Integrating Jenkins with Power Automate to call Jenkins API

November 18, 2022 November 20, 2022 Chris Edit

Objectives:

To call the Internal Jenkins API using Microsoft Power Automate Flows

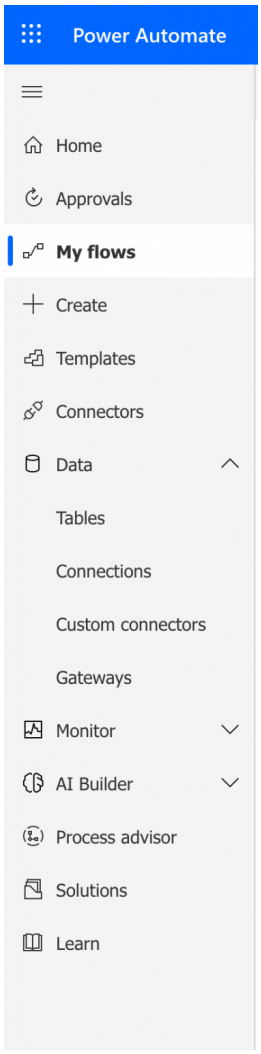
This article will help you to set up a continuous integration environment using Jenkins for and Microsoft Power Automate.

Building the Jenkins Custom Connector

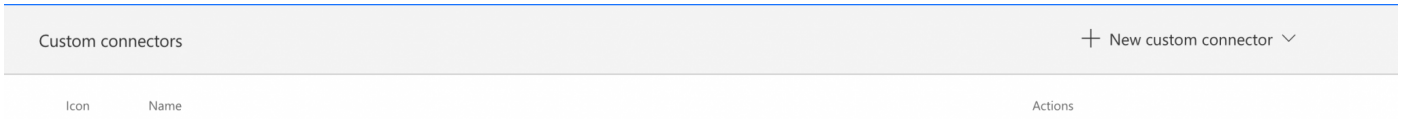
Let's start with the assumption that you have an In-house Jenkins server sitting somewhere in your Infrastructure stack.

Power Automate has an On-premises gateway you can install on Windows. The Idea is to Install the Gateway on a Windows host that has network access to the Jenkins box. Microsoft has a pretty decent Article on this here
Once Installed & configured, we can head over to https://make.powerautomate.com

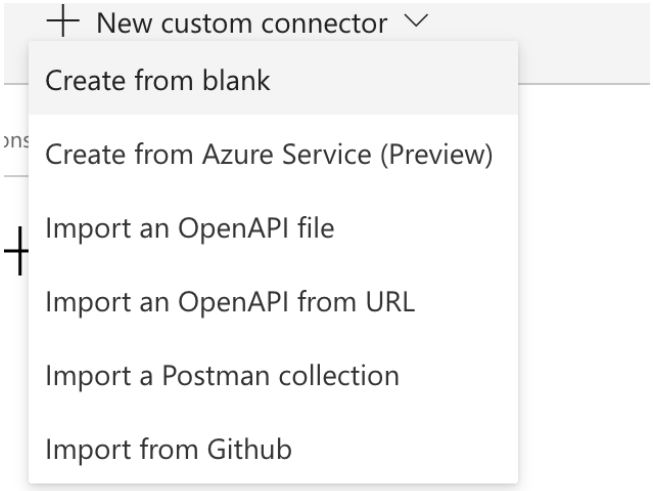
On the Left Menu, expand Data.



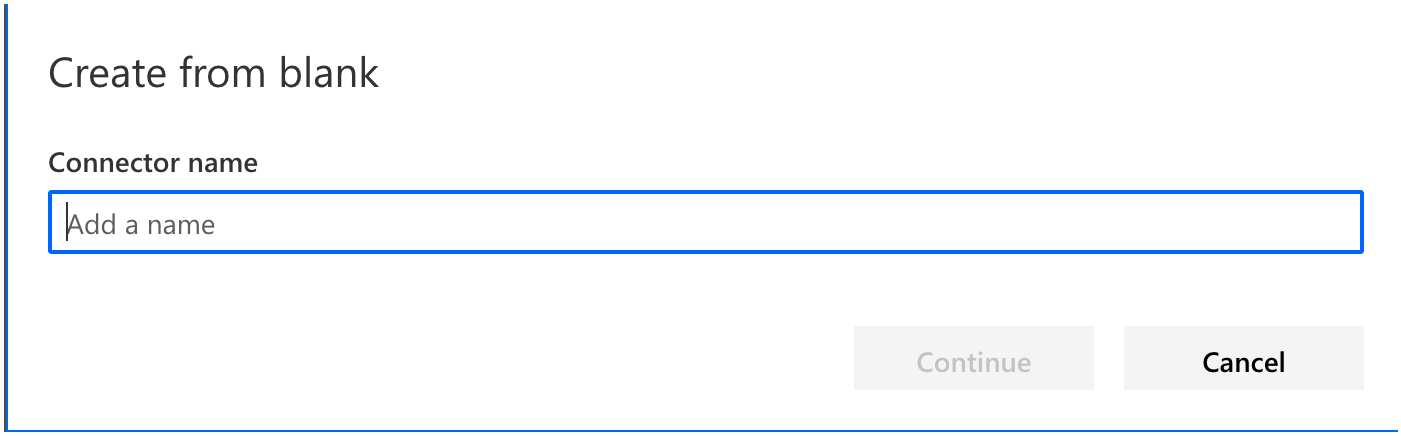
Now Click on "Custom connectors"



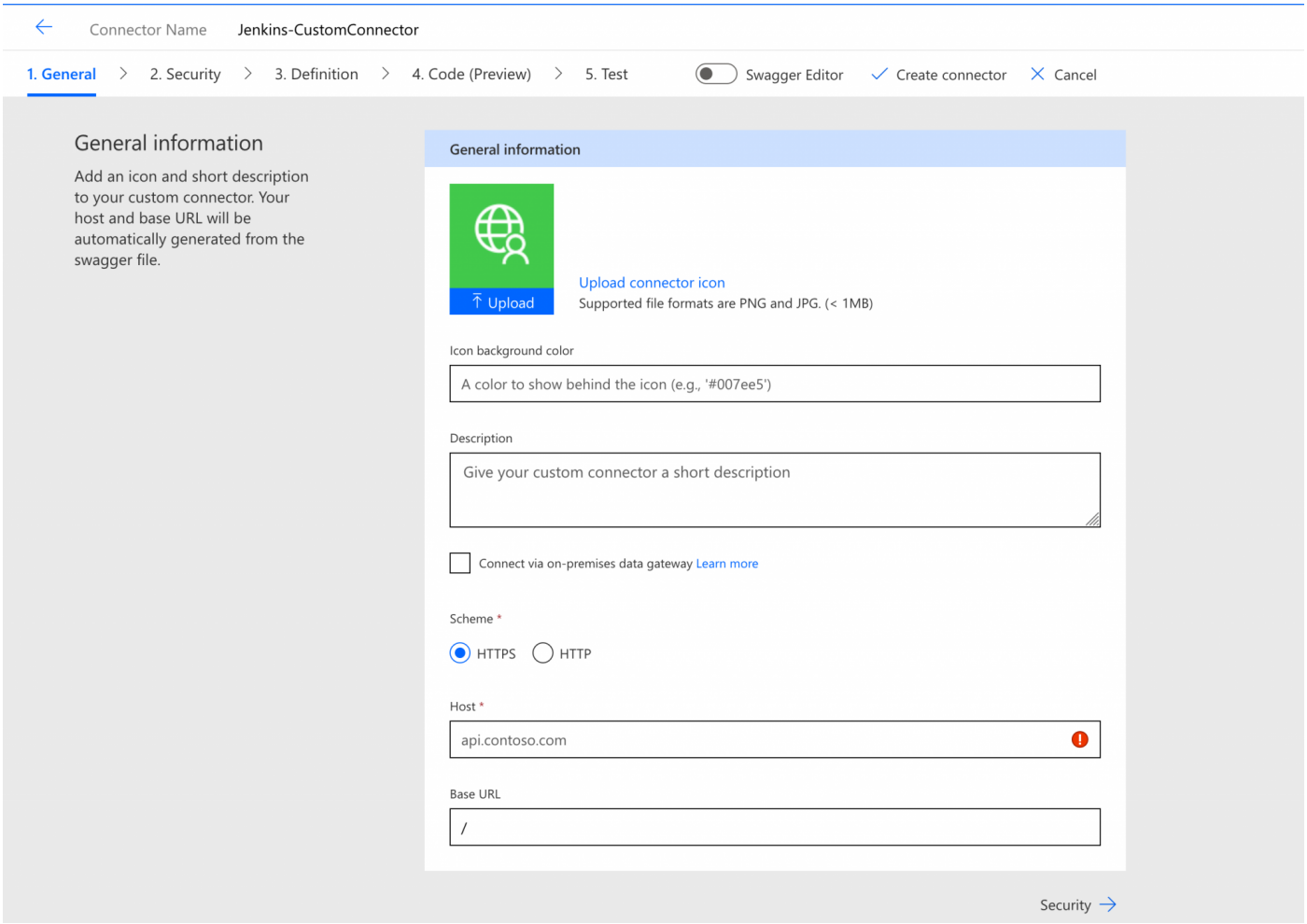
Click "New custom connector"



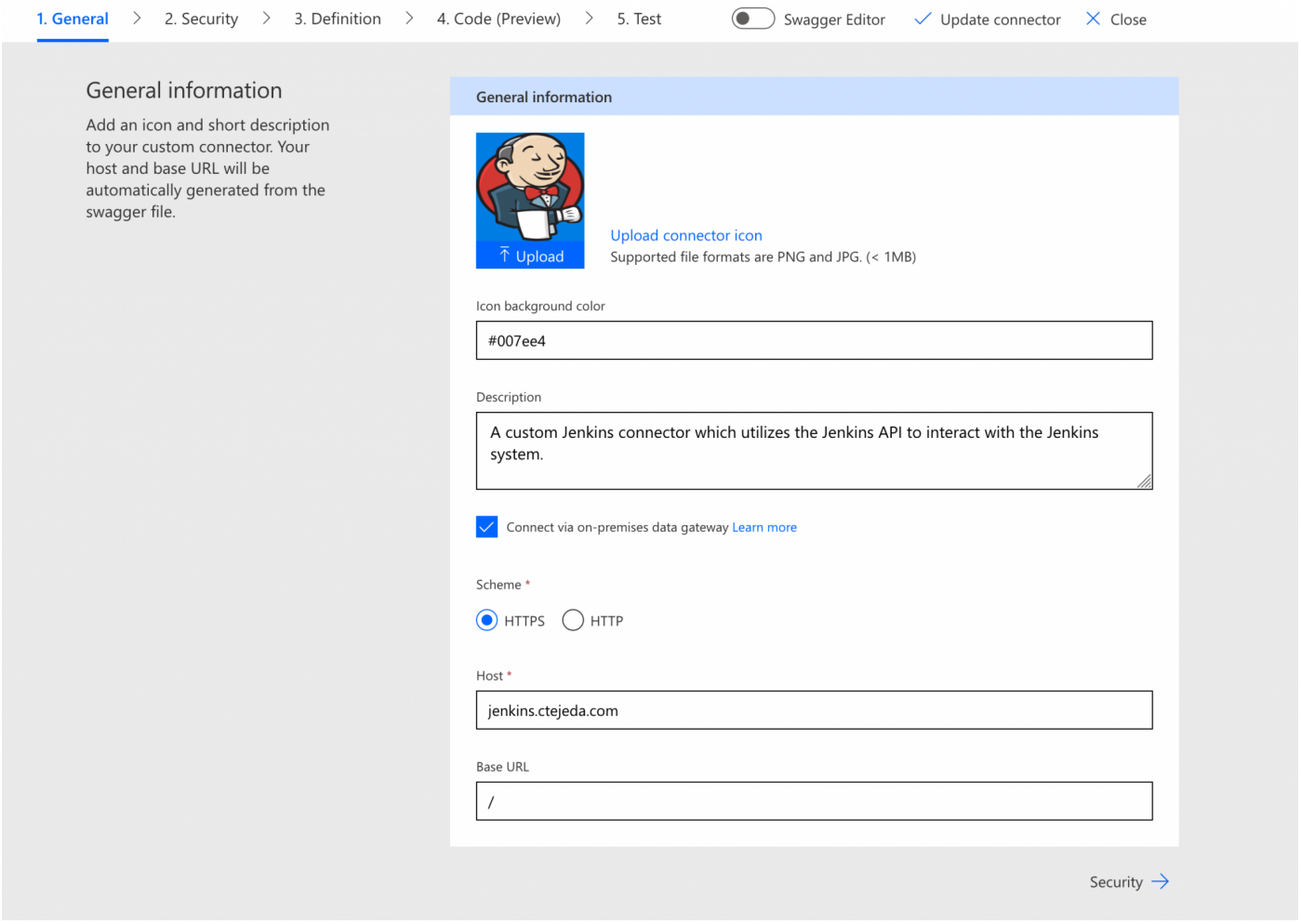
Choose “Create from blank”



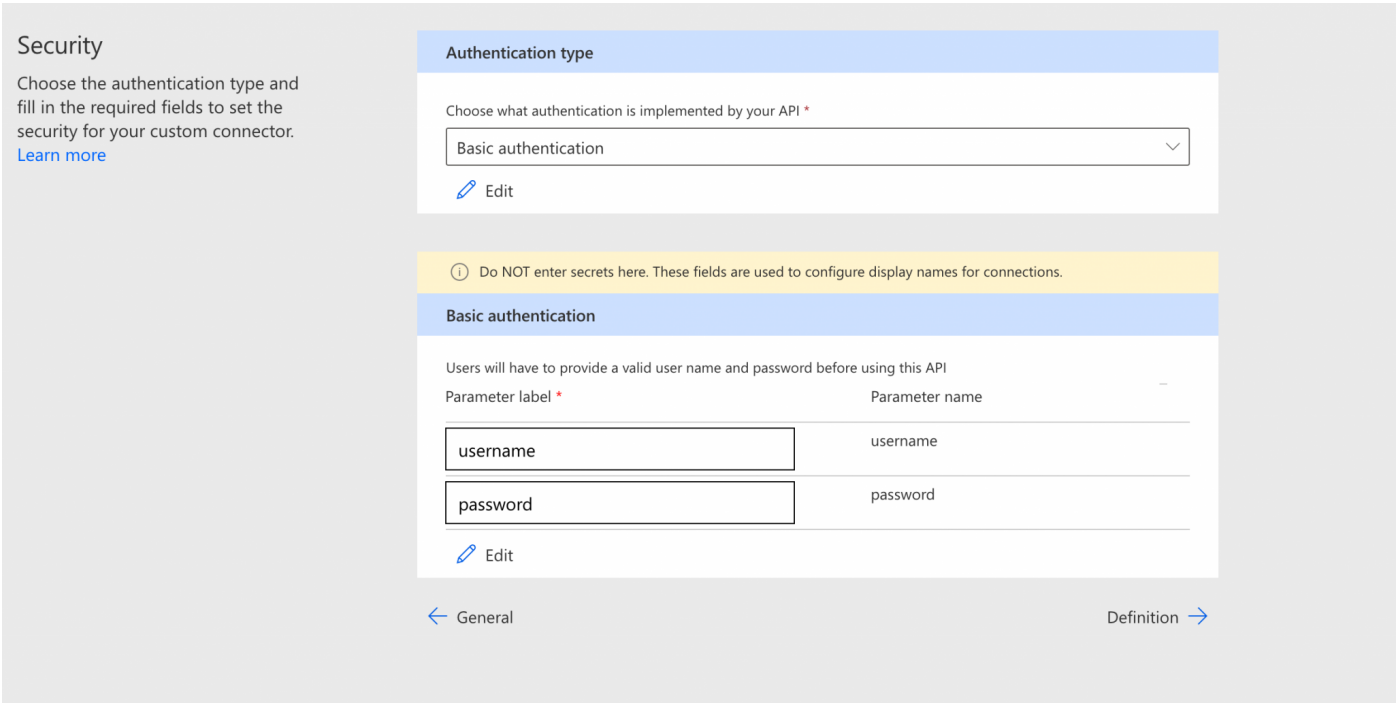
Give your Custom connector a name. I gave my custom connector the name “Jenkins-CustomConnector”
Once you click Continue, you should see the below.



The First Option allows you to choose an **Icon** for your Custom Connector.
You have the option to choose a **background** color.
Give your custom connector a **description**.
Tick the "**Connect via on-premises data gateway**" option.
Choose your API **scheme**.
Populate the Jenkins API **hostname**. Example: jenkins.domain.com

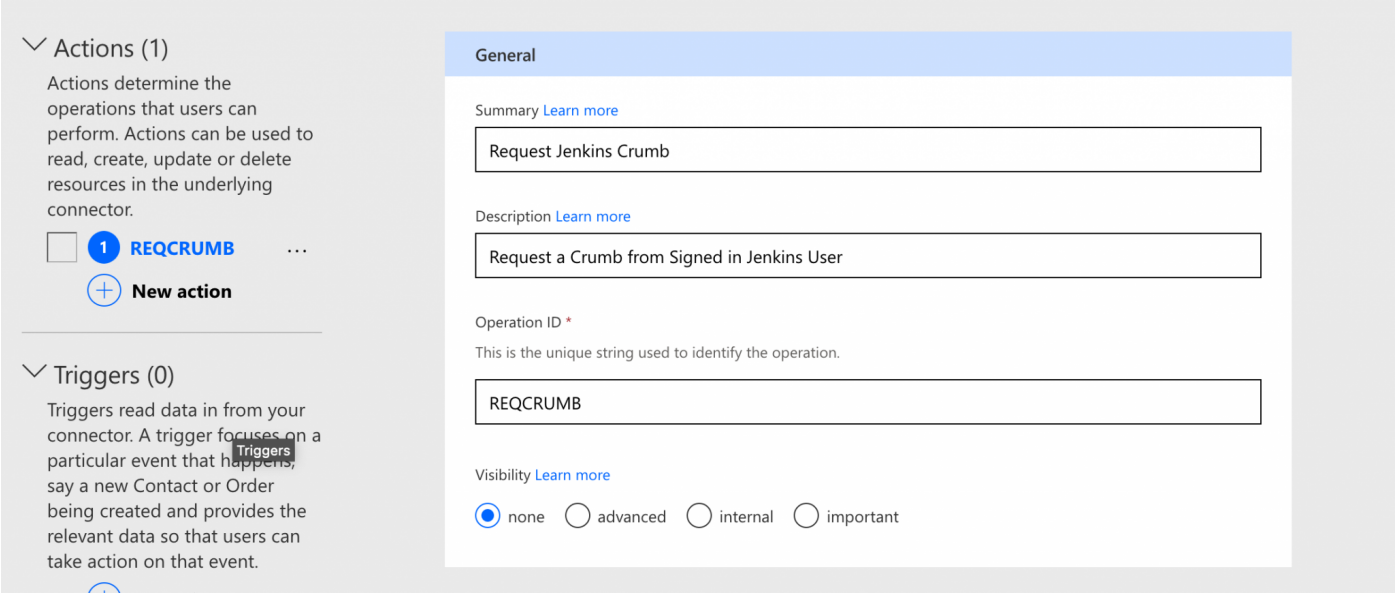


The Next page allows you to choose the authentication type for the API call to Jenkins from the on-premises gateway to on-premises Jenkins box. We can choose "Basic authentication"
Enter the parameters names. username for the **username** field and password for the **password** field.

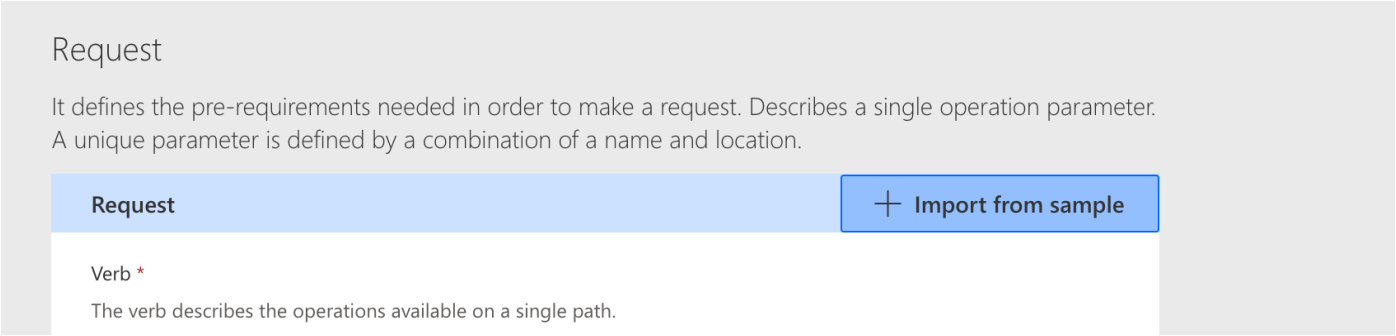


The Next page is called the Definition page. Here you're able to define your Action and triggers.
for the purpose of this article, we will be creating two actions.
first 1 for the Jenkins CRUMB request
second one to start a Jenkins build.

For the Jenkins CRUMB, click “New Action”
In the **Summary** enter something that makes sense like “Request Jenkins Crumb”
Description: Request a Crumb from Signed in Jenkins User
for the **Operation ID** I used: REQCRUMB

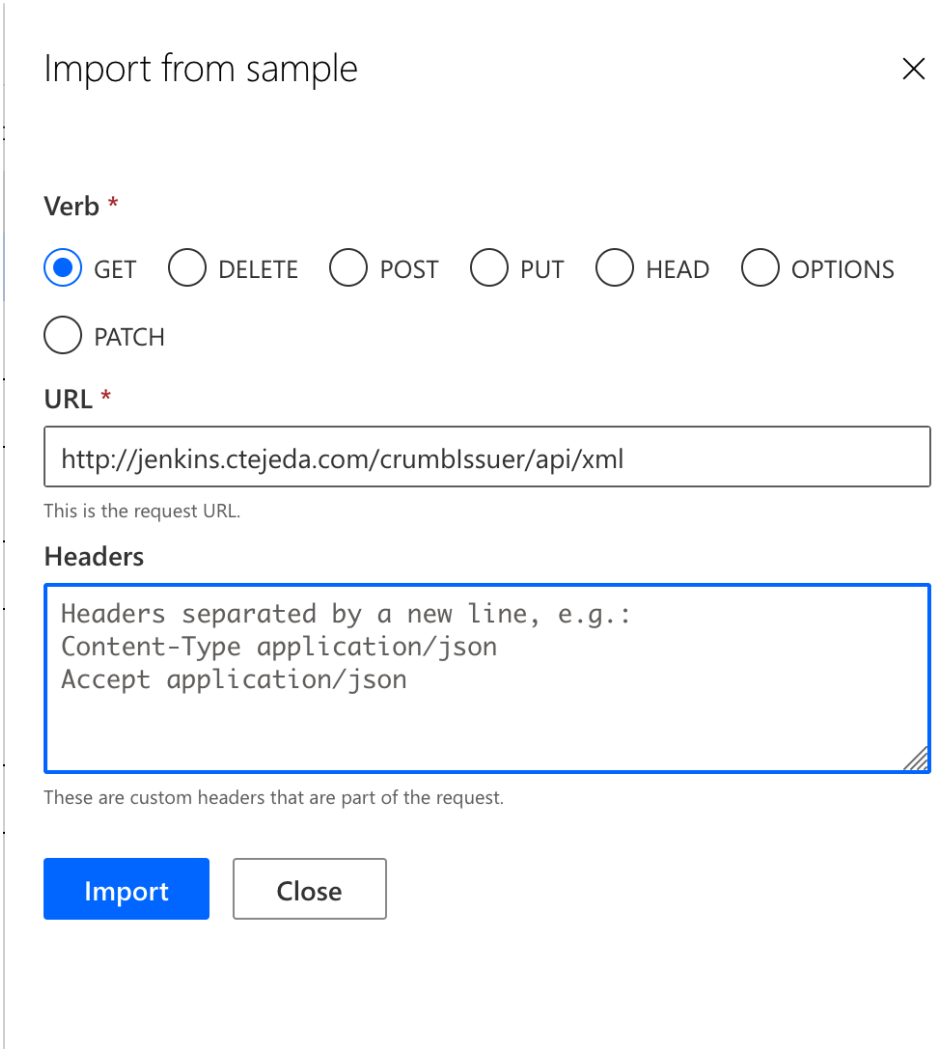


Scroll down to the “Request” section. You should see the “Import from sample” option. Let’s Click on that.



The Import from sample window should appear an you will see an option to select an API verb. Select “GET” and for the URL paste the Internal API call to request a Jenkins Crumb. For me it’s http://jenkins.ctejeda.com/crumbIssuer/api/xml

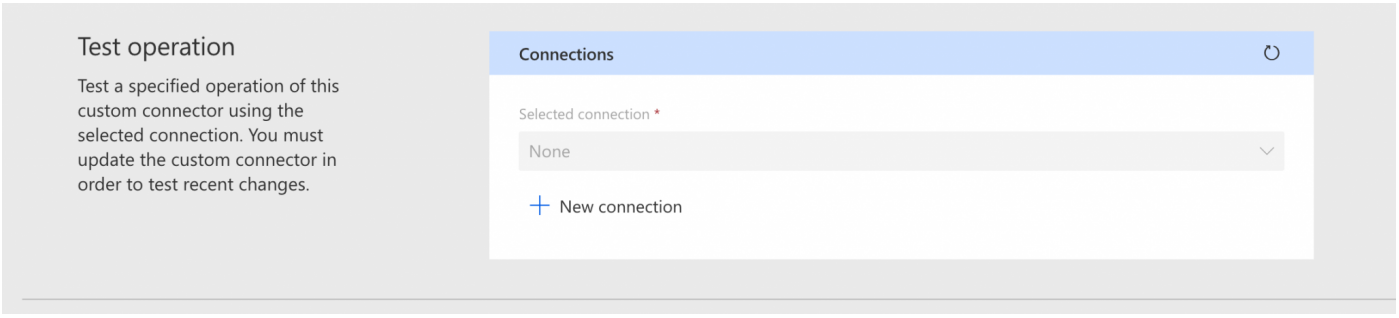
My sample setting look like this:



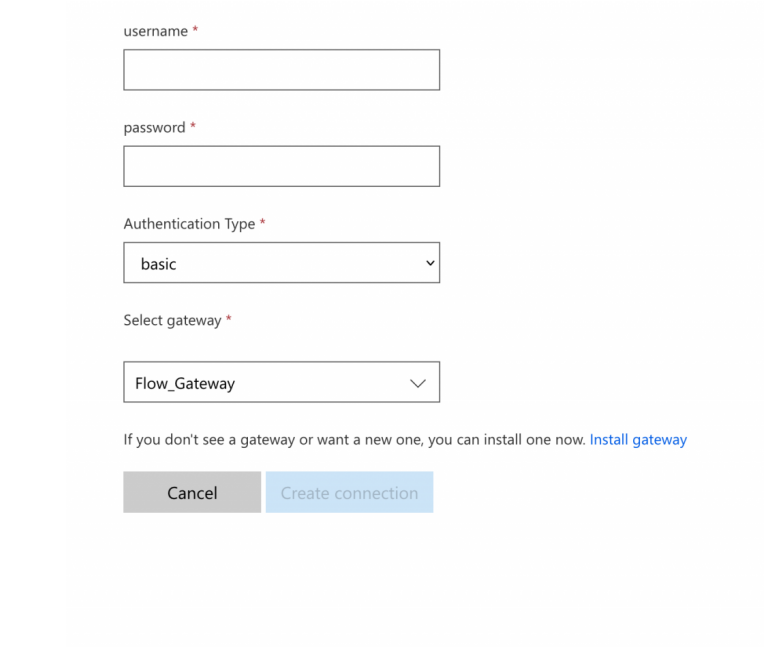
Click import.

At this point you may choose to test an confirm if you are able to retrieve a response that shows the Jenkins Crumb in the API response.

Click on the “Test”

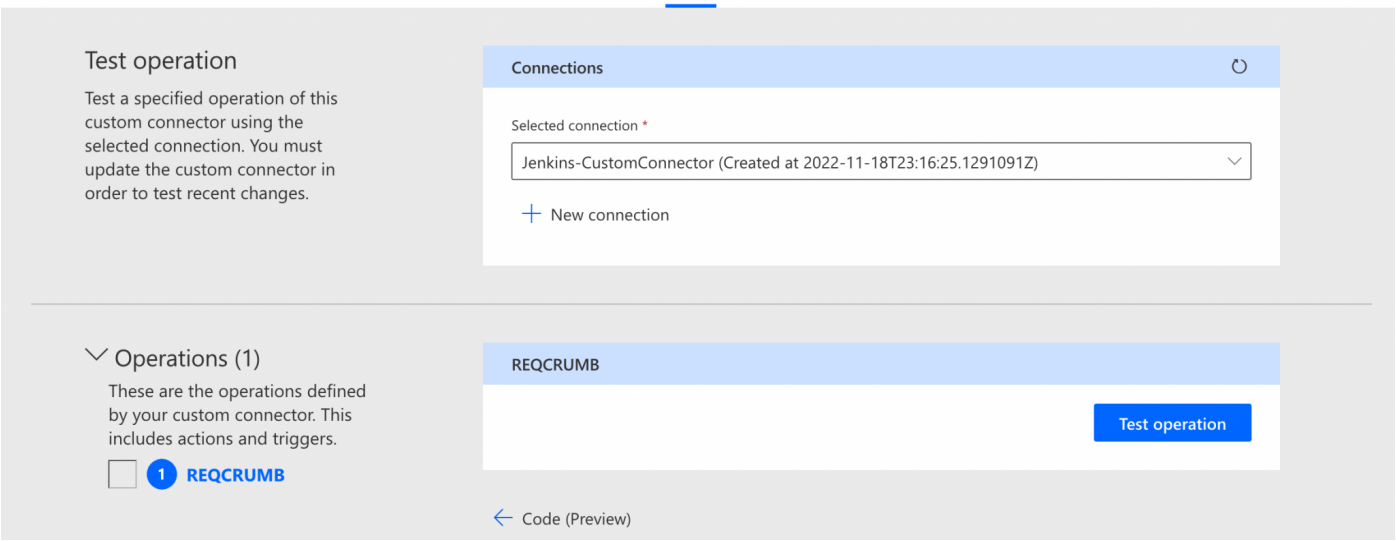


Click on “New connection”
You will be redirected to a new page to create a new secure connection.

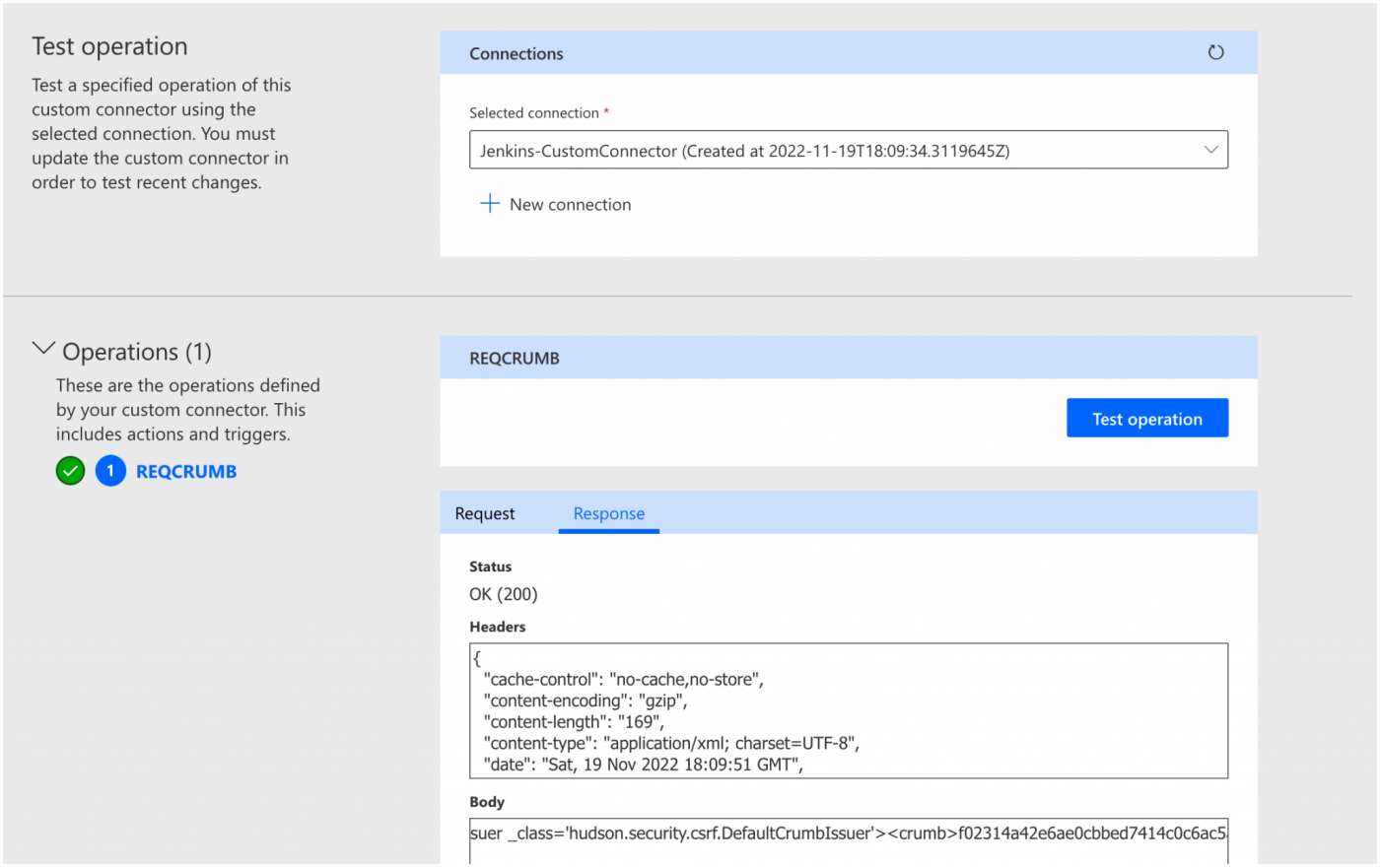


Populate the required fields
Username = Jenkins username
Password = Jenkins API Token. ***Note**: To Create the **API TOKEN**, In Jenkins go to: Accounts icon -> configure -> API Token -> Add New token*
Authentication Type = Basic
Select Gateway = Select the Gateway you installed earlier
click “Create connection”

The Test page should now have the connection you created as an option.

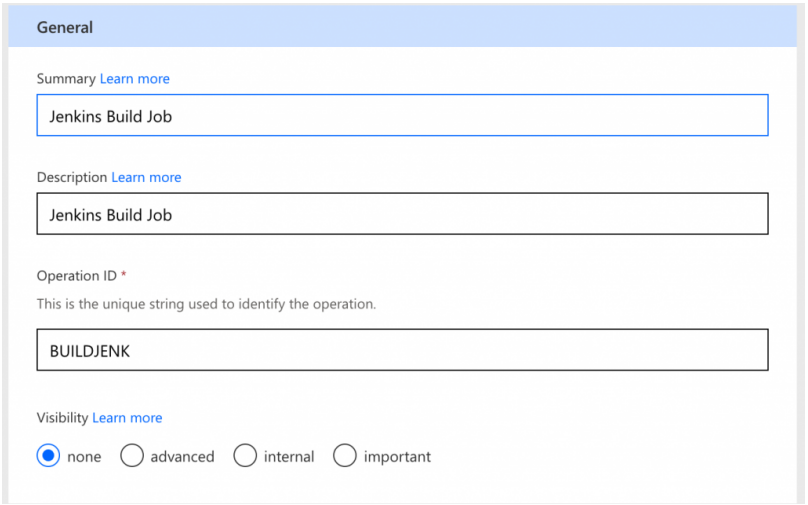


Click on the Test Operation button to test the Request CRUMB action we built. If successful, you should see a 200 status response, and the body will contain the crumb like below.



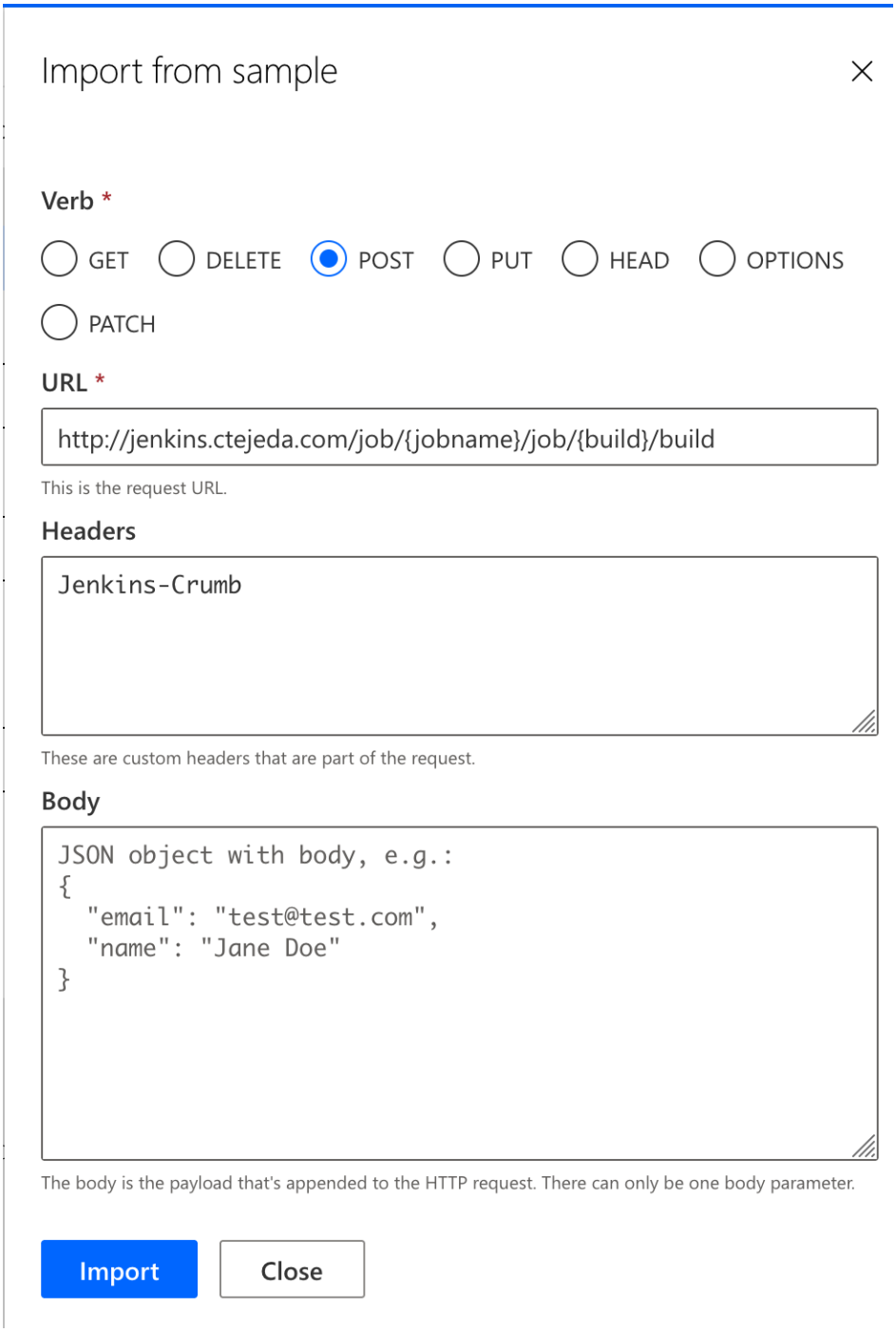
Click on the Definitions tab to create the second Jenkins Action.
Click “New action”

Summary: Jenkins Build Job
Description: Jenkins Build Job
Operation ID: BUILDJENK

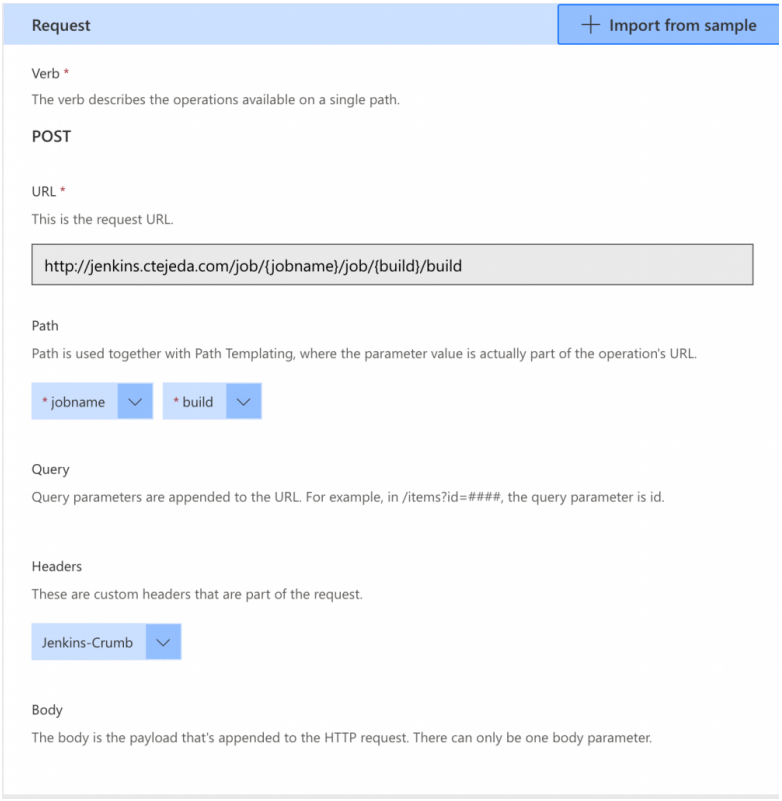


Click “Import from Sample”
API **Verb**: POST
URL: http://jenkins.yourdomain.com/job/{jobname}/job/{build}/build
the above values inside of the curly brackets serve as variables
Header: Jenkins-Crumb

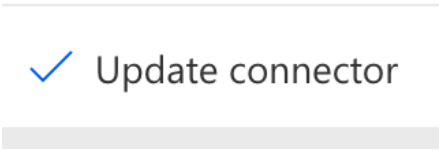
The Import from sample should look similar to this:



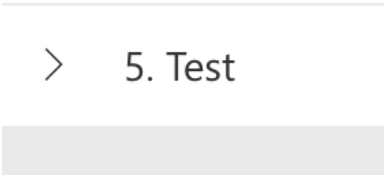
Click Import
The import should generate a Post API similar to this:



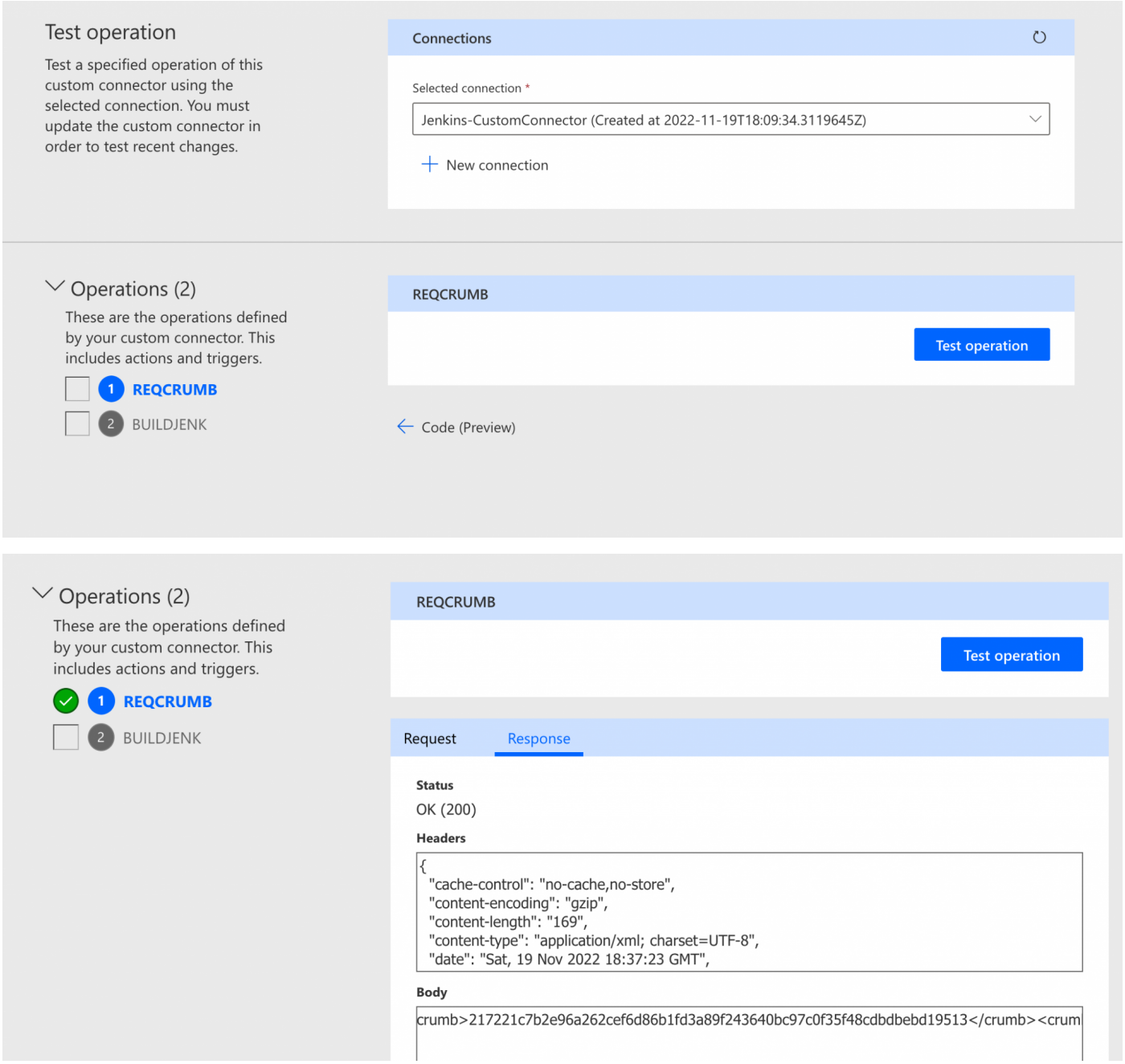
Now we can call this custom [POST] Jenkins Build API using the CRUMB value we get from the custom [GET] REQCRUMB API. Let's test it.
Save your changes by clicking "Update connector"



Click on the Test Tab.

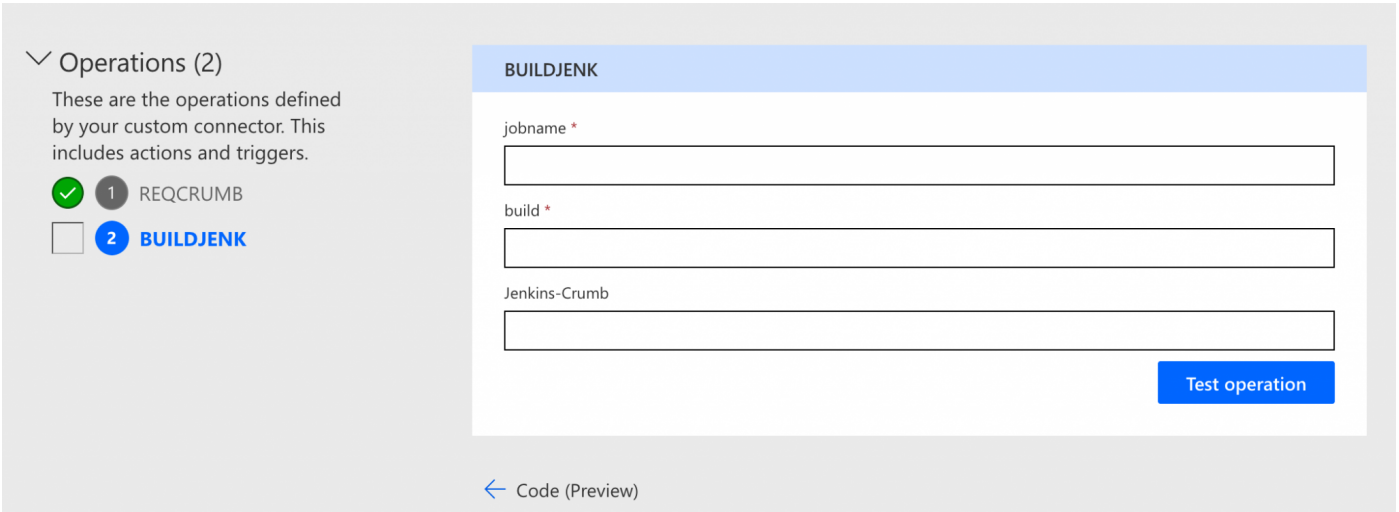


If you've already created a connection, with the REQCRUMB operation selected, click "Test Operation"

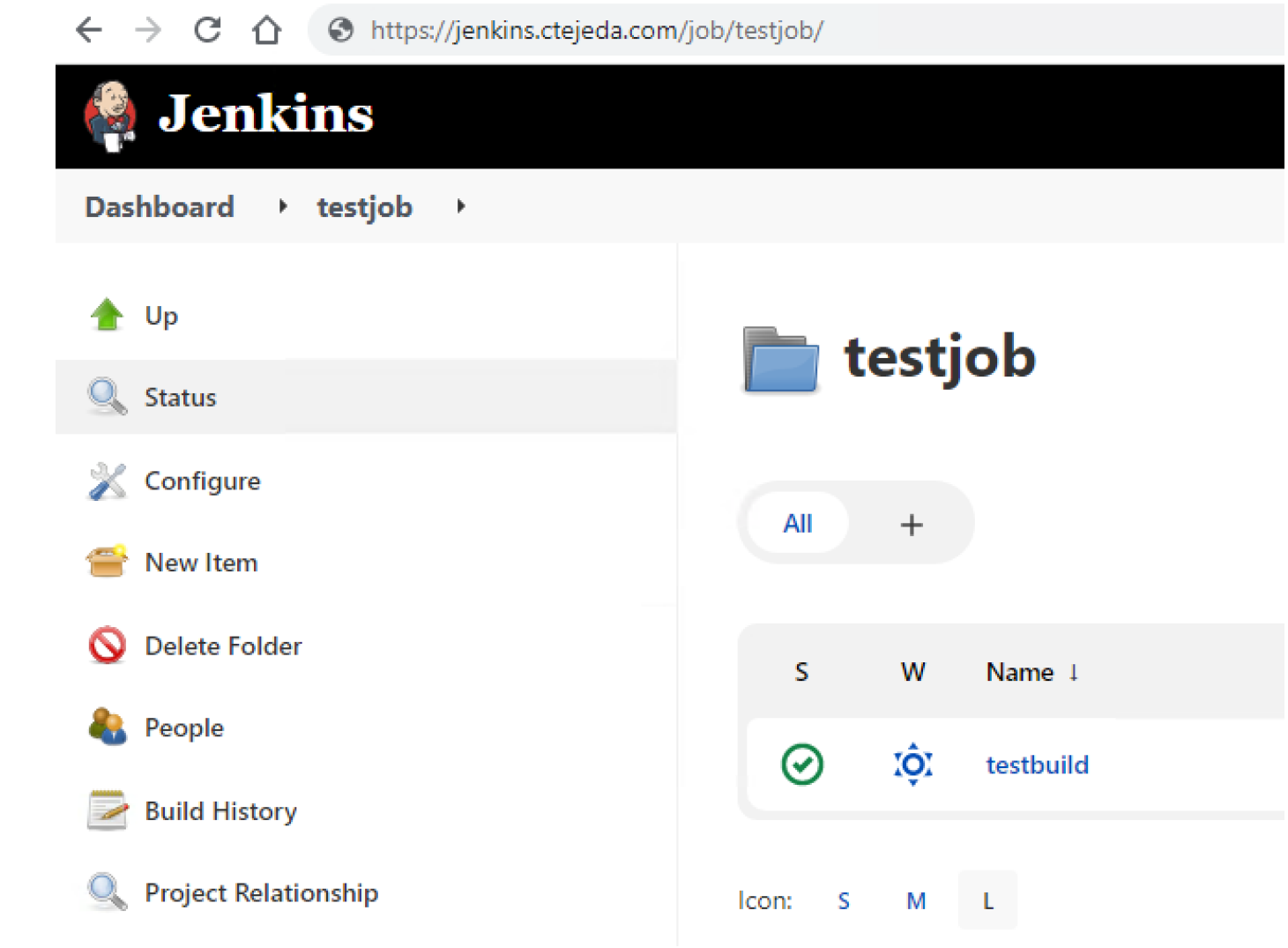


Copy the Jenkins Crumb from the response. It should be all of the values between `<crumb> </crumb>`

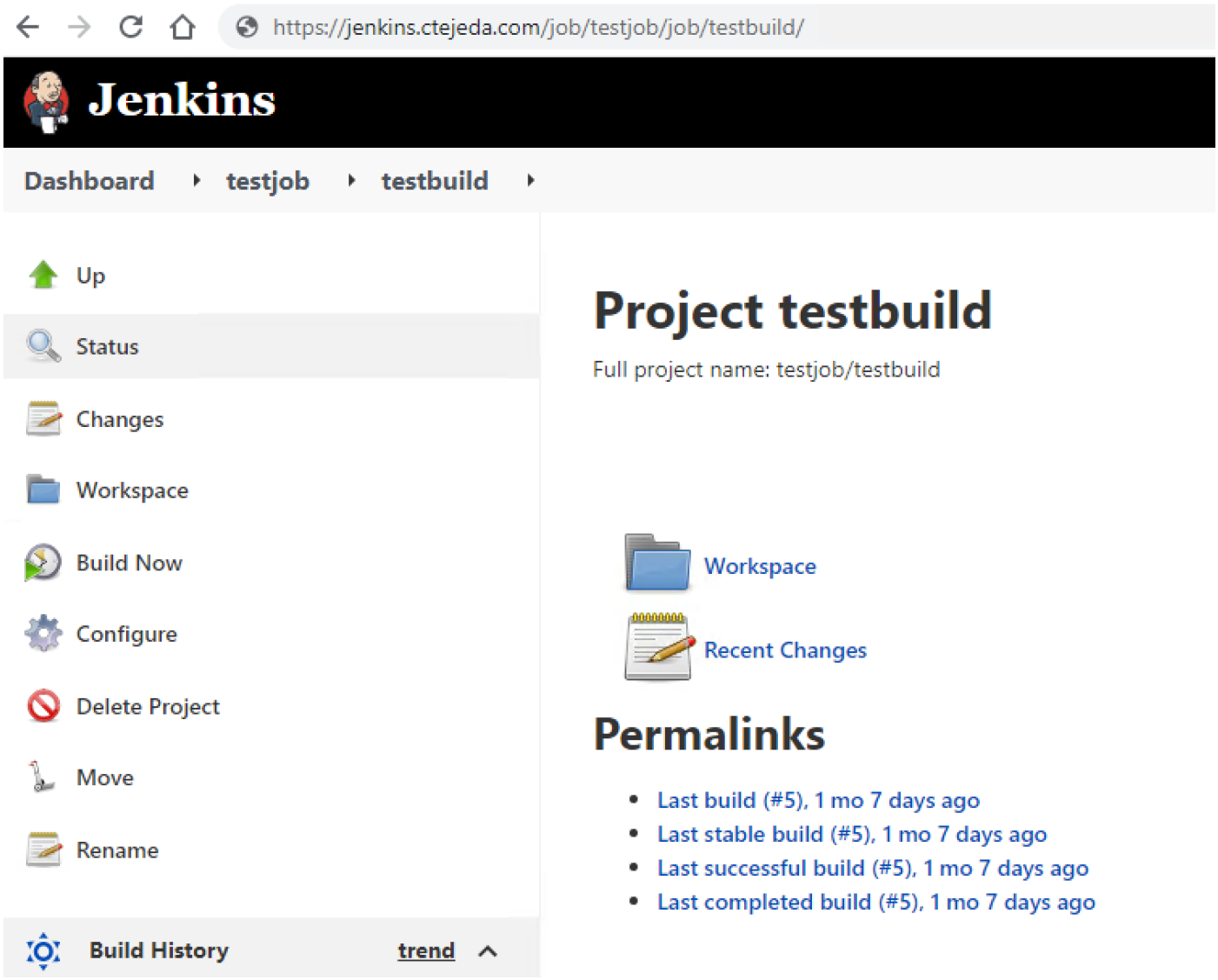
Now select the BUILDJENK operation.



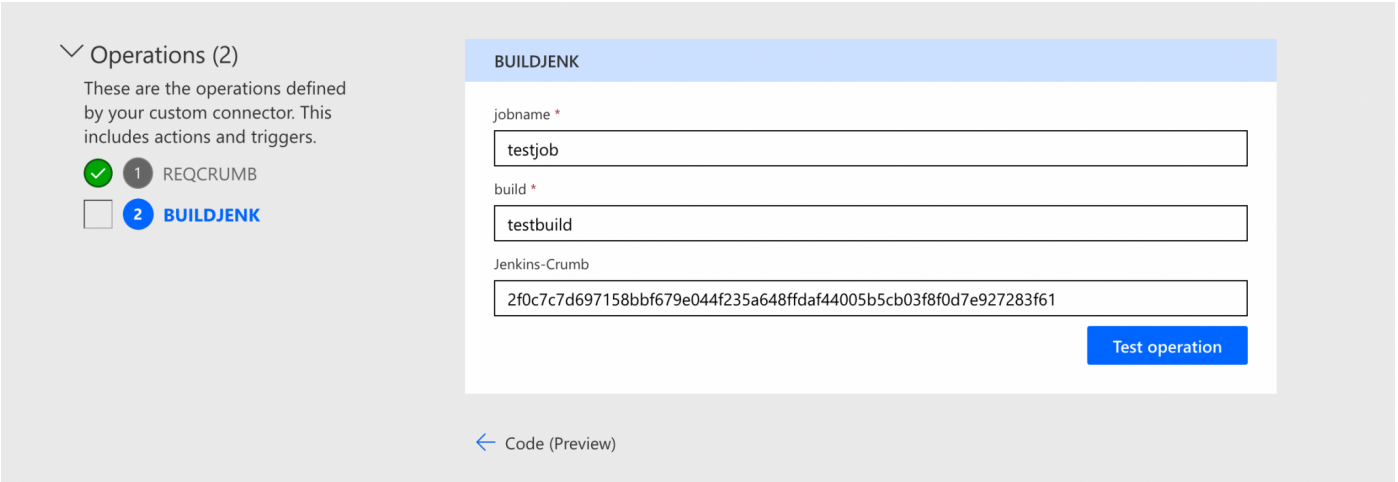
Jobname: enter your Jenkins Job name. You can usually find this in the address bar when accessing Jenkins Job over the web.



Build: enter the build name or number. You can usually find this in the address bar when accessing the Jenkins build over the web.

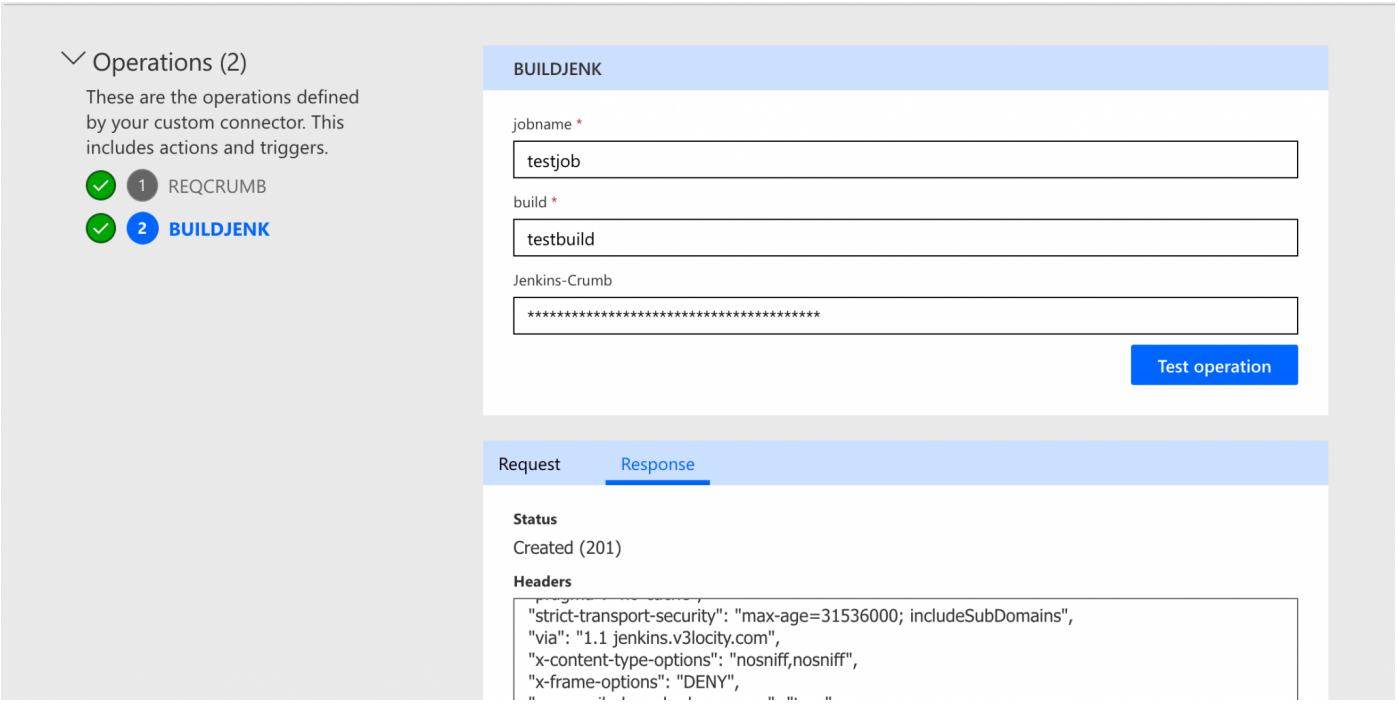


Jenkins-Crumb: Enter the Jenkins Crumb you copied here.
Your BUILDJENK operation should look similar to this:

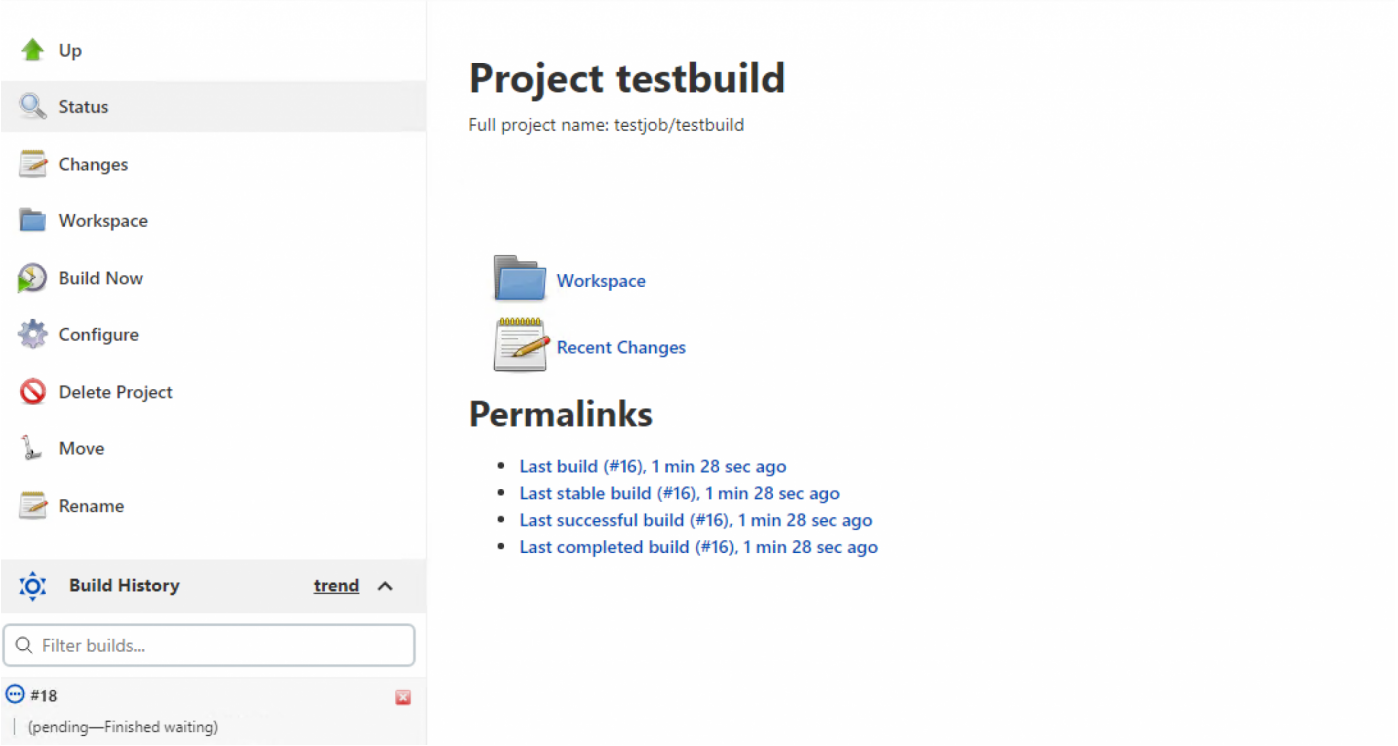


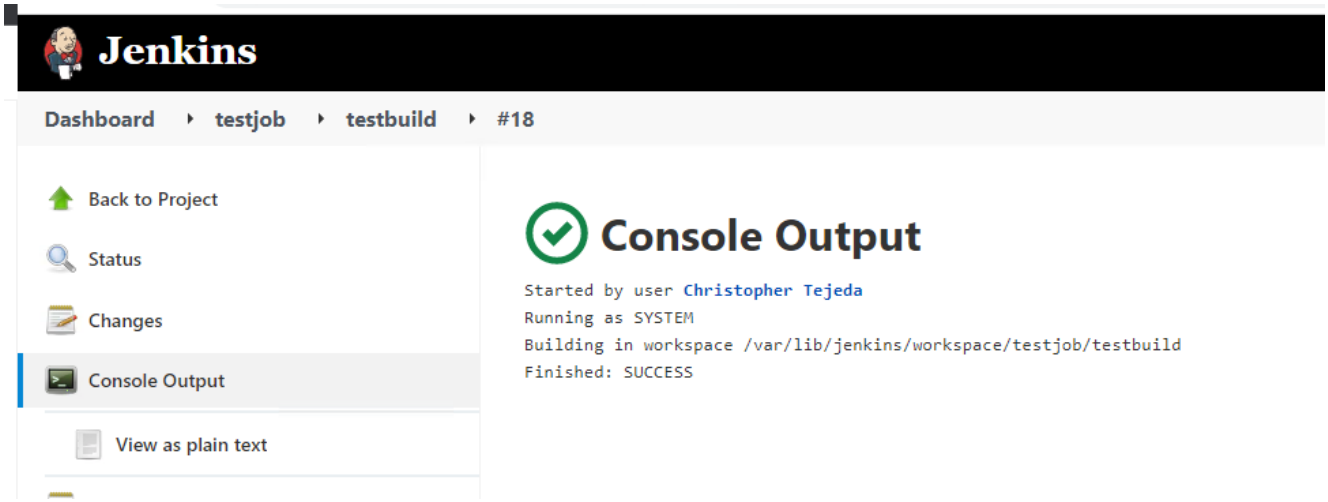
Click on “Test Operation”

You should get a Created (201) response similar to the one below.



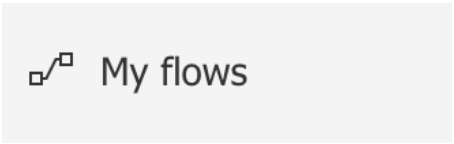
As a result, You should see the Jenkins Job building.



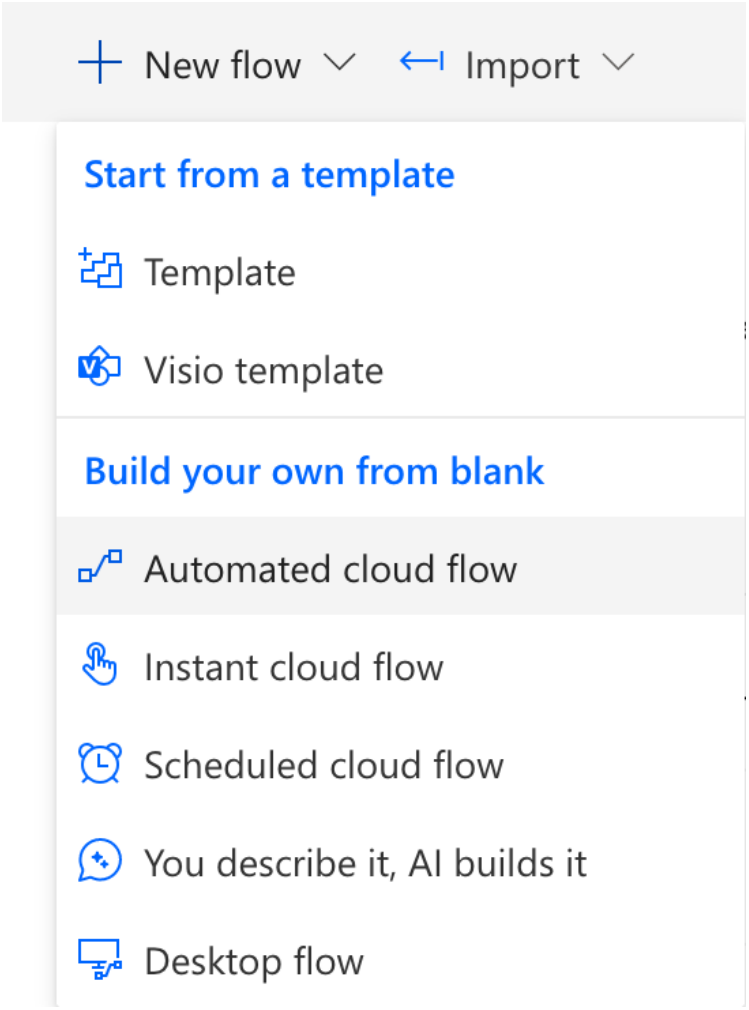


Building the Power Automate Flow

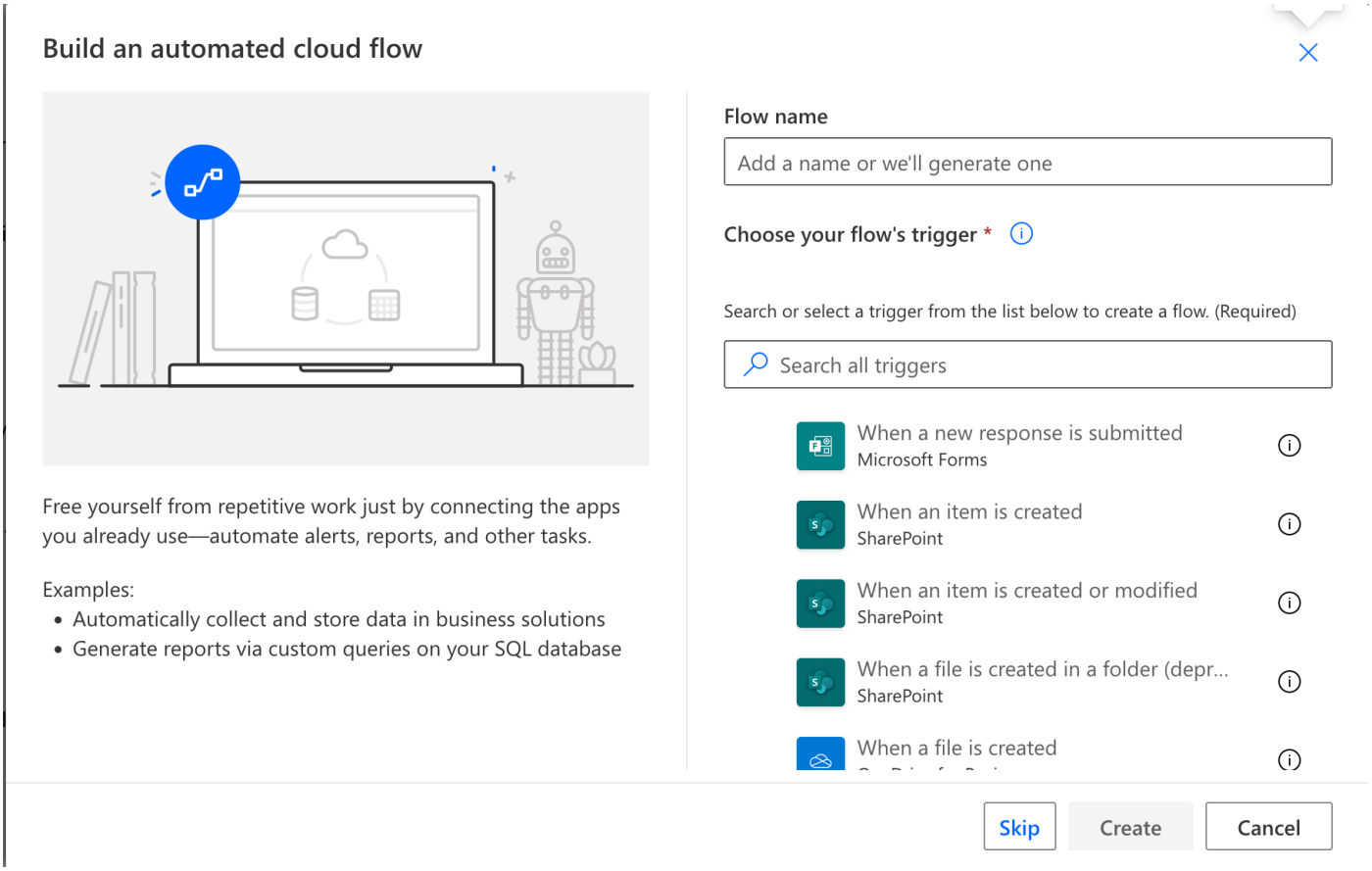
Now we can start to build out our Power Automate Flow.
Click on My flows in the left Menu.



Click on “New flow” and choose “Automated cloud flow”

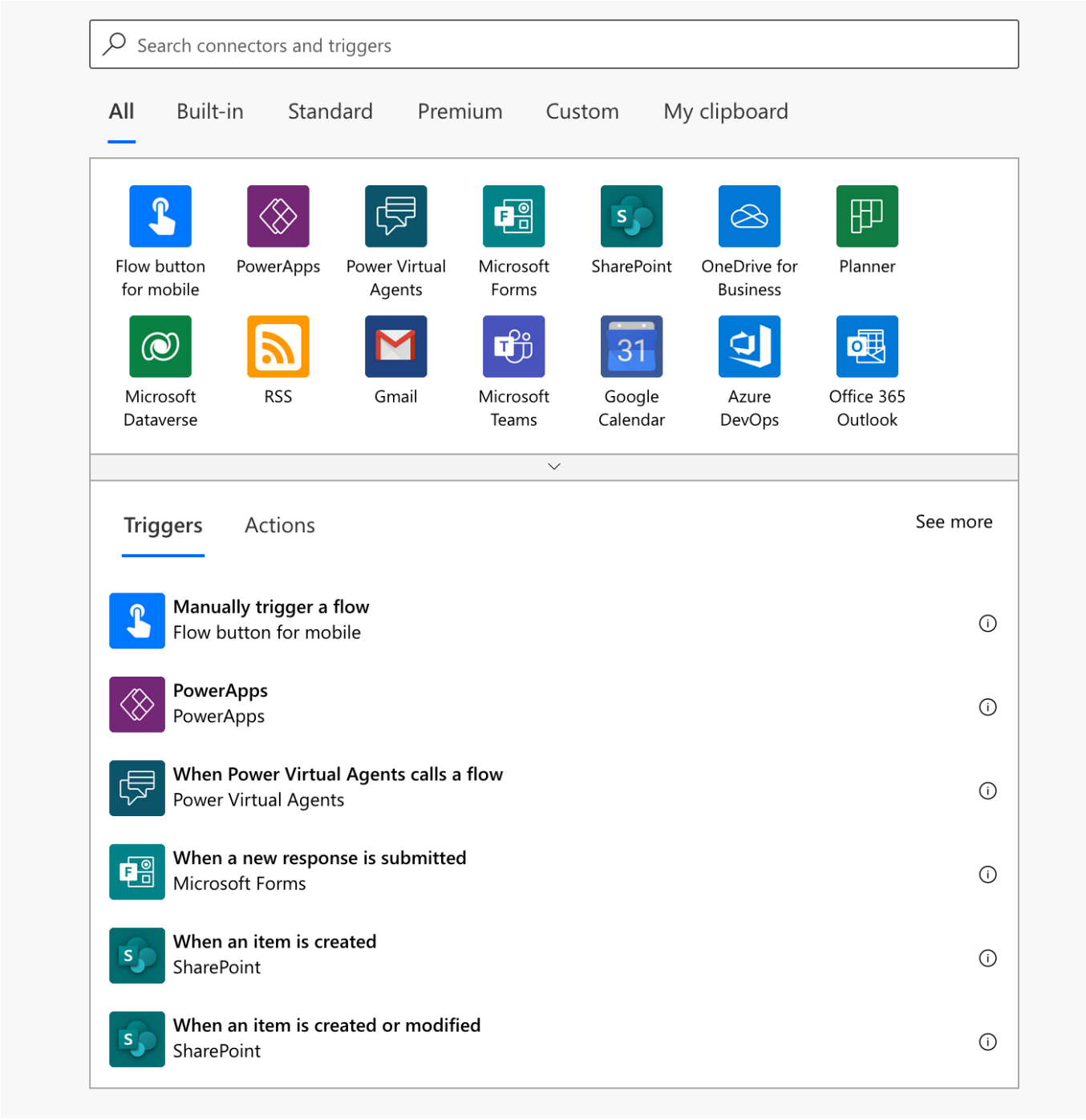


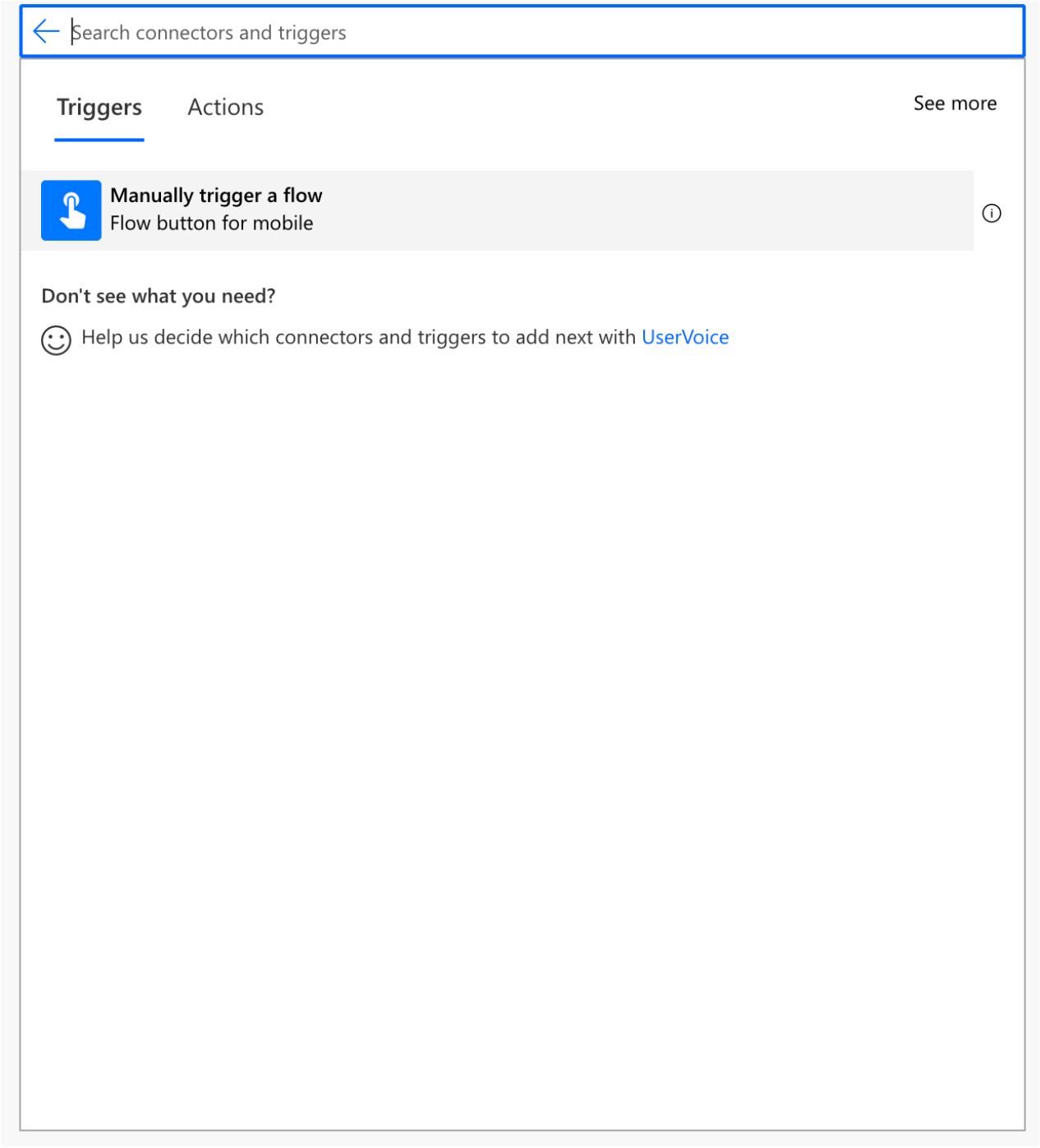
the “Build an automated cloud flow” window will appear.

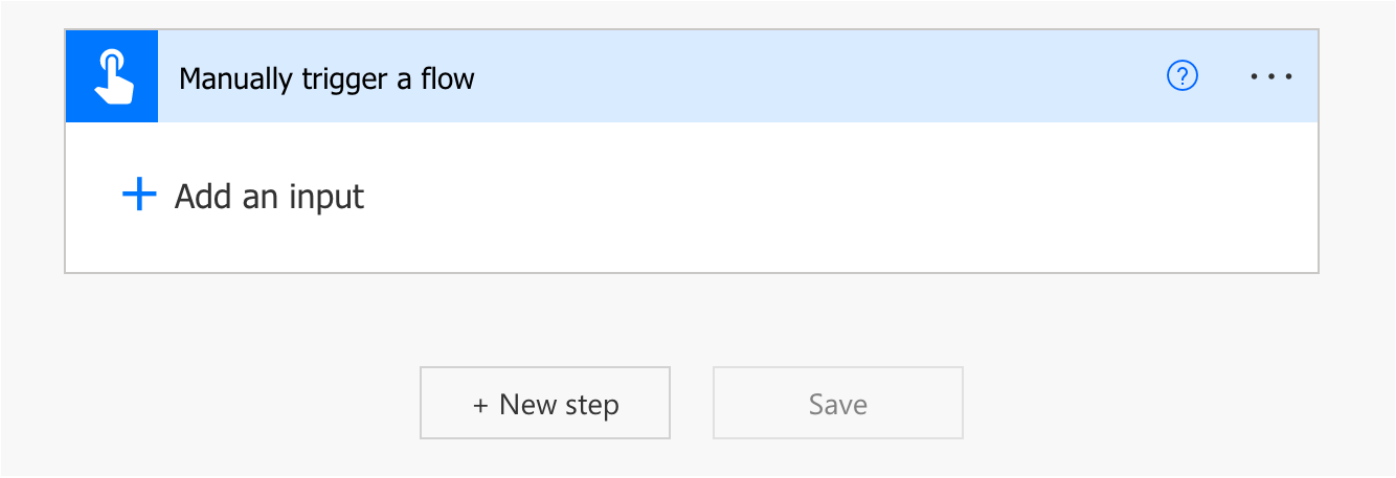


click on “skip”

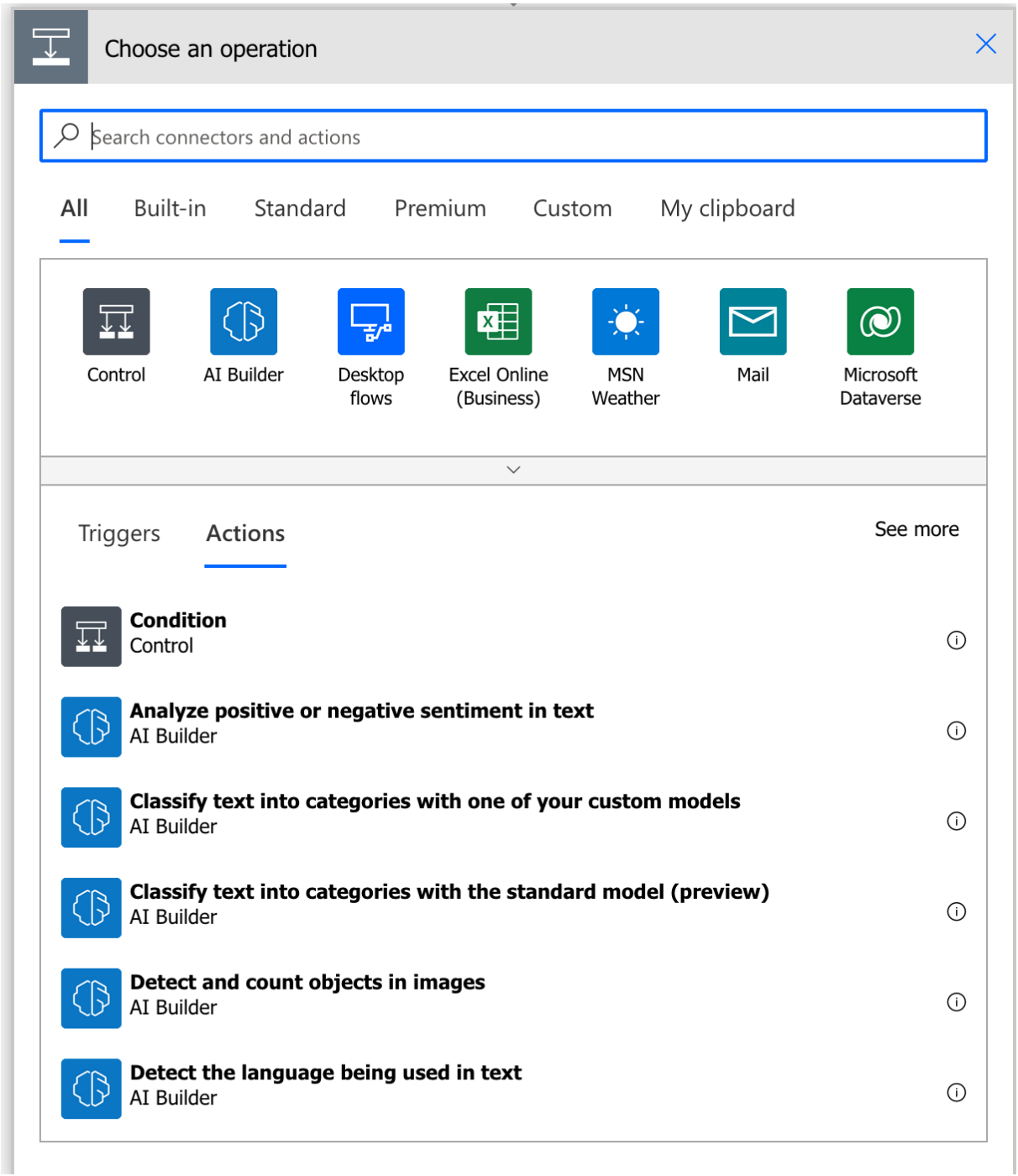
The first screen you will see is an option to choose a trigger for your flow. for the purpose if this article, I will choose the “Flow button for mobile” to test.


Search connectors and triggers
All
Built-in
Standard
Premium
Custom
My clipboard
Flow button for mobile
PowerApps
Power Virtual Agents
Microsoft Forms
SharePoint
OneDrive for Business
Planner
Microsoft Dataverse
RSS
Gmail
Microsoft Teams
Google Calendar
Azure DevOps
Office 365 Outlook
Triggers
Actions
See more
Manually trigger a flow
Flow button for mobile
PowerApps
PowerApps
When Power Virtual Agents calls a flow
Power Virtual Agents
When a new response is submitted
Microsoft Forms
When an item is created
SharePoint
When an item is created or modified
SharePoint

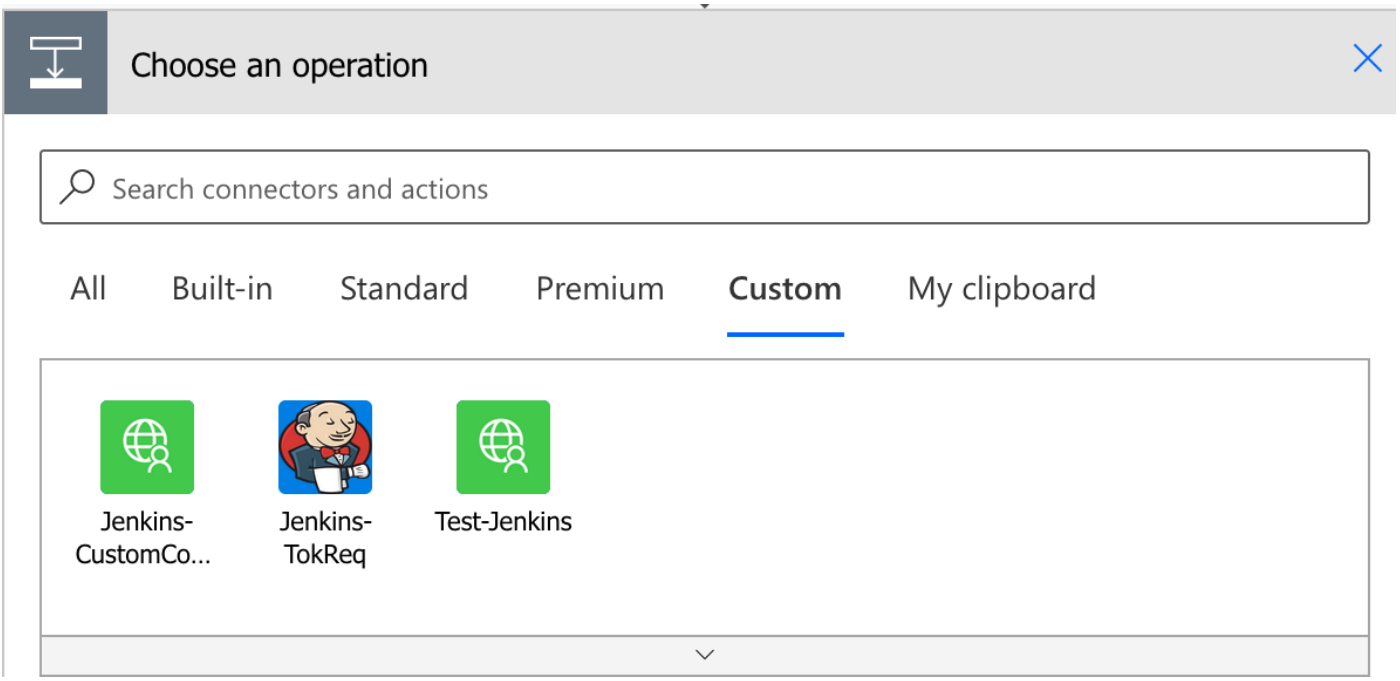




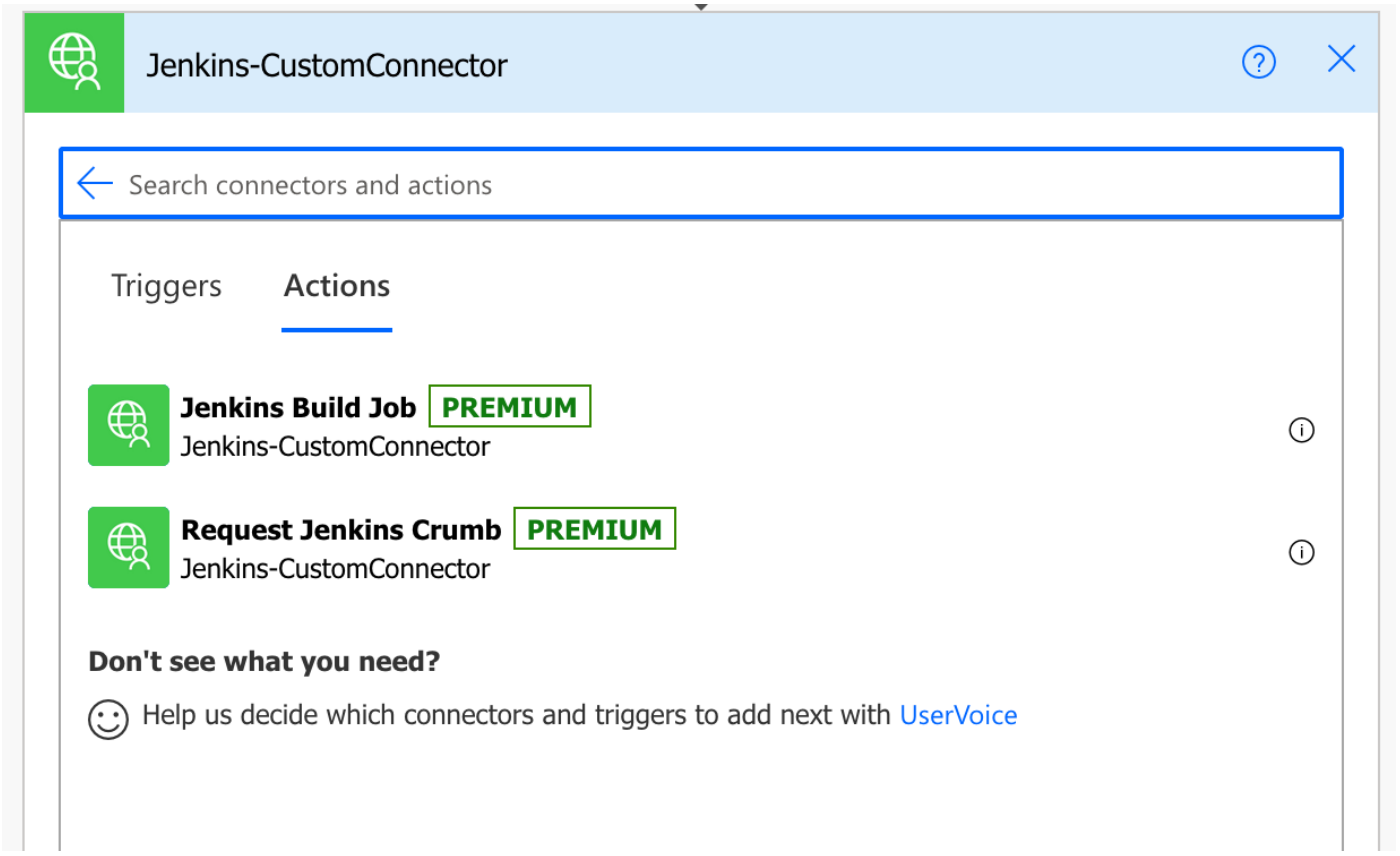
Click Next step



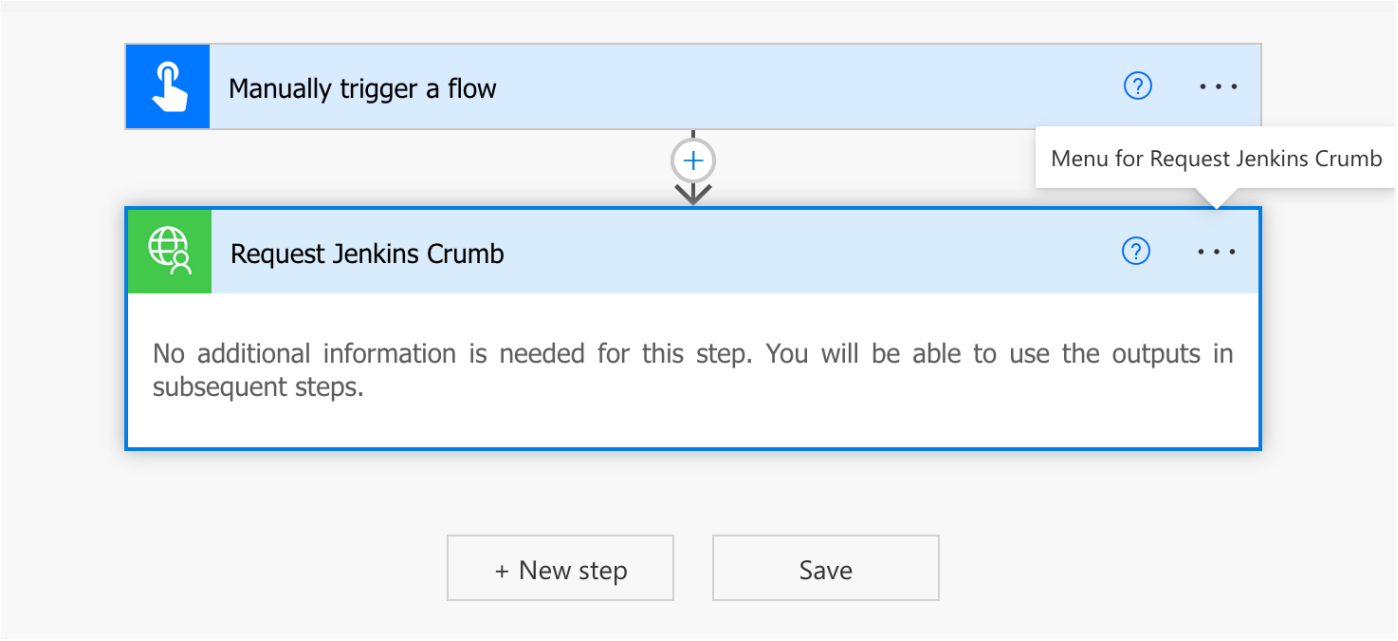
click the “custom” tab under “Choose an operation”
Here you should see your custom Jenkins connector (I have a few). Choose the one you created.



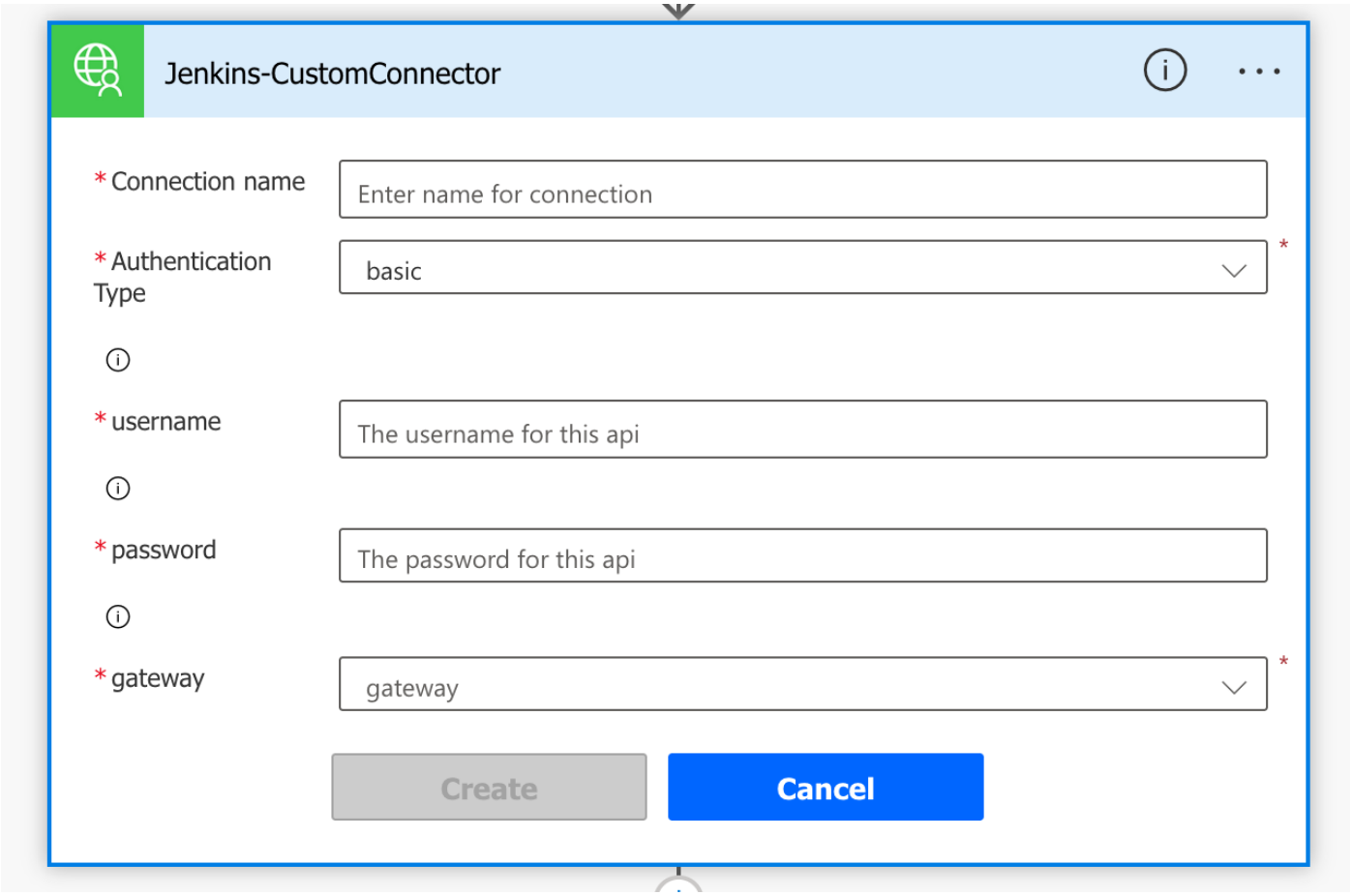
A list of Actions should appear.



Choose “Request Jenkins Crumb” (or the description you used for your custom connector)



Click on the ellipsis on the right of the action and select “Add a connection”



Connection name: Enter a name for your connection.

Authentication Type: Authentication can be left as Basic

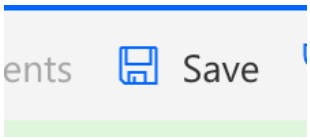
Username: enter a username (service account if you have one)

Password: enter the API Token we generated

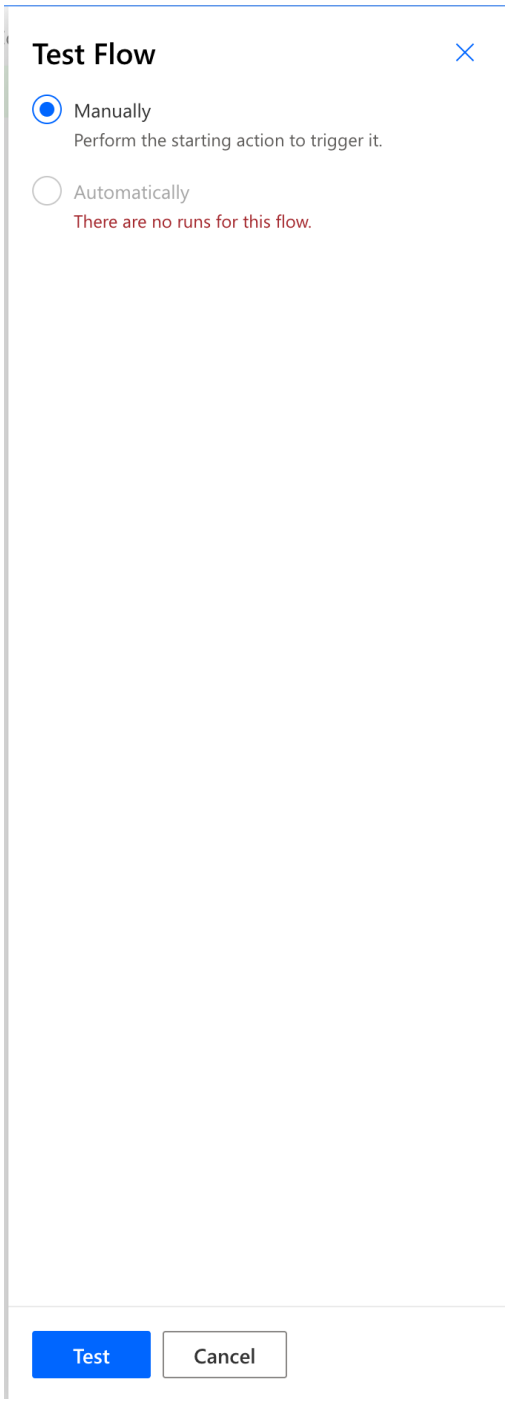
Gateway: Select the on-premises gateway we configured earlier

Click Create

click Save

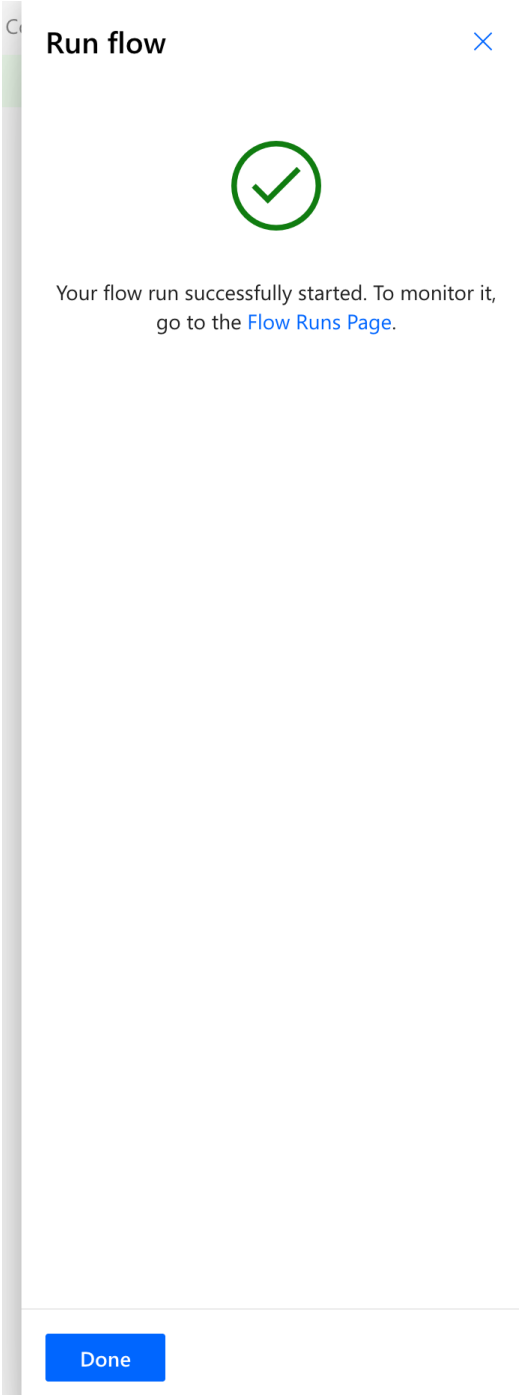


click test to test our flow with our custom connector.

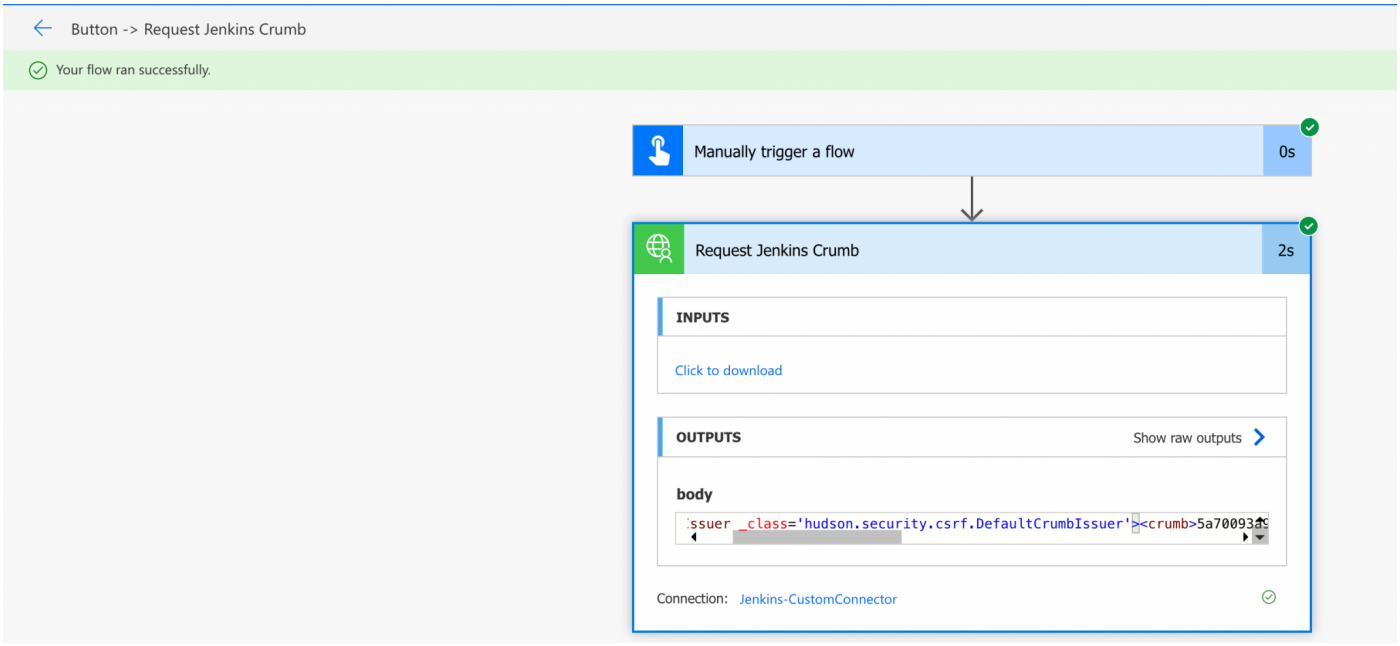


Click Test

Click Continue & Run Flow



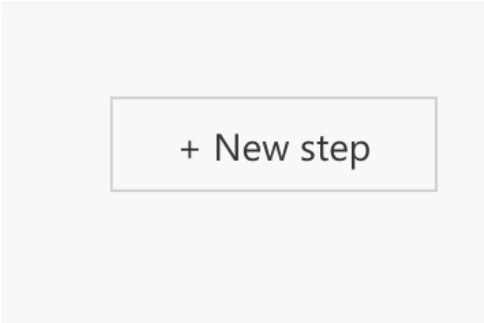
As a result, the “Request Jenkins Crumb” action should show the Jenkins Crumb in the body



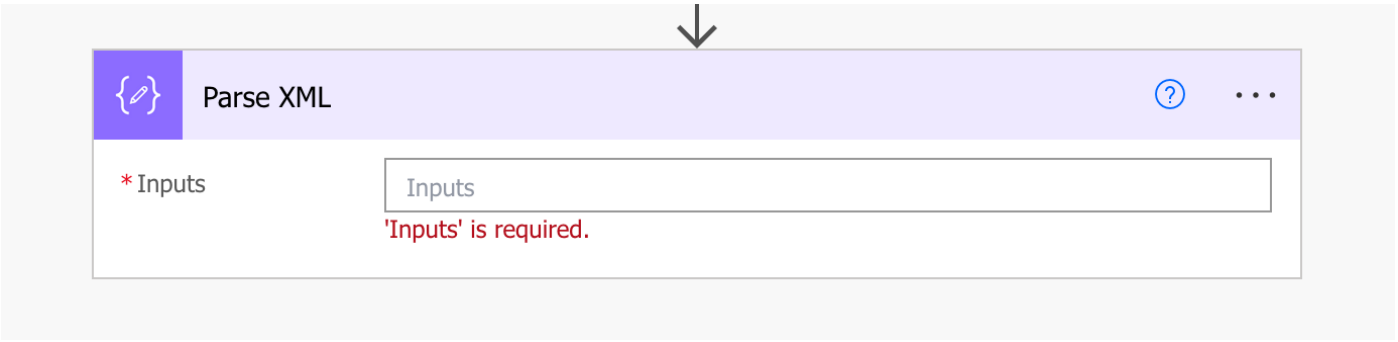
Now that we know we can use the Request Jenkins Crumb action, The next step is to parse the body from the outputs of “Request Jenkins Crumb” action.

Click on the Edit button to go back to editing mode

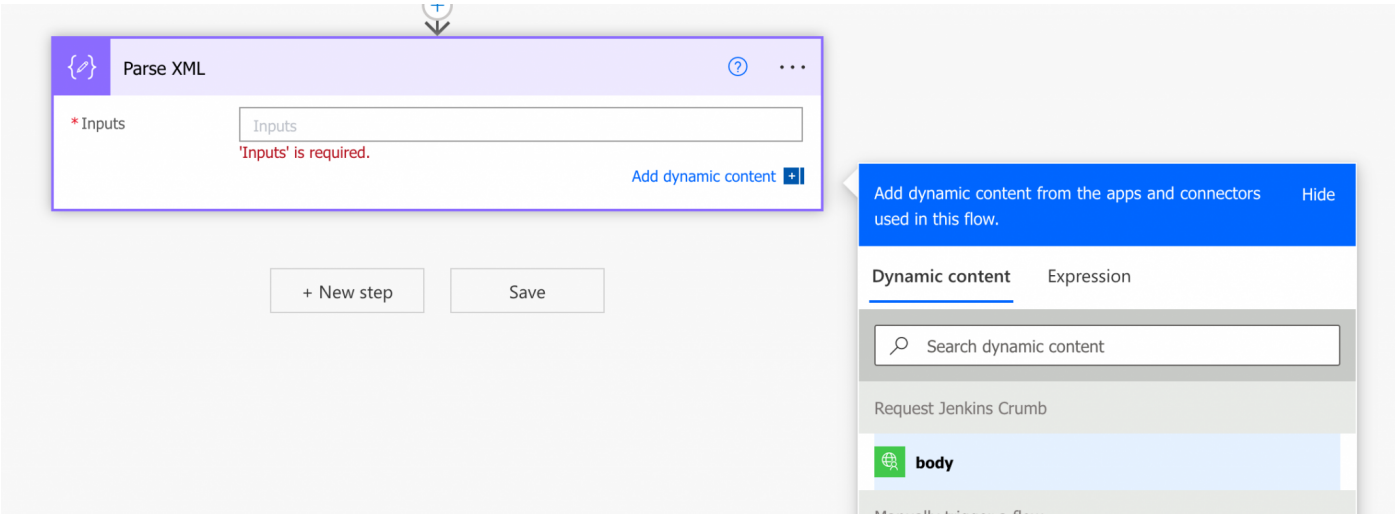
Click Next Step



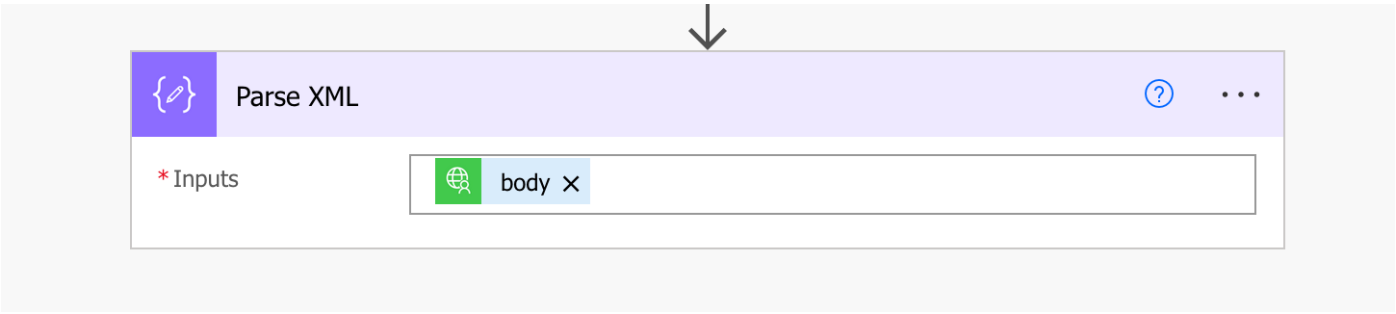
When the “Choose an operation” window appears, search for Compose.
Select the 3 dots on the right of the Compose action, and rename it to “Parse XML” or something descriptive.



Click on the “Inputs” field. You should see an option to add the dynamic data from the previous step. Select the body from the “Request Jenkins Crumb” action.



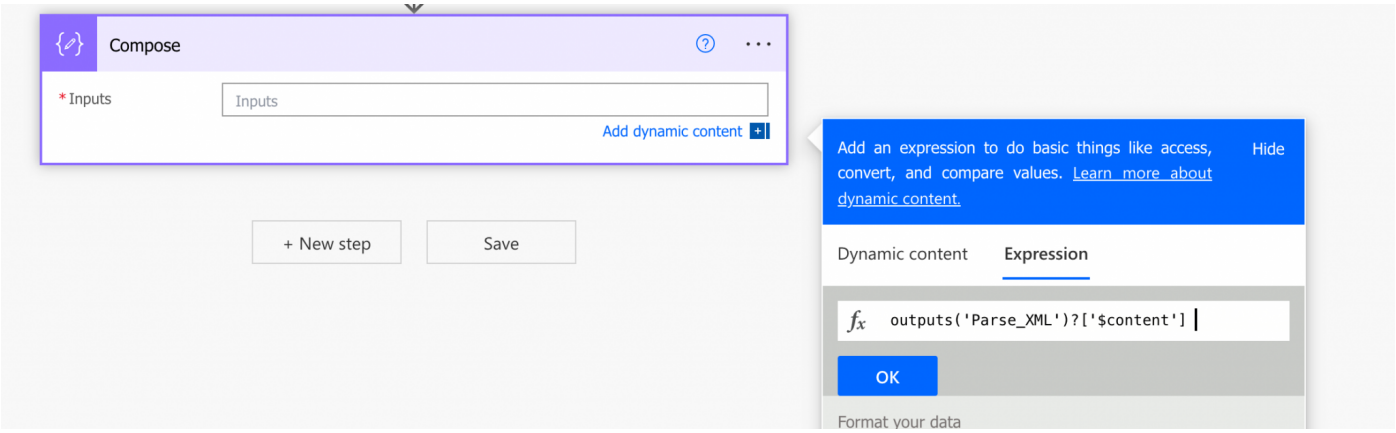
you new Compose operation should look similar to the below:



Click Next step

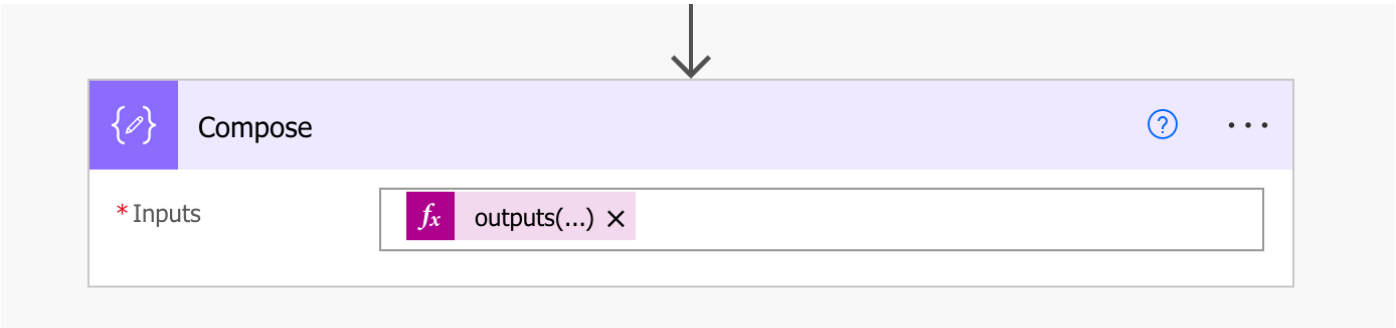
Search for Compose

Click on the “Inputs” field. This time, click on the “Expression” tab and enter the following regular expression: `outputs('Parse_XML')?['$content']`



click OK

You new Compose operation should look similar to this:

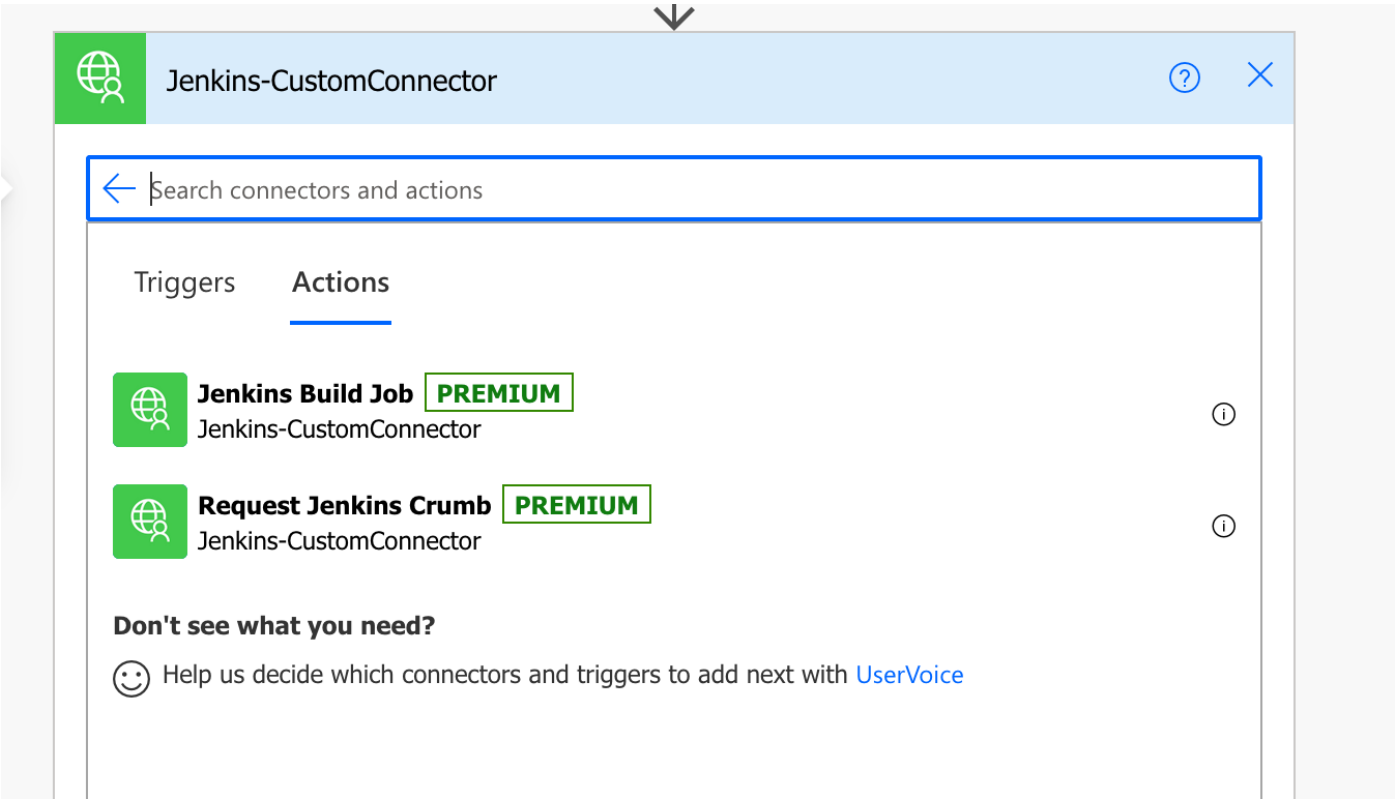


If you save an run the flow now, the last Compose Operation will show the Jenkins Crumb only. We can now use this to call our Jenkins Job.

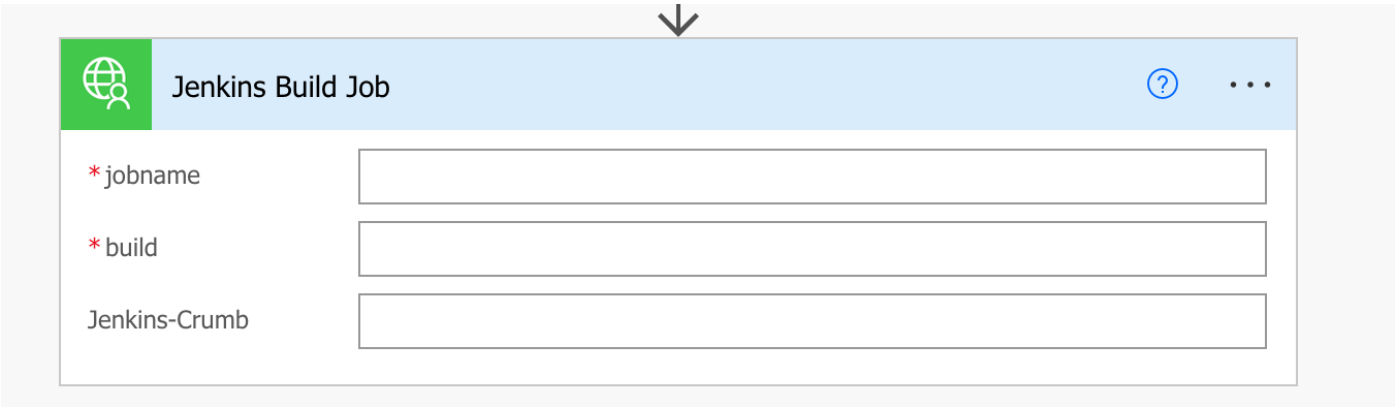
Click Next Step and choose the Custom tab when the “Choose an operation” window appears.

Click the Custom Jenkins connector we built.

Click on the “Jenkins Build Job” Action.

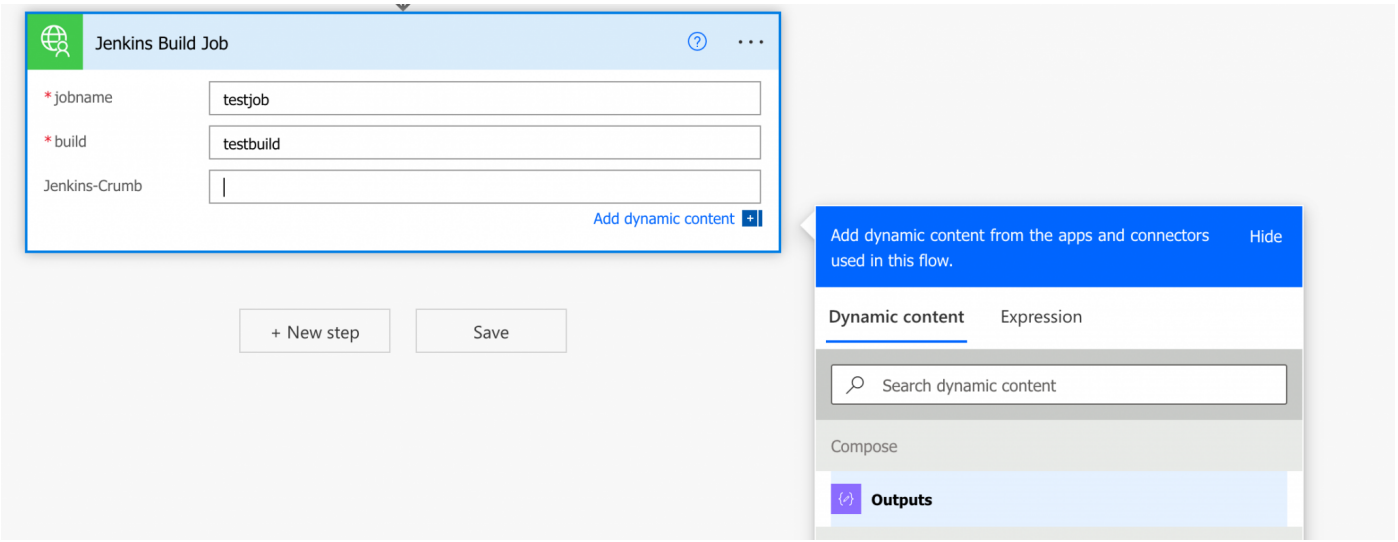


Click on the ellipsis on the right of the action and choose the connection we created earlier.

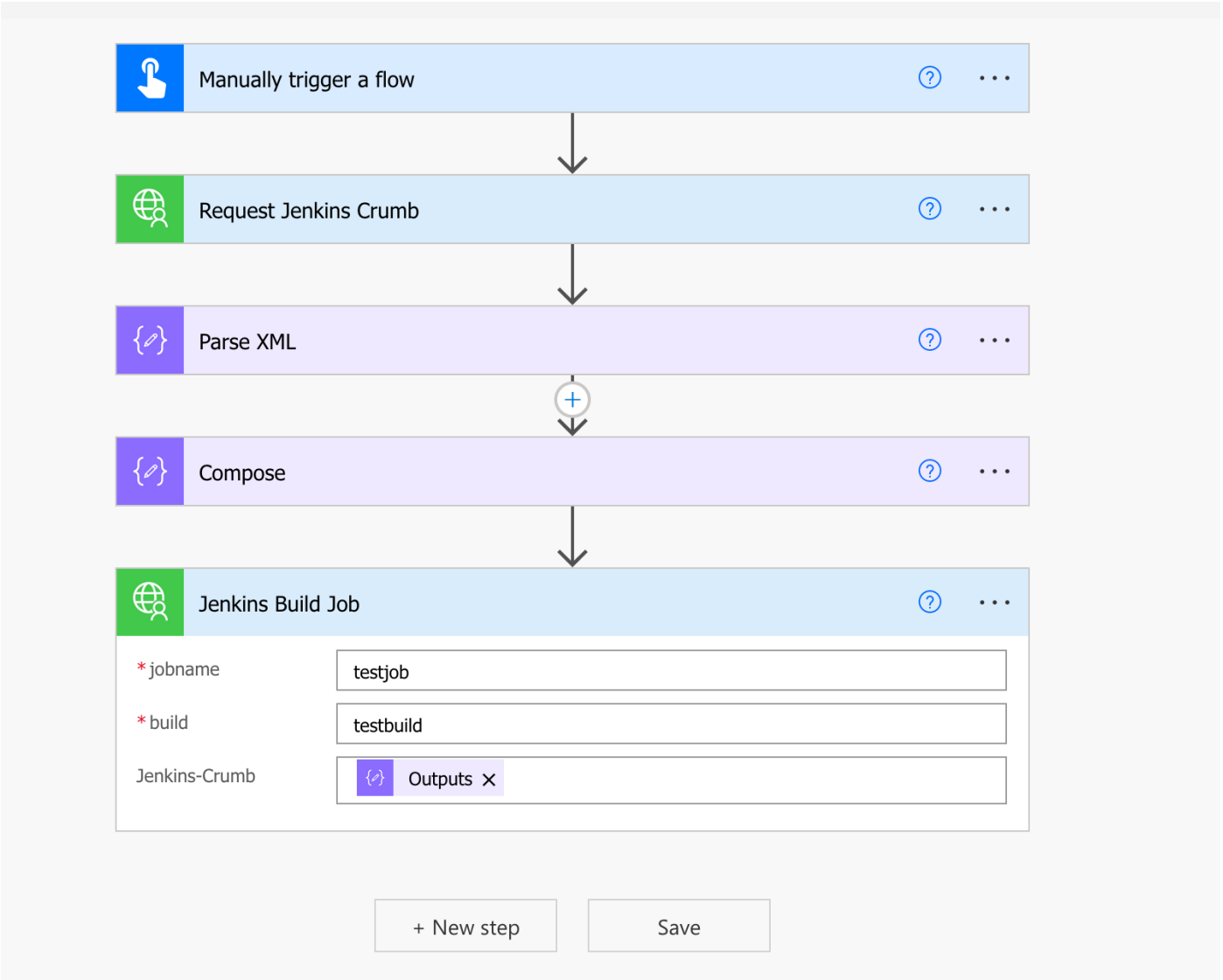


enter your parameters. for the purpose of this article, I will be starting a Jenkins test job I have created.

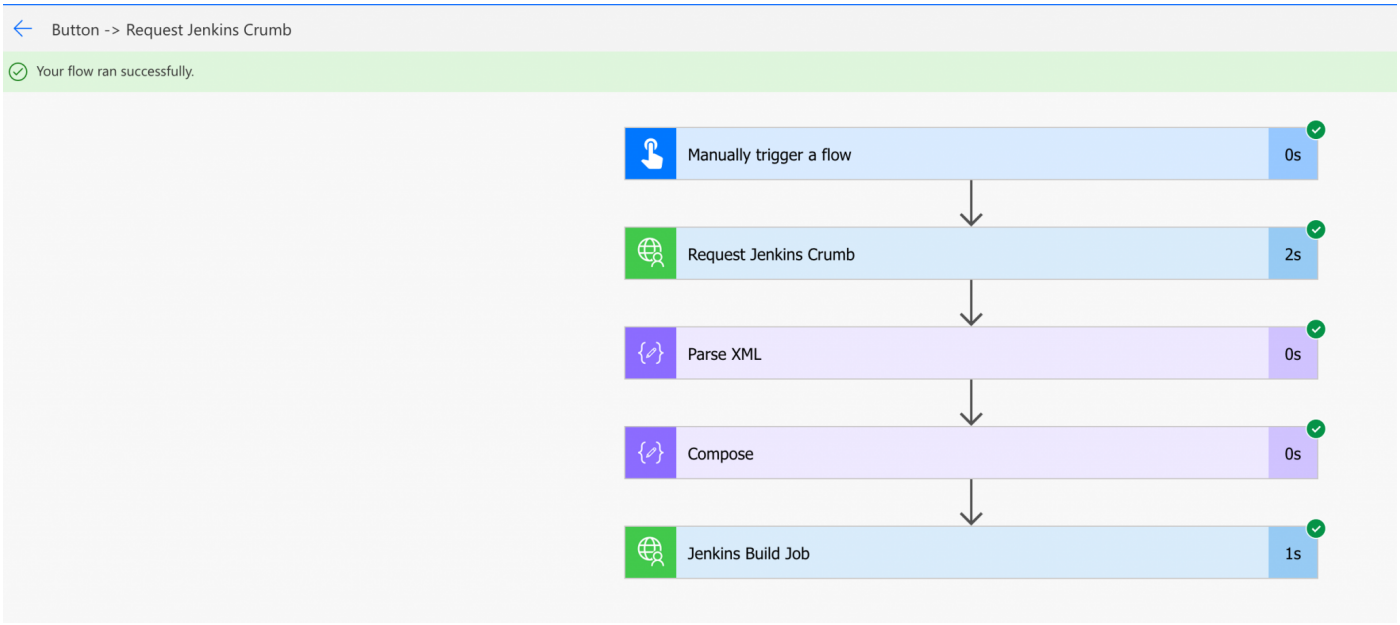
For the Jenkins-Crumb parameter, click on the “Jenkins-Crumb” field and select the outputs from the previous “Compose” action.



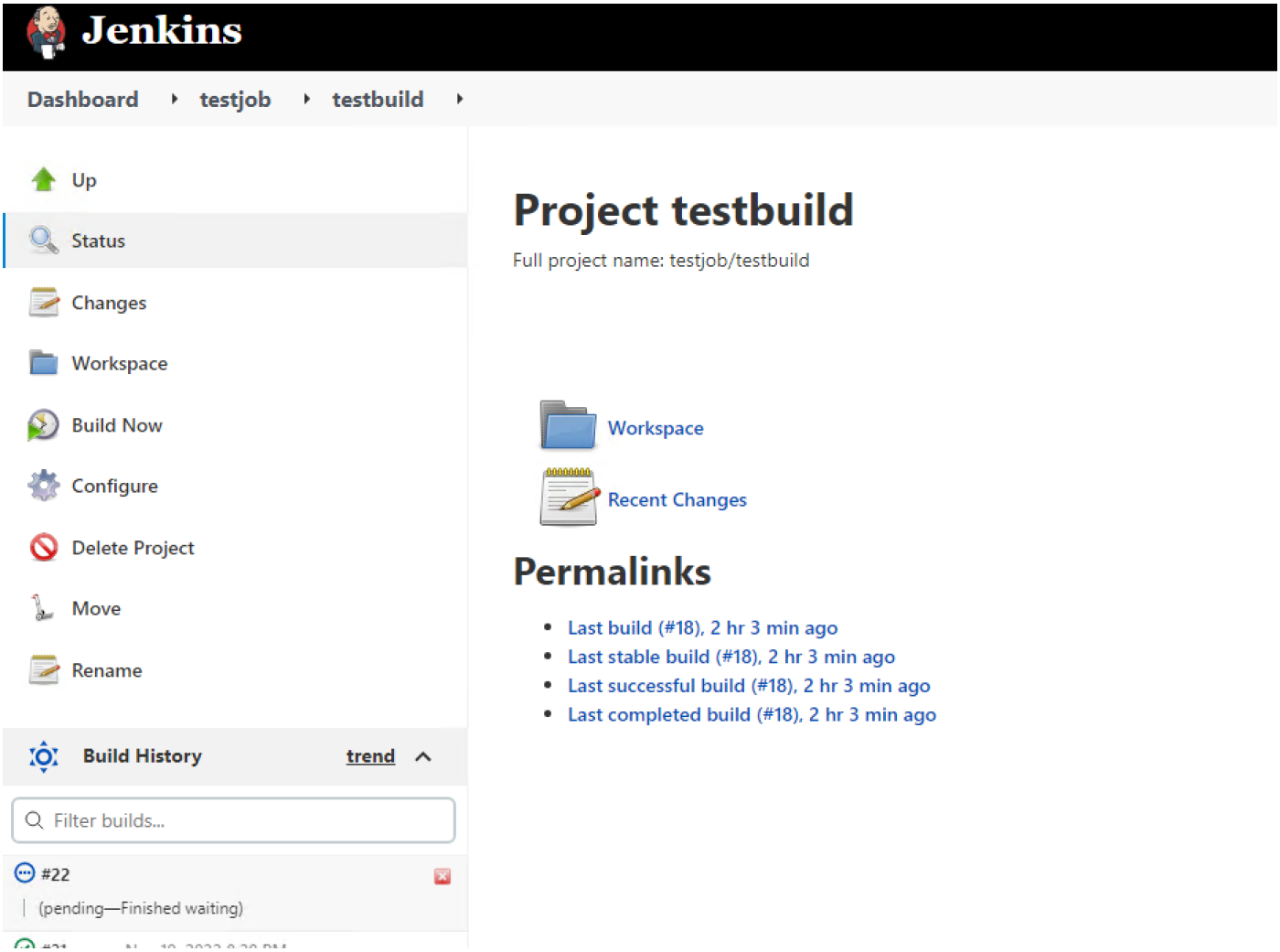
You flow should now look similar to the below:

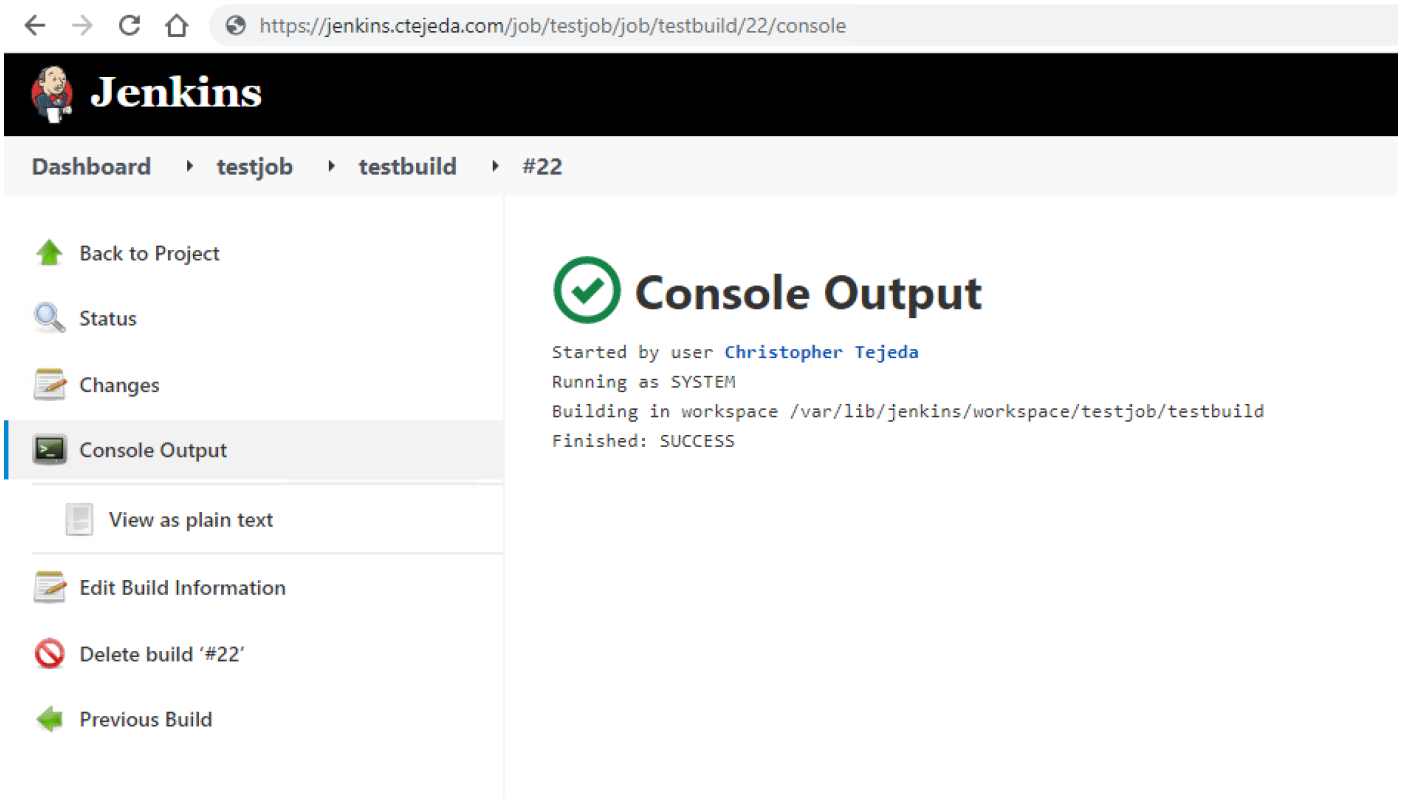


Finally, you can save and test your flow.



As a result, your Jenkins Job should be Running.





Conclusion

We can now call Jenkins Build using the Internal Jenkins API, and Microsoft Power Automate (Flows)

as an option, you can upload the custom connector by using the below JSON

```
{
  "swagger": "2.0",
  "info": {
    "title": "Jenkins-CustomConnector",
    "description": "A custom Jenkins connector which utilizes the Jenkins API to interact with the Jenkins system. \nAuthor: Chris Tejeda",
    "version": "1.0"
  },
  "host": "jenkins.ctejeda.com",
  "basePath": "/",
  "schemes": [
    "http"
  ],
  "consumes": [],
  "produces": [],
  "paths": {
    "/crumbIssuer/api/xml": {
      "get": {
        "responses": {
          "default": {
            "description": "default",
            "schema": {}
          }
        },
        "summary": "Request Jenkins Crumb",
        "description": "Request a Crumb from Signed in Jenkins User",
        "operationId": "REQCRUMB",
        "parameters": []
      }
    },
    "/job/{jobname}/job/{build}/build": {
      "post": {
        "responses": {
          "default": {
            "description": "default",
            "schema": {}
          }
        },
        "summary": "Jenkins Build Job",
        "description": "Jenkins Build Job",
        "operationId": "BUILDJENK",
        "parameters": [
          {
            "name": "jobname",
            "in": "path",
            "required": true,
            "type": "string"
          },
          {
```

```
            "name": "build",
            "in": "path",
            "required": true,
            "type": "string"
          },
          {
            "name": "Jenkins-Crumb",
            "in": "header",
            "required": false,
            "type": "string"
          }
        ]
      }
    }
  },
  "definitions": {},
  "parameters": {},
  "responses": {},
  "securityDefinitions": {
    "basic-auth": {
      "type": "basic"
    }
  },
  "security": [
    {
      "basic-auth": []
    }
  ],
  "tags": []
}
```