PGN CorpCom script examples

redeployment script on /home/developers/data/script/
/home/developers/data/script/redeployapi.sh
/home/developers/data/script/redeploycms.sh
/home/developers/data/script/redeployweb.sh

checker or temporary trigger set on hourly cron
/etc/cron.hourly/2podmanflagchecker # cronjob to check if new image exist on harbor
/etc/cron.hourly/4containerchecker # cronjob to check container state

/home/developers/data/script/containerchecker.sh

```
# /home/developers/data/script/redeployapi.sh

#!/bin/bash
set -e
date
harbor_repo=pgn-website
#prefix_img=temp-pgn-web
prefix_img=pgn-web
prefix_con=pgn-web
svc_name=be-api
branch_name=main

image_tag=latest
[ ! -z "$1" ] && image_tag=$1
echo ${image_tag}
echo ${branch_name}

base_dir=/home/developers/data

cd /home/developers/data/script

echo "o8JZNW842pmh+" | podman login --username=pgn-website --password-stdin registry.pgn.co.id

podman pull registry.pgn.co.id/${harbor_repo}/flag-${prefix_img}-${svc_name}-${branch_name}:latest


echo " ======== pull ${image_tag} images for ${svc_name} ${branch_name} ========== "

podman pull registry.pgn.co.id/${harbor_repo}/${prefix_img}-${svc_name}-${branch_name}:${image_tag}
date
echo " ========= rerun container ========== "

set +e
podman stop ${prefix_con}-${svc_name}-${branch_name}
podman rm ${prefix_con}-${svc_name}-${branch_name}

#sleep 5
set -e

podman run --name ${prefix_con}-${svc_name}-${branch_name} \
-d --restart always \
-p 8082:80 \
-v ${base_dir}/logs/api:/app/logs:Z \
registry.pgn.co.id/${harbor_repo}/${prefix_img}-${svc_name}-${branch_name}:latest
#${prefix_img}-${svc_name}-${branch_name}:latest

# if log dir not needed on host, remove or comment -v or volume option


echo " ======== delete flag from harbor ========== "
```

```
echo
"https://registry.pgn.co.id/api/v2.0/projects/${harbor_repo}/repositories/flag-${prefix_img}-${svc_name}-${br
anch_name}" \

curl -X 'DELETE' \

"https://registry.pgn.co.id/api/v2.0/projects/${harbor_repo}/repositories/flag-${prefix_img}-${svc_name}-${br
anch_name}" \
  -u 'pgn-website:o8JZNW842pmh+' \
  -H 'accept: application/json'

echo " ================== "

date
echo " ========== finish update ${svc_name} ${branch_name} ${image_tag} ========= "
echo ""
echo " ================== "
echo "remove unused images"
IMAGE_ID_REM=$(podman images -f "dangling=true" -q)
if test -z "$IMAGE_ID_REM"
then
     echo "nothing to remove"
else
     echo "remove ${IMAGE_ID_REM}"
     podman rmi $IMAGE_ID_REM
fi

echo ""
```

```
# /home/developers/data/script/redeploycms.sh

#!/bin/bash
set -e

date
harbor_repo=pgn-website
#prefix_img=temp-pgn-web
prefix_img=pgn-web
prefix_con=pgn-web
svc_name=fe-cms
branch_name=main

image_tag=latest
[ ! -z "$1" ] && image_tag=$1
echo ${image_tag}
echo ${branch_name}

base_dir=/home/developer/data

cd /home/developers/data/script

echo "o8JZNW842pmh+" | podman login --username=pgn-website --password-stdin registry.pgn.co.id

podman pull registry.pgn.co.id/${harbor_repo}/flag-${prefix_img}-${svc_name}-${branch_name}:latest


echo " ======== pull ${image_tag} images for ${svc_name} ${branch_name} ========== "

podman pull registry.pgn.co.id/${harbor_repo}/${prefix_img}-${svc_name}-${branch_name}:${image_tag}
date
echo " ========= rerun container ========== "

set +e
podman stop ${prefix_con}-${svc_name}-${branch_name}
podman rm ${prefix_con}-${svc_name}-${branch_name}

#sleep 5
set -e

podman run --name ${prefix_con}-${svc_name}-${branch_name} \
-d --restart always \
-p 8081:80 \
registry.pgn.co.id/${harbor_repo}/${prefix_img}-${svc_name}-${branch_name}:latest
#${prefix_img}-${svc_name}-${branch_name}:latest


echo " ======== delete flag from harbor ========== "
echo
"https://registry.pgn.co.id/api/v2.0/projects/${harbor_repo}/repositories/flag-${prefix_img}-${svc_name}-${br
anch_name}" \
```

```
curl -X 'DELETE' \

"https://registry.pgn.co.id/api/v2.0/projects/${harbor_repo}/repositories/flag-${prefix_img}-${svc_name}-${br
anch_name}" \
  -u 'pgn-website:o8JZNW842pmh+' \
  -H 'accept: application/json'

echo " ================== "

date
echo " ========== finish update ${svc_name} ${branch_name} ${image_tag} ========= "
echo ""
echo " ================== "
echo "remove unused images"
IMAGE_ID_REM=$(podman images -f "dangling=true" -q)
if test -z "$IMAGE_ID_REM"
then
      echo "nothing to remove"
else
      echo "remove ${IMAGE_ID_REM}"
      podman rmi $IMAGE_ID_REM
fi
echo ""
```

```
# /home/developers/data/script/redeployweb.sh

#!/bin/bash
set -e
date
harbor_repo=pgn-website
#prefix_img=temp-pgn-web
prefix_img=pgn-web
prefix_con=pgn-web
svc_name=fe-web
branch_name=main

image_tag=latest
[ ! -z "$1" ] && image_tag=$1
echo ${image_tag}
echo ${branch_name}

base_dir=/home/developer/data

cd /home/developers/data/script

echo "o8JZNW842pmh+" | podman login --username=pgn-website --password-stdin registry.pgn.co.id

podman pull registry.pgn.co.id/${harbor_repo}/flag-${prefix_img}-${svc_name}-${branch_name}:latest


echo " ======== pull ${image_tag} images for ${svc_name} ${branch_name} ========== "

podman pull registry.pgn.co.id/${harbor_repo}/${prefix_img}-${svc_name}-${branch_name}:${image_tag}
date
echo " ========= rerun container ========== "

set +e
podman stop ${prefix_con}-${svc_name}-${branch_name}
podman rm ${prefix_con}-${svc_name}-${branch_name}

#sleep 5
set -e

podman run --name ${prefix_con}-${svc_name}-${branch_name} \
-d --restart always \
-p 8080:4321 \
registry.pgn.co.id/${harbor_repo}/${prefix_img}-${svc_name}-${branch_name}:latest
#${prefix_img}-${svc_name}-${branch_name}:latest


echo " ======== delete flag from harbor ========== "
echo
"https://registry.pgn.co.id/api/v2.0/projects/${harbor_repo}/repositories/flag-${prefix_img}-${svc_name}-${br
anch_name}" \

curl -X 'DELETE' \
```

```
"https://registry.pgn.co.id/api/v2.0/projects/${harbor_repo}/repositories/flag-${prefix_img}-${svc_name}-${br
anch_name}" \
  -u 'pgn-website:o8JZNW842pmh+' \
  -H 'accept: application/json'

echo " ==================== "

date
echo " ========== finish update ${svc_name} ${branch_name} ${image_tag} ========= "
echo " ==================== "
echo "remove unused images"
IMAGE_ID_REM=$(podman images -f "dangling=true" -q)
if test -z "$IMAGE_ID_REM"
then
      echo "nothing to remove"
else
      echo "remove ${IMAGE_ID_REM}"
      podman rmi $IMAGE_ID_REM
fi
echo ""
```

```
# /home/developers/data/script/containerchecker.sh

#!/bin/bash
set -e

cd /home/developers/data/script

# Define environment (e.g., "dev" or "stg" or “main”)
branch_name="main" # Change to "stg" or “main” as needed

# List of container names without the environment suffix
CONTAINERS=(
    "pgn-web-be-api"
    "pgn-web-fe-cms"
    "pgn-web-fe-web"
)

echo ""
date

podman ps -a

for CONTAINER_BASE_NAME in "${CONTAINERS[@]}"; do
    # Construct the full container name with the environment suffix
    CONTAINER_NAME="${CONTAINER_BASE_NAME}-${branch_name}"

    echo "Checking container: $CONTAINER_NAME"

    # Check if the container is running
    container_status=$(podman inspect -f '{{.State.Status}}' "$CONTAINER_NAME" 2>/dev/null || echo
"notfound")

    # If the container is not running or doesn't exist, attempt to restart it
    if [ "$container_status" != "running" ]; then
        echo "Container '$CONTAINER_NAME' is not running. Attempting to restart..."
        if podman restart "$CONTAINER_NAME" 2>/dev/null; then
            echo "Started container '$CONTAINER_NAME'."
        else
            echo "Container '$CONTAINER_NAME' does not exist. Attempting to run..."
            # Uncomment and specify the image name if the container needs to be recreated
            # podman run --name "$CONTAINER_NAME" your_image_name
        fi
    else
        echo "Container '$CONTAINER_NAME' is already running."
    fi
    echo ""
done

date
echo ""
podman ps -a
echo " ================= "
```

```
# /etc/cron.hourly/2podmanflagchecker   # cronjob to check if new image exist on harbor

#!/usr/bin/sh

# /etc/cron.hourly/2podmanflagchecker

# Configurable Script for Cron Hourly Execution

# CONFIGURATION
INTERVAL_MINUTES=15                 # Interval in minutes (e.g., 5, 15, etc.)
USER_TYPE="non-root"                # Set "root" or "non-root"
#USER_TYPE="root"                   # Set "root" or "non-root"
NON_ROOT_USER="developers"          # Specify the non-root username
SCRIPT_DIR="/home/developers/data/script" # Directory containing scripts
LOG_DIR="$SCRIPT_DIR"               # Directory for logs

# CALCULATIONS
INTERVAL_SECONDS=$((INTERVAL_MINUTES * 60))   # Convert interval to seconds
MAX_REPETITIONS=$((60 / INTERVAL_MINUTES))   # Calculate repetitions in an hour

# COMMANDS BASED ON USER TYPE
if [[ "$USER_TYPE" == "root" ]]; then
   RUN_CMD="/bin/bash"
else
   RUN_CMD="sudo -Hu $NON_ROOT_USER /bin/bash"
fi

# FUNCTION TO EXECUTE COMMANDS
run_scripts() {
   local script=$1
   local log=$2
   echo " $RUN_CMD $script >> $log 2>&1  "
   $RUN_CMD "$script" >> "$log" 2>&1
}

# MAIN LOOP
cd $SCRIPT_DIR
for ((i = 0; i < MAX_REPETITIONS; i++)); do
   echo "Executing batch $((i + 1))/$MAX_REPETITIONS at $(date)"

   # Run scripts with logs
   run_scripts "$SCRIPT_DIR/redeployapi.sh" "$LOG_DIR/redeployapi.log"
   run_scripts "$SCRIPT_DIR/redeploycms.sh" "$LOG_DIR/redeploycms.log"
   run_scripts "$SCRIPT_DIR/redeployweb.sh" "$LOG_DIR/redeployweb.log"

   # Sleep if not the last iteration
   if ((i < MAX_REPETITIONS - 1)); then
      echo "sleep $INTERVAL_SECONDS"
      #sleep "$INTERVAL_SECONDS"
   fi
done
```

```
# /etc/cron.hourly/4containerchecker       # cronjob to check container state

#!/bin/sh

# /etc/cron.hourly/2podmanflagchecker
# CONFIGURATION
INTERVAL_MINUTES=5                  # Interval in minutes (e.g., 5, 15, etc.)
USER_TYPE="non-root"                # Set "root" or "non-root"
USER_TYPE="root"                    # Set "root" or "non-root"
NON_ROOT_USER="developers"          # Specify the non-root username
SCRIPT_DIR="/home/developers/data/script" # Directory containing scripts
LOG_DIR="$SCRIPT_DIR"               # Directory for logs

# CALCULATIONS
INTERVAL_SECONDS=$((INTERVAL_MINUTES * 60))   # Convert interval to seconds
MAX_REPETITIONS=$((60 / INTERVAL_MINUTES))   # Calculate repetitions in an hour

# COMMANDS BASED ON USER TYPE
if [ "$USER_TYPE" = "root" ]; then
    RUN_CMD="/bin/bash"
else
    RUN_CMD="sudo -Hu $NON_ROOT_USER /bin/bash"
fi

# FUNCTION TO EXECUTE COMMANDS
run_scripts() {
    script="$1"
    log="$2"
    echo " $RUN_CMD $script >> $log 2>&1  "
    $RUN_CMD "$script" >> "$log" 2>&1
}

# MAIN LOOP
if ! cd "$SCRIPT_DIR"; then
    echo "Error: Cannot change directory to $SCRIPT_DIR" >&2
    exit 1
fi

i=0
while [ "$i" -lt "$MAX_REPETITIONS" ]; do
    echo "Executing batch $(($i + 1))/$MAX_REPETITIONS at $(date)"

    # Run scripts with logs
    run_scripts "$SCRIPT_DIR/containerchecker.sh" "$LOG_DIR/containerchecker.log"

    # Sleep if not the last iteration
    i=$(($i + 1))
    if [ "$i" -lt "$MAX_REPETITIONS" ]; then
        echo "sleep $INTERVAL_SECONDS"
        sleep "$INTERVAL_SECONDS"
    fi
done
```

Examples of manual repetition checker,
Running hourly (on cron.hourly)
As a root user
With interval 5 minutes

/etc/cron.hourly/2podmanflagchecker

```
#!/usr/bin/sh
# Check
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
```

/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

sleep 300
/bin/bash /home/developers/data/script/redeployapi.sh >> /home/developers/data/script/redeployapi.log 2>&1
/bin/bash /home/developers/data/script/redeploycms.sh >> /home/developers/data/script/redeploycms.log 2>&1
/bin/bash /home/developers/data/script/redeployweb.sh >> /home/developers/data/script/redeployweb.log 2>&1

```
# /etc/cron.hourly/4containerchecker
#!/usr/sbin/sh

/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
#sleep 600
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
sleep 300
/bin/bash /home/developers/data/script/containerchecker.sh >>
/home/developers/data/script/containerchecker.log 2>&1
```