

# Guida Sicura alla Gestione di File .env su Server Fisici e VM

## Struttura consigliata (2 app di esempio)

/var/www/app1/ # CODICE (repo)

■■ server.js

■■ package.json

■■ ...

/var/www/app2/ # CODICE (repo)

■■ index.js

■■ package.json

■■ ...

/etc/app1/ # CONFIG (fuori dal repo)

■■ .env # default (es. staging)

■■ .env.production # produzione

/etc/app2/

■■ .env

■■ .env.production

## Permessi e sicurezza

```
sudo mkdir -p /etc/app1 /etc/app2
```

```
sudo touch /etc/app1/.env /etc/app1/.env.production
```

```
sudo touch /etc/app2/.env /etc/app2/.env.production
```

```
sudo chown -R appuser:appuser /etc/app1 /etc/app2
```

```
sudo chmod 600 /etc/app1/.env /etc/app1/.env.production
```

```
sudo chmod 600 /etc/app2/.env /etc/app2/.env.production
```

```
sudo chmod 700 /etc/app1 /etc/app2
```

## Esempio file .env.production

```
/etc/app1/.env.production
```

```
NODE_ENV=production
```

```
HOST=0.0.0.0
```

```
PORT=4000
DB_HOST=127.0.0.1
DB_USER=app1
DB_PASSWORD=***molto-segreta***
JWT_SECRET=***lunghissima-random***

/etc/app2/.env.production
NODE_ENV=production
HOST=0.0.0.0
PORT=5000
REDIS_URL=redis://127.0.0.1:6379/0
API_BASE=https://api.example.com
```

## Caricamento nel codice

ESM (server.js):

```
import dotenv from "dotenv";
import fs from "fs";
const ENV_PATH = process.env.DOTENV_CONFIG_PATH || "/etc/app1/.env";
if (!fs.existsSync(ENV_PATH)) {
  if (process.env.NODE_ENV === "production") {
    throw new Error(`File .env mancante: ${ENV_PATH}`);
  }
} else {
  dotenv.config({ path: ENV_PATH, override: false });
}
```

## PM2

ecosystem.config.js:

```
module.exports = {
  apps: [
    {
      name: "app1",
      script: "server.js",
      cwd: "/var/www/app1",
      node_args: "-r dotenv/config",
```

```

env_production: {
  NODE_ENV: "production",
  DOTENV_CONFIG_PATH: "/etc/app1/.env.production"
},
{
  name: "app2",
  script: "index.js",
  cwd: "/var/www/app2",
  node_args: "-r dotenv/config",
  env_production: {
    NODE_ENV: "production",
    DOTENV_CONFIG_PATH: "/etc/app2/.env.production"
  }
}
]
}

```

Avvio:

```
pm2 start ecosystem.config.js --env production
```

```
pm2 save
```

## systemd

```
/etc/systemd/system/app1.service
```

```
[Unit]
```

```
Description=App1 Node.js
```

```
After=network.target
```

```
[Service]
```

```
User=appuser
```

```
Group=appuser
```

```
WorkingDirectory=/var/www/app1
```

```
ExecStart=/usr/bin/node /var/www/app1/server.js
```

```
EnvironmentFile=/etc/app1/app.env
```

```
Restart=always
```

```
RestartSec=3
```

```
NoNewPrivileges=true
```

```
PrivateTmp=true
```

ProtectSystem=full

ProtectHome=true

[Install]

WantedBy=multi-user.target

## Ambienti multipli (symlink)

```
sudo ln -sf /etc/app1/.env.production /etc/app1/.env.current
```

```
pm2 restart app1
```

## Backup sicuro

```
cd /etc
```

```
sudo tar cz app1 app2 | gpg -c > env-backup-$(date +%F).tar.gz.gpg
```

```
gpg -d env-backup-2025-09-25.tar.gz.gpg | sudo tar xz -C /
```

## Checklist

- .env **\*\*fuori\*\*** dal repo (/etc/.env\*)
- Proprietario = utente dell'app, chmod 600
- Niente segreti nei log
- In dev: .env nel progetto ok; in prod: /etc/...
- PM2: usa -r dotenv/config + DOTENV\_CONFIG\_PATH o systemd EnvironmentFile=
- .sample-env nel repo (senza valori) per documentare
- Backup cifrato dei file in /etc/