

a tool to register Kubernetes PODs as Consul Services

#kube-consul-register #registered-services

70 commits

1 branch

6 releases

10 contributors

Apache-2.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

tczekajlo Merge pull request #43 from aswinkarthik/use-go-modules ...

Latest commit 77e166b 20 days ago

config	fix deprecated uses of Core()	2 months ago
consul	Add metrics	2 years ago
controller	Handle errors in services controller	2 months ago
examples	fix example for registering only key from labels	7 months ago
metrics	Fix typo from comment of a exported type	2 months ago
utils	Add support for Kubernetes Services, add vendor	2 years ago
vendor	Migrate to go mod from dep	2 months ago
.gitignore	Add support for Kubernetes Services, add vendor	2 years ago
.travis.yml	Migrate to go mod from dep	2 months ago
Dockerfile	Update Dockerfile and an examples	2 years ago
LICENSE	Update README.md and add license	2 years ago
Makefile	Migrate to go mod from dep	2 months ago
README.md	Fix typos in annotation names	5 months ago
go.mod	Migrate to go mod from dep	2 months ago
go.sum	Migrate to go mod from dep	2 months ago
main.go	Migrate to go mod from dep	2 months ago

README.md

Kubernetes Consul Register

build passing

The kube-consul-register is a tool to register Kubernetes PODs as Consul Services.

kube-consul-register watches Kubernetes events and converts information about PODs to Consul Agent.

Usage

```
-alsologtostderr
    log to standard error as well as files
-clean-interval duration
    time in seconds, what period of time will be done cleaning of inactive services (default 30m0s)
-configmap string
    name of the ConfigMap that contains the custom configuration to use (default "default/kube-consul-
register-config")
-in-cluster
    use in-cluster config. Use always in case when controller is running on Kubernetes cluster (default true)
-kubeconfig string
    absolute path to the kubeconfig file (default "./kubeconfig")
-log_backtrace_at value
    when logging hits line file:N, emit a stack trace
-log_dir string
    If non-empty, write log files in this directory
-logtostderr
    log to standard error instead of files
-stderrthreshold value
```

```

    logs at or above this threshold go to stderr
-sync-interval duration
    time in seconds, what period of time will be done synchronization (default 2m0s)
-v value
    log level for V logs
-version
    print version and exit
-vmodule value
    comma-separated list of pattern=N settings for file-filtered logging
-watch-namespaces string
    namespace to watch for Pods. Default is to watch all namespaces

```

Configuration

To store configuration is used [ConfigMap](#). You can find [example of configuration](#) with default values in examples directory. In order to use ConfigMap configuration you've to use `configmap` flag. Value of this flag has format `namespace/configmap_name`, e.g. `-configmap="default/kube-consul-register-config"`.

Option name	Default value	Description
<code>consul_address</code>	<code>localhost</code>	The address of Consul Agent. This option is taken into account only in case where <code>register_mode</code> is set to <code>single</code>
<code>consul_port</code>	<code>8500</code>	The port number of Consul Agent
<code>consul_scheme</code>	<code>http</code>	Connection scheme. Available options: <code>http</code> , <code>https</code> , <code>consul-unix</code>
<code>consul_ca_file</code>		Path to a CA certificate file
<code>consul_cert_file</code>		Path to an SSL client certificate to use to authenticate to the Consul server
<code>consul_key_file</code>		Path to an SSL client certificate key to use to authenticate to the Consul server
<code>consul_insecure_skip_verify</code>	<code>false</code>	Skip verifying certificates when connecting via SSL
<code>consul_token</code>		The Consul ACL token. Token is used to provide a per-request ACL token which overrides the agent's default token
<code>consul_timeout</code>	<code>2s</code>	Time limit for requests made by the Consul HTTP client. A Timeout of zero means no timeout
<code>consul_container_name</code>	<code>consul</code>	The name of container in POD with Consul Agent. The container with given name will be skip and not registered in Consul. This options is taken into account only if <code>register_mode</code> is set to <code>pod</code>
<code>consul_node_selector</code>	<code>consul=enabled</code>	Node label which is used to select nodes with Consul agent. This option is taken into account only if <code>register_mode</code> is equal to <code>node</code>
<code>pod_label_selector</code>		Pay heed only to PODs with the given label
<code>k8s_tag</code>	<code>kubernetes</code>	The name of tag which is added to every Consul Service. This tag identifies all Consul Services which has been registered by kube-consul-register
<code>register_mode</code>	<code>single</code>	The mode of register. Available options: <code>single</code> , <code>pod</code> , <code>node</code>
<code>register_source</code>	<code>pod</code>	Source name which is watching in order to add services to Consul. Available options: <code>pod</code> , <code>service</code> , <code>endpoint</code>

Register mode

The `register_mode` option determine to which Consul Agent a services should be registered.

- `single` - registers all services in one agent. The address of agent is taken from `consul_address` option.
- `pod` - registers service in agent which is running as container is the same pod, as Consul Agent address is taken a IP address of pod.

- `node` - register service in agent which is running on the same node where service, as Consul Agent address is taken a name of node.

Register source

`kube-consul-register` as default watches PODs and converts information about them into Consul Services, as alternative you can use Kubernetes Services or Endpoints.

In order to use Kubernetes Endpoints as source of information you have to set value of `register_source` option on endpoints, additionally you have to add annotation into specific endpoint.

```
# create service
kubectl expose deployment my-nginx --port=80 --type=LoadBalancer

# add annotation
kubectl annotate endpoints my-nginx consul.register/enabled=true
```

If you want to use Kubernetes Services you have to set value of `register_source` on service, only service with type `NodePort` is take into account.

Annotations

There are available annotations which can be used as pod's annotations.

Name	Value	Description
<code>consul.register/enabled</code>	<code>true</code> <code>false</code>	Determine if pod should be registered in Consul. This annotation is require in order to register pod as Consul service
<code>consul.register/service.name</code>	<code>service_name</code>	Determine name of service in Consul. If not given then is used the name of resource which created the POD. Only available if <code>register_source</code> is set on pod
<code>consul.register/pod.container.name</code>	<code>container_name</code>	Container name or list of names (next name should be separated by comma) which will be taken into account. If omitted, all containers in POD will be registered

The example of how to use annotation you can see [here](#).

Examples of usage

Run out-of-cluster

```
$ kube-consul-register -logtostderr=true -kubeconfig=/my/kubeconfig -configmap="default/kube-consul-register" -in-cluster=false
```

Run in-cluster

Example of usage in-cluster you can find [here](#). `kube-consul-register` is run as ReplicaSet.

Metrics

Prometheus metrics are available by `/metrics` endpoint on `:8080` address.