Generalizing the Azure Key Vault configuration

Let’s generalize the Azure Key vault configuration that all of our microservices will use from here on.

In Play.Common repo

1. Add the following NuGet packages:
   Azure.Extensions.AspNetCore.Configuration.Secrets
   Azure.Identity

2. Update ServiceSettings:

```
public class ServiceSettings
{
    …
    public string MessageBroker { get; init; }
    public string KeyVaultName { get; init; }
}
```

3. Add the Configuration directory

4. Add Extensions.cs:

```
namespace Play.Common.Configuration
{
    public static class Extensions
    {
        public static IHostBuilder ConfigureAzureKeyVault(this IHostBuilder builder)
        {
            return builder.ConfigureAppConfiguration((context, configurationBuilder) =>
                {
                    if (context.HostingEnvironment.IsProduction())
                    {
                        var configuration = configurationBuilder.Build();
                        var serviceSettings =
configuration.GetSection(nameof(ServiceSettings)).Get<ServiceSettings>();
                        configurationBuilder.AddAzureKeyVault(
                            new Uri($"https://{serviceSettings.KeyVaultName}.vault.azure.net/"),
                            new DefaultAzureCredential());
                    }
                });
        }
    }
}
```

5. Bump NuGet package version in README

6. Create and publish new NuGet package

7. Commit and push

In Play.Identity repo

8. Bump the Common NuGet version

9. Update Program.cs:

```
public class Program
{
    …
    public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
            .ConfigureAzureKeyVault()
            .ConfigureWebHostDefaults(webBuilder =>
            {
                …
            });
}
```

10. Bump container image version in README

11. Build and deploy new container image

12. Update the identity.yaml file:

```
spec:
  …
  template:
    …
    spec:
      containers:
        - name: identity
          image: playeconomy.azurecr.io/play.identity:1.0.7
          env:
            - name: ServiceSettings__MessageBroker
              value: SERVICEBUS
            - name: ServiceSettings__KeyVaultName
              value: playeconomy
          resources:
          …
```

13. Apply the updated configuration:

```
kubectl apply -f .\kubernetes\identity.yaml -n $namespace
```

14. Commit and push

Now that you know how to read secrets from Azure Key Vault, please move on to the next assignment where you will update all of your other Play Economy microservices to also read secrets from Key Vault.