**Docker, Docker Compose**

**Virtual Machine (VM)**

**Jira, Bitbucket**

**Docker, Docker Compose**

**What is Docker?**

The word "Docker" refers to several things, including an open source community project; tools from the open source project; Docker Inc., the company that primarily supports that project; and the tools that company formally supports. The IT software "Docker” is containerization technology that enables the creation and use of [Linux containers](https://www.redhat.com/en/topics/containers-v1-old).

With Docker, you can treat containers like extremely lightweight, modular virtual machines. And you get flexibility with those containers & mdash; you can create, deploy, copy, and move them from environment to environment, which helps [optimize your apps for the cloud](https://www.redhat.com/en/topics/cloud-native-apps).

A screenshot of a computer

Description automatically generated

**How to Install?**

Download and install via <https://www.docker.com/products/docker-desktop/>

**What is Docker Compose?**

Docker Compose is a tool that helps you define and share multi-container applications. With Compose, you can create a YAML file to define the services and with a single command, you can spin everything up or tear it all down.

**How to install?**

Visit install instructions via: <https://docs.docker.com/compose/install/>

**How does Docker work?**

The Docker technology uses the [Linux kernel](https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel) and features of the kernel, like [Cgroups](https://www.redhat.com/en/blog/world-domination-cgroups-rhel-8-welcome-cgroups-v2) and [namespaces](https://lwn.net/Articles/528078/), to segregate processes so they can run independently. This independence is the intention of containers—the ability to run multiple processes and apps separately from one another to make better use of your infrastructure while [retaining the security](https://www.redhat.com/en/topics/security) you would have with separate systems.

Container tools, including Docker, provide an image-based deployment model. This makes it easy to share an application, or set of services, with all of their dependencies across multiple environments. Docker also automates deploying the application (or combined sets of processes that make up an app) inside this container environment.

A screenshot of a computer

Description automatically generated

These tools built on top of [Linux containers](https://www.redhat.com/en/topics/containers/whats-a-linux-container)—what makes Docker user-friendly and unique—gives users unprecedented access to apps, the ability to rapidly deploy, and control over versions and version distribution.

**Pros and cons of Docker**

**Pros:**

* **Cross-platform consistency**: Compatibility across a range of systems and environments makes developers’ jobs easier.
* **Serverless storage:**Docker containers are cloud-based and don’t require tons of active memory to run reliably.
* **High-speed deployment:**Eliminating redundant installations and configurations makes deployment fast and easy.
* **Flexibility and scalability:**Developers can use any programming language and scale container resources up and down as needed.

**Cons:**

* **Outdated documentation:**Docker’s extensive documentation doesn’t always keep pace with platform updates.
* **Steep learning curve:**Developers transitioning from other infrastructure might find Docker easy to begin but hard to master.
* **Security issues:**The lack of segmentation means that multiple containers can be vulnerable to host system attacks.
* **Limited orchestration:**It can be difficult to effectively manage multiple containers at once without orchestration capabilities.

Read more via:

<https://www.redhat.com/en/topics/containers/what-is-docker#advantages-of-docker>

<https://duplocloud.com/blog/docker-advantages-and-disadvantages/>

**Virtual Machine (VM)**

**What is VM?**

Virtual machines play a crucial role in cloud computing, emulating physical computers by running operating systems in isolated instances. Multiple VMs are commonly hosted on a single server, with a hypervisor acting as a lightweight software layer positioned between the physical host and the VMs. This hypervisor efficiently manages access to resources, enabling virtual machines to function as distinct servers while offering enhanced flexibility and agility.

Virtual machines contain their own operating system, allowing them to perform multiple resource-intensive functions at once. The increased resources available to VMs allow them to abstract, split, duplicate, and emulate entire servers, operating systems, desktops, databases, and[networks](https://www.redhat.com/en/topics/virtualization/what-is-nfv).

**VMs vs Containers**

**How do they work?**

**A comparison of a table

Description automatically generated with medium confidence**

Learn more via: <https://www.redhat.com/en/topics/containers/containers-vs-vms#how-do-they-work>

**How are they related?**

**A screenshot of a white box

Description automatically generated**

Learn more via: <https://cloud.google.com/discover/containers-vs-vms#section-5>

**Which one should use?**

Compared to VMs, containers are best used to:

* Build cloud-native apps
* Package microservices
* Incorporate applications into DevOps or CI/CD practices
* Move scalable IT projects across a diverse IT footprint

Compared to containers, VMs are best used to:

* House traditional, legacy, and monolithic workloads
* Isolate risky development cycles
* Provision [infrastructural](https://www.redhat.com/en/topics/cloud-computing/what-is-it-infrastructure) resources (such as networks, servers, and [data](https://www.redhat.com/en/topics/big-data))
* Run a different OS inside another OS (such as running Unix on [Linux](https://www.redhat.com/en/topics/linux))

Learn more via: <https://www.redhat.com/en/topics/containers/containers-vs-vms#which-one-should-i-use>

**Jira, Bitbucket**

**What is Jira?**

Jira is a project management tool developed by Atlassian. It's primarily used for issue tracking, bug tracking, and project management functions. Jira allows teams to organize tasks, assign work, track progress, and manage projects efficiently. It's widely used in software development but can be adapted to various project management needs across different industries.

A screen shot of a project

Description automatically generated

**What is Bitbucket?**

Bitbucket, on the other hand, is also a product of Atlassian. It's a web-based version control repository hosting service. Essentially, Bitbucket helps teams collaborate on code development by providing a platform for version control using Git or Mercurial.

A screenshot of a computer

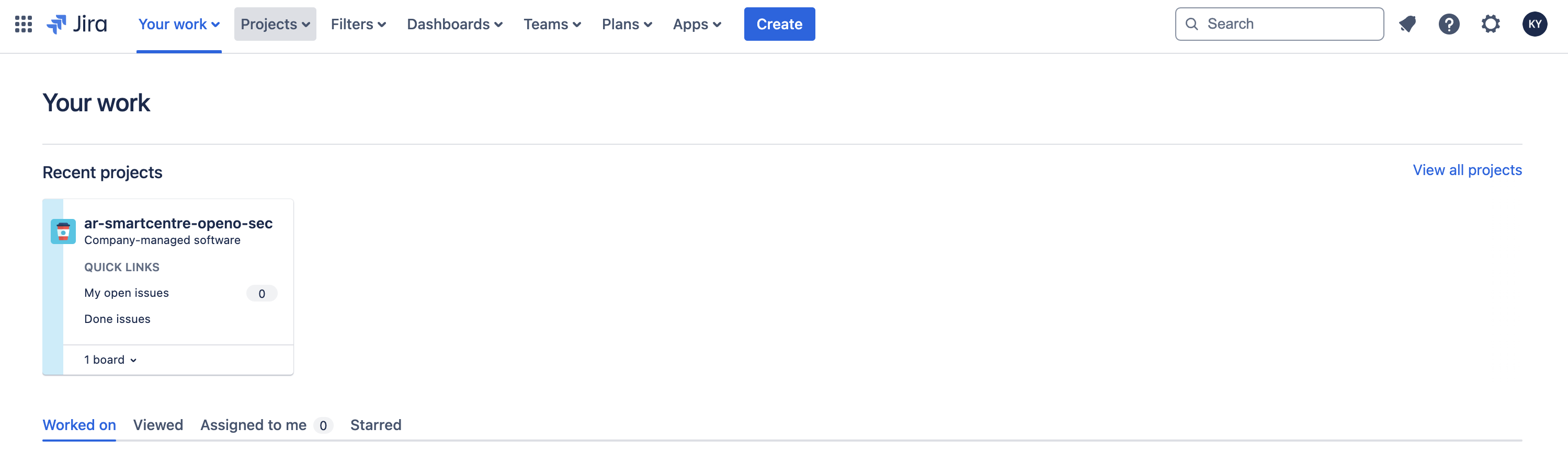
Description automatically generated

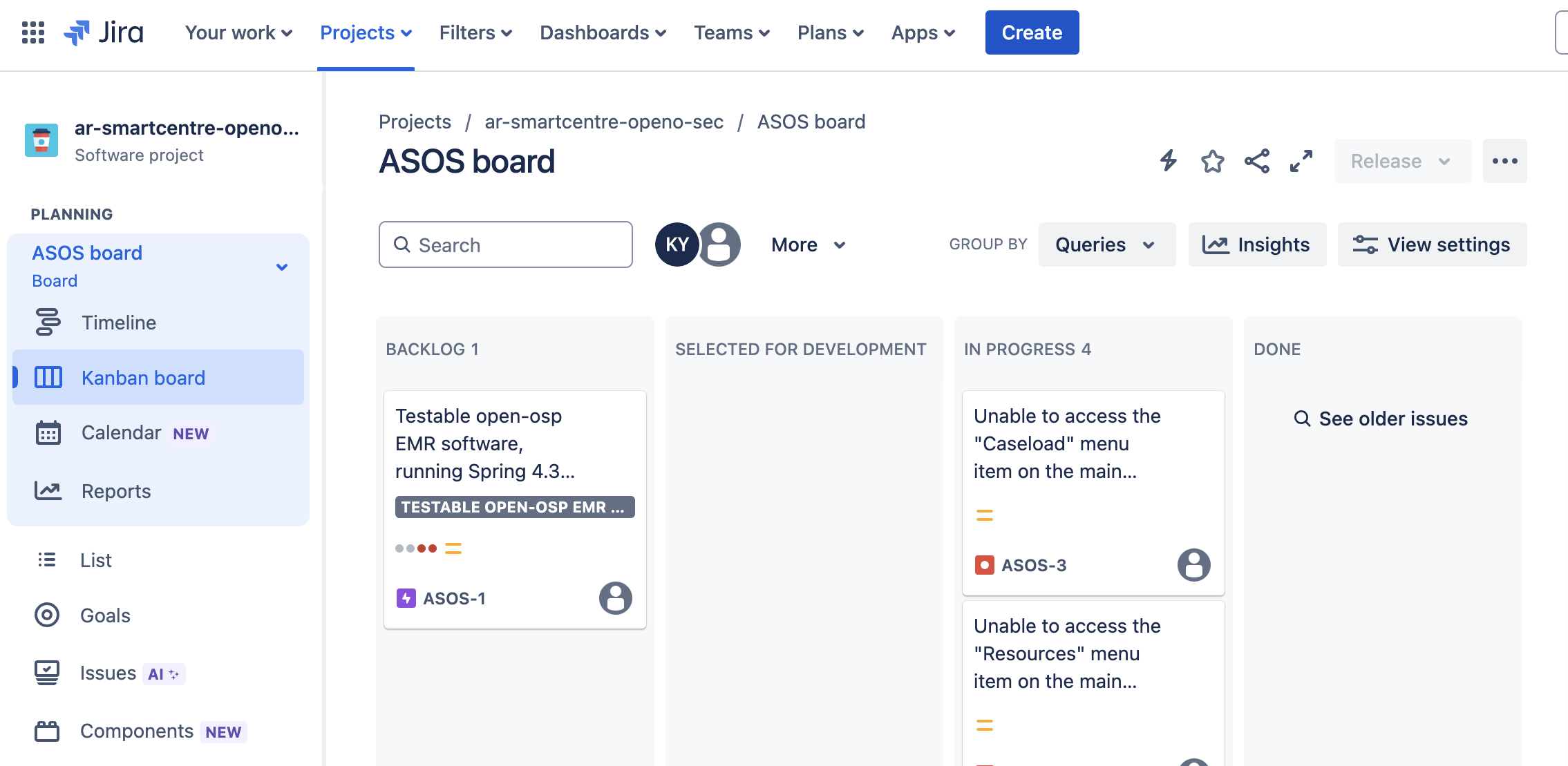
**Relationships between Jira and Bitbucket**

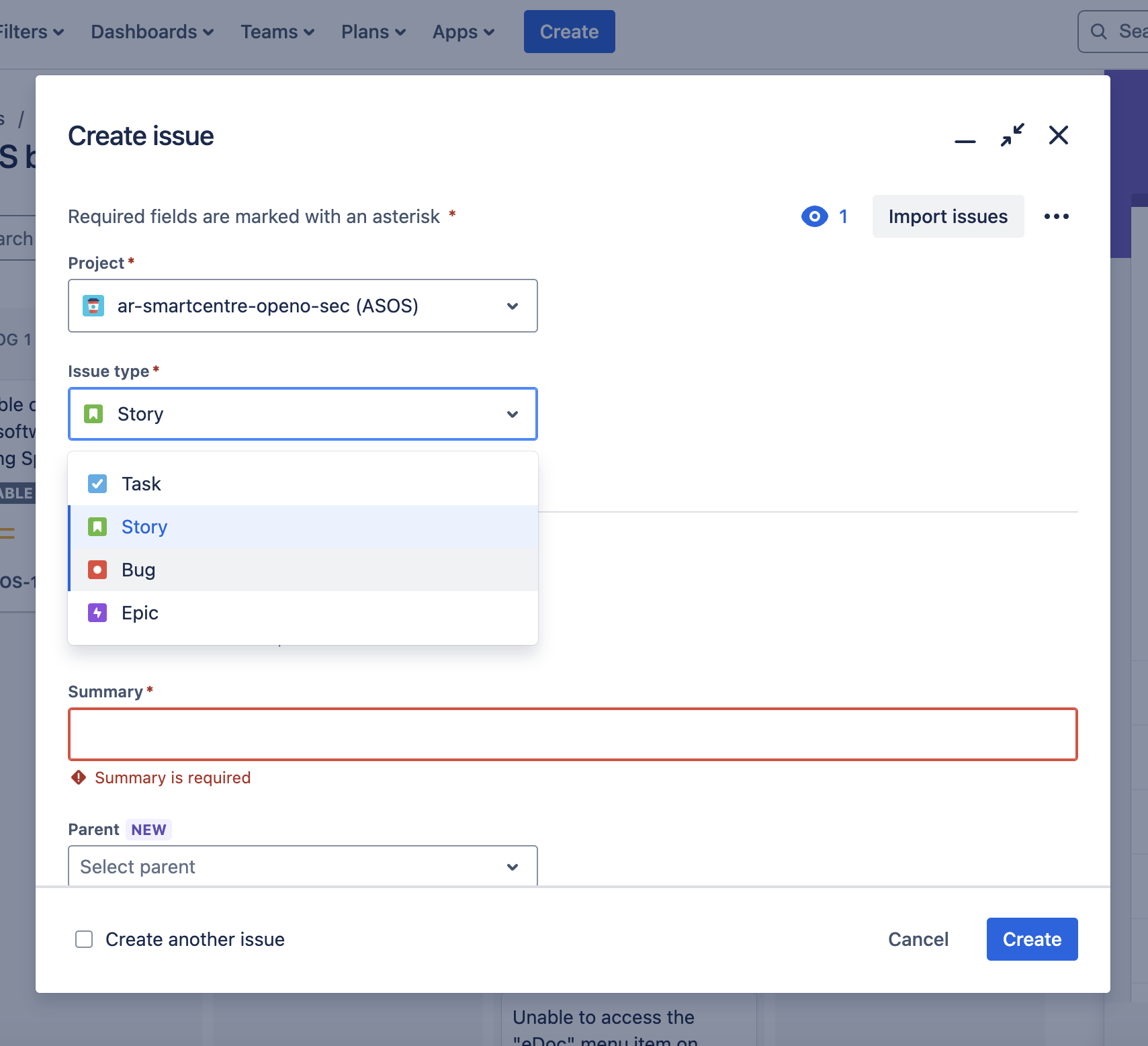
The relationship between Jira and Bitbucket lies in integration. Many software development teams use both tools together. They integrate Bitbucket with Jira to streamline their development process. This integration allows for better traceability between code changes and project tasks. For instance, developers can link code commits, branches, and pull requests in Bitbucket to specific issues or tasks in Jira. This linkage enables teams to have a comprehensive view of their project progress, from planning and tracking in Jira to code development and collaboration in Bitbucket.

**Create Issue Ticket in Jira**

Log in to your Jira account and go to the project you are working on.

Click the 'Create' button; then you will see the windows to create the issue.Log in to your Jira account and go to the project you are working on.

You can choose the issue type and write the summary of the issue, then click 'Create' to create your issue ticket.

Go to the Issue page; you can find all the project issues.

