

온라인 게임 개발도 쉽고 빠르게

GameAnvil on NHN Cloud

전만철 / NHN Bigfoot / 게임서버엔진팀

게임 개발 이야기

게임 개발







싱글 게임





온라인 게임



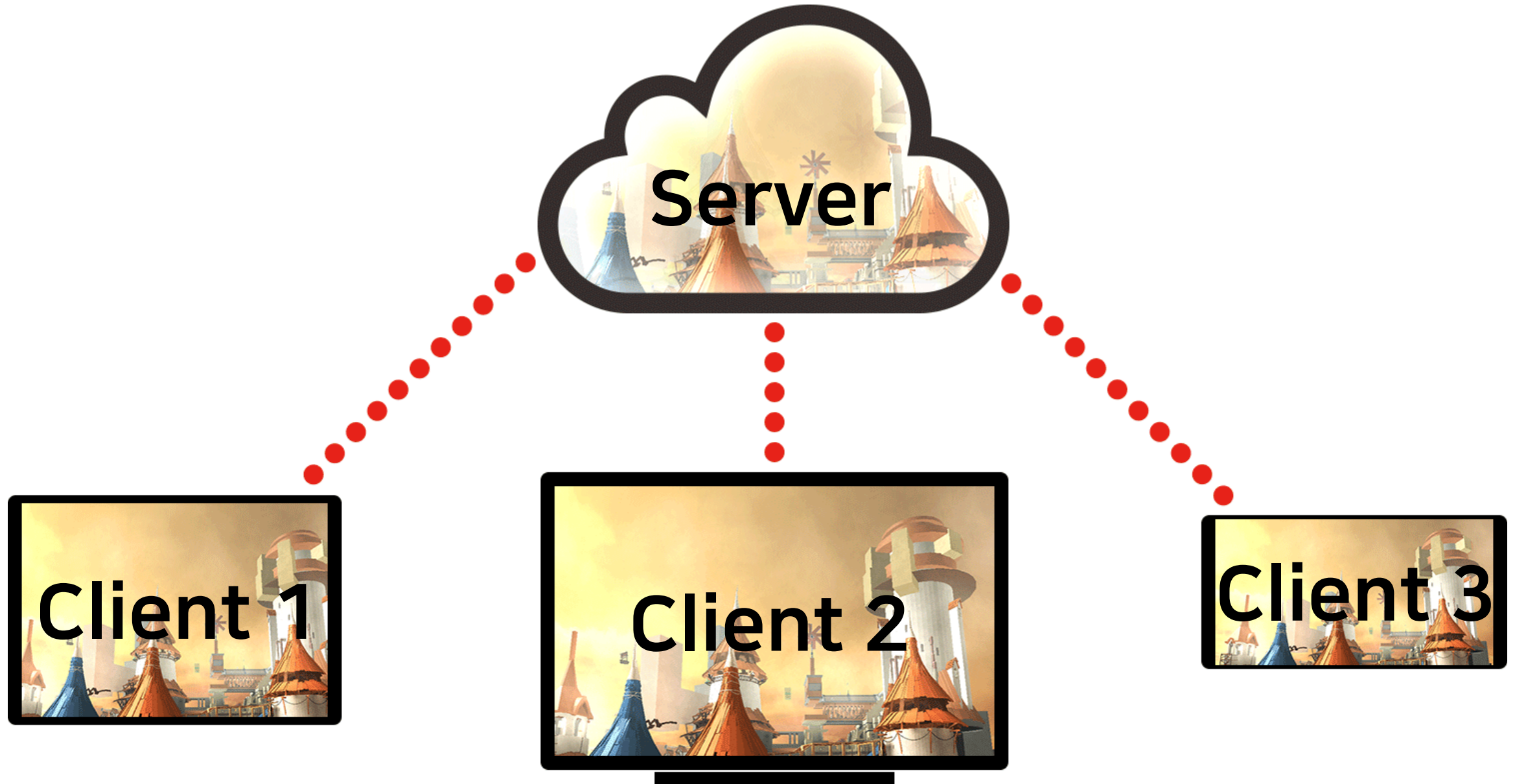
온라인 게임





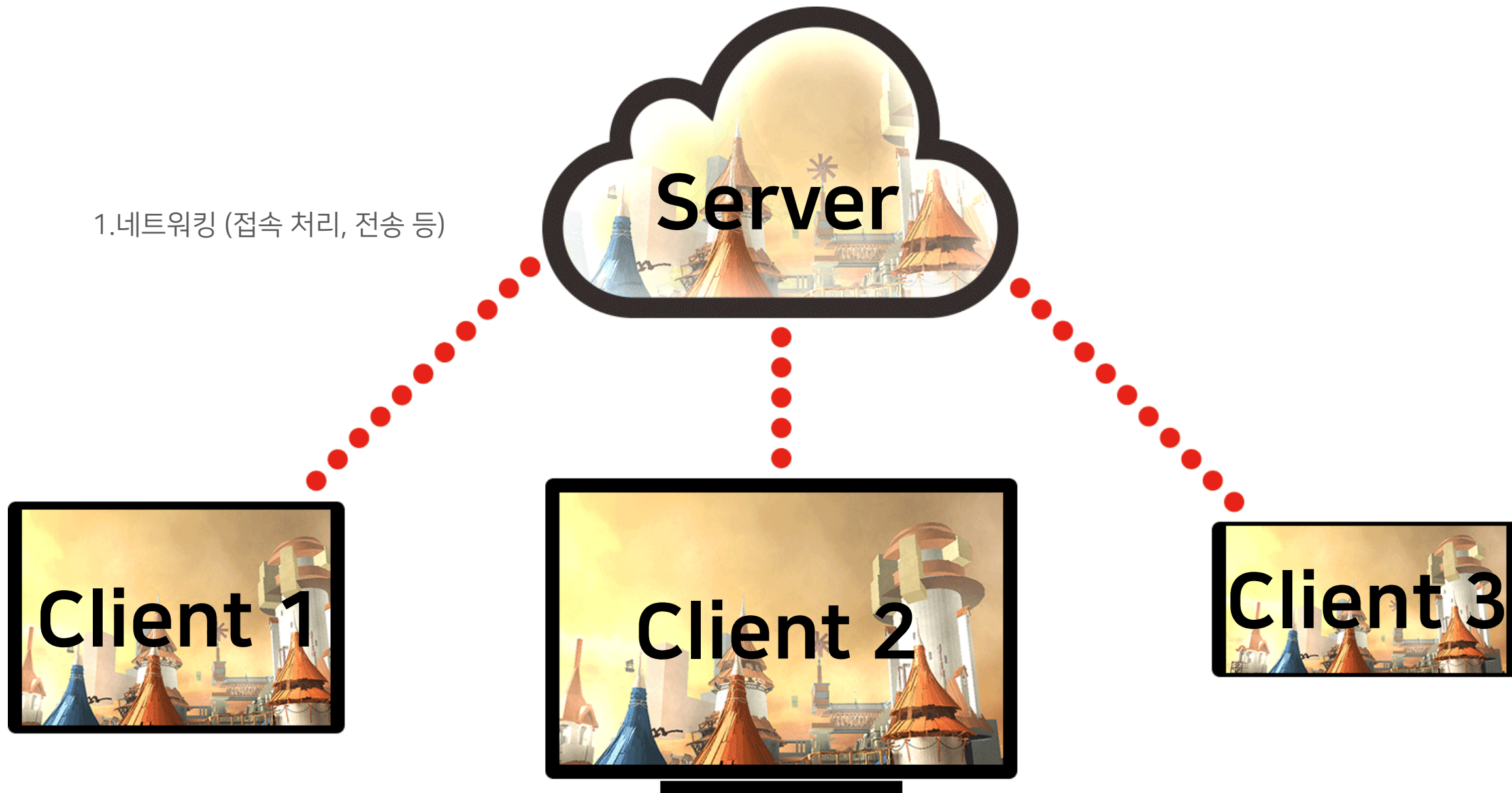
온라인 게임 개발은 왜 까다로울까?

클라이언트-서버 모델

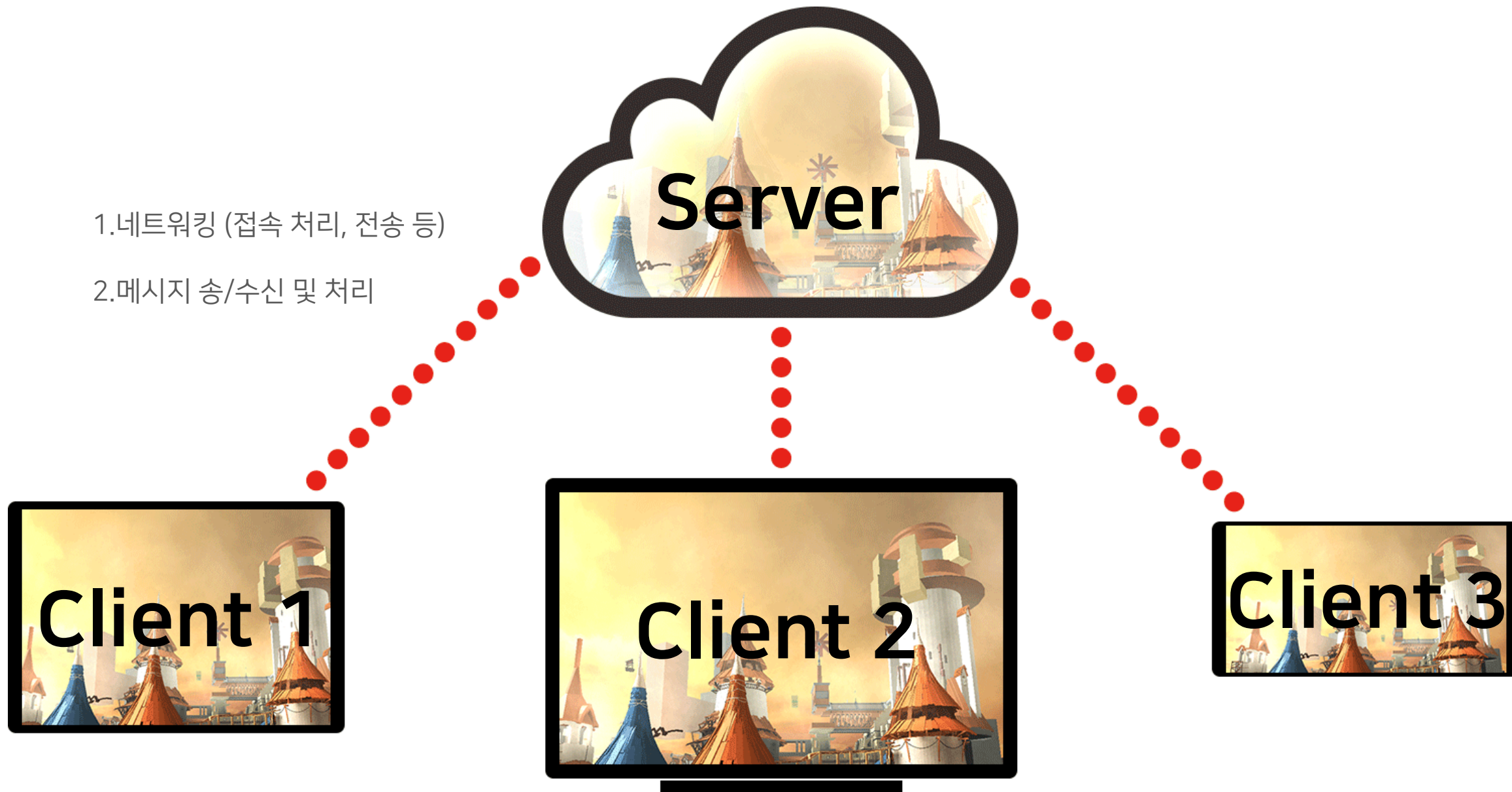


클라이언트-서버 모델

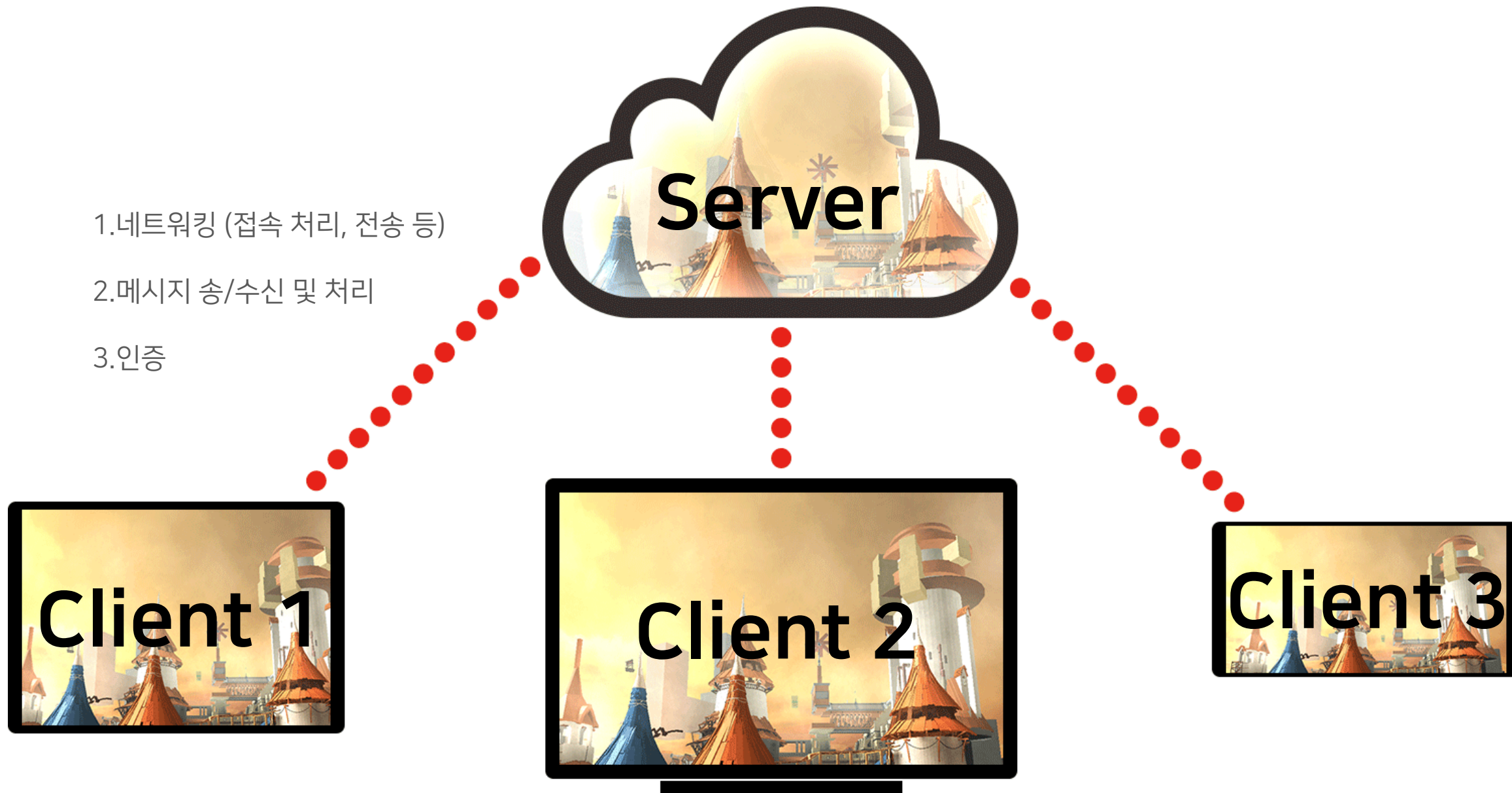
1.네트워킹 (접속 처리, 전송 등)



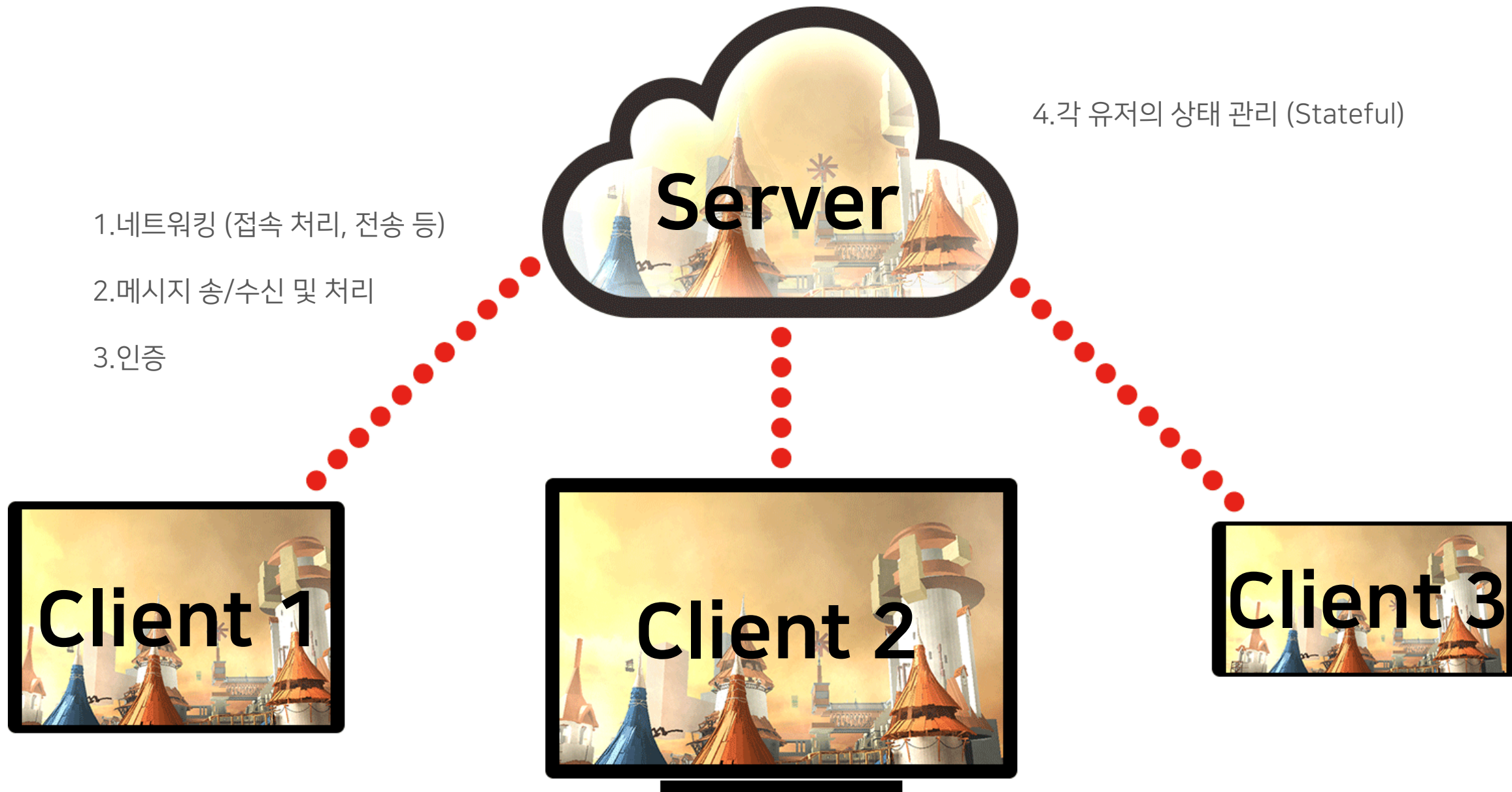
클라이언트-서버 모델



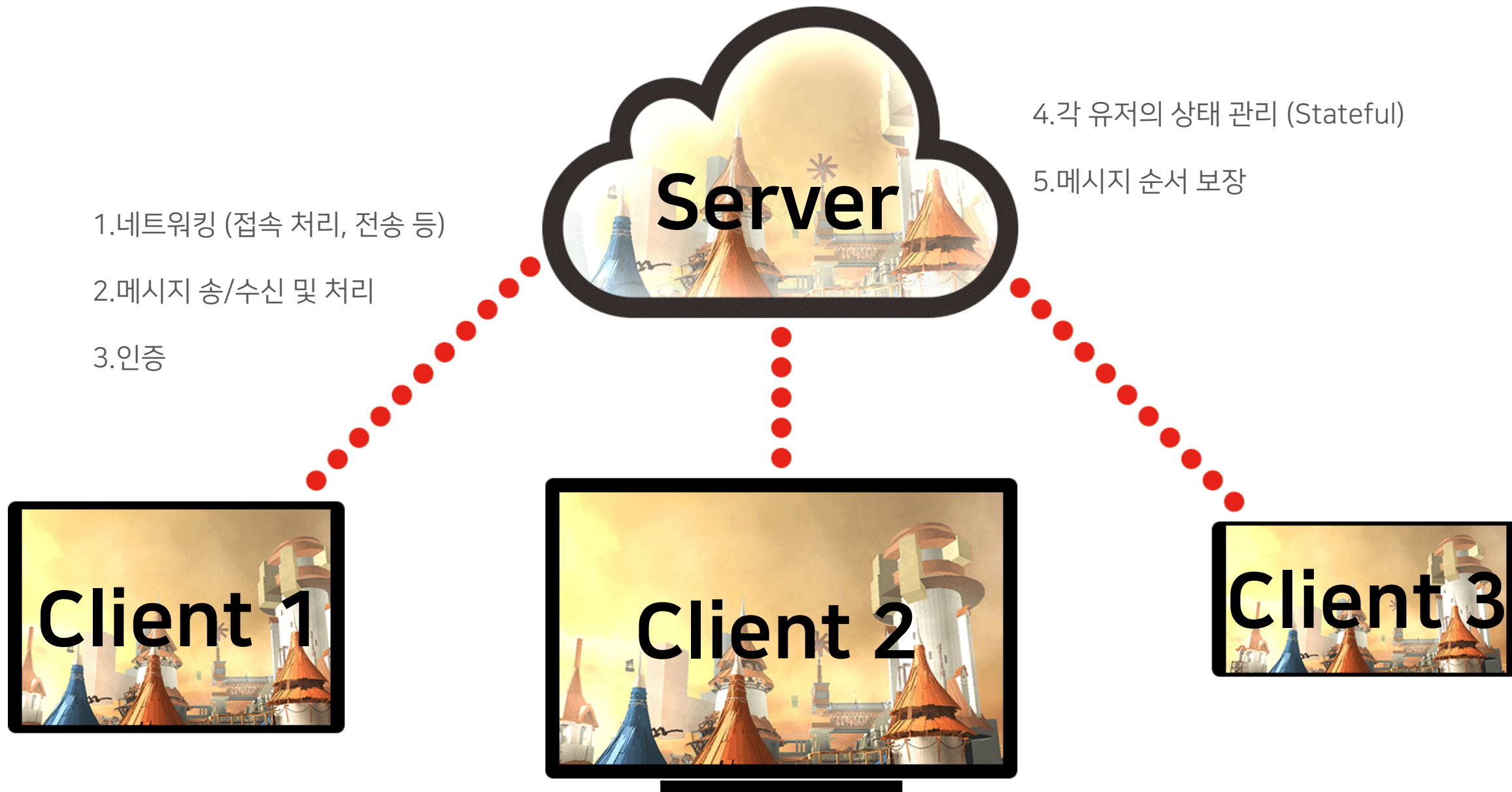
클라이언트-서버 모델



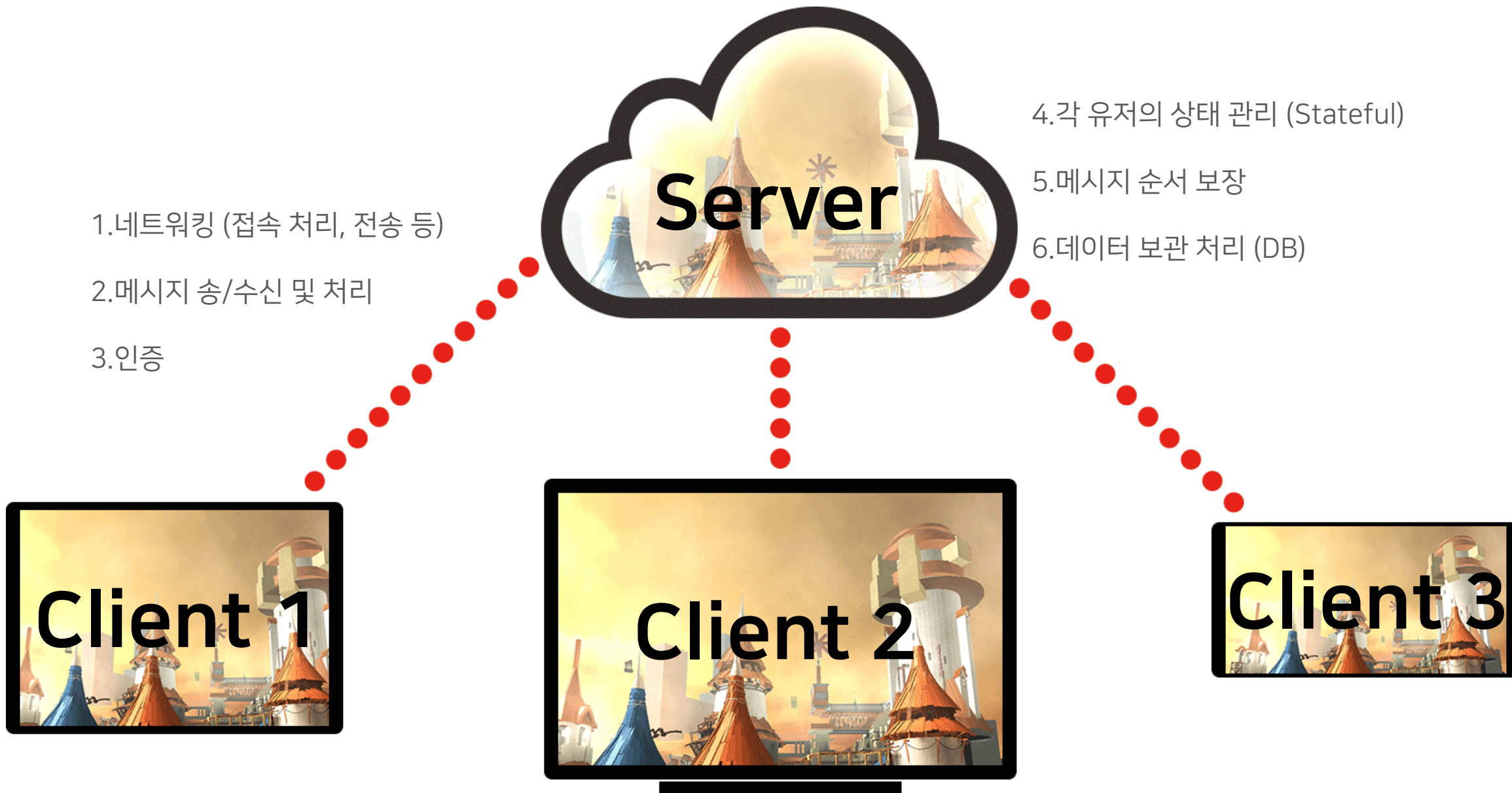
클라이언트-서버 모델



클라이언트-서버 모델



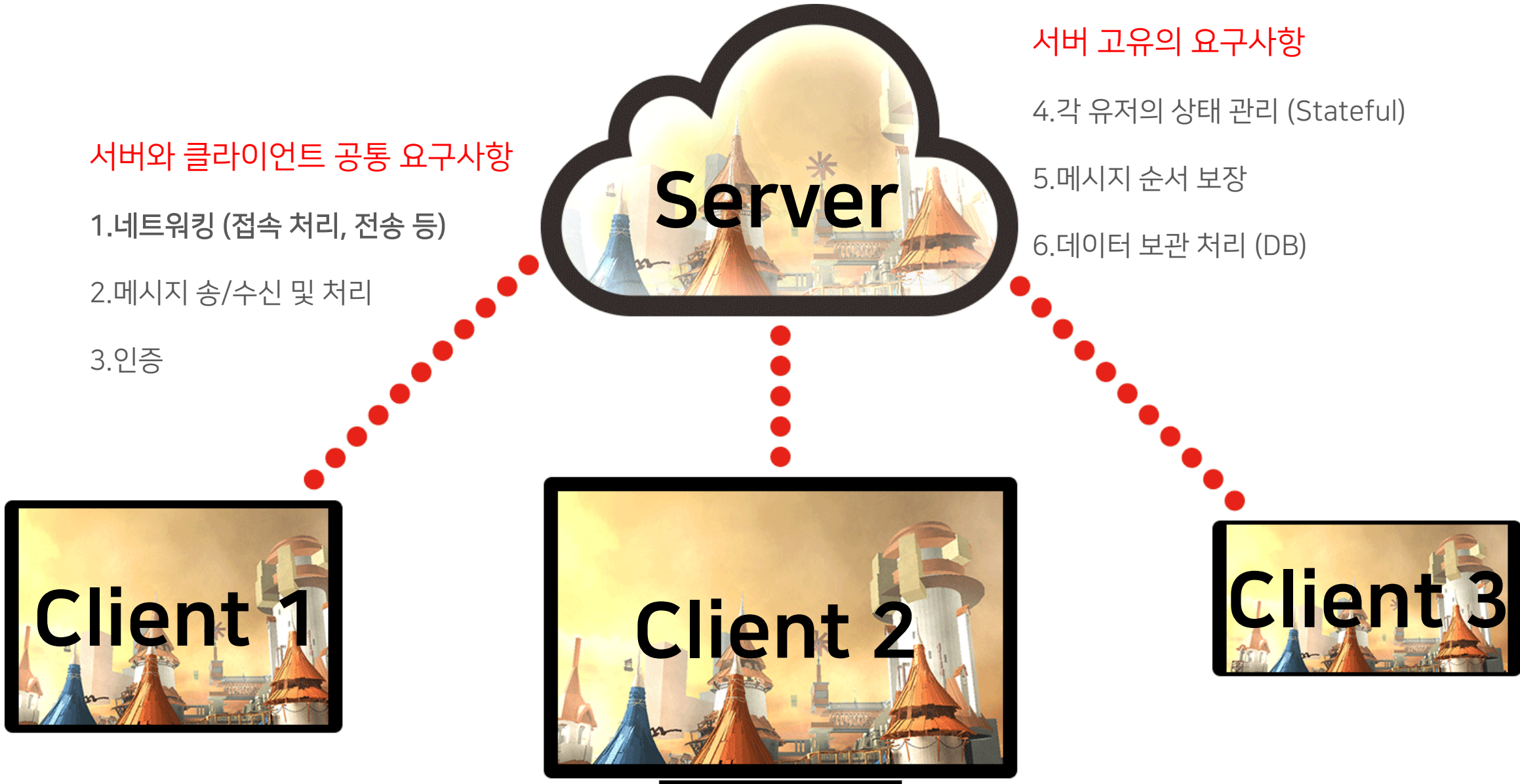
클라이언트-서버 모델



클라이언트-서버 모델

서버와 클라이언트 공통 요구사항

- 1.네트워킹 (접속 처리, 전송 등)
- 2.메시지 송/수신 및 처리
- 3.인증



Server

서버 고유의 요구사항

- 4.각 유저의 상태 관리 (Stateful)
- 5.메시지 순서 보장
- 6.데이터 보관 처리 (DB)

Client 1

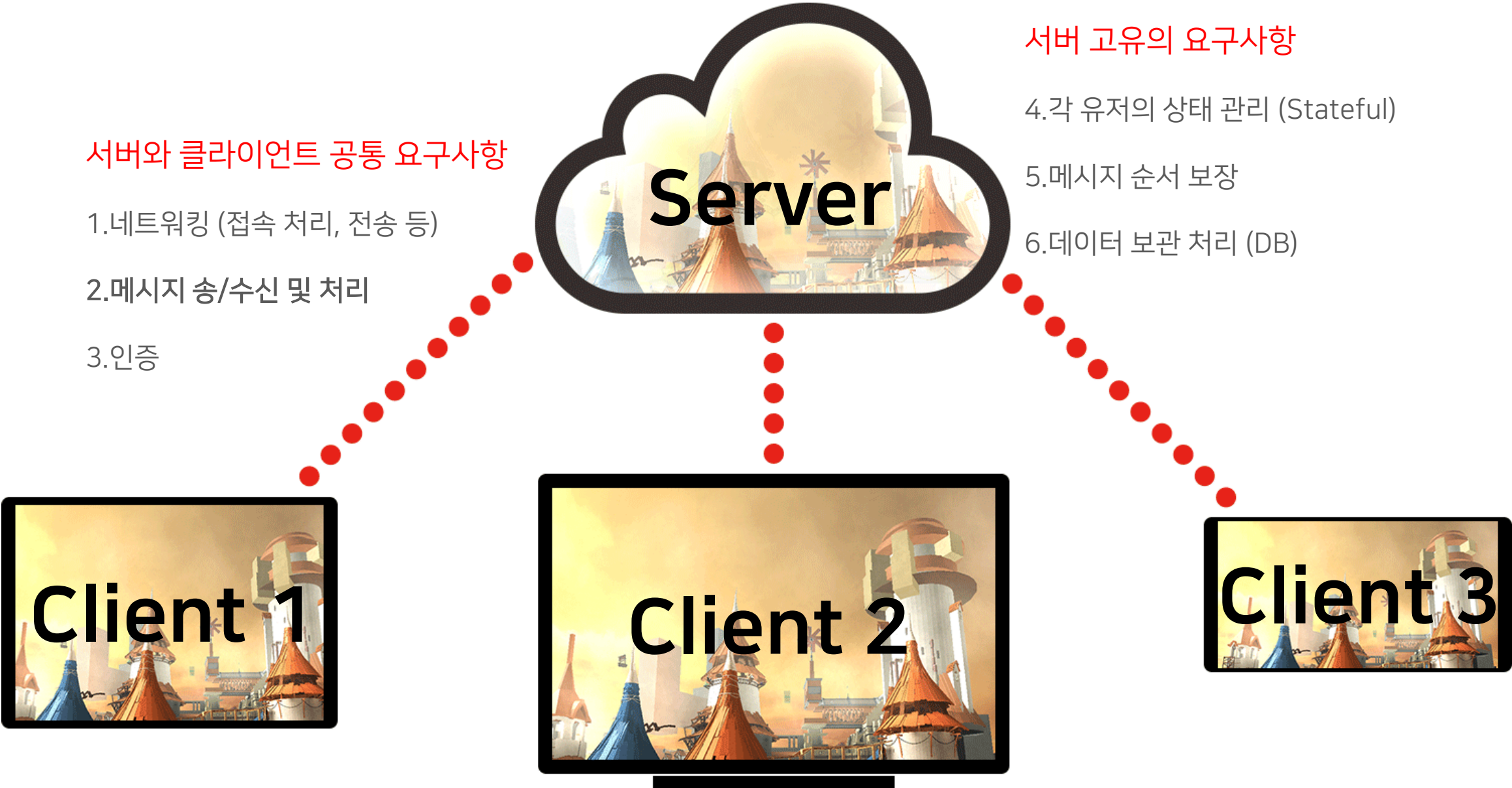
Client 2

Client 3

클라이언트-서버 모델

서버와 클라이언트 공통 요구사항

- 1.네트워킹 (접속 처리, 전송 등)
- 2.메시지 송/수신 및 처리
- 3.인증



Server

서버 고유의 요구사항

- 4.각 유저의 상태 관리 (Stateful)
- 5.메시지 순서 보장
- 6.데이터 보관 처리 (DB)

Client 1

Client 2

Client 3

클라이언트-서버 모델

서버와 클라이언트 공통 요구사항

- 1.네트워킹 (접속 처리, 전송 등)
- 2.메시지 송/수신 및 처리
- 3.인증

서버 고유의 요구사항

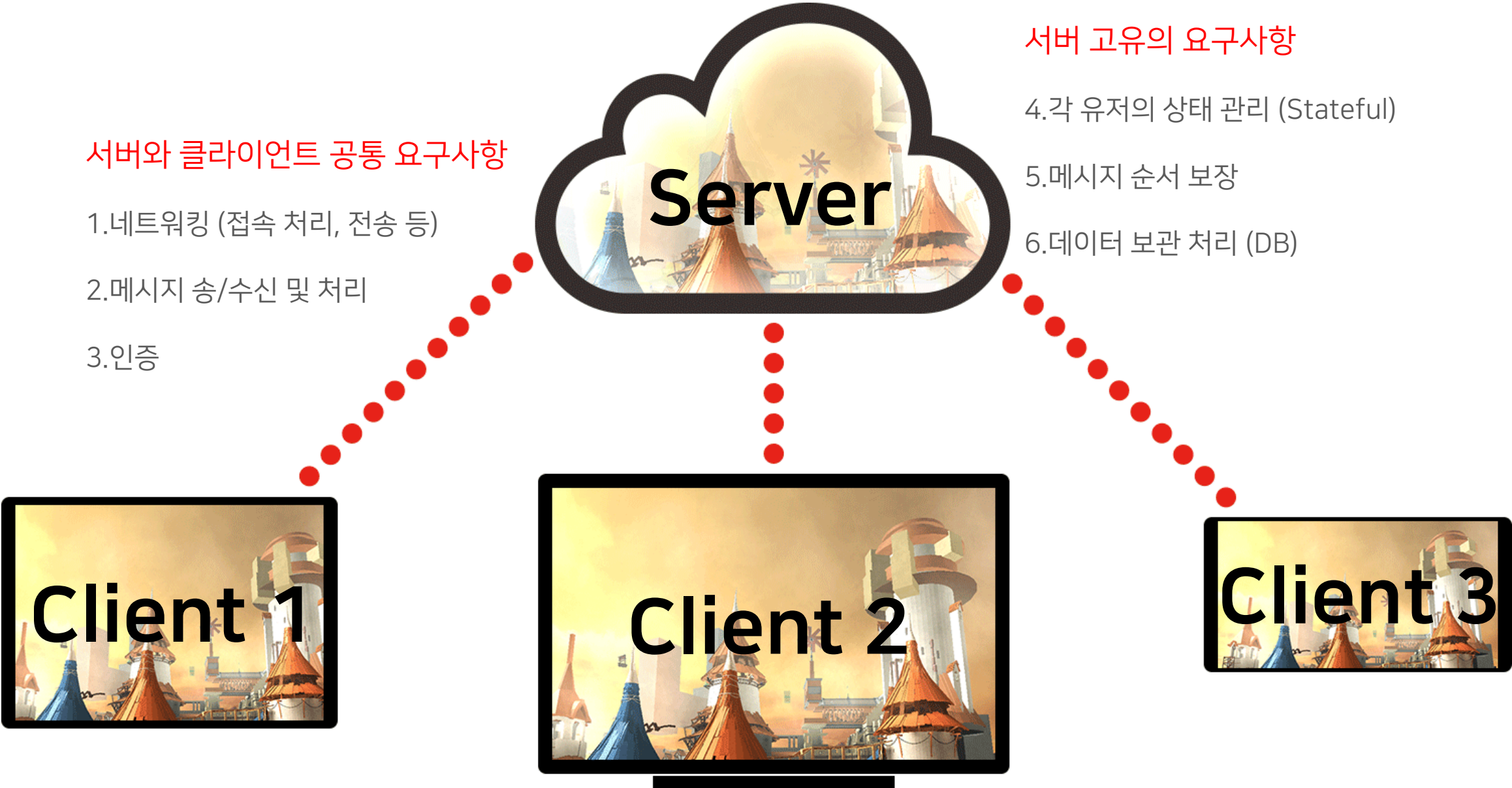
- 4.각 유저의 상태 관리 (Stateful)
- 5.메시지 순서 보장
- 6.데이터 보관 처리 (DB)



클라이언트-서버 모델

서버와 클라이언트 공통 요구사항

- 1.네트워킹 (접속 처리, 전송 등)
- 2.메시지 송/수신 및 처리
- 3.인증



Server

서버 고유의 요구사항

- 4.각 유저의 상태 관리 (Stateful)
- 5.메시지 순서 보장
- 6.데이터 보관 처리 (DB)

Client 1

Client 2

Client 3

GameAnvil

Java 게임 서버 엔진



Java 게임 서버 엔진

최고의
가성비



Java™



Linux

Difficulty: Super Noob



클라이언트 상호 작용

2> 메시지 수신

“클라이언트1이 서버에 방을 생성”

1> 메시지 송신

“방 생성”



클라이언트 상호 작용

2> 메시지 수신

“클라이언트1이 서버에 방을 생성”

1> 메시지 송신

“방 생성”



Server

3> 메시지 송신

“방에 참여”



4> 메시지 송신

“방에 참여”



클라이언트 상호 작용



클라이언트 상호 작용

2> 메시지 수신

"클라이언트 1 오른쪽으로 2만큼 이동"

1> 메시지 송신

"나 오른쪽으로 2만큼 이동"



클라이언트 상호 작용

2> 메시지 수신

"클라이언트 1 오른쪽으로 2만큼 이동"

3> 메시지 송신 : *다른 유저들에게 상태 전파*

"클라이언트 1 오른쪽으로 2만큼 이동"

1> 메시지 송신

"나 오른쪽으로 2만큼 이동"

4> 메시지 수신

"클라이언트 1이 오른쪽으로 2만큼 이동"



4> 메시지 수신

"클라이언트 1이 오른쪽으로 2만큼 이동"

정말 쉬운 온라인 상호 작용 (Pseudo)

```
1: GameRoom 생성; // GameAnvil에서 Room 기능 제공
```

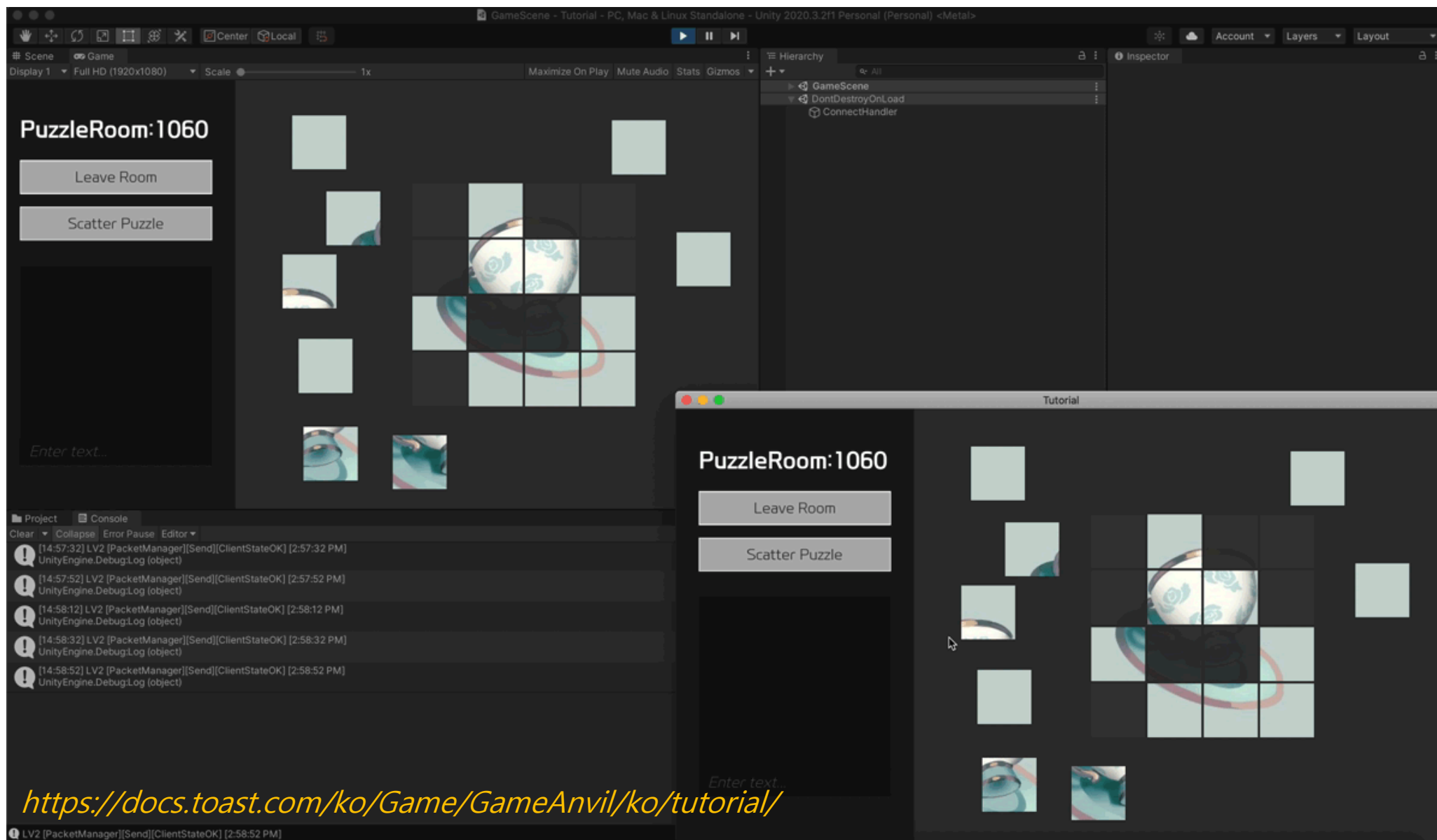

정말 쉬운 온라인 상호 작용 (Pseudo)

```
1: GameRoom 생성; // GameAnvil에서 Room 기능 제공  
2: GameRoom.registerMsg(모든 메시지, RelayAllHandler.class);
```

정말 쉬운 온라인 상호 작용 (Pseudo)

```
1: GameRoom 생성; // GameAnvil에서 Room 기능 제공
2: GameRoom.registerMsg(모든 메시지, RelayAllHandler.class);
3: class RelayAllHandler implements RoomPacketHandler {
4:     void execute(GameRoom room, Packet packet) {
5:         room.sendToAll(packet); //방의 모든 유저에게 메시지 릴레이
6:     }
7: }
```

정말 쉬운 온라인 상호 작용





GameAnvil의 특징

- Java
- Asynchronous based on Fibers

GameAnvil의 특징

- Java
- Asynchronous based on Fibers
- Node : Single Threaded & Lock Free

GameAnvil의 특징

- Java
- Asynchronous based on Fibers
- Node : Single Threaded & Lock Free
- TCP/IP
- Runtime Scalable

GameAnvil의 특징

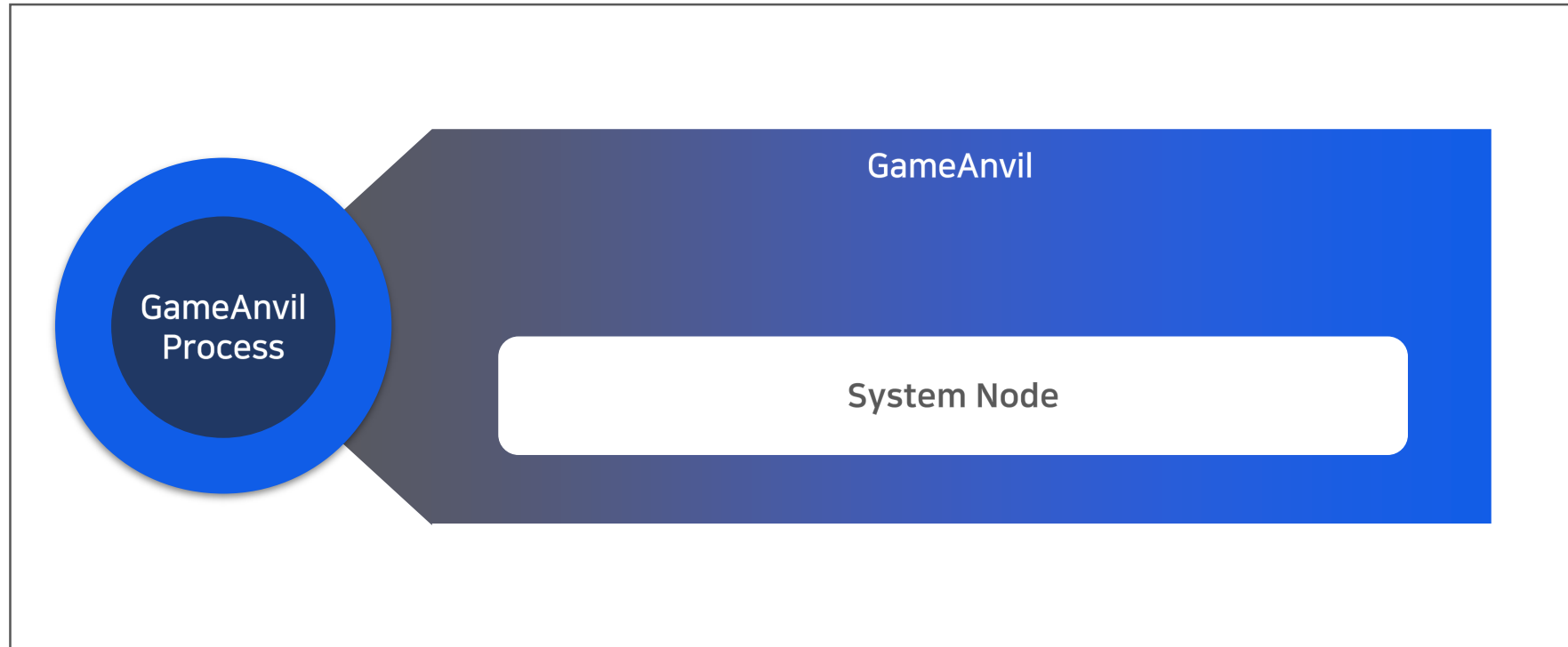
- Java
- Asynchronous based on Fibers
- Node : Single Threaded & Lock Free
- TCP/IP
- Runtime Scalable
- No SPOF (Single Point Of Failure)

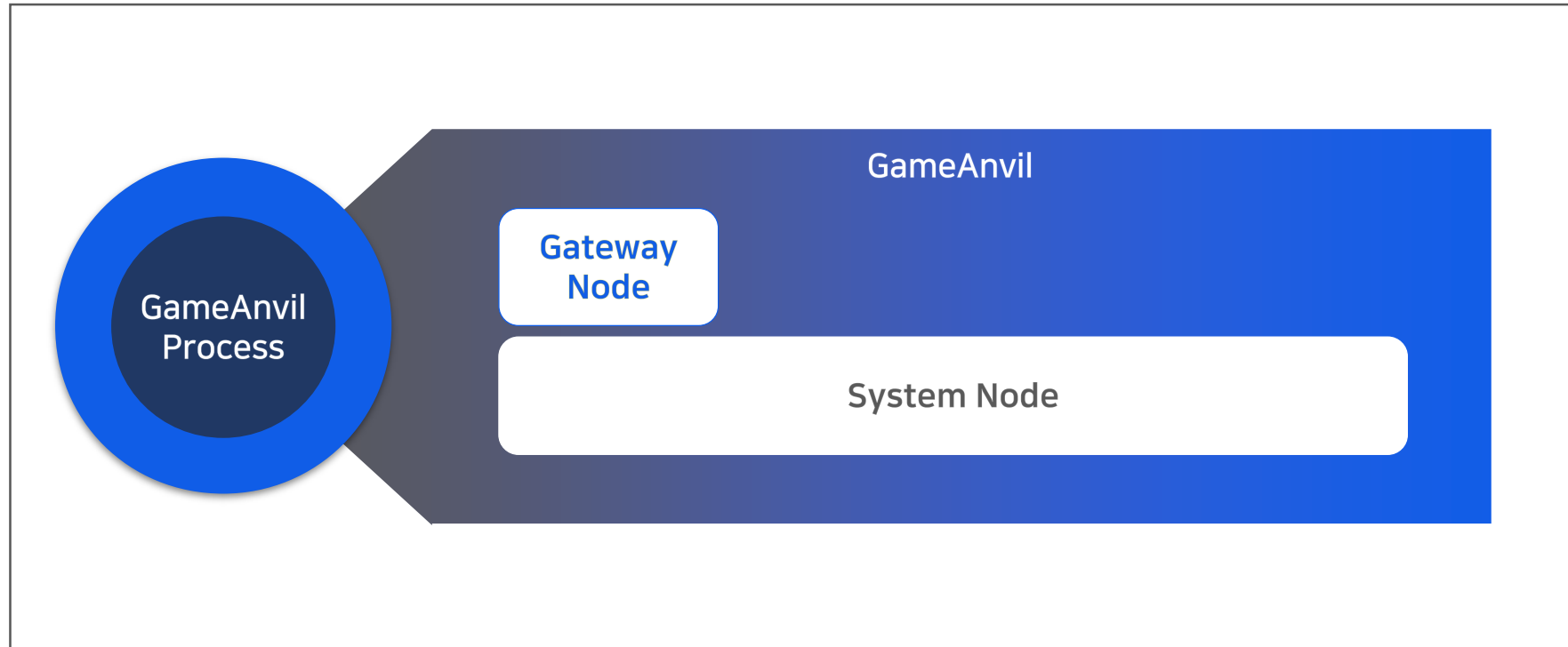
GameAnvil의 특징

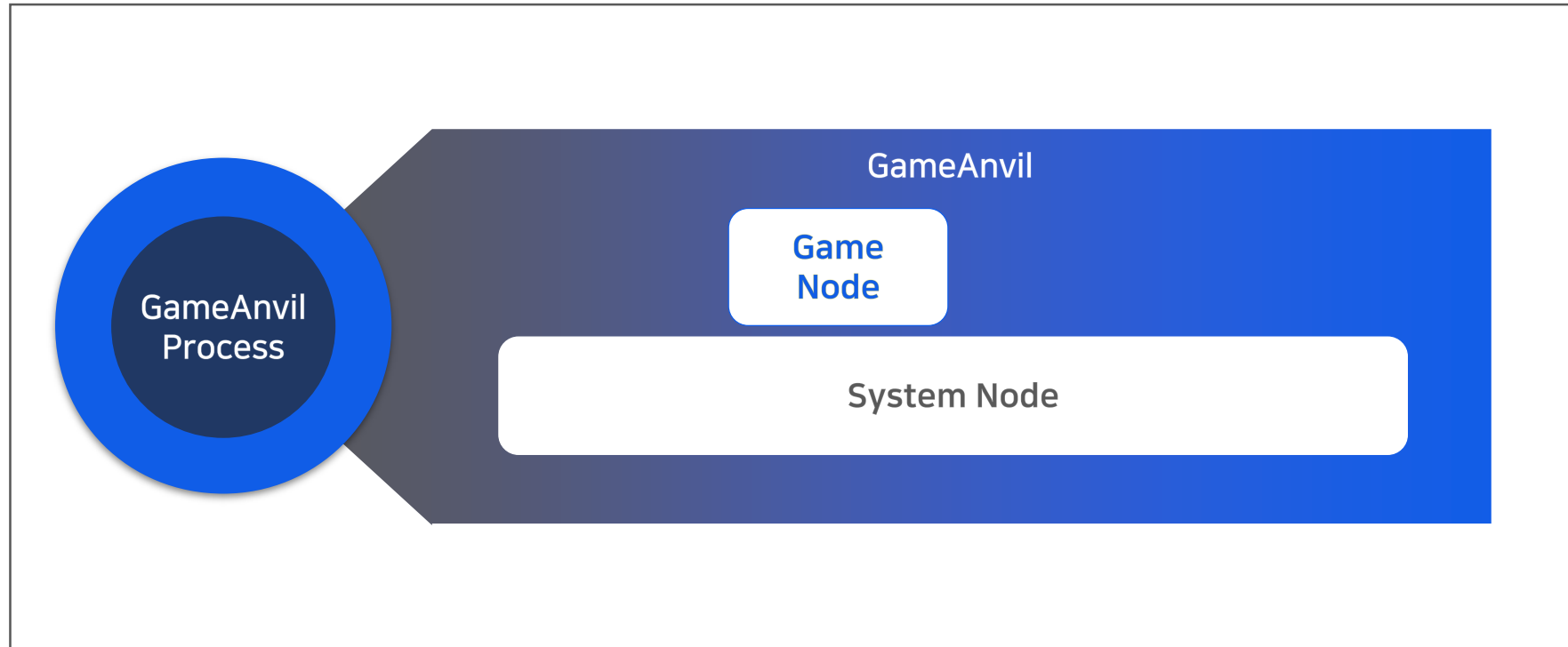
- Java
- Asynchronous based on Fibers
- Node : Single Threaded & Lock Free
- TCP/IP
- Runtime Scalable
- No SPOF (Single Point Of Failure)
- Client Connector (Unity, TypeScript)

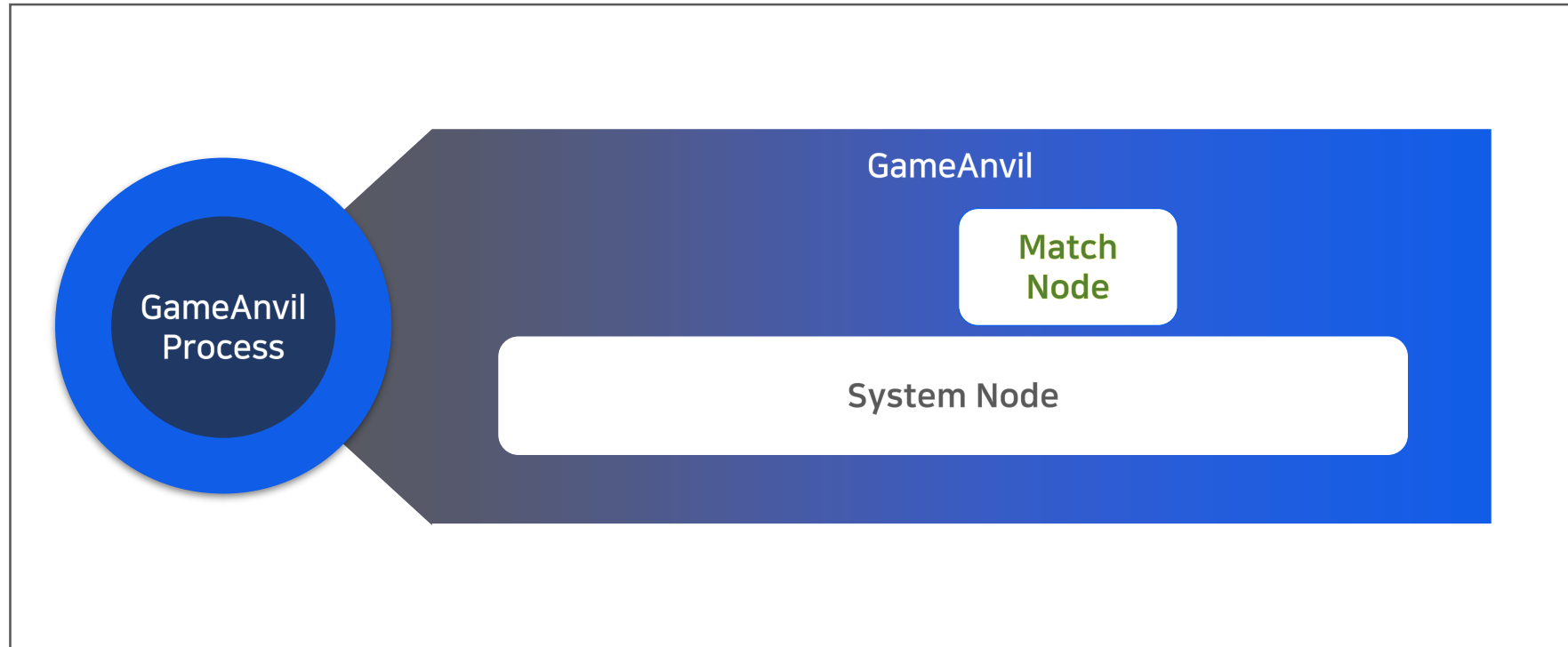
GameAnvil의 특징

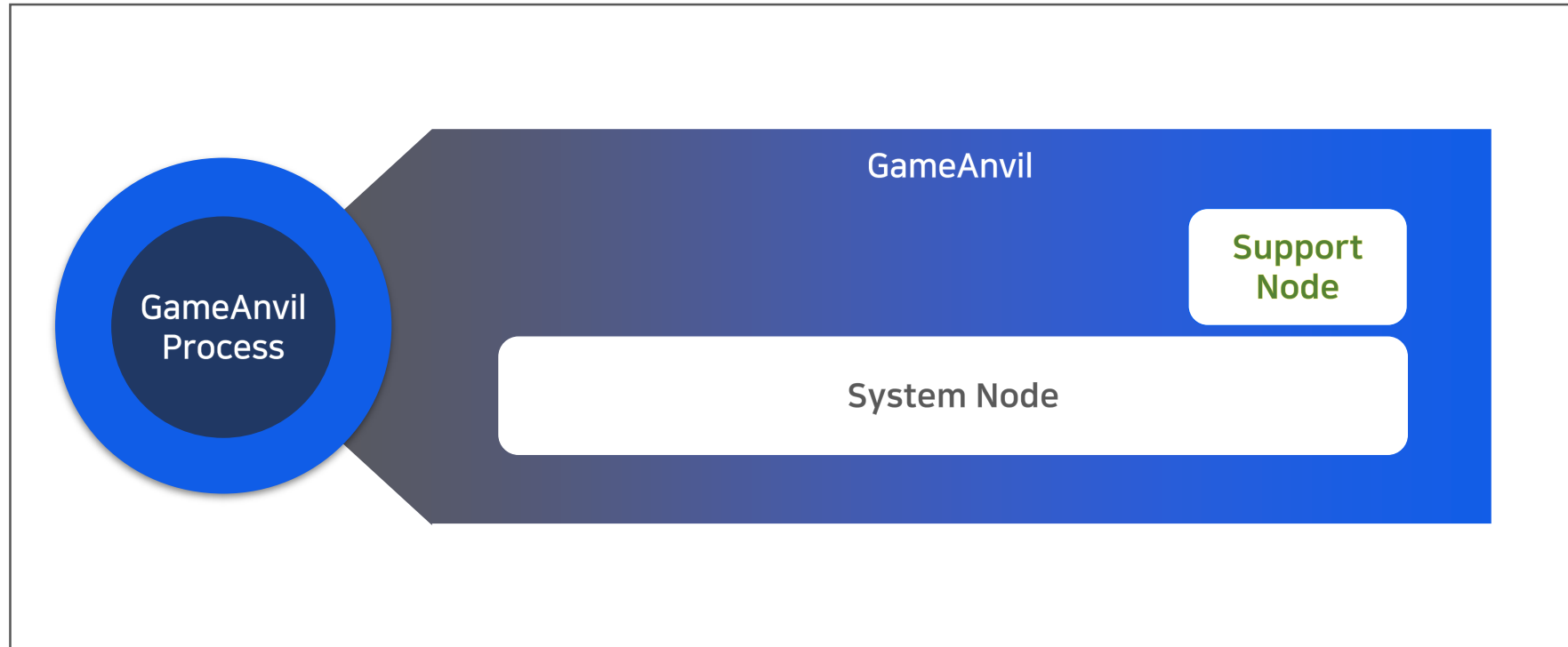
- Java
- Asynchronous based on Fibers
- Node : Single Threaded & Lock Free
- TCP/IP
- Runtime Scalable
- No SPOF (Single Point Of Failure)
- Client Connector (Unity, TypeScript)
- Test support - GameHammer



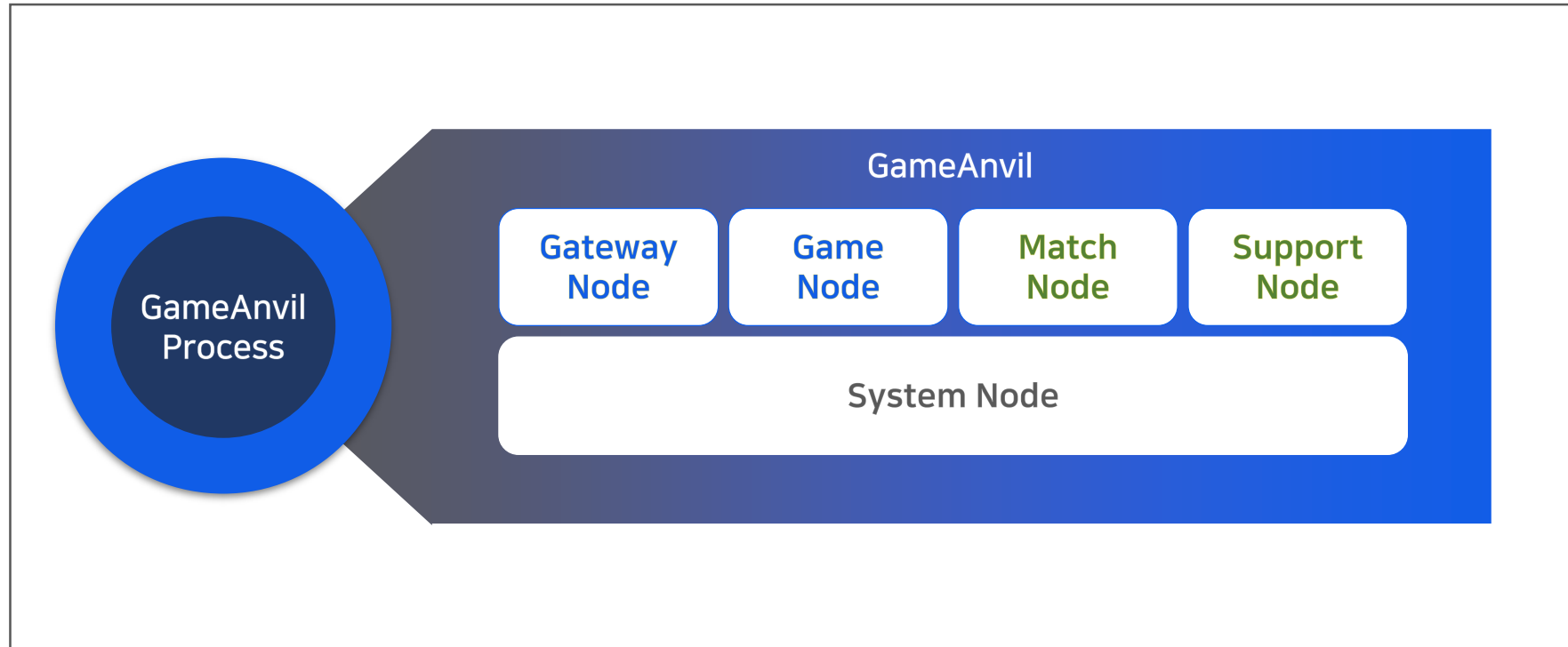




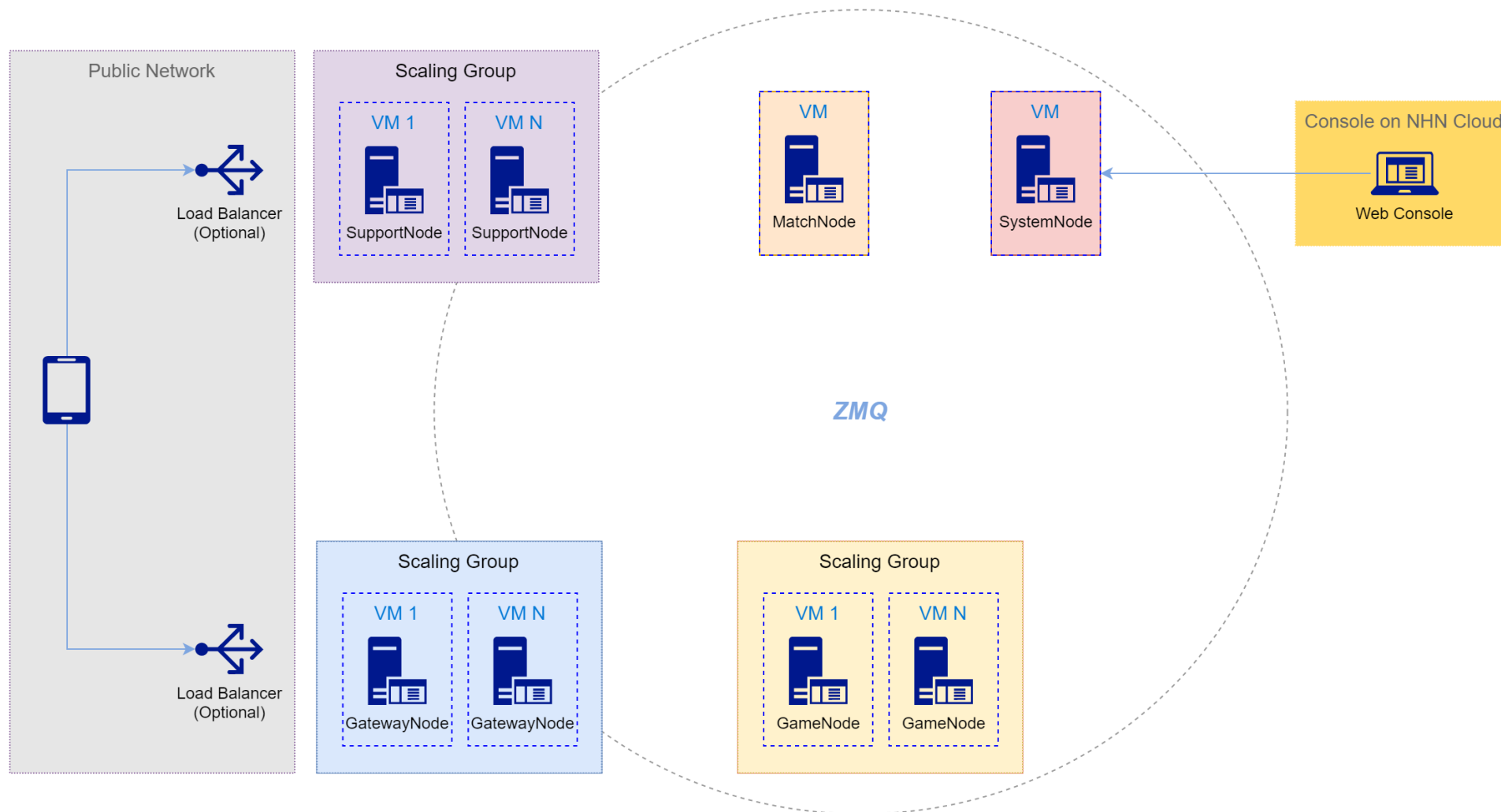




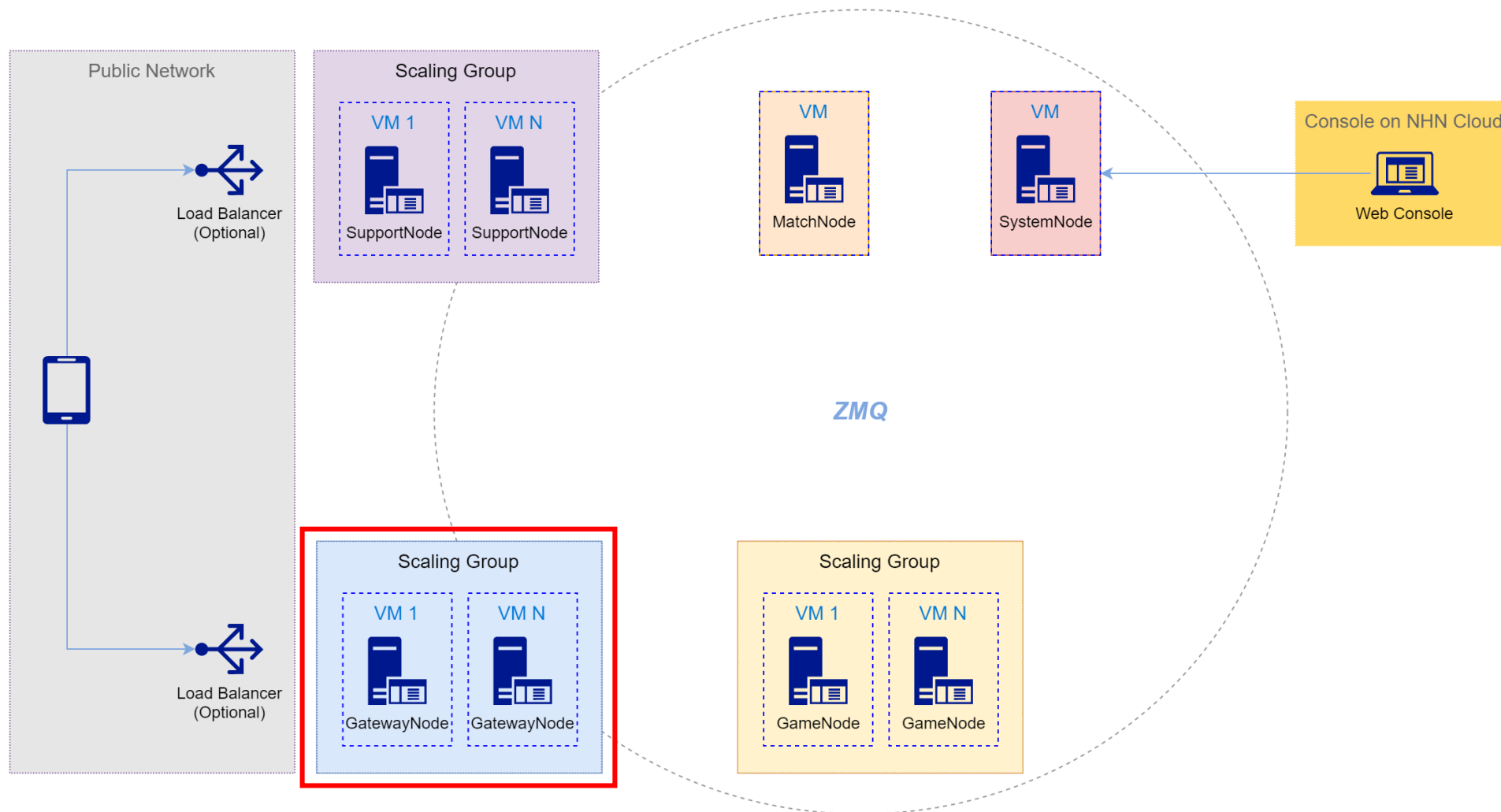
Node



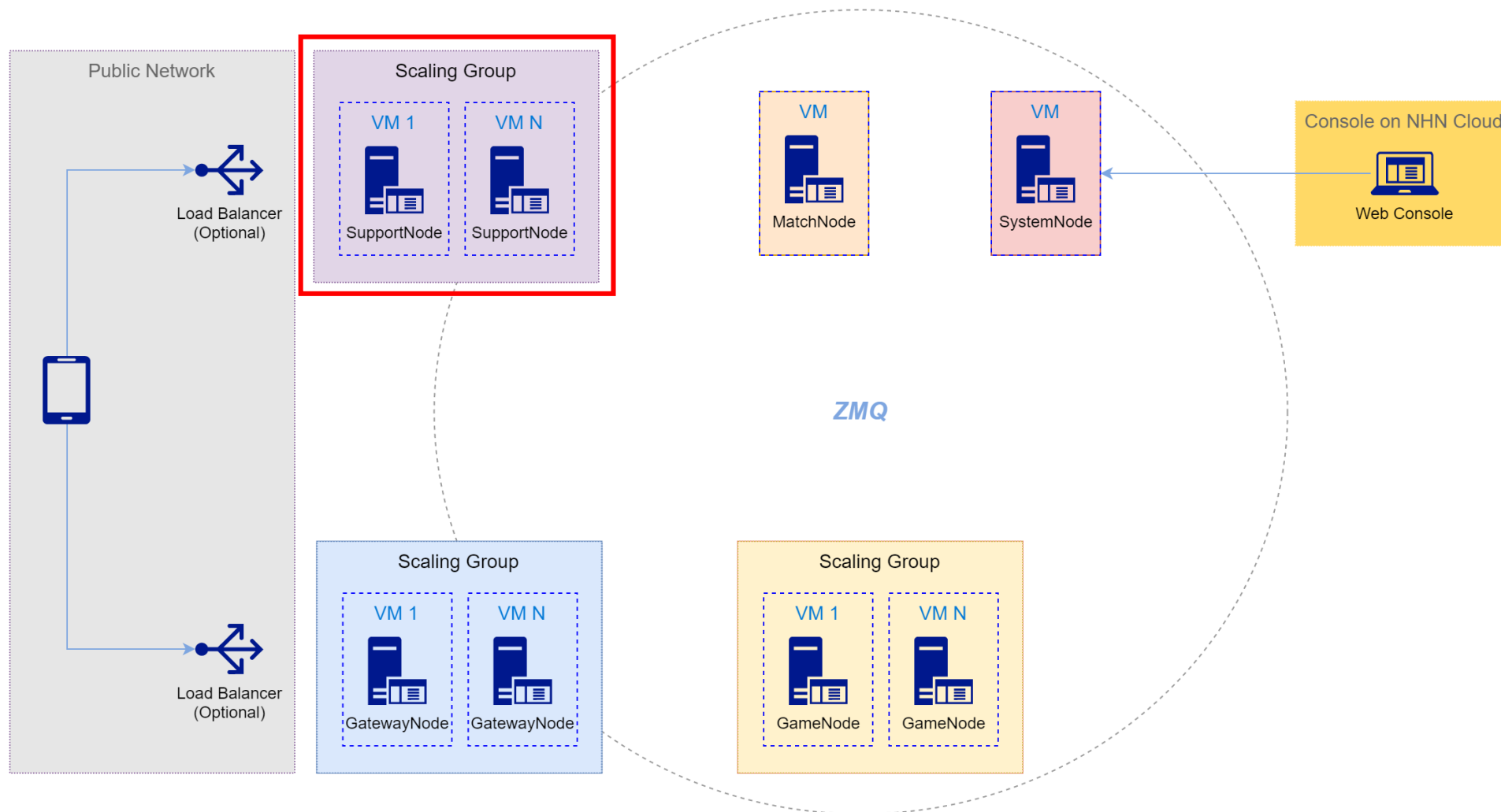
Basic Architecture on Nodes



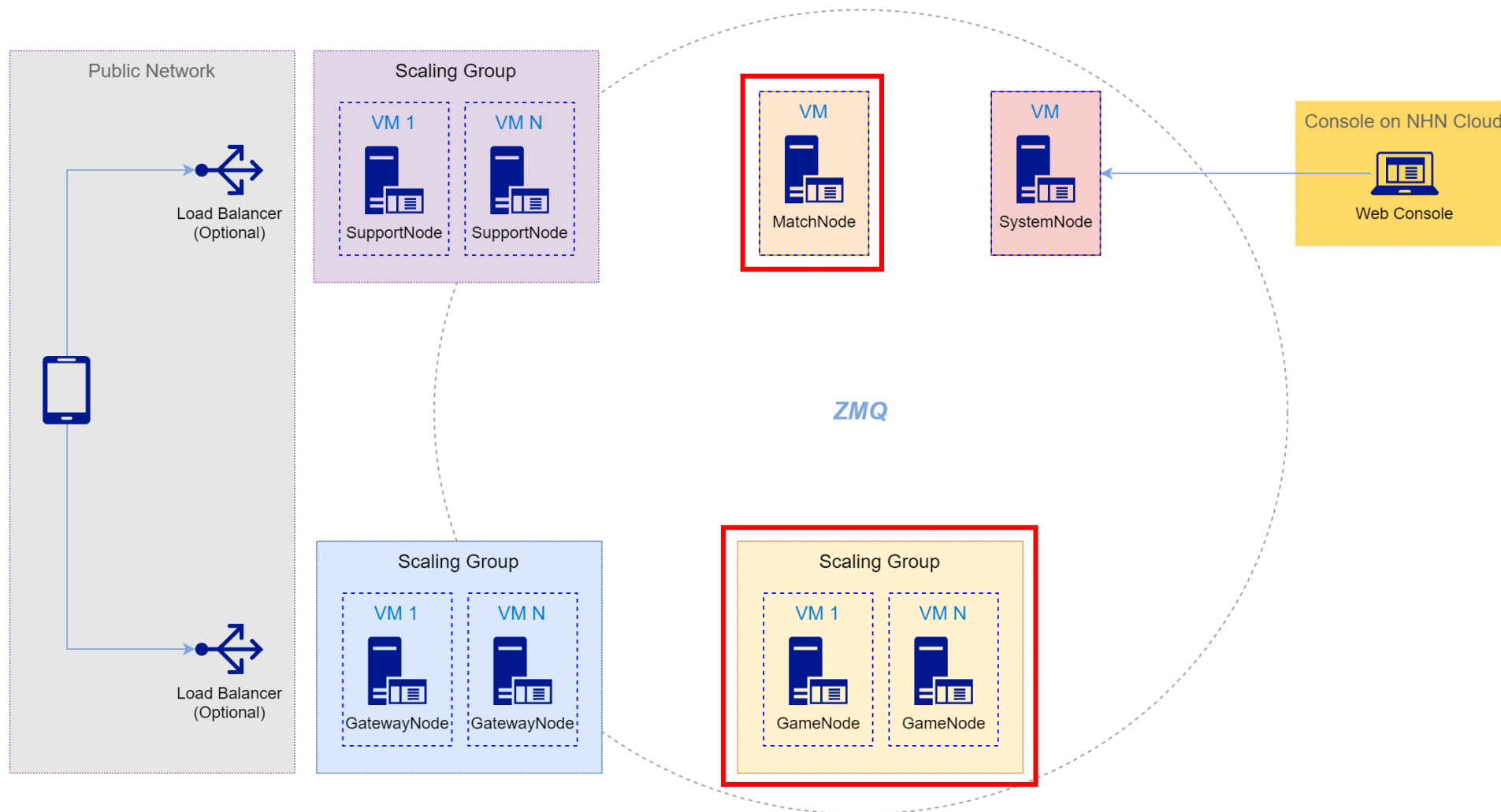
Basic Architecture on Nodes



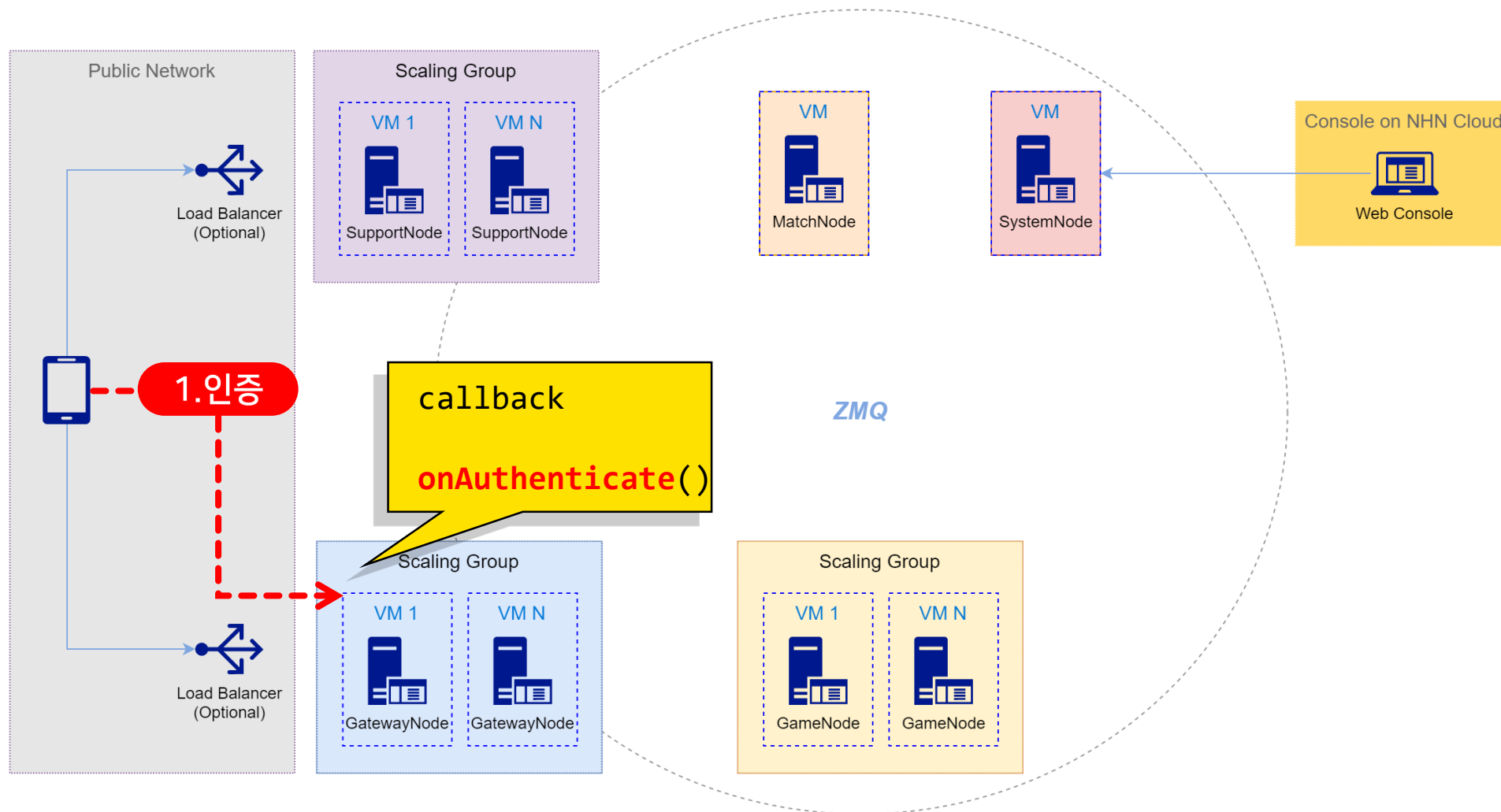
Basic Architecture on Nodes



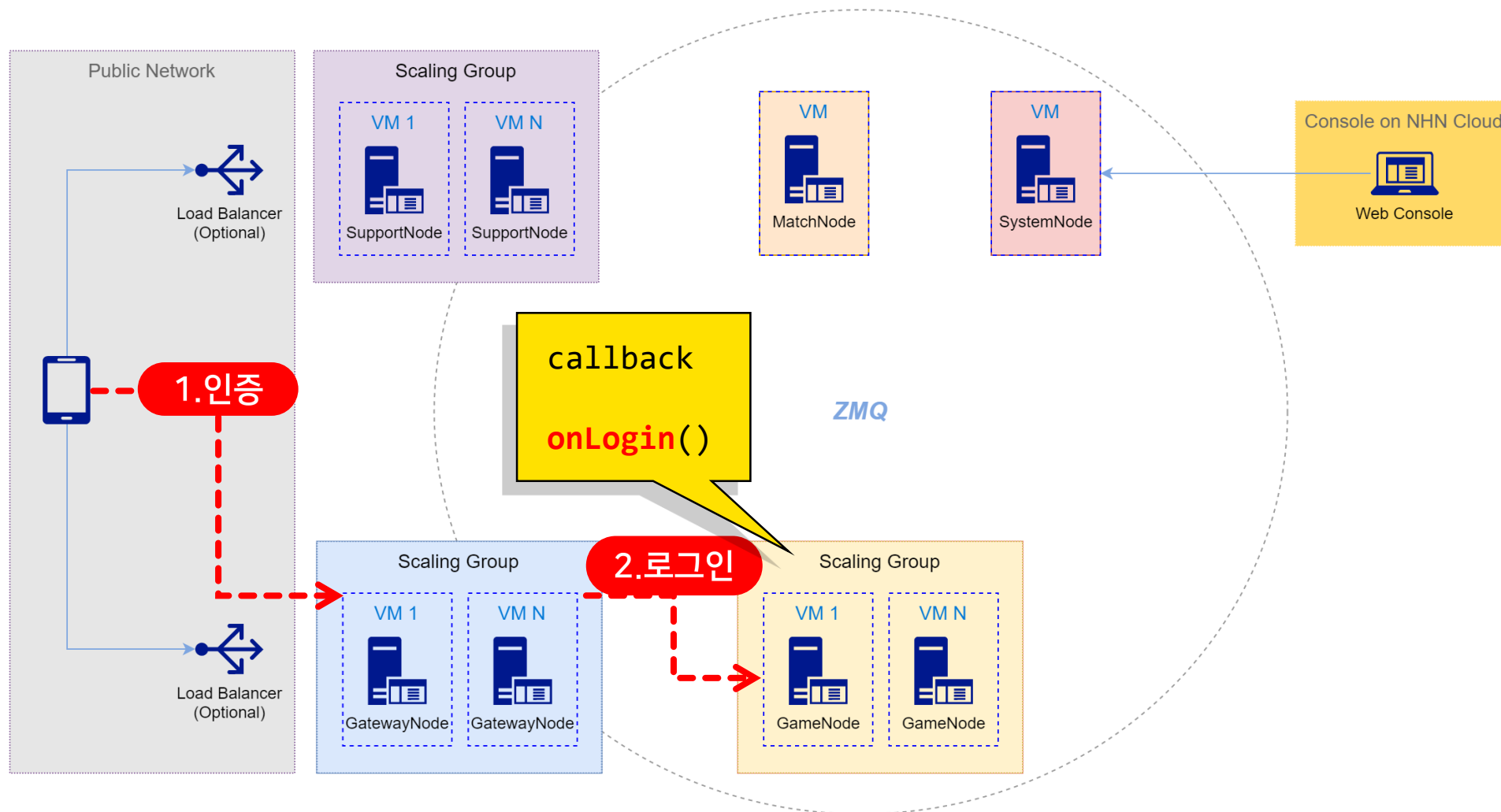
Basic Architecture on Nodes



인증 콜백



로그인 콜백



순차적 코드의 간결함

```
boolean onLogin() {  
    http.get("https://gameplatform.toast.com"); // blocking the fiber  
    ... // 그리고 URL에서 획득한 정보 처리  
  
    selectFromDB(); // blocking the fiber  
    ... // 그리고 DB에서 읽어온 값으로 작업  
}
```

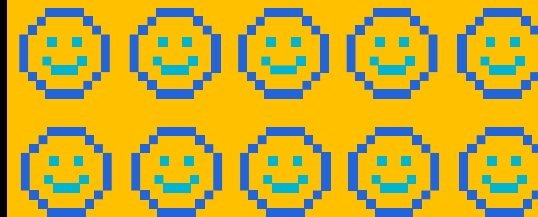
유저 매치 메이킹



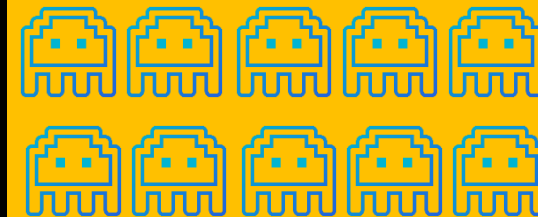
비슷한 실력끼리 매칭!

매칭된 게임

챌린저



브론즈



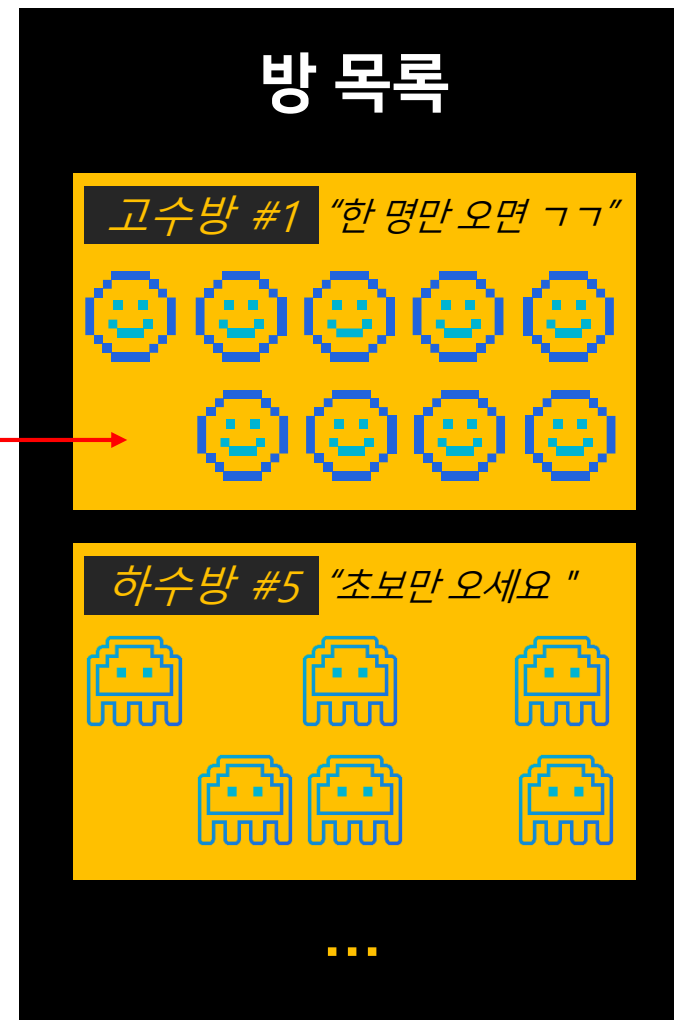
...

"리그 오브 레전드"와 같은 매칭 방식

룸(Room) 매치 메이킹

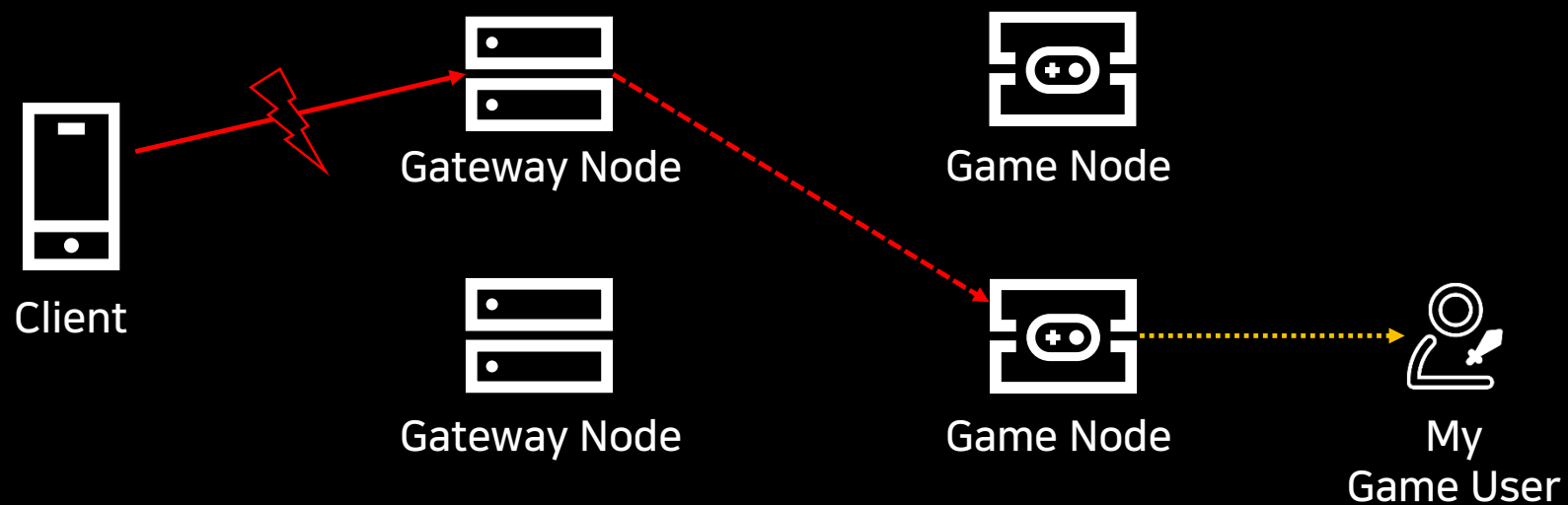


바로 플레이 가능한 방으로 매칭



"한게임 포커"와 같은 매칭 방식

게임 세션 복구

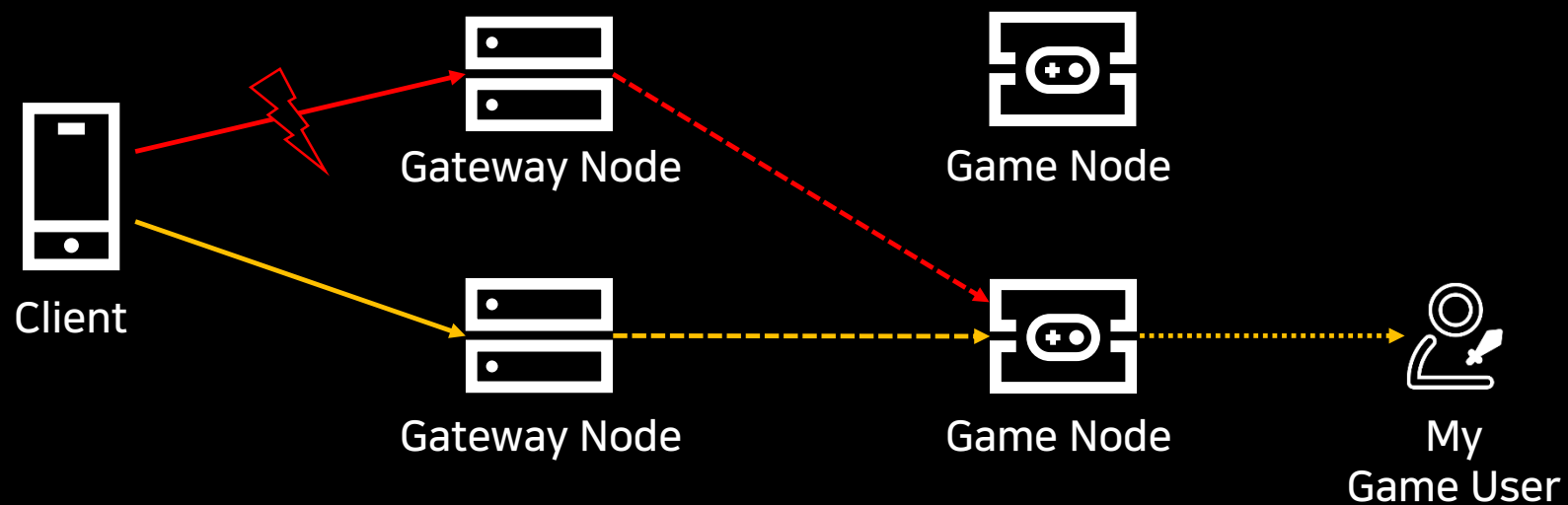


단절된 접속



정리된 서버간 세션

게임 세션 복구



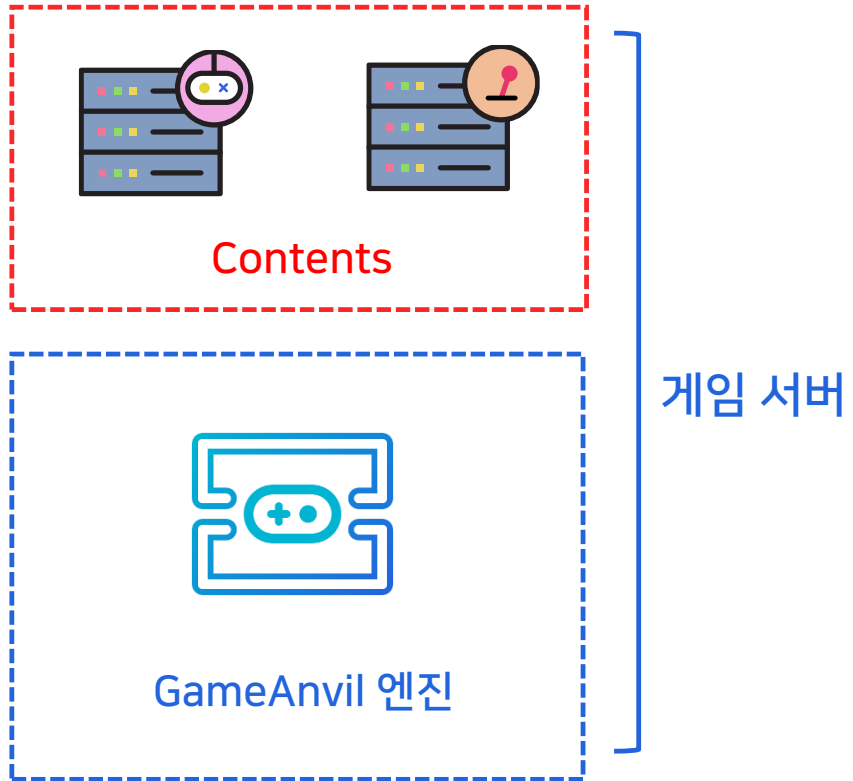
— 단절된 접속

- - - 정리된 서버간 세션

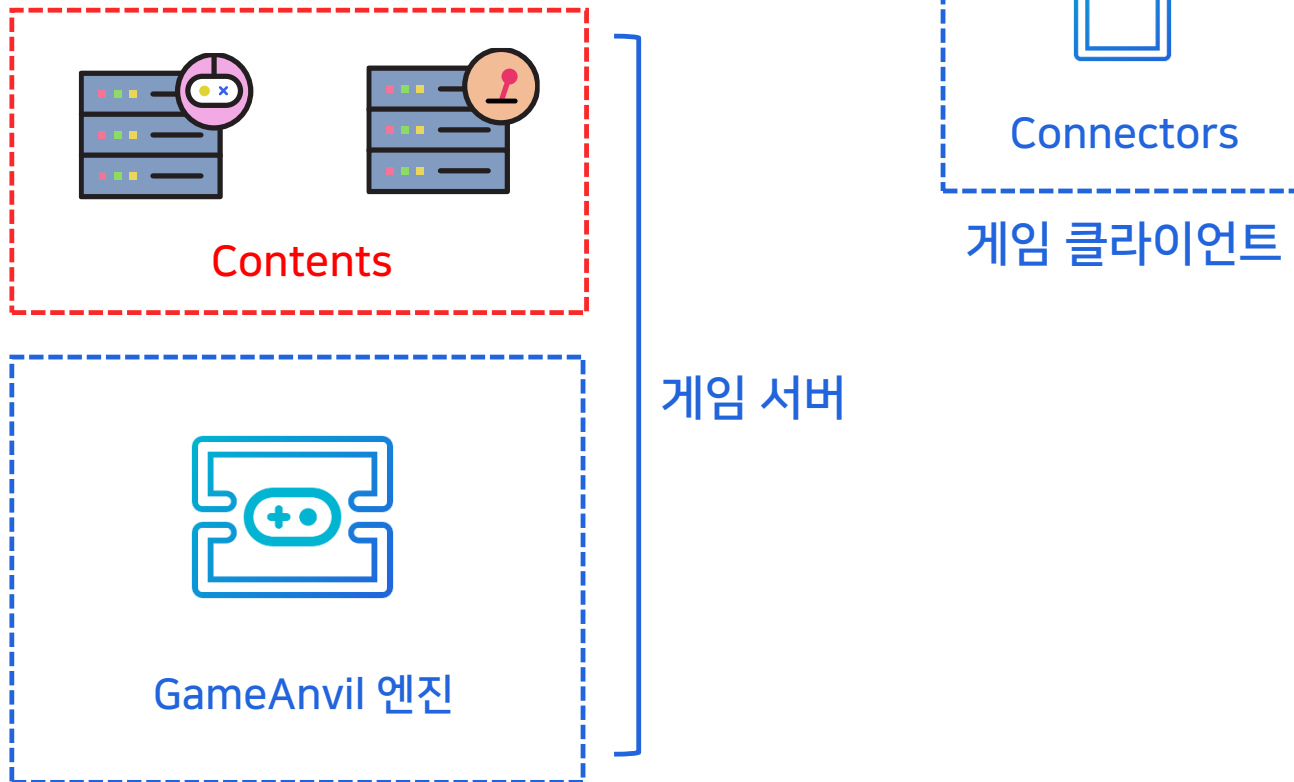
— 재접속

- - - 새로운 서버간 세션

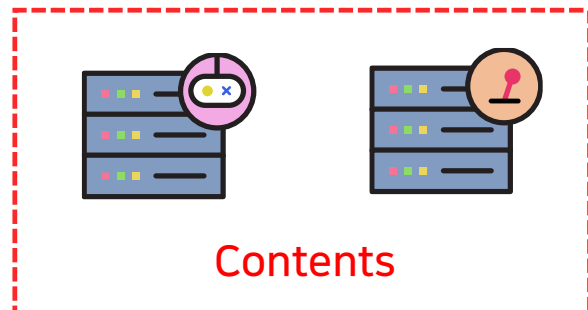
GameAnvil System



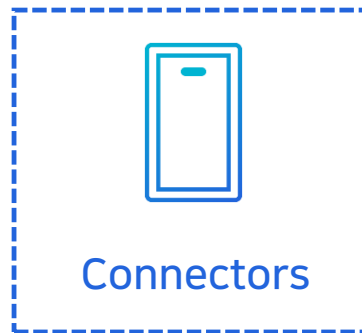
GameAnvil System



GameAnvil System



게임 서버



게임 클라이언트

테스트 시스템



게임 서버, 개발 다음 이야기

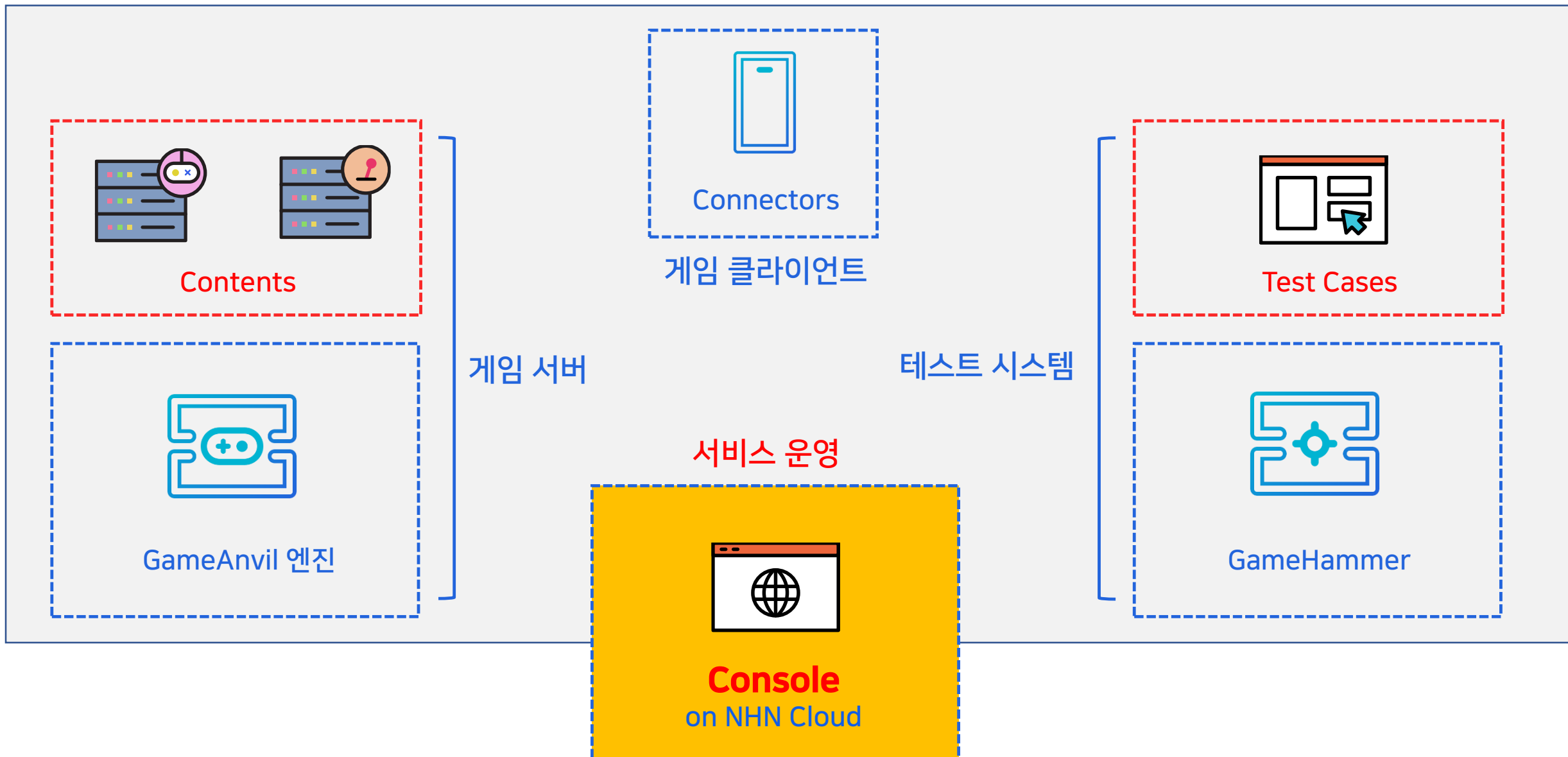
Deploy Minecraft in One-Click

Launch a Minecraft application on Vultr with a few simple clicks!

[Deploy Minecraft](#)[Learn more](#)

GameAnvil on NHN Cloud

GameAnvil System on NHN Cloud



모니터링

운영

서버

Audit

테스트

관리

배포 파일

서비스

▼

구동현황

갱신 시각 : 2022-05-18 14:32

↺ 새로고침

Default 배포 파일 ? : gameanvil-rps-DEV-1.3.0-jdk8_cherry.jar

엔진 버전 ? : 1.2.0

ERROR
2

RUNNING
4

STANDBY
2

TRANSIT
1

PATCHING
1

not STANDBY
1

+ 서버 생성

▼ 전체보기

서버명 ▼

검색어 입력

🔍 검색

다중 선택

전체 11개

서버 필터 :

🗪

☰

뷰 옵션

Kevin_Server1	Kevin_Server2	Kevin_Server3	Kevin_Server4	Kevin_Server5
Kevin_Server6	Kevin_Server7	Kevin_Server8	Kevin_Server9	Kevin_Server10
Kevin_Server11				

서버 관리

모니터링

운영

서버

Audit

테스트

관리

배포 파일

서비스



구동현황

갱신 시각 : 2022-05-18 14:32

새로고침

Default 배포 파일 ? : gameanvil-rps-DEV-1.3.0-jdk8_cherry.jar | 엔진 버전 ? : 1.2.0

ERROR
2RUNNING
4STANDBY
2TRANSIT
1PATCHING
1not STANDBY
1

+ 서버 생성

▼ 전체보기

서버명 ▼

검색어 입력

🔍 검색

다중 선택

전체 11개

서버 필터 :



뷰 옵션

Kevin_Server1

Kevin_Server2

Kevin_Server3

Kevin_Server4

Kevin_Server5

Kevin_Server6

Kevin_Server7

Kevin_Server8

Kevin_Server9

Kevin_Server10

Kevin_Server11

서버 생성

서버 생성 | 신규 서버를 생성합니다.

서버 기본 정보

서버 이름 *

GameAnvilServer

MEMO

For Service

배포 파일 *

gameanvil-rps-DEV-1.3.0-jdk8_kevin.jar,...

등록

서버 수량

3

오토스케일 사용 ?

노드 구성 ?

GATEWAY4개

MATCH1개

GAME3개

a(1)

a

n

v

GAME2개

adfshn__(75)

채널사용안함

채널사용안함

GAME4개

afds(53)

채널사용안함

sdfg

채널사용안함

GAME

노드 추가

GATEWAY

MATCH

GAME

SUPPORT

서비스 추가

b(2)

dfgh_(65)

인스턴스 ?

인스턴스 타입

m2.c8m16

태그

태그

데이터 ✕

서버 생성

서버 기본 정보

서버 이름 *

GameAnvilServer

MEMO

For Service

배포 파일 *

gameanvil-rps-DEV-1.3.0-jdk8_kevin.jar,...

등록

서버 수량

3

오토스케일 사용 ?



서버 생성

노드 구성 ?

GATEWAY

4 개

MATCH

1 개

GAME

3 개

a(1)

a

n

v

GAME

2 개

adfshn__(75)

채널사용안함

채널사용안함

GAME

4 개

afds(53)

채널사용안함

sdfg

채널사용안함

GAME

노드 추가

GATEWAY

MATCH

GAME

SUPPORT

서비스 추가

b(2)

dfgh_(65)

서버 생성

인스턴스 ?	인스턴스 타입	m2.c4m8 ▼
태그	태그	데이터 ✕

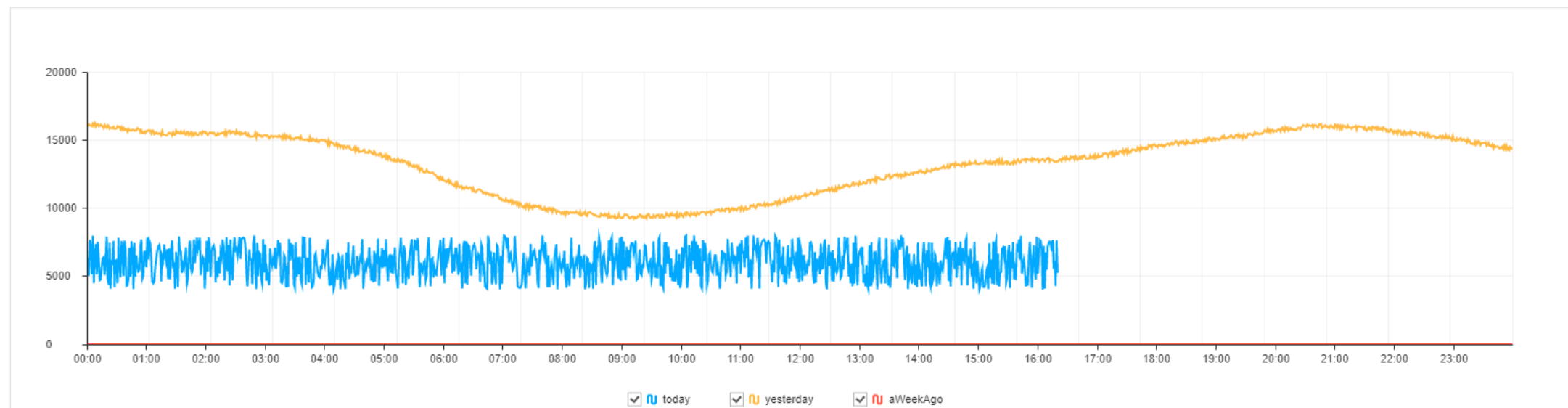
모니터링

모니터링 관리 Audit

대시보드	유저 분포 모니터링	시스템 모니터링	인스턴스 모니터링	노드 모니터링
머신 ? 2/3 모니터링	인스턴스 ? 2/3 모니터링	유저 ? 0	룸 ? 0	세션 ? 0
GATEWAY ? 4/4 모니터링	GAME ? 2/2 모니터링	SUPPORT ? 4/4 모니터링	MATCH ? 1/1 모니터링	

동시 접속자 변화 그래프 GAME 노드 기준 ? 2021-05-14 16:21 기준 갱신

서비스 : 전체 필터



Auto-Scale

오토스케일 그룹 생성

그룹 이름

서버 이름을 입력하세요

MEMO

최소 서버 *

대

최대 서버 *

대

시작 서버 *

대

오토스케일 설정 *

타입	스케일 단위 ?	대기 시간 ?	조건1	*	조건2	상태 지속 시간	연산
증설 ▼	2 ▲▼ 대	5 ▲▼ 분	CPU ▼	>	70 ▲▼ %	5 ▲▼ 분	-
증설 ▼	2 ▲▼ 대	5 ▲▼ 분	Memory ▼	>	70 ▲▼ %	5 ▲▼ 분	and ▼
감축 ▼	2 ▲▼ 대	5 ▲▼ 분	유저 수 ▼	<	70 ▲▼ %	5 ▲▼ 분	-
감축 ▼	2 ▲▼ 대	5 ▲▼ 분	Memory ▼	<	70 ▲▼ %	5 ▲▼ 분	or ▼
+ 행 추가							

• 설정된 조건이 충족되는 경우 서버가 자동으로 오토스케일 됩니다.

Auto-Scale

오토스케일 그룹 생성

그룹 이름

서버 이름을 입력하세요

MEMO

최소 서버 *

대

최대 서버 *

대

시작 서버 *

대

오토스케일 설정 *

타입	스케일 단위 ?	대기 시간 ?	조건1	*	조건2	상태 지속 시간	연산
증설 ▼	2 ▲▼ 대	5 ▲▼ 분	CPU ▼	>	70 ▲▼ %	5 ▲▼ 분	-
증설 ▼	2 ▲▼ 대	5 ▲▼ 분	Memory ▼	>	70 ▲▼ %	5 ▲▼ 분	and ▼
감축 ▼	2 ▲▼ 대	5 ▲▼ 분	유저 수 ▼	<	70 ▲▼ %	5 ▲▼ 분	-
감축 ▼	2 ▲▼ 대	5 ▲▼ 분	Memory ▼	<	70 ▲▼ %	5 ▲▼ 분	or ▼

+ 행 추가

- 설정된 조건이 충족되는 경우 서버가 자동으로 오토스케일 됩니다.

Safe Pause

GameAnvil ALPHA > 운영

URL & Appkey

사용자 가이드

모니터링 운영 관리 Audit 테스트

Safe Pause

실행

이력

Safe Pause | 출발지 노드의 유저와 룸 정보를 도착지 노드로 옮기고 노드를 정지할 수 있습니다. 무정지 점검이 가능합니다.

노드 선택

점검 실행

강제 종료

출발지 노드 2 도착지 노드 2 2021-11-11 11:32 기준 갱신

새로고침

노드 구분	노드 ID	인스턴스 이름	호스트 이름	서비스 ID	서비스 이름	채널 ID	유저 수	룸 수	노드 상태	-
출발지 노드	167700100	Game	tdsa-dedi-t1803	1	RPSGame	ch1	1020	510	● READY	-
도착지 노드	167700101	Game	tdsa-dedi-t1803	1	RPSGame	ch1	0	0	● READY	-
출발지 노드	167700102	Game	tdsa-dedi-t1803	1	RPSGame	ch2	180	90	● READY	-
도착지 노드	167700103	Game	tdsa-dedi-t1803	1	RPSGame	ch2	0	0	● READY	-

4 건 1/1 page

<< < 1 > >>

인스턴스

배포 파일

성능 테스트

로그 수집 인스턴스 | 테스트 결과 로그를 수집할 인스턴스를 설정합니다.

+ 인스턴스 등록

▶ 시작

■ 중지

...

2022-05-06 17:36 기준 갱신 새로고침

호스트 이름	IP 주소	Logging Port	머신 상태	인스턴스 상태	
I-AM-YOUR-SVR-0	10.162.3.111	18888	● 연결됨	● 동작	테스트 결과 리포트로 이동

GameHammer | GameHammer 인스턴스 목록입니다.

+ 인스턴스 등록

↑ 배포

배포 파일 업로드

...

2022-05-06 17:36 기준 갱신 새로고침

☐	인스턴스 이름	호스트 이름	IP 주소	GameAnvil Agent Port	GameHammer 빌드 업로드 경로	테스트 상태	배포 상태
☐	hammerAtRPSGame	I-AM-YOUR-SVR-1	10.162.3.128	19080	bin/hammer/hammerAtRPSGame		● 등록 전
☐	hammerAtTapTap	I-AM-YOUR-SVR-1	10.162.3.128	19080	bin/hammer/hammerAtTapTap		● 등록 전
☐	hammerAtRPSGame	I-AM-YOUR-SVR-2	10.162.3.132	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료
☐	hammerAtRPSGame	I-AM-YOUR-SVR-4	10.162.3.134	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료
☐	hammerAtRPSGame	I-AM-YOUR-SVR-5	10.162.3.135	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료
☐	hammerAtRPSGame	I-AM-YOUR-SVR-6	10.162.3.136	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료
☐	hammerAtRPSGame	I-AM-YOUR-SVR-7	10.162.3.137	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료
☐	hammerAtRPSGame	I-AM-YOUR-SVR-8	10.162.3.138	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료
☐	hammerAtRPSGame	I-AM-YOUR-SVR-9	10.162.3.139	19080	bin/hammer/hammerAtRPSGame	테스트 중	● 배포 완료

성능 테스트 | 테스트 목록입니다.

+ 테스트 등록

🕒 2022-05-26 17:36 기준 갱신 ↺ 새로고침 | 🔍

테스트명	수정자	수정 시간	시작 시간	테스트 기간	테스트 유저 수	실패 건수	TPS 평균	PPS 평균	상태
20차 테스트	ji*****@nhn.com	2022-05-24T16:39:25+09:00		00:50:00	5,000				
19차 테스트	ji*****@nhn.com	2022-05-23T16:39:25+09:00	2022-05-23T17:39:25+09:00	00:50:00	5,000	0	51,242	82,890	진행 중
18차 테스트	ji*****@nhn.com	2022-05-22T16:39:25+09:00	2022-05-22T17:39:25+09:00	00:50:00	5,000	0	51,234	122,567	완료
17차 테스트	ji*****@nhn.com	2022-05-21T16:39:25+09:00	2022-05-21T17:39:25+09:00	00:50:00	5,000	0	50,523	12,113	완료
16차 테스트	ji*****@nhn.com	2022-05-20T16:39:25+09:00	2022-05-20T17:39:25+09:00	00:50:00	5,000	2390	97,789	243,790	강제 종료
15차 테스트	ji*****@nhn.com	2022-05-19T16:39:25+09:00	2022-05-19T18:39:25+09:00	00:50:00	5,000	0	51,653	124,346	완료
14차 테스트	ji*****@nhn.com	2022-05-18T16:39:25+09:00	2022-05-18T16:49:25+09:00	00:50:00	5,000	0	52,456	126,479	완료
13차 테스트	ji*****@nhn.com	2022-04-24T16:39:25+09:00	2022-04-24T16:39:45+09:00	00:50:00	5,000	0	50,494	121,289	완료
12차 테스트	ji*****@nhn.com	2022-04-14T16:29:25+09:00	2022-04-21T17:39:25+09:00	00:50:00	5,000	0	49,980	119,128	완료
11차 테스트	ji*****@nhn.com	2022-03-14T16:39:25+09:00	2022-03-14T16:39:26+09:00	00:50:00	5,000	0	48,908	111,478	완료

C	NHN Cloud	+
🖨	Compute	+
📦	Container	+
🌐	Network	+
📁	Storage	+
🗄	Database	+
🎮	Game	-
	Gamebase	+
	GameAnvil	-
	개요	
	튜토리얼	
	콘솔 사용 가이드	+
	서버 개발 가이드	+
	서버 기본 개념	+
	Unity 개발 가이드	+
	CocosCreator 개발 가이드	+
	테스트 개발 가이드	+
	레퍼런스 프로젝트	+
	릴리즈 노트	+
	오류 코드	

Game > GameAnvil > 튜토리얼

GameAnvil로 게임 서버 쉽게 만들기

이 문서는 GameAnvil을 이용하여 기본적인 서버를 개발하는 과정을 설명합니다. 이를 통해 GameAnvil의 기본 개념, 프로젝트 시작 방법 및 구현 방법 등을 쉽게 익힐 수 있도록 도움을 주고자 합니다. 또한 서버뿐만 아니라 Unity를 이용한 클라이언트도 함께 개발하면서 서버와 클라이언트가 상호 작용하는 모습을 보여주고자 합니다.

최대한 이해를 돕기 위해 실제 플레이가 가능한 멀티플레이 직소 퍼즐 게임을 개발하는 과정을 순서대로 보여줍니다. 글을 읽는 사용자 여러분은 이 문서의 내용을 직접 하나씩 따라해보면 더욱 쉽게 이해할 수 있으리라 믿습니다.

1. 프로젝트 구성

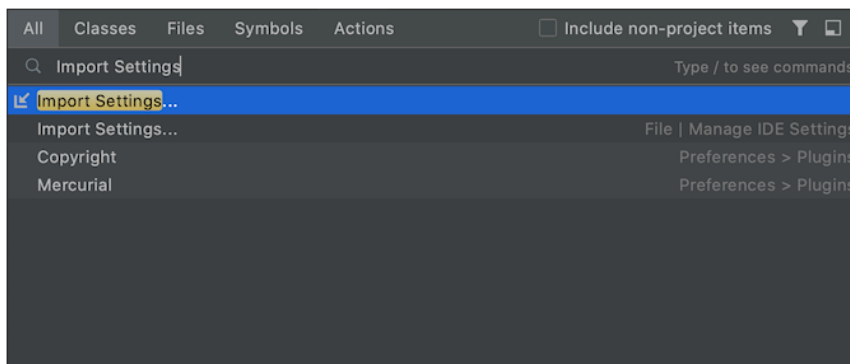
1.1. GameAnvil 프로젝트 구성

다음 링크를 통해서 IntelliJ용 프로젝트 파일 템플릿과 튜토리얼용 프로젝트를 다운로드 받습니다.

[프로젝트 템플릿 다운로드](#)

[튜토리얼용 프로젝트 다운로드](#)

다운로드 받은 템플릿을 적용하기 위해 IntelliJ에서 단축키 **Shift Shift** 로 전체 검색창을 띄운 다음 **Import Settings...** 을 검색합니다.



목차

GameAnvil로 게임 서버 쉽게 만들기

1. 프로젝트 구성

- 1.1. GameAnvil 프로젝트 구성
- 1.2. Unity 프로젝트 구성

2. 서버 구동 및 연결

- 2.1. GameAnvil 서버 구동
- 2.2. 커넥션 핸들러 작성
- 2.3. 서버 연결 코드 추가
- 2.4. 서버 연결 확인

3. Room 및 User생성

- 3.1. 서버 작업

4. 접속 인증

- 4.1. 접속 인증 코드 추가
- 4.2. 접속 인증 확인

5. 로그인

- 5.1. 로그인 코드 추가
- 5.2. 로그인 확인

6. 방 생성 및 참가

- 6.1. 클라이언트 작업
- 6.2. Room 생성 및 참가 테스트

7. 인게임 채팅 구현

- 7.1. 클라이언트 측 전송 구현

고맙습니다.

Cloud

**유연하게, 안전하게
비즈니스에 힘이 되다.**

