m... | 20 Branches | 0 Tags | Go to file | Code | ...

atupem Add prisma dev dependency and update client to latest ✕
3d37894 · 2 months ago 499 Commits

.github/workflows	Update Docker definitons	5 months ago
docker	Implement a simplified version of th...	3 months ago
docs	Update Helm docs	3 months ago
helm	Update Helm docs	3 months ago
packages	Add prisma dev dependency and up...	2 months ago
static	Update README, discord links, and s...	4 months ago
.gitignore	Revamp build script, add run script	8 months ago
.prettierignore	Documentation updates	5 months ago
LICENSE	Update LICENSE	4 months ago
README.md	doc: add links to translated README ...	2 months ago

README | Apache-2.0 license

bytebot

Bytebot: Open-Source AI Desktop Agent

GITHUB TRENDING
1
#1 Repository Of The Day

An AI that has its own computer to complete tasks for you

Deploy on Railway

docker ready | license Apache 2.0 | discord 74 online

🌐 Website • 📚 Documentation • 💬 Discord • 𝕏 Twitter

Deutsch | Español | français | 日本語 | 한국어 | Português | Русский | 中文

bytebot_github_login.mp4

bytebot_uc_2.mp4

About

Bytebot is a self-hosted AI desktop agent that automates computer tasks through natural language commands, operating within a containerized Linux desktop environment.

www.bytebot.ai/

#agent #docker #automation #ai #mcp #desktop #gemini #openai #desktop-automation #agents #cua #ai-agents #ai-tools #llm #anthropic #agentic-ai #computer-use #computer-use-agent #bytebot

Readme
Apache-2.0 license
Activity
Custom properties
9.6k stars
65 watching
1.2k forks
Report repository

Releases

No releases published

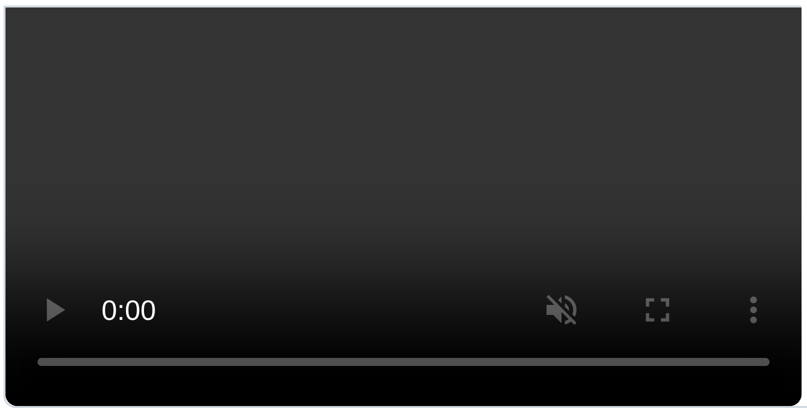
Packages 3

bytebot-desktop
bytebot-agent
bytebot-ui

Contributors 11

Languages

● TypeScript 92.1%	● Dockerfile 2.4%
● CSS 2.0%	● Smarty 1.5%
● PLpgSQL 0.7%	● Scheme 0.7%
● JavaScript 0.6%	



What is a Desktop Agent?

A desktop agent is an AI that has its own computer. Unlike browser-only agents or traditional RPA tools, Bytebot comes with a full virtual desktop where it can:

- Use any application (browsers, email clients, office tools, IDEs)
- Download and organize files with its own file system
- Log into websites and applications using password managers
- Read and process documents, PDFs, and spreadsheets
- Complete complex multi-step workflows across different programs

Think of it as a virtual employee with their own computer who can see the screen, move the mouse, type on the keyboard, and complete tasks just like a human would.

Why Give AI Its Own Computer?

When AI has access to a complete desktop environment, it unlocks capabilities that aren't possible with browser-only agents or API integrations:

Complete Task Autonomy

Give Bytebot a task like "Download all invoices from our vendor portals and organize them into a folder" and it will:

- Open the browser
- Navigate to each portal
- Handle authentication (including 2FA via password managers)
- Download the files to its local file system
- Organize them into a folder

Process Documents

Upload files directly to Bytebot's desktop and it can:

- Read entire PDFs into its context
- Extract data from complex documents
- Cross-reference information across multiple files
- Create new documents based on analysis
- Handle formats that APIs can't access

Use Real Applications

Bytebot isn't limited to web interfaces. It can:

- Use desktop applications like text editors, VS Code, or email clients
- Run scripts and command-line tools
- Install new software as needed
- Configure applications for specific workflows

Quick Start

Deploy in 2 Minutes

Option 1: Railway (Easiest) Deploy on Railway

Just click and add your AI provider API key.

Option 2: Docker Compose

```
git clone https://github.com/bytebot-ai/bytebot.git
cd bytebot

# Add your AI provider key (choose one)
echo "ANTHROPIC_API_KEY=sk-ant-..." > docker/.env
# Or: echo "OPENAI_API_KEY=sk-..." > docker/.env
# Or: echo "GEMINI_API_KEY=..." > docker/.env

docker-compose -f docker/docker-compose.yml up -d
```

```
# Open http://localhost:9992
```

Full deployment guide →

How It Works

Bytebot consists of four integrated components:

1. **Virtual Desktop**: A complete Ubuntu Linux environment with pre-installed applications
2. **AI Agent**: Understands your tasks and controls the desktop to complete them
3. **Task Interface**: Web UI where you create tasks and watch Bytebot work
4. **APIs**: REST endpoints for programmatic task creation and desktop control

Key Features

- **Natural Language Tasks:** Just describe what you need done
- **File Uploads:** Drop files onto tasks for Bytebot to process
- **Live Desktop View:** Watch Bytebot work in real-time
- **Takeover Mode:** Take control when you need to help or configure something
- **Password Manager Support:** Install 1Password, Bitwarden, etc. for automatic authentication
- **Persistent Environment:** Install programs and they stay available for future tasks

Example Tasks

Basic Examples

```
"Go to Wikipedia and create a summary of quantum computing"
"Research flights from NYC to London and create a comparison document"
"Take screenshots of the top 5 news websites"
```

Document Processing

```
"Read the uploaded contracts.pdf and extract all payment terms and deadlines"
"Process these 5 invoice PDFs and create a summary report"
"Download and analyze the latest financial report and answer: What were the key risks
mentioned?"
```

Multi-Application Workflows

```
"Download last month's bank statements from our three banks and consolidate them"
"Check all our vendor portals for new invoices and create a summary report"
"Log into our CRM, export the customer list, and update records in the ERP system"
```

Programmatic Control

Create Tasks via API

```
import requests

# Simple task
response = requests.post('http://localhost:9991/tasks', json={
    'description': 'Download the latest sales report and create a summary'
})

# Task with file upload
files = {'files': open('contracts.pdf', 'rb')}
response = requests.post('http://localhost:9991/tasks',
    data={'description': 'Review these contracts for important dates'},
    files=files
)
```

Direct Desktop Control

```
# Take a screenshot
curl -X POST http://localhost:9990/computer-use \
  -H "Content-Type: application/json" \
  -d '{"action": "screenshot"}'

# Click at specific coordinates
curl -X POST http://localhost:9990/computer-use \
  -H "Content-Type: application/json" \
  -d '{"action": "click_mouse", "coordinate": [500, 300]}'
```

Full API documentation →

Setting Up Your Desktop Agent

1. Deploy Bytebot

Use one of the deployment methods above to get Bytebot running.

2. Configure the Desktop

Use the Desktop tab in the UI to:

- Install additional programs you need
- Set up password managers for authentication
- Configure applications with your preferences
- Log into websites you want Bytebot to access

3. Start Giving Tasks

Create tasks in natural language and watch Bytebot complete them using the configured desktop.

Use Cases

Business Process Automation

- Invoice processing and data extraction
- Multi-system data synchronization
- Report generation from multiple sources
- Compliance checking across platforms

Development & Testing

- Automated UI testing
- Cross-browser compatibility checks
- Documentation generation with screenshots
- Code deployment verification

Research & Analysis

- Competitive analysis across websites
- Data gathering from multiple sources
- Document analysis and summarization
- Market research compilation

Architecture

Bytebot is built with:

- **Desktop**: Ubuntu 22.04 with XFCE, Firefox, VS Code, and other tools
- **Agent**: NestJS service that coordinates AI and desktop actions
- **UI**: Next.js application for task management
- **AI Support**: Works with Anthropic Claude, OpenAI GPT, Google Gemini
- **Deployment**: Docker containers for easy self-hosting

Why Self-Host?

- **Data Privacy**: Everything runs on your infrastructure
- **Full Control**: Customize the desktop environment as needed
- **No Limits**: Use your own AI API keys without platform restrictions
- **Flexibility**: Install any software, access any systems

Advanced Features

Multiple AI Providers

Use any AI provider through our LiteLLM integration:

- Azure OpenAI
- AWS Bedrock
- Local models via Ollama
- 100+ other providers

Enterprise Deployment

Deploy on Kubernetes with Helm:

```
# Clone the repository
git clone https://github.com/bytebot-ai/bytebot.git
```

```
cd bytebot

# Install with Helm
helm install bytebot ./helm \
  --set agent.env.ANTHROPIC_API_KEY=sk-ant-...
```

Enterprise deployment guide →

Community & Support

- **Discord**: Join our community for help and discussions
- **Documentation**: Comprehensive guides at docs.bytebot.ai
- **GitHub Issues**: Report bugs and request features

Contributing

We welcome contributions! Whether it's:

- 🐛 Bug fixes
- ✨ New features
- 📚 Documentation improvements
- 🌐 Translations

Please:

1. Check existing issues first
2. Open an issue to discuss major changes
3. Submit PRs with clear descriptions
4. Join our Discord to discuss ideas

License

Bytebot is open source under the Apache 2.0 license.