

OpenStack 2025.1 (Epoxy) GPU/PCI Passthrough 설치 가이드

0. 사전 환경 가정

- **OS:** Ubuntu 24.04 LTS (모든 노드 동일)
- **Control Node IP:** 10.10.10.60
- **Compute Node IP:** 10.10.10.62 (GPU 장착)
- **Target GPU:** NVIDIA (Vendor: 10de, Product: 20b7)

1. [모든 노드] 공통 기본 설정

대상: Control Node(Master), Compute Node 등 모든 서버 **권한:** Root (`sudo -i`)

```
# 1. Root 권한 획득
sudo -i

# 2. 한국 표준시 설정
timedatectl set-timezone Asia/Seoul

# 3. 패키지 업데이트 및 필수 유틸리티 설치
apt update && apt install -y git python3-dev libffi-dev gcc libssl-dev python3-venv net-tools

# 4. 호스트 등록 (DNS가 없을 경우 /etc/hosts 등록 필수)
# (아래 IP와 호스트명은 본인 환경에 맞게 수정하세요)
echo "10.10.10.60 control" >> /etc/hosts
echo "10.10.10.62 compute" >> /etc/hosts
```

2. [Compute 노드] GPU 격리 및 하드웨어 설정

대상: GPU가 장착된 Compute Node (10.10.10.62) **목적:** 커널 레벨에서 GPU를 격리하여 VM에 할당할 준비를 합니다.

```
# 1. IOMMU 활성화 (GRUB 설정 수정)
# 기존 설정을 백업 후 intel_iommu=on iommu=pt 추가 (AMD CPU인 경우 intel_iommu 대신 amd_iommu 사용)
cp /etc/default/grub /etc/default/grub.bak
sed -i 's/GRUB_CMDLINE_LINUX_DEFAULT="/GRUB_CMDLINE_LINUX_DEFAULT="intel_iommu=on iommu=pt /' /etc/default/grub
update-grub

# 2. VFIO 모듈 로드 설정
cat <<EOF > /etc/modules-load.d/vfio.conf
```

```

vfio
vfio_iommu_type1
vfio_pci
EOF

# 3. GPU 장치 바인딩 (vfio-pci)
# [주의] 10de:20b7 부분은 'lspci -nn | grep NVIDIA'로 확인된 본인의 ID로 반드시 변경
# 하세요.
echo "options vfio-pci ids=10de:20b7" > /etc/modprobe.d/vfio.conf

# 4. 설정 적용을 위해 initramfs 갱신 및 재부팅
update-initramfs -u
reboot

# --- 재부팅 후 확인 명령어 ---
# lspci -nnk -d 10de:20b7
# 결과에 "Kernel driver in use: vfio-pci"가 출력되어야 성공

```

3. [Control 노드] Kolla-Ansible 설치 및 준비

대상: Control Node (10.10.10.60) **목적:** 배포 도구 설치 및 기본 설정

```

# 1. Python 가상환경 생성 (시스템 패키지 충돌 방지)
python3 -m venv $HOME/kolla-venv
source $HOME/kolla-venv/bin/activate

# 2. pip 및 Ansible 설치
pip install -U pip
pip install 'ansible-core>=2.16,<2.18'

# 3. Kolla-Ansible 2025.1 설치
pip install git+https://opendev.org/openstack/kolla-ansible@stable/2025.1

# 4. 설정 디렉토리 생성 및 예제 파일 복사
sudo mkdir -p /etc/kolla
sudo chown $USER:$USER /etc/kolla
cp -r $HOME/kolla-venv/share/kolla-ansible/etc_examples/kolla/* /etc/kolla/
cp $HOME/kolla-venv/share/kolla-ansible/ansible/inventory/* .

# 5. 의존성 설치 및 패스워드 생성
kolla-ansible install-deps
kolla-genpwd

# 6. SSH 키 배포 (Control -> Compute 접속용)
# (이미 되어 있다면 생략 가능)
ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
ssh-copy-id root@10.10.10.62

# 7. 인벤토리 파일 수정
# vi multinode 명령으로 아래 내용 반영

```

```
# [control]
# 10.10.10.60
# [compute]
# 10.10.10.62
```

4. [Control 노드] Placement & PCI 연동 설정 (★중요★)

대상: Control Node (10.10.10.60) **설명:** 배포 시 적용될 **Nova** 설정 파일을 미리 작성합니다.

```
# 설정 파일 저장 경로 생성
mkdir -p /etc/kolla/config/nova

# -----
# 4-1. Nova Compute 설정 (Compute 노드에 적용될 설정)
# -----
cat <<EOF > /etc/kolla/config/nova/nova-compute.conf
[pci]
# Placement API에 PCI 장치를 보고하도록 설정 (최신 표준)
report_in_placement = True

# PCI 장치 정의 (device_spec)
# resource_class는 반드시 'CUSTOM_'으로 시작하고 대문자여야 함
device_spec = { "vendor_id": "10de", "product_id": "20b7", "resource_class":
"CUSTOM_GPU_NVIDIA", "name": "nvidia-gpu" }
EOF

# -----
# 4-2. Nova API 설정 (API 서비스가 알게 될 별칭)
# -----
cat <<EOF > /etc/kolla/config/nova/nova-api.conf
[pci]
# Flavor 생성 시 사용할 별칭 정의
alias = { "name": "gpu-alias", "device_type": "type-PF", "custom_resource_class":
"CUSTOM_GPU_NVIDIA" }
EOF

# -----
# 4-3. Nova Scheduler 설정
# -----
cat <<EOF > /etc/kolla/config/nova/nova-scheduler.conf
[filter_scheduler]
# PciPassthroughFilter 활성화
enabled_filters = PciPassthroughFilter, ComputeFilter, AvailabilityZoneFilter,
ImagePropertiesFilter, ValidationFilter
EOF
```

5. [Control 노드] 배포 실행

대상: Control Node (10.10.10.60)

```
# 1. 가상환경 활성화 (필수)
source $HOME/kolla-venv/bin/activate

# 2. 연결 테스트
ansible -i multinode all -m ping

# 3. OpenStack 배포 (순서대로 실행)
kolla-ansible -i multinode bootstrap-servers
kolla-ansible -i multinode prechecks
kolla-ansible -i multinode deploy

# 4. 클라이언트 도구 설치 (배포 완료 후)
pip install python-openstackclient
```

6. [Control 노드] 검증 및 Flavor 생성

대상: Control Node (10.10.10.60) **설명:** 배포 후 GPU 자원이 정상적으로 Placement에 등록되었는지 확인하고 Flavor를 생성합니다.

```
# 1. Admin 환경 변수 로드
source /etc/kolla/admin-openrc.sh

# 2. Placement 자원 확인 (가장 중요한 검증 단계)
# Compute 노드가 GPU를 'CUSTOM_GPU_NVIDIA'라는 자원으로 보고했는지 확인
# 아래 명령 결과에서 USAGE / CAPACITY 확인
openstack resource provider list
openstack resource provider inventory list <COMPUTE_NODE_UUID>

# 3. GPU Flavor 생성 (Placement 문법 사용)
# --property resources:자원클래스명=개수
openstack flavor create --vcpus 8 --ram 16384 --disk 100 gpu.flavor
openstack flavor set --property "resources:CUSTOM_GPU_NVIDIA=1" gpu.flavor

# 4. 인스턴스 생성 테스트
openstack server create --flavor gpu.flavor --image <IMAGE_NAME> --network
<NETWORK_NAME> gpu-test-vm
```