УДК 004.896

ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ИНТЕГРАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ В ПРОЦЕСС РАЗРАБОТКИ ИНФРАСТРУКТУРЫ

Меркулов А. Е.[1]
Научный руководитель – канд. техн. наук, старший преподаватель факультета прикладной информатики Береснев А. Д.[1]
[1]Университет ИТМО
ae-m7@mail.ru
artem.beresnev@itmo.ru

Введение

Современный этап развития DevOps характеризуется переходом от реактивного управления инфраструктурой к проактивному [1], что требует качественно новых подходов к автоматизации. Однако анализ отечественной и зарубежной практики показывает: несмотря на появление специализированных ИИ-инструментов (Project Infragraph, AWS DevOps Agent, Ansible Lightspeed), их применение в области инфраструктуры как кода (IaC) носит фрагментарный характер и не решает фундаментальных проблем. Зарубежные решения либо привязаны к конкретным экосистемам (AWS), либо демонстрируют низкое качество генерации кода, требующее ручной доработки. Российский рынок находится на стадии активного внедрения ИИ-инструментов, но системные подходы к интеграции интеллектуальных агентов в процессы разработки IaC отсутствуют. Ключевое противоречие заключается в том, что скорость генерации кода современными LLM опережает способность гарантировать его безопасность [2] и соответствие best practices, что создает новые риски при эксплуатации.

Основная часть

Ключевая часть работы заключалась в практическом тестировании доступных инструментов. Эксперименты с Ansible Lightspeed выявили его неспособность корректно интерпретировать предметную область (генерация кода для Minio свелась к созданию виртуальных машин), склонность к предложению небезопасных конфигураций и необходимость значительной ручной доработки сгенерированного кода [3]. Попытка использования Ansible Code Bot оказалась неудачной из-за неработоспособности инструментария на стороне провайдера. На этом фоне было проведено сравнительное тестирование четырех универсальных LLM-моделей (Llama 3.3 70B, MiMo-V2-Flash, Gemini 2.5 Flash, DeepSeek R1T2 Chimera) на задаче анализа безопасности намеренно уязвимого Ansible-плейбука. Результаты показали, что три из четырех моделей обнаруживают 13 из 14 уязвимостей (полнота 92%) при полном отсутствии ложных срабатываний (точность 100%), что существенно превосходит качество работы специализированных коммерческих инструментов.

На основе полученных данных сформировано предложение по созданию мультиагентной системы интеллектуальной поддержки жизненного цикла IaC. Оптимальность решения заключается в отказе от разработки узкоспециализированных моделей в пользу использования универсальных LLM в качестве интеллектуального ядра, доступного через открытые протоколы типа MCP. Архитектура включает четыре типа агентов: статического анализа безопасности (интегрируется в IDE и процесс ревью pull request), прогнозирования дефектов (анализирует метаданные и историю изменений для оценки рисков деплоя), управления конфигурационным дрейфом (в реальном времени сопоставляет код с фактическим состоянием среды и генерирует

корректирующие плейбуки) и контекстно-осведомленной генерации кода (учитывает документацию, историю изменений и данные мониторинга).

Экономическая эффективность достигается за счет использования открытых LLM-моделей (например, через Ollama) и открытых протоколов, что позволяет разворачивать систему в собственном контуре без лицензионных отчислений [4]. Новизна подхода заключается в разработке методов оркестрации специализированных агентов вокруг универсального языкового ядра, обеспечивающих сквозной контроль качества и безопасности на всех этапах жизненного цикла IaC — от локальной разработки до постдеплойного мониторинга. Такая архитектура позволяет трансформировать процесс создания инфраструктурного кода из рутинного ремесла в управляемую инженерную дисциплину с минимальным участием человека в рутинных операциях.

Выводы

Результаты исследования могут быть использованы в деятельности DevOps-команд, platform-инжиниринговых и SRE-подразделений, занимающихся разработкой и сопровождением инфраструктуры как кода. Сформированные требования к мультиагентной системе позволяют приступить к созданию прототипа, который может быть интегрирован в существующие процессы разработки в качестве надстройки над текущими инструментами (Ansible, Terraform, CI/CD-пайплайны) [5].

Испытания эффективности предполагают сравнение ключевых метрик (количество инцидентов безопасности, время устранения дефектов, частота конфигурационного дрейфа, доля ручного труда) с базовыми показателями, полученными в ходе данного исследования.

Литература

1. Искандарова С. А. ИИ-АГЕНТЫ В КОРПОРАТИВНОМ УПРАВЛЕНИИ: АРХИТЕКТУРНЫЕ РЕШЕНИЯ И ПРАКТИКИ ВНЕДРЕНИЯ // Вестник науки. 2025. №6 (87). URL: https://cyberleninka.ru/article/n/ii-agenty-v-korporativnom-upravlenii-arhitekturnye-resheniya-i-praktiki-vnedreniya (дата обращения: 13.01.2026).
2. Claus Pahl, Niyazi Gokberk Gunduz, ¨ Ovg¨um Can Sezen, Ali Ghamgosar and Nabil El Ioini. Infrastructure as Code: Technology Review and Research Challenges: — Текст электронный // scitepress.org: [сайт]. — 2025. — URL: https://www.scitepress.org/Papers/2025/132477/132477.pdf (дата обращения: 12.01.2026).
3. Priyam Sahoo, Saurabh Pujar, Ganesh Nalawade, Richard Gebhardt, Louis Mandel, Luca Buratti. Insights from the Usage of the Ansible Lightspeed Code Completion Service: — Текст электронный // arxiv.org: [сайт]. — 2024. — URL: https://arxiv.org/html/2402.17442v4 (дата обращения: 28.12.2025).
4. Zhe Hou. Vibe Coding an LLM-powered Theorem Prover: — Текст электронный // arXiv.org: [сайт]. — 2026. — URL: https://arxiv.org/abs/2601.04653 (дата обращения: 08.01.2026).
5. Top 10 IaC Scanning Tools to Consider in 2026: — Текст электронный // sentinelone.com: [сайт]. — 2026. — URL: https://www.sentinelone.com/cybersecurity-101/cloud-security/iac-scanning-tools/ (дата обращения: 12.01.2026).