Docker Setup with Git, GitHub, Jenkins, and Tomcat

1. Create a Dedicated Admin User

Create a new user and set a password:

```
useradd dockeradmin
passwd dockeradmin
vi /etc/ssh/sshd_config   # Modify SSH settings if needed
systemctl reload sshd.service  # Reload SSH service
Verify user creation:
su dockeradmin
Add the user to the Docker group:
usermod -aG docker dockeradmin
Check ownership and permissions:
cd /opt/
chown dockeradmin:dockeradmin docker
cat /etc/passwd | grep dockeradmin
sudo chown dockeradmin:dockeradmin docker
ls -ld docker

2. Integration with Jenkins
```

Install the required Jenkins plugin:

- Install the **"Publish Over SSH"** plugin.
- Go to **Jenkins Dashboard > Manage Jenkins > System > SSH Server**.
- Add a new SSH server with the following details:
  - **Name:** Any recognizable name
  - **Host:** Docker Server Private IP
  - **Username:** dockeradmin
  - **Under Advanced:** Set password to **0000**.

3. Deploying Application using Jenkins & Docker

Send build artifacts over SSH:

- **Transfer Set:**
  - **Source Files:** `webapp/target/*.war`
  - **Remove Prefix:** `webapp/target/`
  - **Remote Directory:** `/opt/docker`

Execute remote deployment commands:

```
cd /opt/docker || exit 1

# Build Docker image
docker build -t tomcatv1 . || exit 1

# Stop and remove existing container (if running)
if docker ps -a --format '{{.Names}}' | grep -q '^demo$'; then
    docker stop demo
    docker rm demo
fi

# Run new container
docker run -d --name demo -p 8084:8080 tomcatv1 || exit 1

# Verify container is running
docker ps | grep demo

### Notes:
- Ensure Docker is installed and running before proceeding.
- The Tomcat application will be accessible on **port 8084**.
- If Tomcat is already configured in Jenkins, remove it before proceeding.
- Use proper credentials and security settings in **production environments**.
```