MES시스템2팀 김유라

1. Git & GitHub
   1. Git은 뭐고 Github은 먼가요?
      * 1. 버전관리 : 프로그램 개발할 때 이전 기능 다시 가져옴, 잘못된 코드 섞은 거 복원, 코드 바뀐 내역 이력관리
      * 2. 협업 : 한 프로젝트에 여러 개발자 역할 나눠 개발 -> 테스트 시 통합 프로젝트 파일이 있어야 테스트 가능
      * 3. Commit : 프로젝트 내용 박제 (의미있는 변화 있을때마다 commit
      * 4. Log : 이전 동안의 박제 내역 확인 후 복원 ( reset –hard XXX)
      * 5. Branch : 주가되는 코드 메인브랜치,
      * 6. Github : git – 영상찍는 툴, github - 유튜브 같은 개념
   2. 가장 쉬운 Git 강좌 – (상) 혼자작업편
      * Cli, 소스트리 사용 -> 소스트리가 시각화 면에서 더 좋은듯
      * Vs code 이용
      * Git init : 빈폴더 생성
      * Git status : 상태 변경
      * Git add
      * Git reset –hard
      * Revert
      * Rebase -> merge랑 비교
      * 바로 add, commit 하는건 지양
   3. 가장 쉬운 Git 강좌 – (하) github 편
      * 코무디 1주차때 실습 완료
      * Gitignore 파일 : 데이터베이스 내용과 같이 보안내용은 git 관리에서 제외시킬 거
      * 커밋 메시지 내용 통일시켜야할 듯
      * Conflict : a가 수정해서 push 했는데 b가 이걸 반영안하고 pull하면 발생
2. Docker
   1. Docker가 뭐고 왜 쓰는건가요?
      * 독립된 업무환경 구축
      * 각 요소들이 설치된 모습 이미지로 박제해서 저장
      * Docker -> dockerhub
      * 컨테이너 : 독립된 가상 공간 -> 다른 버전 이미지라도 다른 컨데이너 내에서 돌릴수 있음
      * 가상화와 비교 : 가상화 – os내 가상 os 여러 개, 도커 – 한 os 내 도커엔진 내에서 여러 컨테이너(독립된 실행환경) -> 가상화보다 도커가 가벼움
3. MSA
   1. 왜 우리는 마이크로서비스를 구현하고자 하는가?
   2. Digital Transformation with Pivotal
      * 김영태 상무,Pivotal
      * Technology : 고객이 원하는 소프트웨어를 빠르게
      * Culture : 고객의 입맛에 맞춰
      * Digital Transformation을 이루기위해 Technology와 Culture 밸런스 맞춰야함
      * Pivotal Platform(Technology) – PAS(Cloud) /PKS
      * Pivotal Labs (Culture) – 에자일 기반
      * 85% 정도 에자일 기반으로? -> 컬쳐만족하려면 에자일 필수?
   3. 마이크로서비스 기반 클라우드 아키텍쳐 모범 사례
      * 10년전부터 활발히 사용
      * Monolith -> Microservies -> Functions
      * Daum
      * Netflix
      * SK Planet 11번가