

Setting up Jenkins

The first step in getting this working is getting Jenkins up. I run Jenkins in a Docker container, and have it integrated with my SSO, which is a more complicated setup than you need, but this is the Matt-Cloud documentation, not the bare-minimum documentation, so here we go. It is still a bit less than what all I have and configuration-anonymized, but it should be enough to get the point across. The important bit is that the location on the docker host `/media/docker/jenkins/ansible` is where your ansible files will all live. If you want these to be a different volume or part of a backup or whatever, you can do as you will. You can also set up your own Github account so you can upload your eventual Jenkinsfiles to Github and then sync Jenkins with Github to get advanced Jenkinsfiles. There are ways you can be even more advanced including running the ansible jobs from within a docker container, but I have not gone down that rabbit hole. The important part is that all your Ansible role files are reachable from the Jenkins container, and having them also set up in that VS in the browser container makes editing your Ansible files easier. I think that at work the Ansible files are checked out from Github and stuffed into a docker container to run, but that is way less necessary than just getting the Jenkinsfiles into a place where Jenkins can check them out before a run.

I am also including samples of the Authelia permission and SSO settings in case you really want full-send a copy of your own Matt-Cloud. The giant on whom's SSO shoulders I stand is this random site called [lbracorp](#). It looks like they yanked their original walkthrough I used for my own Authelia, but I have my own documentation that might even be thorough enough to copy.

PROGRAMMERS



Man, I stole your code.



It's not my code.

Dockerfile.matt-code

This is the container for the VS code environment; I just have my github keys added to it so I can do that from the built-in terminal.

```
# # Use the official Python image from the Docker Hub
# FROM python:3.11

# Starting with code-server now
FROM lscr.io/linuxserver/code-server:latest

# Set the working directory
WORKDIR /workspace

# Install git & ssh
RUN apt-get update && \
    apt-get install -y git && \
    apt-get clean

# Grab SSH key pair
RUN mkdir /root/.ssh
COPY ./ssh-keys/id_rsa /root/.ssh/id_rsa
COPY ./ssh-keys/id_rsa.pub /root/.ssh/id_rsa.pub
COPY ./ssh-keys/id_matt /root/.ssh/id_matt
COPY ./ssh-keys/id_matt.pub /root/.ssh/id_matt.pub

# Secure SSH keys
RUN chmod 600 /root/.ssh/id_rsa
RUN chmod 600 /root/.ssh/id_rsa.pub
RUN chmod 600 /root/.ssh/id_matt
RUN chmod 600 /root/.ssh/id_matt.pub
RUN touch /root/.ssh/config && \
    \echo "Host *\n\tStrictHostKeyChecking no\n" >> /root/.ssh/config

# Run some updates and install some stuff
RUN apt-get update && \
    apt-get install -y python3 python3-pip python3-venv php-cli php-common && \
    apt-get clean

# RUN cp /bin/python3 /bin/python
RUN ln -s /usr/bin/python3 /usr/bin/python

# Set up Git user information
RUN git config --global user.name "github_username" && \
    git config --global user.email "github_email@domain.com" && \
```

```
git config --global init.defaultBranch "main"
```

```
# Add a script to keep the container running (optional, useful for development)
```

```
CMD ["tail", "-f", "/dev/null"]
```

Dockerfile.matt-jenkins-new

This is the Dockerfile for the Jenkins container I use. I wanted things like **ping** for troubleshooting and of course I needed **ansible**.

```
# Start with the latest official Jenkins image
```

```
FROM jenkins/jenkins:lts
```

```
# Switch to the root user to install packages
```

```
USER root
```

```
# Install prereq packages
```

```
RUN apt-get update && \
```

```
apt-get install -y python3-pip python3-venv iputils-ping net-tools nano sshpass sqlite3 && \
```

```
apt-get clean && \
```

```
rm -rf /var/lib/apt/lists/*
```

```
# Install ansible with python
```

```
# Create a virtual environment
```

```
RUN python3 -m venv /home/jenkins/venv
```

```
# Activate the virtual environment
```

```
ENV PATH "/home/jenkins/venv/bin:$PATH"
```

```
# Upgrade pip within the virtual environment
```

```
RUN pip install --upgrade pip
```

```
RUN pip install passlib
```

```
# Install the latest version of Ansible using pip
```

```
RUN pip install ansible
```

```
# Install ansible plugins
```

```
RUN ansible-galaxy collection install \
```

```
community.crypto \
```

```
community.general \
```

```
community.docker
```

```
# Switch back to the Jenkins user
```

```
USER jenkins
```

```
# Expose necessary ports
```

```
EXPOSE 8080
```

```
EXPOSE 50000
```

```
# Define the default command to run Jenkins
```

```
CMD ["bash", "-c", "jenkins-plugin-cli --list && exec java -jar /usr/share/jenkins/jenkins.war"]
```

docker-compose.yml

```
# docker-compose.yml
```

```
services:
```

```
jenkins:
```

```
hostname: jenkins
```

```
image: matt-jenkins-new
```

```
container_name: jenkins
```

```
privileged: true
```

```
user: root
```

```
ports:
```

```
- 8080:8080
```

```
volumes:
```

```
- /media/docker/jenkins/jenkins-config:/var/jenkins_home # this is for jenkins configs
```

```
- /media/docker/jenkins/ansible:/var/jenkins_home/ansible # put your ansible file structure here
```

```
restart: always
```

```
networks:
```

```
- net
```

```
ansible:
```

```
image: matt-code
```

```
container_name: ansible
```

```
ports:
```

```
- 8443:8443
```

```
environment:
```

- PUID=0
- PGID=0
- TZ=America/Los_Angeles
- PROXY_DOMAIN=https://ansible.domain.com
- DEFAULT_WORKSPACE=/workspace

volumes:

- ansible-config:/config
- /media/docker/jenkins/ansible-old:/workspace/ansible-old
- /media/docker/jenkins/ansible:/workspace/ansible

restart: always

networks:

- net

volumes:

ansible-config:

driver: local # Define the driver and options under the volume name

driver_opts:

type: none

device: /media/docker/jenkins/ansible-config

o: bind

networks:

net:

driver: bridge

ipam:

driver: default

config:

-

subnet: 10.10.0.0/24

Authelia SSO Config Sample

These are the pertinent portions of the [Authelia](#) config file.

```
# Permissions
# Jenkins Auth Tree
- domain:
```

- "jenkins.domain.com"

resources:

- "/securityRealm/finishLogin"

policy: bypass

- domain:

- "jenkins.domain.com"

subject:

- "group:Jenkins-Admin"

- "group:Jenkins-Users"

policy: two_factor

- domain:

- "jenkins.domain.com"

subject:

- "group:Jenkins-View"

policy: one_factor

SSO Portion

- id: oidc-jenkins

client_name: "Jenkins"

description: SSO-Account

secret: 'deadbeefloldeadbeeflol'

sector_identifier: 'auth.domain.com'

public: false

authorization_policy: one_factor

require_pkce: false

consent_mode: implicit

pre_configured_consent_duration: 6m

audience: []

scopes:

- openid

- groups

- email

- profile

- name

- preferred_username

redirect_uris:

- https://jenkins.domain.com/login/oauth2/code/authelia

- https://jenkins.domain.com/
- https://jenkins.domain.com/securityRealm/finishLogin
- https://jenkins.domain.com/securityRealm
- https://jenkins.domain.com/accounts/authelia/login/callback

grant_types:

- refresh_token
- authorization_code
- implicit

response_types:

- code
- token
- id_token

response_modes:

- form_post
- query
- fragment
- jwt

userinfo_signing_algorithm: none

token_endpoint_auth_method: 'client_secret_post'

Revision #5

Created 29 September 2025 01:44:46 by Matt Anderson

Updated 12 November 2025 05:44:57 by Matt Anderson