1. Document troubleshooting steps for potential issues that may arise during deployment.

1. Deployment Failures:

Issue: Deployment fails with no clear error message.

Troubleshooting:

Check deployment logs for error details.

Review build and deployment scripts for syntax errors.

Ensure the repository and branch are correctly configured in the deployment settings.

2. Environment Variable Issues:

Issue: Application isn't functioning properly due to incorrect environment variable settings.

Troubleshooting:

Verify that environment variables are correctly configured in the Azure portal.

Check if you're using the correct variable names in your code.

Ensure the values are properly formatted and don't contain typos.

3. Application Not Accessible:

Issue: The deployed application isn't accessible via the provided URL.

Troubleshooting:

Check if the deployment is complete; it might take a few minutes.

Verify that the app name and URL are correctly configured in the Azure portal.

Ensure that your domain settings and DNS records are correctly configured.

4. Connection Errors:

Issue: Your application fails to connect to external services (e.g., databases, APIs).

Troubleshooting:

Verify that environment variables containing connection strings are accurate.

Check for IP restrictions or firewall rules that might be blocking outgoing connections.

Test the connection manually from within the application's environment.

5. Incorrect Deployment Branch:

Issue: The application deploys from the wrong branch, causing code discrepancies.

Troubleshooting:

Review deployment settings and ensure the correct branch is selected.

Check your repository's webhook or trigger settings for proper branch selection.

6. Missing Dependencies:

Issue: Your application relies on certain dependencies that are missing or not installed.

Troubleshooting:

Double-check your deployment scripts to ensure they include commands to install dependencies.

Verify that the required dependencies are correctly listed in your project's configuration files (e.g., requirements.txt, package.json).

7. Performance Issues:

Issue: Your application experiences slow performance or crashes under load.

Troubleshooting:

Use Azure Monitor to identify performance bottlenecks and resource utilization.

Consider upgrading your App Service plan to handle increased traffic.

Optimize your application code and database queries for better performance.

8. Configuration Errors:

Issue: Your application behaves unexpectedly due to incorrect configuration settings.

Troubleshooting:

Review your application's configuration files for any errors or inconsistencies.

Compare your local configuration with the deployed configuration to identify discrepancies.

9. SSL/TLS Certificate Issues:

Issue: You encounter issues with SSL/TLS certificates or HTTPS configuration.

Troubleshooting:

Ensure that you've configured the SSL settings correctly in the Azure portal.

Check that your certificate is valid and matches the configured domain.

10. Outdated Cached Content:

- Issue: Your application is serving outdated cached content to users.

- Troubleshooting:

- If your application uses caching, ensure you're properly clearing or invalidating the cache when content changes.

- Check your application's caching mechanisms and consider adjusting cache expiration times.