

KASM Container Build

Archived / Decommissioned (2026-10-19): Kasm Workspaces on `u1` has been decommissioned. This page is retained as an archival reference for container packaging and KasmVNC configuration.

This guide covers the technical workflow for packaging desktop GUI applications into custom Docker images for Kasm Workspaces (version **1.19.0+**), registering them in the database and web interface, configuring advanced features, and understanding display server architecture (X11 vs. Wayland).

1. Selecting a Base Image

Kasm Workspaces runs applications inside containerized desktop environments streamed via KasmVNC. Always extend an official Kasm 1.19 base image to ensure compatibility with session management, audio forwarding, clipboard synchronization, and dynamic resizing:

Base Image	Base OS	Desktop Environment	Recommended Use Case
<code>kasmweb/terminal:1.19.0-rolling-weekly</code>	Ubuntu 24.04 (Noble)	XFCE4 + Terminal	GUI utilities, editors, lightweight tools
<code>kasmweb/core-ubuntu-noble:1.19.0</code>	Ubuntu 24.04 (Noble)	Minimal XFCE4 Core	Clean custom application desktop builds
<code>kasmweb/core-debian-bookworm:1.19.0</code>	Debian 12 (Bookworm)	Minimal XFCE4 Core	Debian-focused software packaging
<code>kasmweb/desktop:1.19.0-rolling-weekly</code>	Ubuntu 24.04 (Noble)	Full XFCE4 + Apps	Full desktop environments

2. Dockerfile Packaging Workflow

Key Packaging Rules

1. **User Isolation:** Official Kasm base images run under non-root user `1000:1000` (`kasm-user`). Switch to `USER root` for package installation and explicitly switch back to `USER 1000` at the end of the Dockerfile.
2. **Ubuntu 24.04 Package Renaming (`libasound2t64`):** In Ubuntu 24.04 (Noble), the 64-bit `time_t` transition renamed ALSA runtime packages (e.g. `libasound2` $\rightarrow$ `libasound2t64`). Use `libasound2t64` (or `libasound2*`) to prevent dependency resolution issues on 24.04 bases.
3. **Autostart Configuration:** To launch an application automatically when the workspace opens, copy an XFCE `.desktop` shortcut into `/home/kasm-user/.config/autostart/`.
4. **Custom Startup Scripts:** For applications requiring pre-launch environment variables or wrapper logic, place a script in `/dockerstartup/custom_startup.sh`.

Example Dockerfile (kasmweb/moebius:latest)

```
FROM kasmweb/terminal:1.19.0-rolling-weekly

USER root

# Install required X11, GTK3, and audio runtime dependencies (Ubuntu 24.04 Noble)
RUN apt-get update && \
    apt-get install -y --no-install-recommends \
        wget curl libnotify4 libnss3 libxss1 libxtst6 xdg-utils \
        libsecret-1-0 libgbm1 libasound2t64 && \
    wget -q -O /tmp/Moebius.deb \
https://github.com/blocktronics/moebius/releases/latest/download/Moebius.deb && \
    apt-get install -y /tmp/Moebius.deb && \
    rm -rf /var/lib/apt/lists/* /tmp/Moebius.deb && \
    mkdir -p /home/kasm-user/Desktop /home/kasm-user/.config/autostart && \
    cp /usr/share/applications/moebius.desktop /home/kasm-user/Desktop/moebius.desktop && \
    cp /usr/share/applications/moebius.desktop /home/kasm-user/.config/autostart/moebius.desktop && \
    chmod +x /home/kasm-user/Desktop/moebius.desktop && \
    chown -R 1000:1000 /home/kasm-user/Desktop /home/kasm-user/.config

USER 1000
```

Building inside Standalone Kasm Containers

When running standalone Kasm Workspaces (`lscr.io/linuxserver/kasm`), Docker images must be built inside the Kasm container engine (`kasm`):

```
sudo docker exec -i kasm docker build -t kasmweb/moebius:latest - < Dockerfile
```

3. Registering the Workspace in the Web Interface

Automated Database Registration (`kasm_db`)

In standalone Kasm deployments, workspaces can be registered directly in the PostgreSQL database (`kasm_db`). Each entry in the `images` table defines the container parameters, and `group_images` grants access to user groups.

“ **[!IMPORTANT]** Unused foreign-key UUID columns (`session_banner_id`, `server_id`, `zone_id`, `filter_policy_id`, `server_pool_id`) must be explicitly set to `NULL`. Empty strings (`''`) cause PostgreSQL UUID syntax errors.

```
-- 1. Insert Workspace definition into images table
INSERT INTO images (
    image_id, cores, description, docker_registry, docker_token, docker_user,
    image_src, enabled, available, friendly_name, hash, memory_bytes, name,
    session_banner_id, session_banner_force_disabled, x_res, y_res, run_config,
    volume_mappings, persistent_profile_path, restrict_to_network, restrict_network_names,
    allow_network_selection, restrict_to_server, server_id, exec_config,
    restrict_to_zone, zone_id, filter_policy_id, filter_policy_force_disabled,
    categories, require_gpu, gpu_count, gpu_graphics_acceleration, gpu_video_encoding,
```

```

hidden, notes, override_egress_gateways, image_type, server_pool_id, link_url,
cpu_allocation_method, uncompressed_size_bytes, launch_config, rdp_client_type,
is_remote_app, remote_app_icon, remote_app_name, remote_app_program, remote_app_args,
created_at, updated_at, enforce_workspace_persistence
) VALUES (
    'a1111111-60eb-4000-a000-00000000060eb', 2,
    'Moebius ANSI and ASCII Art Editor full desktop client.',
    'https://index.docker.io/v1/', NULL, NULL,
    'img/thumbnails/terminal.png', true, true, 'Moebius', NULL, 2902458368,
    'kasmweb/moebius:latest', NULL, false, 1280, 720, '{"hostname": "moebius"}',
    '{}', '/data/kasm_profiles/{username}/', false, '[]', false, false, NULL, '{}',
    false, NULL, NULL, false, '["Design"]', false, 0, '["MESA"]', '["SW"]',
    false, NULL, false, 'Container', NULL, NULL, 'Inherit', 1500000000, '{}', NULL,
    false, NULL, NULL, NULL, NULL, NOW(), NOW(), true
) ON CONFLICT (image_id) DO NOTHING;

-- 2. Assign image to default All Users group (68d557ac-4cac-42cc-a9f3-1c7c853de0f3)
INSERT INTO group_images (group_image_id, group_id, image_id)
VALUES (
    'a2222222-60eb-4000-a000-00000000060eb',
    '68d557ac-4cac-42cc-a9f3-1c7c853de0f3',
    'a1111111-60eb-4000-a000-00000000060eb'
) ON CONFLICT (group_image_id) DO NOTHING;

```

Manual UI Registration

1. Navigate to **Kasm Admin Panel** -> **Workspaces** -> **Add Workspace**.
2. **Workspace Type:**
3. **Friendly Name:**
4. **Docker Image:**
5. **Cores / Memory:** / (MB)
6. **Persistent Profile Path:**
7. Click **Save**.

4. Advanced Configuration Options

When packaging complex software, evaluate these key operational options:

1. Persistent Profiles (`persistent_profile_path`)

- **Purpose:** Retains user files, application settings, and browser profiles across session restarts.
- **Configuration:** Set `persistent_profile_path` to `/data/kasm_profiles/{username}/`.
- **Behavior:** When a session launches, Kasm bind-mounts the user's persistent directory to `/home/kasm-user/`. Set `enforce_workspace_persistence = true` to prevent unmanaged ephemeral sessions.

2. GPU Hardware Acceleration (AMD, Intel & NVIDIA)

- **Purpose:** Offloads WebGL, 3D rendering, compute, and video stream encoding to host GPUs.
- **Configuration for AMD & Intel GPUs:**
 - Pass `/dev/dri` and `/dev/kfd` device nodes in `run_config`:

```
{
  "devices": [
    "/dev/dri:/dev/dri",
    "/dev/kfd:/dev/kfd"
  ],
  "group_add": ["render", "video"]
}
```

- **Graphics Acceleration:** Set `gpu_graphics_acceleration: ["MESA"]` (RADV Vulkan / OpenGL).
 - **Hardware Video Encoding:** Set `gpu_video_encoding: ["VAAPI", "SW"]` for hardware H.264/HEVC/AV1 encoding.
 - Set `require_gpu = true` and `gpu_count = 1`.
- **Configuration for NVIDIA GPUs:**
 - Ensure NVIDIA Container Toolkit is enabled on the host.
 - Set `gpu_graphics_acceleration: ["NVIDIA"]` and `gpu_video_encoding: ["NVENC", "SW"]`.

3. Bidirectional Audio & Microphone Forwarding

- **Purpose:** Streams 48 kHz stereo audio output and enables low-latency WebRTC microphone input.
- **Configuration:**
 - Kasm base images include ALSA and PulseAudio (`pulseaudio -D`) bridged to WebRTC / WebSocket audio codecs. Ensure packages install `libasound2t64` and `pulseaudio-utils`.
 - Enable microphone pass-through in `launch_config` if recording or voice chat is required:

```
{  
  "allow_microphone": true  
}
```

4. Gamepad & Controller Passthrough

- **Purpose:** Passes browser HTML5 Gamepad API inputs directly into container `/dev/input/js*` virtual joystick devices.
- **Configuration:** In `launch_config`, enable gamepad forwarding:

```
{  
  "allow_gamepad": true  
}
```

5. Network Isolation & Security (`restrict_to_network`)

- **Purpose:** Restricts container egress or connects workspaces to internal backend VLANs/docker networks.
- **Configuration:** Set `restrict_to_network = true` and specify target Docker network names in `restrict_network_names`:

```
["infra_net", "isolated_workspaces"]
```

5. Display Server Architecture: X11 vs. Wayland

1. Current KasmVNC Standard (X11)

- **Architecture:** Official Kasm Workspaces desktop containers (`kasmweb/desktop`, `kasmweb/terminal`) operate on the **X11 display server protocol** using `Xvfb` / `Xorg` virtual framebuffers on display `:1`.
- **Why X11 is Default:**
 - **Input Injection:** Keyboard and mouse event simulation across web browsers maps reliably to X11 input extensions (`libxtst6`).
 - **Dynamic Resolution Resizing:** Automatic desktop resizing when browser windows resize relies on `xrandr` / X11 Resize and Rotate extensions.
 - **Clipboard Synchronization:** Web-to-desktop clipboard sharing uses standard X11 selection buffers (`PRIMARY` and `CLIPBOARD`).

2. Wayland & Xwayland Support

- **Can Kasm run Wayland?:** Yes. Modern KasmVNC releases support Wayland compositors (such as Sway or Weston) via **Xwayland** and virtual DRM/KMS backend framebuffers.
- **Limitations in Wayland Mode:**
 - **Clipboard & Capture:** Pure Wayland restricts client-to-client screen capture and clipboard access by design. Applications running natively under Wayland require `xdg-desktop-portal-wlr` or compositor-specific protocols to share clipboards with KasmVNC.
 - **Application Compatibility:** Most packaged Linux GUI applications (Electron, GTK3, Qt5) run under X11 or Xwayland by default. For production packaging in Kasm Workspaces, **X11 remains the recommended display server** unless testing specialized Wayland-only compositors.

Revision #2

Created 2026-07-31 16:56:03 UTC by admin

Updated 2026-10-19 20:42:20 UTC by luKe