ConverSight

Task and Flow Management

Empowering Collaboration and Efficiency for Streamlined Next-Generation Analytics.

Whitepaper

Table of Contents

I. Introduction

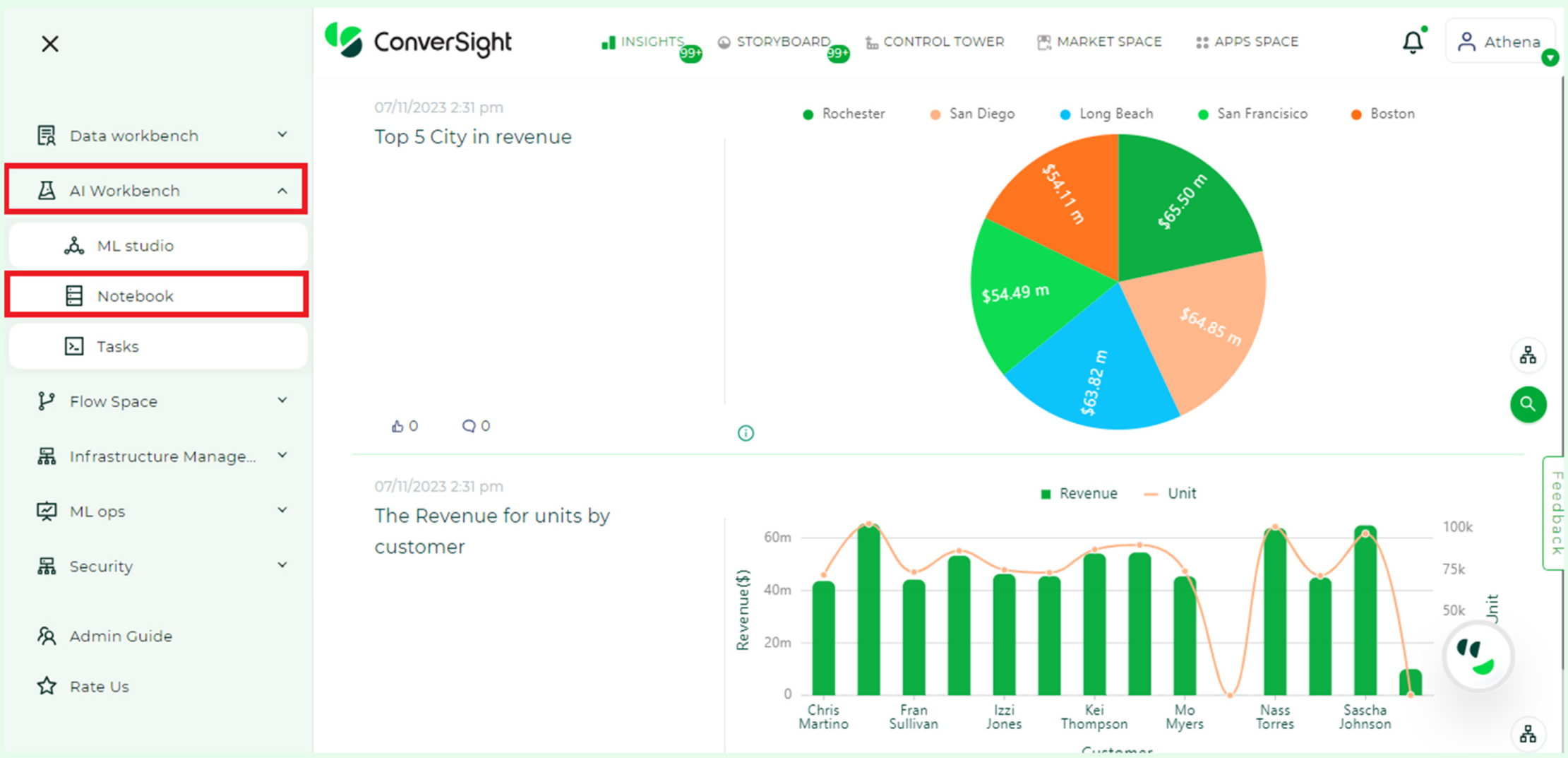
ConverSight's AI Workbench empowers organizations in navigating data-driven decision-making with its integrated augmented analytics model. At its core, the AI Workbench comprises two fundamental elements: tasks and flows. Tasks represent discrete units of work within the AI workflow, ranging from data preprocessing to model deployment. Designed for modularity and efficiency, tasks prioritize reusability to enhance workflow effectiveness. Flows, on the other hand, dictate the sequential arrangement of tasks, ensuring seamless data flow and task execution. Together, these components form a cohesive ecosystem poised to revolutionize AI-powered analytics. This whitepaper delves into task and flow creation and management within ConverSight's AI Workbench. Readers will learn to design specialized tasks for data processing and model training, seamlessly integrating them into coherent flows. Advanced features such as flow scheduling and notifications are also explored, offering insights into optimizing AI processes for enhanced productivity and efficiency.

2. Role of Notebook

In ConverSight, Notebooks hold a vital role as interactive and collaborative spaces for data exploration, analysis and model development. These dynamic environments empower users to write, execute and document code seamlessly, serving as a hub where data-related tasks converge. They facilitate interactive engagement with datasets, enabling a deeper understanding of data intricacies.

2.1 Accessing and Using Notebook

Initiate the creation of tasks and flows within ConverSight by first accessing the Notebook feature. Head to the Configuration sidebar and select **'Notebook'** from the **'AI Workbench'** menu.

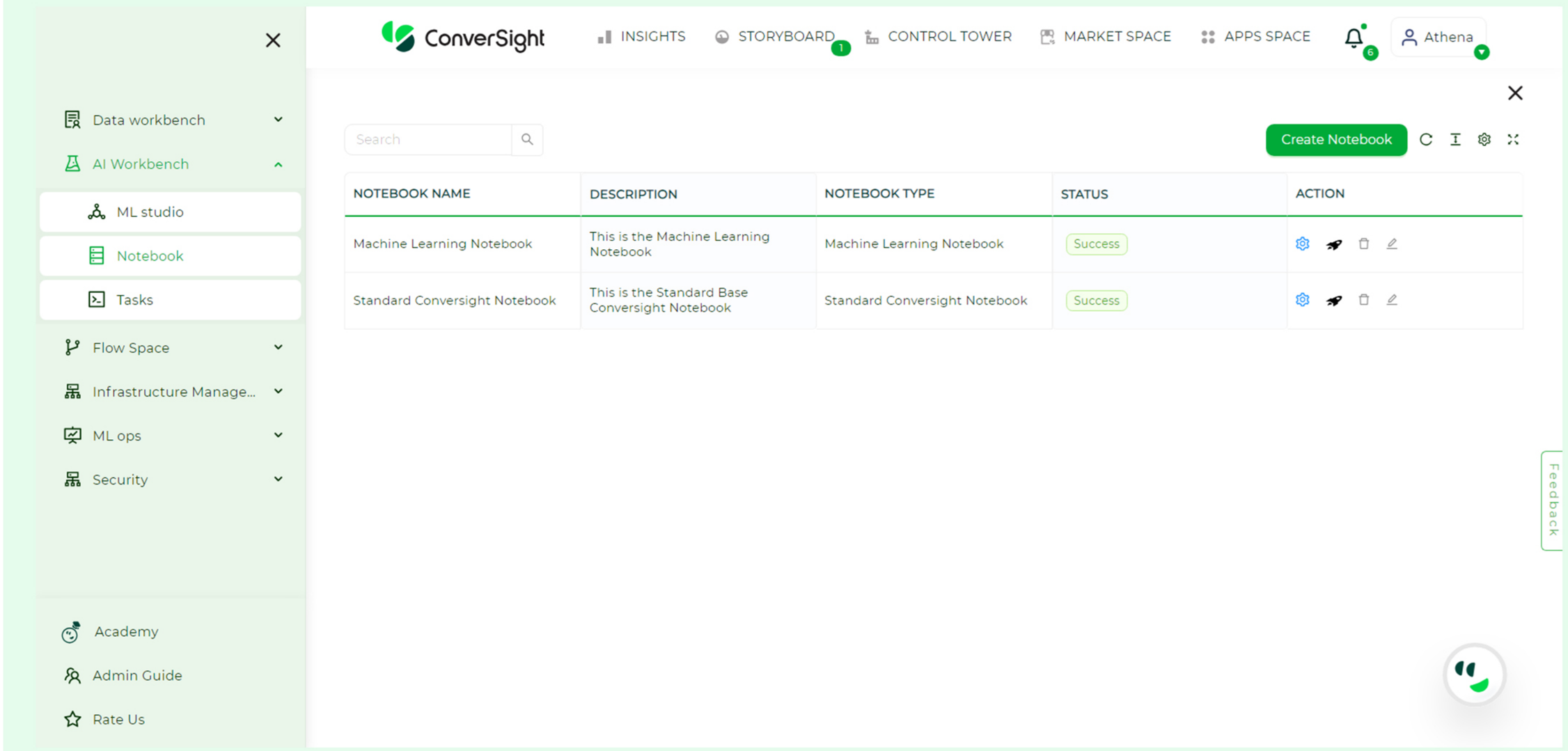


ConverSight provides two default Base Notebooks: The Standard ConverSight Notebook and the Machine Learning Notebook.

Standard ConverSight Notebook: The Standard ConverSight Notebook serves as a foundational tool, encompassing essential packages for executing tasks, flows, CS App and conducting fundamental analytical operations.

Machine Learning Notebook: The Machine Learning Notebook is specifically crafted to empower users with an extensive set of packages catering to the intricate requirements of machine learning and advanced analytics.

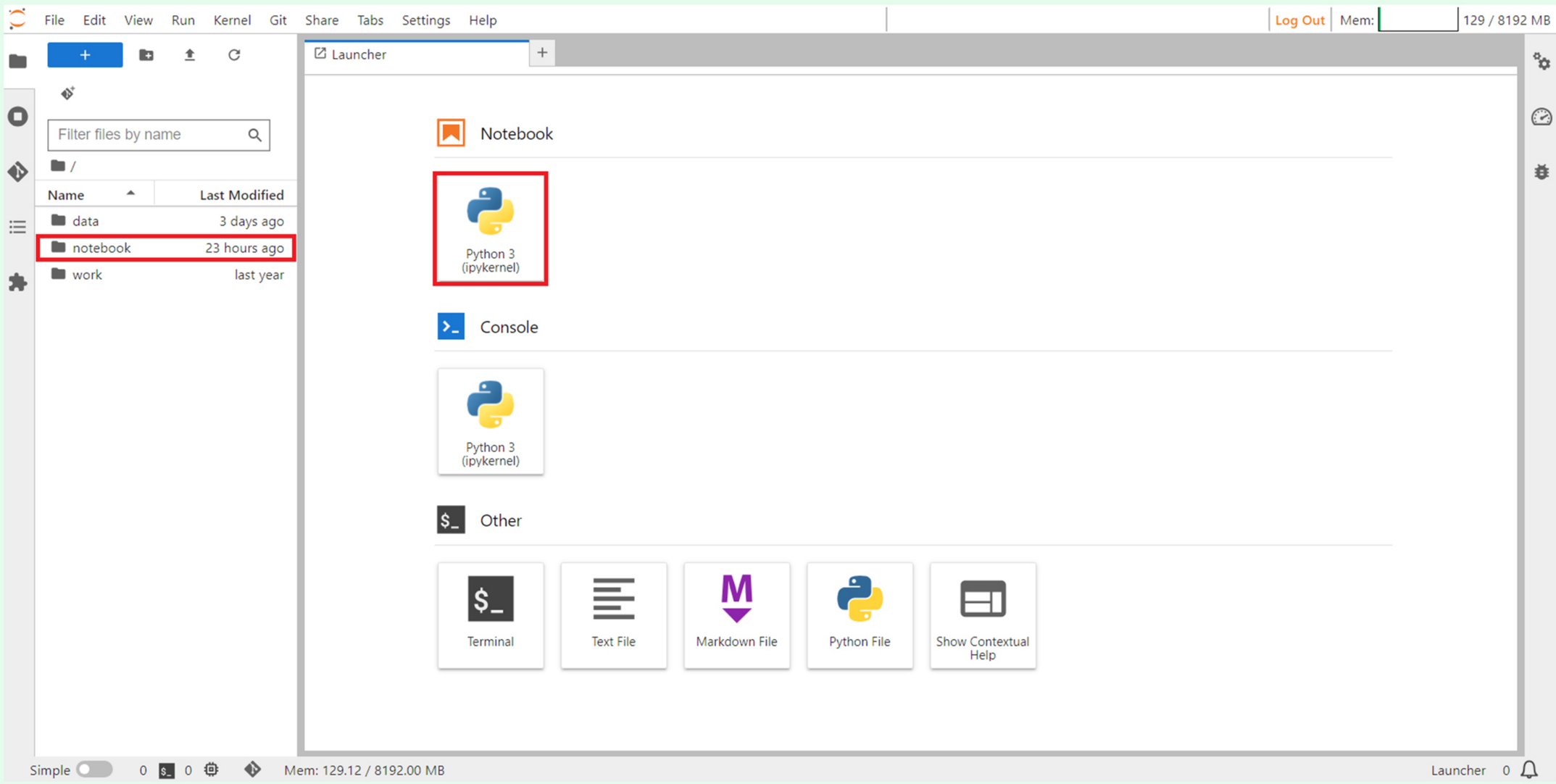
Users can create custom Notebook by integrating a custom package or a ConverSight verified package into the Base Standard ConverSight Notebook, resulting in the creation of a custom Standard ConverSight Notebook. Likewise, incorporating custom packages into the Machine Learning Notebook facilitates the development of a customized Machine Learning Notebook, catering to specific requirements.



To initiate the Notebook, you can simply click on the **'Rocket'** icon located in the action field of the associated Notebook.

2.2 Creating a Workflow in Notebook

In order to initiate a new workflow, users must navigate to the designated 'notebook' folder and create a workflow by selecting a desired kernel. The Notebook operates within a Python environment, complete with pre-installed packages to facilitate seamless execution.



Note

It is important to note that any packages installed during a session will only be stored for a maximum of four hours. Once the server is stopped, they will be deleted.

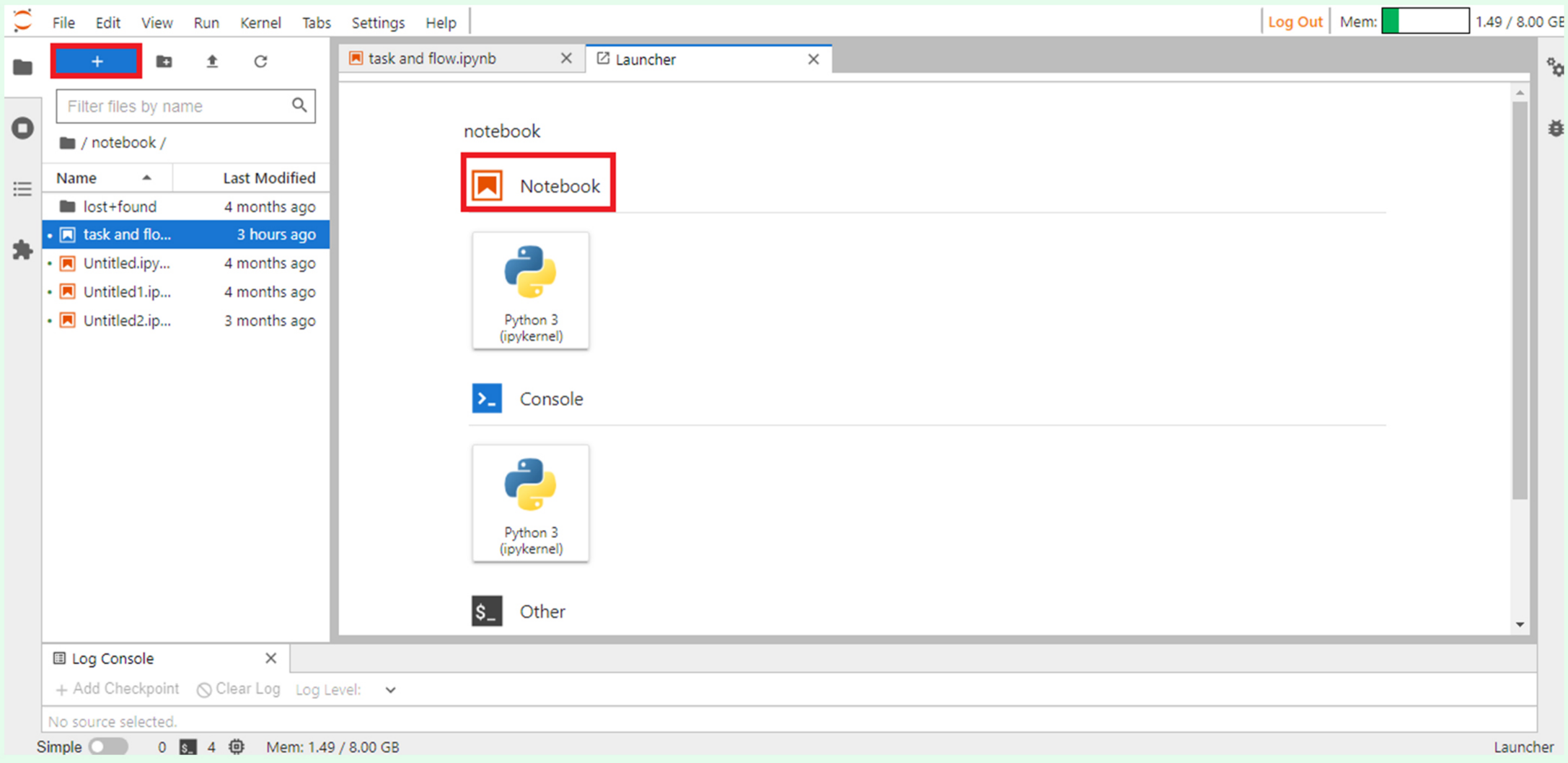
3. Tasks

The task is comparable to a standalone function that represents a distinct and specific piece of work. The beauty of tasks lies in their ability to encapsulate portions of your workflow's logic within clearly defined units that can be easily reused in different parts of your flows. Tasks can accept inputs, executing operations and generating outputs. Perfect tasks are quite versatile, as they offer functionality akin to what you would expect from a Python function. To designate a function as a task, the '@task' decorator is employed, which essentially marks that function as a task to be utilized within the workflow. These tasks can be thought of as building blocks that can be seamlessly combined to create more complex workflows. The collaborative nature of tasks enables seamless sharing and reuse across multiple flows within ConverSight AI Workbench without starting from scratch each time, empowering users to construct adaptable and scalable workflows.

3.1 Creating Tasks

Sequential steps for Task creation are:

- Click ‘Notebook’ to initiate the creation of a fresh notebook.



- Import the necessary packages and modules required for your task's operation.

- To utilize tasks and flows in Jupyter Notebook within the ConverSight platform, users must import the **‘conversight’** library. It provides a wide range of pre-built tasks, simplifying complex tasks without coding from scratch.

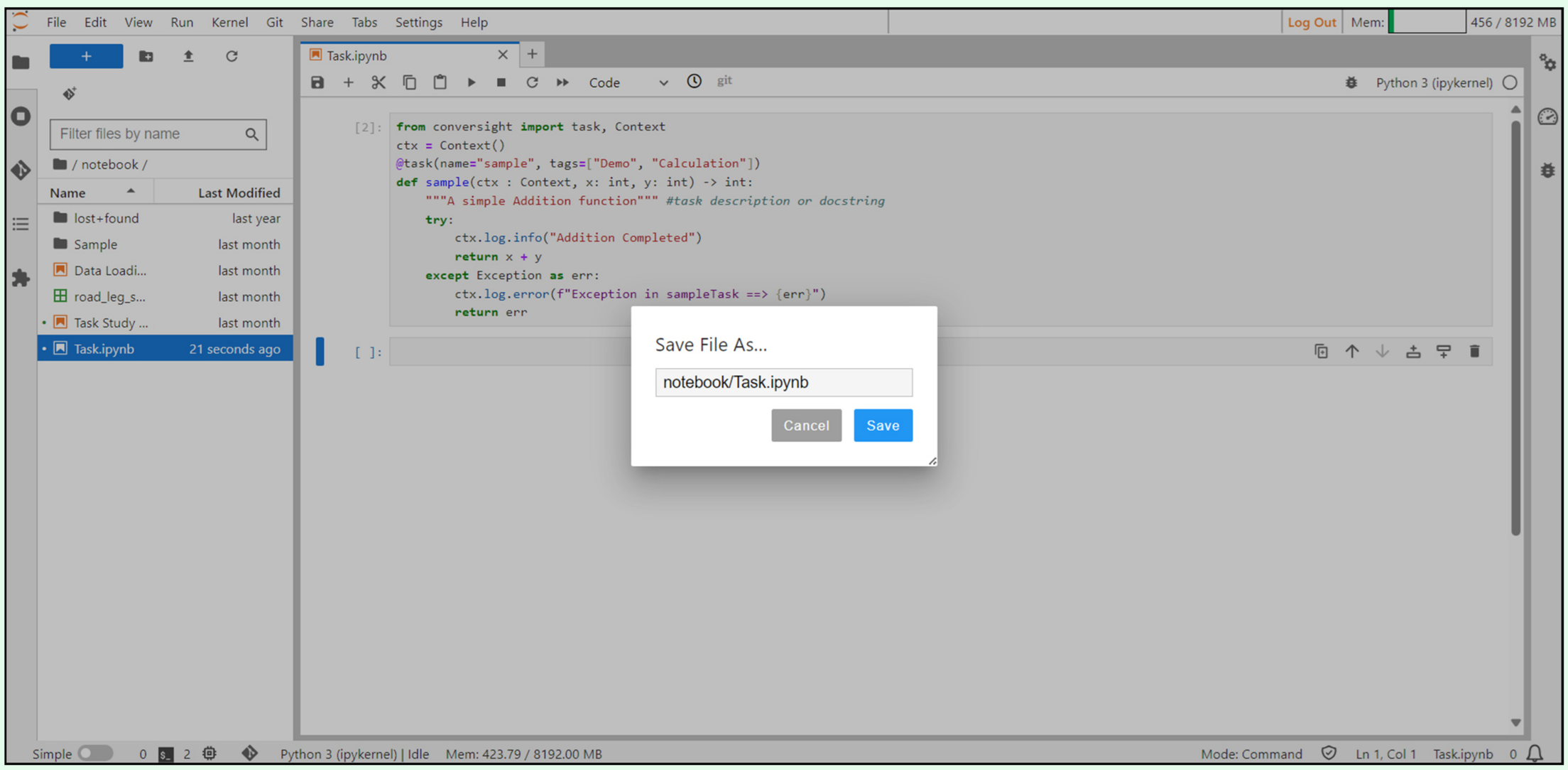
- Importing the Context function from the conversight library is also necessary. The initial step involves using the context() function as a default parameter.

- It is important to note that in ConverSight the **‘print’** statement is restricted for tasks and flows.

Sample Code

```
from conversight import task, Context
ctx = Context()
@task(name="sample", tags=["Demo", "Calculation"])
def sample(ctx : Context, x: int, y: int) -> int:
    """A simple Addition function""" #task description or docstring
    try:
        ctx.log.info("Addition Completed")
        return x + y
    except Exception as err:
        ctx.log.error(f"Exception in sampleTask ==> {err}")
```

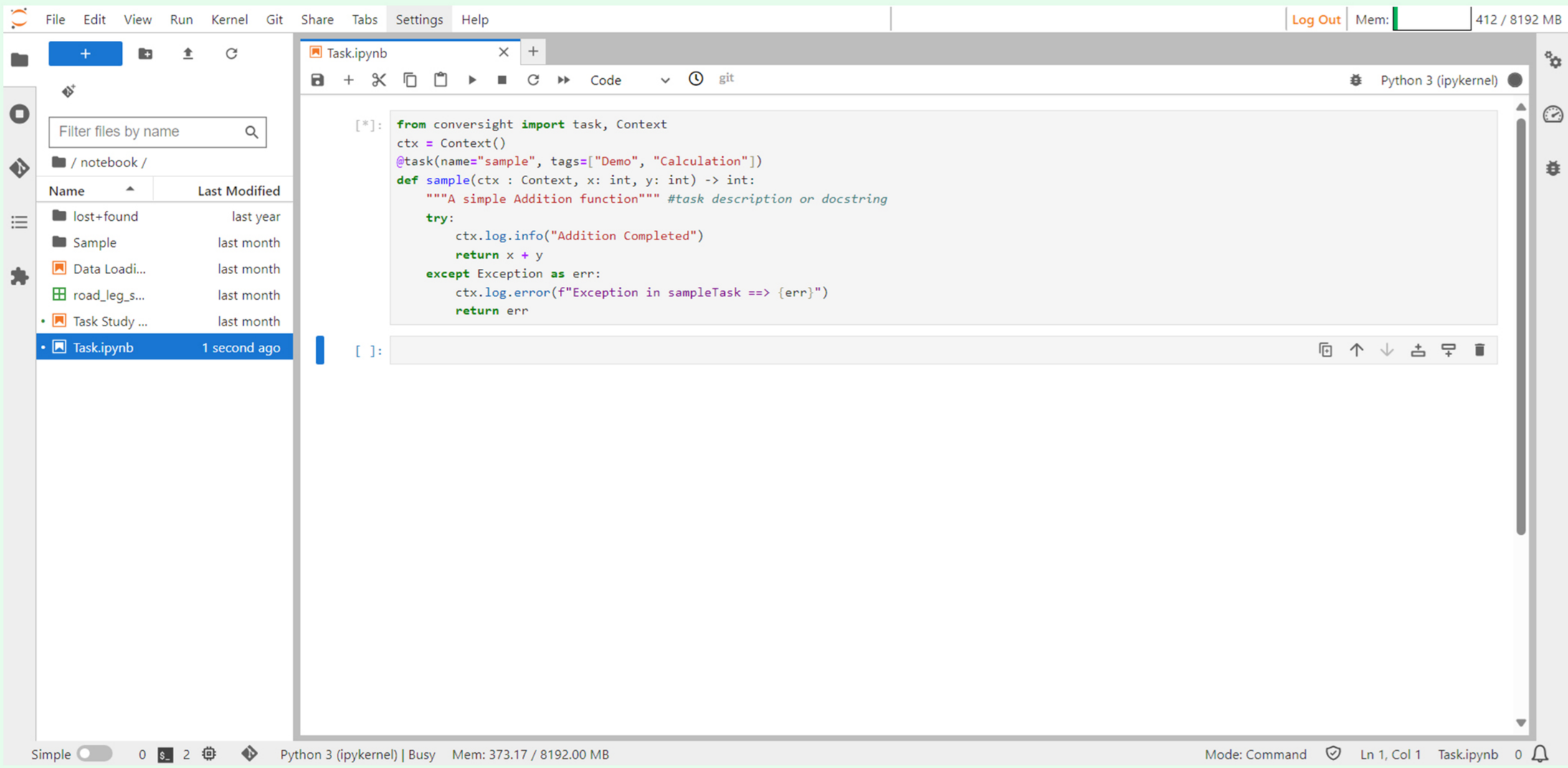
- Give your task a name, which will be crucial for calling it into workflows.
- Save the Notebook containing your task by clicking File--> Save.



- Define the function you wish to transform into a task using the **'@task'** decorator at the beginning of the code.

Note

In ConverSight, it is imperative to include a try and except block within every task as a fundamental practice.



- Now, when creating a workflow, you can call the task you created as a step in the workflow.

Overall, creating a task involves defining a function with a try and except block, adding the @task decorator to it and assigning a name to the task. Once saved, the task can be used in workflows and executed as a standalone unit.

3.2 Elevating Performance with Intelligent Task Functions

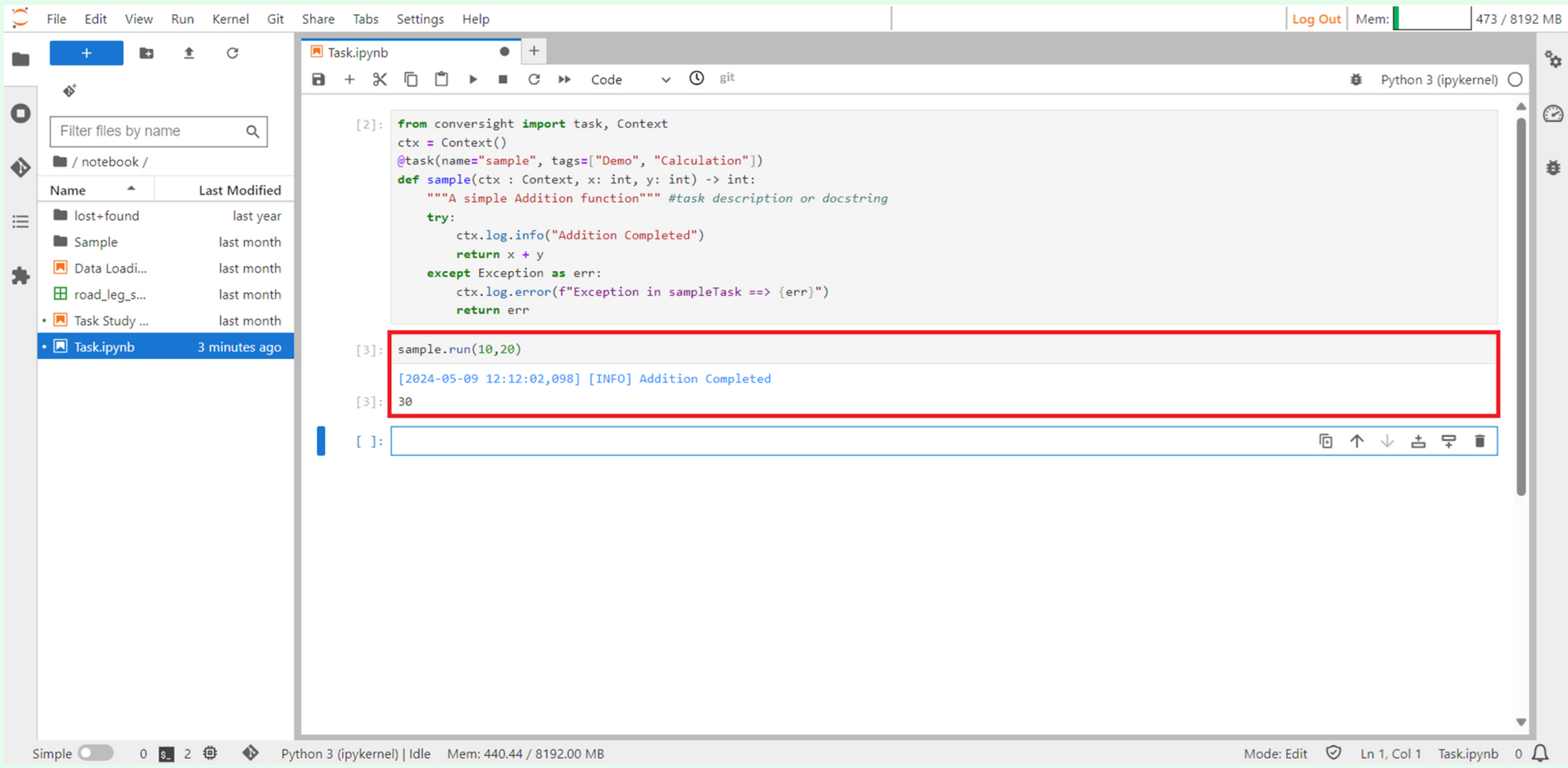
After creating a Task, you can perform functions like run, register, promote and reload.

Here are the functions of Tasks:

Running a Task

Tasks can be executed within the system using the **'run'** method, enabling data processing, manipulation or analysis operations in a desired format.

Function_name.run(Argument 1, Argument 2)

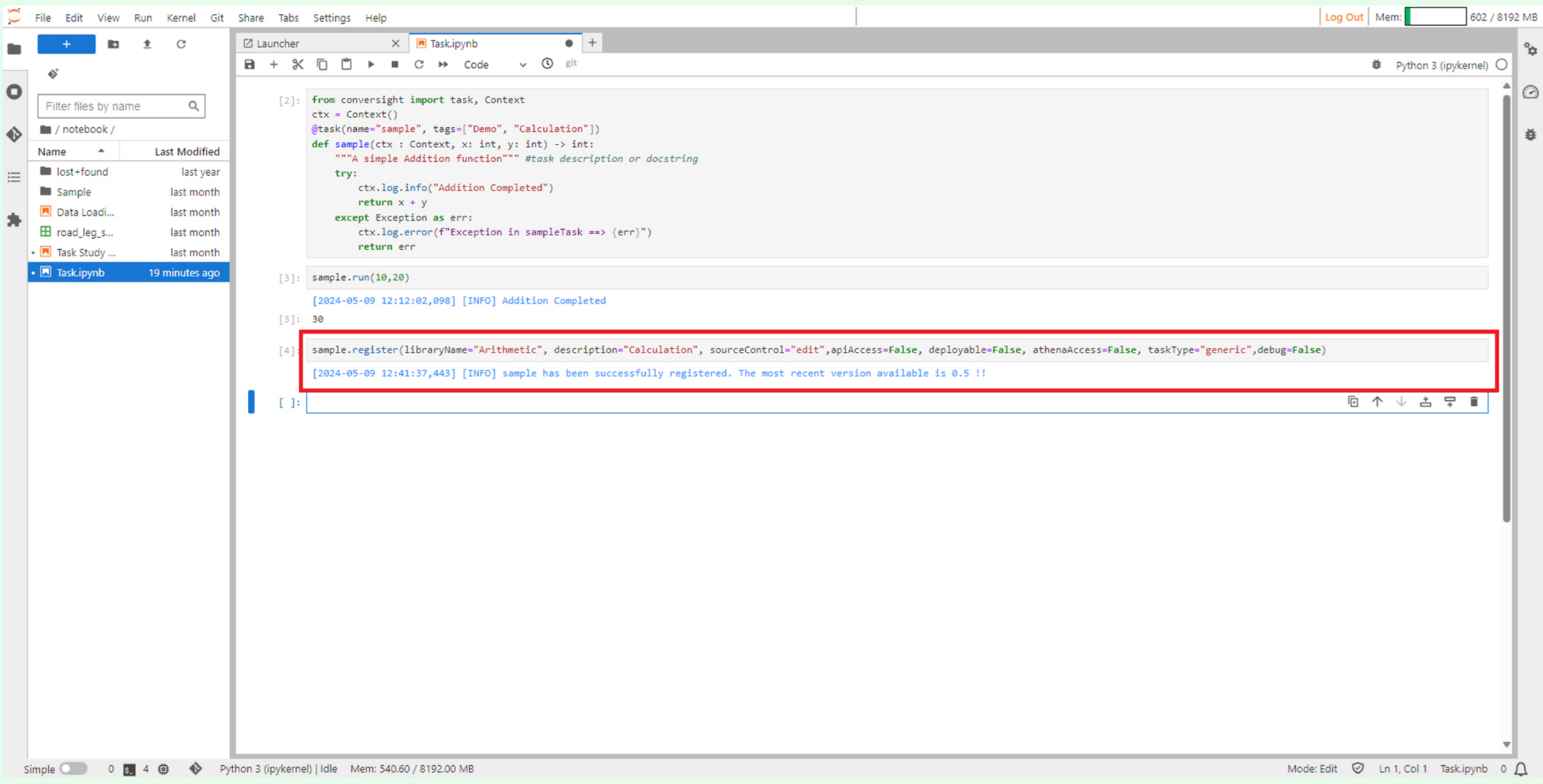


Registering a Task

Incorporating a task into the ConverSight platform necessitates the process of task registration. This imperative step empowers users to seamlessly integrate tasks, while also offering the flexibility to register multiple versions of a task, each encompassing distinct code enhancements. Furthermore, in the event of code refinements and subsequent re-registration, a new version of the task is generated, ensuring that the evolving nature of tasks is aptly captured. During task registration, essential input parameters such as library name, description, source control preferences, API access configuration, deployable status and task type specifications must be furnished. This meticulous registration process serves as the bridge that facilitates the seamless integration of tasks into the platform.

ARGUMENT	DESCRIPTION
Library Name	The library name is mandatory to register a task. It can be either a new or existing library name. If a new name is given, it will create a new library. The library is a logical grouping of tasks.
Description	Textual explanation or summary that provides context, details or information about the Task. It aims to convey a clear understanding of the subject matter to the audience.

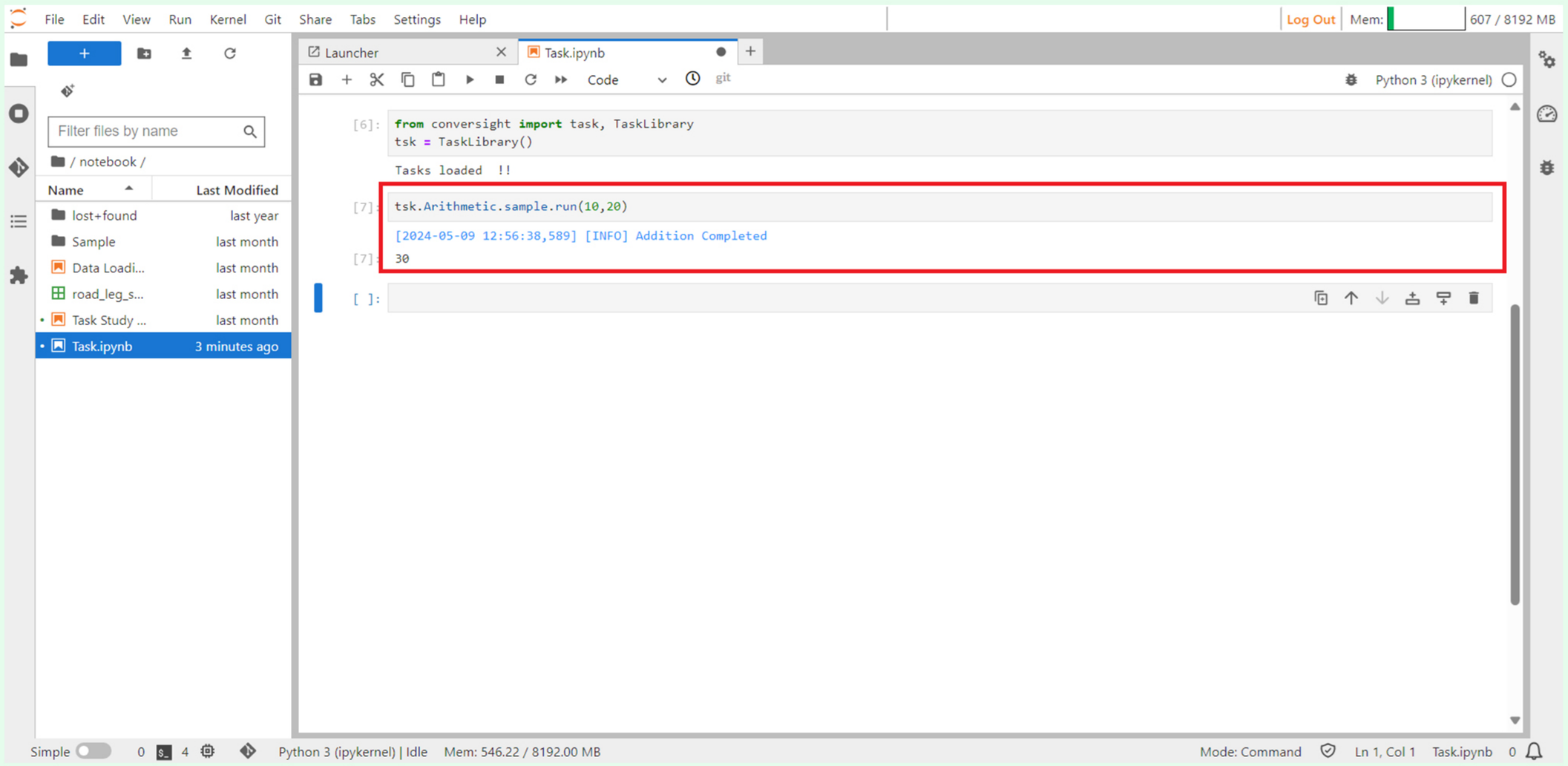
ARGUMENT	DESCRIPTION
Source Control	The Task created has access controls such as view, edit and noAccess. The default is set to noAccess. **View -** Permits users to exclusively observe the code. **Edit -** Empowers users to both review and modify the code if required. **NoAccess -** Restricts the user from both viewing and editing the code.
API Access	This is a Boolean parameter. The default is set to false. When we set it to true, the task can be executed through the API resource.
Deployable	Deployable is almost the same as apiAccess. The task will always be in memory, running at the back end and will give an instant output when input is provided. This is also a Boolean parameter. The default is set to false.
Task Type	There are two task types. They are generic tasks and UI tasks, any task we create using **'@task'** is generic. UI task is created with **'@uitask'**, which will work only on the user interface. The default is set to generic task.



Upon successful registration of a task, users gain the ability to execute it directly within the Notebook, eliminating the need for the underlying code.

Here is how to execute a task without code directly from the notebook:

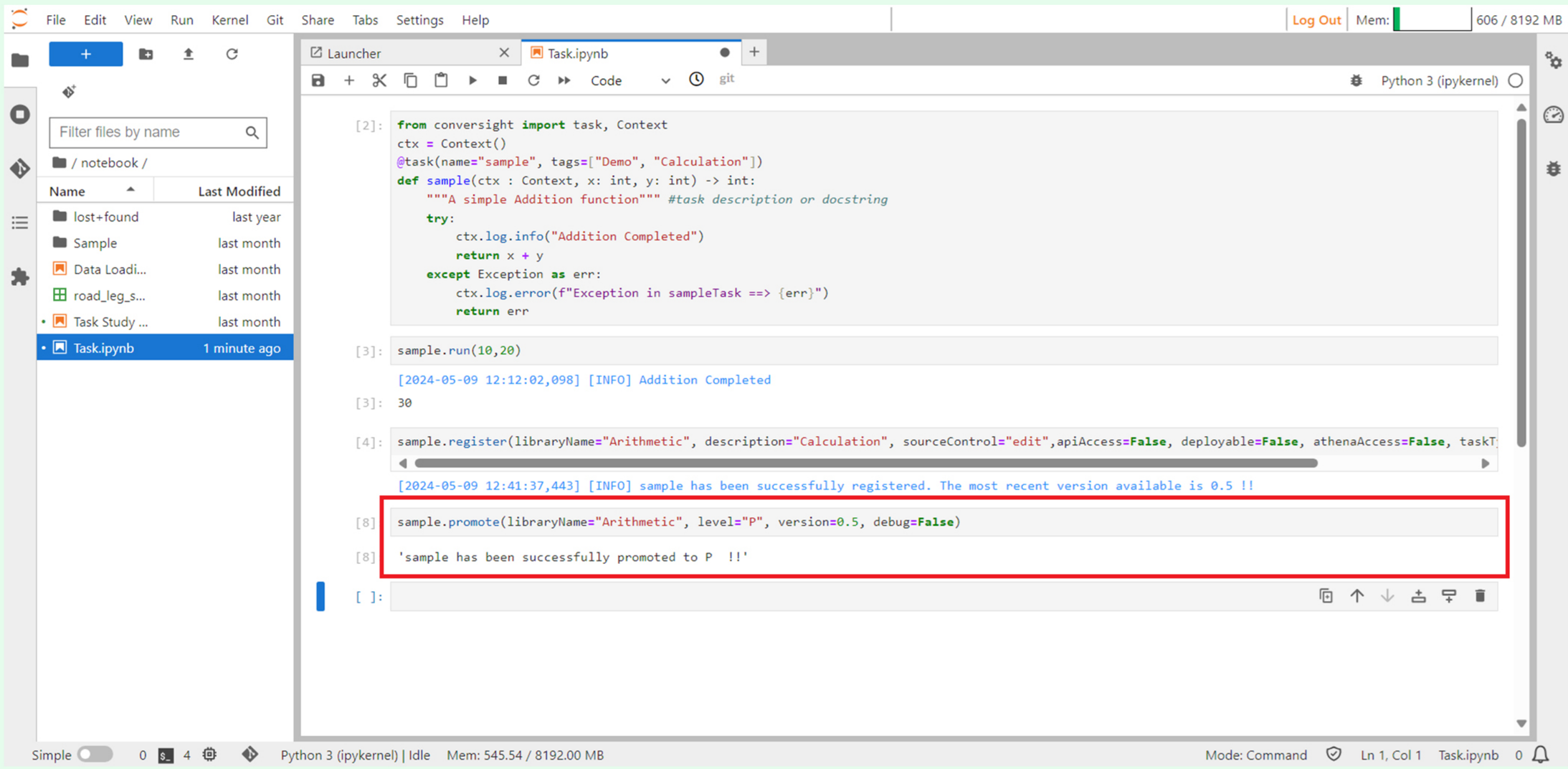
- Go to the configuration sidebar and choose the Notebook option.
- Open a new Notebook.
- Import the required libraries and modules.
- Utilize the **‘run()’** function using the provided syntax to execute the task without code directly from the notebook.
  tsk.LibraryName.TaskName.run(Argument1, Argument2).



Promoting a Task

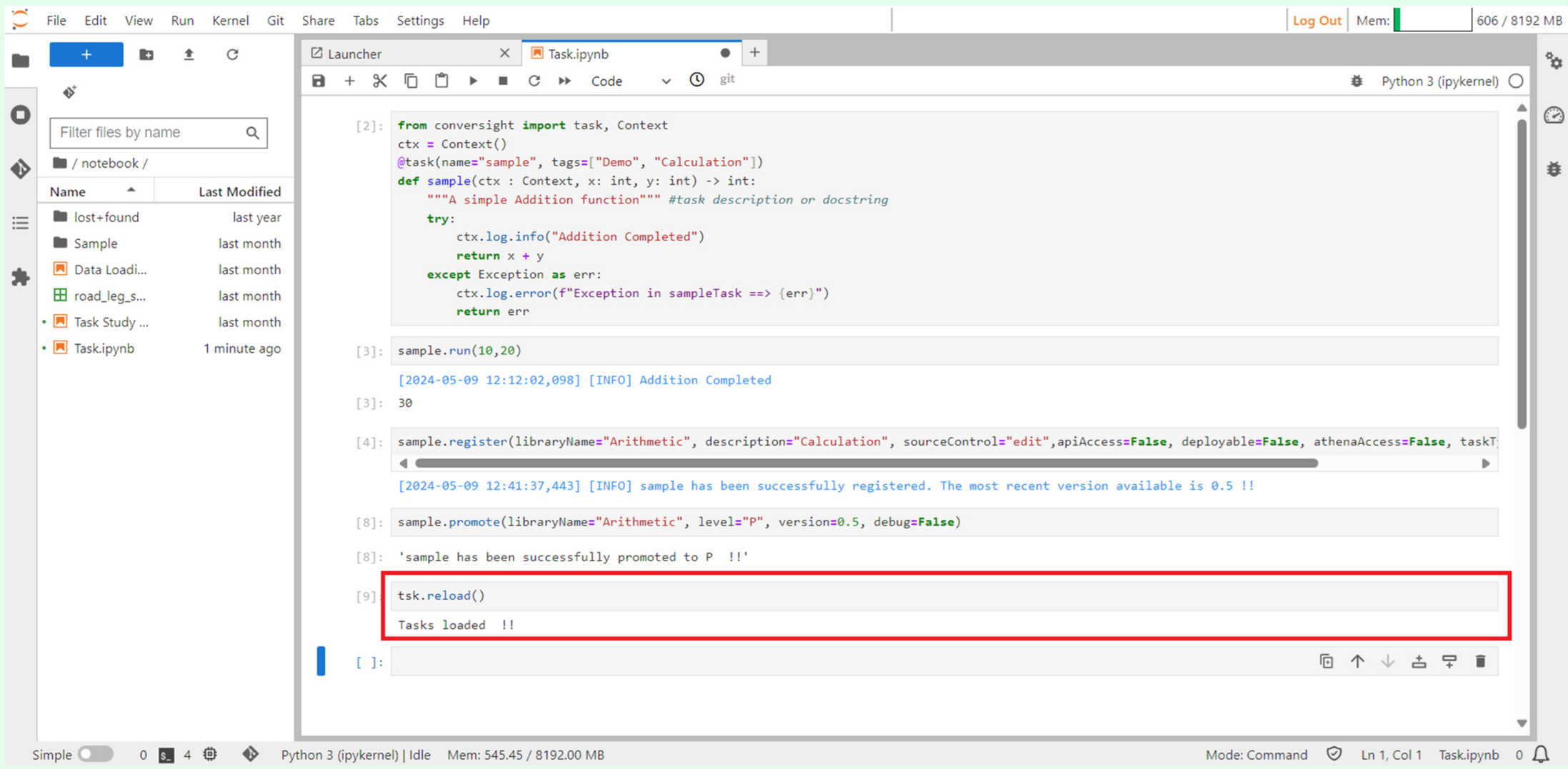
ConverSight boasts a distinguished Access Control Hierarchy, characterized by three discernible tiers of task promotion: User level (**’U’**), which caters solely to the individual; Organizational level (**’O’**), extending accessibility to all within the organization and Platform level (**’P’**), granting accessibility rights to every member of the platform. Aligned with specific needs, tasks can ascend to any of these esteemed tiers. Once promoted, a task can be used across diverse workflows.

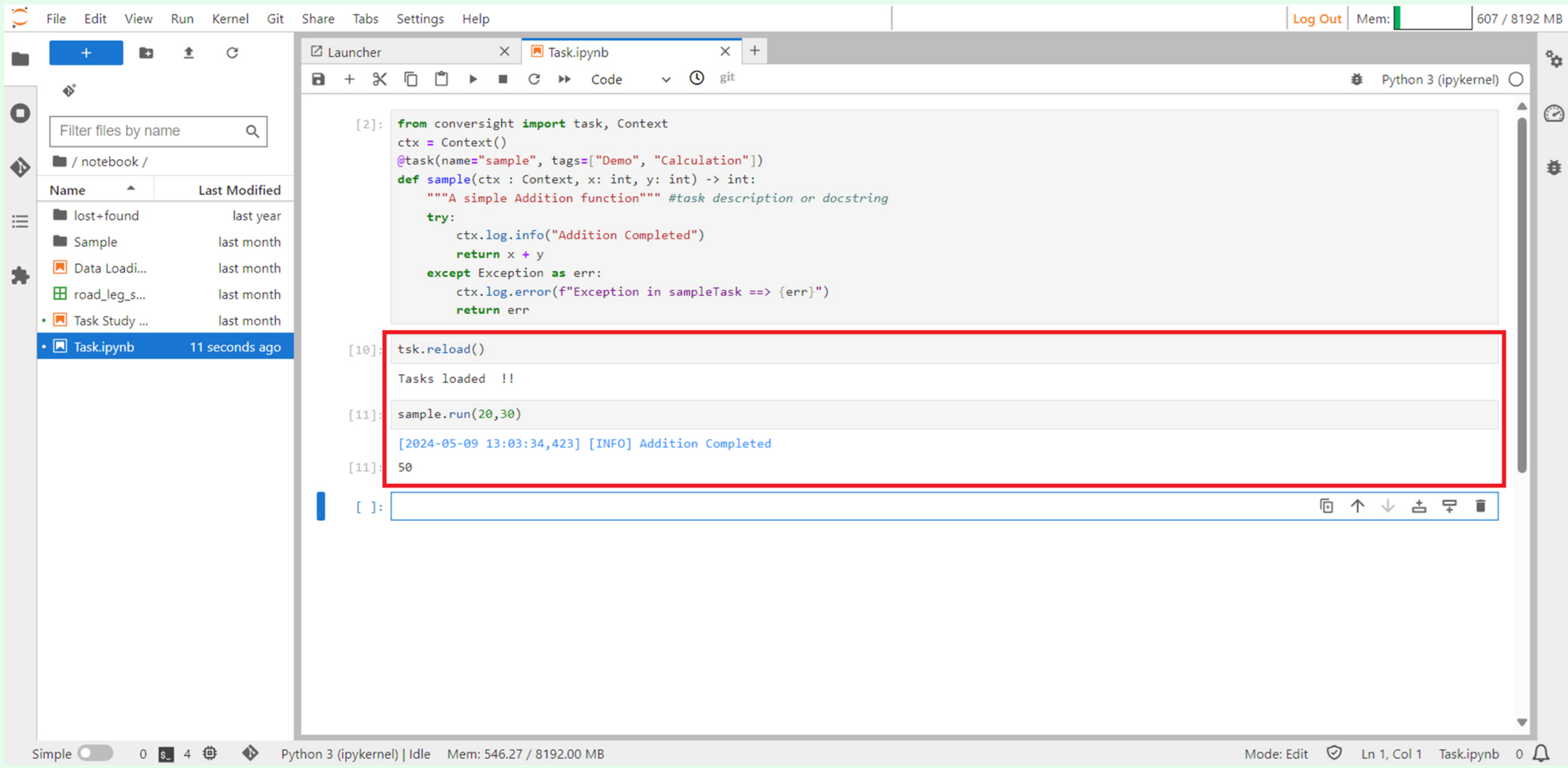
Taskname.promote(“LibraryName”,”level”)



Task Reloading

In the process of task registration, the initial version of the task is saved. However, as code enhancements and modifications are made, subsequent registrations create new versions of the task. To ensure the seamless utilization of the latest updates, the 'reload()' function becomes indispensable. By employing the 'reload()' function, users can refresh the task and access the most up-to-date version, thereby optimizing workflow efficiency and promoting the use of the latest task enhancements. In cases where the task's output demands an extended duration, users might find it necessary to employ the 'reload()' function to refresh the libraries and execute the task anew to optimize time utilization.

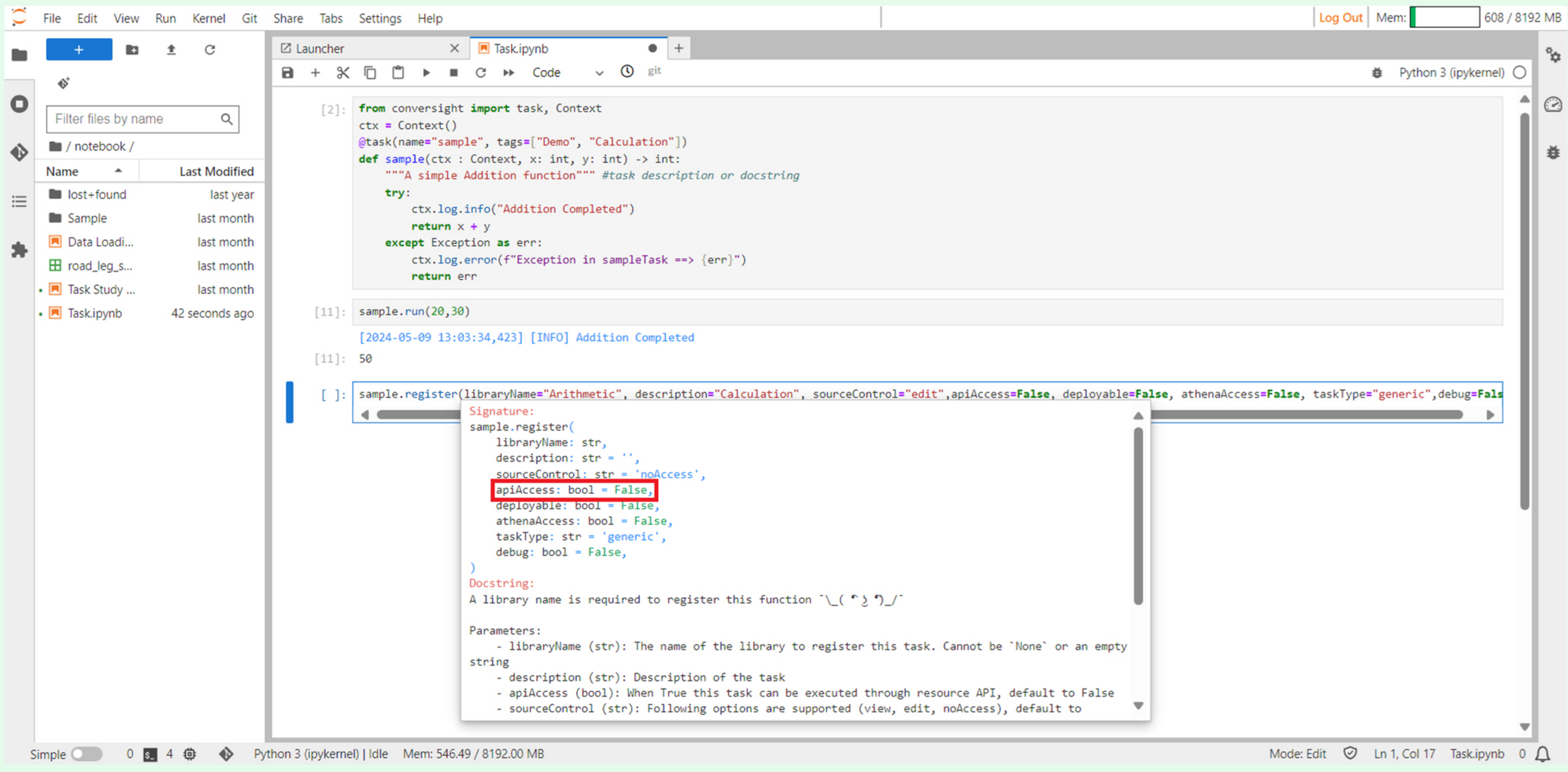




3.3 Task as API

Task as API in Python refers to the practice of exposing a specific task or function as a web-based interface, allowing external systems or applications to remotely trigger and execute the task by sending HTTP requests. This approach facilitates seamless integration and automation between different software components, enabling efficient communication and remote task execution.

To establish a task as an API, during the task registration process, users are required to enable the apiAccess option. This ensures that the task operates as an API when executed.

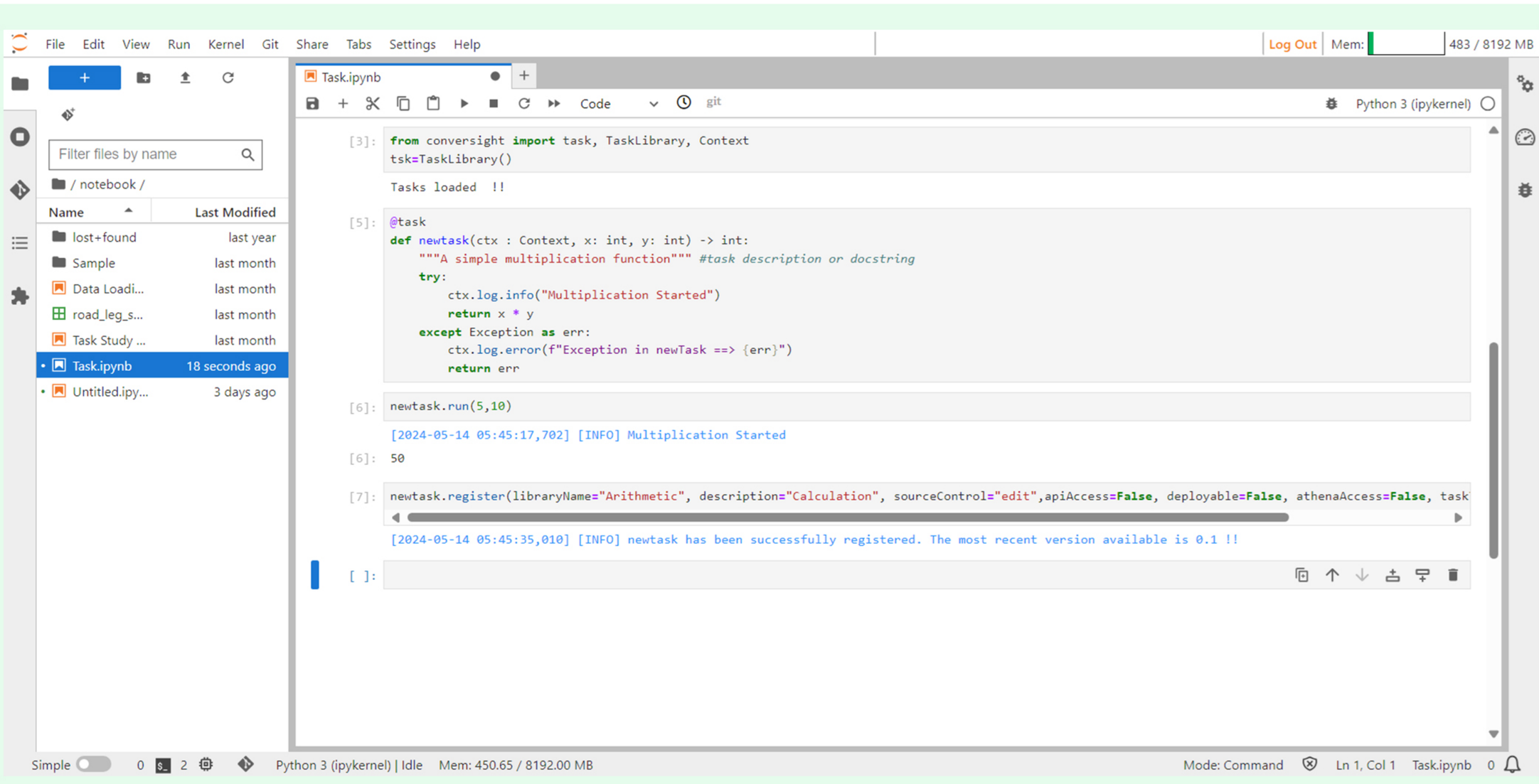


3.4 Utilizing Chained Tasks

Chained Tasks in ConverSight refers to a series of tasks where the output of one function becomes the input for the next task in a sequential manner. This composition of tasks creates a chain or pipeline, allowing the reuse of functions or tasks, efficient code design and an organized way to process data or perform complex operations.

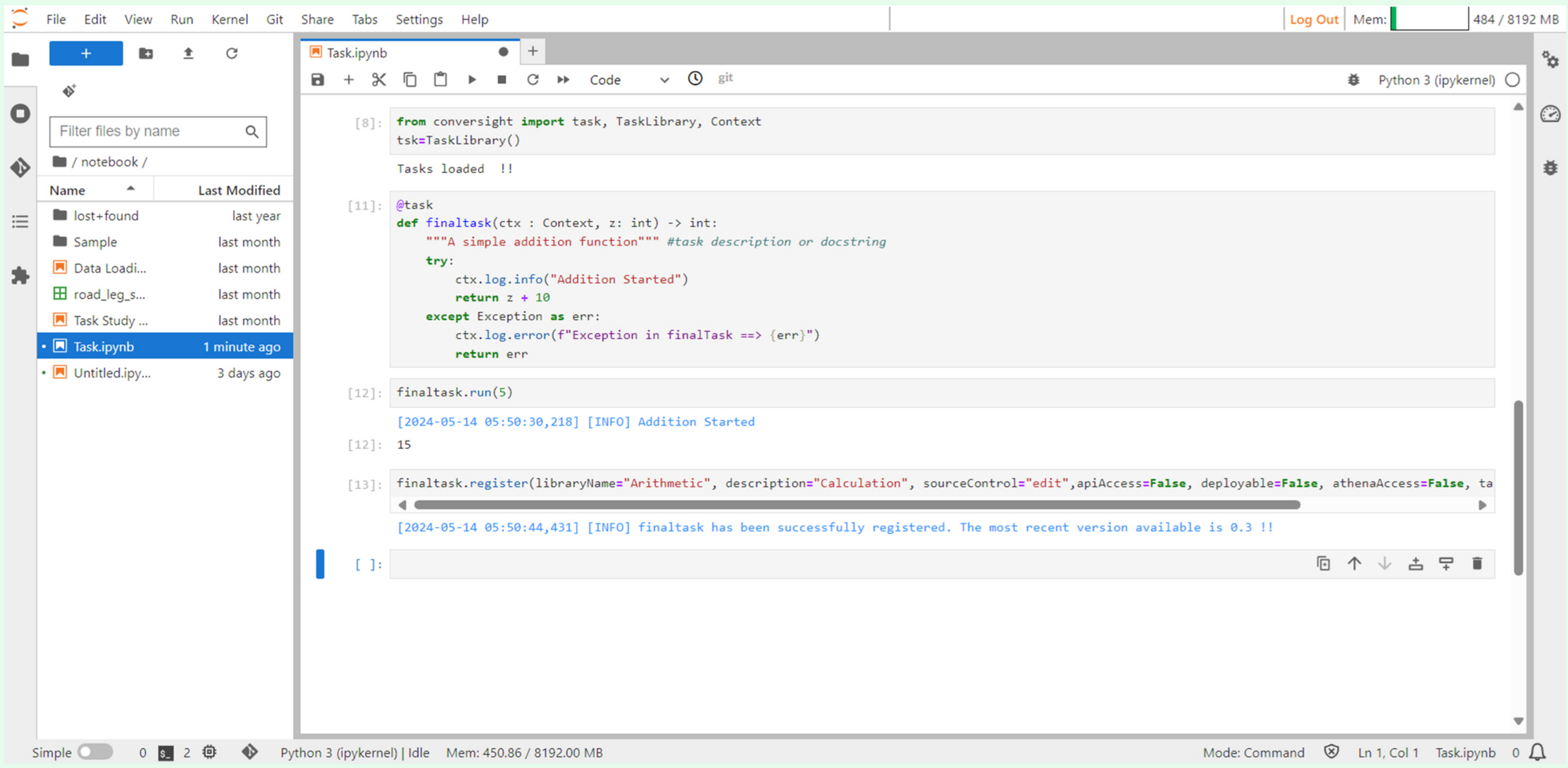
To use the chain tasks or output of one task which acts as the input for another, follow these steps:

- Create two separate tasks, as explained earlier



- Once created, register both tasks using the register() command,

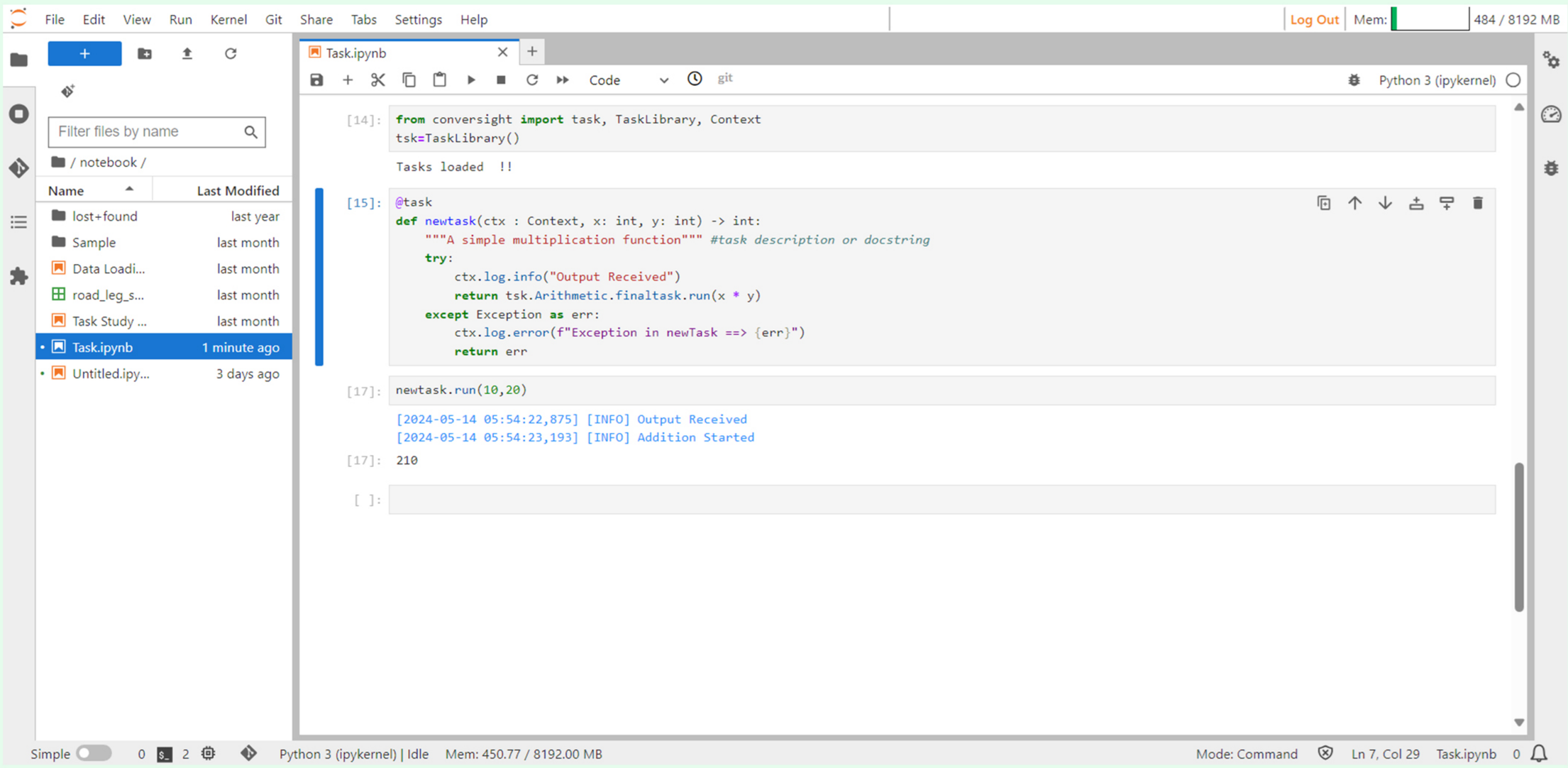
Syntax:TaskName.register(“LibraryName”,”description”,”sourceControl”, apiAccess, deployable, athenaAccess, ”taskType”, debug)



Pass the output of the first task as the input parameter as shown below for the second task to get the final result.

```
from conversight import task, TaskLibrary, Context
tsk=TaskLibrary()
@task
def newtask(ctx : Context, x: int, y: int) -> int:
    """A simple multiplication function""" #task description or docstring
    try:
        ctx.log.info("Output Received")
        return tsk.Arithmetic.finaltask.run(x * y)
    except Exception as err:
        ctx.log.error(f"Exception in newTask ==> {err}")
        return err
```

Execute the task that yields the required final outcome using the **‘run()’** command.



3.5 Benefits of Tasks

- **Enhanced Flexibility and Accessibility:** Tasks offer a level of flexibility and accessibility like conventional programming language functions. Users can execute specific data processing, manipulation or analysis operations effortlessly, tailoring workflows to suit their unique needs.

- **Efficient Concurrent Operations:** Tasks can be executed sequentially or in parallel, allowing for concurrent operations and streamlined workflow sequences. This capability optimizes resource utilization, reduces processing time and enhances overall efficiency.

- **Reusable Components for Workflow Creation:** Tasks encapsulate elements that can be easily reused across various flows in AI Workbench. This modularity simplifies workflow creation, promoting component reuse and increasing the scalability of data processing pipelines.

- **Collaboration and Knowledge Sharing:** Tasks can be easily shared and reused across different projects within ConverSight AI Workbench. This fosters collaboration among users, encourages knowledge sharing and promotes the development of adaptable and scalable workflows.

- **Easy Task Creation and Execution:** The step-by-step process for Task creation and execution simplifies workflow management. Users can define functions, transform them into Tasks using the **'@task'** decorator and call them into workflows with ease. Task execution is straightforward, allowing users to monitor progress and update results as needed.

- **Version Control and Reloading:** Task registration with detailed input arguments ensures version history is preserved, facilitating easy retrieval of previous versions. Additionally, Task reloading allows for the creation of updated versions, ensuring the most current functionalities are used for data processing tasks.

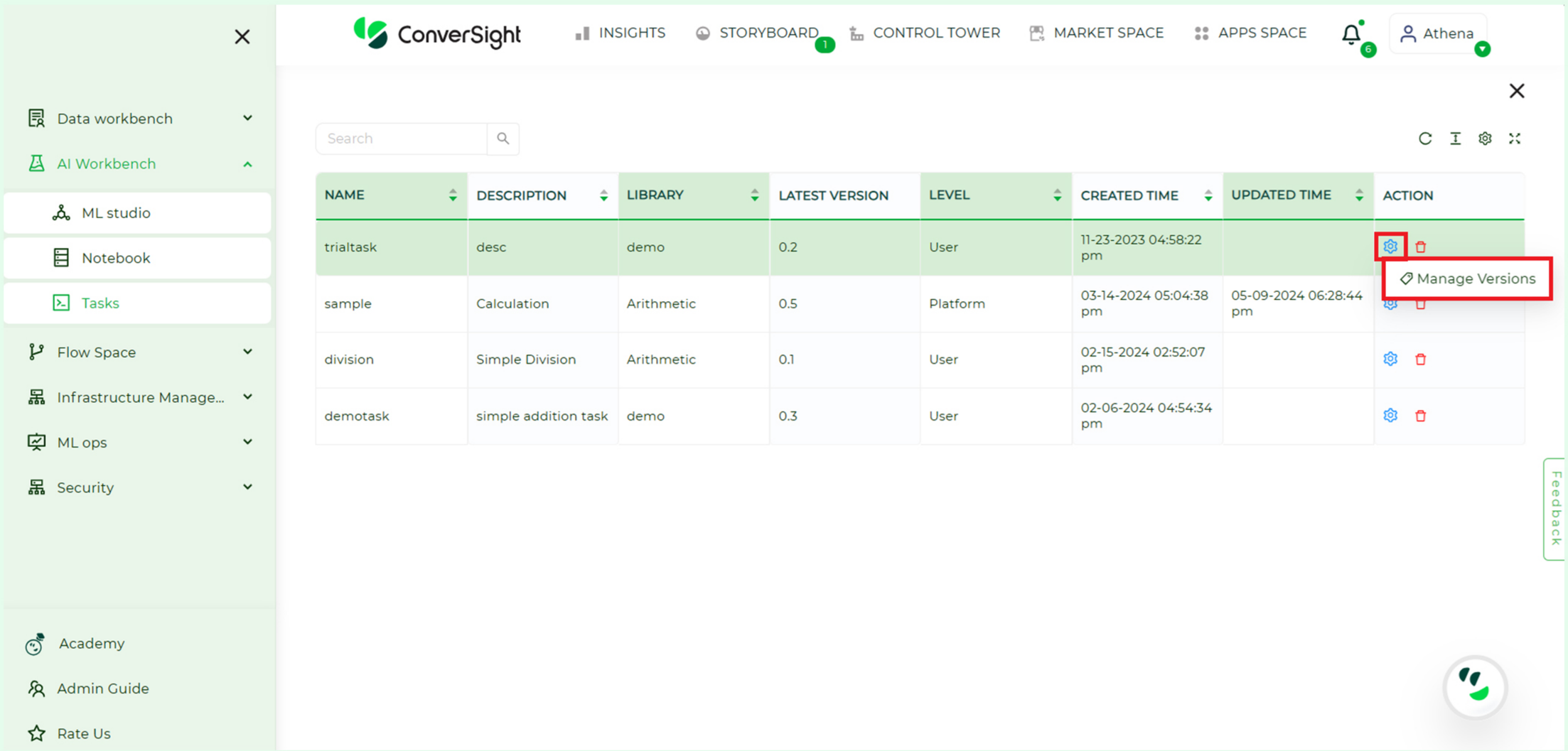
3.6 Task Management

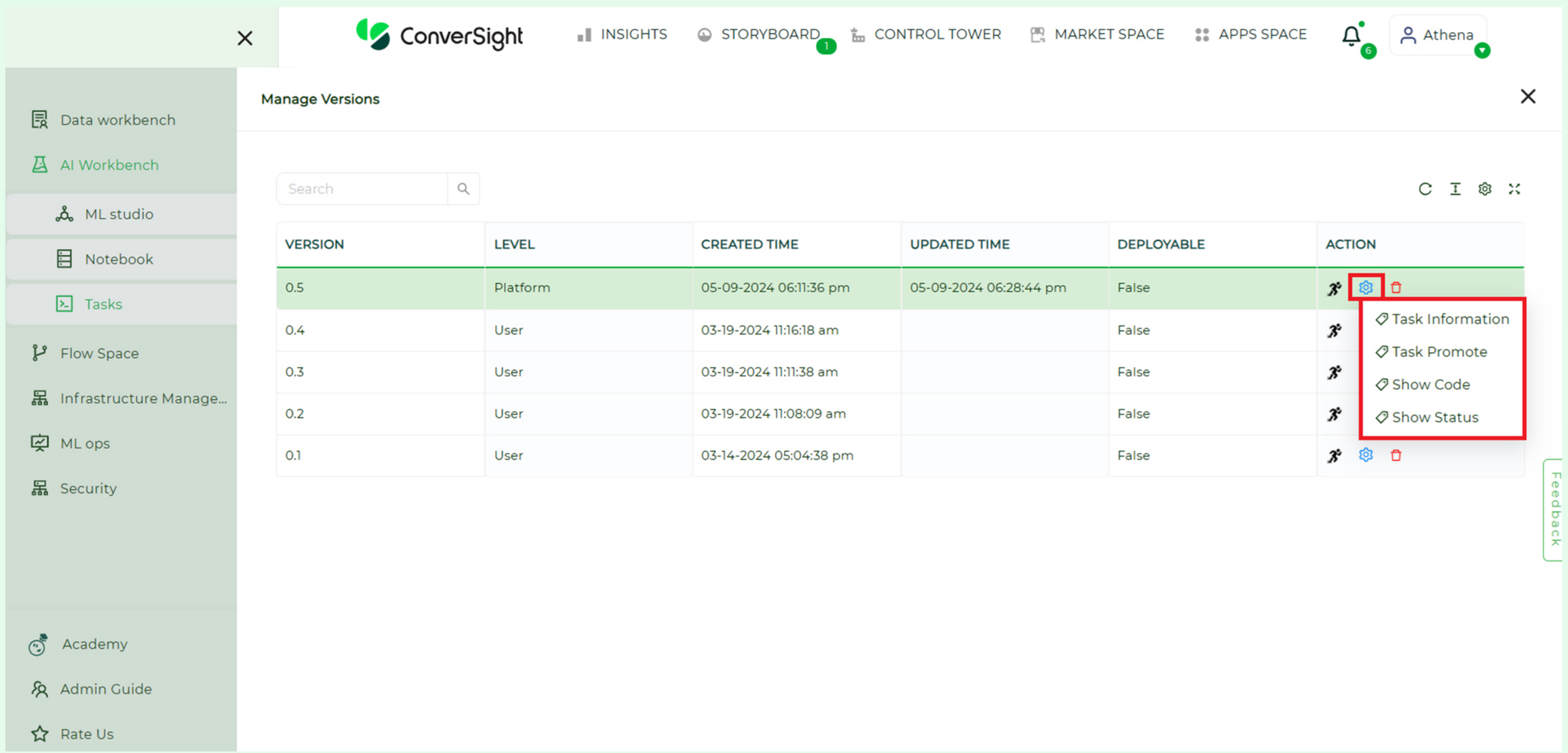
Within ConverSight's AI Workbench, the "Tasks" feature stands as a user-friendly UI menu, serving as a centralized hub for users to conveniently monitor and manage all their created and registered tasks. This intuitive interface offers easy access to a comprehensive list of registered tasks, neatly organized into different libraries, streamlining task management for enhanced productivity.

Accessing

Upon creating and registering a task within the Notebook, it seamlessly integrates into the **'Tasks'** menu. Users can effortlessly access their tasks by navigating to the AI Workbench and selecting the **'Tasks'** option. Here, they gain the ability to promote or delete tasks, providing full control over task management.

To manage versions of the task in the **'Tasks'** menu, users can click on the **'Settings'** icon under the Action Menu of the desired task and select **'Manage Versions'**. This displays the task's version, promotion level, created and updated time. The settings icon under the Action column in the Manage Versions page, displays the Task Information, Task Promote- allowing the task to be promoted to three different levels and Show Code displaying the code of the associated task.





- **Task Information –** Displays the description of the task.
- **Task Promote** – The task can be promoted to three different levels – User, Organization and Platform.
- **Show Code** – Displays the code of the task which has view and edit access levels.
- **Show Status** – Displays the run status of the selected task.

4. Flow

A Flow in ConverSight is the core component of the system. Similar to tasks, Flows take inputs, perform operations and generate outcomes. Flows act as managers, coordinating tasks for efficient execution, making complex tasks achievable within ConverSight. It is crucial to note that a Flow is made up entirely of Tasks. When dealing with multiple tasks, organizing them into a Flow is helpful. This enables tasks to run sequentially or parallelly, tailored to your needs. Think of a Flow as a series of steps leading to a result. By using a Flow, tasks are executed systematically and effectively helping you reach your goals efficiently.

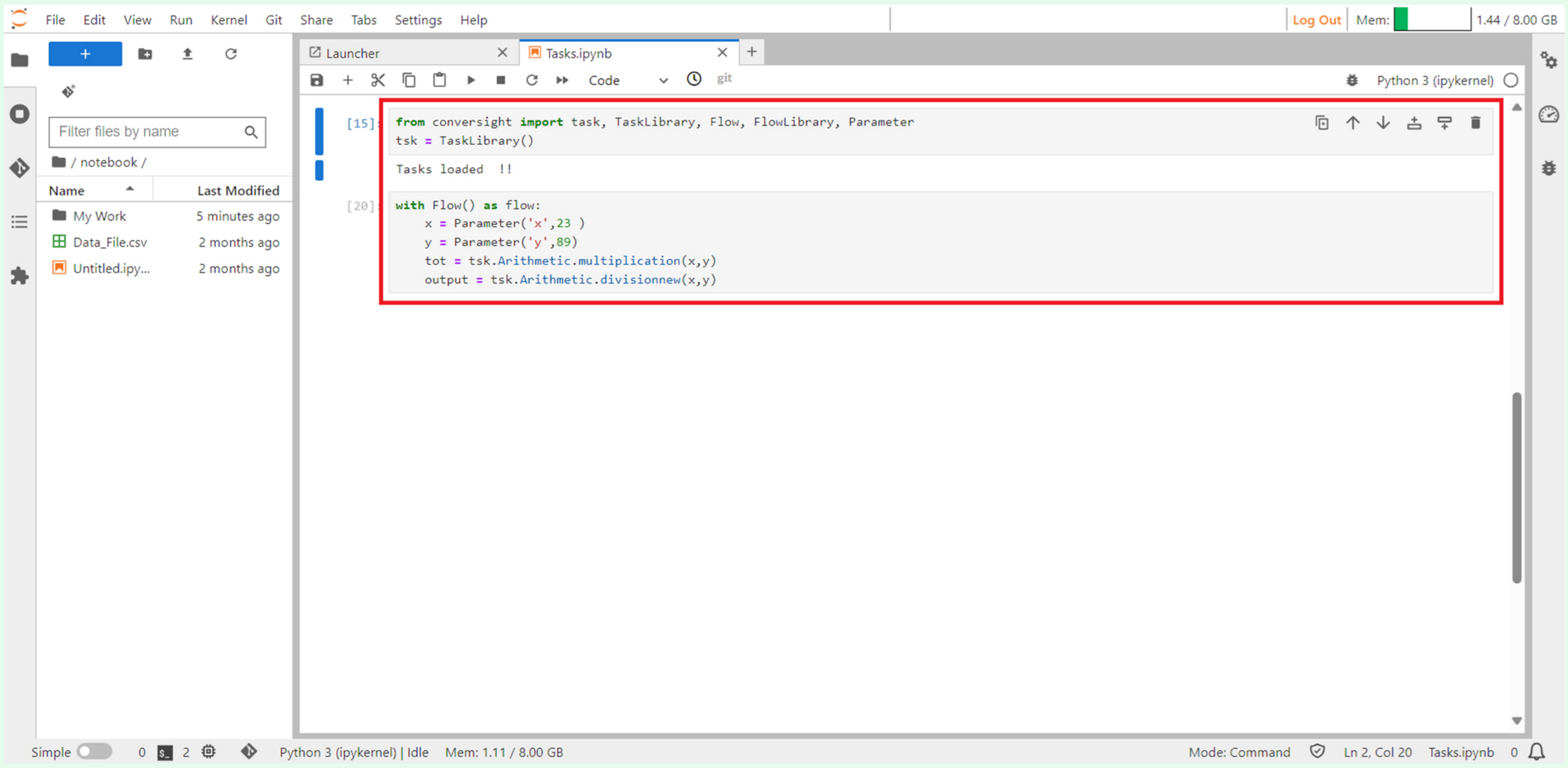
4.1 Creation

A Flow can be created in ConverSight using Jupyter Notebook available in the AI Workbench module. During the process of constructing a Flow, it is imperative for users to undertake certain preliminary steps. This encompasses the importation of essential libraries and modules, the commencement with the **'with Flow() as flow'** construct and the subsequent initialization of requisite parameters.

Sample Code

```
from conversight import task, TaskLibrary, Flow, FlowLibrary, Parameter
tsk = TaskLibrary()
with Flow() as flow:
    x = Parameter('x',23 )
    y = Parameter('y',89)
    tot = tsk.Arithmetic.multiplication(x,y)
    output = tsk.Arithmetic.divisionnew(x,y)
```

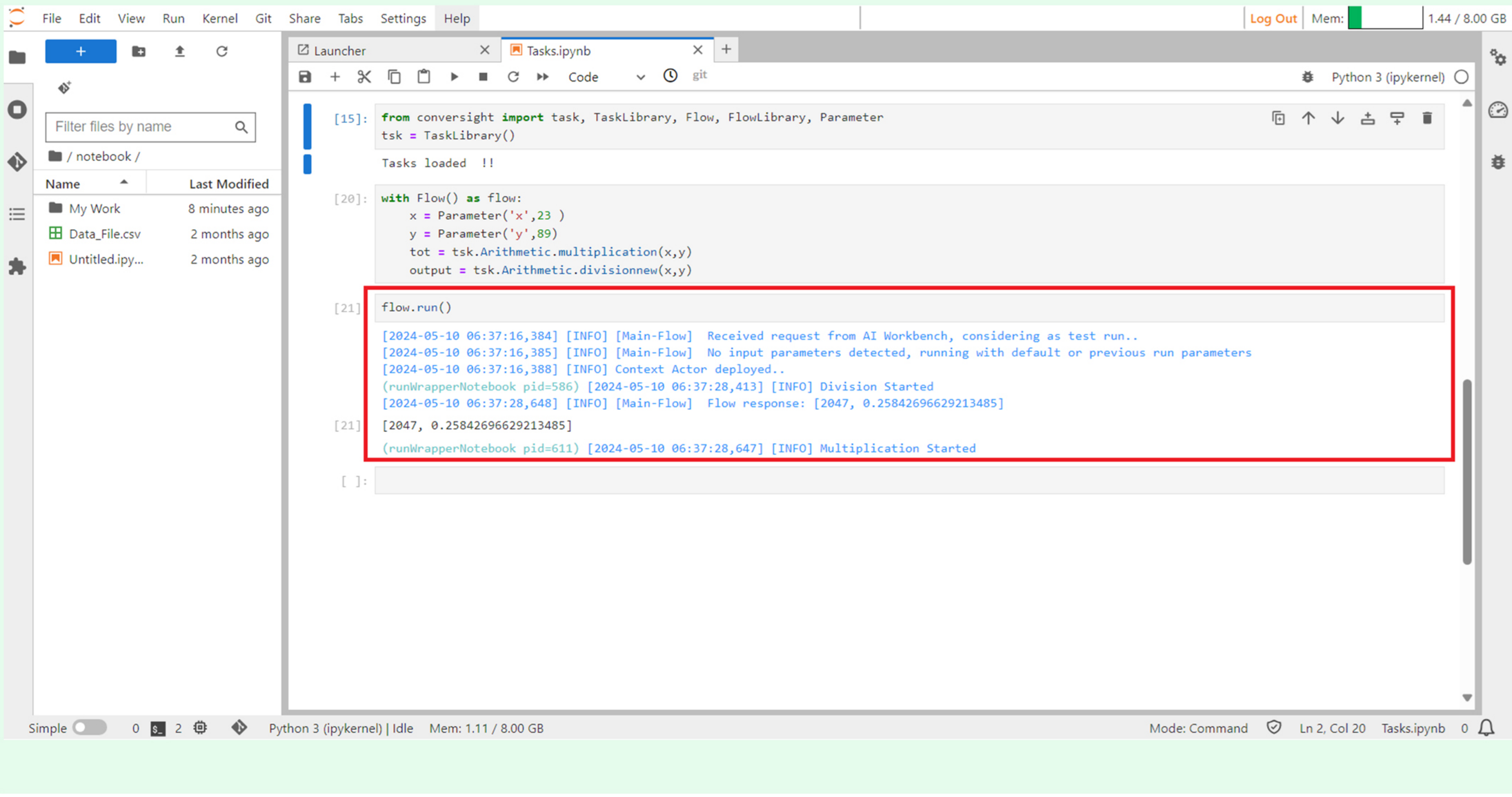
Creating a Flow is similar to the creation of Tasks, to create a Flow, you must first create tasks and register them and then use the **'with Flow() as flow'** structure and create a flow.



4.2 Empowering Seamless Flow Integration and Collaboration

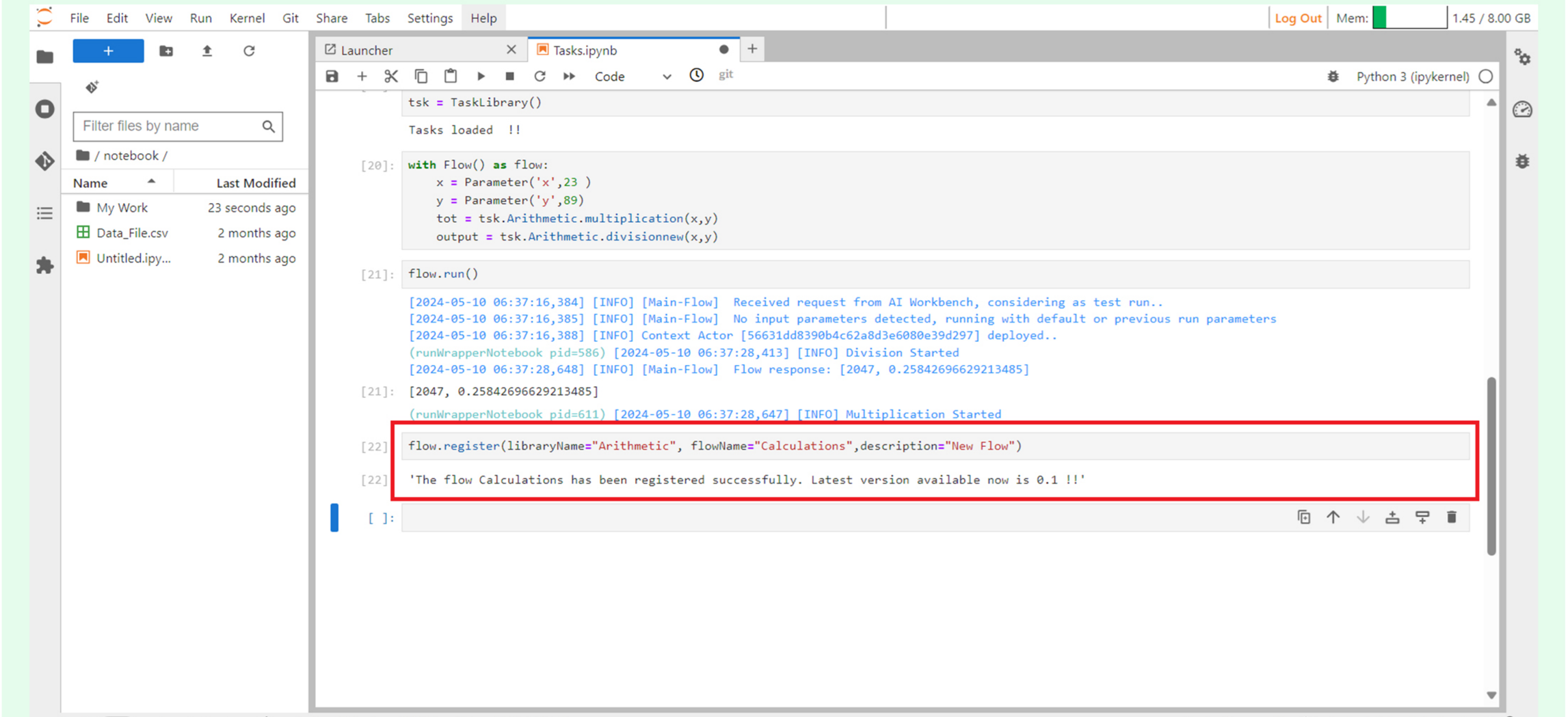
Running a Flow

Flows can be executed within the system using the **'run'** method, enabling data processing, manipulation or analysis operations in a desired format.



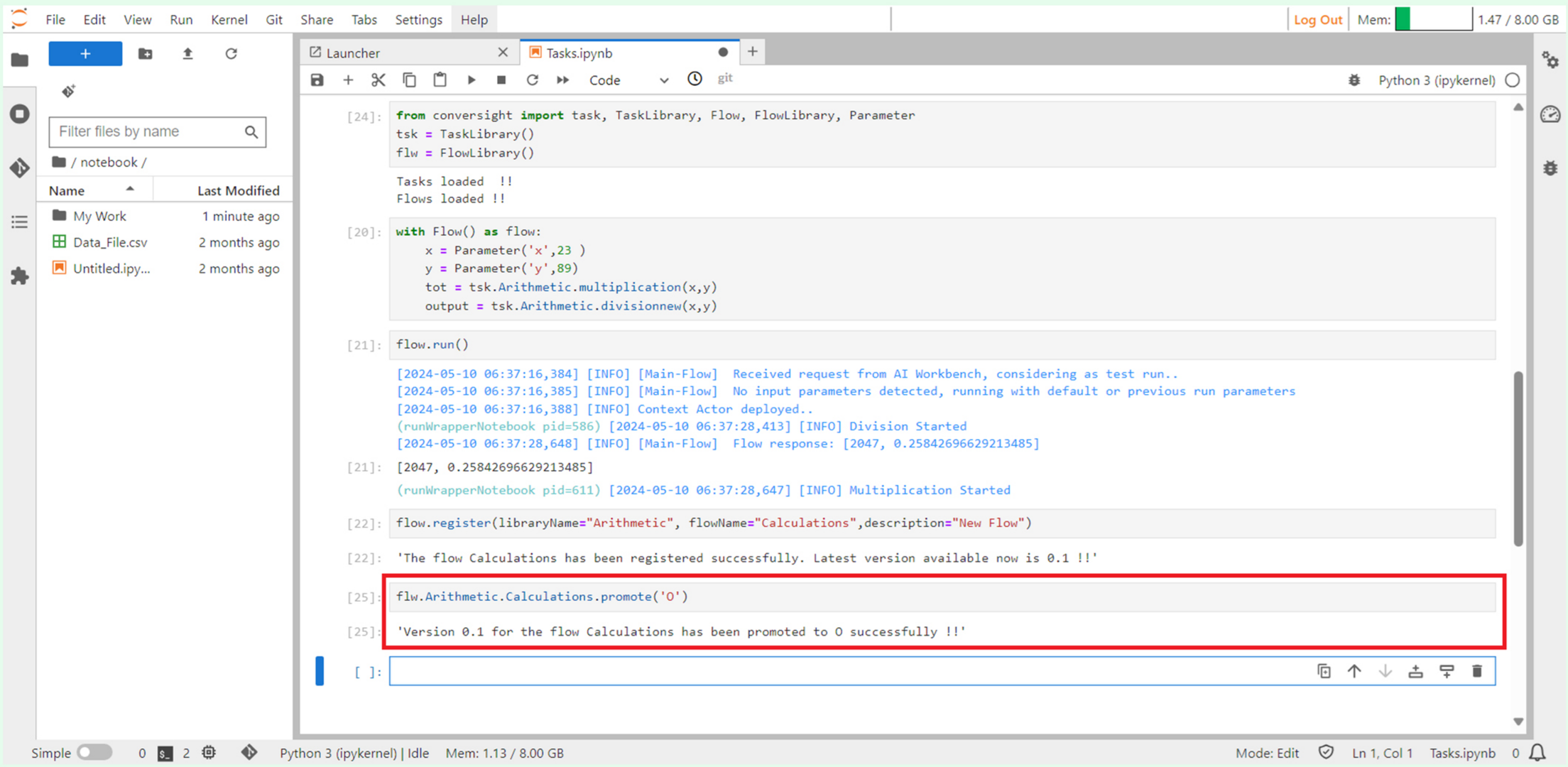
Registering a Flow

Integrating a workflow into the ConverSight platform involves the essential procedure of flow registration. This crucial phase grants users the ability to effortlessly assimilate workflows, while also providing the option to register multiple versions of a given flow. Registering a flow requires providing input arguments such as library name, flow name and description. Flow registration preserves version history, facilitating easy retrieval of the latest version.



Promoting a Flow

Promoting a Flow to different levels, such as **‘O’, ‘U’ or ‘P’**.
‘O’ represents the organization level, **‘U’** represents the user level and **‘P’** represents the platform level. Before promoting a task, it should be registered.



4.3 Benefits of Flows

Conversational AI systems heavily rely on the quality of their conversational Flows to effectively engage with users. With ConverSight AI Workbench, Flows support real-time data processing and streaming, making it suitable for a variety of use cases.

Some advantages of using Flows in ConverSight AI Workbench are:

- **Mapping Customer Journey:** Flows help visualize and map out customer journeys, enabling businesses to better understand customer behaviour and preferences.
- **Flexibility:** Tasks and Flows in ConverSight AI Workbench offer a powerful and flexible way to organize, store and reuse code, promoting easier and efficient collaboration, sharing of code and maintenance of Data Governance.
- **Testing:** Flows in AI Workbench can be utilized to create test cases and automate testing of conversational Flows and other AI applications.

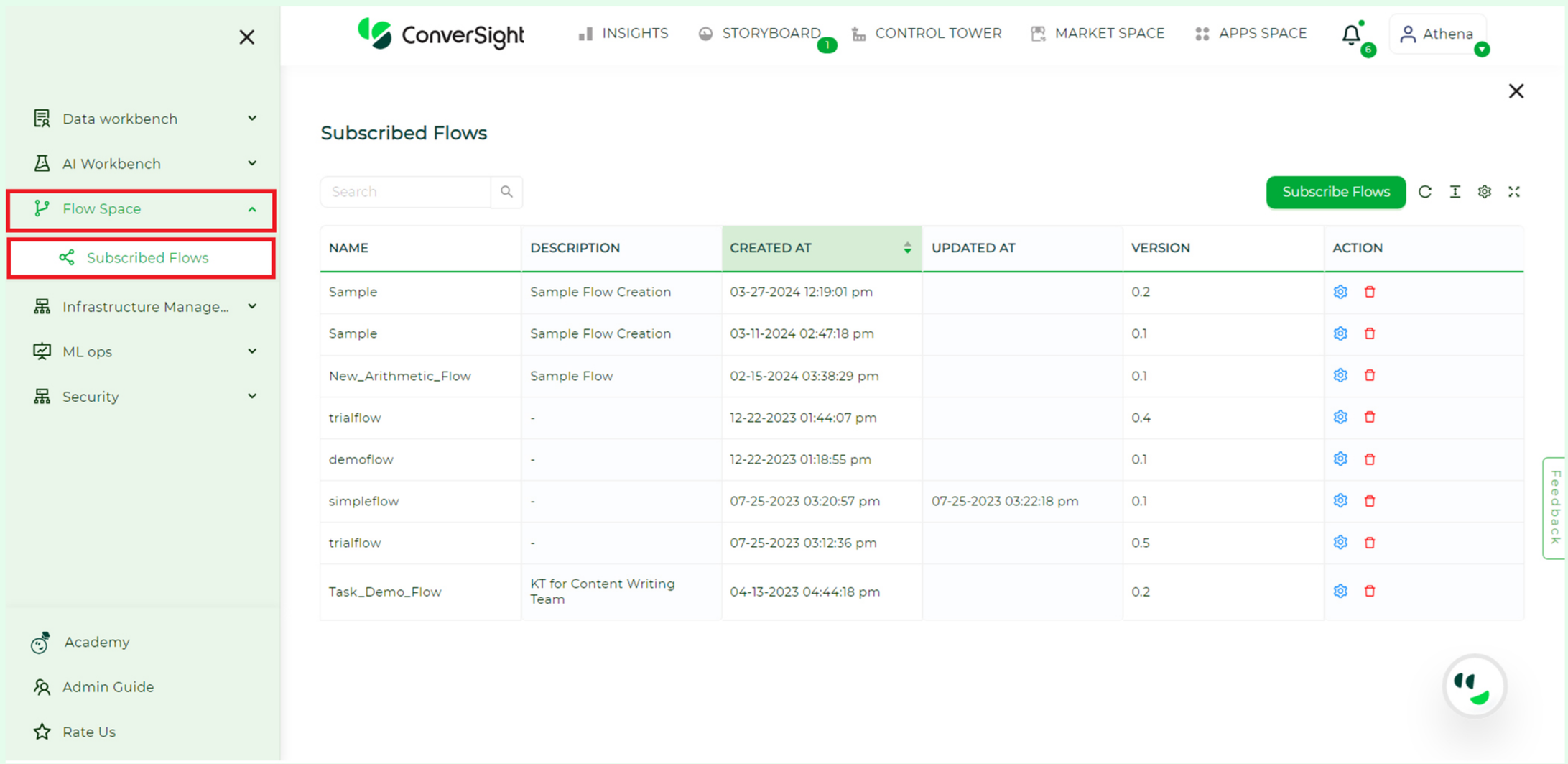
5. Subscribed Flows

5.1 Flow Management and Subscription

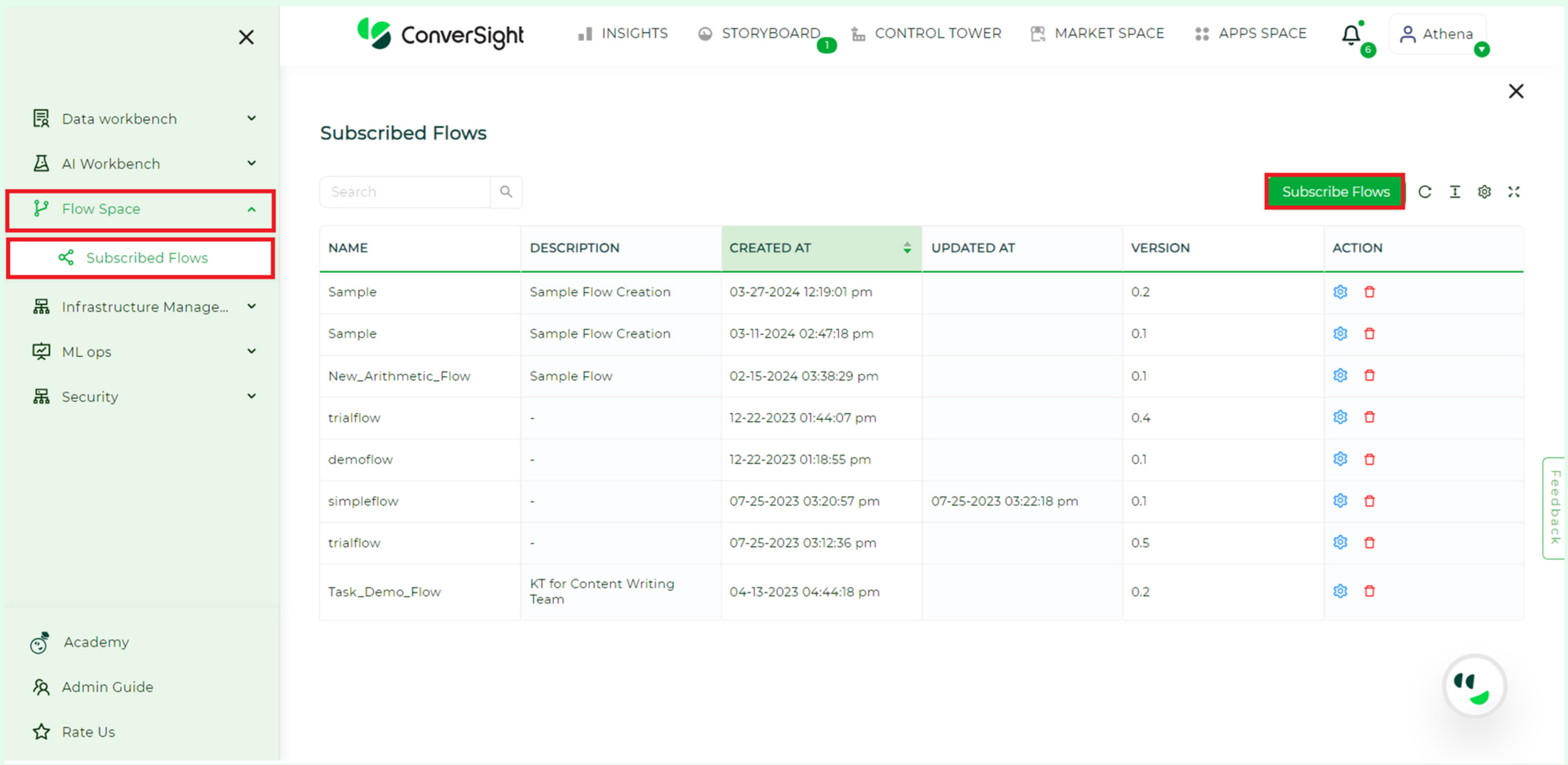
On the Subscribed Flows page, users can view all the Flows they have created and subscribed for their specific needs in one convenient location in ConverSight. In addition to listing subscribed flows, the Subscribed Flows page provides details such as flow description, creation and updation dates and currently available versions for each subscribed flow. Users can also subscribe new Flows that are promoted to platform level in ConverSight.

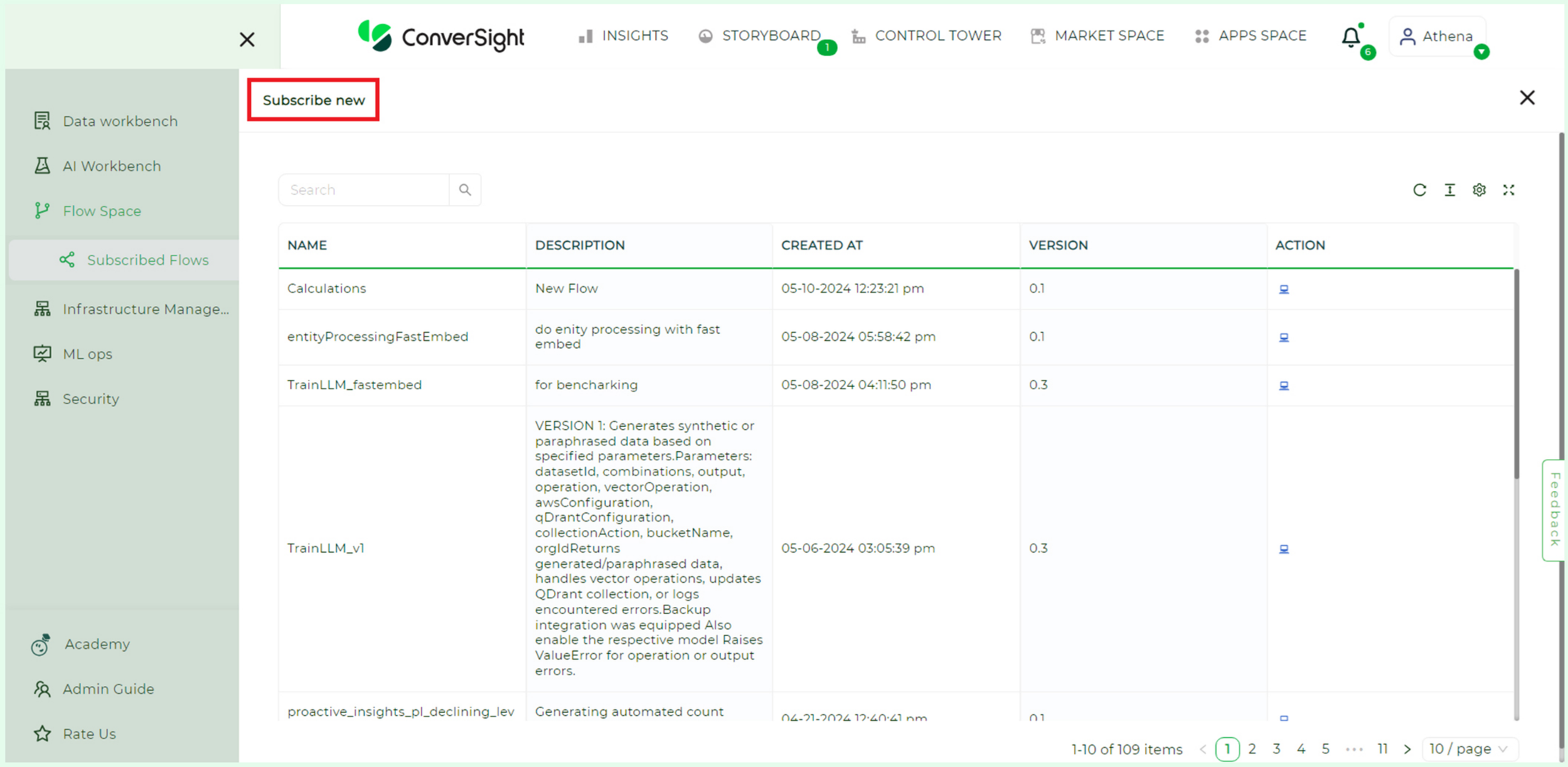
Accessing Subscribed Flows

Navigate to the Configuration sidebar and locate Flow Space and click on **'Subscribed Flows'**.



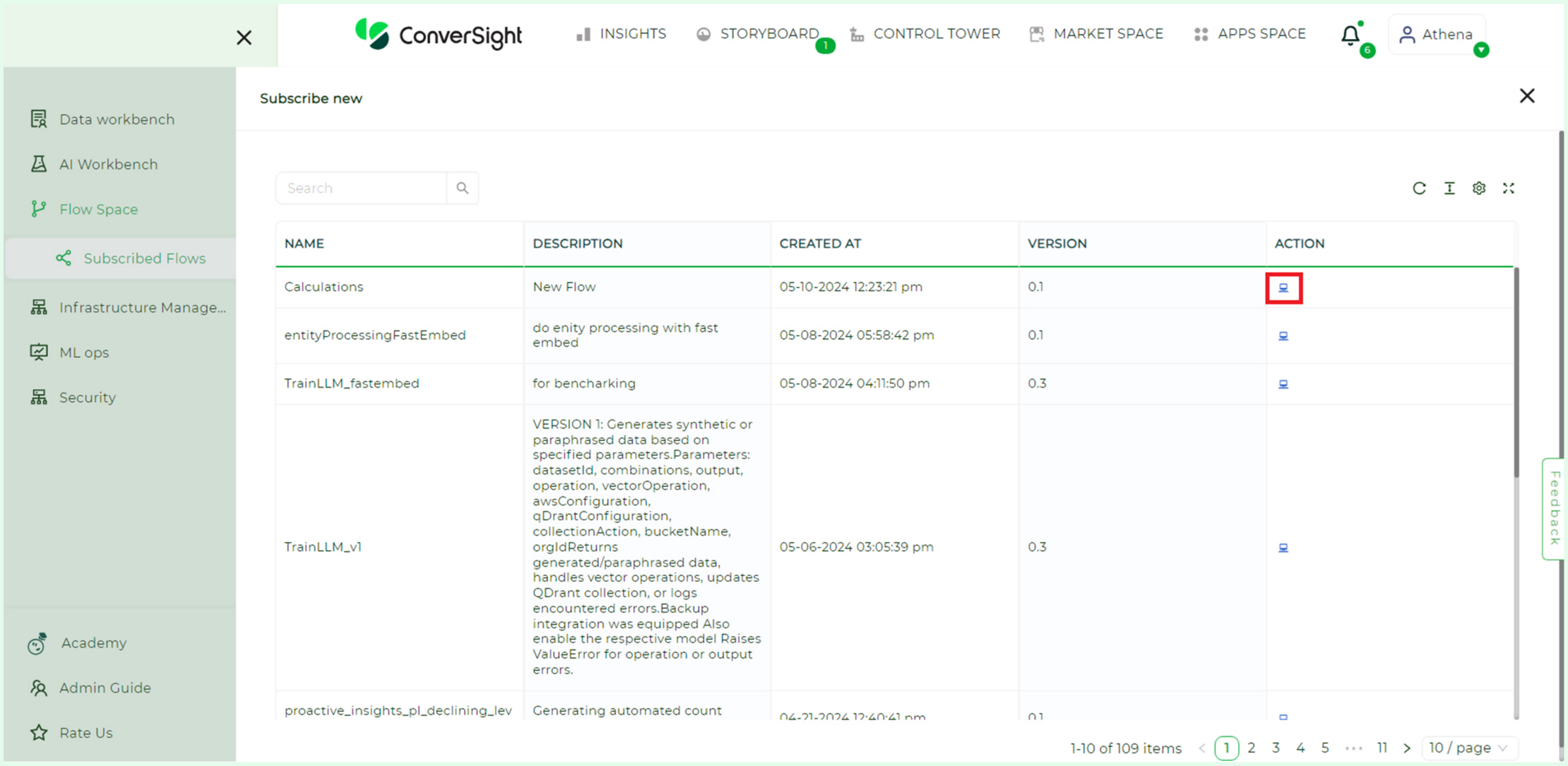
By selecting the **'Subscribe Flows'** button, users gain access to all the available Flows that have been promoted to the platform level.

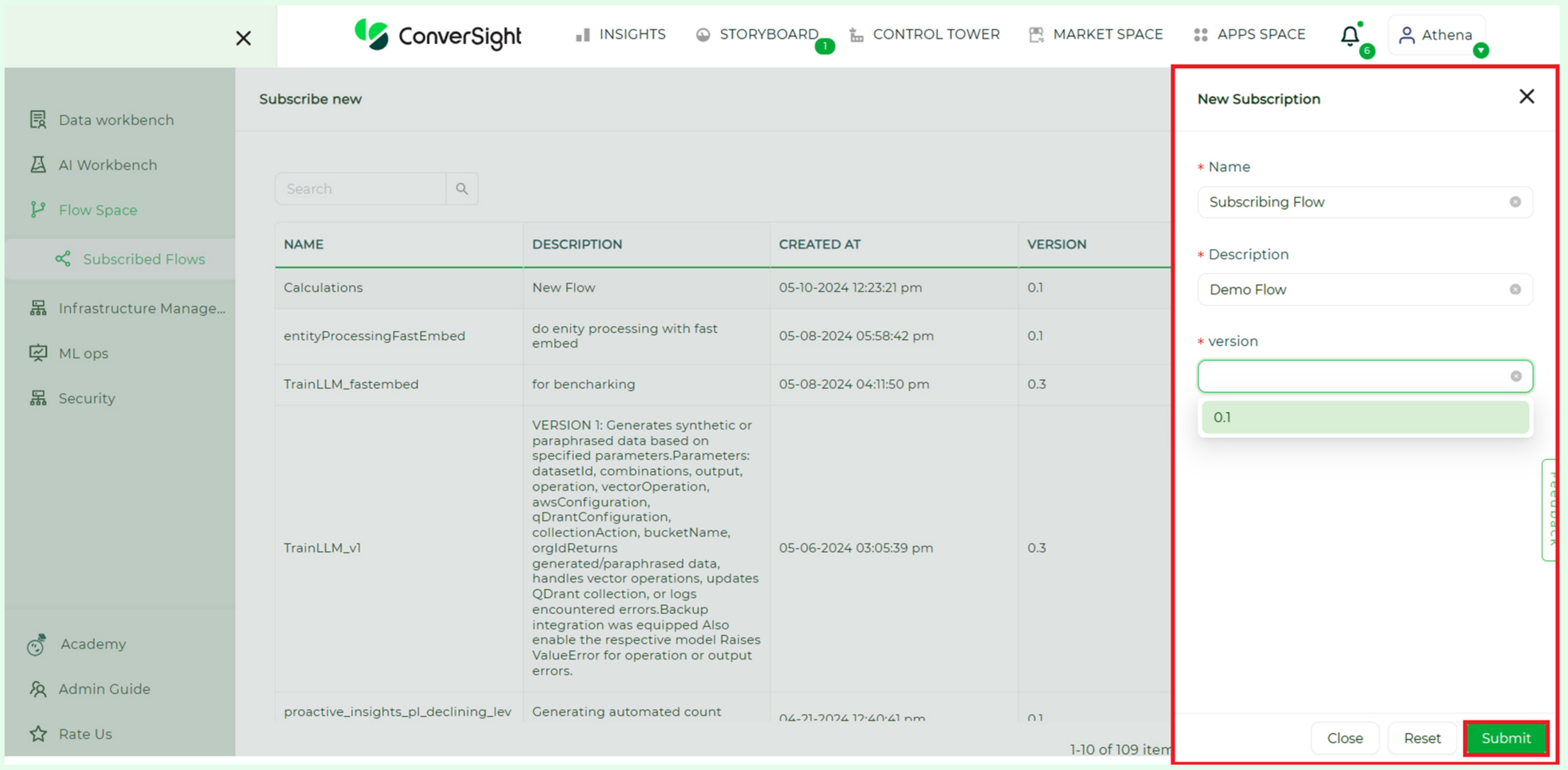




Subscribing New Flows

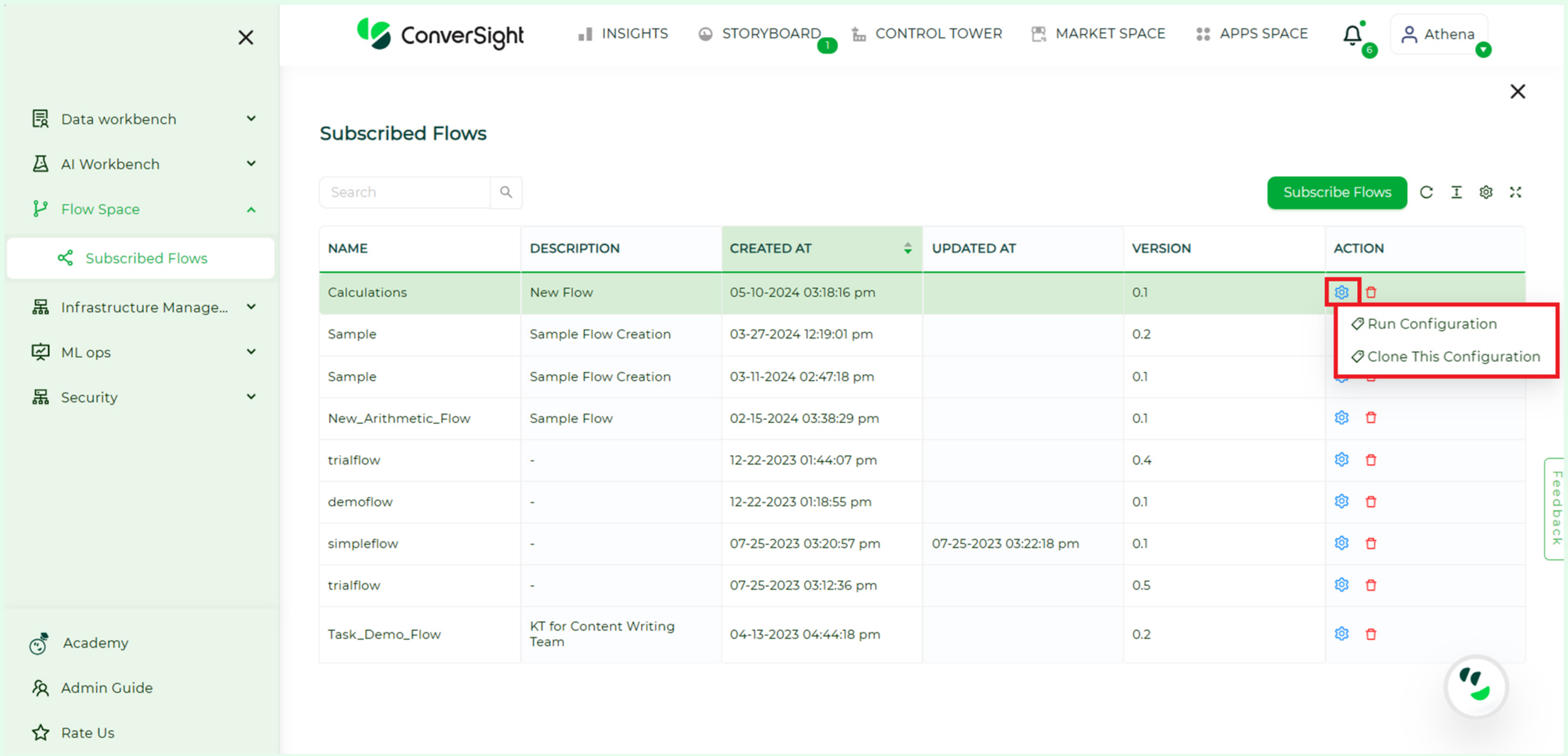
To initiate the subscription process for new flows, users can simply click on the **'Computer'** icon located within the Action menu. This action will lead them to a subsequent screen where they can conveniently provide the desired name, description, version and parameters for the flow they wish to subscribe to and click on **'Submit'**.



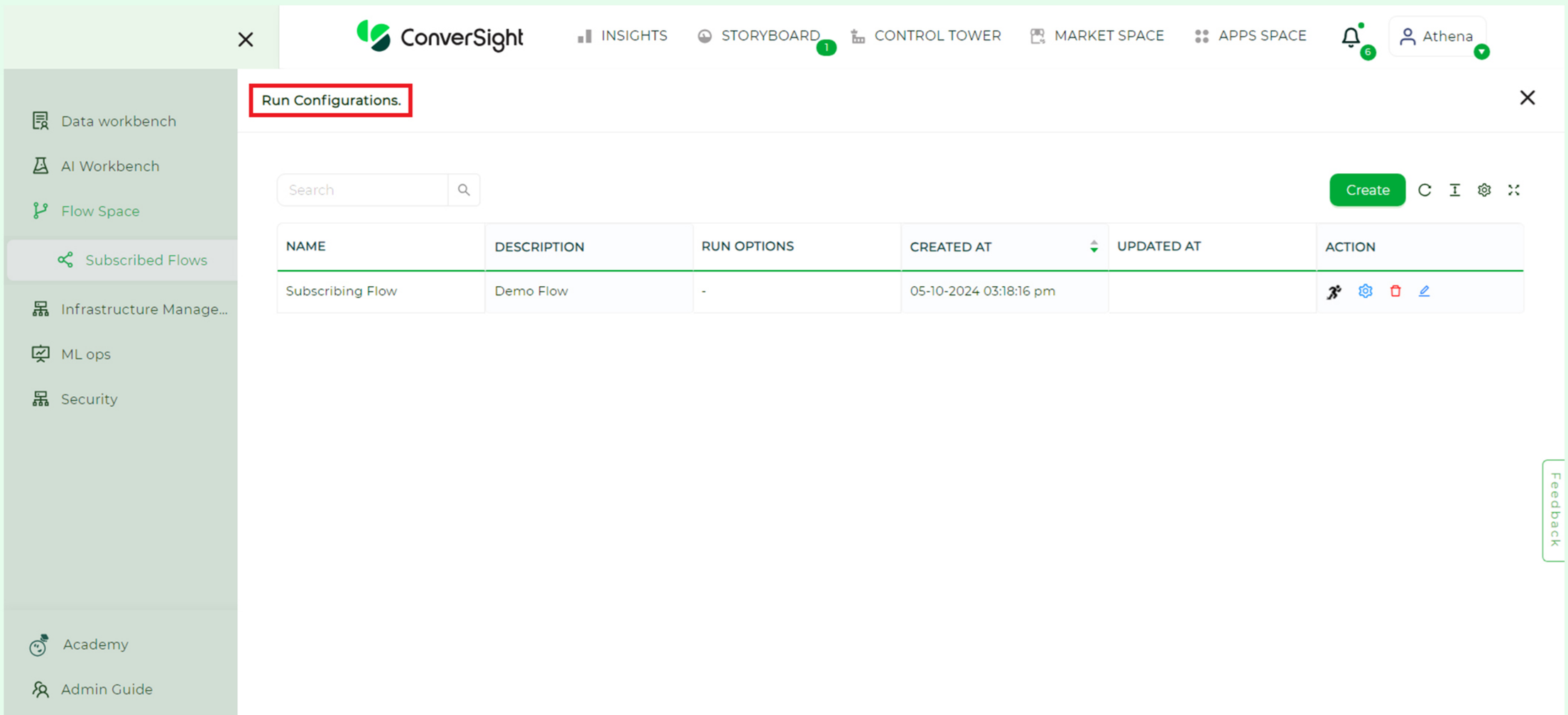


5.2 Run Configurations for Customizing Flows

The Run Configuration feature empowers users to execute Flows according to configurations tailored to their individual preferences. With the Run Configurations feature, users gain access to vital information about Flows, such as names, descriptions, run preferences, creation and modification dates. Additionally, users can take actions to modify parameters and initiate the execution settings of the Subscribed Flow.



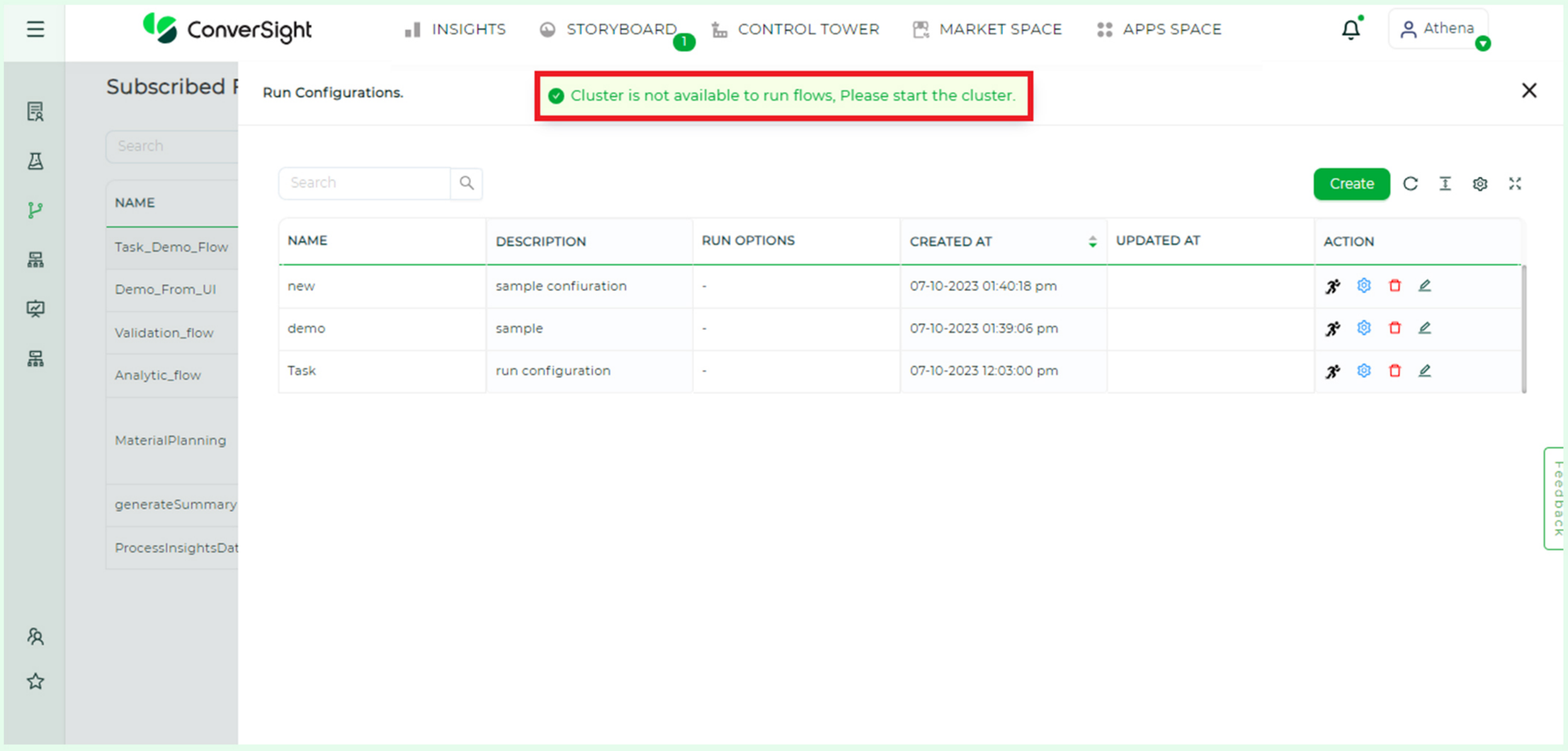
To run the configuration, users can simply click on the **'Settings'** icon within the Action menu of the desired Flow.



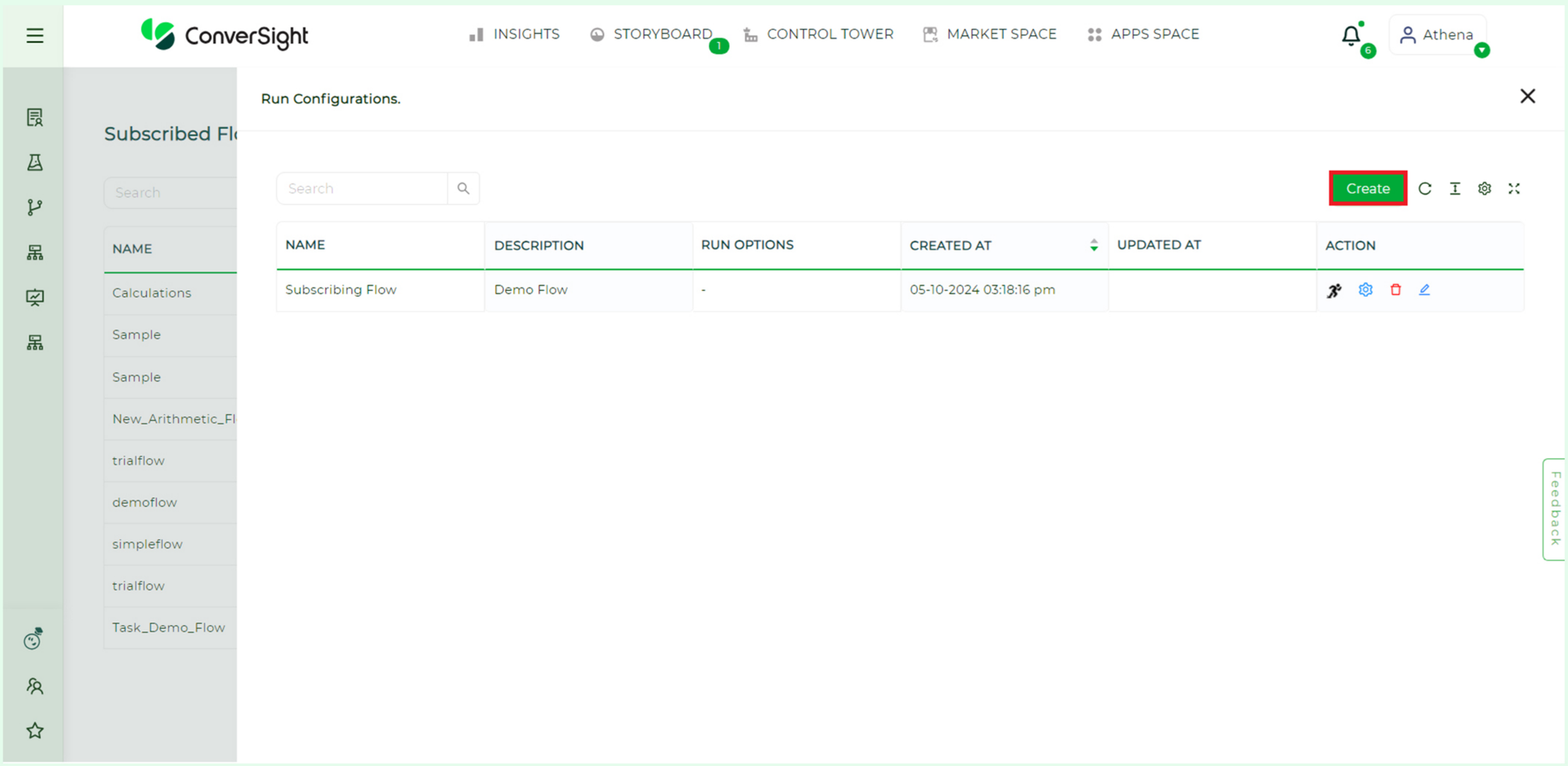
Note

To initiate the execution of a flow, users must first launch a cluster by selecting the ‘Start a Cluster’ icon located within the Compute Farm section of ConverSight.

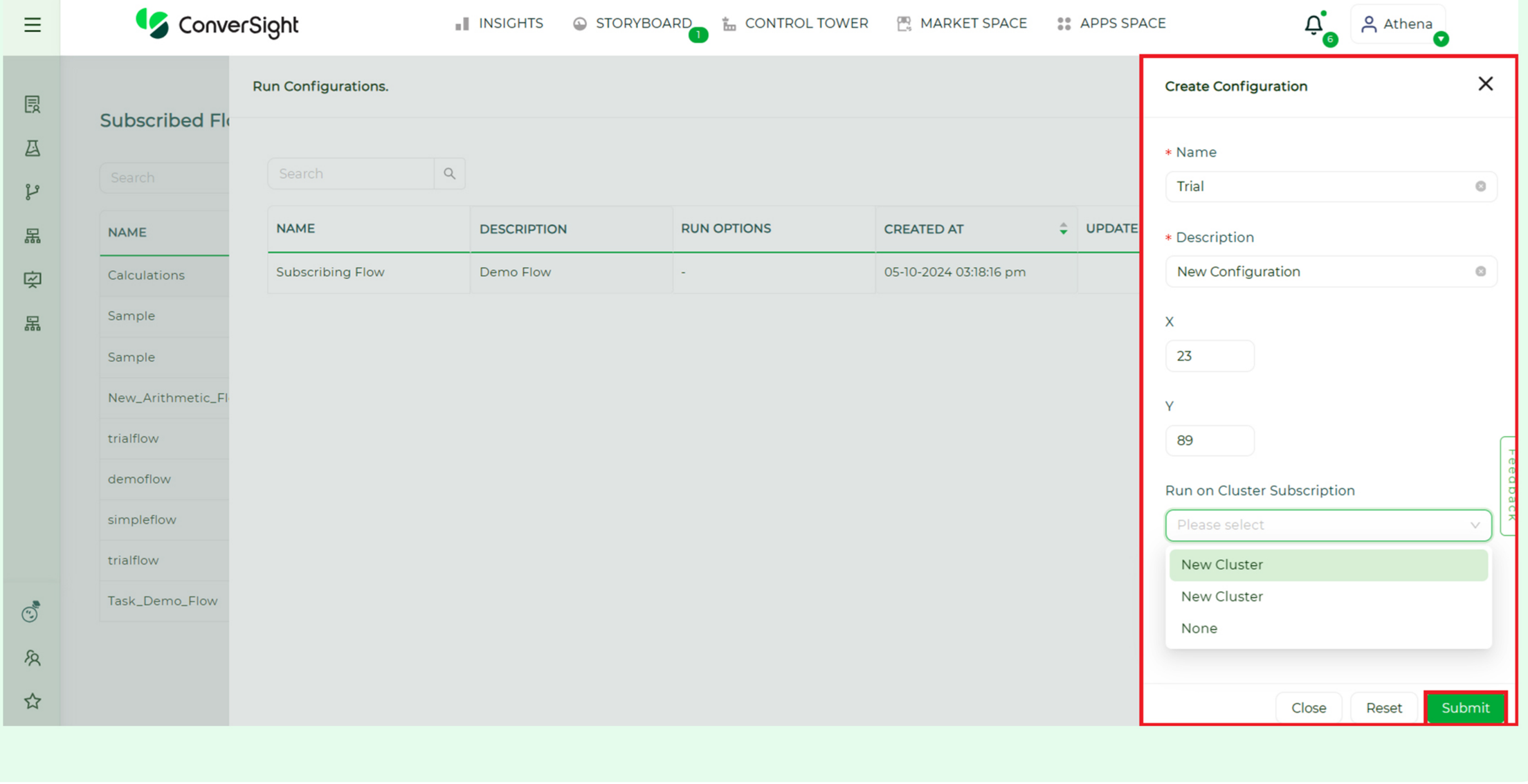
After initializing a cluster, users can simply click on the 'Run this configuration' icon found in the Actions menu. However, if the cluster has not been started, a popup notification will appear, informing users that the cluster is unavailable and prompting them to start it.



By clicking the **‘Create’** button in the run configuration menu, users have the ability to craft their own custom configurations. This empowers users to tailor settings according to their preferences and requirements.

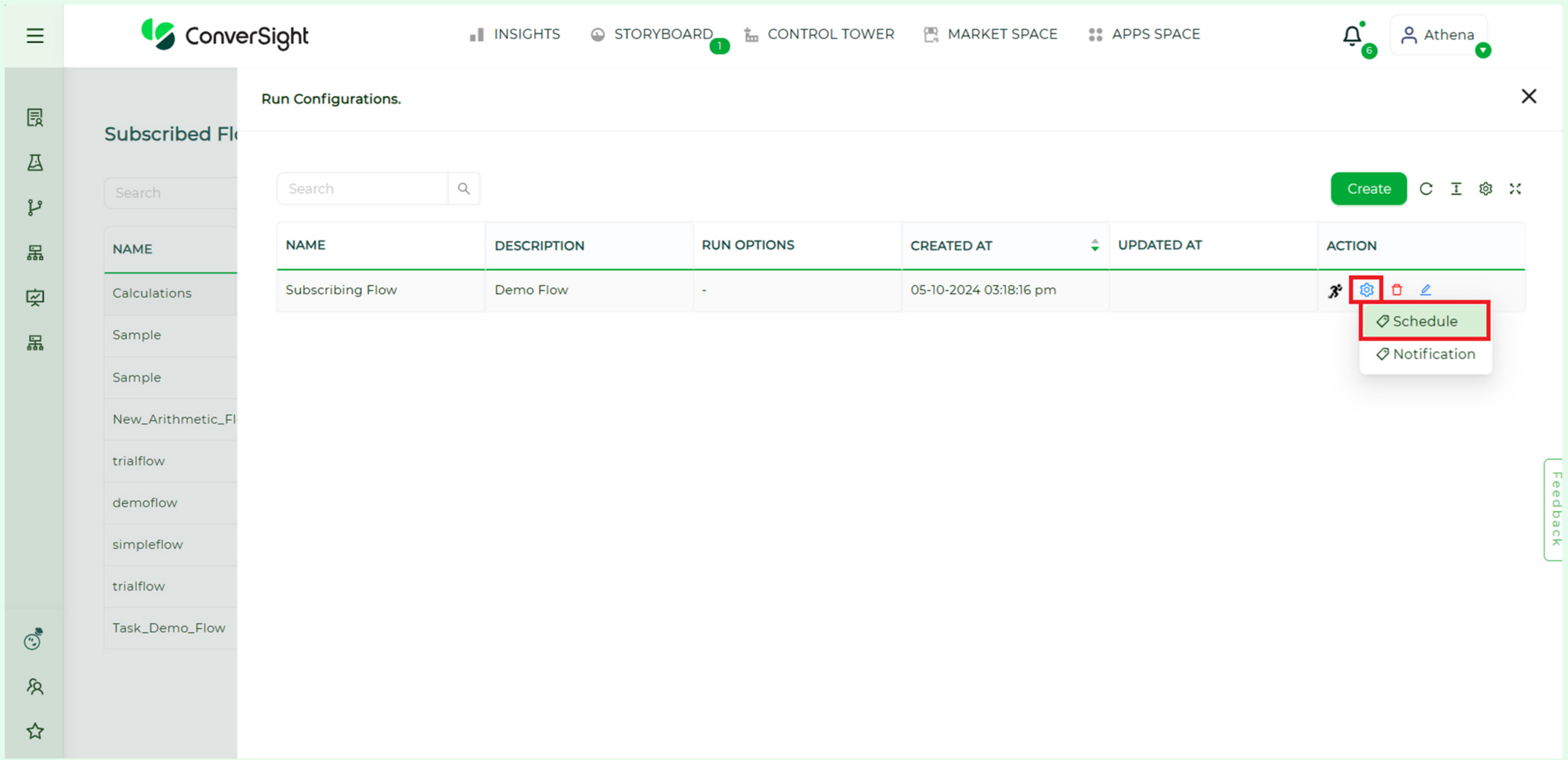


When you click the **'Create'** button, the subsequent screen will emerge, allowing users to input the flow's name, description and parameters. It is essential for users to then choose the suitable cluster from the options provided, determining the environment in which this configuration will operate and click on **'Submit'** to create the configuration.

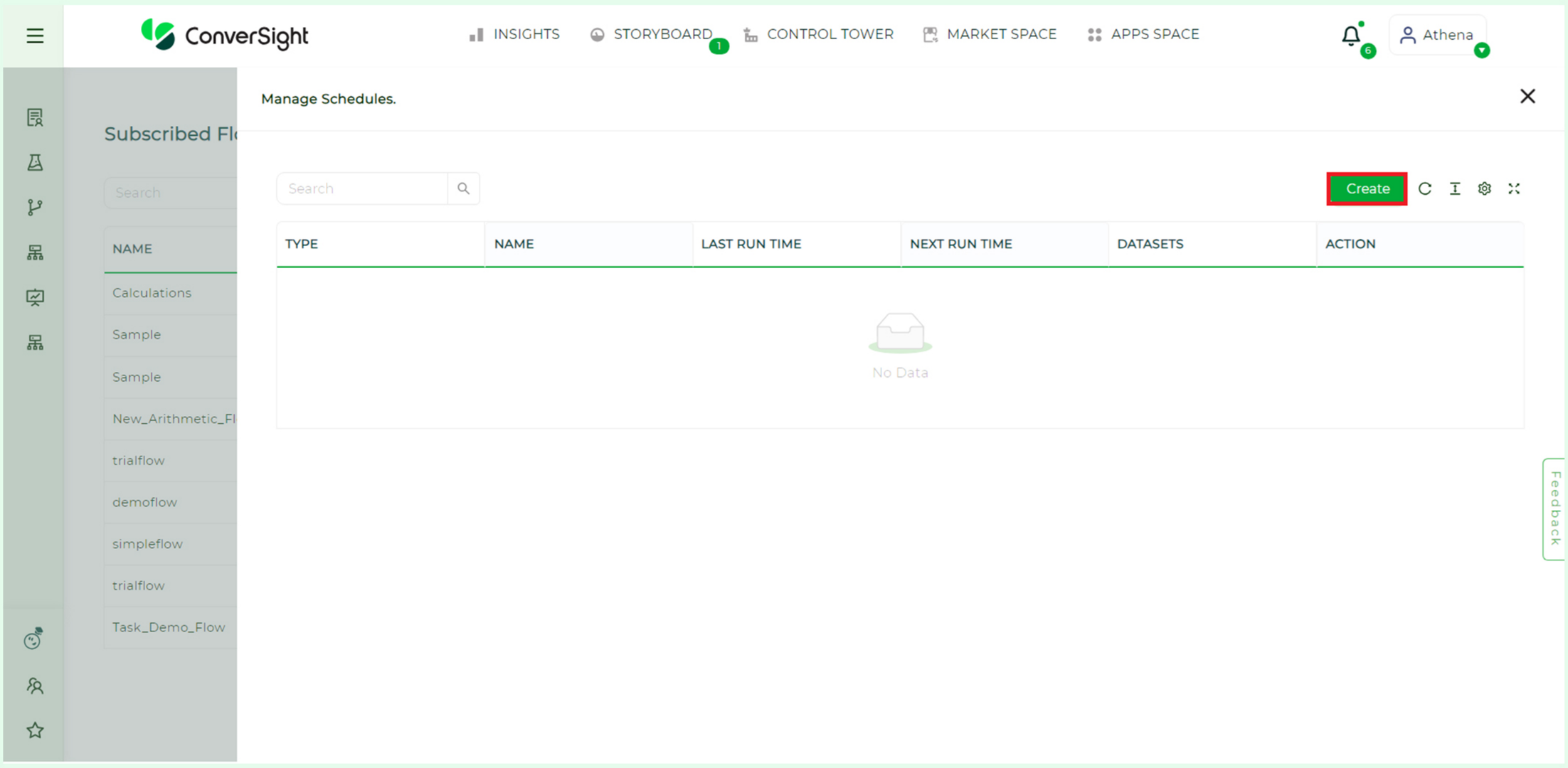


Schedule

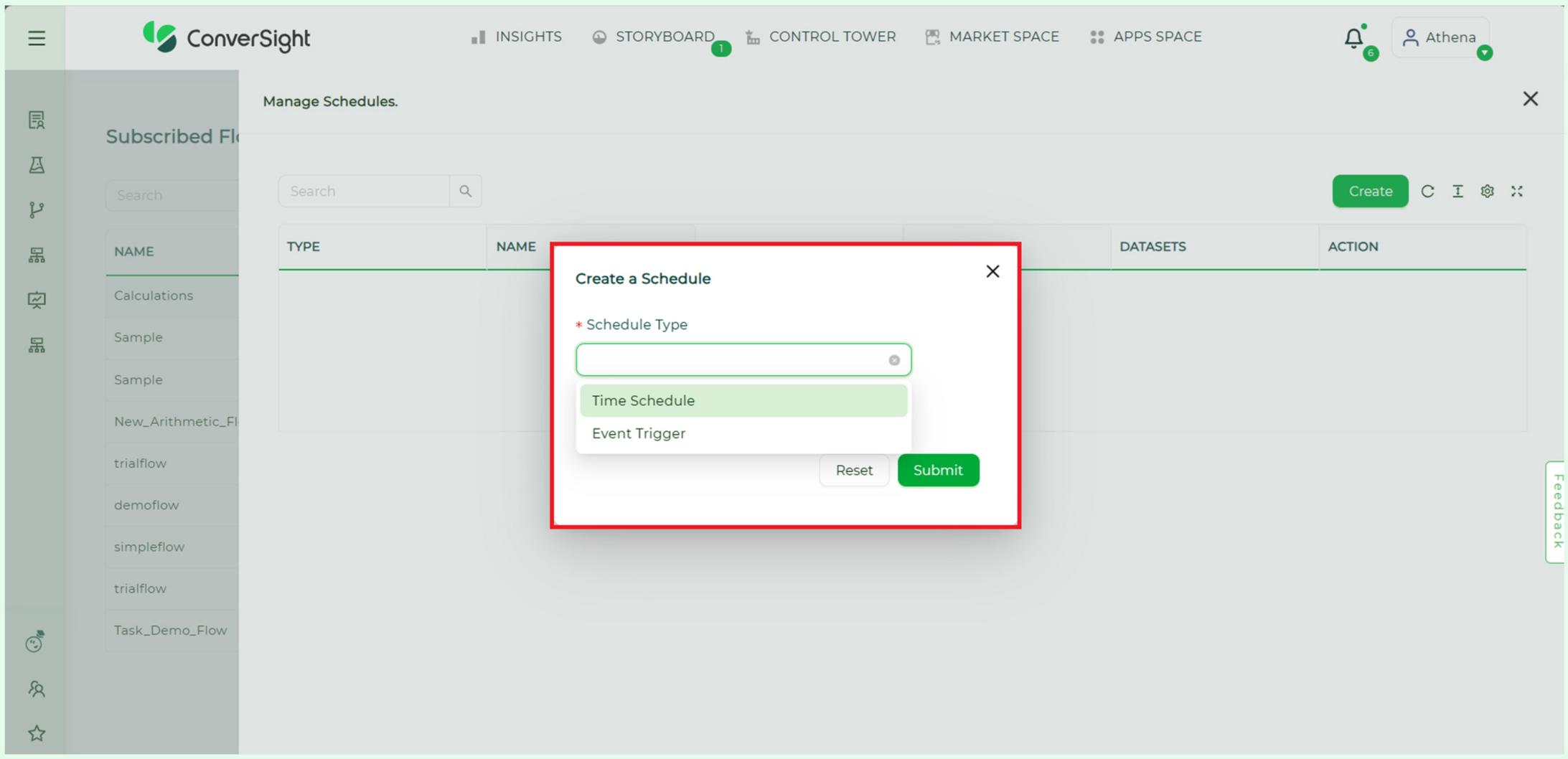
Users have the flexibility to schedule their preferred workflow to execute according to their requirements. They can choose between a time-based schedule or an event-triggered schedule for their convenience improving time optimization.
By clicking the **'Settings'** icon in the action menu, users can access the schedule option, allowing them to set up scheduled flows.



Users have the option to create a new scheduled flow by clicking on the **'Create'** button.

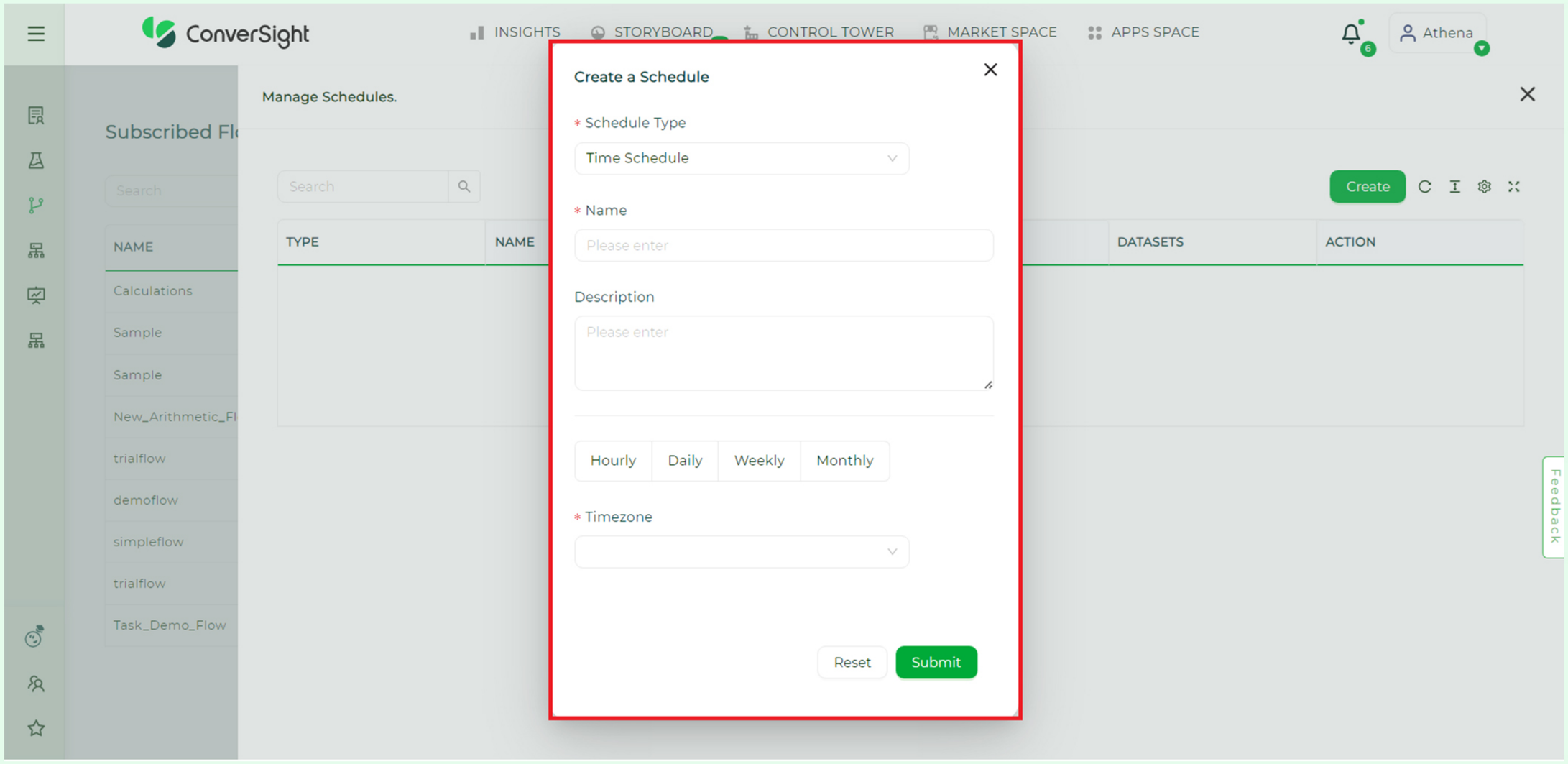


Users can choose the Scheduler type as per their need from the available types mentioned in the image.



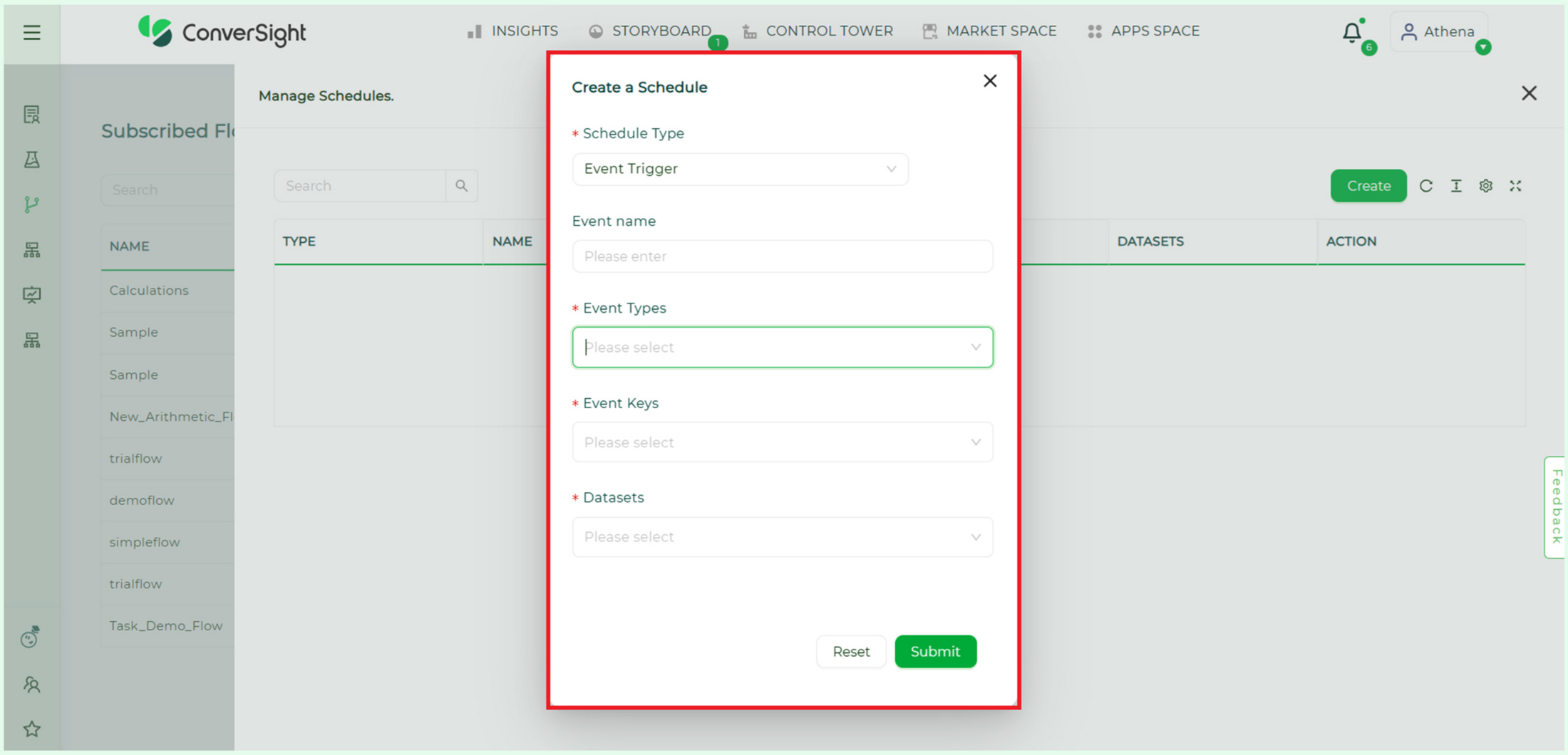
To set up a Time Schedule, provide the following and click on Submit to create the Time Schedule.

ARGUMENT	DESCRIPTION
Name	The Name of a schedule is a brief yet descriptive title that helps to identify the schedule.
Description	A Description provides clarity and guides others to a better understanding. It includes detailed information about what the schedule entails, its objectives and any other relevant details that could help others understand it better.
Hourly / Daily / Weekly / Monthly	Users can choose how frequently their Flows should be executed. **Hourly** – Executes the Flow based on intervals set for hours. **Daily** – The Flow will be executed everyday. **Weekly** – The Flow will be executed once a week.



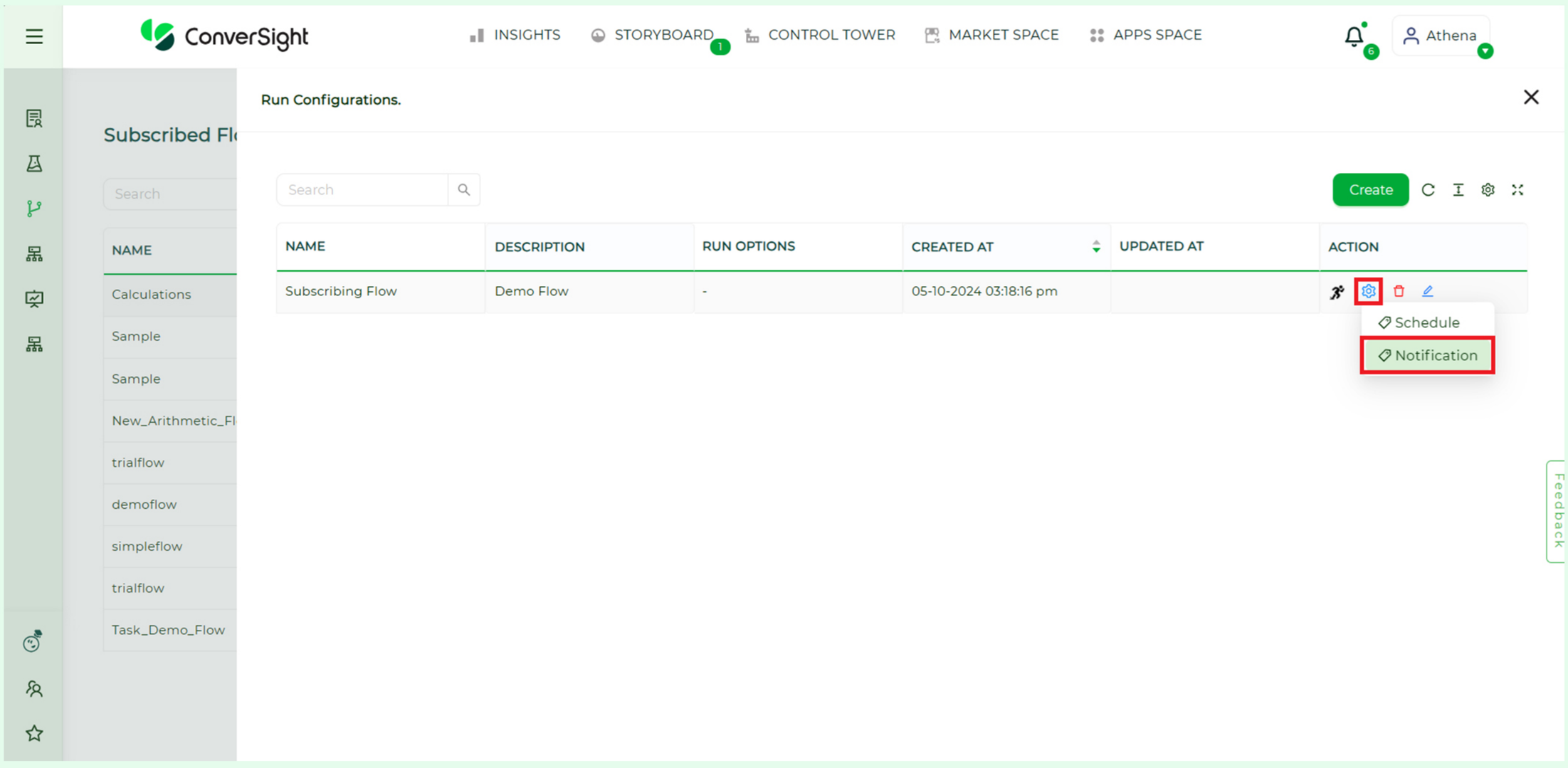
To establish an Event Trigger Schedule, provide the following details and click on **'Submit'** to create the Event Trigger Schedule.

ARGUMENT	DESCRIPTION
Event Name	The Name of an Event Schedule is a brief yet descriptive title that helps to identify the schedule.
Event Type	To choose to what scenario the schedule must occur. Users can choose the Event Type based on their specific requirements. **Dataload** – Creates a schedule when the selected Dataset is loaded. **Republish** – Creates a schedule when the selected Dataset is republished.
Event Keys	Datasets involved in this schedule.
Dataset	Users can choose the desired Datasets they want to schedule from the dropdown menu.

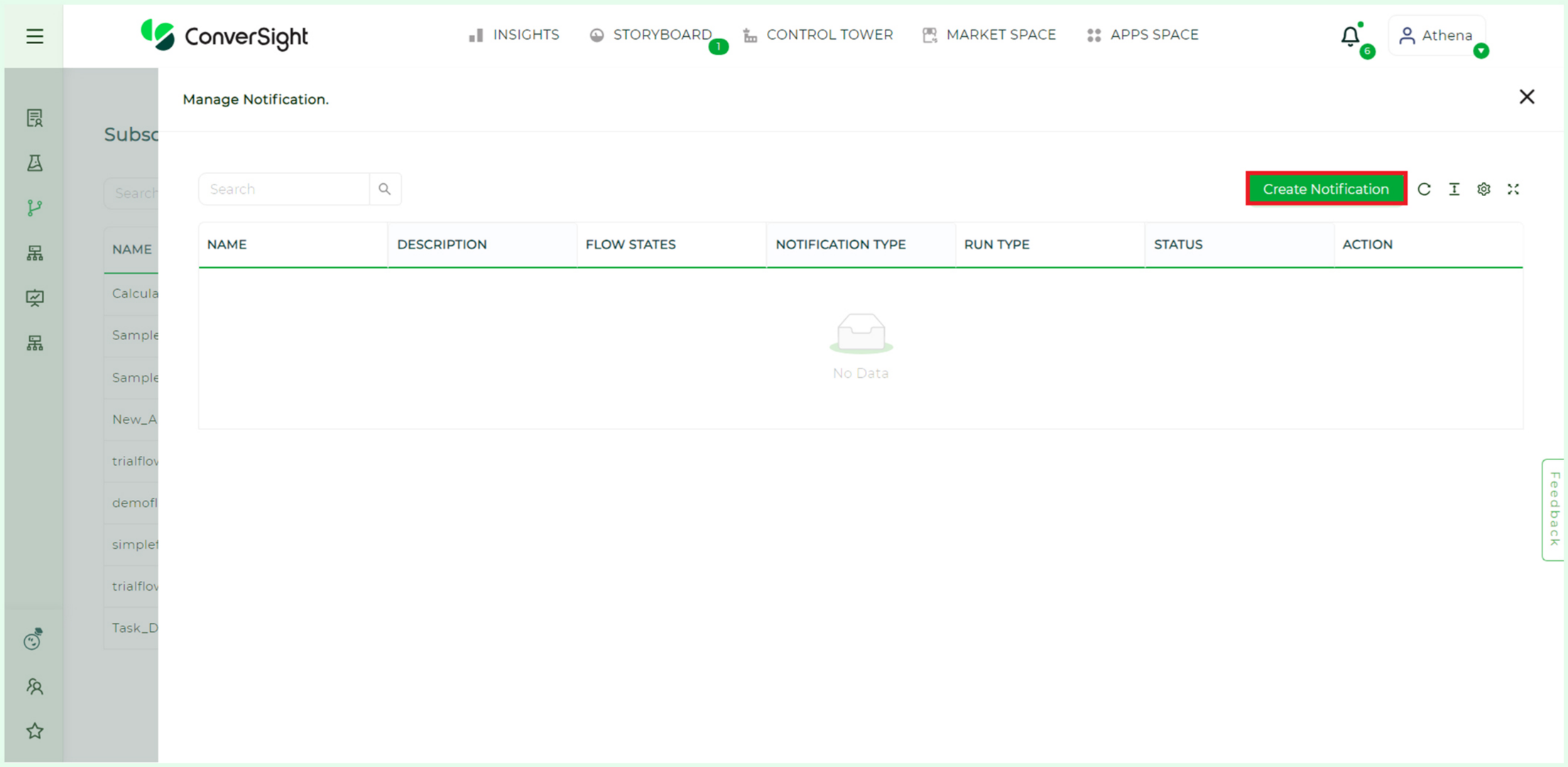


Notifications

Users can generate personalized notifications to stay well-informed about their flow's status, aligning perfectly with their unique needs. This feature plays a pivotal role in enhancing flow management and optimizing time utilization. By clicking the **'Settings'** icon in the Action menu, users can access the Notification option, allowing them to set up notifications for flows.

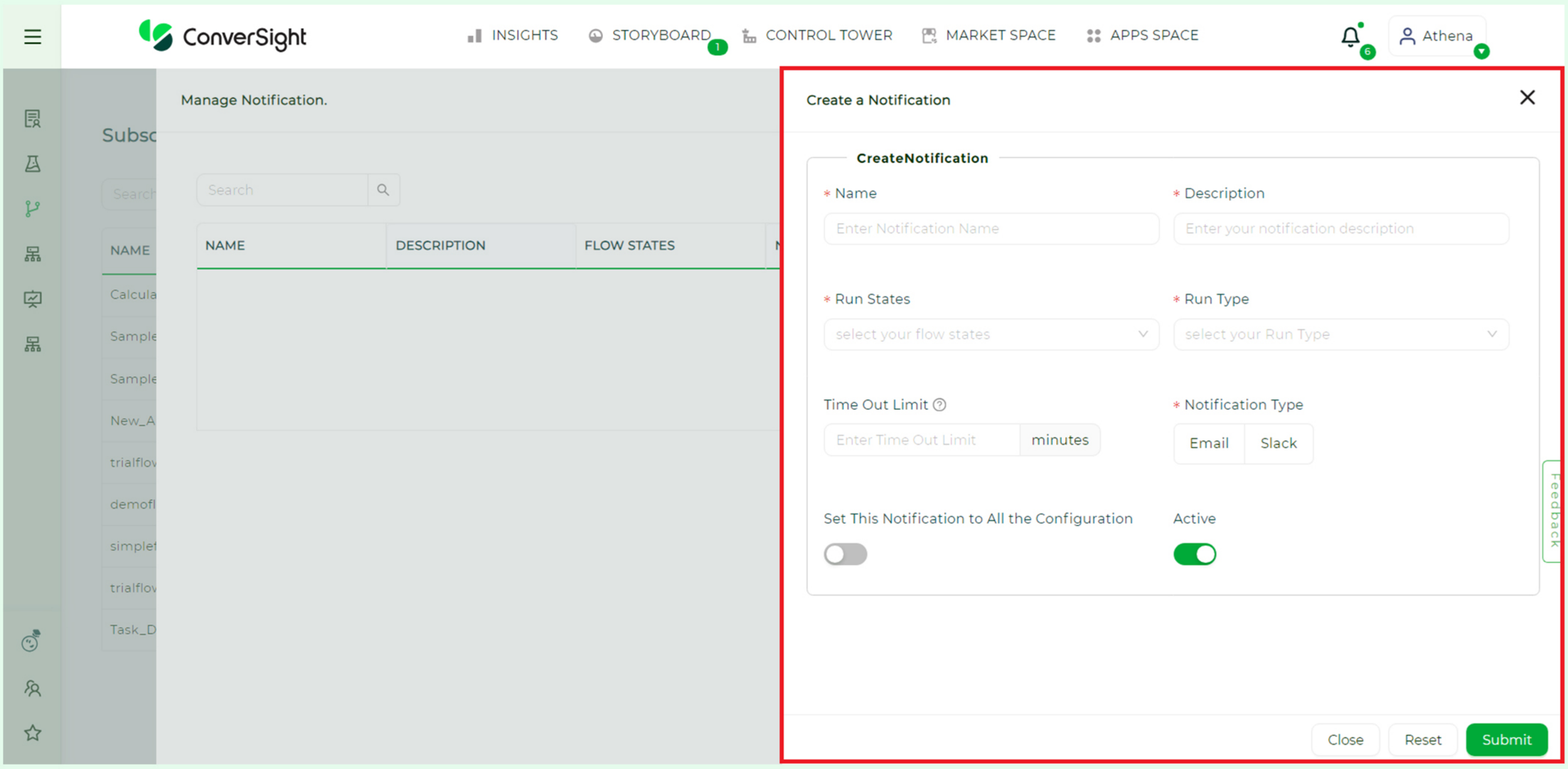


Users have the option to create a new notification by clicking on the **'Create Notification'** button.



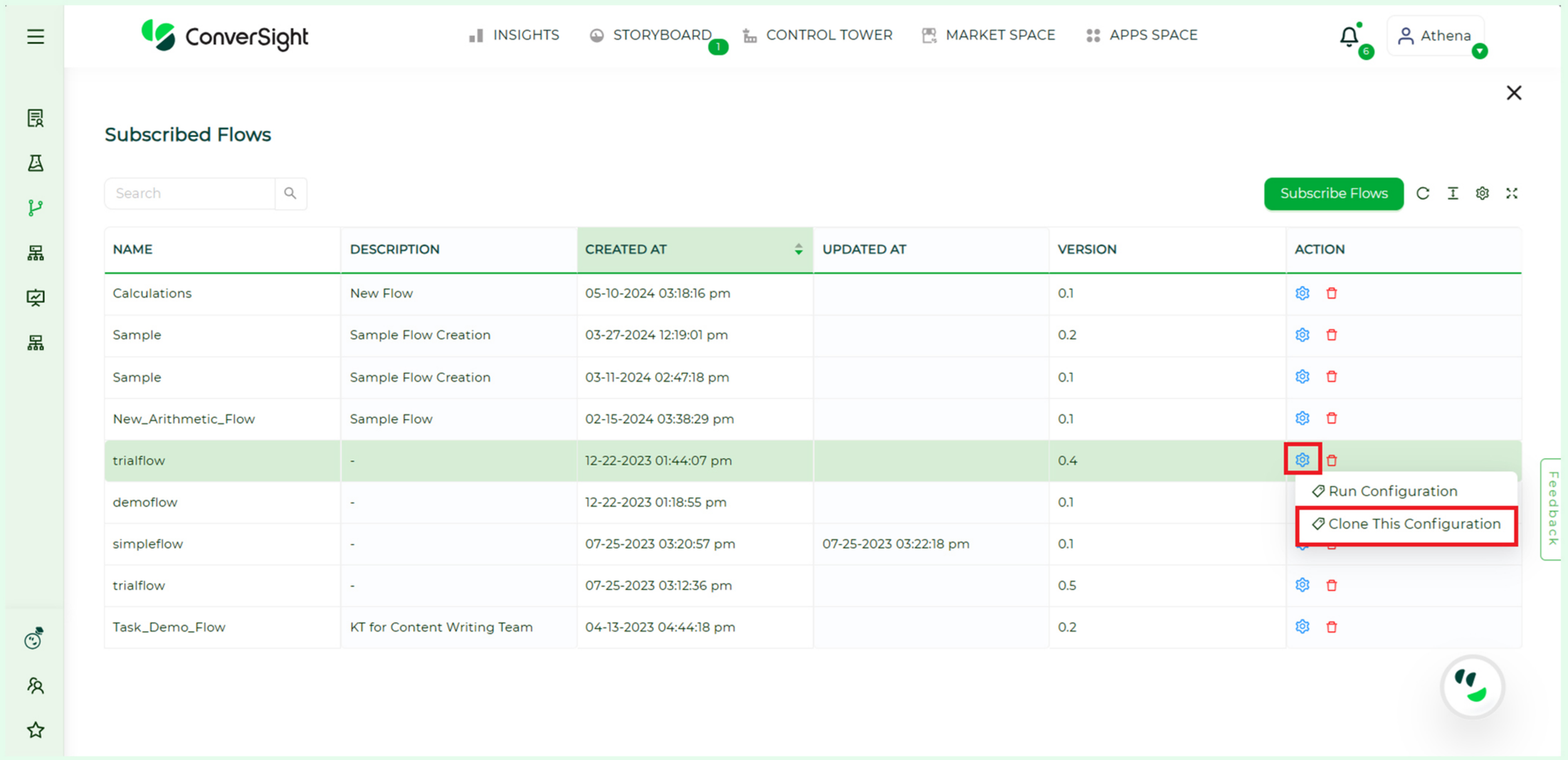
Users have to provide the following and finally click on **'Submit'** to create the Notification.

ARGUMENT	DESCRIPTION
Name	The Name of the Notification is a brief yet descriptive title that helps to identify the notification.
Description	A Description provides clarity and guides others to a better understanding.
Run States	The status of the Flow chosen must be given here like Completed, Crashed, Deployed, Failed, Running.
Run Types	Describes how you want to run the Notification, like through API, OnDemand, Time and Trigger schedule.
Time Out Limit	Indicates the time out limit to trigger the notification for the flow run status.
Notification Type	Choose the preferred type of notification whether Email or Slack.
Set This Notification to All the Configuration	Activating this toggle permits users to set this notification for all flow configurations.
Active	This toggle Activates the Notification for the flow.



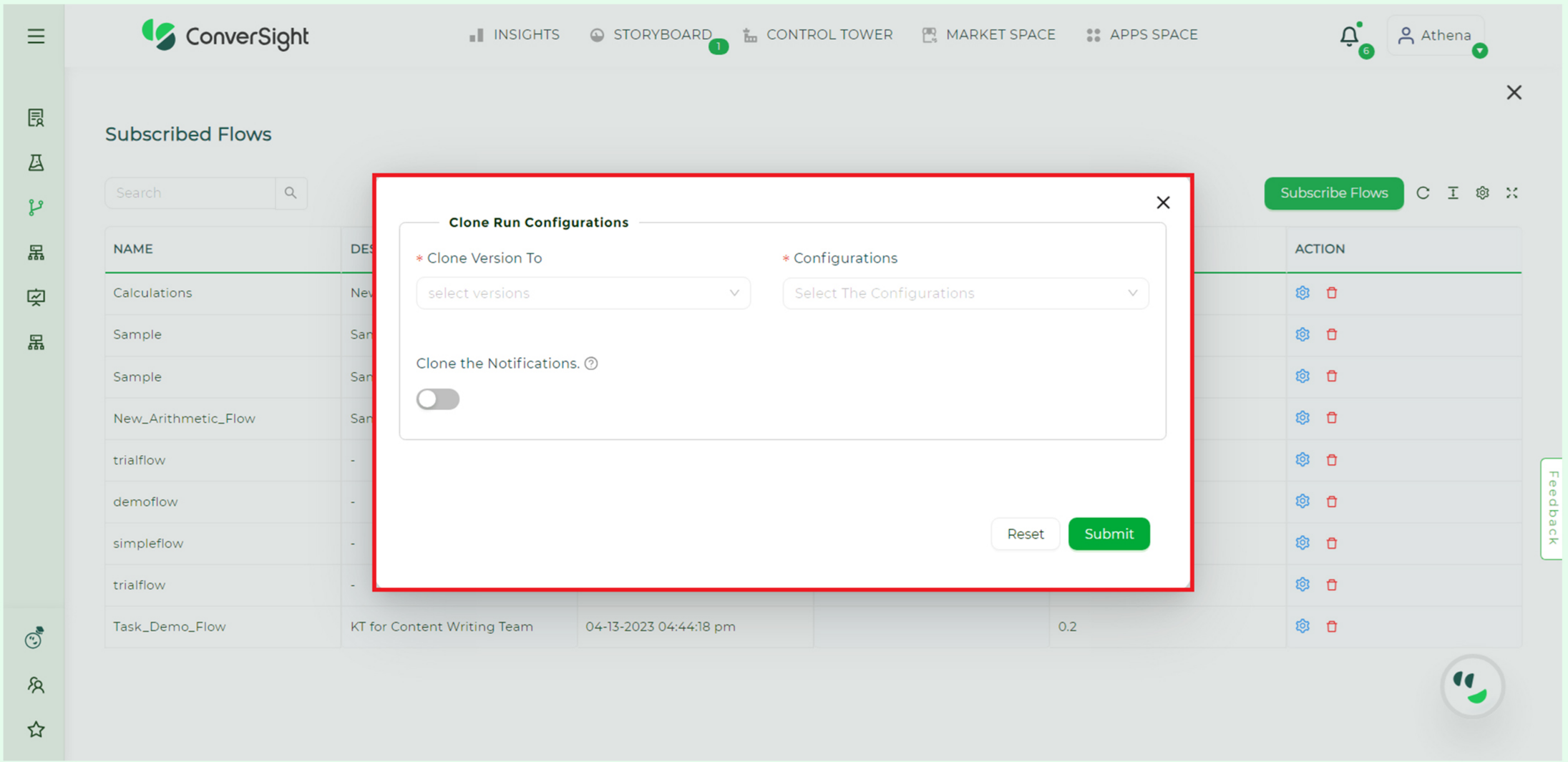
5.3 Clone This Configuration

The **'Clone This Configuration'** feature is to duplicate configurations from one version of the flow to another. Users can access the Clone This Configuration feature by clicking on the **'Settings'** icon found within the Action field.

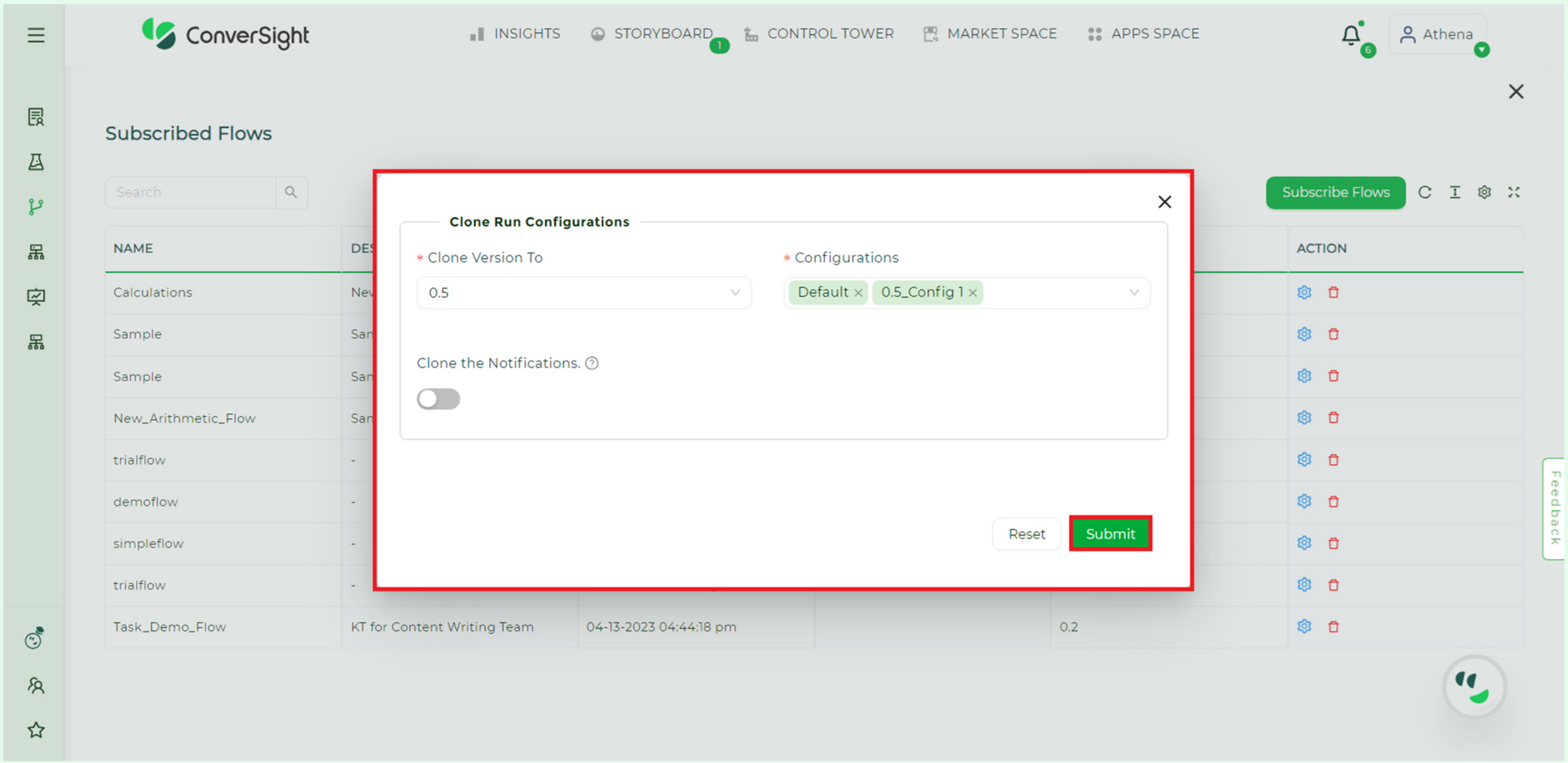


The following inputs are required to Clone a Configuration.

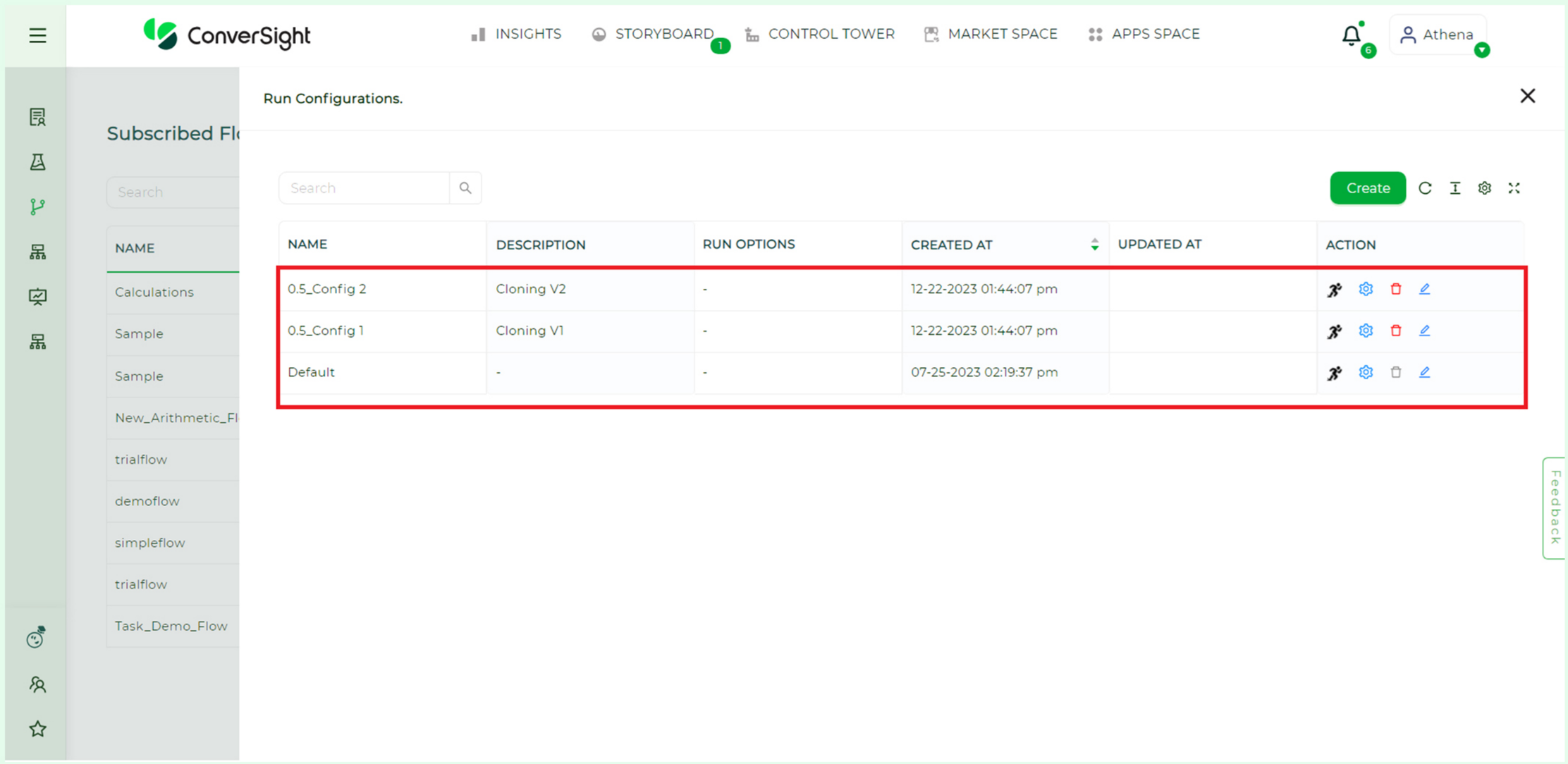
ARGUMENT	DESCRIPTION
Clone Version To	Users can choose the specific version of the flow to which they want the configurations cloned.
Configurations	It shows the configurations that are accessible within the chosen flow.
Clone the Notifications	This toggle allows users to clone the Notifications of the created configurations.



Once all the required details have been entered, the user can click on the **'Submit'** button to add the newly created configuration.



To ensure successful configuration cloning, users can select the target flow, click **'Run Configuration'** and verify all configurations, including clones, on the Run Configuration page.



6. Conclusion

In this whitepaper, we explored the Task and Flow creation, execution and elevation, providing users with the tools to effortlessly build scalable workflows. Through detailed instructions, users can craft streamlined data processing pipelines that encourage collaboration and reusability. The versatility of ConverSight AI Workbench's Tasks and Flows positions it as the ultimate hub for formal data analysis tasks, unlocking the full potential of data-driven endeavors. Embrace efficiency, foster collaboration and adopt ConverSight AI Workbench for an unmatched data processing journey.

Join our customers who have accelerated growth with ConverSight

DAIMLER | Kokon USA | CEAT

Dr. Agarwals EYE HOSPITAL | gimme SEAWEED | EXTRACT

THERMO BOND BUILDINGS | ATD AMERICAN TIRE DISTRIBUTORS | ASHOK LEYLAND

Mahindra | [MAVPAK] | salesbricks

CHAMPIONX | motherson | ISTHARA

ZOOX | meditrend | LEADER PROMOS

About ConverSight

ConverSight's Adaptive Analytics platform uses conversational AI, Natural Language Processing and machine learning to converge the distance between humans and data through data stories, presenting the meaning of data in the most effective, personalized and efficient form possible. ConverSight's patented AI business assistant, Athena, connects distributed databases to answer questions and Augment the consumers through 4 key functions: Information on demand, Automated Story Telling, Proactive Insights, and Recommended Actions.

For more information, visit www.conversight.ai

Gartner peerinsights™ | Users Love Us