**Elastic Beanstalk (EB)**

**Deployment through CI/CD in App team's perspective**

**Elastic Beanstalk** is a simple and effective AWS service for deploying and scaling web applications.

Elastic Beanstalk solution has been made as a **fully private EB** setup making the environment internal and secured.

**This document covers few easy and best ways to do the Elastic Beanstalk deployment through Jenkins CI/CD.**

Following 4 ways were the identified to perform EB deployment through Jenkins CI/CD:

* AWSEB Deployment Plugin in Jenkins
* Pipeline configuration through script using AWS CLI commands
* AWS CodePipeline Plugin for Jenkins
* AWSElasticBeanstalk Publisher Plugin in Jenkins

**AWSEB Deployment Plugin in Jenkins:**

* This is a direct and simpler approach to be used in Jenkins CI/CD.
* It performs source code bundling (zip), upload to S3 and deploy to AWS EB environment.
* Once this plugin is enabled, we can create our Jenkins job as a Freestyle project or as a pipeline.
* As a freestyle project, we can configure it directly from Jenkins UI by providing the source code github repo details, S3 bucket details to bundle the code and upload, EB environment details along with AWS credentials details.
* As a pipeline project, we can create specifications for the above parameters and write our custom pipeline script to perform the deployment along with every features of pipeline or a direct configuration in UI.

This approach has been tested for sample deployment using freestyle project and pipeline script. Detailed implementation procedure on the same is given at the last section of this document.

References: <https://plugins.jenkins.io/awseb-deployment-plugin/>

<https://www.jenkins.io/doc/pipeline/steps/awseb-deployment-plugin/>

<https://dzone.com/articles/cicd-processes-and-tools-for-aws-elastic-beanstalk>

**Pipeline configuration through script using AWS CLI commands**

* This is another good approach where the pipeline can be configured using pipeline script directly involving AWS CLI or EB CLI commands
* Pipeline script flow can be as simple as git pull stage for code, shell script execution to zip, AWS CLI commands to upload to S3 (if required) and then deploy to EB from S3 or direct deploy of local zip file into EB environment using AWS CLI or EB CCI commands to create application version and update environment.
* It is up to the Application team to decide which method they can adopt.

**AWS CodePipeline plugin for Jenkins**

* This approach also works as a Jenkins plugin.
* This is another good approach where AWS CodePipeline is integrated with GitHub, Jenkins, Elastic Beanstalk.
* A Maven project is created in Jenkins and utilized for this approach.
* Full details on this setup: <https://aws.amazon.com/blogs/devops/building-continuous-deployment-on-aws-with-aws-codepipeline-jenkins-and-aws-elastic-beanstalk/>

**AWSElasticBeanstalk Publisher Plugin in Jenkins**

* This plugin in Jenkins is similar as AWSEB Deployment plugin.
* However, this plugin has currently vulnerable by storing credentials as plain text and Jenkins does not recommend usage of this.

Apart from these 4 possibilities there are other ways in which Jenkins can be configured for Elastic beanstalk deployment. However, the 3 best ways as mentioned here are straightforward and it is up to the Application team requirement that they can adopt any of the configuration setup.

**Restricted IAM role and policies:**

The basic IAM role (with restricted permissions to selected environments and resources) and policies is given in the embedded document below.

Same can be assigned to users/groups with other required S3 permissions.

**Implementation of Elastic Beanstalk deployment through Jenkins CI/CD using AWSEB Deployment Plugin:**

* Freestyle project
* Pipeline project

**Steps to create credentials:**

* Open Jenkins dashboard in console.
* Go to “Manage Jenkins” > “Manage Credentials” > Click on ‘Jenkins’ under ‘Stores scoped to Jenkins’.
* Click on ‘Global Credentials’ domain -> Click ‘Add Credentials’ found at the left. Alternatively you can add a domain and add credentials.

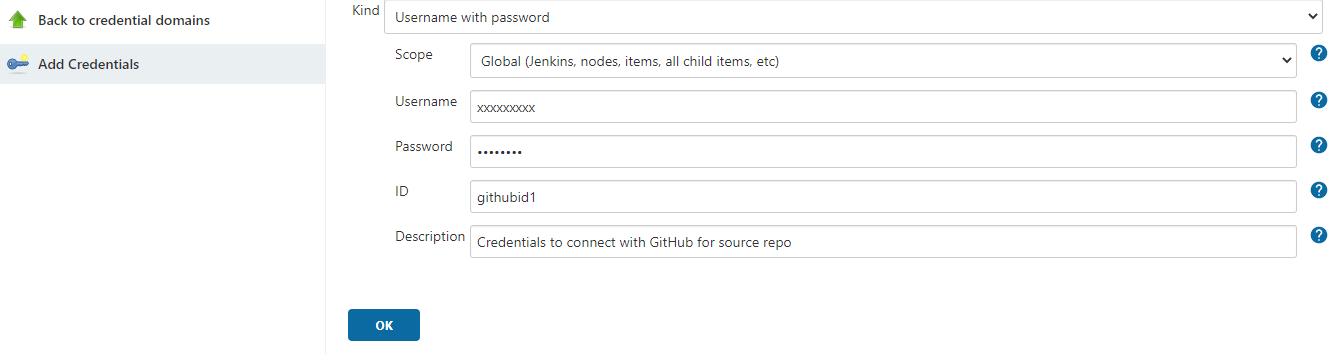
**GitHub credentials:**

Kind – Username with Password

Scope – System (Jenkins and nodes) or Global (Jenkins, nodes, items and child items etc). Choose appropriate scope.

Give an ID name to reference this credential in Jenkins configuration.

Fill in with required details.



**AWS credentials for EB deployment:**

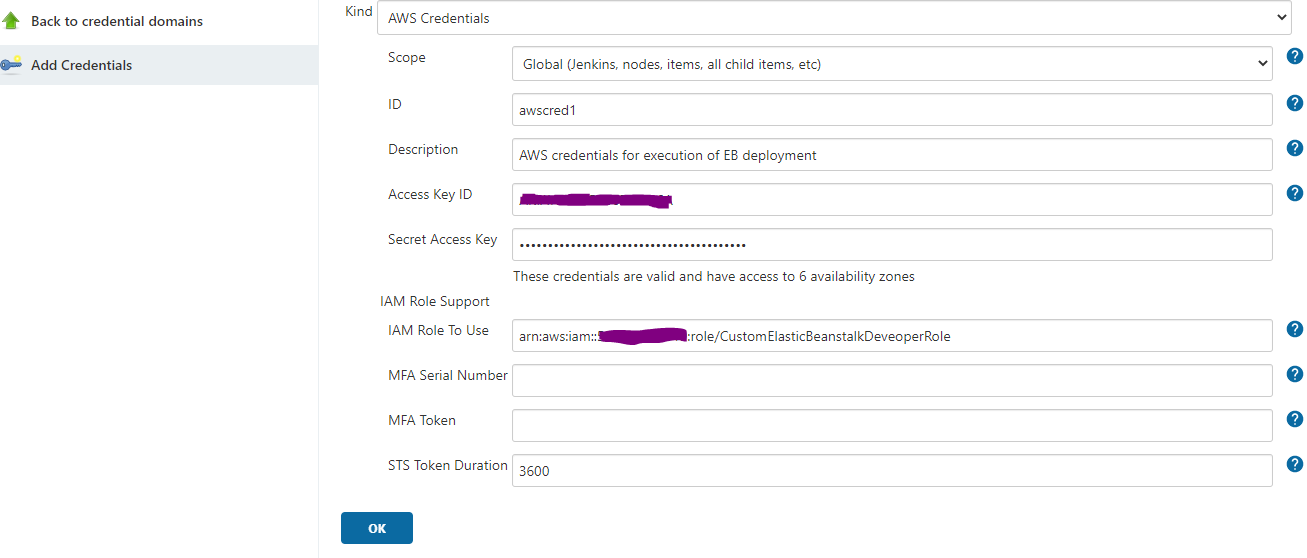
Kind – AWS credentials

Scope – System (Jenkins and nodes) or Global (Jenkins, nodes, items and child items etc). Choose appropriate scope.

IAM role – Provide the complete ARN of the role.

Fill in with required details.

Inorder to make it more secure MFA can be used.



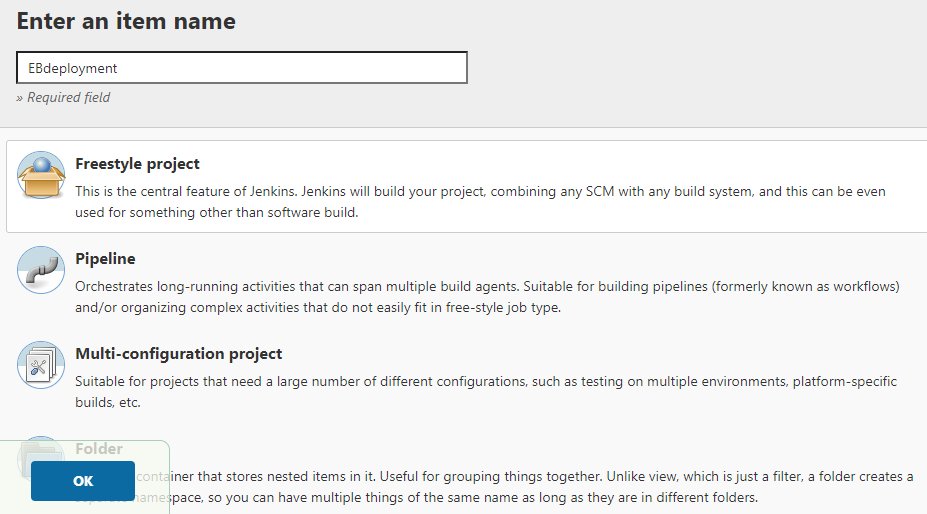
**Freestyle project (Jenkins UI):**

. Open Jenkins UI console.

. Click on ‘New Item’ in the left pane.

. Give appropriate Item name and choose ‘Freestyle project’.

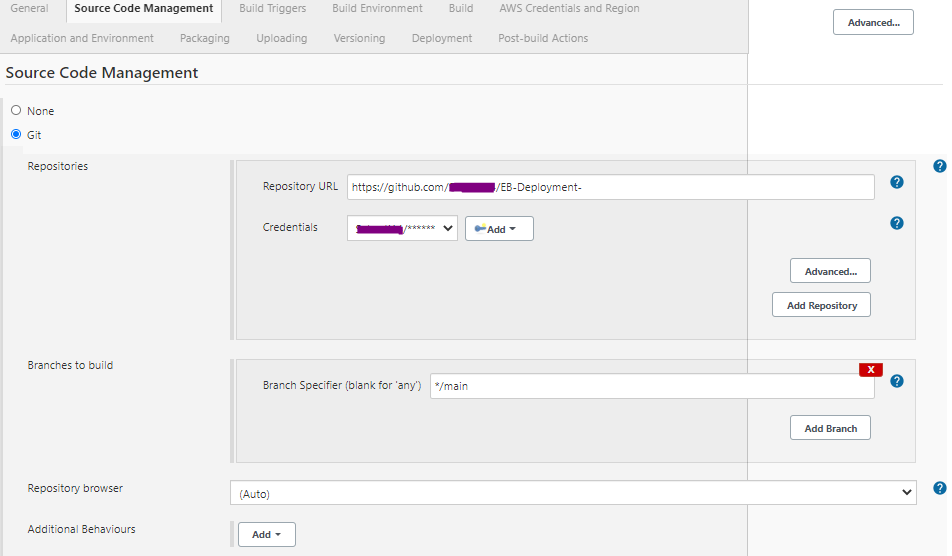
. Click OK.



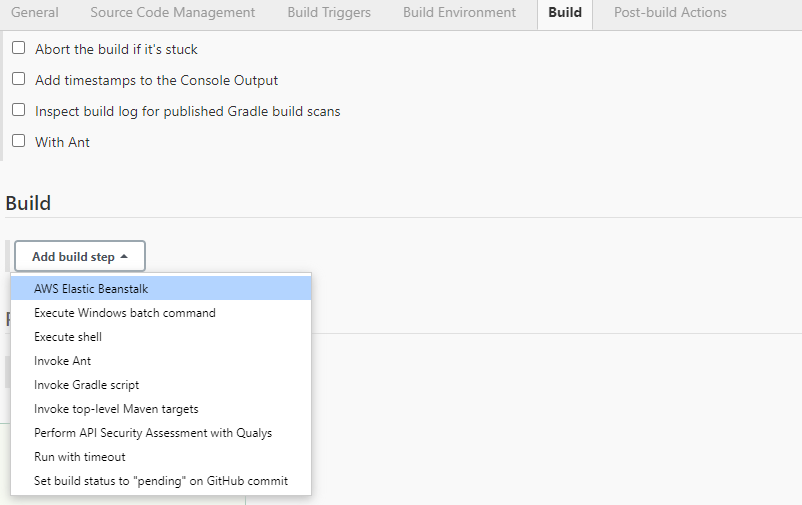
. Configuration page of this project will be opened.

. Provide Description.

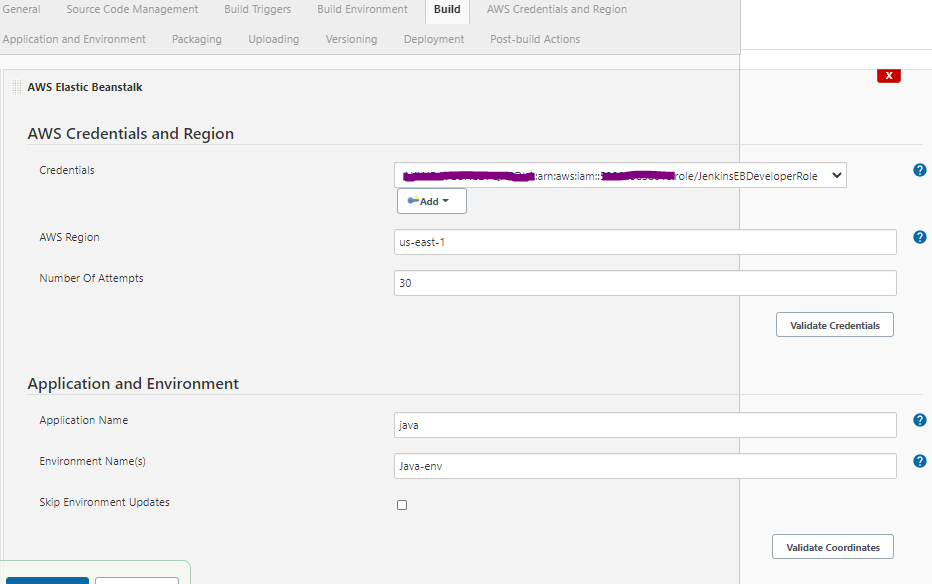
. Under ‘Source Code Management’ section: Choose Git. Provide GitHub repo URL, Choose the GitHub credentials added before, provide branch details.



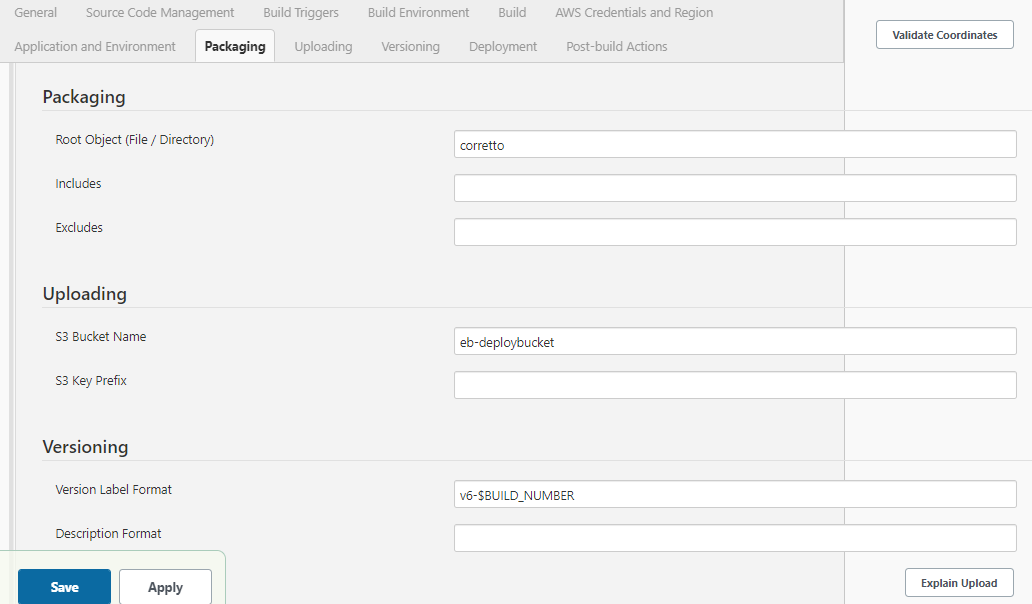
. Under Build section, Click on Dropdown Add build step and Choose AWS ElasticBeanstalk.



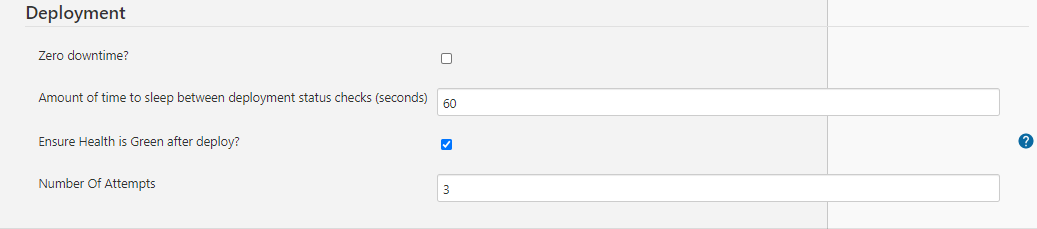
. AWS Elastic Beanstalk configuration tab opens. Provide the AWS configurations and beanstalk environment details.



. Provide the source bundle details, S3 bucket details and application version details.



. Provide details of the deployment

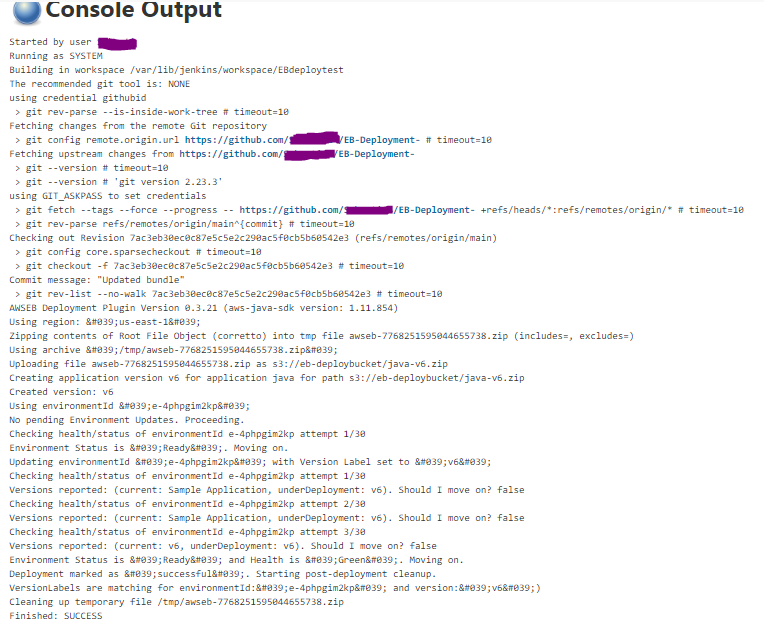


. Click ‘Save’.

Once Project is configured, you can click on “Build Now” option in the left pane to trigger the job.

We can set various other parameters for Git polling, Periodic or scheduled builds, Post build step actions like email notification etc. Configuration options can be explored and utilized further.

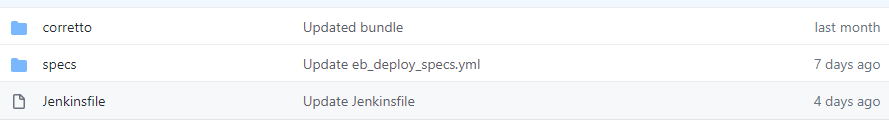
**Sample console output:**



The same process can be followed for new pipeline configuration as Pipeline project.

**Pipeline using script in SCM:**

. Update the GitHub repository where the source bundle directory is created with a File named Jenkinsfile and the specs directory having eb\_deploy\_specs.yml file.

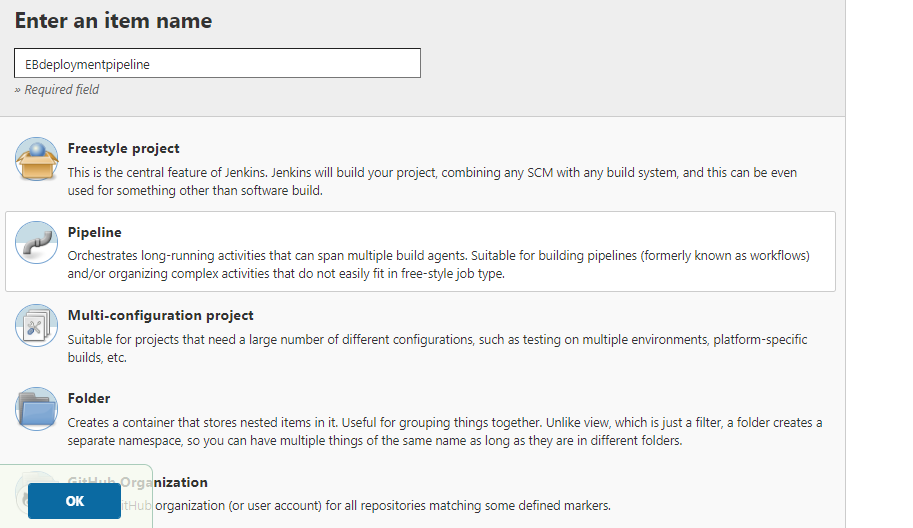


Contents of the Jenkinsfile and eb\_deploy\_specs.yml file is embedded below: 

EB specs file is variablized and hence can be modified accordingly for any environment and comment is given for every value to understand what it is for.

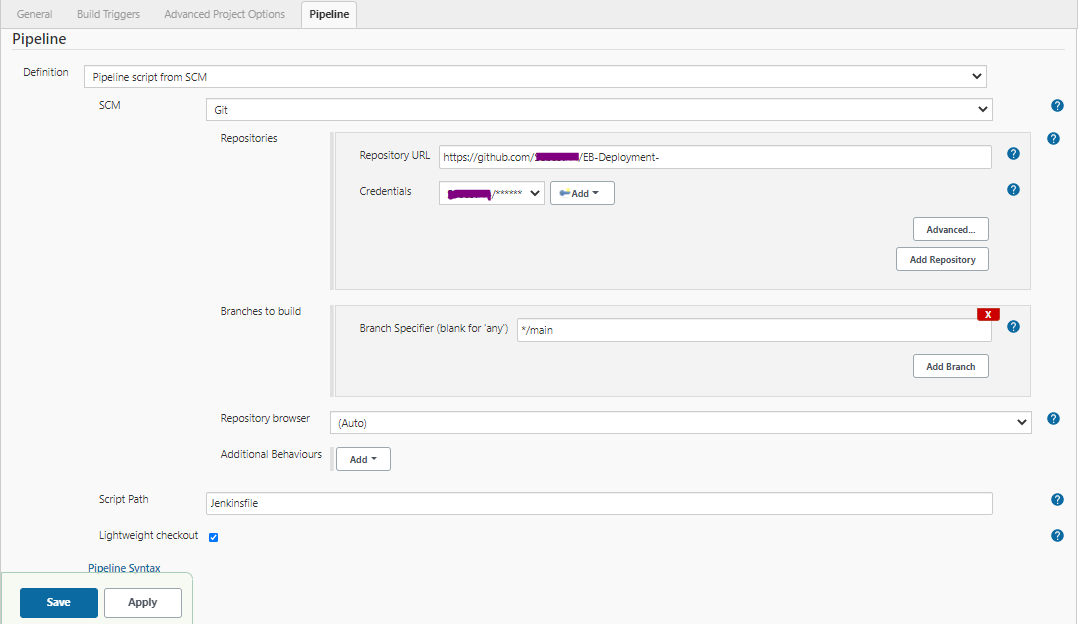
. Go to Jenkins dashboard in the console.

. Click on ‘New Item’. Provide a name and choose ‘Pipeline’. Click OK.



. Go to Pipeline section. Choose Definition as Pipeline script from SCM.

. Provide Git details. Save your configuration.



You can configure other build triggers, post build actions etc as required.