AZURE DEBUGGING APP SERVICES

MODULE: APPLICATION DEVELOPMENT 3A

CONTENT DEVELOPED BY: CASSIM VANKER

LEARNING OBJECTIVES

- Azure Debugging App Services
  - Debugging on azure
  - Types of logs for app services
  - How to grab Diagnostic Logs?
  - Application Insights
  - How to add application insights in vs?
  - Log Stream
  - Kudu debug console
  - Kudu debug console functionality
  - Remote debugging

DEBUGGING ON AZURE

- Azure has a number of tool to assist with debugging.
- Azure resource-level diagnostic logs are logs emitted by a resource that provide rich, frequent data about the operation of that resource.
- The content of these logs varies by resource type.
- Application logs can be saved to the file system or to blob storage.
- Application logs to the file system are automatically turned off after 12 hours. However, logging to Blog storage continues indefinitely.
- Logs stored on the file system, can be accessed by FTP, or downloaded as a Zip archive by using the Azure PowerShell or Azure Command-Line Interface (Azure CLI).
- By default, logs are not automatically deleted. To automatically delete logs, set the **Retention Period (Days)** field.

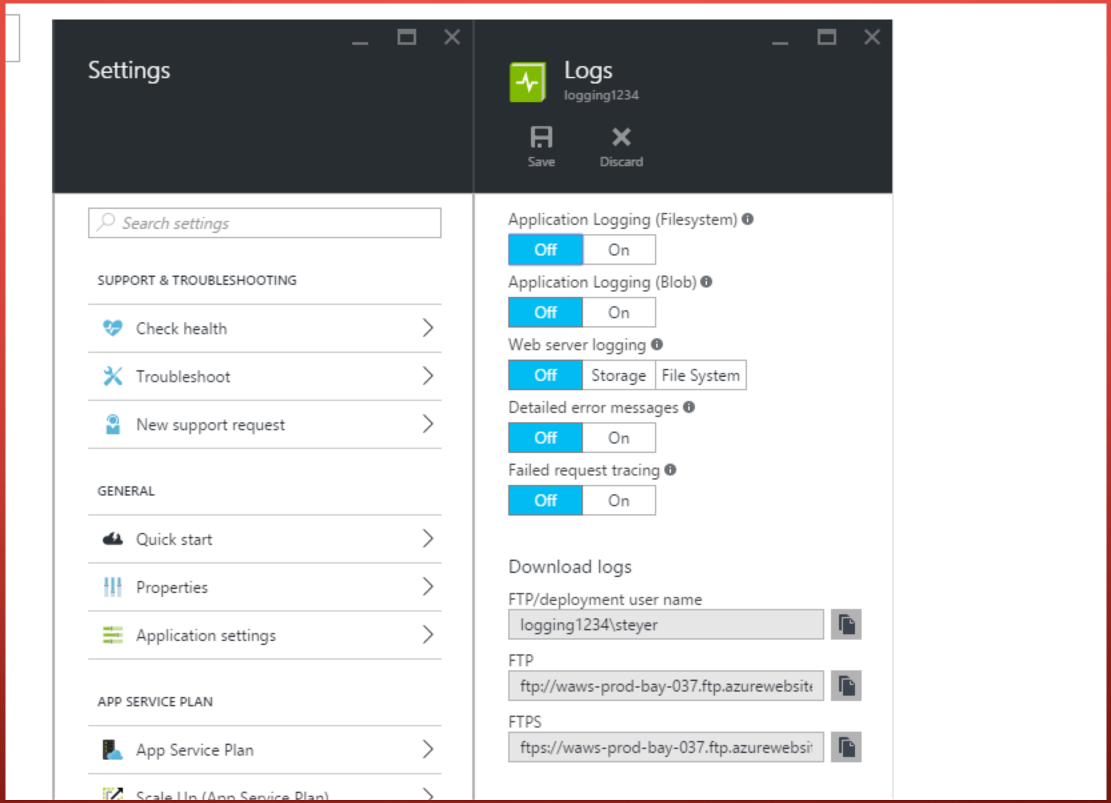
TYPES OF LOGS FOR APP SERVICES

Type's of Logs:

- **Detailed Error Logging** - Detailed error information for HTTP status codes that indicate a failure (status code 400 or greater). It may contain information that can help determine why the server returned the error code. (/LogFiles/Application/)
- **Failed Request Tracing** - Detailed information on failed requests, including a trace of the IIS components used to process the request and the time taken in each component. It is useful if you are attempting to increase site performance or isolate what is causing a specific HTTP error to be returned. (/LogFiles/W3SVC#########/)
- **Web Server Logging** - Information about HTTP transactions using the W3C extended log file format. It is useful when determining overall site metrics such as the number of requests handled or how many requests are from a specific IP address. (/LogFiles/http/RawLogs)

HOW TO GRAB DIAGNOSTIC LOGS?

- To enable diagnostics in the Azure portal, go to the page for your web app and click Settings > Diagnostics logs.



Application Logging (Filesystem): Off | On

Application Logging (Blob): Off | On

Web server logging: Off | Storage | File System

Detailed error messages: Off | On

Failed request tracing: Off | On

Download logs

FTP/deployment user name: logging1234\steyer

FTP: ftp://waws-prod-bay-037.ftp.azurewebsite

FTPS: ftps://waws-prod-bay-037.ftp.azurewebsi

APPLICATION INSIGHTS

- Application Insights is an extensible Application Performance Management (APM) service for web developers on multiple platforms.
- It is used to monitor your live web application.
- It will automatically detect performance anomalies.
- It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app.
- It's designed to help you continuously improve performance and usability.
- It works for apps on a wide variety of platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud.

HOW TO ADD APPLICATION INSIGHTS IN VS?

- In the Azure Control Panel in the blade for the App Service. Choose Application Insights under monitoring and give the resource a name and location.
- In Visual Studio, right click on you web application and choose "Add Application Insights Telemetry".
- You can enable the free version. This give you 1 gig of data.
- Choose your subscription and the resource you created in the first step.
- Finally, select register.
- The installation will install the necessary nu-get packages.

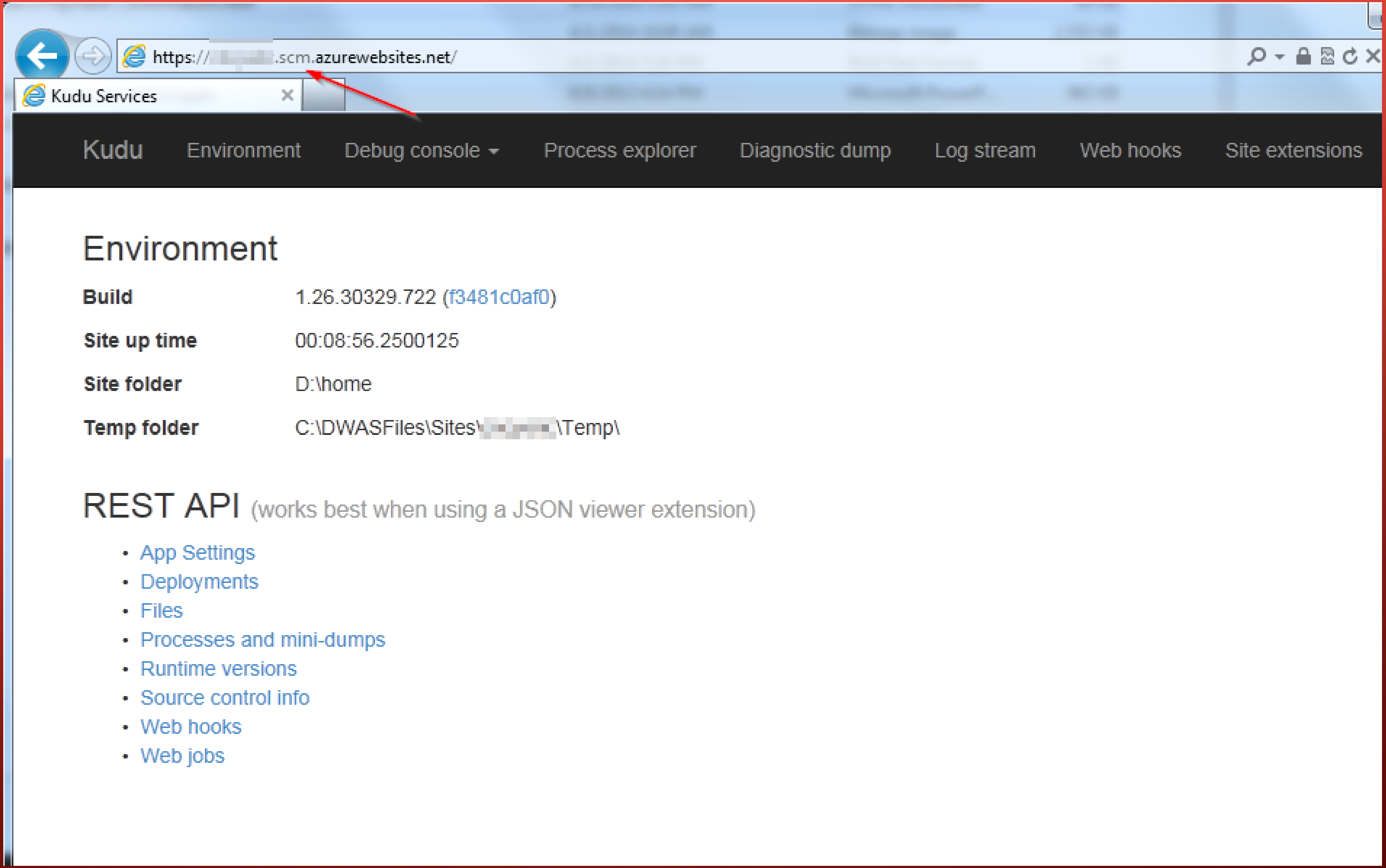
LOG STREAM

- Log stream allows you to view live logs
- In the blade for the App Service. Choose log stream (under monitoring).
- You will be able to view the live request coming into the web server as well as the application logs.
- The logs are slightly delayed but show updates in Realtime.
- Cloud Explorer in visual studio allows you to download the log.

KUDU DEBUG CONSOLE

- One of the advantages of using Azure App Service is that it's a Platform-as-a-Service, meaning that you don't have to maintain and patch the operating system.
- A side effect you can't use remote desktop to connect directly to the VM.
- To assist with debugging Azure has provided an amazing tool called Kudu.
- Kudu is the engine behind git deployments, WebJobs, and various other features in Azure Web Sites.
- Kudu is open source and you can find the source code at GitHub.
- In order to access KUDU interface simply navigate to your Azure Web Site and specify "scm" as part of the address. Such as https://mysite.**scm**.azurewebsites.net.

KUDU DEBUG CONSOLE



Environment

Build	1.26.30329.722 (f3481c0af0)
Site up time	00:08:56.2500125
Site folder	D:\home
Temp folder	C:\DWASFiles\Sites\[redacted]\Temp\

REST API (works best when using a JSON viewer extension)

- App Settings
- Deployments
- Files
- Processes and mini-dumps
- Runtime versions
- Source control info
- Web hooks
- Web jobs

KUDU DEBUG CONSOLE FUNCTIONALITY

- Diagnostic Dump (Tools Menu): download all the log files as a single zip file.
- Processor Explorer: See what is running on the web server.
- Debug Console: Open a command shell prompt on the server. In this section there are two views. The first view shows you the file system. The second view is the command shell. Files can also be edited in their folders.

REMOTE DEBUGGING

- If you can't simulate an error locally but only on the server. You can use remote debugging to diagnose the error.
- In visual studio open up you application and publish you application as a debug release.
- In cloud explorer, right click on the App Service and choose Attach Debugger.
- Under application setting in the Azure App Service Blade.
- Turn remote debugging on