

[600003] Cloud Computing (F'18)

Term Project (Week 3, Final) Report

팀 정보

No.	이름	학번
1	임병준	20165157
2	염희수	20165143
3	한승탁	20135357
4	김동민	20165111
5	이헌우	20165156
팀 이름: 3 팀		

<목차>

팀 정보.....	1
결과물 제출	2
보고서 작성 가이드	2
Part I: 최종 보고서	3
Part II: 절차서.....	5
Part III: 프로젝트 진행중 발생한 문제점 및 해결방안.....	6
Part IV: 개선방향	7

* 보고서 작성이 완료되면 <목차>를 업데이트 해 주세요. (방법: “목차” 클릭>“목차 업데이트”>“목차 전체 업데이트”)

결과물 제출

- SmartCampus 에 업로드 해야 하는 제출물 목록은 다음과 같습니다.
 - Report (프로젝트 보고서), **PDF 형식**
 - 프로젝트 발표자료 (ppt, pptx, pdf 등)
 - GitHub 프로젝트를 다운로드 한 zip 파일
- 제출 기한을 넘기면 자동으로 0 점 처리됩니다.
- 팀 과제이며, 팀별로 한 명만 제출하면 됩니다.

* 보고서 및 발표자료도 GitHub 에서 관리해야 합니다!!!

보고서 작성 가이드

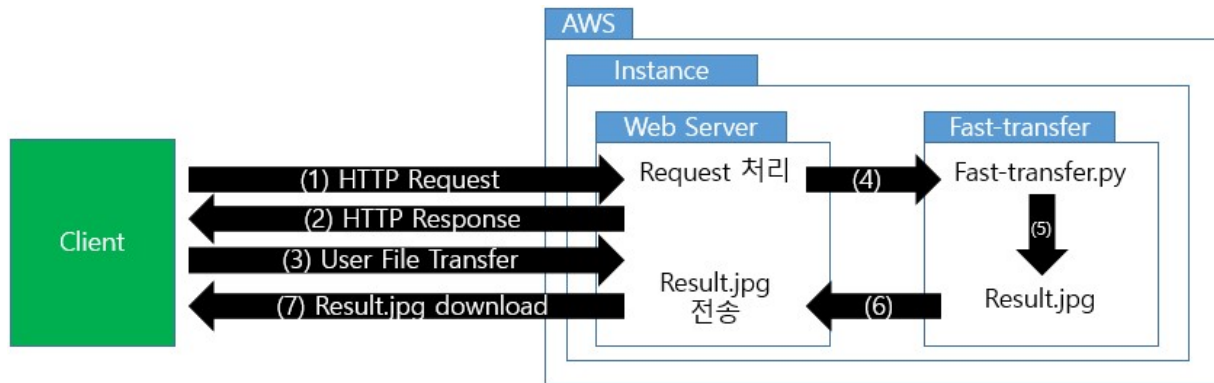
- Week 3 시작: 2018. 12. 11. (Tue) 11:00am
- Week 3 마감: 2018. 12. 14. (Tue) 10:59am
- 매 주차별로 반드시 1 회 이상 GitHub 에 commit 한 내역이 있어야 합니다. (팀별로 1 회 이상, 개인별 1 회 아님). GitHub 에 commit 한 내역이 없을 경우, 큰 감점을 받게 됩니다.

본 보고서는 총 3 개의 파트로 구성되어 있습니다.

- Part I: 최종 보고서 [25pts]
 - 1 주차 보고서 내용 전체를 포함하여 작성하고, 그 동안 추가/수정된 내용이 있다면 그에 맞게 보고서를 수정해야 합니다.
- Part II: 절차서 [25pts]
 - 팀 프로젝트로 개발한 결과물을 누구나 손쉽게 따라할 수 있도록, 각 단계별로 어떻게 진행하는지 절차서를 작성하세요. 스크린샷을 첨부하면 더 좋습니다.
- Part III: 프로젝트 진행중 발생한 문제점 및 해결방안 [25pts]
 - 2 주차 보고서 내용중 “프로젝트 진행 중 발생하는 문제점 및 해결방안”에 대한 내용을 포함하는 내용으로 작성하세요.
 - 그 외에 추가로 발생한 문제점이 있었다면 해당 문제점 및 해결방안을 기술하세요.
- Part IV: 개선방향 [25pts]
 - 개발 결과물을 앞으로 어떻게 개선할 수 있을지에 대한 의견을 서술하세요.

Part I: 최종 보고서

- github 주소 : <https://github.com/BJ-Lim/Cloud>
- 제목 : AWS 를 이용하여 내 사진을 명화로 만드는 웹 서버 구축하기
- 선정 동기 : 딥러닝은 현재 매우 활발히 연구되고 있는 분야입니다. 우리가 과거에는 상상도 할 수 없었던 많은 일들을 가능하게 해주고, 우리의 삶을 한층 더 낮게 만들 수 있는 기술임에 분명합니다. 얼마전 구글의 deepmind 팀에서 딥러닝 기술을 이용해 알파고로 세상을 떠들썩하게 만들었습니다. 하지만 이러한 기술을 우리는 보기만할 뿐 직접 경험해 보기는 힘들었습니다. 우리는 비전공자들도 함께 딥러닝을 느끼게 하고 싶었고, 인공지능에 관심을 가지게 만들고 싶었습니다. 이 프로젝트에서 AWS 로 구축된 웹 서버를 통해 비전공자들은 AI 가 얼마나 우리의 삶에 가까이 다가왔는지를 손쉽게 느낄 수 있을 것입니다. 또한 전공자(혹은 관심 있는 사람)들은 직접 tutorial 을 통해 AWS 로 간단하게 서버를 구축해 볼 수 있습니다. 이 프로젝트를 통해 많은 사람들이 AI 에 관심을 가지고, 손쉽게 다가갈 수 있는 분야라는 것을 느꼈으면 좋겠습니다.
- 내용 : 사용자가 웹으로 사진을 올리고 fast-style-transfer 기술로 변환된 사진을 다운받을 수 있는 웹 페이지 제공합니다. 이 때 fast-style-transfer 기술이란, 컴퓨터가 특정한 사진을 기계학습 하고 새로운 사진을 받았을 때 특정한 사진과 같은 스타일로 변환해 주는 기술입니다.
- 목표 : 우리는 클라우드, 오픈소스를 이용하면 쉽게 개발이 가능하다고 배웠습니다. 하지만 이게 실제로 가능한 것인지 직접 해볼 기회는 없었습니다. 이 프로젝트를 통해 오픈소스인 fast-style-transfer 프로젝트를 웹 서버에 올리고, 사용자와 연결하여 이 기술을 커맨드 창이 아닌 웹 GUI 를 통해 제공할 수 있는 플랫폼을 만들어 보고자 합니다.
- 개발 / 구현 내용
 - 웹 페이지 (업로드 / 다운로드 페이지)
 - 사용자는 웹 페이지에 자신의 사진을 업로드하고, 변환된 사진을 다운로드 받을 수 있습니다.
 - 웹 페이지는 어떤 스타일의 학습된 사진들(명화)을 가지고 있는지 사용자에게 보여줍니다.
 - 사용자가 선택한 스타일을 웹 서버로 전송합니다.
 - 웹 서버
 - fast-style-transfer 프로젝트가 실행 가능하도록 환경을 구성합니다.
 - 웹 페이지로부터 파일을 업로드 받고 open-source 프로젝트인 fast-style-transfer 을 실행합니다. 이때 사용자가 선택한 스타일을 매개변수로 전달합니다.
- 구성도(다음 페이지)



(1) HTTP 페이지를 요청합니다.

※주의사항 : DNS 서버 관련된 과정은 생략되었습니다.

(2) 서버는 클라이언트로 HTTP 웹 페이지를 전송합니다.

(3) 사용자는 웹 페이지에서 변환할 파일을 업로드합니다.

(4) 웹 서버는 사용자가 업로드한 파일을 수신하고, fast-transfer.py 를 실행합니다.

(5) fast-transfer.py 는 실행 후 결과 파일인 result.jpg 를 반환합니다.

(6),(7) 웹 서버는 (5)에서 생성된 파일인 result.jpg 를 사용자에게 전송합니다.

● 기대 효과 및 활용 방안

- AWS 를 사용하여 간단하게 서버를 구축해보고, tutorial 을 통해 과정을 배울 수 있다.
- 비전공자들이 쉽게 AI 기술을 경험해 볼 수 있다.
- 전공자들은 쉽게 플랫폼을 구성하는 방법을 배우고 해볼 수 있다.
- 실제 오픈소스 프로젝트와 클라우드 서비스가 어떻게 결합되는지 좋은 예제로 활용이 가능하다.

Part II: 절차서

※ 과정이 너무 길어 github 의 링크로 대체합니다.

[전체 과정]

<https://github.com/BJ-Lim/Cloud/tree/master/tutorial>

[1. AWS 인스턴스 생성]

[2. 웹 서버 설치]

[3. 파일 옮기기]

[4. 환경변수 설정]

[5. 데모]

Part III: 프로젝트 진행중 발생한 문제점 및 해결방안

- 1) 1. Framework 가 너무 복잡해서, 설치해야 할 파일이 많았고 충돌 문제도 빈번하게 발생했습니다.
 - ➔ 미리 만들어놓은 이미지를 사용하여 환경을 구축하기로 했습니다. 아무것도 깔려있지 않은 인스턴스 대신, machine learning 관련해서 사전 구축된 famework image 를 인스턴스로 설치하기로 했습니다. (Deep Learning AMI Ubuntu)
- 2) Fast-transfer 파일을 어떻게 AWS 의 인스턴스로 옮길 것인가가 문제였습니다. 파일 크기가 약 700MB 정도 되기 때문에, 깃 허브는 불가능했고, 마찬가지로 링크 또한 구글 드라이브로 연동되어 있어 wget 이 불가능했습니다.
 - ➔ 먼저 파일을 파일 공유 폴더를 이용하여 local Vitual Machine 으로 옮겼습니다.그 후 리눅스 서버간에 scp 명령을 사용하여 파일을 전송했습니다.
- 3) 인스턴스가 제한을 초과했다는 메시지가 발생하며 인스턴스를 생성할 수 없었습니다.
 - ➔ 지원(support)를 통해 인스턴스의 개수를 늘렸습니다. AWS 는 특이한 것이, 각각의 인스턴스 등급(t2.2xlarge 등)별로 인스턴스의 개수 제한이 걸려있고, 보통 처음은 0 으로 셋팅됩니다. 문의하면 약 10 분 이내에 인스턴스의 개수 제한을 상향조정할 수 있습니다.
- 4) GPU 가속기가 linux 환경에서는 사용이 불가능했습니다.(GPU 인스턴스가 사용이 불가능한 것은 아닙니다.)
 - ➔ GPU Instance 로 테스트 해 봤지만, 단순하게 학습된 네트워크로 실행하는 경우 CPU 가 2 배정도 더 빠른 속도를 보여줬습니다. 따라서 CPU 서버를 사용하기로 했습니다.
- 5) Php 로 쉘 코드 실행시 명령이 제대로 수행하지 않는 문제가 발생했습니다.
 - ➔ 환경변수의 문제였고, www-data 사용자가 실행될 때 ubuntu 사용자와 동일한 환경변수를 가질 수 있도록 셋팅해 주었습니다.

Part IV: 개선방향

- 보안 문제 : 만약 이 서버를 상용화하려면 파일을 서버로 업로드 할때 파일을 체크해 보고 서버로 업로드 하는 것이 안전합니다.
- 부하 분산 : 현재 서버는 auto scaling 을 지원하지 않습니다. load balancer 와 automated scaling listener 등을 추가하여 부하를 분산시키고, 부하에 따라 자동으로 서버를 추가하는 시스템을 구축할 수 있습니다.
- 가용성 향상 : 현재 서버는 고 가용성을 보장하지 않습니다. 하지만 이 서비스의 특성을 고려한다면 고 가용성은 꼭 필요한 요소는 아닙니다.
- 다중 사용자 지원 : 현재는 동시에 한 명의 사용자만 서버를 이용할 수 있습니다. 그 이유는 fast-style-transfer 을 실행하면 그 트랜잭션이 해당 서버의 자원을 거의 대부분 소모하기 때문입니다. 이를 막기 위해서 사용자별로 CPU 사용량을 제한하는 등의 방법을 사용할 수 있습니다.

[끝]