

Installation Guide (HPC / Admin)

Purpose

This document explains how to install and verify the full Param Shavak ANUGA simulation stack.

This guide is intended for:

- HPC Administrators
- DevOps Engineers
- Infra Setup Engineers

If you only want to run simulations → See RUNNING.md

If you only want to view dashboard → See USER_GUIDE.md

1. Supported Systems

Recommended OS:

- Ubuntu 20.04+
- Debian 11+
- RHEL 8+
- AlmaLinux 8+
- BOSS OS (CDAC HPC environments)

Hardware:

- Multi-core CPU recommended
 - MPI cluster support optional but preferred
-

2. Pre-Installation Requirements

IMPORTANT — Disable Conda Environments

Conda conflicts with system MPI and NetCDF libraries.

Run:

```
conda deactivate
```

Repeat until fully out of conda.

3. Step 0 — Get The Base Repository (FIRST TIME ONLY)

```
git clone https://github.com/anup619/Param-Shavak-Anuga  
  
cd Param-Shavak-Anuga  
git lfs pull
```

Notes:

- `git lfs pull` required only first clone OR if large files missing
- All installation commands assume you are in repo root

4. Air-Gapped / Offline HPC Preparation

If system has no internet access:

Place these inside:

```
opensource_tools/
```

Required archives:

- geoserver-2.28.x-bin.zip
- node-vXX-linux-x64.tar.xz

Installer will auto-detect and extract if present.

5. MPI Environment Note (HPC Only)

Ensure MPI is accessible:

```
module load mpi/openmpi-x86_64
```

Or verify manually:

```
which mpicc  
which mpirun
```

6. Full Installation (Automated)

Run:

```
make setup
```

What `make setup` Does Internally

System Layer

Installs:

- GCC / Build tools
- Python dev stack
- NetCDF headers
- OpenMPI
- Java (for GeoServer)

Python Layer

Installs:

- numpy, scipy, matplotlib
- netcdf4
- meson build tools
- ANUGA dependencies
- mpi4py compiled using system MPI

ANUGA Layer

CMake:

- Clones `anuga_core`
- Installs locally into user Python site packages
- Generates MPI environment script

Tool Layer (Optional)

If archives exist:

- Extract GeoServer locally
- Extract Node locally
- Build `anuga-viewer` if dependencies exist

7. Post Installation Verification

Verify ANUGA + MPI

```
cd build
make test_anuga
```

Expected:

- mpi4py imports successfully
- ANUGA imports successfully
- MPI compiler detected

Verify Tools Environment

```
source build/setup_tools_env.sh
make test_tools
```

Checks:

- Node path
- GeoServer presence
- Viewer path (if built)

8. Environment Scripts (IMPORTANT)

Simulation Environment

Before running simulations:

```
source build/setup_mpi_env.sh
```

Tools Environment

Before running GeoServer / Node / Viewer:

```
source build/setup_tools_env.sh
```

9. Directory Layout After Install

```
├─ build/
│   ├── setup_mpi_env.sh
│   ├── setup_tools_env.sh
│   ├── geoserver_start.sh
│   └── geoserver_stop.sh
└─ anuga_core/
```

```
|— opensource_tools/  
|— mahanadi_test_case/
```

10. Common Installation Issues

MPI Not Found

Load module OR verify OpenMPI install.

mpi4py Build Fails

Usually MPI compiler not visible.

Fix:

```
export MPICC=$(which mpicc)
```

GeoServer Not Starting

Check Java:

```
java -version
```

11. Clean Reinstall

Remove build artifacts:

```
make clean
```

Full clean:

```
make clean-all
```

Then reinstall:

```
make setup
```

Next Step

Once installation is verified → Go to: [RUNNING.md](#)