



DELVE

A Debugger for the Go Programming Language

Notes – 2023/01

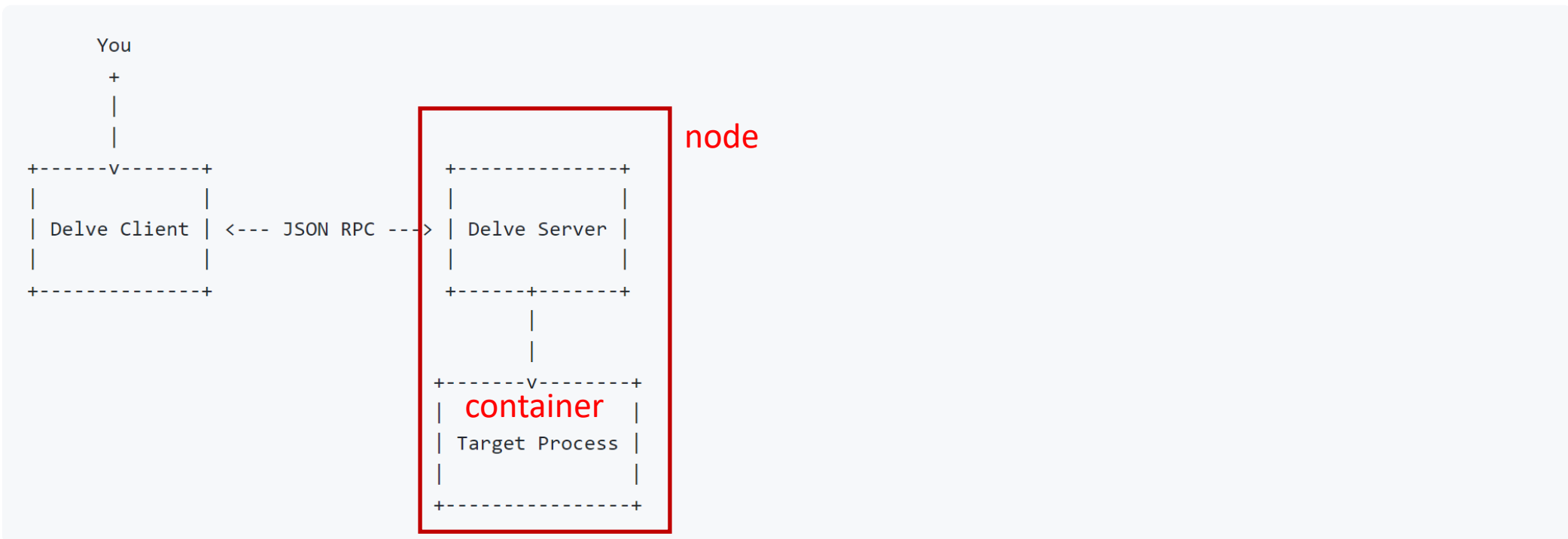
Background

- We want to debug/trace the program easier

Remote Debugging

Delve has the ability to attach to processes over a network via a JSON RPC API. You attach to your target process as a server and then connect that server with a client:

[Reference](#)



Preparation

Build binary without optimization

```
jeff@SUSE-387793:~/go/src/github.com/neuvector/neuvector/controller$ cat Makefile
all:
    # go build --ldflags '-extldflags "-static"'
    # go build -ldflags='-s -w'
    go build -gcflags=all="-N -l"
```

Install delve on the node
(running container pod)

```
neuvector@ubuntu2204-G:~$ which dlv
/home/neuvector/go/bin/dlv
neuvector@ubuntu2204-G:~$ dlv version
Delve Debugger
Version: 1.20.1
Build: $Id: 96e65b6c615845d42e0e31d903f6475b0e4ece6e $
neuvector@ubuntu2204-G:~$ |
```

Install delve on build environment

```
jeff@SUSE-387793:~$ which dlv
/home/jeff/go/bin/dlv
jeff@SUSE-387793:~$ dlv version
Delve Debugger
Version: 1.20.1
Build: $Id: 96e65b6c615845d42e0e31d903f6475b0e4ece6e $
jeff@SUSE-387793:~$ |
```


[Installation](#)

debug (1)

On the node, start debug server

```
neuvector@ubuntu2204-G:~$ cat dlv_server.sh
#!/usr/bin/bash

CONTROLLER_PID=`pidof controller`
sudo /home/neuvector/go/bin/dlv attach ${CONTROLLER_PID} --headless --accept-multiclient --only-same-user=false --api-version 2 --listen=:12345
neuvector@ubuntu2204-G:~$
```



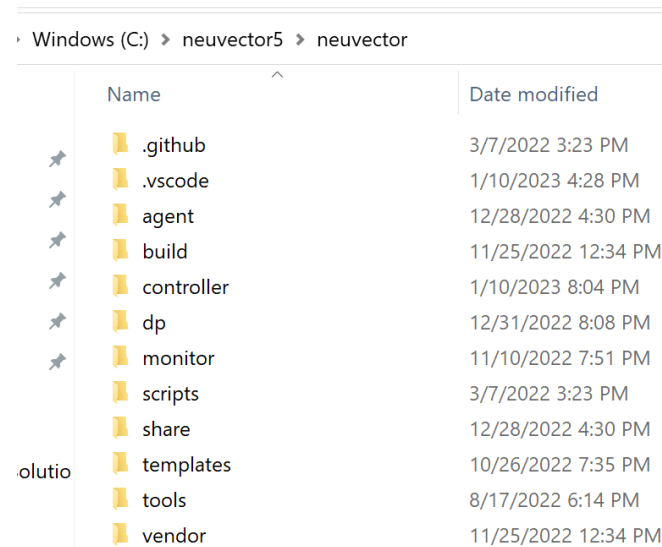
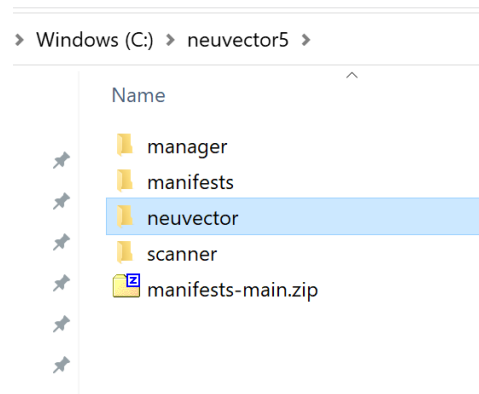
#1 - On the build env, connect to debug server

```
jeff@SUSE-387793:~/go/src/github.com/neuvector/neuvector/controller$ dlv connect 10.1.45.45:12345
Type 'help' for list of commands.
(dlv) |
```

#2 – On non-build env, like Windows. See next page.

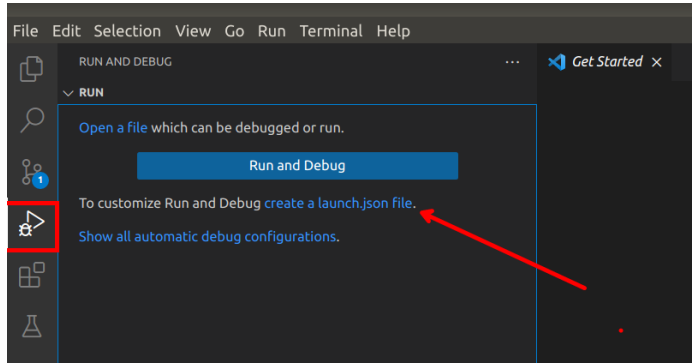
debug (2) from Windows

- In my case, these code is checkout from github directly. Nothing to do with build environment.

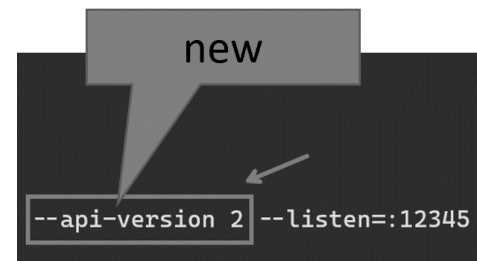
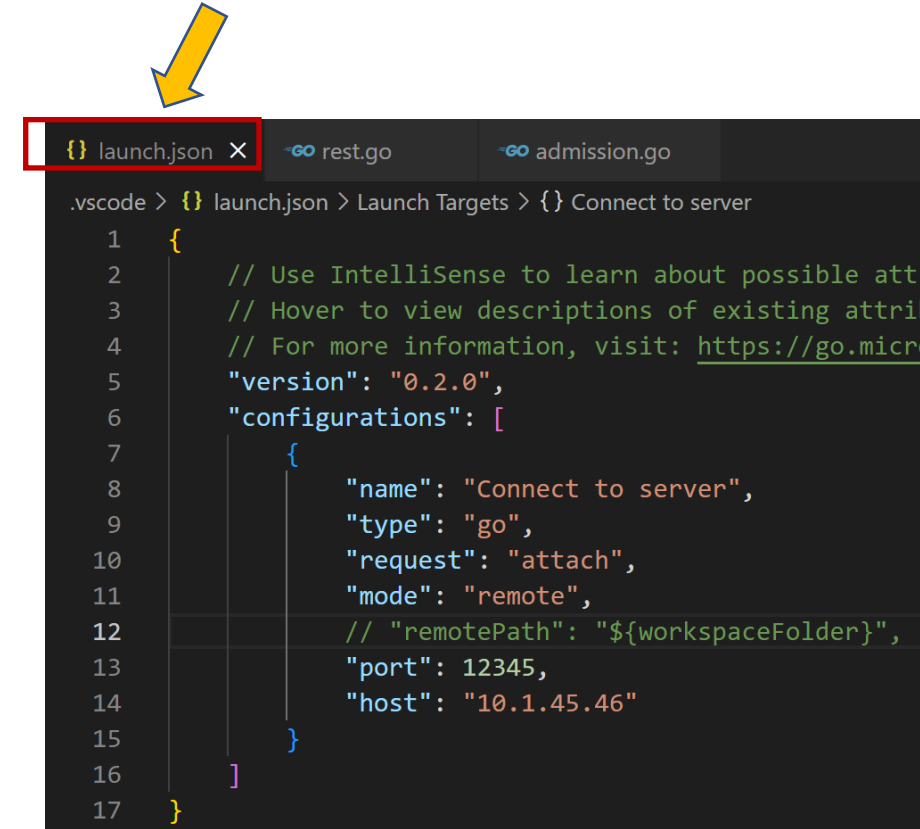
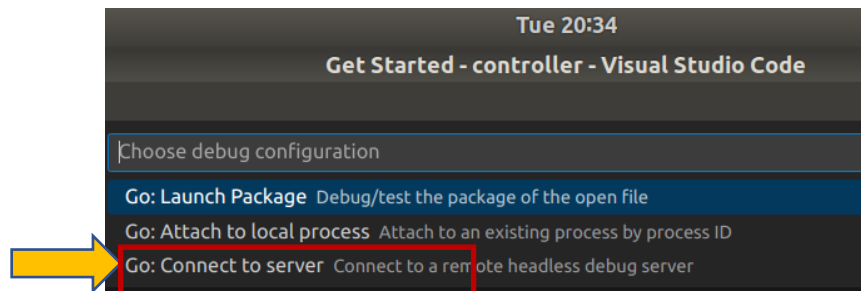
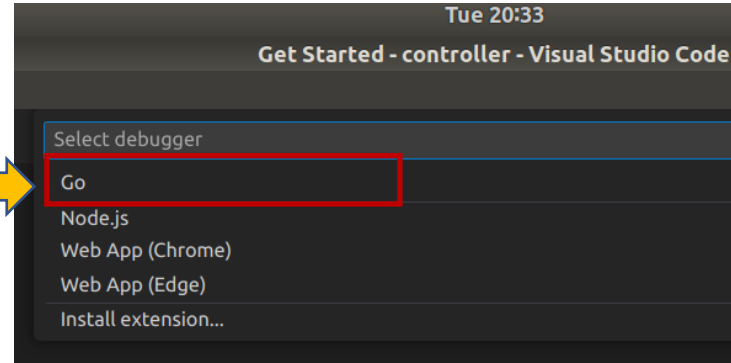


```
PS C:\neuvector5\neuvector\controller> code .
```

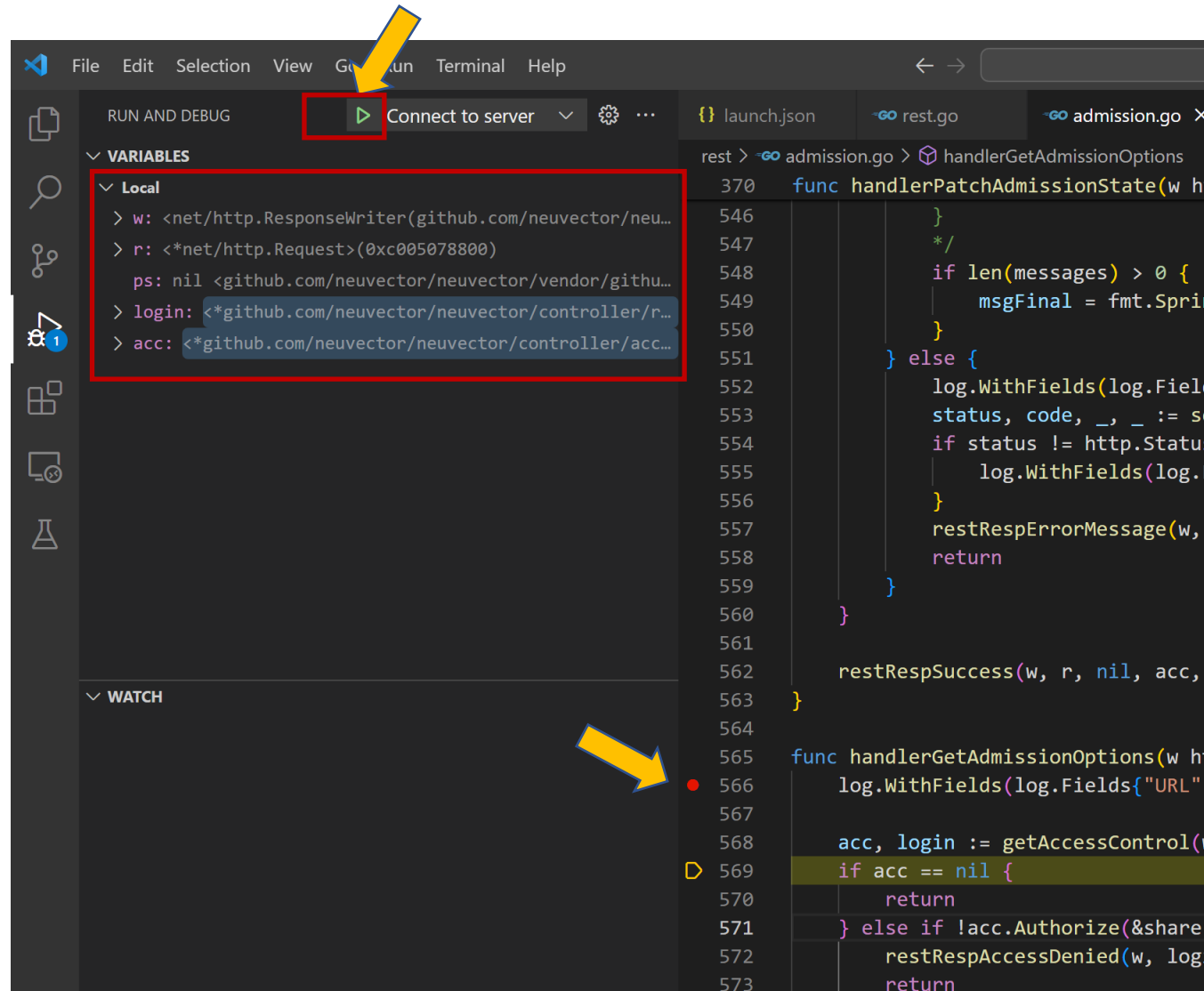
debug (2) from Windows



create launch.json
(select remote debug)



debug (2) from Windows



usage examples (1)

funcs (find symbols)

```
(dlv) funcs handlerGet  
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions  
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionRule  
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionRules  
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionState  
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionTest  
github.com/neuvector/neuvector/controller/rest.handlerGetFedJoinToken  
github.com/neuvector/neuvector/controller/rest.handlerGetFedMember  
github.com/neuvector/neuvector/controller/rest.handlerGetGroupCfgImport  
github.com/neuvector/neuvector/controller/rest.handlerGetIBMSAConfig  
github.com/neuvector/neuvector/controller/rest.handlerGetIBMSAEPInfo  
github.com/neuvector/neuvector/controller/rest.handlerGetIBMSAEPSetupToken  
github.com/neuvector/neuvector/controller/rest.handlerGetIBMSASetupURL  
github.com/neuvector/neuvector/controller/rest.handlerGetJointClusterView  
github.com/neuvector/neuvector/controller/rest.handlerGetRolePermissionOptions  
(dlv)
```

// set breakpoint

```
(dlv) funcs handlerGetAdmissionOptions  
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions  
(dlv)
```

```
(dlv) b github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions  
Breakpoint 2 set at 0x30439eb for github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions() ./rest/admission.go:565
```

when attach to the process, it is in paused status.

After you done action like set breakpoint, observe values... remember to [continue] via c

If you want to pause it, press [ctrl+c]

```
(dlv) c  
|
```


(dlv) n

> github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions() ./rest/admission.go:566 (PC: 0x3043a02)

```
561:
562:     restRespSuccess(w, r, nil, acc, login, &rconf, msgFinal)
563: }
564:
565: func handlerGetAdmissionOptions(w http.ResponseWriter, r *http.Request, ps httprouter.Params) {
=> 566:     log.WithFields(log.Fields{"URL": r.URL.String()}).Debug()
567:
568:     acc, login := getAccessControl(w, r, "")
569:     if acc == nil {
570:         return
571:     } else if !acc.Authorize(&share.CLUSAdmissionState{}, nil) {
```

(dlv) p w

```
net/http.ResponseWriter(github.com/neuvector/neuvector/controller/rest.writer) {
  req: *net/http.Request {
    Method: "GET",
    URL: (*"net/url.URL")(0xc00071b880),
    Proto: "HTTP/1.1",
    ProtoMajor: 1,
    ProtoMinor: 1,
    Header: net/http.Header [...],
    Body: io.ReadCloser(net/http.noBody) (*io.ReadCloser)(0xc000125340),
    GetBody: nil,
    ContentLength: 0,
    TransferEncoding: []string len: 0, cap: 0, nil,
    Close: false,
    Host: "10.1.45.44:31693",
    Form: net/url.Values nil,
    PostForm: net/url.Values nil,
    MultipartForm: *mime/multipart.Form nil,
    Trailer: net/http.Header nil,
    RemoteAddr: "10.32.0.1:54853",
    RequestURI: "/v1/admission/options",
    TLS: (*"crypto/tls.ConnectionState")(0xc00097d1e0),
```

>> we can also see inner field..

(dlv) p w.req.Method

"GET"

(dlv) c

T

```

(dlv) p cache.admValidateDenyCache ←
github.com/neuvector/neuvector/share.CLUSAdmissionRules {
  RuleMap: map[uint32]*github.com/neuvector/neuvector/share.CLUSAdmissionRule [
    1001: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc004720b60),
    1003: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc0066bb880),
    1004: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc0061d7b20),
    1005: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc00663f340),
    1008: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc00044e4d0),
    1000: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc000566af0),
    1002: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc007baabd0),
    1006: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc006df27e0),
    1007: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc00039fb90),
    1009: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc007a7d1f0),
    1010: (*"github.com/neuvector/neuvector/share.CLUSAdmissionRule")(0xc005897e30),
  ],
  RuleHeads: []*github.com/neuvector/neuvector/share.CLUSRuleHead len: 11, cap: 13, [
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb280),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb320),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb460),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb500),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb5a0),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb640),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb780),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb820),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb8c0),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cb960),
    (*"github.com/neuvector/neuvector/share.CLUSRuleHead")(0xc0183cbaa0),
  ],}
(dlv) p cache.admValidateDenyCache.RuleMap["1001"]
Command failed: can not convert "1001" constant to uint32
(dlv) p cache.admValidateDenyCache.RuleMap[1001] ←
*github.com/neuvector/neuvector/share.CLUSAdmissionRule {
  ID: 1001,
  Category: "Kubernetes",
  Comment: "test2",
  Criteria: []*github.com/neuvector/neuvector/share.CLUSAdmRuleCriterion len: 1, cap: 4, [
    (*"github.com/neuvector/neuvector/share.CLUSAdmRuleCriterion")(0xc00f2ac6e0),
  ],
  Disable: false,
  Critical: false,
  CfgType: 2,
  RuleType: "deny",
  UseAsRiskyRoleTag: false,}
(dlv) |

```

usage examples (3)

trace

Set tracepoint.

```
trace [name] [locspec]
```

```
jeff@SUSE-387793:~/go/src/github.com/neuvector/neuvector/controller$ dlv connect 10.1.45.45:12345
Type 'help' for list of commands.
(dlv) funcs handlerGetAdmissionOptions
github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions
(dlv) trace github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions
Tracepoint 4 set at 0x304392b for github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions
(dlv) c
> goroutine(4713): github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions(net/http.ResponseWriter(*net/http.response) ... }, ("*net/http.Request")(0xc002e40000), github.com/neuvector/neuvector/
>> goroutine(4713): => ()
> goroutine(4688): github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions(net/http.ResponseWriter(*net/http.response) ... }, ("*net/http.Request")(0xc000600400), github.com/neuvector/neuvector/
>> goroutine(4688): => ()
> goroutine(4701): github.com/neuvector/neuvector/controller/rest.handlerGetAdmissionOptions(net/http.ResponseWriter(*net/http.response) ... }, ("*net/http.Request")(0xc00079b400), github.com/neuvector/neuvector/
>> goroutine(4701): => ()
```

trace and print

Frame 1: /home/jeff/go/src/github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter/router.go:387 (PC: 2efd260)

```
382:
383:     path := req.URL.Path
384:
385:     if root := r.trees[req.Method]; root != nil {
386:         if handle, ps, tsr := root.getValue(path); handle != nil {
=> 387:             handle(w, req, ps)
388:             return
389:         } else if req.Method != http.MethodConnect && path != "/" {
390:             code := 301 // Permanent redirect, request with GET method
391:             if req.Method != http.MethodGet {
392:                 // Temporary redirect, request with same method
```

```
>> set tracepoint on line 388 and print the req.Method and req.URL.Path
jeff@SUSE-387793:~/go/src/github.com/neuvector/neuvector/controller$ dlv connect 10.1.45.45:12345
```

```
trace [name] [locspec]
```

```
(dlv) t /home/jeff/go/src/github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter/router.go:388
```

```
Tracepoint 5 set at 0x2efd260 for github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter.(*Router).S
/home/jeff/go/src/github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter/router.go:388
```

```
(dlv) on 5 p req.URL.Path
```

```
(dlv) c
```

```
> goroutine(172424):
```

```
github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter.(*Router).ServeHTTP(("*github.com/neuvector/n
uter.Router")(0xc00042aa20), net/http.ResponseWriter(github.com/neuvector/neuvector/controller/rest.writer) {req: ("*ne
net/http.ResponseWriter(*net/http.response) ...}, ("*net/http.Request")(0xc0028b4a00))
```

```
    req.URL.Path: "/v1/admission/rule"
```

```
> goroutine(172776):
```

```
github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter.(*Router).ServeHTTP(("*github.com/neuvector/n
uter.Router")(0xc00042aa20), net/http.ResponseWriter(github.com/neuvector/neuvector/controller/rest.writer) {req: ("*ne
net/http.ResponseWriter(*net/http.response) ...}, ("*net/http.Request")(0xc005a9ee00))
```

```
    req.URL.Path: "/v1/auth"
```

```
> goroutine(172777):
```

```
github.com/neuvector/neuvector/vendor/github.com/julienschmidt/httprouter.(*Router).ServeHTTP(("*github.com/neuvector/n
```

```
// delve show global variable
```

```
controller > resource > kubernetes\_rbac.go > ...  
1 package resource
```

```
133 var nvSA string = "default"
```

```
(dlv) p nvSA
Command failed: could not find symbol value for nvSA
(dlv) funcs k8s2NVRole
github.com/neuvector/neuvector/controller/resource.k8s2NVRole
(dlv) p resource.nvSA
"default"
```

```
(dlv) p resource.nvClusterRoles
map[string]*github.com/neuvector/neuvector/controller/resource.k8sClusterRoleInfo [
    "neuvector-binding-app": *{
        rules: []*github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo len: 1, cap: 1, [
            (*"github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo")(0xc000113080),
        ],},
    "neuvector-binding-rbac": *{
        rules: []*github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo len: 1, cap: 1, [
            (*"github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo")(0xc0001130e0),
        ],},
    "neuvector-binding-admission": *{
        rules: []*github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo len: 1, cap: 1, [
            (*"github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo")(0xc000113140),
        ],},
}
```

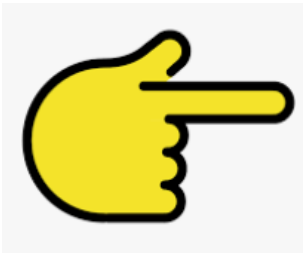


```
(dlv) p resource.nvClusterRoles["neuvector-binding-app"].rules.apiGroup
Command failed: (type []*github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo) is not a struct
(dlv) p resource.nvClusterRoles["neuvector-binding-app"].rules[0] ←
*github.com/neuvector/neuvector/controller/resource.k8sClusterRoleRuleInfo {
    apiGroup: "",
    resources: github.com/neuvector/neuvector/share/utils.Set(*github.com/neuvector/neuvector/share/utils.threadU
        (*interface {})(0xc00037bb98): {},
        (*interface {})(0xc00037bba8): {},
        (*interface {})(0xc00037bbb8): {},
        (*interface {})(0xc00037bbc8): {},
    ],
    verbs: github.com/neuvector/neuvector/share/utils.Set(*github.com/neuvector/neuvector/share/utils.threadUnsaf
        (*interface {})(0xc00037b7a8): {},
        (*interface {})(0xc00037b7b8): {},
        (*interface {})(0xc00037b7c8): {},
        (*interface {})(0xc00037b7d8): {},
    ],}
(dlv) p resource.nvClusterRoles["neuvector-binding-app"].rules[0].verbs ←
github.com/neuvector/neuvector/share/utils.Set(*github.com/neuvector/neuvector/share/utils.threadUnsafeSet) *[]
    "get": {},
    "list": {},
    "update": {},
    "watch": {},
]
(dlv) p resource.nvClusterRoles["neuvector-binding-app"].rules[0].resources ←
github.com/neuvector/neuvector/share/utils.Set(*github.com/neuvector/neuvector/share/utils.threadUnsafeSet) *[]
    "namespaces": {},
    "nodes": {},
    "pods": {},
    "services": {},
]
(dlv) |
```

Nice resource

- <https://github.com/jasonkeene/debugging-workshop>

MUST READ



☰ README.md

Gophercon 2020 Debugging Workshop

To get back to here: bit.ly/debugging-workshop

Slack Channel: [#gophercon-2020-debugging-workshop](#)

Course Objective

Solving bugs can be incredibly challenging. It can often feel like our programs have the upper hand. My goal for this course is to instill in you the confidence that you can understand and squash your bugs, no matter how challenging they are.

The next time you encounter a difficult bug, I'd like for you to be able to say:

I don't know what is wrong with my program but I know how to find out!

In this course, we will be covering a number of tools that I have found useful in understanding Go programs.

Exercises

This workshop is broken down into a series of exercises. We will go through these individually during the class:

- [Introduction](#)
- [Starting a Debugging Session](#)
- [Navigating Your Program](#)
- [Demystifying Debuggers](#)
- [Remote Debugging](#)
- [Debugging Inside a Container](#)
- [Debugging from Core Dumps](#)
- [Automating Delve](#)
- [Deterministic Debugging](#)
- [pprof](#)
- [bpftrace](#)

References (1)

3.9 Delve 调试器

<https://chai2010.cn/advanced-go-programming-book/ch3-asm/ch3-09-debug.html>

Commands

Running the program

Command	Description
call	Resumes process, injecting a function call (EXPERIMENTAL!!!)
continue	Run until breakpoint or program termination.
next	Step over to next source line.
rebuild	Rebuild the target executable and restarts it. It does not work if the executable was not built by delve.
restart	Restart process.
rev	Reverses the execution of the target program for the command specified.
rewind	Run backwards until breakpoint or start of recorded history.
step	Single step through program.
step-instruction	Single step a single cpu instruction.
stepout	Step out of the current function.

<https://github.com/go-delve/delve/blob/master/Documentation/cli/README.md>

References (2)

Delve lets you script the debugger and add your own commands

To provide an API for users to automate Delve or add new commands, Delve uses the scripting language [Starlark](#), which is similar to Python and was developed by the project that creates the [Bazel build tool](#). See [Delve's API documentation](#) for more about this tool and how to use it.

[Link](#)

substitute-path

Can not set breakpoints or see source listing in a complicated debugging environment

This problem manifests when one or more of these things happen:

- Can not see source code when the program stops at a breakpoint
- Setting a breakpoint using full path, or through an IDE, does not work

[Source](#)

While doing one of the following things:

- **The program is built and run inside a container** and Delve (or an IDE) is remotely connecting to it
- Generally, every time the build environment (VM, container, computer...) differs from the environment where Delve's front-end (dlv or a IDE) runs
- Using `-trimpath` or `-gcflags=-trimpath`
- Using a build system other than `go build` (eg. bazel)
- Using symlinks in your source tree

If you are affected by this problem then the `list main.main` command (in the command line interface) will have this result:

```
(dlv) list main.main
Showing /path/to/the/mainfile.go:42 (PC: 0x47dfca)
Command failed: open /path/to/the/mainfile.go: no such file or directory
(dlv)
```

This is not a bug. The Go compiler embeds the paths of source files into the executable so that debuggers, including Delve, can use them. Doing any of the things listed above will prevent this feature from working seamlessly.

The substitute-path feature can be used to solve this problem, see `help config` or the `substitutePath` option in `launch.json`.