sysdig

Overview

本文の内容は、Overviewのドキュメント(https://docs.sysdig.com/en/overview.html)を元に日本語に翻訳・再構成した内容となっております。

Overview **8**

主な利点 8

Overviewユーザーインターフェイスへのアクセス 9

Overviewユーザーインターフェイスについて 9

Overview 行 10

スコープエディター 10

タイムナビゲーション 11

プリセット 11

範囲 12

時間形式 12

ライブ 12

イベントストリームの統合 13

イベントの種類 13

イベントのステータス 13

一般的なガイドライン 14

初めての使用 14

チューニングOverviewデータ 14

オンプレミスデプロイメントにおけるOverviewの有効化 14

機能フラグ 15

クラスターデータ **16**

クラスターのOverviewについて 16
クラスターデータの解釈 17
Node Ready Status 17
それは何ですか？ 17
何を期待します？ 17
他に何をすべきか？ 18
Pods Available vs Desired 18
それは何ですか？ 18
何を期待します？ 18
他に何をすべきか？ 19
CPU Requests vs Allocatable 19
それは何ですか？ 19
何を期待します？ 20
他に何をすべきか？ 20
値は100%を超えることができますか？ 21
Memory Requests vs Allocatable 21
それは何ですか？ 22
何を期待します？ 22
そうでなければ何をすべきか 22
値は100%より高くできますか？ 23
コンプライアンススコア 24
それは何ですか？ 24
何を期待しますか？ 24

他に何をすべきか？ 24

ノードデータ **25**

ノードのOverviewについて 25

ノードデータの解釈 25

ノード準備ステータス 25

それは何ですか？ 25

何を期待します？ 26

他に何をすべきか？ 26

Pods Ready vs Allocatable 27

それは何ですか？ 27

何を期待します？ 28

他に何をすべきか？ 28

CPU Requests vs Allocatable 29

それは何ですか？ 29

何を期待します？ 29

他に何をすべきか？ 29

値は100%を超えることができますか？ 30

Memory Requests vs Allocatable 30

それは何ですか？ 30

何を期待します？ 31

他に何をすべきか？ 31

値は100%より高くできますか？ 31

Network I/O 32

それは何ですか？ 32
何を期待します？ 32
ネームスペースデータ 33
ネームスペースのOverviewについて 33
ネームスペースデータを解釈する 33
ポッドの再起動 33
それは何ですか？ 33
何を期待します？ 34
他に何をすべきか？ 34
Pods Available vs Desired 35
それは何ですか？ 35
何を期待します？ 35
他に何をすべきか？ 36
CPU Used vs Requests 36
それは何ですか？ 36
何を期待します？ 37
他に何をすべきか？ 37
値は100%を超えることができますか？ 37
Memory Used vs Requests 38
それは何ですか？ 38
何を期待します？ 38
他に何をすべきか？ 38
値は100%を超えることができますか？ 39

Network I/O 39
それは何ですか？ 39
何を期待します？ 40
ワークロードデータ 41
ワークロードのOverviewについて 41
ワークロードデータの解釈 42
ポッドの再起動 42
それは何ですか？ 42
何を期待します？ 42
他に何をすべきか？ 42
Pods Available vs Desired 43
それは何ですか？ 43
何を期待します？ 44
他に何をすべきか？ 44
CPU Used vs Requests 45
それは何ですか？ 45
何を期待します？ 45
他に何をすべきか？ 45
値は100%を超えることができますか？ 46
「No Data」とはどういう意味ですか？ 46
Memory Used vs Requests 46
それは何ですか？ 46
何を期待します？ 47

他に何をすべきか？ 47
値は100%を超えることができますか？ 48
「No Data」とはどういう意味ですか？ 48
Network I/O 48
それは何ですか？ 48
何を期待します？ 49

Overview

Overviewでは、Sysdigの統合されたKubernetesデータプラットフォームを利用して、Kubernetesクラスターとワークロードを監視、保護、トラブルシューティングに活用します。
Kubernetesインフラストラクチャの健全性、リスク、キャパシティの統一されたビューを提供します。これは、マルチおよびハイブリッドクラウド環境全体のKubernetesクラスター、ノード、ネームスペース、およびワークロードの単一のペインです。これらのエンティティのいずれかで簡単にフィルタリングし、関連するイベントとヘルスデータを表示できます。
Overviewには、イベント数と重大度によって優先順位が付けられたメトリクスが表示されるため、問題の根本原因にすばやく到達できます。 Sysdigモニターは、インフラストラクチャデータを10分ごとにポーリングし、システムの健全性が把握できるOverviewページのメトリクスとイベントを更新します。

ヒント

- Overviewデータの詳細については、以下を参照してください。
  - クラスターデータの解釈
  - ノードデータの解釈
  - ネームスペースデータを解釈する
  - ワークロードデータの解釈
- On-Premデプロイメントでは、Overviewはデフォルトでは利用できません。 この機能を有効にするには、オンプレミスデプロイメントにおけるOverviewの有効化を参照してください。

主な利点

Overviewには次の利点があります。

- 大規模なインフラストラクチャ環境の健全性、リスク、リソースの使用、およびキャパシティの統一されたビューを表示します
  - メトリクス、セキュリティイベント、コンプライアンスCISベンチマーク結果、およびコンテキストイベントを単一の場所でレンダリングする
  - スタンドアロンのセキュリティ、監視、フォレンジックツールの必要性を排除
  - ワークロードまたはインフラストラクチャごとにオンザフライでデータを表示
- アラート、Kubernetes、コンテナ、ポリシー、イメージスキャン結果からのコンテキストライブイベントストリームを表示します
- イベント数と重大度に基づいてエンティティをインテリジェントに表面化
- クラスタからノードとネームスペースにドリルダウン

- マルチおよびハイブリッドクラウド環境のインフラストラクチャ監視をサポート
- コア運用ユーザーに基づいて関連情報を公開します。
    - DevOps / Platform Ops
    - セキュリティアナリスト
    - サービスオーナー

Overviewユーザーインターフェイスへのアクセス

Sysdig モニター UIまたはURLを使用して、Overviewのスコープにアクセスして設定できます。

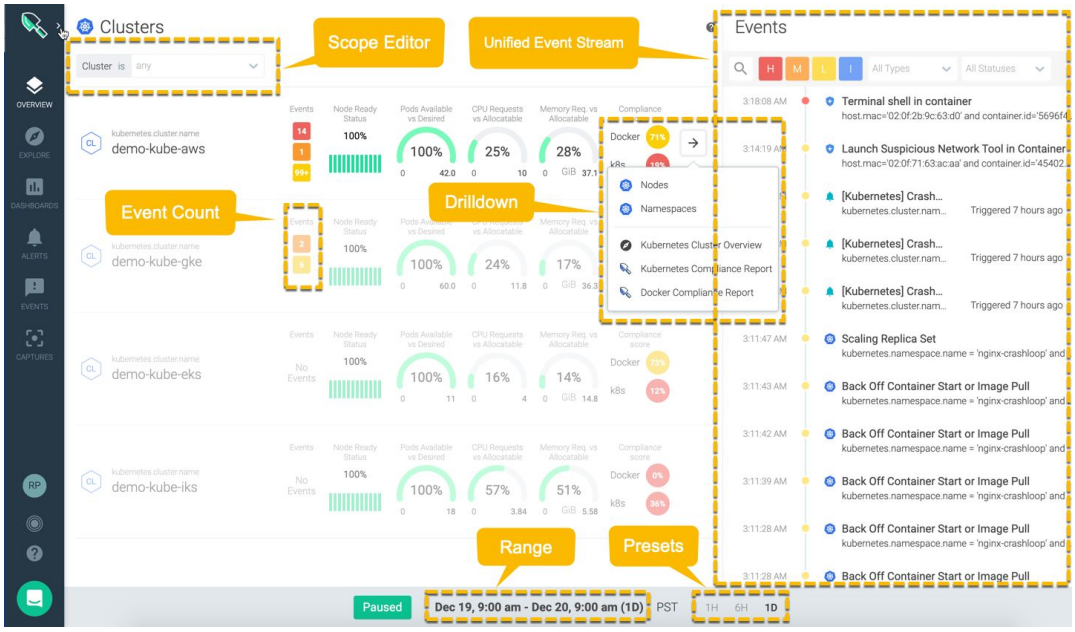
- オンプレミス：https://[Sysdig URL]/#/overview
- SAAS：https://app.sysdigcloud.com/#/overview

左側のナビゲーションで[Overview]をクリックし、Kubernetesエンティティの1つを選択します。

- クラスター
- ノード
- ネームスペース
- ワークロード

Overviewユーザーインターフェイスについて

Overviewインターフェイスが開き、クラスターのOverviewページが表示されます。このセクションでは、インターフェースの主要コンポーネントとナビゲーションオプションについて説明します。



Overview 行

各行はKubernetesエンティティを表します：クラスター、ノード、ネームスペース、またはワークロード。上のスクリーンショットでは、各行がKubernetesクラスターを示しています。

- 行の移動は簡単です

  左側のナビゲーションの[Overview]アイコンをクリックして[Overview]ページを選択するか、次の[Overview]ページにドリルダウンして、次の詳細なデータレベルを確認します。各Overviewページには、デフォルトで10行、最大100行が表示されます。ページごとに10行を超える場合は、[`Load More`]をクリックして追加の行を表示します。

- Overviewページで特定の行を選択する機能

  各行には、データを表示している関連エンティティのスコープが含まれています。特定の行をクリックすると、残りの行の選択が解除され（たとえば、ステージングを選択すると、上記のスクリーンショットの他のすべての行の選択が解除されます）、その行でスコープされるイベントを含む、選択したエンティティのスコープに焦点が当てられます。さらに、ライブバッジは一時停止に変わり、新しいデータが入力されても行が更新されないことを意味します。単一の行に集中するために一時停止すると、エンティティが監視下にある瞬間まで何が起こっているかのスナップショットが提供されます。

- エンティティは、検出されたイベントの重大度と数に従って表示されます。イベントの新しさではありません。

  行は、エンティティに関連付けられているイベントの数と重大度レベルで並べ替えられ、降順で表示されます。重大度の高いイベントの数が最も多いアイテムが最初に表示され、次に中、低、および情報が表示されます。この組織は、数千のエンティティが含まれる可能性のある環境で、即時の注意を必要とするイベントを強調し、トラブルシューティングの作業を合理化するのに役立ちます。

スコープエディター

スコープエディターを使用すると、何千ものエンティティが含まれる可能性のある環境から、特定のワークロードや名ネームスペースなどの特定のエンティティにターゲットを絞ることができます。Kubernetes階層によって決定されるスコープのレベルは、ワークロードからクラスターに進み、クラ

スターは最上位レベルにあります。小規模な環境では、スコープエディターを使用することは、スコープが適用されていない[Overview]ページで1行をクリックすることと同じです。

クラスター：階層の最高レベル。ページに適用される唯一のスコープはクラスターです。使用可能なクラスターのリストから特定のクラスターを選択できます。

ノード：階層の2番目のレベル。スコープはクラスターとノードによって決定されます。選択は、選択したクラスター内の特定のノードに絞り込まれます。

ネームスペース：階層の3番目のレベル。スコープは、クラスターとネームスペースによって決定されます。選択は、選択したクラスター内の特定のネームスペースに絞り込まれます。

ワークロード：階層の最後のエンティティ。スコープは最初にクラスターとネームスペースによって決定され、その後、選択は特定のデプロイメント、サービス、またはステートフルセットに絞り込まれます。 3つのオプションすべてを選択することはできません。

タイムナビゲーション

Overview機能は時間に基づいています。Sysdigモニターは、インフラストラクチャデータを10分ごとにポーリングし、健全性を把握するOverviewページのメトリクスとイベントを更新します。プリセット間隔と時間範囲を選択して、この収集されたデータを表示する方法を選択します。

プリセット

プリセットは、Sysdigモニターが10分ごとに収集するデータを視覚化する方法です。表示するデータサンプルを決定するプリセットを選択します。Overviewでは、次のプリセットがサポートされています。

1時間：過去1時間にポーリングされたデータ。これがデフォルト値です。

6時間：過去6時間にポーリングされたデータ。

1日：最終日にポーリングされたデータ。

プリセットは、範囲選択と連動して機能します。特定のプリセットインターバルを選択すると、範囲の選択が更新され、Overviewの行とイベントが後でリロードされます。例えば：

10分：範囲を7月9日2.20 pm～7月9日2.30 pmにリセットします。

6時間：範囲を7月9日8.30 am～7月9日2.30 pmにリセットします。

1日：範囲を7月8日午後2時30分から7月9日午後2時30分にリセットします。

メトリクスとイベントは[Overview]ページで10分ごとに更新されるため、[Overview]ページで10分以上滞在すると、データが更新され、新しく計算された値が表示されます。

注意

プリセットは、Sysdig モニターインターフェース全体でグローバルです。 たとえば、エクスプローラビューで10分を選択した場合、Overviewプリセットも10分になり、その逆も同様です。エクスプローラーでサポートされていないプリセットを選択すると、Overviewで1日後に戻ります。

範囲

範囲には、日付と時間の間隔、および選択したプリセットが括弧内に表示されます。 UIに表示される範囲は、プリセットによって決まります。指定された時間は最も近い時間間隔であり、デフォルトでは、1時間で事前設定された現在の日付と時刻です。 範囲がプリセットとどのように機能するかを理解するには、プリセットをご覧ください。

時間形式

Overviewでは、UTCおよびPDT時間形式がサポートされています。 [範囲]の横にあるトグルボタンを使用して、[範囲]に表示されるスロットの時間形式を変更します。デフォルトはPDTです。

ライブ

ライブバッジは、フィード（データのある行のOverview）がライブか一時停止かを示します。

- ライブ：データは、Sysdigバックエンドの10分のポーリングに基づいて継続的に更新されます。Overviewフィードは通常、常にライブです。
- 一時停止：特定の行が選択されると、データの更新が一時停止し、行は新しいデータで更新されません。

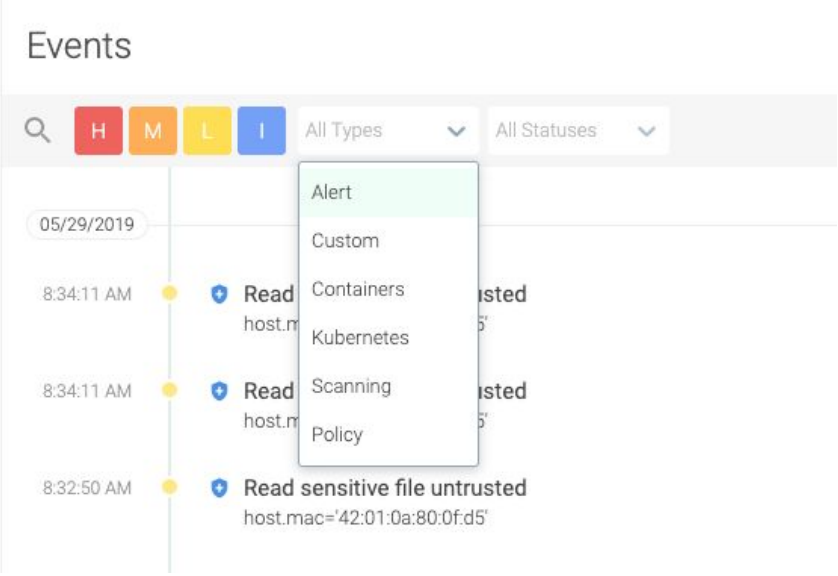
イベントストリームの統合

Overviewの右側のパネルには、状況依存のイベントフィードが表示されます。
Overview行をクリックすると、関連するイベントが右側に表示されます。各イベントには、エンドツーエンドのメタデータがインテリジェントに入力され、コンテキストを提供してトラブルシューティングを可能にします。

イベントの種類

Overviewでは、次のイベントタイプがレンダリングされます。

- Alert：アラートを参照してください。
- Custom：このタイプのイベントを表示するには、カスタムラベルが有効になっていることを確認してください。
- Containers：コンテナに関連付けられたイベント。
- Kubernetes：Kubernetesインフラストラクチャーに関連するイベント。
- Scanning：イメージスキャンを参照してください。
- Policy：ポリシーを参照してください。



イベントのステータス

Overviewでは、次のアラート生成イベントステータスがレンダリングされます。

- **Triggered**：アラート条件が満たされ、まだ持続しています。
- **Resolved**：以前に存在していたアラート条件が解消されました。

- **Acknowledged**：予定の受信者によってイベントが確認されました。
- **Un-acknowledged**：予定の受信者がイベントを確認していません。すべてのイベントはデフォルトで未確認としてマークされます。

一般的なガイドライン

初めての使用

- 環境が初めて作成された場合、Sysdigモニターはデータをフェッチし、関連するページを生成します。Overview機能はすぐに有効になります。ただし、必要なデータを含むOverviewページが表示されるまで、最大で1時間待ちます。
- Overviewでは、1H、6H、および1Dのセグメントでタイムウィンドウを使用するため、それぞれ1H、6H、および1Dが[Overview]ページでデータを表示できるようになるまで待機します。
- 最初の1時間に十分なデータが利用できない場合、最初の1時間が経過するまで[No Data Available]ページが表示されます。

チューニングOverviewデータ

Sysdig モニターは、キャッシュメカニズムを利用して、Overview画面の事前計算済みデータをフェッチします。
事前計算されたデータが利用できない場合、フェッチされたデータは非計算データとなり、表示する前に計算する必要があります。この追加の計算時間は遅延を追加します。Overviewではキャッシュが有効になっていますが、最適なパフォーマンスを得るには、Overviewを初めて使用するときに1H、6H、および1Dウィンドウを待つ必要があります。指定した時間が経過すると、データは1分ごとに自動的にキャッシュされます。

オンプレミスデプロイメントにおけるOverviewの有効化

On-Premデプロイメントでは、Overview機能はデフォルトでは使用できません。次のAPIを使用して有効にします。

1. 次のようにベータ設定を取得します。

```
curl -X GET 'https://<Sysdig URL>/api/on-prem/settings/overviews' \
-H 'Authorization: Bearer <GLOBAL_SUPER_ADMIN_SDC_TOKEN>' \
-H 'X-Sysdig-Product: SDC' -k
```

<Sysdig URL>をデプロイメントに関連付けられたSysdig URLに置き換え、<GLOBAL_SUPER_ADMIN_SDC_TOKEN>をデプロイメントに関連付けられたSDCトークンに置き換えます。

2. ペイロードをコピーし、設定で目的の値を変更します。
3. 次のように設定を更新します。

```
curl X PUT 'https://<Sysdig URL>/api/on-prem/settings/overview' \
-H 'Authorization: Bearer <GLOBAL_SUPER_ADMIN_SDC_TOKEN>' \
-H 'X-Sysdig-Product: SDC' \
-d '{  "overviews": true,  "eventScopeExpansion": true}'
```

機能フラグ

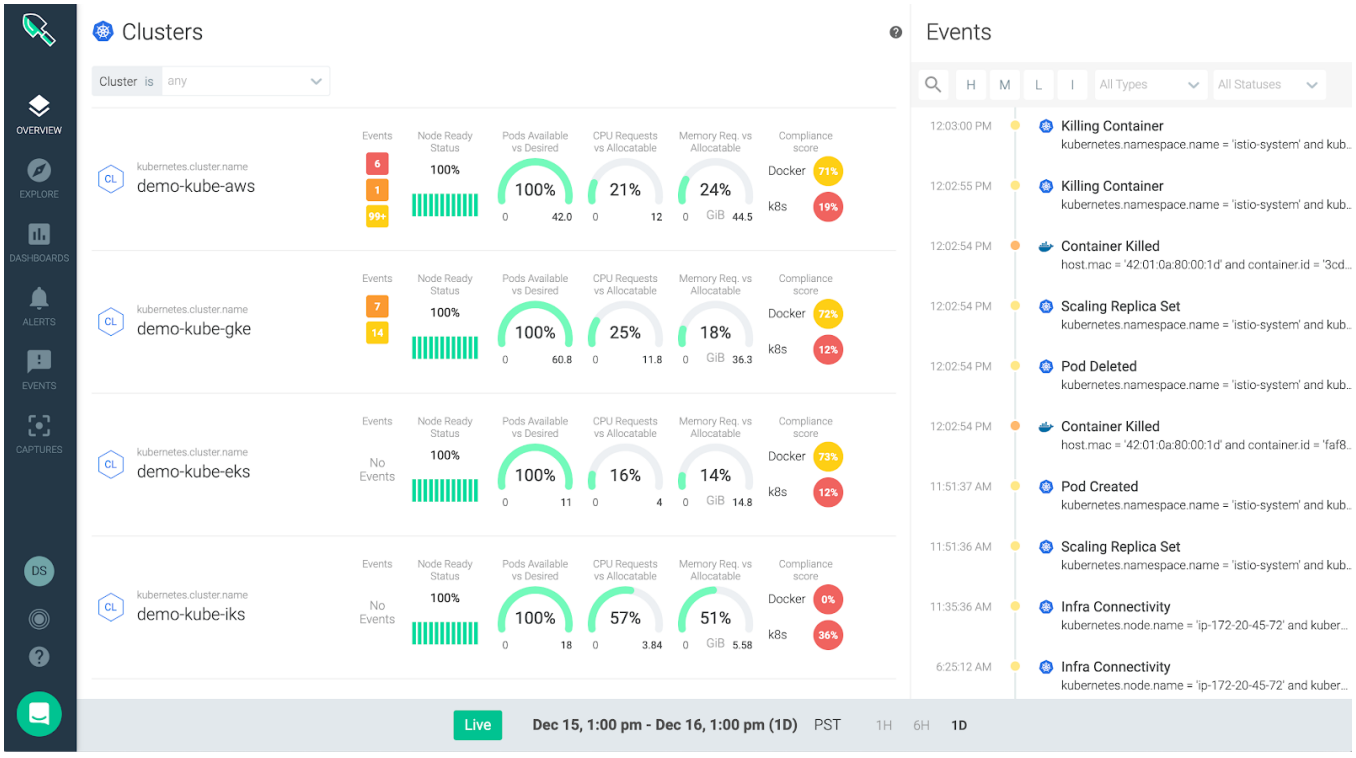
- overviews：overviewsをtrueに設定して、バックエンドコンポーネントとUIを有効にします。
- eventScopeExpansion：すべてのイベントタイプのスコープ拡張を有効にするには、eventScopeExpansionをtrueに設定します。

クラスターデータ

このトピックでは、[クラスターのOverview]ページについて説明し、そのゲージチャートとそこに表示されるデータを理解するのに役立ちます。

クラスターのOverviewについて

Kubernetesでは、ノードのプールがリソースを結合して、より強力なマシン、つまりクラスターを形成します。[クラスターのOverview]ページには、各クラスターの健全性、リスク、キャパシティ、およびコンプライアンスを示す主要なメトリクスが表示されます。 クラスターは、任意のクラウドまたはマルチクラウド環境に配置できます。



「クラスター」ページの各行はクラスターを表します。注意が必要な領域を強調するために、クラスターは対応するイベントの重大度でソートされます。たとえば、重大度の高いイベントのあるクラスターは、ページの上部にバブリングされ、問題が強調表示されます。さらにノードまたはネームスペースのOverviewページにドリルダウンして、各レベルで調査することができます。

クラスターデータの解釈

このトピックでは、[クラスターのOverview]画面に表示されるメトリクスについて説明します。

Node Ready Status

グラフには、`avg(min(kubernetes.node.ready))`によって返された最新の値が表示されます。

それは何ですか？

この数は、ノードがクラスター全体でポッドを受け入れる準備ができていることを示しています。数値の可用性は、ノードがKubernetesによって準備ができていると報告されている時間の割合を示します。例えば：

- 100%は、10個のノードのうち10個が、たとえば過去1時間の時間枠全体に対して準備ができている場合に表示されます。
- 10個のノードのうち9個が全時間枠に対して準備ができており、1つのノードが50%の時間だけ準備ができている場合、95%が表示されます。

棒グラフは、選択した時間ウィンドウ全体の傾向を表示し、各棒はタイムスライスを表します。たとえば、最後の1時間のウィンドウを選択すると6つのバーが表示され、それぞれが10分のタイムスライスを示します。各バーは、タイムスライス全体の可用性（緑）または非可用性（赤）を表します。
たとえば、次の画像は、過去1時間の平均可用性が80%であることを示しています。各10分のタイムスライスは、同じ時間枠の一定の可用性を示しています。

何を期待します？

常に100%であると期待してください。

他に何をすべきか？
値が100%未満の場合は、ノードがまったく使用できないか、または1つ以上のノードが部分的に使用可能かを判断します。

- Overviewの[**Nodes**]画面またはエクスプローラの[**Kubernetes Cluster Overview**]にドリルダウンして、ノードのリストとその可用性を確認します。
- エクスプローラーの**Kubernetes Node Overview**ダッシュボードを確認して、Kubernetesが報告する問題を特定します。

Pods Available vs Desired

グラフには、The chart shows the latest value returned by `sum(avg(kubernetes.namespace.pod.available.count)) / sum(avg(kubernetes.namespace.pod.desired.count))`によって返された最新の値が表示されます。

それは何ですか？
グラフには、特定のクラスター内のすべてのポッドについて、選択した時間ウィンドウ全体で平均化された、利用可能なポッドと望ましいポッドの比率が表示されます。 上限は、クラスター内の必要なポッドの数を示します。
たとえば、次の画像は、42個の必要なポッドが使用できることを示しています。

何を期待します？
通常は100%を期待する必要があります。

特定のポッドが使用可能になるまでに時間がかかる場合、一時的に100%未満の値が表示されることがあります。イメージのプル、ポッドの初期化、準備プローブなどにより、このような遅延が発生します。

他に何をすべきか？

可用性が低い1つ以上のネームスペースを特定します。これを行うには、[Namespaces]画面にドリルダウンしてから、[Workloads]画面にドリルダウンして、使用できないポッドを特定します。

使用できないポッドの数がかなり多い（比率が大幅に低い）場合は、ノードのステータスを確認します。ノードに障害が発生すると、ほとんどのネームスペースでいくつかのポッドが使用できなくなります。

いくつかの要因により、ポッドがPending状態でスタックする可能性があります。

- ポッドは、ノード全体で利用可能なリソース（残りの割り当て可能なポッド）を超えるリソースをリクエストします。
- ポッドは、すべての単一ノードの可用性よりも高いリクエストを作成します。たとえば、8コアのノードがあり、16コアのリクエストでポッドを作成します。これらのポッドでは、ノードのアフィニティと非アフィニティの制約に関連する再構成と特定のセットアップが必要になる場合があります。
- 大量のリソースをリクエストする前に、のネームスペースのクウォータに達しました。
  ネームスペースレベルでクォータが適用されている場合、ノード全体のリソースの可用性に関係なく、リミットに達する可能性があります。

CPU Requests vs Allocatable

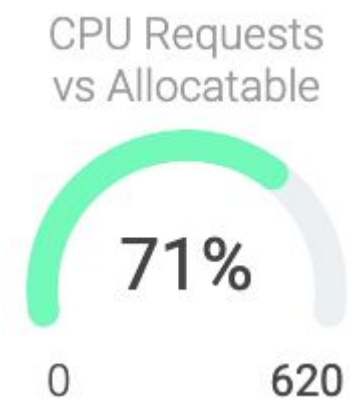
グラフには、`sum(avg(kubernetes.pod.resourceRequests.cpuCores)) / sum(avg(kubernetes.node.allocatable.cpuCores))`によって返される最新の値が表示されます。

それは何ですか？

グラフには、選択したクラスター内のすべてのポッドに構成されたCPUリクエストと、すべてのノードに割り当て可能なCPUの比率が表示されます。

上限は、クラスター内のすべてのノードにわたる割り当て可能なCPUコアの数を示します。

たとえば、次の画像は、すべてのノードで620の使用可能なCPUコア（割り当て可能なCPU）のうち、71%がポッドによって要求されていることを示しています。



何を期待します？

リソース使用率戦略によって、期待できる比率が決まります。健康的な比率は50%から80%の間です。

すべてのノードに同じ量の割り当て可能なリソースがあると仮定すると、妥当な上限は`(node_count - 1) / node_count x 100`の値です。たとえば、9つのノードがある場合、比率は90%になります。このパーセンテージがあると、ノードが使用できなくなるのを防ぎます。

他に何をすべきか？

比率が低いほど、インフラストラクチャの使用率が低いリソース（および対応するコスト）を示します。より高い比率は、不十分なリソースを示します。結果として

- アプリケーションの実行をスケジュールすることはできません。
- ポッドが起動せず、ペンディング/スケジュールされていない状態のままになることがあります。

トリアージするには、次の手順を実行します。

- Nodes画面にドリルダウンして、すべてのノード間でリソースがどのように利用されているかを分析します。
- Namespaces画面にドリルダウンして、ネームスペース全体でリソースがどのように要求されるかを理解します。
- エクスプローラにドリルダウンして、次のダッシュボードを参照します。
    - **Kubernetes CPU Allocation Optimization**：インフラストラクチャーでかなりの量のリソースが十分に活用されていないかどうかを評価します。
    - **Kubernetes Workloads CPU Usage and Allocation**：ポッドが適切に構成され、期待どおりにリソースを使用しているかどうかを確認します。

値は100%を超えることができますか？

現在、比率はスケジュールされたポッドのみを考慮していますが、ペンディングのポッドは計算から除外されています。つまり、ポッドは割り当て可能なポッドのノードで実行するようにスケジュールされています。したがって、比率は100%を超えることはできません。

オーバーコミットメント（利用可能なリソースよりも多くのリソースを要求するポッド）の場合、リクエストと割り当て可能な比率が高くなり、使用可能なポッドと必要なポッドが低くなることが予想されます。これは、利用可能なリソースのほとんどが使用されていることを示しており、残りは追加のポッドをスケジュールするのに十分ではありません。したがって、ポッドの利用可能と希望の比率は減少します。

注意

頻繁に更新されるポッド、または頻繁に削除および作成されるポッドが環境にある場合（クラスターのテストなど）、リクエストの総数は常にそれよりも高く表示されることがあります。 その結果、選択した時間枠全体で比率が高くなり、100%を超える値が表示される場合があります。 このエラーは、データエンジンが集計比率を計算する方法が原因で表示されます。

Kubernetes Cluster Overviewにドリルダウンして、**CPU Cores Usage vs Requests vs Allocatable** 時系列を確認し、リクエストコミットメントの傾向を正しく評価します。

以下に、ポッドがPending状態でスタックする原因となる可能性があるいくつかの要因を示します。

- ポッドは、ノード全体で利用可能なもの（残りの割り当て可能なポッド）を超えるリクエストを行います。Requests vs Allocatable比率は、この問題の指標です。

- ポッドは、すべての単一ノードの可用性よりも高いリクエストを行います。たとえば、8コアのノードがあり、16コアのリクエストでポッドを作成します。これらのポッドでは、ノードのアフィニティと非アフィニティの制約に関連する再構成と特定のセットアップが必要になる場合があります。

- ネームスペースレベルで設定された割り当て量に達してから、リクエストが構成されます。Requests vs Allocatable比率は問題を示唆していない可能性がありますが、特に特定のネームスペースでは、Pods Available vs Desired比率が低下します。OverviewのNamespace画面を参照してください。

Memory Requests vs Allocatable

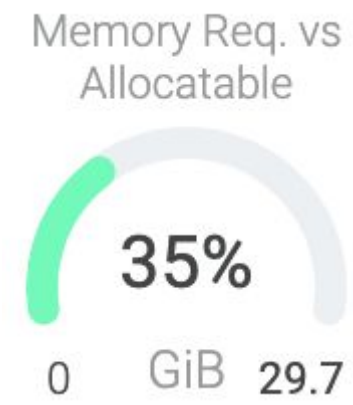
グラフには、`sum(avg(kubernetes.pod.resourceRequests.memBytes)) / sum(avg(kubernetes.node.allocatable.memBytes))`によって返された最新の値が表示されます。

それは何ですか？

グラフには、クラスター内のすべてのポッドに構成されたメモリリクエストと、すべてのノードで利用可能な割り当て可能なメモリの比率が表示されます。

上限は、すべてのノードで使用可能な割り当て可能なメモリを示します。値はバイトで表され、指定された単位で表示されます。

たとえば、次の画像は、すべてのノード（割り当て可能なメモリ）で利用可能な29.7 GiBのうち、35%がポッドによって要求されていることを示しています。



何を期待します？

リソース使用率戦略によって、期待できる比率が決まります。健康的な比率は50%から80%の間です。

すべてのノードに同じ量の割り当て可能なリソースがあると仮定すると、妥当な上限は`(node_count - 1) / node_count x 100`の値です。たとえば、9つのノードがある場合は90%です。この比率は、ノードが使用できなくなることからシステムを保護します。

そうでなければ何をすべきか

比率が低いほど、インフラストラクチャの使用率が低いリソース（および対応するコスト）を示します。より高い比率は、不十分なリソースを示します。結果として

- アプリケーションの実行をスケジュールすることはできません。
- ポッドが起動せず、ペンディング/スケジュールされていない状態のままになることがあります。

トラブルシューティングを行うには、次のようにします。

- Nodes画面にドリルダウンして、すべてのノードでリソースがどのように利用されているかについての洞察を得ます。
- Namespacesにドリルダウンして、ネームスペース全体でリソースがどのように要求されるかを理解します。
- エクスプローラにドリルダウンして、次のダッシュボードを参照します。
    - **Kubernetes Memory Allocation Optimization**：インフラストラクチャで大量のリソースが十分に活用されていないかどうかを評価します。
    - **Kubernetes Workloads Memory Usage and Allocation**：ポッドが適切に構成され、期待どおりにリソースを使用しているかどうかを確認します。

値は100%より高くできますか？

現在、比率はスケジュールされたポッドのみを考慮していますが、ペンディング中のポッドは計算から除外されています。これが意味することは、ポッドは、利用可能な割り当て可能なリソースのうち、ノードで実行するようにスケジュールされているということです。したがって、比率は100%を超えることはできません。

オーバーコミットメント（利用可能なリソースよりも多くのリソースを要求するポッド）の場合は、リクエストと割り当て可能率の比率が高く、ポッドの利用可能率と必要な比率の比率が低いと予想されます。これは、利用可能なリソースのほとんどが使用されており、残りのポッドで追加のポッドをスケジュールするには不十分であることを示しています。したがって、使用可能なポッドと必要なポッドの比率が減少します。

注意

頻繁に更新されるポッド、または頻繁に削除および作成されるポッドが環境にある場合（クラスターのテストなど）、リクエストの総数は常にそれよりも高く表示されることがあります。 その結果、選択した時間枠全体で比率が高くなり、100%を超える値が表示される場合があります。 このエラーは、データエンジンが集計比率を計算する方法が原因で表示されます。

Kubernetes Cluster Overviewにドリルダウンして、**Memory Usage vs Requests vs Allocatable**時系列を確認し、リクエストコミットメントの傾向を正しく評価します。

リストされているのは、ポッドがPending状態でスタックする原因となる可能性があるいくつかの要因です。

- ポッドは、ノード全体で利用可能なもの（残りの割り当て可能なポッド）を超えるリクエストを行います。 **Requests vs Allocatable**比率は、この問題の指標です。

- ポッドは、すべての単一ノードの可用性よりも高いリクエストを行います。 たとえば、8コアのノードがあり、16コアのリクエストでポッドを作成します。 これらのポッドでは、ノードのアフィニティおよび非アフィニティファクターに関連する構成変更と特定のセットアップが必要になる場合があります。
- ネームスペースレベルで設定されたクォータに達した後、高い要求が構成されます。 **Requests vs Allocatable**比率は問題を示唆していない可能性がありますが、特に特定のNamespaseでは、**Pods Available vs Desired**比率が低下します。 OverviewのNamespase画面を参照してください。

コンプライアンススコア

Docker：`avg(avg(compliance.k8s-bench.pass_pct))`によって返される最新の値。

Kubernetes：`avg(avg(compliance.docker-bench.pass_pct))`によって返される最新の値。

それは何ですか？

数値は、DockerエンティティとKubernetesエンティティのそれぞれについて、選択した時間枠で成功したベンチマークの割合を示しています。

何を期待しますか？

Sysdig Secureを有効にしていない場合、またはベンチマークをスケジュールしていない場合は、使用可能なデータがないと予想されます。

それ以外の場合、スコアが高いほど、インフラストラクチャーのコンプライアンスが高くなります。

他に何をすべきか？

スコアが予想よりも低い場合は、 **Docker Compliance Report**または**Kubernetes Compliance Report**にドリルダウンして、ベンチマークチェックとその結果の詳細を確認します。

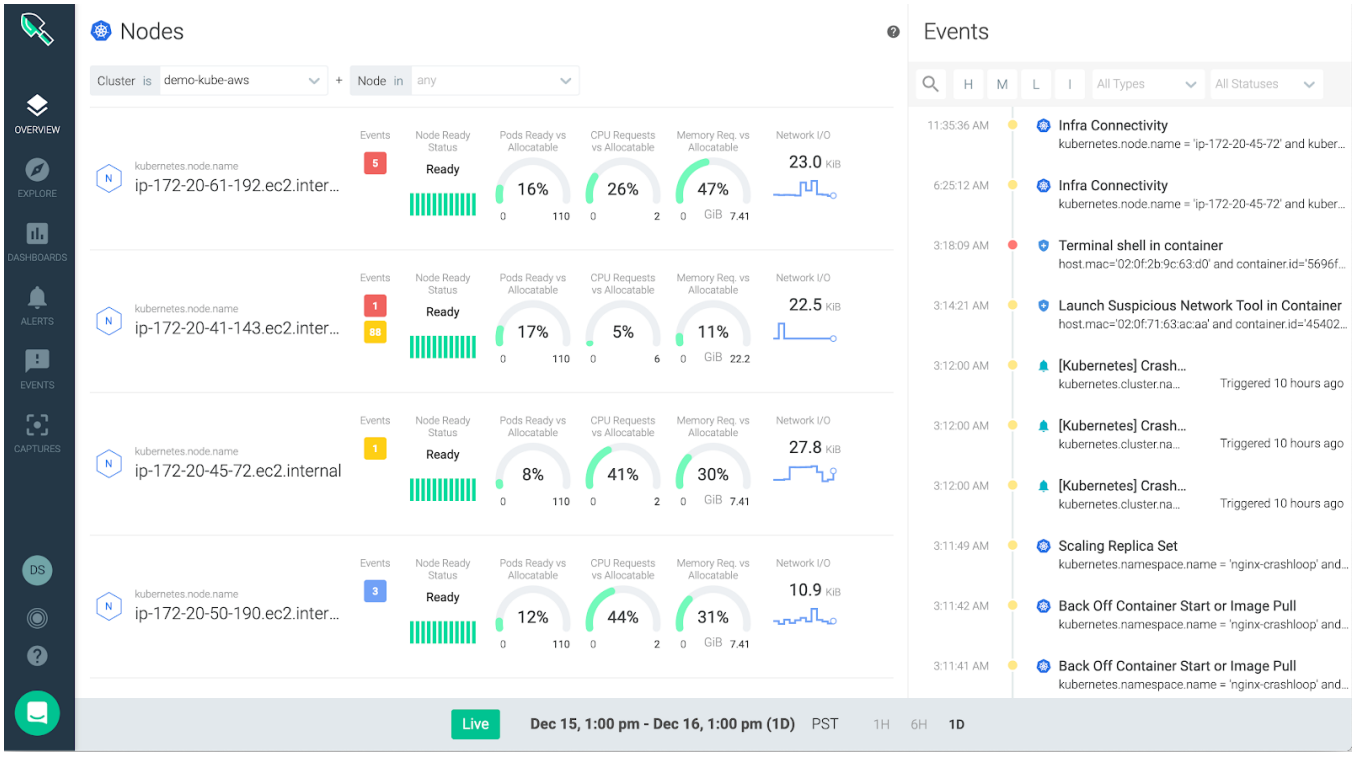
また、チェックの履歴を確認するには、Sysdig Secureの[**Benchmarks / Results**]ページを使用することもできます。

ノードデータ

このトピックでは、[ノードのOverview]ページについて説明し、ゲージチャートとそこに表示されるデータを理解するのに役立ちます。

ノードのOverviewについて

ノードは、Kubernetesのワーカーマシンを指します。 物理マシンまたはVMはノードを表すことができます。 [ノードのOverview]ページには、クラスター内の各ノードの健全性、キャパシティ、コンプライアンスを示す主要なメトリクスが表示されます。



ノードデータの解釈

このトピックでは、[ノードのOverview]ページに表示されるメトリクスについて説明します。

ノード準備ステータス

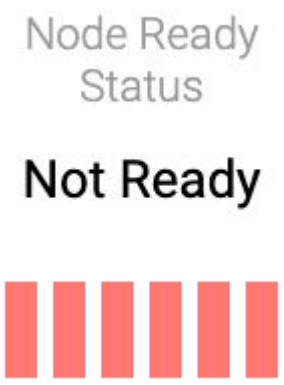
グラフには、`avg(min(kubernetes.node.ready))`によって返された最新の値が表示されます。

それは何ですか？

この数は、クラスター全体でポッドを受け入れるノードの準備状態を表します。 数値の可用性は、ノードがKubernetesによって準備ができていると報告されている時間の割合を示します。 例えば：

- 100%は、ノードが時間枠全体、たとえば過去1時間の準備ができると表示されます。
- ノードが95%の時間枠の準備ができているときの95%、たとえば60分のうち57分。

棒グラフは、選択した時間ウィンドウ全体の傾向を表示し、各棒はタイムスライスを表します。 たとえば、「過去1時間」を選択すると6つのバーが表示され、それぞれが10分のタイムスライスを示します。 各バーは、タイムスライス全体の可用性（緑）と非可用性（赤）を示します。
たとえば、次の画像は、ノードが最後の1時間の時間枠全体に対して準備ができていないことを示しています。

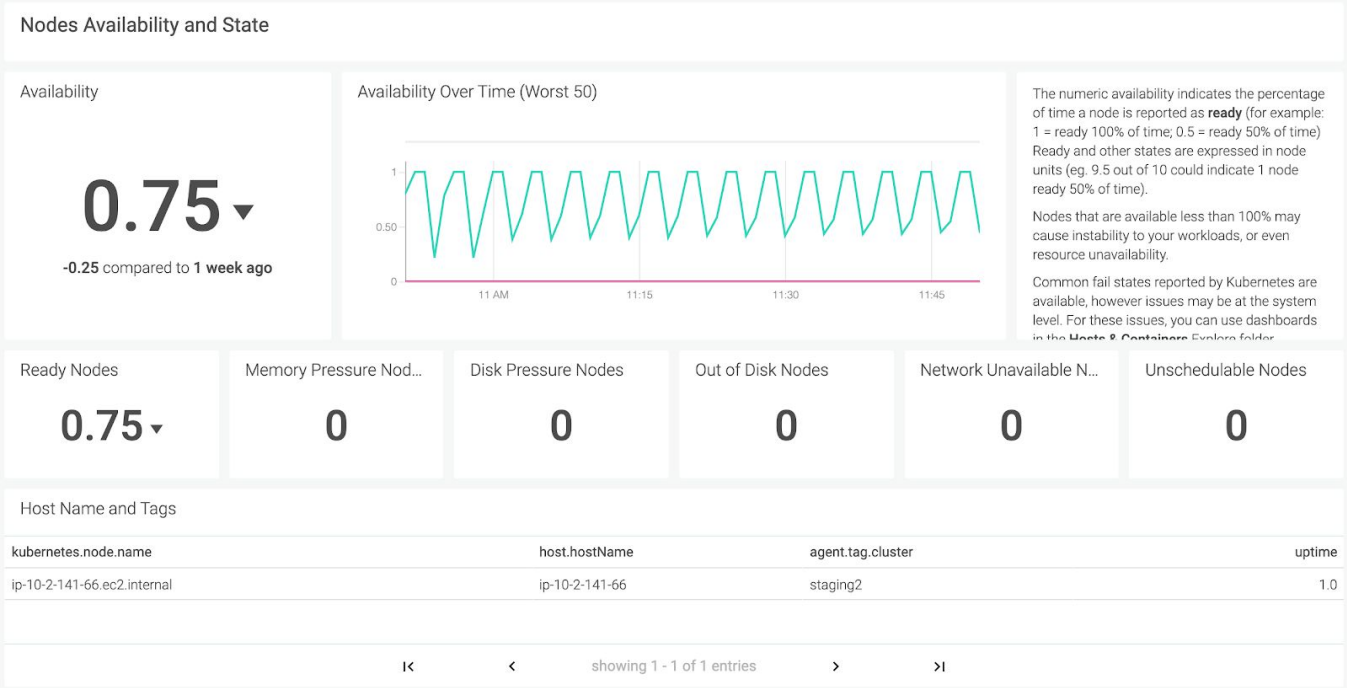


何を期待します？

グラフには常に100%と表示されます。

他に何をすべきか？

数が100%未満の場合は、Kubernetesによって報告されたステータスを確認します。 エクスプローラのKubernetes Node Overviewダッシュボードにドリルダウンして、ノードの準備状況の詳細を確認します。



図に示すように、ノード準備ステータスに交互の動作がある場合、ノードはフラッピングしています。 羽ばたきは、kubeletが正常でないことを示します。 ノードの準備ができていない原因を特定するのに役立つKubernetesによって報告された特定の条件を確認します。このような状態には、ネットワークの問題とメモリの負荷が含まれます。

Pods Ready vs Allocatable

グラフは、`sum(avg(kubernetes.pod.status.ready)) / avg(avg(kubernetes.node.allocatable.pods))`の最新の値を報告します。

それは何ですか？

これは、ノードで構成された使用可能なポッドと割り当て可能なポッドの比率であり、選択された時間ウィンドウ全体で平均化されます。

注意

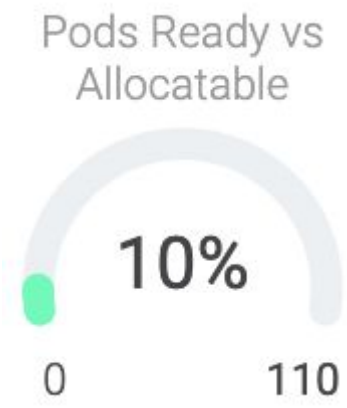
Clustersページには、Pods Available vs Desiredという名前の同様のチャートが含まれています。 ただし、意味は異なります。

- クラスターの利用可能なポッドと必要なポッドのグラフは、予想されるポッドの数と実際に

利用可能なポッドの数を示しています。 詳細な定義については、IsPodAvailableを参照してください。

- ノードのPods Ready vs Allocatableチャートは、各ノードでスケジュールできるポッドの数と実際に準備ができているポッドの数を示します。

上限は、ノードに割り当てることができるポッドの数を示します。 ノード構成を参照してください。たとえば、次の画像は、ノードに110のポッドを割り当てることができることを示していますが（デフォルトの構成）、11のポッドしか準備できていません。



何を期待します？

この比率はリソース使用率とは関係ありませんが、各ノードのポッド密度を測定します。 1つのノードでポッドを増やすと、ポッド、ルーティングメカニズム、およびKubernetes全体を管理するためにkubeletが費やす労力が増えます。
割り当て可能値が適切に設定されている場合、80%未満の値は正常な状態を示します。

他に何をすべきか？

CPUとメモリのリクエストと割り当て可能な比率を参照して、ワークロードのリソース使用率を把握してください。この比率は、クラスターが正常かどうかを示します。値が高い（100%に近い）場合は、次の点を検討してください。

- Kubeletのデフォルトの最大ポッド構成を確認して、特にCPUとメモリの使用率が正常な場合に、より多くのポッドを許可します。
- ノードを追加して、より多くのポッドをスケジュールできるようにします。
- 一般的なkubeletプロセスのパフォーマンスとノードリソースの使用率の確認。比率が高いほど、オペレーティングシステムとKubernetes自体の負荷が高いことを示します。

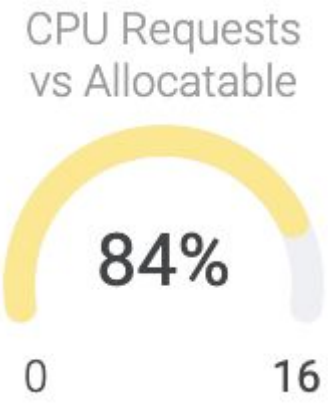
CPU Requests vs Allocatable

グラフには、`sum(avg(kubernetes.pod.resourceRequests.cpuCores)) / sum(avg(kubernetes.node.allocatable.cpuCores))`によって返される最新の値が表示されます。

それは何ですか？

このグラフは、ノードでスケジュールされたポッドによって要求されたCPUコアの数と、ポッドで使用可能なコアの数の比率を示しています。 上限は、ポッドで使用可能なCPUコアを示します。これは、割り当て可能なCPUのユーザー定義構成に対応しています。

たとえば、次の画像は、ノードに使用可能な16個のCPUコアがあり、そのうち84%がノードでスケジュールされたポッドによって要求されていることを示しています。



何を期待します？

最大80%の値を期待してください。

すべてのノードに同じ量の割り当て可能なリソースがあると仮定すると、妥当な上限は`(node_count - 1) / node_count x 100`の値です。たとえば、9つのノードがある場合は90%です。比率が高いと、ノードが使用できなくなることからシステムが保護されます。

他に何をすべきか？

- 比率が低い場合は、ノードが十分に活用されていないことを示しています。[**Clusters**]ページで対応するクラスターにドリルアップして、現在実行中のポッドの数が少ないかどうか、または他の理由でポッドを実行できないかどうかを判断します。
- 比率が高い場合は、ノードに追加のポッドをスケジュールできないリスクがある可能性を示しています。
  KubernetesノードのOverviewダッシュボードにドリルダウンして、実行されているネームスペース、ワークロード、ポッドを評価します。さらに、クラスターページをドリルアップし

て、CPUリソースをオーバーコミットしているかどうかを評価します。リクエストを満たすのに十分なリソースがない可能性があり、その結果、ポッドがノードで実行できない可能性があります。ノードを追加するか、ノードを追加のCPUコアで置き換えることを検討してください。

値は100%を超えることができますか？

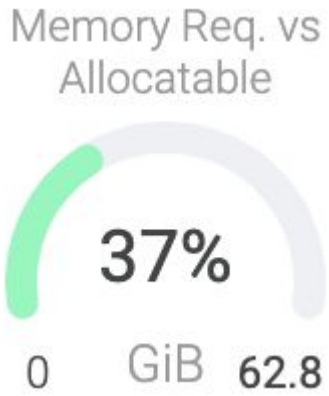
Kubernetesは、ポッドリクエストを満たすために十分な割り当て可能なリソースが利用可能なノードでポッドをスケジュールします。つまり、Kubernetesでは、割り当て可能な合計よりも高い合計リクエストを許可していません。したがって、比率は100%を超えることはできません。
オーバーコミット（容量を超えるリソースを要求するポッド）は、クラスターレベルでの要求と割り当て可能率の比率が高く、ポッドの使用可能と必要な比率の比率が低くなります。これは、使用可能なリソースのほとんどが使用されていることを示しているため、追加のポッドをスケジュールするには、使用可能なリソースでは不十分です。したがって、使用可能なポッドと必要なポッドの比率も減少します。

Memory Requests vs Allocatable

グラフは、`sum(avg(kubernetes.pod.resourceRequests.memBytes)) / sum(avg(kubernetes.node.allocatable.memBytes))`によって返された最新の値を強調表示しています。

それは何ですか？

ノードにスケジュールされたポッドと使用可能なメモリのバイト数によってメモリのバイト数が要求されます。上限は、ポッドで使用可能なメモリを示します。これは、ユーザー定義の割り当て可能なメモリ構成に対応しています。
たとえば、次の画像は、ノードに62.8 GiBのメモリが使用可能であることを示しています。そのうち37%はノードでスケジュールされたポッドによって要求されます。



何を期待します？

健全な比率は80%を下回ります。

すべてのノードに同じ量の割り当て可能なリソースがあると仮定すると、妥当な上限は`(node_count - 1) / node_count x 100`の値です。たとえば、9つのノードがある場合、比率は90%です。比率が高いと、ノードが使用できなくなることからシステムが保護されます。

他に何をすべきか？

- 比率が低い場合は、ノードが十分に活用されていないことを示しています。[**Clusters**]ページで対応するクラスターにドリルアップして、実行中のポッドの数が少ないかどうか、または他の理由でポッドを実行できないかどうかを判断します。
- 比率が高い場合は、ノードで追加のポッドをスケジュールできないリスクがある可能性を示しています。
    - KubernetesノードのOverviewダッシュボードにドリルダウンして、実行されているネームスペース、ワークロード、ポッドを評価します。
    - さらに、クラスターページをドリルアップして、メモリリソースをオーバーコミットしているかどうかを評価します。その結果、リクエストを満たすのに十分なリソースがなく、ポッドが実行できない可能性があります。ノードを追加するか、ノードをより多くのメモリで置き換えることを検討してください。

値は100%より高くできますか？

Kubernetesは、ポッドリクエストを満たすのに十分な割り当て可能なリソースが利用可能なノードでポッドをスケジュールします。つまり、Kubernetesでは、割り当て可能な合計よりも高い合計リクエストを許可していません。したがって、比率は100%を超えることはできません。

オーバーコミット（利用可能なリソースよりも多くのリソースを要求するポッド）を行うと、ノードレベルではリクエストと割り当て可能の比率が高くなり、クラスターレベルではポッドの利用可能と

希望の比率が低くなります。これは、ほとんどのリソースが使用されていることを示しているため、追加のポッドをスケジュールするには十分なリソースがありません。したがって、使用可能なポッドと必要なポッドの比率も減少します。

Network I/O

グラフには、`avg(avg(net.bytes.total))`によって返された最新の値が表示されます。

それは何ですか？

スパークラインは、ノードのネットワークトラフィック（受信および送信）の傾向を示します。 数値は、特定の単位の1秒あたりのバイト数で表される最新の速度を示します。

参考までに、スパークラインは次のステップ数（サンプリング）を示しています。

- 最後の1時間：6ステップ、それぞれ10分のタイムスライス
- 過去6時間：12ステップ、それぞれ20分のタイムスライス
- 最終日：12ステップ、それぞれ2時間のタイムスライス

何を期待します？

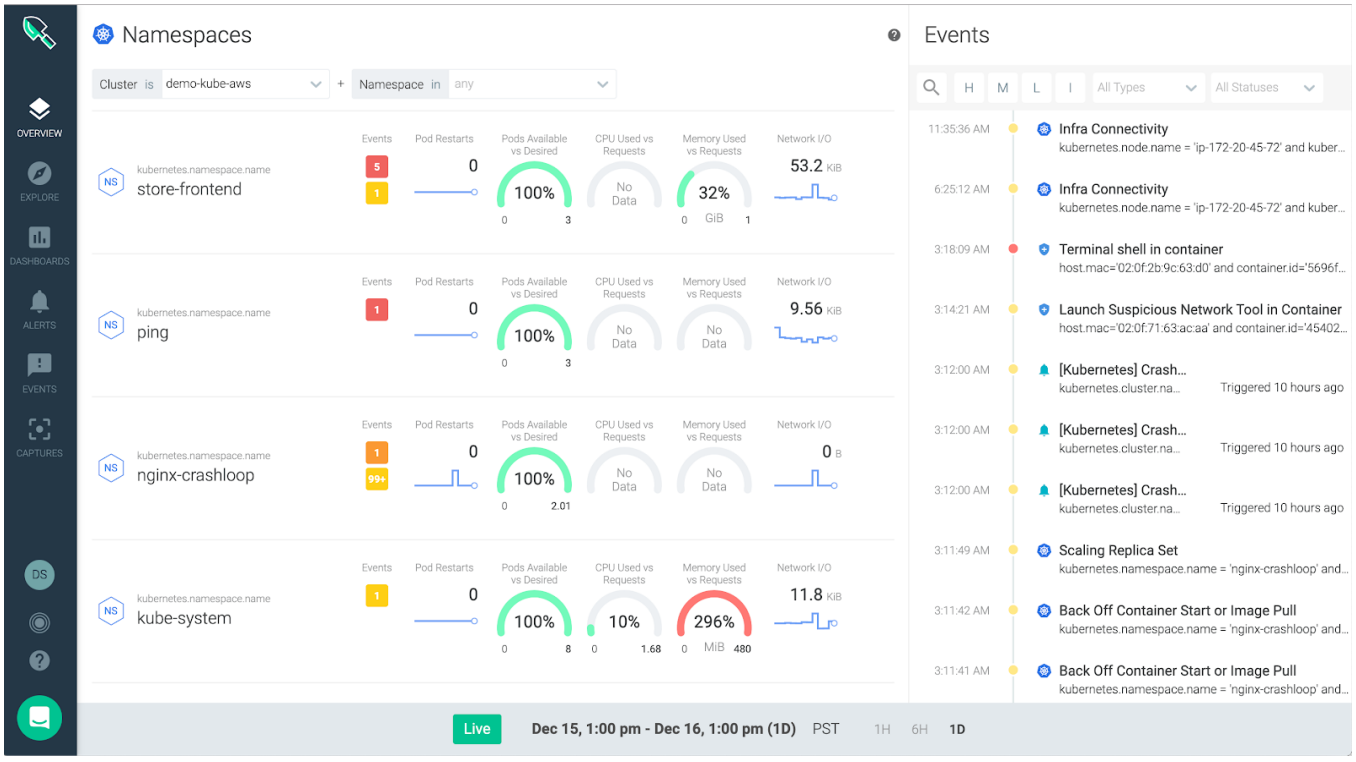
メトリクスは、ノードで実行されるアプリケーションのタイプに大きく依存します。Kubernetes関連のオペレーションには、いくつかのネットワークアクティビティが想定されます。
エクスプローラのKubernetes Node Overviewダッシュボードにドリルダウンすると、ポッド全体のネットワークアクティビティなどの詳細が表示されます。

ネームスペースデータ

このトピックでは、ネームスペースのOverviewページについて説明し、ゲージチャートとそれらに表示されるデータを理解するのに役立ちます。

ネームスペースのOverviewについて

ネームスペースは、物理クラスター上の仮想クラスターです。 チームとその環境を論理的に分離します。 [ネームスペースのOverview]ページには、クラスター内の各ネームスペースの状態、キャパシティ、およびパフォーマンスを示す主要なメトリクスが表示されます。



ネームスペースデータを解釈する

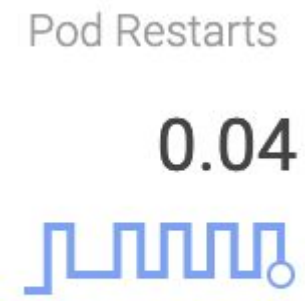
このトピックでは、ネームスペースのOverview画面に表示されるメトリックについて説明します。

ポッドの再起動

グラフは、`avg(timeAvg(kubernetes.pod.restart.rate))`によって返された最新の値を強調表示しています。

それは何ですか？

スパークラインは、選択されたネームスペース内のすべてのポッドにわたるポッド再起動率の傾向を示します。この数値は、1秒あたりの最新の再起動率を示します。



たとえば、選択した時間枠が1日の場合、この画像は過去2時間の毎秒0.04回の再起動率を示しています。この傾向はまた、非フラットパターン（定期的なクラッシュ）を示唆しています。

- 最後の1時間：6ステップ、それぞれ10分のタイムスライス
- 過去6時間：12ステップ、それぞれ20分のタイムスライス
- 最終日：12ステップ、それぞれ2時間のタイムスライス

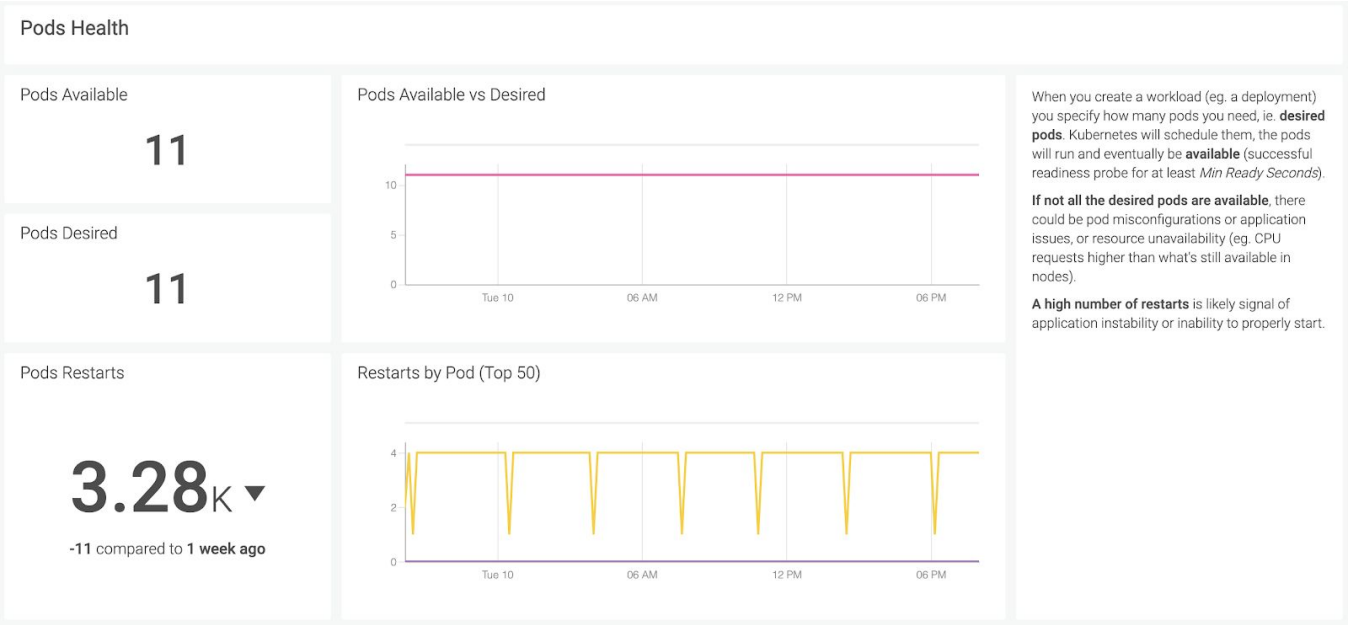
何を期待します？

すべてのポッドで0の再起動が期待されます。

他に何をすべきか？

過去1時間またはそれ以上の時間枠で数回の再起動を行っても、深刻な問題を示しているとは限りません。イベント再起動ループで、次のように根本原因を特定します。

- 「Overview」の「ワークロード」ページにドリルダウンして、再起動ループでスタックしているワークロードを特定します。
- Kubernetes Namespace Overviewにドリルダウンして、ポッドごとの詳細な傾向を確認します。



Pods Available vs Desired

グラフには、`sum(avg(kubernetes.namespace.pod.available.count)) / sum(avg(kubernetes.namespace.pod.desired.count))`によって返された最新の値が表示されます。

それは何ですか？

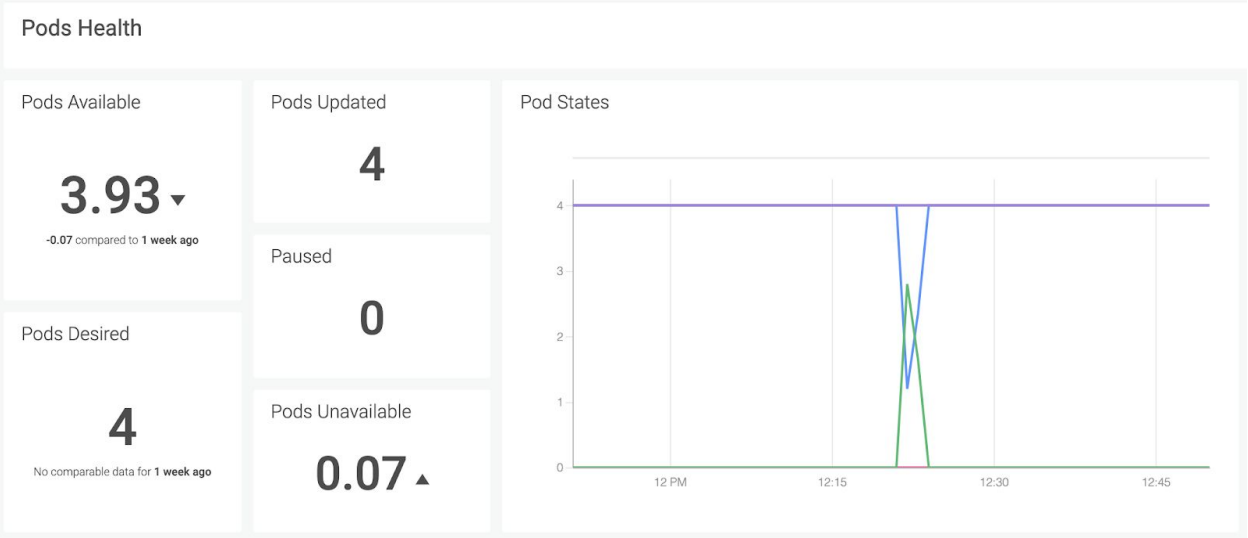
グラフには、特定のネームスペースにおける、選択された時間枠で平均化された、利用可能なポッドと望ましいポッドの比率が表示されます。
上限は、ネームスペース内の必要なポッドの数を示しています。
たとえば、以下の画像は、利用可能な42の必要なポッドを示しています。

何を期待します？

グラフは100%になると考えてください。
遅延（イメージのプル時間、ポッドの初期化、readinessプローブ）のために特定のポッドが使用可能になるまでにかなりの時間がかかる場合、一時的に100%未満の比率が表示されることがあります。

他に何をすべきか？

- [ワークロード]ページにドリルダウンして、可用性の低い1つ以上のワークロードを特定します。
- ワークロードを特定したら、エクスプローラの関連ダッシュボードにドリルダウンします。たとえば、ポッドの傾向と状態を判断するためのKubernetes Deployment Overview。
- たとえば、次の画像では、比率は98%（3.93 / 4 x 100）です。この低下は、ポッドが終了し、その結果新しいバージョンで開始された更新が原因です。



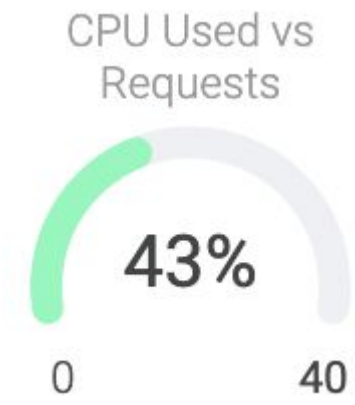
CPU Used vs Requests

グラフには、`sum(avg(cpu.cores.used)) / sum(avg(kubernetes.pod.resourceRequests.cpuCores))`によって返された最新の値が表示されます。

それは何ですか？

このグラフは、ネームスペース内のすべてのポッド全体の合計CPU使用率と、すべてのポッドによって要求された合計CPUの比率を示しています。
上限は、すべてのポッドによって要求された合計CPUを示します。この値は、CPUコアの数として表されます。

たとえば、次の画像は、40 CPUコアに対するネームスペースリクエスト内のポッドを示していますが、そのうち43%しか使用されていません（約17コア）。



何を期待します？

表示される値は、ネームスペースで実行されているワークロードのタイプによって異なります。
通常、80%から120%の間の値は正常と見なされます。 100%を超える値は、短時間は比較的正常と見なされます。
リソース使用率が一定のアプリケーション（バックグラウンドプロセスなど）の場合、比率は100%に近いと予想されます。
APIサーバーなどの「バースト性」アプリケーションの場合、比率は100%未満になると予想されます。この値は選択した時間枠で平均化されるため、使用量の急増はアイドル期間によって補われることに注意してください。

他に何をすべきか？

使用率が低い場合は、アプリケーションが適切に実行されていない（予期された機能が実行されていない）か、ワークロードの構成が正確ではない（ポッドが実際に必要とするものと比較して要求が多すぎる）ことを示します。
使用率が高い場合は、アプリケーションが高負荷で動作しているか、ワークロードの構成が正確ではないことを示します（要求が実際に必要なポッドと比べて低すぎる）。
どちらの場合も、[ワークロード]ページにドリルダウンして、より詳細な分析が必要なワークロードを特定します。

値は100%を超えることができますか？

はい、できます。

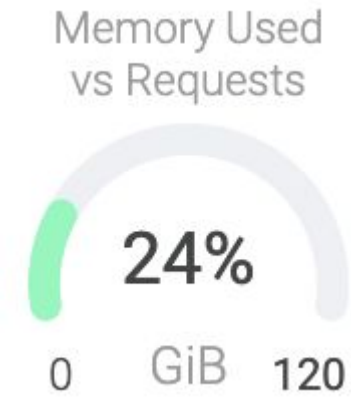
- 制限なしのリクエスト、または制限未満のリクエストを設定できます。どちらの場合でも、通常は一時的な過負荷を処理するために、コンテナが要求よりも多くのリソースを使用できるようにします。
- それぞれが1つのポッドを持つ2つのワークロードを持つネームスペースについて考えます。たとえば、1つのワークロードが1つのCPUコアを要求するように構成され、1つのCPUコアを使用しているとします（使用済みと要求の比率は100%です）。もう1つのワークロードはリクエストなしで構成され、1つのCPUコアを使用します。この例では、1つのCPUコアに対して使用される2つのCPUコアが、ネームスペースレベルで要求された比率は200%です。

Memory Used vs Requests

グラフには、`sum(avg(memory.bytes.used)) / sum(avg(kubernetes.pod.resourceRequests.memBytes))`によって返された最新の値が表示されます。

それは何ですか？

このグラフは、ネームスペースのすべてのポッド全体の合計メモリ使用量と、すべてのポッドによって要求された合計メモリの比率を示しています。
上限は、すべてのポッドによって要求されたメモリの合計を示し、バイトの指定単位で表されます。
たとえば、以下の画像は、ネームスペース内のすべてのポッドが120 GiBを要求し、そのうち24%のみが使用されていることを示しています（約29 GiB）。



何を期待します？

ネームスペースで実行するワークロードのタイプによって異なります。通常、80%から120%の間の値は正常と見なされます。

100%を超える値は、比較的短時間で正常と見なされます。

他に何をすべきか？

使用率が低い場合は、アプリケーションが適切に実行されていない（期待される機能が実行されていない）か、ワークロードの構成が正確でない（ポッドが実際に必要とするものと比較して要求が多い）ことを示します。
使用率が高い場合は、アプリケーションが高負荷で動作しているか、ワークロードの構成が正確でないことを示します（ポッドが実際に必要とするものと比較してリクエストが少ない）。
ワークロードに設定されたリミットとノードのメモリ負荷を考慮して、ワークロードが要求されたものより多くのメモリを使用する場合、それらは追い出されるリスクがあります。詳細については、コンテナのリミットを超えるを参照してください。
どちらの場合も、[ワークロード]ページにドリルダウンして、より詳細な分析が必要なワークロードを特定することができます。

値は100%を超えることができますか？

はい、できます。

- リミットなしのリクエスト、またはリミット未満のリクエストを設定できます。どちらの場合でも、通常は一時的な過負荷を処理するために、コンテナが要求よりも多くのリソースを使用できるようにします。
- それぞれが1つのポッドを持つ2つのワークロードを持つネームスペースについて考えます。たとえば、1つのワークロードが1 GiBのメモリを要求するように構成されており、1 GiBを使用しているとします（使用済みと要求の比率は100%です）。もう1つのワークロードはリクエストなしで構成され、1 GiBを使用します。この例では、ネームスペースレベルでの2 GiBのメモリ使用量と1 GiBの要求比は200%です。

Network I/O

グラフには、`avg(avg(net.bytes.total))`によって返された最新の値が表示されます。

それは何ですか？

スパークラインは、ネームスペース内のすべてのポッドのネットワークトラフィック（受信および送信）の傾向を示します。 数値は、特定の単位の1秒あたりのバイト数で表される最新の速度を示します。

参考までに、スパークラインは次のステップ数（サンプリング）を示しています。

- 最後の1時間：6ステップ、それぞれ10分のタイムスライス
- 過去6時間：12ステップ、それぞれ30分のタイムスライス
- 最終日：12ステップ、それぞれ2時間のタイムスライス

何を期待します？

ネームスペースで実行されるアプリケーションのタイプによって、メトリクスが決まります。 エクスプローラのKubernetes Namespace Overviewダッシュボードにドリルダウンすると、ポッド全体のネットワークアクティビティなどの詳細が表示されます。

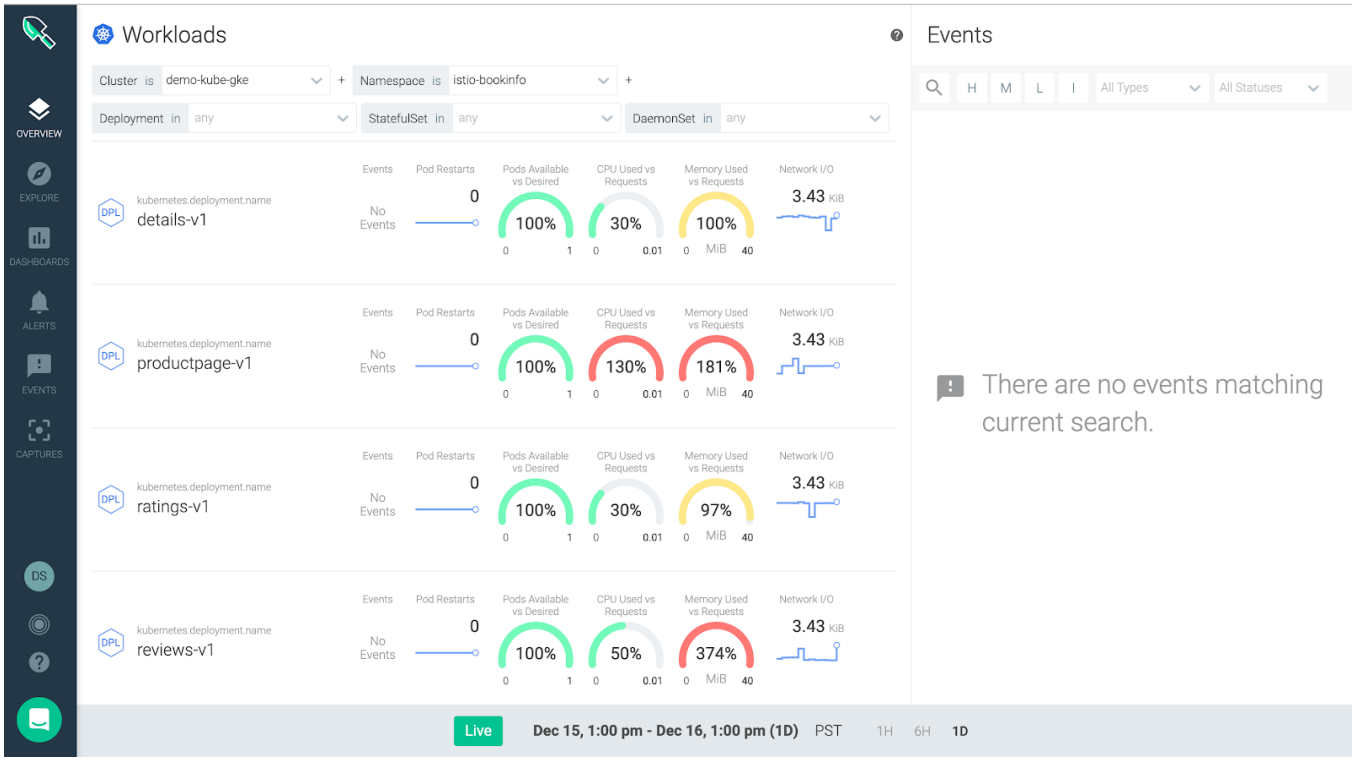
ワークロードデータ

このトピックでは、[ワークロードのOverview]ページについて説明し、そのゲージチャートとそこに表示されるデータを理解するのに役立ちます。

ワークロードのOverviewについて

ワークロードは、Kubernetesの用語では、コンテナ化されたアプリケーションを指します。 ワークロードは、ネームスペース内のデプロイメント、ステートフルセット、およびデーモンセットで構成されています。

クラスターでは、ワーカーノードがアプリケーションワークロードを実行しますが、マスターノードはアプリケーションワークロードにコアKubernetesサービスとオーケストレーションを提供します。[ワークロードのOverview]ページには、健全性、キャパシティ、コンプライアンスを示す主要なメトリクスが表示されます。



ワークロードデータの解釈

このトピックでは、[ワークロードのOverview]ページに表示されるメロちくすについて説明します。

ポッドの再起動

グラフには、`sum(timeAvg(kubernetes.pod.restart.rate))`によって返された最新の値が表示されます。

それは何ですか？

スパークラインは、選択したワークロード内のすべてのポッドにおけるポッド再起動率の傾向を示します。数値は、1秒あたりの再起動数で表される最新のレートを示します。
たとえば、下の画像は過去1時間の傾向を示しています。この数値は、ポッドの再起動率が最後の10分間で0.01未満であることを示しています。

参考までに、スパークラインは次のステップ数（サンプリング）を示しています。

- 最後の1時間：6ステップ、それぞれ10分のタイムスライス。
- 過去6時間：12ステップ、それぞれ20分のタイムスライス。
- 最終日：12ステップ、それぞれ2時間のタイムスライス。

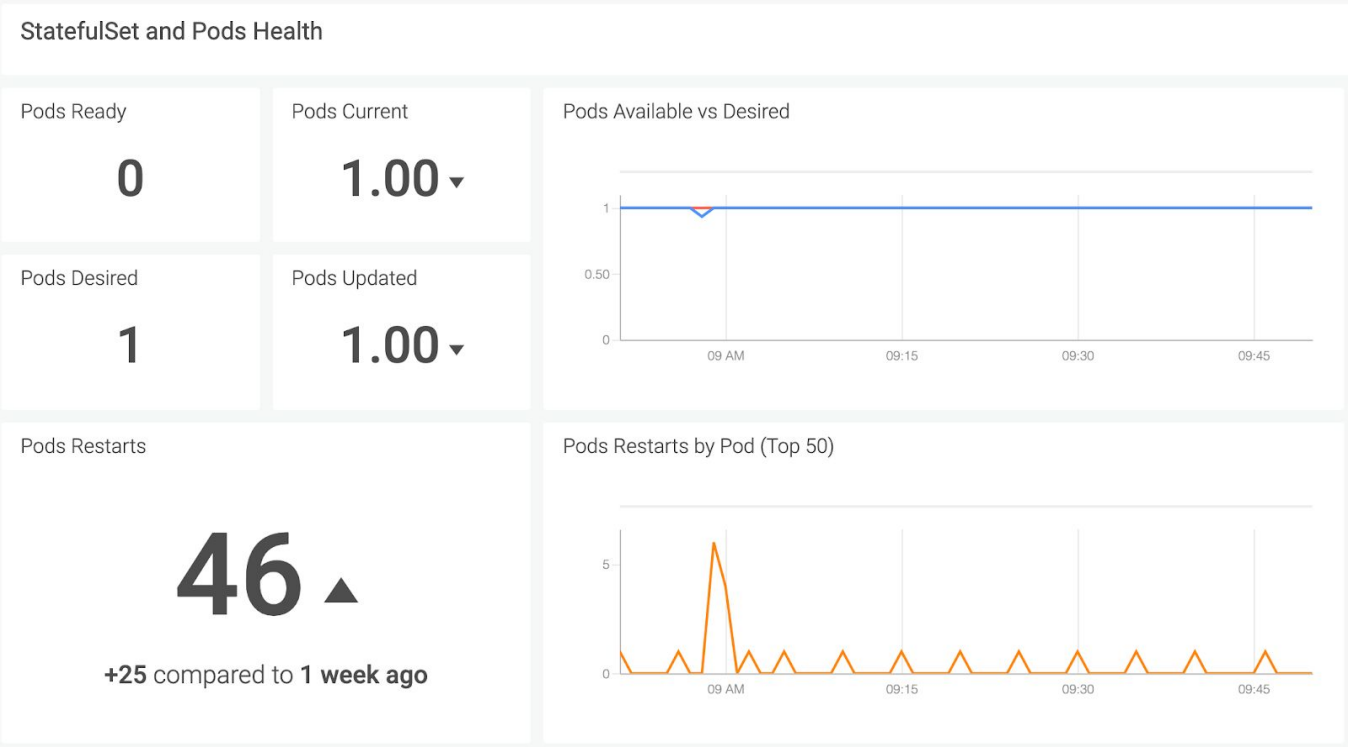
何を期待します？

正常なポッドの再起動は常に0になります。

他に何をすべきか？

ほとんどの場合、過去1時間の再起動が少ない（または時間枠が大きい）場合は、深刻な問題を示していません。エクスプローラーのワークロードに関連するKubernetes Overviewダッシュボードにドリルダウンします。たとえば、Kubernetes StatefulSet Overviewでは、ポッドごとの詳細なトレンドが提供

されます。



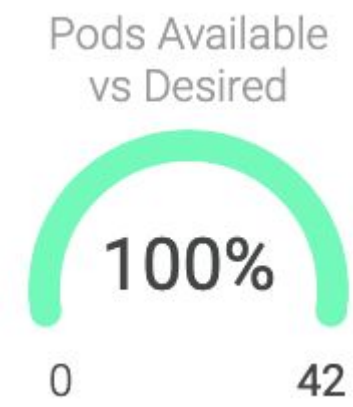
この例では、再起動の数は一定であり（ほぼ5分ごと）、ポッドの準備ができていません。これは、クラッシュループバックオフを示している可能性があります。

Pods Available vs Desired

グラフには、`sum(avg(kubernetes.deployment.replicas.available)) / sum(avg(kubernetes.deployment.replicas.desired))`によって返された最新の値が表示されます。

それは何ですか？

グラフには、特定のワークロード内のすべてのポッドについて、選択した時間ウィンドウ全体で平均化された、利用可能なポッドと望ましいポッドの比率が表示されます。
上限は、ワークロード内の必要なポッドの数を示します。
たとえば、以下の画像は、必要な42個のポッドがすべて利用可能であることを示しています。



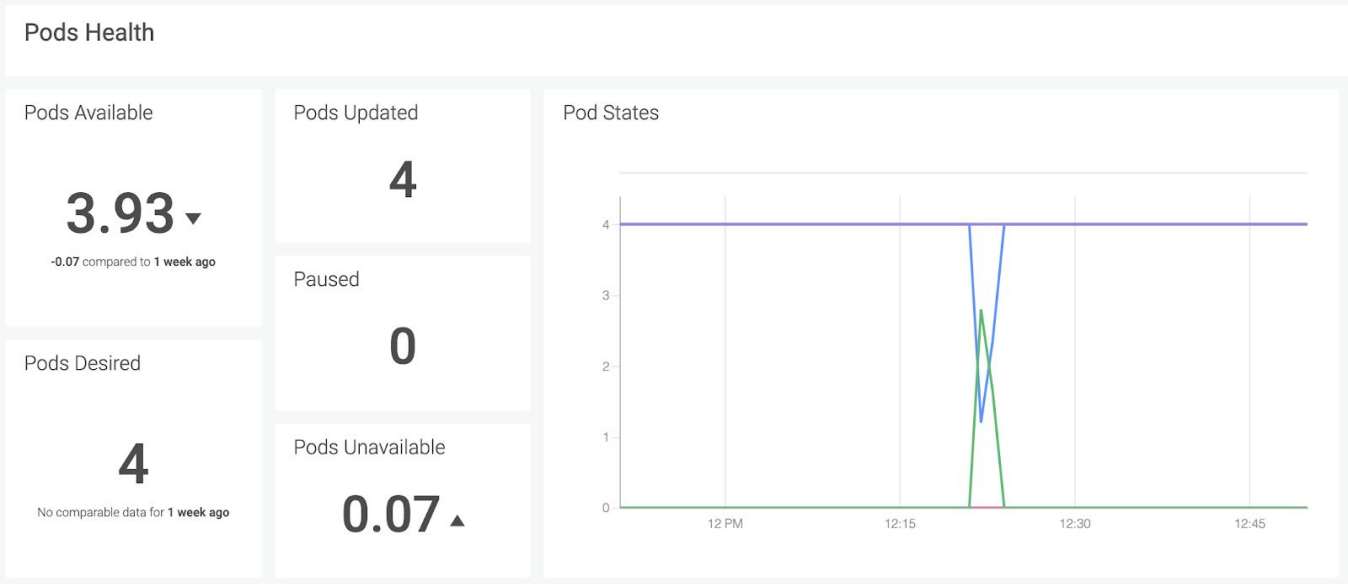
何を期待します？

通常は100%を期待する必要があります。

特定のポッドが使用可能になるまでにかなりの時間がかかる場合（イメージのプル時間、ポッドの初期化、readinessプローブ）、一時的に100%未満の比率が表示されることがあります。

他に何をすべきか？

エクスプローラで関連するダッシュボードにドリルダウンして、可用性の低いワークロードを特定します。たとえば、Kubernetes Deployment Overviewは、ポッドの傾向と状態を理解するのに役立ちます。



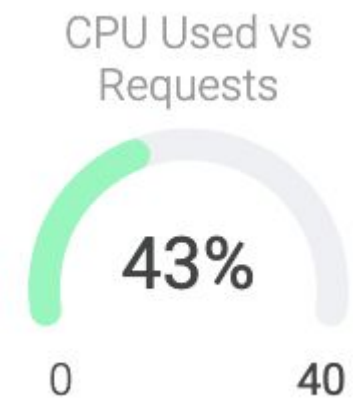
たとえば、上の画像は比率が98%（3.93 / 4 x 100）であることを示しています。 わずかな減少は、ポッドが終了し、その結果、新しいバージョンで開始された更新が原因です。

CPU Used vs Requests

グラフには、`sum(avg(cpu.cores.used)) / sum(avg(kubernetes.pod.resourceRequests.cpuCores))`によって返された最新の値が表示されます。

それは何ですか？

このグラフは、選択したワークロードのすべてのポッド全体の合計CPU使用率と、すべてのポッドによって要求された合計CPUの比率を示しています。
上限は、すべてのポッドによって要求された合計CPUを示します。値はCPUコアの数を示します。



この画像では、ワークロードのポッドが40 CPUコアを要求し、そのうち43%が実際に使用されています（約17コア）。

何を期待します？

ワークロードのタイプによって異なります。
リソース使用率が一定のアプリケーション（バックグラウンドプロセス）の場合、比率は約100%になると予想されます。
APIサーバーなどの「バースト性」アプリケーションの場合、比率は100%未満になると予想されます。値は選択した時間枠で平均化されるため、使用量の急増はアイドル期間によって補われることに注意してください。
通常、80%から120%の値は正常と見なされます。 100%を超える値は、比較的短時間だけ観察された場合に正常と見なされます。

他に何をすべきか？

- 使用率が低い場合は、アプリケーションが適切に実行されていない（予期された機能が実行されていない）か、ワークロードの構成が正確ではない（ポッドが実際に必要とするものと比較して要求が多すぎる）ことを示します。

- 使用率が高い場合は、アプリケーションの負荷が高いか、ワークロードの構成が正確でないことを示します（ポッドが実際に必要とするものと比較して要求が少ない）。

どちらの場合も、エクスプローラーのワークロードに対応するKubernetes Overviewダッシュボードにドリルダウンします。たとえば、Kubernetes Deployment Overviewダッシュボードは、リソースの使用状況と構成に関する洞察を提供します。

値は100%を超えることができますか？

はい、できます。

- 制限なしのCPUリクエストまたはリミットよりも低いリクエストの構成は許容されます。これらの場合、通常は一時的な過負荷を処理するために、コンテナがリクエストよりも多くのリソースを使用できるようにします。
- 2つのコンテナを持つワークロードを考えます。たとえば、1つのコンテナが1つのCPUコアをリクエストするように構成され、1つのCPUコアを使用しているとします（使用済みとリクエストの比率は100%です）。もう1つはリクエストなしで構成され、1つのCPUコアを使用します。この例では、2 CPUコアの1 CPUコアに対するリクエスト率は、ワークロードレベルで200%です。

「No Data」とはどういう意味ですか？

ワークロードがリクエストとリミットなしで構成されている場合、使用量とリクエストの比率は計算できません。この場合、グラフには「No Data」と表示されます。データエクスプローラーのダッシュボードにドリルダウンして、実際の使用状況を評価します。

常にリクエストを設定する必要があります。リクエストを設定すると、再構成が必要なワークロードを検出できます。

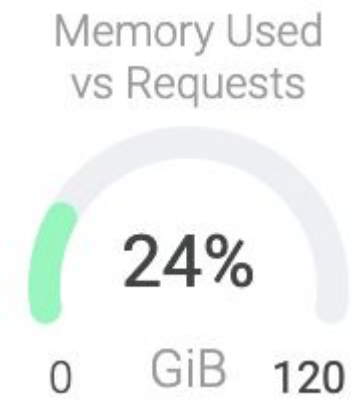
> **注意**
>
> Kubernetes自体は、リクエストやリミットが構成されていないワークロードを公開する可能性があります。 たとえば、kube-systemネームスペースには、リクエストを構成せずにワークロードを含めることができます。

Memory Used vs Requests

グラフには、`sum(avg(memory.bytes.used)) / sum(avg(kubernetes.pod.resourceRequests.memBytes))`によって返された最新の値が表示されます。

それは何ですか？

このグラフは、ワークロード内のすべてのポッド全体の合計メモリ使用量と、ワークロードによって要求された合計メモリの比率を示しています。
上限は、指定されたバイト単位で表された、すべてのポッドによって要求された合計メモリを示します。



たとえば、画像は、選択したワークロードのポッドが120 GiBを要求したことを示しています。そのうち24%が実際に使用されています（約29 GiB）。

何を期待します？

ワークロードのタイプによって比率が決まります。 80%から120%の値は正常と見なされます。 100%を超える値は、比較的短時間だけ観察された場合は正常と見なされます。

他に何をすべきか？

メモリ使用量が少ない場合は、アプリケーションが適切に実行されていない（予期された機能が実行されていない）か、ワークロードの構成が正確ではない（ポッドが実際に必要とするものと比較して要求が多すぎる）ことを示します。
高いメモリ使用量は、アプリケーションの負荷が高いか、ワークロードの構成が正確でないことを示します（ポッドが実際に必要とするものと比較して要求が少ない）。

注意

ワークロードに設定されたリミットとノードのメモリ負荷を考慮して、ワークロードがリクエストされたものより多くのメモリを使用する場合、それらは追い出されるリスクがあります。 詳細については、コンテナのメモリリミットを参照してください。

どちらの場合も、[ワークロード]ページにドリルダウンして、より詳細な分析が必要なワークロードを特定します。

値は100%を超えることができますか？

はい、できます。

- リミットなしのメモリリクエストまたはリミット未満のリクエストを構成することは許容されます。これらの場合、通常は一時的な過負荷を処理するために、コンテナがリクエストよりも多くのリソースを使用できるようにします。
- 2つのコンテナを持つワークロードを考えます。たとえば、1つのコンテナーが1 GiBのメモリをリクエストするように構成され、1 GiBを使用する（使用済みと要求の比率は100%）一方で、もう1つのコンテナはリクエストなしで構成され、1 GiBのメモリを使用します。この例では、1 GiBのリクエストされた比率に対して使用される2 GiBのメモリは、ワークロードレベルで200 %です。

「No Data」とはどういう意味ですか？

ワークロードがメモリリクエストとリミットなしで構成されている場合、使用量とリクエストの比率は計算できません。この場合、グラフには「No Data」と表示されます。データエクスプローラーのダッシュボードにドリルダウンして、実際の使用状況を評価します。
リクエストを設定する必要があります。再構成が必要なワークロードを検出するのに役立ちます。

> **注意**
>
> Kubernetes自体は、リクエストやリミットが構成されていないワークロードを公開する可能性があります。 たとえば、kube-systemネームスペースには、リクエストを構成せずにワークロードを含めることができます。

Network I/O

グラフには、`avg(avg(net.bytes.total))`によって返された最新の値が表示されます。

それは何ですか？

スパークラインは、ワークロードのネットワークトラフィック（受信および送信）の傾向を示します。 数値は、特定の単位の1秒あたりのバイト数で表される最新の速度を示します。

参考までに、スパークラインは次のステップ数（サンプリング）を示しています。

- 最後の1時間：6ステップ、それぞれ10分のタイムスライス
- 過去6時間：12ステップ、それぞれ30分のタイムスライス
- 最終日：12ステップ、それぞれ2時間のタイムスライス

何を期待します？

ワークロードで実行されるアプリケーションのタイプによって、メトリクスが決まります。 エクスプローラのワークロードに対応するKubernetes Overviewダッシュボードにドリルダウンします。たとえば、Kubernetes Deployment Overviewには、ポッド全体のネットワークアクティビティなどの追加の詳細が表示されます