SECURITY

Security

Authentication

There are three main points to remember with authentication in Kubernetes:

- In its straightforward form, authentication is done with certificates, tokens or basic authentication (i.e. username and password).
- Users are not created by the API, but should be managed by an external system.
- System accounts are used by processes to access the API (to learn more read *Configure Service Accounts for Pods*).

If you want to learn more on how system accounts are used by processes to access the API:

There are two more advanced authentication mechanisms. Webhooks can be used to verify bearer tokens, and connection with an external OpenID provider.

The type of authentication used is defined in the kube-apiserver startup options. Below are four examples of a subset of configuration options that would need to be set depending on what choice of authentication mechanism you choose:

```
--basic-auth-file
```

```
--oidc-issuer-url
```

```
--token-auth-file
```

```
--authorization-webhook-config-file
```

One or more Authenticator Modules are used:

- x509 Client Certs;
- static token, bearer or bootstrap token;
- static password file;
- service account;
- OpenID connect tokens.

Each is tried until successful, and the order is not guaranteed. Anonymous access can also be enabled, otherwise you will get a 401 response. Users are not created by the API, and should be managed by an external system.

To learn more about authentication, see the official Kubernetes documentation.