M11-02 · GIBALJA VISUAL MANUAL

도메인·HTTPS·클라우드로 배포하기

한 문장 목표: 실제 domain 구매·DNS 변경·certificate 발급·cloud account·외부 network·live 배포 없이, Browser부터 managed data까지 공개 요청 경로를 설계하고 24개 합성 시나리오로 `public_release_ready` 를 판정합니다.

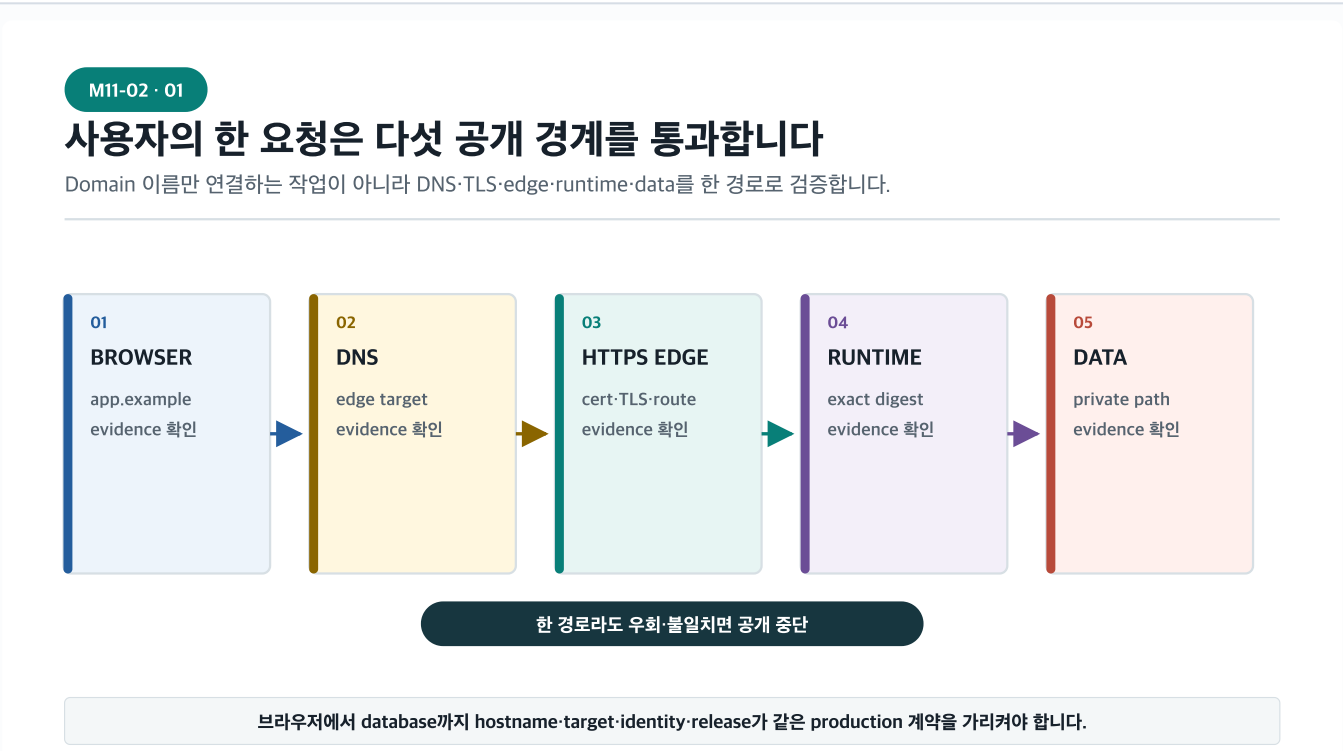
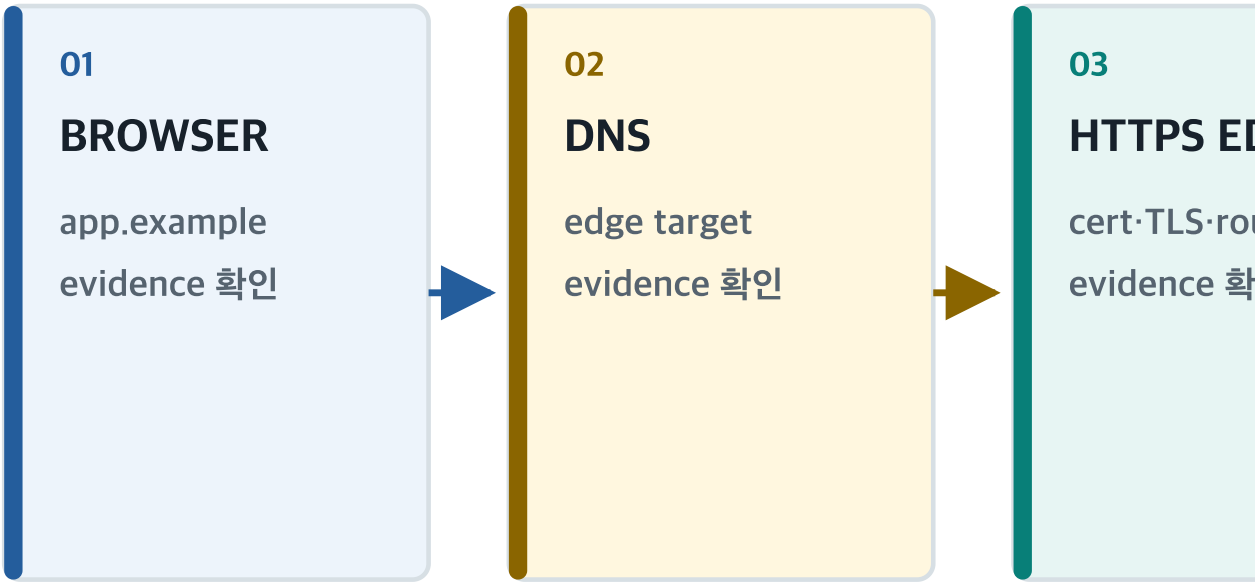


그림 1. 사용자가 보는 것은 URL 하나지만 실제 요청은 DNS·TLS·edge·runtime·data의 다섯 경계를 통과합니다. 한 경계라도 다른 environment나 승인하지 않은 target을 가리키면 공개를 멈춥니다.

Domain 이름만 연결하는 작업이 아니라 DNS·TLS·edge·runtime·data



를 한 경로로 검증합니다.

불일치면 공개 중단

0. 한눈에 보는 두 번째 지도

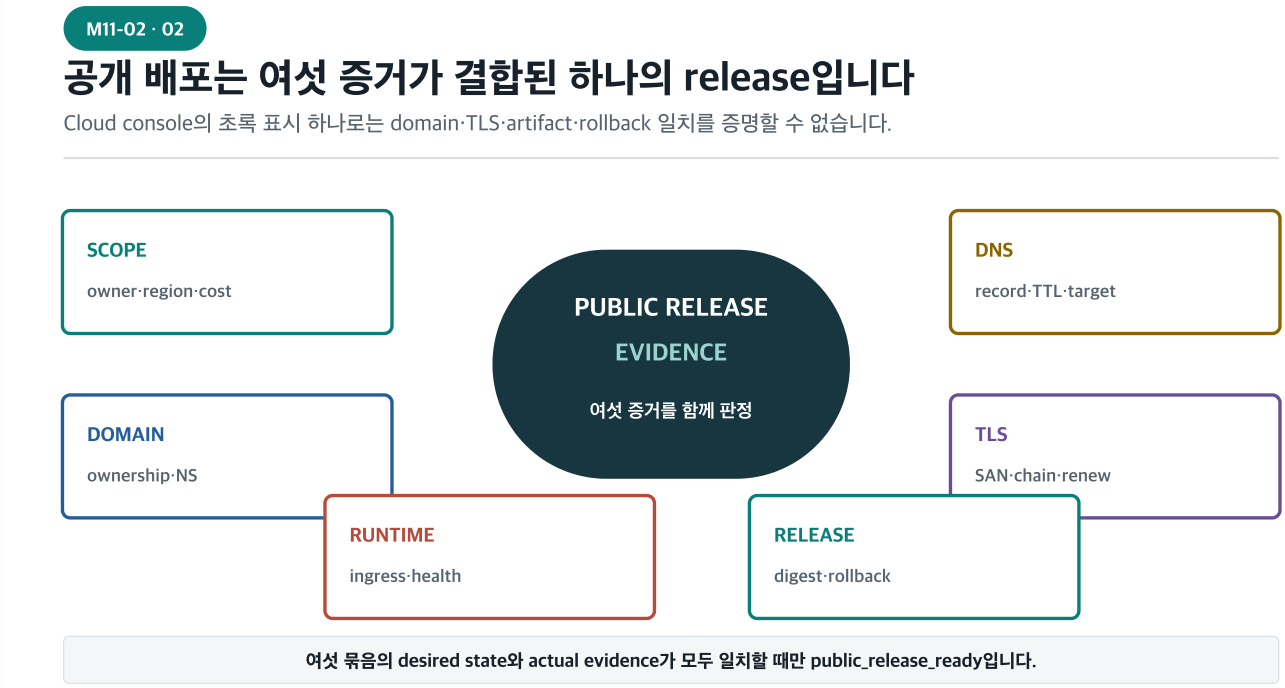


그림 2. Cloud console의 초록 아이콘 하나가 공개 가능성을 증명하지 않습니다. 범위·domain·DNS·TLS·runtime·release evidence를 하나의 exact release로 묶어 판정합니다.

Cloud console의 초록 표시 하나로는 domain·TLS·artifact·rollback 을

SCOPE

owner·region·cost

DOMAIN

ownership·NS

RUNTIME

ingress·health

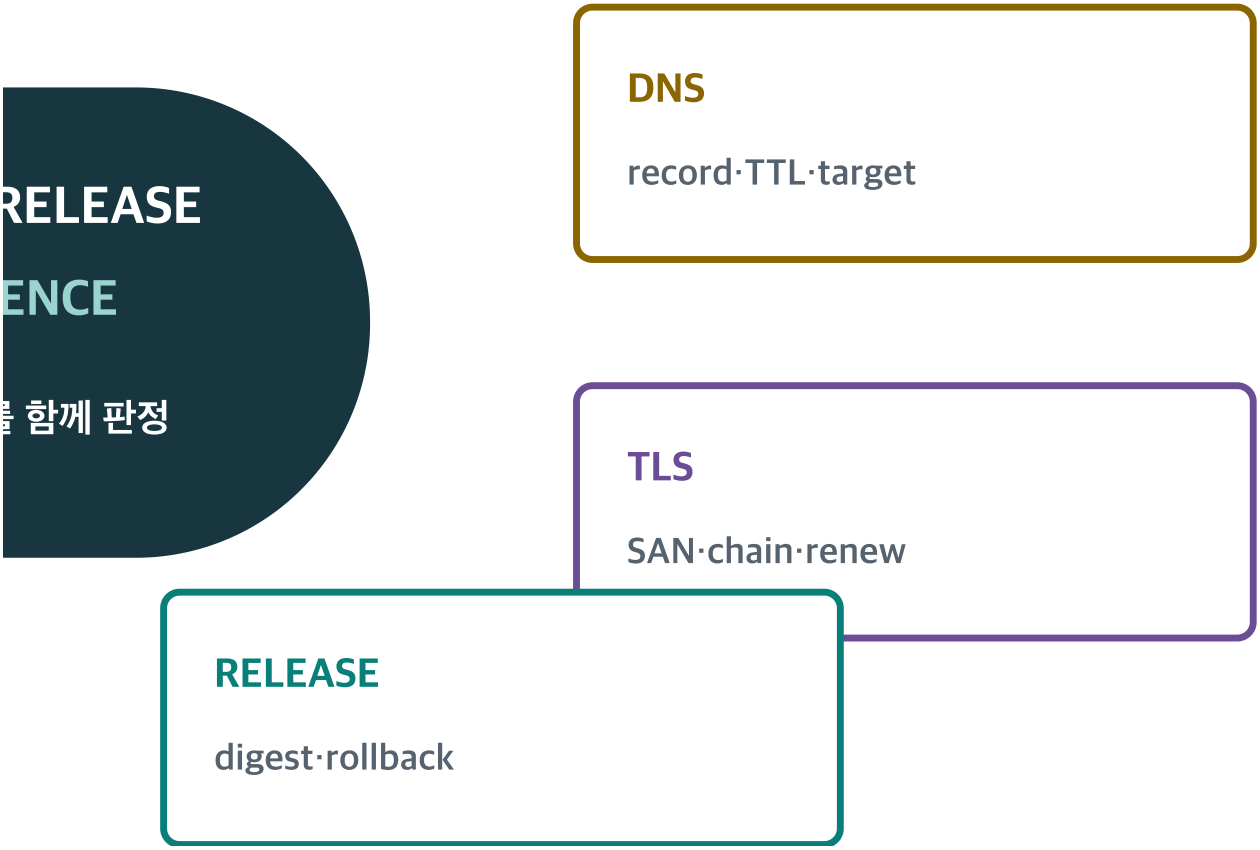
PUBLIC

EVID

여섯 증거를

Release입 수

일치를 증명할 수 없습니다.



난이도	그림 먼저	개념·판정	실습	셀프 테스트	최종 산출물
Level 2	35분	70분	90분	15분	공개 경로·24개 결과·Public Release Gate

이 교재는 provider console 사용법을 외우는 안내서가 아닙니다. Domain·DNS·certificate·cloud product는 account·region·plan과 시점에 따라 기능이 달라집니다. 따라서 provider-neutral한 통제와 evidence를 먼저 익힌 뒤, 실제 작업 직전 해당 provider의 최신 공식 문서와 조직 절차를 확인합니다. 이 교재와 실습은 실제 cloud 구성·TLS 보안 감사·침투 테스트·개인정보 영향평가·운영 승인을 대신하지 않습니다.

0.1 첫 두 장에서 기억할 열두 문장

```
Domain은 소유권이고 DNS는 그 이름의 해석 경로다.
Parent delegation과 authoritative record는 서로 다른 evidence다.
DNS 변경은 즉시 전체 client에 보이지 않는다.
Certificate 발급 성공과 다음 자동 갱신 성공은 다른 문제다.
HTTPS edge가 있어도 public origin이 남으면 우회할 수 있다.
HSTS는 모든 hostname이 준비된 뒤 짧은 기간부터 적용한다.
Cloud를 쓰면 책임이 사라지는 것이 아니라 provider와 나뉜다.
Managed database도 private path·권한·migration을 대신 결정하지 않는다.
Production target은 account·project·region·resource ID로 고정한다.
시험한 exact digest와 배포한 exact digest가 같아야 한다.
Traffic은 health evidence를 보며 단계적으로 올린다.
Rollback은 DNS·traffic·release 세 축을 함께 복구한다.
```

1. 이 PDF를 공부하는 방법

1.1 1회차 · 그림 16장만 읽기 · 35분

그림 제목과 caption만 읽고 다음 흐름을 소리 내어 설명합니다.

```
사용자 요청
→ 공개 release evidence
→ 공동 책임·runtime 선택
→ public edge·private runtime·managed data
→ domain·delegation·record·TTL
→ certificate·TLS·origin
→ identity·secret·migration
→ canary·health·rollback
→ 24개 scenario·Public Release Gate
```

각 그림에서 다음 한 문장을 완성합니다.

> 이 단계의 desired state는 _____이고, actual evidence는 _____에서 확인하며, 다르면 _____한다.

1.2 2회차 · 공개 배포 검수 스튜디오 · 55분

실습 생성기를 실행합니다.

```
./02_Labs/G11_Deployment_Operations/L11-02_create-public-cloud-deployment-practice.sh
```

생성된 evidence에서 세 version을 비교합니다.

Version	Pass	Decision
`domain-first-v1`	6/24	`blocked_public_exposure`
`managed-edge-v2`	16/24	`blocked_domain_release_drift`
`verified-public-release-v3`	24/24	`public_release_ready`

1.3 3회차 · 내 서비스 결과서 작성 · 120분

공개 배포 설계·검수 결과서를 복제해 다음 순서로 채웁니다.

1. 한 장 요청 경로와 공개 표면
2. 공동 책임·account·region·resource inventory
3. Domain ownership·delegation·DNS record·cutover
4. Certificate·TLS·edge-origin
5. Runtime·identity·secret·managed data
6. Exact release·IaC·migration
7. Progressive traffic·health·rollback
8. 12 control·24 scenario·잔여 위험·Public Release Gate

1.4 읽다가 막힐 때

Domain·HTTPS·Cloud 배포 용어집 300에서 지금 단계의 20개 묶음만 읽습니다. 모든 용어를 먼저 외우지 않습니다.

2. “배포됐다”를 다시 정의하기

2.1 네 문장은 서로 다르다

문장	실제 의미	아직 모르는 것
Build가 성공했다.	Artifact를 만들었다.	실행·network·health·domain
Cloud deploy가 성공했다.	Provider가 resource 변경을 받아들였다.	Public path·기능·data
Default URL이 응답한다.	Runtime 입구 하나가 응답한다.	Custom domain·TLS·edge
Public hostname이 healthy다.	실제 사용자 경로가 기준을 통과한다.	지속 관찰·incident readiness

PUBLIC DEPLOYMENT “배포 완료”는 command exit code가 아니라

정확한 hostname이 정확한 edge와 release를 거쳐 정상 기능을 제공하고, 실패하면 복구할 수 있다는 evidence 상태

입니다.

2.2 Public path의 최소 계약

경계	Desired state	Evidence	실패 행동
Domain	조직 소유·owner·갱신	Registrar state	공개 중단
Delegation	Parent NS = authoritative	NS answer	DNS 변경 중단
DNS	승인 record·TTL·target	Authoritative+resolver	Cutover 중단
Certificate	Hostname·chain·기간·renew	TLS handshake·manager	Traffic 중단
Edge	HTTPS·host routing·origin 보호	Edge config·public check	Route 차단
Runtime	Exact digest·config·health	Runtime state	Traffic 제거
Data	Private path·least privilege	Network·IAM·query smoke	Release 중단
Release	Canary·health·rollback	Deployment event·metrics	Rollback

2.3 네 가지 흔한 오해

오해 1 · Domain을 샀으니 배포가 끝났다.
Domain은 이름의 등록 권리입니다. 어떤 DNS가 답하고, 그 record가 어떤 edge를 가리키며, 그 edge가 어떤 runtime에 전달하는지는 별도입니다.

오해 2 · 자물쇠가 보이니 안전하다.
Browser 자물쇠는 현재 connection의 certificate와 encryption에 관한 신호입니다. Origin 우회·과권한·취약한 application·잘못된 data target·갱신 실패까지 증명하지 않습니다.

오해 3 · Managed service니까 provider가 다 책임진다.
Provider가 platform을 관리해도 account·identity·data·domain·DNS·release·monitoring 선택은 service team 책임으로 남습니다.

오해 4 · Rollback은 이전 code를 재배포하면 된다.
DNS와 traffic이 새 edge를 계속 가리키거나 schema가 이전 code와 호환되지 않으면 복구가 끝나지 않습니다.

3. Cloud 공동 책임과 Runtime 선택

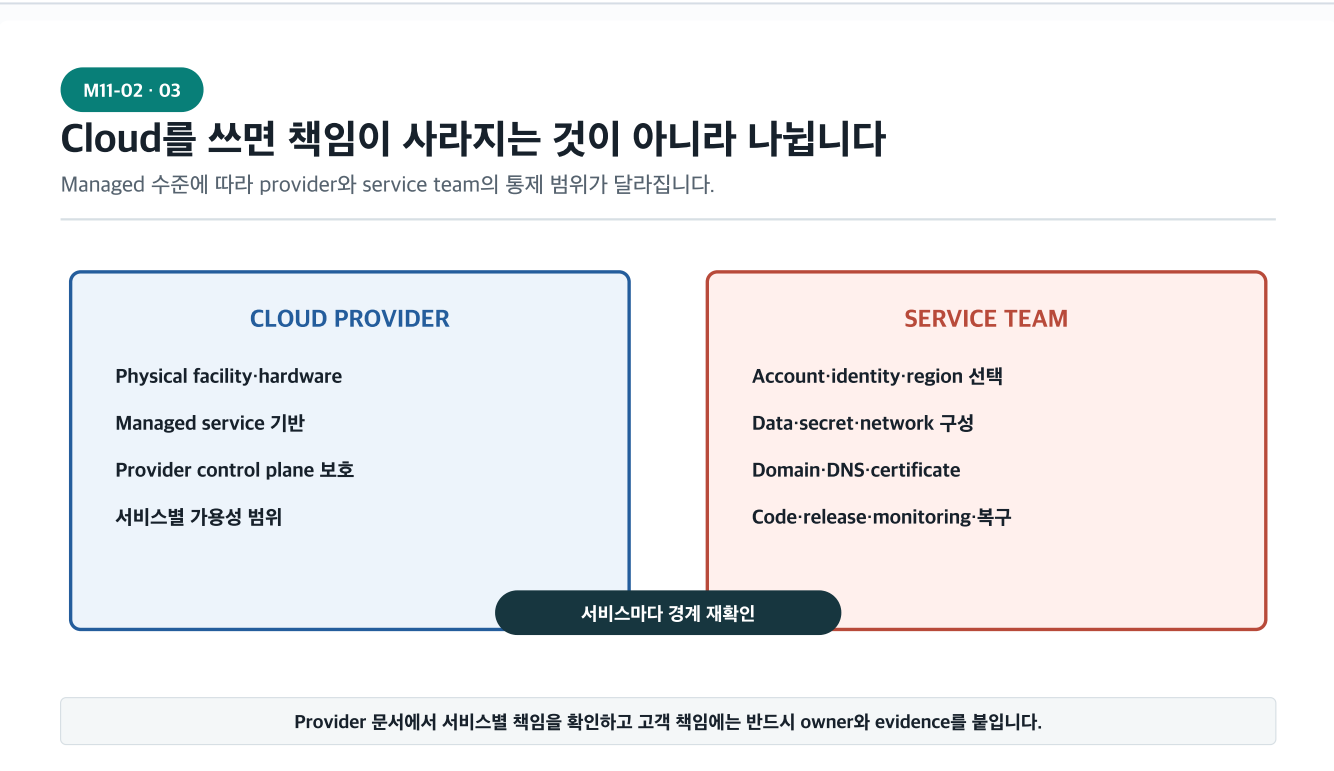


그림 3. Managed 범위가 커질수록 팀의 infrastructure 작업은 줄 수 있지만 account·identity·application·data·domain·release 결과에 대한 책임은 남습니다.

Cloud를 쓴다는 게 무엇을 뜻하는가

Managed 수준에 따라 provider와 service team의 통제 범위가 달라짐

CLOUD PROVIDER

Physical facility·hardware

Managed service 기반

Provider control plane 보호

서비스별 가용성 범위

서비스마다

니다.

SERVICE TEAM

Account·identity·region 선택

Data·secret·network 구성

Domain·DNS·certificate

Code·release·monitoring·복구

경계 재확인

3.1 책임 표를 먼저 쓰는 이유

“Cloud가 알아서 한다”는 문장은 owner를 지웁니다. 공개 전에 다음 형식으로 책임을 분해합니다.

Layer	Provider가 관리	팀이 관리	반드시 확인할 질문
Facility·hardware	대체로 provider	Provider 선택	어떤 장애 범위를 보장하는가.
Managed platform	Service별 다름	Version·setting	Patch·runtime lifecycle은 어디까지인가.
Account·IAM	Mechanism 제공	구조·role·binding	누가 production을 바꿀 수 있는가.
Network	Service 제공	Public·private policy	Origin·DB가 직접 공개되는가.
Domain·DNS	Tool 제공 가능	소유권·record·target	갱신과 rollback owner는 누구인가.
Certificate	자동화 가능	Hostname·validation·renewal	실패를 누가 언제 아는가.
Application·data	Runtime 제공	Code·schema·permission	어떤 release와 data가 연결됐는가.
Monitoring·incident	Signal 제공	기준·routing·대응	실제 public path를 보는가.

공식 문서의 “provider responsibility”와 “customer responsibility”를 그대로 복사하지 말고, 우리 resource·owner·evidence에 연결합니다.

3.2 Runtime 선택

M11-02 · 04

Runtime은 유행보다 팀이 감당할 운영 범위로 고릅니다

첫 배포는 관리형 서비스가 유리하지만 network·protocol·region 요구가 크면 선택이 달라집니다.

선택지	운영 부담	제어 범위	확장	적합한 시작
Managed app	낮음	중간	자동	첫 공개 서비스
Container runtime	중간	중간~높음	자동	Container 보유
Kubernetes	높음	높음	정교	복잡한 platform
Virtual machine	높음	매우 높음	직접	특수 runtime

같은 container라도 provider·region·plan별 domain·certificate·network 기능은 현재 공식 문서로 재확인합니다.

그림 4. 첫 공개 서비스는 팀이 감당할 운영 범위가 기준입니다. 더 많은 제어는 더 많은 patch·network·capacity·incident 책임을 뜻합니다.

Runtime은 배포는 쉽지만 network·protocol·region 요구

첫 배포는 관리형 서비스가 유리하지만 network·protocol·region 요구

선택지	운영 부담	제어
Managed app	낮음	제
Container runtime	중간	중간
Kubernetes	높음	높
Virtual machine	높음	매우

가 크면 선택이 달라집니다.

범위	확장	적합한 시작
간	자동	첫 공개 서비스
~높음	자동	Container 보유
음	정교	복잡한 platform
높음	직접	특수 runtime

질문	Managed app	Container runtime	Kubernetes	VM
Server patch 부담	낮음	낮음~중간	중간~높음	높음
Custom network 제어	중간	중간	높음	높음
Scaling 기본 기능	강함	강함	정교	직접 구성
초기 학습 난이도	낮음	중간	높음	중간~높음
특수 system dependency	제한	Container 범위	넓음	매우 넓음
적합한 출발	일반 web app	이미 image 보유	Platform 운영 역량	특수 legacy·kernel 요구

3.3 선택 기록

Runtime을 고를 때 제품 이름 대신 다음 decision record를 씁니다.

```
Workload shape:
  HTTP request / background / event / long-running

Runtime requirement:
  language version / port / filesystem / CPU / memory / GPU

Network requirement:
  public edge / private origin / managed DB / outbound allowlist

Operations capacity:
  patch owner / scaling owner / on-call / cost owner

Decision:
  selected option / rejected options / review date / current provider docs
```

3.4 Provider 문서는 현재 상태로 확인한다

Custom domain 연결 방식은 product·region·plan에 따라 다릅니다. 2026. 7. 16. 검토 시점의 예:

- AWS App Runner 문서는 custom domain의 certificate validation record를 자동 갱신을 위해 유지하도록 안내합니다.
- Azure Container Apps 문서는 apex에 A record, subdomain에 CNAME 등 domain 유형별 DNS 요구와 managed·BYO certificate 흐름을 설명합니다.
- Google Cloud Run 문서는 direct domain mapping의 제한과 preview 상태를 밝히며 production에는 global external Application Load Balancer 등 권장 경로를 확인하도록 합니다.

이 예를 provider 비교표로 암기하지 않습니다. 실제 작업 시 선택한 service의 최신 문서·region·plan을 다시 확인합니다.

4. Production Cloud 경계

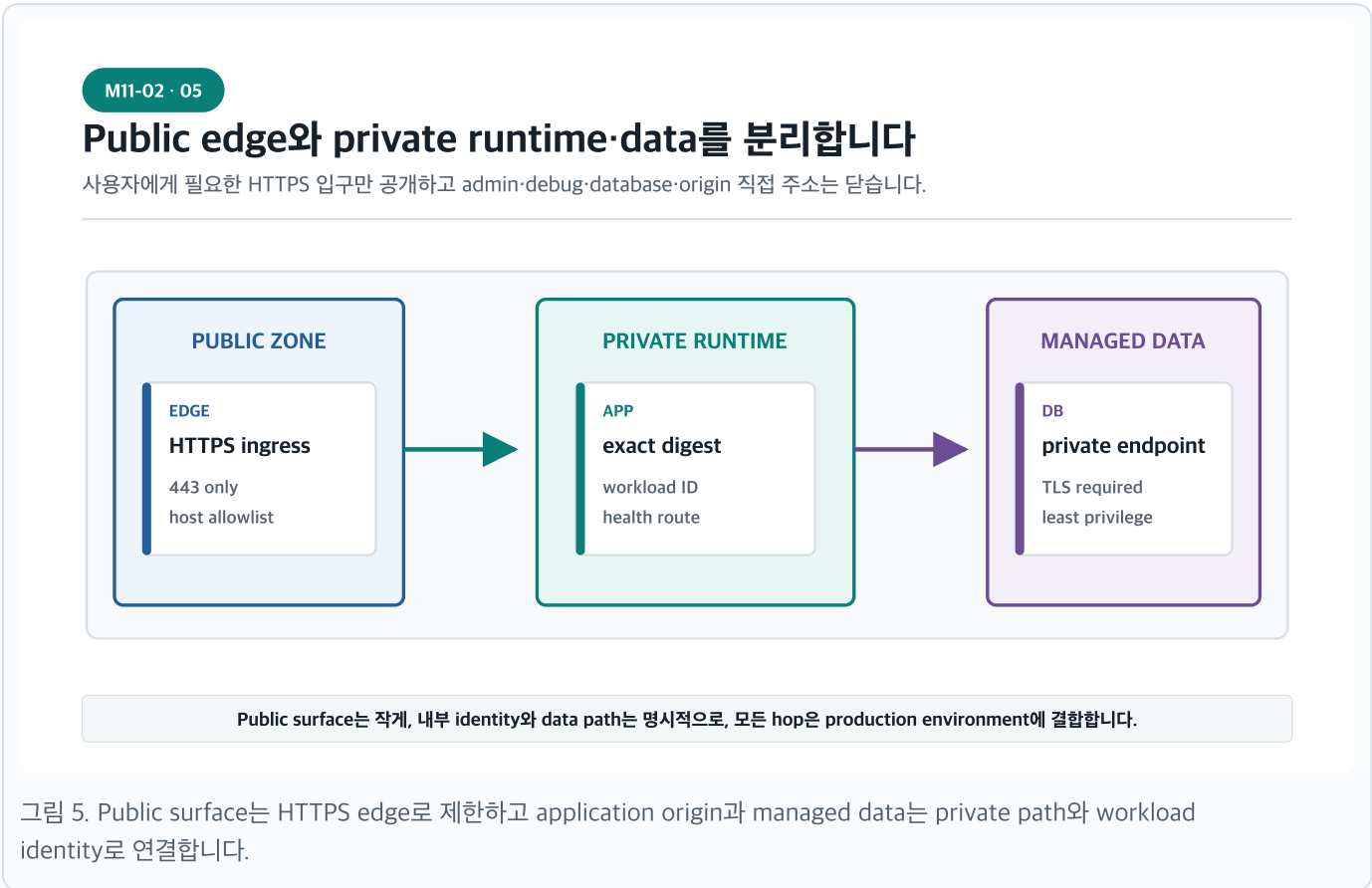
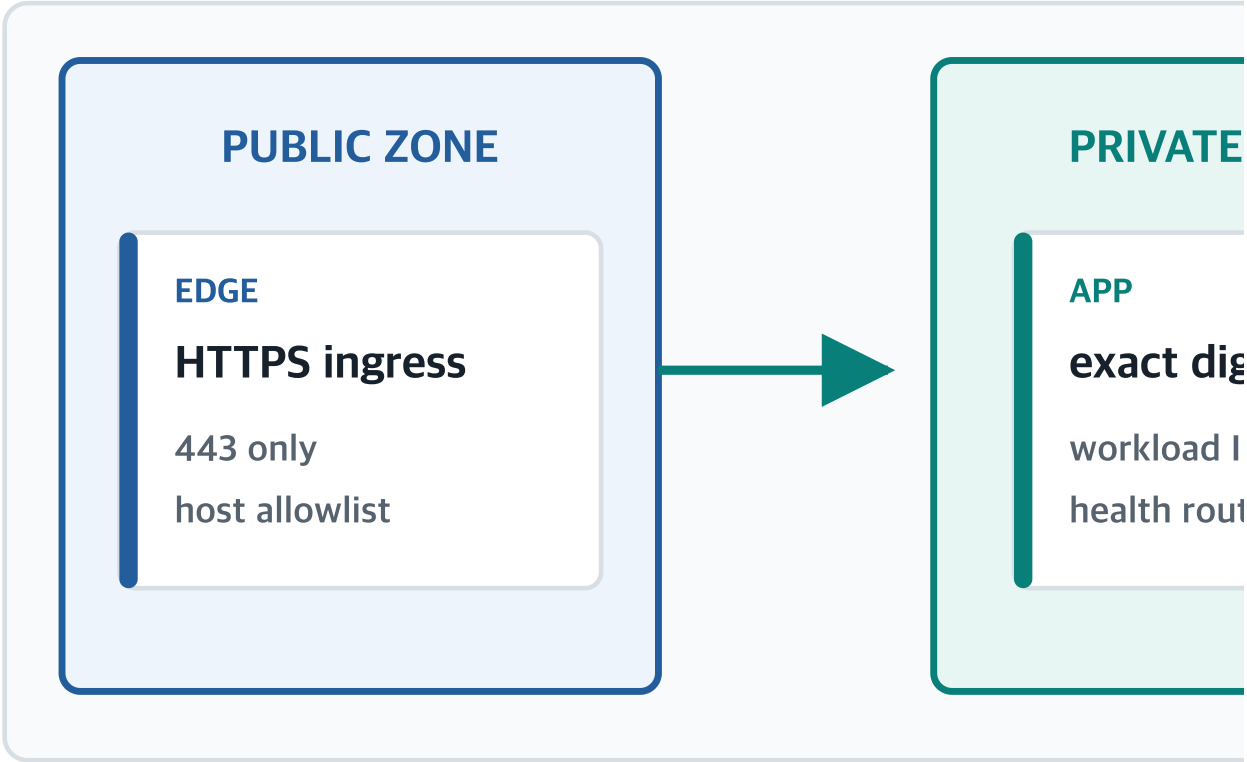


그림 5. Public surface는 HTTPS edge로 제한하고 application origin과 managed data는 private path와 workload identity로 연결합니다.

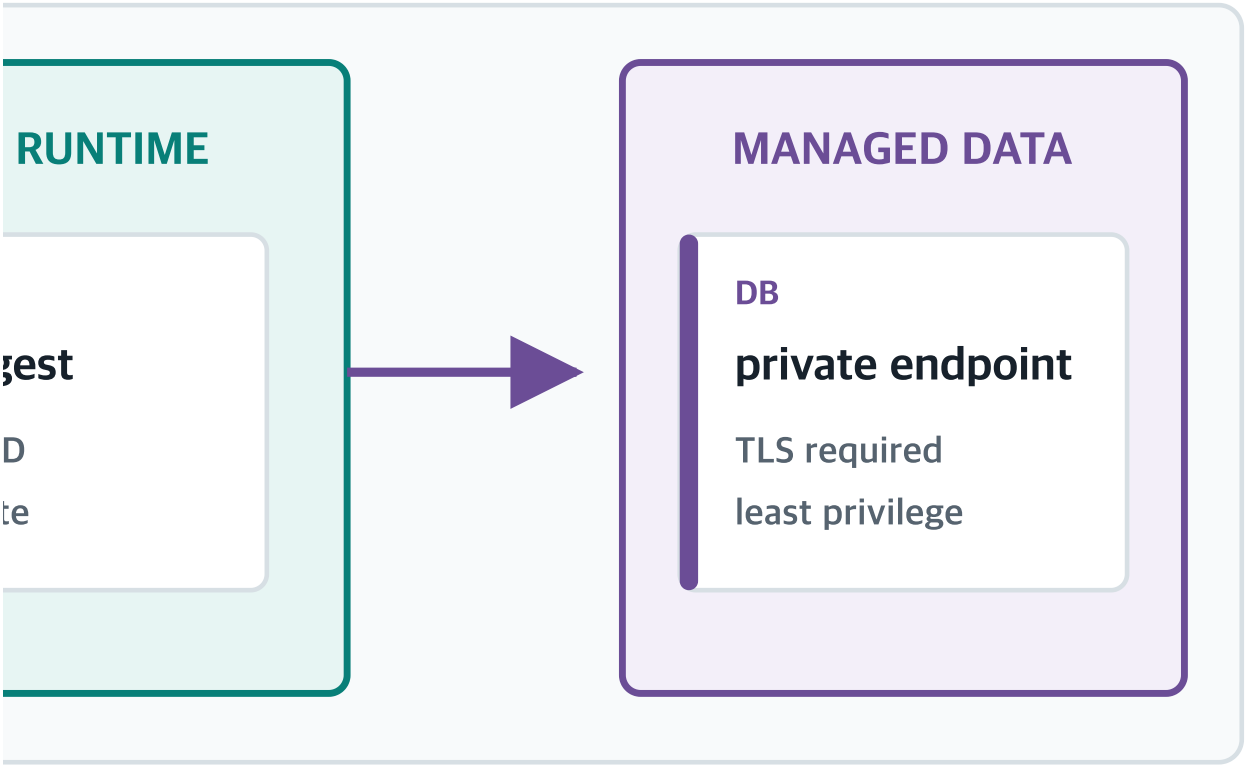
Public edge와 private runtime 분

사용자에게 필요한 HTTPS 입구만 공개하고 admin·debug·database·c



[illegible]

origin 직접 주소는 닫습니다.



4.1 Account·project·region부터 고정한다

잘못된 target 배포는 code 결함이 아니라 경계 결함입니다. 다음 값을 배포 시작 전에 기계적으로 비교합니다.

Field	Expected 예	Mismatch 행동
Organization	`org-prod`	Block
Account·project	`service-prod`	Block
Region	승인 region	Block
Environment	`prod`	Block
Resource ID	승인 exact ID	Block
Artifact digest	승인 digest	Block
Config version	승인 version	Block
Approval ID	유효 approval	Block

화면 색·resource 이름 접두사만 믿지 않습니다. API가 반환한 immutable ID와 desired state를 비교합니다.

4.2 Resource inventory

Public deployment는 application 하나가 아니라 resource graph를 바꿉니다.

```
account
├── public edge
├── runtime service
├── workload identity
├── secret references
├── managed database
├── network policy
├── log·metric destination
├── DNS·certificate binding
└── budget·quota·owner tags
```

모든 resource에 최소 다음 metadata를 붙입니다.

- Service
- Environment
- Owner

- Region
- Data class
- Cost center
- IaC source
- Lifecycle

4.3 Ingress는 좁게

허용 후보:

- Public HTTPS 443
- HTTP 80의 명시적 redirect 또는 reject policy
- Edge에서 origin으로 가는 private·authenticated path
- Runtime에서 managed database로 가는 최소 port·identity

차단 후보:

- SSH·RDP·admin console의 public ingress
- Debug port
- Database public ingress
- Unauthenticated origin URL
- 모든 source를 허용하는 wildcard rule

4.4 Health는 세 층으로 나눈다

Probe	질문	실패 행동
Startup	Process가 초기화됐는가.	배포 중단
Readiness	새 request를 받을 준비가 됐는가.	Traffic에서 제거
Liveness	스스로 회복할 수 없는가.	제한된 restart
Public path	실제 hostname·TLS·edge·route가 정상인가.	Traffic 중단·rollback

Liveness에서 database처럼 외부 dependency를 강하게 검사하면 dependency 장애 때 모든 instance가 재시작되어 상황을 악화할 수 있습니다. 각 probe의 목적을 분리합니다.

4.5 Quota와 budget도 release 조건이다

기능이 정상이어도 다음은 장애가 됩니다.

- Region quota 부족으로 instance 생성 실패
- IP·certificate·load balancer limit 초과
- Scale-out 상한이 traffic보다 낮음
- Minimum instance 0으로 latency 급증
- 잘못된 loop나 공격으로 비용 급증

그래서 quota evidence·budget alert·cost owner를 `CTRL-01·02`에 포함합니다.

---

5. Domain과 DNS를 분리해서 이해하기

5.1 Domain은 소유권, DNS는 해석

개념	답하는 질문
Domain registration	이 이름을 사용할 권리는 누구에게 있는가.
Registrar	등록·갱신·transfer는 어디서 관리하는가.
Delegation	Parent는 어느 nameserver에게 답을 맡겼는가.
Authoritative zone	원본 record는 어디서 관리하는가.
Resolver	사용자를 대신해 어떤 답을 찾고 cache하는가.
Resource record	Hostname은 어떤 target·policy를 가리키는가.

5.2 Domain 소유권 checklist

- [ ] Organization 소유 계정
- [ ] Registrar MFA
- [ ] Transfer lock
- [ ] 최소 2인 recovery path
- [ ] Expiration·auto-renewal·payment owner
- [ ] Administrative contact 최신화
- [ ] Domain inventory
- [ ] 변경 audit

개인 계정 한 명과 개인 결제 수단에 domain을 묶으면 그 사람이 부재한 순간 service continuity가 흔들립니다.

5.3 DNS resolution 사슬

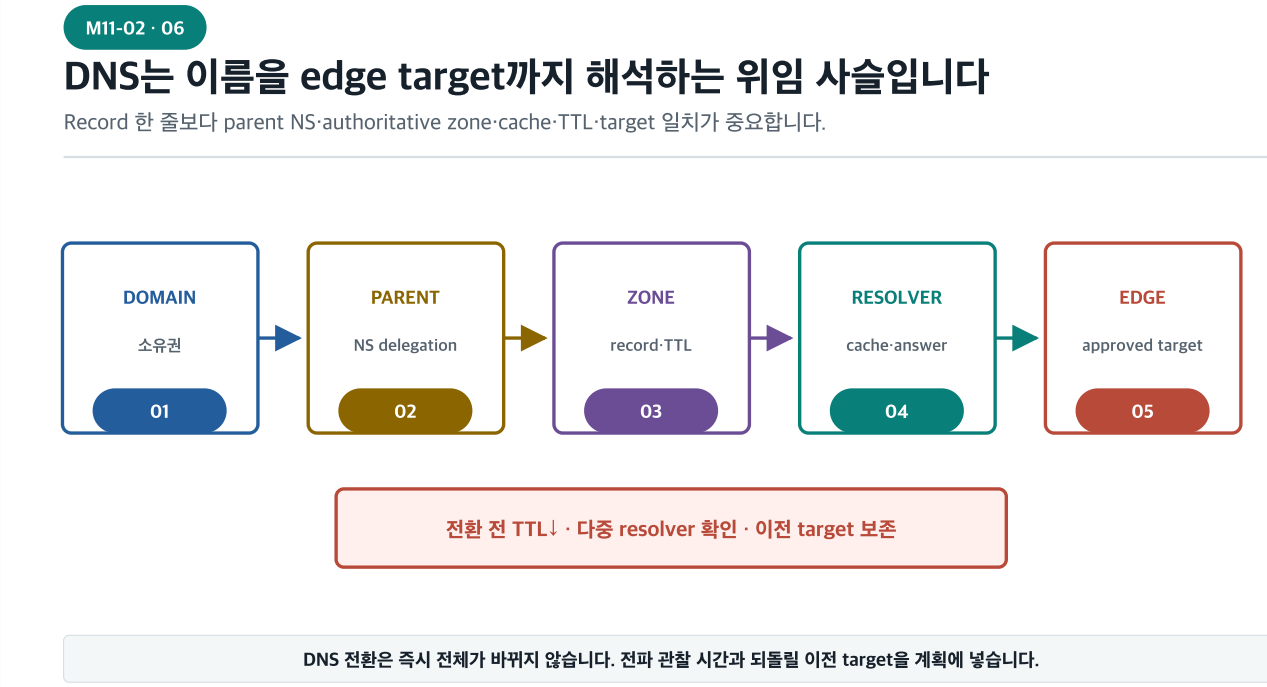
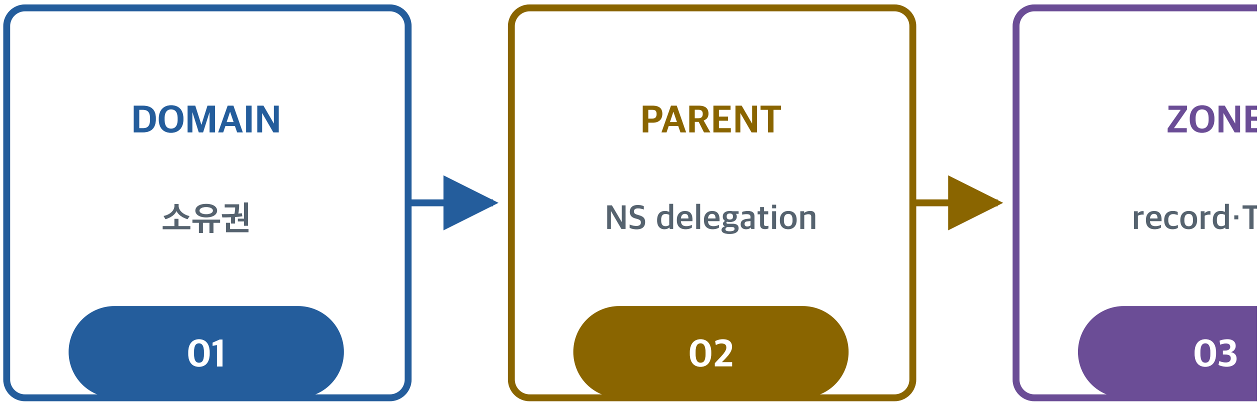


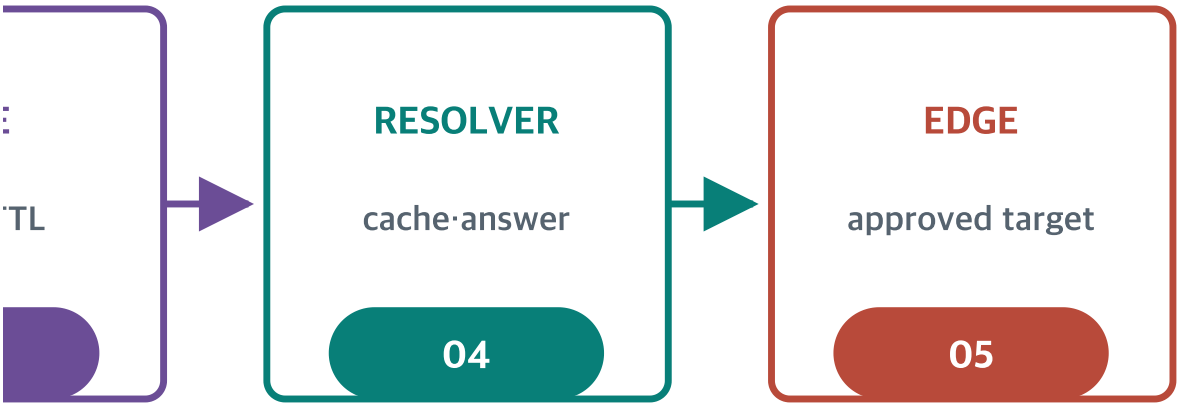
그림 6. DNS record를 바꾸기 전에 parent NS와 authoritative zone이 일치하는지 확인합니다. 전환 뒤에는 한 resolver가 아니라 여러 cache에서 target과 TTL을 관찰합니다.

Record 한 줄보다 parent NS·authoritative zone·cache·TTL·target 읺



전환 전 TTL↓ · 다중 resolv

일치가 중요합니다.



ver 확인 · 이전 target 보존

단순화한 해석:

1. User agent가 stub resolver에 `app.example.test`를 묻습니다.
2. Recursive resolver가 cache를 확인합니다.
3. Cache miss면 root와 TLD를 통해 parent delegation을 찾습니다.
4. Parent의 NS가 authoritative nameserver를 가리킵니다.
5. Authoritative server가 CNAME·A·AAAA 등을 답합니다.
6. Resolver는 TTL 동안 답을 cache합니다.
7. Client가 최종 IP·edge target에 연결합니다.

5.4 “DNS 전파를 기다린다”의 정확한 뜻

Authoritative record 변경은 보통 즉시 반영될 수 있지만, 이미 답을 받은 resolver는 이전 TTL이 끝날 때까지 old answer를 사용할 수 있습니다. 따라서 모든 resolver가 동시에 새 값을 받는 단일 “전파 완료 시각”이 있는 것이 아닙니다.

Cutover 전에:

1. 기존 TTL을 충분히 일찍 낮춥니다.
2. 새 edge와 certificate를 먼저 준비합니다.
3. Previous target을 보존합니다.
4. Authoritative answer와 resolver 여러 곳을 기록합니다.
5. Stop·rollback 조건을 승인합니다.

5.5 Record 선택

그림 7. Subdomain·zone apex·certificate validation은 요구 record가 다릅니다. Provider가 제공한 exact name·type·value와 현재 zone의 충돌을 함께 검사합니다.

Subdomain과 zone apex는 같은 방식으로 연결되지 않을 수 있고 cert

01

SUBDOMAIN

app.example.test

CNAME → provider host

충돌 record=0

02

APEX

example.test

A/AAAA 또는 provid

충돌 record=0

Provider 요구값

:ificate validation record도 별도입니다.

er alias

03

VALIDATION

_acme-challenge

TXT/CNAME → 검증 token

충돌 record=0

값을 그대로 검증

Record	일반 역할	공개 배포 주의
A	Hostname→IPv4	Provider IP lifecycle 확인
AAAA	Hostname→IPv6	IPv6 path·firewall도 검증
CNAME	Hostname→다른 hostname	Apex 제한·다른 record 충돌
Alias·ANAME	Apex→provider target	Provider 비표준 기능·동작 확인
TXT	Domain validation·policy	Token·owner·lifecycle 관리
CAA	허용 CA policy	조직 certificate 정책과 정합
NS	Delegation	Parent·child 일치

5.6 Conflict와 dangling

Conflict: 같은 hostname에 CNAME과 양립할 수 없는 다른 record가 함께 있거나, provider가 기대한 target과 다른 값이 남은 상태입니다.

Dangling: DNS가 삭제된 cloud resource나 더 이상 소유하지 않은 external hostname을 가리키는 상태입니다. 제3자가 그 resource name을 다시 확보할 수 있다면 subdomain takeover 위험으로 이어질 수 있습니다.

삭제 순서:

```
traffic 제거
→ DNS record 제거 또는 안전 target 전환
→ cache·access 확인
→ cloud resource 폐기
→ inventory·certificate 정리
```

5.7 DNS evidence 표

Evidence	반드시 기록할 값
Ownership	Registrar·registrant·owner·expiry
Delegation	Parent NS·authoritative NS
Record	Name·type·TTL·target
Resolver	Resolver·time·answer·observed TTL

Evidence	반드시 기록할 값
Cutover	T-72h·T0·T+5m 등 timeline
Rollback	Previous target·TTL·verify route

6. Certificate·TLS·HTTPS의 수명주기

6.1 Certificate가 증명하는 것

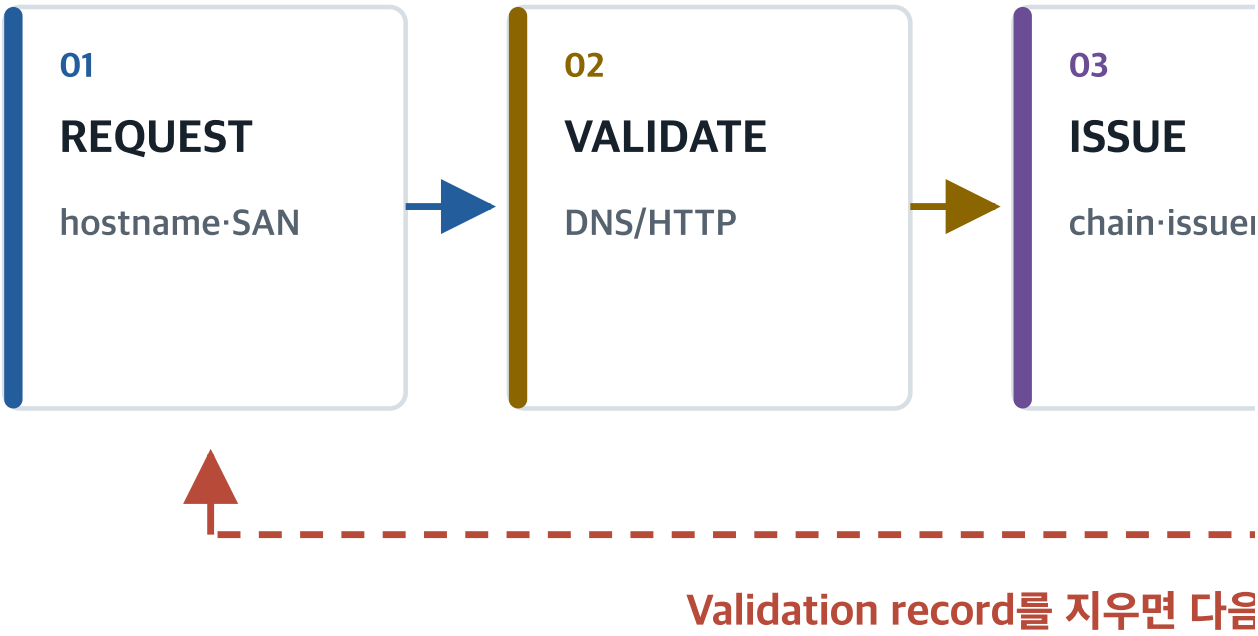
TLS certificate는 server가 해당 hostname의 private key를 가지고 있고, trusted chain이 그 identity binding에 서명했다는 근거를 제공합니다. 다음을 자동 증명하지는 않습니다.

- Application code가 안전함
- Domain account가 침해되지 않음
- Origin이 private임
- User authorization이 올바름
- Database 권한이 최소임
- 다음 renewal이 성공함

6.2 Certificate lifecycle

그림 8. 현재 certificate 설치로 끝나지 않습니다. Validation record·자동 갱신·만료 alert·failure runbook이 다음 주기의 availability를 결정합니다.

Hostname·SAN·chain·기간·issuer를 확인하고 validation record·rene



ewal owner·만료 alert를 유지합니다.

자동 갱신이 실패할 수 있습니다.

Request

- Exact hostname과 SAN 목록
- Wildcard 필요성
- Managed certificate 또는 BYO
- Owner와 endpoint inventory

Validate

- DNS-01·HTTP-01·provider-specific validation
- Validation record의 owner와 TTL
- 자동 갱신 때도 필요한지 확인

Issue·serve

- Hostname match
- Complete chain
- Trusted issuer
- Validity period
- Private key 보호

Renew

- Auto-renew enabled
- Validation path 유지
- Renewal dry run·status
- 30·14·7일 alert
- Failure runbook

6.3 Hostname·SAN·chain·time 네 검사

검사	Expected	실패 증상
Hostname	요청 host가 SAN과 match	Name mismatch
Chain	Intermediate 포함·trusted root 연결	Unknown issuer

검사	Expected	실패 증상
Time	Not before ≤ now < not after	Not yet valid·expired
Endpoint	Edge가 새 certificate serve	Old certificate 지속

Certificate manager 화면과 실제 public handshake를 둘 다 확인합니다. Manager가 “issued”여도 edge binding이 이전 certificate를 serve할 수 있습니다.

6.4 ACME

ACME는 certificate 발급·validation·renewal 자동화를 위한 protocol입니다. 자동화는 반복 작업을 줄이지만 다음 책임을 없애지 않습니다.

- Domain owner와 account recovery
- Challenge record·route의 availability
- Private key·account key 보호
- Renewal failure monitoring
- Endpoint가 새 certificate를 serve하는지 검증

6.5 TLS version과 HTTPS

RFC 8446은 TLS 1.3을 정의합니다. 실무 policy는 client 호환과 조직 기준을 고려해 TLS 1.2와 1.3의 안전한 configuration을 사용하고 더 오래된 protocol은 비활성화하는 방향을 검토합니다. NIST SP 800-52 Rev. 2는 TLS selection과 configuration의 참고 문서지만, NIST는 2026. 5. 7. planning note에서 해당 publication이 review 중이라고 밝혔으므로 최신 상태를 다시 확인합니다.

6.6 HTTPS edge와 private origin

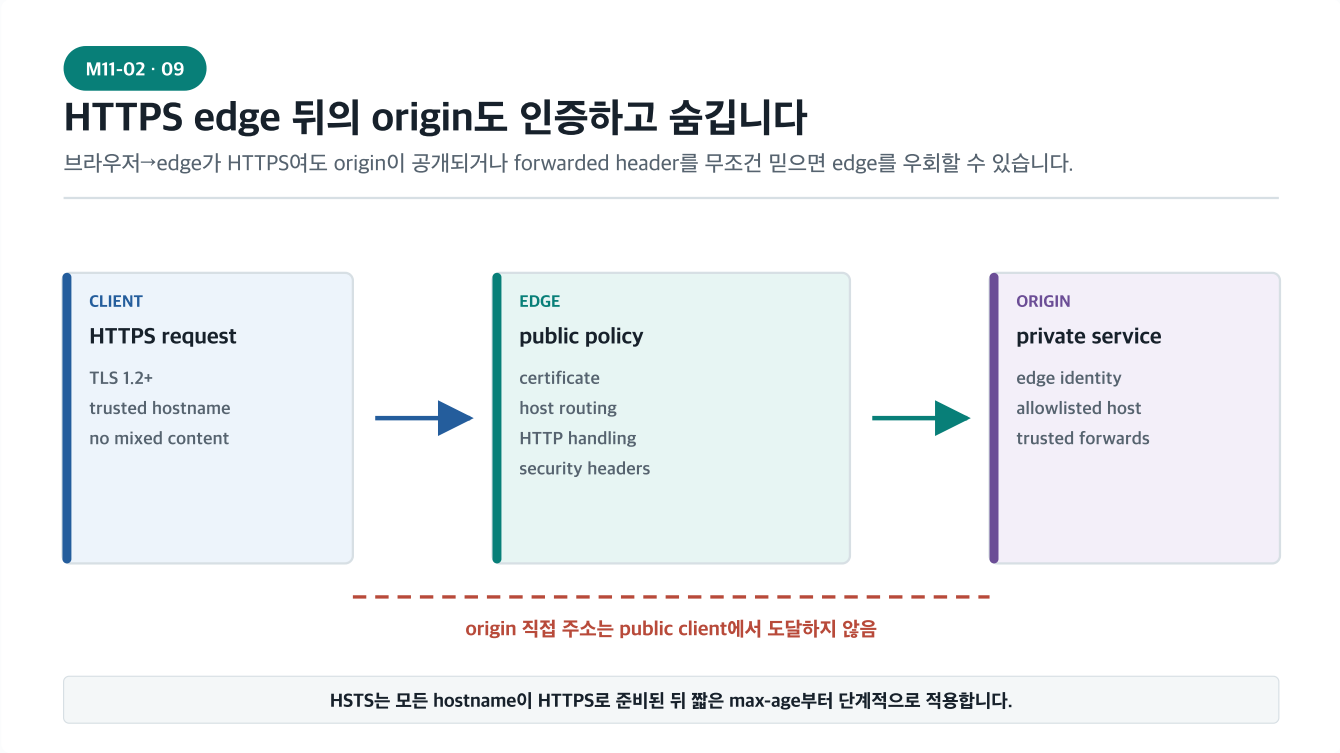


그림 9. Client→edge의 HTTPS만으로 충분하지 않습니다. Origin direct access를 닫고 edge identity·Host allowlist·trusted forwarded header로 edge→origin도 보호합니다.

HTTPS edge 뒤 origin은 공개되거나 forwarded heade

브라우저→edge가 HTTPS여도 origin이 공개되거나 forwarded heade

CLIENT

HTTPS request

TLS 1.2+

trusted hostname

no mixed content

EDGE

public policy

certificate

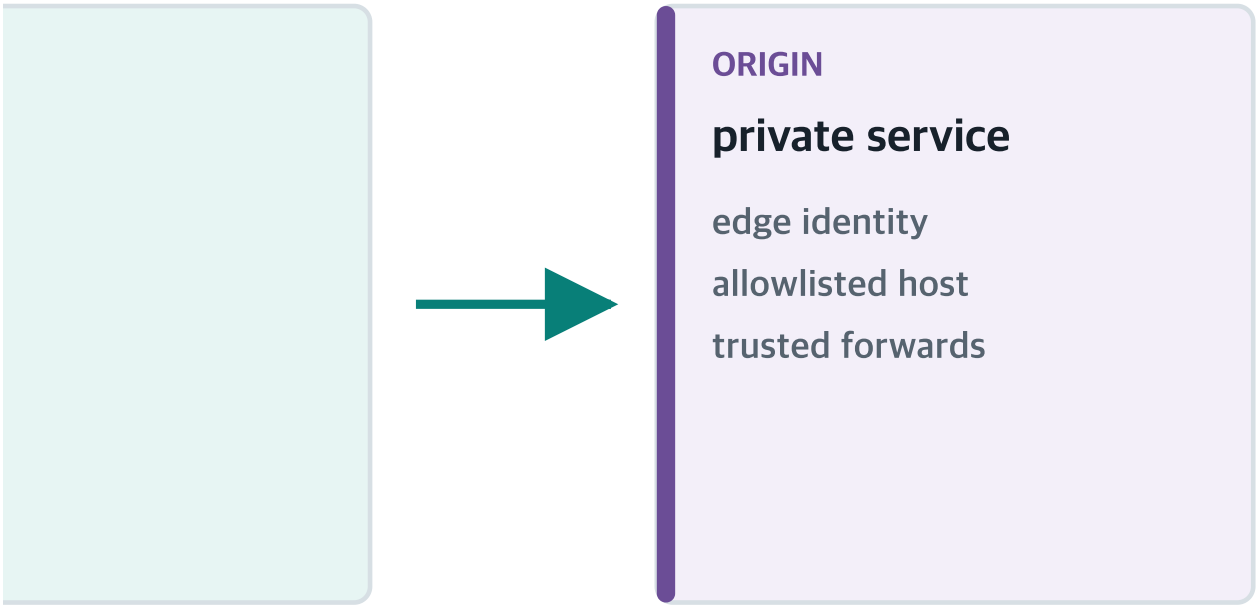
host routing

HTTP handling

security headers

origin 직접 주소는 public

r를 무조건 믿으면 edge를 우회할 수 있습니다.



: client에서 도달하지 않음

Origin bypass가 열리면 공격자는 WAF·rate limit·host routing을 건너뛸 수 있습니다. 다음을 함께 적용합니다.

- Origin private network 또는 firewall source 제한
- Edge-to-origin identity·mTLS·signed request 등
- Exact Host allowlist
- Trusted proxy source에서만 forwarded header 신뢰
- Admin·debug route 미노출
- Alternate provider URL 공개 여부 결정

6.7 HTTP 처리

Web page의 HTTP 요청은 HTTPS로 redirect할 수 있습니다. API와 unsafe method는 client 동작·body 재전송·method 보존을 고려해 명시적으로 redirect 또는 reject합니다. “모두 301” 같은 한 줄 정책은 충분하지 않습니다.

6.8 Mixed content

HTTPS page 안의 HTTP script·image·font·API는 protection을 약화하고 browser가 차단할 수 있습니다.

검사 대상:

- HTML resource URL
- CSS font·image URL
- JavaScript API endpoint
- WebSocket scheme
- Redirect chain
- Download URL

6.9 HSTS는 천천히

HSTS를 받은 browser는 max-age 동안 HTTP를 HTTPS로 바꿉니다. `includeSubDomains`는 모든 하위 host에 영향을 줍니다. `preload`는 browser 사전 목록과 긴 수명주기 때문에 별도 승인이 필요합니다.

권장 학습 순서:

```
모든 hostname HTTPS 준비
→ 짧은 max-age
→ 실제 navigation·asset·API 관찰
→ includeSubDomains 영향 검토
→ 기간 확대
→ preload 별도 결정
```

6.10 Security response headers

Header	핵심 목적	주의
Strict-Transport-Security	HTTPS-only 기억	준비 전 긴 기간 금지
Content-Security-Policy	Resource source 제한	Report·기능 영향 검증
X-Content-Type-Options	MIME sniffing 억제	`nosniff`
Referrer-Policy	Referrer 정보 제한	분석·업무 영향
Set-Cookie Secure	HTTPS에서만 cookie 전송	HttpOnly·SameSite도 별도

Header는 application context에 맞춰 검증합니다. Copy-paste한 강한 policy가 실제 기능을 깨뜨리거나, 너무 느슨해 아무 통제가 아닐 수 있습니다.

---

7. Identity·Config·Secret을 Release에 묶기

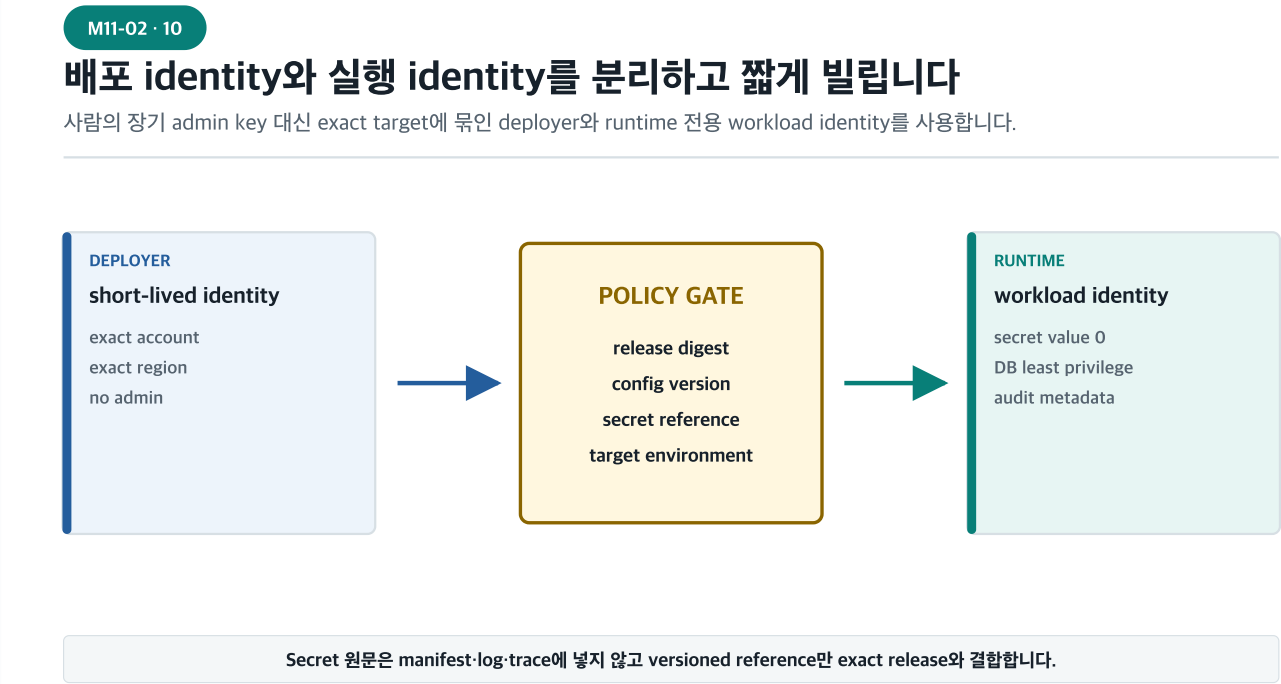


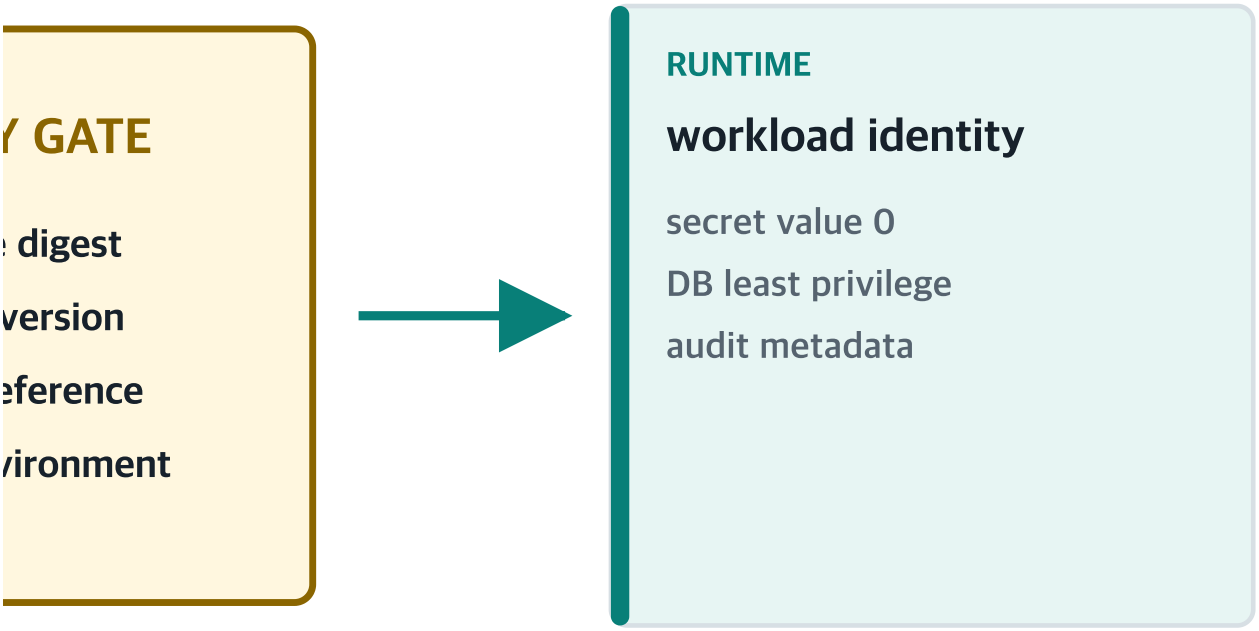
그림 10. 배포 identity와 runtime identity를 분리합니다. Long-lived admin key 대신 exact target에 묶인 short-lived deployer를 쓰고, runtime에는 secret 원문이 아니라 reference를 전달합니다.

[illegible]

사람의 장기 admin key 대신 exact target에 묶인 deployer와 runtime

[illegible]

e 전용 workload identity를 사용합니다.



7.1 세 identity

Identity	용도	허용 범위	금지
Human reviewer	변경 검토·승인	Read·approve	Runtime credential 공유
Deployment identity	Exact release 적용	Exact account·region·resource	Global admin
Workload identity	Runtime의 DB·API 접근	필요한 operation	Deploy·IAM admin

7.2 Short-lived federation

CI가 cloud에 접근할 때 repository에 long-lived access key를 저장하지 않고, 승인 issuer·audience·claim을 검증해 짧은 token으로 교환하는 방식을 우선 검토합니다.

검사:

- Trusted issuer
- Exact audience
- Repository·branch·workflow claim
- Environment claim
- Short expiry
- Exact role
- Audit event

7.3 Secret reference

Release bundle에 넣을 것:

```
secret name
secret version or alias policy
expected environment
consumer identity
rotation owner
```

넣지 않을 것:

```
secret value
private key
access token
database password
validation token
```

7.4 Exact target binding

Approval은 “배포해도 됨”이 아니라 다음 tuple에 결합합니다.

```
artifact digest
+ config version
+ secret references
+ IaC revision
+ migration digest
+ target account·project·region·resource
+ traffic plan
+ rollback target
```

하나라도 바뀌면 approval을 재검토합니다.

---

8. Managed Database와 Migration

8.1 Managed의 의미

Provider가 backup·patch·replication 일부를 제공할 수 있어도 application team은 다음을 결정합니다.

- Public 또는 private connectivity
- Workload identity·database role
- TLS requirement
- Schema design·migration order
- Connection limit·timeout
- Data residency·retention
- Backup·restore 목표와 검증

8.2 Private data path

Public runtime이 반드시 public database를 요구하는 것은 아닙니다. Edge만 public으로 두고 runtime과 database는 private network·private endpoint·service policy로 연결합니다.

Expected:

```
public_database = false
workload_identity_scoped = true
tls_required = true
production_role_only = true
```

8.3 Migration을 release와 호환시키기

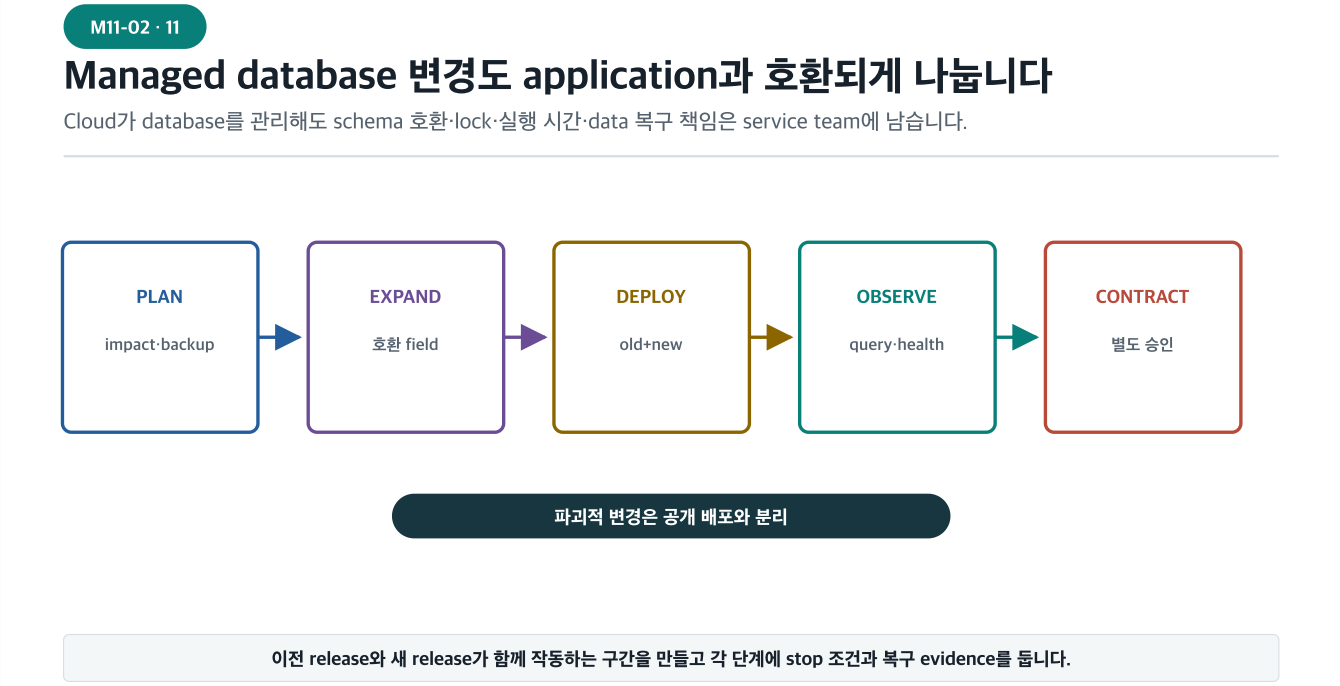
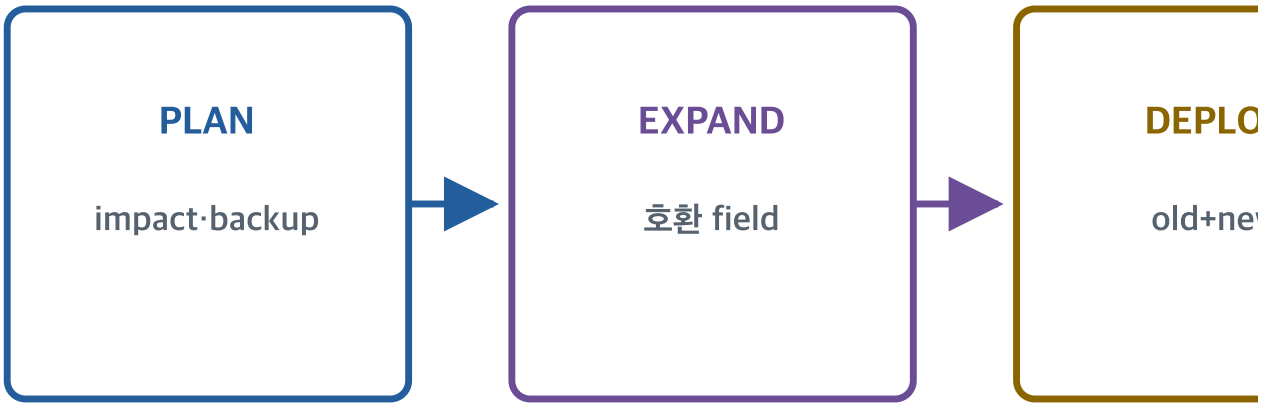


그림 11. Managed database도 application compatibility를 자동 해결하지 않습니다. Expand와 contract를 나누고 이전·새 release가 함께 동작하는 구간을 만듭니다.

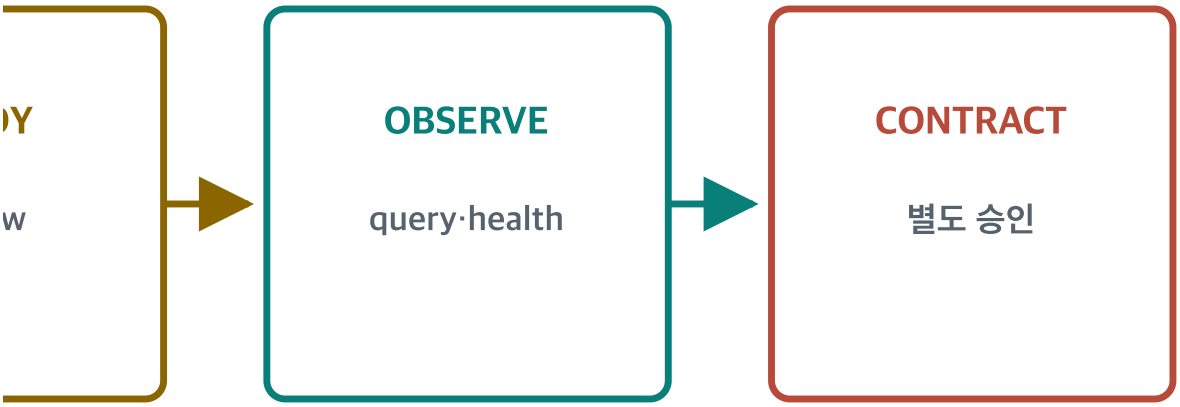
Managed database 변경 application

Cloud가 database를 관리해도 schema 호환·lock·실행 시간·data 복구



파괴적 변경은 금

· 책임은 service team에 남습니다.



공개 배포와 분리

단계	행동	Gate
Plan	영향·lock·duration·backup·owner	Dry run
Expand	호환 field·table·index 추가	Old·new 모두 동작
Deploy	새 code를 canary로 적용	Error·latency·query
Observe	Data·performance·compatibility 확인	Stop 조건 미충족
Contract	오래된 field 제거	별도 승인·복구 계획

8.4 파괴적 변경

다음은 공개 traffic 전환과 같은 한 번의 step으로 묶지 않습니다.

- Column·table 삭제
- Type 축소
- Large backfill
- Long lock
- Irreversible transform
- Old release가 읽을 수 없는 변경

8.5 Backup은 M11-03으로 넘기되 빈칸으로 두지 않는다

이번 manual에서는 최소 handoff를 기록합니다.

- Backup owner
- Backup schedule
- Restore evidence location
- RPO·RTO 초안
- Migration 이전 restore point
- Incident contact

깊은 log·monitoring·backup·incident 설계는 M11-03에서 완성합니다.

9. Exact Release와 IaC

9.1 Build once·promote

M11-01에서 만든 동일 digest 승격 원칙을 public deployment에 적용합니다.

```
source revision
→ build once
→ artifact digest
→ test evidence
→ exact release approval
→ production target
```

Production에서 같은 source를 다시 build하면 dependency·builder·time 차이로 다른 bytes가 나올 수 있습니다.

9.2 Provenance를 검증한다

보관만 하지 않고 다음 expectation과 비교합니다.

- Subject digest = deploy digest
- Signer·builder = 승인 identity
- Source repository = canonical source
- Source revision = 승인 revision
- Build parameters = 승인 값
- Signature valid

9.3 IaC plan

Apply 전 plan에서 특히 봅니다.

Operation	고위험 질문
Create	Public surface·cost·quota가 늘어나는가.
Update	Ingress·IAM·DNS·certificate가 바뀌는가.
Replace	Downtime·data·IP·hostname 영향은 무엇인가.
Delete	Domain·rollback·backup에 필요한 resource인가.

9.4 Desired state와 actual state

Cloud console에서 급히 바꾼 값은 actual에만 있고 source desired state에는 없을 수 있습니다. 이런 out-of-band change는 다음 배포에서 사라지거나 계속 drift로 남습니다.

판정:

```
desired = actual → evidence 기록
desired ≠ actual → release block 또는 승인 exception
unknown actual → 조사 전 block
```

---

10. Progressive Release와 Health

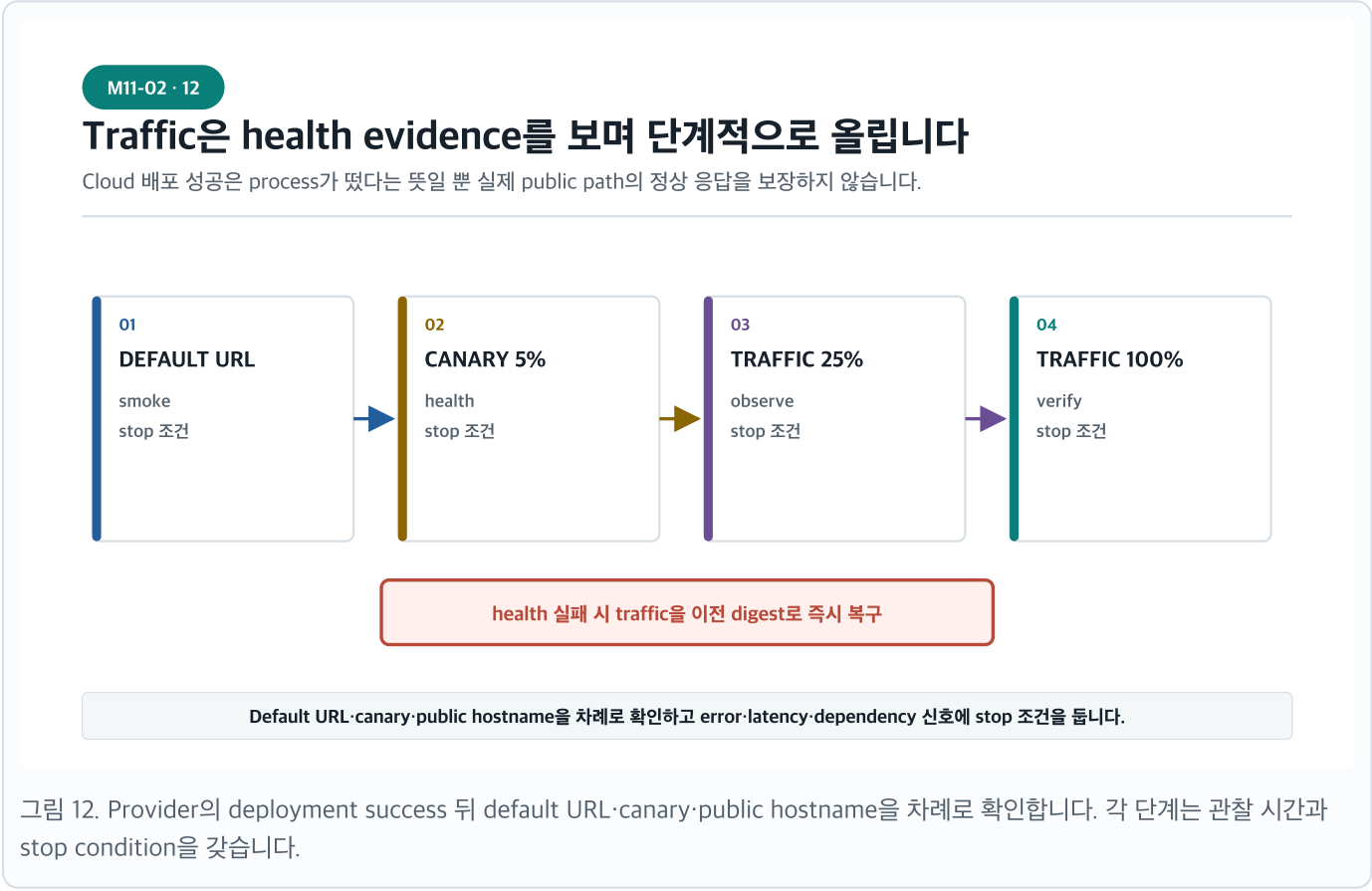
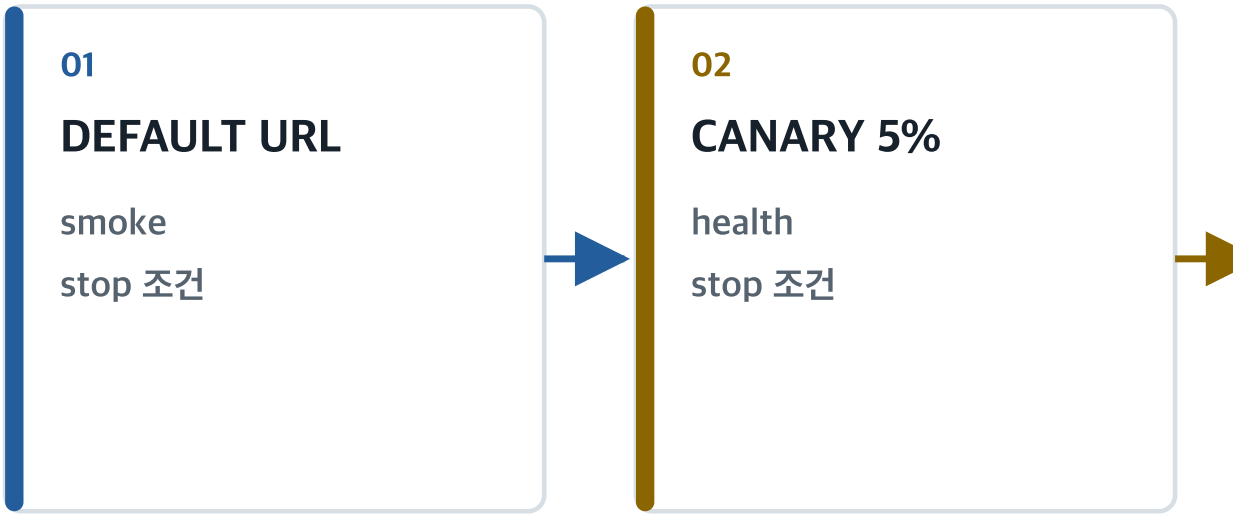


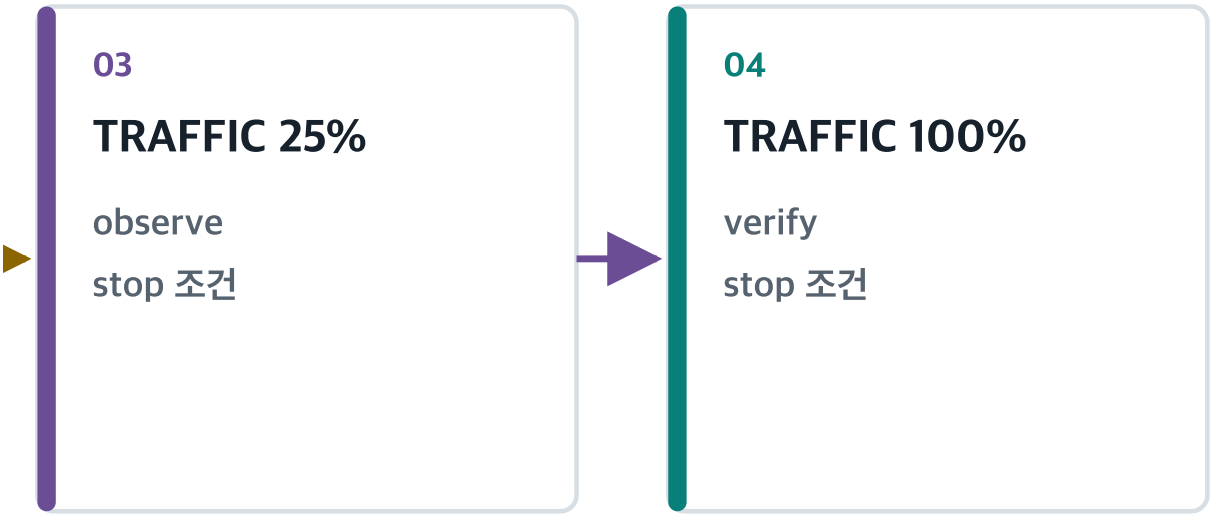
그림 12. Provider의 deployment success 뒤 default URL·canary·public hostname을 차례로 확인합니다. 각 단계는 관찰 시간과 stop condition을 갖습니다.

Cloud 배포 성공은 process가 떴다는 뜻일 뿐 실제 public path의 정상



health 실패 시 traffic을

응답을 보장하지 않습니다.



이전 digest로 즉시 복구

10.1 다섯 단계

단계	Traffic	확인	실패 행동
Resource apply	0%	IaC·runtime·identity	Stop
Default URL smoke	0%	Startup·readiness·core route	Stop
Canary	1~5%	Public path·error·latency	Rollback canary
Expand	25~50%	Business·dependency·capacity	Pause·rollback
Full	100%	All Gate·monitoring handoff	Continue observe

10.2 Health signal을 release ID로 나눈다

전체 service 평균만 보면 canary의 작은 실패가 숨습니다. Metric·log·trace에 최소 다음 label을 붙입니다.

- Service
- Environment
- Region
- Release ID
- Artifact digest 또는 short ID
- Route
- Outcome

10.3 Stop condition 예

Signal	Condition	Window	Action
Startup	하나라도 지속 실패	즉시	Stop
Readiness	Healthy capacity 부족	2~5분	Pause
HTTP 5xx	Baseline 초과	5분	Rollback
Latency	SLO threshold 초과	5~10분	Pause·rollback
Data error	Integrity violation	하나라도	Stop·isolate
Certificate	Hostname·chain failure	하나라도	Traffic 중단

Signal	Condition	Window	Action
DNS	Unexpected target	하나라도	Cutover 중단

Threshold는 서비스 baseline과 SLO에 맞게 정합니다. 위 시간은 개념 예시이지 모든 서비스의 운영 기준이 아닙니다.

10.4 Deployment success와 release success

```
Deployment success = resource change accepted + process ready
Release success = public user path healthy + business correctness + recoverability
```

두 event를 별도로 기록합니다.

---

11. DNS·Traffic·Release Rollback

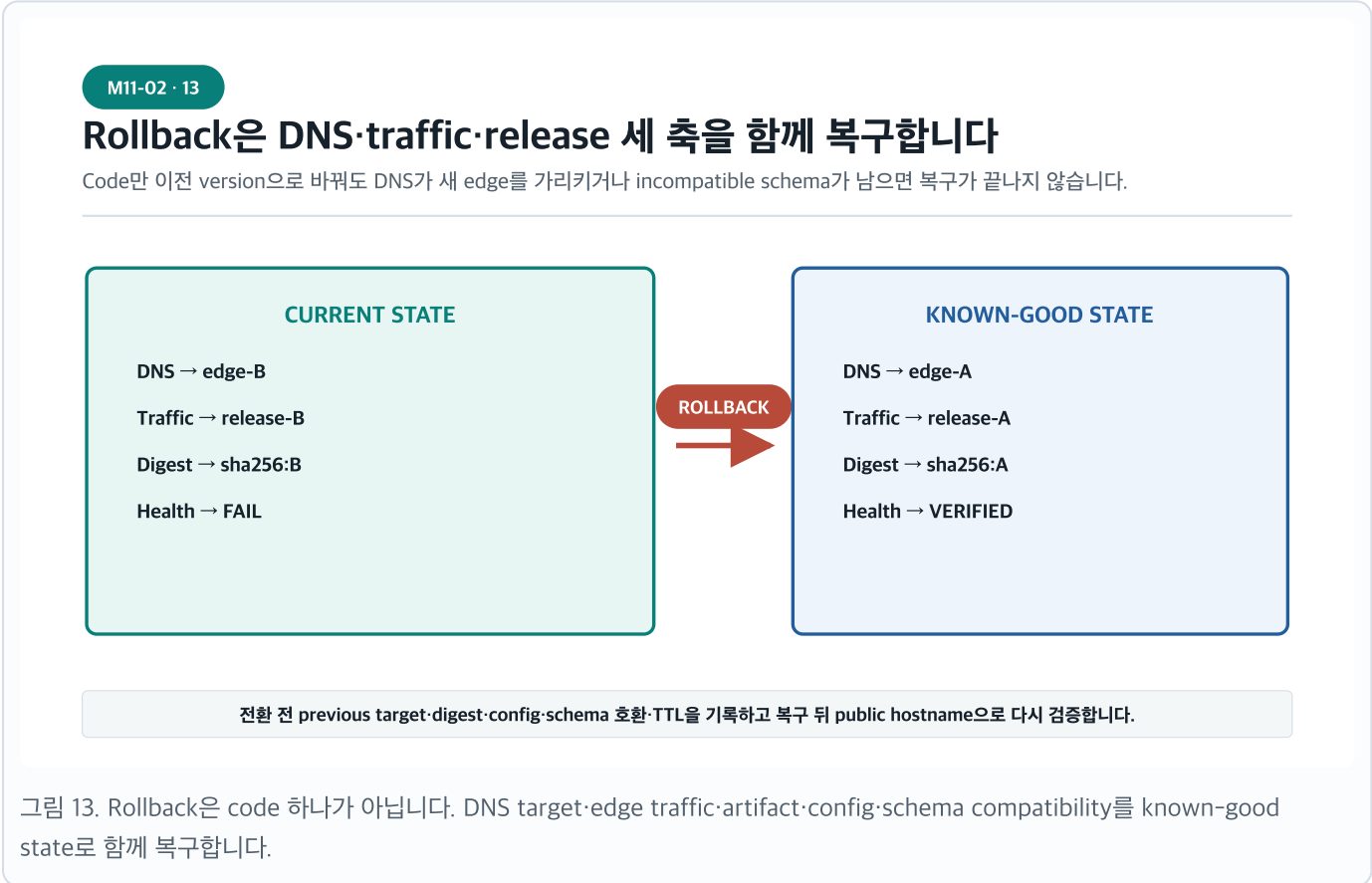


그림 13. Rollback은 code 하나가 아닙니다. DNS target·edge traffic·artifact·config·schema compatibility를 known-good state로 함께 복구합니다.

Code만 이전 version으로 바꿔도 DNS가 새 edge를 가리키거나 incom

CURRENT STATE

DNS → edge-B

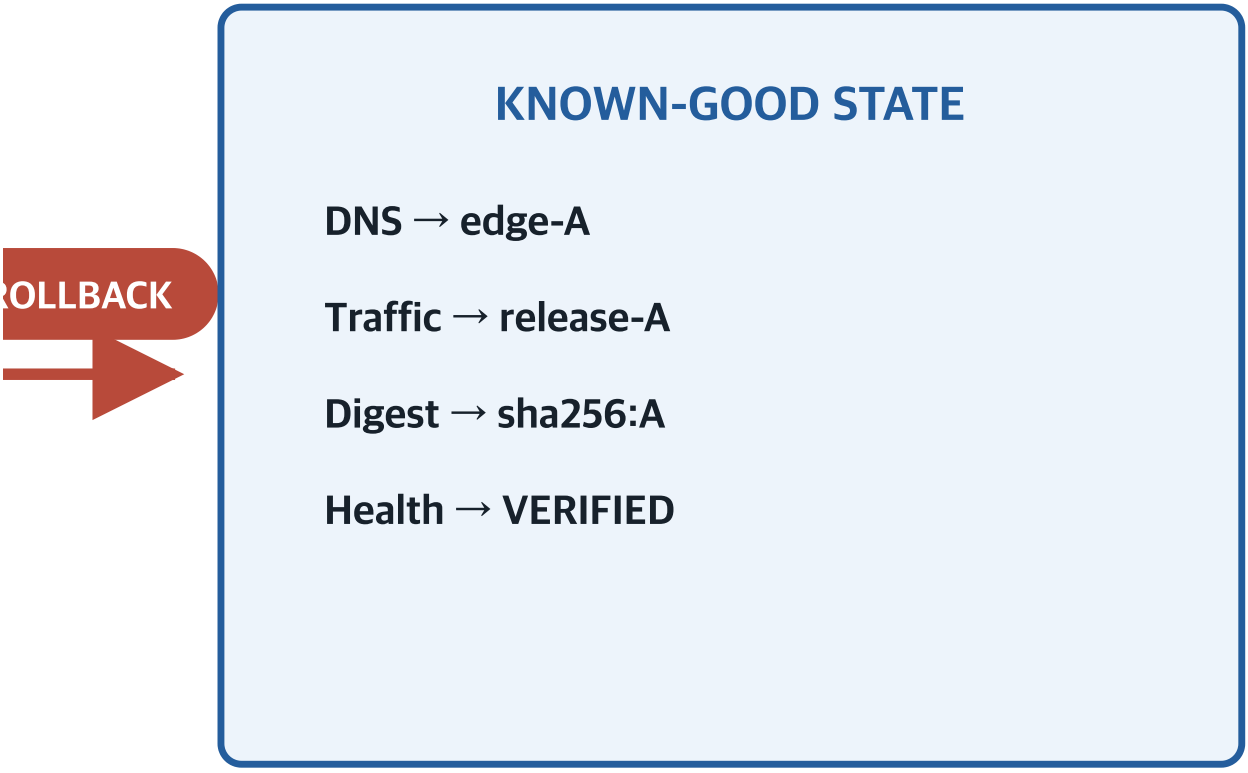
Traffic → release-B

Digest → sha256:B

Health → FAIL

[illegible]

ıpatible schema가 남으면 복구가 끝나지 않습니다.



11.1 Known-good state

배포 전에 기록합니다.

축	Previous	Candidate
DNS target		
Edge config		
Traffic rule		
Artifact digest		
Config version		
Secret reference		
Schema compatibility		
Public health	Verified	Pending

11.2 Rollback 순서 예

1. Traffic 확대를 멈춥니다.
2. Candidate로 가는 edge weight를 0으로 낮춥니다.
3. Previous release의 healthy capacity를 확인합니다.
4. 필요하면 DNS를 previous target으로 되돌립니다.
5. Resolver별 answer를 관찰합니다.
6. Previous digest·config·schema compatibility를 확인합니다.
7. Public hostname으로 smoke와 business path를 재검증합니다.
8. Desired state·incident·evidence를 갱신합니다.

상황에 따라 순서가 달라질 수 있으므로 실제 runbook은 architecture에 맞게 사전 검증합니다.

11.3 DNS rollback의 지연

DNS를 되돌려도 일부 resolver는 TTL 동안 candidate를 계속 가리킬 수 있습니다. 그래서 edge-level traffic switch가 DNS보다 빠른 1차 rollback 수단이 될 수 있습니다. 그러나 edge 자체가 잘못됐거나 target ownership이 문제면 DNS rollback이 필요합니다.

11.4 Roll forward

Schema나 state 때문에 previous release로 돌아가기 어렵다면 수정 release를 forward deploy할 수 있습니다. 이 선택도 즉흥적으로 하지 않고 다음을 비교합니다.

- Restore 예상 시간
- Data 손상 가능성
- Previous compatibility
- Fix 확실성
- Traffic 차단 가능성
- Owner 승인

---

12. 공개 배포 검수 스튜디오

12.1 안전 경계

실습은 다음을 절대 수행하지 않습니다.

- 실제 domain 구매
- DNS record 변경
- Certificate 발급
- External network 호출
- Cloud account 접근
- Production resource 접근
- Live deployment·traffic·migration
- 실제 비용·side effect
- 실제 개인정보·secret 사용

12.2 데스크톱 화면

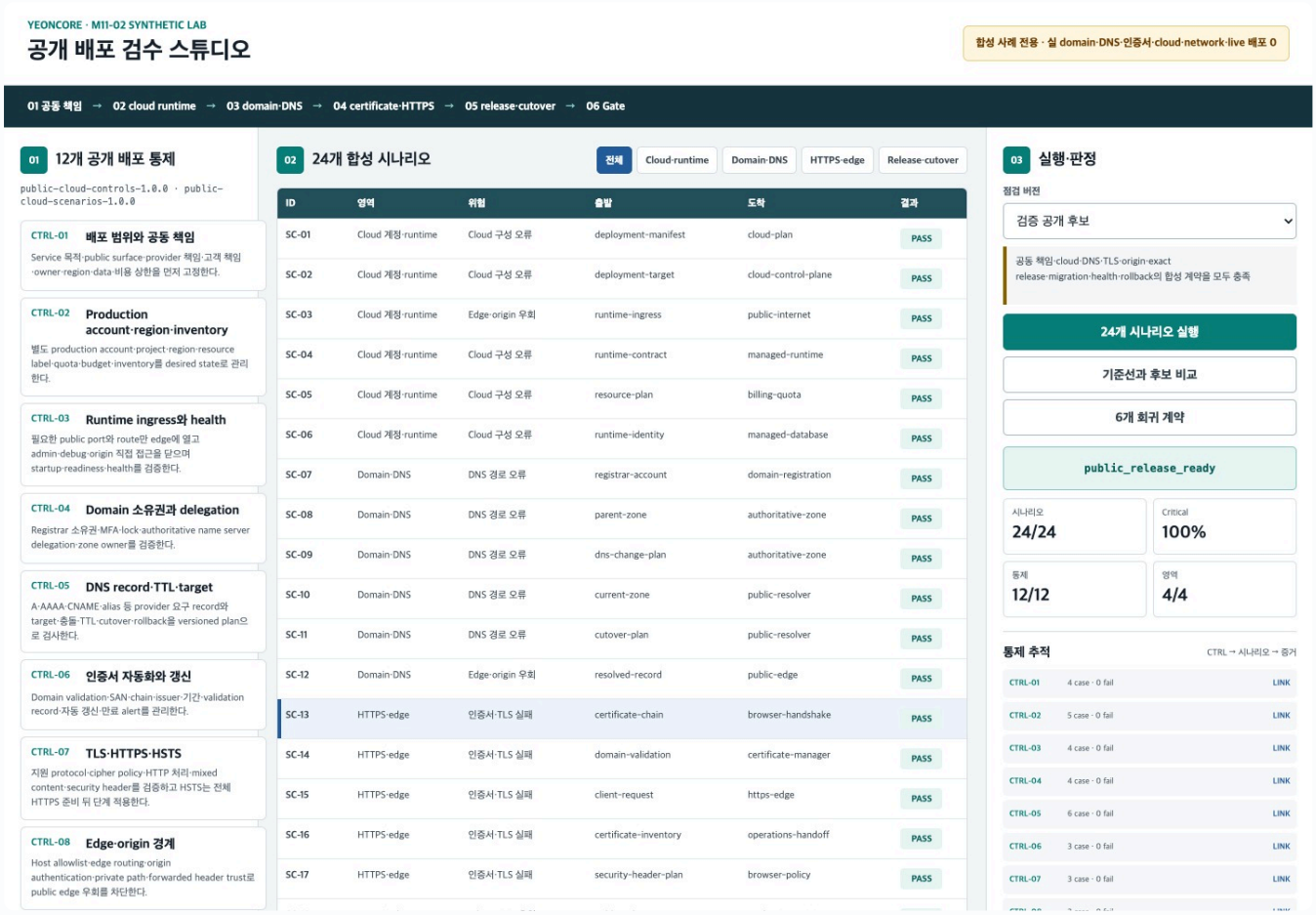


그림 14. Candidate는 24/24·Critical 100%·control 12/12·lane 4/4를 통과합니다. SC-13을 선택해 certificate expected와 actual evidence를 상세 비교할 수 있습니다.

12.3 화면의 세 panel

왼쪽 · 12 control

공동 책임·runtime·domain·DNS·certificate·TLS·origin·identity·artifact·cutover·drift를 scenario에 연결합니다.

가운데 · 24 scenario

Lane·risk·source·sink·result를 비교하고 한 행을 선택해 expected와 actual을 봅니다.

오른쪽 · 실행·판정

Variant·pass count·Critical·control·lane·traceability·run log를 봅니다.

12.4 모바일 화면

그림 15. Managed edge를 사용해도 admin ingress·DNS conflict·certificate renewal·origin·exact release·health evidence가 남으면 `blocked_domain_release_drift`입니다.

12.5 자동 evidence

검사	결과
Contract·engine·server·frontend	333 tests PASS
Public deployment audit	57/57 PASS
Regression contract	6/6 PASS
Baseline	6/24·18 fail

검사	결과
Partial	16/24·8 fail
Candidate	24/24·0 fail
Comparison	fixed 18·regressed 0

12.6 실행

자세한 순서는 90분 실습서를 따릅니다.

```
cd gibalja-public-cloud-deployment-practice/app
python3 app.py --host 127.0.0.1 --port 4295
```

Browser:

```
http://127.0.0.1:4295
```

Loopback가 아닌 host는 server가 거절합니다.

---

13. 열두 공개 배포 Control

ID	Control	핵심 질문	Owner 예
CTRL-01	배포 범위와 공동 책임	Public surface·provider·고객·비용 책임이 고정됐는가.	Service owner
CTRL-02	Production account·region·inventory	Exact target·quota·budget·resource가 맞는가.	Cloud owner
CTRL-03	Runtime ingress와 health	필요한 port만 열리고 startup·readiness가 통과하는가.	Runtime owner
CTRL-04	Domain 소유권과 delegation	Registrar ownership·MFA·lock·NS가 맞는가.	Domain owner

ID	Control	핵심 질문	Owner 예
CTRL-05	DNS record·TTL·target	Type·conflict·TTL·approved target·rollback이 맞는가.	DNS owner
CTRL-06	인증서 자동화와 갱신	SAN·chain·validation·renewal·alert가 있는가.	Certificate owner
CTRL-07	TLS·HTTPS·HSTS	TLS 1.2+·HTTP·mixed content·header policy가 맞는가.	Edge security owner
CTRL-08	Edge·origin 경계	Origin direct access와 forwarded header 우회가 막혔는가.	Network owner
CTRL-09	Config·secret·workload identity	Production reference와 최소 권한이 release에 묶였는가.	Config owner
CTRL-10	Artifact·provenance·migration	Exact digest와 compatible migration만 승격하는가.	Release engineering
CTRL-11	Progressive cutover와 rollback	Health를 보고 traffic을 올리고 이전 상태로 복구 가능한가.	Release owner
CTRL-12	Drift·evidence·Public Gate	Desired=actual이고 잔여 위험이 승인됐는가.	Operations owner

13.1 Control과 checklist의 차이

Checklist 항목이 체크돼도 실제 enforcement와 scenario evidence가 없으면 control이 아닙니다.

```
Control statement
→ implementation
→ scenario
→ expected oracle
→ actual evidence
→ defect·retest
→ release decision
```

13.2 Orphan 0

- Orphan control: 구현·scenario·evidence가 없는 통제
- Orphan scenario: 어떤 위험·control을 검증하는지 모르는 사례

둘 다 0이어야 합니다.

14. 24개 합성 Scenario

14.1 Lane 1 · Cloud account·runtime

ID	Scenario	Expected code	Critical
SC-01	공동 책임·공개 범위·owner	`DEPLOYMENT_SCOPE_IDENTIFIED`	No
SC-02	잘못된 account·project·region 거절	`CLOUD_TARGET_MISMATCH_BLOCKED`	Yes
SC-03	Admin·debug·origin 직접 공개 차단	`UNSAFE_INGRESS_BLOCKED`	Yes
SC-04	Port·startup·readiness	`RUNTIME_HEALTH_READY`	No
SC-05	Resource label·quota·budget	`COST_GUARD_CONFIGURED`	No
SC-06	Managed DB private path·최소 권한	`PRIVATE_DATA_PATH_VERIFIED`	Yes

14.2 Lane 2 · Domain·DNS

ID	Scenario	Expected code	Critical
SC-07	Domain ownership·MFA·lock	`DOMAIN_OWNERSHIP_VERIFIED`	No
SC-08	Authoritative NS delegation	`DNS_DELEGATION_VERIFIED`	Yes
SC-09	Apex·subdomain record type	`DNS_RECORD_TYPE_VALID`	No
SC-10	Conflict·dangling·old record 차단	`DNS_CONFLICT_BLOCKED`	Yes
SC-11	TTL·resolver·rollback 준비	`DNS_CUTOVER_PREPARED`	No
SC-12	Public hostname→approved edge	`PUBLIC_ROUTE_VERIFIED`	Yes

14.3 Lane 3 · HTTPS·edge

ID	Scenario	Expected code	Critical
SC-13	Certificate hostname·SAN·chain·기간	`CERTIFICATE_VALID`	Yes
SC-14	Validation record·auto-renew	`CERTIFICATE_RENEWAL_READY`	Yes

ID	Scenario	Expected code	Critical
SC-15	TLS 1.2+·HTTPS·mixed content	`HTTPS_POLICY_VERIFIED`	No
SC-16	Expiry alert·renew failure runbook	`CERTIFICATE_EXPIRY_MONITORED`	Yes
SC-17	HSTS 단계 적용	`HSTS_STAGED_SAFELY`	No
SC-18	Edge→origin private·authenticated	`ORIGIN_BYPASS_BLOCKED`	Yes

14.4 Lane 4 · Release·cutover

ID	Scenario	Expected code	Critical
SC-19	Exact digest·config·secret ref·provenance	`EXACT_PUBLIC_RELEASE_VERIFIED`	Yes
SC-20	Compatible managed DB migration	`CLOUD_MIGRATION_COMPATIBLE`	No
SC-21	Deployer least privilege·exact target	`DEPLOYER_SCOPE_VERIFIED`	Yes
SC-22	Default URL→canary→public health	`PROGRESSIVE_RELEASE_HEALTHY`	Yes
SC-23	DNS·traffic·release rollback	`PUBLIC_RELEASE_ROLLED_BACK`	Yes
SC-24	24개 계약·잔여 위험 Gate	`PUBLIC_RELEASE_READY`	Yes

14.5 Expected와 Actual

Example SC-14:

```
Expected
  validation_record_present: true
  auto_renew_enabled: true
  renewal_dry_run_passed: true

Defect actual
  validation_record_present: false
  auto_renew_enabled: true
  renewal_dry_run_passed: false
```

Switch 하나만 `true`라고 control을 통과시키지 않습니다. Lifecycle에 필요한 다른 evidence와 함께 비교합니다.

14.6 세 version의 학습 의미

6/24 · 주소부터 공개

Domain과 runtime 이름은 있지만 account·delegation·record·certificate·origin·release·rollback evidence가 섞여 있습니다.

16/24 · Managed edge·미검증 승격

Managed runtime·DNS·certificate 기능은 있지만 unsafe ingress·record conflict·renewal·origin·exact release·health가 불완전합니다.

24/24 · 검증 공개 후보

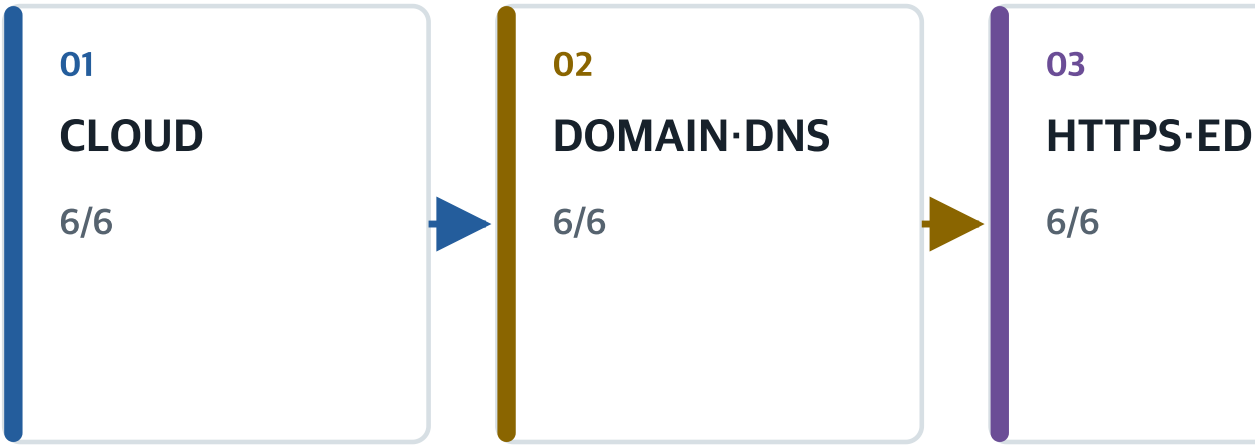
모든 evidence가 합성 expected와 맞고 Critical·control·lane·안전 경계가 통과합니다.

15. Public Release Gate

그림 16. 네 lane 6개씩과 안전 경계를 모두 통과해야 합니다. Origin 노출·DNS conflict·renewal 실패·mutable release 한 건도 평균 점수로 희석하지 않습니다.

Public Release Gate로 배포 위험을

전체·Critical·control·lane coverage 100%와 domain·HTTPS eviden



desired = actual · residual risk 승인

ce·잔여 위험 승인이 함께 필요합니다.

15.1 Gate 기준

Gate	Threshold
Scenario pass rate	100%
Critical pass rate	100%
Lane coverage	4/4
Control coverage	12/12
Orphan control	0
Orphan scenario	0
Duplicate scenario ID	0
Real data·secret	0
External network·cloud account	0
Domain purchase·DNS change·certificate issuance	0
Production access·live deployment·side effect	0

15.2 Decision

Decision	의미
`blocked_public_exposure`	Public ingress·DNS·certificate·data 등 중대한 노출과 경로 결함이 남음
`blocked_domain_release_drift`	Managed 기능은 있으나 domain lifecycle·origin·exact release·health drift가 남음
`public_release_ready`	합성 범위의 모든 Gate가 통과해 실제 배포 검토 후보가 됨

15.3 실제 배포 전 추가 Gate

합성 PASS 뒤에도 실제 조직에서는 다음을 추가합니다.

- Current provider·region·plan validation
- Organization IAM·approval
- Real domain ownership·registrar recovery

- Real DNS change window
- Real certificate endpoint test
- Security·privacy·legal review
- Capacity·load·cost test
- Backup·restore evidence
- Monitoring·on-call·incident readiness
- Change management·stakeholder communication

---

16. 실제 프로젝트 적용 순서

Phase 0 · Scope

산출물:

- One-sentence public objective
- Service owner·domain owner·cloud owner·release owner
- Public hostname·surface
- Region·data·cost·availability 요구
- Provider shared responsibility record

Gate:

```
Unknown owner = BLOCK
Unknown account·region = BLOCK
Unknown public surface = BLOCK
```

Phase 1 · Runtime 먼저 준비

Custom domain보다 provider default URL 또는 internal path에서 다음을 확인합니다.

- Exact artifact digest
- Startup·readiness
- Config version·secret reference
- Workload identity

- Managed database private path
- Admin·debug disabled

Phase 2 · Domain과 DNS inventory

- Registrar ownership·MFA·lock·expiry
- Parent NS·authoritative zone
- Current record·conflict·dangling
- Target record·TTL
- Validation record
- Previous target·rollback

Phase 3 · Certificate와 edge

- SAN·chain·validity
- Validation·auto-renew
- HTTPS policy
- Host routing
- Origin private·authenticated
- Header baseline
- Expiry monitoring

Phase 4 · Pre-cutover

- TTL lowered early
- Default URL smoke
- Canary rule prepared
- Exact release approval
- IaC plan reviewed
- Migration compatibility
- Known-good state
- Stop·rollback owner

Phase 5 · Cutover

```
apply exact release
→ default URL smoke
→ certificate handshake
→ authoritative DNS change
→ resolver evidence
→ canary traffic
→ public path health
→ traffic expand
```

Phase 6 · Post-cutover

- Public hostname from multiple networks
- Error·latency·business correctness
- Certificate serve·expiry
- DNS answer·TTL
- Runtime capacity·database
- Cost·quota
- Desired=actual drift
- Monitoring·M11-03 handoff

Phase 7 · Cleanup

Old target과 record를 즉시 지우지 않습니다.

1. Rollback window 종료 확인
2. Resolver cache와 traffic 0 확인
3. Certificate binding·alternate hostname 확인
4. DNS record 안전 제거
5. Cloud resource 폐기 승인
6. Inventory·IaC·cost 정리

17. 문제 해결 지도

증상	먼저 볼 경계	확인	위험한 즉흥 조치
Domain이 열리지 않음	Delegation·DNS	NS·record·TTL	Record 반복 변경
일부 사용자만 old site	DNS cache	Resolver별 answer	TTL을 T0에만 낮춤
Certificate name mismatch	SAN·edge binding	Actual served cert	Browser warning 무시
Certificate 곧 만료	Renewal lifecycle	Validation·owner·alert	Manual 교체만 반복
HTTPS page asset 깨짐	Mixed content·CSP	Browser network·console	Header 전부 제거
Edge는 정상, origin 403	Origin auth·Host	Edge identity·Host allowlist	Origin public 공개
Direct origin 접근 가능	Network policy	Public IP·firewall	WAF만 믿음
Deploy success, 502	Port·readiness	Listen port·startup log	Health check 비활성화
Wrong database	Environment binding	DB endpoint·role	Credential 복사
Migration 후 rollback 실패	Compatibility	Expand-contract	Schema 강제 복원
Canary만 오류	Release labels	Candidate metric	전체 평균만 확인
Traffic 전환 후 지연	Capacity·cold start	Instance·quota	무제한 scaling
비용 급증	Budget·loop·attack	Cost labels·request	Resource 무조건 삭제
DNS rollback 느림	TTL·edge traffic	Resolver cache	계속 record 변경
HSTS 뒤 subdomain 불가	HTTPS readiness	includeSubDomains	임의 browser 설정 안내
Provider default URL 노출	Alternate surface	Ingress policy	Security by obscurity
IaC 다음 배포가 원복	Drift	Console vs source	Console 재수정
Secret가 log에 보임	Logging boundary	Sanitization·trace	Log만 삭제
Deployment admin key 노출	Identity	Revoke·audit·federation	Key rename
Public path만 실패	DNS·TLS·edge	End-to-end check	Runtime만 restart

17.1 문제 해결 순서

```
증상 재현
→ public hostname 기준 경계 찾기
→ desired와 actual 수집
→ latest change·release ID 연결
→ stop condition 판단
→ rollback 또는 수정
→ 같은 경로 재검
→ desired state와 runbook 갱신
```

---

18. 공식 근거와 현재성

18.1 DNS·TLS·ACME

- RFC 1034 · Domain Names – Concepts and Facilities
- RFC 8446 · The Transport Layer Security Protocol Version 1.3
- RFC 8555 · Automatic Certificate Management Environment
- NIST SP 800-52 Rev. 2 · Guidelines for TLS Implementations
- Let's Encrypt Documentation

NIST page의 2026. 5. 7. planning note는 SP 800-52 Rev. 2가 review 중이라고 밝힙니다. 이 manual은 review date의 개념 근거로 사용하며 실제 policy는 최신 publication과 조직 기준을 확인합니다.

18.2 Web security

- OWASP Transport Layer Security Cheat Sheet
- OWASP HTTP Strict Transport Security Cheat Sheet
- OWASP HTTP Headers Cheat Sheet

18.3 Shared responsibility

- AWS Shared Responsibility Model
- Microsoft Azure Shared Responsibility in the Cloud
- Google Cloud Shared Responsibility and Shared Fate

18.4 Provider custom domain examples

- AWS App Runner Custom Domains
- Azure Container Apps Custom Domains and Certificates
- Google Cloud Run Mapping Custom Domains

Provider 기능은 자주 바뀝니다. 이 문서들은 특정 provider 추천이 아니라 ownership·record·validation·renewal·production suitability를 확인하는 예입니다.

18.5 출처를 읽는 질문

1. 이 문서는 어떤 service·region·plan에 적용되는가.
2. Feature가 GA·preview·제한 중 무엇인가.
3. Domain ownership을 무엇으로 검증하는가.
4. Apex와 subdomain record 요구가 다른가.
5. Certificate 발급과 renewal 책임은 누구에게 있는가.
6. Validation record를 계속 유지해야 하는가.
7. Origin은 private·authenticated할 수 있는가.
8. Production 권장 architecture는 무엇인가.
9. Rollback과 deletion lifecycle은 무엇인가.
10. 문서 update date와 deprecation notice는 무엇인가.

---

19. 셀프 테스트 30

Q01. Domain과 DNS의 차이는 무엇인가.

모범 답안
Domain은 이름을 등록·소유·갱신할 권리와 관리 대상이고, DNS는 그 이름을 nameserver·record·resolver를 통해 target으로 해석하는 분산 system입니다. Domain을 소유해도 delegation과 record가 틀리면 service에 연결되지 않습니다.

Q02. Parent delegation과 authoritative record를 왜 따로 확인하는가.

모범 답안

Parent NS가 다른 nameserver를 가리키면 내가 수정한 zone이 authoritative하지 않을 수 있습니다. Parent NS·zone NS·실제 authoritative answer가 같은 관리 경계를 향하는지 확인해야 합니다.

Q03. DNS 변경이 모든 사용자에게 동시에 보이지 않는 이유는 무엇인가.

모범 답안

Recursive resolver가 이전 answer를 TTL 동안 cache하기 때문입니다. Resolver마다 query 시점이 달라 cache expiry도 다르므로 일정 시간 old와 new target이 함께 관찰될 수 있습니다.

Q04. TTL을 cutover 직전에 낮추면 충분하지 않은 이유는 무엇인가.

모범 답안

이미 이전 높은 TTL로 cache한 resolver는 그 TTL이 끝날 때까지 old answer를 유지합니다. 사전에 충분히 일찍 낮춰 기존 cache가 새 짧은 TTL로 갱신될 시간을 줘야 합니다.

Q05. Zone apex와 subdomain의 record 선택이 달라질 수 있는 이유는 무엇인가.

모범 답안

일반 DNS 규칙에서 apex는 NS·SOA 등 다른 record와 함께 있어 CNAME 제약이 생깁니다. Provider는 apex에 A·AAAA·alias·ANAME 등을 요구하고 subdomain에는 CNAME을 요구할 수 있으므로 공식 요구를 확인합니다.

Q06. Dangling DNS record의 위험은 무엇인가.

모범 답안

DNS가 삭제됐거나 더 이상 소유하지 않은 cloud·external resource를 가리키면 제3자가 그 target을 확보해 hostname traffic을 받을 가능성이 생깁니다. Traffic과 record를 먼저 정리한 뒤 resource를 폐기합니다.

Q07. Certificate의 SAN은 무엇을 확인하는가.

모범 답안

Certificate가 보호하도록 발급된 DNS hostname·IP 목록입니다. Client가 요청한 hostname이 SAN과 match해야 server identity 검증이 통과합니다.

Q08. Certificate manager에서 issued가 보여도 public handshake를 확인해야 하는 이유는 무엇인가.

모범 답안
발급 상태와 edge binding·실제 serve 상태는 다를 수 있습니다. Edge가 이전 certificate를 serve하거나 다른 hostname route에 연결했을 수 있으므로 public endpoint에서 hostname·chain·기간을 확인합니다.

Q09. Auto-renew를 켰는데도 renewal이 실패할 수 있는 이유 세 가지를 쓰라.

모범 답안
Validation DNS record가 삭제됨, certificate manager의 domain 권한이 사라짐, route·challenge path가 변경됨 등이 있습니다. 그래서 validation evidence·dry run·expiry alert·owner를 함께 관리합니다.

Q10. ACME의 역할은 무엇인가.

모범 답안
Domain control validation, certificate request·issuance·renewal을 자동화하는 protocol입니다. 자동화해도 domain ownership·key 보호·failure monitoring 책임은 남습니다.

Q11. HTTPS가 application 안전 전체를 증명하지 않는 이유는 무엇인가.

모범 답안
HTTPS는 주로 통신 encryption·integrity와 endpoint identity를 보호합니다. Authorization·input validation·data permission·origin exposure·secure coding·renewal 성공까지 증명하지 않습니다.

Q12. Mixed content란 무엇인가.

모범 답안
HTTPS page가 HTTP script·image·font·API 등 덜 보호된 resource를 불러오는 상태입니다. Browser가 차단하거나 공격자가 resource를 변경할 수 있어 모든 resource와 redirect chain을 HTTPS로 검증합니다.

Q13. HSTS를 짧은 max-age부터 적용하는 이유는 무엇인가.

모범 답안
Browser가 policy를 cache하므로 잘못 적용하면 즉시 server에서 되돌려도 client에 남습니다. 모든 host를 준비하고 짧게 시험한 뒤 영향과 includeSubDomains를 검토해 기간을 늘립니다.

Q14. Origin bypass는 무엇인가.

모범 답안

Client가 승인 public edge를 거치지 않고 application origin에 직접 접근하는 경로입니다. WAF·rate limit·host policy를 우회하므로 private path·firewall·edge identity·Host allowlist로 막습니다.

Q15. Forwarded header를 아무 source에서나 믿으면 안 되는 이유는 무엇인가.

모범 답안

Client가 X-Forwarded-For·Proto·Host 등을 위조해 IP·HTTPS·host 기반 policy를 속일 수 있습니다. 승인 proxy source에서 받은 header만 신뢰하고 나머지는 제거·재작성합니다.

Q16. Shared responsibility model의 핵심은 무엇인가.

모범 답안

Provider와 고객이 service layer별 책임을 나눈다는 뜻입니다. Managed 범위가 달라져도 account·identity·configuration·data·domain·release 등 고객 책임을 service별 공식 문서와 owner로 고정해야 합니다.

Q17. 첫 공개 서비스에 managed runtime이 유리할 수 있는 이유는 무엇인가.

모범 답안

Server patch·basic scaling·routing 같은 운영 부담을 줄여 application과 release evidence에 집중할 수 있습니다. 다만 특수 network·runtime·compliance 요구와 provider 제한을 함께 검토합니다.

Q18. Production target mismatch를 시작 전에 차단해야 하는 값은 무엇인가.

모범 답안

Organization·account·project·subscription·region·environment·resource ID·artifact digest·config version·approval ID입니다. 화면 색이나 이름 접두사 대신 immutable ID를 비교합니다.

Q19. Managed database가 자동으로 보장하지 않는 것 네 가지를 쓰라.

모범 답안

Private connectivity, workload least privilege, application schema compatibility, migration 순서가 대표적입니다. Data residency·backup restore objective·query correctness도 팀 책임으로 남을 수 있습니다.

Q20. Startup·readiness·liveness의 차이는 무엇인가.

모범 답안

Startup은 초기화 완료, readiness는 새 traffic 수신 가능, liveness는 회복 불가능해 restart가 필요한지를 판단합니다. 같은 endpoint 하나로 모든 dependency를 강하게 검사하면 restart cascade가 생길 수 있습니다.

Q21. Deployment identity와 workload identity를 왜 분리하는가.

모범 답안

배포자는 resource·release 변경 권한이 필요하고 workload는 실행 중 DB·API 최소 권한만 필요합니다. 공유하면 runtime 침해가 deployment admin 권한으로 확대됩니다.

Q22. Secret reference를 release bundle에 넣고 secret value를 넣지 않는 이유는 무엇인가.

모범 답안

어떤 secret version을 썼는지는 재현해야 하지만 원문을 manifest·log·approval에 복제하면 노출 범위가 커집니다. Reference·version·consumer identity·owner만 결합합니다.

Q23. Exact release를 이루는 핵심 요소를 쓰라.

모범 답안

Pinned artifact digest, config version, secret references, IaC revision, migration digest, exact target, approval, traffic plan과 rollback target입니다. Mutable tag만으로는 exact release가 아닙니다.

Q24. IaC plan에서 replace와 delete를 특별히 보는 이유는 무엇인가.

모범 답안

Replace는 downtime·IP·hostname·data lifecycle을 바꿀 수 있고 delete는 rollback·certificate·database·network resource를 없앨 수 있습니다. 예상하지 않은 destructive operation은 apply 전에 차단합니다.

Q25. Expand-contract migration의 목적은 무엇인가.

모범 답안

호환 구조를 먼저 추가하고 이전·새 application이 함께 동작한 뒤 오래된 구조를 별도 단계로 제거해 rolling deploy와 rollback 가능성을 확보하는 것입니다.

Q26. Canary metric을 release ID로 나눠야 하는 이유는 무엇인가.

모범 답안
Canary traffic 비율이 작으면 전체 평균에서 오류와 latency가 희석됩니다. Environment·region·release·route label로 candidate만 분리해 baseline과 비교합니다.

Q27. Deployment success와 release success의 차이는 무엇인가.

모범 답안
Deployment success는 resource와 process가 적용·준비된 상태이고, release success는 실제 public hostname 경로·업무 기능·dependency·recoverability가 기준을 통과한 상태입니다.

Q28. Rollback의 세 축은 무엇인가.

모범 답안
DNS 또는 edge target, traffic policy, artifact digest·configuration입니다. 여기에 database schema가 previous release와 호환되는지 반드시 확인합니다.

Q29. Candidate가 24/24여도 실제 production 승인이 아닌 이유는 무엇인가.

모범 답안
합성 고정 사례와 loopback 환경만 검증했기 때문입니다. 실제 provider·account·domain·DNS·certificate·traffic·data·cost·organization approval·incident readiness는 별도 evidence가 필요합니다.

Q30. `public_release_ready`의 정확한 의미는 무엇인가.

모범 답안
정의한 합성 범위에서 24개 scenario·Critical·12 control·4 lane·안전 경계·잔여 위험 조건이 모두 통과해 실제 배포 검토 후보가 됐다는 판정입니다. Cloud·TLS 보안 인증이나 무사고 보증이 아닙니다.

20. 마지막 한 장 요약

요청 경로

```
Browser URL
→ Parent delegation
→ Authoritative DNS record·TTL
→ Public resolver cache
→ Certificate·TLS·HTTPS edge
→ Private authenticated origin
→ Exact digest runtime
→ Private least-privilege data
```

공개 전 다섯 질문

1. **누가 소유하는가:** Domain·account·resource·cost·incident owner가 있는가.
2. **어디를 가리키는가:** DNS·edge·runtime·data가 같은 production target인가.
3. **어떻게 보호하는가:** Certificate·TLS·origin·identity·secret 경계가 있는가.
4. **무엇을 배포하는가:** Exact digest·config·migration·approval이 고정됐는가.
5. **어떻게 되돌리는가:** DNS·traffic·release·schema known-good state가 있는가.

Gate

```
Cloud 6/6
+ Domain·DNS 6/6
+ HTTPS·edge 6/6
+ Release·cutover 6/6
+ Critical 100%
+ Control 12/12
+ Real·live action 0
= public_release_ready
```

다음 매뉴얼

M11-03에서는 이 공개 endpoint가 지속적으로 정상인지 **로그·모니터링·백업·장애 대응**으로 이어서 설계합니다. M11-02의 handoff는 environment·region·release ID label, public path health, DNS target, certificate expiry, database backup owner, rollback trigger입니다.