Lab: CI/CD Using AWX

Introduction:

Ansible Automation Platform automates the major stages of continuous integration, delivery, and deployment (CI/CD) pipelines—becoming the activating tool of DevOps methodologies.

CI/CD allows organizations to ship software quickly and efficiently.

Objectives:

- **Adding webhooks**
- **Creating a template**
- **Launch the AWX template on the AWX UI.**

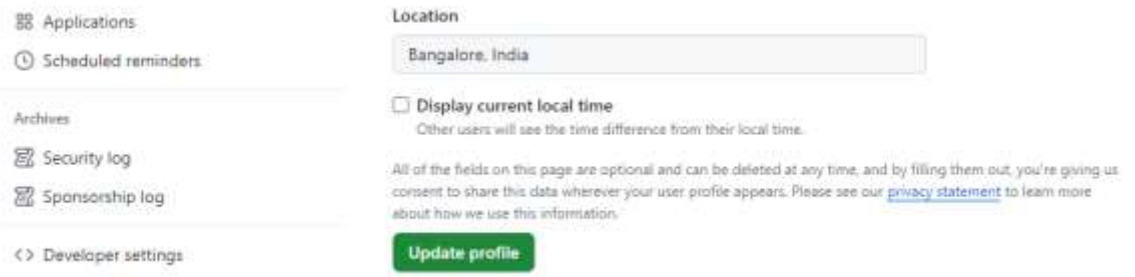
Note: Login to **eoc-controller** as **admin** user with password as **linux**

1. Adding webhooks.

1.1 In the profile settings of your GitHub account, click Settings.

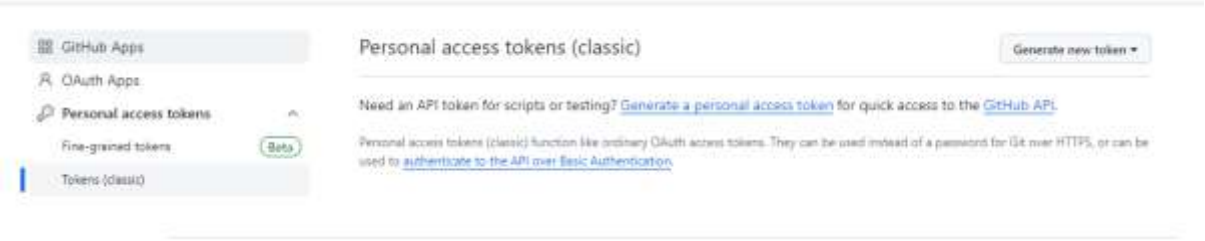
1.2 In the left pane click on developer settings.

Output:



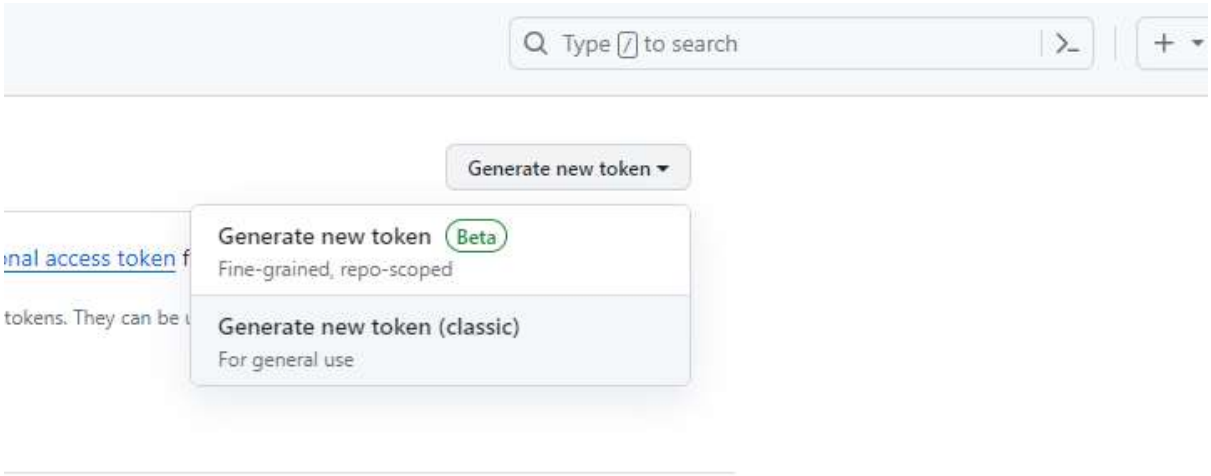
1.3 Click on Personal access token, select token (classic) and select the generate token button.

Output:

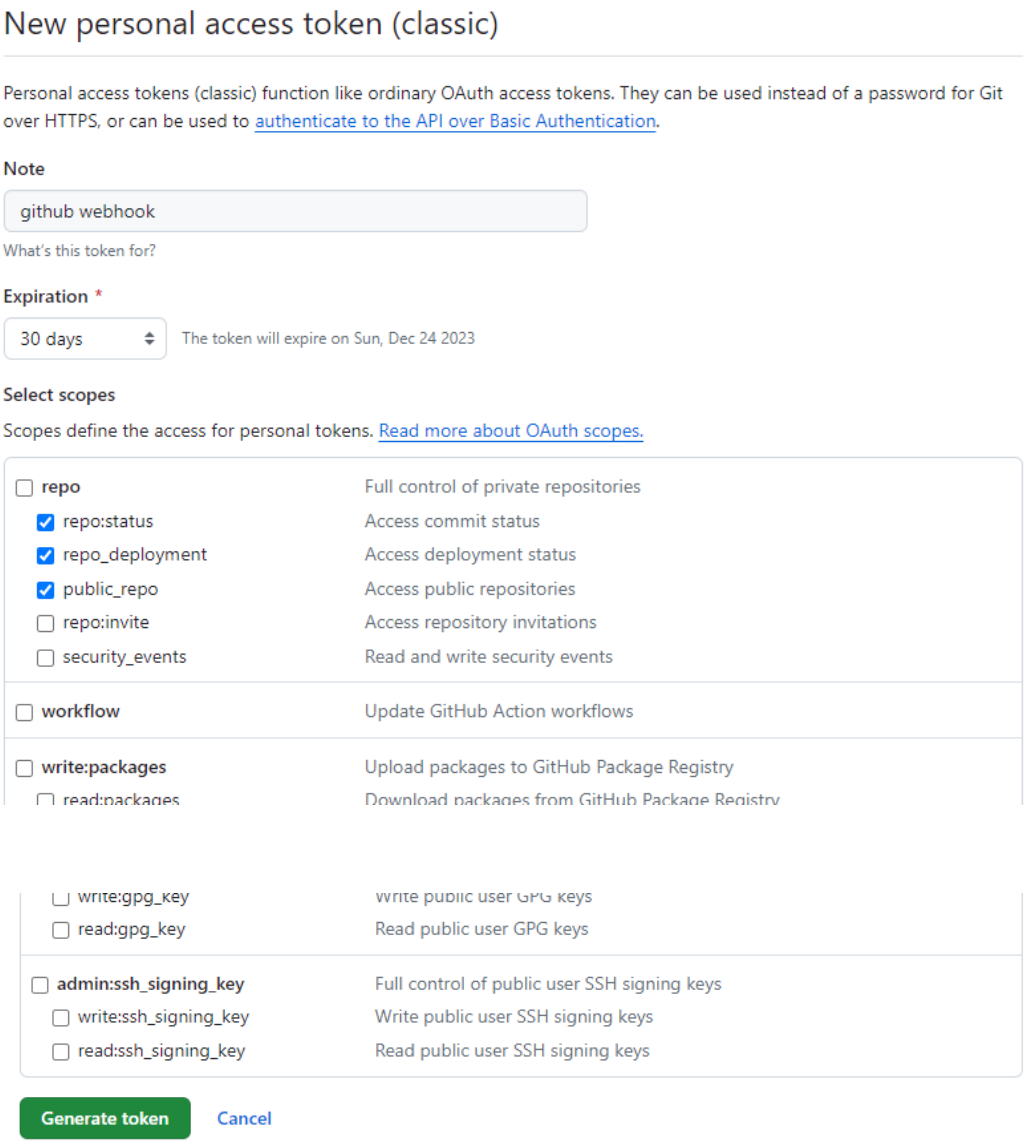


1.4 Select generate new classic token.

Output:

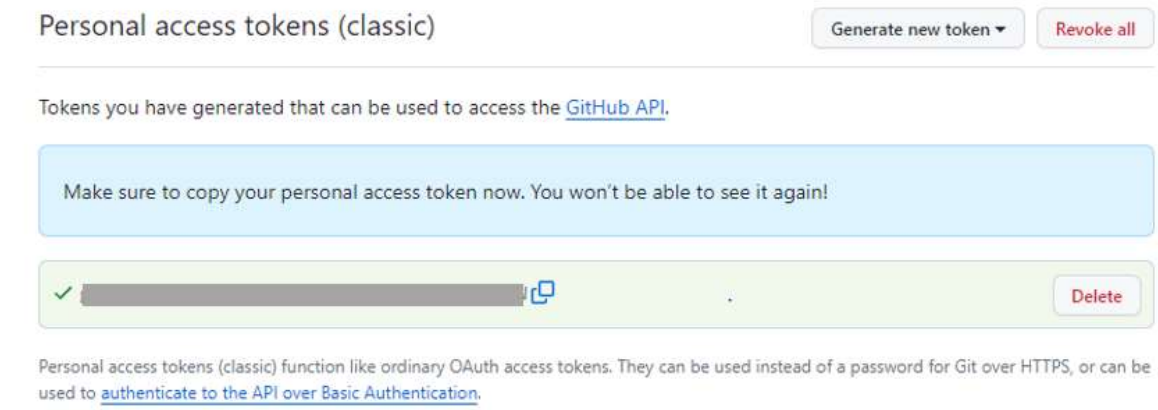


1.5 Select the required permissions and click on generate token.

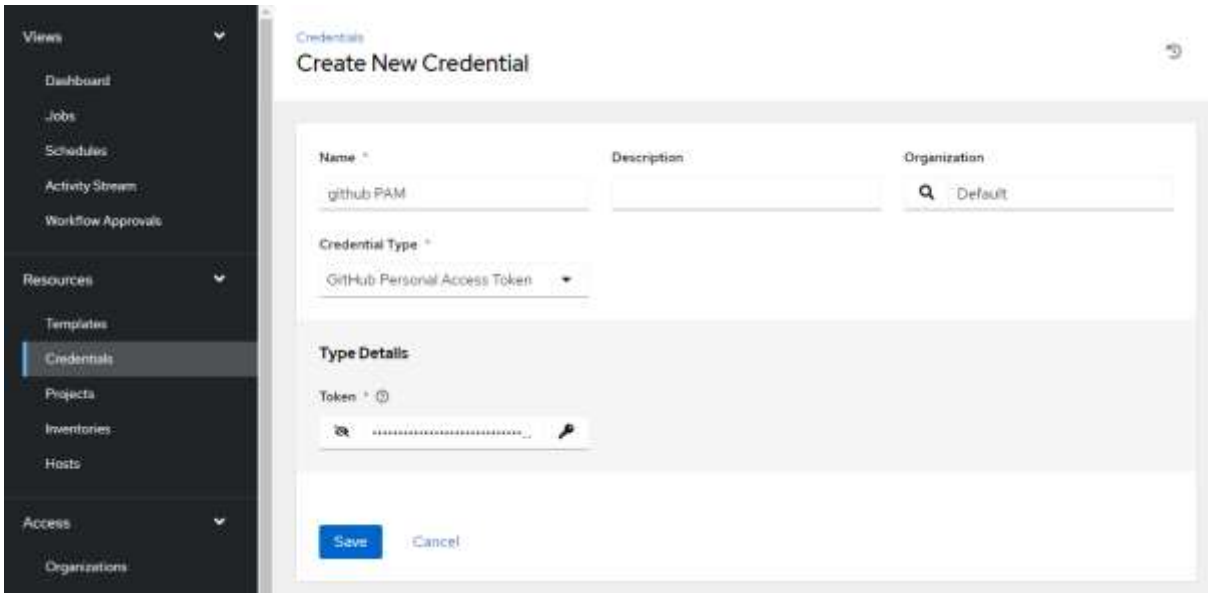


1.6 Copy the token to reuse it.

Output:



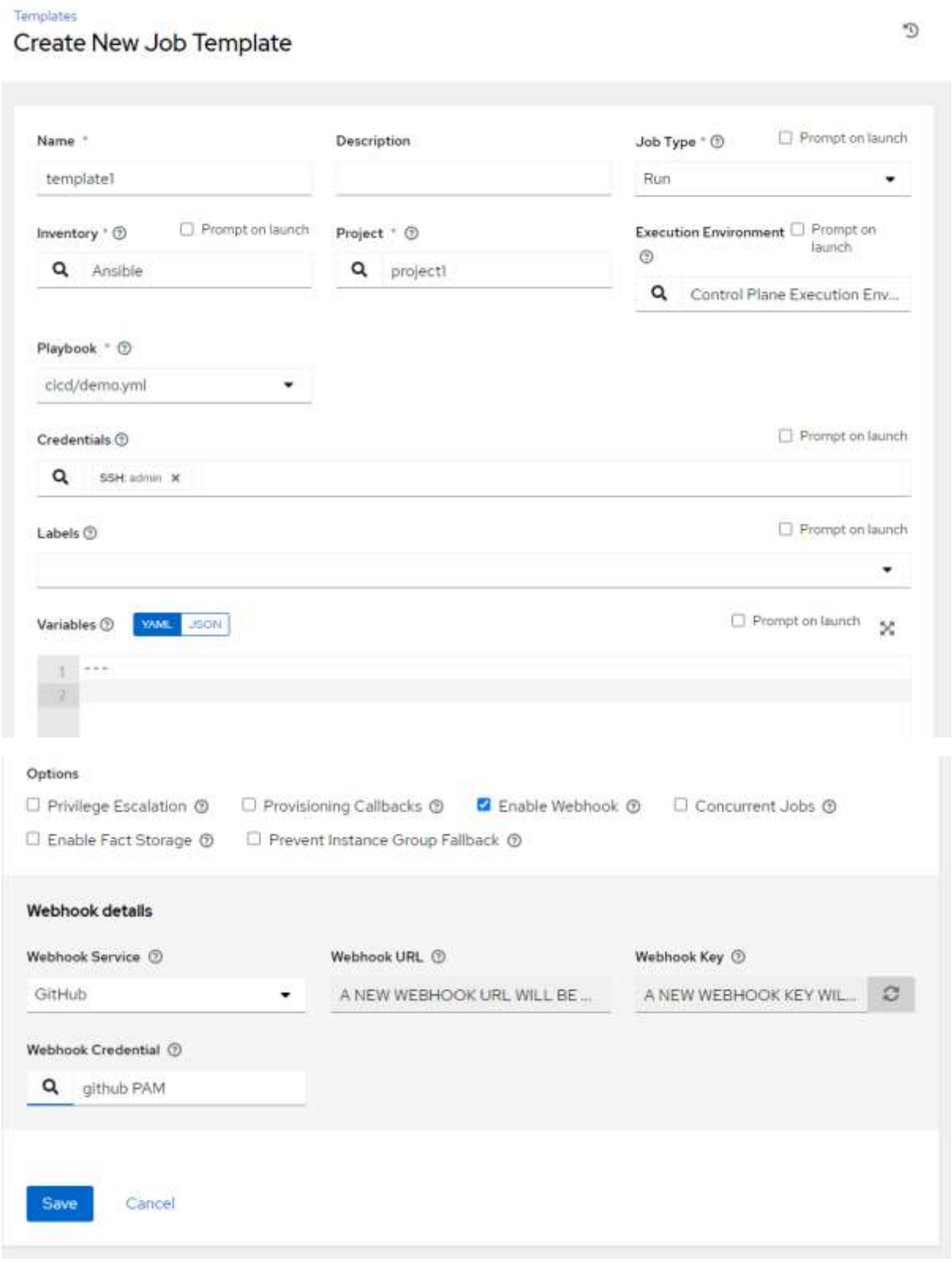
1.7 In the awx create a new credential.



Note: Paste the generated token in the token box.

2. Creating a template

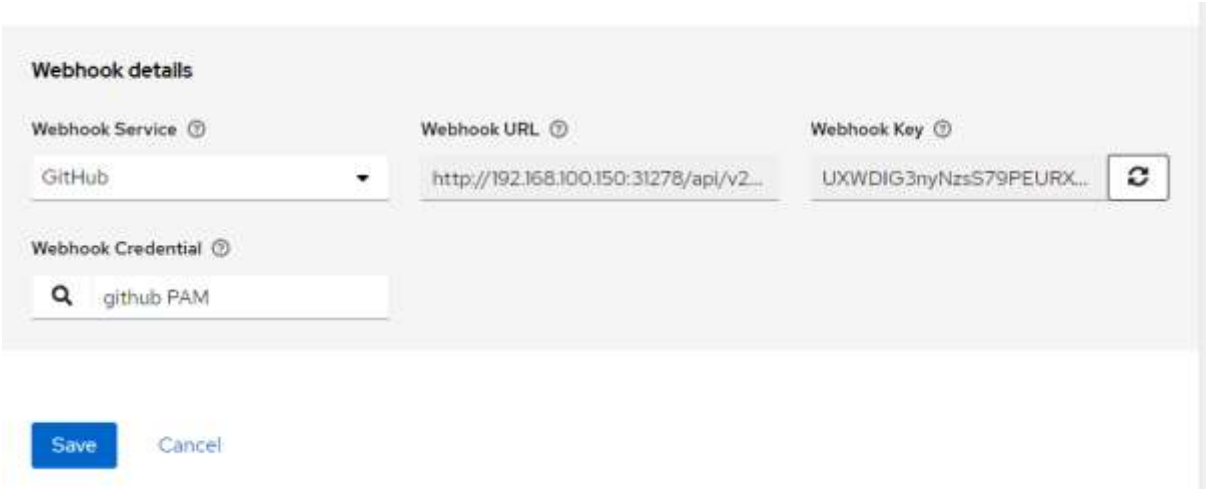
2.1 Create a new template using the same inventory and project.



Note: Add the credential that we created.

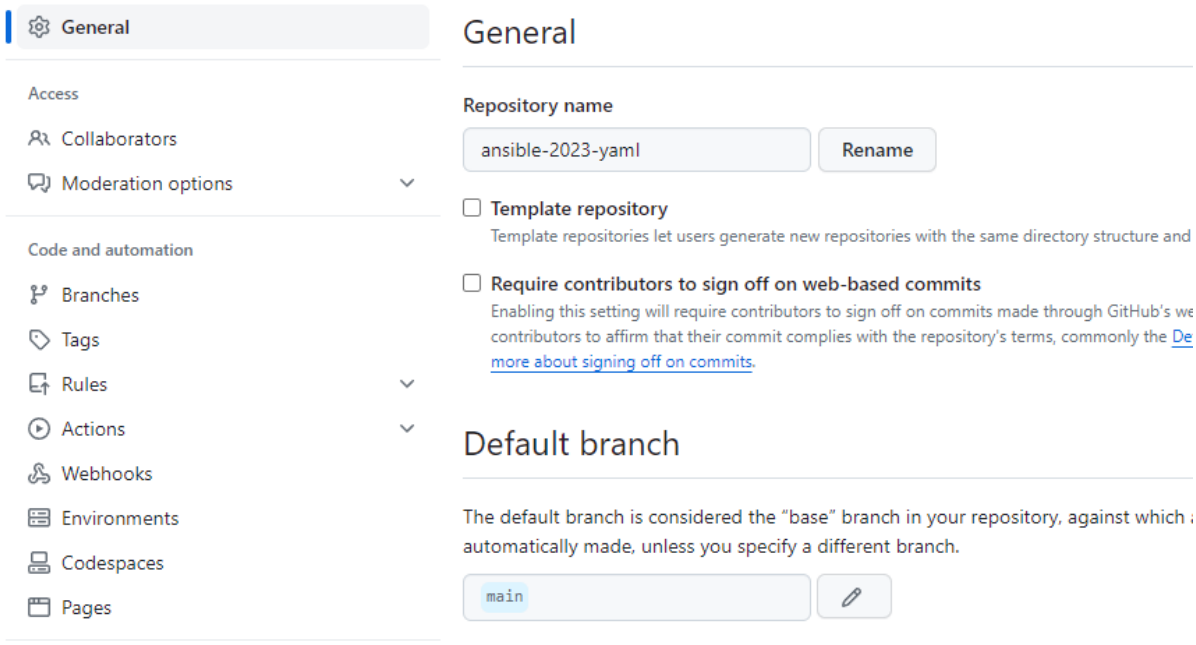
Save and open the template again to get the **URL** and **KEY**

2.2 Open the template in edit mode and copy the url and key

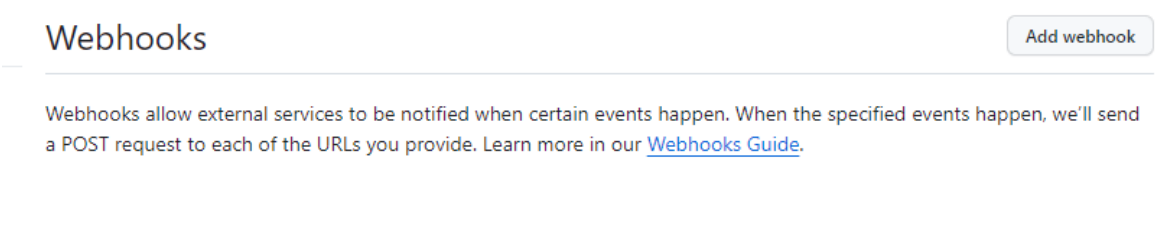


2.3 Go back to your github repo and repo settings.

Note: In the left pane find the webhooks ops and open it



2.4 Select add webhook.



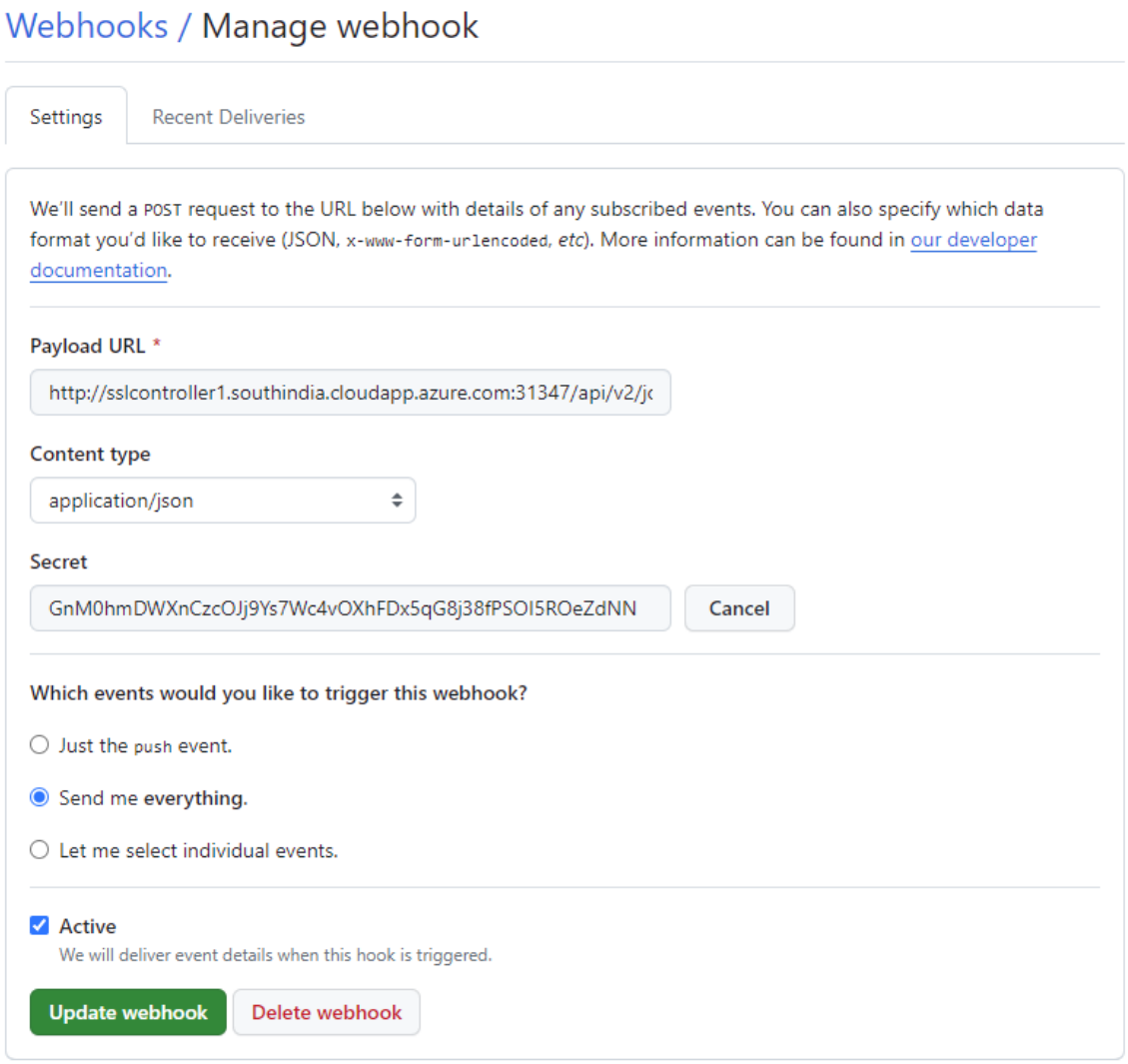
2.5 Add the URL and SECRET KEY that we got from the template.

Set the content type to json.

Select the send me everything option.

And select add webhook option.

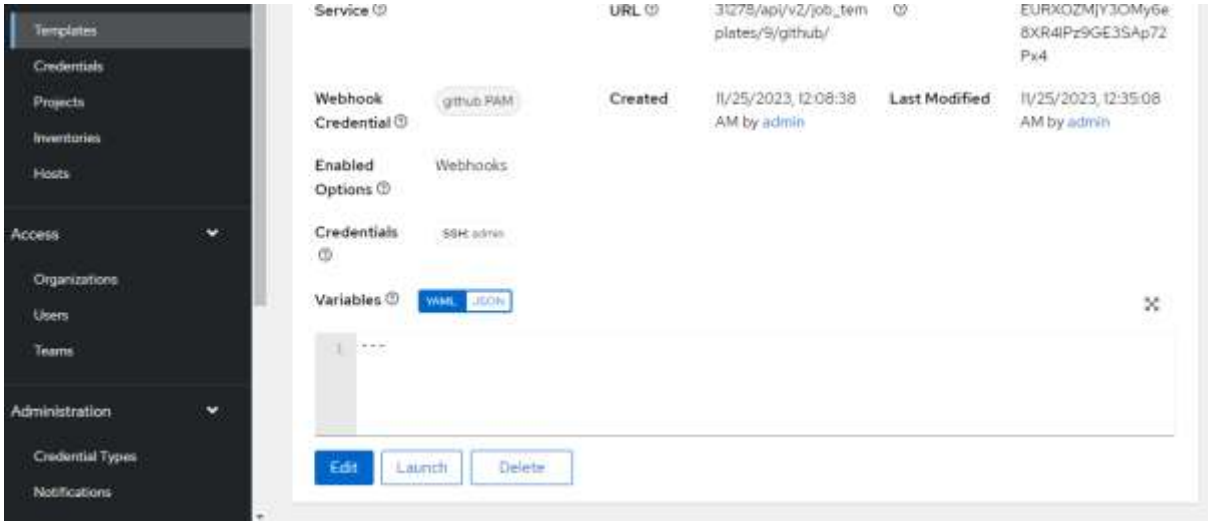
Output:



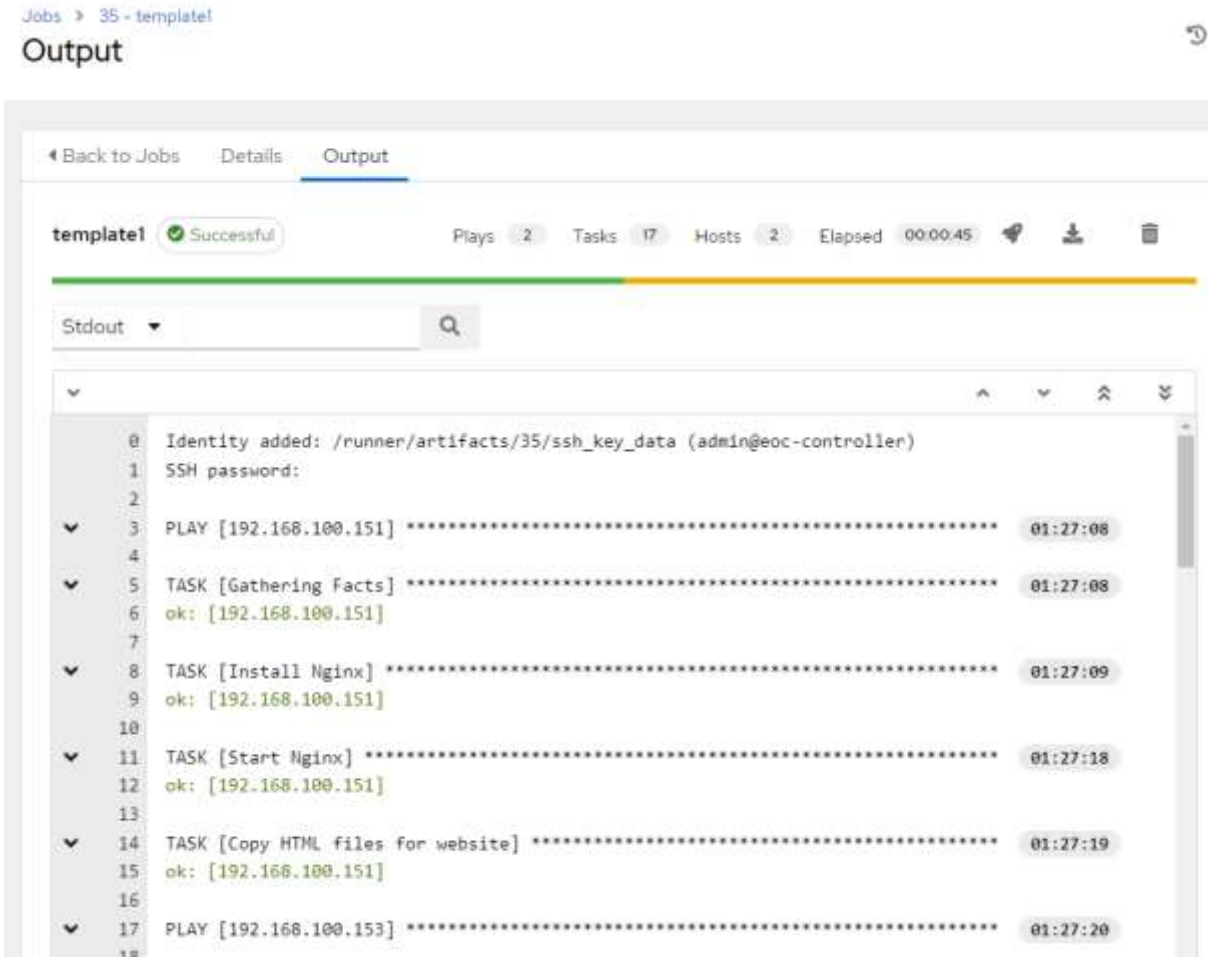
2.6 Webhook check and pushes the task into the AWX

3. Launch the awx template on the AWX UI.

3.1 Launch the template.



3.2 Verify the output.



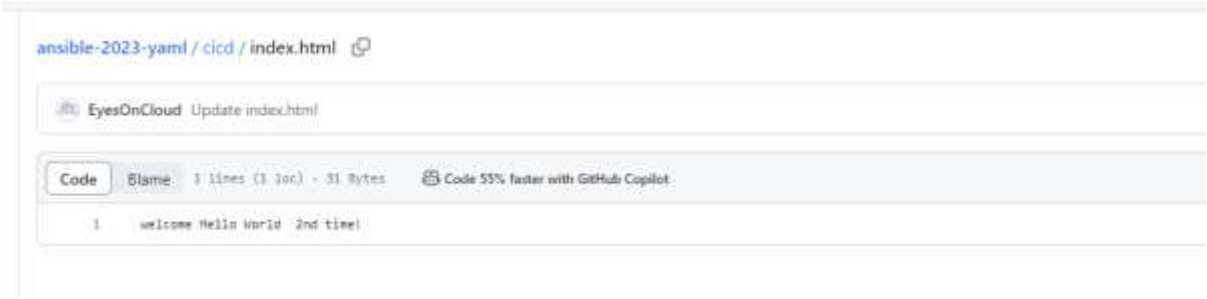
```
55  TASK [Start Node.js app on custom IP and port] ********************************  01:27:46
56  changed: [192.168.100.153]
57
58  TASK [Restart Nginx] ***********************************************************  01:27:47
59  changed: [192.168.100.153]
60
61  PLAY RECAP *********************************************************************  01:27:47
62  192.168.100.151            : ok=4    changed=0    unreachable=0    failed=0    skipped=0    res
    cued=0    ignored=0
63  192.168.100.153            : ok=13   changed=7    unreachable=0    failed=0    skipped=0    res
    cued=0    ignored=1
```

3.3 Verify our webpages.

Hello, World!

welcome Hello World!

3.4 Now change the **index.html** file in the github.



Note: Once the webhook is triggered the job will be running again.

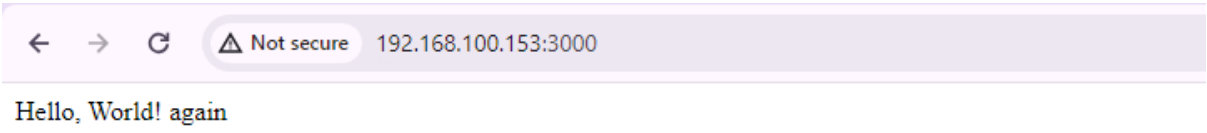
3.5 When the job is finished verify the output by refreshing the page.

welcome Hello World 2nd time!

3.6 Now you can also change the output of the play 2 in the task copy node.js app code to the server.

```
55        - name: Copy Node.js app code to the server
56          copy:
57            content: |
58              // app.js
59              const express = require('express');
60              const bodyParser = require('body-parser');
61
62              const app = express();
63              const port = 3000;
64
65              // Parse incoming requests with JSON payloads
66              app.use(bodyParser.json());
67
68              // Respond to GET requests with "Hello, World!"
69              app.get('/', (req, res) => {
70                res.send('Hello, World! again');
71              });
72
73              // Listen on port 3000
74              app.listen(port, () => {
75                console.log(`Server is running on http://192.168.100.153:${port}`);
76              });
77            dest: nodeapp/app.js
78            mode: 0644
```

3.7 Again, the job is launched and you can see the output:



Hello, World! again