Exam : **070-487**

Title : Developing Windows Azure and Web Services

Version : DEMO

1.Topic 1, Flight Information

Background

You are developing a flight information consolidation service. The service retrieves flight information from a number of sources and combines them into a single data set. The consolidated flight information is stored in a SQL Server database. Customers can query and retrieve the data by using a REST API provided by the service.

The service also offers access to historical flight information. The historical flight information can be filtered and queried in an ad hoc manner.

The service runs on a Windows Azure Web Role. SSL is not used.

Business Requirements

A new data source for historical flight information is being developed by a contractor located on another continent.

If a time zone is not specified, then it should be interpreted as Coordinated Universal Time (UTC).

When you upgrade a service from a staging deployment to a production deployment, the time that the service is unavailable must be minimized.

The default port must be used for HTTP.

Technical Requirements

The existing sources of flight information and the mechanism of exchange are listed below.

Blue Yonder Airlines provides flight information in an XML file.

Consolidated Messenger provides flight information in a Microsoft Access database that is uploaded every 12 hours to the service using SFTP. The company uses port 22 for SFTP.

Margie's Travel provides and consumes flight information using serialized ADO.NET DataSets. Data is periodically synced between the service and Margie's Travel.

Trey Research provides data from multiple sources serialized in proprietary binary formats. The data must be read by using .NET assemblies provided by Trey Research. The assemblies use a common set of dependencies. The current version of the Trey Research assemblies is 1.2.0.0. All assemblies provided by Trey Research are signed with a key pair contained in a file named Trey.snk, which Trey Research also supplies.

The application specification requires that any third-party assemblies must have strong names.

Application Structure

FlightInfo.cs

```
public class FlightInfo
{
  string DataSource { get; set; }
  public string Airline { get; set; }
  public string Flight { get; set; }
  public DateTimeOffset Arrival { get; set; }
  public int Seats { get; set; }
  public bool WasLate { get; set; }
}
```

BlueYonderLoader.cs

```
public class BlueYonderLoader
{
  public IEnumerable<RawFlightData> LoadFlights(XDocument feed)
  {
    ...
  }

  private RawFlightData Parse(XElement flightElement)
  {
    ...
  }
}
```

HistoricalDataLoader.cs

```
public class HistoricalDataLoader
{
  public static IEnumerable<HistoricalFlightInfo> LoadHistoricalFlights()
  {
    ...
  }

  public void StreamHistoricalFlights(XmlWriter responseWriter, string airline)
  {

    ...
  }

  private XElement ConvertToHistoricalFlight(XElement flight)
  {
    return new XElement("Flight", flight);
  }

  private string GetAirline(XElement flightName)
  {
    return flightName.Value.Substring(0, 2);
  }

  IEnumerable<XElement> RemoteDataStream()
  {
    return XDocument.Load("").Elements();
  }
}
```

MargiesTravelSync.cs

```
public class MargiesTravelSync
{
  public void Sync()
  {
    ...
  }

  private DataSet LoadLocal()
  {
    var dataSet = new DataSet();
    dataSet.ReadXml("local");
    return dataSet;
  }

  private StreamWriter SendStream()
  {
    return new StreamWriter("SendStream");
  }

  private StreamReader ReceiveStream()
  {
    return new StreamReader("ReceiveStream");
  }
}
```

FlightInfoContext.cs

```
public class FlightInfoContext : DbContext
{
  public DbSet<FlightInfo> FlightInfo { get; set; }

  public override int SaveChanges()
  {
    return base.SaveChanges();
  }

  private bool IsTransient(int ex)
  {
    var errors = new[] { 10053, 10054, 64 };
    return errors.Contains(ex);
  }
}
```

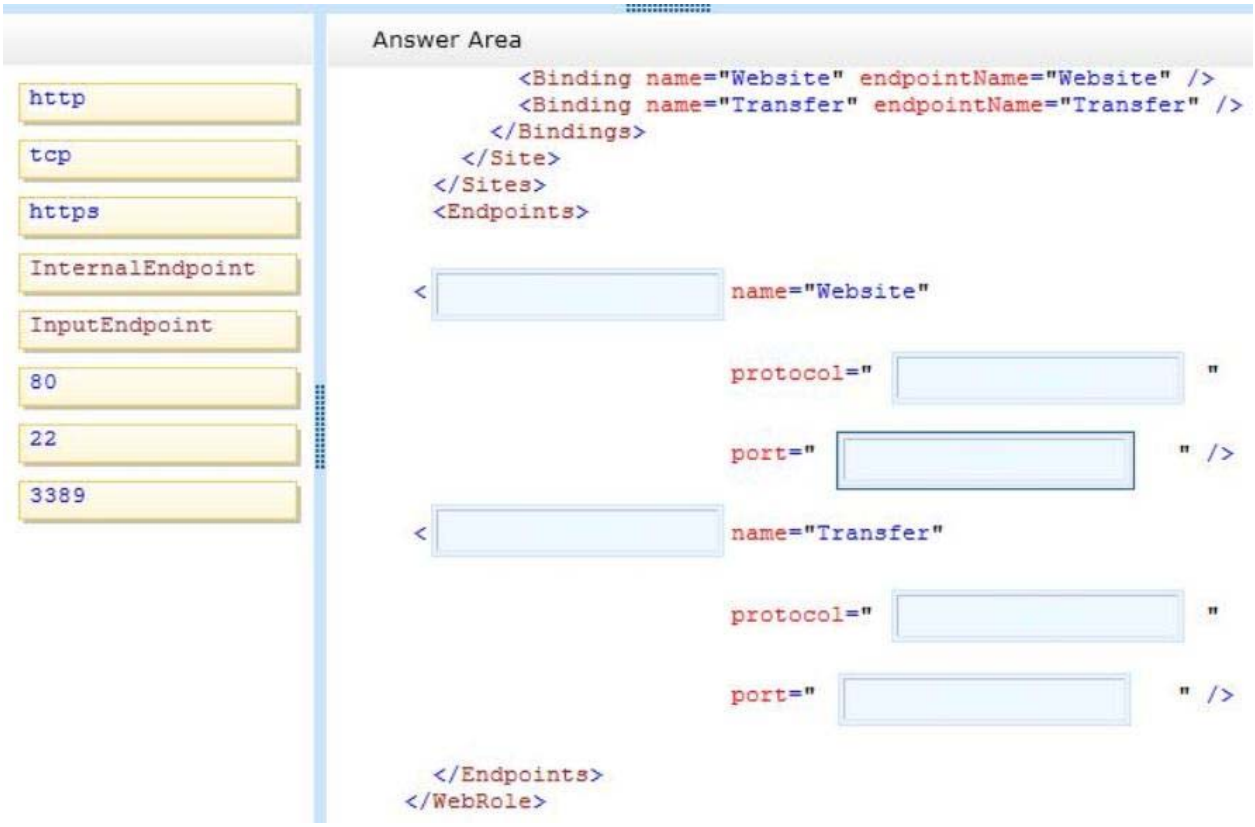
FlightDataController.cs

```
public class FlightDataController : ApiController
{
  FlightInfoContext _Context;

  public FlightDataController()
  {
    _Context = new FlightInfoContext();
  }

  [HttpGet]
  public IEnumerable<FlightInfo> GetFlightInfo()
  {
    return _Context.FlightInfo.Select(x => x).AsEnumerable();
  }

  private IEnumerable<HistoricalFlightInfo> LoadHistorical()
  {
    return HistoricalDataLoader.LoadHistoricalFlights();
  }
}
```

DRAG DROP

You need to configure the Windows Azure service definition to enable Consolidated Messenger to upload files.

What should you do? (To answer, drag the appropriate configuration items to the correct location or locations. Each configuration item may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

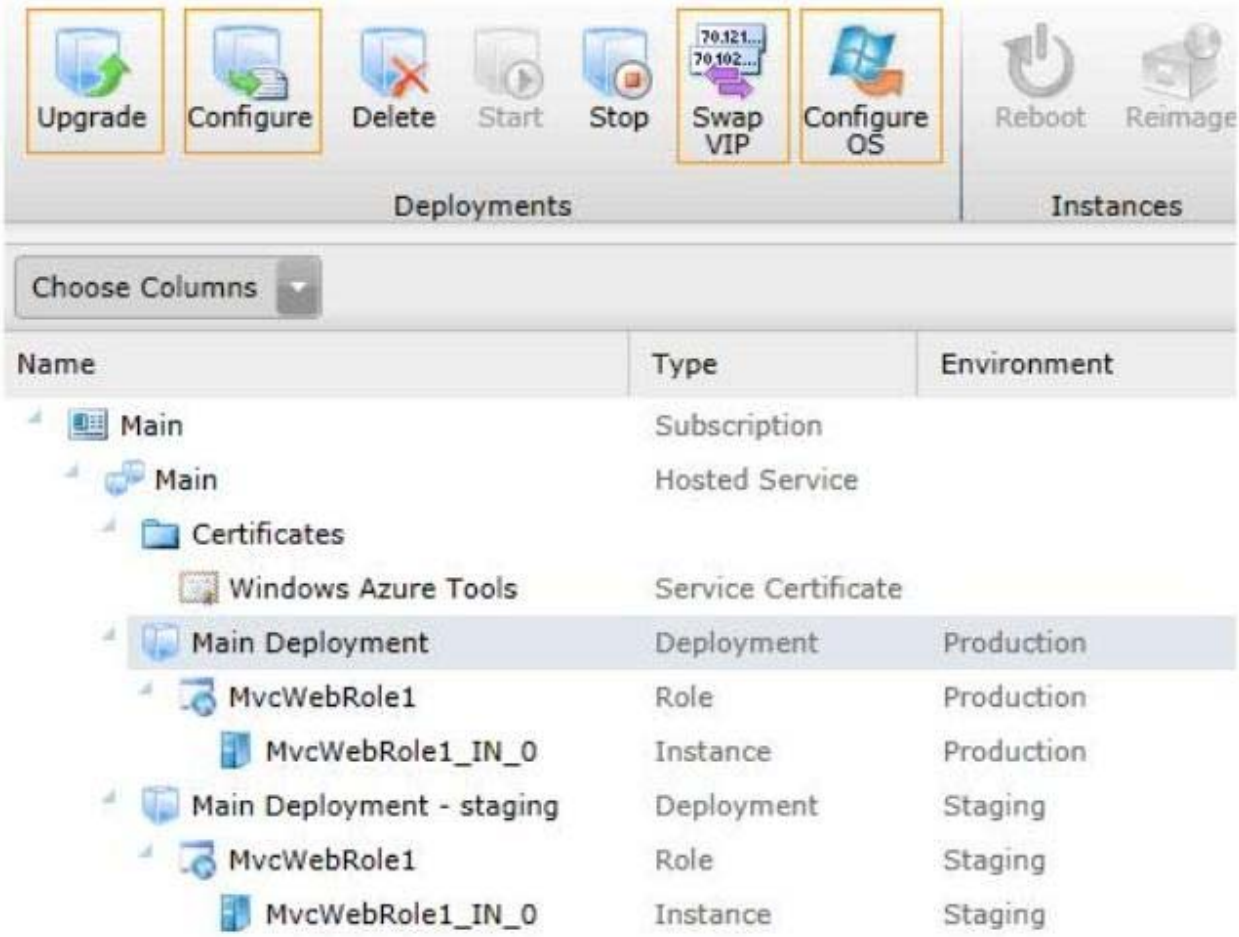


Answer:

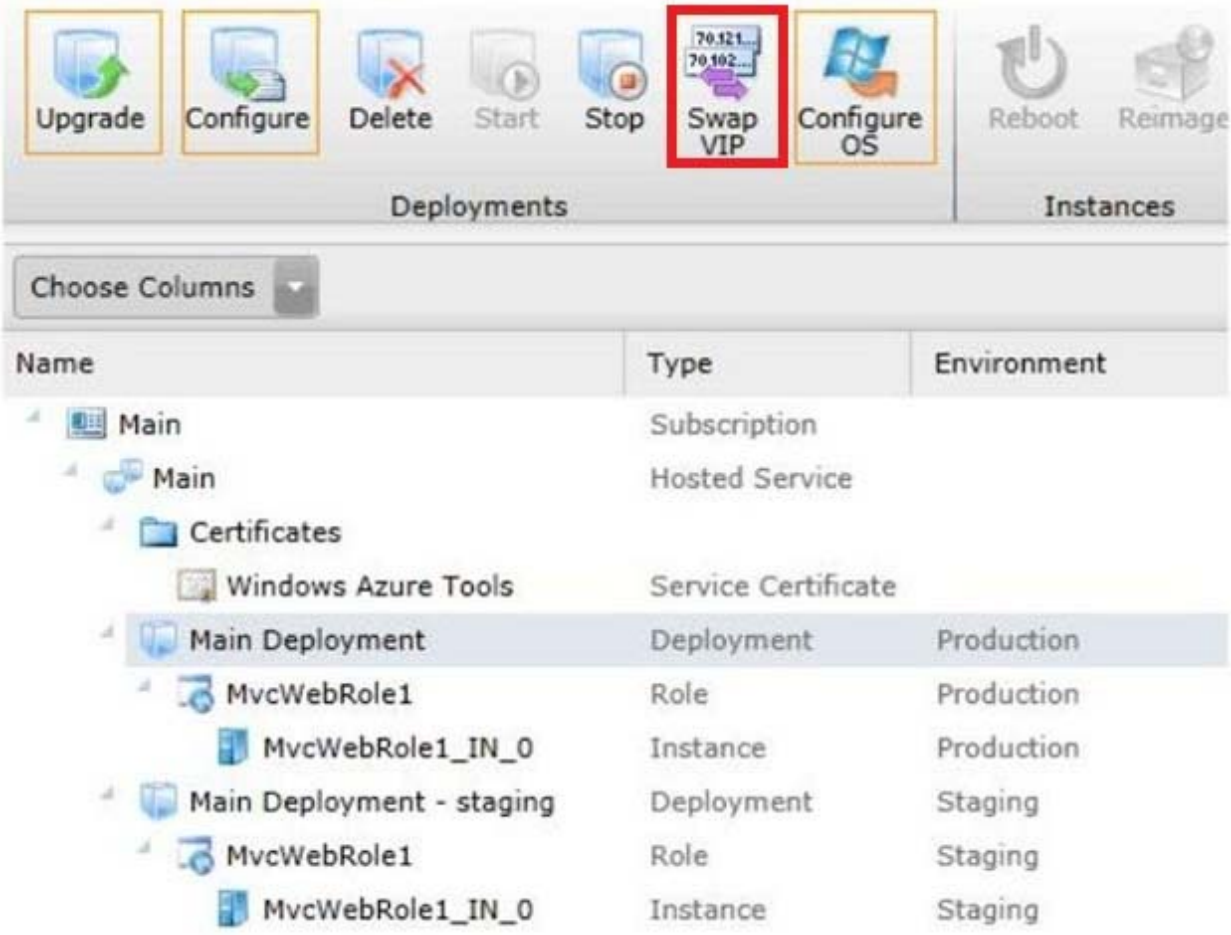
2.HOTSPOT

You need to deploy the application to the Windows Azure production environment to meet the business requirements.

What should you do? (To answer, select the appropriate button in the answer area.)

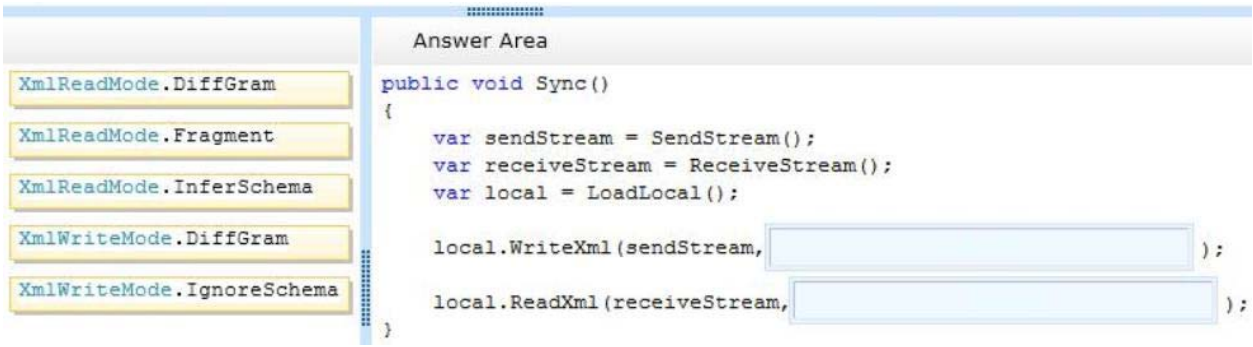


Answer:



3.You need to recommend a data access technology to the contractor to retrieve data from the new data source.
Which data access technology should you recommend?
A. LINQ to XML
B. ADO.NET Entity Framework
C. ADO.NET DataSets
D. WCF Data Services
Answer: D

4.DRAG DROP
Flight information data provided by Margie's Travel is updated both locally and remotely. When the data is synced, all changes need to be merged together without causing any data loss or corruption.
You need to implement the Sync() method in the MargiesTravelSync.es file.
What should you do? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)



Answer:

Explanation:
http://msdn.microsoft.com/en-us/library/ms135424.aspx

5.DRAG DROP
Historical flight information data will be stored in Windows Azure Table Storage using the FlightInfo class as the table entity.
There are millions of entries in the table. Queries for historical flight information specify a set of airlines to search and whether the query should return only late flights. Results should be ordered by flight name.
You need to specify which properties of the FlightInfo class should be used at the partition and row keys to ensure that query results are returned as quickly as possible.
What should you do? (To answer, drag the appropriate properties to the correct location or locations in the answer area. Each property may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)

Answer:
Airline

Flight