

Viante installation guide

- [Viante installation guide](#)
- [Viante installation guide minimal](#)

Viante installation guide

https://cloud.znotes.ru/public.php/dav/files/vcloud/Platform_4-lic.zip

<https://cloud.znotes.ru/public.php/dav/files/vcloudai/CluebaseVMS.zip>

<https://drive.google.com/file/d/1x2HiTNweSBRE1tzionbT7wdtT8pkEETP/view>

Check if port is open: `netstat -tulnp | grep 8000`

If multiple ports:

```
ports="8080 80 554 15672 2379 40000 5180 9100 4001 3000 3307 8565 8095 10000 8081"
for port in $ports; do
    if nc -z localhost $port >/dev/null 2>&1; then
        echo "Port $port is in use"
    else
        echo "Port $port is free"
    fi
done
```

```
ports="3478 5349 49152-65535"
for port in $ports; do
    if nc -z localhost $port >/dev/null 2>&1; then
        echo "Port $port is in use"
    else
        echo "Port $port is free"
    fi
done
```

If VM check for avx2:

```
grep "flags" /proc/cpuinfo | grep "avx2"
```

-=== Installing the NVIDIA Container Toolkit 1.19 ===-

```
sudo apt-get update && sudo apt-get install -y --no-install-recommends ca-certificates curl  
gnupg2 docker-compose-plugin
```

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o  
/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \  
&& curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-  
toolkit.list | \  
    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-  
keyring.gpg] https://#g' | \  
    sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

```
sudo apt-get update
```

```
export NVIDIA_CONTAINER_TOOLKIT_VERSION=1.19.0-1  
sudo apt-get install -y \  
    nvidia-container-toolkit=${NVIDIA_CONTAINER_TOOLKIT_VERSION} \  
    nvidia-container-toolkit-base=${NVIDIA_CONTAINER_TOOLKIT_VERSION} \  
    libnvidia-container-tools=${NVIDIA_CONTAINER_TOOLKIT_VERSION} \  
    libnvidia-container1=${NVIDIA_CONTAINER_TOOLKIT_VERSION}
```

```
sudo nvidia-ctk runtime configure --runtime=docker
```

```
nano /etc/docker/daemon.json
```

Add "min-api-version": "1.24", in file /etc/docker/daemon.json

```
sudo systemctl restart docker
```

```
sysctl fs.aio-max-nr
```

```
nano /etc/sysctl.d/99-aio.conf
```

```
fs.aio-max-nr = 1048576
```

```
sysctl --system
```

```
chmod a+x \  
./install.py \  

```

```
./uninstall.py \  
./restart.sh \  
./start.sh \  
./stop.sh \  
./collect_logs.sh
```

```
vms-phpmyadmin:  
  image: phpmyadmin:latest  
  container_name: vms-phpmyadmin  
  restart: always  
  ports:  
    - 18080:80  
  environment:  
    PMA_HOST: vms-db  
    PMA_PORT: 3306  
    MYSQL_ROOT_PASSWORD: ${MYSQL_PASSWORD}  
  depends_on:  
    vms-db:  
      condition: service_healthy
```

```
NVIDIA_VISIBLE_DEVICES: all  
NVIDIA_DRIVER_CAPABILITIES: all  
runtime: nvidia  
deploy:  
  resources:  
    reservations:  
      devices:  
        - driver: nvidia  
          count: all  
          capabilities: [gpu]
```

===== ollama

=====

=====

```
docker run -d \  
--gpus=all \  
--name ollama \  
--restart=always \  

```

```
-e OLLAMA_SCHED_SPREAD=1 \  
-v ollama:/root/.ollama \  
-p 11434:11434 \  
--health-cmd='curl -fsS http://0.0.0.0:11434/api/tags || exit 1' \  
--health-interval=30s \  
--health-timeout=5s \  
--health-retries=3 \  
--health-start-period=20s \  
ollama/ollama
```

=====platform 4 CPU =====

```
./install.py \  
--ecm-rabbitmq-management-port=15673 \  
--ecm-port=8081 \  
--skip-docker-images-load \  
--im-fr-algorithm alg2 \  
--im-enroll-hardware-type cpu \  
--fd-hardware-type cpu \  
--ecm-data-storage-period-in-days 30 \  
--ecm-data-index-period-in-days 30 \  
--ecm-webhook-url http://118.107.96.20/api/event/addTevianEvent \  
--ecm-admin-password a10@2025 \  
--ecm-rabbitmq-password a10@2025
```

```
--ecm-webhook-url http://127.0.0.1/api/event/addTevianEvent \  
--ecm-no-webhook
```

=====platform 4 GPU =====

```
./install.py \  
--ecm-rabbitmq-management-port=15673 \  
--ecm-port=8081 \  
--im-fr-algorithm alg2 \  
--im-enroll-hardware-type gpu \  
--im-enroll-gpu-device-id 1 \  
--fd-hardware-type gpu \  
--fd-gpu-device-id 1 \  
--with-silhouette \  

```

```
--silhouette-with-classifiers \  
--silhouette-gpu-device-id 1 \  
--ecm-data-storage-period-in-days 30 \  
--ecm-data-index-period-in-days 30 \  
--ecm-webhook-url http://192.168.69.2:8000/api/event/addTevianEvent \  
--ecm-admin-password admin \  
--ecm-rabbitmq-password admin
```

```
--with-silhouette \  
--silhouette-with-classifiers \  
--silhouette-gpu-device-id 0 \  
--im-api-port 40001 \  
  
--im-scylla-skip-schema-load \  
--skip-docker-images-load \  

```

=====platform 5 CPU =====

```
./install.py \  
--ecm-rabbitmq-management-port=15673 \  
--ecm-port=8081 \  
--im-fr-algorithm alg2 \  
--im-enroll-hardware-type cpu \  
--fd-hardware-type cpu \  
--ecm-data-storage-period-in-days 30 \  
--ecm-data-index-period-in-days 30 \  
--with-silhouette \  
--silhouette-hardware-type cpu \  
--silhouette-with-classifiers \  
--ecm-webhook-url http://192.168.0.72/api/event/addTevianEvent \  
--ecm-admin-password admin \  
--ecm-rabbitmq-password admin
```

=====platform 5 GPU =====

```
./install.py \  
--ecm-rabbitmq-management-port=15673 \  
--ecm-port=8081 \  
--im-fr-algorithm alg5 \  
--im-enroll-hardware-type gpu \  
--im-enroll-gpu-device-id 0 \  

```

```
--fd-hardware-type gpu \  
--fd-gpu-device-id 0 \  
--ecm-data-storage-period-in-days 30 \  
--ecm-data-index-period-in-days 30 \  
--with-silhouette \  
--silhouette-hardware-type gpu \  
--silhouette-gpu-device-id 1 \  
--silhouette-with-classifiers \  
--ecm-webhook-url http://192.168.73.2/api/event/addTevianEvent \  
--ecm-admin-password admin \  
--ecm-rabbitmq-password admin
```

Viante installation guide

minimal

1. Check if port is open: `netstat -tulnp | grep 8081`

If multiple ports:

```
ports="8080 80 554 15672 2379 40000 5180 9100 4001 3000 3307 8565 8095 10000 8081"
for port in $ports; do
  if nc -z localhost $port >/dev/null 2>&1; then
    echo "Port $port is in use"
  else
    echo "Port $port is free"
  fi
done
```

2. Checking the processor for avx2 support:

```
grep "flags" /proc/cpuinfo | grep "avx2"
```

3. Add "min-api-version": "1.24", in file `/etc/docker/daemon.json`

```
sudo nano /etc/docker/daemon.json
```

```
sudo systemctl restart docker
```

```
sudo sysctl fs.aio-max-nr
```

4. Add "fs.aio-max-nr = 1048576" in file `/etc/sysctl.d/99-aio.conf`

```
sudo nano /etc/sysctl.d/99-aio.conf
```

```
fs.aio-max-nr = 1048576
```

```
sudo sysctl --system
```


Go to the folder where the archive is unpacked (for example, /opt/Platform-4-lic)

```
sudo chmod a+x \  
./install.py \  
./uninstall.py \  
./restart.sh \  
./start.sh \  
./stop.sh \  
./collect_logs.sh
```

=====platform 4 CPU =====

```
./install.py \  
--ecm-rabbitmq-management-port=15673 \  
--ecm-port=8081 \  
--im-fr-algorithm alg2 \  
--im-enroll-hardware-type cpu \  
--fd-hardware-type cpu \  
--ecm-data-storage-period-in-days 30 \  
--ecm-data-index-period-in-days 30 \  
--ecm-webhook-url http://vms-server-ip:port/api/event/addTevianEvent \  
--ecm-admin-password admin \  
--ecm-rabbitmq-password admin
```

=====platform 4 GPU =====

```
./install.py \  
--ecm-rabbitmq-management-port=15673 \  
--ecm-port=8081 \  
--im-fr-algorithm alg2 \  
--im-enroll-hardware-type gpu \  
--im-enroll-gpu-device-id 0 \  
--fd-hardware-type gpu \  
--fd-gpu-device-id 0 \  
--ecm-data-storage-period-in-days 30 \  
--ecm-data-index-period-in-days 30 \  
--ecm-webhook-url http://vms-server-ip:port/api/event/addTevianEvent \  
--ecm-admin-password admin \  
--ecm-rabbitmq-password admin
```

--ecm-no-webhook \

--with-silhouette \

--silhouette-with-classifiers \

--silhouette-gpu-device-id 0 \

--im-api-port 40001 \

--im-scylla-skip-schema-load \

--skip-docker-images-load \