.NET Core Engineering Services Operations info: - {Process Name}

Process Details

Summary:

Why does this "process" exist in the first place? Who benefits? What happens if it doesn't happen?

Process Boundaries

What is, and isn't part of this process? Provide data that provides boundaries both in terms of code base and time. As this is a template, feel free to remove / add as applicable. If you find yourself adding content that should be global, please make a pull request to this template doc.

- Related repositories: Links to relevant repos where work will be done
- Task scope: List guidance/examples of in-scope and out-of-scope for the task
- Contacts for non-owned parts of the process: For external ownership, who can we talk to?

Process Inputs / Outputs

Descriptions of what/where the inputs to the process come from (the answer to "what do I or the automated process neeed to consider to perform this task?", and what performing the below steps correctly achieves ("what comes out the other side?")

Examples:

Inputs: - Base Docker images from DockerHub.io / mcr.microsoft.com - Gallery Images available from the Azure Portal - Package versions from public / internal NuPkg feeds - State of the objects in an Azure Subscription

Outputs: - Updated dependencies / images - Changed state of - Assorted reports or telemetry used in reporting

Execution Steps

These steps will vary based off the type of operation involved. As your process may use any combination of automated / manual stages, use what you need from the template and delete

Fully-automatable routines:

- Description of the process
- Links to:

    - Source code, any other (say, in-repo) documentation for the automation
    - Pipelines associated with build / deployment of relevant components.
    - Telemetry pages / Grafana alerts related to this process
- Known issues impacting the area
- Known tech debt that may cause validation "blindness"

Manual processes:

- Step-by-step description of the process in sequential markdown list format.
- Known issues impacting the area
- Known tech debt that may cause validation "blindness"
- Troubleshooting guide per-step, ideally tested by execution by an individual unfamiliar with the feature area(s) involved.

Troubleshooting: List of what to do when "known" things go wrong. When a new problem occurs and requires investigation and fixing, it should be added here.

Validation Steps

After completing manual steps, or on some regular cadence (to be determined), list any follow up checks/activities that need to be done, including things like:

- Which build(s) need to be in a green state
- Sites to check
- "Smoke testing" steps for functionality known to lack automation/ have historical regressions

Checklist for reviewing this document This part is more for guidance but can be retained in documents deriving from the template; it gives the writer a means to try to warn against any recurring problems seen.

- Have the supplied steps been executed by a non-SME, non-author IC?
- Do any references to resources include how to obtain and which security permissions are required (if any)
- Are links pointing to other documents or locations valid?
    - Will they be readable by the target audience? If restricted, do they tell the reader where to gain access?
    - Will they continue to exist in the future? (some links, like non-retained AzDO builds, are impermanent)
- Is/are there at least one (ideally two) SME IC(s) listed as contact for clarification?
- Does the document specify sufficient detail that an artibrary reader would be able to reason about and execute the processes described?

Was this helpful? 👍 👎