

Infrastructure Architecture

[Author Name]

Université de Toulon, LIS UMR CNRS 7020

2026-10-14

Table des matières

- 1 Purpose of this Document 1
- 2 Infrastructure Strategy 1
- 3 Infrastructure Model 1
- 4 Deployment Architecture 1
- 5 Physical Architecture (Cloud-specific) 2
- 6 Network Architecture 2
- 7 Platform Configuration 3
- 8 Environment Matrix 3
- 9 Infrastructure as Code Examples 4
 - 9.1 Terraform Example 4
 - 9.2 Kubernetes Example 4
- 10 Operational Considerations 5
- 11 Security Controls 5
- 12 Cost Optimization 6
- 13 Template Validation Checklist 6
- 14 Related Views 6

1 Purpose of this Document

This document defines the infrastructure architecture for [System Name], including deployment models, environments, and operational requirements.

2 Infrastructure Strategy

Cloud Strategy: [Cloud-native|Hybrid|Multi-cloud|On-premise]

Deployment Philosophy: [DevOps approach]

Infrastructure as Code: [Approach and tools]

Key Principles:

- [Principle 1]
- [Principle 2]
- [Principle 3]

3 Infrastructure Model

Architecture

Pattern: [Microservices|Serverless|Hybrid|etc.]

Deployment Model: [Blue-green|Canary|Rolling|etc.]

Container Platform:

Orchestration: [Kubernetes|ECS|Nomad|etc.]

Registry: [ECR|Docker Hub|GitHub Container Registry|etc.]

Management: [Tools used]

Scale Requirements:

Peak Load: [Metrics]

Normal Load: [Metrics]

Growth Projections: [Expected changes]

4 Deployment Architecture

graph TD

subgraph "Public Internet"

A[End Users]

B[Partner Systems]

end

```

subgraph "Edge Services"
    C[CDN]
    D[WAF]
    E[Load Balancer]
end

subgraph "Application Environment"
    F[Container Cluster]
    G[Serverless Functions]
    H[Managed Services]
end

subgraph "Data Storage"
    I[Databases]
    J[Object Storage]
    K[Caches]
end

A --> C
B --> E
C --> E
D --> E
E --> F
E --> G
F --> H
F --> I
F --> K
G --> I
G --> J

```

5 Physical Architecture (Cloud-specific)

```

@startuml
!include <awslib/AWSCommon>
!include <awslib/General/all>
!include <awslib/Compute/all>
!include <awslib/Storage/all>
!include <awslib/Database/all>

package "Production Environment" {
    [Load Balancer] as LB

```

```

[Web Tier] as Web
[Application Tier] as App
[Data Tier] as Data
[Caching Layer] as Cache

LB --> Web
Web --> App
App --> Data
App --> Cache
}

cloud "External Services" {
    [Third-party APIs]
}

App --> [Third-party APIs]
@enduml

```

6 Network Architecture

```

@startuml
nwdiag {
    internet [shape = cloud];

    network public {
        internet;
        web01 [address = "10.0.0.1"];
        web02 [address = "10.0.0.2"];
        lb01 [address = "10.0.0.100"];
    }

    network private {
        web01 [address = "10.0.1.1"];
        web02 [address = "10.0.1.2"];
        app01 [address = "10.0.1.10"];
        app02 [address = "10.0.1.11"];
    }

    network data {
        app01 [address = "10.0.2.10"];
        app02 [address = "10.0.2.11"];
        db01 [address = "10.0.2.20"];
        db02 [address = "10.0.2.21"];
    }
}

```

```

    }
  }
  @enduml

```

7 Platform Configuration

```

Compute:
  Type: [VM|Container|Serverless|
etc.]
  Sizing:
    CPU: [Requirements]
    Memory: [Requirements]
    Instances: [Count/scaling]
  Scaling:
    Strategy: [Auto-scaling approach]
    Triggers: [Metrics that trigger
scaling]
    Limits: [Min/max instances]
  Resources:
    Limits: [Resource constraints]
    Requests: [Resource requests]
    Quotas: [Usage quotas]

Network:
  Topology: [Network design]
  Security:
    Zones: [Security zones]
    Firewalls: [Rules and approach]
    WAF: [Web application firewall
config]
  Performance:
    Latency: [Requirements]
    Bandwidth: [Requirements]
    Acceleration: [CDN/caching
strategies]

Storage:
  Types:
    Block: [Usage and sizing]
    Object: [Usage and sizing]
    File: [Usage and sizing]
  Backup:
    Strategy: [Backup approach]

```

```

Schedule: [Frequency]
Retention: [Policy]
Recovery:
  RTO: [Recovery Time Objective]
  RPO: [Recovery Point Objective]
  Plan: [Disaster recovery approach]

```

8 Environment Matrix

```

Environments:
  Development:
    Purpose: Feature development
    Compute: [Specifications, e.g.,
"t3.medium x 2"]
    Memory: [Allocation, e.g., "8GB
per instance"]
    Storage: [Allocation, e.g., "100GB
SSD"]
    Network: [Configuration, e.g.,
"private subnet"]
    Scaling: [Strategy, e.g.,
"manual"]
    Access: [Restrictions]

  Testing:
    Purpose: Integration and load
testing
    Compute: [Specifications]
    Memory: [Allocation]
    Storage: [Allocation]
    Network: [Configuration]
    Scaling: [Strategy]
    Access: [Restrictions]

  Staging:
    Purpose: Pre-production
verification
    Compute: [Specifications]
    Memory: [Allocation]
    Storage: [Allocation]
    Network: [Configuration]
    Scaling: [Strategy]
    Access: [Restrictions]

```

```

Production:
  Purpose: Live system
  Compute: [Specifications, e.g., "t3.large x 4"]
  Memory: [Allocation, e.g., "16GB per instance"]
  Storage: [Allocation, e.g., "500GB SSD"]
  Network: [Configuration, e.g., "private subnet"]
  Scaling: [Strategy, e.g., "auto (50-80% CPU)"]
  Access: [Restrictions]

```

9 Infrastructure as Code Examples

9.1 Terraform Example

```

# Terraform example for core infrastructure
provider "aws" {
  region = "us-west-2"
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"

  tags = {
    Name = "main-vpc"
    Environment = "production"
  }
}

resource "aws_subnet" "public" {
  vpc_id      = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"

  tags = {
    Name = "public-subnet"
  }
}

```

```

resource "aws_ecs_cluster" "main" {
  name = "service-cluster"
  capacity_providers = ["FARGATE"]
}

resource "aws_ecs_task_definition" "app" {
  family = "service"
  requires_compatibilities = ["FARGATE"]
  cpu = 256
  memory = 512

  container_definitions = jsonencode([
    {
      name = "app"
      image = "app:latest"
      essential = true
      portMappings = [
        {
          containerPort = 8080
          hostPort = 8080
        }
      ]
    }
  ])
}

```

9.2 Kubernetes Example

```

# Example Kubernetes deployment
apiVersion: apps/v1
kind: Deployment
metadata:
  name: app-deployment
  labels:
    app: my-app
spec:
  replicas: 3
  selector:
    matchLabels:

```

```

    app: my-app
template:
  metadata:
    labels:
      app: my-app
  spec:
    containers:
      - name: app
        image: my-app:latest
        ports:
          - containerPort: 8080
        resources:
          limits:
            cpu: "0.5"
            memory: "512Mi"
          requests:
            cpu: "0.2"
            memory: "256Mi"
        livenessProbe:
          httpGet:
            path: /health
            port: 8080
          initialDelaySeconds: 30
          periodSeconds: 10

```

10 Operational Considerations

```

Monitoring:
  Tools: [List of monitoring tools]
  Key Metrics:
    - [Metric 1]: [Description and thresholds]
    - [Metric 2]: [Description and thresholds]
  Alerts:
    - [Alert condition 1]: [Response procedure]
    - [Alert condition 2]: [Response procedure]
Logging:

```

```

Strategy: [Centralized/distributed]
Retention: [Policy]
Key Logs:
  - [Log type 1]: [Purpose and location]
  - [Log type 2]: [Purpose and location]

Backup & Recovery:
  Data Backup:
    - [Database]: [Strategy and schedule]
    - [File storage]: [Strategy and schedule]
  System Backup:
    - [Configuration]: [Approach]
    - [Images]: [Approach]
  Disaster Recovery:
    - [Scenario 1]: [Recovery process]
    - [Scenario 2]: [Recovery process]

```

11 Security Controls

```

Network Security:
  - Perimeter: [Firewalls, WAF, etc.]
  - Segmentation: [Network isolation approach]
    - Encryption: [TLS versions, certificates]

Access Control:
  - Identity: [IAM approach]
  - Authentication: [Methods]
  - Authorization: [Policies]
  - Least Privilege: [Implementation]

Compliance:
  - Regulations: [Applicable requirements]
  - Auditing: [Approach]
  - Evidence: [Collection method]

```

12 Cost Optimization

Resource Efficiency:

- Right-sizing: [Approach]
- Auto-scaling: [Strategy]
- Reserved Instances: [Strategy]

Cost Allocation:

- Tagging: [Strategy]
- Chargeback: [Method]
- Monitoring: [Tools]

Optimization Strategies:

- Storage Tiering: [Approach]
- Spot Instances: [Usage]
- Resource Scheduling: [Off-hours strategy]

13 Template Validation Checklist

- ☐ Infrastructure strategy defined
- ☐ Deployment architecture documented
- ☐ Environment configurations specified
- ☐ IaC examples provided
- ☐ Network architecture described
- ☐ Security controls identified
- ☐ Operational considerations addressed
- ☐ Cost optimization strategy included
- ☐ Team review completed

14 Related Views

- System Context - System boundaries
- Component Architecture - Component details
- Container Architecture - Container strategy