

# Zarr Format

[blocked URL](#)

## Zarr Overview:

Zarr is a relatively new cloud-based data format specifically for improving access to N-Dimensional arrays. It is an effective way to store large N-dimensional data in the cloud and access the data in predefined chunks. Zarr can be viewed as the cloud based version of HDF5/[NetCDF](#) files as it follows a similar data model. Zarr does not come in a single file as [NetCDF](#) or [HDF5](#) does but as a directory with chunks of data in compressed binary files and metadata describing the binary content in external JSON files.

The semantic mapping from the [NetCDF Data Model](#) to Zarr Data Model is as follows:

NetCDF Data Model	Zarr V2 Data Model
File	Store
Group	Group
Variable	Array
Attribute	User Attribute
Dimension	Not supported natively. (See 4.1)

A Zarr array can be stored in any storage system that provides a key/value interface, where a key is an ASCII string and a value is an arbitrary sequence of bytes, and the supported operations are read (get the sequence of bytes associated with a given key), write (set the sequence of bytes associated with a given key) and delete (remove a key/value pair).

Zarr V2 is the current stable version, Zarr V3 is considered experimental.

## GeoZarr extension:

Zarr is a generic data format for Scientific Array Data. Groups have found ways to store spatial metadata within the existing structure, however since this is not a single standardized method, users would have to create custom tools to retrieve those fields. There is a [GeoZarr](#) Specification to provide a geospatial extension to the Zarr data format. GeoZarr can store multidimensional georeferenced earth observation data.

## Zarr file organization:

With Zarr, Multiple arrays can be organized in hierarchies of groups. These arrays have the metadata as well as the actual chunked data. The .zarray describes how these binaries are encoded by giving unique key representations.

[blocked URL](#)

## Zarr data format support:

Creating and using Zarr data files at this time appears to be most well supported using one of these methods:

- 1.) The primary implementation of Zarr format is the Python implementation (<https://zarr.readthedocs.io/en/stable/>),
- 2.) A Java implementation also is available (<https://jzarr.readthedocs.io/en/latest/>).
- 3.) There is also support within the gdal libraries: <https://gdal.org/drivers/raster/zarr.html>
- 4.) The R statistics package can also read Zarr files: <https://r-spatial.org/r/2022/09/13/zarr.html>
- 5.) OpenLayers is working on a prototype for adding support: <https://github.com/spacebel/geozarr-openlayers>

## Community Standards:

The Open Geospatial Consortium approved Zarr V2 as a community standard (<https://www.ogc.org/standards/community>) in June 2022. The documentation can be view at: <https://portal.ogc.org/files/100727>

## Available Zarr data sets: