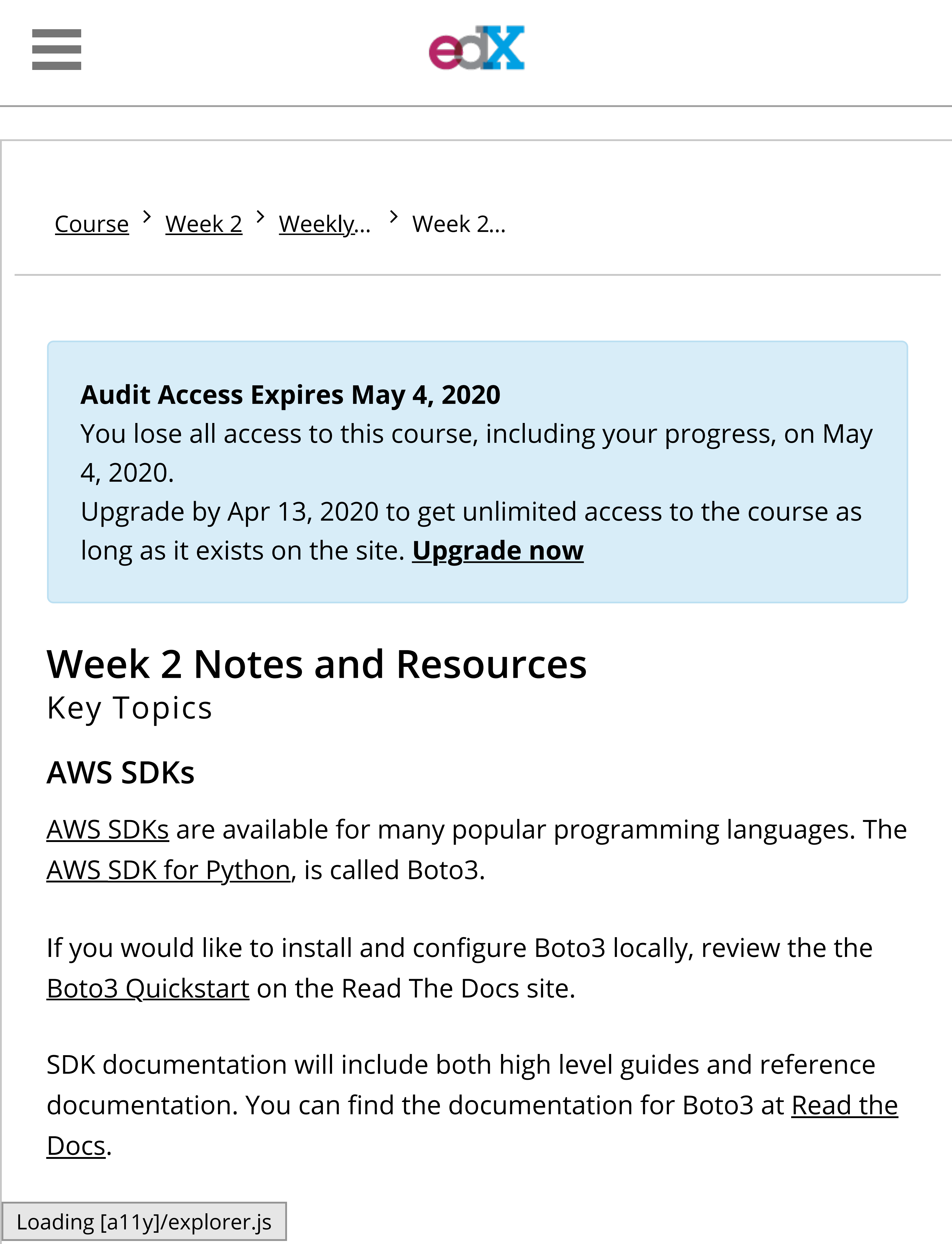

Audit Access Expires May 4, 2020
You lose all access to this course, including your progress, on May 4, 2020.
Upgrade by Apr 13, 2020 to get unlimited access to the course as long as it exists on the site. **Upgrade now**

Week 2 Notes and Resources

Key Topics

AWS SDKs

AWS SDKs are available for many popular programming languages. The AWS SDK for Python, is called Boto3.

If you would like to install and configure Boto3 locally, review the the Boto3 Quickstart on the Read The Docs site.

SDK documentation will include both high level guides and reference documentation. You can find the documentation for Boto3 at Read the Docs.

Loading [a11y]/explorer.js

AWS Identity and Access Management

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. We typically use credentials from *IAM Users* or *IAM Roles* to authenticate with AWS when making API calls. We control the permissions for which API actions those Users or Roles can perform with *IAM Policies*.

Making AWS API Calls

AWS API requests are made against a specific API endpoints located in a specific AWS Region. In this class, we are using the "us-west-2" region which is located in Oregon. For example, here are the API endpoints for Rekognition and the API endpoints for S3.

API requests are typically *signed* with an access key belonging to an IAM User and using the Signature Version 4 Signing Process. It is also possible to sign these API requests using temporary security credentials such as those derived from an IAM Role.

AWS SDKs check several locations for credentials such as local configuration files and environment variables. If the SDK finds credentials, it will automatically sign your API requests for you. For example, here is how the Python SDK checks for Credentials.

Developing in the cloud with AWS Cloud9

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. You can run this development environment on a managed Amazon EC2 instance that automatically sleeps when you aren't using it.

Make sure to follow the exercise directions carefully when setting up your Cloud9 instance - it needs to be launched in the specified VPC so that you can access resources that you'll be creating in the coming weeks.

Amazon S3

Amazon Simple Storage Service (S3) is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices. You can store files as Objects within S3 Buckets.

In this course, we are using Amazon S3 to store photos.

By default, the objects you put into S3 are private. S3 allows you to use Bucket Policies, IAM Policies, and ACLs to grant permissions to the contents of the bucket. You can also use Presigned URLs for time-limited access to objects.

Presigned URLs are how we are granting access to images in our Python Application. The S3 Service Feature Guide for Boto3 contains Python examples of working with S3. In particular, you should review the sectin on Generating Presigned Urls.

Amazon Rekognition

Amazon Rekognition is a service that applies deep learning to analyze the contents of images and videos. It supports functionality such scene detection, face detection, even celebrity recognition. In this class, we are using the Detect Labels functionality which takes an image as input and returns labels with confidence values such as `{Name: lighthouse, Confidence: 98.4629}`

Additional Details

More on AWS Identity And Access Management

When you log in to AWS using your email address and password, you are authenticating as the Root User for the account. The best practice is to avoid logging in as Root except for a handful of operations that only the root user can perform.

Instead, you can create IAM Users within your AWS Account for yourself and for any others that need access to resources in your account. IAM Users have a set of *permanent credentials* such as an Access Key for API access or a Console Password.

Permissions are granted or denied with IAM Policies. By default, all permissions are denied unless explicitly granted. You may select predefined permissions from the list of AWS Managed Policies or define your own custom IAM policies.

If you find that you'll have several users who need similar permissions, you can define an IAM Group and associate your users to the group.

When working with AWS Services, you may also encounter IAM Roles. Many AWS services require that you use roles to control what that service can access. IAM Roles provide only *temporary security credentials*. One common case is allowing the EC2 service to distribute credentials to your application code running on an EC2 instance. Roles can also enable other scenarios in the enterprise such as cross-account access and identity federation.

AWS Signature V4 Process

If you'd like to explore Signature V4 Signing further, here's some example code that creates HTTP requests and generates the appropriate signature.

What you accomplished this week

- You created an IAM User and are following the best practice of not using Root User in your AWS Account
- You installed the AWS SDK and configured credentials
- You launched and configured Cloud9 IDE environment
- You ran a Python Application that makes AWS API calls to Amazon S3 and AWS Rekognition

Learn About Verified Certificates