





# CEL PLAYGROUND

**CEL PLAYGROUND** | Quickly test Common Expression Language

[Share](#) | [Visit our GitHub](#)

**Expression (CEL format)** Examples ▾

```
1 // Welcome to the CEL Playground!
2 // CEL Playground is an interactive WebAssembly powered environment
3 //
4 // - Write your CEL expression here
5 // - Use the area on the side for input data, in YAML or JSON format
6 // - Press 'Run' to evaluate your CEL expression against the input data
7 // - Explore our collection of examples for inspiration
8
9 account.balance >= transaction.withdrawal
10 || (account.overdraftProtection
11    && account.overdraftLimit >= transaction.withdrawal - account.b
12
```

**Input (YAML or JSON format)** Run

```
1 # Here is the input data in YAML or JSON format.
2
3 - account:
4   balance: 500
5   overdraftProtection: true
6 overdraftLimit: 1000
7 - transaction:
8   withdrawal: 700
9
```

**Output**

```
true
```

Powered by [Undistro](#) cel-go v0.16.0

**Kevin Conner, Director of Engineering**  
[kev.conner@getupcloud.com](mailto:kev.conner@getupcloud.com)

**GETUP**

# What is CEL Playground?

CEL Playground is an interactive WebAssembly (Wasm) powered environment to explore and experiment with the [Common Expression Language \(CEL\)](#) providing a simple and user-friendly interface to write and quickly evaluate CEL expressions for use in Kubernetes, Istio and other Cloud Native technologies.



Easy Sharing



Learning



Quick Testing



Avoid Mistakes

GETUP





CEL PLAYGROUND


**GETUP**

The original idea was to provide a simple browser-based testing environment for Common Expression Language (CEL), the Playground enable and enhance the user experience and exploration of CEL.

### # sig-api-machinery-cel-dev

 **Matheus Moraes** 11:03 AM  
Hello everyone,  
Does anyone working with `ValidatingAdmissionPolicy` use a playground for CEL or something similar?  
I'm working on `Marvin`, a cluster scanner that also uses CEL, and I think a playground would help a lot in testing CEL expressions.  
Something similar to `rego playground` and `CUE playground`

 **jpbetz** 4 months ago  
This has been suggested a few times. I think it is very much needed, but I'm not aware of anyone building one yet.

 **jpbetz** 4 months ago  
It might be possible to build it as a static site using WASM if anyone is interested in building it but concerned about hosting costs.

2

SIG API machinery Agenda & Notes... ☆ 📁 ☁ ⌚

File Edit View Insert Format Tools Extensions ...

🔍 ↶ ↷ 🖨 ⌨ 100% ▾ | Heading 3 ▾ ⋮

1 2 3 4 5 6

- Downside
  - Maintain effort
  - Immature itself(missing tooling)
- Tooling to experience with CEL in K8s (kubectl get, dry run, etc.)
  - Command line tool ([repo](#))
  - A cel playground would be a great idea (not client validation) (could be kubectl get format)

GETUP



CEL PLAYGROUND

GETUP

# Common Expression Language

- Developed by Google
  - Simple, portable expressions
  - Embeddable
  - Fast evaluation
  - Typed
  - Safe
    - No unbounded loops
  - Extensible

GETUP



CEL PLAYGROUND

**GETUP**

# Supported Modes

## CEL Expressions

The screenshot displays the CEL Playground web application. At the top, there's a header with the 'CEL PLAYGROUND' logo, a 'CEL Expression' tab, a 'Modes' dropdown, a 'Share' button, a star icon, and a 'Visit our GitHub' link. The main interface is divided into three sections: a 'CEL Expression' editor on the left, an 'Input' section in the top right, and an 'Output' section in the bottom right. The 'CEL Expression' editor contains a code block with line numbers 1 through 12, featuring a welcome message and a CEL expression: `account.balance >= transaction.withdrawal || (account.overdraftProtection && account.overdraftLimit >= transaction.withdrawal - account.b`. The 'Input' section has a 'Run' button and contains a code block with line numbers 1 through 9, showing input data in YAML/JSON format: `# Here is the input data in YAML or JSON format.  
account:  
 balance: 500  
 overdraftProtection: true  
 overdraftLimit: 1000  
transaction:  
 withdrawal: 700`. The 'Output' section shows the result 'true' and a 'Cost: 15' indicator. At the bottom, it says 'Powered by Getup' and provides links for 'Language Definition' and 'cel-go v0.17.8'.

CEL PLAYGROUND CEL Expression Modes Share Visit our GitHub

CEL Expression Examples

```
1 // Welcome to the CEL Playground!  
2 // CEL Playground is an interactive WebAssembly powered environment  
3 //  
4 // - Write your CEL expression here  
5 // - Use the area on the side for input data, in YAML or JSON format  
6 // - Press 'Run' to evaluate your CEL expression against the input data  
7 // - Explore our collection of examples for inspiration  
8  
9 account.balance >= transaction.withdrawal  
10 || (account.overdraftProtection  
11    && account.overdraftLimit >= transaction.withdrawal - account.b  
12
```

Input Run

```
1 # Here is the input data in YAML or JSON format.  
2  
3  
4 account:  
5   balance: 500  
6   overdraftProtection: true  
7   overdraftLimit: 1000  
8 transaction:  
9   withdrawal: 700
```

Output Cost: 15

```
true
```

Powered by Getup Language Definition cel-go v0.17.8

GETUP



CEL PLAYGROUND

GETUP

# Supported Modes

## Validating Admission Policy

CEL PLAYGROUND Validating Admission Policy Modes

Share \* Visit our GitHub

### Validating Admission Policy

Examples

```
1 apiVersion: admissionregistration.k8s.io/v1alpha1
2 kind: ValidatingAdmissionPolicy
3 metadata:
4   name: "force-ha-in-prod"
5 spec:
6   failurePolicy: Fail
7   matchConstraints:
8     resourceRules:
9     - apiGroups: ["apps"]
10       apiVersions: ["v1"]
11       operations: ["CREATE", "UPDATE"]
12       resources: ["deployments"]
13   validations:
14     - expression: "object.spec.replicas >= 3"
15       message: "All production deployments should be HA with at least three
16
```

Inputs: Object Old Object... Namespaces... Request Authorization... Run

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   labels:
5     app: "kubernetes-bootcamp"
6   name: "kubernetes-bootcamp"
7   namespace: "default"
8 spec:
9   progressDeadlineSeconds: 600
10  replicas: 3
11  revisionHistoryLimit: 10
12  selector:
13    matchLabels:
14      app: "kubernetes-bootcamp"
15  strategy:
16    rollingUpdate:
17      maxSurge: 25%
18      maxUnavailable: 25%
```

Output Total cost: 4

validations[0] Cost: 4

true

Powered by [Getup](#) Language Definition | cel-go v0.17.8

GETUP



CEL PLAYGROUND

GETUP

# Supported Modes

## Webhooks

CEL PLAYGROUND Web Hooks Modes [Share](#) [Visit our GitHub](#)

Web Hooks [Request Ignore Leases](#)

```
1 apiVersion: admissionregistration.k8s.io/v1
2 kind: ValidatingWebhookConfiguration
3 webhooks:
4   - name: my-webhook.example.com
5     matchPolicy: Equivalent
6     rules:
7       - operations: ['CREATE', 'UPDATE']
8         apiGroups: ['*']
9         apiVersions: ['*']
10        resources: ['*']
11      failurePolicy: 'Ignore' # Fail-open (optional)
12      sideEffects: None
13      clientConfig:
14        service:
15          namespace: my-namespace
16          name: my-webhook
17          caBundle: 'QGVhbnVhZGxPgo='
18      # You can have up to 64 matchConditions per webhook
19      matchConditions:
20        - name: 'exclude-leases' # Each match condition must have a unique name
21          expression: '!request.resource.group == "coordination.k8s.io"'
22        - name: 'exclude-kubelet-requests'
23          expression: '!request.resource.resource == "leases"' # Match non-lease
24        - name: 'exclude-kubelet-requests'
25          expression: '!request.userInfo.groups in request.userInfo.groups' # Match non-lease
```

Inputs: [Object](#) [Old Object](#) [Request](#) [Authorization](#) [Run](#)

```
1 apiVersion: coordination.k8s.io/v1
2 kind: Lease
3 metadata:
4   name: ingress-nginx-leader
5   namespace: ingress-nginx
6 spec:
7   acquireTime: "2023-11-24T16:51:02.229818Z"
8   holderIdentity: ingress-nginx-controller-6597456577-s5h9w
9   leaseDurationSeconds: 30
10  leaseTransitions: 7
11  renewTime: "2024-04-09T21:59:30.694589Z"
```

Output [Total cost: 15](#)

webhookMatchConditions.exclude-leases	Cost: 10
false	
webhookMatchConditions.exclude-kubelet-requests	Cost: 5
true	

Powered by [Getup](#) [Language Definition](#) | cel-go v0.17.8

GETUP



CEL PLAYGROUND

GETUP



# Demo

[playcel.undistro.io](https://playcel.undistro.io)

GETUP



CEL PLAYGROUND

**GETUP**

# CEL Playground RoadMap

On the Road to [CNCF sandbox!](#)

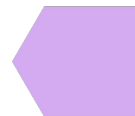
Better support for Kubernetes use cases



Authentication Configuration Claim Mapping



Authorization Configuration



Custom Resource Validation



Structured Editors

**Supporting other  
CNCF & OSS projects**

Istio, Envoy  
Others?

**Testing on live clusters**



CEL PLAYGROUND

**GETUP**

# KubeCon 2024

## CEL-Ebrating Simplicity: Mastering Kubernetes Policy Enforcement - Kevin Conner, Getup Cloud & Anish Ramasekar, Microsoft

📅 Wednesday November 13, 2024 3:25pm - 4:00pm MST

📍 151

As Kubernetes deployments grow increasingly complex, robust policy enforcement is crucial. The Common Expression Language (CEL) provides a powerful solution, enabling the creation of sophisticated, human-readable expressions for Kubernetes policies. This session explores CEL's integration with Kubernetes, simplifying policy definition and enforcement. Key takeaways: - Fundamentals of CEL and its Kubernetes integration. - Practical use cases for CEL in admission control, resource management, and security. - Enhancing policy expressiveness and flexibility with CEL. - Introduction to CEL Playground for testing and validating CEL expressions. Through live demos, learn to leverage CEL and CEL Playground for streamlined policy management in Kubernetes. Ideal for administrators, developers, and DevOps professionals, this session equips you to enhance your Kubernetes policies using CEL. Join us to discover how CEL and CEL Playground can transform your Kubernetes policy management.

### Speakers



#### Anish Ramasekar

Senior Software Engineer, Microsoft

Anish Ramasekar is a software engineer at Microsoft. He is on the Azure Container Upstream team building features for Kubernetes upstream and various CNCF projects that are part of the Azure Kubernetes Service. Anish is a maintainer of the Secrets Store CSI Driver project.



#### Kevin Conner

Chief Engineer, Getup Cloud

Kevin Conner is the Chief Engineer at GetUp Cloud, a startup focused on Kubernetes and DevSecOps. He has worked at startups like Integrated Micro Products, Arjuna Technologies, JBoss, and Aviatix, as well as Sun Microsystems and Red Hat where he led teams for Cloud Enablement, Service... [Read More →](#)

● Security

Content Experience Level [Intermediate](#)

GETUP



CEL PLAYGROUND

GETUP

# Are you already using CEL Playground?

**Support our community efforts!**

Add your organization to our Adopters.md file here:

<https://github.com/undistro/cel-playground/blob/main/ADOPTERS.md>



# Contributions are welcome!

- ▶ **CEL Playground:** [playcel.undistro.io](https://playcel.undistro.io)
- ▶ **GitHub Repository:** [github.com/undistro/cel-playground](https://github.com/undistro/cel-playground)
- ▶ **Slack:** [#cel-playground](#)

## CONTACT US

**Kevin Conner:** [kev.conner@getupcloud.com](mailto:kev.conner@getupcloud.com)

GETUP



CEL PLAYGROUND

GETUP



GETUP

THANK YOU!

GETUP.IO