Kubernetes outside of Daemonset

Running outside of a daemonset

It is highly recommended that customers run the agent under a daemonset because it provides for a lot of automatic configuration. If the agent is run outside of Kubernetes, e.g. directly via docker run or host-native, the Kubernetes cluster information needs to be manually configured for the agent.

IP (and optional port) of the k8s api server

```
k8s_uri: https://1.2.3.4:6443/
```

Client auth info

Bearer token

Recent versions of kubernetes have RBAC enabled by default, so the agent needs to authenticate with the api server. When using a serviceaccount, you can get the account's bearer token using kubectl. This example assumes you're using the "sysdig-agent" serviceaccount. See https://kubernetes.io/docs/reference/access-authn-authz/authentication/ for more details.

```
kubectl get secret `kubectl get sa/sysdig-agent -o yaml | grep token |
awk '{print $3}'` -o yaml | grep token | awk '{print $2}' | base64 -D >
/local_path/to/sysdig-agent-bearer-token
```

Once you have the bearer token, it needs to be mounted in the container and passed to the config.

```
k8s_bt_auth_token: /container_path/to/sysdig-agent-bearer-token
```

Client cert+key

Alternatively, you can pass the client certificate and key. How to get these files depends on your kubernetes installation.

```
k8s_ssl_cert: /container_path/to/client.crt
k8s_ssl_key: /container_path/to/client.key
```

Server auth info

The server certificate can also be obtained via kubectl if you're using a serviceaccount, and it must be mounted in the container.

```
kubectl get secret `kubectl get sa/sysdig-agent -o yaml | grep token |
awk '{print $3}'` -o yaml | grep ca.crt | awk '{print $2}' | base64 -D >
/local_path/to/ca.crt
```

```
k8s_ca_certificate: /container_path/to/ca.crt
```