Documentation of Semaphore

**Created by : Sonu kumar**

**Keen & Able Computers Pvt Ltd**

**Table of Contents**

[**What is Semaphore**](#_lnkoeineuvl4) **3**

[Support Database:](#_bc0oy7ecuciu) 3

[Used for:](#_7bvhv9m3b691) 3

[User Guide:](#_jrs0lxd0fhcx) 3

[**Setup Semaphore through docker compose file**](#_plso7f3chwu8) **4**

[Details of the User Guide:](#_syx1m4szumqi) 7

[Key Store:](#_hl62myuq1y8w) 7

[Key store with SSH:](#_w990ywj51p14) 7

[Login with password:](#_a2kkorpe7277) 7

[None](#_xwjdutfs0ek6) 7

[Repositories](#_6ofna4zarcva) 8

[Environment](#_uljb8963s37x) 8

[Inventory](#_bhsa6jc090kr) 8

[Task Template](#_np89hm1ixfja) 8

[Run playbook through semaphore UI](#_9r5jghbmejpe) 8

[Run simple “Hello World” playbook through semaphore UI](#_y08cse2cvnwl) 8

# **What is Semaphore**

It is the modern UI for Ansible. It helps in easily running the Ansible Playbook, getting notification about failures, and controlling access to the deployment system.

* It is written in pure Golang.
* It is available for Windows, Mac OS, Linux.
* It is responsible for the Web UI for Ansible Playbook.

## **Support Database:**

* MySQL
* Postgres
* BoltDB

## **Used for:**

* Semaphore is used for build, deploy, and rollback.
* It manages the Environment, Inventory, Repository and Access Key.
* It is used to run the playbook from the browser.
* It also runs the playbook by schedule.
* View details logs of any playbook run at any time.
* Getting the notification about the playbooks runs.

## **User Guide:**

* Key Store
* Repository
* Environment
* Inventory
* Task
* Task Template

We'll review the User Guide in detail, providing practical and precise instructions while organising the content.

# **Setup Semaphore through docker compose file**

---------------------------------------------------------------------------------------------------------------------------

I have follow these step to setup and user creation in semaphore with MySQL database are listed below:

**Step 1:** In the linux terminal firstly I have created the Semaphore directory.

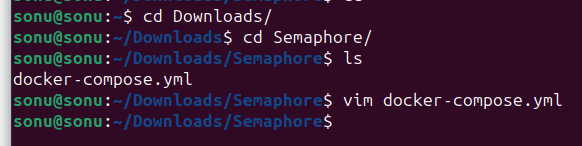
**Command Used: #** mkdir Semaphore

**Step 2:** Go inside that folder

**Command Used: #** cd Semaphore

**Step 3:** Create the **docker-compose.yml** file using **Vim** command.

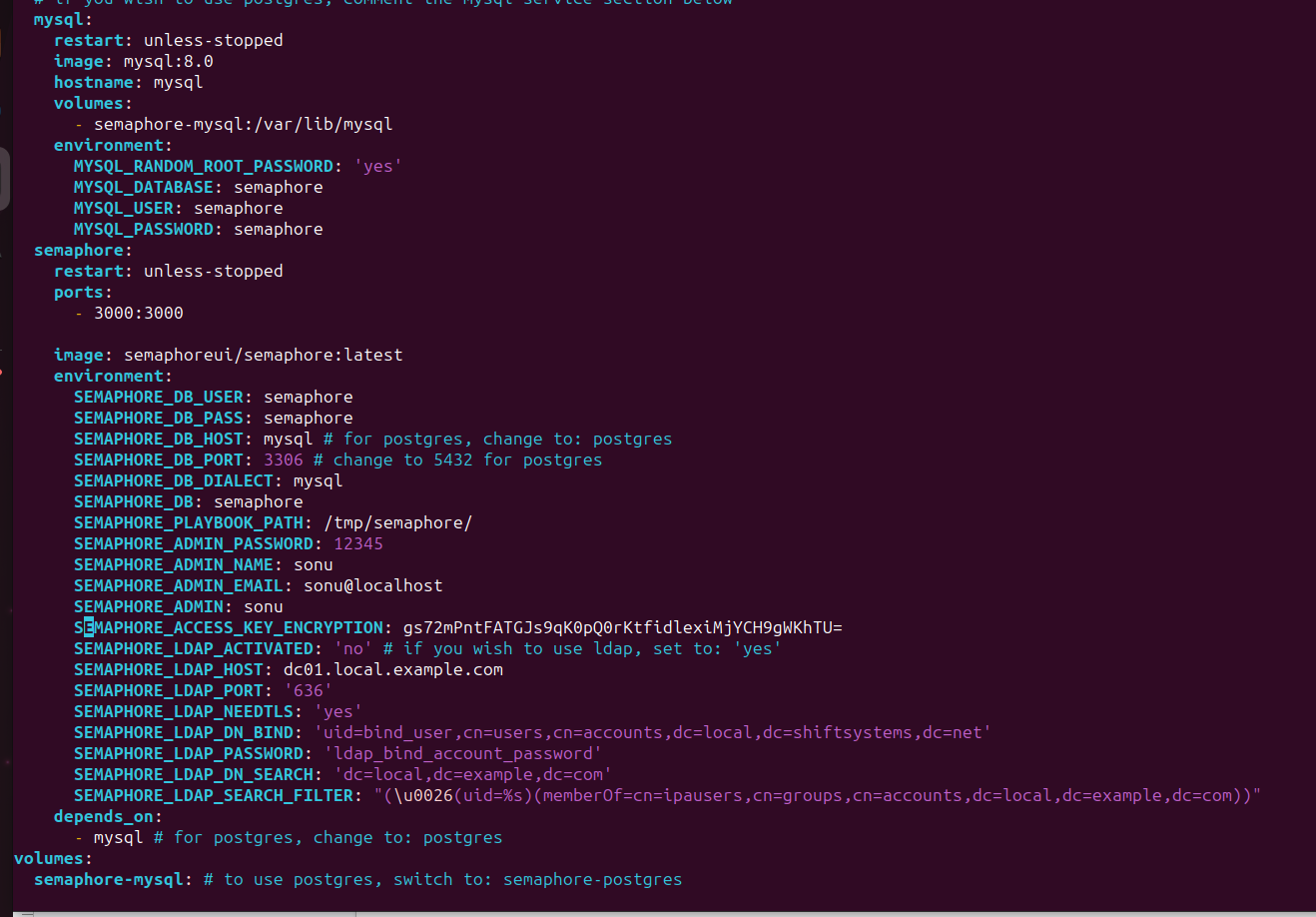
**Command Used: #** vim docker-compose.yml



**Step 4:** Then write the following scripts in the docker-compose file.

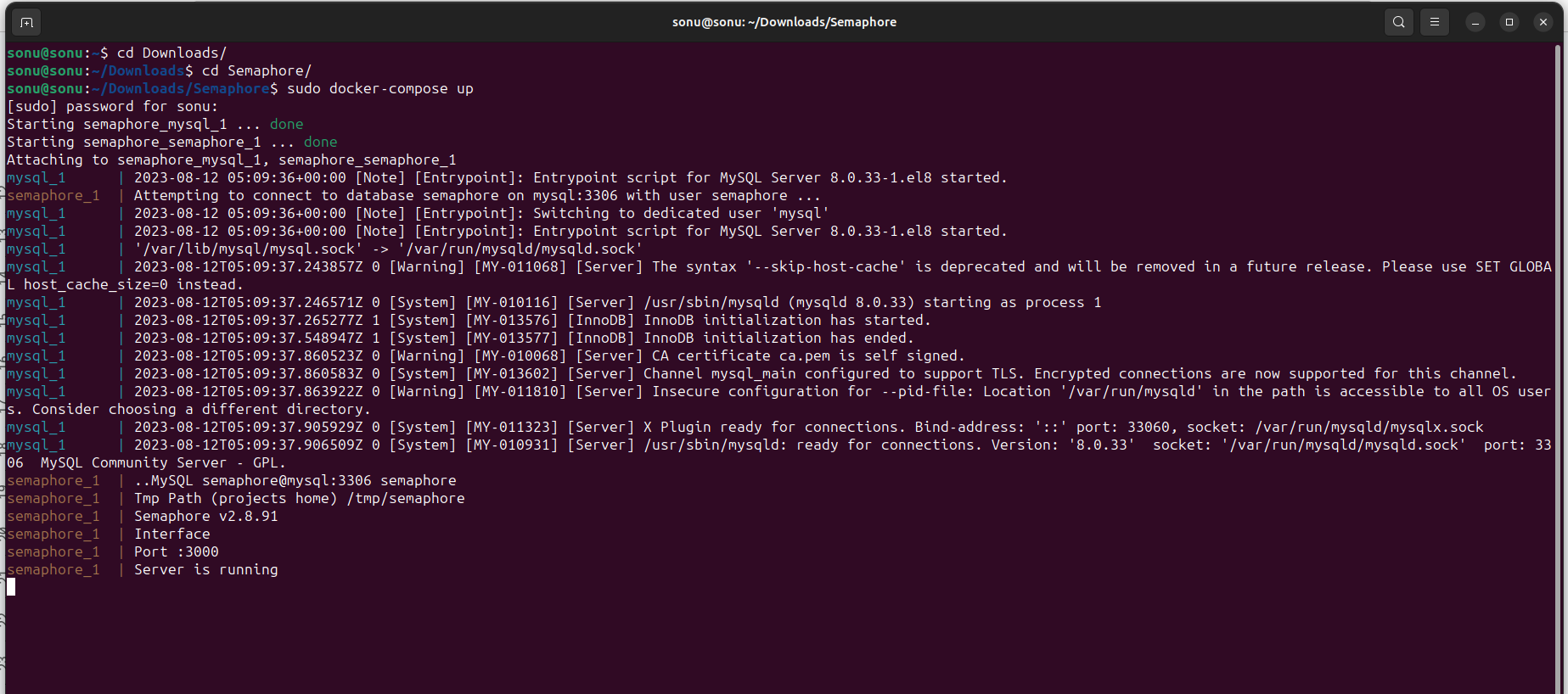
**Script:**

**Step 5:** For creation of the semaphore admin user I have changed the semaphore Username, Admin, Password as I want which is already added in this below script.

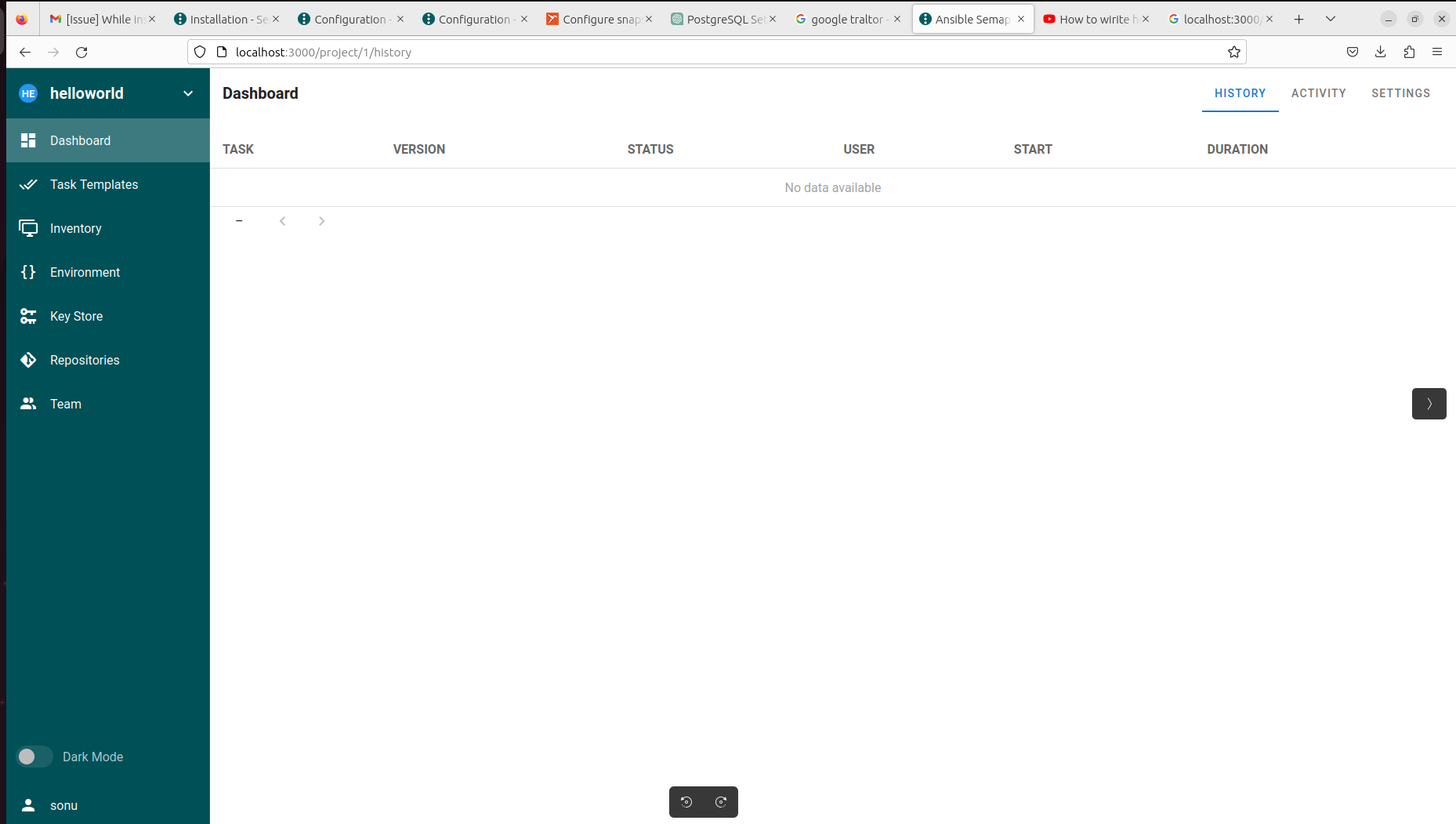


**Step 6:** Then I execute the docker compose file. Then our server starts in running mode. Whose screenshots are attached below:

**Command used: #** docker-compose up



**Step 7:** Finally goto the browser and type localhost and the port of the semaphore 3000 and press enter button. Then the semaphore UI tools open on the browser. Whose screenshots are attached below:



As we see in the screenshot the following User Guide attributes are as shown in the figure.

## **Details of the User Guide:**

* Key Store
* Repositories
* Environment
* Inventory
* Task Template

### **Key Store:**

It is used to store the credentials for accessing remote repositories, accessing the remote hosts, sudo credentials and Ansible vaults password.

It helps in the configuration of all required access keys.

In this we have create the key in 3 ways:

* With SSH
* Login with Password
* None

#### **Key store with SSH:**

It accesses the remote server as well as remote repositories.

In SSH semaphore doesn't support the SSH key that is password protected.

For the “Git Repositories” we use the SSH authentication, for cloning the “git repo” we need the public key associated with the private key.

#### **Login with password:**

It means the Username & Password for the access token combination.

* In this it authenticates to remote hosts (less secure authentication).
* Sudo credentials on remote hosts.
* Authenticate to remote “Git Repositories” over HTTPS (although SSH is not secure).
* It unlocks the ansible vaults.

#### **None**

It is like the filler of the repositories that do not require any authentication.

### **Repositories**

Repositories are the place where key store and manage the content like playbook and roles.

All tasks must be required by the repositories in order to run the playbook.

### **Environment**

In this where we have to store the additional variable for an inventory and must be stored in a JSON file.

### **Inventory**

This is the file where it contains all the lists of Ansible hosts. It also stores the variable that can be used in the playbook. It stores the file in JSON, YAML, or TOML.

### **Task Template**

It define how we run the playbook with the parameters:

* Inventory Name
* Playbook file name
* Inventory
* Environment
* Playbook repositories
* Vault password file
* Extra CLI argument.

In this we have test the task in three different ways:

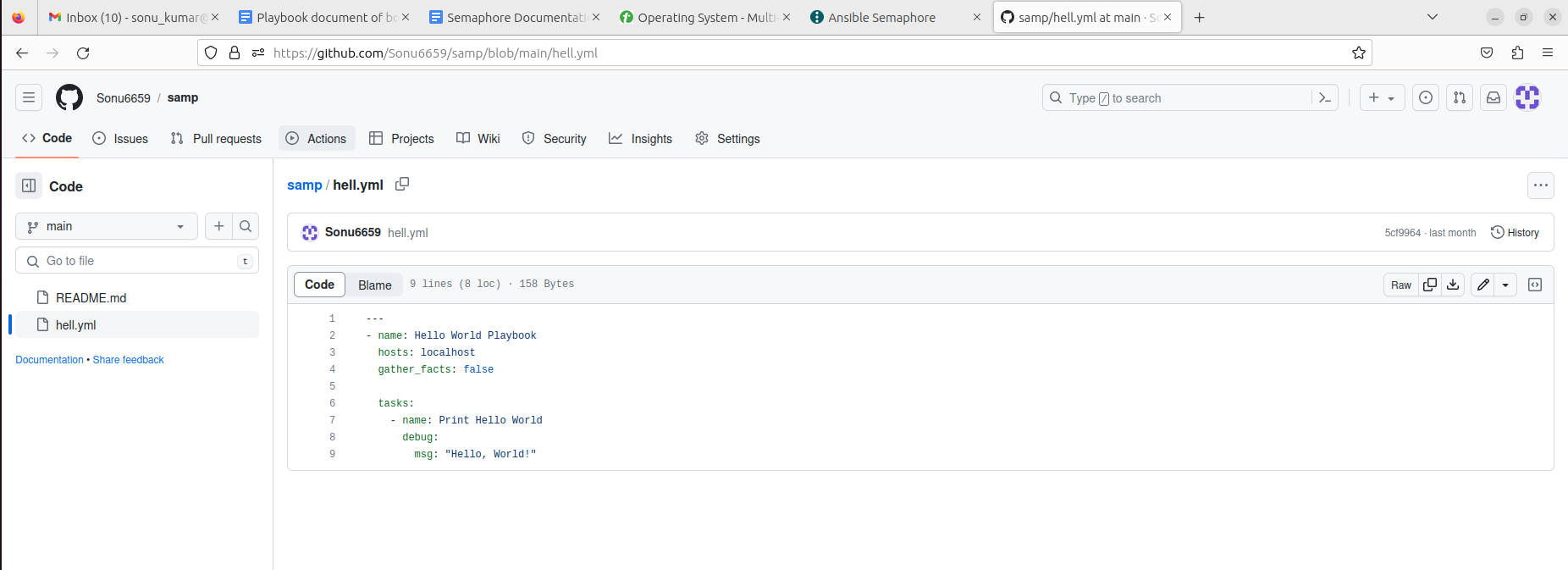
* Task
* Build
* Deploy

## **Run playbook through semaphore UI**

### **Run simple “Hello World” playbook through semaphore UI**

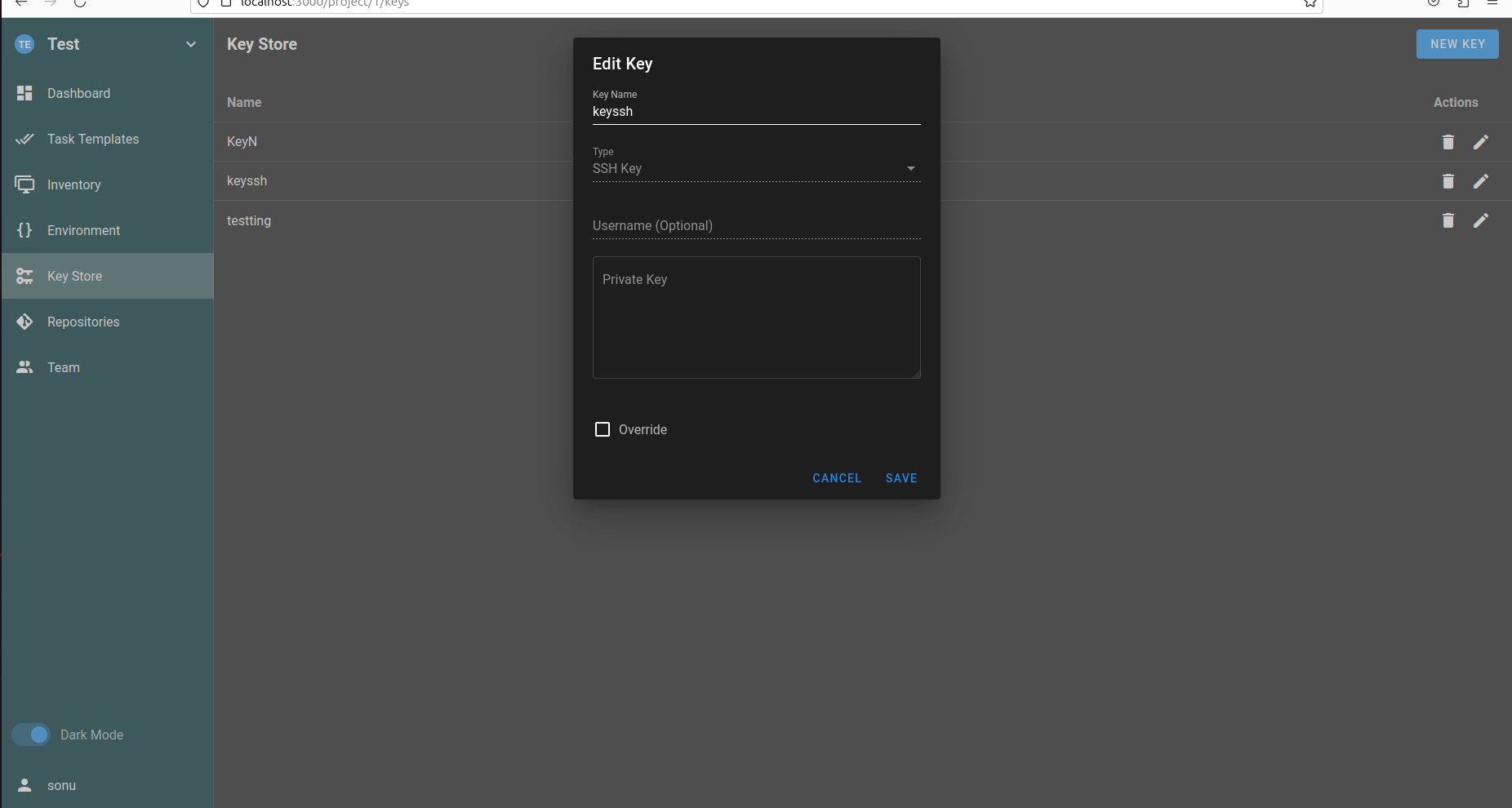
For run the simple Hello world playbook we have to do the following steps:

**Step 1:** Firstly I create the playbook on git repositories in .yml file.

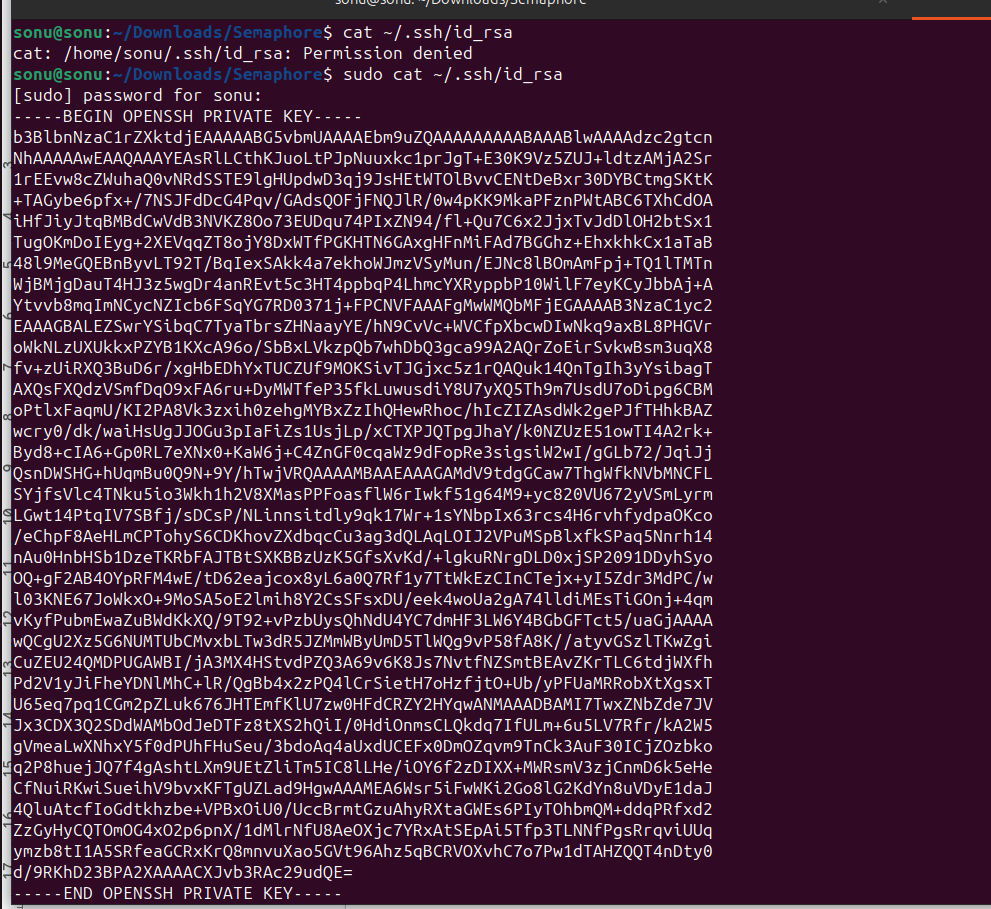


**Step 2:**

Firstly in my Semaphore Interface i create a Key store in which i create the name “**keyssh”** then select the option **“SSH key”** then through terminal i fetch our private key and paste it from in the private key box which is shown in images then click on Save button.

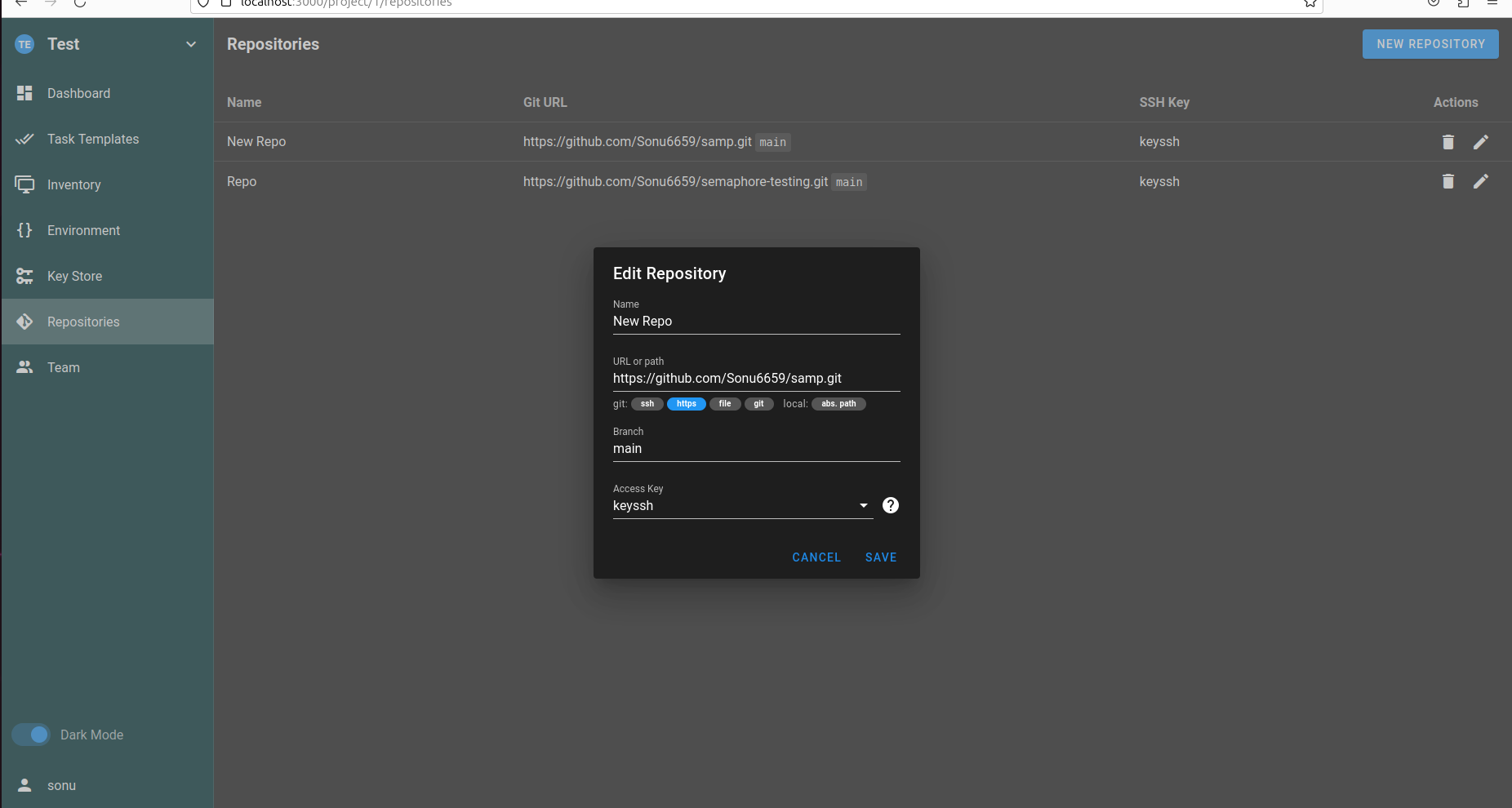


Here, I have passed my own system private key whose screenshots are attached below:



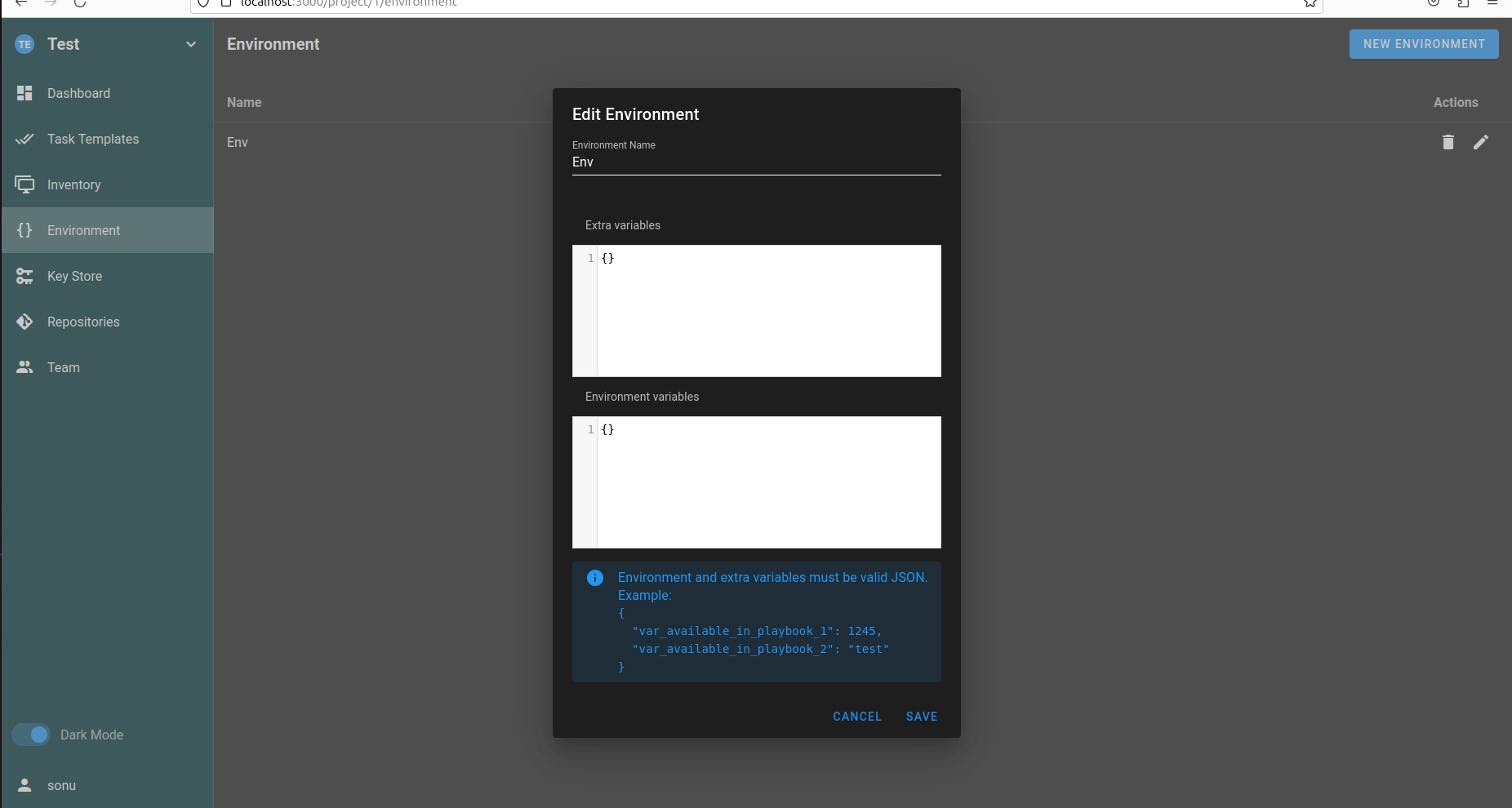
**Step 3:**

I made a repository in which i gave the repository name **“test”** and in the second attribute I passed the “**git ssh url”**of the **hell.yml** file and also input the branch name **“main”** and in another attribute automatically it show the **“keyssh”** which is i created in the “Key store” previously shown in figure then click on Save button.



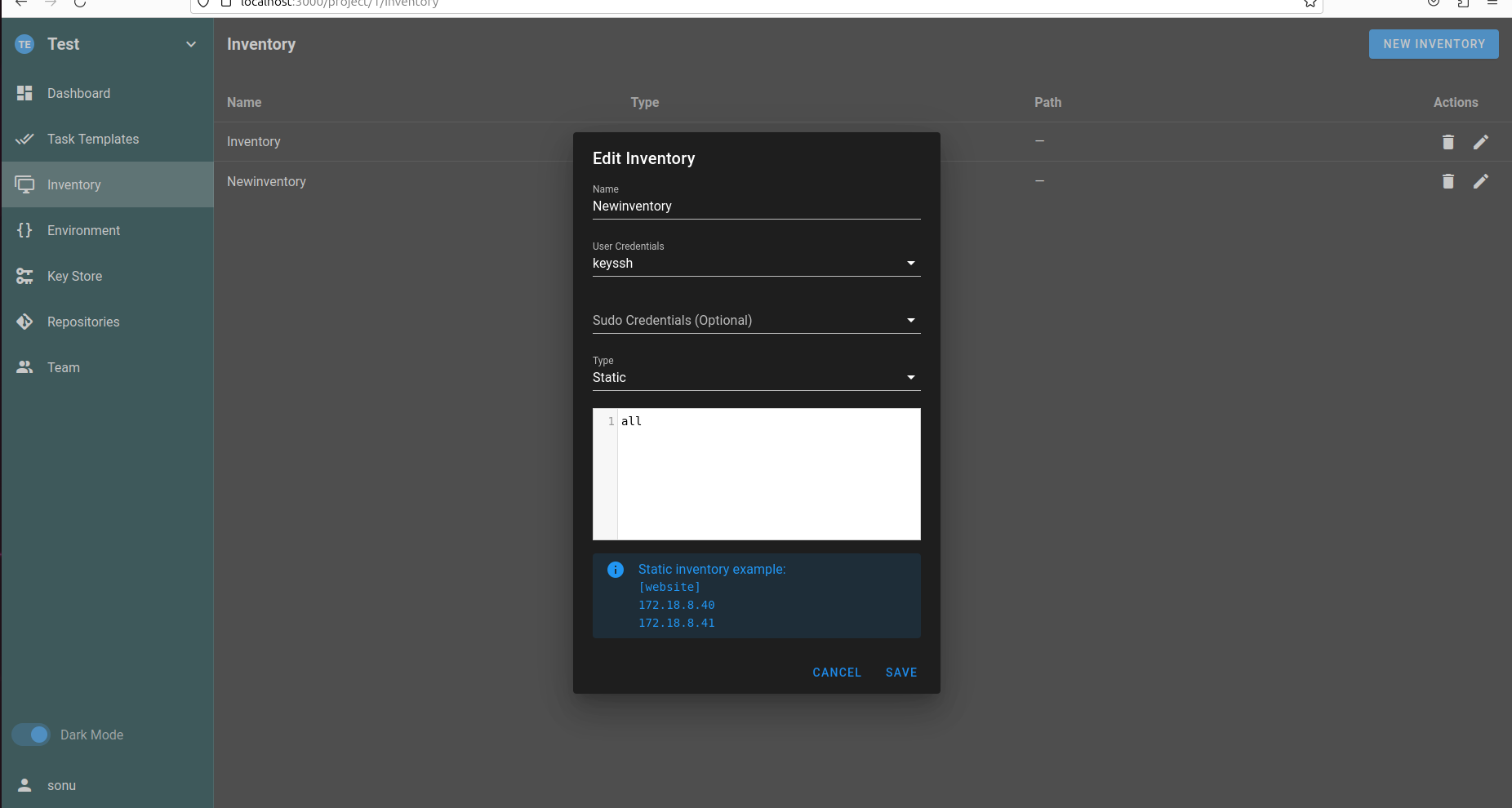
**Step 4:**

Then we set the Environment in which I created the environment name **“Env”** and I just passed the **{ } (curly braces)** in both variable boxes then click on save button

****

**Step 5:**

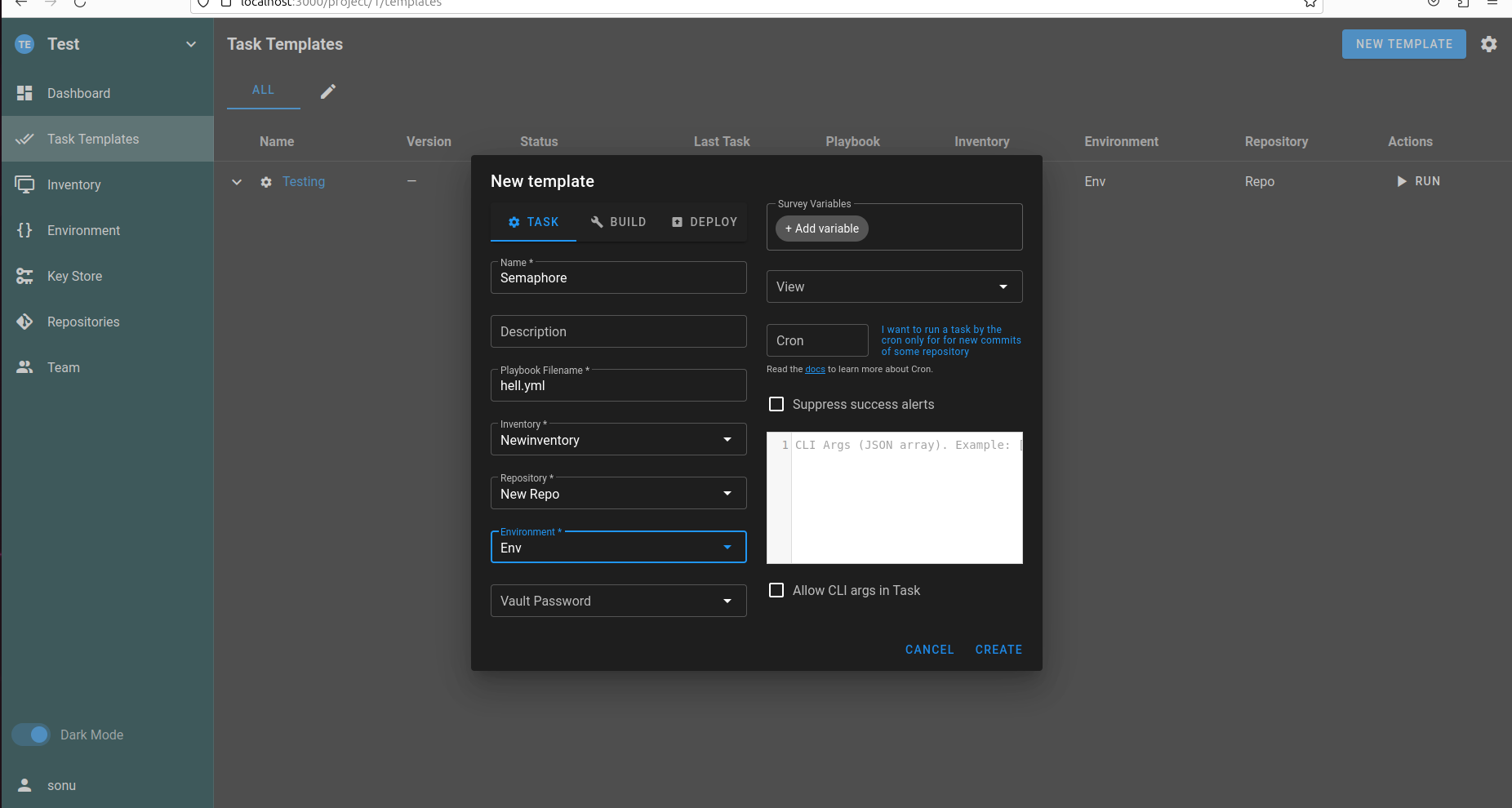
I make the inventory in which the inventory name is **“Newinventory”** and in another attribute “**User credentials”**option it shows the SSH key which is already created by me, then another attribute i select the **“Static YAML”** and in the box i just passed the **“all”** then click on Save button.



**Step 6:**

Then go to the Task Templates and click on the **“New Templates”** in which i input the name of the file is **“semaphore”** then input the our playbook file which is already created on github, then select the option Inventory, Repository, and Environment which is already we create and automatically get in the option box.

At last we click on the **Create button.**



**Step 7:**

In the last step simply i click on the **RUN** then our playbook is executed through the semaphore UI. The output of the Playbook are shown below:

