

AI-03951 AI Agents Deployment

Owner	Ⓜ Rigel Arcayan
Tags	Project
Created time	June 19, 2026 8:34 AM

What are AI Agents? Explain and give examples. List top 10 AI Agents based on Docker which I can install in my Google Cloud Platform VM instance. Explain the functions and what it can do. What are AI agents for the internet. List top 10 internet search AI agents I can run on Docker GCP.

An **AI Agent** is an autonomous software entity powered by a Large Language Model (LLM) that goes beyond simple text generation. While a standard AI chatbot relies entirely on human prompting to answer a question, an AI agent is designed to **perceive its environment, reason through a multi-step objective, use external tools, and take actions autonomously** to achieve a defined goal.

Core Anatomy of an AI Agent

An AI agent generally consists of four primary pillars:

1. **Goal/Role:** A defined persona or core directive (e.g., "You are a senior software engineer optimizing database queries").
2. **Memory:** Short-term memory (in-context conversation history) and long-term memory (vector databases providing context over time).
3. **Planning & Reasoning Loop:** The ability to break a massive task down into smaller, logical sub-tasks using frameworks like ReAct (Reasoning and Acting) or Chain-of-Thought.
4. **Tools:** Connectors allowing the agent to interface with the real world (e.g., executing Python code, executing SQL queries, reading files, browsing the web, calling APIs).

Practical Examples

- **A Coding Agent:** Instead of just writing a block of code, it accesses a local directory via an IDE extension, analyzes existing dependencies, writes a multi-stage Dockerfile, executes a test suite, reads the error stack trace, corrects its own code, and pushes a commit to a Git repository.
- **A Personal Assistant Agent:** Instead of just writing an email draft, it accesses your calendar, interfaces with a flight booking API, cross-references prices, coordinates time zones, books the flight, and sends confirmation details to your inbox.

Top 10 Docker-Based AI Agent Platforms for GCP

If you deploy a Google Cloud Platform (GCP) Compute Engine Virtual Machine (VM) running Ubuntu Server and Docker/Docker Compose, these are the top open-source frameworks, execution engines, and orchestration tools to host and manage custom AI agents.

1. Docker Agent (cagent)

- **What it is:** Docker's official, native framework designed to build and run teams of specialized AI agents via container architecture.
- **Core Function:** Reads structured declarative YAML configuration files to provision multi-agent pipelines. It uses the Model Context Protocol (MCP) to seamlessly connect agents to external resources via OCI-compliant environments.

2. Dify

- **What it is:** A comprehensive, full-stack open-source LLM app development platform with a visual canvas.
- **Core Function:** Operates as a visual workflow orchestrator for agents. It easily deploys on a GCP VM using Docker Compose, providing a drag-and-drop workflow, agent memory state management, and integrated Retrieval-Augmented Generation (RAG) toolkits.

3. Flowise

- **What it is:** A popular low-code/no-code UI orchestrator for building customized LLM applications and autonomous agents.
- **Core Function:** Allows you to rapidly assemble agents, memory buffers, and custom tool interfaces via a node-based interface. It runs efficiently in a lightweight single-container Docker setup on a micro-tier VM instance.

4. n8n

- **What it is:** A powerful, node-based workflow automation platform with dedicated advanced AI agent nodes.
- **Core Function:** Weaves deterministic code blocks with adaptive AI reasoning. It can be paired with local LLMs or external APIs, empowering an agent to intercept webhooks, manage complex internal workflows, and act as a backend operations agent.

5. CrewAI (Enterprise/Self-Hosted Engine)

- **What it is:** An agentic framework centered entirely on multi-agent collaboration and role-playing execution loops.
- **Core Function:** Orchestrates distinct agents (e.g., a "Researcher" paired with a "Writer") that hand off execution states to one another. Packaged within a Docker container, it can run complex automation scripts natively on a cloud server.

6. Autogen (by Microsoft)

- **What it is:** A highly scalable developer framework for constructing multi-agent conversation topologies.
- **Core Function:** Enables complex interactions among multiple specialized agents that can chat with one another to solve tasks. It features robust support for secure, sandboxed code execution loops inside secondary containers.

7. Langflow

- **What it is:** An open-source visual IDE for prototyping, building, and deploying LangChain-based AI agents.

- **Core Function:** Simplifies the complicated architecture of agent memory components, vector stores, and tool execution matrices into logical drag-and-drop components that expose the built agent as an independent REST API.

8. OpenDevin / AllHands

- **What it is:** An open-source, autonomous AI software engineer designed to replicate advanced coding behaviors.
- **Core Function:** Interacts directly with workspaces, runs a bash terminal inside a highly secure Docker sandbox, edits source files, and tests applications autonomously over ssh/cloud instances.

9. Google ADK (Agent Development Kit)

- **What it is:** Google's enterprise-grade open-source framework tailored for agent workflows.
- **Core Function:** Weaves deterministic logic with LLM reasoning. It uses an `api_server` command that packages an agent directly into a standard Docker container for seamless, native deployment onto GCP resources (like Cloud Run or Compute Engine VMs).

10. AutoGPT / Forge

- **What it is:** One of the original pioneering autonomous agent platforms built for continuous multi-step task execution.
- **Core Function:** Operates via a continuous loop of thinking, deciding, and acting. It tracks progress toward long-term goals by executing system processes and writing to local storage within its isolated container workspace.

AI Agents for the Internet (Web Search & Research Agents)

An **Internet Search AI Agent** is a specific class of agent specialized in real-time information retrieval, verification, and knowledge synthesis. Unlike a generic model that only knows facts up to its training data cutoff, an internet agent leverages tools like search engines, headless web browsers, and text scrapers to explore the live web.

What they do:

- **Deep Web Exploration:** They don't just look at the first link; they perform parallel searches, evaluate sources, and click through sub-pages.
- **Anti-Hallucination Filtering:** They read multiple sources to cross-reference data points, validating dates, metrics, and technical facts before summarizing.
- **Continuous Monitoring:** They can run on automated cron-triggers to watch specific domains, documentation pages, or financial indexes, compiling reports on changes.

Top 10 Internet Search & Research AI Agents (Docker/GCP Compatible)

These platforms and search-specific agent engines can be readily deployed on a self-hosted GCP instance using Docker:

1. Perplexica

- **What it is:** An open-source, self-hosted AI-powered search engine designed to act as an alternative to Perplexity AI.
- **What it can do:** Uses advanced agents to map queries, fetch live data using search APIs (like SearXNG), scrape target pages, and synthesize deeply cited research papers.

2. SearXNG

- **What it is:** A privacy-respecting, open-source metasearch engine that serves as a vital backend tool for AI agents.
- **What it can do:** Aggregates search results from over 70 web search engines without tracking users. Internet agents use it as a robust API layer to crawl the web without getting blocked by Captchas.

3. OpenPerplexity

- **What it is:** A minimalist, lightweight open-source web search assistant.
- **What it can do:** Leverages open LLM endpoints combined with live web-scraping pipelines to offer a clean, chat-based research interface directly out

of a single Docker container.

4. GPT-Pilot / Pygma

- **What it is:** A research and scraping agent pipeline optimized for gathering structural data.
- **What it can do:** Navigates complex, JavaScript-heavy sites using headless browsing elements, bypasses basic anti-bot structures, and extracts cleanly formatted JSON knowledge blocks from raw web pages.

5. Scraping Assistant Agent (by Flowise/Langflow)

- **What it is:** A customized agent setup utilizing web-browser nodes within Langflow or Flowise.
- **What it can do:** Accepts a high-level goal, uses a search engine tool node to discover relevant links, parses the raw HTML, cleans the noise out of the text, and processes it into a vector store.

6. Crawl4AI

- **What it is:** An open-source, LLM-friendly web crawler and scraping tool optimized for autonomous agent use.
- **What it can do:** Runs as a Docker service that accepts a URL, strips away ads, scripts, and layout elements, and returns clean, compressed markdown specifically formatted for an AI agent's prompt window.

7. Jina AI Reader

- **What it is:** A web-to-markdown API tool that can easily be run locally via Docker.
- **What it can do:** Allows your autonomous agent to search the live web or read any URL by appending it to a local container endpoint, converting complex web data into token-efficient reading material for the agent's core model.

8. MindOS / Autonomous Web Agent

- **What it is:** A framework focusing heavily on web-automation capabilities for AI assistants.

- **What it can do:** Controls browser instances to handle multi-step actions on the live web, such as searching forums, logging into specific portals, navigating pagination elements, and generating research summaries.

9. Browser-Use

- **What it is:** A fast-growing python-based agentic library explicitly built to let AI agents control web browsers.
- **What it can do:** Renders a Chromium browser instance inside a Docker container. The agent can literally see the webpage layout, move a virtual cursor, type into fields, click search buttons, and interact with the internet exactly like a human user.

10. Multi-On Agent (Self-Hosted/Docker Wrapper)

- **What it is:** An advanced web execution agent framework designed for deep web tasks.
- **What it can do:** Takes a high-level instruction (e.g., "Find the cheapest direct flight from Cebu to Tokyo next Tuesday and compare it to historical prices"), searches multiple platforms simultaneously, balances multiple tabs, and extracts an actionable final decision matrix.

I'm been using the Microsoft Log Parser at work to query Windows event Logs and AMAT equipment logs from 60 equipment with one click. Can the AI agent replace my Log Parser so that in the morning or real time , I have a screen showing which server is having an issue like log files are full and clean up for me?

The short answer is **yes, absolutely**. An AI Agent can replace manual Log Parser scripts and go a step further by not only detecting issues in real time but also taking autonomous action (like cleaning up disk space) and presenting the status on a clean dashboard.

However, because you are dealing with **AMAT (Applied Materials) semiconductor equipment logs** and critical Windows Event logs across 60 machines, the implementation requires a structured, secure approach. In the semiconductor industry, letting an AI agent blindly execute cleanup scripts on an active tool or server can be a major risk to manufacturing yield or uptime if it deletes the wrong file.

Here is a breakdown of how an AI Agent can transform your morning routine, how it compares to your current setup, and how to build a safe, closed-loop automation system.

The Shift: Log Parser vs. AI Agent

Feature	Your Current Microsoft Log Parser Setup	The AI Agent Setup
Trigger	Manual: You click a button in the morning to run the query.	Continuous/Real-time: Runs 24/7 or on a cron schedule before you wake up.
Analysis	Deterministic: Looks for exact string matches or thresholds you hardcoded.	Cognitive: Detects anomalies, sudden volume spikes, or unique error patterns.
Output	Raw text, CSV, or basic datagrid outputs.	Dashboard + Natural Language: A visual status board with a summary of <i>why</i> an alert fired.
Action	Passive: Tells you there is a problem; you have to fix it.	Active (Closed-Loop): Detects full logs, executes a cleanup script, and verifies it's fixed.

How to Architect This on Your System

To achieve this without putting your 60 pieces of equipment at risk, you should use an **Agentic Workflow Engine** (like **n8n** or **Dify**) deployed via Docker, combined with an open-source dashboard solution.

Here is the step-by-step logic the agent would execute every morning at 5:00 AM:

1. The Data Gathering Layer (The "Perception")

Instead of the agent directly logging into 60 machines individually via SSH/WinRM every minute (which causes massive network overhead), you use a lightweight log shipper (like Fluent Bit or OpenTelemetry) on the 60 servers. They stream the Windows Event Logs and AMAT log directory sizes into a centralized system (like a lightweight database or a local directory on your server).

2. The Agent Reasoning Layer (The "Thinking")


Using a self-hosted agent framework, you set up an agent with a specific toolset:

- **Tool A:** Read disk usage database.
- **Tool B:** Execute pre-approved PowerShell cleanup script.
- **Tool C:** Update Dashboard UI.

The agent checks the data. If it sees Server_42 has a log directory at 98% capacity, it enters its reasoning loop: *"Server 42 log files are full. I need to clear the temporary IIS logs or rotate the older AMAT equipment logs, but preserve the last 7 days of operational data."*

3. The Action Layer (The "Execution")



The agent calls **Tool B** to run a deterministic script that targets only the safe-to-delete files.

 **Critical Semiconductor Guardrail:** Never give an AI agent a raw, unconstrained Bash or PowerShell terminal to "figure out" how to clean a disk on production equipment. Instead, give the agent a **tightly scoped tool** (e.g., a script that *only* runs Remove-Item on specific .tmp or .log files older than 30 days).

4. The Morning Dashboard (The "Interface")

When you sit down at your desk, you don't look at code. You open a local web page (hosted via your Docker setup) powered by something clean like **Grafana** or the built-in dashboard of **Dify/n8n**.

It shows you a clean grid of your 60 servers:

-  **59 Servers Healthy**
-  **1 Action Taken:** *Server_42 disk was at 98% at 05:14 AM due to full event logs. Executed standard 30-day rotation script. Current capacity: 42%. Status: Resolved.*

Recommended Stack for Your Setup

If you want to host this entire pipeline locally or on a private cloud VM via Docker, this is the most reliable stack to achieve it:

1. **Orchestration Engine: n8n (Docker)**

- *Why:* It excels at mixing deterministic logic (if disk > 90%, do this) with advanced AI agent nodes. It connects easily to Windows environments via webhooks or SSH nodes.

1. The LLM Brain: Local Ollama (or Secure Enterprise API)

- *Why:* If your equipment logs contain sensitive proprietary operational data, running a local model like Llama-3 via Ollama inside your network ensures no data ever leaves your perimeter.

1. The Frontend: Grafana or n8n Dashboard

- *Why:* It provides that "one-screen" visual grid showing the real-time health of all 60 endpoints.

Would you prefer to look at a conceptual workflow of how n8n maps out an agentic log-cleanup pipeline, or should we dive into how to securely connect a Docker-based agent to Windows Event logs?

Are there AI Agents for Postgres and mariaDB?

Yes. Both **PostgreSQL** and **MariaDB** have mature, native AI agent ecosystems.

Database AI agents generally fall into two categories:

1. **DBA Agents (Ops):** Agents that act like virtual Site Reliability Engineers (SREs). They monitor your database, find slow queries, analyze logs, detect deadlocks, and suggest configuration tuning or index changes.
2. **Data Agents (App Dev):** Agents that let users or applications query data using natural language (Text-to-SQL) or interact with vector/semantic data via the new **Model Context Protocol (MCP)**.

Both ecosystems have open-source, containerized options that you can spin up via Docker Compose on your Google Cloud Platform VM.

PostgreSQL AI Agents

Postgres has a highly advanced AI ecosystem, driven heavily by its popularity in open-source cloud architectures.

1. Xata Agent (Open Source)

- **What it is:** A specialized, open-source AI agent designed to act as an on-call PostgreSQL expert and SRE.

- **What it does:** It monitors database metrics and logs 24/7. It uses advanced LLM reasoning to pinpoint root causes for high CPU usage, deadlocks, and slow queries, then generates actionable playbooks and SQL fixes.
- **Docker/GCP Ready:** Yes. They provide official Docker images and a docker-compose.yml file. You can deploy it right next to your Postgres instance on GCP to watch performance metrics securely without your data leaving the network.

2. PostgresAI Engine & Assistant

- **What it is:** A deep production-grade optimization platform built for automated performance tuning.
- **What it does:** It focuses heavily on "self-driving" Postgres operations. It automatically continuously cleans up unused or redundant indexes, mitigates database bloat, and optimizes over 20+ runtime configuration parameters based on real-world workloads.
- **Docker/GCP Ready:** Yes, built specifically for fast-moving DevOps teams running containerized Postgres environments.

3. pgEdge Agentic AI Toolkit

- **What it is:** An enterprise-focused framework designed to connect autonomous agents safely to distributed Postgres databases.
- **What it does:** It includes a dedicated high-performance **MCP Server** that gives external AI agents structured, deep schema reasoning capabilities. It lets agents generate and validate safe business logic without human intervention.

MariaDB / MySQL AI Agents

MariaDB has heavily integrated native AI features directly into its enterprise and open-source stacks, removing the need for a lot of third-party middle-tier code.

1. MariaDB DBA Copilot (MariaDB Cloud / Enterprise)

- **What it is:** An autonomous database administrator agent built natively into the MariaDB platform.

- **What it does:** It is tailored for operational tasks like debugging engine errors, reading system state logs, and performing real-time query performance tuning.
- **Docker/GCP Ready:** You can leverage the MariaDB Enterprise stack via Docker containers to give these built-in agents a safe, isolated workspace to manage your relational data.

2. MariaDB Enterprise MCP Server

- **What it is:** A bridge designed to connect language models directly to MariaDB database kernels using the Model Context Protocol (MCP).
- **What it does:** It exposes your MariaDB schemas securely to external agent frameworks or AI-native IDEs (like Cursor or Windsurf). The AI can read schemas, understand tables, execute standard SQL queries, and perform semantic vector searches natively inside the database.

3. MariaDB AI RAG (Multi-Container Stack)

- **What it is:** An all-in-one open-source pipeline provided by MariaDB on GitHub.
- **What it does:** It coordinates background workers (using Celery, Redis, and FastAPI) to let AI agents ingest documents, convert them to vectors, and execute complex semantic search loops entirely inside a MariaDB (v11.8+) database.
- **Docker/GCP Ready:** Yes, it is explicitly packaged as a multi-container docker-compose.yml stack designed for self-hosting on cloud instances.



How they run safely in production

Just like monitoring your AMAT or Windows server logs, you shouldn't let an AI agent have unconstrained root access to drop tables or rewrite production schemas.

When deploying these agents on your GCP Docker instance, the standard practice is to use a **Least Privilege Connection**:

1. Create a restricted database user (e.g., ai_agent_user).

2. Grant it SELECT access only to performance schemas, system catalogs (like `pg_stat_activity` in Postgres), or log views.
3. If using an optimization agent like Xata, configure it to run in **Advisory Mode** first—where it outputs the recommended CREATE INDEX or ANALYZE commands to your dashboard, requiring a human click to execute.