

MIXI TECH CONFERENCE 2023

全世界で成長し続けるサービスのインフラと開発の裏側

清水 勲

Vantageスタジオ みてね事業部 SREグループ

#mixitechcon

About Me

清水 勲 @isaoshimizu

- 2011年～ 株式会社ミクシィ（現MIXI）
- 2011年～ SNS「mixi」運用エンジニア
- 2014年～ モンスターストライク SRE
- 2018年～ 家族アルバム みてね SRE
- 2022年～ SREグループ マネージャー



週末は社会人吹奏楽団での活動（楽団長、トロンボーン約30年、たまに指揮者）。
キャンプとクラフトビールが好き。

家族アルバム みてね

家族アルバム みてね



パパ・ママが撮った子どもの写真や動画を、祖父母や
親戚など招待した家族だけに簡単に共有できる写真・
動画共有アプリ

商品の例



フォトブック



写真プリント

家族アルバム みてね

みてねみまもりGPS

アプリの使い方 製品仕様 ヘルプ

【第2世代 50%OFF & 第3世代 15%OFF】入学・入園前に、子どもみまもり進化キャンペーン

CMで話題！
みてねみまもりGPS

学校も、買いもの、お出かけも、
みてねの位置情報みまもりサービス

ご購入はこちら

みてねみまもりGPS

みてね 年賀状

年賀アプリの決定版
みてね 年賀状

2023 謹賀新年

HAPPY NEW YEAR 2023

Happy New Year Love

HAPPY Together

2023 1 JAN

2023

みてね年賀状

みてね

ブログトップ 見まもりと、これから コラム みてね基金 ヘルプ みてねプレミアム

特別な日も、日常も
みてね出張撮影で家族写真を撮ろう

75歳以上
データで
もらえる！

家族への感謝を
みてねから
ワンタップ

30歳未満
キャンセル
無料

みてね出張撮影

みてねコールドクター

1020-897-146 WEBで診察予約はこちら

利用料金 診察の流れ 往診対応エリア 可能な診察・検査 提携医療機関 医師のご挨拶

PC/スマホにも対応！

大丈夫。
お医者さんが
来てくれる。

アプリで往診依頼、オンライン診療、医師相談
24時間365日、おうちで診療

まもなく医師が到着いたします

診察予約時間 21:50 頃

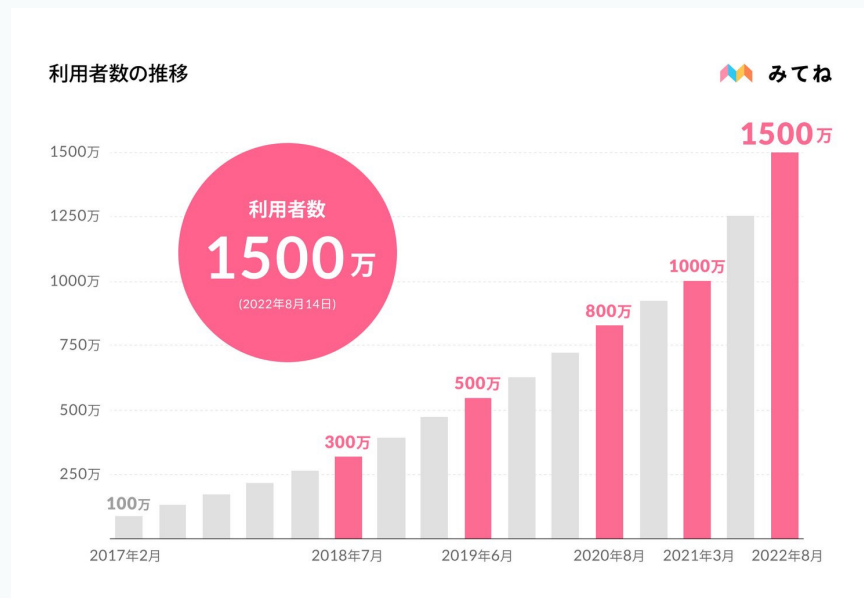
診察医師 村井 浩彦

予約完了を確認してください

みてねコールドクター

家族アルバム みてね

- 2015年4月リリース
- 現在は7言語・175の国と地域でサービスを提供
- 海外では「FamilyAlbum」という名称で展開中
- 2022年8月14日に利用者数が1,500万人※1 を突破
- 日本国内ではママやパパの半数となる47.1%の方※2 がご利用



※1 iOS・Android™ アプリ登録者数、ブラウザ版登録者数の合計

※2 「みてね」登録時に入力されたお子さまの誕生日と厚生労働省発表「人口動態統計」から算出。2022年8月時点で47.1%

システム構成

みてねを支えるクラウド



AWS

EKS / EC2 / S3 / Aurora
ElastiCache / SQS / SNS など

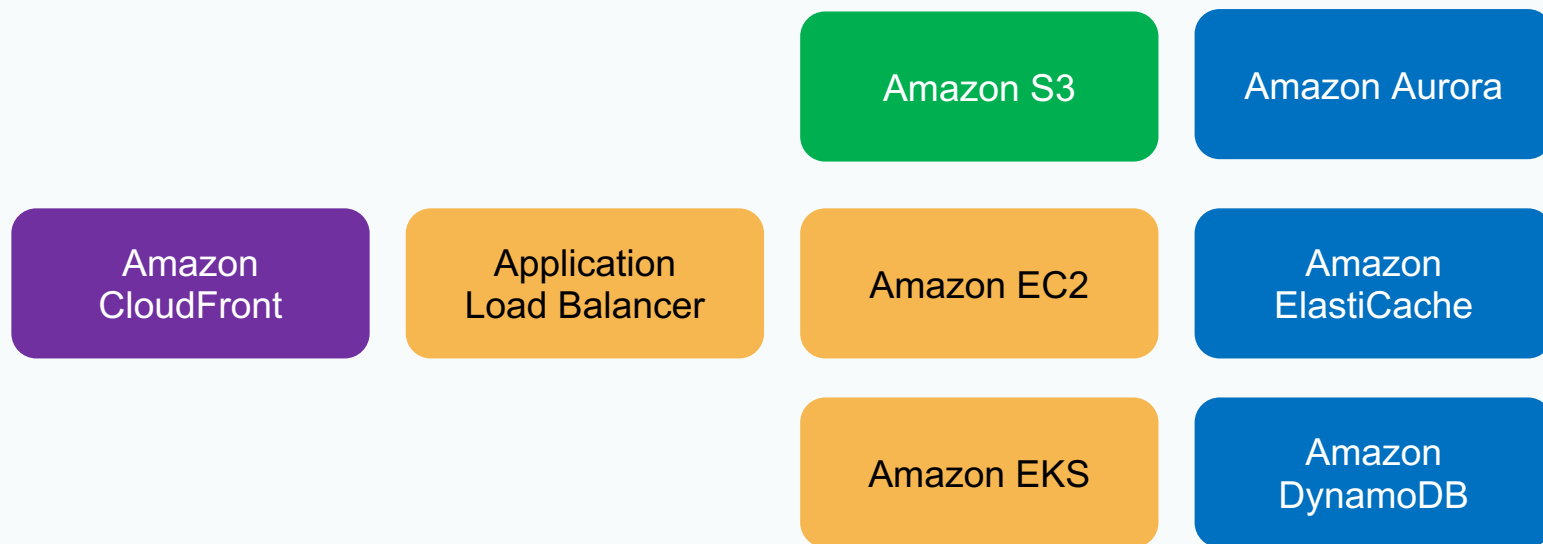


Google Cloud

BigQuery など

大半がAWSで構成されている

AWSの構成（概略図）



システム構成

- AWSアカウントは環境ごとに完全に分割
- AWSのリソースはTerraformによって管理
- ALB、S3の前面にCloudFrontを置き、ネットワーク経路の最適化とキャッシュ利用
- EKSクラスターは半年に一度アップデートする運用（新旧クラスター入れ替え）
- KubernetesのリソースはArgo CDとHelm Chartによって管理
- EC2は9割以上がスポットインスタンス（オンデマンドはごくわずか）
- EC2上ではRails Application、Sidekiq Workerが動作
- AWS Load Balancer ControllerとTargetGroupBindingによってALBからPodまでのルーティングをコントロール
- AuroraはGlobal Databaseを利用（詳細は後述）

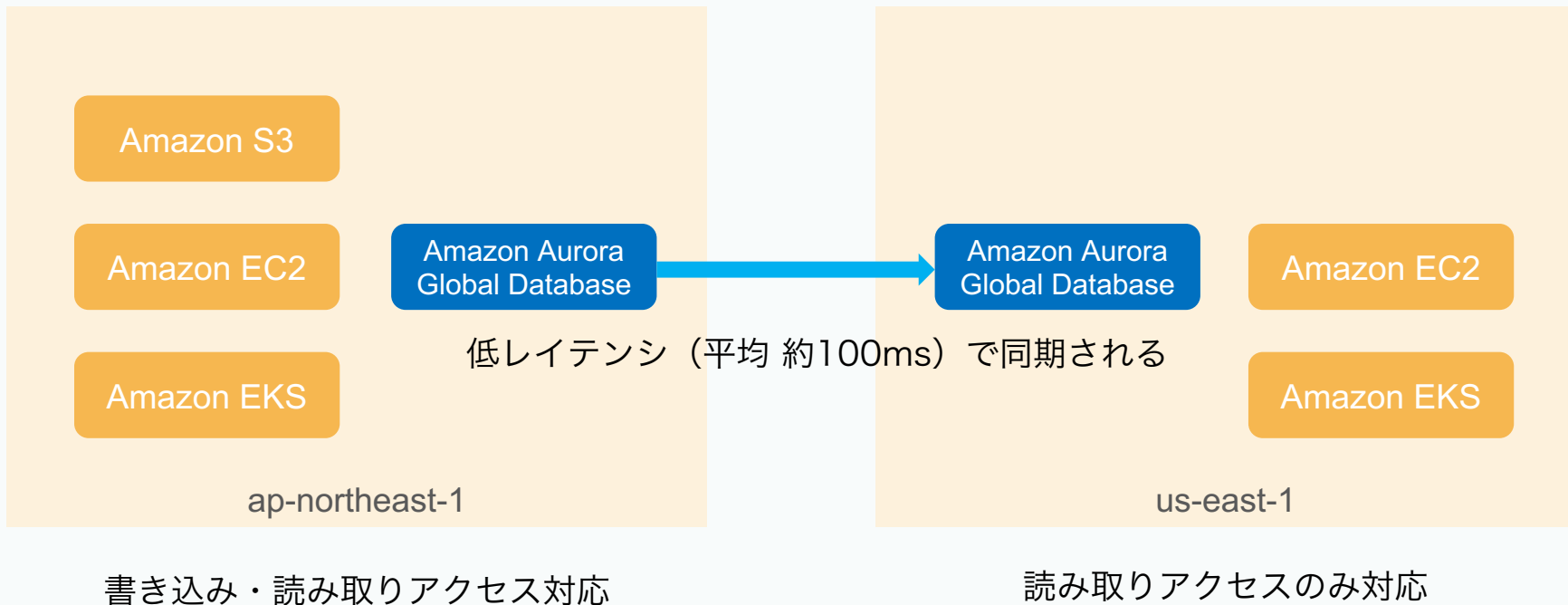
マルチリージョン構成

マルチリージョン構成



ユーザーから近いリージョンにアクセスすることで
レスポンスタイムを短くすることができる

マルチリージョン構成（データベース）

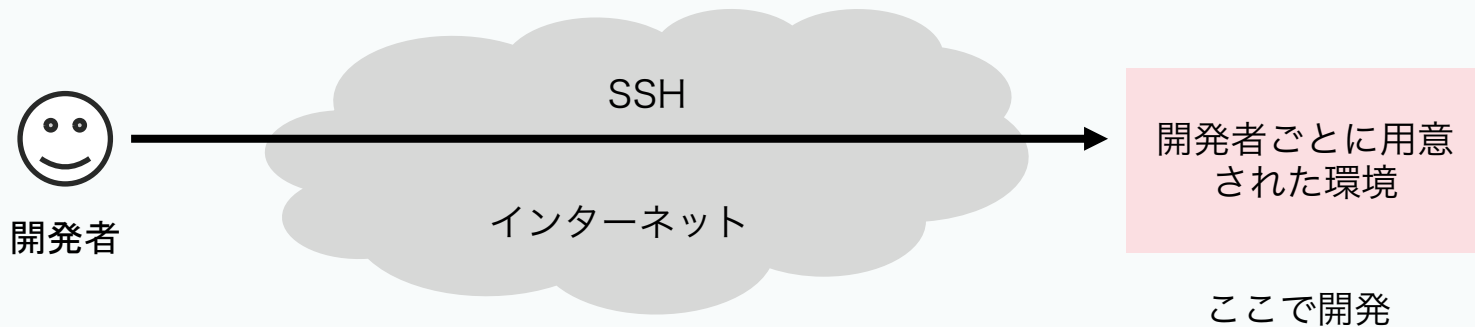


マルチリージョン構成における注意点

- みてねではユーザー体験の向上を目的としたマルチリージョン
 - いわゆるBCP対策ではない
- 各リージョンで同様の監視をおこなう必要がある
 - IaCによって手間はあまりかからない
- リージョンやAZによって提供されないサービス、インスタンスタイプがある
- EC2のキャパシティ状況はリージョン、AZによって大きく変わる
- リージョン間でネットワーク障害があるとデータベースの同期に大きな遅延が起きる

サーバーアプリケーションの 開発環境

サーバーアプリケーションの開発環境



とてもシンプルに書くとこんな感じ

サーバーアプリケーションの開発環境

Aさんの
開発環境Pod

Bさんの
開発環境Pod

Cさんの
開発環境Pod

Dさんの
開発環境Pod

Eさんの
開発環境Pod

Kubernetes

インターネット



Aさん



Bさん



Cさん



Dさん



Eさん

開発チーム

開発環境の作り方

ステップ1

YAMLファイルを
GitHubにプッシュ

※YAMLの中身はSSH
の公開鍵

ステップ2

Argo CDでSync操作を
して開発環境が起動

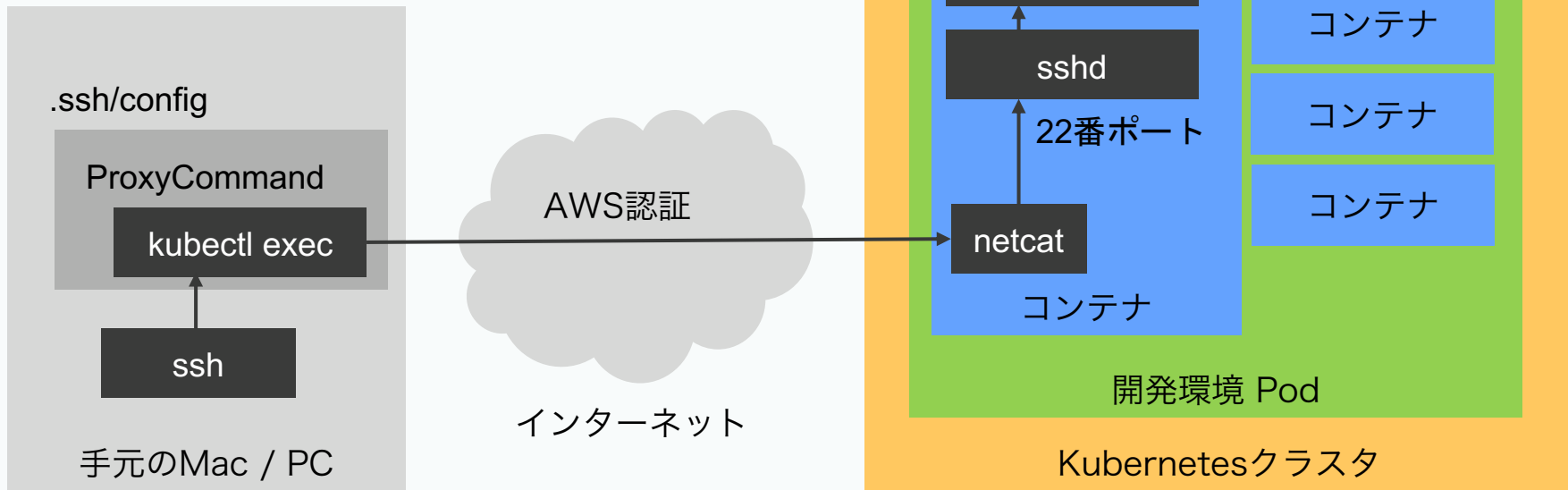
※数十秒後に環境が作成
される

ステップ3

SSHする

開発者自身で作成・削除を自由に行うことが可能

開発環境へのログイン周りのセキュリティ



SSHのポートをインターネットに公開する必要はない

開発環境が新しい構成になって

- 以前は個人ごとのEC2にSSHしてDockerを利用していた
 - SSHの鍵管理、Terraformの適用と構築のステップが多かった
 - **YAMLの追加、Argo CDの管理画面からの適用と簡単に**
 - 手順がわかりづらくSREメンバーのサポートが必要になったり
 - **誰でも簡単に構築・削除できるように**
 - 22番ポートのインターネット公開（公開鍵認証とはいえ危ない）
 - **ポート公開なしで、AWSの認証（SSO）と公開鍵認証の組み合わせに**
 - **VS Code Remote Developmentにも対応**
 - ログが溜まりすぎたり、検索しづらかったり
 - **Grafana Lokiで検索しやすく**

オブザーバビリティの工夫

オブザーバビリティはなぜ必要？

- システムで何が起きているのかを把握するため
- Kubernetesなどコンテナのオーケストレーションが利用されることが増えた結果、システム内部が複雑になり、各コンポーネントを観測可能な状態にする必要がある
- メトリクス、ログ、トレースなどの情報は多くあり、問題の原因を調査するためにはそれらを的確に集計し、可視化する仕組みが求められる

オブザーバビリティの実現

モバイルアプリ

New Relic Mobile
Firebase Crashlytics

サーバーアプリ

New Relic APM
New Relic Browser

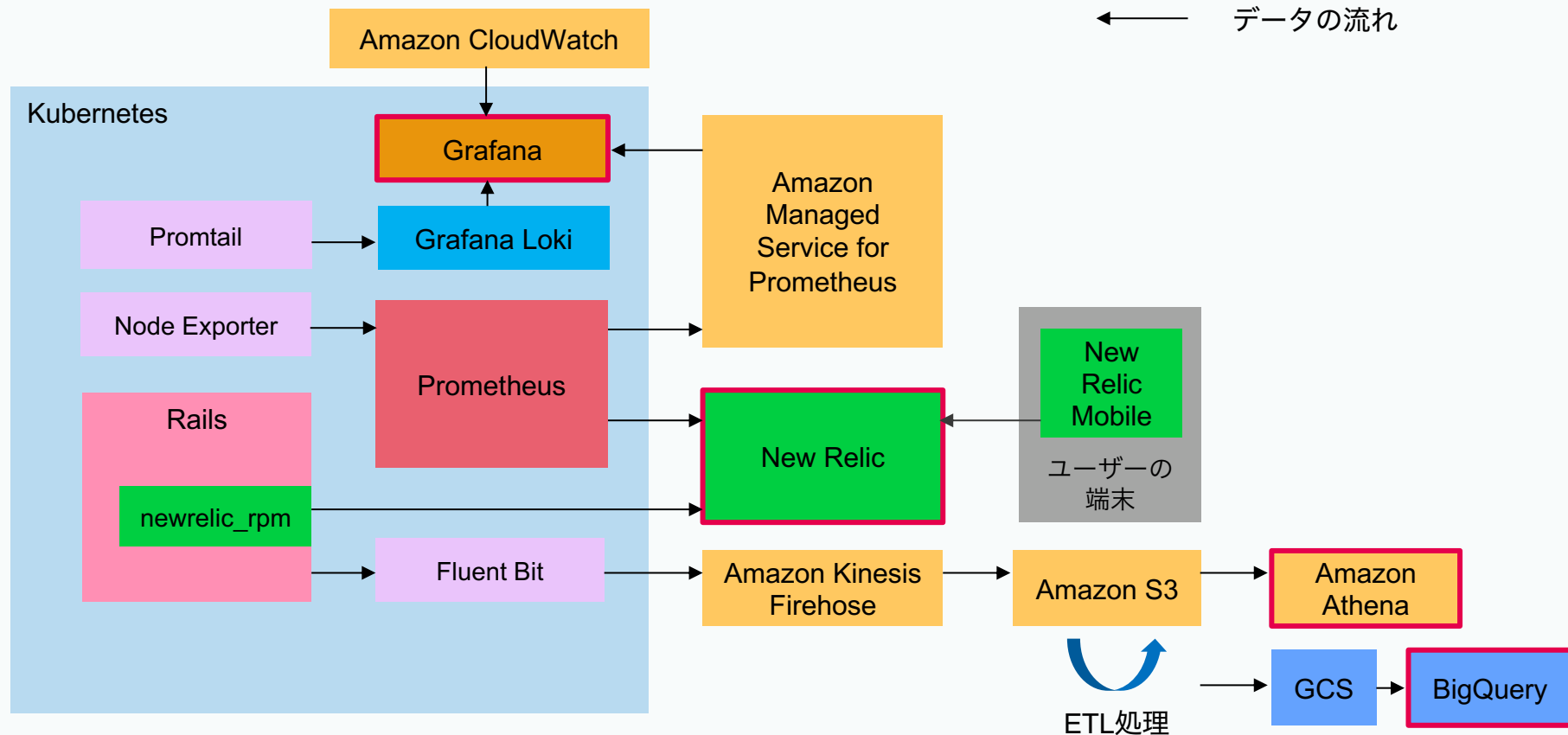
各種ログ

Grafana Loki
Amazon Athena
Google BigQuery

各種メトリクス

Prometheus / Amazon Managed
Service for Prometheus
Amazon CloudWatch
Grafana

オブザーバビリティの全体像



**SREを実践する上でも
オブザーバビリティはとても大事**

まとめ

まとめ

- 全世界で1,500万人以上のユーザーに利用されるサービスのシステム構成が参考になれば幸いです。
- Aurora Global Databaseのおかげでマルチリージョン構成を作ることができ、世界中のユーザー体験を改善できました。
- Kubernetesクラスタ上の開発環境はセキュアで扱いやすい環境になりました。
- オブザーバビリティは今まで試行錯誤を繰り返し、SREだけでなく開発者全体で利用できる環境を作ることができました。

WE ARE HIRING

さまざまなポジションで募集中です！

みてね 採用

検索