CST-323 Activity Guide

Contents

© 2023. Grand Canyon University. All Rights Reserved.

Activity 1: Design Cloud Test Application & Cloud Research

Overview

In this activity, students will:

1. Get an Azure Student Account.
2. Design and build a cloud test application to be used in future activities.
3. Complete cloud computing research on the basic concepts of cloud computing.

IMPORTANT:

The cloud test application (referred to as the test application throughout this guide) mentioned above will be used in future topics and activities to test and validate several cloud platforms. The following are the approved technologies, depending on the program you are in, that can be used to design and build the test application:

- BSCP Program:
  - Leverage the Legacy How to Guide, found in your course materials, for build and deployment instructions.
  - Use the following technologies: PHP Laravel or Spring Framework with MySQL.
  - Leverage a logging framework, such as SLF4J/Log4J for Java or Monolog for PHP.
- BSSD Program:
  - Leverage the How to Guide, found in your course materials, for build and deployment instructions.
  - Use the following technologies: Spring Boot with MySQL.
  - Leverage a logging framework, such as SLF4J/Log4J for Java.

Execution

Execute this activity according to the following guidelines:

1. Get a Microsoft Azure account (this will be used in the Topic 3 and 4 activities):
   a. Sign up for a Microsoft Azure for Student Starter subscription by searching in the Student Success Center for GCU Cloud Hosting Solutions. Select the GCU Cloud Hosting Solutions resource from the search results.
   b. Select the Microsoft Azure link. Follow the instructions in the sign-up guide.
   c. Sign into the Azure Portal.
   d. Validate that you have Microsoft Azure for Students subscription by selecting your Home menu (from the hamburger menu), then select the Subscriptions icon, and validate that you have an Microsoft Azure for Students listed under Offer. If you do not have a Microsoft Azure for Students subscription, see your instructor. DO NOT proceed forward with any Activities in this course unless you have a Microsoft Azure

© 2023. Grand Canyon University. All Rights Reserved.

for Students subscription. DO NOT sign up for the standard free Microsoft Azure account, which requires a credit card.

d. Take a screenshot of the Azure Portal with the screen demonstrating that you have the Microsoft Azure for Student Starter subscription.

2. Start cloud test application.
    a. You will design and build a test application that will be used in future topics to test and validate a number of cloud platforms.
    b. The goal of the test application is NOT to build a complicated feature rich application but to build an application that can be easily deployed onto and validated on a number of cloud platforms.
    c. The test application must meet the following requirements:
        i. Designed and built per the technical requirements outlined in Activity 1.
        ii. Designed and built using the MySQL database.
        iii. Designed and built to include three to four pages that includes a combination of pages to enter user data into forms and display user data.
        iv. Designed to support all the database CRUD methods.
        v. Designed using Bootstrap.
        vi. Developed using a GIT repository.
        vii. Demonstrate functionality in a screencast.
3. Cloud computing research:
    a. Read the assigned textbook required readings for this topic. In 75-100 words, describe the evolution of cloud computing.
    b. Pick one of the case studies from Chapter 1 of the textbook. What are the advantages and disadvantages the company encountered when moving their applications and data to the cloud? Provide three advantages and three disadvantages along with details regarding any challenges the company faced. In addition, provide a list of three features the company was able to take advantage of in the cloud.
    c. Analyze the Cloud vs. On Premise Server Prototyping Example in Chapter 1 of the textbook. Select either the Cloud or On-Premise solution that could be used to deploy your company's business applications. Provide a detailed (100 words minimum) rationale on your recommendation and justification for your solution choice.

Documentation

Submit the following to the learning management system:

An activity report documenting the progress of the completion for the following application components:

1. Cover sheet
2. A screenshot of being logged into the Azure Portal

© 2023. Grand Canyon University. All Rights Reserved.

3. The framework and technology chosen for the test application
4. The progress and status on the database designed for the test application, using an ER diagram, as well as outlining what tables have been built and what tables are remaining to be built
5. The progress and status on the test application development, outlining what pages and services have been built, as well as what pages and services are remaining to be built
6. A list of issues that are currently ongoing that are hampering the completion of the test application
7. A screencast URL demonstration of the test application functionality
8. The cloud computing research questions

Activity 2: Build Cloud Test Application and Cloud Research

Overview

In this activity, students will:

1. Design and build a cloud test application to be used in future activities.
2. Complete cloud computing research on the cloud deployment models and cloud service models.

Execution

Execute this activity according to the following guidelines:

1. Finish Cloud Test Application.
    a. You will design and build a test application that will be used in future topics to test and validate a number of cloud platforms.
    b. The goal of the test application is NOT to build a complicated feature rich application but to build an application that can be easily deployed onto and validated on a number of cloud platforms.
    c. The test application must meet the following requirements:
        i. Designed and built per the technical requirements outlined in Activity 1.
        ii. Designed and built using the MySQL database.
        iii. Designed and built to include three to four pages that includes a combination of pages to enter user data into forms and display user data.
        iv. Designed to support all the database CRUD methods.
        v. Designed using Bootstrap.
        vi. Developed using a GIT repository.
        vii. Demonstrate functionality in a screencast.
2. Cloud Computing Research:

© 2023. Grand Canyon University. All Rights Reserved.

a. Analyze each of the cloud deployment models (public cloud model, private cloud model, and the hybrid cloud model). Provide three advantages and three disadvantages for using each cloud deployment model. Explain your rationale.

b. Identify two SaaS applications. Provide three advantages and three disadvantages to the identified application when compared to building and hosting those same applications yourself. Explain your rationale.

c. Read the assigned textbook required readings for this topic. From an application developer's perspective, what are three primary differences between using a PaaS Cloud Server versus an IaaS Cloud Server for your environment? Explain your rationale.

Documentation

Submit the following to the learning management system:

An Activity Report documenting the progress of the completion for the following application components:

1. Cover sheet
2. Updated test application database and application design
3. A URL of the screencast demonstration of the running test application
4. Cloud computing research questions
5. A zip file of the code for the cloud test application

Activity 3: Deploy Application to Cloud and Cloud Research Part 1

Overview

In this activity, students will:

1. Deploy the cloud test application onto the Microsoft Azure and Heroku.
2. Complete cloud computing research on the cloud deployment best practices and cloud platform feature comparison.

Execution

Execute this activity according to the following guidelines:

1. Cloud platform deployment. Complete the following for the Microsoft Azure and Heroku cloud platforms. Refer to the "How to Guide," located within Class Resources, for Cloud Platform deployment instructions. For each cloud platform:

    a. Create an account on the cloud platform.

    b. Research the container(s) and service(s) available on the cloud platform that will be required to support your test application. Provision the necessary container(s) and service(s) required to deploy your application and database.

    c. Configure the database using your DDL script.

    d. Configure and deploy your application code.

© 2023. Grand Canyon University. All Rights Reserved.

e. Test your application and database.

f. Prepare a screencast demonstrating all the functionality of your test application running on the cloud platform.

2. Cloud Computing Research:

   a. Read the assigned textbook required readings for this topic. Identify eight "worst practices" when migrating your application to the cloud. Identify the worst practice in two to three sentences and then provide three to five sentences on how to prevent the worst practice from becoming a risk or issue during an application cloud migration.

   b. Compare the cloud features of Microsoft Azure and Heroku. Present at least 10 features, explaining how they are similar and/or different. Explain your rationale.

Documentation

Submit the following to the learning management system:

An Activity Report that contains the following on Cloud Application Deployment and Cloud Computing:

1. Cover sheet
2. Final database and application design
3. Cloud deployment report for each cloud platform containing the following information:
   a. Step-by-step instructions used to configure and deploy the test application. The instructions should be detailed and comprehensive that includes all appropriate screenshots from the Cloud Platform that fully document the PaaS container procurement, database configuration, test application configuration, and test application deployment. NOTE: do not copy the instructions from the How To Guide as this will not be accepted.
   b. Challenges encountered during the database and application deployment
   c. A screencast of the test application being run on the cloud platform
4. Cloud Computing Research questions.

Activity 4: Deploy Application to Cloud and Cloud Research Part 2

Overview

In this activity, students will:

1. Deploy the cloud test application onto the Amazon AWS and the Google Cloud platforms.
2. Complete cloud computing research on REST API solution design and cloud platform feature comparison.

Execution

Execute this activity according to the following guidelines:

© 2023. Grand Canyon University. All Rights Reserved.

1. Cloud platform deployment. Complete the following for the AWS and Google Cloud platforms. Refer to Parts 4 and 5 within the "How to Guide," located within Class Resources, for cloud platform deployment instructions. For each cloud platform:
    a. Create an account on the cloud platform.
    b. Research the container(s) and service(s) available on the cloud platform that will be required to support your test application. Provision the necessary container(s) and service(s) required to deploy your application and database.
    c. Configure the database using your DDL script.
    d. Deploy your application code.
    e. Test your application and database.
    f. Prepare a screencast demonstrating all the functionality in your test application running on the cloud platform.
2. Cloud Computing Research:
    a. Read the assigned textbook required readings for this topic. Let's say you decide to leverage (i.e., consume) a REST API that is published from a social media website, such as Facebook or Twitter. What are five relevant technical questions related to non-functional requirements and security that would you need to ask the vendor?
    b. Compare the cloud features of Amazon AWS and Google Cloud. Present at least 10 features, explaining how they are similar and/or different. Explain your rationale.
    c. Identify five technical or business limitations that could restrict an existing application from being deployed onto any of the Cloud Platforms that were utilized in Topic 3 or Topic 4. Explain your rationale.

Documentation

Submit the following to the learning management system:

An activity report that contains the following on cloud application Deployment and cloud computing:

1. Cover sheet
2. Cloud deployment report for each cloud platform containing the following information:
    a. Step-by-step instructions used to configure and deploy the test application. The instructions should be detailed and comprehensive that includes all appropriate screenshots from the Cloud Platform that fully document the PaaS container procurement, database configuration, test application configuration, and test application deployment. NOTE: do not copy the instructions from the How To Guide as this will not be accepted.
    b. Challenges encountered during the database and application deployment.
    c. A screenshot of the current billing report for the cloud platform.
    d. A screencast of the test application being run on the cloud platform.

© 2023. Grand Canyon University. All Rights Reserved.

3. Cloud computing research questions

Activity 5: Cloud Containers and Cloud Research

Overview

In this activity, students will:

1. Complete Docker container and Kubernetes tutorials.
2. Complete cloud computing research on cloud architecture, Docker containers, and cloud availability.

Execution

Execute this activity according to the following guidelines:

1. Create an account at Docker Hub., the link for which is located in the Topic 6 materials. This will be required to complete the following tutorials.
2. Complete the following tutorial on Docker:
    a. Clear your browser data and cache.
    b. Go to "Play with Docker Classroom," located in the Topic 6 materials, and log in to the training site using your Docker Hub account.
    c. Complete the following tutorials by clicking on the "Full list of individual labs" and then under the Beginner section access the following tutorials:
        - Docker for Beginners – Linux
        - Doing More With Docker Images
        - First Alpine Linux Containers
        - Docker Images Deeper Dive
    d. Take a screenshot for each of the completed tutorials.
    e. Answer the following questions on Docker:
        - Include a screenshot of completed Docker tutorials.
        - What is Docker?
        - What is a Docker File?
        - What is a Docker Image?
        - What is a Docker Container?
        - What is Docker Hub?
        - What are five advantages to using Docker Containers?
        - What are five Docker commands you used in the tutorial and what was the purpose for using each of the commands?
3. Complete the following tutorial on Kubernetes:

© 2023. Grand Canyon University. All Rights Reserved.

a. Go to https://kubebyexample.com/learning-paths/kubernetes-fundamentals/what-kubernetes-3-minutes tutorial.

b. Complete the following tutorials:

- What is Kubernetes in 3 minutes?
- Kubernetes CLI with kubectl
- What is a Pod?
- What is a ReplicaSet?
- What is a Deployment?
- What is a Label?
- What is a Service?

c. For each completed tutorial write up 10 technical things you learned about Kubernetes.

d. Answer the following questions on Kubernetes:

- Include a screenshot of the completed Kubernetes tutorial.
- What is Kubernetes?
- Why would you have a need to use Kubernetes?
- What are five features that you could leverage from Kubernetes?

3. Cloud Computing Research:

a. Read the assigned textbook required readings for this topic. Provide the following research:

- Define the business problem statement for your milestone project.
- Draw a business architecture diagram for your milestone project.
- Identify the business and technical requirements for your milestone project.

b. Describe three elements that can be defined in a DockerFile. Provide a brief description for each and discuss what the elements are used for.

c. Research the concepts of high availability (HA), failover, and the number of nines. What is HA, failover, and the number of nines? How does the number of nines help solve HA and failover?

Documentation

Submit the following to the learning management system:

An activity report that contains the following related to Docker, Kubernetes, and cloud computing:

1. Cover sheet
2. Tutorial screenshots and questions on Docker.
3. Tutorial screenshots and questions on Kubernetes.

© 2023. Grand Canyon University. All Rights Reserved.

4. Cloud computing research questions.

Activity 6: DevOps and Cloud Research

Overview

In this activity, students will:

1. Integrate DevOps concepts into the cloud test application.
2. Complete cloud computing research on DevOps capabilities and tools.

Execution

Execute this activity according to the following guidelines:

1. DevOps logging integration with the test application.
    a. Leverage a logging framework, as outlined in the technical requirements in Activity, into your test application and then add logging statements for all method entry and exit paths, as well as for all exceptions. The logging statements should leverage the proper log levels and be formatted using consistent naming conventions, consistent error messages, contain a date/time stamp, and contain the class name, as well as the method name. To save coding effort it is strongly recommended that you use the interceptor design pattern in your design and implementation.
    b. Redeploy your test application to the AWS and Heroku cloud platforms.
    c. View the log files for your deployed test application on the following cloud platforms. Take a screenshot of the log trace for each cloud platform.
        i. AWS: To view the Application Server log, Application log, or stdout from the Portal, select the Elastic Beanstalk service, click on the Application link from the left pane, click on the environment link under the Environments column for your application, click on the Logs link from the left pane, and then from the dropdown select either to view the last 100 lines or all lines from the logs, and then click on the Download link to view the log file. You must set the logging configuration to log to stdout (system console). The logs will be found under the /var/log/web.stdout.log section.
        ii. Heroku: To view the Application Server log, Application log, or stdout from the Portal, select your application, select the View Logs menu. You must set the logging configuration to log to stdout (system console).
    d. Answer the following questions on DevOps logging:
        i. Why is adding robust logging important for an application deployed to the cloud?
        ii. What are three features of the logging framework that you did not implement but would be important to implement for a production-level application?
        iii. What are two enterprise class logging products besides Loggly that could be used to search and archive application or system log files?

© 2023. Grand Canyon University. All Rights Reserved.

2. DevOps monitoring integration with the test application.
    a. Create a free trial account on Loggly. It should be noted that the free trial for Loggly only lasts for 14 days. If for some reason your trial expires simply create another account to finish this activity.
    b. Heroku has a Drain that will send logs in "real time" to Loggly. Also refer to the Log Drains article. NOTE: for instructions on how to integrate Loggly with Express, Angular, and React see Appendix A. Since these are client-side frameworks integration with Heroku will not be required.
        1) Log into Heroku CLI.
        2) You will need to get your Loggly Customer Token for running the command below. You can get your Loggly Customer Token by accessing your Settings from the Loggly main menu (icon on the left side of the page) then clicking the API Tokens menu option. The Customer Token is retrieved by clicking on the hyperlink at the top of this page.
        3) Setup a Heroku Log Drain by running the following command:

```
heroku drains:add http://logs-01.loggly.com/bulk/[LOGGLY_CUSTOMER_TOKEN]/tag/cst323_logfile_heroku_upload --app [APP_NAME]
```

        4) To demonstrate a Loggy Tag Query search in Loggly using tag:[TAG NAME] query (i.e. cst323_logfile_heroku_upload in this example and note that searching by a desired tag is done using the tag: search command). Take a screenshot of the search results.
        5) To demonstrate a Loggy Text Query search in Loggly using a [DESIRED TEXT] query. Refer to the online Loggly Help documentation for additional information on search queries. Take a screenshot of the search results.

        NOTE: Refer to Appendix A for implementation notes.
    c. Create a free account on Uptime Robot.
        1) Log in to Uptime Robot.
        2) Set up a New Monitor using the URL of your test application's main index page on one of the cloud platforms. The Monitor Type should be HTTP(s) using the URL of test application's main index page with a monitor interval of five minutes using an Alert Contact for your GCU email address.
        3) Test your monitor by stopping your test application on the cloud platform. Take a screenshot of your Monitor Alert email.

        NOTE: Refer to Appendix B for implementation notes.
    d. Answer the following questions on DevOps Monitoring:
        a. What is the purpose for setting up a log file alert?
        b. What is the purpose for setting up an application availability alert?
3. DevOps CI/CD integration with the test application.

© 2023. Grand Canyon University. All Rights Reserved.

a. Explore the build pipeline tools from one the Cloud Platforms where the Test Application was deployed onto. Students can optionally explore other build tools, such as GitLab.

b. Setup a build pipeline using the selected tool from the above list and deploy your test application to one of desired cloud platforms.

c. Complete a screencast of an automated build and deployment.

d. Answer the following questions on DevOps CI/CD:

    1) Screencast of the automated build and deployment.

    2) What roles does Maven play when supporting CI/CD?

    3) What role does a Source Control System play when supporting CI/CD?

    4) How did your chosen build pipeline tool support CI/CD?

    5) Besides build and deployment, what are three other features that could be integrated into a build pipeline to support a robust CI/CD?

4. Cloud Computing Research:

    a. Research the Enterprise Class Logging Analytics and Reporting tool called Splunk (at www.splunk.com). How would this tool be used by DevOps and what features in this tool help DevOps Engineers be proactive rather than reactive to application or infrastructure issues?

    b. Read the assigned textbook required readings for this topic. What data information is relevant and should be provided in a log file to support your application in the cloud? Provide three best practices that you should adhere to when adding logging to an application. Provide three issues or risks that could occur if inadequate logging is designed into an application.

    c. Research three tools that could support a CI/CD build pipeline. What are the tools and how are they used to support CI/CD?

    d. From Chapter 14 in the textbook, identify five capabilities that drive the definition of DevOps. What are the five capabilities and how are these used to help improve application development, testing, and delivery?

Documentation

Submit the following to the learning management system:

An activity report that contains the following on DevOps logging, monitoring, CI/CD, and cloud computing:

1. Cover sheet
2. Screenshots and questions on DevOps logging.
3. Screenshots and questions on DevOps monitoring.
4. Screencast and questions on DevOps CI/CD.

© 2023. Grand Canyon University. All Rights Reserved.

5. Cloud computing research questions.

Activity 7: Future Cloud Computing and Cloud Research

Overview

In this activity, students will:

1. Research future cloud-based development IDEs.
2. Research future cloud computing features, such as feature toggles and an A/B switch.

Execution

Execute this activity according to the following guidelines:

1. Explore cloud IDE.
    a. Research the features of the following cloud-based IDEs: VS Code (there are a number of cloud based online implementations), Visual Studio Online, Codenvy, Codeanywhere, and Cloud9 (only on AWS).
    b. Choose to import your test application into one of the following cloud-based IDEs: VS Code (there are a number of cloud based online implementations), Visual Studio Online, Codenvy, Codeanywhere, and Cloud9 (only on AWS).
    c. Identify 10 common day-to-day use cases that you perform using your desktop development IDE that would be required to support the development of your test application. For each use case:
        1) Analyze your chosen cloud IDE against the 10 use cases and validate if each of the use cases is supported in the cloud IDE.
        2) If the use case is supported, exercise that use case using the features of your chosen cloud IDE. While you are exercising the use case also evaluate if the feature in the cloud IDE is productive and easy to use such that you could use the cloud IDE to reasonable replace your desktop IDE.
        3) If the use case is not supported, summarize why that feature is important to your development and whether this would be a "must have" feature for you to do development using your cloud IDE.
    d. Write a 300- to 500-word research paper summarizing your observations and findings. Use the following table as a guide. Make sure to address whether you could use this cloud-based environment as a day to day development tool to replace your desktop IDE. Justify and rationalize your answer.

Use Case	Supported (Y/N)	Observations on Usability and Productivity	Other Notes

2. Cloud computing research:

© 2023. Grand Canyon University. All Rights Reserved.

a. Read the assigned textbook required readings for this topic. Answer the following questions:
    1) What is a feature flag or feature toggle?
    2) What is A/B testing?
    3) What is continuous delivery?
    4) What is continuous integration?
b. One trend in cloud computing is the ability to develop code in a cloud and browser-based IDE. Research two viable existing cloud and browser-based IDEs on the market. How might these cloud-based IDEs conceptually be used to lower the cost of developing code for a company? What are some disadvantages or features that are missing in the IDEs you researched when comparing them to a desktop-based IDE, such as Eclipse or Visual Studio?
c. Complete a Scrum retrospective. Document the following:
    1) What worked well in this class?
    2) What did not work well in this class?
    3) What would you like to see improved in the design of the class?

Documentation

Submit the following to the learning management system:

An activity report that contains the following on cloud IDE analysis and cloud computing:

1. Cover sheet
2. Results of your cloud IDE research.
3. Cloud computing research questions

© 2023. Grand Canyon University. All Rights Reserved.

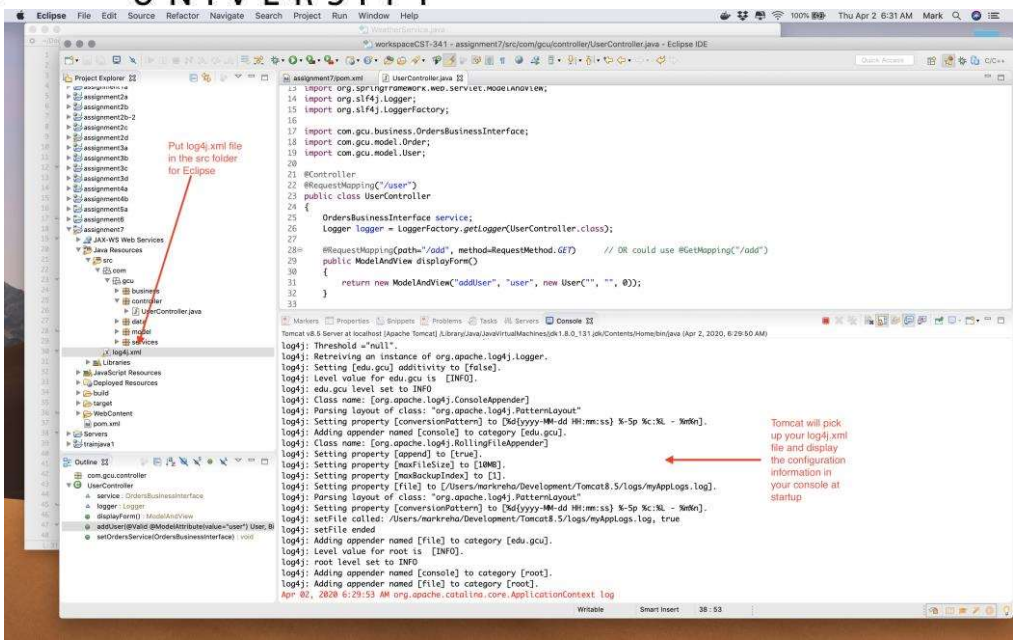
Appendix A – Logging Implementation Notes

To setup a log drain for Loggy in a PHP Laravel application you must develop a logger that exports its log statements to Loggly. See the instructions at the end of this Appendix for how Loggly can be integrated with Express, Angular, or React. The following illustrations provide an example logger that can be used. Also refer to the following article from sitepoint, "More Effective PHP Logging with Loggly.".

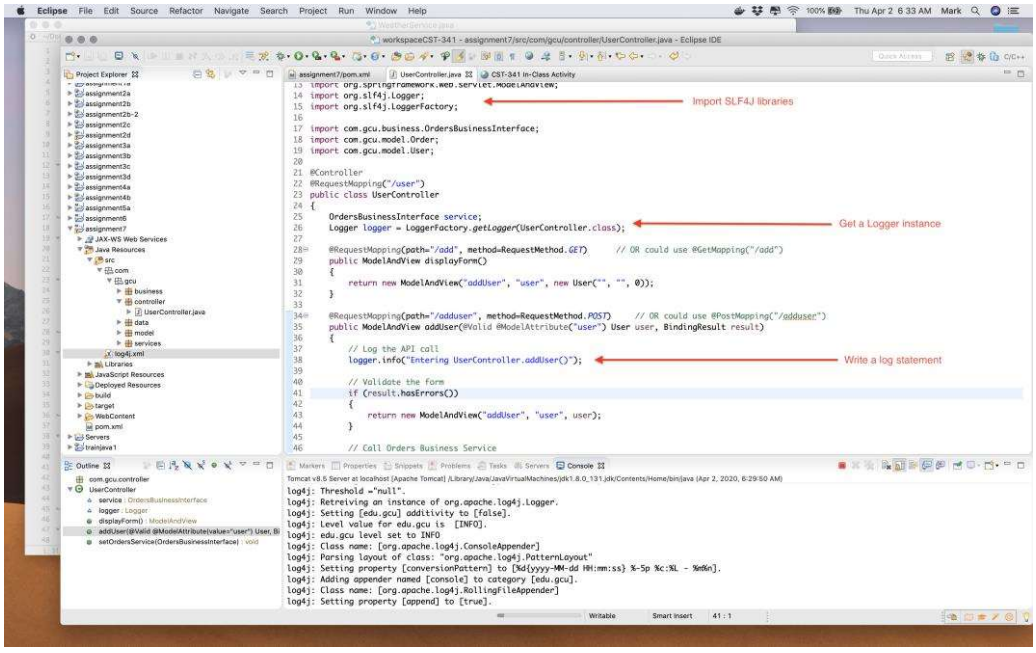
```php
<?php

namespace App\Services\Utility;

use Monolog\Logger;
use Monolog\Handler\LogglyHandler;

class MyLogger2 implements ILogger
{
    private static $logger = null;

    static function getLogger()
    {
        if (self::$logger == null)
        {
            self::$logger = new Logger('playlaravel');
            self::$logger->pushHandler(new LogglyHandler('[LOGGLY TOKEN]/tag/[LOGGLY TAG]', Logger::DEBUG));
        }
        return self::$logger;
    }

    public static function debug($message, $data=array())
    {
        self::getLogger()->addDebug($message, $data);
    }

    public static function info($message, $data=array())
    {
        self::getLogger()->addInfo($message, $data);
    }

    public static function warning($message, $data=array())
    {
        self::getLogger()->addWarning($message, $data);
    }

    public static function error($message, $data=array())
    {
        self::getLogger()->addError($message, $data);
    }
}
```

To setup a Java Spring application to support logging it is recommended to use the SLF4J logging framework with the Log4J log provider. The following screenshots provide some guidance for how to setup SLF4J in a standard Java Spring application.

© 2023. Grand Canyon University. All Rights Reserved.

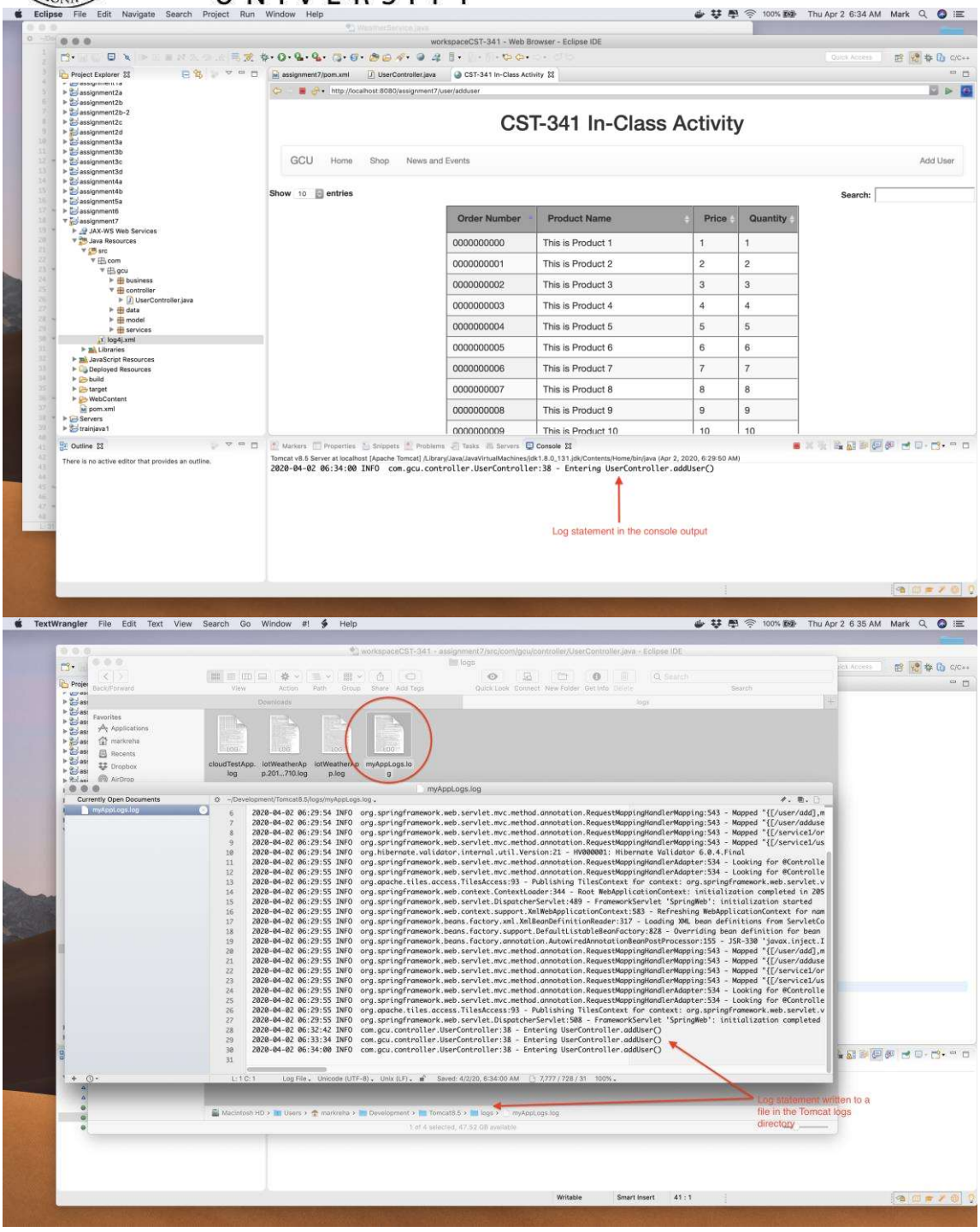


The Log4j Configuration file when run in your IDE should be placed in your project *src* directory



SLF4J can be imported, a Logger instantiated, and used to write log statements

© 2023. Grand Canyon University. All Rights Reserved.



Your log statements can be written to both the console and optionally to a Tomcat logs directory

© 2023. Grand Canyon University. All Rights Reserved.

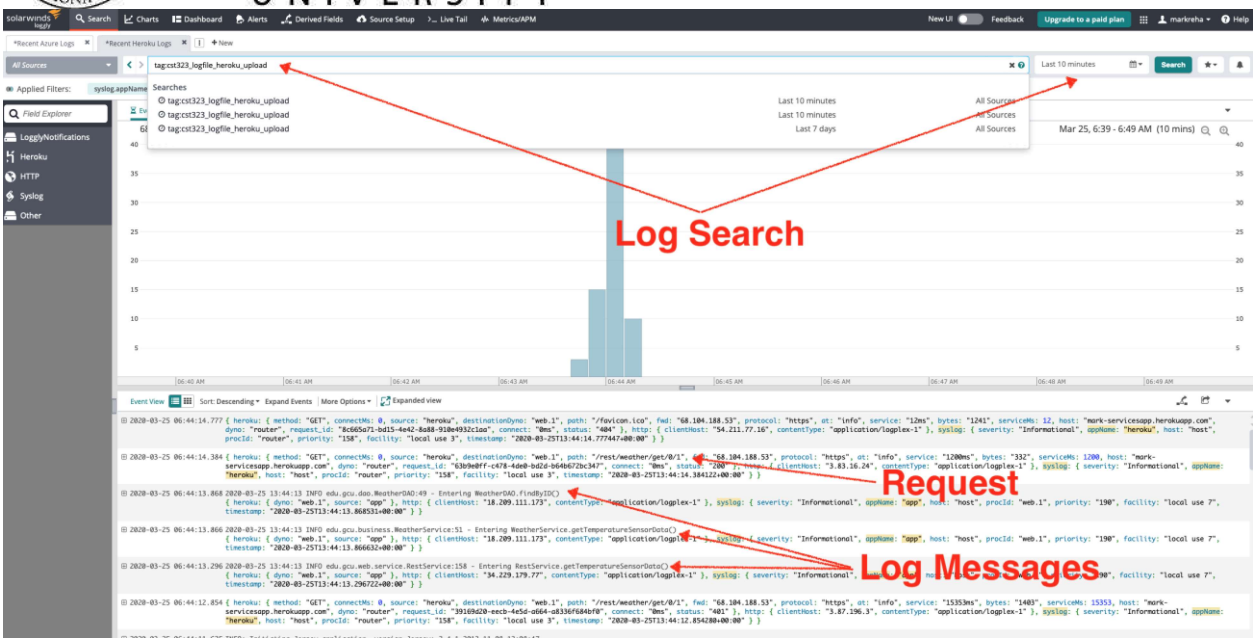
For the Cloud Platforms you must make sure your Maven POM is setup properly to copy the Log4J Configuration file to the WEB-INF/classes directory so it is in the Java Spring applications classpath at runtime.

© 2023. Grand Canyon University. All Rights Reserved.

An example Log4j Configuration that logs to the console and a log file within the Tomcat logs directory and that will work with the Cloud Platforms is shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="true" xmlns:log4j='http://jakarta.apache.org/log4j/'>
    <appender name="console" class="org.apache.log4j.ConsoleAppender">
        <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern"
          value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c:%L - %m%n" />
        </layout>
    </appender>
    <appender name="file" class="org.apache.log4j.RollingFileAppender">
        <param name="append" value="true" />
        <param name="maxFileSize" value="10MB" />
        <param name="maxBackupIndex" value="1" />
       <param name="file" value="${catalina.home}/logs/myAppLogs.log" />
        <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern"
            value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c:%L - %m%n" />
        </layout>
    </appender>

    <logger name="edu.gcu" additivity="false">
        <level value="INFO" />
        <appender-ref ref="console" />
        <appender-ref ref="file" />
    </logger>
    <root>
        <level value="INFO" />
        <appender-ref ref="console" />
        <appender-ref ref="file" />
    </root>
</log4j:configuration>
```

© 2023. Grand Canyon University. All Rights Reserved.



Example Loggly Queries

© 2023. Grand Canyon University. All Rights Reserved.

The following shows you how to send log events to Loggly Express, Angular, or React JavaScript frameworks. It is highly recommended that you apply the dependency injection and façade design patterns (whenever possible) to support a proper and extendable design. A custom functional utility library or class should be utilized in the implementation.

Express

1. Install the Loggly Node Module by running 'npm -g install loggly' from a terminal window in the root directory of your project.
2. Update the package.json file in your project to include the Loggly library dependency.

```
"dependencies": {
  "express": "^4.17.1",
  "loggly": "^1.1.1"
}
```

3. Import the Loggly library into your application and initialize Loggly using your Loggly CUSTOMER TOKEN and desired Loggly Log Event Tag.

```
const express = require('express')
const app = express()
const port = 3000

var loggly = require('loggly');
var logger = loggly.createClient({ token:"YOUR_CUSTOMER_TOKEN", subdomain:"localhost", sendConsoleErrors: false, tag:"YOUR LOG EVENT TAG" });
```

4. Write log events to Loggly using the log() method.

```
app.listen(port, () => {
    console.log(`Example app listening on port ${port}!`)
    logger.log("This is a test from Express app");
})
```

Angular

1. Add the Loggly JavaScript library to your application by using a script tag in the *index.html* file in the root directory of your project.

```
<head>
  <meta charset="utf-8">
  <title>Testapp</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script type="text/javascript" src="https://cloudfront.loggly.com/js/loggly.tracker-2.2.4.min.js"></script>
</head>
```

2. Create a file *typings.d.ts* file in the app and add the following declaration.

```
declare var LogglyTracker;
```

3. Initialize Loggly by adding the following code to the ngOnInit() method of the App Component using your Loggly CUSTOMER TOKEN and desired Loggly Log Event Tag. NOTE: this could be optimized by using Angular Services and Dependency Injection to prevent redundant initialization of Loggly.

© 2023. Grand Canyon University. All Rights Reserved.

```
export class AppComponent implements OnInit
{
  title = 'testapp';
  private loggly: any = new LogglyTracker();
  ngOnInit()
  {
    this.loggly.push({ logglyKey: 'YOUR CUSTOMER TOKEN', sendConsoleErrors: false, tag: 'YOUR LOG EVENT TAG' });
    console.log("This is a test Angular console");
    this.loggly.push("This is a test from Angular app!");
  }
}
```

4. Write log events to Loggly using the push() method.

```
this.loggly.push("This is a test from Angular app!");
```

React

1. Add the Loggly JavaScript library to your application by using a script tag in the *index.html* file in the *public* directory of your project.

```
<script type="text/javascript" src="https://cloudfront.loggly.com/js/loggly.tracker-2.2.4.min.js"></script>
```

2. Initialize Loggly by adding the following script in the *index.html* file using your Loggly CUSTOMER TOKEN and desired Loggly Log Event Tag.

```
<script>
  const loggly = new LogglyTracker();
  window.logger = { loggly: loggly }
  loggly.push({ logglyKey: 'YOUR CUSTOMER TOKEN', sendConsoleErrors: false, tag: 'YOUR LOG EVENT TAG' });
</script>
```

3. Write log events to Loggly using the push() method from your Loggly instance created in the previous step.

```
window.logger.loggly.push("This is a test from React app");
```

© 2023. Grand Canyon University. All Rights Reserved.

Refer to the Loggly query cheat sheet here.

Also refer to the following logging resources:

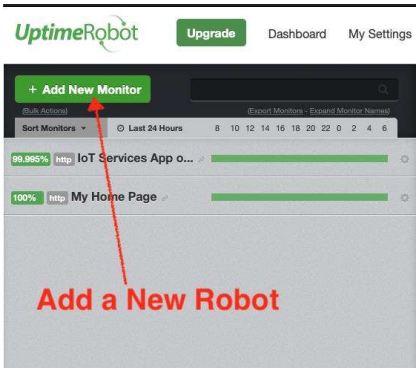
Java Logging Resources:
https://www.tutorialspoint.com/slf4j/index.htm
https://www.tutorialspoint.com/log4j/log4j_configuration.htm
https://www.baeldung.com/slf4j-with-log4j2-logback
http://www.slf4j.org/manual.html
https://dzone.com/articles/logging-with-slf4j
https://mkyong.com/logging/slf4j-logback-tutorial/

© 2023. Grand Canyon University. All Rights Reserved.

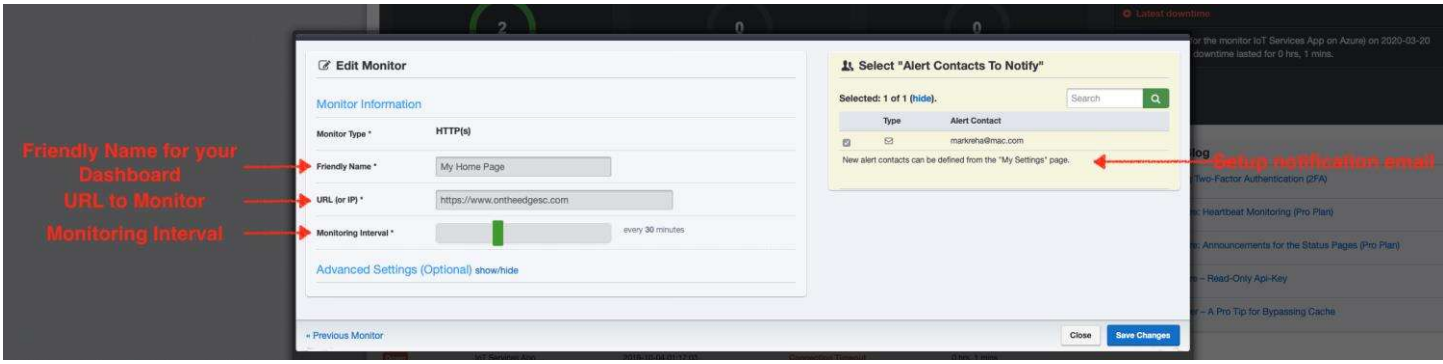
Appendix B – Application Availability Implementation Notes

To monitor an application for availability one Cloud Service Provider that can be used is Uptime Robot.

To setup a Robot, log into your Uptime Robot account, and then click on the Add New button.



First, setup a an Alert Contact in My Settings Setup for your account. Then setup a Robot configuration by selecting the HTTP(s) monitor type, then give your monitor a friendly name (this is the name that will show up on your Uptime Robot Dashboard, set the URL to your HTTP URL main page (or any page that does not require authentication), and finally set the monitor frequency. for every 5 minutes with an alert to your email address.



© 2023. Grand Canyon University. All Rights Reserved.