WebJobs in Azure

CSCI E-94

Fundamentals of Cloud Computing - Azure

Joseph Ficara

Portions © 2013-2025

Agenda

- Web Jobs
  - Overview
  - Configuration
  - Implementation & Execution

Overview

- Logic Apps
  - Serverless & Codeless way to build workflows
    - UI for connecting actions together
  - Usage based billing
    - Per-Action billing
  - Connectors for many common scenarios

Azure	Google	Salesforce
Office 365	IBM 3270	Slack
Twitter	SAP	Cognitive Services
Dropbox	Asana	And more…

Overview

- Web Jobs
  - Are background "jobs" that run
    - Continuously
    - Based on a trigger
      - Example triggers:
        - Integrated service event
          - Queue, Service Bus, Event Hub …
        - Timer
        - Webhook call
        - Manual invocation
  - Run within an App Service
    - Consume resources with Website / API

Overview

Web Jobs …

- Run in Azure App Services
  - Overview of Azure App Service
  - Variety of tech stack choices
    - .cmd, .bat, .exe (using Windows cmd)
    - .ps1 (using PowerShell)
    - .sh (using Bash)
    - .php (using PHP)
    - .py (using Python)
    - .js (using Node.js)
    - .jar (using Java)

Overview

Web Jobs …

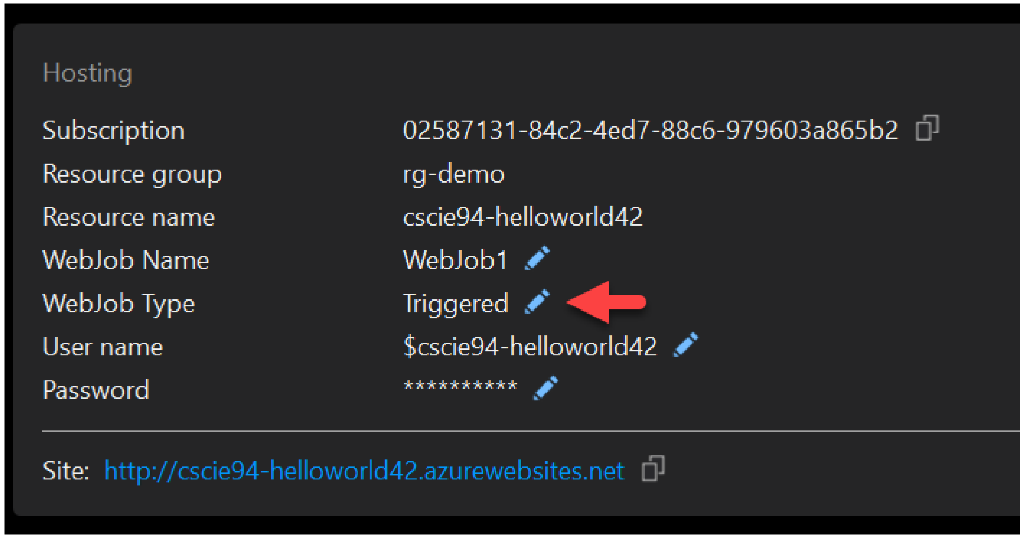
- Scales with App Role (App Service Plan)
- Medium weight background operations
- Integration with Storage
  - Table, Queue & Blob triggered workloads
- Automatic Trigger Support
  - Service Bus
  - Event Hubs
  - CosmosDB
  - SendGrid

Overview
Web Jobs …

- Why do you care?
  - Easy to use
  - Support for many tech stacks
  - Fully managed **PaaS**
  - Helps maximize use of App Service Plan
- How do they work
  - You write Methods
    - Methods are triggered on some event
    - Your code is executed

Overview
Web Jobs …

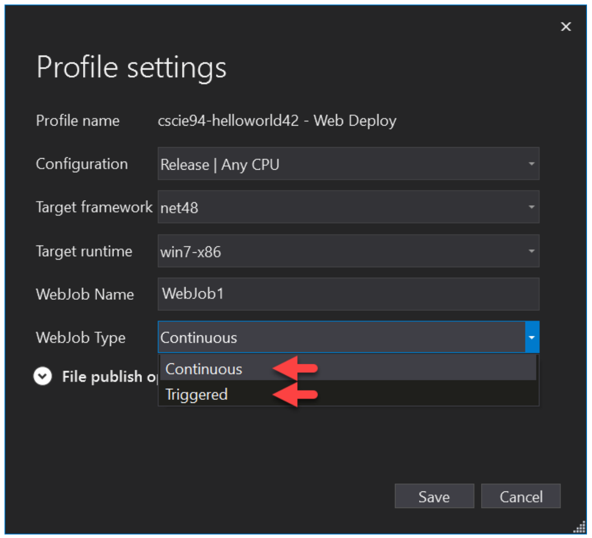
- Execution Options (Webjob Types)
  - Configured via publish settings
    - Triggered, or Continuous



Hosting

Subscription	02587131-84c2-4ed7-88c6-979603a865b2
Resource group	rg-demo
Resource name	cscie94-helloworld42
WebJob Name	WebJob1
WebJob Type	Triggered
User name	$cscie94-helloworld42
Password	**********

Site: http://cscie94-helloworld42.azurewebsites.net



Profile settings

Profile name	cscie94-helloworld42 - Web Deploy
Configuration	Release | Any CPU
Target framework	net48
Target runtime	win7-x86
WebJob Name	WebJob1
WebJob Type	Continuous

Overview

Web Jobs …

- Execution Options (Webjob Types) …
  - There is also a "Scheduled option"
    - **Settings.job** file has examples
      - Requires **Always On** enabled
      - Requires **basic tier or higher**
  - Note:
    - Can't change execution mode in dashboard
      - Edit the **Settings.job** to change schedule
    - To reconfigure execution mode
      - In Visual Studio delete the file
        - properties\webjob-publish-settings.json

Overview

Web Jobs …

- You can run WebJobs locally
  - While connecting to cloud storage account
    - React to automatic triggers
      - Process Queue and/or Blob entries
      - CosmosDB and Event Hub, etc…
      - Run continuously non triggered
  - Need to add connection strings
    - App.config

Overview
Web Jobs …

- To support automatic invocation
  - Run with **continuous** schedule
    - **Basic tier or higher**
  - Automatic triggering setup
    - Via .NET Attributes
      - See the Functions class in Visual Studio Template
        - Functions.cs
- To run in free tier
  - Use REST API to "run" or "start" the webjob
    - Will terminate after execution or abort after a time

Overview

Web Jobs …

- REST API Options:
  - To execute **triggered** / Test in Azure
    REST API See: Github WebJobs API
    - Execute POST call URL consists of
      - The website name with the scm suffix
        - The job resource is located under
          - api/triggeredwebjobs/<web job name>/run
            - The action to run it is run
      - Include the Basic Authorization key in the header
        - Key: Authorization
        - Value: Basic [base 64 encoded credentials]
          - credentials: [user name]:[encrypted password]
      - Include the query string parameter **arguments**
        - POST /api/triggeredwebjobs/{job name}/run?**arguments**={arguments}
          - Presented in WEBJOBS_COMMAND_ARGUMENTS environment variable

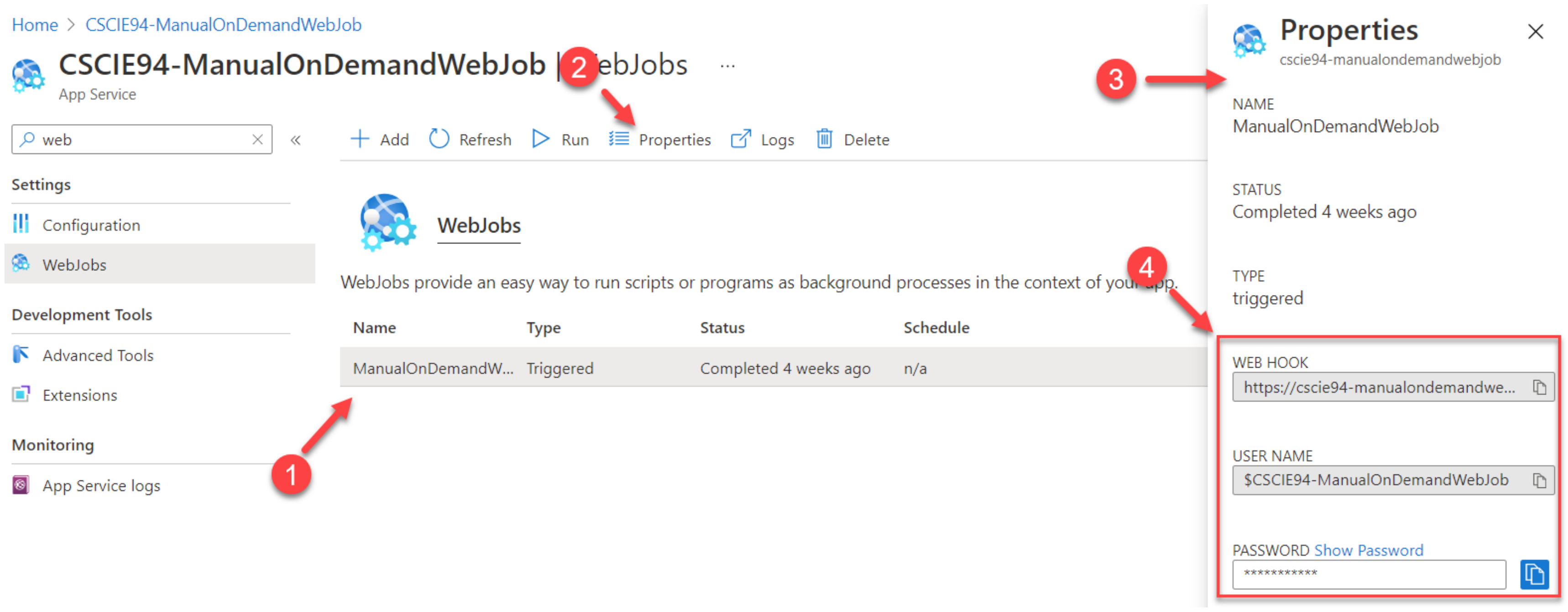
Overview

Web Jobs …

- REST API Options
  - To start **continuous** webjobs in Azure
    REST API See: Github WebJobs API
    - Execute POST call - URL consists of
      - The website name with the scm suffix
        - The job resource is located under
          - api/triggeredwebjobs/<web job name>/start
            - The action to run it is start
      - Include the Basic Authorization key in the header
        - Key: Authorization
        - Value: **Basic [base 64 encoded credentials]**
          - credentials: **[user name]:[encrypted password]**

Overview
Web Jobs …

- Credentials and URL can be obtained here
  - Note: The URL will always have a suffix of start
    - Change it to Run for triggered jobs



WebJobs provide an easy way to run scripts or programs as background processes in the context of your app.

Name	Type	Status	Schedule
ManualOnDemandW...	Triggered	Completed 4 weeks ago	n/a

Properties
cscie94-manualondemandwebjob

NAME
ManualOnDemandWebJob

STATUS
Completed 4 weeks ago

TYPE
triggered

WEB HOOK
https://cscie94-manualondemandwe...

USER NAME
$CSCIE94-ManualOnDemandWebJob

PASSWORD Show Password
****

Overview

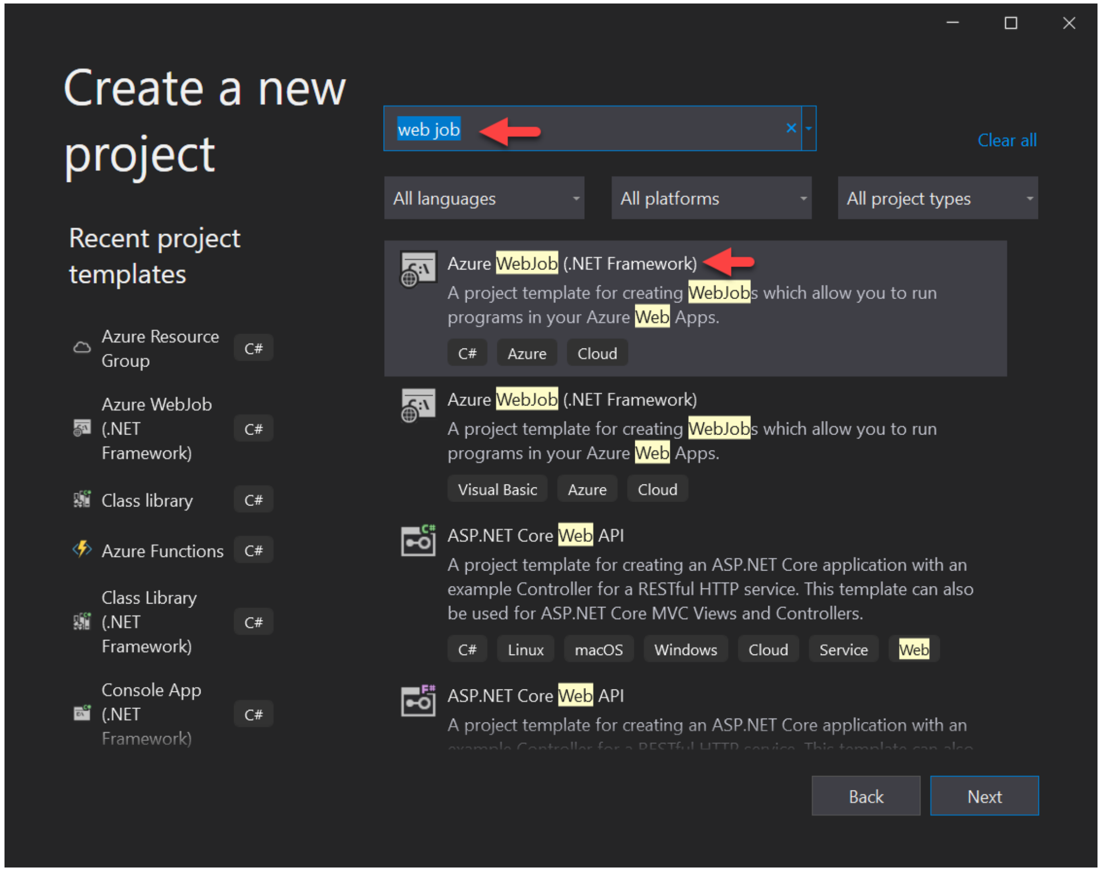
Web Jobs

Web Jobs
Configuration

- Configuration via Visual Studio
  - Create a Web Job Project under Visual Studio
    - Search: Web Job
      - Choose "Azure WebJob (.NET Framework)"
  - This creates a single WebJob
    - With two classes
      - Functions (**`Functions.cs`**)
        - Has a default ProcessQueueMessage method
          - Listens for entries into the storage queue, named queue
      - Program (**`Program.cs`**)
        - Has a Main method
          - That runs the job forever

Web Jobs
Configuration

- Configuration via Visual Studio



Create a new project

web job

Clear all

All languages | All platforms | All project types

Recent project templates

Azure Resource Group	C#
Azure WebJob (.NET Framework)	C#
Class library	C#
Azure Functions	C#
Class Library (.NET Framework)	C#
Console App (.NET Framework)	C#

Azure WebJob (.NET Framework)
A project template for creating WebJobs which allow you to run programs in your Azure Web Apps.
C# · Azure · Cloud

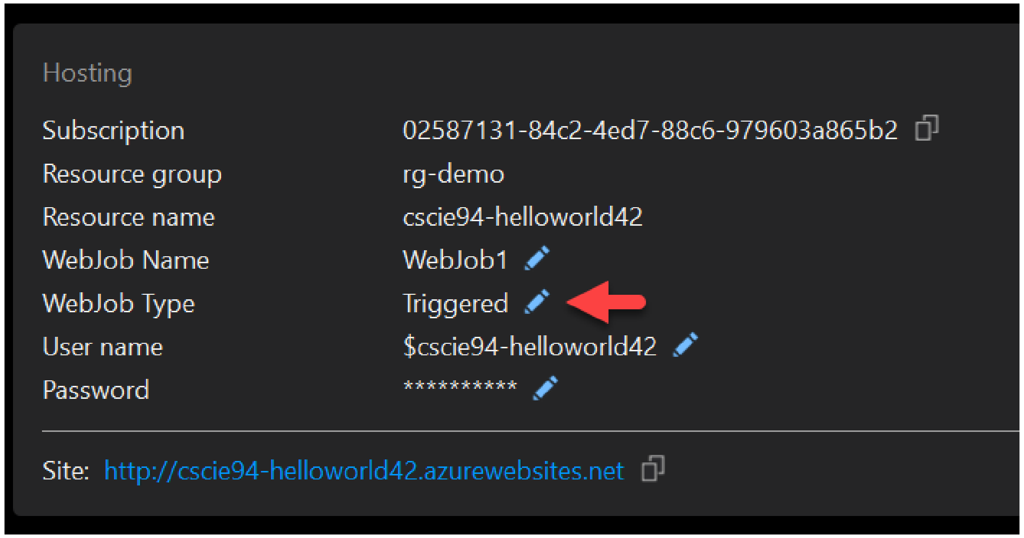
Azure WebJob (.NET Framework)
A project template for creating WebJobs which allow you to run programs in your Azure Web Apps.
Visual Basic · Azure · Cloud

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.
C# · Linux · macOS · Windows · Cloud · Service · Web

ASP.NET Core Web API
A project template for creating an ASP.NET Core application with an

Web Jobs
Configuration …

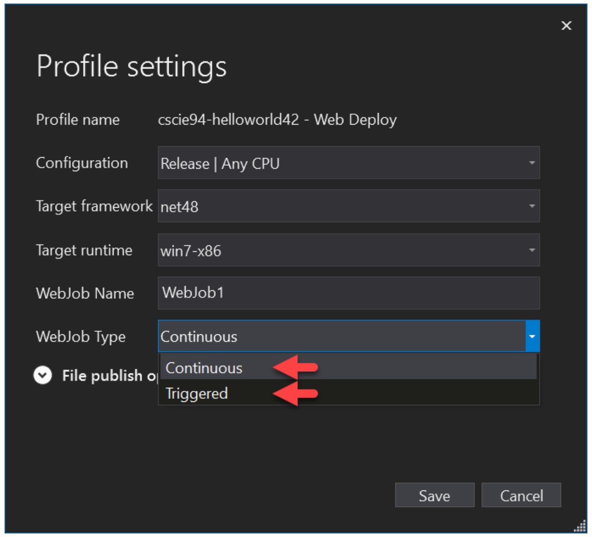
- When publishing to Azure
  - Scheduling options are set in Visual Studio



Hosting

Subscription	02587131-84c2-4ed7-88c6-979603a865b2
Resource group	rg-demo
Resource name	cscie94-helloworld42
WebJob Name	WebJob1
WebJob Type	Triggered
User name	$cscie94-helloworld42
Password	**********

Site: http://cscie94-helloworld42.azurewebsites.net

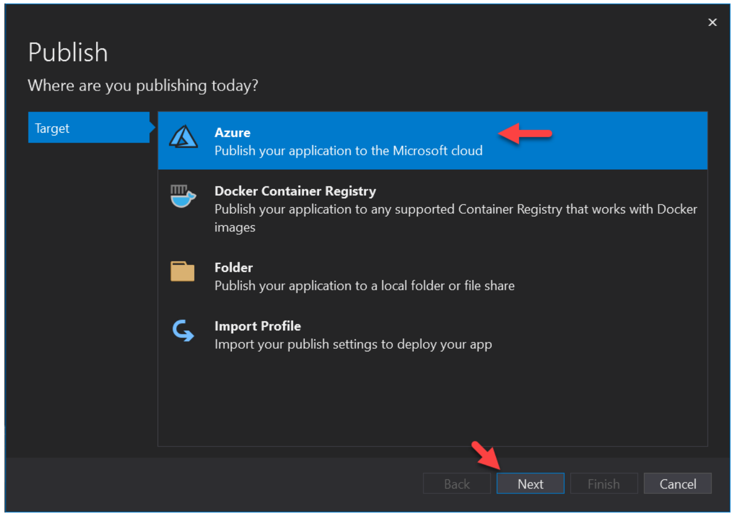


Profile settings

Profile name	cscie94-helloworld42 - Web Deploy
Configuration	Release | Any CPU
Target framework	net48
Target runtime	win7-x86
WebJob Name	WebJob1
WebJob Type	Continuous

Web Jobs
Configuration …

- Next - Choose where the Web Job goes
  - *Published to Azure Websites*
    - Create a new site or publish to an existing site



Publish

Where are you publishing today?

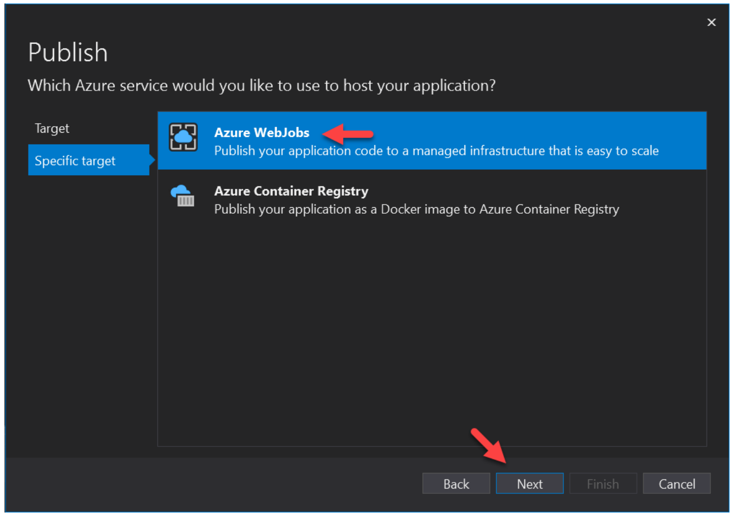
Target

Azure
Publish your application to the Microsoft cloud

Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images

Folder
Publish your application to a local folder or file share

Import Profile
Import your publish settings to deploy your app



Publish

Which Azure service would you like to use to host your application?

Target

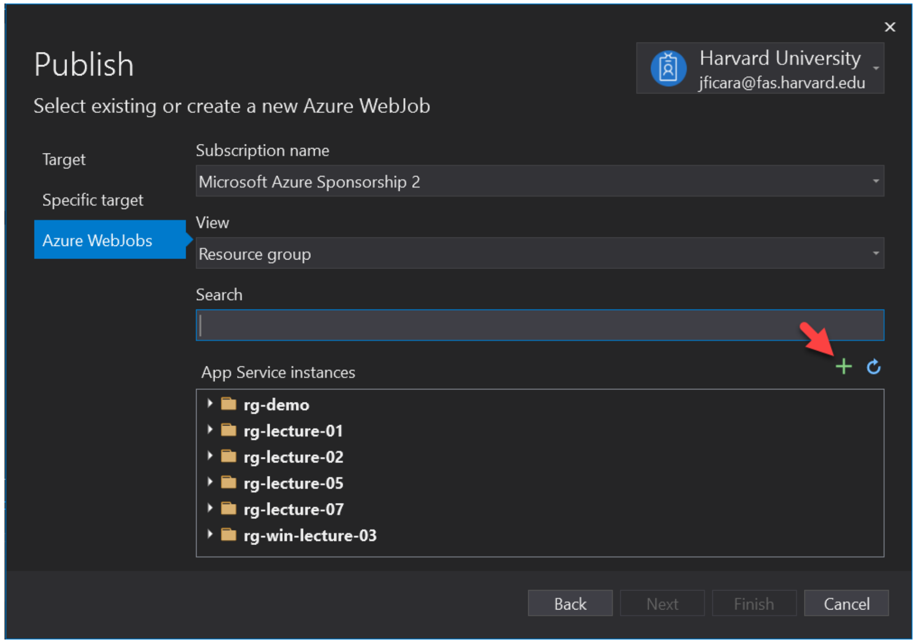
Specific target

Azure WebJobs
Publish your application code to a managed infrastructure that is easy to scale

Azure Container Registry
Publish your application as a Docker image to Azure Container Registry

Web Jobs
Configuration …

- Next - Choose where the Web Job goes
  - *Published to Azure Websites*
    - Create a new site …



Publish

Select existing or create a new Azure WebJob

Harvard University
jficara@fas.harvard.edu

Target
Specific target
Azure WebJobs

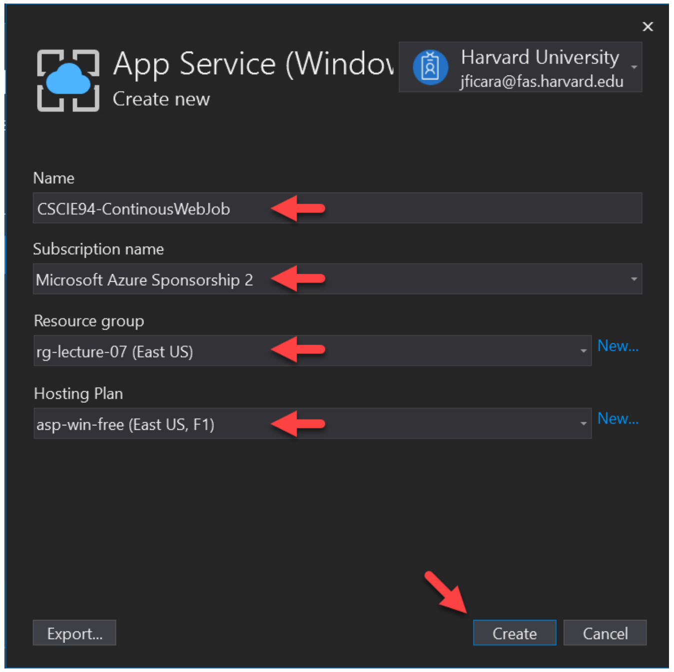
Subscription name: Microsoft Azure Sponsorship 2

View: Resource group

Search:

App Service instances
- rg-demo
- rg-lecture-01
- rg-lecture-02
- rg-lecture-05
- rg-lecture-07
- rg-win-lecture-03

Back | Next | Finish | Cancel



App Service (Windov
Create new

Harvard University
jficara@fas.harvard.edu

Name: CSCIE94-ContinousWebJob

Subscription name: Microsoft Azure Sponsorship 2

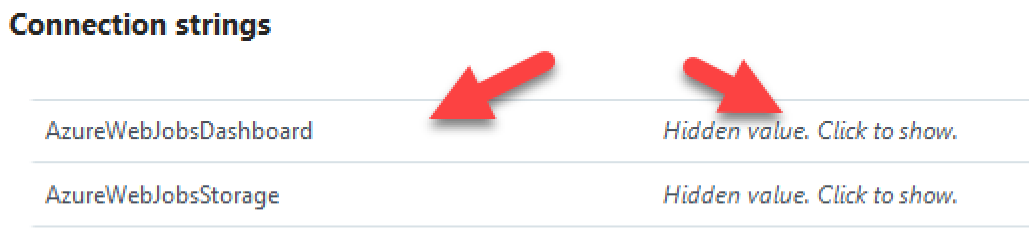
Resource group: rg-lecture-07 (East US) New...

Hosting Plan: asp-win-free (East US, F1) New...

Export... | Create | Cancel

Web Jobs
Configuration …

- Before you can run the Web Job
  - Set these entries to your storage account
    - AzureWebJobsDashboard
    - AzureWebJobsStorage
  - Set under the CONFIGURE tab
    - In the connection strings section



Connection strings

AzureWebJobsDashboard	*Hidden value. Click to show.*
AzureWebJobsStorage	*Hidden value. Click to show.*

    - To run locally set in **app.config**

Web Jobs

Configuration …

- Configuration takes a few more steps …
  - Set the configuration strings to Custom
    - The storage account format looks like this
      - AzureWebJobsDashboard
        - DefaultEndpointsProtocol=https; AccountName=[accountname]; AccountKey=[accesskey]
      - AzureWebJobsStorage
        - DefaultEndpointsProtocol=https; AccountName=[accountname]; AccountKey=[accesskey]

Web Jobs

Configuration …

Web Jobs
Implementation

- Implementation depends
  - Do you want to
    - Run continuously?
      - Implement a producer consumer pattern?
    - Run on a schedule
      - Run triggered, also less expensive
- Function.cs
  - Implements a ProcessQueueMessage
    - Connected to a Queue
      - Can be used to implement Producer Consumer pattern

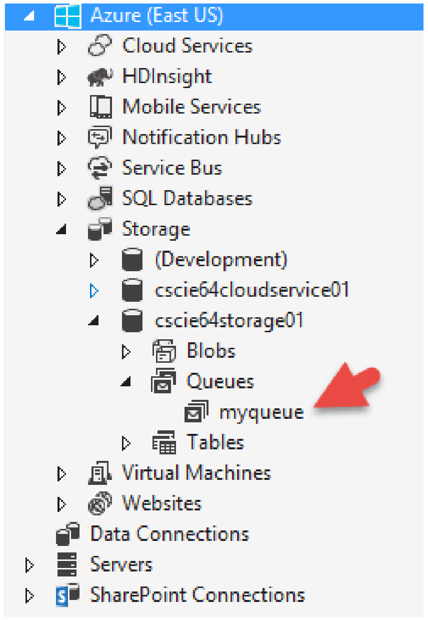
Web Jobs
Implementation …

- Program.cs
  - Calls host.RunAndBlock()
    - **Runs forever**
    - Can be used in continuous mode
  - What about OnDemand / Manual Trigger?
    - Don't call host.RunAndBlock()
      - Unless you want to start and run forever
    - Add calls in Main
      - To invoke methods to execute triggered
      - Use **Logic Apps** to schedule execution

Web Jobs

Continuous Mode: Monitoring myqueue for messages – Function.cs

```
public class Functions
{
  // This function will get triggered/executed when a new message is written
  // on an Azure Queue called queue.
  public static void ProcessQueueMessage([QueueTrigger("myqueue")]
                                          string message, TextWriter log)
  {
       log.WriteLine(message);
  }
}
```



Web Jobs

Supporting OnDemand (Manual Trigger) – Program.cs

```
public class void Program // ← Ensure this class is public!
{
   static void Main()
   {
     var c = new JobHost();

     // Executes this call once "triggered"
     host.Call(typeof(Functions).GetMethod("ProcessQueueMessageOnDemand"));

    // Note Don't use this if running triggered
    // Disabled for triggered jobs
    // The following code ensures that the WebJob will
    // be running continuously
    // host.RunAndBlock();
}
```

Web Jobs

Supporting OnDemand (Manual Trigger) – Functions.cs

```
[NoAutomaticTrigger]
public static void ProcessQueueMessageOnDemand(TextWriter log)
{
     CloudQueue queue = GetCloudQueue();
     // Retrieve the message
     var message = queue.GetMessage();
     if (message != null)
     {
         log.WriteLine($"Message: {message.AsString}");
         // Remove the message from the queue
          queue.DeleteMessage(message);
     }
     else
     {
          log.WriteLine("No message found");
     }
}
```

Web Jobs

Implementation …

- Manually Invoking via HTTPS POST call
  - You must base 64 encode the credentials
    - Include them in the Header
      - Include the Authorization key in the header
        - Key:
          - Authorization
        - Value
          - Basic [base 64 encoded credentials]
            credentials: [user name]:[encrypted password]
          - Ex:
            - joe:xkeialzidiqldizaeie
              - Basic am9lOnhrZWlhbHppZGlxbGRpemFlaWUK

Web Jobs

Implementation …

- To run via an HTTPS POST call …
  - The URL consists of
    - The website name with the scm suffix
    - The job resource is located under
      - api/triggeredwebjobs/<web job name>
        - The action to run it is run
  - Ex:

https://e94webjobsapidemo.scm.azurewebsites.net/api/triggeredwebjobs/WebJob1/run

Web Jobs
Implementation …

- Manually invoking the Web Job
  - Via HTTPClient
    - Be sure to do this asynchronously
  - Remember the Web Job may take time
    - Don't make the user wait
      - Ideally notify the user when the Job is complete
  - Can pass data to the Job via the HTTP call
    - Passed using query string parameter
      - POST /api/triggeredwebjobs/{job name}/run?**arguments**={arguments}
      - Available via environment variable
        - WEBJOBS_COMMAND_ARGUMENTS
    - Data can also be in persistent storage
      - A Database Table, Queue, Table Storage etc…

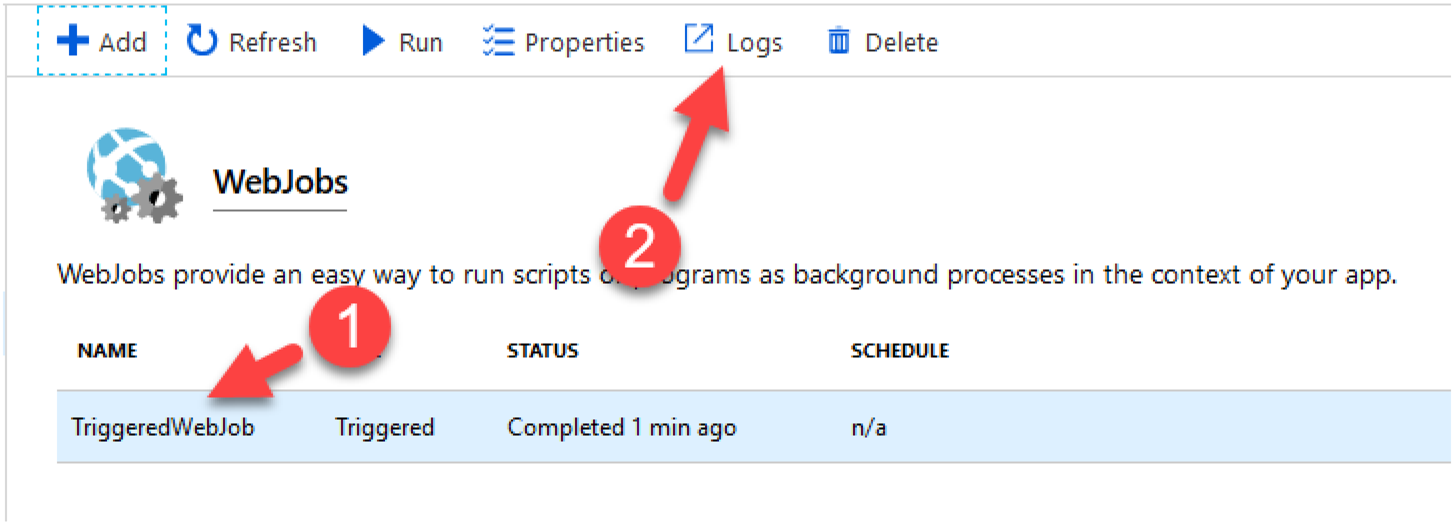
Web Job

Manual Web Job invocation

```
// Create the http client to invoke the job dynamically
HttpClient client = new HttpClient();
string encodedCredentials = "Basic " + Base64Encode(JobCredentials);

// Add the header so the job can be dynamically invoked
client.DefaultRequestHeaders.Add("Authorization", encodedCredentials);

// Retrieve the public facing Url for the mobile service
string webJobUrl= "https://e94webjobsapidemo.scm.azurewebsites.net
/api/triggeredwebjobs/WebJob1/run";

// Invoke the job dynamically
HttpResponseMessage x = await client.PostAsync(new Uri(webJobUrl), null);

// If the request failed log an error
if (!x.IsSuccessStatusCode)
{ ... }
```

Web Jobs
Implementation …

- Logging is available per
  - WebJob
    - Per method within the WebJob
      - Can expand log information per method!

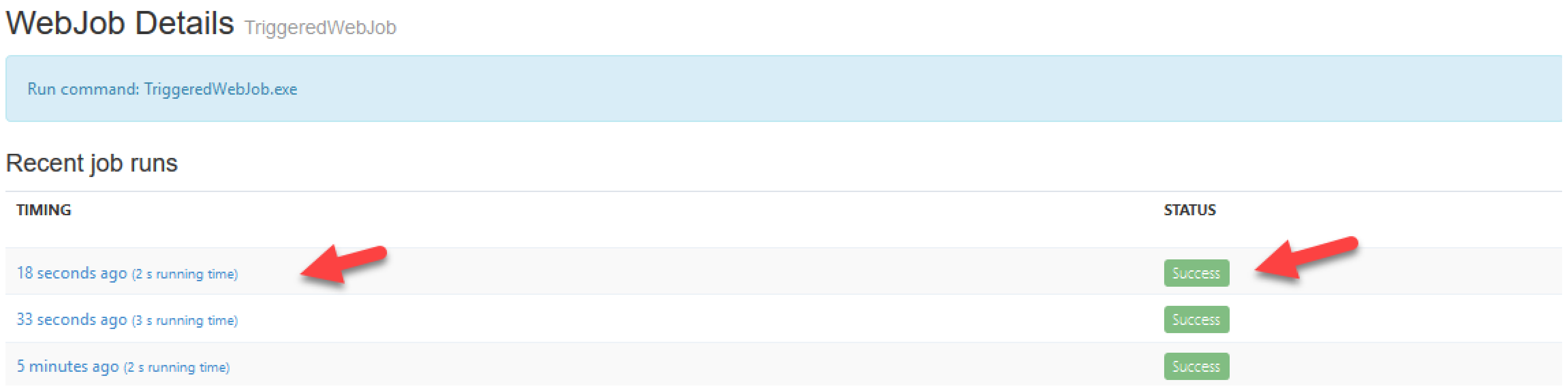


WebJobs provide an easy way to run scripts or programs as background processes in the context of your app.

NAME		STATUS	SCHEDULE
TriggeredWebJob	Triggered	Completed 1 min ago	n/a

Web Jobs
Implementation …

- Logging is available per
  - WebJob
    - Per method within the WebJob
      - Can expand log information per method!



WebJob Details TriggeredWebJob

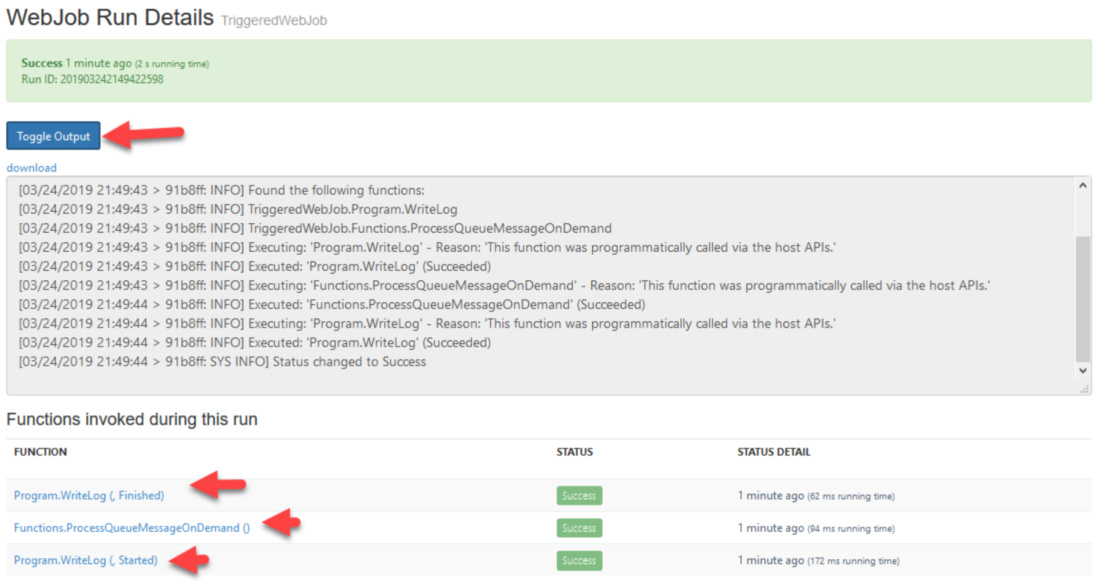
Run command: TriggeredWebJob.exe

Recent job runs

TIMING	STATUS
18 seconds ago (2 s running time)	Success
33 seconds ago (3 s running time)	Success
5 minutes ago (2 s running time)	Success

Web Jobs
Implementation …

- Logging is available per
  - WebJob
    - Per method within the WebJob
      - Can expand log information per method!



WebJob Run Details TriggeredWebJob

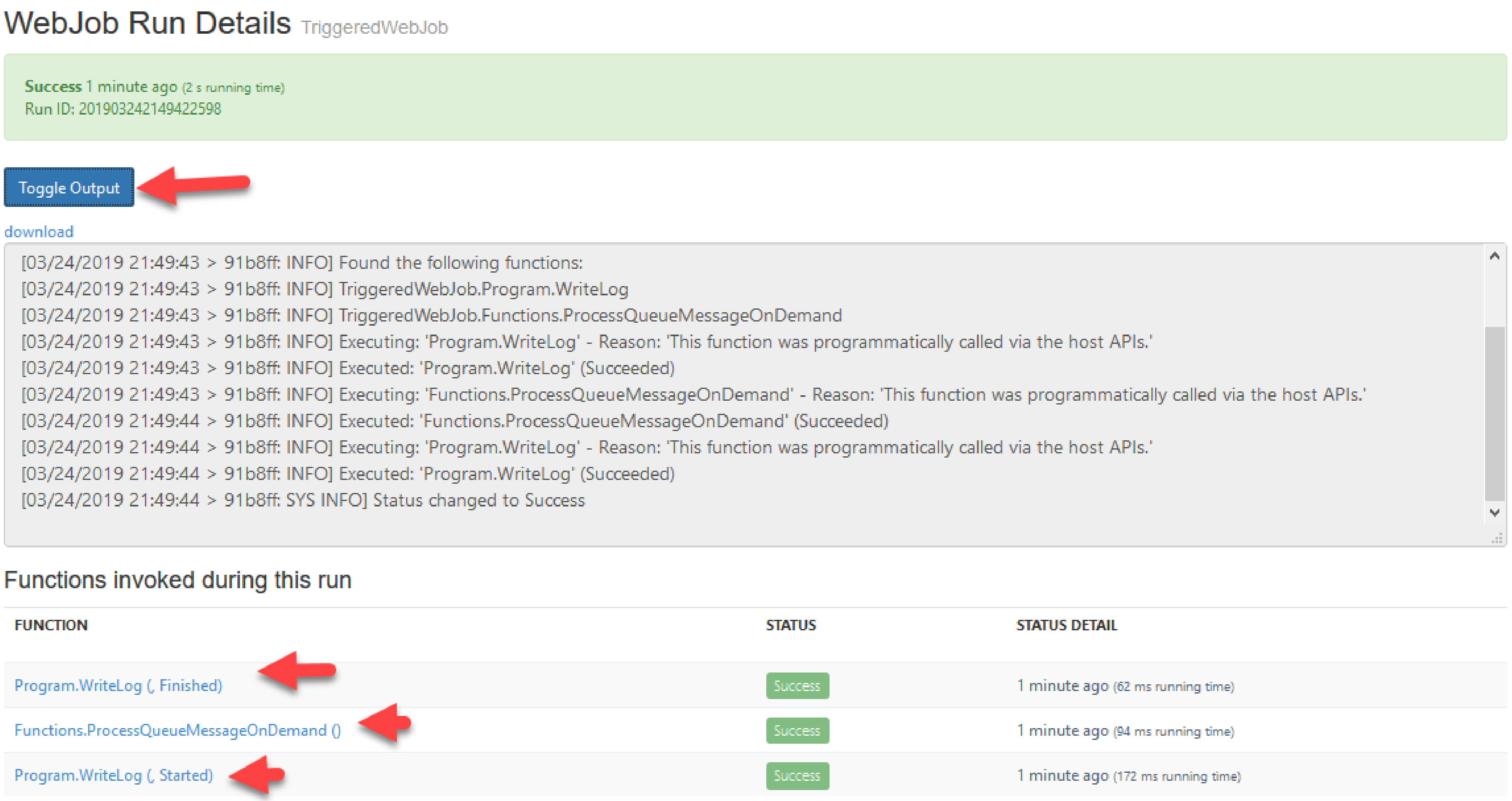
Success 1 minute ago (2 s running time)
Run ID: 201903242149422598

```
[03/24/2019 21:49:43 > 91b8ff: INFO] Found the following functions:
[03/24/2019 21:49:43 > 91b8ff: INFO] TriggeredWebJob.Program.WriteLog
[03/24/2019 21:49:43 > 91b8ff: INFO] TriggeredWebJob.Functions.ProcessQueueMessageOnDemand
[03/24/2019 21:49:43 > 91b8ff: INFO] Executing: 'Program.WriteLog' - Reason: 'This function was programmatically called via the host APIs.'
[03/24/2019 21:49:43 > 91b8ff: INFO] Executed: 'Program.WriteLog' (Succeeded)
[03/24/2019 21:49:43 > 91b8ff: INFO] Executing: 'Functions.ProcessQueueMessageOnDemand' - Reason: 'This function was programmatically called via the host APIs.'
[03/24/2019 21:49:44 > 91b8ff: INFO] Executed: 'Functions.ProcessQueueMessageOnDemand' (Succeeded)
[03/24/2019 21:49:44 > 91b8ff: INFO] Executing: 'Program.WriteLog' - Reason: 'This function was programmatically called via the host APIs.'
[03/24/2019 21:49:44 > 91b8ff: INFO] Executed: 'Program.WriteLog' (Succeeded)
[03/24/2019 21:49:44 > 91b8ff: SYS INFO] Status changed to Success
```

Functions invoked during this run

FUNCTION	STATUS	STATUS DETAIL
Program.WriteLog (, Finished)	Success	1 minute ago (62 ms running time)
Functions.ProcessQueueMessageOnDemand ()	Success	1 minute ago (94 ms running time)
Program.WriteLog (, Started)	Success	1 minute ago (172 ms running time)

Web Jobs
Implementation …



WebJob Run Details TriggeredWebJob

Success 1 minute ago (2 s running time)
Run ID: 201903242149422598

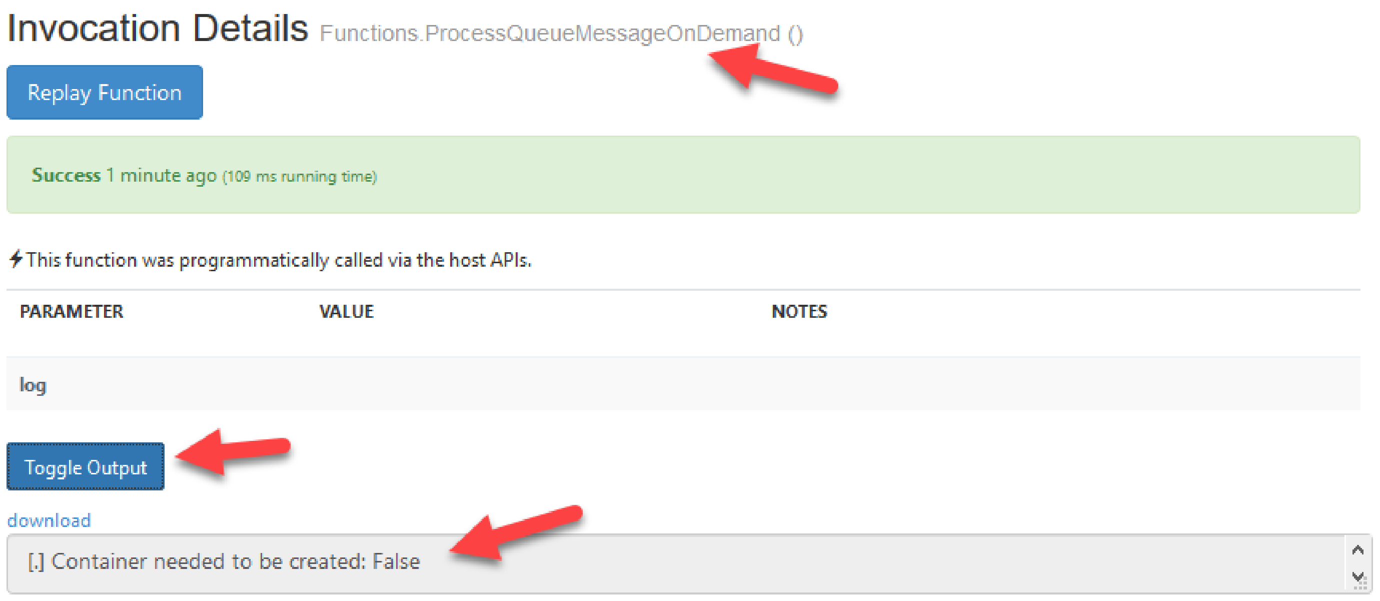
```
[03/24/2019 21:49:43 > 91b8ff: INFO] Found the following functions:
[03/24/2019 21:49:43 > 91b8ff: INFO] TriggeredWebJob.Program.WriteLog
[03/24/2019 21:49:43 > 91b8ff: INFO] TriggeredWebJob.Functions.ProcessQueueMessageOnDemand
[03/24/2019 21:49:43 > 91b8ff: INFO] Executing: 'Program.WriteLog' - Reason: 'This function was programmatically called via the host APIs.'
[03/24/2019 21:49:43 > 91b8ff: INFO] Executed: 'Program.WriteLog' (Succeeded)
[03/24/2019 21:49:43 > 91b8ff: INFO] Executing: 'Functions.ProcessQueueMessageOnDemand' - Reason: 'This function was programmatically called via the host APIs.'
[03/24/2019 21:49:44 > 91b8ff: INFO] Executed: 'Functions.ProcessQueueMessageOnDemand' (Succeeded)
[03/24/2019 21:49:44 > 91b8ff: INFO] Executing: 'Program.WriteLog' - Reason: 'This function was programmatically called via the host APIs.'
[03/24/2019 21:49:44 > 91b8ff: INFO] Executed: 'Program.WriteLog' (Succeeded)
[03/24/2019 21:49:44 > 91b8ff: SYS INFO] Status changed to Success
```

Functions invoked during this run

FUNCTION	STATUS	STATUS DETAIL
Program.WriteLog (, Finished)	Success	1 minute ago (62 ms running time)
Functions.ProcessQueueMessageOnDemand ()	Success	1 minute ago (94 ms running time)
Program.WriteLog (, Started)	Success	1 minute ago (172 ms running time)

Web Jobs
Implementation …

- Logging is available per
  - WebJob
    - Per method within the WebJob
      - Can expand log information per method!



Invocation Details Functions.ProcessQueueMessageOnDemand ()

Replay Function

Success 1 minute ago (109 ms running time)

This function was programmatically called via the host APIs.

PARAMETER	VALUE	NOTES
log		

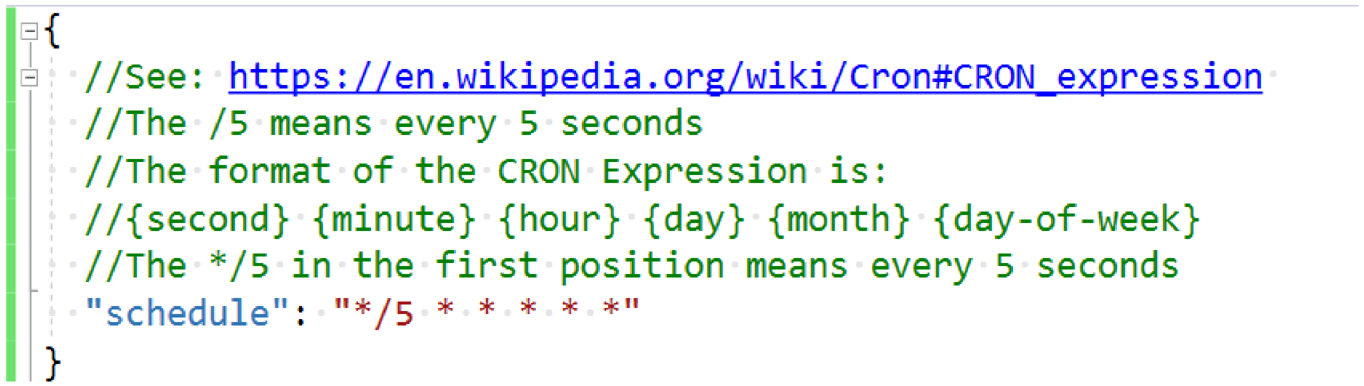
Toggle Output

download

```
[.] Container needed to be created: False
```

Web Jobs
Implementation …

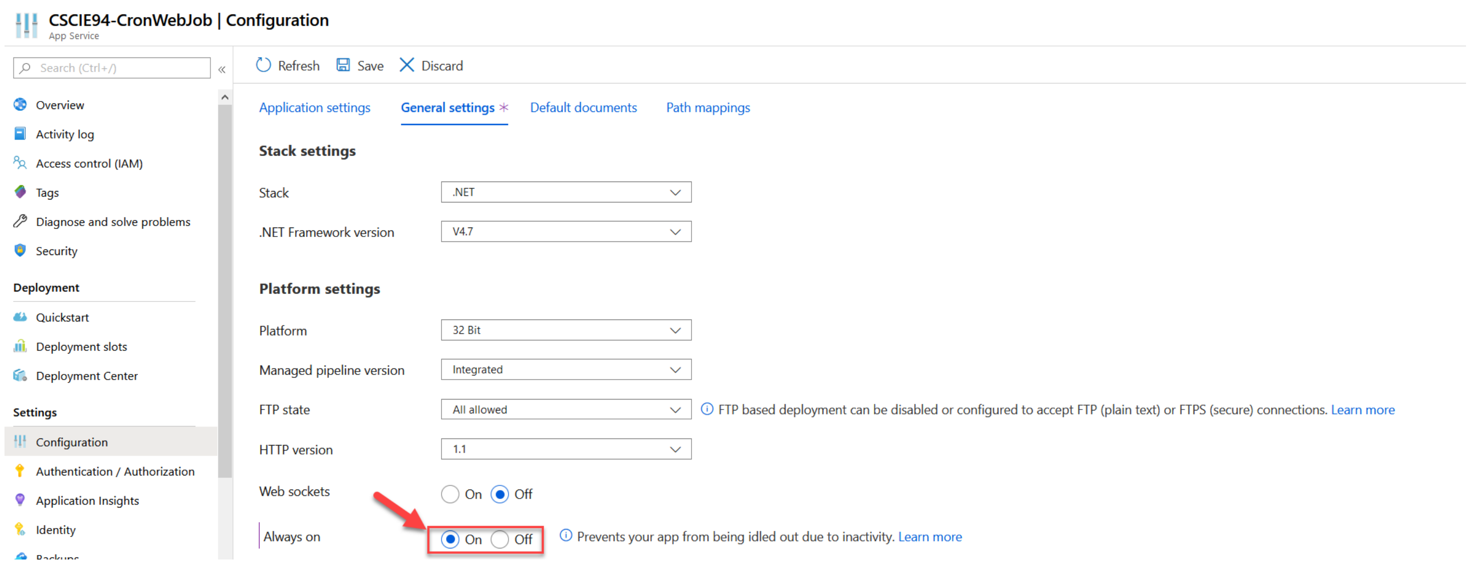
- You can also Trigger based on a timer
  See: Scheduling a triggered WebJob
  - Configured in **settings.job**
    - CRON syntax



```
{
  //See: https://en.wikipedia.org/wiki/Cron#CRON_expression
  //The /5 means every 5 seconds
  //The format of the CRON Expression is:
  //{second} {minute} {hour} {day} {month} {day-of-week}
  //The */5 in the first position means every 5 seconds
  "schedule": "*/5 * * * * *"
}
```

Web Jobs
Implementation …

- Requires **always on** to be set
  See: Configure general settings
  - Webjobs will stop after about 20 minutes
  - Note: **Continuous** also requires **always on**


CSCIE94-CronWebJob | Configuration
App Service

Stack settings

Stack	.NET
.NET Framework version	V4.7

Platform settings

Platform	32 Bit
Managed pipeline version	Integrated
FTP state	All allowed
HTTP version	1.1
Web sockets	On [✓ Off]
Always on	[✓ On] Off

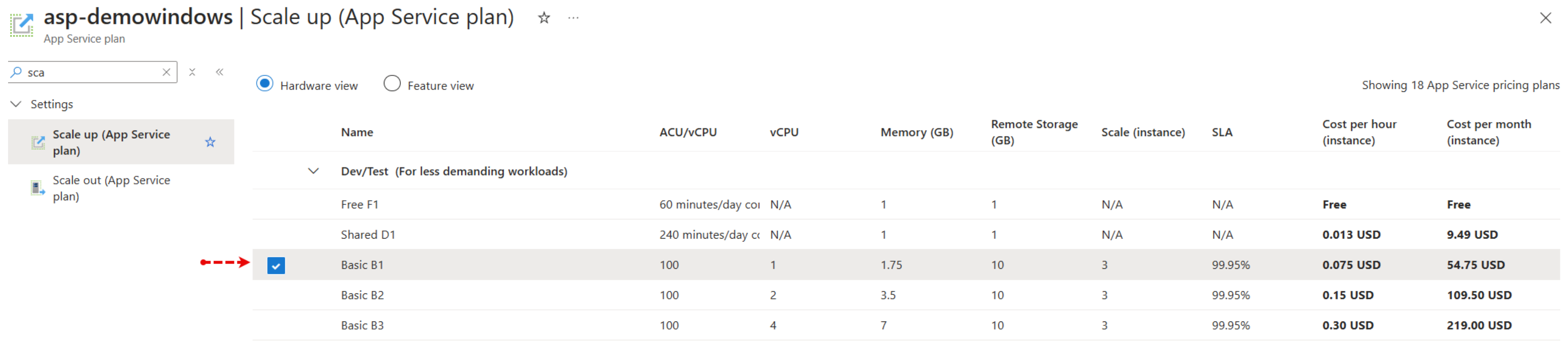
FTP based deployment can be disabled or configured to accept FTP (plain text) or FTPS (secure) connections. Learn more

Prevents your app from being idled out due to inactivity. Learn more

Web Jobs

Implementation …

- Can't use the free plan
  - B1 or above
  - Windows App Service
    - Support for Linux is in preview



asp-demowindows | Scale up (App Service plan)

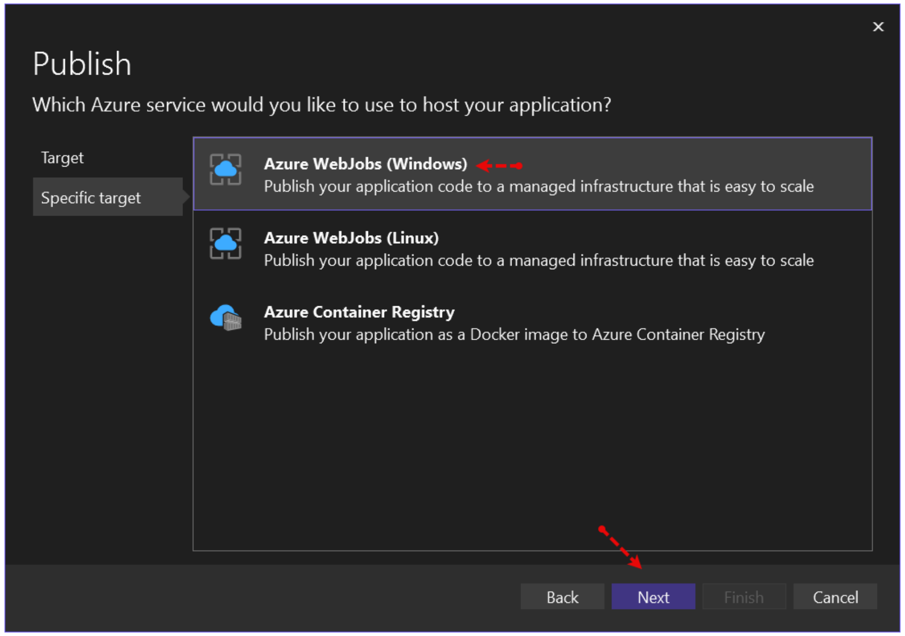
App Service plan

[✓ Hardware view] Feature view

Showing 18 App Service pricing plans

	Name	ACU/vCPU	vCPU	Memory (GB)	Remote Storage (GB)	Scale (instance)	SLA	Cost per hour (instance)	Cost per month (instance)
	Dev/Test (For less demanding workloads)								
	Free F1	60 minutes/day co	N/A	1	1	N/A	N/A	**Free**	**Free**
	Shared D1	240 minutes/day c	N/A	1	1	N/A	N/A	**0.013 USD**	**9.49 USD**
✓	Basic B1	100	1	1.75	10	3	99.95%	**0.075 USD**	**54.75 USD**
	Basic B2	100	2	3.5	10	3	99.95%	**0.15 USD**	**109.50 USD**
	Basic B3	100	4	7	10	3	99.95%	**0.30 USD**	**219.00 USD**

Web Jobs
Implementation …



Publish

Which Azure service would you like to use to host your application?

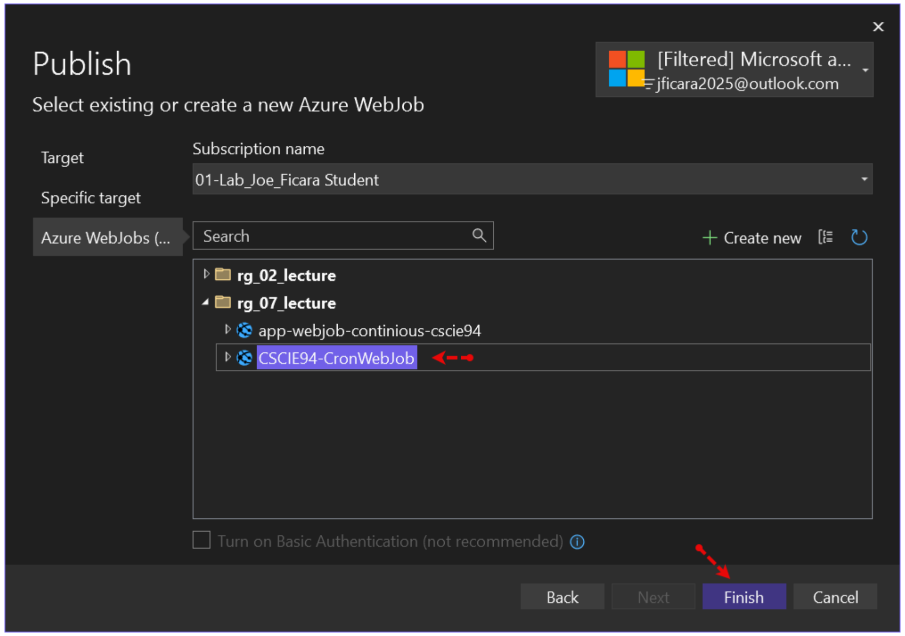
Target

Specific target

Azure WebJobs (Windows)
Publish your application code to a managed infrastructure that is easy to scale

Azure WebJobs (Linux)
Publish your application code to a managed infrastructure that is easy to scale

Azure Container Registry
Publish your application as a Docker image to Azure Container Registry



Publish

Select existing or create a new Azure WebJob

[Filtered] Microsoft a...
jficara2025@outlook.com

Target

Specific target

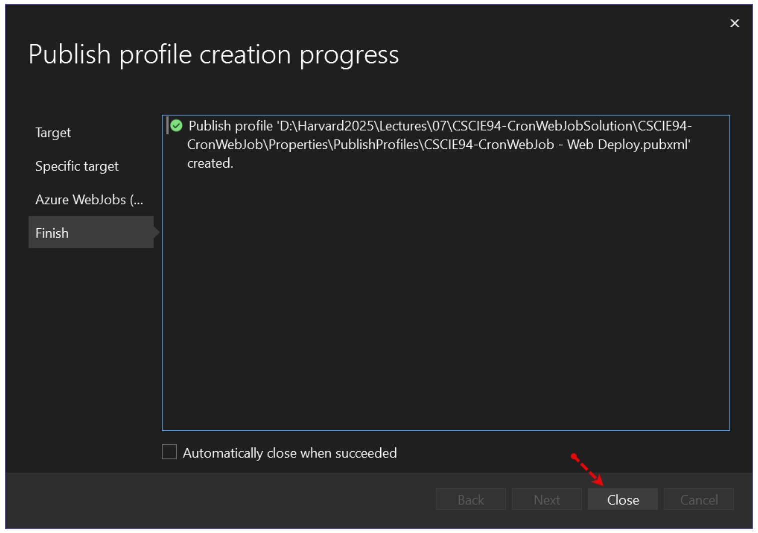
Azure WebJobs (...

Subscription name: 01-Lab_Joe_Ficara Student

- rg_02_lecture
- rg_07_lecture
  - app-webjob-continious-cscie94
  - CSCIE94-CronWebJob

Turn on Basic Authentication (not recommended)

Web Jobs Implementation …

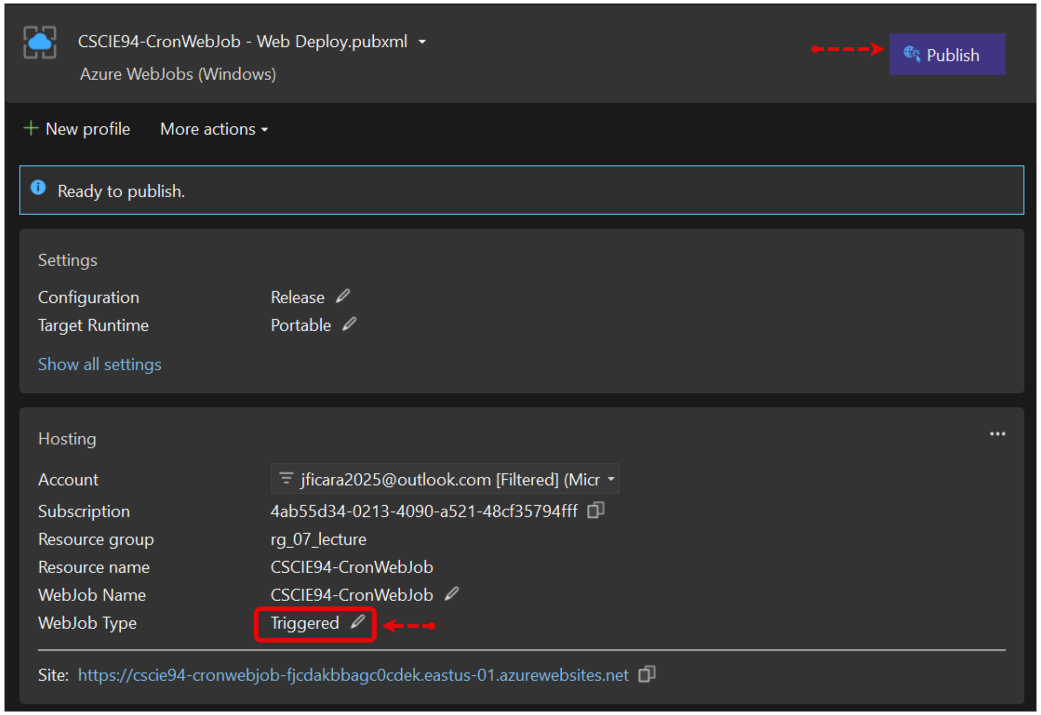


Publish profile creation progress

- Target
- Specific target
- Azure WebJobs (...
- Finish

Publish profile 'D:\Harvard2025\Lectures\07\CSCIE94-CronWebJobSolution\CSCIE94-CronWebJob\Properties\PublishProfiles\CSCIE94-CronWebJob - Web Deploy.pubxml' created.

Automatically close when succeeded



CSCIE94-CronWebJob - Web Deploy.pubxml
Azure WebJobs (Windows)

Ready to publish.

Settings

Configuration	Release
Target Runtime	Portable

Show all settings

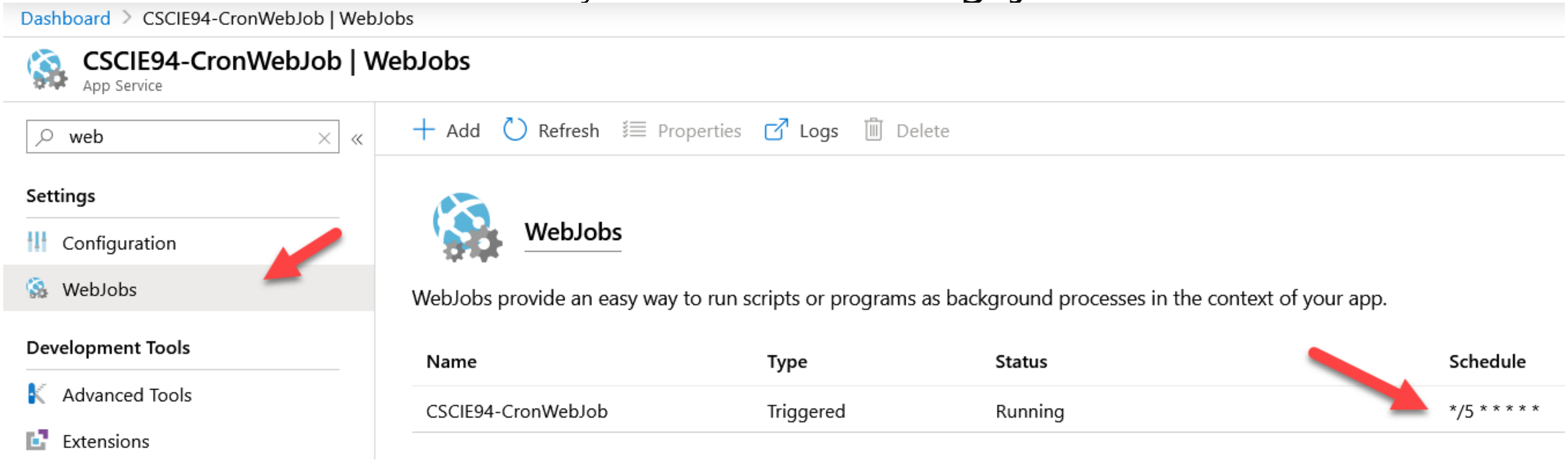
Hosting

Account	jficara2025@outlook.com [Filtered] (Micr
Subscription	4ab55d34-0213-4090-a521-48cf35794fff
Resource group	rg_07_lecture
Resource name	CSCIE94-CronWebJob
WebJob Name	CSCIE94-CronWebJob
WebJob Type	Triggered

Site: https://cscie94-cronwebjob-fjcdakbbagc0cdek.eastus-01.azurewebsites.net

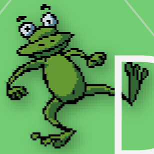
Web Jobs
Implementation …

- Portal indicates schedule
  - Showing the CRON setting
    - From **setting.job** file
      - Modify schedule in **settings.job file**



WebJobs provide an easy way to run scripts or programs as background processes in the context of your app.

Name	Type	Status	Schedule
CSCIE94-CronWebJob	Triggered	Running	*/5 * * * * *



Demo

Web Jobs

ContinuousWebJobSolution

ManualOnDemandWebJobSolution

RunWebJobSolution

CSCIE94-CronWebJobSolution

Web Jobs

.NET Core Web Job …

- There is no built-in template
- We need to
  - Build a .NET 9 Console Application
  - Add some nuget packages
    - `Microsoft.Azure.WebJobs.Extensions`
    - `Microsoft.Azure.WebJobs.Extensions.Storage`
    - `Microsoft.Extensions.Configuration.UserSecrets`
    - `Microsoft.Extensions.Logging.Console`

Web Jobs

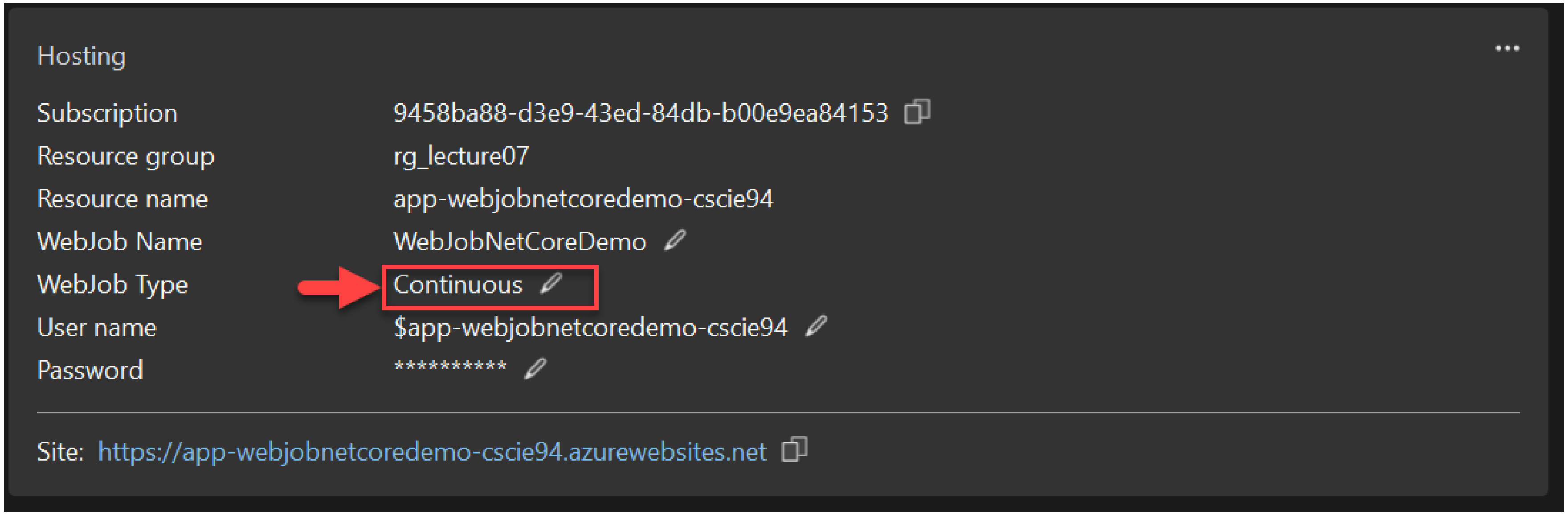
.NET Core Web Job …

- Configure the code in **program.cs**
  - Create a HostBuilder()
  - Register services
  - Configure
    - Logging, Storage SDK
    - Settings location & Dependency Injection
- Add Function classes
- Let's walk through the code

Web Jobs

.NET Core Web Job ...

- Note:
  - Deploy as "continuous"



Hosting

Subscription	9458ba88-d3e9-43ed-84db-b00e9ea84153
Resource group	rg_lecture07
Resource name	app-webjobnetcoredemo-cscie94
WebJob Name	WebJobNetCoreDemo
WebJob Type	Continuous
User name	$app-webjobnetcoredemo-cscie94
Password	**********

Site: https://app-webjobnetcoredemo-cscie94.azurewebsites.net

Demo

.NET Core Web Jobs

`WebJobNetCoreDemoSolution`

Web Jobs

- **Remember**
  - **Base 64 Encode** your messages
    - Before sending them from your App Service
  - **Enable SCM Basic Auth** Publishing Credentials
    - To obtain Credentials from the portal
  - **Add environment variable app settings**
    - With the storage connection string
      - **AzureWebJobsDashboard**
      - **AzureWebJobsStorage**

Azure Webjobs

Links & Resources

- Web Jobs
  - .NET Core Web Jobs
    - Tutorial: Get started with the Azure WebJobs SDK for event-driven background processing
  - Create a .NET WebJob in Azure App Service
    - Run background tasks with WebJobs in Azure App Service
  - Deploying webjobs
    - Develop and deploy WebJobs using Visual Studio
  - Using Webjobs with Kudu API
    - WebJobs API · projectkudu/kudu Wiki (github.com)
  - Trouble shooting
    - No functions found in trigger-less job
      - Azure WebJobs - No functions found - How do I make a trigger-less job?