

Installation notes:

AWS Palace installation using Apptainer

This document describes the method how Palace was installed for developing the gds2palace workflow for IHP. There are many other possible options to install Palace, so please take this as an example only.

The file versions mentioned here were up-to-date in Summer 2025, you might check for newer versions also.

Operating system

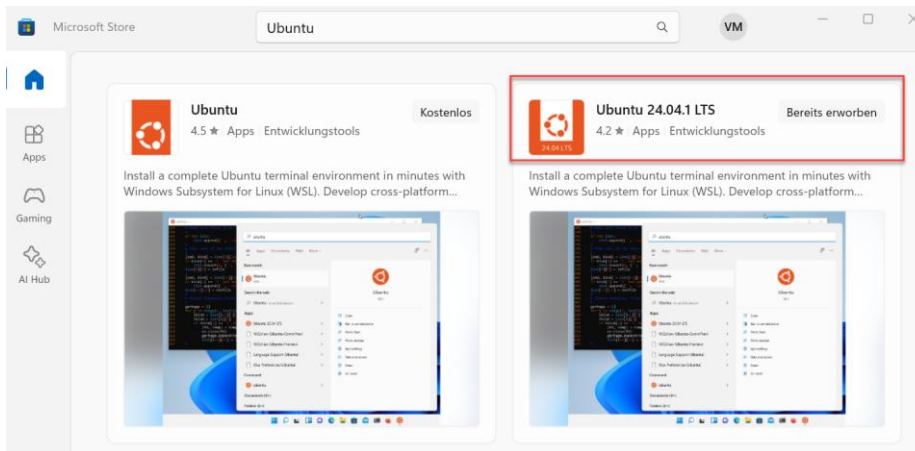
This documents refers to installation on **Linux** (Ubuntu 24) on AMD64 (normal desktop PC).

Linux installation

Please skip the “Windows installation” step, and proceed to the “Install Apptainer” chapter.

Windows installation

If you want to run Palace on a Windows system, you need to make use of the Windows Subsystem for Linux (WSL). This is rather easy to install: in Windows, open the “Microsoft Store” program and search for Ubuntu 24.04.1 LTS, which is a free download. Install this to activate Ubuntu with the Windows Subsystem for Linux.



You now have an Ubuntu subsystem that can be started from command line by the “wsl” command. From here, you can use the same installation procedure as described below.

```
volker@OMEN15: /mnt/c/Use  x  +  v
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\volker>wsl
volker@OMEN15:/mnt/c/Users/volker$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.2 LTS
Release:        24.04
Codename:       noble
volker@OMEN15:/mnt/c/Users/volker$ |
```

Windows command line

command wsl starts Ubuntu

Linux home dir = Windows home dir

Install Apptainer (previously called Singularity)

In this document, we describe the use of Palace from an apptainer container solution, which can be easily copied between machines once we have created the apptainer file.

To get started, we first need to install the apptainer software.

```
$ sudo apt update
$ sudo apt install -y wget
$ cd /tmp
$ wget https://github.com/apptainer/apptainer/releases/download/v1.4.0/apptainer_1.4.0_amd64.deb
$ sudo apt install -y ./apptainer_1.4.0_amd64.deb
```

Apptainer is now ready for use from the commandline, using command apptainer or singularity, both commands are identical.

Build palace.sif

When the apptainer software is installed, we can now create the Palace container (apptainer) file.

Get source files

```
mkdir /home/volker/github
cd /home/volker/github
git clone https://github.com/aws-labs/palace.git
```

This places the Palace source in /home/volker/github/palace

Build using Singularity/Apptainer

```
cd /home/volker
singularity build palace.sif /home/volker/github/palace/singularity/singularity.def
```

This creates container palace.sif in /home/volker

The build process can take 30 to 60 minutes, you might see some warning during build but the final message should tell you that the file was created successfully.

When you are ready, continue to the next step, running Palace from the apptainer file.

Run Palace

If we have palace.sif available in /home/volker:

```
cd /home/volker
apptainer shell palace.sif
```

The prompt changes to Apptainer> and opens a shell window.

We can run palace from there, with the *.json model name as the only parameter. All the normal file system can be accessed from within apptainer.



Exit from apptainer: use command “exit”

Running Palace with the IHP gds2palace workflow

The gds2palace workflow for IHP will create model files (*.msh and config.json) that we can simulate using Palace.

To simplify the overall workflow, we can add some scripts now. The script directory should be added to your PATH, so that these scripts can be executed from anywhere in your file system.

The first script “run_palace” is used to start Palace from the apptainer image, with a number of threads that makes good use of your available CPU resources. In the example below, we have configured with for 8 threads using the -np 8 parameter.

```
#!/bin/bash
apptainer exec ~/palace.sif palace -np 8 $1
```

This is all you need to start a Palace simulation job from the command line using

```
run_palace somemodel.json
```

The next script is used after running the simulation, to convert Palace results in *.csv format to standard *.snp Touchstone file format. This script is called “combine_snp” and calls a Python code that actually does the job.

```
#!/bin/bash
~/venv/palace/bin/python3 ~/scripts/combine_extend_snp.py
```

That Python code required the scikit-rf Python library, and we had used a venv named “palace” to install this library together with the gdsapy and gmsh libraries required by the gds2palace workflow.

All these scripts and codes can be found in the scripts directory of the gds2palace github repository, you just need to adjust them to your actual path settings.