Iteration 1

Code Sheriff responsibilities:

- handle downstreams
    - only simple, obvious cases
- retrigger/backfill
- hand off alert summaries to Perf Sheriff
    - leave useful notes to Perf Sheriff
- extra coordination with Perf Sheriff & other Code Sheriffs, via Perfherder

Handle downstreams [1]
It is the first thing a CS* does, when a new alert summary lands.

CS only marks obvious alert summary as downstream.
A wiki description similar to this one should provide multiple detailed examples.

If CS is unsure, she retriggers/backfills [2], with the distinction that she retriggers 2 times per cherry-picked alert changesets. She does this for both branches**.
Then CS hands off [3] the alert summaries to PS*** -- with notes mentioning her uncertainty.

Specific edge cases CS is instructed to look for:
(1) There is a downstream and there is no alert on the other branch.
CS leaves note mentioning this. She then performs the same steps from (**).

(2) There looks to be a downstream but really isn't -- say we are missing data on the other branch.
CS leaves note mentioning this, then performs the same steps from (**).

Retrigger/backfill [2]
When a new alert summary lands, a CS must be able to cherry-pick the minimum amount of alerts, out of which a conclusion can be drawn. This selection must be visible, for others to see. A maximum of 5 alerts can be cherry-picked, per alert summary.

For each of the cherry-picked alerts, the CS retriggers 5 changesets, 4 times each: the one that caused the alert + the previous/succeeding 2 changesets.

For this 1st iteration, CSs will only work with Talos and AWSY alerts. This also applies to all steps from [1] to [4].

IMPORTANT: A CS can only retrigger a **max** of **100 performance jobs** per **platform** per **day** per **shift**. Reason for this is that perf jobs run on a small pool of hardware.

By all means, the CS will try to not go beyond this limit, as we consider it more than sufficient even for alert intensive days. This limit includes the Handle downstreams [1] step also.

For each of the cherry-picked alerts, the CS then backfills any missing changeset data between these 5 changesets.

Hand off alert summaries to Perf Sheriff [3]
CS *immediately* marks an alert summary as "confirming", after she retriggered/backfilled [2] it as stated above. Alert summaries *in this state* are no longer her concern.

CS is encouraged to leave notes for each alert summary, for whatever reason: i.e. beginners can leave questions, asking the PS to review their cherry-picked selection, CS can leave warnings about the noise levels etc.

Extra coordination with Perf Sheriff & other Code Sheriffs, via Perfherder [4]
This happens when retriggers/backfills from round 1 aren't enough for the PS to conclude an alert summary handed off [3] by CS, *during* the PS's shift.

Perf Sheriff responsibilities:

- same as before; cover all Code Sheriff responsibilities from Stage 1
- extra coordination with other Code Sheriffs, via Perfherder
- document all suggestions/complains left in the notes & work on fixing docs/tools
- leave useful notes for future iterations
- check system load

Same as before; cover all Code Sheriff responsibilities from Stage 1 [5]

Nothing more to add here.

Extra coordination with other Code Sheriffs, via Perfherder [6]

Same story from [4].

Document all suggestions/complains left in the notes & work on fixing docs/tools [7]

PSs keep a Google spreadsheet record, for all workflow cases -- ideally they can summarize with counts. It tracks when we have to do more retriggering after the first round and logs what/why was required i.e. specific tests/configs, edge cases like broken builds/merges, too much noise etc.

Examples of workflow situations which must be recorded:

- confirming wasn't enough -- put into buckets (i.e. broken build, broken tests, infra problems, pending queue, need bisection, more revisions needed, more retriggers needed)
- obvious downstream is missing alerts
- downstream alert that isn't obvious -- put into buckets (i.e. noisy test, needed more data, bimodal etc.)
- review monthly to find patterns -- ideally creating actionable bugs for tools to make "data ready" easier to hit

Leave useful notes for future iterations [8]

PSs leave whatever notes they consider useful inside alert summaries i.e. complains about noise levels on a specific alert/platform, backfilling not returning results etc.

PS will then get usefulness out of them, according to [7].

Check system load [9]

PSs must ensure CSs don't overload the Taskcluster system from doing too many retriggers.

PSs achieve this by multiple means:

- check twice a day the performance graphs from Perfherder
- check twice a day the load via this Taskcluster tool
- graph the data over time and communicate via #sheriffs if we are too backlogged

Implementation details:

- confirming -- new Perfherder state for alert summary
- notes for alert summary
    - allows communication + state + protocols between sheriffs
- cherry-pick alerts
    - * Sheriff selects an alert subset from an alert summary
    - only cherry-picked alerts get retriggered/backfilled
    - leftover alerts are considered redundant & should be ignored
    - requires knowledge/training about alert grouping
- wiki docs with edge cases i.e (**)
- actual team training
    - 30 days for initial training, starting from April 19th
    - 30 more days to ensure all is going well

Iteration 2

To be established around June, assuming Iteration 1 goes smoothly. Perfherder & other tools will be adjusted in this timeframe.

Team will be instructed in July.

Enhancements

These can be added outside of any iteration above.

- make it easy to get a view of +-2 revisions for retriggering from a perfherder graph or alert
- detect if we have enough data and show status in the alert
- for a given alert summary recommend jobs to retrigger/backfill- i.e. if damp regresses on all platforms choose one (possibly based on noise and confidence/severity)
- auto detect downstream and offer a recommendation in the alert view (only for simple cases)
- suggest similar regressions across alert summaries that might have the same root cause (same test but revisions within small range +/-2 revisions)

* CS means Code Sheriff
*** PS means Perf Sheriff
Red means TODO