Demystify module federation, and deployment strategy on AWS

Chengwei Lim
Lead Software Engineer
UI Architect

www.linkedin.com/in/chengwei-lim-9985583

Life of a UI developer

I usually wait for 1 or 2 minutes for the SPA to build to test my changes. I navigate multiple screens to get to the feature I am working on. Repeat the same process n times until the feature is working.

Life of a UI developer

The CI/CD Pipeline takes a long time to run. It takes hours later to learn that tests have failed for some reasons. Let’s find out which team owns the test and notify the team.

Life of a UI developer

Last night, production release did not go well. Although our feature was verified successfully, we had to roll back the whole release because Feature A by Team A did not work as expected.

What if …

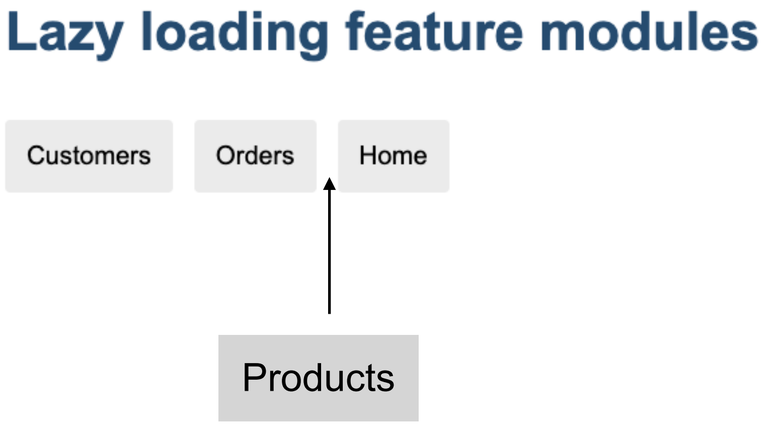
I can just focus on the feature that I am working on and make it perfect.
I can verify the feature without going through the entire application all the time.
I can release anytime but also limit failure blast radius.

Agenda

- Webpack Module Federation
- Web Component
- Build first and integrate later
- AWS serverless hosting

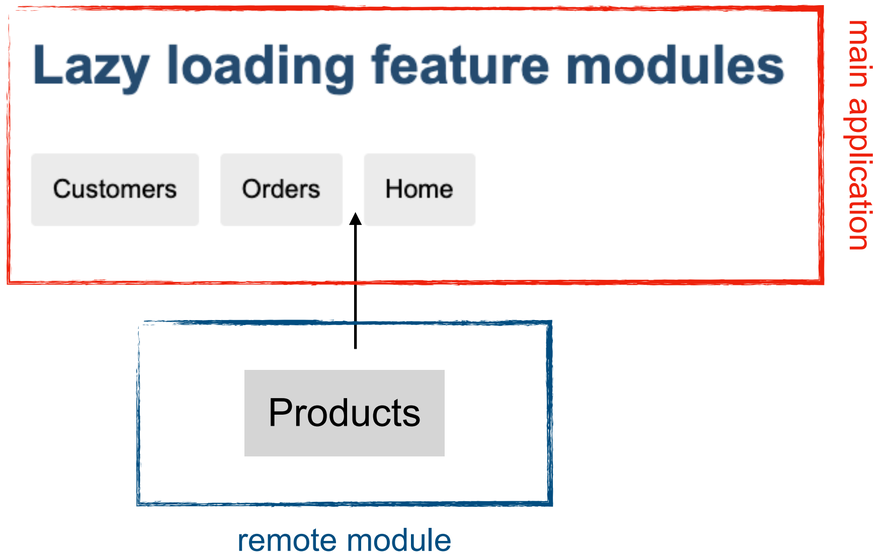
Requirement

Add Products feature to an existing Angular Application (https://angular.io/guide/lazy-loading-ngmodules)

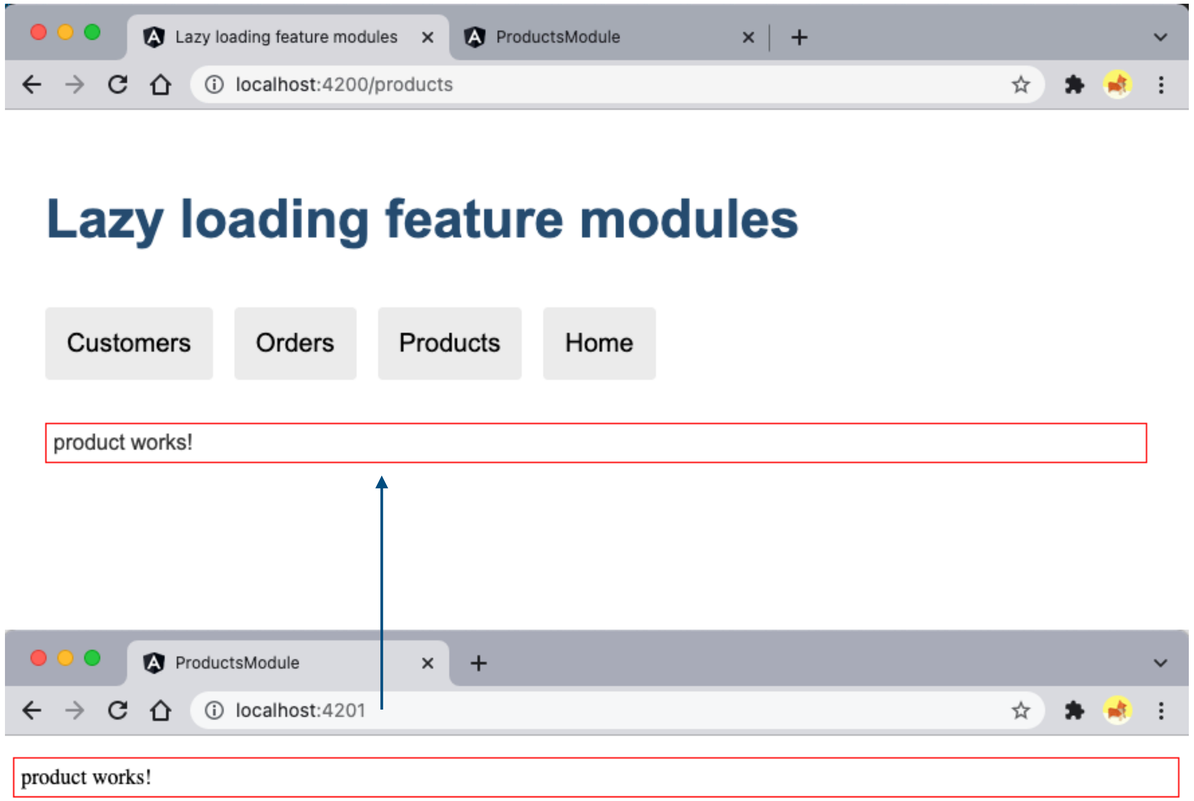


Requirement

Products module can be deployed independently and can be imported into the main application



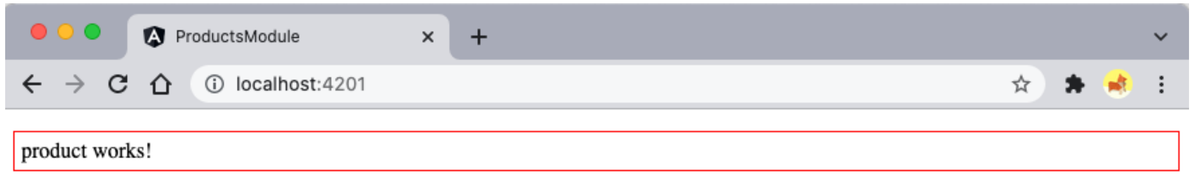
External Angular Module



ng add @angular-architects/module-federation (port 4201)

Products Module

```
## webpack.config.js

plugins: [
    new ModuleFederationPlugin({
        name: "productsModule",
        filename: "remoteEntry.js",
        exposes: {
            './module': './src/app/products.module.ts',
        },
        shared: share({
          "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },

          ...sharedMappings.getDescriptors()
        })
    }),
    sharedMappings.getPlugin()
  ],
```



ng add @angular-architects/module-federation (port 4200)

Angular main app

```
## webpack.config.js

plugins: [
    new ModuleFederationPlugin({
        remotes: {
            "productsModule": "productsModule@http://localhost:4201/remoteEntry.js",
        },
        shared: share({
          "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          ...sharedMappings.getDescriptors()
        })
    }),
    sharedMappings.getPlugin()
  ],


 ## app-routing.module.ts

 const routes: Routes = [
 …
   {
     path: 'products',
     loadChildren: () => import('productsModule/module').then( m => m.ProductsModule)
   },
 …
 ];


<!-- app.component.html ->

<button routerLink="/customers">Customers</button>
<button routerLink="/orders">Orders</button>
<button routerLink="/products">Products</button>
```

Putting together

Lazy loading feature modules

Customers | Orders | Products | Home

product works!

```
## app-routing.module.ts

const routes: Routes = [
…
  {
    path: 'products',
    loadChildren: () => import('productsModule/module').then( m => m.ProductsModule)
  },
…
];
```

3

product works!

Angular main app

```
ng add @angular-architects/module-federation

## webpack.config.js

plugins: [
    new ModuleFederationPlugin({
        remotes: {
            "productsModule": "productsModule@http://localhost:4201/remoteEntry.js",
        },
        shared: share({
          "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          ...sharedMappings.getDescriptors()
        })
    }),
    sharedMappings.getPlugin()
],
```

2

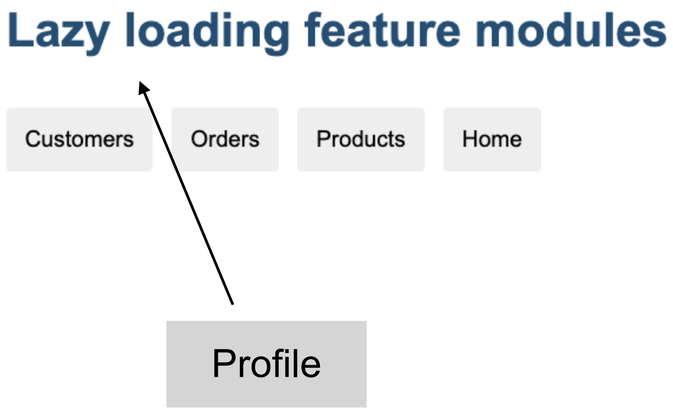
Products Module

```
ng add @angular-architects/module-federation


## webpack.config.js

plugins: [
    new ModuleFederationPlugin({
        name: "productsModule",
        filename: "remoteEntry.js",
        exposes: {
          './module': './src/app/products.module.ts',
        },
        shared: share({
          "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },

          ...sharedMappings.getDescriptors()
        })
    }),
    sharedMappings.getPlugin()
],
```

1

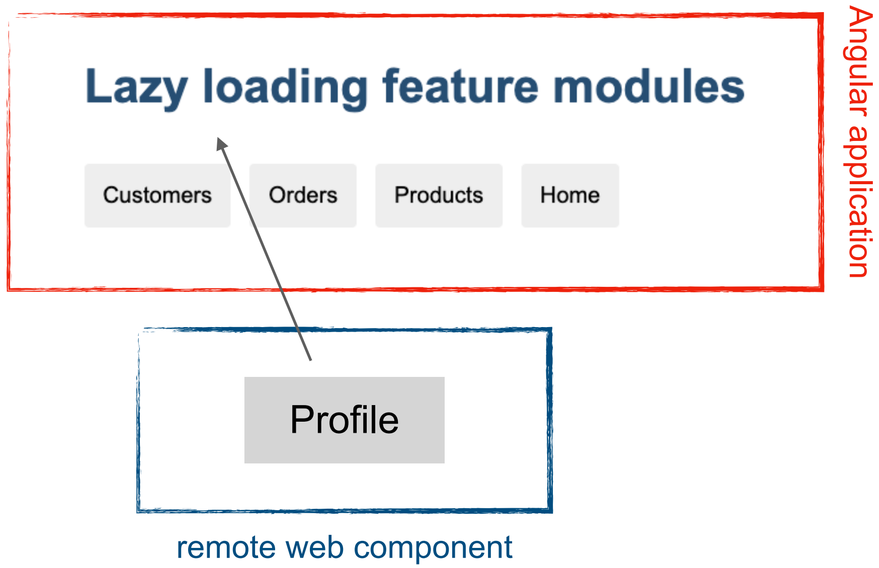
Requirement

Add a Profile feature to the main application.

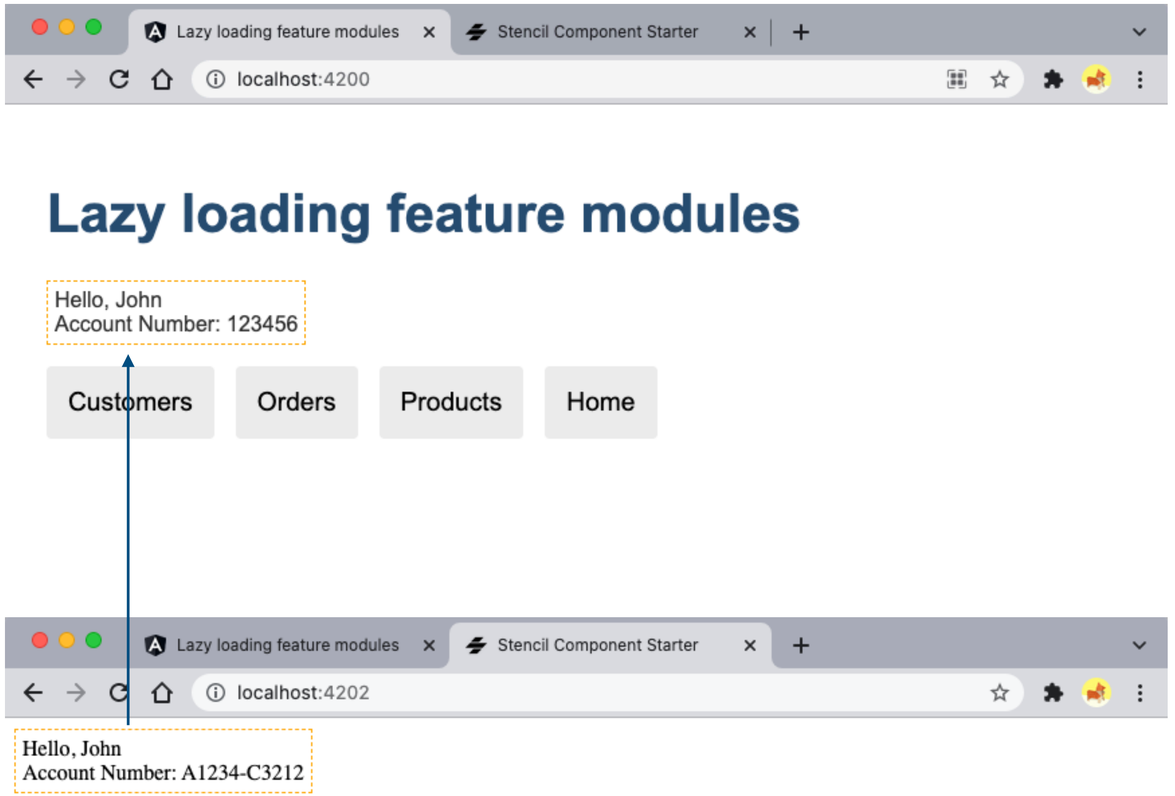


Requirement

The team who is responsible for this feature would like to build it as a web component.



Web Component



Importing Web Component in Angular

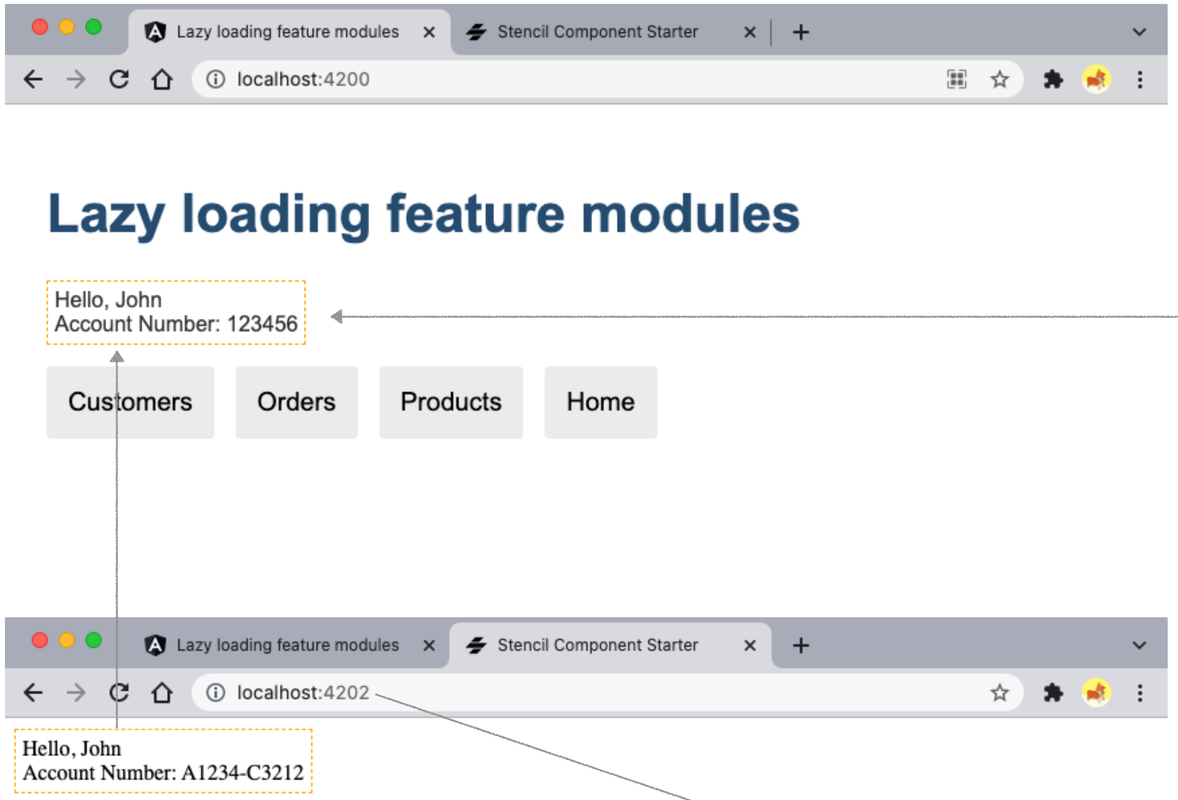
Angular main app

```
## app.module.ts

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA],
})
export class AppModule {
  constructor(){
    import(/* webpackIgnore: true */ `http://localhost:4202/profile-web-component.esm.js`);
  }
}
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Lazy loading feature modules';
  accountNumber = '12344';
  enabled = true;
}
```

```
<!-- app.component.html -->

<profile-component [attr.account-number]="accountNumber"></profile-component>

<button routerLink="/customers">Customers</button>
<button routerLink="/orders">Orders</button>
<button routerLink="/products">Products</button>
<button routerLink="">Home</button>


<router-outlet></router-outlet>
```

Putting together




```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Lazy loading feature modules';
  accountNumber = '12344';
  enabled = true;
}


<h1>
  {{title}}
</h1>
<profile-component [attr.account-number]="accountNumber"></profile-component>

<button routerLink="/customers">Customers</button>
<button routerLink="/orders">Orders</button>
<button routerLink="/products">Products</button>
<button routerLink="">Home</button>


<router-outlet></router-outlet>



@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA],
})
export class AppModule {
  constructor(){
    import(/* webpackIgnore: true */ `http://localhost:4202/profile-web-component.esm.js`);
  }
}
```

2

1

Angular main app

localhost only ???

```
## webpack.config.js

plugins: [
    new ModuleFederationPlugin({
        remotes: {
            "productsModule": "productsModule@http://localhost:4201/remoteEntry.js",
        },
        shared: share({
          "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          ...sharedMappings.getDescriptors()
        })
    }),
    sharedMappings.getPlugin()
  ],
```

```
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    AppRoutingModule
  ],
  providers: [],
  bootstrap: [AppComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA],
})
export class AppModule {
  constructor(){
    import(/* webpackIgnore: true */`http://localhost:4202/profile-web-component.esm.js`);
  }
}
```

ExternalTemplateRemotesPlugin

webpack.config.js

```
plugins: [
    new ModuleFederationPlugin({

        remotes: {
            "productsModule": "productsModule@[window.remote.productsModuleUrl]/remoteEntry.js",
        },

        shared: share({
          "@angular/core": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/common/http": { singleton: true, strictVersion: true, requiredVersion: 'auto' },
          "@angular/router": { singleton: true, strictVersion: true, requiredVersion: 'auto' },

          ...sharedMappings.getDescriptors()
        })

    }),
    new ExternalTemplateRemotesPlugin(),
    sharedMappings.getPlugin()
  ],
```

main.ts

```
const remote = {
  productsModuleUrl: 'http://demo-products-module.s3-website-us-east-1.amazonaws.com',
  profileComponentUrl: 'http://demo-profile-wc.s3-website-us-east-1.amazonaws.com',
};

(window as any).remote = remote;
```

```
import('./bootstrap')
  .catch(err => console.error(err));
```

app.module.ts

```
export class AppModule {
  constructor(){
    import(/* webpackIgnore: true */`${(window as any).remote.profileComponentUrl}/profile-web-component.esm.js`);
  }
}
```

Import a remote URL

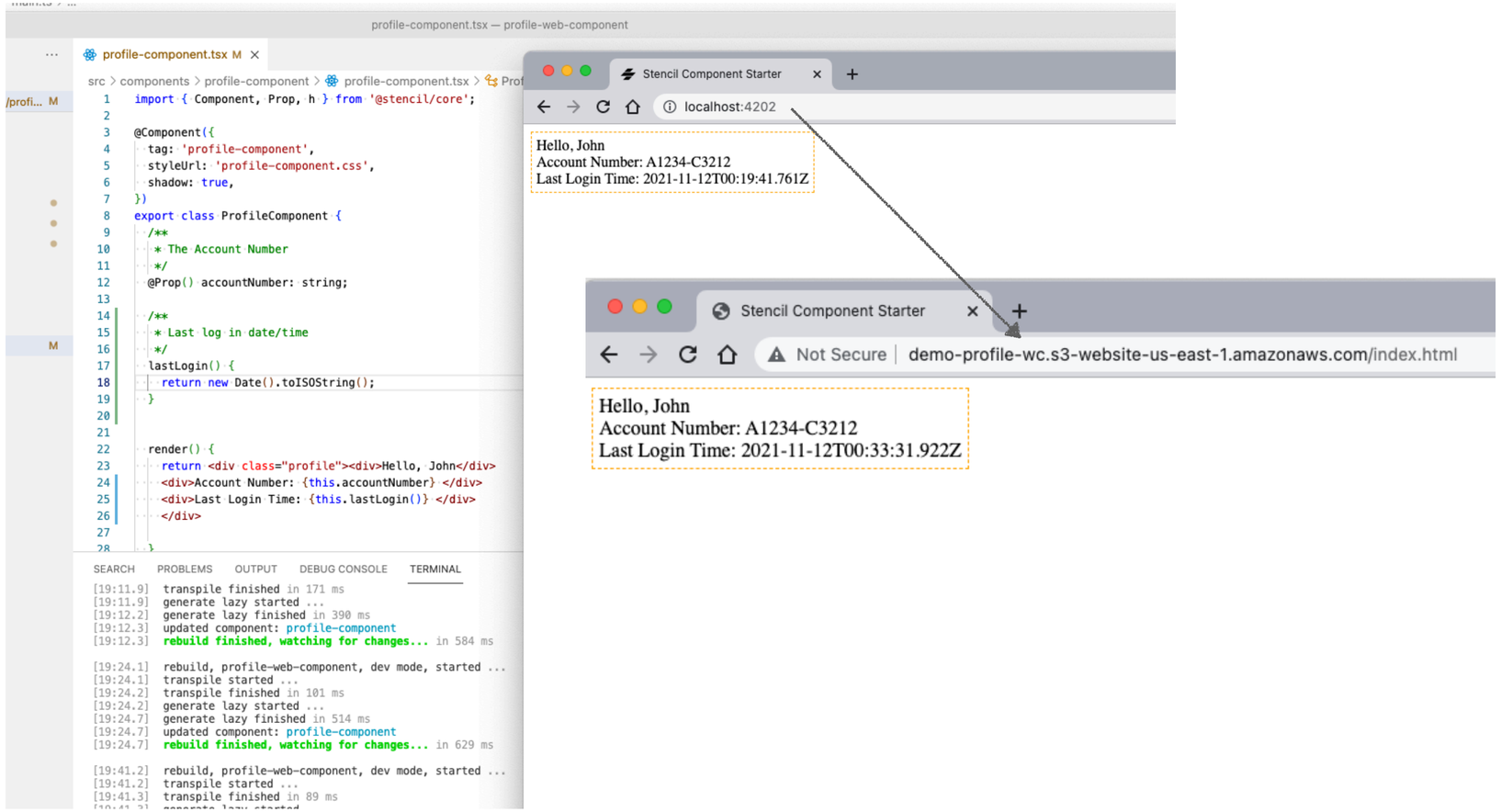
```
## app-routing.module.ts

const routes: Routes = [
…
  {
    path: 'products',
    loadChildren: () => import('productsModule/module').then( m => m.ProductsModule)
  },
…
];
```

Alias from webpack config

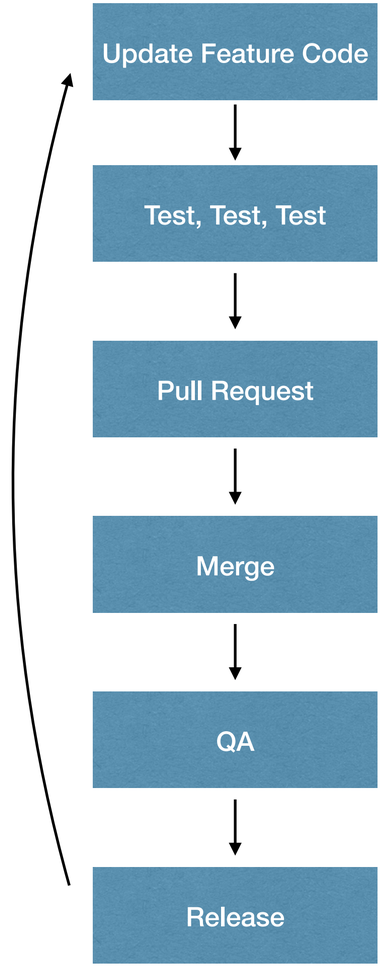
```
## app.module.ts

export class AppModule {
  constructor(){
    import(/* webpackIgnore: true */ `${(window as any).remote.profileComponentUrl}/profile-web-component.esm.js`);
  }
}
```

```
<!-- this works too -->

<script type="module" src="http://demo-profile-wc.s3-website-us-east-1.amazonaws.com/profile-web-component.esm.js"></script>
```

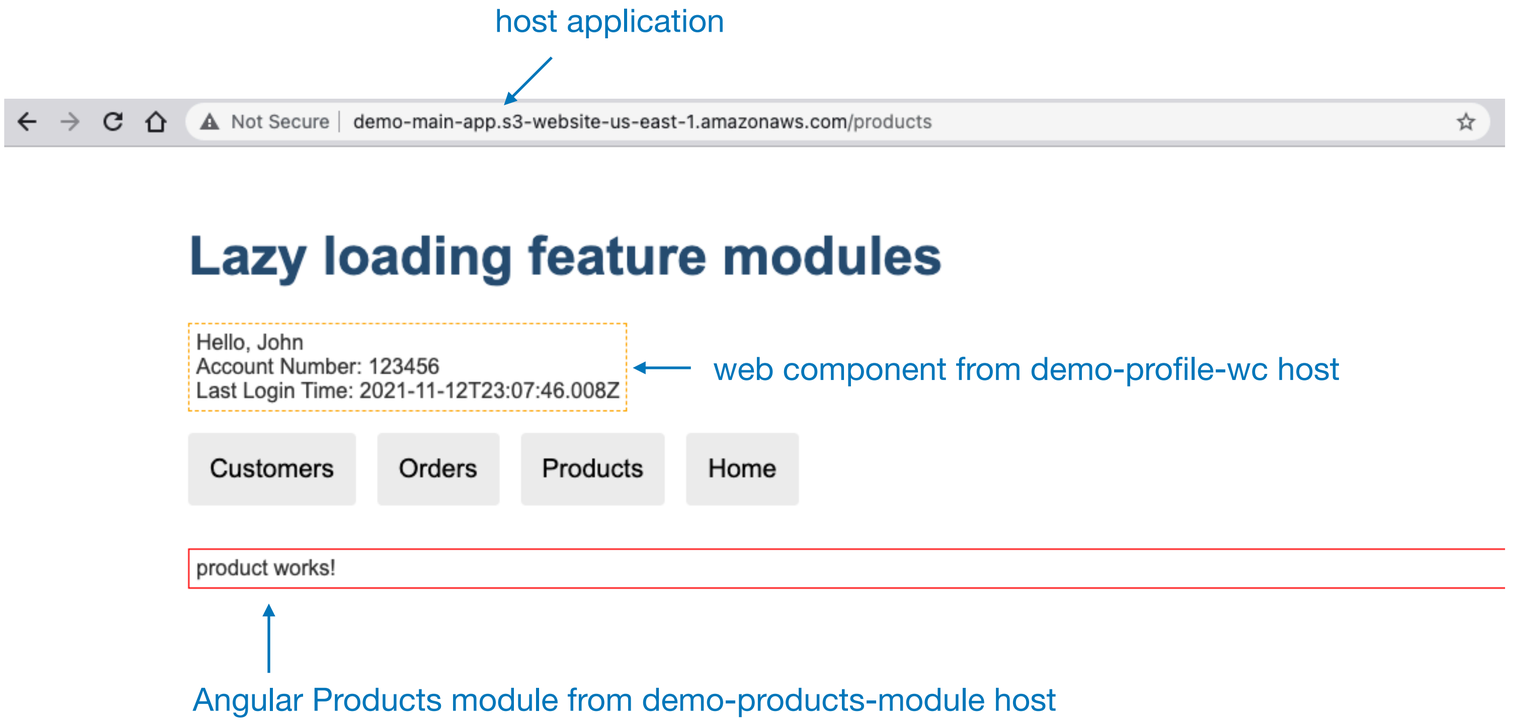
Focus on the target feature: build first and integrate later



Team Profile



Main App is updated without a release



Demo: Update the Profile Component and deploy it

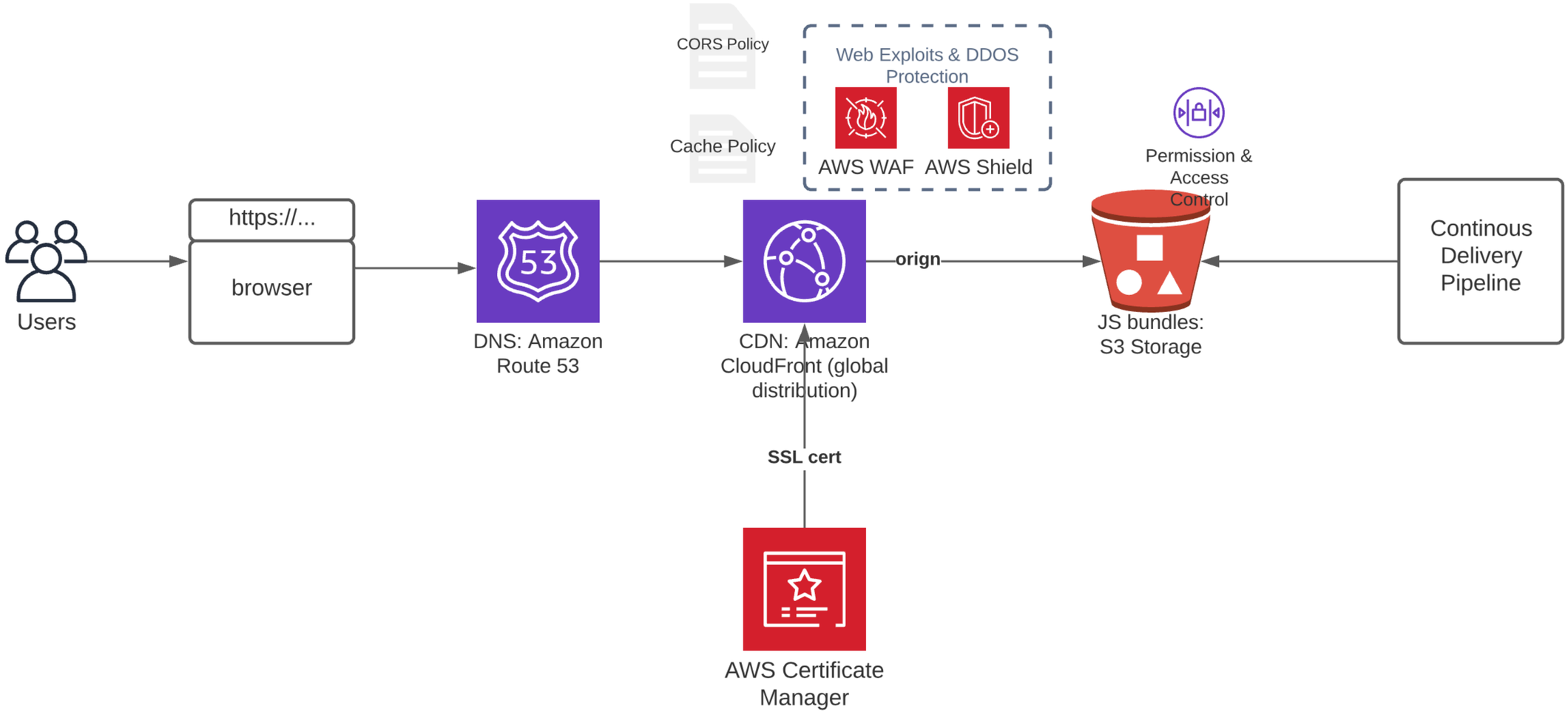
Will it be a nightmare to maintain so many web application servers?

Serverless: Hosting web application static assets without managing any servers

AWS for web application

DNS	Route53	Look up domain name and route traffic to your application
Web Server	CloudFront	Global content delivery network. The content is fetched from its origin and is cached and served to end users at the nearest point.
WAF	AWS WAF	Web exploits protection
DDoS	AWS Shield	DDoS protection
SSL Certification	AWS Certificate Manager (ACM)	Create or import SSL certificates for CloudFront
Static content files storage	AWS S3 Bucket	Store static contents for the web application. In our case, the HTML and JS files.

Go serverless with AWS CloudFront



Some considerations

- Use ShadowDOM
- Limit sharing any state or storage
- Agree on a common design system
- Enable CORS

Recap

- ☑ Forgot to mention. That is Micro Front End.
- ☑ Modern SPA web application:
  - ☑ Webpack Module Federation or ES Module
  - ☑ Cloud Serverless simplifies infrastructure design
- ☑ Focus on something small is easier than something big
- ☑ Build and test in isolation
- ☑ Add more features but keeping the main application small
- ☑ Speed to market. More features in less time.

Resources

- https://webpack.js.org/concepts/module-federation/
- https://www.npmjs.com/package/@angular-architects/module-federation
- https://www.npmjs.com/package/external-remotes-plugin
- https://stenciljs.com/docs/getting-started
- https://aws.amazon.com/cloudfront/

Questions?

- www.linkedin.com/in/chengwei-lim-9985583
- chengwei.lim@capitalone.com