Hands-on Labs Content Production and Publication

This guide provides instructions to request access, connect to the content repository in Visual Studio Online (VSO), and perform the actions required to begin work as a content author.

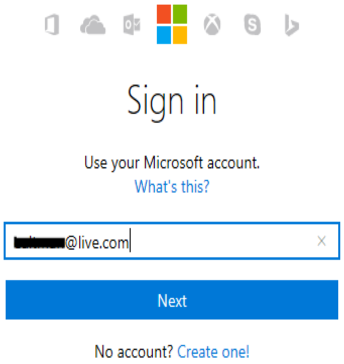
These instructions demonstrate both the Visual Studio Online interface and the Git command-line tool, though you may translate these operations into any tool you prefer.

Requesting Access to Become a Content Author

All content in the Hands-on Labs platform is stored in a Git repository hosted by Visual Studio Online. The Content-Private repository is available to content developers and provides a version control system for managing changes to content packages.

Contact Valorem and Verify Repository Access

1. Contact Valorem Hands-on Labs at holcontent@valorem.com to request access to the Content-Private repository. Provide the following information in your request:
    a. Microsoft Live ID associated with an MSDN or VSTS license
    b. Content author's name
    c. Content Package and/or Lab names, and a brief synopsis of the lab goal
    d. Vendor name
    e. Content package primary purpose: self-paced, workshop or demo
2. Once you receive your invitation to connect, login to the Git repository for Hands-on Labs Content: Content-Private
    a. Use the Live ID you provided in step 1 as the credentials to connect.

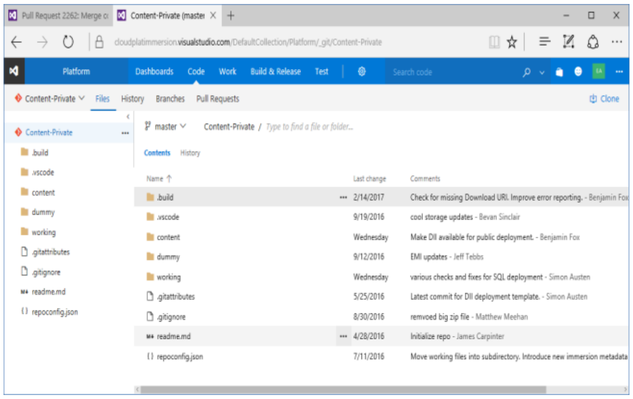


Sign in

Use your Microsoft account.

What's this?

[redacted]@live.com



cloudplatimmersion.visualstudio.com/DefaultCollection/Platform/_git/Content-Private

Content-Private / Type to find a file or folder...

Name ↑	Last change	Comments
.build	2/14/2017	Check for missing Download URI. Improve error reporting. - Benjamin Fox
.vscode	9/19/2016	cool storage updates - Bevan Sinclair
content	Wednesday	Make Dll available for public deployment. - Benjamin Fox
dummy	9/12/2016	EMI updates - Jeff Tebbs
working	Wednesday	various checks and fixes for SQL deployment - Simon Austen
.gitattributes	5/25/2016	Latest commit for Dll deployment template. - Simon Austen
.gitignore	8/30/2016	remvoed big zip file - Matthew Meehan
readme.md	4/28/2016	Initialize repo - James Carpinter
repoconfig.json	7/11/2016	Move working files into subdirectory. Introduce new immersion metadata

Getting Started on your Local Machine

Git is a version control system (VCS) that allows the tracking of changes to a file or set of files over time. The branch concept represents a contained area of work and tracks changes to a specific file or set of files. The commit is how you indicate when a version should be preserved. Git uses a local version control methodology, meaning changes occur on your local machine and must be synchronized periodically with the server.

Within the repository, there are two important branches:

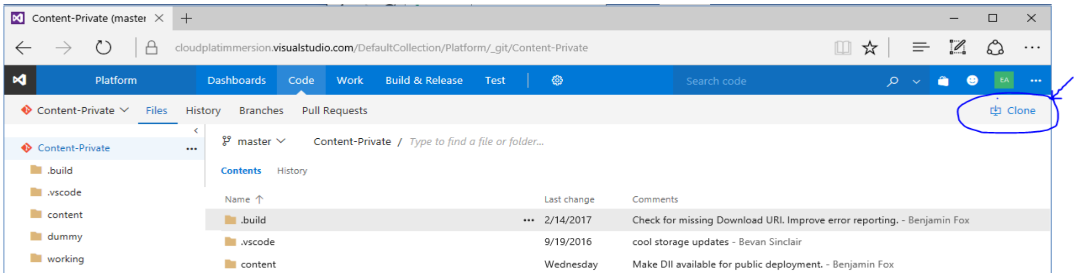
- `master` - Tracks content that has gone through deployment testing and is approved for deployment in the Production environment. This content is generally stable.

- testenvironment - Tracks content deployed to the Test environment. Continuous integration will move work in progress into the testenvironment branch when you create a pull request against the Master branch.

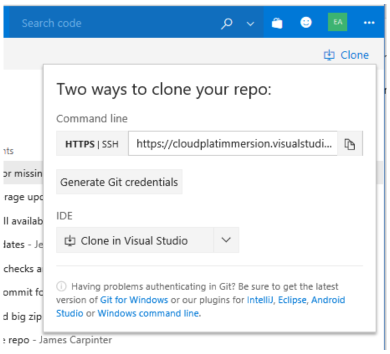
These branches are commit gated - all commits must go through a Pull Request, where they must pass the Content Build process and then pass review and approval by a Valorem platform team member.

All content development occurs in a feature branch, or “working” branch. Working branches should be prefixed with a vendor identifier and have a unique, descriptive name. You must create a local copy of the Git repository before you can begin work; this is referred to as cloning. The following steps will walk you through cloning the repository and creating your working branch.

1. Clone the repository to your local machine from the VSO UI.
   a. From Visual Studio Online (VSO), click the “Clone” button in the upper right hand corner to open a dialog with the URL to the repository.
   b. From the dialog, you have two options: copy the URL provided and paste it into the Integrated Development Environment (IDE) you will be using to create your content, or use the drop down list to select your IDE and follow the prompts.



Name ↑	Last change	Comments
.build	2/14/2017	Check for missing Download URI. Improve error reporting. - Benjamin Fox
.vscode	9/19/2016	cool storage updates - Bevan Sinclair
content	Wednesday	Make Dll available for public deployment. - Benjamin Fox

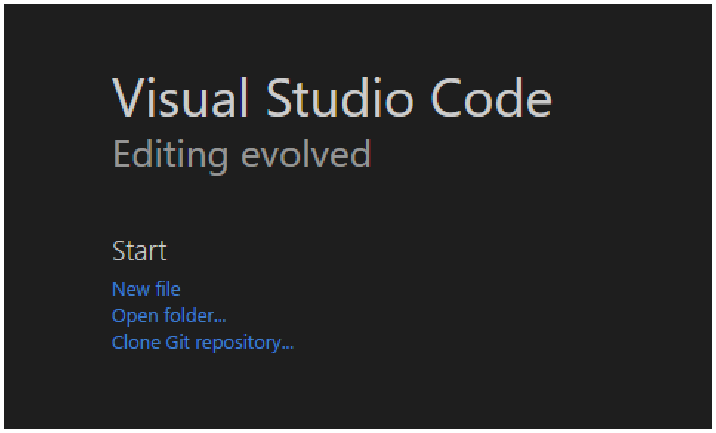


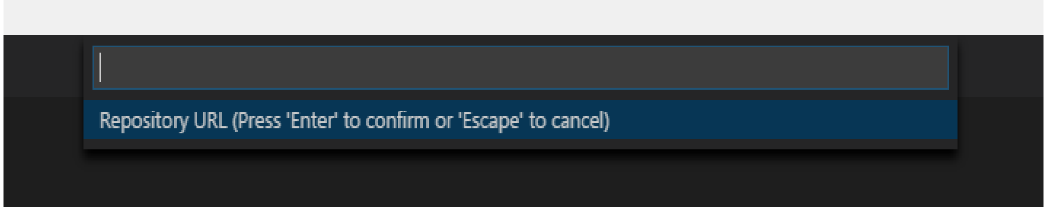
Two ways to clone your repo:

https://cloudplatimmersion.visualstudi...

Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

c. Or, from the Welcome tab in Visual Studio Code, select the “Clone Git repository…” option. Then, paste the URL to the repository into the text box that appears at the top of the window. You can copy the URL from the VSO dialog as described above. Press ‘Enter’ to complete the cloning process.



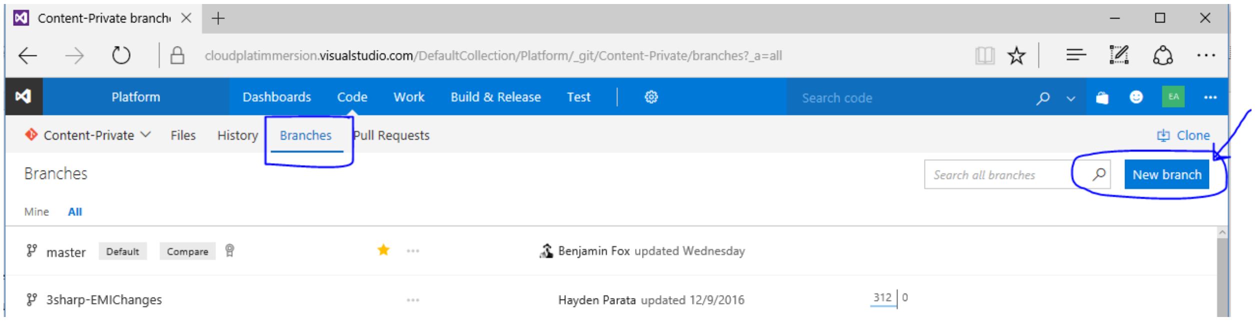

Repository URL (Press 'Enter' to confirm or 'Escape' to cancel)

d. Or from the command-line, create a clone of the repository.

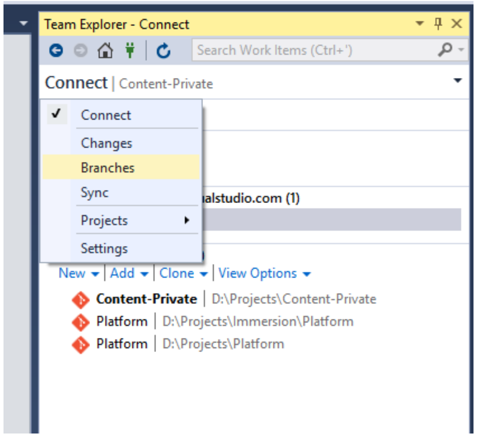
```
git clone https://cloudplatimmersion.visualstudio.com/DefaultCollection
/Platform/_git/Content-Private
```

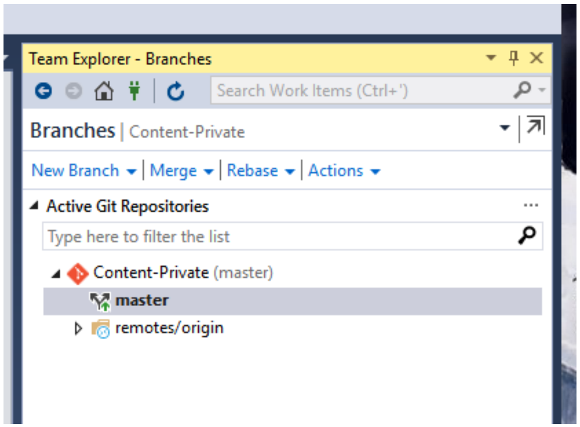
2. Create and check out a new branch based on the Master branch.

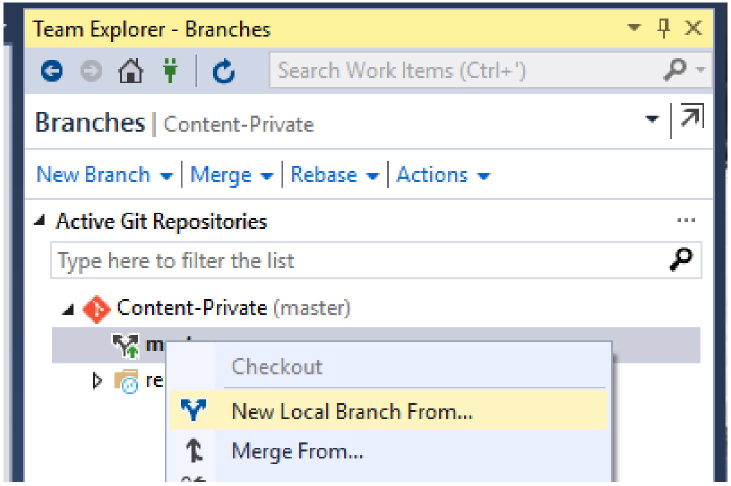
a. From VSO, navigate to the Branch tab. Then, click the “New Branch” button in the upper right hand corner. Then, use your preferred Git tool to check out the branch.



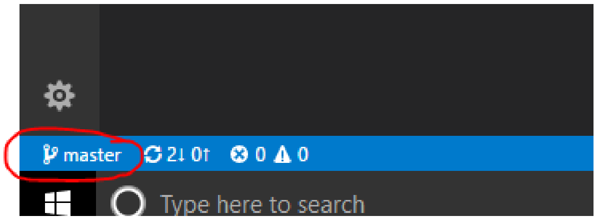
b. Or, from Visual Studio, navigate to the Team Explorer > Branches view. Right-click the Master branch, listed in bold in the following image, and select the “New Local Branch From…” option in the drop menu.

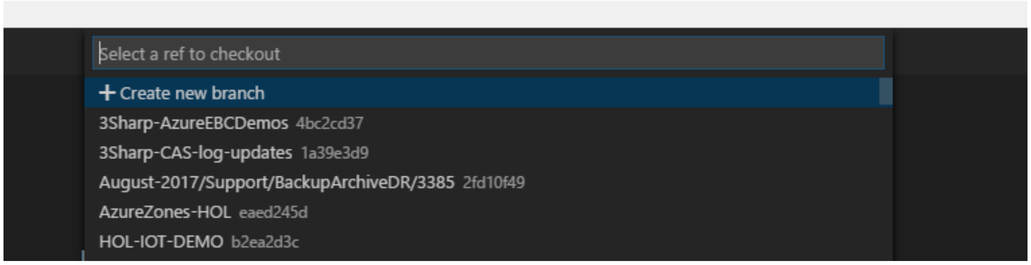

Team Explorer - Connect
Search Work Items (Ctrl+')
Connect | Content-Private
✓ Connect
Changes
Branches
Sync
Projects
Settings
alstudio.com (1)
New | Add | Clone | View Options
Content-Private | D:\Projects\Content-Private
Platform | D:\Projects\Immersion\Platform
Platform | D:\Projects\Platform


Team Explorer - Branches
Search Work Items (Ctrl+')
Branches | Content-Private
New Branch | Merge | Rebase | Actions
Active Git Repositories
Type here to filter the list
Content-Private (master)
master
remotes/origin


Team Explorer - Branches
Search Work Items (Ctrl+')
Branches | Content-Private
New Branch | Merge | Rebase | Actions
Active Git Repositories
Type here to filter the list
Content-Private (master)
Checkout
New Local Branch From...
Merge From...

c. Or, from Visual Studio Code, click the branch name in the lower left-hand corner to open the Branch selection drop list at the top of the window. Use the Create new branch option to create your working branch.

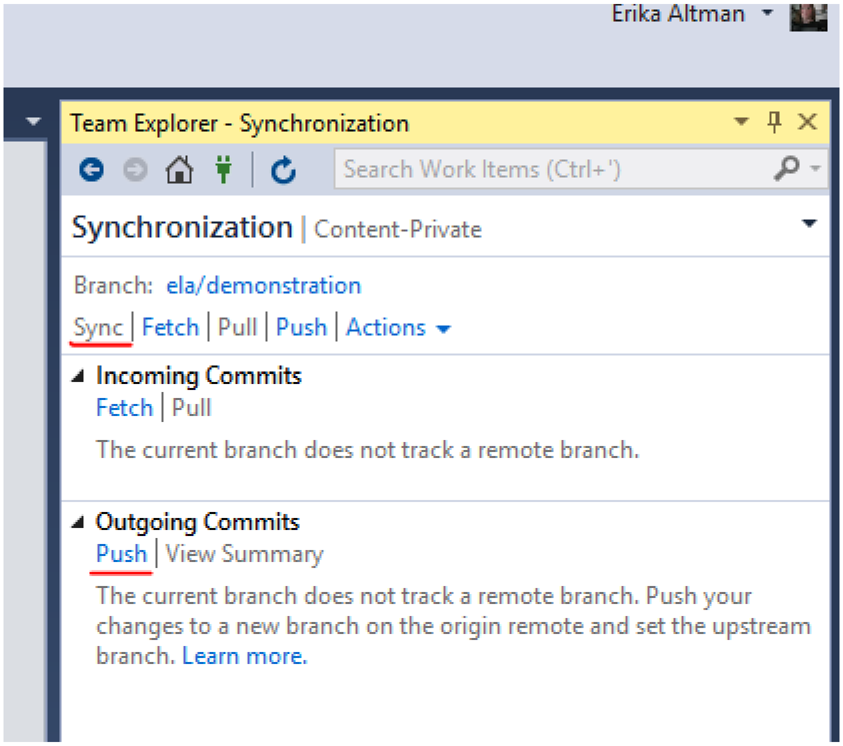



Select a ref to checkout
+ Create new branch
3Sharp-AzureEBCDemos 4bc2cd37
3Sharp-CAS-log-updates 1a39e3d9
August-2017/Support/BackupArchiveDR/3385 2fd10f49
AzureZones-HOL eaed245d
HOL-IOT-DEMO b2ea2d3c

d. Alternatively, from the command line create a new local branch and switch to it.

```
cd 'path\to\Content-Private'
git checkout master
# update local master to latest from the server
git pull
# create a new branch and check it out
git checkout -b 'branch-name'
```

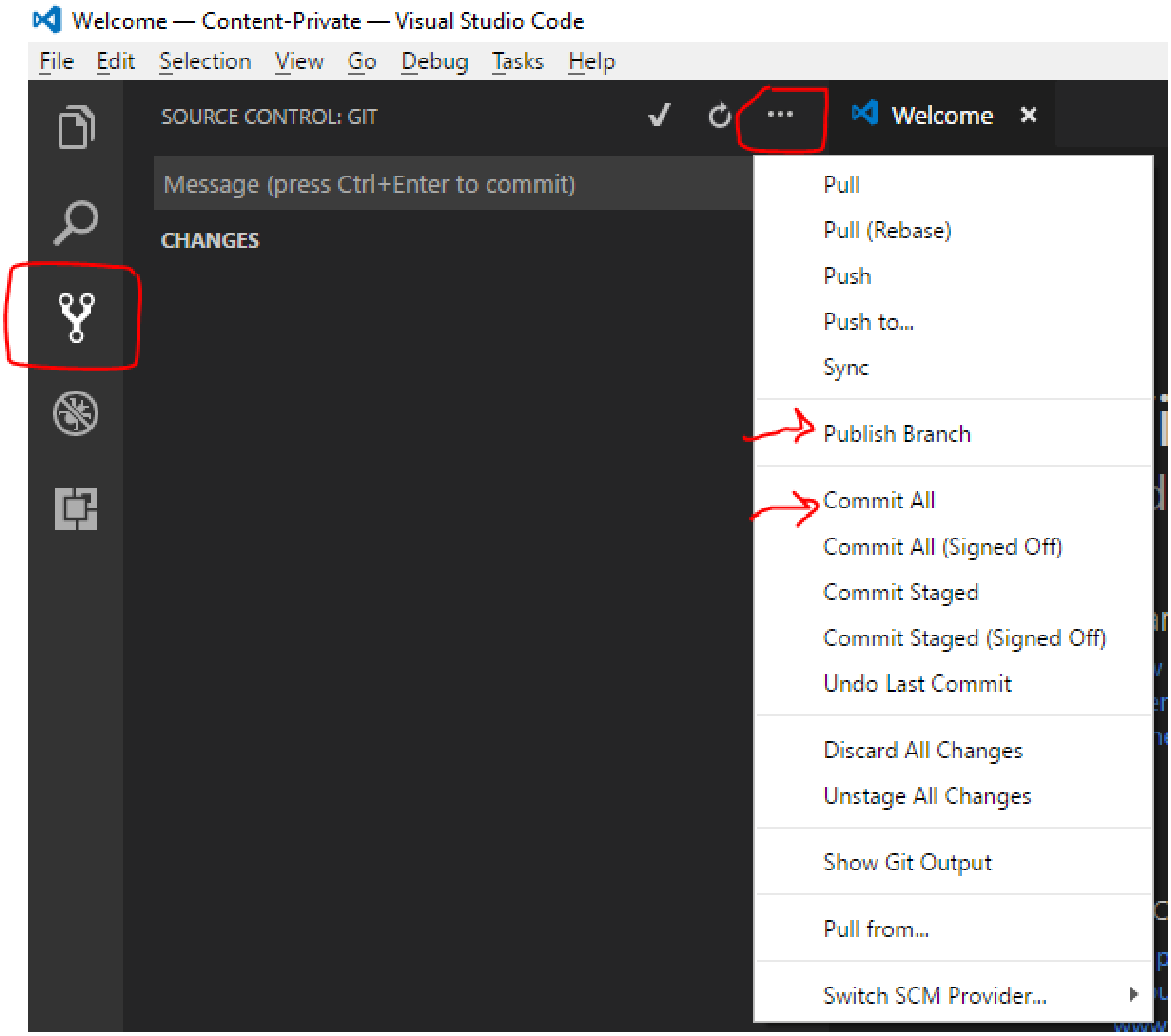
3. Once you have checked-out the new branch, you can begin work. Open the `$/Content-Private/working` folder and create a new folder to contain your lab(s). Use your initials and/or company name as a prefix to identify the folder as yours. This allows you to create, test, and update your content before submitting it to the main `content` location.
    a. Valorem will provide a `content` destination folder name for new labs with your welcome message.
    b. You can use the Content Processor application to verify that your content package will build before you submit it.
    c. You can use a Markdown preview tool, such as Visual Studio Code or Visual Studio Community, to review your formatted instructions.
    d. When you have completed constructing your content and verified your lab is working as expected locally, move the content package from `working` to the assigned `content` location.
4. Make your changes, then commit and push your new branch to the server. Committing the changes to the server will share your local branch with the rest of the repository, making it available for others to review and potentially perform work in.
    a. From Visual Studio Team Explorer, navigate to the Synchronization view. Under the 'Outgoing Commits' section use the 'Push' option to publish your local branch to the server, then use the 'Sync' command to push your local changes to the server.


Erika Altman
Team Explorer - Synchronization
Search Work Items (Ctrl+')
Synchronization | Content-Private
Branch: ela/demonstration
Sync | Fetch | Pull | Push | Actions
Incoming Commits
Fetch | Pull
Outgoing Commits
Push | View Summary
Learn more.

The current branch does not track a remote branch.

The current branch does not track a remote branch. Push your changes to a new branch on the origin remote and set the upstream branch.

b. From VS Code, select the Git tab from the left hand navigation. Then, click the ellipsis at the top of the explorer window to open the Git menu. Use the ‘Publish Branch’ option to push your local branch to the server, then use the ‘Commit All’ option to push your content changes to the server.



c. From the command line:

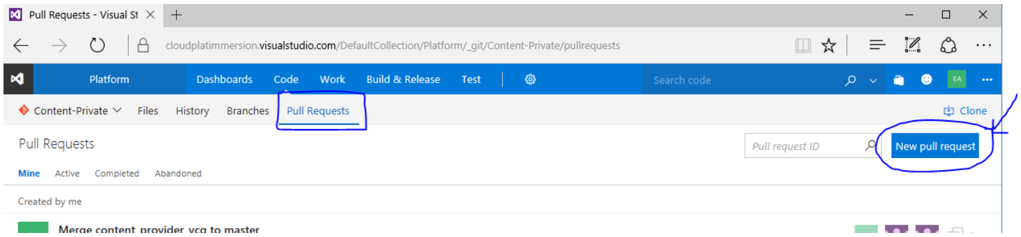
```
git add 'subdir\with\my\changes'
git commit -m 'Description of changes'
git push origin 'branch-name'
```

Requesting Review and Approval

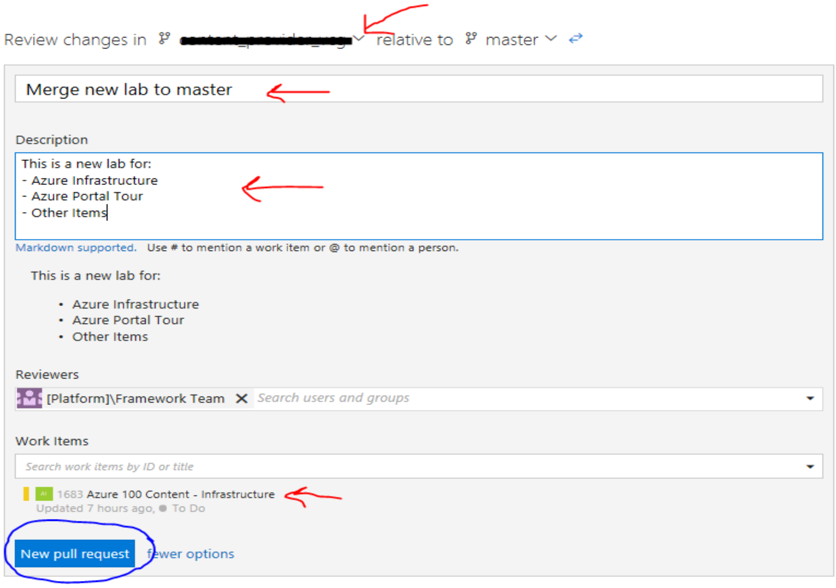
Content changes are reviewed using the Git Pull Request feature. After publishing your feature branch to the server, you will create a pull request in VSO and notify holcontent@valorem.com that your package is ready for review. If you need to complete additional testing or user acceptance testing (UAT), that can be done from the Test environment.

Creating a Pull Request

1. Create a pull request from your branch to the Master branch in VSO in the [Hands-on Labs Content-Private Pull Requests](#) screen.
   a. From VSO, select the Pull Requests tab.
      i. Then, click the “New Pull Request” button in the upper right hand corner.



      ii. Verify that the branch specified in the pull request is correct, and that the direction of the request is correct. You should see a title similar to, “Review changes in <branch_name> relative to master” where <branch_name> is the branch you created in step 4.
      iii. Enter a descriptive title and comment to inform the Valorem reviewer about the changes.
      iv. Enter the work item number in the “Work Items” search box to attach the pull request to the ticket.
      v. Click the “New pull request” button at the bottom of the form to submit.



Review changes in [redacted] relative to master

Merge new lab to master

Description

This is a new lab for:
- Azure Infrastructure
- Azure Portal Tour
- Other Items

Markdown supported. Use # to mention a work item or @ to mention a person.

This is a new lab for:

- Azure Infrastructure
- Azure Portal Tour
- Other Items

Reviewers

[Platform]\Framework Team

Work Items

1683 Azure 100 Content - Infrastructure
Updated 7 hours ago, To Do

   b. Send an email to holcontent@valorem.com to indicate that your content is ready for review.

Valorem Review

2. The pull request will be built on the server, and if the content build is successful, the continuous integration workflow will deploy the package into the Test environment.
3. The pull request is updated with comments from the continuous integration workflow when content is deployed and available in the Test environment.

4. Additionally, the Valorem platform team will review the pull request, and may leave comments as well.
    a. If the reviewer requests changes, you can return to the branch on your local machine, make the fixes and push your new changes back to the server as described above. The Pull Request will automatically update to reflect your changes.
    b. Content providers are responsible for correcting any issues discovered within the content package prior to deployment.
    c. The Pull Request will be approved and completed only after it has sign-off from:
        i. The Ops team – testing the content deployment
        ii. The content author – testing the content itself

Providing Payload Content

Payload content generally refers to those items that are too large to add to the Content repository. These would be VHD images, large datasets for populating example data stores, application installation files that will be run on the jumphost during provisioning, etc. These items are required to be hosted in the Platform Storage service so the content is load balanced and easily available within the end-user's region. Currently, the process to provide payload files is manual.

1. Upload your payload files to an Azure blob storage account, or other remotely accessible location.
2. Send an email message to holcontent@valorem.com with the URL that will provide access to the payload, and a list of the files to pull.
3. Valorem will upload the payload files to the Storage Service, and provide back a blob resource URI in the format: `blob:{line}/{uploadDate}/{payloadFileName}`; where {line} is the line of content you are developing, {uploadDate} is the date the file was uploaded to the Storage Service, and {payloadFileName} is the name of the file. This resource URI should be used in the ARM template parameters file to refer to the payload file, as described in the Getting Started Guide section Story Deployment – Parameters Files.