Microsoft

AZ-300T04 Module 04 - Understanding Azure Functions

Ahmad Majeed Zahoory

Module 04 - Understanding Azure Functions

Lesson 01: Azure Functions overview

Azure Functions

- Solution for running small pieces of code, or "functions," in the cloud:
  - Write only code that is relevant to business logic
  - Removes the necessity to write “plumbing” code to connect or host application components
- Build on open-source WebJobs code
- Supports a wide variety of programming languages, for instance:
  - C#
  - Node.js
  - Java
  - PHP
  - Python
- Even supports scripting languages, such as:
  - Bash
  - PowerShell

Event-Based Triggers

- Azure Functions, features no-code triggers that can invoke a function based on changes in the following services:
  - Azure
    - Storage Blobs
    - Cosmos DB
    - Storage Tables
    - Mobile Apps
    - Office 365 files
  - Third-Party
    - Twilio
    - SendGrid
    - Many more

Azure Functions scale and hosting concepts

Azure Functions run in two modes:

- Consumption plan
- Azure App Service plan

The Consumption plan automatically allocates compute power when code is running. An app is scaled out when needed to handle load, and scaled down when code is not running. Users don't have to pay for idle VMs or reserve capacity in advance.

When a function app created you choose the hosting plan for functions in the app.

In either plan, an instance of the *Azure Functions host* executes the functions.

The type of plan controls:

- How host instances are scaled out.
- The resources that are available to each host.

Optimize the performance and reliability of Azure Functions

General best practices

The following are best practices in how you build and architect your serverless solutions using Azure Functions:

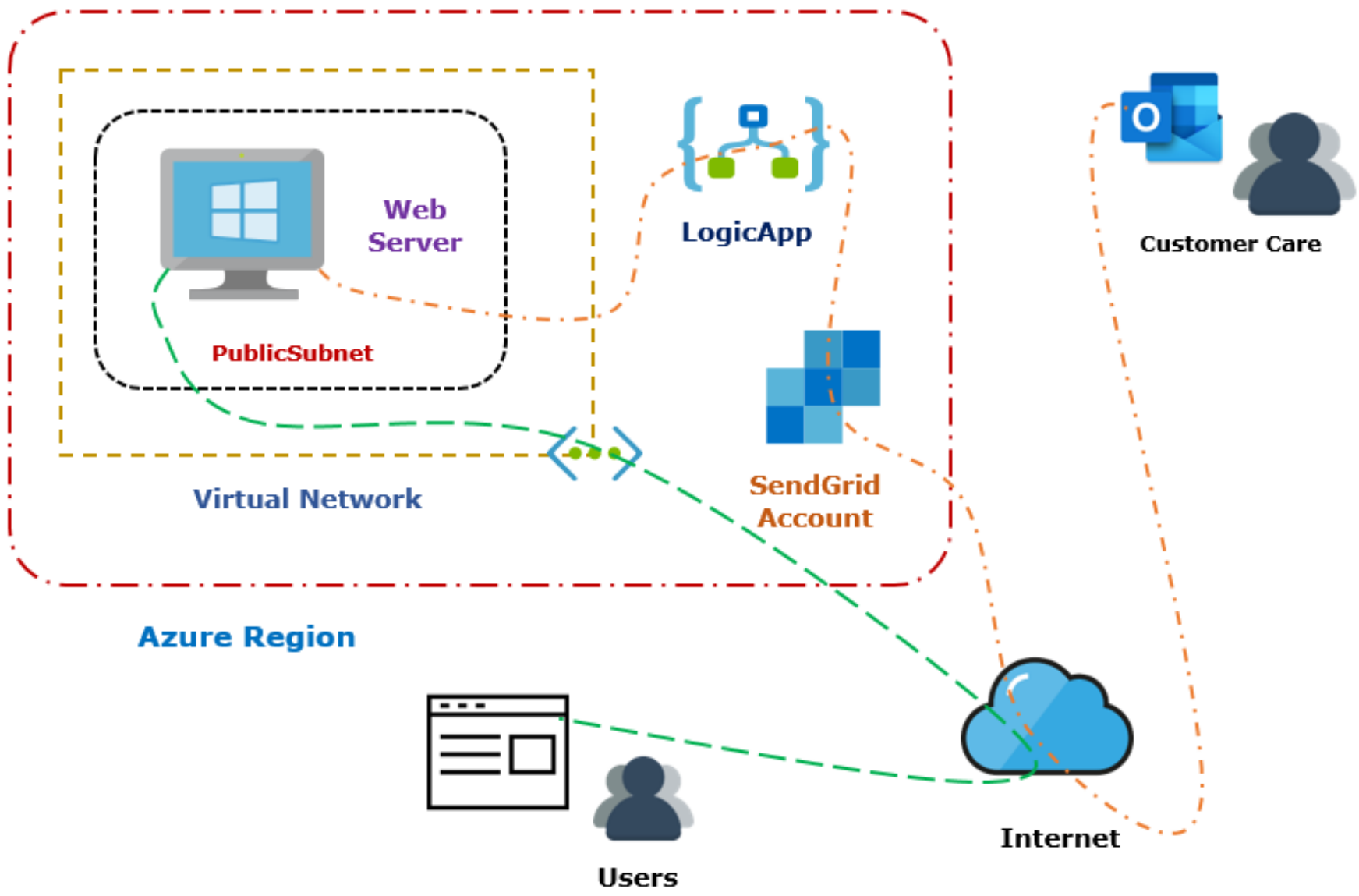
- Avoid long running functions
- Write functions to be stateless

Assume your function could encounter an exception at any time. Design your functions with the ability to continue from a previous fail point during the next execution.

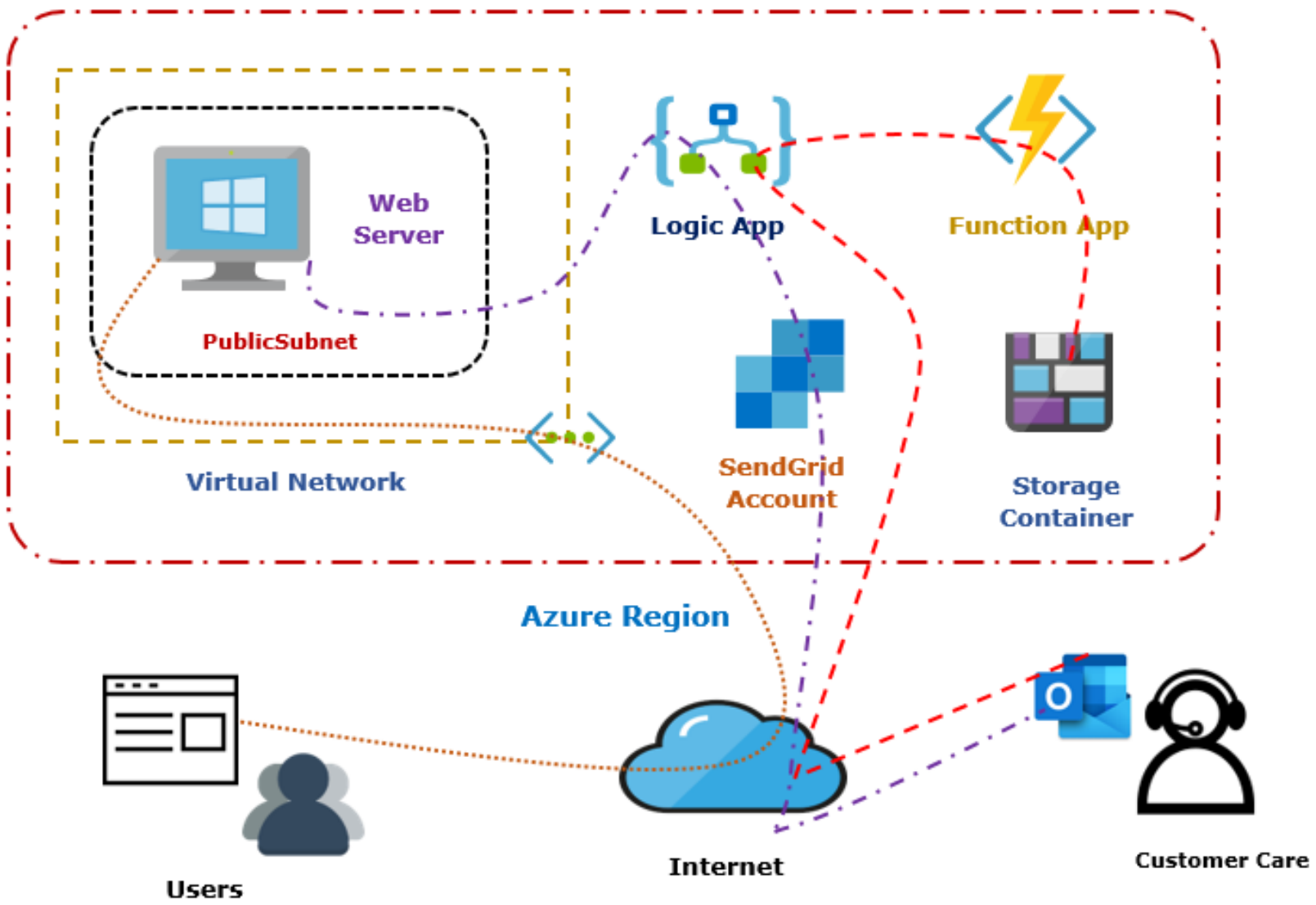
LAB [300TO04-M04-01]

1. Create Business Logic for Website.

Part A


Web
Server
PublicSubnet
LogicApp
Customer Care
Virtual Network
SendGrid
Account
Azure Region
Internet
Users

Part B


Web
Server
PublicSubnet
Logic App
Function App
SendGrid
Account
Storage
Container
Virtual Network
Azure Region
Users
Internet
Customer Care

LAB [300TO04-M04-01]

1. Create Business Logic for Website.
    a. **Services, Tools & Code used**
        i. Azure Virtual Machine
        ii. Azure Logic App
        iii. Azure Function App
        iv. Azure Storage Account
        v. SendGrid
        vi. HTML Code
        vii. Email Box

Duration: 30 mnts.