**Deployment Guide**

|  |  |
| --- | --- |
| UMGC Fall 2020 | Benjamin Fetterman, Benjamin Murray, Hanim Danur, James Cornelius, Robert Lee  SWEN 670 |

Project Plan Approvals

|  |  |  |
| --- | --- | --- |
| Name | Signature | Date |
| Approved by:  Dr. Mir Assadullah |  |  |
| Approved by:  "Stakeholder" |  |  |
| Approved by:  "Project Manager" | Robert Lee | 10/20/2020 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Name | Description of Change(s) |
| 1.0 | 10/20/2020 | Robert Lee | Initial Release |
| 2.0 | 11/3/2020 | Robert Lee | Updates based on production version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1.0 Introduction 5](#_Toc55321215)

[1.1 Purpose 5](#_Toc55321216)

[1.2 Intended Audience 5](#_Toc55321217)

[1.3 Technical Project Stakeholders 5](#_Toc55321218)

[2.0 Software Overview 5](#_Toc55321219)

[3.0 Configurations 6](#_Toc55321220)

[3.1 REST API 6](#_Toc55321221)

[3.1.1 Prerequisites 6](#_Toc55321222)

[3.1.2 Features 6](#_Toc55321223)

[3.2 IBM Watson® AI Chatbot 6](#_Toc55321224)

[3.2.1 Prerequisites 6](#_Toc55321225)

[3.2.2 Features 9](#_Toc55321226)

[3.2.3 Packages 9](#_Toc55321227)

[4.0 Software Installation 9](#_Toc55321228)

[4.1 REST API 9](#_Toc55321229)

[4.1.1 Prerequisites 9](#_Toc55321230)

[4.1.2 Installations Steps 9](#_Toc55321231)

[4.1.3 Configuration 11](#_Toc55321232)

[4.1.4 Administration 12](#_Toc55321233)

[4.2 Watson 12](#_Toc55321234)

[4.2.1 Prerequisites 12](#_Toc55321235)

[4.2.2 Installation Steps 12](#_Toc55321236)

[4.2.3 Configuration 12](#_Toc55321237)

[4.2.4 Configured Values 13](#_Toc55321238)

[4.2.5 Administration 13](#_Toc55321239)

[5.0 Known Bugs 13](#_Toc55321240)

[6.0 Testing Installation 13](#_Toc55321241)

[Appendix A: Acronyms and Abbreviations 15](#_Toc55321242)

[Appendix B: References 16](#_Toc55321243)

List of Tables

[Table 1 Stakeholders 5](#_Toc55321244)

[Table 2 Configuration Ports for WML 8](#_Toc55321245)

List of Figures

[**Figure 1 Install Deployment Dependencies** 10](#_Toc55321246)

[**Figure 2 Clone GitHub Repository** 10](#_Toc55321247)

[**Figure 3 Change Directory to Clone Repository** 11](#_Toc55321248)

[**Figure 4 Start Deployment to Azure** 11](#_Toc55321249)

[**Figure 5 Log In to Azure Account** 11](#_Toc55321250)

[**Figure 6 Integration Panel 12**](#_Toc55321251)

[**Figure 7 Script Tag for Deploying Chatbot 13**](#_Toc55321252)

1. Introduction

Purpose

The purpose of this deployment guide is to provide the technical information necessary to install and make operational the City of Pasadena City Chatbot application for a specific city, here unto described as the system, agent, or application.

Intended Audience

This deployment guide is intended for use by technical stakeholders in the planning, setup, and installation or deployment of the application. The specific roles of the technical stakeholders may include System Administrators, Analyst, Developers, and Database administrators.

The information contained here should allow said stakeholders to coordinate the deployment of a city chatbot instance for an individual city instance.

Technical Project Stakeholders

This section provides a list of all current stakeholders with an interest in this project at the time of release.

Table 1 Stakeholders

|  |  |  |
| --- | --- | --- |
| Name | E-mail address | Role |
| Professor Assadullah | mir.assadullah@faculty.umgc.edu | Stakeholder |
| Robert Lee | rlee97@student.umgc.edu | Project Manager |
| Benjamin Fetterman | bfetterman2@student.umgc.edu | Developer |
| Benjamin Murray | bmurray19@student.umgc.edu | Developer |
| Hanim Danur | hdanur@student.umgc.edu | Developer |
| James Cornelius | jcornelius10@student.umgc.edu | Developer |
| Glenn Goodlett | ggoodlett1@student.umgc.edu | DevOps |
| Dustin Emerson | demerson2@student.umgc.edu | DevOps |

1. Software Overview

The system allows users to be able to search for permits, zoning, and city regulations in the City of Pasadena’s Zoning and Planning website. The system uses IBM Watson Assistant to create a conversation with visitors to the website. The system shall welcome the user and asks the user to enter a complete address in Pasadena. If the address is found, the system will ask what type of information the user is requiring and display that information. If the system cannot determine the address, it will ask the user to enter the correct/full address. If the user requested unrelated permit/regulation, the system should display the not found message and requests the user to provide the city official contact information. Users can type their questions to get accurate information if available. If a response does not exist in the knowledge base of the chatbot or the system databases, the system will return a generic response. In the case where there is no direct response to the user question, the system shall redirect the user to refer FAQ and provide a URL to the website.

1. Configurations

REST API

* + 1. Prerequisites

The REST API consists of a Spring Boot application and a MySQL database. The REST API is used to perform address verification and database querying for the chatbot. The MySQL database contains information on the permits and regulations for the city of Pasadena. A system must have the following software and tools installed before deploying the REST API.

* make – version 4.2.1 or higher
* git – version 2.25.1 or higher
* docker – version 19.03.8 or higher
* azure-cli – version 2.0.81

Additionally, an Azure account is needed in order to be able to deploy on Azure. An Azure account can be created at <https://azure.microsoft.com/en-us/free/>.

* + 1. Features

The REST API application makes use of the following features found in Spring Boot.

* RestTemplate for MapQuest API calls
* JpaRepository for interfacing with the MySQL database
* Entity to represent tables in the MySQL database
* RestController to define the controller for the API
* RequestMapping to define the REST API routes
* RequestBody to access parameters sent from the chatbot

IBM Watson® AI Chatbot

IBM Watson®, or the agent, is a proprietary AI chatbot agent that provides a customized dialogue workflow that captures and processes user input based specified parameters. The purpose of the agent is to reduce the burden on customer service interactions by providing information on routine questions and operations.

* + 1. Prerequisites

The agent’s hardware and software requirements for installation for z/OS, V22 on IBM z and Linux on Z or Linux, are as follows:

* Set Up, of IBM Watson® engine within application prior to installing WML for z/OS – Only used if the agent itself is embedded within the system. The current configuration calls IBM Cloud infrastructure to access the agent.
  + z15, z14, z13®, or zEnterprise® EC12 system.
  + z/OS 2.4, 2.3, or 2.2.
  + For z/OS 2.4, apply PTFs UI64839, UI64940, UI64837, and UI64830.
  + For z/OS 2.3, apply PTFs UA98440, UI61308, UI61376, UI61747, and UI61375.
  + For z/OS 2.2, apply PTFs UA98441, UI62788, UI46658, UI62416, and UI62415.
  + z/OS Integrated Cryptographic Service Facility (ICSF).
  + z/OS OpenSSH.
  + IBM SDK for Node.js 12.15.0 or 12.16.1.
  + IBM 64-bit SDK for z/OS Java™ Technology Edition Version 8 Service Refresh 4 Fix Pack 6 (Java 8 SR4 FP6) or later. Set JAVA\_HOME to your installation directory.
  + Db2® 11 for z/OS or later.
  + IBM CICS® Transaction Server for z/OS 5.4.0 (or 5.3.0 with APAR PI63005 applied) only if you plan to configure and run a scoring service in a CICS region.

This information was gathered from, <https://www.ibm.com/support/pages>

* Calling IBM Watson® agent from IBM Cloud, Directed from Local Host
  + Any common OS such as Windows 10, MAC OS, and Linux – the local system will call to IBM Cloud to access the agent.
* Configuring ports for WML for z/OS base – Applicable if standing up the agent within the application outside of IBM Cloud.

| System or service | Port number | Outbound system | Inbound system |
| --- | --- | --- | --- |
| Db2® for z/OS | User-defined1 | z/OS system | Db2 subsystem |
| z/OS Spark master | 7077 or user-defined | z/OS system | z/OS Spark system |
| z/OS Spark master REST API | 6066 or user-defined | z/OS system | z/OS Spark system |
| z/OS Spark master UI | 8080 or user-defined | z/OS system | z/OS Spark system |
| Spark-integration service | 10080 or user-defined | z/OS system | z/OS system |
| Scoring service | User-defined2 | z/OS system | Liberty Profile for z/OS system |
| Scoring service | Dynamically assigned or user-defined3 | z/OS system | z/OS system |
| Jupyter Kernel Gateway | 8889 or user-defined | z/OS system | z/OS Spark system, Python run time for z/OS |
| Apache Toree | User-defined (A range of port numbers in consecutive order)4 | None | z/OS system |
| WMLz base UI service | 9888 or user-defined | Your network | z/OS system |
| WMLz base core services | 11442 or user-defined | z/OS system, Liberty Profile for z/OS, Python run time for z/OS | z/OS system |
| Configuration tool service | 5000 or user-defined | Your network | z/OS system |

(IBM, 2020)

Table 2 Configuration Ports for WML

* Port Configuration, Accessing IBM Watson® through IBM Cloud®
  + Customized port configuration is not required when accessing the agent through IBM Cloud®
    1. Features

IBM Watson® contains a variety of features. All of these features can be accessed through the IBM Cloud®. Below is the list of features:

* Natural Language Dialogue – Ability to interact with end user using natural language in a dialogue that is comparable to human to human interaction.
* Simplified Analysis – IBM Watson facilitates analysis of information by providing an AI engine that conducts the bulk of analysis.
* Automated Predictive Analytics – IBM Watson provides a framework where the data owner can define custom models.
* Accessible Advanced Analytics – IBM Watson provides a framework of advanced analytics that can be accessed through the same data model controller for automated predictive analytics.
* Speech to Text – IBM Watson provides the functionality for end-users to user verbal input technologies. The agent will convert the audio information into text for processing.
* Text to Speech – IBM Watson is able to output verbal audio.

(newgenapps, 2020)

* + 1. Packages

Not Applicable – Local host accessing IBM Watson® through IBM Cloud® will rely on IBM Cloud@ for maintenance operations.

1. Software Installation

### REST API

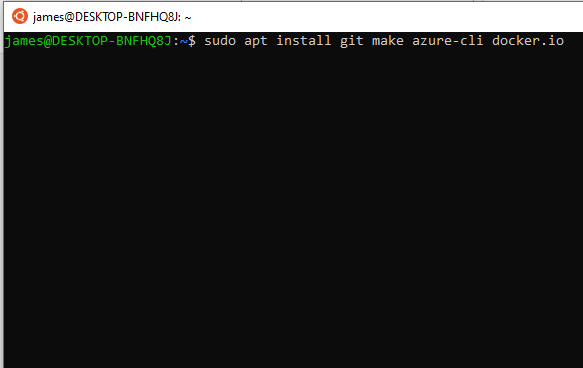
* + 1. Prerequisites

The deployment instructions are done from a bash shell on an Ubuntu 20.04 machine. For Windows users, the Windows Subsystem for Linux (WSL) can be used to run the deployment instructions instead. Instructions for installing WSL can be found at <https://docs.microsoft.com/en-us/windows/wsl/install-win10>. Choose Ubuntu 20.04 as the Linux distribution. The deployment instructions will be the same for Ubuntu 20.04 machines and the WSL except where noted.

* + 1. Installations Steps

The steps below outline all the steps needed to deploy the REST API to Azure. An Azure account is needed before running these steps.

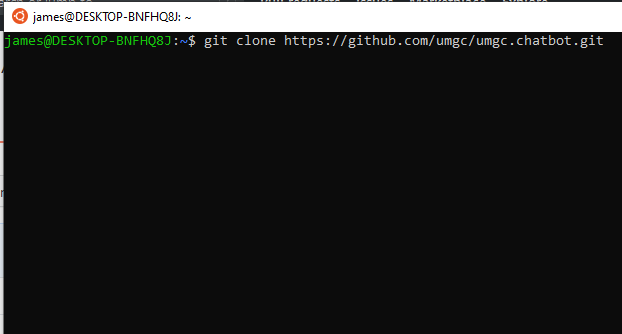
1. Install the packages mention in section 3.2.1.



**Figure 1 Install Deployment Dependencies**

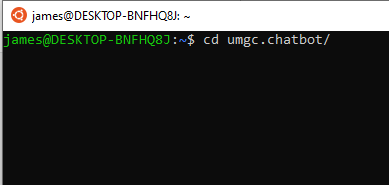
If using WSL, docker cannot be installed from the command line. It can be installed by following the instructions at <https://docs.docker.com/docker-for-windows/wsl/>.

1. Clone the GitHub repository containing the Spring Boot application.



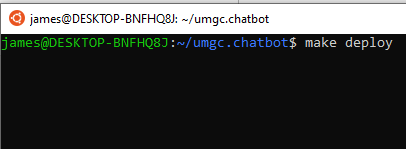
**Figure 2 Clone GitHub Repository**

1. Change the current working directory to the cloned repository.

****

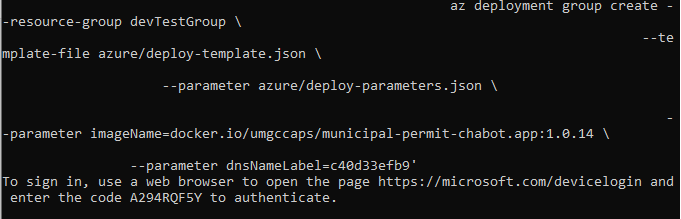
**Figure 3 Change Directory to Clone Repository**

1. Run make deploy to begin building the Docker images and deploying them to Azure.

****

**Figure 4 Start Deployment to Azure**

1. The make deploy process will require that you log in to your Azure account to complete the deployment. Follow the instructions from the make deploy output to log in.

****

**Figure 5 Log In to Azure Account**

1. Once the make deploy completes, go to your Azure portal to view the deployed application.
   * 1. Configuration

The deployment process has been configured to deploy the MySQL database with the Spring Boot application. The URL, username, and password to access the database will be injected into the Spring Boot application as part of the deployment process. The MapQuest Developer API key is injected as well during the deployment process. The city and state for the chatbot are also inject during the deployment process.

* + 1. Administration

Administrative functions such as restarting, deleting, and viewing network information on the deployed application can be done from the Azure portal.

### Watson

IBM Watson® when deploying from IBM Cloud® is a stand alone component that operates primarily outside of the application framework. If the agent resource is maintained on internal servers, outside of IBM Cloud services, a more in depth configuration manual is required. This section describes deploying the agent from IBM Cloud®.

* + 1. Prerequisites

The primary prerequisites for deploying IBM Watson® within the application are basic. It requires the normal use of a laptop, desktop or similar computer hardware device. The acceptable operating systems are the common and widely used Windows 10, MAC OS, Linus, etc.

* + 1. Installation Steps

Deploying and integrating Watson into the application will occur through integrating the webchat onto a website. This area is accessed when viewing the City Chatbot skills, in the “Integration” panel.

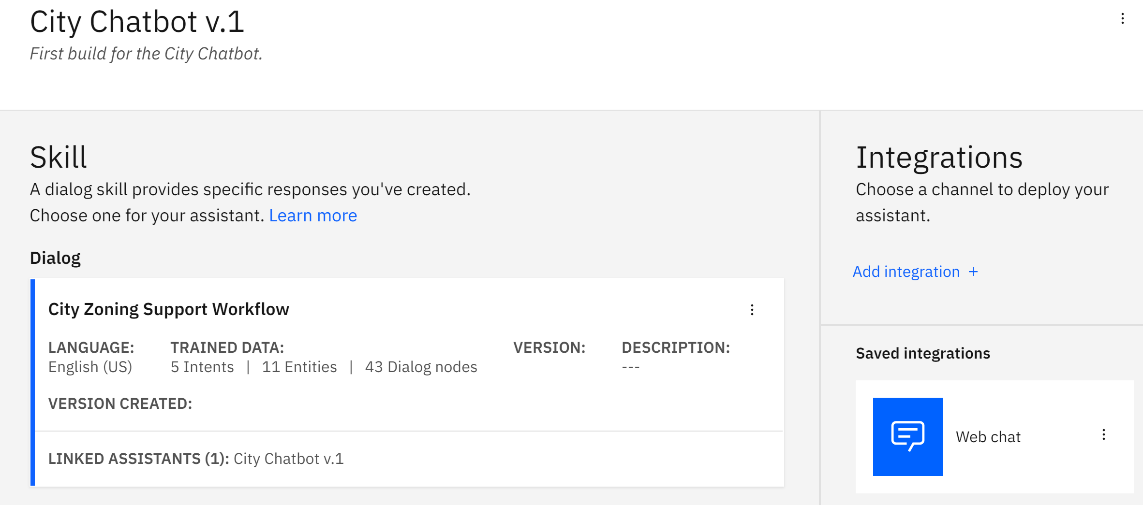


Figure 6 Integration Panel

* + 1. Configuration

The web chat integration portal provides a variety of configurable options. For the purposes of deployment, the “Embed” tab provides the agent script to embed within the website HTML code. The agent is a stand-alone component accessed by the host website calling IBM Cloud services. IBM Cloud services will run an instance of the agent and provide all processing and functionality.

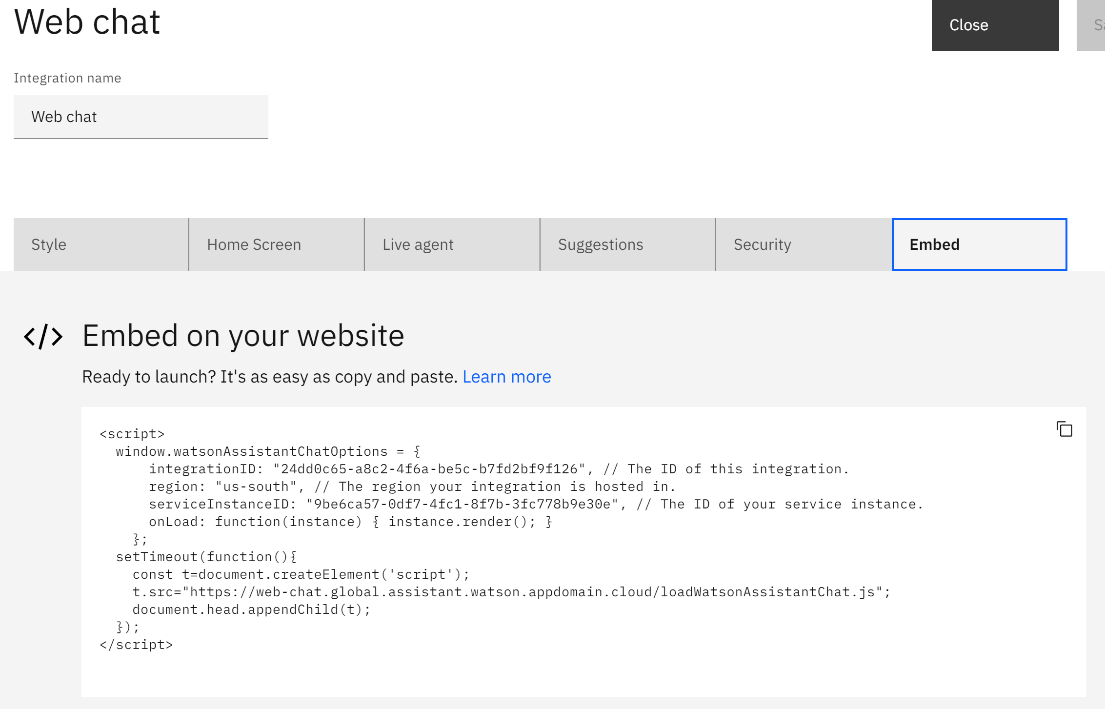


Figure 7 Script Tag for Deploying Chatbot

* + 1. Configured Values

For deployment, there are minimal configured values. Within the application, the script embodies the whole of the agent. IBM Cloud services autoconfigures the agent’s representative script when updates are made to the agent.

* + 1. Administration

Administration occurs through the same IBM Cloud® URL used to manage the agent.

https://cloud.ibm.com

IBM Cloud services allows for multiple integrations of the agent.

1. Known Bugs

Some of the known bugs within this application are the ability to provide detailed information to the user without the non-free version of the IBM Watson Chatbot. Additionally, the deployment of the application to Azure is depending upon an Azure account or a free instance of it. For production deployment the deployment team will need to identify additional resources to facilitate the deployment.

1. Testing Installation

The following steps should be taken to test the proper deployment and functionality of the chatbot application:

* Navigate a web browser to installation location.
* Verify Watson Conversation start.
* Verify proper address return to ensure geocoding API is configured correctly.
* Verify proper response to request for permit to ensure Database connection is configured correctly.

1. Acronyms and Abbreviations

|  |  |
| --- | --- |
| Acronym/Abbreviation | Definition/References |
| AI | Artificial Intelligence – An application that aims to mimic human intelligence. |
| API | Application Programming Interface |
| CH | Chatbot |
| CU | Conditional Use Permit |
| DevSecOps | Development, Security and Operations – Group of developers responsible for the deployment and security of an application. |
| ECUP | Expressive Use Permit |
| EPSP | East Pasadena Specific Plan |
| ETL | Extract, transfer, and load |
| FGSP | Fair Oaks/Orange Grove Specific Plan |
| GIS | Geographical Information System – System for working with geographical data. |
| HTTP | Hyper Text Transfer Protocol – A network protocol for specifying how servers and clients communicate with each other. |
| IDE | Integrated Development Environment |
| KML | Keyhold Markup Language – A markup language for visualizing geographical data. |
| LASP | Lincoln Avenue Specific Plan |
| MCUP | Minor Conditional Use Permit |
| MVC | Model-View-Controller |
| REST API | Representation State Transfer – An API for interacting with data. |
| SRS | Software Requirements Specification |
| UI | User Interface – The part of the application that users use to interact with the application. |
| URL | Uniform Resource Locator |
| WSL | Windows Subsystem for Linux |

1. References

Prerequisites and Maintenance for IBM Watson Machine Learning for z/OS, V2.2. (n.d.). IBM Watson Documentation. Retrieved October 20, 2020, from https://www.ibm.com/support/pages/prerequisites-and-maintenance-ibm-watson-machine-learning-zos-v22

Configuration Reference. (2020, May 6). Watson Documentation, Configuration Reference. https://cloud.ibm.com/docs/discovery?topic=discovery-configref