DevOps: Automating Semantic Versioning Using GitHub Actions

November 25, 2022 7:21 am No Comments

Author: Pulkit Bindal

What are GitHub Actions?

With GitHub Actions, you may automate, modify, and carry out the processes for software development in your repository. To do any job, including CI/CD, you can find, create, and share actions. You may then combine your efforts into a unique workflow.

In other words,

1. GitHub Actions makes it easy to automate all your software workflows.
2. GitHub actions let you build, test, and deploy your code right from Github.
3. We can also automate other applications via GitHub Actions by integrating them with Github.

What is Semantic Versioning?

This procedure is intended to make it easier to automatically assign version numbers during a build while publishing versions that only increase by one value per release. To do this, the following version number is calculated together with a commit increment that shows how many commits were made for this version. To identify the kind of version change the following version reflects, the commit messages are examined. The sort of modification the upcoming version will represent will change if the title message for the pull request uses the terms major, minor, or patch for the major, minor, or patch version, respectively.

Recent Posts

- Salesforce Data Loader – Import & Export Data
- How To Move Bulk Reports And Dashboard To Another Folder Using Metadata API
- How To Configure SAML 2.0 Via Okta For MuleSoft – Anypoint Platform
- Subscribe To Salesforce Platform Events Using MuleSoft
- Auto-Publish APIs With GitHub Actions – March 4th, 2023

Categories

- Business Insights (6)
- Events (3)
- MuleSoft (333)
- News (5)
- Salesforce (51)
- Snowflake (23)
- Tableau (5)
- Tech Tutorials (415)
- Webinars (19)

Facebook Twitter LinkedIn

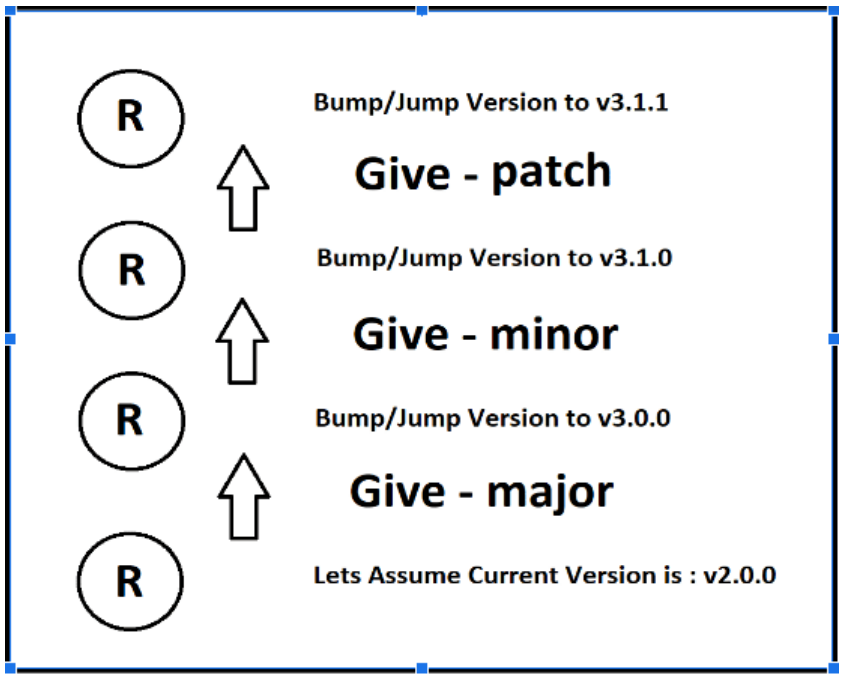
What is the background of Automatic Semantic Versioning?

The fundamental drawback of the previous systems is that they lack descriptiveness. It might be challenging for a user to determine whether non-breaking changes have been included in a new version when comparing various versions that come after an incremented version. The new version's purpose cannot be inferred only from its version number.

A way to make version numbers more evocative is by using semantic versioning. The format of a semantic version number is MAJOR.MINOR.PATCH.

The different sections are numbers. We increment:

- the MAJOR part by passing major, on after successfully merging the Pull request,
- the MINOR part by passing minor, on after successfully merging the Pull request, and
- the PATCH part by passing the patch, on after successfully merging the Pull request.

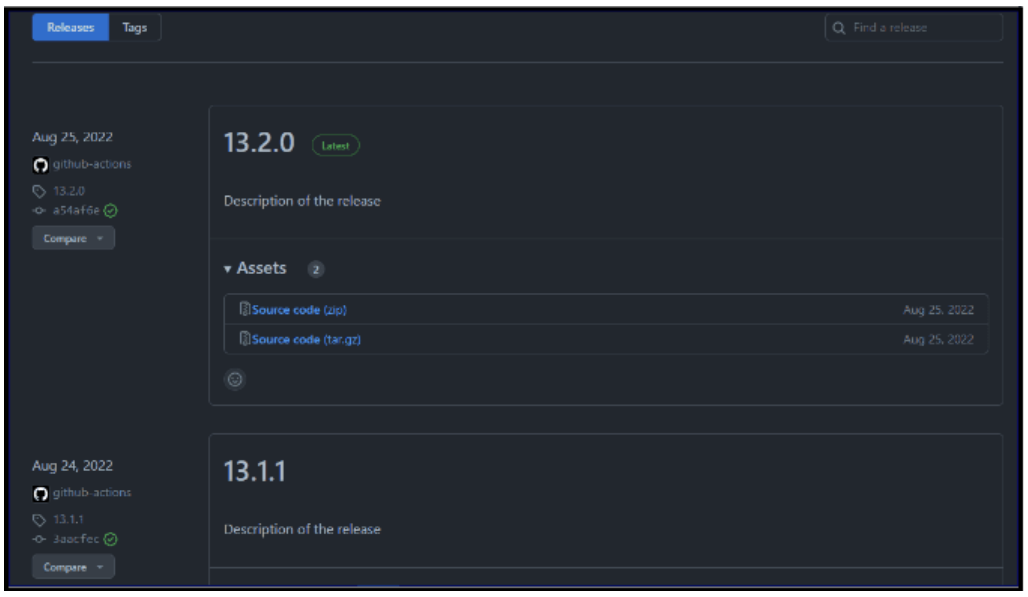


Make GitHub Action Workflow Dir like -> .github/workflows/increment.yaml

```
name: Creation of Release Version
on:
  pull_request:
    branches: [main]
    types:
      - closed

jobs:
  build:
    name: Create Release
    if: github.event.pull_request.merged == true
    runs-on: ubuntu-latest
    steps:
      - name: Taking the Latest Release Tag number
        id: releaseVersion
        run: |
          owner="<owner`s name>"
          repo="<repo`s name>"
          release_json=$(curl
https://api.github.com/repos/$owner/$repo/releases)
          Release_tag=$(echo "$release_json" | jq -r '.
[0].tag_name')
          echo "Release_tag: Latest Tag is : $Release_tag"
          echo ::set-output name=Release_tag::"$Release_tag"
```

```
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Bumping Major Index
        id: bump_version_major
        if: contains(github.event.pull_request.title, 'major')
        uses: christian-draeger/increment-semantic-version@1.0.2
        with:
          current-version: ${{
steps.releaseVersion.outputs.Release_tag }}
          version-fragment: 'major'

      - name: Bumping Minor Index
        id: bump_version_minor
        if: contains(github.event.pull_request.title, 'minor')
        uses: christian-draeger/increment-semantic-version@1.0.2
        with:
          current-version: ${{
steps.releaseVersion.outputs.Release_tag }}
          version-fragment: 'feature'

      - name: Bumping Patch Index
        id: bump_version_patch
        if: contains(github.event.pull_request.title, 'patch')
        uses: christian-draeger/increment-semantic-version@1.0.2
        with:
          current-version: ${{
steps.releaseVersion.outputs.Release_tag }}
          version-fragment: 'bug'

      - name: Create release version for bump_version_major
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        run: |
          owner="<owner`s name>"
          repo="<repo`s name>"
          curl \
            -X POST \
            -H "Accept: application/vnd.github+json" \
            -H "Authorization: token ${{ secrets.GITHUB_TOKEN }}"
\
             https://api.github.com/repos/$owner/$repo/releases \
            -d '{"tag_name":"${{
steps.bump_version_major.outputs.next-version
}}","target_commitish":"master","name":"${{
steps.bump_version_major.outputs.next-version
}}","body":"Description of the
release","draft":false,"prerelease":false,"generate_release_notes
":false}'
          curl \
            -X POST \
            -H "Accept: application/vnd.github+json" \
            -H "Authorization: token ${{ secrets.GITHUB_TOKEN }}"
\
             https://api.github.com/repos/$owner/$repo/releases \
            -d '{"tag_name":"${{
steps.bump_version_minor.outputs.next-version
}}","target_commitish":"master","name":"${{
steps.bump_version_minor.outputs.next-version
}}","body":"Description of the
release","draft":false,"prerelease":false,"generate_release_notes
":false}'
          curl \
            -X POST \
            -H "Accept: application/vnd.github+json" \
            -H "Authorization: token ${{ secrets.GITHUB_TOKEN }}"
\
             https://api.github.com/repos/$owner/$repo/releases \
            -d '{"tag_name":"${{
steps.bump_version_patch.outputs.next-version
}}","target_commitish":"master","name":"${{
steps.bump_version_patch.outputs.next-version
}}","body":"Description of the
release","draft":false,"prerelease":false,"generate_release_notes
":false}'
```

Output:



PREVIOUS
Converting A cXML To EDI Of Desired Version

NEXT
Enable State And Country Picklist