**DAY 6 ASSIGNMENT [17/03/23]**

**Nitish Raj |** [**nitishrajrocks@gmail.com**](mailto:nitishrajrocks@gmail.com)

**Q1. Write the steps “Using Git How to push file in Bitbucket/Gitlab/Github”?**

**Answer:**

Here are the steps to push a file in Git and upload it to a remote repository like Bitbucket, GitLab, or GitHub:

1. **Download and Install** Git from https://git-scm.com/downloads
2. Open your **git bash terminal** or command prompt and navigate to the local repository where your file is located.
3. Initialize Git in the repository by running the command "**git init**".
4. Add the file to the staging area by running the command "**git add** <file\_name>".
5. Commit the file to the repository by running the command "**git commit -m**” commit message'".
6. Link your local repository to the remote repository by running the command "**git remote add origin <remote repository\_url>**". Replace <remote repository url> with the URL of your remote repository like github, GitLab, or bitbucket.
7. Push the changes to the remote repository by running the command "**git push -u origin <branch name>**". Replace <branch name> with the name of the branch you want to push the changes to. By default, the branch name is "master".
8. If this is your first time pushing to the remote repository, Git will prompt you to enter your Bitbucket/GitLab/GitHub username and password. Enter the correct credentials to authenticate yourself.
9. After authentication, the file will be uploaded to the remote repository.

**Q2. Write a program in C to sort an array using Data Structure?**

**Answer:**

//Bubble Sort Program in C

//Author: Nitish Raj

#include <stdio.h>

void bubbleSort**(**int arr**[],** int n**);**

int main**()**

**{**

int arr**[]** **=** **{**64**,** 34**,** 25**,** 12**,** 22**,** 11**,** 90**};**

int n **=** **sizeof(**arr**)/sizeof(**arr**[**0**]);**

printf**(**"Array before sorting:\n"**);**

**for(**int i**=**0**;** i**<**n**;** i**++)**

**{**

printf**(**"%d "**,** arr**[**i**]);**

**}**

bubbleSort**(**arr**,** n**);**

printf**(**"\nArray after sorting:\n"**);**

**for(**int i**=**0**;** i**<**n**;** i**++)**

**{**

printf**(**"%d "**,** arr**[**i**]);**

**}**

**return** 0**;**

**}**

void bubbleSort**(**int arr**[],** int n**)**

**{**

int i**,** j**;**

**for(**i**=**0**;** i**<**n**-**1**;** i**++)**

**{**

**for(**j**=**0**;** j**<**n**-**i**-**1**;** j**++)**

**{**

**if(**arr**[**j**]** **>** arr**[**j**+**1**])**

**{**

// Swap arr[j] and arr[j+1]

int temp **=** arr**[**j**];**

arr**[**j**]** **=** arr**[**j**+**1**];**

arr**[**j**+**1**]** **=** temp**;**

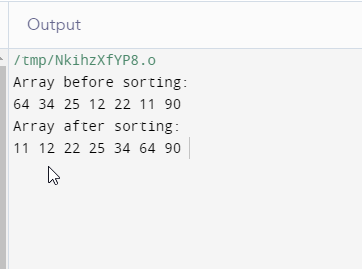
**}**

**}**

**}**

**}**

**Output**

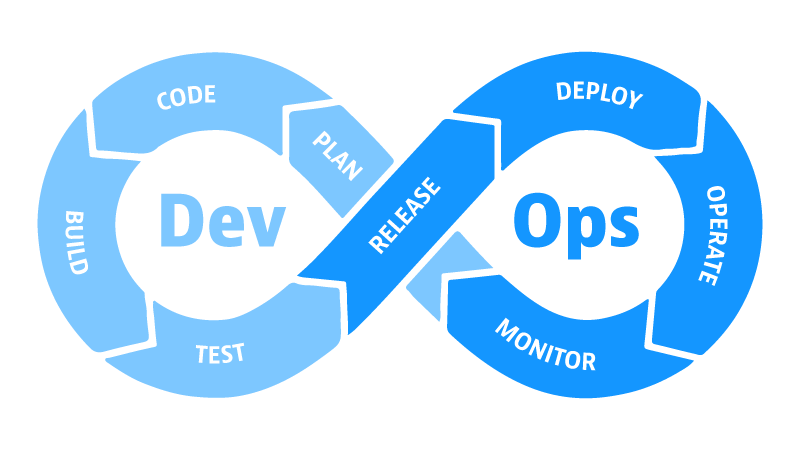
****

**Q3. Explain Waterfall Model, Agile Model, and DevOps in brief?**

**Answer:**

**Waterfall Model:** The waterfall model is a sequential software development process in which progress flows in one direction, downwards like a waterfall, through distinct phases such as requirements gathering, design, implementation, testing, and maintenance. Each phase of the waterfall model must be completed before moving on to the next phase, and changes made during any phase may require changes in all subsequent phases. It is a rigid and linear approach to software development.

**Agile Model:** Agile is a flexible and iterative approach to software development that emphasizes collaboration, customer feedback, and rapid prototyping. In contrast to the waterfall model, agile development is characterized by short development cycles, called sprints, which typically last from one to four weeks. Agile development teams work closely with customers to identify their needs and prioritize features. The most popular frameworks for agile development include Scrum, Kanban, and XP.

**DevOps:** DevOps is a software development approach that emphasizes collaboration, communication, and automation between software developers and IT operations teams. The goal of DevOps is to create a culture of continuous delivery and deployment, where code changes can be quickly and reliably deployed to production environments. DevOps involves the integration of development, testing, and operations teams, and relies on automation tools to streamline the development and deployment process. DevOps practices include continuous integration, continuous delivery, and infrastructure as code.

**Q4. What is Operating System. List 20 Commands of Linux Operating system.**

**Answer:**

An operating system (OS) is a set of software programs that manages computer hardware resources and provides common services for computer programs. The OS acts as an intermediary between the computer hardware and the application software. It provides a user interface, manages file systems, allocates system resources like CPU time and memory, and provides security features like user authentication and authorization.

Here are 20 common Linux commands:

cd - change directory

ls - list files and directories

pwd - print working directory

mkdir - create a new directory

touch - create a new file or update the modification time of an existing file

cp - copy files or directories

mv - move or rename files or directories

rm - remove files or directories

cat - concatenate and display the contents of a file

less - display the contents of a file, one page at a time

head - display the first few lines of a file

tail - display the last few lines of a file

grep - search for a pattern in a file

ps - display information about running processes

top - display a real-time view of system resource usage

ifconfig - configure network interfaces

ping - test network connectivity to another host

ssh - connect to a remote system using Secure Shell (SSH)

tar - create or extract compressed archive files

sudo - execute a command with superuser privileges.

**Q5. What is Shell Script. Write Program for**

* Hello World
* Variable
* Operator
* Control Statement
* Function

**Answer:**

A shell script is a program written in a shell programming language to automate repetitive tasks, perform system administration tasks, or run complex command-line operations. Shell scripts are interpreted by the operating system's shell interpreter, and can be used to execute commands, perform arithmetic and logical operations, read and write files, and more.

Here are the shell script programs for:

**Hello World**

#!/bin/bash

echo "Hello World"

**Variable**

#!/bin/bash

name="John"

age=25

echo "My name is $name and I am $age years old"

**Operator**

#!/bin/bash

num1=10

num2=5

sum=$((num1 + num2))

echo "The sum of $num1 and $num2 is $sum"

**Control Statement**

#!/bin/bash

num=10

if [ $num -gt 5 ]; then

echo "The number is greater than 5"

else

echo "The number is less than or equal to 5"

fi

**Function**

#!/bin/bash

greeting() {

echo "Hello $1"

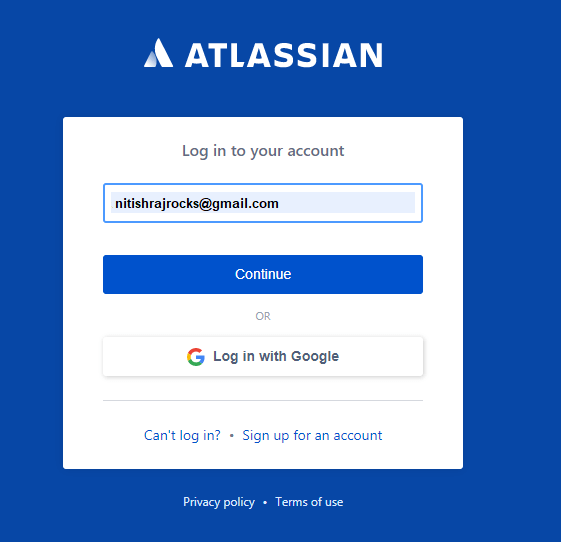
}

greeting "John"

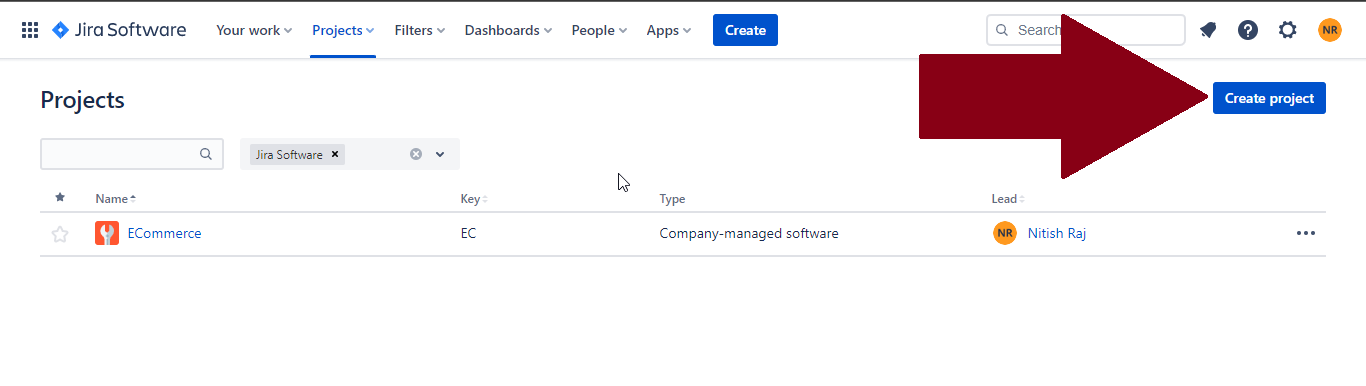
**Q6. Write Steps with Screenshots for creating Projects, Epic story, Task and Sprint in JIRA?**

**Answer:**

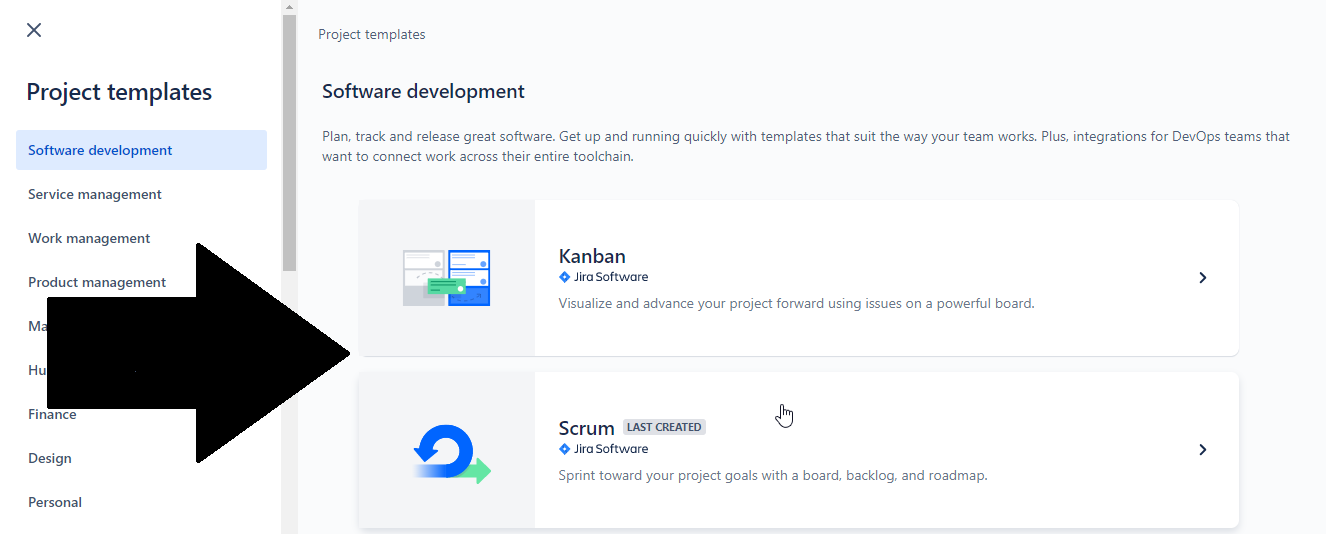
Steps for creating Projects, Epic story, Task and Sprint in JIRA.

**Step 1: Create a Project in JIRA**

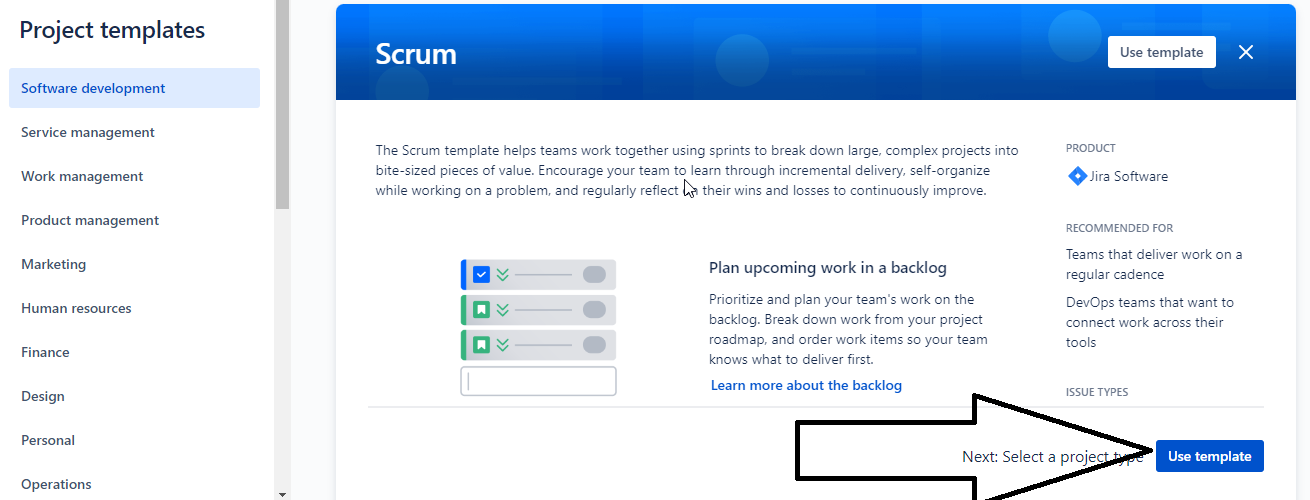
1. Log in to your JIRA account
2. Click on the “**Create Project**” button in the top right-hand corner.



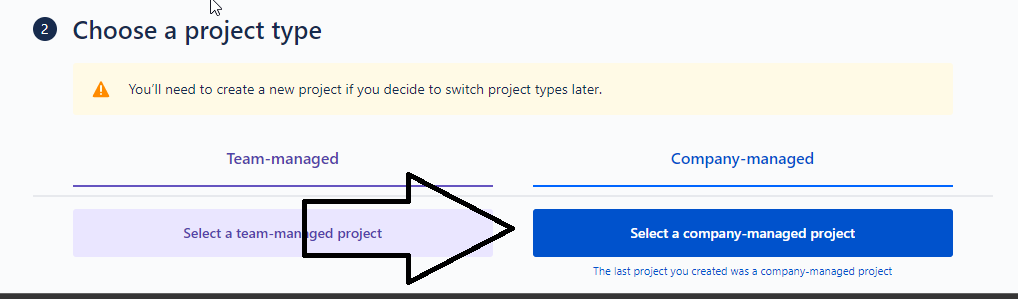
1. Choose the type of project you want to create (e.g., Scrum, Kanban, or Basic Software Development).



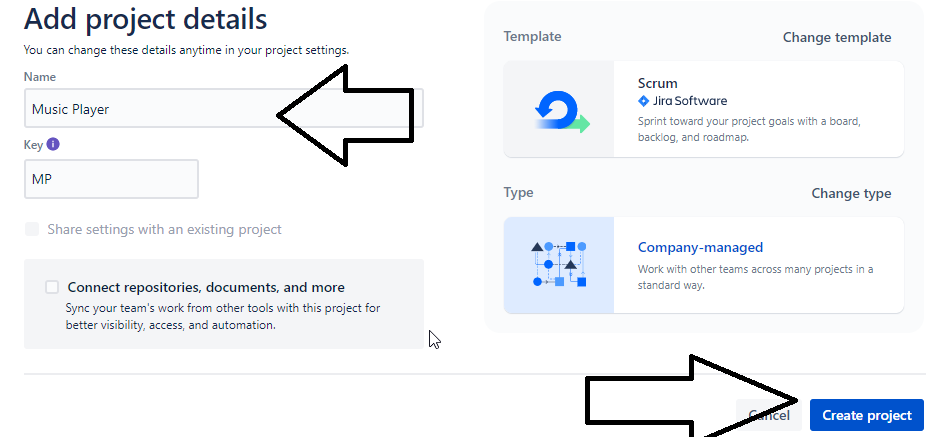
1. Click on **Use Template.**



1. Choose **Project Type.**

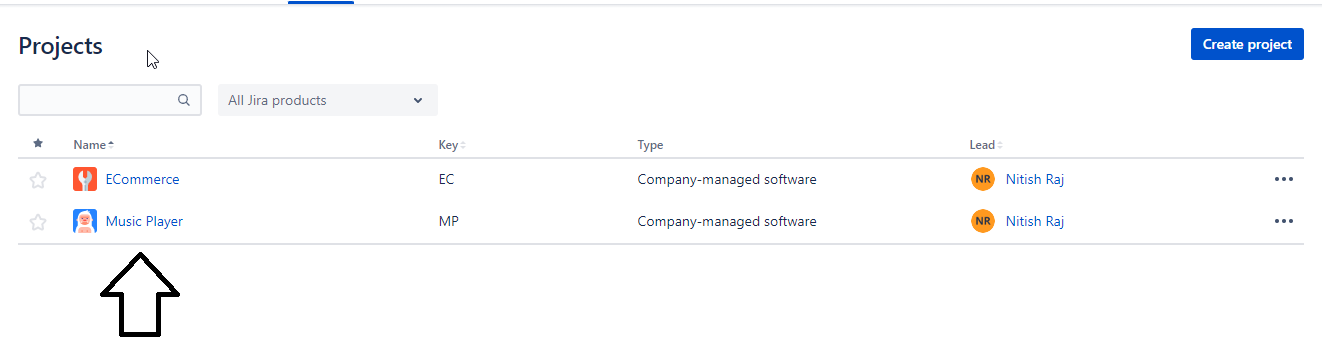


1. Fill in the project details such as **name**, key, and project lead then Click on the “**Create**” button to create the project.

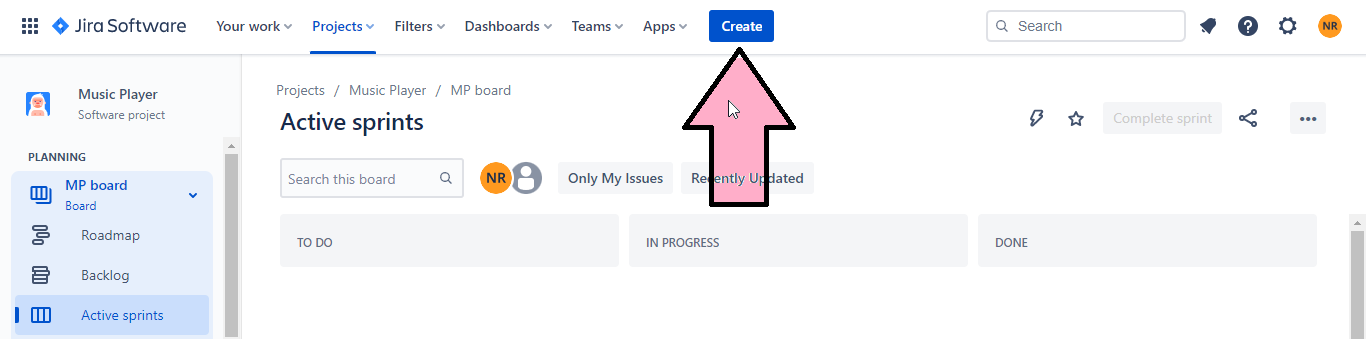


**Step 2: Create an Epic**

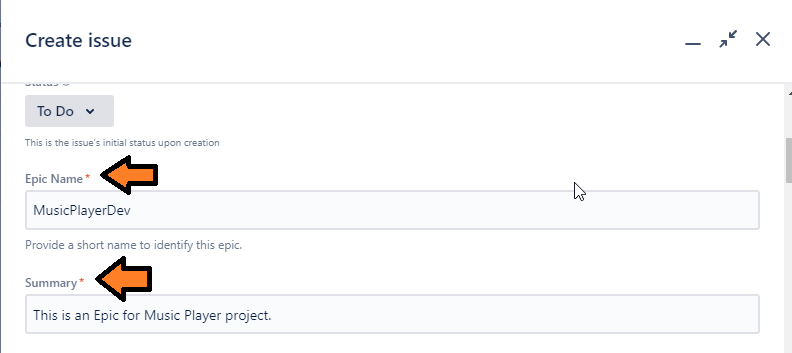
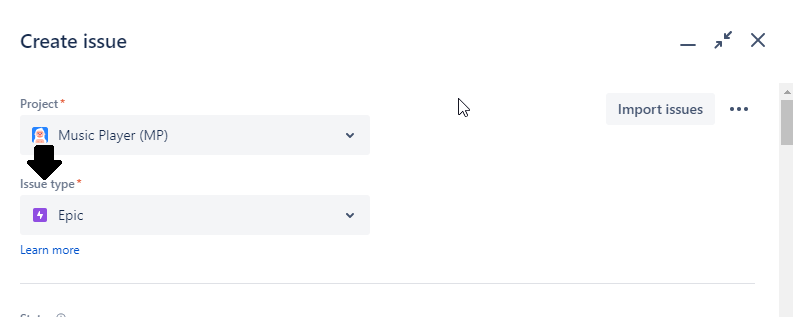
1. From your JIRA dashboard, click on the project you want to add an **epic** to.



1. Click on the “**Create**” tab.



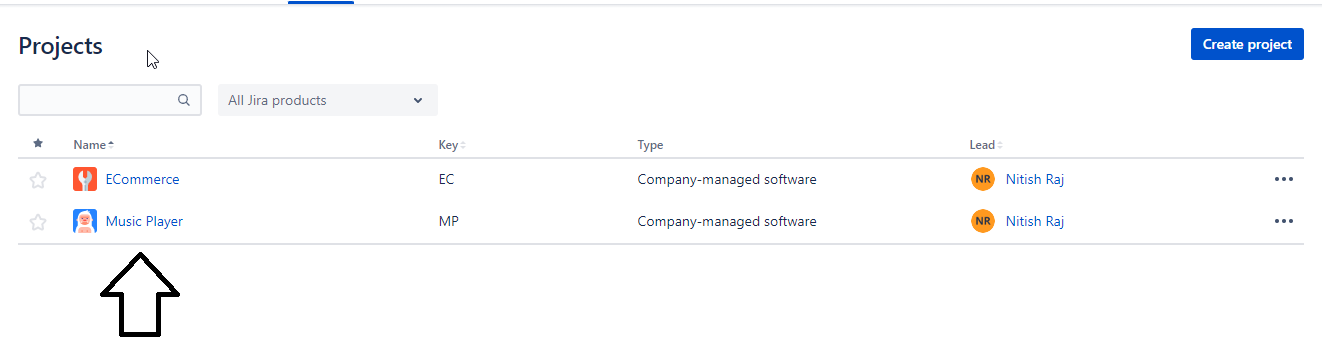
1. Then under Issue type Choose “**EPIC**” give name and description of your EPIC.



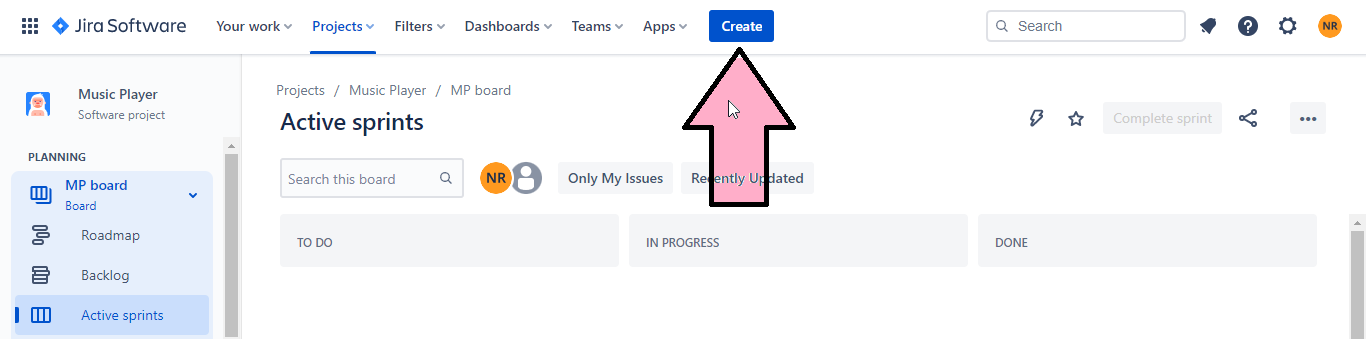
1. Then click on “**Create**”.

**Step 3: Create a Task**

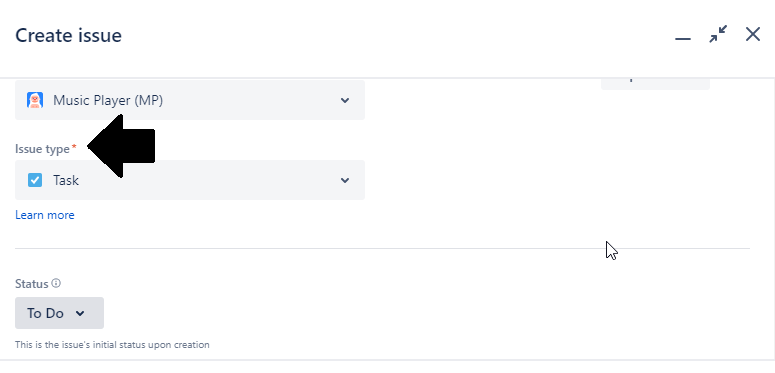
1. From your JIRA dashboard, click on the project you want to add an **TASK** to.

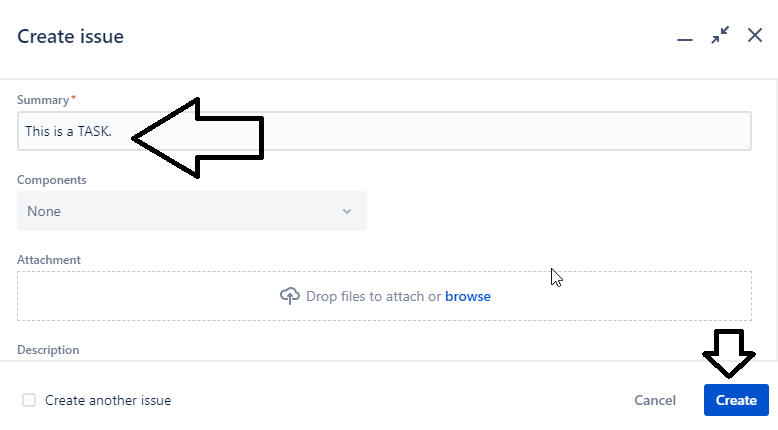


1. Click on the “**Create**” tab.



1. Then under Issue type Choose “**TASK**” give name and description of your TASK.

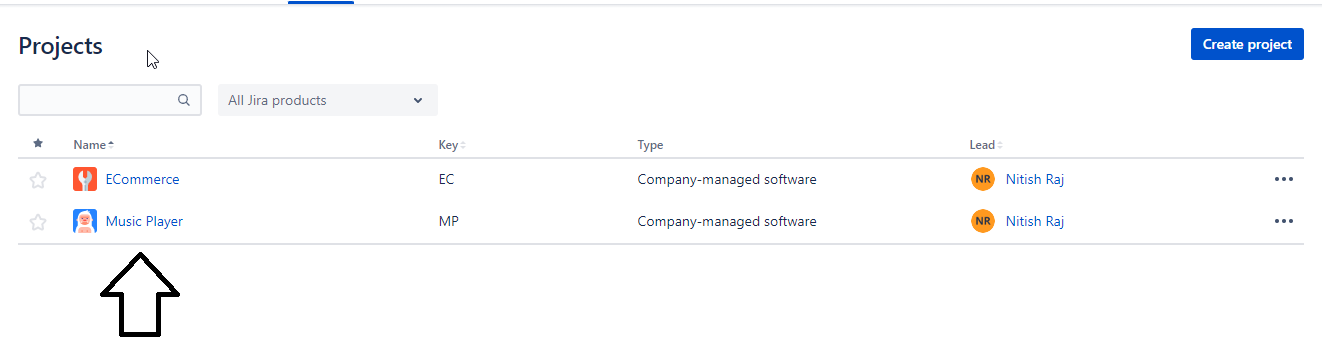




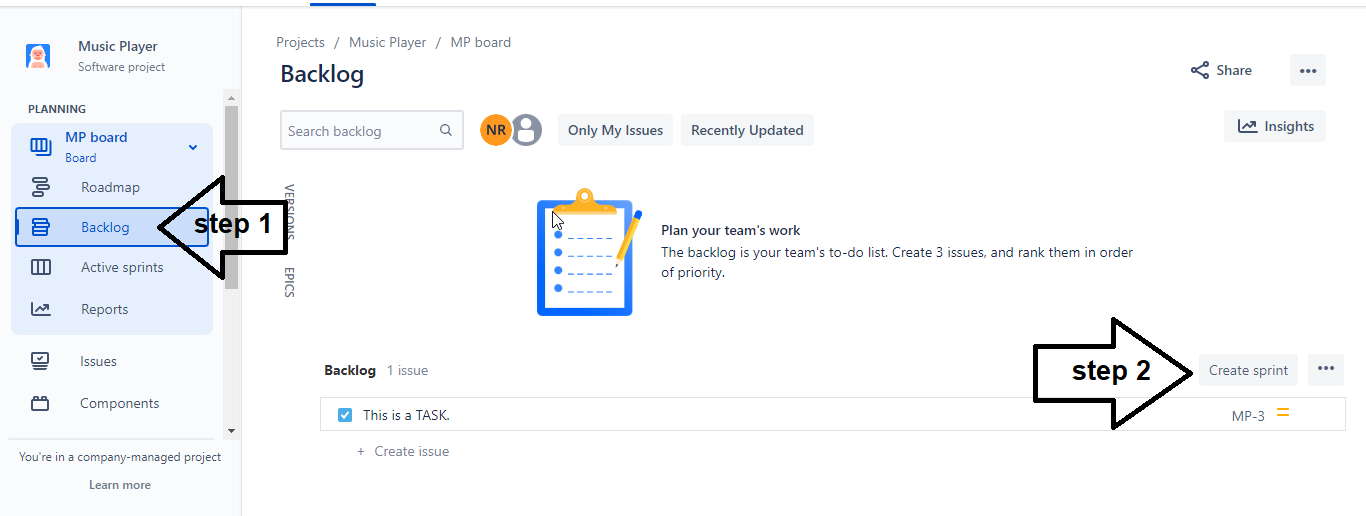
1. Then click on “**Create**”.

**Step 4: Create a Sprint**

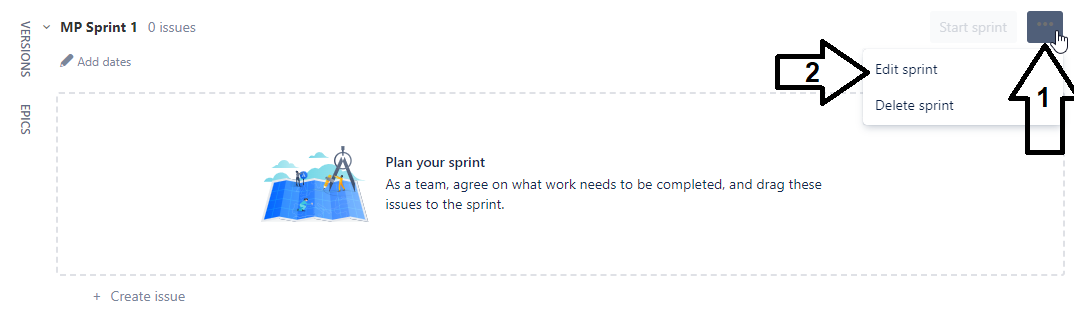
1. From your JIRA dashboard, click on the project you want to add a sprint to.



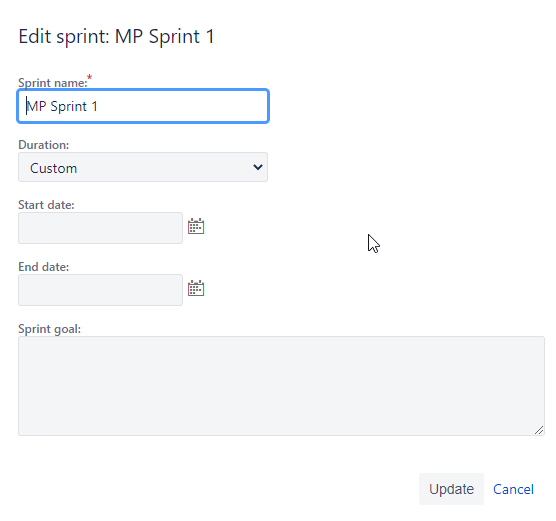
1. Click on the “**Backlog**” tab then “**Create sprint**”.



1. Edit the Sprint by clicking on 3 dots.

****

1. Update the sprint name, duration. You can set start date and End date for the sprint. Or can Add a Sprint goal if desired.



**Q7. Write various stages of Agile Methodology.**

**Answer:**

Agile methodology is a project management approach that emphasizes flexibility, collaboration, and customer satisfaction. The methodology consists of several stages, including:

**Project Initiation:** This is the first stage of Agile methodology where the team comes together to define the project objectives, scope, and stakeholders. This stage includes identifying the project requirements, gathering initial feedback from stakeholders, and establishing the overall project vision.

**Planning:** In the planning stage, the team creates a roadmap for the project. This includes identifying project milestones, breaking the project into smaller tasks, and assigning tasks to team members. The planning stage also involves estimating the time and effort required to complete each task, and creating a project schedule.

**Execution:** The execution stage is where the actual work on the project begins. The team works together to complete the tasks assigned to them and make progress towards the project milestones. The team also holds regular meetings to discuss progress, identify and address any obstacles, and adjust the project plan as necessary.

**Review and Reflection:** In this stage, the team reflects on their work and evaluates their progress. This includes reviewing the completed work and measuring it against the project objectives and goals. The team also identifies areas for improvement and discusses how to make adjustments moving forward.

**Iteration:** Agile methodology involves iterative development, which means that the team works in short, iterative cycles. In this stage, the team repeats the planning, execution, and review stages until the project is completed. Each iteration builds upon the previous one, and the team continually adapts and adjusts the project plan based on feedback and lessons learned.

**Delivery:** The delivery stage is where the completed project is delivered to the customer or end-user. This includes testing the final product, making any necessary adjustments, and providing documentation and training to ensure that the customer can use the product effectively.

**Maintenance:** After the project is delivered, the team provides ongoing support and maintenance to ensure that the product continues to meet the customer's needs. This includes addressing any issues that arise, making updates or improvements to the product, and providing ongoing training and support as needed

**Q8. Difference Between**

* **Waterfall Model vs Agile vs DevOps**
* **Scrum vs Kanban**
* **Git vs Bitbucket**
* **LVCS vs CVCS vs DVCS**
* **DOS vs Windows**

**Answer:**

**Waterfall Model vs Agile vs DevOps**

Waterfall Model, Agile, and DevOps are three different project management methodologies that are used to develop and deliver software products. Each of these methodologies has its own unique characteristics and advantages, which are described below:

**Waterfall Model**: The waterfall model is a traditional project management methodology that follows a linear sequential approach. The methodology consists of several stages, including requirements gathering, design, implementation, testing, deployment, and maintenance. Each stage is completed before moving on to the next one, and there is little or no feedback loop between the stages. This methodology is most suitable for projects that have well-defined requirements and a fixed scope, as it allows for better predictability and control over the project timeline.

**Agile**: Agile methodology is an iterative approach to software development that emphasizes collaboration, flexibility, and customer satisfaction. Agile methodologies consist of short development cycles called sprints, where the team works collaboratively to deliver a working product incrementally. The methodology relies on continuous feedback from stakeholders, and the team adapts the project plan as necessary based on feedback and lessons learned. This methodology is most suitable for projects with changing requirements and a high degree of uncertainty, as it allows for more flexibility and adaptability.

**DevOps**: DevOps is a methodology that emphasizes the collaboration and integration between the development and operations teams. The methodology aims to reduce the time it takes to develop, test, and deploy software products by automating and streamlining the development and deployment processes. The methodology relies on continuous integration and continuous delivery (CI/CD), where the team integrates code changes and tests the code continuously, and deploys the product incrementally. This methodology is most suitable for projects that require a high degree of automation and continuous delivery, as it allows for faster time-to-market and better-quality control.

**Scrum vs Kanban**

Scrum and Kanban are two popular Agile methodologies used in software development. Both methodologies share some similarities, but they also have distinct differences. Here is a comparison of Scrum vs Kanban:

**Scrum:**

Scrum is an Agile methodology that emphasizes teamwork, collaboration, and iterative development. The methodology consists of a set of prescribed roles, events, and artifacts, and relies on self-organizing and cross-functional teams to deliver a working product incrementally.

Roles: Scrum teams consist of a Product Owner, Scrum Master, and Development Team.

Events: Scrum events include Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective.

Artifacts: Scrum artifacts include the Product Backlog, Sprint Backlog, and Increment.

Process: Scrum follows a time-boxed approach, with development cycles called Sprints that are typically 2-4 weeks long. The team works on a defined set of features during the Sprint and delivers a working product increment at the end of each Sprint.

**Kanban:**

Kanban is an Agile methodology that focuses on visualization and continuous flow. The methodology emphasizes limiting work in progress (WIP) and optimizing the flow of work through the system.

Roles: Kanban teams typically have no prescribed roles, with each team member responsible for the work they are assigned.

Events: Kanban has no prescribed events, but some teams may hold regular meetings to discuss progress and identify bottlenecks.

Artifacts: Kanban uses visual boards to represent the workflow and visualize the status of work items.

Process: Kanban is a continuous flow methodology, where work is pulled through the system as capacity becomes available. The team focuses on completing work items before taking on new ones, and the goal is to achieve a steady flow of work through the system.

**Git vs Bitbucket**

Git and Bitbucket are both software development tools used for version control and collaboration on code repositories. Here's a comparison of Git vs Bitbucket:

**Git:**

Git is an open-source distributed version control system used for tracking changes in source code during software development. It allows multiple developers to work on the same codebase simultaneously and enables easy collaboration, code review, and branching. Git is a command-line tool, but there are several GUI-based tools available as well.

Key features of Git:

* Distributed version control system
* Fast and efficient
* Supports branching and merging
* Enables easy collaboration and code review
* Open-source

**Bitbucket:**

Bitbucket is a web-based version control repository hosting service that supports both Git and Mercurial. It provides a secure and centralized location for storing and managing code repositories and enables easy collaboration, code review, and continuous integration and delivery. Bitbucket is owned by Atlassian, and it integrates with other Atlassian tools such as JIRA and Confluence.

Key features of Bitbucket:

* Web-based repository hosting service
* Supports Git and Mercurial
* Enables easy collaboration and code review
* Provides access control and security features
* Integrates with other Atlassian tools

**LVCS vs CVCS vs DVCS**

Version control systems (VCS) are tools used in software development to manage changes to source code and other project artifacts. There are three main types of version control systems: Local Version Control System (LVCS), Centralized Version Control System (CVCS), and Distributed Version Control System (DVCS). Here's a comparison of LVCS vs CVCS vs DVCS:

**Local Version Control System (LVCS):**

An LVCS is a version control system that runs on a single machine, where changes are tracked in a local database. This system is typically used by individual developers or small teams working on a single project. An example of an LVCS is RCS (Revision Control System).

**Advantages of LVCS:**

* Simple to set up and use
* Fast and efficient
* Good for small projects and individual developers
* No network connection required

**Disadvantages of LVCS:**

* Limited collaboration features
* No backup or redundancy
* Difficult to scale to larger projects or teams

**Centralized Version Control System (CVCS):**

A CVCS is a version control system where a single server hosts the source code repository, and developers check out and check in code to this central server. This system is typically used by larger teams working on complex projects. Examples of CVCS include Subversion (SVN) and Perforce.

**Advantages of CVCS:**

* Centralized source code repository
* Good collaboration features
* Version history and tracking
* Access control and security

**Disadvantages of CVCS:**

* Single point of failure
* Network dependency
* Limited offline capabilities
* Slow performance on large repositories

**Distributed Version Control System (DVCS):**

A DVCS is a version control system where each developer has a local copy of the source code repository, and changes are synchronized between repositories. This system is typically used by distributed teams working on complex projects. Examples of DVCS include Git and Mercurial.

**Advantages of DVCS:**

* No single point of failure
* Good collaboration features
* Fast and efficient
* Distributed workflow
* Offline capabilities

**Disadvantages of DVCS:**

* Steep learning curve
* Difficult to manage large repositories
* No access control by default

**DOS vs Windows**

***DOS (Disk Operating System) and Windows are two different operating systems for personal computers.***

DOS is a command-line based operating system developed by Microsoft in the 1980s. It was used extensively in the 1980s and 1990s, but has since been largely replaced by graphical user interface (GUI) operating systems like Windows. DOS requires users to type in commands to navigate and execute tasks, which can be less user-friendly than GUI interfaces.

Windows, on the other hand, is a GUI-based operating system also developed by Microsoft. It was first released in 1985 and has since become the dominant operating system for personal computers. Windows allows users to navigate and execute tasks through a graphical interface, using icons, windows, and menus.

While DOS is no longer commonly used, it is still sometimes used for specific applications that require it. Windows, on the other hand, has evolved over time and is now available in many different versions, with varying features and capabilities. The most recent version of Windows is Windows 11, released in 2021.

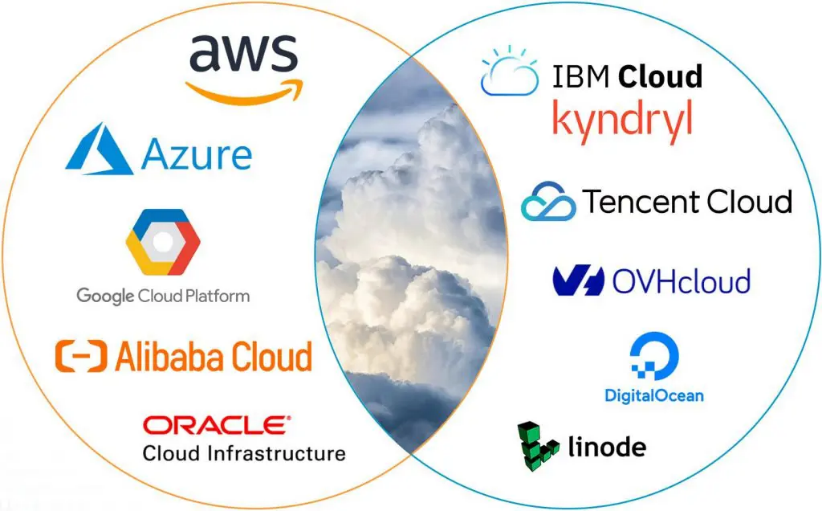
**Q9. Write names of 15 DevOps tools?**

**Answer:**

|  |  |  |
| --- | --- | --- |
| **1.Kubernetes** | **2.Katalon TestOps** | **3. Docker** |
| **4. Katalon Studio** | **5.Ranorex Studio** | **6. Jenkins** |
| **7.Azure DevOps** | **8.Ansible** | **9.Chef** |
| **10. Git** | **11. Terraform** | **12. Gradle** |
| **13.Jira** | **14 Trello** | **15. Raygun** |

**Q10. Write Names of 10 Cloud Providers.**

**Answer:**

1. ****Amazon Web Services (AWS)
2. Microsoft Azure
3. Google Cloud Platform (GCP)
4. Alibaba Cloud
5. Oracle Cloud
6. IBM Cloud (Kyndryl)
7. Tencent Cloud
8. OVHcloud
9. DigitalOcean
10. Linode (Akamai)