SSH File Transfer and Docker Management Commands

Table of Contents

SSH File Transfer (Windows to Linux)

SCP (Secure Copy Protocol)

Transfer Single File:

```bash
scp file.txt username@linux-ip:/path/to/destination/
```

Transfer Directory Recursively:

```bash
scp -r folder/ username@linux-ip:/path/to/destination/
```

Transfer with Specific SSH Key:

```bash
scp -i /path/to/private-key file.txt username@linux-ip:/destination/
```

Transfer with Custom Port:

```bash
scp -P 2222 file.txt username@linux-ip:/destination/
```

RSYNC (Recommended for Large Transfers)

Basic File Sync:

```bash
```

```
rsync -avz file.txt username@linux-ip:/path/to/destination/
```

Directory Sync with Progress:

```
bash
rsync -avz --progress folder/ username@linux-ip:/destination/
```

Sync with Exclusions:

```
bash
rsync -avz --exclude='*.log' --exclude='node_modules/' folder/ username@linux-ip:/destination/
```

Dry Run (Test Before Transfer):

```
bash
rsync -avz --dry-run folder/ username@linux-ip:/destination/
```

Docker Container Management

Listing Containers

List Running Containers:

```
bash
docker ps
```

List All Containers (Including Stopped):

```
bash
docker ps -a
```

List with Custom Format:

```
bash
docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}\t{{.Image}}"
```

Container Operations

Start/Stop Containers:

bash

```
# Start container
docker start container-name

# Stop container
docker stop container-name

# Restart container
docker restart container-name
```

Execute Commands in Running Container:

bash

```
# Interactive shell
docker exec -it container-name /bin/bash

# Run specific command
docker exec container-name ls -la
```

Docker Image Management

Listing Images

List All Images:

bash

```
docker images
```

List with Custom Format:

bash

```
docker images --format "table {{.Repository}}\t{{.Tag}}\t{{.Size}}\t{{.CreatedAt}}"
```

List Images by Repository:

bash

```
docker images repository-name
```

Image Operations

Build Image:

bash

```
# Build from Dockerfile in current directory
docker build -t image-name:tag .

# Build with specific Dockerfile
docker build -f Dockerfile.prod -t image-name:tag .
```

Pull/Push Images:

bash

```
# Pull image from registry
docker pull image-name:tag

# Push image to registry
docker push image-name:tag
```

Docker Cleanup Operations

Remove Containers

Stop All Running Containers:

bash

```
docker stop $(docker ps -aq)
```

Remove All Containers:

bash

```
docker rm $(docker ps -aq)
```

Remove All Stopped Containers:

bash

```
docker container prune -f
```

Remove Images

Remove Specific Image:

bash

```
docker rmi image-name:tag
```

Remove All Images:

```
bash
docker rmi $(docker images -q)
```

Remove Unused Images:

```
bash
docker image prune -a -f
```

Complete System Cleanup

Nuclear Option - Remove Everything:

```
bash
docker system prune -a -f --volumes
```

Remove Specific Components:

```
bash
# Remove unused containers
docker container prune -f

# Remove unused images
docker image prune -a -f

# Remove unused volumes
docker volume prune -f

# Remove unused networks
docker network prune -f
```

Docker Compose Operations

Basic Operations

Build and Start Services:

```
bash
```

```
# Build and start in foreground
docker-compose up --build

# Build and start in background
docker-compose up -d --build

# Start without building
docker-compose up -d
```

Stop Services:

```
bash

# Stop services (containers remain)
docker-compose stop

# Stop and remove containers
docker-compose down

# Stop and remove containers, volumes, and images
docker-compose down -v --rmi all
```

Advanced Operations

Build Specific Service:

```
bash

docker-compose build service-name
```

Rebuild Without Cache:

```
bash

docker-compose build --no-cache
```

Scale Services:

```
bash

docker-compose up -d --scale web=3 --scale worker=2
```

View Logs:

```
bash
```

```bash
# View all logs
docker-compose logs -f

# View logs for specific service
docker-compose logs -f service-name
```

Execute Commands:

```bash
# Interactive shell in service
docker-compose exec service-name /bin/bash

# Run one-off command
docker-compose run service-name command
```

Complete Rebuild Workflow

Fresh Start Process:

```bash
# Step 1: Stop and remove everything
docker-compose down -v --rmi all

# Step 2: Remove any remaining containers/images (optional)
docker system prune -a -f

# Step 3: Rebuild and start fresh
docker-compose up --build -d
```

Common Flags and Options

SSH/SCP Options

- `-r` : Recursive (for directories)
- `-P` : Port number
- `-i` : Identity file (private key)
- `-v` : Verbose output

RSYNC Options

- `-a` : Archive mode (preserves permissions, timestamps, etc.)
- `-v` : Verbose

- `-z` : Compress during transfer
- `--progress` : Show progress
- `--dry-run` : Test run without actual transfer

Docker Options

- `-d` : Run in background (detached)
- `-f` : Force operation
- `-a` : All items
- `-q` : Quiet mode (only show IDs)
- `--no-cache` : Build without using cache
- `-v` : Remove volumes
- `--rmi all` : Remove all images

Prerequisites

For SSH Operations:

- SSH client installed on Windows (OpenSSH, PuTTY, or WSL)
- SSH access configured on Linux target
- Valid credentials or SSH key pair

For Docker Operations:

- Docker installed and running
- Docker Compose installed (if using compose commands)
- Appropriate permissions for Docker operations

Note: All commands assume you have appropriate permissions and network connectivity between systems.