

peertube_playbook.yaml:

```
---
- name: Install Peertube
  hosts: all
  remote_user: root
  vars_files:
    - vars.yaml

tasks:
- name: Print all available facts
  debug:
    var: ansible_facts

- name: install python3-apt and gpg
  apt:
    name:
      - python3-apt
      - gpg

- name: Install the gpg key for nodejs LTS
  apt_key:
    url: https://deb.nodesource.com/gpgkey/nodesource.gpg.key
    state: present

- name: Install the nodejs LTS repos
  apt_repository:
    repo: "deb https://deb.nodesource.com/node_14.x {{ ansible_facts['distribution_release'] }} main"
    state: present
    update_cache: yes

- name: Install dependencies
  apt:
    name: "{{ item }}"
    state: present
  loop: [curl, sudo, unzip, vim, nodejs, python3-dev, python-is-python3, certbot, nginx, ffmpeg, postgresql, postgresql-contrib, openssl, g++,
make, redis-server, git, cron, wget]

- name: Create peertube user
  user:
    name: peertube
    home: /var/www/peertube
    password: $6$mysecretsalt$qJbapG68nyRab3gxvKWPUcs2g3t0oMHSHMnSKecYNpSi3CuZm.GbBqX08BE6EI6P1JUefhA0qvD7b5LSh./PU1
```

```
- name: Install psycopg2
  apt:
    name: python3-psycopg2
    state: present

- name: Create postgres user
  become: yes
  become_user: postgres
  community.postgresql.postgresql_user:
    name: "peertube"
    password: "peertube"
    db: peertube_prod

- name: create postgresql db
  become: true
  become_user: postgres
  community.postgresql.postgresql_db:
    name: "peertube_prod"

- name: Add postgresql extensions
  become: true
  become_user: postgres
  community.postgresql.postgresql_ext:
    db: peertube_prod
    name: "{{ item }}"
  loop: [pg_trgm, unaccent]

- name: Create directories
  become: true
  become_user: peertube
  file:
    path: "{{ item }}"
    state: directory
  loop:
    - /var/www/peertube/config
    - /var/www/peertube/storage
    - /var/www/peertube/versions

- name: Download peertube zip
  become: true
  become_user: peertube
  get_url:
    url: https://github.com/Chocoboxxx/PeerTube/releases/download/v4.1.1/peertube-v4.1.1.zip
    dest: /var/www/peertube/versions
```

```
- name: Extract peertube zip
  become: true
  become_user: peertube
  unarchive:
    src: /var/www/peertube/versions/peertube-v4.1.1.zip
    dest: /var/www/peertube/versions
    remote_src: yes

- name: Link peertube version
  become: true
  become_user: peertube
  file:
    src: /var/www/peertube/versions/peertube-v4.1.1
    dest: /var/www/peertube/peertube-latest
    state: link

- name: Ensure apt-transport-https is installed
  apt:
    name: apt-transport-https

- name: Add Yarn apt key
  apt_key:
    url: https://dl.yarnpkg.com/debian/pubkey.gpg

- name: Add Yarn repository
  apt_repository:
    repo: "deb https://dl.yarnpkg.com/debian/ stable main"
    filename: yarn

- name: Install Yarn
  apt:
    name: yarn

- name: Install yarn dependencies
  become: true
  become_user: peertube
  community.general.yarn:
    path: /var/www/peertube/peertube-latest
    production: true
    # --pure-lockfile

- name: Copy default config file
  become: true
  become_user: peertube
  copy:
```

```
src: /var/www/peertube/peertube-latest/config/default.yaml
dest: /var/www/peertube/config/default.yaml
remote_src: yes

- name: Copy configuration file
  become: true
  become_user: peertube
  template:
    src: production.yaml.j2
    dest: "/var/www/peertube/config/production.yaml"
    owner: peertube
    group: peertube
    mode: 0644

- name: Copy nginx configuration file
  template:
    src: nginx.conf.j2
    dest: /etc/nginx/sites-available/peertube

- name: Activate nginx peertube site
  file:
    src: /etc/nginx/sites-available/peertube
    dest: /etc/nginx/sites-enabled/peertube
    state: link

- name: Restart NGINX
  service:
    name: nginx
    state: restarted

- name: Copy systemd service config
  template:
    src: peertube.service.j2
    dest: /etc/systemd/system/peertube.service

- name: Restart systemctl service
  service:
    name: peertube
    daemon_reload: yes
    enabled: yes
    state: restarted
```

vars.yaml

```
peertube_domain: peertube.edt
```

hosts

```
peertube.edt
```

nginx.conf.j2

```
# Minimum Nginx version required: 1.13.0 (released Apr 25, 2017)
# Please check your Nginx installation features the following modules via 'nginx -V':
# STANDARD HTTP MODULES: Core, Proxy, Rewrite, Access, Gzip, Headers, HTTP/2, Log, Real IP, SSL, Thread Pool, Upstream, AIO Multithreading.
# THIRD PARTY MODULES: None.

upstream backend {
    server 127.0.0.1:9000;
}

server {
    listen 80;
    server_name {{ peertube_domain }};

    access_log /var/log/nginx/peertube.access.log; # reduce I/O with buffer=10m flush=5m
    error_log /var/log/nginx/peertube.error.log;

    location ^~ '/.well-known/acme-challenge' {
        default_type "text/plain";
        root /var/www/certbot;
    }

    ##
    # Application
    ##

    location @api {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
```

```

proxy_set_header X-Real-IP      $remote_addr;

client_max_body_size 100k; # default is 1M

proxy_connect_timeout 10m;
proxy_send_timeout    10m;
proxy_read_timeout    10m;
send_timeout          10m;

proxy_pass http://backend;
}

location / {
    try_files /dev/null @api;
}

location = /api/v1/videos/upload-resumable {
    client_max_body_size    0;
    proxy_request_buffering off;

    try_files /dev/null @api;
}

location ~ ^/api/v1/videos/(upload|([^\s]+)/studio/edit))$ {
    limit_except POST HEAD { deny all; }

    # This is the maximum upload size, which roughly matches the maximum size of a video file.
    # Note that temporary space is needed equal to the total size of all concurrent uploads.
    # This data gets stored in /var/lib/nginx by default, so you may want to put this directory
    # on a dedicated filesystem.
    client_max_body_size      12G; # default is 1M
    add_header                X-File-Maximum-Size 8G always; # inform backend of the set value in bytes before mime-encoding (x * 1.4 >=
client_max_body_size)

    try_files /dev/null @api;
}

location ~ ^/api/v1/(videos|video-playlists|video-channels|users/me) {
    client_max_body_size      6M; # default is 1M
    add_header                X-File-Maximum-Size 4M always; # inform backend of the set value in bytes before mime-encoding (x * 1.4 >=
client_max_body_size)

    try_files /dev/null @api;
}

```

```

##
# Websocket
##

location @api_websocket {
    proxy_http_version 1.1;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header     Host           $host;
    proxy_set_header     X-Real-IP      $remote_addr;
    proxy_set_header     Upgrade        $http_upgrade;
    proxy_set_header     Connection     "upgrade";

    proxy_pass http://backend;
}

location /socket.io {
    try_files /dev/null @api_websocket;
}

location /tracker/socket {
    # Peers send a message to the tracker every 15 minutes
    # Don't close the websocket before then
    proxy_read_timeout 15m; # default is 60s

    try_files /dev/null @api_websocket;
}

##
# Performance optimizations
# For extra performance please refer to https://github.com/denji/nginx-tuning
##

root /var/www/peertube/storage;

# Enable compression for JS/CSS/HTML, for improved client load times.
# It might be nice to compress JSON/XML as returned by the API, but
# leaving that out to protect against potential BREACH attack.
gzip            on;
gzip_vary       on;
gzip_types      # text/html is always compressed by HttpGzipModule
                text/css
                application/javascript
                font/truetype
                font/opentype
                application/vnd.ms-fontobject

```

```

        image/svg+xml;
gzip_min_length 1000; # default is 20 bytes
gzip_buffers    16 8k;
gzip_comp_level 2; # default is 1

client_body_timeout      30s; # default is 60
client_header_timeout    10s; # default is 60
send_timeout             10s; # default is 60
keepalive_timeout        10s; # default is 75
resolver_timeout         10s; # default is 30
reset_timedout_connection on;
proxy_ignore_client_abort on;

tcp_nopush          on; # send headers in one piece
tcp_nodelay          on; # don't buffer data sent, good for small data bursts in real time

# If you have a small /var/lib partition, it could be interesting to store temp nginx uploads in a different place
# See https://nginx.org/en/docs/http/nginx_http_core_module.html#client_body_temp_path
#client_body_temp_path /var/www/peertube/storage/nginx/;

# Bypass PeerTube for performance reasons. Optional.
# Should be consistent with client-overrides assets list in /server/controllers/client.ts
location ~
^/client/(assets/images/(icons/icon-36x36\.png|icons/icon-48x48\.png|icons/icon-72x72\.png|icons/icon-96x96\.png|icons/icon-144x144\.png|icons/icon-1
92x192\.png|icons/icon-512x512\.png|logo\.svg|favicon\.png|default-playlist\.jpg|default-avatar-account\.png|default-avatar-account-48x48\.png|default
t-avatar-video-channel\.png|default-avatar-video-channel-48x48\.png))$ {
    add_header Cache-Control "public, max-age=31536000, immutable"; # Cache 1 year

    root /var/www/peertube;

    try_files /storage/client-overrides/$1 /peertube-latest/client/dist/$1 @api;
}

# Bypass PeerTube for performance reasons. Optional.
location ~ ^/client/(.*\.(js|css|png|svg|woff2|otf|ttf|woff|eot))$ {
    add_header Cache-Control "public, max-age=31536000, immutable"; # Cache 1 year

    alias /var/www/peertube/peertube-latest/client/dist/$1;
}

# Bypass PeerTube for performance reasons. Optional.
location ~ ^/static/(thumbnails|avatars)/ {
    if ($request_method = 'OPTIONS') {
        add_header Access-Control-Allow-Origin '*';
        add_header Access-Control-Allow-Methods 'GET, OPTIONS';

```



```

    add_header Access-Control-Allow-Headers
'Range,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';
    add_header Access-Control-Max-Age      1728000; # Preflight request can be cached 20 days
    add_header Content-Type                'text/plain charset=UTF-8';
    add_header Content-Length              0;
    return 204;
}

add_header Access-Control-Allow-Origin    '*';
add_header Access-Control-Allow-Methods  'GET, OPTIONS';
add_header Access-Control-Allow-Headers
'Range,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';
add_header Cache-Control                  "public, max-age=7200"; # Cache response 2 hours

rewrite ^/static/(.*)$ /$1 break;

try_files $uri @api;
}

# Bypass PeerTube for performance reasons. Optional.
location ~ ^/static/(webseed|redundancy|streaming-playlists)/ {
    limit_rate_after      5M;

    # Clients usually have 4 simultaneous webseed connections, so the real limit is 3MB/s per client
    set $peertube_limit_rate 800k;

    # Increase rate limit in HLS mode, because we don't have multiple simultaneous connections
    if ($request_uri ~ -fragmented.mp4$) {
        set $peertube_limit_rate 5M;
    }

    # Use this line with nginx >= 1.17.0
    #limit_rate $peertube_limit_rate;
    # Or this line if your nginx < 1.17.0
    set $limit_rate $peertube_limit_rate;

    if ($request_method = 'OPTIONS') {
        add_header Access-Control-Allow-Origin    '*';
        add_header Access-Control-Allow-Methods  'GET, OPTIONS';
        add_header Access-Control-Allow-Headers
'Range,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';
        add_header Access-Control-Max-Age      1728000; # Preflight request can be cached 20 days
        add_header Content-Type                'text/plain charset=UTF-8';
        add_header Content-Length              0;
        return 204;
    }

```

```

}

if ($request_method = 'GET') {
    add_header Access-Control-Allow-Origin  '';
    add_header Access-Control-Allow-Methods 'GET, OPTIONS';
    add_header Access-Control-Allow-Headers
'Range,DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';

    # Don't spam access log file with byte range requests
    access_log off;
}

# Enabling the sendfile directive eliminates the step of copying the data into the buffer
# and enables direct copying data from one file descriptor to another.
sendfile on;
sendfile_max_chunk 1M; # prevent one fast connection from entirely occupying the worker process. should be > 800k.
aio threads;

rewrite ^/static/webseed/(.*)$ /videos/$1 break;
rewrite ^/static/(.*)$          /$1          break;

try_files $uri @api;
}
}

```

production.yaml.j2

```

listen:
  hostname: 'localhost'
  port: 9000

# Correspond to your reverse proxy server_name/listen configuration (i.e., your public PeerTube instance URL)
webserver:
  https: false
  hostname: '{{ peertube_domain }}'
  port: 80

rates_limit:
  api:
    # 50 attempts in 10 seconds
    window: 10 seconds
    max: 50
  login:

```

```
# 15 attempts in 5 min
window: 5 minutes
max: 15
signup:
  # 2 attempts in 5 min (only succeeded attempts are taken into account)
  window: 5 minutes
  max: 2
ask_send_email:
  # 3 attempts in 5 min
  window: 5 minutes
  max: 3

# Proxies to trust to get real client IP
# If you run PeerTube just behind a local proxy (nginx), keep 'loopback'
# If you run PeerTube behind a remote proxy, add the proxy IP address (or subnet)
trust_proxy:
  - 'loopback'

# Your database name will be database.name OR 'peertube'+database.suffix
database:
  hostname: 'localhost'
  port: 5432
  ssl: false
  suffix: '_prod'
  username: 'peertube'
  password: 'peertube'
  pool:
    max: 5

# Redis server for short time storage
# You can also specify a 'socket' path to a unix socket but first need to
# comment out hostname and port
redis:
  hostname: 'localhost'
  port: 6379
  auth: null
  db: 0

# SMTP server to send emails
smtp:
  # smtp or sendmail
  transport: smtp
  # Path to sendmail command. Required if you use sendmail transport
  sendmail: null
  hostname: null
```

```
port: 465 # If you use StartTLS: 587
username: null
password: null
tls: true # If you use StartTLS: false
disable_starttls: false
ca_file: null # Used for self signed certificates
from_address: 'admin@example.com'

email:
  body:
    signature: 'PeerTube'
  subject:
    prefix: '[PeerTube]'

# Update default PeerTube values
# Set by API when the field is not provided and put as default value in client
defaults:
  # Change default values when publishing a video (upload/import/go Live)
  publish:
    download_enabled: true

    comments_enabled: true

  # public = 1, unlisted = 2, private = 3, internal = 4
  privacy: 1

  # CC-BY = 1, CC-SA = 2, CC-ND = 3, CC-NC = 4, CC-NC-SA = 5, CC-NC-ND = 6, Public Domain = 7
  # You can also choose a custom licence value added by a plugin
  # No licence by default
  licence: null

p2p:
  # Enable P2P by default
  # Can be enabled/disabled by anonymous users and logged in users
  webapp:
    enabled: true

  embed:
    enabled: true

# From the project root directory
storage:
  tmp: '/var/www/peertube/storage/tmp/' # Use to download data (imports etc), store uploaded files before and during processing...
  bin: '/var/www/peertube/storage/bin/'
  avatars: '/var/www/peertube/storage/avatars/'
```

```
videos: '/var/www/peertube/storage/videos/'
streaming_playlists: '/var/www/peertube/storage/streaming-playlists/'
redundancy: '/var/www/peertube/storage/redundancy/'
logs: '/var/www/peertube/storage/logs/'
previews: '/var/www/peertube/storage/previews/'
thumbnails: '/var/www/peertube/storage/thumbnails/'
torrents: '/var/www/peertube/storage/torrents/'
captions: '/var/www/peertube/storage/captions/'
cache: '/var/www/peertube/storage/cache/'
plugins: '/var/www/peertube/storage/plugins/'
# Overridable client files in client/dist/assets/images:
# - logo.svg
# - favicon.png
# - default-playlist.jpg
# - default-avatar-account.png
# - default-avatar-video-channel.png
# - and icons/*.png (FWA)
# Could contain for example assets/images/favicon.png
# If the file exists, peertube will serve it
# If not, peertube will fallback to the default file
client_overrides: '/var/www/peertube/storage/client-overrides/'

object_storage:
  enabled: false

# Without protocol, will default to HTTPS
endpoint: '' # 's3.amazonaws.com' or 's3.fr-par.scw.cloud' for example

region: 'us-east-1'

# Set this ACL on each uploaded object
upload_acl: 'public'

credentials:
  # You can also use AWS_ACCESS_KEY_ID env variable
  access_key_id: ''
  # You can also use AWS_SECRET_ACCESS_KEY env variable
  secret_access_key: ''

# Maximum amount to upload in one request to object storage
max_upload_part: 2GB

streaming_playlists:
  bucket_name: 'streaming-playlists'
```

```

# Allows setting all buckets to the same value but with a different prefix
prefix: '' # Example: 'streaming-playlists:'

# Base url for object URL generation, scheme and host will be replaced by this URL
# Useful when you want to use a CDN/external proxy
base_url: '' # Example: 'https://mirror.example.com'

# Same settings but for webtorrent videos
videos:
  bucket_name: 'videos'
  prefix: ''
  base_url: ''

log:
  level: 'info' # 'debug' | 'info' | 'warn' | 'error'
  rotation:
    enabled: true # Enabled by default, if disabled make sure that 'storage.logs' is pointing to a folder handled by logrotate
    max_file_size: 12MB
    max_files: 20
  anonymize_ip: false
  log_ping_requests: true
  prettify_sql: false

trending:
  videos:
    interval_days: 7 # Compute trending videos for the last x days
  algorithms:
    enabled:
      - 'best' # adaptation of Reddit's 'Best' algorithm (Hot minus History)
      - 'hot' # adaptation of Reddit's 'Hot' algorithm
      - 'most-viewed' # default, used initially by PeerTube as the trending page
      - 'most-liked'
    default: 'most-viewed'

# Cache remote videos on your server, to help other instances to broadcast the video
# You can define multiple caches using different sizes/strategies
# Once you have defined your strategies, choose which instances you want to cache in admin -> manage follows -> following
redundancy:
  videos:
    check_interval: '1 hour' # How often you want to check new videos to cache
    strategies: # Just uncomment strategies you want
# -
#   size: '10GB'
#   # Minimum time the video must remain in the cache. Only accept values > 10 hours (to not overload remote instances)
#   min_lifetime: '48 hours'

```

```

#     strategy: 'most-views' # Cache videos that have the most views
#
#     size: '10GB'
#     # Minimum time the video must remain in the cache. Only accept values > 10 hours (to not overload remote instances)
#     min_lifetime: '48 hours'
#     strategy: 'trending' # Cache trending videos
#
#     size: '10GB'
#     # Minimum time the video must remain in the cache. Only accept values > 10 hours (to not overload remote instances)
#     min_lifetime: '48 hours'
#     strategy: 'recently-added' # Cache recently added videos
#     min_views: 10 # Having at least x views

# Other instances that duplicate your content
remote_redundancy:
  videos:
    # 'nobody': Do not accept remote redundancies
    # 'anybody': Accept remote redundancies from anybody
    # 'followings': Accept redundancies from instance followings
    accept_from: 'anybody'

csp:
  enabled: false
  report_only: true # CSP directives are still being tested, so disable the report only mode at your own risk!
  report_uri:

security:
  # Set the X-Frame-Options header to help to mitigate clickjacking attacks
  frameguard:
    enabled: true

tracker:
  # If you disable the tracker, you disable the P2P aspect of PeerTube
  enabled: true
  # Only handle requests on your videos
  # If you set this to false it means you have a public tracker
  # Then, it is possible that clients overload your instance with external torrents
  private: true
  # Reject peers that do a lot of announces (could improve privacy of TCP/UDP peers)
  reject_too_many_announces: false

history:
  videos:
    # If you want to limit users videos history
    # -1 means there is no limitations

```

```
# Other values could be '6 months' or '30 days' etc (PeerTube will periodically delete old entries from database)
max_age: -1

views:
  videos:
    # PeerTube creates a database entry every hour for each video to track views over a period of time
    # This is used in particular by the Trending page
    # PeerTube could remove old remote video views if you want to reduce your database size (video view counter will not be altered)
    # -1 means no cleanup
    # Other values could be '6 months' or '30 days' etc (PeerTube will periodically delete old entries from database)
    remote:
      max_age: '30 days'

    # PeerTube buffers local video views before updating and federating the video
    local_buffer_update_interval: '30 minutes'

    ip_view_expiration: '1 hour'

plugins:
  # The website PeerTube will ask for available PeerTube plugins and themes
  # This is an unmoderated plugin index, so only install plugins/themes you trust
  index:
    enabled: true
    check_latest_versions_interval: '12 hours' # How often you want to check new plugins/themes versions
    url: 'https://packages.joinpeertube.org'

federation:
  videos:
    federate_unlisted: false

    # Add a weekly job that cleans up remote AP interactions on local videos (shares, rates and comments)
    # It removes objects that do not exist anymore, and potentially fix their URLs
    cleanup_remote_interactions: true

peertube:
  check_latest_version:
    # Check and notify admins of new PeerTube versions
    enabled: true
    # You can use a custom URL if you want, that respect the format behind https://joinpeertube.org/api/v1/versions.json
    url: 'https://joinpeertube.org/api/v1/versions.json'

webadmin:
  configuration:
    edition:
      # Set this to false if you don't want to allow config edition in the web interface by instance admins
```



```
    allowed: true

#####
#
# From this point, all the following keys can be overridden by the web interface
# (local-production.json file). If you need to change some values, prefer to
# use the web interface because the configuration will be automatically
# reloaded without any need to restart PeerTube
#
# /\ If you already have a local-production.json file, the modification of the
# following keys will have no effect /\
#
#####

cache:
  previews:
    size: 500 # Max number of previews you want to cache
  captions:
    size: 500 # Max number of video captions/subtitles you want to cache
  torrents:
    size: 500 # Max number of video torrents you want to cache

admin:
  # Used to generate the root user at first startup
  # And to receive emails from the contact form
  email: 'admin@example.com'

contact_form:
  enabled: true

signup:
  enabled: false
  limit: 10 # When the limit is reached, registrations are disabled. -1 == unlimited
  minimum_age: 16 # Used to configure the signup form
  requires_email_verification: false
  filters:
    cidr: # You can specify CIDR ranges to whitelist (empty = no filtering) or blacklist
      whitelist: []
      blacklist: []

user:
  # Default value of maximum video bytes the user can upload (does not take into account transcoded files)
  # Byte format is supported ("1GB" etc)
  # -1 == unlimited
  video_quota: -1
```

```
video_quota_daily: -1

video_channels:
  max_per_user: 20 # Allows each user to create up to 20 video channels.

# If enabled, the video will be transcoded to mp4 (x264) with `faststart` flag
# In addition, if some resolutions are enabled the mp4 video file will be transcoded to these new resolutions
# Please, do not disable transcoding since many uploaded videos will not work
transcoding:
  enabled: true

# Allow your users to upload .mkv, .mov, .avi, .wmv, .flv, .f4v, .3g2, .3gp, .mts, m2ts, .mxf, .nut videos
allow_additional_extensions: true

# If a user uploads an audio file, PeerTube will create a video by merging the preview file and the audio file
allow_audio_files: true

# Amount of threads used by ffmpeg for 1 transcoding job
threads: 1

# Amount of transcoding jobs to execute in parallel
concurrency: 1

# Choose the transcoding profile
# New profiles can be added by plugins
# Available in core PeerTube: 'default'
profile: 'default'

resolutions: # Only created if the original video has a higher resolution, uses more storage!
  0p: false # audio-only (creates mp4 without video stream, always created when enabled)
  144p: false
  240p: false
  360p: false
  480p: false
  720p: false
  1080p: false
  1440p: false
  2160p: false

# Generate videos in a WebTorrent format (what we do since the first PeerTube release)
# If you also enabled the hls format, it will multiply videos storage by 2
# If disabled, breaks federation with PeerTube instances < 2.1
webtorrent:
  enabled: false

# /\ Requires ffmpeg >= 4.1
```

```
# Generate HLS playlists and fragmented MP4 files. Better playback than with WebTorrent:
#   * Resolution change is smoother
#   * Faster playback in particular with long videos
#   * More stable playback (less bugs/infinite loading)
# If you also enabled the webtorrent format, it will multiply videos storage by 2
hls:
  enabled: true

live:
  enabled: false

# Limit lives duration
# -1 == unlimited
max_duration: -1 # For example: '5 hours'

# Limit max number of live videos created on your instance
# -1 == unlimited
max_instance_lives: 20

# Limit max number of live videos created by a user on your instance
# -1 == unlimited
max_user_lives: 3

# Allow your users to save a replay of their live
# PeerTube will transcode segments in a video file
# If the user daily/total quota is reached, PeerTube will stop the live
# /\ transcoding.enabled (and not live.transcoding.enabled) has to be true to create a replay
allow_replay: true

# Allow your users to change latency settings (small latency/default/high latency)
# Small latency live streams cannot use P2P
# High latency live streams can increase P2P ratio
latency_setting:
  enabled: true

# Your firewall should accept traffic from this port in TCP if you enable live
rtmp:
  enabled: true
  port: 1935
  hostname: 'localhost'

rtmps:
  enabled: false
  port: 1936
  # Absolute path
```

```
key_file: ''
# Absolute path
cert_file: ''
hostname: 'localhost'

# Allow to transcode the live streaming in multiple live resolutions
transcoding:
  enabled: true
  threads: 2

# Choose the transcoding profile
# New profiles can be added by plugins
# Available in core PeerTube: 'default'
profile: 'default'

resolutions:
  144p: false
  240p: false
  360p: false
  480p: false
  720p: false
  1080p: false
  1440p: false
  2160p: false

video_studio:
# Enable video edition by users (cut, add intro/outro, add watermark etc)
enabled: false

import:
# Add ability for your users to import remote videos (from YouTube, torrent...)
videos:
  # Amount of import jobs to execute in parallel
  concurrency: 1

# Classic HTTP or all sites supported by youtube-dl https://rg3.github.io/youtube-dl/supportedsites.html
http:
  # We recommend to use a HTTP proxy if you enable HTTP import to prevent private URL access from this server
  # See https://docs.joinpeertube.org/maintain-configuration?id=security for more information
  enabled: false

youtube_dl_release:
  # Direct download URL to youtube-dl binary
  # Github releases API is also supported
  # Examples:
```

```
# * https://api.github.com/repos/ytdl-org/youtube-dl/releases
# * https://api.github.com/repos/yt-dlp/yt-dlp/releases
url: 'https://yt-dl.org/downloads/latest/youtube-dl'

# youtube-dl binary name
# yt-dlp is also supported
name: 'youtube-dl'

# Path to the python binary to execute for youtube-dl or yt-dlp
python_path: '/usr/bin/python3'

# IPv6 is very strongly rate-limited on most sites supported by youtube-dl
force_ipv4: false

# Magnet URI or torrent file (use classic TCP/UDP/WebSeed to download the file)
torrent:
  # We recommend to only enable magnet URI/torrent import if you trust your users
  # See https://docs.joinpeertube.org/maintain-configuration?id=security for more information
  enabled: false

auto_blacklist:
  # New videos automatically blacklisted so moderators can review before publishing
  videos:
    of_users:
      enabled: false

# Instance settings
instance:
  name: 'PeerTube'
  short_description: 'PeerTube, an ActivityPub-federated video streaming platform using P2P directly in your web browser.'
  description: 'Welcome to this PeerTube instance!' # Support markdown
  terms: 'No terms for now.' # Support markdown
  code_of_conduct: '' # Supports markdown

# Who moderates the instance? What is the policy regarding NSFW videos? Political videos? etc
moderation_information: '' # Supports markdown

# Why did you create this instance?
creation_reason: '' # Supports Markdown

# Who is behind the instance? A single person? A non profit?
administrator: '' # Supports Markdown

# How long do you plan to maintain this instance?
maintenance_lifetime: '' # Supports Markdown
```

```
# How will you pay the PeerTube instance server? With your own funds? With users donations? Advertising?
business_model: '' # Supports Markdown

# If you want to explain on what type of hardware your PeerTube instance runs
# Example: '2 vCore, 2GB RAM...'
hardware_information: '' # Supports Markdown

# What are the main languages of your instance? To interact with your users for example
# Uncomment or add the languages you want
# List of supported languages: https://peertube.cpy.re/api/v1/videos/languages
languages:
#   - en
#   - es
#   - fr

# You can specify the main categories of your instance (dedicated to music, gaming or politics etc)
# Uncomment or add the category ids you want
# List of supported categories: https://peertube.cpy.re/api/v1/videos/categories
categories:
#   - 1 # Music
#   - 2 # Films
#   - 3 # Vehicles
#   - 4 # Art
#   - 5 # Sports
#   - 6 # Travels
#   - 7 # Gaming
#   - 8 # People
#   - 9 # Comedy
#   - 10 # Entertainment
#   - 11 # News & Politics
#   - 12 # How To
#   - 13 # Education
#   - 14 # Activism
#   - 15 # Science & Technology
#   - 16 # Animals
#   - 17 # Kids
#   - 18 # Food

default_client_route: '/videos/trending'

# Whether or not the instance is dedicated to NSFW content
# Enabling it will allow other administrators to know that you are mainly federating sensitive content
# Moreover, the NSFW checkbox on video upload will be automatically checked by default
is_nsfw: false
```

```
# By default, `do_not_list` or `blur` or `display` NSFW videos
# Could be overridden per user with a setting
default_nsfw_policy: 'do_not_list'

customizations:
  javascript: '' # Directly your JavaScript code (without <script> tags). Will be eval at runtime
  css: '' # Directly your CSS code (without <style> tags). Will be injected at runtime
# Robot.txt rules. To disallow robots to crawl your instance and disallow indexation of your site, add `/` to `Disallow:`
robots: |
  User-agent: *
  Disallow:
# Security.txt rules. To discourage researchers from testing your instance and disable security.txt integration, set this to an empty string
securitytxt:
  '# If you would like to report a security issue\n# you may report it to:\nContact:
https://github.com/Chocoboxxx/PeerTube/blob/develop/SECURITY.md\nContact: mailto:'

services:
# Cards configuration to format video in Twitter
twitter:
  username: '@Chocoboxxx' # Indicates the Twitter account for the website or platform on which the content was published
# If true, a video player will be embedded in the Twitter feed on PeerTube video share
# If false, we use an image link card that will redirect on your PeerTube instance
# Change it to `true`, and then test on https://cards-dev.twitter.com/validator to see if you are whitelisted
whitelisted: false

followers:
instance:
  # Allow or not other instances to follow yours
  enabled: true
  # Whether or not an administrator must manually validate a new follower
  manual_approval: false

followings:
instance:
  # If you want to automatically follow back new instance followers
  # If this option is enabled, use the mute feature instead of deleting followings
  # /\ Don't enable this if you don't have a reactive moderation team /\
  auto_follow_back:
    enabled: false

  # If you want to automatically follow instances of the public index
  # If this option is enabled, use the mute feature instead of deleting followings
  # /\ Don't enable this if you don't have a reactive moderation team /\
  auto_follow_index:
    enabled: false
```

```

# Host your own using https://framagit.org/framasoft/peertube/instances-peertube#peertube-auto-follow
index_url: ''

theme:
  default: 'default'

broadcast_message:
  enabled: false
  message: '' # Support markdown
  level: 'info' # 'info' | 'warning' | 'error'
  dismissable: false

search:
  # Add ability to fetch remote videos/actors by their URI, that may not be federated with your instance
  # If enabled, the associated group will be able to "escape" from the instance follows
  # That means they will be able to follow channels, watch videos, list videos of non followed instances
  remote_uri:
    users: true
    anonymous: false

  # Use a third party index instead of your local index, only for search results
  # Useful to discover content outside of your instance
  # If you enable search_index, you must enable remote_uri search for users
  # If you do not enable remote_uri search for anonymous user, your instance will redirect the user on the origin instance
  # instead of loading the video locally
  search_index:
    enabled: false
    # URL of the search index, that should use the same search API and routes
    # than PeerTube: https://docs.joinpeertube.org/api-rest-reference.html
    # You should deploy your own with https://framagit.org/framasoft/peertube/search-index,
    # and can use https://search.joinpeertube.org/ for tests, but keep in mind the latter is an unmoderated search index
    url: ''
    # You can disable local search, so users only use the search index
    disable_local_search: false
    # If you did not disable local search, you can decide to use the search index by default
    is_default_search: false

# PeerTube client/interface configuration
client:
  videos:
    miniature:
      # By default PeerTube client displays author username
      prefer_author_display_name: false
      display_author_avatar: false
    resumable_upload:

```



```
# Max size of upload chunks, e.g. '90MB'
# If null, it will be calculated based on network speed
max_chunk_size: null

menu:
  login:
    # If you enable only one external auth plugin
    # You can automatically redirect your users on this external platform when they click on the login button
    redirect_on_single_external_auth: false
```