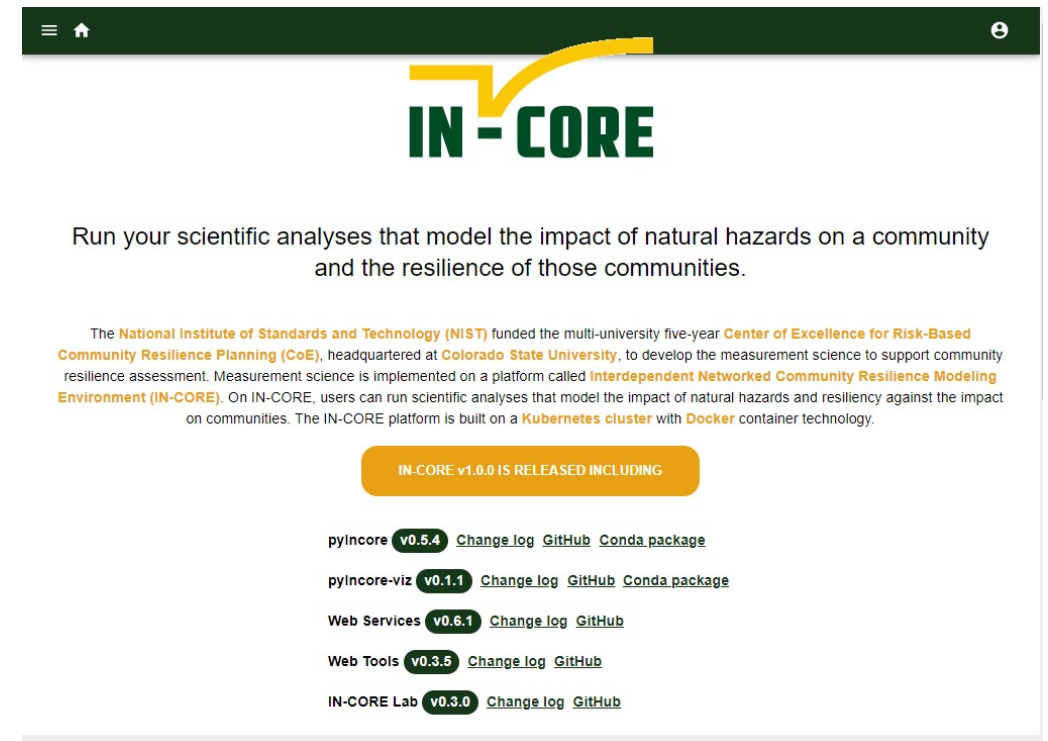
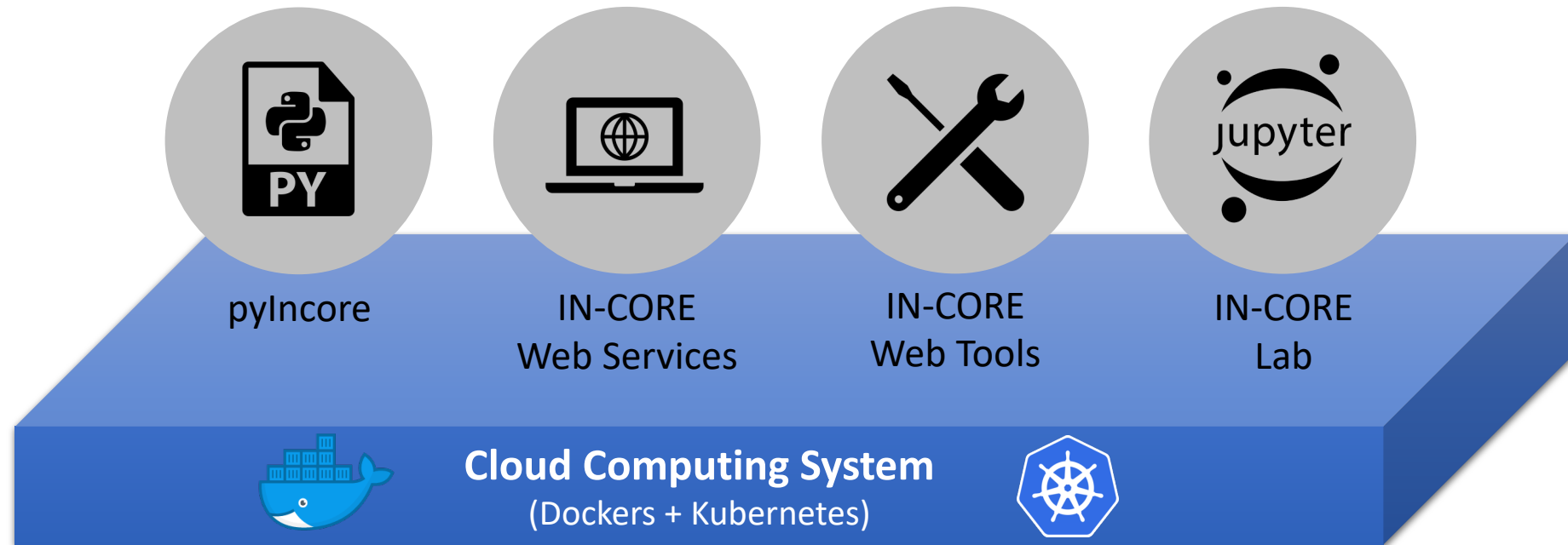


IN-CORE Release

- Latest release: 2.2.1 (Released on May 25, 2021)
- Source code at GitHub
 - <https://github.com/IN-CORE>
 - Mozilla Public License v2.0 (MPL-2.0)
- Conda packages
 - <https://anaconda.org/IN-CORE>
- IN-CORE landing page
 - <https://incore.ncsa.illinois.edu/>



Architecture



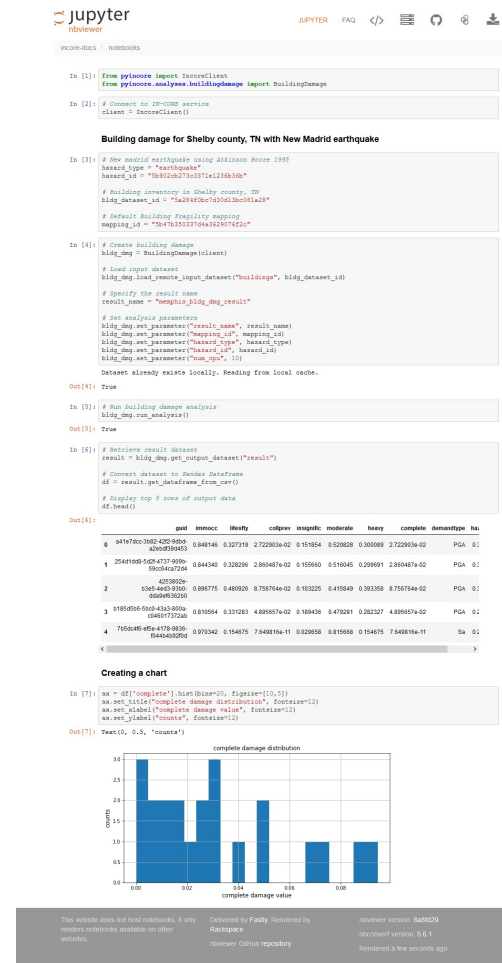
Containers on Kubernetes

- Container (Docker): A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it
- Kubernetes is a container management system
- The technology brings us
 - Automatic scaling corresponding to demands
 - Portability – deployable to different cloud
 - Streamline deployment from development and testing



pyIncore & pyIncore-viz

- Python library (modules) for IN-CORE
- Three components
 - Interact with IN-CORE web services
 - Base classes for analysis and datasets
 - Analyses
- pyIncore-viz
 - Visualization methods and utilities
- How to install
 - `conda install -c in-core pyincore`
 - `conda install -c in-core pyincore-viz`
- Documentation is available
 - Jupyter notebooks with example analysis
 - Technical reference documents



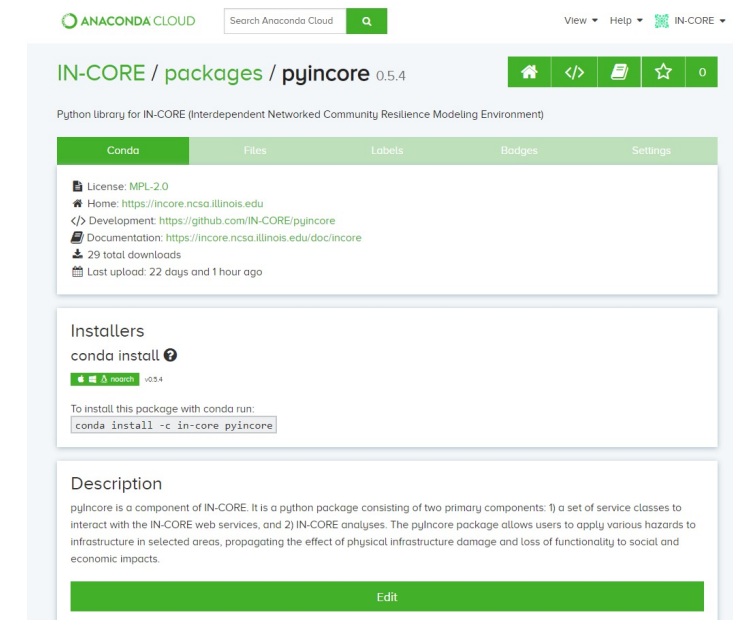
Currently Available Analyses

- Bridge damage
- Building damage
- Building functionality
- Building structural loss
- Cumulative building damage
- Electric power facility damage
- Nonstructural building damage
- Tornado Electric Power Network (EPN) damage
- Pipeline damage
- Pipeline damage with repair rate
- Water facility damage
- Capital shocks
- Mean damage
- Monte Carlo failure probability
- Building Portfolio recovery
- Transportation recovery
- Housing unit allocation
- Population dislocation
- Joplin Computable General Equilibrium (CGE)
- Seaside CGE
- Road damage

Three testbeds (Joplin, Seaside, Galveston) are available as Jupyter Notebook
More analyses will be added in near future

pyIncore Resources

- pyIncore
 - GitHub: <https://github.com/IN-CORE/pyincore>
 - Anaconda: <https://anaconda.org/IN-CORE/pyincore>
 - General documentation: <https://incore.ncsa.illinois.edu/doc/incore/pyincore.html>
 - Technical reference documentation: <https://incore.ncsa.illinois.edu/doc/pyincore/>
- pyIncore-viz
 - More capability will come in future release
 - GitHub: <https://github.com/IN-CORE/pyincore-viz>
 - Anaconda: <https://anaconda.org/IN-CORE/pyincore-viz>



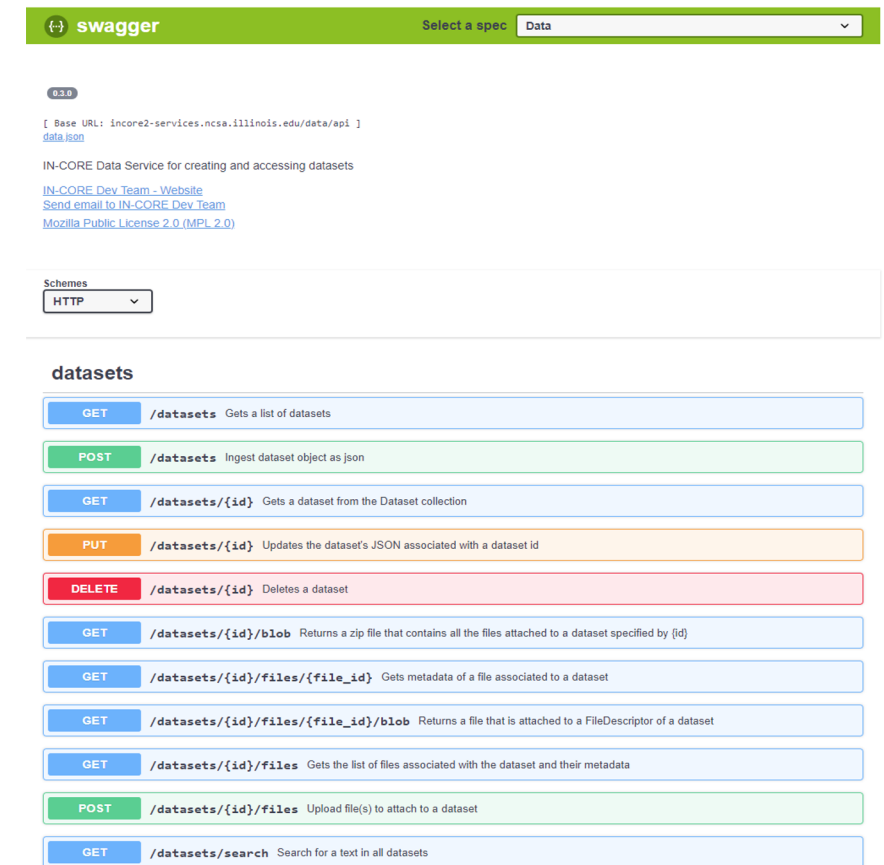
The screenshot shows the Anaconda Cloud interface for the package 'pyincore' version 0.5.4. The page includes a search bar, navigation tabs (Conda, Files, Labels, Badges, Settings), and a 'Conda' tab selected. Under the 'Conda' tab, it lists the license (MPL-2.0), home page, development URL, documentation URL, total downloads (29), and last upload time (22 days and 1 hour ago). Below this, the 'Installers' section shows the command 'conda install' and a button to 'Install'. The 'Description' section explains that pyincore is a component of IN-CORE, a Python package with two primary components: a set of service classes to interact with IN-CORE web services, and IN-CORE analyses. The package allows users to apply various hazards to infrastructure in selected areas, propagating the effect of physical infrastructure damage and loss of functionality to social and economic impacts. An 'Edit' button is visible at the bottom of the description section.

IN-CORE Web Services

- RESTful Web Service Technology
- Database: MongoDB
- Authentication service
- Data service
 - Storing/managing datasets
- Hazard service
 - Storing hazard definitions
 - Getting hazard value by location
 - Earthquake
 - Tsunami
 - Tornado
 - Hurricane wind field
- DFR3 service
 - Storing/managing fragility curve sets, damage functions, repair, recovery, restoration
 - Matching inventory to fragility curve set
- Geospatial Viz service
 - Generating geospatial map/layer images
- Semantic service
 - Storing/managing definition of datasets
 - Coming to next release
- Space service
 - Creating content spaces
 - Access control

IN-CORE Web Services

- How to use IN-CORE Web Services
 - Need to have a user account managed by NCSA identity management system
 - For authentication
 - For authorization (access control)
 - Various ways
 - RESTful web service clients
 - Web browser
 - pyIncore
 - IN-CORE Web Tools (browsing only)
- Technical reference documentation is available



The image shows the Swagger UI for the IN-CORE Data Service. At the top, there is a green header with the Swagger logo and a dropdown menu labeled "Select a spec" with "Data" selected. Below the header, the version "0.3.0" is displayed. The base URL is "incore2-services.ncsa.illinois.edu/data/api" and the data is in JSON format. The service description is "IN-CORE Data Service for creating and accessing datasets". Links for the IN-CORE Dev Team website, email, and the Mozilla Public License 2.0 are provided. A "Schemes" dropdown is set to "HTTP". The main section, titled "datasets", lists the following endpoints:

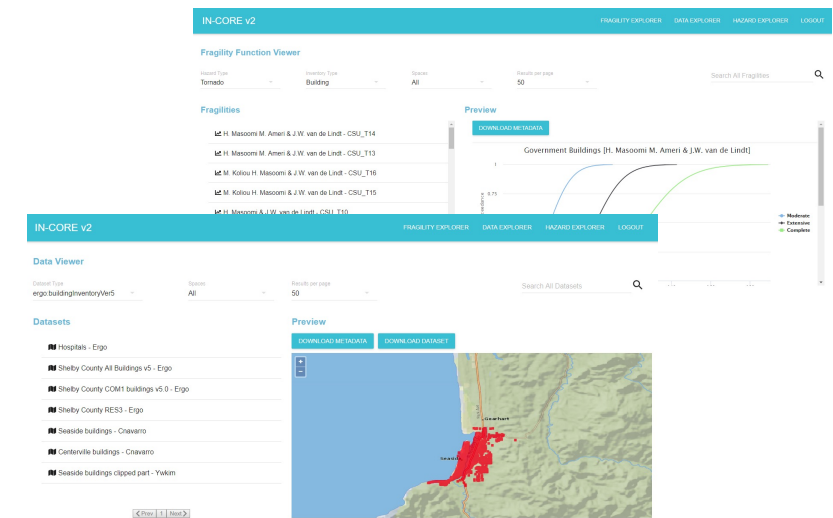
Method	Endpoint	Description
GET	/datasets	Gets a list of datasets
POST	/datasets	Ingest dataset object as json
GET	/datasets/{id}	Gets a dataset from the Dataset collection
PUT	/datasets/{id}	Updates the dataset's JSON associated with a dataset id
DELETE	/datasets/{id}	Deletes a dataset
GET	/datasets/{id}/blob	Returns a zip file that contains all the files attached to a dataset specified by {id}
GET	/datasets/{id}/files/{file_id}	Gets metadata of a file associated to a dataset
GET	/datasets/{id}/files/{file_id}/blob	Returns a file that is attached to a FileDescriptor of a dataset
GET	/datasets/{id}/files	Gets the list of files associated with the dataset and their metadata
POST	/datasets/{id}/files	Upload file(s) to attach to a dataset
GET	/datasets/search	Search for a text in all datasets

IN-CORE Web Services Resources

- GitHub:
 - <https://github.com/IN-CORE/incore-services>
- Technical reference documentation:
 - <https://incore.ncsa.illinois.edu/doc/api/>

IN-CORE Web Tools

- Lightweight web applications for IN-CORE Web Services
- Allows users to browse, search, and preview data from the service
- Data browser
 - Client to data service
- Fragility browser
 - Client to DFR3 service
 - Currently it shows fragilities
- Hazard browser
 - Client to hazard service
- Login with your account credential to access tools at
 - <https://incore.ncsa.illinois.edu>

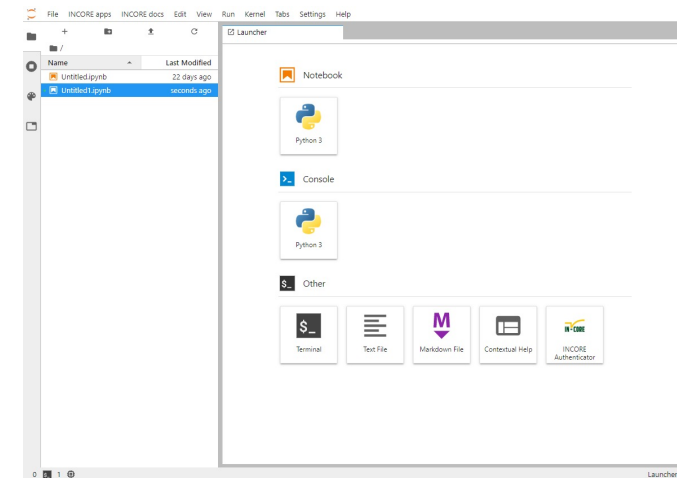


IN-CORE Web Tools Resources

- Access at
 - <https://incore.ncsa.illinois.edu>
- GitHub:
 - <https://github.com/IN-CORE/incore-ui>
- General documentation:
 - <https://incore.ncsa.illinois.edu/doc/incore/webtools.html>

IN-CORE Lab

- Customized JupyterLab
- Integrated environments for developing algorithms
 - Menu items to access documentations, IN-CORE Web Tools
 - Authentication (single-sign-on)
- Two ways to use IN-CORE Lab
 - Locally (a docker image will be available)
 - Online (JupyterHub at NCSA)
- Online version:
 - pyIncore is installed with all dependent libraries
 - Includes popular python libraries such as Pandas, GeoPandas, Matplotlib, etc.
 - Account and allocation policy are under development for public access



Support

- Email: incore-dev@lists.illinois.edu
- Documentation: tutorials, tips, and FAQ
- Slack channel will be available soon

AGENDA

Mountain Time	Join Zoom Meeting: https://zoom.us/j/95020363783?pwd=aUFFYU5FNmxVUGdwM3F0clVuTGlyUT09 Meeting ID: 950 2036 3783 Passcode: 451704
11:00 – 11:35AM	Session 1: Overview of IN-CORE & User Workshop Objectives <ul style="list-style-type: none">• Welcome - John van de Lindt (5 min)• Scientific overview of IN-CORE - John van de Lindt (10 min)• Platform overview - Jong Lee (15 min)• Workshop Objectives - Jong Lee (5min)
11:35 – 11:50AM	BREAK and PREP <ul style="list-style-type: none">• Prepare their training environments (logging in and uploading files to incore-lab etc.)
11:50 AM – 12:50 PM	Session 2: Hazard (Earthquake, Tornado) <ul style="list-style-type: none">• Basics of IN-CORE modules (5 min)• Tornado (15 min)• Earthquake (15 min)• Hands-on exe - assignments (20 min)• Review answers (5 min)
12:50 – 12:55PM	BREAK

12:55 – 1:55PM	Session 3: Damage Analysis (Buildings and EPF) <ul style="list-style-type: none"> • Fragility curve and Mapping (just presentation) - concept (5 min) • Inventory data (10 min) • Building damage analysis (10 min) • EPF damage analysis (5 min) • Hands-on exercise (25 min) • Review answers (5 min)
1:55 – 2:05PM	BREAK
2:05 – 2:45PM	Session 4: Visualization of Outputs <ul style="list-style-type: none"> • Joining datasets (5 min) • Pyincore-viz (10 min) • Pandas (5 min) • Hands-on exe (15 min) • Review answers (5 min)
2:45 – 3:25PM	Session 5: Use Case - How to do research with IN-CORE <ul style="list-style-type: none"> • Yousef Darestani - (15-minute presentation, 5 min Q&A) • Dylan Sanderson - (15-minute presentation, 5 min Q&A)
3:25 – 3:30PM	CLOSING and NEXT USER WORKSHOP

Preparing Session Materials

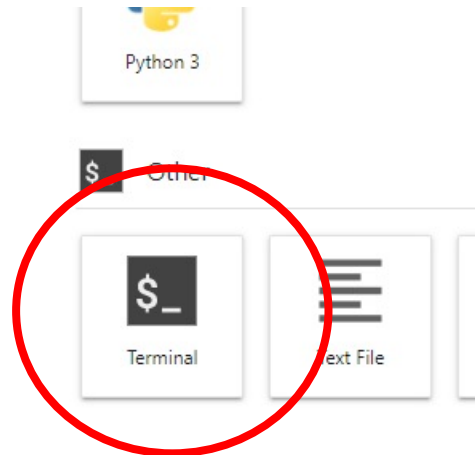
- At <https://incore.ncsa.illinois.edu>, Login



- After login, click on “IN-CORE lab”



- Click on “Terminal”



- Type the following command to download the file

```
wget -O workshop.zip https://bit.ly/3fMRuRL
```

- Type the following command to unzip the file

```
unzip workshop.zip
```