# 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/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 error message itself isn’t clear and informative. |
| --- |

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?

| The first error message point to internal/btpcli/actions.go:6:2 which clearly indicate where the line of error |
| --- |

## Question 2

<<https://github.com/tinylib/msgp/pull/320/commits/79004a02122d12b59ada6f99b4c36f96b53fc629>> fixes a <Misuse of type conversion and declaration> at <line 122-129> in <msgp/elsize.go>.

For each tool:

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

<[msgp-bb8909c937c78279debe87ca18c5cb3332920f38-tool\_1.txt](https://docs.google.com/document/d/1tgCVCEnTuKRwj5dFGQphOeoRc4uF-wMSGz2PWtK5CG0/edit?usp=sharing)>

The output is effective at convincing me that <line 122-129> in <msgp/elsize.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?

| It is complaining about type undefined and/or missing func of interface, which hard to link to error of “Misuse of type conversion and declaration” This is what I got:  msgp/file\_test.go:93:24: data.Msgsize undefined (type Blobs has no field or method Msgsize)  msgp/file\_test.go:95:23: cannot use data (variable of type Blobs) as msgp.MarshalSizer value in argument to msgp.WriteFile: Blobs does not implement msgp.MarshalSizer (missing method MarshalMsg) |
| --- |

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?

| The commit is showing change of declare type but the output link is about missing implementation of interface |
| --- |

## Question 3

<<https://github.com/influxdata/telegraf/pull/11964/commits/654ceecd34e49f5461bda4f9fecc2201ff35642c>> fixes a <Misuse of defer> at <line 213> in <plugins/common/starlark/builtins.go>.

For each tool:

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

<[telegraf-6606c5ed33ac26652251b7afa2c1e6a28074ec0f-tool\_3.txt](https://docs.google.com/document/d/1eDU9wAGgcCRcVKpZ3d-0M_GyMqt4zW829OElA1izbNE/edit?usp=sharing)>

The output is effective at convincing me that <line 213> in <plugins/common/starlark/builtins.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?

| Not the first line but it does mentioned about  plugins/common/starlark/builtins.go:213:5 revive defer: prefer not to defer inside loops Which clearly indicate that the defer within loop has issue |
| --- |

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?

| The output clearly indicate the issue of using defer in loop |
| --- |

## 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 error message about “plugins/inputs/rethinkdb/rethinkdb\_server.go:183:3 revive defer: prefer not to defer inside loops” is hiding far behind the first error message.  The first impression on the output would make think about issue related to return type |
| --- |

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?

| The error message related to misuse of defer is hiding far behind the error message, without knowledge about this issue, it would be hard to make fix as the commit. |
| --- |

## Question 5

<<https://github.com/pub-solar/terraform-provider-hostingde/pull/3/commits/3945d2ca7b38c128592b9cc727eb1949286dbdea>> fixes a <Missing a null check> at <line 29-30> in <hostingde/client.go>.

For each tool:

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

<[terraform-provider-hostingde-ab94adadb16dcbc6c1127ccb34f5cfd80586c8df-tool\_1.txt](https://docs.google.com/document/d/10NPec2eex0B-9sUMCvUZY26UhX7PWff82cyDf2wckV8/edit?usp=sharing)>

The output is effective at convincing me that <line 29-30> in <hostingde/client.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?

| It is clearly indicate that line 29 & 30 has a possible nil pointer exception |
| --- |

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?

| The error message is clear but the change in the commit seems a bit weird.  I don’t see the reason for using references in the first place. For example, if accountId is nil, the the account filed in the struct would end up with “”, which won’t have the reference either. |
| --- |

## Question 6

<<https://github.com/supabase/gotrue/pull/636/commits/579be1dd6cb32160f727a17237f05b04626f2469>> fixes a <Misuse of error handling> at <line 151 & 236> in <api/hooks.go>.

For each tool:

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

<[gotrue-6399fb04c135edab42682c9f4bcf44e6c1bece89-tool\_4.txt](https://docs.google.com/document/d/1iT1qfddEgCm0a8qHorjpOUL63qobH9tGftouGSnTq5U/edit?usp=sharing)>

The output is effective at convincing me that <line 151 & 236> in <api/hooks.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?

| It clearly indicate the error like  [gotrue/gotrue-6399fb04c135edab42682c9f4bcf44e6c1bece89/api/hooks.go:236] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)  235: if body != nil {  > 236: body.Close()  237: }  [gotrue/gotrue-6399fb04c135edab42682c9f4bcf44e6c1bece89/api/hooks.go:151] - G104 (CWE-703): Errors unhandled. (Confidence: HIGH, Severity: LOW)  150: if rsp != nil && rsp.Body != nil {  > 151: rsp.Body.Close()  152: } |
| --- |

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?

| It also shows the prefer fix, which is not exactly as in the commit but very similar |
| --- |