

## **СПОСОБЫ БЕЗОПАСНОГО ВЫПОЛНЕНИЯ ПОЛЬЗОВАТЕЛЬСКОГО КОДА**

В некоторых сервисах необходима возможность запуска пользовательского кода, чаще всего это применяется в сервисах для тренинга программированию или других образовательных ресурсах. Выполнение пользовательского кода непосредственно на хост-системе, без использования дополнительных инструментов, представляет собой значительный риск для безопасности и целостности системы, включая: уязвимости безопасности, конфликты с другими ресурсами системы и риск потери данных.

Контейнеризация обеспечивает изоляцию и безопасность при выполнении пользовательского кода путем создания отдельной среды выполнения внутри контейнера. Это позволяет избежать воздействия кода на хост-систему и другие контейнеры, обеспечивая высокий уровень безопасности и стабильности выполнения приложений. Преимущества контейнеризации включают легковесность, легкость масштабирования и управления зависимостями приложений.

Виртуализация предоставляет возможность выполнения пользовательского кода в изолированной виртуальной среде, независимой от хост-системы. Это обеспечивает полную изоляцию кода и гарантирует его безопасность и целостность. Преимущества виртуализации включают эффективное использование ресурсов хост-системы, управление сетью и безопасностью. Использование сторонних API, таких как Godbolt, позволяет безопасно выполнить пользовательский код в удаленной среде. Эти сервисы предоставляют изолированные среды выполнения, где код может быть протестирован и проанализирован без риска для хост-системы. Преимущества использования сторонних API включают доступность и предельный уровень безопасности.

Выбор конкретного метода зависит от требований к безопасности, производительности и управляемости приложения. Однако, вне зависимости от выбранного метода, обеспечение безопасности при выполнении пользовательского кода должно быть приоритетной задачей для разработчиков и администраторов систем. В рамках дипломного проектирования был выбран метод использования сторонних API, так как данный метод облегчает расширяемость. На каждый из компиляторов желательно создавать от двух контейнеров или виртуальных машин, поэтому при использовании большого количества компиляторов, приложение станет очень требовательным.