THE UNIVERSITY OF DODOMA



COLLEGE OF INFORMATICS AND VIRTUAL EDUCATION(CIVE)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(CSE)
COURSE NAME: SOFTWARE DEFINED NETWORKS: CG 311
LAB 3 ASSIGNEMENT

SN	NAMES	REGISTRATION NUMBER	PROGRAMME
2.	AGRIPINA EMMANUEL WILLIAM	T23-03-16504	BSC. CNISE
3.	EVES VITUS SYLVESTER	T23-03-11142	BSC. CNISE

Step 1: Title & Objective

Lab 3: Integrating an SDN Controller with Open vSwitch
Objective: To demonstrate SDN principles by connecting an Open vSwitch to an SDN controller (ONOS or OpenDaylight), observing the separation of control and data planes, and enabling reactive forwarding.

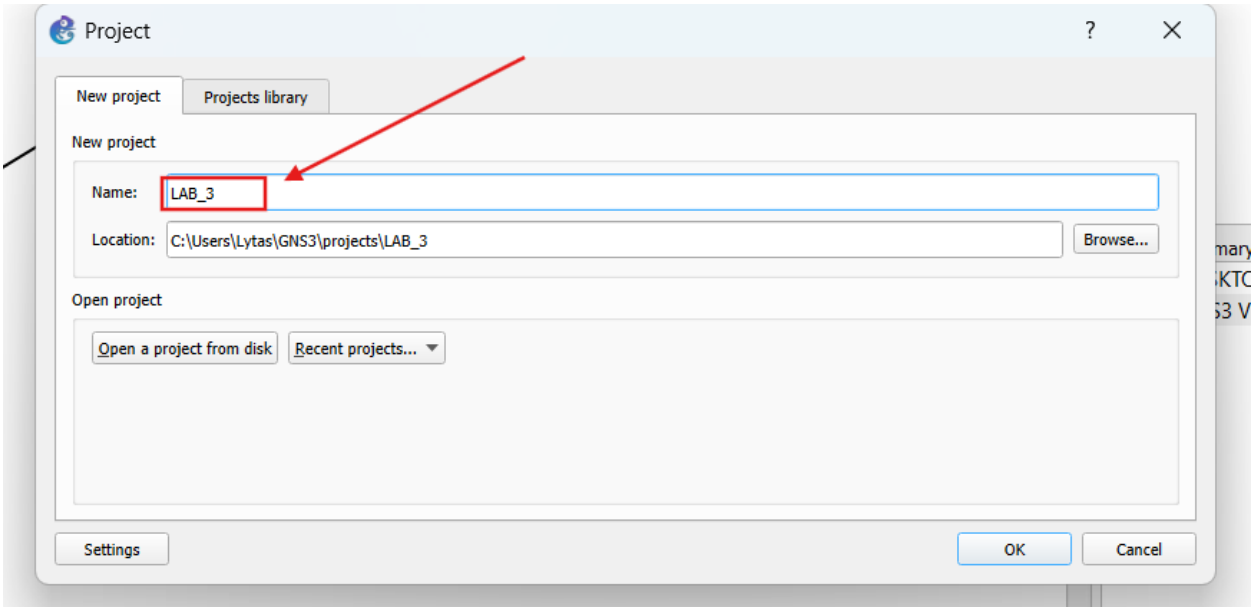
Step 2: Technical Background

In SDN, the **control plane** (logic) is separated from the **data plane** (forwarding). An SDN controller communicates with switches via the **OpenFlow protocol**, pushing flow rules dynamically. This lab extends Lab 2 by adding a controller to manage OVS, moving from standalone learning switching to centrally programmed forwarding.

Step 3: Topology Setup

Perform:

First create a project named LAB_3



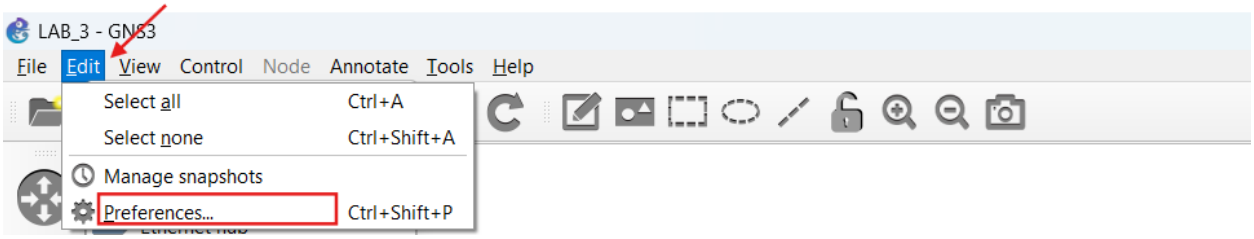
1. In GNS3, drag and drop:
    i. 1× SDN Controller (ONOS or OpenDaylight)
    ii. 1× OVS Appliance
    iii. 2× VPCS (Host A, Host B)
    iv. 1× NAT node (optional, for controller updates)

Steps for ONOS Docker Container:

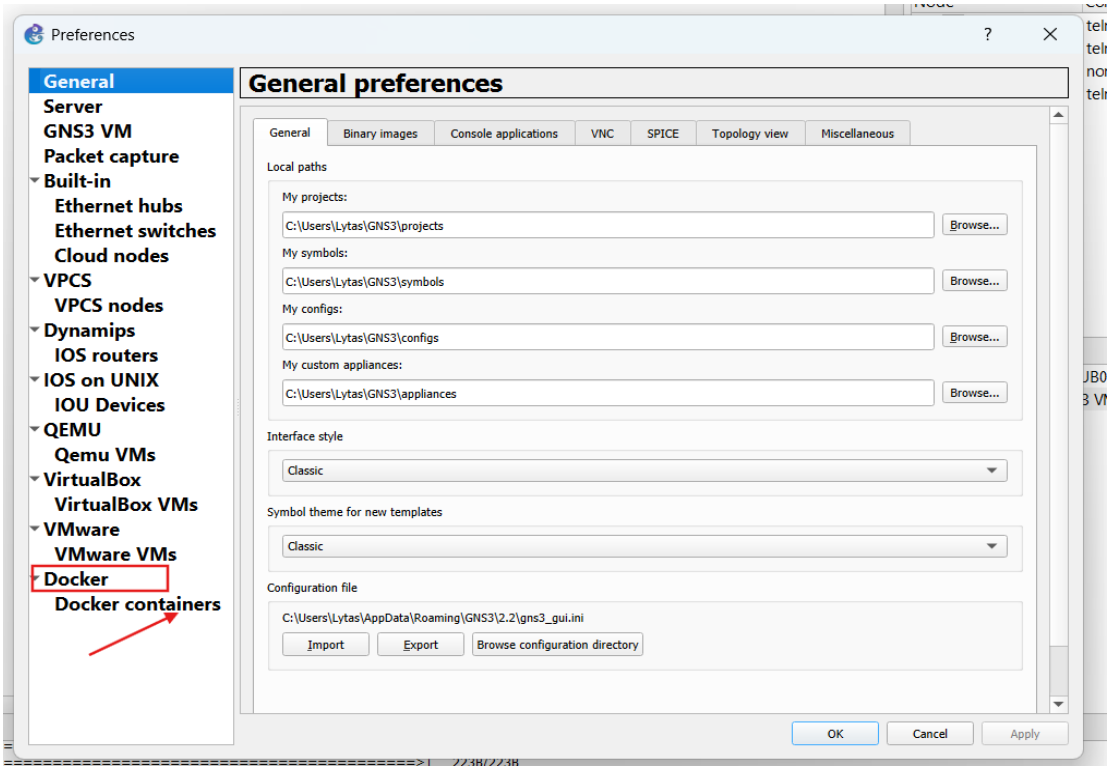
1. Ensure Docker is installed on your GNS3 VM.
2. In GNS3:

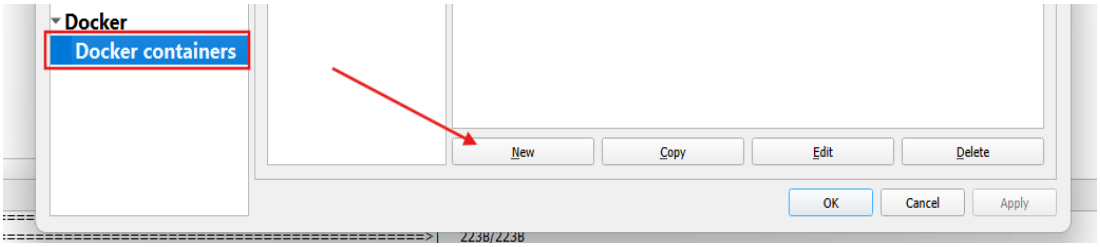
Edit → Preferences → Docker → New

Edit → Preferences → Docker → New

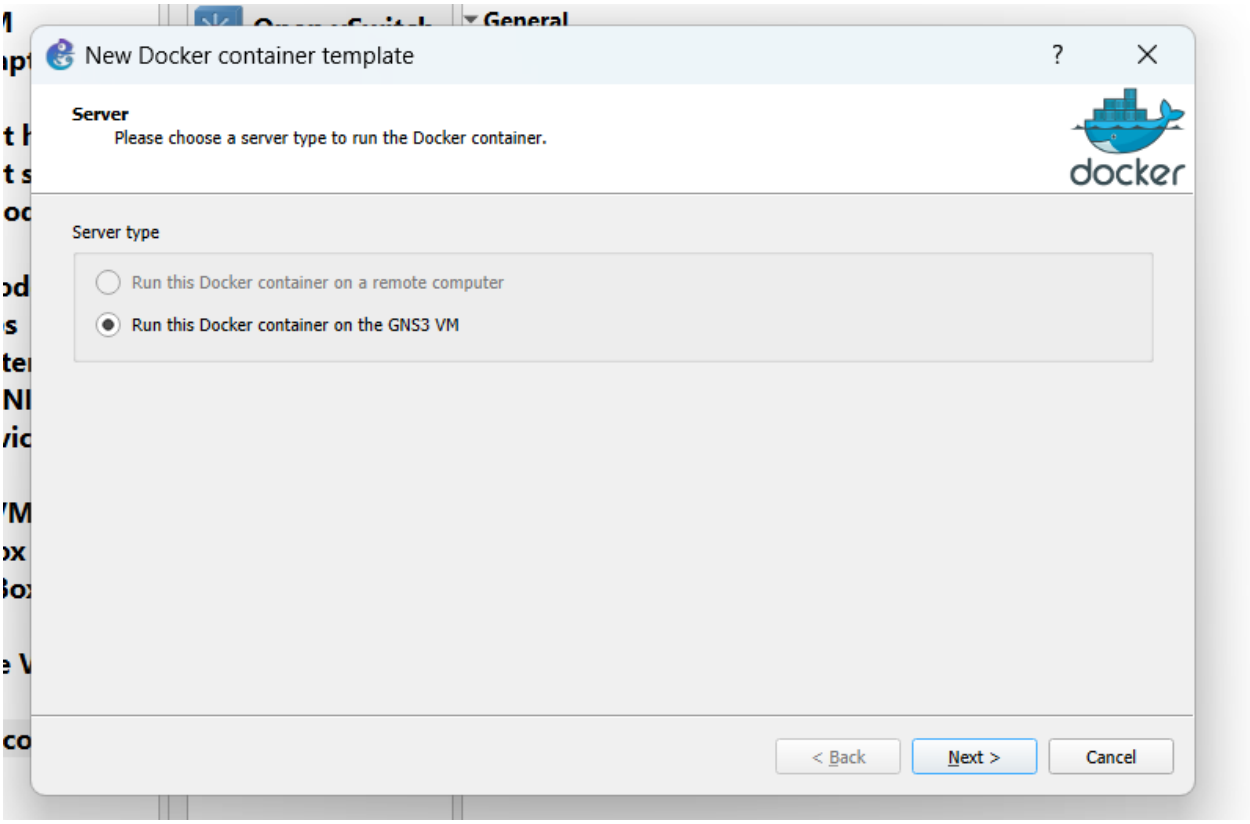


Go directly to docker and then click new.

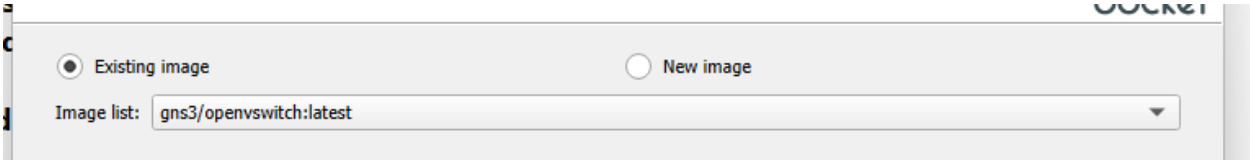




It will pop up a message to install doctor to vm.



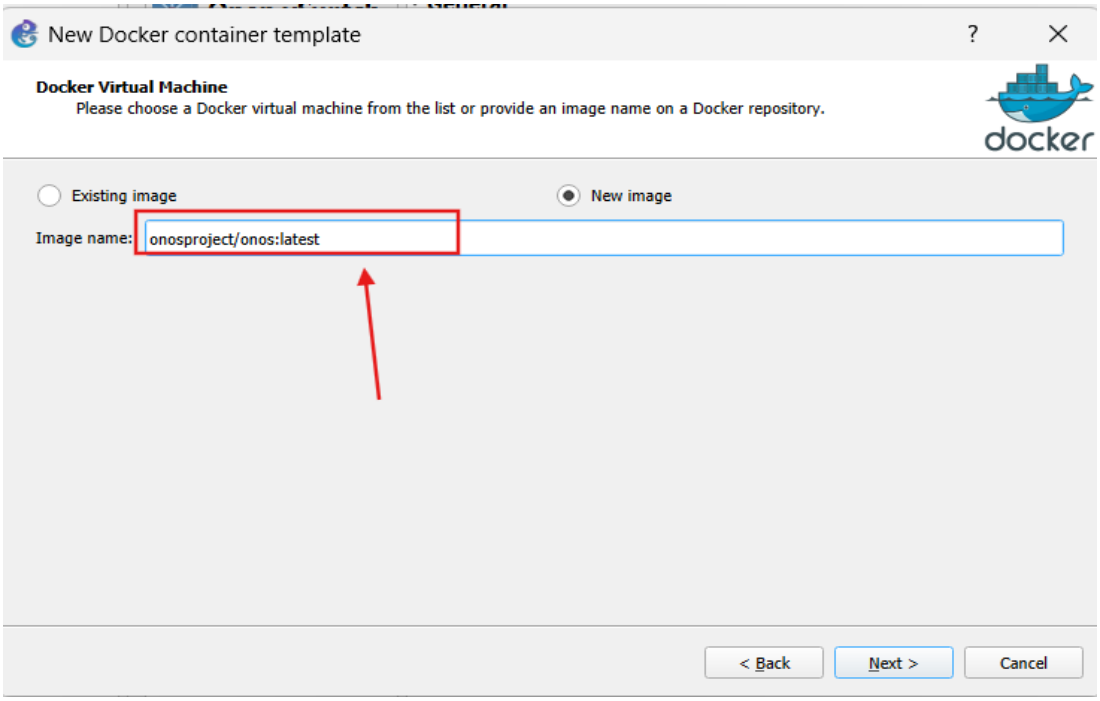
Click ”next” to continue. Then after you will see this

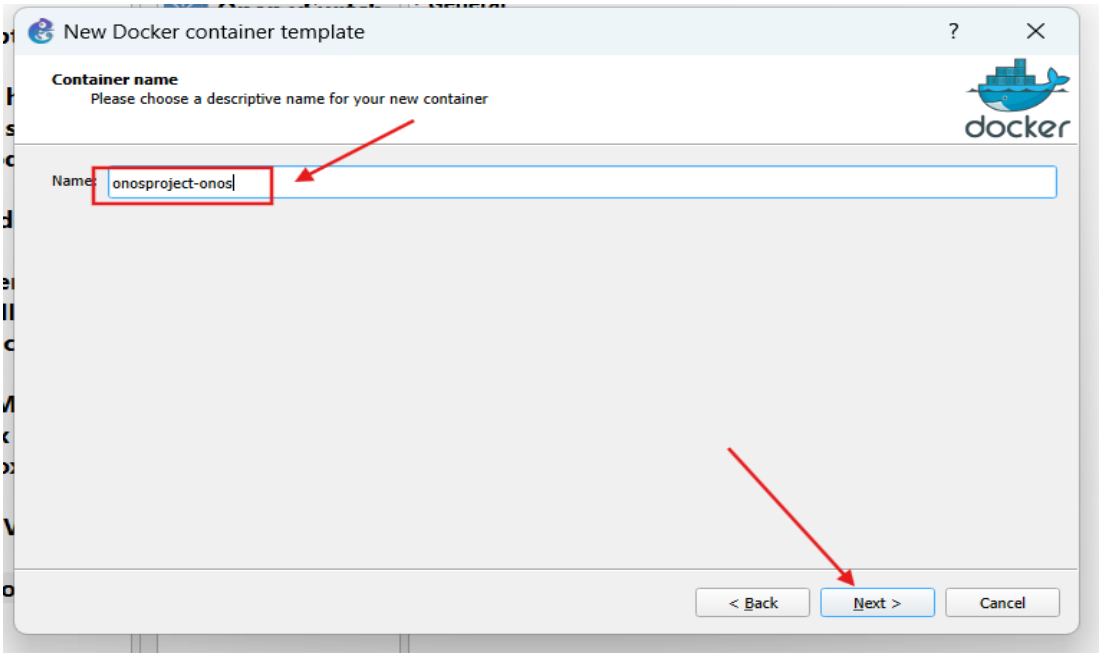


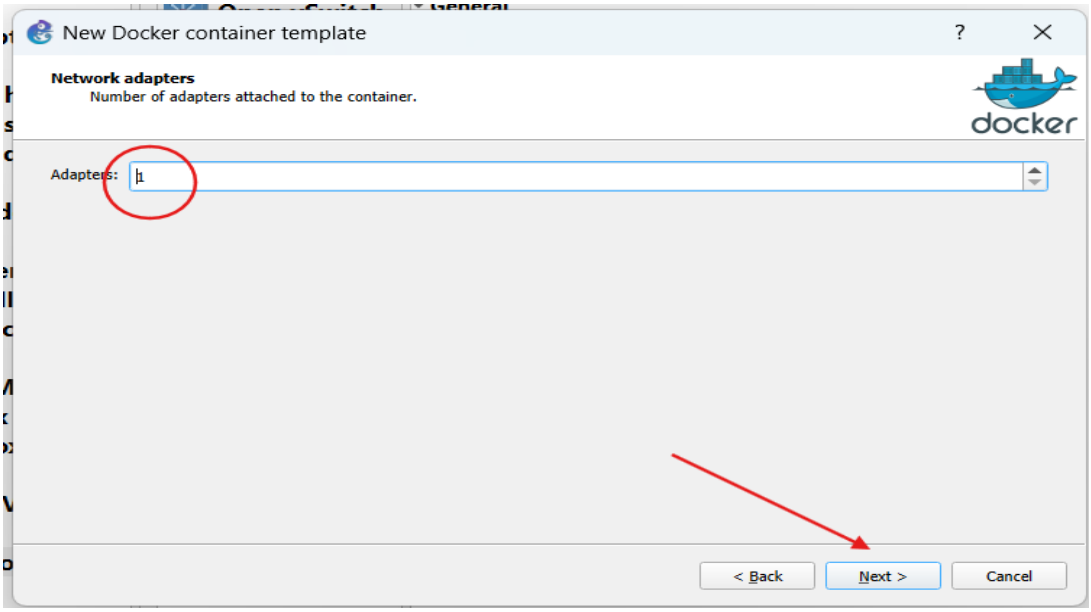
Click “new image”

3. Enter ONOS or OpenDaylight image name

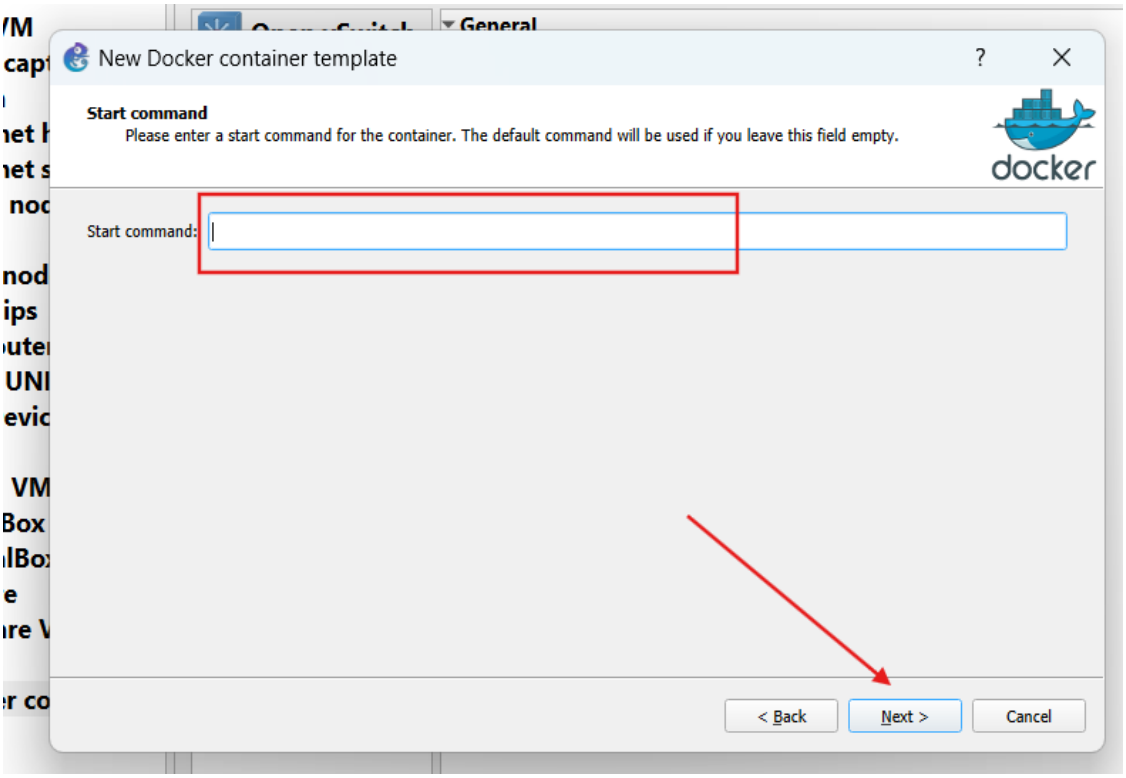
1. In the **Image name** field, type exactly:
    i. For **ONOS**: onosproject/onos:latest
    ii. For **OpenDaylight**: opendaylight/odl:latest


New Docker container template
Docker Virtual Machine
Please choose a Docker virtual machine from the list or provide an image name on a Docker repository.
docker
Existing image
New image
Image name:
onosproject/onos:latest
< Back
Next >
Cancel

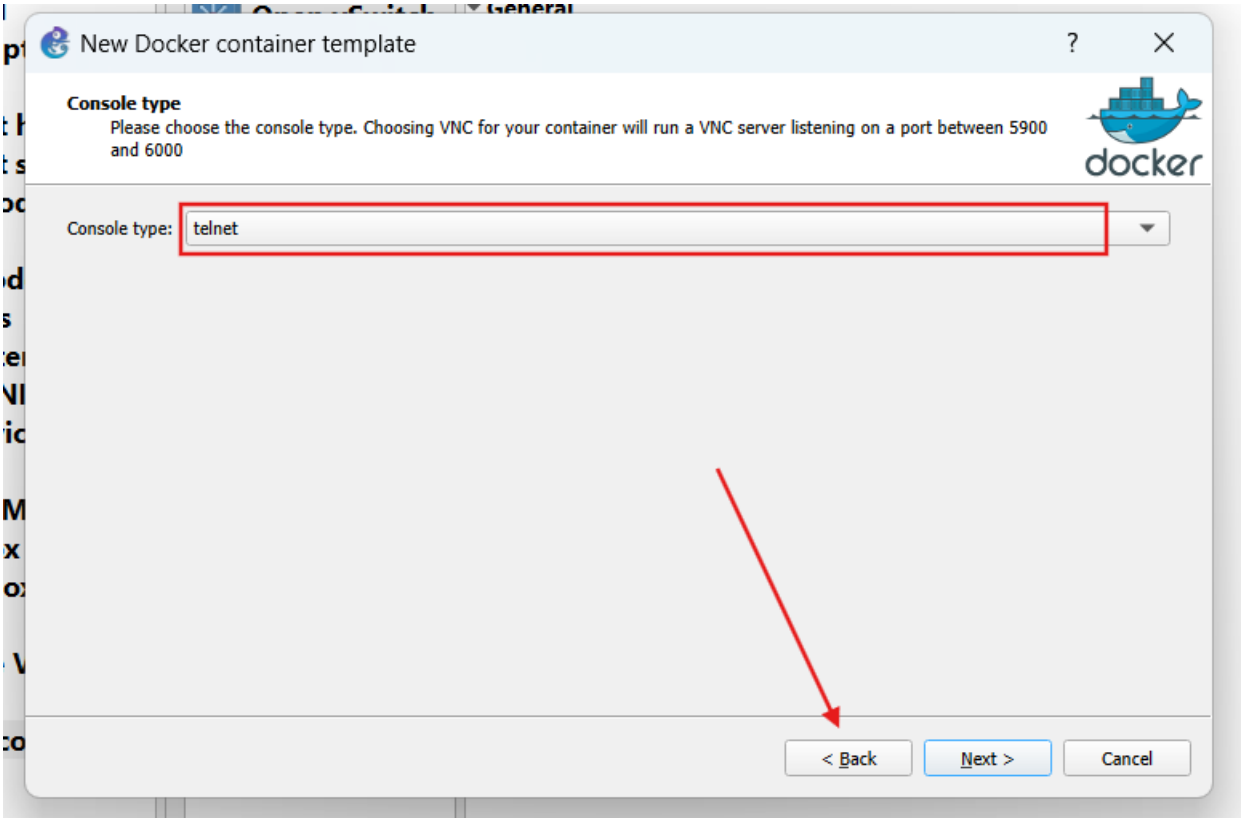

New Docker container template
Container name
Please choose a descriptive name for your new container
docker
Name:
onosproject-onos
< Back
Next >
Cancel

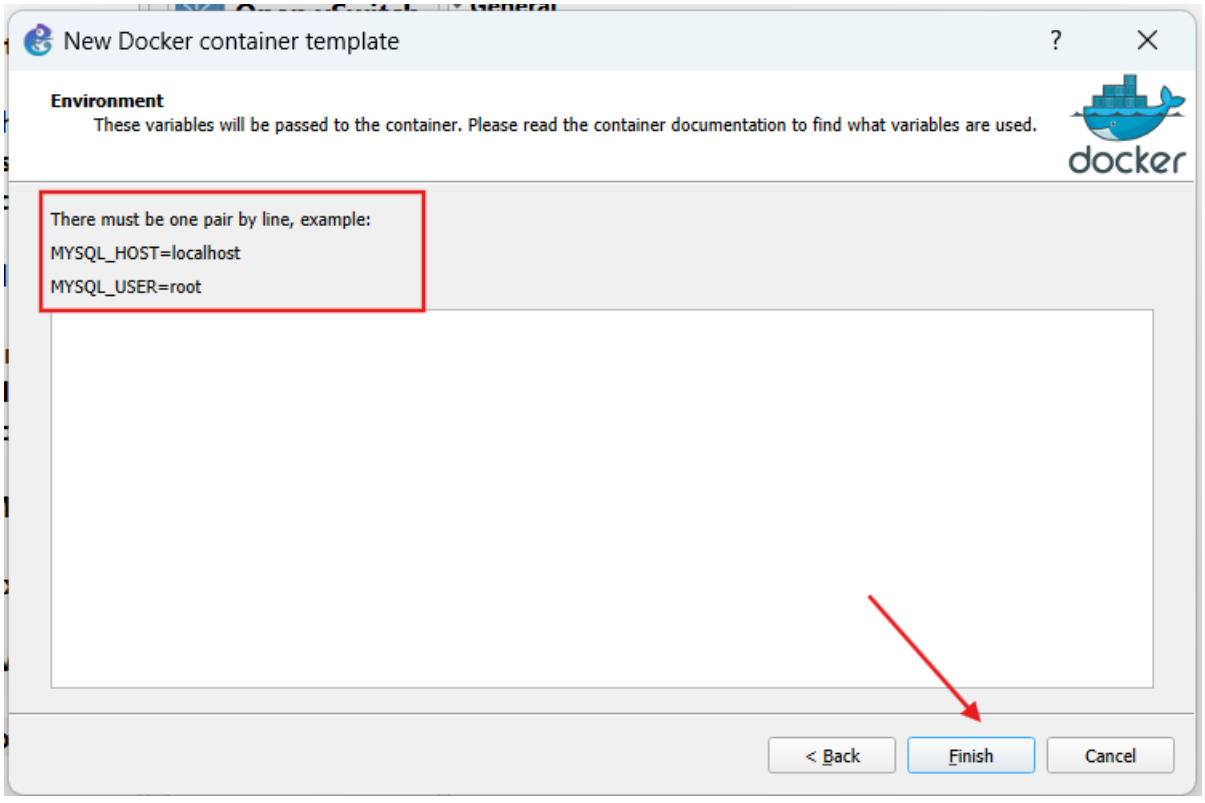


Leave empty to the start command.

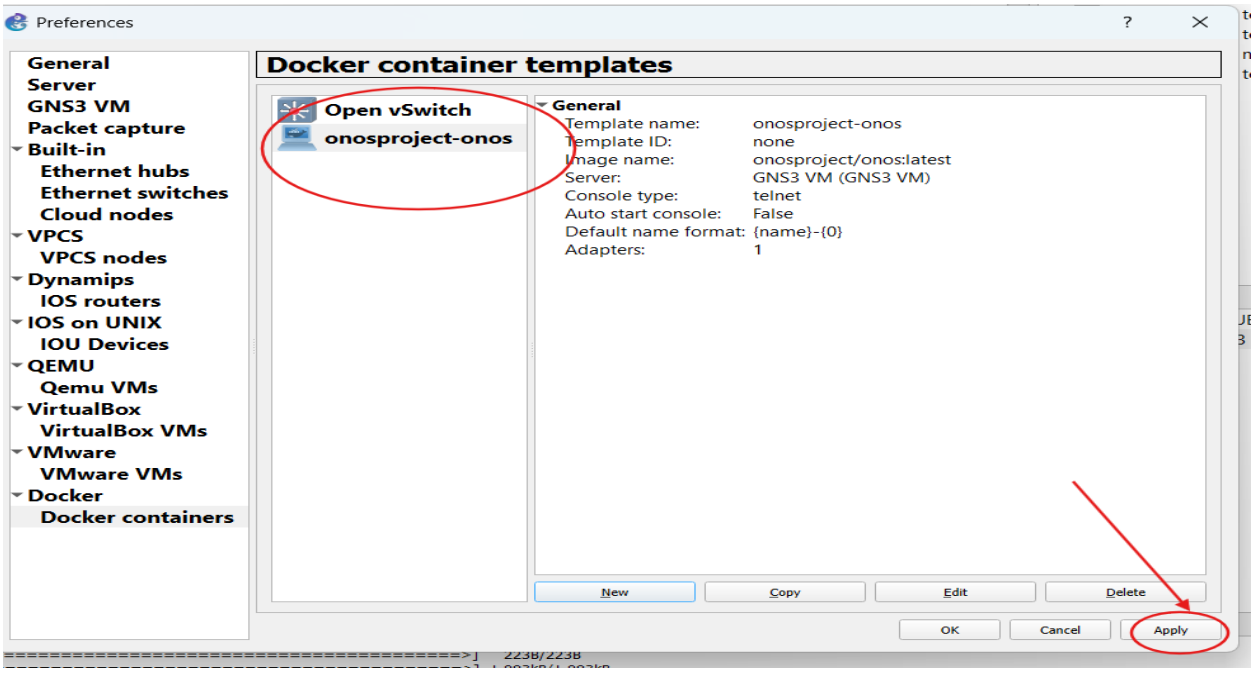


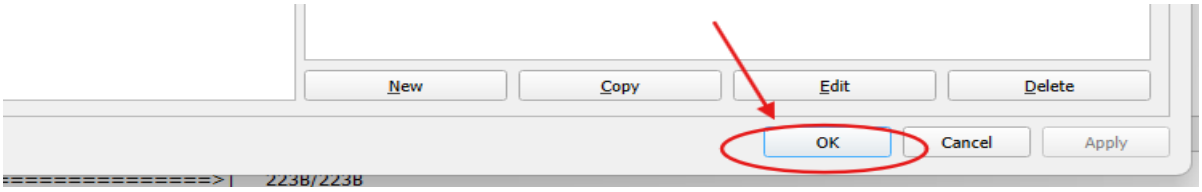
Choose the telnet as the console type.



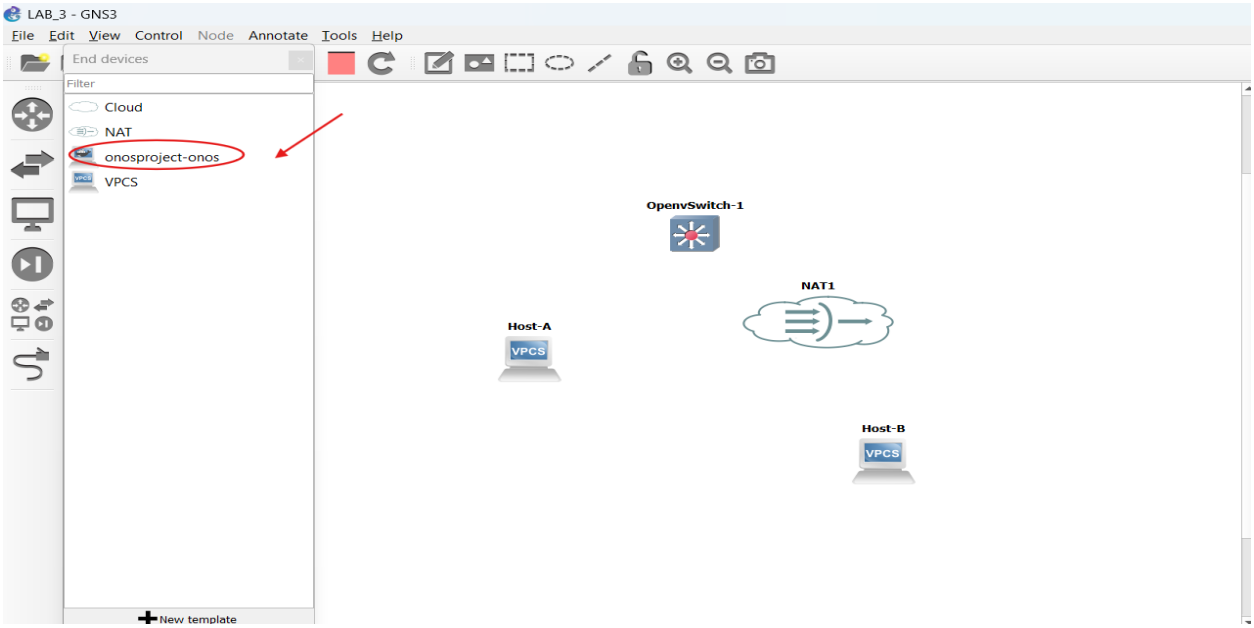


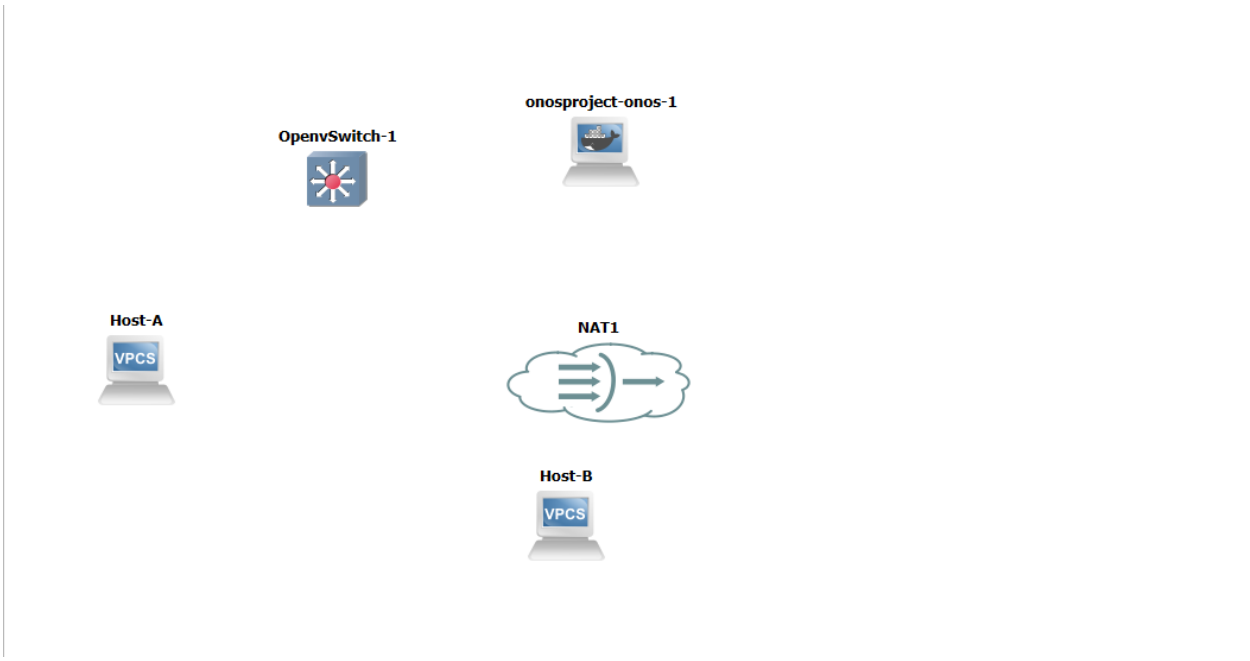
Leave the empty then click “finish”



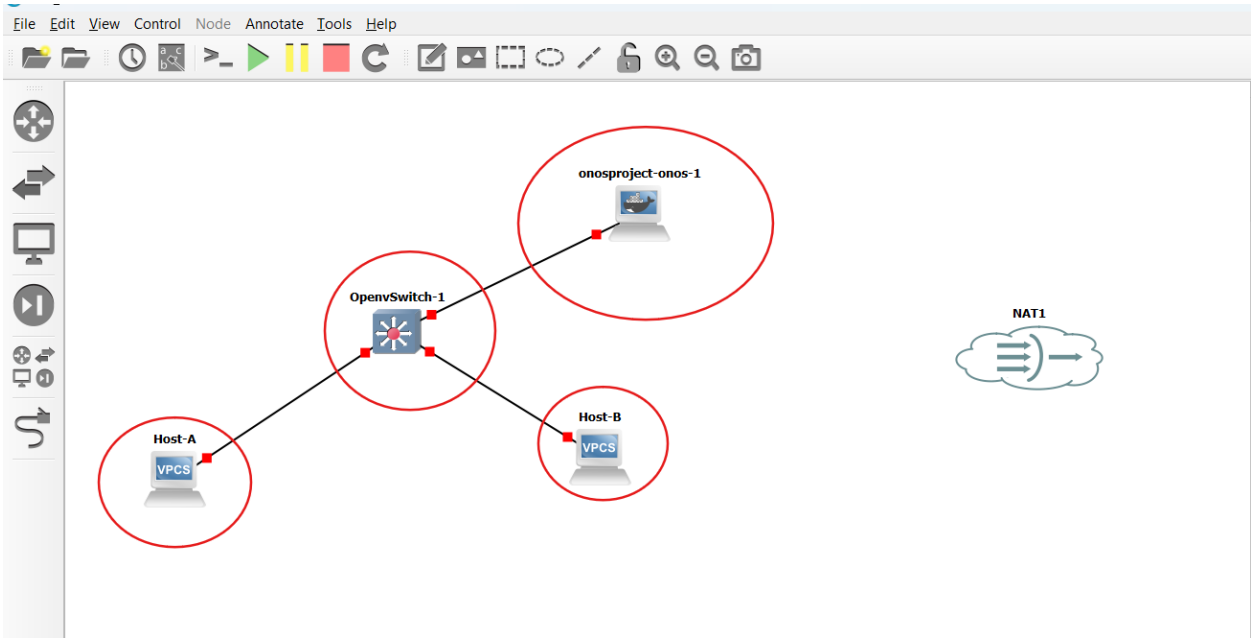


Finally, the ono it has seen, drag and drop to our project. It will be pulled from docker {make sure you have stable internet.

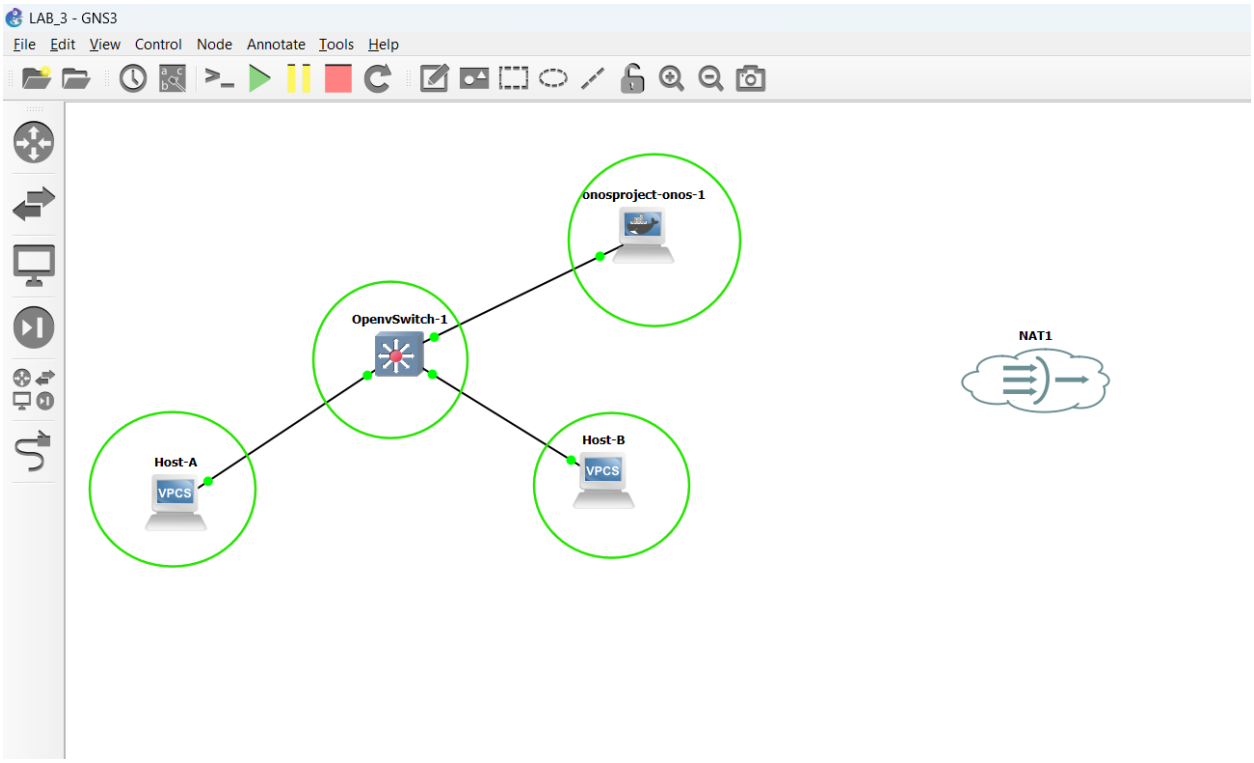




2. Connect:
    i. Controller Eth0 → OVS (any free port eth0 for management)
    ii. Host A Eth0 → OVS eth0
    iii. Host B Eth0 → OVS eth1



3. Power on all nodes.

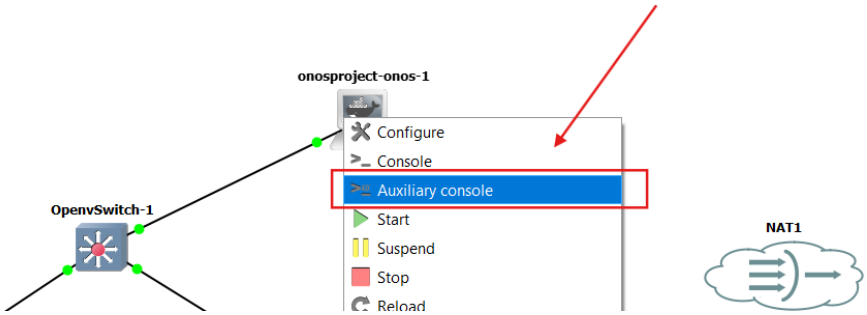


The topology consisted of an SDN controller, an Open vSwitch appliance, and two VPCS hosts. The controller was connected to OVS via a dedicated management port, establishing the **control plane link**. Host A and Host B were connected to OVS ports eth0 and eth1 respectively.

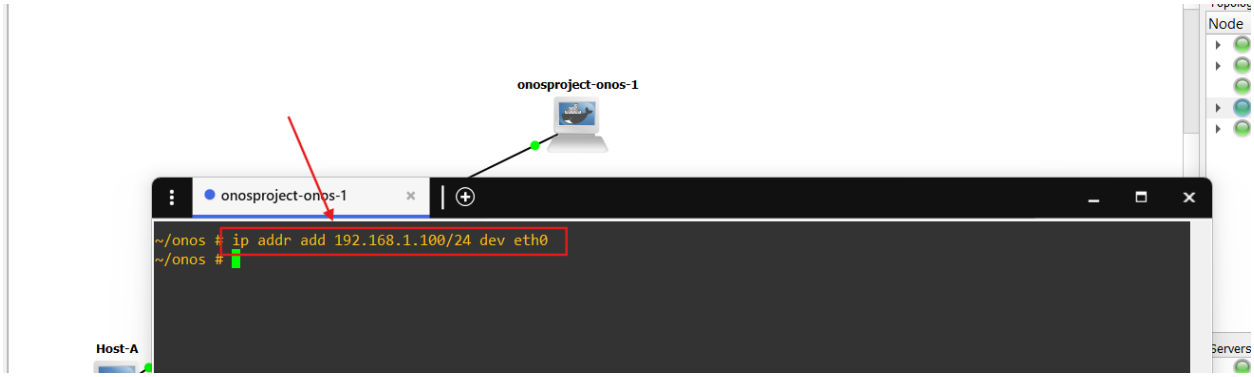
Step 4: Deploying the SDN Controller

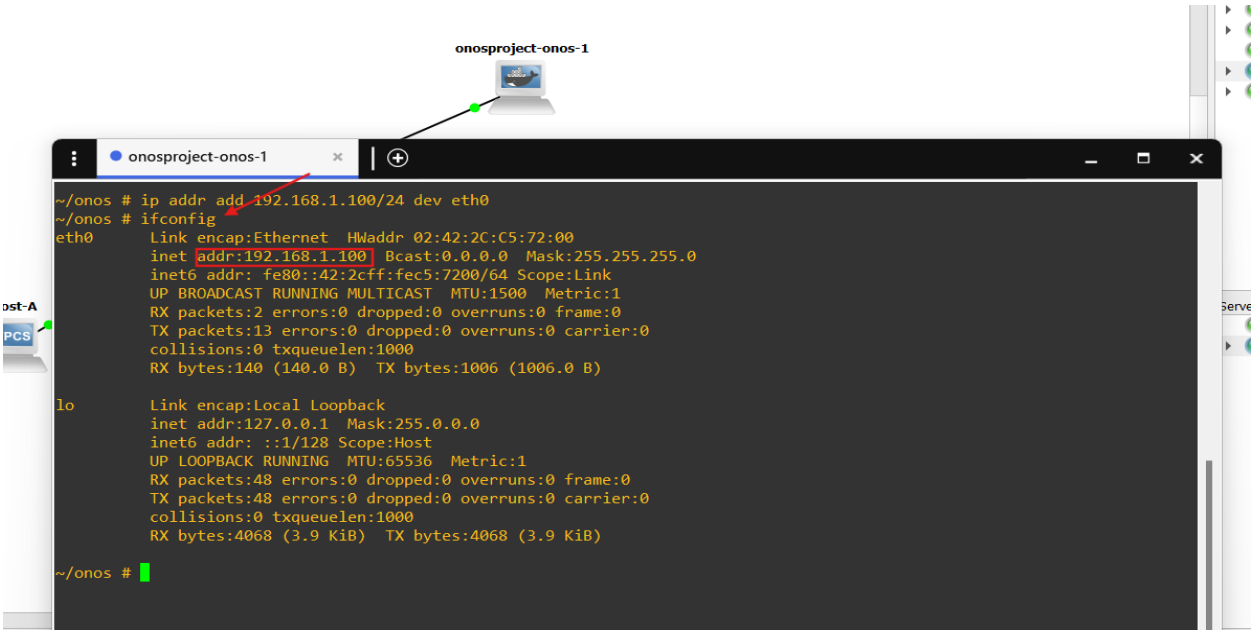
Perform:

1. Import the ONOS appliance if not already available.
2. Start the controller node.
3. Note the controller's IP address (check via ifconfig or GNS3 node status).



Right click the controller and click the console option and assign Static IP to ONOS.



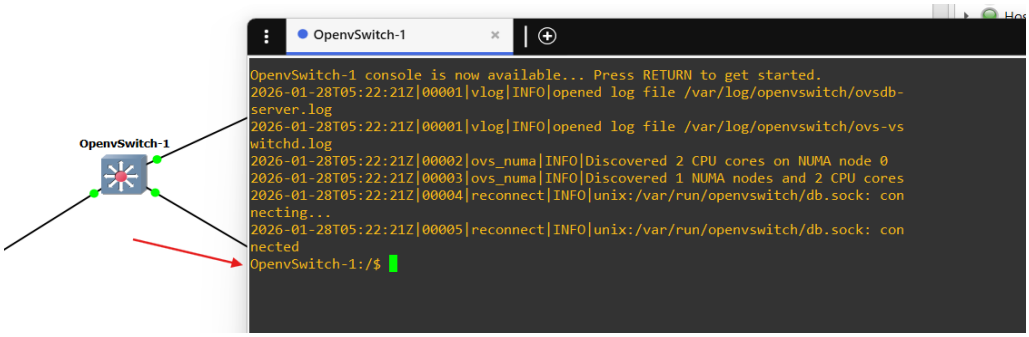


The SDN controller (ONOS) was deployed as a GNS3 appliance and powered on. Its management IP was not identified , I have setted manually to obtain ip 192.168.1.100

Step 5: Configuring OVS to Connect to the Controller

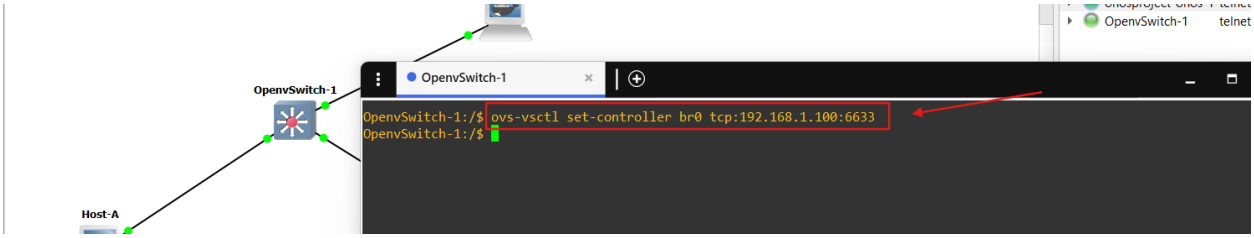
Perform:

1. Open OVS console.

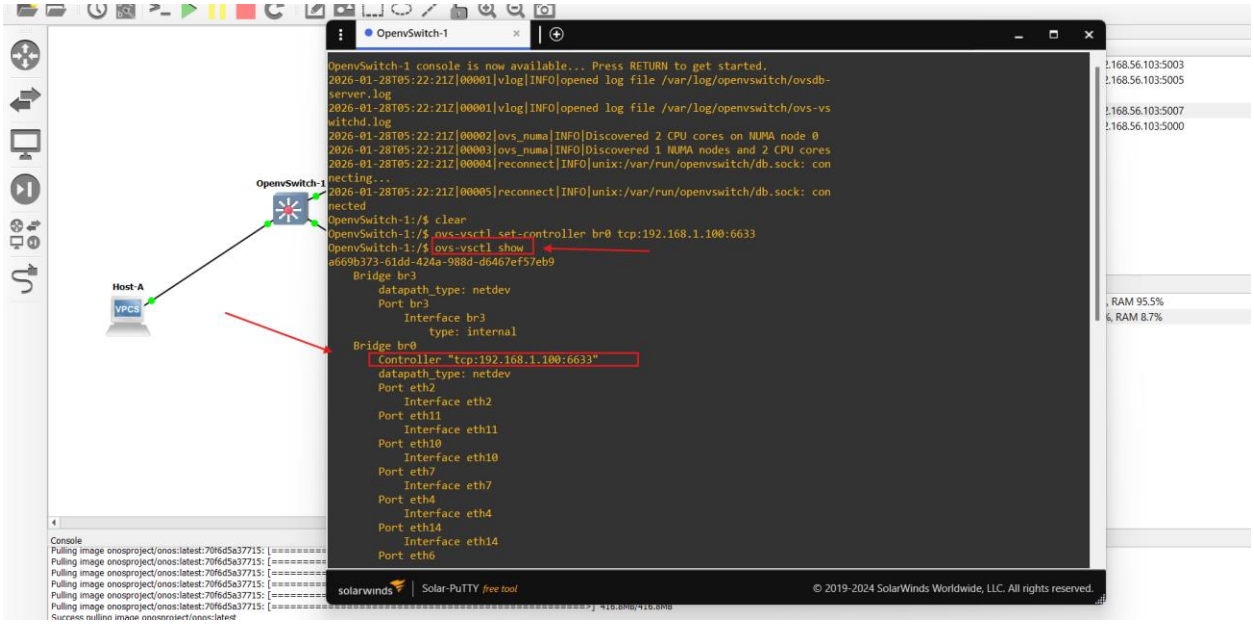


2. Set the controller IP (replace with actual IP)

ovs-vsctl set-controller br0 tcp:<CONTROLLER_IP>:6633



3. Verify: ovs-vsctl show



Look for is_connected: true under Controller.

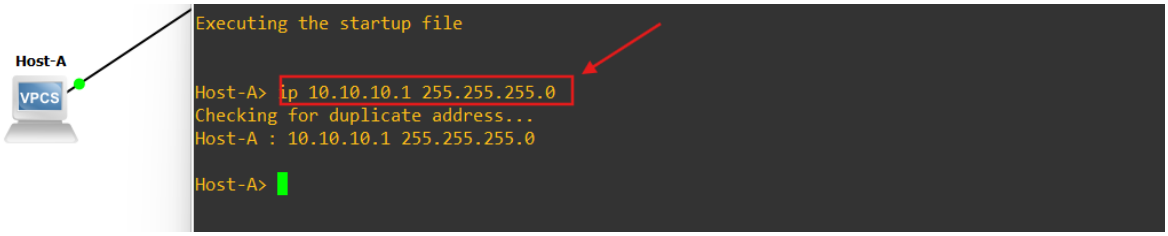
The OVS bridge br0 was configured to connect to the SDN controller using the command: “ovs-vsctl set-controller br0 tcp:192.168.1.100:6633” Verification via “ovs-vsctl show” confirmed the controller connection was established.

If you don’t ssee “is_connected”

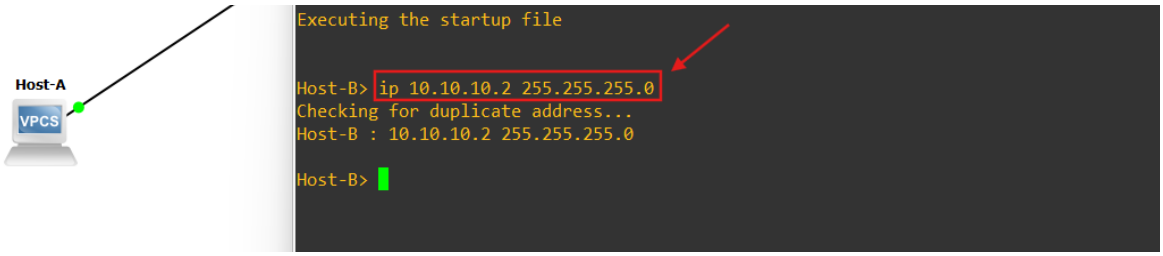
Step 6: Observing the "Silent" Network

Perform:

1. Assign IPs:
    i. Host A: 10.0.0.1/24



    ii. Host B: 10.0.0.2/24



2. Try pinging from Host A to Host B.
3. It will **fail** initially.

After assigning IP addresses to Host A (10.0.0.1) and Host B (10.0.0.2), a ping from Host A to Host B **failed**. This occurred because OVS, now under controller control, had no flow rules installed and was waiting for the SDN controller to dictate forwarding behavior.

Step 7: Activating Reactive Forwarding

Perform:

1. Access the controller's web UI (ONOS: http://<CONTROLLER_IP>:8181).
2. Log in (default credentials often onos/rocks).
3. Activate the **Reactive Forwarding** app.
4. Retry ping from Host A to Host B — it should now succeed.

The SDN controller's web interface was accessed, and the **Reactive Forwarding** application was enabled. Upon retrying the ping from Host A to Host B, **ICMP replies were received successfully**. The controller dynamically installed flow rules in OVS upon detecting the first packet of the flow, demonstrating **reactive flow programming**.

Step 8: Technical Analysis

Write in report:

This lab clearly demonstrated the **separation of control and data planes**:

1. **Control Plane:** SDN controller (ONOS/OpenDaylight) makes forwarding decisions.
2. **Data Plane:** OVS executes flow rules pushed by the controller.

Unlike Lab 2 where OVS acted as a standalone learning switch, here forwarding logic is centralized and programmable. This aligns with the ETSI NFV framework where the **SDN controller** (part of the management and orchestration layer) interacts with the **VIM** to manage virtualized network resources.

Step 9: Answer Assessment Questions

Question 1: Describe the difference in traffic handling between Lab 2 (standalone) and Lab 3 (controller-led).
Answer: In Lab 2, OVS operated as a **learning switch**, building a MAC table independently (data plane only). In Lab 3, OVS relied on the **SDN controller** to install flow rules via OpenFlow (control plane driven). Without controller rules, OVS could not forward packets.

Question 2: How does the "Inventory & State Database" relate to the controller's ability to manage the OVS bridge?
Answer: The controller's **Inventory & State Database** maintains a real-time view of network topology, switch status, port states, and installed flows. This allows the controller to make informed decisions, push appropriate rules to OVS, and adapt to network changes dynamically.

Step 10: Conclusion

Lab 3 successfully demonstrated the integration of an SDN controller with Open vSwitch, highlighting the core SDN principle of control-data plane separation. The use of reactive forwarding illustrated how centralized intelligence can dynamically manage network behavior, enabling flexible, programmable, and automated network operations—key to modern NFV/SDN deployments.