**Epic 1: Bulk Agent Status Monitoring System**

**User Story 1: Configure Asterisk for bulk calling to automate real-time status monitoring**

* **Description**: Set up Asterisk to enable bulk calls for monitoring agent statuses and logging updates automatically.
* **Acceptance Criteria**:
  1. Asterisk is installed and configured for bulk calling.
  2. Bulk calls are successfully initiated to a predefined list of agents.
  3. Real-time agent status updates are captured in the monitoring system.

**User Story 2: Automate notifications for agent availability updates**

* **Description**: Create triggers and notifications for status changes to ensure teams are updated on agent availability.
* **Acceptance Criteria**:
  1. Notifications are triggered when an agent's status changes (e.g., offline, busy, available).
  2. Notifications are sent via email, Slack, or SMS to relevant stakeholders.
  3. Notifications can be customized based on specific thresholds (e.g., offline for 10 minutes).

**User Story 3: Integrate Asterisk with reporting tools for seamless agent performance reporting**

* **Description**: Automate the integration between Asterisk and reporting tools to generate performance reports.
* **Acceptance Criteria**:
  1. Reports are automatically generated on a daily/weekly basis.
  2. Reports include metrics like agent availability time, call statuses, and response times.
  3. Reports are accessible via the dashboard or emailed to stakeholders.

**Epic 2: TFN-IVR Performance Matrix Dashboard**

**User Story 1: Create automated data pipelines to feed TFN-IVR metrics into the dashboard**

* **Description**: Build automated data pipelines to gather and process TFN-IVR call flow metrics for visualization.
* **Acceptance Criteria**:
  1. Data pipelines collect real-time call metrics (e.g., call duration, routing success rate).
  2. Metrics are processed and stored in a format compatible with the dashboard.
  3. Pipelines handle data errors and retry failed processes automatically.

**User Story 2: Build a real-time dashboard to visualize TFN-IVR call flow metrics**

* **Description**: Create a dashboard in Grafana/Kibana to display key TFN-IVR metrics for actionable insights.
* **Acceptance Criteria**:
  1. Dashboard includes key metrics such as call volume, average handling time, and drop-off rates.
  2. Metrics are updated in real-time with minimal delay.
  3. The dashboard is accessible to stakeholders with role-based permissions.

**User Story 3: Implement alerting mechanisms for performance anomalies in TFN-IVR**

* **Description**: Set up alerts to notify stakeholders of unusual call flow performance.
* **Acceptance Criteria**:
  1. Alerts are triggered when metrics exceed predefined thresholds (e.g., drop-off rate > 10%).
  2. Notifications are sent via email or Slack when anomalies are detected.
  3. Alerts are logged and accessible for audit purposes.

**Epic 3: Unified Patch and Monitor System**

**User Story 1: Automate patch deployment for Windows and Linux servers**

* **Description**: Develop scripts to automate patch deployment across operating systems.
* **Acceptance Criteria**:
  1. Automated scripts schedule and deploy patches without manual intervention.
  2. Patching success/failure is logged and visible in the monitoring system.
  3. Rollback procedures are automated for failed patches.

**User Story 2: Integrate monitoring tools with the patching system**

* **Description**: Link patching workflows with monitoring tools for real-time compliance visibility.
* **Acceptance Criteria**:
  1. Monitoring tools display real-time patching status (e.g., pending, successful, failed).
  2. Alerts are generated for failed or missed patches.
  3. Compliance reports are auto-generated for audit purposes.

**User Story 3: Schedule automated patch deployment workflows**

* **Description**: Create workflows to schedule patches during non-peak hours to minimize downtime.
* **Acceptance Criteria**:
  1. Patch schedules can be configured via the system UI.
  2. Patches are deployed during defined maintenance windows.
  3. Notifications are sent to stakeholders before and after patch deployment.

**Epic 4: Centralized Syslog System**

**User Story 1: Configure Syslog servers for Windows and Linux systems**

* **Description**: Set up Syslog to collect and forward logs from multiple servers.
* **Acceptance Criteria**:
  1. Syslog servers are configured on both Windows and Linux systems.
  2. Logs are forwarded to a central server without data loss.
  3. Compatibility between Syslog servers and Kibana is verified.

**User Story 2: Automate log forwarding and error handling**

* **Description**: Automate the process of forwarding logs to Kibana and handle errors during transmission.
* **Acceptance Criteria**:
  1. Logs are forwarded to Kibana in real-time with minimal delay.
  2. Error handling is automated with retries for failed transmissions.
  3. Logs are tagged and indexed correctly in Kibana.

**User Story 3: Set up dashboards for centralized log visualization**

* **Description**: Create dashboards in Kibana for log visualization and monitoring.
* **Acceptance Criteria**:
  1. Dashboards display log data from all integrated servers.
  2. Search and filter functionality is enabled for efficient troubleshooting.
  3. Alerts are triggered for critical log events (e.g., server errors, unauthorized access).

**Epic 5: Engineering Insight Dashboards**

**User Story 1: Automate updates for Kibana and Grafana dashboards**

* **Description**: Write scripts to automatically update dashboard panels based on new data sources.
* **Acceptance Criteria**:
  1. Dashboards automatically reflect changes in underlying data sources.
  2. Updates to dashboard configurations are applied without manual intervention.
  3. Dashboards remain functional and error-free after updates.

**User Story 2: Customize dashboards for clearer insights and better user experience**

* **Description**: Optimize existing dashboards for usability and better visibility of critical metrics.
* **Acceptance Criteria**:
  1. Critical metrics are prioritized and displayed prominently.
  2. Dashboards are accessible on mobile and desktop devices with responsive layouts.
  3. Users can filter and drill down into metrics seamlessly.

**User Story 3: Implement alerts for critical metrics in dashboards**

* **Description**: Add alerting mechanisms to notify stakeholders of critical threshold breaches.
* **Acceptance Criteria**:
  1. Alerts are triggered when defined thresholds are breached (e.g., CPU usage > 90%).
  2. Notifications are sent via email or Slack with detailed information.
  3. Alert history is logged for analysis and audit purposes.

**Epic 6: DevOps Pipeline Automation**

**User Story 1: Set up CI/CD pipelines for application deployments**

* **Description**: Use Jenkins or Azure DevOps to automate application build and deployment processes.
* **Acceptance Criteria**:
  1. Pipelines are configured to build, test, and deploy applications automatically.
  2. Deployment steps are logged for traceability and debugging.
  3. Pipelines support rollback in case of deployment failures.

**User Story 2: Integrate automated testing into pipelines**

* **Description**: Add automated testing to ensure error-free deployments.
* **Acceptance Criteria**:
  1. Test cases are executed automatically during the pipeline run.
  2. Pipelines fail if critical test cases do not pass.
  3. Test results are logged and accessible for review.

**User Story 3: Automate pipeline monitoring and failure notifications**

* **Description**: Implement monitoring and notifications for pipeline performance and failures.
* **Acceptance Criteria**:
  1. Pipelines are monitored in real-time, and performance metrics (e.g., duration, success rate) are logged.
  2. Notifications are sent to stakeholders on pipeline failures with error details.
  3. Alerts are logged and categorized for future analysis.

**Epic 7: Certificate Management System**

**User Story 1: Build a tool to track SSL/TLS certificate expiry**

* **Description**: Develop a monitoring tool to track certificate expiry and notify teams in advance.
* **Acceptance Criteria**:
  1. The tool tracks all active certificates and their expiry dates.
  2. Notifications are sent 30, 15, and 5 days before expiry.
  3. Expiry details are logged and accessible via the dashboard.

**User Story 2: Automate certificate renewal processes**

* **Description**: Create automation scripts for SSL/TLS certificate renewal using APIs or other tools.
* **Acceptance Criteria**:
  1. Certificates are automatically renewed before expiry without manual intervention.
  2. Renewed certificates are deployed to the relevant systems automatically.
  3. Renewal logs are maintained for audit purposes.

**User Story 3: Integrate certificate management with notification systems**

* **Description**: Link the certificate management tool with notification platforms (e.g., Slack, email).
* **Acceptance Criteria**:
  1. Notifications include details about expiring certificates and next steps.
  2. Notifications are customizable based on urgency or team preferences.
  3. Notification history is logged for tracking purposes.

**Epic 8: Disaster Recovery (DR) Setup in Azure**

**User Story 1: Automate DR infrastructure provisioning in Azure**

* **Description**: Use Terraform to automate the setup of DR infrastructure in Azure.
* **Acceptance Criteria**:
  1. Terraform scripts provision DR infrastructure (e.g., virtual networks, storage, VMs) automatically.
  2. DR setup adheres to organizational policies and standards.
  3. Infrastructure is validated post-provisioning for availability and performance.

**User Story 2: Automate DR drills to validate recovery processes**

* **Description**: Schedule and execute automated DR drills to test the recovery process.
* **Acceptance Criteria**:
  1. DR drills simulate outages and validate the recovery of critical systems.
  2. Drill results are logged, including recovery time and success rate.
  3. Stakeholders are notified of drill outcomes and potential improvements.

**User Story 3: Implement monitoring for DR failovers**

* **Description**: Set up monitoring and alerting for DR failovers to ensure smooth transitions during disasters.
* **Acceptance Criteria**:
  1. Monitoring tools track failover performance metrics (e.g., downtime, failover success).
  2. Alerts are triggered during failover events with detailed logs.
  3. Reports on failover performance are auto-generated for review.