## Collecting Data Using Docker Containers on Linux

### 1. Creating a Dockerfile for a Stress Container on Linux

To collect power consumption data under high CPU utilization, we can create a Docker container that runs a stress test. Below is the Dockerfile used to create such a container:

```Dockerfile

# Use the official Ubuntu 20.04 image as the base

FROM ubuntu:20.04

# Install the 'stress' utility

RUN apt-get update && \

apt-get install -y stress

# Command to run 'stress' and utilize 8 CPU cores

CMD ["stress", "--cpu", "8"]

```

### 2. Building the Docker Image

With the Dockerfile prepared, build the Docker image by running the following command in the directory containing the Dockerfile:

```sh

docker build -t stress-container .

```

This command creates an image named `stress-container` from the instructions provided in the Dockerfile.

### 3. Running the Stress Container

To execute the stress test and generate high CPU load, run the Docker container with the following command:

```sh

docker run --rm stress-container

```

This command runs the `stress-container` image and starts the stress test. The `--rm` flag ensures that the container is removed after it stops.

### 4. Monitoring Power Consumption Using a Bash Script

To automate the process of running the stress test and logging power consumption data, we use a bash script. This script runs the stress container, monitors the system's power consumption, and saves the logs.

Below is the bash script for this purpose:

```bash

#!/bin/bash

# Function to run powerstat and save logs

save\_logs() {

battery\_percentage=$(cat /sys/class/power\_supply/BAT0/capacity)

timestamp=$(date +%s)

destination\_folder="Power\_Logs"

mkdir -p "$destination\_folder"

destination\_path="$destination\_folder/${battery\_percentage}%\_${timestamp}.log"

powerstat -d 1 30 > "$destination\_path"

}

# Run stress container

docker run --rm stress-container &

# Allow the stress container to impact the system

sleep 300 # Stress for 5 minutes

# Save power consumption logs

save\_logs

# Stop the stress container

docker stop $(docker ps -q --filter ancestor=stress-container)

```

### Running the Bash Script

To run the script, save it as `run\_stress\_and\_log.sh`, make it executable, and execute it:

```sh

chmod +x run\_stress\_and\_log.sh

./run\_stress\_and\_log.sh

```

### Conclusion

This setup allows us to generate high CPU load using Docker containers and monitor power consumption on a Linux host. The combination of Docker for creating the CPU load and scripts on the host for monitoring power consumption provides an efficient and isolated method for collecting power consumption data under controlled stress conditions.