

Open Cloud Testbed - Building P4-FPGA applications

In the OCT, we provide building machines for building P4-FPGA applications. You can use our build machines to either build ESNet based or PRATE-based P4-FPGA applications. Here are the instructions for getting access to a VM and setting up the build machines.

1. Requesting an account on the build machine

To start with the build process, you'll need access to a VM that we have set up for this project. Please fill out the following Google form so we can set up your account on this machine.

https://docs.google.com/forms/d/16lZhwckYJlBL5gaC9tdNQmynjGhXOnb87_Le_b55vBM/

2. Build machine setup

You'll access the build machine via an intermediate bastion host. A bastion host, also known as a jump server or a gateway host, serves as an intermediary point for securely accessing the build machine. The information provided below will give you details on how to access the build machine through the bastion host.

- a. Create a configuration file (use any name you like, for example config) in your ~/.ssh directory with the details provided below. There are two options for this case. First if a config file already exists you can add the information below to that file. Second you can create a new file (named as you like) that includes the information shown below. Replace <username> and <private key> with your actual credentials. If you plan to use VNC for accessing the build machine, you can include the VNC port number. To do this, uncomment the line `# LocalForward <vnc port> localhost:<vnc port>` and specify your VNC port number to facilitate VNC port forwarding. Example: `LocalForward 5901 localhost:5901`

```
Host p4-tools
  HostName 192.168.0.22
  User <username>
  IdentityFile <private key>
  # LocalForward <vnc port> localhost:<vnc port>
  ProxyJump bastion

Host bastion
  HostName 199.94.60.18
  User <username>
  IdentityFile <private key>
```

- b. Access the build machine by running `ssh p4-tools`, if you use the file config, if not call `ssh -F <config_file_name> p4-tools`
- c. (Optional) When you are on the build machine, you can start a VNC server instance by running the following command:
`vncserver -geometry <screen resolution> :<port number>`
Example: `vncserver -geometry 1920x1080`

When starting a VNC server without explicitly specifying the display number as shown above, the server will typically attempt to use the default port number (5901). If the default port is already in use, the VNC server will increment the port number and continue checking for availability until it finds an open port.

Alternatively, to start the VNC server with a specific display number, you can use the following command.

```
vncserver -geometry 1920x1080 :<display number>
```

Example: `vncserver -geometry 1920x1080 :6`

In this case, the VNC server will attempt to use the port number associated the the display number (5905). If it is not available, you will see an error message like the following:

A Xtigervnc server is already running for display :6 on machine p4-tools.

If you encounter this error, try using a different port number. If your VNC server is successfully created, you should see a message like the following:

New Xtigervnc server 'p4-tools:6 (ubuntu)' on port 5906 for display :6.
Use `xtigervncviewer -SecurityTypes VncAuth -passwd /home/ubuntu/.vnc/passwd :6` to connect to the VNC server.

Edit the SSH configuration file to include this port number.

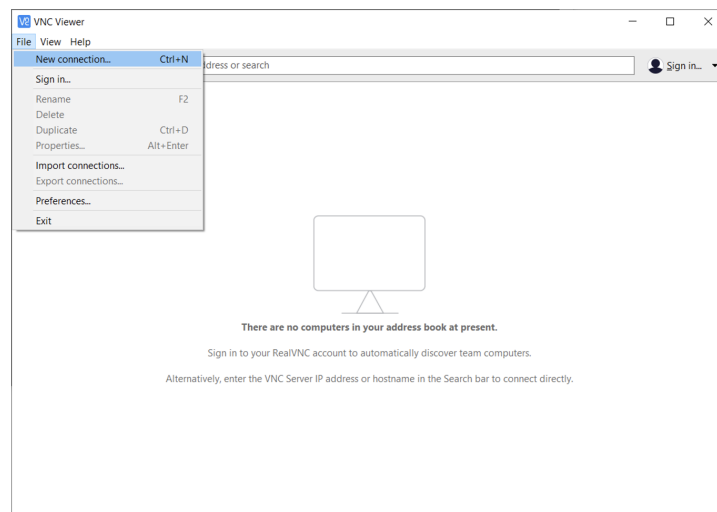
```
LocalForward 5906 localhost:5906
```

Next, close the current SSH connection and reconnect. The server is now ready to accept incoming VNC connections through the specified port number.

You may use any VNC client application to connect to the VNC server. In the following, you'll see using RealVNC client which you can download from <https://www.realvnc.com/en/connect/download/viewer/>.

Install RealVNC viewer on your local machine.

Open the VNC viewer. Click File → New Connection.



Enter the VNC Server name as `localhost:<port_number>` where `<port_number>` is the VNC port number. Click OK.

build_machine - Properties

General Options Expert

VNC Server: localhost:5901

Name: build_machine

Labels

To nest labels, separate names with a forward slash (/)

Enter a label name, or press Down to apply existing labels

Security

Encryption: Let VNC Server choose

☒ Authenticate using single sign-on (SSO) if possible


☒ Authenticate using a smartcard or certificate store if possible

Privacy


☒ Update desktop preview automatically

Cancel OK

You should see an icon on the VNC viewer window which you double click and enter the VNC password. Check the Remember password checkbox to avoid having to enter your password during future login attempts.

 **Authenticate to VNC Server**
localhost::5901 (TCP)

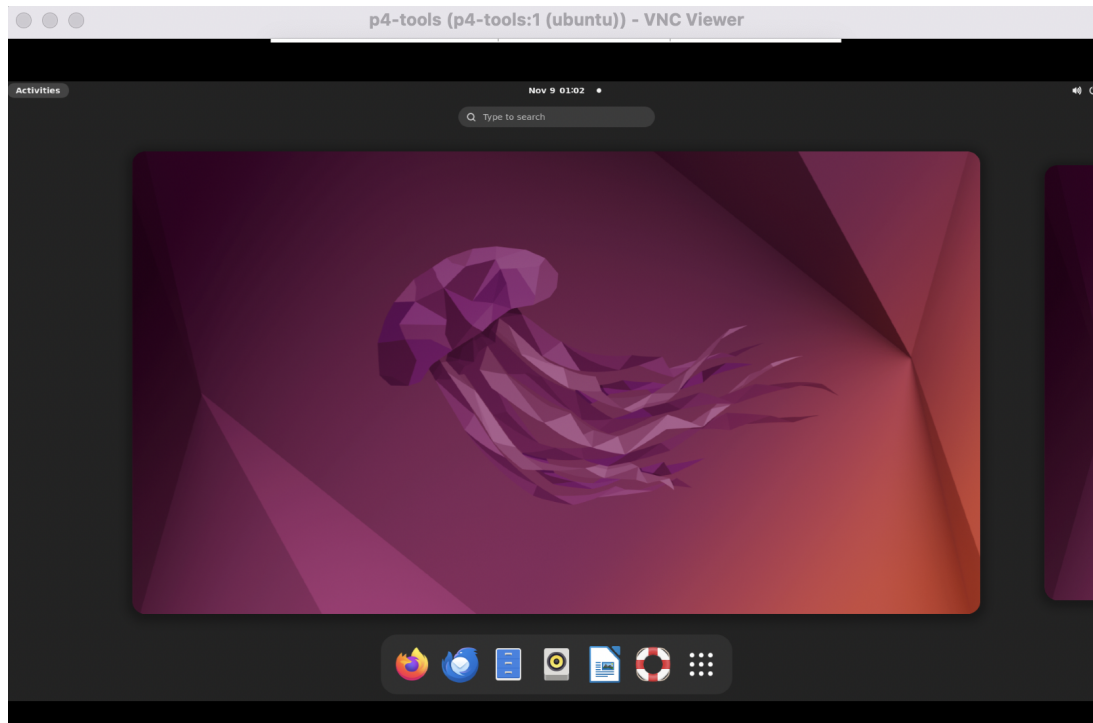
Username:

Password: 

☒ Remember password [Forgot password?](#)

Cancel OK

Click OK, and you should be able to see the VNC desktop now.



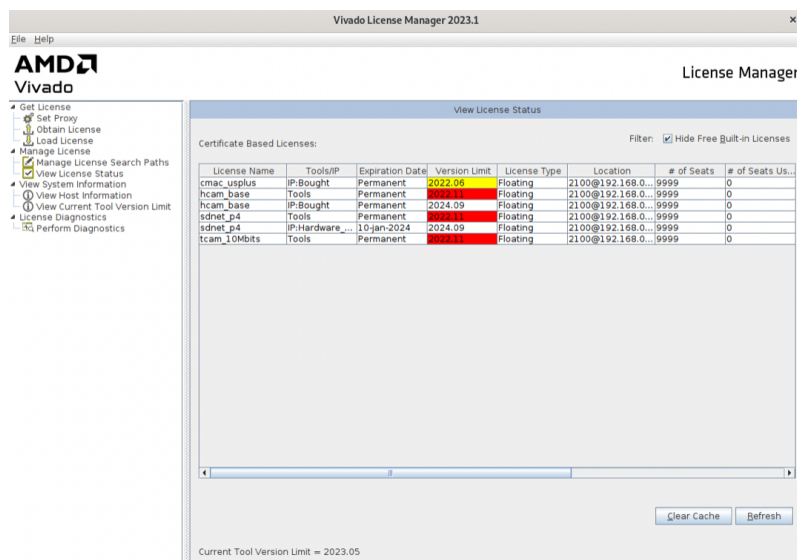
3. Build P4 applications:

You can use this build machine for both ESNNet P4 flow and OCT-P4 flow. However, they are using different versions of the Vivado.

To build P4 applications, you'll require the VitisNet P4 license. To check out the license, run the following command before you start the build.

```
export XILINXD_LICENSE_FILE=2100@192.168.0.54
```

To make sure that your license checkout was successful, run `vlm` (Vivado License Manager). You should see the P4 licenses:



For ESNet P4 flow:

You can build both hardware and firmware on the OCT build machines. For the hardware, you need to run “source /tools//Xilinx/Vivado/2023.1/settings64.sh” to load the necessary environment path for the build flow. After that, refer to ESNet flow for further build [instructions](#).

For building the firmware, you can refer to this building [instructions](#). **Please note we have already installed docker and docker-compose. Please do not install it again. You can directly follow the instructions to build.**

For OCT-P4 flow:

Run “source /tools/Xilinx/Vivado/2021.2/settings64.sh” to load the path. And then refer to the P4Framework documents. (<https://github.com/OCT-FPGA/P4Framework/tree/master>)