AWS for Librarians 2: Using AWS services

vanderbi.lt/learnaws

Steve Baskauf

Command line interface (CLI)

- The CLI is the simplest and fastest way to carry out AWS operations.
- CLI commands are carried out:
  - on Mac OS in the Linux shell (Terminal application)
  - on Windows using the Command Prompt application or PowerShell
  - on AWS servers (EC2 instances) running Linux
- Sequences of CLI commands can be automated using Linux shell scripts or Windows batch files.
- Scripting via software alternative: software development kits (SDKs) for common languages

Installing the CLI

- Simplest installation option is using Python's PIP package manager.
- Other specific options depending on Mac or PC
- Go to the link on the landing page for setup and configuration instructions.

- Test with S3 bucket commands

AWS Machine Learning and Media Services

Media Services

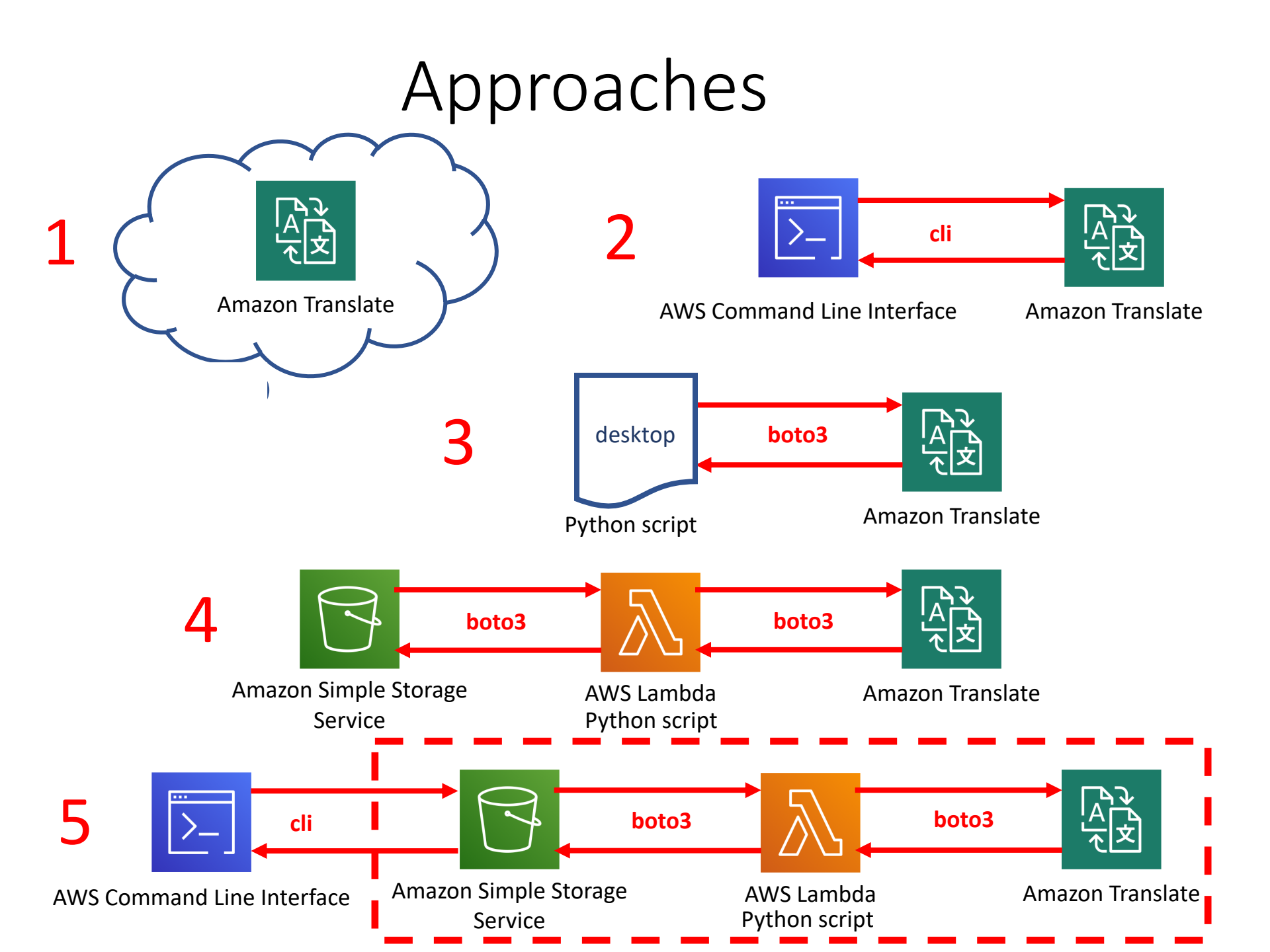
- Elastic Transcoder
- Kinesis Video Streams
- MediaConnect
- MediaConvert
- MediaLive
- MediaPackage
- MediaStore
- MediaTailor
- Elemental Appliances & Software

Machine Learning

- Amazon SageMaker
- Amazon Comprehend
- AWS DeepLens
- Amazon Lex
- Machine Learning
- Amazon Polly
- Rekognition
- Amazon Transcribe
- Amazon Translate
- Amazon Personalize
- Amazon Forecast
- Amazon Textract
- AWS DeepRacer

Services of interest to librarians and amenable to following approach.

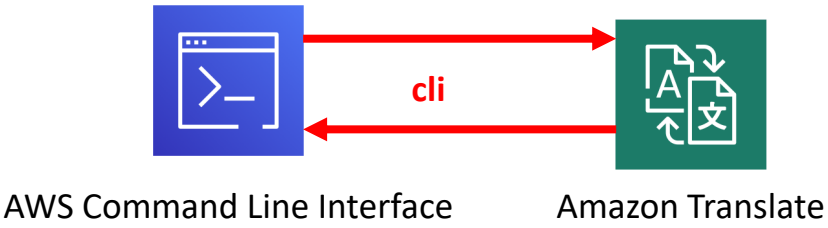
Approaches


1
Amazon Translate
2
cli
AWS Command Line Interface
Amazon Translate
3
desktop
boto3
Python script
Amazon Translate
4
boto3
boto3
Amazon Simple Storage Service
AWS Lambda Python script
Amazon Translate
5
cli
boto3
boto3
AWS Command Line Interface
Amazon Simple Storage Service
AWS Lambda Python script
Amazon Translate

Approach 1: Run Translate using web interface

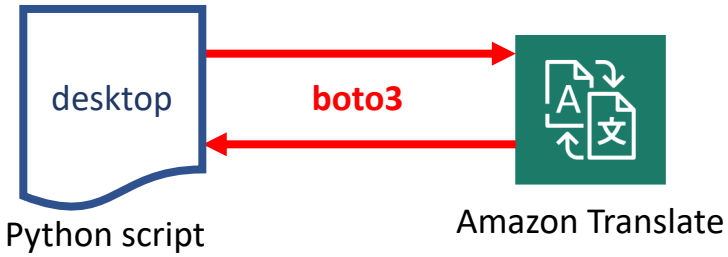
Instructions on web page "Using a service by web interface, CLI, and by script"

Approach 2: Invoke Translate using CLI



Instructions on web page "Using a service by web interface, CLI, and by script"

Approach 3: Invoke Translate using a desktop Python script



Instructions on web page "Using a service by web interface, CLI, and by script"

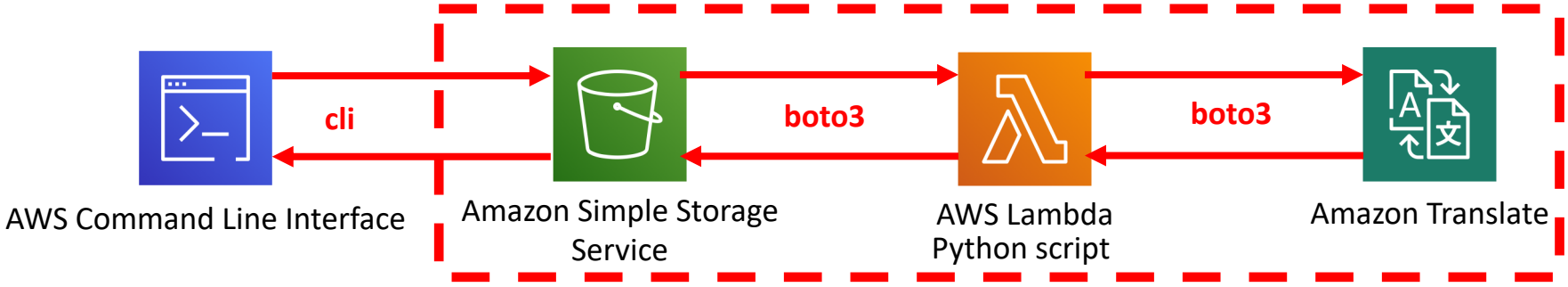
Approach 4: Invoke Translate using a Lambda Python script and S3 storage (S3 triggered)

Instructions on web page "Creating a serverless application using AWS Lambda"

How Lambdas behave

- Lambdas need permissions just like a user
- Application permissions are called "**roles**".
- Lambdas are triggered by some event.
- Lambdas can be created and tested using the web interface. The user must have permissions for Lambdas.
- Performance of independently operation Lambdas is done through CloudWatch logs.
- The logs show errors and output of **`print()`** functions.

Approach 5: Invoke Translate using a Lambda Python script and S3 storage (S3 triggered via CLI)



Instructions on web page "Creating a serverless application using AWS Lambda"

Creating other kinds of serverless applications

- The output of one service can trigger a second service (e.g. Textract can extract text from an image, then Translate can translate that text to another language).
- **CRON** jobs can be used to trigger monitoring Lambdas that can pull data from an API and carry out some action
  - Example rainfall email, weather visualization
  - Plan to use this to pull API data (social media, etc.) into GitHub for the library dashboard.
- **Step functions** are a way to control serverless flow