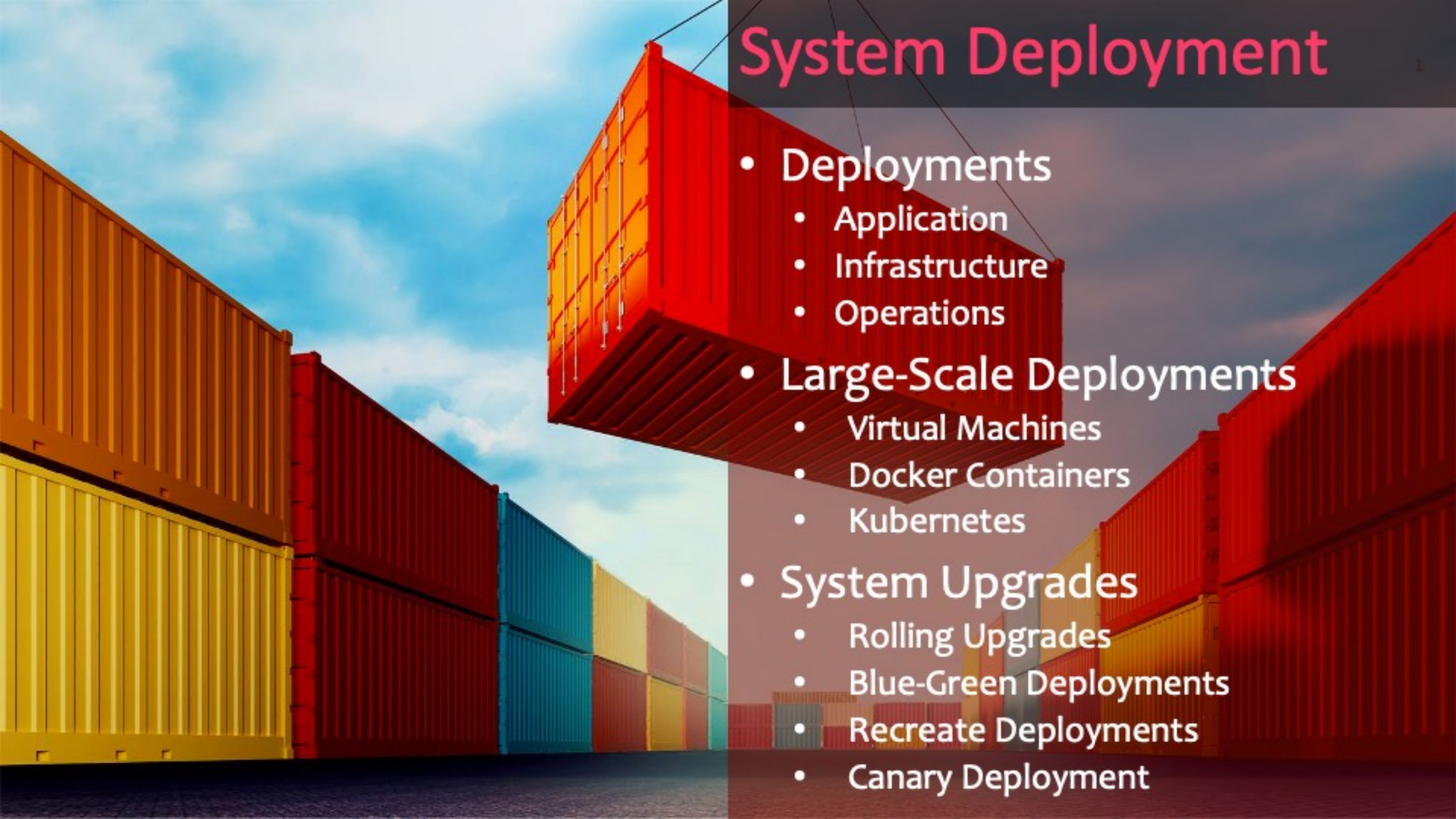
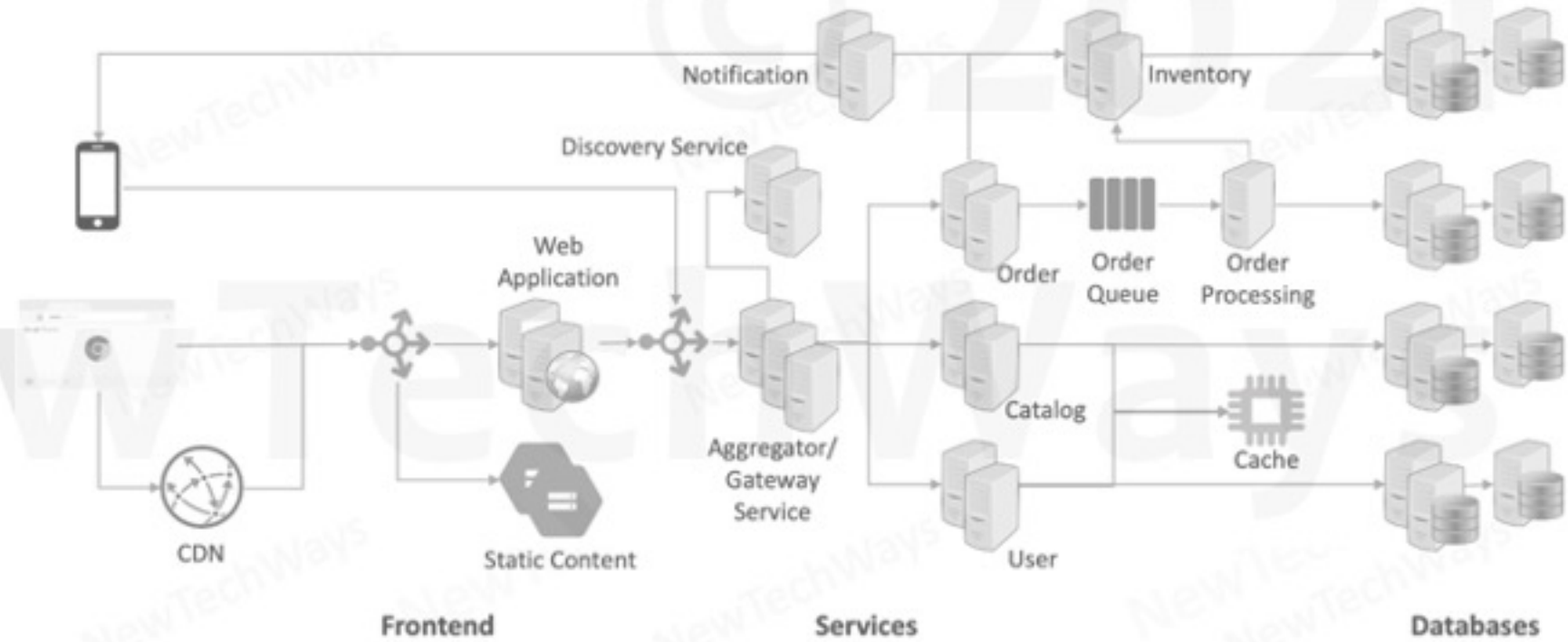


System Deployment

- Deployments
  - Application
  - Infrastructure
  - Operations
- Large-Scale Deployments
  - Virtual Machines
  - Docker Containers
  - Kubernetes
- System Upgrades
  - Rolling Upgrades
  - Blue-Green Deployments
  - Recreate Deployments
  - Canary Deployment

Large Scale Deployment - Challenges

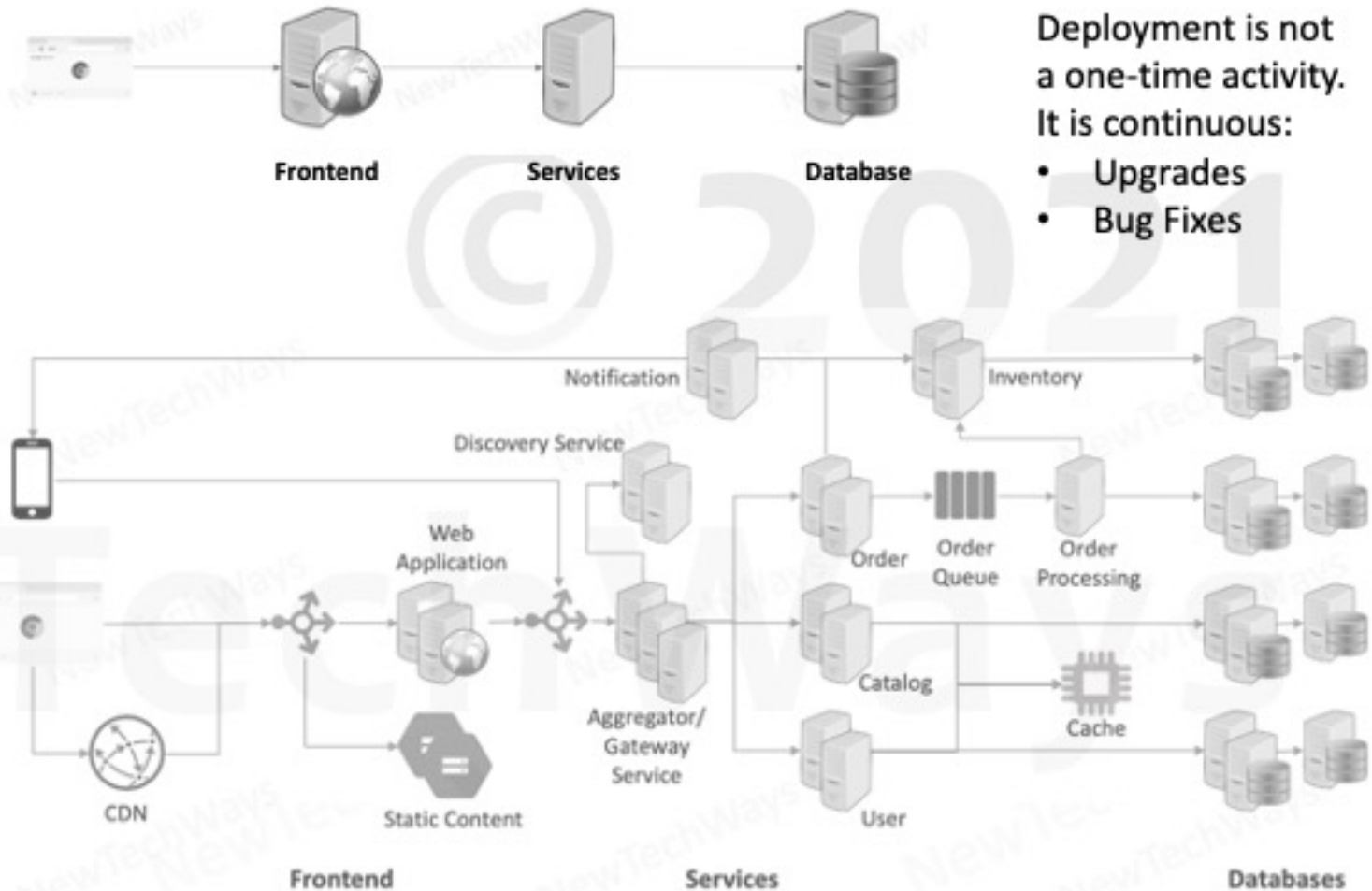
- Application Deployment
- Infrastructure Deployment
- Operations



NewTechWays

Application Deployment

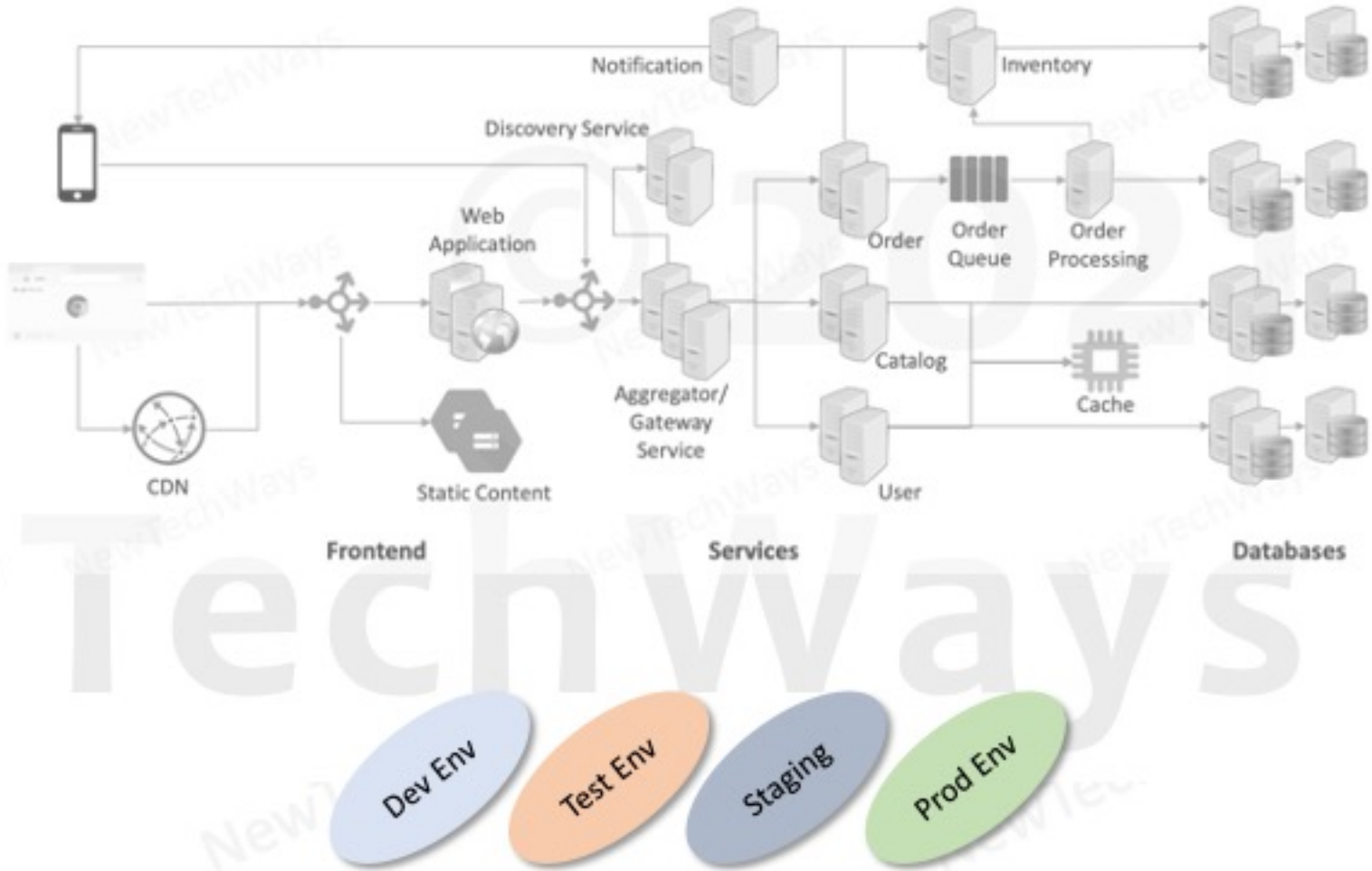
- Web Apps
  - Replicas
- Services
  - Microservices & Replicas
- Databases
  - RDBMS, NoSQL
  - Replication & Partitioning
- Message Queues
  - Replication & Partitioning
- Caches
- Directory Servers/LDAP
- Content Storage
- Log File Storage
- Search & Analytics



NewTechWays

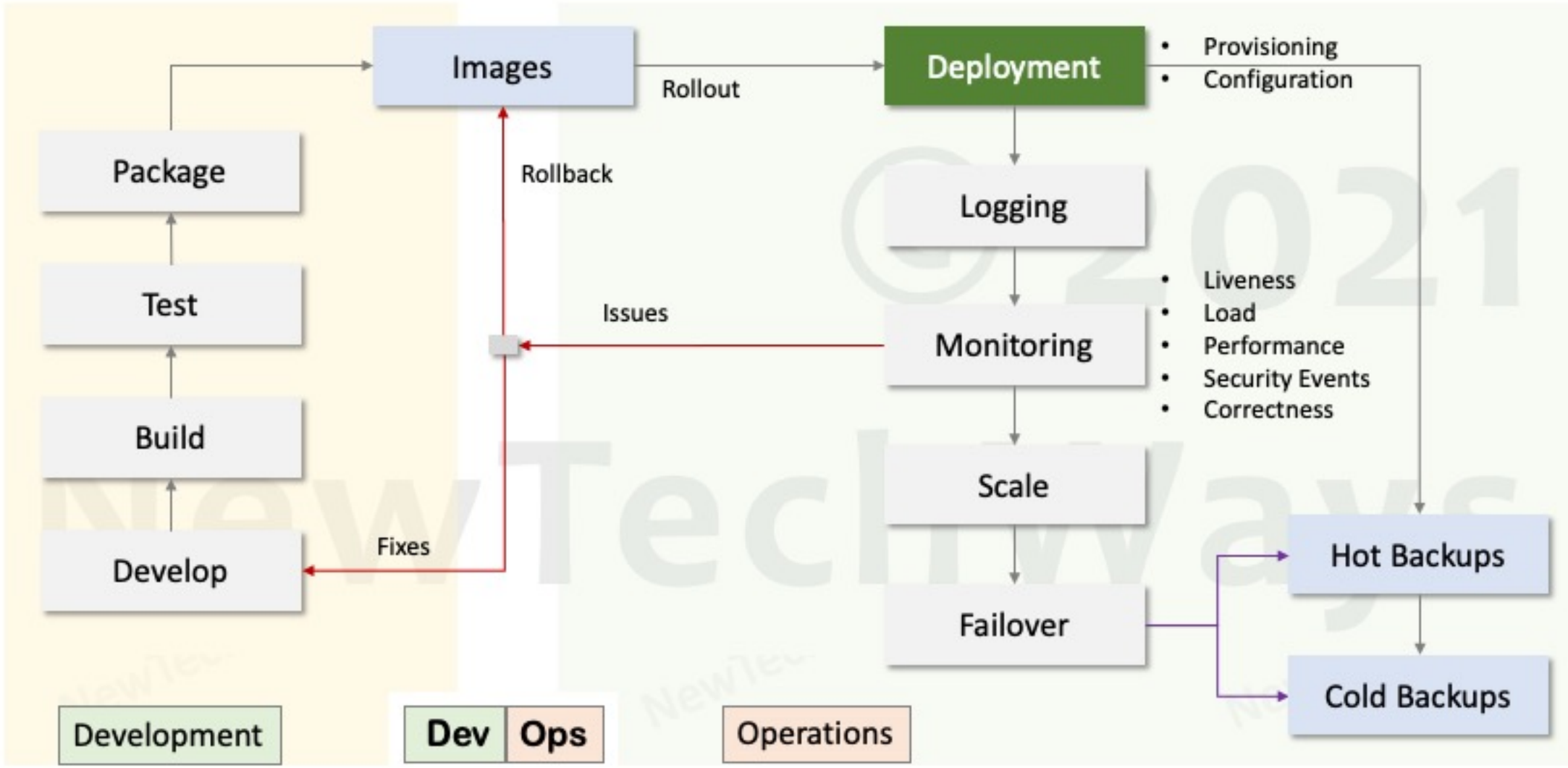
Infrastructure Deployment

- Compute Infrastructure
  - Compute VMs
    - CPU, RAM, Disks
- Network
  - Routing, Domains
    - Datacenters
  - Internet access
  - Secure access
    - Firewalls, Certificates
- Load Balancers
  - HLB, SLB
- DNS & Discovery Services
- Storage
  - Content, VM/Container Images, Backups, Logs
- Mail Servers
- CDN



NewTechWays

Operations



NewTechWays

Modern Deployment Solutions

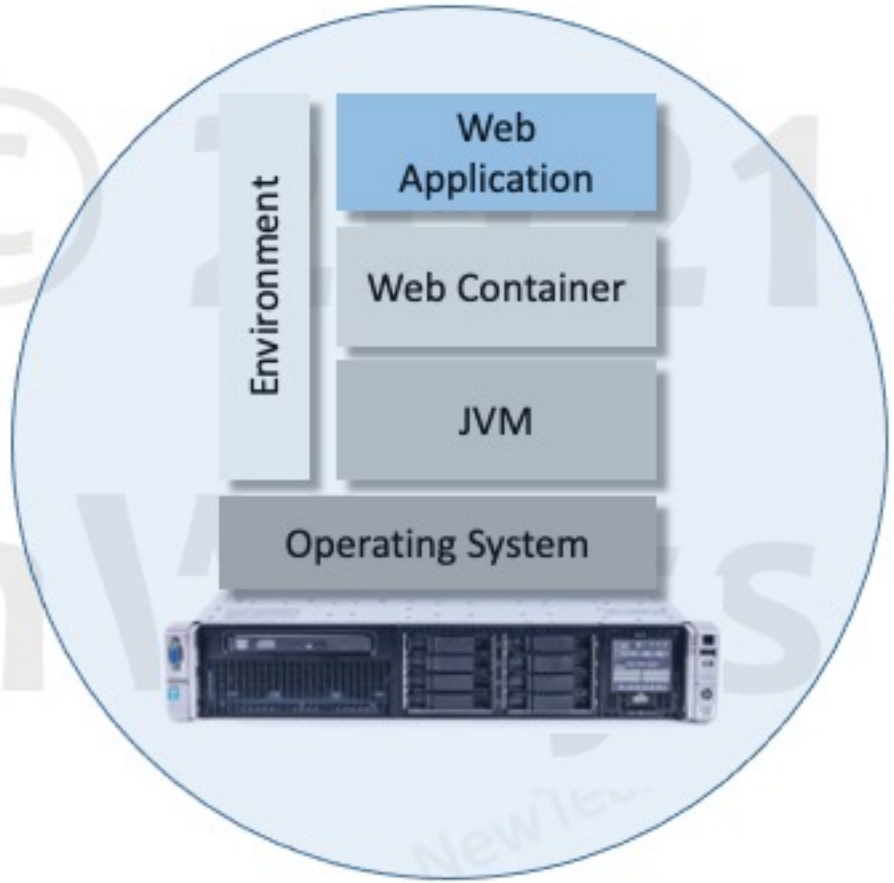
Application Deployment	Containers	
Infrastructure Deployment	Cloud	 
Operations	Kubernetes	
Automation	DevOps Tools	  

Application Deployment

Component Deployment

- Process
  - Install JVM
  - Install Web Container
    - Configuration
  - Deploy Web App
    - Configuration
- Issues
  - Error Prone
  - Time consuming
  - Repetitive

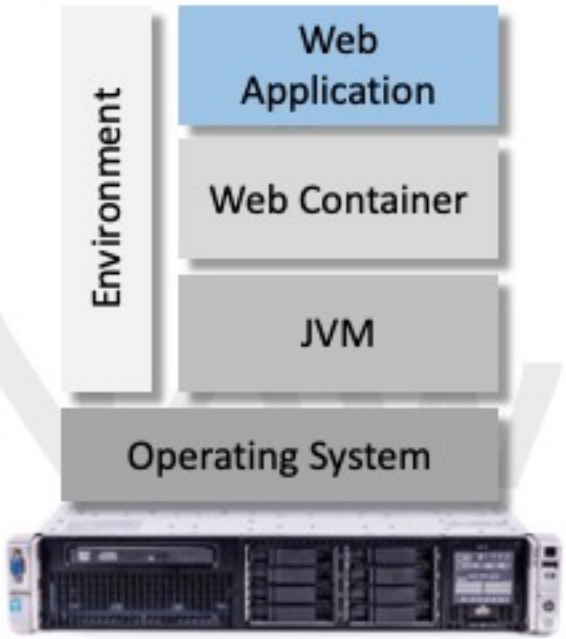
Web Application



NewTechWays

Component Deployment Automation

- Provision a machine with required operating system
- Run a script to install the required software



- ✓ Reliable
- ✓ Easily repeatable
- ✓ Less time consuming

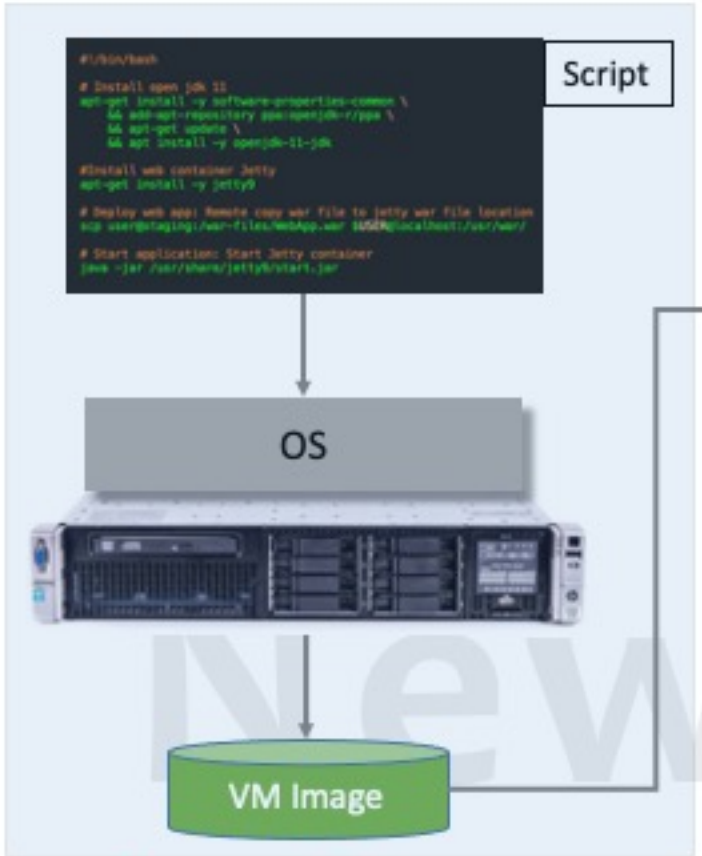
```
#!/bin/bash

# Install open jdk 11
apt-get install -y software-properties-common \
    && add-apt-repository ppa:openjdk-r/ppa \
    && apt-get update \
    && apt install -y openjdk-11-jdk

#Install web container Jetty
apt-get install -y jetty9

# Deploy web app: Remote copy war file to jetty war file location
scp user@staging:/war-files/WebApp.war $USER@localhost:/usr/war/

# Start application: Start Jetty container
java -jar /usr/share/jetty9/start.jar
```

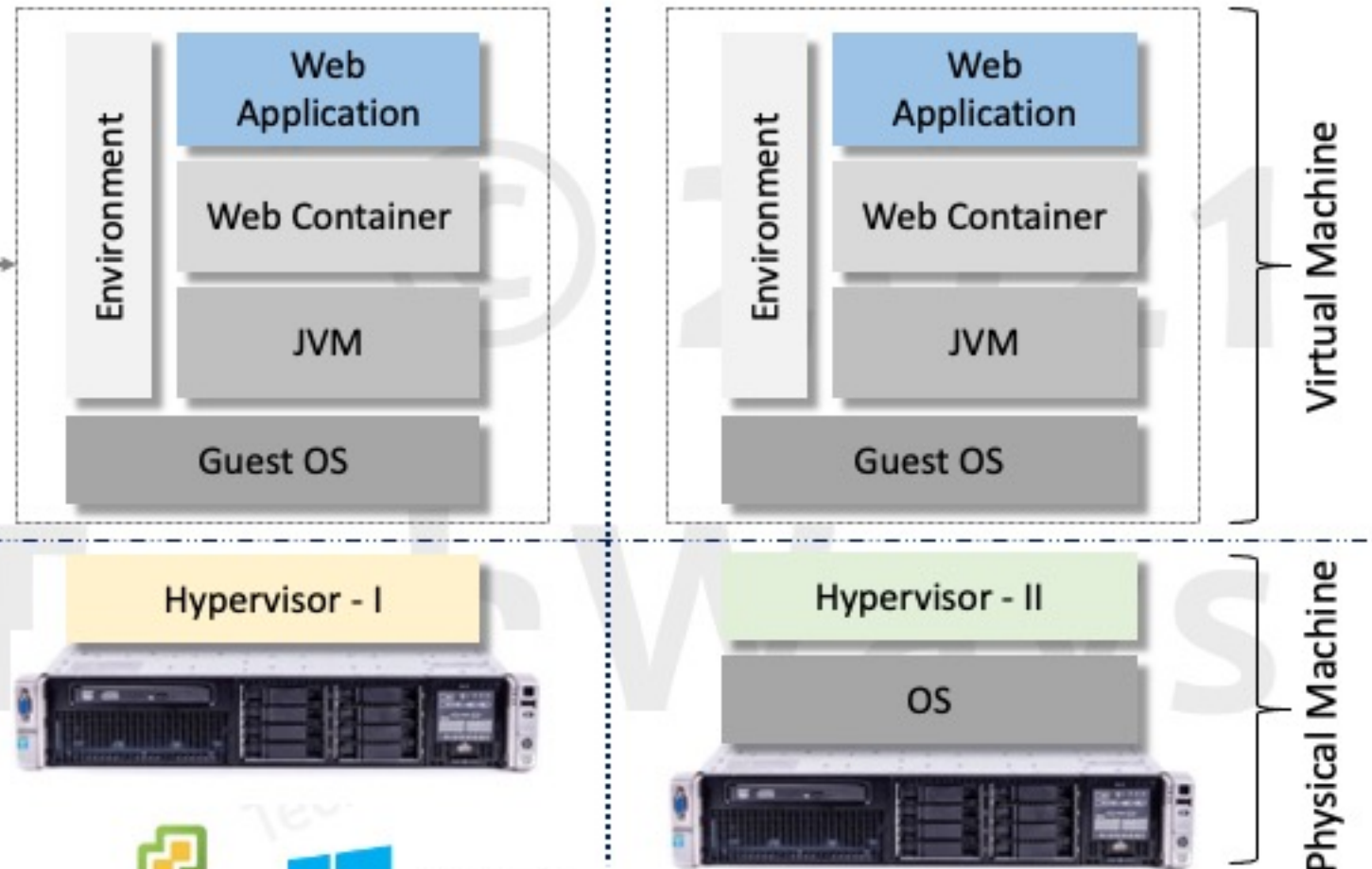
- ✓ Idempotent
- ✓ Declarative
- ✓ Remote

CHEF ANSIBLE

NewTechWays

Deployment With Virtual Machines

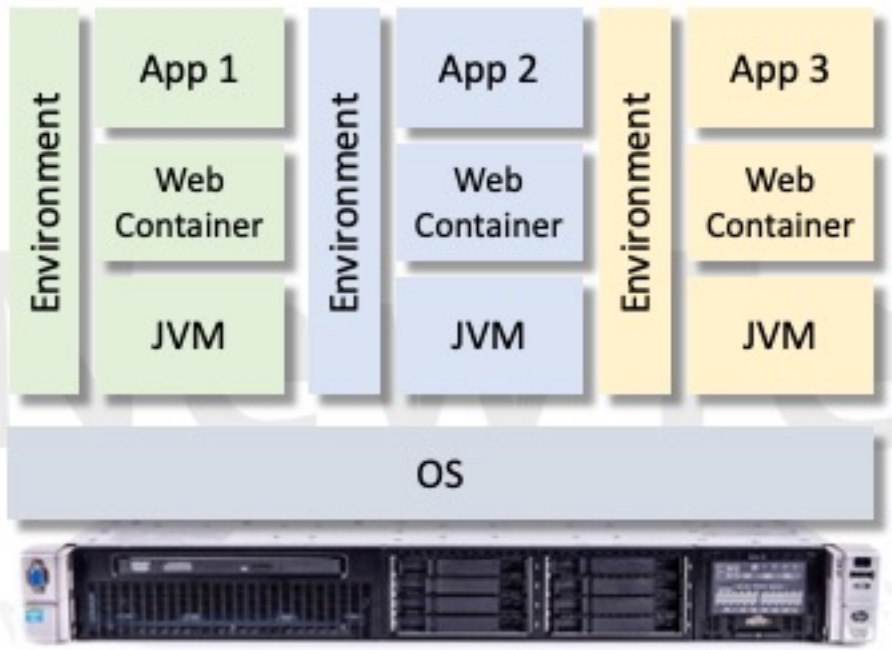


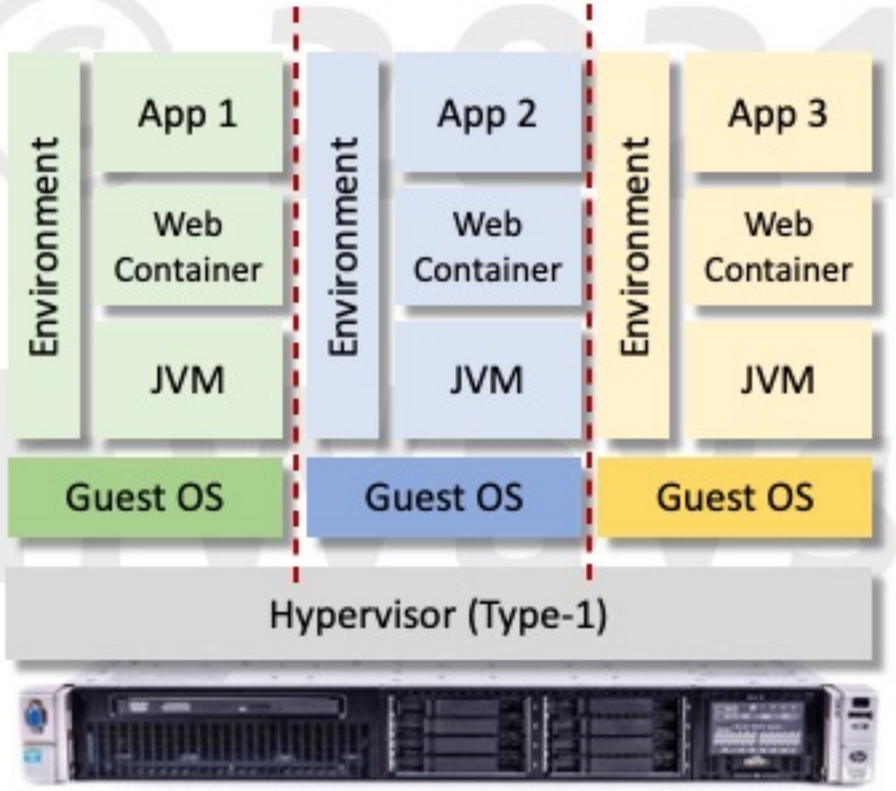


- ✓ Very Reliable
- ✓ Efficiently Repeatable
- ✓ Less Time Consuming

Isolation With Virtual Machines

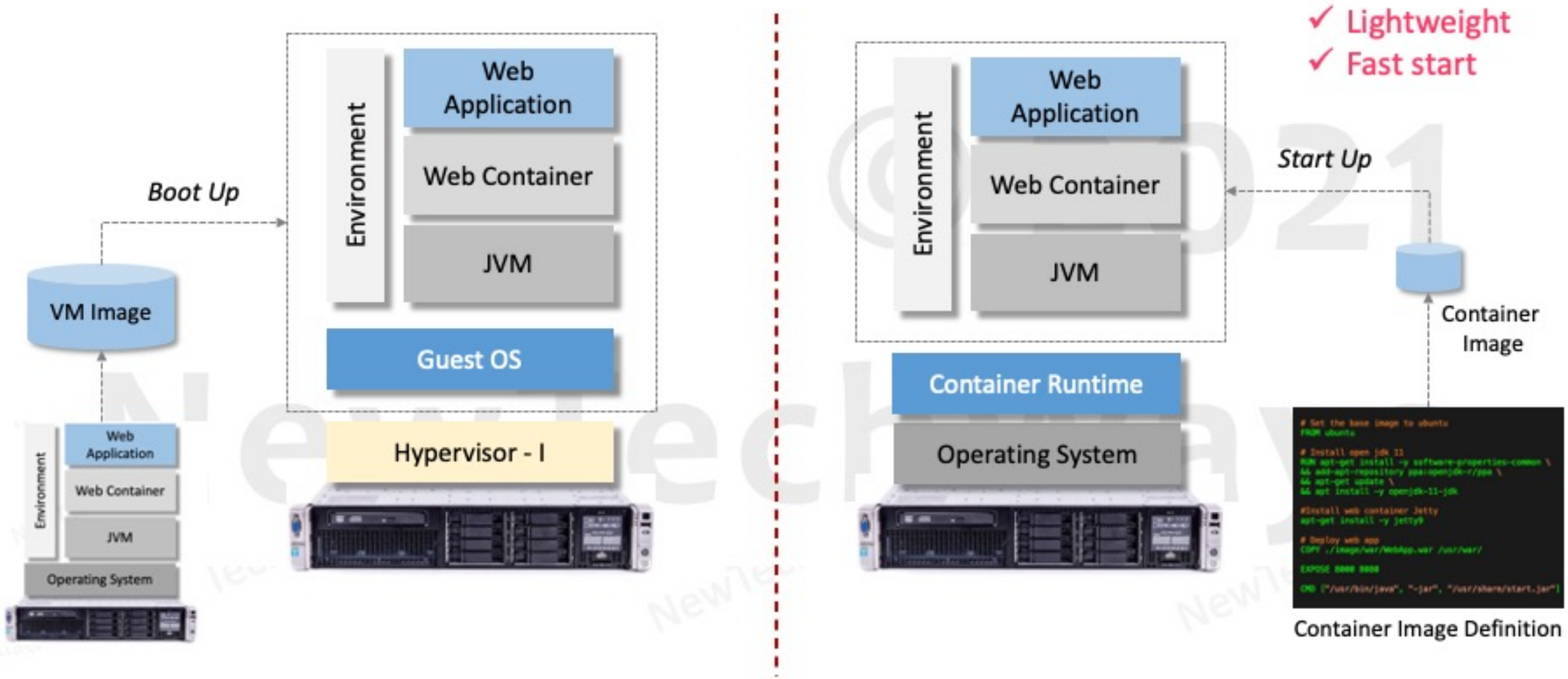
- No shared resources between applications running virtual machine on the same host machine





NewTechWays

Deployment With Containers



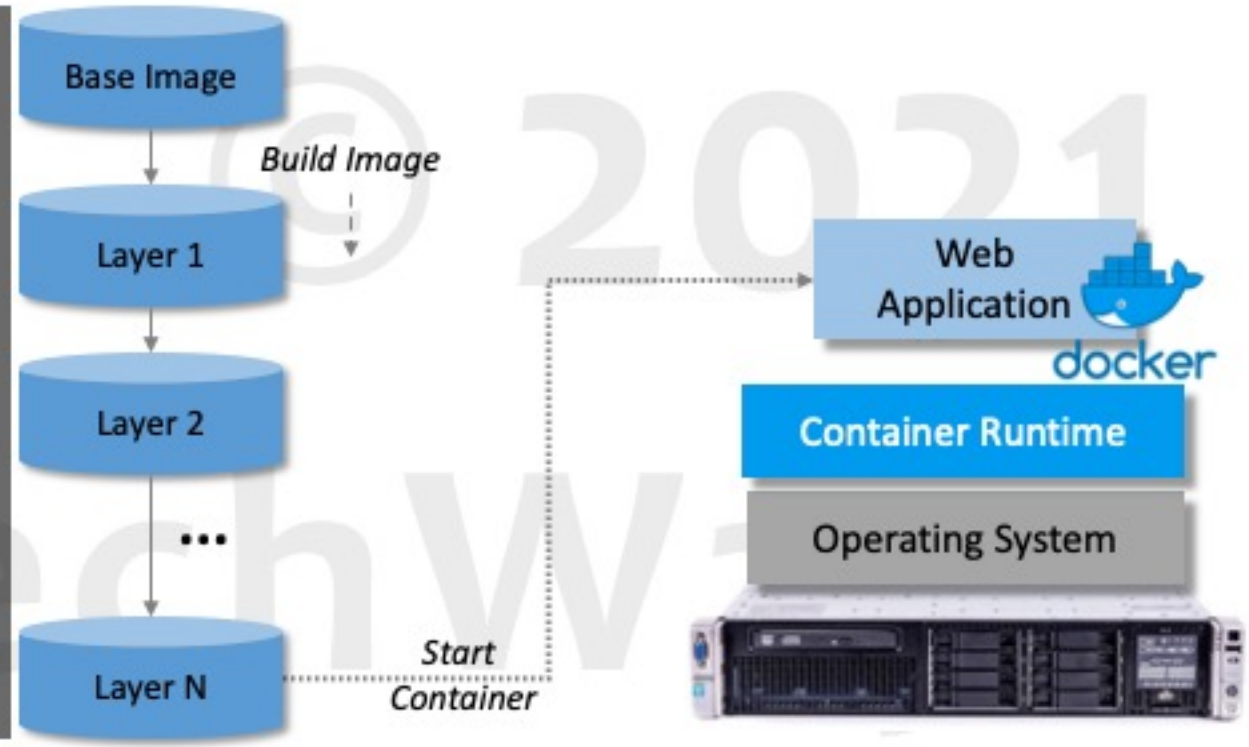
- ✓ Lightweight
- ✓ Fast start

NewTechWays

Docker Containers

Dockerfile

```
# Set the base image to ubuntu
FROM ubuntu

# Install open jdk 11
RUN apt-get install -y software-properties-common \
&& add-apt-repository ppa:openjdk-r/ppa \
&& apt-get update \
&& apt install -y openjdk-11-jdk

#Install web container Jetty
apt-get install -y jetty9

# Deploy web app
COPY ./image/war/WebApp.war /usr/war/

EXPOSE 8000 8080

CMD ["/usr/bin/java", "-jar", "/usr/share/start.jar"]
```

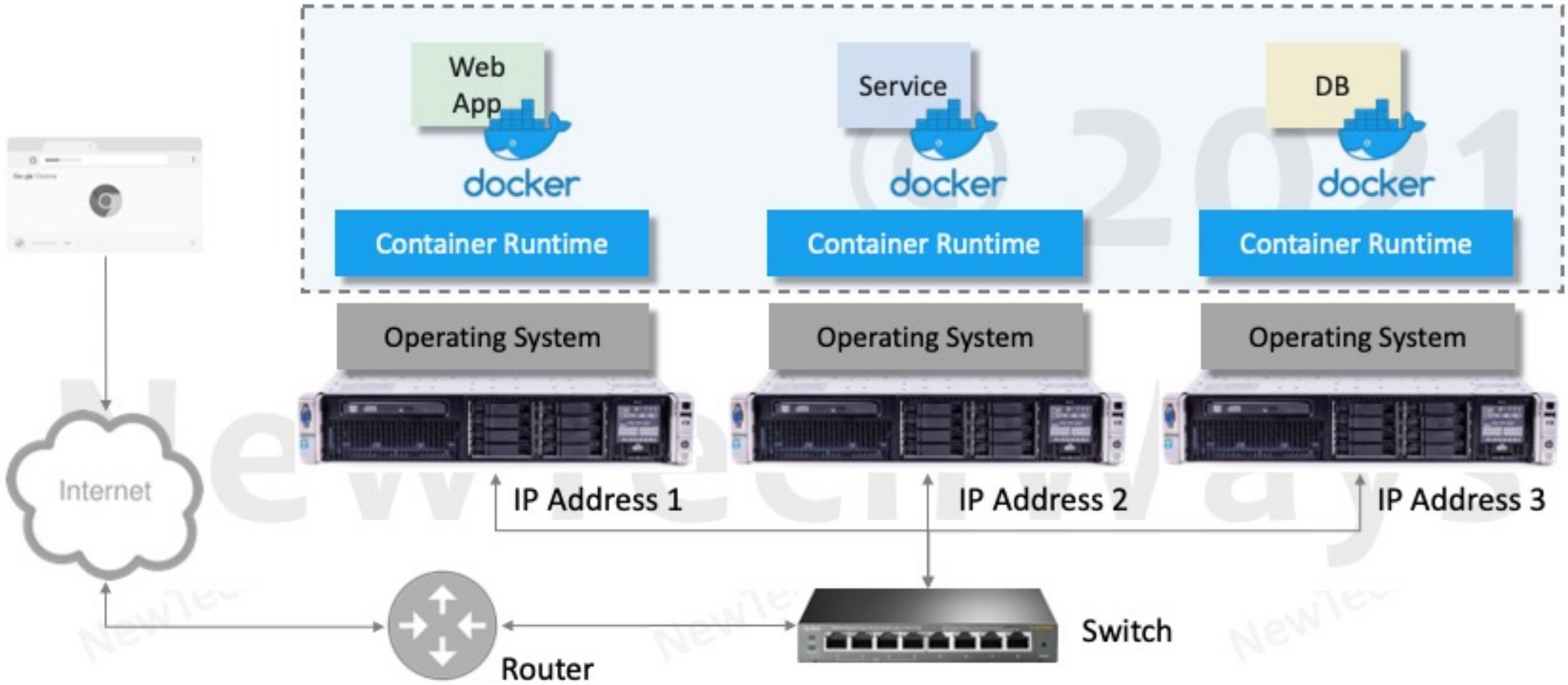


Docker Image

✓ Version control image instructions

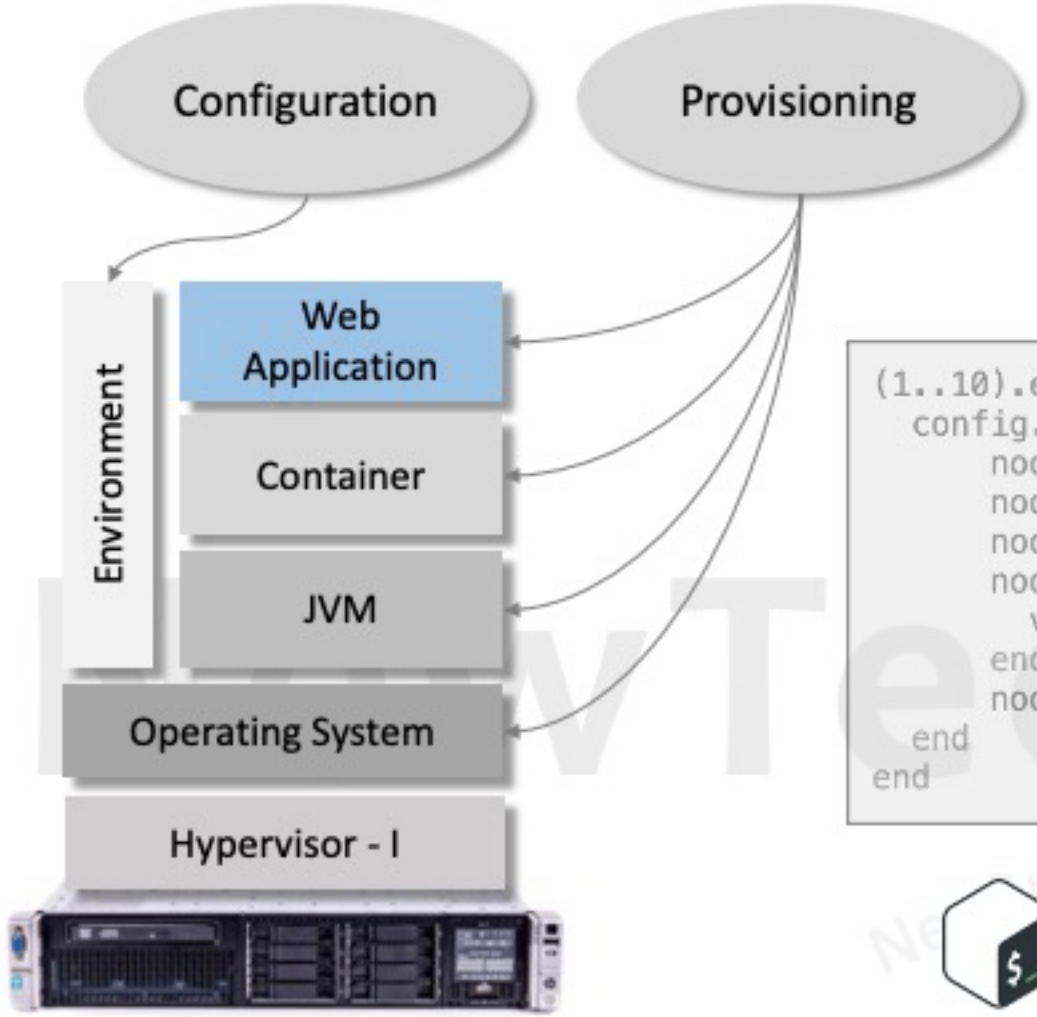
Infrastructure Deployment

Infrastructure Requirements



NewTechWays

Provisioning & Configuration



- Network Setup ?
- Bare Metal Provisioning?

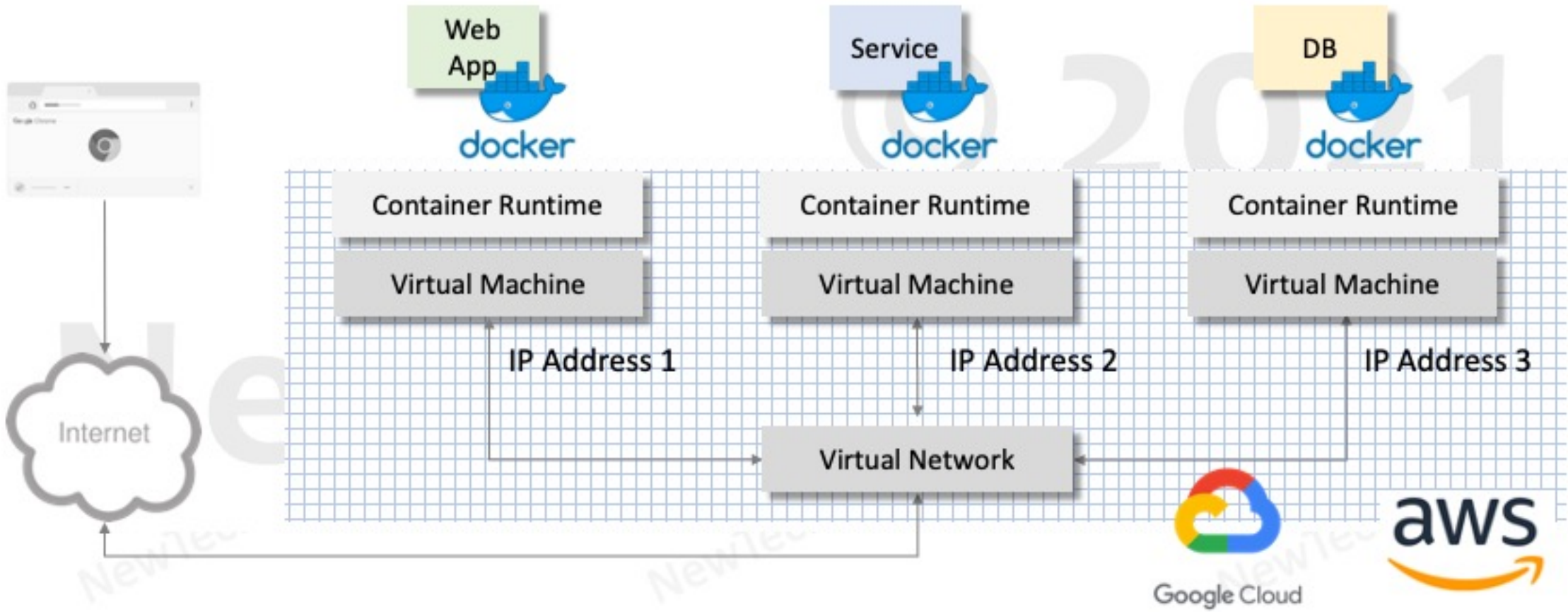
VAGRANT

```
(1..10).each do |i|
  config.vm.define "web#{i}" do |node|
      node.vm.box = "UbuntuTrusty64Web"
      node.vm.hostname = "web#{i}"
      node.vm.network :private_network, ip: "192.168.56.4#{i}"
      node.vm.provider "virtualbox" do |vb|
        vb.memory = "5120"
      end
      node.vm.provision :shell, path: "web/bootstrap.sh"
  end
end
```

BASH CHEF ANSIBLE

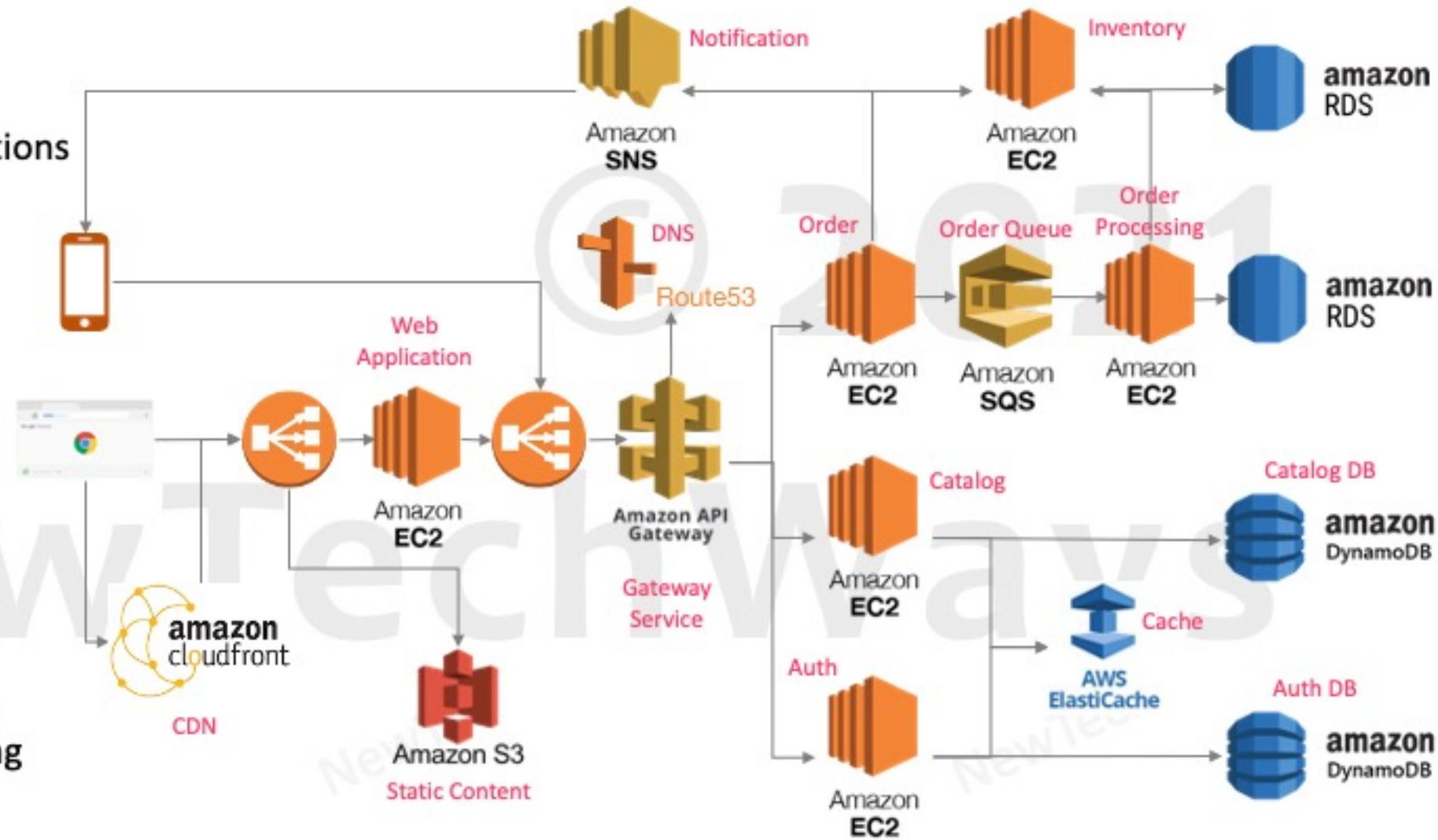
NewTechWays

Deployment on Cloud



Deployment with Cloud Stack

- Compute VM
  - On Demand
  - Multiple global locations
- Serverless Compute
- Network
  - Global Connectivity
  - Firewalls
- Fully Cloud Managed Services
  - Load Balancers
    - External, Internal
  - Databases
    - RDBMS, NoSQL
  - Storage
    - Content, Backups
  - Logging & Monitoring

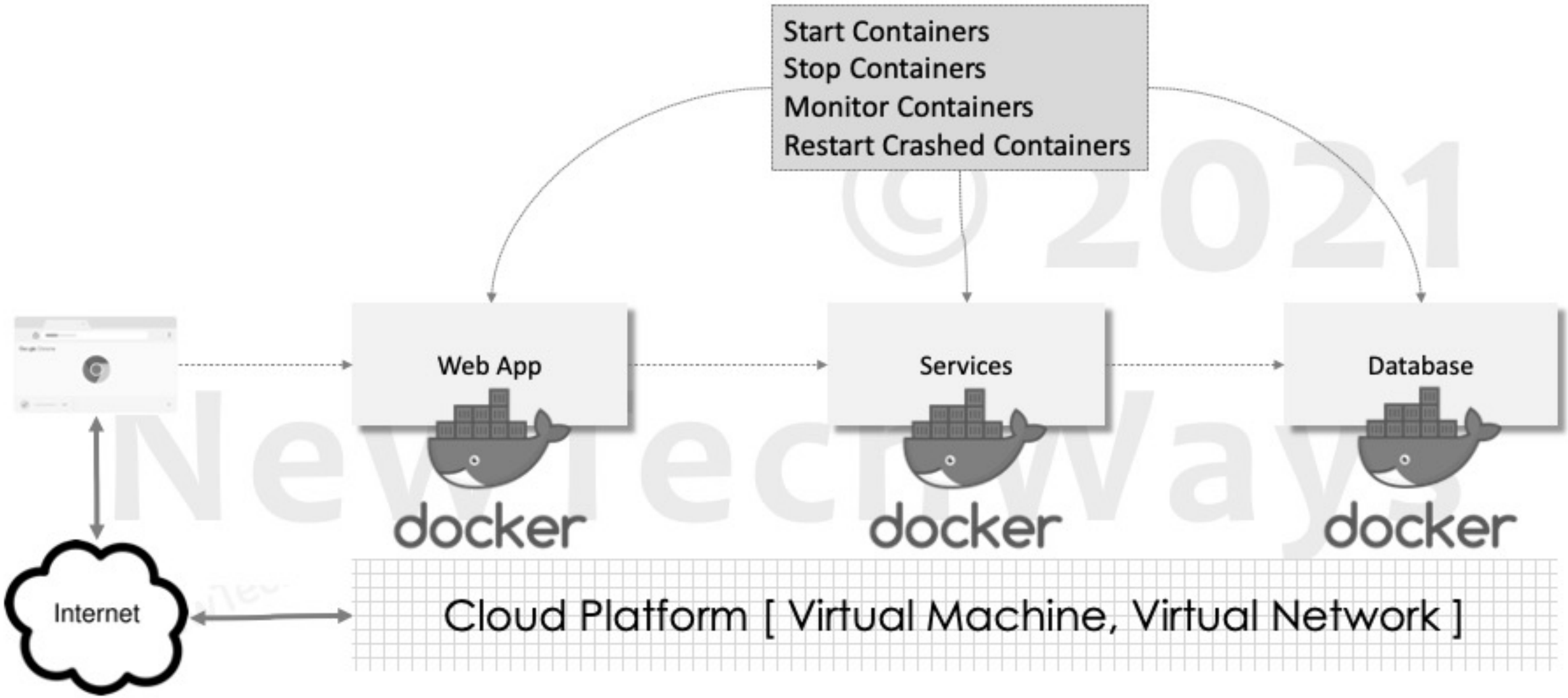


NewTechWays

Deployment & Operations with Kubernetes

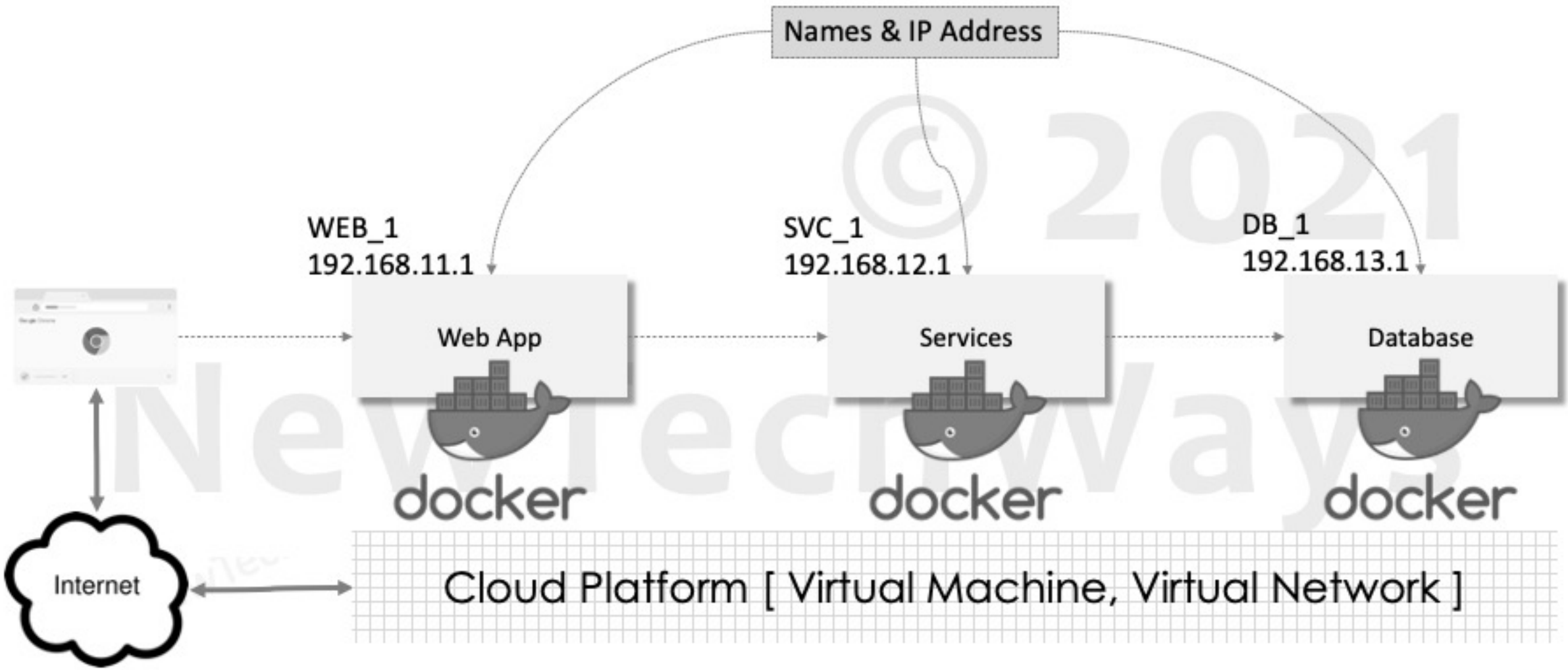
NewTechWays

Lifecycle Management



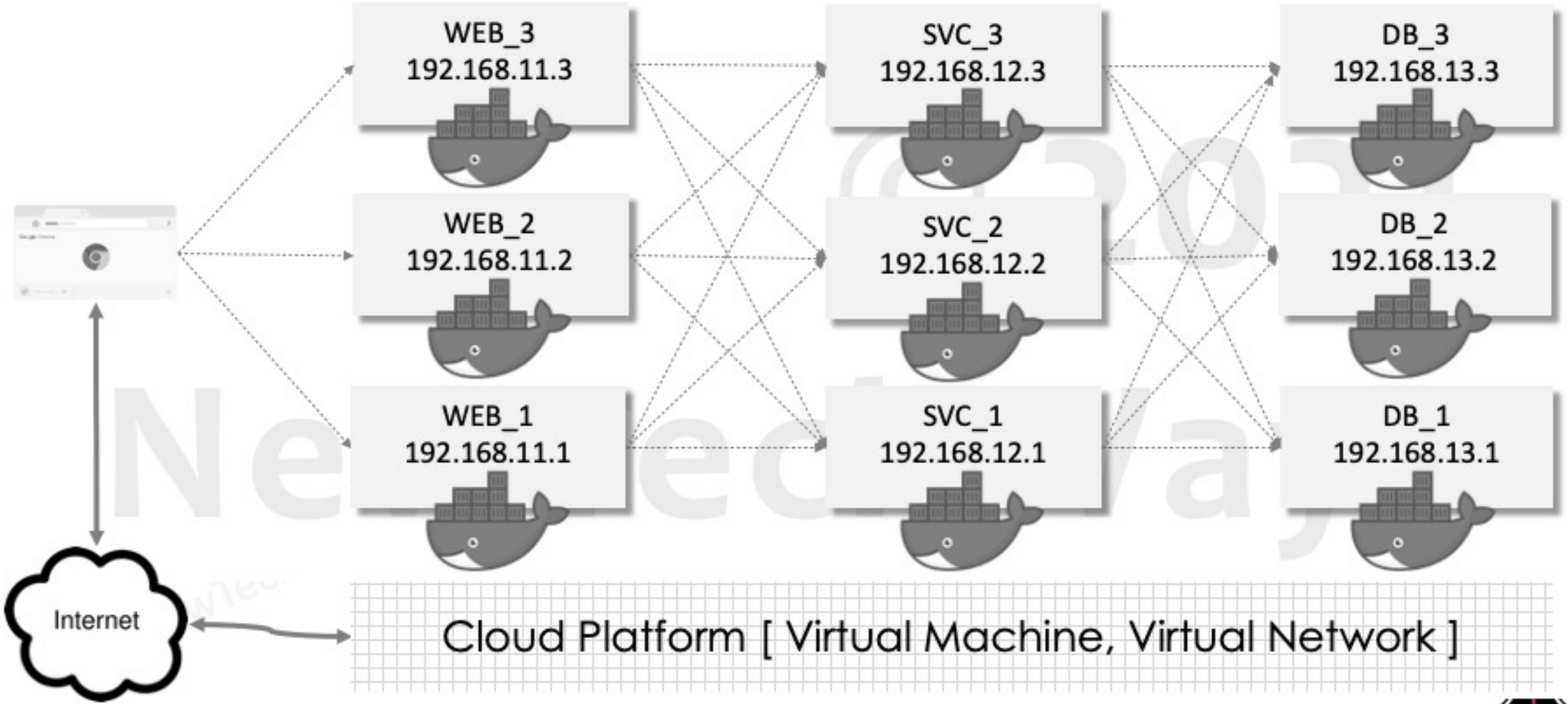
NewTechWays

Naming & Addressing



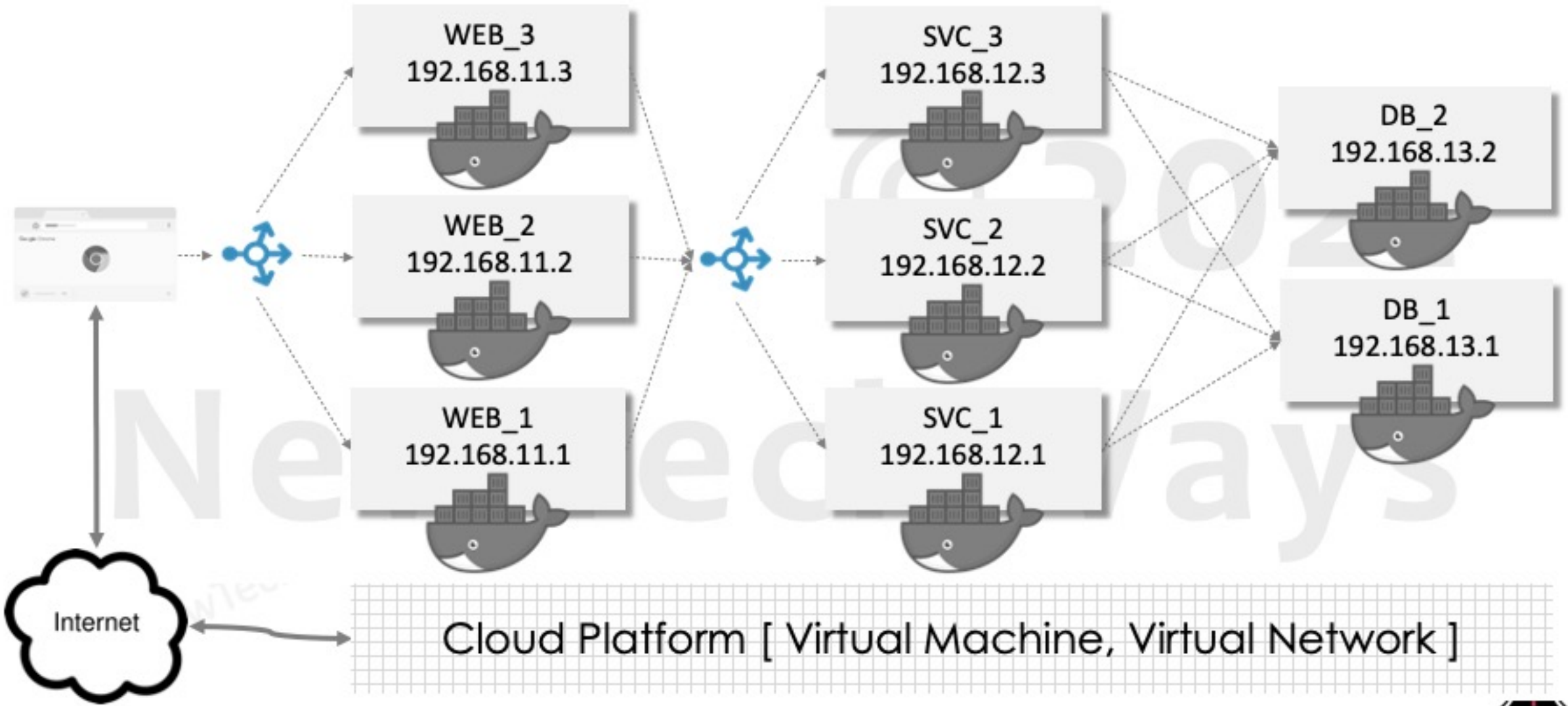
NewTechWays

Scaling To Multiple Instances



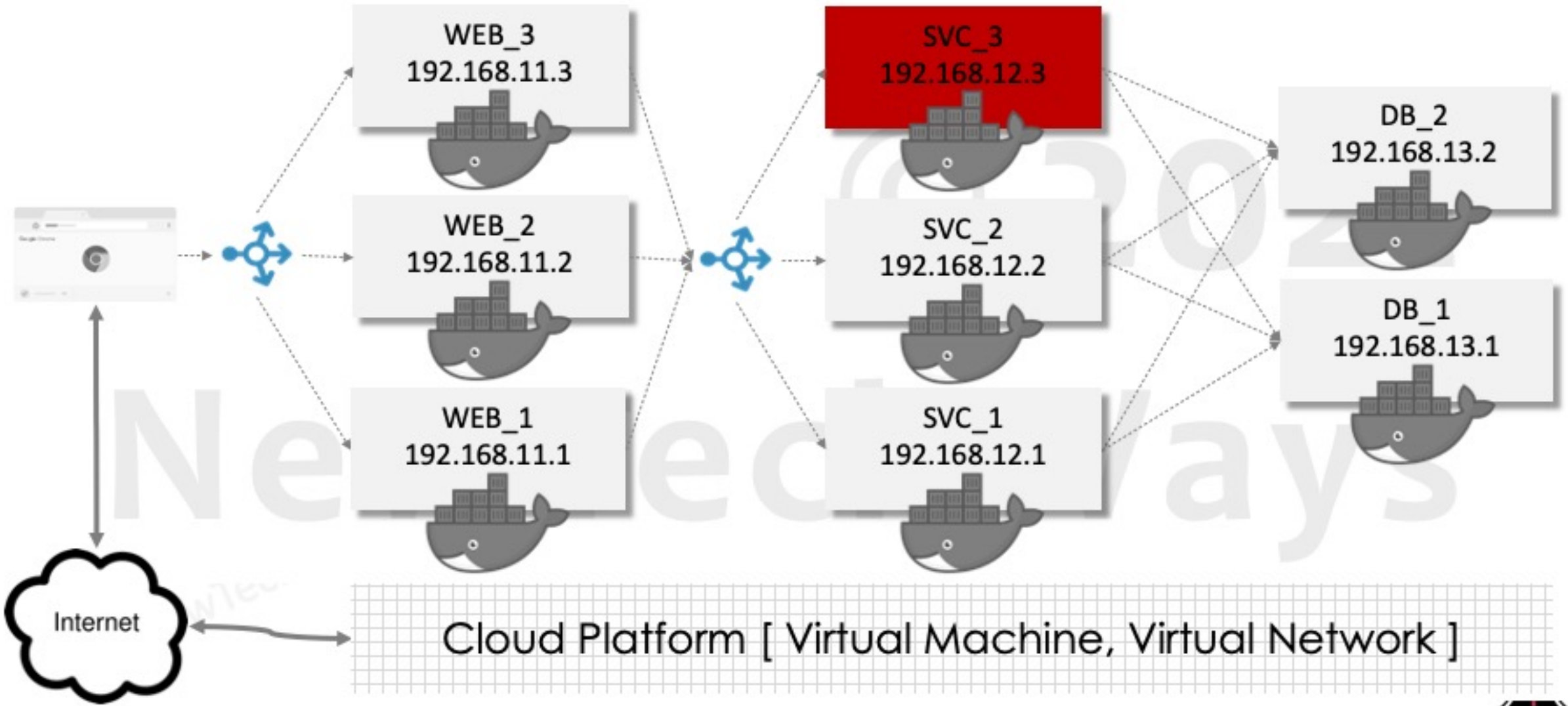
NewTechWays

Load Balancing



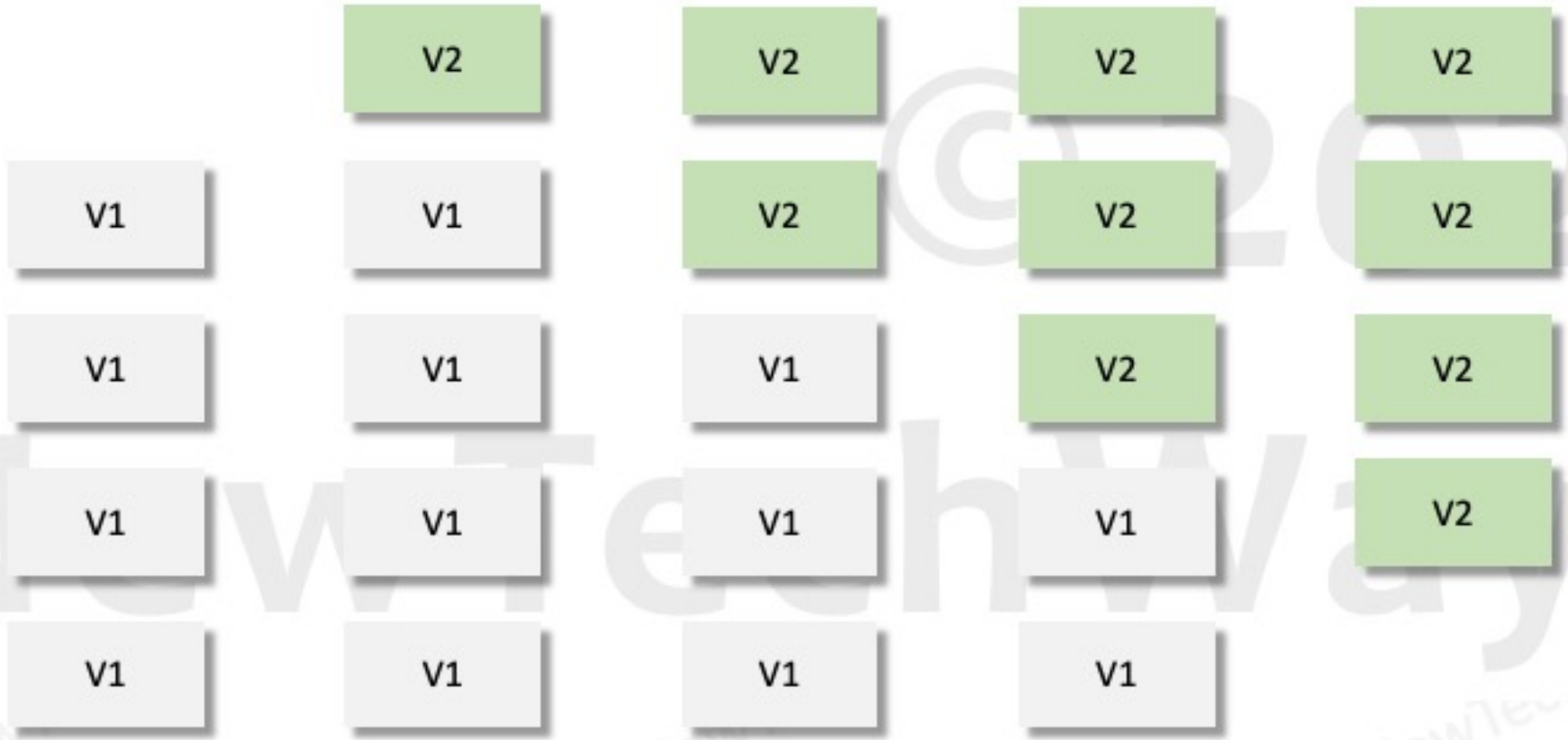
NewTechWays

High Availability



NewTechWays

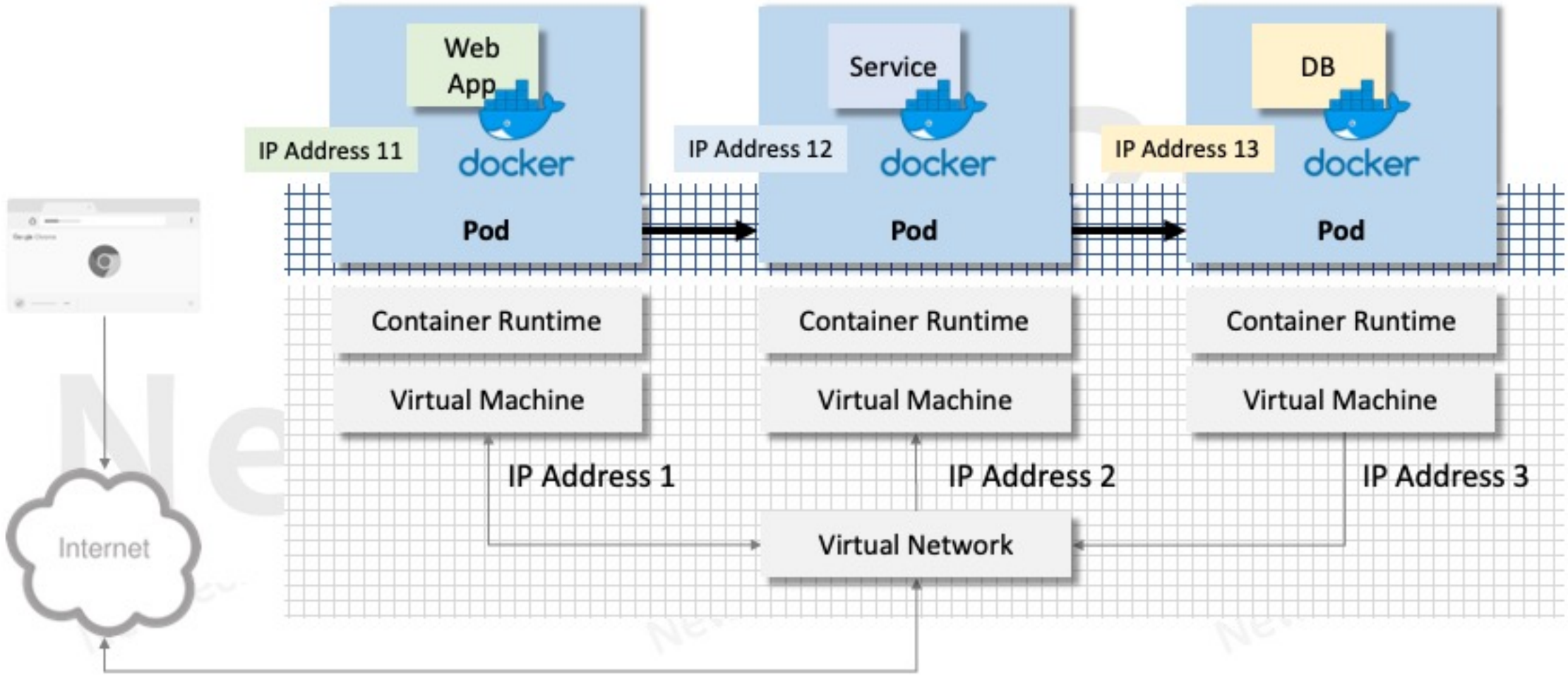
Rolling Upgrades



Kubernetes

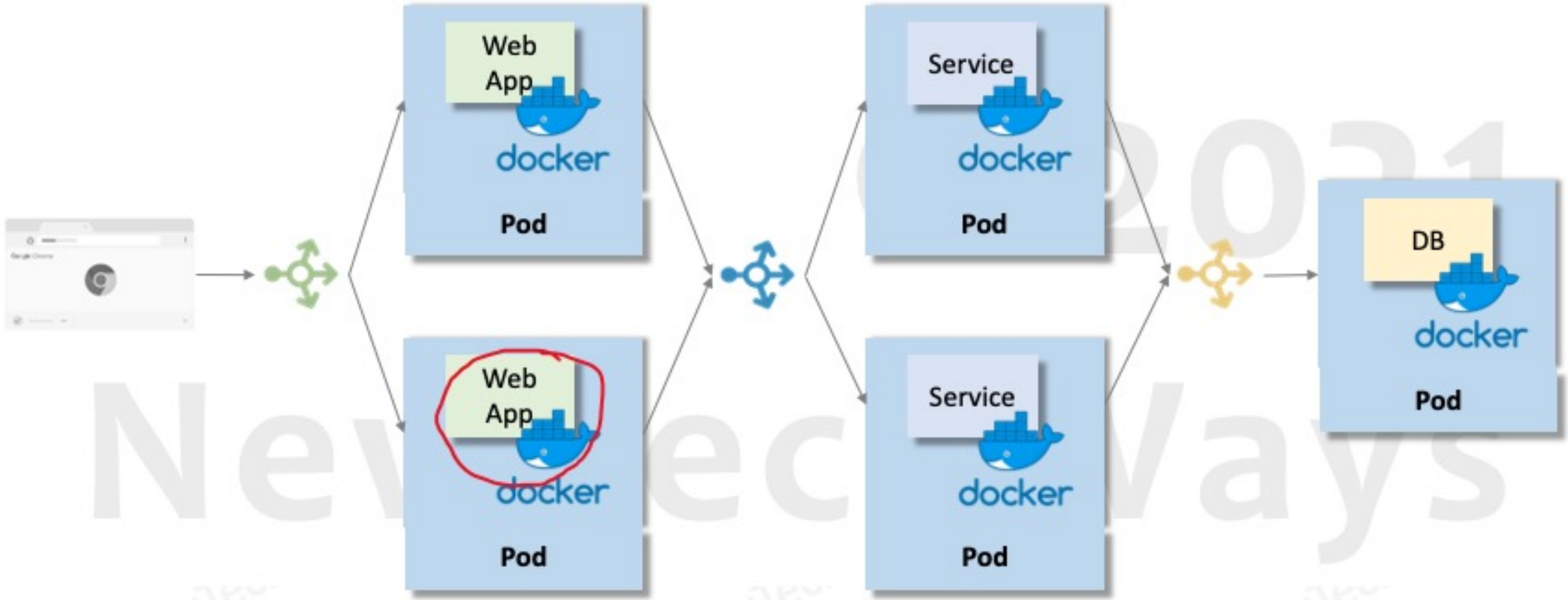
- Service Naming & Discovery
  - Using DNS
- Container Lifecycle Management
  - Health checks, restarts, replacements of unhealthy containers
- Load balancing
- Automated rollouts and rollbacks
- Automatic bin packing
  - Allows to specify resource (CPU, memory) requirements for containers
- Automating mounting of storage systems of choice

Deployment With Kubernetes



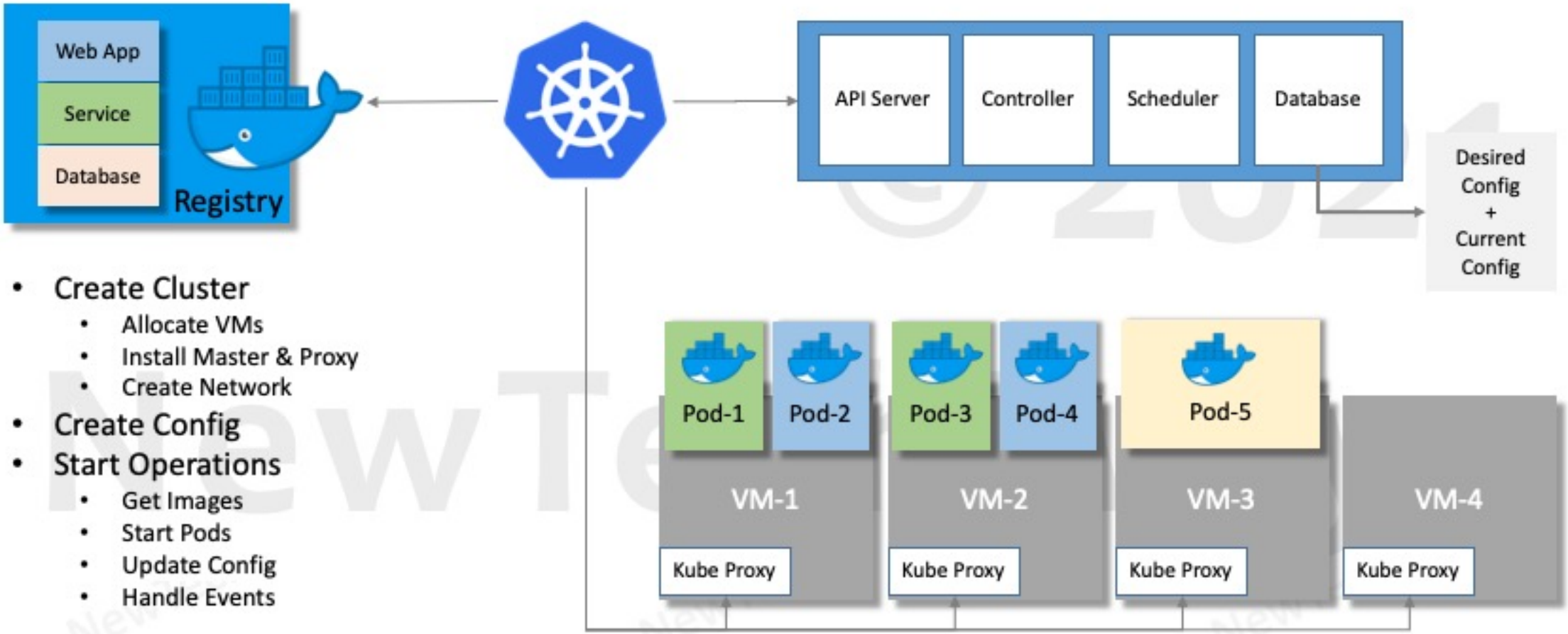
NewTechWays

Deployment With Kubernetes



NewTechWays

Kubernetes Architecture

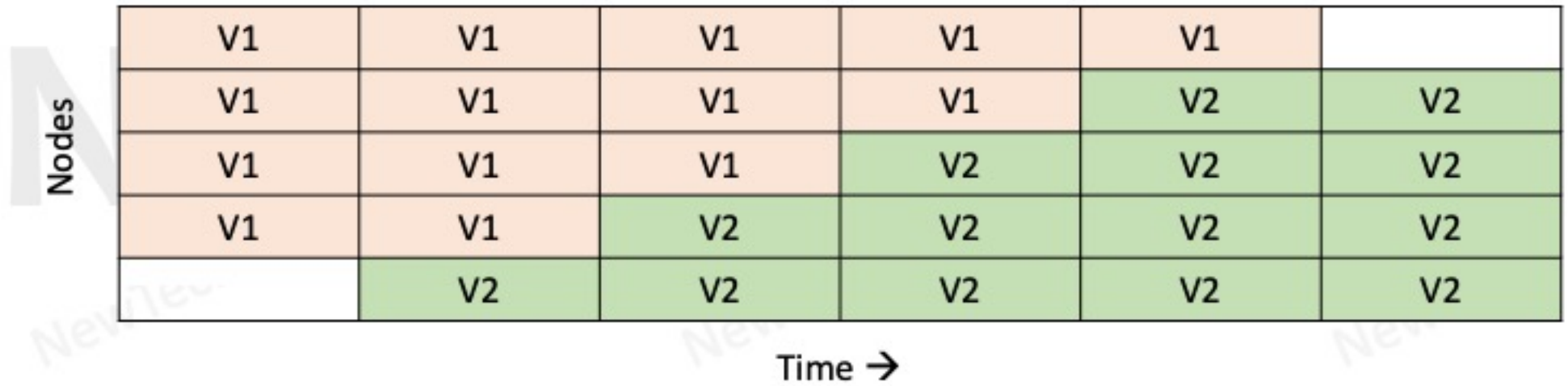


- Create Cluster
  - Allocate VMs
  - Install Master & Proxy
  - Create Network
- Create Config
- Start Operations
  - Get Images
  - Start Pods
  - Update Config
  - Handle Events

Deploying Upgrades/Fixes

Rolling Update

- Used when it is okay to have both old version and new version simultaneously
- Old version is incrementally replaced by a new version as the new version is incrementally rolled-out
- New version nodes are added to load-balancer node-pool and old version nodes are removed



Canary Deployment

- New update is exposed to a small fraction of requests
- Once confident, the update is rolled out completely
- No downtime involved

Recreate Deployment

- Old and New Versions cannot run at the same time
- Migrations or Data Transformations required before new code starts
- Involves a downtime

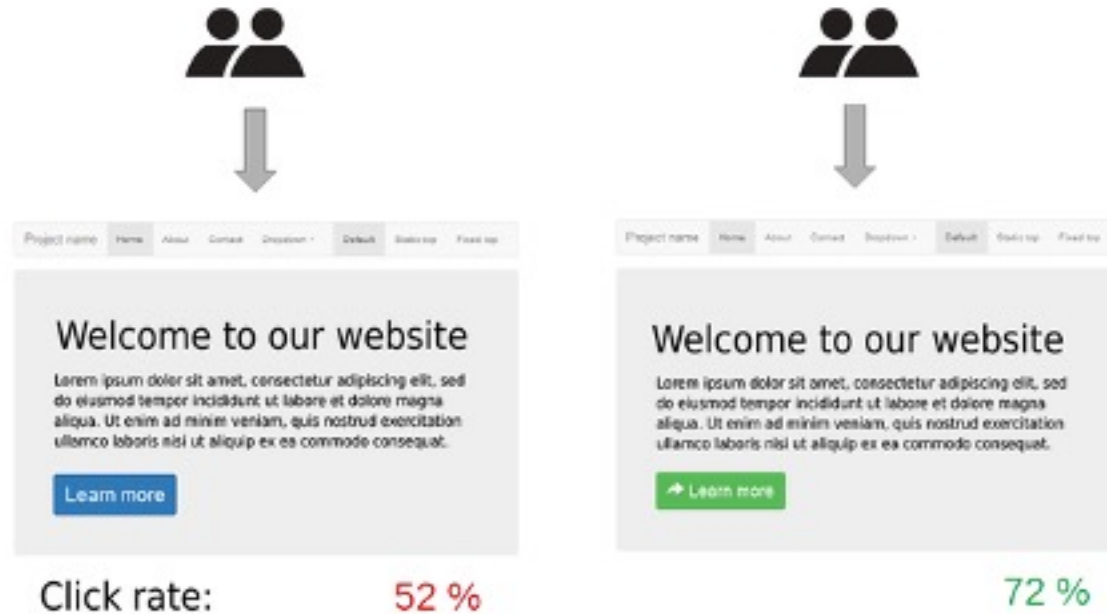
Blue Green Deployment

- Entire environment is switched to a new environment using a reverse proxy
- Quick rollback to old environment on failure
- Goal is to avoid any downtime at the cost of extra hardware

V1	V1	V1	V1	V1		
V1	V1	V1	V1	V1		
V1	V1	V1	V1	V1		
V1	V1	V1	V1	V1		
		V2	V2	V2	V2	V2
		V2	V2	V2	V2	V2
		V2	V2	V2	V2	V2
		V2	V2	V2	V2	V2

A/B Testing

- Much like canary release which splits requests
  - May or may not track users
  - Low % of requests for new version
- A/B Testing is augmented Canary release that tracks users
- Compares user behavior for two feature variants
- User splits may be even for A/B
- Feature toggles (flags), smart logic switches are used



Summary

- Deployment
  - Application
  - Infrastructure
  - Operations
- Application Deployment
  - Virtual Machines
  - Docker Containers
- Infrastructure Deployment
  - Cloud
  - Kubernetes
- Upgrading Deployments
  - Recreate
  - Rolling
  - Canary
  - Blue-Green
  - A/B Testing

Thanks!

NewTechWays

https://www.newtechways.com