



2020 Programming Bootcamp

Tapis from the Command Line Using **tapis-cli**

Frank McKenna

UC Berkeley



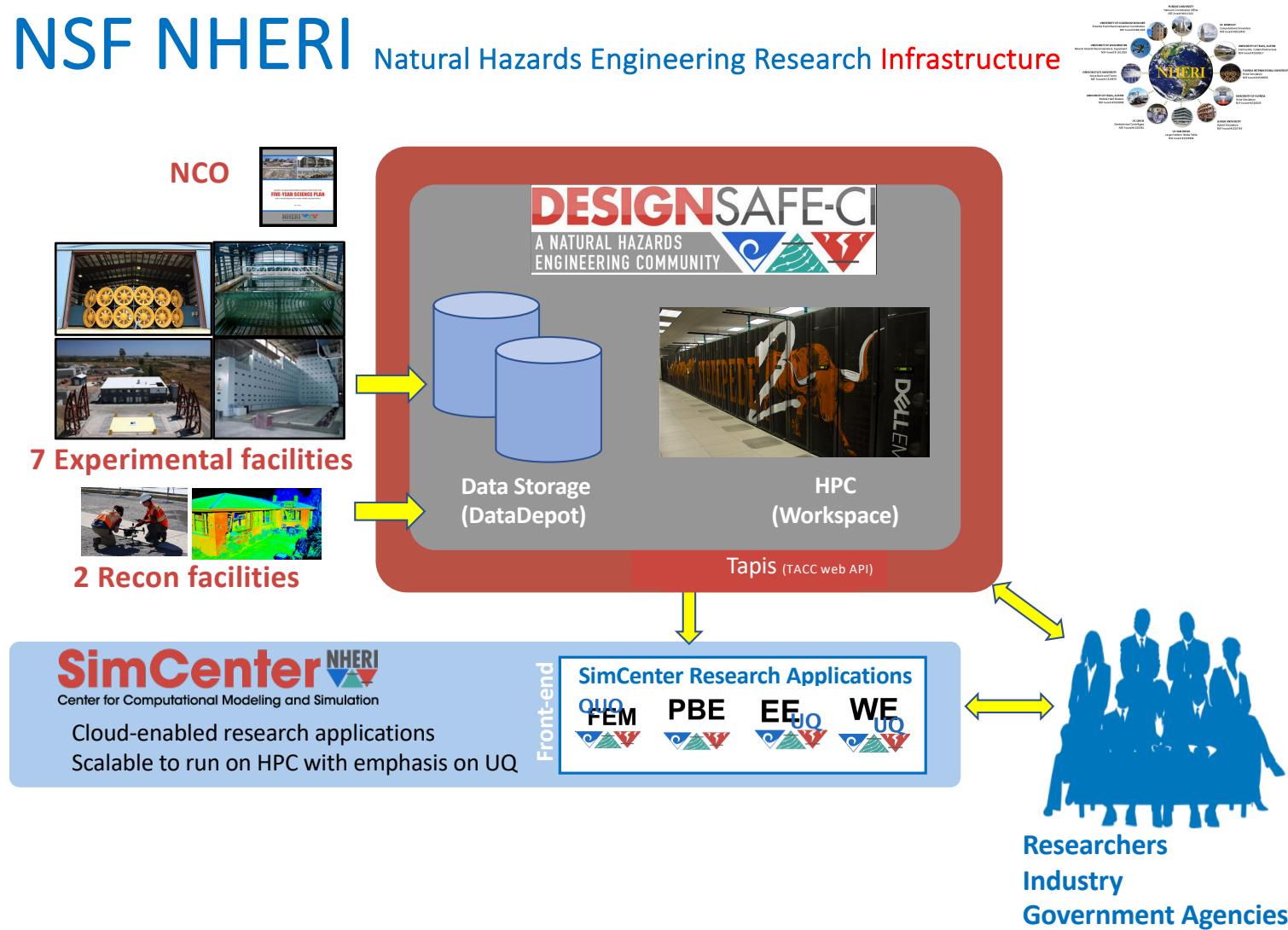
NSF award: CMMI 1612843

Anything in grey box is something to type
in a terminal/cmd/powershell window

```
pip install tapis-cli
```

NSF NHERI

Natural Hazards Engineering Research Infrastructure



TAPIS is a cloud service that provides access to distributed HPC and Storage resources through a RESTful API

The screenshot shows a web browser displaying the TAPIS documentation at tacc-cloud.readthedocs.io/projects/agave/en/latest/agave/introduction/introduction.html. The page title is "Introduction". The left sidebar contains a navigation menu with sections: "Introduction", "GUIDES", "SYSTEMS", and "FILES". Under "Introduction", there are links to "Conventions", "Customizing Responses", "Status Codes", "Best Practices", and "Tutorials". Under "GUIDES", there are links to "Guides" and "Authorization". Under "SYSTEMS", there are links to "Clients and API Keys", "Systems", "Storage systems", "Execution Systems", "Disabling", "System roles", "System scope", "Private systems", and "Default systems". Under "FILES", there are links to "Files", "Transferring data", and "Basic data operations". The main content area starts with a heading "Introduction" followed by a paragraph about the Tapis Platform being an open source, science-as-a-service API platform for powering your digital lab. It lists four bullet points: "Run code", "Manage data", "Collaborate meaningfully", and "Integrate anywhere". Below this is another paragraph about the documentation site containing examples to help build a digital lab. At the bottom, there are copyright information ("© Copyright 2018, Texas Advanced Computing Center, University of Texas, Austin Revision 6ad0da6c."), a note about being built with Sphinx, and links for "Previous" and "Next".

Up until today that Interaction has been through your Browser

The screenshot shows two side-by-side browser windows of the DesignSafe-CI platform.

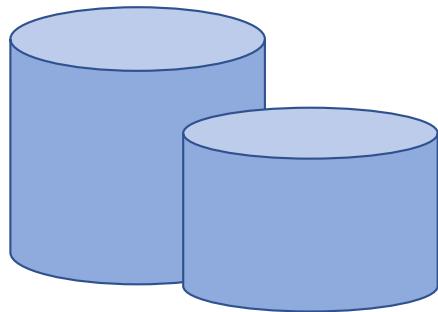
Left Window (Data Depot):

- Header: NSF, DESIGNSAFE-CI, NHERI: NATURAL HAZARDS ENGINEERING RESEARCH INFRASTRUCTURE.
- Top navigation: Research Workbench, Learning Center, NHERI Facilities, NHERI Community.
- Left sidebar: DATA DEPOT, + Add, My Data, My Projects, Shared with Me, Box.com, Dropbox.com, Google Drive, Published, Published (NEES), Community Data, Help.
- Main content: A file browser for workspace tg457427. It lists several folders: .Trash, aloetest, AloeTest1, applications, archive, blah, dir1, EE-UQ, example, ExampleScripts, and ExampleScripts.tar.gz. A checkbox for 'Name' is checked.

Right Window (Workspace):

- Header: Welcome, Frank! (with a notification icon showing 0).
- Top navigation: Research Workbench, Learning Center, NHERI Facilities, NHERI Community, About, Help.
- Search bar: Search DesignSafe.
- Central content:
 - WORKSPACE** section: Shows tabs for Simulation [8], Visualization [9], Data Processing [2], Partner Data Apps [5], Utilities [2], and My Apps [2].
 - DATA DEPOT BROWSER** section: Select data source dropdown set to 'My Data'. Browsing path: tg457427. File name dropdown set to 'Trash'. Below it is a list of folder icons: Trash, aloetest, AloeTest1, applications, archive, blah, and dir1.
 - OpenSees User Guide** section: Describes OpenSees as a software framework for simulating static and seismic response of structural and geotechnical systems. It mentions sequential (OpenSees EXPRESS) and parallel interpreters (OpenSeesSP and OpenSeesMP). A dropdown menu is set to 'OpenSeesMP (V 3.2)'.
 - RUN OPENSEESMP (V 3.2)** section: ver. 3.2.0. Text: OpenSeesMP is an OpenSees interpreter intended for high performance computers for performing finite element simulations with parametric studies and very large models on parallel machines. OpenSeesMP requires understanding of parallel processing and the capabilities to write parallel scripts. OpenSeesMP runs on up to 12 KNL Nodes on Stampede2, with 64 cores per Node.
 - Inputs** section: Input Directory dropdown set to 'Select Click to select input data'.

But there are other ways to utilize resources



And they allow you to make much wider
use of TACC (and others) resources

tapis-cli is one alternative

tapis-cli is a human-friendly, scriptable command line interface, implemented in Python

tapis-cli is installed from the command line using pip:

```
pip install tapis-cli
```

https://www.designsafe-ci.org/media/filer_public/90/d5/90d5ff98-3ca1-40a5-a2cb-2ead8f51ecb9/tapis-cli-how-to-guide-readthedocs-io-en-latest.pdf

tapis-cli auth setup

tapis auth init

- For tenant name enter designsafe
- For username your tacc login
- For password your tacc password.
- For rest of prompts just hit enter (Container Registry, git server)

tapis -h

Systems

Files

Apps

Jobs

tapis systems are our hardware resources

systems

Storage

Execution

tapis systems search --default eq true -f json

```
[  
  {  
    "id": "designsafe.storage.default",  
    "name": "DesignSafe Default Storage Host",  
    "type": "STORAGE",  
    "default": true  
  }  
]
```

--default
-f json

tapis systems -h

```
systems create
systems default set
systems default unset
systems disable
systems enable
systems history
systems list
systems publish
systems queues list
systems roles drop
systems roles grant
systems roles list
systems roles revoke
systems roles show
systems search
systems show
systems status
systems unpublish
systems update
```

systems

We will **create** and
possibly **update** later

Why not now (because it may be tricky)

1. In code/agave is a file `fronteraSystem.json`
edit it replacing following:

`${USERNAME}`

`${SCRATCH_DIR}`

`${PASSWORD}`

`${ALLOCATION}`

Courtesy: Sal Tijerina, DesignSafe-ci

`ssh frontera.tacc.utexas.edu
cds
pwd`

-A FTA-DD-SimCenter

```
tapis systems create -F fronteraSystem.json
```

Systems

DEMO to demonstrate how to Create
an EXECUTION system.

files

tapis files commads (services) are for dealing with the file system

tapis files list agave://designsafe.storage.default/fmk

- Assumes designsafe.storage.default is default storage system
- Also assuming that fmk is your username!

files

Typical file operations

```
tapis files mkdir agave://designsafe.storage.default/fmk test
```

```
tapis files upload agave://designsafe.storage.default/fmk/test piMPI.c
```

```
tapis files download agave://designsafe.storage.default/fmk/test/piMPI.c
```

```
tapis files copy agave://designsafe.storage.default/fmk/test/piMPI.c fmk/test/pi.c
```

```
tapis files delete agave://designsafe.storage.default/fmk/test
```

- **Delete deletes all files and subdirectories**

apps

tapis apps is code that can be run on a specific execution system. They are stored in the storage system.

wrapper.sh In order to run your application, you will need to create a wrapper template that calls your executable code. Your exe can exist in your app folder or elsewhere (as long as it is accessible for other users and yourself depending on exe system you use).
wrapper.sh is just a bash script, runs unix commands like cd, pwd, ls, ibrun.



Files

DEMO to demonstrate interacting with
remote default storage system.

apps

best way to see what is up (i.e. create one)
is to clone an existing working one!

to clone one you need to be a PUBLISHER on an existing exe service. If
you created one earlier you are!

```
tapis apps clone -e designsafe.demo.exec.frontera.fmk -n mpiCR -x 0.0.1 simcenter-dakota-1.0.0u1
```

- e name of execution system
- n name of application, something meaningful, e.g. mpiCR == mpi compile and run
- x version

Lastly name of app to clone, here one of simcenters

```
tapis apps show -f json myClone-0.0.1 > myClone.json
```

This puts output of command into myClone.json file

apps

myClone.json has stuff we are going to change:

```
"defaultProcessorsPerNode": 56,  
"defaultNodeCount": 1,  
  
"tags": [  
    "mpi"  
,  
  
"parameters":{  
    replace with myCloneParamaters.json  
}
```

You will notice myClone.json has some other inputs of interest:
"deploymentPath": "tg457427/applications/myClone-0.0.1",
"templatePath": "wrapper.sh",

apps

You will notice myClone.json has some other inputs of interest:

```
"deploymentPath": "fmk/applications/myClone-0.0.1",
"templatePath": "wrapper.sh",
```

We are going to replace the existing wrapper.sh with the one in
code/agave/wrapper.sh

```
programFILE="${programFile##*/}"
cd "${inputDirectory}"
echo "currentDIR"
pwd
echo "directory listing"
ls -sal
mpicc $programFILE
ibrun ./a.out
```

It's just some linux commands
to cd to inputDirectory, compile
programName with mpi
and then ibrun the exe a.out

tapis files upload agave://designsafe.storage.default/ applications/myClone-0.0.1. wrapper.sh

apps

Finally update the app

```
tapis apps update -F myClone.json myClone-0.0.1
```

DEMO to demonstrate how to create an App
that will run using our EXECUTION system



Apps

jobs

tapis jobs service allows you to run the app with inputs and parameters you provide.

```
{  
    "name": "example",  
    "appId": "myClone-0.0.1",  
    "inputs": {  
        "inputDirectory": "agave://designsafe.storage.default/tg457427/testClone"  
    },  
    "parameters" : {  
        "programFile": "mpiPI.c"  
    },  
    "maxRunTime": "00:01:00",  
    "memoryPerNode": "1GB",  
    "nodeCount": 1,  
    "processorsPerNode": 8,  
    "maxRunTime": "00:01:00",  
    "archive": true,  
    "archiveOnAppError": true,  
    "archiveSystem": "designsafe.storage.default",  
    "notifications": [  
        {  
            "url": "fmckenna@berkeley.edu",  
            "events": "*"  
        }  
    ]  
}
```

```
tapis jobs submit -F submitMPI.json
```

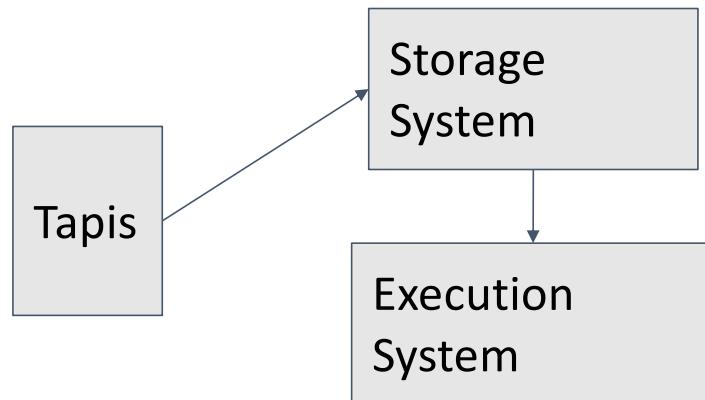
Field	Value
id	5ce7f59d-0c4f-46c1-806a-35965317525f-007
name	example
status	ACCEPTED

```
tapis jobs status 5ce7f59d-0c4f-46c1-806a-35965317525f-007
```

Field	Value
id	5ce7f59d-0c4f-46c1-806a-35965317525f-007
name	example
status	RUNNING

What happens when submitting a job

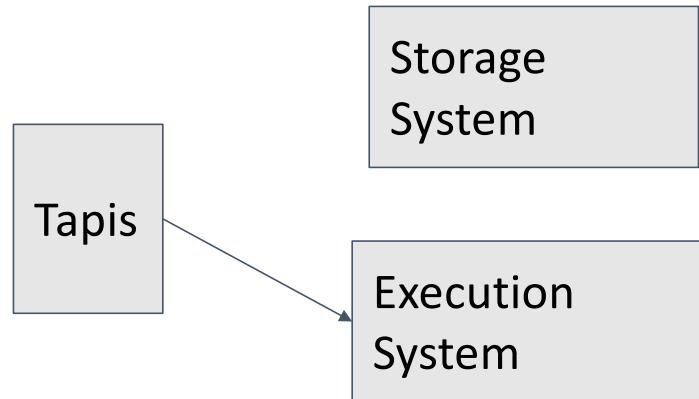
1. Tapis SSHs into the storage system and moves the necessary app files into the execution system.



Courtesy: Sal Tijerina, DesignSafe-ci

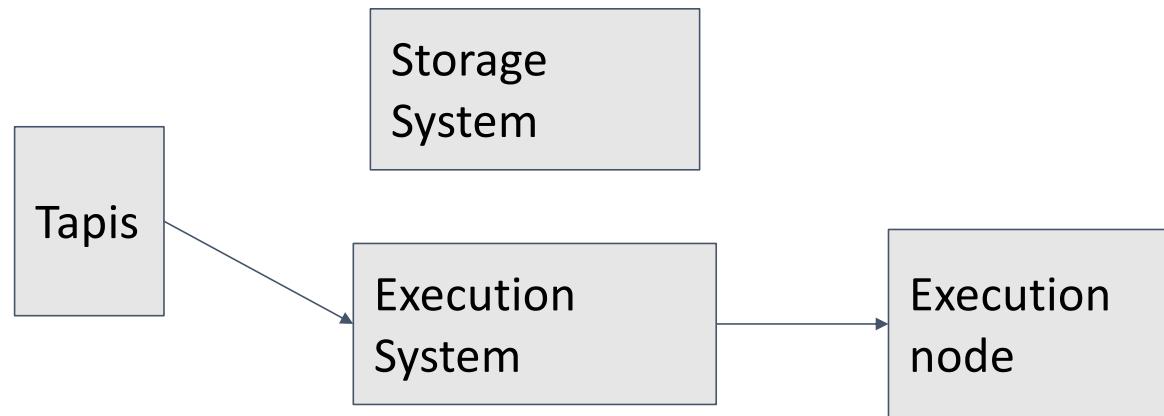
What happens when submitting a job

1. Tapis SSHs into the storage system and moves the necessary app files into the execution system.
2. Tapis SSHs into the execution system, creates the SLURM job text and submits it.



What happens when submitting a job

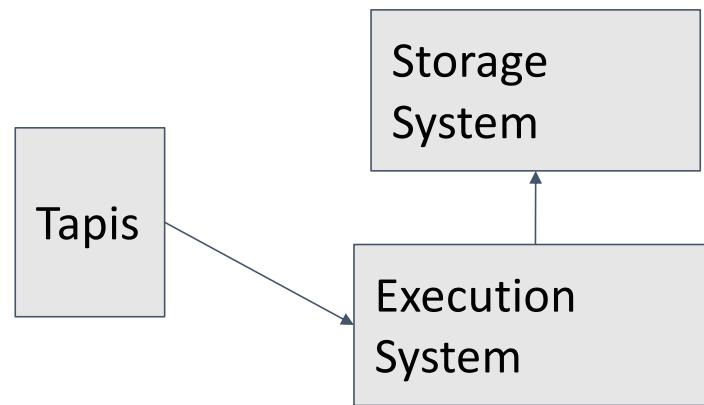
1. Tapis SSHs into the storage system and moves the necessary app files into the execution system.
2. Tapis SSHs into the execution system, creates the SLURM job text and submits it.
3. Tapis monitors the execution node.



Courtesy: Sal Tijerina, DesignSafe-ci

What happens when submitting a job

1. Tapis SSHs into the storage system and moves the necessary app files into the execution system.
2. Tapis SSHs into the execution system, creates the SLURM job text and submits it.
3. Tapis monitors the execution node.
4. Tapis SSHs into the execution system and archives the output files to the storage system.



Courtesy: Sal Tijerina, DesignSafe-ci

When FINISHED

```
tapis jobs show -f json 5ce7f59d-0c4f-46c1-806a-35965317525f-007
```

```
"executionSystem": {  
    "href": "https://agave.designsafe-ci.org/systems/v2/designsafe.demo.exec.frontera.tg457427"  
},  
"archiveSystem": {  
    "href": "https://agave.designsafe-ci.org/systems/v2/designsafe.storage.default"  
},  
"archiveData": {  
    "href": "https://agave.designsafe-ci.org/files/v2/listings/system/designsafe.storage.default/tg457427/archive/jobs/job-5ce7f5  
},  
"owner": {  
    "href": "https://agave.designsafe-ci.org/profiles/v2/tg457427"  
},
```

```
tapis files list agave:// designsafe.storage.default/tg457427/archive/jobs/job-5ce7f59d-0c4f-46c1-806a-35965317525f-007
```

DEMO to demonstrate how to run a job using
our newly created app



Jobs