AWS Managed Container Service
ECS or EKS or ROSA ?? + ROSA HCP 紹介

Red Hat K.K.

2025.05

Red Hat

Agenda

- ECS/EKS/ROSAの概要とライフサイクル
- EKS/ROSAサンプルユースケース その1
- EKS/ROSAサンプルユースケース その2
- ROSA Hosted Control Plane サービス仕様
- AWSサービスとの連携
- サポート体制
- Appendix:
  AWSに持ち込むRed Hat製品 Bring your Own Subscription (BYOS)

Red Hat

ECS/EKS/ROSAの概要とライフサイクル

Red Hat

AWS上でのコンテナサービスの選択肢

https://aws.amazon.com/jp/getting-started/decision-guides/containers-on-aws-how-to-choose/

ページの内容

- はじめに
- 理解
- 検討事項
- 選択
- 使用
- 詳しく見る

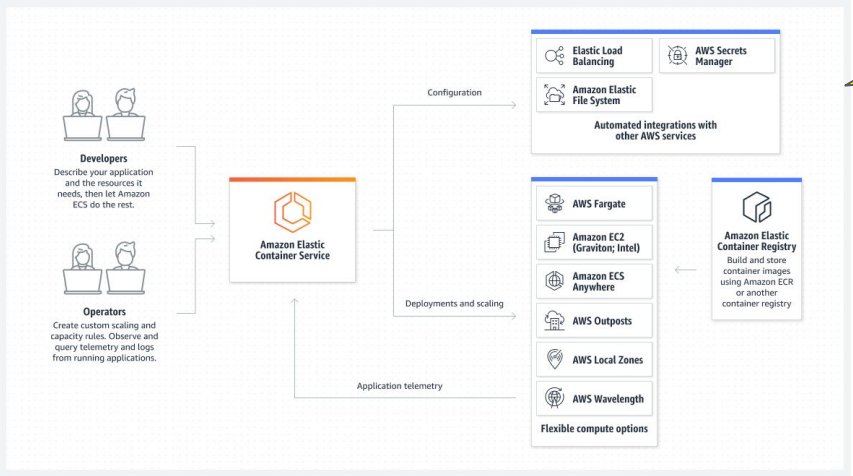
コンテナ	使用するタイミングは?	どのような用途に最適化されていますか?	関連するコンテナサービスまたはツール
容量	セルフマネージド型の AWS 仮想マシンや AWS マネージドコンピューティングでコンテナを実行する場合に使用します。	AWS コンピューティングでコンテナを実行するのに最適化されています。	AWS Fargate Amazon EC2
オーケストレーション	最大数千のコンテナをデプロイして管理する必要がある場合に使用します。	AWS でコンテナ化されたアプリケーションをデプロイ、管理、スケーリングするのに最適化されています。	Amazon ECS Amazon Elastic Kubernetes Service Red Hat OpenShift Service on AWS (ROSA)
プロビジョニング	ユーザーや自身のチームがコンテナやインフラストラクチャを使用した経験があまりない場合に使用します。	使いやすさを考慮して最適化されています。	AWS App Runner Amazon Lightsail AWS Elastic Beanstalk
ツール	コンテナレジストリを提供するだけでなく、既存のアプリケーションをコンテナ化および移行するツールが必要な場合に使用します。	コンテナ運用のサポートに最適化されています。	Amazon Elastic Container Registry
オンプレミス	使い慣れたコントロールプレーンを実行する必要がある場合に使用すると、コンテナベースのアプリケーションがどこで実行されていても一貫したエクスペリエンスを実現できます。	コンテナベースのアプリケーションを実行する場所に柔軟性を持たせるように最適化されています。	Amazon Elastic Container Service (ECS) Anywhere Amazon EKS Anywhere Amazon EKS Distro

コンテナオーケストレーションサービスにフォーカス

Red Hat

Amazon ECS

- マネージドのコンテナオーケストレーションサービス
- AWSネイティブな方法で、Dockerフォーマットのコンテナの大規模実行が可能
  - 様々なAWSサービスとの連携を、迅速に設定して利用可能
  - コンテナクラスター利用料金が無いことによる、コスト最適化が可能
  - コントロールプレーンは完全に管理され、利用者によるアップグレードなどの対応は不要
- **AWS上でのアプリケーションを開発 /実行するユースケースを想定**
  - CNCF(後述)のプロジェクトを利用したい Kubernetesユーザーには非推奨



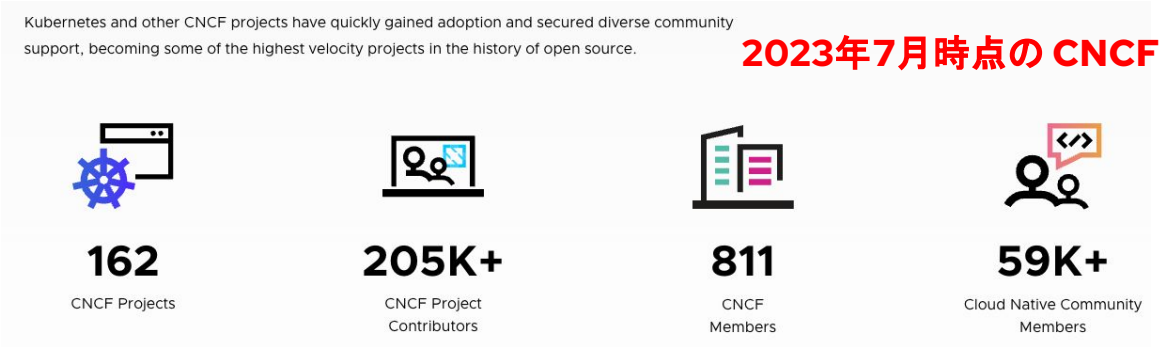
様々なAWSサービスとの連携が可能

Red Hat

Cloud Native Computing Foundation (CNCF)

https://www.cncf.io/about/who-we-are/

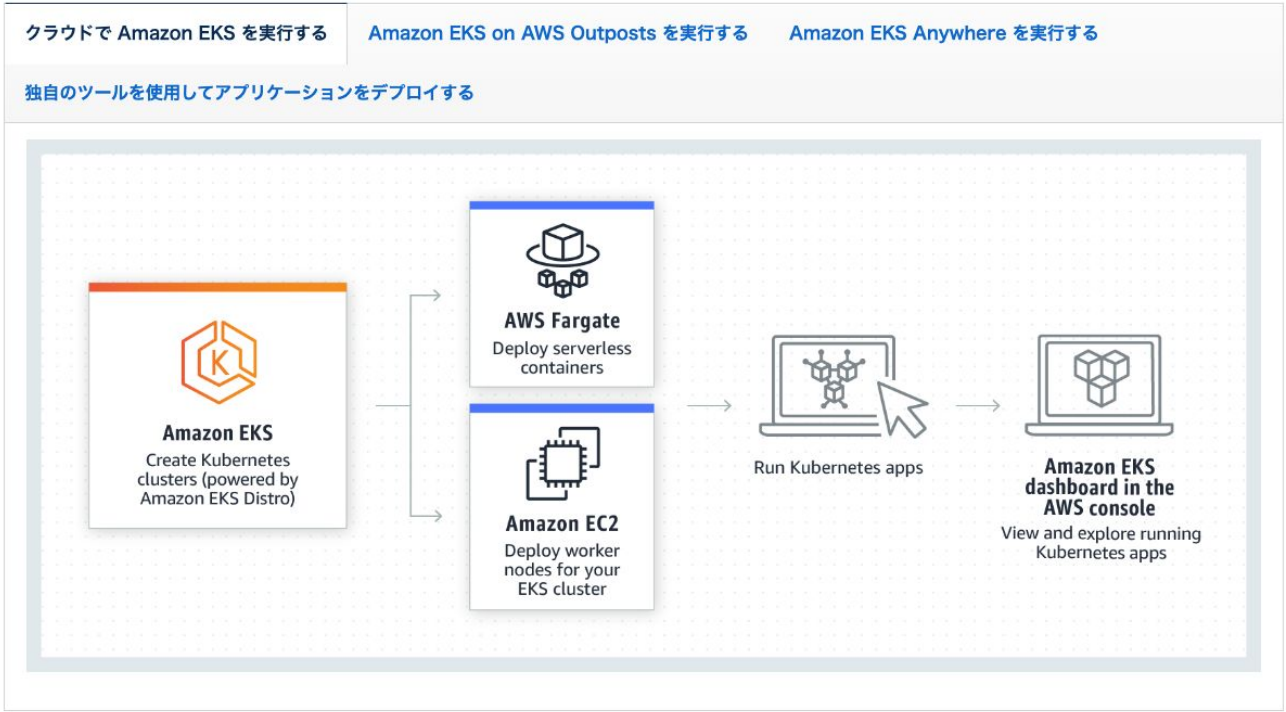
- CNCFはクラウドネイティブコンピューティング技術を推進する非営利団体であり、Linux Foundationプロジェクトの1つ
- 2015年のGoogleによるKubernetes v1発表と同時に、CNCFも発表
  - https://cloudplatform.googleblog.com/2015/07/Kubernetes-V1-Released.html
- AWSとRed Hatを含む多くの企業が参画しており、業界標準ソフトウェアとなっているKubernetesの発展と密接にリンク
- Amazon EKSの提供は、AWS顧客のKubernetesサービス需要によるもの
  - https://aws.amazon.com/jp/blogs/opensource/cloud-native-computing/



Amazon EKS

https://aws.amazon.com/jp/eks/

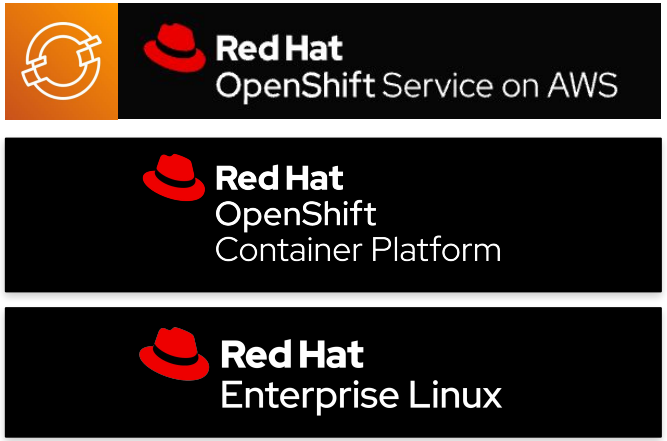
- AWS クラウドおよびオンプレで Kubernetes を実行するためのマネージド Kubernetes サービス



Red Hat OpenShift と Red Hat OpenShift Service on AWS (ROSA)

https://www.redhat.com/ja/technologies/cloud-computing/openshift/red-hat-openshift-kubernetes

- Red Hat OpenShift
  - Red Hat Enterprise Linux (RHEL) と並ぶ主力製品の1つ
  - RHELと統合されており、RHELをベースとしたコンテナ専用OSの上で動作
  - Kubernetesがベース
  - エンタープライズシステムで必要となる、様々な開発/運用関連の機能が統合
    - 前述のCNCFプロジェクトの機能を多数搭載
- ROSA
  - AWS上のマネージドOpenShiftサービス
  - Red HatとAWSによるサポートを提供



サービス基盤の機能実装に必要なAWSサービス

ROSAにより、基盤構築に必要な時間の短縮が可能

ROSA提供の
OpenShift標準機能
を利用可能

要件に応じたAWSサービスの取捨選択と、
EKSからそのサービスを利用する場合、
EKSクラスター内での、
AWS IAMによる権限設定が別途必要

Kubernetesによる
システム作り込み時に
必要な主要機能

OpenShiftでは、統合された認証/認可機能により、各標準機能を利用するための権限設定がデフォルトで適用済み

	Amazon EKSを使う場合	ROSAを使う場合 ROSA 標準機能
Dashboard	Amazon EKS Console	Built in Console
Developer IDE/tools	AWS Cloud9 / Amazon CodeCatalyst	OpenShift Dev Spaces
Build Automation	AWS CodeBuild	S2i (Source-to-Image) Build
Pipeline (CI)	AWS CodePipeline	OpenShift Pipelines
Deployment Automation (CD)		Openshift GitOps
Serverless	AWS Lambda	OpenShift Serverless
Service Mesh	AWS AppMesh / AWS X-Ray	OpenShift Service Mesh
Monitoring	Amazon Managed Service for Prometheus	OpenShift Monitoring
Logging	Amazon CloudWatch	OpenShift Logging
Registry	Amazon ECR	OpenShift Internal Registry
Orchestration	Amazon EKS	Kubernetes
Infrastructure	AWS Compute Resources	AWS Compute Resources

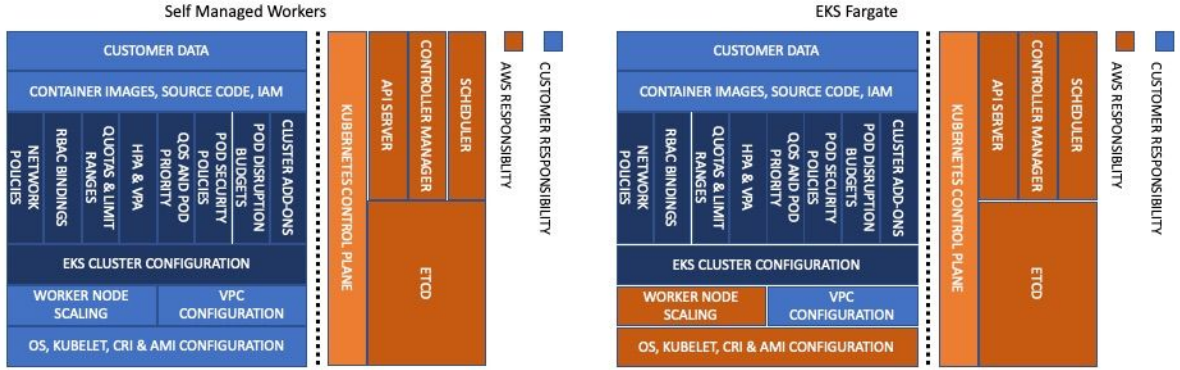
Red Hat

Amazon EKS Shared Responsibility Model

https://aws.github.io/aws-eks-best-practices/security/docs/

EKSは、Kubernetesクラスターの管理を、一部委任したいという希望者向けのサービス
(クラスターの作成 /削除/更新をセルフサービスで実施可能)

- AWSが管理/サポート
  - Amazon EKS コントロールプレーン
  - Amazon EKS ワーカーノード (Fargate, Managed Node Groups)
  - Amazon EKS アドオン (CoreDNS, Kube-proxyなど)
- ユーザーが管理※
  - Amazon EKS ワーカーノード (Self Managed Workers)
  - EKSに含まれない、CNCFなどが提供するKubernetes (K8s) 用アドオン
  - K8sのセキュリティ設定 (Pod, Image, Network, RBAC, Multi-tenancyなど)
  - ユーザーアプリやデータ、および、それらの実行基盤となるコンテナ上のミドルウェア



※ 各AWSサービスの利用支援や障害復旧はAWSがサポートしますが、EKSへの統合や更新は、ユーザーが実施する必要があります。

Red Hat

ROSA Shared Responsibility Model

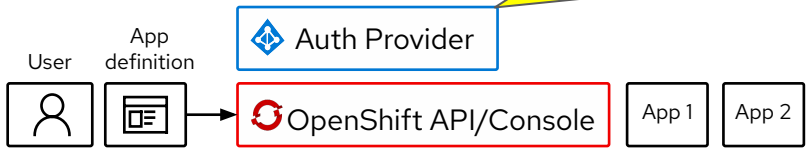
https://docs.openshift.com/rosa/rosa_architecture/rosa_policy_service_definition/rosa-policy-responsibility-matrix.html

ROSAは、Kubernetesからコンテナの OS、ミドルウェアまでの統合サポートの希望者向けのサービス

- Red HatとAWSが管理/サポート
  - コントロールプレーン/ワーカーノード、K8s Network/DNS関連機能、ロードバランサーを含めたクラスター管理
  - OpenShiftは予めセキュリティが強化されたK8sとして設定され、Red Hatがその利用をサポート
    (Pod, Image, Network, RBAC, Multi-tenancy, ホストOS など)

Responsibilities	
User management	Customer
Project and quota management	Customer
Application lifecycle	Customer
Cluster creation	Self-service automation
Cluster management	Self-service automation
Monitoring and logging	Self-service automation
Network configuration	Self-service automation
Software and security updates	Red Hat/AWS
Platform support	Red Hat/AWS

OpenID Connect/LDAP/Google/GitLab/GitHub を OpenShiftの認証プロバイダーとして設定可能



Let Red Hat...

- Manage all your clusters
- Secure your nodes
- Monitor and operate your VMs
- Manage environment patches

- ユーザーは、ユーザーアプリ/データや認証情報を管理し、Red Hatにより提供/サポートされるRHELやミドルウェア※を使ったコンテナアプリ開発が可能
- クラスターの作成/削除/更新/リソース利用率の確認、アプリのモニタリング/ロギング、仮想ネットワークの設定、などの機能をセルフサービスで利用可能

※ 一部のRed Hat提供のミドルウェア利用については、Red Hatからサブスクリプションを別途購入する必要があります。

EKS/ROSA ライフサイクル (2025年5月時点)

EKS: 最低14ヶ月※、ROSA: 16ヶ月のサポートを提供

Kubernetes Version	Upstream release	EKS release	EKS End of standard support	EKS End of extended support (注: クラスター料金が 6倍になります)
1.32	2024年12月11日	2025年1月23日	2026年3月23日	2027年3月23日
1.31	2024年8月13日	2024年9月26日	2025年11月26日	2026年11月26日
1.30	2024年4月17日	2024年5月23日	2025年7月23日	2026年7月23日

Kubernetes version	ROSA Version	ROSA release	ROSA End of Life
1.30	4.17	2024年10月14日	2026年2月14日
1.29	4.16	2024年7月2日	2025年11月2日
1.28	4.15	2024年2月27日	2025年6月30日

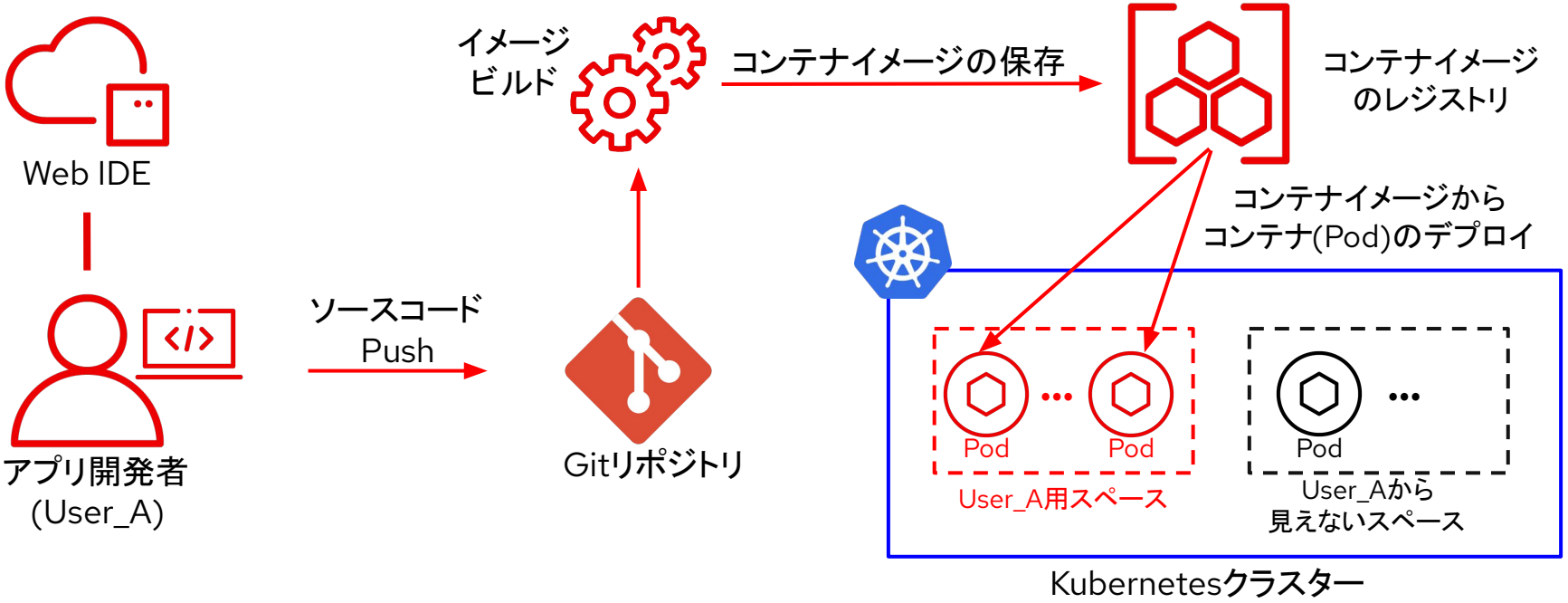
※ EKSでは、最低 4つのKubernetesバージョンをサポートするように定義しており、 Upstreamのリリース状況に応じて、14ヶ月以上のサポートを提供する場合があります。

Red Hat

EKS/ROSA
サンプルユースケース その1

EKS/ROSA のサンプルユースケース その1

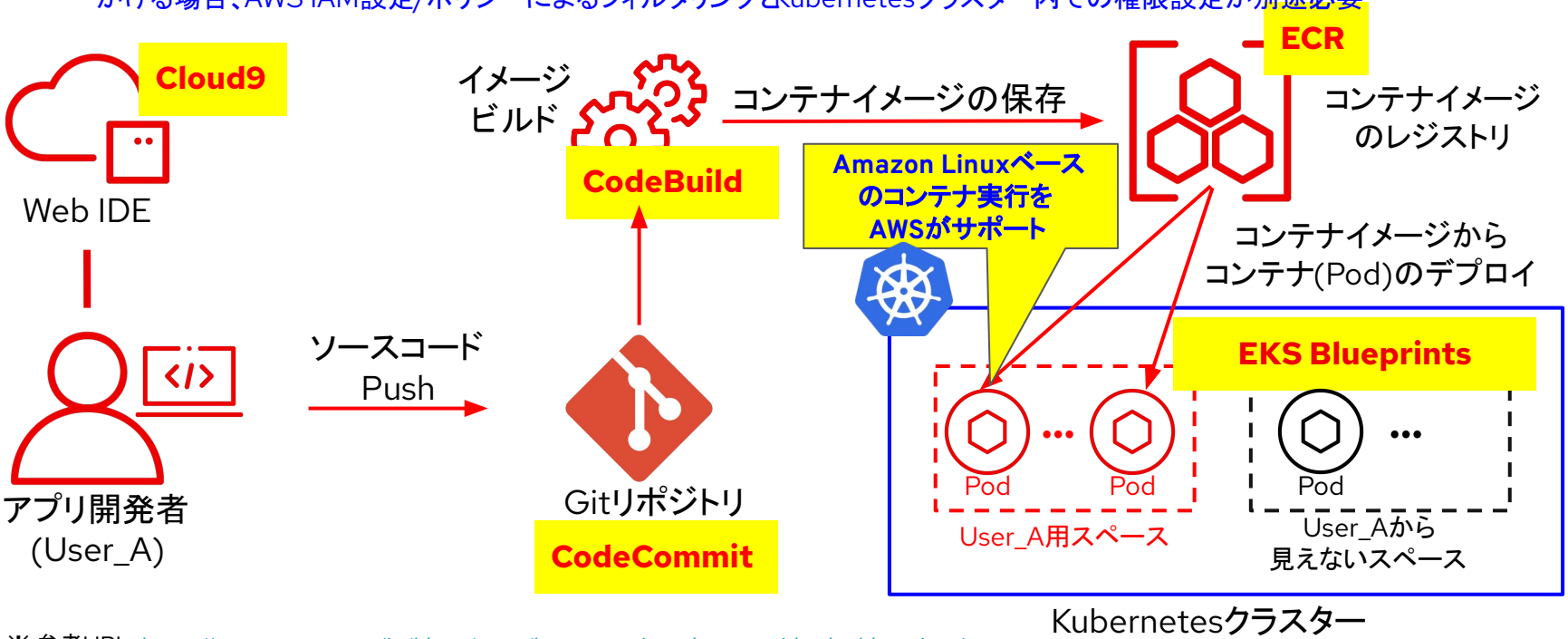
- Web IDEを使ったコンテナアプリの開発とデプロイ
- 開発者は、自分の開発/デプロイしたコンテナアプリしか見えない状態を想定



Red Hat

EKSを使う場合のイメージ

- 他のAWSサービス (AWS Cloud9/CodeCommit/CodeBuild/ECR) を合わせて利用
- EKSのマルチテナント化には、EKS Blueprints※というオープンソースプロジェクトを利用可能
- 各AWSサービスを利用するためのAWS IAM設定(認証/認可)と、ユーザーごとに参照可能なECRレジストリの制限をかける場合、AWS IAM設定/ポリシーによるフィルタリングとKubernetesクラスター内での権限設定が別途必要

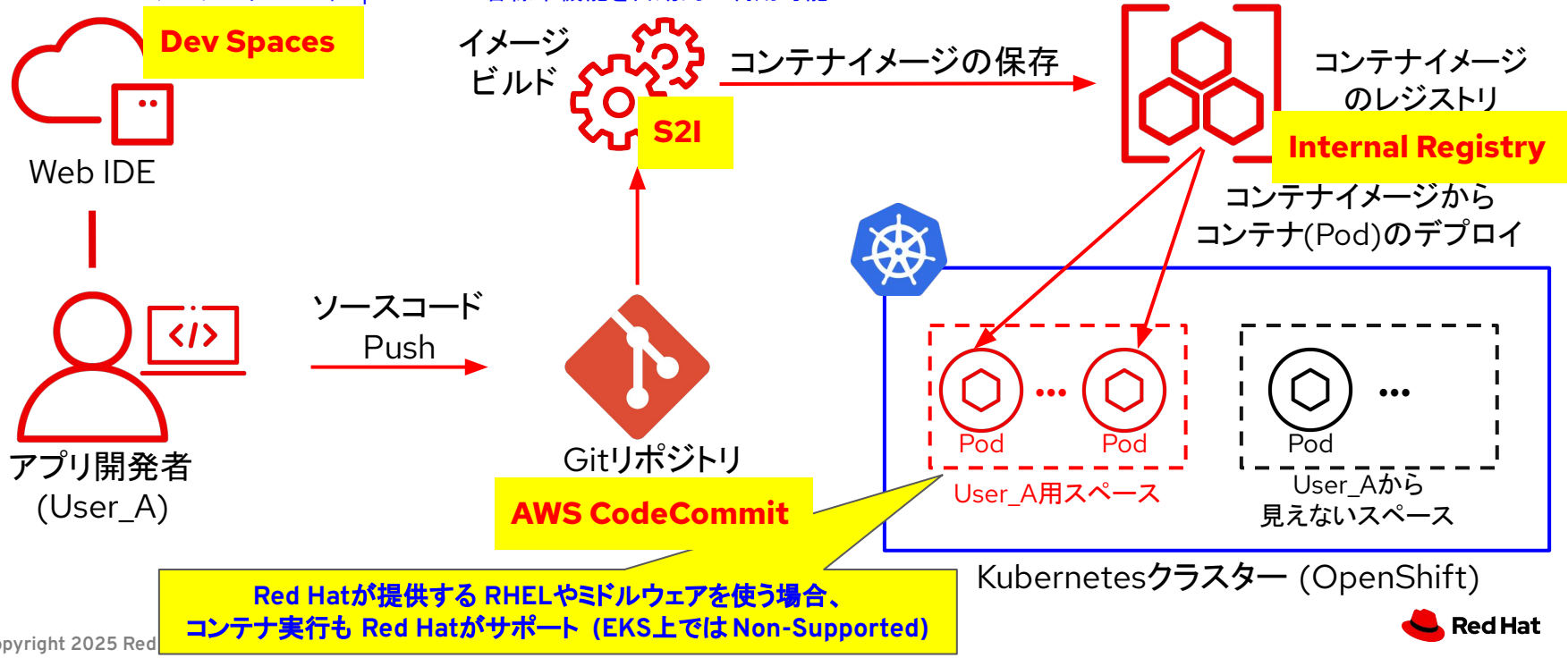


※ 参考URL: https://aws.amazon.com/jp/blogs/news/bootstrapping-clusters-with-eks-blueprints/

Red Hat

ROSAを使う場合のイメージ

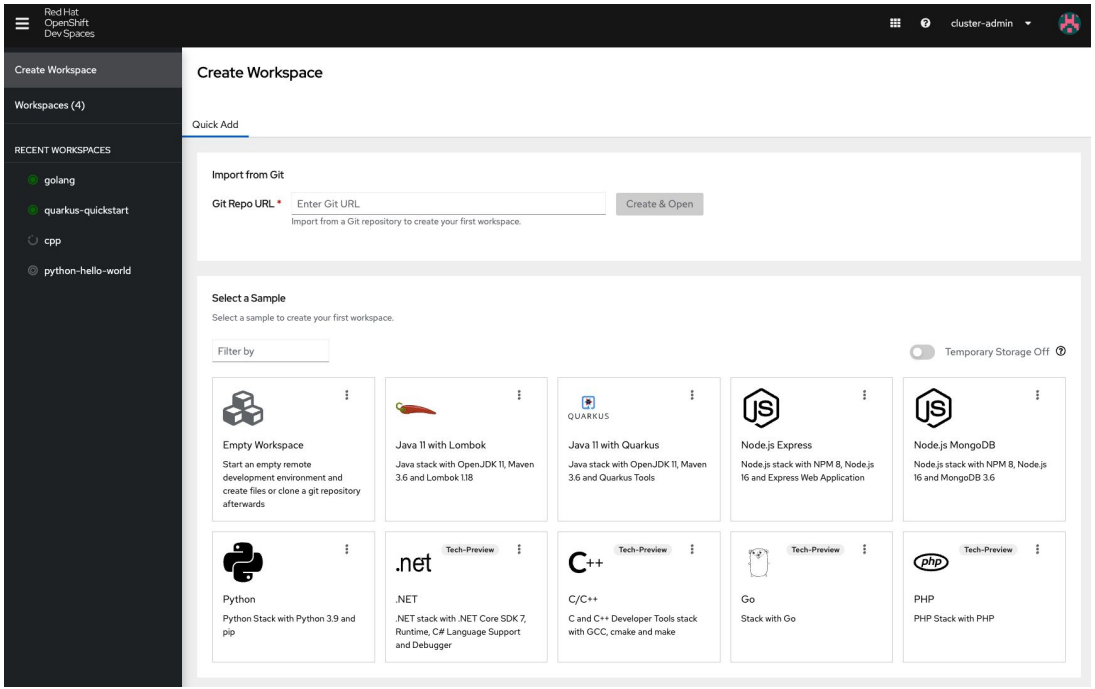
- ROSAにあるOpenShift標準機能 (OpenShift Dev Spaces/S2I/Internal Registry)を利用
- Gitリポジトリには、AWS CodeCommitを利用 (AWS IAM設定が必要)
- OpenShiftではマルチテナント化を考慮しており、ユーザーごとのスペースやレジストリがデフォルトで隔離済み
- OpenShiftに統合された認証/認可機能により、OpenShiftクラスターと連携した認証プロバイダー(OpenID Connectなど)のユーザーアカウントで、OpenShiftの各標準機能を自動的に利用可能



Red Hat

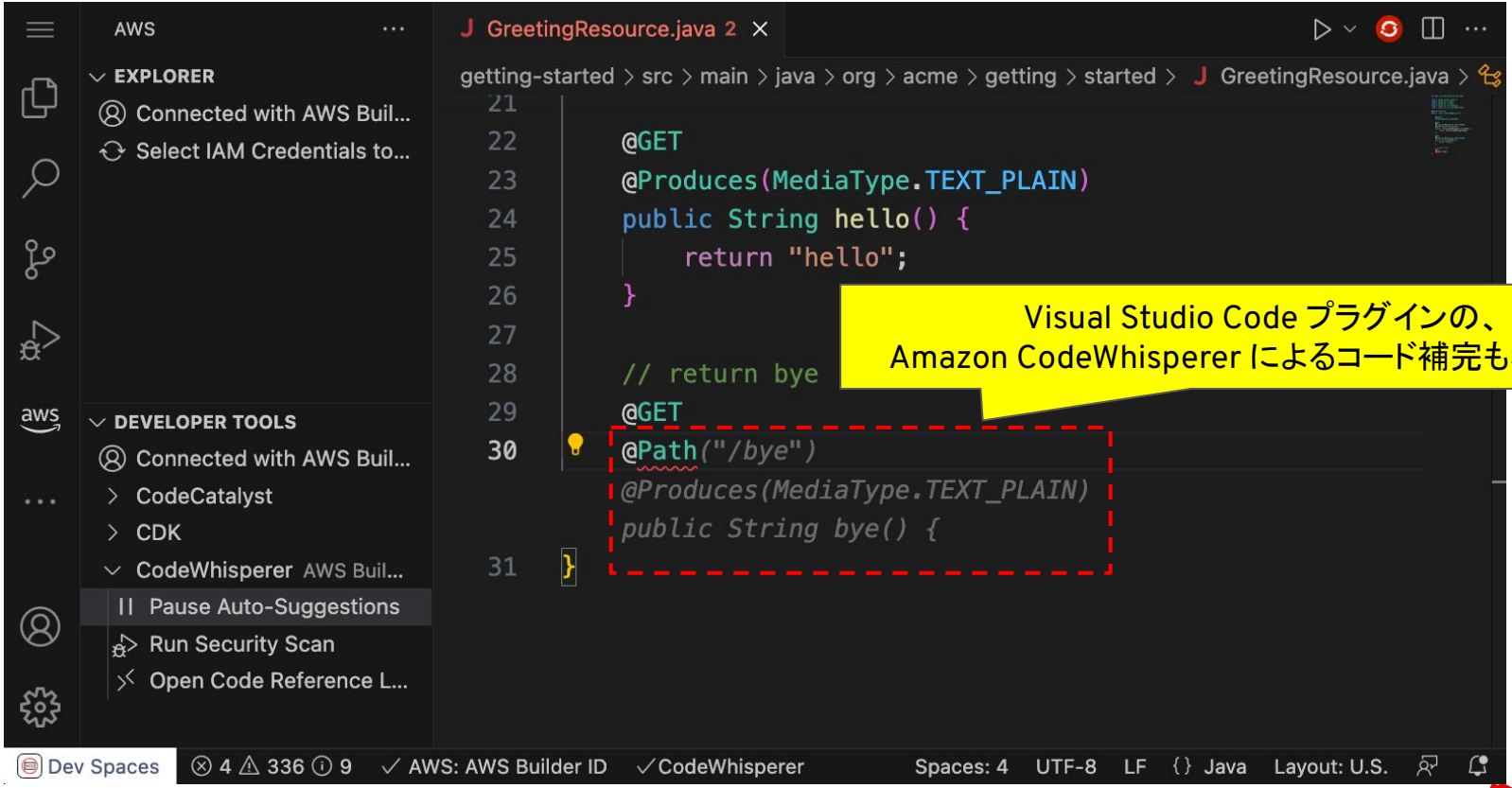
OpenShift Dev Spaces用のテンプレート

- 標準テンプレートの他に、カスタマイズされたWeb IDEも利用可能
- カスタマイズには専用のテンプレート(Devfile)※を利用

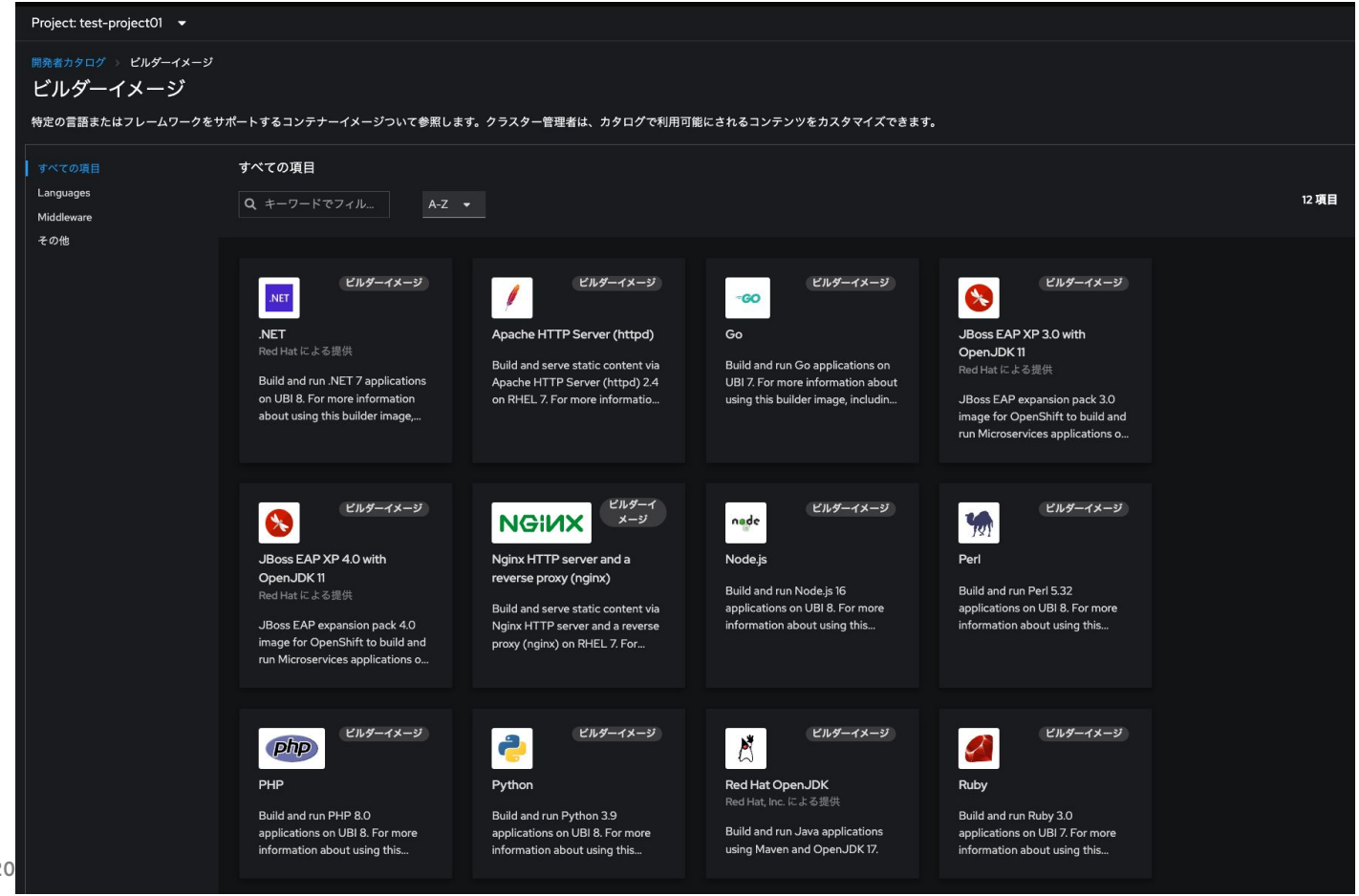


※ Devfile: Web IDEを定義するフォーマット(YAML)であり、ベースコンテナイメージやカスタムコマンドを定義。
CNCFプロジェクトの1つ。https://www.eclipse.org/che/

OpenShift Dev Spacesの画面
(Visual Studio Code的な Web IDE)



OpenShift S2I に利用可能なカタログ



Red Hat

OpenShift S2I の画面

OpenShift上のアプリケーショントポロジー
(ユーザーは、作成したアプリケーションの様々な情報を確認可能)

EKS/ROSA 利用時の月額(730時間)料金イメージ その1

※ ROSAのHosted Control Plane利用を想定。2024年4月時点では、東京と大阪リージョンで利用できます。

- AWSの東京リージョンのMulti-AZ構成を想定 (ワーカーノードが3台構成)
- EKSではManaged Node Group (EC2インスタンス)の利用を想定
- 計算の簡単化のために、EKS/ROSA共に利用が想定される NAT Gateway, AWS Load Balancer, Route53のサービス利用料金や、AWS内外でのデータ転送利用料金などを除外
- AWS CodeCommit は無料利用枠の利用を想定

AWSサービス	利用料金 (USD)
EKS (0.10USD/hour/cluster)	73 (0.10 x 730)
EC2 インスタンス (Worker) (m5.xlarge x3, 0.248USD/hour EBS(gp3): 300GB, 0.08USD/GB/month)	615.12 (0.248 x 3 x 730 + 300 x 3 x 0.08)
Cloud9用 EC2インスタンス (共有環境 m5.large(2vCPU, メモリ8GB) x3, EBS(gp3): 100GB)	295.56 (0.124 x 3 x 730 + 100 x 3 x 0.08)
CodeBuild (16時間/monthのビルドを想定 ビルド用インスタンス general1.medium (4vCPU, 7GB), 0.60USD/hour)	9.6 (0.6 x 16)
ECR (計1,000GBのイメージ保存を想定 保存料金: 0.10USD/GB/month)	100 (0.1 x 1000)
EKS利用時の想定合計金額(USD): 1093.28	

AWSサービス	利用料金 (USD)
ROSA サービス料金 (Cluster fee: 0.25USD/hour/cluster Worker Node service fee: 0.171USD/4vCPU/hour)	557 (0.25 x 730 + 0.171 x 3 x 730)
EC2 インスタンス (Worker) (m5.xlarge x3, 0.248USD/hour EBS(gp3): 300GB, 0.08USD/GB/month)	615.12 (0.248 x 3 x 730 + 300 x 3 x 0.08)
S3 (Internal Registryのバックエンド) (計1,000GBのイメージ保存(0.1GB/1リクエスト)を想定 書き込み料金 $0.0047/1,000リクエスト 保存料金: 0.025USD/GB/month)	25.47 (0.0047 x 10 + 0.025 x 1000)
ROSA利用時の想定合計金額(USD): 1197.59	

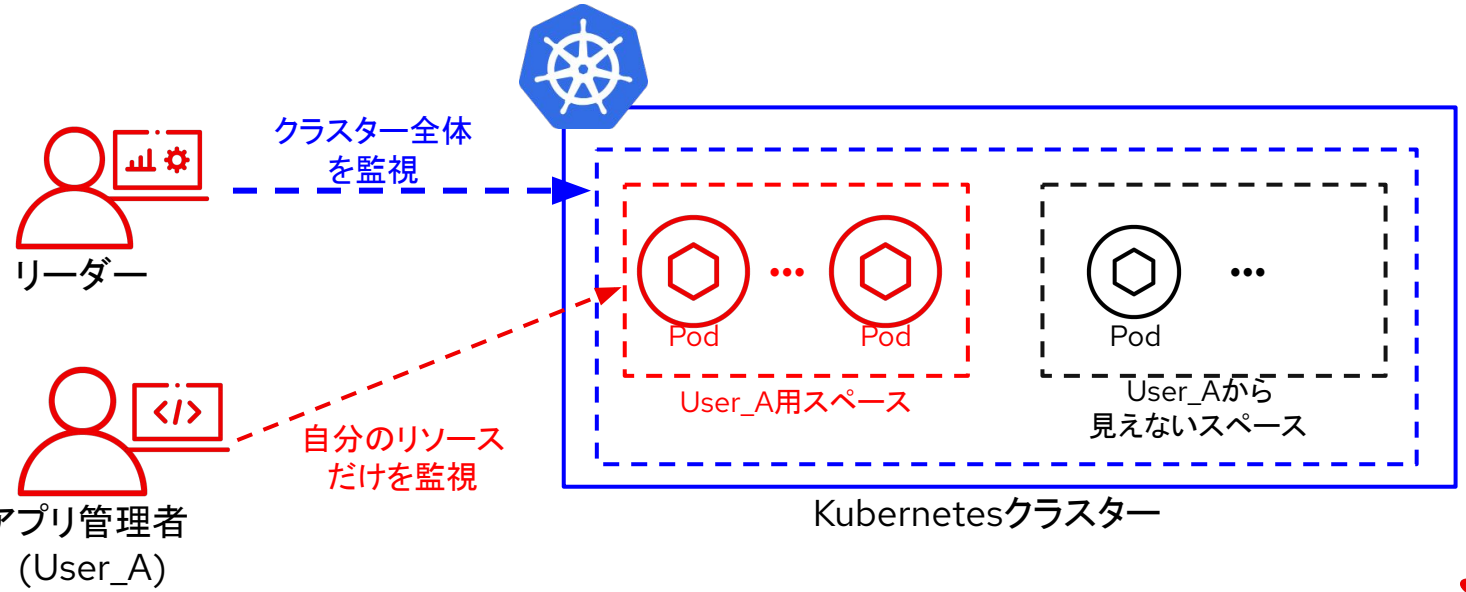
Red Hat

EKS/ROSA
サンプルユースケース その2

Red Hat

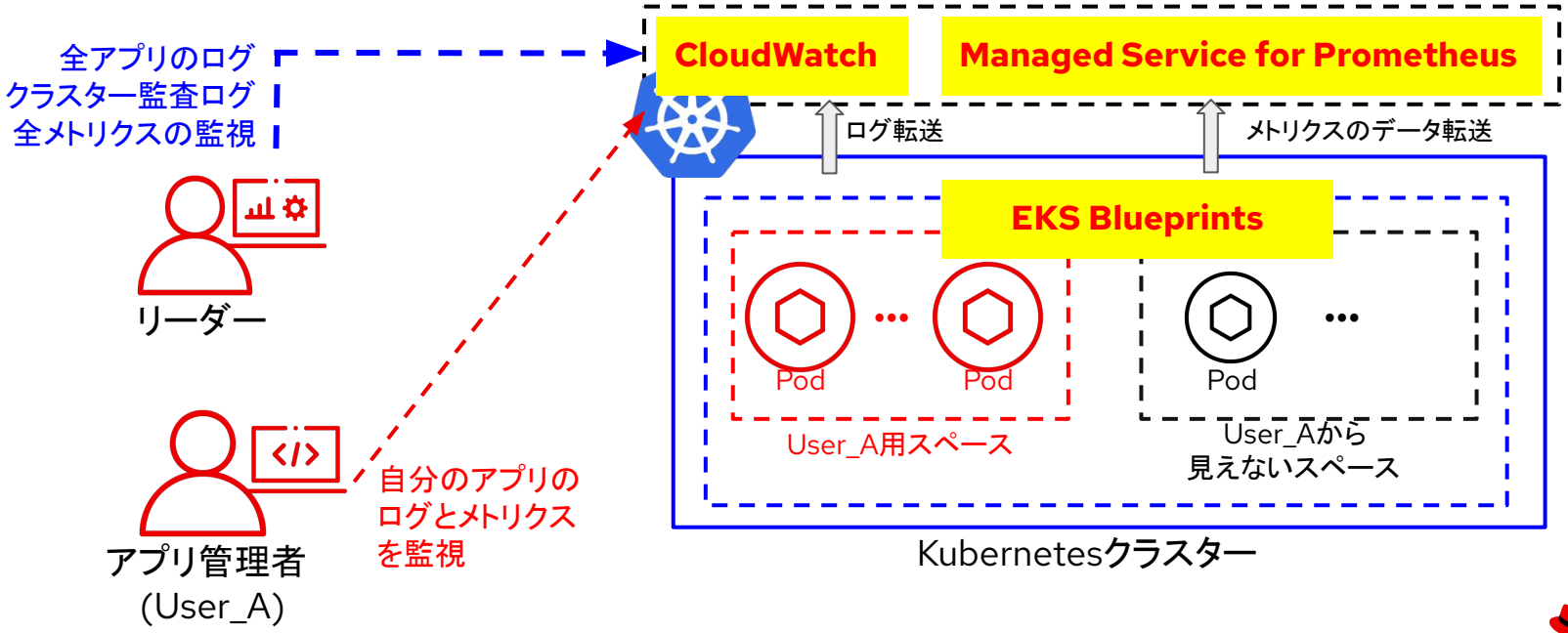
EKS/ROSA のサンプルユースケース その2

- Kubernetesクラスターのロギングとモニタリングを設定 /利用
- 開発部門のリーダーは、クラスター全体ログの集約や、アプリを実行するワーカーノードを中心としたクラスター全体のリソース (CPU/メモリなど)利用率を監視
- 開発者は、自分のスペースにあるアプリログやリソース利用率を監視



EKSを使う場合のイメージ

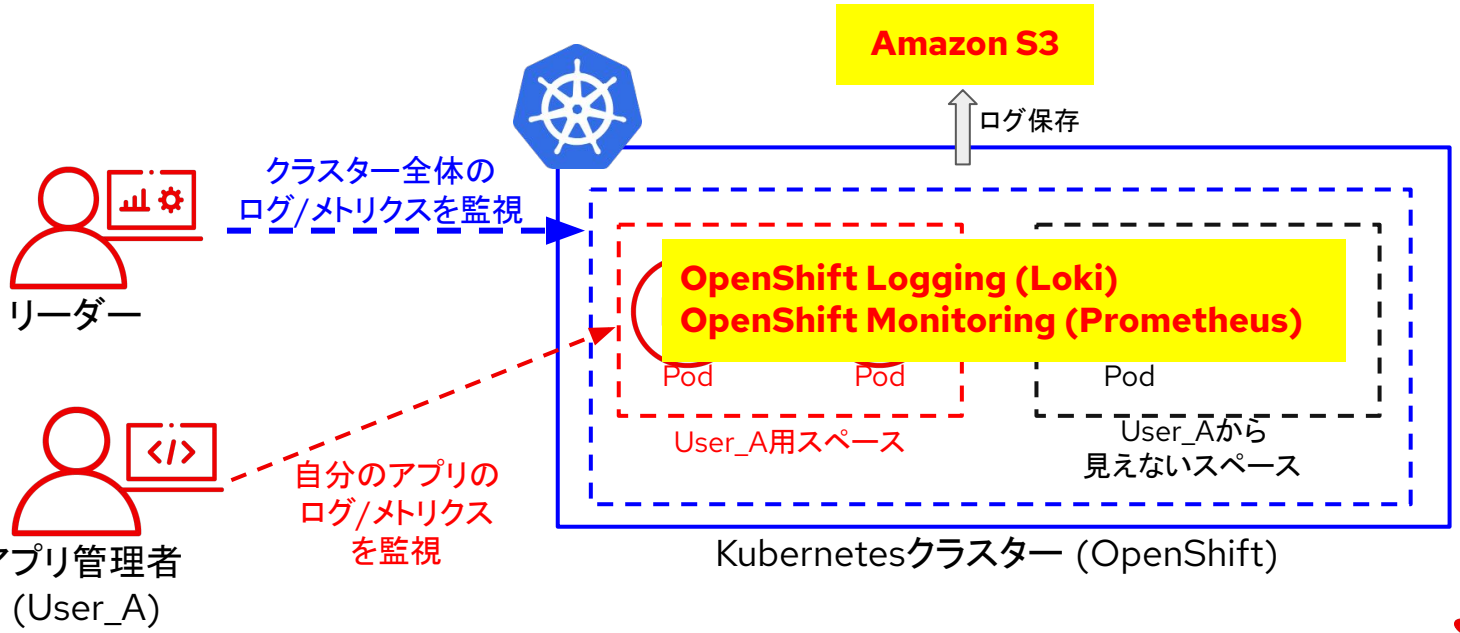
- ロギングには、Amazon CloudWatchを利用 (AWS CLIによるログのダウンロードも可能)
- モニタリングには、Amazon Managed Service for Prometheus (AMP) を利用
- CloudWatch/AMPを利用するための、AWS IAM設定/ポリシーによるフィルタリングや、データ転送のための Kubernetesクラスター内での権限設定が別途必要



Red Hat

ROSAを使う場合のイメージ

- ロギングには、OpenShift Logging (Loki) + Amazon S3 を利用 (CloudWatchも利用可能)
- モニタリングには、ROSA/OpenShiftのデフォルトで有効化されている Prometehusを利用
  - OpenShiftクラスターと連携した認証プロバイダーのユーザーアカウントが、
    管理するリソース利用率だけが見えるような設定が、デフォルトで適用済み



EKS/ROSAでのCloudWatchによるログ集約のイメージ

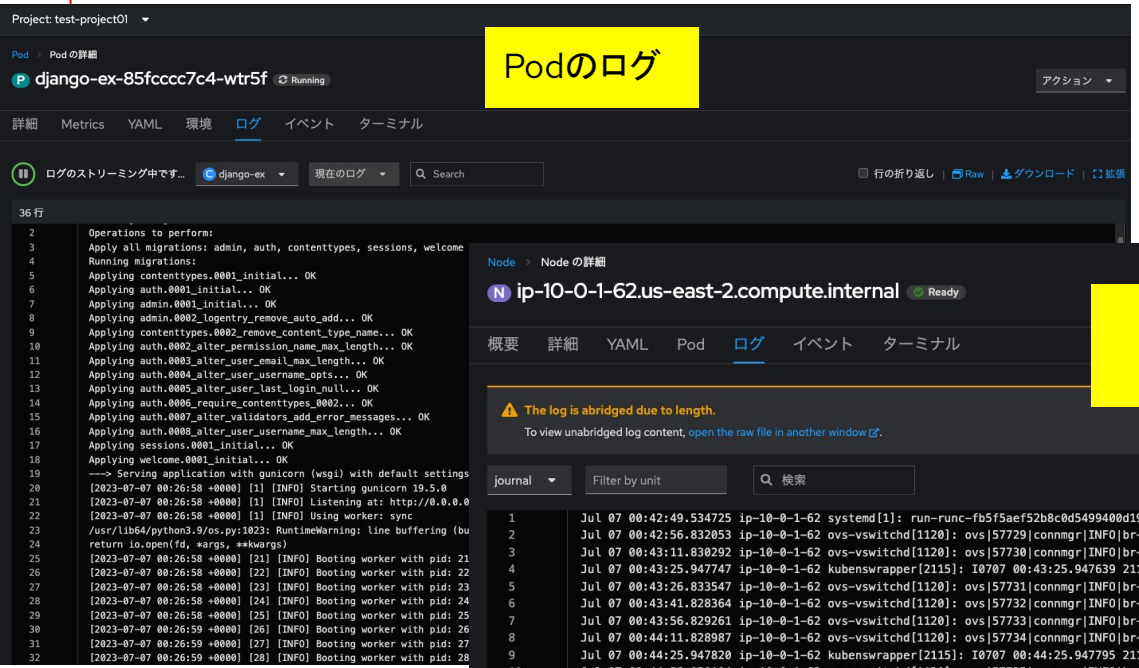
ログタイプごとのロググループ作成(アプリケーション/監査/インフラストラクチャー)
ROSAの場合、Red Hatのサポートケース経由で、監査ログのリクエストも可能です。

ユーザーアプリのログを参照可能

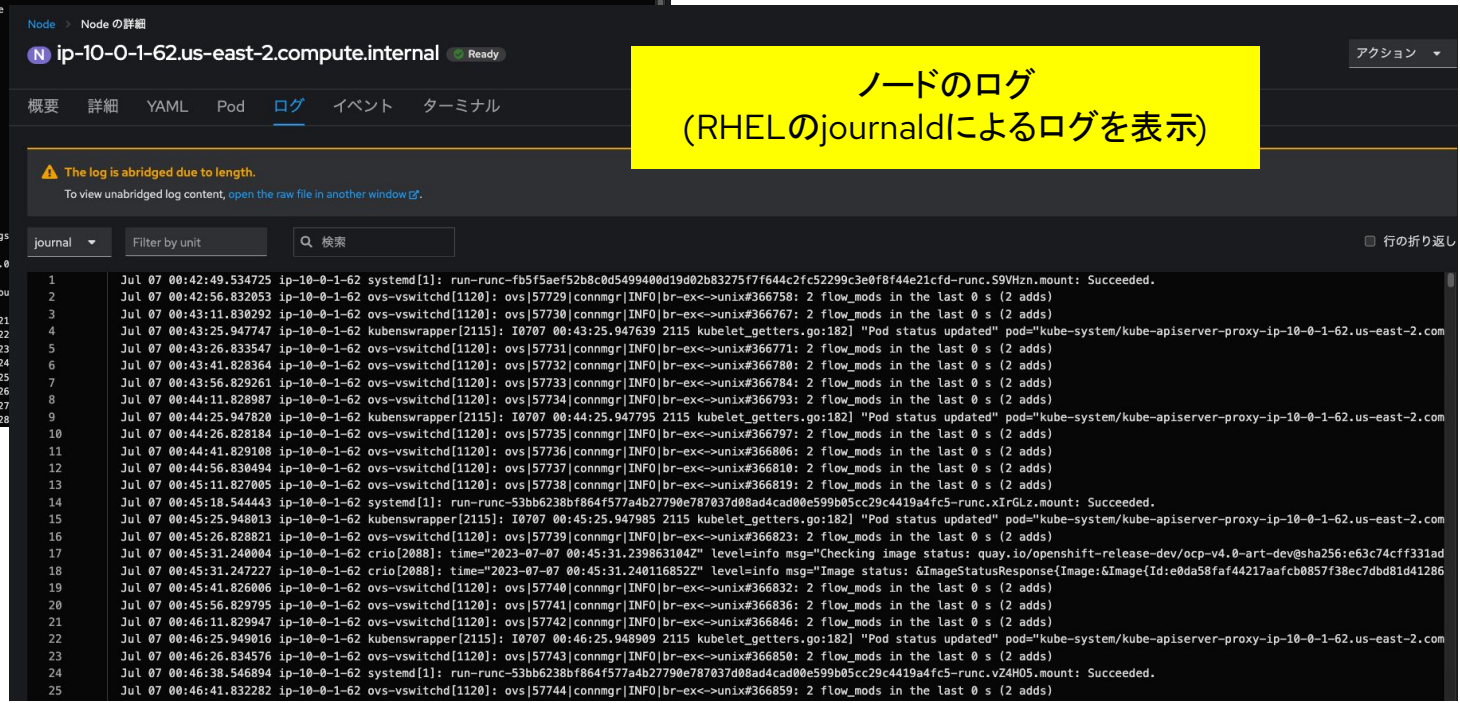
ROSAのOpenShift Logging Operatorによるログ転送では、ユーザーのスペース(この例では「test-project01」を指定)ごとにロググループを作成するような設定が可能

(EKSでもFluent-Bitによるログ転送時のフィルタリングルールを利用した、同等の設定が可能)

OpenShiftのコンソールでのログ確認



Podのログ

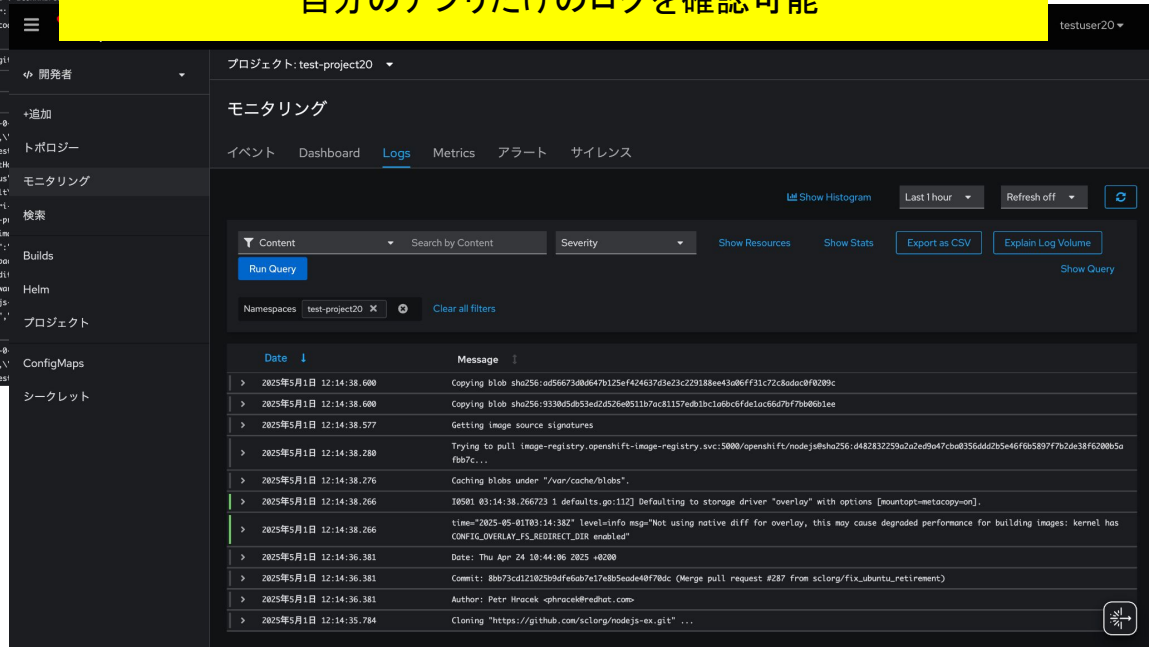


ノードのログ
(RHELのjournaldによるログを表示)

OpenShift Logging でのログ集約の確認

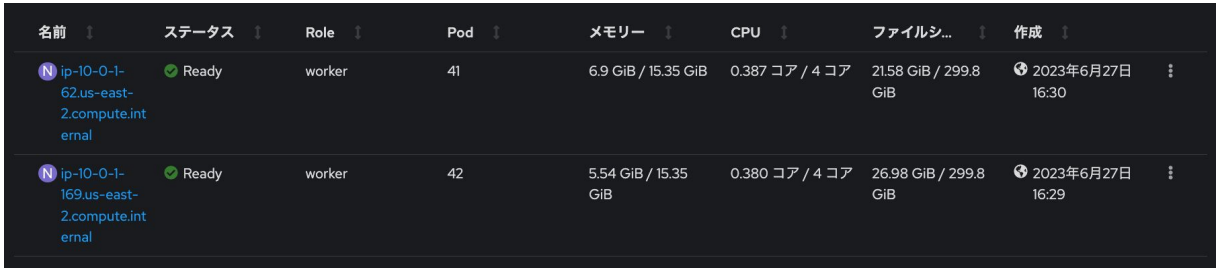
ROSAクラスター管理者は、
クラスター上の全てのアプリのログを確認可能

アプリ管理者は、クラスター管理者が設定した権限により、
自分のアプリだけのログを確認可能



OpenShiftのモニタリング (Prometheus)

クラスター全体や
各ノード/Podのリソース利用情報



OpenShiftのモニタリング (Prometheus)

※ ユーザーは自分のアプリに関する情報しか見えない状態

メトリクスの時間範囲や
更新間隔も適宜変更可能

ユーザーは自分のPodについて、
AMPと類似したコンソールによるメトリクス確認が可能

EKS/ROSA 利用時の月額(730時間)料金イメージ その2

※ ROSAのHosted Control Plane利用を想定。2024年4月時点では、東京と大阪リージョンで利用できます。

- 基本的には、「サンプルユースケース その1」と同じ利用条件と計算過程を想定
- EKS/ROSA以外に利用するAWSサービスは、CloudWatchとManaged Service for Prometheus (AMP)を想定
- CloudWatchでは、ログの収集と保存のみを利用すると想定
- AMPでは、下記の「Example 1 - EKS on EC2 and Kubernetes」のシナリオを想定
  - https://aws.amazon.com/jp/prometheus/pricing/

AWSサービス	利用料金 (USD)
EKS (0.10USD/hour/cluster)	73 (0.10 x 730)
EC2 インスタンス (Worker) (m5.xlarge x3, 0.248USD/hour EBS(gp3): 300GB, 0.08USD/GB/month)	615.12 (0.248 x 3 x 730 + 300 x 3 x 0.08)
CloudWatch (1,500GB/month のログサイズを想定 ログ収集: 0.76USD/GB/month ログ保存: 0.033USD/GB/month, ログ圧縮率0.2)	1,149.9 (0.76 x 1500 + 0.033 x 300)
AMP (メトリクスのストレージは、3.34GB)	81.75
EKS利用時の想定合計金額(USD): 1,919.77	

AWSサービス	利用料金 (USD)
ROSA サービス料金 (Cluster fee: 0.25USD/hour/cluster Worker Node service fee: 0.171USD/4vCPU/hour)	806.65 (0.25 x 730 + 0.171 x 5 x 730)
EC2 インスタンス (Worker) (m5.xlarge x3, 0.248USD/hour EBS(gp3): 300GB, 0.08USD/GB/month)	615.12 (0.248 x 3 x 730 + 300 x 3 x 0.08)
EC2 インスタンス (Logging) (m5.xlarge x2, 0.248USD/hour EBS(gp3): 150GB, 0.08USD/GB/month)	386.08 (0.248 x 2 x 730 + 150 x 2 x 0.08)
S3 (1,500GB/month のログサイズ(1MB/1リクエスト)を想定 書き込み/保存時のログ圧縮率0.4を想定 書き込み料金 $0.0047/1,000リクエスト 保存料金: 0.025USD/GB/month)	43.2 (0.047 x 600 + 0.025 x 600)
ROSA利用時の想定合計金額(USD): 1,851.05	

ここまでのまとめ: ECS/EKS/ROSAのどれを使うべきか??

- アプリ開発や実行のために、様々なAWSネイティブのサービスを活用したい
- Kubernetes (K8s) や関連するCNCFプロジェクトは使わない

ECS

- 最小限のインフラコストでKubernetes環境を利用したい
- K8sのセキュリティ設定に関するノウハウがある
- 選択したOSS、K8sアドオンやAWSサービスを合わせた運用体制を構築済み
  - 各サービスを組み合わせて商用レベルの開発・運用環境を構築可能
  - AWS IAMの設定やクラスター内の権限設定を実現可能
    - AWSによるIAM標準ポリシーのアップデートにも対応可能な体制がある
- コンテナを実行するためのミドルウェアのサポートが不要
  - 脆弱性対策やトラブルシューティングなどを自力で実施可能

EKS

- K8sレイヤやコンテナ開発/運用に必要なコンポーネントがパッケージングされた製品を活用したい
  - 基盤構築に必要な時間を短縮したい
  - セキュリティが強化されたK8sを利用したい
  - AWS IAMの設定やクラスター内の権限設定の手間を、最小限に抑えたい
  - 他サービスと統合されたコンソールを利用したい
- Red HatによるコンテナのOS(RHEL)やミドルウェアのサポートが必要
  - 脆弱性対策やトラブルシューティングなどの支援を依頼したい

ROSA

Red Hat

ROSA Hosted Control Plane サービス仕様

Red Hat

ROSAの契約形態

ROSAは、AWSの請求書にまとめたいお客様を想定した、従量課金のサービス

	ROSA
処理主体 (Transacted)	AWS
請求元 (Billed by) ※1	AWS
契約条件 (Contract terms)	AWS + Red Hat
マネージド管理主体 (Managed by) ※2	AWS + Red Hat
サポート作業主体 (Supported by) ※3	AWS + Red Hat

※1: 請求元が発行する請求書には、OpenShiftサービスの利用料金や、仮想マシンやストレージなどのクラウドサービスの利用料金が記載されます。
※2: ロギング、モニタリング、プラットフォームのアップグレード、セキュリティ、などを担当する主体者。
※3: インストール、利用方法、設定、問題診断、バグ解決などに関して、チャット、電話、メールなどでの問い合わせ対応を実施する主体者。

Red Hat

ROSA overview of responsibility assignment matrix

- AWSがIaaS基盤、Red HatがOpenShiftを管理
- お客様は、アプリ、アプリの{データ, ロギング, ネットワークポリシー}、VPN/VPC接続、クラスターのプライベートネットワークなどを管理
- 詳細: https://docs.openshift.com/rosa/rosa_architecture/rosa_policy_service_definition/rosa-policy-responsibility-matrix.html

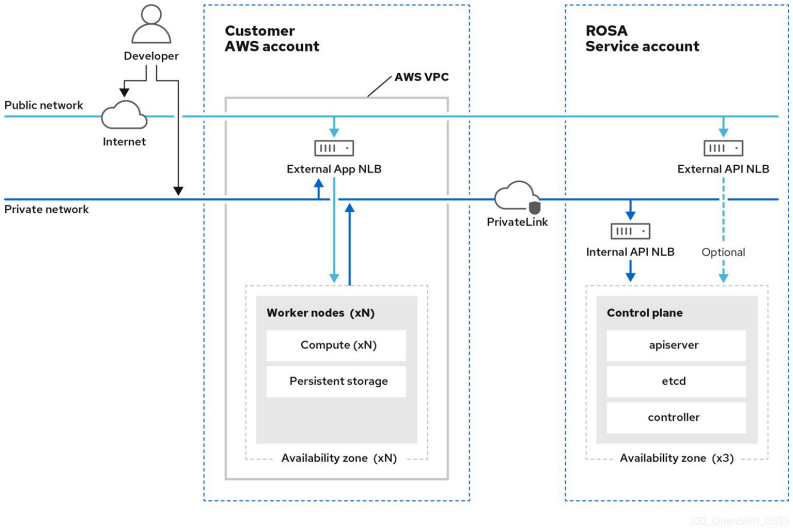
Resource	Incident and operations management	Change management	Identity and access management	Security and regulation compliance	Disaster recovery
Customer Data	Customer				
Customer applications	Customer				
Developer services	Customer				
Platform monitoring	Red Hat				
Logging	Red Hat	Shared			Red Hat
Application networking	Shared			Red Hat	
Cluster networking	Red Hat	Shared		Red Hat	
Virtual networking	Shared				
Master and infrastructure nodes	Red Hat				
Worker nodes	Red Hat				
Cluster version	Red Hat	Shared	Red Hat		
Capacity management	Red Hat	Shared	Red Hat		
Virtual storage	Red Hat				
Physical infrastructure and security	AWS				

ROSA Hosted Control Plane (HCP) の主なサービス仕様

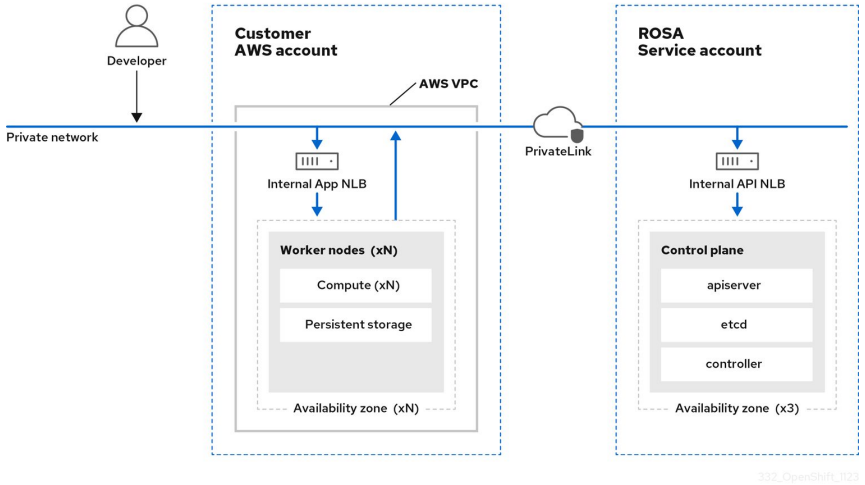
Specification	ROSA
デフォルトの アーキテクチャ (SingleAZ/MultiAZ共通)	Worker Node: m5.xlarge (4vCPU/RAM 16GiB) x2 (最小は2台)
Max Worker Nodes	500
対応リージョン	東京&大阪リージョン、他多数
プライベートクラスター	対応 (ROSA) AWS VPC peering, AWS VPN, AWS Direct Connect が利用可能
IDプロバイダー (認証)	GitHub, GitHub Enterprise, GitLab, Google, LDAP, OpenID Connect, HTPasswd (HTPasswdはROSAクラスター管理者用途としてのみサポート)
AWSサービスとの連携	Amazon EBS/EFS(ストレージ機能), CloudWatch (ログ転送) AWS Controllers for Kubernetes (ACK) によるAWS SDKを利用したサービスデプロイが可能
アップデート作業	Red HatによるOpenShiftの自動アップデート (お客様による、アップデート時間の設定が可能)
SLA	99.95%

ROSA HCPクラスターのアーキテクチャ

- EKSと同様に、ユーザーアカウントにはワーカーノードしか見えない状態のクラスター
- ROSA Service Accountは、Red Hat SREチームが管理
- AWS ファイアウォールによる送信トラフィックの制御 が可能



パブリッククラスター

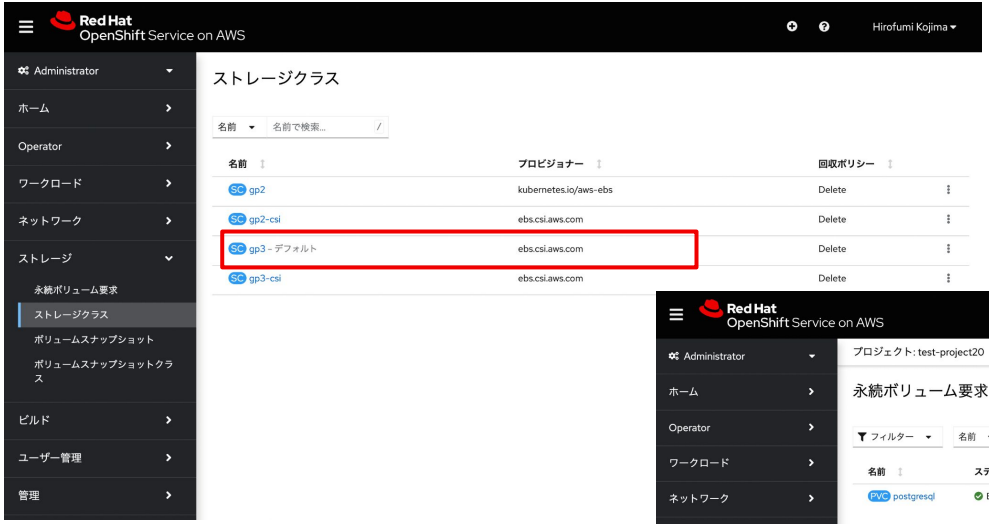


プライベートクラスター

AWSサービスとの連携

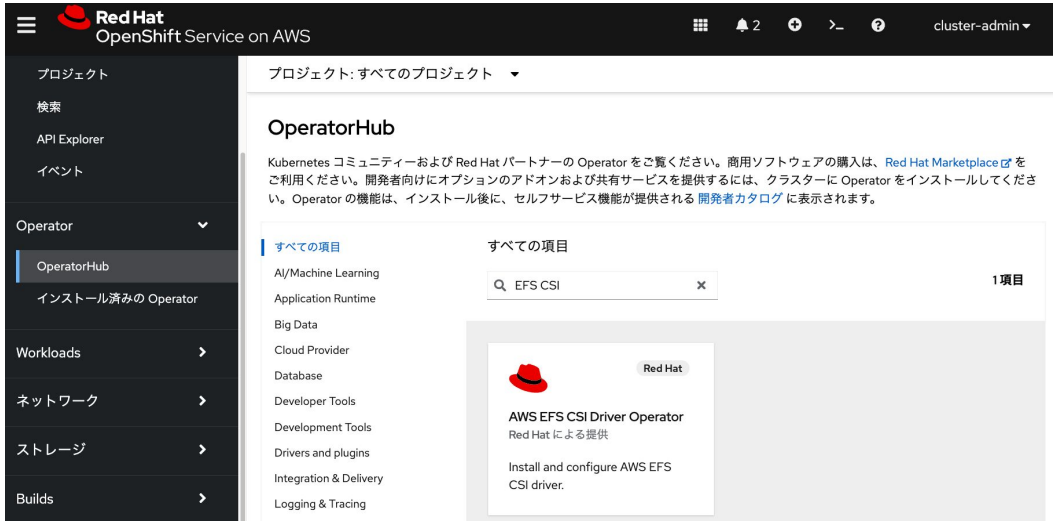
ROSAクラスターでのAmazon EBSの利用

- ROSAクラスターでは、EBSを利用するためのストレージクラスが予め設定されており、追加設定することなく、Pod用の永続ボリュームとして利用可能



ROSAクラスターのAmazon EFSの利用

- ROSAクラスターの複数のワーカーノードから同時に読み書き可能な永続ボリュームとして、 EFSを利用するように設定可能
- EFSを利用するための、ファイルシステムとアクセスポイントを事前作成
- ROSAクラスターにある「AWS EFS CSI Driver Operator」を利用して、EFSを利用するためのストレージドライバーをインストール
- EFSのストレージドライバーを利用するストレージクラスを作成して、
  Podが利用するイメージ (NFS v4プロトコルによるアクセス)

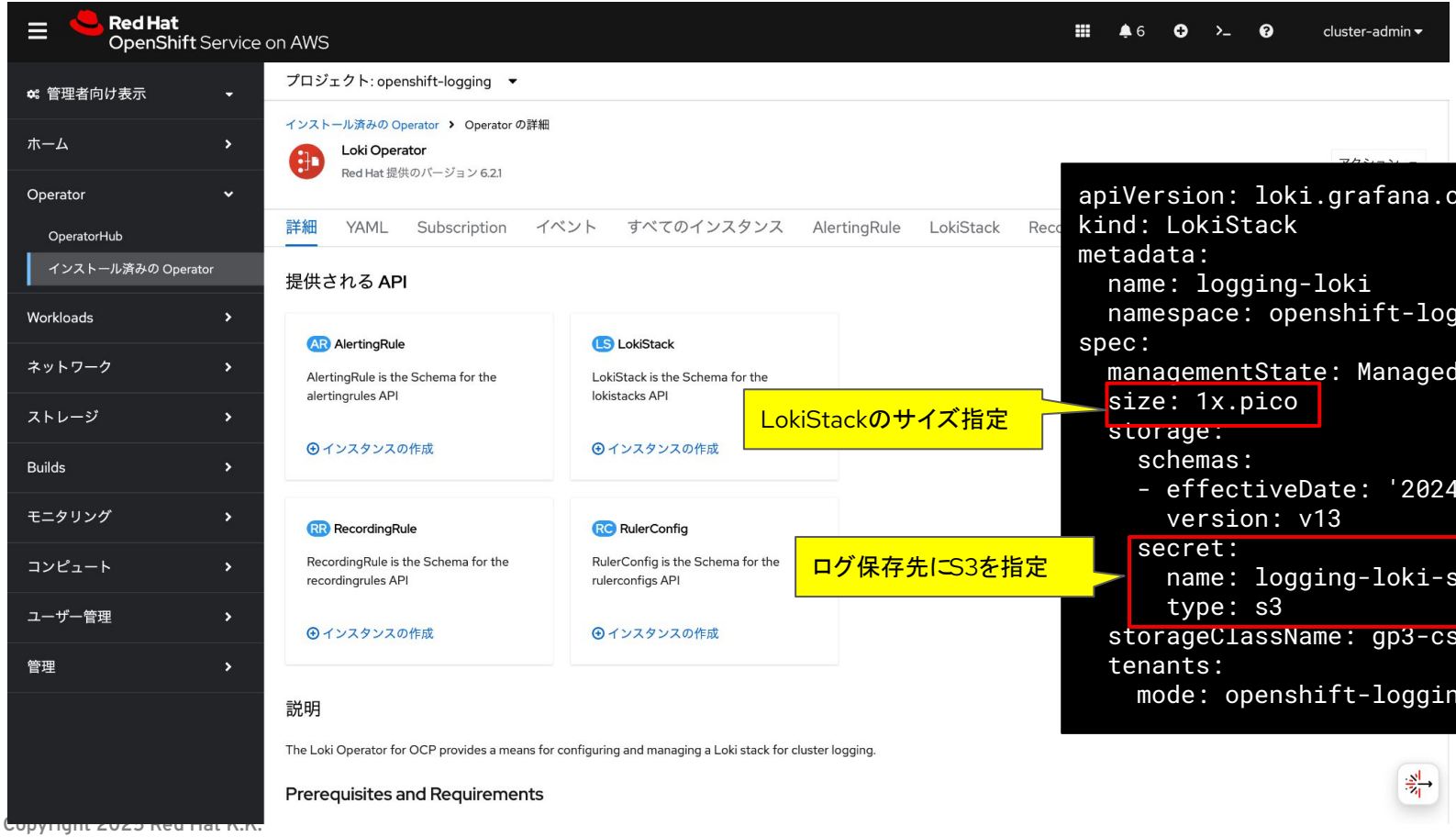


ROSAクラスターのロギング

- OpenShift LoggingやAmazon CloudWatchでの下記ログ転送/集約が可能
  - ユーザーアプリケーションのログ
  - インフラストラクチャー関連のログ
    - openshift-*,kube-*などのプロジェクトにあるログ
  - ワーカーノードのセキュリティ監査に関連するログ(audit.log)
    - コントロールプレーンは、OAuth関連のログのみCloudWatchに転送可能
    - コントロールプレーンの監査ログは Red HatのSREチームによって、別途収集され、ユーザーリクエストに応じて随時提供
- OpenShift Loggingでのログ集約の設定方法の概要
  - Lokiのログ保存用のS3をAWS STSで利用するためのIAMロールを作成
  - ROSAクラスターでOpenShift LoggingなどのOperatorをインストールして、vectorによるLokiStackへのログ転送設定を作成

Red Hat

ROSAクラスターのロギング設定例 (Loki)

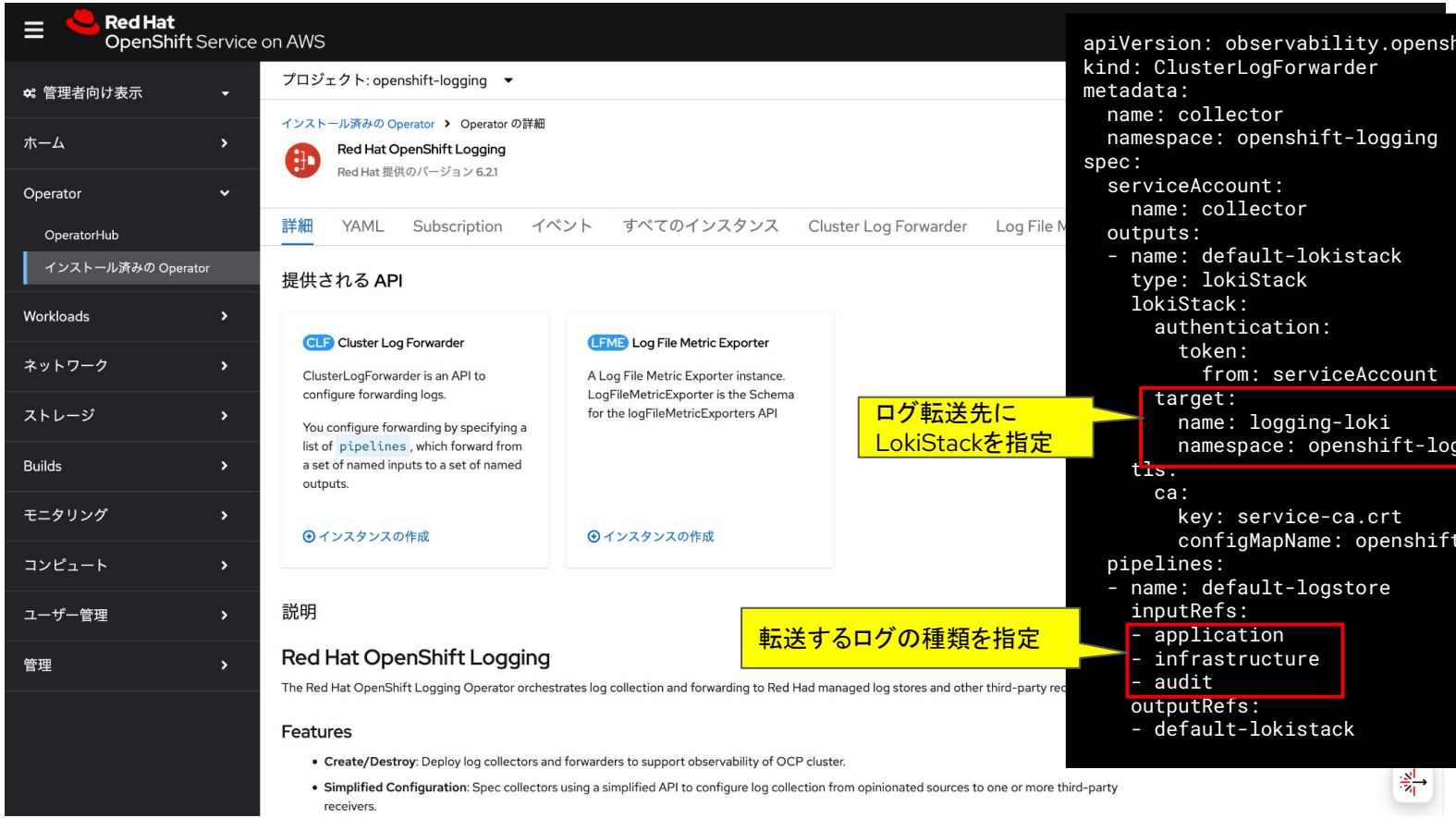



```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: logging-loki
  namespace: openshift-logging
spec:
  managementState: Managed
  size: 1x.pico
  storage:
    schemas:
    - effectiveDate: '2024-10-01'
      version: v13
    secret:
      name: logging-loki-s3
      type: s3
  storageClassName: gp3-csi
  tenants:
    mode: openshift-logging
```

LokiStackのサイズ指定

ログ保存先にS3を指定

ROSAクラスターのロギング設定例 (vectorによるログ転送)




```yaml
apiVersion: observability.openshift.io/v1
kind: ClusterLogForwarder
metadata:
  name: collector
  namespace: openshift-logging
spec:
  serviceAccount:
    name: collector
  outputs:
  - name: default-lokistack
    type: lokiStack
    lokiStack:
      authentication:
        token:
          from: serviceAccount
      target:
        name: logging-loki
        namespace: openshift-logging
    tls:
      ca:
        key: service-ca.crt
        configMapName: openshift-service-ca.crt
  pipelines:
  - name: default-logstore
    inputRefs:
    - application
    - infrastructure
    - audit
    outputRefs:
    - default-lokistack
```

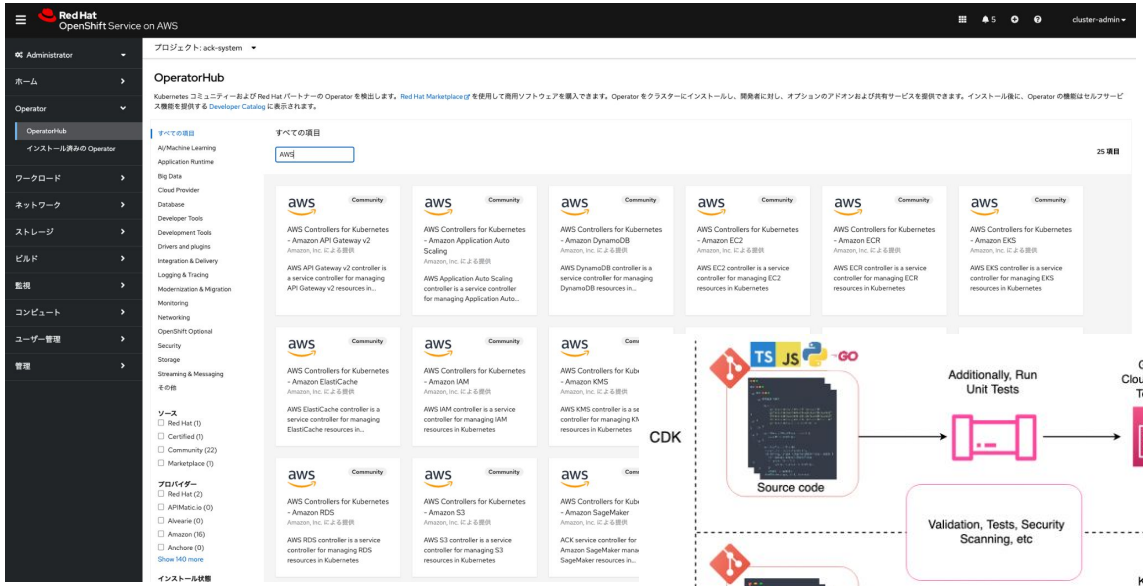
ログ転送先に
LokiStackを指定

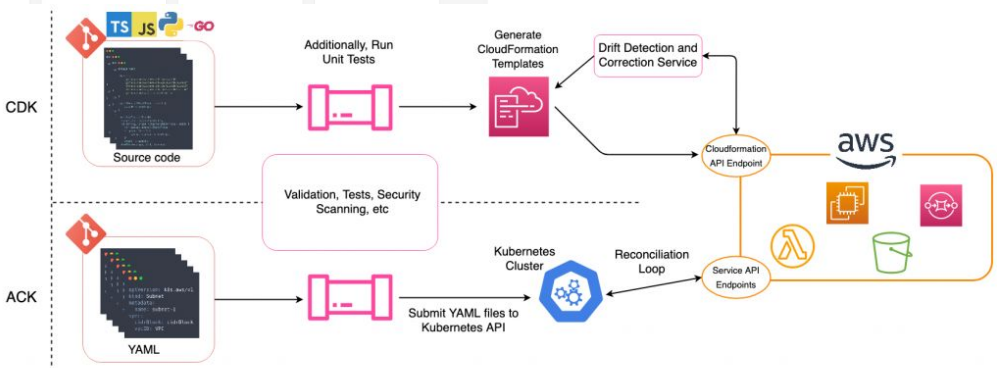
転送するログの種類を指定

Red Hat

ACKによるAWSサービスのデプロイ

- さまざまなAWSサービスの作成/削除が可能



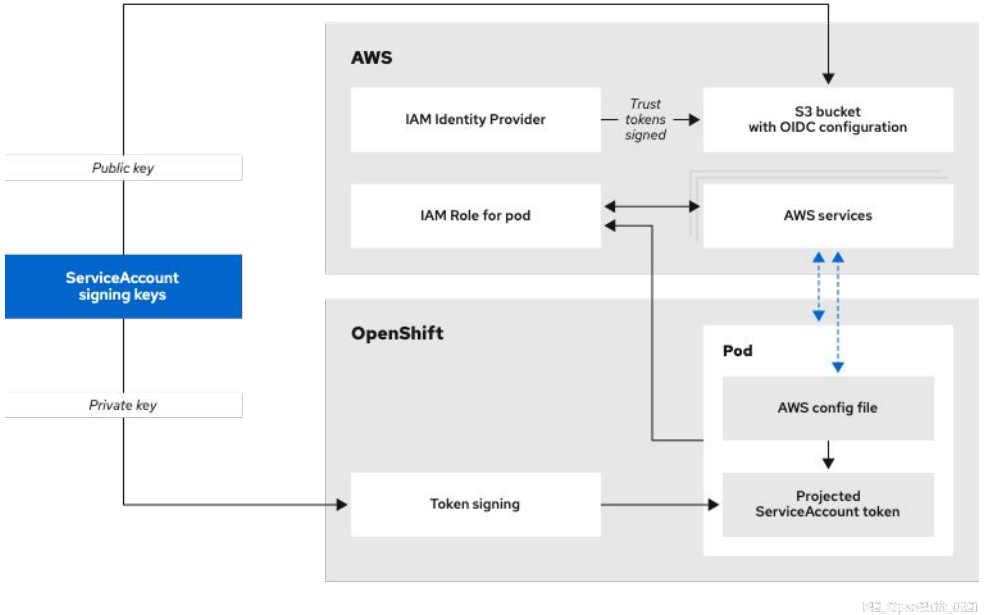


引用元: [Comparing AWS Cloud Development Kit and AWS Controllers for Kubernetes]

ROSA上のアプリケーションとAWSサービスとの連携

https://aws.amazon.com/jp/blogs/containers/fine-grained-iam-roles-for-red-hat-openshift-service-on-aws-rosa-workloads-with-sts/

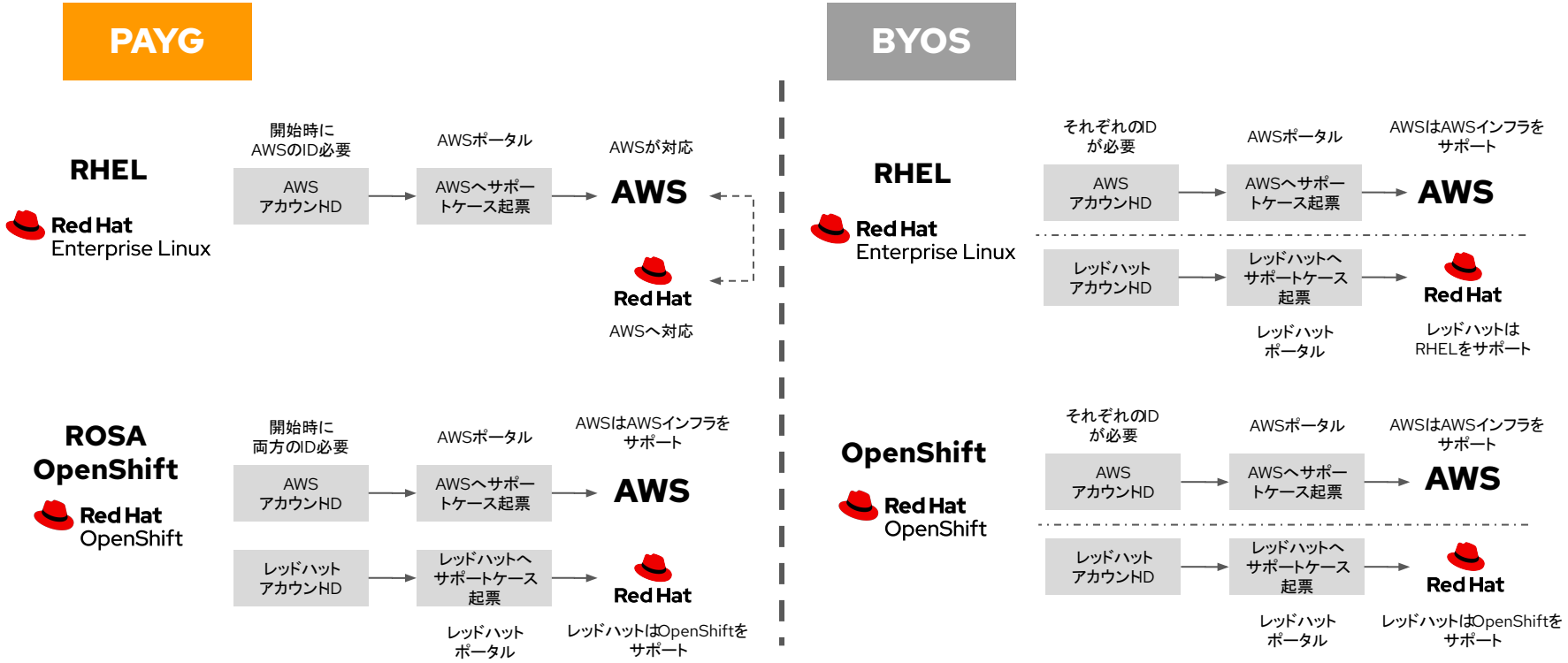
- AWS STSを使った、一時的な認証設定が可能
- OpenShiftの各Podは、サービスアカウントのトークン(システムコンポーネント間で利用される認証情報)により、AWSサービスを呼び出すことが可能
  - 参考: サービスアカウントのIAMロール (AWS公式ドキュメント)



サポート体制

Red Hat

PAYGとBYOSのサポート体制



Appendix:
AWSに持ち込むRed Hat製品
Bring your Own Subscription (BYOS)

RHEL/OpenShiftのサブスクリプション情報

- RHELのサブスクリプションガイド
  - AWSに持ち込む場合、実行する仮想インスタンス数を 2で割った数が、RHELのサブスクリプションの必要本数（vCPUの数は関係ありません）
  - RHELのゴールドイメージ(BYOS用のEC2プライベートイメージとしてRed Hatが提供)を利用する場合、Cloud Accessという登録手続きが必要
    https://access.redhat.com/ja/articles/5855161
  - 「RHEL サブスクリプションガイド」で検索
    https://www.redhat.com/ja/resources/red-hat-enterprise-linux-subscription-guide
- Self-Managed OpenShiftのサブスクリプションガイド
  - AWSに持ち込む場合、OpenShiftワーカーノードの vCPU数の総数を 4で割った数が、OpenShiftのサブスクリプションの必要本数（2コアまたは 4vCPU単位）
  - 「OpenShift サブスクリプションガイド」で検索
    https://www.redhat.com/ja/resources/self-managed-openshift-sizing-subscription-guide
- 参考: BYOSからPAYG (Pay-As-You-Go)への変換
  - RHEL: PAYG RHELへのデータ移行ガイド
  - Self-Managed OpenShift:
    - AWS Marketplaceから購入したPAYGのSelf-Managed OpenShiftに変換可能
      (ワーカーノードで利用するAMIのIDを従量課金専用のAMI IDに変更)

Red Hat

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat