`cargo lintcheck`

Runs clippy on a fixed set of crates read from `lintcheck/lintcheck_crates.toml` and saves logs of the lint warnings into the repo. We can then check the diff and spot new or disappearing warnings.

From the repo root, run:

```
cargo run --target-dir lintcheck/target --manifest-path lintcheck/Cargo.toml
```

or

```
cargo lintcheck
```

By default the logs will be saved into `lintcheck-logs/lintcheck_crates_logs.txt`.

You can set a custom sources.toml by adding `--crates-toml custom.toml` or using `LINTCHECK_TOML="custom.toml"` where `custom.toml` must be a relative path from the repo root.

The results will then be saved to `lintcheck-logs/custom_logs.toml`.

Configuring the Crate Sources

The sources to check are saved in a `toml` file. There are three types of sources.

1. Crates-io Source

   ```
   bitflags = {name = "bitflags", versions = ['1.2.1']}
   ```

   Requires a "name" and one or multiple "versions" to be checked.

2. `git` Source

   ```
   puffin = {name = "puffin", git_url =
   "https://github.com/EmbarkStudios/puffin", git_hash = "02dd4a3"}
   ```

   Requires a name, the url to the repo and unique identifier of a commit, branch or tag which is checked out before linting. There is no way to always check `HEAD` because that would lead to changing lint-results as the repo would get updated. If `git_url` or `git_hash` is missing, an error will be thrown.

3. Local Dependency

   ```
   clippy = {name = "clippy", path = "/home/user/clippy"}
   ```

   For when you want to add a repository that is not published yet.

Command Line Options (optional)

```
bitflags = {name = "bitflags", versions = ['1.2.1'], options = ['-
Wclippy::pedantic', '-Wclippy::cargo']}
```

It is possible to specify command line options for each crate. This makes it possible to only check a crate for certain lint groups. If no options are specified, the lint groups `clippy::all`, `clippy::pedantic`, and `clippy::cargo` are checked. If an empty array is specified only `clippy::all` is checked.

Note: `-Wclippy::all` is always enabled by default, unless `-Aclippy::all` is explicitly specified in the options.

Fix mode

You can run `./lintcheck/target/debug/lintcheck --fix` which will run Clippy with `--fix` and print a warning if Clippys suggestions fail to apply (if the resulting code does not build).
This lets us spot bad suggestions or false positives automatically in some cases.

Please note that the target dir should be cleaned afterwards since clippy will modify the downloaded sources which can lead to unexpected results when running lintcheck again afterwards.