

✓ 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)

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

tejaswinimaddula110@...
DEFAULT DIRECTORY

Home > Compute infrastructure | Virtual machines >

JfrogServer

Virtual machine

Search

Help me copy this VM in any region

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Connect

Networking

Network settings

Load balancing

Application security groups

Network manager

Settings

Disks

Extensions + applications

Operating system

Configuration

Advisor recommendations

Properties

Locks

Availability + scale

Security

Backup + disaster recovery

Operations

Monitoring

Automation

Help

Connect Start Restart Stop Hibernate Capture Delete Refresh Open in mobile Feedback CLI / PS

Essentials

Resource group (move) : JfrogBg

Status : Running

Location : East US

Subscription (move) : Azure for Students

Subscription ID : 06aa5329-5b5a-4789-bfe4-fbc2bfd81041

Operating system : Linux (ubuntu 22.04)

Size : Standard D4s v3 (4 vcpus, 16 GiB memory)

Public IP address : 20.185.243.16

Virtual network/subnet : JfrogServer-vnet/default

DNS name : Not configured

Health state : -

Time created : 5/6/2025, 2:43 AM UTC

Tags (edit) : Add tags

JSON View

Properties

Monitoring

Capabilities (7)

Recommendations

Tutorials

Virtual machine

Computer name : JfrogServer

Operating system : Linux (ubuntu 22.04)

VM generation : V2

VM architecture : x64

Agent status : Ready

Agent version : 2.13.1.1

Hibernation : Disabled

Host group : -

Host : -

Proximity placement group : -

Colocation status : N/A

Capacity reservation group : -

Disk controller type : SCSI

Azure Spot

Azure Spot : -

Azure Spot eviction policy : -

Availability + scaling

Availability zone (edit) : -

Availability set : -

Scale Set (attach) : -

Security

Security type : Trusted launch

Enable secure boot : Enabled

Enable vTPM : Enabled

Isolation : Disabled

Networking

Public IP address : 20.185.243.16 (Network interface Jfrogserver271)

Public IP address (IPv6) : -

Private IP address : 10.0.0.4

Private IP address (IPv6) : -

Virtual network/subnet : JfrogServer-vnet/default

DNS name : Configure

Size

Size : Standard D4s v3

vCPUs : 4

RAM : 16 GiB

Source image details

Source image publisher : canonical

Source image offer : 0001-com-ubuntu-server-jammy

Source image plan : 22_04-lts-gen2

Disk

OS disk : JfrogServer_OsDisk_1_077a1d73df3b4138b972490f7a7916c

Encryption at host : Disabled

Azure disk encryption : Not enabled

Ephemeral OS disk : N/A

Data disks : 0

Auto-shutdown

Auto-shutdown : Not enabled

Scheduled shutdown : -

Ad Add or remove favorites by pressing Cmd+Shift+F

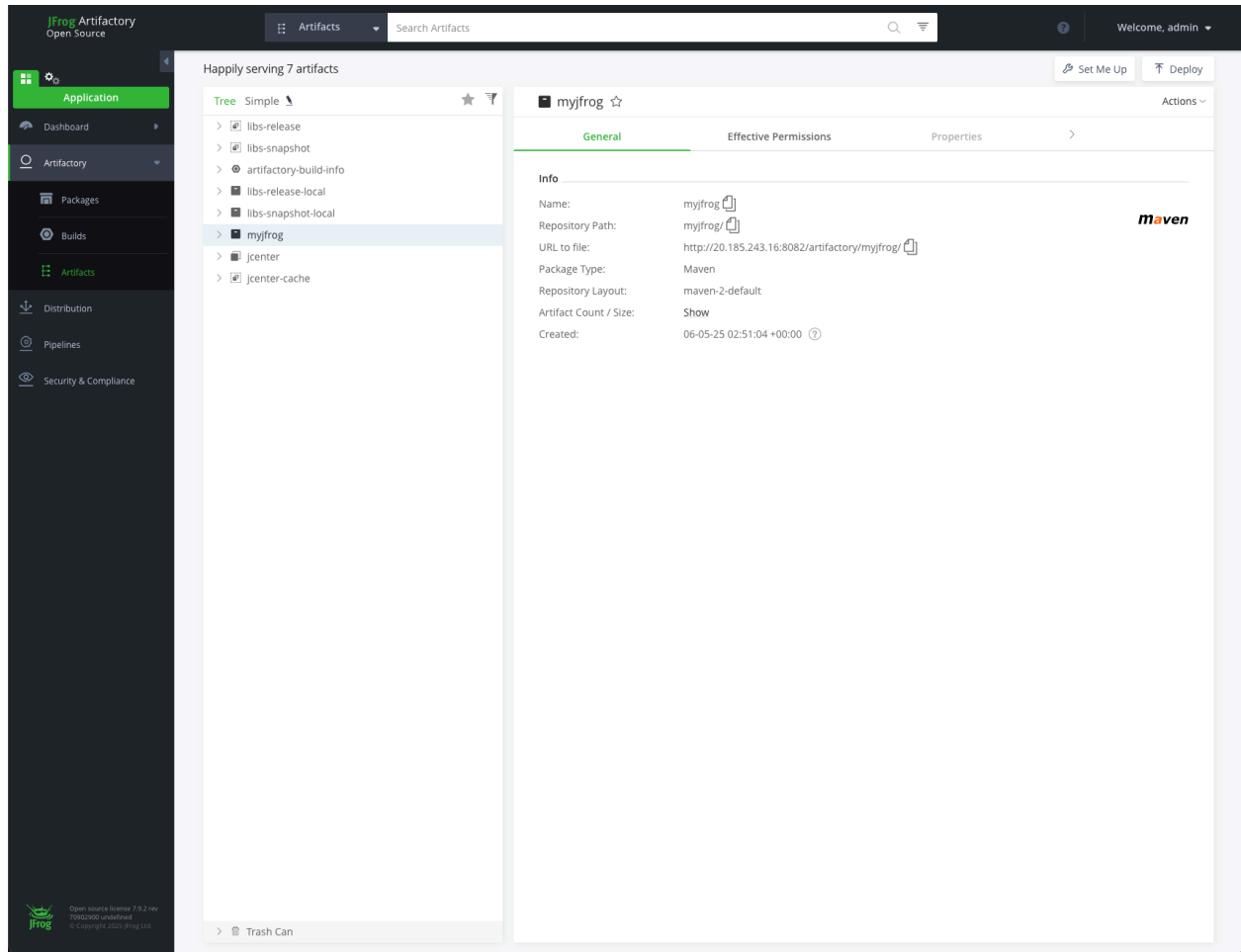
The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a Copilot button. The main content area is titled 'JfrogServer | Network settings'. On the left, there is a sidebar with various navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Connect, Networking, Load balancing, Application security groups, Network manager, Settings, Disks, Extensions + applications, Operating system, Configuration, Advisor recommendations, Properties, Locks, Availability + scale, Security, Backup + disaster recovery, Operations, Monitoring, Automation, and Help. The 'Networking' section is expanded, and 'Network settings' is selected. The main content area shows the network configuration for the virtual machine 'JfrogServer'. It includes a dropdown menu for 'Network interface / IP configuration' with the selected option 'jfrogserver271 (primary) / ipconfig1 (primary)'. Below this, there are sections for 'Essentials' and 'Rules'. The 'Essentials' section lists various network configuration options like Network interface, Virtual network / subnet, Public IP address, Private IP address, and Admin security rules. The 'Rules' section shows the 'Network security group JfrogServer-nsg' with a list of inbound and outbound port rules. The inbound rules include SSH, AllowAnyCustomAnyInbound, AllowVnetInbound, AllowAzureLoadBalancerInbound, and DenyAllInbound. The outbound rules include AllowAnyCustomAnyOutbound, AllowVnetOutbound, AllowInternetOutbound, and DenyAllOutbound.

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

Dashboard > Manage Jenkins > System >

JFrog Platform Instances

List of JFrog platform instances that projects will want to deploy artifacts and build info

JFrog instance details

Instance ID ?

myjfrog

JFrog Platform URL ?

http://20.185.243.16:8082

Advanced Configuration ^ Edited

Advanced Configuration

JFrog Artifactory URL ?

http://20.185.243.16:8082/artifactory

JFrog Distribution URL ?

http://20.185.243.16:8082/distribution

Connection timeout ?

300

Number of retries ?

3

Number of threads for Generic uploads ?

3

☐ Bypass HTTP proxy ?

Default Deployer Credentials

Username ?

admin

Password ?

Concealed [Change Password](#)

[Save](#) [Apply](#)

 Configure Jenkins Job:

1. Source Code Management:

- Git → Provide Repository URL

2. Build Section:

- Add Invoke top-level Maven targets
- Goals: install

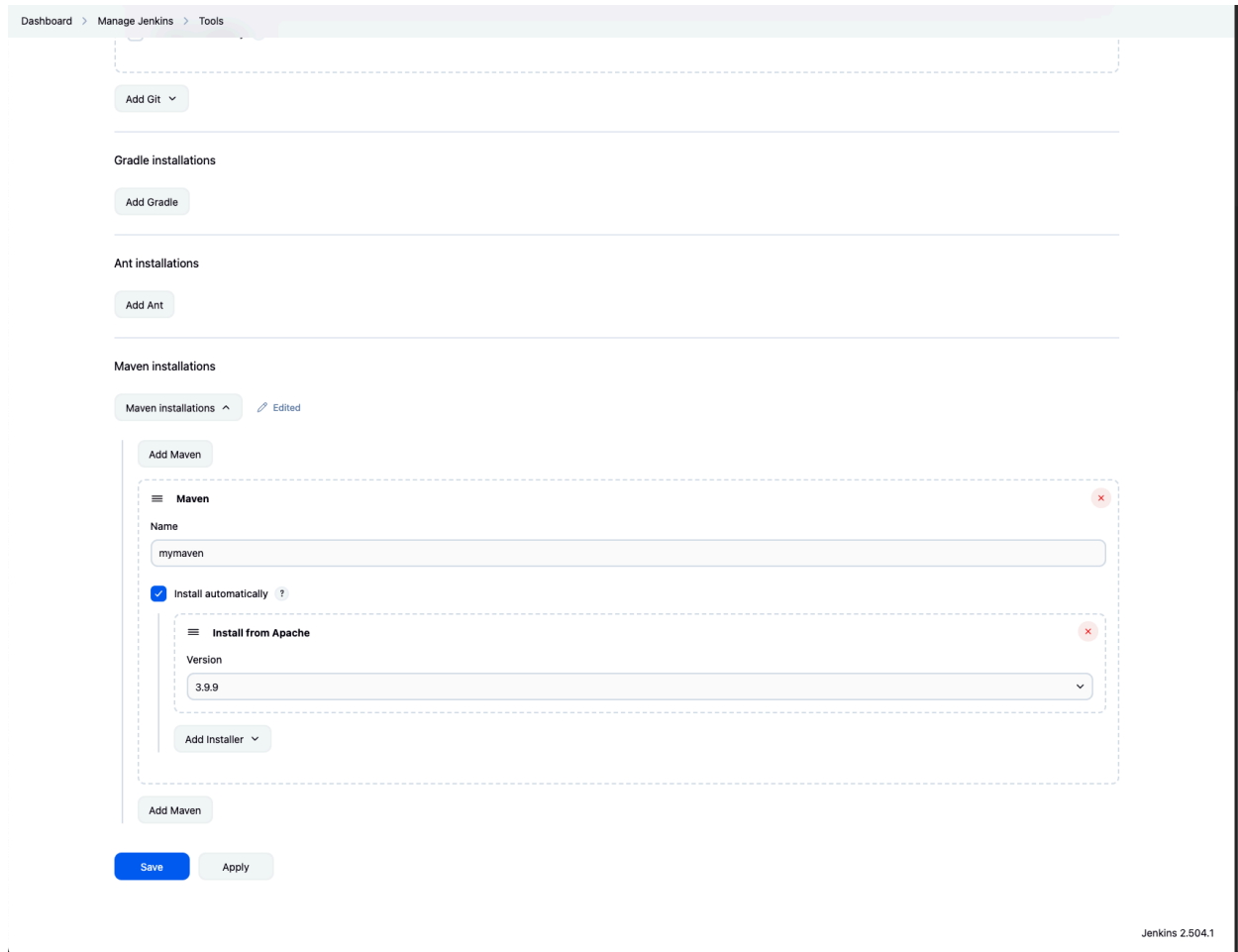
3. Post-build Actions:

- 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



2. 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 <code>.war</code>
Archive	Jenkins	Stores <code>.war</code> under Build Artifacts
Artifact Upload	JFrog	Jenkins uploads to JFrog (versioned)
Deployment	Ansible/K8s	Deployment tools pull artifact from JFrog

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

Git

Repositories

Repository URL

https://github.com/AbdulRehman082493/mahalogin.git

Credentials

- none -

+ Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master

Add Branch

Repository browser

(Auto)

Additional Behaviours

Add

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ Enable Artifactory trigger
- ☒ GitHub hook trigger for GITScm polling



Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

- ☐ Enable Artifactory trigger
- ☒ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) ?
- ☐ Provide Configuration files ?
- ☐ Add timestamps to the Console Output
- ☐ Ant/Ivy-Artifactory Integration
- ☐ Generic-Artifactory Integration
- ☐ Gradle-Artifactory Integration
- ☐ Inspect build log for published build scans
- ☒ Maven3-Artifactory Integration

Artifactory Configuration

Artifactory server

myjfrog http://20.185.243.16:8082/artifactory

Target releases repository ?

myjfrog

Select from List

Target snapshots repository ?

myjfrog

Select from List

Refresh Repositories

- ☐ Override default credentials
- ☐ Resolve artifacts from Artifactory ?
- ☐ Override build name ?
- ☒ Deploy artifacts to Artifactory ?
- Include Patterns ?



Configure

- General
- Source Code Management
- Triggers
- Environment**
- Build Steps
- Post-build Actions

☒ Maven3-Artifactory Integration

Artifactory Configuration

Artifactory server

myfrog http://20.185.243.16:8082/artifactory

Target releases repository ?

myfrog

Select from List

Target snapshots repository ?

myfrog

Select from List

Refresh Repositories

☐ Override default credentials

☐ Resolve artifacts from Artifactory ?

☐ Override build name ?

☒ Deploy artifacts to Artifactory ?

Include Patterns ?

Exclude Patterns ?

/.war=>\${BUILD_NUMBER}

☒ Filter excluded artifacts from build info ?

Deployment properties ?

☒ Capture and publish build info ?

☐ Include environment variables ?

☐ Record implicit project dependencies and build-time dependencies ?

☐ Discard old builds from Artifactory (requires Artifactory Pro) ?

Project ?



Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

☒ Deploy artifacts to Artifactory ?

Include Patterns ?

Exclude Patterns ?

*/war=>\${BUILD_NUMBER}

☒ Filter excluded artifacts from build info ?

Deployment properties ?

☒ Capture and publish build info ?

☐ Include environment variables ?

☐ Record implicit project dependencies and build-time dependencies ?

☐ Discard old builds from Artifactory (requires Artifactory Pro) ?

Project ?

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

☒ Invoke Artifactory Maven 3 ?

Maven Version

mymaven

Root POM ?

pom.xml

Goals and options ?

install



Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

- ☒ Capture and publish build info ?
 - ☐ Include environment variables ?
 - ☐ Record implicit project dependencies and build-time dependencies ?
 - ☐ Discard old builds from Artifactory (requires Artifactory Pro) ?

Project ?

- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Invoke Artifactory Maven 3 ?

Maven Version

Root POM ?

Goals and options ?

Advanced ▾

Add build step ▾

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action ▾

Save

Apply



- Status
- </> Changes
- 📁 Workspace
- ▶ Build Now
- ⚙️ Configure
- 🗑️ Delete Project
- 🔄 Artifactory Build Info
- 📄 GitHub Hook Log
- ✎ Rename

✔️ jobA

✎ Add description

🔄

[Artifactory Build info](#)

Permalinks

- [Last build \(#3\), 16 min ago](#)
- [Last stable build \(#3\), 16 min ago](#)
- [Last successful build \(#3\), 16 min ago](#)
- [Last completed build \(#3\), 16 min ago](#)

BUILDS

*** 🔗

🔍 Filter

📄

Today

✔️ #3 3:49 AM

🔄 🔄 ▾

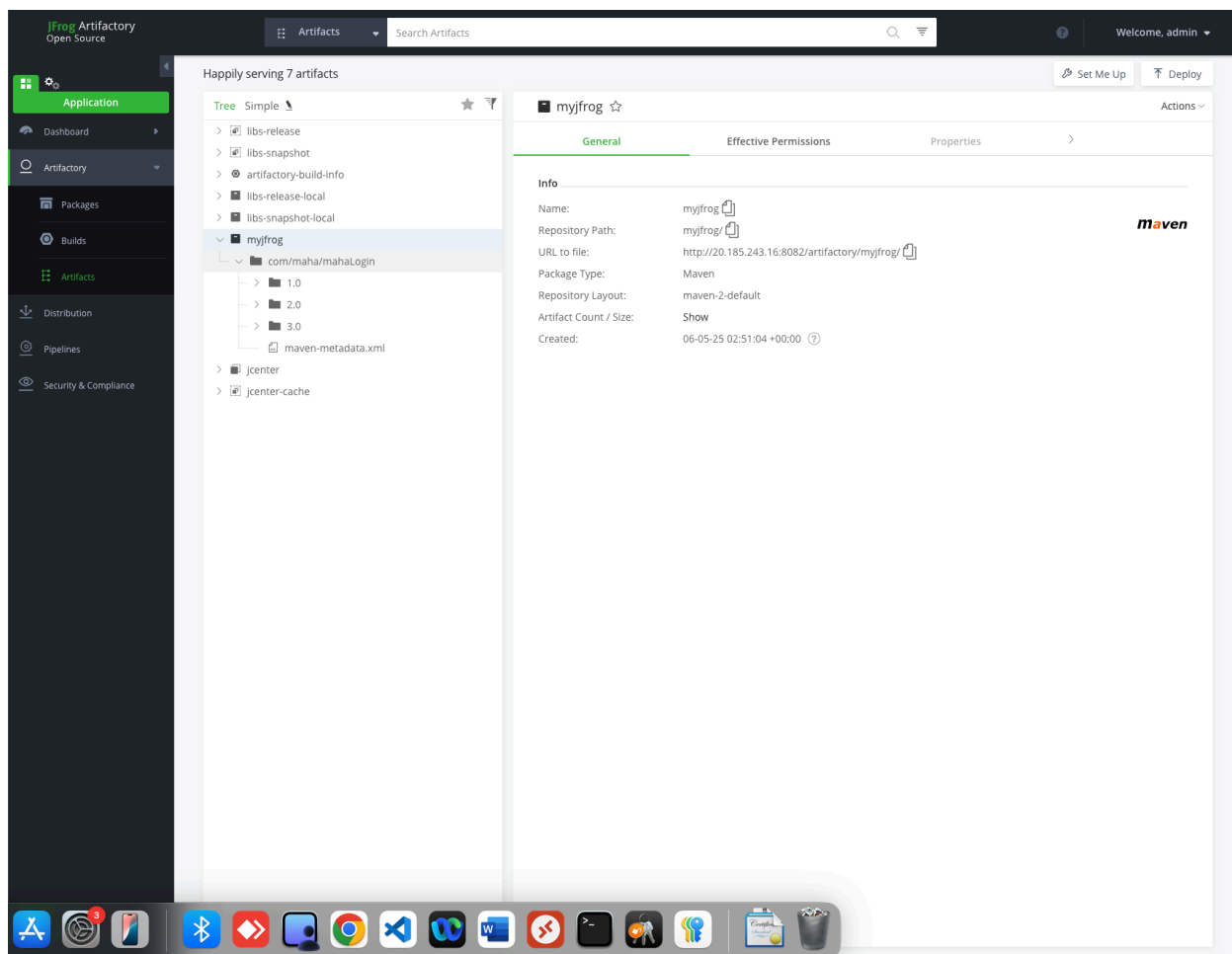
✔️ #2 3:41 AM

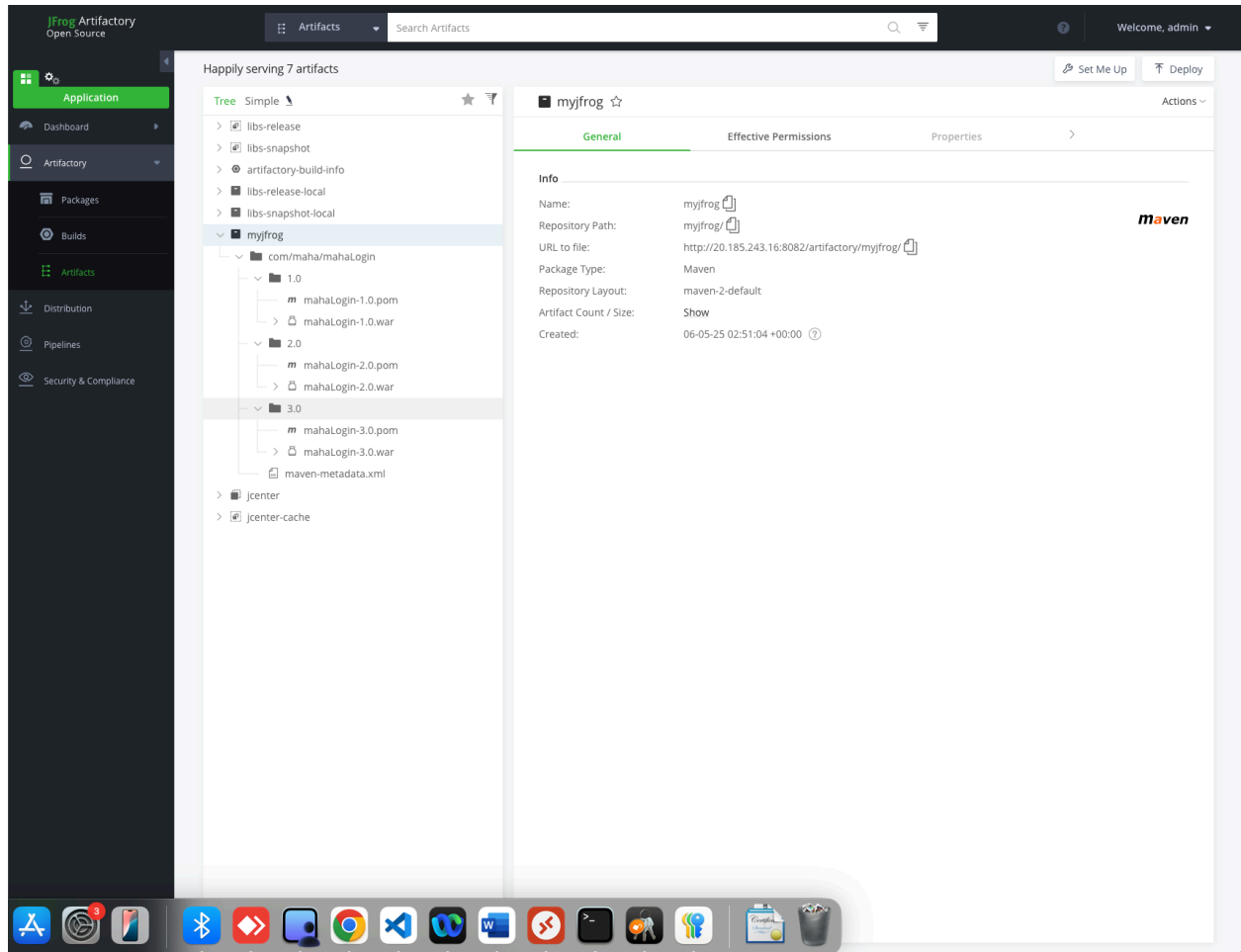
🔄 🔄 ▾

✔️ #1 3:16 AM

🔄 🔄 ▾











AbdulRehaman082493 / mahalogin


Q


Type  to search



















<> Code


 Pull requests


 Actions

 Projects


 Wiki


 Security

 Insights



 Settings

Commit 9044a5d


 Browse files


 AbdulRehaman082493 authored 20 minutes ago Verified

Update pom.xml


 master 1 parent [a2d45fb](#) commit 9044a5d 

Q Filter files...



 pom.xml

1 file changed +1 -1 lines changed

pom.xml 

+

-

@@ -4,7 +4,7 @@

4 4 <groupId>com.maha</groupId>

5 5 <artifactId>mahaLogin</artifactId>

6 6 <packaging>war</packaging>

7 - <version>1.0</version>

7 + <version>3.0</version>


8 8 <name>Account Registration Webapp</name>


9 9 <url>http://maven.apache.org</url>

10 10 <properties>

+

Comments 0

 Lock conversation

 Comment

Customizable line height

The default line height has been increased for improved accessibility. You can choose to enable a more compact line height from the view settings menu.

Enable compact line height

Dismiss