



Connect

Red Hat Research:

Deep-research agents on Red Hat AI
Factory with NVIDIA

Robbie Jerrom

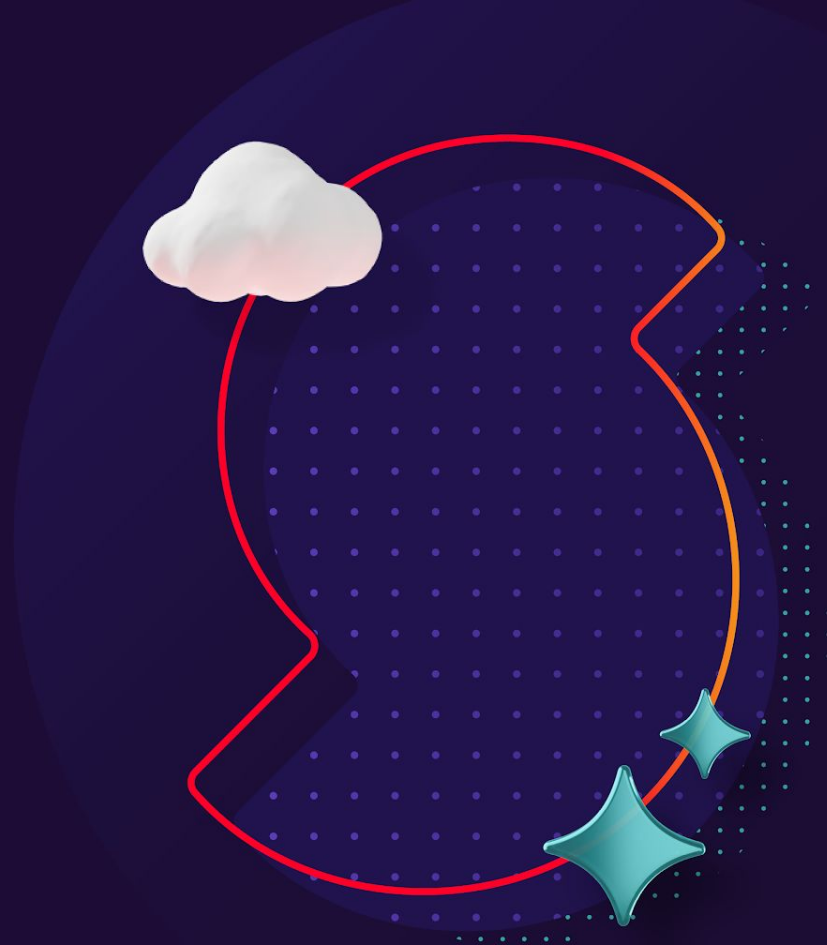
Senior Principal

Technologist, AI, Red

Hat

Tobacco Dock, London

8 October 2026





Connect

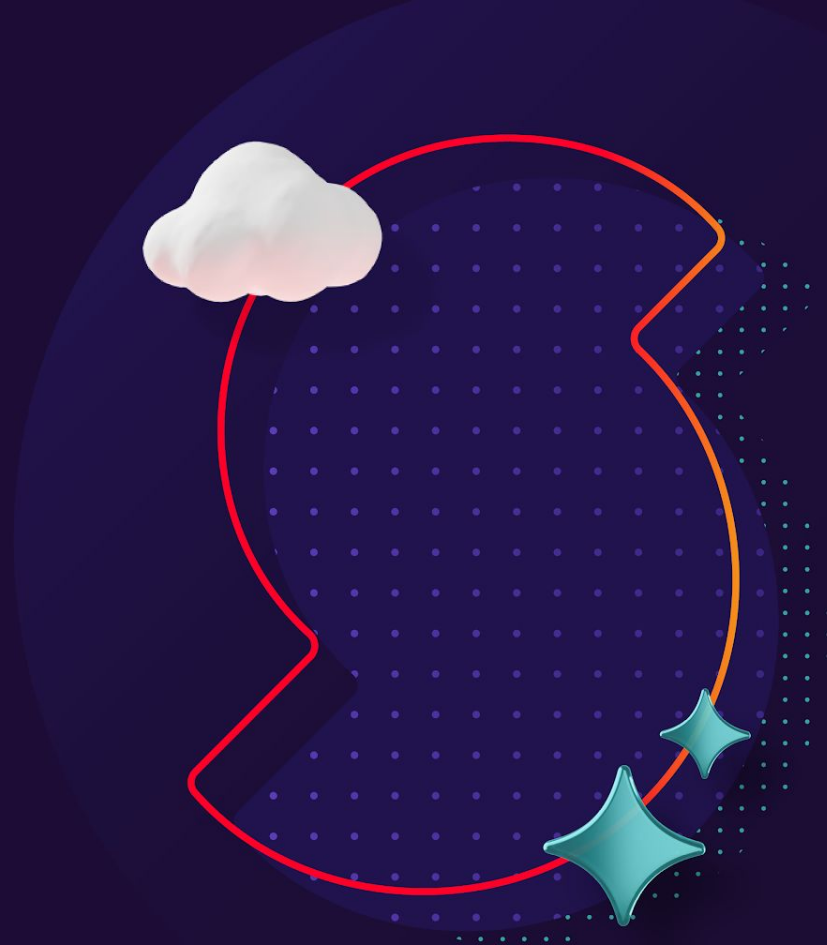
Red Hat Research

Deep-research agents on Red Hat AI
Factory with NVIDIA

Robbie Jerrom

Senior Principal Technologist AI,

Red Hat



An analyst's afternoon, in minutes

THE QUESTION WE GAVE THE AGENT

What do the EU AI Act and UK AI regulation each require of organizations deploying general-purpose AI models in 2026?

1 Classify

Intent: deep research



2 Route

Pick the sources



3 Plan

Structured queries



4 Research

N workers in parallel



5 Write

Cited report

WHAT YOU WILL SEE

Now The run

Recorded on our cluster, sped up

Section 3 Its trace

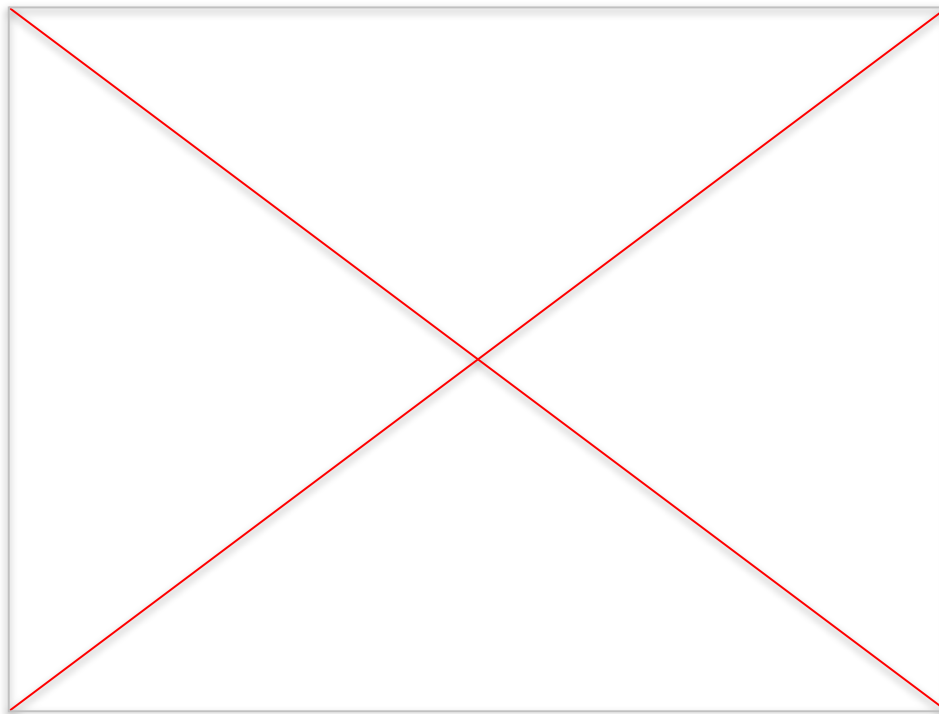
Every step in MLflow

The close The report

Cited, checked, finished



Research Agent In Action



Recorded on OpenShift AI: Nemotron 3 Super on one H200, AI-Q 2.2. Sped up.

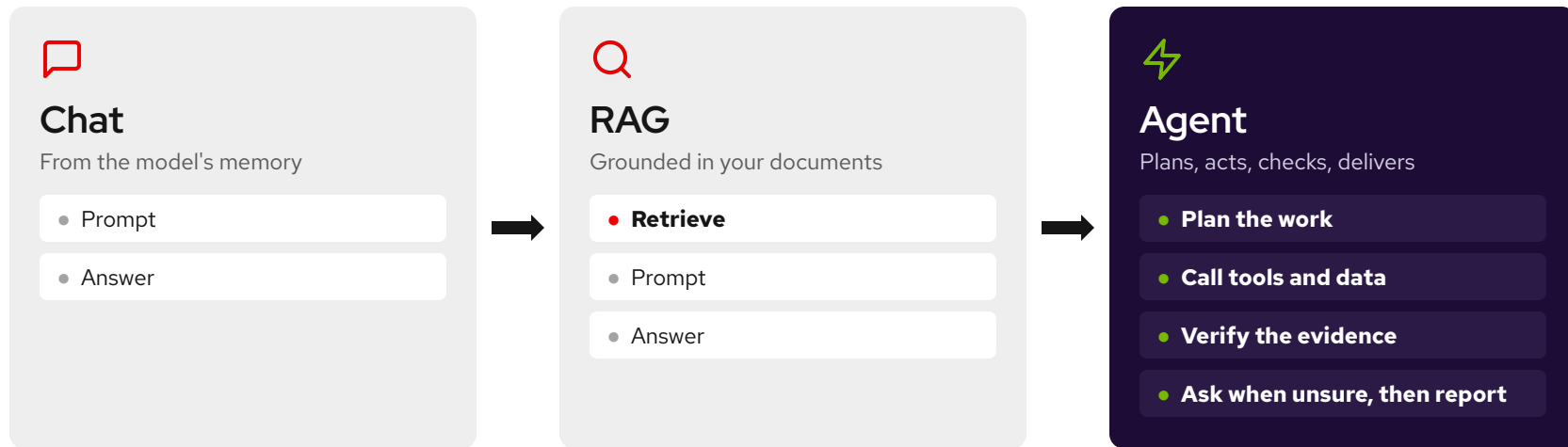


The power of agents

From answering questions to doing the research



From answers to outcomes: the agent loop



The shift: every loop multiplies tokens, tool calls and risk. That demands a platform.



Why research is the right first agent

REACHED IN MARCH 2026

#1

DeepResearch Bench I · 55.95

#1

DeepResearch Bench II · 54.50

Leaderboard runs used a fine-tuned Nemotron 3 Super: the model family our demo runs.



Every team does it

Analysts, policy, competitive intelligence and literature review: hours of search, read and synthesize.



It can be measured

Citations are checkable, and public benchmarks (FreshQA, DeepResearch Bench) track quality.



The risk is bounded

It reads sources and writes reports. Default profiles run no generated code.

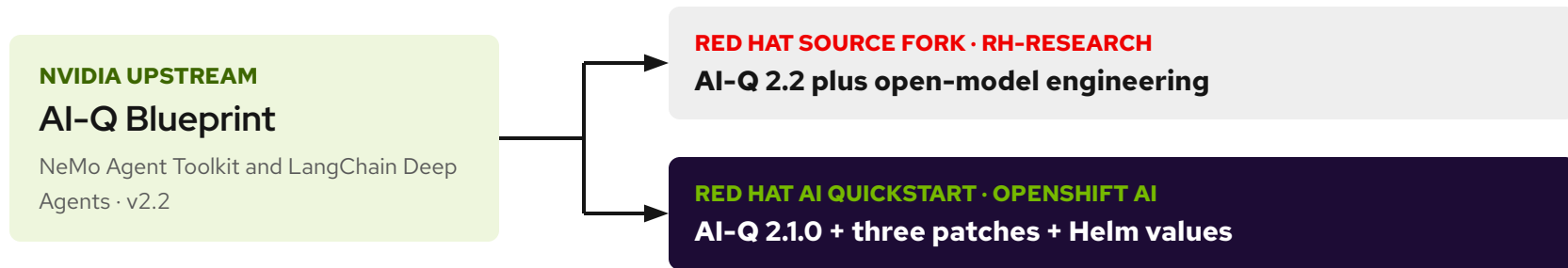


Inside Red Hat Research

The NVIDIA AI-Q architecture, made open-model and OpenShift ready



Same AI-Q engine, open to any model



WHAT RED HAT ADDS

Open models

OpenAI to vLLM, MaaS or NIM, per role

Robust on vLLM

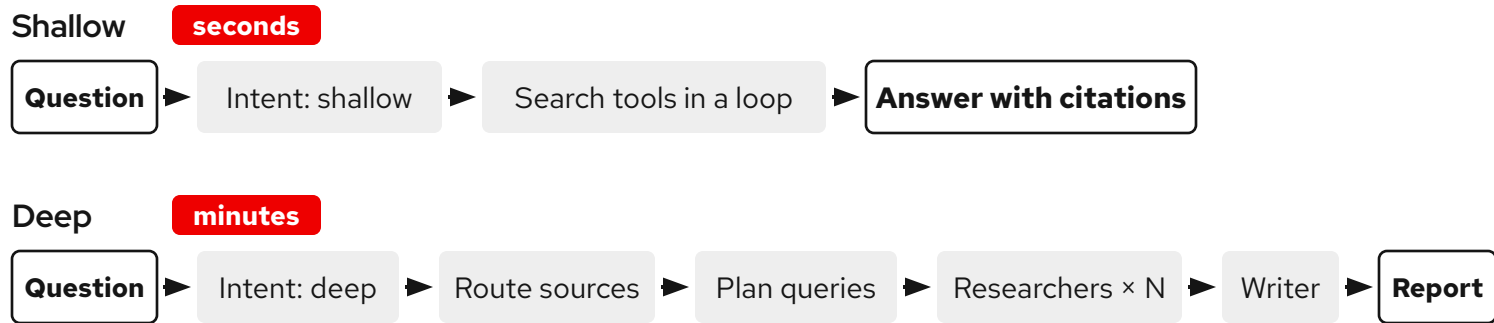
Tool choice, timeouts, citation repair

Red Hat AI ready

Helm, KServe, branding, MLflow export



One question, two paths



Every step streams to the UI as events: Tasks, Thinking and Report tabs update live.

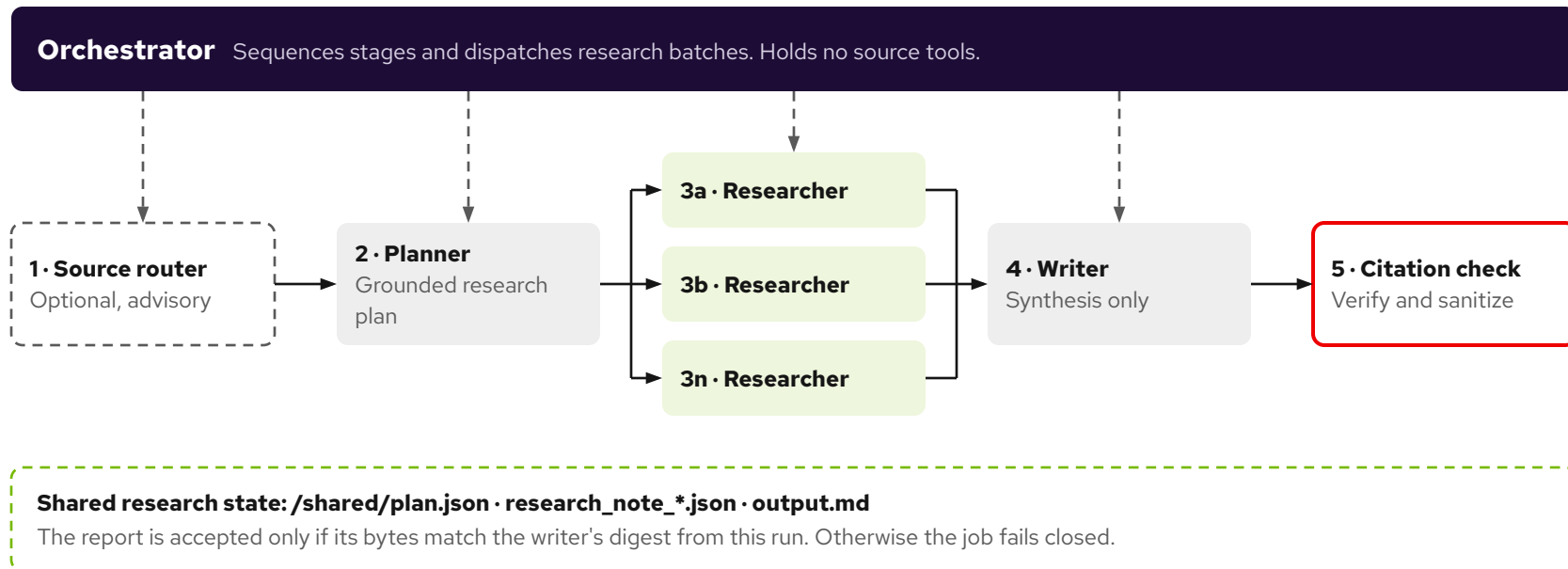
The intent classifier decides how much work a question needs.

Shallow questions get a sourced answer in one tool-calling loop.

Deep questions become a plan, parallel research and a written report.



Plan, fan out, write, verify



Bring your own tools, data and models

 **Web and papers**

TavilyExaYou.comNimbleSerper

 **Enterprise**

MCP serversPer-user OAuth (2.2)

 **Knowledge**

LlamaIndex + ChromaDBNVIDIA RAG Blueprint

OpenSearchAzure AI Search

 **Skills**

AnalysisReport writingCharts

Shown for the rh-research fork (AI-Q 2.2). Sources are chosen per request.

ONE MODEL, EVERY ROLE (DEMO)

Nemotron 3 Super

120B-A12B · FP8 · Red Hat AI validated

Serves

IntentClarifierPlanner

ResearchersOrchestrator

WriterSummaries

Swap any role to another OpenAI-compatible endpoint with one environment variable.

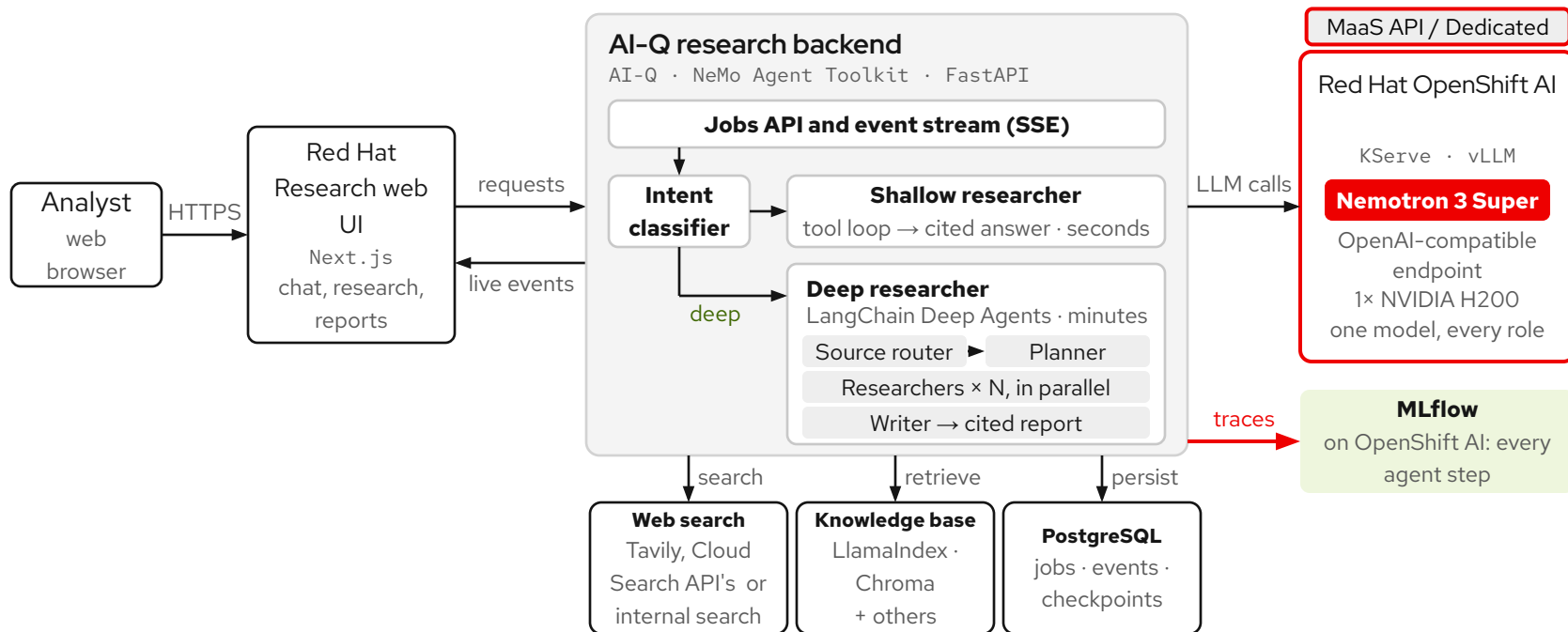


Production on AI Factory

What tokens, tool calls and risk demand from the platform



From a question to a cited report



One H200 runs the whole agent

Nemotron 3 Super - 120B-A12B · FP8 · Red Hat AI validated

1× NVIDIA H200 · 141 GB

Memory on the GPU (about 131 GiB)

Weights 114 GiB

-  Weights (FP8): about 114 GiB
-  KV cache: about 7 GiB, about 308k tokens
-  Runtime and reserve

Only 12B of 120B parameters are active per token, so one GPU has the compute. All weights still load into memory.

120B / 12B

Total and active parameters. Hybrid Mamba-Transformer, latent MoE.

83.63

MMLU-Pro at FP8, against 83.73 at BF16: effectively no loss.

256k

Context served, of up to 1M tokens the model supports.

MTP

Multi-token prediction layers ready for speculative decoding. Not enabled yet.



However... right-size the model for every role

(development on smaller models saves £££)

Large: Nemotron 3 Super

Planner, orchestrator, writer: the hard reasoning

Mid: Nemotron 3 Nano 30B

Researchers, shallow research, clarifier

Small: Nemotron Mini 4B

Intent classifier, document summaries

Illustrative split. The published quickstart also uses three tiers.

Shared, not blocked. vLLM batches requests continuously, so a classifier call never locks the GPU. It still takes batch slots, KV cache and compute on the big model, and queues behind long deep-research prompts when it is busy.



Faster easy steps

Small models answer routine calls with lower latency per call.



More room for hard steps

The large model's capacity goes to planning and writing.



Share the small models

Serve them once through Models-as-a-Service for every agent.

Evaluate before you split

Run the full workflow per model and compare MLflow traces and EvalHub scores: a model can pass every tool test and still fail every report.

Extra models need room: a GPU, MIG slices or MaaS.



Test the whole workflow, not single tool calls

Model	Where	Context	Tool calls	Deep research	Speed	Quality /10
<i>What good looks like</i>		128k+	20/20	Full report	25+ tok/s	Within 1 of best
Nemotron 3 Super FP8	H200 · demo	256k	98% live	Full, 23 min	59 tok/s	4.9 to 5.8 for all five: within noise
Nemotron 3 Super NVFP4	DGX Spark	128k	Deep runs OK	Full, 40 min	16 tok/s	
Gemma 4 26B-A4B	DGX Spark	128k	20/20	Full, 13 min	24 tok/s	
Qwen3.5 35B-A3B	DGX Spark	128k	20/20	Full, 18 min	55 tok/s	
Nemotron 3 Nano 30B	DGX Spark	256k	20/20	Short, 16 min	63 tok/s	
Mistral Small 24B	DGX Spark	32k	20/20	Failed, 0 of 3	5 tok/s	No report

WHAT THE RUN TOLD US

3.7× faster per token: H200 vs Spark

20/20 tool tests, yet 0 of 3 reports

Citations weakest score for every model

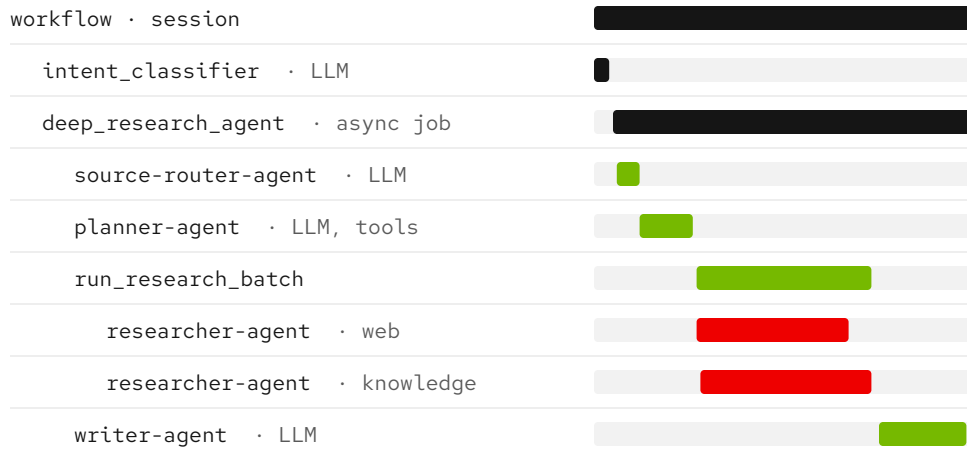
Also: 128k+ context for full reports · match tool-call and reasoning parsers to the model · max_tokens at 2 to 3× the expected output



Every step of every run, on the record

One deep-research run as an MLflow trace

Timing illustrative



Agent backend

NeMo Agent Toolkit OTLP exporter



OTLP over HTTPS

Experiment header and service token



MLflow on OpenShift AI

Fully supported since 3.4



Evaluate the traces

mlflow.genai.evaluate with scorers



Debug: slowest span first



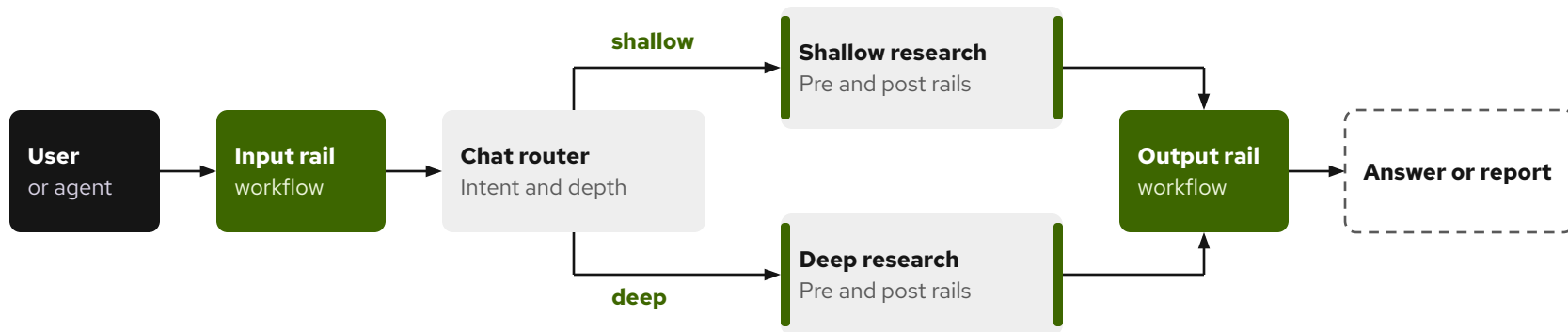
Tune: tokens by role



Evaluate: score from traces



Guardrails at every agent boundary



Pass

Modify

Block → refusal

Fails closed: rail errors return the refusal

Opt-in · AI-Q 2.2

NeMo Guardrails service

GA 3.4

On OpenShift AI: sensitive data, content filters, regex rails

MCP Gateway rails

TP 3.5

Checks agent tool calls on the platform

EvalHub red teaming

GA 3.5

Tests models, apps and agents before release



Production checklist: what is ready today

 Identity	Quickstart ships with auth off: enable it, plus OpenShift RBAC	Configure
 Data residency	Fully local: in-cluster models and embeddings, internal search	Air-gap ready
 Tracing	MLflow on OpenShift AI receives every agent span	GA · 3.4
 Guardrails	NeMo Guardrails service; opt-in agent middleware in AI-Q 2.2	GA · 3.4
 Evaluation	EvalHub for models, apps and agents, with red teaming	GA · 3.5
 Cost and quotas	Models-as-a-Service quotas and rate limits; per-user usage views	GA / TP
 Code execution	Off by default. OpenShell on OpenShift AI 3.5	Dev Preview



Demo, pilot, scale

01

Demo

Hosted models, no GPUs; running the same day



02

Pilot

vLLM on the customer's OpenShift AI, fully local if needed



03

Scale

Shared MaaS models; research offered to other agents over MCP

```
# from deploy/helm on the quickstart branch
helm install vllm-models vllm-models/ -n ns-aiq
helm install aiq aiq-rh/ -n ns-aiq \
  -f aiq-rh/values-vllm.yaml
```

LEAD WITH

Literature review

Policy analysis

Market intelligence

Knowledge synthesis



Get started

github.com/rh-ai-quickstart/rh-research

Quickstart: docs.redhat.com > AI quickstarts > Deploy Multi-Agent Research Workflows





Connect

Thank you

An analyst's afternoon in under half an hour: traced, evaluated and run on your own GPUs.



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



x.com/RedHat



Backup

Install detail and the published quickstart diagram



Two Helm installs to a running research agent

```
# 1. From deploy/helm: namespace and secrets
oc create namespace ns-aiq
oc create secret generic aiq-credentials \
  -n ns-aiq --from-literal=...

# 2. Models on KServe (vLLM)
helm install vllm-models vllm-models/ -n ns-aiq

# 3. Red Hat Research
helm install aiq aiq-rh/ -n ns-aiq \
  -f aiq-rh/values-vllm.yaml \
  -f aiq-rh/values-branding.yaml
```

MEASURED ON ONE H200, 1 OCT 2026

0.3 s

First token, Nemotron 3 Super FP8

About 15 s

Shallow research with citations

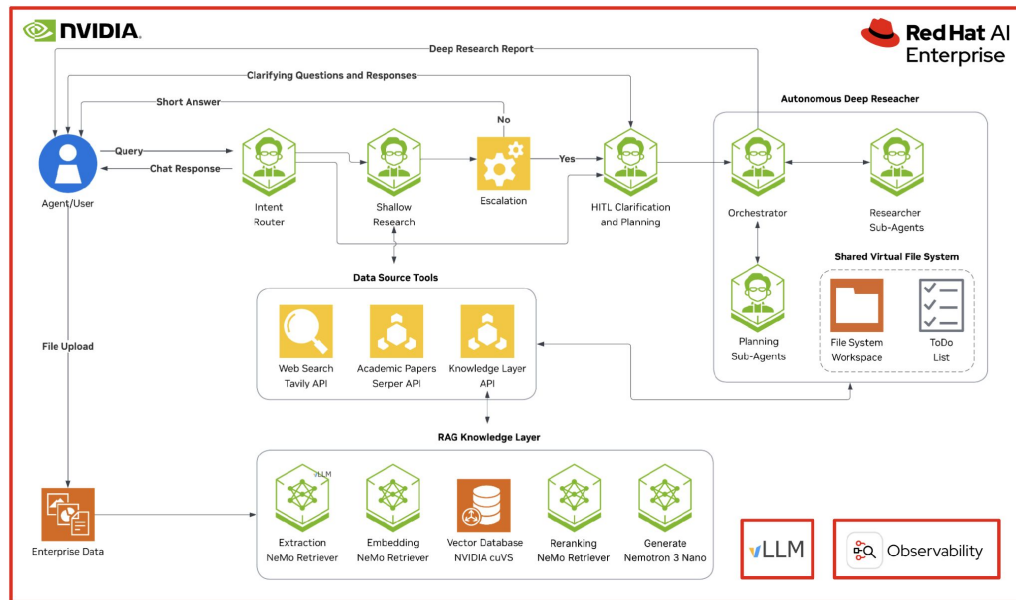
About 23 min

Deep research report, three-question average

The quickstart docs quote 10 to 30 s and 2 to 5 min for its 2.1 build and default models. Tested on OpenShift 4.20 and OpenShift AI 3.3.2.



The quickstart, as published



Green: NVIDIA AI-Q

Agents, tools and the NeMo Retriever knowledge layer

Red: Red Hat AI

vLLM serving and the observability stack

Find it

docs.redhat.com > AI quickstarts > Deploy Multi-Agent Research Workflows

