

Operations deployment doctrine

Doctrines

When fixing an issue for security-related patches in an open source project we follow these procedures:

1. **Create a new issue in DSEC project, describing the issue with all the required details.**
2. **Create a ticket in your application's project(e.g. CMR, MMT, EDSC). The title should be "Fix the issue in DSEC-X" where "X" is the ticket number generated in step 1.**
 - a. Do not put any details in the project ticket's description or acceptance criteria other than "Fix the issue listed in DSEC-X"
 - b. Link the ticket to the DSEC ticket with a status of "resolves", e.g. MMT-1282 resolves DSEC-14 (These numbers are made up, it's just an example)
3. **Contact your friendly security team member for severity and priority for work.**
4. **Coordinate with your Dev Team Lead and Scrum Master to discuss priority.**
5. **Once approved for work, reference the project ticket and not the DSEC ticket.**
 - a. Do not reference 'security' or any other terms in your repository fix. This is because it will be published to the public repository before it has actually been applied to the environments.
6. **Backport to any required branches and prepare releases.**
7. **Upon completion of the work contact your local friendly Product Owner/OPS Designated person for deployment.**

When deploying a release we apply the following doctrines:

1. **We expect the development group/product owner to create the release**
2. **We will only deploy a release to an environment if that release has successfully been applied to the prior environment.**
 - PROD release will only occur if it has successfully applied to UAT
 - UAT release will only occur if it has successfully applied to SIT
3. **When selecting a minor release for an environment that release must match the current minor release of the prior environment.**
 - If 1.18.4 is in UAT and we wish to promote 1.18 to PROD then we must apply 1.18.4 to PROD
 - If 1.19.2 is in SIT and we wish to promote 1.19 to UAT then we must apply 1.19.2 to UAT
4. **When applying a minor release to SIT we must apply that same minor release to UAT once SIT has been verified.**
 - If we apply 1.19.2 to SIT successfully we must then apply 1.19.2 to UAT as soon as possible.
5. All of the tickets associated with the release have a FixVersion attached to them, and is reviewed before deployment.
6. All issues should be in the "Verified Internal" or higher state before deployment to PROD, excepting those tickets that can ONLY be applied to PROD. A note should be included in the ticket.

A 'successful' release is defined as follows,

1. The deployment is successful
2. Sanity check is successful
3. If the release has an issue that can be tested in the environment it must be tested successfully.

In the event a change needs to be reverted:

1. Create a new issue for the reversion and give it a FixVersion of a +1 minor version. (e.g. reverting a change in 1.102.1, the reversion should be fixVersion 1.102.2)
2. Revert the release, and reference the new ticket # in the commit.
3. Follow normal release processes with the reverted code.

Exceptions

1. Emergency releases
 - In rare cases, we will have a situation where a problem in PROD will cause severe service degradation. In those cases, with the agreement of the product owner, a release can be applied directly to PROD.
2. PROD/UAT fixes
 - PROD is always one major release behind UAT. If a fix is applied to 1.19 in the form of 1.19.2 then that would be applied to SIT and UAT. But PROD is on 1.18.4 so a new release would be created for PROD (1.18.5) and deployed directly without testing in SIT and UAT

Minor Release process

This process is triggered every time a development sprint ends.

Step no.	Step description	Further documentation	Responsibility
1	Complete sprint	<ol style="list-style-type: none">1. Navigate to Agile board in JIRA2. Click on 'Active Sprint'3. Click on 'Complete Sprint'	DEV

2	Produce release notes page	In order to ensure that deployment occurs correctly, any pre-requisites or special instructions for a release need to be captured. An example is here - CMR Version 1.60 Notable changes is important for our DSR presentations so provide it.	DEV
3	Create minor release	<ol style="list-style-type: none"> 1. <a href="https://ci.earthdata.nasa.gov/browse/<your project>">https://ci.earthdata.nasa.gov/browse/<your project> 2. Click on last build associated with the above sprint 3. Click on 'Create Release' button 4. Create minor release. For example, '1.01.0' 	DEV
4	Create patch release branch in git	<p>Earthdata Bit Bucket</p> <ol style="list-style-type: none"> 1. <a href="https://git.earthdata.nasa.gov/projects/<your project>">https://git.earthdata.nasa.gov/projects/<your project> 2. Click on repository for project 3. Select 'master' branch 4. Click on '...' 5. Select 'create branch from here' 6. Select 'custom' branch type 7. Create patch release branch. For example, if minor release is '1.01.0' then patch release is '1.01.x' <p>NASA Github</p> <ol style="list-style-type: none"> 1. <a href="https://github.com/nasa/<your project>">https://github.com/nasa/<your project> 2. Click on 'Branch: Master' 3. Type your release branch name into text input field. For example, if minor release is '1.01.0' then patch release is '1.01.x' 4. Hit return 	DEV
5	Deploy to SIT - unless continuous build in place	On the Monday after the Friday sprint closure, Deploy minor release to SIT environment via Bamboo	OPS
6	Create verification Kanban board	<ol style="list-style-type: none"> 1. Navigate to JIRA (https://bugs.earthdata.nasa.gov) 2. On Projects pull-down, select Common Metadata Repository 3. On the left sidebar, there is a pull-down at the top under the name of the project; select 'CMR' from that pull-down. 4. On the sidebar, Click on 'Reports' 5. At the top of the page, on the 'Switch Report' pull-down, click on 'Sprint Report' (if you don't see the Sprint Report there, click on the Boards pull-down) 6. On the second horizontal panel on the page, at the top of the panel there is a sprint pull-down. Select the just-completed sprint for CMR Metadata. 7. The sprint report for that sprint will appear. Find the section titled 'Completed Issues', and click on 'View in Issue Navigator' of 'Completed issues' section 8. At the top of the page, a search query appears in a box, listing all of the completed issues in that sprint. Filter out the tasks by adding ' AND type != task' to issue filter. 9. Save issue filter with a name like 'UAT Verifications <start date of verification period> - <end date of verification period>' 10. On a separate tab, repeat steps 1-7 for the Earthdata Search Client project. (In step 2, select Earthdata Search Client; in step 3, select 'Earthdata Search' from the pull-down) 11. At the top of the page, a search query appears in a box, listing all of the completed issues in that sprint. FILTER OUT ISSUES WITH THE 'searchlab' LABEL. Copy and paste the list of issues into the filter you created in step 9. Save the updated filter. 12. In the same tab as step 10, repeat steps 1-7 for the Metadata Management Tool project (In step 2, select Metadata Management Tool; in step 3, select 'Metadata Management Tool' from the pull-down) 13. At the top of the page, a search query appears in a box, listing all of the completed issues in that sprint. Copy and paste the list of issues into the filter you created in step 9. Save the updated filter. 14. To the right of the filter name at the top of the page, click on the Details link, and in the dropdown, click on Edit Permissions. Edit the filter, and click on +Add to add Everyone to the Share list. 15. Go to the Boards pull-down in the JIRA banner, and click on View All Boards. On that page, at the top right, click on 'Create Board'. 16. On the Create Board pop-up, choose 'Create a Kanban board', and on the next pop-up, choose 'Board from an Existing Filter' 17. On the next pop-up, select the name of your saved filter in the Saved Filter drop-down, and name the Kanban board the same as the filter name. Click on 'Create Board'. 18. Click on the Board drop-down on the upper right of page, choose Configure, then in the left panel, choose Columns. Add verified internal and verified external columns to the Kanban board (using the Add Column link on the page). Move issue status 'types' (e.g., Ready for Test, Verified Internal) into the correct columns of the board. Click on Back to Board. 19. Check that all issues are in the appropriate columns on the board, according to their issue status. 20. (if necessary) Click on the Board drop-down on the upper right of page, choose Configure, then in the left panel, choose General. At the bottom of the page, under Kanban board sub-filter, remove the default sub-filter and Update. 21. Assign all issues to be verified 	TEST
7	Deploy to UAT	On the Wednesday after SIT deployment, if SIT deployment was successful, Deploy minor release to UAT environment via Bamboo	OPS
8	Verify release issues	Using sprint demos and issue test instructions, verify all issues in the verification Kanban board. Confirm all issues in the release are in the Verified Internal State or higher.	TEST
9	Deploy to Production	On the second Wednesday after UAT deployment if all issues were verified, Deploy major release to PROD environment via Bamboo	OPS

10	Delete maintenance branch	<p>Once a release has been superceeded in PROD the maintenance branch is no longer required. For, example, when 1.02 is deployed to PROD, the maintenance branch 1.01.x can and should be deleted.</p> <p>Earthdata Bit Bucket</p> <ol style="list-style-type: none"> 1. <a href="https://git.earthdata.nasa.gov/projects/<your project>">https://git.earthdata.nasa.gov/projects/<your project> 2. Click on repository for project 3. Click on 'branches' icon on left side panel 4. Find the branch to delete, e.g. 1.01.x 5. Click on '...' 6. Select delete and confirm <p>NASA Github</p> <ol style="list-style-type: none"> 1. <a href="https://github.com/nasa/<your project>">https://github.com/nasa/<your project> 2. Click on 'branches' 3. Click on trashcan of the maintenance branch 	
----	---------------------------	---	--

Patch Release process

This process is triggered every time an issue is detected that needs to be pushed to an environment outside of the sprint process.

For example,

- Showstopper issue detected
- Failed verification in UAT

Step no.	Step description	Further documentation	Responsibility
1	Create git feature branch on master	git checkout -b <JIRA issue number>	DEV
2	Complete fix	Ensure all tests pass	DEV
3	Push to master	git push -u origin HEAD	DEV
4	Cherry pick from master to patch release branch	<ol style="list-style-type: none"> 1. git fetch 2. git branch -r 3. git checkout <minor release branch> eg 1.01.x 4. git pull --rebase 5. git log master 6. Copy the hex code of the commit you made to master in step 3 7. git cherry-pick <hex code> Note: if there were multiple commits do the oldest one first etc. 8. git push 	DEV
5	Create patch release	<ol style="list-style-type: none"> 1. https://ci.earthdata.nasa.gov/browse/ project> 2. Click on last build associated with the minor branch 3. Click on 'Create Release' button 4. Create patch release. For example, '1.01.1' 	DEV
6	Create tag in repository	<ol style="list-style-type: none"> 1. git tag -a <patch release number> -m "Tagging <patch release number> release" 2. CMR: git push upstream <patch release number> 3. CMR OS and CSW: git push origin <patch release number> 	DEV
7	Create patch release branch in bamboo	<ol style="list-style-type: none"> 1. <a href="https://ci.earthdata.nasa.gov/browse/<your project>">https://ci.earthdata.nasa.gov/browse/<your project> 2. Click on 'Actions'->'Configure Plan' 3. Click on 'Branches' tab 4. Click on 'Create branch plan' 5. Select patch release branch from step 3 of minor release steps 6. Click on 'save' 	OPS
8	Deploy to appropriate environments	Deploy minor release to SIT UAT PROD environment via Bamboo	
9	Verify patch release		OPS