Azure Active Directory plugin for client authentication

This plugin provides an integration with Azure Active Directory device flow. If no tokens are present in the kubectl configuration, it will prompt a device code which can be used to login in a browser. After login it will automatically fetch the tokens and store them in the kubectl configuration. In addition it will refresh and update the tokens in the configuration when expired.

Usage

1. Create an Azure Active Directory *Web App / API* application for `apiserver` following these instructions. The callback URL does not matter (just cannot be empty).

2. Create a second Azure Active Directory native application for `kubectl`. The callback URL does not matter (just cannot be empty).

3. On `kubectl` application's configuration page in Azure portal grant permissions to `apiserver` application by clicking on *Required Permissions*, click the *Add* button and search for the apiserver application created in step 1. Select "Access apiserver" under the *DELEGATED PERMISSIONS*. Once added click the *Grant Permissions* button to apply the changes.

4. Configure the `apiserver` to use the Azure Active Directory as an OIDC provider with following options

   ```
   --oidc-client-id="spn:APISERVER_APPLICATION_ID" \
   --oidc-issuer-url="https://sts.windows.net/TENANT_ID/"
   --oidc-username-claim="sub"
   ```

   - Replace the `APISERVER_APPLICATION_ID` with the application ID of `apiserver` application
   - Replace `TENANT_ID` with your tenant ID.

 * For a list of alternative username claims that are supported by the OIDC issuer check the JSON response at `https://sts.windows.net/TENANT_ID/.well-known/openid-configuration`.

5. Configure `kubectl` to use the `azure` authentication provider

   ```
   kubectl config set-credentials "USER_NAME" --auth-provider=azure \
     --auth-provider-arg=environment=AzurePublicCloud \
     --auth-provider-arg=client-id=APPLICATION_ID \
     --auth-provider-arg=tenant-id=TENANT_ID \
     --auth-provider-arg=apiserver-id=APISERVER_APPLICATION_ID
   ```

   - Supported environments: `AzurePublicCloud`, `AzureUSGovernmentCloud`, `AzureChinaCloud`, `AzureGermanCloud`
   - Replace `USER_NAME` and `TENANT_ID` with your user name and tenant ID
   - Replace `APPLICATION_ID` with the application ID of your `kubectl` application ID
   - Replace `APISERVER_APPLICATION_ID` with the application ID of your `apiserver` application ID
   - Be sure to also (create and) select a context that uses above user

6. (Optionally) the AAD token has `aud` claim with `spn:` prefix. To omit that, add following auth configuration:

   ```
     --auth-provider-arg=config-mode="1"
   ```

7. The access token is acquired when first `kubectl` command is executed

```
kubectl get pods

To sign in, use a web browser to open the page https://aka.ms/devicelogin and enter
the code DEC7D48GA to authenticate.
```

- After signing in a web browser, the token is stored in the configuration, and it will be reused when executing further commands.
- The resulting username in Kubernetes depends on your configuration of the `--oidc-username-claim` and `--oidc-username-prefix` flags on the API server. If you are using any authorization method you need to give permissions to that user, e.g. by binding the user to a role in the case of RBAC.