

Commit

✓ **fix: AdmissionPolicyGroup must not have context aware resources**

[Browse files](#)

Prior to this commit, it was possible to declare AdmissionPolicyGroup resources with policies that had context aware resources.

Context awareness allows a policy to query the Kubernetes API with `list` and `get` verbs. The queries are run using the ServiceAccount of the Policy Server instance that hosts the policy.

AdmissionPolicyGroup are namespaced CRDs which can be created by non-admin Kubernetes users.

The ability to deploy AdmissionPolicyGroup with context aware resources exposes the cluster to potential leaks of data.

For example, a non-admin user, might create an AdmissionPolicyGroup with one of its policies that has access to Kubernetes Secrets. The policy would then be able to `get` Secrets that are not available to the non-admin user (if the Policy Server is running with a ServiceAccount that has been explicitly configured to have `get` access to all the Secrets of the cluster).

This commit fixes the by removing the `ContextAwareResources` field from the `AdmissionPolicyGroup`.

== What happens after this commit is deployed

The existing AdmissionPolicyGroup will be automatically updated and they will lose their `.spec.policies[*].ContextAwareResources` field (if set).

The kubewarden controller will also automatically reconcile the Policy Server configuration.

Signed-off-by: Flavio Castelli <fcastelli@suse.com>

main (#984)
v1.21.0 v1.21.0-rc2

flavio committed 2 weeks ago Verified 1 parent [04225f4](#) commit [51a88df](#)

Showing 13 changed files with 522 additions and 237 deletions. Whitesp... Ignore whitesp... S... Uni...

Filter changed files

19 ■■■■■
api/policies/v1/admissionpolicygroup_types.go

- admissionpolicygro... 
 - clusteradmissionpol... 
 - factories.go 
 - policy.go 
 - policy_types.go 
 - policygroup_validati... 
 - policygroup_validati... 
 - zz_generated.deepc... 
- config/crd/bases
 - policies.kubewarde... 
- docs/crds
 - CRD-docs-for-docs-r... 
 - CRD-docs-for-docs-r... 
- internal/controller
 - policyserver_control... 
 - policyserver_control... 

```

75     }
76     }
77     func (r *AdmissionPolicyGroup) IsContextAware()
78     bool {
79         -     for _, policy := range r.Spec.Policies {
80             -         if
81                 len(policy.ContextAwareResources) > 0 {
82                     -                 return true
83             }
84         }
85     }
86     // We return false here because
87     AdmissionPolicyGroup have no access to context
88     aware resources.
89     return false
90 }
91
92 func (r *AdmissionPolicyGroup)
93 GetPolicyGroupMembers() PolicyGroupMembers {
94     -     return r.Spec.Policies
95 }
96 + func (r *AdmissionPolicyGroup)
97 GetPolicyGroupMembersWithContext()
98 PolicyGroupMembersWithContext {
99     +     policies :=
100     make(PolicyGroupMembersWithContext)
101     +     for name, policy := range r.Spec.Policies
102     {
103         +         policies[name] =
104         PolicyGroupMemberWithContext{
105             +             PolicyGroupMember: policy,
106             +             // AdmissionPolicyGroup
107             does not have access to context aware resources.
108             ContextAwareResources:
109             []ContextAwareResource{},
110         }
111     }
112     +     return policies
113 }
114
115 func (r *AdmissionPolicyGroup) GetSettings()
116 runtime.RawExtension {

```

14 

api/policies/v1/clusteradmissionpolicygroup_types.go 

```

22     "k8s.io/apimachinery/pkg/runtime"
23 )
24
25 + type ClusterPolicyGroupSpec struct {
26 +     GroupSpec `json:""`

```

```

27 +
28 + // Policies is a list of policies that are
    part of the group that will
29 + // be available to be called in the
    evaluation expression field.
30 + // Each policy in the group should be a
    Kubewarden policy.
31 + // +kubebuilder:validation:Required
32 + Policies PolicyGroupMembersWithContext
    `json:"policies"`
33 + }
34 +
25 35 // ClusterAdmissionPolicyGroupSpec defines the
    desired state of ClusterAdmissionPolicyGroup.
26 36 type ClusterAdmissionPolicyGroupSpec struct {
27 37 - PolicyGroupSpec `json:""`
    + ClusterPolicyGroupSpec `json:""`
28 38
29 39 // NamespaceSelector decides whether to
    run the webhook on an object based
30 40 // on whether the namespace for that
    object matches the selector. If the
148 158 return &r.Status
149 159 }
150 160
151 - func (r *ClusterAdmissionPolicyGroup)
    GetPolicyGroupMembers() PolicyGroupMembers {
161 + func (r *ClusterAdmissionPolicyGroup)
    GetPolicyGroupMembersWithContext()
    PolicyGroupMembersWithContext {
152 162 return r.Spec.Policies
153 163 }
154 164

```

44 api/policies/v1/factories.go

```

314 314 },
315 315 Spec: AdmissionPolicyGroupSpec{
316 316 PolicyGroupSpec:
    PolicyGroupSpec{
317 - PolicyServer:
    f.policyServer,
318 - Policies:
    f.policyMembers,
319 - Expression:
    f.expression,
320 - Rules:
    f.rules,
321 - MatchConditions:
    f.matchConds,
322 - Mode:
    f.mode,

```

```

317 +                                     GroupSpec:
318 + GroupSpec{
319 +     PolicyServer:    f.policyServer,
320 +     Expression:     f.expression,
321 +                                     Rules:
322 +     f.rules,
323 +     MatchConditions: f.matchConds,
324 +                                     Mode:
325 +     f.mode,
326 +                                     },
327 +                                     Policies:
328 +     f.policyMembers,
329 +                                     },
330 +                                     },
331 +                                     }
332 +                                     policyServer string
333 +                                     rules
334 +                                     []admissionregistrationv1.RuleWithOperations
335 +                                     expression string
336 -                                     policyMembers PolicyGroupMembers
337 +                                     policyMembers
338 +                                     PolicyGroupMembersWithContext
339 +                                     matchConds
340 +                                     []admissionregistrationv1.MatchCondition
341 +                                     mode PolicyMode
342 +                                     }
343 +                                     },
344 +                                     },
345 +                                     expression: "pod_privileged() &&
346 +                                     user_group_psp()",
347 -                                     policyMembers: PolicyGroupMembers{
348 +                                     policyMembers:
349 +                                     PolicyGroupMembersWithContext{
350 +                                     "pod_privileged": {
351 -                                     Module:
352 +                                     "registry://ghcr.io/kubewarden/tests/pod-
353 +                                     privileged:v0.2.5",
354 +                                     PolicyGroupMember:
355 +                                     PolicyGroupMember{
356 +                                     Module:
357 +                                     "registry://ghcr.io/kubewarden/tests/pod-
358 +                                     privileged:v0.2.5",
359 +                                     },
360 +                                     },
361 +                                     "user_group_psp": {
362 -                                     Module:
363 +                                     "registry://ghcr.io/kubewarden/tests/user-group-
364 +                                     psp:v0.4.9",

```

```

365 + PolicyGroupMember{
366 + Module:
    "registry://ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
367 + },
362 368 },
363 369 },
364 370 matchConds:
    []admissionregistrationv1.MatchCondition{
378 384     return f
379 385 }
380 386
381 - func (f *ClusterAdmissionPolicyGroupFactory)
    WithMembers(members PolicyGroupMembers)
    *ClusterAdmissionPolicyGroupFactory {
387 + func (f *ClusterAdmissionPolicyGroupFactory)
    WithMembers(members PolicyGroupMembersWithContext)
    *ClusterAdmissionPolicyGroupFactory {
382 388     f.policyMembers = members
383 389     return f
384 390 }
413 419 },
414 420 },
415 421 Spec:
    ClusterAdmissionPolicyGroupSpec{
416 - PolicyGroupSpec:
    PolicyGroupSpec{
417 - PolicyServer:
    f.policyServer,
418 - Policies:
    f.policyMembers,
419 - Expression:
    f.expression,
420 - Rules:
    f.rules,
421 - MatchConditions:
    f.matchConds,
422 - Mode:
    f.mode,
422 + ClusterPolicyGroupSpec:
    ClusterPolicyGroupSpec{
423 + GroupSpec:
    GroupSpec{
424 + PolicyServer: f.policyServer,
425 + Expression: f.expression,
426 + Rules:
    f.rules,
427 + MatchConditions: f.matchConds,

```

```

428 +                                     Mode:
429 +                                     },
430 +                                     Policies:
      f.policyMembers,
423 431                                     },
424 432                                     },
425 433                                     }

```

2 api/policies/v1/policy.go

```

151 151 // +kubebuilder:object:generate:=false
152 152 type PolicyGroup interface {
153 153     Policy
154 154 -     GetPolicyGroupMembers() PolicyGroupMembers
      154 +     GetPolicyGroupMembersWithContext()
      PolicyGroupMembersWithContext
155 155     GetExpression() string
156 156     GetMessage() string
157 157 }

```

12 api/policies/v1/policy_types.go

```

170 170 //
      +kubebuilder:pruning:PreserveUnknownFields
171 171 // x-kubernetes-embedded-resource: false
172 172 Settings runtime.RawExtension
      `json:"settings,omitempty"`
      173 + }
      174 +
175 175 + type PolicyGroupMembersWithContext
      map[string]PolicyGroupMemberWithContext
176 176 +
177 177 + type PolicyGroupMemberWithContext struct {
178 178 +     PolicyGroupMember `json:""`
173 179
174 180 // List of Kubernetes resources the policy
      is allowed to access at evaluation time.
175 181 // Access to these resources is done using
      the `ServiceAccount` of the PolicyServer
178 184 ContextAwareResources
      []ContextAwareResource
      `json:"contextAwareResources,omitempty"`
179 185 }
180 186
181 187 - type PolicyGroupSpec struct {
      187 + type GroupSpec struct {
182 188 // PolicyServer identifies an existing
      PolicyServer resource.
183 189 // +kubebuilder:default:=default
184 190 // +optional
302 308 // returned in the warning field of the
      response.

```

```

303     309 // +kubebuilder:validation:Required
304     310     Message string `json:"message"`
305     311 + }
306     312 +
307     313 + type PolicyGroupSpec struct {
308     314 +     GroupSpec `json:""`
309     315
310     316 // Policies is a list of policies that are
311     317 // part of the group that will
312     // be available to be called in the
313     // evaluation expression field.

```

6 api/policies/v1/policygroup_validation.go

```

56     56 // validatePolicyGroupMembers validates that a
57     57 // policy group has at least one policy member.
58     58 func validatePolicyGroupMembers(policyGroup
59     59 PolicyGroup) field.ErrorList {
60     60     var allErrors field.ErrorList
61     61     if
62     62 - len(policyGroup.GetPolicyGroupMembers()) == 0 {
63     63 + len(policyGroup.GetPolicyGroupMembersWithContext()
64     64 ) == 0 {
65     65         allErrors = append(allErrors,
66     66         field.Required(field.NewPath("spec").Child("policie
67     67         s"), "policy groups must have at least one policy
68     68         member"))
69     69     }
70     70     for memberName := range
71     71 policyGroup.GetPolicyGroupMembers() {
72     72 + for memberName := range
73     73 policyGroup.GetPolicyGroupMembersWithContext() {
74     74         _, matchReservedSymbol :=
75     75         celReservedSymbols[memberName]
76     76         if len(memberName) == 0 ||
77     77         matchReservedSymbol ||
78     78         !idenRegex.MatchString(memberName) {
79     79             allErrors =
80     80             append(allErrors,
81     81             field.Invalid(field.NewPath("spec").Child("policie
82     82             s"), memberName, "policy group member name is
83     83             invalid"))
84     84
85     85 // Create a CEL environment with custom
86     86 // functions for each policy member
87     87     var opts []cel.EnvOption
88     88     for policyName := range
89     89 policyGroup.GetPolicyGroupMembers() {
90     90 + for policyName := range
91     91 policyGroup.GetPolicyGroupMembersWithContext() {
92     92         fn := cel.Function(policyName,
93     93         cel.Overload(policyName, []*cel.Type{},

```

```
types.BoolType))
86      86      opts = append(opts, fn)
87      87      }
```

150  api/policies/v1/policygroup_validation_test.go 

```
21      21      Name:
22      22      "testing-cluster-policy-group",
23      23      },
24      23      Spec:
25      24      ClusterAdmissionPolicyGroupSpec{
26      25      - PolicyGroupSpec: PolicyGroupSpec{
27      26      - Expression: "policy1() && policy2()",
28      27      - Message: "This is a test policy",
29      28      - Policies: PolicyGroupMembers{
30      29      + ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
31      30      + GroupSpec: GroupSpec{
32      31      + Expression: "policy1() && policy2()",
33      32      + Message: "This is a test policy",
34      33      + Policies: PolicyGroupMembersWithContext{
35      34      "policy1": {
36      35      - Module: "ghcr.io/kubewarden/tests/user-group-
37      36      - psp:v0.4.9",
38      37      + PolicyGroupMember: PolicyGroupMember{
39      38      + Module: "ghcr.io/kubewarden/tests/user-group-
40      39      + psp:v0.4.9",
41      40      + },
42      41      },
43      42      },
44      43      "policy2": {
45      44      - Module: "ghcr.io/kubewarden/tests/safe-
46      45      labels:v1.0.0",
47      46      + PolicyGroupMember: PolicyGroupMember{
```

```

37 + Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",
38 +
    },
33 39
    },
34 40
    },
35 41
    },
44 50 Name:
    "testing-cluster-policy-group",
45 51
    },
46 52 Spec:
    ClusterAdmissionPolicyGroupSpec{
47 -
    PolicyGroupSpec: PolicyGroupSpec{
48 -
    Expression: "",
49 -
    Message: "This is a test policy",
50 -
    Policies: PolicyGroupMembers{
53 +
    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
54 +
    GroupSpec: GroupSpec{
55 +
    Expression: "",
56 +
    Message: "This is a test policy",
57 +
    },
58 +
    Policies: PolicyGroupMembersWithContext{
51 59
    "policy1": {
52 -
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
60 +
    PolicyGroupMember: PolicyGroupMember{
61 +
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
62 +
    },
53 63
    },
54 64
    "policy2": {
55 -
    Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",

```

```

65 + PolicyGroupMember: PolicyGroupMember{
66 +
    Module: "ghcr.io/kubewarden/tests/safe-
        labels:v1.0.0",
67 +
    },
56 68
    },
57 69
58 70
59 71
60 72
61 73
62 74
63 75
64 76
65 77
66 78
67 79
68 80
69 81
70 82
71 83
72 84
73 85
74 86
75 87
76 88
77 89
78 90
79 91
80 92
81 93
82 94
83 95
84 96
85 97
86 98
87 99
88 100
89 101
90 102
91 103
92 104
93 105
94 106
95 107
96 108
97 109
98 110
99 111
100 112
101 113
102 114
103 115
104 116
105 117
106 118
107 119
108 120
109 121
110 122
111 123
112 124
113 125
114 126
115 127
116 128
117 129
118 130
119 131
120 132
121 133
122 134
123 135
124 136
125 137
126 138
127 139
128 140
129 141
130 142
131 143
132 144
133 145
134 146
135 147
136 148
137 149
138 150
139 151
140 152
141 153
142 154
143 155
144 156
145 157
146 158
147 159
148 160
149 161
150 162
151 163
152 164
153 165
154 166
155 167
156 168
157 169
158 170
159 171
160 172
161 173
162 174
163 175
164 176
165 177
166 178
167 179
168 180
169 181
170 182
171 183
172 184
173 185
174 186
175 187
176 188
177 189
178 190
179 191
180 192
181 193
182 194
183 195
184 196
185 197
186 198
187 199
188 200
189 201
190 202
191 203
192 204
193 205
194 206
195 207
196 208
197 209
198 210
199 211
200 212
201 213
202 214
203 215
204 216
205 217
206 218
207 219
208 220
209 221
210 222
211 223
212 224
213 225
214 226
215 227
216 228
217 229
218 230
219 231
220 232
221 233
222 234
223 235
224 236
225 237
226 238
227 239
228 240
229 241
230 242
231 243
232 244
233 245
234 246
235 247
236 248
237 249
238 250
239 251
240 252
241 253
242 254
243 255
244 256
245 257
246 258
247 259
248 260
249 261
250 262
251 263
252 264
253 265
254 266
255 267
256 268
257 269
258 270
259 271
260 272
261 273
262 274
263 275
264 276
265 277
266 278
267 279
268 280
269 281
270 282
271 283
272 284
273 285
274 286
275 287
276 288
277 289
278 290
279 291
280 292
281 293
282 294
283 295
284 296
285 297
286 298
287 299
288 300
289 301
290 302
291 303
292 304
293 305
294 306
295 307
296 308
297 309
298 310
299 311
300 312
301 313
302 314
303 315
304 316
305 317
306 318
307 319
308 320
309 321
310 322
311 323
312 324
313 325
314 326
315 327
316 328
317 329
318 330
319 331
320 332
321 333
322 334
323 335
324 336
325 337
326 338
327 339
328 340
329 341
330 342
331 343
332 344
333 345
334 346
335 347
336 348
337 349
338 350
339 351
340 352
341 353
342 354
343 355
344 356
345 357
346 358
347 359
348 360
349 361
350 362
351 363
352 364
353 365
354 366
355 367
356 368
357 369
358 370
359 371
360 372
361 373
362 374
363 375
364 376
365 377
366 378
367 379
368 380
369 381
370 382
371 383
372 384
373 385
374 386
375 387
376 388
377 389
378 390
379 391
380 392
381 393
382 394
383 395
384 396
385 397
386 398
387 399
388 400
389 401
390 402
391 403
392 404
393 405
394 406
395 407
396 408
397 409
398 410
399 411
400 412
401 413
402 414
403 415
404 416
405 417
406 418
407 419
408 420
409 421
410 422
411 423
412 424
413 425
414 426
415 427
416 428
417 429
418 430
419 431
420 432
421 433
422 434
423 435
424 436
425 437
426 438
427 439
428 440
429 441
430 442
431 443
432 444
433 445
434 446
435 447
436 448
437 449
438 450
439 451
440 452
441 453
442 454
443 455
444 456
445 457
446 458
447 459
448 460
449 461
450 462
451 463
452 464
453 465
454 466
455 467
456 468
457 469
458 470
459 471
460 472
461 473
462 474
463 475
464 476
465 477
466 478
467 479
468 480
469 481
470 482
471 483
472 484
473 485
474 486
475 487
476 488
477 489
478 490
479 491
480 492
481 493
482 494
483 495
484 496
485 497
486 498
487 499
488 500
489 501
490 502
491 503
492 504
493 505
494 506
495 507
496 508
497 509
498 510
499 511
500 512
501 513
502 514
503 515
504 516
505 517
506 518
507 519
508 520
509 521
510 522
511 523
512 524
513 525
514 526
515 527
516 528
517 529
518 530
519 531
520 532
521 533
522 534
523 535
524 536
525 537
526 538
527 539
528 540
529 541
530 542
531 543
532 544
533 545
534 546
535 547
536 548
537 549
538 550
539 551
540 552
541 553
542 554
543 555
544 556
545 557
546 558
547 559
548 560
549 561
550 562
551 563
552 564
553 565
554 566
555 567
556 568
557 569
558 570
559 571
560 572
561 573
562 574
563 575
564 576
565 577
566 578
567 579
568 580
569 581
570 582
571 583
572 584
573 585
574 586
575 587
576 588
577 589
578 590
579 591
580 592
581 593
582 594
583 595
584 596
585 597
586 598
587 599
588 600
589 601
590 602
591 603
592 604
593 605
594 606
595 607
596 608
597 609
598 610
599 611
600 612
601 613
602 614
603 615
604 616
605 617
606 618
607 619
608 620
609 621
610 622
611 623
612 624
613 625
614 626
615 627
616 628
617 629
618 630
619 631
620 632
621 633
622 634
623 635
624 636
625 637
626 638
627 639
628 640
629 641
630 642
631 643
632 644
633 645
634 646
635 647
636 648
637 649
638 650
639 651
640 652
641 653
642 654
643 655
644 656
645 657
646 658
647 659
648 660
649 661
650 662
651 663
652 664
653 665
654 666
655 667
656 668
657 669
658 670
659 671
660 672
661 673
662 674
663 675
664 676
665 677
666 678
667 679
668 680
669 681
670 682
671 683
672 684
673 685
674 686
675 687
676 688
677 689
678 690
679 691
680 692
681 693
682 694
683 695
684 696
685 697
686 698
687 699
688 700
689 701
690 702
691 703
692 704
693 705
694 706
695 707
696 708
697 709
698 710
699 711
700 712
701 713
702 714
703 715
704 716
705 717
706 718
707 719
708 720
709 721
710 722
711 723
712 724
713 725
714 726
715 727
716 728
717 729
718 730
719 731
720 732
721 733
722 734
723 735
724 736
725 737
726 738
727 739
728 740
729 741
730 742
731 743
732 744
733 745
734 746
735 747
736 748
737 749
738 750
739 751
740 752
741 753
742 754
743 755
744 756
745 757
746 758
747 759
748 760
749 761
750 762
751 763
752 764
753 765
754 766
755 767
756 768
757 769
758 770
759 771
760 772
761 773
762 774
763 775
764 776
765 777
766 778
767 779
768 780
769 781
770 782
771 783
772 784
773 785
774 786
775 787
776 788
777 789
778 790
779 791
780 792
781 793
782 794
783 795
784 796
785 797
786 798
787 799
788 800
789 801
790 802
791 803
792 804
793 805
794 806
795 807
796 808
797 809
798 810
799 811
800 812
801 813
802 814
803 815
804 816
805 817
806 818
807 819
808 820
809 821
810 822
811 823
812 824
813 825
814 826
815 827
816 828
817 829
818 830
819 831
820 832
821 833
822 834
823 835
824 836
825 837
826 838
827 839
828 840
829 841
830 842
831 843
832 844
833 845
834 846
835 847
836 848
837 849
838 850
839 851
840 852
841 853
842 854
843 855
844 856
845 857
846 858
847 859
848 860
849 861
850 862
851 863
852 864
853 865
854 866
855 867
856 868
857 869
858 870
859 871
860 872
861 873
862 874
863 875
864 876
865 877
866 878
867 879
868 880
869 881
870 882
871 883
872 884
873 885
874 886
875 887
876 888
877 889
878 890
879 891
880 892
881 893
882 894
883 895
884 896
885 897
886 898
887 899
888 900
889 901
890 902
891 903
892 904
893 905
894 906
895 907
896 908
897 909
898 910
899 911
900 912
901 913
902 914
903 915
904 916
905 917
906 918
907 919
908 920
909 921
910 922
911 923
912 924
913 925
914 926
915 927
916 928
917 929
918 930
919 931
920 932
921 933
922 934
923 935
924 936
925 937
926 938
927 939
928 940
929 941
930 942
931 943
932 944
933 945
934 946
935 947
936 948
937 949
938 950
939 951
940 952
941 953
942 954
943 955
944 956
945 957
946 958
947 959
948 960
949 961
950 962
951 963
952 964
953 965
954 966
955 967
956 968
957 969
958 970
959 971
960 972
961 973
962 974
963 975
964 976
965 977
966 978
967 979
968 980
969 981
970 982
971 983
972 984
973 985
974 986
975 987
976 988
977 989
978 990
979 991
980 992
981 993
982 994
983 995
984 996
985 997
986 998
987 999
1000 1000

```

```

78 - Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",
94 + PolicyGroupMember: PolicyGroupMember{
95 +
    Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",
96 +
    },
79 97
    },
80 98
81 99
90 108 Name:
    "testing-cluster-policy-group",
91 109
92 110 Spec:
    ClusterAdmissionPolicyGroupSpec{
93 -
    PolicyGroupSpec: PolicyGroupSpec{
94 -
    Expression: "2 > 1",
95 -
    Message: "This is a test policy",
96 -
    Policies: PolicyGroupMembers{
111 +
    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
112 +
    GroupSpec: GroupSpec{
113 +
    Expression: "2 > 1",
114 +
    Message: "This is a test policy",
115 +
116 +
    Policies: PolicyGroupMembersWithContext{
97 117
    "policy1": {
98 -
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
118 +
    PolicyGroupMember: PolicyGroupMember{
119 +
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
120 +
    },
99 121
    },

```

```
100 122 "policy2": {
101 -
    Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",
123 +
    PolicyGroupMember: PolicyGroupMember{
124 +
    Module: "ghcr.io/kubewarden/tests/safe-
    labels:v1.0.0",
125 +
    },
102 126
    },
103 127 },
104 128 },
134 158 Name:
    "testing-cluster-policy-group",
135 159 },
136 160 Spec:
    ClusterAdmissionPolicyGroupSpec{
137 -
    PolicyGroupSpec: PolicyGroupSpec{
138 -
    Expression: "policy1() && policy2()",
139 -
    Message: "This is a test policy",
140 -
    Policies: PolicyGroupMembers{
161 +
    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
162 +
    GroupSpec: GroupSpec{
163 +
    Expression: "policy1() && policy2()",
164 +
    Message: "This is a test policy",
165 +
    },
166 +
    Policies: PolicyGroupMembersWithContext{
141 167
    "policy1": {
142 -
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
168 +
    PolicyGroupMember: PolicyGroupMember{
169 +
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
170 +
    },
    },
```

```

143 171 },
144 172 "policy2": {
145 -
Module: "ghcr.io/kubewarden/tests/safe-
labels:v1.0.0",
173 +
PolicyGroupMember: PolicyGroupMember{
174 +
Module: "ghcr.io/kubewarden/tests/safe-
labels:v1.0.0",
175 +
},
146 176 },
147 177 },
148 178 },
157 187 Name:
"testing-cluster-policy-group",
158 188 },
159 189 Spec:
ClusterAdmissionPolicyGroupSpec{
160 -
PolicyGroupSpec: PolicyGroupSpec{
161 -
Policies: map[string]PolicyGroupMember{},
190 +
ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
191 +
Policies:
map[string]PolicyGroupMemberWithContext{},
162 192 },
163 193 },
164 194 },
171 201 Name:
"testing-cluster-policy-group",
172 202 },
173 203 Spec:
ClusterAdmissionPolicyGroupSpec{
174 -
PolicyGroupSpec: PolicyGroupSpec{
175 -
Policies: PolicyGroupMembers{
204 +
ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
205 +
Policies: PolicyGroupMembersWithContext{
176 206
"": {
177 -
Module: "ghcr.io/kubewarden/tests/user-group-

```

```

    psp:v0.4.9",
207 +
    PolicyGroupMember: PolicyGroupMember{
208 +
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
209 +
    },
178 210
    },
179 211
    },
180 212
    },
189 221
    Name:
    "testing-cluster-policy-group",
190 222
    },
191 223
    Spec:
    ClusterAdmissionPolicyGroupSpec{
192 -
    PolicyGroupSpec: PolicyGroupSpec{
193 -
    Policies: PolicyGroupMembers{
224 +
    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
225 +
    Policies: PolicyGroupMembersWithContext{
194 226
    "in": {
195 -
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
227 +
    PolicyGroupMember: PolicyGroupMember{
228 +
    Module: "ghcr.io/kubewarden/tests/user-group-
    psp:v0.4.9",
229 +
    },
196 230
    },
197 231
    },
198 232
    },
207 241
    Name:
    "testing-cluster-policy-group",
208 242
    },
209 243
    Spec:
    ClusterAdmissionPolicyGroupSpec{
210 -
    PolicyGroupSpec: PolicyGroupSpec{
211 -
    Policies: PolicyGroupMembers{
244 +
    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{

```

```

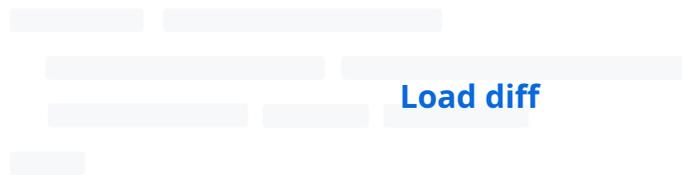
245 + Policies: PolicyGroupMembersWithContext{
212 246
    "policy1": {
213 -
        Module: "ghcr.io/kubewarden/tests/user-group-
            psp:v0.4.9",
247 +
        PolicyGroupMember: PolicyGroupMember{
248 +
            Module: "ghcr.io/kubewarden/tests/user-group-
                psp:v0.4.9",
249 +
            },
214 250
        },
215 251
    },
216 252
    },
225 261
    Name:
        "testing-cluster-policy-group",
226 262
    },
227 263
    Spec:
        ClusterAdmissionPolicyGroupSpec{
228 -
            PolicyGroupSpec: PolicyGroupSpec{
229 -
                Policies: PolicyGroupMembers{
264 +
                    ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
265 +
                        Policies: PolicyGroupMembersWithContext{
230 266
                            "policy1": {
231 -
                                Module: "ghcr.io/kubewarden/tests/user-group-
                                    psp:v0.4.9",
267 +
                                PolicyGroupMember: PolicyGroupMember{
268 +
                                    Module: "ghcr.io/kubewarden/tests/user-group-
                                        psp:v0.4.9",
269 +
                                },
232 270
                            },
233 271
                        },
234 272
                    },
243 281
                Name:
                    "testing-cluster-policy-group",
244 282
            },
245 283
        Spec:
            ClusterAdmissionPolicyGroupSpec{

```

```
246 - PolicyGroupSpec: PolicyGroupSpec{
247 -
284 + Policies: PolicyGroupMembers{
285 +
248 286 ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
249 -
287 + Policies: PolicyGroupMembersWithContext{
288 +
289 +
250 290
251 291
252 -
292 +
293 +
294 +
253 295
254 296
255 297
264 306
265 307
266 308
267 -
268 -
309 +
310 +
Name:
"testing-cluster-policy-group",
},
Spec:
ClusterAdmissionPolicyGroupSpec{
PolicyGroupSpec: PolicyGroupSpec{
-
Policies: PolicyGroupMembers{
+
ClusterPolicyGroupSpec: ClusterPolicyGroupSpec{
+
Policies: PolicyGroupMembersWithContext{
```

269	311	"po0licy1": {
270	-	Module: "ghcr.io/kubewarden/tests/user-group- psp:v0.4.9",
	312	+ PolicyGroupMember: PolicyGroupMember{
	313	+ Module: "ghcr.io/kubewarden/tests/user-group- psp:v0.4.9",
	314	+ },
271	315	},
272	316	"policy21": {
273	-	Module: "ghcr.io/kubewarden/tests/safe- labels:v1.0.0",
	317	+ PolicyGroupMember: PolicyGroupMember{
	318	+ Module: "ghcr.io/kubewarden/tests/safe- labels:v1.0.0",
	319	+ },
274	320	},
275	321	},
276	322	},

150 ■■■■ api/policies/v1/zz_generated.deepcopy.go 



21 ■■■■■

config/crd/bases/policies.kubewarden.io_admissionpolicygroup... 

261	261	policies:
262	262	additionalProperties:
263	263	properties:
264	-	contextAwareResources:
265	-	description: -
266	-	List of Kubernetes resources the policy is allowed to access at evaluation time.

```

267 - Access to these resources
    is done using the `ServiceAccount` of the
    PolicyServer
268 - the policy is assigned to.
269 - items:
270 - description:
    ContextAwareResource identifies a Kubernetes
271 - resource.
272 - properties:
273 - apiVersion:
274 - description:
    apiVersion of the resource (v1 for core group,
275 -
    groupName/groupVersions for other).
276 - type: string
277 - kind:
278 - description: Singular
    PascalCase name of the resource
279 - type: string
280 - required:
281 - - apiVersion
282 - - kind
283 - type: object
284 - type: array
285 264 module:
286 265 description: |-
287 266 Module is the location of
    the WASM module to be loaded. Can be a

```

 200  docs/crds/CRD-docs-for-docs-repo.adoc 

```

253 253 [cols="20a,50a,15a,15a", options="header"]
254 254 |===
255 255 | Field | Description | Default | Validation
256 - | *`PolicyGroupSpec`* __xref:{anchor_prefix}-
    github-com-kubewarden-kubewarden-controller-api-
    policies-v1-policygroupspec[PolicyGroupSpec]-
    | | |
256 + | *`ClusterPolicyGroupSpec`* __xref:
    {anchor_prefix}-github-com-kubewarden-kubewarden-
    controller-api-policies-v1-
    clusterpolicygroupspec[ClusterPolicyGroupSpec]-
    - | | |
257 257 | *`namespaceSelector`*
    __link:https://kubernetes.io/docs/reference/generat
    ed/kubernetes-api/v1.31/#labelselector-v1-
    meta[LabelSelector]-
    | NamespaceSelector
    decides whether to run the webhook on an object
    based +
258 258 on whether the namespace for that object matches
    the selector. If the +
259 259 object itself is a namespace, the matching is
    performed on +

```

```

393 |===
394 |
395 |
396 | - [id="{anchor_prefix}-github-com-kubewarden-
    | kubewarden-controller-api-policies-v1-
    | contextawareresource"]
397 | - ==== ContextAwareResource
    | 396 + [id="{anchor_prefix}-github-com-kubewarden-
    | kubewarden-controller-api-policies-v1-
    | clusterpolicygroupspec"]
    | 397 + ==== ClusterPolicyGroupSpec
    | 398 +
398 | 399
399 | 400
400 | 401
401 | - ContextAwareResource identifies a Kubernetes
    | resource.
402 | 402
403 | 403
404 | 404
405 | 405 .Appears In:
406 | 406 ****
407 | - - xref:{anchor_prefix}-github-com-kubewarden-
    | kubewarden-controller-api-policies-v1-
    | clusteradmissionpolicyspec[$$ClusterAdmissionPolicy
    | Spec$$]
408 | - - xref:{anchor_prefix}-github-com-kubewarden-
    | kubewarden-controller-api-policies-v1-
    | policygroupmember[$$PolicyGroupMember$$]
    | 407 + - xref:{anchor_prefix}-github-com-kubewarden-
    | kubewarden-controller-api-policies-v1-
    | clusteradmissionpolicygroupspec[$$ClusterAdmissionP
    | olicyGroupSpec$$]
409 | 408 ****
410 | 409
411 | 410 [cols="20a,50a,15a,15a", options="header"]
412 | 411 |===
413 | 412 | Field | Description | Default | Validation
414 | - | *`apiVersion`* __string__ | apiVersion of the
    | resource (v1 for core group,
415 | - | *`kind`* __string__ | Singular PascalCase name of
    | the resource + | |
416 | - |===
417 | -
418 | -
419 | -
420 | -
421 | -
422 | -
423 | -
424 | -

```

```

425 -
426 -
427 -
428 -
413 + | *`GroupSpec`* __xref:{anchor_prefix}-github-com-
      kubewarden-kubewarden-controller-api-policies-v1-
      groupspec[{$$GroupSpec$$]__ | | |
414 + | *`policies`* __xref:{anchor_prefix}-github-com-
      kubewarden-kubewarden-controller-api-policies-v1-
      policygroupmemberswithcontext[{$$PolicyGroupMembersW
      ithContext$$]__ | Policies is a list of policies
      that are part of the group that will +
415 + be available to be called in the evaluation
      expression field. +
416 + Each policy in the group should be a Kubewarden
      policy. + | | Required: {} +
429 417
418 + |===
430 419
431 - [id="{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      policygroupmember"]
432 - ===== PolicyGroupMember
433 420
421 + [id="{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      contextawareresource"]
422 + ===== ContextAwareResource
434 423
435 424
436 425
426 + ContextAwareResource identifies a Kubernetes
      resource.
437 427
438 428
439 429
440 430 .Appears In:
441 431 ****
442 - - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      policygroupmembers[{$$PolicyGroupMembers$$]
432 + - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      clusteradmissionpolicyspec[{$$ClusterAdmissionPolicy
      Spec$$]
433 + - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      policygroupmemberwithcontext[{$$PolicyGroupMemberWit
      hContext$$]
443 434 ****
444 435
445 436 [cols="20a,50a,15a,15a", options="header"]

```

```

446 437 |===
447 438 | Field | Description | Default | Validation
448 - | *`module`* __string__ | Module is the location of
449 - the WASM module to be loaded. Can be a +
450 - local file (file://), a remote file served by an
451 - HTTP server +
452 - (http://, https://), or an artifact served by an
453 - OCI-compatible +
454 - registry (registry://). +
455 - If prefix is missing, it will default to
456 - registry:// and use that +
457 - internally. + | | Required: {} +
458 -
459 - | *`settings`*
460 - __link:https://kubernetes.io/docs/reference/generat
461 - ed/kubernetes-api/v1.31/#rawextension-runtime-
462 - pkg[ $RawExtension$ ]__ | Settings is a free-form
463 - object that contains the policy configuration +
464 - values. +
465 - x-kubernetes-embedded-resource: false + | |
466 - | *`contextAwareResources`* __xref:{anchor_prefix}-
467 - github-com-kubewarden-kubewarden-controller-api-
468 - policies-v1-
469 - contextawareresource[ $ContextAwareResource$ ]
470 - array__ | List of Kubernetes resources the policy
471 - is allowed to access at evaluation time. +
472 - Access to these resources is done using the
473 - `ServiceAccount` of the PolicyServer +
474 - the policy is assigned to. + | |
475 + | *`apiVersion`* __string__ | apiVersion of the
476 - resource (v1 for core group,
477 - groupName/groupVersions for other). + | |
478 + | *`kind`* __string__ | Singular PascalCase name of
479 - the resource + | |
480 |===
481 441 |===
482 442 |
483 443 |
484 - [id="{anchor_prefix}-github-com-kubewarden-
485 - kubewarden-controller-api-policies-v1-
486 - policygroupmembers"]
487 - ===== PolicyGroupMembers
488 -
489 - _Underlying type:_ __xref:{anchor_prefix}-github-
490 - com-kubewarden-kubewarden-controller-api-policies-
491 - v1-map-string-
492 - policygroupmember[ $map[string]PolicyGroupMember$ ]
493 -
494 -
495 -
496 -
497 -
498 -
499 -

```

```

473     - .Appears In:
474     - ****
475     - - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      policygroupspec[$$PolicyGroupSpec$$]
476     - ****
477     -
478     -
479     -
480     - [id="{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      policygroupspec"]
481     - ==== PolicyGroupSpec
482     444 + [id="{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-groups-
      spec"]
483     445 + ==== GroupSpec
484     446
485     447
486     448
487     449
488     450
489     451     .Appears In:
490     452     ****
491     - - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      admissionpolicygroupspec[$$AdmissionPolicyGroupSpec
      $$]
492     - - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      clusteradmissionpolicygroupspec[$$ClusterAdmissionP
      olicyGroupSpec$$]
493     455 + - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      clusterpolicygroupspec[$$ClusterPolicyGroupSpec$$]
494     456 + - xref:{anchor_prefix}-github-com-kubewarden-
      kubewarden-controller-api-policies-v1-
      policygroupspec[$$PolicyGroupSpec$$]
495     457     ****
496     458
497     459     [cols="20a,50a,15a,15a", options="header"]
498     548     the policy group is rejected. The specific policy
499     549     results will be +
500     550     returned in the warning field of the response. + |
501     551     | Required: {} +
502     552     + |===
503     553     +
504     554     +
505     555     +
506     556     +
507     557     +
508     558     +

```

```

559 +
560 +
561 +
562 +
563 +
564 +
565 +
566 + [id="{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmember"]
567 + ==== PolicyGroupMember
568 +
569 +
570 +
571 +
572 +
573 +
574 +
575 + .Appears In:
576 + ****
577 + - xref:{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmemberwithcontext[{$PolicyGroupMemberWithContext$}]
578 + - xref:{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmembers[{$PolicyGroupMembers$}]
579 + ****
580 +
581 + [cols="20a,50a,15a,15a", options="header"]
582 + |===
583 + | Field | Description | Default | Validation
584 + | *`module`* __string__ | Module is the location of
    the WASM module to be loaded. Can be a +
585 + local file (file://), a remote file served by an
    HTTP server +
586 + (http://, https://), or an artifact served by an
    OCI-compatible +
587 + registry (registry://). +
588 + If prefix is missing, it will default to
    registry:// and use that +
589 + internally. + | | Required: {} +
590 +
591 + | *`settings`*
    __link:https://kubernetes.io/docs/reference/generat
    ed/kubernetes-api/v1.31/#rawextension-runtime-
    pkg[{$RawExtension$}]__ | Settings is a free-form
    object that contains the policy configuration +
592 + values. +
593 + x-kubernetes-embedded-resource: false + | |
594 + |===
595 +

```

```

596 +
597 + [id="{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmemberwithcontext"]
598 + ==== PolicyGroupMemberWithContext
599 +
600 +
601 +
602 +
603 +
604 +
605 +
606 + .Appears In:
607 + ****
608 + - xref:{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmemberswithcontext[$$PolicyGroupMembersw
    ithContext$$]
609 + ****
610 +
611 + [cols="20a,50a,15a,15a", options="header"]
612 + |===
613 + | Field | Description | Default | Validation
614 + | *`PolicyGroupMember`* __xref:{anchor_prefix}-
    github-com-kubewarden-kubewarden-controller-api-
    policies-v1-
    policygroupmember[$$PolicyGroupMember$$]__ | | |
615 + | *`contextAwareResources`* __xref:{anchor_prefix}-
    github-com-kubewarden-kubewarden-controller-api-
    policies-v1-
    contextawareresource[$$ContextAwareResource$$]
    array__ | List of Kubernetes resources the policy
    is allowed to access at evaluation time. +
616 + Access to these resources is done using the
    `ServiceAccount` of the PolicyServer +
617 + the policy is assigned to. + | |
618 + |===
619 +
620 +
621 + [id="{anchor_prefix}-github-com-kubewarden-
    kubewarden-controller-api-policies-v1-
    policygroupmembers"]
622 + ==== PolicyGroupMembers
623 +
624 + _Underlying type:_ __xref:{anchor_prefix}-github-
    com-kubewarden-kubewarden-controller-api-policies-
    v1-map-string-
    policygroupmember[$$map[string]PolicyGroupMember$$]
    -
625 +
626 +
627 +

```

628 +
629 +
630 + **.Appears In:**
631 + ****
632 + - xref:{anchor_prefix}-github-com-kubewarden-
kubewarden-controller-api-policies-v1-
policygroupspec[\$\$PolicyGroupSpec\$\$]
633 + ****
634 +
635 +
636 +
637 + [id="{anchor_prefix}-github-com-kubewarden-
kubewarden-controller-api-policies-v1-
policygroupmemberswithcontext"]
638 + ==== PolicyGroupMembersWithContext
639 +
640 + _Underlying type:_ _xref:{anchor_prefix}-github-
com-kubewarden-kubewarden-controller-api-policies-
v1-map-string-
policygroupmemberwithcontext[\$\$map[string]PolicyGro
upMemberWithContext\$\$]_
641 +
642 +
643 +
644 +
645 +
646 + **.Appears In:**
647 + ****
648 + - xref:{anchor_prefix}-github-com-kubewarden-
kubewarden-controller-api-policies-v1-
clusterpolicygroupspec[\$\$ClusterPolicyGroupSpec\$\$]
649 + ****
650 +
651 +
652 +
653 + [id="{anchor_prefix}-github-com-kubewarden-
kubewarden-controller-api-policies-v1-
policygroupspec"]
654 + ==== PolicyGroupSpec
655 +
656 +
657 +
658 +
659 +
660 +
661 +
662 + **.Appears In:**
663 + ****
664 + - xref:{anchor_prefix}-github-com-kubewarden-
kubewarden-controller-api-policies-v1-
admissionpolicygroupspec[\$\$AdmissionPolicyGroupSpec
\$\$]

665	665	+ ****
666	666	+
667	667	+ [cols="20a,50a,15a,15a", options="header"]
668	668	+ ===
669	669	+ Field Description Default Validation
670	670	+ *`GroupSpec`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-groupspect[<u>\$\$GroupSpec\$\$</u>]__
587	671	*`policies`* __xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupmembers[<u>\$\$PolicyGroupMembers\$\$</u>]__
		Policies is a list of policies that are part of the group that will +
588	672	be available to be called in the evaluation expression field. +
589	673	Each policy in the group should be a Kubewarden policy. + Required: {} +
607	691	
608	692	.Appears In:
609	693	****
610	694	- - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policygroupspect[<u>\$\$PolicyGroupSpec\$\$</u>]
	694	+ - xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-groupspect[<u>\$\$GroupSpec\$\$</u>]
611	695	- xref:{anchor_prefix}-github-com-kubewarden-kubewarden-controller-api-policies-v1-policyspect[<u>\$\$PolicySpec\$\$</u>]
612	696	****
613	697	

95 docs/crds/CRD-docs-for-docs-repo.md

Load diff

Large diffs are not rendered by default.

26

internal/controller/policyserver_controller_configmap.go

22	22	
23	23	const dataType string = "Data" // only data type is supported
24	24	
25	25	- type <u>policyGroupMember</u> struct {
	25	+ type <u>policyGroupMemberWithContext</u> struct {
26	26	Module string
		`json:"module`

```

27 27         Settings runtime.RawExtension
    `json:"settings,omitempty"`
28 28         ContextAwareResources
    []policiesv1.ContextAwareResource
    `json:"contextAwareResources,omitempty"`
36 36         ContextAwareResources
    []policiesv1.ContextAwareResource
    `json:"contextAwareResources,omitempty"`
37 37         Settings runtime.RawExtension
    `json:"settings,omitempty"`
38 38         // The following fields are used by policy
    groups only.
39 -         Policies map[string]policyGroupMember
    `json:"policies,omitempty"`
40 -         Expression string
    `json:"expression,omitempty"`
41 -         Message string
    `json:"message,omitempty"`
39 +         Policies
    map[string]policyGroupMemberWithContext
    `json:"policies,omitempty"`
40 +         Expression string
    `json:"expression,omitempty"`
41 +         Message string
    `json:"message,omitempty"`
42 42     }
43 43
44 44     // The following MarshalJSON and UnmarshalJSON
    methods are used to serialize
64 64     func (p policyServerConfigEntry) MarshalJSON()
    ([]byte, error) {
65 65         if len(p.Policies) > 0 {
66 66             return json.Marshal(struct {
67 -                 NamespacedName
    types.NamespacedName
    `json:"namespacedName"`
68 -                 PolicyMode string
    `json:"policyMode"`
69 -                 Policies
    map[string]policyGroupMember `json:"policies"`
70 -                 Expression string
    `json:"expression"`
71 -                 Message string
    `json:"message"`
67 +                 NamespacedName
    types.NamespacedName
    `json:"namespacedName"`
68 +                 PolicyMode string
    `json:"policyMode"`
69 +                 Policies
    map[string]policyGroupMemberWithContext
    `json:"policies"`

```

```

70 +                                     Expression      string
    `json:"expression"`
71 +                                     Message         string
    `json:"message"`
72 72                                     }{
73 73                                     NamespacedName:
74 74                                     PolicyMode:
    p.NamespacedName,
    p.PolicyMode,
174 174         return
    unstructuredObj.GetResourceVersion(), nil
175 175     }
176 176
177 - func buildPolicyGroupMembers(policies
    policiesv1.PolicyGroupMembers)
    map[string]policyGroupMember {
178 -     policyGroupMembers :=
    map[string]policyGroupMember{}
177 + func buildPolicyGroupMembersWithContext(policies
    policiesv1.PolicyGroupMembersWithContext)
    map[string]policyGroupMemberWithContext {
178 +     policyGroupMembers :=
    map[string]policyGroupMemberWithContext{}
179 179         for name, policy := range policies {
180 -             policyGroupMembers[name] =
    policyGroupMember{
180 +             policyGroupMembers[name] =
    policyGroupMemberWithContext{
181 181                                     Module:
    policy.Module,
182 182                                     Settings:
    policy.Settings,
183 183                                     ContextAwareResources:
    policy.ContextAwareResources,
202 202                                     }
203 203
204 204         if policyGroup, ok :=
    admissionPolicy.(policiesv1.PolicyGroup); ok {
205 -             configEntry.Policies =
    buildPolicyGroupMembers(policyGroup.GetPolicyGroup
    Members())
205 +             configEntry.Policies =
    buildPolicyGroupMembersWithContext(policyGroup.Get
    PolicyGroupMembersWithContext())
206 206             configEntry.Expression =
    policyGroup.GetExpression()
207 207             configEntry.Message =
    policyGroup.GetMessage()
208 208         }

```



 20 ■■■■

internal/controller/policyserver_controller_test.go 

```

276 276
277 277         clusterPolicyGroup :=
        policiesv1.NewClusterAdmissionPolicyGroupFactory()
        .
278 278
        WithPolicyServer(policyServerName).
279 -
        WithMembers(policiesv1.PolicyGroupMembers{
279 +
        WithMembers(policiesv1.PolicyGroupMembersWithConte
        xt{
280 280
        "pod_privileged": {
281 -
        Module: "registry://ghcr.io/kubewarden/tests/pod-
        privileged:v0.2.5",
281 +
        PolicyGroupMember: policiesv1.PolicyGroupMember{
282 +
        Module: "registry://ghcr.io/kubewarden/tests/pod-
        privileged:v0.2.5",
283 +
        },
282 284
        ContextAwareResources:
        []policiesv1.ContextAwareResource{
283 285
        {
284 286
        APIVersion: "v1",
287 289
        },
288 290
        },
289 291
        "user_group_psp": {
290 -
        Module: "registry://ghcr.io/kubewarden/tests/user-
        group-psp:v0.4.9",
292 +
        PolicyGroupMember: policiesv1.PolicyGroupMember{
293 +
        Module: "registry://ghcr.io/kubewarden/tests/user-
        group-psp:v0.4.9",
294 +
        },
291 295
        ContextAwareResources:
        []policiesv1.ContextAwareResource{
292 296
        {
293 297
        APIVersion: "v1",
332 336
        AllowedToMutate:
        admissionPolicyGroup.IsMutating(),

```

```

333 337      Settings:
334 338      admissionPolicyGroup.GetSettings(),
335 339      ContextAwareResources:
336 340      admissionPolicyGroup.GetContextAwareResources(),
337 341      Policies:
338 342      - buildPolicyGroupMembers(admissionPolicyGroup.GetPo
339 343      licyGroupMembers()),
340 344      + buildPolicyGroupMembersWithContext(admissionPolicy
341 345      Group.GetPolicyGroupMembersWithContext()),
342 346      Expression:
343 347      admissionPolicyGroup.GetExpression(),
344 348      Message:
345 349      admissionPolicyGroup.GetMessage(),
346 350      }
347 351      Settings:
348 352      clusterPolicyGroup.GetSettings(),
349 353      ContextAwareResources:
350 354      clusterPolicyGroup.GetContextAwareResources(),
351 355      PolicyMode:
352 356      string(clusterPolicyGroup.GetPolicyMode()),
353 357      Policies:
354 358      - buildPolicyGroupMembers(clusterPolicyGroup.GetPoli
355 359      cyGroupMembers()),
356 360      + buildPolicyGroupMembersWithContext(clusterPolicyGr
357 361      oup.GetPolicyGroupMembersWithContext()),
358 362      Expression:
359 363      clusterPolicyGroup.GetExpression(),
360 364      Message:
361 365      clusterPolicyGroup.GetMessage(),
362 366      }
363 367      }
364 368      }
365 369      }
366 370      }
367 371      }
368 372      }
369 373      }
370 374      }
371 375      }
372 376      }
373 377      }
374 378      }
375 379      }
376 380      }
377 381      }
378 382      }
379 383      }
380 384      }
381 385      }
382 386      }
383 387      }
384 388      }
385 389      }
386 390      }
387 391      }
388 392      }
389 393      }
390 394      }
391 395      }
392 396      }
393 397      }
394 398      }
395 399      }
396 400      }
397 401      }
398 402      }
399 403      }
400 404      }
401 405      }
402 406      }
403 407      }
404 408      }
405 409      }
406 410      }
407 411      }
408 412      }
409 413      }
410 414      }
411 415      }
412 416      }
413 417      }
414 418      }
415 419      }
416 420      }
417 421      }
418 422      }
419 423      }
420 424      }
421 425      }
422 426      }
423 427      }
424 428      }
425 429      }
426 430      }
427 431      }
428 432      }
429 433      }
430 434      }
431 435      }
432 436      }
433 437      }
434 438      }
435 439      }
436 440      }
437 441      }
438 442      }
439 443      }
440 444      }
441 445      }
442 446      }
443 447      }
444 448      }
445 449      }
446 450      }
447 451      }
448 452      }
449 453      }
450 454      }
451 455      }
452 456      }
453 457      }
454 458      }
455 459      }
456 460      }
457 461      }
458 462      }
459 463      }
460 464      }
461 465      }
462 466      }
463 467      }
464 468      }
465 469      }
466 470      }
467 471      }
468 472      }
469 473      }
470 474      }
471 475      }
472 476      }
473 477      }
474 478      }
475 479      }
476 480      }
477 481      }
478 482      }
479 483      }
480 484      }
481 485      }
482 486      }
483 487      }
484 488      }
485 489      }
486 490      }
487 491      }
488 492      }
489 493      }
490 494      }
491 495      }
492 496      }
493 497      }
494 498      }
495 499      }
496 500      }
497 501      }
498 502      }
499 503      }
500 504      }
501 505      }
502 506      }
503 507      }
504 508      }
505 509      }
506 510      }
507 511      }
508 512      }
509 513      }
510 514      }
511 515      }
512 516      }
513 517      }
514 518      }
515 519      }
516 520      }
517 521      }
518 522      }
519 523      }
520 524      }
521 525      }
522 526      }
523 527      }
524 528      }
525 529      }
526 530      }
527 531      }
528 532      }
529 533      }
530 534      }
531 535      }
532 536      }
533 537      }
534 538      }
535 539      }
536 540      }
537 541      }
538 542      }
539 543      }
540 544      }
541 545      }
542 546      }
543 547      }
544 548      }
545 549      }
546 550      }
547 551      }
548 552      }
549 553      }
550 554      }
551 555      }
552 556      }
553 557      }
554 558      }
555 559      }
556 560      }
557 561      }
558 562      }
559 563      }
560 564      }
561 565      }
562 566      }
563 567      }
564 568      }
565 569      }
566 570      }
567 571      }
568 572      }
569 573      }
570 574      }
571 575      }
572 576      }
573 577      }
574 578      }
575 579      }
576 580      }
577 581      }
578 582      }
579 583      }
580 584      }
581 585      }
582 586      }
583 587      }
584 588      }
585 589      }
586 590      }
587 591      }
588 592      }
589 593      }
590 594      }
591 595      }
592 596      }
593 597      }
594 598      }
595 599      }
596 600      }
597 601      }
598 602      }
599 603      }
600 604      }
601 605      }
602 606      }
603 607      }
604 608      }
605 609      }
606 610      }
607 611      }
608 612      }
609 613      }
610 614      }
611 615      }
612 616      }
613 617      }
614 618      }
615 619      }
616 620      }
617 621      }
618 622      }
619 623      }
620 624      }
621 625      }
622 626      }
623 627      }
624 628      }
625 629      }
626 630      }
627 631      }
628 632      }
629 633      }
630 634      }
631 635      }
632 636      }
633 637      }
634 638      }
635 639      }
636 640      }
637 641      }
638 642      }
639 643      }
640 644      }
641 645      }
642 646      }
643 647      }
644 648      }
645 649      }
646 650      }
647 651      }
648 652      }
649 653      }
650 654      }
651 655      }
652 656      }
653 657      }
654 658      }
655 659      }
656 660      }
657 661      }
658 662      }
659 663      }
660 664      }
661 665      }
662 666      }
663 667      }
664 668      }
665 669      }
666 670      }
667 671      }
668 672      }
669 673      }
670 674      }
671 675      }
672 676      }
673 677      }
674 678      }
675 679      }
676 680      }
677 681      }
678 682      }
679 683      }
680 684      }
681 685      }
682 686      }
683 687      }
684 688      }
685 689      }
686 690      }
687 691      }
688 692      }
689 693      }
690 694      }
691 695      }
692 696      }
693 697      }
694 698      }
695 699      }
696 700      }
697 701      }
698 702      }
699 703      }
700 704      }
701 705      }
702 706      }
703 707      }
704 708      }
705 709      }
706 710      }
707 711      }
708 712      }
709 713      }
710 714      }
711 715      }
712 716      }
713 717      }
714 718      }
715 719      }
716 720      }
717 721      }
718 722      }
719 723      }
720 724      }
721 725      }
722 726      }
723 727      }
724 728      }
725 729      }
726 730      }
727 731      }
728 732      }
729 733      }
730 734      }
731 735      }
732 736      }
733 737      }
734 738      }
735 739      }
736 740      }
737 741      }
738 742      }
739 743      }
740 744      }
741 745      }
742 746      }
743 747      }
744 748      }
745 749      }
746 750      }
747 751      }
748 752      }
749 753      }
750 754      }
751 755      }
752 756      }
753 757      }
754 758      }
755 759      }
756 760      }
757 761      }
758 762      }
759 763      }
760 764      }
761 765      }
762 766      }
763 767      }
764 768      }
765 769      }
766 770      }
767 771      }
768 772      }
769 773      }
770 774      }
771 775      }
772 776      }
773 777      }
774 778      }
775 779      }
776 780      }
777 781      }
778 782      }
779 783      }
780 784      }
781 785      }
782 786      }
783 787      }
784 788      }
785 789      }
786 790      }
787 791      }
788 792      }
789 793      }
790 794      }
791 795      }
792 796      }
793 797      }
794 798      }
795 799      }
796 800      }
797 801      }
798 802      }
799 803      }
800 804      }
801 805      }
802 806      }
803 807      }
804 808      }
805 809      }
806 810      }
807 811      }
808 812      }
809 813      }
810 814      }
811 815      }
812 816      }
813 817      }
814 818      }
815 819      }
816 820      }
817 821      }
818 822      }
819 823      }
820 824      }
821 825      }
822 826      }
823 827      }
824 828      }
825 829      }
826 830      }
827 831      }
828 832      }
829 833      }
830 834      }
831 835      }
832 836      }
833 837      }
834 838      }
835 839      }
836 840      }
837 841      }
838 842      }
839 843      }
840 844      }
841 845      }
842 846      }
843 847      }
844 848      }
845 849      }
846 850      }
847 851      }
848 852      }
849 853      }
850 854      }
851 855      }
852 856      }
853 857      }
854 858      }
855 859      }
856 860      }
857 861      }
858 862      }
859 863      }
860 864      }
861 865      }
862 866      }
863 867      }
864 868      }
865 869      }
866 870      }
867 871      }
868 872      }
869 873      }
870 874      }
871 875      }
872 876      }
873 877      }
874 878      }
875 879      }
876 880      }
877 881      }
878 882      }
879 883      }
880 884      }
881 885      }
882 886      }
883 887      }
884 888      }
885 889      }
886 890      }
887 891      }
888 892      }
889 893      }
890 894      }
891 895      }
892 896      }
893 897      }
894 898      }
895 899      }
896 900      }
897 901      }
898 902      }
899 903      }
900 904      }
901 905      }
902 906      }
903 907      }
904 908      }
905 909      }
906 910      }
907 911      }
908 912      }
909 913      }
910 914      }
911 915      }
912 916      }
913 917      }
914 918      }
915 919      }
916 920      }
917 921      }
918 922      }
919 923      }
920 924      }
921 925      }
922 926      }
923 927      }
924 928      }
925 929      }
926 930      }
927 931      }
928 932      }
929 933      }
930 934      }
931 935      }
932 936      }
933 937      }
934 938      }
935 939      }
936 940      }
937 941      }
938 942      }
939 943      }
940 944      }
941 945      }
942 946      }
943 947      }
944 948      }
945 949      }
946 950      }
947 951      }
948 952      }
949 953      }
950 954      }
951 955      }
952 956      }
953 957      }
954 958      }
955 959      }
956 960      }
957 961      }
958 962      }
959 963      }
960 964      }
961 965      }
962 966      }
963 967      }
964 968      }
965 969      }
966 970      }
967 971      }
968 972      }
969 973      }
970 974      }
971 975      }
972 976      }
973 977      }
974 978      }
975 979      }
976 980      }
977 981      }
978 982      }
979 983      }
980 984      }
981 985      }
982 986      }
983 987      }
984 988      }
985 989      }
986 990      }
987 991      }
988 992      }
989 993      }
990 994      }
991 995      }
992 996      }
993 997      }
994 998      }
995 999      }
996 1000      }
997 1001      }
998 1002      }
999 1003      }
1000 1004      }
1001 1005      }
1002 1006      }
1003 1007      }
1004 1008      }
1005 1009      }
1006 1010      }
1007 1011      }
1008 1012      }
1009 1013      }
1010 1014      }
1011 1015      }
1012 1016      }
1013 1017      }
1014 1018      }
1015 1019      }
1016 1020      }
1017 1021      }
1018 1022      }
1019 1023      }
1020 1024      }
1021 1025      }
1022 1026      }
1023 1027      }
1024 1028      }
1025 1029      }
1026 1030      }
1027 1031      }
1028 1032      }
1029 1033      }
1030 1034      }
1031 1035      }
1032 1036      }
1033 1037      }
1034 1038      }
1035 1039      }
1036 1040      }
1037 1041      }
1038 1042      }
1039 1043      }
1040 1044      }
1041 1045      }
1042 1046      }
1043 1047      }
1044 1048      }
1045 1049      }
1046 1050      }
1047 1051      }
1048 1052      }
1049 1053      }
1050 1054      }
1051 1055      }
1052 1056      }
1053 1057      }
1054 1058      }
1055 1059      }
1056 1060      }
1057 1061      }
1058 1062      }
1059 1063      }
1060 1064      }
1061 1065      }
1062 1066      }
1063 1067      }
1064 1068      }
1065 1069      }
1066 1070      }
1067 1071      }
1068 1072      }
1069 1073      }
1070 1074      }
1071 1075      }
1072 1076      }
1073 1077      }
1074 1078      }
1075 1079      }
1076 1080      }
1077 1081      }
1078 1082      }
1079 1083      }
1080 1084      }
1081 1085      }
1082 1086      }
1083 1087      }
1084 1088      }
1085 1089      }
1086 1090      }
1087 1091      }
1088 1092      }
1089 1093      }
1090 1094      }
1091 1095      }
1092 1096      }
1093 1097      }
1094 1098      }
1095 1099      }
1096 1100      }
1097 1101      }
1098 1102      }
1099 1103      }
1100 1104      }
1101 1105      }
1102 1106      }
1103 1107      }
1104 1108      }
1105 1109      }
1106 1110      }
1107 1111      }
1108 1112      }
1109 1113      }
1110 1114      }
1111 1115      }
1112 1116      }
1113 1117      }
1114 1118      }
1115 1119      }
1116 1120      }
1117 1121      }
1118 1122      }
1119 1123      }
1120 1124      }
1121 1125      }
1122 1126      }
1123 1127      }
1124 1128      }
1125 1129      }
1126 1130      }
1127 1131      }
1128 1132      }
1129 1133      }
1130 1134      }
1131 1135      }
1132 1136      }
1133 1137      }
1134 1138      }
1135 1139      }
1136 1140      }
1137 1141      }
1138 1142      }
1139 1143      }
1140 1144      }
1141 1145      }
1142 1146      }
1143 1147      }
1144 1148      }
1145 1149      }
1146 1150      }
1147 1151      }
1148 1152      }
1149 1153      }
1150 1154      }
1151 1155      }
1152 1156      }
1153 1157      }
1154 1158      }
1155 1159      }
1156 1160      }
1157 1161      }
1158 1162      }
1159 1163      }
1160 1164      }
1161 1165      }
1162 1166      }
1163 1167      }
1164 1168      }
1165 1169      }
1166 1170      }
1167 1171      }
1168 1172      }
1169 1173      }
1170 1174      }
1171 1175      }
1172 1176      }
1173 1177      }
1174 1178      }
1175 1179      }
1176 1180      }
1177 1181      }
1178 1182      }
1179 1183      }
1180 1184      }
1181 1185      }
1182 1186      }
1183 1187      }
1184 1188      }
1185 1189      }
1186 1190      }
1187 1191      }
1188 1192      }
1189 1193      }
1190 1194      }
1191 1195      }
1192 1196      }
1193 1197      }
1194 1198      }
1195 1199      }
1196 1200      }
1197 1201      }
1198 1202      }
1199 1203      }
1200 1204      }
1201 1205      }
1202 1206      }
1203 1207      }
1204 1208      }
1205 1209      }
1206 1210      }
1207 1211      }
1208 1212      }
1209 1213      }
1210 1214      }
1211 1215      }
1212 1216      }
1213 1217      }
1214 1218      }
1215 1219      }
1216 1220      }
1217 1221      }
1218 1222      }
1219 1223      }
1220 1224      }
1221 1225      }
1222 1226      }
1223 1227      }
1224 1228      }
1225 1229      }
1226 1230      }
1227 1231      }
1228 1232      }
1229 1233      }
1230 1234      }
1231 1235      }
1232 1236      }
1233 1237      }
1234 1238      }
1235 1239      }
1236 1240      }
1237 1241      }
1238 1242      }
1239 1243      }
1240 1244      }
1241 1245      }
1242 1246      }
1243 1247      }
1244 1248      }
1245 1249      }
1246 1250      }
1247 1251      }
1248 1252      }
1249 1253      }
1250 1254      }
1251 1255      }
1252 1256      }
1253 1257      }
1254 1258      }
1255 1259      }
1256 1260      }
1257 1261      }
1258 1262      }
1259 1263      }
1260 1264      }
1261 1265      }
1262 1266      }
1263 1267      }
1264 1268      }
1265 1269      }
1266 1270      }
1267 1271      }
1268 1272      }
1269 1273      }
1270 1274      }
1271 1275      }
1272 1276      }
1273 1277      }
1274 1278      }
1275 1279      }
1276 1280      }
1277 1281      }
1278 1282      }
1279 1283      }
1280 1284      }
1281 1285      }
1282 1286      }
1283 1287      }
1284 1288      }
1285 1289      }
1286 1290      }
1287 1291      }
1288 1292      }
1289 1293      }
1290 1294      }
1291 1295      }
1292 1296      }
1293 1297      }
1294 1298      }
1295 1299      }
1296 1300      }
1297 1301      }
1298 1302      }
1299 1303      }
1300 1304      }
1301 1305      }
1302 1306      }
1303 1307      }
1304 1308      }
1305 1309      }
1306 1310      }
1307 1311      }
1308 1312      }
1309 1313      }
1310 1314      }
1311 1315      }
1312 1316      }
1313 1317      }
1314 1318      }
1315 1319      }
1316 1320      }
1317 1321      }
1318 1322      }
1319 1323      }
1320 1324      }
1321 1325      }
1322 1326      }
1323 1327      }
1324 1328      }
1325 1329      }
1326 1330      }
1327 1331      }
1328 1332      }
1329 1333      }
1330 1334      }
1331 1335      }
1332 1336      }
1333 1337      }
1334 1338      }
1335 1339      }
1336 1340      }
1337 1341      }
1338 1342      }
1339 1343      }
1340 1344      }
1341 1345      }
1342 1346      }
1343 1347      }
1344 1348      }
1345 1349      }
1346 1350      }
1347 1351      }
1348 1352      }
1349 1353      }
1350 1354      }
1351 1355      }
1352 1356      }
1353 1357      }
1354 1358      }
1355 1359      }
1356 1360      }
1357 1361      }
1358 1362      }
1359 1363      }
1360 1364      }
1361 1365      }
1362 1366      }
1363 1367      }
1364 1368      }
1365 1369      }
1366 1370      }
1367 1371      }
1368 1372      }
1369 1373      }
1370 1374      }
1371 1375      }
1372 1376      }
1373 1377      }
1374 1378      }
1375 1379      }
1376 1380      }
1377 1381      }
1378 1382      }
1379 1383      }
1380 1384      }
1381 1385      }
1382 1386      }
1383 1387      }
1384 1388      }
1385 1389      }
1386 1390      }
1387 1391      }
1388 1392      }
1389 1393      }
1390 1394      }
1391 1395      }
1392 1396      }
1393 1397      }
1394 1398      }
1395 1399      }
1396 1400      }
1397 1401      }
1398 1402      }
1399 1403      }
1400 1404      }
1401 1405      }
1402 1406      }
1403 1407      }
1404 1408      }
1405 1409      }
1406 1410      }
1407 1411      }
1408 1412      }
1409 1413      }
1410 1414      }
1411 1415      }
1412 1416      }
1413 1417      }
1414 1418      }
1415 1419      }
1416 1420      }
1417 1421      }
1418 1422      }
1419 1423      }
1420 1424      }
1421 1425      }
1422 1426      }
1423 1427      }
1424 1428      }
1425 1429      }
1426 1430      }
1427 1431      }
1428 1432      }
1429 1433      }
1430 1434      }
1431 1435      }
1432 1436      }
1433 1437      }
1434 1438      }
1435 1439      }
1436 1440      }
1437 1441      }
1438 1442      }
1439 1443      }
1440 1444      }
1441 1445      }
1442 1446      }
1443 1447      }
1444 1448      }
1445 1449      }
1446 1450      }
1447 1451      }
1448 1452      }
1449 1453      }
1450 1454      }
1451 1455      }
1452 1456      }
1453 1457      }
1454 1458      }
1455 1459      }
1456 1460      }
1457 1461      }
1458 146
```

412	416	"policyMode":
		Equal(string(admissionPolicyGroup.GetPolicyMode())
),
420	424	}),
421	425	"policies": MatchKeys(IgnoreExtras, Keys{
422	426	"pod_privileged": MatchAllKeys(Keys{
423	-	"module":
		Equal(clusterPolicyGroup.GetPolicyGroupMembers()
		["pod_privileged"].Module),
	427	+ "module":
		Equal(clusterPolicyGroup.GetPolicyGroupMembersWith
		Context()["pod_privileged"].Module),
424	428	"settings": Ignore(),
425	429	"contextAwareResources":
		And(ContainElement(MatchAllKeys(Keys{
426	430	"apiVersion": Equal("v1"),
427	431	"kind": Equal("Pod"),
428	432))), HaveLen(1)),
429	433	}),
430	434	"user_group_psp": MatchAllKeys(Keys{
431	-	"module":
		Equal(clusterPolicyGroup.GetPolicyGroupMembers()
		["user_group_psp"].Module),
	435	+ "module":
		Equal(clusterPolicyGroup.GetPolicyGroupMembersWith
		Context()["user_group_psp"].Module),
432	436	"settings": Ignore(),
433	437	"contextAwareResources":
		And(ContainElement(MatchAllKeys(Keys{
434	438	"apiVersion": Equal("v1"),

0 comments on commit `51a88df`

Please [sign in](#) to comment.