# Introduction

Thank you for participating in our study. The purpose of this study is to evaluate the performance of popular Golang linters. Please complete this questionnaire within the next 3 weeks. If you have any questions or need more time, feel free to contact us at [wjwcis@udel.edu](mailto:wjwcis@udel.edu).

Please answer the following questions independently without comparing or referring to other questions; or asking for help from other participants/online forums/LLM-based chatbots.

* *I agree*
* *I disagree*

# Survey Questions

## Question 1

<<https://github.com/pulumi/pulumi/pull/11857/commits/6da69abdeabaffcb5c6307c41fb49829cd0d6b81>> fixes a <Misuse of type conversion and declaration> at <line 157-163> in <pkg/codegen/testing/tstypes/tstypes.go>.

For each tool:

This is the output of <tool 1> on the code prior to the commit:

<[pulumi-09aee82c2c72b6dc4357dd7237b4279a71e485b2-tool\_1.txt](https://docs.google.com/document/d/144xI6Nub1yLn_QfIVB2bKOVq-5wFpZYYnxLH00p300k/edit?usp=sharing)>

The output is effective at convincing me that <line 157-163> in <pkg/codegen/testing/tstypes/tstypes.go> is <Misuse of type conversion and declaration>

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| The output succinctly identifies the problem within Go code |
| --- |

The output is effective at guiding me to the fix captured in the commit

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| Output does not provide a valid solution to fix type problem |
| --- |

## Question 2

<<https://github.com/SAP/terraform-provider-btp/pull/39/commits/44be7d1a5d8480d0116ca102bf3b0e67d2de8875>> fixes a <Misuse of type conversion and declaration> at <line 6-22> in <internal/btpcli/actions.go>.

For each tool:

This is the output of <tool 1> on the code prior to the commit:

<[terraform-provider-btp-443356199ef4bfe631c820c7e334bca1e99eb889-tool\_1.txt](https://docs.google.com/document/d/1CnJUNATunVQnRkYLTpYLZfDvPsa17OgRW0ToERMYQ08/edit?usp=sharing)>

The output is effective at convincing me that <line 6-22> in <internal/btpcli/actions.go> is <Misuse of type conversion and declaration>

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| The output correctly identifies the type conversion problem within Go code |
| --- |

The output is effective at guiding me to the fix captured in the commit

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| Although output identifies the type conversion problem, it does not provide a valid solution to fix type problem |
| --- |

## Question 3

<<https://github.com/influxdata/telegraf/pull/11956/commits/a39dc708b9e6afe67792b06214fe58efb26c9a10>> fixes a <Misuse of defer> at <line 314> in <agent/agent.go>.

For each tool:

This is the output of <tool 3> on the code prior to the commit:

<[telegraf-ed8bd1dd525f266f204c921986de40bdf92c4e94-tool\_3.txt](https://docs.google.com/document/d/1kGCSkedN4FqbxX0MX0tJ4fS0yvqZ6MPXSVqyk1IgzOs/edit?usp=sharing)>

The output is effective at convincing me that <line 314> in <agent/agent.go> is <Misuse of defer>

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| The output correctly identifies the using defer problem in loop within Go code |
| --- |

The output is effective at guiding me to the fix captured in the commit

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| Output does not explain how to either move defer out of loop or how to refactor defer |
| --- |

## Question 4

<<https://github.com/influxdata/telegraf/pull/10193/commits/8c7704722bfa01019ab3d3e3aa8a67272c4b015b>> fixes a <Misuse of defer> at <line 183> in <plugins/inputs/rethinkdb/rethinkdb\_server.go>.

For each tool:

This is the output of <tool 3> on the code prior to the commit:

<[telegraf-9bd0c6121e3a94f8a693e32ec06fb6c293e75381-tool\_3.txt](https://docs.google.com/document/d/1YLlnqeP8jfdt5v62UKw05SV0IjGKOiCiI1LMkBE6W5Q/edit?usp=sharing)>

The output is effective at convincing me that <line 183> in <plugins/inputs/rethinkdb/rethinkdb\_server.go> is <Misuse of defer>

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| The output correctly identifies the using defer problem in loop within Go code |
| --- |

The output is effective at guiding me to the fix captured in the commit

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| Output does not explain how to either move defer out of loop or how to refactor defer |
| --- |

## Question 5

<<https://github.com/ActiveState/cli/pull/2943/commits/45f24274148c6062c9bcdb9e6869762303102822>> fixes a <Missing a null check> at <line 18-21> in <internal/runbits/runtime/rationalize.go>.

For each tool:

This is the output of <tool 1> on the code prior to the commit:

<[cli-25b620b5f2b003e36aeba2f59736217732606dba-tool\_1.txt](https://docs.google.com/document/d/1Nqy4-axOfA0jL4ODxrUdYsMaKNlyGHb5Q2hVBJ0MVLg/edit?usp=sharing)>

The output is effective at convincing me that <line 18-21> in <internal/runbits/runtime/rationalize.go> is <Missing a null check>

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| internal/runbits/runtime/rationalize.go is missing a null check at line 21, |
| --- |

The output is effective at guiding me to the fix captured in the commit

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| I can fix it by referring the output, it says where the nil pointer dereference is and I can add null check for it |
| --- |

## Question 6

<<https://github.com/integr8ly/integreatly-operator/pull/2782/commits/bcc3220f04affb7950feba8020d8a6d60994dae2>> fixes a <Misuse of error handling> at <line 34> in <controllers/rhmiconfig/helpers/update\_status.go>.

For each tool:

This is the output of <tool 4> on the code prior to the commit:

<[integreatly-operator-e88a2dfc9d6c992e4ecc8bd4bd604774ec8e41ec-tool\_4.txt](https://docs.google.com/document/d/1GG3WlNncA4P42vIpnIgtUtQC6oYxPbzg8n_N8B80Shw/edit?usp=sharing)>

The output is effective at convincing me that <line 34> in <controllers/rhmiconfig/helpers/update\_status.go> is <Misuse of error handling>

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| This statement might need a null check, not sure |
| --- |

The output is effective at guiding me to the fix captured in the commit

* Strongly Disagree
* 2
* 3
* 4
* Strongly Agree
* No issues at all (N/A)

Why do you believe that the output is effective or not?

| Output points out where the unhandled error statement is, but I can’t find a valid solution to check it, it does not provide that solution, hard to fix |
| --- |

# 