**IA Series - Sentiment Analysis NLP cook-book (using Pre-Built NLP Model)**

**By Ashish Easow**

## Introduction

As part of mandatory reading for folks in RPA these days, machine learning is a topic which most of the time remains in the to-do list. But building large Machine learning models itself require machines having considerable processing power, which may not be possible for every RPA developer.  
Sentiment analysis is a very old topic in NLP space, the guide gives a brief how to for implementation of a pre-built Sentiment Analysis model and its use in a basic solution without using high performance GPU based machines.

Actual use cases are varied in the fields of Social media monitoring, Brand monitoring, Voice of customer (VoC),Customer service, Market research etc.

## Background

The guide tries to give a simplistic set up which can be set up and tested in a decently sized work laptop (for development and evaluation only). BERT is a language model first open sourced by Google in 2018 and rolled out in 2019. It is one of the best models available today for NLP and is based on the transformer model (not the movie).

The current model is created from fine tuning the existing BERT model using the IMDb movie reviews data set and should cater to most English content for sentiment analysis. It can except paragraphs with a maximum of 500 words. Ktrain is a python package which is used for simplification of the existing implementation. The actual model creation is not in the scope of this guide, but the source code will be added as part of the solution. In case you are looking for model creation from scratch, it took me 2-3 hours on a 12 GB GPU for creation of the final model using the source code and boasts of above 93% accuracy on the dataset tested.

# Software information

### Software List and links

|  |  |
| --- | --- |
| Term | Link |
| Blue Prism | Robotic Process Automation tool - <https://portal.blueprism.com/products/current> (v6.7) |
| GIT | Source code repository. You need to have GIT installed in your system.  For this guide a public repo was created at:  https://github.com/ashz30/Sentiment-Analysis-BERT-KTrain.git |
| Python | Python Version 3.6 is used, Download link [here](https://www.python.org/downloads/release/python-367/).  Python IDE used is PyCharm. Download link [here](https://www.jetbrains.com/pycharm/download/). (users can also run the code without the IDE as a command line too) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Pre-requisites

Prior to starting this process the below steps need to be configured for GIT repository set up

* Clone git repository - (above 1GB) – open cmd in a blank folder (mine is C:\GIT) and type

“git clone <https://github.com/ashz30/Sentiment-Analysis-BERT-KTrain.git>”

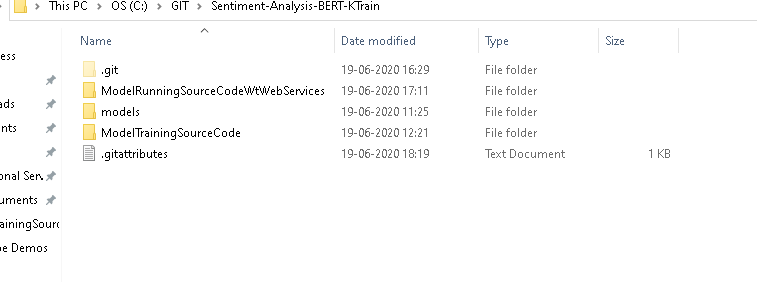
* Install Python, open any cmd window and run the below commands:
  + Check python version – “python --version”

(if this does not show the correct python version, you will need to edit windows path to point to python folderand open a new cmd window to try again)

* + “pip install ktrain” (to download and install Ktrain)
  + “pip install flask” (to download and install wrapper REST API framework)
* User will need to install Blue Prism in any machine.

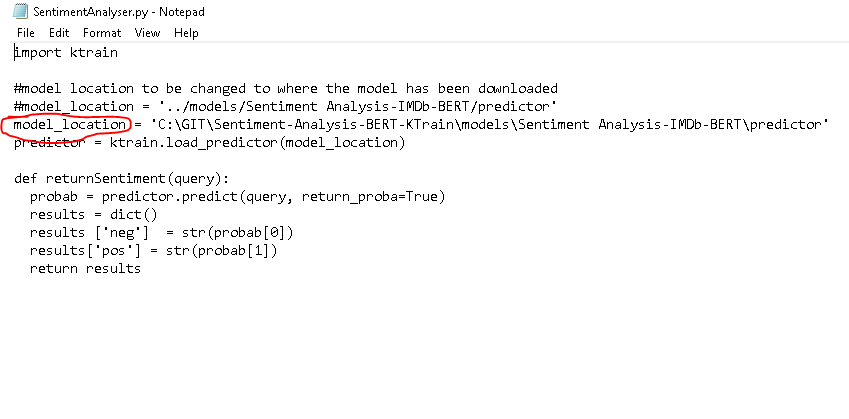
# Python Web service set up steps

* Once cloned this is the structure of the GIT folder –



* Goto folder **ModelRunningSourceCodeWtWebServices**, open file SentimentAnalyser.py and update

model\_location = <path of the predictor folder in **models** folder>



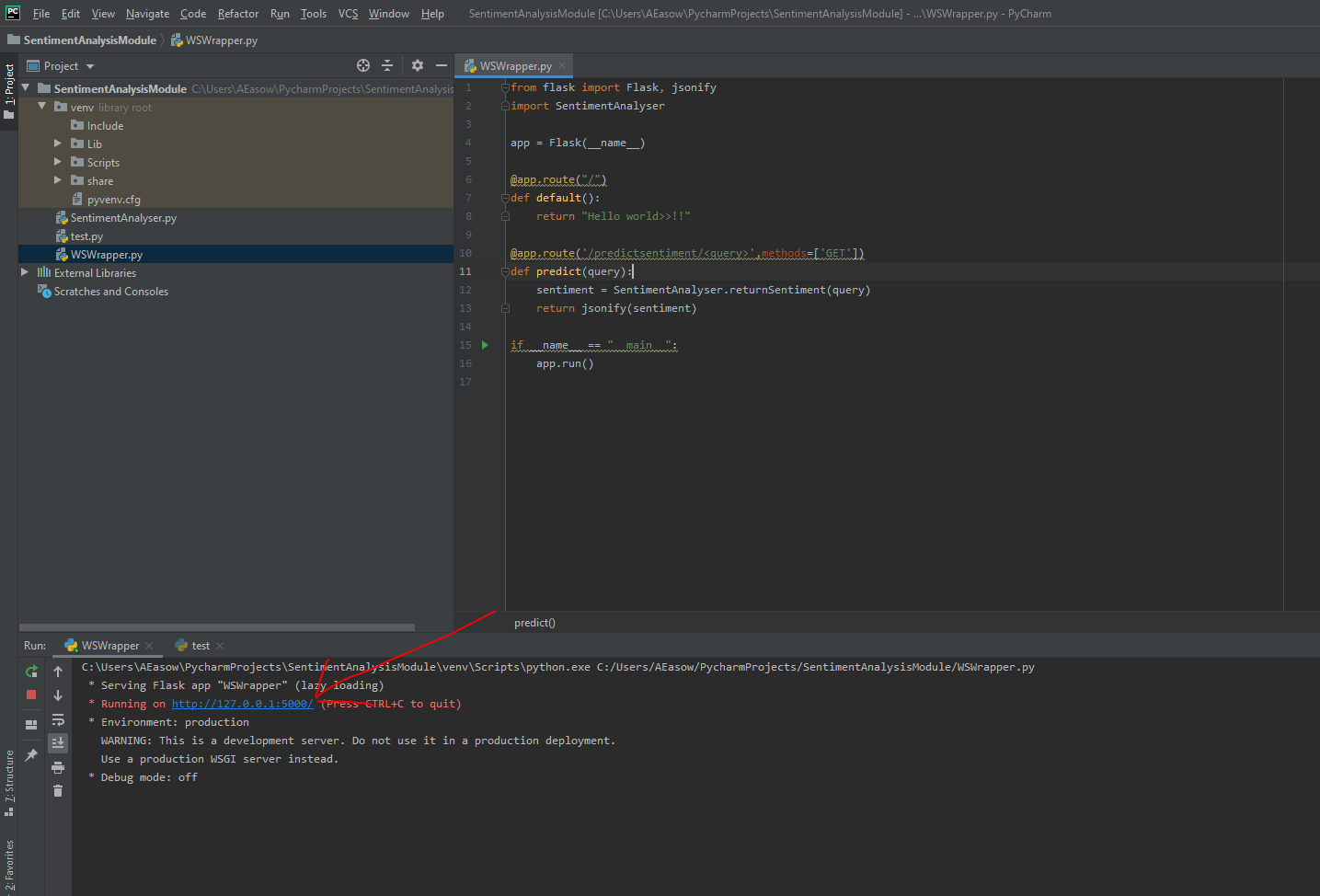
Save the file once done.

* Open command line in folder ‘**ModelRunningSourceCodeWtWebServices’** and run command

‘python WSWrapper.py’

alternately run WSWrapper.py through the IDE.

* Provided everything is set up correctly, it will load the model (takes a min) and start a web server in your local system.

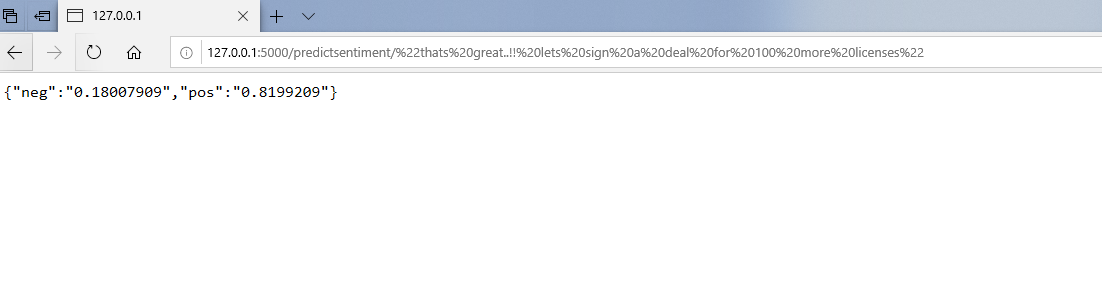


* Open your browser to goto link –

[127.0.0.1:5000/predictsentiment/"thats great..!! lets sign a deal for 100 more licenses"](http://127.0.0.1:5000/predictsentiment/%22thats%20great..!!%20lets%20sign%20a%20deal%20for%20100%20more%20licenses%22)

Hit enter

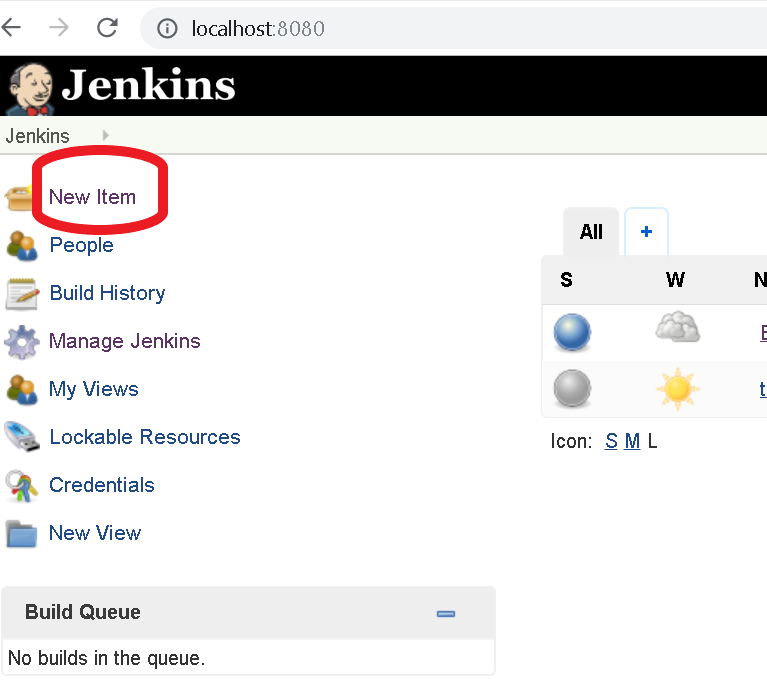
* You should see the result after a short pause the first time.



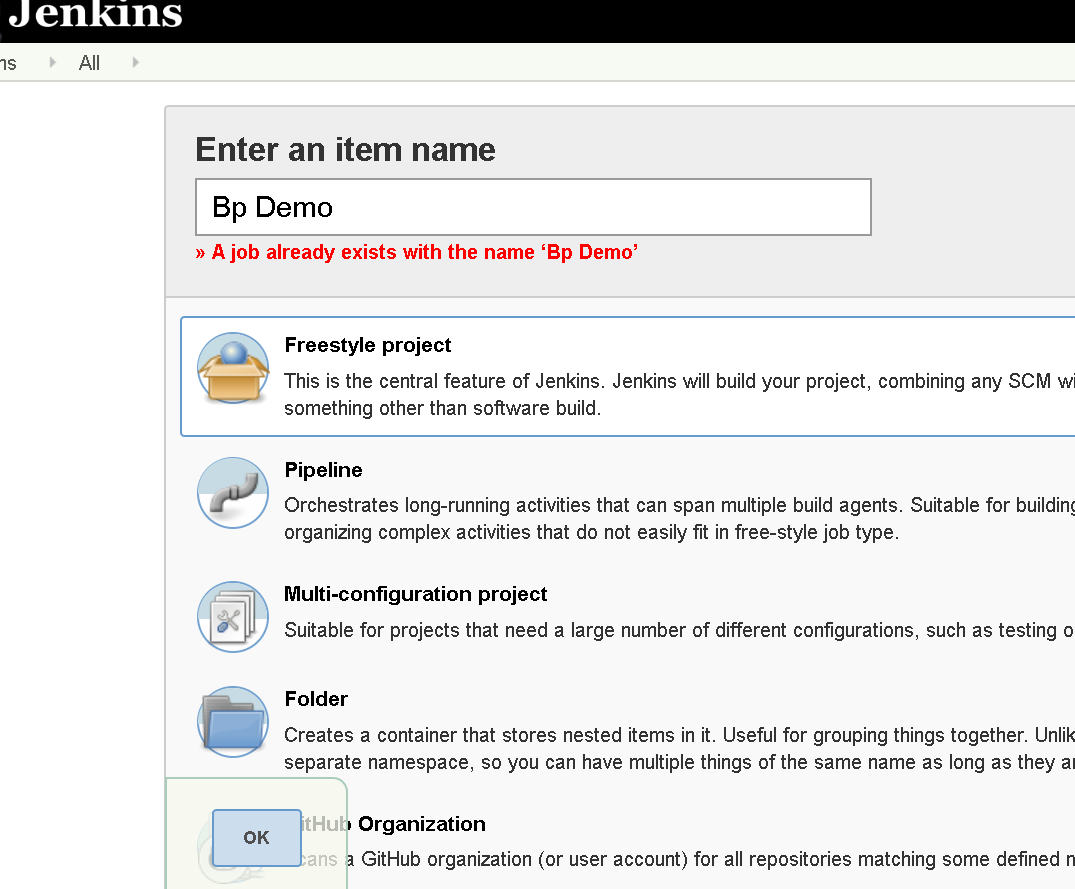
* There will never be a value for 100 percent, but rather a split between positive and negative.
* Also keep in mind, sentiment is a measure of feeling, so its never an exact science. To figure out if a statement is neutral it is best to take a call based on the confidence score. For eg. If both ‘pos’ and ‘neg’ are approaching 50%, then that’s an indicator of the statement being a neutral one.

# Blue Prism Steps

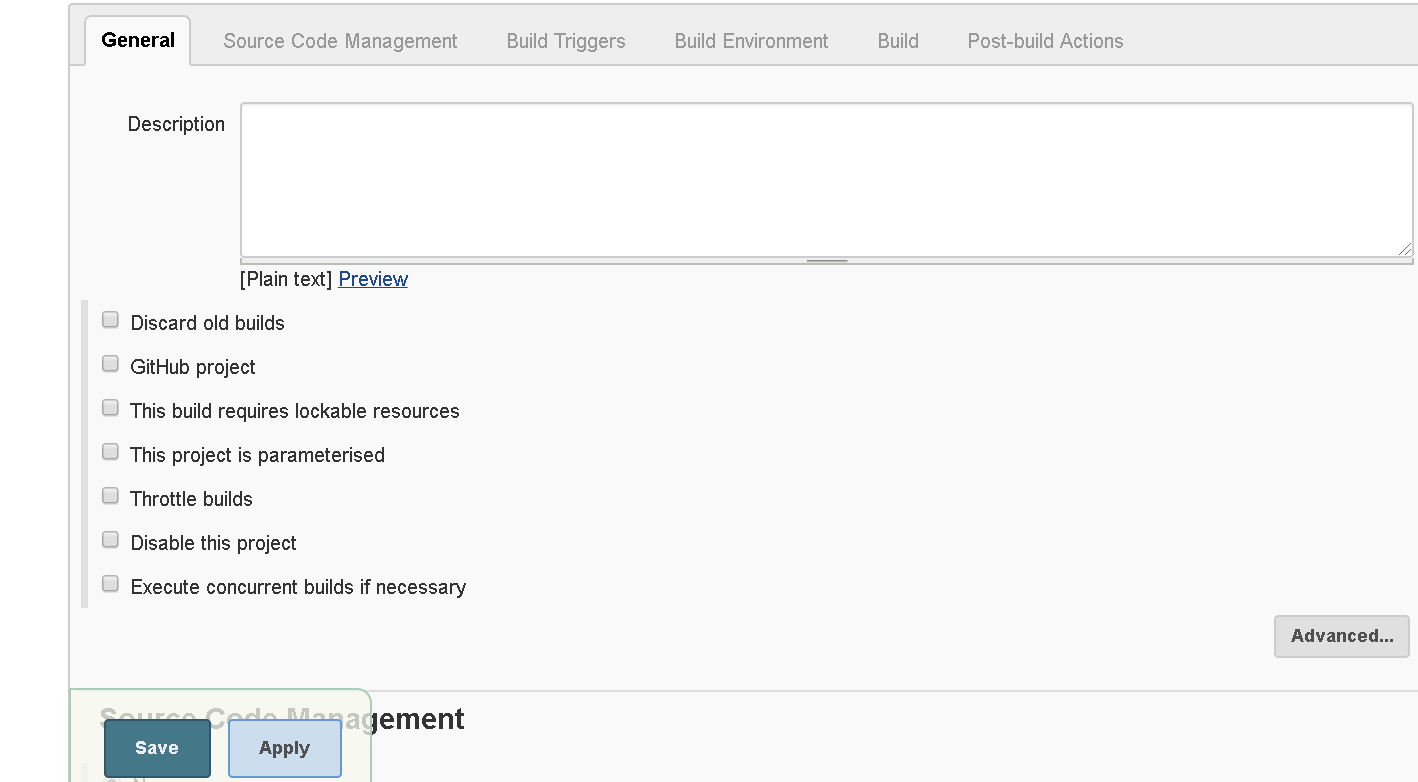
* Web Api re.



* Give a name to the project (Bp Demo in this case), and Select Freestyle Project and Click on Ok.



* Below screen should open up after clicking on Ok.



* Use the below configuration for the Demo in this guide, or add your own config as required.

**Git Hub Project** – Checked, value - [https://github.com/\*\*\*\*/\*\*\*\*\*\*\*\*.git](https://github.com/****/********.git)

**This Project is Parameterised** – Checked

Add Parameters –

ReleaseNumber, Value- release 1 (or what the user provides for the build)

Username, Value- admin (or what the user provides for the build)

Password, Value- \*\*\*\* (or what the user provides for the build)

Environment, Value- choices DEV,UAT,PROD (or what the user provides for the build)

**Source Code management** – GIT Selected

Repository url populated, Value- [https://github.com/\*\*\*\*/\*\*\*\*\*\*\*\*\*.git](https://github.com/****/*********.git) (or what the user provides for the build)

Credentials- None (or what the user provides for the build)

Branch to build- \*/master (or what the user provides for the build)

Additional Behaviours- Clean before checkout

**Build Environment** –

Delete workspace Before build starts selected

Add timestamps to the console output selected

**Build -**

1. Execute windows batch command -

*cd %ReleaseNumber%*

*cd objects*

*objectdeploy.bat %Username% %Password% %Environment%*

1. Execute windows batch command -

*cd %ReleaseNumber%*

*cd processes*

*processdeploy.bat %Username% %Password% %Environment%*

1. Execute windows batch command -

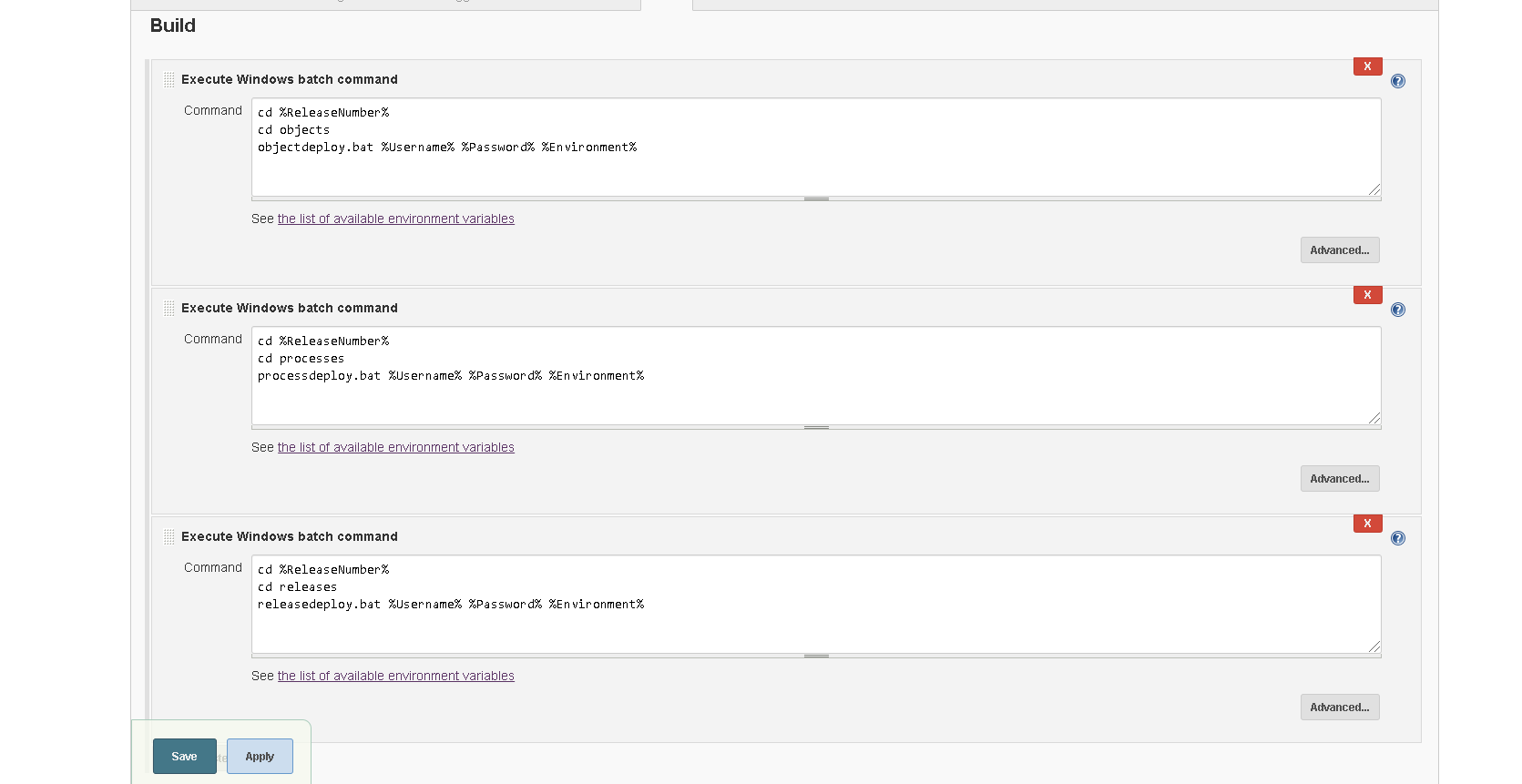
*cd %ReleaseNumber%*

*cd releases*

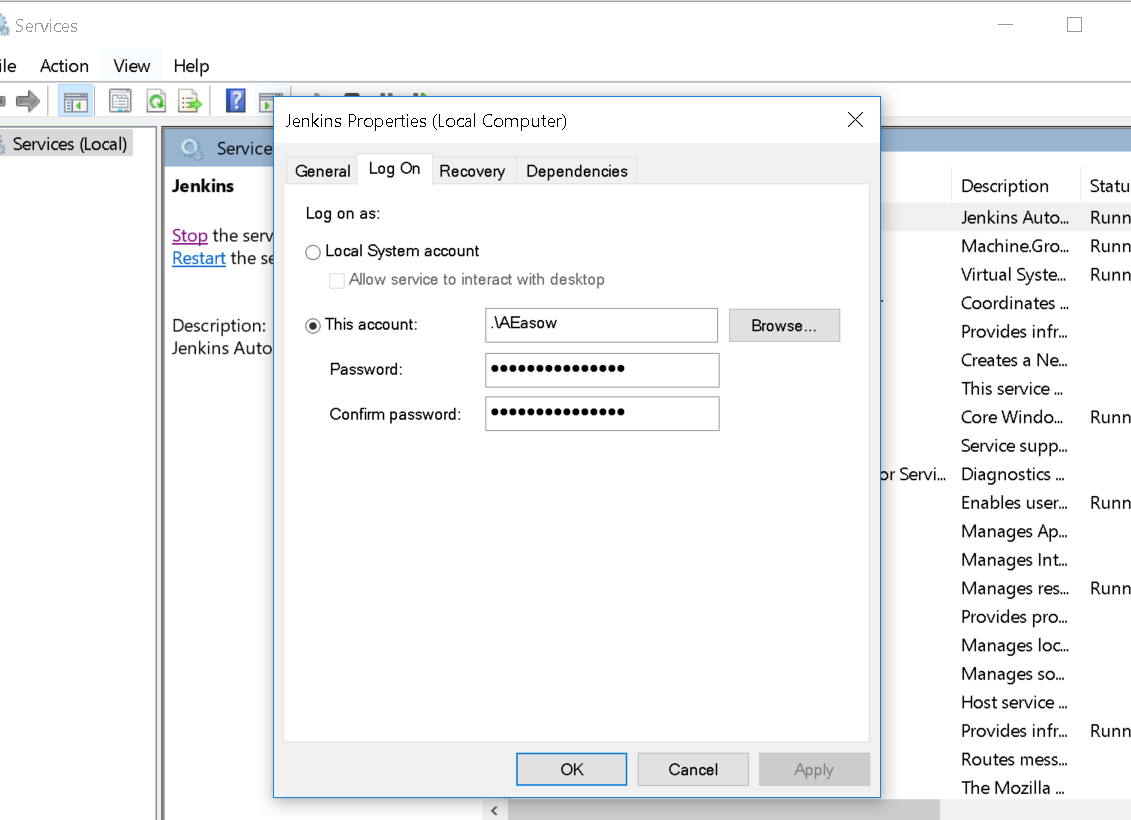
*releasedeploy.bat %Username% %Password% %Environment%*

(user can also define custom scripts as required, or chain multiple build commands as done above).

* Click on Save button at bottom after all configuration changes are done.



* Run Services.msc from windows and configure the Jenkins service to run from a specified user account under which BP is installed.



# Deployment Steps

* Add runtime deployment files to GIT



Above is the GIT directory with all the files, it contains the required folder structure as well as the indvidual bat files paramterised for Jenkins deployment. The choice to add objects, process and release artefacts to a list file is left to the user. For the purpose of this guide each artefact file name is present in the bat files.

* As per the directory structure, unzip and overwrite the files in the local GIT repo created.
* Open GIT bash. And type the below commands:

$ cd /C/GIT/blueprism

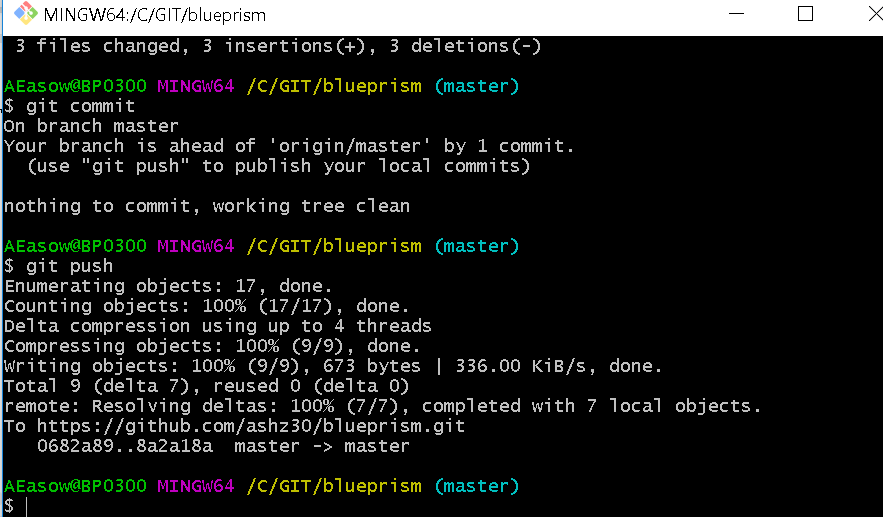
$ git add -A

$ git commit

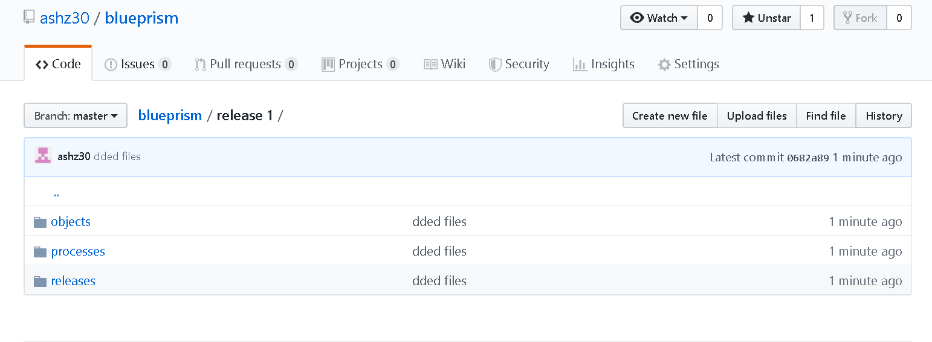
Type comment -> esc -> wq!

$ git push

* Once the push command is issued, you should be able to see something similar to the creen below:



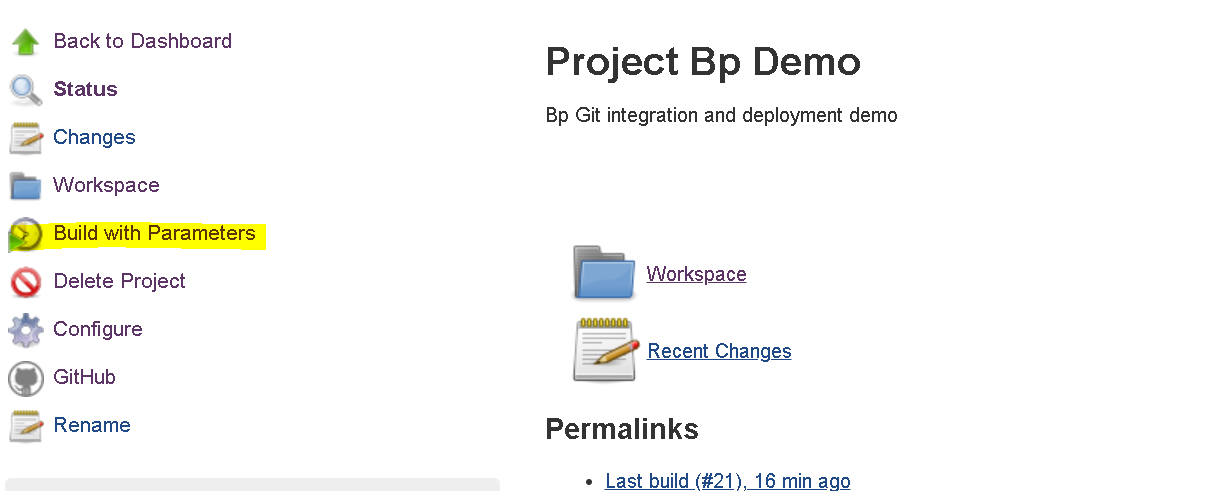
* If you refresh your GIT repository online, you should be able to see all the files added in the GIT GUI:



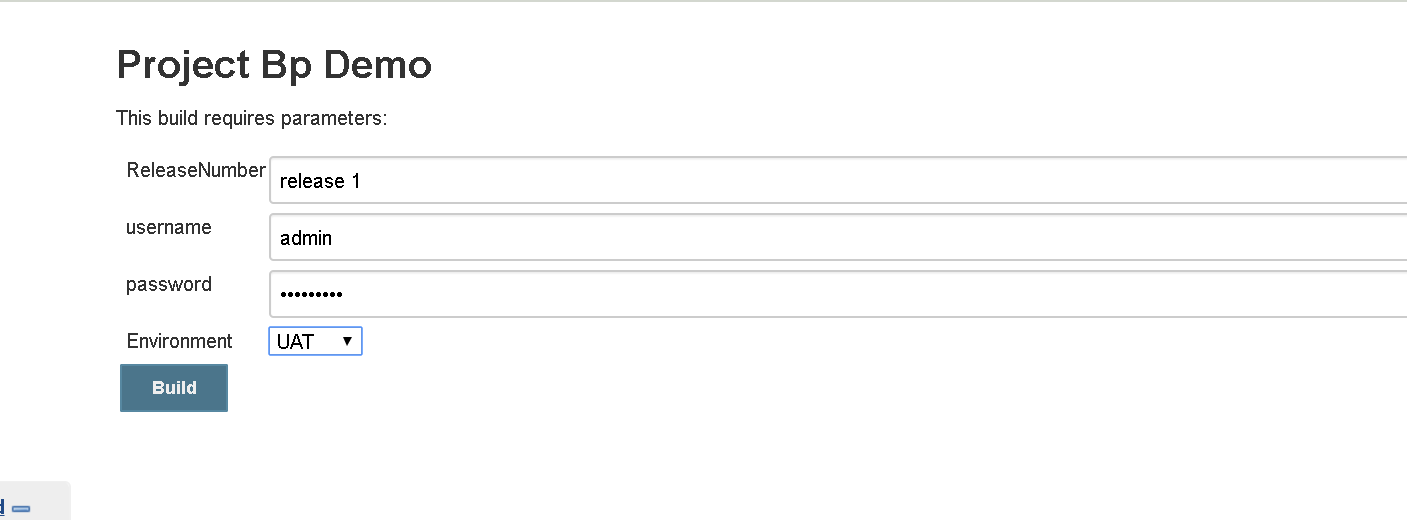
* Create a connection in Jenkins BP install, which points to DEV/UAT/PROD and the connection name should match the choice which has been provided in the Job parameters for Jenkins above. *(Environment, Value- choices DEV,UAT,PROD or what the user provides for the build)*

# Job Creation and Tracking

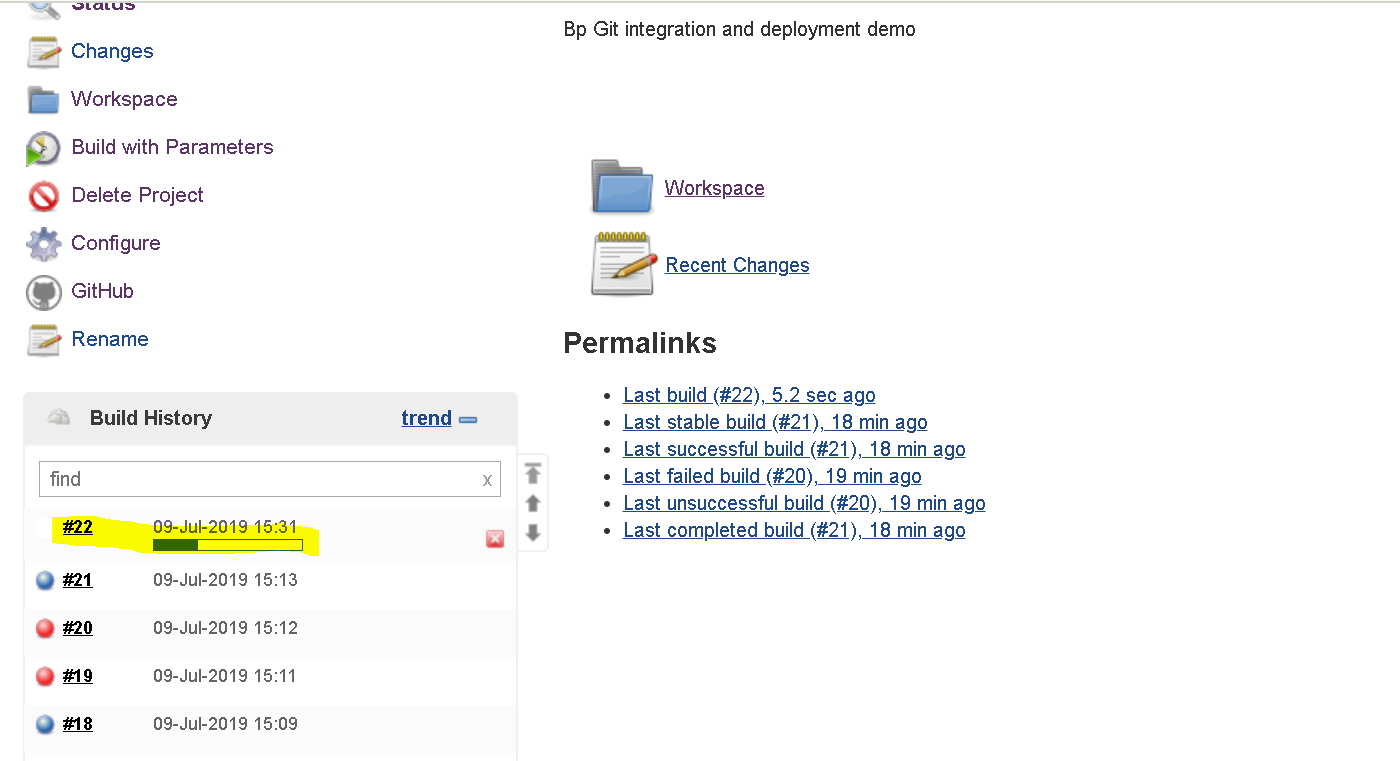
* For creating a runtime Job, log into Jenkins, click on BP Demo (the project which has been configured). And click on Build with parameters option on the left.



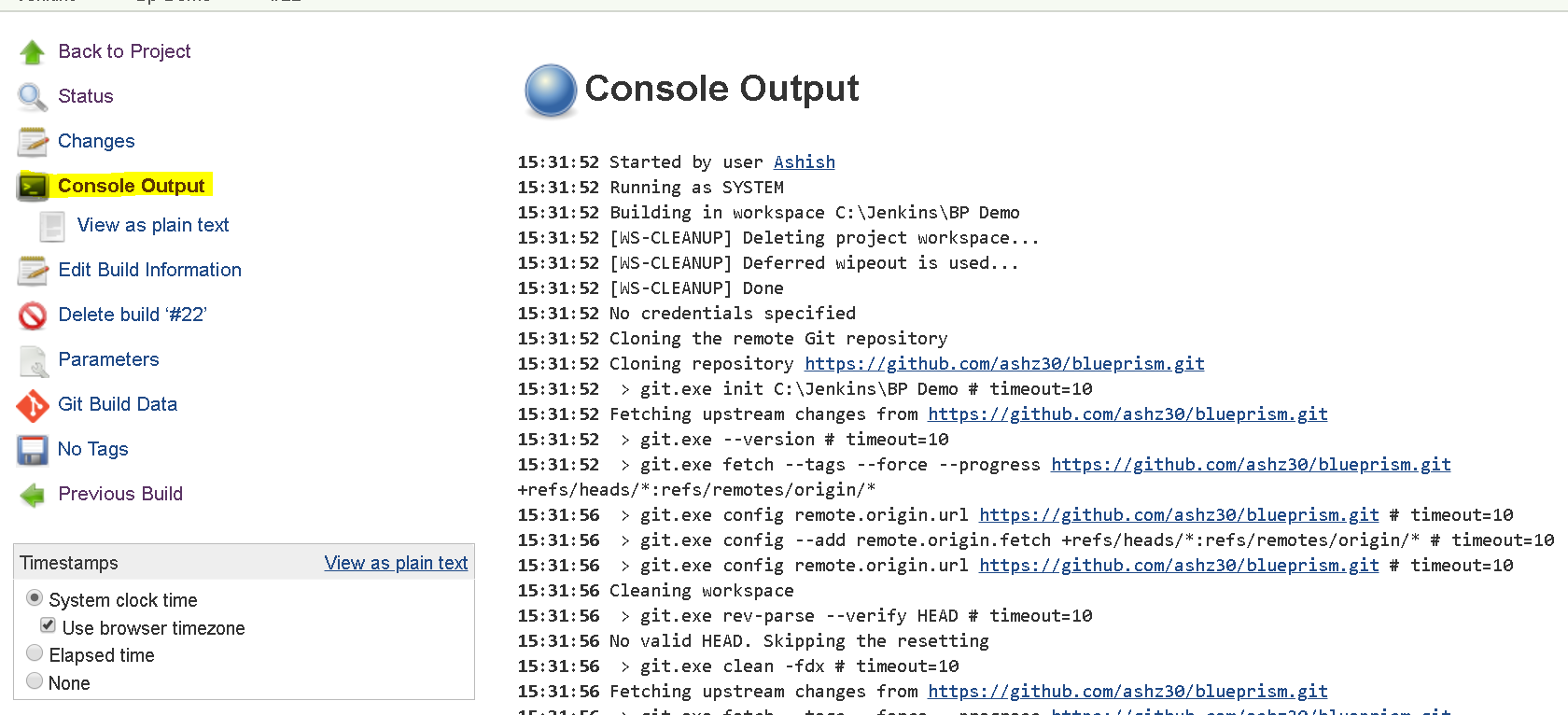
* Override any parameters required and click on build:



* If you refresh the page, you should be able to see your job getting executed.



* Click on the build number and console output on the next page shown:



* The console output gives the status of your current build. Attached is the output of the console output obtained on following this guide.



* Any job run can be viewed and tracked to completion this way and jenkins provides a centralised deployment automation capability for this purpose.

# Conclusion and Additional configuration

The user can add more variations to the build process, by adding more parameters, making the list of deployment artefacts configurable and even making the entire process of extracting the release from UAT environment, committing in GIT and deploying to production automated. Jenkins can also be integrated with other SCM tools and tests can be run (Junit, Selenium, Maven) and reports aggregated. There is considerable collateral present online for all these features for Jenkins integration. I have previously created a pipeline for user approvals for higher environments, it works great.

Users are encouraged to explore.

Today spent in worrying about travel plans for next month,

Your friendly experimental Release Manager,

Ashish Easow