Table of Contents

A Little Background 3

Deploy the App 4

What’s Going On? 4

Configure Credentials in Eclipse 5

Get Access Key and Secret Key from QwikLab 5

Configure the AWS Toolkit for Eclipse 5

Provide AWS Credentials as JVM Args 6

Achievement Unlocked! 8

Download Source and Create Project 9

Import Application to Eclipse 9

Import EB Environment Into Eclipse 11

Open the Elastic Beanstalk Console 14

OK, Now What? 15

Import App Config for Local Dev 16

Modify and Deploy Application 18

Roll Back to Initial Version 21

Project Anatomy 22

Coding a Java Web Application

A Little Background

# A Little Background

Our sample application is a scalable Java web app that allows users to upload, convert, and share videos via a web browser. The app uses several AWS services that allow it to grow to very large scale: Amazon S3 for uploaded videos and app logs; Amazon RDS for storing searchable video metadata; Amazon DynamoDB for storing user profile information; and Amazon ElastiCache to cache database query results and improve application performance. We develop the application locally in Eclipse and deploy it to AWS Elastic Beanstalk, which gives us Tomcat 7 instances that scale automatically behind an Elastic Load Balancer.

Deploy the App

# Deploy the App

We’re going to start by launching a fully functional and implemented version of our application. Throughout the workshop we’ll visit individual components of the application and implement them ourselves.

**INTERNAL: Launch the app stack via [this link](https://console.aws.amazon.com/cloudformation/home?region=us-west-2" \l "cstack=sn%7Eamm-reinvent%7Cturl%7Ehttp://amm-us-west-2.s3.amazonaws.com/public/launch-app.json). Provide a value for the EC2KeyName param; leave others as default.**

Open QwikLab in your browser, create an account, sign in, and click ‘Start Lab’. Behind the scenes, QwikLab is using AWS CloudFormation to create your Elastic Beanstalk application ad supporting components (DynamodDB tables, RDS databases, etc). We’ll talk about this is a group in more detail.

## What’s Going On?

CloudFormation is creating an RDS database, DynamoDB table, S3 bucket, and ElastiCache cluster. Once those things are created, CloudFormation will deploy a functional version of the sample application to Elastic Beanstalk and will pass in the IDs of the database, table, bucket, and cluster it created previously.

Configure Credentials In Eclipse

# Configure Credentials in Eclipse

Before we import the source for our sample application and begin coding, let’s configure our IDE with the AWS API credentials (i.e., Access Key and Secret Key) we’ll need to interact with various AWS services.

**INTERNAL: Use AK/SK from your own account.**

## Get Access Key and Secret Key from QwikLab

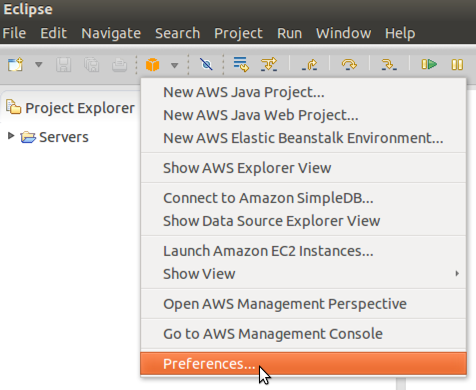
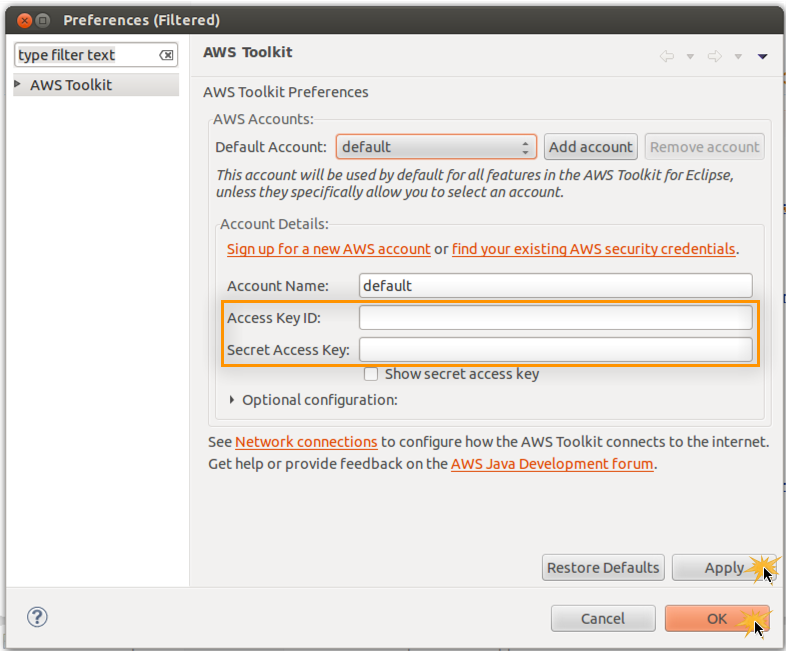
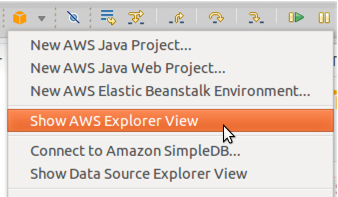
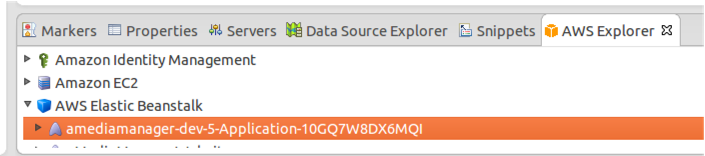
An Access Key (AK) and Secret Key (SK) are used to sign API requests that you make to AWS. QwikLab has helped us out by automatically generating an AK and SK for you. Open the QwikLab tab in your browser and locate them:

**TODO: Provide screenshots of QwikLab UI for getting keys.**

If we weren’t using QwikLab in this workshop, you would manage your keys by using the Identity and Access Management service in the AWS Management Console at <https://console.aws.amazon.com/iam>.

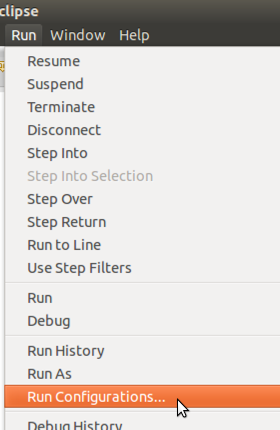
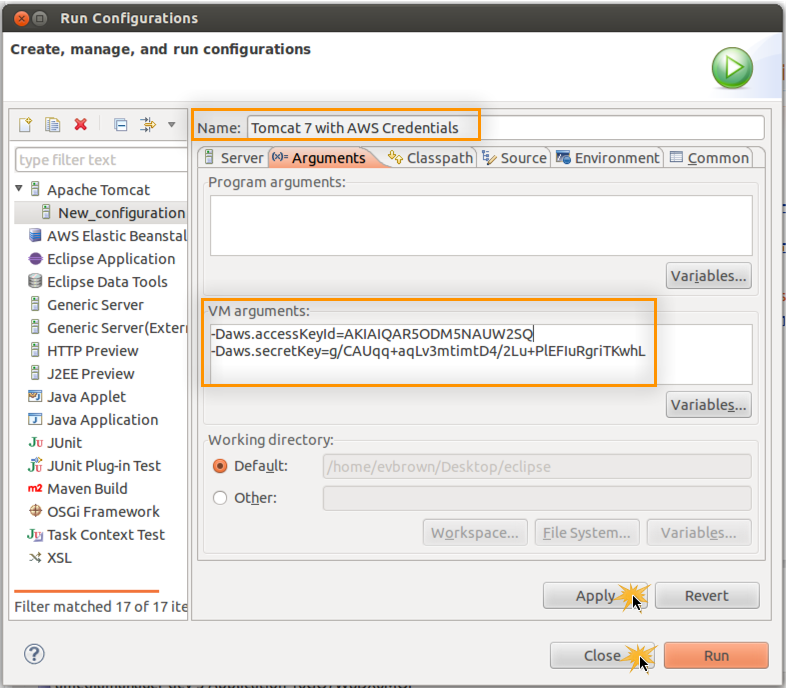
## Configure the AWS Toolkit for Eclipse

The toolkit will help you deploy your application to AWS Elastic Beanstalk and needs an Access Key and Secret Key configured to be able to call the EB APIs.

1. Select the Preferences option in the AWS Toolkit dropdown menu:  
   
2. Enter the Access Key ID and Secret Access Key you retrieved from QwikLab, then click Apply and OK:  
   
3. Confirm that your credentials are correctly configured by opening the AWS Explorer view and listing your AWS Elastic Beanstalk Environments:  
     
     
   

## 

right?

1. **
2. *VM*

## Achievement Unlocked!

Good work. You’ve configured Eclipse to allow the AWS Toolkit to work, and your local Tomcat 7 installation will make your AK and SK available to any apps running on it via System.getProperty() calls (and we don’t have to worry about hard-coding those credentials in code or properties files).

In the next section, you’ll import the application source code into a new project in Eclipse.

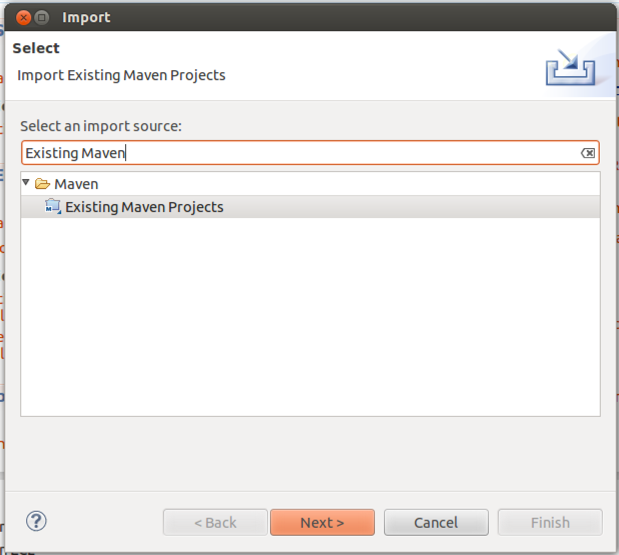
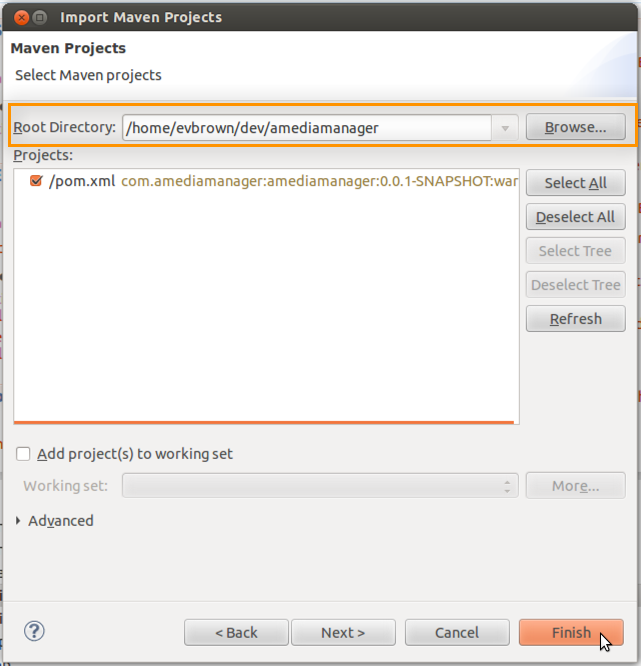
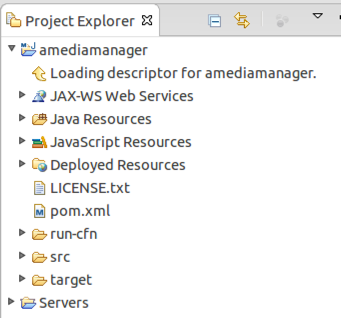
Download Source and Create Project

# Download Source and Create Project

Now that you’ve configured your local IDE, import the application into Eclipse and finally deploy the first version of the code to Elastic Beanstalk.

## Import Application to Eclipse

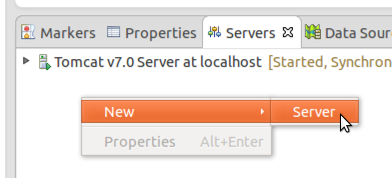
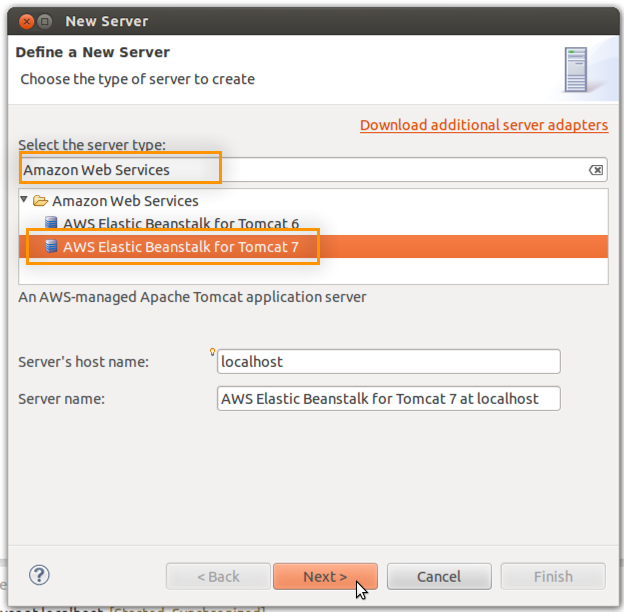
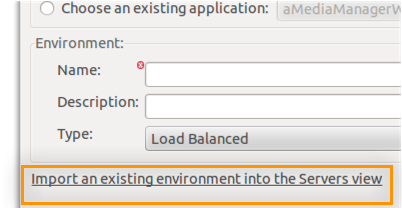
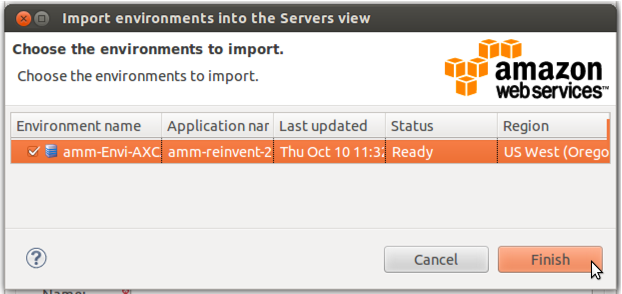
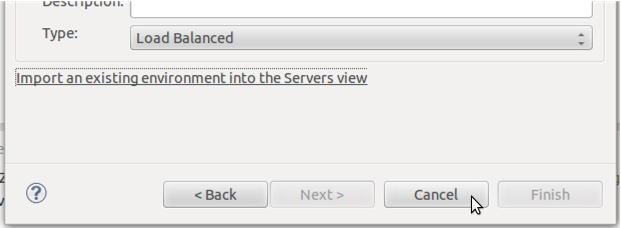
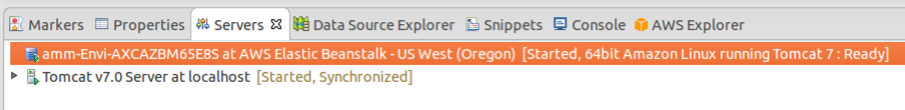
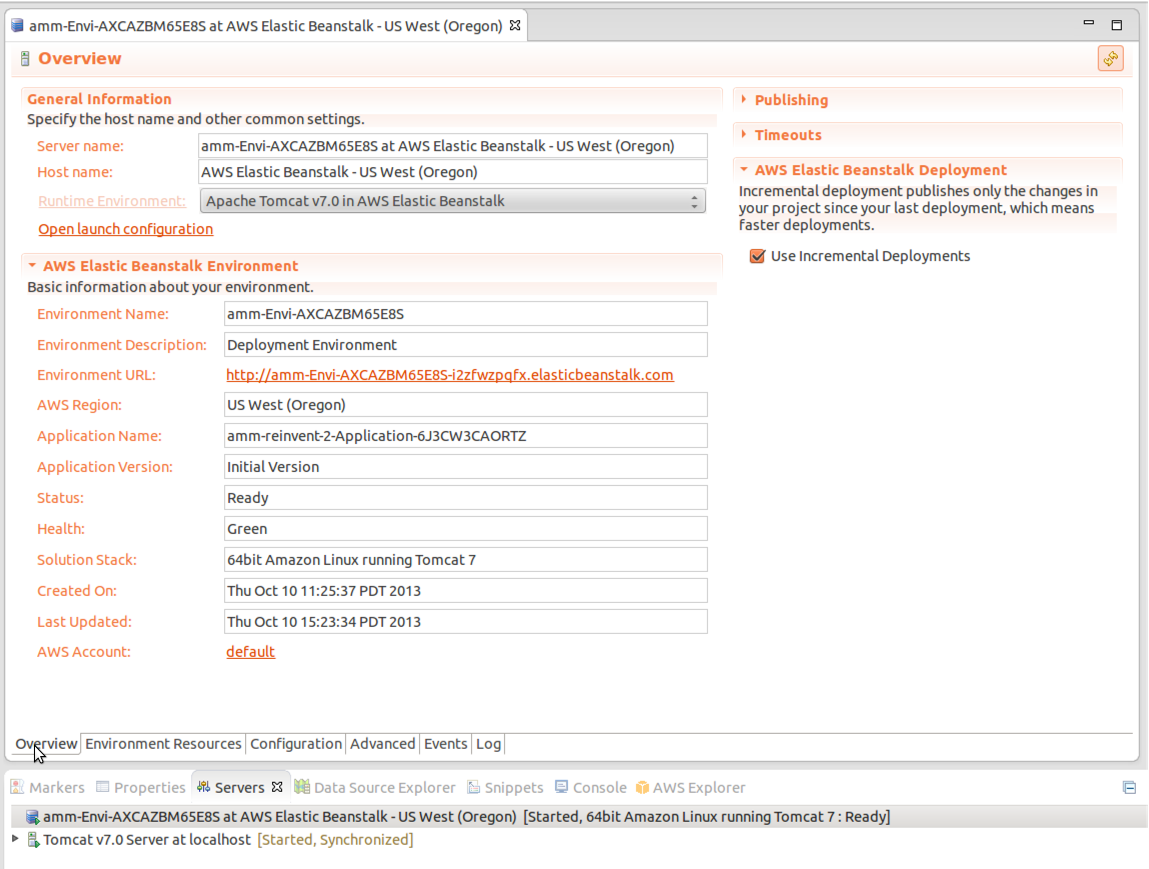
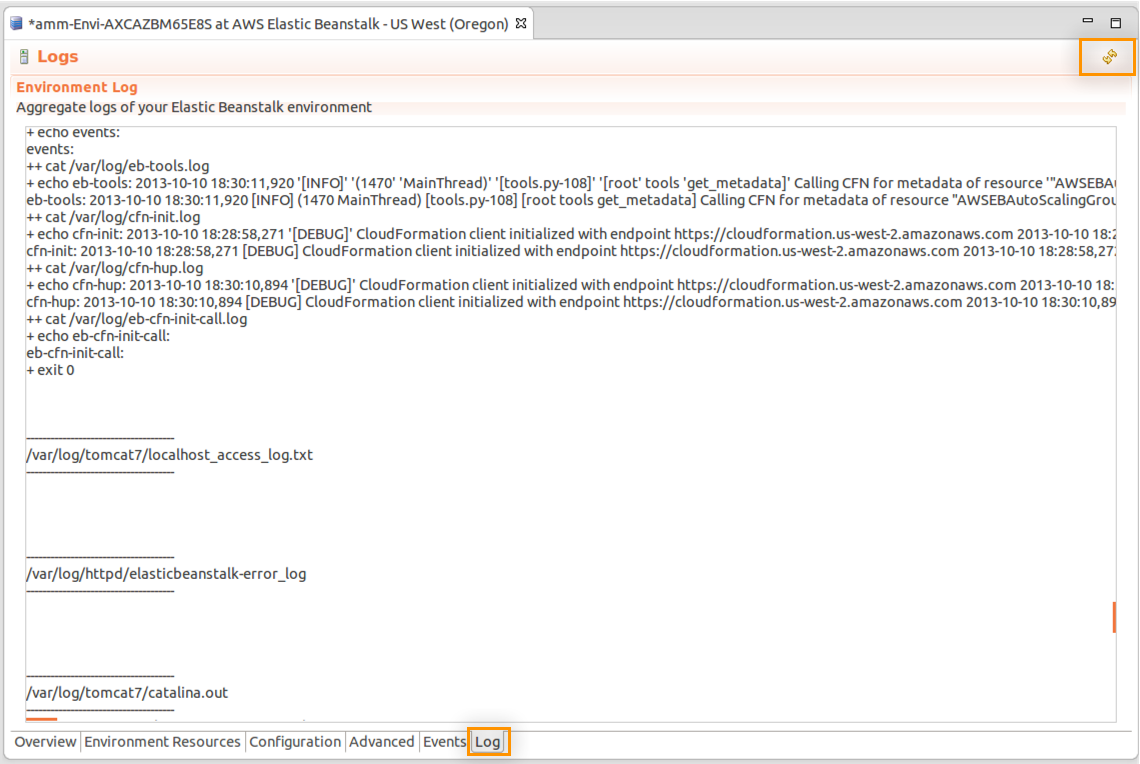
Note: in live workshop, customers will download ZIP from S3 or possibly USB keys.

1. Clone repo from <https://github.com/evandbrown/amediamanager>
2. In Eclipse, choose File > Import, then choose Existing Maven Projects and click Next:  
   
3. Browse to and select the folder you extracted above that contains *pom.xml* and choose Finish:  
   
4. Confirm that the project was added to your Project Explorer pane:

Import EB Environment Into Eclipse

# Import EB Environment Into Eclipse

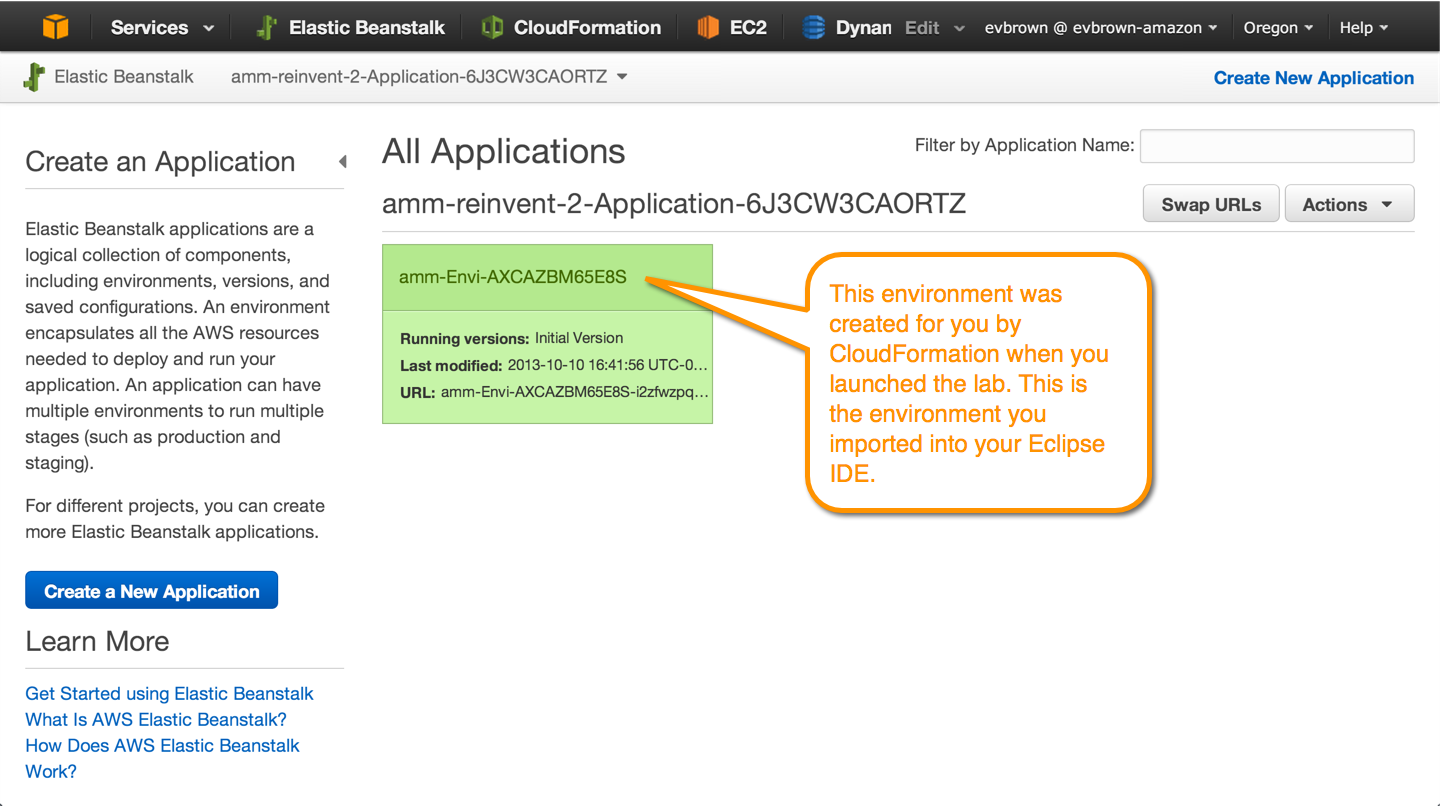
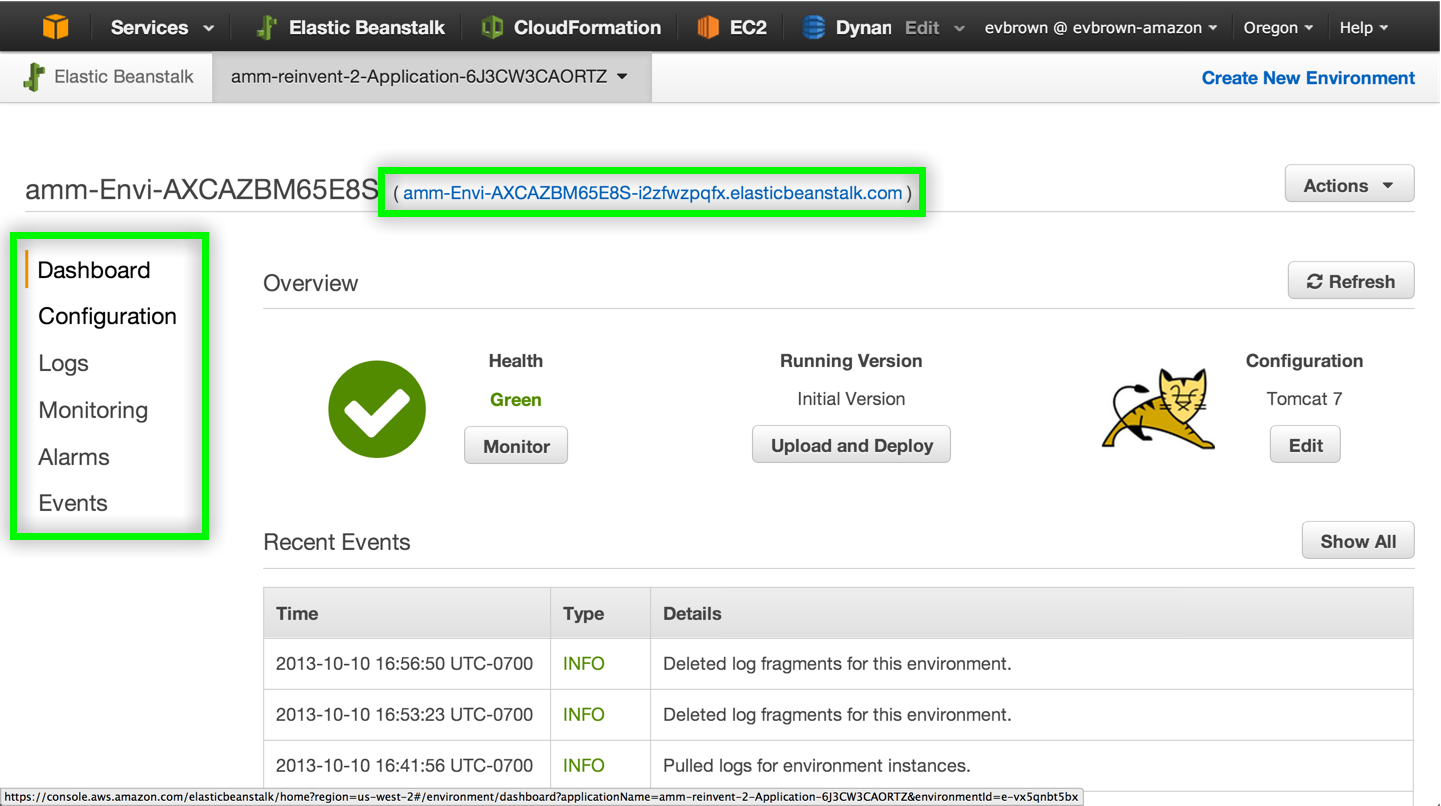
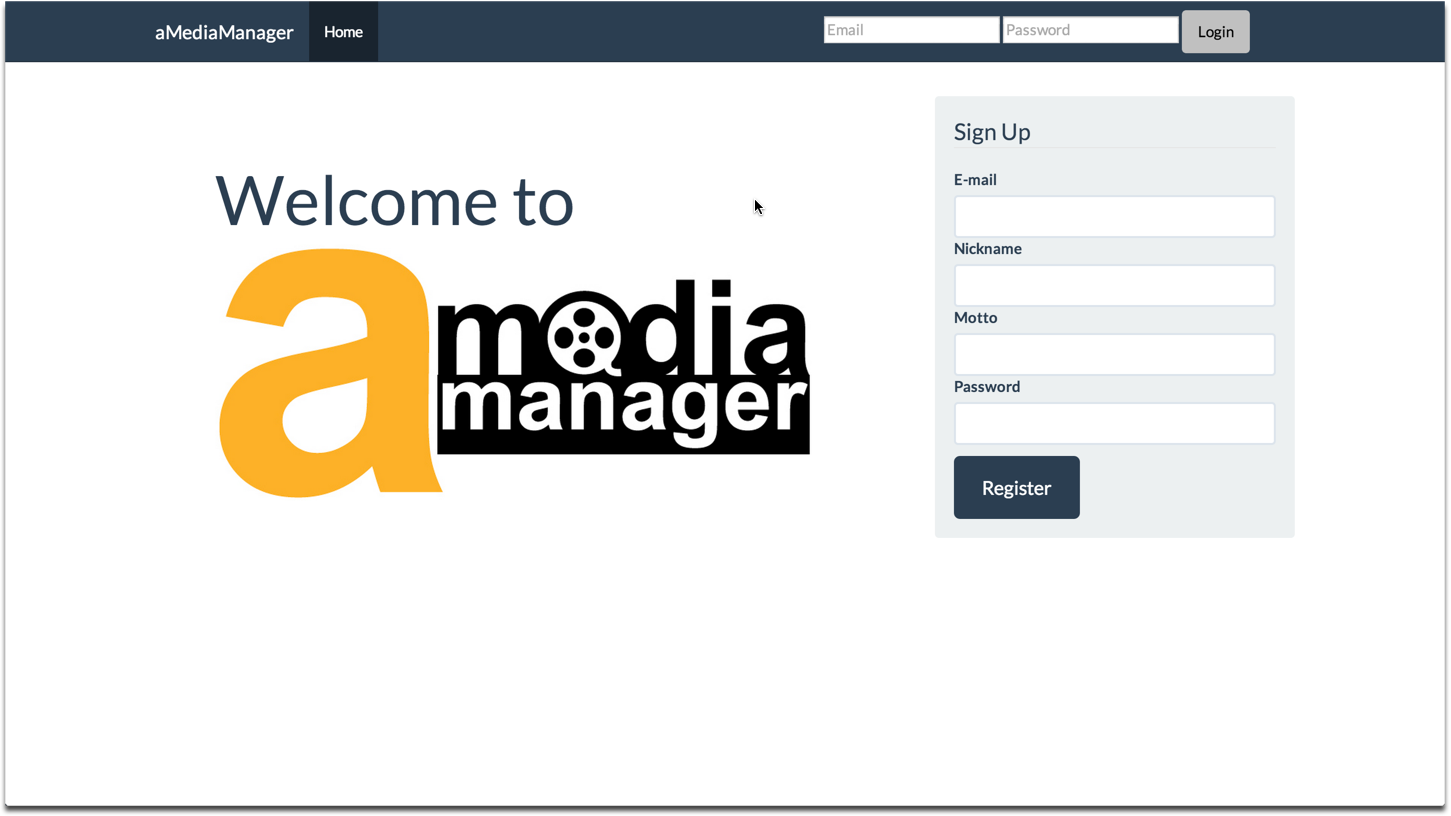
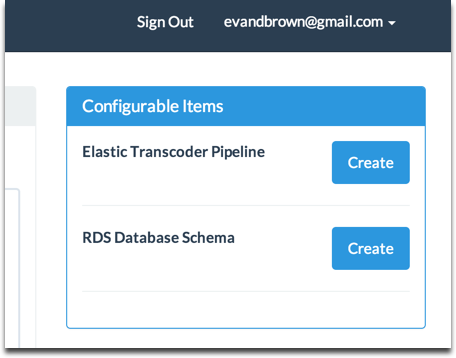
In the previous *Deploy the App* section of this lab, you used CloudFormation to launch your EB application and all of its dependencies (i.e., database, S3 bucket, etc). You should now have an Elastic Beanstalk environment available that is hosting an initial version of the sample application. In the following steps, you will import that environment into your IDE using the AWS Toolkit for Eclipse. This will allow you to deploy your project to Elastic Beanstalk directly from Eclipse.

1. Right-click the Servers view area in Eclipse and choose New > Server:  
   
2. Locate and select the AWS Elastic Beanstalk for Tomcat 7 server type and click Next:  
   
3. Click the link to Import an existing environment into the Servers view:  
   
4. Select the Elastic Beanstalk environment to import (there should be only 1) and click Finish:  
   
5. The environment will be imported and you may click Cancel to exit the New Server wizard:  
   
6. You should now see your Elastic Beanstalk environment as an available server in the Eclipse servers view:  
   
7. Double-click the Elastic Beanstalk server to view its complete configuration within your IDE:  
   
8. Click the Logs tab followed by the refresh button in the top-right of the screen to view a snapshot of the logs from the servers running in your Elastic Beanstalk environment:  
   

Open the Elastic Beanstalk Console

# Open the Elastic Beanstalk Console

In the previous section you viewed the configuration of your Elastic Beanstalk environment using the AWS Toolkit. In this section, use the Elastic Beanstalk Management Console to view and manage your environment from a web browser.

1. Navigate to <https://console.aws.amazon.com/elasticbeanstalk> in a web browser
2. If this is your first time signing into the console, see Appendix Q for instructions on authenticating using QwikLab credentials.
3. The Elastic Beanstalk dashboard shows a view of your applications and environments. Your environment should be in the GREEN state:  
   
4. Click the environment’s name to drill down. The Environment dashboard allows you to view and control the environment’s Configuration, Logs, Monitoring, Alarms, and Events, as well as view the running application via the URL at the top:  
   
5. Click your environment’s URL to view the version of the application that was deployed automatically when you started the lab:  
   
6. Sign up for a new account to access the app.
7. Click the App Config button to view an admin page:  
   
8. Click to create any Configurable Items:  
   

## OK, Now What?

We’re so close to writing code! There’s just one short section to complete before we start developing and shipping. Quickly, let’s recap what you’ve done so far:

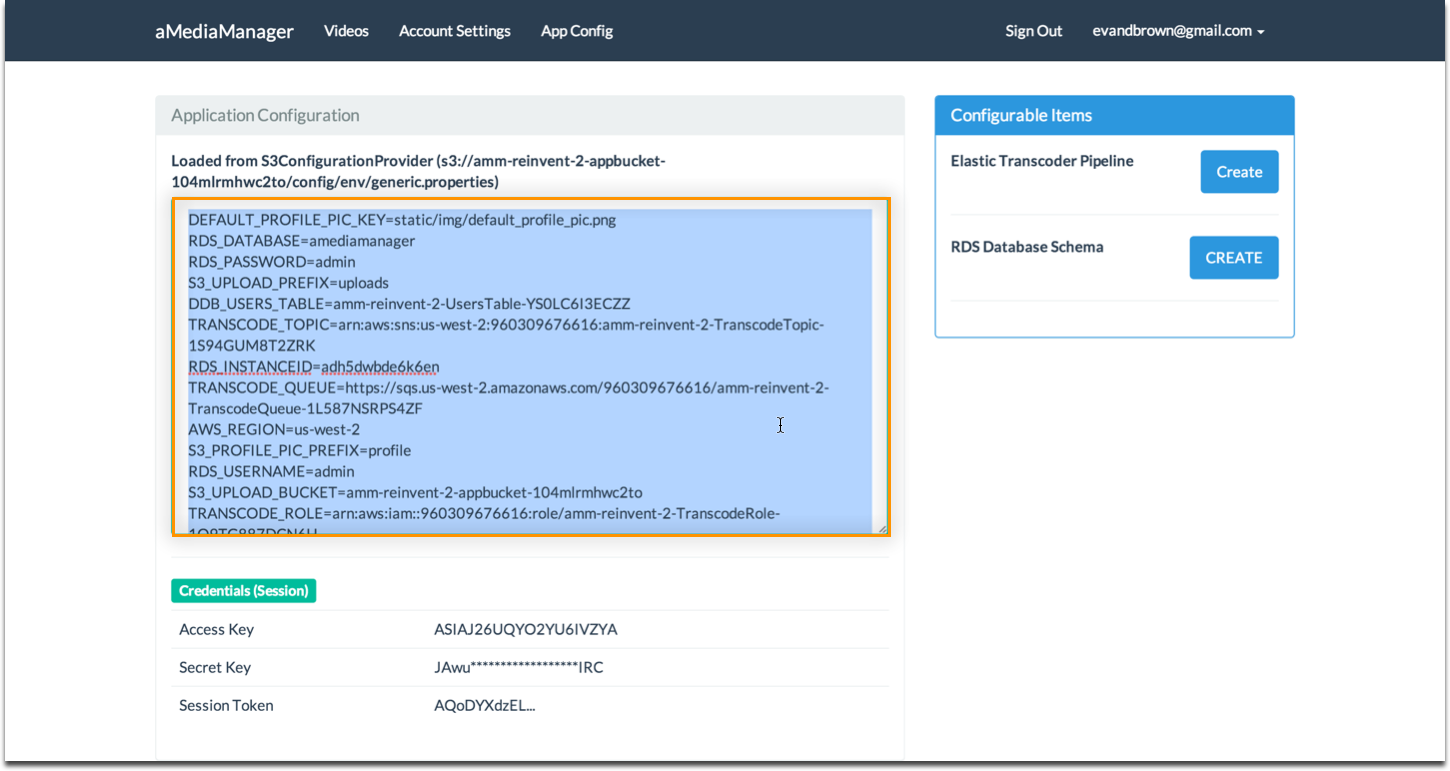
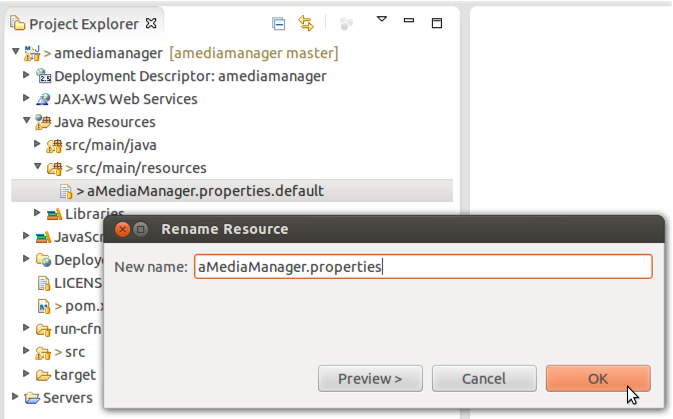
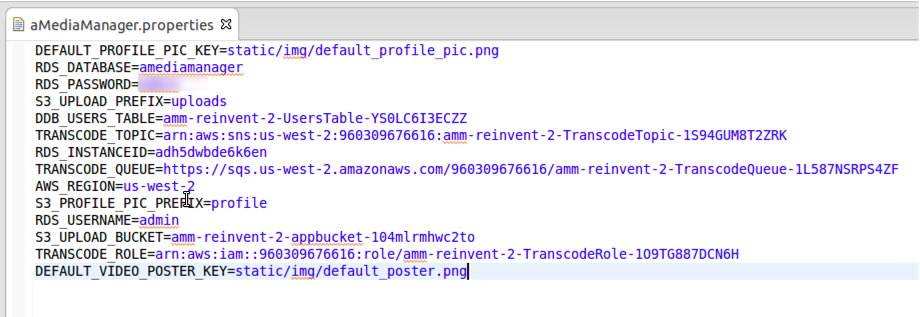
* Used QwikLab to deploy the sample application to Elastic Beanstalk
* Configured Eclipse with your AWS API keys (and remember, you didn’t store them in code or a version-controlled file. Good work!)
* Downloaded and imported the app source into a new project in Eclipse
* Imported the Elastic Beanstalk into Eclipse so you can deploy directly to it from the IDE

Import App Config for Local Dev

# Import App Config for Local Dev

When you started the lab in the very first section, CloudFormation provisioned your Elastic Beanstalk environment and all of your app’s dependencies, things like an RDS database, S3 bucket, DynamoDB table, etc. CloudFormation told Elastic Beanstalk about the names of those resources, and Elastic Beanstalk is making them available to your application via environment variables. The application version running in EB (which you just looked at in a browser in the last section) is good-to-go with its configuration.

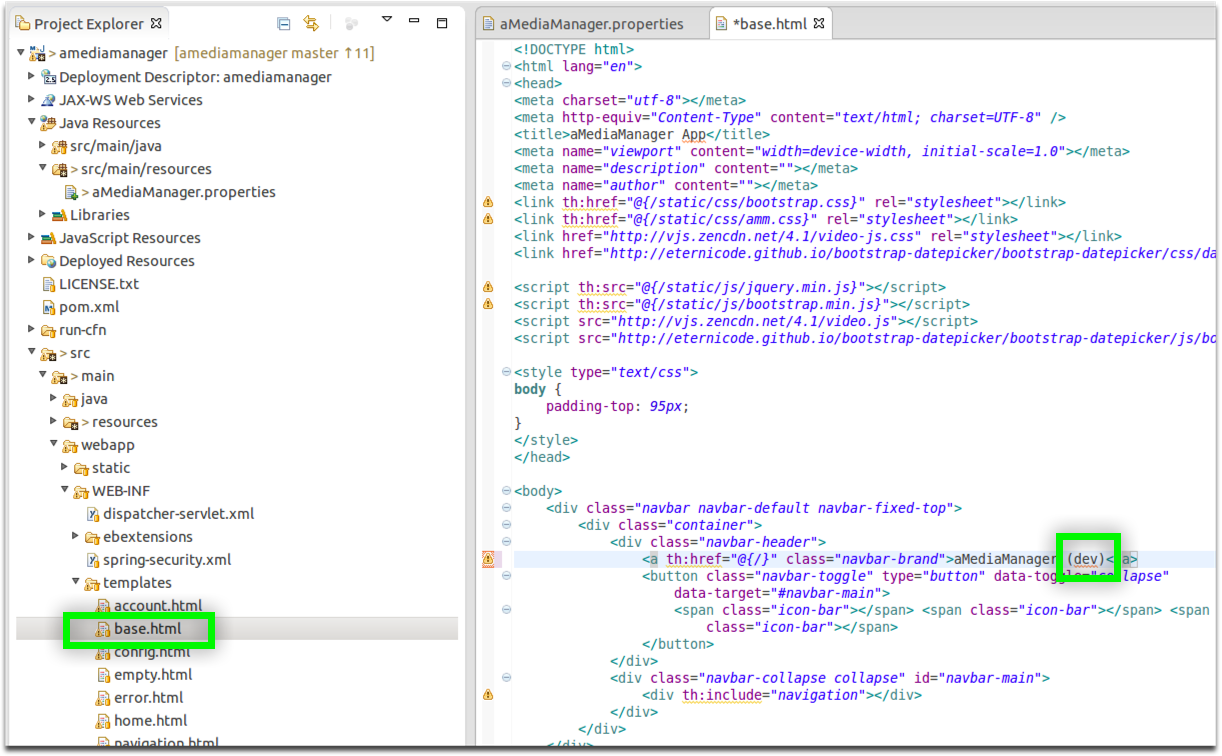
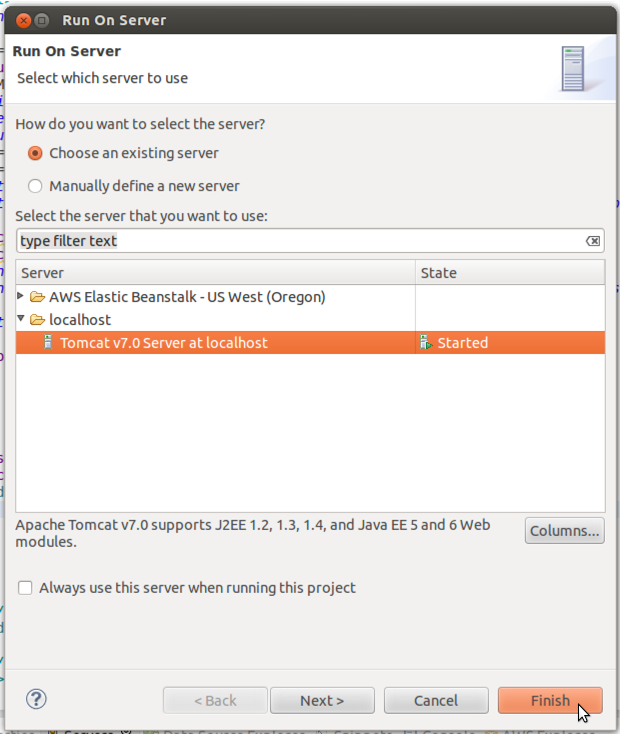
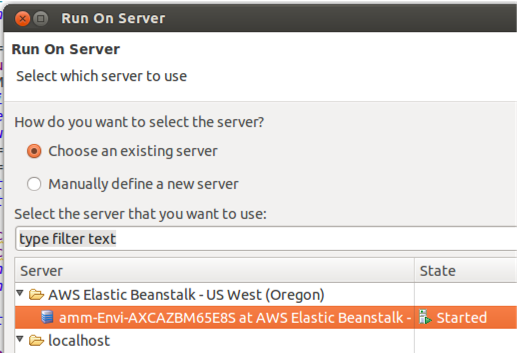
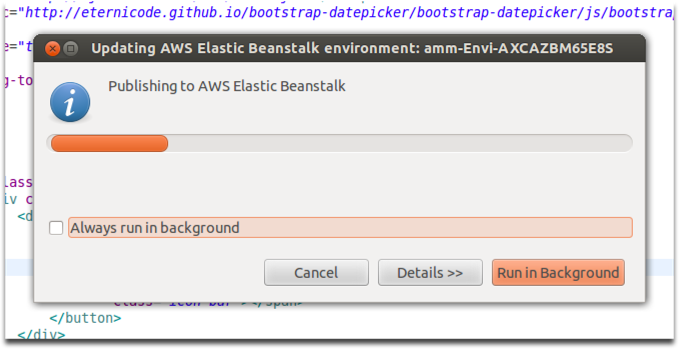
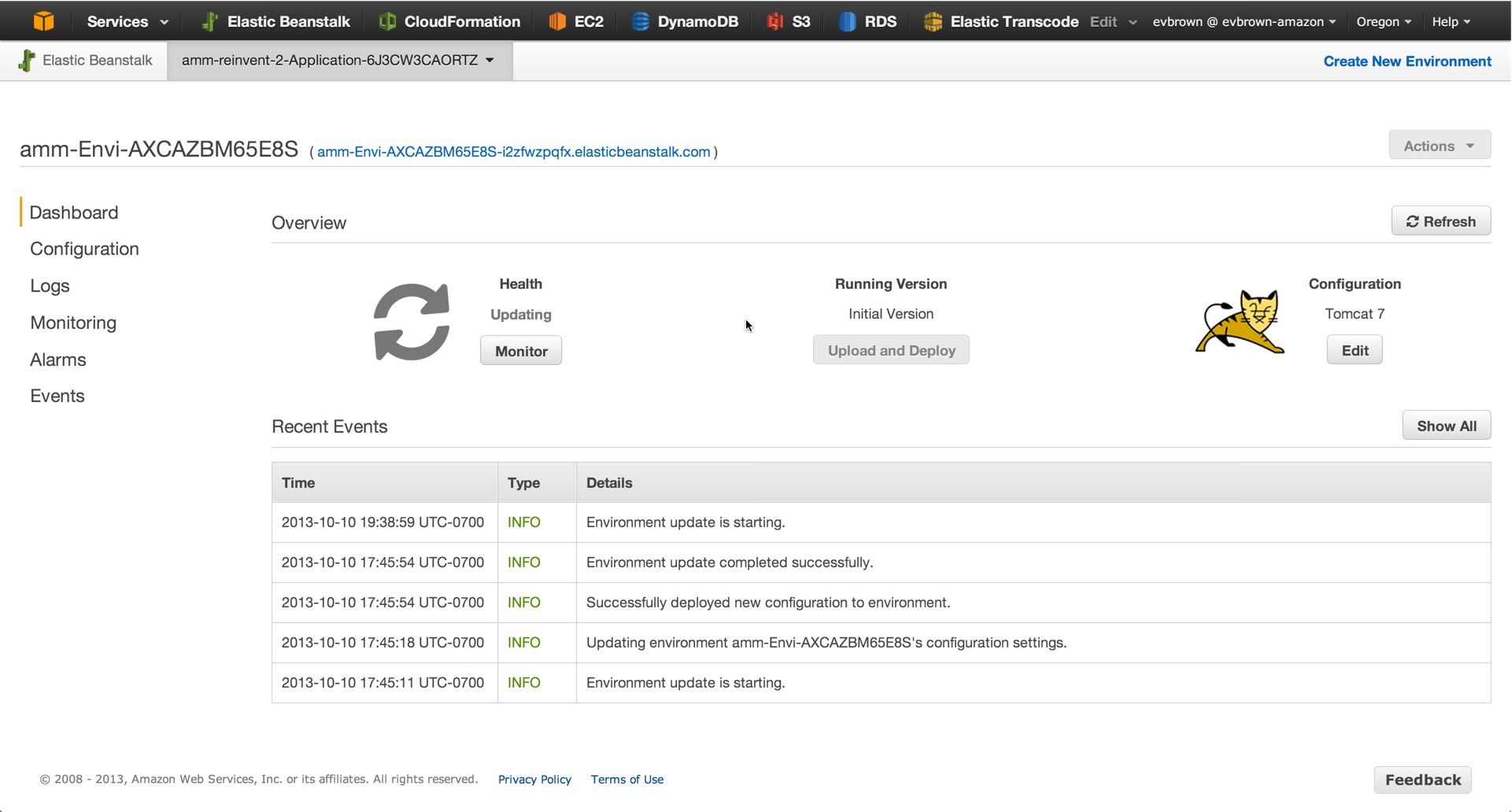
For this lab we want our local development environment to have the same configuration as the application deployed in EB. We’ve added a simple way to copy configuration from the app in EB to your local development environment:

1. From the App Config page, copy the configuration in the textarea to your clipboard:  
   
2. In Eclipse, navigate to amediamanager > Java Resources > src/main/resources and rename aMediaManager.properties.default to aMediaManager.properties:  
   
3. Open the new aMediaManager.properties file and paste in the configuration you copied previously:  
   

Modify and Deploy Application

# Modify and Deploy Application

In this section you will make a small change to the application, then deploy it to both your local Tomcat 7 server as well as your Elastic Beanstalk environment.

1. Open src > main > webapp > WEB-INF > templates > base.html and append “(dev”) to the navbar header:  
   
2. Right-click the amediamanager project and choose Run As > Run on Server
3. First, deploy the app to Tomcat v7.0 Server at localhost:  
   
4. After the application deploys it should open either in your default web browser or in a browser tab in Eclipse.
5. Deploy the application to Elastic Beanstalk by again right-clicking the project, choosing Run As > Run on Server, and choosing your Elastic Beanstalk environment from the server list:  
   
6. Eclipse will indicate the progress of your deployment in the IDE:  
   
7. Open your Elastic Beanstalk environment in a web browser view its status, which should be Updating while the new version is deployed from Eclipse:  
   

Roll Back to Initial Version

# Roll Back to Initial Version

Lorem ipsum

Project Anatomy

# Project Anatomy

Lorem ipsum