

보고서에 숨겨진 과학

이 섹션에는 전체 결과가 정리되어 있으며, 테스트한 솔루션과 테스트 방법론에 관한 설명이 나와 있다.

Hands-on testing은 2024년 4월 9일에 종료했다. 테스트하는 동안 적절한 하드웨어 및 소프트웨어 구성을 파악하고 업데이트가 제공될 때 이를 적용했다. 이 보고서의 결과는 2024년 3월 11일 이전에 확정된 구성을 반영하므로 이 같은 구성은 이 보고서가 공개될 당시의 최신 버전과 다를 수 있다.

차트

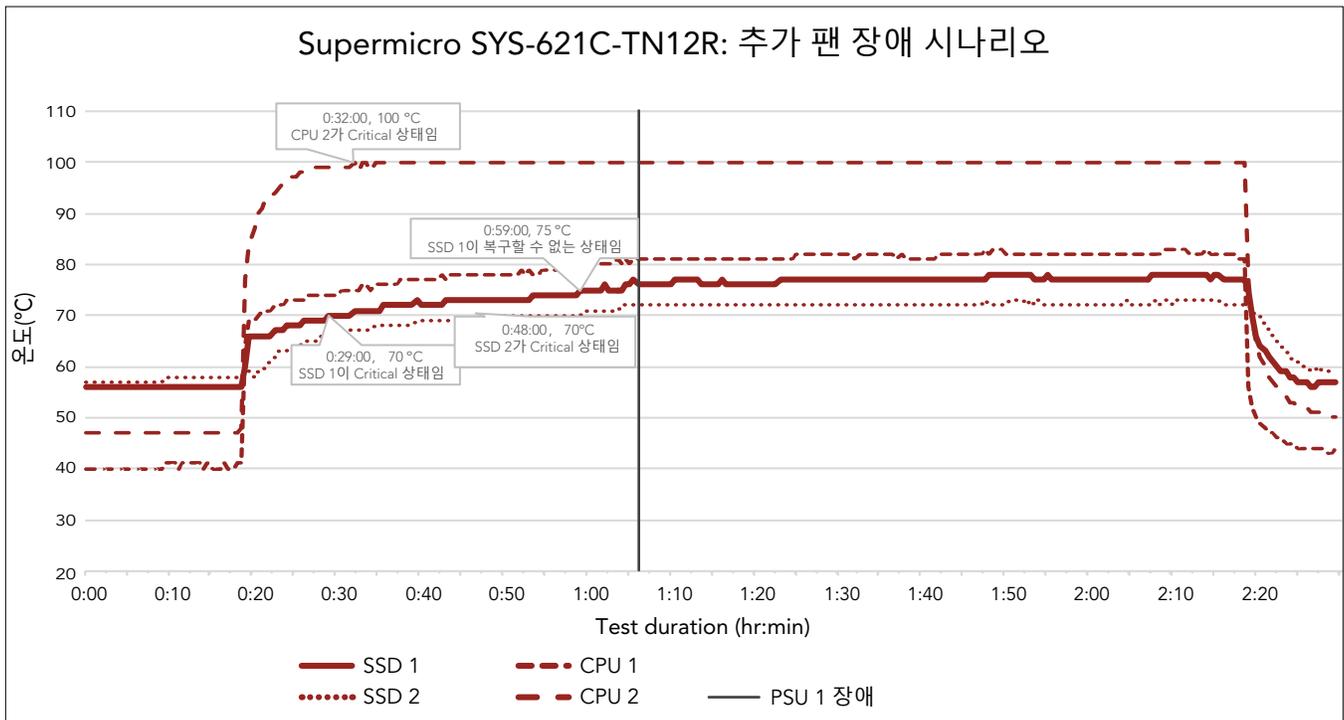


그림 1: 팬 3이 비활성화된 상태로 서버가 부동 소수점 워크로드를 실행한 추가 팬 장애 시나리오 중 Supermicro® SYS-621C-TN12R의 SSD 및 프로세서 온도다. 워크로드는 0:15에 시작하여 2:15에 완료되었다. SSD 1은 OS를 실행했고, SSD 2는 유휴 상태였다. 출처: Principled Technologies.

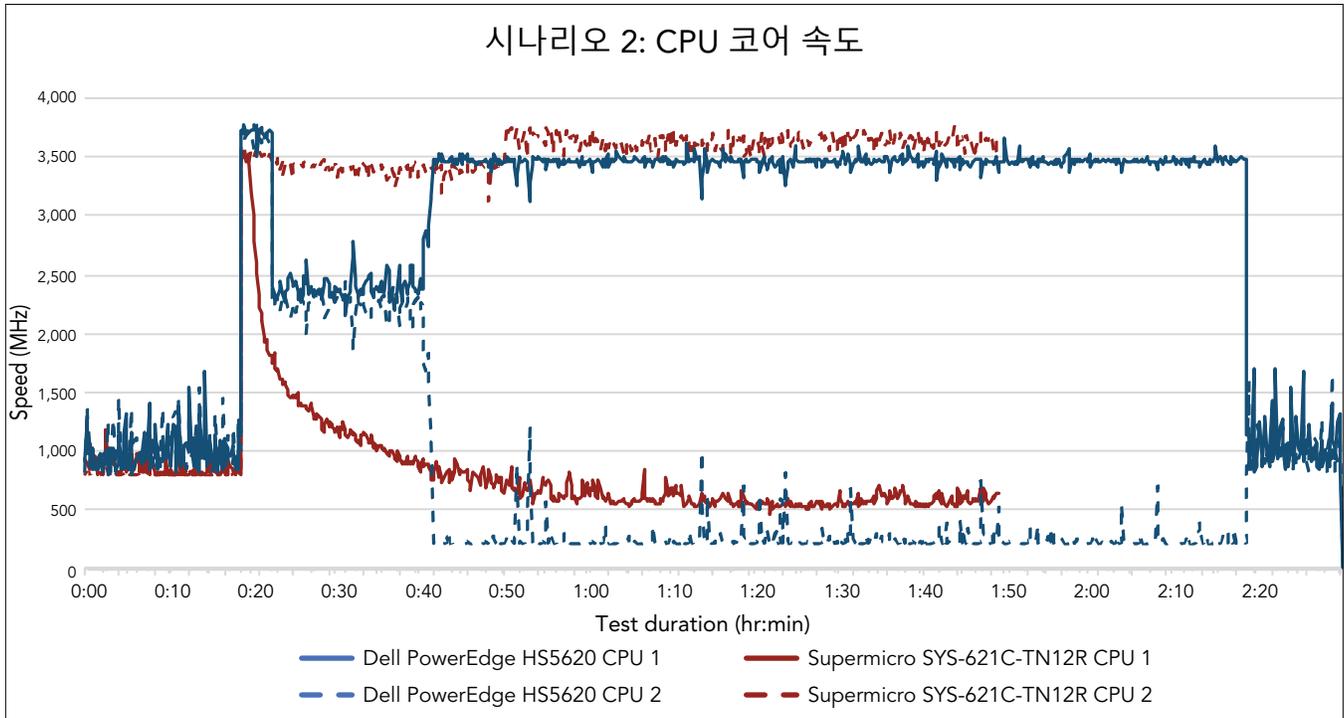


그림 2: 첫 번째 팬 장애 시나리오 중 Dell™ PowerEdge™ HS5620 및 Supermicro SYS-621C-TN12의 프로세서 코어 속도다. 부동 소수점 워크로드는 0:15에 시작하여 2:15에 완료되었다. 출처: Principled Technologies.

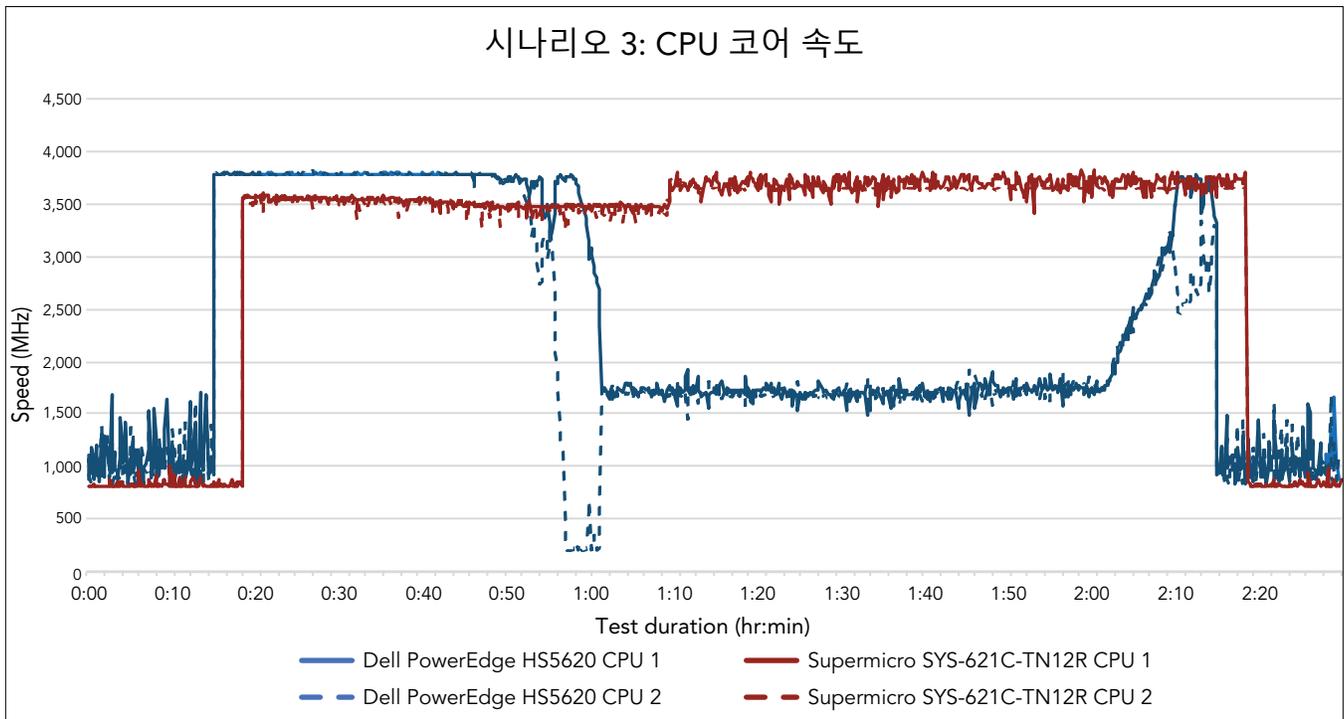


그림 3: 오작동 HVAC 시나리오 중 Dell PowerEdge HS5620 및 Supermicro SYS-621C-TN12의 프로세서 코어 속도다. 부동 소수점 워크로드는 0:15에 시작하여 2:15에 완료되었다. 출처: Principled Technologies.

시스템 구성 정보

표 1: 테스트 대상 시스템에 대한 자세한 정보

시스템 구성 정보	Dell PowerEdge HS5620	Supermicro SYS-621C-TN12R
BIOS 이름 및 버전	Dell 2.1.3	Supermicro 2.1
기본 BIOS 설정 변경	와트당 성능(OS)	해당 없음
운영 체제 이름 및 버전/빌드 번호	Ubuntu 22.04.3	Ubuntu 22.04.3
마지막 OS 업데이트/패치 적용 날짜	2024년 3월 11일	2024년 1월 28일
전원 관리 정책	와트당 성능(OS)	균형 잡힌 성능
프로세서		
프로세서 수	2	2
공급업체 및 모델	인텔® 제온® Gold 6444Y	인텔 제온 Gold 6444Y
프로세서당 코어 수	16	16
코어 주파수(GHz)	3.60(4.0 터보)	3.60(4.0 터보)
스태핑	8	8
메모리 모듈		
시스템의 총 메모리(GB)	1,024	1,024
메모리 모듈 수	16	16
공급업체 및 모델	Hynix® HMC94AEBRA109N	SK Hynix HMC94MEBRA123N
Size (GB)	64	64
유형	DDR5 DIMM	DDR5
속도(MHz)	4,800	4,800
서버에서의 실행 속도(MHz)	4,800	4,800
스토리지 컨트롤러(전면 스토리지)		
공급업체 및 모델	Dell HBA355i Adp	Supermicro MegaRAID AOC-S3916L-H16iR-32DD-P
캐시 크기(GB)	0	8
펌웨어 버전	24.15.14.00	5.240.02-3768
드라이버 버전	해당 없음	52.24.0-4766
스토리지 컨트롤러(NVMe® M.2)		
공급업체 및 모델	Dell BOSS-N1 Monolithic	해당 없음
캐시 크기(GB)	0	해당 없음
펌웨어 버전	2.1.13.2025	해당 없음
로컬 스토리지(OS)		
드라이브 수	2	2
드라이브 공급업체 및 모델	Dell NVMe PE8010 RI M.2 960GB	Micron® 7450 MTFDKBA960TFR

시스템 구성 정보	Dell PowerEdge HS5620	Supermicro SYS-621C-TN12R
드라이브 크기(GB)	960	960
드라이브 정보(속도, 인터페이스, 유형)	8GT/s M.2 SSD	PCIe® M.2 NVMe
로컬 스토리지(데이터)		
드라이브 수	12	12
드라이브 공급업체 및 모델	HGST HUH721212AL5200	WDC WUH721814ALE6L4
드라이브 크기(GB)	120,000	1,400
드라이브 정보(속도, 인터페이스, 유형)	12Gbps SAS 3.5" HDD	6Gb SATA 3.5" HDD
네트워크 어댑터 A		
공급업체 및 모델	인텔 25G 2P E810-XXV 4개	인텔 E810-XXVAM2(AOC-S25GC-i2S) 3개
포트 수/유형	25Gb 2개	25Gb 2개
드라이버 버전	22.5.7	4.20(0x800177B4)
네트워크 어댑터 B		
공급업체 및 모델	Broadcom® NetXtreme 기가비트 이더넷 (BCM5720) 1개	인텔 E810-XXVAM2(AOC-A25G-i2SM) 1개
포트 수/유형	1Gb 2개	25Gb 2개
드라이버 버전	22.71.3	4.30(0x800177B4)
냉각팬		
공급업체, 모델, 수량	Dell HPR Gold 1개 Dell HPR Silver 5개	Supermicro 중간 팬 FAN-0206L4 3개
전원 공급 장치		
공급업체 및 모델	Dell 05222NA00	Supermicro PWS-1K23A-1R
전원 공급 장치 수	2	2
개별 전력량(W)	1,800	1,200

테스트 방법

온도를 제어하고 측정할 수 있는 환경을 조성하기 위해 완전히 적재된 42U 서버 랙 주위에 맞춤형 인클로저를 구축했다. Dell 시스템과 Supermicro 시스템 모두 랙의 동일한 위치에서 테스트했으며, 세 가지 시나리오에서 내부 서버 온도를 수집했다. 랙에 있는 서버에서 stress-ng 벤치마크를 네 단계로 실행했으며, 각 단계는 1분 10초 간격으로 실행했다. 테스트한 Dell 및 Supermicro 시스템은 첫 번째 서버가 워크로드를 시작하고 3분 30초 후에 네 번째 단계에서 워크로드를 실행했다. 아래에서 테스트를 설정하고 실행하기 위해 수행한 단계를 간략히 설명한다.

Ubuntu 22.04.3 설치 및 구성

1. Ubuntu 22.04.3 미디어에서 부팅한다.
2. Try 또는 Install Ubuntu Server를 선택한다.
3. 언어 메뉴에서 기본값을 그대로 두고 Done을 선택한다.
4. Update to the new installer를 선택한다.
5. 키보드 구성에서 기본값을 그대로 두고 Done을 선택한다.
6. 설치 유형에서 기본값을 그대로 두고 Done을 클릭한다.
7. 네트워크 연결 메뉴에서 기본값을 그대로 두고 Done을 선택한다.
8. 프록시 구성 화면에서 기본값을 그대로 두고 Done을 선택한다.
9. Ubuntu 아카이브 미리 구성 화면에서 테스트가 통과될 때까지 기다린 후 Done을 선택한다.
10. 가이드 스토리지 구성 화면에서 기본값을 그대로 두고 Done을 선택한다.
11. 스토리지 구성 요약 화면에서 기본값을 그대로 두고 Done을 선택한다.
12. 데이터 삭제 작업을 확인하려면 Continue를 선택한다.
13. 프로필 설정 화면의 이름 및 사용자 이름 아래에 ptuser를 입력한다. 서버 이름 아래에 이름을 입력하고 비밀번호를 확인한다.
14. Done을 선택한다.
15. Ubuntu Pro로 업그레이드 화면에서 기본값을 그대로 두고 Continue를 선택한다.
16. SSH 설정 화면에서 Install OpenSSH server를 선택하고 Done을 선택한다.
17. 주요 서버 스냅 화면에서 기본값을 그대로 두고 Done을 선택한다.
18. 설치가 완료되면 Reboot now를 선택한다.
19. 위에서 만든 자격 증명을 사용하여 Ubuntu에 로그인한다.
20. 프로세스 업데이트:

```
sudo apt update
sudo apt upgrade
```

21. CIFS 유틸리티 설치 및 PT 공유 매핑:

```
sudo apt install cifs-utils
sudo mkdir /mnt/pt-data01
sudo mount -t cifs //10.41.1.21/pt /mnt/pt-data01/ -o "rw,user=<useraccount>,pass=<password>"
```

22. 네트워킹 구성:

```
sudo cp /etc/netplan/*.yaml /etc/netplan/00-installer-config.yaml.bak
sudo nano /etc/netplan/*.yaml
```

23. 원하는 네트워크 어댑터를 확인하고 다음 조정 수행:

```
addresses:
- <IP_Address>/<CIDR>
routes:
- to: default
  via: <Default_Gateway>
nameservers:
  search: [<NameServer1>, <NameServer2>]
  addresses: [<DNS_IP1>, <DNS_IP2>, <DNS_IP3>]
```

24. 변경된 파일을 테스트하고 적용:

```
sudo netplan try
sudo netplan apply
```

25. 호스트 이름 설정:

```
sudo hostnamectl set-hostname <NewHostname>
```

26. 호스트 다시 시작:

```
sudo shutdown -r now
```

비밀번호 없는 sudo 배포

클라이언트 측에 배포

1. sudoers 파일 편집:

```
sudo visudo /etc/sudoers
```

2. 파일의 맨 끝에 다음 내용 추가:

```
ptuser ALL=(ALL:ALL) NOPASSWD: ALL
```

컨트롤러 측에 배포

1. SSH 키 쌍 생성:

```
ssh-keygen -t rsa -b 4096 -N "" -f "$HOME/.ssh/id_rsa.pub"
```

2. SSH 공개 키를 각 원격 서버에 복사:

```
ssh-copy-id ptuser@<remote_server_IP>
```

데이터 수집을 위해 TIG-P 스택 구현

Docker 구성

1. 로깅 머신에 ptuser로 로그인한다.

2. Docker 설치 준비:

```
sudo apt update
sudo apt install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

3. Apt 소스에 리포지토리 추가 및 설치:

```
echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.
```

```
docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME" stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Huntabyte TIG 스택 구성

1. 로깅 머신에서 tig-stack 리포지토리 복제:

```
git clone https://github.com/huntabyte/tig-stack.git
```

2. 배포에 맞게 .env 파일 편집:

```
sudo nano tig-stack/.env
```

3. FluxDB의 경우 다음과 같이 사용자 이름, 비밀번호, 조직, 버킷 및 보존 기간 입력:

```
DOCKER_INFLUXDB_INIT_USERNAME: admin
DOCKER_INFLUXDB_INIT_Password: <PasswordHere>
DOCKER_INFLUXDB_INIT_ORG: PT
DOCKER_INFLUXDB_INIT_BUCKET: <BucketName>
DOCKER_INFLUXDB_INIT_RETENTION: 52w
```

4. 다음 명령을 사용하여 임의의 32자 16진수 문자열을 생성하고 .env 파일에 관리자 토큰의 결과 입력:

```
openssl rand -hex 32
```

5. 저장하고 종료한다.

6. telegraf.conf 편집:

```
sudo nano tig-stack/telegraf/telegraf.conf
```

7. 다음 값 설정:

```
services:
  influxdb:
    image: influxdb
  telegraf:
    image: gibletron/telegraf-ipmitool
  grafana:
    image: grafana/grafana-oss
    links:
      - prometheus
```

8. 저장하고 종료한다.

9. Docker compose(headless/detached) 시작:

```
sudo docker-compose up -d
```

10. 모니터링하려는 각 서버에서 다음 명령 실행:

```
sudo apt install telegraf
```

11. InfluxDB 관리 인터페이스를 열려면 포트 8086이 있는 InfluxDB 호스트의 IP 주소로 이동한다.

- 필요에 따라 API 토큰을 생성하고, 창을 닫기 전에 반드시 기록해 둔다.
- Load Data 아래에서 API Tokens를 클릭하고 Generate API Token을 클릭한다.
- 모니터링하려는 각 서버에서 다음을 사용하여 /etc/Telegraf/Telegraf.conf 편집:

```
[[outputs.influxdb_v2]]
  urls = ["<influxDB_IP>:8086"]
  token = "<API_token>"
  organization = "PT"
  bucket = "<bucket_name>"
```

- 테스트 중인 각 시스템에서 다음 추가:

```
[[inputs.intel_powerstat]]
  cpu_metrics = ["cpu_frequency"]
```

- 저장하고 종료한다.
- Telegraf 다시 시작:

```
sudo systemctl restart telegraf
```

Prometheus 구성

- /home/ptuser/tig-stack/docker-compose.yml에 다음 추가:

```
prometheus:
  image: prom/prometheus:latest
  volumes:
    - ${PROM_CFG_PATH}:/etc/prometheus/prometheus.yml
    - prom-storage:/prometheus
  ports:
    - 9090:9090
  volumes:
    prom-storage:
```

- 저장하고 종료한다.
- .env 파일을 편집하고 다음 추가:

```
PROM_CFG_PATH=./prometheus/prometheus.yml
```

- 저장하고 종료한다.
- 모니터링하려는 각 서버에서 다음 명령 실행:

```
sudo apt install dbus prometheus-node-exporter prometheus-node-exporter-collectors -y
```

- 테스트 중인 각 시스템에서 다음 명령 실행:

```
sudo apt install prometheus -y
```

- Prometheus에서 모니터링 작업을 생성하려면 /home/ptuser/tig-stack/prometheus/prometheus.yml에 다음 추가:

```
- job_name: "<custom_name>"
  static_configs:
    - targets: ["<target_IP:9090>"]
```

- 7단계와 유사한 추가 항목을 생성하여 추가 작업 및/또는 타겟을 추가한다. 동일한 작업에 대해 다른 타겟을 또 다른 타겟 줄로 추가할 수 있다.
- 저장하고 종료한다.

stress-ng를 사용하여 테스트

각 테스트 시나리오에서 다음 단계를 따라 stress-ng 부동 소수점 워크로드를 실행했다.

1. 각 서버에서 다음 명령 실행:

```
sudo apt install stress-ng linux-tools-generic -y
```

2. 테스트 중인 각 서버에서 다음 명령 실행:

```
sudo modprobe rapl
sudo modprobe intel_rapl_common
sudo modprobe intel_rapl_msr
sudo modprobe msr
sudo modprobe intel-uncore-frequency
sudo setcap cap_sys_rawio,cap_dac_read_search,cap_sys_admin+ep /usr/bin/telegraf
sudo chmod -R a+rx /sys/devices/virtual/powercap/intel-rapl/
```

3. 테스트 중인 각 서버에서 https://github.com/andikleen/pmu-tools/blob/master/event_download.py로 이동하여 원시 파일을 다운로드한 후 실행:

```
sudo chmod +x event_download.py
./event_download.py
```

4. 컨트롤러에서 PSSH 설치:

```
sudo apt install pssh -y
```

5. 컨트롤러에서 stress-ng 실행 중에 사용할 파일 생성:

```
sudo touch ~/.pssh_hosts_file
sudo touch ~/.pssh_hosts_file_wave1
sudo touch ~/.pssh_hosts_file_wave2
sudo touch ~/.pssh_hosts_file_wave3
sudo touch ~/.pssh_hosts_file_wave4
```

6. ~/.pssh_hosts_file 파일을 편집하고 각 줄에 하나씩 모든 서버 IP 주소를 입력한다.
7. ~/.pssh_hosts_file_wave1 파일부터 ~/.pssh_hosts_file_wave4 파일까지 편집하고 각 파일에 IP 주소의 1/4을 적절히 입력한다.
8. 모든 서버가 온라인 상태이고 원격 명령에 응답하는지 테스트:

```
sudo pssh -i -h ~/.pssh_hosts_file uptime
```

9. 컨트롤러에서 stress-ng 테스트에 대한 로그 폴더 생성:

```
sudo mkdir /var/log/stress-ng
sudo chmod 777 /var/log/stress-ng
```

10. 다음 명령으로 테스트를 실행하며, 적절한 단계 번호를 사용하여 "wave1"을 편집한다.

```
pssh -i -h ~/.pssh_hosts_file_wave1 sudo stress-ng --cpu 4 --matrix 0 --cpu-method matrixprod --mq 4 --hdd 6 --tz --metrics --perf --times --aggressive -t 2h --log-file /var/log/stress-ng/$(date +%Y%m%d_%H%M%S').log
```

▶ 이 보고서의 원본(영문) 보기:
<https://facts.pt/gPS09my>

이 프로젝트는 Dell Technologies의 의뢰로 진행되었습니다.



Facts matter.®

Principled Technologies는 Principled Technologies, Inc.의 등록 상표입니다.
다른 모든 제품 이름은 해당 소유주의 상표입니다.

보증 및 책임의 면책:

Principled Technologies, Inc.는 테스트의 정확성과 유효성을 보장하기 위한 합리적인 노력을 기울였습니다. 하지만 Principled Technologies, Inc.는 특정 목적에의 적합성에 대한 암시적인 보증을 비롯하여 테스트 결과 및 분석, 정확성, 완전성 또는 품질에 대한 어떠한 명시적 또는 암시적 보증도 명확히 거부합니다. 테스트 결과 이용에 따른 모든 위험은 해당 개인 또는 단체가 스스로 감수해야 하며 Principled Technologies, Inc., 그 직원 및 하청업체가 테스트 절차 또는 결과의 오류 또는 결함으로 인해 발생하는 손해 또는 배상 소송에 대해 어떠한 책임도 지지 않는다는 것에 동의합니다.

어떠한 경우에도 Principled Technologies, Inc.는 테스트와 관련하여 발생하는 간접적, 특별적, 우발적 또는 파생적 손해에 대해 책임지지 않습니다. 이는 그러한 손해의 가능성을 사전에 인지한 경우에도 마찬가지입니다. 어떠한 경우에도 직접적 손해를 포함한 Principled Technologies, Inc.의 책임은 Principled Technologies, Inc.에게 지불된 테스트 비용을 초과하지 않습니다. 여기에 명시된 내용이 고객의 유일한 구제책입니다.