コンテナ技術リカレント教育プログラム

本コンテナ技術リカレント教育プログラムは、文部科学省の教育政策推進事業委託費による委託事業として、一般社団法人全国専門学校情報教育協会が実施した令和７年度「専門職業人材の最新技能アップデートのための専修学校リカレント教育（リ・スキリング）推進事業」の成果物です。

情報技術者の技能アップデートのためのリカレント教育推進事業

目次

１．ネットワーク技術者向けリカレント教育プログラム・・・・・・・・・P 2

２．サーバー技術者向けリカレント教育プログラム・・・・・・・・・・・P20

３．アプリケーション技術者向けリカレント教育プログラム・・・・・・P39

はじめに

昨今、デジタル技術の進展は、情報システムの構成や開発・運用の在り方を根本から変えつつある。オンプレミスを前提とした従来型のシステムから、クラウド、コンテナ、クラウドネイティブ技術を基盤とするシステムへと移行が進み、企業の DX 推進やサービス提供の高度化は、もはや一部の先進企業に限られたものではなくなっている。

この変化は、情報技術者に求められる役割やスキルセットにも大きな影響を及ぼしている。ネットワーク、サーバー、アプリケーションといった従来の専門分野は依然として重要である一方、それぞれが独立して完結する時代は終わり、クラウドサービス、コンテナ基盤、Kubernetes、CI/CD、セキュリティ、可観測性といった要素を横断的に理解し、システム全体を俯瞰して設計・運用できる力が求められるようになっている。

しかしながら、多くの現場では、既存業務に追われる中で体系的に新技術を学び直す機会を確保することが難しく、個々の技術者の自己学習に依存したスキルアップが続いてきた。その結果、現場で求められる技術と、技術者が保有する知識・経験との間にギャップが生じ、DX 推進やクラウド移行のボトルネックとなるケースも少なくない。

こうした課題を背景として、本事業では、すでに実務経験を有する情報技術者を対象に、最新の技術動向を踏まえた「学び直し」の機会を提供するリカレント教育プログラムの構築に取り組んでいる。本プログラムは、単なる新技術の紹介や知識習得にとどまらず、現場で実際に利用されているクラウド構成やアーキテクチャを題材とし、設計・構築・運用・改善までを一連の流れとして学ぶことを重視している点に特徴がある。

また、ネットワーク技術者、サーバー技術者、アプリケーション技術者という異なる専門分野ごとに教育指針とカリキュラムを整理し、それぞれの強みや既存スキルを活かしながら、クラウドネイティブ時代に対応した技術者像へと段階的にステップアップできる構成としている。これにより、技術者が大きな職種転換を強いられることなく、現場に立脚した形でスキルの更新を図ることを可能としている。

本リカレントプログラムでは、こうした考え方のもとに設計された教育指針、教育内容、カリキュラム構成、ならびに産業界ニーズとの整合性を整理し、本事業が目指す方向性とその意義を明らかにしながら、リカレントによる人材育成を目指す。

１．ネットワーク技術者向けリカレント教育プログラム

１－１．教育指針（ネットワーク技術者科目）

本リカレント科目の目的は、ネットワーク設計・運用に携わる社会人を対象に、**仮想化・コンテナ・クラウド・クラウドネイティブ・サービスメッシュ**といった最新のネットワーク関連技術を体系的に学び、オンプレミス中心のネットワークエンジニアからクラウドネイティブ時代のネットワーク技術者へとステップアップすることをめざす点に置いている。

従来のネットワーク技術者は、L2/L3 スイッチやルータ、オンプレミスのファイアウォールやロードバランサを中心に扱ってきたが、クラウド環境では VPC・サブネット・セキュリティグループ・WAF・ALB/NLB、さらには**Kubernetes のネットワーク、サービスメッシュ上の L7 通信制御、ゼロトラストに基づく通信設計**が求められている。本プログラムは、こうした変化に対応するために、ネットワーク技術者がクラウド／コンテナ／Kubernetes／サービスメッシュを前提としたネットワークを設計・運用できる知識と技能を獲得することを教育目標としている。

具体的には、最初にコンテナ技術と Docker の基礎から学び、それを支える仮想化・名前空間・ネットワークの仕組みを理解したうえで、Kubernetes による**コンテナオーケストレーションとサービス提供の仕組み（Pod、Service、LoadBalancer、Ingress など）**を体系的に学ぶ構成としている。続いて、AWS を中心に VPC・サブネット・ルーティング・セキュリティグループ・ALB・WAF などのクラウドネットワーク要素を扱い、ECS/Fargate・EKS・RDS で構成されるクラウドネイティブな三層アーキテクチャを題材に、ネットワーク視点からの設計・構成・トラフィック制御を実践的に学ぶこととしている。

さらに、Istio を用いたサービスメッシュ環境での mTLS・認可制御・サービス間通信の制御や、Kiali・Prometheus・Grafana による通信経路の可視化と可観測性、ゼロトラストや DevSecOps に基づくネットワークセキュリティを扱い、**ネットワーク技術者がアプリケーションと基盤をまたいで安全な通信経路をデザインできる人材**となることを目標に据えている。

本科目は、社会人の学び直し（リカレント教育）として、既存のネットワーク知識を前提にしつつ、クラウドネイティブ環境で求められるネットワーク技術を短期間で集中的にアップデートする教育プログラムとして位置づけている。

１－２．教育内容の特徴

(1) ネットワーク技術者の視点からコンテナ・Kubernetes・クラウドを再構成している

本プログラムの特徴は、コンテナ・Kubernetes・クラウドを**「ネットワーク技術者の視点から」学び直す**点にある。コンテナや Kubernetes の基礎内容はサーバー技術者科目と共通する部分も多いが、ネットワーク技術者科目では、

- コンテナ間・Pod 間通信
- Service／LoadBalancer／Ingress によるトラフィック制御
- Kubernetes ネットワークモデル（オーバーレイ・CNI）
- AWS における VPC・サブネット・ルート・セキュリティグループ・NACL
- ALB・WAF による L7 レベルの制御

といった**“通信経路の設計と制御”に関わる要素**を中心に取り扱う教育内容としている。

これにより、従来のネットワーク機器中心の知識に加えて、「コンテナ・クラスタ・クラウドサービスをまたいだネットワーク」を一体として捉えられる力を育成することを特徴としている。

(2) クラウドネットワーク構成要素を体系的に扱い、オンプレとの違いを明確にしている

AWS・GCP・Azure の特徴、AWS のサービス群、ECS/Fargate・Aurora・Auto Scaling などを学ぶ際には、**「オンプレのネットワーク要素とどのように対応するか」**を意識した教育内容としている。たとえば、

- VPC／サブネット → 仮想的な L2/L3 セグメント
- セキュリティグループ／NACL → 仮想 FW 的な制御
- ALB → L7 ロードバランサ
- VPC エンドポイント → クラウド内の閉域的な通信経路

といった形で、従来のネットワーク構成との対応関係を整理し、クラウドネットワーク特有の設計や注意点を理解できるようにすることをねらいとしている。

(3) Kubernetes／EKS におけるネットワーク運用を実務レベルで扱っている

Kubernetes の科目群では、単に Pod や Deployment の操作を学ぶだけでなく、**Kubernetes 上で通信がどのように行われるのか**を詳細に扱う内容としている。具体的には、

- Pod IP と Service IP の関係
- ClusterIP／NodePort／LoadBalancer／ExternalName の違い
- Ingress による HTTP/HTTPS 経路の集約
- オーバーレイネットワークによる Pod 間通信

- EKS 上でのロードバランサ連携、EBS/EFS 連携
  などを通じて、「Kubernetes クラスタ全体を一つのネットワークとして設計・運用する」観点を重視している。

ネットワーク技術者が Kubernetes／EKS 環境を前提に、サービス公開・負荷分散・セキュリティ・ストレージ連携を含めたネットワーク設計を行えるようにすることを特徴としている。

(4) 三層アーキテクチャを題材に、ネットワークとアプリケーションを一体として設計する力を養っている

AWS 上に ALB＋ECS＋RDS を構成する科目では、三層アーキテクチャ（Web／AP／DB）を題材として、

- VPC・サブネット・セキュリティグループの設計
- インターネット公開部分と内部ネットワークの分離
- ALB と WAF による L7 防御と HTTPS 化
- ECS サービスへのトラフィック分配
- DB サブネットと RDS への安全な通信経路
- CI/CD による更新時の無停止リリース
  などを、ネットワーク視点で整理しながら構築する教育内容としている。

これにより、ネットワーク技術者が「アプリケーション構成と連動したネットワーク設計（トポロジ・アドレス設計・セキュリティ・可用性）」を一連のプロセスとして捉え、現場で求められるアーキテクチャ設計力を身につけることをめざしている。

(5) サービスメッシュと可観測性を通じて、上位レイヤの通信制御・分析力を育成している

Kubernetes セキュリティ・可観測性の科目群では、ネットワーク技術者にとって新しい領域である**サービスメッシュ／ゼロトラスト／DevSecOps／可観測性**を、通信経路の制御と分析という観点から扱う構成としている。
Istio による mTLS 通信、AuthorizationPolicy による認可制御、Secrets 管理に加え、Kiali によるサービス間通信の可視化、Prometheus／Grafana によるメトリクス分析を通じて、
「L3/L4 だけではなく、L7 レベルでサービス間の通信を制御・監視し、全体を俯瞰するネットワーク運用」
を学ぶ内容としている。

これにより、ネットワーク技術者が、クラウドネイティブ環境で求められる**通信の安全性・信頼性・可視性**を総合的に扱える人材となることを特徴としている。

１－３．カリキュラムの構成

本カリキュラムは、ネットワーク技術者がクラウドネイティブ環境のネットワークを設計・運用できるようになることを目的に、以下のような**複数モジュール構成**としている。いずれも、すでに現場でネットワークに携わる技術者を対象とし、段階的に理解を深める構成としている。

(1) コンテナ・クラウド基礎モジュール

コンテナとは何か、DevOps とコンテナの関係、コンテナ技術とサーバーレスの比較、Docker のアーキテクチャ、コンテナの作成・管理、Docker の基本操作と設計思想、コンテナオーケストレーションの目的、Kubernetes の概要と役割、Pod と Deployment、Service の仕組み、主要クラウド（AWS・GCP・Azure）の特徴、AWS の概要と主要サービス、ECS・Fargate を用いたコンテナアーキテクチャ、AWS 上でのネットワークとアプリケーション構築、Aurora 連携・Auto Scaling・ログ収集などを扱い、
クラウドとコンテナを前提としたネットワーク設計の基礎を固めるモジュールとしている。

(2) Kubernetes・EKS 実践モジュール

Docker の利用、Kubernetes の基本、マニフェストによるシステム構築、Kubernetes ネットワーク通信（ClusterIP・NodePort・LoadBalancer・Ingress など）、ストレージ・ConfigMap・Secret、スケジューリング、リソース管理、可用性管理、パブリッククラウド上のコンテナサービス（EKS や ACI）、EKS 上でのストレージ連携とシークレット管理などを取り扱い、
Kubernetes クラスタ上のネットワーク設計・公開・スケーリングを実践的に学ぶモジュールとしている。

(3) AWS 三層アーキテクチャ・ネットワーク実装モジュール

AWS 上に ALB＋ECS＋RDS で構成された三層アーキテクチャを構築する科目群であり、CloudShell と CDK による IaC 環境準備、VPC・サブネット・セキュリティグループ・VPC エンドポイント・ALB・WAF によるネットワーク構築、ECR・ECS Fargate によるコンテナ基盤、RDS と Secrets Manager による DB/認証情報管理、CodePipeline/CodeBuild/CodeDeploy による CI/CD パイプライン、ALB の HTTPS 化・ACM 証明書による暗号化、テストステージの追加、総合演習などを通じて、
クラウドネットワーク設計とアプリケーション基盤を一体的に構築する力を養うモジュールとしている。

(4) ネットワークセキュリティ・可観測性モジュール（Kubernetes/サービスメッシュ）

コンテナ・クラウドサービスのセキュリティ、4C モデルと 7 つの基本原則、Kubernetes 環境の診断と防御（RBAC・NetworkPolicy・PSS）、Trivy・Cosign・Kyverno を用いたイメージ安全性とランタイム防御、GitHub Actions によるセキュリティゲート付き CI パイプライン、Istio による mTLS 通信と AuthorizationPolicy・Secrets 管理、Kiali・Prometheus・Grafana によるサービスメッシュの可視化と通信分析などを扱い、
クラウドネイティブ環境におけるネットワークの安全性と可観測性を強化するモジュールとしている。

これらのモジュールを通じて、受講者が「オンプレ中心のネットワーク」から「クラウドネイティブなネットワーク基盤」へと視野を拡張し、クラウド・コンテナ・Kubernetes・サービスメッシュを前提としたネットワーク設計に対応できるようにする構成としている。

１－４．産業界ニーズとの整合性

本リカレントプログラムは、クラウドサービスの普及、マイクロサービス化、ゼロトラストセキュリティの浸透といった産業界の変化を踏まえ、ネットワーク技術者に求められる役割の変化に対応することを目的としている。

企業のシステムは、オンプレミスのネットワーク機器だけで完結する構成から、**VPC・ALB・NLB・WAF・ECS・EKS・RDS・CDN・サービスメッシュ**など、多数のクラウドサービスと連携した構成へと移行している。また、マルチクラウドやハイブリッドクラウドも増え、ネットワーク設計は「社内 LAN の設計」から、「インターネット／クラウド／オンプレをまたぐトータルな通信設計」へと広がっている。

こうした状況のなかで、企業は次のようなネットワーク技術者像を求めている。

- クラウドサービスを前提に、ネットワークを論理構成として設計できる技術者
- Kubernetes やサービスメッシュを含むアプリケーション基盤のネットワークを理解し、安定運用できる技術者
- ゼロトラストや DevSecOps の考え方を踏まえ、通信の安全性を担保できる技術者
- ログ・メトリクス・トレースを用いて、ネットワーク障害の原因分析と改善提案ができる技術者

本プログラムは、これらのニーズを踏まえ、実際のクラウド構成例（ALB＋ECS＋RDS、EKS＋EBS/EFS、Istio サービスメッシュなど）を学習対象として取り込み、**現場のアーキテクチャと整合したネットワーク教育**を行うこととしている。

また、IT 人材不足が続くなか、既存のネットワーク技術者が退職や大幅な職種転換を行うことなく、段階的にクラウドネイティブなネットワーク技術へ移行できるようにすることも重要な課題である。本プログラムは、コンテナ・Kubernetes・クラウド・サービスメッシュといった新領域を、「従来ネットワークの延長線上で理解できるように再構成したリカレント教育」として位置づけ、**産業界のニーズに即したネットワーク技術者の育成**に貢献することをめざしている。

１－５．科目概要

■コンテナ技術の教育プログラム開発

コンテナとは

アプリケーション単位で環境を隔離し、ゲスト OS を使用せずに軽量かつ高速に動作する仮想化技術を理解します。インストールの手間を減らし、高い柔軟性とパフォーマンスを実現するコンテナ型仮想化の利点を習得します。

DevOps とは

開発者と運用者が連携し、迅速かつ継続的にサービスを提供する DevOps の考え方を理解します。コンテナは高い可搬性を持ち、開発環境から運用環境への移行を容易にすることで、開発と運用のスムーズな統合を実現します。

コンテナ技術とそれ以外の技術

代表的なコンテナ基盤である Docker の特徴を理解し、マイクロサービスとの関係を把握します。さらに、近年注目されるサーバーレス技術について、管理不要でコスト効率の高い利点と、自由度の制約や長時間処理に不向きといった課題を整理し、開発手法としての特徴を習得します。

Docker とは

コンテナの実行やコンテナイメージの作成・配布を行うためのプラットフォームとして、環境ごとにアプリケーションをパッケージ化し、開発から運用まで同一環境で動作させる仕組みを理解します。代表的なコンテナ基盤として広く活用されています。

コンテナの作成

コンテナを外部から隔離する仕組みである名前空間の役割を理解し、Docker コマンドを用いたコンテナの生成やイメージ作成の手順を習得します。コンテナの状態変化と管理の流れを通じて、実践的な操作スキルを身につけます。

Docker アーキテクチャ

クライアント・サーバー型で動作する Docker エンジンの仕組みを理解し、Docker クライアント、Docker デーモン、コンテナレジストリの 3 つの役割と連携によってコンテナ管理が実現される仕組みを習得します。

Docker を使う上での基本的な考え方

プロセス単位でコンテナを分離する原則を理解し、アプリケーション構成の適切な分割基準を把握します。さらに、ボリュームや名前空間など、Docker を支える基盤技術の仕組みを通じて、効率的なコンテナ運用の基礎を習得します。

Docker の基本的な使い方

コンテナやイメージを操作するための基本コマンドを理解し、ターミナルや PowerShell を用いた実行方法を習得します。マネージメントコマンドやオプション指定の仕組みを通じて、効率的に Docker 環境を操作する基礎を身につけます。

コンテナオーケストレーションとは

コンテナや実行マシンが増えると手動での管理が難しくなるため、負荷分散・死活監視・スケーリングなどの

運用を自動化する仕組みを理解します。オーケストレーションによって、安定的で効率的なコンテナ運用を実現する手法を習得します。

Kubernetes とは

Google が開発し、2014 年にオープンソース化されたコンテナオーケストレーションツールである Kubernetes の仕組みを理解します。複数のコンテナ管理を自動化し、現在では Docker Desktop にも統合されるなど、業界標準として広く利用されています。

ポッドとデプロイメントコントローラ

ポッドはコンテナをまとめて管理する最小単位で、ストレージやネットワークを共有して動作します。デプロイメントコントローラはシステムの状態を監視し、期待する状態との差を自動的に調整することで、安定したアプリケーション運用を実現します。

Kubernetes を使う上での基本的な考え方

サービスはコンテナへのアクセスを提供するエンドポイントとして機能し、外部からの通信を安全かつ効率的に仲介します。さらに、負荷分散・死活監視・スケーリングなどの機能を通じて、高可用性と安定稼働を実現する設計思想を理解します。

Kubernetes の基本的な使い方

クラスタの構成や設定は、YAML や JSON 形式で記述するマニフェストファイルによって宣言的に管理します。ターミナルや PowerShell を用いたコマンド操作を通じて、期待する状態を自動的に反映する Kubernetes の運用手法を習得します。

AWS と GCP と Azure

AWS は世界最大のクラウドサービスで幅広い機能を提供し、GCP は機械学習やデータ分析に強みを持ち、Azure は Microsoft 製品との高い互換性を備えています。それぞれの特徴を理解し、目的に応じたクラウド活用の基礎を習得します。

AWS とは

Amazon が提供する世界最大規模のクラウドサービスであり、IaaS および PaaS 市場で圧倒的なシェアを誇ります。豊富なサービス群と高い可用性を備え、他クラウドに先駆けて提供を開始した先進的なプラットフォームとして活用されています。

AWS の主要なサービス

メール、データベース、サーバーなどの代表的なサービスを理解し、用途に応じた最適な選択を行う力を身につけます。たとえば、メールでは Amazon SES と WorkMail の特徴と利点を比較し、要件に応じたクラウドサービスの活用方法を習得します。

コンテナを利用した AWS アーキテクチャ

運用・セキュリティ・信頼性・パフォーマンス・コスト最適化の各設計観点を理解し、ECS や Fargate を用いた最新のコンテナ運用事例を通じて、効率的で拡張性の高い AWS アーキテクチャ設計を習得します。

AWS の使い方①

コンテナを稼働させるためのネットワーク設計を行い、フロントエンドとバックエンドのアプリケーションをコンテナ上で構築・連携させます。AWS 環境における実践的なシステム設計と構築手法を習得します。

AWS の使い方②

コンテナアプリケーションから Amazon Aurora へ安全に接続する構成を理解し、認証情報を環境変数で管理するセキュリティ対策を習得します。さらに、AWS サービスを活用した効率的で安全なアプリケーション運用手法を身につけます。

AWS の使い方③

ECS の Auto Scaling 機能を用いて負荷に応じた自動スケールアウトを実現し、安定したサービス提供を行う手法を理解します。さらに、ログ収集を通じて運用監視を強化し、信頼性と拡張性の高い AWS 環境構築を習得します。

■コンテナ技術のシステム構築

Docker の利用

具体的には、コンテナ実行・イメージ作成・Dockerfile・Compose・依存関係管理などを通して、開発・運用における Docker 活用の全体像を理解します。

Kubernetes の基本

Kubernetes のアーキテクチャや主要コンポーネントを理解し、minikube や Docker Desktop を用いたクラスタ構築・kubectl による操作演習を通して、コンテナオーケストレーションの基本と実践力を身につけます。

Kubernetes のシステム構築

Pod や Deployment の作成、スケーリング、ローリングアップデート、ロールバック演習を通じて、IaC（Infrastructure as Code）の考え方と継続的デプロイ運用の基礎を習得します。

Kubernetes のネットワーク通信

Pod 内通信やオーバーレイネットワーク、ClusterIP・NodePort・LoadBalancer・ExternalName の動作を理解し、内部・外部通信を安全かつ効率的に設計するネットワーク構築の実践力を習得します。

Kubernetes のストレージとデータ管理

emptyDir や Persistent Volume（PV）・Persistent Volume Claim（PVC）による永続化の方法を理解し、さらに ConfigMap や Secret を用いて設定情報や機密情報を安全に管理する手法を習得します。

Kubernetes のスケジューリング

nodeSelector や nodeAffinity によるノード選択、Pod アフィニティ／アンチアフィニティ、Taint・Toleration による制約設定、topologySpreadConstraints による分散配置を通じて、柔軟で効率的なスケジューリング戦略の設計力を養います。

Kubernetes のリソース管理

Pod レベルの Requests/Limits 設定や QoS クラス、Namespace によるリソース分割、ResourceQuota・LimitRange の制御を理解し、さらに metrics-server と HPA/VPA を活用した自動スケーリングによる効率的なクラスタ運用を習得します。

Kubernetes の可用性管理

Liveness／Readiness／Startup Probe によるヘルスチェック、Deployment・StatefulSet・DaemonSet などのコントローラー活用を通じて、障害に強く自己修復可能なクラスタ設計と高可用性システムの構築手法を習得し

ます。

パブリッククラウド上のコンテナサービス①

Azure の ACI や EKS のクラスタ構築、ロードバランサー連携、Ingress によるアプリ公開などを通じて、クラウド環境でのコンテナ運用と可用性・拡張性の高いアーキテクチャ設計を習得します。

パブリッククラウド上のコンテナサービス②

EBS/EFS による永続ストレージ設定、Secrets Manager や Parameter Store との連携を実践し、セキュアでスケーラブルな EKS 環境を構築・運用する手法を習得します。

■クラウドネイティブのシステム開発入門

クラウドネイティブ基盤実装の概要および環境準備

システムの全体像を理解したうえで、AWS CloudShell 上に CDK 開発環境を整備し、CloudFormation を通じて IaC を実行可能とします。CDK の基本的なコマンド操作を学び、クラウドネイティブ開発の第一歩を踏み出します。

ネットワーク構築

VPC エンドポイントによる通信経路の最適化、ALB・WAF による L7 レイヤのアクセス制御を通じて、セキュアかつ拡張性のあるネットワーク設計を実装を通じて習得します。

コンテナ構築

ECS のタスク定義、サービスを理解し、スケーラブルなアプリケーション基盤を構築し、オートスケーリングなどコンテナオーケストレーションを体感します。

DB 構築

AWS Secrets Manager を活用してデータベース認証情報を安全に管理し、セキュアな環境構築の手法を学習します。

CI/CD パイプライン構築

テストからデプロイまでを自動化し、Blue/Green デプロイによる無停止リリースを実現します。CI/CD の実践を通じて、DevOps の考え方を理解します。

ロードバランサーの HTTPS 移行

ACM で SSL/TLS 証明書を取得・適用し、HTTP から HTTPS へのリダイレクト設定を学習し、安全な通信経路を構築する実践的な運用手順を身につけます。

CI/CD パイプラインのユニットテスト追加

pytest や JUnit レポートを活用し、テスト結果を CodeBuild レポートとして可視化することで、品質保証を組み込んだ継続的デリバリーの設計思想を理解します。

総合演習

これまでに学んだネットワーク・ECS・RDS・CI/CD・セキュリティの知識を活用し、実運用を想定した拡張性と保守性の高いクラウドネイティブ基盤を完成させます。

■コンテナ・クラウドサービスのセキュリティ

コンテナ・クラウドサービスのセキュリティの概要

最小権限や多層防御など 7 つの基本原則を習得し、これらを実践する 5 つの演習の位置づけを把握します。

Kubernetes 環境の診断と防御

過剰権限や制限のない通信、特権実行を含む脆弱環境を構築し、RBAC による権限最小化、NetworkPolicy による通信制御、PSS による特権拒否というセキュリティ対策を適用します。そして、対策前後の挙動を比較し、最小権限・ネットワーク分離・特権制限の重要性を体験的に理解します。

安全なコンテナ構築とランタイム防御

Trivy で脆弱性を診断し、Cosign で署名を付与し、Kyverno で署名検証を自動化します。さらに seccomp や権限削減などの防御を適用し、安全なコンテナ運用方法を体験します。

パイプラインへのセキュリティゲート導入

静的解析や Trivy スキャン、Dependabot 監視を実装し、さらに AWS と連携して安全なイメージビルドの流れを整備します。開発初期から継続的にセキュリティ検査を行う DevSecOps 実践を学びます。

サービス間通信の暗号化とシークレット管理

Kubernetes Secrets で機密情報を安全に管理し、アクセスを必要最小限に制御します。さらに Istio を導入してサイドカーを注入し、mTLS による通信暗号化と AuthorizationPolicy による認可制御を構築します。これらを通じて、ゼロトラストな Kubernetes 環境を体験的に学びます。

Kiali による可視化と通信分析

Istio 上で動作するアプリを対象に、Kiali で通信経路を可視化し、Prometheus でメトリクスを収集し、Grafana で分析します。異常検知や原因特定の流れを体験し、安定運用とセキュリティ強化に役立つ可観測性のスキルを習得します。

1－6．シラバス

科目名	コンテナ技術の教育プログラム開発

科目の概要
本講座では、仮想化・コンテナ・クラウドの基礎からAWSを用いた実践的なシステム構築までを体系的に学びます。 まず、サーバー・ネットワーク仮想化の仕組みと種類を理解し、クラウドサービス（SaaS・PaaS・IaaS）の構成や主要クラウドであるAWS・GCP・Azureの特徴を把握します。次に、Dockerによるコンテナ技術を学び、アプリケーション単位での仮想化やイメージ管理、アーキテクチャ構造、基本コマンド操作を習得します。 さらに、Kubernetesを用いたコンテナオーケストレーションを実践し、ポッドやデプロイメントコントローラによる自動運用、マニフェストによる宣言的管理の考え方を理解します。後半では、AWSを中心にECS・Fargateを活用したコンテナ運用、Aurora連携によるデータ管理、Auto Scalingによる可用性向上などを通じて、クラウドネイティブなアーキテクチャ設計と運用スキルを実践的に身につけます。

授業計画		
回数	項目	内容
第1回	コンテナとは	コンテナの仕組みと特徴を学びます。
第2回	DevOpsとは	DevOpsの概念とその実現を支えるコンテナ技術を学びます。
第3回	コンテナ技術とそれ以外の技術	コンテナ技術と関連技術の違いを学びます。
第4回	Dockerとは	Dockerの仕組みと役割を学びます。
第5回	コンテナの作成	コンテナの構造と作成方法を学びます。
第6回	Dockerアーキテクチャ	Dockerのアーキテクチャと主要コンポーネントを学びます。
第7回	Dockerを使う上での基本的な考え方	Dockerの基本的な設計思想と運用の考え方を学びます。
第8回	Dockerの基本的な使い方	Dockerの基本操作とコマンドの使い方を学びます。
第9回	コンテナオーケストレーションとは	コンテナオーケストレーションの目的と重要性を学びます。
第10回	Kubernetesとは	Kubernetesの概要と役割を学びます。
第11回	ポッドとデプロイメントコントローラ	Kubernetesにおけるポッドとデプロイメントコントローラの役割を学びます。
第12回	Kubernetesを使う上での基本的な考え方	Kubernetesの基本的な考え方とサービスの仕組みを学びます。
第13回	Kubernetesの基本的な使い方	Kubernetesの基本操作とマニフェストの役割を学びます。
第14回	AWSとGCPとAzure	主要なクラウドプロバイダーであるAWS・GCP・Azureの特徴を学びます。
第15回	AWSとは	AWSの概要と特徴を学びます。

第16回	AWSの主要なサービス	AWSの主要サービスと選定の考え方を学びます。
第17回	コンテナを利用したAWSアーキテクチャ	AWSにおけるコンテナ活用型アーキテクチャを学びます。
第18回	AWSの使い方①	AWSを活用したネットワークとアプリケーション構築を学びます。
第19回	AWSの使い方②	AWSを用いたデータベース構築とメール送信の方法を学びます。
第20回	AWSの使い方③	AWSを活用した可用性向上とパフォーマンス設計を学びます。

科目名	コンテナ技術のシステム構築

科目の概要

本講座では、DockerからKubernetes、さらにAWS EKSまでを通じて、クラウドネイティブなシステム構築を総合的に学びます。Dockerによるコンテナ操作やイメージ管理、Composeによる複数コンテナの連携を実践し、開発環境構築の基礎を習得します。
Kubernetesでは、PodやDeploymentによるマニフェスト運用、スケーリングやローリングアップデートを通じて自動復旧や継続的デリバリーの仕組みを理解します。加えて、ServiceやLoadBalancerを用いたネットワーク設計、PersistentVolumeやSecretによるデータの永続化と安全管理、リソース最適化や自動スケーリングなど、運用効率化の技術も習得します。
さらに、ProbeやStatefulSet、DaemonSetを活用し、障害に強い高可用性クラスタを構築します。最終的に、AWS EKS環境でEBS・EFSによるストレージ連携やSecrets Managerとの統合を行い、セキュアで拡張性の高いクラウド基盤構築スキルを身につけます。

授業計画

回数	項目	内容
第1回	Dockerの利用	Dockerの基本操作から複数コンテナの統合管理まで、コンテナ開発の実践的スキルを体系的に学びます。
第2回	Kubernetesの基本	Kubernetesの基礎概念からクラスタ構築・操作までを体系的に学びます。
第3回	Kubernetesのシステム構築	Kubernetesにおけるマニフェスト運用とデプロイメントの仕組みを通して、宣言的なシステム構築の実践を学びます。
第4回	Kubernetesのネットワーク通信	Kubernetesにおけるネットワーク通信の仕組みとServiceリソースの活用を体系的に学びます。
第5回	Kubernetesのストレージとデータ管理	Kubernetesにおけるストレージとデータ管理の仕組みを実践的に学びます。
第6回	Kubernetesのスケジューリング	KubernetesにおけるPodスケジューリングの仕組みと制御手法を体系的に学びます。
第7回	Kubernetesのリソース管理	Kubernetesにおけるリソース管理の仕組みと最適化手法を学びます。
第8回	Kubernetesの可用性管理	Kubernetesにおける可用性管理の仕組みと実践を学びます。
第9回	パブリッククラウド上のコンテナサービス①	AWS EKSを中心にパブリッククラウド上のコンテナサービスを理解し、実践的に学びます。
第10回	パブリッククラウド上のコンテナサービス②	AWS EKS上でのストレージ連携とシークレット管理を中心に、クラウド上のKubernetes運用を深化させることを学びます。

科目名	クラウドネイティブのシステム開発入門

科目の概要
本講座では、AWS上に、ALB＋ECS＋RDSで構成された三層アーキテクチャのWebアプリケーションを構築します。 すべてのリソースをAWS CDK（Infrastructure as Code）で自動化し、開発からデプロイまでの一連の流れを体得し、演習を通じて基盤構築・DevOps・アプリ開発・セキュリティに至るまで、フルスタックなスキルを習得します。 受講後には、自立してクラウド環境を設計・構築・運用できるエンジニアとしての基礎力を身につけることを目指します。

授業計画		
回数	項目	内容
第1回	クラウドネイティブ基盤実装の概要および環境準備	AWS上に顧客情報管理システムを構築します。
第2回	ネットワーク構築	VPC・サブネット・セキュリティグループなど仮想ネットワークを構築し、トラフィック制御の基礎を理解します。
第3回	コンテナ構築	ECRを用いたコンテナイメージ管理を行い、ECS Fargateでサーバレスコンテナを起動します。
第4回	DB構築	アプリケーションデータを永続化するために、RDS MySQLを構築します。
第5回	CI/CDパイプライン構築	GitHubと連携し、CodePipeline・CodeBuild・CodeDeployを組み合わせたCI/CDパイプラインを構築します。
第6回	ロードバランサーのHTTPS移行	ALBをHTTPS化し、通信を暗号化することでセキュアなWebアプリケーションを実現します。
第7回	CI/CDパイプラインのユニットテスト追加	既存のCI/CDパイプラインにテストステージを追加し、品質を自動的に検証する仕組みを実装します。
第8回	総合演習	おみくじ占いアプリケーションを実装し、複数アプリケーションを統合管理できるCDK構成を実践します。

科目名	コンテナ・クラウドサービスのセキュリティ

科目の概要

本講座では、Kubernetesを中心としたクラウドネイティブ環境に必要なセキュリティと可観測性、DevSecOpsの実践を演習を通じて体系的に学びます。
まず、RBAC・NetworkPolicy・PSSを用いて脆弱なクラスターを安全な構成へと改善し、最小権限・通信制御・特権制限の重要性を理解します。続いて、Trivy・Cosign・Kyvernoによるイメージ堅牢化と署名検証の自動化、そしてseccompなどのランタイム防御により、安全なイメージ運用を実現します。
さらに、GitHub Actionsでコード解析や脆弱性スキャンを組み込んだCIパイプラインを構築し、Shift Left型のDevSecOpsを体験します。加えて、Istioを用いてmTLS通信や認可制御、Secrets管理を実装し、ゼロトラストな通信基盤を構築します。最後に、Kiali・Prometheus・Grafanaでサービスメッシュの可観測性を強化し、通信の可視化と異常検知を通じて安定運用に必要な分析スキルを習得します。

授業計画		
回数	項目	内容
第1回	コンテナ・クラウドサービスのセキュリティの概要	クラウドネイティブ環境のコンテナセキュリティを4Cモデルで体系的に理解し、コードからホストまでのライフサイクル全体に必要な対策を学びます。
第2回	Kubernetes環境の診断と防御	RBAC・NetworkPolicy・Pod Security Standardsを用い、脆弱なKubernetes環境を安全な構成へ変換する演習です。
第3回	安全なコンテナ構築とランタイム防御	Trivy・Cosign・Kyvernoを用いてイメージ堅牢化とランタイム防御を行う演習です。
第4回	パイプラインへのセキュリティゲート導入	GitHub Actionsでコード解析と脆弱性スキャンを自動化するCIパイプラインを構築する演習です。
第5回	サービス間通信の暗号化とシークレット管理	サービスメッシュによる通信暗号化とシークレット管理を実践する演習です。
第6回	Kialiによる可視化と通信分析	サービスメッシュの可観測性を実践する演習です。

２．サーバー技術者向けリカレント教育プログラム

２－１．教育指針

本リカレント科目の目的は、サーバー構築・運用・インフラ設計に携わる社会人を対象に、**コンテナ・Kubernetes・クラウドネイティブ・AWS・IaC・セキュリティ・可観測性**といった最新の基盤技術を体系的にアップデートする機会を提供することとしている。従来型の物理サーバーや単純な仮想サーバー運用だけでは、クラウドネイティブ時代のシステム要件に応えることが難しくなっており、コンテナ基盤やマネージドサービス、インフラ自動化を前提としたサーバー技術者への転換が求められている。

本プログラムでは、まず Docker によるコンテナ技術と、その運用を支える Kubernetes の基礎から学び、ポッドやデプロイメント、サービスといった概念を通じて、**宣言的な基盤構築と自動復旧・スケーリングの仕組み**を理解することをめざしている。そのうえで、AWSECS／Fargate、Aurora、Auto Scaling などのマネージドサービスを活用したクラウドネイティブなアーキテクチャ設計に進み、三層構造の Web アプリケーション基盤を AWS 上で実装することで、実務に直結するサーバー基盤構築スキルの獲得を図っている。

さらに、クラウドネイティブの基盤を安定運用するために不可欠な、**CI/CD による自動デプロイ、CloudFormation／CDK／Terraform による IaC、ログ・メトリクス・トレースを用いた可観測性、RBAC・NetworkPolicy・ゼロトラスト・DevSecOps によるセキュリティ対策**を一連の流れとして取り扱う構成としている。サーバー技術者が、構築・運用・セキュリティ・監視を一体として理解し、クラウドネイティブ環境全体を設計・改善できる人材へとステップアップすることを目標に据えている。

本プログラムは、社会人の学び直し（リカレント教育）として、オンデマンドの知識習得とハンズオン演習を組み合わせ、短期間で実務に直結する形でスキルを更新できる教育体系と位置づけている。

２－２．教育内容の特徴

(1) コンテナとクラウドネイティブ基盤を軸とした教育内容としている

本プログラムでは、サーバー技術者がこれからの基盤技術の中心となる**コンテナ／Kubernetes／クラウドネイティブ**を軸に学習を進める構成としている。Docker の基本的な仕組みとイメージ管理、コンテナアーキテクチャを起点として、コンテナオーケストレーションとしての Kubernetes を段階的に扱い、ポッド・デプロイメントコントローラ・サービス・ストレージ・スケジューリング・リソース管理・可用性設計といった要素を体系的に学ぶことにより、「コンテナを単に動かす」のではなく、**クラスタ全体を設計・運用するサーバー技術者**をめざす内容としている。

(2) AWS を中心としたクラウド基盤構築を具体的に扱っている

サーバー技術者に求められる実務力として、クラウドサービスの理解と活用力が不可欠であることから、AWS を中心としたクラウド基盤構築を具体的に取り扱う教育内容としている。ECS／Fargate によるコンテナ実行基盤、Aurora を用いたデータベース連携、ALB・Auto Scaling による可用性・スケーラビリティ確保、RDS と Secrets Manager による安全な認証情報管理など、シラバスで示されている具体的な構成要素を通じて、**「実際に利用される三層アーキテクチャを自ら設計・構築できること」**を狙いとしている。

また、CloudShell や AWS CDK を用いた IaC によって、VPC・サブネット・セキュリティグループ・ALB・ECS・RDS・CI/CD パイプラインといったリソースをコードで定義し、自動構築する演習を含めることで、手作業の設定に依存しない再現性の高いインフラ運用の基礎を身につける教育内容としている。

(3) クラウドネイティブの概念と DevOps／CI/CD／IaC を一体のものとして扱っている

クラウドネイティブ環境では、サーバー構築とアプリケーション開発が密接に結びつくことから、本プログラムでは**クラウドネイティブの概念、アジャイル・DevOps、CI/CD、IaC** を分断せず、ひとつの流れとして学ぶ構成としている。クラウドネイティブの基本概念と構成要素を整理したうえで、マイクロサービス、サーバーレスアーキテクチャ、API Gateway、サービスメッシュといった要素を位置付け、Git／GitHub によるバージョン管理、Jenkins や GitHub Actions、CodePipeline などを利用した自動ビルド・テスト・デプロイの仕組みを取り入れている。

これにより、サーバー技術者が「構築した基盤上でアプリケーションがどのように継続的にデリバリーされるのか」を理解し、開発チームとの連携を前提とした **DevOps 型のサーバー技術者**を育成することをねらいとしている。

(4) 可観測性とセキュリティを、基盤運用の必須要素として明示的に扱っている

現代のサーバー運用では、単にサービスを動作させるだけでなく、**状態を把握し、異常を検知し、継続的に改善する可観測性**と、**ライフサイクル全体にわたるセキュリティ対策**が重要となる。本プログラムでは、ログ・

メトリクス・トレースを中心とした可観測性の概念を整理し、Prometheus・Grafana・ELK などを用いた監視・可視化の方法、Istio＋Kiali によるサービスメッシュ上での通信経路の可視化と分析を扱う構成としている。

セキュリティ面では、コンテナセキュリティの 4C モデルを起点に、RBAC・NetworkPolicy・Pod Security Standards による権限・通信・特権の制御、Trivy・Cosign・Kyverno によるイメージスキャン・署名・ポリシー適用、GitHub Actions を用いたシフトレフト型の DevSecOps パイプライン、Istio を用いた mTLS・認可制御・シークレット管理など、演習を通じて**攻めと守りを統合したサーバー基盤運用**を学ぶ内容としている。

(5) 段階的な演習構成により、学習内容を即実務に活かせるようにしている

教育内容全体は、**コンテナ／Kubernetes の基礎 → クラウドネイティブ概論 → Kubernetes／EKS の実践 → AWS 上の三層アーキテクチャ構築 → セキュリティと可観測性の強化**という段階構成としており、各科目が互いに連続した学びの流れを形成するように設計している。

序盤では概念と基本操作を中心とした内容とし、中盤では Kubernetes や EKS 上でのクラスタ構築・運用、終盤では AWS CDK による実システムの実装やセキュリティ・可観測性の演習を通じて、学んだ技術を現場のシナリオに近い形で統合できるようにしている。これにより、受講者が講座修了後すぐに、自組織のクラウド基盤構築や運用改善に取り組めるようにすることを特徴としている。

２－３．カリキュラムの構成

本リカレントプログラムのカリキュラムは、**複数の科目を組み合わせたモジュール型**として構成しており、サーバー技術者がクラウドネイティブ基盤に必要な知識と技能を段階的に習得できるようにしている。全体を大きく次の 5 つのモジュールに分け、それぞれが役割を分担しながらも相互に連携するよう設計している。

1. **コンテナ・Kubernetes・AWS 基礎モジュール**
   Docker によるコンテナ技術、コンテナオーケストレーションの基本、Kubernetes のポッド・デプロイメント・サービス、主要クラウド（AWS・GCP・Azure）の特徴、ECS・Fargate・Aurora・Auto Scaling などを扱い、コンテナ基盤と AWS を中心としたクラウドアーキテクチャの基礎を固めるモジュールとしている。
2. **クラウドネイティブ概論・DevOps モジュール**
   クラウドネイティブの概念、マイクロサービス、サーバーレス、アジャイルと DevOps、CI/CD、IaC、可観測性、クラウドネイティブセキュリティなどを扱い、クラウドネイティブ環境における開発・運用・基盤の関係性を整理するモジュールとしている。
3. **Kubernetes・EKS 実践モジュール**
   Docker から Kubernetes への展開、マニフェストによる宣言的管理、ネットワーク設計、ストレージとデータ管理、スケジューリングとリソース管理、可用性設計、EKS 上でのクラスタ構築・ストレージ連携・シークレット管理などを通じて、クラウド上での Kubernetes 運用スキルを獲得するモジュールとしている。
4. **AWS 三層アーキテクチャ実装モジュール**
   AWS CDK による IaC を用いて、VPC・ALB・ECS Fargate・RDS・CI/CD パイプラインなどから構成される三層アーキテクチャの Web アプリケーション基盤を実装する総合演習モジュールとしている。ネットワーク構築、コンテナ基盤構築、DB 構築、パイプライン構築、HTTPS 化、テスト自動化などを通じて、実務に近い一連のプロセスを体験する構成としている。
5. **Kubernetes セキュリティ・可観測性モジュール**
   RBAC・NetworkPolicy・Pod Security Standards、Trivy・Cosign・Kyverno、GitHub Actions によるセキュリティゲート、Istio による mTLS・認可・シークレット管理、Kiali・Prometheus・Grafana による可観測性強化などを扱い、クラウドネイティブ環境の安全性と信頼性を確保するための実践的スキルを育成するモジュールとしている。

これらのモジュールを通じて、受講者が**基礎から応用・実装・セキュリティ・運用改善まで、一貫したストーリーとして学べるカリキュラム構成**としている。

２－４．産業界ニーズとの整合性

本リカレントプログラムは、DX 白書などに示される産業界の動向や、クラウド事業者・システムインテグレータ・ユーザー企業のニーズを踏まえ、サーバー技術者に求められる役割の変化に対応することを目的としている。オンプレミス中心のインフラから、クラウド／コンテナ／クラウドネイティブ基盤へ移行が進むなか、企業は**AWS や Kubernetes を前提としたシステム基盤を安全かつ効率的に構築・運用できるサーバー技術者**を求めている。

産業界では、ALB＋ECS＋RDS といった三層アーキテクチャや、EKS／Fargate を活用したコンテナ基盤、CloudFormation／CDK／Terraform を用いた IaC、CI/CD パイプラインによる自動デプロイ、RBAC やゼロトラストを前提としたセキュアなクラスタ設計などが標準的な手法になりつつある。本プログラムは、こうした構成要素をそのまま教育内容に取り込み、現場のアーキテクチャや運用プロセスと乖離しない形でカリキュラムを編成している。

また、システムの複雑化に伴い、単にサーバーを構築・監視するのではなく、**マイクロサービス、サービスメッシュ、可観測性、DevSecOps** といった考え方を踏まえて基盤全体を継続的に改善できる人材が求められている。本プログラムでは、Kubernetes セキュリティ、イメージスキャンや署名検証、サービスメッシュ上での暗号化通信と認可制御、Kiali／Prometheus／Grafana による可視化・分析など、最新の運用・セキュリティ実務に直結する内容を組み込むことで、産業界のニーズとの一体化を図っている。

さらに、IT 人材不足が課題となるなか、現職のサーバー技術者が離職することなく、最新の技術体系へ軟着陸できる学び直しの場の整備は、企業・教育機関双方にとって重要なテーマとなっている。本プログラムは、既に基礎的なサーバー運用スキルを持つ社会人を対象とし、短期集中かつ段階的な学習構成とすることで、**現場から離れずにスキル転換を図れる実践的なリカレント教育**として位置づけている。

２－５．科目概要

■コンテナ技術の教育プログラム開発

コンテナとは

アプリケーション単位で環境を隔離し、ゲスト OS を使用せずに軽量かつ高速に動作する仮想化技術を理解します。インストールの手間を減らし、高い柔軟性とパフォーマンスを実現するコンテナ型仮想化の利点を習得します。

DevOps とは

開発者と運用者が連携し、迅速かつ継続的にサービスを提供する DevOps の考え方を理解します。コンテナは高い可搬性を持ち、開発環境から運用環境への移行を容易にすることで、開発と運用のスムーズな統合を実現します。

コンテナ技術とそれ以外の技術

代表的なコンテナ基盤である Docker の特徴を理解し、マイクロサービスとの関係を把握します。さらに、近年注目されるサーバーレス技術について、管理不要でコスト効率の高い利点と、自由度の制約や長時間処理に不向きといった課題を整理し、開発手法としての特徴を習得します。

Docker とは

コンテナの実行やコンテナイメージの作成・配布を行うためのプラットフォームとして、環境ごとにアプリケーションをパッケージ化し、開発から運用まで同一環境で動作させる仕組みを理解します。代表的なコンテナ基盤として広く活用されています。

コンテナの作成

コンテナを外部から隔離する仕組みである名前空間の役割を理解し、Docker コマンドを用いたコンテナの生成やイメージ作成の手順を習得します。コンテナの状態変化と管理の流れを通じて、実践的な操作スキルを身につけます。

Docker アーキテクチャ

クライアント・サーバー型で動作する Docker エンジンの仕組みを理解し、Docker クライアント、Docker デーモン、コンテナレジストリの 3 つの役割と連携によってコンテナ管理が実現される仕組みを習得します。

Docker を使う上での基本的な考え方

プロセス単位でコンテナを分離する原則を理解し、アプリケーション構成の適切な分割基準を把握します。さらに、ボリュームや名前空間など、Docker を支える基盤技術の仕組みを通じて、効率的なコンテナ運用の基礎を習得します。

Docker の基本的な使い方

コンテナやイメージを操作するための基本コマンドを理解し、ターミナルや PowerShell を用いた実行方法を習得します。マネージメントコマンドやオプション指定の仕組みを通じて、効率的に Docker 環境を操作する基礎を身につけます。

コンテナオーケストレーションとは

コンテナや実行マシンが増えると手動での管理が難しくなるため、負荷分散・死活監視・スケーリングなどの運用を自動化する仕組みを理解します。オーケストレーションによって、安定的で効率的なコンテナ運用を実現する手法を習得します。

Kubernetes とは

Google が開発し、2014 年にオープンソース化されたコンテナオーケストレーションツールである Kubernetes の仕組みを理解します。複数のコンテナ管理を自動化し、現在では Docker Desktop にも統合されるなど、業界標準として広く利用されています。

ポッドとデプロイメントコントローラ

ポッドはコンテナをまとめて管理する最小単位で、ストレージやネットワークを共有して動作します。デプロイメントコントローラはシステムの状態を監視し、期待する状態との差を自動的に調整することで、安定したアプリケーション運用を実現します。

Kubernetes を使う上での基本的な考え方

サービスはコンテナへのアクセスを提供するエンドポイントとして機能し、外部からの通信を安全かつ効率的に仲介します。さらに、負荷分散・死活監視・スケーリングなどの機能を通じて、高可用性と安定稼働を実現する設計思想を理解します。

Kubernetes の基本的な使い方

クラスタの構成や設定は、YAML や JSON 形式で記述するマニフェストファイルによって宣言的に管理します。ターミナルや PowerShell を用いたコマンド操作を通じて、期待する状態を自動的に反映する Kubernetes の運用手法を習得します。

AWS と GCP と Azure

AWS は世界最大のクラウドサービスで幅広い機能を提供し、GCP は機械学習やデータ分析に強みを持ち、Azure は Microsoft 製品との高い互換性を備えています。それぞれの特徴を理解し、目的に応じたクラウド活用の基礎を習得します。

AWS とは

Amazon が提供する世界最大規模のクラウドサービスであり、IaaS および PaaS 市場で圧倒的なシェアを誇ります。豊富なサービス群と高い可用性を備え、他クラウドに先駆けて提供を開始した先進的なプラットフォームとして活用されています。

AWS の主要なサービス

メール、データベース、サーバーなどの代表的なサービスを理解し、用途に応じた最適な選択を行う力を身につけます。たとえば、メールでは AmazonSES と WorkMail の特徴と利点を比較し、要件に応じたクラウドサービスの活用方法を習得します。

コンテナを利用した AWS アーキテクチャ

運用・セキュリティ・信頼性・パフォーマンス・コスト最適化の各設計観点を理解し、ECS や Fargate を用いた最新のコンテナ運用事例を通じて、効率的で拡張性の高い AWS アーキテクチャ設計を習得します。

AWS の使い方①

コンテナを稼働させるためのネットワーク設計を行い、フロントエンドとバックエンドのアプリケーションを

コンテナ上で構築・連携させます。AWS 環境における実践的なシステム設計と構築手法を習得します。

AWS の使い方②

コンテナアプリケーションから Amazon Aurora へ安全に接続する構成を理解し、認証情報を環境変数で管理するセキュリティ対策を習得します。さらに、AWS サービスを活用した効率的で安全なアプリケーション運用手法を身につけます。

AWS の使い方③

ECS の Auto Scaling 機能を用いて負荷に応じた自動スケールアウトを実現し、安定したサービス提供を行う手法を理解します。さらに、ログ収集を通じて運用監視を強化し、信頼性と拡張性の高い AWS 環境構築を習得します。

■クラウドネイティブ概論

クラウドネイティブの基本

クラウドコンピューティング、マイクロサービス、DevOps、CI/CD、IaC などの要素を通じて、柔軟で拡張性・可用性の高いシステム設計思想を理解し、最新の開発・運用手法を習得します。

クラウド技術

サービスモデル（IaaS/PaaS/SaaS）やデプロイモデル（パブリック／プライベート／ハイブリッド）を整理し、AWS の構成要素・リージョン・AZ・マネージドサービスの仕組みを通して、クラウド利用の全体像を把握します。

コンテナ技術

仮想化との違いや軽量性、Immutable Infrastructure の概念、コンテナランタイム・イメージ・レジストリの構造を理解し、Docker Desktop での実践を通じて、Kubernetes との関係を含むコンテナ運用の基礎を習得します。

サーバーレスアーキテクチャ

AWS Lambda を中心に、イベント駆動型処理やトリガー設定、API 連携などを実践的に解説し、インフラ管理を不要とする柔軟で拡張性の高いアプリケーション設計手法を習得します。

マイクロサービス

モノリシック構造との違いや疎結合の利点を理解し、通信方式（REST/gRPC・メッセージングなど）やサービスメッシュ、API Gateway の役割を把握します。さらに、結果整合性や Saga パターンなど、分散システム特有の課題と解決策を学びます。

アジャイルと DevOps

アジャイルの反復型開発やスクラム・かんばんの手法を通じて柔軟な開発を実践し、DevOps・SRE・DevSecOps などの考え方を組み合わせて、開発と運用を統合し継続的改善を実現するアプローチを習得します。

CICD

Git・GitHub によるバージョン管理から、Jenkins・GitHub Actions・AWS CodePipeline などを活用した自動ビルド／テスト／デプロイの仕組みを理解し、DevOps 実現に向けた効率的で高品質な開発運用プロセスを習得します。

IaC(Infrastructure as Code)

CloudFormation や Terraform、Ansible などの特徴と利点を比較し、宣言型・手続型の違いや GitOps との連携を通じて、再現性・効率性・一貫性の高いインフラ運用を実現する手法を習得します。

可観測性

ログ・メトリクス・トレースの 3 要素を中心に、Prometheus・Grafana・ELK による監視と可視化、ログ分析、アラート運用を通じて、システムの信頼性とパフォーマンスを高める手法を習得します。

クラウドネイティブにおけるセキュリティ

ゼロトラストモデル、シフトレフト、DevSecOps の重要性を理解し、IAM・暗号化・ネットワーク防御・ログ監視などを通じて、セキュアなクラウド運用と継続的なリスク管理を実現する手法を習得します。

■コンテナ技術のシステム構築

Docker の利用

具体的には、コンテナ実行・イメージ作成・Dockerfile・Compose・依存関係管理などを通して、開発・運用における Docker 活用の全体像を理解します。

Kubernetes の基本

Kubernetes のアーキテクチャや主要コンポーネントを理解し、minikube や Docker Desktop を用いたクラスタ構築・kubectl による操作演習を通して、コンテナオーケストレーションの基本と実践力を身につけます。

Kubernetes のシステム構築

Pod や Deployment の作成、スケーリング、ローリングアップデート、ロールバック演習を通じて、IaC（Infrastructure as Code）の考え方と継続的デプロイ運用の基礎を習得します。

Kubernetes のネットワーク通信

Pod 内通信やオーバーレイネットワーク、ClusterIP・NodePort・LoadBalancer・ExternalName の動作を理解し、内部・外部通信を安全かつ効率的に設計するネットワーク構築の実践力を習得します。

Kubernetes のストレージとデータ管理

emptyDir や Persistent Volume (PV)・Persistent Volume Claim (PVC) による永続化の方法を理解し、さらに ConfigMap や Secret を用いて設定情報や機密情報を安全に管理する手法を習得します。

Kubernetes のスケジューリング

nodeSelector や nodeAffinity によるノード選択、Pod アフィニティ／アンチアフィニティ、Taint・Toleration による制約設定、topologySpreadConstraints による分散配置を通じて、柔軟で効率的なスケジューリング戦略の設計力を養います。

Kubernetes のリソース管理

Pod レベルの Requests/Limits 設定や QoS クラス、Namespace によるリソース分割、ResourceQuota・LimitRange の制御を理解し、さらに metrics-server と HPA/VPA を活用した自動スケーリングによる効率的なクラスタ運用を習得します。

Kubernetes の可用性管理

Liveness／Readiness／Startup Probe によるヘルスチェック、Deployment・StatefulSet・DaemonSet などのコントローラー活用を通じて、障害に強く自己修復可能なクラスタ設計と高可用性システムの構築手法を習得します。

パブリッククラウド上のコンテナサービス①

Azure の ACI や EKS のクラスタ構築、ロードバランサー連携、Ingress によるアプリ公開などを通じて、クラウド環境でのコンテナ運用と可用性・拡張性の高いアーキテクチャ設計を習得します。

パブリッククラウド上のコンテナサービス②

EBS/EFS による永続ストレージ設定、Secrets Manager や Parameter Store との連携を実践し、セキュアでスケーラブルな EKS 環境を構築・運用する手法を習得します。

■クラウドネイティブのシステム開発入門

クラウドネイティブ基盤実装の概要および環境準備

システムの全体像を理解したうえで、AWS CloudShell 上に CDK 開発環境を整備し、CloudFormation を通じて IaC を実行可能とします。CDK の基本的なコマンド操作を学び、クラウドネイティブ開発の第一歩を踏み出します。

ネットワーク構築

VPC エンドポイントによる通信経路の最適化、ALB・WAF による L7 レイヤのアクセス制御を通じて、セキュアかつ拡張性のあるネットワーク設計を実装を通じて習得します。

コンテナ構築

ECS のタスク定義、サービスを理解し、スケーラブルなアプリケーション基盤を構築し、オートスケーリングなどコンテナオーケストレーションを体感します。

DB 構築

AWS Secrets Manager を活用してデータベース認証情報を安全に管理し、セキュアな環境構築の手法を学習します。

CI/CD パイプライン構築

テストからデプロイまでを自動化し、Blue/Green デプロイによる無停止リリースを実現します。CI/CD の実践を通じて、DevOps の考え方を理解します。

ロードバランサーの HTTPS 移行

ACM で SSL/TLS 証明書を取得・適用し、HTTP から HTTPS へのリダイレクト設定を学習し、安全な通信経路を構築する実践的な運用手順を身につけます。

CI/CD パイプラインのユニットテスト追加

pytest や JUnit レポートを活用し、テスト結果を CodeBuild レポートとして可視化することで、品質保証を組み込んだ継続的デリバリーの設計思想を理解します。

総合演習

これまでに学んだネットワーク・ECS・RDS・CI/CD・セキュリティの知識を活用し、実運用を想定した拡張性と保守性の高いクラウドネイティブ基盤を完成させます。

■コンテナ・クラウドサービスのセキュリティ

コンテナ・クラウドサービスのセキュリティの概要

最小権限や多層防御など 7 つの基本原則を習得し、これらを実践する 5 つの演習の位置づけを把握します。

Kubernetes 環境の診断と防御

過剰権限や制限のない通信、特権実行を含む脆弱環境を構築し、RBAC による権限最小化、NetworkPolicy による通信制御、PSS による特権拒否というセキュリティ対策を適用します。そして、対策前後の挙動を比較し、最小権限・ネットワーク分離・特権制限の重要性を体験的に理解します。

安全なコンテナ構築とランタイム防御

Trivy で脆弱性を診断し、Cosign で署名を付与し、Kyverno で署名検証を自動化します。さらに seccomp や権限削減などの防御を適用し、安全なコンテナ運用方法を体験します。

パイプラインへのセキュリティゲート導入

静的解析や Trivy スキャン、Dependabot 監視を実装し、さらに AWS と連携して安全なイメージビルドの流れを整備します。開発初期から継続的にセキュリティ検査を行う DevSecOps 実践を学びます。

サービス間通信の暗号化とシークレット管理

Kubernetes Secrets で機密情報を安全に管理し、アクセスを必要最小限に制御します。さらに Istio を導入してサイドカーを注入し、mTLS による通信暗号化と AuthorizationPolicy による認可制御を構築します。これらを通じて、ゼロトラストな Kubernetes 環境を体験的に学びます。

Kiali による可視化と通信分析

Istio 上で動作するアプリを対象に、Kiali で通信経路を可視化し、Prometheus でメトリクスを収集し、Grafana で分析します。異常検知や原因特定の流れを体験し、安定運用とセキュリティ強化に役立つ可観測性のスキルを習得します。

2−6．シラバス

科目名	コンテナ技術の教育プログラム開発

科目の概要
本講座では、仮想化・コンテナ・クラウドの基礎からAWSを用いた実践的なシステム構築までを体系的に学びます。 まず、サーバー・ネットワーク仮想化の仕組みと種類を理解し、クラウドサービス（SaaS・PaaS・IaaS）の構成や主要クラウドであるAWS・GCP・Azureの特徴を把握します。次に、Dockerによるコンテナ技術を学び、アプリケーション単位での仮想化やイメージ管理、アーキテクチャ構造、基本コマンド操作を習得します。 さらに、Kubernetesを用いたコンテナオーケストレーションを実践し、ポッドやデプロイメントコントローラによる自動運用、マニフェストによる宣言的管理の考え方を理解します。後半では、AWSを中心にECS・Fargateを活用したコンテナ運用、Aurora連携によるデータ管理、Auto Scalingによる可用性向上などを通じて、クラウドネイティブなアーキテクチャ設計と運用スキルを実践的に身につけます。

授業計画		
回数	項目	内容
第1回	コンテナとは	コンテナの仕組みと特徴を学びます。
第2回	DevOpsとは	DevOpsの概念とその実現を支えるコンテナ技術を学びます。
第3回	コンテナ技術とそれ以外の技術	コンテナ技術と関連技術の違いを学びます。
第4回	Dockerとは	Dockerの仕組みと役割を学びます。
第5回	コンテナの作成	コンテナの構造と作成方法を学びます。
第6回	Dockerアーキテクチャ	Dockerのアーキテクチャと主要コンポーネントを学びます。
第7回	Dockerを使う上での基本的な考え方	Dockerの基本的な設計思想と運用の考え方を学びます。
第8回	Dockerの基本的な使い方	Dockerの基本操作とコマンドの使い方を学びます。
第9回	コンテナオーケストレーションとは	コンテナオーケストレーションの目的と重要性を学びます。
第10回	Kubernetesとは	Kubernetesの概要と役割を学びます。
第11回	ポッドとデプロイメントコントローラ	Kubernetesにおけるポッドとデプロイメントコントローラの役割を学びます。
第12回	Kubernetesを使う上での基本的な考え方	Kubernetesの基本的な考え方とサービスの仕組みを学びます。
第13回	Kubernetesの基本的な使い方	Kubernetesの基本操作とマニフェストの役割を学びます。
第14回	AWSとGCPとAzure	主要なクラウドプロバイダーであるAWS・GCP・Azureの特徴を学びます。
第15回	AWSとは	AWSの概要と特徴を学びます。

第16回	AWSの主要なサービス	AWSの主要サービスと選定の考え方を学びます。
第17回	コンテナを利用したAWSアーキテクチャ	AWSにおけるコンテナ活用型アーキテクチャを学びます。
第18回	AWSの使い方①	AWSを活用したネットワークとアプリケーション構築を学びます。
第19回	AWSの使い方②	AWSを用いたデータベース構築とメール送信の方法を学びます。
第20回	AWSの使い方③	AWSを活用した可用性向上とパフォーマンス設計を学びます。

科目名	クラウドネイティブ概論

科目の概要

本講座では、クラウドネイティブの基礎から実践的な開発・運用までを体系的に学びます。まず、クラウドネイティブの概念を理解し、クラウドサービスの仕組みやAWSを中心とした主要技術を把握します。次に、Dockerによるコンテナ技術、Kubernetesによるオーケストレーション、サーバーレスやマイクロサービスによる分散型アプリケーション設計を学習します。さらに、アジャイルとDevOpsの手法を通じて継続的改善と迅速な開発を実現し、CI/CDによる自動化パイプラインで効率的なデプロイと品質向上を体験します。IaC（Infrastructure as Code）では、CloudFormationやTerraformを用いたインフラ自動化を実践し、可観測性の章では、ログ・メトリクス・トレースを活用したシステム監視と分析手法を習得します。最後に、ゼロトラストやDevSecOpsなどの考え方を通じて、クラウドネイティブ環境におけるセキュリティ対策を体系的に学び、安全で持続可能なクラウド運用を実現する力を身につけます。

授業計画

回数	項目	内容
第1回	クラウドネイティブの基本	クラウドネイティブの基本概念と構成要素を体系的に学びます。
第2回	クラウド技術	クラウドコンピューティングの仕組みと主要サービスを理解し、AWSを中心にクラウド技術の基礎を学びます。
第3回	コンテナ技術	コンテナ技術の仕組みとDockerを中心とした基本操作を体系的に学びます。
第4回	サーバーレスアーキテクチャ	サーバーレスアーキテクチャの概念と仕組みを理解し、FaaSやCaaSの活用を学びます。
第5回	マイクロサービス	マイクロサービスアーキテクチャの概念と設計手法を体系的に学びます。
第6回	アジャイルとDevOps	アジャイルとDevOpsの理念・手法を理解し、クラウドネイティブを支える開発文化を学びます。
第7回	CICD	CI/CD（継続的インテグレーション／デリバリー）の概念と実装を学びます。
第8回	IaC (Infrastructure as Code)	IaC（Infrastructure as Code）の概念と主要ツールを理解し、インフラ自動化の基礎を学びます。
第9回	可観測性	クラウドネイティブ環境における可観測性（Observability）の概念と実践を学びます。
第10回	クラウドネイティブにおけるセキュリティ	クラウドネイティブ環境におけるセキュリティの考え方と実践的対策を学びます。

科目名	コンテナ技術のシステム構築

科目の概要

本講座では、DockerからKubernetes、さらにAWS EKSまでを通じて、クラウドネイティブなシステム構築を総合的に学びます。Dockerによるコンテナ操作やイメージ管理、Composeによる複数コンテナの連携を実践し、開発環境構築の基礎を習得します。
Kubernetesでは、PodやDeploymentによるマニフェスト運用、スケーリングやローリングアップデートを通じて自動復旧や継続的デリバリーの仕組みを理解します。加えて、ServiceやLoadBalancerを用いたネットワーク設計、PersistentVolumeやSecretによるデータの永続化と安全管理、リソース最適化や自動スケーリングなど、運用効率化の技術も習得します。
さらに、ProbeやStatefulSet、DaemonSetを活用し、障害に強い高可用性クラスタを構築します。最終的に、AWS EKS環境でEBS・EFSによるストレージ連携やSecrets Managerとの統合を行い、セキュアで拡張性の高いクラウド基盤構築スキルを身につけます。

授業計画

回数	項目	内容
第1回	Dockerの利用	Dockerの基本操作から複数コンテナの統合管理まで、コンテナ開発の実践的スキルを体系的に学びます。
第2回	Kubernetesの基本	Kubernetesの基礎概念からクラスタ構築・操作までを体系的に学びます。
第3回	Kubernetesのシステム構築	Kubernetesにおけるマニフェスト運用とデプロイメントの仕組みを通して、宣言的なシステム構築の実践を学びます。
第4回	Kubernetesのネットワーク通信	Kubernetesにおけるネットワーク通信の仕組みとServiceリソースの活用を体系的に学びます。
第5回	Kubernetesのストレージとデータ管理	Kubernetesにおけるストレージとデータ管理の仕組みを実践的に学びます。
第6回	Kubernetesのスケジューリング	KubernetesにおけるPodスケジューリングの仕組みと制御手法を体系的に学びます。
第7回	Kubernetesのリソース管理	Kubernetesにおけるリソース管理の仕組みと最適化手法を学びます。
第8回	Kubernetesの可用性管理	Kubernetesにおける可用性管理の仕組みと実践を学びます。
第9回	パブリッククラウド上のコンテナサービス①	AWS EKSを中心にパブリッククラウド上のコンテナサービスを理解し、実践的に学びます。
第10回	パブリッククラウド上のコンテナサービス②	AWS EKS上でのストレージ連携とシークレット管理を中心に、クラウド上のKubernetes運用を深化させることを学びます。

科目名	クラウドネイティブのシステム開発入門

科目の概要
本講座では、AWS上に、ALB＋ECS＋RDSで構成された三層アーキテクチャのWebアプリケーションを構築します。 すべてのリソースをAWS CDK（Infrastructure as Code）で自動化し、開発からデプロイまでの一連の流れを体得し、演習を通じて基盤構築・DevOps・アプリ開発・セキュリティに至るまで、フルスタックなスキルを習得します。 受講後には、自立してクラウド環境を設計・構築・運用できるエンジニアとしての基礎力を身につけることを目指します。

授業計画		
回数	項目	内容
第1回	クラウドネイティブ基盤実装の概要および環境準備	AWS上に顧客情報管理システムを構築します。
第2回	ネットワーク構築	VPC・サブネット・セキュリティグループなど仮想ネットワークを構築し、トラフィック制御の基礎を理解します。
第3回	コンテナ構築	ECRを用いたコンテナイメージ管理を行い、ECS Fargateでサーバレスコンテナを起動します。
第4回	DB構築	アプリケーションデータを永続化するために、RDS MySQLを構築します。
第5回	CI/CDパイプライン構築	GitHubと連携し、CodePipeline・CodeBuild・CodeDeployを組み合わせたCI/CDパイプラインを構築します。
第6回	ロードバランサーのHTTPS移行	ALBをHTTPS化し、通信を暗号化することでセキュアなWebアプリケーションを実現します。
第7回	CI/CDパイプラインのユニットテスト追加	既存のCI/CDパイプラインにテストステージを追加し、品質を自動的に検証する仕組みを実装します。
第8回	総合演習	おみくじ占いアプリケーションを実装し、複数アプリケーションを統合管理できるCDK構成を実践します。

科目名	コンテナ・クラウドサービスのセキュリティ

科目の概要
本講座では、Kubernetesを中心としたクラウドネイティブ環境に必要なセキュリティと可観測性、DevSecOpsの実践を演習を通じて体系的に学びます。 まず、RBAC・NetworkPolicy・PSSを用いて脆弱なクラスターを安全な構成へと改善し、最小権限・通信制御・特権制限の重要性を理解します。続いて、Trivy・Cosign・Kyvernoによるイメージ堅牢化と署名検証の自動化、そしてseccompなどのランタイム防御により、安全なイメージ運用を実現します。 さらに、GitHub Actionsでコード解析や脆弱性スキャンを組み込んだCIパイプラインを構築し、Shift Left型のDevSecOpsを体験します。加えて、Istioを用いてmTLS通信や認可制御、Secrets管理を実装し、ゼロトラストな通信基盤を構築します。最後に、Kiali・Prometheus・Grafanaでサービスメッシュの可観測性を強化し、通信の可視化と異常検知を通じて安定運用に必要な分析スキルを習得します。

授業計画		
回数	項目	内容
第1回	コンテナ・クラウドサービスのセキュリティの概要	クラウドネイティブ環境のコンテナセキュリティを4Cモデルで体系的に理解し、コードからホストまでのライフサイクル全体に必要な対策を学びます。
第2回	Kubernetes環境の診断と防御	RBAC・NetworkPolicy・Pod Security Standardsを用い、脆弱なKubernetes環境を安全な構成へ変換する演習です。
第3回	安全なコンテナ構築とランタイム防御	Trivy・Cosign・Kyvernoを用いてイメージ堅牢化とランタイム防御を行う演習です。
第4回	パイプラインへのセキュリティゲート導入	GitHub Actionsでコード解析と脆弱性スキャンを自動化するCIパイプラインを構築する演習です。
第5回	サービス間通信の暗号化とシークレット管理	サービスメッシュによる通信暗号化とシークレット管理を実践する演習です。
第6回	Kialiによる可視化と通信分析	サービスメッシュの可観測性を実践する演習です。

３．アプリケーション技術者向けリカレント教育プログラム

３―１．教育指針

本リカレント科目の目的は、すでに業務でアプリケーション開発・運用に携わる社会人に対して、**仮想化・コンテナ・クラウド・クラウドネイティブ**を前提とした最新のアプリケーション基盤技術を学び直す機会を提供し、コードを書く技術者から「サービス全体を設計・説明できるアプリケーション技術者」へとステップアップすることに置いている。

近年のシステム開発は、オンプレミス環境でのモノリシックなアプリケーションから、クラウド上で動作するマイクロサービスやサーバーレス、コンテナ基盤上の分散アプリケーションへと大きく変化している。アプリケーション技術者には、言語やフレームワークだけでなく、**仮想化された実行環境、クラウドサービス、ネットワーク・ストレージ、セキュリティ、CI/CD** といった周辺要素を含めて理解する力が求められている。

本プログラムでは、最初に仮想化技術とクラウドの基礎を整理し、サーバー・ネットワーク・ストレージ・アプリケーションといったリソースがどのように仮想化され、クラウドサービスとして提供されているかを理解するところから始める。そのうえで、コンテナ技術（Docker）・コンテナオーケストレーション（Kubernetes）・クラウド基盤（AWS／GCP／Azure）を、アプリケーションの実行環境という視点から体系的に学ぶ構成としている。

また、クラウドネイティブの科目群では、マイクロサービス・サーバーレス・アジャイル・DevOps・CI/CD・IaC・可観測性・セキュリティといった要素を組み合わせ、**クラウド上でアプリケーションを継続的に改善・運用するための一連のプロセス**を学ぶことを重視している。さらに、ALB＋ECS＋RDS による三層アーキテクチャ構築や、EKS・サービスメッシュ・DevSecOps といった実践的な演習を通じて、アプリケーション技術者が基盤を含めて設計・説明できる人材となることを目標としている。

本科目は、社会人の学び直し（リカレント教育）として、既存の開発経験や言語スキルを前提にしつつ、クラウドネイティブ時代に必要なアプリケーション基盤技術を短期間で効率的にアップデートする教育プログラムとして位置づけている。

３－２．教育内容の特徴

(1)「アプリケーション視点」で仮想化・クラウド・基盤を学び直す構成としている

本プログラムは、仮想化・コンテナ・クラウドといったインフラ寄りの内容を、**あくまでアプリケーション技術者の視点から**再構成している点に特徴を置いている。

仮想化の科目では、ホストOS型・ハイパーバイザ型・コンテナ型の仕組みを、

- どのようなアプリケーションを
- どの環境に載せると
- どのような柔軟性・性能・運用性が得られるか

という観点で整理し、クラウド仮想化サービス（EC2／GCE／Azure VM）やネットワーク仮想化（VPC／Virtual Network）を、**アプリケーションの動作環境を選ぶための選択肢**として理解できるようにする構成としている。

(2) コンテナ／Kubernetesを「アプリケーションの実行プラットフォーム」として扱っている

コンテナ・Kubernetesの科目群では、コンテナを単なる技術要素として学ぶのではなく、

- アプリケーションをどのように分割し
- どの単位でコンテナ化し
- どうデプロイし、どのようにスケールさせるか

というアプリケーション設計と直結したテーマとして扱う内容としている。

Dockerのアーキテクチャや基本コマンド、マニフェストによるKubernetesの構成管理、Service／Ingressによる公開、スケーリング・ローリングアップデート・ヘルスチェック・自動復旧などを通じて、**「アプリケーションを継続的に提供するプラットフォーム」としてのコンテナ／Kubernetes**を理解できるようにすることを特徴としている。

(3) クラウドネイティブ開発プロセスを一連の流れとして学ぶ構成としている

クラウドネイティブの科目群では、アプリケーションのライフサイクル全体を見据え、

- マイクロサービス／サーバーレスによるアーキテクチャ
- アジャイル・DevOpsによる開発プロセス
- CI/CDによるビルド・テスト・デプロイの自動化
- IaCによるインフラ構築のコード化
- 可観測性による運用・改善サイクル
- ゼロトラスト・DevSecOpsによるセキュリティ組み込み

を連続したテーマとして扱う構成としている。

これにより、アプリケーション技術者が「コードを書いて終わり」ではなく、**設計・実装・テスト・デプロイ・運用・改善・安全性**までを含めたクラウドネイティブな開発運用プロセスを俯瞰し、自身の業務にどの部分から取り込むかを考えられるようにすることをねらいとしている。

(4) 具体的なクラウド構成を題材に、実務そのままの演習を取り入れている

ALB＋ECS＋RDS の三層アーキテクチャ構築科目では、顧客情報管理システムなどの具体的な題材を用いながら、

- CDK による IaC
- VPC・サブネット・セキュリティグループ・ALB・WAF によるネットワーク設計
- ECS Fargate によるアプリケーションコンテナ実行
- RDS MySQL によるデータ永続化と Secrets Manager による認証情報管理
- CI/CD パイプライン構築と Blue/Green デプロイ
- HTTPS 化と証明書管理

といった要素を組み合わせて、一連のシステムとして構築する演習を行う構成としている。

これにより、アプリケーション技術者が**「自分の書いたアプリケーションを、クラウド上のどのような構成で提供するのが適切か」を説明しながら構築できる力**を身につけることを重視している。

(5) DevSecOps・サービスメッシュ・可観測性を通じて「安全に動かし続ける力」を育成している

Kubernetes セキュリティ・サービスメッシュ・可観測性の科目群では、アプリケーション技術者が

- RBAC／NetworkPolicy／PSS による Kubernetes の安全な構成
- コンテナイメージの脆弱性診断と署名検証
- CI パイプラインにおけるセキュリティゲートの組み込み
- Istio によるサービス間通信の暗号化・認可制御
- Kiali・Prometheus・Grafana による通信経路の可視化と異常検知

といったテーマを通じて、**「アプリケーションを安全に動かし続けるための設計と運用」**を具体的な手順として学べる構成としている。

これにより、アプリケーション技術者が、コードレベルの品質だけでなく、サービスレベルの信頼性・セキュリティ・運用性までを含めて設計できる人材となることを目標としている。

３－３．カリキュラムの構成

本カリキュラムは、アプリケーション技術者が「コード＋基盤」の両面を理解しながら開発・運用できるようになることをねらいとして、次のような段階的構成としている。

(1) 仮想化・クラウド基礎モジュール

仮想化技術の基礎と種類、仮想化とクラウドの関係、サーバー仮想化（ホスト OS 型・ハイパーバイザ型・コンテナ型）、主要クラウド（AWS EC2／GCP GCE／Azure Virtual Machines）におけるサーバー仮想化、ネットワーク仮想化（VPN・SDN・NFV・SD-WAN）、クラウドのネットワーク仮想化（VPC／Virtual Network など）を扱う構成としている。

これにより、アプリケーション技術者が、アプリケーションが動作する**下位レイヤ（サーバー・ネットワーク・ストレージ）の仮想化構造とクラウド上の位置づけ**を整理し、後続のコンテナ・クラウドネイティブ学習の基盤を固めることを目的としている。

(2) コンテナ・クラウド基盤モジュール

コンテナとは何か、DevOps とコンテナの関係、コンテナ技術とサーバーレスの違い、Docker の仕組みとアーキテクチャ、コンテナの作成と管理、Docker の設計思想と基本操作、コンテナオーケストレーションの目的、Kubernetes の概要と役割、Pod と Deployment、Service の仕組み、主要クラウド（AWS／GCP／Azure）の特徴、AWS の概要と主要サービス、ECS／Fargate を用いたコンテナアーキテクチャ、AWS 上でのアプリケーション構築・Aurora 連携・Auto Scaling などを扱い、
「コンテナ＋クラウド上でアプリケーションを動かす」ための実行基盤を学ぶモジュールとしている。

(3) クラウドネイティブ開発・運用モジュール

クラウドネイティブの基本概念、クラウド技術の整理、コンテナ技術の再確認、サーバーレスアーキテクチャ、マイクロサービス、アジャイル・DevOps、CI/CD、IaC（CloudFormation・Terraform 等）、可観測性（ログ・メトリクス・トレース）、クラウドネイティブにおけるセキュリティ（ゼロトラスト・DevSecOps）を扱う構成としている。

このモジュールでは、アプリケーション技術者が**開発プロセス・運用プロセスを含めた「クラウドネイティブなソフトウェアライフサイクル」**を理解し、継続的デリバリーやインフラ自動化、監視・分析といった実務で必要となる要素を体系的に整理できるようにすることを目的としている。

(4) Kubernetes／EKS 実践モジュール

Docker の利用、Kubernetes の基本とクラスタ構築、マニフェストによる宣言的構成管理、Kubernetes ネットワーク通信、ストレージとデータ管理、スケジューリングとリソース管理、可用性管理、パブリッククラウド上のコンテナサービス（EKS など）、EBS/EFS と Secrets Manager 連携を通じて、

Kubernetes／EKS を基盤としたクラウドネイティブアプリケーションの運用を実践的に学ぶモジュールとしている。

(5) AWS 三層アーキテクチャ・アプリ実装モジュール

AWS 上に ALB＋ECS＋RDS で構成された三層アーキテクチャの Web アプリケーションを構築する科目群であり、CloudShell＋CDK による IaC 環境準備、VPC・サブネット・セキュリティグループ・ALB・WAF によるネットワーク構築、ECR＋ECS Fargate によるアプリケーション実行基盤、RDS MySQL と Secrets Manager によるデータ永続化と認証情報管理、CodePipeline／CodeBuild／CodeDeploy による CI/CD パイプライン、ALB の HTTPS 化、テストステージ追加、総合演習などで構成している。

このモジュールでは、アプリケーション技術者が**１つの業務システムをクラウド上で設計・実装・デプロイ・運用する一連の流れ**を体験し、開発と基盤構築を一体として理解できるようにすることをめざしている。

(6) DevSecOps・サービスメッシュ・可観測性モジュール

コンテナ／クラウドサービスのセキュリティの全体像、Kubernetes 環境の診断と防御（RBAC・NetworkPolicy・PSS）、安全なコンテナ構築とランタイム防御（Trivy・Cosign・Kyverno・seccomp 等）、GitHub Actions によるセキュリティゲート付き CI パイプライン、Istio を用いたサービス間通信の暗号化と認可制御、Secrets 管理、Kiali・Prometheus・Grafana によるサービスメッシュの可視化と通信分析などを扱う構成としている。

これにより、アプリケーション技術者が**セキュリティと可観測性を前提にアプリケーションを設計・運用する DevSecOps 型の実務**を学ぶことを目的としている。

３－４．産業界ニーズとの整合性

本リカレントプログラムは、DX 推進・クラウドシフト・マイクロサービス化・DevOps／DevSecOps の普及といった産業界の変化を踏まえ、アプリケーション技術者に求められるスキルセットの変容に対応することを目的としている。

企業の開発現場では、オンプレミス環境での単一システム開発から、クラウド上で動作する複数サービスの組み合わせによるシステム構築へと移行している。したがって、アプリケーション技術者は、

- どのクラウドにどのような構成でアプリケーションを配置するか
- コンテナ・サーバーレス・マイクロサービスをどう使い分けるか
- CI/CD・IaC を利用して、どのように素早く安全にリリースするか
- 可観測性・セキュリティをどのレベルで組み込むか

といった観点を踏まえて開発することが求められている。

こうした状況の中で、企業は次のようなアプリケーション技術者像を求めている。

- クラウド上で完結するアプリケーションアーキテクチャを自ら設計できる技術者
- コンテナ・Kubernetes・クラウドサービスを理解し、システム全体像を意識して開発できる技術者
- DevOps／DevSecOps の考え方を理解し、CI/CD や IaC を活用した開発プロセスに参加できる技術者
- ログ・メトリクス・トレースを用いて、システムの状態を説明し、改善提案ができる技術者

本プログラムは、これらのニーズを踏まえ、仮想化基盤・コンテナ・クラウド・クラウドネイティブ・Kubernetes／EKS・ALB＋ECS＋RDS 構成・サービスメッシュ・DevSecOps といったテーマを、**実際の企業システム構成をモデルとする演習**として取り込む構成としている。

また、IT 人材不足や技術変化の加速により、既存のアプリケーション技術者が継続的にスキルを更新しなければならない状況が続いている一方で、企業内研修だけでは最新技術を十分にカバーしきれないケースも多い。本プログラムは、社会人を対象としたリカレント教育として、既存の開発スキルを土台に、クラウドネイティブ時代に必要な知識と実践力を**短期間・段階的に補完する場**として位置づけている。

このように、本リカレント教育プログラムは、産業界が求めるクラウドネイティブ時代のアプリケーション技術者像と整合した内容として設計しており、現場で即戦力として活躍できる人材育成に寄与することをめざしている。

３－５．科目概要

■仮想化技術の教育プログラム開発

仮想化技術の基礎とその種別

サーバー、ネットワーク、ストレージ、アプリケーションなどの仮想化の特徴を理解し、リソースの効率化や可用性向上といったメリット、専門知識の必要性などの課題を踏まえ、仮想化がもたらす柔軟な IT 基盤の仕組みを習得します。

仮想化とクラウド

サーバー、ネットワーク、ストレージ、アプリケーションなどの仮想化の特徴を理解し、リソースの効率化や可用性向上といったメリット、専門知識の必要性などの課題を踏まえ、仮想化がもたらす柔軟な IT 基盤の仕組みを習得します。

サーバー仮想化とは

物理サーバー上に複数の仮想サーバーを構築し、異なる OS を同時に稼働させる技術を理解します。ホスト OS 型・ハイパーバイザ型・コンテナ型の特徴を比較し、柔軟で効率的なサーバー運用の基礎を習得します。

クラウドのサーバー仮想化

AWS の Amazon EC2、GCP の GCE、Azure の Virtual Machines を比較し、それぞれの特徴や利用シーンを理解します。代表的クラウドを通じて、仮想サーバーの柔軟な運用と拡張性の仕組みを習得します。

サーバー仮想化の種類：ホスト OS 型

物理サーバー上のホスト OS で複数の仮想マシンを動作させ、異なる OS を同時に実行できる柔軟な環境を構築します。最も手軽に試せる仮想化方式であり、代表的なソフトウェアとして VirtualBox があります。

サーバー仮想化の種類：ハイパーバイザ型

物理サーバー上で直接ハイパーバイザを実行し、複数の仮想マシンを効率的に管理・制御します。各仮想マシンは独立したゲスト OS を持ち、異なる環境を同時に稼働できるため、高性能で柔軟なサーバー運用を実現します。

サーバー仮想化の種類：コンテナ型

アプリケーション単位で環境を隔離し、軽量かつ高速に動作する仮想化技術を理解します。ゲスト OS を必要とせず、コンテナ作成のみで環境を構築できる柔軟性が特長であり、代表的なツールとして Docker があります。

ネットワーク仮想化とは①

物理的なネットワーク機器や構成を仮想化し、1 つのネットワークを分割したり統合したりできる柔軟な設計を実現します。ソフトウェアによる設定変更が可能となり、物理構成に依存しない効率的なネットワーク運用を習得します。

ネットワーク仮想化とは②

VPN・SDN・OpenFlow・NFV・SD-WAN など複数の技術やソフトウェアの組み合わせによって実現される仕組みを理解し、柔軟で効率的なネットワーク運用を可能にする仮想化技術の全体像を習得します。

クラウドのネットワーク仮想化

AWS の AmazonVPC、GCP の VPC、Azure の Virtual Network を比較し、それぞれの特徴や役割を理解します。クラウド環境における柔軟で安全なネットワーク構築の仕組みを習得します。

■コンテナ技術の教育プログラム開発

コンテナとは

アプリケーション単位で環境を隔離し、ゲスト OS を使用せずに軽量かつ高速に動作する仮想化技術を理解します。インストールの手間を減らし、高い柔軟性とパフォーマンスを実現するコンテナ型仮想化の利点を習得します。

DevOps とは

開発者と運用者が連携し、迅速かつ継続的にサービスを提供する DevOps の考え方を理解します。コンテナは高い可搬性を持ち、開発環境から運用環境への移行を容易にすることで、開発と運用のスムーズな統合を実現します。

コンテナ技術とそれ以外の技術

代表的なコンテナ基盤である Docker の特徴を理解し、マイクロサービスとの関係を把握します。さらに、近年注目されるサーバーレス技術について、管理不要でコスト効率の高い利点と、自由度の制約や長時間処理に不向きといった課題を整理し、開発手法としての特徴を習得します。

Docker とは

コンテナの実行やコンテナイメージの作成・配布を行うためのプラットフォームとして、環境ごとにアプリケーションをパッケージ化し、開発から運用まで同一環境で動作させる仕組みを理解します。代表的なコンテナ基盤として広く活用されています。

コンテナの作成

コンテナを外部から隔離する仕組みである名前空間の役割を理解し、Docker コマンドを用いたコンテナの生成やイメージ作成の手順を習得します。コンテナの状態変化と管理の流れを通じて、実践的な操作スキルを身につけます。

Docker アーキテクチャ

クライアント・サーバー型で動作する Docker エンジンの仕組みを理解し、Docker クライアント、Docker デーモン、コンテナレジストリの 3 つの役割と連携によってコンテナ管理が実現される仕組みを習得します。

Docker を使う上での基本的な考え方

プロセス単位でコンテナを分離する原則を理解し、アプリケーション構成の適切な分割基準を把握します。さらに、ボリュームや名前空間など、Docker を支える基盤技術の仕組みを通じて、効率的なコンテナ運用の基礎を習得します。

Docker の基本的な使い方

コンテナやイメージを操作するための基本コマンドを理解し、ターミナルや PowerShell を用いた実行方法を習得します。マネージメントコマンドやオプション指定の仕組みを通じて、効率的に Docker 環境を操作する基礎を身につけます。

コンテナオーケストレーションとは

コンテナや実行マシンが増えると手動での管理が難しくなるため、負荷分散・死活監視・スケーリングなどの運用を自動化する仕組みを理解します。オーケストレーションによって、安定的で効率的なコンテナ運用を実現する手法を習得します。

Kubernetes とは

Google が開発し、2014 年にオープンソース化されたコンテナオーケストレーションツールである Kubernetes の仕組みを理解します。複数のコンテナ管理を自動化し、現在では Docker Desktop にも統合されるなど、業界標準として広く利用されています。

ポッドとデプロイメントコントローラ

ポッドはコンテナをまとめて管理する最小単位で、ストレージやネットワークを共有して動作します。デプロイメントコントローラはシステムの状態を監視し、期待する状態との差を自動的に調整することで、安定したアプリケーション運用を実現します。

Kubernetes を使う上での基本的な考え方

サービスはコンテナへのアクセスを提供するエンドポイントとして機能し、外部からの通信を安全かつ効率的に仲介します。さらに、負荷分散・死活監視・スケーリングなどの機能を通じて、高可用性と安定稼働を実現する設計思想を理解します。

Kubernetes の基本的な使い方

クラスタの構成や設定は、YAML や JSON 形式で記述するマニフェストファイルによって宣言的に管理します。ターミナルや PowerShell を用いたコマンド操作を通じて、期待する状態を自動的に反映する Kubernetes の運用手法を習得します。

AWS と GCP と Azure

AWS は世界最大のクラウドサービスで幅広い機能を提供し、GCP は機械学習やデータ分析に強みを持ち、Azure は Microsoft 製品との高い互換性を備えています。それぞれの特徴を理解し、目的に応じたクラウド活用の基礎を習得します。

AWS とは

Amazon が提供する世界最大規模のクラウドサービスであり、IaaS および PaaS 市場で圧倒的なシェアを誇ります。豊富なサービス群と高い可用性を備え、他クラウドに先駆けて提供を開始した先進的なプラットフォームとして活用されています。

AWS の主要なサービス

メール、データベース、サーバーなどの代表的なサービスを理解し、用途に応じた最適な選択を行う力を身につけます。たとえば、メールでは Amazon SES と WorkMail の特徴と利点を比較し、要件に応じたクラウドサービスの活用方法を習得します。

コンテナを利用した AWS アーキテクチャ

運用・セキュリティ・信頼性・パフォーマンス・コスト最適化の各設計観点を理解し、ECS や Fargate を用いた最新のコンテナ運用事例を通じて、効率的で拡張性の高い AWS アーキテクチャ設計を習得します。

AWS の使い方①

コンテナを稼働させるためのネットワーク設計を行い、フロントエンドとバックエンドのアプリケーションをコンテナ上で構築・連携させます。AWS 環境における実践的なシステム設計と構築手法を習得します。

AWS の使い方②

コンテナアプリケーションから Amazon Aurora へ安全に接続する構成を理解し、認証情報を環境変数で管理す

るセキュリティ対策を習得します。さらに、AWS サービスを活用した効率的で安全なアプリケーション運用手法を身につけます。

AWS の使い方③

ECS の Auto Scaling 機能を用いて負荷に応じた自動スケールアウトを実現し、安定したサービス提供を行う手法を理解します。さらに、ログ収集を通じて運用監視を強化し、信頼性と拡張性の高い AWS 環境構築を習得します。

■クラウドネイティブ概論

クラウドネイティブの基本

クラウドコンピューティング、マイクロサービス、DevOps、CI/CD、IaC などの要素を通じて、柔軟で拡張性・可用性の高いシステム設計思想を理解し、最新の開発・運用手法を習得します。

クラウド技術

サービスモデル（IaaS/PaaS/SaaS）やデプロイモデル（パブリック／プライベート／ハイブリッド）を整理し、AWS の構成要素・リージョン・AZ・マネージドサービスの仕組みを通して、クラウド利用の全体像を把握します。

コンテナ技術

仮想化との違いや軽量性、Immutable Infrastructure の概念、コンテナランタイム・イメージ・レジストリの構造を理解し、Docker Desktop での実践を通じて、Kubernetes との関係を含むコンテナ運用の基礎を習得します。

サーバーレスアーキテクチャ

AWS Lambda を中心に、イベント駆動型処理やトリガー設定、API 連携などを実践的に解説し、インフラ管理を不要とする柔軟で拡張性の高いアプリケーション設計手法を習得します。

マイクロサービス

モノリシック構造との違いや疎結合の利点を理解し、通信方式（REST/gRPC・メッセージングなど）やサービスメッシュ、API Gateway の役割を把握します。さらに、結果整合性や Saga パターンなど、分散システム特有の課題と解決策を学びます。

アジャイルと DevOps

アジャイルの反復型開発やスクラム・かんばんの手法を通じて柔軟な開発を実践し、DevOps・SRE・DevSecOps などの考え方を組み合わせて、開発と運用を統合し継続的改善を実現するアプローチを習得します。

CICD

Git・GitHub によるバージョン管理から、Jenkins・GitHub Actions・AWS CodePipeline などを活用した自動ビルド／テスト／デプロイの仕組みを理解し、DevOps 実現に向けた効率的で高品質な開発運用プロセスを習得します。

IaC（Infrastructure as Code）

CloudFormation や Terraform、Ansible などの特徴と利点を比較し、宣言型・手続型の違いや GitOps との連携

を通じて、再現性・効率性・一貫性の高いインフラ運用を実現する手法を習得します。

可観測性

ログ・メトリクス・トレースの 3 要素を中心に、Prometheus・Grafana・ELK による監視と可視化、ログ分析、アラート運用を通じて、システムの信頼性とパフォーマンスを高める手法を習得します。

クラウドネイティブにおけるセキュリティ

ゼロトラストモデル、シフトレフト、DevSecOps の重要性を理解し、IAM・暗号化・ネットワーク防御・ログ監視などを通じて、セキュアなクラウド運用と継続的なリスク管理を実現する手法を習得します。

■コンテナ技術のシステム構築

Docker の利用

具体的には、コンテナ実行・イメージ作成・Dockerfile・Compose・依存関係管理などを通して、開発・運用における Docker 活用の全体像を理解します。

Kubernetes の基本

Kubernetes のアーキテクチャや主要コンポーネントを理解し、minikube や Docker Desktop を用いたクラスタ構築・kubectl による操作演習を通して、コンテナオーケストレーションの基本と実践力を身につけます。

Kubernetes のシステム構築

Pod や Deployment の作成、スケーリング、ローリングアップデート、ロールバック演習を通じて、IaC（Infrastructure as Code）の考え方と継続的デプロイ運用の基礎を習得します。

Kubernetes のネットワーク通信

Pod 内通信やオーバーレイネットワーク、ClusterIP・NodePort・LoadBalancer・ExternalName の動作を理解し、内部・外部通信を安全かつ効率的に設計するネットワーク構築の実践力を習得します。

Kubernetes のストレージとデータ管理

emptyDir や Persistent Volume（PV）・Persistent Volume Claim（PVC）による永続化の方法を理解し、さらに ConfigMap や Secret を用いて設定情報や機密情報を安全に管理する手法を習得します。

Kubernetes のスケジューリング

nodeSelector や nodeAffinity によるノード選択、Pod アフィニティ／アンチアフィニティ、Taint・Toleration による制約設定、topologySpreadConstraints による分散配置を通じて、柔軟で効率的なスケジューリング戦略の設計力を養います。

Kubernetes のリソース管理

Pod レベルの Requests/Limits 設定や QoS クラス、Namespace によるリソース分割、ResourceQuota・LimitRange の制御を理解し、さらに metrics-server と HPA/VPA を活用した自動スケーリングによる効率的なクラスタ運用を習得します。

Kubernetes の可用性管理

Liveness／Readiness／Startup Probe によるヘルスチェック、Deployment・StatefulSet・DaemonSet などのコントローラー活用を通じて、障害に強く自己修復可能なクラスタ設計と高可用性システムの構築手法を習得します。

パブリッククラウド上のコンテナサービス①

Azure の ACI や EKS のクラスタ構築、ロードバランサー連携、Ingress によるアプリ公開などを通じて、クラウ

ド環境でのコンテナ運用と可用性・拡張性の高いアーキテクチャ設計を習得します。

パブリッククラウド上のコンテナサービス②

EBS/EFS による永続ストレージ設定、Secrets Manager や Parameter Store との連携を実践し、セキュアでスケーラブルな EKS 環境を構築・運用する手法を習得します。

■クラウドネイティブのシステム開発入門

クラウドネイティブ基盤実装の概要および環境準備

システムの全体像を理解したうえで、AWS CloudShell 上に CDK 開発環境を整備し、CloudFormation を通じて IaC を実行可能とします。CDK の基本的なコマンド操作を学び、クラウドネイティブ開発の第一歩を踏み出します。

ネットワーク構築

VPC エンドポイントによる通信経路の最適化、ALB・WAF による L7 レイヤのアクセス制御を通じて、セキュアかつ拡張性のあるネットワーク設計を実装を通じて習得します。

コンテナ構築

ECS のタスク定義、サービスを理解し、スケーラブルなアプリケーション基盤を構築し、オートスケーリングなどコンテナオーケストレーションを体感します。

DB 構築

AWS Secrets Manager を活用してデータベース認証情報を安全に管理し、セキュアな環境構築の手法を学習します。

CI/CD パイプライン構築

テストからデプロイまでを自動化し、Blue/Green デプロイによる無停止リリースを実現します。CI/CD の実践を通じて、DevOps の考え方を理解します。

ロードバランサーの HTTPS 移行

ACM で SSL/TLS 証明書を取得・適用し、HTTP から HTTPS へのリダイレクト設定を学習し、安全な通信経路を構築する実践的な運用手順を身につけます。

CI/CD パイプラインのユニットテスト追加

pytest や JUnit レポートを活用し、テスト結果を CodeBuild レポートとして可視化することで、品質保証を組み込んだ継続的デリバリーの設計思想を理解します。

総合演習

これまでに学んだネットワーク・ECS・RDS・CI/CD・セキュリティの知識を活用し、実運用を想定した拡張性と保守性の高いクラウドネイティブ基盤を完成させます。

■コンテナ・クラウドサービスのセキュリティ

コンテナ・クラウドサービスのセキュリティの概要

最小権限や多層防御など 7 つの基本原則を習得し、これらを実践する 5 つの演習の位置づけを把握します。

Kubernetes 環境の診断と防御

過剰権限や制限のない通信、特権実行を含む脆弱環境を構築し、RBAC による権限最小化、NetworkPolicy によ

る通信制御、PSS による特権拒否というセキュリティ対策を適用します。そして、対策前後の挙動を比較し、最小権限・ネットワーク分離・特権制限の重要性を体験的に理解します。

安全なコンテナ構築とランタイム防御

Trivy で脆弱性を診断し、Cosign で署名を付与し、Kyverno で署名検証を自動化します。さらに seccomp や権限削減などの防御を適用し、安全なコンテナ運用方法を体験します。

パイプラインへのセキュリティゲート導入

静的解析や Trivy スキャン、Dependabot 監視を実装し、さらに AWS と連携して安全なイメージビルドの流れを整備します。開発初期から継続的にセキュリティ検査を行う DevSecOps 実践を学びます。

サービス間通信の暗号化とシークレット管理

Kubernetes Secrets で機密情報を安全に管理し、アクセスを必要最小限に制御します。さらに Istio を導入してサイドカーを注入し、mTLS による通信暗号化と AuthorizationPolicy による認可制御を構築します。これらを通じて、ゼロトラストな Kubernetes 環境を体験的に学びます。

Kiali による可視化と通信分析

Istio 上で動作するアプリを対象に、Kiali で通信経路を可視化し、Prometheus でメトリクスを収集し、Grafana で分析します。異常検知や原因特定の流れを体験し、安定運用とセキュリティ強化に役立つ可観測性のスキルを習得します。

3－6．シラバス

科目名	仮想化技術の教育プログラム開発

科目の概要
本講座では、仮想化技術とクラウドの基礎から応用までを体系的に学びます。 まず、サーバー・ネットワーク・ストレージ・アプリケーションなどの仮想化の概念を理解し、リソースの効率化や可用性向上といった仮想化の利点を把握します。サーバー仮想化では、ホストOS型・ハイパーバイザ型・コンテナ型の仕組みと特徴を学び、VirtualBox・VMware・Dockerなど代表的なツールを通じて、環境構築と運用の違いを実践的に理解します。さらに、AWSのEC2、GCPのGCE、AzureのVirtual Machinesといった主要クラウドの仮想化サービスを比較し、クラウド上での柔軟なリソース運用を体験します。 ネットワーク仮想化では、VPN・SDN・NFV・SD-WANなどの技術を組み合わせて構築される仕組みを学び、物理構成に依存しない柔軟なネットワーク設計を習得します。 最後に、AWS・GCP・Azureが提供するVPCやVirtual Networkを通じて、クラウドにおけるセキュアで拡張性の高いネットワーク運用の基礎を身につけます。

授業計画		
回数	項目	内容
第1回	仮想化技術の基礎とその種別	仮想化技術の基礎と主な種類を学びます。
第2回	仮想化とクラウド	仮想化技術を基盤とするクラウドの仕組みと種類を学びます
第3回	サーバー仮想化とは	サーバー仮想化の仕組みと種類を学びます。
第4回	クラウドのサーバー仮想化	主要クラウドが提供するサーバー仮想化サービスを学びます。
第5回	サーバー仮想化の種類：ホストOS型	ホストOS型仮想化の仕組みと特徴を学びます。
第6回	サーバー仮想化の種類：ハイパーバイザ型	ハイパーバイザ型仮想化の仕組みと特徴を学びます。
第7回	サーバー仮想化の種類：コンテナ型	コンテナ型仮想化の仕組みと特徴を学びます。
第8回	ネットワーク仮想化とは①	ネットワーク仮想化の仕組みと利点を学びます。
第9回	ネットワーク仮想化とは②	ネットワーク仮想化を支える主要技術を学びます。
第10回	クラウドのネットワーク仮想化	主要クラウドが提供するネットワーク仮想化サービスを学びます。

科目名	コンテナ技術の教育プログラム開発

科目の概要

本講座では、仮想化・コンテナ・クラウドの基礎からAWSを用いた実践的なシステム構築までを体系的に学びます。
まず、サーバー・ネットワーク仮想化の仕組みと種類を理解し、クラウドサービス（SaaS・PaaS・IaaS）の構成や主要クラウドであるAWS・GCP・Azureの特徴を把握します。次に、Dockerによるコンテナ技術を学び、アプリケーション単位での仮想化やイメージ管理、アーキテクチャ構造、基本コマンド操作を習得します。
さらに、Kubernetesを用いたコンテナオーケストレーションを実践し、ポッドやデプロイメントコントローラによる自動運用、マニフェストによる宣言的管理の考え方を理解します。後半では、AWSを中心にECS・Fargateを活用したコンテナ運用、Aurora連携によるデータ管理、Auto Scalingによる可用性向上などを通じて、クラウドネイティブなアーキテクチャ設計と運用スキルを実践的に身につけます。

授業計画

回数	項目	内容
第1回	コンテナとは	コンテナの仕組みと特徴を学びます。
第2回	DevOpsとは	DevOpsの概念とその実現を支えるコンテナ技術を学びます。
第3回	コンテナ技術とそれ以外の技術	コンテナ技術と関連技術の違いを学びます。
第4回	Dockerとは	Dockerの仕組みと役割を学びます。
第5回	コンテナの作成	コンテナの構造と作成方法を学びます。
第6回	Dockerアーキテクチャ	Dockerのアーキテクチャと主要コンポーネントを学びます。
第7回	Dockerを使う上での基本的な考え方	Dockerの基本的な設計思想と運用の考え方を学びます。
第8回	Dockerの基本的な使い方	Dockerの基本操作とコマンドの使い方を学びます。
第9回	コンテナオーケストレーションとは	コンテナオーケストレーションの目的と重要性を学びます。
第10回	Kubernetesとは	Kubernetesの概要と役割を学びます。
第11回	ポッドとデプロイメントコントローラ	Kubernetesにおけるポッドとデプロイメントコントローラの役割を学びます。
第12回	Kubernetesを使う上での基本的な考え方	Kubernetesの基本的な考え方とサービスの仕組みを学びます。
第13回	Kubernetesの基本的な使い方	Kubernetesの基本操作とマニフェストの役割を学びます。
第14回	AWSとGCPとAzure	主要なクラウドプロバイダーであるAWS・GCP・Azureの特徴を学びます。
第15回	AWSとは	AWSの概要と特徴を学びます。

第16回	AWSの主要なサービス	AWSの主要サービスと選定の考え方を学びます。
第17回	コンテナを利用したAWSアーキテクチャ	AWSにおけるコンテナ活用型アーキテクチャを学びます。
第18回	AWSの使い方①	AWSを活用したネットワークとアプリケーション構築を学びます。
第19回	AWSの使い方②	AWSを用いたデータベース構築とメール送信の方法を学びます。
第20回	AWSの使い方③	AWSを活用した可用性向上とパフォーマンス設計を学びます。

科目名	クラウドネイティブ概論

科目の概要

本講座では、クラウドネイティブの基礎から実践的な開発・運用までを体系的に学びます。まず、クラウドネイティブの概念を理解し、クラウドサービスの仕組みやAWSを中心とした主要技術を把握します。
次に、Dockerによるコンテナ技術、Kubernetesによるオーケストレーション、サーバーレスやマイクロサービスによる分散型アプリケーション設計を学習します。さらに、アジャイルとDevOpsの手法を通じて継続的改善と迅速な開発を実現し、CI/CDによる自動化パイプラインで効率的なデプロイと品質向上を体験します。IaC（Infrastructure as Code）では、CloudFormationやTerraformを用いたインフラ自動化を実践し、可観測性の章では、ログ・メトリクス・トレースを活用したシステム監視と分析手法を習得します。
最後に、ゼロトラストやDevSecOpsなどの考え方を通じて、クラウドネイティブ環境におけるセキュリティ対策を体系的に学び、安全で持続可能なクラウド運用を実現する力を身につけます。

授業計画

回数	項目	内容
第1回	クラウドネイティブの基本	クラウドネイティブの基本概念と構成要素を体系的に学びます。
第2回	クラウド技術	クラウドコンピューティングの仕組みと主要サービスを理解し、AWSを中心にクラウド技術の基礎を学びます。
第3回	コンテナ技術	コンテナ技術の仕組みとDockerを中心とした基本操作を体系的に学びます。
第4回	サーバーレスアーキテクチャ	サーバーレスアーキテクチャの概念と仕組みを理解し、FaaSやCaaSの活用を学びます。
第5回	マイクロサービス	マイクロサービスアーキテクチャの概念と設計手法を体系的に学びます。
第6回	アジャイルとDevOps	アジャイルとDevOpsの理念・手法を理解し、クラウドネイティブを支える開発文化を学びます。
第7回	CICD	CI/CD（継続的インテグレーション／デリバリー）の概念と実装を学びます。
第8回	IaC (Infrastructure as Code)	IaC（Infrastructure as Code）の概念と主要ツールを理解し、インフラ自動化の基礎を学びます。
第9回	可観測性	クラウドネイティブ環境における可観測性（Observability）の概念と実践を学びます。
第10回	クラウドネイティブにおけるセキュリティ	クラウドネイティブ環境におけるセキュリティの考え方と実践的対策を学びます。

科目名	コンテナ技術のシステム構築

科目の概要

本講座では、DockerからKubernetes、さらにAWS EKSまでを通じて、クラウドネイティブなシステム構築を総合的に学びます。Dockerによるコンテナ操作やイメージ管理、Composeによる複数コンテナの連携を実践し、開発環境構築の基礎を習得します。
Kubernetesでは、PodやDeploymentによるマニフェスト運用、スケーリングやローリングアップデートを通じて自動復旧や継続的デリバリーの仕組みを理解します。加えて、ServiceやLoadBalancerを用いたネットワーク設計、PersistentVolumeやSecretによるデータの永続化と安全管理、リソース最適化や自動スケーリングなど、運用効率化の技術も習得します。
さらに、ProbeやStatefulSet、DaemonSetを活用し、障害に強い高可用性クラスタを構築します。最終的に、AWS EKS環境でEBS・EFSによるストレージ連携やSecrets Managerとの統合を行い、セキュアで拡張性の高いクラウド基盤構築スキルを身につけます。

授業計画

回数	項目	内容
第1回	Dockerの利用	Dockerの基本操作から複数コンテナの統合管理まで、コンテナ開発の実践的スキルを体系的に学びます。
第2回	Kubernetesの基本	Kubernetesの基礎概念からクラスタ構築・操作までを体系的に学びます。
第3回	Kubernetesのシステム構築	Kubernetesにおけるマニフェスト運用とデプロイメントの仕組みを通して、宣言的なシステム構築の実践を学びます。
第4回	Kubernetesのネットワーク通信	Kubernetesにおけるネットワーク通信の仕組みとServiceリソースの活用を体系的に学びます。
第5回	Kubernetesのストレージとデータ管理	Kubernetesにおけるストレージとデータ管理の仕組みを実践的に学びます。
第6回	Kubernetesのスケジューリング	KubernetesにおけるPodスケジューリングの仕組みと制御手法を体系的に学びます。
第7回	Kubernetesのリソース管理	Kubernetesにおけるリソース管理の仕組みと最適化手法を学びます。
第8回	Kubernetesの可用性管理	Kubernetesにおける可用性管理の仕組みと実践を学びます。
第9回	パブリッククラウド上のコンテナサービス①	AWS EKSを中心にパブリッククラウド上のコンテナサービスを理解し、実践的に学びます。
第10回	パブリッククラウド上のコンテナサービス②	AWS EKS上でのストレージ連携とシークレット管理を中心に、クラウド上のKubernetes運用を深化させることを学びます。

科目名	クラウドネイティブのシステム開発入門

科目の概要
本講座では、AWS上に、ALB＋ECS＋RDSで構成された三層アーキテクチャのWebアプリケーションを構築します。 すべてのリソースをAWS CDK（Infrastructure as Code）で自動化し、開発からデプロイまでの一連の流れを体得し、演習を通じて基盤構築・DevOps・アプリ開発・セキュリティに至るまで、フルスタックなスキルを習得します。 受講後には、自立してクラウド環境を設計・構築・運用できるエンジニアとしての基礎力を身につけることを目指します。

授業計画		
回数	項目	内容
第1回	クラウドネイティブ基盤実装の概要および環境準備	AWS上に顧客情報管理システムを構築します。
第2回	ネットワーク構築	VPC・サブネット・セキュリティグループなど仮想ネットワークを構築し、トラフィック制御の基礎を理解します。
第3回	コンテナ構築	ECRを用いたコンテナイメージ管理を行い、ECS Fargateでサーバレスコンテナを起動します。
第4回	DB構築	アプリケーションデータを永続化するために、RDS MySQLを構築します。
第5回	CI/CDパイプライン構築	GitHubと連携し、CodePipeline・CodeBuild・CodeDeployを組み合わせたCI/CDパイプラインを構築します。
第6回	ロードバランサーのHTTPS移行	ALBをHTTPS化し、通信を暗号化することでセキュアなWebアプリケーションを実現します。
第7回	CI/CDパイプラインのユニットテスト追加	既存のCI/CDパイプラインにテストステージを追加し、品質を自動的に検証する仕組みを実装します。
第8回	総合演習	おみくじ占いアプリケーションを実装し、複数アプリケーションを統合管理できるCDK構成を実践します。

科目名	コンテナ・クラウドサービスのセキュリティ

科目の概要

本講座では、Kubernetesを中心としたクラウドネイティブ環境に必要なセキュリティと可観測性、DevSecOpsの実践を演習を通じて体系的に学びます。
まず、RBAC・NetworkPolicy・PSSを用いて脆弱なクラスターを安全な構成へと改善し、最小権限・通信制御・特権制限の重要性を理解します。続いて、Trivy・Cosign・Kyvernoによるイメージ堅牢化と署名検証の自動化、そしてseccompなどのランタイム防御により、安全なイメージ運用を実現します。
さらに、GitHub Actionsでコード解析や脆弱性スキャンを組み込んだCIパイプラインを構築し、Shift Left型のDevSecOpsを体験します。加えて、Istioを用いてmTLS通信や認可制御、Secrets管理を実装し、ゼロトラストな通信基盤を構築します。最後に、Kiali・Prometheus・Grafanaでサービスメッシュの可観測性を強化し、通信の可視化と異常検知を通じて安定運用に必要な分析スキルを習得します。

授業計画		
回数	項目	内容
第1回	コンテナ・クラウドサービスのセキュリティの概要	クラウドネイティブ環境のコンテナセキュリティを4Cモデルで体系的に理解し、コードからホストまでのライフサイクル全体に必要な対策を学びます。
第2回	Kubernetes環境の診断と防御	RBAC・NetworkPolicy・Pod Security Standardsを用い、脆弱なKubernetes環境を安全な構成へ変換する演習です。
第3回	安全なコンテナ構築とランタイム防御	Trivy・Cosign・Kyvernoを用いてイメージ堅牢化とランタイム防御を行う演習です。
第4回	パイプラインへのセキュリティゲート導入	GitHub Actionsでコード解析と脆弱性スキャンを自動化するCIパイプラインを構築する演習です。
第5回	サービス間通信の暗号化とシークレット管理	サービスメッシュによる通信暗号化とシークレット管理を実践する演習です。
第6回	Kialiによる可視化と通信分析	サービスメッシュの可観測性を実践する演習です。

令和７年度「専門職業人材の最新技能アップデートのための専修学校リカレント教育（リ・スキリング）推進事業」

情報技術者の技能アップデートのためのリカレント教育推進事業

コンテナ技術リカレント教育プログラム

令和８年２月

一般社団法人全国専門学校情報教育協会

〒164-0003 東京都中野区東中野 1-57-8 辻沢ビル３F

電話：03-5332-5081 FAX.03-5332-5083

●本書の内容を無断で転記、掲載することは禁じます。