AMIX

Для кого цей посібник:

- Власники стартапів що зростають
- Солофаундери
- Продакт овнери

Від MVP до зрілої інфраструктури

для продуктових компаній

12 структурованих частин · Практичний підхід від AMIX

Зміст

AMIX

1 Інфраструктура як Код (IaC)

Перенесення інфраструктури в код — це серйозне оновлення для бізнесу.

- Жодних ручних налаштувань серверів
- Автоматичні деплойменти в один клік
- Версіонування та повний аудит змін
- Швидке відновлення після збоїв

AMIX

Що таке Infrastructure as Code?

IaC — це метод, за якого вся інфраструктура (сервери, бази даних, мережа, правила безпеки) описується у вигляді файлів коду. Замість ручного налаштування через панелі управління — декларативний код, який можна версіонувати, тестувати і повторно використовувати.

Як це працює:

AMIX

Технічні переваги IaC

1 Автоматичний деплой

Весь стек інфраструктури запускається однією командою — без ручних дій в консолі.

2 Контроль версій

Вся інфра зберігається в Git: повний аудит змін та миттєвий відкат до будь-якої версії.

3 Консистентні середовища

Dev, staging та production використовують однаковий код — 'у мене працювало' залишилось в минулому.

4 Швидке відновлення

Вся інфраструктура відновлюється за хвилини замість годин ручної роботи.

AMIX

Бізнес-переваги

Швидший вихід на ринок

Dev/test середовища за хвилини, функції виходять швидше.

Нижчі витрати

Автоскейлінг під пікові навантаження, усунення простоюючих ресурсів.

Гнучке масштабування

Миттєва адаптація до зростання трафіку або нових ринків.

Чеклист готовності

- ✓ Вся інфраструктура описана в коді та збережена в Git
- ✓ Деплойменти успішно тестуються в sandbox
- ✓ Процедура відкату перевірена
- ✓ Жодних секретів у конфіг-файлах

Ключові висновки

- IaC = швидкість + надійність + економія
- Версіонування → нуль розходжень між середовищами
- Автоскейлінг → оптимальне використання ресурсів

AMIX

2 Dev, Staging, Prod: Золоте тріо якості

Запускати все в одному середовищі — ризик. Три окремих середовища = швидше, безпечніше, без сюрпризів.

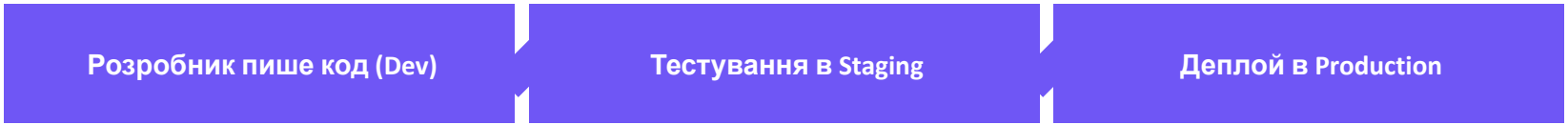
- Жодного 'у мене все працювало'
- Помилки виявляються до продакшну
- Впевнені та контрольовані деплойменти
- Менше збоїв = більше довіри користувачів

AMIX

Що таке IT-середовище?

IT-середовище — це повний набір компонентів (сервери, бази даних, мережа, конфіг), необхідний для роботи застосунку. Три окремих середовища утворюють систему фільтрів якості: кожне наступне ближче до реального стану системи.

Як це працює:



AMIX

Три середовища: принципи роботи

1 Development

Ваш полігон для експериментів. Розробляйте, ламайте, тестуйте — без впливу на реальних користувачів.

2 Staging

Дзеркало продакшну: реальні дані, повний стек, справжні тести. Тут ловляться всі баги до виходу.

3 Production

Тільки перевірений та затверджений код. Моніторинг увімкнено постійно.

4 Зниження ризиків

Кожне середовище — це фільтр якості: менше помилок, менше відкатів.

AMIX

Бізнес-переваги

Швидша розробка

Розробники рухаються швидше, маючи страховочну мережу.

Менше витрат на виправлення

Помилки на ранніх стадіях значно дешевші у виправленні.

Більша довіра користувачів

Стабільні релізи = задоволені та лояльні клієнти.

Чеклист готовності

- ✓ Всі середовища повністю ізольовані та автоматизовані
- ✓ Staging відображає Production для точного тестування
- ✓ Плани відкату перевірені та протестовані
- ✓ Production моніториться постійно

Ключові висновки

- Три середовища = страховочна мережа від збоїв у Production
- Staging є обов'язковим — це не розкіш, а необхідність
- Розробники працюють швидше без страху зламати продакшн

AMIX

3

Public, Private та VPC: Вибір хмарної стратегії

Вибір хмарної моделі впливає на безпеку, масштабованість та витрати. Яка підходить саме вам?

- Public Cloud: дешево, але спільна інфраструктура
- Private Cloud: безпечно, але дорого
- VPC: найкраще з обох варіантів

AMIX

Що таке Virtual Private Cloud (VPC)?

VPC — це ваш приватний мережевий простір всередині публічного хмарного провайдера (AWS, Azure, GCP). Ви отримуєте ефективність та гнучкість публічної хмари з рівнем ізоляції та контролю приватної мережі. Ваші додатки та дані захищені, але ви не платите за фізичне обладнання.

Як це працює:

AMIX

VPC: Технічна архітектура

1 Приватні IP-діапазони

Ваш адресний простір ізольований — ніхто інший не має до нього доступу.

2 Розділення підмереж

Публічні та приватні сервіси розділені на рівні проектування.

3 Правила Firewall

Кастомний контроль доступу для кожного сервісу та з'єднання.

4 Identity-Based Access

Користувачі бачать тільки те, що потрібно — жодного випадкового розкриття даних.

AMIX

Бізнес-переваги

Сильніший захист даних

Дані ізольовані від інших користувачів хмари.

Нижча вартість інфраструктури

Жодного фізичного обладнання — платите лише за використані ресурси.

Швидке масштабування

Миттєве виділення ресурсів без затримок і вузьких місць.

Чеклист готовності

- ✓ Кордони VPC чітко визначені (IP, підмережі, Firewall)
- ✓ Політики Identity Access налаштовані та протестовані
- ✓ Моніторинг та алертинг увімкнено всередині VPC
- ✓ Налаштовано автоскейлінг та міжрегіональне зростання

Ключові висновки

- VPC поєднує переваги публічної та приватної хмари
- Контроль над мережею без придбання обладнання
- Стабільна продуктивність навіть під час пікового навантаження

AMIX

4

Kubernetes: Основа сучасної інфраструктури

Контейнери дали нам гнучкість. Kubernetes — контроль над ними.

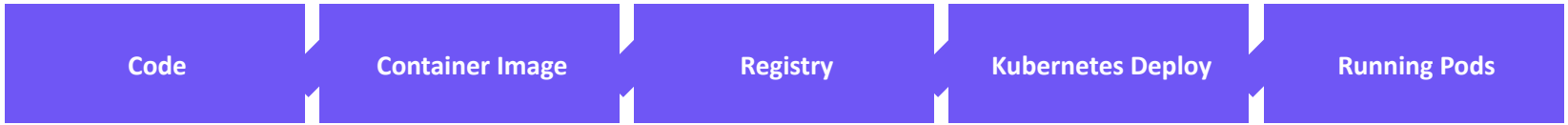
- Оркестрація сотень контейнерів автоматично
- Самовідновлення при збоях вузлів і pods
- Blue/green деплойменти та canary релізи
- Підтримка multi-cloud та on-prem сценаріїв

AMIX

Що таке Kubernetes?

Kubernetes — open-source платформа для оркестрації контейнеризованих застосунків. Автоматизує деплоймент, масштабування, мережеву взаємодію та доступність. Спочатку створений Google, тепер підтримується CNCF та провідними хмарними провайдерами: AWS, Azure, GCP.

Як це працює:



AMIX

Архітектура Kubernetes

1 Control Plane

API Server, Scheduler, Controller Manager, etcd — мозковий центр кластера, що управляє всім станом.

2 Worker Nodes

Kubelet, kube-proxy, Container Runtime — фізичні або віртуальні вузли, де виконуються Pod'и.

3 Pods та Deployments

Pod — найменша одиниця деплою. Deployment визначає бажаний стан та кількість реплік.

4 Services та Helm

Services — мережева адресація Pod'ів. Helm — пакетний менеджер для K8s застосунків.

AMIX

Ключові компоненти та виклики Kubernetes

Ключові компоненти

- Namespaces → Віртуальні кластери всередині кластера
- ReplicaSets → Підтримання кількості реплік Pod'ів
- StatefulSets → Pods з унікальним ідентифікатором та станом
- DaemonSets → Запуск агента на кожному вузлі
- Volumes → Постійне спільне сховище
- Helm → Пакетний менеджер (як apt/yum для K8s)

Виклики впровадження

Крива навчання
DevOps + контейнери + інфра = значне ускладнення стеку

Навички команди
Ops, dev та архітектура мають злитись воєдино

Операційна складність
Реальні збої бувають, і вони болючі без підготовки

Культурне вирівнювання
Бізнес та інженерія мають бути на одній стороні

AMIX

Бізнес-переваги

Прискорена розробка

Автоматизований деплоймент скорочує цикл виходу фічей.

Cloud-agnostic

Запускайте де завгодно: AWS, GCP, Azure або on-prem.

Ефективне масштабування

Автоматичне масштабування під навантаження без ручного втручання.

Чеклист готовності

- ✓ Контейнери запущені як Pods з правильними специфікаціями
- ✓ Helm використовується для управління деплойментами
- ✓ Логування та моніторинг інтегровані
- ✓ Namespaces використовуються для команд/проектів
- ✓ Rollbacks та scaling policies протестовані

Ключові висновки

- K8s робить контейнери масштабованими та production-ready
- Helm та Istio розширюють можливості платформи
- Kubernetes — стандарт cloud-native інфраструктури

AMIX

5 Надійна система баз даних у хмарі

Коли падає код — перезапускаєш. Коли падають дані — це game over.

- Managed Cloud БД: вбудована відмовостійкість
- Автоматичні резервні копії та point-in-time відновлення
- Ізоляція доступу через VPC та IAM
- Моніторинг продуктивності в реальному часі

AMIX

Managed vs Self-Hosted бази даних

Cloud-база даних — це база, розміщена та керована хмарною інфраструктурою (AWS, Azure). Managed-сервіси (RDS, Aurora) забезпечують автоматичне резервне копіювання, масштабування та multi-zone failover без необхідності управляти залізом. На відміну від self-hosted, вони пропонують швидше відновлення та спрощений скейлінг.

Як це працює:

AMIX

Архітектура надійної хмарної БД

1 Managed Cloud Services

Висока доступність, автоскейлінг та вбудовані снепшоти — AWS RDS, Aurora.

2 Стратегія резервних копій

Щоденні інкрементальні бекапи о 4:00, тижнева ротація, місячні снепшоти. Point-in-time відновлення до хвилини.

3 Identity-Based Access

Жодного публічного доступу — тільки внесені до whitelist IP та bastion-хости.

4 Розширений моніторинг

CPU, пам'ять, повільні запити — все в одному дашборді (Grafana/CloudWatch).

AMIX

Бізнес-переваги

Zero-downtime відновлення

Failover-ready архітектура за дизайном.

Менші витрати на обслуговування

Не потрібен штатний DBA для рутинних задач.

Ефективне масштабування

Масштабування під реальний трафік в реальному часі.

Ключові висновки

- БД — фундамент застосунку, не піклуватись про неї ризиковано
- Managed-сервіси дають вбудовану стійкість — використовуйте їх
- Бекапи та моніторинг — не опція, а інструменти керування ризиками

Чеклист готовності

- ✓ Використовуються managed DB (RDS, Aurora тощо)
- ✓ Multi-zone реплікація увімкнена
- ✓ Доступ тільки через VPC — жодного публічного
- ✓ Бекапи заплановані та протестовані
- ✓ Аналіз повільних запитів налаштований
- ✓ Grafana або аналог підключений

AMIX

6

DevOps Моніторинг: Основи надійних систем

Ваш pipeline може бути швидким. Інфраструктура — сучасною. Але без моніторингу ви летите наосліп.

- Раннє виявлення проблем до впливу на користувачів
- Метрики Golden Signals та DORA
- Real-time дашборди для різних ролей команди
- Автоматична ескалація алертів

AMIX

Що таке DevOps Моніторинг?

DevOps моніторинг — це безперервне відстеження здоров'я, продуктивності та поведінки інфраструктури, застосунків та сервісів. Це не просто алерти — це раннє виявлення, проактивне запобігання та постійне покращення системи.

Як це працює:

AMIX

Ключові метрики та інструменти

1 Golden Signals (Google SRE)

Latency (час відповіді), Traffic (навантаження), Errors (помилки), Saturation (наближення до ліміту).

2 DORA Метрики

Deployment Frequency, Lead Time for Changes, Change Failure Rate, Time to Restore Service.

3 Prometheus + Grafana

Збір метрик, інтерактивні дашборди, алертинг через Alertmanager. Де-факто стандарт.

4 Loki + CloudWatch

Агрегація логів (Loki) та нативний AWS-моніторинг (CloudWatch) для повної observability.

AMIX

Дашборди за ролями та стек моніторингу

Дашборди за ролями

Dev Помилки деплойменту, латентність API, логи сервісів, stack traces

Ops CPU/пам'ять, мережа, failover-статус, alert-черги

Biz Uptime SLA, MTTR, кількість інцидентів, cost per service

Рекомендований стек моніторингу

Prometheus
Open-source збір метрик та алертинг

Grafana
Інтерактивні дашборди для всіх джерел

Alertmanager
Роутинг та ескалація нотифікацій

Loki
Масштабована агрегація логів

CloudWatch
Нативний моніторинг AWS-ресурсів

AMIX

Бізнес-переваги

Швидше відновлення (MTTR ↓)

Real-time алерти + дані про першопричину = швидші виправлення.

Менший downtime

Помилки виявляються до того, як вони впливають на користувачів.

Оптимізація витрат

Знаєте, що масштабувати, а що скорочувати.

Чеклист готовності

- ✓ Core signals відстежуються: latency, errors, saturation
- ✓ Real-time моніторинг + historical trends доступні
- ✓ Role-based дашборди є (Dev, Ops, Biz)
- ✓ Prometheus / Grafana / CloudWatch інтегровані
- ✓ Алерти запускають автоматичну ескалацію

Ключові висновки

- Моніторинг — це не опція, це ваш зворотний зв'язок
- Комбінуйте логи, метрики та трейси в одному місці
- Вирівнюйте дашборди з потребами команд та бізнесу

AMIX

7 Контейнеризація: Навіщо це бізнесу?

Контейнеризація стала стандартом сучасного деплойменту — і ось чому.

- Швидкі та консистентні деплойменти між середовищами
- Покращена відмовостійкість через мікросервіси
- Нижчі витрати на інфраструктуру та обслуговування
- Портативність: збери один раз — запусти будь-де

AMIX

Що таке контейнеризація?

Контейнеризація пакує застосунок та всі його залежності в легкий, портативний блок — контейнер. На відміну від VM, контейнери спільно використовують ОС хоста, що робить їх швидшими та ресурсоефективнішими. Розробники збирають застосунок один раз та запускають його будь-де — на будь-якому сервері, в будь-якій хмарі.

Як це працює:

AMIX

Технічні переваги контейнеризації

1 Портативність

Запускайте де завгодно без переналаштування середовища — dev, staging, prod ідентичні.

2 Ізоляція

Кожен контейнер працює незалежно, збій одного сервісу не зачіпає інших.

3 Масштабованість

Легко запускайте або зупиняйте контейнери залежно від навантаження.

4 Ефективність

Швидший старт та менші накладні витрати порівняно з VM.

AMIX

Бізнес-переваги

Швидший вихід на ринок

Деплоймент фічей без проблем 'локально працює'.

Вища надійність

Ізоляція збоїв: один неробочий сервіс не валить весь застосунок.

Нижчі витрати

Максимальне використання серверів, мінімум простоюючих ресурсів.

Чеклист готовності

- ✓ Застосунок та залежності запаковані в образ контейнера
- ✓ Container registry налаштований для зберігання образів
- ✓ Оркестратор (K8s) налаштований для скейлінгу та відновлення
- ✓ Жодних чутливих даних в образах контейнерів

Ключові висновки

- Docker будує контейнери, Kubernetes управляє ними
- Ідеально для cloud migration та мікросервісної архітектури
- Контейнеризація вирішує проблему несумісності середовищ

AMIX

8 Критичні елементи безпеки DevOps

Безпека — не шар поверх системи. Це мислення, вбудоване в кожен крок.

- Firewall: контрольний пункт для всього трафіку
- IAM: правильний доступ для правильних ідентичностей
- Security Groups та Bastion-хости для ізоляції
- Audit-ready політики для compliance-перевірок

AMIX

Firewall та Identity Access Management (IAM)

Firewall перевіряє кожен пакет даних, що намагається ввійти або вийти з мережі: дозволити чи заблокувати на основі правил. IAM — комплексна система контролю: хто ви? Що можете отримати? Що можете робити з доступом? Разом вони формують основу безпеки DevOps.

Як це працює:

AMIX

IAM та мережева безпека: деталі

1 Firewall за замовчуванням закритий

Дозволяйте лише те, що явно необхідно — блокуйте все решта.

2 IAM Groups & Roles

Призначайте дозволи ролям, а не окремим особам. Принцип найменших привілеїв.

3 Service-to-Service Access

Security groups та bastion-хости розділяють рівні: app/db/network.

4 Audit-Ready Policies

Маппінг користувачів, логи доступу та аудит — для безпроблемних compliance-перевірок.

AMIX

IAM: три ключових питання безпеки

1 **Хто ти?**

Authentication: перевірка ідентичності через credentials, MFA, SSO.

2 **Що можеш отримати?**

Authorization: на які ресурси та сервіси має доступ ця ідентичність.

3 **Що можеш робити?**

Permissions: читання, запис, виконання — мінімально необхідний рівень.

Правило: кожна ідентичність отримує мінімум необхідних дозволів і нічого більше.

AMIX

Бізнес-переваги

Менша поверхня атаки

Запобігання бічному руху та зовнішнім експлоїтам.

Швидший onboarding/offboarding

Надати або відкликати доступ однією дією.

Спрощений compliance-аудит

Рольовий доступ та логування роблять перевірки безпроблемними.

Чеклист готовності

- ✓ Default-deny Firewall у всіх середовищах
- ✓ Bastion host та ізоляція підмереж налаштовані
- ✓ IAM ролі та групи створені (admin, read-only, service)
- ✓ Логування доступу увімкнено
- ✓ Сторонні інструменти ізольовані через service accounts

Ключові висновки

- Firewall зупиняє несанкціонований трафік до застосунків
- IAM гарантує мінімально необхідний доступ
- Безпека DevOps = кордони, видимість та контроль

AMIX

9 Compliance-стратегії DevOps

DevOps — це швидкість. Compliance — це контроль. Разом? Так, якщо вбудувати compliance в pipeline.

- Автоматичні перевірки в кожному CI/CD commit
- IaC + рольовий доступ для аудит-ready інфри
- Централізоване зберігання логів та доказів
- Стандарти: ISO 27001, GDPR, SOC 2, HIPAA

AMIX

Що таке DevOps Compliance?

DevOps Compliance — це практика узгодження розробки з галузевими регуляторними вимогами (ISO 27001, GDPR, SOC 2). Мета — не уповільнити команди, а вбудувати безпеку, довіру та аудитованість у workflow з першого дня. Автоматизація перевірок в pipeline перетворює compliance з тягаря на природний процес.

Як це працює:

AMIX

Ключові регуляторні стандарти

Стандарт	Фокус	Вплив на DevOps
ISO 27001	Управління інформаційною безпекою	Контроль змін, доступ, логи
GDPR	Захист персональних даних (ЄС)	Consent, retention, deletion workflows
HIPAA	Медичні дані (США)	Шифрування, IAM, breach response
SOC 2 Type 2	Операційна послідовність	Збір доказів по всьому pipeline

***Інструменти:** Terraform · OPA/Conftest · AWS IAM · CloudWatch · Snyk · Bridgecrew*

AMIX

Бізнес-переваги

Швидкі аудити

Ви завжди готові — жодного панічного збирання доказів.

Довіра як перевага

Compliance зміцнює довіру клієнтів та enterprise-партнерів.

Security by Design

Менше реактивних виправлень, більше превентивної структури.

Чеклист готовності

- ✓ IaC використовується для інфри та дозволів
- ✓ CI/CD включає security та compliance scanners
- ✓ Логи та audit-дані збираються централізовано
- ✓ Рольовий доступ з MFA enforced
- ✓ Політики конфіденційності закодовані

Ключові висновки

- Compliance ≠ блокер, якщо інтегрований рано та автоматизований
- IaC та policy-as-code — ваша compliance-основа
- Рольовий доступ + audit logging = миттєва готовність

AMIX

10

Незламна система управління секретами

Ваша система настільки захищена, наскільки захищений ваш найслабший секрет.

- Централізоване, зашифроване сховище
- Автоматична ротація без downtime
- Жодних credentials в Git або конфіг-файлах
- Ізоляція секретів по сервісах та середовищах

AMIX

Що таке управління секретами?

Secret Management — процес безпечного зберігання, розповсюдження, ротації та аудиту credentials: API-ключів, паролів, токенів, сертифікатів. Головне питання: хто повинен мати доступ до яких секретів — і як тримати всіх інших подалі? Відповідь: централізований vault з рольовим доступом.

Як це працює:

AMIX

Технічна реалізація Secret Management

1 Жодних hardcoded секретів

Credentials ніколи не торкаються Git-репозиторію чи кодової бази.

2 Авто-ротація та версіонування

Секрети оновлюються без downtime та конфліктів.

3 Ізоляція по середовищах

Секрети ін'єктуються тільки в ті сервіси, яким вони потрібні.

4 External Secrets Operator

Синхронізує секрети з AWS Secrets Manager → Kubernetes автоматично та безпечно.

AMIX

Типові помилки та правильні рішення

❌ Не робіть так	✅ Робіть натомість
Hardcode секретів у source code	Ін'єкція через env vars під час виконання
Config-файли з credentials в Git	Vault-backed secrets + External Secrets Operator
Ручний обмін паролями між командою	IAM та scoped access per team role
Відкладена ротація секретів	Автоматичне версіонування + auto-reload застосунків

AMIX

Бізнес-переваги

Менше векторів злому

Секрети ніколи не зберігаються локально, в Git чи файлах.

Швидший DevOps-процес

Команди більше не діляться credentials вручну.

Zero-downtime ротація

Секрети оновлюються автоматично, застосунки перезапускаються миттєво.

Чеклист готовності

- ✓ Всі секрети в зашифрованому vault (AWS Secrets Manager)
- ✓ Жодних hardcoded credentials у коді або конфігу
- ✓ Ін'єкція через environment variables
- ✓ Operator sync + reload-механізм налаштований
- ✓ Доступ до секретів рольовий та логований

Ключові висновки

- Секрети мають залишатись секретами — не в Git
- Vault + оператор для управління, синхронізації та ротації
- Правильний secret management = кращий захист + плавніша робота

AMIX

11

ElastiCache: Прискорення продуктивності застосунку

Повільні застосунки вбивають retention. Кешування = швидкість.

- In-memory відповіді за мікросекунди замість мілісекунд
- Розвантаження БД від повторюваних ідентичних запитів
- Автоматичне масштабування шару кешування
- Zero maintenance — Amazon управляє patching та failover

AMIX

Що таке Amazon ElastiCache?

Amazon ElastiCache — managed in-memory data store, що забезпечує блискавичне отримання часто запитуваних даних. Він діє як шар пам'яті між вашим застосунком та базою даних, зберігаючи повторювані запити, списки продуктів, профілі користувачів та інші популярні дані.

Як це працює:

AMIX

Як працює кешування:

1 In-memory швидкість

Відповіді за мікросекунди (vs мілісекунди для БД) — різниця відчутна одразу.

2 Автоскейлінг

Шар кешу росте або зменшується залежно від трафіку автоматично.

3 Розвантаження БД

Звільняє базу від тисяч однакових запитів — більше ресурсів для важливих операцій.

4 Zero Maintenance

Amazon обробляє patching, failover та реплікацію за вас.

AMIX

Що кешувати, а що — ні

✅ Кешувати варто	❌ Не варто кешувати
✔ Списки продуктів	✘ Ціни акцій у реальному часі
✔ Профілі користувачів	✘ Прогнози погоди
✔ FAQ та статичні блоки	✘ Дані, що змінюються щосекунди
✔ Популярні сторінки з низькою частотою змін	✘ Рідко запитуваний контент

AMIX

Бізнес-переваги

Швидші відповіді

Кешовані дані = миттєва доставка контенту.

Краще масштабування

Обробляйте пікове навантаження без bottleneck у БД.

Зниження витрат

Менші інстанси БД завдяки кешуванню read-трафіку.

Ключові висновки

- ElastiCache прискорює застосунки без переписування коду
- Ідеально для часто-використовуваних, рідко-оновлюваних даних
- Знижує навантаження на БД, покращує UX, зменшує витрати

Чеклист готовності

- ✓ Шар кешу перед БД налаштований
- ✓ High-read, low-change дані кешуються
- ✓ ElastiCache autoscaling увімкнено
- ✓ CloudWatch моніторинг налаштований
- ✓ Стратегія failover протестована

AMIX

12

Яку документацію має надавати DevOps?

Коли DevOps іде — залишається документація. Це ваш операційний страховий поліс.

- Архітектурні схеми та опис системи
- Операційні гайди для кожного сервісу
- CI/CD pipeline та monitoring документація
- Плани відновлення та disaster recovery

AMIX

Що таке DevOps-документація?

DevOps-документація — повний гайд по вашій інфраструктурі. Пояснює, як розуміти, обслуговувати та виправляти технічні системи. Охоплює процеси деплойменту, налаштування моніторингу та процедури відновлення. Мета: зробити передачу знань плавною та уникнути ситуацій, коли тільки одна людина знає всі нюанси системи.

Як це працює:

AMIX

Обов'язкові типи DevOps-документів

System Design Docs

Схема інфраструктури, архітектура хмари, середовища, зони безпеки

Service Ops Docs

Гайди доступу, інструкції перезапуску, підключення до кластера

CI/CD Pipeline Docs

Тригери збірки, flow деплойменту, кроки підтвердження

Monitoring Docs

Дашборди, налаштування алертів, доступ до логів

Recovery Docs

Плани downtime, процедури бекапу, дерево контактів

GitFlow Strategy

Branching, PR-процес, release flow — задокументований та погоджений

Ключовий принцип: якщо документ важко знайти або оновити — ним не будуть користуватись.

AMIX

Бізнес-переваги

Швидше вирішення проблем

Немає здогадок. Коли системи ламаються — є playbook, а не Slack-треди.

Плавніший onboarding

Нові інженери не починають з нуля. Доки = ярлики до продуктивності.

Послідовність між командами

Той самий процес деплойменту. Ті самі кроки відновлення. Без сюрпризів.

Чеклист готовності

- ✓ Знаємо, як перезапустити кожен сервіс?
- ✓ Алертинг-контакти задокументовані?
- ✓ Новий інженер може виправити Failed Pod самостійно?
- ✓ CI/CD поведінка повністю прозора?
- ✓ Є DR-план у цифровому та друкованому вигляді?

Ключові висновки

- DevOps без документації = втрата знань
- Доки дають команді незалежність після хендовера
- Хороша документація скорочує onboarding та час відновлення

AMIX

Дякуємо за увагу!

Разом ми дослідили ключові компоненти сучасної, надійної інфраструктури.
AMIX — це ваш партнер у побудові стабільного DevOps-фундаменту.

Потрібна консультація або впровадження?

Зв'яжіться з командою AMIX