

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ ім. М. Є.
Жуковського «Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій
Кафедра комп'ютерних систем, мереж і кібербезпеки

Лабораторна робота №5

З дисципліни: «Теорія та технології розроблення безпечних розподільних
систем»

Виконав:

студент 5 курсу групи №555 ім

Напряму підготовки

125 Кібербезпека та захист інформації

ст. Орлов Станіслав Валерійович

Прийняв:

аспірант

Карпенко Андрій Сергійович

Харків, 2023

Step 1. Running a project in Visual Studio

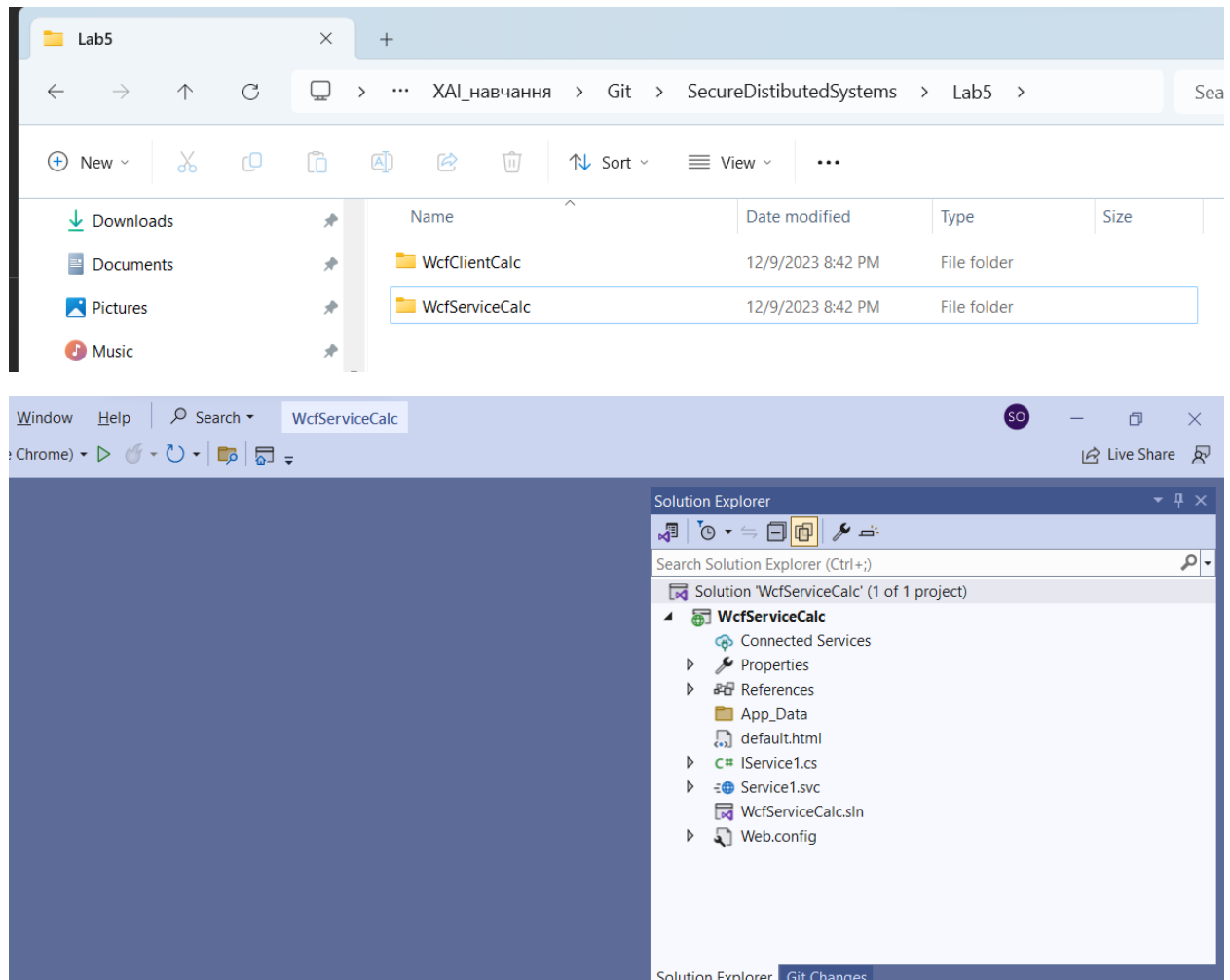


Рисунок 1 – WCF сервіс солюшен відкритий у Visual Studio

Step 2. Exploring C# WCF Service

```
namespace WcfServiceCalc
{
    /// NOTE: You can use the "Rename" command on the "Re
    /// NOTE: In order to launch WCF Test Client for test
    public class Service1 : IService1
    {
        public double Add(double n1, double n2)
        {
            return n1 + n2;
        }
        public double Subtract(double n1, double n2)
        {
            return n1 - n2;
        }
        public double Multiply(double n1, double n2)
        {
            return n1 * n2;
        }
        public double Divide(double n1, double n2)
        {
            return n1 / n2;
        }
    }
}
```

```
namespace WcfServiceCalc
{
    /// NOTE: You can use the "Rename" command on the
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        double Add(double n1, double n2);
        [OperationContract]
        double Subtract(double n1, double n2);
        [OperationContract]
        double Multiply(double n1, double n2);
        [OperationContract]
        double Divide(double n1, double n2);
    }
}
```

Рисунок 2 – контракт та реалізація сервісу

Launching and testing



Рисунок 3 – запуск сервісу та перегляд через браузер

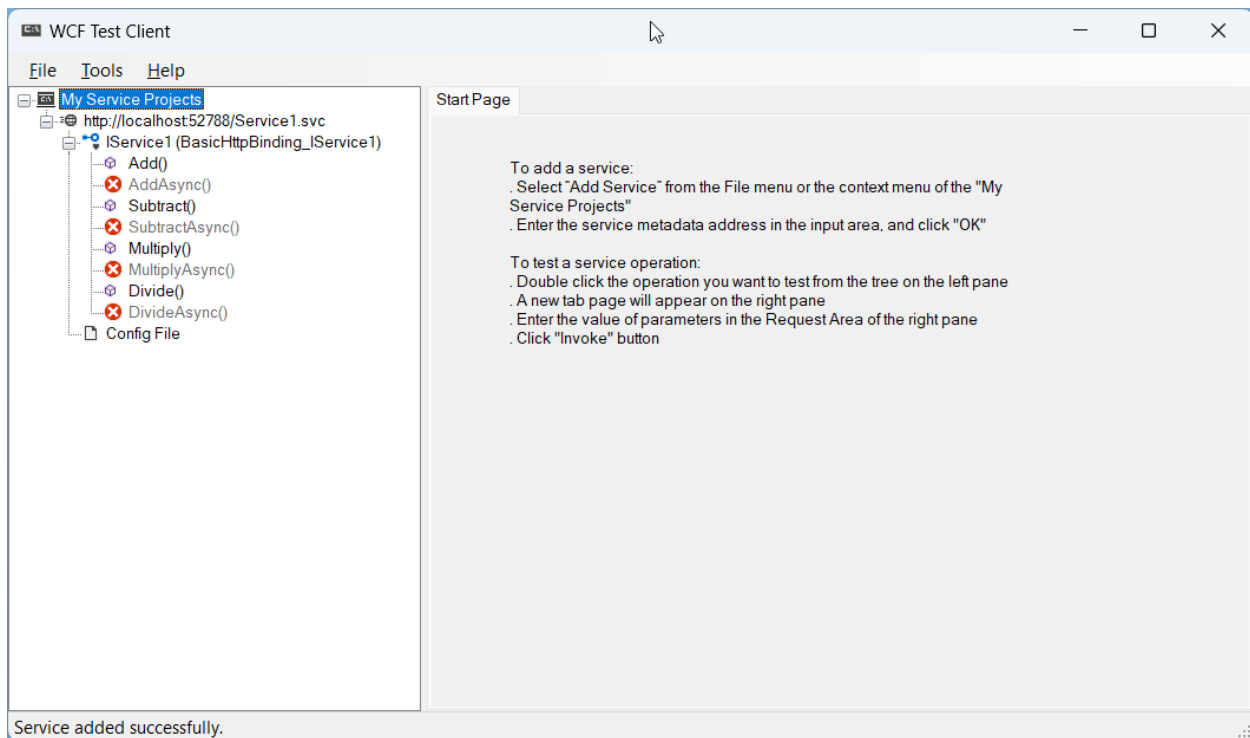


Рисунок 4 - WCF Test Client

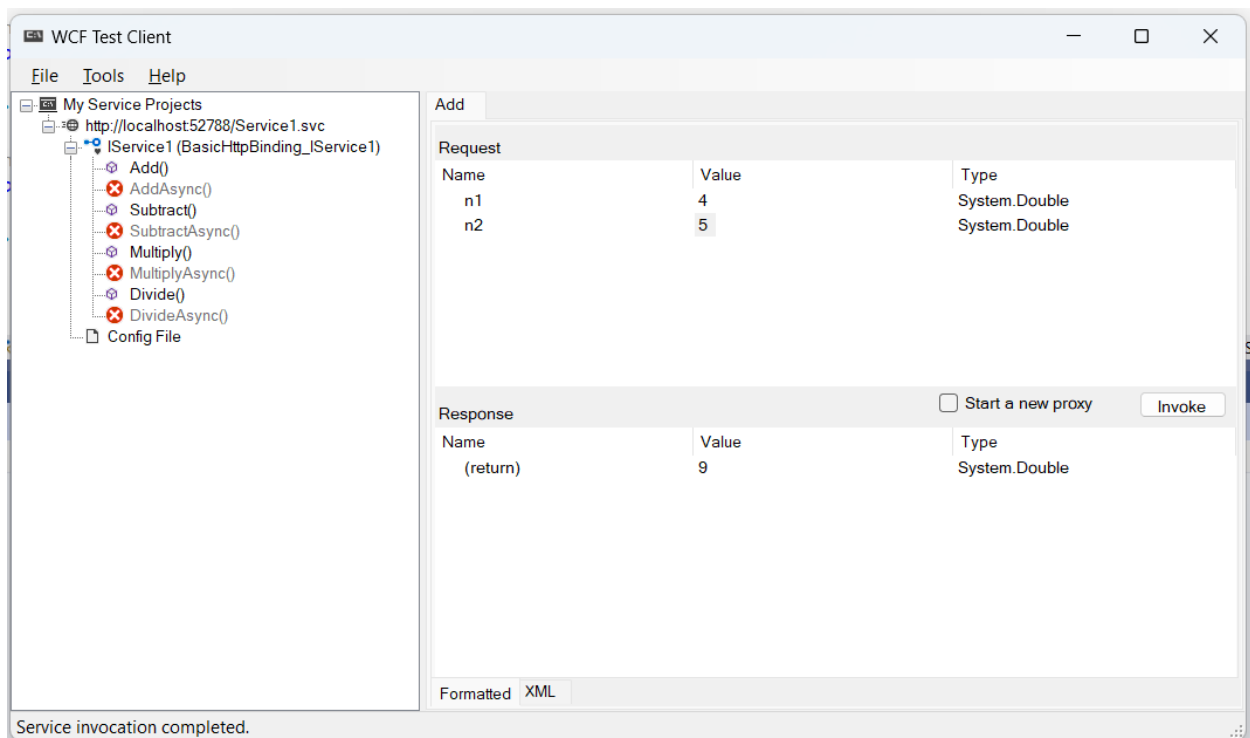


Рисунок 5 – результат виконання операції сервісу через Test Client

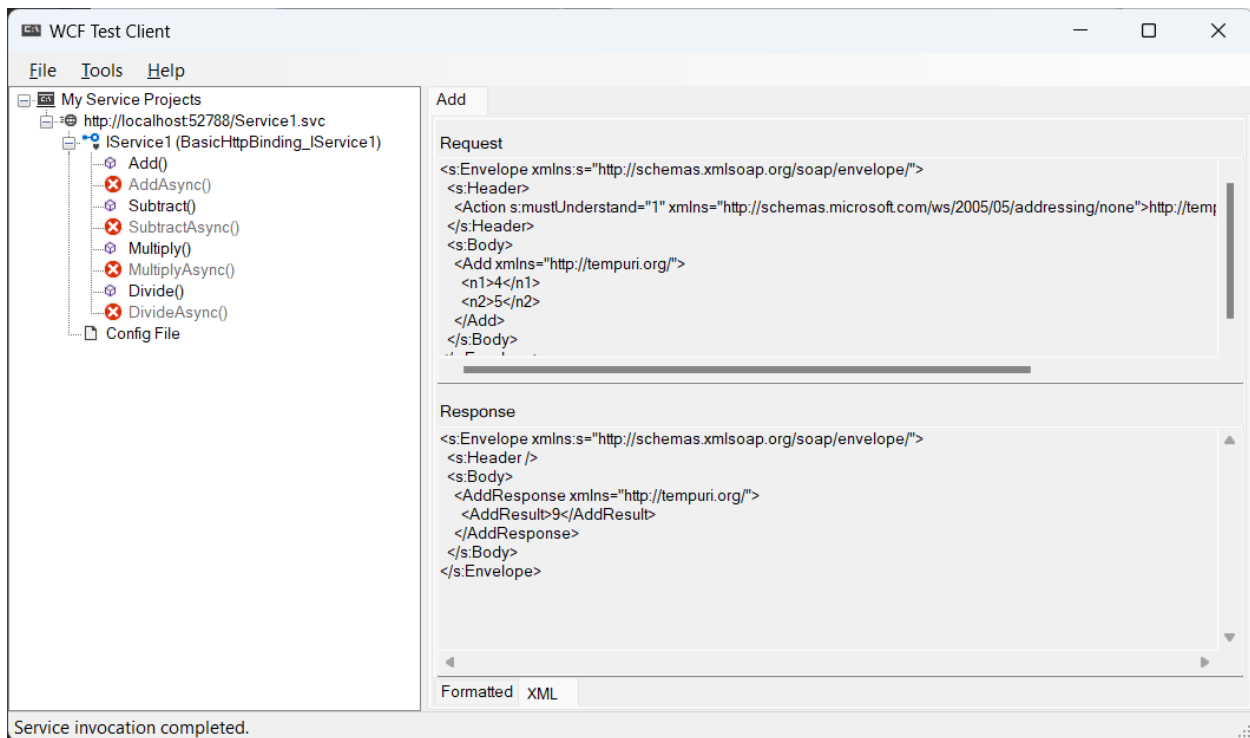


Рисунок 6 – SOAP контракт операції додавання у XML форматі

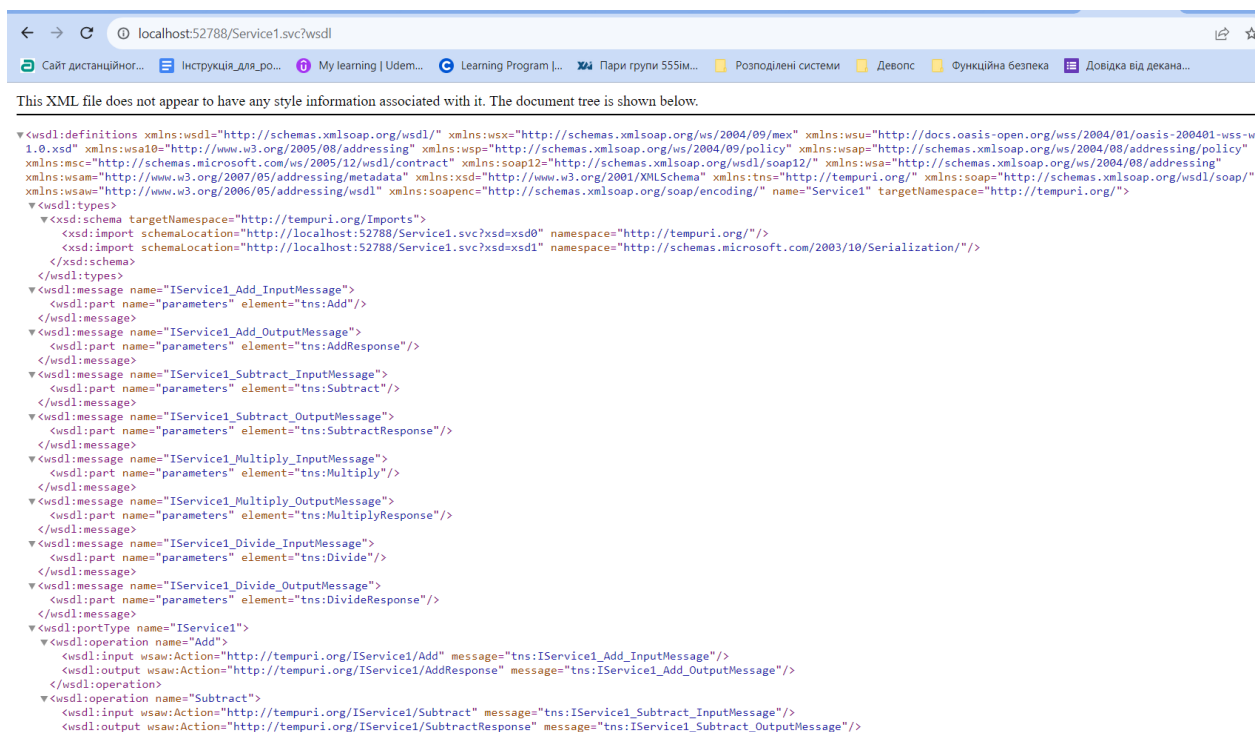


Рисунок 7 – WSDL опис сервісу у браузері

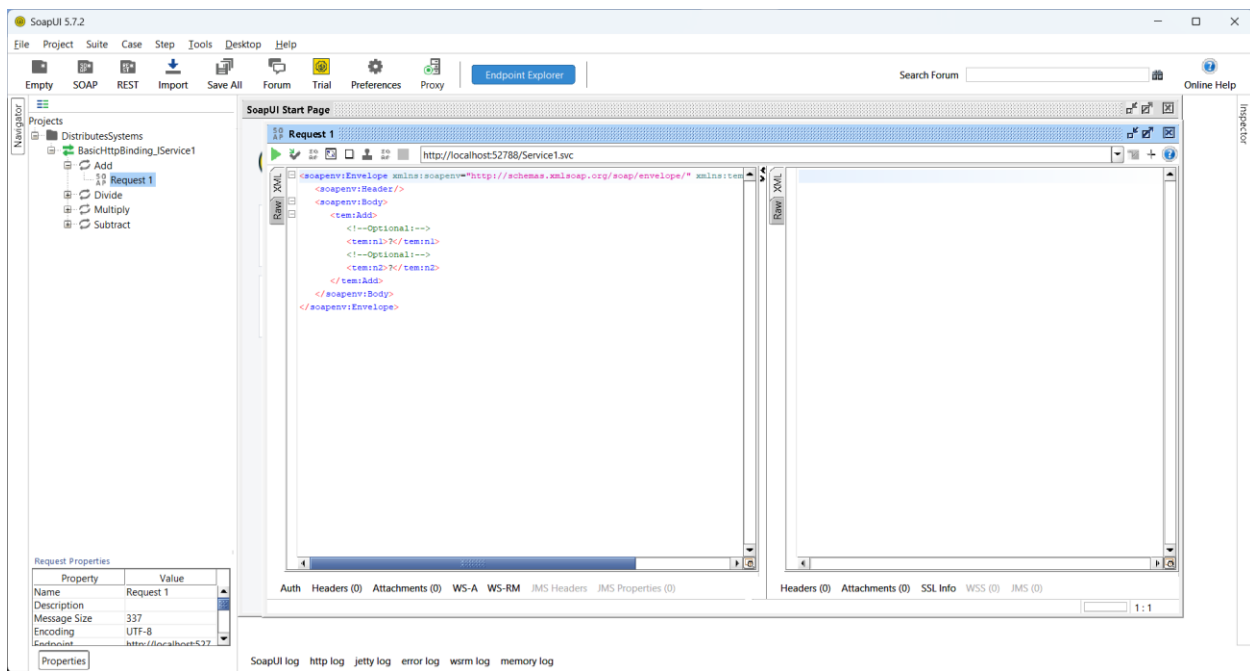


Рисунок 8 – WCF сервіс відкритий у SoapUI

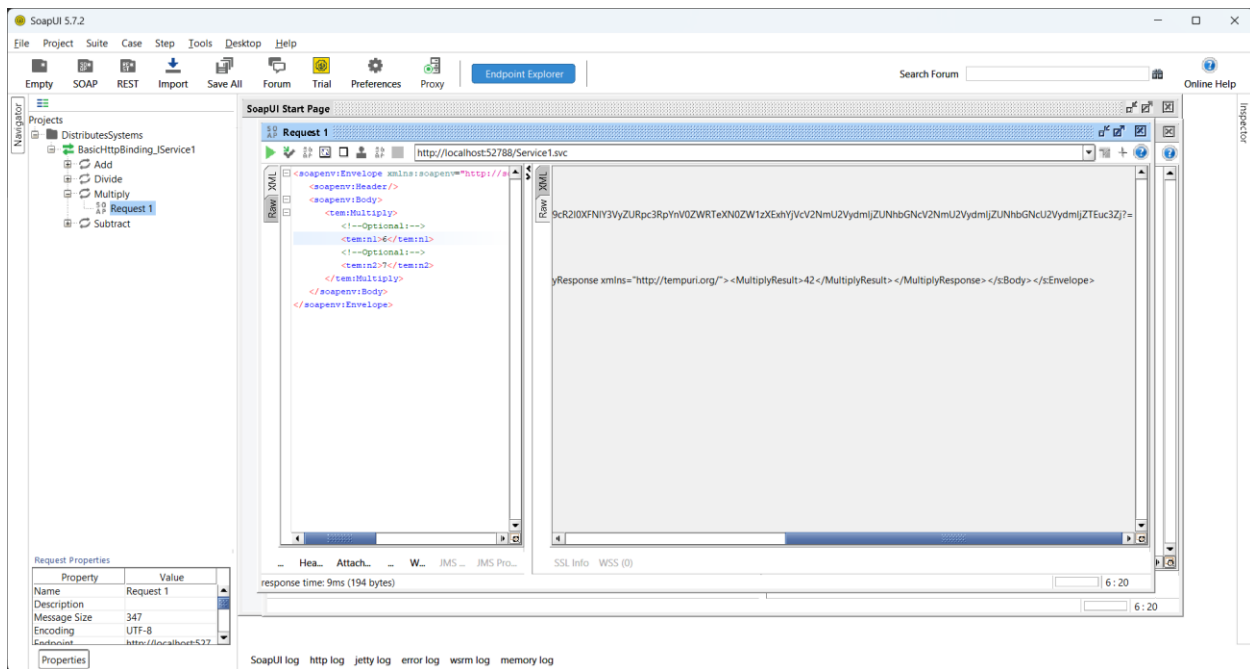


Рисунок 9 – результат виконання операції множення

Step 3. Exploring and Executing the WCF Service Client

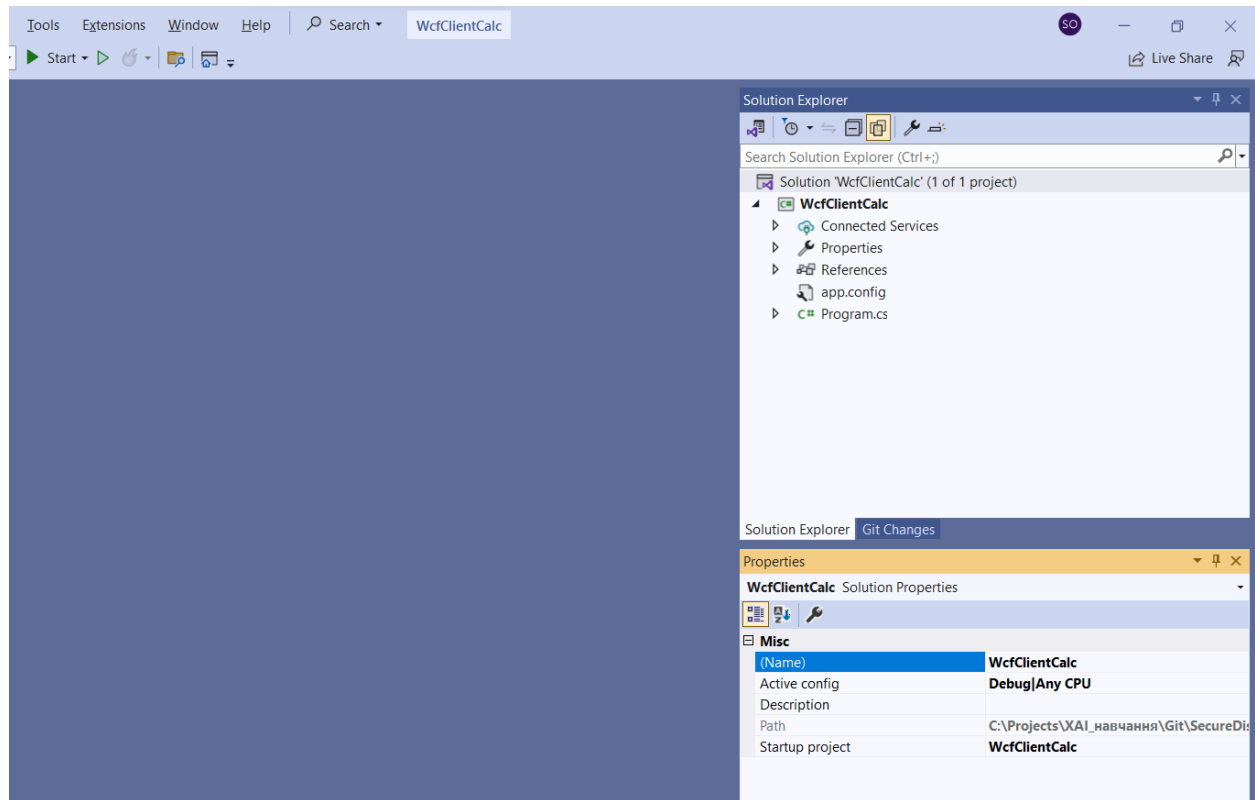


Рисунок 10 - WCF Web Service Client відкритий у Visual Studio

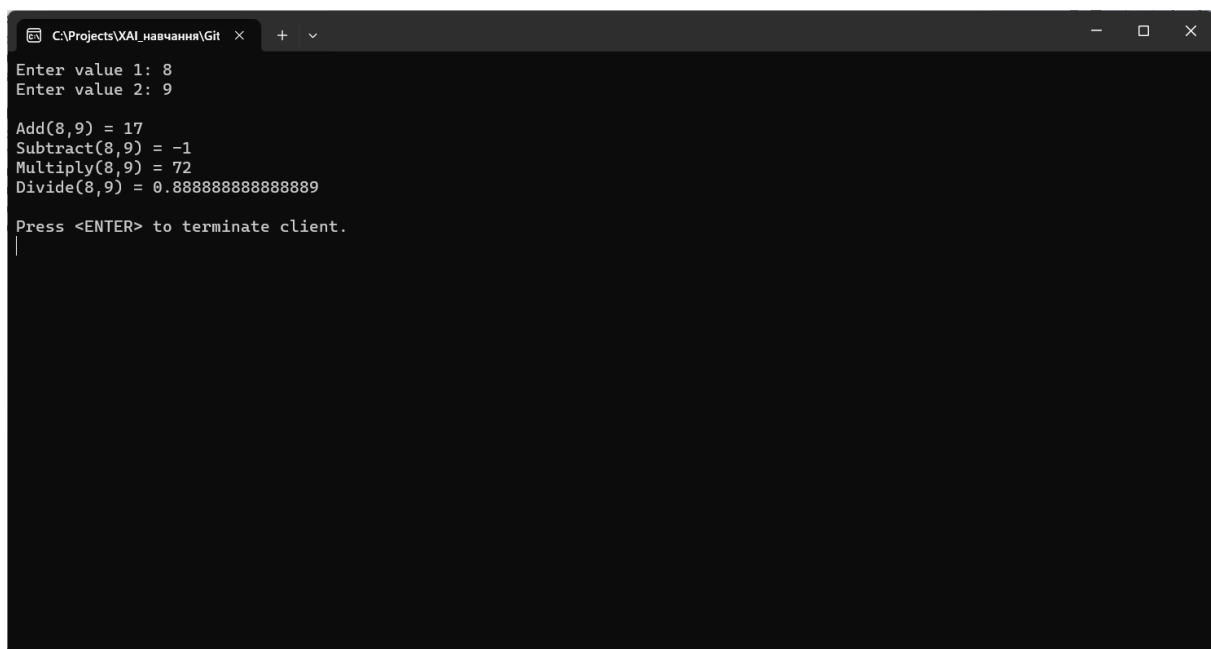


Рисунок 11 – результат виконання WCF Web Service Client

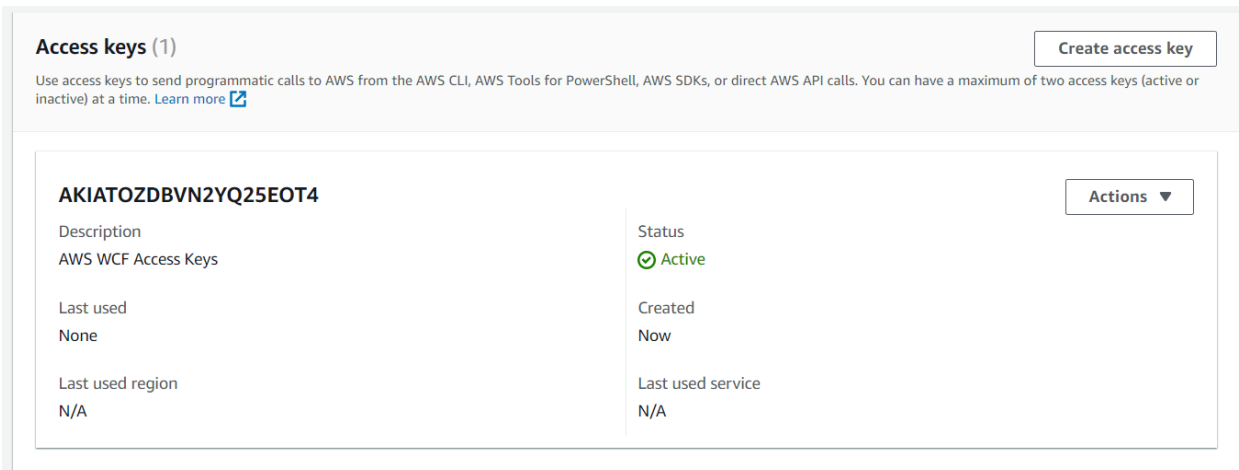


Рисунок 12 – створені ключі доступу через AWS Management Console

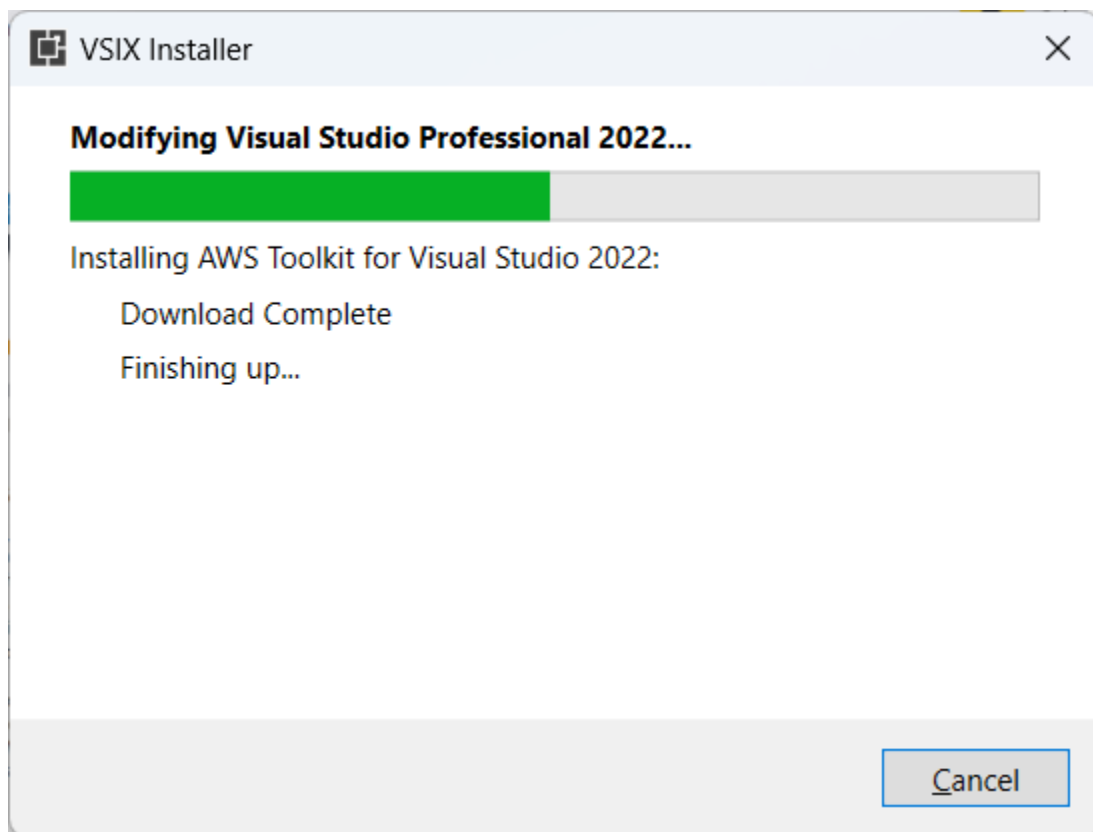


Рисунок 13 – встановлення AWS Toolkit

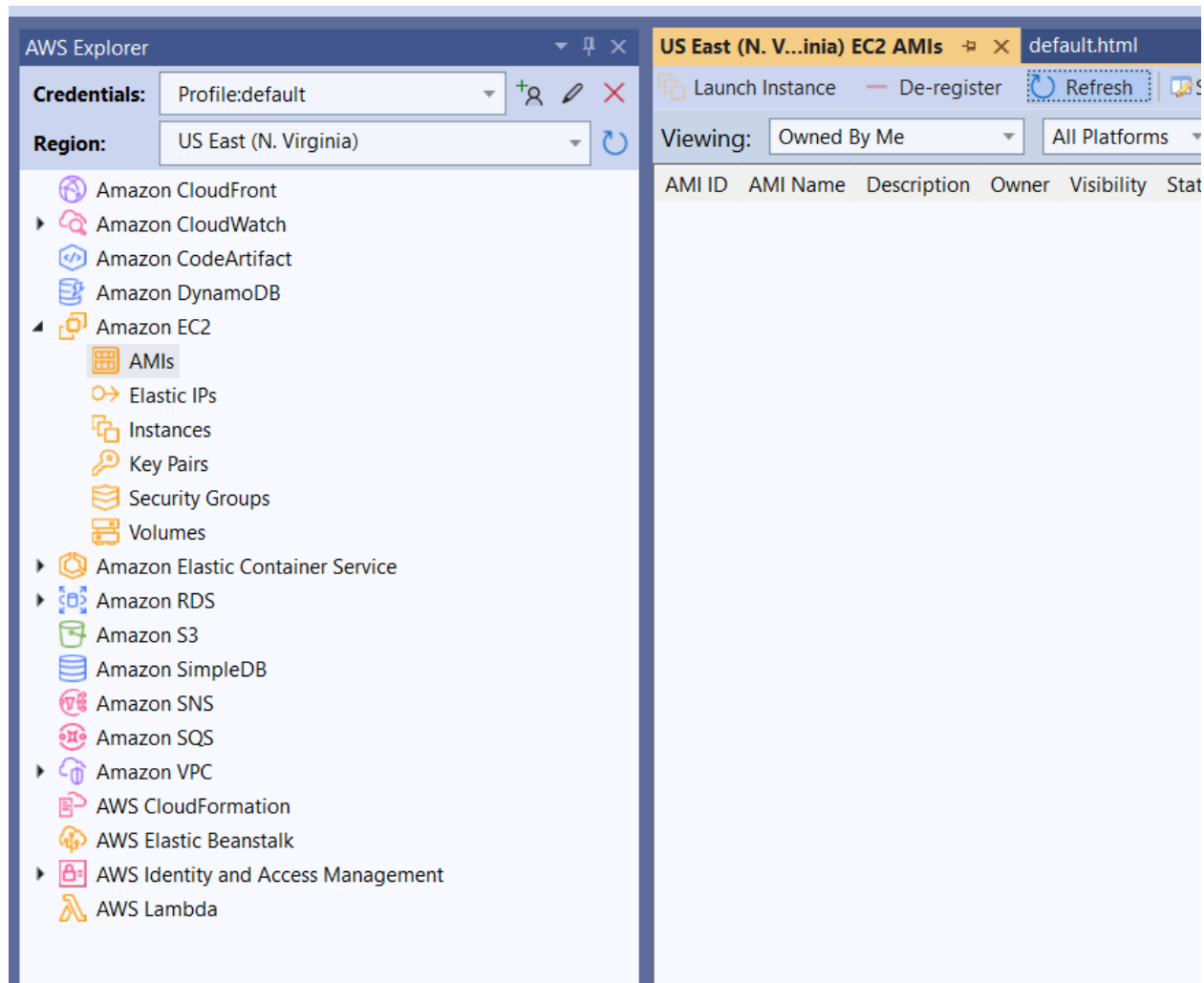


Рисунок 14 – AWS Explorer відкритий у Visual Studio

Publish to Amazon Web Services

aws

Application Environment

Enter the details for your new application environment. To create a new environment for an existing application, select the appropriate application.

Application

Environment

AWS Options

VPC

Updates

Permissions

Options

Review

Application

Name: WcfServiceCalc

Environment

Name: WcfServiceCalc-test

URL

http://wcfservicecalc-test-orlov.elasticbeanstalk.com

Check availability...

✓The requested URL is available

Close

Back

Next

Finish

Publish to Amazon Web Services

aws

AWS

Set Amazon EC2 and other AWS-related options for the deployed application.

Application

Environment

AWS Options

VPC

Updates

Permissions

Options

Review

Amazon EC2 Launch Configuration

Container type *: 64bit Windows Server 2019 v2.13.0 running IIS 10.0

Instance type *: t2.micro

Key pair *: StanislavOrlov-key-pair

Use custom AMI:

☐ Use non-default VPC ☒ Single instance environment ☐ Enable Rolling Deployments

Load balancer type

Choose the type of load balancer to use when not creating a single instance environment.

Application

An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS).

Relational Database Access

Select the Amazon RDS security groups to be modified to permit access from the EC2 instance(s) hosting your application.

Close

Back

Next

Finish

Publish to Amazon Web Services

aws

Application Options

Set additional build and deployment options for your application.

Application

Environment

AWS Options

VPC

Updates

Permissions

Options

Review

Build and IIS Deployment Settings

Project build configuration: Debug|Any CPU

App pool runtime: 4.0 .NET Runtime

☐ Enable 32-bit applications

App path: Default Web Site/

AWS Elastic Beanstalk Environment Options

☐ Enable AWS X-Ray Tracing Support

[Learn More.](#)

☐ Enable Enhanced Health Reporting

[Learn More.](#)

Application Settings

Health check URL: The status of the single EC2 instance is used to determine environment health.

Deployment version label: v20231209193855

Key	Value
-----	-------

CloseBackNextFinish

Publish to Amazon Web Services

aws

Review

Review the information below, then click Finish to start deployment.

Application

Environment

AWS Options

VPC

Updates

Permissions

Options

Review

Profile

Publish to AWS Elastic Beanstalk in region 'US East (N. Virginia)' (us-east-1) using account credentials from profile 'default'.

Application

Create a new application 'WcfServiceCalc' with environment 'WcfServiceCalc-test'.

Use CNAME 'wcfservicecalc-test-orlov' for environment.

(The application will be accessible at http://wcfservicecalc-test-orlov.elasticbeanstalk.com.)

AWS Options

Create a single instance environment using container '64bit Windows Server 2019 v2.13.0 running IIS 10.0', with instance type 'General Purpose Burstable Micro' (t2.micro).

Use the default AMI for the container.

Use existing key pair 'StanislavOrlov-key-pair' when launching EC2 instances.

The application will launch in the default VPC 'vpc-00aa9ff9e57f9cab3'

Permissions

☒ Open environment status window when wizard closes.

CloseBackNextDeploy

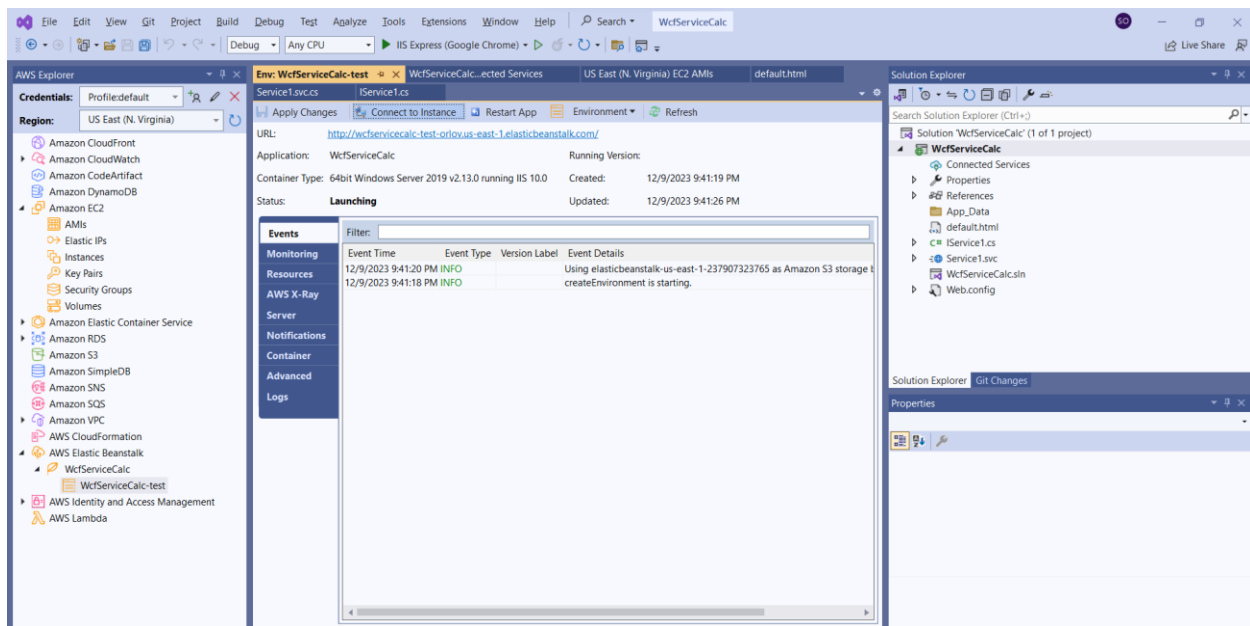


Рисунок 19 – початок публікації WCF сервісу на Amazon EC2 instance

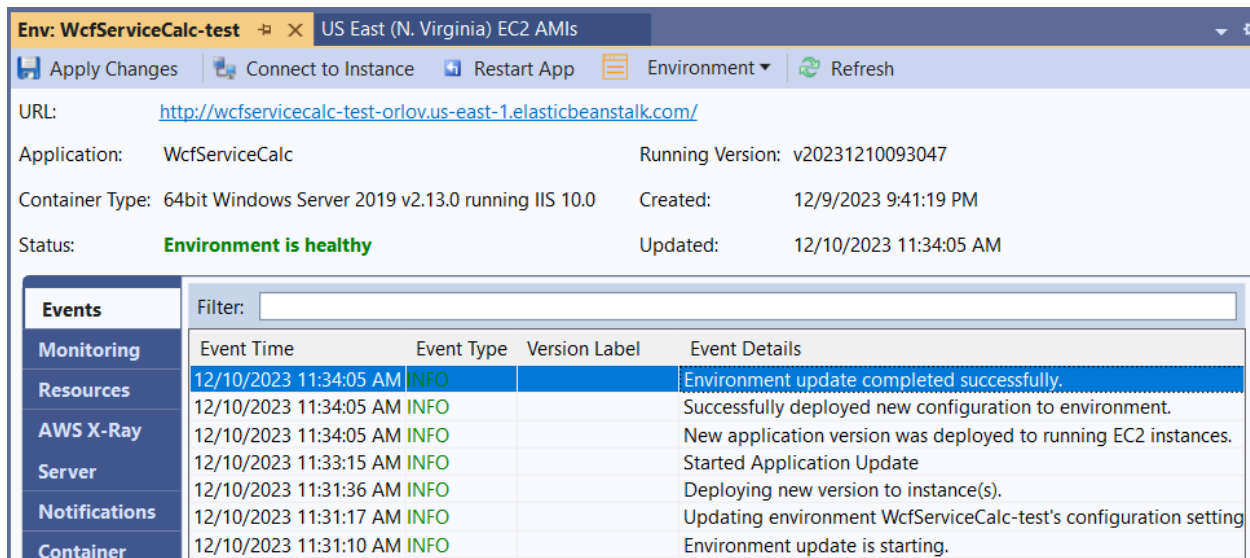
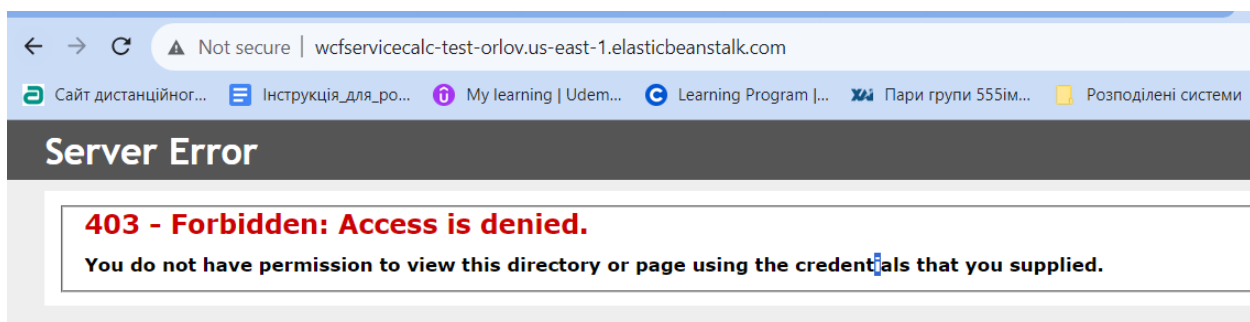


Рисунок 20 – завершення публікації WCF сервісу



Created a new Host service per latest Amazon recommendation.

<https://aws.amazon.com/blogs/modernizing-with-aws/migrate-wcf-apps-to-wcfcore-using-porting-assistant/>

```
namespace WCFSelfHostTest
{
    0 references | 0 changes | 0 authors, 0 changes
    class Program
    {
        0 references | 0 changes | 0 authors, 0 changes
        static void Main(string[] args)
        {
            // Create the ServiceHost.
            using (ServiceHost host = new ServiceHost(typeof(Service1)))
            {
                // Open the ServiceHost to start listening for messages. Since
                // no endpoints are explicitly configured, the runtime will create
                // one endpoint per base address for each service contract implemented
                // by the service.
                host.Open();

                Console.WriteLine("The service is ready at {0}", host.BaseAddresses[0]);

                Console.WriteLine("Press <Enter> to stop the service.");
                Console.ReadLine();

                // Close the ServiceHost.
                host.Close();
            }
        }
    }
}
```

Рисунок 21 – оновлений WCF сервіс по рекомендаціям Amazon

Updated Web.Config service model endpoint mappings:

```
<?xml version='1.0'?>
<configuration>
  <system.web>
    <system.serviceModel>
      <services>
        <service name="WcfServiceCalc.Service1">
          <host>
            <baseAddresses>
              <add baseAddress="http://localhost:52788/Service1/" />
            </baseAddresses>
          </host>
          <endpoint address=""
            binding="wsHttpBinding"
            contract="WcfServiceCalc.IService1" />
          <endpoint address="/Add"
            binding="wsHttpBinding"
            contract="WcfServiceCalc.IService1" />
          <endpoint address="/Subtract"
            binding="wsHttpBinding"
            contract="WcfServiceCalc.IService1" />
          <endpoint address="/Multiply"
            binding="wsHttpBinding"
            contract="WcfServiceCalc.IService1" />
          <endpoint address="/Divide"
            binding="wsHttpBinding"
            contract="WcfServiceCalc.IService1" />
        </service>
      </services>
      <behaviors>
        <serviceBehaviors>

```

Рисунок 22 – оновлений web.config по рекомендаціям Amazon

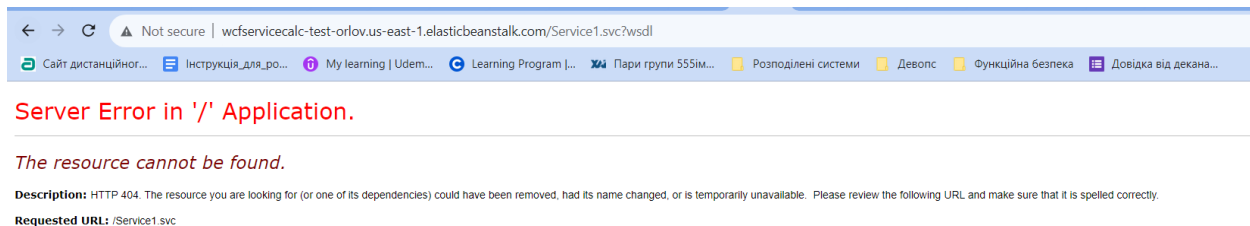
Application is up and running:



The screenshot shows the AWS Management Console for the application 'WcfServiceCalc'. It displays a table of environments with columns for Environment name, Health, Date created, Domain, Running version, Platform, and Platform version. The environment 'WcfServiceCalc-test-orlov.us-east-1' is shown with a 'Green' health status, created on December 9, 2023, with domain 'wcfservicecalc-test-orlov.us-east-1', version 'v202312101043...', and platform 'IIS 10.0 running on Amazon Linux 2'. A 'Create new environment' button is visible in the top right.

Environment ...	Health	Date created	Domain	Running vers...	Platform	Platform
WcfServiceCalc-test-orlov.us-east-1	Green	December 9, 2023	wcfservicecalc-test-orlov.us-east-1	v202312101043...	IIS 10.0 running on Amazon Linux 2	Supporting...

But still not able to access it via Browser not SoapUI:



Посилання на Git репозиторій з завантаженням репозиторієм

[SecureDistibutedSystems/Lab5/Lab5_md/Lab5.md at master · stanislvorlov/SecureDistibutedSystems \(github.com\)](https://github.com/stanislvorlov/SecureDistibutedSystems/blob/master/Lab5/Lab5_md/Lab5.md)

Оновлений WCF сервіс

[SecureDistibutedSystems/Lab5/WcfServiceCalc at master · stanislvorlov/SecureDistibutedSystems \(github.com\)](https://github.com/stanislvorlov/SecureDistibutedSystems/blob/master/Lab5/WcfServiceCalc)

Оновлений клієнт

[SecureDistibutedSystems/Lab5/WcfClientCalc at master · stanislvorlov/SecureDistibutedSystems \(github.com\)](https://github.com/stanislvorlov/SecureDistibutedSystems/blob/master/Lab5/WcfClientCalc)

Висновки:

У ході виконання лабораторної роботи ознайомився з Wcf Service, навчився запускати його локально та тестувати за допомогою Soap утиліт. Сервіс було розгорнуто у середовищі Amazon AWS Elastic Beanstalk.