Using git to publish content

\*\*\*READ Everything on the screen first\*\*\*

\*\*\*The scripts provide instructions, examples and post completion instructions\*\*\*\*

**Pre create-deploy steps**

Deploying websites this way requires the use of a git client.

To download Git go here

<http://git-scm.com/downloads>

once at the website click on your operating system link and the download will start once git is installed you want to download a git graphical client.

The most popular client used is smartgit which can be downloaded here

[**http://www.syntevo.com/smartgit/download.html**](http://www.syntevo.com/smartgit/download.html)

If you want to download the client for another operating system, you can go here.

[**http://www.syntevo.com/smartgit/download.html?all=true**](http://www.syntevo.com/smartgit/download.html?all=true)

The user that is using smartgit to deploy must be in the deploy group and have their shared key passed to the web server you are deploying to. As long as this is true you will be able to publish content.

If there are any issues pushing to one of the remote repositories follow the below instructions on forcing a push.

To add the user to the group just vi the /etc/group file.

To pass the shared key to the web servers you must log in as root on the server then run the following commands, replacing username for the user you are updating.

|  |
| --- |
| su – username |

Now create the key

|  |
| --- |
| ssh-keygen -t rsa -b 5120 |

Now make sure the permissions are correct for the user

|  |
| --- |
| chmod 700 .ssh  chmod 600 .ssh/id\_rsa  chmod 644 .ssh/id\_rsa.pub  chmod 600 .ssh/id\_dsa  chmod 644 .ssh/id\_dsa.pub  chmod 644 .ssh/authorized\_keys |

Next you put the files on the server

|  |
| --- |
| cd ~/.ssh  sftp gituser@ipaddress  put id\_rsa.pub  exit |

Next you import the key on the remote server.

|  |
| --- |
| ssh gituser[@ipaddress](mailto:root@ipaddress)  cat id\_rsa.pub >> .ssh/authorized\_keys |

Finally you make sure the permissions are correct on the remote server

|  |
| --- |
| chmod 700 .ssh  chmod 600 .ssh/id\_rsa  chmod 644 .ssh/id\_rsa.pub  chmod 600 .ssh/id\_dsa  chmod 644 .ssh/id\_dsa.pub  chmod 644 .ssh/authorized\_keys |

**Forcing a Push**

1. First open git from the start menu( Not smartgit).
2. Next click “<ctrl>+p” on your keyboard to open the push menu.
3. Choose the destination repository you are trying to push to.
4. Check Force overwrite existing branch.
5. Finally click the “Push” button.

This will overwrite the remote branch you are pushing to with what you have locally.

**create-deploy steps**

To create a deploy repo where you can run a commit on it and it pushes to the production server just run the following commands on the git server and follow the instructions

First ssh into the git server using the deploy user.

Make sure there is not already a git repo in /deploy

Then run the below commands.

|  |
| --- |
| cd /deploy/deploy-scripts  sh ./create-deploy.sh |

You will need to know if it’s on webe web4 or web5 and you will need the full path to the site as it is on the web server. The path is case sensitive so you should log into the web server via ssh first to get the path.

Once the repo is created you can use smart git. To publish you pull the content from the dev repo Switch to the deploy repo

Commit the changes and push to the deploy repo.

**To deploy the submodules**

First ssh into the git server using the deploy user.

Then run the below commands.

|  |
| --- |
| cd /deploy/deploy-scripts  sh ./deploy-lead.sh |

You will need to know what server you are deploying to, the short name of the server to keep the files easier to read, and you will need the full path to the sites submodule as it is on the web server. The path is case sensitive so you should log into the web server via ssh first to get the path.

To deploy to do a mass deploy of the submodule to all sites that have been previously deployed using these scripts

First ssh into the git server using the deploy user.

Then run the below commands.

|  |
| --- |
| cd /deploy/deploy-scripts  sh ./deploy-lead-all.sh |

This uses the text files in /deploy/deploy-scripts/modules/sites to get the information it needs so there are no questions to answer.

These files where updated by the deploy-lead.sh script when you first deployed the submodule to the site using these scripts.

If you want to provide your own list you can temporarily move the text files to another location and place a file in the directory that contains

Web server name<space >module full path

The file will also need to have the .txt extension. That rule was scripted in as a precaution.

**Fin.**