WSL에서 GPU 오케스트레이션

HashiCorp Nomad

이규석

안녕하세요. 이규석 입니다.

揆石

揆: 헤아릴 규
石: 돌 석

GS

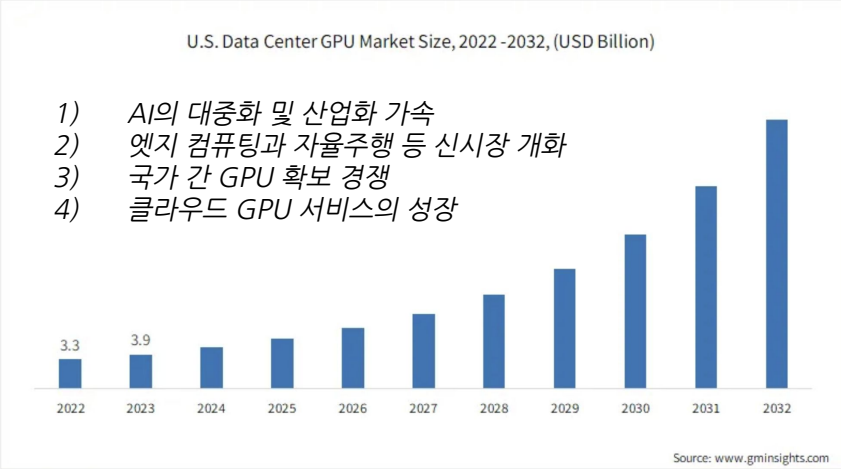
Sr. Solutions Engineer, HashiCorp an IBM Company

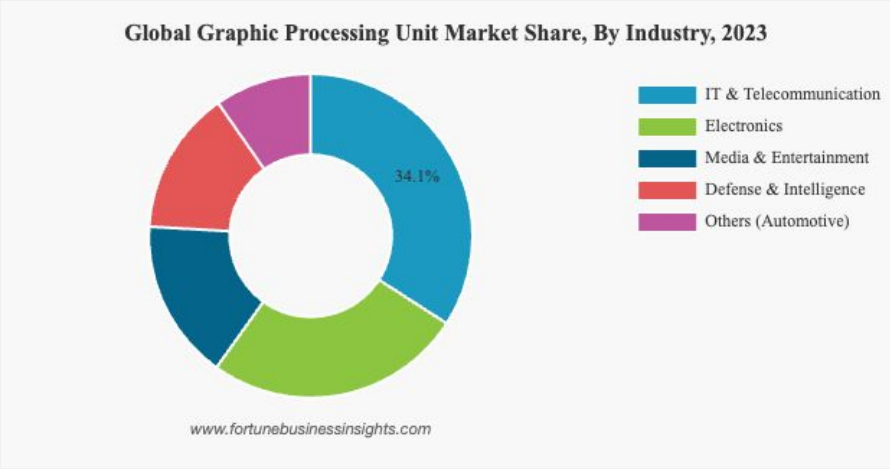
great-stone

terraform101

great-stone

GPU 성장세





GPU 투자는 AI 산업 성장, 데이터 센터 확대, 엣지 디바이스 증가, 그리고 국가적 기술 경쟁에 의해 계속 늘어날 것으로 보입니다.

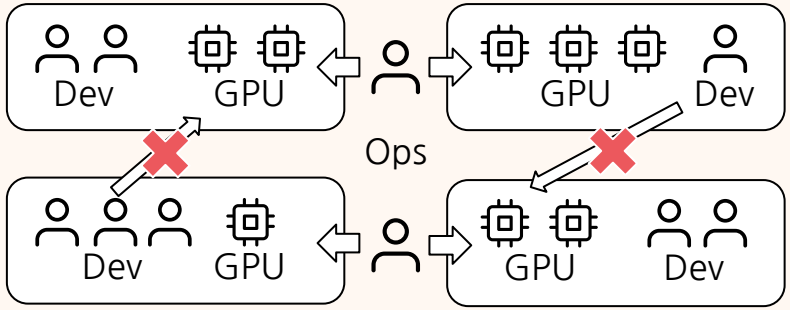
특히 AI 대중화가 GPU 시장 성장의 가장 강력한 엔진이며, GPU는 **"미래의 연료"**로 여겨지고 있어 장기적으로도 시장이 커질 가능성이 높습니다.

- NTT DOCOMO와 NVIDIA는 일본 네트워크 전반에 GPU 강화 무선 솔루션을 통합했습니다.
- 인공 지능(AI), 빅 데이터 및 IoT 메커니즘의 잠재력을 연결할 수 있는 기능을 갖춘 GPU 기반 컴퓨팅 솔루션은 현대화된 군사 및 항공 우주 애플리케이션에서 점점 더 중요한 역할을 합니다.

https://www.fortunebusinessinsights.com/graphic-processing-unit-gpu-market-109831

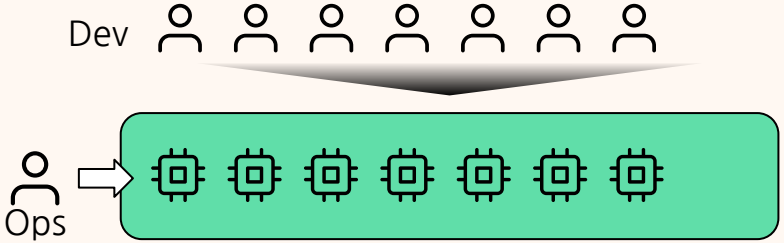
GPU 성장세 + 과제

Before



- 여러 AI 아이디어를 위한 무분별한 고가의 장비 투자
- 효율적이지 못한 GPU 활용 방식의 부재
- 팀별, 조직별 GPU 장비 관리
- 잦은 Fail에 대한 비효율적인 수동 작업

To-Be



- GPU Farm을 통한 통합 관리 방식
- 자동화된 Job 관리
- 소수점 단위의 GPU 코어 제공
- 서비스 개발, 데이터 분석 등에 집중
- 비용 절감!

쉽고 운영이 간단한 통합 GPU 환경?

GPU 스케줄링이란?

GPU는 인공 지능(AI) 및 머신 러닝(ML)을 포함한 광범위한 워크로드를 가속화하는 데 사용됩니다. 결과를 빠르게 제공하기 위해 AI 및 ML 워크로드는 다중 GPU 클러스터를 포함하여 엄청난 양의 리소스를 소비합니다 . GPU 스케줄링은 AL 및 ML 워크로드를 많은 수의 GPU에 분산하고 리소스를 효과적으로 활용하는 데 도움이 됩니다. 이는 일반적으로 스케줄러를 사용하여 달성됩니다.

전통적으로 작업 스케줄링은 Slurm이나 IBM LSF와 같은 전담 스케줄러에 의해 이루어졌습니다. 이러한 도구의 복잡성으로 인해 많은 조직이 Kubernetes나 **Nomad와 같은 컨테이너 오케스트레이터로 전환**하고 있습니다. 하지만 복잡성은 거기서 끝나지 않습니다. 컨테이너 오케스트레이터는 기본적으로 GPU 스케줄링을 지원하지 않습니다.

run.ai, NVIDIA

HashiCorp Nomad는 2019년부터 자체 GPU 플러그인을 제공하고 있습니다. 또한 컨테이너와 비 컨테이너 모두를 지원하여 개발자의 편의성을 높여줍니다.

Case Study:

“고객 워크로드에 대한 가용성이 향상되었습니다. 더욱 중요한 것은 Conductor와 고객 모두 렌더링 프로젝트 완료까지의 전체 시간을 단축하여 상당한 비용 절감 효과를 누리고 있다는 것입니다.”

더 빠른 처리량

Nomad를 사용하면 지연이 2~3초에 불과하며, 너무 빨라서 더 빨리 작동하도록 코드를 다시 작성

유휴 시간/유지보수 단축

Nomad를 사용하기 전에는 Kubernetes 클러스터 업그레이드와 관리가 운영상 힘든 작업이였지만 이제 쉽게 업그레이드 가능

유연한 작업 부하 지원

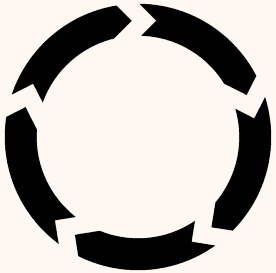
단일 플랫폼으로 레거시 앱, Windows 기반 앱, 컨테이너화된 앱, GPU기반 앱을 일괄 관리

클라우드에 무관

온프레미스 데이터 센터와 퍼블릭 클라우드에서 동일한 방식으로 배포되고 운영

Nomad 대 관리형 Kubernetes

관리형 Kubernetes(GKE)의 한계로 Google Cloud의 두 리전에서만 사용했지만, 지금은 소규모 팀으로도 다섯 지역에서 운영



GPU ROI Flywheel

https://www.linkedin.com/pulse/beyond-gpu-power-compute-roi-flywheel-jitendra-agarwal-a2ezc

GPU ROI(투자 수익률)

GPU 수익
- 더 나은 모델 품질
- 개발자 생산성 향상
- 전달/생성 속도

GPU 투자
- GPU 머신 추가
- 엔지니어의 실행 및 장비를 관리 시간
- 효율성을 위해 투자된 시간

$$GPU\ ROI = \frac{GPU\ machine\ Throughput\ X\ Developer\ productivity}{\$Cost}$$

- **GPU machine throughput**
  : GPU가 처리할 수 있는 작업량(예: 훈련 속도, 처리되는 데이터).
- **Developer productivity**
  : 엔지니어가 GPU를 사용하여 얼마나 빨리 작업을 완료할 수 있는가.
- **Cost**
  : GPU를 소유하고 운영하는 데 드는 총 비용입니다.

GPU ROI 예시

GPU Type	Throughput	Dev velocity	Cost Factor	Overall ROI
A100 40G	1x	1x	1x	1x
A100 80G	1.6x	1.28x	1.25x	1.6x
H100	3.x	2.4x	3x	**2.4x**

Example Table: GPU Performance vs. Cost

H100은 3배의 성능에 3배의 비용이 들기 때문에 비슷한 ROI를 제공하는 것처럼 보일 수 있습니다. 그러나 활용도를 최적화하면 이야기가 달라집니다. H100이 80% 활용도 로 작동하면 개발자 속도가 2.4배 증가 합니다 (3배 처리량의 80%). 생산성이 증가함에 따라 H100의 실제 ROI는 2.4배가 되어 기준선을 훨씬 넘어섭니다.

일반 열차에서 고속열차로 업그레이드하는 것과 같습니다. 고속열차는 승객을 A지점에서 B지점으로 3배 더 빨리 이동할 수 있을 뿐만 아니라 같은 시간에 더 많은 승객을 태우거나 더 먼 거리를 이동할 수 있어 훨씬 더 많은 가치를 제공합니다.

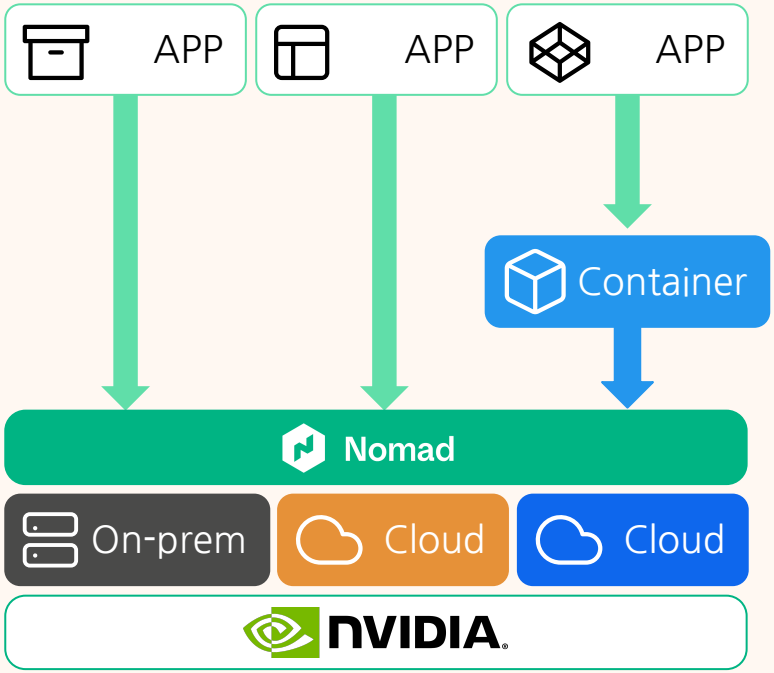
GPU 오케스트레이션 프랙티스 가이드

GPU 오케스트레이션, 쉽게 적용할 수 있어야 합니다

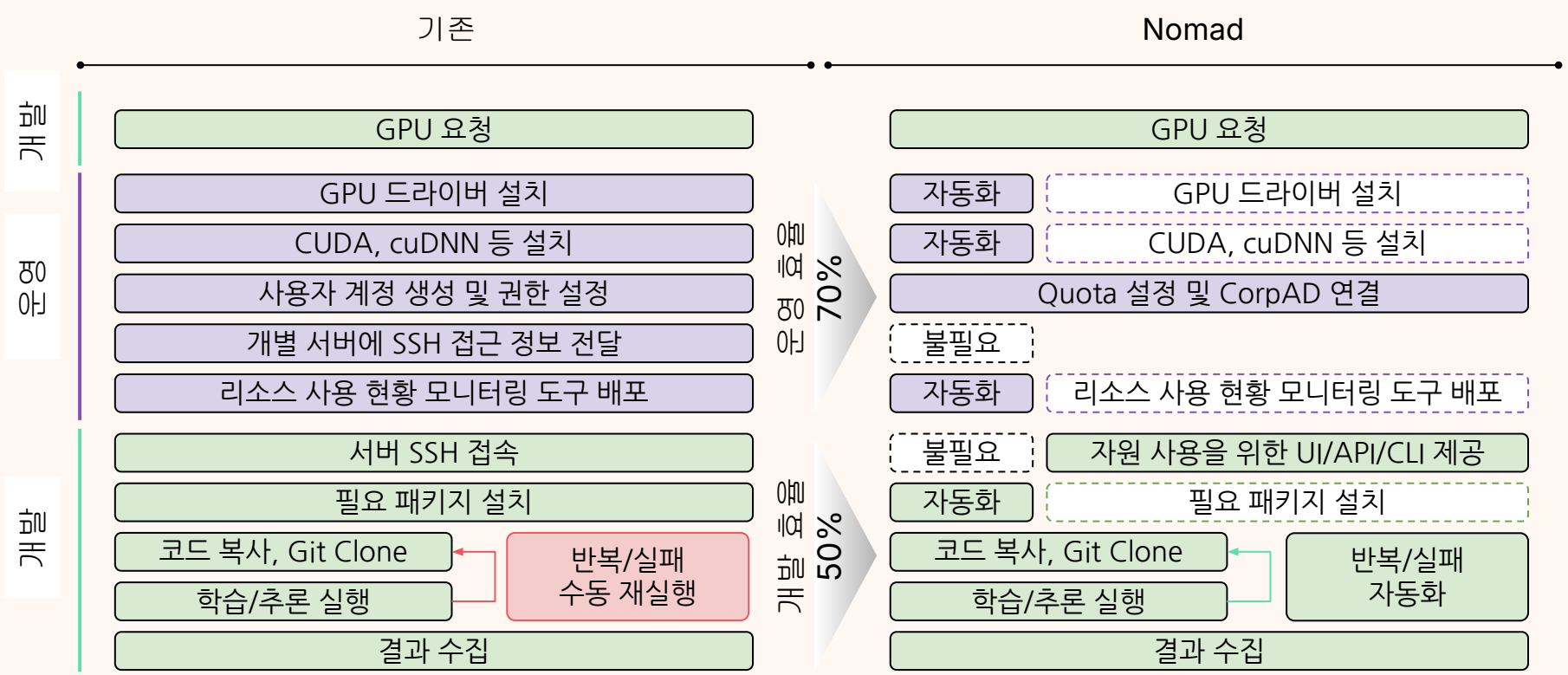
효율적인 GPU 자원 관리는 AI 워크로드 확장과 비용 절감, 그리고 성능 최적화의 핵심입니다. 특히, HashiCorp Nomad를 활용하면 복잡한 GPU 워크로드도 경량화된 방식으로 유연하게 배치하고 관리할 수 있습니다.

Nomad는 GPU워크로드의 이동성, 복구성, 확장성, 이식성, 배포 용의성을 패키징 방식에 제한하지 않습니다. 일반적인 장기 실행 앱(서비스), 일회성 또는 반복적인 배치, 컨테이너 등 다양 워크로드에 현대화된 실행 워크로드를 제공합니다.

Nomad는 실행을 위한 BM 또는 VM만 있으면 됩니다.
멀티/하이브리드 오케스트레이션 환경을 구축하는데는 몇 분, 몇 시간 내로 구성할 수 있습니다. 운영자는 Nomad 만을 위한 특별한 운영지식이 필요하지 않아 기존 지식을 활용하여 운영하면됩니다.



기존 방식과 개발 운영 방식 비교



익숙한 사용자 경험 제공

기존

- GPU 장비 접속
- AI/ML을 위한 패키지 준비 및 명령
- 학습 결과 확인
- 재학습 실행

Nomad

수행 내용

대상지정

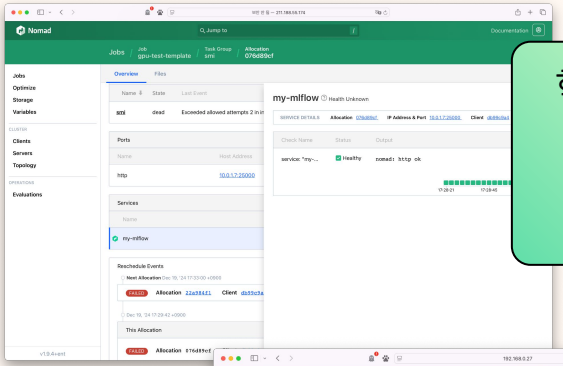
```
task "smi" {
  driver = "raw_exec"

  config {
    command = "local/start.sh"
  }

  template {
    data = <<-EOF
    #!/bin/bash
    pip install virtualenv
    virtualenv my_gpu_sandbox
    source my_gpu_sandbox/bin/activate
    pip install mlflow
    mlflow server --host 0.0.0.0 --port {{ env "NOMAD_PORT_http" }}
    EOF
    destination = "local/start.sh"
  }

  resources {
    device "nvidia/gpu" {
      constraint {
        attribute = "${device.model}"
        value     = "Tesla T4"
      }
    }
  }
}
```

익숙한 사용자 경험 제공

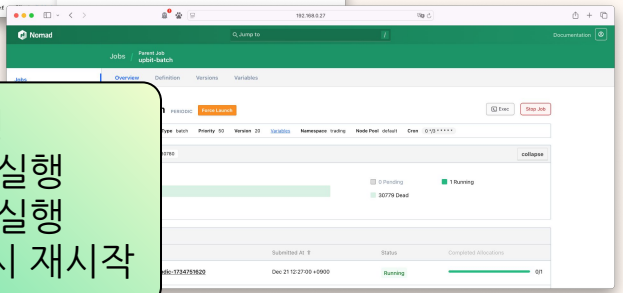
기존

- GPU 장비 접속
- AI/ML을 위한 패키지 준비 및 명령
- 학습 결과 확인
- 재학습 실행

Nomad



학습 결과 확인
- 파일
- 성공/실패
- 다운로드



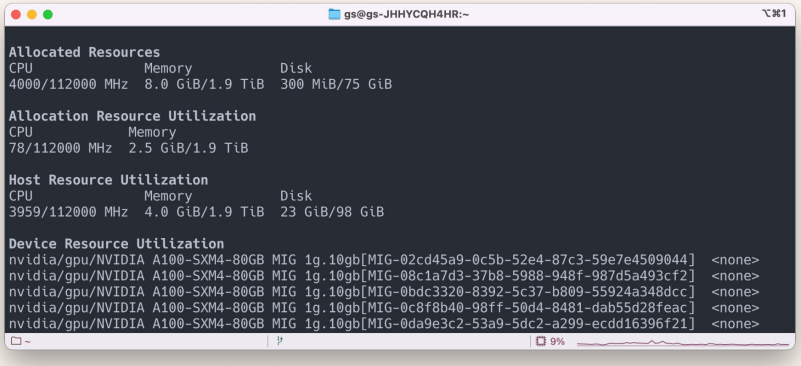
재학습 실행
- 반복 실행
- 강제 실행
- 실패시 재시작

MIG Multi Instance GPU 와 NUMA 지원

전체 GPU 메모리 공간이 필요하지 않은 소규모의 독립적인 추론 워크로드를 여러 개 실행하거나 여러 팀이 추론을 위해 GPU 리소스를 공유하는 경우, MIG 리소스 격리를 활용하여 더 큰 GPU 인스턴스를 분할하는 것이 좋습니다.

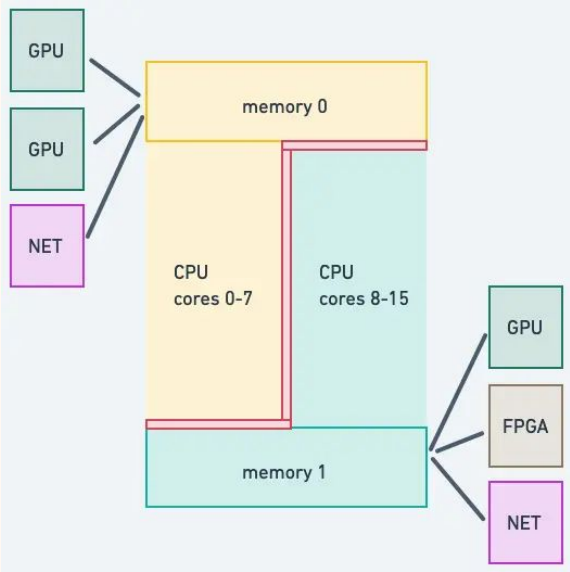
MIG 인스턴스로 파티셔닝된 대형 GPU를 보다 효율적으로 활용할 수 있습니다. 따라서 Nomad를 사용하여 GPU 워크로드를 실행하는 데 드는 전반적인 비용이 절감됩니다.

(지원 GPU : A30, A100, H100, H200)



높은 계산 처리를 위한 GPU 장치와 같은 장치가 있는 노드에서 워크로드를 예약할 수 있는 기능을 제공합니다.

스케줄링 구성을 통해 워크로드를 GPU 장치에 더 가깝게 배치하면 개발자는 더 높은 대역폭에서 GPU 장치와 CPU 노드 간에 데이터를 이동할 수 있습니다.



GPU Quota 제공

Quota(쿼터) 사양을 통해 운영자는 개별 사용자에게 네임스페이스 단위로 적용할 수 있는 GPU 리소스 제한을 지정할 수 있습니다.

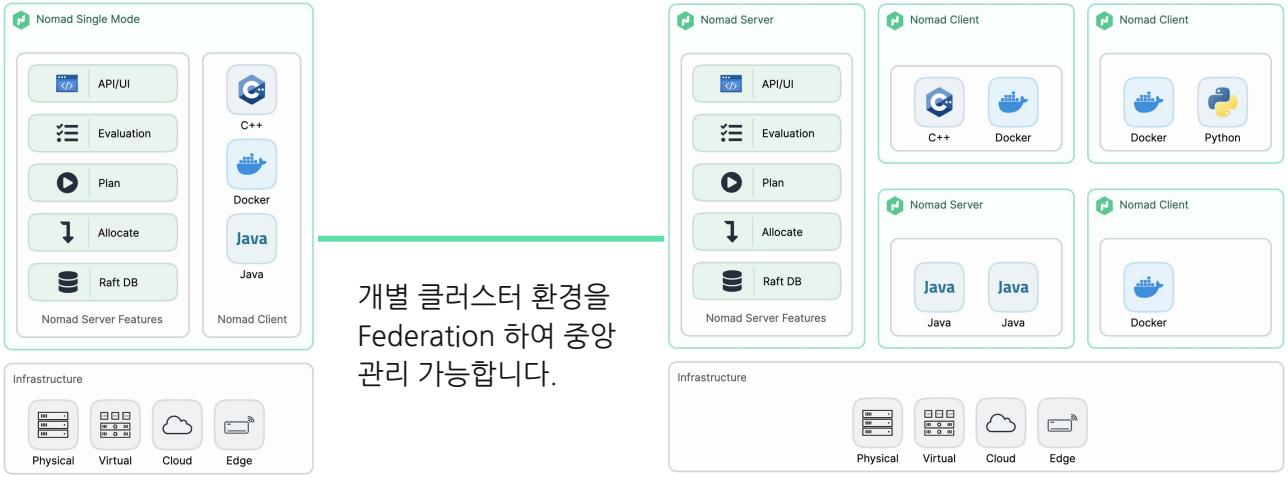
쿼터는 GPU 리소스와 같은 부족한 리소스를 효율적으로 관리할 수 있는 방법을 제공하며, 사용자가 소비할 수 있는 양을 제한하려는 운영자를 위한 거버넌스 계층을 제공합니다.

```
limit {
 region = "global"
 region_limit {
   cores       = 0
   cpu         = 2500
   memory      = 1000
   memory_max = 1000
   device "nvidia/gpu/h100" {
     count = 1
   }
 }
 variables_limit = 1000
}
```

다양한 클러스터 지원 환경

On-prem / Public Cloud / Edge / IoT 통합

- 지원 OS : Linux / Windows / macOS
- CPU Arch : x86, arm64, darwin



WSL은 왜?

GPU 쓰고 싶어요!


intel CORE ULTRA 9
NVIDIA GEFORCE RTX
Powering Advanced AI
REPUBLIC OF GAMERS
ASUS Perfect Warranty
AURA SYNC
RTX5090
ULTRA 9
RAM 32GB 추가 증정
신세계 상품권 30만원
최대 혜택가 669 만
Total 최대 250W / 240Hz / 1200nits / NVMe 2TB / Win11
ROG 스트릭스 SCAR G635LX-RW047W


NVIDIA GEFORCE RTX
78,090,390원
7% 71,968,110원
AMD 쓰레드리퍼PRO 7995WX
NVIDIA H100 80GB 수냉식

GPU 나도 좀 쓰자!


intel CORE ULTRA 9
GEFORCE RTX
Powering Advanced AI
REPUBLIC OF GAMERS
ROG NEBULA HDR
AURA SYNC
RTX5090
ULTRA 9
RAM 32GB 추가 증정
신세계 상품권 30만원
최대 혜택가 669 만
Total 최대 250W / 240Hz / 1200nits / NVMe 2TB / Win11
ROG 스트릭스 SCAR G635LX-RW047W
NVIDIA
GEFORCE RTX
78,090,390원
7% 71,968,110원
AMD 쓰레드리퍼PRO 7995WX
NVIDIA H100 80GB 수냉식

GPU-Enabled Nomad 클러스터 구성

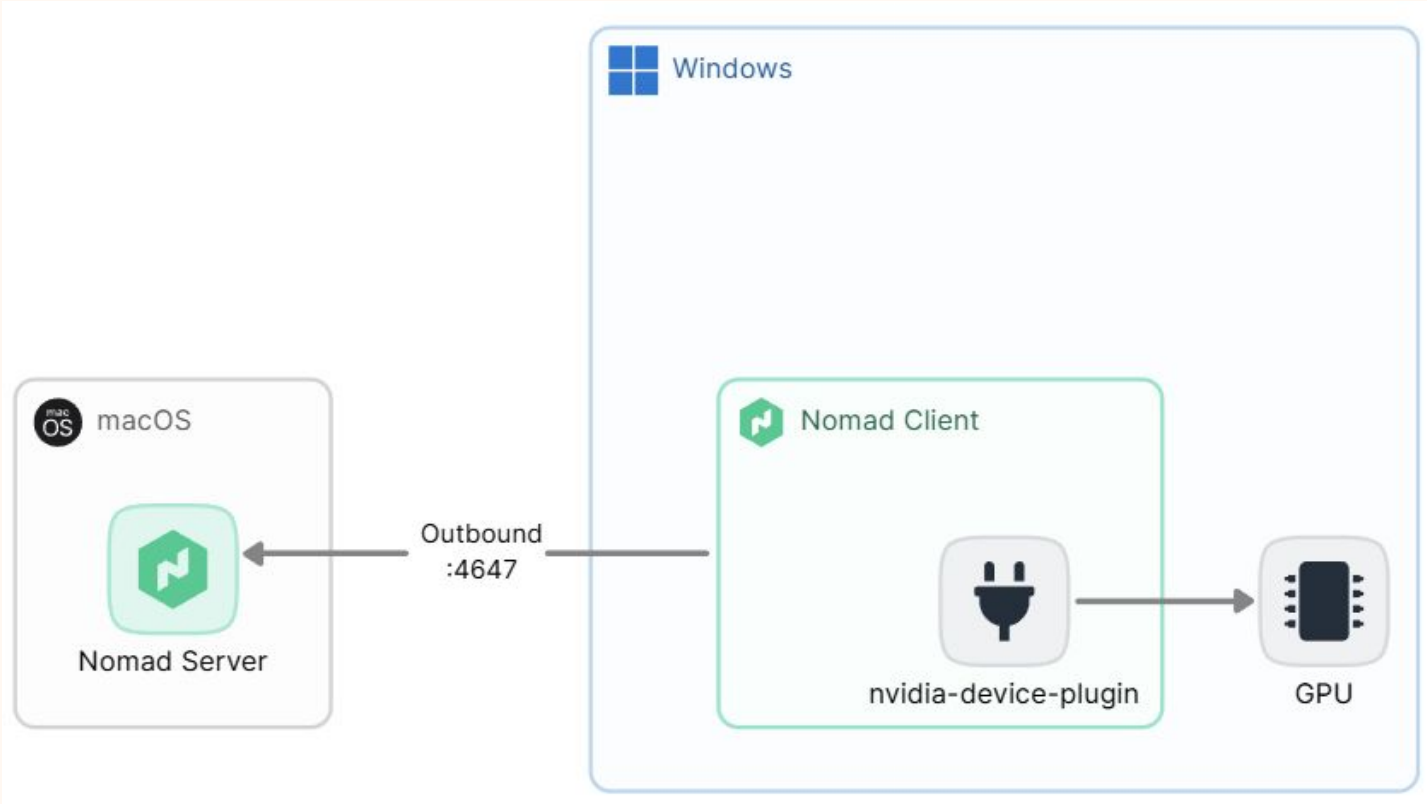
먼저 GPU 리소스를 사용할 수 있는 클러스터 구성이 핵심입니다. Nomad는 NVIDIA GPU를 지원하며, 노드 수준에서 GPU 리소스를 인식하고 작업(Job) 단위로 할당할 수 있습니다.

How to:

- GPU 드라이버 및 *nvidia-container-runtime* 설치
- Nomad 클라이언트 설정에서 **device_plugins** 활성화 및 nvidia 플러그인 등록
- Nomad에서 *resources { device { count = ... }}*를 통해 GPU 할당

```
resources {
 device "nvidia/gpu" {
   count = 1

   affinity {
     attribute = "${device.model}"
     value     = "Nvidia H100"
     weight    = 50
   }
 }
}
```

처음 구성했던 안

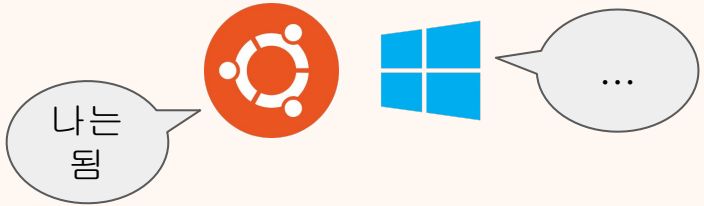

Windows
macOS
Nomad Server
Outbound
:4647
Nomad Client
nvidia-device-plugin
GPU

GPU-Enabled Nomad 클러스터 구성

먼저 GPU 리소스를 사용할 수 있는 클러스터 구성이 핵심입니다. Nomad는 NVIDIA GPU를 지원하며, 노드 수준에서 GPU 리소스를 인식하고 작업(Job) 단위로 할당할 수 있습니다.

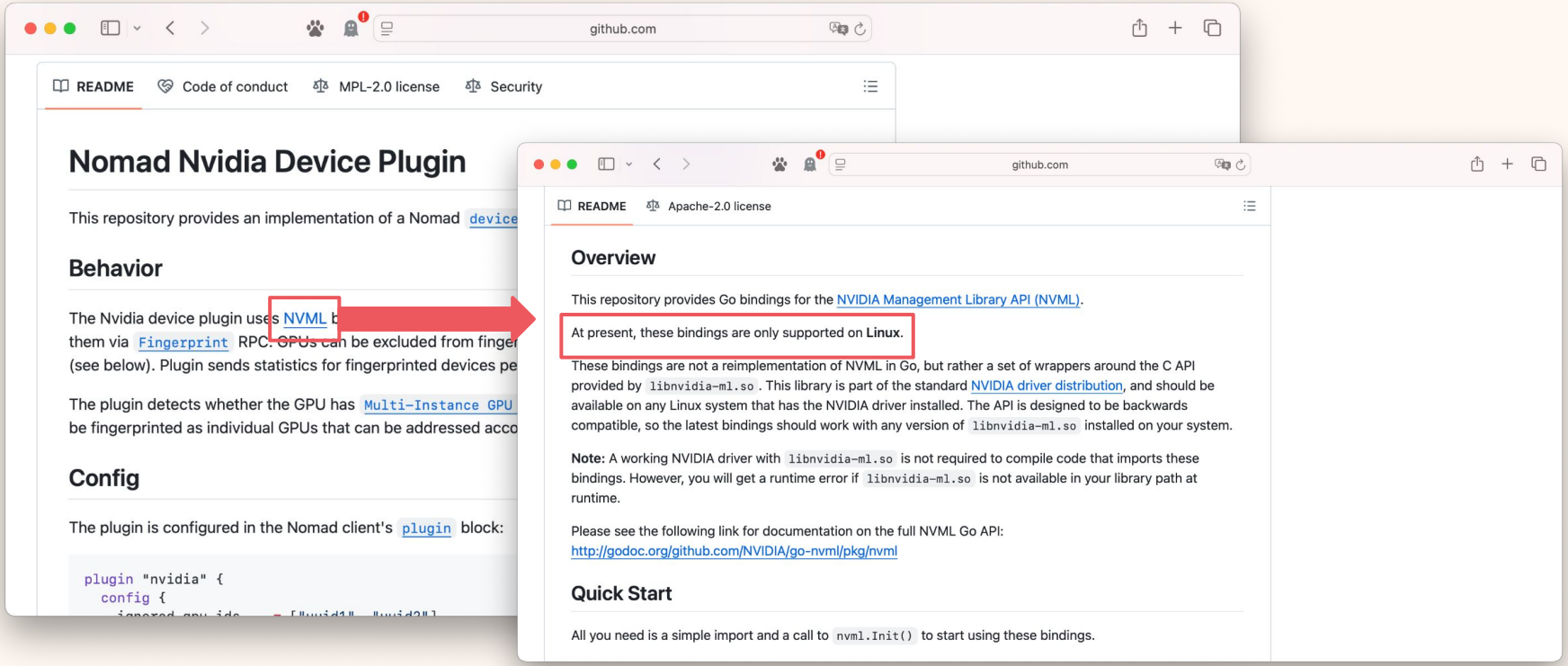
How to:

- GPU 드라이버 및 *nvidia-container-runtime* 설치
- Nomad 클라이언트 설정에서 **device_plugins** 활성화 및 nvidia 플러그인 등록
- Nomad에서 *resources { device { count = ... }}*를 통해 GPU 할당

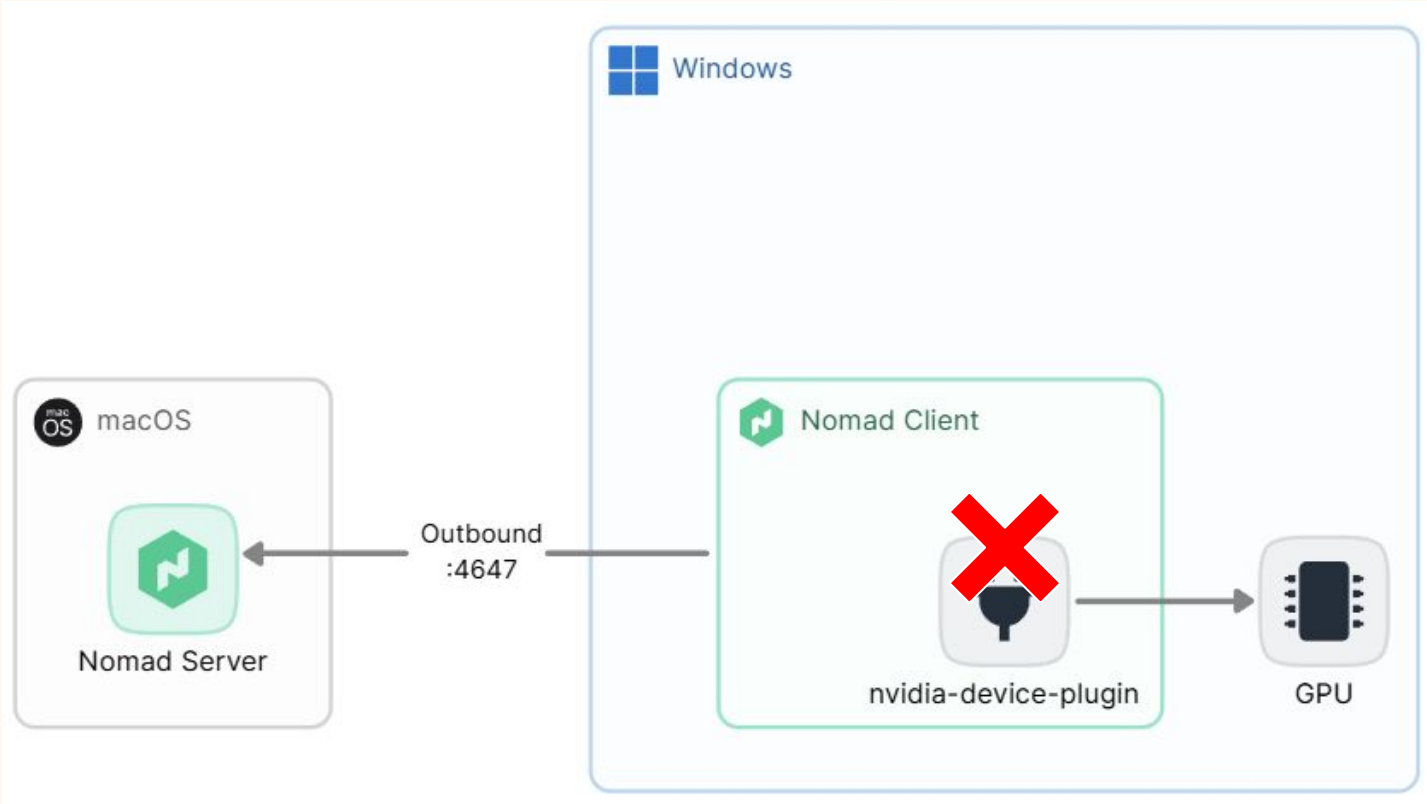



```
resources {
 device "nvidia/gpu" {
   count = 1

   affinity {
     attribute = "${device.model}"
     value     = "Nvidia H100"
     weight    = 50
   }
 }
}
```

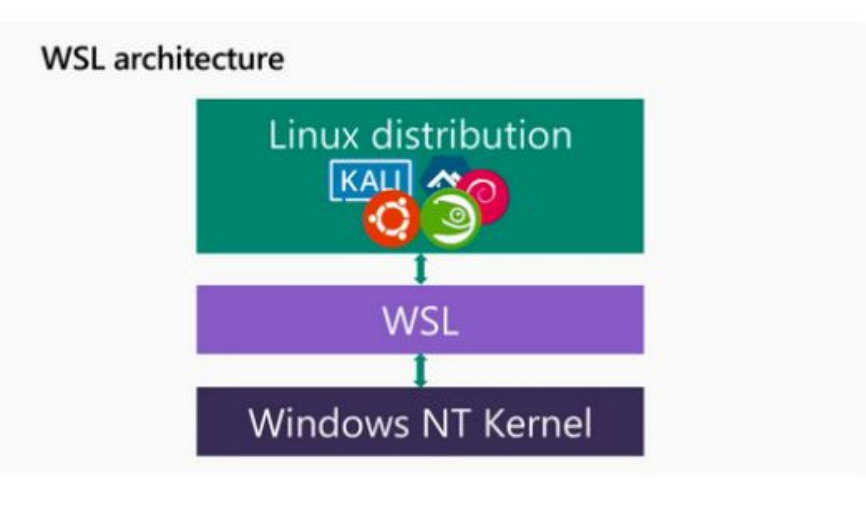
Nomad GPU Plugin > NVIDIA NVML

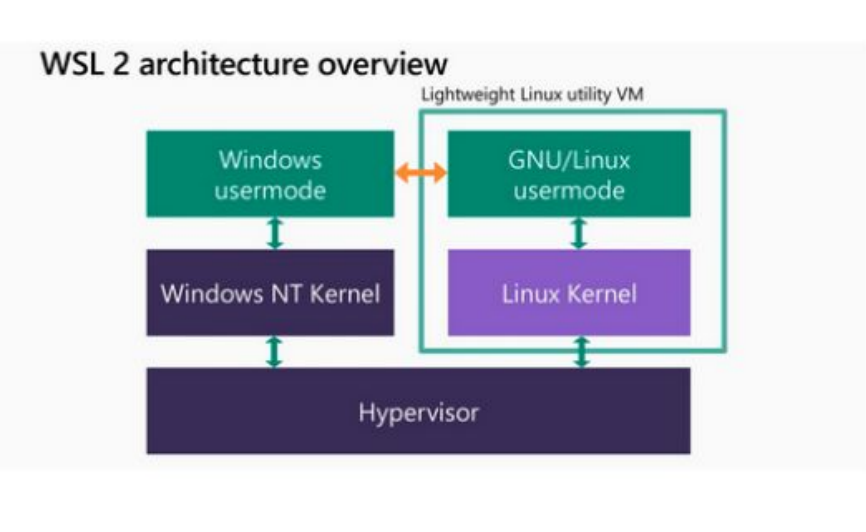


Linux만 지원한다고?

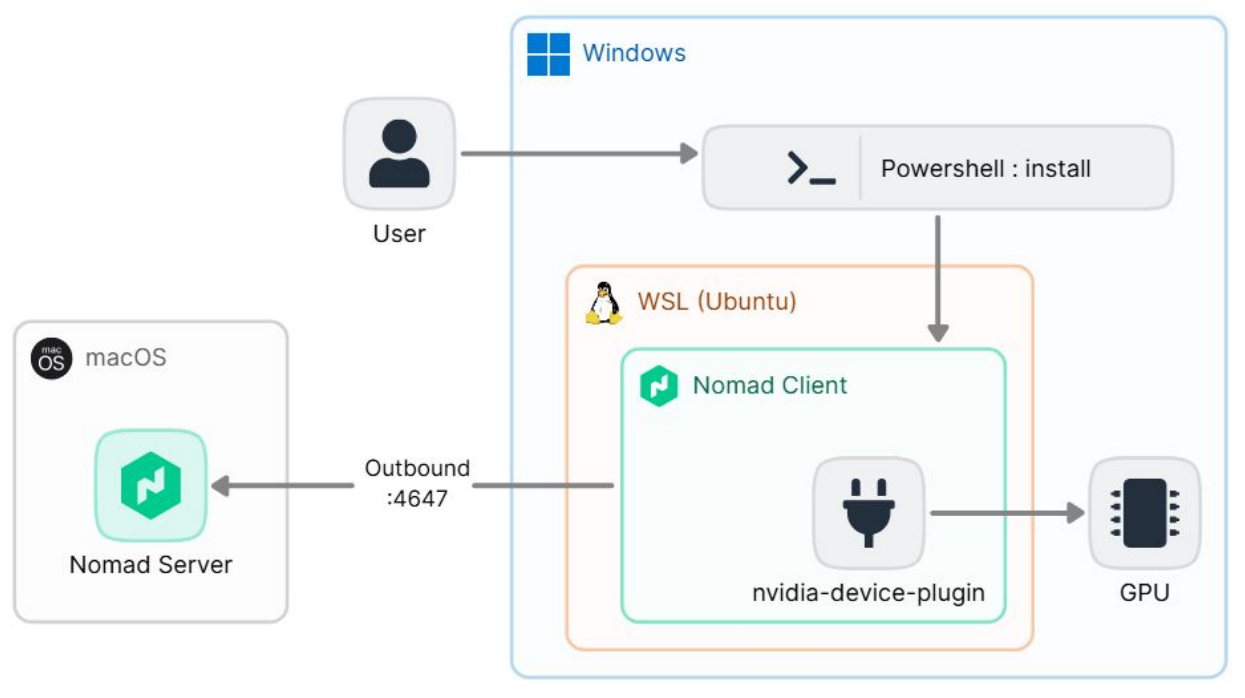


더 좋아진 **WSL 2**

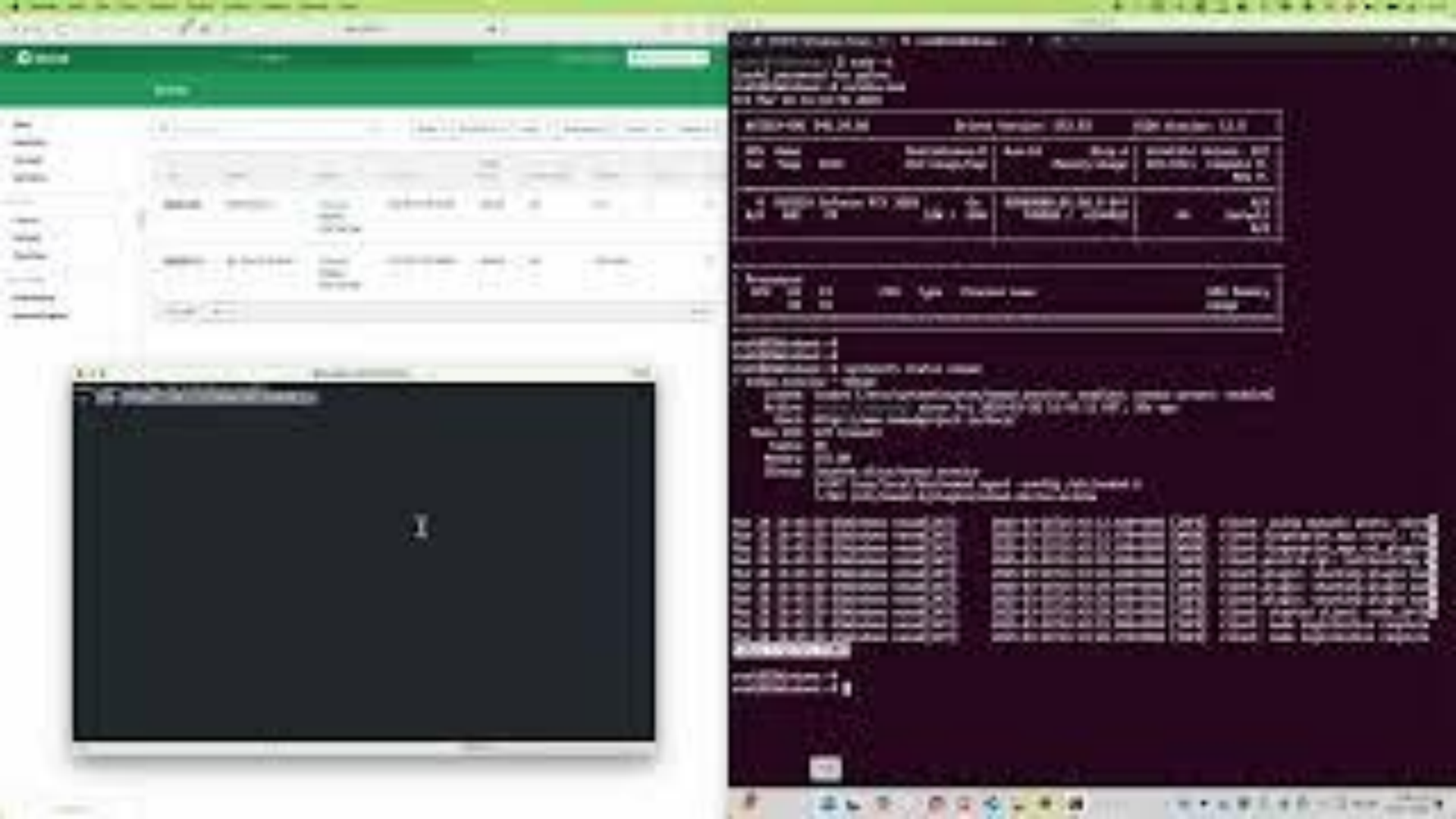

WSL architecture
Linux distribution
KALI
WSL
Windows NT Kernel


WSL 2 architecture overview
Lightweight Linux utility VM
Windows usermode
GNU/Linux usermode
Windows NT Kernel
Linux Kernel
Hypervisor

Windows에서 설치된 Driver로도 WSL에서 바로 사용



https://github.com/Great-Stone/nomad-client-on-wsl-with-nvidia-plugin



감사합니다