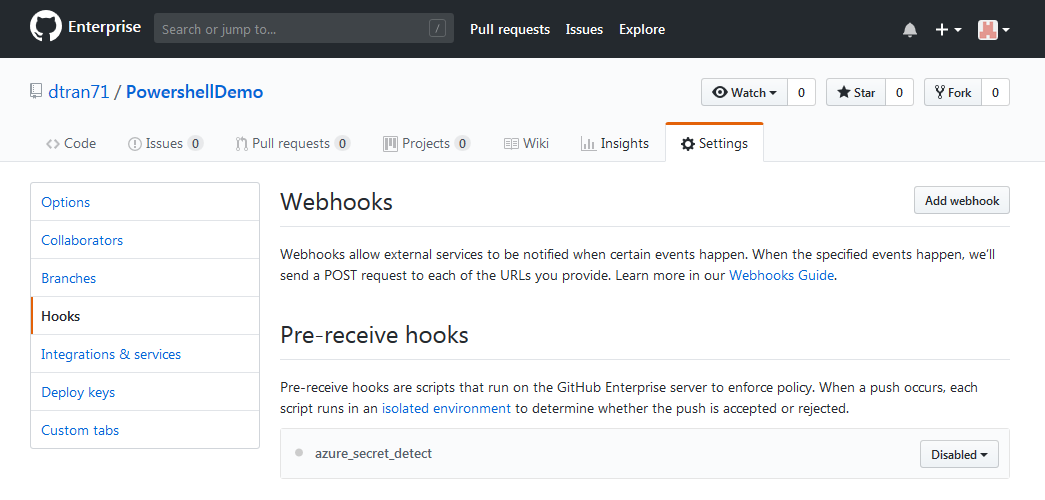
Sau khi thiết lập Azure VM agent để tạo máy từ xa , chúng em sẽ bắt đầu tạo build trên Jenkins cho việc triển khai code. Build khi chạy sẽ kích hoạt plugin Azure VM agent để tạo một máy ảo trên tài khoản Azure đã được cấu hình sẵn và sẽ chạy code và các tùy chỉnh của Build trên máy ảo ấy. Để có được code chạy cho build , chúng em làm việc đó thông qua tới một thứ có thể giúp cho Jenkins đi tới Github để có thể kéo các code nguồn về để chạy build , thứ đó gọi là Github webhook.

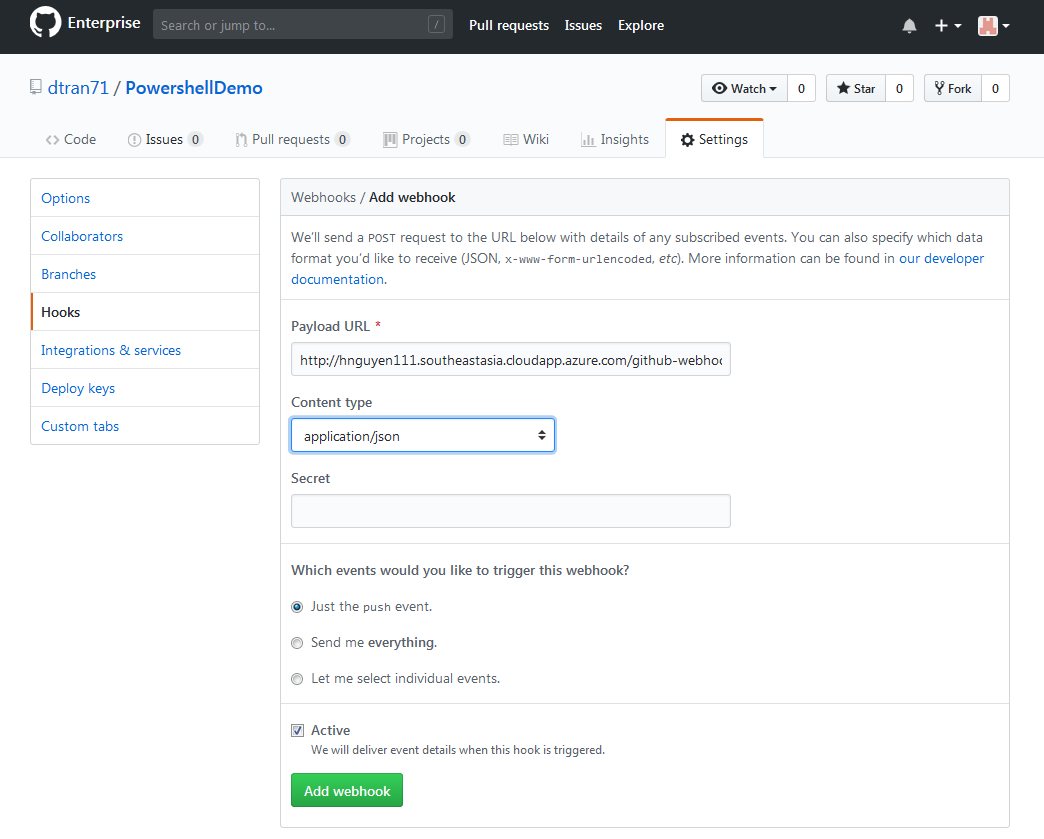
Github webhook hoạt động như sau : Khi có một sự kiện nào đó xảy ra trên kho lưu trữ từ xa ( Merge , push , fork ,.. ) , Github webhook sẽ gửi một payload HTTPS POST đến địa chỉ URL đã được cấu hình trên WebHook. Việc này sẽ giúp kích hoạt build của Jenkins dựa trên Github. Github webhook có thể được cấu hình như sau :

Github Webhook nằm ở mục hook trong Setting của kho lưu trữ từ xa trên giao diện web :



Hình 2.?? Github Webhook

Khi nhấp vào mục “Add webhook” , Github sẽ đưa người dùng tới trang cấu hình webhook cho kho lưu trữ. Trang này sẽ cho chúng ta cấu hình nhiều nhứ cho webhook như địa chỉ URL mà webhook sẽ gửi Payload , Secret token để tăng cường bảo mật , loại event để kích hoạt webhook,..

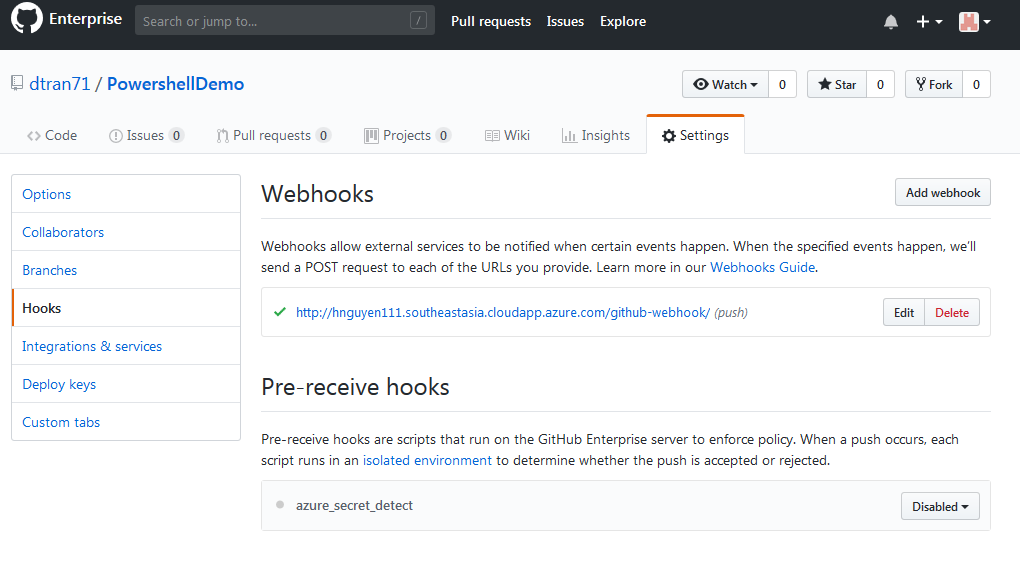


Hình 2.?? Cấu hình Github webhook

Trong đó :

* **Payload URL :** Đây là địa chỉ URL mà Github webhook sẽ gửi payload tới . Chúng em sẽ để URL của Jenkins.
* **Content Type :** Kiểu thông điệp của payload . Github webhook hỗ trợ application/x-www-form-urlencoded và application/json.
* **Secret :** Người dùng có thể nhập vào đây một đoạn chữ để có thể làm secret token cho webhook .
* **Which events would you like to trigger this webhook ?:** Loại sự kiện xảy ra trên kho lưu trữ để kích hoạt webhook. Có rất nhiều loại sự kiện có thể dùng để kích hoạt webhook nhưng chúng em tập trung vào chỉ sự kiện push.
* Mục cuối cùng sẽ cho phép người dùng khởi động hay ngưng webhook tùy ý muốn.

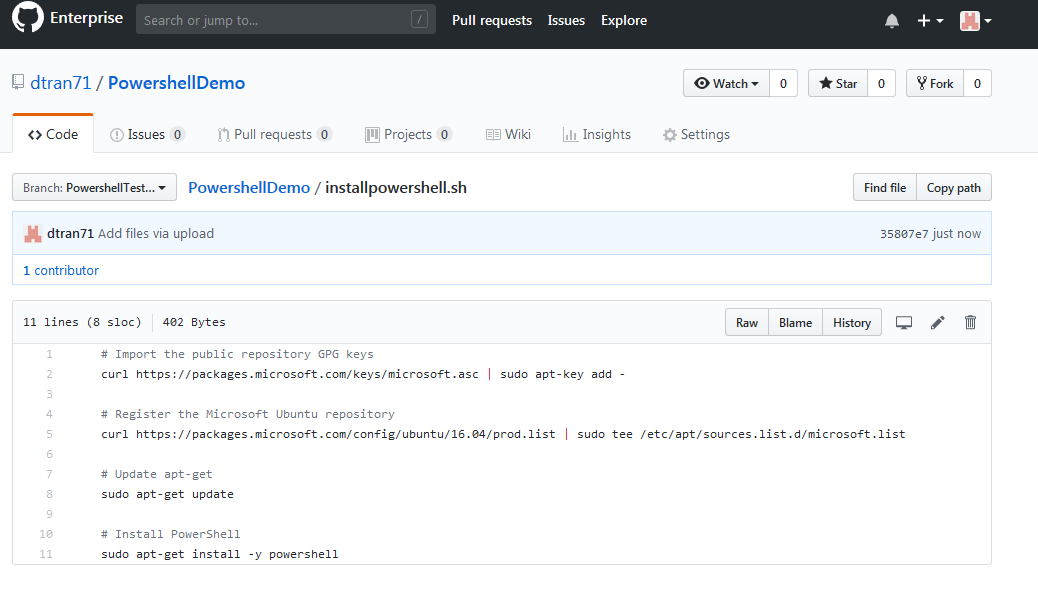
Sau khi nhấn Add webhook , webhook mới được tạo sẽ xuất hiện ở trang Hook.



Hình 2.??? Webhook đã được tạo

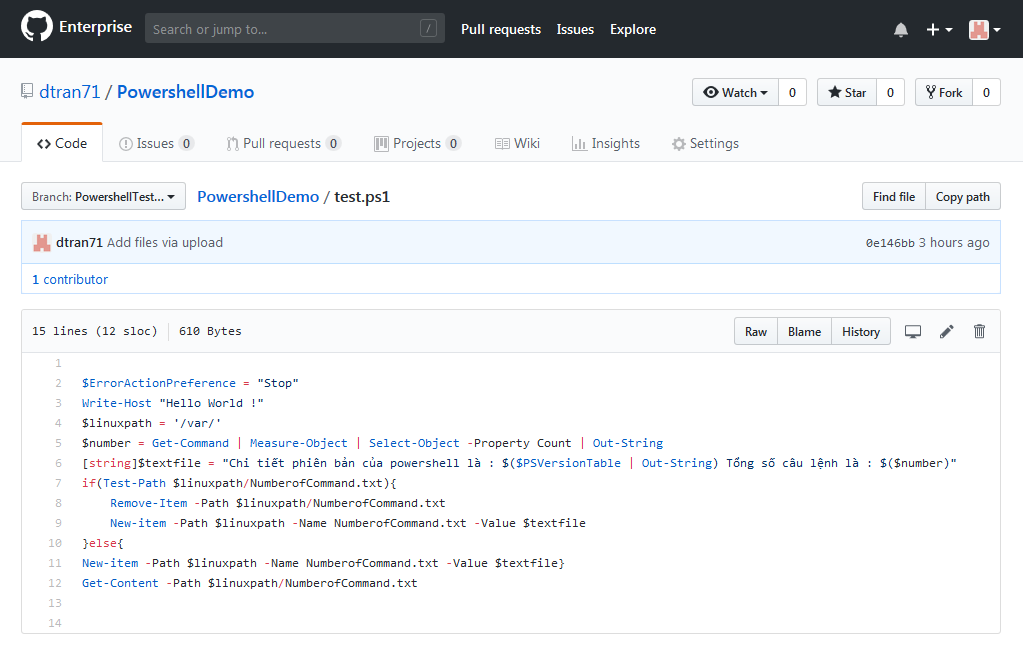
Việc cấu hình build cũng sẽ hoàn toàn dựa vào người dùng sẽ chạy code gì trên build đó. Ở đây chúng em sẽ chạy một đoạn code powershell trên máy ảo với hệ điều hành Ubuntu 16.04 đã được cấu hình trên Azure VM agent plugin. Build của chúng em sẽ tập trung vào 2 file chính có trên kho lưu trữ PowershellDemo nhánh PowershellTestBranch đó là installpowershell.sh và test.ps1.

* **installpowershell.sh :** Không như hệ điều hành Window , các hệ điều hành linux như Ubuntu không có sẵn powershell để chạy code powershell. Thay vào đó , người dùng phải cài đặt thủ công powershell qua các câu lệnh. installpowershell.sh là một file shell script có nhiệm vụ là cài đặt powershell cho máy ảo của Azure VM agent thay vì phải cài đặt thủ công powershell trên máy ảo.



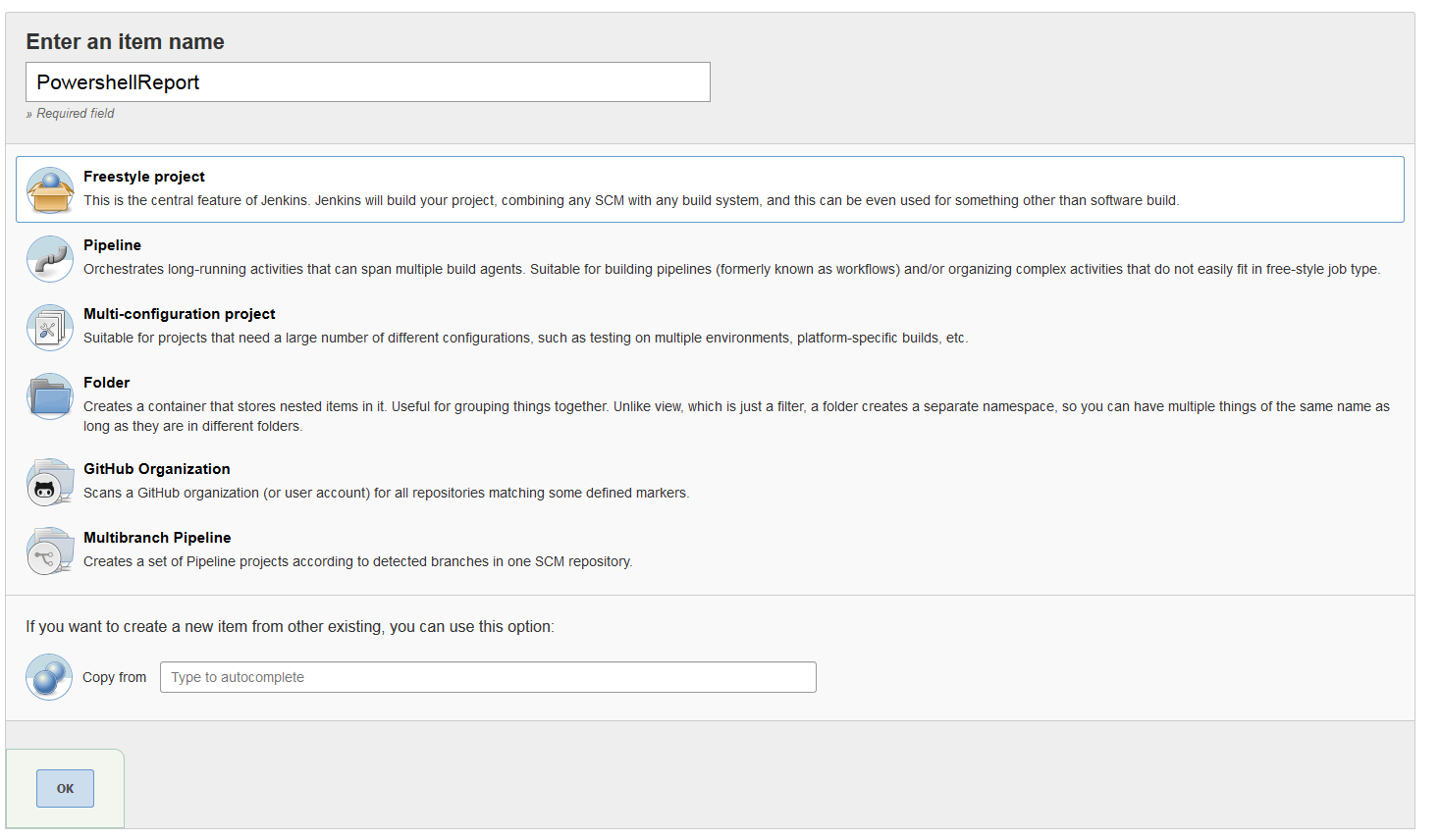
Hình 2.?? : Đoạn code của installpowershell.sh

* **test.ps1 :** test.ps1 là file script được viết bằng ngôn ngữ powershell. Đuôi ps1 là mặc định cho các file script của powershell . Script của test.ps1 được chạy như sau : Trước tiên nó sẽ viết ra dòng chữ “Hello World !” và lấy thông tin về chi tiết phiên bản powershell hiện cùng với tổng số câu lệnh , lưu chúng vào một biến dạng string , sau đó tạo một file mới ở dường dẫn định sẵn rồi cuối cùng là lấy thông tin của file đó.



Hình 2.?? : Đoạn code của test.ps1

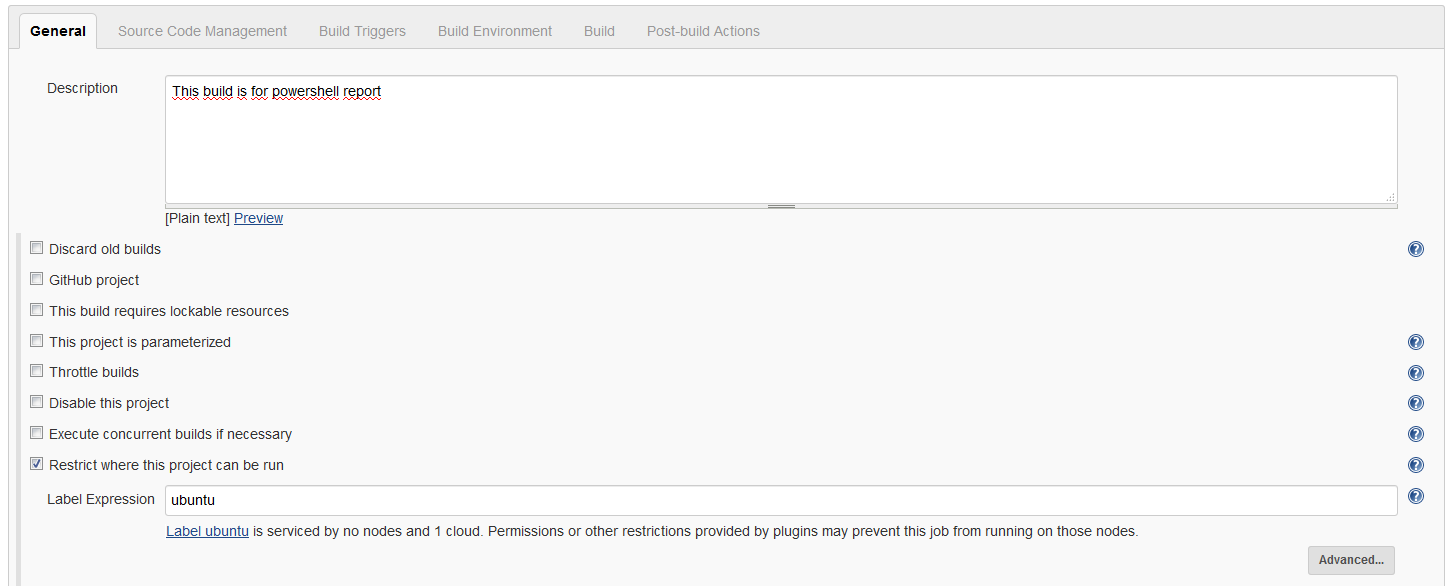
Để xây dựng một build cho Jenkins , ta nhấp nút vào New Item ở trang chủ Jenkins , nó sẽ dẫn ta đến một trang khác cho ta chọn đặt tên và chọn loại project của build . Các loại project khác nhau thì sẽ có cấu hình build khác nhau. Dưới đây chúng em tạo một build tên là “PowershellReport” , loại project của build là freestyle vì freestyle kiểu dạng project linh hoạt nhất , nó thích hợp với nhiều loại build đơn giản và không có cần tập trung quá nhiều chi tiết về cấu hình để mang lại hiệu quả mong muốn.



Hình 2.?? : Tên và loại project của build

Sau khi đặt tên và chọn loại project thì Jenkins sẽ đưa người dùng tới trang cấu hình build . Với build thuộc loại freestyle project thì trang cấu hình build sẽ chia làm 6 phần chính để cấu hình :

1. **General :** Đây là các tùy chỉnh tổng quát của build . Có các tùy chỉnh như xóa lịch sử build , dừng project của build , chạy build song song ,..nhưng cái chúng em tập trung là giới hạn build này sẽ chạy ở đâu.
2. **Source Code Management :** Source Code Management là mục dùng để tùy chỉnh các tích hợp của Git và Jenkins .
3. **Build Trigger :** Mục này sẽ cho người dùng tùy chỉnh một hoặc nhiều cách để kích hoạt build chạy tự động trên Jenkins .
4. **Build Environment :** Build Environment cho phép người dùng chọn các thao tác cần làm trên môi trường build như thêm thời gian vào log của Console , xóa folder làm việc trước khi build,…
5. **Build :** Mục này là một mục quan trọng trong việc cấu hình các build . Mục này cho phép người dùng thêm vào các bước xử lý code nguồn như thế nào để chạy phù hợp với mong muốn . Nó có các tùy chỉnh như là chạy lệnh shell , chạy lệnh batch , chạy với Maven ,…
6. **Post build actions :** Nếu như mục Build là dùng để thêm bước xử lý code nguồn thì Post build actions dùng để thêm bước hậu xử lý của build với những bước như xóa folder làm việc sau khi build xong , gửi email , tiếp tục build các project khác ,…



Hình 2.?? : Mục General

Ở mục General , chúng em viết miêu tả cho build để dễ dàng nhận biết build với các build khác nếu có . Chúng em cũng tích chọn vào mục “Restrict where this project can be run” , mục này sẽ xác định xem agent nào sẽ chạy các build của project này bằng các sử dụng nhãn hoặc tên của agent . Sử dụng nhãn “ubuntu” đã được tạo trên Azure VM agent plugin sẽ cho build được chạy trên các máy ảo có nhãn đó .

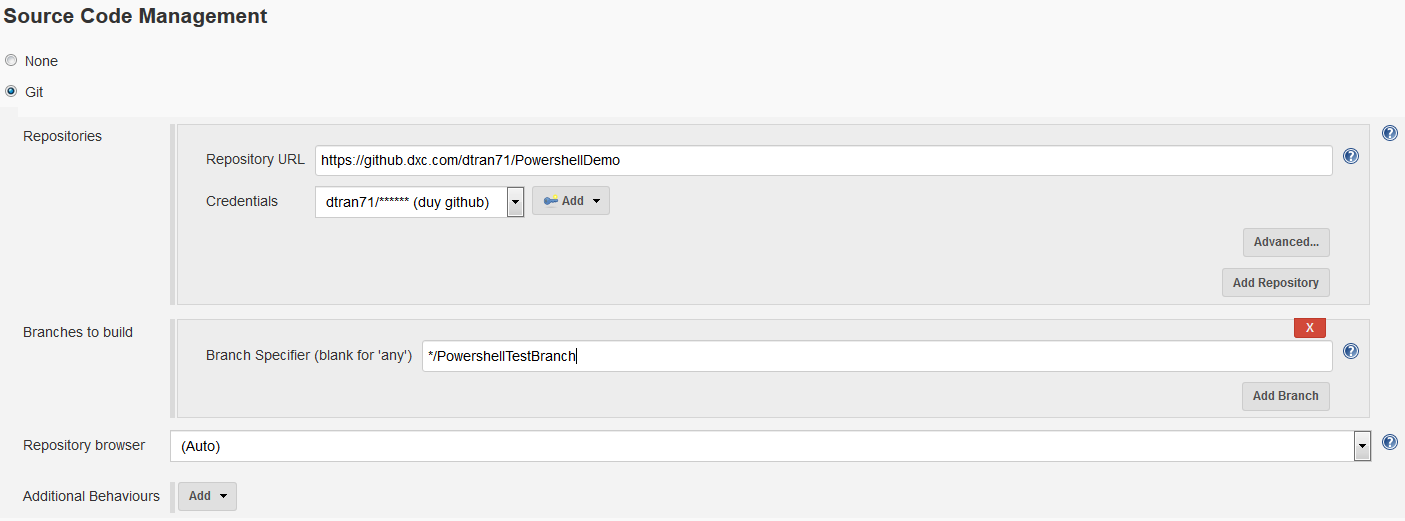
Với mục Source Code Management , thứ chúng em sẽ tùy chỉnh sẽ gồm ba thứ :

1. **Repository URL** : đây sẽ là địa chỉ của kho lưu trữ từ xa mà chúng em sử dụng để lưu trữ code nguồn : <https://github.dxc.com/dtran71/PowershellDemo> .
2. **Credentials :** Credentials là thông tin xác thực để Jenkins có thể dùng truy cập tới trang kho lưu trữ và lấy code nguồn . Nhấn vào nút add sẽ hiện lên bảng thêm Credential và chúng em thêm Credential như sau :

Hình 2.?? : Bảng tạo Credential

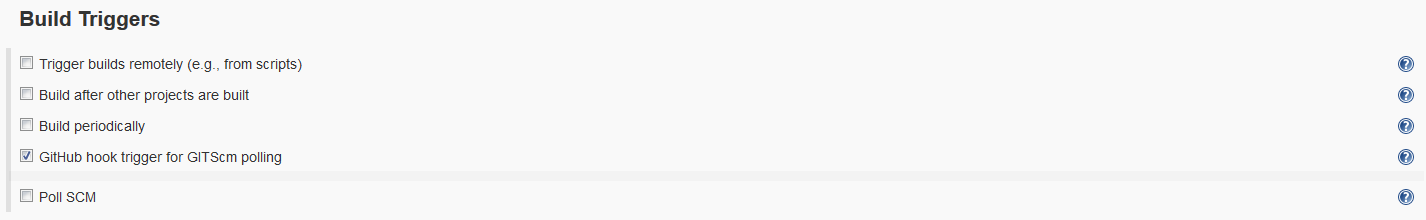
Với Credentials Plugin của Jenkins , Credentials có nhiều loại khác nhau tùy theo mục đích sử dụng của người dùng , trong đây chúng em sử dụng loại “Username with password” là tên tài khoản với mật khẩu . **Username** sẽ là tên đăng nhập vào Github , **Password** sẽ là mật khẩu đăng nhập vào Github nhưng thay vì xài mật khẩu chúng em sử dụng chuỗi Personal Token của Github để thay cho mật khẩu nhằm tăng tính bảo mật . **ID** là mã định danh độc nhất cho từng Credential , người dùng có thể nhập mã vào đây bằng tay hoặc nếu để trống Jenkins sẽ tự động sinh ra mã . **Decripstion** sẽ là miêu tả của Credential .

1. **Branches to build :** Tùy chỉnh này là dùng để lựa chọn một hoặc nhiều nhánh cụ thể của kho lưu trữ mà Jenkins sẽ lấy code nguồn để build. Thay vì nhánh master , chúng em sẽ chọn nhánh PowershellTestBranch .



Hình 2.?? : Mục Source Code Management

Tới mục Build Triggers , vì build sẽ được kích hoạt hông qua Github webhook nên chúng em sẽ tích vào lựa chọn “GitHub hook trigger for GITScm polling” . Nếu như Jenkins nhận được Webhook từ URL Github đã được cấu hình ở trên thì nó sẽ kích hoạt build này.



Hình 2.?? : Mục Build Triggers