Center for Spatial Information Science and Systems
George Mason University

CropScape for Nepal Installation Guide

Date Revised: 11 April 2023
Revised by: Gil Heo

Table of Contents

Contents

1. Revisions

Date	Author	Description
11 April 2023	Gil Heo	Initial Document

2. Prerequisites

2.1 System Requirements

The CropScape portals consist of web application and external binaries.

2.1.1 Hardware Requirements

The CropScape portal is running on a web application server. The minimum hardware requirement of the portal is same as the web application server and JVM requirement. Because the minimum requirements are too low, the portal can run any modern PC environment, but for the best performance, multicore processor, sufficient memory, and large capacity of storage are recommended.

This is the test environment for the CropScape portal. Note, it's not a minimum requirement.

- Processor: Dual Xeon(R) Silver 4116 CPU @ 2.10GHz (24 cores, 48 threads)
- Memory: 128 GB
- Storage: 60 TB (minimum required 3 TB)

2.1.2 Software Requirements

To run a web application, a web application server is required.

- Operating System: Linux (ubuntu 18.04 LTS or higher)
- Java version: Java 11 or higher
- HTTP server: Apache HTTP server 2.4.x
- Web Application Server: Apache Tomcat 9.x (recommended)
- Geospatial Data Library: gdal 2.x (or higher)
- Web Downloading: wget 1.19.x (or higher), curl 7.x (or higher)
- Python 3: Python 3.10.x (or higher)

2.2 Apache HTTP Server

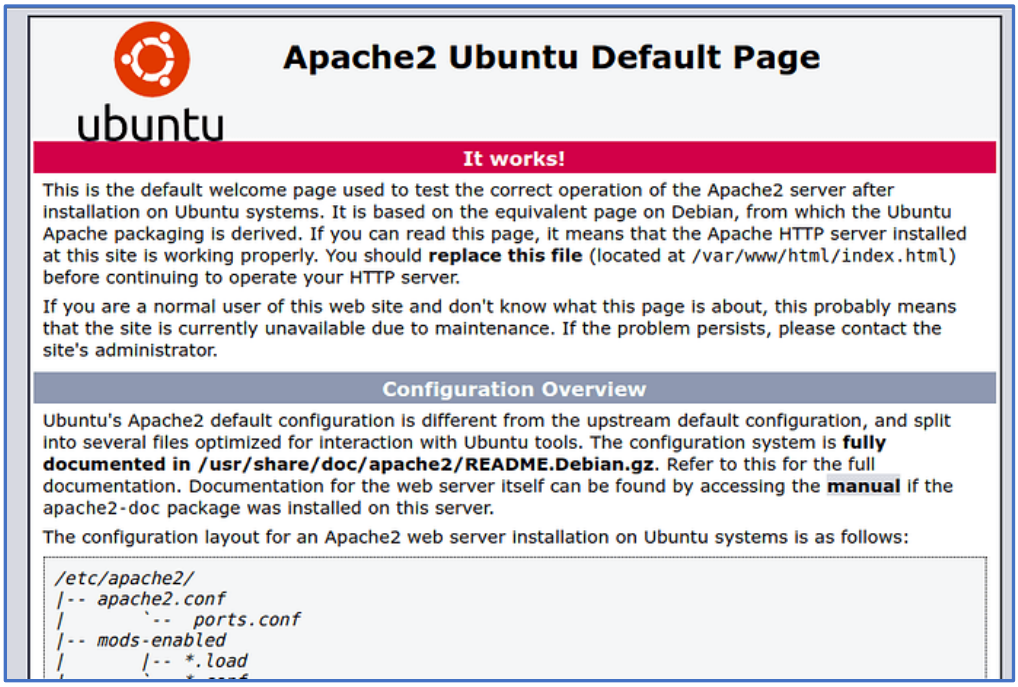
The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient, and extensible server that provides HTTP services in sync with the current HTTP standards.

2.2.1 Installing Apache on Debian/Ubuntu

To install Apache binaries, install the latest meta-package apache2 by running:

```
$ sudo apt update
$ sudo apt install apache2
$ sudo apachectl start
```

After letting the command run, all required packages are installed and we can test it out by typing in our IP address for the web server.



2.2.2 Installing Apache from Source codes

To install Apache from source codes, go to the download section in the official Apache HTTP server website, and download the last version of source codes of the Apache HTTP Server.

```
https://httpd.apache.org/download.cgi
```

2.3 MapServer

MapServer is an Open Source platform for publishing spatial data and interactive mapping applications to the web. Originally developed in the mid-1990's at the University of Minnesota, MapServer is released under an MIT-style license, and runs on all major

platforms (Windows, Linux, Mac OS X). MapServer is not a full-featured GIS system, nor does it aspire to be. See the Site Gallery for live examples of MapServer in action.

2.3.1 Installing MapServer on Debian/Ubuntu

To install MapServer binaries, install the latest cgi-mapserver package and its dependents by running:

```
$ sudo apt install cgi-mapserver
```

Check mapserv executable binary file is installed correctly at the following directory:

```
$ cd /usr/lib/cgi-bin
$ ./mapserv -v
MapServer version 7.6.4 …
```

2.3.2 Installing MapServer from source codes

To install MapServer from source codes, go to the download section in the official MapServer website, and download the last version of source codes, and follow the instructions.

```
https://mapserver.org/download.html
https://mapserver.org/installation/index.html
```

2.4 Apache Tomcat

The Apache Tomcat software is an open source implementation of the Jakarta Servlet, Jakarta Server Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta Authentication specifications. These specifications are part of the Jakarta EE platform.

2.4.1 Installing Apache Tomcat from the official website

To install Apache Tomcat from the official website, go to the download section in the official Apache Tomcat website, and download a binary distribution. The CropScape for Nepal portal is tested on Apache Tomcat 9.x, so we recommended to choose Tomcat 9.x version. The version number may be different from current distribution.

```
https://tomcat.apache.org/download-90.cgi
https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.73/bin/apache-tomcat-9.0.73.tar.gz
```

Extract the downloaded archive package to a directory. Usually, to choose "/usr/local" is a good directory to install the package.

```
$ cd /usr/local
$ sudo tar xvf apache-tomcat-9.0.73.tar.gz
$ cd apache-tomcat-9.0.73
$ ls -l
```

For safer use, we recommend creating a user "tomcat", and execute the Tomcat server as the user.

```
$ sudo adduser tomcat
$ sudo chown -R tomcat.tomcat /usr/local/apache-tomcat-9.0.73
$ cd /usr/local/apache-tomcat-9.0.73/bin
$ sudo -u tomcat ./startup.sh
```

The default port number of the Tomcat server is "8080". If you want to change the number, change the port number you want.

```
$ sudo -u tomcat nano /usr/local/apache-tomcat-9.0.73/conf/server.xml

…
  <Service name="Catalina">
    …
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    …
```

If the Tomcat server does not startup, check Java location and log directory, and access permission, etc. More information, check the contents in the following URL:

```
https://tomcat.apache.org/tomcat-9.0-doc/setup.html
```

2.4.2 Installing Apache Tomcat on Debian/Ubuntu

Especially, Debian/Ubuntu supports Apache Tomcat 9.x as their packages, so you can install the Tomcat like any other package installation.

```
$ sudo apt install tomcat9
```

Check the following directories for the installation:

```
/etc/tomcat9   (environmental settings, …)
/usr/share/tomcat9   (binaries, libraries, …)
/var/log/tomcat9   (logs)
/var/lib/tomcat9   (webapps)
```

Note) If you have a plan to use "tomcat" user account for executing the server, instead of using superuser, we do NOT recommend installing Debian/Ubuntu apt packages.

2.5 Other Requirements

2.5.1 gdal binaries and libraries

"gdal" binaries and libraries packages are essential to run CropScape portal.

```
$ sudo apt install gdal-bin
```

2.5.2 Java runtime

"openjdk" (or similar java runtimes) package are essential. The portal is tested in JDK 11 release.

```
$ sudo apt install openjdk-11-jdk
```

2.5.3 Web downloading tools

"wget" and "curl" tools are using for downloading MODIS data.

```
$ sudo apt install wget curl
```

2.5.4 Python 3

Python version 3 uses backend mapfile generation logic.

```
$ sudo apt install python3
```

3. Installation

3.1 CropScape for Nepal

The CropScape for Nepal portal is a web application, and it runs on web application server, like Apache Tomcat. The portal is tested on Apache Tomcat 9.x.

The portal packages consist of 2 essential packages and 1 optional package.

- The CropScape for Nepal portal binaries (.war, required)
- Product builder (.tar.gz, required)
- NDVI/VCI Products Year 2000 to 2023 (.tar.gz, optional)

3.1.1 Installing the CropScape for Nepal portal binaries

The CropScape for Nepal is a web application, and it distributes as a WAR file. Download and install the Web application ARchive (WAR) file from the following URL (the ending date may be different):

```
https://geobrain.csiss.gmu.edu/Geofairy/CropScape/CropScapeNepal_mm-dd-yyyy.war
```

Go to the "webapps" directory in your Tomcat server, and create a directory named the portal name and extract the WAR file into the directory (the version number may be different). If you install an admin App, install the WAR file via the admin App.

```
$ cd /usr/local/apache-tomcat-9.0.73
$ ./bin/shutdown.sh
$ cd webapps
$ sudo -u tomcat mkdir CropScapeNepal
$ cd CropScapeNepal
$ sudo -u tomcat jar xvf {your_download_directory}/CropScapeNepal_mm-dd-yyyy.war
```

For other two portals, "CropScape for Pakistan" and "CropScape for Bangladesh" are using the same portal binaries, but they are using different configurations. Make two more directories and extract the same binaries into the directories.

```
$ cd /usr/local/apache-tomcat-9.0.73/webapps
$ sudo -u tomcat mkdir CropScapePakistan
$ cd CropScapePakistan
$ sudo -u tomcat jar xvf {your_download_directory}/CropScapeNepal_mm-dd-yyyy.war

$ cd ..
$ sudo -u tomcat mkdir CropScapeBangladesh
$ cd CropScapeBangladesh
$ sudo -u tomcat jar xvf {your_download_directory}/CropScapeNepal_mm-dd-yyyy.war
```

After extracting, set configuration in each portal.

```
$ cd /usr/local/apache-tomcat-9.0.73/webapps

$ cd CropScapeNepal
$ sudo -u tomcat cp js/confglobal_Nepal.js js/confglobal.js
$ sudo -u tomcat cp NDVIDownload_Nepal.properties NDVIDownload.properties

$ cd ../CropScapePakistan
$ sudo -u tomcat cp js/confglobal_Pakistan.js js/confglobal.js
$ sudo -u tomcat cp NDVIDownload_Pakistan.properties NDVIDownload.properties

$ cd ../CropScapeBangladesh
$ sudo -u tomcat cp js/confglobal_Bangladesh.js js/confglobal.js
$ sudo -u tomcat cp NDVIDownload_Bangladesh.properties NDVIDownload.properties
```

3.1.2 Installing product builders

The CropScape portals refer daily, weekly, and biweekly NDVI products, and weekly and biweekly VCI from year 2000 to current year (2023). This product builder package also has important product building status to share with the portal.

Download and install the product builder package from the following URL (the ending date may be different):

```
https://geobrain.csiss.gmu.edu/Geofairy/CropScape/CROPSCAPE_mm-dd-yyyy.tar.gz
```

Choose a directory for installing the package. Wherever can be possible, but the directory should have writable permission by the web application owner account (ex, tomcat). After extracting, create a symbolic link to the root(/) directory.

```
$ cd {your_work_dirctory}
$ tar xvf {your_downlad_directory}/CROPSCAPE_mm-dd-yyyy.tar.gz
$ sudo chown -R tomcat.tomcat CROPSCAPE
$ cd /
$ sudo ln -s {your_work_dirctory}/CROPSCAPE .
```

Register cronjobs for running product building jobs periodically (refer /CROPSCAPE/crontab.cfg file)

Read current cron configuration of the user "tomcat"

```
$ cd /CROPSCAPE
$ sudo crontab -u tomcat -l > tmp
$ sudo nano tmp
```

Append the following lines into the end of "tmp" file

```
# daily job, at 12:00 AM
0 0 * * * /CROPSCAPE/cropscape-daily.sh

# weekly job, at 12:00 AM on Sunday
0 0 * * SUN /CROPSCAPE/cropscape-weekly.sh

# yearly job, on Jan 7
0 0 1 7 * /CROPSCAPE/cropscape-yearly.sh
```

Register new cron jobs to the user "tomcat". Check the jobs are registered correctly.

```
$ sudo crontab -u tomcat < tmp
$ sudo crontab -u tomcat -l
```

3.1.3 Installing CGI programs

The CropScape for Nepal uses its map images and boundary shape features by requesting to installed local MapServer. To respond from the portals requests, the CGI programs in the "/CROPSCAPE/cgi-bin" directory should be placed in your "/cgi-bin" directory.

```
$ cd /usr/lib/cgi-bin
$ sudo cp /CROPSCAPE/cgi-bin/* .
$ ls -l
```

3.1.4 Installing NDVI/VCI Products

Building NDVI and VCI products is a time-consuming job. Each product building job will be done in reasonable time, but all building jobs (daily, weekly, and biweekly NDVI and VCI products since year 2000) at the same time takes a very long time. This package is very helpful to reduce your installation time.

Download and install the product builder package from the following URL (the ending date may be different). Each package separated by year is all products in given year of the three portals.

```
https://geobrain.csiss.gmu.edu/Geofairy/CropScape/CROPSCAPE_products_year_mm-dd-yyyy.tar  (about 100GB per year)

ex)
CROPSCAPE_products_2023_mm-dd-yyyy.tar    <-- for year 2023
CROPSCAPE_products_2022_mm-dd-yyyy.tar    <-- for year 2022
…
CROPSCAPE_products_2000_mm-dd-yyyy.tar    <-- for year 2000
```

Go to CROPSCAPE directory and extract each year products into the directory.

```
$ cd /CROPSCAPE

$ sudo -u tomcat tar xvf {your_downlad_directory}/CROPSCAPE_products_2023_mm-dd-yyyy.tar
$ sudo -u tomcat tar xvf {your_downlad_directory}/CROPSCAPE_products_2022_mm-dd-yyyy.tar
…
```

3.1.5 Starting the portal

All installation steps are finished, now be ready to start the three portals. Start up the Apache Tomcat server.

```
$ sudo -u tomcat /usr/local/apache-tomcat-9.0.73/bin/startup.sh
```

Open a web browser, and check the three portals are running:

```
http://localhost:8080/CropScapeNepal
http://localhost:8080/CropScapePakistan
http://localhost:8080/CropScapeBangladesh
```

3.2 Apache HTTP Server settings

Hiding the Tomcat server to the outside world and avoiding cross-origin resource sharing (CORS) restriction, Apache HTTP server should become a front-end web access gateway of the CropScape portal. For making Apache HTTP server into a gateway, two server settings are required.

3.2.1 Reverse Proxy settings

These settings are for accessing the CropScape portal without port number.

Step 1) Enabling Necessary Apache modules

The modules you need are "mod_proxy" itself and several of its add-on modules, which extend its functionality to support different network protocols. Especially, you will use:

- *mod_proxy*, the main proxy module Apache module for redirecting connections; it allows Apache to act as a gateway to the underlying application servers.
- *mod_proxy_http*, which adds support for proxying HTTP connections.
- *mod_proxy_balancer* and *mod_lbmethod_byrequests*, which add load balancing features for multiple backend servers.

To enable these four modules, execute the following commands in succession:

```
$ sudo a2enmod proxy
$ sudo a2enmod proxy_http
$ sudo a2enmod proxy_balancer
$ sudo a2enmod lbmethod_byrequests
```

Step 2) Modifying the Default Configuration to Enable Reverse Proxy

Set "ProxyPass" and "ProxyPassReverse" for the three portals: CropScape for Nepal, CropScape for Pakistan, and CropScape for Bangladesh. Open the "000-default.conf" and put the following settings.

```
$ sudo nano /etc/apache2/sites-available/000-default.conf


<VirtualHost *:80>

        …
        ProxyPass /CropScapeNepal http://localhost:8080/CropScapeNepal
        ProxyPassReverse /CropScapeNepal http://localhost:8080/CropScapeNepal

        ProxyPass /CropScapePakistan http://localhost:8080/CropScapePakistan
        ProxyPassReverse /CropScapePakistan http://localhost:8080/CropScapePakistan

        ProxyPass /CropScapeBangladesh http://localhost:8080/CropScapeBangladesh
        ProxyPassReverse /CropScapeBangladesh http://localhost:8080/CropScapeBangladesh
        …
</VirtualHost>
```

3.2.2 Alias settings

These settings are for downloading image file user requested created by the portal. These image files are in local directories, so the files in the directories should be accessible by Apache HTTP server.

Enable "mod_alias", and set "Alias" and "<Directory>…</Directory>" for the three portals: CropScape for Nepal, CropScape for Pakistan, and CropScape for Bangladesh. Open the "000-default.conf" and put the following settings.

```
$ sudo a2enmod alias
$ sudo nano /etc/apache2/mods-available/alias.conf


<IfModule alias_module>

    …
    Alias /nepal_nass_data_cache/ "/CROPSCAPE/NEPAL/NDVI_DATA_CACHE/"
    <Directory "/CROPSCAPE/NEPAL/NDVI_DATA_CACHE">
        Options FollowSymlinks MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    Alias /pakistan_nass_data_cache/ "/CROPSCAPE/PAKISTAN/NDVI_DATA_CACHE/"
    <Directory "/CROPSCAPE/PAKISTAN/NDVI_DATA_CACHE">
        Options FollowSymlinks MultiViews
        AllowOverride None
        Require all granted
    </Directory>


    Alias /bangladesh_nass_data_cache/ "/CROPSCAPE/BANGLADESH/NDVI_DATA_CACHE/"
    <Directory "/CROPSCAPE/BANGLADESH/NDVI_DATA_CACHE">
        Options FollowSymlinks MultiViews
        AllowOverride None
        Require all granted
    </Directory>
    ….

</IfModule>
```

3.2.3 Applying new settings

After finishing the HTTP server settings, restart your Apache HTTP server.

```
$ sudo service apache2 restart
```

Check the portals can be accessible via the following URLs:

```
http://localhost/CropScapeNepal
http://localhost/CropScapePakistan
http://localhost/CropScapeBangladesh
```

3.3 Other settings

3.3.1 EPSG codes

Check required EPSG codes (EPSG:32645, EPSG:32642, and EPSG:3106) are defined in your server.

```
$ gdalsrsinfo EPSG:32645
$ gdalsrsinfo EPSG:32642
$ gdalsrsinfo EPSG:3106
```

If they do not exist, append them to your "epsg" file.

```
$ sudo nano /usr/share/proj/epsg

# WGS 84 / UTM zone 45N
<32645> +proj=utm +zone=45 +datum=WGS84 +units=m +no_defs  <>

# WGS 84 / UTM zone 42N
<32642> +proj=utm +zone=42 +datum=WGS84 +units=m +no_defs  <>

# Gulshan 303 / Bangladesh Transverse Mercator
<3106> +proj=tmerc +lat_0=0 +lon_0=90 +k=0.9996 +x_0=500000 +y_0=0 +a=6377276.345
+b=6356075.41314024 +towgs84=283.7,735.9,261.1,0,0,0,0 +units=m +no_defs  <>
```

3.3.2 Native binary builds

Some external core builder binaries are written in C/C++ languages, and they are in the package distribution with their source codes. If the binaries are working fine, you don't need to do anything. But the binaries invoke an error, you should build the binaries in your PC. Most of case, the errors happen because of nonexistence of required shared libraries.

There are 2 native binary tools in each portal of the three portals.

- NDVI/VCI builder
- Statistics Chart

Check if it is running (in case of Nepal). If the command works fine, you don't need to do anything.

```
$ cd /CROPSCAPE/NEPAL/programs
$ ./ndviall.x
```

Install shared and development libraries first.

```
$ sudo apt install libgdal-dev, libgeotiff-dev, libhdfeos-dev, libgctp-dev, libjpeg-dev, libgsl-dev, libhdf4-dev
```

Check again that the test program is executable. If you meet an error something like "shared library not found", many cases, it can be solved that you just install the required shared library.

If you can't solve the issue, you can build binaries from source code for your system. If there are no C/C++ build tools in your system, please install them first.

```
$ sudo apt install build-essential
$ gcc –version
```

Build all binaries from their C source code for your system. If you have any problems, please contact the CropScape development team.

```
$ sudo -u tomcat ./mycc.sh cjulian2date
$ sudo -u tomcat ./mycc.sh cndvi
$ sudo -u tomcat ./mycc.sh createRecordPeriods3
$ sudo -u tomcat ./mycc.sh generatePeriod
$ sudo -u tomcat ./mycc.sh generateWeek
$ sudo -u tomcat ./mycc.sh genmask
$ sudo -u tomcat ./mycc.sh getdays2years
$ sudo -u tomcat ./mycc.sh getdays
$ sudo -u tomcat ./mycc.sh gtiffutil
$ sudo -u tomcat ./mycc.sh ndviall
$ sudo -u tomcat ./mycc.sh periods2
$ sudo -u tomcat ./mycc.sh periods3
$ sudo -u tomcat ./mycc.sh periods
$ sudo -u tomcat ./mycc.sh vci_improved
$ sudo -u tomcat ./mycc.sh vciminmax
$ sudo -u tomcat ./mycc.sh vciperiodminmax
```

Check the statistics chart binaries. Go to the following directory, and check the following tool is working without any problem.

```
$ cd /CROPSCAPE/NEPAL/javaprogs/cpp
$ ./retrieveNDVI
```

If you meet an error, install the required shared library first. All of your trials are failed, build binaries for your system. If you have any problems, please contact the CropScape development team.

```
$ sudo -u tomcat ./calMake
$ sudo -u tomcat ./makeapp
```

4. Post-installation Steps

4.1 Building NDVI/VCI products

This is for building NDVI/VCI products manually.

4.1.1 Daily NDVI products

A building job for daily NDVI products can be executable the following commands (in case of Nepal).

```
$ cd /CROPSCAPE/NEPAL/bin/updatedailyndvi_dynamic
$ sudo -u tomcat ./processall.sh {start_date} {end_date}

ex) for building on March 1, 2023
$ sudo -u tomcat ./processall.sh 2023.03.01 2023.03.02
```

4.1.2 Weekly NDVI products

A building job for weekly NDVI products can be executable the following commands (in case of Nepal).

```
$ cd /CROPSCAPE/NEPAL/bin/updateweeklyndvi_dynamic
$ sudo -u tomcat ./processall.sh {start_year} {end_year}

ex) for building in 2023
$ sudo -u tomcat ./processall.sh 2023 2023
```

4.1.3 Biweekly NDVI products

A building job for biweekly NDVI products can be executable the following commands (in case of Nepal).

```
$ cd /CROPSCAPE/NEPAL/bin/updatebiweeklyndvi_dynamic
$ sudo -u tomcat ./processall.sh {start_year} {end_year}

ex) for building in 2023
$ sudo -u tomcat ./processall.sh 2023 2023
```

4.1.4 Weekly VCI products

A building job for weekly VCI products can be executable the following commands (in case of Nepal).

```
$ cd /CROPSCAPE/NEPAL/bin/updateweeklyvci_dynamic
$ sudo -u tomcat ./processall.sh {start_year} {end_year}

ex) for building in 2023
$ sudo -u tomcat ./processall.sh 2023 2023
```

4.1.5 Biweekly VCI products

A building job for biweekly VCI products can be executable the following commands (in case of Nepal).

```
$ cd /CROPSCAPE/NEPAL/bin/updatebiweeklyvci_dynamic
$ sudo -u tomcat ./processall.sh {start_year} {end_year}

ex) for building in 2023
$ sudo -u tomcat ./processall.sh 2023 2023
```

4.1.6 Yearly Min/Max NDVI products

A building job is required at the beginning dates of every new year. Because VCI is derived from periodic minimum and maximum NDVI values. Every new year, all Min/Max NDVI products should be updated for calculating the new year.

Building for Min/Max NDVI products can be executable the following commands (in case of Nepal).

```
$ cd /CROPSCAPE/NEPAL/bin/updateyearlyminmaxweekly_dynamic
$ sudo -u tomcat ./processall.sh {start_year} {end_year}

$ cd /CROPSCAPE/NEPAL/bin/updateyearlyminmaxbiweekly_dynamic
$ sudo -u tomcat ./processall.sh {start_year} {end_year}

ex) for building in 2023
$ sudo -u tomcat ./processall.sh 2023 2023
```

4.2 Managing rice map layers

Rice map image layers of each portal can be configurable by the portal administrator editing the rice map configuration file.

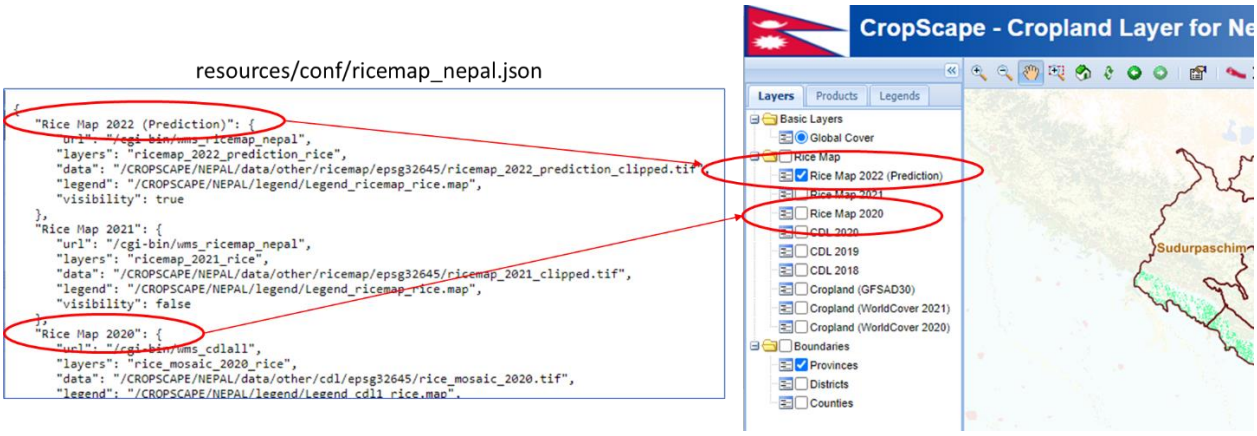


Figure. Relationship between configuration and the rice map layers in the portal

4.2.1 Editing rice map configuration

The rice map configuration file is in the "resources/conf" directory in each country (NEPAL, PAKISTAN, or BANGLADESH) directory.

```
$ cd /CROPSCAPE/NEPAL/resources/conf/ricemap_nepal.json
$ sudo -u tomcat vi ricemap_nepal.json

$ cd /CROPSCAPE/PAKISTAN/resources/conf/ricemap_pakistan.json
$ sudo -u tomcat vi ricemap_pakistan.json

$ cd /CROPSCAPE/BANGLADESH/resources/conf/ricemap_bangladesh.json
$ sudo -u tomcat vi ricemap_bangladesh.json
```

The file format of the configuration is JSON. Note, string terms must be enclosed in double quotation marks. The name of rice map layer object will become the title. The following properties are required in an object:

Properties	Description	Example
"url"	WMS service endoint	"/cgi-bin/wms_ricemap_nepal"
"layers"	WMS layer name	"ricemap_2022_prediction_rice"
"data"	Physical data file location	"/CROPSCAPE/NEPAL/data/other/ricemap/ epsg32645/ricemap_2022_prediction_clipped.tif"
"legend"	mapfile location which contains legend range values	"/CROPSCAPE/NEPAL/legend/ Legend_ricemap_rice.map"

"visibility"	Default layer visibility	true

The following is the front part of rice map configuration file of Nepal. You can add, remove, or change layer title or its properties. After editing, you can get updated rice map layers just by reloading the portal website in your web browser.

```
{
  "Rice Map 2022 (Prediction)": {
    "url": "/cgi-bin/wms_ricemap_nepal",
    "layers": "ricemap_2022_prediction_rice",
    "data":
"/CROPSCAPE/NEPAL/data/other/ricemap/epsg32645/ricemap_2022_prediction_clipped.tif",
    "legend": "/CROPSCAPE/NEPAL/legend/Legend_ricemap_rice.map",
    "visibility": true
  },
  "Rice Map 2021": {
    "url": "/cgi-bin/wms_ricemap_nepal",
    "layers": "ricemap_2021_rice",
    "data": "/CROPSCAPE/NEPAL/data/other/ricemap/epsg32645/ricemap_2021_clipped.tif",
    "legend": "/CROPSCAPE/NEPAL/legend/Legend_ricemap_rice.map",
    "visibility": false
  },
  "Rice Map 2020": {
    "url": "/cgi-bin/wms_cdlall",
    "layers": "rice_mosaic_2020_rice",
    "data": "/CROPSCAPE/NEPAL/data/other/cdl/epsg32645/rice_mosaic_2020.tif",
    "legend": "/CROPSCAPE/NEPAL/legend/Legend_cdl1_rice.map",
    "visibility": false
  },
  "CDL 2020": {
    "url": "/cgi-bin/wms_cdlall",
    "layers": "cdl_2020_rice",
    "data": "/CROPSCAPE/NEPAL/data/other/cdl/epsg32645/nepal2020_clipped.tif",
    "legend": "/CROPSCAPE/NEPAL/legend/Legend_cdl1_rice.map",
    "visibility": false
  },
  ...
}
```

4.2.2 Rebuilding mask mapfiles

If you change any rice map configuration, you should rebuild all mask mapfiles for supporting layer masking function in the portal correctly.

The following commands are for rebuilding all mask mapfiles. The mask mapfile generation tool, "build_rice_mapfile_v2.py", creates a mask mapfile from the original mapfile. The tool also accepts a batch job if inputting mapfile list. So, the first step of rebuilding all mask mapfiles, creating a mapfile list which contains all required rebuilding mapfiles. Please, don't forget to add your rice map configuration file with the following "-m" parameter.

```
$ cd /CROPSCAPE/NEPAL/bin/tools

$ sudo -u tomcat find /CROPSCAPE/NEPAL/data/products -name “*[0-9].map” > mapfile.lst

$ sudo -u tomcat ./build_rice_mapfile_v2.py -l mapfile.lst -m
/CROPSCAPE/NEPAL/resources/conf/ricemap_nepal.json
```