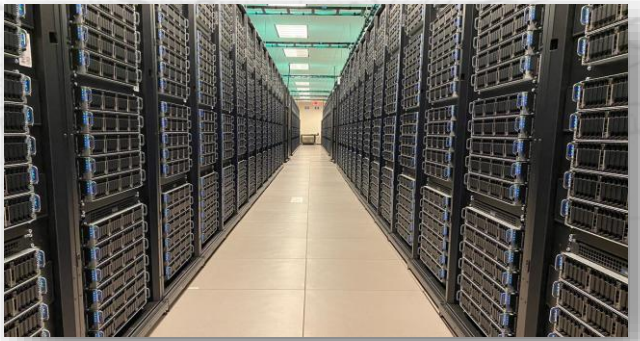
OPENSEES-ON-DESIGNSAFE TRAINING

LEVERAGING TACC'S HPC RESOURCES

Silvia Mazzoni, PhD

Applications Specialist @ DesignSafe

18 September 2025



WORKSHOP OBJECTIVE

- Provide an overview of OpenSees on DesignSafe & TACC
- Introduce the new OpenSees-on-DesignSafe Training Document
- For me to gain a better understanding of how users approach and use OpenSees on DesignSafe

…. I hope you “walk” away with more questions than answers!

WHAT IS YOUR EXPERIENCE LEVEL

- OpenSees Tcl
- OpenSeesMP
- OpenSeesSP

- OpenSeesPy
- OpenSeesPy-MPI
- OpenSeesPy + concurrent futures

- OpenSees on DesignSafe Web Portal
- OpenSees on DS JupyterHub
- OpenSees & Tapis
- OpenSees on HPC via SSH & slurm Jobs

Low
Medium
High

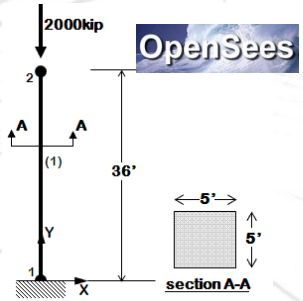
OUTLINE

Why Run OpenSees on DesignSafe

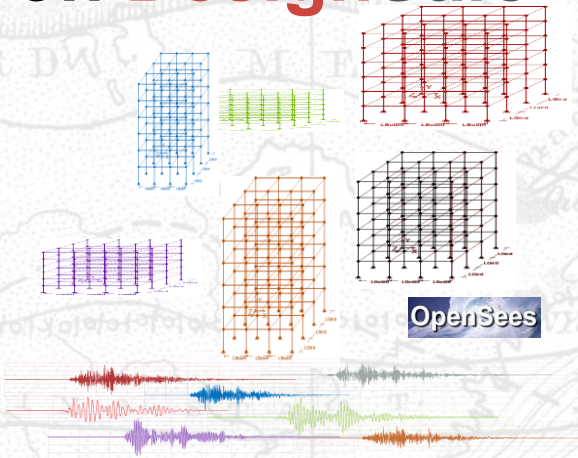
When To Run OpenSees on DesignSafe

How To Run OpenSees on DesignSafe

Why Run OpenSees on DesignSafe



OpenSees Wiki
Example 1



Scalable Resources & Environments

NSF NHERI DESIGNSAFE | TEXAS The University of Texas at Austin | UCLA | TACC | RICE | Florida Tech

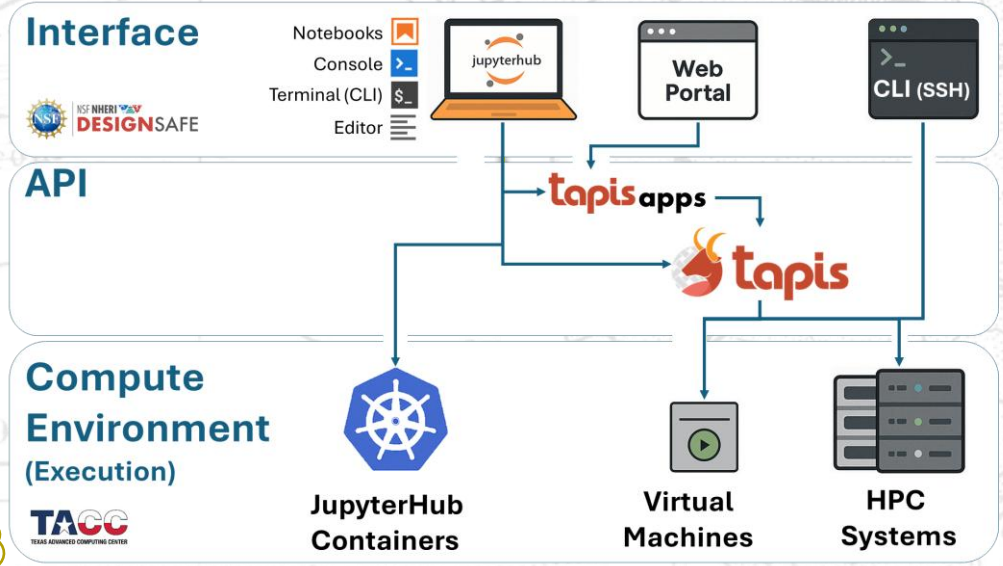
When To Run OpenSees on DesignSafe

Analysis Type	Memory Usage	CPU vs GPU	Parallelism Needs	Speed Sensitivity	Description + Reason / Notes
Parallel Monte Carlo	🔹 Low–Moderate	✅ CPU preferred	🔹 High (trivially parallel)	🔸 Low–Moderate	Monte Carlo methods involve running repeated random samples to solve deterministic problems that are too complex for direct computation. Each run is independent; minimal memory overhead, great for CPU clusters or grid computing.
Parametric Sweeps	🔹 Low–Moderate	✅ CPU preferred	🔹 High	🔸 Low–Moderate	Run a model or simulation across a range of input parameters to study system behavior or performance variation. Like Monte Carlo, easily parallelized with little inter-process communication.
ML Training Loops	🔸 Moderate–High	🔴 GPU accelerated	🔹 Moderate–High	🔴 High	ML training involves iterative updates to a model's parameters using a training dataset. GPU-accelerated for fast matrix ops; training requires high compute, memory usage depends on model size.
Coupled Simulations	🔴 High	✅ CPU dominated	🔸 Low–Moderate	🔴 High	Simultaneously simulate multiple interacting physical domains or solvers (e.g., fluid + structure, heat + stress). Memory-intensive due to mesh/data exchange between solvers; limited parallelization unless domain-decomposed.
Stepwise Simulation	🔸 Moderate	✅ CPU preferred	🔸 Moderate	🔸 Moderate	A sequential, time-marching simulation solving physical phenomena across a time domain, e.g., using finite difference, finite element, or finite volume methods. Each step may use iterative solvers; memory builds with history data.
Batch Pre/Post-Processing	🔹 Low	✅ 🔴 CPU or GPU capable	🔹 High	🔸 Low–Moderate	Transform or clean a large batch of data files or inputs to prepare for further analysis or training. Lightweight tasks like data cleaning, often highly parallel and not memory-intensive.

Scalable & Adaptable Resources & Environments

NSF NHERI DESIGNSAFE · TEXAS The University of Texas at Austin · UCLA · TACC · RICE · Florida Tech

How To Run OpenSees on DesignSafe



Interface Environments:

1. Jupyter Hub
2. Web Portal
3. CLI (SSH)

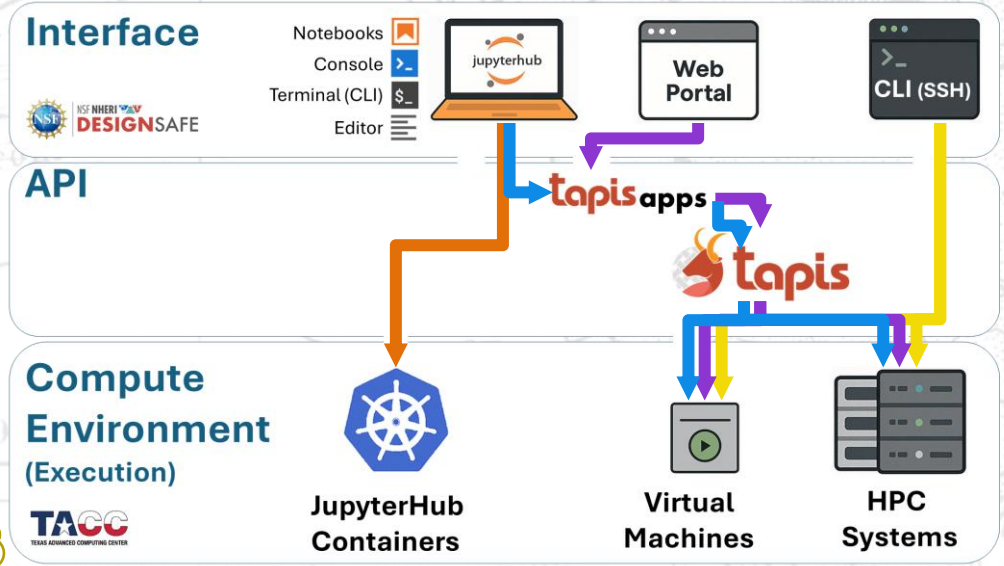
API:

1. Tapis Apps
2. Tapis

Compute Environments:

1. Jupyter Hub
2. Virtual Machines
3. HPC Systems on TACC

Ways To Run OpenSees on DesignSafe



Interface Environments:
- Jupyter Hub
- Web Portal
- CLI (SSH)

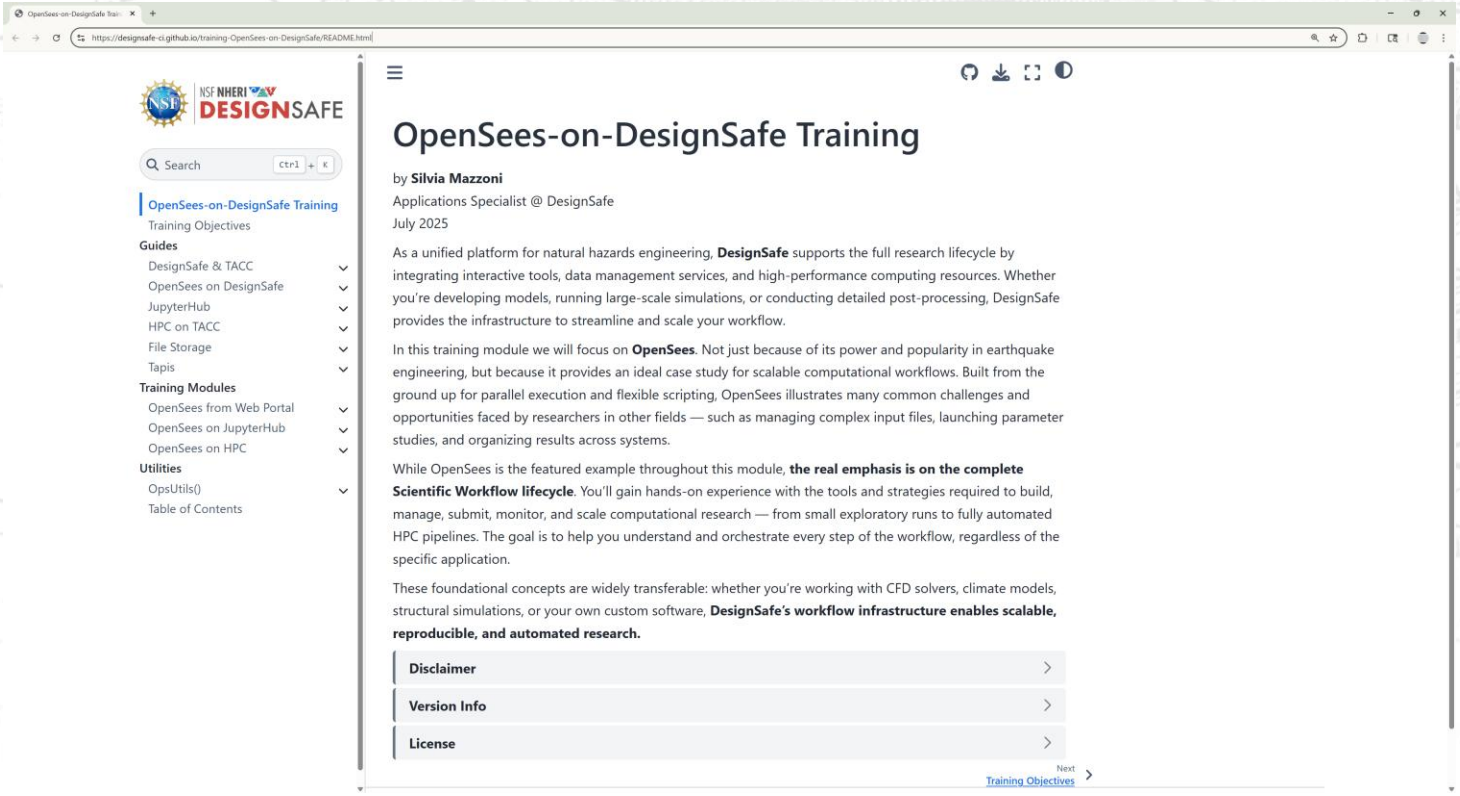
API
- Tapis Apps
- Tapis

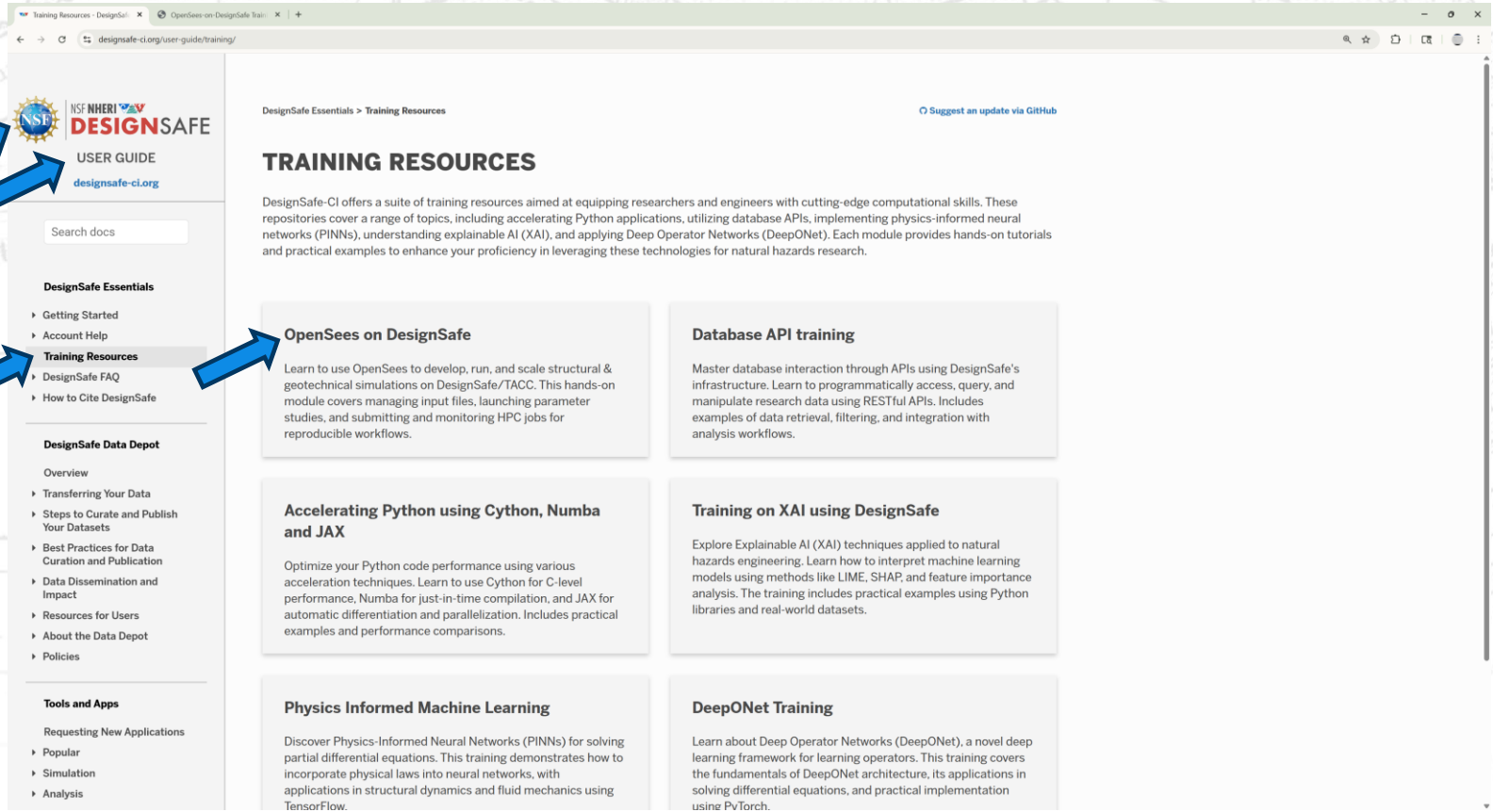
Compute Environments:
- Jupyter Hub
- Virtual Machines
- HPC Systems on TACC

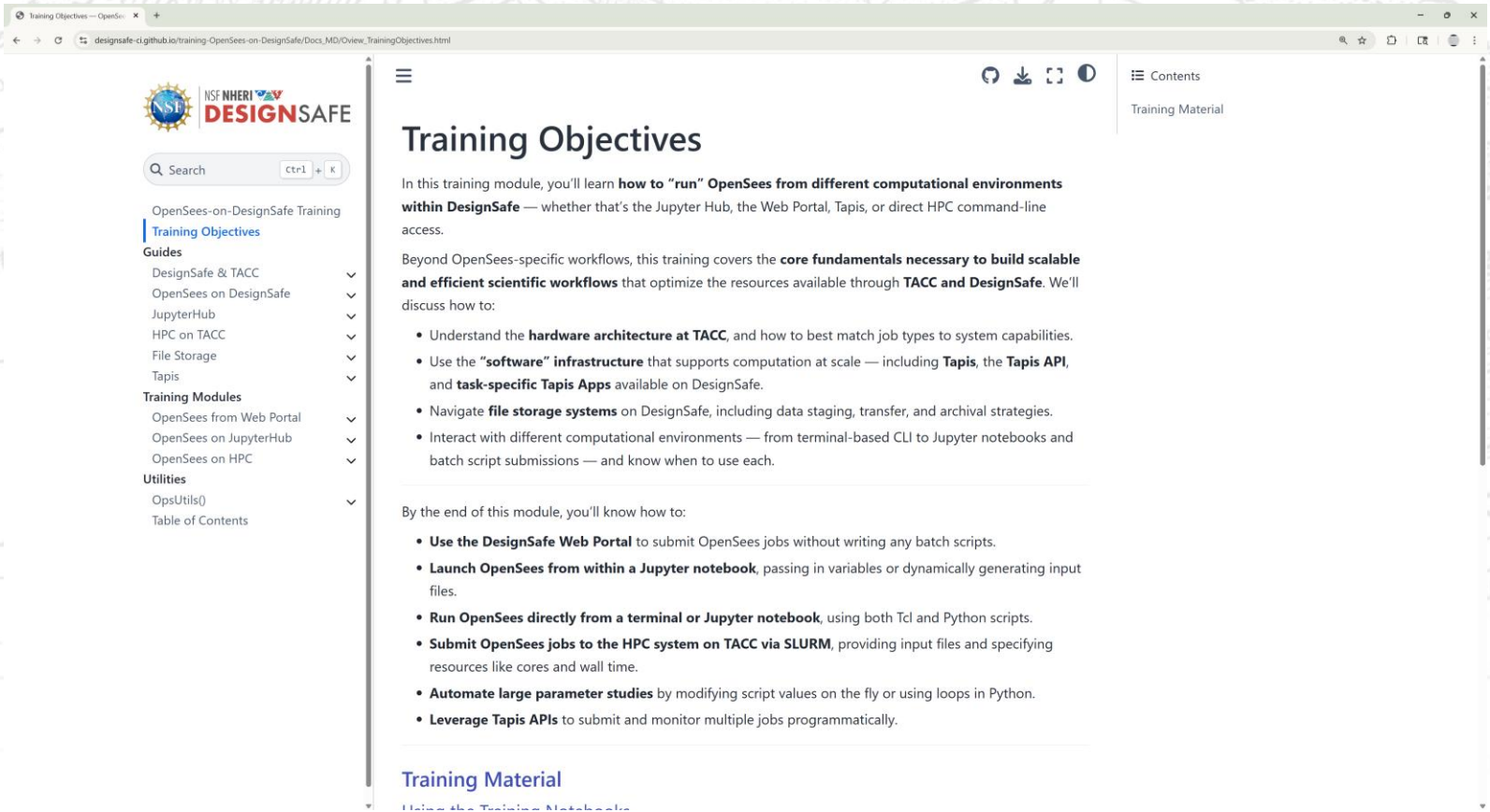
Run OpenSees on DesignSafe

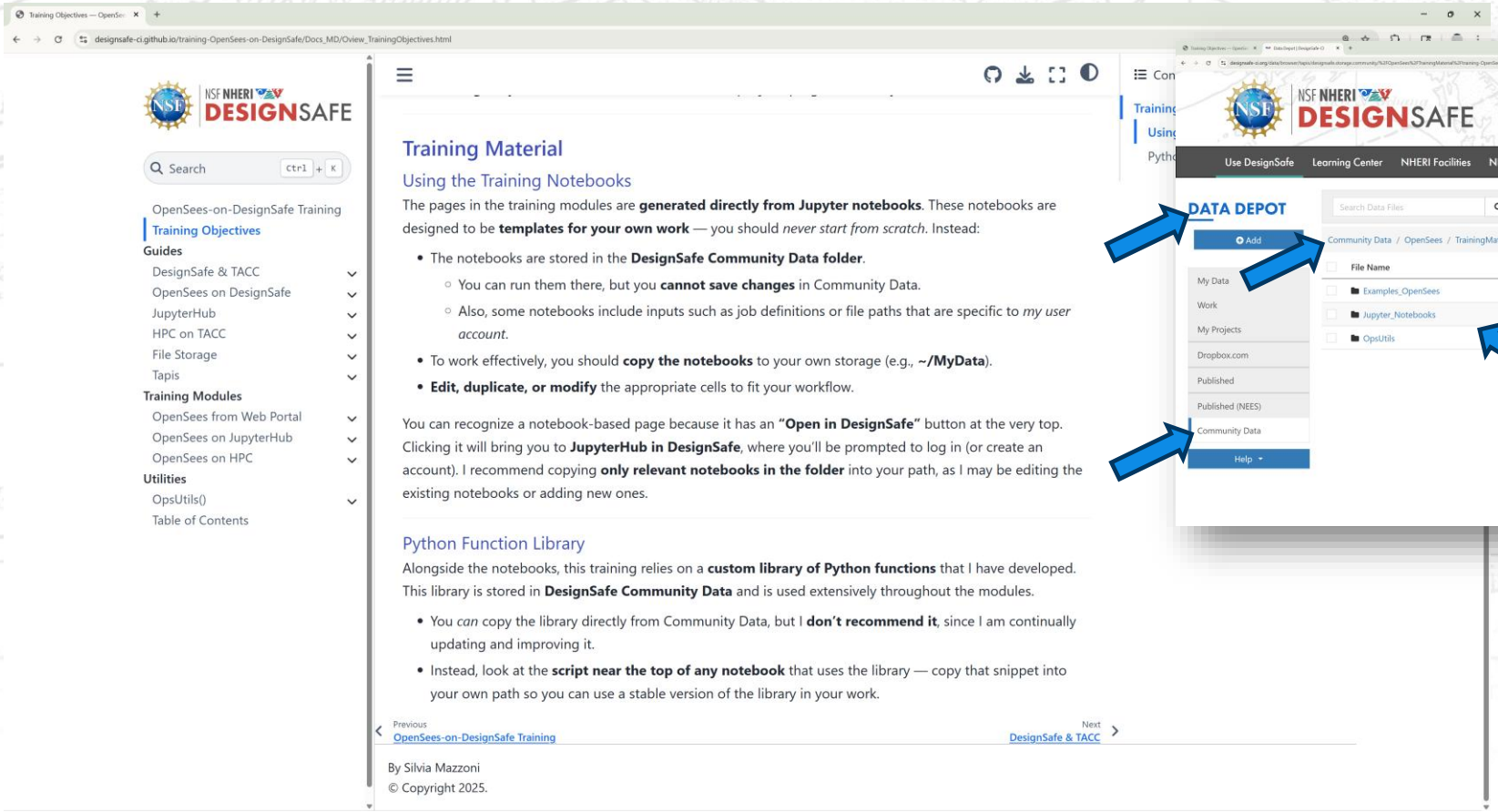
1. Submit to HPC from Web Portal via Tapis Apps
2. On Jupyter Hub
3. Submit to HPC from JupyterHub via Tapis Apps
4. SSH to HPC and submit Slurm Job

Ways To Run OpenSees on DesignSafe



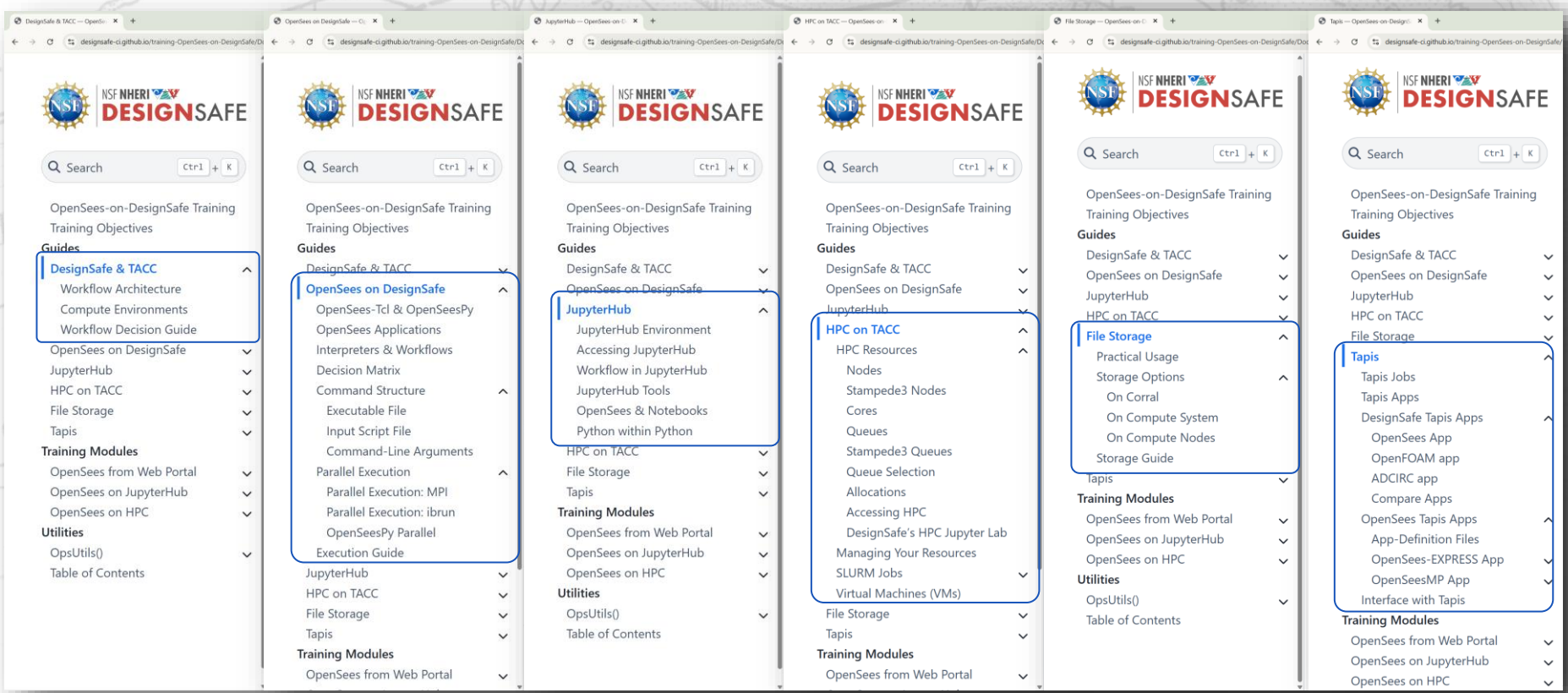

designsafe-ci.org/user-guide/training/
NSF NHERI DESIGNSAFE
USER GUIDE
designsafe-ci.org
Search docs
DesignSafe Essentials
Getting Started
Account Help
Training Resources
DesignSafe FAQ
How to Cite DesignSafe
DesignSafe Data Depot
Overview
Transferring Your Data
Steps to Curate and Publish Your Datasets
Best Practices for Data Curation and Publication
Data Dissemination and Impact
Resources for Users
About the Data Depot
Policies
Tools and Apps
Requesting New Applications
Popular
Simulation
Analysis
DesignSafe Essentials > Training Resources
Suggest an update via GitHub
TRAINING RESOURCES
DesignSafe-CI offers a suite of training resources aimed at equipping researchers and engineers with cutting-edge computational skills. These repositories cover a range of topics, including accelerating Python applications, utilizing database APIs, implementing physics-informed neural networks (PINNs), understanding explainable AI (XAI), and applying Deep Operator Networks (DeepONet). Each module provides hands-on tutorials and practical examples to enhance your proficiency in leveraging these technologies for natural hazards research.
OpenSees on DesignSafe
Learn to use OpenSees to develop, run, and scale structural & geotechnical simulations on DesignSafe/TACC. This hands-on module covers managing input files, launching parameter studies, and submitting and monitoring HPC jobs for reproducible workflows.
Database API training
Master database interaction through APIs using DesignSafe's infrastructure. Learn to programmatically access, query, and manipulate research data using RESTful APIs. Includes examples of data retrieval, filtering, and integration with analysis workflows.
Accelerating Python using Cython, Numba and JAX
Optimize your Python code performance using various acceleration techniques. Learn to use Cython for C-level performance, Numba for just-in-time compilation, and JAX for automatic differentiation and parallelization. Includes practical examples and performance comparisons.
Training on XAI using DesignSafe
Explore Explainable AI (XAI) techniques applied to natural hazards engineering. Learn how to interpret machine learning models using methods like LIME, SHAP, and feature importance analysis. The training includes practical examples using Python libraries and real-world datasets.
Physics Informed Machine Learning
Discover Physics-Informed Neural Networks (PINNs) for solving partial differential equations. This training demonstrates how to incorporate physical laws into neural networks, with applications in structural dynamics and fluid mechanics using TensorFlow.
DeepONet Training
Learn about Deep Operator Networks (DeepONet), a novel deep learning framework for learning operators. This training covers the fundamentals of DeepONet architecture, its applications in solving differential equations, and practical implementation using PyTorch.


designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/Oview_TrainingObjectives.html
NSF NHERI DESIGNSAFE
Search Ctrl + K
OpenSees-on-DesignSafe Training
Training Objectives
Guides
DesignSafe & TACC
OpenSees on DesignSafe
JupyterHub
HPC on TACC
File Storage
Tapis
Training Modules
OpenSees from Web Portal
OpenSees on JupyterHub
OpenSees on HPC
Utilities
OpsUtils()
Table of Contents
Contents
Training Material
Training Objectives
In this training module, you'll learn how to "run" OpenSees from different computational environments within DesignSafe — whether that's the Jupyter Hub, the Web Portal, Tapis, or direct HPC command-line access.
Beyond OpenSees-specific workflows, this training covers the core fundamentals necessary to build scalable and efficient scientific workflows that optimize the resources available through TACC and DesignSafe. We'll discuss how to:
Understand the hardware architecture at TACC, and how to best match job types to system capabilities.
Use the "software" infrastructure that supports computation at scale — including Tapis, the Tapis API, and task-specific Tapis Apps available on DesignSafe.
Navigate file storage systems on DesignSafe, including data staging, transfer, and archival strategies.
Interact with different computational environments — from terminal-based CLI to Jupyter notebooks and batch script submissions — and know when to use each.
By the end of this module, you'll know how to:
Use the DesignSafe Web Portal to submit OpenSees jobs without writing any batch scripts.
Launch OpenSees from within a Jupyter notebook, passing in variables or dynamically generating input files.
Run OpenSees directly from a terminal or Jupyter notebook, using both Tcl and Python scripts.
Submit OpenSees jobs to the HPC system on TACC via SLURM, providing input files and specifying resources like cores and wall time.
Automate large parameter studies by modifying script values on the fly or using loops in Python.
Leverage Tapis APIs to submit and monitor multiple jobs programmatically.
Training Material

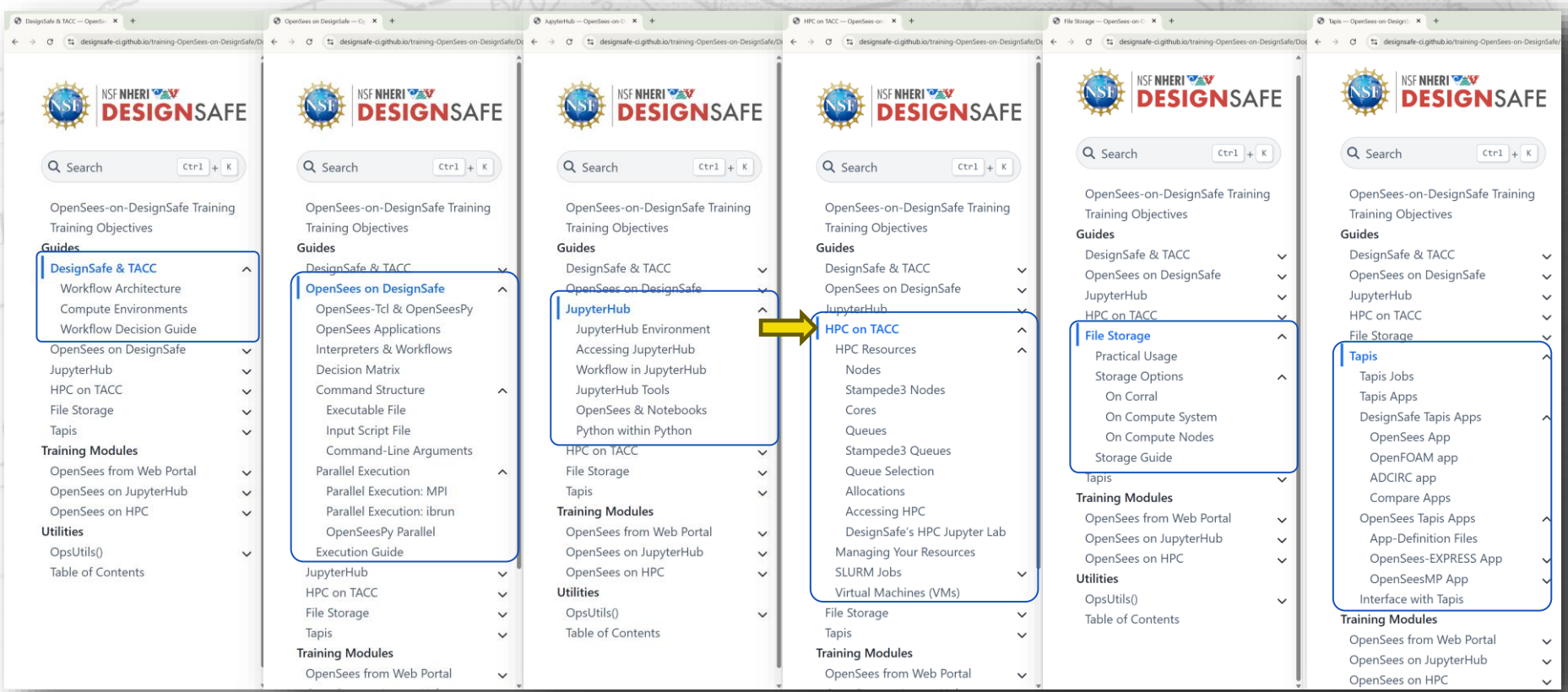

designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/Oview_TrainingObjectives.html
NSF NHERI DESIGNSAFE
Search Ctrl + K
OpenSees-on-DesignSafe Training
Training Objectives
Guides
DesignSafe & TACC
OpenSees on DesignSafe
JupyterHub
HPC on TACC
File Storage
Tapis
Training Modules
OpenSees from Web Portal
OpenSees on JupyterHub
OpenSees on HPC
Utilities
OpsUtils()
Table of Contents
Training Material
Using the Training Notebooks
The pages in the training modules are generated directly from Jupyter notebooks. These notebooks are designed to be templates for your own work — you should never start from scratch. Instead:
The notebooks are stored in the DesignSafe Community Data folder.
You can run them there, but you cannot save changes in Community Data.
Also, some notebooks include inputs such as job definitions or file paths that are specific to my user account.
To work effectively, you should copy the notebooks to your own storage (e.g., ~/MyData).
Edit, duplicate, or modify the appropriate cells to fit your workflow.
You can recognize a notebook-based page because it has an "Open in DesignSafe" button at the very top. Clicking it will bring you to JupyterHub in DesignSafe, where you'll be prompted to log in (or create an account). I recommend copying only relevant notebooks in the folder into your path, as I may be editing the existing notebooks or adding new ones.
Python Function Library
Alongside the notebooks, this training relies on a custom library of Python functions that I have developed. This library is stored in DesignSafe Community Data and is used extensively throughout the modules.
You can copy the library directly from Community Data, but I don't recommend it, since I am continually updating and improving it.
Instead, look at the script near the top of any notebook that uses the library — copy that snippet into your own path so you can use a stable version of the library in your work.
Previous
OpenSees-on-DesignSafe Training
Next
DesignSafe & TACC
By Silvia Mazzoni
© Copyright 2025.
NSF NHERI DESIGNSAFE
Use DesignSafe
Learning Center
NHERI Facilities
NHERI Community
News
Help
DATA DEPOT
Add
My Data
Work
My Projects
Dropbox.com
Published
Published (NEES)
Community Data
Help
Community Data / OpenSees / TrainingMaterial / training-OpenSees-on-DesignSafe
File Name
Examples_OpenSees
Jupyter_Notebooks
OpsUtils

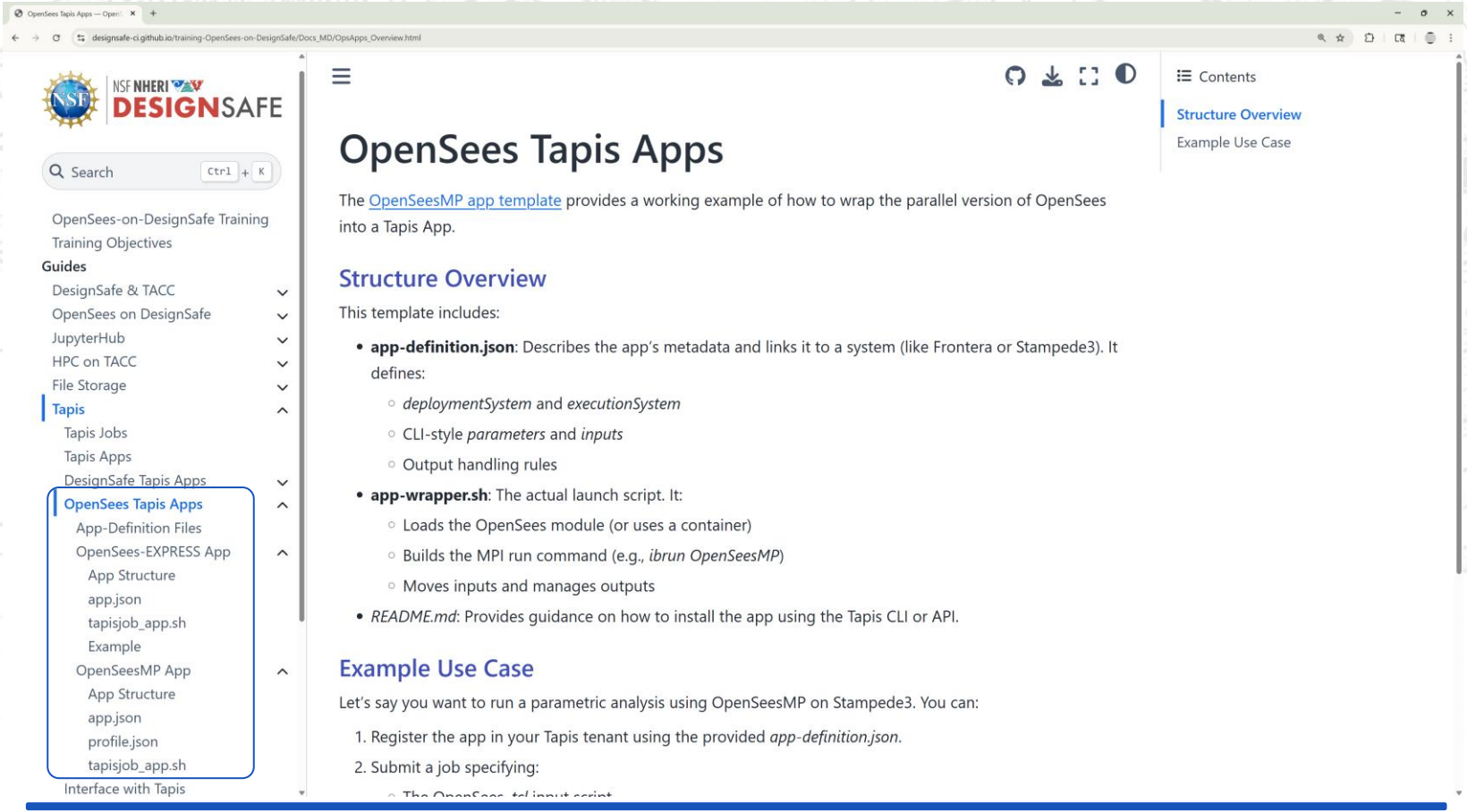
REFERENCE GUIDES

Everything you don't need to know, but really should!



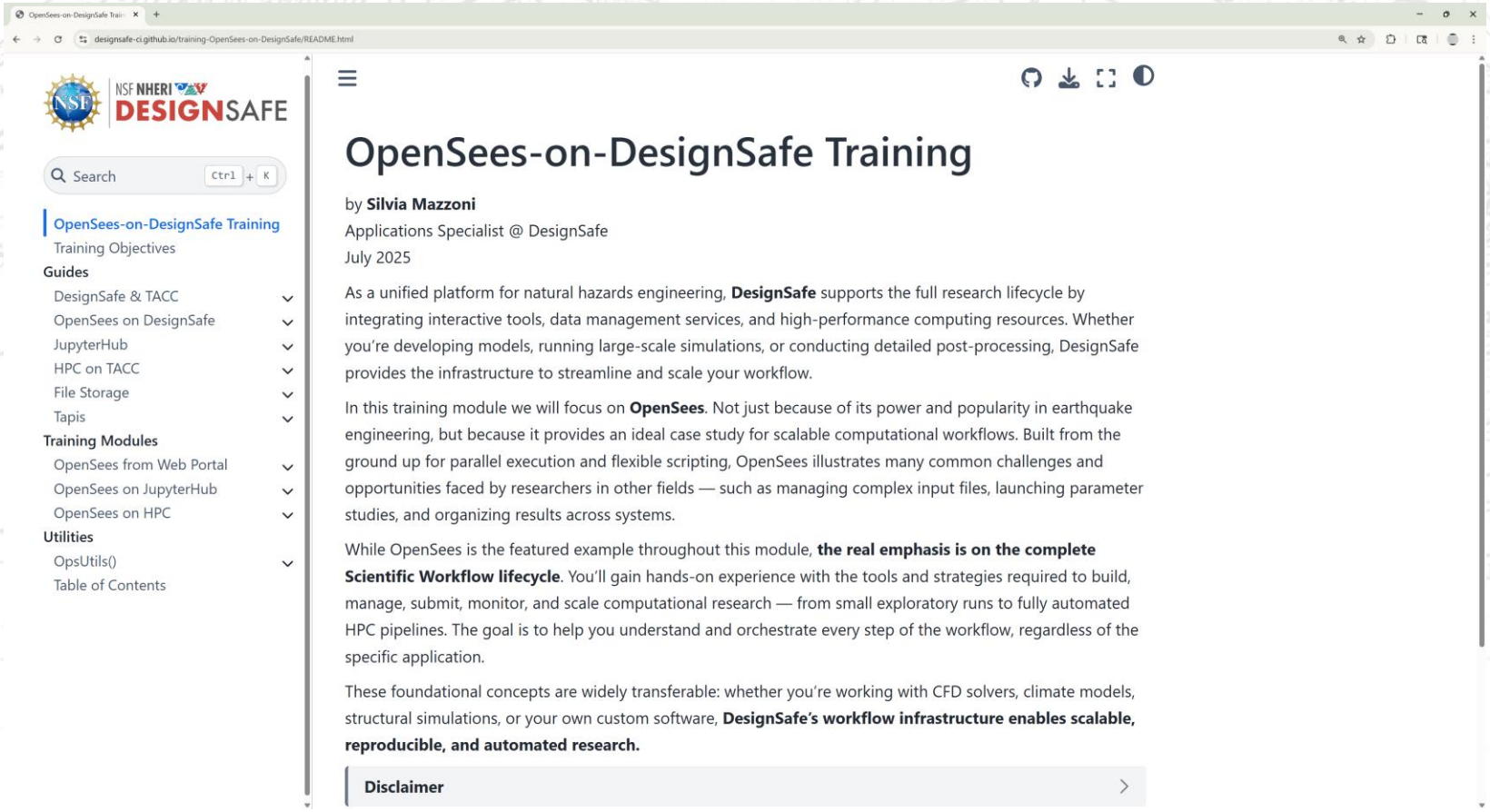
NSF NHERI DESIGNSAFE · TEXAS The University of Texas at Austin · UCLA · TACC · RICE · Florida Tech


NSF NHERI DESIGNSAFE
Search Ctrl + K
OpenSees-on-DesignSafe Training
Training Objectives
Guides
DesignSafe & TACC
Workflow Architecture
Compute Environments
Workflow Decision Guide
OpenSees on DesignSafe
OpenSees-Tcl & OpenSeesPy
OpenSees Applications
Interpreters & Workflows
Decision Matrix
Command Structure
Executable File
Input Script File
Command-Line Arguments
Parallel Execution
Parallel Execution: MPI
Parallel Execution: ibrun
OpenSeesPy Parallel
Execution Guide
JupyterHub
JupyterHub Environment
Accessing JupyterHub
Workflow in JupyterHub
JupyterHub Tools
OpenSees & Notebooks
Python within Python
HPC on TACC
HPC Resources
Nodes
Stampede3 Nodes
Cores
Queues
Stampede3 Queues
Queue Selection
Allocations
Accessing HPC
DesignSafe's HPC Jupyter Lab
Managing Your Resources
SLURM Jobs
Virtual Machines (VMs)
File Storage
Practical Usage
Storage Options
On Corral
On Compute System
On Compute Nodes
Storage Guide
Tapis
Tapis Jobs
Tapis Apps
DesignSafe Tapis Apps
OpenSees App
OpenFOAM app
ADCIRC app
Compare Apps
OpenSees Tapis Apps
App-Definition Files
OpenSees-EXPRESS App
OpenSeesMP App
Interface with Tapis
Training Modules
OpenSees from Web Portal
OpenSees on JupyterHub
OpenSees on HPC
Utilities
OpsUtils()
Table of Contents


designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/OpsApps_Overview.html
NSF NHERI DESIGNSAFE
Search Ctrl + K
OpenSees-on-DesignSafe Training
Training Objectives
Guides
DesignSafe & TACC
OpenSees on DesignSafe
JupyterHub
HPC on TACC
File Storage
Tapis
Tapis Jobs
Tapis Apps
DesignSafe Tapis Apps
OpenSees Tapis Apps
App-Definition Files
OpenSees-EXPRESS App
App Structure
app.json
tapisjob_app.sh
Example
OpenSeesMP App
App Structure
app.json
profile.json
tapisjob_app.sh
Interface with Tapis
Contents
Structure Overview
Example Use Case
OpenSees Tapis Apps
The OpenSeesMP app template provides a working example of how to wrap the parallel version of OpenSees into a Tapis App.
Structure Overview
This template includes:
app-definition.json: Describes the app's metadata and links it to a system (like Frontera or Stampede3). It defines:
deploymentSystem and executionSystem
CLI-style parameters and inputs
Output handling rules
app-wrapper.sh: The actual launch script. It:
Loads the OpenSees module (or uses a container)
Builds the MPI run command (e.g., ibrun OpenSeesMP)
Moves inputs and manages outputs
README.md: Provides guidance on how to install the app using the Tapis CLI or API.
Example Use Case
Let's say you want to run a parametric analysis using OpenSeesMP on Stampede3. You can:
1. Register the app in your Tapis tenant using the provided app-definition.json.
2. Submit a job specifying:

TRAINING MODULES

3 Ways to Run OpenSees



designsafe-ci.github.io/training-OpenSees-on-DesignSafe/README.html

NSF NHERI DESIGNSAFE

Search Ctrl + K

OpenSees-on-DesignSafe Training

by **Silvia Mazzoni**
Applications Specialist @ DesignSafe
July 2025

As a unified platform for natural hazards engineering, **DesignSafe** supports the full research lifecycle by integrating interactive tools, data management services, and high-performance computing resources. Whether you're developing models, running large-scale simulations, or conducting detailed post-processing, DesignSafe provides the infrastructure to streamline and scale your workflow.

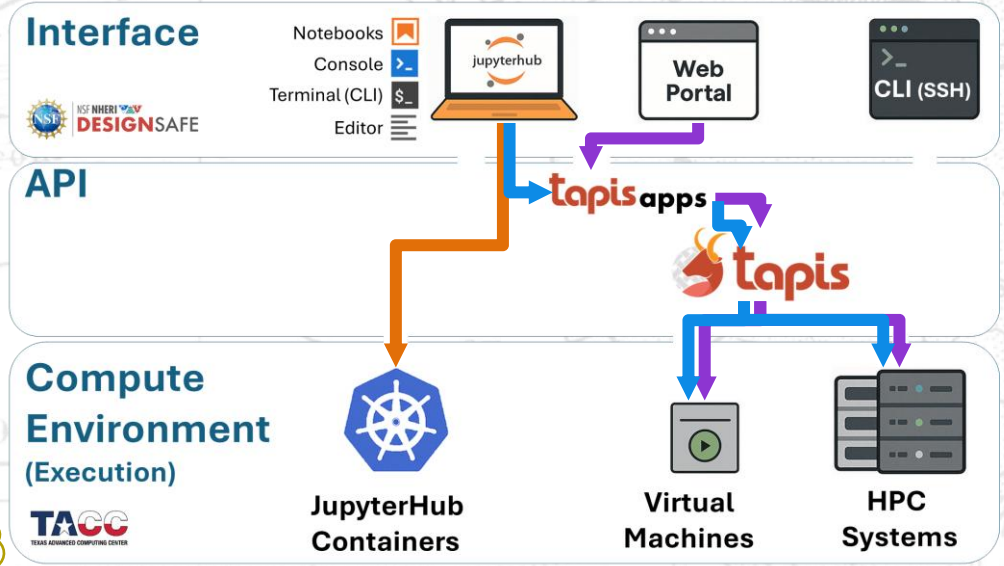
In this training module we will focus on **OpenSees**. Not just because of its power and popularity in earthquake engineering, but because it provides an ideal case study for scalable computational workflows. Built from the ground up for parallel execution and flexible scripting, OpenSees illustrates many common challenges and opportunities faced by researchers in other fields — such as managing complex input files, launching parameter studies, and organizing results across systems.

While OpenSees is the featured example throughout this module, **the real emphasis is on the complete Scientific Workflow lifecycle**. You'll gain hands-on experience with the tools and strategies required to build, manage, submit, monitor, and scale computational research — from small exploratory runs to fully automated HPC pipelines. The goal is to help you understand and orchestrate every step of the workflow, regardless of the specific application.

These foundational concepts are widely transferable: whether you're working with CFD solvers, climate models, structural simulations, or your own custom software, **DesignSafe's workflow infrastructure enables scalable, reproducible, and automated research.**

Disclaimer

Ways To Run OpenSees on DesignSafe



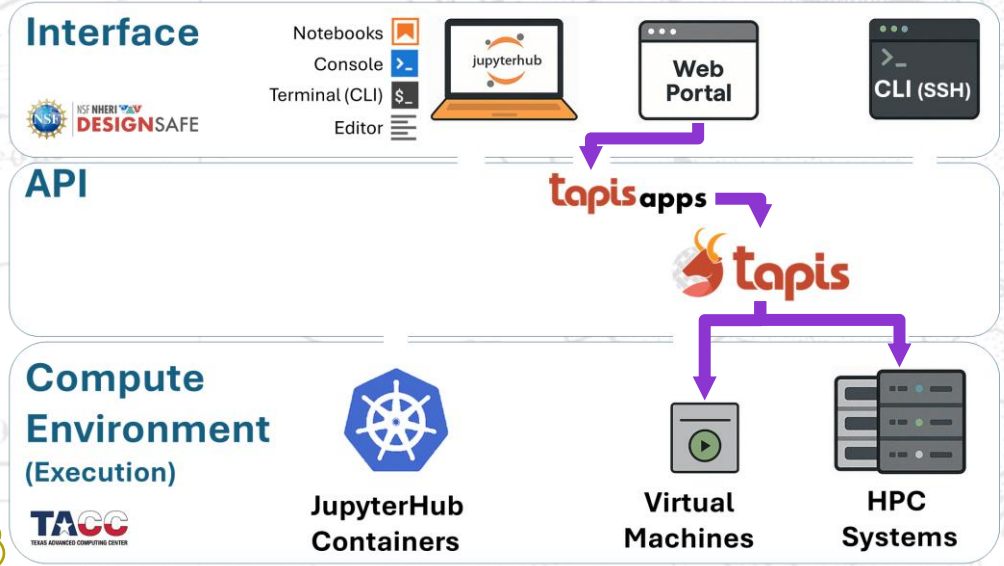
Interface Environments:
- Jupyter Hub
- Web Portal
- CLI (SSH)

API
- Tapis Apps
- Tapis

Compute Environments:
- Jupyter Hub
- Virtual Machines
- HPC Systems on TACC

Run OpenSees on DesignSafe
1. Submit to HPC from Web Portal via Tapis Apps
2. On Jupyter Hub
3. Submit to HPC from JupyterHub via Tapis Apps

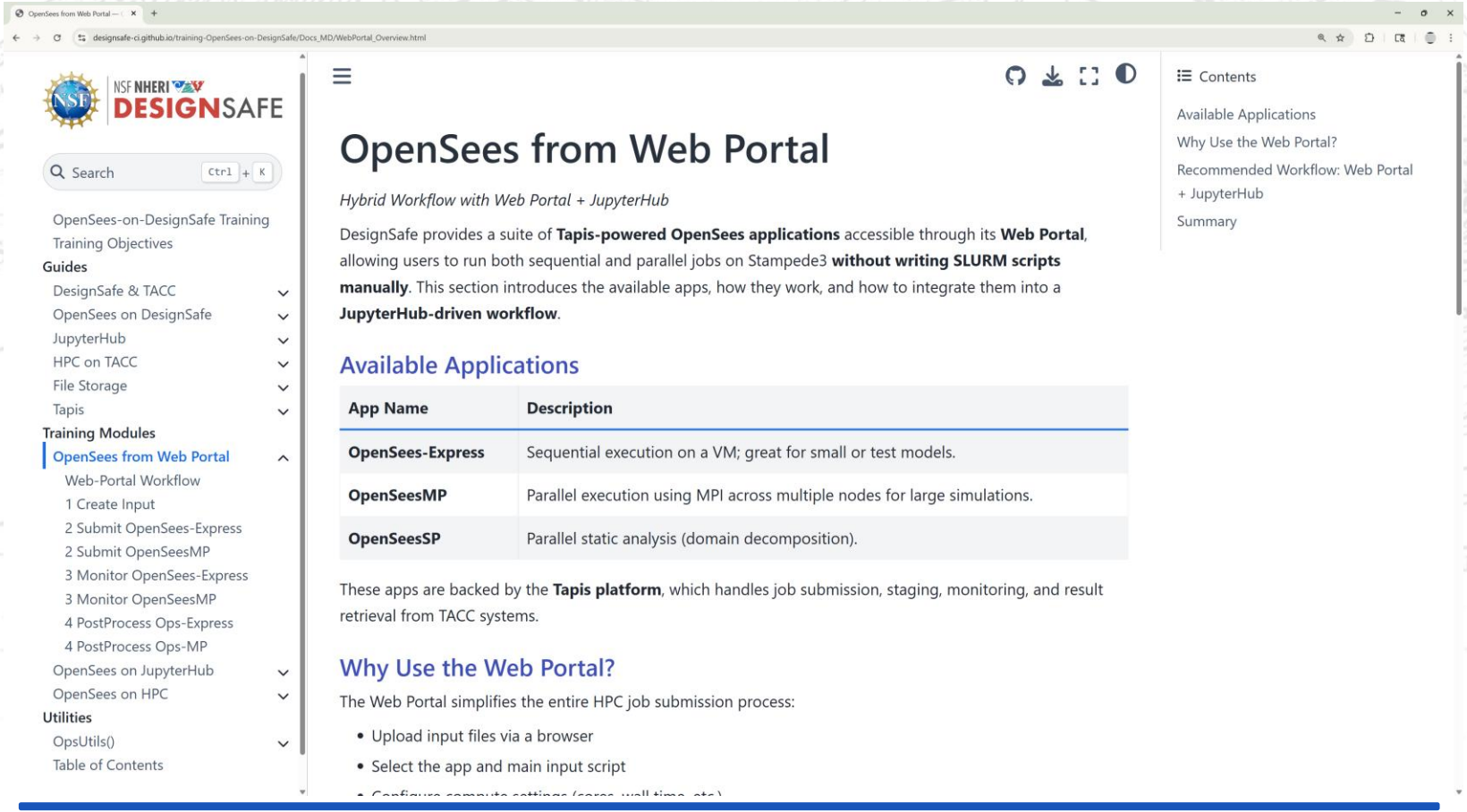
1. SUBMIT TO HPC FROM WEB PORTAL



Interface Environments:
- Jupyter Hub
- Web Portal
- CLI (SSH)

API
- Tapis Apps
- Tapis

Compute Environments:
- Jupyter Hub
- Virtual Machines
- HPC Systems on TACC



designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/WebPortal_Overview.html

NSF NHERI DESIGNSAFE

Search Ctrl + K

- OpenSees-on-DesignSafe Training
- Training Objectives

Guides

- DesignSafe & TACC
- OpenSees on DesignSafe
- JupyterHub
- HPC on TACC
- File Storage
- Tapis

Training Modules

- OpenSees from Web Portal
  - Web-Portal Workflow
  - 1 Create Input
  - 2 Submit OpenSees-Express
  - 2 Submit OpenSeesMP
  - 3 Monitor OpenSees-Express
  - 3 Monitor OpenSeesMP
  - 4 PostProcess Ops-Express
  - 4 PostProcess Ops-MP
- OpenSees on JupyterHub
- OpenSees on HPC

Utilities

- OpsUtils()
- Table of Contents

OpenSees from Web Portal

Hybrid Workflow with Web Portal + JupyterHub

DesignSafe provides a suite of **Tapis-powered OpenSees applications** accessible through its **Web Portal**, allowing users to run both sequential and parallel jobs on Stampede3 **without writing SLURM scripts manually**. This section introduces the available apps, how they work, and how to integrate them into a **JupyterHub-driven workflow**.

Available Applications

App Name	Description
OpenSees-Express	Sequential execution on a VM; great for small or test models.
OpenSeesMP	Parallel execution using MPI across multiple nodes for large simulations.
OpenSeesSP	Parallel static analysis (domain decomposition).

These apps are backed by the **Tapis platform**, which handles job submission, staging, monitoring, and result retrieval from TACC systems.

Why Use the Web Portal?

The Web Portal simplifies the entire HPC job submission process:

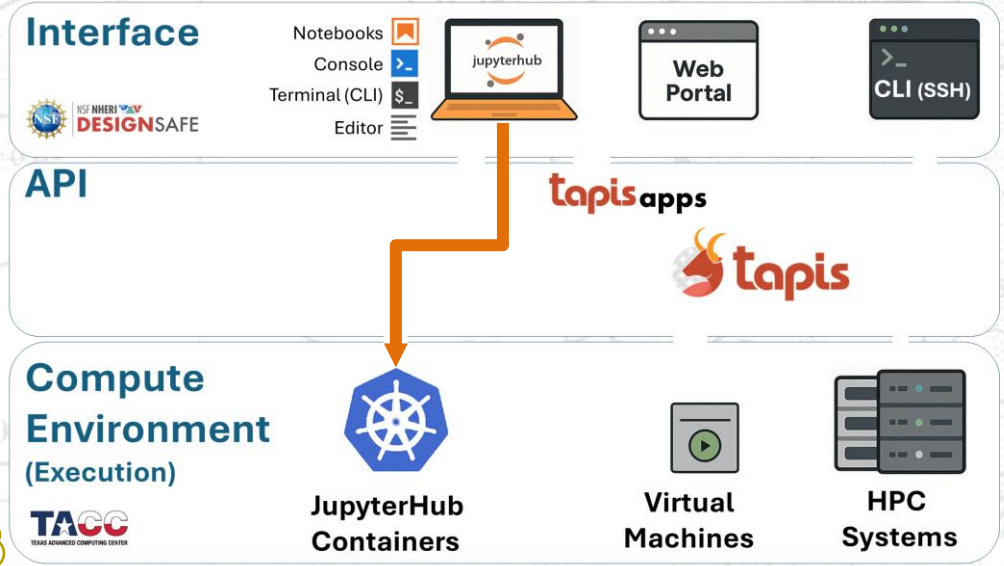
- Upload input files via a browser
- Select the app and main input script
- Configure compute settings (cores, wall time, etc.)

Contents

- Available Applications
- Why Use the Web Portal?
- Recommended Workflow: Web Portal + JupyterHub
- Summary

LIVE DEMO

2. RUN ON JUPYTERHUB



Interface Environments:

- Jupyter Hub
- Web Portal
- CLI (SSH)

API:

- Tapis Apps
- Tapis

Compute Environments:

- Jupyter Hub
- Virtual Machines
- HPC Systems on TACC

OpenSees on JupyterHub

The Primary Interface for Scripting, Testing, and Orchestrating Your Workflow

The **DesignSafe JupyterHub** is your command center for running OpenSees in all its forms—whether you're writing Tcl scripts for the traditional interpreter or Python scripts using OpenSeesPy. It is the **primary portal for scripting, visualization, data management, and HPC job orchestration**.

This environment is ideal for:

- Interactive script development
- Pre- and post-processing of model data
- Exploratory analysis and documentation
- Submitting HPC jobs directly via the Tapis API

Each JupyterHub session runs in a dedicated container on a TACC-managed Kube**CPU cores and 20 GB of RAM**—perfect for medium-scale simulations and full-fe workflows.

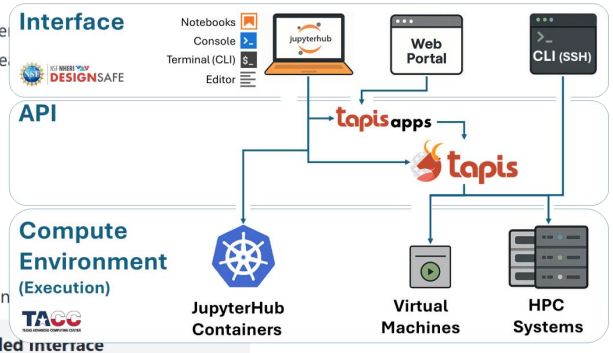
All OpenSees have been pre-installed and are available in JupyterHub:

- Tcl: OpenSees, OpenSeesMP, and OpenSeesSP
- Pythong: OpenSeesPy

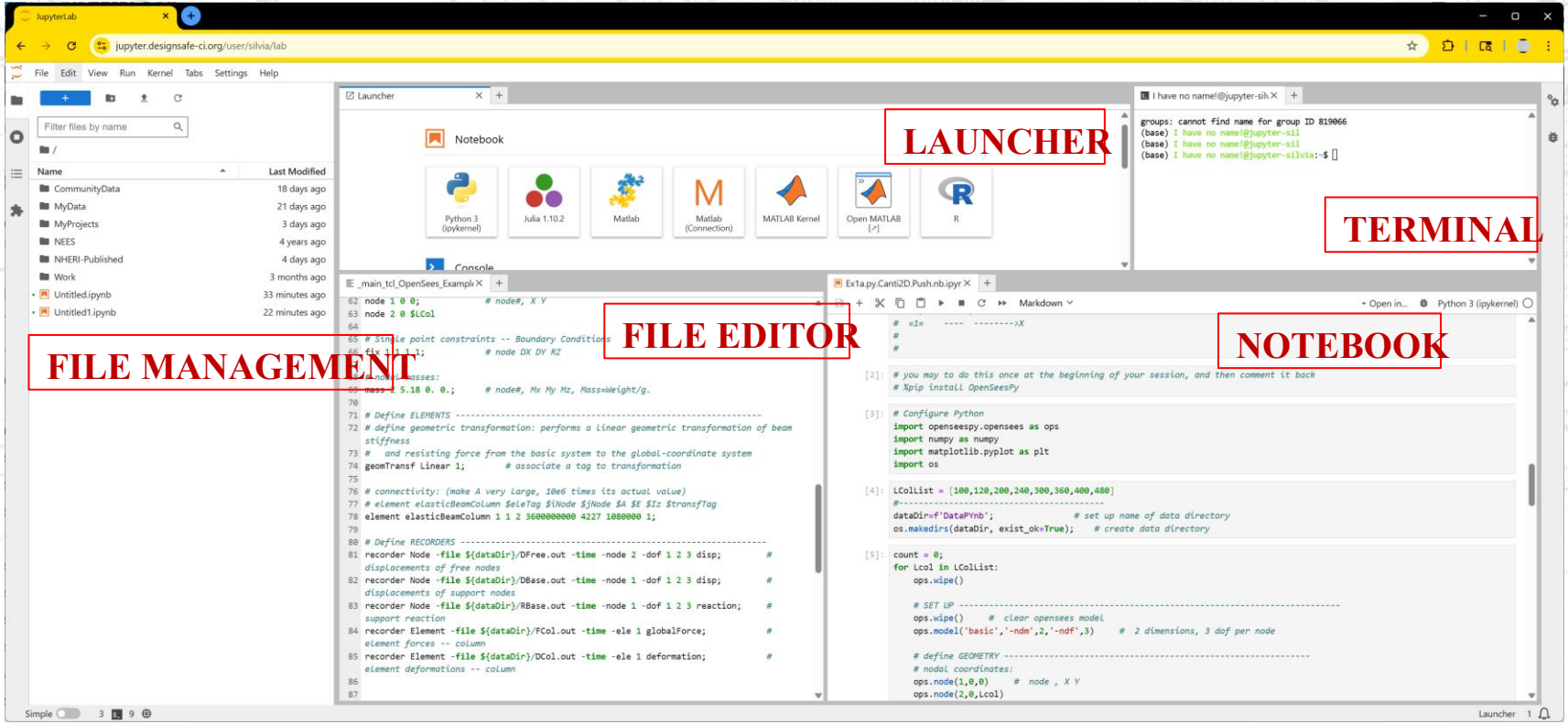
Choosing the Right Interface for Your Task

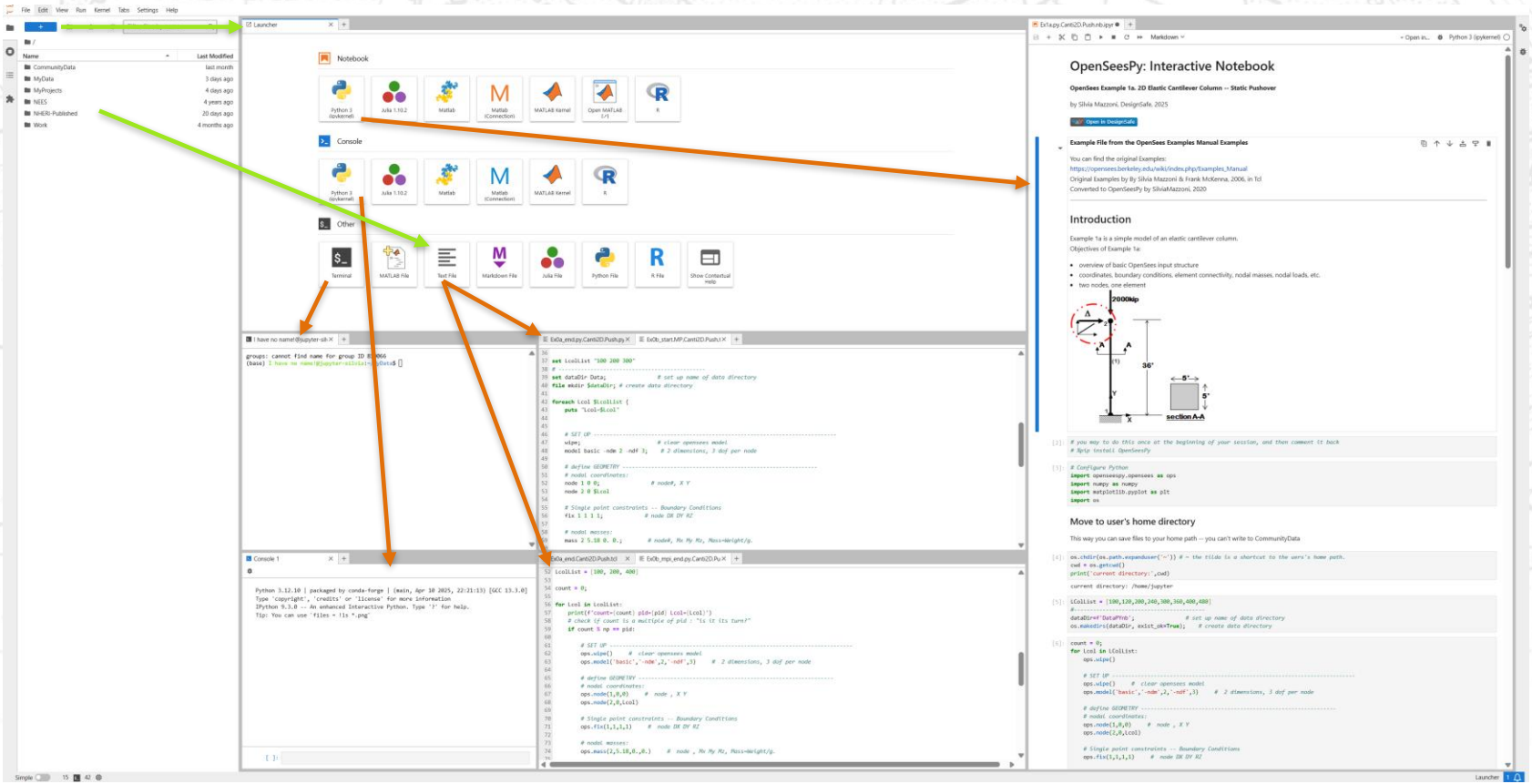
JupyterHub offers a suite of interfaces tailored to different stages of your modelin

Task	Recommended Interface



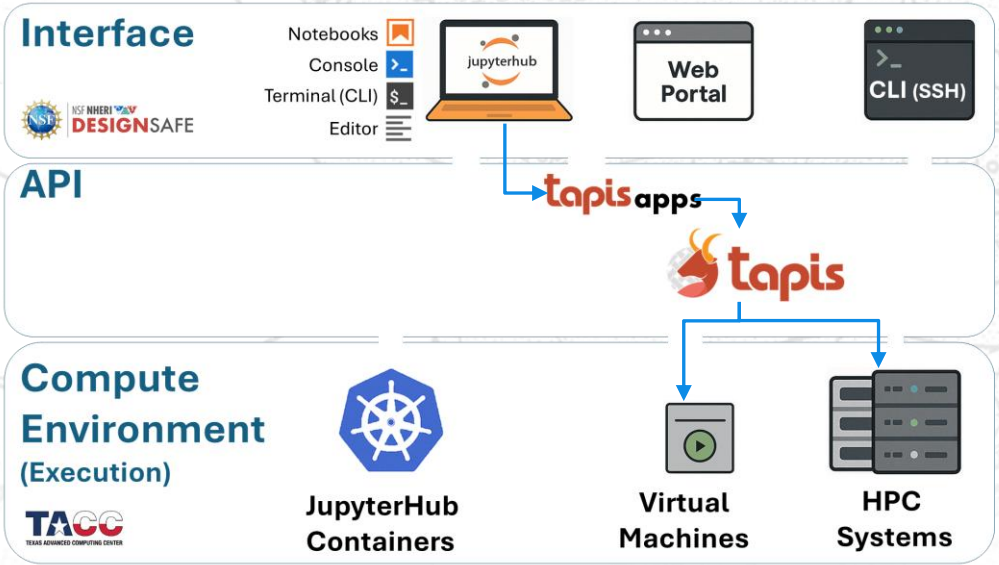
JUPYTER IS THE BEST! IDE



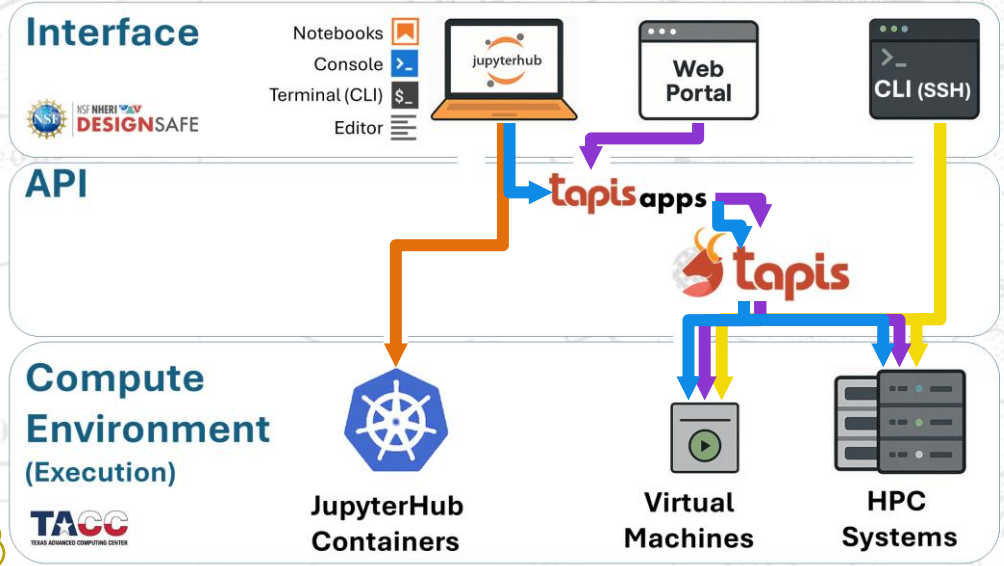

Launcher
Notebook
Console
Other
Terminal
Text File
Markdown File
Python File
R File
OpenSeesPy: Interactive Notebook
Introduction
Move to user's home directory
2000kip
36'
5'
section A-A

LIVE DEMO

3. SUBMIT TO HPC FROM JUPYTERHUB



3. SUBMIT TO HPC FROM JUPYTERHUB



Interface Environments:

- Jupyter Hub
- Web Portal
- CLI (SSH)

API:

- Tapis Apps
- Tapis

Compute Environments:

- Jupyter Hub
- Virtual Machines
- HPC Systems on TACC

OpenSees on HPC

OpenSees on HPC via JupyterHub

DesignSafe's JupyterHub provides an interactive environment where you can prepare, submit, and monitor jobs directly on TACC's HPC systems. Instead of writing your own SLURM scripts, you can use **Tapis Apps**—predefined job templates that take care of loading the correct modules, staging input files, running OpenSees on HPC nodes, and saving results back to your storage.

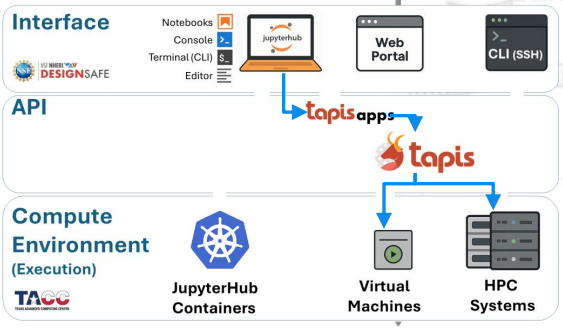
Running OpenSees this way is especially useful when you want to combine the flexibility of Jupyter notebooks with the power of HPC. You can test or pre-process models interactively, then submit larger production runs to Stampede3 without leaving JupyterHub.

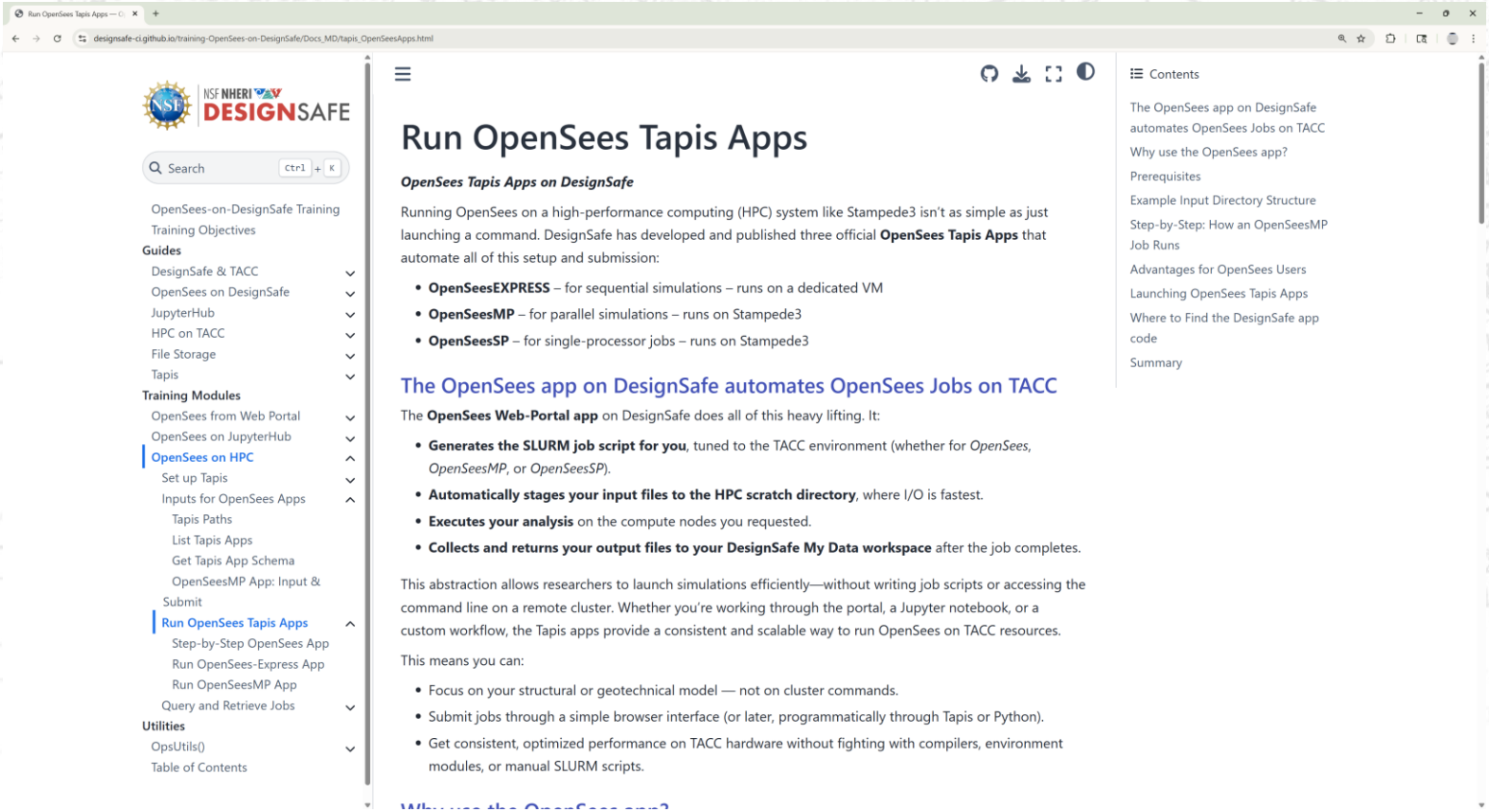
When comparing the available workflows:

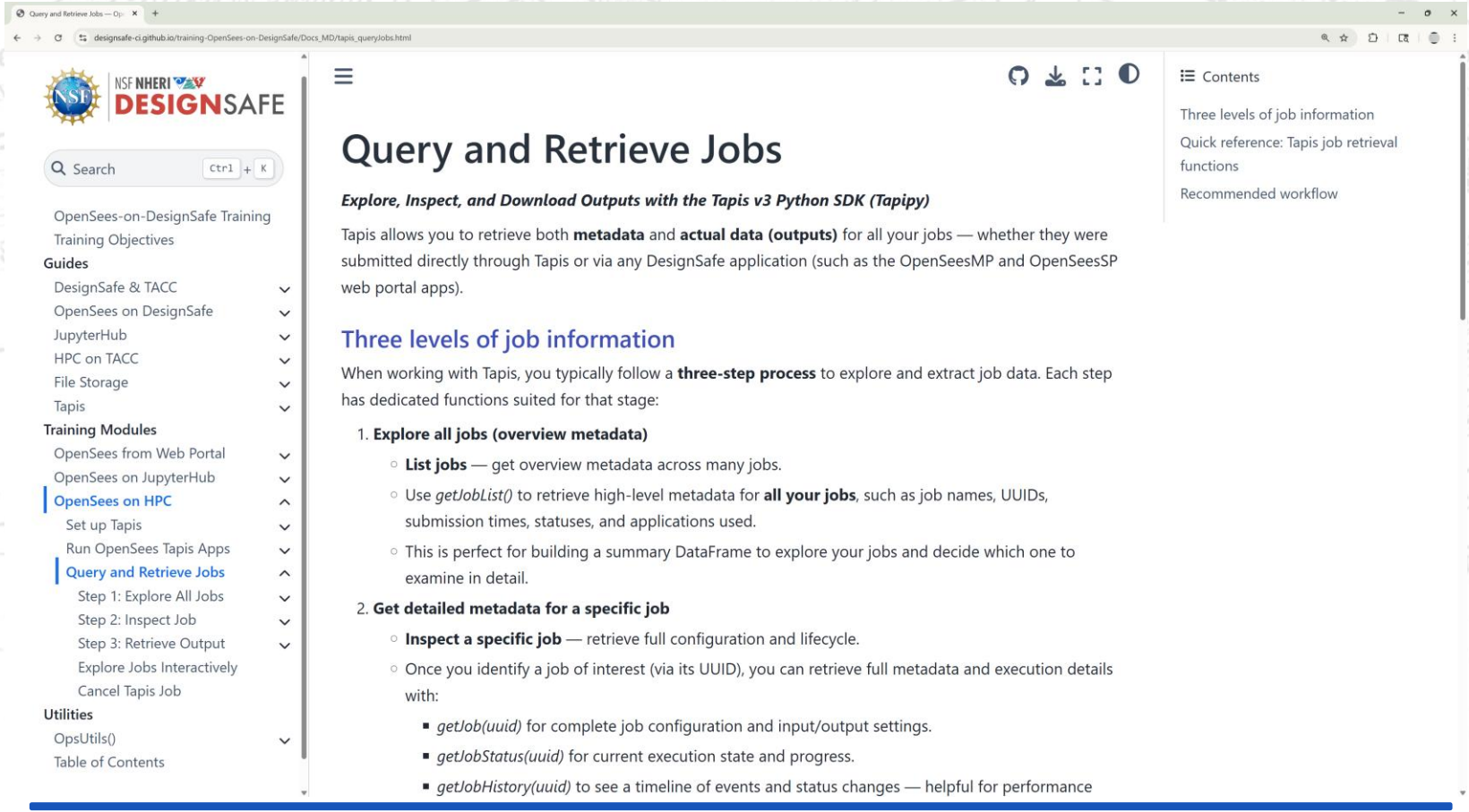
- **Web Portal → HPC** Easiest to use with a point-and-click interface, but less flexible for automation or scripting.
- **JupyterHub → Local** Good for testing small models interactively, but limited to the resources available in JupyterHub (single node).
- **SSH → HPC (manual SLURM)** The traditional method: log into Stampede3 via SSH, write your own SLURM job script, and submit with `sbatch`. In addition, you must manually move your input files to the correct **scratch location** before submitting the job, and copy the results back afterward. This provides **maximum control**, but requires more setup and becomes tedious when you need to run many jobs.
  - **Note:** Because Tapis makes this process much more fluid, we will not be covering the SSH method in detail in this training.
- **JupyterHub → HPC via Tapis Apps** Combines the best of both worlds: scripting and automation from JupyterHub with the full scalability of HPC. Tapis Apps handle SLURM submission and also take care of staging files in and out of scratch for you, making it especially convenient for large studies or repeated runs.

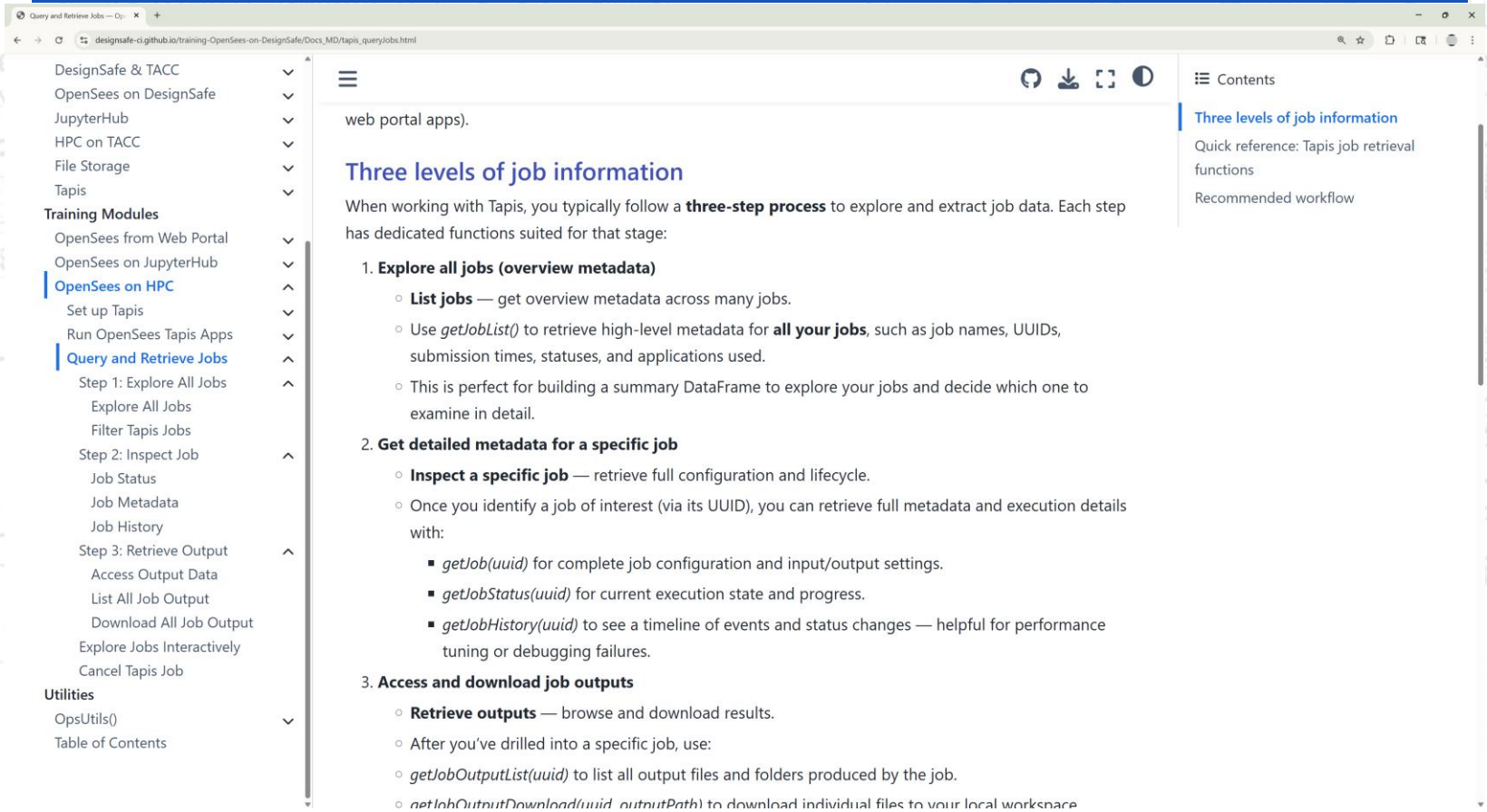
Step-by-step workflow (JupyterHub + Tapis Apps):

1. Prepare your OpenSees input files in your DesignSafe storage — you can do this within JupyterHub.




designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/tapis_OpenSeesApps.html
NSF NHERI DESIGNSAFE
Search Ctrl + K
OpenSees-on-DesignSafe Training
Training Objectives
Guides
DesignSafe & TACC
OpenSees on DesignSafe
JupyterHub
HPC on TACC
File Storage
Tapis
Training Modules
OpenSees from Web Portal
OpenSees on JupyterHub
OpenSees on HPC
Set up Tapis
Inputs for OpenSees Apps
Tapis Paths
List Tapis Apps
Get Tapis App Schema
OpenSeesMP App: Input & Submit
Run OpenSees Tapis Apps
Step-by-Step OpenSees App
Run OpenSees-Express App
Run OpenSeesMP App
Query and Retrieve Jobs
Utilities
OpsUtils()
Table of Contents
Run OpenSees Tapis Apps
OpenSees Tapis Apps on DesignSafe
Running OpenSees on a high-performance computing (HPC) system like Stampede3 isn't as simple as just launching a command. DesignSafe has developed and published three official OpenSees Tapis Apps that automate all of this setup and submission:
OpenSeesEXPRESS – for sequential simulations – runs on a dedicated VM
OpenSeesMP – for parallel simulations – runs on Stampede3
OpenSeesSP – for single-processor jobs – runs on Stampede3
The OpenSees app on DesignSafe automates OpenSees Jobs on TACC
The OpenSees Web-Portal app on DesignSafe does all of this heavy lifting. It:
Generates the SLURM job script for you, tuned to the TACC environment (whether for OpenSees, OpenSeesMP, or OpenSeesSP).
Automatically stages your input files to the HPC scratch directory, where I/O is fastest.
Executes your analysis on the compute nodes you requested.
Collects and returns your output files to your DesignSafe My Data workspace after the job completes.
This abstraction allows researchers to launch simulations efficiently—without writing job scripts or accessing the command line on a remote cluster. Whether you're working through the portal, a Jupyter notebook, or a custom workflow, the Tapis apps provide a consistent and scalable way to run OpenSees on TACC resources.
This means you can:
Focus on your structural or geotechnical model — not on cluster commands.
Submit jobs through a simple browser interface (or later, programmatically through Tapis or Python).
Get consistent, optimized performance on TACC hardware without fighting with compilers, environment modules, or manual SLURM scripts.
Contents
The OpenSees app on DesignSafe automates OpenSees Jobs on TACC
Why use the OpenSees app?
Prerequisites
Example Input Directory Structure
Step-by-Step: How an OpenSeesMP Job Runs
Advantages for OpenSees Users
Launching OpenSees Tapis Apps
Where to Find the DesignSafe app code
Summary


designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/tapis_queryJobs.html
NSF NHERI DESIGNSAFE
Search Ctrl + K
OpenSees-on-DesignSafe Training
Training Objectives
Guides
DesignSafe & TACC
OpenSees on DesignSafe
JupyterHub
HPC on TACC
File Storage
Tapis
Training Modules
OpenSees from Web Portal
OpenSees on JupyterHub
OpenSees on HPC
Set up Tapis
Run OpenSees Tapis Apps
Query and Retrieve Jobs
Step 1: Explore All Jobs
Step 2: Inspect Job
Step 3: Retrieve Output
Explore Jobs Interactively
Cancel Tapis Job
Utilities
OpsUtils()
Table of Contents
Contents
Three levels of job information
Quick reference: Tapis job retrieval functions
Recommended workflow
Query and Retrieve Jobs
Explore, Inspect, and Download Outputs with the Tapis v3 Python SDK (Tapipy)
Tapis allows you to retrieve both metadata and actual data (outputs) for all your jobs — whether they were submitted directly through Tapis or via any DesignSafe application (such as the OpenSeesMP and OpenSeesSP web portal apps).
Three levels of job information
When working with Tapis, you typically follow a three-step process to explore and extract job data. Each step has dedicated functions suited for that stage:
1. Explore all jobs (overview metadata)
List jobs — get overview metadata across many jobs.
Use getJobList() to retrieve high-level metadata for all your jobs, such as job names, UUIDs, submission times, statuses, and applications used.
This is perfect for building a summary DataFrame to explore your jobs and decide which one to examine in detail.
2. Get detailed metadata for a specific job
Inspect a specific job — retrieve full configuration and lifecycle.
Once you identify a job of interest (via its UUID), you can retrieve full metadata and execution details with:
getJob(uuid) for complete job configuration and input/output settings.
getJobStatus(uuid) for current execution state and progress.
getJobHistory(uuid) to see a timeline of events and status changes — helpful for performance


designsafe-ci.github.io/training-OpenSees-on-DesignSafe/Docs_MD/tapis_queryJobs.html
DesignSafe & TACC
OpenSees on DesignSafe
JupyterHub
HPC on TACC
File Storage
Tapis
Training Modules
OpenSees from Web Portal
OpenSees on JupyterHub
OpenSees on HPC
Set up Tapis
Run OpenSees Tapis Apps
Query and Retrieve Jobs
Step 1: Explore All Jobs
Explore All Jobs
Filter Tapis Jobs
Step 2: Inspect Job
Job Status
Job Metadata
Job History
Step 3: Retrieve Output
Access Output Data
List All Job Output
Download All Job Output
Explore Jobs Interactively
Cancel Tapis Job
Utilities
OpsUtils()
Table of Contents
web portal apps).
Three levels of job information
When working with Tapis, you typically follow a three-step process to explore and extract job data. Each step has dedicated functions suited for that stage:
1. Explore all jobs (overview metadata)
List jobs — get overview metadata across many jobs.
Use getJobList() to retrieve high-level metadata for all your jobs, such as job names, UUIDs, submission times, statuses, and applications used.
This is perfect for building a summary DataFrame to explore your jobs and decide which one to examine in detail.
2. Get detailed metadata for a specific job
Inspect a specific job — retrieve full configuration and lifecycle.
Once you identify a job of interest (via its UUID), you can retrieve full metadata and execution details with:
getJob(uuid) for complete job configuration and input/output settings.
getJobStatus(uuid) for current execution state and progress.
getJobHistory(uuid) to see a timeline of events and status changes — helpful for performance tuning or debugging failures.
3. Access and download job outputs
Retrieve outputs — browse and download results.
After you've drilled into a specific job, use:
getJobOutputList(uuid) to list all output files and folders produced by the job.
getJobOutputDownload(uuid, outputPath) to download individual files to your local workspace
Contents
Three levels of job information
Quick reference: Tapis job retrieval functions
Recommended workflow

LIVE DEMO

USER RESOURCES ON DESIGNSAFE

- User Guides
- Training Modules
- Webinars
- Running-OpenSees-on-DesignSafe Slack Channel
- OpenSees Slack Channel
- Submit a ticket for issues
- Message me on slack. (I prefer the channel so others can participate)

SUMMARY & QUESTIONS

DesignSafe provides the computational environment to integrate your OpenSees analyses into your complete workflow

Recommended Getting Started:

1. With a MWE – minimum working example
2. Test the MWE in Jupyter Hub
3. Personalize and test your script in JupyterHub
4. Submit your modified script to the Web Portal
5. Use Tapis to query your Job Metadata
6. Access your analysis results in Jupyter Hub
7. Submit the SAME script to HPC via Tapis