1부 도커 컨테이너와 이미지 이해하기

1장 시작하기 전에 ····· 25

1.1 컨테이너가 IT 세상을 점령한 이유 26
1.2 대상 독자 35
1.3 실습 환경 구축하기 36
1.4 바로 활용하기 43

2장 도커의 기본적인 사용법 ····· 45

2.1 컨테이너로 Hello World 실행하기 46
2.2 컨테이너란 무엇인가? 49
2.3 컨테이너를 원격 컴퓨터처럼 사용하기 52
2.4 컨테이너를 사용해 웹 사이트 호스팅하기 56
2.5 도커가 컨테이너를 실행하는 원리 60
2.6 연습 문제: 컨테이너 파일 시스템 다루기 63

3장 도커 이미지 만들기 ····· 65

3.1 도커 허브에 공유된 이미지 사용하기 66
3.2 Dockerfile 작성하기 71
3.3 컨테이너 이미지 빌드하기 73

3.4 도커 이미지와 이미지 레이어 이해하기 76
3.5 이미지 레이어 캐시를 이용한 Dockerfile 스크립트 최적화 79
3.6 연습 문제 82

4장 애플리케이션 소스 코드에서 도커 이미지까지 ····· 83

4.1 Dockerfile이 있는데 빌드 서버가 필요할까? 84
4.2 애플리케이션 빌드 실전 예제: 자바 소스 코드 88
4.3 애플리케이션 빌드 실전 예제: Node.js 소스 코드 93
4.4 애플리케이션 빌드 실전 예제: Go 소스 코드 97
4.5 멀티 스테이지 Dockerfile 스크립트 이해하기 101
4.6 연습 문제 102

5장 도커 허브 등 레지스트리에 이미지 공유하기 ····· 103

5.1 레지스트리, 리포지터리, 이미지 태그 다루기 104
5.2 도커 허브에 직접 빌드한 이미지 푸시하기 106
5.3 나만의 도커 레지스트리 운영하기 111
5.4 이미지 태그를 효율적으로 사용하기 115
5.5 공식 이미지에서 골든 이미지로 전환하기 117
5.6 연습 문제 120

6장 도커 볼륨을 이용한 퍼시스턴트 스토리지 ····· 121

6.1 컨테이너 속 데이터가 사라지는 이유 122

6.2 도커 볼륨을 사용하는 컨테이너 실행하기 127

6.3 파일 시스템 마운트를 사용하는 컨테이너 실행하기 133

6.4 파일 시스템 마운트의 한계점 137

6.5 컨테이너의 파일 시스템은 어떻게 만들어지는가? 142

6.6 연습 문제 144

2부 컨테이너로 분산 애플리케이션 실행하기

7장 도커 컴포즈로 분산 애플리케이션 실행하기 ····· 149

7.1 도커 컴포즈 파일의 구조 150

7.2 도커 컴포즈를 사용해 여러 컨테이너로 구성된 애플리케이션 실행하기 155

7.3 도커 컨테이너 간의 통신 162

7.4 도커 컴포즈로 애플리케이션 설정값 지정하기 166

7.5 도커 컴포즈도 만능은 아니다 170

7.6 연습 문제 172

8장 헬스 체크와 디펜던시 체크로 애플리케이션의 신뢰성 확보하기 ····· 173

8.1 헬스 체크를 지원하는 도커 이미지 빌드하기 174

8.2 디펜던시 체크가 적용된 컨테이너 실행하기 181

8.3 애플리케이션 체크를 위한 커스텀 유틸리티 만들기 185

8.4 도커 컴포즈에 헬스 체크와 디펜던시 체크 정의하기 190

8.5 헬스 체크와 디펜던시 체크로 복원력 있는 애플리케이션을 만들 수 있는 이유 194

8.6 연습 문제 196

9장 컨테이너 모니터링으로 투명성 있는 애플리케이션 만들기 ····· 197

9.1 컨테이너화된 애플리케이션에서 사용되는 모니터링 기술 스택 198

9.2 애플리케이션의 측정값 출력하기 204

9.3 측정값 수집을 맡을 프로메테우스 컨테이너 실행하기 209

9.4 측정값 시각화를 위한 그라파나 컨테이너 실행하기 215

9.5 투명성의 수준 223

9.6 연습 문제 225

10장 도커 컴포즈를 이용한 여러 환경 구성 ····· 227

10.1 도커 컴포즈로 여러 개의 애플리케이션 배포하기 228

10.2 도커 컴포즈의 오버라이드 파일 232

10.3 환경 변수와 비밀값을 이용해 설정 주입하기 240

10.4 확장 필드로 중복 제거하기 246

10.5 도커를 이용한 설정 워크플로 이해하기 249

10.6 연습 문제 251

11장 도커와 도커 컴포즈를 이용한 애플리케이션 빌드 및 테스트 ····· 253

11.1 도커를 이용한 지속적 통합 절차 254

11.2 도커를 이용한 빌드 인프라스트럭처 구축하기 256

11.3 도커 컴포즈를 이용한 빌드 설정 266

11.4 도커 외의 의존 모듈이 불필요한 CI 작업 만들기 271

11.5 CI 파이프라인에 관계된 컨테이너 275

11.6 연습 문제 277

3부 컨테이너 오케스트레이션을 이용한 스케일링

12장 컨테이너 오케스트레이션: 도커 스웜과 쿠버네티스 ····· 281

12.1 컨테이너 오케스트레이션 도구란? 282

12.2 도커 스웜으로 클러스터 만들기 285

12.3 도커 스웜 서비스로 애플리케이션 실행하기 289

12.4 클러스터 환경에서 네트워크 트래픽 관리하기 297

12.5 도커 스웜과 쿠버네티스 중 무엇을 사용할까? 305

12.6 연습 문제 308

13장 도커 스웜 스택으로 분산 애플리케이션 배포하기 ····· 309

13.1 도커 컴포즈를 사용한 운영 환경 310

13.2 컨피그 객체를 이용한 설정값 관리 316

13.3 비밀값을 이용한 대외비 설정 정보 관리하기 321

13.4 스웜에서 볼륨 사용하기 325

13.5 클러스터는 스택을 어떻게 관리하는가? 330

13.6 연습 문제 332

14장 업그레이드와 롤백을 이용한 업데이트 자동화 ····· 333

14.1 도커를 사용한 애플리케이션 업그레이드 프로세스 334
14.2 운영 환경을 위한 롤링 업데이트 설정하기 340
14.3 서비스 롤백 설정하기 344
14.4 클러스터의 중단 시간 350
14.5 스웜 클러스터의 고가용성 354
14.6 연습 문제 356

15장 보안 원격 접근 및 CI/CD를 위한 도커 설정 ····· 357

15.1 도커 API의 엔드포인트 형태 358
15.2 보안 원격 접근을 위한 도커 엔진 설정 364
15.3 도커 컨텍스트를 사용해 원격 엔진에서 작업하기 373
15.4 지속적 통합 파이프라인에 지속적 배포 추가하기 377
15.5 도커 리소스의 접근 모델 384
15.6 연습 문제 385

16장 어디서든 실행할 수 있는 도커 이미지 만들기: 리눅스, 윈도, 인텔, ARM ····· 387

16.1 다중 아키텍처 이미지가 중요한 이유 388

16.2 다중 아키텍처 이미지를 만들기 위한 Dockerfile 스크립트 392

16.3 다중 아키텍처 이미지를 레지스트리에 푸시하기 398

16.4 도커 Buildx를 사용해 다중 아키텍처 이미지 빌드하기 404

16.5 개발 로드맵과 다중 아키텍처 이미지 409

16.6 연습 문제 410

4부 운영 환경 투입을 위한 컨테이너 준비하기

17장 도커 이미지 최적화하기: 보안, 용량, 속도 ····· 415

17.1 도커 이미지를 최적화하는 방법 416

17.2 좋은 기반 이미지를 고르는 법 421

17.3 이미지 레이어 수와 이미지 크기는 최소한으로 428

17.4 멀티 스테이지 빌드를 한 단계 업그레이드하기 432

17.5 최적화가 중요한 이유 436

17.6 연습 문제 437

18장 컨테이너의 애플리케이션 설정 관리 ····· 439

18.1 다단 애플리케이션 설정 440
18.2 환경별 설정 패키징하기 445
18.3 런타임에서 설정 읽어 들이기 449
18.4 레거시 애플리케이션에 설정 전략 적용하기 454
18.5 유연한 설정 모델의 이점 460
18.6 연습 문제 462

19장 도커를 이용한 로그 생성 및 관리 ····· 463

19.1 표준 에러 스트림과 표준 출력 스트림 464
19.2 다른 곳으로 출력된 로그를 stdout 스트림에 전달하기 469
19.3 컨테이너 로그 수집 및 포워딩하기 474
19.4 로그 출력 및 로그 컬렉션 관리하기 481
19.5 컨테이너의 로깅 모델 486
19.6 연습 문제 488

20장 리버스 프록시를 이용해 컨테이너 HTTP 트래픽 제어하기 ····· 489

20.1 리버스 프록시란? 490
20.2 리버스 프록시의 라우팅과 SSL 적용하기 496
20.3 프록시를 이용한 성능 및 신뢰성 개선 502

20.4 클라우드 네이티브 리버스 프록시 507

20.5 리버스 프록시를 활용한 패턴의 이해 514

20.6 연습 문제 517

21장 메시지 큐를 이용한 비동기 통신 ····· 519

21.1 비동기 메시징이란? 520

21.2 클라우드 네이티브 메시지 큐 사용하기 525

21.3 메시지 수신 및 처리 530

21.4 메시지 핸들러로 기능 추가하기 533

21.5 비동기 메시징 패턴 이해하기 537

21.6 연습 문제 539

22장 끝없는 정진 ····· 541

22.1 도커를 이용한 개념 검증 542

22.2 소속 조직에서 도커의 유용함을 입증하라 543

22.3 운영 환경으로 가는 길 544

22.4 도커 커뮤니티 소개 545

찾아보기 546