**Sockshop application + OpenShift + Azure + Jenkins + Ansible**

This is a modification to the code base used in ACM workshop given int the below link.

<https://github.com/dynatrace-acm/dtacmworkshop>

The requirement was to have the application deployed in Openshift and Azure.

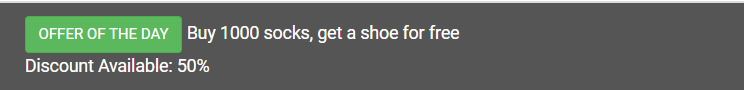
To enable Shift Left and Self-Healing use case Jenkins and Ansible deployed in a server.

The discount applied in promotion campaign appears in the top nav bar after page refresh. Once checkout is done and error is displayed when campaign is on. The error is not displayed when campaign is off.

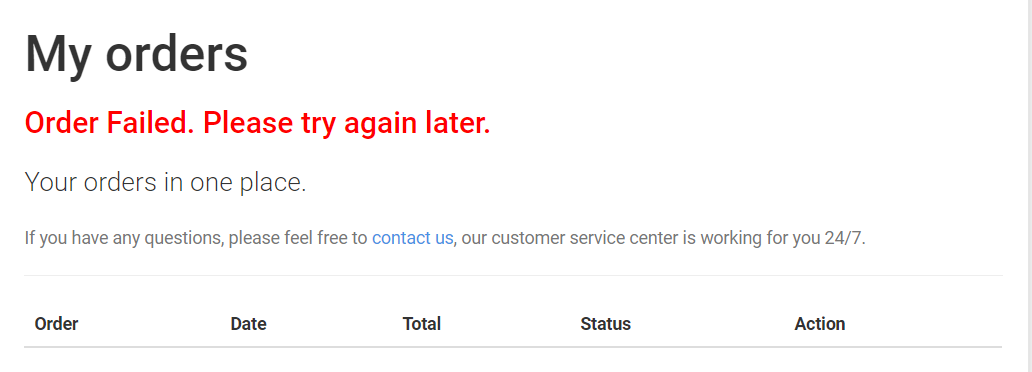
# The Demo steps:

1. Sockshop microservices application deployed in OpenShift in Azure. Dynatrace UI reflecting automated monitoring of the application in OpenShift and Azure monitoring
2. Automated Quality check – Shift left. Run Jenkins step two to demonstrate this scenario. The load test fails as criteria is not met and code is not deployed to production
3. Self-Healing – When ansible campaign is started. Refresh the page and the discount in the top navigation bar will get updated. When check out is done the error message will be shown in the last page. When campaign is closed the Discount will be set to zero and the error message will not appear in the checkout page.
4. If session replay is setup. You can replay all the steps.

Campaign started the discount gets updated to 50%



When campaign is on and checkout is done, the below error appear.



# Step 1 – Provisioning

* Provision Cloud Instances
  + 1 Instance for Active Gate to run the Kubernetes plugin (use min. Active Gate specs) OPTIONAL but good to show
  + 1 instance for a single node Openshift in Ubuntu (provision at least 16 GB RAM, CPU does not matter)
  + 1 instance of RHEL to setup ansible and ansible tower
* Create a userid and pwd to login using Putty

# Step 2 – Enable network permissions for inbound/outbound traffic

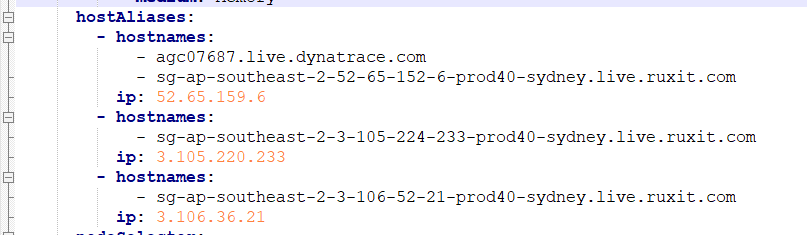
* Enable inbound port for 80, 8080, 22(SSH), 443
* Enable outbound port for 8443, 443, 9999
* Create a DNS name and make IP as Static

# Step 3 – Prep system for OKD

* Run installokd.sh script as root
* Install script assumes Ubuntu thus it uses apt-get to install packages. If you are running any other Linux distro, please modify script accordingly
* This will provision
  + Docker engine
  + Setup docker engine for OKD
  + Install OC CLI

# Step 4 – Run OpenShift 3.11 as a single node cluster

* Login to a user in linux (suggested not to use root)
* Execute this command to include this user in the docker group so that you can run docker commands without using sudo
* sudo usermod -aG docker <USERID>
* Add Dynatrace IP and hostnames mapping in /etc/hosts.
* In the manifest folder there are YAML files – In these files, there is host aliases added with IP and hostname. Replace this with the IP and hostname for your Dynatrace tenant. Sometimes the pods will not be able to resolve the host names. The host aliases will add it to each pod which gets created. If not required remove hostaliases from the yaml file. You can remove these lines if your pods are enabled to access external sites.



* Execute runSockshop.sh (Replace the Public IP and Public DNS name of the VM)
* This will:
  + Start OKD with the detected public IP and public DNS
  + Prep OpenShift to deploy sockshop application
  + Create an OC project
  + Sockshop application is created in dev and prod environments
  + Start the Sockshop application
  + Access the OpenShift console URL by copying the URL output given in the script
* After the script is run successfully go to the openshift console. Login with developer/developer.
* Validate that the app has been deployed via the OpenShift console. The pods may take some time to load. Wait until all pods are running. Check the logs of each pod in openshift. Ensure there is no errors.
* Click on the Frontend URL to access the sockshop application in browser.

# Step 5 – Run Carts load test

Run cartsLoadTest.sh in background. This will create traffic to the sockshop backend services. The below command will run the script in background and put the output to the log file.

./cartsLoadTest.sh > loadtestoutput.log 2>&1 &

# Step 6 – Setup Ansible and Ansible Tower

Follow the instructions in ansible sop document to setup ansible and ansible tower. Once setup is done, login to ansible tower, follow the below steps to configure self-healing for sockshop application.

1. Keep your tenant Id, Dynatrace API token ready.
2. Create a new Credential Type. Add the below fields
3. Input config:

fields:

- id: dt\_api\_token

type: string

label: Dynatrace API Token

secret: true

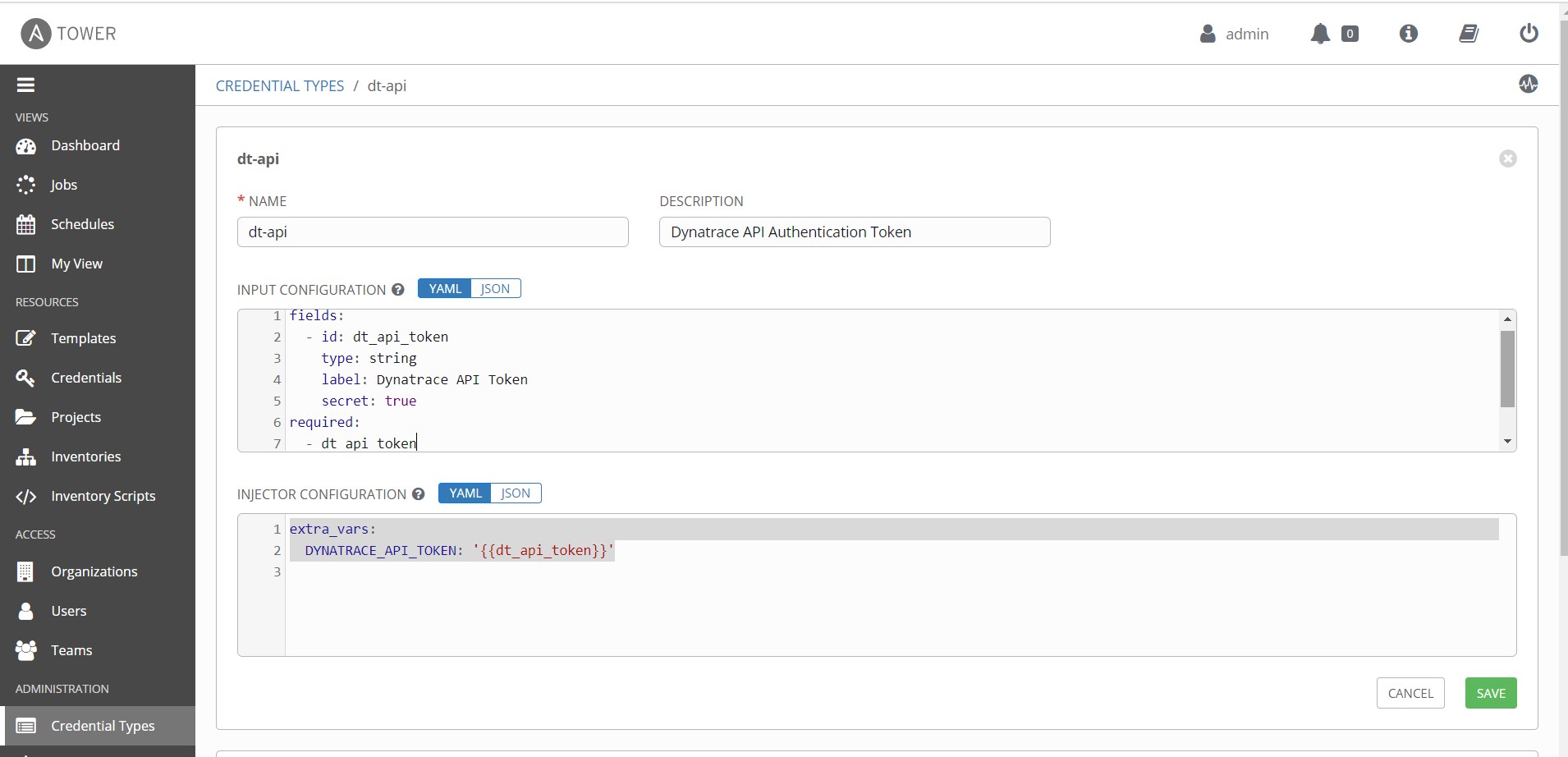
required:

- dt\_api\_token

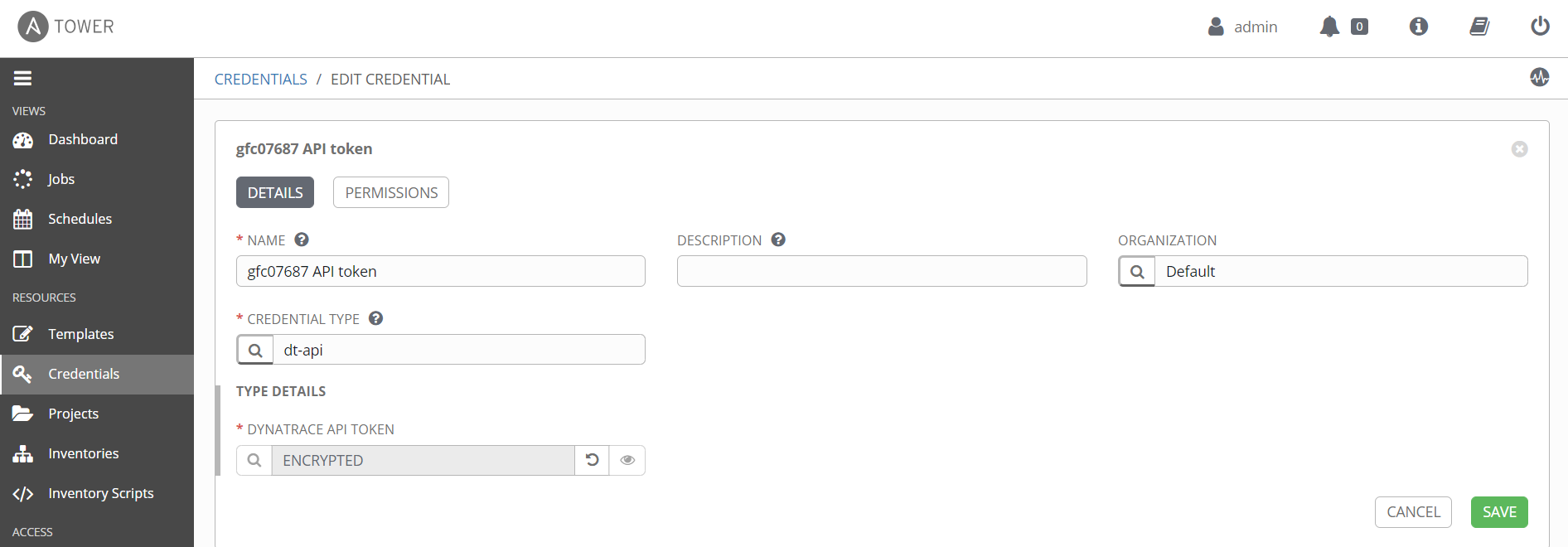
1. Injector config:

extra\_vars:

DYNATRACE\_API\_TOKEN: '{{dt\_api\_token}}'

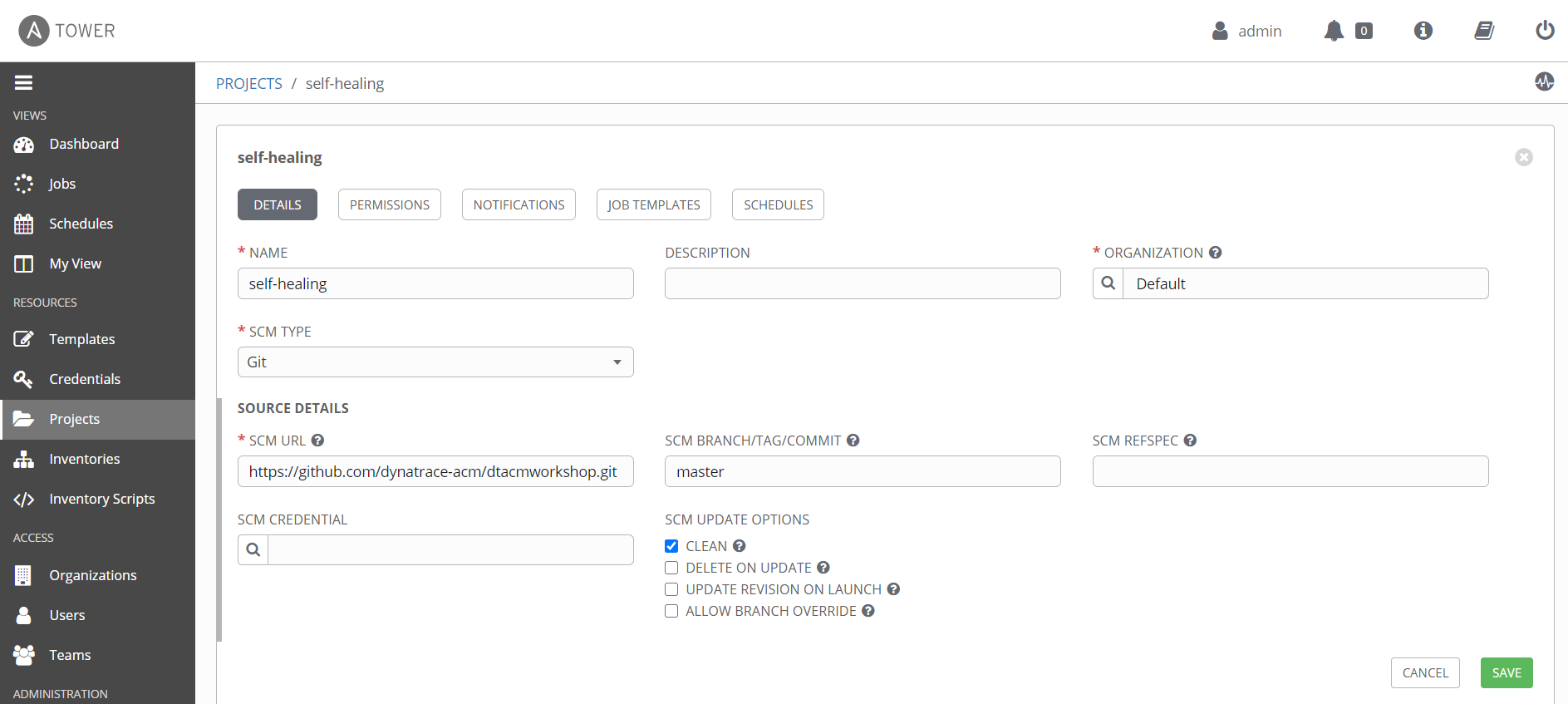


1. Create a new Credentials. Select credential type as ‘dt-api’ (previously created cred type). Add Dynatrace API token in the field.



1. Create a Project. Provide the details as per the screenshot.

SCM URL https://github.com/dynatrace-acm/dtacmworkshop.git



1. Create an Inventory.

Add these in the variables. Update the tenant url to your tenant id.

Update the cards promotion URL to the IP address as your application

http://**carts-production.13.92.239.40.nip.io**/carts/1/items/promotion

tenanturl: "https://gfc07687.live.dynatrace.com"

carts\_promotion\_url: "http://carts-production.13.92.239.40.nip.io/carts/1/items/promotion"

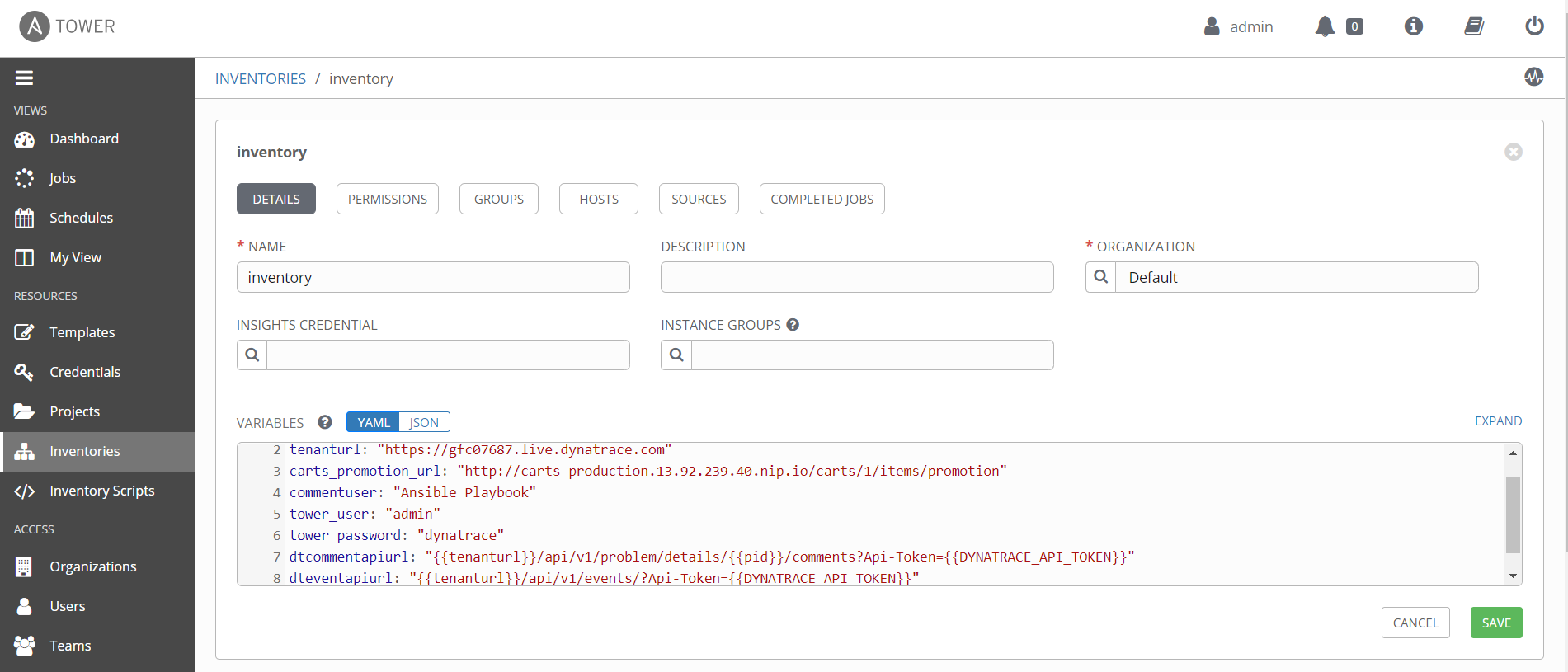
commentuser: "Ansible Playbook"

tower\_user: "admin"

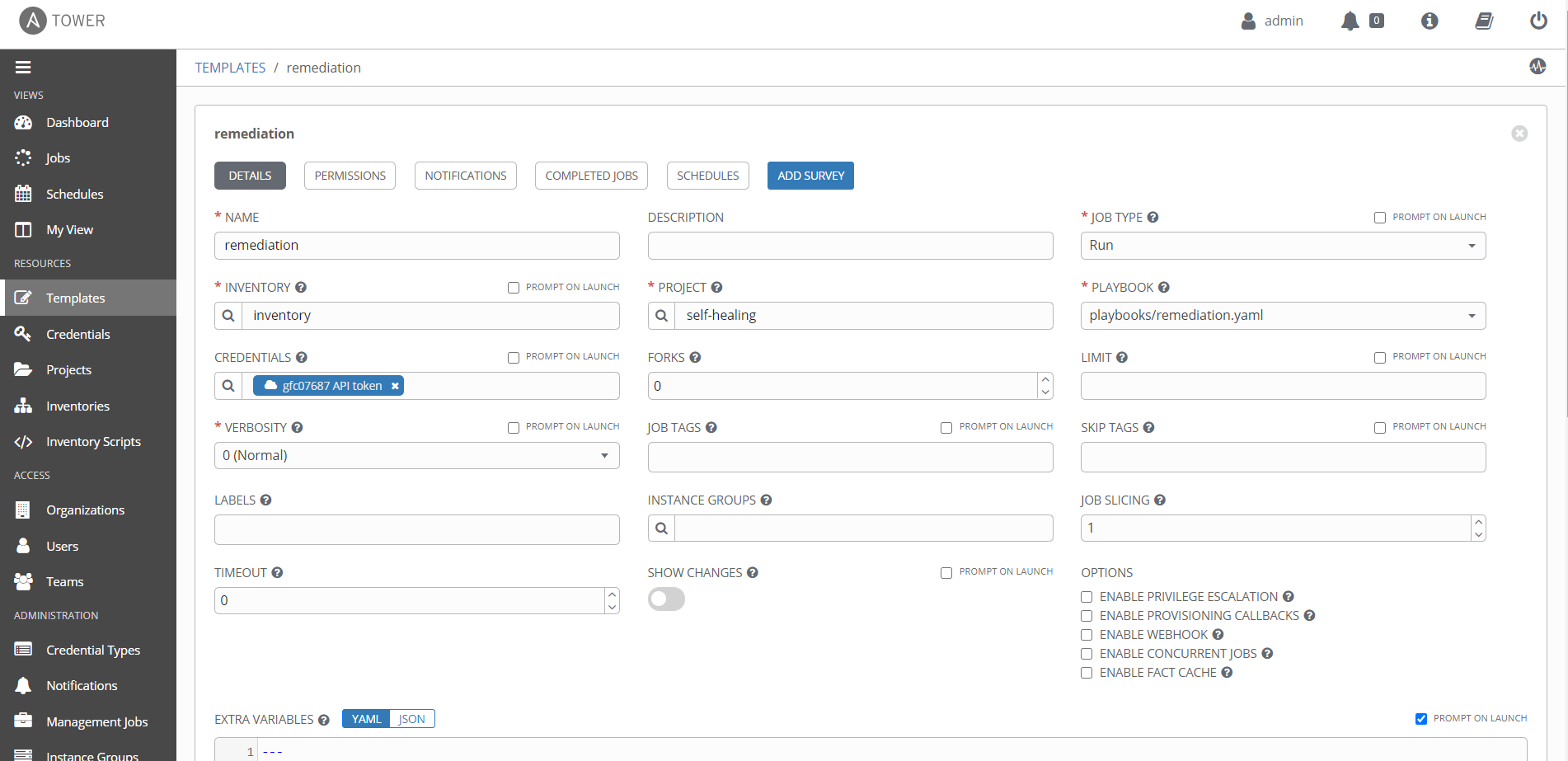
tower\_password: "dynatrace"

dtcommentapiurl: "{{tenanturl}}/api/v1/problem/details/{{pid}}/comments?Api-Token={{DYNATRACE\_API\_TOKEN}}"

dteventapiurl: "{{tenanturl}}/api/v1/events/?Api-Token={{DYNATRACE\_API\_TOKEN}}"



1. Create Remediation Template. Select the options are provided.



1. Create Stop Campaign Template. Include these variables and save.

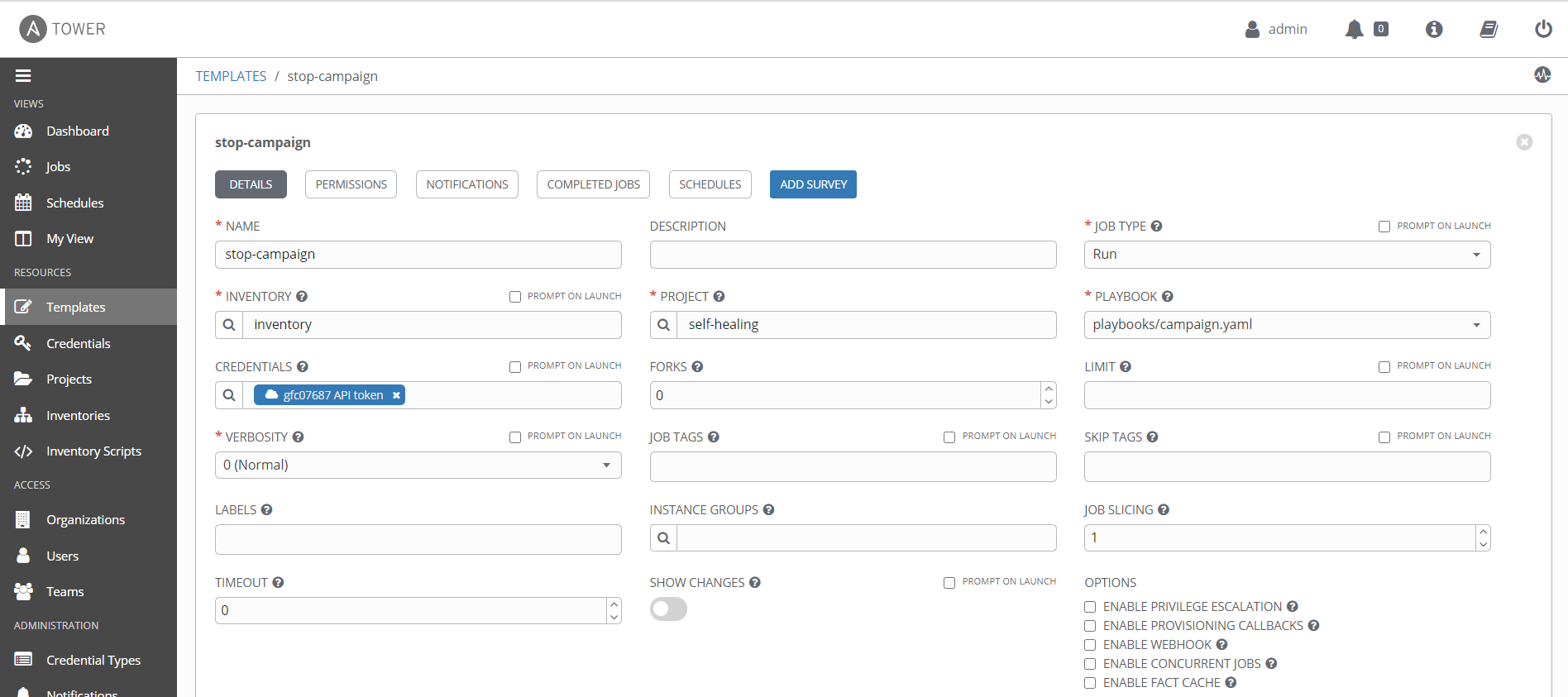
promotion\_rate: "0"

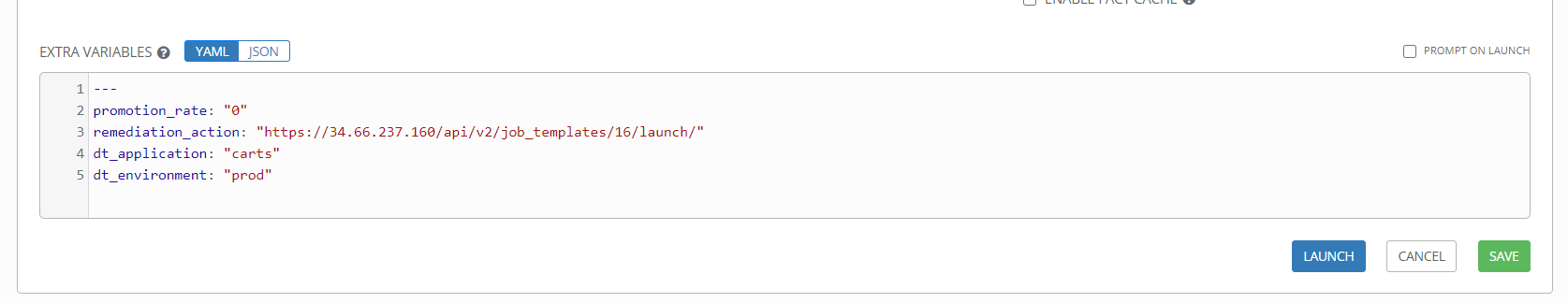
remediation\_action: "https://34.66.237.160/api/v2/job\_templates/16/launch/"

dt\_application: "carts"

dt\_environment: "prod"

After saving, copy the template id from the URL and update the IP where the Ansible tower is accessible.





1. Create the Start Campaign template. Include these variables and save.

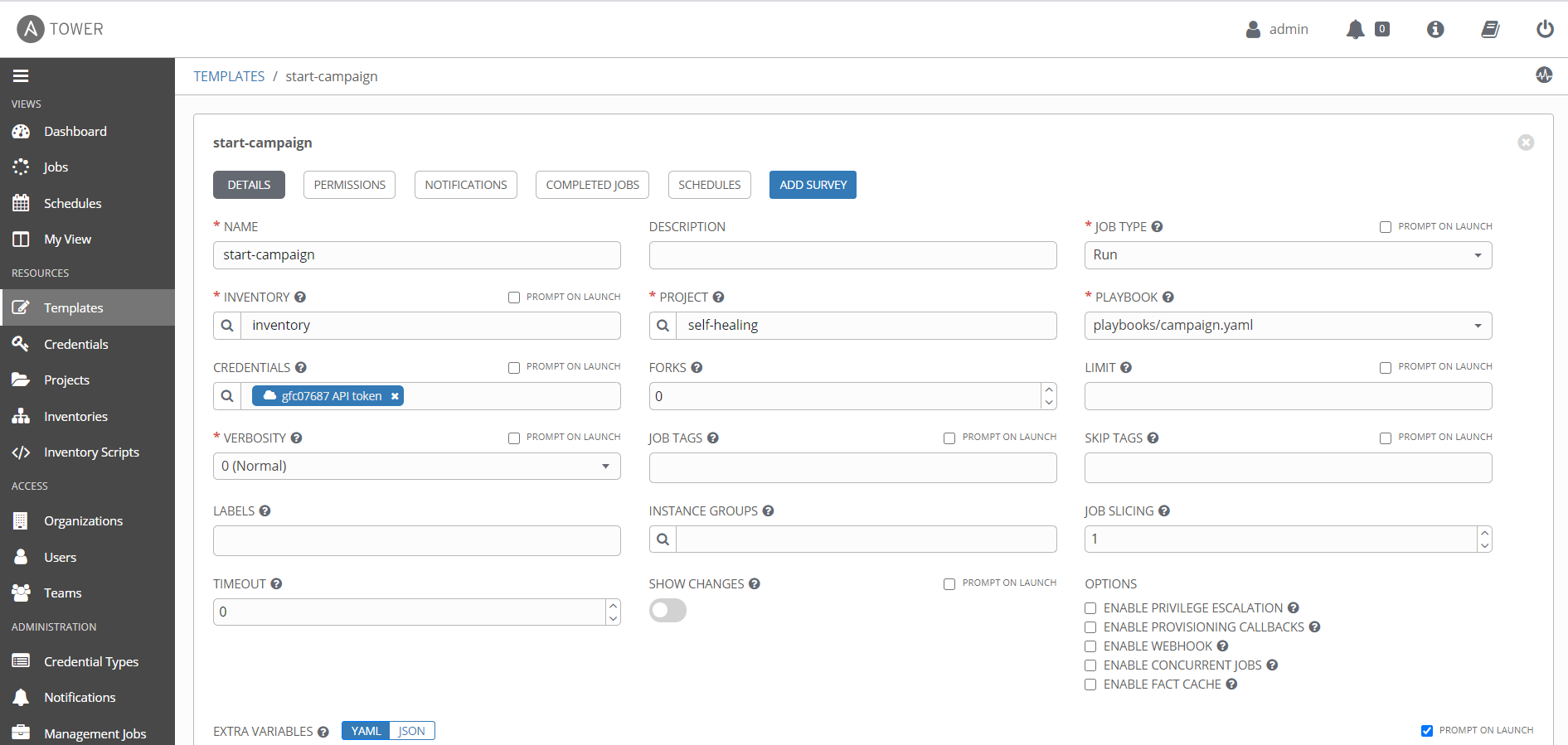
promotion\_rate: "50"

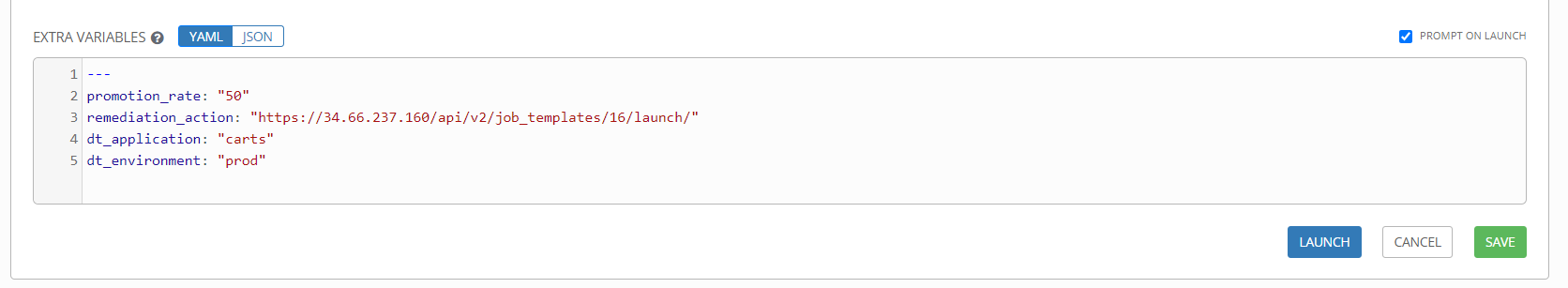
remediation\_action: "https://34.66.237.160/api/v2/job\_templates/16/launch/"

dt\_application: "carts"

dt\_environment: "prod"

Copy the stop template id from its URL and update the IP where the Ansible tower is accessible.





1. Integrate with Dynatrace using the remediation template URL.

# Step 7 – Jenkins Setup

1. Install Java first
   1. Below is the command for installing java in Ubuntu

sudo apt update

sudo apt search openjdk

sudo apt install openjdk-8-jdk

java -version

1. Install Jenkins
   1. Below is the command for installing jenkins in Ubuntu

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > \

/etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins

sudo systemctl status Jenkins

1. By default Jenkins uses 8080 port. change port of Jenkins if needed

sudo nano /etc/default/Jenkins

Update the port in HTTP\_PORT = 8080

sudo service jenkins restart

1. Install JMETER

wget http://apachemirror.wuchna.com//jmeter/binaries/apache-jmeter-5.3.tgz

tar xf apache-jmeter-5.3.tgz

cd apache-jmeter-4.0/bin/

Start Jmeter in background ./jmeter-server &

If you get this error

Server failed to start: java.rmi.server.ExportException: Listen failed on port:

java.io.FileNotFoundException: rmi\_keystore.jks

Then change this line in jmeter.properties and restart server

server.rmi.ssl.disable=true

1. Jenkins uses this path while working /var/lib/jenkins/workspace

Copy the Sockshop folder to this path

chown -R jenkins:jenkins Sockshop - use this command to update the user of the folder.

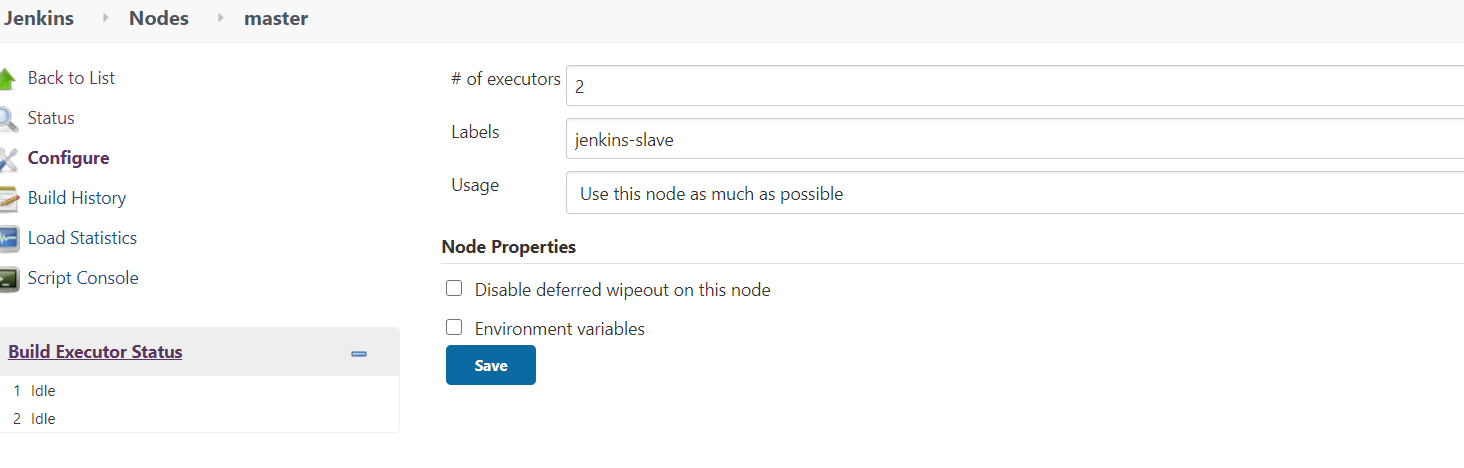
Jenkins executes all the scripts using jenkins user

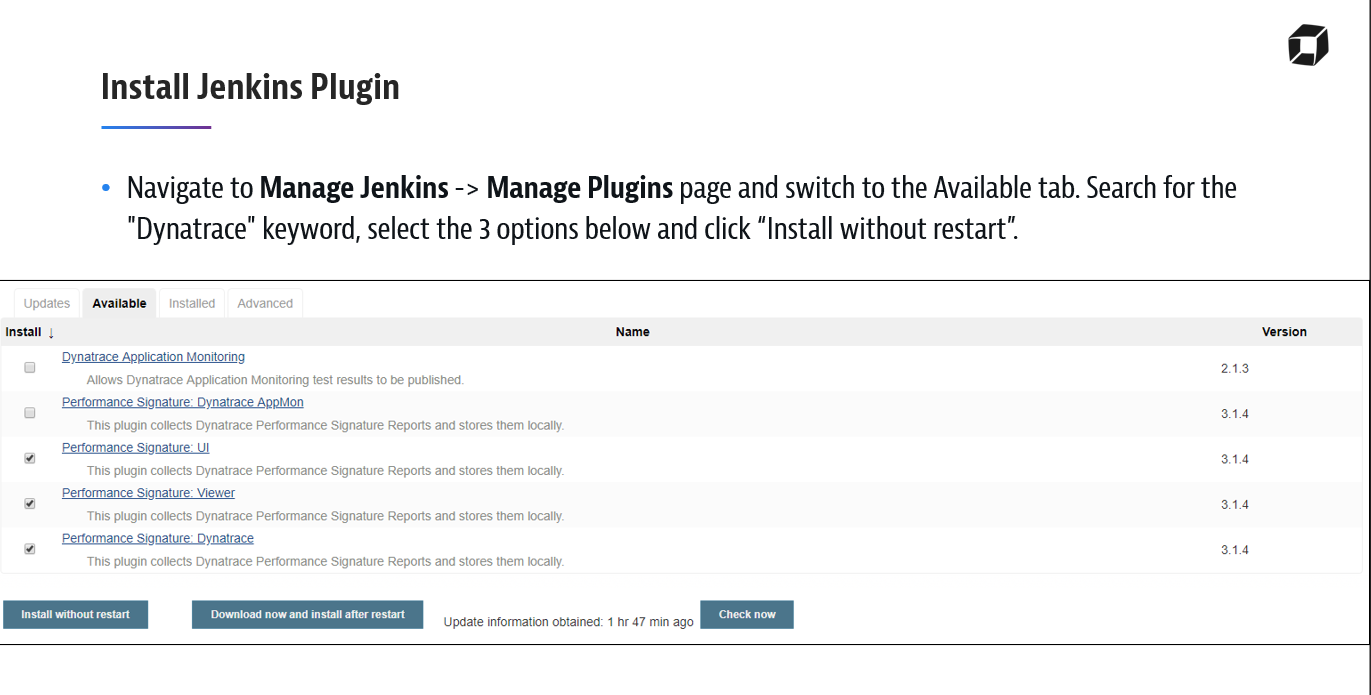
Use the below to give permissions to folder and scripts accordingly

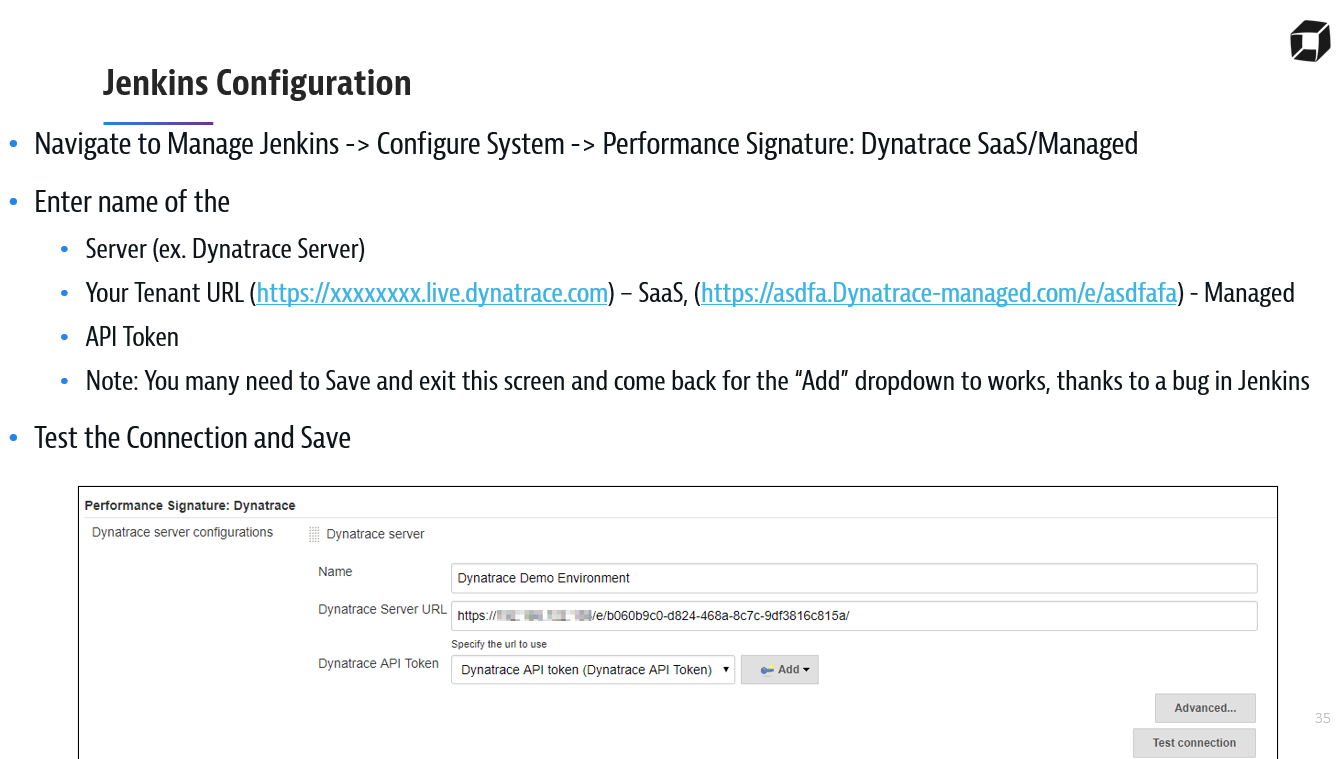
chmod ugo+rwx foldername to give read, write, and execute to everyone

chmod a=r foldername to give only read permission for everyone

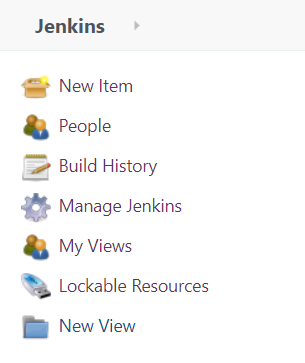
1. Configurations in Jenkins
   1. Manage Jenkins 🡪 Manage Nodes and Clouds
   2. Select Master and click configure. Update the label as shown below

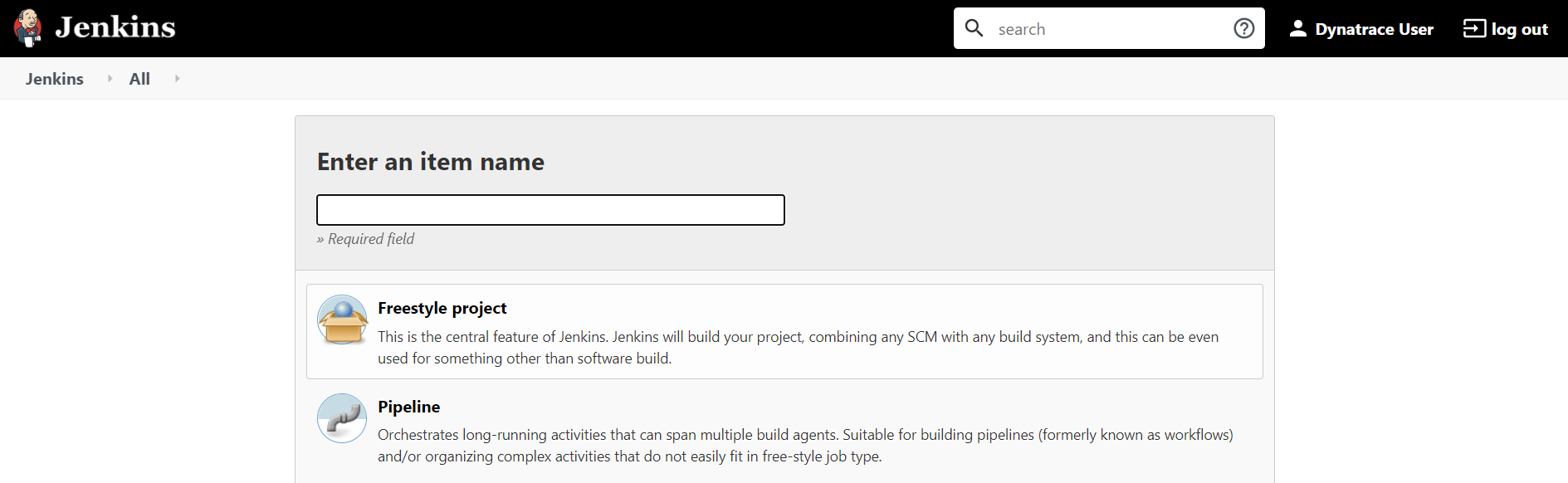


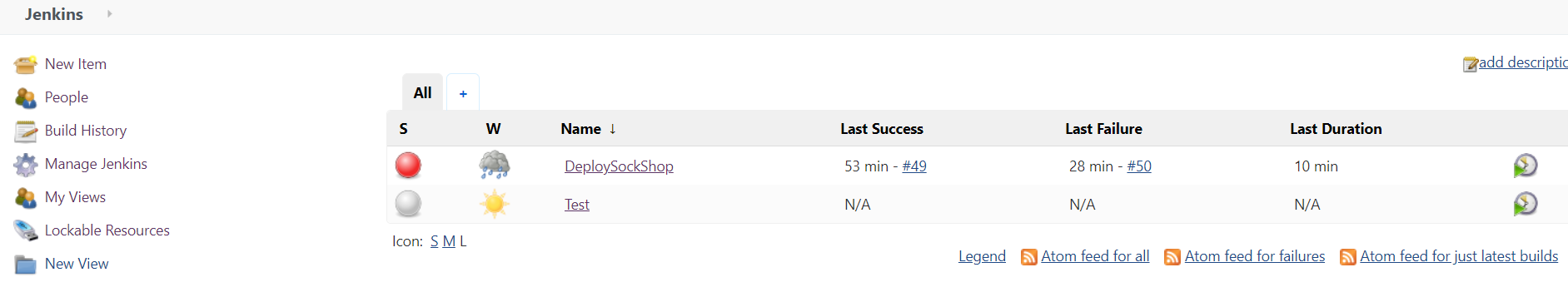
* 1. Install Performance Plugins
  2. Configure Jenkins

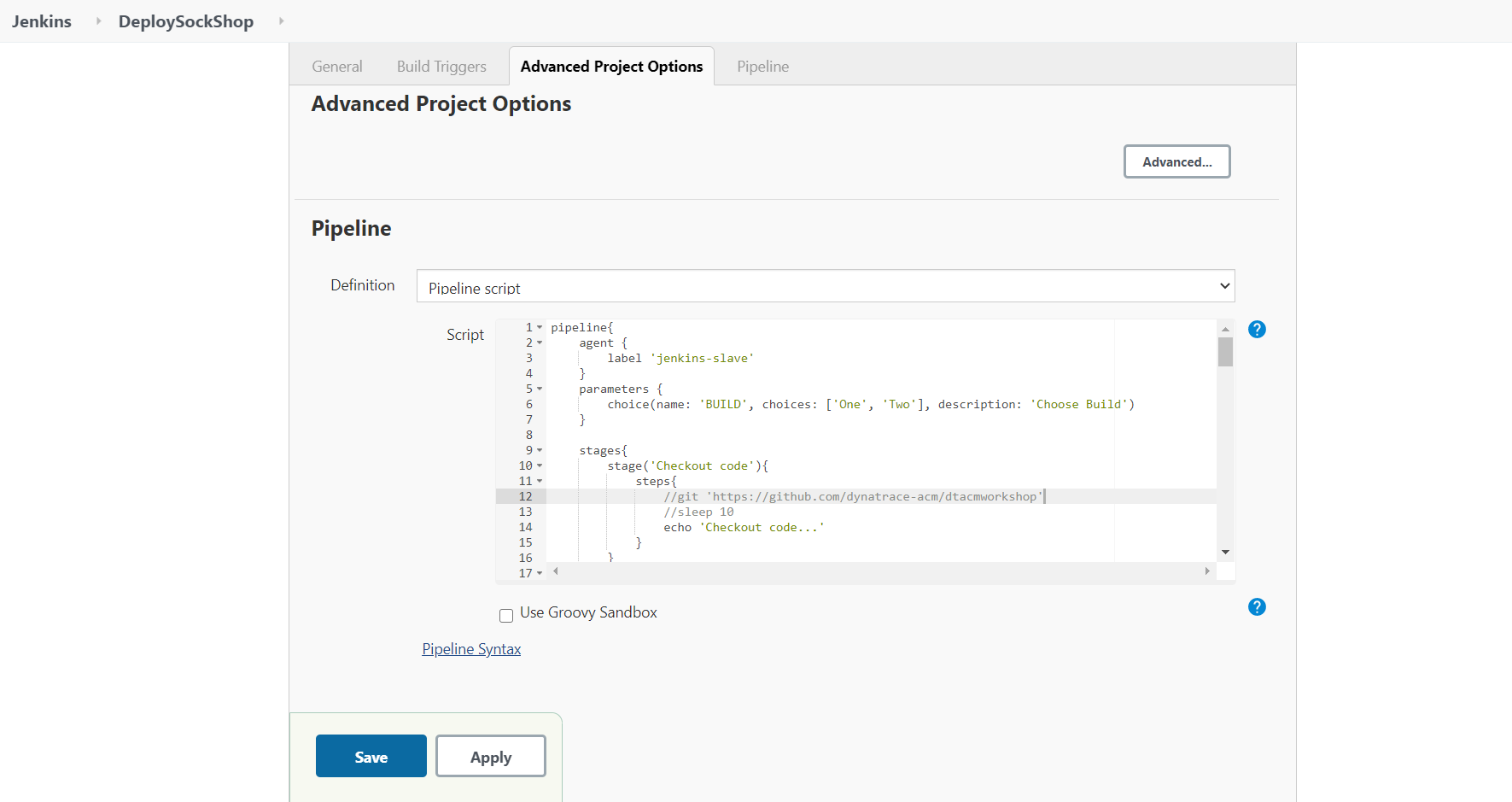


* 1. Create a New project by clicking on New Item. Select pipeline. Give name as DeploySockshop



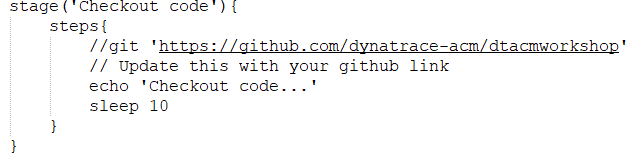


* 1. This creates a new project as show below 
  2. Select DeploySockshop and click configure in left side. Go to pipeline section and add the script in **PipelineWithIntegration.txt** and save.



In the script, update github repo link and update the jmeter folder

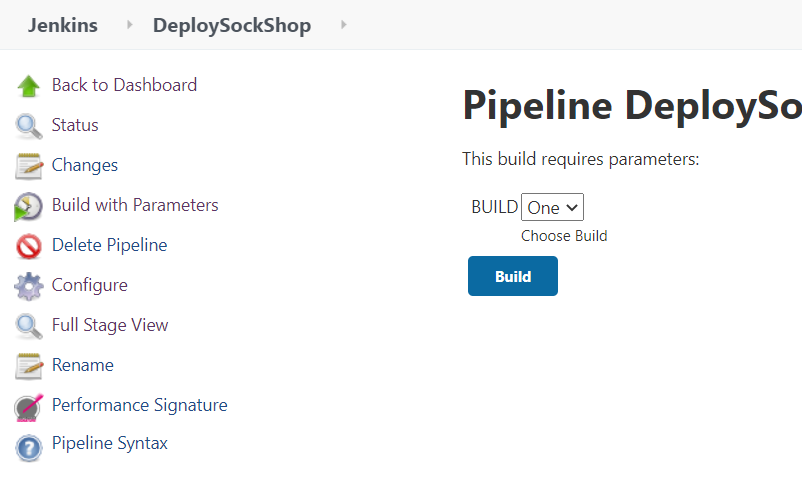
**Github repository link**



**JMeter path**



* 1. Click on “Build with Parameters”. Select option one and click on build. Next select option two and click on build. Build one is clean build. Build Two is bad build and fails in Quality Gate.



g. Image of a clean and bad build

