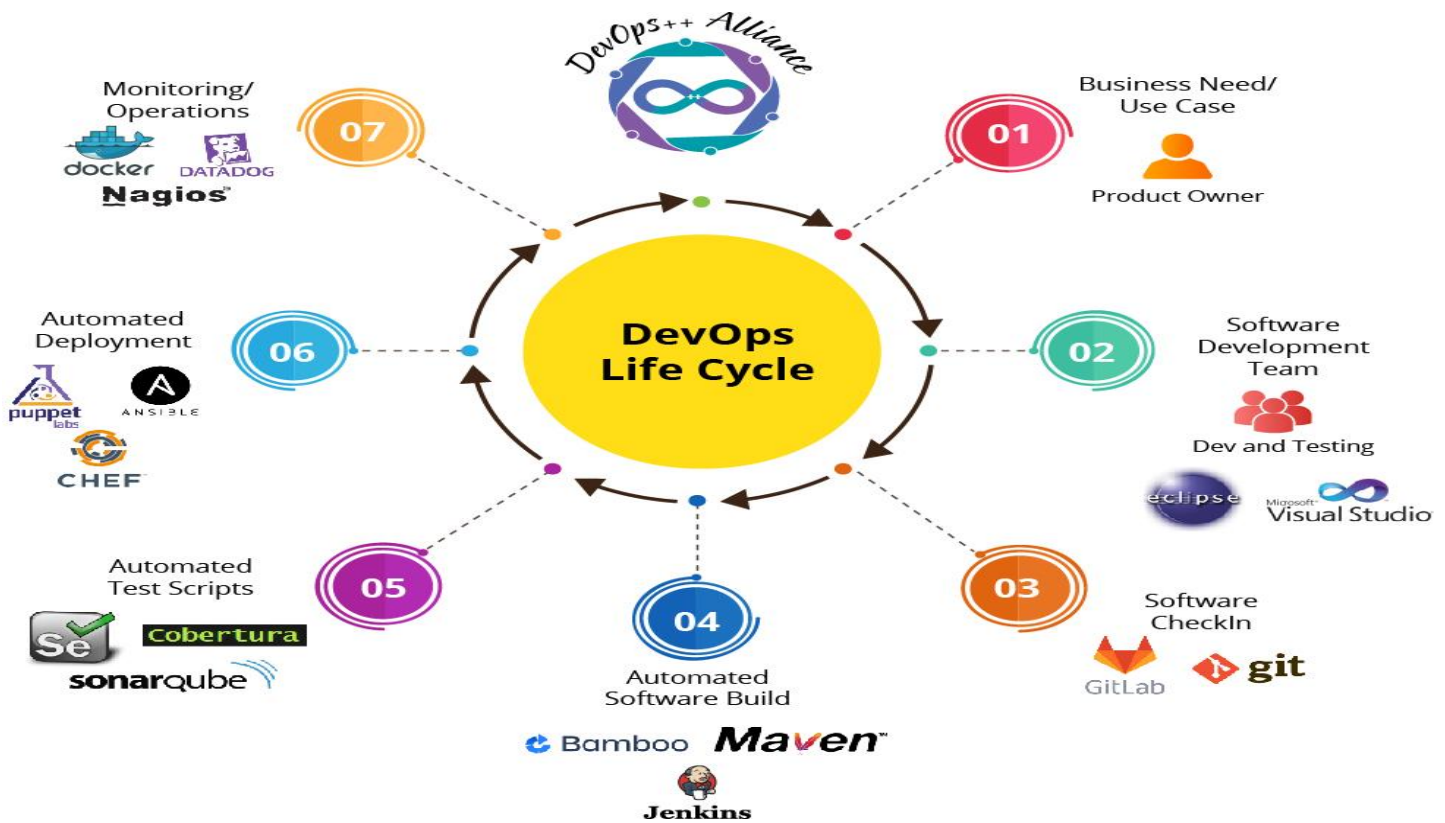


Calculator Assignment, building and deploying using DevOps tools and culture


by Shubham Aggarwal (MS2019017)



DevOps Pipeline

The plan is to automate the integration and deployment chain which includes source control management, continuous integration and continuous deployment. The cycle would include pushing latest changes to git (GitHub), building code includes using tool (maven), pushing build docker image on to docker hub, this part of continuous automation is called continuous integration. Then we integrate rundeck with jenkins, rundeck here does the continuous deployment. Rundeck pulls the pushed docker image from docker hub and deploys it on a docker container. Tools used includes

- SCM – GitHub - <https://github.com/shubhamaggarwal890/calculator-devops>
- Building – maven
- Docker image - <https://hub.docker.com/r/shubhamaggarwal890/calculator-devops>
- Continuous Integration – Jenkins
- Continuous deployment - rundeck


 Search or jump to... Pull requests Issues Marketplace Explore


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Repository name *

 shubhamaggarwal890

 / calculator-devops 

Great repository names are short and memorable. Need inspiration? How about **bookish-winner**?

Description (optional)

his is a calculator based project built using various tools of devops, including jenkins, docker, rundeck, ELK etc

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

shubhamaggarwal890 / calculator-devops

Unwatch

1

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security 0

Insights

Settings

This is a calculator based project built using various tools of devops, including jenkins, docker, rundeck, ELK etc

Manage topics

1 commit

1 branch

0 packages

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

shubhamaggarwal890 Initial commit

README.md Initial commit

README.md

calculator-devops

This is a calculator based project built using various tools of devops, including jenkins, docker, rundeck, ELK etc

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/shubhamaggarwal890

Download ZIP

Source Control Management - 1

Creating new repository on to <https://www.github.com>. This includes adding repository name and description. The repository name should be unique to the signed in user. The similar is done and one such repository is created for the calculator dev-ops project and can be found at <https://github.com/shubhamaggarwal890/calculator-devops>. The SCM handles our code and is used to connect as input to Jenkins. Other SCM are gitlab, bitbucket.

```
shubham@machine: ~/Documents/IIITB/Software Production Engineering
File Edit View Search Terminal Help
shubham@machine:~/Documents/IIITB/Software Production Engineering$ git clone https://github.com/shubhamaggarwal890/calculator-devops.git
Cloning into 'calculator-devops'... https://github.com/shubhamaggarwal890/calculator-devops
shubhamaggarwal890 / calculator-devops
Unwatch 1 Star 0 Fork 0
Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings
```

```
shubham@machine: ~/Documents/IIITB/Software Production Engineering
File Edit View Search Terminal Tabs Help
shubham@machine:~/Documents/IIITB/Software Production Engineering$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java):
  Selection Path Priority Status
  -----
  0 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 auto mode
  1 /usr/lib/jvm/java-11-openjdk-amd64/bin/java 1111 manual mode
  * 2 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java 1081 manual mode

Press <enter> to keep the current choice[*], or type selection number: 2
shubham@machine:~/Documents/IIITB/Software Production Engineering$ java -version
openjdk version "1.8.0_252"
OpenJDK Runtime Environment (build 1.8.0_252-b09-1-18.04-b09)
OpenJDK 64-Bit Server VM (build 25.252-b09, mixed mode)
shubham@machine:~/Documents/IIITB/Software Production Engineering$
```

Source Control Management - 2

Now if you have already developed the project and want to push to yours create repo, you can initiate the current directory –

git init ./

git remote add origin <git_repo_url>

git push -f origin <branch_name>

Or if you creating a new project I suggest doing a **git clone <git_repo_url>**

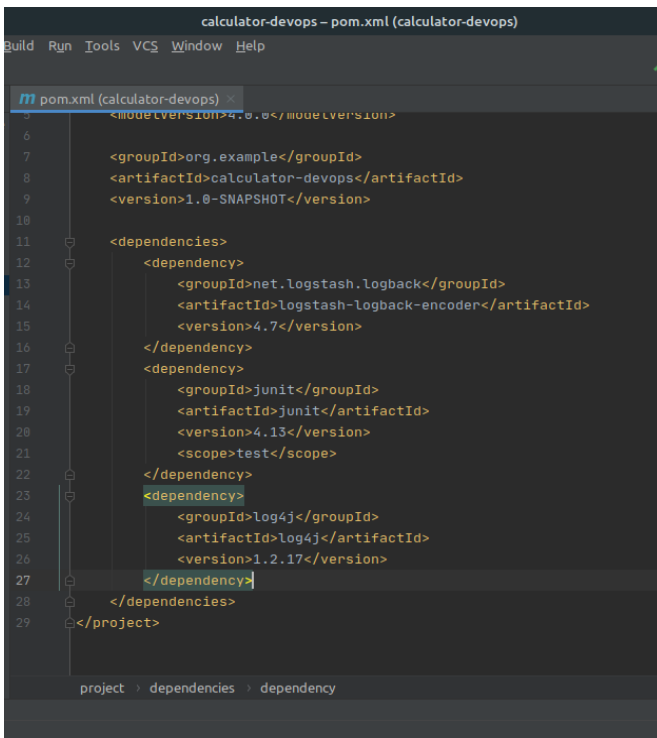
***Here we are using java 8 since rundeck supports only java 8, you can configure to java 8 with sudo update-alternatives --config java after installing it.**

To push the code on to repository follow following commands

git add <changed_files_path>

git commit -m “Commit message name”

git push (only if you are on master branch, otherwise I suggest merging your branch with master first, resolving conflicts and then do git push)



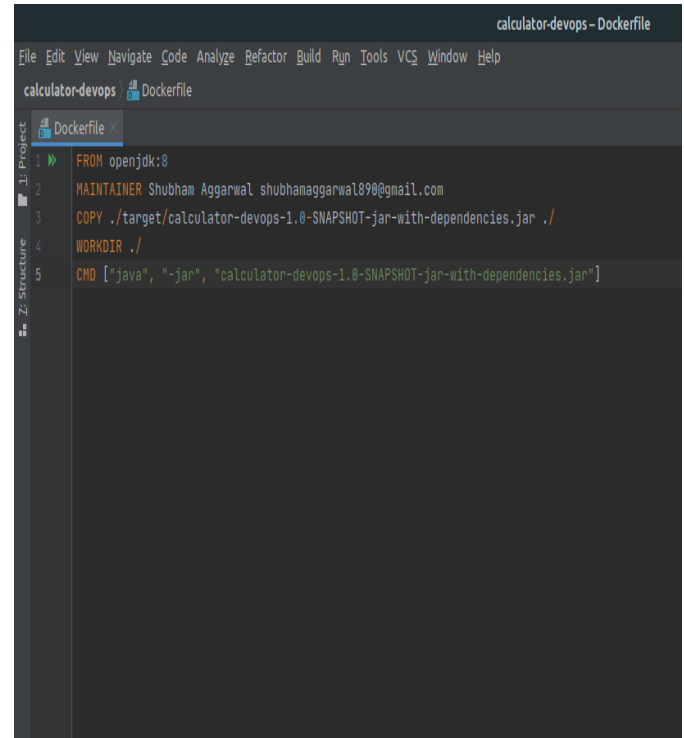
```
calculator-devops - pom.xml (calculator-devops)
Build Run Tools VCS Window Help

pom.xml (calculator-devops)
<modelVersion>4.0.0</modelVersion>

<groupId>org.example</groupId>
<artifactId>calculator-devops</artifactId>
<version>1.0-SNAPSHOT</version>

<dependencies>
  <dependency>
    <groupId>net.logstash.logback</groupId>
    <artifactId>logstash-logback-encoder</artifactId>
    <version>4.7</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
  </dependency>
</dependencies>
</project>

project > dependencies > dependency
```

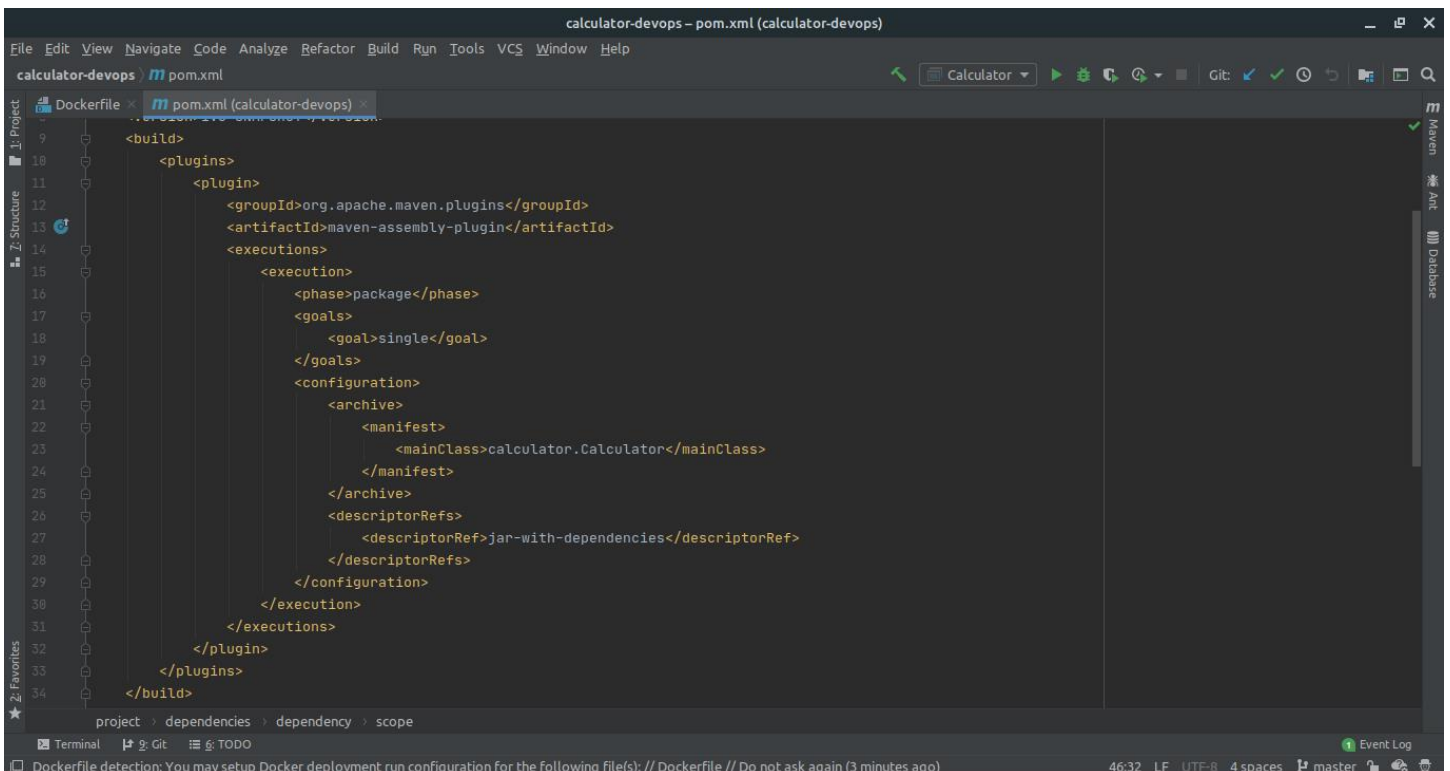


```
calculator-devops - Dockerfile
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

calculator-devops Dockerfile

Dockerfile
1 FROM openjdk:8
2 MAINTAINER Shubham Aggarwal shubhamaggarwal89@gmail.com
3 COPY ./target/calculator-devops-1.0-SNAPSHOT-jar-with-dependencies.jar ./
4 WORKDIR ./
5 CMD ["java", "-jar", "calculator-devops-1.0-SNAPSHOT-jar-with-dependencies.jar"]

1: Project
2: Structure
```



```
calculator-devops - pom.xml (calculator-devops)
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

calculator-devops pom.xml
Dockerfile pom.xml (calculator-devops)

<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-assembly-plugin</artifactId>
      <executions>
        <execution>
          <phase>package</phase>
          <goals>
            <goal>single</goal>
          </goals>
          <configuration>
            <archive>
              <manifest>
                <mainClass>calculator.Calculator</mainClass>
              </manifest>
            </archive>
            <descriptorRefs>
              <descriptorRef>jar-with-dependencies</descriptorRef>
            </descriptorRefs>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>

project > dependencies > dependency > scope

Calculator
Terminal Git TODO
Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): // Dockerfile // Do not ask again (3 minutes ago)
46:32 LF UTF-8 4 spaces master
```

Development and Software Build

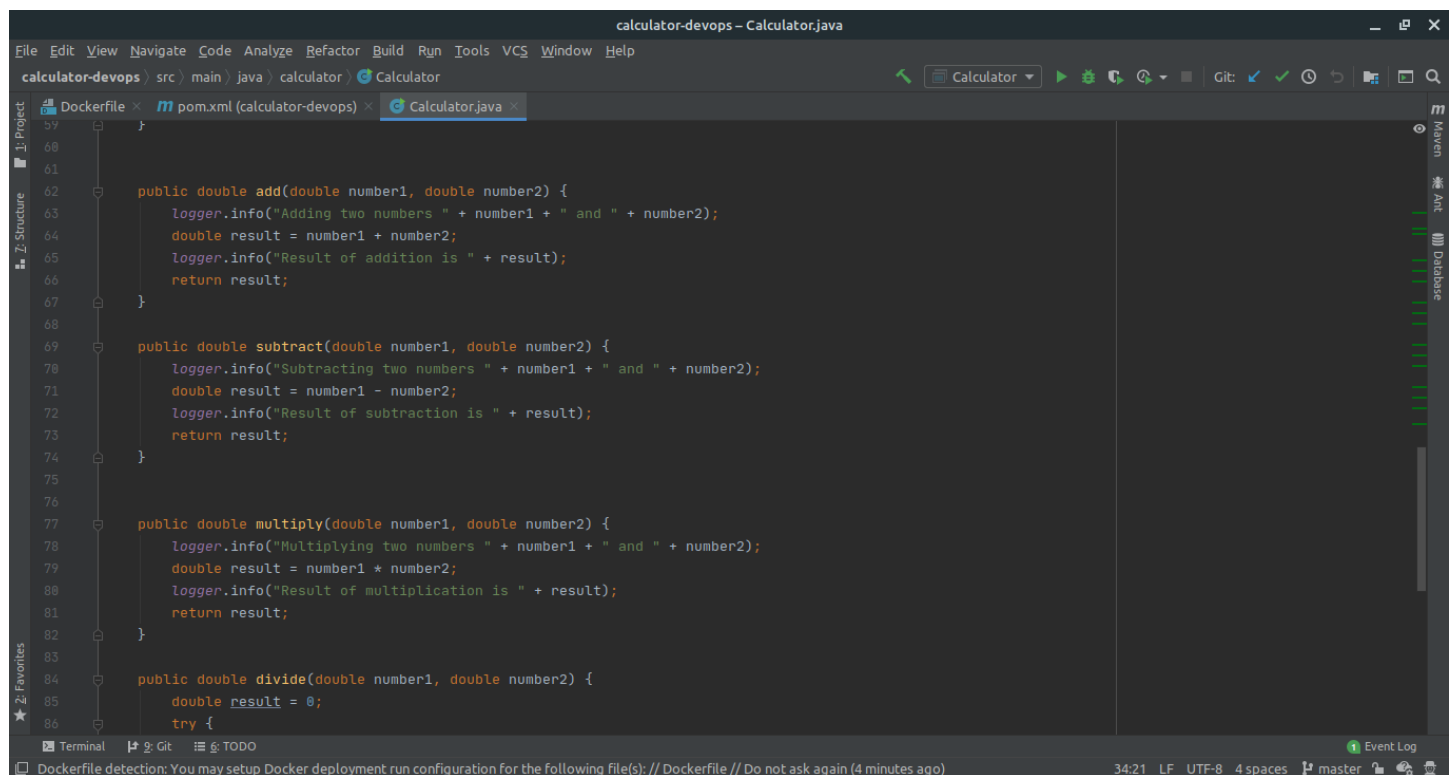
Here to build the code I'm using maven, maven is used to resolve dependencies to the code.

Dependencies are external libraries which we use to integrate it with in our code. For development environment I used IntelliJ. So started with a maven project and added basic functionalities of the calculator program. Also **added log4j.properties which is a must file if using log4j apache jar**. The

log4j.properties handles projects log functionalities, be its structure or where it should be stored or append or create a new file based on date and time. I also **created my own rollingappender file which creates missing directory and file if the file or directory not present within the specified path.** While adding the log4j I also added necessary **Dockerfile which would be used to create docker image of the build.** The docker file tells the build should be built on what image, **here it is openjdk 8,** after that we copy the created jar file and copy it to the working directory, and what command **should run when container is running.** For unit testing I have included junit dependency and written true positive and false positive cases to test while we build our project.

***The necessary thing to do is to add manifest within pom.xml which tells java to where to look for main class in project for an executable jar and since we are using dependencies here I also made it a jar with dependencies.**

You can also add Jenkins pipeline script over here in your git repository.



```
calculator-devops - Calculator.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
calculator-devops \src\main\java\calculator\ Calculator
Dockerfile x pom.xml (calculator-devops) x Calculator.java x
59
60
61
62 public double add(double number1, double number2) {
63     logger.info("Adding two numbers " + number1 + " and " + number2);
64     double result = number1 + number2;
65     logger.info("Result of addition is " + result);
66     return result;
67 }
68
69 public double subtract(double number1, double number2) {
70     logger.info("Subtracting two numbers " + number1 + " and " + number2);
71     double result = number1 - number2;
72     logger.info("Result of subtraction is " + result);
73     return result;
74 }
75
76
77 public double multiply(double number1, double number2) {
78     logger.info("Multiplying two numbers " + number1 + " and " + number2);
79     double result = number1 * number2;
80     logger.info("Result of multiplication is " + result);
81     return result;
82 }
83
84 public double divide(double number1, double number2) {
85     double result = 0;
86     try {
87
88     }
89 }
```

Terminal Git TODO Event Log
Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): // Dockerfile // Do not ask again (4 minutes ago) 34:21 LF UTF-8 4 spaces master

The left screenshot shows the Jenkins 'Create new job' page. The job name 'Calculator devOps Pipeline' is entered. Below the name field, there are four options: Freestyle project, Maven project, Pipeline (selected), and External Job. The Pipeline option is highlighted with a blue border.

The right screenshot shows the 'Advanced Project Options' tab for the 'Calculator-DevOps Pipeline'. The 'Pipeline' section is active, showing a Groovy script for a pipeline. The script is as follows:

```

1 pipeline {
2   environment {
3     registry = 'shubhanaggarwal890/calculator-devops'
4     registryCredential = 'docker-hub'
5     dockerImage = ''
6     dockerImageLatest = ''
7   }
8   agent any
9   stages {
10    stage('Cloning Git') {
11      steps {
12        git 'https://github.com/shubhanaggarwal890/calculator-devops.git'
13      }
14    }
15    stage('Build Executable Jar'){
16      steps {
17        sh 'mvn clean test package'
18      }
19    }
20  }
21 }

```

At the bottom of the right screenshot, there are 'Save' and 'Apply' buttons, and a checkbox labeled 'Use Groovy Sandbox' which is checked.

Environment setup - Jenkins

The development process is complete and we have successfully pushed the code on to git, now we setup Jenkins and other plugins of it.

Make sure to add Jenkins to docker group, so that Jenkins can use docker for build docker image.

sudo usermod -aG docker Jenkins, you can verify it with **sudo grep jenkins /etc/gshadow**

We can now start Jenkins with, by command **sudo systemctl start jenkins**, jenkins starts at port number 8080, so login on to <http://localhost:8080> onto your browser.

Now we manage the plugins of jenkins under manage plugins in manage jenkins. We download Build pipeline plugin, Docker plugin, GitHub, Maven integration plugin, Rundeck plugin.

After its done downloading, jenkins will restart and now you can create a new job for jenkins, enter jenkins job name and choose pipeline as job functionality. Properties of pipeline is it is script based and each stage of pipeline script runs one after another. Making it perfect for continuous integration and then deployment. Properties of continuous integration is it includes SCM, unit testing and integration testing.

Jenkins > configuration

Build Executor Status

1 Idle

2 Idle

Instances

Invalidate cache

Rundeck job cache configuration

Name

Rundeck Calculator DevOps

URL

http://localhost:4440/

Login

admin

Password

Concealed

Change Password

Auth Token

API Version

Your Rundeck instance is alive, and your credentials are valid !

Test Connection

Delete Rundeck

Add Rundeck

List of Rundeck instances

Maven Project Configuration

Save

Apply

Jenkins

search

monitor 1

admin

log out

Jenkins > Credentials > System > Global credentials (unrestricted) >

Back to credential domains

Add Credentials

Kind

Username with password

Scope

Global (Jenkins, nodes, items, all child items, etc)

Username

shubhamaggarwal890

Password

.....

ID

docker-hub

Description

Pipelining Jenkins with dockerhub

OK

Challenges and Solutions - Jenkins

While setting up Jenkins there were many challenges I faced which included incorrect Docker ID, not connecting to Git or Rundeck, and Docker not working. Here's the summary of it all –

1. **Got permission denied while trying to connect to Docker daemon** – This usually happens if your Docker is not running, you can check it via **sudo systemctl status docker**, it also occurs if your

jenkins user doesn't have docker permission which can be easily given via **sudo usermod -aG jenkins docker**. Don't forget to restart jenkins job after that using **sudo systemctl restart jenkins**

2. **Setting credentials for docker hub** – In the jenkins dashboard add credentials to your docker hub repository and set a unique id to it, this id should be equal to docker with Registry credentials id in pipeline script.
3. **Rundeck instance not running** – If rundeck instance is not alive, this can be configured under jenkins configuration. Add rundeck instance name along with url on which rundeck is running and its credentials.
4. **Rundeck pipeline script not validated** – This I couldn't resolve, I wasn't able to run the proper pipeline script for rundeck, so I created a new job in jenkins and under post build, I created a new instance of rundeck and entered rundeck job id and other related details. And added that jenkins job build script to be built within the pipeline script.
5. **Maven not working** – In many cases maven integration plugin doesn't work, while building the code it would throw the error of **mvn not found**. While we build it on intellij it has inbuilt maven so building over it doesn't require maven to be installed on your machine. So if maven is not installed then mvn won't work on jenkins because its plugin is a maven integration plugin so maven should be there on your host machine, which can be downloaded via **sudo apt maven** (debian only).

Jenkins > Calculator-DevOps Pipeline >

General Build Triggers **Advanced Project Options** Pipeline

Advanced Project Options

Advanced...

Pipeline

Definition Pipeline script

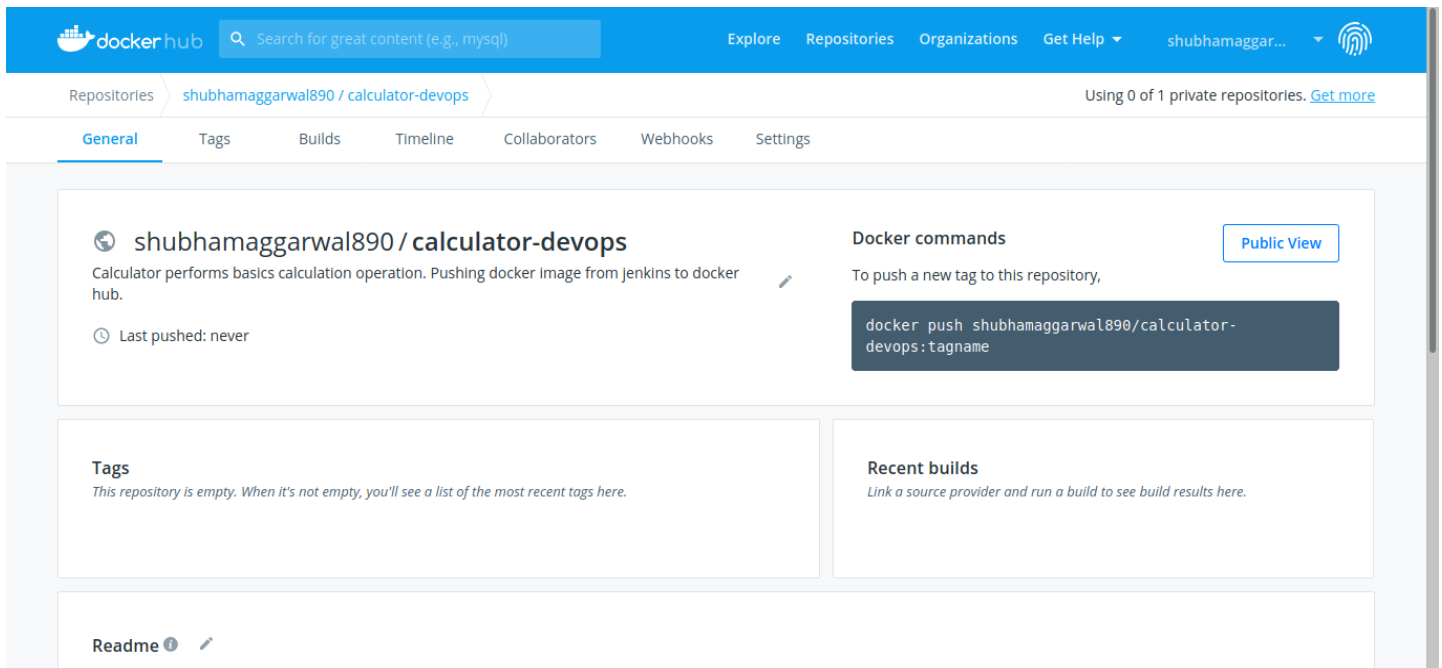
Script

```
19 }
20 stage('Building Image') {
21   steps {
22     script {
23       dockerImageLatest = docker.build registry + ":latest"
24     }
25   }
26 }
27 stage('Deploy Image') {
28   steps {
29     script {
30       docker.withRegistry( '', registryCredential ) {
31         dockerImageLatest.push()
32       }
33     }
34   }
35 }
36 stage('Execute Rundeck job') {
37   steps {
```

☒ Use Groovy Sandbox

Pipeline Syntax

Save Apply



Docker Setup

Docker is platform which provides OS level virtualization to deliver software as packages. So the plan is to create a docker image of jar file created after build process and push the latest image on to docker hub. The pushed procedure would be done after every build which would include removing preciously pushed docker image and replace with latest built one. The pushed images can be found at <https://hub.docker.com/r/shubhamaggarwal890/calculator-devops>

To run docker on your machine, either run it as root or as user. To run it as user you have to enter user to the group of docker which can be done using **sudo usermod -aG docker <username>**

Run the docker on to terminal via command **sudo systemctl start docker**

To pull the docker image use command **docker pull <docker_image_name:tag>**

To push the docker image use command **docker push <username>/<repository_name>:tagname**

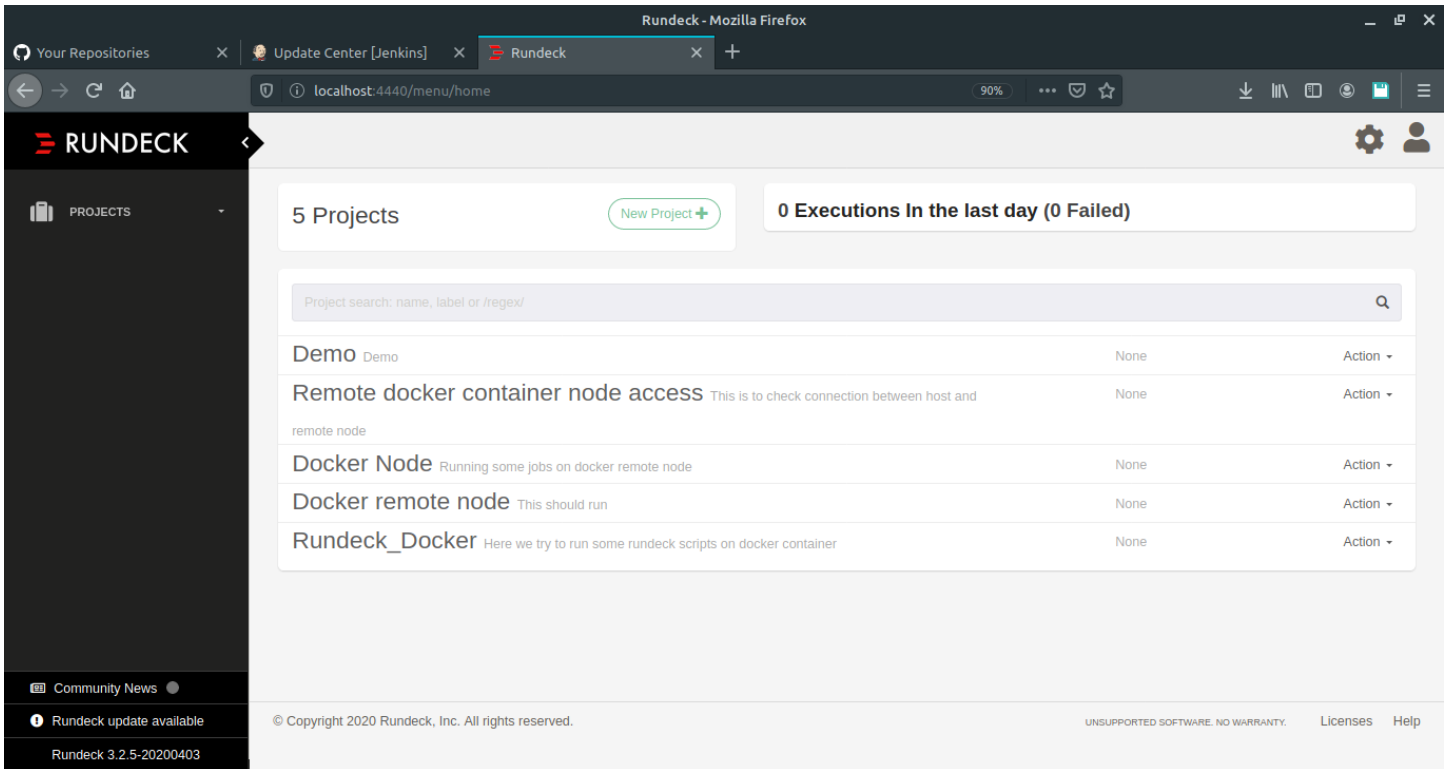
The docker image is pushed from jenkins after the build stage and pulled on from rundeck job. The docker image can be run using command **docker run -i -t <image_name>**

Challenges and Solution – Docker

While experimenting and deciding the architecture of the project there were some hurdles I encountered. Here's the list of it -

1. **Docker within a docker can lead to docker sock issue** – So earlier the plan was to connect rundeck with a docker container started over the image of Ubuntu and then run **docker pull <calculator_image>**. This gave me constant error of **Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.40/images/json: dial unix /var/run/docker.sock: connect: permission denied** even though I was logged in as root. This issue was then resolved when I ran docker image as **docker run --privileged -t -i <image_name>**
2. **Docker within a docker can lead to file system issues** – After the first docker issue was resolved, I was able to successfully pull the docker image, but after running it, **docker run -t -i <image_name>** within a docker container I was prompted with **docker: Error response from daemon: error creating aufs mount to /var/lib/docker/aufs/mnt/be6e52c353fa0fe27d9c30b6c14e0b596ebf5fa411a93b4661d66c867756a012-init**. After discussion with professor we both concluded that it's not safe to docker within docker and could lead to file system errors and run docker locally, on a local node.
3. **Docker within a docker official image doesn't support ssh** – Now to go a step further I tried docker within a docker official image from docker hub. But upon pulling I realized that you cannot create ssh instance over it and then further cannot connect to rundeck. To resolve such issues, I had three option **either run docker image locally which would still create its own container**, or **connect to a third party node platform such as azure, aws or google cloud and pull docker image over it and run the image over there**, or **change the docker image build of the code over Ubuntu docker image and openjdk and run our created docker image over it**. I chose to pull docker image on local node from rundeck and run it over the created container.

```
shubham@machine: ~  
File Edit View Search Terminal Tabs Help  
shubham@machine: ~/Documents/Tools/logstash-7.6.2 x shubham@machine: ~/Documents/IIITB/Software Produc... x shubham@machine: ~ x  
shubham@machine:~$ sudo systemctl start rundeckd.service  
[sudo] password for shubham:  
shubham@machine:~$ sudo systemctl status rundeckd.service  
● rundeckd.service - LSB: rundeck job automation console  
   Loaded: loaded (/etc/init.d/rundeckd; generated)  
   Active: active (running) since Tue 2020-05-05 17:38:04 IST; 4s ago  
     Docs: man:systemd-sysv-generator(8)  
   Process: 8482 ExecStart=/etc/init.d/rundeckd start (code=exited, status=0/SUCCESS)  
 Main PID: 8520 (java)  
    Tasks: 15 (limit: 4915)  
   CGroup: /system.slice/rundeckd.service  
           └─8520 java -Drundeck.jaaslogin=true -Djava.security.auth.login.config=/etc/rundeck/jaas-loginmodule.conf -Dloginmodule.name=RDpropertyfilelogin -Drundeck.conf  
Product requirements  
May 05 17:38:03 machine systemd[1]: Starting LSB: rundeck job automation console...  
May 05 17:38:04 machine rundeckd[8482]: * Starting rundeckd  
May 05 17:38:04 machine rundeckd[8482]: Created new User (vbehar), last modified on Jun 06, 2011  
May 05 17:38:04 machine systemd[1]: Started LSB: rundeck job automation console.  
lines 1-14/14 (END) titles  
CHILD PAGES  
You are viewing an old version of this page. View the current version.
```



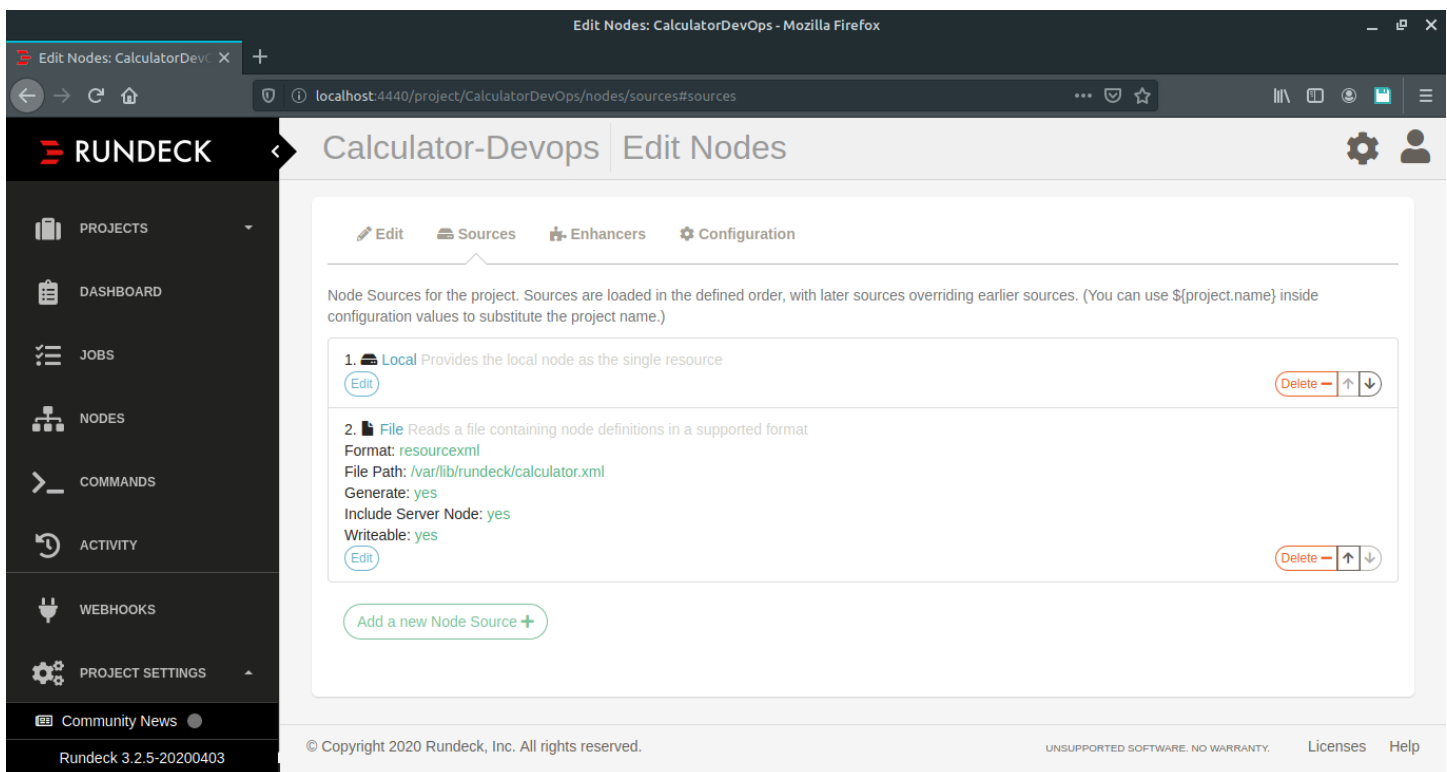
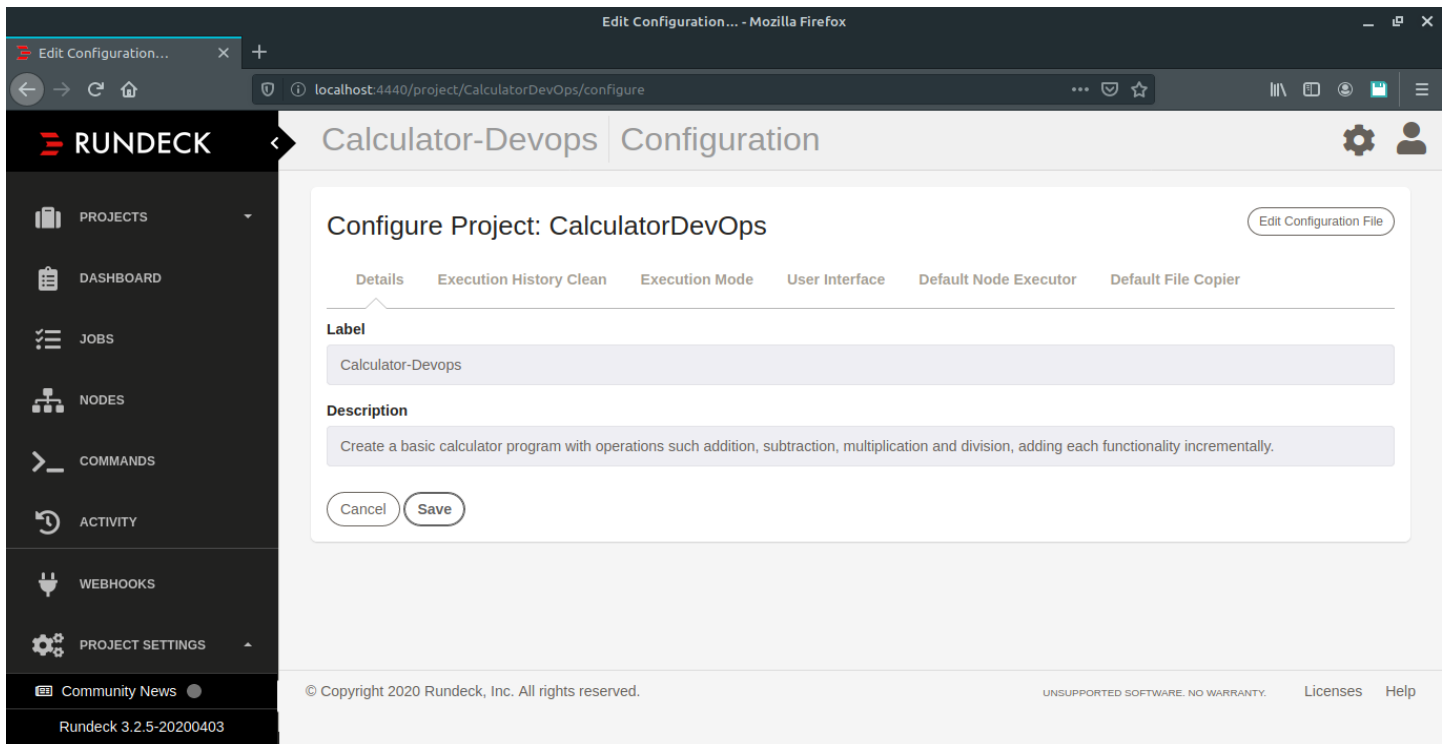
Rundeck Setup - 1

Rundeck is tool for continuous deployment where we can configure any number of nodes to a project and connect them via ssh. After successfully connecting you can create any job related to the project and run commands or scripts or ansible playbooks over the set of nodes. Here we connect rundeck with jenkins pipeline and rundeck in sequence runs the created job and pulls the docker image.

Since rundeck would be used to pull docker image so rundeck must have docker permission so here we add rundeck to docker group using command **sudo usermod -aG docker rundeck**.

Now run rundeck start with **sudo systemctl start rundeckd**

Rundeck runs at http://localhost:4440



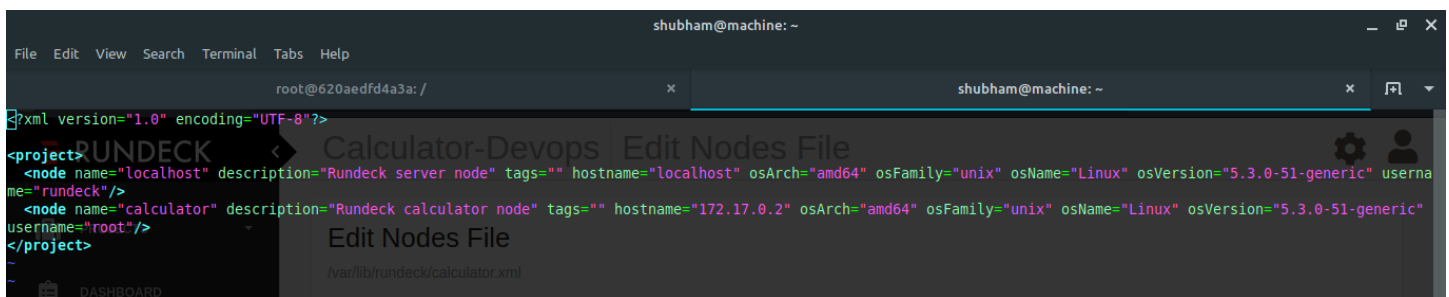
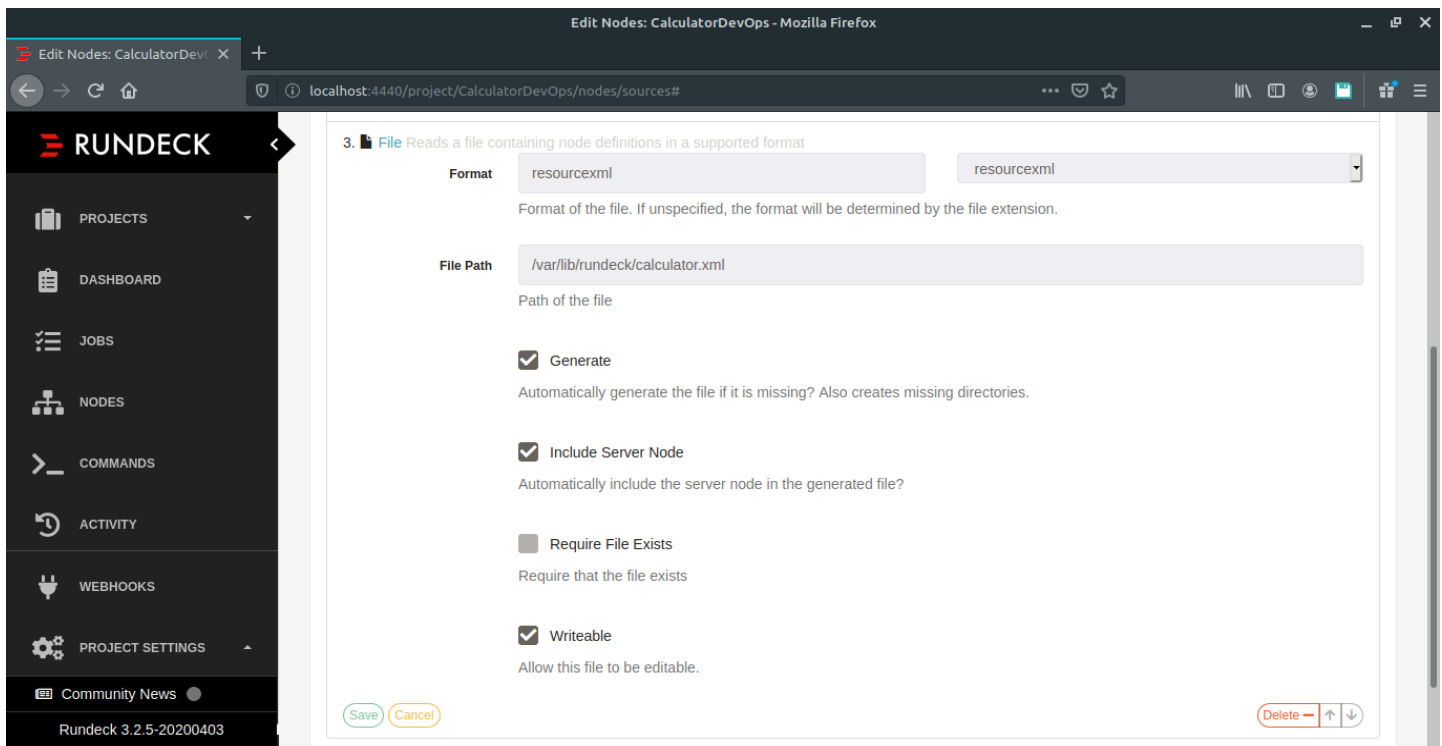
Rundeck Setup - 2

In rundeck we create a project and to that project you can add description and other project details. After creating project one has to add nodes to the project. Nodes can be added under project settings and edit nodes.

After configuring the node, I created a job and added the node configured to it and also added the script what this job would actually do. The motto of job was to pull the docker image. So accordingly the workflow was set to it and job description was set, the job id was then fed to the rundeck build created in jenkins so that jenkins can then trigger rundeck job after finishing of earlier stages. To test if the node is set properly you can test it under commands in project.

The screenshot shows the RunDeck interface for a project named 'Calculator-Devops'. The left sidebar contains navigation links: PROJECTS, DASHBOARD, JOBS, NODES, COMMANDS, ACTIVITY, WEBHOOKS, and PROJECT SETTINGS. The main content area is titled 'Edit Job: Pulling docker image and running it' with a job ID 'c5ce4418-aebb-4d4d-abd7-641e16e33f82'. It features tabs for Details, Workflow, Nodes, Schedule, Notifications, and Other. The 'Options' section shows 'No Options' with an 'Add an option' button. The 'Workflow' section includes a 'If a step fails:' dropdown set to 'Run remaining steps before failing', a 'Strategy:' dropdown set to 'Node First', and an 'Explain' link. Below this is a 'Global Log Filters' section with an 'add' button. At the bottom, a list of steps is shown, with the first step being 'docker pull shubhamaggarwal890/calculator-devops:latest' with a sub-label 'Pulling docker image'. There are 'Undo', 'Redo', and 'Revert All Changes' buttons at the bottom of the workflow section.

The screenshot shows the RunDeck interface for the same 'Calculator-Devops' project, but in a 'Run' mode. The job title is 'Pulling docker image and running it' with the same job ID. The description is 'From docker pull the docker image and pull new image from docker hub'. The interface includes a 'Follow execution' section with a 'Nodes' dropdown and a 'Run Job Now' button. Below this is a 'Stats' section with a 'Activity' tab. The 'Stats' section displays '1 EXECUTIONS', a '100% SUCCESS RATE', and a '20s AVG DURATION'.



Setting nodes on Rundeck – Running docker image on docker container or on cloud.

Setting nodes on rundeck was a little tricky for a while but after a fixed set of commands and lot of experimenting it was well cracked. So it starts with project settings > Edit nodes. Under Edit nodes in source tab we add resource xml file by adding a new node source. Set the path of the required details. Now either from rundeck itself or terminal modify the resource xml file by adding node details such as hostname and username.

On the attached docker container or running third party cloud machine. Either create a new user by command **useradd username or modify to login as root via ssh**. If doing on docker container make sure you have installed openssh-server by command **apt update && apt install openssh-server**.

Now make change in **/etc/ssh/sshd_config** and make **PermitRootLogin yes**

Then restart ssh by **service ssh restart**, you can also use systemctl provided it is installed in container.

Then generated keys for the ssh, for you to login from host machine without password you can do this by command **ssh-keygen**

Now on from host machine do the same generate the keys if not generated by **ssh-keygen**,

Then copy the keygen to docker container by **ssh-copy-id [root@172.17.0.x](#)** or **ssh-copy-id username@ipaddress** for your third party or within network another machine.

After this copy **your host private key to /etc/lib/rundeck/.ssh**, via command **cp ~/.ssh/* /var/lib/rundeck/.ssh**.

Now the copied ssh files would have user and group owner as not rundeck we have to give permissions so that rundeck can access it too. This can be done using command **sudo chown rundeck:rundeck /var/lib/rundeck/.ssh**

Your rundeck can now access nodes present within resource xml file.

RUNDECK Calculator-Devops

Nodes **All Nodes** .*

Browse Result: 2 Nodes Actions

NODE	TAGS	USER @ HOST
> calculator Rundeck calculator node		root @ 172.17.0.2
> localhost server Rundeck server node		rundeck @ localhost

PROJECTS
DASHBOARD
JOBS
NODES
COMMANDS
ACTIVITY
WEBHOOKS
PROJECT SETTINGS
Community News
Rundeck 3.2.5-20200403

Challenges and Solution – Rundeck

While working on rundeck many challenges were encountered. Some hurdles could be resolved and some couldn't because of architecture constraints, here's the list of it –

1. **Rundeck cannot take input from a running job** – So I tried couple of things here I wanted to run my docker image pulled from docker hub on to a rundeck job. But since Rundeck doesn't support input from job command desk it would throw an error **the input device is not a tty**. This can be resolved if we pass parameters via echo example **echo "1" && docker run -i -t <docker image name>** but again it's not a convenient method for multiple inputs. You can run a docker image provided it's doesn't ask for any input.
2. **Rundeck cannot access docker, doesn't have permission for docker.sock** – This usually occurs if the docker is not running which can be checked via **sudo systemctl status docker** and make sure rundeck is in group of docker, which can be checked via **sudo grep docker /etc/gshadow** if not this can be done using command **sudo usermod -aG docker rundeck** and then restart **sudo systemctl restart docker**
3. **SSH configuration and not able to connect to node** – This issue usually occurs because of miss in steps of ssh configuration while setting up nodes, check for **/etc/ssh/sshd_config** file for **permit root login**, add **ssh-keygen** of host machine to docker node via **ssh-copy-id root@<docker ip address>**, copy **~/.ssh/*** to **/var/lib/rundeck/.ssh** and **set user and group owner of /var/lib/rundeck/.ssh as rundeck**
4. **apt-get-add doesn't work on rundeck job** – So what I tried was to setup docker and other relative environment on docker container via rundeck job surprisingly apt-get-add doesn't work over it and throws an error, to setup nodes you can either use chef cookbook or ansible playbook.

Jenkins

Calculator-DevOps Pipeline

enable auto refresh

Back to Dashboard

Status

Changes

Build Now

Delete Pipeline

Configure

Full Stage View

Rename

Pipeline Syntax

Pipeline Calculator-DevOps Pipeline

Create a basic calculator program with operations such addition, subtraction, multiplication and division, adding each functionality incrementally.

 Use DevOps tools to create a pipeline. Create a report describing each step such that following those instructions the same result can be replicated also add screenshots of outcomes and configurations, along with the links of your GitHub, Docker Hub etc.

edit description

Disable Project

Recent Changes

Stage View

Build History

find

#24

May 7, 2020 12:16 PM

#23

May 6, 2020 11:01 PM

#22

May 6, 2020 9:46 PM

#16

May 6, 2020 8:14 PM

#7

May 6, 2020 12:19 AM

#4

May 6, 2020 12:06 AM

May 07

12:16

1 commit

May 06

23:01

No Changes

Average stage times:

(Average full run time: ~2min 2s)

Cloning Git	Build Executable Jar	Building image	Deploy Image	Execute Rundeck Job
6s	14s	30s	45s	25s
4s	10s	1min 46s	38s	32s
15s	19s	10s	49s	19s

Jenkins pipeline in function

After we are done setting up every aspect of our DevOps tool chain, we may now build the jenkins job. This job for now can be manually triggered but for other SCM like gitlab it can be triggered based on events such as new push or new pull but for GitHub you might have set up a webhook. This can be done via port forwarding; you have make your jenkins port 8080 open on wide internet so it can be triggered based on event.

For now, we'll skip it and we'll build it manually, every stage of the pipeline script works in sequence which includes

- Cloning Git
- Build Executable Jar
- Building image (docker)
- Deploy Image
- Execute Rundeck job

After successful jenkins job build you'll find that maven test runs successfully while writing logs to /var/log/calculator-devops/calculator.log, building new docker image and pushing it on to docker hub and then rundeck pull the docker image on the local node.

```
root@machine: ~
File Edit View Search Terminal Tabs Help

root@machine: ~ x root@machine: ~ x [?] v

root@machine:~# docker run -i -t shubhamaggarwal890/calculator-devops
Calculator-DevOps, Choose to perform operation
Press 1 to Add
Press 2 to Subtract
Press 3 to Multiply
Press 4 to Divide
Press any other key to exit
Enter your choice:
2
Enter the first number : 5
Enter the second number : 1
[INFO ] 2020-05-10 16:48:25,243 method:calculator.Calculator.subtract(Calculator.java:70) - Subtracting two numbers 5.0 and 1.0
[INFO ] 2020-05-10 16:48:25,276 method:calculator.Calculator.subtract(Calculator.java:72) - Result of subtraction is 4.0
Subtraction result is : 4.0
Calculator-DevOps, Choose to perform operation
Press 1 to Add
Press 2 to Subtract
Press 3 to Multiply
Press 4 to Divide
Press any other key to exit
Enter your choice:
█
```

```
root@machine: ~
File Edit View Search Terminal Tabs Help

root@machine: ~ x root@machine: ~ x [?] v

root@machine:~# tail -f /var/log/calculator-devops/calculator.log
[INFO ] 2020-05-07 12:28:23 [ calculator.Calculator.subtract(Calculator.java:72) ] - Result of subtraction is -1.1
[INFO ] 2020-05-07 12:28:23 [ calculator.Calculator.multiply(Calculator.java:78) ] - Multiplying two numbers 2.0 and 2.0
[INFO ] 2020-05-07 12:28:23 [ calculator.Calculator.multiply(Calculator.java:80) ] - Result of multiplication is 4.0
[INFO ] 2020-05-07 12:28:23 [ calculator.Calculator.multiply(Calculator.java:78) ] - Multiplying two numbers 2.1 and 3.2
[INFO ] 2020-05-07 12:28:23 [ calculator.Calculator.multiply(Calculator.java:80) ] - Result of multiplication is 6.720000000000001
[INFO ] 2020-05-07 23:18:09 [ calculator.Calculator.subtract(Calculator.java:70) ] - Subtracting two numbers 7.0 and 5.0
[INFO ] 2020-05-07 23:18:09 [ calculator.Calculator.subtract(Calculator.java:72) ] - Result of subtraction is 2.0
[INFO ] 2020-05-07 23:18:26 [ calculator.Calculator.divide(Calculator.java:87) ] - Dividing two numbers 1.0 and 0.0
[ERROR] 2020-05-07 23:18:26 [ calculator.Calculator.divide(Calculator.java:101) ] - Number cannot be divided by zero Case of Positive Infinity 1.0/0.0
[INFO ] 2020-05-07 23:18:26 [ calculator.Calculator.divide(Calculator.java:103) ] - Result of dividing is Infinity
█
```

Execution of Docker image

After we are done pulling docker image from rundeck on local node we can run it as **docker run -i -t shubhamaggarwal890/calculator-devops**, but the logs won't be created on to your local machine since the docker is running on a docker image container. But jenkins maven build logs can be seen through from described path of logs within log4j.properties file.

If you do only docker run shubhamaggarwal890/calculator-devops it would throw a scanner error because it won't find any input so make sure you make it interactive and attached to terminal.

```

input
{
    file {
        path => "var/logs/calculator-devops/calculator.log"
    }
}

output
{
    elasticsearch {
    }

    stdout {
        codec => dots
    }
}

```

```

shubham@machine:~/Documents/Tools$ ./elasticsearch-7.6.2/bin/elasticsearch
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
[2020-05-10T22:33:05,186][INFO ][o.e.e.NodeEnvironment ] [machine] using [1] data paths, mounts [ [/dev/sda2]], net usable_space [113.8gb], net total_space [915.3gb], types [ext4]
[2020-05-10T22:33:05,219][INFO ][o.e.e.NodeEnvironment ] [machine] heap size [990.7mb], compressed ordinary object pointers [true]
[2020-05-10T22:33:06,667][INFO ][o.e.n.Node ] [machine] node name [machine], node ID [VgJXrTDbTS0fUHTKpbaS7g], cluster name [elasticsearch]
[2020-05-10T22:33:06,667][INFO ][o.e.n.Node ] [machine] version[7.6.2], pid[22829], build[default/tar/ef48eb35cf30adf4db14086e8aabd0ef6fb113f/2020-03-26T06:34:37.794943Z], OS[Linux/5.3.0-51-generic/amd64], JVM[AdoptOpenJDK/OpenJDK 64-Bit Server VM/13.0.2/13.0.2+8]
[2020-05-10T22:33:06,668][INFO ][o.e.n.Node ] [machine] JVM home [/home/shubham/Documents/Tools/elasticsearch-7.6.2/jdk]
[2020-05-10T22:33:06,669][INFO ][o.e.n.Node ] [machine] JVM arguments [-Des.networkaddress.cache.ttl=60, -Des.networkaddress.cache.negative.ttl=10, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dio.netty.allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.locale.providers=COMPAT, -Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -Djava.io.tmpdir=/tmp/elasticsearch-7187029776826659886, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,pid,tags:filecount=32,filesize=64m, -XX:MaxDirectMemorySize=536870912, -Des.path.home=/home/shubham/Documents/Tools/elasticsearch-7.6.2, -Des.path.conf=/home/shubham/Documents/Tools/elasticsearch-7.6.2/config, -Des.distribution.flavor=default, -Des.distribution.type=tar, -Des.bundled_jdk=true]
[2020-05-10T22:33:13,527][INFO ][o.e.p.PluginsService ] [machine] loaded module [aggs-matrix-stats]
[2020-05-10T22:33:13,528][INFO ][o.e.p.PluginsService ] [machine] loaded module [analysis-common]
[2020-05-10T22:33:13,529][INFO ][o.e.p.PluginsService ] [machine] loaded module [flattened]
[2020-05-10T22:33:13,529][INFO ][o.e.p.PluginsService ] [machine] loaded module [frozen-indices]
[2020-05-10T22:33:13,530][INFO ][o.e.p.PluginsService ] [machine] loaded module [ingest-common]

```

```

shubham@machine:~/Documents/Tools$ ./logstash-7.6.2/bin/logstash -f ./logstash.conf
Sending Logstash logs to /home/shubham/Documents/Tools/logstash-7.6.2/logs which is now configured via log4j2.properties
[2020-05-10T22:37:43,055][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified
[2020-05-10T22:37:45,270][INFO ][logstash.runner ] Starting Logstash {"logstash.version":"7.6.2"}
[2020-05-10T22:37:54,430][INFO ][org.reflections.Reflections] Reflections took 2404 ms to scan 1 urls, producing 20 keys and 40 values
[2020-05-10T22:38:09,960][INFO ][logstash.outputs.elasticsearch][main] Elasticsearch pool URLs updated {:changes=>{:removed=>[], :added=>[http://127.0.0.1:9200/]}
[2020-05-10T22:38:16,842][WARN ][logstash.outputs.elasticsearch][main] Restored connection to ES instance {:url=>http://127.0.0.1:9200/}
[2020-05-10T22:38:17,873][INFO ][logstash.outputs.elasticsearch][main] ES Output version determined {:es_version=>7}
[2020-05-10T22:38:17,879][WARN ][logstash.outputs.elasticsearch][main] Detected a 6.x and above cluster: the 'type' event field won't be used to determine the document type {:es_version=>7}
[2020-05-10T22:38:20,727][INFO ][logstash.outputs.elasticsearch][main] New Elasticsearch output {:class=>"LogStash::Outputs::ElasticSearch", :hosts=>[http://127.0.0.1]}
[2020-05-10T22:38:20,803][INFO ][logstash.outputs.elasticsearch][main] Using default mapping template
[2020-05-10T22:38:20,880][WARN ][org.logstash.instrument.metrics.gauge.LazyDelegatingGauge][main] A gauge metric of an unknown type (org.jruby.specialized.RubyArrayOneObject) has been created for key: cluster_uuids. This may result in invalid serialization. It is recommended to log an issue to the responsible developer/development team
[2020-05-10T22:38:20,901][INFO ][logstash.javapipeline ] [main] Starting pipeline {:pipeline_id=>"main", "pipeline.workers">4, "pipeline.batch.size">125, "pipeline.batch.delay">50, "pipeline.max_inflight">500, "pipeline.sources">[/home/shubham/Documents/Tools/logstash.conf], :thread=>#<Thread:0x12f69a94 run>}
[2020-05-10T22:38:21,405][INFO ][logstash.outputs.elasticsearch][main] Attempting to install template {:manage_template=>{"index_patterns">"logstash-*", "version">60001, "settings">{"index.refresh_interval">"5s", "number_of_shards">1, "index.lifecycle.name">"logstash-policy", "index.lifecycle.rollover_alias">"logstash"}, "mapping">{"dynamic_templates">[{"message_field">{"path_match">"message", "match_mapping_type">"string", "mapping">{"type">"text", "norms">false}}, {"string_fields">{"match">"*", "match_mapping_type">"string", "mapping">{"type">"text", "norms">false}}, {"keyword">{"type">"keyword", "ignore_above">256}}]}, "properties">{"@timestamp">{"type">"date"}, "@version">{"type">"keyword"}, "geoip">{"dynamic">true, "properties">{"ip">{"type">"ip"}, "location">{"type">"geo_point"}, "latitude">{"type">"half_float"}, "longitude">{"type">"half_float"}}}}]}
[2020-05-10T22:38:31,716][INFO ][logstash.inputs.file ] [main] No sincedb path set, generating one based on the "path" setting {:sincedb_path=>/home/shubham/Documents/Tools/logstash-7.6.2/data/plugins/inputs/file/sincedb_e25e6794032acel186f63e6c0cf287f55, :path=>[/var/log/calculator-devops/calculator.log]}
[2020-05-10T22:38:31,769][INFO ][logstash.javapipeline ] [main] Pipeline started {"pipeline.id">"main"}
[2020-05-10T22:38:31,908][INFO ][filewatch.observingtail ] [main] START, creating Discoverer, Watch with file and sincedb collections
[2020-05-10T22:38:31,917][INFO ][logstash.agent ] Pipelines running {:count=>1, :running_pipelines=>[main], :non_running_pipelines=>[]}

```

Elastic Stack – Elastic Search, Logstash, Kibana

ELK stack makes the monitoring tool for any deployed software, it analyzes the logs and the same analysis can then be viewed on kibana dashboard.

To start with download elastic search, logstash and kibana from <https://www.elastic.co>

Run them side by side and feed the logstash your log set after configuring it config file.

In the config file we provide the details for logs. Here the config file logstash.conf

Elastic search starts at localhost:9200, and kibana starts at localhost:5601

Commands to run

- Elastic search - `./path_to_elastic_search/bin/elasticsearch`
- Logstash - `./path_to_logstash/bin/logstash -f ./path_to_logstash.conf`
- Kibana - `./path_to_kibana/bin/kibana`

```
shubham@machine:~/Documents/Tools$ ./Kibana-7.6.2-linux-x86_64/bin/kibana
log [17:07:59.604] [info][plugins-service] Plugin "case" is disabled.
log [17:09:10.111] [info][plugins-system] Setting up [37] plugins: [usageCollection,metrics,canvas,encryptedSavedObjects,code,timelion,features,apm_oss,taskManager,s
iem,licensing,security,infra,uiActions,data,navigation,status_page,share,newsfeed,kibana_legacy,management,dev_tools,inspector,expressions,visualizations,embeddable,dash
board_embeddable,container,advancedUiActions,home,spaces,cloud,apm,graph,eui_utils,bfetch,translations,reporting]
log [17:09:10.113] [info][plugins][usageCollection] Setting up plugin
log [17:09:10.115] [info][metrics][plugins] Setting up plugin
log [17:09:10.115] [info][canvas][plugins] Setting up plugin
log [17:09:10.136] [info][encryptedSavedObjects][plugins] Setting up plugin
log [17:09:10.138] [warning][config][encryptedSavedObjects][plugins] Generating a random key for xpack.encryptedSavedObjects.encryptionKey. To be able to decrypt enc
rypted saved objects attributes after restart, please set xpack.encryptedSavedObjects.encryptionKey in kibana.yml
log [17:09:10.206] [info][code][plugins] Setting up plugin
log [17:09:10.207] [info][plugins][timelion] Setting up plugin
log [17:09:10.208] [info][features][plugins] Setting up plugin
log [17:09:10.209] [info][apm_oss][plugins] Setting up plugin
log [17:09:10.210] [info][plugins][taskManager] Setting up plugin
log [17:09:10.213] [info][plugins][siem] Setting up plugin
log [17:09:10.214] [info][licensing][plugins] Setting up plugin
log [17:09:10.216] [info][plugins][security] Setting up plugin
log [17:09:10.217] [warning][config][plugins][security] Generating a random key for xpack.security.encryptionKey. To prevent sessions from being invalidated on resta
rt, please set xpack.security.encryptionKey in kibana.yml
log [17:09:10.218] [warning][config][plugins][security] Session cookies will be transmitted over insecure connections. This is not recommended.
log [17:09:10.234] [info][infra][plugins] Setting up plugin
log [17:09:10.235] [info][data][plugins] Setting up plugin
log [17:09:10.239] [info][plugins][share] Setting up plugin
log [17:09:10.241] [info][home][plugins] Setting up plugin
log [17:09:10.245] [info][plugins][spaces] Setting up plugin
log [17:09:10.249] [info][cloud][plugins] Setting up plugin
log [17:09:10.250] [info][apm][plugins] Setting up plugin
log [17:09:10.293] [info][graph][plugins] Setting up plugin
log [17:09:10.299] [info][bfetch][plugins] Setting up plugin
log [17:09:10.300] [info][plugins][translations] Setting up plugin
log [17:09:10.305] [info][savedobjects-service] Waiting until all Elasticsearch nodes are compatible with Kibana before starting saved objects migrations...
log [17:09:10.488] [info][savedobjects-service] Starting saved objects migrations
log [17:09:11.002] [info][plugins-system] Starting [22] plugins: [usageCollection,metrics,canvas,encryptedSavedObjects,code,timelion,features,apm_oss,taskManager,sie
m,licensing,security,infra,data,share,home,spaces,cloud,apm,graph,bfetch,translations]
```

The screenshot shows the Kibana home page with a sidebar on the left containing navigation icons. The main content area is divided into two primary sections: **Observability** and **Security**.

Observability includes:

- APM**: APM automatically collects in-depth performance metrics and errors from inside your applications. [Add APM](#)
- Logs**: Ingest logs from popular data sources and easily visualize in preconfigured dashboards. [Add log data](#)
- Metrics**: Collect metrics from the operating system and services running on your servers. [Add metric data](#)

Security includes:

- SIEM**: Centralize security events for interactive investigation in ready-to-go visualizations. [Add events](#)

Below these sections, there are three options to add data:

- Add sample data**: Load a data set and a Kibana dashboard
- Upload data from log file**: Import a CSV, NDJSON, or log file
- Use Elasticsearch data**: Connect to your Elasticsearch index

At the bottom, there are two main categories:

- Visualize and Explore Data**: Includes [APM](#) and [Canvas](#).
- Manage and Administer the Elastic Stack**: Includes [Console](#) and [Index Patterns](#).

Setting up kibana includes creating a new index pattern under management. Add a new index pattern and press next to choose @timestamp from next window and this will create a new index pattern to view your logs.

D

Management / Index patterns / Create index pattern

Elasticsearch

Index Management

Index Lifecycle Policies

Rollup Jobs

Transforms

Remote Clusters

Snapshot and Restore

License Management

8.0 Upgrade Assistant

Kibana

[Index Patterns](#)

Saved Objects

Spaces

Reporting

Advanced Settings

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

☐ Include system indices

Step 1 of 2: Define index pattern

Index pattern

index-name-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, *, <, >, |.

Your index pattern can match any of your **3 indices**, below.

ilm-history-1-000001

kibana_sample_data_ecommerce

logstash-2020.05.09-000001

Rows per page: 10

> Next step

D

Management / Index patterns / Create index pattern

Elasticsearch

Index Management

Index Lifecycle Policies

Rollup Jobs

Transforms

Remote Clusters

Snapshot and Restore

License Management

8.0 Upgrade Assistant

Kibana

[Index Patterns](#)

Saved Objects

Spaces

Reporting

Advanced Settings

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

☐ Include system indices

Step 2 of 2: Configure settings

You've defined * as your index pattern. Now you can specify some settings before we create it.

Time Filter field name [Refresh](#)

@timestamp

The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.

> Show advanced options

< Back

Create index pattern

Now you can view your logs and virtualize them based on the chosen index pattern. Make sure to select the time range before it, it is usually configured to ~before 15 minutes, set it appropriately. And click on the discover or visualize to see through the logs and monitor them.

