Building a Custom WSL2 Kernel

This Ansible Playbook Will Enable You to Build a Custom Kernel

Overview

Currently, when running under Windows, Docker Desktop has moved from using a full virtual machine as a container environment to using the Windows Subsystem for Linux, version 2 (WSL2). Unlike a Hyper-V backed virtual machine, the WSL2 implementation does not provide a fully isolated environment. Rather, it uses a shared utility VM with a custom kernel that is shared across the entire WSL2 subsystem.

That said, it is worth noting that security should be enforced from the Windows side, as WSL2 runs as a standard Windows Service. That means that all WSL2 users - including root - have the same privileges within Windows as the service account that owns the service.

For enterprise administrators, it may be necessary to exercise a more granular level of control over Docker deployment using WSL2. This repo provides an Ansible Playbook that automates the process of downloading, building, and deploying a WSL2 kernel using the official microsoft sources.

Requirements

In order to run this playbook, the following are required:

1. A Windows installation with WSL2 configured.
2. A Ubuntu distribution that runs under WSL2; the default Ubuntu distribution is fine for our purposes, however if you have other needs (or existing tooling) you can use any distribution that runs under WSL2, although you may need to change some of the commands being used by Ansible.
3. An installed and working Ansible distribution. This was tested with 2.9.6.
4. The ability to use `sudo` to become root within the WSL2 distribution.

Other Notes

This process was tested with the built-in Windows terminal; using other terminals should be fine, but has not been tested.

Key Files

wsl2-vars.yml

This file contains the variables that can be edited/modified in order to build a new kernel. Comments are provided inline inside the file.

wsl2-kernel.yml

This is the Ansible playbook; comments are provided inline inside the file.

hosts.ini

This is the hosts file used by Ansible; since this process is designed to run locally we are using the local drivers and communicating with localhost. You should not need to change this.

Building a Kernel

1. Clone the Whalesongs repository to your workstation.
2. Change to this directory.
3. Edit the wsl2-vars.yml file. At a minimum you will need to specify:
    i. Your Windows account name.
    ii. Your WSL2 account name.
    iii. The version of the WSL2 Kernel to use as your base.
4. Run the process with `ansible-playbook -i hosts.ini wsl2-kernel.yml -K`
5. Answer the `BECOME password:` prompt with the password for your WSL2 account; this is used to elevate permissions inside the Linux distribution.
6. If you wish to edit the source files, you will need to do so when you see this message:

```
TASK [Display kernel version]
*********************************************************************************************
ok: [localhost] => {
"msg": "The WSL2 Kernel repository has been cloned. If you wish to make any changes to the kernel
files, please do so now. Once done, you can proceed with the next steps." }

TASK [Pause for user to edit kernel files]
*****************************************************************************
[Pause for user to edit kernel files]
Once you've made the desired changes to the kernel files, press Enter to continue.:
```

7. When you are done with your changes (if any) press enter.
8. The process completes and explains the next steps:

```
 TASK [Instruct user to shutdown WSL]
*********************************************************************************************
 ok: [localhost] => {
 "msg": "Please manually shut down WSL by running 'wsl --shutdown' in PowerShell or Command Prompt."
}

 TASK [Instruct user to verify new kernel]
*****************************************************************************
 ok: [localhost] => {
 "msg": "After restarting WSL, verify the new kernel by running '$ uname -r' in the WSL terminal." }
```

9. Once you restart your linux distribution you will be running the newly compiled kernel.

Example Run

SOFTWARE DISCLAIMER

This software is provided as a "Proof of Concept" (PoC) and is not intended for production use.

NO WARRANTIES: The author expressly disclaims any warranty for this software. The software and any related documentation is provided "as is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. The entire risk arising out of use or performance of the software remains with the user.

NO LIABILITY FOR DAMAGES: In no event shall the author be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use or inability to use this product, even if the author has been advised of the possibility of such damages.

USE AT YOUR OWN RISK: This software is intended for educational or demonstration purposes only. Users are strongly cautioned against using it in production or mission-critical environments. If you choose to use the software, it is at your own discretion and responsibility to ensure that it does not cause any harm or issues to your systems or data.

MODIFICATIONS: Users are free to modify the software for their own use, but redistribution should include this disclaimer.

Always take a backup of your data and test any software in a controlled environment before any widespread use.

Citations and Helpful Information

- Ansible
- WSL2 Kernel Source
- Another Automated Build Script
- WSL2 FAQ