

<i>Definition</i>	<i>Practices around developing IaC</i>	<i>Build, tests, publish IaC build artifacts</i>	<i>Using IaC to provision infrastructure</i>	<i>Managing running infrastructure</i>	<i>Monitor and measure infrastructure</i>
<b>Area of Practice</b>	<b>Development</b>	<b>Continuous Integration</b>	<b>Provisioning</b>	<b>Management</b>	<b>Observability</b>
<b>Level 3 - Optimizing</b> Focus on process improvement	<ul style="list-style-type: none"> <li>Continual improvement and optimization of IaC based on evolving industry standards</li> </ul>	<ul style="list-style-type: none"> <li>Continuous improvement of tests at various levels</li> </ul>	<ul style="list-style-type: none"> <li>Zero-downtime provisioning of infra</li> <li>Ability to automatically roll back changes</li> <li>Self-service provisioning</li> </ul>	<ul style="list-style-type: none"> <li>Infrastructure is self-healing, self-configurable, and self-optimizing</li> </ul>	<ul style="list-style-type: none"> <li>Metrics are regularly reviewed</li> <li>Metrics are available in real-time</li> <li>Production incidents related to infrastructure are rare and always reviewed</li> </ul>
<b>Level 2 - Managed</b> Processes measured and controlled	<ul style="list-style-type: none"> <li>All changes are tracked in an Application Lifecycle Management tool</li> <li>All defects and bugs are tracked in the ALM</li> <li>Code is declarative</li> </ul>	<ul style="list-style-type: none"> <li>Builds are not left broken</li> <li>Changes are always promoted through a consistent path to production</li> </ul>	<ul style="list-style-type: none"> <li>Ability to manually roll back changes quickly and safely</li> </ul>	<ul style="list-style-type: none"> <li>Infrastructure is highly available and fault tolerant</li> <li>Configuration drift is impossible</li> </ul>	<ul style="list-style-type: none"> <li>Automated alerting based on active monitoring</li> <li>IaC processes and practices are documented and available</li> </ul>
<b>Level 1 - Consistent</b> Processes characterized and proactive	<ul style="list-style-type: none"> <li>All infrastructure is defined as code</li> <li>All IaC under version control</li> <li>Industry-standard tooling is used to write code</li> </ul>	<ul style="list-style-type: none"> <li>CI Server to pull, build, test, and publish IaC artifacts</li> <li>Automated tests are run for every check-in</li> <li>Test are run in a 'prod-like' environment</li> </ul>	<ul style="list-style-type: none"> <li>Result of an automated delivery pipeline</li> <li>Provisioning is idempotent</li> </ul>	<ul style="list-style-type: none"> <li>Immutable infrastructure (<i>no SSHing into boxes</i>)</li> <li>Infrastructure is reliable and performs predictably</li> </ul>	<ul style="list-style-type: none"> <li>Metrics are calculated automatically but not regularly reviewed</li> <li>Centralized infrastructure monitoring and logging</li> </ul>
<b>Level 0 - Repeatable</b> Processes characterized but often reactive	<ul style="list-style-type: none"> <li>Infrastructure partially automated using scripts</li> <li>Not all is checked into VCS</li> <li>Automation doesn't rely on industry-standard tooling</li> </ul>	<ul style="list-style-type: none"> <li>IaC tests are only run locally</li> </ul>	<ul style="list-style-type: none"> <li>Provisioning is scripted but executed ad-hoc</li> </ul>	<ul style="list-style-type: none"> <li>Patching and upgrades are done through provisioning processes</li> </ul>	<ul style="list-style-type: none"> <li>Metrics are defined, but no way to collect or consistently measure</li> <li>Decentralized infrastructure monitoring and logging</li> </ul>
<b>Level -1 - Regressive</b> Processes unrepeatable, poorly controlled, and reactive	<ul style="list-style-type: none"> <li>Nothing is stored in VCS</li> <li>Scripts are stored on infra, local workstations, or as notes</li> </ul>	<ul style="list-style-type: none"> <li>No Continuous Integration Server</li> <li>No written IaC tests</li> <li>No way to test infrastructure before provisioning</li> </ul>	<ul style="list-style-type: none"> <li>Infrastructure is built manually from command line or from a UI</li> <li>Existing infrastructure cannot be easily rebuilt</li> <li>Provisioning new infrastructure is painful and inconsistent</li> </ul>	<ul style="list-style-type: none"> <li>Existing infrastructure is brittle and unreliable</li> <li>Patching and upgrades are done directly on running infra</li> <li>Troubleshooting is done directly on running infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>No defined infrastructure metrics: SLAs, KPIs, CSFs</li> <li>Monitoring and logging done directly on running infra</li> <li>No automated alerting</li> </ul>

\* Assumes all practices follow security requirements and best practices