

# ????

[mirror.nju.edu.cn](https://mirror.nju.edu.cn)

- [□□](#)
- [GCR](#)
- [GHCR](#)
- [GLCR](#)
- [NGC](#)
- [Quay](#)
- [K8s](#)
- [Go](#)
- [Maven](#)
- [npm](#)

??

□□

- mirror.nju.edu.cn
- mirrors.nju.edu.cn

□□

□□	□□
HTTP	80
HTTPS	443
RSYNC	873

# GCR

Google Container Registry

## gcr.io

☐ `gcr.io` ☐ `gcr.nju.edu.cn` ☐

☐

```
docker pull gcr.io/cadvisor/cadvisor:v0.39.3
```

☐

```
docker pull gcr.nju.edu.cn/cadvisor/cadvisor:v0.39.3
```

## k8s.gcr.io

☐ `k8s.gcr.io` ☐ `gcr.nju.edu.cn/google-containers` ☐

☐

```
docker pull k8s.gcr.io/kube-proxy:v1.18.6
```

☐

```
docker pull gcr.nju.edu.cn/google-containers/kube-proxy:v1.18.6
```

# GHCR

Github Container Registry

ghcr.io

☐ ghcr.io ☐

☐ ghcr.nju.edu.cn ☐

☐

```
docker pull ghcr.io/github/super-linter:latest
```

☐

```
docker pull ghcr.nju.edu.cn/github/super-linter:latest
```

# GLCR

GitLab Container Registry

## registry.gitlab.com

 `registry.gitlab.com`  `glcr.nju.edu.cn` 



```
docker pull registry.gitlab.com/pipeline-components/yamllint
```




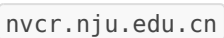

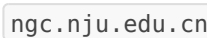

 

```
docker pull glcr.nju.edu.cn/pipeline-components/yamllint
```

# NGC

NVIDIA NGC Catalog

## nvcr.io



```
docker pull nvcr.io/nvidia/cuda
```




 

```
docker pull nvcr.nju.edu.cn/nvidia/cuda
```

# Quay

Quay Container Registry

## quay.io

 `quay.io`  `quay.nju.edu.cn` 



```
docker pull quay.io/prometheus/prometheus:latest
```



```
docker pull quay.nju.edu.cn/prometheus/prometheus:latest
```

# K8s

Kubernetes's container image registry

## registry.k8s.io

registry.k8s.io k8s.nju.edu.cn

```
docker pull registry.k8s.io/busybox
```

```
docker pull k8s.nju.edu.cn/busybox
```



# Go

1. 安装 Go 语言环境

2. 配置 Go 语言环境

```
go env -w GOMODCACHE=on # 设置 Go Modules 缓存
go env -w GOPROXY="https://repo.nju.edu.cn/go/,direct" # 设置代理
go env | grep GOPROXY # linux 查看

go env | findstr "GOPROXY" # windows 查看
```

3. 配置 Go 语言环境

```
go env -w GOPRIVATE="*.nju.edu.cn"
```

# Maven

???? maven central + gradle-plugin ????? gradle ? maven  
????????????????????

## maven ????

?????

???????????????????? maven ?????????????????

- linux ?????????? :

```
mkdir ~/.m2
vim ~/.m2/settings.xml
```

- windows ??? powershell (?? pwsh ) ?????

```
mkdir ~/.m2
cd ~/.m2
notepad settings.xml
```

????????????????

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <mirrors>
    <mirror>
      <id>nju_mirror</id>
      <mirrorOf>*</mirrorOf>
      <url>https://repo.nju.edu.cn/maven/</url>
    </mirror>
  </mirrors>
</settings>
```

XXXXXXXXXX

mirrors XXXXXXXX

(XX mirrorOf XXXXXXXXXXXXXXXX

)X

```
<mirror>
  <id>nju_mirror</id>
  <mirrorOf>*</mirrorOf>
  <url>https://repo.nju.edu.cn/maven/</url>
</mirror>
```

XXXXXXXXXX

?????

XXXXXXXXXXXXXXXXXX

pom.xml XXXX

```
<project>
  .....[XXXXX]

  <repositories>
    <repository>
      <id>nju</id>
      <url>https://repo.nju.edu.cn/maven/</url>
      <releases>
        <enabled>true</enabled>
      </releases>
      <snapshots>
        <enabled>false</enabled>
      </snapshots>
    </repository>
  </repositories>
</project>
```

# gradle-plugin ?????

XXXXXXXXXX

gradle XXXXXXXXXXXXXXXX

gradle\wapper\gradle-

wrapper.properties X distributionUrl XXX http://mirror.nju.edu.cn/gradle/gradle-[version].zip

XXXXXXXXXX

XXXX\_XX

?????

•

•

- ```
mkdir ~/.gradle
vim ~/.gradle/init.gradle
```

- ```
mkdir ~/.gradle
cd ~/.gradle
notepad init.gradle
```

```
allprojects {
    buildscript {
        repositories {
            def NJU_REPOSITORY_URL = 'https://repo.nju.edu.cn/maven/'
            all { ArtifactRepository repo ->
                if (repo instanceof MavenArtifactRepository) {
                    def url = repo.url.toString()
                    if (url.startsWith('https://repo1.maven.org/maven2')) {
                        project.logger.lifecycle "Repository ${repo.url} replaced by
$NJU_REPOSITORY_URL."

                        remove repo
                    }
                    if (url.startsWith('https://jcenter.bintray.com/')) {
                        project.logger.lifecycle "Repository ${repo.url} deleted."
                        remove repo
                    }
                    if (url.startsWith('https://dl.google.com/dl/android/maven2/')) {
                        project.logger.lifecycle "Repository ${repo.url} deleted."
                        remove repo
                    }
                    if (url.contains('plugins.gradle.org/m2')) {
                        project.logger.lifecycle "Repository ${repo.url} deleted."
                        remove repo
                    }
                }
            }
        }
    }
}
```

```

        maven { url NJU_REPOSITORY_URL }
        mavenLocal()
    }
}

repositories {
    def NJU_REPOSITORY_URL = 'https://repo.nju.edu.cn/maven/'
    all { ArtifactRepository repo ->
        if (repo instanceof MavenArtifactRepository) {
            def url = repo.url.toString()
            if (url.startsWith('https://repo1.maven.org/maven2')) {
                project.logger.lifecycle "Repository ${repo.url} replaced by
$NJU_REPOSITORY_URL."
                remove repo
            }
            if (url.startsWith('https://jcenter.bintray.com/')) {
                project.logger.lifecycle "Repository ${repo.url} deleted."
                remove repo
            }
            if (url.startsWith('https://dl.google.com/dl/android/maven2/')) {
                project.logger.lifecycle "Repository ${repo.url} deleted."
                remove repo
            }
            if (url.contains('plugins.gradle.org/m2')) {
                project.logger.lifecycle "Repository ${repo.url} deleted."
                remove repo
            }
        }
    }
}

maven { url NJU_REPOSITORY_URL }
mavenLocal()
}
}

```

■■■■■■

gradle ■■■■■■

gradle --stop ■■■■■

gradle ■ daemon■

■■■■■■■■

????

■■■■■■■■■■■■■■■■■■■■

build.gradle ■■■ :

```
buildscript {  
    repositories {  
        maven { url 'https://repo.nju.edu.cn/maven/' }  
    }  
}  
  
plugins {  
    ...[com.gradle:gradle-enterprise-gradle-plugin:  
}  
  
allprojects {  
    repositories {  
        maven { url 'https://repo.nju.edu.cn/maven/' }  
    }  
}
```

com.gradle:gradle-enterprise-gradle-plugin:

gradle-enterprise-gradle-plugin

# npm

## Node Package Manager

# npm ?????

???

```
npm install [packageName] --registry=https://repo.nju.edu.cn/repository/npm/
```

[illegible]

??????

[illegible]







```
npm config set registry https://repo.nju.edu.cn/repository/npm/ # 配置
```







# npm ??

??????

□□□□□□□□
https://repo.nju.edu.cn/repository/npm-nju/
□□



- 
  - 
  - 
- 
  - 
  - 

pypi 
- 
  - 
  - 
  - 
  - 

- 1. 注册 npm 用户

## ???????

1. 注册 npm 用户

```
npm adduser --registry https://repo.nju.edu.cn/repository/npm-releases/
```

```
npm adduser --registry https://repo.nju.edu.cn/repository/npm-releases/
```

2. 配置 package.json

```
"publishConfig": {  
  "registry" : "https://repo.nju.edu.cn/repository/npm-releases/"  
},
```

3. 配置 package.json

4. 配置 .npmignore

5. 构建并发布

```
npm run build  
npm publish
```

## yarn ????

```
yarn config set registry https://repo.nju.edu.cn/repository/npm/ # 配置
```

yarn 安装

npm 安装

yarn login

yarn publish