

01 Prepare the host

FortiAI Gate 8.0.2 build 0055 | CPU lab | Ubuntu 22.04 amd64

Where: EC2 instance > Connect > Session Manager. These are commands for the Ubuntu instance. Paste one complete shaded block at a time, including its first and last lines. Each block starts Bash explicitly and works from the default Session Manager sh prompt.

Sizing The bundled installer guide targets 8+ CPU cores and 32 GB RAM. Use a host matching that baseline, such as t3a.2xlarge. The earlier t3a.large (2 vCPU / 8 GiB) is below it; a WARN does not establish that the stack will run reliably.

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
cat /etc/os-release
uname -m
nproc
free -h
df -h /
printf 'PASS: host checks completed; review sizing above.\n'
FORTI_SCRIPT
```

Copy and run safely

Use the companion TXT if copying from the PDF joins lines. Include the opening **/bin/bash** line and the closing **FORTI_SCRIPT** line. Press Enter once after the closing line if the shell is still waiting.

If a block fails A failed command stops that block and prints STOP. Resolve the error before continuing. The child Bash settings do not change the parent Session Manager shell.

The setup blocks check passwordless sudo before changing the host. Package installation disables interactive prompts; needrestart does not restart existing services automatically. The wizard has its own launcher so it can read your answers normally.

Using this public edition

Use your own Ubuntu instance and the staging bucket supplied for your lab. No instance, account or personal test results are included in this edition.

Before page 4 Replace **YOUR-STAGING-BUCKET** in the S3 source URI with your actual bucket name. The instance role must be permitted to read that bucket and release prefix. This guide does not grant access to software or licenses.

02 Repair the Docker repository

Restore a signed Docker APT source before installing packages.

For this lab's failed install, back up the two Docker source files, then replace the key and source together. The key/source writes are silent; wait for the final PASS and review the Candidate.

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
. /etc/os-release
[ "$ID" = ubuntu ]
[ "$VERSION_ID" = 22.04 ]
[ "$(dpkg --print-architecture)" = amd64 ]
printf 'Repairing Docker repository...\n'
forti_backup=$(sudo mktemp -d /etc/apt/docker-backup.XXXXXX)
for name in docker.list docker.sources; do
    path="/etc/apt/sources.list.d/$name"
    if [ -f "$path" ]; then sudo mv "$path" "$forti_backup/"; fi
done
sudo apt-get update
sudo env DEBIAN_FRONTEND=noninteractive NEEDRESTART_MODE=1 \
    apt-get install -y ca-certificates curl unzip
sudo install -m 0755 -d /etc/apt/keyrings
key=/etc/apt/keyrings/docker.asc
sudo curl -fsSL --retry 3 \
    https://download.docker.com/linux/ubuntu/gpg -o "$key"
sudo chmod 0644 "$key"
head -n 1 "$key"
printf '%s\n' \
    'Types: deb' \
    'URIs: https://download.docker.com/linux/ubuntu' \
    'Suites: jammy' \
    'Components: stable' \
    'Architectures: amd64' \
    'Signed-By: /etc/apt/keyrings/docker.asc' \
    | sudo tee /etc/apt/sources.list.d/docker.sources >/dev/null
sudo apt-get update
apt-cache policy docker-ce | sed -n '1,8p'
printf 'Source backup: %s\n' "$forti_backup"
printf 'PASS: repository refresh completed.\n'
FORTI_SCRIPT
```

Continue only if Ubuntu is 22.04 / x86_64, the key begins with BEGIN PGP PUBLIC KEY BLOCK, APT reports no signing errors, and docker-ce has a real Candidate (not none). If another source file defines Docker, resolve that duplicate first. [1]

03 Install Docker and AWS CLI

Complete both checks before downloading FortiAI Gate.

Install and verify Docker

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
printf 'Installing Docker Engine and Compose...\n'
sudo env DEBIAN_FRONTEND=noninteractive NEEDRESTART_MODE=1 \
  apt-get install -y docker-ce docker-ce-cli containerd.io \
  docker-buildx-plugin docker-compose-plugin
sudo systemctl enable --now docker
sudo docker version
sudo docker compose version
sudo docker run --rm hello-world
sudo systemctl is-active docker
forti_docker_root=$(sudo docker info --format '{{.DockerRootDir}}')
df -h "$forti_docker_root" /opt
printf 'PASS: Docker and Compose checks completed.\n'
FORTI_SCRIPT
```

Expected "Hello from Docker!", daemon active, and Compose 2.23.1 or newer. Keep at least 60 GB free for Docker (120 GB recommended) and 5 GB for assistant state (10 GB recommended). Continue using sudo for Docker; no group change is needed. [1, 4]

Install AWS CLI v2 on Ubuntu

CloudShell includes AWS CLI; an Ubuntu EC2 instance may not. This installs the official x86_64 package and uses its absolute path to avoid PATH errors. [2]

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
printf 'Downloading and installing AWS CLI v2...\n'
forti_cli_tmp=$(mktemp -d)
trap 'rm -rf -- "$forti_cli_tmp"' EXIT
curl -fsSL --retry 3 \
  https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip \
  -o "$forti_cli_tmp/awscliv2.zip"
unzip -q "$forti_cli_tmp/awscliv2.zip" -d "$forti_cli_tmp"
if [ -d /usr/local/aws-cli/v2 ]; then
  sudo "$forti_cli_tmp/aws/install" --update
else
  sudo "$forti_cli_tmp/aws/install"
fi
/usr/local/bin/aws --version
/usr/local/bin/aws sts get-caller-identity --region us-east-1 --no-cli-pager
printf 'PASS: AWS CLI v2 and EC2 identity verified.\n'
FORTI_SCRIPT
```

Expected aws-cli/2... and the intended AWS account / EC2 assumed role. Use the instance profile's temporary credentials; no aws configure or access keys are needed. The role needs ListBucket for the staging prefix and GetObject for its files. [3]

If a block fails: stop at the error. Do not paste the next block. Each block runs in a child Bash process. Its strict settings do not change your Session Manager shell; a failed block prints STOP and leaves the session open. Package prompts are disabled; needrestart does not restart existing services automatically.

04 Download and check the files

Your staging bucket | Region: us-east-1

The bucket must already contain **FortiAIGate_8.0.2_CPU/Images/** and **FortiAIGate_8.0.2_CPU/Installer/**. Names are case-sensitive. Replace **YOUR-STAGING-BUCKET** in the source URI below before running the block.

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
forti_source=s3://YOUR-STAGING-BUCKET/FortiAIGate_8.0.2_CPU/
/usr/local/bin/aws s3 ls "$forti_source" --recursive --region us-east-1
sudo install -d -m 0755 -o "$(id -un)" -g "$(id -gn)" /opt/fortiaigate
/usr/local/bin/aws s3 sync "$forti_source" /opt/fortiaigate/ \
  --exclude "*" --include "Images/*.tar" --include "Installer/*.tar.gz" \
  --region us-east-1
cd /opt/fortiaigate
forti_build=V8.0.2-build0055-FORTINET
for part in api webui llm-gateway guardrail guardrail-scanner \
  license_manager logd; do
  file="Images/FAIG_${part}-${forti_build}.tar"
  printf 'Checking %s\n' "$file"
  test -s "$file"
  tar -tf "$file" >/dev/null
done
test -s "Installer/FAIG_deploy-docker-compose-cli-${forti_build}.tar.gz"
printf 'PASS: seven CPU images and the installer are present.\n'
FORTI_SCRIPT
```

Check Expect seven image TARs plus one installer TAR.GZ. Hidden .DS_Store files are excluded from the download. A repeat sync may copy nothing when files already match; that is normal. Image TARs are listed for validation, not extracted. [3, 4]

Extract only the Deployment Assistant

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
cd /opt/fortiaigate/Installer
forti_installer=FAIG_deploy-docker-compose-cli-V8.0.2-build0055-FORTINET.tar.gz
tar -tzf "$forti_installer"
tar -xzf "$forti_installer" deploy-docker-compose.sh USER-GUIDE.md
chmod 755 deploy-docker-compose.sh
bash -n deploy-docker-compose.sh
./deploy-docker-compose.sh --self-test
FORTI_SCRIPT
```

Continue only if The archive lists deploy-docker-compose.sh and USER-GUIDE.md, and the self-test ends with Self-test passed. Run the assistant as your non-root SSM user; it invokes sudo when required. [4]

AccessDenied? Verify the attached instance profile and its S3 ListBucket/GetObject permissions. For objects encrypted with a customer-managed KMS key, check kms:Decrypt permission too.

05 Configure and deploy

Settings below match the bundled Deployment Assistant 0.6.3. [4]

```
/bin/bash -c '
cd /opt/fortiaigate/Installer &&
exec /bin/bash ./deploy-docker-compose.sh
'
```

This launcher keeps terminal input available for the wizard. Resolve every prerequisite **FAIL** before continuing. CPU/RAM warnings should be addressed using the sizing on page 1.

Wizard prompt	Selection
Image source	1 - Release archive folder
Image directory	/opt/fortiaigate/Images
Target release	Select the detected 8.0.2-build0055 entry by number
HTTPS / HTTP redirect	443 / 8081 ; loopback-only redirect: 1 = Yes
GPU content inspection	2 = No
Gateway memory / log budget	Enter 4 GB / 8 GB explicitly
Deployment namespace	fortiaigate
FortiCare proxy	2 = No when direct Fortinet HTTPS access works

Import the images

Main menu **5 > 1**: import and verify target images. Expect **Imported 7 image archive(s)** and PASS for the seven CPU components. Choose **7** to return.

Deploy the stack

Main menu **3**: deploy or reapply Compose. Confirm the target release and ports, choose **1 = Yes** for the recovery archive, record its protected path, and type **APPLY**.

The assistant also needs Docker Hub access for nginx, PostgreSQL and Redis, plus outbound HTTPS to **fds1.fortinet.com**. Allow its startup checks to finish.

Ready to verify Generated configuration should be ready and Compose validation should pass. Main menu 4 > 1 opens status and health. A logd-api-ready helper that finishes with Exited (0) is normal.

CPU mode Management and regular-expression guardrails are available. Triton DLP/model inspection and vLLM natural-language policy evaluation require the GPU deployment. Verify a real deny test before assuming inspection is enforced.

06 Verify and finish setup

Use the current instance public IPv4 for browser access.

Check containers and the local WebUI

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
cd /opt/fortiaigate/Installer/faig_deployment_assistant_data/deployment
sudo ./compose.sh ps -a
forti_http=$(curl -ksS --max-time 15 -o /dev/null -w '%{http_code}' \
  https://127.0.0.1/ui)
printf 'WebUI HTTP %s\n' "$forti_http"
test "$forti_http" = 200
printf 'PASS: local WebUI returned HTTP 200.\n'
FORTI_SCRIPT
```

Expected PostgreSQL and Redis are healthy; application containers remain up without restart loops; /ui returns HTTP 200. The local curl check permits the installer's self-signed certificate. For failures use assistant 5 > 3, or compose.sh logs --tail 200.

Onboard and apply the license

1. Verify the instance security group allows HTTPS 443 from your current public IP /32. Open **https://<public-ip>/ui** on your computer; verify the instance before accepting its self-signed certificate.
2. Create the administrator and organization. Set the external URL to **https://<public-ip>** without /ui. Upload the valid FortiAIGate .lic under **Settings > License** and verify Active.
3. Back in EC2, restart the API, then recheck readiness in assistant **4 > 1**.

```
/bin/bash <<'FORTI_SCRIPT' || printf 'STOP: block failed.\n' >&2
set -euo pipefail
sudo -n true
cd /opt/fortiaigate/Installer/faig_deployment_assistant_data/deployment
sudo ./compose.sh restart api
sudo ./compose.sh ps -a
printf 'PASS: API restart command completed; review status.\n'
FORTI_SCRIPT
```

Acceptance: verify login, Active license and a normal gateway request. If using CPU regex protection, submit a synthetic matching test string and confirm the request is denied and logged. Keep generated state and the recovery archive secure.

If a check fails

NO_PUBKEY / no candidate: repeat page 2. Keep APT signature verification enabled.

aws not found / dpkg permission denied: use /usr/local/bin/aws; package installs need sudo.

Joined commands / unexpected > prompt: Ctrl+C, then copy the complete block from the TXT, including /bin/bash and FORTI_SCRIPT.

WebUI 502 / restarts: review assistant health/logs and CPU/RAM sizing before proceeding.

Sources and validation

[1] [Docker Engine on Ubuntu](#). [2] [AWS CLI v2 installation](#). [3] [EC2 role credentials / S3 sync](#).

[4] USER-GUIDE.md and deploy-docker-compose.sh from the local 8.0.2 build0055 installer archive, CLI 0.6.3. Sizing also checked against [AWS T3a specifications](#).

Public edition of the v1.3 workflow. Shell syntax and PDF extraction verified; run the checks on your own host.