**0) Co będziesz robić (w skrócie)**

1. Na komputerze **z internetem** zbudujesz package-registry.exe i pobierzesz paczki integracji (ZIP). ([GitHub](https://github.com/elastic/package-registry))
2. Skopiujesz to na **Windows Server offline**.
3. Uruchomisz EPR z **lokalnego katalogu paczek** i wskażesz go w Kibanie (xpack.fleet.registryUrl, isAirGapped: true). ([Elastic](https://www.elastic.co/docs/reference/fleet/air-gapped))

**1) Budowa binarki i zebranie paczek (na Windows z internetem)**

**1.1 Zbuduj package-registry.exe**

W PowerShell (uruchom jako zwykły user):

# Wymagane: Go (wg wersji w pliku .go-version repo), Git

git clone https://github.com/elastic/package-registry.git

cd package-registry

# Zainstaluj Mage (narzędzie build):

go install github.com/magefile/mage@latest

$env:PATH += ";$([Environment]::GetFolderPath('UserProfile'))\go\bin"

# Zbuduj single-binary:

mage build

# => powstanie binarka: .\package-registry.exe

Repo EPR wspiera build „single binary” przez mage build. ([GitHub](https://github.com/elastic/package-registry))

**1.2 Pobierz paczki integracji (ZIP)**

EPR udostępnia API /search (lista) i /epr/{name}/{name}-{version}.zip (konkretny ZIP). Przykład w PowerShell:

# Sprawdź dostępne wersje paczki "system":

Invoke-WebRequest "https://epr.elastic.co/search?package=system&all=true" -OutFile system.json

# Pobierz konkretną wersję ZIP:

Invoke-WebRequest "https://epr.elastic.co/epr/system/system-1.53.0.zip" -OutFile "system-1.53.0.zip"

Endpointy i wzorce URL są w README EPR. ([GitHub](https://github.com/elastic/package-registry))

Tip: filtrowanie po wersji Kibany: dodaj &kibana.version=9.1.2 do /search, by pobrać zgodne paczki. ([GitHub](https://github.com/elastic/package-registry))

**1.3 Przenieś offline**

Skopiuj na pendrive / dysk:

* package-registry.exe
* wszystkie zebrane \*.zip paczek

**2) Windows Server (OFFLINE) – katalogi i konfiguracja**

**2.1 Utwórz strukturę katalogów**

New-Item -Type Directory C:\EPR\bin, C:\EPR\packages, C:\EPR\config | Out-Null

Copy-Item X:\package-registry.exe C:\EPR\bin\

Copy-Item X:\\*.zip C:\EPR\packages\

**2.2 config.yml (EPR czyta paczki z dysku)**

C:\EPR\config\config.yml:

server:

address: "0.0.0.0:8080" # nasłuch HTTP

storage:

filesystem:

paths:

- "C:/EPR/packages" # używaj / w ścieżkach

cache:

max\_age: "1h" # opcjonalnie: nagłówki cache

EPR domyślnie ładuje konfigurację z config.yml i ma „File system indexer” do folderów/ZIP-ów. ([GitHub](https://github.com/elastic/package-registry))

**3) Uruchomienie i test**

**3.1 Start (tryb „ręczny”)**

# Ustaw adres jako zmienna (bez zgadywania nazwy flagi):

$env:EPR\_ADDRESS = "0.0.0.0:8080"

# (opcjonalnie metryki Prometheus na 9000)

# $env:EPR\_METRICS\_ADDRESS = "0.0.0.0:9000"

& "C:\EPR\bin\package-registry.exe" -log-level info -config "C:\EPR\config\config.yml"

* **Healthcheck**:

(Invoke-WebRequest "http://127.0.0.1:8080/health").StatusCode

# oczekiwane: 200

* (opc.) metryki: http://<host>:9000/metrics po ustawieniu EPR\_METRICS\_ADDRESS.  
  /health, /metrics i env-y są opisane w README EPR. ([GitHub](https://github.com/elastic/package-registry))

**3.2 Jako usługa Windows (auto-start)**

Najprościej „sc.exe”:

sc.exe create EPR binPath= "\"C:\EPR\bin\package-registry.exe\" -log-level info -config \"C:\EPR\config\config.yml\"" start= auto DisplayName= "Elastic Package Registry"

sc.exe start EPR

Uwaga: w sc.exe **musi** być spacja po = (to nie błąd w instrukcji).

**3.3 Zapora (jeśli włączona)**

netsh advfirewall firewall add rule name="EPR 8080" dir=in action=allow protocol=TCP localport=8080

**4) Podłączenie Kibany/Fleet do Twojego EPR (OFFLINE)**

Edytuj kibana.yml (na serwerze z Kibaną):

xpack.fleet.isAirGapped: true

xpack.fleet.registryUrl: "http://twoj-epr:8080"

To oficjalny sposób dla środowisk air-gapped. ([Elastic](https://www.elastic.co/docs/reference/fleet/air-gapped" \o "Air-gapped environments | Elastic Docs))

**4.1 Własne CA / self-signed TLS (opcjonalnie)**

Jeśli wystawisz EPR na HTTPS z własnym certyfikatem, dodaj w plikach startowych Kibany:

NODE\_EXTRA\_CA\_CERTS="C:\kibana\certs\ca-cert.pem"

Instrukcja użycia custom CA dla Kibany jest w dokumentacji air-gapped. ([Elastic](https://www.elastic.co/docs/reference/fleet/air-gapped))

**5) (Ważne w offline) Binarne artefakty dla aktualizacji Agenta**

EPR to **tylko paczki integracji**. Do aktualizacji agentów potrzebny jest lokalny **Artifact Registry** i ustawienie go w Fleet:

1. Zgromadź pliki z https://artifacts.elastic.co/downloads/… (dla Windows: warianty windows\_x86\_64 + pliki .sha512 i .asc) na własnym serwerze HTTP w sieci.
2. W Kibanie: **Fleet → Settings → Agent Binary Download → Add agent binary source** i ustaw swój serwer jako domyślny. ([Elastic](https://www.elastic.co/docs/reference/fleet/air-gapped))

**6) Szybka diagnostyka**

* Invoke-WebRequest http://<host>:8080/health → **200** (EPR gotowy). ([GitHub](https://github.com/elastic/package-registry))
* http://<host>:8080/search?package=system&all=true zwraca JSON (widać paczki). ([GitHub](https://github.com/elastic/package-registry))
* W Kibanie: Integrations się ładują z Twojego EPR po ustawieniu registryUrl i isAirGapped. ([Elastic](https://www.elastic.co/docs/reference/fleet/air-gapped))

Chcesz, żebym dorzucił gotowe pliki (config.yml, skrypt PS do pobierania ZIP-ów po kibana.version, oraz komendę do włączenia TLS na porcie 443 z env-ami EPR\_TLS\_CERT/KEY)? ([Elastic](https://www.elastic.co/docs/reference/fleet/air-gapped), [GitHub](https://github.com/elastic/package-registry))