Contents

[Overview 2](#_Toc91813294)

[Hosting Environment 2](#_Toc91813295)

[Problem statement 2](#_Toc91813296)

[Solution 2](#_Toc91813297)

[Implementation 3](#_Toc91813298)

[Existing Process: 3](#_Toc91813299)

[Proposed Process: 3](#_Toc91813300)

[Branch Creation Process and GIT commands: 4](#_Toc91813301)

[Process to Issue the PR to Merge the feature branch to Development. 5](#_Toc91813302)

[Creating Azure Tenant 5](#_Toc91813303)

[Creating Pipelines. 6](#_Toc91813304)

[Creating Build Pipeline. 6](#_Toc91813305)

[Creating Release Pipeline. 9](#_Toc91813306)

# Overview

The perceptive portal is a web-based application under corporate IT and digital solution units. This product is primarily used by enterprise users to create Motion control devices line Gateways, Sensors, Gear boxes, chains, sprockets etc. This portal has the features to maintain enterprises and their users with various roles. This project is developed in ASP.NET MVC, SQL Server, Web API, and jQuery.

# Hosting Environment

This application has UI. Middle tier and Data tier. The UI and Middle tier are hosted in Azure VM. The data tier is hosted in Azure SQL Server VM.

Below are the software installed in Azure VM

1. Dot net Framework 4.5
2. IIS

Below is the software installed in Data tier VM

1. SQL Server 2016

# Problem statement

The Perceptive project release process is manual. Presently, the Development team is preparing the package from local machines using Visual studio 2019 and manually copying the package to the Virtual machine for DEV, QA, Stage and Production deployments. This process is taking long time to release and verification and error prone.

# Solution

Adopt Azure Dev Ops Build and release pipelines to automate deployments.

Azure devops services offers solution to address all kinds of deployment issues. It has the services to Build any project hosted on any repository. It has capability to connect to Git, VSTS, SVN, RTC, Git Hib etc. repositories and create Build and release pipelines.

# Implementation

## Existing Process:

The perceptive portal has following listed technologies.

1. Technology: Asp.Net MVC, Asp.Net Web API and SQL Server.
2. Repository: GitHub. <https://github.com/regalbeloit/Perceptiv-IoT-Platform>
3. Branching Strategy:

No specific branching strategy is followed for release management. Presently an Ad-Hoc branch is identified during sprint planning and all the release features are checked-in to that branch and deployment is given as and wen required during the release.

## Proposed Process:

1. Create Four Branches for four environments.
   1. Develop
   2. Staging
   3. UAT
   4. Master
2. Developers Responsibilities
   1. Create Feature branch from Develop at the start of the sprint.
   2. Make the code changes in the feature branch and unit test.
   3. Create a PR with at least two reviewers to merge the feature branch to Develop branch.
   4. Resolve any Review comments from reviewer.
3. Reviewer responsibility
   1. thoroughly review the newly checked in code with respect to Functionality, logic and coding standards.
   2. Provide review comments appropriately.
   3. Approve the PR if all the code looks good.
4. QA Responsibilities
   1. Approve the Staging Build.
   2. Test the Features in Staging environment to provide signoff
5. Release manager responsibilities.
   1. Approve UAT and Master Releases.
   2. Sync all branches after production deployment.
   3. Make sure Master branch has the production running code always.
   4. Create hotfix branches from Master for any urgent issues and merge the hotfix branch to Master.
   5. After end of sprint, Merge the code from Master branch to downstream branches (UAT, Staging, Develop) if there are any hotfix done to master during the sprint.

# Branch Creation Process and GIT commands:

1. Feature Branch:
   1. Create a User story for all the stories assigned during the sprint planning in Jira portal.
   2. A unique Id will be generated for each User story by Jira.
   3. Create a Branch from Develop with “IoT-“ as prefix to the user story number. For Example, if the user story number is 5919 then the feature branch name can be “IoT-5919”
2. Hotfix branch:
   1. Create a Bug with High as priority field in the JIRA.
   2. Create a branch from Master by prefixing “HotFix-“ to the Bug number generated.
3. Process to create Branch:
   1. Branches can be created with Git Commands from the command line tool as well as UI tools like Visual studio, Source Tree etc.
   2. Below are the git commands to execute from command prompt.
      1. Step 1: Checkout the master branch

***git checkout Develop***

* + 1. Get the latest code from remote branch.

***git pull***

* + 1. Create a feature branch. In the below command IoT-5919 is a feature branch name. This will create branch in local workspace.

***git checkout -b IoT-5919***

* + 1. Once code changes are done, use below command to commit the changes to local workspace with some comment.

***git commit -m “comment”***

* + 1. Use below command to push the committed changes to remote repository. This will create remote repository if not exist. If the remote repository already exist then this will update the repository with new code changes.

***git push***

# Process to Issue the PR to Merge the feature branch to Development.

1. Once the feature branch is pushed to remote repository. We can merge the feature branch to Develop
2. Navigate to Git hub page and sign in with user id/password.
3. Click on the Repository.
4. Git his will display prompt in the repository home page to create a Pull request as should below.
5. Click on Create PR button.
6. The wizard will guide to enter comments, Reviewers and linking the User story to the PR.
7. Once data is provided, click on create PR button at bottom.

# Creating Azure Tenant

It is required to create an Organization (Tenant) in the Azure Dev ops services page in order to setup CI/CD pipeline.

Contacts:

1. James Hopper
2. Mike Heath Brian

Send an email to the above contact to create a tenant in the azure dev ops services page for your project. It is required to request admin access for the resource who is working to set up CI/CD pipeline.

The screen shot after creating a tenant looks like below in the azure portal after creating the organization.

Graphical user interface, text, application

Description automatically generated

Here MCS.Perceptive is the Tenant created for this project.

# Creating Pipelines.

1. Pipelines menu in Azure portal lets us create a Build and release pipeline for a particular project.
2. A pipeline is typically a series of tasks to be executed to create or deploy a package.
3. The pipeline will use an agent job running on a Build machine. Azure provides some hosted agents that can be used for majority of the projects. That way we can avoid creating a build machine.
4. The Build pipeline will create a package using an agent machine and place that package in an agent share location.
5. The release pipeline will copy the package to a target machine/ process to make the release happen.

# Creating Build Pipeline.

1. Creating a new pipeline. Clicking on the Pipeline submenu  will bring the below screen.

A screenshot of a computer

Description automatically generated

1. Click on New Pipeline Button at top right to create a new pipeine. Screen looks like below. Here it is required to choose the Source code repository. Azure pipelines support multiple repositories like Azure Repos, Git, Git Hub, TFS etc. Perceptive is hosted in GitHub repository. Hence choose Git Hub option here.

Graphical user interface, text, application, email

Description automatically generated

1. It is required to connect to Git Hub with credentials to choose the repository as shown below.

Graphical user interface, application

Description automatically generated

1. Click on Browse Button (…) under the Repository in above screen and choose the Repository and the Branch which you need to setup the pipelines.

Once the Branch and repository is selected, the screen shot looks as shown below.

Graphical user interface, application

Description automatically generated

1. Click on Continue button to proceed.

A screenshot of a computer

Description automatically generated

1. The next step is to select the project template. Azure provides various project templates to choose. Based on your project you can select one. As the project is developed in Asp.Net, I have selected Asp.Net in this screen. We can also start from empty template.
2. After the project template selection the screen looks line below.

A screenshot of a computer

Description automatically generated

1. This wizard provides various options to build your project. The majority of the options are self-explanatory. Once the build definition is created with required options then using the Save & Queue menu at top, we can queue a new build that will create a package to deploy.

# Creating Release Pipeline.

1. The release pipeline will copy the Package from Agent machine and copy to the appropriate process. The process can be a virtual machine, Azure App service or any other histing environment.
2. The release pipeline provides an option to provide a Power shell script to run on a target virtual machine that acts as an agent between Azure pipeline and VM to copy the package.
3. Navigate to Deployment groups option from left menu as shown below.

Graphical user interface, text, application

Description automatically generated

1. Click on Copy wizard button to copy the power shell script.
2. Login to the IIS server VM and run the Power shell script as admin.
3. That will create an agent which can be used in release pipeline.
4. Select the Artifacts and Stages of the release as shown below.

A screenshot of a computer

Description automatically generated

1. After release definition is completed, the same will appear under releases and can be triggered as shown below.

Graphical user interface, text, application

Description automatically generated