

# CKAN

Installation Guide (v1.0)

PROMPT

# Contents

I

## Installing CKAN

1. Installing Git
2. Installing Docker
3. Setting up with Docker Compose

II

## Adding Extensions

1. Harvest
2. DCAT
3. Preview Extensions
4. Report

III

## Localizing

1. Adding ICU Tokenizer

# I

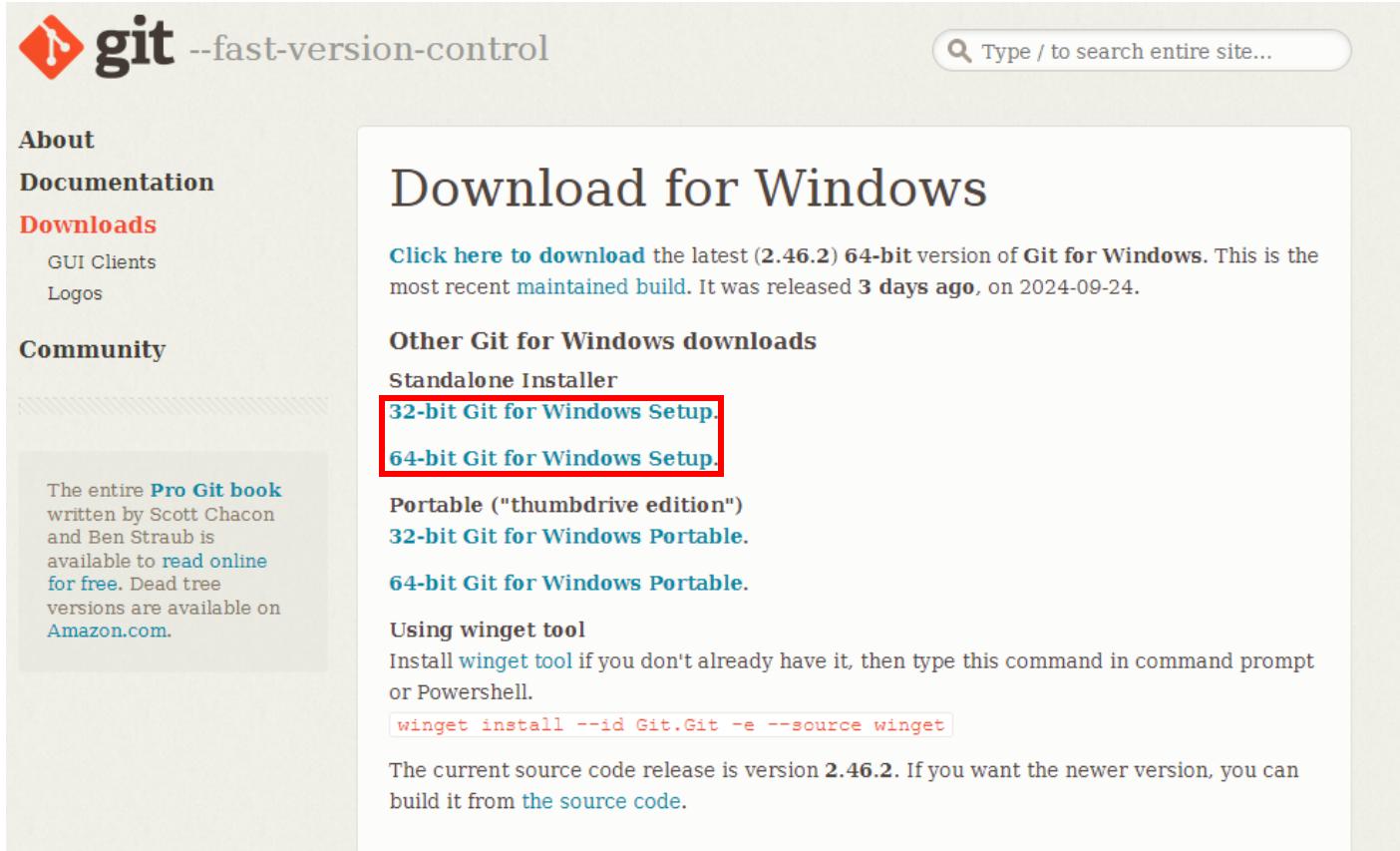
## Installing CKAN

1. Installing Git
  1. Windows
  2. Mac OS
  3. Linux (Debian/Ubuntu)
2. Installing Docker
  1. Windows
  2. Mac OS
  3. Linux (Debian/Ubuntu)
3. Setting up with Docker Compose



### Git for Windows

Browse to Git official website and download Git installer for Windows



The screenshot shows the official Git website's download page for Windows. The top navigation bar includes links for "About", "Documentation", "Downloads" (which is highlighted in red), and "Community". The main content area is titled "Download for Windows". It features a prominent call-to-action button: "Click here to download the latest (2.46.2) 64-bit version of Git for Windows." Below this, there are sections for "Other Git for Windows downloads" (including "Standalone Installer", "32-bit Git for Windows Setup.", and "64-bit Git for Windows Setup.", the latter of which is also highlighted in red). Further down, there are links for "Portable ("thumbdrive edition")" and "64-bit Git for Windows Portable.". A section for "Using winget tool" provides instructions on how to install it via command prompt or Powershell, with a code snippet: "winget install --id Git.Git -e --source winget". At the bottom, it notes the current source code release is version 2.46.2 and provides a link to "the source code".

<https://git-scm.com/downloads/win>



### Git for Windows

Execute the installer and click Next button



## I

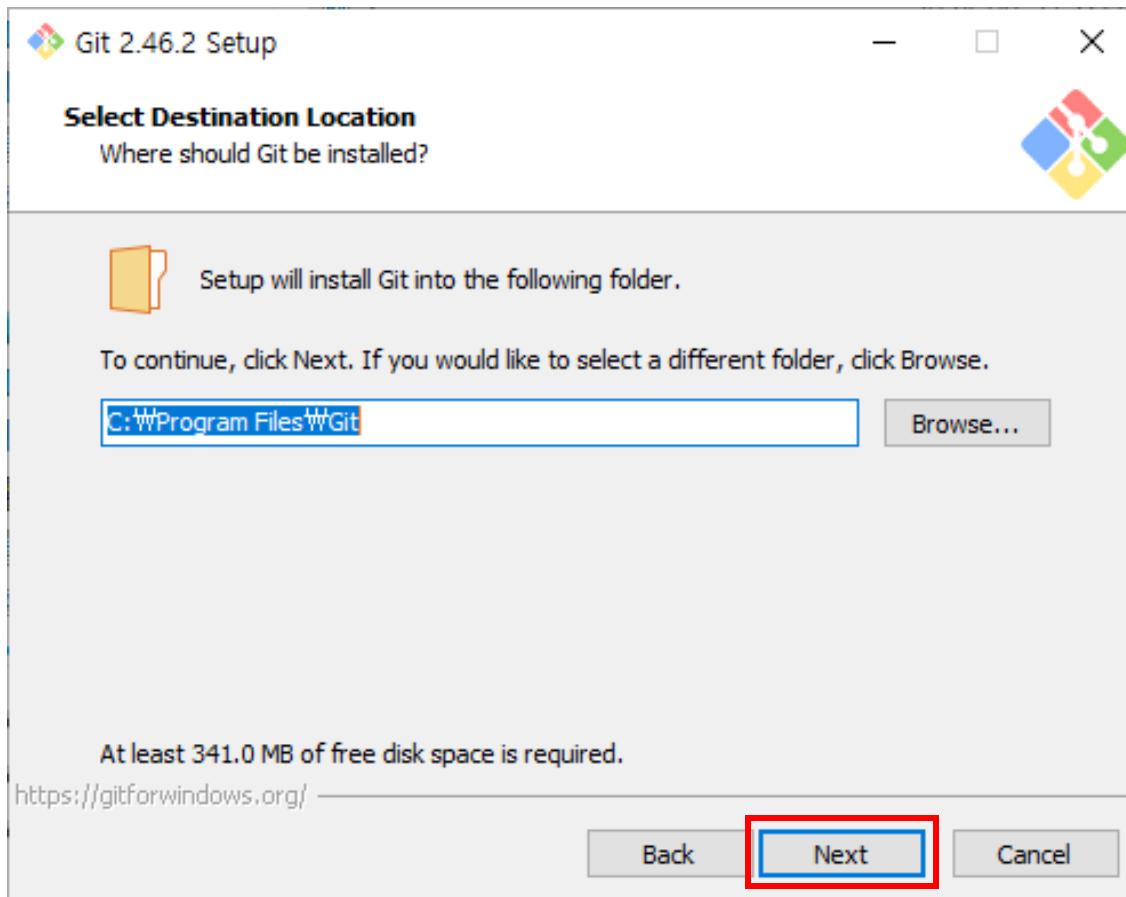
# Installing CKAN

## Installing Git



### Git for Windows

Click Next button

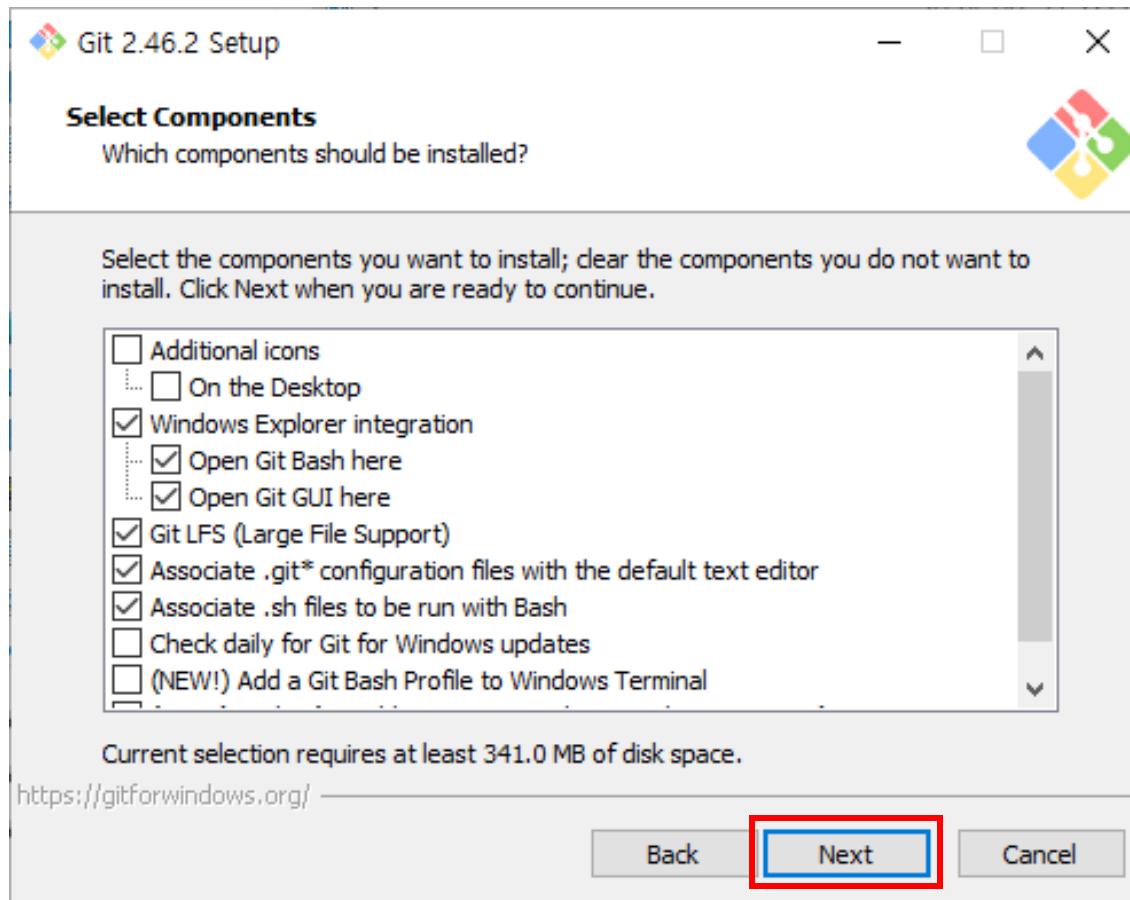


## Installing Git



### Git for Windows

Click Next button



## I

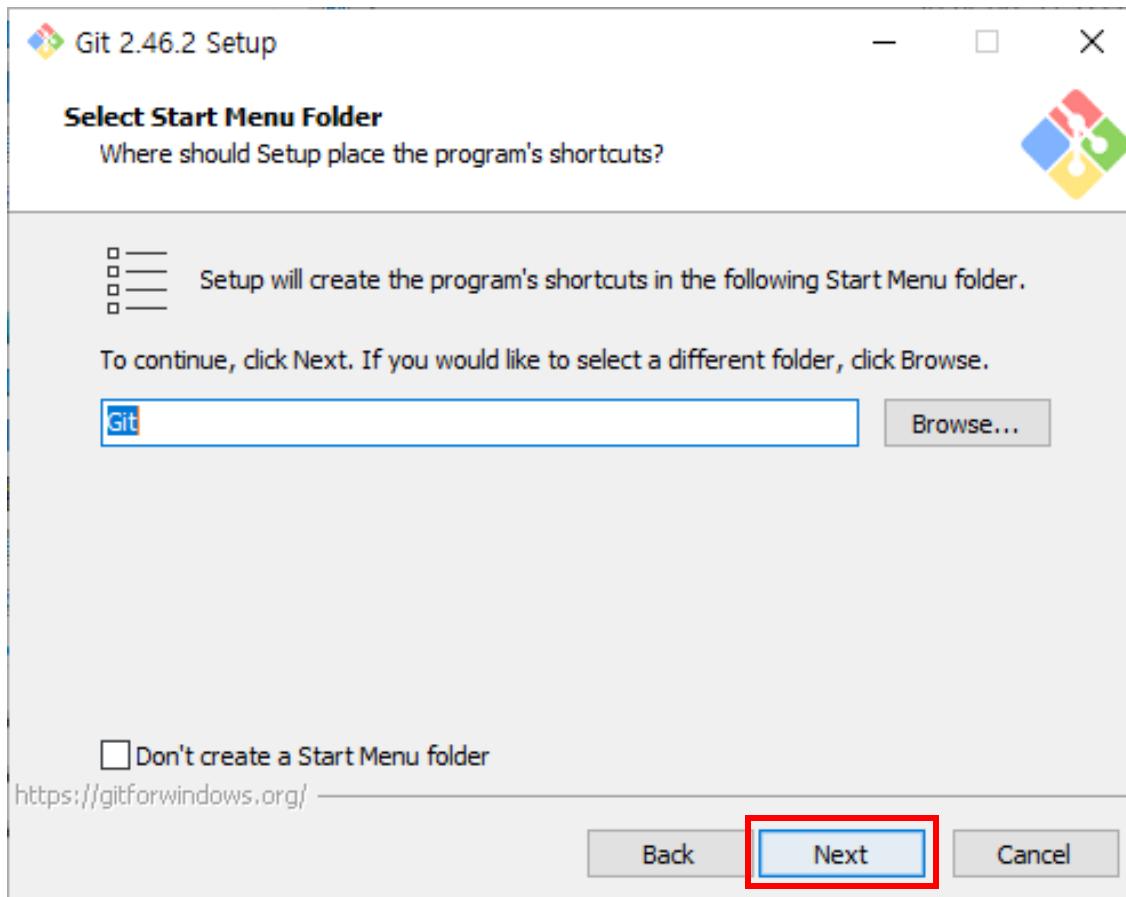
# Installing CKAN

## Installing Git



### Git for Windows

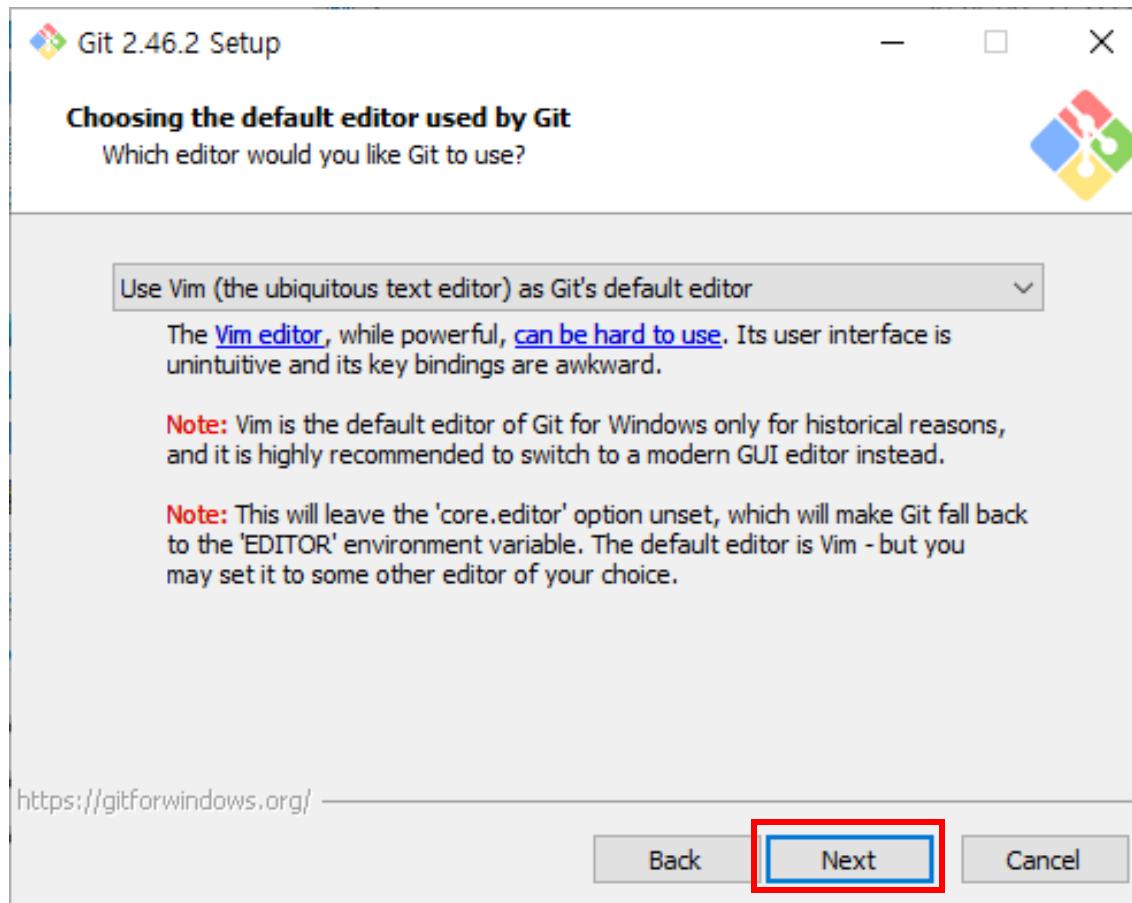
Click Next button





### Git for Windows

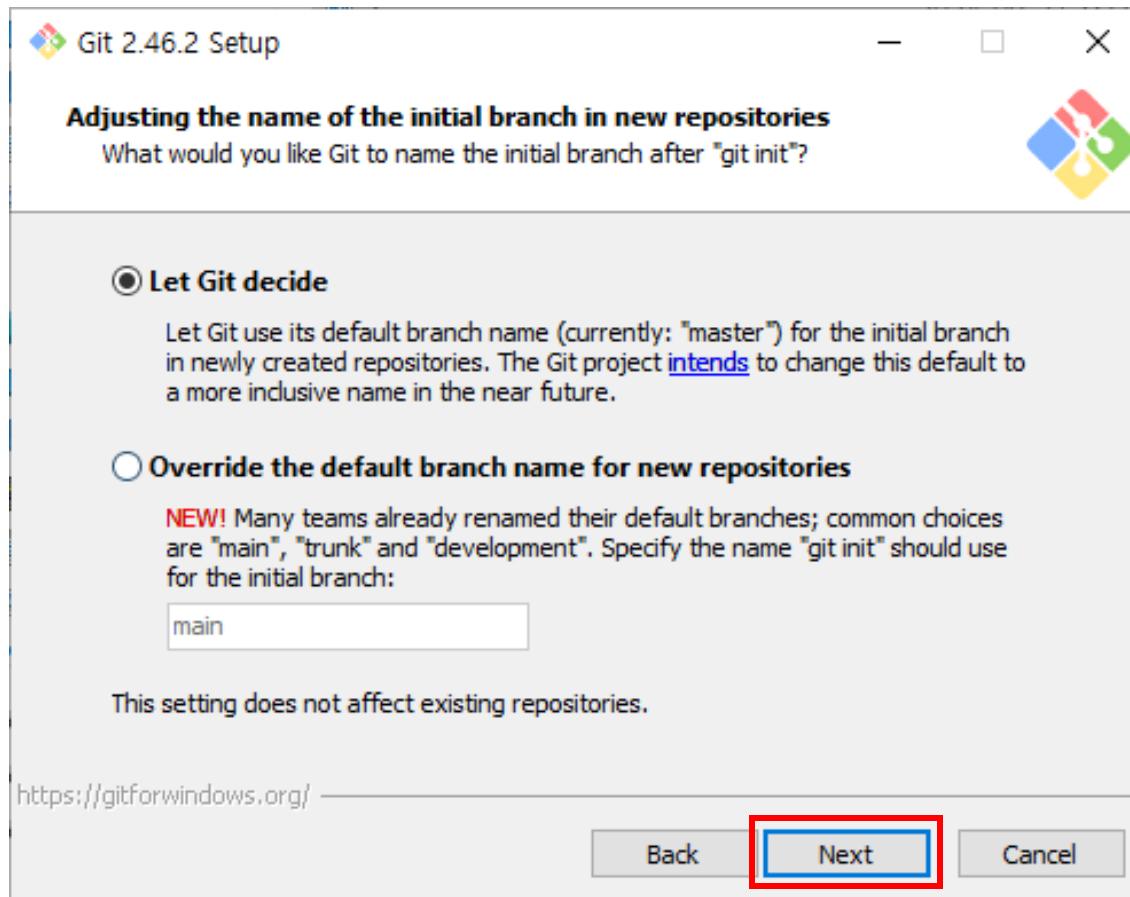
Click Next button





### Git for Windows

Click Next button



## I

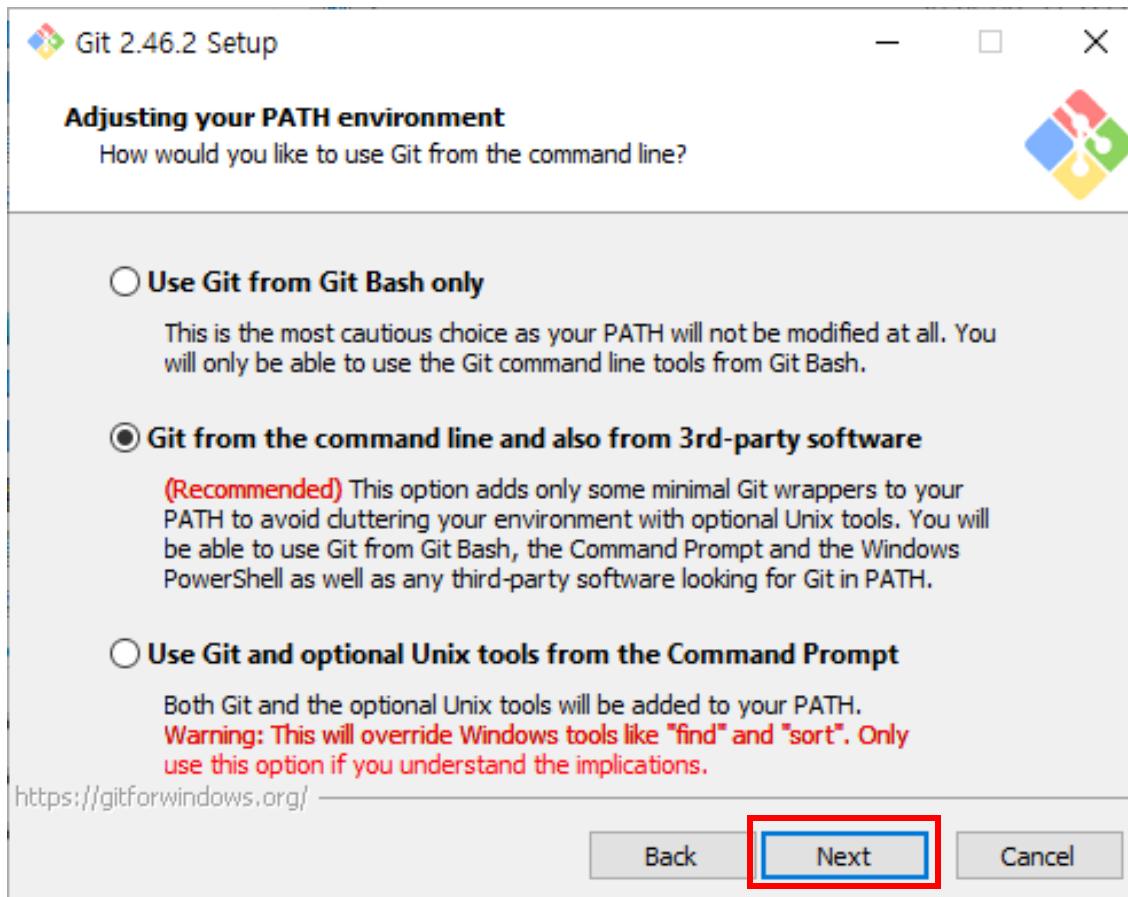
# Installing CKAN

## Installing Git



### Git for Windows

Click Next button



## I

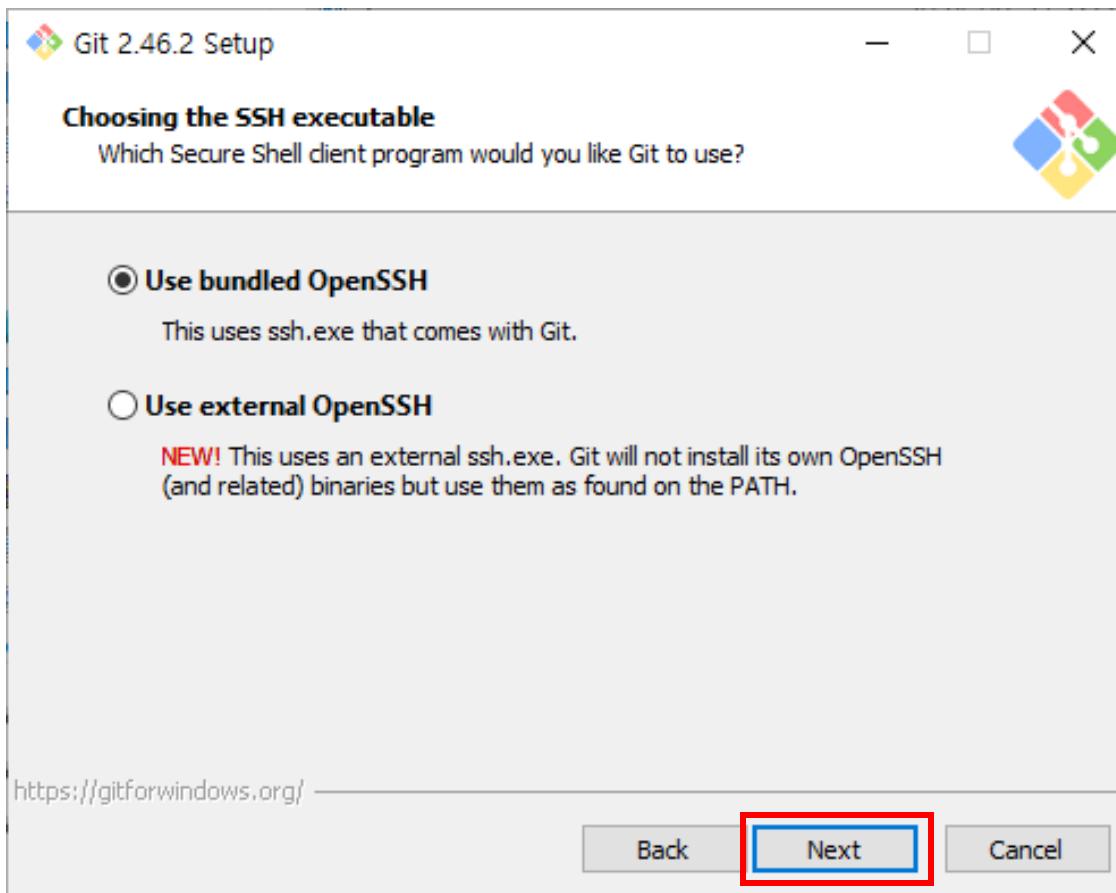
# Installing CKAN

## Installing Git



### Git for Windows

Click Next button



## I

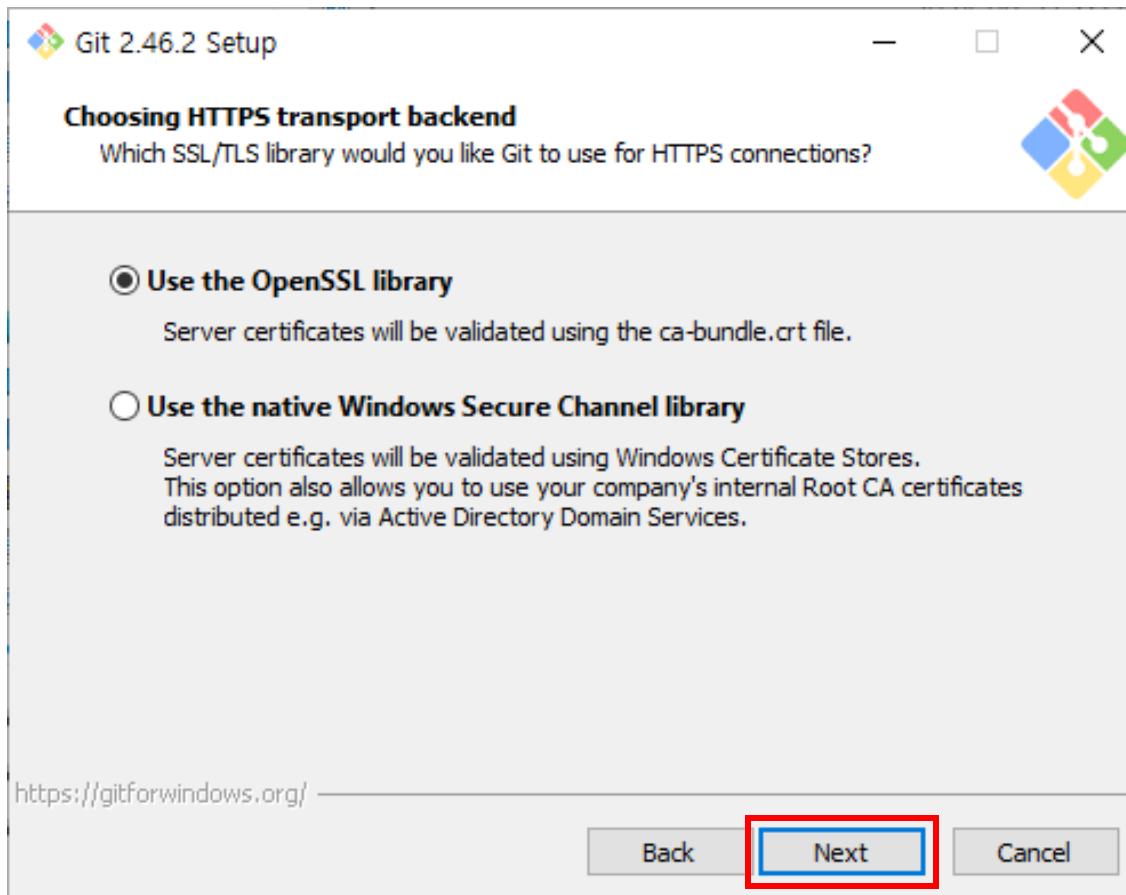
# Installing CKAN

## Installing Git



### Git for Windows

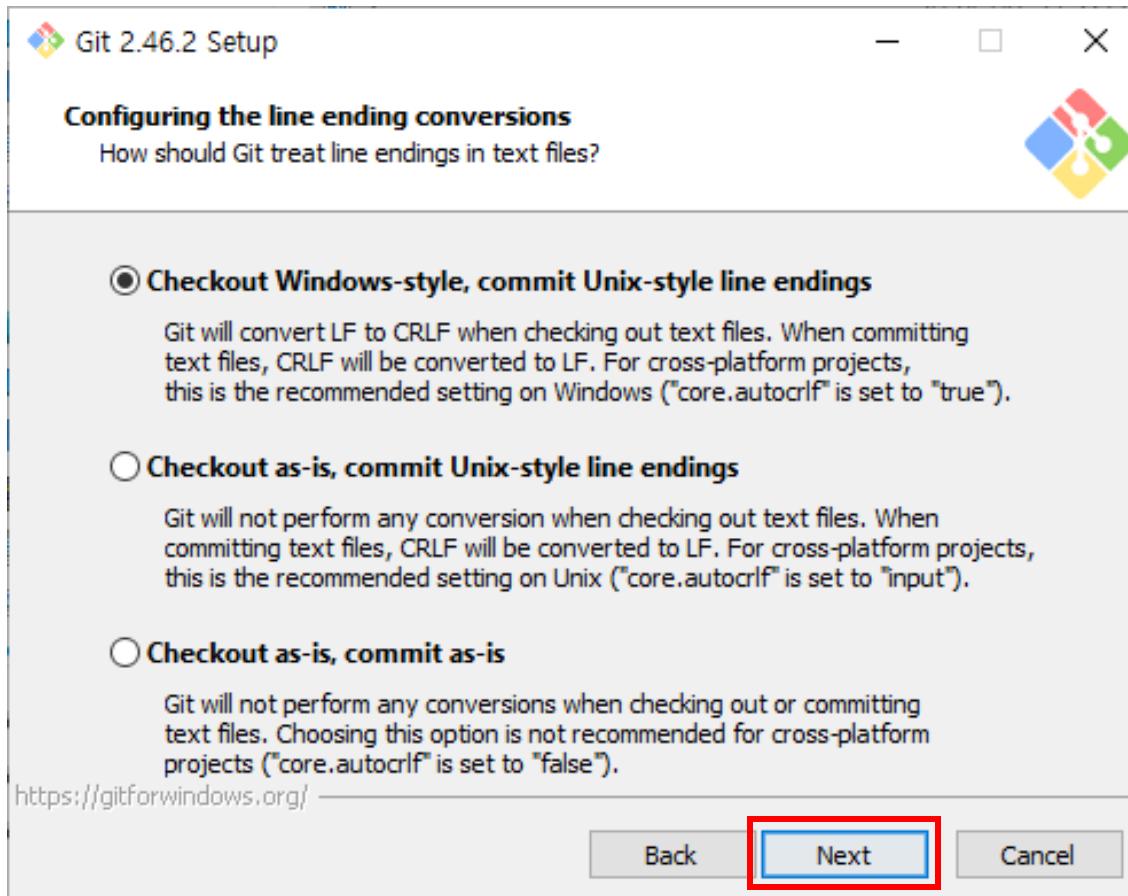
Click Next button





### Git for Windows

Click Next button



## I

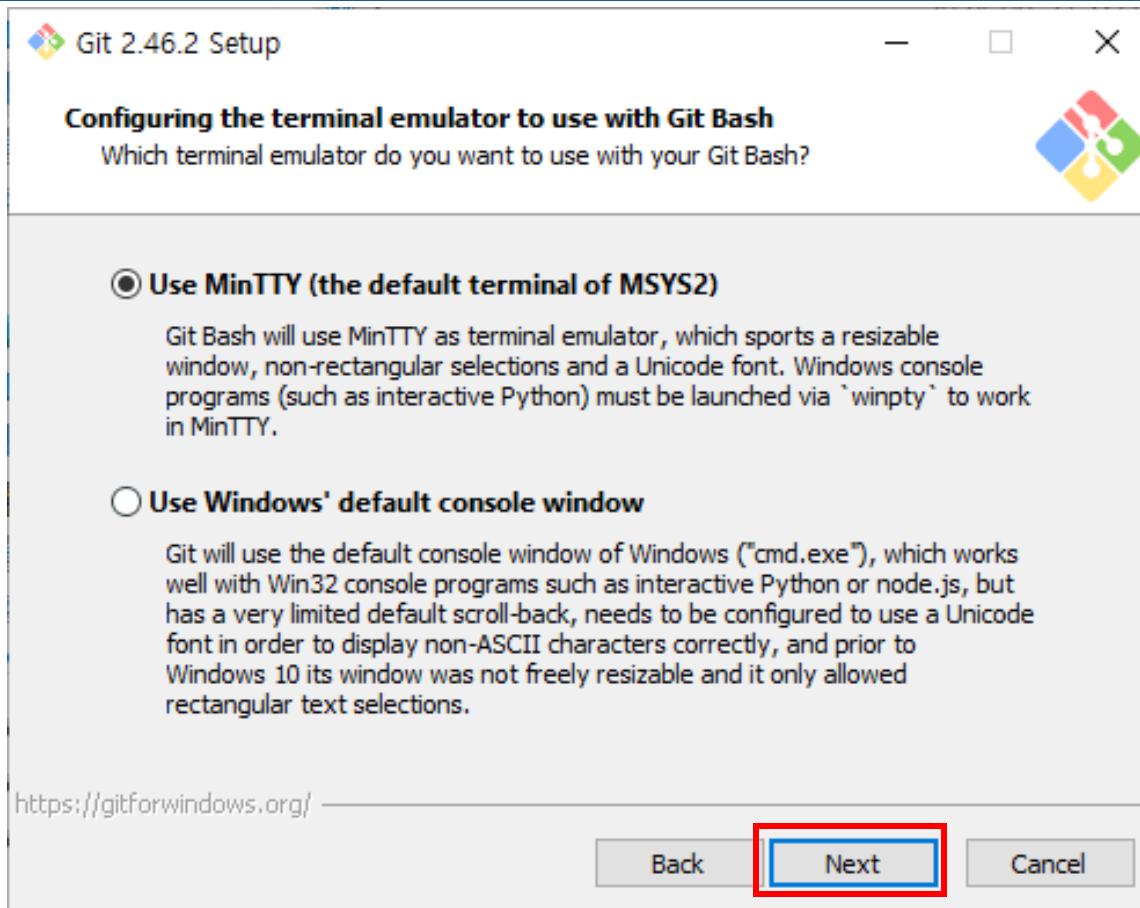
# Installing CKAN

## Installing Git



### Git for Windows

Click Next button



## I

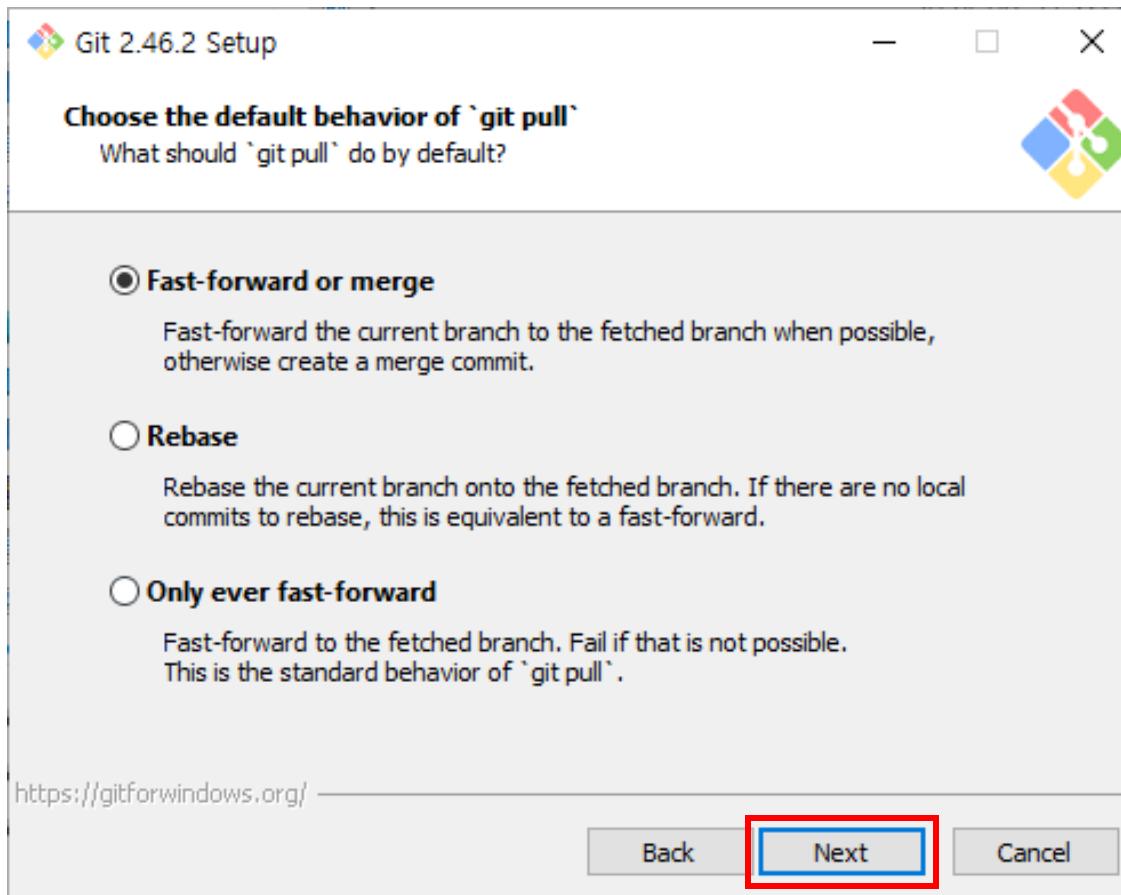
# Installing CKAN

## Installing Git



### Git for Windows

Click Next button



## I

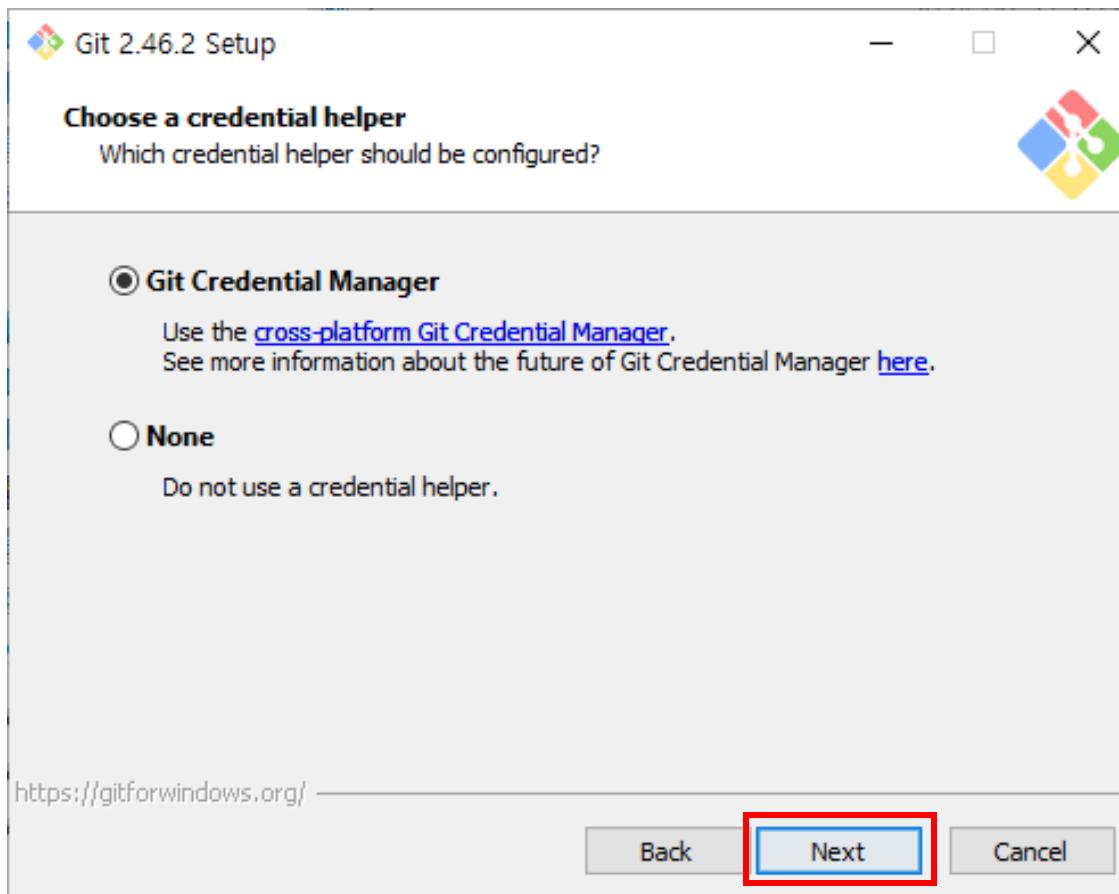
# Installing CKAN

## Installing Git



### Git for Windows

Click Next button

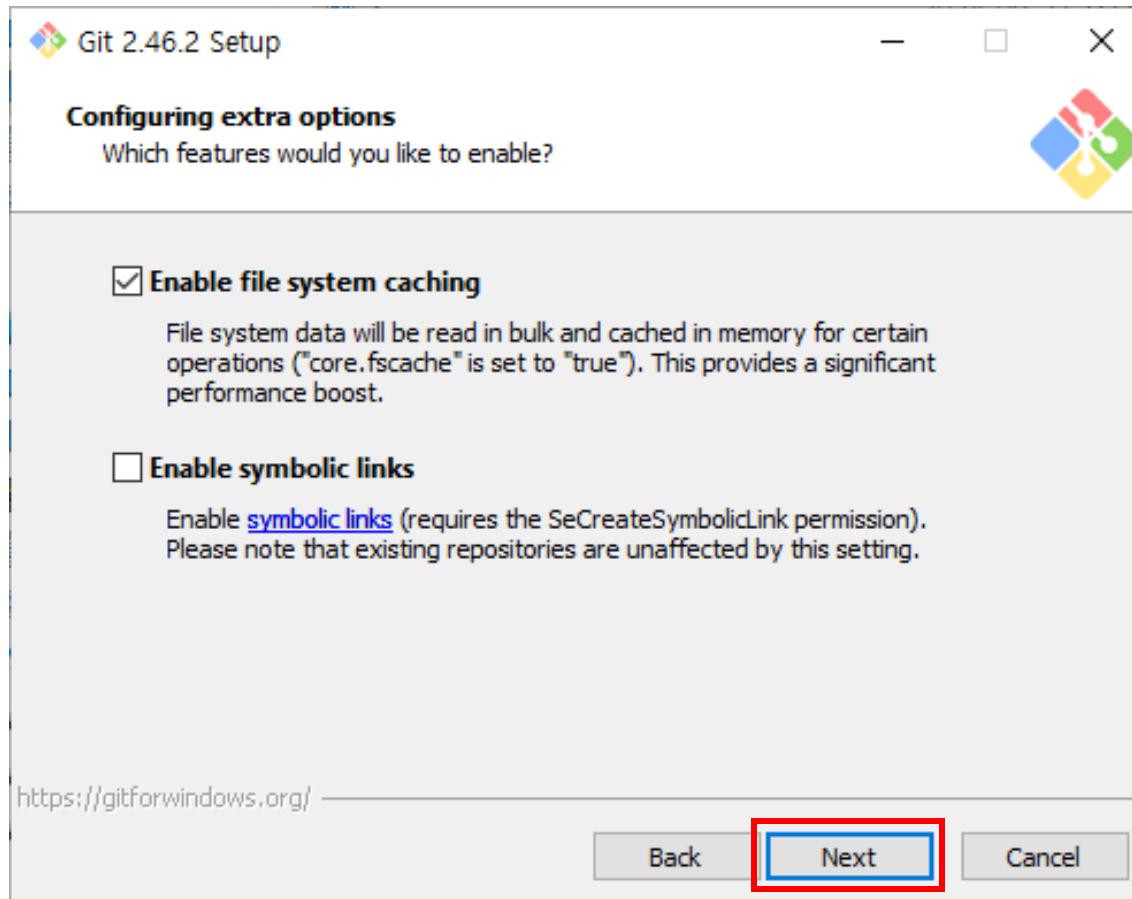


## Installing Git



### Git for Windows

Click Next button



## I

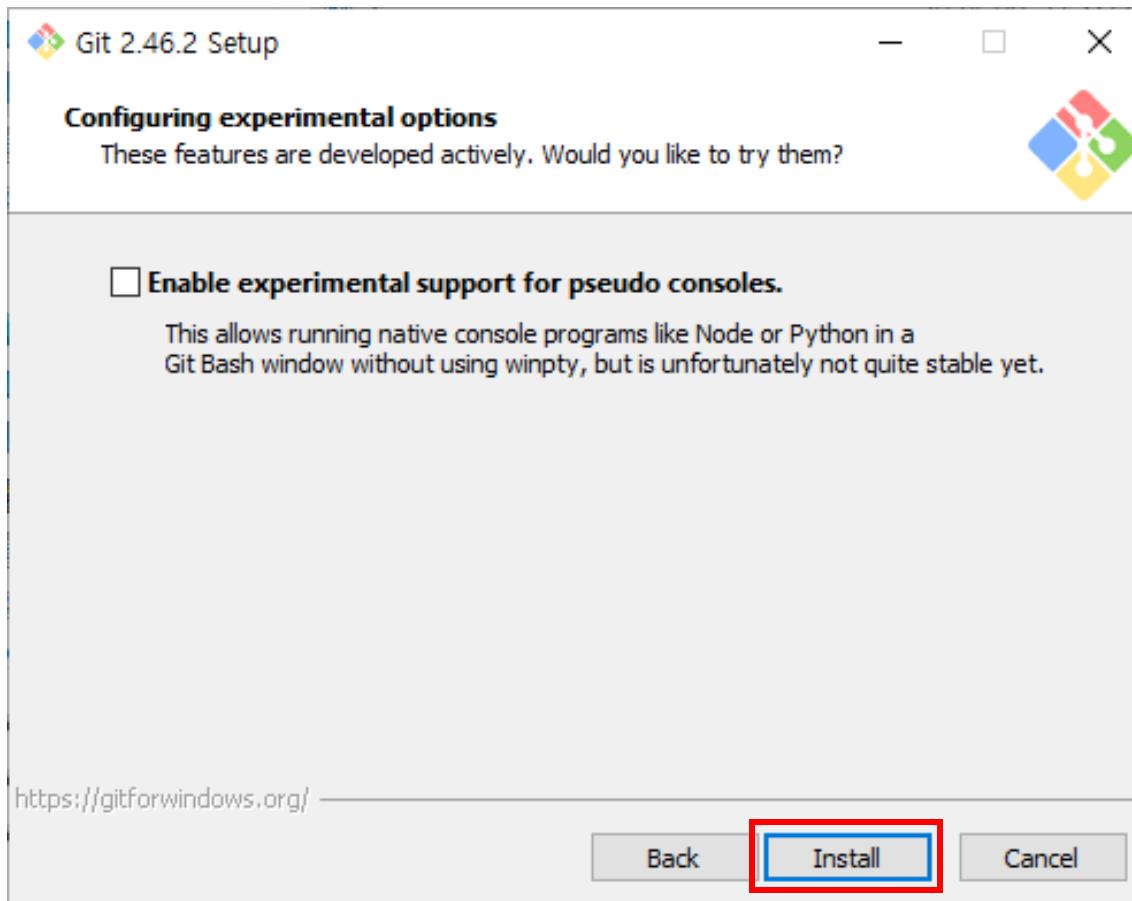
# Installing CKAN

## Installing Git



### Git for Windows

Click Install button



## I

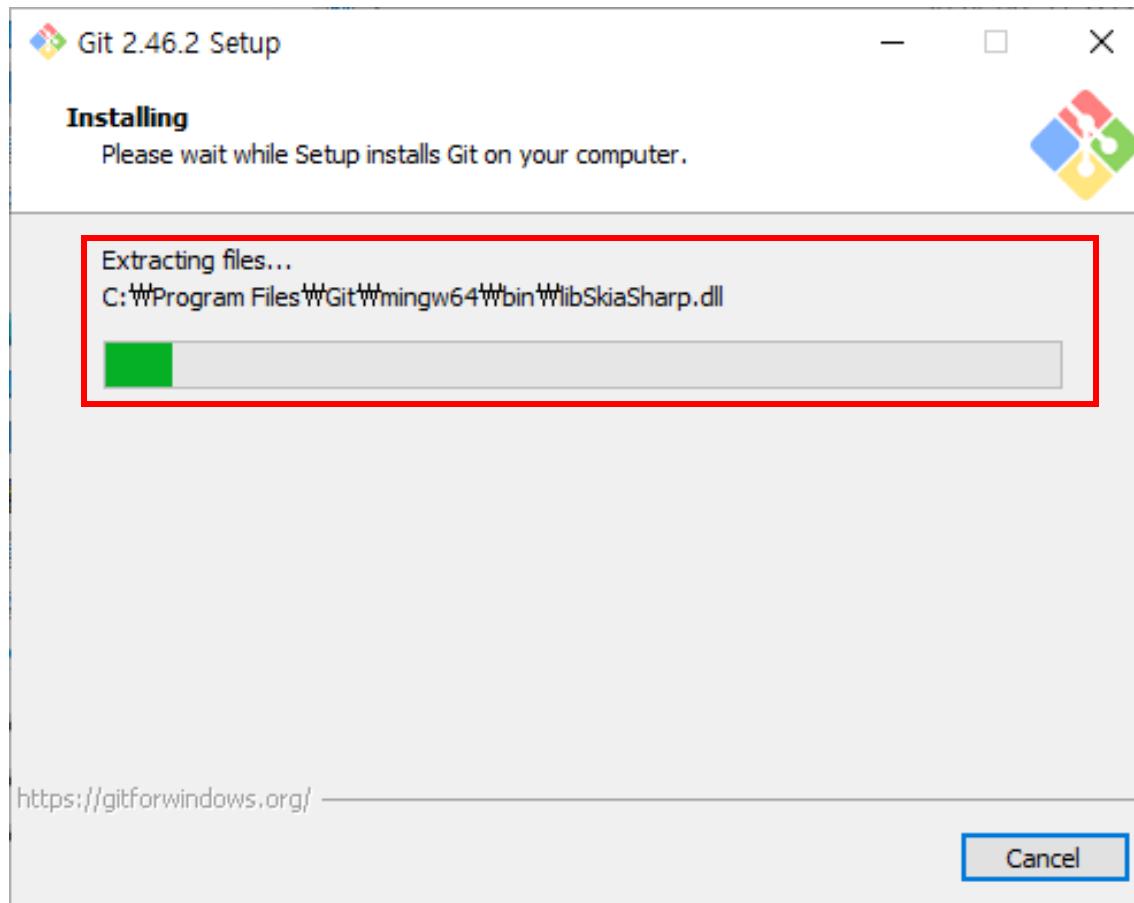
# Installing CKAN

## Installing Git



### Git for Windows

#### Wait while Git is installed



## I

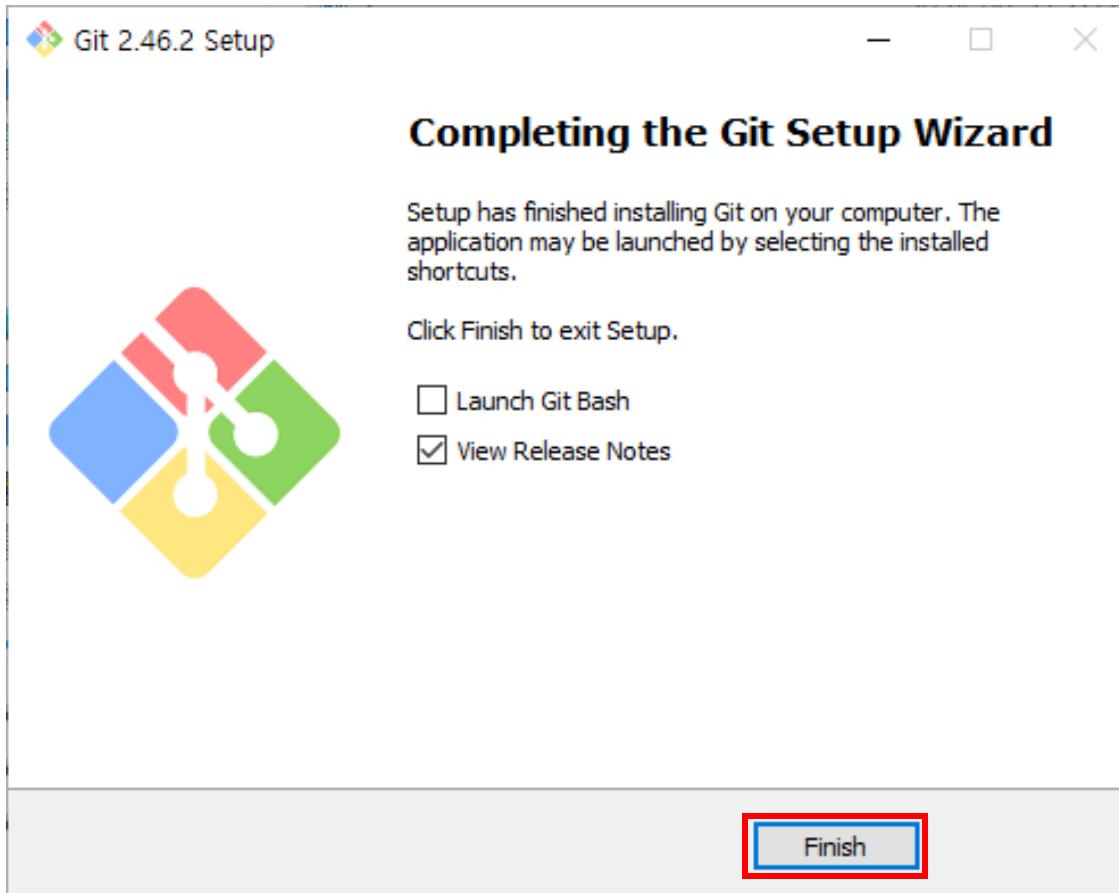
# Installing CKAN

## Installing Git



### Git for Windows

Click Finish button to exit setup



# I

# Installing CKAN

Installing Git



Git for Mac OS

Install Xcode Command Line Tools



```
xcode-select --install
```

# I

# Installing CKAN

Installing Git



Git for Mac OS

## Install Xcode Command Line Tools

```
xcode-select --install
```

## I

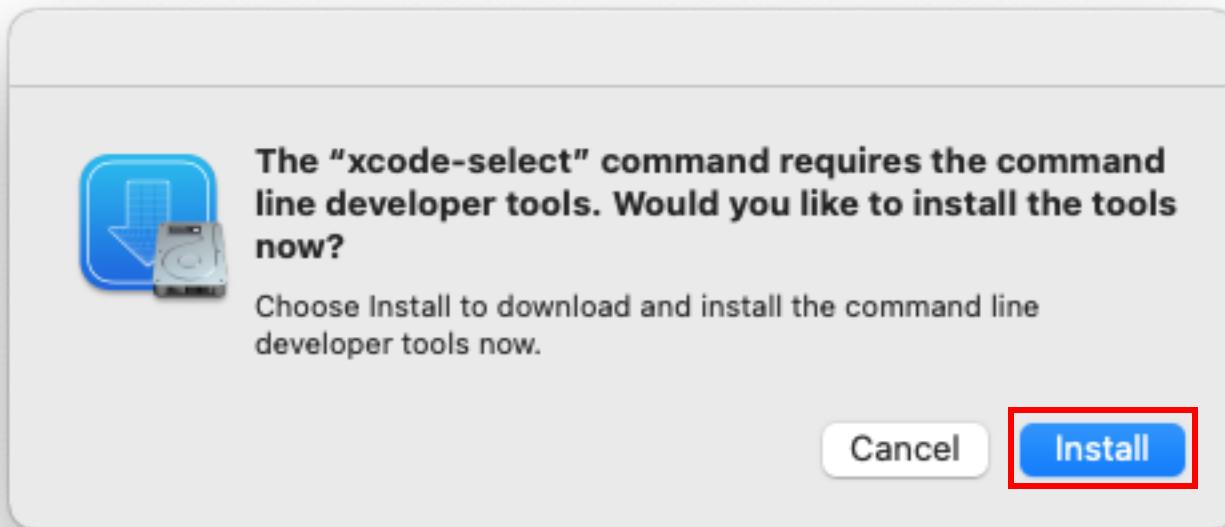
# Installing CKAN

## Installing Git



### Git for Mac OS

Click Install button



## I

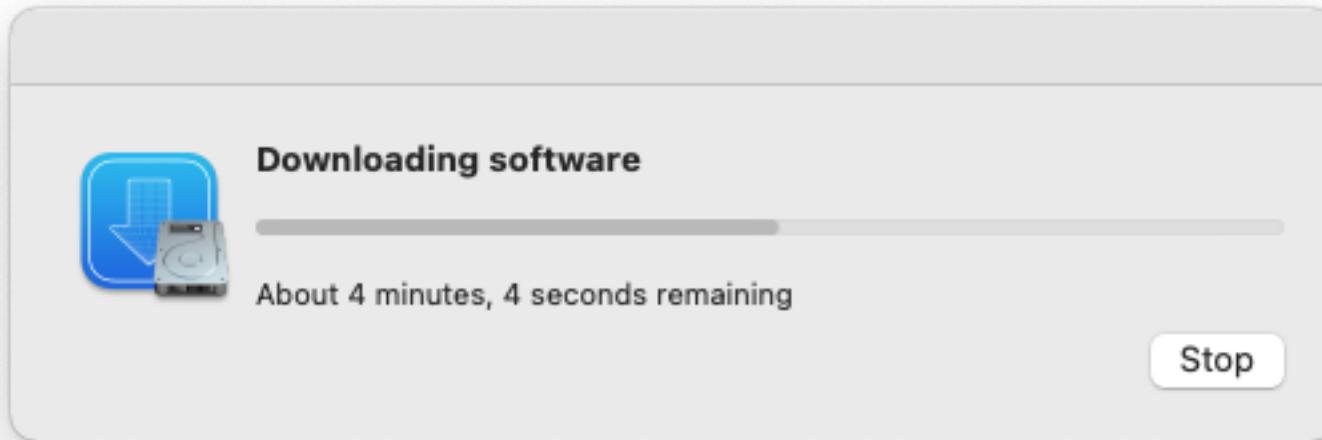
# Installing CKAN

Installing Git



Git for Mac OS

Wait while Xcode Command Line Tools is installed



# I

## Installing CKAN

Installing Git



Git for Mac OS

Click Done button to exit setup



The software was installed.

Done

## I

# Installing CKAN

## Installing Git



### Git for Linux (Debian/Ubuntu)

#### Install git using the apt



```
sudo apt update  
sudo apt install git -y
```



### Git for Linux (Debian/Ubuntu)

#### Install git using the apt

```
sudo apt update  
sudo apt install git -y
```

# Installing CKAN

## Installing Git



### Git for Linux (Debian/Ubuntu)

#### Install git using the apt

```
ckan@ckan-ubuntu:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
ckan@ckan-ubuntu:~$ sudo apt install git -y
Reading package lists... done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man libcurl3-gnutls liberror-perl libgdbm-compat4 libgdbm6 libperl5.34 patch perl perl-modules-5.34
Suggested packages:
  gettext-base git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
  | libterm-readline-perl-perl make libtap-harness-archive-perl
The following NEW packages will be installed:
  git git-man libcurl3-gnutls liberror-perl libgdbm-compat4 libgdbm6 libperl5.34 patch perl perl-modules-5.34
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 12.6 MB of archives.
After this operation, 70.1 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 perl-modules-5.34 all 5.34.0-3ubuntu1.3 [2976 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 libgdbm6 amd64 1.23-1 [33.9 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 libgdbm-compat4 amd64 1.23-1 [6606 B]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libperl5.34 amd64 5.34.0-3ubuntu1.3 [4820 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 perl amd64 5.34.0-3ubuntu1.3 [232 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl3-gnutls amd64 7.81.0-1ubuntu1.18 [284 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.11 [955 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.11 [3165 kB]
Get:10 http://archive.ubuntu.com/ubuntu jammy/main amd64 patch amd64 2.7.6-7build2 [109 kB]
Fetched 12.6 MB in 5s (2699 kB/s)
Selecting previously unselected package perl-modules-5.34.
```

## I

# Installing CKAN

## Installing Docker



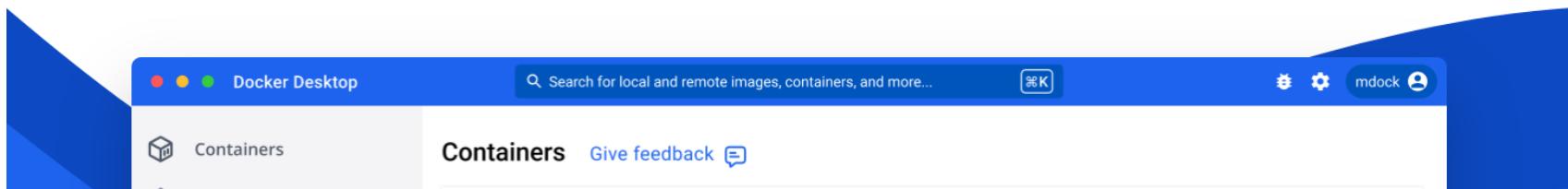
### Docker Desktop for Windows

Browse to Docker official website and download Docker Desktop installer for Windows

[Products](#)[Developers](#)[Pricing](#)[Support](#)[Blog](#)[Company](#)[Sign In](#)[Get started](#)

# Develop faster. Run anywhere.

Build with the #1 most-used developer tool

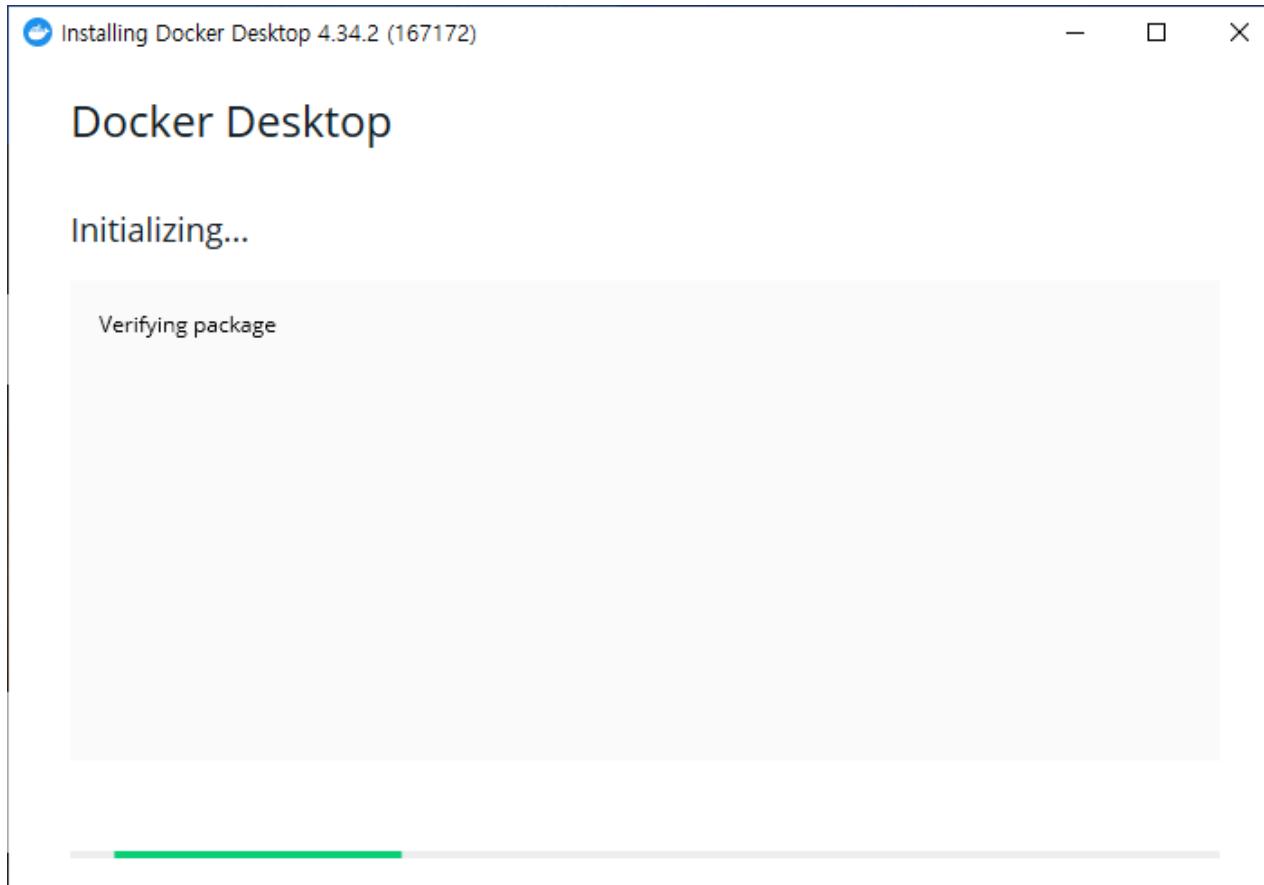
[Download for Windows - AMD64](#)[Learn more about Docker](#)

<https://www.docker.com/>



### Docker Desktop for Windows

Execute the installer and wait for initializing



## I

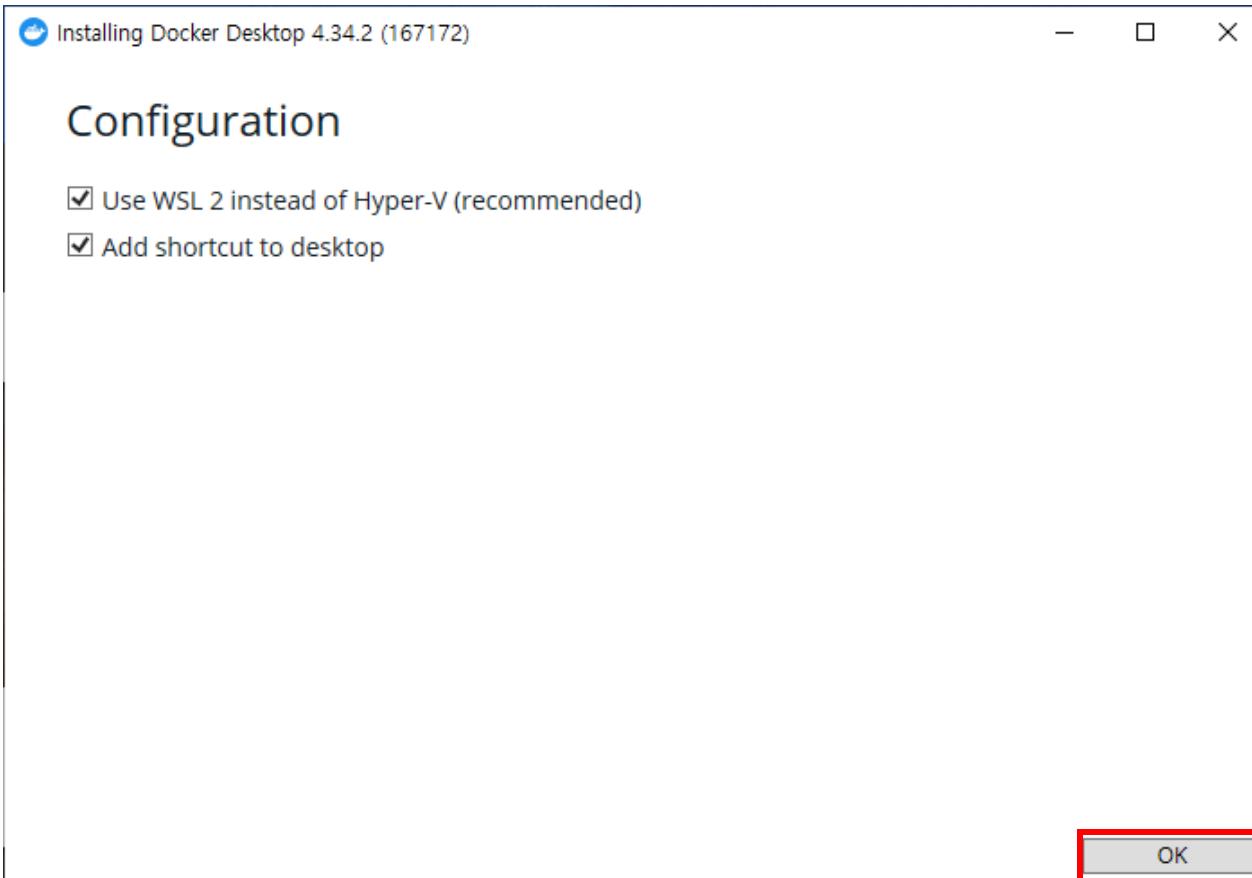
# Installing CKAN

## Installing Docker



### Docker Desktop for Windows

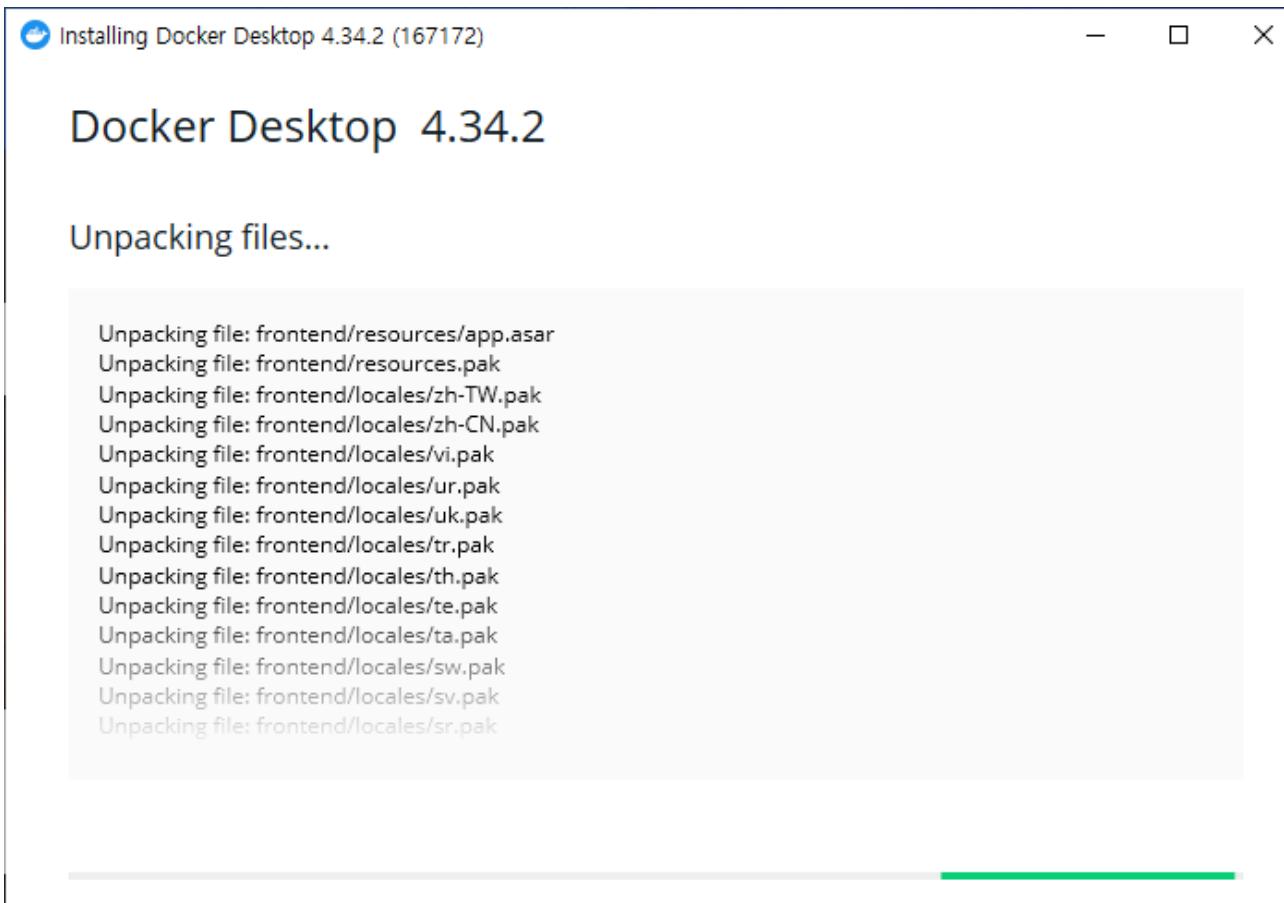
Click OK button





### Docker Desktop for Windows

#### Wait while Docker Desktop is installed



## I

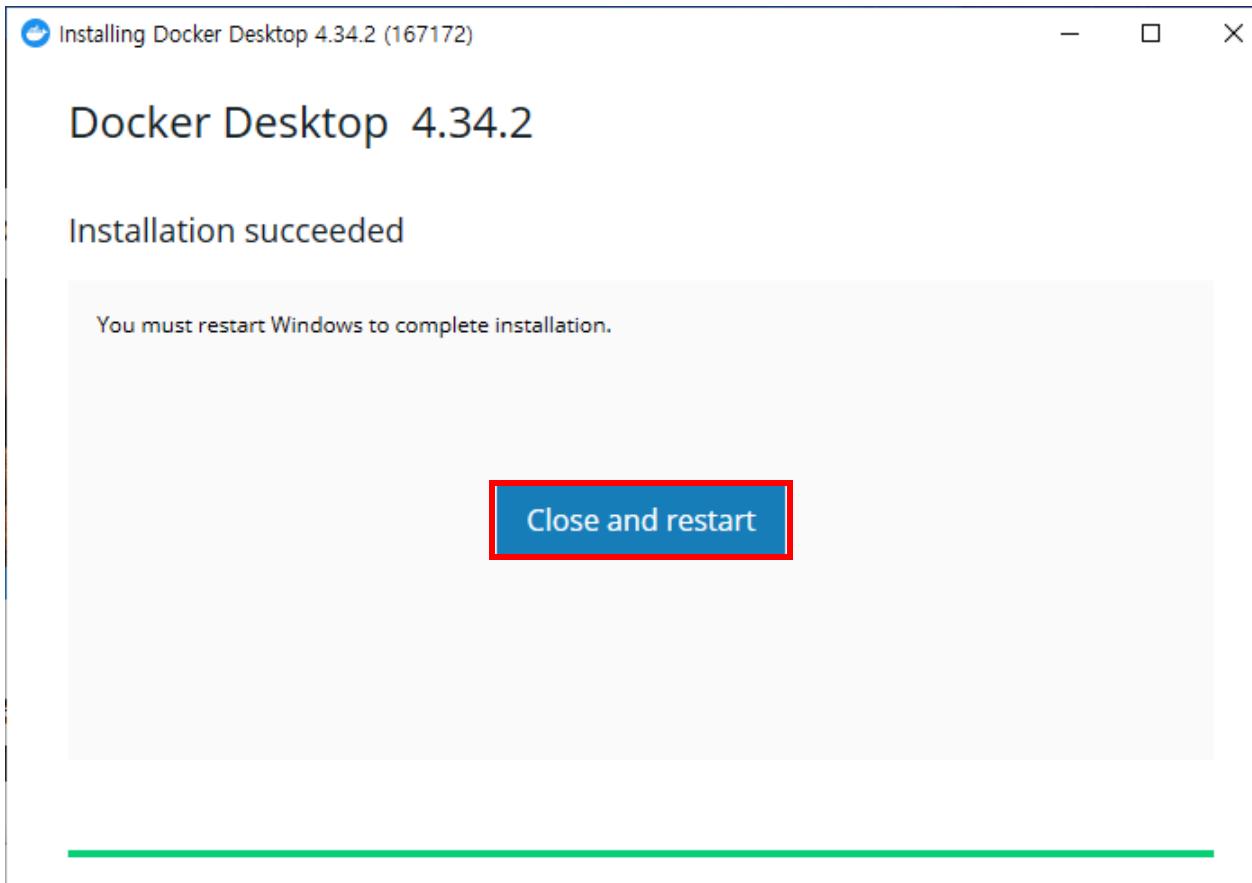
# Installing CKAN

## Installing Docker



### Docker Desktop for Windows

Click Close and restart button to restart the computer



## I

# Installing CKAN

## Installing Docker



### Docker Desktop for Windows

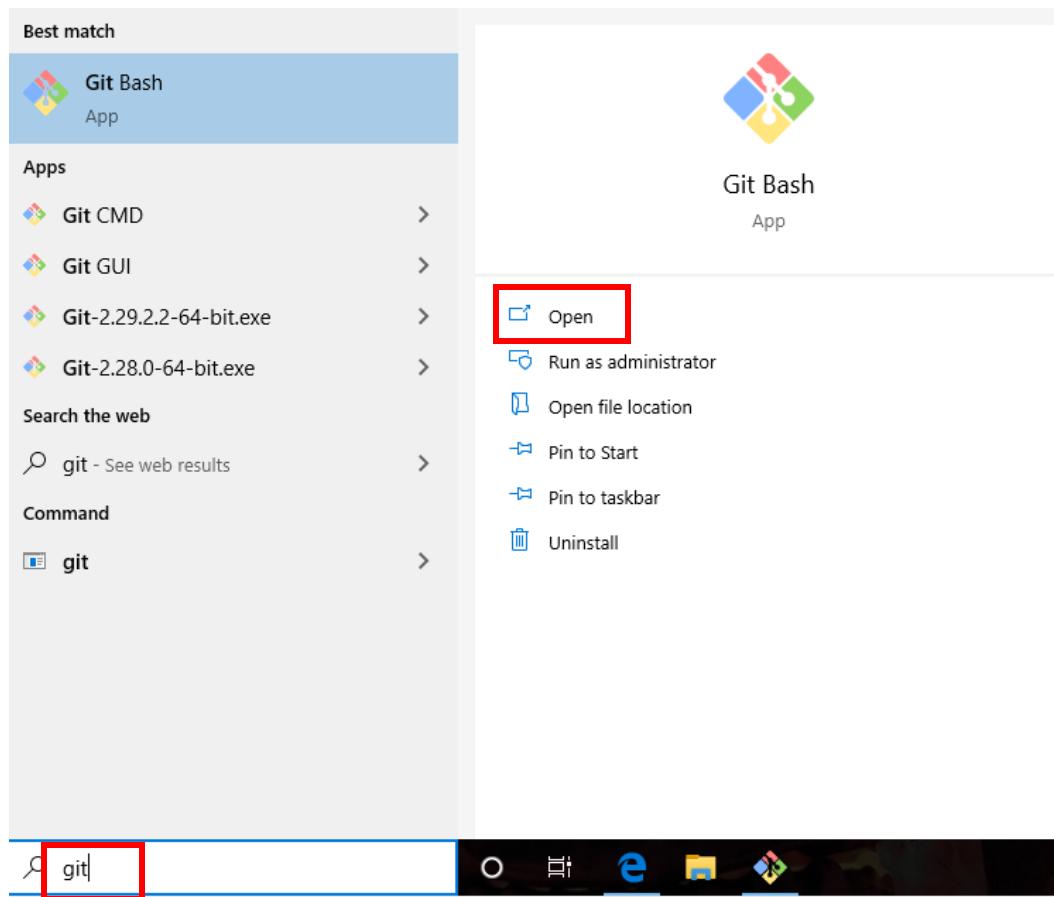
Docker Desktop will be started automatically after restarting

The screenshot shows the Docker Desktop application window. The title bar reads "docker desktop". The main interface has a sidebar on the left with icons for Containers, Images, Volumes, Builds (NEW), Dev Environments (BETA), and Docker Scout. Below the sidebar are sections for Extensions and Add Extensions. The main area is titled "Containers" with a "Give feedback" link. It features a central image of three containers and the text "Your running containers show up here". A subtitle explains that a container is an isolated environment for your code. At the bottom, there are two cards: "What is a container?" (5 mins) and "How do I run a container?" (6 mins), each with a snippet of Dockerfile code. A "View more in the Learning center" link is at the bottom of these cards. The status bar at the bottom shows "Engine running", system resources (RAM 3.70 GB, CPU 0.00%), and a note that the user is Not signed in. It also displays the version "v4.26.1" and a notification count of 1.



### Post-installation steps (Windows)

#### Open Git Bash





### Post-installation steps (Windows)

Verify that the Docker installation is successful by running hello-world image



```
docker run hello-world
```



### Post-installation steps (Windows)

Verify that the Docker installation is successful by running hello-world image

```
docker run hello-world
```



### Post-installation steps (Windows)

Verify that the Docker installation is successful by running hello-world image

```
MINGW64:/c/Users/Sean
Sean@DESKTOP-L3U210M MINGW64 ~
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pulling fs layer
c1ec31eb5944: Download complete
c1ec31eb5944: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

Sean@DESKTOP-L3U210M MINGW64 ~
$
```

# Installing CKAN

## Installing Docker



### Docker Desktop for Mac OS

Browse to Docker official website and download Docker Desktop installer for Mac OS

The screenshot shows the Docker website homepage. At the top, there's a navigation bar with links for Products, Developers, Pricing, Support, Blog, and Company. On the right side of the nav bar are a search icon, a "Sign In" button, and a "Get started" button. The main headline reads "Develop faster. Run anywhere." Below it, a sub-headline says "Build with the #1 most-used developer tool". There are two prominent buttons: a blue "Download Docker Desktop" button and a white "Learn more about Docker" button. A dropdown menu for Mac OS offers two options: "Download for Mac - Intel Chip" and "Download for Mac - Apple Silicon", with the latter being highlighted by a red box. To the left, a sidebar shows the Docker Desktop application interface with tabs for Containers, Images, and Volumes. The bottom of the page displays system resource usage: "5 MB / 15.1 GB (10 cores allocated)" and a "Show charts" button.

<https://www.docker.com/>

## I

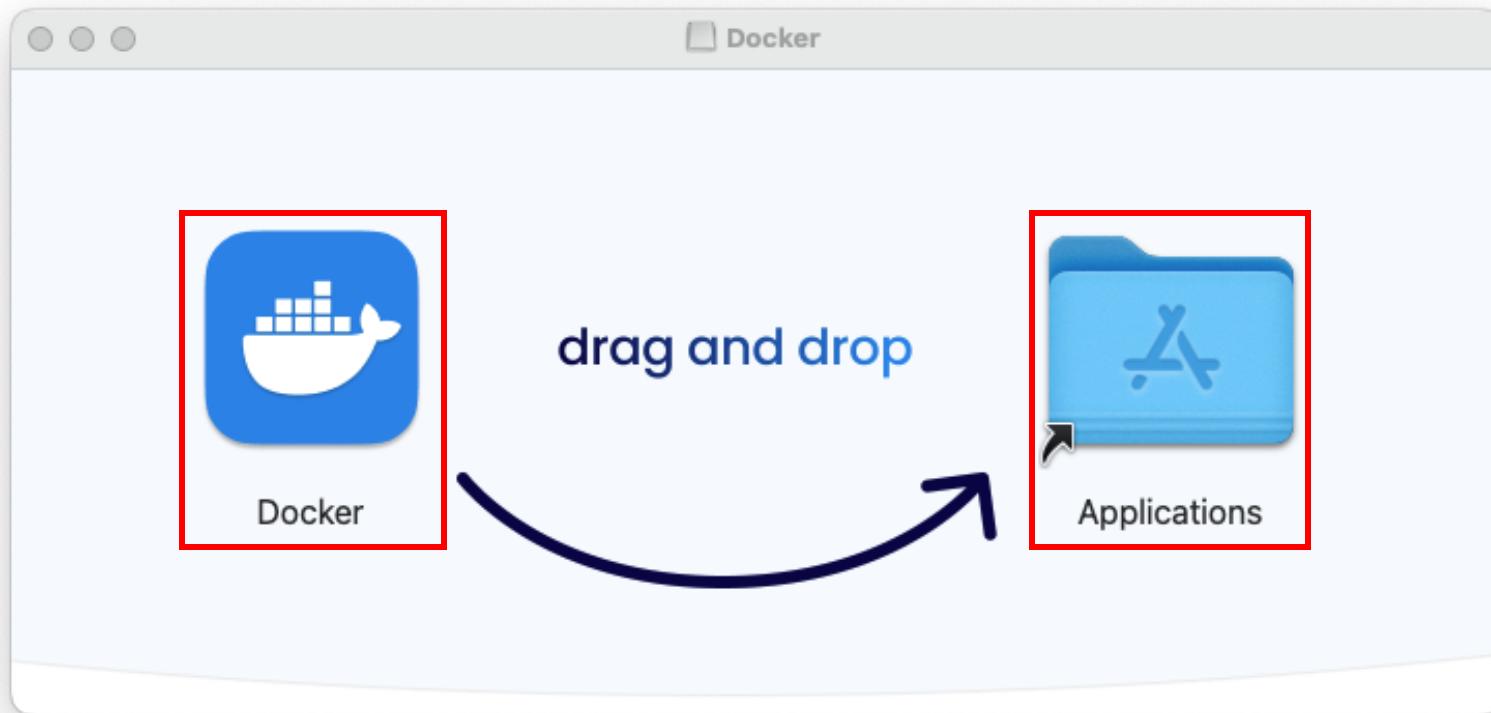
# Installing CKAN

Installing Docker



Docker Desktop for Mac OS

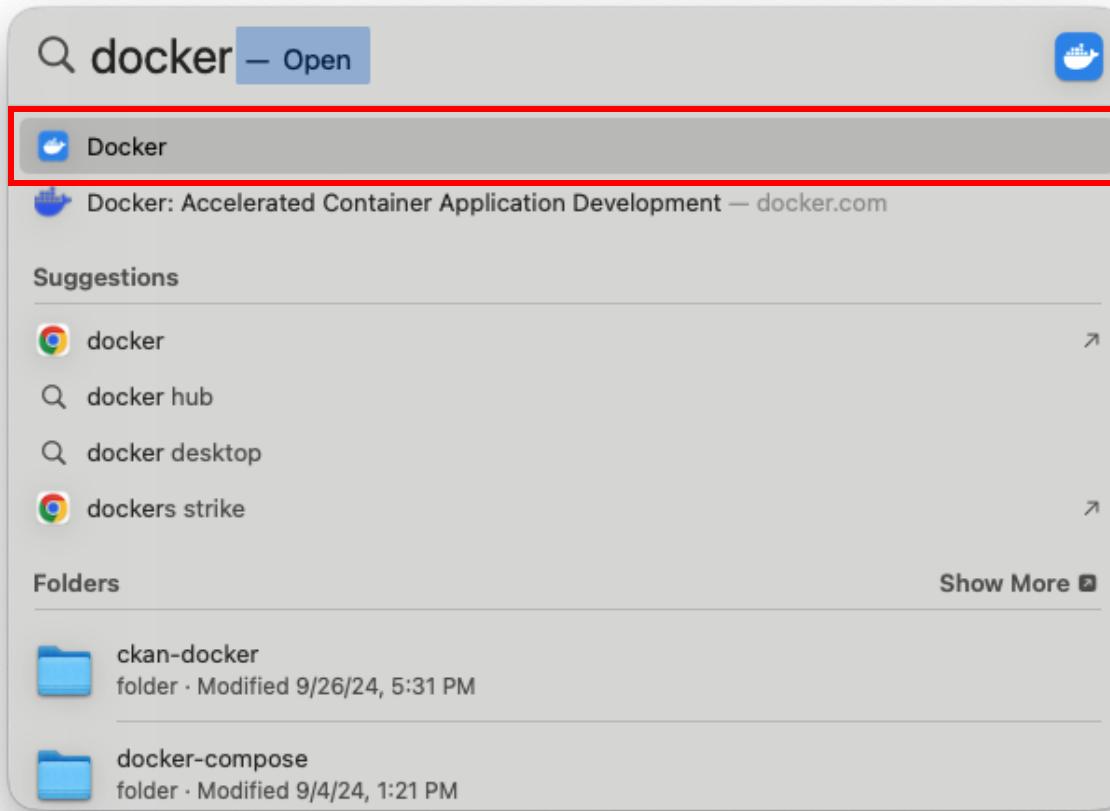
Drag and drop Docker icon to Applications folder





### Docker Desktop for Mac OS

Search for docker on Spotlight and open it





### Post-installation steps (Mac OS)

#### Manage Docker as a non-root user

```
# Create the docker group.  
sudo groupadd docker  
# Add your user to the docker group.  
sudo usermod -aG docker $USER  
# Activate the changes to groups.  
newgrp docker
```



### Post-installation steps (Mac OS)

#### Manage Docker as a non-root user

```
# Create the docker group.  
sudo groupadd docker  
# Add your user to the docker group.  
sudo usermod -aG docker $USER  
# Activate the changes to groups.  
newgrp docker
```



### Post-installation steps (Mac OS)

#### Manage Docker as a non-root user

```
ckan@ubuntu:~$ sudo groupadd docker
ckan@ubuntu:~$ sudo usermod -aG docker $USER
ckan@ubuntu:~$ newgrp docker
ckan@ubuntu:~$ █
```



### Post-installation steps (Mac OS)

Verify that the Docker installation is successful by running hello-world image



```
docker run hello-world
```



### Post-installation steps (Mac OS)

Verify that the Docker installation is successful by running hello-world image

```
docker run hello-world
```



### Post-installation steps (Mac OS)

Verify that the Docker installation is successful by running hello-world image

```
ckan@ckan-ubuntu:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>



### Docker Engine for Linux (Debian/Ubuntu)

#### Add Docker's official GPG key and add the repository to apt resources

```
# Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl  
sudo install -m 0755 -d /etc/apt/keyrings  
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc  
sudo chmod a+r /etc/apt/keyrings/docker.asc  
  
# Add the repository to Apt sources:  
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu \  
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

<https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>



### Docker Engine for Linux (Debian/Ubuntu)

#### Add Docker's official GPG key and add the repository to apt resources

```
# Add Docker's official GPG key:
```

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
# Add the repository to Apt sources:
```

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
$(./etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

## I

# Installing CKAN

## Installing Docker



### Docker Engine for Linux (Debian/Ubuntu)

#### Add Docker's official GPG key and add the repository to apt resources

```
ckan@ckan-was:~$ sudo apt-get update
[sudo] password for ckan:
Hit:1 http://la.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://la.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Hit:3 http://la.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://la.archive.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:5 http://la.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,396 kB]
Fetched 3,651 kB in 10s (376 kB/s)
Reading package lists... Done
ckan@ckan-was:~$ sudo apt-get install ca-certificates curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.20.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.68.0-1ubuntu2.22).
curl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
ckan@ckan-was:~$ sudo install -m 0755 -d /etc/apt/keyrings
ckan@ckan-was:~$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
ckan@ckan-was:~$ sudo chmod a+r /etc/apt/keyrings/docker.asc
ckan@ckan-was:~$ echo \
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \
> $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ckan@ckan-was:~$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:2 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [46.4 kB]
Hit:3 http://la.archive.ubuntu.com/ubuntu focal InRelease
Hit:4 http://la.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:5 http://la.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:6 http://la.archive.ubuntu.com/ubuntu focal-security InRelease
Fetched 104 kB in 1s (73.7 kB/s)
Reading package lists... Done
ckan@ckan-was:~$
```



### Docker Engine for Linux (Debian/Ubuntu)

Install the Docker packages using apt



```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```



### Docker Engine for Linux (Debian/Ubuntu)

Install the Docker packages using apt

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```



### Docker Engine for Linux (Debian/Ubuntu)

#### Install the Docker packages using apt

```
ckan@ckan-was:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin pigz slirp4netns
0 upgraded, 8 newly installed, 0 to remove and 4 not upgraded.
Need to get 122 MB of archives.
After this operation, 437 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://download.docker.com/linux/ubuntu focal/stable amd64 containerd.io amd64 1.7.18-1 [30.5 MB]
Get:2 http://la.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:3 http://la.archive.ubuntu.com/ubuntu focal/universe amd64 slirp4netns amd64 0.4.3-1 [74.3 kB]
Get:4 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-buildx-plugin amd64 0.15.1-1~ubuntu.20.04~focal [29.8 MB]
Get:5 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-cli amd64 5:27.0.3-1~ubuntu.20.04~focal [14.6 MB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce amd64 5:27.0.3-1~ubuntu.20.04~focal [25.3 MB]
Get:7 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-ce-rootless-extras amd64 5:27.0.3-1~ubuntu.20.04~focal [9,327 kB]
Get:8 https://download.docker.com/linux/ubuntu focal/stable amd64 docker-compose-plugin amd64 2.28.1-1~ubuntu.20.04~focal [12.5 MB]
Fetched 122 MB in 3min 37s (562 kB/s)
Selecting previously unselected package pigz.
(Reading database ... 72425 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.7.18-1_amd64.deb ...
Unpacking containerd.io (1.7.18-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.15.1-1~ubuntu.20.04~focal_amd64.deb ...
```



### Post-installation steps (Debian/Ubuntu)

#### Manage Docker as a non-root user

```
# Create the docker group.  
sudo groupadd docker  
# Add your user to the docker group.  
sudo usermod -aG docker $USER  
# Activate the changes to groups.  
newgrp docker
```



### Post-installation steps (Debian/Ubuntu)

#### Manage Docker as a non-root user

```
# Create the docker group.  
sudo groupadd docker  
# Add your user to the docker group.  
sudo usermod -aG docker $USER  
# Activate the changes to groups.  
newgrp docker
```



### Post-installation steps (Debian/Ubuntu)

#### Manage Docker as a non-root user

```
ckan@ubuntu:~$ sudo groupadd docker
ckan@ubuntu:~$ sudo usermod -aG docker $USER
ckan@ubuntu:~$ newgrp docker
ckan@ubuntu:~$ █
```



### Post-installation steps (Debian/Ubuntu)

Verify that the Docker installation is successful by running hello-world image



```
docker run hello-world
```



### Post-installation steps (Debian/Ubuntu)

Verify that the Docker installation is successful by running hello-world image

```
docker run hello-world
```



### Post-installation steps (Debian/Ubuntu)

Verify that the Docker installation is successful by running hello-world image

```
ckan@ckan-ubuntu:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:d211f485f2dd1dee407a80973c8f129f00d54604d2c90732e8e320e5038a0348
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID:

<https://hub.docker.com/>

For more examples and ideas, visit:

<https://docs.docker.com/get-started/>



## Prerequisites

**Clone ckan-docker git repository and open it with code editor (e.g Visual Studio Code)**

```
● ● ●  
git clone https://github.com/ckan/ckan-docker.git  
cd ckan-docker  
# Use ckan 2.10.4  
git reset --hard a1ae332cb91f83b209f1979d2b642746c16ee164  
# Command for Visual Studio Code  
code .
```



### Prerequisites

#### Clone ckan-docker git repository and open it with code editor (e.g Visual Studio Code)

```
git clone https://github.com/ckan/ckan-docker.git
cd ckan-docker
# Use ckan 2.10.4
git reset --hard a1ae332cb91f83b209f1979d2b642746c16ee164
# Command for Visual Studio Code
code .
```

## I

# Installing CKAN

Setting up with Docker Compose



## Prerequisites

Clone ckan-docker git repository and open it with code editor (e.g Visual Studio Code)

The terminal session shows the following commands:

```
x sean > ~/temp > git clone https://github.com/ckan/ckan-docker.git
Cloning into 'ckan-docker'...
remote: Enumerating objects: 1615, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (35/35), done.
remote: Total 1615 (delta 18), reused 17 (delta 8), pack-reused 1572 (from 1)
Receiving objects: 100% (1615/1615), 746.58 KiB | 6.49 MiB/s, done.
Resolving deltas: 100% (751/751), done.

sean > ~/temp > cd ckan-docker
sean > ~/temp/ckan-docker > master > git reset --hard a1ae332cb91f83b209f1979d2b642746c16ee164
HEAD is now at a1ae332 Merge pull request #160 from ckan/no-ckan-port
sean > ~/temp/ckan-docker > master > code .
```

The Visual Studio Code interface shows the project structure and the contents of the `docker-compose.yml` file.

`docker-compose.yml` content:

```
volumes:
  ckan_storage:
    pg_data:
    solr_data:
    pip_cache:
    site_packages:

services:
  nginx:
    build:
      context: nginx/
      dockerfile: Dockerfile
    networks:
      - webnet
      - ckannet
```

# I Installing CKAN

# Setting up with Docker Compose



## Prerequisites

## Make a copy of .env.example file and rename it to .env

The screenshot shows the CKAN-DOCKER project structure in the Explorer sidebar. The .env file is selected and highlighted with a red box. The .env.example file is also visible in the editor. The code editor displays environment variables for CKAN, including host ports, database credentials, and datastore settings.

```
1 # Host Ports
2 CKAN_PORT_HOST=5000
3 NGINX_PORT_HOST=80
4
5 # CKAN databases
6 POSTGRES_USER=postgres
7 POSTGRES_PASSWORD=postgres
8 POSTGRES_DB=postgres
9 POSTGRES_HOST=db
10 CKAN_DB_USER=ckandbuser
11 CKAN_DB_PASSWORD=ckandbpassword
12 CKAN_DB=ckandb
13 DATASTORE_READONLY_USER=datastore_ro
14 DATASTORE_READONLY_PASSWORD=datastore
```



### Disabling HTTPS configurations

#### [project directory]/docker-compose.yml

```
version: '3.8'
services:
  nginx:
    build:
      context: nginx/
      dockerfile: Dockerfile
    networks:
      - webnet
      - ckannet
    depends_on:
      - ckan
    condition: service healthy
    ports:
      - "0.0.0.0:${NGINX_PORT_HOST}:${NGINX_PORT}"
  ckan:
    build:
      context: ckan/
      dockerfile: Dockerfile
      args:
        - TZ=${TZ}
    networks:
      - ckannet
      - dbnet
      - solrnet
      - redisnet
    ports:
      - "0.0.0.0:${NGINX_PORT_HOST}:${NGINX_PORT}"
```

In this tutorial, we will not run CKAN instance under HTTPS.

So, we use `NGINX_PORT_HOST` instead of `NGINX_SSLPORT_HOST`, and use `NGINX_PORT` instead of `NGINX_SSLPORT`.



### Disabling HTTPS configurations

[project directory]/docker-compose.yml

ports:

- "0.0.0.0:\${NGINX\_PORT\_HOST}:\${NGINX\_PORT}"



### Disabling HTTPS configurations

#### [project directory]/nginx/setup/default.conf

```
default.conf M ×
nginx > setup > default.conf
1 server {
2     listen 80;
3     listen [::]:80;
4     # listen 443 ssl;
5     # listen [::]:443 ssl;
6     server_name localhost;
7     ssl_certificate /etc/nginx/certs/ckan-local.crt;
8     ssl_certificate_key /etc/nginx/certs/ckan-local.key;
9
10    # # TLS 1.2 & 1.3 only
11    # ssl_protocols TLSv1.2 TLSv1.3;
12
13    # # Disable weak ciphers
14    # ssl_prefer_server_ciphers on;
15    # ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305@TLSv1.3:!ECDSA-3DES-EDE-CBC-SHA:@STRENGTH';
16
17    # # SSL sessions
18    # ssl_session_timeout 1d;
19    # # ssl_session_cache d fine in stream and http
20    # ssl_session_tickets off;
21
22    # access_log /var/log/nginx/host.access.log main;
23
24    location / {
25        proxy_pass http://ckan:5000/;
26        proxy_set_header X-Forwarded-For $remote_addr;
27        proxy_set_header Host $host;
28        #proxy_cache cache;
29        proxy_cache_bypass $cookie_auth_tkt;
30        proxy_no_cache $cookie_auth_tkt;
31        proxy_cache_valid 30m;
32        proxy_cache_key $host$scheme$proxy_host$request_uri;
33    }
34}
```

01

listen 80;  
listen [::]:80;

02

# ssl\_certificate /etc/nginx/certs/ckan-local.crt;  
# ssl\_certificate\_key /etc/nginx/certs/ckan-local.key;

Default HTTPS port is 443, while default HTTP port is 80.

Therefore, we should replace nginx's listening ports from 443 to 80.

01

Comment out some directives under server\_name, starting from ssl\_certificate /etc/nginx/... to ssl\_session\_tickets off.

02

These directives configure the HTTPS server.



### Disabling HTTPS configurations

[project directory]/nginx/setup/default.conf

```
listen 80;  
listen [::]:80;
```



### Disabling HTTPS configurations

#### [project directory]/nginx/Dockerfile

```
Dockerfile M ×  
nginx > Dockerfile > ...  
You, 1 second ago | 2 authors (You and one other)  
1 FROM nginx:stable-alpine  
2  
3 ENV NGINX_DIR=/etc/nginx  
4  
5 RUN apk update --no-cache && \  
6     apk upgrade --no-cache  
7     # apk upgrade --no-cache && \  
8     # apk add --no-cache openssl  
9  
10 COPY setup/nginx.conf ${NGINX_DIR}/nginx.conf  
11 COPY setup/index.html /usr/share/nginx/html/index.html  
12 COPY setup/default.conf ${NGINX_DIR}/conf.d/  
13  
14 # RUN mkdir -p ${NGINX_DIR}/certs  
15  
16 ENTRYPOINT \  
17     # openssl req \  
18     #     -subj '/C=KR/ST=Seoul/L=Seoul/O=Promptech/CN=localhost' \  
19     #     -x509 -newkey rsa:4096 \  
20     #     -nodes -keyout /etc/nginx/ssl/default_key.pem \  
21     #     -keyout ${NGINX_DIR}/certs/ckan-local.key \  
22     #     -out ${NGINX_DIR}/certs/ckan-local.crt \  
23     #     -days 365 && \  
24     nginx -g 'daemon off;'
```

Comment out some lines in nginx's Dockerfile.

What the lines perform are installing openssl, making a directory for saving ssl certifications and creating the certifications using openssl.

The lines which should be commented out are marked with red boxes in the picture on the left.



### Disabling HTTPS configurations

[project directory]/.env

```
.env
.
.
.
29 # CKAN core
30 CKAN_VERSION=2.10.0
31 CKAN_SITE_ID=default
32 CKAN_SITE_URL=http://localhost:81
33 CKAN__BEAKER__SESSION__SECRET=CHANGE_ME
34 # See https://docs.ckan.org/en/latest/maintaining/configuration
35 CKAN__API_TOKEN__JWT__ENCODE__SECRET=string:CHANGE_ME
36 CKAN__API_TOKEN__JWT__DECODE__SECRET=string:CHANGE_ME
37 CKAN_SYSADMIN_NAME=ckan_admin
38 CKAN_SYSADMIN_PASSWORD=test1234
39 CKAN_SYSADMIN_EMAIL=your_email@example.com
40 CKAN_STORAGE_PATH=/var/lib/ckan
41 CKAN_SMTP_SERVER=smtp.corporateict.domain:25
42 CKAN_SMTP_STARTTLS=True
43 CKAN_SMTP_USER=user
44 CKAN_SMTP_PASSWORD=pass
45 CKAN_SMTP_MAIL_FROM=ckan@localhost
46 TZ=UTC
```

Set CKAN\_SITE\_URL to 'localhost:81' with http:// prefix. This will be the URL of our CKAN site.

Many CKAN features that need an absolute URL to our site use this value.



### Disabling HTTPS configurations

[project directory]/.env

```
CKAN_SITE_URL=http://localhost:81
```



### Increasing maximum file upload size

[project directory]/.env

```
.env
.
.
28 # CKAN core
29 CKAN_VERSION=2.10.4
30 CKAN_SITE_ID=default
31 CKAN_SITE_URL=http://192.168.110.39
32 CKAN__BEAKER__SESSION__SECRET=CHANGE_ME
33 # See https://docs.ckan.org/en/latest/maintaining/config.html#ckan.max_resource_size
34 CKAN__API_TOKEN__JWT__ENCODE__SECRET=string:CHANGE_ME
35 CKAN__API_TOKEN__JWT__DECODE__SECRET=string:CHANGE_ME
36 CKAN_SYSADMIN_NAME=ckan_admin
37 CKAN_SYSADMIN_PASSWORD=test1234
38 CKAN_SYSADMIN_EMAIL=your_email@example.com
39 CKAN_STORAGE_PATH=/var/lib/ckan
40 CKAN_SMTP_SERVER=smtp.corporateict.domain:25
41 CKAN_SMTP_STARTTLS=True
42 CKAN_SMTP_USER=user
43 CKAN_SMTP_PASSWORD=pass
44 CKAN_SMTP_MAIL_FROM=ckan@localhost
45 CKAN__MAX_RESOURCE_SIZE=256
46 TZ=UTC
```

In CKAN, by default, the maximum size of a resource can be uploaded is 10MB.

If we want to upload a file over 10MB, we can increase the maximum resource size limit by overriding 'ckan.max\_resource\_size' configuration in .env.

In this tutorial, we will increase the maximum size up to 256MB.

Add CKAN\_MAX\_RESOURCE\_SIZE key and set the value to 256.



### Increasing maximum file upload size

[project directory]/.env

```
CKAN__MAX_RESOURCE_SIZE=256
```



### Increasing maximum file upload size

[project directory]/nginx/setup/default.conf

```
default.conf ×

nginx > setup > default.conf
You, 5 days ago | 2 authors (Brett and one other)
1 server [
2   listen      80;
3   listen  [::]:80;
4   server_name  localhost;
5
6   #access_log  /var/log/nginx/host.access.log  main;
7
8   client_max_body_size 256M;
9
10  # ckan
11  location / {
12    proxy_pass http://ckan:5000/;
13    proxy_set_header X-Forwarded-For $remote_addr;
14    proxy_set_header Host $host;
15    #proxy_cache cache;
16    proxy_cache_bypass $cookie_auth_tkt;
17    proxy_no_cache $cookie_auth_tkt;
18    proxy_cache_valid 30m;
19    proxy_cache_key $host$scheme$proxy_host$request_uri;
20  }
```

Due to the Nginx default file upload limit is 1MB, we also should increase nginx's maximum upload limit up to 256MB.

Set 'client\_max\_body\_size' to 256M.



### Increasing maximum file upload size

[project directory]/nginx/setup/default.conf

```
client_max_body_size 256M;
```

## I

# Installing CKAN

Setting up with Docker Compose



## Allowing new user accounts to be created via the web UI

[project directory]/.env

```
! .env x
! .env
73 # Extensions
74 CKAN__PLUGINS="resource_proxy image_view text_view audio_view video_view datatables_view pdf_view dataexplorer_view geojson_view visualize"
75 CKAN__HARVEST__MQ__TYPE=redis
76 CKAN__HARVEST__MQ__HOSTNAME=redis
77 CKAN__HARVEST__MQ__PORT=6379
78 CKAN__HARVEST__MQ__REDIS_DB=1
79
80 # Auth
81 CKAN__AUTH__CREATE_USER_VIA_WEB=True
82
83 # Views
84 CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view dataexplorer_view geojson_view visualize
```

In CKAN, by default, user accounts can only be created by organization admin, sysadmin or using CLI command. But we might want to make new users create accounts themselves.

Then we can allow the users to create new accounts on our CKAN website.

Add CKAN\_AUTH\_CREATE\_USER\_VIA\_WEB key and set the value to True.

# I

# Installing CKAN

Setting up with Docker Compose



**Allowing new user accounts to be created via the web UI**

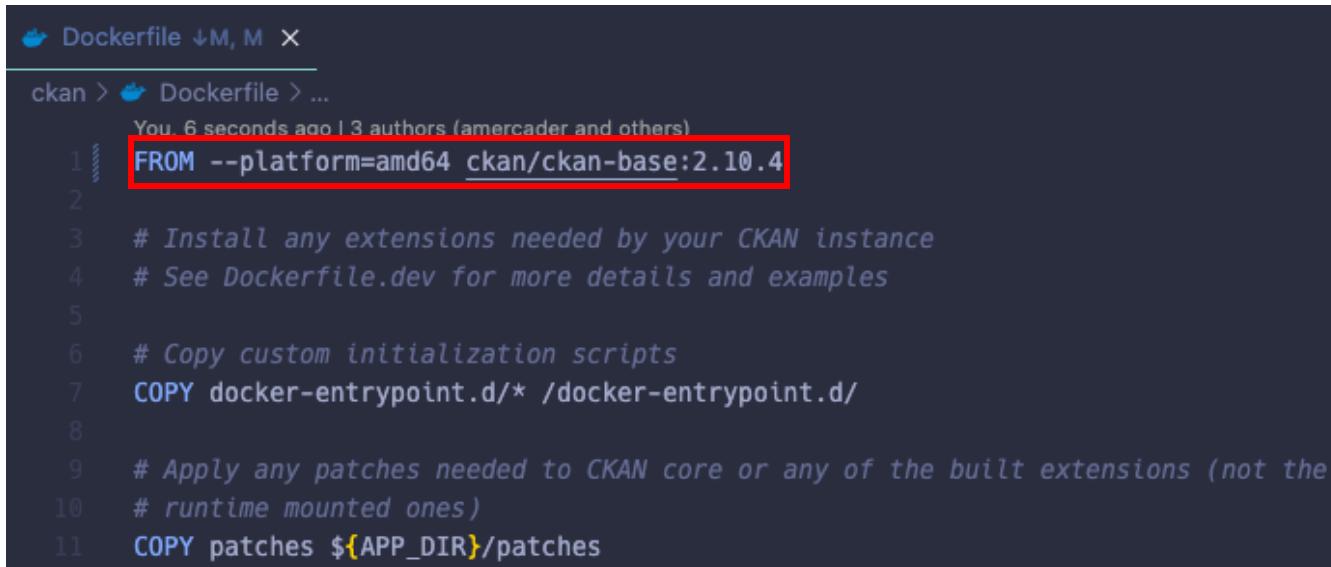
[project directory]/.env

```
CKAN__AUTH__CREATE_USER_VIA_WEB=True
```



### Building docker images

[project directory]/ckan/Dockerfile



```
Dockerfile ↵M, M ×
ckan > Dockerfile > ...
You, 6 seconds ago | 3 authors (amercader and others)
1 | FROM --platform=amd64 ckan/ckan-base:2.10.4
2 |
3 | # Install any extensions needed by your CKAN instance
4 | # See Dockerfile.dev for more details and examples
5 |
6 | # Copy custom initialization scripts
7 | COPY docker-entrypoint.d/* /docker-entrypoint.d/
8 |
9 | # Apply any patches needed to CKAN core or any of the built extensions (not the
10 | # runtime mounted ones)
11 | COPY patches ${APP_DIR}/patches
```

Running CKAN with Docker requires building the images and running the containers.

It may not be built in an ARM architecture environment (Windows on ARM or Mac computers with Apple silicon).

Therefore, you need to force the architecture of the image.

Open the Dockerfile in the ckan folder and add '--platform=amd64' to the first line.



### Building docker images

[project directory]/ckan/Dockerfile

```
FROM --platform=amd64 ckan/ckan-base:2.10.4
```



### Building docker images

[project directory]

```
sean ➤ ~/ckan-projects/ckan-docker ➤ master ± ➤ docker compose build
[+] Building 2.0s (10/10) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecc764158709b781b
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:3fdd26b9cbd3a17477001d1fd5f140486ad9ec9beb6c2a8d05e83c14bd1900c8
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 723B
```

Build the docker images by entering the 'build' command of docker compose in the CKAN project directory.

# I

# Installing CKAN

Setting up with Docker Compose



## Building docker images

[project directory]

`docker compose build`



### Running docker containers

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± docker compose up -d
[+] Running 19/19
✓ Network ckan-docker_redisnet          Created      0.0s
✓ Network ckan-docker_webnet           Created      0.0s
✓ Network ckan-docker_ckannet          Created      0.0s
✓ Network ckan-docker_dbnet            Created      0.0s
✓ Network ckan-docker_solrnet          Created      0.0s
✓ Volume "ckan-docker_site_packages"   Created      0.0s
✓ Volume "ckan-docker_pg_data"         Created      0.0s
✓ Volume "ckan-docker_solr_data"       Created      0.0s
✓ Volume "ckan-docker_ckan_storage"    Created      0.0s
✓ Volume "ckan-docker_pip_cache"       Created      0.0s
✓ Container ckan-docker-redis-1        Healthy     31.3s
✓ Container ckan-docker-solr-1         Healthy     31.3s
✓ Container ckan-docker-datapusher-1   Started      0.8s
✓ Container ckan-docker-db-1          Healthy     31.3s
! solr The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
✓ Container ckan-docker-ckan-1         Healthy     93.0s
! datapusher The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
! ckan The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
✓ Container ckan-docker-nginx-1        Started     93.0s
sean ~/ckan-projects/ckan-docker master ±
```

Run the containers by entering the up command of docker compose in the CKAN project directory.  
Add the '-d' option to make the containers run in the background.  
It will take about 2 minutes for all containers to run.



### Running docker containers

[project directory]

```
docker compose up -d
```

# Installing CKAN

Setting up with Docker Compose



## Running docker containers

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9ba6f327bc2c	ckan-docker-nginx	"/bin/sh -c 'nginx -'"	15 hours ago	Up 15 hours	0.0.0.0:81->80/tcp	ckan-docker-nginx-1
75ba358007a4	ckan-docker-ckan	"/srv/app/start_ckan..."	15 hours ago	Up 15 hours (healthy)	5000/tcp	ckan-docker-ckan-1
46973963c477	ckan/ckan-solr:2.10-solr9	"docker-entrypoint.s..."	15 hours ago	Up 15 hours (healthy)	8800/tcp	ckan-docker-solr-1
b07d10b33af0	ckan/ckan-base-datapusher:0.0.20	"sh -c 'uwsgi --plug..."	15 hours ago	Up 15 hours (unhealthy)		ckan-docker-datapusher-1
1c098375fcbe	ckan-docker-db	"docker-entrypoint.s..."	15 hours ago	Up 15 hours (healthy)		ckan-docker-db-1
dec51a4d133e	redis:6	"docker-entrypoint.s..."	15 hours ago	Up 15 hours (healthy)		ckan-docker-redis-1

When the container is executed, enter the docker's 'ps' command to check that the six containers are running normally.

# I

# Installing CKAN

Setting up with Docker Compose



## Running docker containers

```
docker ps
```



### Opening CKAN web page in the browser

<http://localhost:81/>

The screenshot shows the CKAN web interface. At the top, there is a dark header with the CKAN logo on the left and 'Log in' and 'Register' buttons on the right. Below the header, the main content area has a teal header bar with the text 'Welcome to CKAN'. The main body contains a large gray box with the text '420 x 220' and a smaller black box at the bottom left containing the text 'This is a featured section'. To the right, there is a teal sidebar with a search bar containing 'E.g. environment' and a magnifying glass icon. Below the search bar is a dark teal footer bar with the text 'Popular tags'.

Now we can open the browser, go to <http://localhost:81/>.

We can see the CKAN web page.



### Logging in to CKAN web page

<http://localhost:81/user/login>

The screenshot shows the CKAN login interface. At the top, there's a navigation bar with links for Datasets, Organizations, Groups, and About, along with a search bar. Below the navigation is a sidebar with links for 'Need an Account?', 'Create an Account', 'Forgotten your password?', and 'Forgot your password?'. The main area is titled 'Login' and contains fields for 'Username or Email:' (containing 'ckan\_admin') and 'Password:' (containing 'test1234'). A 'Remember me' checkbox is checked. A 'Login' button is located at the bottom right of the form. The entire login form is highlighted with a thick red border.

We can log in with the sysadmin account that is created by default on the website. Click the 'Login' button on the top right of the main page to go to the login page and log in to CKAN web page.

Username is 'ckan\_admin' and password is 'test1234'.



### Logging in to CKAN web page

Username: ckan\_admin

Password: test1234

# II

## Adding Extensions

1. Harvest
2. DCAT
3. Preview Extensions
4. Report



### Basic setup for ckanext-harvest

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
You, 1 second ago | 3 authors (amercader and others)
1 FROM --platform=amd64 ckan/ckan-base:2.10.4
2
3 # Install any extensions needed by your CKAN instance
4 # See Dockerfile.dev for more details and examples
5
6 # Install harvest (git ref: bf849f177cf437652944d7696fbb219cbe2d52aa)
7 RUN pip install -e "git+https://github.com/ckan/ckanext-harvest.git@bf849f177cf437652944d7696fbb219cbe2d52aa#egg=ckanext-harvest" && \
8     pip install -r ${APP_DIR}/src/ckanext-harvest/requirements.txt
9
10 # Copy custom initialization scripts
11 COPY docker-entrypoint.d/* /docker-entrypoint.d/
```

CKAN extensions can be installed using PIP. In this tutorial, every extension will be installed directly from git repository, rather than PyPi.

The ckanext-harvest extension provides a common harvesting framework for CKAN extensions and adds a CLI and a WUI to CKAN to manage harvesting sources and jobs.

Add RUN command for installing ckanext-harvest extension and requirements packages in the ckan's Dockerfile.



### Basic setup for ckanext-harvest

#### [project directory]/ckan/Dockerfile

```
# Install harvest (git ref: bf849f177cf437652944d7696fbb219cbe2d52aa)
RUN pip install -e "git+https://github.com/ckan/ckanext-harvest.git@bf849f177cf437652944d7696fbb219cbe2d52aa#egg=ckanext-harvest" && \
    pip install -r ${APP_DIR}/src/ckanext-harvest/requirements.txt
```



## Basic setup for ckanext-harvest

[project directory]/.env

```
!+ .env X
!+ .env
69 # Extensions
70 CKAN__PLUGINS="image_view text_view datatables_view datastore datapusher harvest ckan_harvester envvars"
71 CKAN__HARVEST__MQ__TYPE=redis
72 CKAN__HARVEST__MQ__HOSTNAME=redis
73 CKAN__HARVEST__MQ__PORT=6379
74 CKAN__HARVEST__MQ__REDIS_DB=1
75
76 CKAN__AUTH__CREATE_USER_VIA_WEB=True
```

To enable extension, we should add our extension to CKAN\_\_PLUGINS plugin list. Every extensions should be added before 'envvars' extension so that they are loaded correctly.

Add 'harvest ckan\_harvester' to the list.



### Basic setup for ckanext-harvest

[project directory]/.env

```
CKAN__PLUGINS="image_view text_view datatables_view datastore harvest ckan_harvester envvars"
```



### Configuration for ckanext-harvest

#### [project directory]/ckan/setup/ckan\_harvesting.conf

```
ckan> setup> ckan_harvesting.conf
5 [program:ckan_gather_consumer]
6
7 command=/usr/bin/ckan --config=/srv/app/ckan.ini harvester gather-consumer
8
9 user=ckan
10
11 numprocs=1
12 stdout_logfile=/srv/app/harvest_logs/gather_consumer.log
13 stderr_logfile=/srv/app/harvest_logs/gather_consumer.log
14 autostart=true
15 autorestart=true
16 startsecs=10
17
18 [program:ckan_fetch_consumer]
19
20 command=/usr/bin/ckan --config=/srv/app/ckan.ini harvester fetch-consumer
21
22 user=ckan
23
24 numprocs=1
25 stdout_logfile=/srv/app/harvest_logs/fetch_consumer.log
26 stderr_logfile=/srv/app/harvest_logs/fetch_consumer.log
27 autostart=true
28 autorestart=true
29 startsecs=10
```

There are two different queues harvest uses, one that handles the gathering and another one that handles the fetching and importing.

The 'gather' and 'fetch' process should be kept running somehow so we need to configure tasks with Supervisor.

Create a file named 'ckan\_harvesting.conf' in ckan's setup directory and write program settings which will describe the tasks that need to be monitored.

The log files stored by supervisor can be found in '/srv/app/harvest\_log's directory, in CKAN docker container.



### Configuration for ckanext-harvest

#### [project directory]/ckan/setup/ckan\_harvesting.conf

```
[program:ckan_gather_consumer]
command=/usr/bin/ckan --config=/srv/app/ckan.ini harvester gather-consumer
```

```
user=ckan
```

```
numprocs=1
stdout_logfile=/srv/app/harvest_logs/gather_consumer.log
stderr_logfile=/srv/app/harvest_logs/gather_consumer.log
autostart=true
autorestart=true
startsecs=10
```

```
[program:ckan_fetch_consumer]
command=/usr/bin/ckan --config=/srv/app/ckan.ini harvester fetch-consumer
```

```
user=ckan
```

```
numprocs=1
stdout_logfile=/srv/app/harvest_logs/fetch_consumer.log
stderr_logfile=/srv/app/harvest_logs/fetch_consumer.log
autostart=true
autorestart=true
startsecs=10
```



### Configuration for ckanext-harvest

[project directory]/ckan/docker-entrypoint.d/02\_setup\_harvest.sh

```
02_setup_harvest.sh ×  
ckan > docker-entrypoint.d > 02_setup_harvest.sh  
1  #!/bin/bash  
2  
3  
4 01 # Create the necessary tables in the database  
5      /usr/bin/ckan --config=/srv/app/ckan.ini db upgrade -p harvest  
6  
7 02 # Make a directory for supervisor and cron logs  
8      mkdir -p /srv/app/harvest_logs
```

02

01

After installing harvest, we need to create the necessary tables in the CKAN main database.

To do it automatically, create shell script file named '02\_setup\_harvest.sh' in ckan's 'docker-entrypoint.d' directory and add 'ckan db upgrade' command for harvest.

02

In the previous step, we created supervisor task file and set log output path.

Add a command for creating the directory which the log files will be saved.



### Configuration for ckanext-harvest

#### [project directory]/ckan/docker-entrypoint.d/02\_setup\_harvest.sh

```
#!/bin/bash

# Create the necessary tables in the database
/usr/bin/ckan --config=/srv/app/ckan.ini db upgrade -p harvest

# Make a directory for supervisor and cron logs
mkdir -p /srv/app/harvest_logs
```



### Configuration for ckanext-harvest

#### [project directory]/ckan/docker-entrypoint.d/02\_setup\_harvest.sh

```
ckan > docker-entrypoint.d > 02_setup_harvest.sh
  6  # Make a directory for supervisor and cron logs
  7  mkdir -p /srv/app/harvest_logs
  8
  9  # Start the supervisor tasks
10  supervisorctl reread
11  supervisorctl add ckan_gather_consumer
12  supervisorctl add ckan_fetch_consumer
13  supervisorctl start ckan_gather_consumer
14  supervisorctl start ckan_fetch_consumer
15
16
17
18
19  # Create cron job to run 'harvester run' command (every minute)
20  ( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini harvester run >> /srv/app/harvest_logs/run.log 2>&1" ) | crontab -
```

01

```
supervisorctl reread
supervisorctl add ckan_gather_consumer
supervisorctl add ckan_fetch_consumer
supervisorctl start ckan_gather_consumer
supervisorctl start ckan_fetch_consumer
```

02

```
/usr/sbin/crond start
```

01

For running the supervisor tasks, the supervisor service should reload configuration files, activate any updates from configuration and start the processes which we added as program.

Add 'supervisorctl reread', 'add' and 'start' commands.

02

Once we have the two consumers running and monitored, we just need to create a cron scheduling job that will run the 'harvester run' command periodically.

Add commands for starting cron daemon and creating cron job which runs harvester every minute.



### Configuration for ckanext-harvest

#### [project directory]/ckan/docker-entrypoint.d/02\_setup\_harvest.sh

```
# Start the supervisor tasks
supervisorctl reread
supervisorctl add ckan_gather_consumer
supervisorctl add ckan_fetch_consumer
supervisorctl start ckan_gather_consumer
supervisorctl start ckan_fetch_consumer

# Start crond
/usr/sbin/crond start

# Create cron job to run 'harvester run' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini harvester run >> /srv/app/harvest_logs/run.log 2>&1" ) | crontab -
```



### Configuration for ckanext-harvest

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
10  # Copy custom initialization scripts
11  COPY docker-entrypoint.d/* /docker-entrypoint.d/
12
13  # Copy the supervisor configuration for harvesting
14  COPY setup/ckan_harvesting.conf /etc/supervisord.d/ckan_harvesting.conf
15
16  # Apply any patches needed to CKAN core or any of the built extensions (not the
17  # runtime mounted ones)
18  COPY patches ${APP_DIR}/patches
```

The supervisor configuration file which we created for harvesting won't be copied to the ckan docker container automatically. So, we have to add COPY command that copies the supervisor configuration file from ckan's setup directory to the docker container's '/etc/supervisord.d' directory in the ckan's Dockerfile.



### Configuration for ckanext-harvest

#### [project directory]/ckan/Dockerfile

```
# Copy the supervisor configuration for harvesting
COPY setup/ckan_harvesting.conf /etc/supervisord.d/ckan_harvesting.conf
```



## Installing ckanext-harvest

[project directory]/install-extensions.sh

```
install-extensions.sh U X
install-extensions.sh
1  #!/bin/sh
2
3  docker compose down
4
5  docker volume rm ckan-docker_site_packages
6
7  docker compose build
8
9  docker compose up -d
10
```

When we add a new extension, we should build docker images and run again.

However, the storage volume where the extension is stored is not automatically renewed, resulting in an error that the extension is not found.

Therefore, we need to stop the container and remove its volume before building, and then build and run the image.

Write a script named 'install-extensions.sh' in the CKAN project directory to do this all at once.



### Installing ckanext-harvest

[project directory]/install-extensions.sh

```
#!/bin/sh

docker compose down

docker volume rm ckan-docker_site_packages

docker compose build

docker compose up -d
```



## Installing ckanext-harvest

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1     Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet      Removed
✓ Network ckan-docker_dbnet        Removed
✓ Network ckan-docker_webnet      Removed
✓ Network ckan-docker_solrnet     Removed
✓ Network ckan-docker_redisnet    Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

After writing the script, run it using the sh command.



### Installing ckanext-harvest

[project directory]

```
sh install-extensions.sh
```



### Usage of ckanext-harvest

<http://localhost:81/harvest>

The screenshot shows the CKAN Harvest Sources interface. At the top, there's a navigation bar with links for Datasets, Organizations, Groups, About, and a search bar. A user icon labeled 'ckan\_admin' is visible. On the left, there are two filter sections: 'Frequency' (showing no results) and 'Type' (showing no results). In the center, a large button says 'Add Harvest Source'. Below it is a search bar with placeholder text 'Search harvest sources...'. A message 'No harvests found' is displayed, along with an 'Order by' dropdown set to 'Relevance'. At the bottom, there are links for 'About CKAN', 'CKAN API', 'CKAN Association', and an 'OPEN DATA' button. The footer includes 'Powered by CKAN' and a language selector set to 'English'.

After logging in as sysadmin, the harvest source listing should be available at <http://localhost:81/harvest>.



## Usage of ckanext-harvest

<http://localhost:81/organization>

The screenshot shows the CKAN interface for managing organizations. At the top, there's a navigation bar with links for Datasets, Organizations (which is highlighted with a red box), Groups, and About. A search bar is also present. Below the navigation, the URL 'http://localhost:81/organization' is shown. The main content area has a sidebar on the left with information about what organizations are used for. The main panel displays a large 'Add Organization' button with a blue box around it. Below this button is a search bar and a message stating 'No organizations found'. At the bottom of the panel, there's a note: 'There are currently no organizations for this site. [How about creating one?](#)'.

Before adding a harvest source, we must create an organization where the datasets harvested from the harvest source will be stored.

Browse to <http://localhost:81/organization> and click the 'Add Organization' button to go to the Organization Creation page.



## Usage of ckanext-harvest

<http://localhost:81/organization/new>

Home / Organizations / Create an Organization

What are Organizations?

CKAN Organizations are used to create, manage and publish collections of datasets. Users can have different roles within an Organization, depending on their level of authorisation to create, edit and publish.

### Create an Organization

Name:  \* URL: localhost:81/organization/my-organization Edit

Description:   
You can use Markdown formatting [here](#)

Name:  \* URL: localhost:81/organization/my-organization

Image: Upload Link

Create an organization with an organization 'name', 'description' and 'image'.



### Usage of ckanext-harvest

<http://localhost:81/harvest>

The screenshot shows the CKAN Harvest Sources page at <http://localhost:81/harvest>. The main content area displays a message: "No harvests found". On the left, there are two filter sections: "Frequency" and "Type", both showing no results. A search bar is present above the main content. A prominent blue button labeled "Add Harvest Source" is centered in the main area. To its left, in the sidebar, another "Add Harvest Source" button is highlighted with a red border.

After creating the organization, go back to the Harvest page.

We can now create a harvest source by clicking 'Add Harvest Source' button.



### Usage of ckanext-harvest

<http://localhost:81/harvest/new>

Home / Harvest Sources / Create Harvest Source

**Harvest sources**

Harvest sources allow importing remote metadata into this catalog. Remote sources can be other catalogs such as other CKAN instances, CSV servers or Web Accessible Folders (WAF) (depending on the actual harvesters enabled for this instance).

**URL:** https://laos-ckan-dev.promptech.co.kr/  
This should include the http:// part of the URL

**Title:** Remote CKAN Source

**Description:** Remote CKAN Source  
URL: localhost:81/harvest/remote-ckan-source

**Update frequency:**

- ✓ Manual
- Monthly
- Weekly
- Biweekly
- Daily
- Always

**Source type:** CKAN

**Organization:** my-organization

Create a harvest source with 'URL', 'title', 'update frequency' and 'organization'.

In this tutorial, we will harvest the datasets from <https://laos-ckan-dev.promptech.co.kr/>, so the 'URL' would be that.

The 'update frequency' means how often to run harvesting. For the tutorial, select 'Manual'.

The 'organization' will be the one we just created before.



### Usage of ckanext-harvest

<http://localhost:81/harvest/remote-ckan-source>

The screenshot shows the CKAN Harvest Sources interface. The URL is <http://localhost:81/harvest/remote-ckan-source>. The page title is "Remote CKAN Source". On the left, there is a sidebar with a "Datasets" section showing "0" datasets. The main content area has tabs for "Datasets" and "About", with "Datasets" selected. It displays the message "There are no datasets associated to this harvest source." In the top right corner, there is an "Admin" button, which is highlighted with a red box.

After creating the harvest source, we need to run the job so that we can harvest.  
Go to the admin page by clicking the 'Admin' button in the top right corner.



## Usage of ckanext-harvest

<http://localhost:81/harvest/admin/remote-ckan-source>

The screenshot shows the CKAN Harvest Admin interface. At the top, there's a navigation bar with links to Home, Organizations, My Organization, Harvest Sources, Remote..., and Admin. Below this, on the left, is a sidebar for a 'Remote CKAN Source' with a note that there is no description for this harvest source. It also shows 'Datasets' count as 0. In the center, a section titled 'Last Harvest Job' displays a message that says 'No jobs yet for this source'. At the top right of the main content area, there are several buttons: 'Dashboard', 'Jobs', 'Edit', 'Reharvest' (which is highlighted with a red box), 'Clear', and 'View harvest source'. A modal dialog box is open over the content, titled 'Please Confirm Action'. The dialog contains a message about re-running the harvesting process and its consequences. At the bottom of the dialog are 'Cancel' and 'Confirm' buttons, with 'Confirm' also highlighted with a red box.

In the harvest admin page, click 'Reharvest' button and a pop-up dialog will appear. Then click 'Confirm' to run the harvest job.



### Usage of ckanext-harvest

<http://localhost:81/harvest/admin/remote-ckan-source>

The screenshot shows the CKAN Harvest Admin interface at the URL <http://localhost:81/harvest/admin/remote-ckan-source>. The page title is "Remote CKAN Source". The top navigation bar includes links for "Dashboard", "Jobs", "Edit", "Reharvest", "Clear", and a redboxed "View harvest source" button. Below the navigation is a section titled "Last Harvest Job" with a red border around its content. Inside this section, it displays "0 errors 39 added 0 updated 0 deleted 0 not modified". A table follows with the following data:

ID	Value
Created	October 15, 2024, 5:27 PM (UTC+09:00)
Started	October 15, 2024, 5:27 PM (UTC+09:00)
Finished	October 15, 2024, 5:27 PM (UTC+09:00)
Status	Finished

At the bottom of the "Last Harvest Job" section is a "View full job report" button.

When a harvesting job runs, the Dashboard displays status information about the job.

We can see the number of datasets harvested, job start/end time, job status, error/success count, etc.

Since we set up a cron job to run every minute to execute the 'havester run' command, the job should finish after about a minute.

Now we can go back to the Harvest Source page by clicking the 'View harvest source' button.



### Usage of ckanext-harvest

<http://localhost:81/harvest/remote-ckan-source>

The screenshot shows the CKAN Harvest Source page for the organization 'Laos CKAN'. The URL is <http://localhost:81/harvest/remote-ckan-source>. The page displays harvested datasets:

- Apprenticeship and traineeship completions**: Spreadsheets containing apprentice and trainee completions in Qld in the specified calendar year - showing the industry training group (ITG), qualification level, DESBT region,....  
Format: PDF  
Link: [PDF](#)
- Dry Spent Fuel Storage Designs: NRC-Approved for Use by General Licensees**: Dry Spent Fuel Storage Designs: NRC-Approved for Use by General Licensees  
Format: XLSX  
Link: [XLSX](#)
- U.S. Nuclear Power Plant Inspection Reports**: Official Correspondence of Nuclear Reactor Inspection Reports  
Format: XLSX  
Link: [XLSX](#)
- U.S. Department of Defense Budget Estimates FY 2002**: National Defense Budget Estimates for the FY 2002 Budget (Green Book). Summary reference source for the National Defense budget estimates for FY 2002.  
Format: PDF  
Link: [PDF](#)

We can see harvested datasets on the Harvest Source page.



## Basic setup for ckanext-dcat and ckanext-scheming

### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
6 # Install harvest (git ref: bf849f177cf437652944d7696fbb219cbe2d52aa)
7 ~ RUN pip install -e "git+https://github.com/ckan/ckanext-harvest.git@bf849f177cf437652944d7696fbb219cbe2d52aa#egg=ckanext-harvest" && \
8     pip install -r ${APP_DIR}/src/ckanext-harvest/requirements.txt
9
10 # Install dcat (git ref: 51d651315ff2ff594ebd6eb1d658bc7e9553310c)
11 ~ RUN pip install -e "git+https://github.com/ckan/ckanext-dcat.git@51d651315ff2ff594ebd6eb1d658bc7e9553310c#egg=ckanext-dcat" && \
12     pip install -r ${APP_DIR}/src/ckanext-dcat/requirements.txt
13
14 # Install scheming (git ref: 27035f4d5b3722c2bc64d39b6c2b1d76c9883636)
15 RUN pip install -e "git+https://github.com/ckan/ckanext-scheming.git@27035f4d5b3722c2bc64d39b6c2b1d76c9883636#egg=ckanext-scheming"
```

The ckanext-dcat extension offers pre-built CKAN schemas for common 'Application Profiles' that can be adapted to each site requirements to provide out-of-the-box DCAT support in data portals, including tailored fields, validation etc.

And the ckanext-scheming provides a way to configure and share metadata schemas using a YAML or JSON schema description.

Add RUN command for installing ckanext-dcat extension and requirements packages.

Then add RUN command for installing ckanext-scheming. Through installing scheming extension, we can use pre-built schemas that enable DCAT support.



## Basic setup for ckanext-dcat and ckanext-scheming

### [project directory]/ckan/Dockerfile

```
# Install dcat (git ref: 51d651315ff2ff594ebd6eb1d658bc7e9553310c)
RUN pip install -e "git+https://github.com/ckan/ckanext-dcat.git@51d651315ff2ff594ebd6eb1d658bc7e9553310c#egg=ckanext-dcat" && \
    pip install -r ${APP_DIR}/src/ckanext-dcat/requirements.txt

# Install scheming (git ref: 27035f4d5b3722c2bc64d39b6c2b1d76c9883636)
RUN pip install -e "git+https://github.com/ckan/ckanext-scheming.git@27035f4d5b3722c2bc64d39b6c2b1d76c9883636#egg=ckanext-
scheming"
```



## Basic setup for ckanext-dcat and ckanext-scheming

[project directory]/.env

```
.env      X
.env
76  # Extensions
77  CKAN__PLUGINS="datastore datapusher harvest ckan_harvester dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface structured_data scheming_datasets envvars"
78  CKAN__HARVEST__MQ__TYPE=redis
79  CKAN__HARVEST__MQ__HOSTNAME=redis
80
81 dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface structured_data scheming_datasets
```

Add 'dcat dcat\_rdf\_harvester dcat\_json\_harvester dcat\_json\_interface structured\_data scheming\_datasets' to the CKAN\_\_PLUGINS plugin list.

The 'dcat\_rdf\_harvest' and 'dcat\_json\_harvester' extensions are optional. They allow us to use DCAT RDF and DCAT JSON source types when we harvest datasets from other harvest sources.



## Basic setup for ckanext-dcat and ckanext-scheming

[project directory]/.env

```
CKAN__PLUGINS="image_view text_view datatables_view datastore datapusher harvest ckan_harvester dcat dcat_rdf_harvester  
dcat_json_harvester dcat_json_interface structured_data scheming_datasets envvars"
```



## Configuration for ckanext-dcat

### [project directory]/ckan/Dockerfile

```
● Dockerfile ↓M, M X
ckan > ● Dockerfile > ...
19
20 # Copy the supervisor configuration for harvesting
21 COPY setup/ckan_harvesting.conf /etc/supervisord.d/ckan_harvesting.conf
22
23 # Comment out extra key validation for RDF schema
24 RUN sed -i '959,961s/^    #' /srv/app/src/ckan/ckan/logic/validators.py
25 RUN sed -i '958a\    pass' /srv/app/src/ckan/ckan/logic/validators.py
26
27 # Apply any patches needed to CKAN core or any of the built extensions (not the
28 # runtime mounted ones)
29 COPY patches ${APP_DIR}/patches
```

When harvesting in DCAT RDF format, a validation error will occur due to the presence of an 'extra' key in the schema. This appears to be a bug in CKAN and can be temporarily fixed by commenting out the code that performs the validation. Add RUN command in the ckan's Dockerfile that comments out the validation code.



## Configuration for ckanext-dcat

### [project directory]/ckan/Dockerfile

```
# Comment out extra key validation for RDF schema  
RUN sed -i '959,961s/^/#/' /srv/app/src/ckan/ckan/logic/validators.py  
RUN sed -i '958a\\ pass' /srv/app/src/ckan/ckan/logic/validators.py
```



## Configuration for ckanext-scheming

[project directory]/.env

```
![[ .env ]]  
![[ .env ]]  
89  # Scheming  
90  CKAN__SCHEMING__DATASET_SCHEMAS=ckanext.dcat.schemas:dcat_ap_2.1_recommended.yaml  
91  CKAN__SCHEMING__PRESETS="ckanext.scheming:presets.json ckanext.dcat.schemas:presets.yaml"  
92  CKANEXT__DCAT__RDF__PROFILES="euro_dcat_ap_2 euro_dcat_ap_scheming"  
-]
```

For using scheming, we should define which schema file will be used. It can be done by adding three options to .env file.

We will use 'dcat\_ap\_2.1\_recommended.yaml' file which includes the recommended properties according to the DCAT 2.1 specification, so add CKAN\_SCHEMING\_DATASET\_SCHEMAS key and set the value to 'ckanext.dcat.schemas:dcat\_ap\_2.1\_recommended.yaml'.

Then for presets, we will use the dcat presets as well as the standard scheming ones. Add CKAN\_SCHEMING\_PRESETS key and set the value to 'ckanext.scheming:presets.json ckanext.dcat.schemas:presets.yaml'.

Lastly, for profiles, add CKANEXT\_DCAT\_RDF\_PROFILES key and set the value to 'euro\_dcat\_ap\_2 euro\_dcat\_ap\_scheming'. The profiles allow customization of how the values defined in the RDF graph are mapped to CKAN and vice versa.



## Configuration for ckanext-scheming

[project directory]/.env

```
# Scheming
CKAN__SCHEMING__DATASET_SCHEMAS=ckanext.dcat.schemas:dcat_ap_2.1_recommended.yaml
CKAN__SCHEMING__PRESETS="ckanext.scheming:presets.json ckanext.dcat.schemas:presets.yaml"
CKANEXT__DCAT__RDF__PROFILES="euro_dcat_ap_2 euro_dcat_ap_scheming"
```



## Configuration for ckanext-scheming

### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
23 # Comment out extra key validation for RDF schema
24 RUN sed -i '959,961s/^    #' /srv/app/src/ckan/ckan/logic/validators.py
25 RUN sed -i '958a\\    pass' /srv/app/src/ckan/ckan/logic/validators.py
26
27 # Handling the absence of extras field in the ckanext-scheming plugin
28 RUN sed -i "282,285s/data_dict\\['extras'\\]/data_dict.get('extras', [])/g" /srv/app/src/ckanext-scheming/ckanext/scheming/plugins.py
29 RUN sed -i "287s/data_dict\\['extras'\\]/data_dict.get('extras', [])/g" /srv/app/src/ckanext-scheming/ckanext/scheming/plugins.py
```

There is an issue with the ckanext-scheming extension where an 'extras' field key error occurs when a resource is uploaded to the datastore.

Therefore, we need to modify the extension's code so that an exception is not thrown even if the 'extras' field key does not exist.

Add RUN command in the ckan's Dockerfile that handles the absence of the extras field.



## Configuration for ckanext-scheming

### [project directory]/ckan/Dockerfile

```
# Handling the absence of extras field in the ckanext-scheming plugin
RUN sed -i "282,285s/data_dict\[\'extras\'\]/data_dict.get('extras', [])/g" /srv/app/src/ckanext-scheming/ckanext/scheming/plugins.py
RUN sed -i "287s/data_dict\[\'extras\'\]/data_dict.get('extras', [])/g" /srv/app/src/ckanext-scheming/ckanext/scheming/plugins.py
```



## Installing ckanext-dcat and ckanext-scheming

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.



## Installing ckanext-dcat and ckanext-scheming

[project directory]

```
sh install-extensions.sh
```



## Usage of ckanext-dcat and ckanext-scheming

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar includes links for Datasets (highlighted with a red box), Organizations, Groups, About, and a search bar. The main content area displays a sidebar with sections for Organizations, Groups, and Tags, each stating "There are no [category] that match this search". The main content area features two prominent "Add Dataset" buttons: one in a red box and another in a blue box. Below these buttons is a search bar and a "No datasets found" message. A "Relevance" dropdown menu is also visible.

Go to the Datasets page and click the 'Add Dataset' button to go to the Dataset Creation page.



## Usage of ckanext-dcat and ckanext-scheming

<http://localhost:81/dataset/new>

Contact points:

Contact point  
URI:  Remove

Name:

Email:

+ Add

① Contact information for enquiries about the dataset.

Publisher:

Publisher  
URI:  Remove

Name:

Email:

URL:

Type:

Temporal coverage:

Temporal coverage  
Start:  Remove

End:  Remove

+ Add

① The temporal period or periods the dataset covers.

Spatial coverage:

Spatial coverage  
URI:  Remove

Label:

Geometry:

Bounding Box:

Centroid:

When entering the metadata for the dataset, we can see that there are some new fields added.

These are the fields that correspond to the schema used by DCAT AP version 2.1.



## Usage of ckanext-dcat and ckanext-scheming

<http://localhost:81/harvest/new>

[Home](#) / Harvest Sources / Create Harvest Source

**Harvest sources**

Harvest sources allow importing remote metadata into this catalog. Remote sources can be other catalogs such as other CKAN instances, CSW servers or Web Accessible Folders (WAF) (depending on the actual harvesters enabled for this instance).

**URL:**  This should include the http:// part of the URL.

**Title:**  URL: localhost:81/harvest/rdf-harvest-source [Edit](#)

**Description:**

Generic DCAT RDF Harvester [?](#)  DCAT JSON Harvester [?](#)

**Source type:**  CKAN [?](#)  Generic DCAT RDF Harvester [?](#)  DCAT JSON Harvester [?](#)

**Update frequency:**

And now, when we create a harvest source, we can select either RDF or JSON type harvester to run harvesting.



### Enabling built-in preview extensions

[project directory]/.env

```
.env x  
.  
# Extensions  
CKAN__PLUGINS="image_view text_view audio_view video_view datatables_view"  
CKAN__HARVEST__MQ_TYPE=redis  
  
CKAN__PLUGINS="image_view text_view audio_view video_view datatables_view"  
  
CKAN__HARVEST__MQ__REDIS_DB=1  
82  
83 # Auth  
  
CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view  
  
86 # Views  
87 CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view
```

There are some data preview extensions which are packaged with CKAN, they can simply be enabled by adding to the ckan.plugins setting.

The extensions that are currently available and not deprecated are 'datatables\_view', 'text\_view', 'image\_view', 'video\_view', 'audio\_view' and 'webpage\_view'.

Add the extensions we want to use to CKAN\_PLUGINS in .env file.

If we want the previews to be created automatically when the resources are uploaded so that the data can be visualized, we can do so by configuring with the ckan.views.default\_views setting.

Add the extensions to enable by default to CKAN\_VIEWS\_DEFAULT\_VIEWS in .env file.



### Enabling built-in preview extensions

[project directory]/.env

```
# Extensions
CKAN__PLUGINS="image_view text_view audio_view video_view datatables_view webpage_view"

# Views
CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view webpage_view
```



### Basic setup for ckanext-pdfview

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile ↓M, M ×  
ckan > Dockerfile > ...  
14 | # Install scheming (git ref: 27035f4d5b3722c2bc64d39b6c2b1d76c9883636)  
15 | RUN pip install -e "git+https://github.com/ckan/ckanext-scheming.git@27035f4d5b3722c2bc64d39b6c2b1d76c9883636#egg=ckanext-scheming"  
16 |  
17 | # Install pdfview (git ref: 95fe59f0d63bb076a9c9b0ffff5a64c0524d5509)  
18 | RUN pip install -e "git+https://github.com/ckan/ckanext-pdfview.git@95fe59f0d63bb076a9c9b0ffff5a64c0524d5509#egg=ckanext-pdfview"  
19 |  
20 | # Copy custom initialization scripts  
21 | COPY docker-entrypoint.d/* /docker-entrypoint.d/  
22 |  
23 | # Copy the supervisor configuration for harvesting  
24 | COPY setup/ckan_harvesting.conf /etc/supervisord.d/ckan_harvesting.conf
```

There are many more preview extensions developed by the CKAN Community.

As one of them, ckanext-pdfview provides a view plugin for PDF files using and html object tag.

Add RUN command for installing ckanext-pdfview extension in the ckan's Dockerfile.



### Basic setup for ckanext-pdfview

#### [project directory]/ckan/Dockerfile

```
# Install pdfview (git ref: 95fe59f0d63bb076a9c9b0ffff5a64c0524d5509)
RUN pip install -e "git+https://github.com/ckan/ckanext-pdfview.git@95fe59f0d63bb076a9c9b0ffff5a64c0524d5509#egg=ckanext-pdfview"
```



### Basic setup for ckanext-pdfview

[project directory]/.env

```

# Extensions
CKAN__PLUGINS="image_view text_view datatables_view pdf_view resource_proxy datastore datapusher harvest"
CKAN__HARVEST__MQ__TYPE=redis
CKAN__HARVEST__MQ__HOSTNAME=redis
CKAN__HARVEST__MQ__PORT=6379
CKAN__HARVEST__MQ__REDIS_DB=1

# Auth
CKAN__AUTH__CREATE_USER_VIA_WEB=True

# Views
CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view
```

Just like the built-in preview extension, add the 'pdf\_view' extension to CKAN\_\_PLUGINS in .env file.

If we want to preview PDF files which are not located in the same server as CKAN we also need to enable the resource proxy plugin. Add 'resource\_proxy' to the CKAN\_\_PLUGINS plugin list of .env file.

Then add 'pdf\_view' to enable by default to CKAN\_\_VIEWS\_\_DEFAULT\_VIEWS in .env file.



### Basic setup for ckanext-pdfview

[project directory]/.env

```
# Extensions
CKAN__PLUGINS="image_view text_view datatables_view pdf_view resource_proxy datastore datapusher harvest ckan_harvester dcat
dcat_rdf_harvester dcat_json_harvester dcat_json_interface structured_data scheming_datasets envvars"

# Views
CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view
```



### Installing ckanext-pdfview

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.



## Installing ckanext-pdfview

[project directory]

```
sh install-extensions.sh
```



### Usage of ckanext-pdfview

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar includes links for Datasets (highlighted with a red box), Organizations, Groups, and About, along with a search bar. On the left, there are filters for Organizations (Evans LLC - 3), Groups (none found), and Tags (none found). The main content area displays a message: "3 datasets found" (Order by: Relevance) and "Season six whether maintain box. Table accept which reflect glass grow would painting." Below this, there are two prominent buttons: a red-bordered "Add Dataset" button and a larger blue-outlined "Add Dataset" button.

To add a PDF resource dataset, click the 'Add Dataset' button on the Datasets page.



### Usage of ckanext-pdfview

<http://localhost:81/dataset/apprenticeship-and-traineeship-completions/resource/new>

Home / Organizations / My Organization / Apprenticeship and... / Edit / Add New Resource

1 What's a resource?

A resource can be any file or link to a file containing useful data.

1 Create Dataset 2 Add data

File: 파일 선택 contextual-information-about-apprenticeship-and-traineeship-completions.pdf Remove

Name: contextual-information-about-apprenticeship-and-traineeship-completions.pdf

① A descriptive title for the resource.

contextual-information-about-apprenticeship-and-traineeship-completions.pdf

You can use [Markdown formatting here](#)

② A free-text account of the resource.

Format: pdf

③ File format. If not provided it will be guessed.

Availability:

④ Indicates how long it is planned to keep the resource available.

License: Creative Commons Attribution 4.0

⑤ License in which the resource is made available. If not provided will be inherited from the dataset.

Enter the metadata for the dataset and go to the page where we add resources.

Since we're adding a PDF resource, select the PDF file to upload in the 'File' field.

Then enter information about the resource, such as 'format', 'license', etc. and submit.

The sample PDF file can be download at <https://laos-ckan-dev.promptech.co.kr/dataset/apprenticeship-and-traineeship-completions/resource/b5d7387b-b81a-4f84-bbfc-02e38c2d18f3>.

# Adding Extensions

## Preview Extensions



### Usage of ckanext-pdfview

<http://localhost:81/dataset/apprenticeship-and-traineeship-completions/resource/b0ceb7bf-561f-4ab5-ad00-3dab410a716f>

/ Organizations / My Organization / Apprenticeship and... / contextual-information-about-apprenticeship-and-traineeship-completions-data-dictionary

**contextual-information-about-apprenticeship-and-traineeship-completions-data-dictionary**

URL: <http://localhost:81/dataset/d3ae864a-f376-4582-93c3-2b271c3fd4eb/resource/b0ceb7bf-561f-4ab5-ad00-3dab410a716f/download/contextual-information-about-apprenticeship-and-traineeship-completions-data-dictionary.pdf>

Notes on the apprentice and trainee completed dataset

PDF

Fullscreen Embed

DELTA Field Label	Description of field data
Training Agreement Type	Identifies whether the training contract is a Traineeship or an apprenticeship.
ITG Name	Identifies the industry the training contract is in, e.g., Primary Industry.
Qualification Level	Uniquely identifies the category of the recognised qualification awarded to an individual on successful completion of a qualification or course.
Employer Region	Identifies which DYESTB region the Employer's workplace is located
Provider Type	Identifies the type of Training Provider delivering the off the Job training

New commencements - Apprentices and/or trainees commencing work within the specified timeframe where DYESTB has been notified via the lodgement of a training contract. It does not include apprentices or trainees who have commenced employment in the specified time period but who have not as yet lodged a training contract due to probationary or other reasons. Also note that recommencements are not included in this data.

When we view the resource which we just added, the PDF viewer is now active.

So, we can preview its content.



### Basic setup for ckanext-dataexplorer-react

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile <M, M X
ckan > Dockerfile > ...
17 | # Install pdfview (git ref: 95fe59f0d63bb076a9c9b0ffff5a64c0524d5509)
18 | RUN pip install -e "git+https://github.com/ckan/ckanext-pdfview.git@95fe59f0d63bb076a9c9b0ffff5a64c0524d5509#egg=ckanext-pdfview"
19 |
20 | # Install dataexplorer-react (git ref: d8f49dd0f920e34a7d3437be454f20fa5f974587)
21 | RUN pip install -e "git+https://github.com/datopian/ckanext-dataexplorer-react.git@d8f49dd0f920e34a7d3437be454f20fa5f974587#egg=ckanext-dataexplorer-react"
22 |
23 | # Copy custom initialization scripts
24 | COPY docker-entrypoint.d/* /docker-entrypoint.d/
25 |
```

The ckanext-dataexplorer-react extension is a modern data explorer which provides multi-view visualization table, chart and map.

Add RUN command for installing ckanext-dataexplorer-react extension in the ckan's Dockerfile.



## Basic setup for ckanext-dataexplorer-react

[project directory]/ckan/Dockerfile

```
# Install dataexplorer-react (git ref: d8f49dd0f920e34a7d3437be454f20fa5f974587)
RUN pip install -e "git+https://github.com/datopian/ckanext-dataexplorer-
react.git@d8f49dd0f920e34a7d3437be454f20fa5f974587#egg=ckanext-dataexplorer-react"
```



### Basic setup for ckanext-dataexplorer-react

[project directory]/.env

```
++ .env  x
++ .env
70 # Extensions
71 CKAN__PLUGINS="image_view text_view datatables_view pdf_view dataexplorer_view resource_proxy datastore datapusher harvest
72 CKAN__HARVEST__MQ__TYPE=redis
73 CKAN__HARVEST__MQ__HOSTNAME=redis
74 CKAN__HARVEST__MQ__PORT=6379
75 CKAN__HARVEST__MQ__REDIS_DB=1
76
77 # Auth
78 CKAN__AUTH__CREATE_USER_VIA_WEB=True
79
80 # Views
81 CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view dataexplorer_view
```

Add the 'dataexplorer\_view' extension to CKAN\_\_PLUGINS in .env file.

Then add 'dataexplorer\_view' to enable by default to CKAN\_\_VIEWS\_\_DEFAULT\_VIEWS in .env file.



## Basic setup for ckanext-dataexplorer-react

[project directory]/.env

```
# Extensions
CKAN__PLUGINS="image_view text_view datatables_view pdf_view dataexplorer_view resource_proxy datastore datapusher harvest
ckan_harvester dcat dcat_rdf harvester dcat_json_harvester dcat_json_interface structured_data scheming_datasets envvars"

# Views
CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view dataexplorer_view
```



### Configuration for ckanext-dataexplorer-react

#### [project directory]/ckan/Dockerfile

```
.env .env  
59 # Datapusher  
60 DATAPUSHER_VERSION=0.0.20  
61 CKAN_DATAPUSHER_URL=http://datapusher:8800  
62 CKAN__DATAPUSHER__CALLBACK_URL_BASE=http://ckan:5000  
63 DATAPUSHER_REWRITE_RESOURCES=True  
64 DATAPUSHER_REWRITE_URL=http://ckan:5000  
65  
66 # Datastore  
67 CKAN__DATASTORE__SQLSEARCH__ENABLED=True  
68  
69 # NGINX  
70 NGINX_PORT=80  
71 NGINX_SSLPORT=443
```

Since the dataexplorer-react extension calls datastore\_search\_sql API for filtering data, we should enable the 'datastore\_search\_sql()' action function.

Add CKAN\_DATASTORE\_SQLSEARCH\_ENABLED key and set the value to True.



## Configuration for ckanext-dataexplorer-react

[project directory]/ckan/Dockerfile

```
# Datastore  
CKAN__DATASTORE__SQLSEARCH__ENABLED=True
```



### Configuration for ckanext-dataexplorer-react

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
33 # Handling the absence of extras field in the ckanext-scheming plugin
34 RUN sed -i "282,285s/data_dict\[\'extras\'\]/data_dict.get('extras', [])/g" /srv/app/src/ckanext-scheming/ckanext/scheming/plugins.py
35 RUN sed -i "287s/data_dict\[\'extras\'\]/data_dict.get('extras', [])/g" /srv/app/src/ckanext-scheming/ckanext/scheming/plugins.py
36
37 # Allow 'over' function when using the datastore_search_sql API
38 RUN sed -i '180a\over' /srv/app/src/ckan/ckanext/datastore/allowed_functions.txt
39
40 # Apply any patches needed to CKAN core or any of the built extensions (not the
41 # runtime mounted ones)
42 COPY patches ${APP_DIR}/patches
```

In CKAN 2.10, we need to make sure that 'over' function of PostgreSQL is enabled via datastore\_search\_sql endpoint. To do so, we should add 'over' function into Datastore's allowed function lists. We will modify the 'allowed\_functions.txt' file of the datastore. Add RUN command to the ckan's Dockerfile that allows the datastore's 'over' function.



## Configuration for ckanext-dataexplorer-react

### [project directory]/ckan/Dockerfile

```
# Allow 'over' function when using the datastore_search_sql API  
RUN sed -i '180a\over' /srv/app/src/ckan/ckanext/datastore/allowed_functions.txt
```



### Installing ckanext-dataexplorer-react

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.

Preview Extensions



## Installing ckanext-dataexplorer-react

[project directory]

```
sh install-extensions.sh
```

# Adding Extensions

Preview Extensions



## Usage of ckanext-dataexplorer-react

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar includes links for Datasets (highlighted with a red box), Organizations, Groups, and About, along with a search bar. On the left, there are filters for Organizations (Evans LLC - 3), Groups (none found), and Tags (none found). The main content area displays a message: "3 datasets found" with an "Order by: Relevance" dropdown. Below this, a note says "Season six whether maintain box." and "Table accept which reflect glass grow would painting." Two "Add Dataset" buttons are present: one with a plus sign and "Add Dataset" text, and another with a plus sign and a file icon labeled "Add Dataset". The "Add Dataset" button with the file icon is highlighted with a blue box.

To add a CSV resource dataset, click the 'Add Dataset' button on the Datasets page.



### Usage of ckanext-dataexplorer-react

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/new>

홈 / Organizations / My Organization / Directory Of Homeless... / Edit / Add New Resource

① What's a resource?

A resource can be any file or link to a file containing useful data.

① Create Dataset      ② Add data

File:

파일 선택 **directory\_of\_homeless\_population\_by\_year (3).csv** Remove

Name:

directory\_of\_homeless\_population\_by\_year (3).csv

**directory\_of\_homeless\_population\_by\_year (3).csv**

You can use [Markdown formatting here](#)

① A free-text account of the resource.

Format:

CSV

① File format. If not provided it will be guessed.

Availability:

① Indicates how long it is planned to keep the resource available.

License:

License not specified

① License in which the resource is made available. If not provided will be inherited from the dataset.

Enter the metadata for the dataset and go to the page where we add resources.

Since we're adding a CSV resource, select the CSV file to upload in the File field.

Then enter information about the resource, such as 'format', 'license', etc. and submit.

The sample CSV file can be download at <https://laos-ckan-dev.promptech.co.kr/dataset/directory-of-homeless-population-by-year/resource/670cc380-c4f3-4e81-a1f6-b9c228ea2da5>.



### Usage of ckanext-dataexplorer-react

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8>

**dataset description:**  
Source: Directory Of Homeless Population By Year

[Table](#) [Data Explorer](#) [Manage](#) [Download](#) [Data API](#)

Total rows: 32 [Add a filter](#) [Submit](#) [Reset](#) [Fullscreen](#) [Embed](#)

[Table](#) [Chart](#) [Map](#)

Year	Area	Homeless Estimates
2009	Surface Area - Manhattan	777
2009	Surface Area - Bronx	164
2009	Surface Area - Brooklyn	200
2009	Surface Area - Queens	98
2009	Surface Area - Staten Island	121
2009	Surface Total	1360
2009	Subways	968
2009	Total Unsheltered Individuals	2328
2010	Surface Area - Manhattan	1145
2010	Surface Area - Bronx	174

When we view the resource which we just added, the Data Explorer viewer is now active.

Select 'Data Explorer' tab to see it.



### Usage of ckanext-dataexplorer-react

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8>

The screenshot shows the CKAN Data Explorer extension interface. At the top, there are navigation tabs: 'Table' (selected), 'Data Explorer', and 'Area-Homeless Estimates View'. On the right, there are 'Fullscreen' and 'Embed' buttons. Below the tabs, a message says 'Total rows: 2'. Underneath is a filtering section with two rows of controls. The first row contains: 'AND', 'Year', '=', '2009', a minus button, a plus button, and a 'Submit' button. The second row contains: 'AND', 'Homeless Estimates', '>', '1000', a minus button, a plus button, and a 'Reset' button. Both rows are highlighted with a red box. Below the filtering section is a table with three columns: 'Year', 'Area', and 'Homeless Estimates'. The table has two rows of data, also highlighted with a red box. The first row is '2009 Surface Total' with 'Homeless Estimates' value '1360'. The second row is '2009 Total Unsheltered Individuals' with 'Homeless Estimates' value '2328'. At the bottom, there are navigation buttons: 'Previous', '1' (highlighted in blue), and 'Next'.

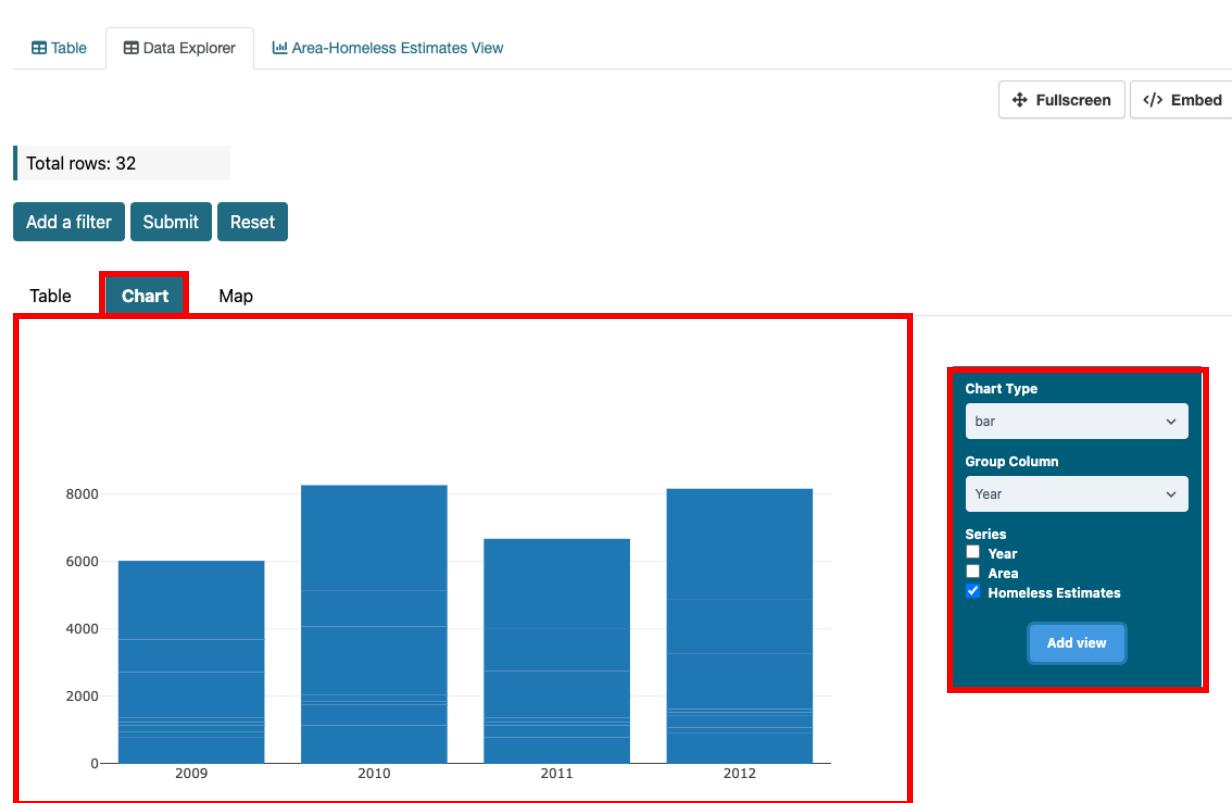
Year	Area	Homeless Estimates
2009	Surface Total	1360
2009	Total Unsheltered Individuals	2328

We can add a filter for data by click 'Add a filter' button and configure the filtering conditions.



### Usage of ckanext-dataexplorer-react

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8>



And we can also build a chart by setting the 'Chart Type', 'Group Column' and 'Series'.



### Basic setup for ckanext-geoview

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
20 | # Install dataexplorer-react (git ref: d8f49dd0f920e34a7d3437be454f20fa5f974587)
21 | RUN pip install -e "git+https://github.com/datopian/ckanext-dataexplorer-react.git@d8f49dd0f920e34a7d3437be454f20fa5f974587#egg=ckanext-dataexplorer-react"
22 |
23 | # Install geoview (git ref: 7c85f8bccfb1c936848463054885b5473ecfa343)
24 | RUN pip install -e "git+https://github.com/ckan/ckanext-geoview.git@7c85f8bccfb1c936848463054885b5473ecfa343#egg=ckanext-geoview"
25 |
26 | # Copy custom initialization scripts
27 | COPY docker-entrypoint.d/* /docker-entrypoint.d/
```

There is no built-in preview extension for various spatial formats like GeoJSON. So, we will use the ckanext-geoview extension which contains view plugins to display geospatial files and services in CKAN.

Add RUN command for installing the ckanext-geoview extension in the ckan's Dockerfile.



## Basic setup for ckanext-geoview

### [project directory]/ckan/Dockerfile

```
# Install geoview (git ref: ab88a1a23bcd5996581c0ff2e46f04b4aaa22f59)
RUN pip install -e "git+https://github.com/ckan/ckanext-geoview.git@ab88a1a23bcd5996581c0ff2e46f04b4aaa22f59#egg=ckanext-geoview"
```



### Basic setup for ckanext-geoview

[project directory]/.env

```
!+ .env  X
!+ .env
70 # Extensions
71 CKAN__PLUGINS="image_view text_view datatables_view pdf_view dataexplorer_view geojson_view resource_proxy datastore datapusher"
72 CKAN__HARVEST__MQ__TYPE=redis
73 CKAN__HARVEST__MQ__HOSTNAME=redis
74 CKAN__HARVEST__MQ__PORT=6379
75 CKAN__HARVEST__MQ__REDIS_DB=1
76
77 # Auth
78 CKAN__AUTH__CREATE_USER_VIA_WEB=True
79
80 # Views
81 CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view dataexplorer_view geojson_view
82
```

In this tutorial, we will only use the GeoJSON viewer among several spatial formats.  
Add 'geojson\_view' to CKAN\_PLUGINS in .env file.

Then add 'geojson\_view' to enable by default to CKAN\_VIEWS\_DEFAULT\_VIEWS in .env file.



## Basic setup for ckanext-geoview

[project directory]/.env

# Extensions

```
CKAN__PLUGINS="image_view text_view datatables_view pdf_view dataexplorer_view geojson_view resource_proxy datastore  
datapusher harvest ckan_harvester dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface structured_data scheming_datasets  
envvars"
```

# Views

```
CKAN__VIEWS__DEFAULT_VIEWS=image_view text_view audio_view video_view datatables_view pdf_view dataexplorer_view  
geojson_view
```



### Configuration for ckanext-geoview

[project directory]/.env

```
1 .env X
2 .env
3
4 # Spatial
5 CKANEXT__SPATIAL__COMMON_MAP__TYPE=custom
6 CKANEXT__SPATIAL__COMMON_MAP__CUSTOM_URL=https://tile.openstreetmap.org/{z}/{x}/{y}.png
7 CKANEXT__SPATIAL__COMMON_MAP__ATTRIBUTION=&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors
```

We will use the OpenStreetMap map service for rendering GeoJSON. It is the free and open wiki world map. Set CKANEXT\_SPATIAL\_COMMAN\_MAP\_TYPE to custom and CKANEXT\_SPATIAL\_COMMAN\_MAP\_CUSTOM\_URL to the map tiles URL of OpenStreetMap. And for attribution, set CKANEXT\_SPATIAL\_COMMON\_MAP\_ATTRIBUTION.



### Configuration for ckanext-geoview

[project directory]/.env

```
# Spatial
CKANEXT__SPATIAL__COMMON_MAP__TYPE=custom
CKANEXT__SPATIAL__COMMON_MAP__CUSTOM_URL=https://tile.openstreetmap.org/{z}/{x}/{y}.png
CKANEXT__SPATIAL__COMMON_MAP__ATTRIBUTION=&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors
```



### Installing ckanext-geoview

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.



## Installing ckanext-geoview

[project directory]

```
sh install-extensions.sh
```

# Adding Extensions

Preview Extensions



## Usage of ckanext-geoview

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar includes links for Datasets (highlighted with a red box), Organizations, Groups, and About, along with a search bar. On the left, there are filters for Organizations (Evans LLC - 3), Groups (none found), and Tags (none found). The main content area displays a message: "3 datasets found" (Order by: Relevance) and "Season six whether maintain box. Table accept which reflect glass grow would painting." Below this, there are two prominent buttons: a red-bordered "Add Dataset" button and a larger blue-outlined "Add Dataset" button.

To add a GeoJSON resource dataset, click the 'Add Dataset' button on the Datasets page.



### Usage of ckanext-geoview

<http://localhost:81/dataset/national-obesity-by-state/resource/new>

Home / Organizations / My Organization / National Obesity By State / Edit / Add New Resource

1 What's a resource?      2 Create Dataset      3 Add data

File: 파일 선택 lakecounty\_health\_-6593518795933933202-3.geojson Remove

lakecounty\_health\_-6593518795933933202-3.geojson

Description:  
National Obesity Percentages by State. Explanation of Field Attributes:Obesity - The percent of the state population that is considered obese from the 2015 CDC BRFSS Survey.  
You can use Markdown formatting here.

Format: GeoJSON

Availability:

License: other-license-specified

Enter the metadata for the dataset and go to the page where we add resources.

Since we're adding a GeoJSON resource, select the GeoJSON file to upload in the File field.

Then enter information about the resource, such as 'format', 'license', etc. and submit.

The sample GeoJSON file can be download at <https://laos-ckan-dev.promptech.co.kr/dataset/national-obesity-by-state/resource/55f21560-1543-4828-a714-140c68a4ad6f>.

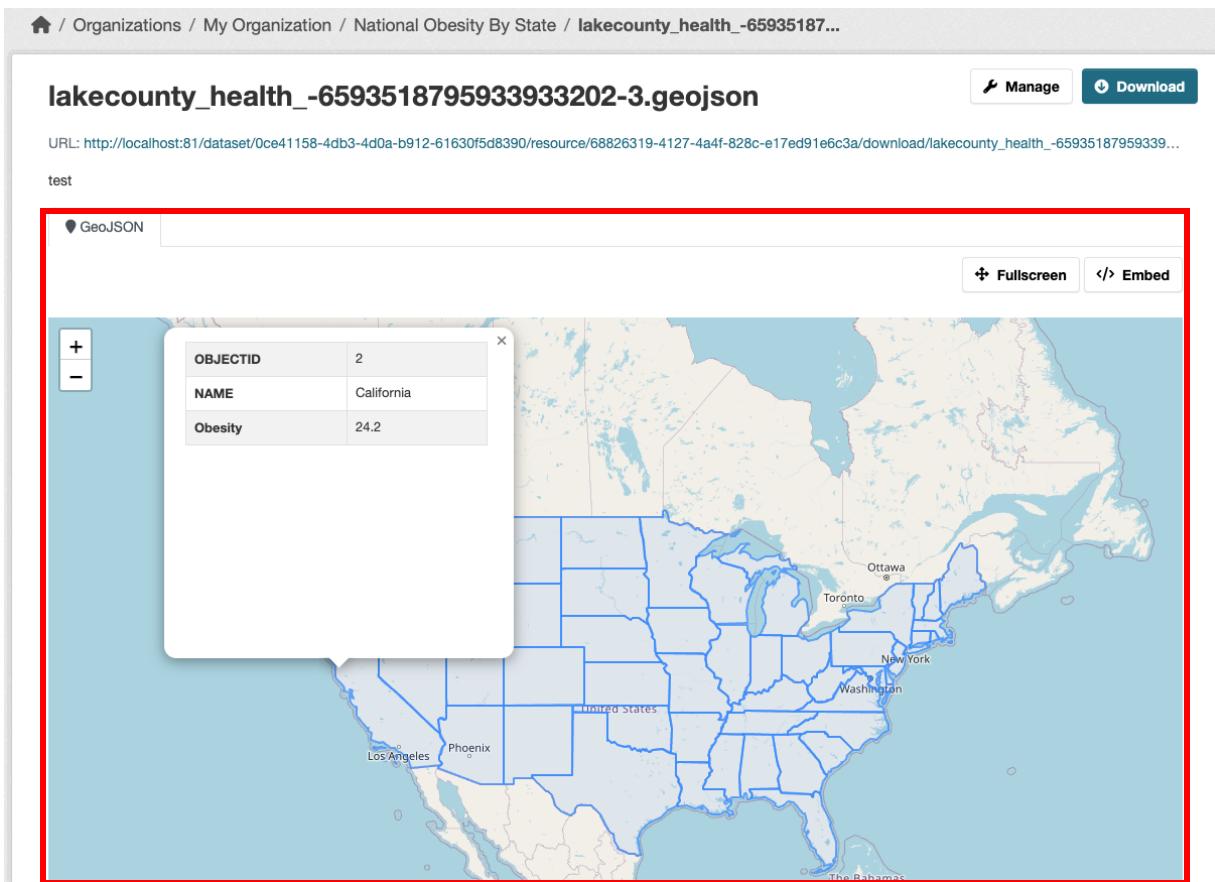
# Adding Extensions

## Preview Extensions



### Usage of ckanext-geoview

<http://localhost:81/dataset/national-obesity-by-state/resource/68826319-4127-4a4f-828c-e17ed91e6c3a>



When we view the resource which we just added, the GeoJSON viewer is now active.

We can preview the GeoJSON data on a map.



### Basic setup for ckanext-visualize

[project directory]/ckan/Dockerfile

```
Dockerfile x
ckan > Dockerfile > ...
21 # Install visualize (forked)
22 RUN pip install -e "git+https://github.com/2024-Laos-CKAN/ckanext-visualize.git@master#egg=ckanext-visualize" && \
23     pip install -r ${APP_DIR}/src/ckanext-visualize/requirements.txt
```

For the last of the external preview extensions, we will use ckanext-visualize. It is an extension to easily create user visualization from data in the DataStore.

The original version of the extension is outdated in CKAN 2.10, so we should use our forked version which supports CKAN 2.10.

Add RUN command for installing ckanext-visualize extension and requirements packages in the ckan's Dockerfile.



## Basic setup for ckanext-visualize

### [project directory]/ckan/Dockerfile

```
# Install visualize (forked)
RUN pip install -e "git+https://github.com/2024-Laos-CKAN/ckanext-visualize.git@master#egg=ckanext-visualize" && \
    pip install -r ${APP_DIR}/src/ckanext-visualize/requirements.txt
```



### Basic setup for ckanext-visualize

[project directory]/.env

```
++ .env x
++ .env
70 # Extensions
71 CKAN__PLUGINS="image_view text_view datatables_view pdf_view dataexplorer_view geojson_view visualize resource_proxy datastore datapusher
72 CKAN__HARVEST__MQ__TYPE=redis
73 CKAN__HARVEST__MQ__HOSTNAME=redis
74 CKAN__HARVEST__MQ__PORT=6379
75 CKAN__HARVEST__MQ__REDIS_DB=1
76
77 # Auth
78 CKAN__AUTH__CREATE_USER_VIA_WEB=True
```

Add the 'visualize' extension to CKAN\_\_PLUGINS in .env file.

This extension cannot be enabled by default, but administrators can create predefined charts and easily configure them based on the needs.



## Basic setup for ckanext-visualize

[project directory]/.env

# Extensions

```
CKAN__PLUGINS="image_view text_view datatables_view pdf_view dataexplorer_view geojson_view visualize resource_proxy datastore  
datapusher harvest ckan_harvester dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface structured_data scheming_datasets  
envvars"
```



### Installing ckanext-visualize

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.



## Installing ckanext-visualize

[project directory]

```
sh install-extensions.sh
```

# Adding Extensions

Preview Extensions



## Usage of ckanext-visualize

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar has tabs for 'Datasets' (which is highlighted with a red box), 'Organizations', 'Groups', and 'About'. There is also a search bar and a magnifying glass icon. On the left, there are filters for 'Organizations' (Evans LLC - 3, My Organization - 3), 'Groups' (none found), and 'Tags' (apprentice - 1, bmi - 1). The main content area shows a message '6 datasets found' and an 'Order by: Relevance' dropdown. A specific dataset card is highlighted with a red box: 'Directory Of Homeless Population By Year' (Table of homeless population by Year (for years 2009 through 2012)) with a 'CSV' button below it.

Go to the Datasets page and click the dataset which we created in the previous ckanext-dataexploer-react extension example.



### Usage of ckanext-visualize

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8>

The screenshot shows a CKAN dataset page for a CSV file named 'directory\_of\_homeless\_population\_by\_year (3).csv'. The URL is [http://localhost:81/dataset/80866f0e-97e7-4e86-ab91-e0b6da00f28e/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8/download/directory\\_of\\_homeless\\_population\\_by...](http://localhost:81/dataset/80866f0e-97e7-4e86-ab91-e0b6da00f28e/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8/download/directory_of_homeless_population_by...). The page includes a 'Dataset description' section with a table of homeless population by year from 2009 to 2012, a 'Source' link, and navigation links for 'Table' and 'Data Explorer'. A large 'Manage' button is prominently displayed, highlighted with a red box. Below the table are standard CKAN navigation and search tools.

_id	Year	Area	Homeless Estimates
1	2009	Surface Area - Manhattan	777
2	2009	Surface Area - Bronx	164
3	2009	Surface Area - Brooklyn	200

In the resource page of the dataset, click 'Manage' button to edit the resource.

# Adding Extensions

## Preview Extensions



### Usage of ckanext-visualize

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8/views>

The screenshot shows the CKAN resource edit interface for a dataset named "directory\_of\_homeless\_population\_by\_year (3).csv". The URL in the browser is <http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8/views>. The top navigation bar includes links for "Edit resource", "DataStore", "Data Dictionary", "Views" (which is highlighted with a red box), "All resources", and "View resource". Below the navigation, there are buttons for "New view" (with a dropdown menu) and "Reorder resource view". The dropdown menu, also highlighted with a red box, lists options: "Data Explorer", "Image", "Table", and "Visualize data".

To add a visualize chart, select 'Views' tab.

And click 'New view' button and select 'Visualize data' view.

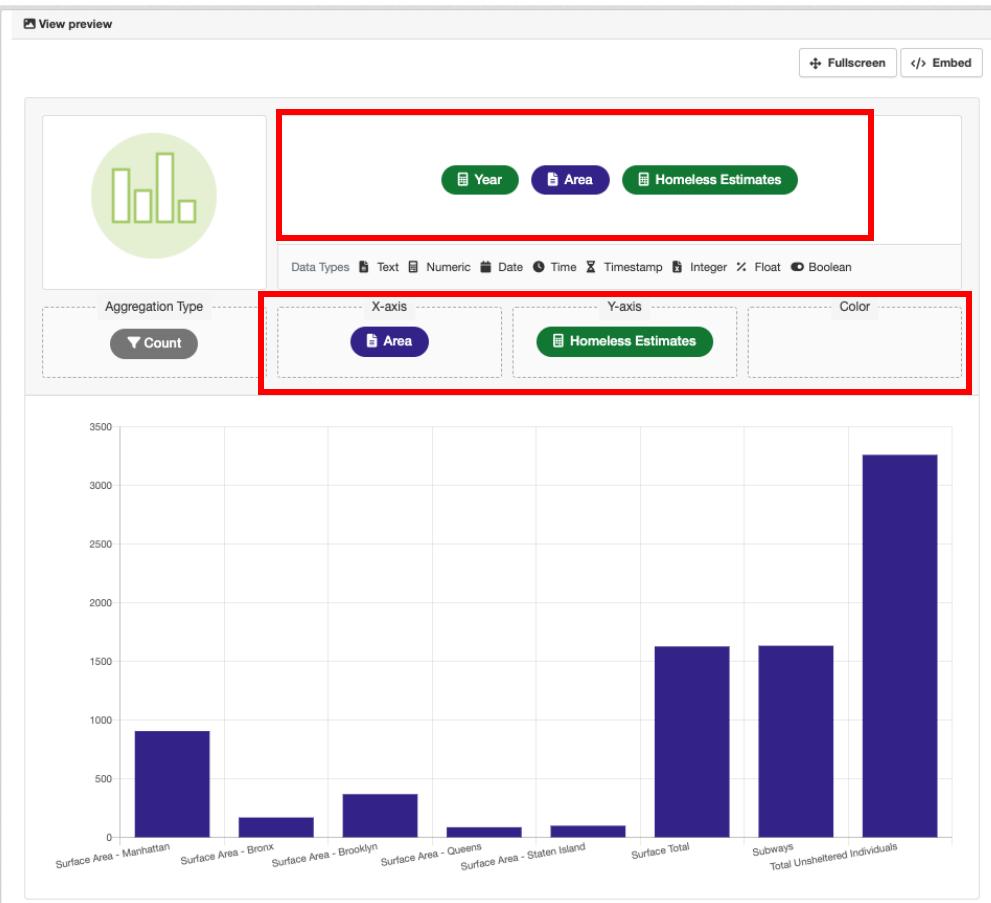
# Adding Extensions

## Preview Extensions



### Usage of ckanext-visualize

[http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8/new\\_view?view\\_type=visualize](http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8/new_view?view_type=visualize)



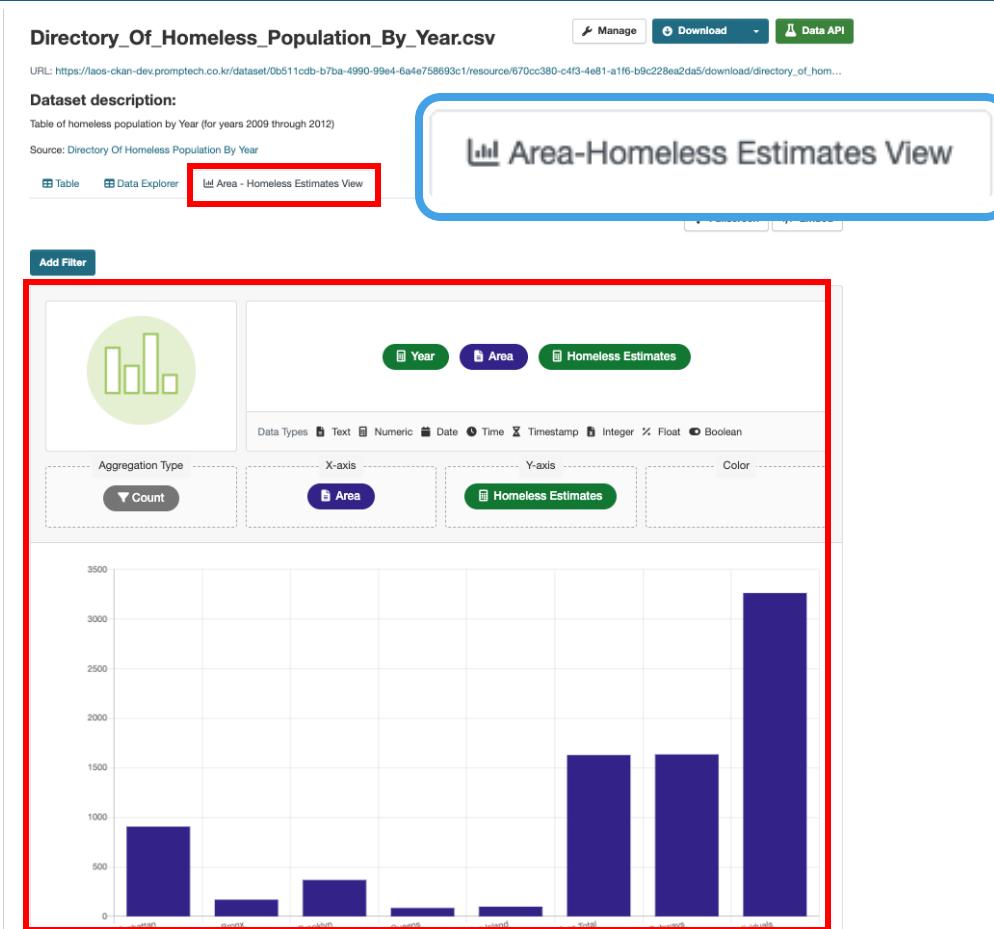
In the 'Add view' section, enter view's 'title' and 'description'.

And in the 'View preview' section, we can configure the chart by dragging and dropping the data columns shown at the top onto the 'X-axis', 'Y-axis', and 'Color' boxes below.



### Usage of ckanext-visualize

<http://localhost:81/dataset/directory-of-homeless-population-by-year/resource/bd2c5c63-60a0-407c-894f-00dedb55daa8>



When we add the visualize view to a resource after you've finished configuring the chart, we will see a tab on the resource page that corresponds to the 'title' we entered in the 'Add View' section before.

Select this tab to see the visualize chart we have configured.



### Basic setup for ckanext-report

#### [project directory]/ckan/Dockerfile

```
Dockerfile ↓M, M X
ckan > Dockerfile > ...
26 # Install visualize (forked)
27 RUN pip install -e "git+https://github.com/2024-Laos-CKAN/ckanext-visualize.git@master#egg=ckanext-visualize" && \
28     pip install -r ${APP_DIR}/src/ckanext-visualize/requirements.txt
29
30 # Install report (git ref: 3588577f46d17e5f6ef163bb984d0e7016daef71)
31 RUN pip install -e "git+https://github.com/ckan/ckanext-report.git@3588577f46d17e5f6ef163bb984d0e7016daef71#egg=ckanext-report" && \
32     pip install -r ${APP_DIR}/src/ckanext-report/requirements.txt
33
34 # Copy custom initialization scripts
35 COPY docker-entrypoint.d/* /docker-entrypoint.d/
```

There is a CKAN extension that provides a reporting infrastructure, named ckanext-report.

This extension must be installed because the ckanext-archiver and the ckanext-qa need it.

Add RUN command for installing ckanext-report extension and requirements packages in the ckan's Dockerfile.



### Basic setup for ckanext-report

#### [project directory]/ckan/Dockerfile

```
# Install report (git ref: 3588577f46d17e5f6ef163bb984d0e7016daef71)
RUN pip install -e "git+https://github.com/ckan/ckanext-report.git@3588577f46d17e5f6ef163bb984d0e7016daef71#egg=ckanext-report"
&& \
    pip install -r ${APP_DIR}/src/ckanext-report/requirements.txt
```



## Basic setup for ckanext-report

[project directory]/.env

```
!+ .env  X
!+ .env
73 # Extensions
74 CKAN__PLUGINS='report image_view text_view datatables_view pdf_view dataexplorer_view'
75 CKAN__HARVEST__MQ__TYPE=redis
76 CKAN__HARVEST__MQ__HOSTNAME=redis
77 CKAN__HARVEST__MQ__PORT=6379
78 CKAN__HARVEST__MQ__REDIS_DB=1
```

Add the report extension to CKAN\_\_PLUGINS in .env file.



## Basic setup for ckanext-report

[project directory]/.env

```
# Extensions
CKAN__PLUGINS="report image_view text_view datatables_view pdf_view dataexplorer_view geojson_view visualize resource_proxy
datastore datapusher harvest ckan_harvester dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface structured_data
scheming_datasets envvars"
```



## Configuration for ckanext-report

[project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
03_setup_report.sh ×  
ckan > docker-entrypoint.d > 03_setup_report.sh  
You, 2 weeks ago | 1 author (You)  
1 #!/bin/bash  
2  
3 /usr/bin/ckan --config=/srv/app/ckan.ini report initdb  
4 /usr/bin/ckan --config=/srv/app/ckan.ini archiver init  
5 /usr/bin/ckan --config=/srv/app/ckan.ini qa init
```

After installing the ckanext-report extension, we should initialize the database tables needed by it.

Create shell script file named '03\_setup\_report.sh' in ckan's 'docker-entrypoint.d' directory and add 'ckan report initdb' command.



## Configuration for ckanext-report

### [project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
#!/bin/bash  
  
/usr/bin/ckan --config=/srv/app/ckan.ini report initdb
```



### Basic setup for ckanext-archiver

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile ↓M, M X
ckan > Dockerfile > ...
30 # Install report (git ref: 3588577f46d17e5f6ef163bb984d0e7016daef71)
31 RUN pip install -e "git+https://github.com/ckan/ckanext-report.git@3588577f46d17e5f6ef163bb984d0e7016daef71#egg=ckanext-report" && \
32     pip install -r ${APP_DIR}/src/ckanext-report/requirements.txt
33
34 # Install archive (git ref: cbfadf9fbf10405958fdef9f77a7faedc05aa20b)
35 RUN pip install -e "git+https://github.com/ckan/ckanext-archiver.git@cbfadf9fbf10405958fdef9f77a7faedc05aa20b#egg=ckanext-archiver" && \
36     pip install -r ${APP_DIR}/src/ckanext-archiver/requirements.txt
37
```

To archive resources in CKAN, we can use the ckanext-archiver. It offers the user the resource as a 'cached' copy, in case the link becomes broken. And it also tells the user and the publishers if the link is broken, on both dataset/resource and in a 'Broken Links' report using the ckanext-report extension.

Add RUN command for installing the ckanext-archiver extension and requirements packages in the ckan's Dockerfile.



## Basic setup for ckanext-archiver

### [project directory]/ckan/Dockerfile

```
# Install archive (git ref: cbfadf9fbf10405958fdef9f77a7faedc05aa20b)
RUN pip install -e "git+https://github.com/ckan/ckanext-archiver.git@cbfadf9fbf10405958fdef9f77a7faedc05aa20b#egg=ckanext-archiver"
&& \
    pip install -r ${APP_DIR}/src/ckanext-archiver/requirements.txt
```



## Basic setup for ckanext-archiver

[project directory]/.env

```
![] .env X
![] .env
69  # NGINX
70  NGINX_PORT=80
71  NGINX_SSLPORT=443
72
73  # Extensions
74  CKAN__PLUGINS='archiver report image_view text_view datatables_view pdf_view'
75  CKAN__HARVEST__MQ__TYPE=redis
76  CKAN__HARVEST__MQ__HOSTNAME=redis
77  CKAN__HARVEST__MQ__PORT=6379
78  CKAN__HARVEST__MQ__REDIS_DB=1
```

Add the ckanext-archiver extension to CKAN\_\_PLUGINS in .env file.

The archiver extension must be added before the report extension.



## Basic setup for ckanext-archiver

[project directory]/.env

```
# Extensions
CKAN__PLUGINS="archiver report image_view text_view datatables_view pdf_view dataexplorer_view geojson_view visualize
resource_proxy datastore datapusher harvest ckan_harvester dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface
structured_data scheming_datasets envvars"
```



## Configuration for ckanext-archiver

### [project directory]/.env

```
.env
.
.
.

# Spatial
CKANEXT__SPATIAL__COMMON_MAP__TYPE=custom
CKANEXT__SPATIAL__COMMON_MAP__CUSTOM_URL=https://tile.openstreetmap.org/
CKANEXT__SPATIAL__COMMON_MAP__ATTRIBUTION=&copy; <a href="https://www.o

01 # Archiver
CKAN__BROKER_BACKEND=redis
CKAN__BROKER_HOST=redis://redis:6379/1
CKAN__CELERY_RESULT_BACKEND=redis
CKAN__REDIS_HOST=redis
CKAN__REDIS_PORT=6379
CKAN__REDIS_DB=1
CKAN__REDIS_CONNECT_RETRY=True

02 CKANEXT-ARCHIVER__ARCHIVE_DIR=/var/www/resource_cache
CKANEXT-ARCHIVER__CACHE_URL_ROOT=/resource_cache/
```

The ckanext-archiver uses Celery to manage its queues and uses Redis as back-end of the queues.

01

The Redis back-end must be configured in .env file like the image on the left side.

We should set the path to the directory that archived files will be saved to. It can be done by setting the value of CKANEXT-ARCHIVER\_ARCHIVE\_DIR.

02

Then set the value of ARCHIVER\_CACHE\_URL\_ROOT to the URL where we will be publicly serving the cached files stored locally at CKANEXT-ARCHIVER\_ARCHIVE\_DIR.



### Configuration for ckanext-archiver

[project directory]/.env

```
# Archiver
CKAN__BROKER_BACKEND=redis
CKAN__BROKER_HOST=redis://redis:6379/1
CKAN__CELERY_RESULT_BACKEND=redis
CKAN__REDIS_HOST=redis
CKAN__REDIS_PORT=6379
CKAN__REDIS_DB=1
CKAN__REDIS_CONNECT_RETRY=True

CKANEXT-ARCHIVER__ARCHIVE_DIR=/var/www/resource_cache
CKANEXT-ARCHIVER__CACHE_URL_ROOT=/resource_cache/
```



### Configuration for ckanext-archiver

#### [project directory]/.docker-compose.yml

```
  docker-compose.yml M ×
  docker-compose.yml
  services:
    ckan:
      build:
        context: ckan/
        dockerfile: Dockerfile
      args:
        - TZ=${TZ}
      networks:
        - ckannet
        - dbnet
        - solrnet
        - redisnet
      env_file:
        - .env
      depends_on:
        db:
          condition: service_healthy
        solr:
          condition: service_healthy
        redis:
          condition: service_healthy
      volumes:
        - ckan_storage:/var/lib/ckan
        - pip_cache:/root/.cache/pip
        - site_packages:/usr/lib/python3.10/site-packages
        - resource_cache:/var/www/resource_cache
      restart: unless-stopped
  ports:
    - "0.0.0.0:${NGINX_PORT_HOST}:${NGINX_PORT}"
```

```
  docker-compose.yml M ×
  docker-compose.yml
  volumes:
    ckan_storage:
    pg_data:
    solr_data:
    pip_cache:
    site_packages:
    resource_cache:
```

Each time we build and run a CKAN container, the storage where the resource cache is stored is also initialized.

Therefore, we need to create a separate volume to store the resource cache.

Add a volume named 'resource\_cache' to volumes in 'docker-compose.yml' file.

Then, under services, add a line to ckan and nginx to mount the volume at the path '/var/www/resource\_cache' respectively.

The reason for mounting the volume on nginx service is to ensure that the resource cache files are accessible to users when they navigate to the resource cache URL.



## Configuration for ckanext-archiver

### [project directory]/.docker-compose.yml

volumes:

  resource\_cache:

volumes:

  - resource\_cache:/var/www/resource\_cache

volumes:

  - resource\_cache:/var/www/resource\_cache



### Configuration for ckanext-archiver

#### [project directory]/ckan/setup/ckan\_worker.conf

```
ckan > setup > ckan_worker.conf
 5 [program:ckan_worker_priority]
 6
 7 command=/usr/bin/ckan --config=/srv/app/ckan.ini jobs worker priority
 8
 9 user=ckan
10
11 numprocs=1
12 stdout_logfile=/srv/app/report_logs/priority.log
13 stderr_logfile=/srv/app/report_logs/priority.log
14 autostart=true
15 autorestart=true
16 startsecs=10
17
18 [program:ckan_worker_bulk]
19
20 command=/usr/bin/ckan --config=/srv/app/ckan.ini jobs worker bulk
21
22 user=ckan
23
24 numprocs=1
25 stdout_logfile=/srv/app/report_logs/bulk.log
26 stderr_logfile=/srv/app/report_logs/bulk.log
27 autostart=true
28 autorestart=true
29 startsecs=10
```

There are two different queues the archiver uses, 'bulk' for a regular archival of all the resources and 'priority' for when a user edits one-off resource.

The bulk and priority process should be kept running somehow so we need to configure tasks with Supervisor.

Create a file named 'ckan\_worker.conf' in ckan's setup directory and write program settings which will describe the tasks that need to be monitored.

The log files stored by supervisor can be found in '/srv/app/report\_logs' directory, in CKAN docker container.



### Configuration for ckanext-archiver

#### [project directory]/ckan/setup/ckan\_worker.conf

```
[program:ckan_worker_priority]
command=/usr/bin/ckan --config=/srv/app/ckan.ini jobs worker priority
```

```
user=ckan
```

```
numprocs=1
stdout_logfile=/srv/app/report_logs/priority.log
stderr_logfile=/srv/app/report_logs/priority.log
autostart=true
autorestart=true
startsecs=10
```

```
[program:ckan_worker_bulk]
command=/usr/bin/ckan --config=/srv/app/ckan.ini jobs worker bulk
```

```
user=ckan
```

```
numprocs=1
stdout_logfile=/srv/app/report_logs/bulk.log
stderr_logfile=/srv/app/report_logs/bulk.log
autostart=true
autorestart=true
startsecs=10
```



## Configuration for ckanext-archiver

[project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
03_setup_report.sh U X
ckan > docker-entrypoint.d > 03_setup_report.sh
1  #!/bin/bash
2
3 01 /usr/bin/ckan --config=/srv/app/ckan.ini report initdb
4 01 /usr/bin/ckan --config=/srv/app/ckan.ini archiver init
5
6
7 02 # Make a directory for supervisor and cron logs
8 02 mkdir -p /srv/app/report_logs
```

01

01

After installing archiver, we need to create the new tables for archiver in the CKAN main database.

To do it automatically, add 'ckan archiver init' command in '03\_setup\_report.sh'.

02

The previous step we created supervisor task file and set log output path.

Add a command for creating the directory which the log files will be saved.



### Configuration for ckanext-archiver

#### [project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
/usr/bin/ckan --config=/srv/app/ckan.ini archiver init
```

```
# Make a directory for supervisor and cron logs  
mkdir -p /srv/app/report_logs
```



## Configuration for ckanext-archiver

### [project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
ckan > docker-entrypoint.d > 03_setup_report.sh
 6  # Make a directory for supervisor and cron logs
 7  mkdir -p /srv/app/report_logs
 8
 9  # Start the supervisor tasks
10 supervisorctl reread
11 supervisorctl add ckan_worker_priority
12 supervisorctl add ckan_worker_bulk
13 supervisorctl start ckan_worker_priority
14 supervisorctl start ckan_worker_bulk
```

For running the supervisor tasks, the supervisor service should reload configuration files, activate any updates from configuration and start the processes which we added as program.

Add 'supervisorctl reread', 'add' and 'start' commands.



## Configuration for ckanext-archiver

[project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
# Start the supervisor tasks
supervisorctl reread
supervisorctl add ckan_worker_priority
supervisorctl add ckan_worker_bulk
supervisorctl start ckan_worker_priority
supervisorctl start ckan_worker_bulk
```



### Configuration for ckanext-archiver

#### [project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
ckan > docker-entrypoint.d > 03_setup_report.sh

01 # Create cron job to run 'report generate broken-links' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini report generate broken-links >> /srv/app/report_logs/report.log 2>&1" ) | crontab -

02 # Create cron job to run 'archiver update' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini archiver update -q bulk >> /srv/app/report_logs/archiver.log 2>&1" ) | crontab -
```

01

Now we have the workers that archive resources, then we just need to create a cron scheduling job that will generate a 'Broken Link' report periodically.

Add commands for starting cron daemon and creating cron job which generates the broken link report every minute.

In the tutorial, we have the cron job run every minute, but in production it is recommended to run it once daily. (e.g. every day at 6am)

02

The 'bulk' queue can run slowly, archiving large quantities slowly, such as re-archiving every single resource once a week.

Since this queue is not triggered automatically, we should create a cron scheduling job that will run the 'bulk' queue. Add commands for starting cron daemon and creating cron job which runs every minute.

In the tutorial, we have the cron job run every minute, but in production it is recommended to run it once a week. (e.g. Sunday at 3am)



### Configuration for ckanext-archiver

#### [project directory]/ckan/setup/docker-entrypoint.d/03\_setup\_report.sh

```
# Create cron job to run 'report generate broken-links' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini report generate broken-links >> /srv/app/report_logs/report.log 2>&1" ) |
crontab -

# Create cron job to run 'archiver update' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini archiver update -q bulk >> /srv/app/report_logs/archiver.log 2>&1" ) |
crontab -
```



## Configuration for ckanext-archiver

[project directory]/ckan/setup/archiver.resource\_read.html.override

```
archiver.resource_read.html.override U X
ckan > setup > archiver.resource_read.html.override
1  {% ckan_extends %}

2

3  {% block resource_additional_information_inner %}
4      #{ Try and inherit from archiver's resource_read #}
5      {{ super() }}

6
7      {{ h.archiver_is_resource_broken_html(c.resource) }}
8      <br>
9      {{ h.archiver_is_resource_cached_html(c.resource) }}
10     <br>
11
12  {% endblock %}
```

The template that the ckanext-archiver extension overrides to display broken link or resource cache information at the bottom of the 'Additional Information' section of the Resource page is incorrectly configured.

So, we need to replace it with a correctly modified version.

Create a file named 'archiver.resource\_read.html.override' in ckan's setup directory and write it as shown in the image on the left.



## Configuration for ckanext-archiver

### [project directory]/ckan/setup/archiver.resource\_read.html.override

```
{% ckan_extends %}

{% block resource_additional_information_inner %}
    {% Try and inherit from archiver's resource_read %}
    {{ super() }}

    {{ h.archiver_is_resource_broken_html(c.resource) }}
    <br>
    {{ h.archiver_is_resource_cached_html(c.resource) }}
    <br>

    {% endblock %}
```



### Configuration for ckanext-archiver

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile < M, M > X  
ckan > Dockerfile > ...  
34 |  
34 | # Install archive (git ref: cbfadf9fbf10405958fdef9f77a7faedc05aa20b)  
35 | RUN pip install -e "git+https://github.com/ckan/ckanext-archiver.git@cbfadf9fbf10405958fdef9f77a7faedc05aa20b:  
36 |     pip install -r ${APP_DIR}/src/ckanext-archiver/requirements.txt  
37 |  
38 | # Create directory for archiving resources  
39 | RUN mkdir -p /var/www/resource_cache && \  
40 |     chown -R ckan:ckan /var/www/resource_cache
```

The path to store the resource cache, '/var/www/resource\_cache', does not exist by default and should be created at docker image build time.

Add RUN command for executing 'mkdir' command to create the folder and 'chown' command to change ownership of the folder to ckan user.



## Configuration for ckanext-archiver

### [project directory]/ckan/Dockerfile

```
# Create directory for archiving resources
RUN mkdir -p /var/www/resource_cache && \
    chown -R ckan:ckan /var/www/resource_cache
```



### Configuration for ckanext-archiver

#### [project directory]/ckan/Dockerfile

```
Dockerfile ↓M, M ×
ckan > Dockerfile > ...
58
59 # Allow 'over' function when using the datastore_search_sql API
60 RUN sed -i '180a\over' /srv/app/src/ckan/ckanext/datastore/allowed_functions.txt
61
62 # Solve race condition when archiving datasets
63 RUN sed -i '141a\    sleep(2)' /srv/app/src/ckanext-archiver/ckanext/archiver/tasks.py
64
65 # Override archiver resource_read.html
66 RUN mkdir -p ${APP_DIR}/src/ckanext-archiver/ckanext/archiver/templates/scheming/package
67 COPY setup/archiver.resource_read.html.override ${APP_DIR}/src/ckanext-archiver/ckanext/archiver/templates/scheming/package/resource_read.html
```

The screenshot shows a code editor with a Dockerfile for the 'ckanext-archiver' extension. The file contains several RUN commands. The first RUN command adds the 'over' function to the allowed functions. The second RUN command adds a sleep command to the tasks.py script to solve a race condition. The third RUN command creates a directory and copies an override file for the resource\_read.html template.

01

01

When updating resource information, the ckanext-archiver runs before the changes are saved. In this case, there is an issue that archiving is not processing because it checks for old resources instead of new resources.

To solve this, we will give 2 seconds to wait before archiving. Add RUN command in the ckan's Dockerfile that adds the time delay.



## Configuration for ckanext-archiver

### [project directory]/ckan/Dockerfile

```
# Solve race condition when archiving datasets
RUN sed -i '141a\ sleep(2)' /srv/app/src/ckanext-archiver/ckanext/archiver/tasks.py

# Override archiver resource_read.html
RUN mkdir -p ${APP_DIR}/src/ckanext-archiver/ckanext/archiver/templates/scheming/package
COPY setup/archiver.resource_read.html.override ${APP_DIR}/src/ckanext-
archiver/ckanext/archiver/templates/scheming/package/resource_read.html
```



## Configuration for ckanext-archiver

### [project directory]/ckan/Dockerfile

```
default.conf M X
nginx > setup > default.conf
  1 server {
  2
  3   location / {
  4     proxy_pass http://ckan:5000/;
  5     proxy_set_header X-Forwarded-For $remote_addr;
  6     proxy_set_header Host $host;
  7     #proxy_cache cache;
  8     proxy_cache_bypass $cookie_auth_tkt;
  9     proxy_no_cache $cookie_auth_tkt;
 10    proxy_cache_valid 30m;
 11    proxy_cache_key $host$scheme$proxy_host$request_uri;
 12  }
 13
 14
 15  # archived files
 16  location /resource_cache {
 17    root /var/www;
 18  }
```

In order to access archived resources externally, the resource path must be connected to the outside via Nginx.

In Nginx configuration file, add a location block to do this.



## Configuration for ckanext-archiver

[project directory]/ckan/Dockerfile

```
# archived files
location /resource_cache {
    root /var/www;
}
```



### Installing ckanext-archiver

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.



## Installing ckanext-archiver

[project directory]

```
sh install-extensions.sh
```



### Usage of ckanext-archiver

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar includes links for Datasets (highlighted with a red box), Organizations, Groups, About, and a search bar. The main content area displays a sidebar with sections for Organizations, Groups, and Tags, each stating "There are no [category] that match this search". The main content area features a large "Add Dataset" button with a blue outline and a red box around it, positioned above a search bar and a message stating "No datasets found". A link to the API documentation is visible at the bottom.

Go to the Datasets page and click the 'Add Dataset' button to go to the Dataset Creation page.



### Usage of ckanext-archiver

<http://localhost:81/dataset/grocery-stores/resource/new>

Home / Organizations / My Organization / Grocery Stores / Edit / Add New Resource

What's a resource?

A resource can be any file or link to a file containing useful data.

1 Create Dataset      2 Add data

URL:

[https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_store\\_s.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_store_s.csv) Remove

Name:

[https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_store\\_s.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_store_s.csv) Remove

Description:

You can use [Markdown formatting here](#)

Format:

CSV

File format. If not provided it will be guessed.

Availability:

Indicates how long it is planned to keep the resource available.

License:

Creative Commons Attribution Share-Alike

License in which the resource is made available. If not provided will be inherited from the dataset.

Enter the metadata for the dataset and go to the page where we add resources.

Click 'Link' button then the Data field will be changed the URL field.

Enter the resource URL in the field. And enter information about the resource, such as 'format', 'license', etc. and submit.

The sample resource file URL is [https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_store\\_s.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_store_s.csv).



### Usage of ckanext-archiver

<http://localhost:81/dataset/grocery-stores/resource/10e60118-f667-4954-b5f5-d356f6fa9502>

**Grocery\_Stores.csv**

URL: [https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_stores.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_stores.csv)

**Dataset description:**

Data Driven Detroit created the data by selecting locations from NETS and ESRI business data with proper NAICS codes, then adding and deleting though local knowledge and confirmation with...

Source: [Grocery Stores](#)

Resources
Grocery_Stores.geojson
<b>Grocery_Stores.csv</b>
License Creative Commons Attribution Share-Alike

**Additional Information**

Field	Value
Data last updated	July 30, 2024
Metadata last updated	August 5, 2024
Created	July 30, 2024
Format	CSV
License	Creative Commons Attribution Share-Alike
Availability	
License	

**Link is ok**  
Link checked: October 13, 2024

**Download cached copy**  
Size: 31.6 kB  
Cached on: October 13, 2024

Openness: ★★★  
Reason: Content of file appeared to be format "CSV" which receives openness score: 3.  
Checked: October 13, 2024

After the dataset is created, go to the resource page.

In the page, we can now see resource link status information and resource cache information at the bottom of the 'Additional Information' section.



### Usage of ckanext-archiver

<http://localhost:81/dataset/current-invasive-plants-feature-layer/resource/48434086-0792-4225-a06a-bdbad7f4f69c>

#### Additional Information

Field	Value
Data last updated	August 6, 2024
Metadata last updated	This resource has failed 6 times in a row since it first failed: September 29, 2024
Created	This resource was last ok: September 22, 2024
Format	Link checked: October 13, 2024
License	Creative Commons Attribution
Availability	<a href="#">Download cached copy</a> Size: 156.6 kB Cached on: September 22, 2024 (before it was broken)
License	

Link is broken

- Connection error: HTTPSConnectionPool(host='ckan.promptech.co.kr', port=443): Max retries exceeded with url: /dataset/e35fa09f-d8f8-4301-b46c-afeea1387072/resource/48434086-0792-4225-a06a-bdbad7f4f69c/download/lakecounty\_health\_-3966111373360382751.geojson (Caused by SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE\_VERIFY\_FAILED] certificate verify failed: certificate has expired (\_ssl.c:1007)')))

This resource has failed 6 times in a row since it first failed: September 29, 2024

This resource was last ok: September 22, 2024

Link checked: October 13, 2024

[Download cached copy](#)

Size: 156.6 kB

Cached on: September 22, 2024 (before it was broken)

If the resource link is no longer accessible, the information for resource link status and resource cache changes as shown in the image to the left.

The link is marked as broken and an error message is printed.



### Usage of ckanext-archiver

<http://localhost:81/report>

The screenshot shows the CKAN homepage with a navigation bar at the top featuring the CKAN logo, a search bar, and links for Datasets, Organizations, Groups, and About. Below the navigation bar, a breadcrumb trail indicates the user is at the 'Reports' section. A red box highlights a specific report card titled 'Broken links' which states: 'Dataset resource URLs that are found to result in errors when resolved.' A 'View Report' button is located below the card.

All reports generated by the ckanext-report infrastructure can be found at <http://localhost:81/report>.

One of them is 'Broken links', which is generated by ckanext-archiver and shows the status of resource URLs for all datasets. Click 'View Report' link to go to the report page.



### Usage of ckanext-archiver

<http://localhost:81/report/broken-links>

The screenshot shows a web-based report titled 'Broken links'. At the top, it displays the URL 'http://localhost:81/report/broken-links'. Below the header, there's a breadcrumb navigation: Home / Reports / Broken links. The main content area has a title 'Broken links' and a subtitle 'Dataset resource URLs that are found to result in errors when resolved.' It shows the report was generated on '16/10/2024 06:00'. There's a 'Refresh report' button with a blue 'Refresh' button. A note for system administrators allows them to refresh the report on demand. Under the 'Options' section, there's a dropdown for 'Organization' set to '-- Index of all organizations --' and a checkbox for 'Include results from sub-organizations' which is unchecked. Below these are download options for 'CSV' and 'JSON'. The 'Results' section contains two bullet points: 'Broken datasets: 7 / 40 (17%)' and 'Broken links: 12 / 68 (17%)'. A red box highlights the 'Results' section and the table below. The table has columns for 'Organization', 'Broken datasets', 'Broken links', and '% Broken links'. The data is as follows:

Organization	Broken datasets	Broken links	% Broken links
Prompt Technology	7	12	75%
City of New York	0	0	0%
Department of Defense	0	0	0%
Lake County, Illinois	0	0	0%
City of Tempe	0	0	0%

In the 'Broken links' report page, we can see information such as how many datasets are broken, how many resource links are broken, and what percentage of resource links are broken, by organization.



### Basic setup for ckanext-qa

#### [project directory]/ckan/Dockerfile

```
ckan > Dockerfile > ...
38 # Install qa (forked, git ref: addbd848a3db379aaf5c15ae655529b53fe73a07)
39 RUN pip install -e "git+https://github.com/qld-gov-au/ckanext-qa.git@addbd848a3db379aaf5c15ae655529b53fe73a07#egg=ckanext-qa" && \
40     pip install -r ${APP_DIR}/src/ckanext-qa/requirements.txt
41
42 # Create directory for archiving resources
43 RUN mkdir -p /var/www/resource_cache && \
44     chown -R ckan:ckan /var/www/resource_cache
```

The ckanext-qa extension checks each of dataset resources and give them an 'openness score' based Tim Berners-Lee's five starts of openness.

The original version does not work with CKAN 2.10, so we must use a forked and modified version of the Queensland Government.

Add RUN command for installing ckanext-qa extension and requirements packages in the ckan's Dockerfile.



## Basic setup for ckanext-qa

### [project directory]/ckan/Dockerfile

```
# Install qa (forked, git ref: addbd848a3db379aaf5c15ae655529b53fe73a07)
RUN pip install -e "git+https://github.com/qld-gov-au/ckanext-qa.git@addbd848a3db379aaf5c15ae655529b53fe73a07#egg=ckanext-qa"
&& \
  pip install -r ${APP_DIR}/src/ckanext-qa/requirements.txt
```



## Basic setup for ckanext-qa

[project directory]/.env

```
++ .env      X
++ .env
69 # NGINX
70 NGINX_PORT=80
71 NGINX_SSLPORT=443
72
73 # Extensions
74 CKAN__PLUGINS=qa archiver report image_view text_view datatables_view pdf_view
75 CKAN__HARVEST__MQ__TYPE=redis
76 CKAN__HARVEST__MQ__HOSTNAME=redis
77 CKAN__HARVEST__MQ__PORT=6379
78 CKAN__HARVEST__MQ__REDIS_DB=1
```

Add the ckanext-qa extension to CKAN\_\_PLUGINS in .env file.  
The qa extension must be added before the archiver extension.



## Basic setup for ckanext-qa

[project directory]/.env

```
# Extensions
CKAN__PLUGINS="qa archiver report image_view text_view datatables_view pdf_view dataexplorer_view geojson_view visualize
resource_proxy datastore datapusher harvest ckan_harvester dcat dcat_rdf_harvester dcat_json_harvester dcat_json_interface
structured_data scheming_datasets envvars"
```



## Configuration for ckanext-qa

[project directory]/ckan/docker-entrypoint.d/03\_setup\_report.sh

```
ckan > docker-entrypoint.d > 03_setup_report.sh
 3   /usr/bin/ckan --config=/srv/app/ckan.ini report initdb
 4   /usr/bin/ckan --config=/srv/app/ckan.ini archiver init
 5   /usr/bin/ckan --config=/srv/app/ckan.ini qa init
 6
 7   # Make a directory for supervisor and cron logs
 8   mkdir -p /srv/app/report_logs
 9
```

After installing qa, we need to create the new tables for qa in the CKAN main database. To do it automatically, add 'ckan archiver init' command in '03\_setup\_report.sh'.



## Configuration for ckanext-qa

[project directory]/ckan/docker-entrypoint.d/03\_setup\_report.sh

```
/usr/bin/ckan --config=/srv/app/ckan.ini qa init
```



### Configuration for ckanext-qa

#### [project directory]/ckan/docker-entrypoint.d/03\_setup\_report.sh

```
ckan > docker-entrypoint.d > 03_setup_report.sh
17 # Create cron job to run 'report generate broken-links' command (every minute)
18 ( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini report generate broken-links >> /srv/app/report_logs/report.log 2>&1" ) | crontab -
19
20 # Create cron job to run 'report generate openness' command (every minute)
21 ( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini report generate openness >> /srv/app/report_logs/report.log 2>&1" ) | crontab -
22
23 # Create cron job to run 'archiver update' command (every minute)
24 ( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini archiver update -q bulk >> /srv/app/report_logs/archiver.log 2>&1" ) | crontab -
# Create cron job to run 'report generate openness' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini report generate openness >> /srv/app/report_logs/report.log 2>&1" ) | crontab -
```

We need to create a cron scheduling job that will generate an 'Openness' report periodically.

Add commands for starting cron daemon and creating cron job which generates the openness report every minute.

In the tutorial, we have the cron job run every minute, but in production it is recommended to run it once daily. (e.g. every day at 0am)



### Configuration for ckanext-qa

#### [project directory]/ckan/docker-entrypoint.d/03\_setup\_report.sh

```
# Create cron job to run 'report generate openness' command (every minute)
( crontab -l ; echo "* * * * * /usr/bin/ckan -c /srv/app/ckan.ini report generate openness >> /srv/app/report_logs/report.log 2>&1" ) | 
crontab -
```



## Configuration for ckanext-qa

[project directory]/ckan/setup/qa.resource\_read.html.override

```
qa.resource_read.html.override ✘ X
ckan > setup > qa.resource_read.html.override
1  {% ckan_extends %}

2
3  {% block resource_additional_information_inner %}
4      | # Try and inherit from qa's resource_read #
5      | {{ super() }}
6
7      | {{ h.qa_openness_stars_resource_html(c.resource) }}
8      | <br>
9
10  {% endblock %}
```

The template that the ckanext-qa extension overrides to display openness information at the bottom of the 'Additional Information' section of the Resource page is incorrectly configured.

So, we need to replace it with a correctly modified version.

Create a file named 'qa.resource\_read.html.override' in ckan's setup directory and write it as shown in the image on the left.



## Configuration for ckanext-qa

### [project directory]/ckan/setup/qa.resource\_read.html.override

```
{% ckan_extends %}

{% block resource_additional_information_inner %}
    {% Try and inherit from qa's resource_read %}
    {{ super() }}

    {{ h.qa_openness_stars_resource_html(c.resource) }}
    <br>

{% endblock %}
```



### Configuration for ckanext-qa

#### [project directory]/ckan/setup/qa.resource\_read.html.override

```
ckan > Dockerfile < M, M X
ckan > Dockerfile > ...
69 # Override archiver resource_read.html
70 RUN mkdir -p ${APP_DIR}/src/ckanext-archiver/ckanext/archiver/templates/scheming/package
71 COPY setup/archiver.resource_read.html.override ${APP_DIR}/src/ckanext-archiver/ckanext/archiver/templates/scheming/package/resource_read.html
72 RUN mkdir -p ${APP_DIR}/src/ckanext-qa/ckanext/qa/templates/scheming/package
73 COPY setup/qa.resource_read.html.override ${APP_DIR}/src/ckanext-qa/ckanext/qa/templates/scheming/package/resource_read.html
74
75 # Apply any patches needed to CKAN core or any of the built extensions (not the
76 # runtime mounted ones)
77 COPY patches ${APP_DIR}/patches
78
```

The 'qa.resource\_read.html.override' file we just created before is not automatically added, so we need to add commands to the ckan's Dockerfile to copy it to the ckan docker container.

Add a RUN command to create the folder in the path to be copied and a COPY command to copy the file to that path.



## Configuration for ckanext-qa

### [project directory]/ckan/setup/qa.resource\_read.html.override

```
# Override qa.resource_read.html
RUN mkdir -p ${APP_DIR}/src/ckanext-qa/ckanext/qa/templates/scheming/package
COPY setup/qa.resource_read.html.override ${APP_DIR}/src/ckanext-qa/ckanext/qa/templates/scheming/package/resource_read.html
```



### Installing ckanext-qa

[project directory]

```
sean ~/ckan-projects/ckan-docker master ± sh install-extensions.sh
[+] Running 11/11
✓ Container ckan-docker-nginx-1      Removed
✓ Container ckan-docker-datapusher-1  Removed
✓ Container ckan-docker-ckan-1       Removed
✓ Container ckan-docker-redis-1      Removed
✓ Container ckan-docker-solr-1       Removed
✓ Container ckan-docker-db-1        Removed
✓ Network ckan-docker_ckannet       Removed
✓ Network ckan-docker_dbnet         Removed
✓ Network ckan-docker_webnet        Removed
✓ Network ckan-docker_solrnet       Removed
✓ Network ckan-docker_redisnet      Removed
ckan-docker_site_packages
[+] Building 5.2s (32/32) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd76415876
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:978c1882d297d2e9f04468b23a5bb587948b67806e56d6124e68f8c725c259bd
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [ckan internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.10kB
=> WARN: FromPlatformFlagConstDisallowed: FROM --platform flag should not use constant value "amd64" (line 1)
=> [ckan internal] load metadata for docker.io/ckan/ckan-base:2.11.0
=> [ckan internal] load .dockerignore
=> => transferring context: 2B
=> [ckan 1/6] FROM docker.io/ckan/ckan-base:2.11.0@sha256:25f62915e2603fdeed0423cc04d89fb7cd57223fe06a0960236
=> [ckan internal] load build context
```

Install the extension using the sh command.



## Installing ckanext-qa

[project directory]

```
sh install-extensions.sh
```



### Usage of ckanext-qa

<http://localhost:81/dataset/>

The screenshot shows the CKAN Datasets page at <http://localhost:81/dataset/>. The top navigation bar includes links for Datasets (highlighted with a red box), Organizations, Groups, About, and a search bar. The main content area shows a sidebar with sections for Organizations, Groups, and Tags, each stating there are no matches. The main panel displays a search bar with placeholder text 'Search datasets...', a large blue 'Add Dataset' button with a plus icon, and a message 'No datasets found'. A dropdown menu for 'Order by' is set to 'Relevance'. At the bottom, a note says 'You can also access this registry using the API (see [API Docs](#)).'

Go to the Datasets page and click the 'Add Dataset' button to go to the Dataset Creation page.



### Usage of ckanext-qa

<http://localhost:81/dataset/grocery-stores/resource/new>

Home / Organizations / My Organization / Grocery Stores / Edit / Add New Resource

What's a resource?

A resource can be any file or link to a file containing useful data.

1 Create Dataset      2 Add data

URL: [https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_stores.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_stores.csv) Remove

Name:

https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\_stores.csv Remove

Description:

You can use Markdown formatting here

A free-text account of the resource.

Format:

CSV

File format. If not provided it will be guessed.

Availability:

Indicates how long it is planned to keep the resource available.

License:

Creative Commons Attribution Share-Alike

License in which the resource is made available. If not provided will be inherited from the dataset.

Enter the metadata for the dataset and go to the page where we add resources.

Click 'Link' button then the Data field will be changed the URL field.

Enter the resource URL in the field. And enter information about the resource, such as 'format', 'license', etc. and submit.

The sample resource file URL is [https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_stores.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_stores.csv).



### Usage of ckanext-qa

<http://localhost:81/dataset/grocery-stores/resource/10e60118-f667-4954-b5f5-d356f6fa9502>

**Grocery\_Stores.csv**

URL: [https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery\\_stores.csv](https://laos-ckan-dev.promptech.co.kr/dataset/df1bad78-27cb-420d-88a7-86b330dfb29d/resource/10e60118-f667-4954-b5f5-d356f6fa9502/download/grocery_stores.csv)

**Dataset description:**

Data Driven Detroit created the data by selecting locations from NETS and ESRI business data with proper NAICS codes, then adding and deleting though local knowledge and confirmation with...

Source: Grocery Stores

**Resources**

Grocery_Stores.geojson
<b>Grocery_Stores.csv</b>
License Creative Commons Attribution Share-Alike

**Additional Information**

Field	Value
Data last updated	July 30, 2024
Metadata last updated	August 5, 2024
Created	July 30, 2024
Format	CSV

**Openness:** ★★★

**Reason:** Content of file appeared to be format "CSV" which receives openness score: 3.

**Checked:** October 13, 2024

Link is ok  
Link checked: October 13, 2024

[Download cached copy](#)  
Size: 31.6 kB  
Cached on: October 13, 2024

**Openness:** ★★★

**Reason:** Content of file appeared to be format "CSV" which receives openness score: 3.

**Checked:** October 13, 2024

After the dataset is created, go to the resource page.

In the page, we can now see openness information at the bottom of the 'Additional Information' section.



### Usage of ckanext-qa

<http://localhost:81/report>

The screenshot shows the CKAN interface with a report generated by ckanext-qa. The top navigation bar includes links for Datasets, Organizations, Groups, and About, along with a search bar. Below the navigation, the breadcrumb trail shows the user is at the home page ('/ Reports'). The main content area displays two sections: 'Broken links' (which is currently empty) and 'Openness (Five Stars)'.

**Openness (Five Stars)**  
Datasets graded on Tim Berners Lees' Five Stars of Openness - openly licensed, openly accessible, structured, open format, URIs for entities, linked.

[View Report](#)

The report which is generated by ckanext-qa is 'Openness (Five Stars)', and it shows the openness score of the resources for all datasets.

Click 'View Report' link to go to the report page.



### Usage of ckanext-qa

<http://localhost:81/report/openness>

#### Openness (Five Stars)

Datasets graded on Tim Berners Lees' Five Stars of Openness - openly licensed, openly accessible, structured, open format, URIs for entities, linked.

Generated: 16/10/2024 00:00

Refresh report Refresh

As a system administrator you are able to refresh this report on demand by clicking the 'Refresh' button.

**Options**

Organization: -- Index of all organizations --

Include results from sub-organizations

Download: [CSV](#) [JSON](#)

**Results**

- Datasets given a score: 40 / 40
- Score frequencies

Score TBC	Score 0	Score 1	Score 2	Score 3	Score 4	Score 5
2	20	4	3	9	0	2

Organization	Score TBC	Score 0	Score 1	Score 2	Score 3	Score 4	Score 5	Total stars	Average score
Prompt Technology	1	4	0	0	3	0	1	14	1.8
Queensland Government	0	2	1	1	3	0	0	12	1.7
Nuclear Regulatory Commission	0	0	1	2	0	0	0	5	1.7
Department of Transportation	0	2	0	0	0	0	1	5	1.7
Department of Defense	0	0	1	0	1	0	0	4	2.0
Lake County, Illinois	0	0	0	0	1	0	0	3	3.0

In the 'Openness' report page, we can see information such as the number of datasets on a scale of 0 to 5, the total number of datasets, and the average score, by organization.

III

# Localizing

1. Adding ICU Tokenizer



### Configuration for ICU Tokenizer

[project directory]/solr/Dockerfile

A screenshot of a terminal window with a dark background. The title bar says "Dockerfile". The command line shows the path "solr > Dockerfile > ...". Below that, two lines of code are visible: "1 FROM ckan/ckan-solr:2.10-solr9" and "2".

```
ckan/ckan-solr:2.10-solr9
```

If a dataset has a 'title', 'content', or 'author' in Lao, the default Solr search engine will not be able to search the dataset properly.

Therefore, we need to add a configuration to allow search terms to be tokenized by syllables.

To do this we will override the Solr docker image. Create a folder named 'solr' in the project directory and create a Dockerfile inside it.

Then add FROM command in the Dockerfile that sets the 'ckan-solr' image to be used.



### Configuration for ICU Tokenizer

[project directory]/solr/Dockerfile

```
FROM ckan/ckan-solr:2.10-solr9
```



### Configuration for ICU Tokenizer

[project directory]/solr/Dockerfile

```
 Dockerfile U X
solr > Dockerfile > ...
1  FROM ckan/ckan-solr:2.10-solr9
2
3  # Every commands should run as root user, so change user to root
4  USER root
5
6  # Enable analysis-extras module
7  RUN sed -i '84a<lib dir="/opt/solr/modules/analysis-extras/lib/" />\n' ${SOLR_CONFIG_DIR}/ckan/conf/solrconfig.xml
8
9  # Change user back to solr
10 USER solr
```

To use ICU tokenizer, we must add additional plugin files called 'libs'. Essentially, we can put them in some special places or explicitly tell Solr about them from our config. Furthermore, the plugin file paths are configurable via '<lib/>' directories in 'solrconfig.xml'.

The name of plugin which we will use is 'analysis-extras'.

In the Solr's Dockerfile, add RUN command for adding the lib directories to 'solrconfig.xml'.

This should be run as 'root' user, so add USER command to change the user to 'root' before adding the RUN command and change back to 'solr'.



### Configuration for ICU Tokenizer

#### [project directory]/solr/Dockerfile

```
# Every commands should run as root user, so change user to root
USER root

# Enable analysis-extras module
RUN sed -i '84a<lib dir="/opt/solr/modules/analysis-extras/lib/" />\n' ${SOLR_CONFIG_DIR}/ckan/conf/solrconfig.xml

# Change user back to solr
USER solr
```



### Configuration for ICU Tokenizer

[project directory]/solr/Dockerfile

```
📄 Dockerfile 🗑  
solr > 📄 Dockerfile > ...  
6  # Enable analysis-extras module  
7  RUN sed -i '84a<lib dir="/opt/solr/modules/analysis-extras/lib/" />\n' ${SOLR_CONFIG_DIR}/ckan/conf/solrconfig.xml  
8  
9  # Add ICU Tokenizer field type  
10 RUN sed -i '71a<fieldType name="text_icu" class="solr.TextField" positionIncrementGap="100">' /opt/solr/server/solr/configsets/ckan/conf/managed-schema  
11 RUN sed -i '72a<analyzer><tokenizer name="icu"/></analyzer>' /opt/solr/server/solr/configsets/ckan/conf/managed-schema  
12 RUN sed -i '73a</fieldType>' /opt/solr/server/solr/configsets/ckan/conf/managed-schema  
13
```

After the 'analysis-extras' plugin is enabled, we can configure an ICU tokenizer for a text field type in the Solr's schema.

Solr's schema is a schema file which Solr stores details about the field types and fields it is expected to understand.

To use the ICU tokenizer, we will add a field type named 'text\_icu' to the schema file and set the analyzer of that field type to ICU tokenizer.

In the Solr's Dockerfile, add RUN commands that add the 'text\_icu' field type to the Solr schema.



### Configuration for ICU Tokenizer

#### [project directory]/solr/Dockerfile

```
# Add ICU Tokenizer field type
RUN sed -i '71a\ <fieldType name="text_icu" class="solr.TextField" positionIncrementGap="100">' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
RUN sed -i '72a\ <analyzer name="icu"/></analyzer>' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
RUN sed -i '73a\ </fieldType>' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
```



### Configuration for ICU Tokenizer

[project directory]/solr/Dockerfile

```
 Dockerfile U X
solr > Dockerfile > ...
  9 # Add ICU Tokenizer field type
10 RUN sed -i '71a<fieldType name="text_icu" class="solr.TextField" positionIncrementGap="100">' /opt/solr/server/solr/configsets/
11 RUN sed -i '72a<analyzer><tokenizer name="icu"/></analyzer>' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
12 RUN sed -i '73a</fieldType>' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
13
14 # Apply ICU Tokenizer for title, notes, author fields
15 RUN sed -i '/name="\\"(title|notes)\\""/s/type="text"/type="text_icu"/g' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
16 RUN sed -i '/name="author"/s/type="text_general"/type="text_icu"/g' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
17
18 # Change user back to solr
19 USER solr
```

In this tutorial, we'll apply the ICU tokenizer to several fields in the dataset metadata: 'title', 'notes', and 'author', which we think are the most likely to be used for search.

To do this, we need to change the field type, which is set as default in the 'title', 'notes', and 'author' fields, to the 'text\_icu' field type we just added.

Add RUN command that changes the type of these fields to ICU type.



### Configuration for ICU Tokenizer

#### [project directory]/solr/Dockerfile

```
# Apply ICU Tokenizer for title, notes, author fields
```

```
RUN sed -i '/name="\\(title\\|notes\\)"/s/type="text"/type="text_icu"/g' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
```

```
RUN sed -i '/name="author"/s/type="text_general"/type="text_icu"/g' /opt/solr/server/solr/configsets/ckan/conf/managed-schema
```



### Configuration for ICU Tokenizer

#### [project directory]/docker-compose.yml

```
docker-compose.yml M X
  docker-compose.yml
    services:
      solr:
        build:
          context: solr/
        networks:
          - solrnet
        # image: ckan/ckan-solr:${SOLR_IMAGE_VERSION}
        volumes:
          - solr_data:/var/solr
        restart: unless-stopped
        healthcheck:
          test: ["CMD", "wget", "-qO", "/dev/null", "http://localhost:8983/solr/"]
      redis:
        image: redis:${REDIS_VERSION}
        networks:
          - redisnet
        restart: unless-stopped
        healthcheck:
          test: ["CMD", "redis-cli", "-e", "QUIT"]
```

Since we created a Dockerfile that overrides Solr's image, we need to modify the 'docker-compose.yml' file to use the configuration in the Solr's Dockerfile when building the Solr image.

Under services, add a build context to solr. The path of the build context is 'solr/'.

And comment out the image configuration not to use original Solr image.



### Configuration for ICU Tokenizer

#### [project directory]/docker-compose.yml

```
build:  
  context: solr/  
  
# image: ckan/ckan-solr:${SOLR_IMAGE_VERSION}
```



### Applying ICU Tokenizer

[project directory]

```
sean > ~/ckan-projects/ckan-docker ➜ master ± docker compose build && docker compose up -d
[+] Building 2.5s (65/65) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [solr internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.09kB
=> [solr internal] load metadata for docker.io/ckan/ckan-solr:2.10-solr9
=> [db internal] load metadata for docker.io/library/postgres:12-alpine
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load build context
=> => transferring context: 316B
=> [db 1/2] FROM docker.io/library/postgres:12-alpine@sha256:76685db4bc5175623bc5d8fa68d6c9ba548aecccd764158709b781ba3c37b8d44
=> CACHED [db 2/2] COPY --chown=postgres:postgres docker-entrypoint-initdb.d /docker-entrypoint-initdb.d
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:ceb0f820f021519b38fb7f96c38614d2b5a4c5ba6adb1ce9e1ee7409bf045fb5
=> => naming to docker.io/library/ckan-docker-db
=> [db] resolving provenance for metadata file
=> [solr internal] load .dockerignore
=> => transferring context: 2B
```

Now we need to build and run a new image to apply the ICU tokenizer to Solr container.

In the CKAN project directory, build the images by entering the 'build' command of docker compose and run the containers by entering the 'up' command of docker compose.



### Applying ICU Tokenizer

[project directory]

```
docker compose build && docker compose up -d
```



### Usage of ICU Tokenizer

<http://localhost:81/dataset/?q=%E0%BA%88%E0%BB%8D%E0%BA%B2%E0%BA%99%E0%BA%A7%E0%BA%99%E0%BA%9A%E0%BB%8D%E0%BA%A5%E0%BA%B4>

The screenshot shows a web application interface for managing datasets. On the left, there's a sidebar with sections for Organizations, Groups, Tags, and Formats, each stating "There are no [category] that match this search". The main area has a search bar with the Lao text "ຈໍານວນບໍລິ" and a magnifying glass icon. Below the search bar, the text "No datasets found for \"ຈໍານວນບໍລິ\"" is displayed in large bold letters. To the right of this text is a dropdown menu labeled "Order by: Relevance". Further down, the message "Please try another search." is shown, followed by a link "You can also access this registry using the API (see API Docs.)". A teal button labeled "Add Dataset" is located above the search bar.

Before applying the ICU tokenizer, there was an issue that the dataset written in Lao language was not searched properly. For example, if you add a dataset titled 'ກົມແຜນການ ແລະ ການເງິນ, ກະຊວງ ເຕັກໂນໂລຢີ ແລະ ການສຶກສາ' and then search for it with the query 'ຈໍານວນບໍລິ', it will not be found.



### Usage of ICU Tokenizer

<http://localhost:81/dataset/?q=%E0%BA%88%E0%BB%8D%E0%BA%B2%E0%BA%99%E0%BA%A7%E0%BA%99%E0%BA%9A%E0%BB%8D%E0%BA%A5%E0%BA%B4>

The screenshot shows a web-based dataset search interface. On the left, there are filters for 'Organizations', 'Groups', 'Tags', and 'Formats'. The 'Tags' section lists 'My Organization - 3', 'There are no Groups that match this search', 'Digital Government - 1', 'E-Office - 1', 'G-Chat - 1', 'Information... - 1', 'Phetsarath Font - 1', and 'Telecommunication - 1'. The main search area has a search bar containing 'ຊ່າງວນບໍລິ' and a red box highlighting the search term. Below the search bar, it says '3 datasets found for "ຊ່າງວນບໍລິ"' and an 'Order by: Relevance' dropdown. Three dataset cards are listed, each with a red border:

- ຊ່າງວນການລົງທະບຽນບໍ່ໄຊໃຫຍ່ລະສັບຕີເຕີມຕົວທັງໝົດ (Active Number)** ໃນແຕ່ລະເດືອນ  
Source: ບົນທະກີປີປະຈຳບົນອາງຂະແນງ ເຕັກໂນໂຟຣີ ແລະ ການສື່ອນ ປີ 2022 Website: <https://mtc.gov.la/index.php?r=site/contents&id=15>  
[XLSX](#) [PDF](#)
- ຊ່າງວນທີ່ພົມ** ແລະ ບໍລິສັດທີ່ເຮັດວຽກ ໄອຊີ້ນ ໃນ ສປປ ລາວ (ICT Business in Laos)  
ແຫ່ງລ່ວງໜີນາ: ບົນທະກີປີປະຈຳບົນອາງຂະແນງ ເຕັກໂນໂຟຣີ ແລະ ການສື່ອນ ປີ 2022 <https://mtc.gov.la/index.php?r=site/contents&id=15>  
[XLSX](#) [PDF](#)
- ຊ່າງວນບໍລິມາດດາວໂຫຼດ** ແລະ ອຸນໃຊ້ ດ້ວຍເຄືອຂ່າຍບໍລິຫານລັດດັບຍອດເລັກໄຕນິຍ...  
ແຫ່ງລ່ວງໜີນາ: ບົນທະກີປີປະຈຳບົນອາງຂະແນງ ເຕັກໂນໂຟຣີ ແລະ ການສື່ອນ ປີ 2022 <https://mtc.gov.la/index.php?r=site/contents&id=15>  
[XLSX](#) [PDF](#)

After applying the tokenizer, datasets written in Lao can now be searched accurately.

However, for datasets that were created before applying the ICU tokenizer, Solr is storing the old indexes. So, we need to delete and recreate the dataset to ensure that it is also searchable.

The example datasets written in Lao can be found at <https://laos-ckan-dev.promptech.co.kr/dataset/>.



# References

## Customized CKAN Source Code

### ✓ Customized CKAN source code for Laos

› <https://github.com/2024-Laos-CKAN>

The screenshot shows the GitHub organization page for '2024-Laos-CKAN'. The page has a blue header with the organization name and a purple profile picture. The 'Overview' tab is selected. Below it, there are sections for 'Popular repositories' and 'Repositories'.

**Popular repositories:**

- ckanext-visualize** (Public)  
Forked from [keitaroinc/ckanext-visualize](#)  
CKAN extension to visualize data in DataStore  
JavaScript
- ckan-docker** (Public)  
Forked from [ckan/ckan-docker](#)  
Scripts and images to run CKAN using Docker Compose  
Python

**Repositories:**

- ckan-docker** (Public)  
Forked from [ckan/ckan-docker](#)  
Scripts and images to run CKAN using Docker Compose  
Python, 0 stars, 167 forks, 0 issues, 0 pull requests, Updated 4 minutes ago
- ckanext-visualize** (Public)  
Forked from [keitaroinc/ckanext-visualize](#)  
CKAN extension to visualize data in DataStore  
JavaScript, 0 stars, MIT license, 11 forks, 0 issues, 0 pull requests, Updated 2 weeks ago

**People:**  
This organization has no public members.  
You must be a member to see who's a part of this organization.

**Top languages:**  
JavaScript, Python

At the bottom, there is a GitHub footer with links to Terms, Privacy, Security, Status, Docs, Contact, Manage cookies, and a link to not share personal information.

# The End

Thank you