**Model Deployment Plan**

Model Deployment Plan using AWS SageMaker:

**Data Preparation on AWS**:

* Upload the pre-processed data and trained logistic regression model to an Amazon Simple Storage Service (S3) bucket.
* Ensure proper permissions are set to allow SageMaker to access the data and model artifacts.

**Model Packaging with SageMaker**:

* Use SageMaker's built-in containers for Scikit-learn to package the logistic regression model along with any necessary preprocessing code.
* Create a SageMaker model object specifying the location of the model artifacts in S3 and the Docker image containing the model code.

**Model Deployment Configuration:**

* Choose the appropriate instance type and number of instances for the SageMaker endpoint based on expected usage and resource requirements.
* Configure endpoint auto-scaling to automatically adjust the number of instances based on incoming traffic patterns.

**Endpoint Deployment:**

* Deploy the SageMaker endpoint using the configured model and instance settings.
* Monitor the deployment process through the SageMaker console or AWS Command Line Interface (CLI).

**API Development with AWS Lambda and API Gateway:**

* Create an AWS Lambda function to handle incoming HTTP requests and invoke the SageMaker endpoint for inference.
* Configure an API Gateway endpoint to route incoming requests to the Lambda function, providing a RESTful API interface for accessing the deployed model.

**Security Setup:**

* Implement security best practices, such as encrypting data in transit and at rest, and configuring IAM roles with least privilege access for SageMaker and other AWS services.
* Set up VPC endpoints for private access to SageMaker resources, if required, to enhance network security.

**Monitoring and Logging:**

* Utilize Amazon CloudWatch to monitor the performance and health of the SageMaker endpoint, tracking metrics such as invocation count, latency, and error rates.
* Configure CloudWatch alarms to alert on anomalous behaviour or performance degradation.

**Continuous Integration/Continuous Deployment (CI/CD):**

* Implement CI/CD pipelines using AWS Code Pipeline and AWS Code Deploy to automate the deployment process for updates to the model or infrastructure configuration.
* Integrate automated testing into the CI/CD pipeline to validate model functionality and performance before promoting changes to production.

**Documentation and Training:**

* Provide comprehensive documentation on how to interact with the deployed model through the API Gateway endpoint, including sample requests and responses.
* Conduct training sessions for relevant personnel on using SageMaker and accessing the deployed model for inference.

**Cost Optimization:**

* Utilize SageMaker's cost management features, such as Spot Instances and Managed Spot Training, to reduce infrastructure costs while maintaining high availability and performance.
* Monitor resource utilization and adjust instance types or scaling policies as needed to optimize costs without sacrificing performance.

By following this deployment plan using AWS SageMaker, organizations can efficiently deploy and manage the NLP model for predicting emotion intensity in text reviews, leveraging the scalability, reliability, and security of AWS cloud services.