Building the Grafana open-source project on a local Windows machine in an offline environment involves several steps, including setting up the development environment, downloading necessary dependencies and tools on an internet-connected machine, and then transferring them to the offline machine. Here's a step-by-step guide:

**Step 1: Prepare the Environment**

1. **Select a Windows Machine:**
   * Choose a Windows machine that will be used to build the Grafana project. Ensure it has the necessary specifications for development.
2. **Install Git:**
   * Download Git for Windows from the [Git official site](https://git-scm.com/download/win) on an internet-connected machine and transfer the installer to your offline machine.
   * Install Git on your offline Windows machine.
3. **Install Node.js and npm:**
   * Download Node.js from the Node.js official site on an internet-connected machine and transfer the installer to your offline machine.
   * Install Node.js on your offline Windows machine. npm (Node Package Manager) is included with Node.js.
4. **Install Yarn:**
   * Download the Yarn installation package from the Yarn official site on an internet-connected machine and transfer it to your offline machine.
   * Install Yarn by running the installer.
5. **Install Go:**
   * Download the Go programming language from the [Go official site](https://golang.org/dl/) on an internet-connected machine and transfer the installer to your offline machine.
   * Install Go on your offline Windows machine.
6. **Download Grafana Source Code:**
   * Clone the Grafana GitHub repository on an internet-connected machine:

bash

Copy code

git clone https://github.com/grafana/grafana.git

* + Transfer the cloned repository to your offline machine.

**Step 2: Prepare Dependencies Offline**

1. **Download Node.js Dependencies:**
   * On the internet-connected machine, navigate to the Grafana repository directory.
   * Run yarn to install all dependencies:

bash

Copy code

cd grafana

yarn install

* + After installation, create an offline package for the dependencies:

bash

Copy code

yarn cache list --pattern grafana > yarn-packages.json

yarn cache dir > yarn-cache-dir.txt

* + Compress the Yarn cache directory and the yarn-packages.json file:

bash

Copy code

zip -r yarn-cache.zip `cat yarn-cache-dir.txt`

1. **Download Go Dependencies:**
   * On the internet-connected machine, navigate to the Grafana repository directory.
   * Download all Go dependencies:

bash

Copy code

go mod download

* + Create a vendor directory:

bash

Copy code

go mod vendor

* + Compress the vendor directory:

bash

Copy code

zip -r go-vendor.zip vendor

1. **Transfer Dependencies to Offline Machine:**
   * Transfer the yarn-cache.zip, yarn-packages.json, and go-vendor.zip files to your offline Windows machine.
   * Unzip these files into the appropriate directories:

bash

Copy code

unzip yarn-cache.zip -d yarn-cache

unzip go-vendor.zip -d vendor

**Step 3: Configure Grafana Build Environment**

1. **Configure Yarn Offline Cache:**
   * Configure Yarn to use the offline cache directory:

bash

Copy code

yarn config set yarn-offline-mirror C:\path\to\yarn-cache

1. **Install Node.js Dependencies Offline:**
   * Navigate to the Grafana project directory and install dependencies using the offline cache:

bash

Copy code

yarn install --offline

1. **Install Go Dependencies Offline:**
   * Set the Go environment to use the vendor directory:

bash

Copy code

set GOFLAGS=-mod=vendor

**Step 4: Build Grafana**

1. **Build Frontend:**
   * Navigate to the Grafana project directory and build the frontend assets:

bash

Copy code

yarn build

1. **Build Backend:**
   * Navigate to the Grafana project directory and build the backend:

bash

Copy code

go run build.go build

**Step 5: Run Grafana**

1. **Run Grafana Server:**
   * Navigate to the Grafana project directory and run the Grafana server:

bash

Copy code

bin\grafana-server.exe

1. **Access Grafana:**
   * Open a web browser and navigate to http://localhost:3000.
   * Log in with the default credentials (username: admin, password: admin).

**Summary**

By following these steps, you should be able to build the Grafana open-source project on a local Windows machine in an offline environment. This process involves preparing the environment, downloading dependencies on an internet-connected machine, transferring them to the offline machine, and configuring the build environment before finally building and running Grafana.