

EntPEP trial: install guide

Everything you need to run your own EntPEP appliance in your AWS account: an **agent host**

(`host.yourdomain.com`) and a **private server** (`data.yourdomain.com`: the data lake and an optional private model). About an hour from nothing to the demo. Trial builds work until **1 January 2027**. Every command is one line: copy it whole. Downloads and videos: <https://entpep.com/trial>

```
your browser —443→ host (Agent Fleet Manager; each agent on its own port 8001–8008)
                        | model calls → Anthropic API, or Claude on Amazon Bedrock,
                        |               or —443→ data (private model, qwen3:14b)
                        └ data lake   —8443→ data (ClickHouse, demo trade blotter)
```

What you need

Item	Requirement
AWS account	Rights to create instances, Elastic IPs and security groups. The AWS CLI v2, signed in (<code>aws login</code> or <code>aws configure</code>)
GPU quota	Running On-Demand G and VT instances at least 4 vCPUs (quota code <code>L-DB2E81BA</code>). New accounts often have 0; request it first, it can take a day
Two DNS names	<code>host.yourdomain.com</code> and <code>data.yourdomain.com</code> , A records you can edit
Microsoft 365	A tenant admin for about 20 minutes: a SharePoint site holds the demo files, and an Entra app gives the host access to that one site only
A model	Any of three: an Anthropic API key ; a Bedrock API key (Claude in your own AWS account); or the private model on the data server (no outside model at all)

Cost: about \$1.25/hour with both running (m8i.2xlarge about \$0.42, g6.xlarge about \$0.80), plus about \$15/month for disks and the two addresses. Stop both when you're not using them.

Ports

Server	Inbound port	From	Why
host	TCP 443	your users' addresses	the Agent Fleet Manager
host	TCP 8000-8999	your users' addresses	each agent opens in its own browser tab on its own port (8001-8008 today)
host	TCP 22	your address	SSH
host	TCP 80	anywhere	Let's Encrypt certificate
data	TCP 443	the host's address	the private model's API
data	TCP 8443	the host's address	the data lake (ClickHouse behind nginx, IP allowlist)
data	TCP 22	your address	SSH
data	TCP 80	anywhere	Let's Encrypt certificate

The host needs outbound HTTPS to the model (api.anthropic.com or Bedrock), to login.microsoftonline.com, graph.microsoft.com and your SharePoint host, and to the data server.

1. On your computer: names and settings

Paste these one at a time (macOS or Linux shell; change the first three):

```
H=host.yourdomain.com
D=data.yourdomain.com
REGION=us-east-2
MYIP=$(curl -s https://checkip.amazonaws.com)/32
VPC=$(aws ec2 describe-vpcs --region $REGION --filters Name=is-default,Values=true --query 'Vpcs[0].VpcId' --output text)
AZ=$(aws ec2 describe-instance-type-offerings --region $REGION --location-type availability-zone --filters Name=instance-type,Values=g6.xlarge --query 'InstanceTypeOfferings[0].Location' --output text)
SUBNET=$(aws ec2 describe-subnets --region $REGION --filters Name=vpc-id,Values=$VPC Name=availability-zone,Values=$AZ --query 'Subnets[0].SubnetId' --output text)
AMI=resolve:ssm:/aws/service/canonical/ubuntu/server/26.04/stable/current/amd64/hvm/ebs-gp3/ami-id
echo "vpc $VPC zone $AZ subnet $SUBNET"
```

An SSH key pair (skip if you have one; then set KEY to its name):

```
KEY=entpep-trial; aws ec2 create-key-pair --region $REGION --key-name $KEY --query KeyMaterial --output text > ~/.ssh/$KEY.pem && chmod 600 ~/.ssh/$KEY.pem
SSHK="-i $HOME/.ssh/$KEY.pem"
```

Check the GPU quota is at least 4: aws service-quotas get-service-quota --region \$REGION --service-code ec2 --quota-code L-DB2E81BA --query Quota.Value

2. Two fixed addresses (Elastic IPs)

Fixed addresses keep the DNS records and the data server's allowlist right after a stop and start.

```
HALLOC=$(aws ec2 allocate-address --region $REGION --domain vpc --query AllocationId --output text)
DALLOC=$(aws ec2 allocate-address --region $REGION --domain vpc --query AllocationId --output text)
HIP=$(aws ec2 describe-addresses --region $REGION --allocation-ids $HALLOC --query
'Addresses[0].PublicIp' --output text)
DIP=$(aws ec2 describe-addresses --region $REGION --allocation-ids $DALLOC --query
'Addresses[0].PublicIp' --output text)
echo "host $H = $HIP    data $D = $DIP"
```

3. DNS

Two **A records** at your DNS provider: `$H` → `$HIP` and `$D` → `$DIP`. On Cloudflare, set them to **DNS only** (grey cloud), not proxied. Check: `dig +short $H`; `dig +short $D`

4. Security groups

```
HSG=$(aws ec2 create-security-group --region $REGION --vpc-id $VPC --group-name entpep-host --
description "EntPEP agent host" --query GroupId --output text)
aws ec2 authorize-security-group-ingress --region $REGION --group-id $HSG --protocol tcp --port 22 --
cidr $MYIP
aws ec2 authorize-security-group-ingress --region $REGION --group-id $HSG --protocol tcp --port 443 --
cidr $MYIP
aws ec2 authorize-security-group-ingress --region $REGION --group-id $HSG --protocol tcp --port 8000-
8999 --cidr $MYIP
aws ec2 authorize-security-group-ingress --region $REGION --group-id $HSG --protocol tcp --port 80 --
cidr 0.0.0.0/0
DSG=$(aws ec2 create-security-group --region $REGION --vpc-id $VPC --group-name entpep-data --
description "EntPEP private server" --query GroupId --output text)
aws ec2 authorize-security-group-ingress --region $REGION --group-id $DSG --protocol tcp --port 22 --
cidr $MYIP
aws ec2 authorize-security-group-ingress --region $REGION --group-id $DSG --protocol tcp --port 443 --
cidr $HIP/32
aws ec2 authorize-security-group-ingress --region $REGION --group-id $DSG --protocol tcp --port 8443 --
-cidr $HIP/32
aws ec2 authorize-security-group-ingress --region $REGION --group-id $DSG --protocol tcp --port 80 --
cidr 0.0.0.0/0
```

To let colleagues use the Fleet Manager, add their addresses to the host's 443 and 8000-8999.

5. Launch both servers

The agent host **must** have nested virtualization on: each agent runs in its own small VM. The AWS web console has no setting for it, so launch from the CLI.

```
HID=$(aws ec2 run-instances --region $REGION --instance-type m8i.2xlarge --image-id $AMI --cpu-options
NestedVirtualization=enabled --key-name $KEY --subnet-id $SUBNET --security-group-ids $HSG --metadata-
options HttpTokens=required,HttpEndpoint=enabled --block-device-mappings 'DeviceName=/dev/sda1,Ebs=
{VolumeSize=30,VolumeType=gp3,Encrypted=true}' --tag-specifications 'ResourceType=instance,Tags=
[{Key=Name,Value=EntPEP-Host}]' --query 'Instances[0].InstanceId' --output text)
DID=$(aws ec2 run-instances --region $REGION --instance-type g6.xlarge --image-id $AMI --key-name $KEY
--subnet-id $SUBNET --security-group-ids $DSG --metadata-options
HttpTokens=required,HttpEndpoint=enabled --block-device-mappings 'DeviceName=/dev/sda1,Ebs=
{VolumeSize=80,VolumeType=gp3,Encrypted=true}' --tag-specifications 'ResourceType=instance,Tags=
[{Key=Name,Value=EntPEP-Data}]' --query 'Instances[0].InstanceId' --output text)
aws ec2 wait instance-running --region $REGION --instance-ids $HID $DID
aws ec2 associate-address --region $REGION --instance-id $HID --allocation-id $HALLOC
aws ec2 associate-address --region $REGION --instance-id $DID --allocation-id $DALLOC
aws ec2 describe-instances --region $REGION --instance-ids $HID --query
'Reservations[0].Instances[0].CpuOptions.NestedVirtualization'
```

The last line must print "enabled". Write down HID and DID (echo \$HID \$DID) for later.

Launched the host from the web console? Stop it, turn nested virtualization on, start it: `aws ec2 stop-instances --region $REGION --instance-ids $HID && aws ec2 wait instance-stopped --region $REGION --instance-ids $HID && aws ec2 modify-instance-cpu-options --region $REGION --instance-id $HID --nested-virtualization enabled && aws ec2 start-instances --region $REGION --instance-ids $HID`

6. Install the private server (data)

`ssh $SSHK ubuntu@$D`, then on data. In the commands below, replace `data.yourdomain.com`, `you@yourdomain.com` (for Let's Encrypt) and `HOST_IP` (the host's Elastic IP, `$HIP`).

Pass 1 installs the NVIDIA driver and stops:

```
curl -sSf0 https://entpep.com/dl/combined-private-install-v1.tgz && tar xzf combined-private-install-
v1.tgz && cd combined-private-install-v1
sudo ./install.sh --hostname data.yourdomain.com --certbot you@yourdomain.com --allow HOST_IP
sudo reboot
```

Log in again. Pass 2 (about 5 minutes) installs the model, ClickHouse with the demo data, and the two nginx front doors, then checks them:

```
cd combined-private-install-v1 && sudo ./install.sh --hostname data.yourdomain.com --certbot
you@yourdomain.com --allow HOST_IP
```

It must end with `== Done` (not `WITH ERRORS`) after four checks: the model answers, the model is on the GPU, `entpep_host` reads `trade_blotter` (37 rows), and the admin login is refused (403).

7. Copy the data lake logins to the host (on your computer)

Two passwords move from data to host without being printed:

```
for n in host desk; do ssh $SSHK ubuntu@$D "sudo cat /etc/entpep-datalake/$n.pass" | ssh $SSHK
ubuntu@$H "umask 077; cat > ch-$n.pass"; done
```

8. Install the agent host

`ssh $SSHK ubuntu@$H`, then on the host (replace the two names and the email):

```
curl -sSf0 https://entpep.com/dl/entpep-install-v1.tgz && tar xzf entpep-install-v1.tgz && cd entpep-
install-v1
sudo ./install.sh --hostname host.yourdomain.com --certbot you@yourdomain.com --clickhouse-url
https://data.yourdomain.com:8443/ --clickhouse-user entpep_host --clickhouse-password-file ~/ch-
host.pass --clickhouse-desk-password-file ~/ch-desk.pass
shred -u ~/ch-host.pass ~/ch-desk.pass
```

It asks for an **Anthropic API key**: paste it, or press Enter if you'll use Bedrock or the private model (step 10). The summary should show:

- data lake: `https://data.yourdomain.com:8443/` answers as `entpep_host`
- Trial build: works until 2027-01-01
- Fleet Manager: `https://host.yourdomain.com/`

Sign in at `https://host.yourdomain.com/` with any name and the password from `sudo cat /opt/entpep/fleetcerts/fleet.pass`.

9. SharePoint (your Microsoft 365 admin, about 20 minutes)

The host signs in to Microsoft as its own Entra app with a certificate it made during the install. The app can reach **one SharePoint site and nothing else** (`Sites.Selected`). The full walkthrough is the **Microsoft setup** PDF; in short:

1. **Get the host's certificate** (public half only; the private key stays on the host). On your computer: `ssh $SSHK ubuntu@$H "cat /opt/entpep/sharepoint-app.cer" > sharepoint-app.cer`
2. **Create a SharePoint team site**, e.g. `EntPEP Trial`, private. In its **Documents** library create a folder `Barclays` with two folders inside it, `source` and `reports`, and upload the six demo files (`EntPEP-Trial-Demo-Files.zip` from `entpep.com/trial`, unzipped) into `Barclays/source`. Optional, for the demo's "check every file" step: also upload the zip's `Legal Agreements` folder (28 files, with Legal's clause library in `standards/`) as a top-level folder, with an empty `reports` folder inside it.
3. **Register an Entra app** (Entra → App registrations → New registration, single tenant, no redirect URI). Copy the **Application (client) ID** and **Directory (tenant) ID**. API permissions: remove `User.Read`, add

Microsoft Graph → **Application** → **Sites.Selected**, Grant admin consent.

4. **Upload the certificate** (sharepoint-app.cer) under the app's **Certificates & secrets** → **Certificates**. (Not under "Certificate authorities": that's tenant-wide trust for user sign-in.)
5. **Grant the app the one site** in Graph Explorer (<https://developer.microsoft.com/graph/graph-explorer>, signed in as the admin): GET `https://graph.microsoft.com/v1.0/sites/<tenant>.sharepoint.com/sites/<SiteAddress>` and copy the top-level `id`; then POST `https://graph.microsoft.com/v1.0/sites/<site id>/permissions` with `{"roles":["write"],"grantedToIdentities":[{"application":{"id":"<client id>","displayName":"EntPEP Trial"}}]}`. Expect 201 Created.
6. **On the host**, in entpep-install-v1: `sudo ./install.sh --entra-tenant TENANT_ID --entra-client CLIENT_ID --sharepoint-site "SITE_ID"` (the site ID has commas: keep the quotes). It should end with `SharePoint library: EntPEP Trial / Documents`.

10. Choose the model

In the Fleet Manager: **Admin** → **Model**. You can enable more than one and pick per agent at launch.

Anthropic API. The key you pasted during the install. To add or change it later, on the host: `umask 077; cat > ~/a.key` (paste, Enter, Ctrl-D), then `sudo ./install.sh --anthropic-key-file ~/a.key && shred -u ~/a.key`.

Claude on Amazon Bedrock (your AWS account; counts as an internal model). Once per account, open the Bedrock console's **Model access** page and complete Anthropic's use-case form. Then, on your computer, a user that may only call Claude, and an API key for it:

```
ACCT=$(aws sts get-caller-identity --query Account --output text); aws iam create-user --user-name entpep-bedrock
aws iam put-user-policy --user-name entpep-bedrock --policy-name invoke-claude-only --policy-document
{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Action":
["bedrock:InvokeModel", "bedrock:InvokeModelWithResponseStream"], "Resource":
["arn:aws:bedrock:*::foundation-model/anthropic.*", "arn:aws:bedrock:*:$ACCT:inference-
profile/*anthropic.*"]},
{"Effect": "Allow", "Action": "bedrock:CallWithBearerToken", "Resource": "*"}]}
aws iam create-service-specific-credential --user-name entpep-bedrock --service-name
bedrock.amazonaws.com --credential-age-days 90 --query
ServiceSpecificCredential.ServiceCredentialSecret --output text
```

The last line prints the key. In **Admin** → **Model** → **Amazon Bedrock**: region (e.g. us-east-2), model ID `us.anthropic.claude-sonnet-5`, the key, **Test & enable**.

Private model on data (nothing leaves your servers). On data: `sudo head -1 /etc/entpep-llm/keys`. In **Admin** → **Model** → **Private server**: host `data.yourdomain.com`, model `qwen3:14b`, the key, **Test & enable**. A 14B model is slower and less precise than Claude: the demo works, with simpler answers.

Keep confidential data internal (Admin, off by default): turn it on to show the policy. Agents that can reach the data lake are then labelled *confidential* and may use only Bedrock or the private model; an Anthropic-API agent with the data lake is refused at launch.

11. Run the demo

The **Demo script** PDF has the full script (about 10 minutes, nine steps, prompts to paste). Setup:

1. **Data Lake card:** hover **Connected**, click **manage**, **Grant 1h** on `trade_blotter` only.
2. **SharePoint card:** on, folder **Barclays**, **Read-write**, RAM workspace **512 MB**.
3. **Data Lake card:** dropdown **All desks**. **Advanced:** Sub-agents 1, Archive **New**.
4. **Launch Agent.** The agent opens in a new tab on port 8001 and lists the 6 files.

Then ask it what decision is waiting, pull the exposure by desk from the data lake (Rates is at 88.7% of its limit), show the Credit desk seeing only its own rows, show the blocked `comms_surveillance` query, start a sub-agent, write the memo to SharePoint, and open the **Data Lake** tab for the audit trail.

12. The trial

Trial builds of both binaries stop working on **1 January 2027 (UTC)**. In the last 30 days the Fleet Manager shows the days left; after the date it shows a notice, running agents stop and new ones can't start. A licensed build installs over the trial as a normal upgrade and keeps your settings and archives. To continue, use the form at <https://entpep.com/trial>.

13. Stop or remove

Stop both (compute billing stops; disks and addresses still cost a little): `aws ec2 stop-instances --region $REGION --instance-ids $HID $DID`. Start them again with `start-instances`; the Elastic IPs keep the names working.

Remove everything: `aws ec2 terminate-instances --region $REGION --instance-ids $HID $DID`, then once they're gone, `aws ec2 release-address --region $REGION --allocation-id $HALLOC`, `aws ec2 release-address --region $REGION --allocation-id $DALLOC`, `aws ec2 delete-security-group --region $REGION --group-id $HSG` and the same for `$DSG`. In Entra, delete the app registration; in SharePoint, delete the site.

Troubleshooting

Symptom	Fix
<code>run-instances: InsufficientInstanceCapacity</code> or <code>Unsupported</code>	That zone has no g6 or m8i right now: pick another <code>AZ</code> and <code>SUBNET</code> in step 1
<code>run-instances: VcpuLimitExceeded</code> for g6	GPU quota (step 1): request 4 or more vCPUs in Service Quotas
data: nothing installed after the first run	That was pass 1 (driver): <code>sudo reboot</code> , then run it again
certbot fails	The DNS record isn't there yet, is proxied, or port 80 is closed
host: <code>install: no /dev/kvm</code>	Nested virtualization is off: the fix under step 5
host: port 443 is in use	Another web server is on the host: remove it
Data Lake card: <code>grant failed: HTTP Error 403 ... nginx</code>	data doesn't allow the host's address: on data, <code>sudo entpep-datalake allow HOST_IP</code>
SharePoint: <code>AADSTS700027</code>	The certificate isn't on the app registration (step 9.4)
SharePoint: <code>HTTP Error 403</code>	The site grant (step 9.5) is missing or names another app or site
Agent stuck on "Starting your sealed agent..."	Open 8000-8999 on the host's security group from your address
Private model: Test fails	data's security group must allow 443 from the host's address

Questions, or a licensed build: <https://entpep.com/trial> (the form at the bottom).