## ✅ 1. Artifact Handling in Jenkins

### 🔍 What Are Artifacts?

Artifacts are files produced as a result of a build, such as:

* .war (Web Application Archive)
* .jar (Java Archive)
* .zip, .ear, .apk, etc.

### 📁 Where Are They Stored?

When you build a Maven project in Jenkins, the output artifact is usually created in:

/var/lib/jenkins/workspace/<job\_name>/target/\*.war

### 📦 Archiving Artifacts in Jenkins:

Jenkins can store artifacts after a successful build.

Steps:

1. Go to Jenkins → Your Job → Configure.
2. Scroll to Post-build Actions.
3. Click Add post-build action → Archive the artifacts.
4. In the field, enter: target/\*.war

Purpose:

* The archived artifacts appear in the "Build Artifacts" section of each Jenkins build.
* It enables manual download or integration with deployment tools.

## 🚀 2. Uploading Artifacts to JFrog Artifactory

### 📌 Why Use JFrog?

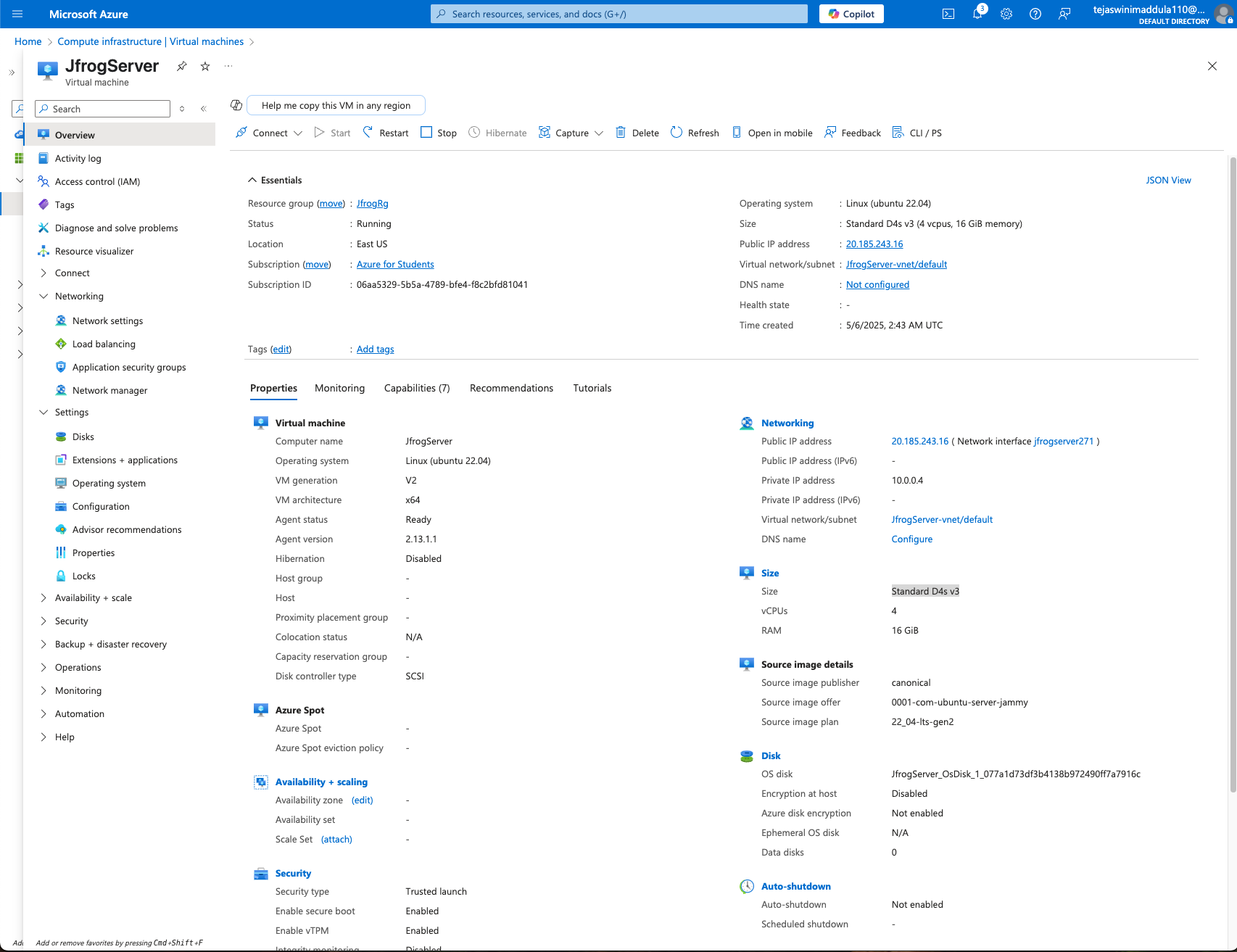
JFrog Artifactory acts as a universal binary repository manager:

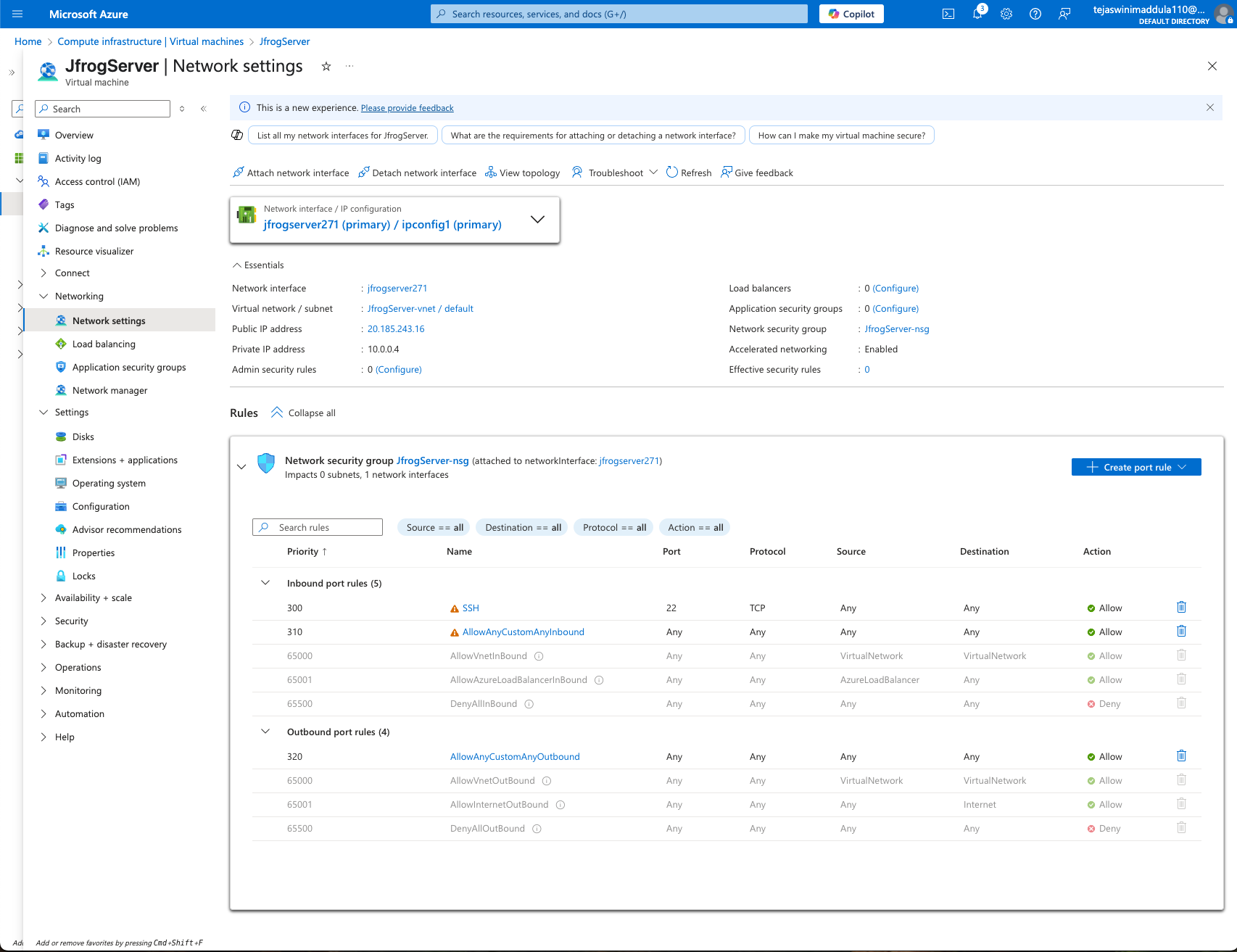
* Stores versioned builds (.war, .jar, etc.)
* Provides metadata and checksum verification
* Facilitates secure and traceable deployments

### 🖥️ Artifactory OSS 6.9.6 Installation on Ubuntu

#### ✅ Pre-Requisites:

* OS: Linux (ubuntu 22.04)
* Size: Standard D4s v3
* vCPUs: 4
* RAM: 16 GiB
* Ports Opened: 8081 (UI), 8082 (Optional), 22 (SSH)



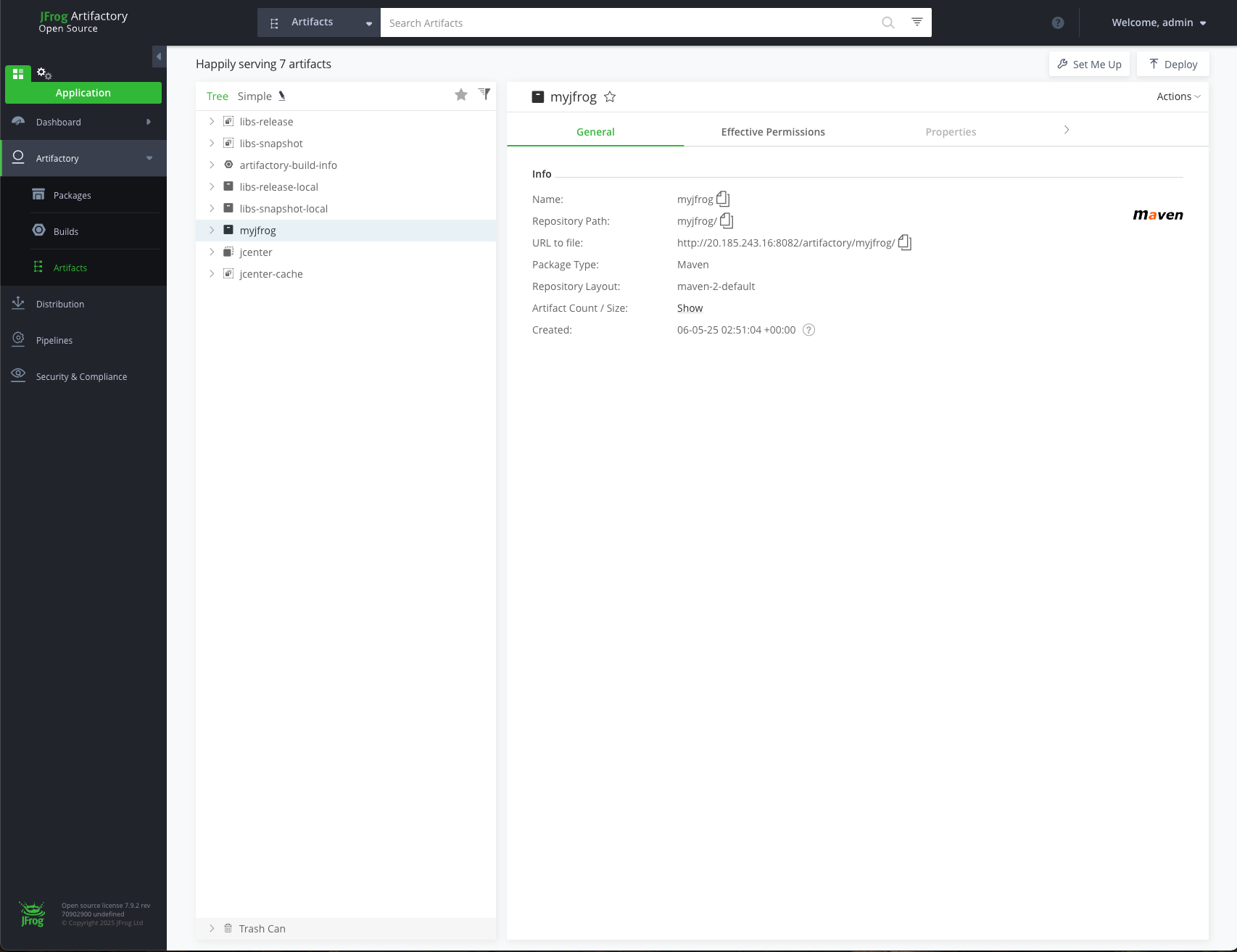


### Username: azureuser

Password: 493Rehamansk.

### 🧱 Create Maven Repositories in JFrog

1. Go to: Admin → Repositories → Local
2. Create:  
   * myjfrog → Type: Maven → For releases
   * myjfrog → Type: Maven → For snapshot builds



Repositories follow naming conventions and provide version control.

## 🔗 3. Integrating Jenkins with JFrog Artifactory

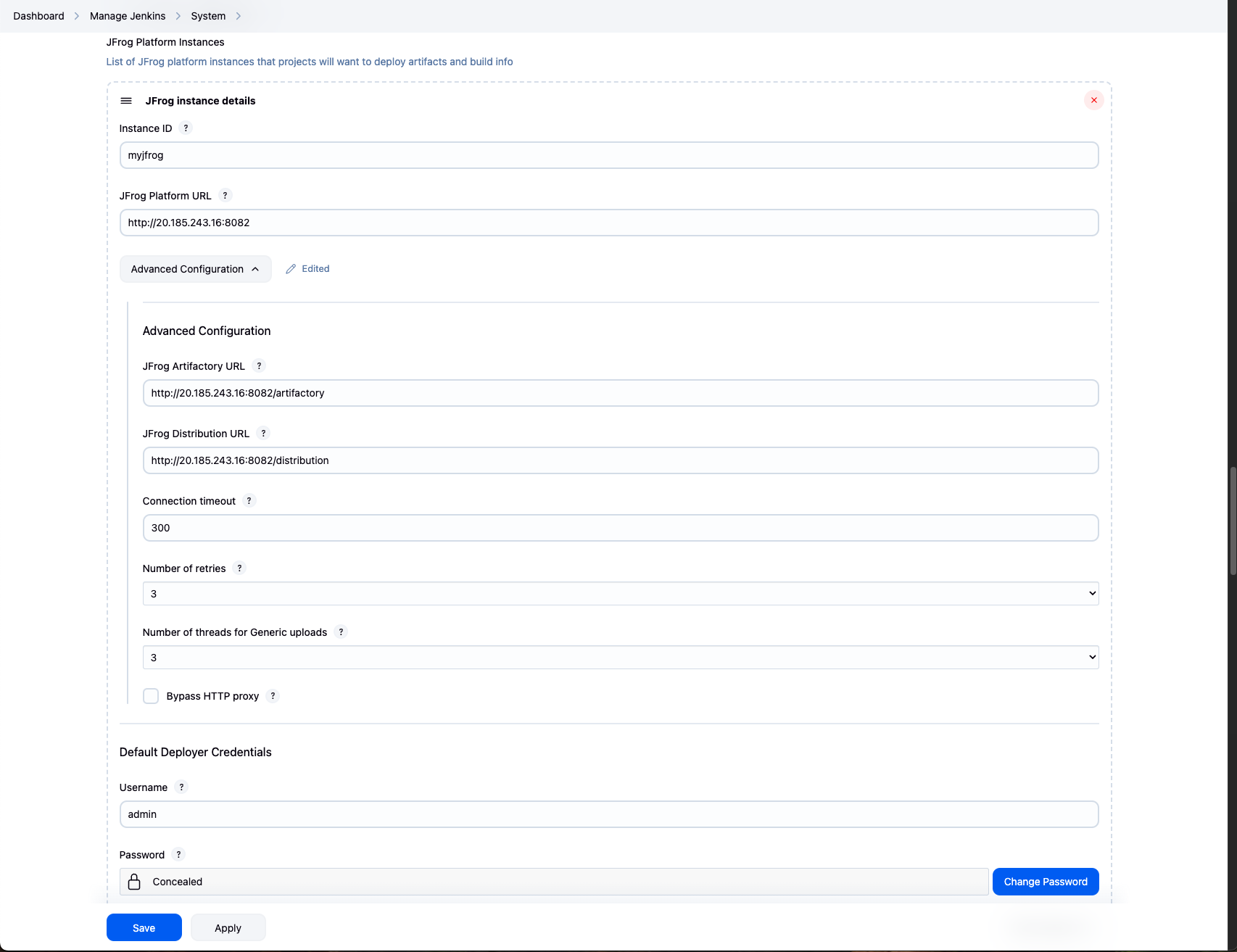
### 🛠️ Install Plugin:

* Go to: Manage Jenkins → Manage Plugins
* Search for: Artifactory Plugin
* Install and restart Jenkins.

### 🔐 Configure Artifactory Server:

Steps:

1. Go to: Manage Jenkins → Configure System
2. Scroll to JFrog Artifactory
3. Click Add Artifactory Server  
   * Name: MyJFrog
   * URL: http://<your-ip>:8081/artifactory
   * Credentials: Add username/password or API key



### 🧪 Configure Jenkins Job:

1. Source Code Management:  
   * Git → Provide Repository URL
2. Build Section:  
   * Add Invoke top-level Maven targets
   * Goals: install

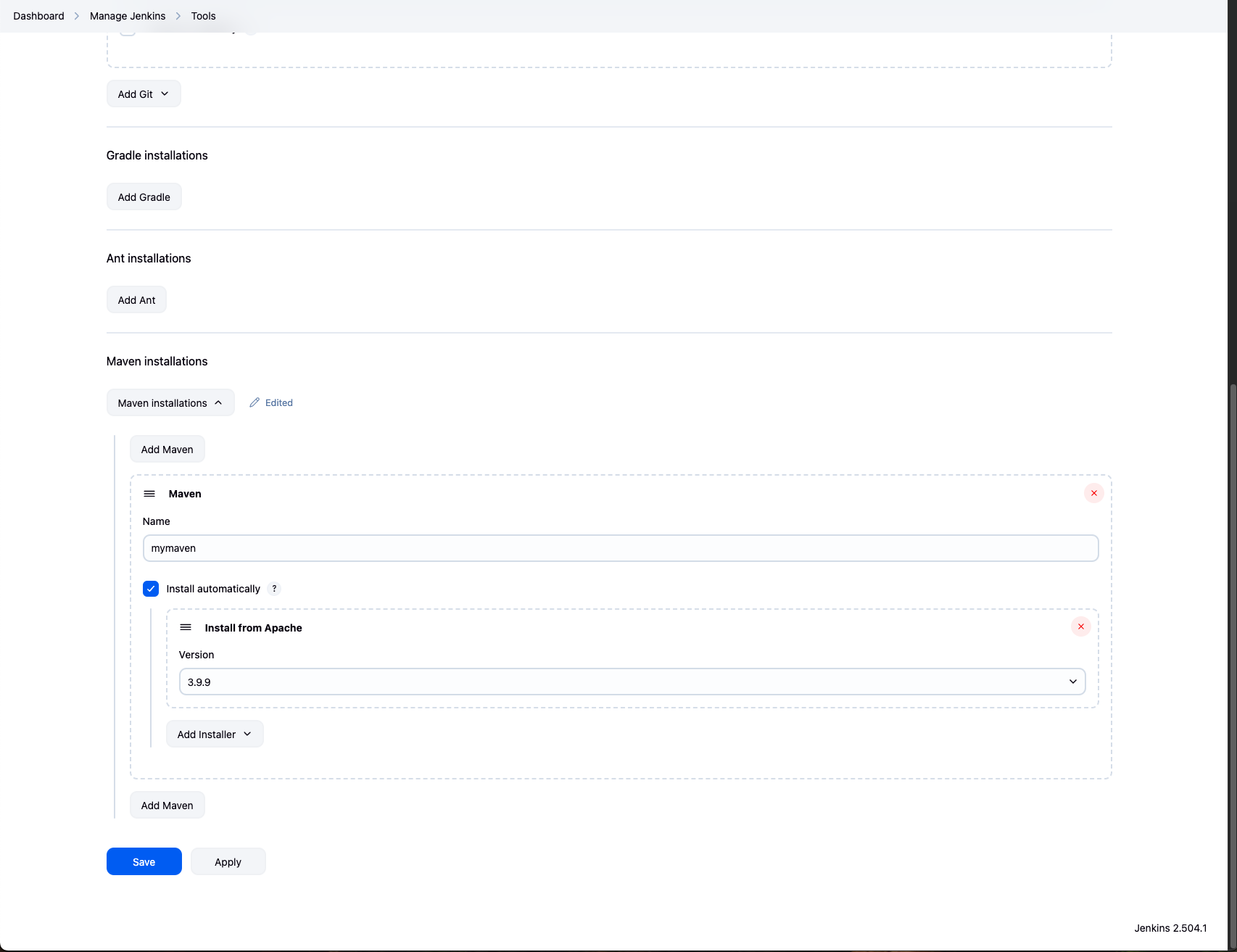
3. Post-build Actions:

* Add Deploy artifacts to Artifactory
* Provide:  
  + Artifact path: target/\*.war
  + Target repository: mytestrepo or mytestrepo-snapshot
  + Deployment patterns (if needed)

⚙️ 4. Maven Configuration in Jenkins

### Add Maven Globally:

1. Go to: Manage Jenkins → Global Tool Configuration



1. Under Maven:  
   * Click Add Maven
   * Name: mymaven
   * Check Install automatically (or specify custom path)

### Use in Job:

* In the build step, select mymaven under "Invoke Maven Targets"

🔁 5. DevOps CI/CD Flow Summary (End-to-End)

| Stage | Tool | Description |
| --- | --- | --- |
| Code Commit | GitHub | Developer commits code to the Git repo |
| Build Trigger | Jenkins | Triggered by webhook or Poll SCM |
| Build & Test | Maven | Jenkins uses Maven to build .war |
| Archive | Jenkins | Stores .war under Build Artifacts |
| Artifact Upload | JFrog | Jenkins uploads to JFrog (versioned) |
| Deployment | Ansible/K8s | Deployment tools pull artifact from JFrog |

