The Container File is a set of instructions for building a container image. This document was created to document the processes created to create the image for the purposes of knowledge sharing and debugging. All information was correct at time of writing and reviewing, Monday 11th December 2023.

1. Base Image: The image is based on Red Hat Universal Base Image (UBI) version 9, retrieved from ‘registry.access.redhat.com/ubi9:latest’.

```
Unset
FROM registry.access.redhat.com/ubi9:latest
```

2. Labels: Sets metadata for the image. The image is maintained by “Red Hat Services”.

```
Unset
LABEL MAINTAINERS="Red Hat Services"
```

3. Environment Variables: Defines environment variables, specifically the OpenShift API URL and token. The OUTPUT_VARS environment variable is default set to false, thus a sample settings directory is not output. To ensure proper functionality, users are advised to supply a settings directory containing the essential files: comments.yml, configs.yml, and recommendations.yml. It is strongly advised to pass this environment variable as “true” unless the required files have been copied and viewed from the Git repository/at least one container run.

4. If you need a settings directory file to configure the settings of the OCP HealthCheck, please set the OUTPUT_VARS variable as true when running the container. After the initial container run, example settings will be generated in the directory in which you mount /home/settings to. Users can modify and subsequently pass these edited settings by mounting the corresponding directory, accordingly to /home/settings for continued customization.

```
Unset
ENV OCP_API_URL https://example.com:6443
ENV OCP_TOKEN example
ENV  OUTPUT_VARS "false"
```

5. Root User: Switches to root user to install dependent packages to generate the pdf using asciidoctor.

```
Unset
USER root
```

6. Installs the required packages inside the container to generate the pdf.

```
Unset
RUN dnf update -y \
  && dnf install -y \
  python3 \
  ruby \
  jq \
  git \
  python3-pip \
  && gem install --no-document \
  "asciidoctor" \
  "asciidoctor-pdf" \
  && rm -rf /usr/local/share/gems/cache \
  && dnf clean all \
  && rm -rf /var/lib/dnf \
  && python3 -m pip install --upgrade pip \
```

7. Download and Install latest OpenShift CLI from https://mirror.openshift.com/pub/openshift-v4/clients/ocp-dev-preview/latest/, unpacking the tar file and making it executable.

```
Unset
  && BUILDNUMBER=$(curl -s
https://mirror.openshift.com/pub/openshift-v4/clients/ocp-dev-preview/latest/re
lease.txt | grep 'Name:' | awk '{print $NF}') \
  && curl -O
https://mirror.openshift.com/pub/openshift-v4/clients/ocp-dev-preview/latest/op
enshift-client-linux-${BUILDNUMBER}.tar.gz \
  && tar zxvf openshift-client-linux-${BUILDNUMBER}.tar.gz -C /usr/bin \
  && rm -f openshift-client-linux-${BUILDNUMBER}.tar.gz /usr/bin/README.md \
  && chmod +x /usr/bin/oc
```

8. Sets the working directory to /home for the container.

```
Unset

WORKDIR /home
```

9. The mounting point for /home/output is linked to the OUTPUT_DIR for persistent data. Within the container, the PDF and customizable AsciiDoc files are generated and directed to the /home/output directory. By connecting the OUTPUT_DIR to /home/output, the resulting PDF and customizable AsciiDoc files become accessible from the OUTPUT_DIR location.

Containerfile:

```
Unset
```

10. `VOLUME /home/output`

Running the container:

```
Unset
podman run -e OCP_API_URL=EXAMPLE_URL -e OCP_TOKEN=EXAMPLE_TOKEN -v
./OUTPUT_DIR:/home/output:Z -v SETTINGS_DIR:/home/settings:Z
automated_openshift_health_check:0.5`
```

11. The mounting point for /home/settings is linked to the SETTINGS_DIR for persistent data. If the consultant has well-known knowledge of the three essential files under the settings directory which includes comments.yml, configs.yml, and recommendations.yml (after copying/seeing the git repo), then they can pass as the volume mount when running the container and not set environment variable OUTPUT_VARS. But if consultant

Containerfile:

```
Unset
VOLUME /home/settings
```

Running the container having seen/knowing the settings directory (comments.yml, configs.yml, and recommendations.yml) and thus not needing to set environment variable OUTPUT_VARS:

```
Unset
podman run -e OCP_API_URL=EXAMPLE_URL -e OCP_TOKEN=EXAMPLE_TOKEN -v
./OUTPUT_DIR:/home/output:Z -v SETTINGS_DIR:/home/settings:Z
automated_openshift_health_check:0.5`
```

Running the container having not seen/knowing the settings directory (comments.yml, configs.yml, and recommendations.yml) and thus needing to set environment variable OUTPUT_VARS as “true”, in which case the corresponding mounted directory will output sample files of the settings directory (in this case the ./tmp directory - this needs at least permission of 060) for the consultant to configure and proceed as above code block:

```
Unset
podman run -e OCP_API_URL=EXAMPLE_URL -e OCP_TOKEN=EXAMPLE_TOKEN -e
OUTPUT_VARS=true -v ./OUTPUT_DIR:/home/output:Z -v ./tmp:/home/settings:Z
automated_openshift_health_check:0.5
```

12. The following copies the necessary files and directories into the image at build time.

```
Unset
COPY ./requirements.txt /home
RUN pip3 install --no-cache-dir -r requirements.txt
COPY generate-report.yml /home/
COPY collections /home/collections
COPY fonts /home/fonts
COPY images /home/images
COPY ocp-manifest /home/ocp-manifest
COPY roles /home/roles
COPY styles /home/styles
COPY templates /home/templates
COPY images /home/images/
```

13. Changes the group ownership of ‘/home/ to match the user’s group.

```
Unset
RUN chmod -R g=u /home
```

14. Switch to non-root user (UID 1001) for security purposes.

```
Unset
USER 1001
```

15. Defining Entry Point for the container:

a. Conditional Copy Operation: This part of the command checks whether the value of the OUTPUT_VARS environment variable is set to 'true'. If true, it copies all files from /home/defaults/ to /home/settings. If “false” takes the settings directory, the consultant mounts to /home/settings to configure and generate the pdf.

```
Unset
if [ $OUTPUT_VARS == 'true' ]; then cp -p /home/defaults/* /home/settings; fi
```

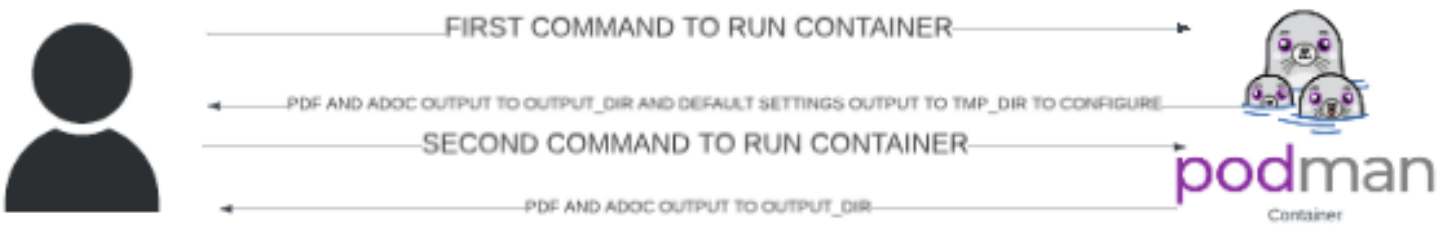
b. OpenShift Login: This logs into the OpenShift cluster using the specified token ($OCP_TOKEN) and OpenShift API URL ($OCP_API_URL). The --insecure-skip-tls-verify flag is used to skip TLS certificate verification.

```
Unset
oc login --token=$OCP_TOKEN --server=$OCP_API_URL --insecure-skip-tls-verify
```

c. Ansible Playbook Execution: After the conditional copy and OpenShift login, this part of the command executes an Ansible playbook (generate-report.yml). It passes two extra variables (-e) to the playbook: output_dir='/home/output' and settings_path=$SETTINGS_PATH. The -vvv flag increases verbosity, providing more detailed output.

```
Unset
ansible-playbook -e output_dir='/home/output' -e settings_path=$SETTINGS_PATH
generate-report.yml -vvv
```

USER FLOW:



First Command not having seen a sample settings directory and the required files (comments.yml, configs.yml, and recommendations.yml):

```
Unset
podman run -e OCP_API_URL=EXAMPLE_URL -e OCP_TOKEN=EXAMPLE_TOKEN -e
OUTPUT_VARS=true -v ./OUTPUT_DIR:/home/output:Z -v ./tmp:/home/settings:Z
automated_openshift_health_check:0.5
```

Second (and onwards) Command (after having access to the required settings directory and configuring the parameters in the three essential files: comments.yml, configs.yml and recommendations.yml):

```
Unset
podman run -e OCP_API_URL=EXAMPLE_URL -e OCP_TOKEN=EXAMPLE_TOKEN -v
./OUTPUT_DIR:/home/output:Z -v SETTINGS_DIR:/home/settings:Z
automated_openshift_health_check:0.5`
```