LLamaMe (LLM) API

LC user meeting

March 13th, 2025

Otto Venezuela | HPC Cloud Enabler Officer

otto@llnl.gov

representing many doers in LC

Prepared by LLNL under Contract DE-AC52-07NA27344.

Workflow Enablement Group

The nexus of cloud technologies, new workflows and traditional LC HPC ecosystems

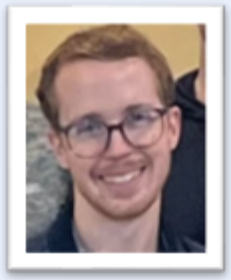
WORKFLOW ENABLEMENT GROUP

A deep bench of diverse talent to aid HPC with challenges of disruptive technology

Conner Cook

John Consolati
(Team Lead)

James Corbett

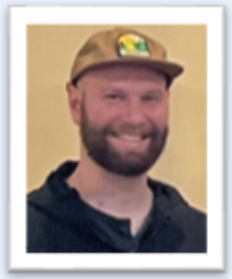


John Westlund

Todd Heer
(Group Lead)

Urwah Mir

Zeke Morton



Neil O'Neill

Jordan Dorham

James Taliaferro

Otto Venezuela
(Team Lead)

Crucial Cross-Group Collaborators/Contractors

Paul Bryant
(ORNL)

Thomas Mendoza
(contractor)

David Fox
(Project Lead)

Delivering on HPC Convergence

Realizing the promise of production DevOps engineering leveraging *On-premises* cloud-native technologies

Workflow Enablement Group

- **Persistent Data Services** (PDS) - curated containerized secure user-provisioned data applications
- **Bespoke software stack deployments** - enabling HPC right next to the "big iron"
- **Object Storage** - s3-protocol object data stores
- **AI** - workflows integrated with hosted LLMs; AI accelerator h/w integration
- **Container orchestration -** Kubernetes / OpenShift
- **Repos and image security scanning -** Nexus, Trivy, Gitlab registries
- **Continuous Integration** (CI) - Custom runners for local & multi-institutional cross-site CI
- **Workflow tools -** Merlin (LLNL), Sina (LLNL), Fireworks, CDash, AiiDA
- **Flux -** scheduler development fused with storage container and cloud integration
- **LC Web Services -** Atlassian suite, Gitlab
- **Container telemetry -** Advanced Cluster Monitoring, Elasticsearch, Kibana, Grafana, Prometheus
- **Automatable LC workflows** from inside & outside "the LC bubble" - trigger tokens, object store keys

Learn more at: hpc.llnl.gov/cloud

LLamaMe (LLM) API Service

- LLamaMe API represents a significant advancement in providing users with seamless access to locally-hosted open-source Large Language Model (LLM) via the REST API endpoint
- Providing this new LLamaMe API access allows users to integrate HPC jobs, integrated development environment (IDE), Retrieval-Augmented Generation (RAG) systems and LLM Agents.
- Currently deployed using Nvidia GPU nodes in CZ, RZ, and SCF OpenShift cluster

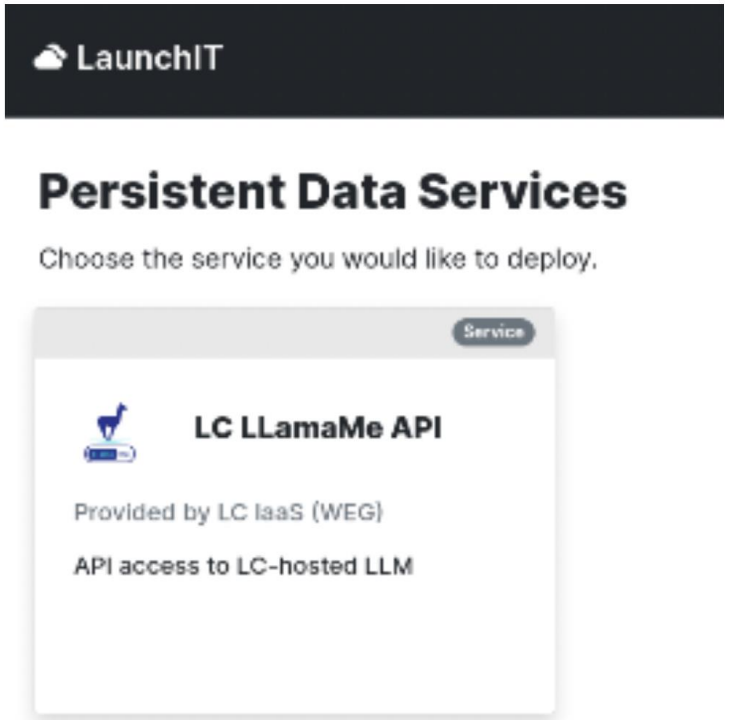
Network	Model	Max Tokens	GPUs
CZ	Meta-Llama-3.1-70B-Instruct-AWQ-INT4	4096	2 NVIDIA A100 80GB
RZ	Meta-Llama-3.1-8B-Instruct	4096	1 NVIDIA A100 40GB
SCF	<TBD: Mixtral 8x7B>	<TBD>	4 NVIDIA H100 80GB

NVIDIA

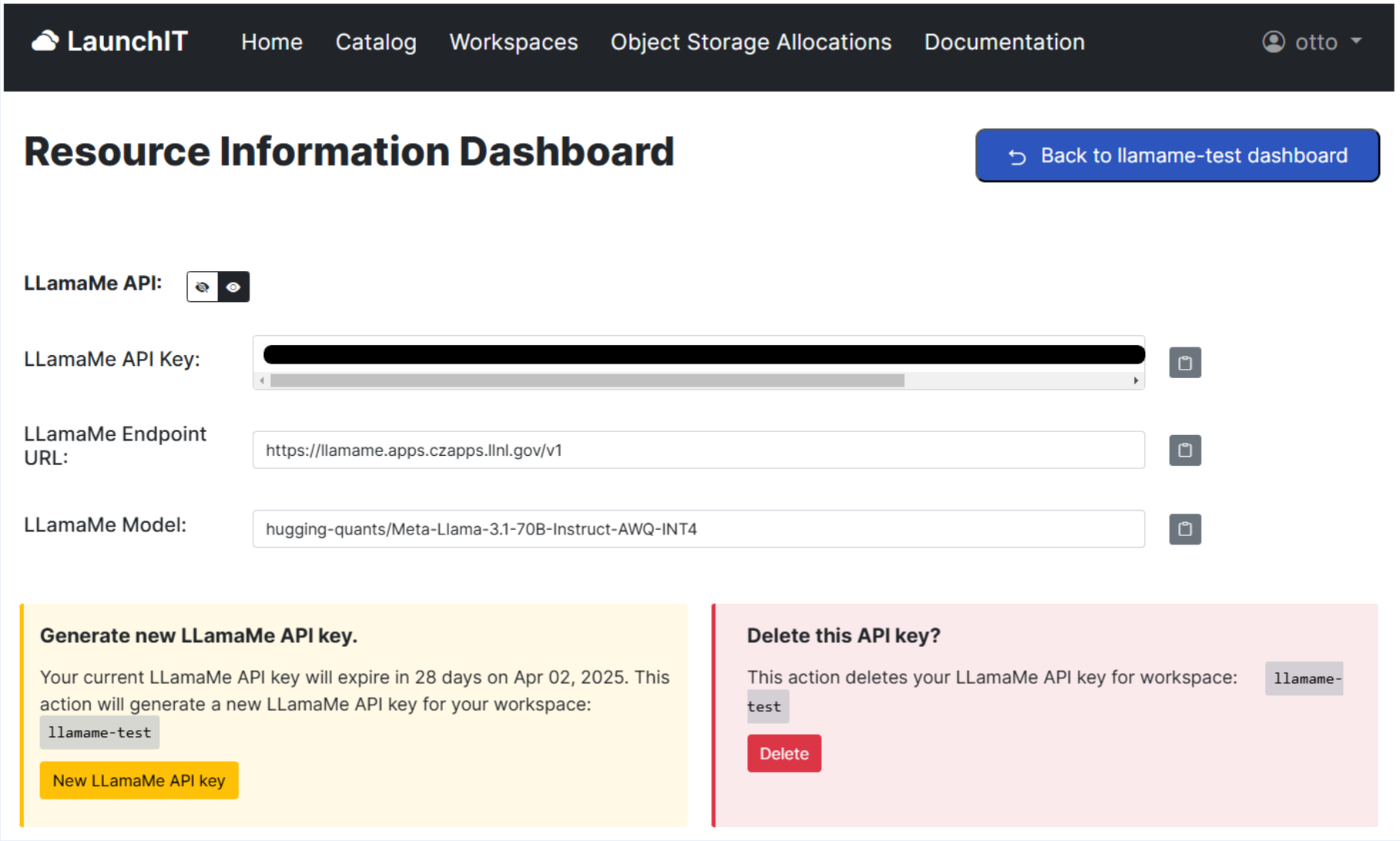
Available in CZ and RZ as Limited Availability (LA) since Sept. 2024 (SCF LA in April 2025)

Adding LLamaMe into LaunchIT upcoming release

- Upcoming LaunchIT ver. 2.3
- LLamaMe as new PDS service in LaunchIT
- Easy access to request API Key
- API Key valid for 30 days
- Regenerate API Key
- Rate limit 10 requests per minute

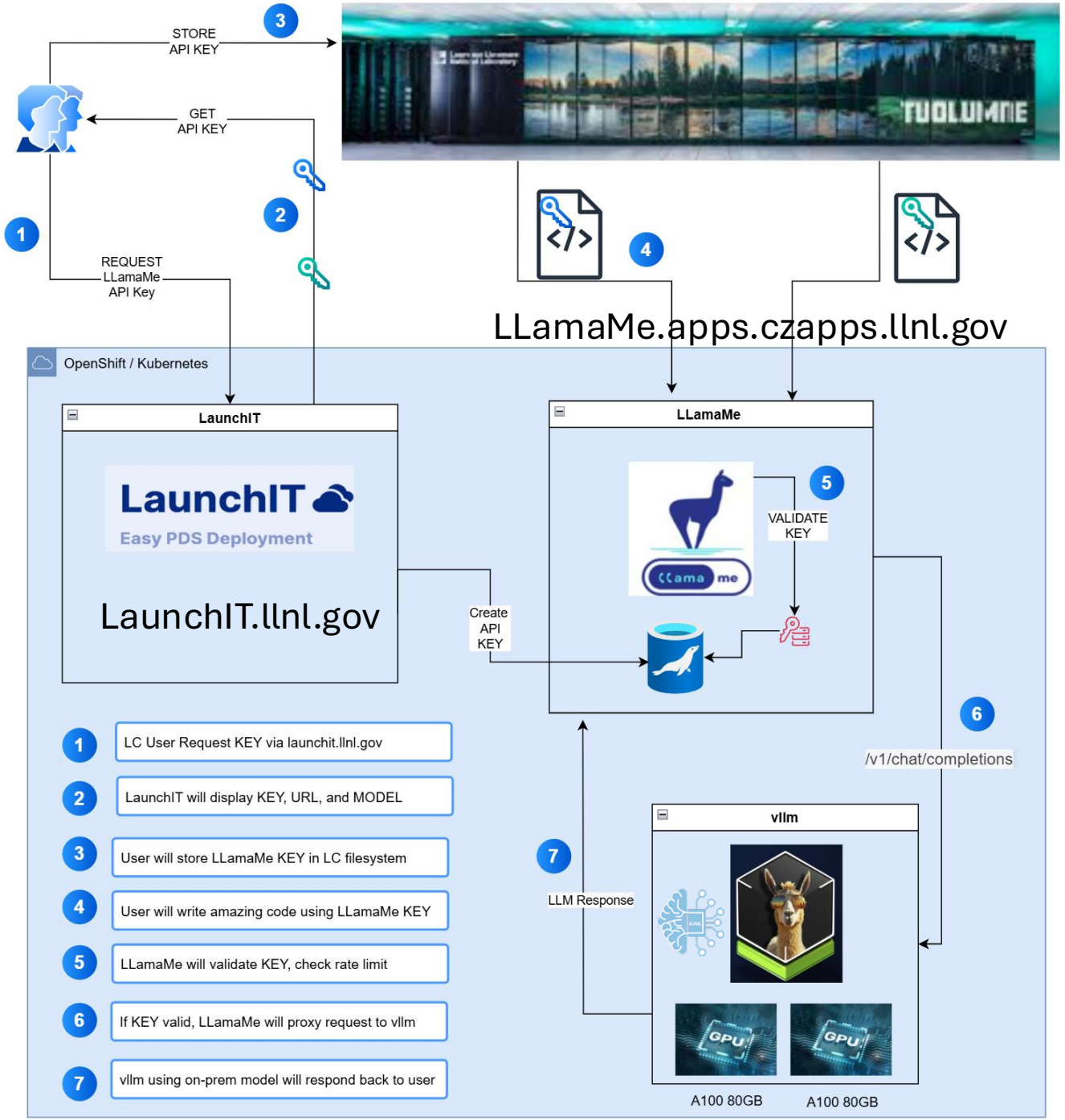


LC LLamaMe API in LaunchIT Catalog



WORKFLOW ENABLEMENT GROUP

API Token is valid for 30 Days and default rate limit of 10 request/minute!


STORE
API KEY
GET
API KEY
REQUEST
LLamaMe
API Key
TUOLUMNE
LLamaMe.apps.czapps.llnl.gov
OpenShift / Kubernetes
LaunchIT
LaunchIT
Easy PDS Deployment
LaunchIT.llnl.gov
LLamaMe
VALIDATE
KEY
Create
API
KEY
/v1/chat/completions
LLM Response
vllm
A100 80GB
A100 80GB
1 LC User Request KEY via launchit.llnl.gov
2 LaunchIT will display KEY, URL, and MODEL
3 User will store LLamaMe KEY in LC filesystem
4 User will write amazing code using LLamaMe KEY
5 LLamaMe will validate KEY, check rate limit
6 If KEY valid, LLamaMe will proxy request to vllm
7 vllm using on-prem model will respond back to user

Python Example


```python
import os
from openai import OpenAI

# LC WEG is hosting Opensource Large Language Model in CZ/RZ (soon in SCF!)
with open(os.path.expanduser('~/.lciaas/lc-llamame-api/API_KEY'), 'r') as f:
    API_KEY = f.read().rstrip()

client = OpenAI(
    base_url="<LLAMAME_URL>/v1",
    api_key=API_KEY,
)
# Check which LLModel LC is hosting
print(client.models.list())

chat_response = client.chat.completions.create(
    model="<ON-PREM MODEL>",
    messages=[
        {"role": "user", "content": "Tell me a joke."},
    ]
)
print("Chat response:", chat_response)

#Developed with :heart: by LC DOERs
```

```
Chat response:
  ChatCompletion(id='chat-787',
    message=ChatCompletionMessage
      (content='Why did the llama refuse to play poker? \n\nBecause he always got fleeced.',

      role='assistant',
      model='hugging-quants/Meta-Llama-3.1-70B-Instruct-AWQ-INT4',
      object='chat.completion', ... )
```

Don't have a LLamaMe KEY?

Go to LaunchIT to get your Key! (version 2.3)

[LaunchIT Documentation]

CZ / SCF: https://LaunchIT.llnl.gov

RZ: https://rzLaunchIT.llnl.gov

What is next?

- Update model to use Llama 3.3 70B
- Deploy LLamaMe - LaunchIT integration in OpenShift
- Enable email notifications when token is about to expire
- Can we support more than 1 model?
  - GPU nodes are expensive $$$
- LC SciChat - powered by paperQA Retrieval Augmented Generation (RAG)
  - Developed by LC DevOps Interns Team (DOIT)
  - DOERs: Kailey Wong and Arely Rosendes

Stay connected … call me (llámame)

WORKFLOW ENABLEMENT GROUP

Microsoft Team: LC LLamaMe (LLM) | General | Microsoft Teams

Docs: https://hpc.llnl.gov/cloud

Gitlab: LLamaMe Example Repo

Feedback: lit-help@llnl.gov

Bugs: lc-hotline@llnl.gov