Associated Engineering
Team Discovery Channel

Release Documentation

Thursday, April 5, 2018
Version 1.0

Document Information

Revision History

Date	Version	Status	Prepared by	Comments
2018/04/05	1.0	Submitted	Ryan Koon, Andrew Chang, Shih-Chun (Joyce) Huang, Nicky Chan, Qiushan Zhao, Sebastian Wels-Lopez, Alan Wong	Demo day

Stakeholders

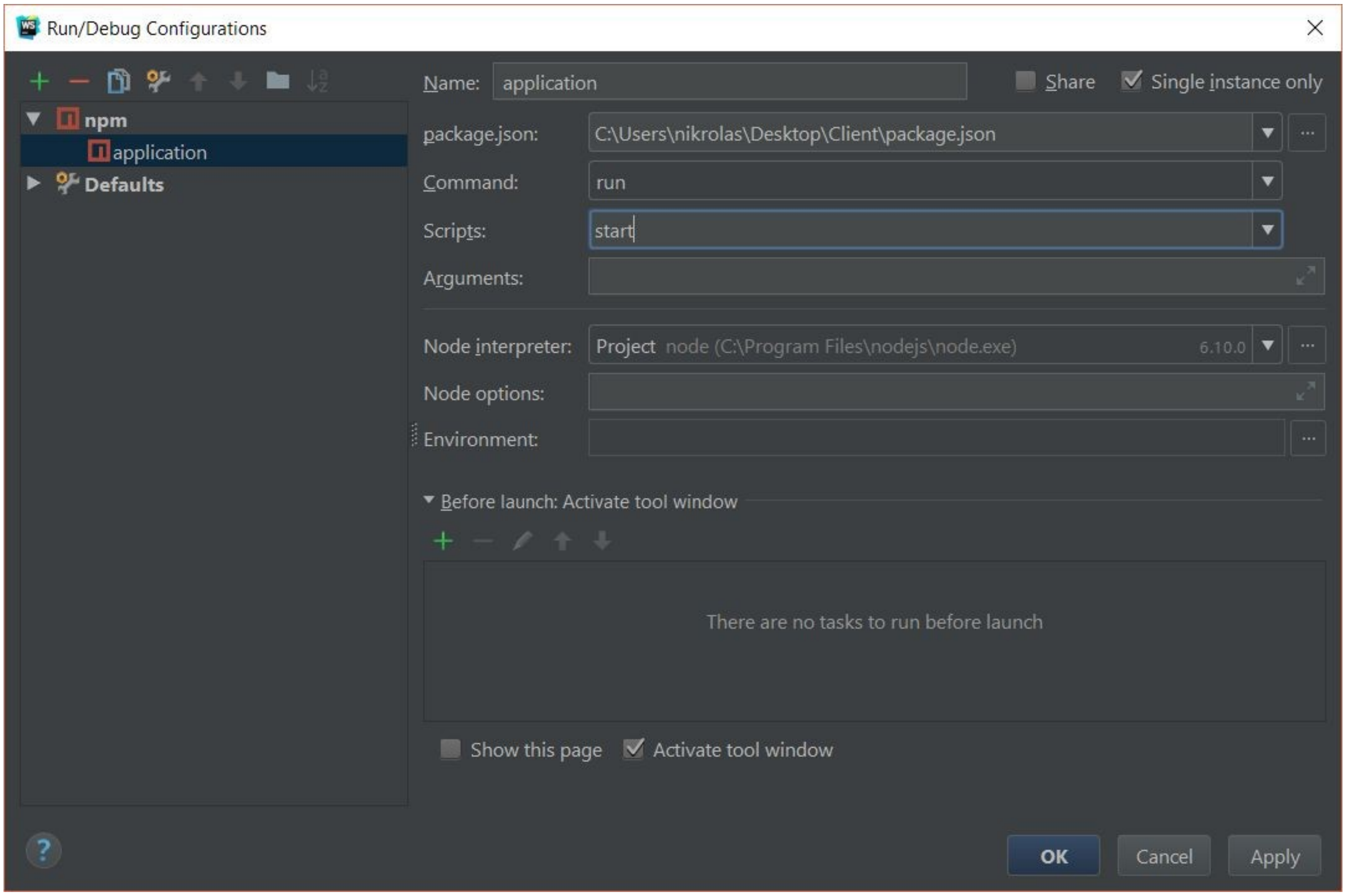
Name	**Role**	**Contact Details**
Shawn Goulet	Sponsor (Main Contact)	(reachable via TAs)
Steve Robinson	Sponsor	(reachable via TAs)
Julin Song	TA	(reachable on Piazza)
Iris Ren	Supporting TA	(reachable on Piazza)
Anna Scholtz	Supporting TA	(reachable on Piazza)
Ryan Koon	Project Manager / Front-end Developer	ryankoon@alumni.ubc.ca
Andrew Chang	Front-end Lead / Developer	[redacted]
Shih-Chun (Joyce) Huang	Back-end Lead / Developer	[redacted]
Nicky Chan	Front-end Developer	(reachable via Front-end Lead)
Qiushan Zhao	Back-end Developer	(reachable via Back-end Lead)
Sebastian Wels-Lopez	Back-end Developer	(reachable via Back-end Lead)
Alan Wong	Back-end Developer	(reachable via Back-end Lead)

Table of Contents

Front-end Configuration (Development)

Setting up local development environment

1. Install Node.js for Windows/Mac https://nodejs.org/en/
2. Install Yarn for Windows/Mac https://yarnpkg.com/en/
3. Install Git for Windows/Mac https://git-scm.com/downloads
4. Install Webstorm for Windows/Mac (or IDE of your choice) https://www.jetbrains.com/webstorm/
5. Drag “Client” file into storage location on computer
6. Within “Client” file, right click and select “Git Bash here”
7. Within git bash command line, type “yarn install”
8. In selected IDE, open project “Client”
9. In top right, select Edit Configuration”
10. Click + icon and select “npm”
11. Copy the following fields and click “Ok”



Name: application ☐ Share ☑ Single instance only

package.json: C:\Users\nikrolas\Desktop\Client\package.json

Command: run

Scripts: start

Arguments:

Node interpreter: Project node (C:\Program Files\nodejs\node.exe) 6.10.0

Node options:

Environment:

Before launch: Activate tool window

There are no tasks to run before launch

☐ Show this page ☑ Activate tool window

12. Press Play button on the top right and will open in browser of choice

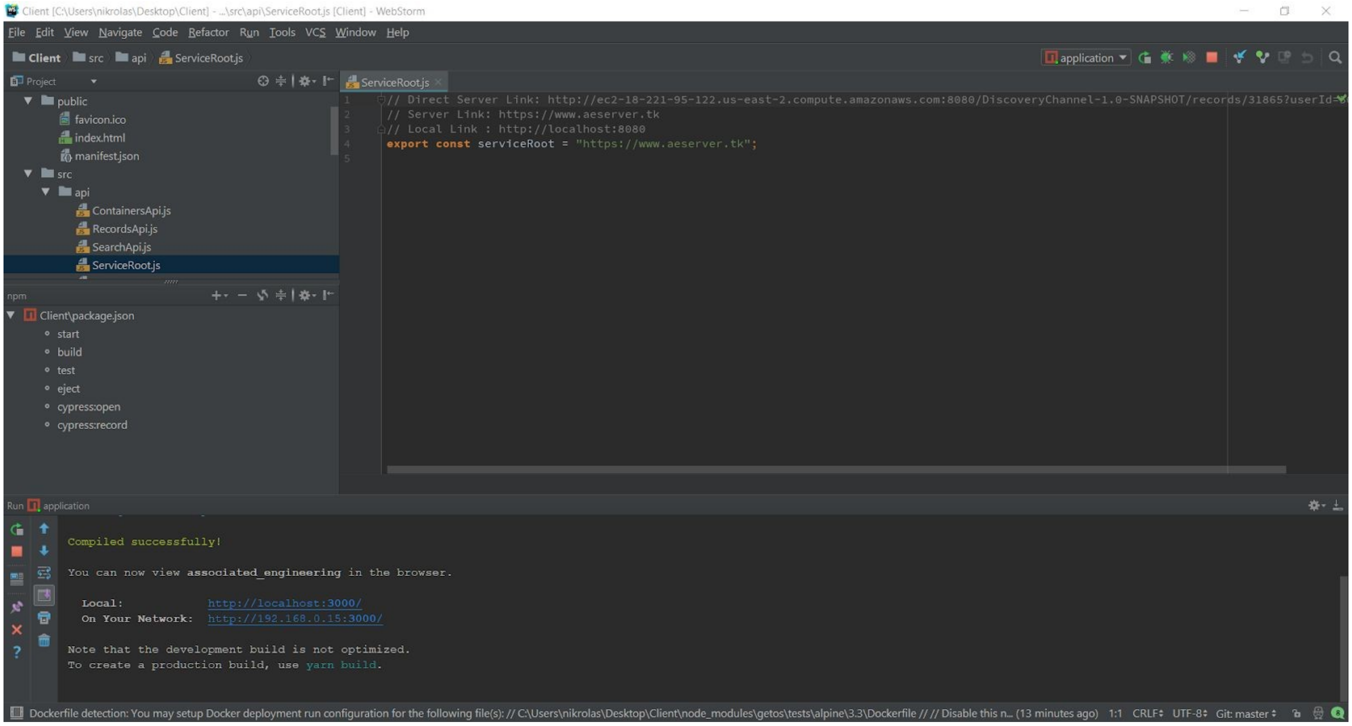
Connecting to local server for application

1. Follow up to step 11. in **Setting up development environment**
2. In “src->api->ServiceRoot.js”, change serviceRoot to url for local link

Running Cypress Tests

1. Follow up to step 12. in **Setting up development environment** (requires application running)
2. Select “View -> Tool Window -> Npm”

3. Double click “cypress:open” in npm window

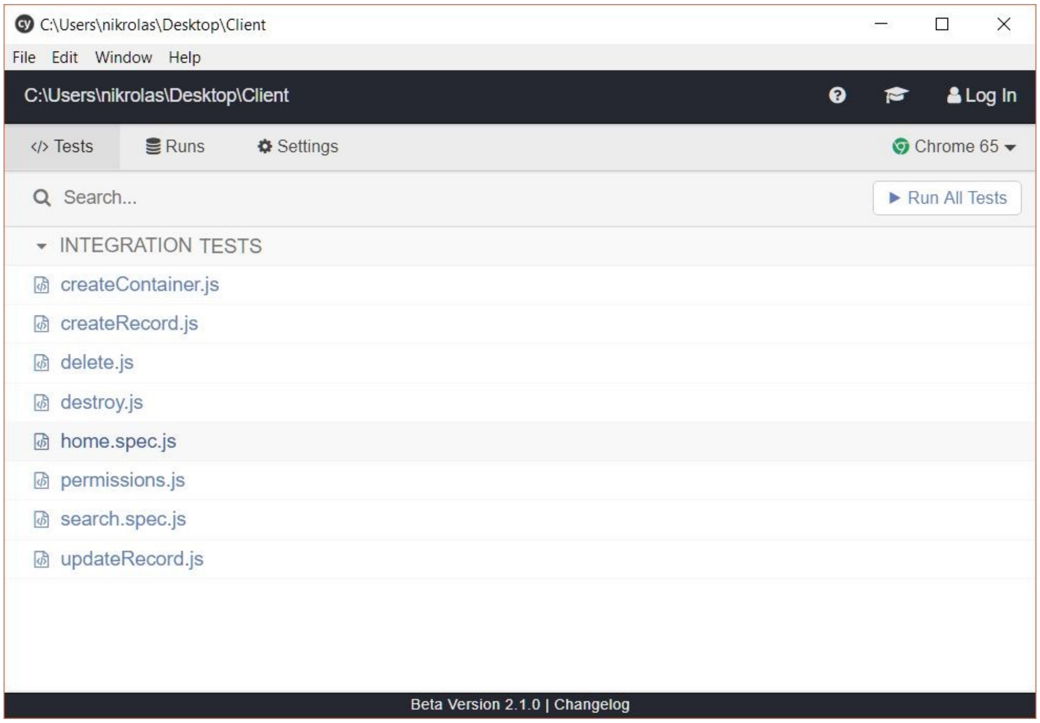


```
// Direct Server Link: http://ec2-18-221-95-122.us-east-2.compute.amazonaws.com:8080/DiscoveryChannel-1.0-SNAPSHOT/records/31865?userId=
// Server Link: https://www.aeserver.tk
// Local Link : http://localhost:8080
export const serviceRoot = "https://www.aeserver.tk";
```

```
Compiled successfully!

You can now view associated_engineering in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.15:3000/

Note that the development build is not optimized.
To create a production build, use yarn build.
```

Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): // C:\Users\nikrolas\Desktop\Client\node_modules\getos\tests\alpine\3.3\Dockerfile // // Disable this n... (13 minutes ago) 1:1 CRLF÷ UTF-8÷ Git: master ÷

4. Select Run All Tests for automatic testing



Front-end Deployment (Staging and Production)

Continuous Integration on GitLab

The Client project contains a YAML file for continuous integration and deployment to Azure.
In the root of the project you will find the file called ".gitlab-ci.yml".
It runs the latest version of node using a docker container, caches the library files in node_modules, and executes two different stages: build and deploy.

In the build stage, you can see that it is running yarn command to build the production version of the project to ensure the project files can be minified and code errors such as unused variables.

The deploy stages are divided up between staging and production. We use lftp to mirror the build folder containing our minified files. Deploy stages run on the master branch with automatic deployment to staging for every commit and production manually by clicking deploy on Gitlab. Secret variables configured on GitLab starts with the dollar sign (e.g. $FTP_PASSWORD_STAGING). They can be set up via CI/CD > Secret Variables on GitLab.

Below are more information about deployment to Azure. For documentation on other deployment options, visit the create-react-app documentation at:
https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#deployment

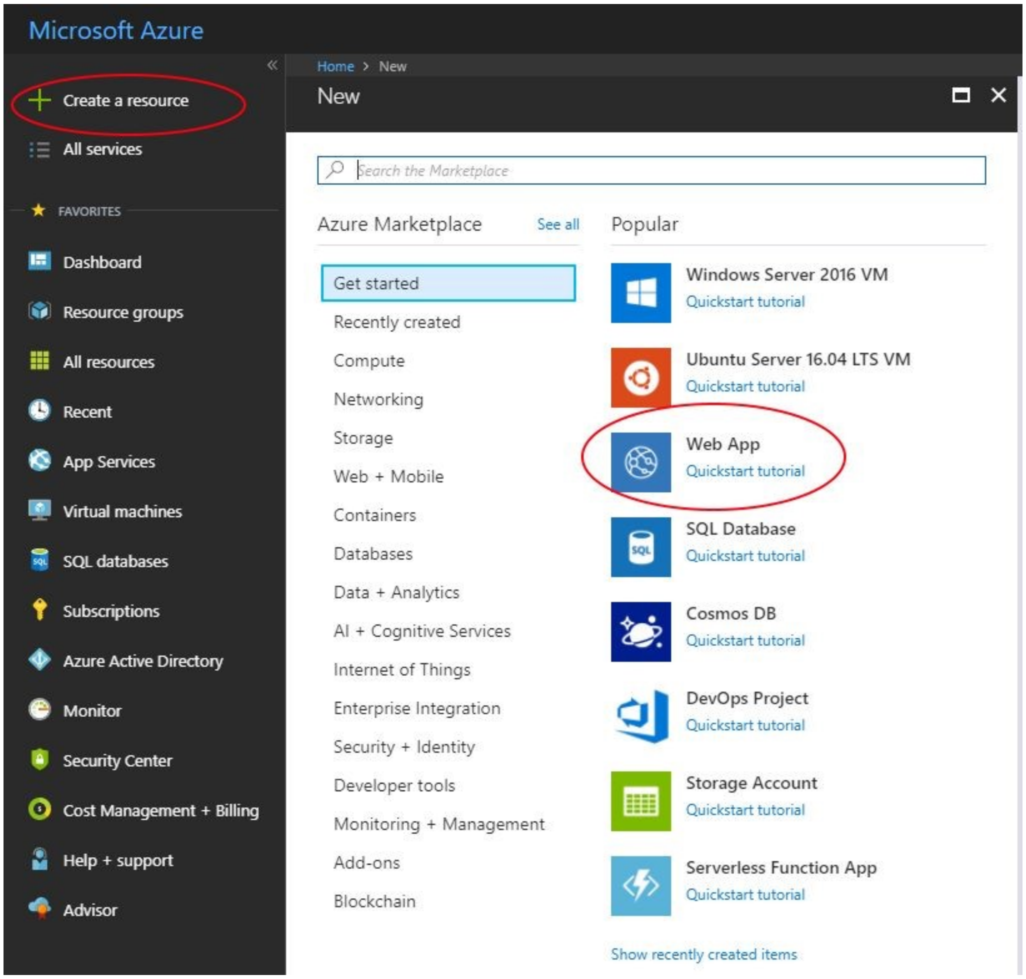
Deployment to Azure

1. Create and run React application

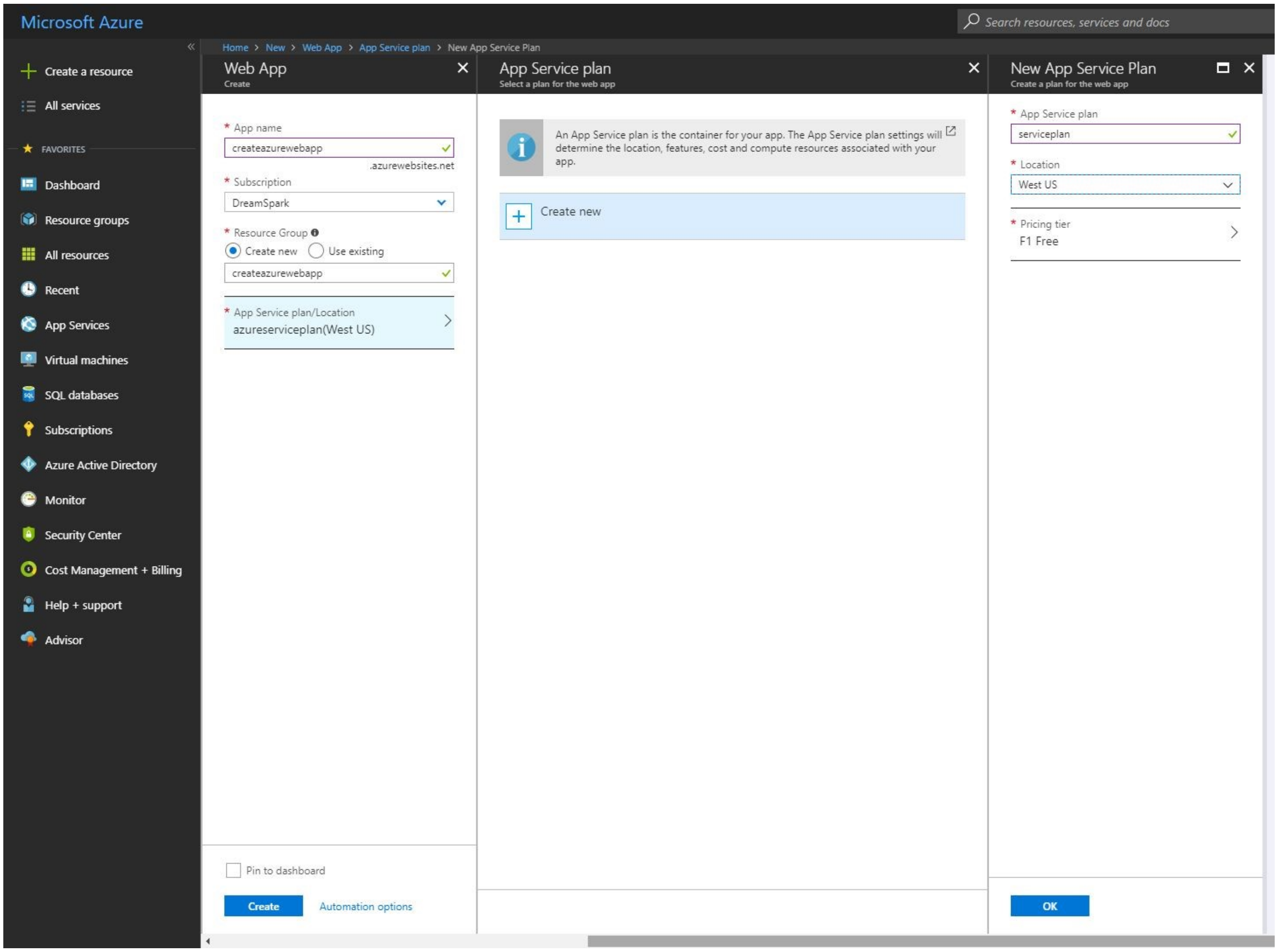
```
npx create-react-app my-app
cd my-app
npm start
```

More details can be found here: https://github.com/facebook/create-react-app#quick-overview

2. Create Web App on Azure



3. Assign to an existing app service plan or create a new one



Web App
Create

* App name: createazurewebapp .azurewebsites.net

* Subscription: DreamSpark

* Resource Group: ● Create new ○ Use existing
createazurewebapp

* App Service plan/Location: azureserviceplan(West US)

□ Pin to dashboard

App Service plan
Select a plan for the web app

An App Service plan is the container for your app. The App Service plan settings will determine the location, features, cost and compute resources associated with your app.

New App Service Plan
Create a plan for the web app

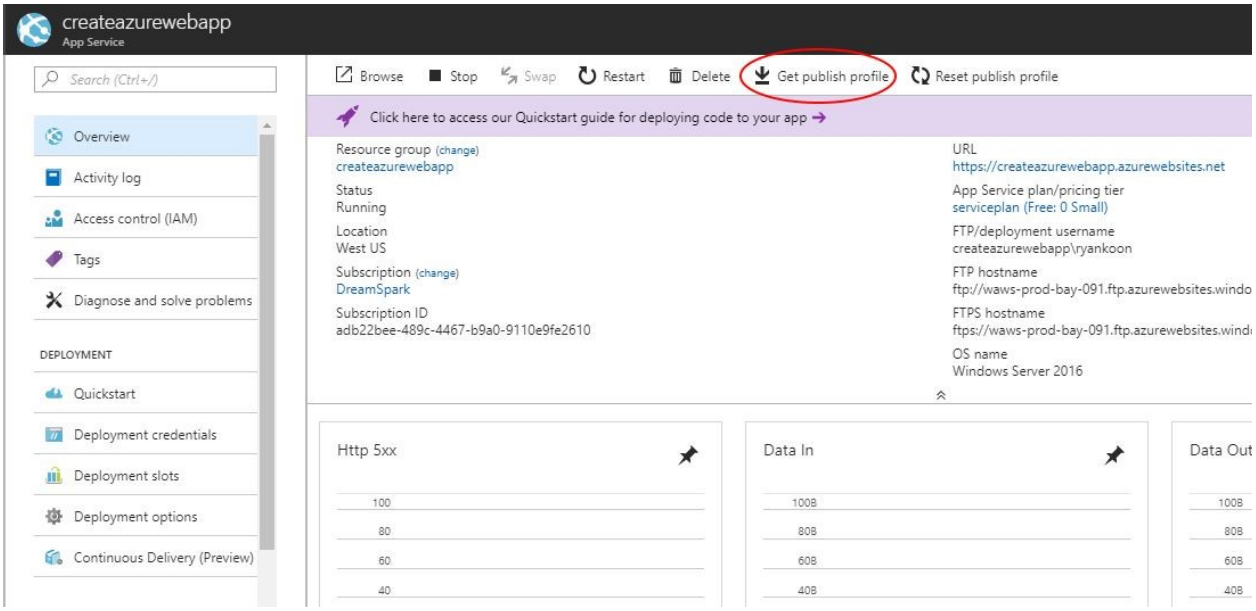
* App Service plan: serviceplan

* Location: West US

* Pricing tier: F1 Free

4. Download publish profile

From the navigation bar or search:
App Services > createazurewebapp > Overview



Click here to access our Quickstart guide for deploying code to your app →

Resource group (change)
createazurewebapp

Status
Running

Location
West US

Subscription (change)
DreamSpark

Subscription ID
adb22bee-489c-4467-b9a0-9110e9fe2610

URL
https://createazurewebapp.azurewebsites.net

App Service plan/pricing tier
serviceplan (Free: 0 Small)

FTP/deployment username
createazurewebapp\ryankoon

FTP hostname
ftp://waws-prod-bay-091.ftp.azurewebsites.windo

FTPS hostname
ftps://waws-prod-bay-091.ftp.azurewebsites.wind

OS name
Windows Server 2016

This file contains the credentials and ftp site for uploading files at a later step.

5. Create a Web.config file

We will need to rewrite urls, otherwise the server on Azure attempts to look for files based on the url instead of loading the correct React components through the React router (if it is set up)

Here is an example:

```
<configuration>
   <system.webServer>
       <handlers>
           <!-- indicates that the app.js file is a node.js application to be handled
by the iisnode module -->
           <add name="iisnode" path="server.js" verb="*" modules="iisnode" />
       </handlers>
       <rewrite>
           <rules>
               <!-- Don't interfere with requests for logs -->
               <rule name="LogFile" patternSyntax="ECMAScript" stopProcessing="true">
                   <match url="^[a-zA-Z0-9_\-]+\.js\.logs\/\d+\.txt$" />
               </rule>
               <!-- Don't interfere with requests for node-inspector debugging -->
               <rule name="NodeInspector" patternSyntax="ECMAScript"
stopProcessing="true">
                   <match url="^server.js\/debug[\/]?" />
               </rule>

               <rule name="React Routes" stopProcessing="true">
                   <match url=".*" />
                   <conditions logicalGrouping="MatchAll">
                       <add input="{REQUEST_FILENAME}" matchType="IsFile"
negate="true" />
                       <add input="{REQUEST_FILENAME}" matchType="IsDirectory"
negate="true" />
                       <add input="{REQUEST_URI}" pattern="^/(api)" negate="true" />
                   </conditions>
                   <action type="Rewrite" url="/" />
               </rule>
           </rules>
       </rewrite>
   </system.webServer>
</configuration>
```

If you are deploying a static web app without a back-end, you should not not need the handlers or rules for server.js above.

More details can be found here:
https://medium.com/@to_pe/deploying-create-react-app-on-microsoft-azure-c0f6686a4321

In addition to the application files, we will need to transfer this file to Azure.

6. Build React app

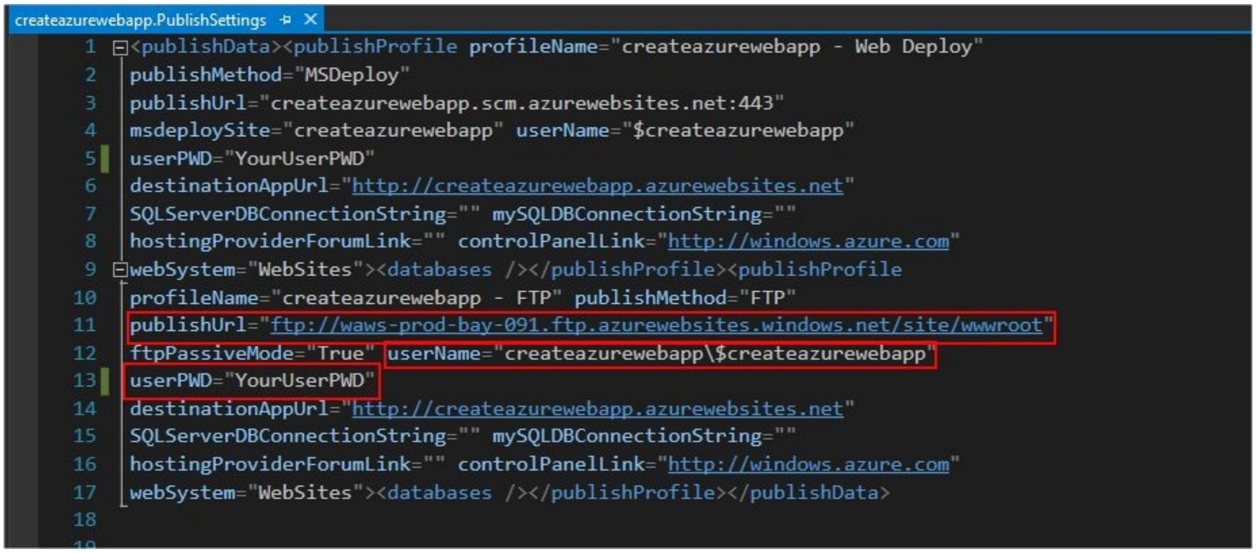
Create a production build of your app.

```
npm run build
```

More details can be found here:
https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#deployment

7. Transfer files via ftp

Sample Publish Profile:



createazurewebapp.PublishSettings

```
<publishData><publishProfile profileName="createazurewebapp - Web Deploy"
publishMethod="MSDeploy"
publishUrl="createazurewebapp.scm.azurewebsites.net:443"
msdeploySite="createazurewebapp" userName="$createazurewebapp"
userPWD="YourUserPWD"
destinationAppUrl="http://createazurewebapp.azurewebsites.net"
SQLServerDBConnectionString="" mySQLDBConnectionString=""
hostingProviderForumLink="" controlPanelLink="http://windows.azure.com"
webSystem="WebSites"><databases /></publishProfile><publishProfile
profileName="createazurewebapp - FTP" publishMethod="FTP"
publishUrl="ftp://waws-prod-bay-091.ftp.azurewebsites.windows.net/site/wwwroot"
ftpPassiveMode="True" userName="createazurewebapp\$createazurewebapp"
userPWD="YourUserPWD"
destinationAppUrl="http://createazurewebapp.azurewebsites.net"
SQLServerDBConnectionString="" mySQLDBConnectionString=""
hostingProviderForumLink="" controlPanelLink="http://windows.azure.com"
webSystem="WebSites"><databases /></publishProfile></publishData>
```

We will use lftp to transfer files to Azure.
The three values we will need from the publish profile are:

- publishUrl
- userName
- userPWD

lftp command:

```
lftp -u $FTP_USER,$FTP_PASSWORD $FTP_URL -e "mirror -R -p -x=$SKIP_FILES
$SOURCE_FOLDER $TARGET_FOLDER"
```

Using the sample publish profile:

```
lftp -u createazurewebapp\\\$createazurewebapp,YourUserPWD
ftp://waws-prod-bay-091.ftp.azurewebsites.windows.net/site/wwwroot -e "mirror -R
-p -x=node_modules build/. ./"
```

```
mirror   - replaces all the files in the target folder to match the source
-R       - transfers files from $SOURCE_FOLDER to $TARGET_FOLDER
-p       - do not set file permissions (needed when deploying from GitLab)
-x=...   - skip matching files
build/. - mirror the contents of the build folder
./       - target folder (wwwroot)
```

The lftp documentation can be found here:
https://lftp.yar.ru/lftp-man.html

Note: The username has escape characters for the backslash and dollar sign.

If you are using windows, you can install the “Windows Subsystem for Linux” to run these commands:
https://docs.microsoft.com/en-us/windows/wsl/install-win10

Do not forget to copy the web.config file.
You can copy the file into the build folder prior to running lftp.

In summary, these are the commands for uploading via lftp:

```
cp web.config build
apt-get update
apt-get install lftp
lftp -u createazurewebapp\\\$createazurewebapp,YourUserPWD
ftp://waws-prod-bay-091.ftp.azurewebsites.windows.net/site/wwwroot -e "mirror -R
-p -x=node_modules build/. ./"
```

8. Deploying from GitLab (optional)

Sample *.gitlab-ci.yml* (to be placed in project root):

```
image: node:latest

cache:
 paths:
 - node_modules/

stages:
 - deploy

deploy_production:
 stage: deploy
 script:
 - yarn install
 - yarn run build
 - cp web.config build
 - apt-get update
 - apt-get install lftp
 - lftp -u $FTP_USER,$FTP_PASSWORD $FTP_URL -e "mirror -R -p -x=node_modules build/. ./"
 environment:
    name: Production
    url: https://createazurewebapp.azurewebsites.net
 only:
 - master
```

image - *run in a docker image with the latest node version*

cache - *cache node_modules folder to speed up updating dependencies*

stages - *defines the stages of the pipeline*

deploy_production - *user defined name of a pipeline job*

script - *commands to run for the job*

environment - *associate the deployment job with an environment to be able to track deployed commits in different environments on GitLab*

only - *the job will only run on the specified branch*

The above sample yaml uses secret variables (e.g. `$FTP_PASSWORD)` that are configured on GitLab for each project (Settings > CI/CD > Secret Variables).
The $ is used by GitLab to indicate a secret variable.
To escape this character for the Azure Username, use two dollar signs.

Example:
Secret Variable: `FTP_USER`
Value: `createazurewebapp\$$createazurewebapp`

Other Resources

Facebook's documentation on deploying React apps to various targets (e.g. Azure, AWS, Firebase, etc.):
https://github.com/facebook/create-react-app/blob/master/packages/react-scripts/template/README.md#deployment

Kudu Services for your Azure web app:
https://<azureappname>.scm.azurewebsites.net/

Retrieve public key of via Kudu (to pull from GitLab):
https://<azureappname>.scm.azurewebsites.net/api/sshkey?ensurePublicKey=1
Add to GitLab via Settings > Repository > Deploy Keys

GitLab YAML configuration:
https://docs.gitlab.com/ce/ci/yaml/README.html#gitlab-ci-yml

Back-end Configuration

Database

Development

The database is a local MySQL instance.
Download MySQL installer.
Download and install MySQL Server, and MySQL Workbench
Connect to your local database instance from the MySQL Workbench
Open and run the following scripts from MySQL Workbench

RecordR-Schema.sql
RecordR-Data.sql
toDeploy.sql
populateUserRoles.sql
populateUserLocations.sql
addContainersTableFields.sql

Production

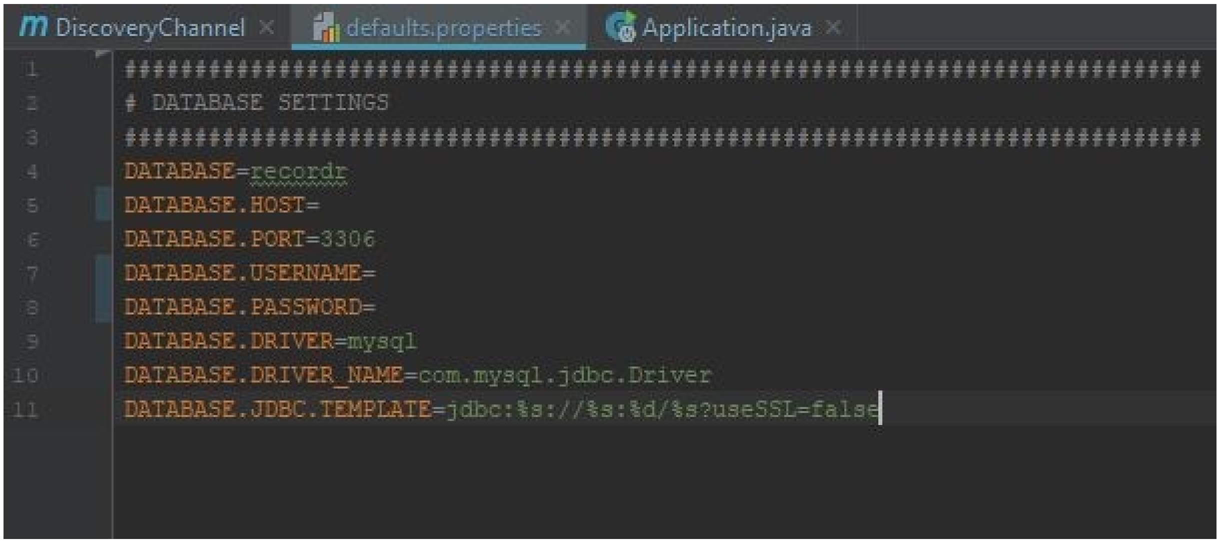
The database is an AWS RDS (Amazon Web Services Relational Database Service) db.t2.micro instance running MySQL 5.7.19.
From the AWS console, select RDS, then Launch a DB instance to begin.
Select MySQL, following the prompts and configuring as needed, making sure it has public accessibility set to yes. For our installation, most (free tier) default settings were sufficient. We selected MySQL 5.7.19 which was the latest at time of set up.
Connect to your AWS RDS instance from the MySQL Workbench using the credentials from set up and hostname from the RDS console.
Open and run the following scripts from MySQL Workbench

RecordR-Schema.sql
RecordR-Data.sql
toDeploy.sql
populateUserRoles.sql
populateUserLocations.sql
addContainersTableFields.sql

Server

You will need to do some common setup for both the development and production environments. Environment specific instructions will come later.

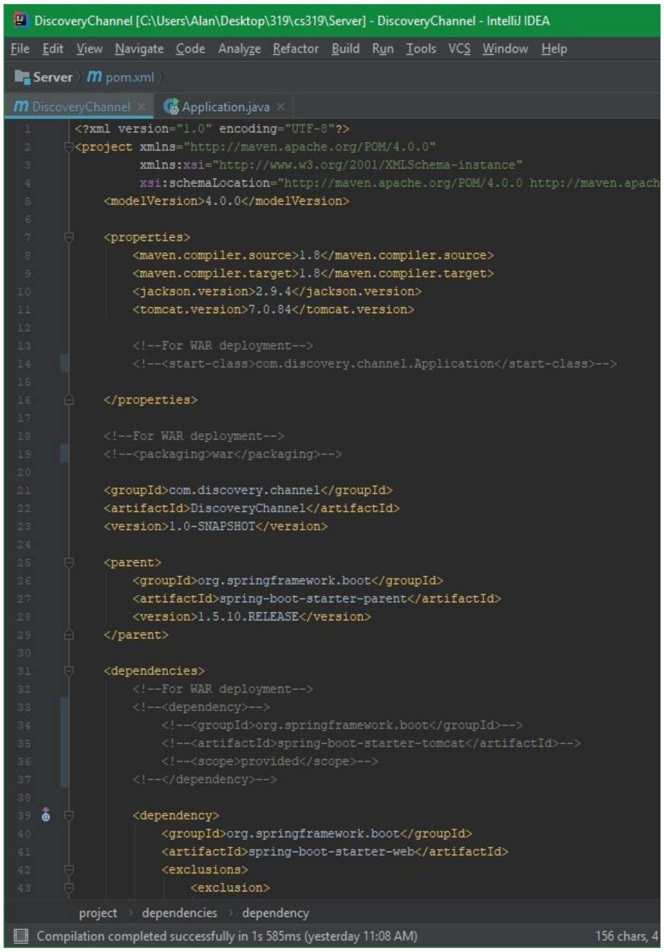
Download and install IntelliJ IDEA 2017.3.4 (Ultimate Edition)
Select File > Open > select “Server” directory
In the file Server\src\main\config\defaults.properties fill out the appropriate properties for connecting to your database of choice. Most importantly, host, username, and password.



```
#############################################################################
# DATABASE SETTINGS
#############################################################################
DATABASE=recordr
DATABASE.HOST=
DATABASE.PORT=3306
DATABASE.USERNAME=
DATABASE.PASSWORD=
DATABASE.DRIVER=mysql
DATABASE.DRIVER_NAME=com.mysql.jdbc.Driver
DATABASE.JDBC.TEMPLATE=jdbc:%s://%s:%d/%s?useSSL=false
```

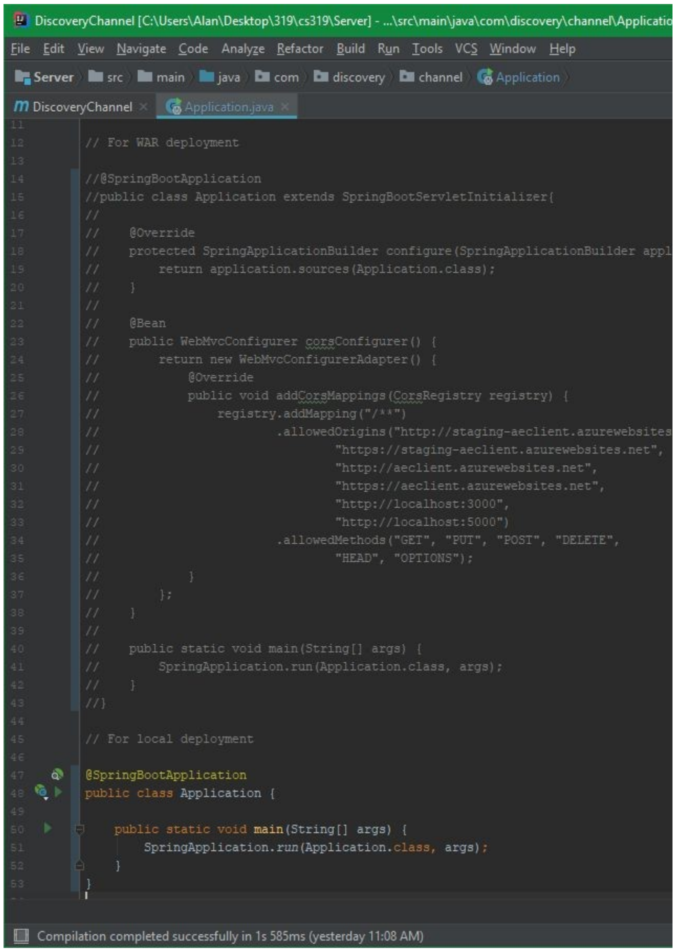
Development
The server is a local Spring Boot application. (for production environment, continue to next section)

We have included options for both local deployment and WAR packaging in the source code. To continue with local deployment, ensure that the sections commented “For WAR deployment” are commented, and the sections commented “For local deployment” are not. The relevant sections of code are in pom.xml (3) and Application.java (1).


DiscoveryChannel [C:\Users\Alan\Desktop\319\cs319\Server] - DiscoveryChannel - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Server
pom.xml
DiscoveryChannel
Application.java
project
dependencies
dependency
156 chars, 4

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apach
    <modelVersion>4.0.0</modelVersion>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <jackson.version>2.9.4</jackson.version>
        <tomcat.version>7.0.84</tomcat.version>

        <!--For WAR deployment-->
        <!--<start-class>com.discovery.channel.Application</start-class>-->

    </properties>

    <!--For WAR deployment-->
    <!--<packaging>war</packaging>-->

    <groupId>com.discovery.channel</groupId>
    <artifactId>DiscoveryChannel</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.10.RELEASE</version>
    </parent>

    <dependencies>
        <!--For WAR deployment-->
        <!--<dependency>-->
            <!--<groupId>org.springframework.boot</groupId>-->
            <!--<artifactId>spring-boot-starter-tomcat</artifactId>-->
            <!--<scope>provided</scope>-->
        <!--</dependency>-->

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <exclusions>
                <exclusion>
```

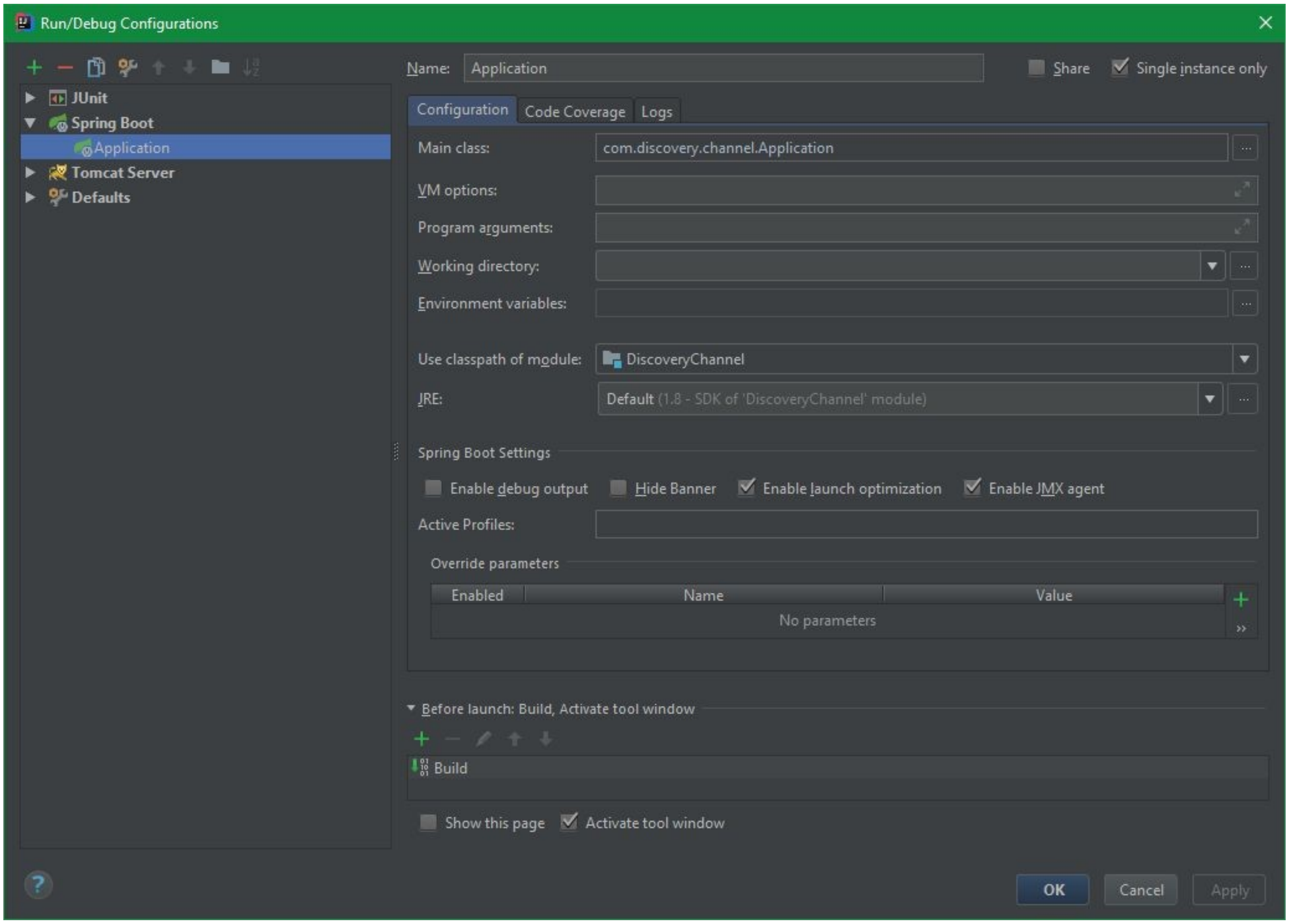
Compilation completed successfully in 1s 585ms (yesterday 11:08 AM)


DiscoveryChannel [C:\Users\Alan\Desktop\319\cs319\Server] - ...\src\main\java\com\discovery\channel\Applicatio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Server
src
main
java
com
discovery
channel
Application
DiscoveryChannel
Application.java

```java

// For WAR deployment

//@SpringBootApplication
//public class Application extends SpringBootServletInitializer{
//
//    @Override
//    protected SpringApplicationBuilder configure(SpringApplicationBuilder appl
//        return application.sources(Application.class);
//    }
//
//    @Bean
//    public WebMvcConfigurer corsConfigurer() {
//        return new WebMvcConfigurerAdapter() {
//            @Override
//            public void addCorsMappings(CorsRegistry registry) {
//                registry.addMapping("/**")
//                        .allowedOrigins("http://staging-aeclient.azurewebsites
//                                "https://staging-aeclient.azurewebsites.net",
//                                "http://aeclient.azurewebsites.net",
//                                "https://aeclient.azurewebsites.net",
//                                "http://localhost:3000",
//                                "http://localhost:5000")
//                        .allowedMethods("GET", "PUT", "POST", "DELETE",
//                                "HEAD", "OPTIONS");
//            }
//        };
//    }
//
//    public static void main(String[] args) {
//        SpringApplication.run(Application.class, args);
//    }
//}

// For local deployment

@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Compilation completed successfully in 1s 585ms (yesterday 11:08 AM)

Select “Edit Configurations...” in the top right corner
Add a new configuration for Spring Boot with ‘com.discovery.channel.Application’ as the main class



Name: Application ☐ Share ☑ Single instance only

Main class: com.discovery.channel.Application

VM options:

Program arguments:

Working directory:

Environment variables:

Use classpath of module: DiscoveryChannel

JRE: Default (1.8 - SDK of 'DiscoveryChannel' module)

Spring Boot Settings

☐ Enable debug output ☐ Hide Banner ☑ Enable launch optimization ☑ Enable JMX agent

Active Profiles:

Override parameters

Enabled	Name	Value
	No parameters	

Before launch: Build, Activate tool window

Build

☐ Show this page ☑ Activate tool window

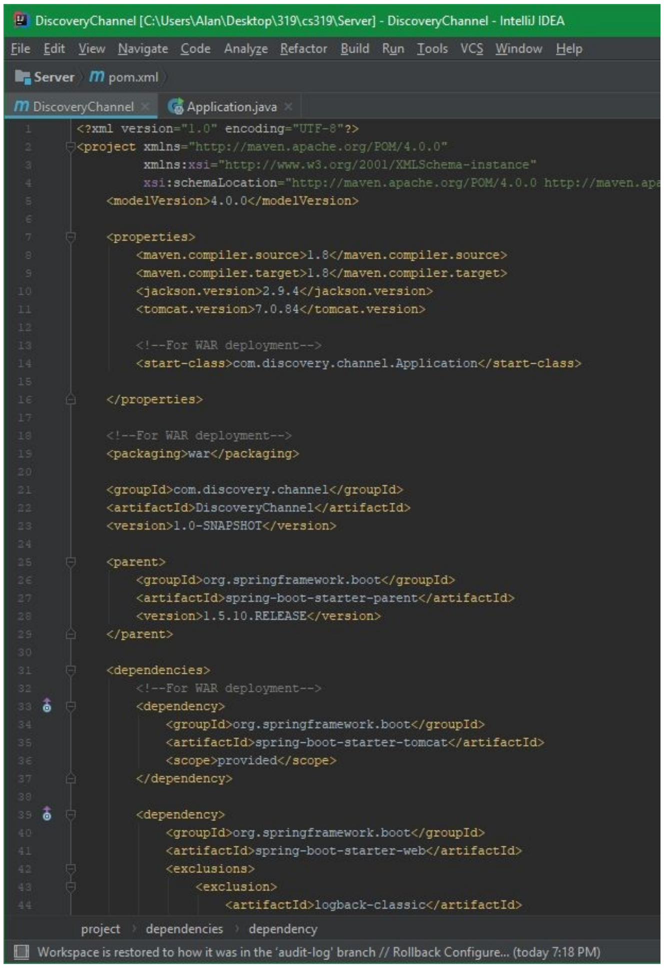
To deploy the server locally, click the green play button.
The server can now be reached at http://localhost:8080/

Production

Server

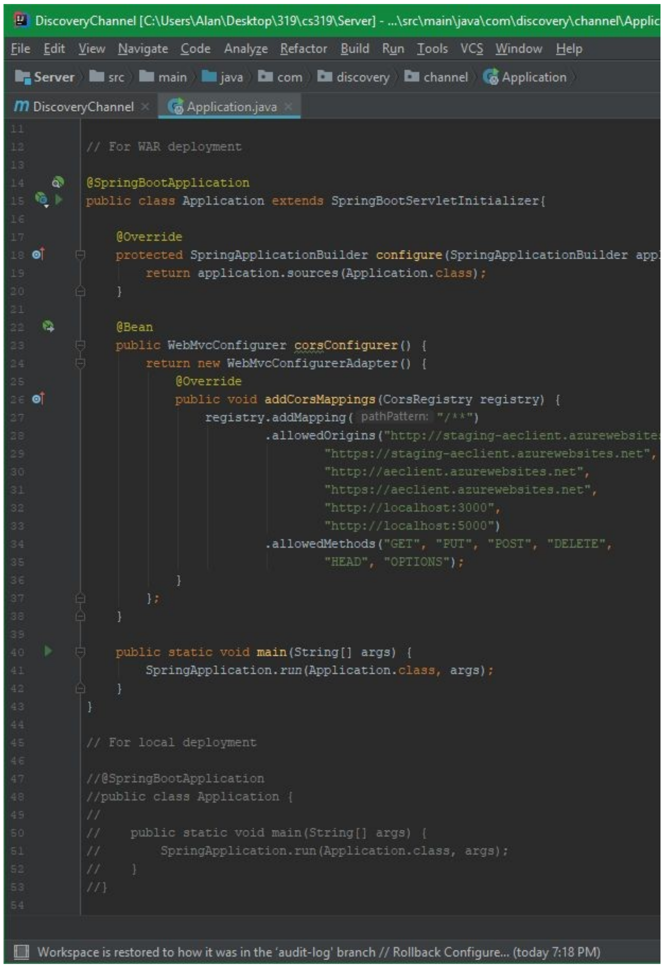
The server is a Spring Boot application running on Tomcat 7.0.84 on an AWS EC2 (Elastic Cloud Compute) t2.micro instance with x86-64 Amazon Linux 2 AMI (Amazon Machine Image).

We have included options for both local deployment and WAR packaging in the source code. To continue with production deployment, ensure that the sections commented “For local deployment” are commented, and the sections commented “For WAR deployment” are not. The relevant sections of code are in pom.xml (3) and Application.java (1)


DiscoveryChannel [C:\Users\Alan\Desktop\319\cs319\Server] - DiscoveryChannel - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Server
pom.xml
DiscoveryChannel
Application.java
project
dependencies
dependency

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apa
    <modelVersion>4.0.0</modelVersion>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
        <jackson.version>2.9.4</jackson.version>
        <tomcat.version>7.0.84</tomcat.version>

        <!--For WAR deployment-->
        <start-class>com.discovery.channel.Application</start-class>

    </properties>

    <!--For WAR deployment-->
    <packaging>war</packaging>

    <groupId>com.discovery.channel</groupId>
    <artifactId>DiscoveryChannel</artifactId>
    <version>1.0-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.5.10.RELEASE</version>
    </parent>

    <dependencies>
        <!--For WAR deployment-->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
            <scope>provided</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <exclusions>
                <exclusion>
                    <artifactId>logback-classic</artifactId>
```

Workspace is restored to how it was in the 'audit-log' branch // Rollback Configure... (today 7:18 PM)


DiscoveryChannel [C:\Users\Alan\Desktop\319\cs319\Server] - ...\src\main\java\com\discovery\channel\Applic
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Server
src
main
java
com
discovery
channel
Application
DiscoveryChannel
Application.java

```java

// For WAR deployment

@SpringBootApplication
public class Application extends SpringBootServletInitializer{

    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder appl
        return application.sources(Application.class);
    }

    @Bean
    public WebMvcConfigurer corsConfigurer() {
        return new WebMvcConfigurerAdapter() {
            @Override
            public void addCorsMappings(CorsRegistry registry) {
                registry.addMapping( pathPattern: "/**")
                        .allowedOrigins("http://staging-aeclient.azurewebsite
                                "https://staging-aeclient.azurewebsites.net",
                                "http://aeclient.azurewebsites.net",
                                "https://aeclient.azurewebsites.net",
                                "http://localhost:3000",
                                "http://localhost:5000")
                        .allowedMethods("GET", "PUT", "POST", "DELETE",
                                "HEAD", "OPTIONS");

            }
        };
    }

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}

// For local deployment

//@SpringBootApplication
//public class Application {
//
//    public static void main(String[] args) {
//        SpringApplication.run(Application.class, args);
//    }
//}

```

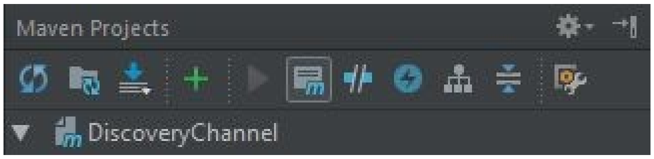
Workspace is restored to how it was in the 'audit-log' branch // Rollback Configure... (today 7:18 PM)

.

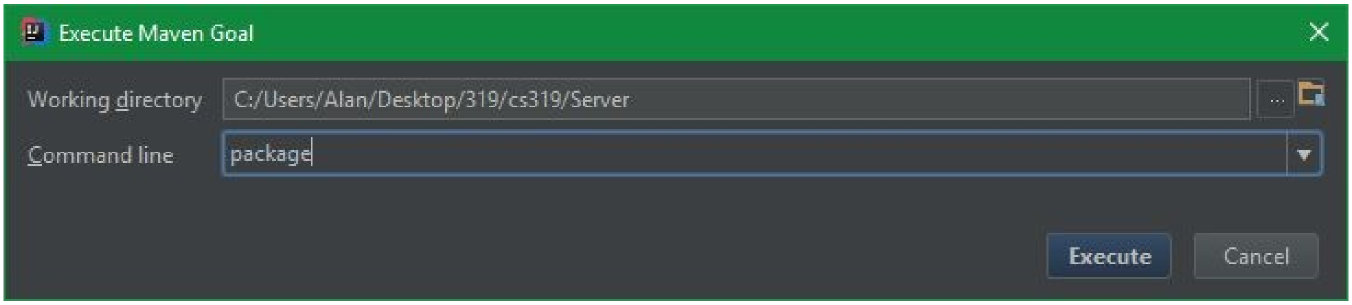
In Application.java, you can also set the allowed origins and methods for CORS configurations.

Select View > Tool Windows > Maven Projects

From the Maven Projects window select the Execute Maven Goal button



Type 'package' and execute. A .WAR file will be generated in Server\target\



Working directory C:/Users/Alan/Desktop/319/cs319/Server

Command line package

Now that we have a .WAR file, we will work on deploying it to a Tomcat server on an AWS EC2 instance.

From the AWS console, select EC2 and then Launch Instance to begin.
For our setup, we used the 64-bit Amazon Linux 2 AMI. Follow the prompts to configure your EC2 instance. Most settings were left at (free tier) defaults.
From the sidebar, select Elastic IPs then Allocate new address.
Once a new address has been allocated, associate it with your EC2 instance by selecting the address and then Associate address from the Actions button.
Next, you will need to create a key pair from the Network & Security section in the sidebar, it will automatically download a .pem file upon creation.

For connecting to our EC2 instance, we used XShell 5.
Create a new session, and fill in the host name of your EC2 instance, and port 22.
Under Authentication, fill in ec2-user as the User Name and select Public Key as Method.
Next to User Key select Browse… and select the .pem file created earlier.
You should now be able to SSH to your EC2 instance.

More information on setting up EC2 on:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-connect-to-instance-linux

Steps:
Create AWS EC2 instance DONE
SSH to your instance DONE
Install Java, and Tomcat
Configure Tomcat users
Package WAR file DONE
Deploy WAR file

-- SSL steps --

Acceptance Criteria

See the business requirements document

User Guide

Login

To login, append a query string “userId” to the end of the service root.

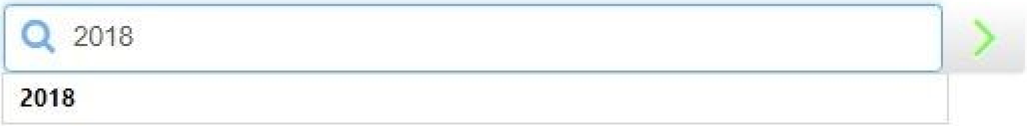
The service roots available as deployed on Azure are:
https://aeclient.azurewebsites.net/
https://staging-aeclient.azurewebsites.net/

Local service root:
http://localhost:3000/

Appending a “userId”:
http://localhost:3000/?userId=500
https://aeclient.azurewebsites.net/?userId=500
https://staging-aeclient.azurewebsites.net/?userId=500

Search

On the landing page, there’ll be a search bar. You can search for records and containers by their numbers.

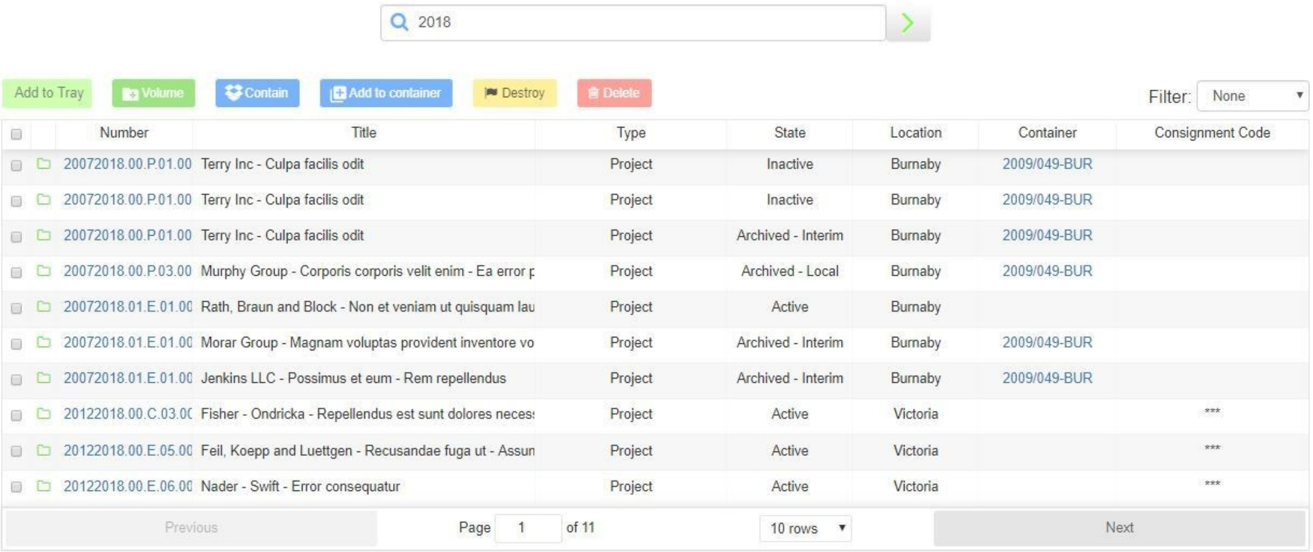


2018

2018

Results

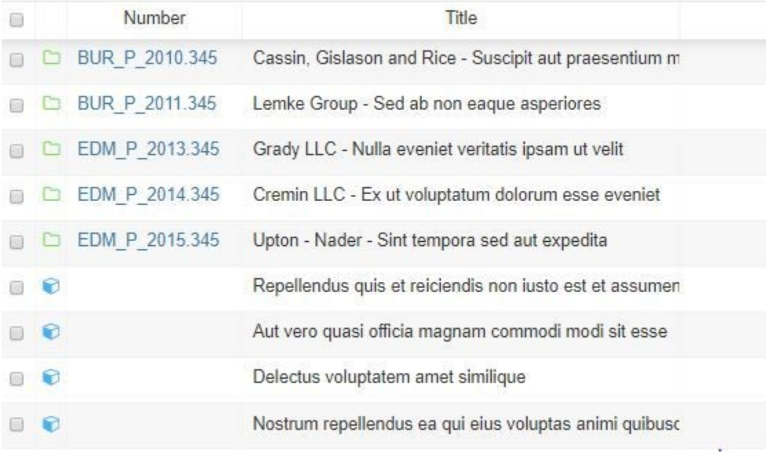
This is the results page with a table containing the results for your search. You can also make a different search. Clicking on the column headers at the top of the table allows you to sort the page result while dragging on the divider between the column headers allow you to change the size of the column.



□		Number	Title	Type	State	Location	Container	Consignment Code
□	📁	20072018.00.P.01.00	Terry Inc - Culpa facilis odit	Project	Inactive	Burnaby	2009/049-BUR	
□	📁	20072018.00.P.01.00	Terry Inc - Culpa facilis odit	Project	Inactive	Burnaby	2009/049-BUR	
□	📁	20072018.00.P.01.00	Terry Inc - Culpa facilis odit	Project	Archived - Interim	Burnaby	2009/049-BUR	
□	📁	20072018.00.P.03.00	Murphy Group - Corporis corporis velit enim - Ea error p	Project	Archived - Local	Burnaby	2009/049-BUR	
□	📁	20072018.01.E.01.00	Rath, Braun and Block - Non et veniam ut quisquam lau	Project	Active	Burnaby		
□	📁	20072018.01.E.01.00	Morar Group - Magnam voluptas provident inventore vo	Project	Archived - Interim	Burnaby	2009/049-BUR	
□	📁	20072018.01.E.01.00	Jenkins LLC - Possimus et eum - Rem repellendus	Project	Archived - Interim	Burnaby	2009/049-BUR	
□	📁	20122018.00.C.03.0(	Fisher - Ondricka - Repellendus est sunt dolores neces:	Project	Active	Victoria		***
□	📁	20122018.00.E.05.00	Feil, Koepp and Luettgen - Recusandae fuga ut - Assun	Project	Active	Victoria		***
□	📁	20122018.00.E.06.00	Nader - Swift - Error consequatur	Project	Active	Victoria		***

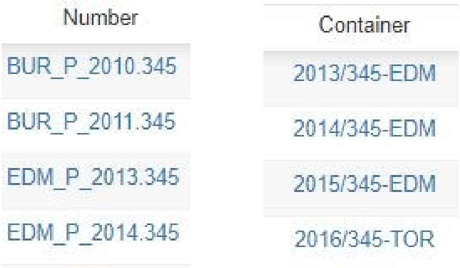
On the bottom of the table, you can change the page, jump to a specific page, or change the page size to show more items.

Between the checkboxes and Number column are green and blue icons. Green icon shows that it's a record, while blue icon is a container.



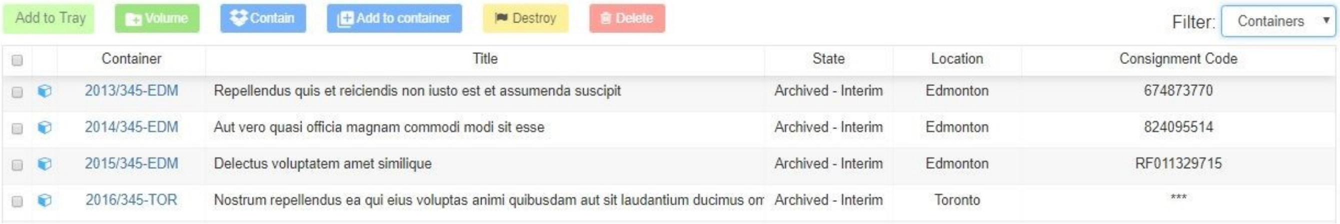
□		Number	Title
□	📁	BUR_P_2010.345	Cassin, Gislason and Rice - Suscipit aut praesentium m
□	📁	BUR_P_2011.345	Lemke Group - Sed ab non eaque asperiores
□	📁	EDM_P_2013.345	Grady LLC - Nulla eveniet veritatis ipsam ut velit
□	📁	EDM_P_2014.345	Cremin LLC - Ex ut voluptatum dolorum esse eveniet
□	📁	EDM_P_2015.345	Upton - Nader - Sint tempora sed aut expedita
□	📦		Repellendus quis et reiciendis non iusto est et assumen
□	📦		Aut vero quasi officia magnam commodi modi sit esse
□	📦		Delectus voluptatem amet similique
□	📦		Nostrum repellendus ea qui eius voluptas animi quibusc

Clicking on the blue links in the Number and Container columns will take you to view the details of the record or container.



Number	Container
BUR_P_2010.345	2013/345-EDM
BUR_P_2011.345	2014/345-EDM
EDM_P_2013.345	2015/345-EDM
EDM_P_2014.345	2016/345-TOR

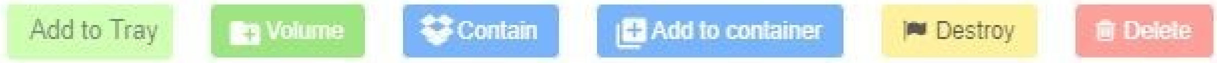
On the top right, you can filter the results to display only the Containers or Records.

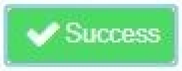


Filter: Containers

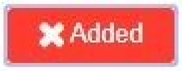
□		Container	Title	State	Location	Consignment Code
□		2013/345-EDM	Repellendus quis et reiciendis non iusto est et assumenda suscipit	Archived - Interim	Edmonton	674873770
□		2014/345-EDM	Aut vero quasi officia magnam commodi modi sit esse	Archived - Interim	Edmonton	824095514
□		2015/345-EDM	Delectus voluptatem amet similique	Archived - Interim	Edmonton	RF011329715
□		2016/345-TOR	Nostrum repellendus ea qui eius voluptas animi quibusdam aut sit laudantium ducimus om	Archived - Interim	Toronto	***

These buttons are disabled and will only be available when you select certain items by checking the checkboxes on the left side of the table. ‘Volume’ requires selecting record(s) with the same number before their volume number. ‘Add to Container’ requires selecting at most one container and some record(s).





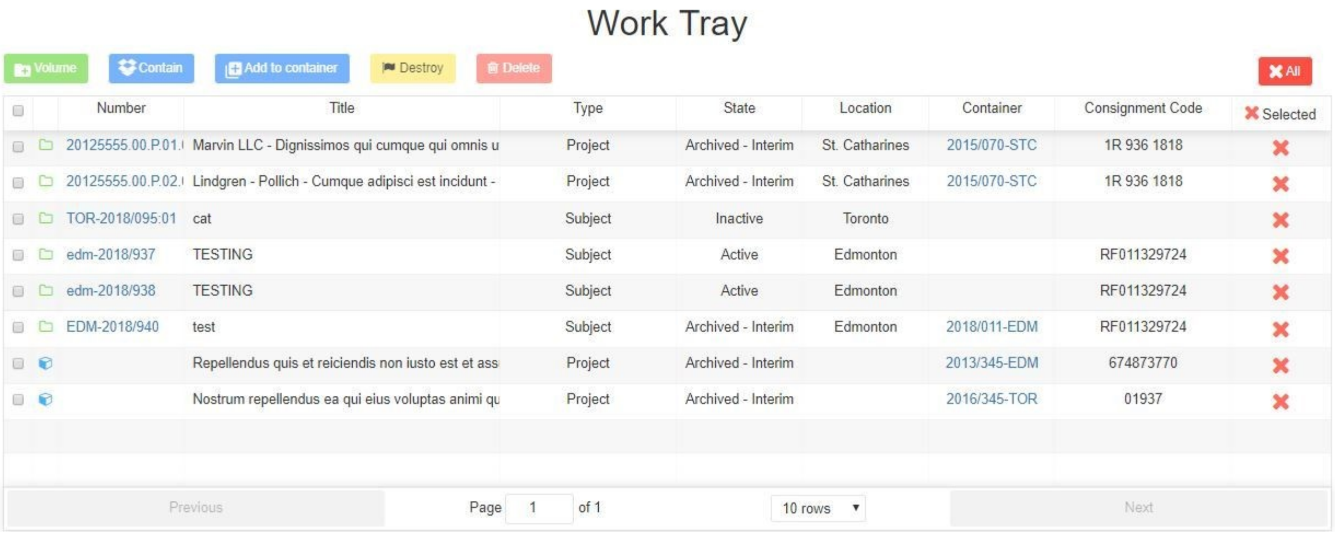
Add to Tray is successful, one or more selected item(s) are added into your work tray.



Add to Tray failed, all selected item(s) are already added in your work tray.

Work Tray

You can access the work tray page in the top banner. The work tray page is similar but allows you to action on items from different searches or simply to store them for later. It also updates the items on load, so if something is already deleted, it'll be removed from your tray.



Work Tray

□		Number	Title	Type	State	Location	Container	Consignment Code	✖ Selected
□	🗀	20125555.00.P.01.	Marvin LLC - Dignissimos qui cumque qui omnis u	Project	Archived - Interim	St. Catharines	2015/070-STC	1R 936 1818	✖
□	🗀	20125555.00.P.02.	Lindgren - Pollich - Cumque adipisci est incidunt -	Project	Archived - Interim	St. Catharines	2015/070-STC	1R 936 1818	✖
□	🗀	TOR-2018/095:01	cat	Subject	Inactive	Toronto			✖
□	🗀	edm-2018/937	TESTING	Subject	Active	Edmonton		RF011329724	✖
□	🗀	edm-2018/938	TESTING	Subject	Active	Edmonton		RF011329724	✖
□	🗀	EDM-2018/940	test	Subject	Archived - Interim	Edmonton	2018/011-EDM	RF011329724	✖
□	📦		Repellendus quis et reiciendis non iusto est et ass	Project	Archived - Interim		2013/345-EDM	674873770	✖
□	📦		Nostrum repellendus ea qui eius voluptas animi qu	Project	Archived - Interim		2016/345-TOR	01937	✖

✖ All

This allows you to quickly empty your tray.

✖ Selected

This removes only the selected items.

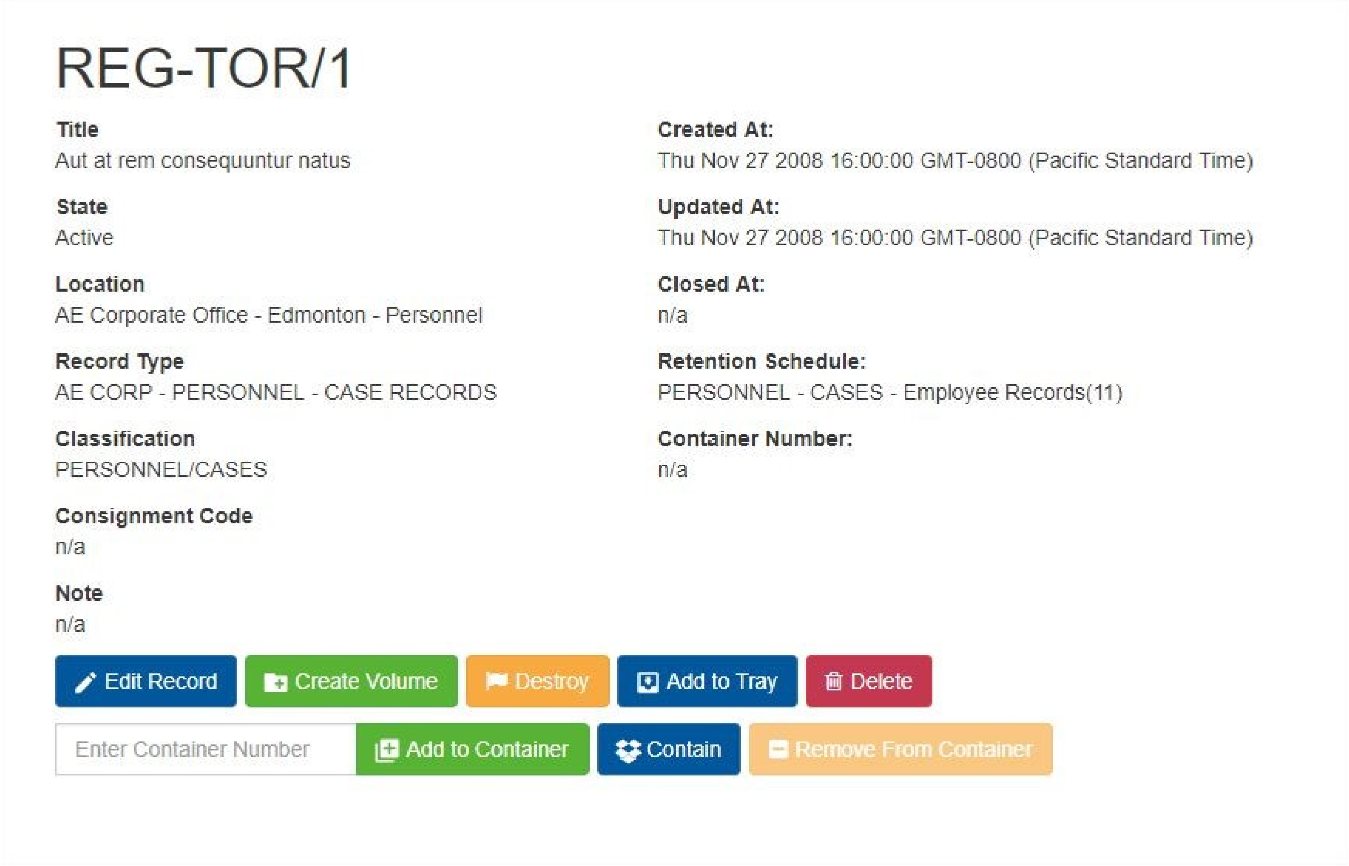
✖

✖

These buttons remove only the individual rows.

Viewing a record

To view a record, click on the blue links in the tables or type its id in the url like so /viewRecord/188133
You will be taken to a page with its details.



REG-TOR/1

Title
Aut at rem consequuntur natus

State
Active

Location
AE Corporate Office - Edmonton - Personnel

Record Type
AE CORP - PERSONNEL - CASE RECORDS

Classification
PERSONNEL/CASES

Consignment Code
n/a

Note
n/a

Created At:
Thu Nov 27 2008 16:00:00 GMT-0800 (Pacific Standard Time)

Updated At:
Thu Nov 27 2008 16:00:00 GMT-0800 (Pacific Standard Time)

Closed At:
n/a

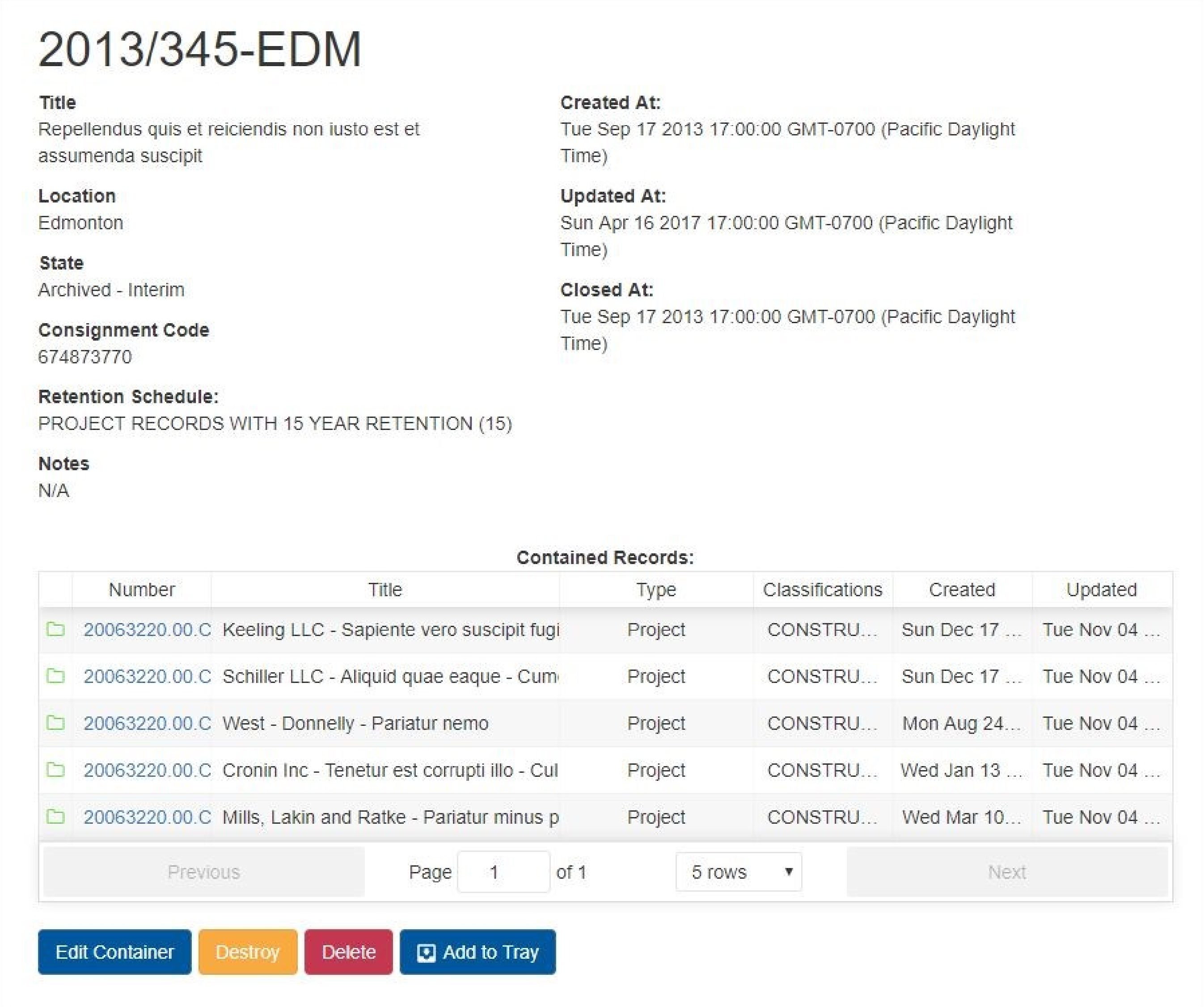
Retention Schedule:
PERSONNEL - CASES - Employee Records(11)

Container Number:
n/a

You can enter an existing container number to add the record to the container.

Viewing a container

Similar to viewing a record, clicking on a container (or typing its id) will direct you to its detail page. It will also show a table with the records it contains.



2013/345-EDM

Title
Repellendus quis et reiciendis non iusto est et assumenda suscipit

Location
Edmonton

State
Archived - Interim

Consignment Code
674873770

Retention Schedule:
PROJECT RECORDS WITH 15 YEAR RETENTION (15)

Notes
N/A

Created At:
Tue Sep 17 2013 17:00:00 GMT-0700 (Pacific Daylight Time)

Updated At:
Sun Apr 16 2017 17:00:00 GMT-0700 (Pacific Daylight Time)

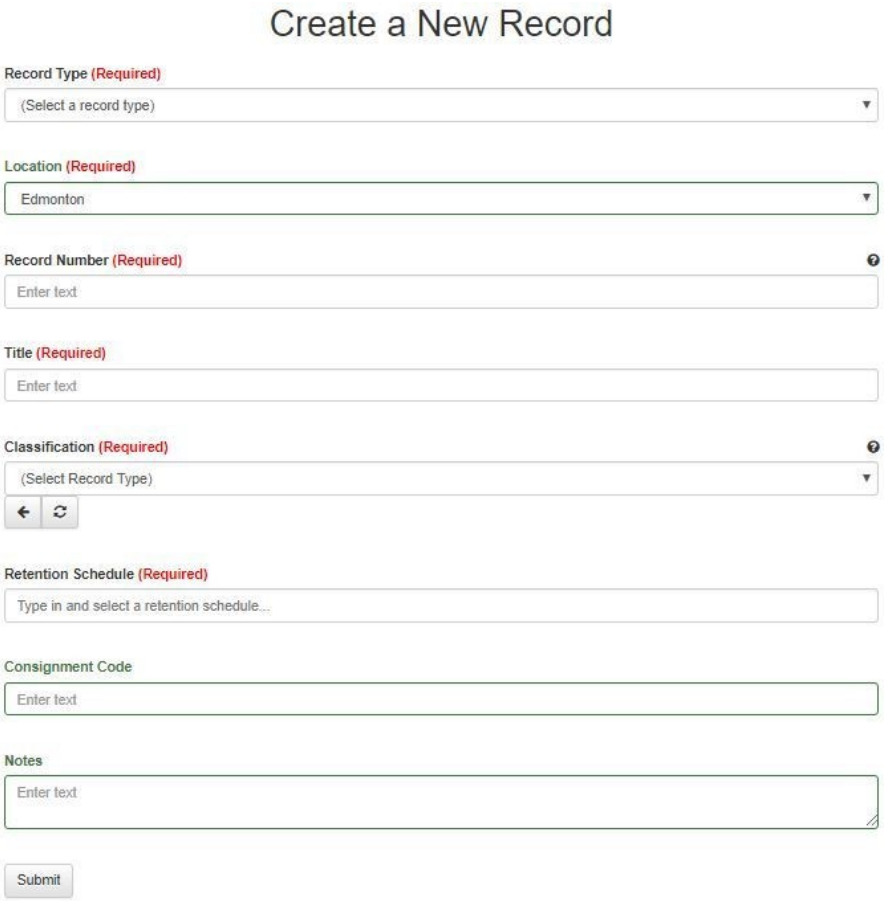
Closed At:
Tue Sep 17 2013 17:00:00 GMT-0700 (Pacific Daylight Time)

Contained Records:

	Number	Title	Type	Classifications	Created	Updated
🗀	20063220.00.C	Keeling LLC - Sapiente vero suscipit fugi	Project	CONSTRU...	Sun Dec 17 ...	Tue Nov 04 ...
🗀	20063220.00.C	Schiller LLC - Aliquid quae eaque - Cum	Project	CONSTRU...	Sun Dec 17 ...	Tue Nov 04 ...
🗀	20063220.00.C	West - Donnelly - Pariatur nemo	Project	CONSTRU...	Mon Aug 24...	Tue Nov 04 ...
🗀	20063220.00.C	Cronin Inc - Tenetur est corrupti illo - Cul	Project	CONSTRU...	Wed Jan 13 ...	Tue Nov 04 ...
🗀	20063220.00.C	Mills, Lakin and Ratke - Pariatur minus p	Project	CONSTRU...	Wed Mar 10...	Tue Nov 04 ...

New Record

Creating a new record will direct you to fill out a form with required fields.



Create a New Record

Record Type (Required)
(Select a record type)

Location (Required)
Edmonton

Record Number (Required)
Enter text

Title (Required)
Enter text

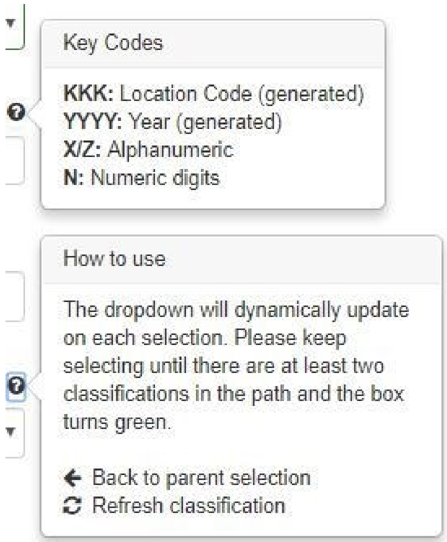
Classification (Required)
(Select Record Type)

Retention Schedule (Required)
Type in and select a retention schedule...

Consignment Code
Enter text

Notes
Enter text

On the right side are little icons you can click for some helpful tooltips.



Key Codes

KKK: Location Code (generated)
YYYY: Year (generated)
X/Z: Alphanumeric
N: Numeric digits

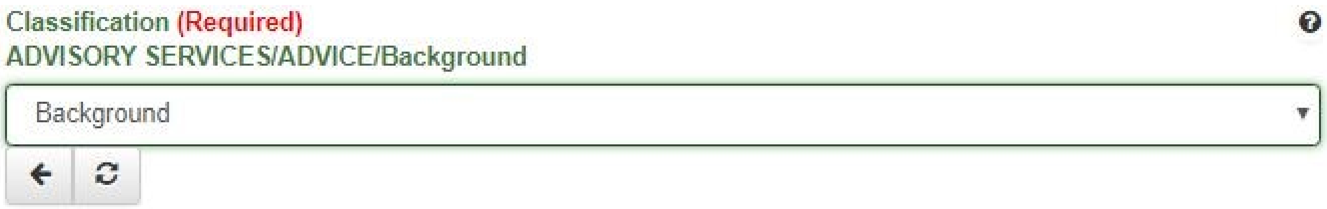
How to use

The dropdown will dynamically update on each selection. Please keep selecting until there are at least two classifications in the path and the box turns green.

← Back to parent selection
⟳ Refresh classification

The classification is a tree structure that may require you to select multiple times before selection is complete.

For example, selecting ADVISORY SERVICES, then ADVICE, and finally Background.



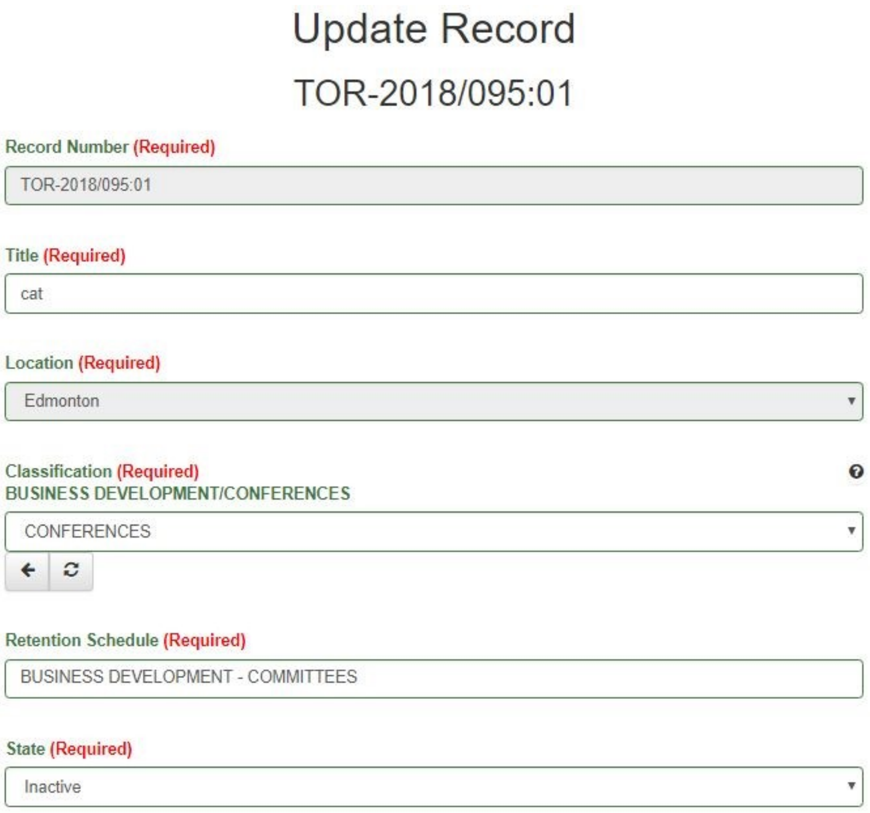
Classification (Required)
ADVISORY SERVICES/ADVICE/Background

Background ▾

You can go back to previous selection or reset at any time.

Updating

Editing a record or container is similar to creating a new record, except it'll pre-populate some of the fields with its original information.



Update Record

TOR-2018/095:01

Record Number (Required)
TOR-2018/095:01

Title (Required)
cat

Location (Required)
Edmonton ▾

Classification (Required)
BUSINESS DEVELOPMENT/CONFERENCES

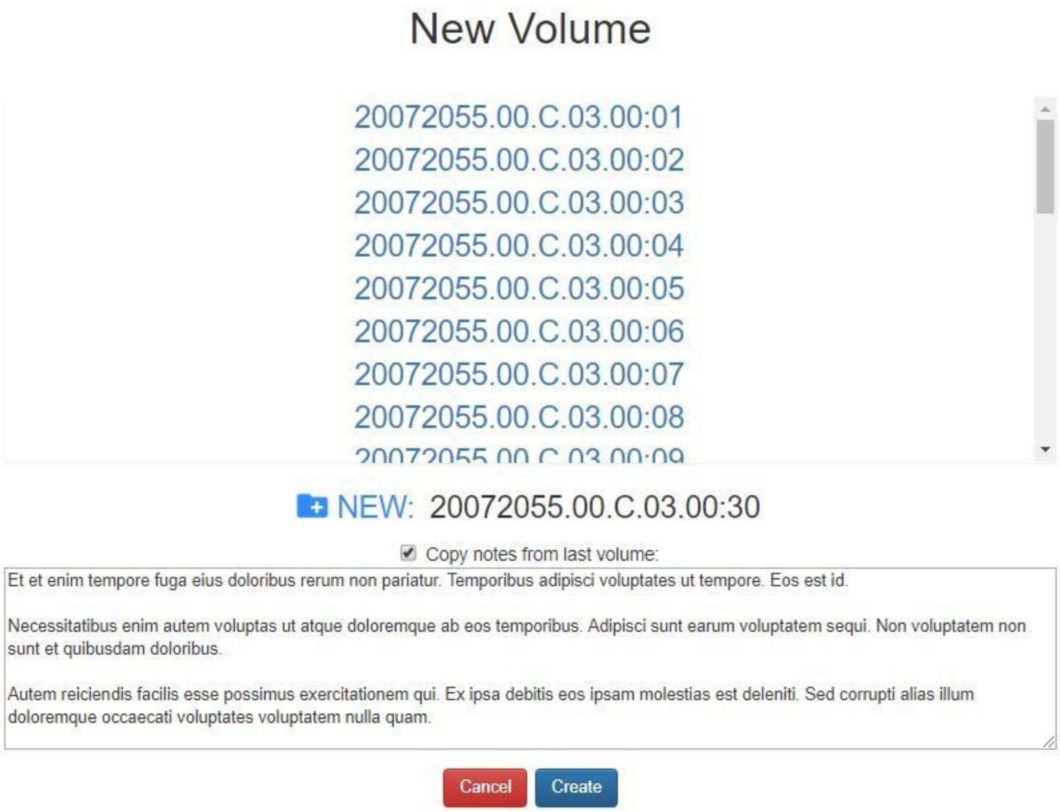
CONFERENCES ▾

Retention Schedule (Required)
BUSINESS DEVELOPMENT - COMMITTEES

State (Required)
Inactive ▾

New Volume

After clicking on the volume button, the volume page will display a list of all the previous volumes of that record number. You can view each one by clicking on their number. It will create a new volume from the last volume. You can choose to copy the notes over from the last volume. On submit, you'll be redirected to the new record volume.



New Volume

20072055.00.C.03.00:01
20072055.00.C.03.00:02
20072055.00.C.03.00:03
20072055.00.C.03.00:04
20072055.00.C.03.00:05
20072055.00.C.03.00:06
20072055.00.C.03.00:07
20072055.00.C.03.00:08
20072055.00.C.03.00:09

NEW: 20072055.00.C.03.00:30

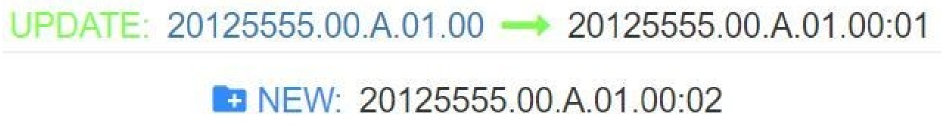
☑ Copy notes from last volume:

Et et enim tempore fuga eius doloribus rerum non pariatur. Temporibus adipisci voluptates ut tempore. Eos est id.

Necessitatibus enim autem voluptas ut atque doloremque ab eos temporibus. Adipisci sunt earum voluptatem sequi. Non voluptatem non sunt et quibusdam doloribus.

Autem reiciendis facilis esse possimus exercitationem qui. Ex ipsa debitis eos ipsam molestias est deleniti. Sed corrupti alias illum doloremque occaecati voluptates voluptatem nulla quam.

If the record wasn't a volume before, it will be updated to one.

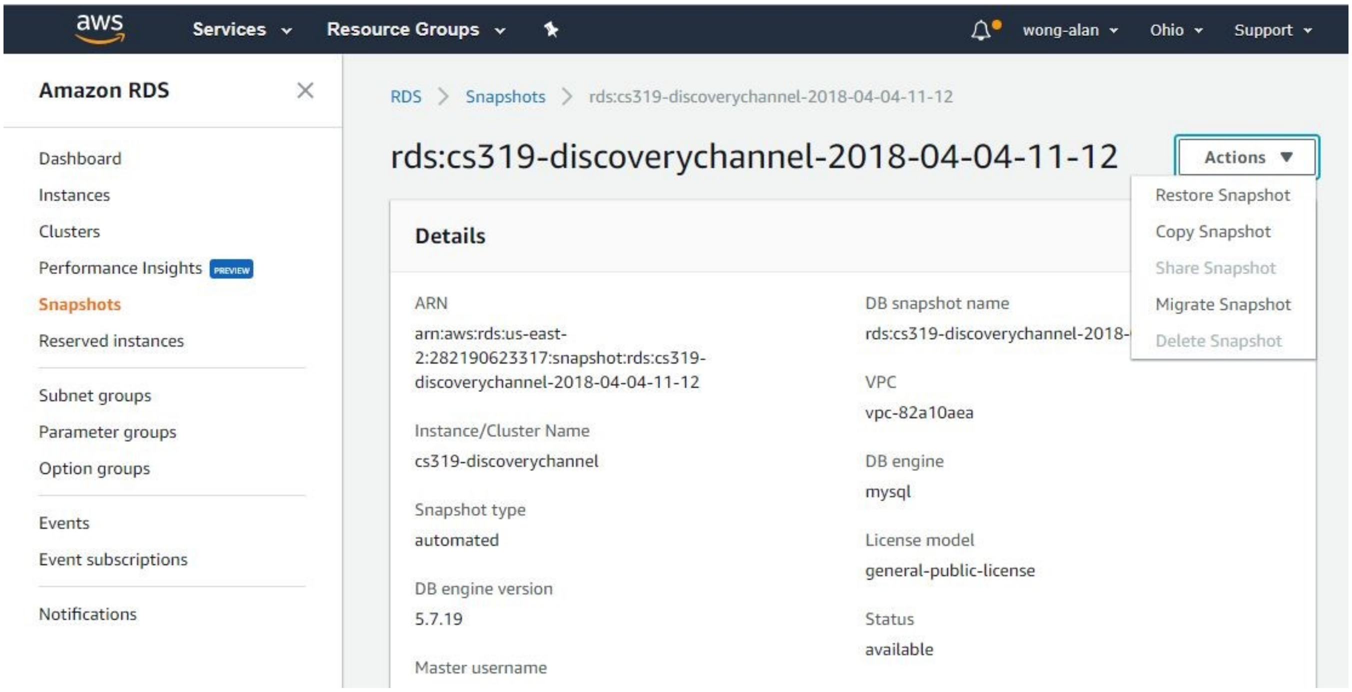


UPDATE: 20125555.00.A.01.00 → 20125555.00.A.01.00:01

NEW: 20125555.00.A.01.00:02

Rollback plan

AWS RDS can be configured to periodically store snapshots of the database that you can then restore to.



rds:cs319-discoverychannel-2018-04-04-11-12

Details

ARN
arn:aws:rds:us-east-2:282190623317:snapshot:rds:cs319-discoverychannel-2018-04-04-11-12

Instance/Cluster Name
cs319-discoverychannel

Snapshot type
automated

DB engine version
5.7.19

Master username

DB snapshot name
rds:cs319-discoverychannel-2018-

VPC
vpc-82a10aea

DB engine
mysql

License model
general-public-license

Status
available

As an alternative, we have provided, within the USB in the folder “DB_Backup”, a MySQL backup of a fresh database. To import the information into MySQL Workbench, use the data export option and select “recordr”.

Electronic repositories

Client project

https://gitlab.com/cpsc3192017w2/ae/Discovery-Channel/Client

Server project

https://gitlab.com/cpsc3192017w2/ae/Discovery-Channel/Server