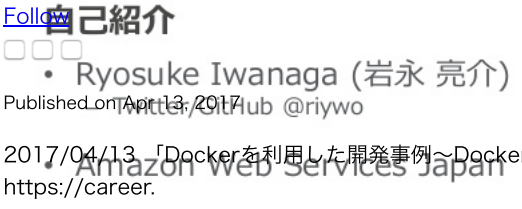
AWSでDockerを扱うためのベストプラクティス

8,781 views

- Share
- Like
- Download
- ...

Amazon Web Services Japan
Follow



Published on Apr 13, 2017

2017/04/13「Dockerを利用した開発事例〜Docker導入から運用まで〜」での発表資料です。
https://career.

...

Published in: Internet

- 1 Comment
- 17 Likes
- Statistics
- Notes

- Full Name
  Comment goes here.
  12 hours ago Delete Reply Block
  Are you sure you want to Yes No
  Your message goes here

Share your thoughts...
Post

- katekoxx
  Hey guys! Who wants to chat with me? More photos with me here 👉 http://www.bit.ly/katekoxx
  4 months ago Reply
  Are you sure you want to Yes No
  Your message goes here

- Kotaro Inoue , Machine Learning Engineer at Metadata Incorporated at Metadata Incorporated
  1 month ago

- Ken Takehara , RICOH - specialist at RICOH

AWSでDockerを扱うためのベストプラクティス

1. 1. 1 AWSでDockerを扱うための ベストプラクティス Ryosuke Iwanaga Amazon Web Services Japan Solutions Architect
2. 2. 2 自己紹介・Ryosuke Iwanaga (岩永 亮介) – Twitter/GitHub @riywo・Amazon Web Services Japan・Solutions Architect – Web, Gaming – Big Data / DevOps / Container・Before Amazon – Software Engineer / Ops Engineer / DBA / etc.
3. 3. 3 Agenda・Dockerで開発するとは？・Dockerを運用するとは？・Amazon ECSの事例
4. 4. 4 Dockerで開発するとは？
5. 5. 5 コンテナは新しい？・古い技術 – これまでにもたくさんのコンテナ技術が生まれている・リソースを隔離するのが主目的 – CPU、メモリ、ファイルシステム、等 – OS上での仮想化技術・DevOpsの文脈での再発見 – Dockerが注目を浴びている理由: CLI, Engine, Registry Server Guest OS Bins/Libs Bins/Libs App2App1
6. 6. 6 Dockerをサービスで利用するには？・サクサクコマンドラインで使うには便利だが、、、 – 本番サービスで使おうとすると、様々なハマりポイントが存在・VMやサーバとは違う勘所が必要になる – インフラより、むしろアプリケーションの作りが異なってくる・Transformationが必要 – 開発を変化させるときに、どこに向かうべきか？ – → 12-factor App
7. 7. 7
8. 8. 8 12-factor App・Herokuのエンジニアが2011年に提唱 – 当時からコンテナを使いこなしているPaaS (Docker登場は2013年)・このドキュメントの対象者 – "サービスとして動くアプリケーションを開発しているすべての開発 者。およびそのようなアプリケーションをデプロイまたは管理してい るインフラエンジニア" – → すなわち全員(今時サービスに関わらないエンジニアはいない) https://12factor.net/ja/
9. 9. 9 12-factor Appとは何か？・サービスを開発する上で、従うべき12の要素 – これらに従うことで、モダンな環境に最適なアプリになる – 2011年の定義だが、今でも変わらないベストプラクティス – 特に、コンテナでアプリを動かす上では必須となる要素・Dockerが使いにくいと感じたら？ – それはアプリが12-factor Appになっていないことがほとんど – どうすればアプリを12-factor Appにできるかをまず考える – 12-factor Appを逸脱するなら、明確な理由を
10. 10. 10 Twelve-Factor I. コードベース バージョン管理される1つのコードベースと複数デプロイ II. 依存関係 依存関係を明示的に宣言し分離する III.設定 設定を環境変数に格納する IV.バックエンドサービス バックエンドサービスをアタッチされたリソースとして扱う V. ビルド、リリース、実行 ビルド、リリース、実行の3つのステージを厳密に分離する VI.プロセス アプリを1つ又は複数のステートレスなプロセスとして実行 VII.ポートバインディング ポートバインディングを通してサービスを公開する VIII.並行性 プロセスモデルによってスケールアウトする IX. 廃棄容易性 高速な起動とグレースフル停止で堅牢性を最大化する X. 開発/本番一致 開発、ステージング、本番環境をできるだけ一致させた状態を保つ XI. ログ ログをイベントストリームとして扱う XII.管理プロセス 管理タスクを1回限りのプロセスとして実行する

11. 11 特におさえておきたいポイント ・設定とビルドは分離する – 設定値はイメージに入れず、環境変数で実行時に外から注入 ・データベースの様なステートフルな要素は、コンテナを使 わない – ポータビリティが低下してしまい、コンテナのメリットが損なわれる – データベースをサービスとして使う(例: Amazon RDS) ・ログはファイルに書かず、ストリームですぐに外に出す – ステートレスにするためにログはすぐに外に飛ばす – STDOUT/STDERRに出力するだけでよい(例: Amazon CloudWatch Logs)
12. 12 Dockerで運用するとは?
13. 13 DevOpsの実態 Build Test ProductionSource Application Artifact コードだけ 書いていればいい?
14. 14 DevOpsの実態 Build Test ProductionSource Application Artifact Provision Config 開発環境の構成の メンテナンスが必要 開発、テスト、本番で 環境に差異がある。。。 テストの需要がバラバラ で管理が大変。。。。 オートスケールや ノード障害対応。。。 なるほど、 全てが必要なんですね。。。
15. 15 コンテナを取り入れたDevOps Build Test ProductionSource Application Image Provision Config コードだけ書いて いればいい!
16. 16 本番環境でDockerを動かす ・Dockerを動かすだけなら簡単 – $ docker run [ENTER] ・問題は、どこでどうやってDockerを動かすか? – sshして実行? クロントからどうやってルーティングする? フェイ ルオーバーするには? スケールするには? ・クラスタ管理の仕組みが必要 1. インスタンス群の状態を管理 2. どこでどのコンテナを動かすかを決めて実行 3. コンテナ群の状態を管理
17. 17 自前でクラスタ管理すると、、、 Masters State Data Store WorkersSchedulers Load Balancers 冗長化・チューニング・運用要件対応・冗長化・運用 安全な付け外し・デプロイ対応 冗長化・チューニング・運用
18. 18 Amazon EC2 Container Service (ECS) ・Amazon EC2インスタンス群をDocker実行環境に 簡単に変身させる – ECS Agentをインストールするだけ、管理インスタンスは不要 – ECS自体の料金は無料 (EC2等、他の利用しているリソース課金) ・1コンテナから数万コンテナ以上を管理 – 開発環境から本番環境までこれ1つで十分 ・AWSの他のサービスとの深い連携 – 例: AWS CloudFormationでデプロイ可能、ALB連携、etc.
19. 19 Amazon EC2 Container Service Scheduler ManagerCluster Task Definition Task Agent
20. 20 Service: 動的ポートマッピング Service scheduler Elastic Load Balancing Application Load Balancer Task Definition = app:1 Desired Count = 4 Amazon ECS 32874 32879 32873 32880 Cluster
21. 21 Service: 追加リソース無しの更新 Service scheduler Task Definition = app:2 Desired Count = 4 Minimum Healthy Percent = 50 Maximum Percent = 100 Elastic Load Balancing Application Load Balancer ClusterAmazon ECS
22. 22 Service: 追加リソース無しの更新Service scheduler Elastic Load Balancing Application Load Balancer Task Definition = app:2 Desired Count = 4 Minimum Healthy Percent = 50 Maximum Percent = 100 ClusterAmazon ECS
23. 23 Service: 追加リソース無しの更新 Service scheduler Elastic Load Balancing Application Load Balancer Task Definition = app:2 Desired Count = 4 Minimum Healthy Percent = 50 Maximum Percent = 100 ClusterAmazon ECS
24. 24 Service: 追加リソース無しの更新 Service scheduler Elastic Load Balancing Application Load Balancer Task Definition = app:2 Desired Count = 4 Minimum Healthy Percent = 50 Maximum Percent = 100 ClusterAmazon ECS
25. 25 Service: 追加リソース無しの更新 Service scheduler Elastic Load Balancing Application Load Balancer Task Definition = app:2 Desired Count = 4 Minimum Healthy Percent = 50 Maximum Percent = 100 ClusterAmazon ECS
26. 26 AWS Batchでジョブスケジューリング ・バッチジョブ実行スケジューラを提供 – 機械学習、数値計算、レンダリング等 – 定期実行用途ではない ・利用者はJobをQueueにSubmitする だけ – AWS BatchがAmazon ECSにスケ ジュールしてくれる ・実行環境はManagedで利用可能 – EC2のスケールは全てAWS Batchが面倒 をみてくれる AWS Batch Amazon ECSOn demand Spot Jobs Queues
27. 27 Amazon ECSの事例
28. 28 Amazon ECSへの適応が着実に進んでいる ・3 clear trends in ECS adoption – Monitoring SaaSのDatadogから2016年に出されたレポート ・Trends 1. ECSは静かに、しかし着実に勢いをつけている ・右肩上がりで利用ユーザ割合が増えている(2015/10: 3%→2016/10: 15%) 2. コンテナ数が増えると、問題が多くなる ・Dockerのホスト数が25台を超えると、ECSの利用は40%近くに 3. オーケストレーションによって成長が生まれる ・ECSを導入すると、コンテナの数が35%も増加する https://www.datadoghq.com/blog/3-clear-trends-in-ecs-adoption/
29. 29 Amazon ECSを利用されている事例の一部
30. 30
31. 31 C4 R3 M4R3 R3 R3 R3 R3 M4 M4 M4 M4 M4 C4 C4 C4 C4 C4 Map Service Search Service Directions Service
32. 32 C4 ECS Cluster R3 M4R3 R3 R3 R3 R3 M4 M4 M4 M4 M4 C4 C4 C4 C4 C4 Map Service Search ServiceDirections Service
33. 33
34. 34 C4 ECS Cluster R3 R3 R3 R3 R3 R3 M4 M4 M4 M4 M4 M4 C4 C4 C4 C4 C4 Map Service Search ServiceDirections Service Spot Fleet C4 C4 R3 R3
35. 35 21 Services 2000 Tasks 1.3 billion Requests per day
36. 36 25% Fewer instances 80-90% Savings per month on EC2
37. 37 まとめ ・Dockerはとても便利で、これからのトレンド ・今まで(VMベース)の考え方からTransformする必要がある – 12-factor Appでアプリ・システムを設計すること ・AWSでDockerを使うなら、Amazon ECS – 豊富な事例・とても柔軟な運用 – バッチ用途ならAWS Batchも ・何か気になるところがあれば、お気軽にお問い合わせ下さい
38. 38

Recommended

7

12-factor App

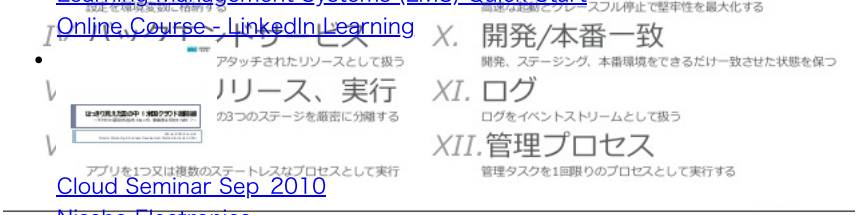
Teaching Techniques: Creating Multimedia Learning
Online Course - LinkedIn Learning

https://12factor.net/ja/

8

Learning Online Marketing
Online Course - LinkedIn Learning

Learning Management Systems (LMS) Quick Start
Online Course - LinkedIn Learning



Cloud Seminar Sep 2010
Nissho Electronics

Dockerの事例紹介
Hiroki Endo

Big DataとContainerとStream - AWSでのクラスタ構成とストリーム処理 -
Amazon Web Services Japan

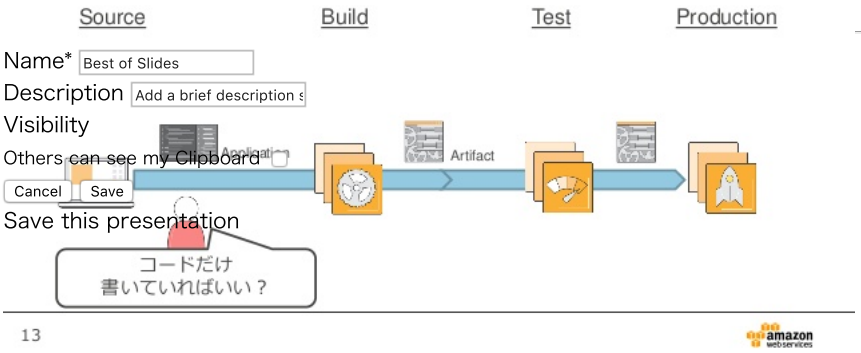
Docker基礎+docker0.9, 0.10概要
Kazuyuki Mori

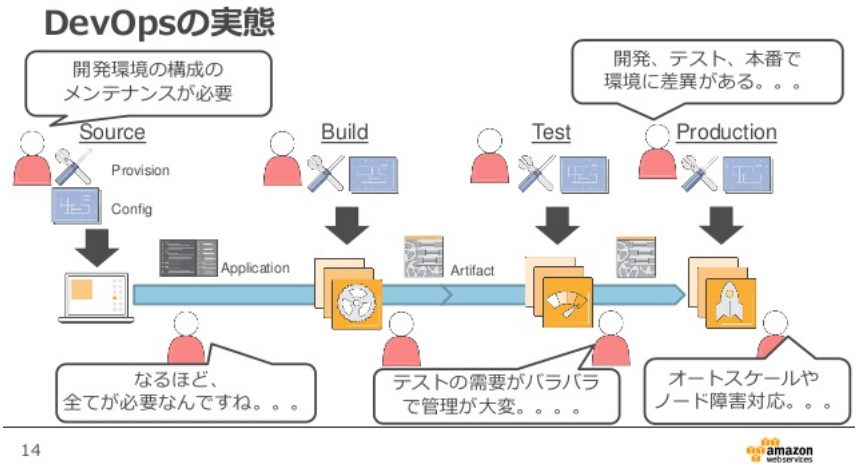
AWS Black Belt Online Seminar 2016 Amazon EC2 Container Service
Amazon Web Services Japan

Dockerを活用したリクルートグループ開発基盤の構築
Recruit Technologies

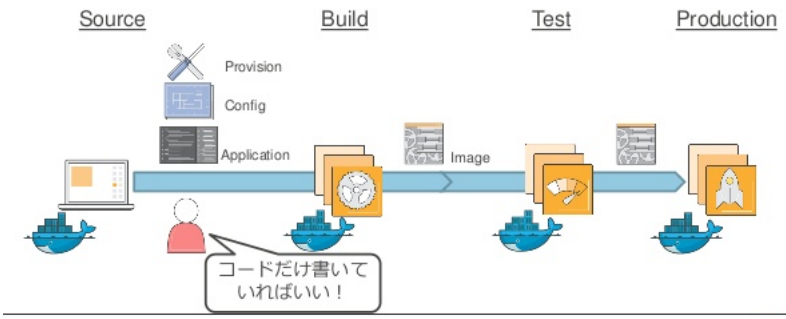
ておきたいポイント

- 設定とビルドは分離する
- 設定値はイメージに入れず、環境変数で実行時に外から注入
- データベースの様なステートフルな要素は、コンテナを使わない
- ポータビリティが低下してしまい、コンテナのメリットが損なわれる
- データベースをサービスとして使う(例: Amazon RDS)
- ログはファイルに書かず、ストリームですぐに外に出す
- ステートレスにするためにログはすぐに外に飛ばす
- STDOUT/STDERRに出力するだけでよい(例: Amazon CloudWatch Logs)





コンテナを取り入れたDevOps

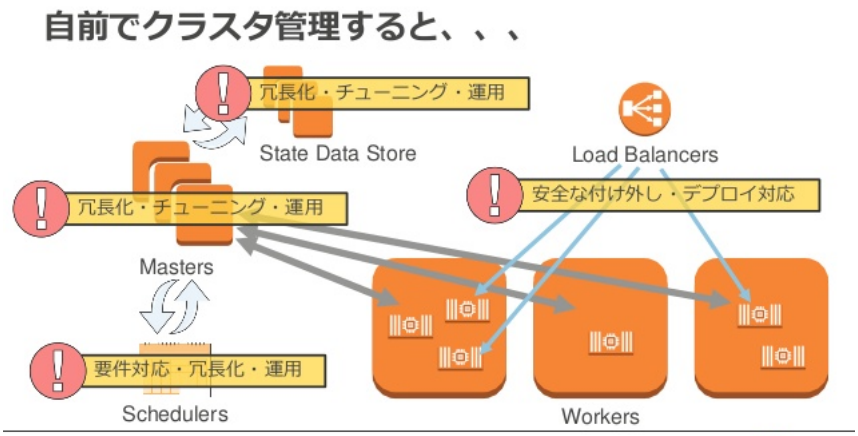


15

本番環境でDockerを動かす

- Dockerを動かすだけなら簡単
  - $ docker run [ENTER]
- 問題は、どこでどうやってDockerを動かすか？
  - sshして実行？フロントからどうやってルーティングする？フェイルオーバーするには？スケールするには？
- クラスタ管理の仕組みが必要
  1. インスタンス群の状態を管理
  2. どこでどのコンテナを動かすかを決めて実行
  3. コンテナ群の状態を管理

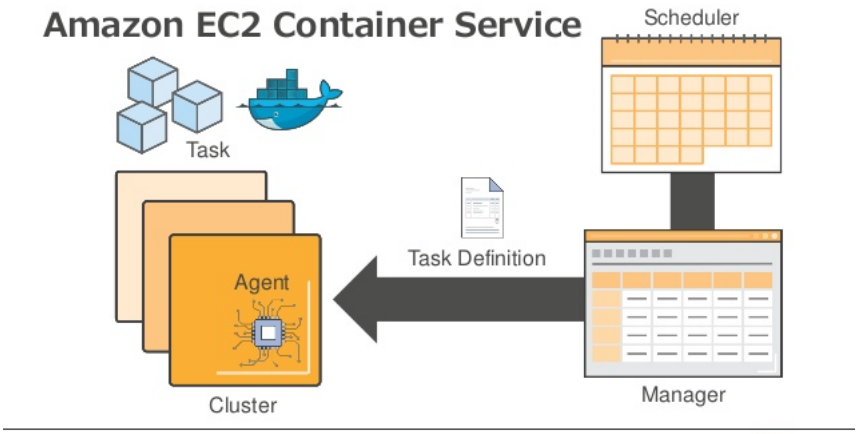
16

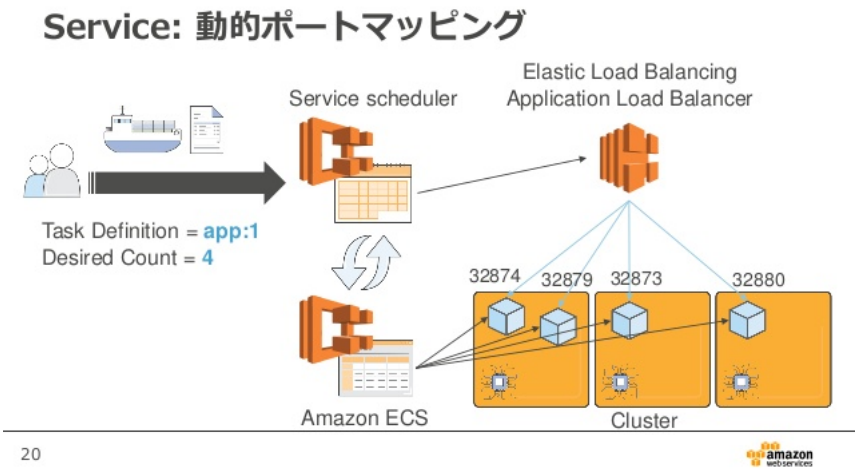


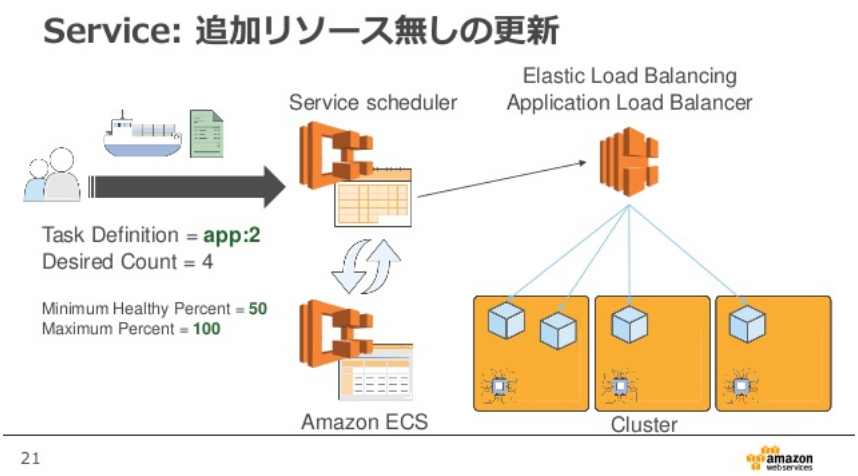
Amazon EC2 Container Service (ECS)

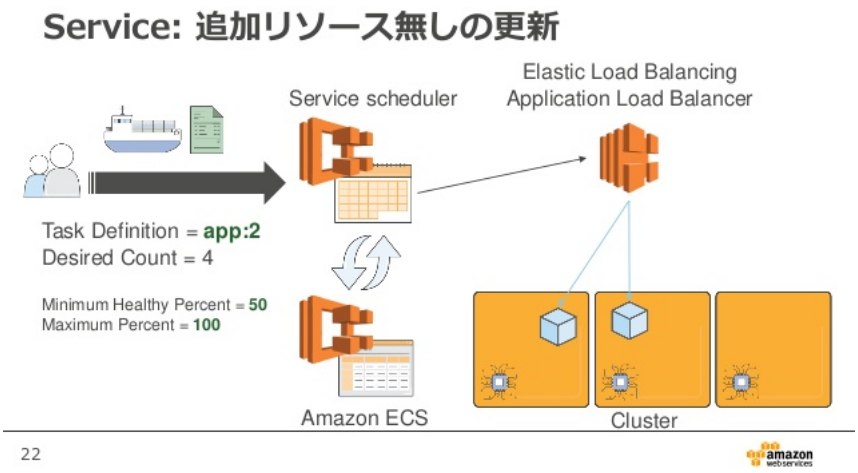
- Amazon EC2インスタンス群をDocker実行環境に簡単に変身させる
  - ECS Agentをインストールするだけ、管理インスタンスは不要
  - ECS自体の料金は無料 (EC2等、他の利用しているリソース課金)
- 1コンテナから数万コンテナ以上を管理
  - 開発環境から本番環境までこれ1つで十分
- AWSの他のサービスとの深い連携
  - 例: AWS CloudFormationでデプロイ可能、ALB連携、etc.

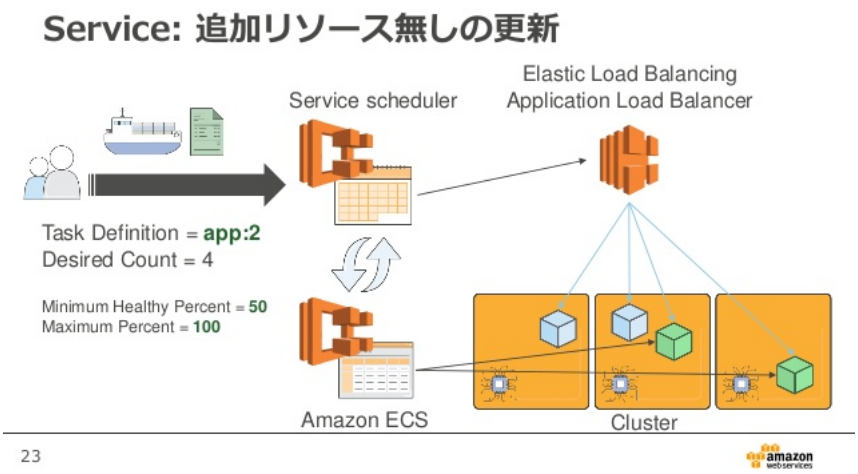
18 amazon web services

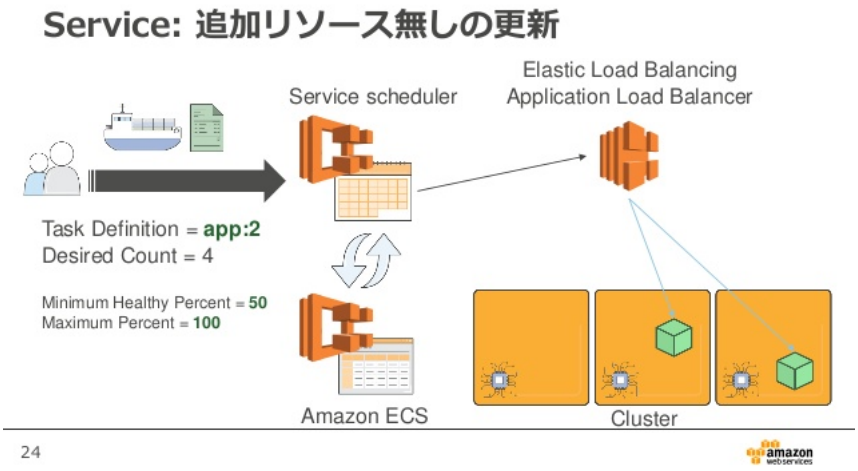

Amazon EC2 Container Service
Scheduler
Task
Task Definition
Agent
Cluster
Manager
19

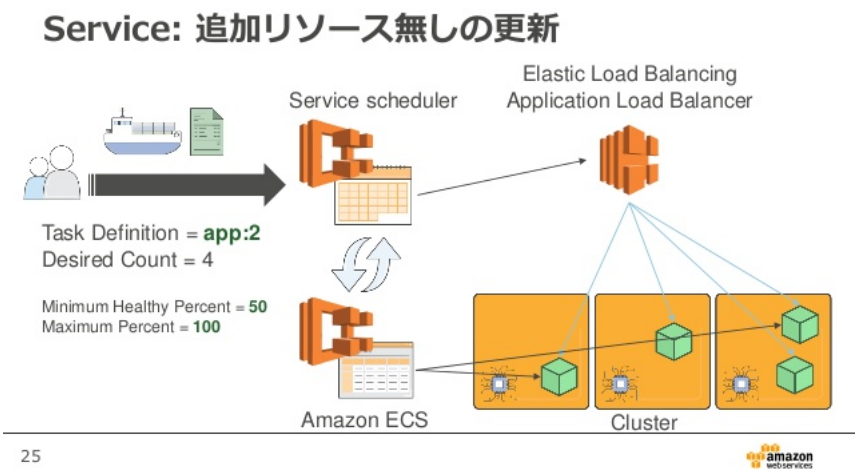

Service: 動的ポートマッピング
Elastic Load Balancing
Application Load Balancer
Service scheduler
Task Definition = app:1
Desired Count = 4
32874
32879
32873
32880
Amazon ECS
Cluster
20

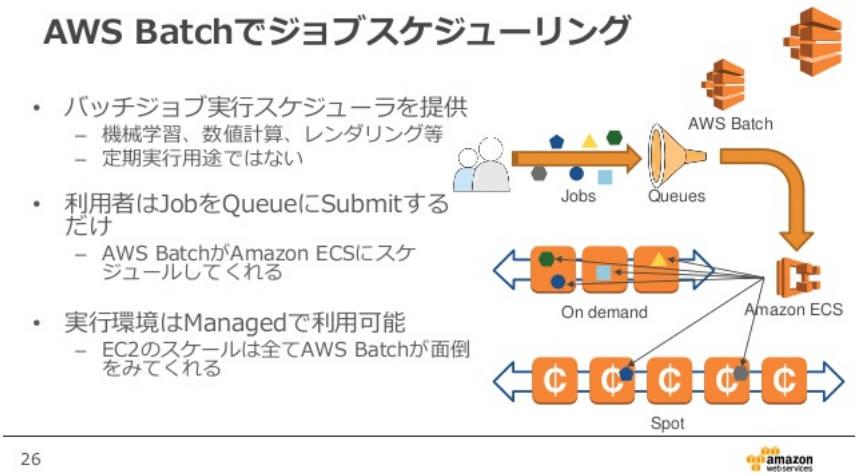

Service: 追加リソース無しの更新
Service scheduler
Elastic Load Balancing
Application Load Balancer
Task Definition = app:2
Desired Count = 4
Minimum Healthy Percent = 50
Maximum Percent = 100
Amazon ECS
Cluster
21
amazon web services


Service: 追加リソース無しの更新
Service scheduler
Elastic Load Balancing
Application Load Balancer
Task Definition = app:2
Desired Count = 4
Minimum Healthy Percent = 50
Maximum Percent = 100
Amazon ECS
Cluster
22
amazon web services


Service: 追加リソース無しの更新
Elastic Load Balancing
Application Load Balancer
Service scheduler
Task Definition = app:2
Desired Count = 4
Minimum Healthy Percent = 50
Maximum Percent = 100
Amazon ECS
Cluster
23
amazon web services


Service: 追加リソース無しの更新
Elastic Load Balancing
Application Load Balancer
Service scheduler
Task Definition = app:2
Desired Count = 4
Minimum Healthy Percent = 50
Maximum Percent = 100
Amazon ECS
Cluster
24
amazon web services


Service: 追加リソース無しの更新
Service scheduler
Elastic Load Balancing
Application Load Balancer
Task Definition = app:2
Desired Count = 4
Minimum Healthy Percent = 50
Maximum Percent = 100
Amazon ECS
Cluster
25


AWS Batchでジョブスケジューリング
• バッチジョブ実行スケジューラを提供
– 機械学習、数値計算、レンダリング等
– 定期実行用途ではない
• 利用者はJobをQueueにSubmitするだけ
– AWS BatchがAmazon ECSにスケジュールしてくれる
• 実行環境はManagedで利用可能
– EC2のスケールは全てAWS Batchが面倒をみてくれる
AWS Batch
Jobs
Queues
On demand
Amazon ECS
Spot
26

Amazon ECSの事例

27

Amazon ECSへの適応が着実に進んでいる

- 3 clear trends in ECS adoption
  - Monitoring SaaSのDatadogから2016年に出されたレポート
- Trends
  1. **ECSは静かに、しかし着実に勢いをつけている**
     - 右肩上がりで利用ユーザ割合が増えている(2015/10: 3%→2016/10: 15%)
  2. **コンテナ数が増えると、問題が多くなる**
     - Dockerのホスト数が25台を超えると、ECSの利用は40%近くに
  3. **オーケストレーションによって成長が生まれる**
     - ECSを導入すると、コンテナの数が35%も増加する

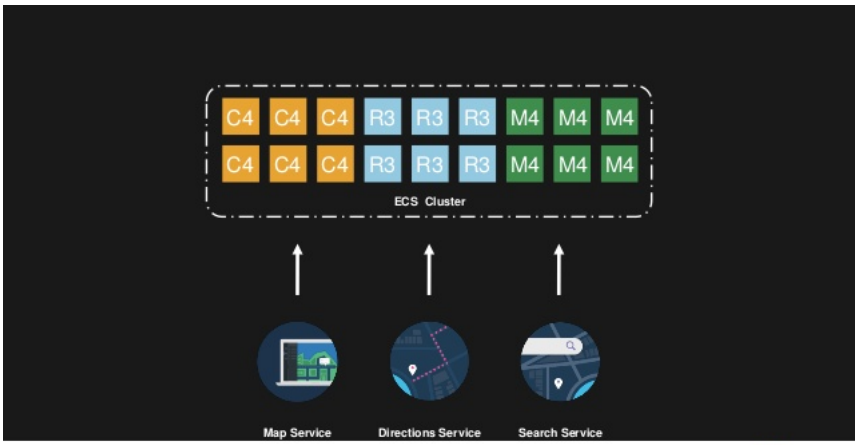
28 https://www.datadoghq.com/blog/3-clear-trends-in-ecs-adoption/


Amazon ECSを利用されている事例の一部
airtime
Atlassian
Aol.
Bb
Blackboard
edmunds.com
Expedia
FOURSQUARE
GILT
here
instacart
lyft
Meetup
mytaxi
okta
Riot Games
slack
SUPERCELL
Trimble
Upserve
29
amazon web services


MAPBOX
Maps
Developer tools
Directions
Geocoding
Analysis
Mobile
30
amazon web services


C4 C4 C4
C4 C4 C4
Map Service
R3 R3 R3
R3 R3 R3
Search Service
M4 M4 M4
M4 M4 M4
Directions Service

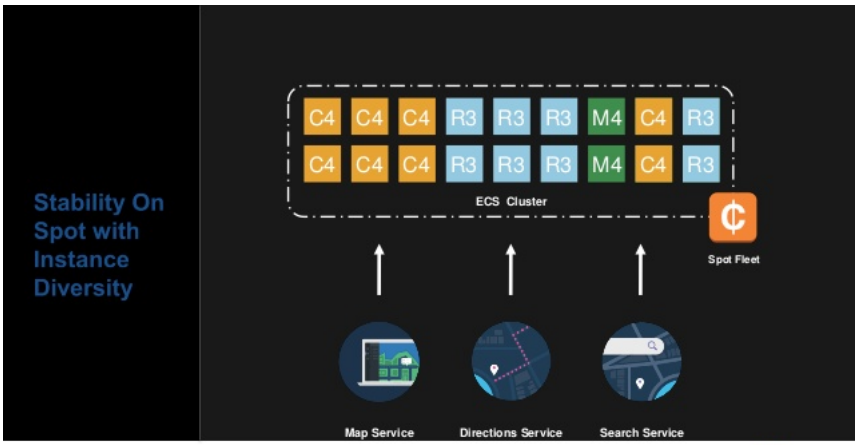
31


C4 C4 C4 R3 R3 R3 M4 M4 M4
C4 C4 C4 R3 R3 R3 M4 M4 M4
ECS Cluster
Map Service
Directions Service
Search Service

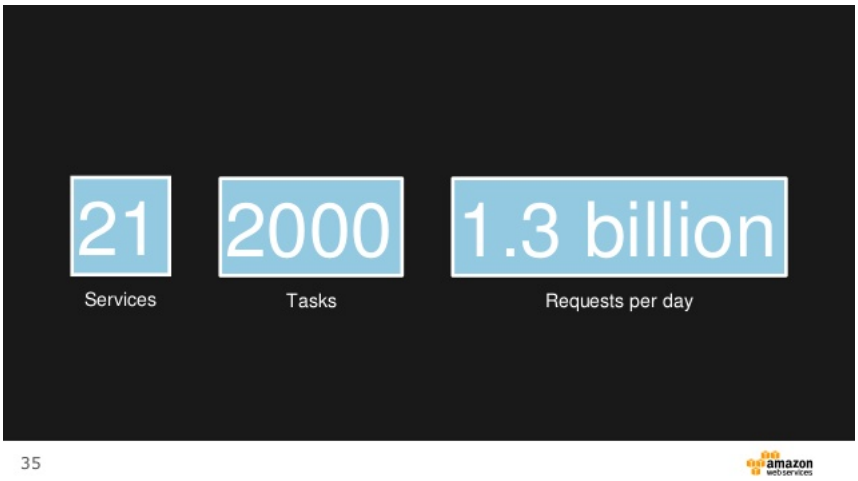
32

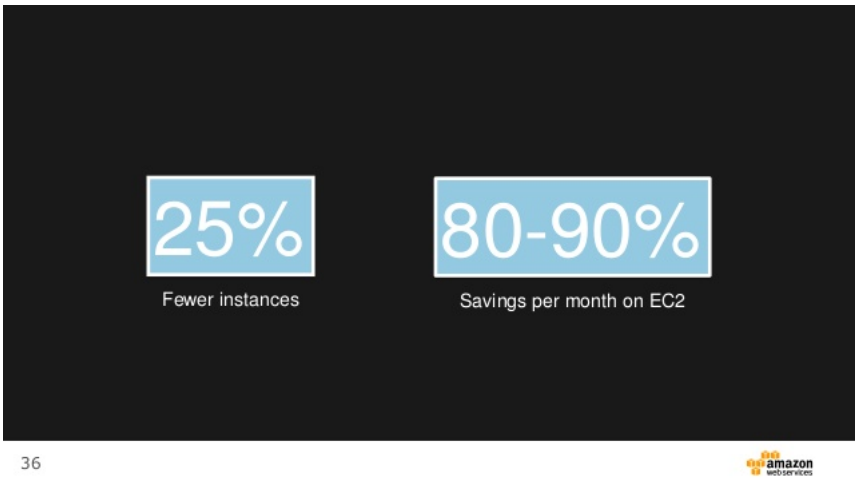

Reduce Waste with Better Instance Packing
Map service
CPU 55%
Mem 5%
Search service
CPU 25%
Mem 75%
Combined services with ECS
CPU 80%
Mem 80%

33


Stability On Spot with Instance Diversity
C4 C4 C4 R3 R3 R3 M4 C4 R3
C4 C4 C4 R3 R3 R3 M4 C4 R3
ECS Cluster
Spot Fleet
Map Service
Directions Service
Search Service

34


21
Services
2000
Tasks
1.3 billion
Requests per day
35
amazon web services


25%
Fewer instances
80-90%
Savings per month on EC2
36
amazon web services

37 まとめ・Dockerはとても便利で、これからのトレンド・今まで(VMベース)の考え方からTransformする必要がある – 12-factor Appでアプリ・システムを設計すること・AWSでDockerを使うなら、Amaz...

38

Upcoming SlideShare

Loading in …5

×