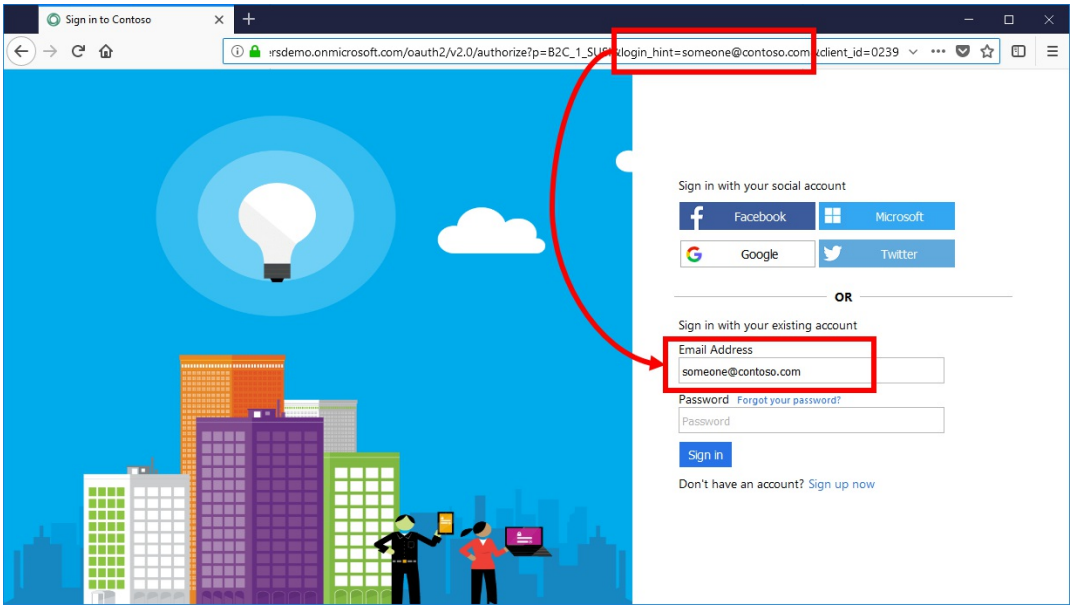
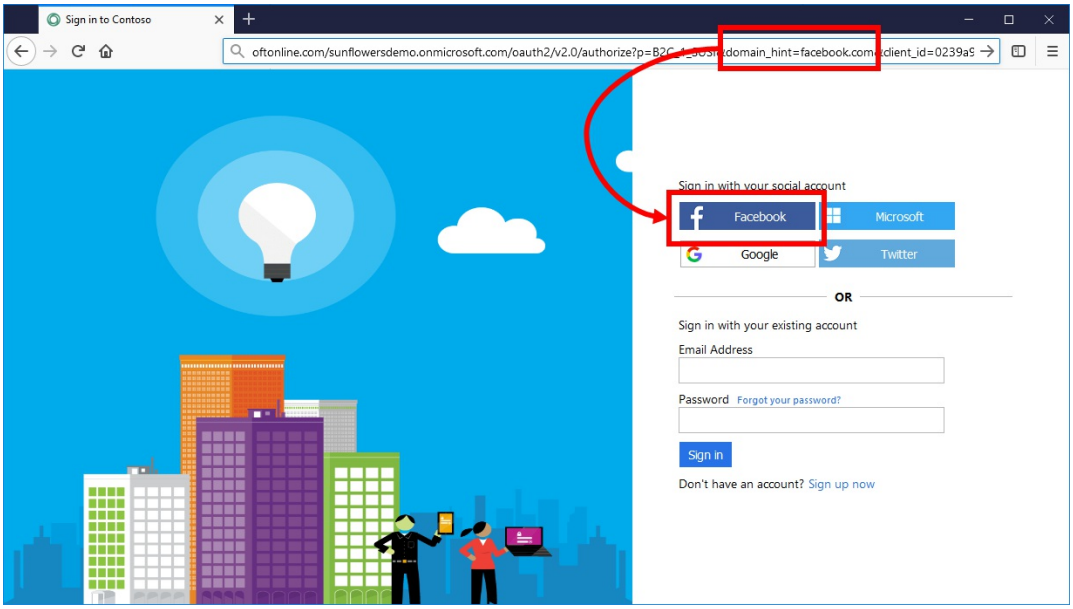
Targeting a sign-in user or domain name using login and domain hint

During sign-in user journey, a relying party application may target a specific user or domain name. When targeting a user, an application can specify (in the authorization request) the login_hint query string parameter with the user sign-in name. Azure AD B2C auto populates the sign-in name, while end user needs only to provide the password. Following screenshot demonstrates the use of login_hint parameter:



If your Azure AD B2C policy cofigured with social accounts, such as Facebook, LinkedIn, or Google, the application can specify the domain_hint parameter. This query parameter provides a hint to Azure AD B2C about the social identity provider user should sing-in. For example, if the application specifies domain_hint=facebook.com, the users will go directly to Facebook sign-in page, skipping Azure AD B2C sign-up or sign-up page (also known as Home Realm Discovery). Following screenshot demonstrates the use of domain_hint parameter:



Using login or domain hint parameter

To use the hint parameter (log-in or domain), the relying party application should know the username or domain name ahead of time, before starting the authorization request. Often relying party application use this parameter during reauthentication. Having already extracted the username or domain name from a previous sign-in. Or the relying party, first asks the end user for their e-mail address or username, and then start the sign-in user journey.

A relying party application adds the login_hint or domain_hint parameters as part of the authorization request. The value of the login_hint parameter is the end-user e-mail address or username, user uses to sign in. The value of the domain_hint is the social identity provider name. In custom policy, you can configure the domain name using `<Domain>facebook.com</Domain>` XML element of any `<ClaimsProvider>`.

Using Login hint in custom policy.

To prepopulate the sign-in name, in your custom policy, override the SelfAsserted-LocalAccountSignin-Email technical profile. In the `<InputClaims>` section, you set the signInName's claim's DefaultValue to {OIDC:LoginHint}. The {OIDC:LoginHint} variable contains the value of the login_hint parameter. Azure AD B2C reads the signInName input claim's value, and pre-populates the signInName textbox.

```
<ClaimsProvider>
    <DisplayName>Local Account</DisplayName>
    <TechnicalProfiles>
    <TechnicalProfile Id="SelfAsserted-LocalAccountSignin-Email">
        <InputClaims>
        <!-- Demo: Add the login hint value to the sign-in names claim type --
>
        <InputClaim ClaimTypeReferenceId="signInName" DefaultValue="
{OIDC:LoginHint}" />
        </InputClaims>
    </TechnicalProfile>
    </TechnicalProfiles>
</ClaimsProvider>
```

Pass the domain hint to enterprise identity provider using custom policy

If you add ADFS as an enterprise identity provider to Azure AD B2C, you can pass the login_hint and domain_hint forward to ADFS. When a user goes to the application and click log-in, user gets redirect to Azure AD B2C with a domain hint, while B2C further redirects the user to ADFS with the same domain hint, or another one you define.

To pass domain hint, first configure domain_hint ClaimType. You use this claim type to pass the domain_hint value the ADFS.

```
<ClaimType Id="domain_hint">
    <DisplayName>domain_hint</DisplayName>
```

```
    <DataType>string</DataType>
  </ClaimType>
```

In your technical profile, add <InputClaim> with the domain name. To pass the original domain name sent from relying party application to Azure AD B2C use the {OIDC:DomainHint}.

```
  <InputClaims>
      <InputClaim ClaimTypeReferenceId="domain_hint"
  DefaultValue="yourdomain.com" />
  </InputClaims>
```

```
<ClaimsProvider>
    <Domain>contoso.com</Domain>
    <DisplayName>Contoso ADFS</DisplayName>
    <TechnicalProfiles>
        <TechnicalProfile Id="Contoso-SAML2">
            <DisplayName>Contoso ADFS</DisplayName>
            <Description>Login with your Contoso account</Description>
            <Protocol Name="SAML2"/>
            <Metadata>
                <Item Key="RequestsSigned">false</Item>
                <Item Key="WantsEncryptedAssertions">false</Item>
                <Item Key="PartnerEntity">https://{your_ADFS_domain}/federationmetadata/2007-06/
            </Metadata>
            <CryptographicKeys>
                <Key Id="SamlAssertionSigning" StorageReferenceId="B2C_1A_ADFSSamlCert"/>
                <Key Id="SamlMessageSigning" StorageReferenceId="B2C_1A_ADFSSamlCert"/>
            </CryptographicKeys>
            <!-- Demo: set the domain hint-->
            <InputClaims>
                <InputClaim ClaimTypeReferenceId="domain_hint" DefaultValue="pinedemo.com" />
            </InputClaims>
```

In the same way, you can also pass the login_hint parameter. Create new claim type login_hint. In your technical profile, set the value to the {OIDC:LoginHint}