**Cursorへの指示:**

以下のタスクをお願いします。

1. **バックアップの検証とテスト:**
   * backup/data ディレクトリに作成されたバックアップデータの整合性をチェックするためのスクリプト (backup/check\_backup\_integrity.sh など) を作成してください。
   * 作成したバックアップデータを使用して、システムの復元テストを行うためのスクリプト (backup/run\_restore\_test.sh など) を作成し、実行してください。復元が正常に行われることを確認できるようなテスト内容を含めてください。
2. **監査レポートの可視化:**
   * audit/ ディレクトリに保存されている監査ログ (internal\_audit.sh や external\_audit.sh が生成するログファイル) をGrafanaで可視化するためのダッシュボード設定ファイル (monitoring/grafana/provisioning/dashboards/audit\_dashboard.json など) を作成してください。
   * 作成したダッシュボードで、監査ログの内容（例えば、アクセス状況、セキュリティイベントなど）を分かりやすく表示できるように設定してください。
   * 監査結果を分析し、レポートを作成するための基本的な手順や注目すべき点などをまとめたドキュメント (docs/audit\_analysis\_guide.md など) を作成してください。

**現在出来ていることのまとめ:**

* **基本的なアクセス制御機能が完了しました。**
  + ユーザー認証機能（JWTベース認証、パスワードハッシュ化、トークン有効期限管理）
  + ロールベースアクセス制御（RBAC）の実装（admin、developer、ai\_agentの3つのロールとそれぞれの権限設定）
  + 権限チェックミドルウェアの実装
* **セキュリティ機能が実装されています。**
  + パスワードのハッシュ化
  + JWTトークンの使用
  + 環境変数による設定管理
* **簡単なテストは完了しています。**

上記の内容をCursorに伝え、次のステップのコーディングを依頼してください。

**修正版 Cursor 実装計画**

**1. システム監視とリソース管理**

* **現状:** PrometheusとGrafanaの設定ファイル (monitoring/prometheus/prometheus.yml, monitoring/grafana/) は存在しますが、具体的な監視項目（CPU、メモリ、ストレージ、API応答時間）の設定やアラート設定の状況は不明です。node-exporter は導入されています。
* **実装内容（継続）:**
  + CPU使用率、メモリ使用率、ストレージ残量の監視設定をPrometheusに行い、それぞれ警告と危険アラートの閾値を設定 (CPU: 80%/95%, メモリ: 75%/90%, ストレージ: 残り 20%/10%)
  + Grafanaで監視ダッシュボードを作成し、上記メトリクスを可視化
  + API応答時間の監視設定をPrometheusに行い、警告と危険の閾値を設定 (200ms/500ms)
  + 設定した閾値に基づいてアラートを発報するようにPrometheus Alertmanagerを設定

**2. エラー処理・ロギング**

* **現状:** ELKスタック (logging/elk/) の設定ファイルは存在し、Nginxのログ (logs/nginx/error.log) は収集可能です。
* **実装内容（継続）:**
  + エラーレベル（軽微、中程度、重大）ごとの対応ロジックを設計・実装
  + 軽微なエラーに対する自動リカバリ（リロード、キャッシュクリア、再起動など）スクリプトの実装
  + 中程度の障害に対する依存サービス再接続、データ復元などの対応スクリプトの実装
  + 重大なエラー発生時に上位システムへ即時報告するための通知システム（Slack連携など、Alertmanagerと連携させることも検討）の実装
  + ログ記録の詳細度に基づいたログ収集設定（エラーログ、操作ログ、セキュリティログ）
  + 構造化ログシステム（JSON形式）への対応とKibanaでの可視化設定

**3. セキュリティ対策**

* **現状:** 基本的なアクセス制御（RBAC）が完了し、3つのロール（admin、developer、ai\_agent）とそれぞれの権限設定、権限チェックミドルウェアが実装済み。簡単なテストも完了。
* **実装内容（継続）:**
  + AES-256によるデータ保存時の暗号化方式を決定し、実装
  + TLS 1.3での通信暗号化設定（Nginxの設定は完了済み）
  + 必要に応じて、AWS IAM、Kubernetes RBAC、DBユーザー管理などの実装
  + 多要素認証（MFA）としてOAuth2 + Google Authenticator連携の実装

**4. バージョン管理とロールバック**

* **現状:** Gitによる基本的なバージョン管理は行われていると思われます（README.md ファイルの存在などから推測）。
* **実装内容（継続）:**
  + Gitのブランチ戦略（main, develop, feature, release, hotfix）を明確化し、運用ルールを整備
  + GitHub Actions を用いたCI/CDパイプラインの構築（コードの自動テスト、ビルド、デプロイなど）
  + AIモデルのバージョン管理にMLflowを導入し、管理・ロールバックの仕組みを実装

**5. データ管理**

* **現状:** 日次増分バックアップ、週次フルバックアップの具体的な実装状況は不明ですが、backup/ ディレクトリにバックアップ関連のスクリプト (backup.sh, restore.sh, test\_backup.sh) が存在します。
* **実装内容（継続）:**
  + UUID/GUIDを用いたデータの一意性確保の実装
  + データ整合性を定期的にチェックするスクリプト（Python + SQLなど）の実装
  + RPO（24時間以内）、RTO（2時間以内）を目標とした自動バックアップスクリプト（cron + rsync, AWS S3連携など）の実装

**6. 監査と評価**

* **現状:** 内部監査と外部監査のためのスクリプト (audit/internal\_audit.sh, audit/external\_audit.sh) が存在します。
* **実装内容（継続）:**
  + 四半期ごとの運用監査の結果をCSV/JSON形式で出力するスクリプトの作成（Python + Pandasなど）
  + 外部監査に必要な監査ログを自動的にエクスポートし、S3などのストレージに連携する仕組みの実装
  + 監査レポートを出力するためのスクリプトの作成
  + **次のステップ:** Grafanaダッシュボードを作成し、監査ログを可視化

**7. AI フェアネス監査**

* **現状:** 未着手
* **実装内容:**
  + LLMの応答ログを解析し、バイアスを測定する仕組みの検討と実装
  + IBM AI Fairness 360 の導入と設定
  + AIの出力ごとにユーザー評価（👍👎）を収集する機能の実装
  + 収集したユーザー評価を記録する機能の実装

**8. エスカレーションシステム**

* **現状:** 未着手
* **実装内容:**
  + AIの役割とリスクレベルに応じたエスカレーションルールを明確化（低リスク：AI自動対応、中リスク：部長AI、高リスク：人間の管理者）
  + Slackなどのコミュニケーションツールと連携したエスカレーションシステムの実装

**実装スケジュール (現状に基づく修正)**

* **即座に実装可能（1-2週間）**
  + **次のステップ:** バックアップの検証とテスト（整合性チェック、復元テストスクリプトの作成と実行）
  + **次のステップ:** 監査レポートの可視化（Grafanaダッシュボード設定ファイルの作成と基本的な表示設定）
  + システム監視スクリプトの具体的な設定（Prometheusアラート、Grafanaダッシュボードの詳細設定）
  + ログ管理システム（ELK）の具体的な設定とKibanaでの可視化
  + セキュリティ対策（データ暗号化、MFAの実装）
  + バージョン管理とロールバック（Gitブランチ戦略の運用開始、GitHub Actionsの基本設定）
  + データ整合性チェックとバックアップスクリプトの具体的な実装と自動化設定
* **中期（1-3ヶ月）**
  + AIフェアネス監査（IBM AI Fairness 360 の導入と初期設定、バイアス測定の仕組み実装）
  + エラー処理の自動化（自動修復スクリプトの本格的な実装とログ分析との連携）
  + 外部監査のデータ収集機能の実装と連携
  + GitHub Actions を用いたCI/CDパイプラインの本格的な構築
  + MLflow の設定と基本的なAIモデル管理機能の実装
* **長期（3ヶ月以上）**
  + 説明可能なAI（LIME, SHAPなどの技術調査と導入検討）
  + 倫理的AI設計のためのユーザーフィードバック収集とプロセスへの組み込み