# NEXTGEN SIMULATION DEVELOPMENT TOOLS: NGEN-DATASTREAM

LYNKER: JORDAN LASER, ZACH WILLS, MIKE JOHNSON, JUSTIN SINGH-MOHUDPUR, NELS FRAZIER, AUSTIN RANEY

ALABAMA WATER INSTITUTE: ARPITA PATEL, JAMES HALGREN, JOSH CUNNINGHAM, SHAHABUL ALAM, TRUPESH PATEL, HARI TEJA JAJULA, BENJAMIN LEE

SPRING 2024

# OVERVIEW

- Motivation for building ngen-datastream
- Development Roadmap
- Conceptual model
- Technical breakdown
- Usage
- Workshop
- Future work

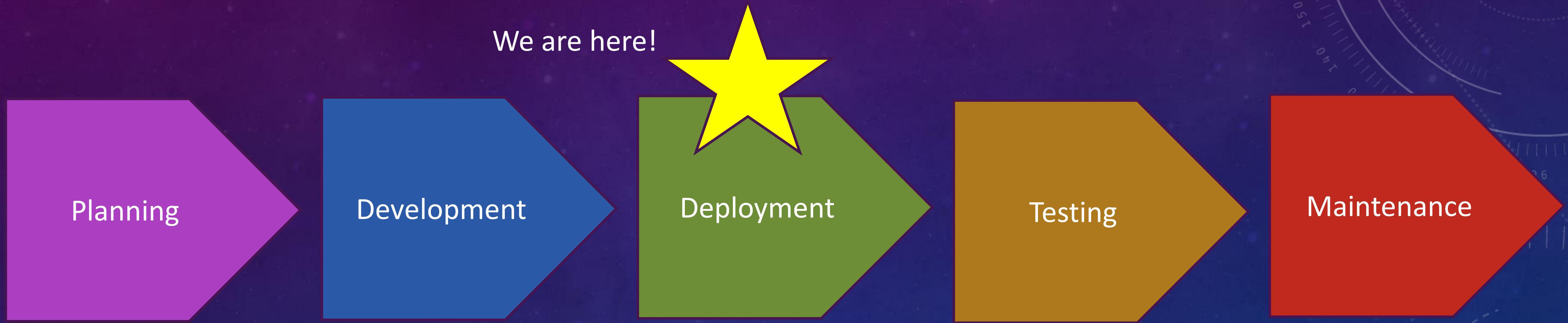
# MOTIVATION

- NextGen In A Box made NextGen more accessible to the community, however we still wrestle with a few issues
  - Uniform data pipeline that abstracts all required steps to run and version a NextGen execution.
  - Reproducibility
  - Lack of widely adopted data standard
  - Scaling from laptop to HPC
  - Need baseline dataset to evaluate new realizations

- ngen-datastream aims to address these issues
  - Batteries included while not dogmatic – ngen-datasteam will perform every required step for you, but the user can compute some steps separately, and provide those files to ngen-datastream via the resource directory or command line arguments.
  - Uses the NextGen In A Box standard run directory.
  - Metadata – all relevant information about user inputs, code versions, host architecture, etc. so that a NextGen execution can be understood and reproduced.
  - Infrastructure as Code, Terraform – ngen-datastream can issue NextGen jobs an AWS state machine that use Lambda functions to coordinate NextGen executions in the cloud. This allows users to customize their host to match their compute requirements.
    - Generates baseling dataset in the cloud

# DEVELOPMENT ROADMAP

ngen-datastream is already a powerful tool, but is still under development and has not been rigorously tested within the community → We encourage community feedback and questions. If you discover a bug, or would appreciate different functionality, let us know by submitting an issue to the repository



We are here!

**Planning**

- Identifying needs in community
- Software architecture decisions

**Development**

- Writing the software
- Developer based testing

**Deployment**

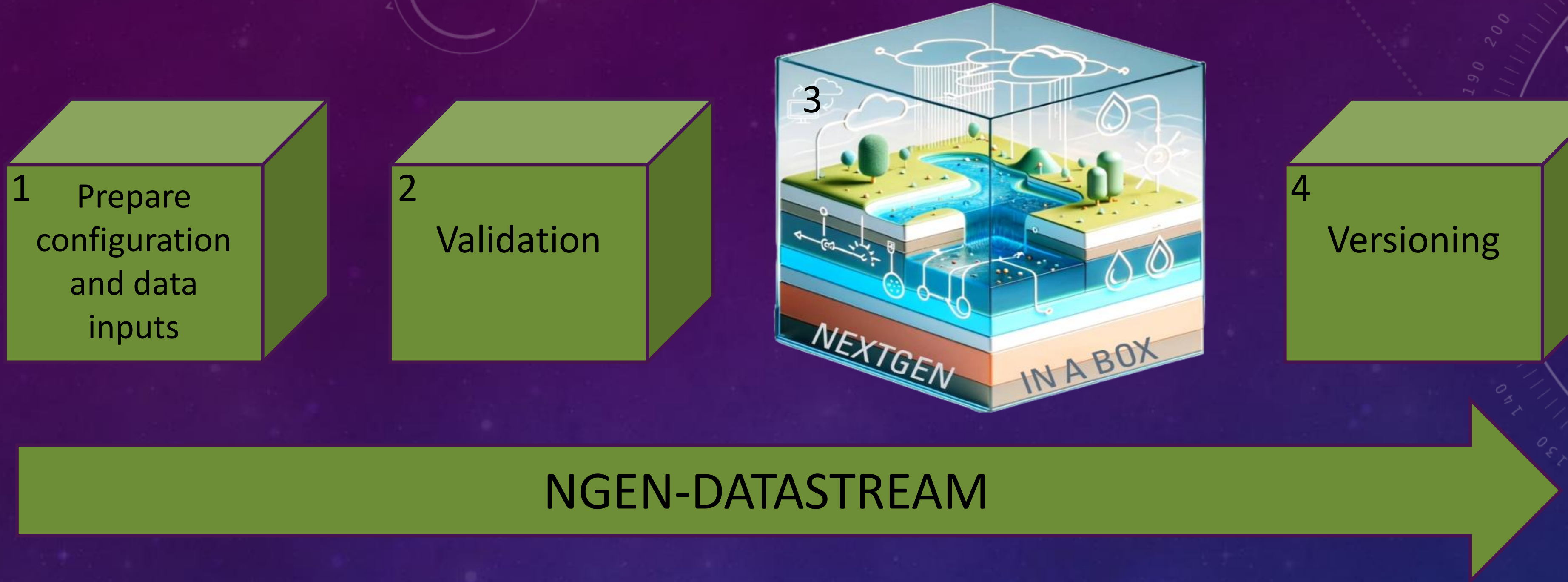
- Releasing software to community
- ngen-datastream version 1.0

**Testing**

- Continuous Integration Continuous Deployment (CI/CD)
- Feedback from community

**Maintenance**

- Add features



1 Prepare configuration and data inputs

2 Validation

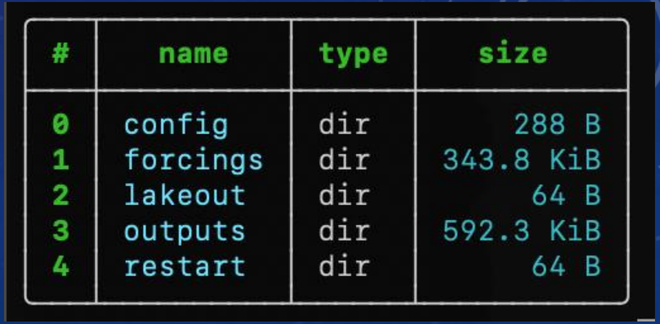
3

4 Versioning

NGEN-DATASTREAM

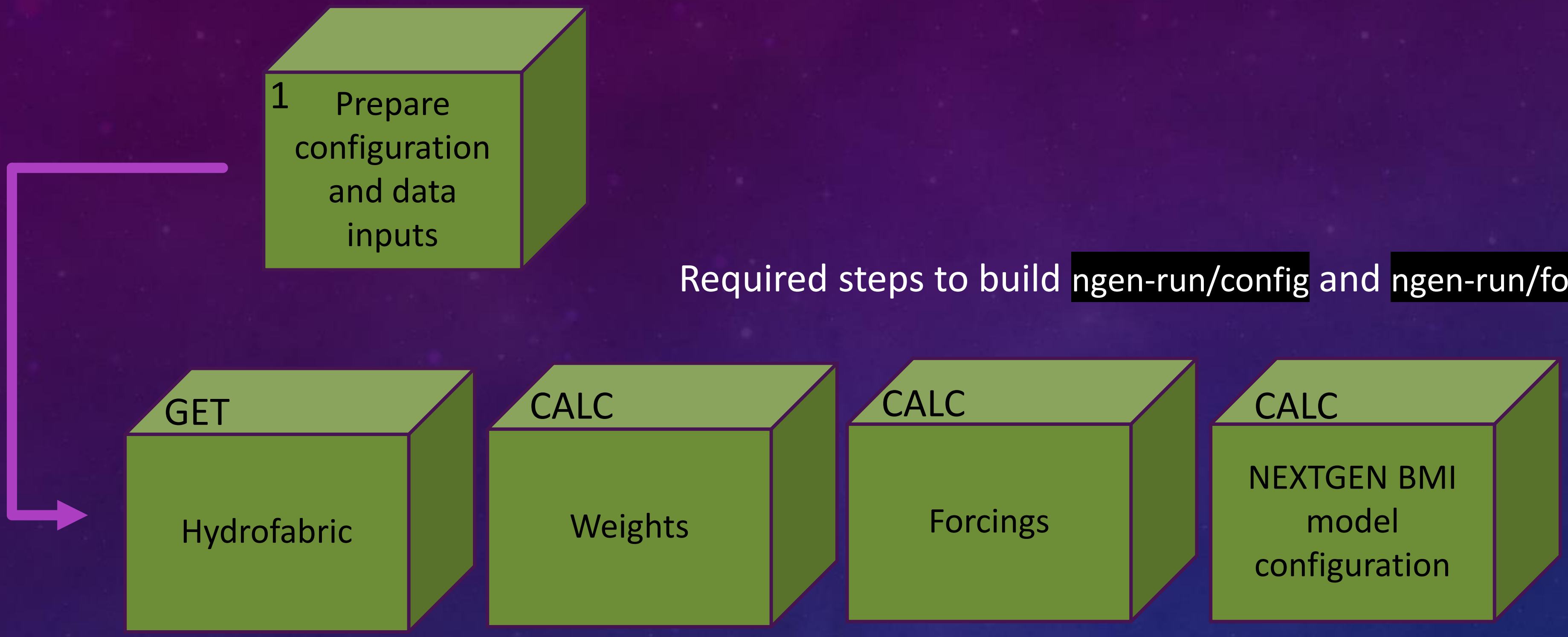
ngen-datastream refers to the software chain that builds and validates NextGen input packages (ngen-run/), executes NextGen through NextGen In A Box (NGIAB), and versions the entire run for reproducibility.

This enforces a standard folder ngen-run/ which makes validation and versioning possible. A standard run folder also allows for other new tools to easily integrate with NGIAB (e.g., DataPreprocessor).



```
│ # │   name    │ type │    size    │
├───┼───────────┼──────┼────────────┤
│ 0 │ config    │ dir  │      288 B │
│ 1 │ forcings  │ dir  │ 343.8 KiB  │
│ 2 │ lakeout   │ dir  │       64 B │
│ 3 │ outputs   │ dir  │ 592.3 KiB  │
│ 4 │ restart   │ dir  │       64 B │
```

## Required steps to build ngen-run/config and ngen-run/forcings



1 Prepare configuration and data inputs

GET Hydrofabric

CALC Weights

CALC Forcings

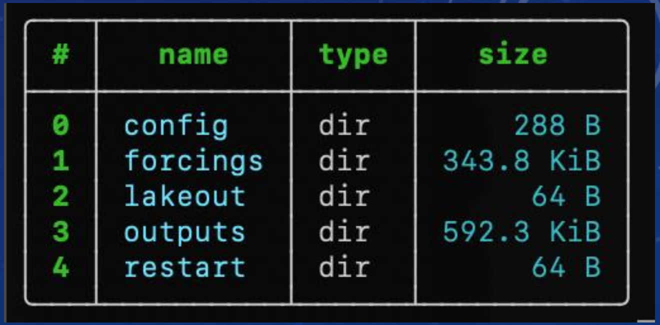
CALC NEXTGEN BMI model configuration

Defines spatial domain

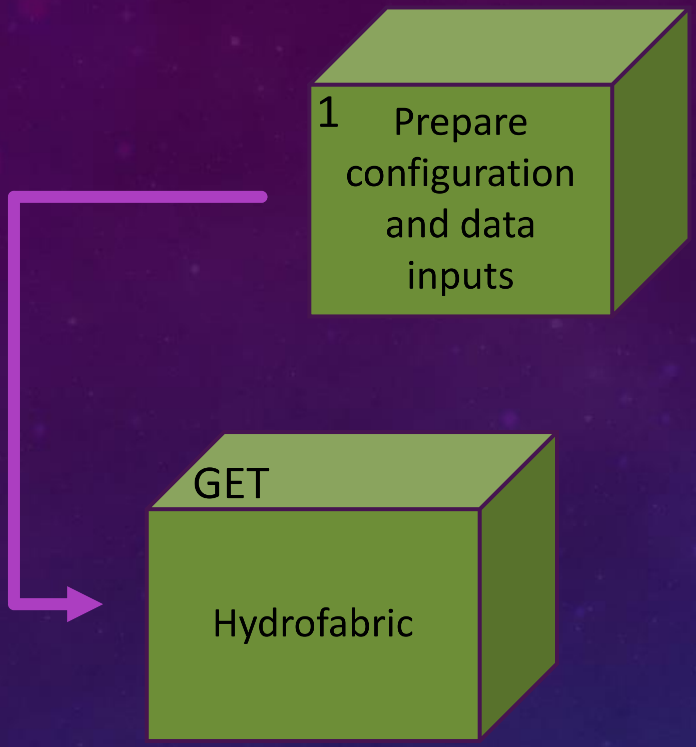
Indices and coverage used to extract catchment averaged forcings

Performs conversion between National Water Model and NextGen forcings formats

Required files for NextGen BMI modules



```
 #    name        type      size
 0    config      dir      288 B
 1    forcings    dir    343.8 KiB
 2    lakeout     dir       64 B
 3    outputs     dir    592.3 KiB
 4    restart     dir       64 B
```

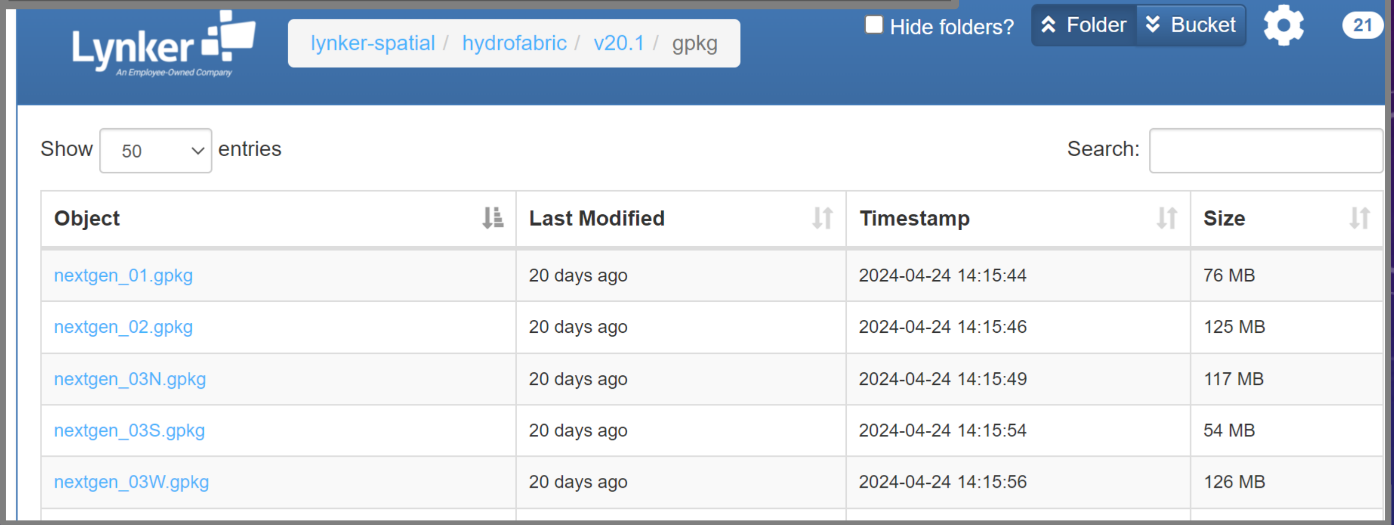

1 Prepare configuration and data inputs

GET

Hydrofabric

Defines spatial domain

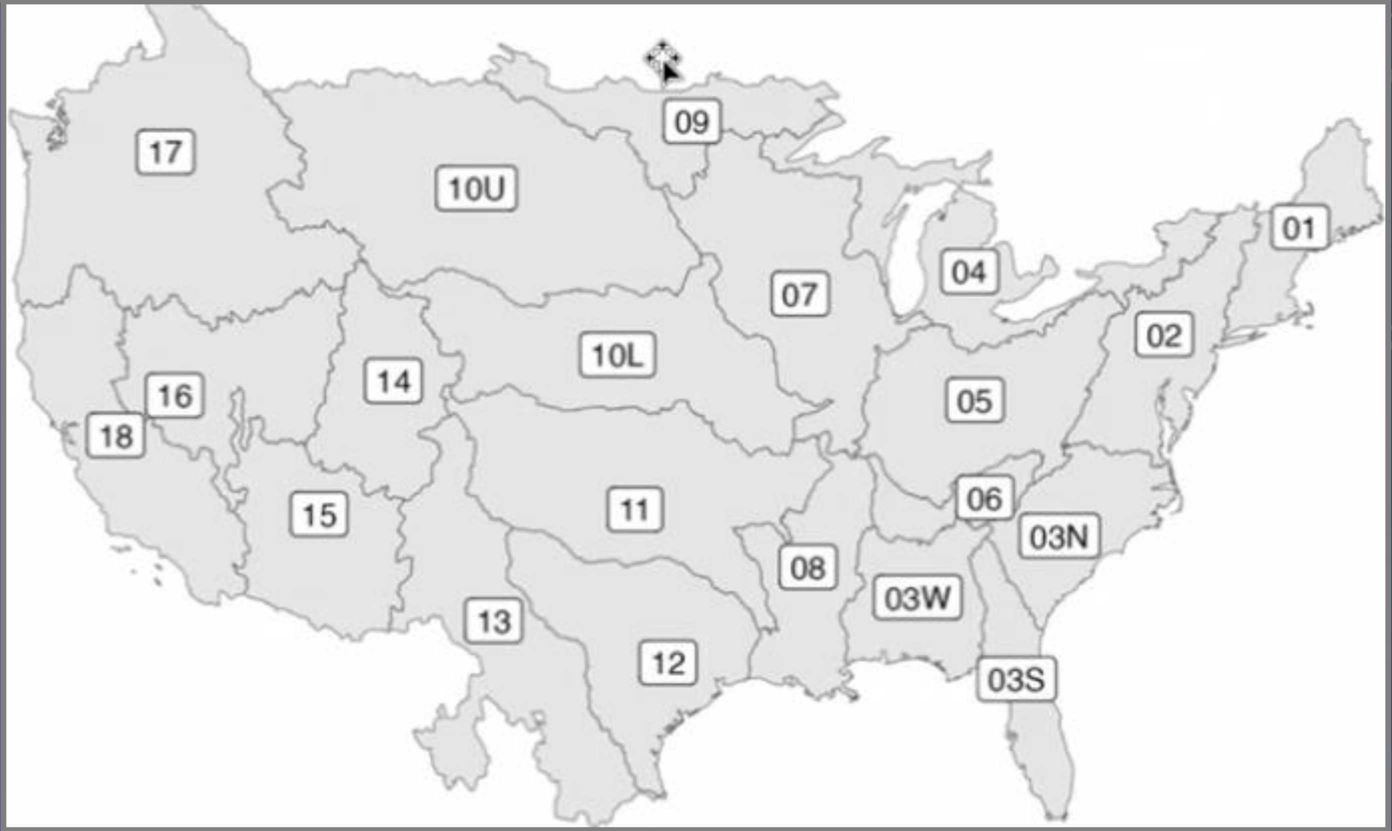
https://www.lynker-spatial.com/#hydrofabric/

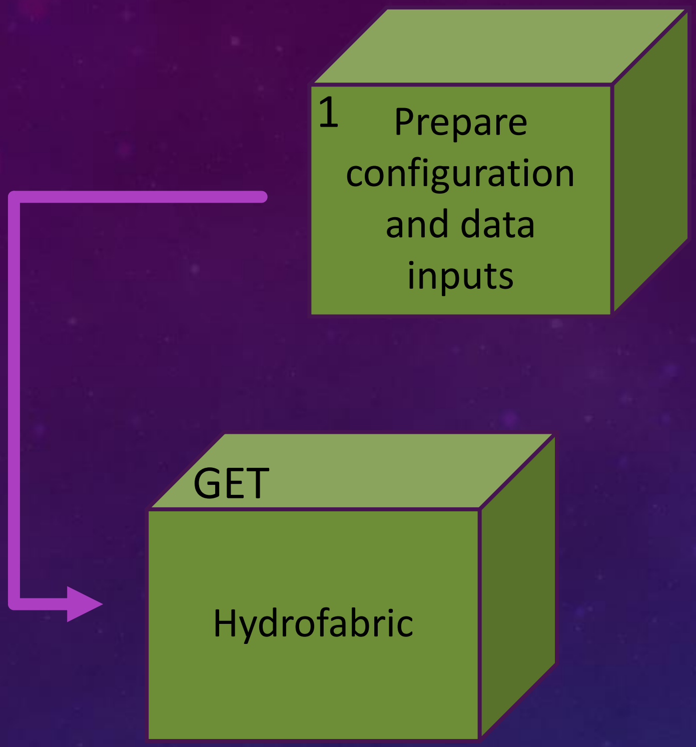

Lynker
An Employee-Owned Company
lynker-spatial / hydrofabric / v20.1 / gpkg
□ Hide folders?
Folder
Bucket
21

Show 50 entries

Search:

| Object | Last Modified | Timestamp | Size |
|---|---|---|---|
| nextgen_01.gpkg | 20 days ago | 2024-04-24 14:15:44 | 76 MB |
| nextgen_02.gpkg | 20 days ago | 2024-04-24 14:15:46 | 125 MB |
| nextgen_03N.gpkg | 20 days ago | 2024-04-24 14:15:49 | 117 MB |
| nextgen_03S.gpkg | 20 days ago | 2024-04-24 14:15:54 | 54 MB |
| nextgen_03W.gpkg | 20 days ago | 2024-04-24 14:15:56 | 126 MB |





1 Prepare configuration and data inputs

GET Hydrofabric

Defines spatial domain

“What if I want to define my own spatial “domain?

Do you know the catchment id you want to subset with?

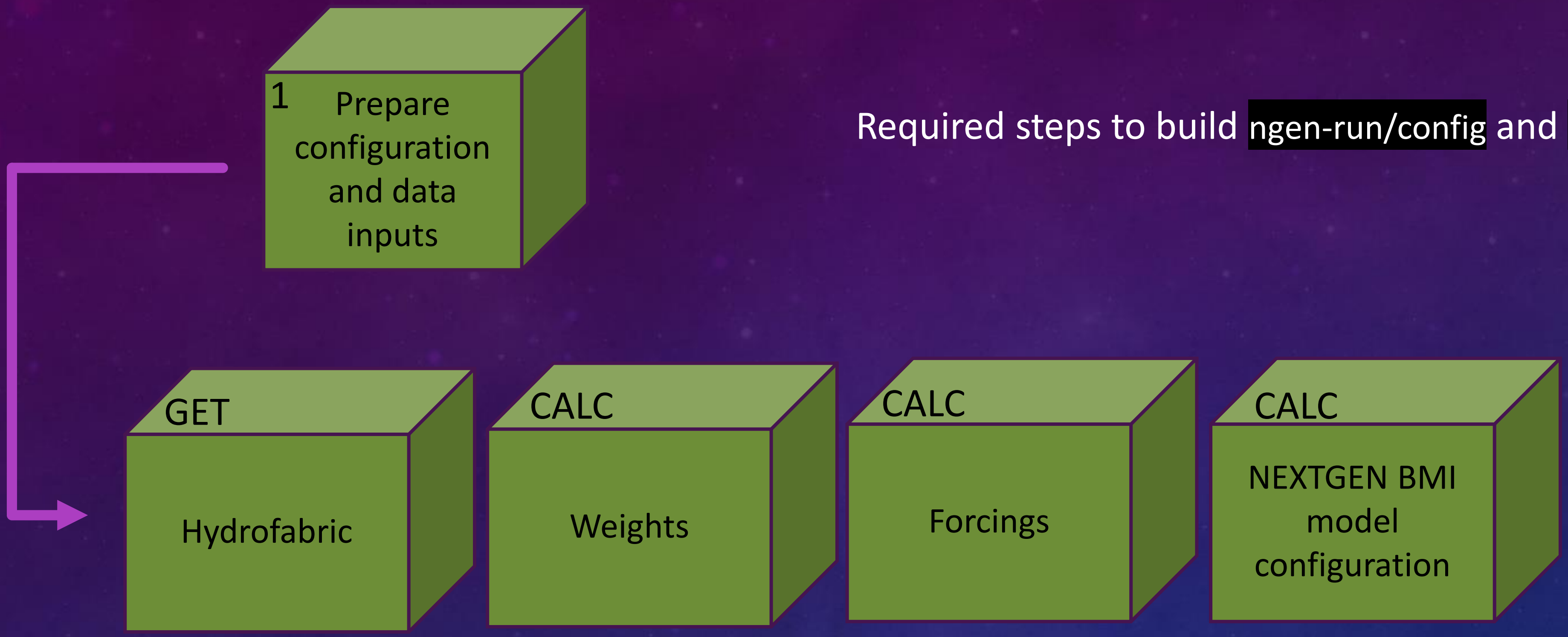
Yes

Use the subsetting options for ngen-datastream!
hfsubset, a tool developed by the hydrofabric team, is integrated into ngen-datastream

No

Use DataPreprocessor for a user-friendly interface to find your desired domain!

Required steps to build ngen-run/config and ngen-run/forcings

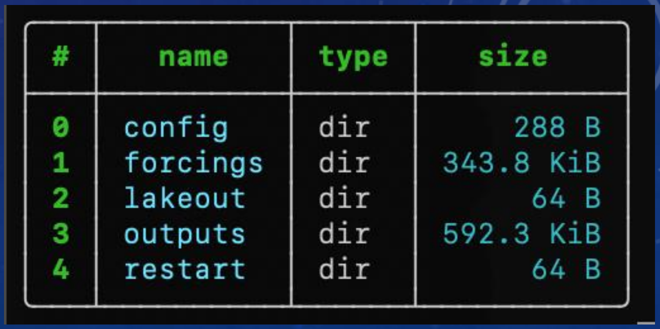

1 Prepare configuration and data inputs
GET
Hydrofabric
CALC
Weights
CALC
Forcings
CALC
NEXTGEN BMI model configuration

Defines spatial domain

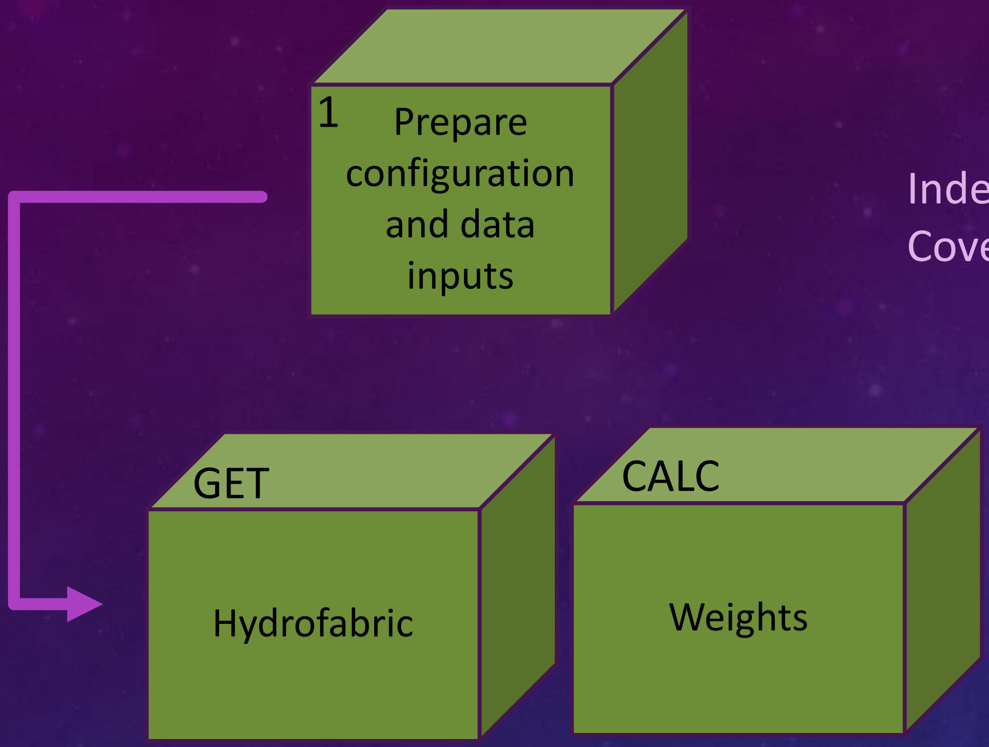
Indices and coverage used to extract catchment averaged forcings

Performs conversion between National Water Model and NextGen forcings formats

Required files for NextGen BMI modules



```
│ # │   name    │ type │   size    │
├───┼───────────┼──────┼───────────┤
│ 0 │ config    │ dir  │     288 B │
│ 1 │ forcings  │ dir  │ 343.8 KiB │
│ 2 │ lakeout   │ dir  │      64 B │
│ 3 │ outputs   │ dir  │ 592.3 KiB │
│ 4 │ restart   │ dir  │      64 B │
```



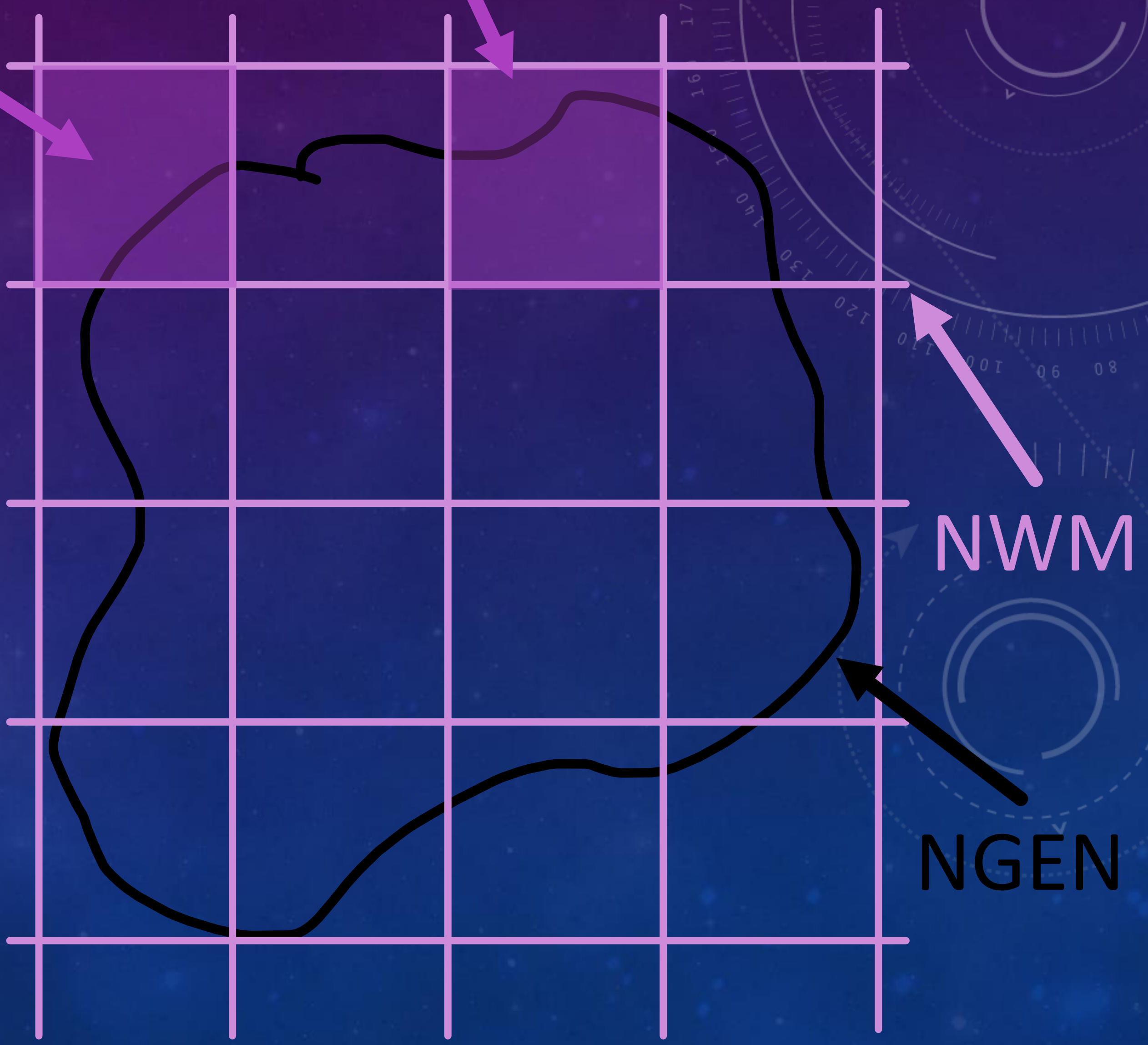
1 Prepare configuration and data inputs

GET Hydrofabric

CALC Weights

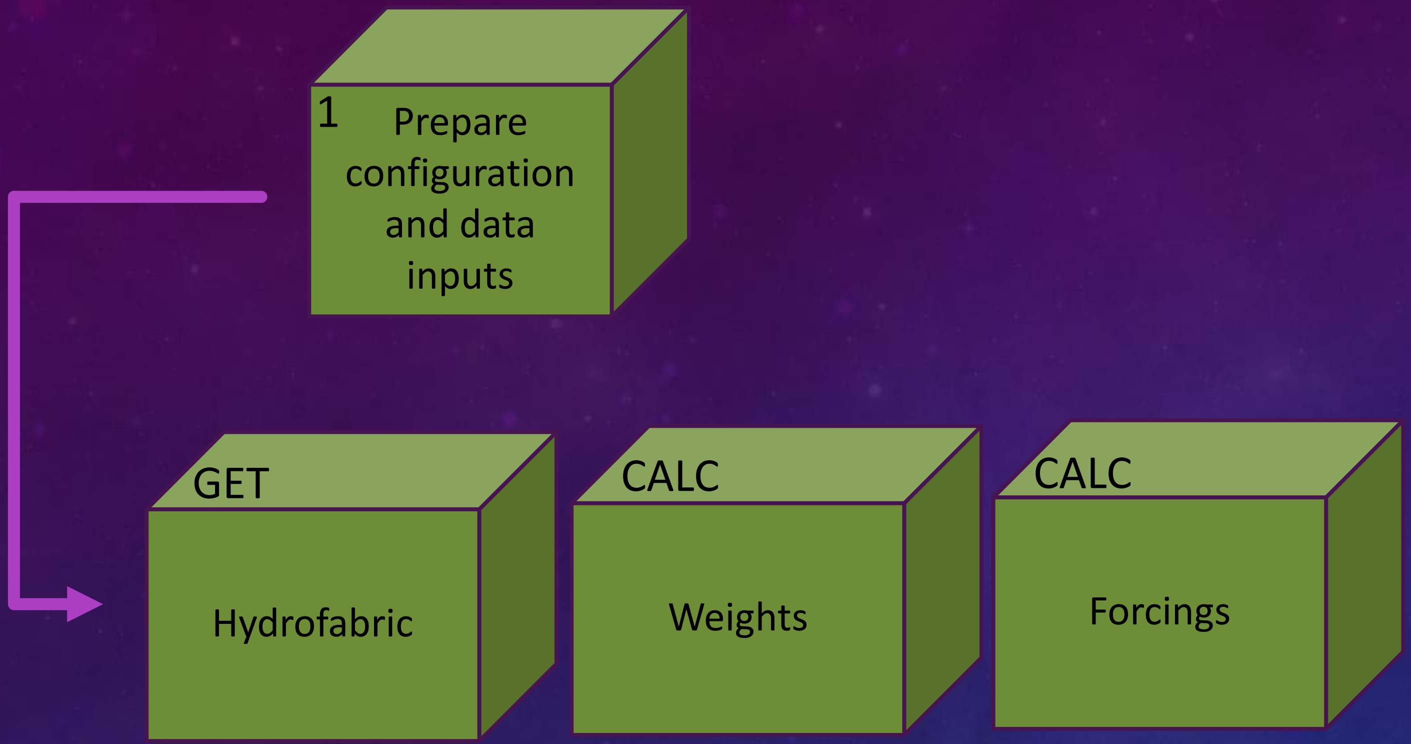
Defines spatial domain

Indices and coverage used to extract catchment averaged forcings



Index: 27463
Coverage: 75%

Index: 27461
Coverage: 25%



1 Prepare configuration and data inputs

GET Hydrofabric

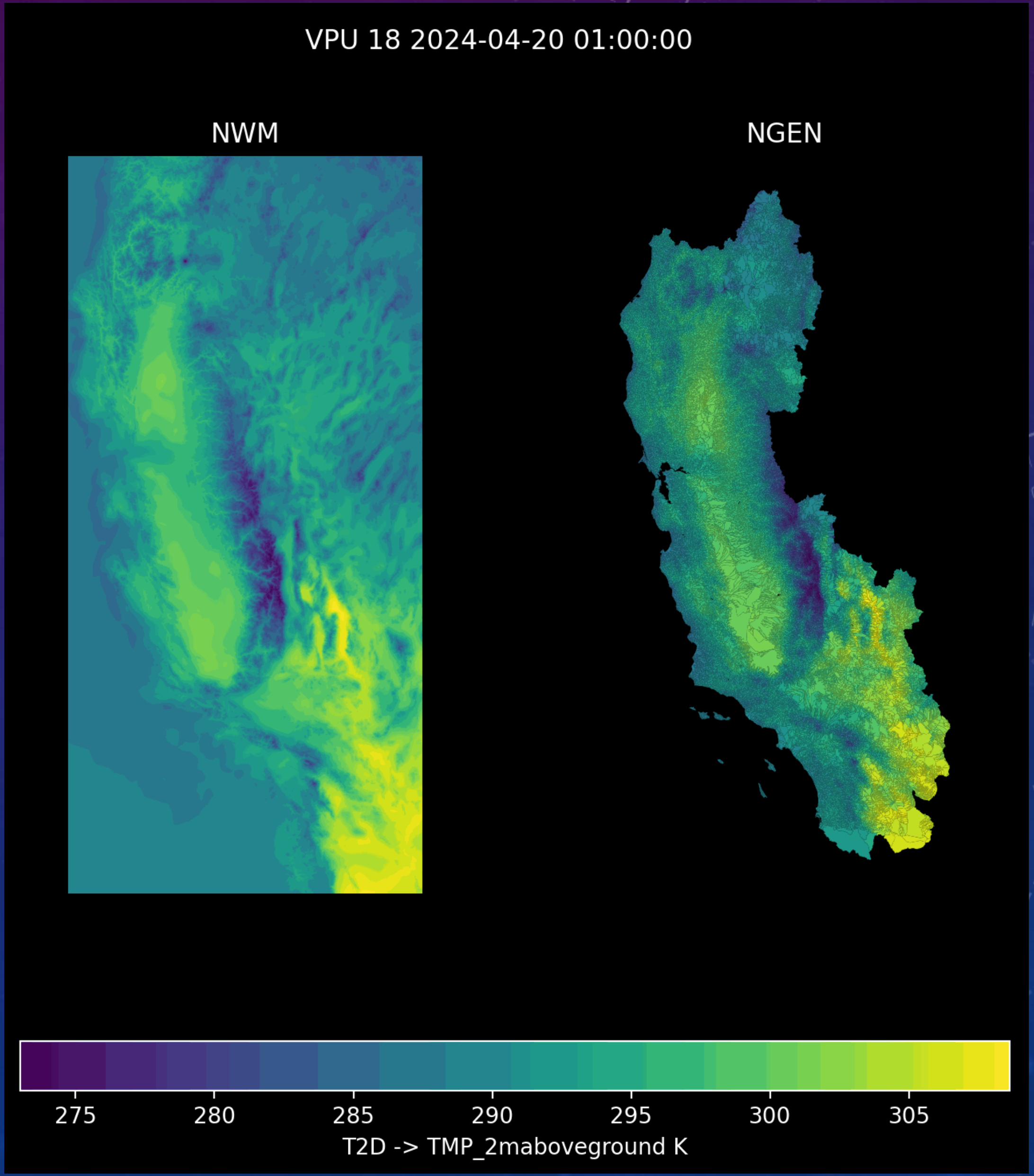
CALC Weights

CALC Forcings

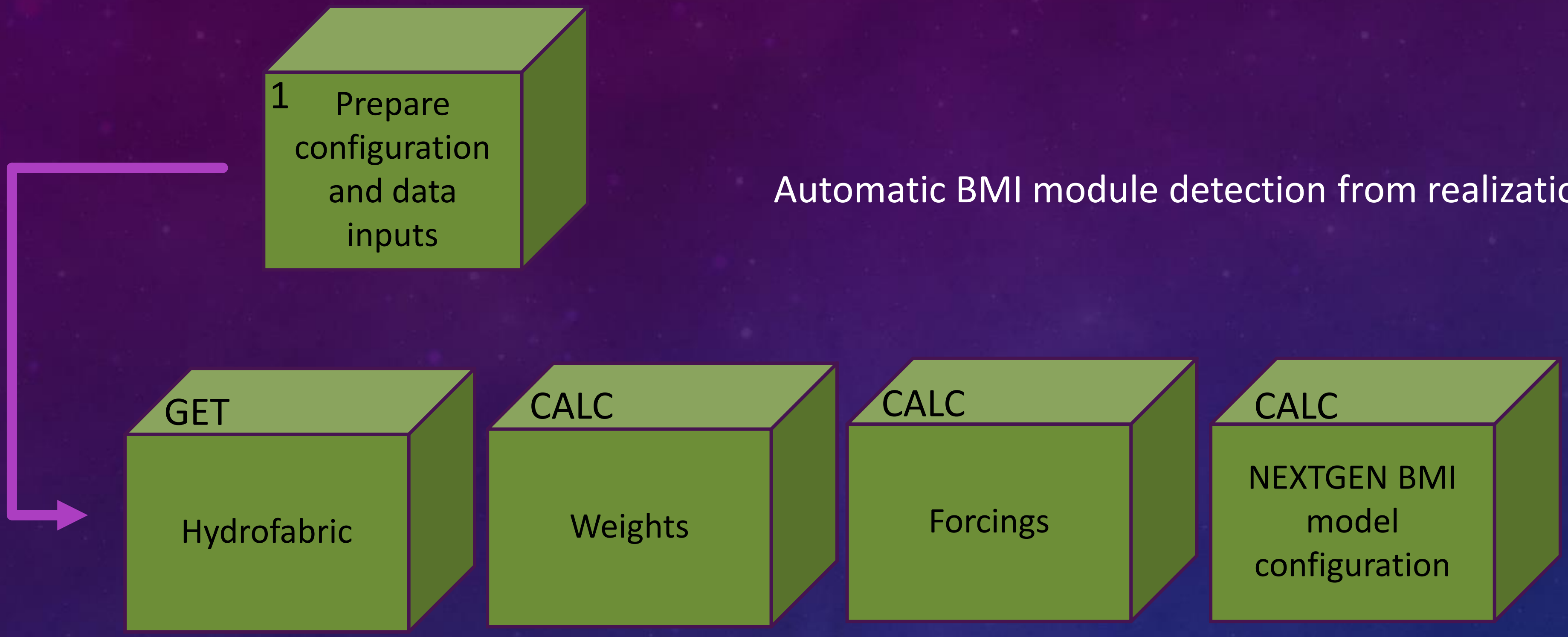
Defines spatial domain

Indices and coverage used to extract catchment averaged forcings

Performs conversion between National Water Model and NextGen forcings formats



## Automatic BMI module detection from realization file

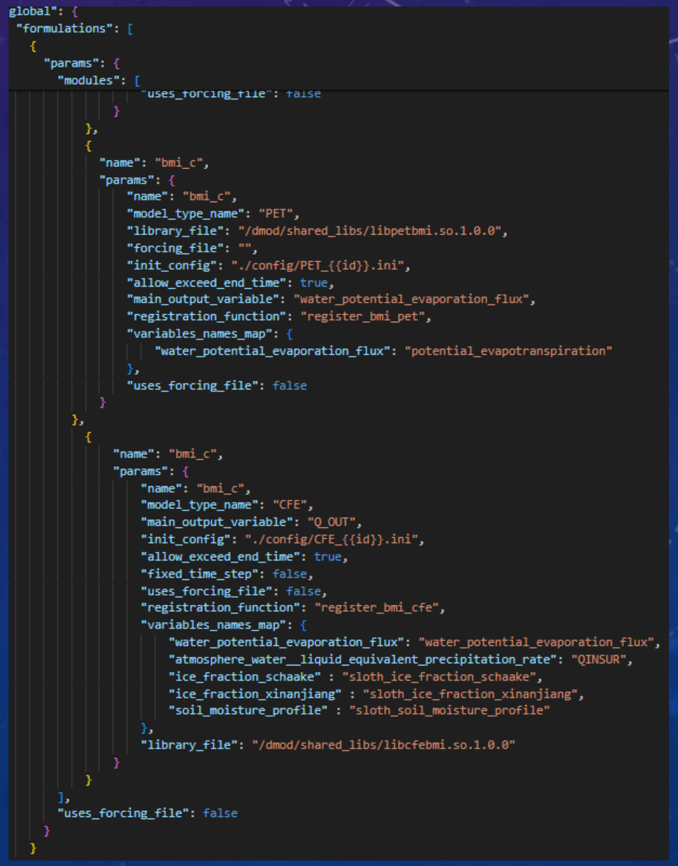

1
Prepare configuration and data inputs
GET
Hydrofabric
CALC
Weights
CALC
Forcings
CALC
NEXTGEN BMI model configuration

Defines spatial domain

Indices and coverage used to extract catchment averaged forcings

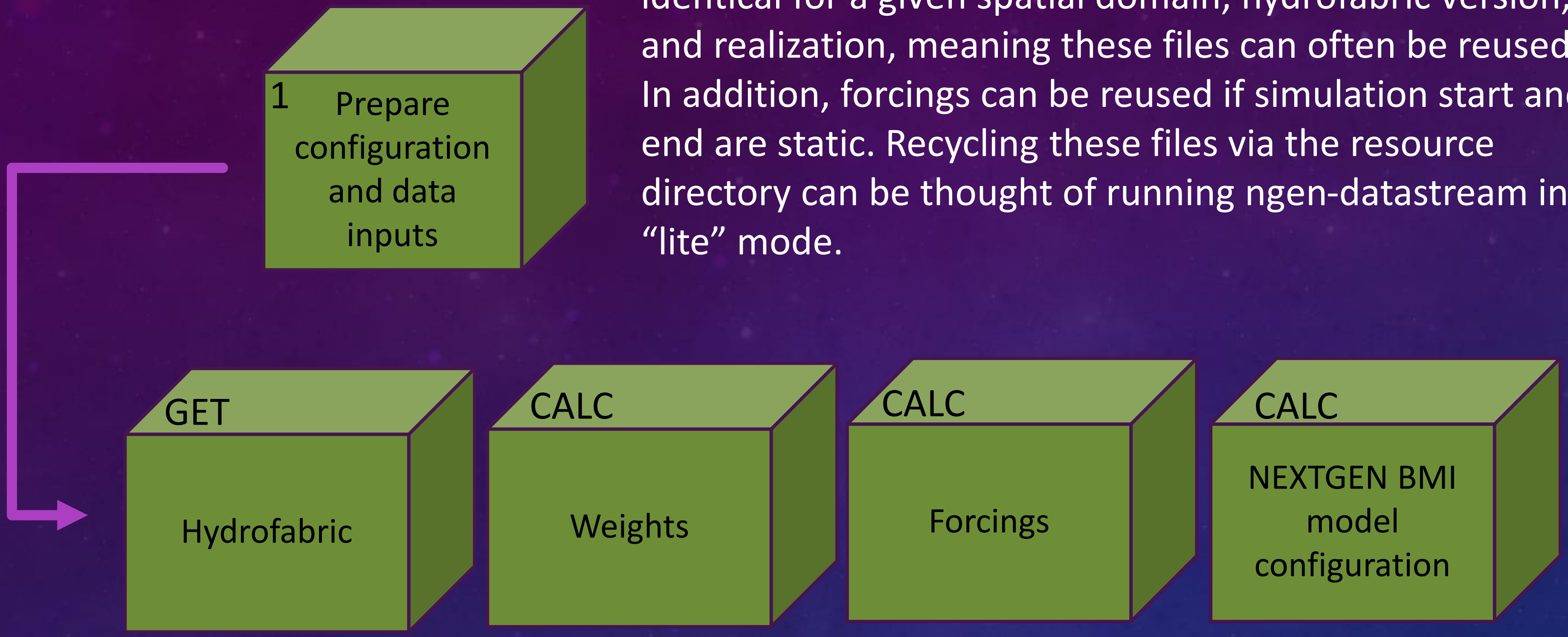
Performs conversion between National Water Model and NextGen forcings formats

Required files for NextGen BMI modules



```
global": {
 "formulations": [
   {
     "params": {
       "modules": [
                   "uses_forcing_file": false
               }
           },
           {
             "name": "bmi_c",
             "params": {
                 "name": "bmi_c",
                 "model_type_name": "PET",
                 "library_file": "/dmod/shared_libs/libpetbmi.so.1.0.0",
                 "forcing_file": "",
                 "init_config": "./config/PET_{{id}}.ini",
                 "allow_exceed_end_time": true,
                 "main_output_variable": "water_potential_evaporation_flux",
                 "registration_function": "register_bmi_pet",
                 "variables_names_map": {
                     "water_potential_evaporation_flux": "potential_evapotranspiration"
                 },
                 "uses_forcing_file": false
             }
         },
           {
               "name": "bmi_c",
               "params": {
                   "name": "bmi_c",
                   "model_type_name": "CFE",
                   "main_output_variable": "Q_OUT",
                   "init_config": "./config/CFE_{{id}}.ini",
                   "allow_exceed_end_time": true,
                   "fixed_time_step": false,
                   "uses_forcing_file": false,
                   "registration_function": "register_bmi_cfe",
                   "variables_names_map": {
                       "water_potential_evaporation_flux": "water_potential_evaporation_flux",
                       "atmosphere_water__liquid_equivalent_precipitation_rate": "QINSUR",
                       "ice_fraction_schaake" : "sloth_ice_fraction_schaake",
                       "ice_fraction_xinanjiang" : "sloth_ice_fraction_xinanjiang",
                       "soil_moisture_profile" : "sloth_soil_moisture_profile"
                   },
                   "library_file": "/dmod/shared_libs/libcfebmi.so.1.0.0"
               }
           }
       ],
       "uses_forcing_file": false
     }
   }
```

Weights and BMI model configuration generation are identical for a given spatial domain, hydrofabric version, and realization, meaning these files can often be reused. In addition, forcings can be reused if simulation start and end are static. Recycling these files via the resource directory can be thought of running ngen-datastream in “lite” mode.


1 Prepare configuration and data inputs
GET
Hydrofabric
CALC
Weights
CALC
Forcings
CALC
NEXTGEN BMI model configuration

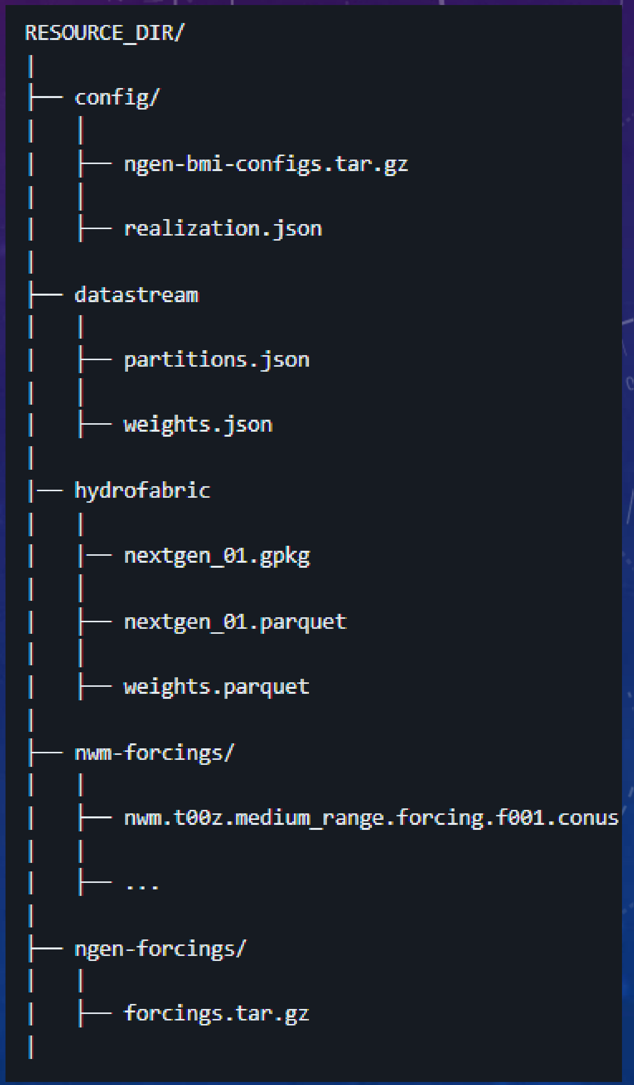
Defines spatial domain

Indices and coverage used to extract catchment averaged forcings

Performs conversion between National Water Model and NextGen forcings formats

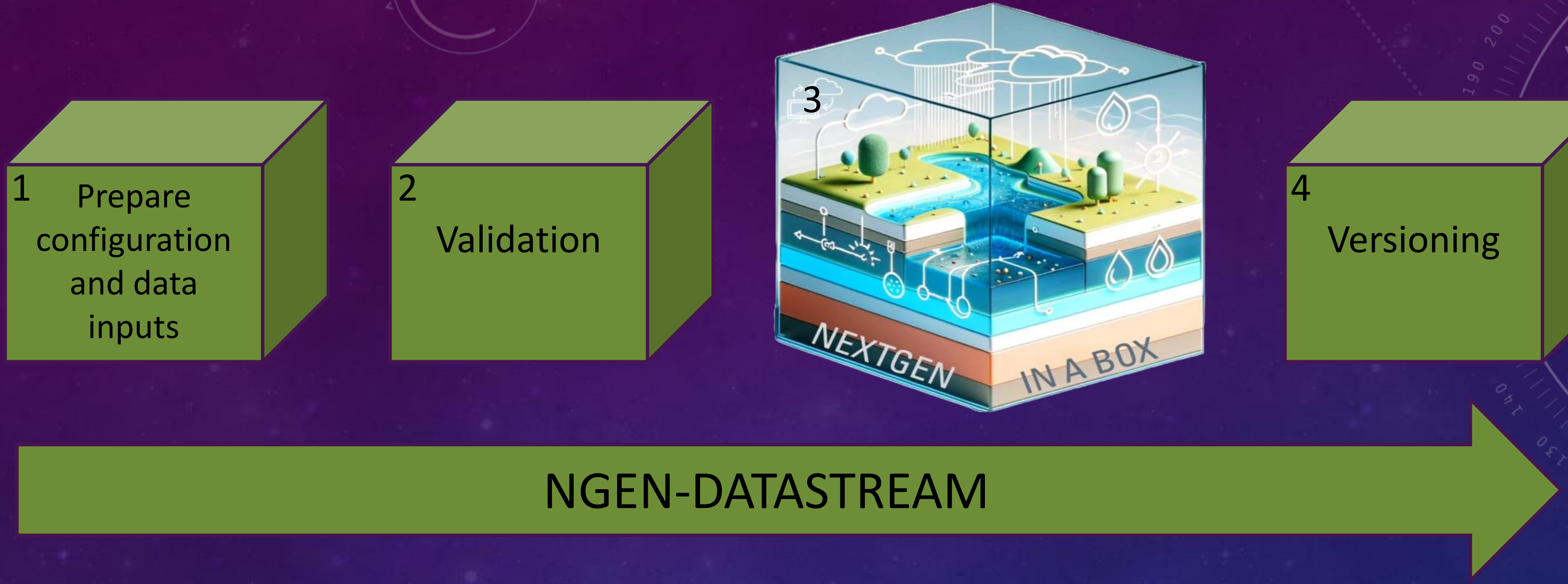
Required files for NextGen BMI modules

***SAVES MONEY***



```
RESOURCE_DIR/
│
├── config/
│   │
│   ├── ngen-bmi-configs.tar.gz
│   │
│   ├── realization.json
│
├── datastream
│   │
│   ├── partitions.json
│   │
│   ├── weights.json
│
|── hydrofabric
│   │
│   |── nextgen_01.gpkg
│   │
│   ├── nextgen_01.parquet
│   │
│   ├── weights.parquet
│
├── nwm-forcings/
│   │
│   ├── nwm.t00z.medium_range.forcing.f001.conus
│   │
│   ├── ...
│
├── ngen-forcings/
│   │
│   ├── forcings.tar.gz
│
```

https://github.com/CIROH-UA/ngen-datastream/tree/main?tab=readme-ov-file#resource_dir-datastream-resources



1 Prepare configuration and data inputs

2 Validation

3

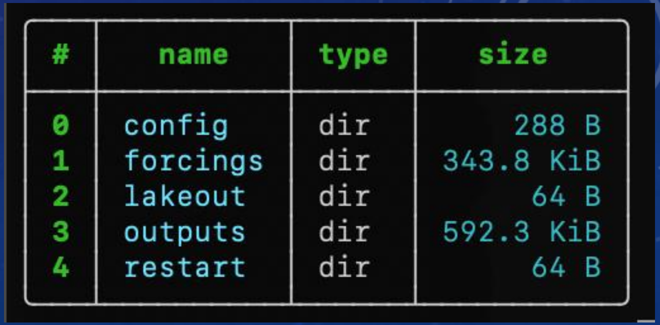
NEXTGEN IN A BOX

4 Versioning

NGEN-DATASTREAM

ngen-datastream refers to the software chain that builds and validates NEXTGEN input packages (ngen-run/), executes NEXTGEN through NEXTGEN In A Box (NGIAB), and versions the entire run for reproducibility.

This enforces a standard folder ngen-run/ which makes validation and versioning possible. A standard run folder also allows for other new tools to easily integrate with NGIAB (e.g., DataPreprocessor).

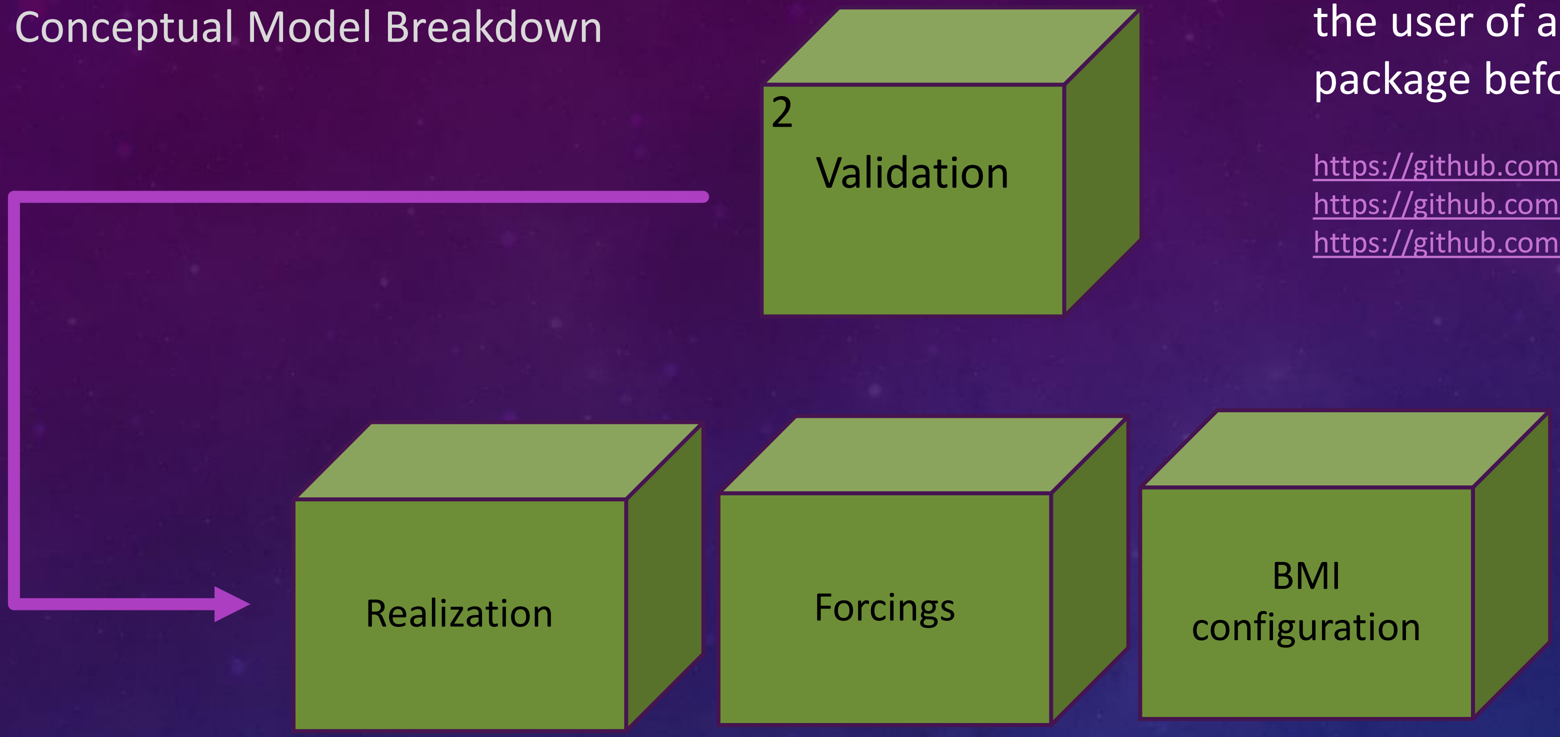


```
 # │   name    │ type │    size
───┼───────────┼──────┼───────────
 0 │ config    │ dir  │     288 B
 1 │ forcings  │ dir  │ 343.8 KiB
 2 │ lakeout   │ dir  │      64 B
 3 │ outputs   │ dir  │ 592.3 KiB
 4 │ restart   │ dir  │      64 B
```

# NEXTGEN Water Modeling Framework Datastream Conceptual Model Breakdown

TLDR: The value of the validation step is to notify the user of any errors in the NEXTGEN run package before beginning the execution

https://github.com/CIROH-UA/ngen-datastream?tab=readme-ov-file#ngen-run
https://github.com/CIROH-UA/ngen-datastream/tree/main/python#run_validatorpy
https://github.com/NOAA-OWP/ngen-cal/tree/master


2 Validation

Realization

Forcings

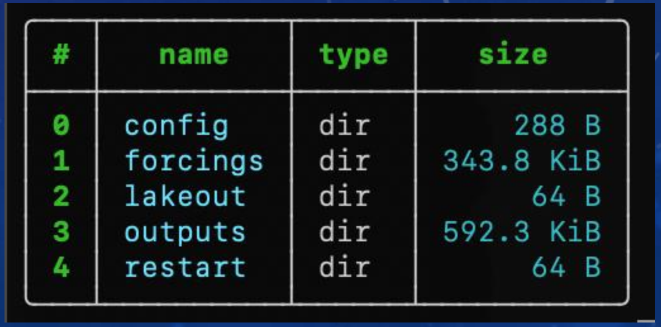
BMI configuration

Required for

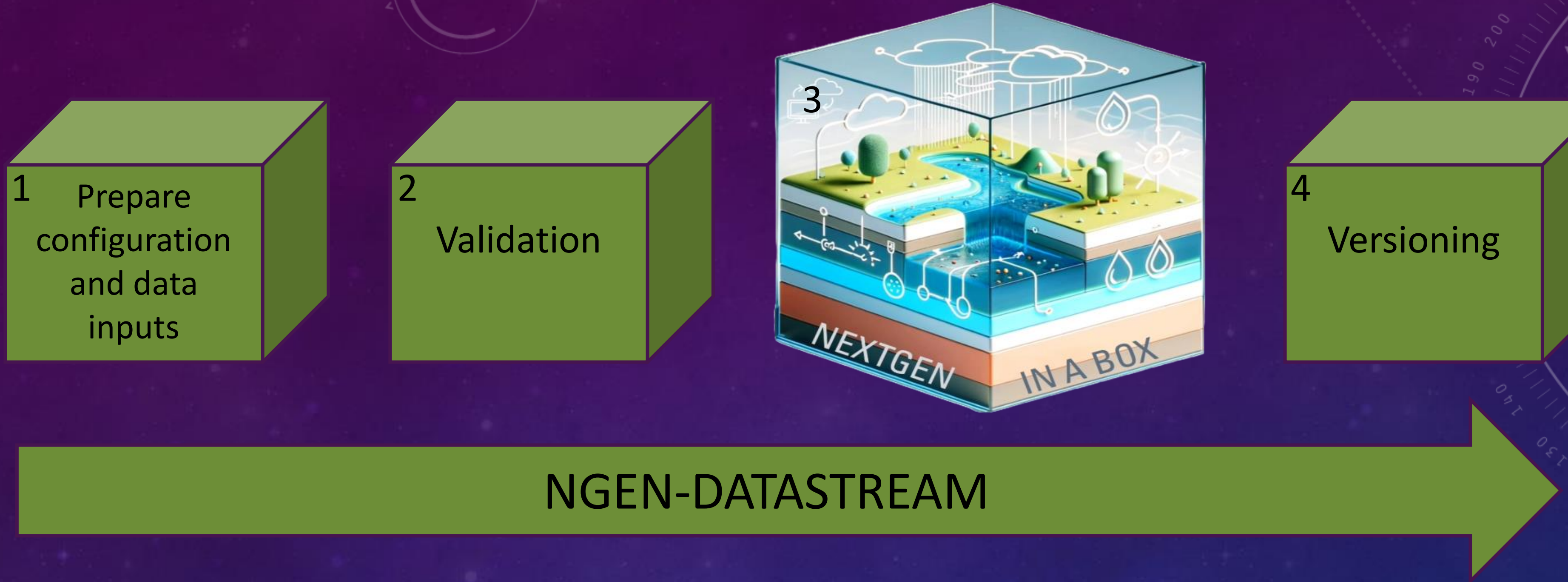
Ensures the user has supplied a valid realization file to configure NEXTGEN

Ensures a forcing file exists for each catchment in the hydrofabric and for each time step specified in the realization

Ensures all BMI model configuration files exist.



```
 #  │    name    │ type │    size
────┼────────────┼──────┼────────────
 0  │ config     │ dir  │     288 B
 1  │ forcings   │ dir  │ 343.8 KiB
 2  │ lakeout    │ dir  │      64 B
 3  │ outputs    │ dir  │ 592.3 KiB
 4  │ restart    │ dir  │      64 B
```



1 Prepare configuration and data inputs

2 Validation

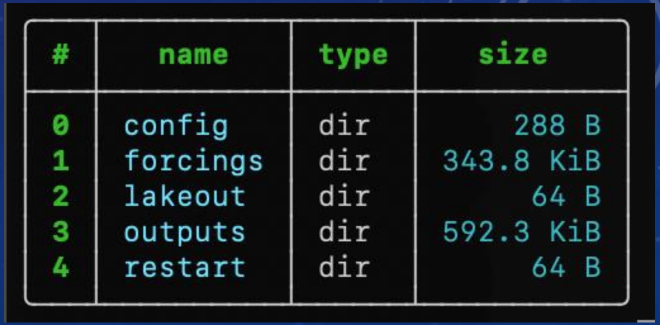
3

4 Versioning

NGEN-DATASTREAM

ngen-datastream refers to the software chain that builds and validates NEXTGEN input packages (ngen-run/), executes NEXTGEN through NEXTGEN In A Box (NGIAB), and versions the entire run for reproducibility.

This enforces a standard folder ngen-run/ which makes validation and versioning possible. A standard run folder also allows for other new tools to easily integrate with NGIAB (e.g., DataPreprocessor).

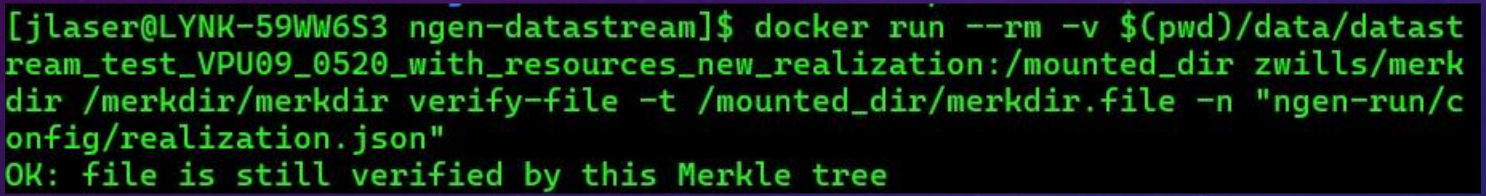


```
 # │   name    │ type │    size
───┼───────────┼──────┼───────────
 0 │ config    │ dir  │     288 B
 1 │ forcings  │ dir  │ 343.8 KiB
 2 │ lakeout   │ dir  │      64 B
 3 │ outputs   │ dir  │ 592.3 KiB
 4 │ restart   │ dir  │      64 B
```

- Merkle Tree based hashing algorithm
  - “Root” hash allows for quickly identifying if two ngen-run directories are different.
  - Ability to query whether some file is a part of the tree represented by the root hash
  - Ability to compare files without opening them

4

Versioning



```
[jlaser@LYNK-59WW6S3 ngen-datastream]$ docker run --rm -v $(pwd)/data/datast
ream_test_VPU09_0520_with_resources_new_realization:/mounted_dir zwills/merk
dir /merkdir/merkdir verify-file -t /mounted_dir/merkdir.file -n "ngen-run/c
onfig/realization.json"
OK: file is still verified by this Merkle tree
```

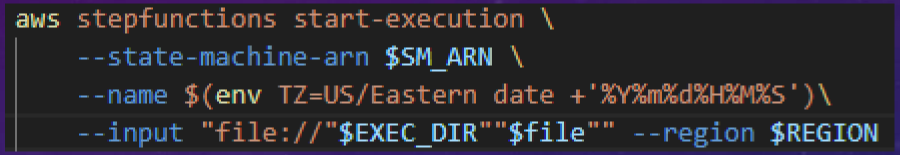
# TECHNICAL BREAKDOWN

- Fundamentally, ngen-datastream is a linux shell script that automates the entire process laid out in the conceptual model. Largely, this is accomplished by moving data between docker containers that handle the necessary computations in pre-configured environments.
    - Portability and scalability.

# CUSTOMIZE NEXTGEN SIMULATION RESOURCES

ngen-datastream allows users to submit NextGen simulation jobs to cloud-based hosts in Amazon Web Services (AWS)

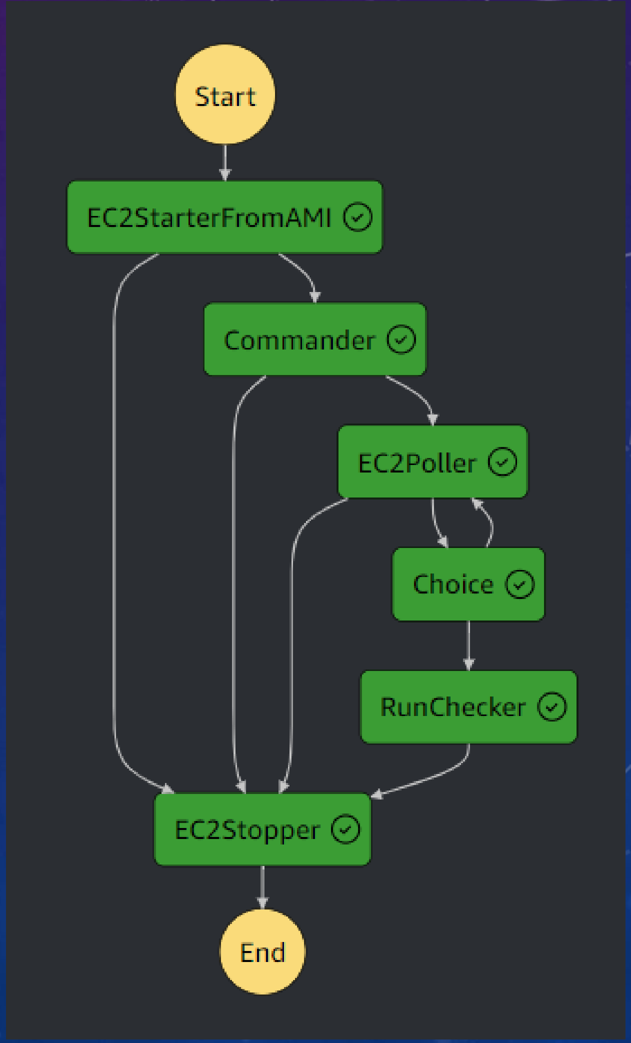
Users submit jobs via a customizable execution json to a generalizable AWS state machine that manages job execution



```
aws stepfunctions start-execution \
    --state-machine-arn $SM_ARN \
    --name $(env TZ=US/Eastern date +'%Y%m%d%H%M%S')\
    --input "file://"$EXEC_DIR""$file"" --region $REGION
```

Customizable parameters in the execution json include:

- Instance Type
- Image Id
- Number of instances
- Volume Size
- Region
- Security Groups / Instance profile (IAM)
- Commands

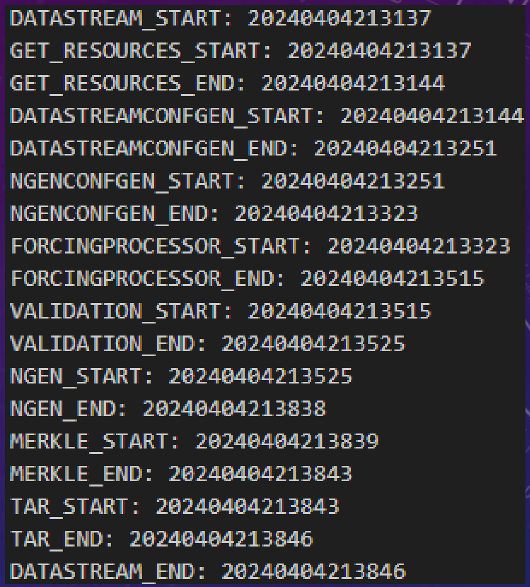


# SCALE TO THE CLOUD

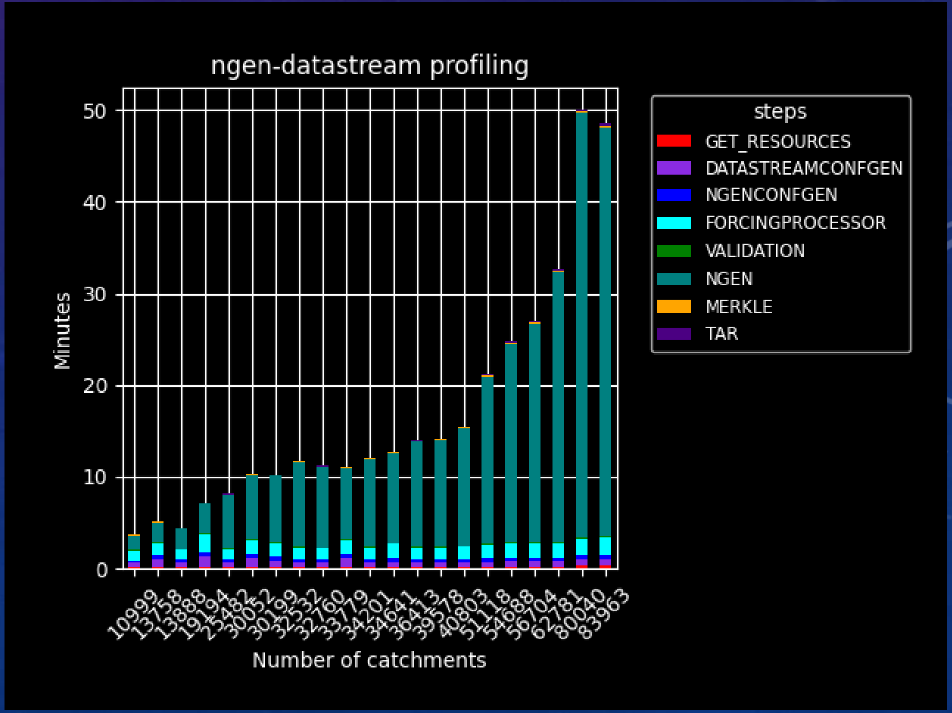
- Split VPU daily run - An application of the ngen-datastream AWS state machine
  - 24-hour CONUS NextGen simulation scheduled each day with AWS EventBridge
  - 22 individual ec2 instances, 1 for forcingprocessor , 21 simultaneously processing for each VPU
  - Runtime determined by the runtime of the largest VPU.

- Takeaways:
  - Terraform allows for quick building of complex AWS infrastructure
  - Allows users to access HPC resources
  - Highly configurable execution file allows users to finely tune resources to their ngen executions

datastream-metadata/profile.txt



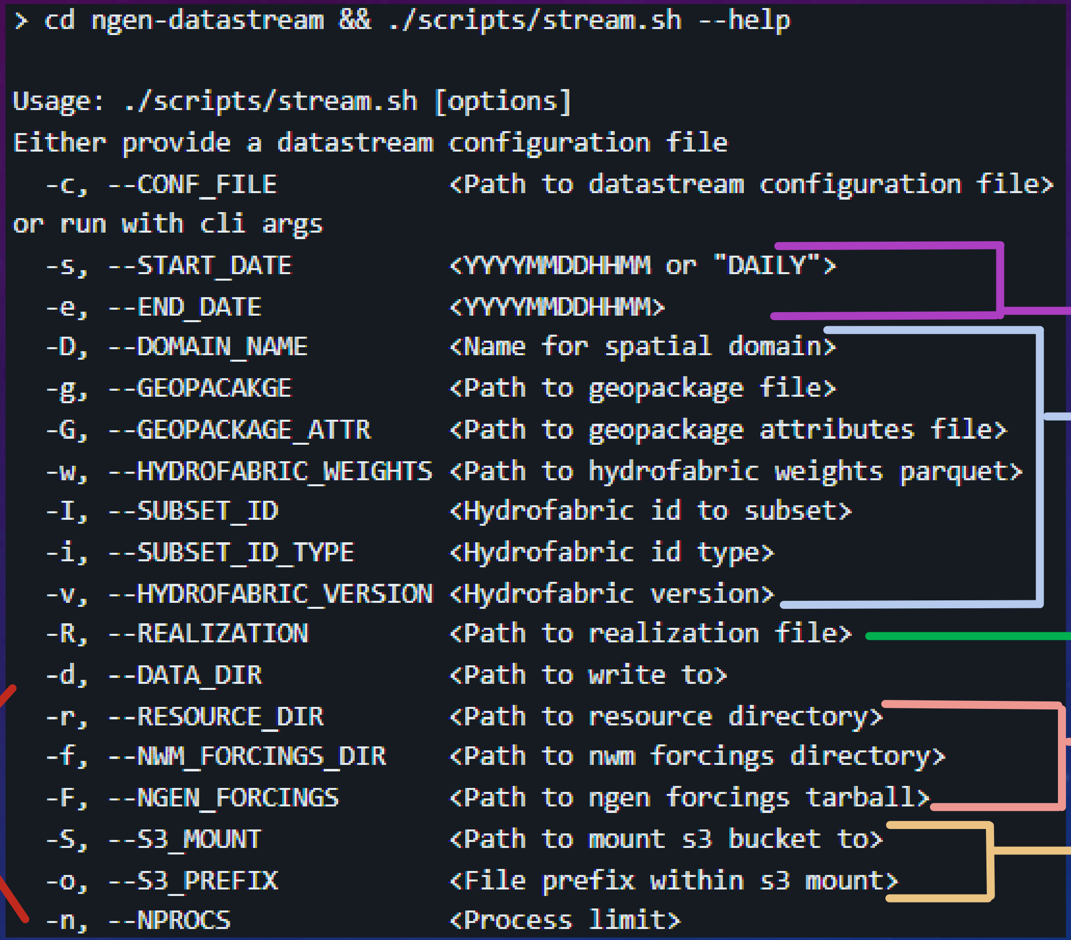
```
DATASTREAM_START: 20240404213137
GET_RESOURCES_START: 20240404213137
GET_RESOURCES_END: 20240404213144
DATASTREAMCONFGEN_START: 20240404213144
DATASTREAMCONFGEN_END: 20240404213251
NGENCONFGEN_START: 20240404213251
NGENCONFGEN_END: 20240404213323
FORCINGPROCESSOR_START: 20240404213323
FORCINGPROCESSOR_END: 20240404213515
VALIDATION_START: 20240404213515
VALIDATION_END: 20240404213525
NGEN_START: 20240404213525
NGEN_END: 20240404213838
MERKLE_START: 20240404213839
MERKLE_END: 20240404213843
TAR_START: 20240404213843
TAR_END: 20240404213846
DATASTREAM_END: 20240404213846
```



# USAGE

- ngen-datastream is a powerful tool designed to utilize compute resources to their maximum extent.
- In general, ngen-datastream memory footprint scales with
  - Number of catchments (size of the spatial domain)
  - Number of time steps (Simulation duration / output_interval)
- If a crash is experienced, either increase the host resources or decrease one or both above dimensions
  - Linux commands to monitor resources
    - Watch memory usage -> `free –h –s2`
    - Watch processes/cpu usage -> `top`
    - Check available processes -> `nprocs`
- https://github.com/CIROH-UA/ngen-datastream/blob/main/USAGE.md

# WORKSHOP: OPTIONS



```
> cd ngen-datastream && ./scripts/stream.sh --help

Usage: ./scripts/stream.sh [options]
Either provide a datastream configuration file
  -c, --CONF_FILE           <Path to datastream configuration file>
or run with cli args
  -s, --START_DATE          <YYYYMMDDHHMM or "DAILY">
  -e, --END_DATE            <YYYYMMDDHHMM>
  -D, --DOMAIN_NAME         <Name for spatial domain>
  -g, --GEOPACAKGE          <Path to geopackage file>
  -G, --GEOPACKAGE_ATTR     <Path to geopackage attributes file>
  -w, --HYDROFABRIC_WEIGHTS <Path to hydrofabric weights parquet>
  -I, --SUBSET_ID           <Hydrofabric id to subset>
  -i, --SUBSET_ID_TYPE      <Hydrofabric id type>
  -v, --HYDROFABRIC_VERSION <Hydrofabric version>
  -R, --REALIZATION         <Path to realization file>
  -d, --DATA_DIR            <Path to write to>
  -r, --RESOURCE_DIR        <Path to resource directory>
  -f, --NWM_FORCINGS_DIR    <Path to nwm forcings directory>
  -F, --NGEN_FORCINGS       <Path to ngen forcings tarball>
  -S, --S3_MOUNT            <Path to mount s3 bucket to>
  -o, --S3_PREFIX           <File prefix within s3 mount>
  -n, --NPROCS              <Process limit>
```

simulation start and end

spatial domain

NextGen configuration

RUN options

ngen-datasteam “lite”

AWS s3 mount

# WORKSHOP

- https://github.com/CIROH-UA/ngen-datastream/blob/main/docs/CIROH_devcon_2024/workshop.md

# TAKE-AWAYS

- Collaborative process! Feel free to contact me with questions
- Let

# FUTURE WORK

- CI/CD
- Feedback/Collaboration with community
- Begin the science!
    - Find regional improvements to NextGen simulations
    - Expand modules ngen-datastream is aware of

# TERMS

- Catchment – geographic area characterized by a single location, a nexus, where all precipitation in the area runs off through. A drainage basin.
- Nexus – the singular point where water flows into or out of a catchment. Often a point along a river.
- Subsetting – To reduce a large geopackage (many catchments) down to a smaller geopackage (fewer catchments) . In effect, this is choosing the domain over which ngen will run.
- Hashing – SHA256 algorithm applied to files to generate a unique id for a file. Useful for preserving and distinguishing unique inputs.
- Validation – Ensuring the ngen input directory data_dir has been constructed properly. Properly meaning that NextGen will not crash and will generate output data.

# ACRONYMS

- NWM – National Water Model
- NGIAB – Next Generation National Water Model in a Box
- NGEN/NEXTGEN - Next Generation National Water Model
- IaC – Infrastructure as code
- VPU – Vector Processing Unit
- CFE – Conceptual Function Equivalent
- PET – Potential Evapotranspiration
- NOM – NOAA-OWP-Modular
- OWP – Office of water prediction
- BMI – Basic Model Interface

# LINKS

- DATASTREAM https://github.com/CIROH-UA/ngen-datastream/tree/main
- FORCINGPROCESSOR https://github.com/CIROH-UA/ngen-datastream/tree/main/forcingprocessor
- REALIZATION GENERATION AND NGEN-RUN FOLDER VALIDATION https://github.com/NOAA-OWP/ngen-cal
- HYDROFABRIC SUBSETTING https://github.com/LynkerIntel/hfsubset
- HASHING/VERSIONING https://github.com/aaraney/ht
- NGIAB https://github.com/CIROH-UA/NGIAB-CloudInfra
- https://docs.ciroh.org/
- https://docs.ciroh.org/docs/products/tools/nextgeninabox/ngiab-intro
- https://github.com/NOAA-OWP/ngen/wiki
- https://mikejohnson51.github.io/hyAggregate/
- https://ciroh.ua.edu/