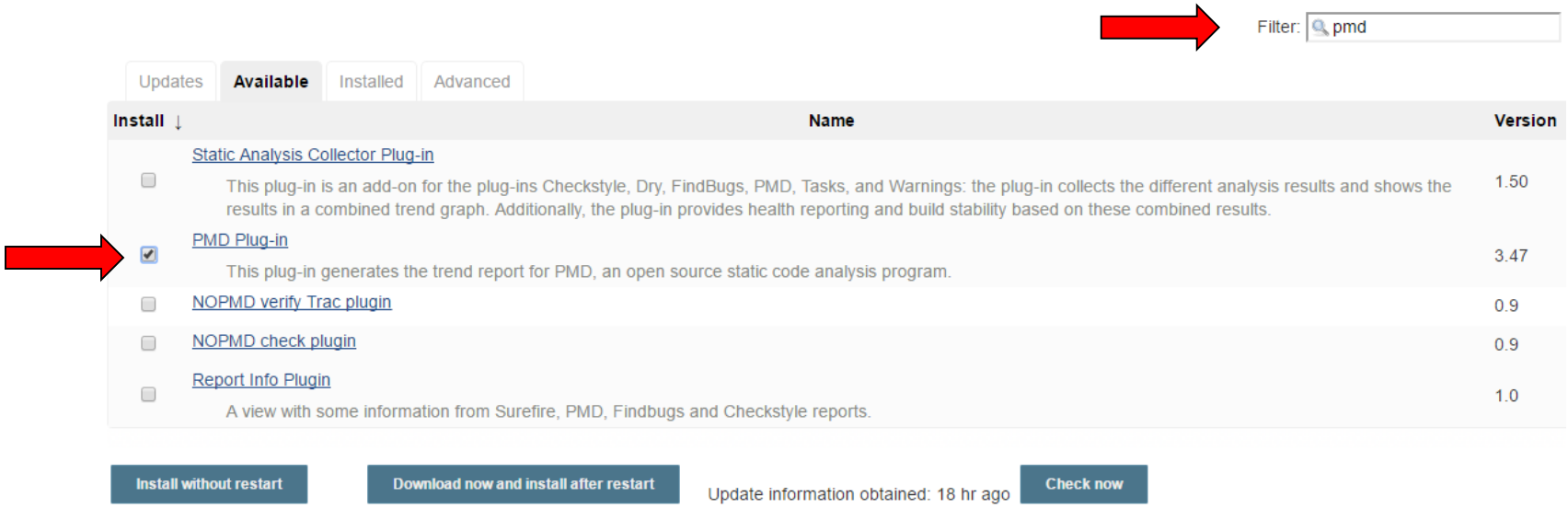
Jenkins – Static Code Analysis

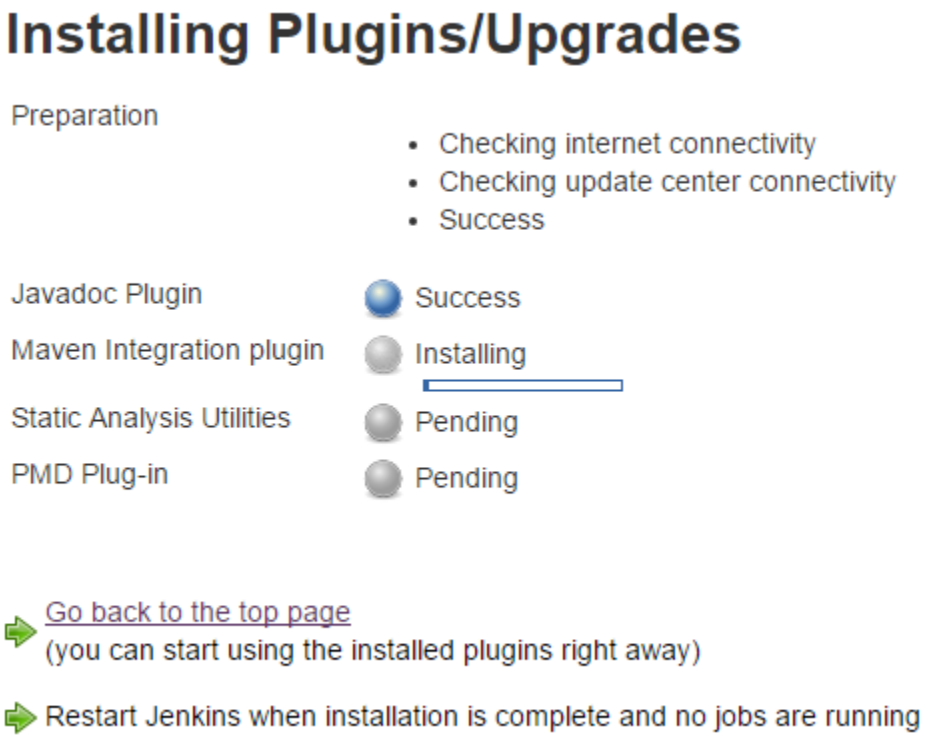
Configure Plugins for Code Analysis

- Go to Manage Jenkins → Manage Plugins
- Search for PMD under ‘Available’ plugins tab
- Install



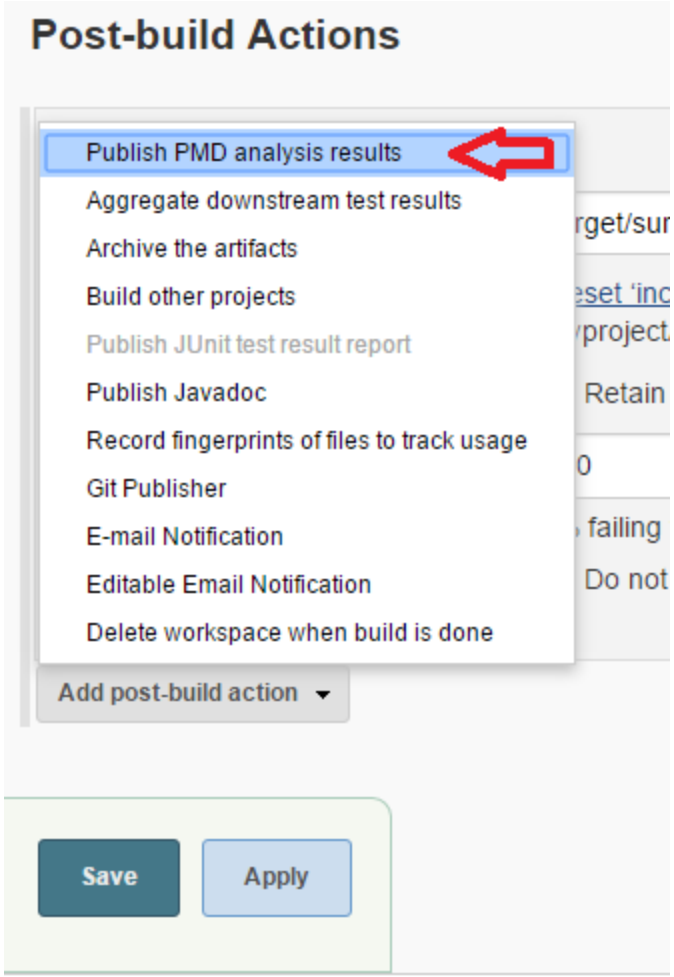
Configure Plugins for Code Analysis

- This will also install other dependencies
- Note that Static analysis utilities plugin is also installed. This basically provides a dashboard view, trends graph etc for data generated by PMD and other tools



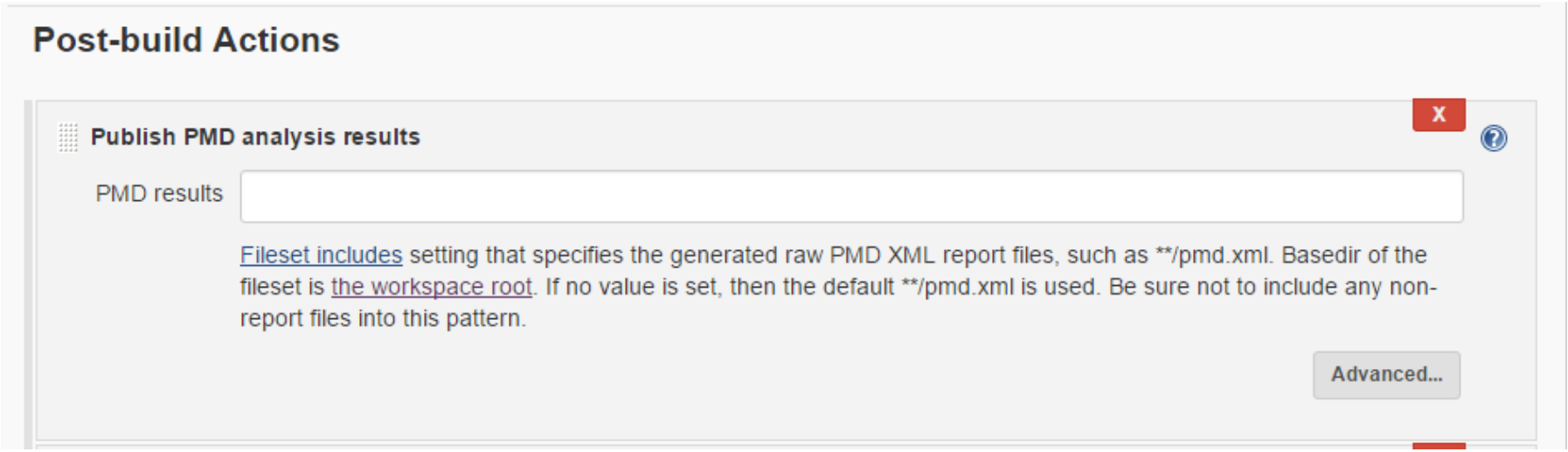
Configure Jenkins job to run PMD

- Go to Jenkins Dashboard, Select ‘mvndemo’ build job to configure
- Add Post-Build-Actions
- Click Apply and Save



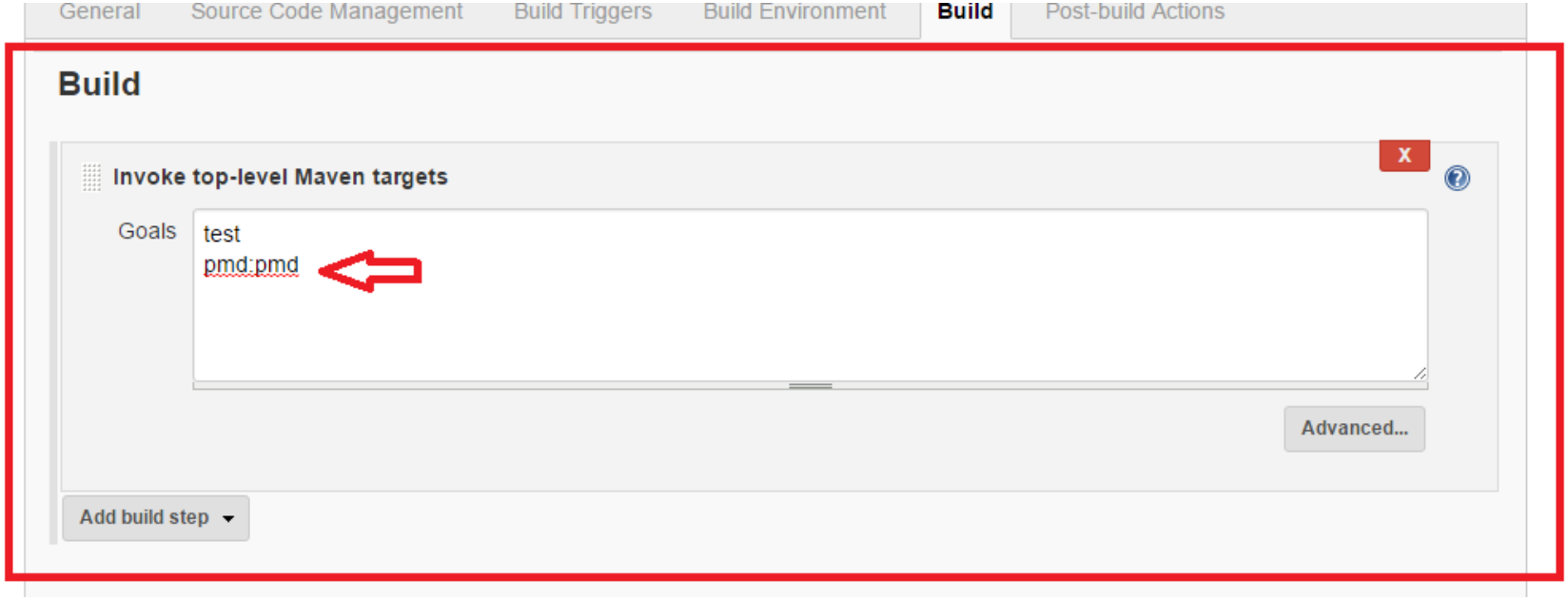
Configure Jenkins job to run PMD

- You can keep the PMD Results directory empty, since it will be picked up from default location in jenkins which is /.jenkins/workspace/mvndemo/target/pmd.xml
- Don’t forget to add the goal pmd:pmd to the build step



Configure Jenkins job to run PMD contd

- Don’t forget to add the goal pmd:pmd to the build step



Configure Maven project for Code Analysis

- Lets configure the project ‘mvndemo’ to be statically analysed by PMD
- PMD will be run using maven-pmd-plugin
- Open pom.xml and make the following entry just below <dependencies>
- Note: PMD has various rulesets, we are just going to use basic ruleset.

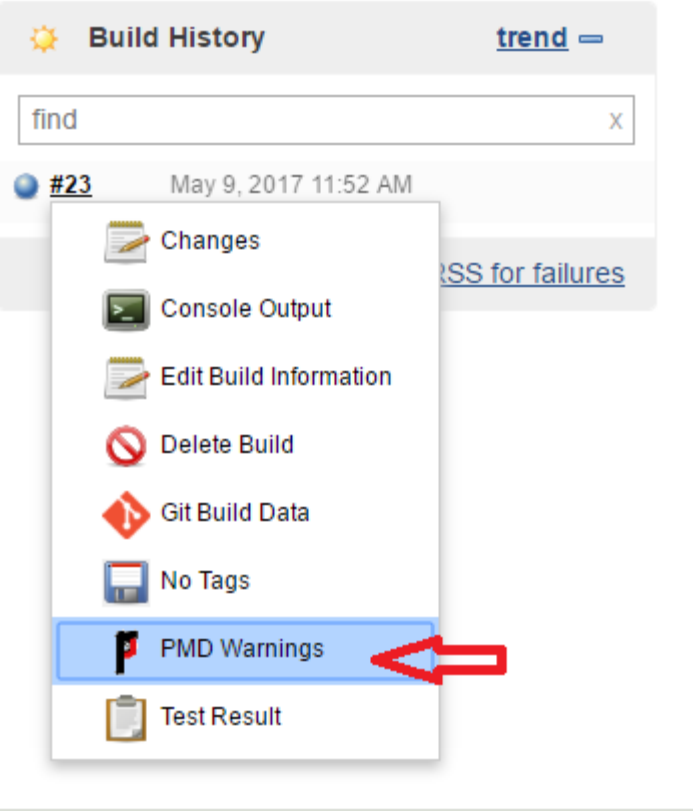
```
<reporting>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-pmd-plugin</artifactId>
            <version>3.7</version>
            <configuration>
                <rulesets>
                    <ruleset>/rulesets/basic.xml</ruleset>
                </rulesets>
            </configuration>
        </plugin>
    </plugins>
</reporting>
```

Configure Maven project for Code Analysis

- Now open Bitbucket Repository and deliberately introduce a violation
- Modify App.java – Here I am introducing a local variable which is unused
- Run the build in Jenkins – PMD should pick this violation

View Code Analysis Results

- To view PMD Violation Report – Click on PMD Warnings



View Code Analysis Results contd..

- You can see that PMD has flagged the Unused Local Variable. Click on the link provided to the source file.

PMD Result

Warnings Trend

All Warnings	New Warnings
1	1

Summary

Total	High Priority	Normal Priority
1	0	1

Details

Details | New

App.java:16, UnusedLocalVariable, Priority: Normal

Avoid unused local variables such as 'x'.

Detects when a local variable is declared and/or assigned, but not used.

```
public class Foo {
        public void doSomething() {
                int i = 5; // Unused
        }
}
```

View Code Analysis Results contd..

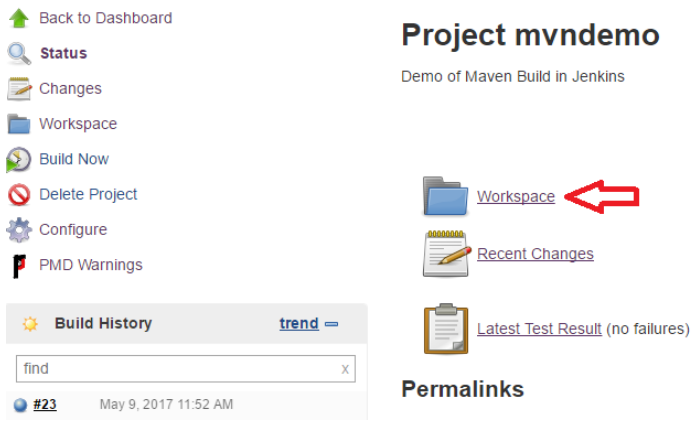
- You can see that PMD has flagged the Unused Local Variable and Source code, line number is displayed

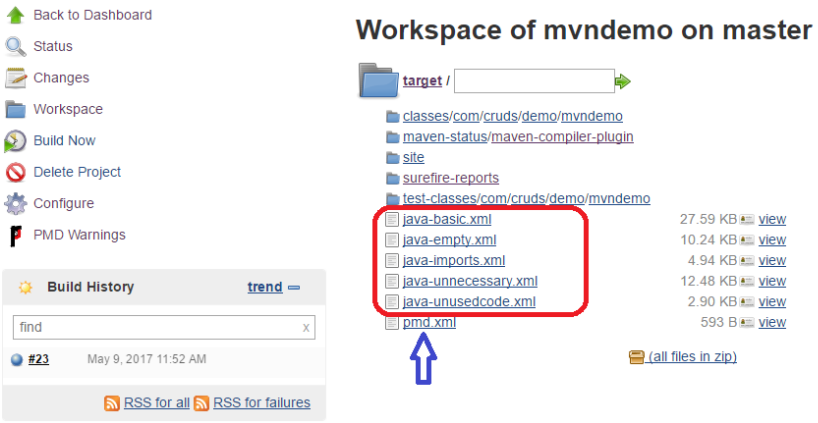
Content of file App.java

```
01 package com.cruds.demo.mvndemo;
02
03 /**
04  * Hello world!
05  *
06  */
07 public class App
08 {
09     public static void main( String[] args )
10     {
11         System.out.println( "Hello World!" );
12     }
13
14     public String getGreeting()
15     {
16       int x = 10;
17         return "Hello from Maven";
18     }
19
20
21 }
```

Behind the scenes..

- For the curious few who want to know what happened behind the scene, lets browse Jenkins workspace and see for ourselves
- Click on the ‘workspace’ folder, then click on ‘target’
- You can see 5 rules xml files and below it the pmd.xml which contains the result of analysis
- You can also click on ‘site’ folder and view the pmd.html
- You can also view by browsing the folder .jenkins/workspace/mvndemo/target





Lets make more violations in App.java

- Lets import a Class which is not used any where in code, there by violating, Unused Imports rule of PMD
- Make the changes in bitbucket repository
- Kick of the Jenkins build again, we should have 2 violations now

master | mvndemo / src / main / java / com / cruds / demo / mvndemo / App.java

40421b9 10 seconds ago | Full commit

```
1  package com.cruds.demo.mvndemo;
2
3  import java.util.Scanner;
4
5  /**
6   * Hello world!
7   *
8   */
9  public class App
10 {
11     public static void main( String[] args )
12     {
13         System.out.println( "Hello World!" );
14     }
15
16     public String getGreeting()
17     {
18         int x = 10;
19         return "Hello from Maven";
20     }
21
22
23 }
```

View PMD Results again

- You can also view various information by clicking on other tabs – Types, Warnings etc

PMD Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
2	1	0

Summary

Total	High Priority	Normal Priority	Low Priority
2	0	2	0

Details

Categories | Types | Warnings | Details | New

Category	Total	Distribution
Import Statements	1	
Unused Code	1	
Total	2	

Static Analysis Plugin Dashboard

- Now we have had 2 build and 2 PMD analysis results
- Now we can see the defect trends in project. Click on ‘Back to Project’ link

Jenkins › mvndemo › #24 › PMD Warnings

- Back to Project
- Status
- Changes
- Console Output
- Edit Build Information
- Delete Build
- Git Build Data
- No Tags
- **PMD Warnings**
- Test Result
- Previous Build

PMD Result

Warnings Trend

All Warnings	New Warnings
2	1

Summary

Total	High Priority	Normal Priority
2	0	2

Details

Categories | Types | Warnings | Details | New

Category	Total	Distribution
Import Statements	1	
Unused Code	1	
Total	2	

Static Analysis Plugin Dashboard contd

- You can see the PMD Trend has increased
- JUnit Trend has maintained itself, since there were no failures
- These reports are accumulated overtime provides valuable inputs to managers on code quality
- Now lets fix it the bugs and rerun the build

View results again

- You can see that App.java was edited online in bitbucket
- You can also see git revision number
- PMD warnings 0
- Test results – no failures

Changes

1. App.java edited online with Bitbucket (detail)

Started by user Jenkins Admin

Revision: edc61599593d80320d906350455a7519c35d6b2c

- refs/remotes/origin/master

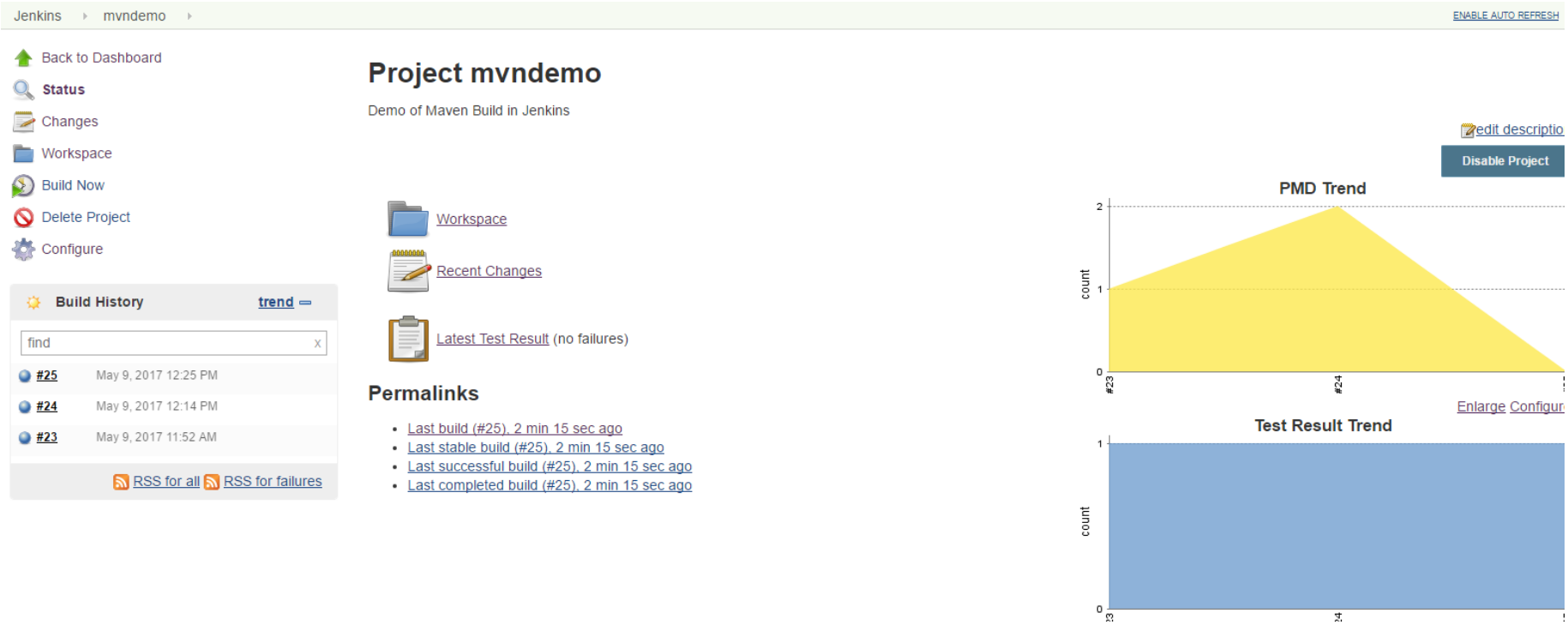
PMD: 0 warnings from one analysis.

- 2 fixed warnings

Test Result (no failures)

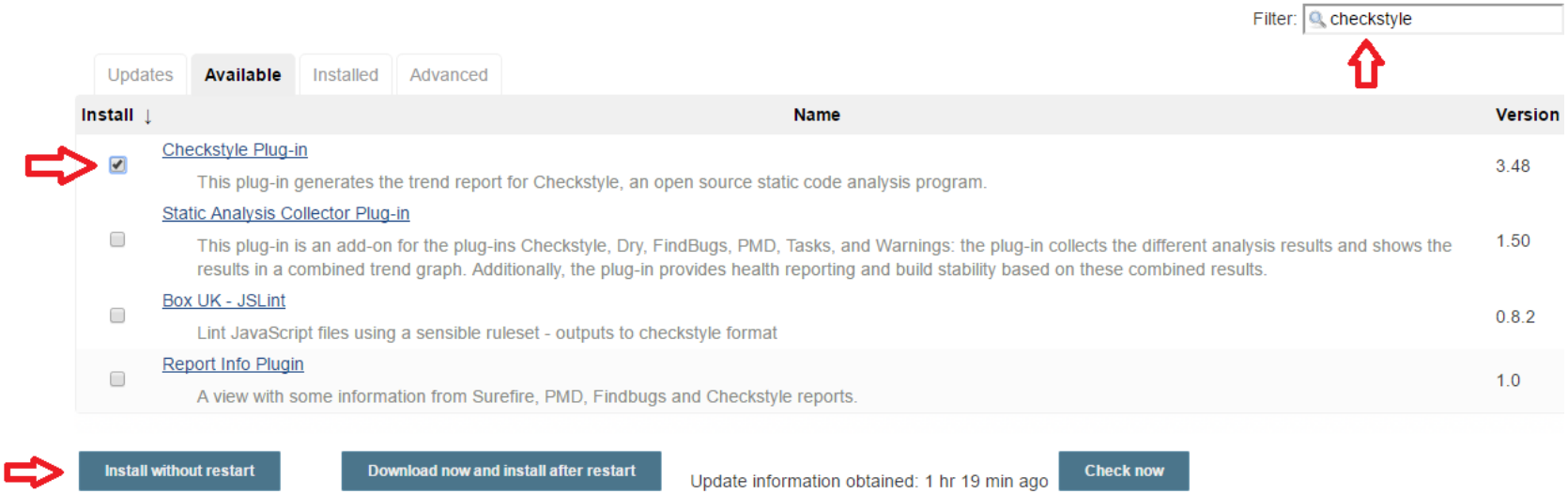
Lets see the trend again

- You can see a sharp reduction in PMD trend graph.



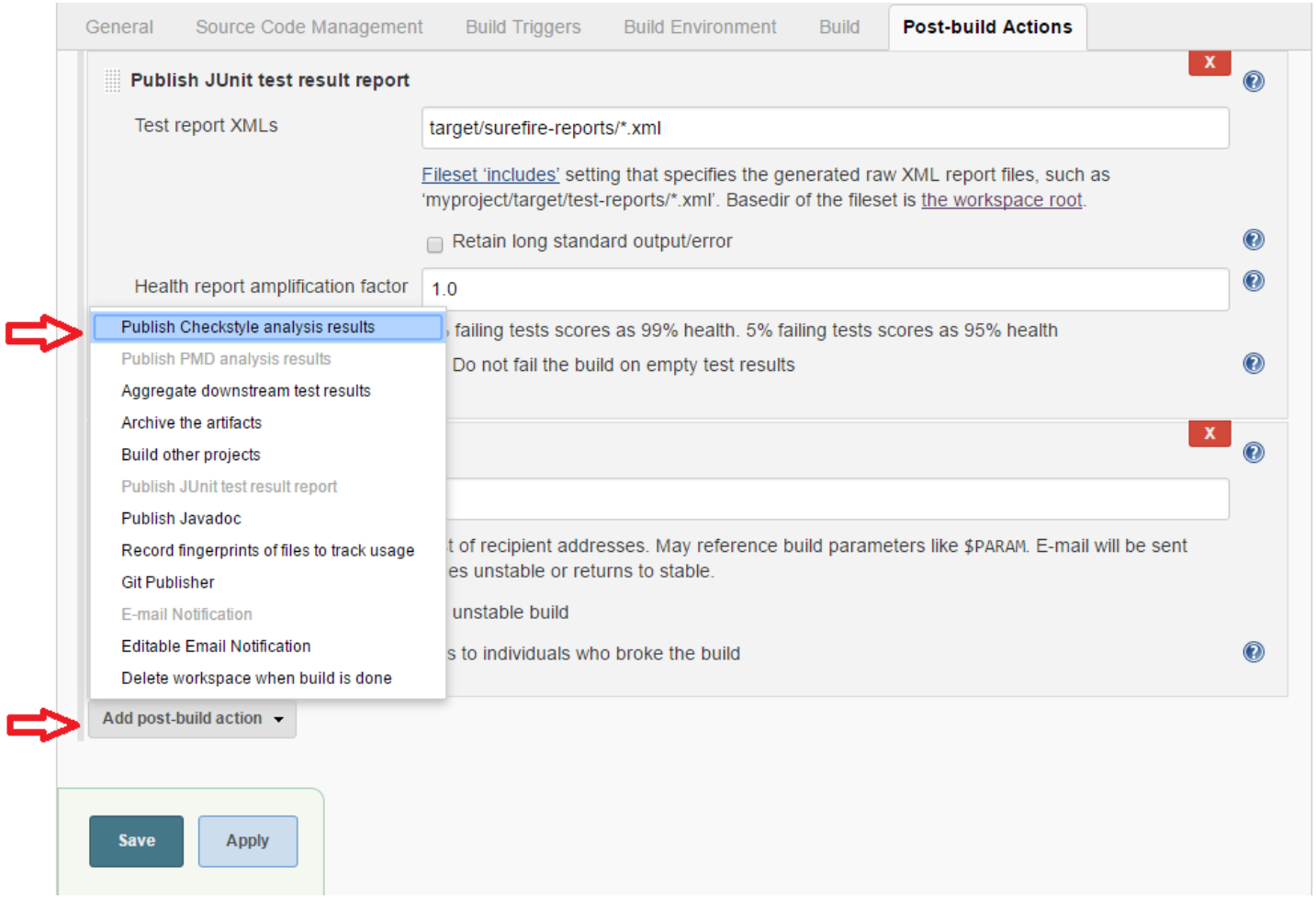
Check Style – Plugin Installation

- Go to Manage Jenkins → Manage Plugins
- Search for Checkstyle
- Select and Install without restart



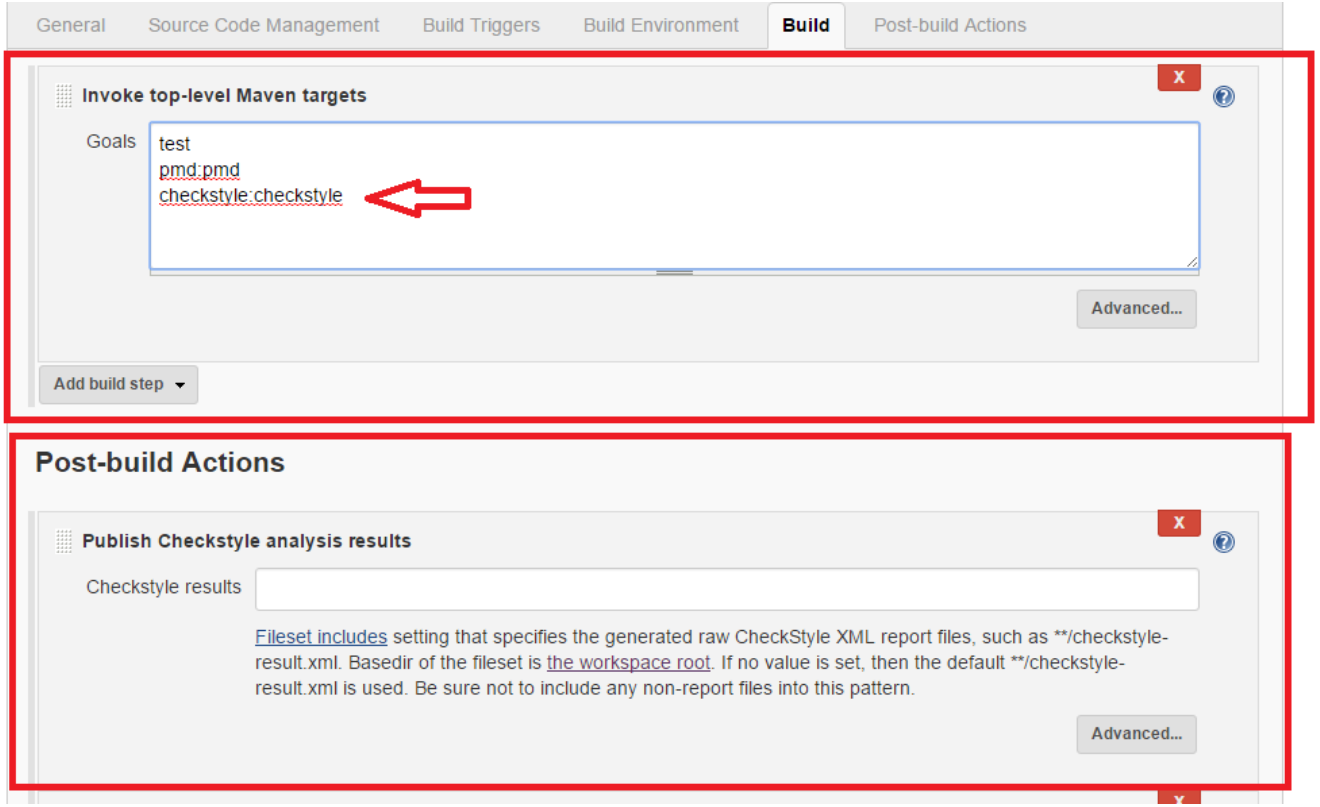
Check Style – configure Jenkins job

- Go to Jenkins dashboard, Configure ‘mvndemo’ job
- Add Post build actions → Publish Checkstyle analysis result



Check Style – configure Jenkins job contd..

- Don't forget to add the maven goal checkstyle:checkstyle to Build step
- Check style results directory can be default directory



Check Style – Configure Maven Project POM

- Make an entry for mvn-checkstyle-plugin in mvndemo project pom.xml
- You can make the changes in Bitbucket repository directly or make changes in Eclipse workspace and do *git push to remote origin*

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-checkstyle-plugin</artifactId>
    <version>2.17</version>
    <reportSets>
      <reportSet>
       <reports>
          <report>checkstyle</report>
        </reports>
      </reportSet>
    </reportSets>
 </plugin>
```

Check Style – Contd..

- After making the changes your pom.xml , ‘reporting’ section should look like this

```
<reporting>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-pmd-plugin</artifactId>
            <version>3.7</version>
            <configuration>
                <rulesets>
                    <ruleset>/rulesets/basic.xml</ruleset>
                </rulesets>
            </configuration>
        </plugin>

        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-checkstyle-plugin</artifactId>
            <version>2.17</version>
            <reportSets>
                <reportSet>
                    <reports>
                        <report>checkstyle</report>
                    </reports>
                </reportSet>
            </reportSets>
        </plugin>
    </plugins>
</reporting>
```

View Check Style Report

- Run the build and check the reports
- Click on warnings link

Changes

1. added checkstyle plugin to pom (detail)

Started by user Jenkins Admin

Revision: 3dd32cdf17f9c7a9661d515c4643a29f57b42fc3

- refs/remotes/origin/master

Checkstyle: 18 warnings from one analysis.

- 18 new warnings

PMD: 0 warnings from one analysis.

- No warnings since build 25.
- New zero warnings highscore: no warnings since yesterday!

Test Result (no failures)

View Check Style Report contd

- You will see detailed information
- You can click individual tabs and links to see details

Checkstyle Warnings - New Warnings

Summary

Total	High Priority	Normal Priority	Low Priority
18	18	0	0

Details

Categories | Types | Warnings | Details

Category	Total	Distribution
Blocks	3	
Checks	1	
Design	1	
Javadoc	3	
Regexp	5	
Whitespace	5	
Total	18	

Activity

- Install Find Bugs Plugin
- Configure Maven pom for maven-findbugs plugin
- Configure a job and run build