

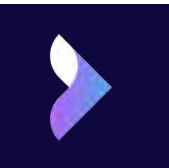
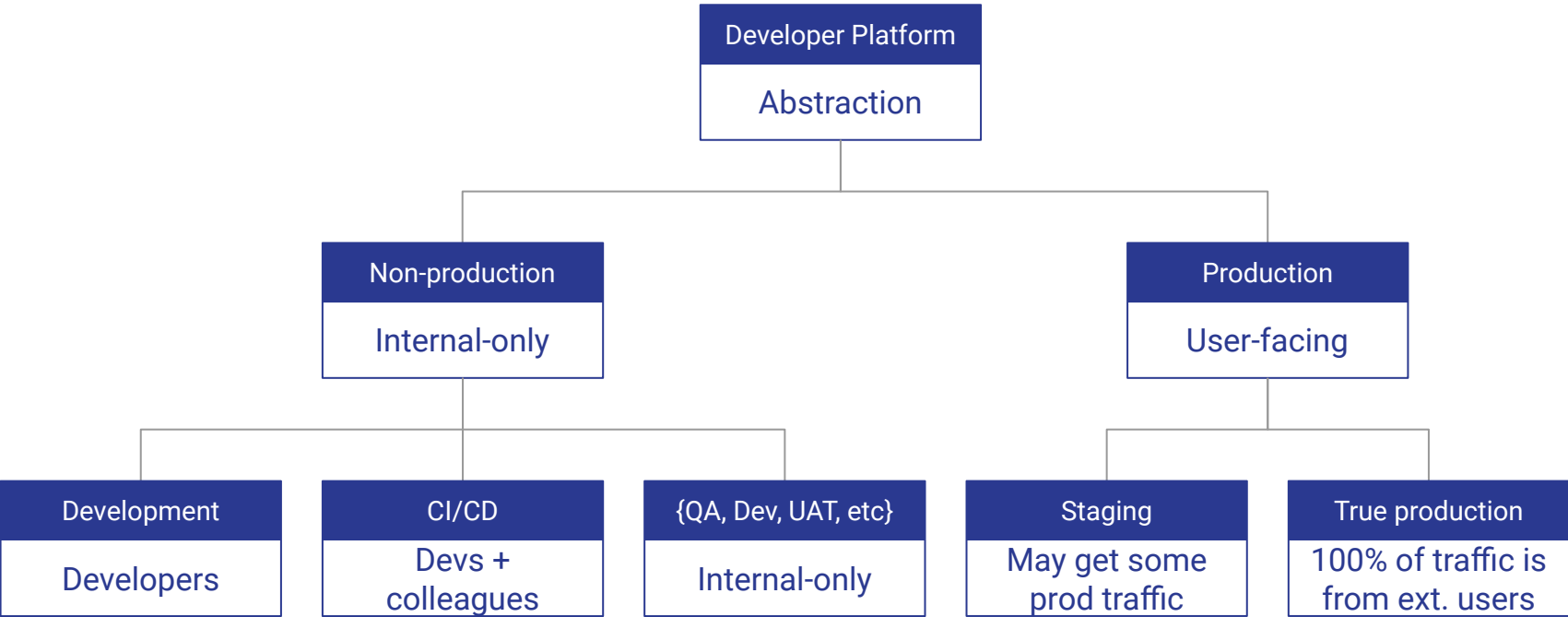
Declarative everything: a GitOps/automation-based approach to building efficient developer platforms

Conf42 DevOps 2024

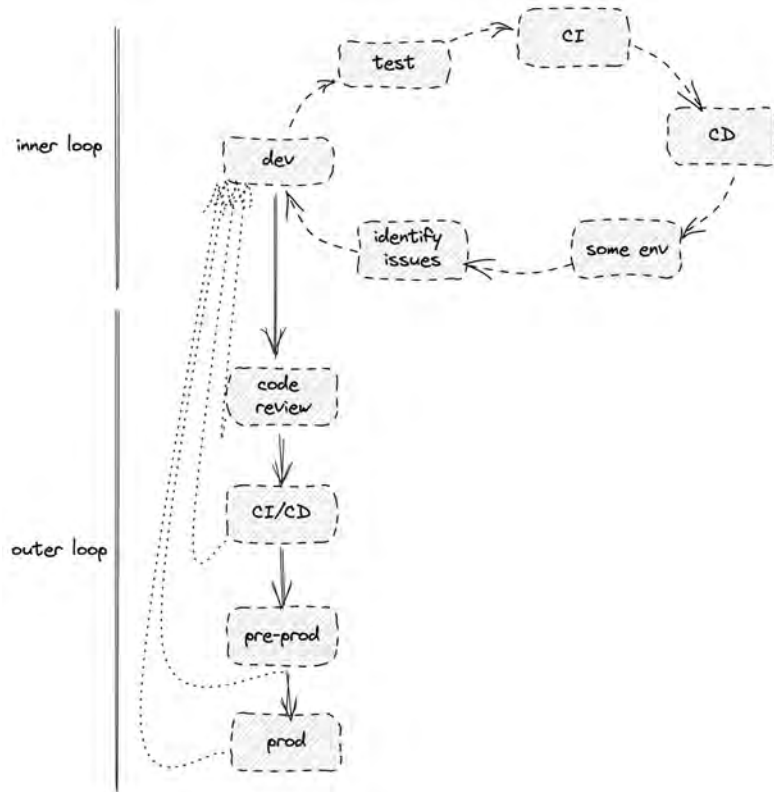
Debo Ray



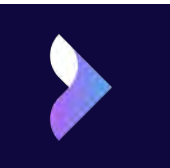
Developer platform coverage

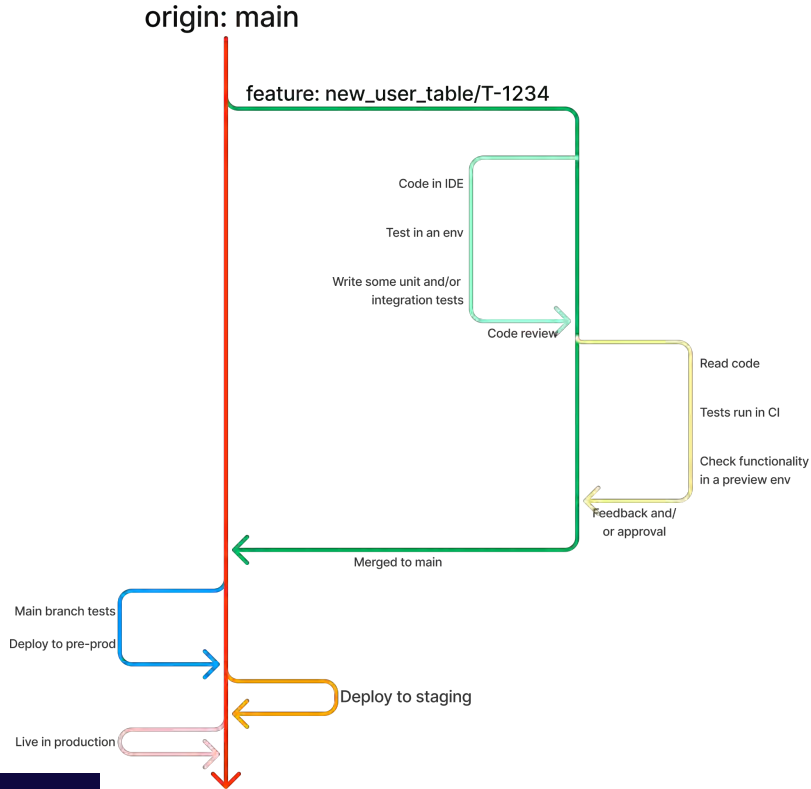


Understanding Developer Workflows



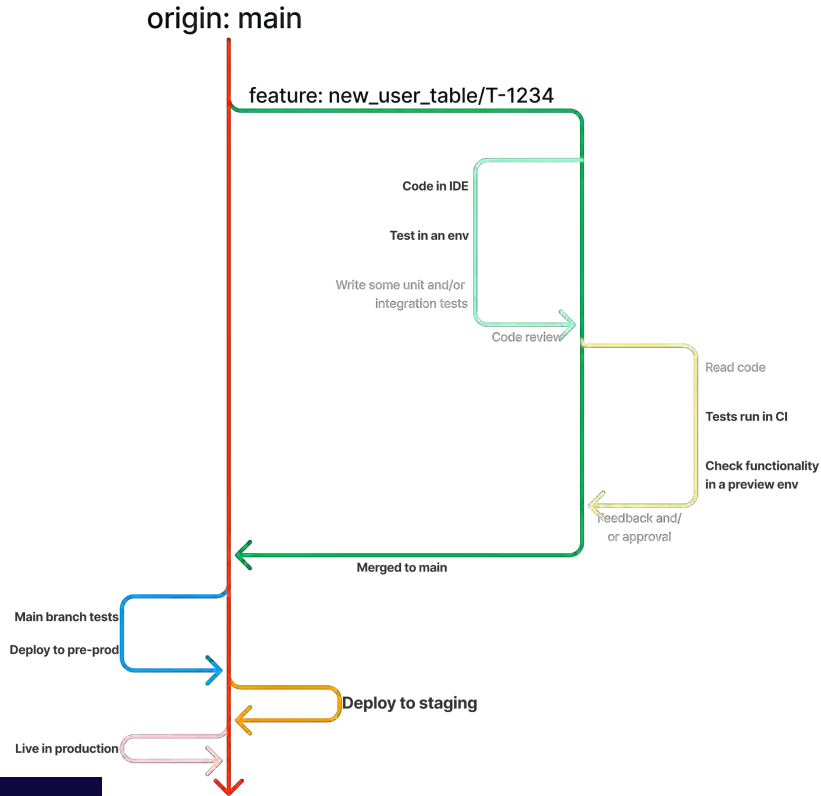
Source: Red Hat





How changes get to production





Where can we benefit from having automation?



GitOps

What: DevOps best practices applied to infrastructure automation (“*golden paths*”)

Focus: Automation/standardization improves the dev experience and helps reduce cost

How: Git-based access to consistent/symmetric environments

Why: Continuous integrations leads to faster development and deployment



An Environment

What *exactly* is it?

Application runtime

Relevant {dev, test, ..} data

Access to downstream services

Accessible from upstream services

Accessible to a human

Isolated from production (if not prod)

Source code, build tools (+ IDE)

{Shared, dedicated} tenancy





We need to rethink our general
“software verification” strategies



Why does testing exist (emphasis on services, **not** libs/modules)?

Functionality

Acceptance criteria

- Is my feature/module meeting product requirements?
- Am I calling downstream APIs per their contracts?

Interoperability

Integrate w/ existing apps/services/patterns

- Although API contracts are met, will these calls actually {return the data, perform the action} I need?
- Are the {latencies, request/response shapes} acceptable?

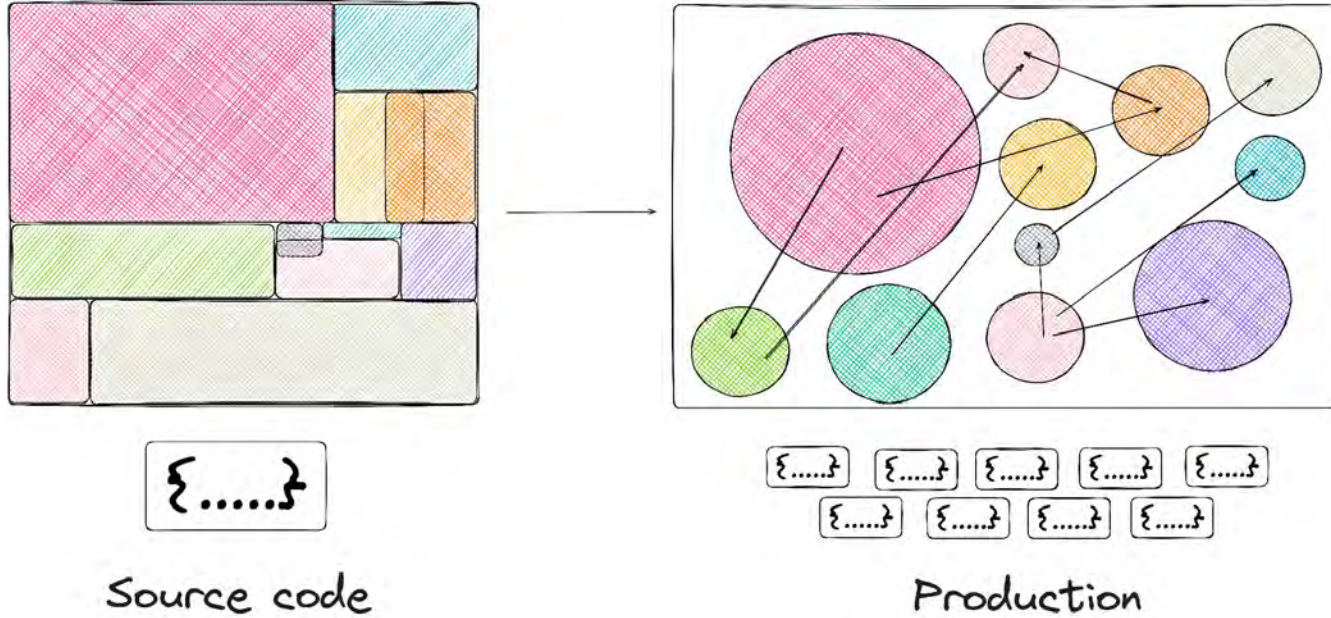
Confidence

Reliability in prod

- Will this make future deployments error-prone?
- Is this reducing idempotence?



Source code ... Running in production



[Production] K8s Deployment YAML, Helm charts etc

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kompose.cmd: kompose convert -f python-todolist-app/docker-compose.yml
    kompose.version: 1.28.0 (c4137012e)
  creationTimestamp: null
  labels:
    io.kompose.service: app
  name: todolist-app
spec:
  ports:
    - name: "6000"
      port: 6000
      targetPort: 6000
  selector:
    io.kompose.service: todolist-app
status:
  loadBalancer: {}
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    io.kompose.service: user-svc
  name: user-svc
spec:
  replicas: 1
  selector:
    matchLabels:
      io.kompose.service: user-svc
  strategy: {}
  templates:
    metadata:
      creationTimestamp: null
      labels:
        io.kompose.network/python-todolist-app-default: "true"
        io.kompose.service: user-svc
    spec:
      containers:
        - image: 749882936473.dkr.ecr.us-east-2.amazonaws.com/python-user-svc:${REV}
          name: user-svc
          ports:
            - containerPort: 8080
          resources: {}
          restartPolicy: Always
status: {}
```

```
## Default values for todolist-app.
## This is a YAML-formatted file.
## Declare variables to be passed into your templates.

## Todolist-app can have only one replica
replicaCount: 1

image:
  repository: "749882936473.dkr.ecr.us-east-2.amazonaws.com/todolist-app"
  pullPolicy: IfNotPresent
  # Overrides the image tag defaults for the pods according
  # to 'latest' or other image tags
  tag: "LATEST-DEPLOY-VERSION"

imagePullSecrets:
  - name: regcred
  nameOverride: ""
  fullImageOverride: ""

serviceAccount:
  # Specifies whether a service account should be created
  # Create: true
  # Annotations to add to the service account
  annotations: {}
  # The name of the service account to use.
  # If not set and create is true, a new one is generated using the following template
  name: "todolist-app-svc-account"

podAnnotations:
  # awslogs: com.amazonaws.com/ecs-logs-agent: ["prometheus"]
  # awslogs: com.amazonaws.com/ecs-logs-agent: [""]
  # awslogs: com.amazonaws.com/ecs-logs-agent: [""]
  # prometheus: url: ["http://prometheus:9090/metrics"]
  # namespace: "k8s-kube-system"
  # metrics: ["*"]
  # send_histogram_buckets: true,
  # send_distribution_buckets: true,
  # send_distribution_counts_at_monotonic: true,
  # max_returned_metrics: 10000,
  # prometheus_timeout: 30

prometheus:
  scrape:
    - port: "8080"
      prometheus: true

podSecurityContext: {}
  # fsGroup: 2000

securityContext: {}
  # capabilities: {}
  # drop: []
  # allowPrivilegeEscalation: true
  # readOnlyRootFilesystem: true
  # runAsGroup: 1000

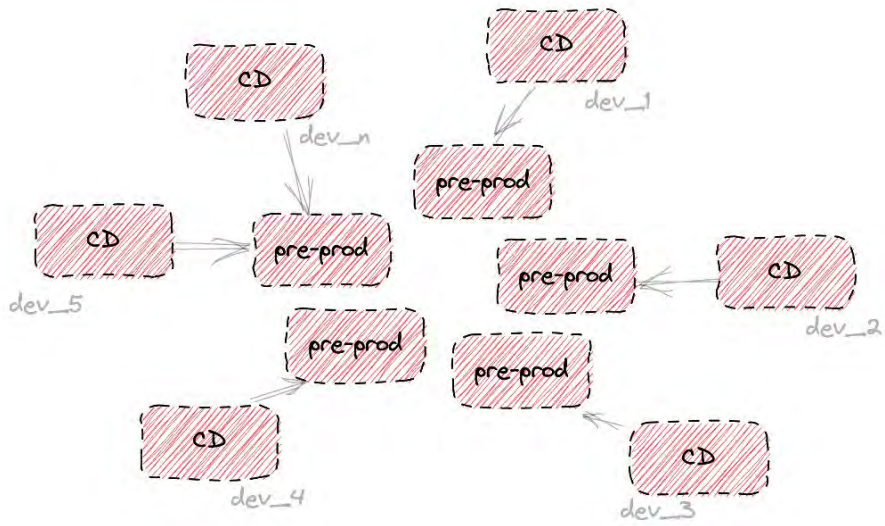
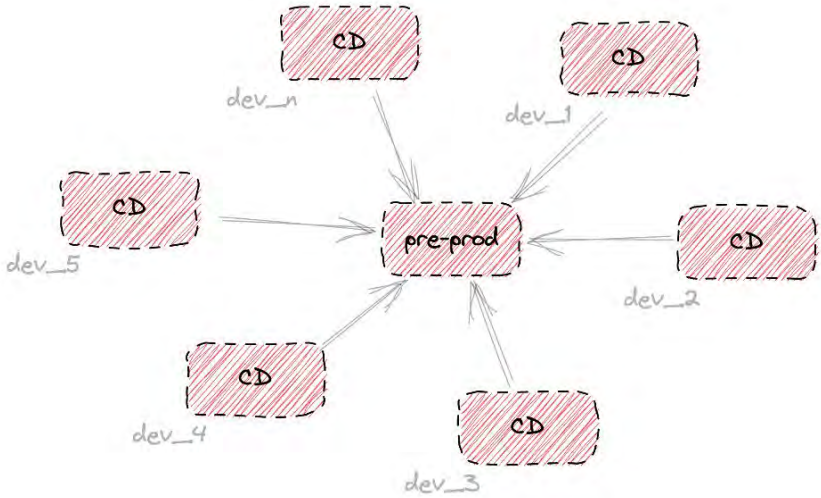
services:
  type: ClusterIP
  port: 8080

resources:
  limits:
    cpu: 80m
    memory: 32M
  requests:
    cpu: 80m
    memory: 50M

nodeSelector: {}
tolerations: {}
affinity: {}
```



Dev -> Code Review -> CI/CD -> Pre-Prod -> Prod



CI might have a flavor of building images, usually not

```
name: Build and Deploy TodoList App
on:
  push:
    branches:
      - main

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v2

      - name: Set up AWS CLI
        uses: aws-actions/configure-aws-credentials@v1
        with:
          aws-access-key-id: ${ secrets.AWS_ACCESS_KEY_ID }
          aws-secret-access-key: ${ secrets.AWS_SECRET_ACCESS_KEY }
          aws-region: us-east-1

      - name: Login to Amazon ECR
        uses: aws-actions/amazon-ecr-login@v1

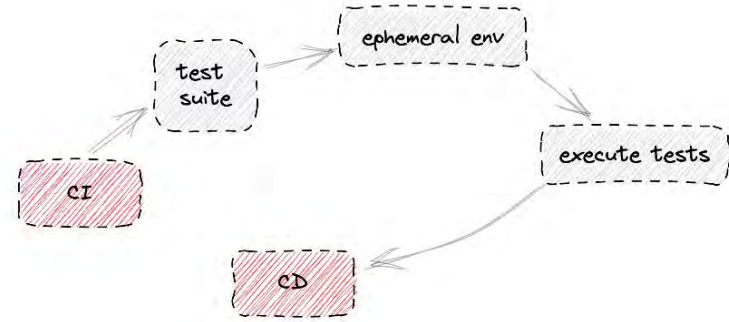
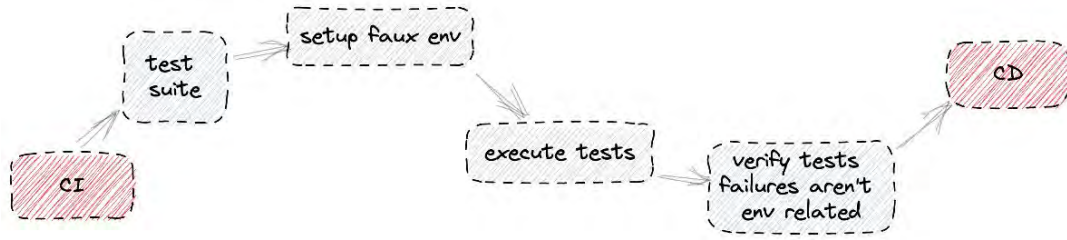
      - name: Build and Push Docker Image
        run: |
          docker build -t your-ecr-repo-uri:latest .
          docker push your-ecr-repo-uri:latest

      - name: Deploy to Kubernetes
        run: |
          aws eks --region us-east-1 update-kubeconfig --name your-eks-cluster

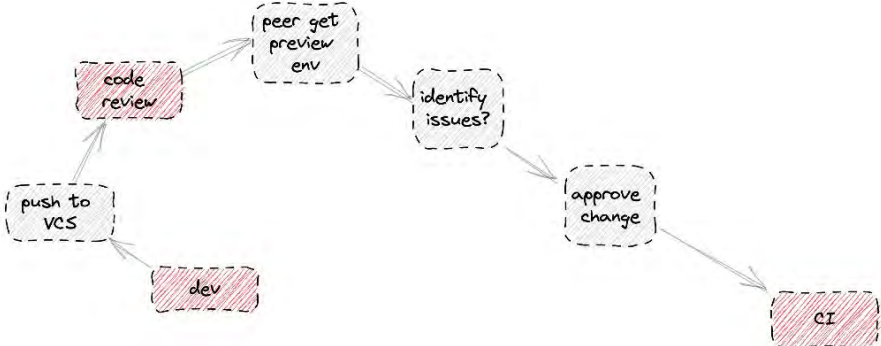
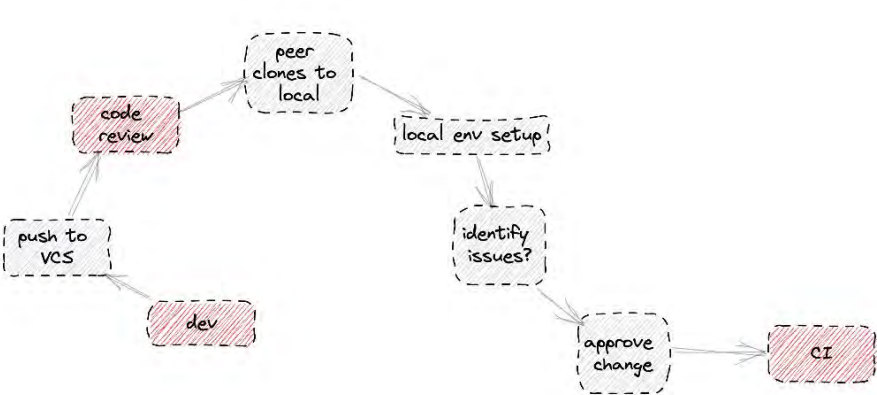
          kubectl apply -f path/to/kubernetes/deployment.yaml
          kubectl apply -f path/to/kubernetes/service.yaml
```



Dev -> Code Review -> CI/CD -> Pre-Prod -> Prod



Dev -> Code Review -> CI/CD -> Pre-Prod -> Prod



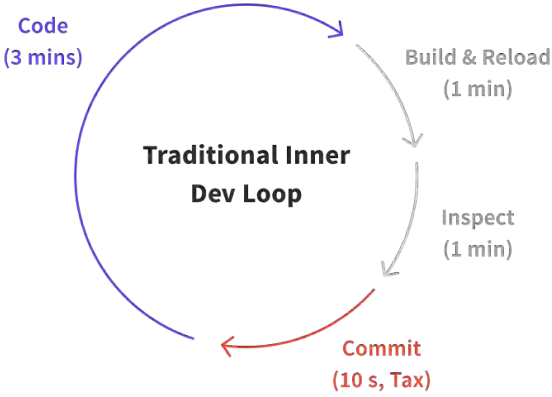
Dev (and CI) usually resorts to docke

```
version: "2"
services:
  todoclist-backend:
    build: ./app
    links:
      - db
      - user-svc
    ports:
      - "6000:6000"
    environment:
      USER_SVC_HOSTPORT: http://user-svc:8000
  db:
    image: mysql:5.7.28
    ports:
      - "32000:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root
    volumes:
      - ./db:/docker-entrypoint-initdb.d:ro
  user-svc:
    build: ../python-user-svc/app
    links:
      - db
    ports:
      - "8000:8000"
  todoclist-web:
    build: ../web-client
    links:
      - user-svc
      - todoclist-backend
    environment:
      USER_SVC_HOSTPORT: http://user-svc:8000
      TODOLIST_APP_SVC_HOSTPORT: http://todolist-app:6000
    ports:
      - "8088:8080"
```

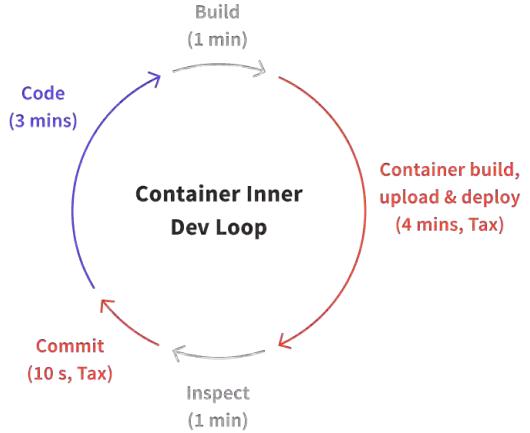
```
version: "2"
services:
  app:
    build: ./app
    links:
      - db
    ports:
      - "8000:8000"
  db:
    image: mysql:5.7.28
    ports:
      - "32000:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root
    volumes:
      - ./db:/docker-entrypoint-initdb.d:ro
```



Deep-dive: Inner-loop of SDLC



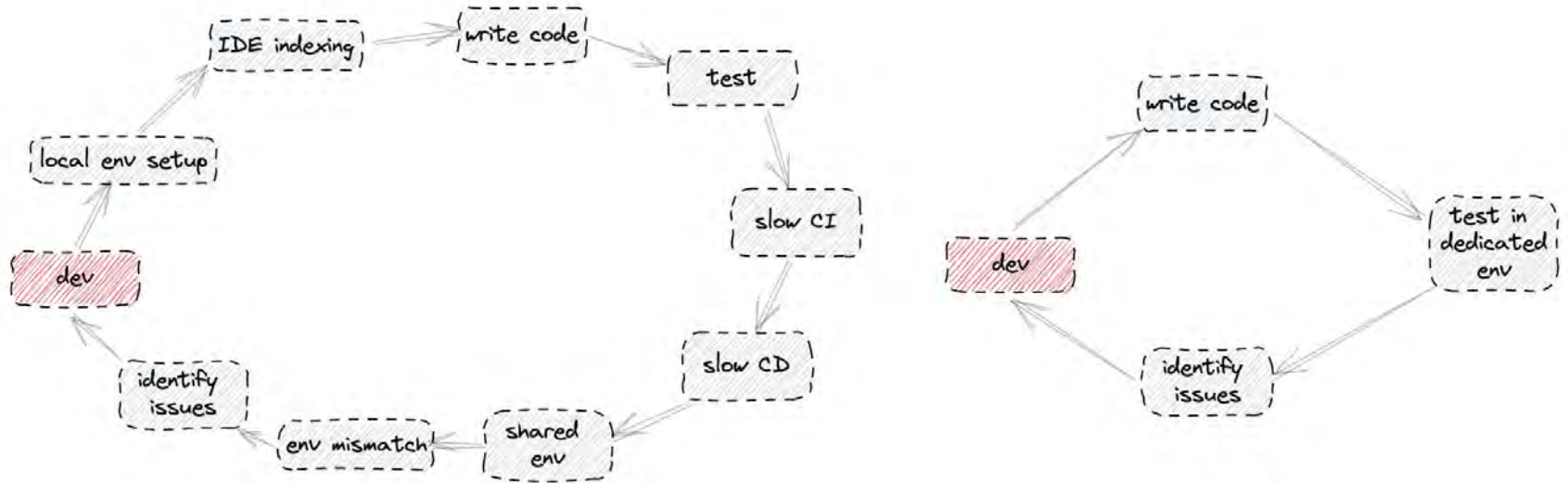
~5 mins



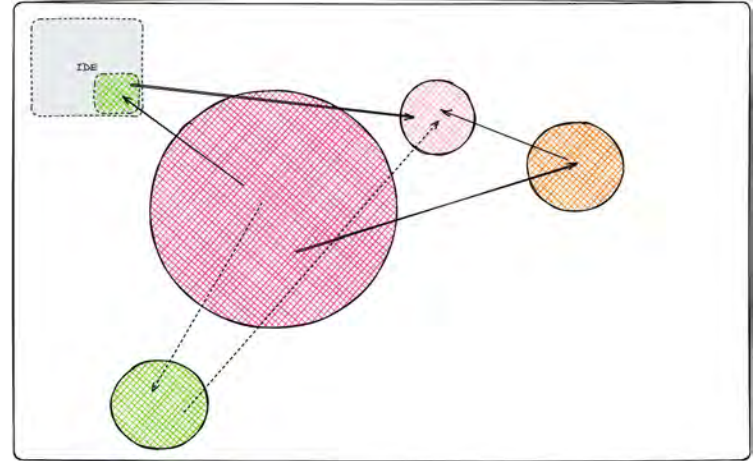
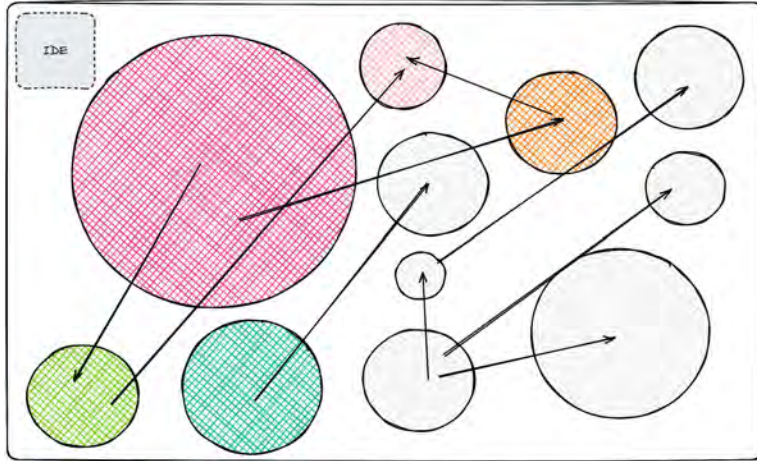
~9 mins
+45%



Dev -> Code Review -> CI/CD -> Pre-Prod -> Prod



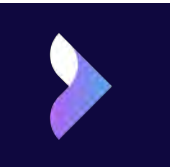
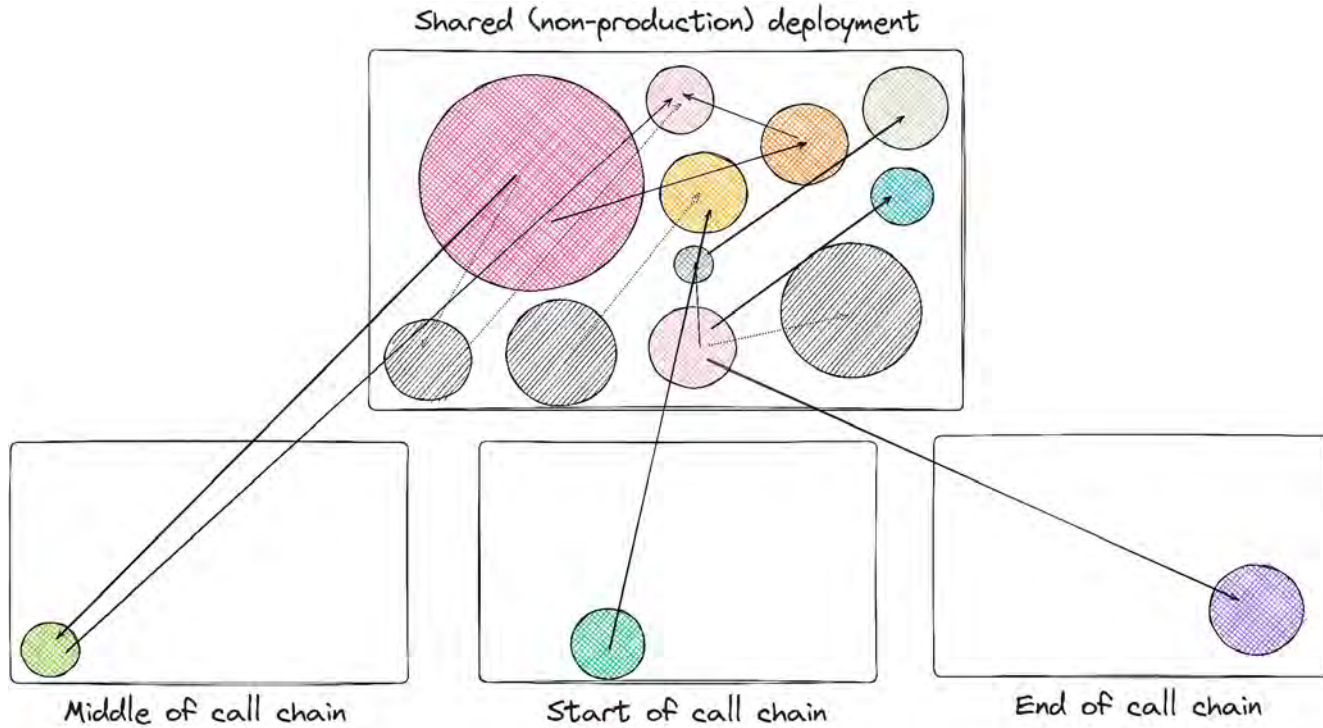
Development purposes (“local dev”, IDE etc)



When service dependencies are somewhat isolated



Testing purposes (“local dev”, CI, etc)



Takeaways

Focus: Issues are always easier to fix when caught before production

How: Easier access to production-symmetric environments

Why: Remove drift between SDLC stages w/ better dev ergonomics



3-up Context Slide

Company

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Context

Ut enim ad minim veniam, quis nostrud exercitation

- Duis aute irure dolor in reprehenderit in voluptate velit

Problem statement

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



3-up Flow Slide

Challenge 1

Expand audience

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Challenge 2

Up 30-day actives

Ut enim ad minim veniam, quis nostrud exercitation

- Duis aute irure dolor in reprehenderit in voluptate velit

Challenge 3

Increase conversion

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Thing

Moar Text Goes Here

Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut
labore et dolore magna aliqua.



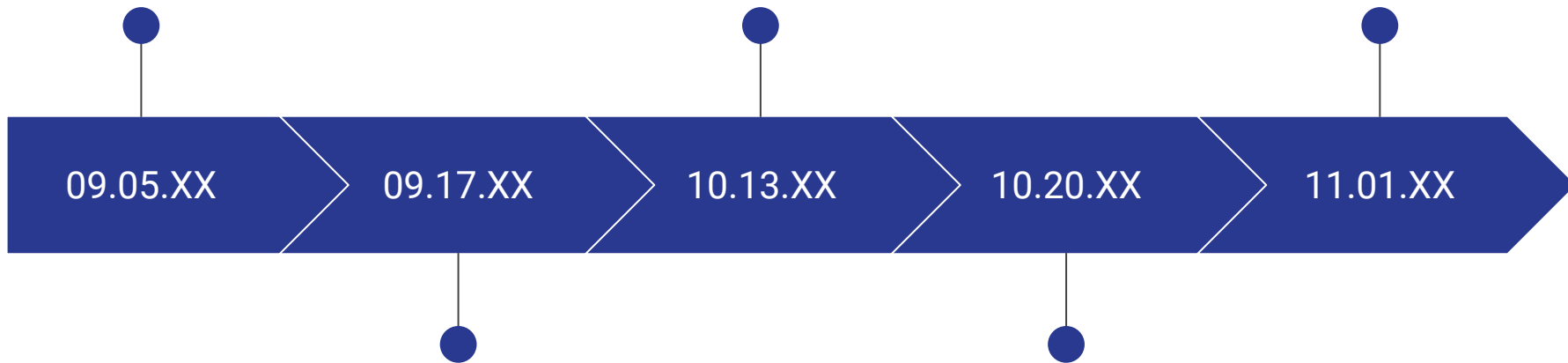
Big Thought/Idea



Lorem ipsum dolor sit
amet, consectetur
adipiscing elit

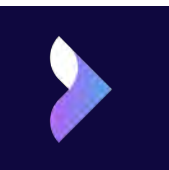
Lorem ipsum dolor sit
amet, consectetur
adipiscing elit

Lorem ipsum dolor sit
amet, consectetur
adipiscing elit

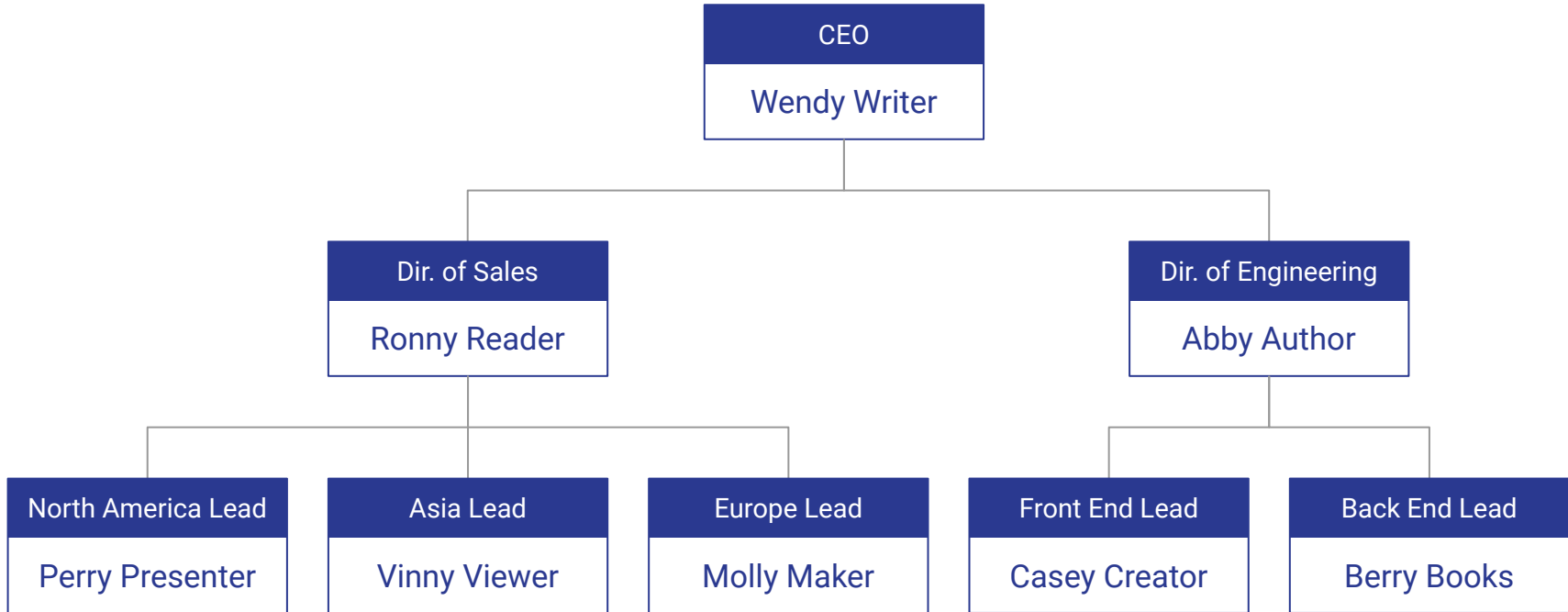


Lorem ipsum dolor sit
amet, consectetur
adipiscing elit

Lorem ipsum dolor sit
amet, consectetur
adipiscing elit



A tree of people



Impact

XX% sales increase

