|  |  |
| --- | --- |
| **Release:** | 2017 |
| **Initiative Name:** | Git version control setup |
|  |  |
|  |  |

**Requirements – Project Summary**

|  |  |
| --- | --- |
| Project Name | Git version control setup, migrate code from SVN |
| Project No.: |  |
| Functional Area |  |
| Request / Originator |  |

**Change Control:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **CR Ref** |
| 06/08/2017 | Document Creation |  | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Resource Listing

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name/Title** | **Phone** | **Email** |
|  |  |  |  |

# Project Overview

## Project Background

- when working for a long time on a ticket and it affects a lot of files, when you are ready to commit

 |\_ svn allows you to commit once

 |\_ git allows logically separate commits which are pushed

- branching takes new meaning

 |\_ ease of creating, merging and deleting branches

 |\_ don't have to think twice about need for creating branches

 |\_ soon developers will create a branch for trivial things

- branches are maintained in one location locally and remotely

 |\_ uncluttered maintenance

- every clone in every developer machine is a backup with full version history

 |\_ reduced risk of single point of failure

- ability to push part of file

 |\_ git add -p

 |\_ select file and follow instructions to first stage sections of file and commit and finally push

- stashing allows saving work for later without committing so you can do a different work on the same or different branch

- git can be used locally for work unrelated to project maintenance

 |\_ take the case of a text file or a batch file or a word document

 |\_ you can create a local repository without bothering to create a remote one

 |\_ you can have versions of the document, branches to have parallel work

- commands are fast to execute mainly because of working locally

 |\_ git clone (with all revision history) takes same time as svn checkout (without history)

## Current Situation

…

## Project Scope/Description

Moving to Git will help with keeping projects and branches uncluttered and allowing Blue Ocean pipeline plugin to be used.

This document deals with [Bonobo Git server](https://bonobogitserver.com/) for ease of [configuring Active Directory](https://bonobogitserver.com/ad-membership/).

### In Scope

### Phase I

The following functionality is in scope for this project:

1. Install Git
2. Install Git server
   1. Configure Active Directory group
   2. Administrator
3. Create repositories
   1. Anonymous read for Jenkins
   2. Block Anonymous write
   3. Assign users allowed to push changes into repositories
4. Prepare batch script to
   1. Clone SVN repository projects
   2. Assign Git repository for a project
   3. Push the project to Git server
5. Other Git servers considered

## Goals and Objectives Including Benefits

|  |  |  |
| --- | --- | --- |
| **#** | **Goals/Objectives** | **Benefits** |
| 1 | Replace SVN with Git | Projects and branches will be uncluttered and Blue Ocean pipeline plugin in Jenkins continuous integration can be used to improve building and deployment process. |

## Success Criteria

This section details what criteria will determine the overall success of the project.

* Success criteria will depend on how much improvement is observed in code management and thus productivity of the IT team.

## Assumptions

This section lists all of the assumptions that are applicable to this project.

* Creating branches will help developer better manage code.

## Risks

Learning curve for developers new to Git.

### Learning Curve Risk

To minimize the Risk, IT development team should communicate with each other actively on how to better use Git. Collective intelligence is better than individual.

## Current Process Flow

This section describes the flow as it currently exists today for checking out SVN projects and committing the changes.

* SVN projects can be checked out either from Eclipse IDE or using a SVN client like TortoiseSVN in a Windows Explorer folder or using SVN command line client.
* Committing the changes is likewise possible from all three said locations.

## Proposed Process Flow

This section describes the flow as it would exist with the implementation of the requirements.

### Install Git

Download and install Git-2.12.2.2-64-bit.exe from [here](https://git-scm.com/download/win)

### Install Git server

This page covers simple Bonobo Git Server installation.

Download and install Bonobo Git Server from [here](https://bonobogitserver.com/)

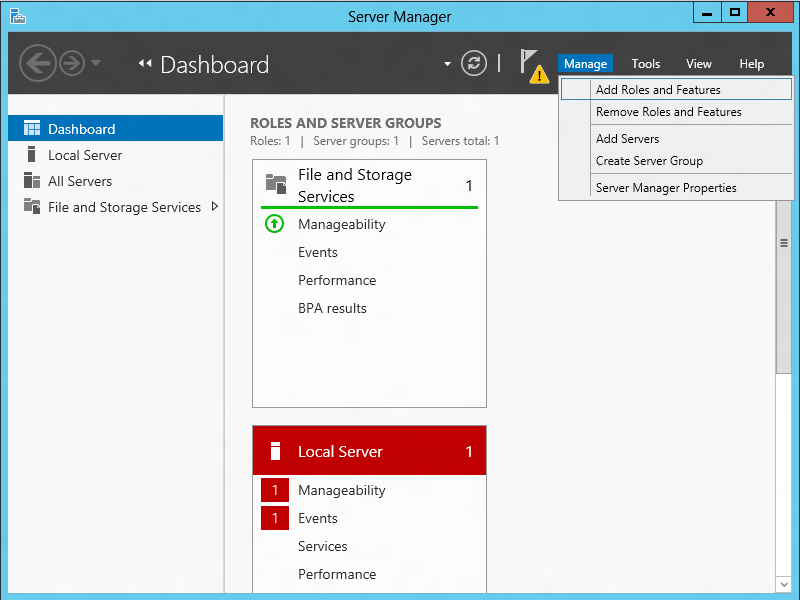
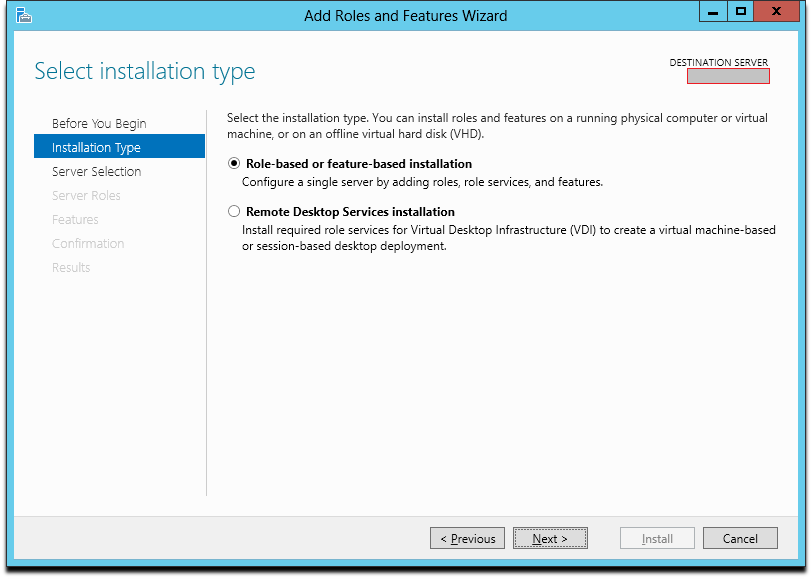
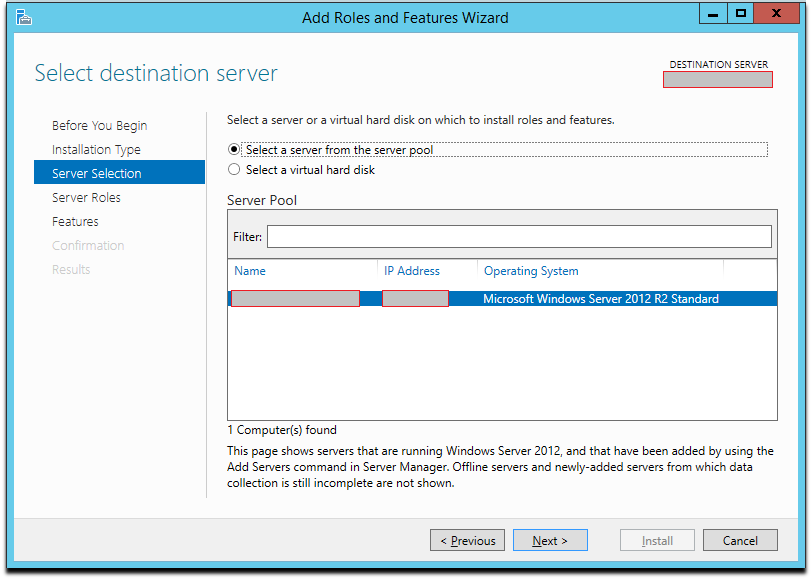
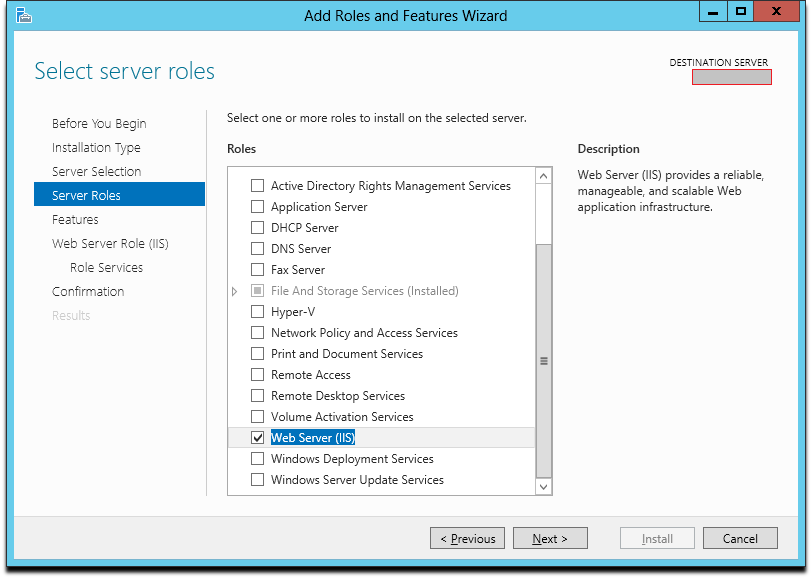
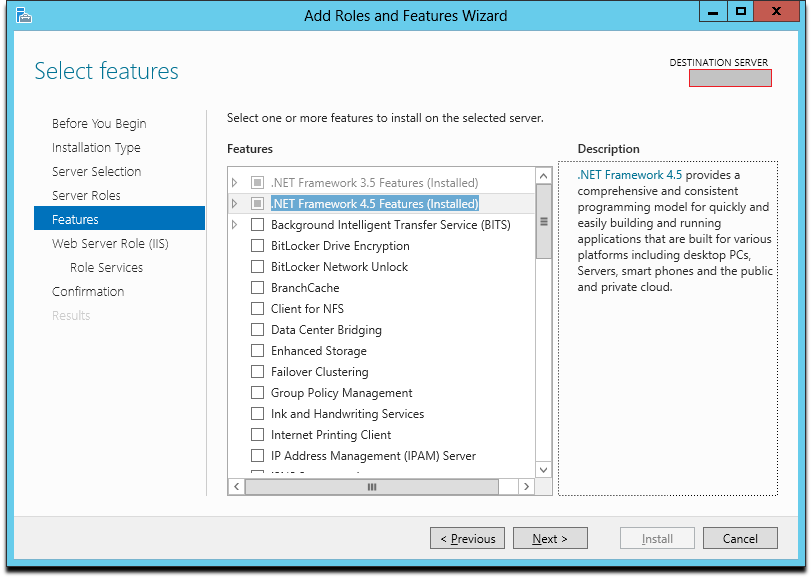
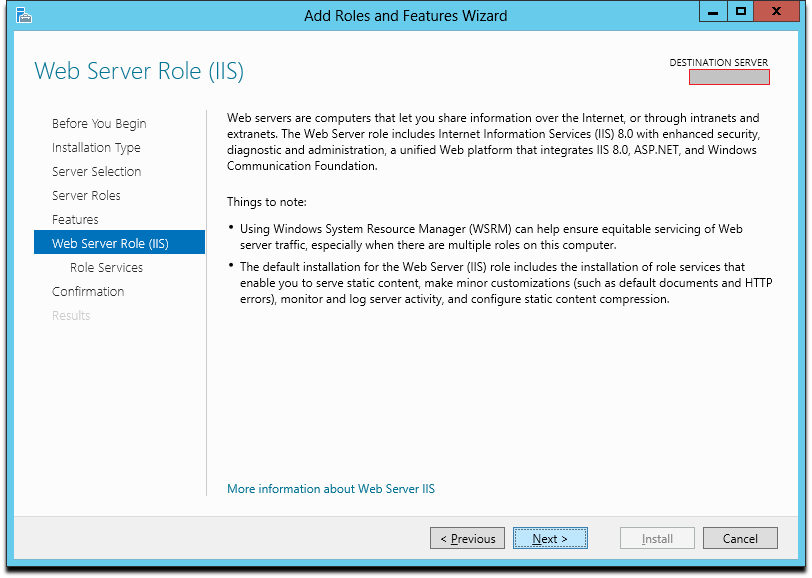
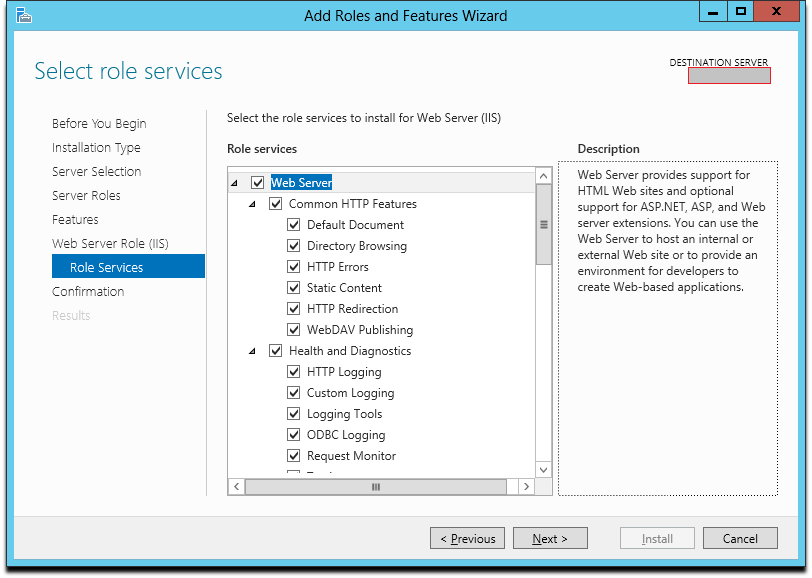
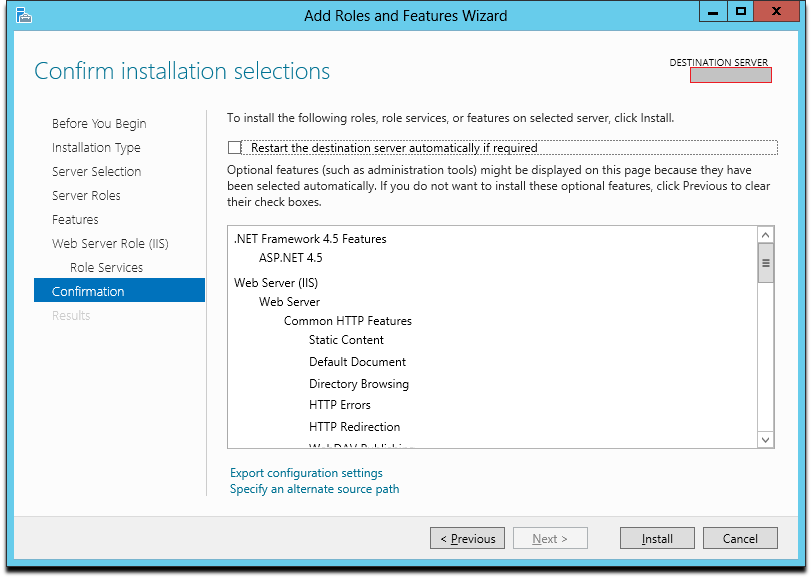
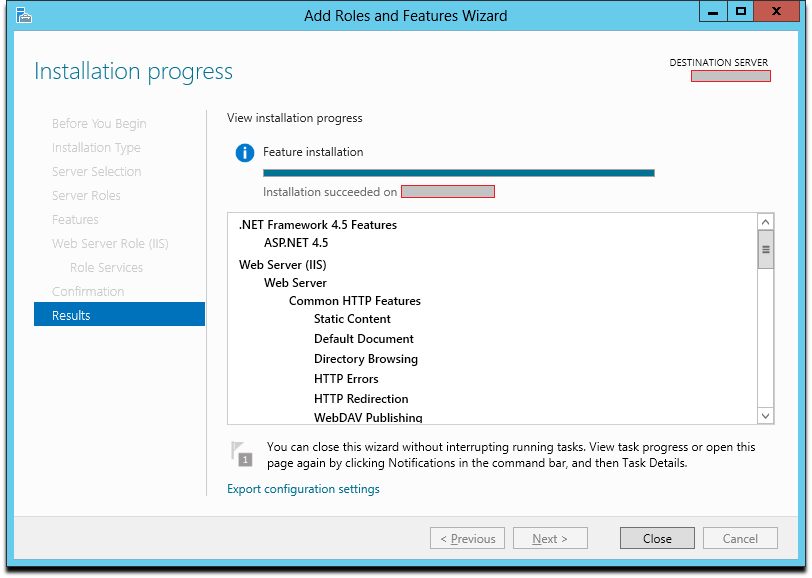
#### Prerequisites

Operating System: Windows 2012 Server.

Prerequisites are [.NET Framework 4.6](https://www.microsoft.com/en-gb/download/details.aspx?id=48130), [IIS 8](https://docs.microsoft.com/en-us/iis/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012)

For other version of Windows see [here](https://bonobogitserver.com/prerequisites/)

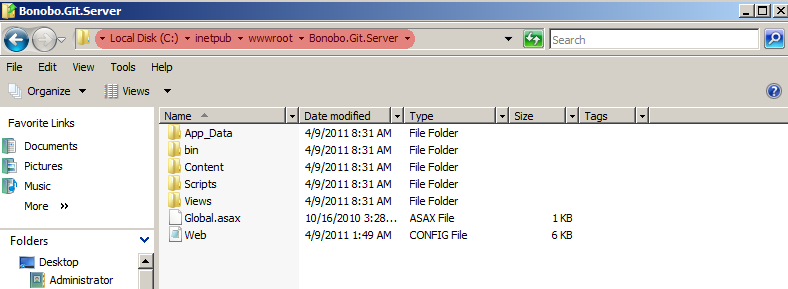
#### Install IIS in Windows

1. Open **Server Manager**.
2. Under **Manage** menu, select **Add Roles and Features**:  
   [](https://docs.microsoft.com/en-us/iis/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012/_static/image1.png)
3. Select **Role-based or Feature-based Installation**:  
   
4. Select the appropriate server (local is selected by default), as shown below:  
   
5. Select **Web Server (IIS)**:  
   
6. No additional features are needed for IIS, so click **Next**:  
   
7. Click **Next**:  
   
8. Customize your installation of IIS, or accept the default settings that have already been selected for you, and then click **Next**:  
   
9. Click **Install**:  
   
10. When the IIS installation completes, the wizard reflects the installation status:  
    
11. Click **Close** to exit the wizard.

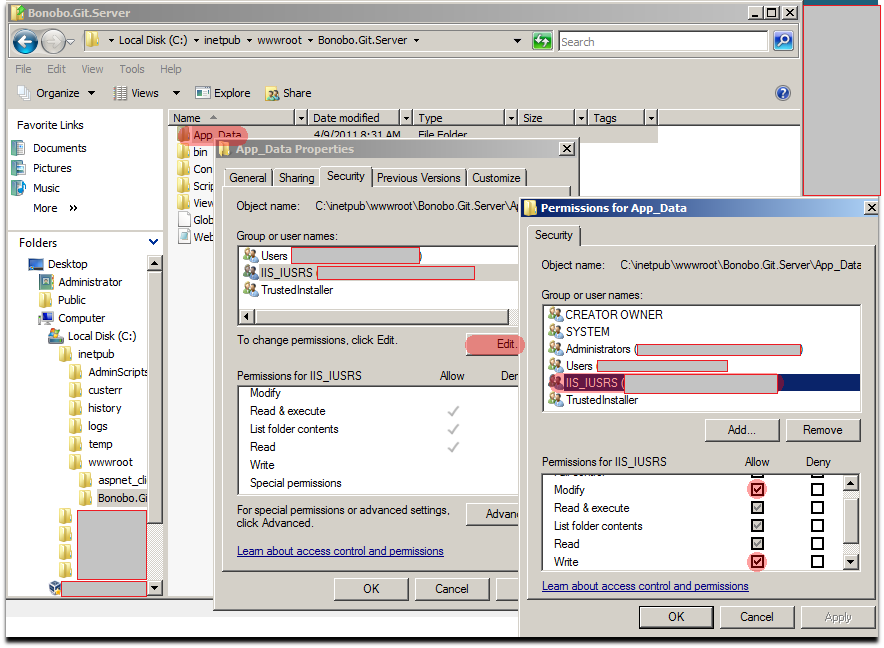
#### Install Bonobo Git Server

The following steps covers an installation with Windows 2008 Server and IIS 7. They are exactly the same for any higher platforms (Windows Server 2012 and IIS 8.0+).

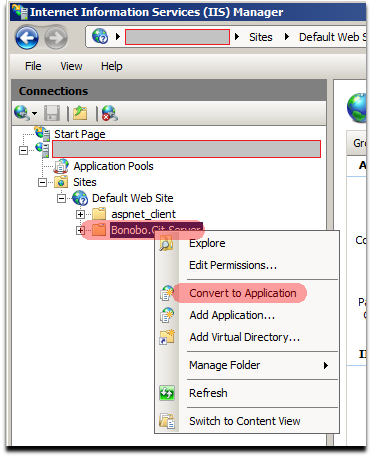
* [Download](https://bonobogitserver.com/) the latest version of Bonobo Git Server from the front page
* **Extract the files** from the installation archive to C:\inetpub\wwwroot



* **Allow IIS User to modify** C:\inetpub\wwwroot\Bonobo.Git.Server\App\_Data folder. To do so
  + select Properties of App\_Data folder,
  + go to Security tab,
  + click edit,
  + select IIS user (in my case IIS\_IUSRS) and add Modify and Write permission,
  + confirm these settings with Apply button.

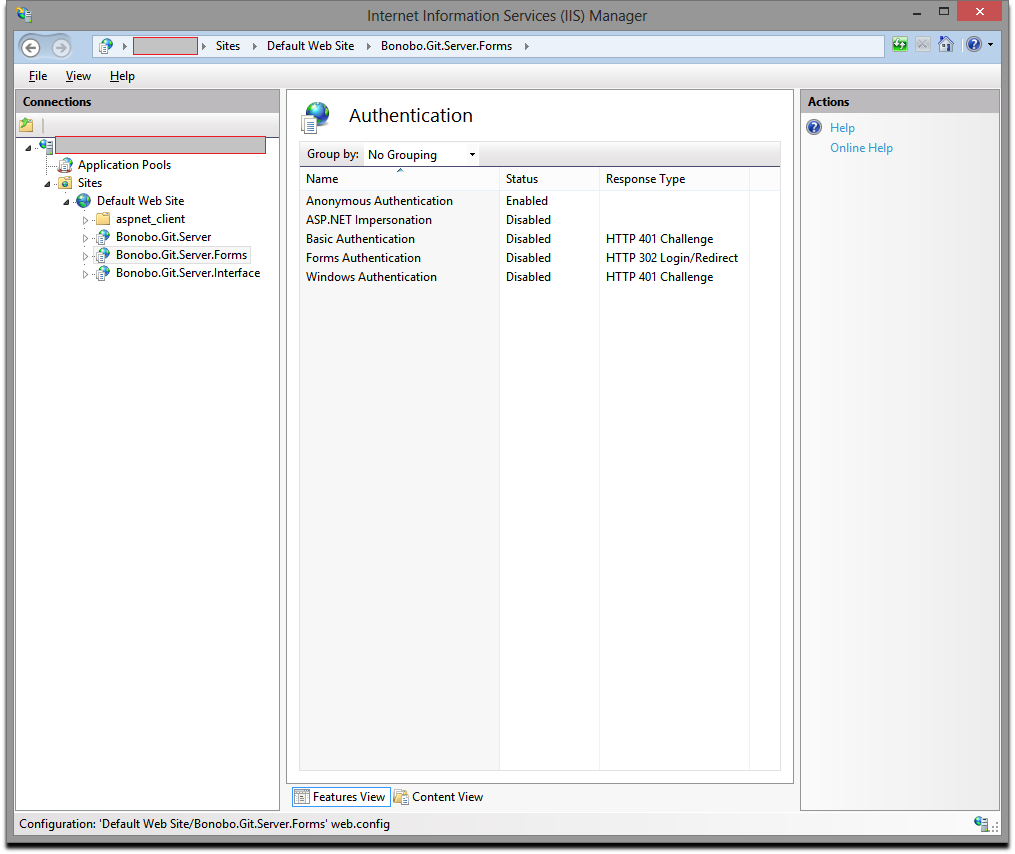


* **Convert Bonobo.Git.Server to Application** in IIS
  + Run IIS Manager and navigate to Sites -> Default Web Site. You should see Bonobo.Git.Server.
  + Right click on Bonobo Git Server and convert to application.
  + Check if the selected application pool runs on .NET 4.0 and convert the site.

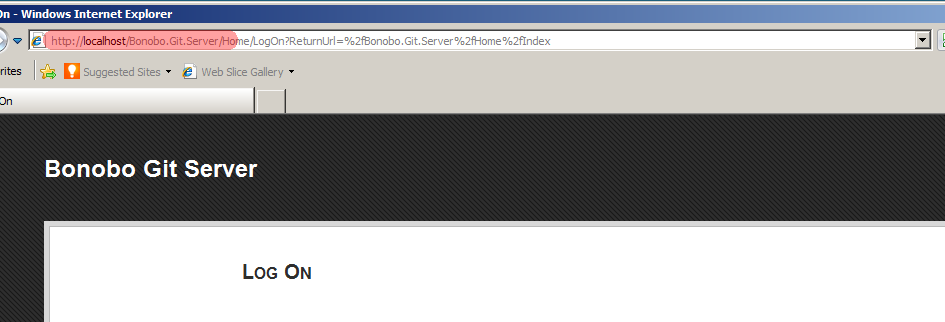


* **Configure Authentication**

Enable **Anonymous Authentication** in IIS and disable the others. To do so, select the application, click on the authentication icon and set the value to of Anonymous Authentication to Enabled. The configuration should look like the following screenshot.



* **Launch your browser** and go to <http://localhost/Bonobo.Git.Server>. Now you can see the initial page of Bonobo Git Server and everything is working.
  + Default credentials are username: **admin** password: **admin**



#### Configure Active Directory group

In order to use the Active Directory Membership Service, change the *MembershipService* value in the C:\inetpub\wwwroot\Bonobo.Git.Server\**web.config** file located in the root of the application according to the following lines.

**<appSettings>**

...

**<add** key="MembershipService" value="ActiveDirectory" **/>**

**<add** key="ActiveDirectoryBackendPath" value="~\App\_Data\ADBackend" **/>**

**<add** key="ActiveDirectoryMemberGroupName" value="Git" **/>**

**<add** key="ActiveDirectoryTeamMapping" value="Team1=GitTeam1, Team2=GitTeam2" **/>**

**<add** key="ActiveDirectoryRoleMapping" value="Administrator=GitAdmins" **/>**

...

**</appSettings>**

Refer to email screenshot attached below to the conversation about creating AD groups.

Comment the *ActiveDirectoryTeamMapping* section. This can be used later if needed.

Next, set the *ActiveDirectoryMemberGroupName*.

Finally, set up the *ActiveDirectoryRoleMapping*.

Reference: <https://bonobogitserver.com/ad-membership/>

#### Administrator

All the members of the group configured in *ActiveDirectoryRoleMapping* have administrator privileges.

### Accessing Git Server UI

Access to the Git UI is using http://< domain >:8070/Bonobo.Git.Server

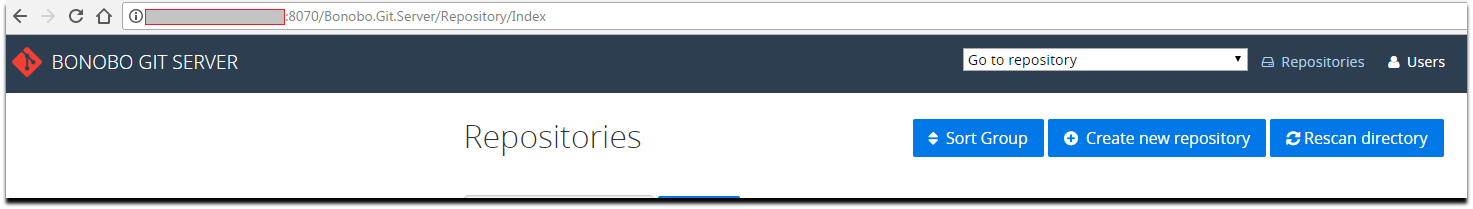
### Accessing Git Server repository

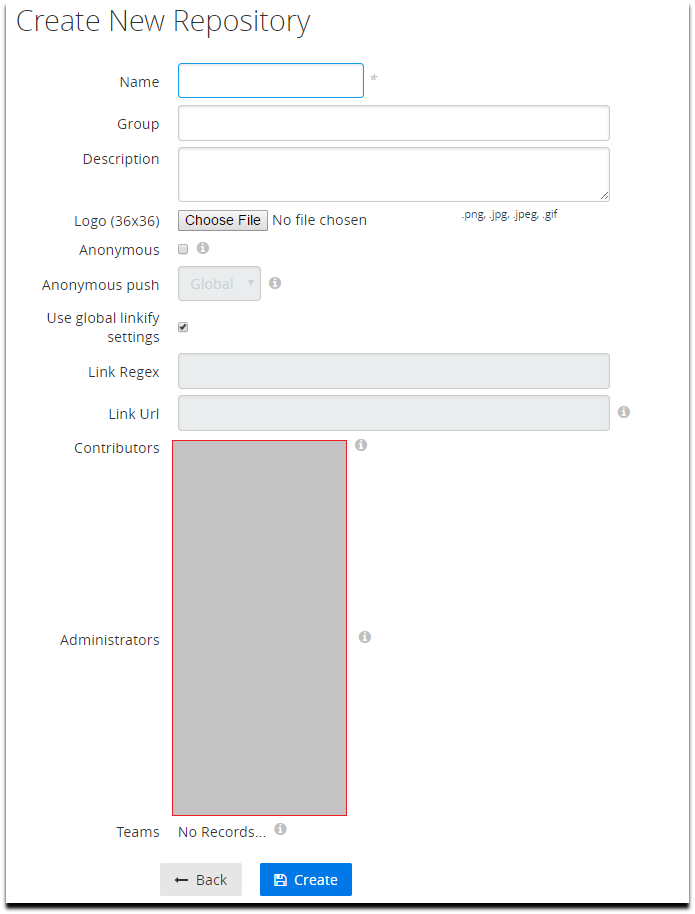
Access to the Git repositories is using http://<domain>:8070/Bonobo.Git.Server/<repository\_name>.git

### Create repositories

Login to http://< domain >:8070/Bonobo.Git.Server

Click on **Create New Repository** button.



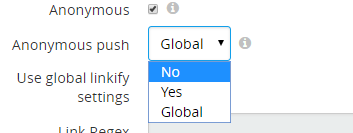


#### Anonymous read for Jenkins

If the project is built in Jenkins, enable **Anonymous** check box.

#### Block Anonymous write

If **Anonymous** check box is selected, select **No** in **Anonymous push** dropdown selection.



#### Assign users allowed to push changes into repositories

**Contributors** are users allowed to push changes into the repository. Select check boxes of the list of users.

These users are part of the Active Directory group configured in **ActiveDirectoryMemberGroupName** in the file C:\inetpub\wwwroot\Bonobo.Git.Server\**web.config**, in our case its **GitMembers**.

### Prepare batch script

Find below batch script used for migrating projects from SVN to Git. Rename the extension of the file to .bat before executing.



Steps involved in the script are listed below. As an example, a project is used to illustrate the commands.

#### Clone SVN repository projects

Command used in the batch file to clone a project.

git svn clone <svn repository url>/Project1 "C:\svn2git-dump\Project1" –no-metadata -A "C:\svn2git\authors-transform.txt"

#### Assign Git repository for a project

Command used in the batch file to assign Git repository for a project.

git remote add origin <git url>/Project1.git

#### Push the project to Git server

Command used in the batch file to push the project to Git server.

git push -u origin master

#### History

Version history of the projects are intact using this process.

### Git tutorial

Refer [https://git-scm.com/docs/gittutorial](https://git-scm.com/docs/gittutorial%20)

### Other Git servers considered

Here are the list of other Git servers tried.

* + - Gogs
    - BitBucket

## Responsibilities

This section lists all of the individuals who are responsible for various tasks for the project.

|  |  |
| --- | --- |
| **Individual** | **Responsibility** |
|  | setup |

## Testing

This section lists all of the individuals who identified for the testing of the implemented functionality and their area of responsibility.

|  |  |
| --- | --- |
| **Individual** | **Responsibility** |
|  | Test integration with Eclipse, Jenkins, Blue Ocean |

## Training

This section pertains to the support the training team shall provide for the project.

|  |  |
| --- | --- |
| **Individual(s)** | **Training Deliverable/Activity** |
|  | Document, support |

### Post Implementation Lessons Learned

This section lists all of the lessons learned on the project

1. It’s better to choose a product (Bonobo Git Server) that meets the basic needs than to choose a product (Atlassian Bitbucket Git Server) that offers extra features but is slower in performance.
2. Git doesn’t like empty folders. Empty folders cannot be migrated from SVN projects to Git. To do this create dummy text files in those folders and then proceed to migrate.

## Project Implementation Success Measures

Successful completion of the below items will constitute that this project met the stated business requirements and satisfied the goals/objectives as stated in the respective section above.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Description** | **Comments** | **Complete (Y/N)** |
| 1 | Code management |  |  |
| 2 | Eclipse workspace management |  |  |
| 3 | Productivity |  |  |