from optimizations
- Track reliability and identify failure patterns
- Inform future optimization priorities

2. Make Images Available Faster (~5 minutes)

CI analysis shows opportunities to reduce Docker image build times through workflow improvements. Initial investigation suggests production images for `mural-api` can be built concurrently rather than sequentially, reducing the time until images are available for provisioning. `mural-api` image availability is the current bottleneck.

Expected Impact: ~5 minutes faster image availability

3. Automatic Provisioning (~3-5 minutes)

Replace manual environment requests (Slack commands, MDK) with automatic provisioning when pull requests are created or updated. This eliminates the human delay between when images are ready and when developers remember to request an environment.

Expected Impact: ~3-5 minutes by removing manual trigger delay

Stretch Goal: Additional Build Optimizations (~2-5 minutes)

Further analysis may reveal additional opportunities in frontend build processes. This would be evaluated after completing core optimizations and reviewing metrics data.

Timeline

Total Duration: 3-4 weeks

- **Week 1**: Implement metrics instrumentation + mural-api CI optimization
- **Week 2**: Collect baseline data + implement automatic provisioning + identify more optimization for stretch goal
- **Week 3**: Monitor optimizations, collect post-implementation data
- **Week 4**: Analysis and reporting

Actual Developer Time: 1.5 weeks (implementation + analysis)

Most time is spent on metrics collection and validation. Weeks 3-4 involve waiting for data to accumulate while developers work on other priorities.

Detailed Financial Analysis

Team Composition

71 engineers total (reflects hiring initiative with 11 new engineers):

- **USA/Canada/Europe**: 43 engineers (+7 new hires)
- **Argentina**: 28 engineers (+4 new hires)

Salary Assumptions

Scenario	USA/Canada/Europe (43 engineers)	Argentina (28 engineers)	Blended Hourly Rate
Low	$120k/year ($62.50/hr)	$50k/year ($26.04/hr)	**$48.12/hour**
Mid	$145k/year ($75.52/hr)	$60k/year ($31.25/hr)	**$58.06/hour**
High	$180k/year ($93.75/hr)	$75k/year ($39.06/hr)	**$72.18/hour**

(Assumes 48 working weeks/year × 40 hours/week = 1,920 hours after 4 weeks vacation)

Time Savings Assumptions

Current Usage: 131 environments provisioned in 2 weeks (65/week current rate) with 60 engineers.

Projected Usage: With the hiring initiative adding 11 engineers (71 total), we project **77 provisions/week** (1.08 provisions per engineer per week). All financial calculations use this projected post-hiring rate.

Time Savings Based on CI Analysis: Conservative estimates based on actual CI pipeline measurements showing 5-minute improvement from mural-api parallelization plus 3-5 minutes from automatic provisioning.

Scenario	Time Saved/Provision	Provisions/Week	Hours Saved/Year
Conservative	8 minutes	77	534 hours
Realistic	10 minutes	77	667 hours
Optimistic (Stretch)	15 minutes	77	1,001 hours

Annual Savings Matrix

Time Savings Scenario	Provisions/Week	Hours/Year	Low Salary ($48.12/hr)	Mid Salary ($58.06/hr)	High Salary ($72.18/hr)
Conservative (8 min/provision)	77	534 hours	$25,700	$31,000	$38,500
Realistic (10 min/provision)	77	667 hours	$32,100	$38,700	$48,200
Optimistic (Stretch) (15 min/provision)	77	1,001 hours	$48,200	$58,100	$72,300

5-Year Present Value Analysis

Assuming a **3% annual discount rate** (typical for inflation-adjusted calculations):

Time Savings Scenario	Low Salary (Annual)	Mid Salary (Annual)	High Salary (Annual)	Low Salary (5Y PV)	Mid Salary (5Y PV)	High Salary (5Y PV)
Conservative (8 min)	$25,700	$31,000	$38,500	$117,706	$141,980	$176,330
Realistic (10 min)	$32,100	$38,700	$48,200	$147,018	$177,246	$220,756
Optimistic/Stretch (15 min)	$48,200	$58,100	$72,300	$220,756	$266,098	$331,134

Present Value Factor: 4.58 (using 3% discount rate over 5 years)

Investment Case Summary

Implementation Cost:

- 1 engineer × 1.5 weeks (60 hours) at high salary (Canada-based): **$5,625**

Realistic Scenario (10 min time savings, Mid salary):

- **Annual Savings**: $38,700
- **5-Year Present Value**: $177,246
- **Less Implementation Cost**: -$5,625
- **Net 5-Year Value**: **$171,621**

- **Hours Reclaimed**: 667 hours/year
- **Payback Period**: 1.7 months

Range of Outcomes (10 min improvement):

- **Low Salary**: $141,393 net (5Y PV)
- **High Salary**: $215,131 net (5Y PV)

Stretch Goal Range (15 min improvement):

- **Low Salary**: $215,131 net (5Y PV)
- **High Salary**: $325,509 net (5Y PV)

Key Insight: Even after accounting for $5k implementation cost, conservative estimates ($112k-$170k net over 5 years) provide strong ROI with minimal risk. The realistic scenario delivers $172k net present value with payback in 1.7 months. Stretch goal scenarios ($215k-$326k net) are achievable with additional optimization effort but are not included in core project scope.

Validation: Baseline metrics collection will confirm actual time savings and inform future optimization priorities.

Next Steps

1. **Approval**: Review and approve this proposal
2. **Resource Allocation**: Assign engineering resources for implementation (1.5 weeks effort)
3. **Implementation**: Execute three-phase approach (metrics, image optimization, auto-provisioning)
4. **Validation**: Collect metrics to confirm actual time savings and ROI
5. **Future Decision**: Evaluate stretch goal based on results

Related Documentation

- [Detailed Metrics Implementation Plan](#)
- [CI Optimization Implementation Plan](#)