Definition	*Practices around developing IaC*	*Build, tests, publish IaC build artifacts*	*Using IaC to provision infrastructure*	*Managing running infrastructure*	*Monitor and measure infrastructure*
Area of Practice	**Development**	**Continuous Integration**	**Provisioning**	**Management**	**Observability**
Level 3 - Optimizing Focus on process improvement	• Continual improvement and optimization of IaC based on evolving industry standards	• Continuous improvement of tests at various levels	• Zero-downtime provisioning of infra • Ability to automatically roll back changes • Self-service provisioning	• Infrastructure is self-healing, self-configurable, and self-optimizing	• Metrics are regularly reviewed • Metrics are available in real-time • Production incidents related to infrastructure are rare and always reviewed
Level 2 - Managed Processes measured and controlled	• All changes are tracked in an Application Lifecycle Management tool • All defects and bugs are tracked in the ALM • Code is declarative	• Builds are not left broken • Changes are always promoted through a consistent path to production	• Ability to manually roll back changes quickly and safely	• Infrastructure is highly available and fault tolerant • Configuration drift is impossible	• Automated alerting based on active monitoring • IaC processes and practices are documented and available
Level 1 - Consistent Processes characterized and proactive	• All infrastructure is defined as code • All IaC under version control • Industry-standard tooling is used to write code	• CI Server to pull, build, test, and publish IaC artifacts • Automated tests are run for every check-in • Test are run in a 'prod-like' environment	• Result of an automated delivery pipeline • Provisioning is idempotent	• Immutable infrastructure (no SSHing into boxes) • Infrastructure is reliable and performs predictably	• Metrics are calculated automatically but not regularly reviewed • Centralized infrastructure monitoring and logging
Level 0 - Repeatable Processes characterized but often reactive	• Infrastructure partially automated using scripts • Not all is checked into VCS • Automation doesn't rely on industry-standard tooling	• IaC tests are only run locally	• Provisioning is scripted but executed ad-hoc	• Patching and upgrades are done through provisioning processes	• Metrics are defined, but no way to collect or consistently measure
Level -1 - Regressive Processes unrepeatable, poorly controlled, and reactive	• Nothing is stored in VCS • Scripts are stored on infra, local workstations, or as notes	• No Continuous Integration Server • No written IaC tests • No way to test infrastructure before provisioning	• Infrastructure is built manually from command line or from a UI • Existing infrastructure cannot be easily rebuilt • Provisioning new infrastructure is painful and inconsistent	• Existing infrastructure is brittle and unreliable • Patching and upgrades are done directly on running infra • Troubleshooting is done directly on running infrastructure	• No defined infrastructure metrics: SLAs, KPIs, CSFs • Monitoring and logging done directly on running infra • No automated alerting

** Assumes all practices follow security requirements and best practices*