GitHub & grafanabot automation

The bot is configured via commands.json and some other GitHub workflows workflows.

Comment commands:

- Write the word `/duplicate #<number>` anywhere in a comment and the bot will add the correct label and standard message.
- Write the word `/needsMoreInfo` anywhere in a comment and the bot will add the correct label and standard message.

Label commands:

- Add label `bot/question` the the bot will close with standard question message and add label `type/question`
- Add label `bot/duplicate` the the bot will close with standard duplicate message and add label `type/duplicate`
- Add label `bot/needs more info` for bot to request more info (or use comment command mentioned above)
- Add label `bot/close feature request` for bot to close a feature request with standard message and adds label `not implemented`
- Add label `bot/no new info` for bot to close an issue where we asked for more info but has not received any updates in at least 14 days.

Metrics

Metrics are configured in metrics-collector.json and are also defined in the metrics-collector GitHub action.

Backport PR

To automatically backport a PR to a release branch like v7.3.x add a label named `backport v7.3.x`. The label name should follow the pattern `backport <branch-name>`. Once merged grafanabot will automatically try to cherry-pick the PR merge commit into that branch and open a PR. It will sync the milestone with the source PR so make sure the source PR also is assigned the milestone for the patch release. If the PR is already merged you can still add this label and trigger the backport automation.

If there are merge conflicts the bot will write a comment on the source PR saying the cherry-pick failed. In this case you have to do the cherry pick and backport PR manually.

The backport logic is written here