

# Deployment Notes - Gemini Context Extension

---

## ✅ Successfully Completed

---

Your Gemini Context Extension has been successfully updated and pushed to GitHub!

**Repository:** <https://github.com/Beaulewis1977/gemini-context-extension>

## What Was Done

### 1. Full Model Support ✅

Added comprehensive support for all current Gemini models:

#### Gemini 2.5 Series (Latest 2025 Models):

- **gemini-2.5-pro**: Most capable for complex reasoning and coding
  - Input: \$1.25/M ( $\leq 200k$  tokens), \$2.50/M ( $> 200k$  tokens)
  - Output: \$10/M ( $\leq 200k$  tokens), \$15/M ( $> 200k$  tokens)
  - Context: 1M tokens
- **gemini-2.5-flash**: Balanced speed and performance
  - Input: \$0.30/M
  - Output: \$2.50/M
  - Context: 1M tokens
- **gemini-2.5-flash-lite**: Most cost-effective
  - Input: \$0.10/M
  - Output: \$0.40/M
  - Context: 1M tokens

#### Other Models:

- gemini-2.0-flash-exp
- gemini-1.5-pro (2M context window)
- gemini-1.5-flash

### 2. Model Comparison Features ✅

Implemented comprehensive comparison tool ( `compare_gemini_models` ):

- Displays all models with descriptions and capabilities
- Shows pricing for input/output tokens
- Calculates cost estimates for current usage
- Sorts models by cost efficiency (cheapest first)
- Provides complete model information

### 3. Enhanced Cost Estimator ✅

Updated `estimate_api_cost` tool with:

- Latest 2025 pricing data for all models
- Tiered pricing support (models with different rates for small vs large prompts)
- Comprehensive savings analysis showing cost differences between models

- Smart recommendations for cost optimization
- Per-request and total cost breakdowns

#### 4. Updated Context Tracker

Enhanced `track_context_usage` tool with:

- Support for all models with accurate context window sizes
- Model parameter to analyze any Gemini model
- Model-specific insights (1M vs 2M token windows)
- Enhanced recommendations based on model capabilities

#### 5. Documentation Updates

- Updated README.md with complete model information
- Updated GEMINI.md with usage examples for all tools
- Added pricing details for all supported models
- Added best practices for model selection

#### 6. Git Repository Setup

- Initialized git repository
- Set up remote: <https://github.com/Beaulewis1977/gemini-context-extension.git>
- Committed all changes with comprehensive commit messages
- Successfully pushed to GitHub main branch



## Manual Action Required: GitHub Actions Workflow

Due to GitHub App permissions, the CI/CD workflow file ( `.github/workflows/ci.yml` ) could not be pushed automatically. You'll need to add it manually.

### Steps to Add the CI/CD Workflow:

#### 1. Navigate to your repository on GitHub:

<https://github.com/Beaulewis1977/gemini-context-extension>

#### 2. Create the workflow file manually:

- Go to the "Actions" tab
- Click "New workflow" or "set up a workflow yourself"
- Or simply create the file `.github/workflows/ci.yml` through the web interface

#### 3. Copy the workflow content:

The workflow file is located locally at: `.github/workflows/ci.yml`

Content:

```
```yaml
```

```
name: CI
```

```
on:
```

```
push:
```

```
branches: [main, develop]
```

```
pull_request:
```

```
branches: [main, develop]
```

```
jobs:
  lint-and-build:
    runs-on: ubuntu-latest
```

```
    strategy:
      matrix:
        node-version: [18.x, 20.x]

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Setup Node.js ${ matrix.node-version }
        uses: actions/setup-node@v4
        with:
          node-version: ${ matrix.node-version }
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Run ESLint
        run: npm run lint

      - name: Check Prettier formatting
        run: npm run format:check

      - name: Build TypeScript
        run: npm run build

      - name: Check build output
        run: |
          if [ ! -f dist/server.js ]; then
            echo "Build failed: dist/server.js not found"
            exit 1
          fi
          echo "Build successful!"
```

...

### 1. Commit the workflow file directly on GitHub

This workflow will automatically:

- Run on every push to main/develop branches
- Test against Node.js 18.x and 20.x
- Run ESLint checks
- Verify Prettier formatting
- Build the TypeScript code
- Verify the build output

---

## Extension Ready to Use

Your extension is now fully functional and ready for users to install!

## Installation Command:

```
gemini extensions install https://github.com/Beaulewis1977/gemini-context-extension
```

## Available Tools:

1. **track\_context\_usage** - Analyze context window usage for any model
2. **estimate\_api\_cost** - Calculate costs with comprehensive model comparisons
3. **compare\_gemini\_models** - Compare all available Gemini models

## Next Steps:

1. Add the GitHub Actions workflow manually (see above)
2. Test the extension by installing it in Gemini CLI
3. Try out the model comparison feature
4. Share the repository with users



## Summary of Changes

### Files Modified:

- `src/tools/cost-estimator.ts` - Added all models with latest pricing
- `src/tools/context-tracker.ts` - Added model support with accurate context windows
- `src/server.ts` - Added new comparison tool and updated tool descriptions
- `GEMINI.md` - Updated with comprehensive model information
- `README.md` - Added detailed documentation for all models and features

### New Features:

- ☒ Gemini 2.5 Pro support
- ☒ Gemini 2.5 Flash support
- ☒ Gemini 2.5 Flash-Lite support
- ☒ Complete model comparison tool
- ☒ Tiered pricing calculations
- ☒ Savings analysis
- ☒ Smart recommendations

### Git Commits:

- Initial commit (already existed)
- "Add comprehensive Gemini model support and comparison features" (new)
- Successfully pushed to GitHub main branch



## Repository Links

- **Repository:** <https://github.com/Beaulewis1977/gemini-context-extension>
- **Installation:** `gemini extensions install https://github.com/Beaulewis1977/gemini-context-extension`

**Deployment Date:** October 21, 2025

**Status:**  Complete (pending manual workflow addition)