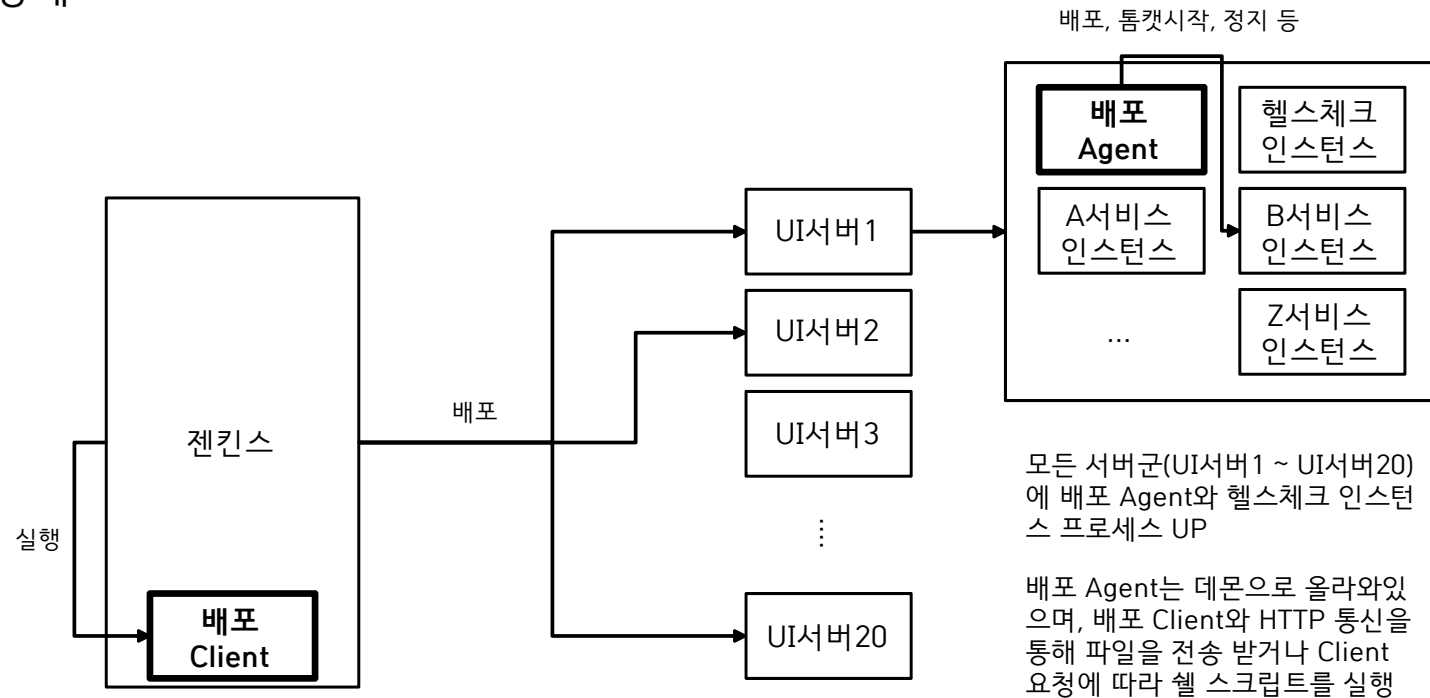


배포툴 관리

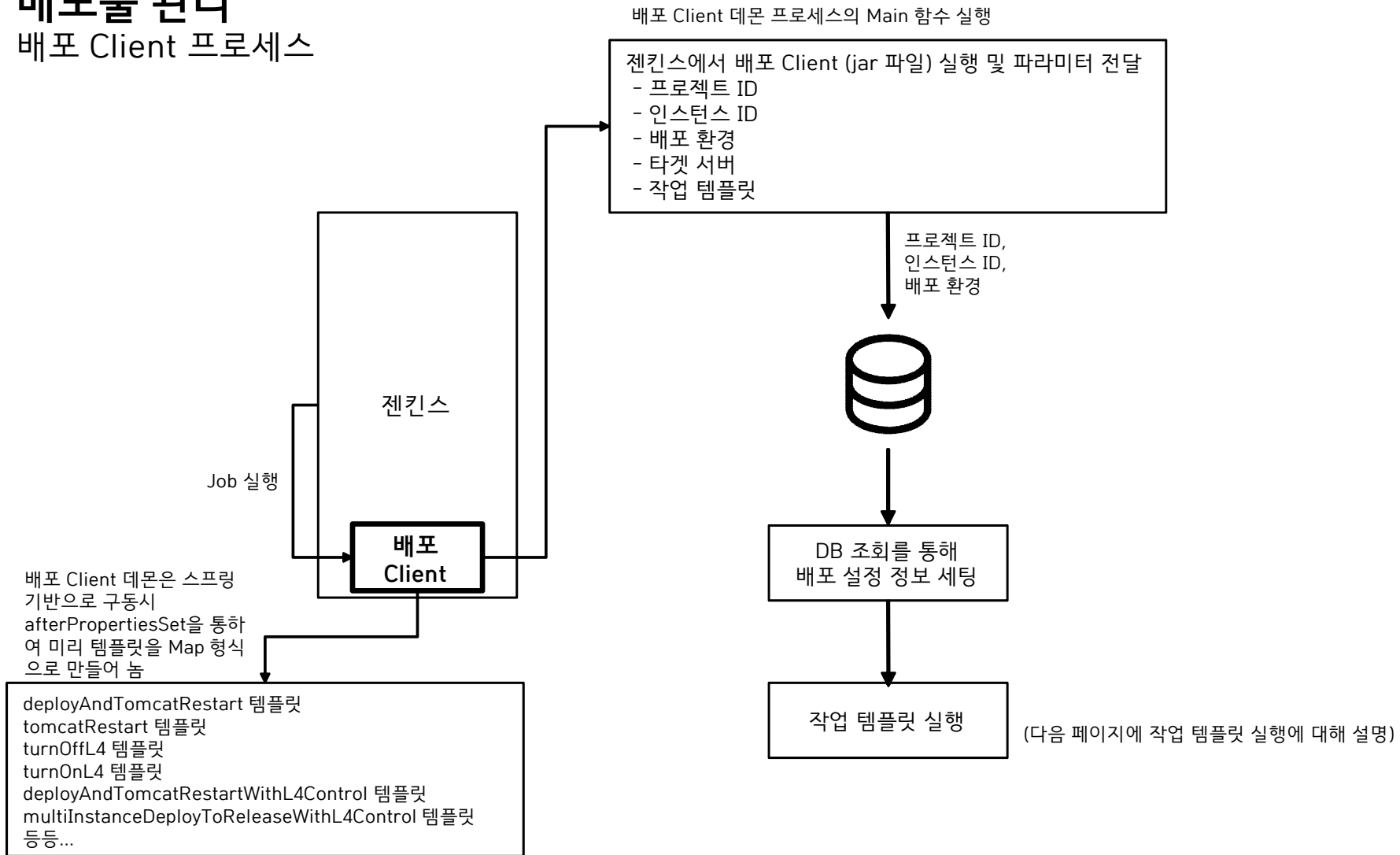
배포 프로세스 명세



jenkins에서 쉘스크립트로 통해 배포 Client(jar 파일)를 실행
이때, 배포할 서버와 어떤 작업(배포, 톰캣 재시작 등)을 실행할
지 배포 Client에 파라미터로 전달

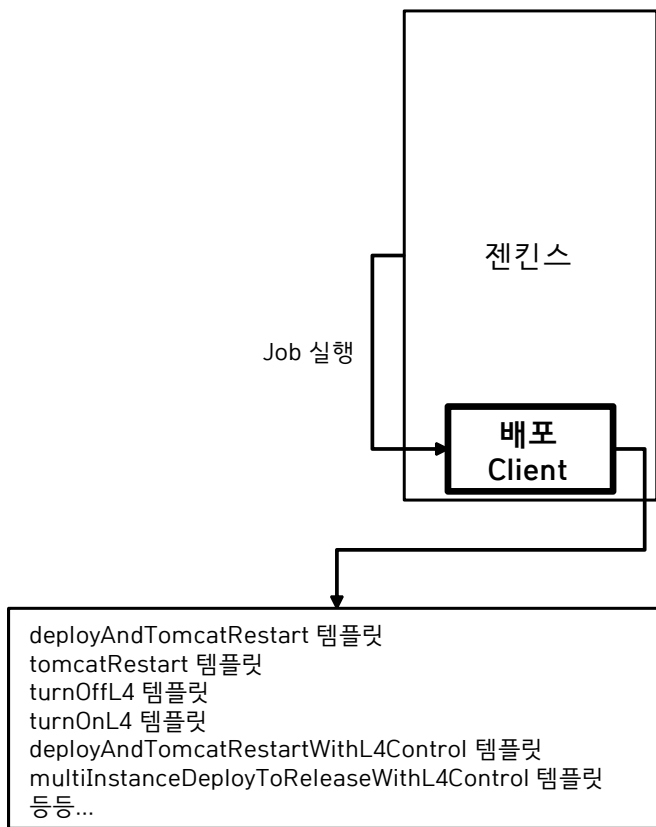
배포툴 관리

배포 Client 프로세스



배포툴 관리

배포 Client 프로세스



템플릿이름과 배포Flow 정보를 Map 형태로 메모리에 저장 (스프링 로드 시점)

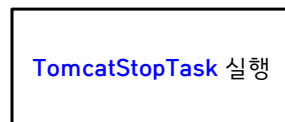
(예시 1)

deployAndTomcatRestart 템플릿에는 아래 3가지 배포 Flow를 가짐

- TomcatStopTask
- ComponentDeployTask
- TomcatStartTask

Jenkins로부터 "deployAndTomcatRestart" 템플릿이 파라미터 값으로 전달되면 메모리에 꺼내서 위 3가지 배포 Flow를 실행

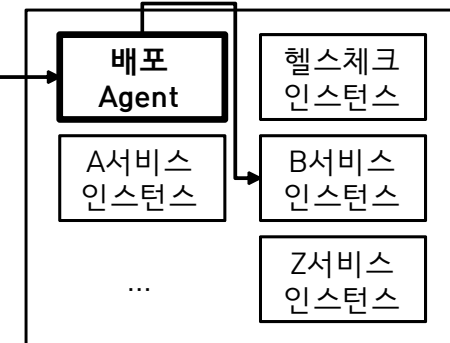
젠킨스 배포 Client



HTTP 통신

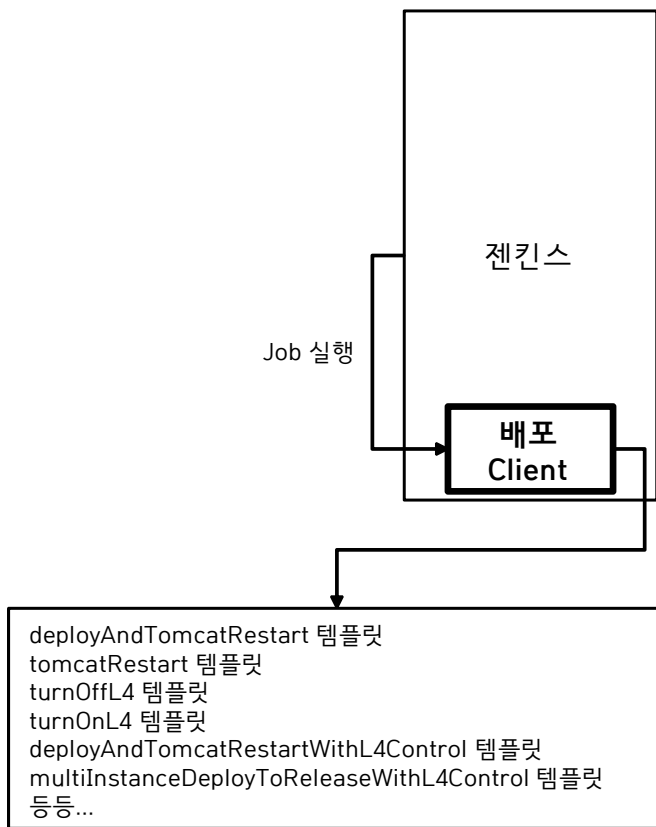
```
{
  TomcatTask,
  stop
}
```

UI서버



배포툴 관리

배포 Client 프로세스



템플릿이름과 배포Flow 정보를 Map 형태로 메모리에 저장 (스프링 로드 시점)

(예시 1)

deployAndTomcatRestart 템플릿에는 아래 3가지 배포 Flow를 가짐

- TomcatStopTask
- **ComponentDeployTask**
- TomcatStartTask

Jenkins로부터 "deployAndTomcatRestart" 템플릿이 파라미터 값으로 전달되면 메모리에 꺼내서 위 3가지 배포 Flow를 실행

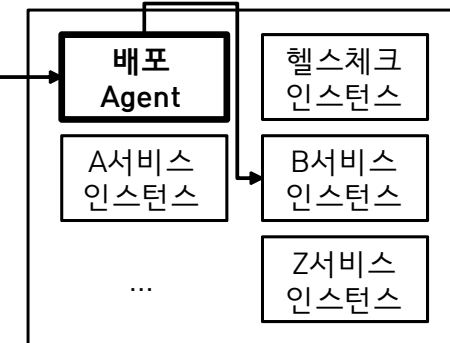
젠킨스 배포 Client

ComponentDeployTask
실행

HTTP 통신

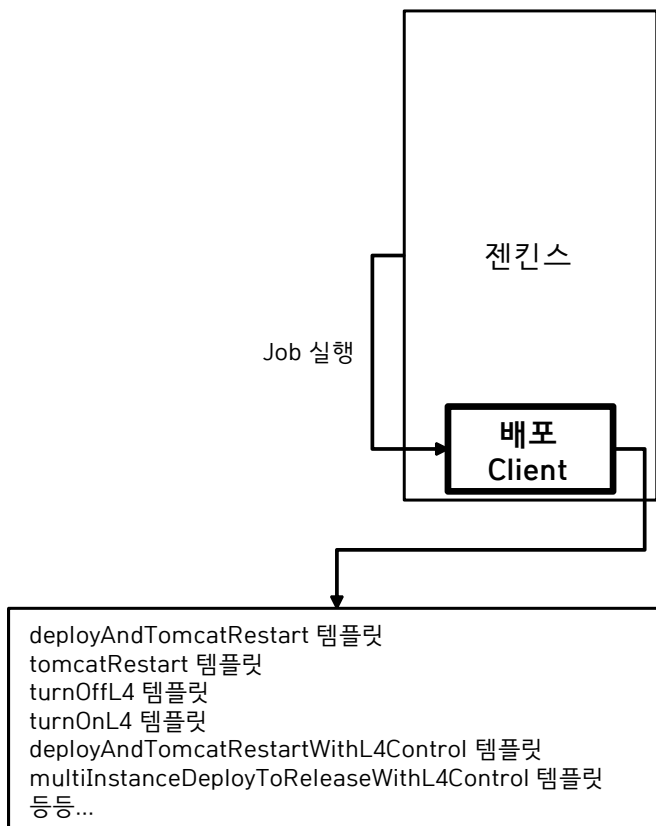
```
{  
  배포 File,  
  인스턴스 ID,  
  componentDeployStart,  
  배포 경로  
}
```

UI서버



배포툴 관리

배포 Client 프로세스



템플릿이름과 배포Flow 정보를 Map 형태로 메모리에 저장 (스프링 로드 시점)

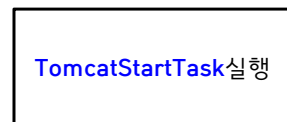
(예시 1)

deployAndTomcatRestart 템플릿에는 아래 3가지 배포 Flow를 가짐

- TomcatStopTask
- ComponentDeployTask
- **TomcatStartTask**

Jenkins로부터 "**deployAndTomcatRestart**" 템플릿이 파라미터 값으로 전달되면 메모리에 꺼내서 위 3가지 배포 Flow를 실행

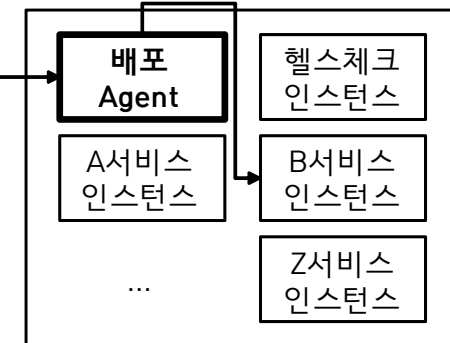
jenkins 배포 Client



HTTP 통신

```
{
  TomcatTask,
  start
}
```

UI서버



"deployAndTomcatRestart" 템플릿 배포 종료

배포툴 관리

배포 Client 프로세스

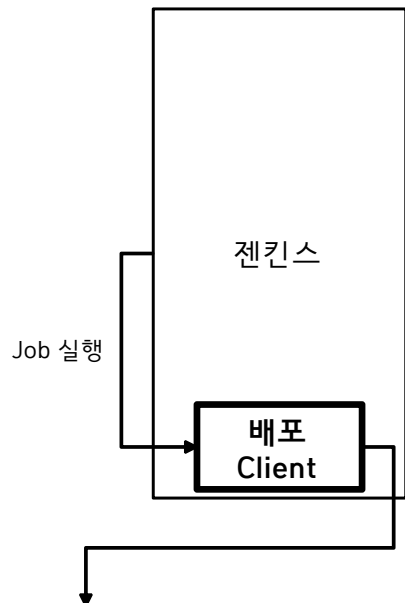
템플릿이름과 배포Flow 정보를 Map 형태로 메모리에 저장 (스프링 로드 시점)

(예시 2)

deployAndTomcatRestartWithL4Control 템플릿

- L4TurnOffTask
- LoopbackStopTask
- TomcatStopTask
- ComponentDeployTask
- TomcatStartTask
- LoopbackStartTask
- L4TurnOnTask

Jenkins로부터 "deployAndTomcatRestart" 템플릿이 파라미터 값으로 전달되면 메모리에 꺼내서 위 3가지 배포 Flow를 실행



deployAndTomcatRestart 템플릿
tomcatRestart 템플릿
turnOffL4 템플릿
turnOnL4 템플릿
deployAndTomcatRestartWithL4Control 템플릿
multiInstanceDeployToReleaseWithL4Control 템플릿
등등...

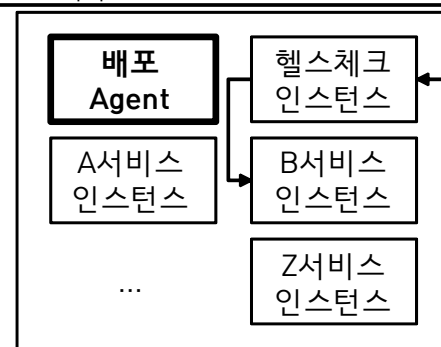
제킨스 배포 Client

L4TurnOffTask 실행

HTTP 통신

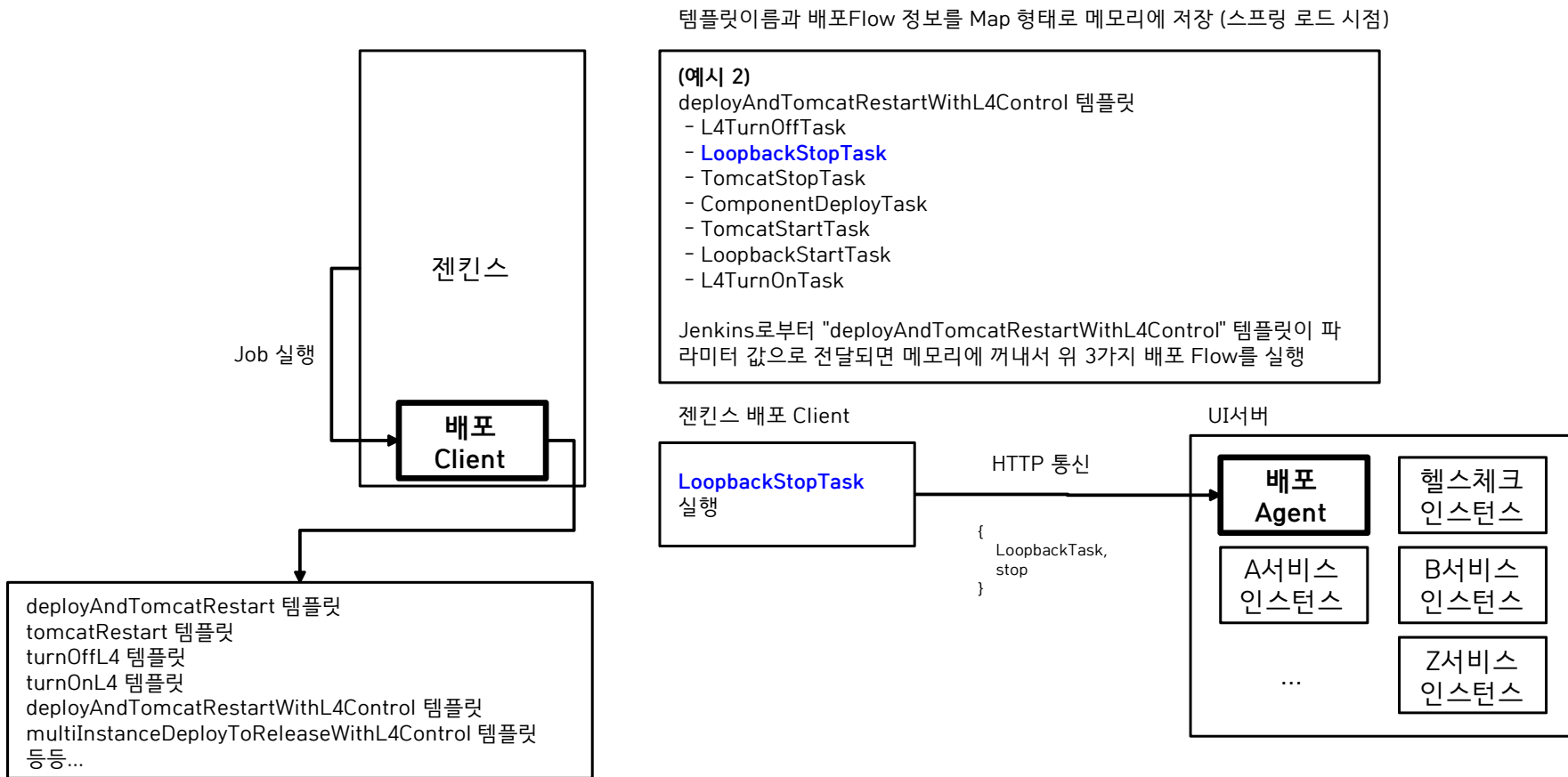
```
{
  L4Task,
  serviceName,
  turnOff,
  port
}
```

UI서버



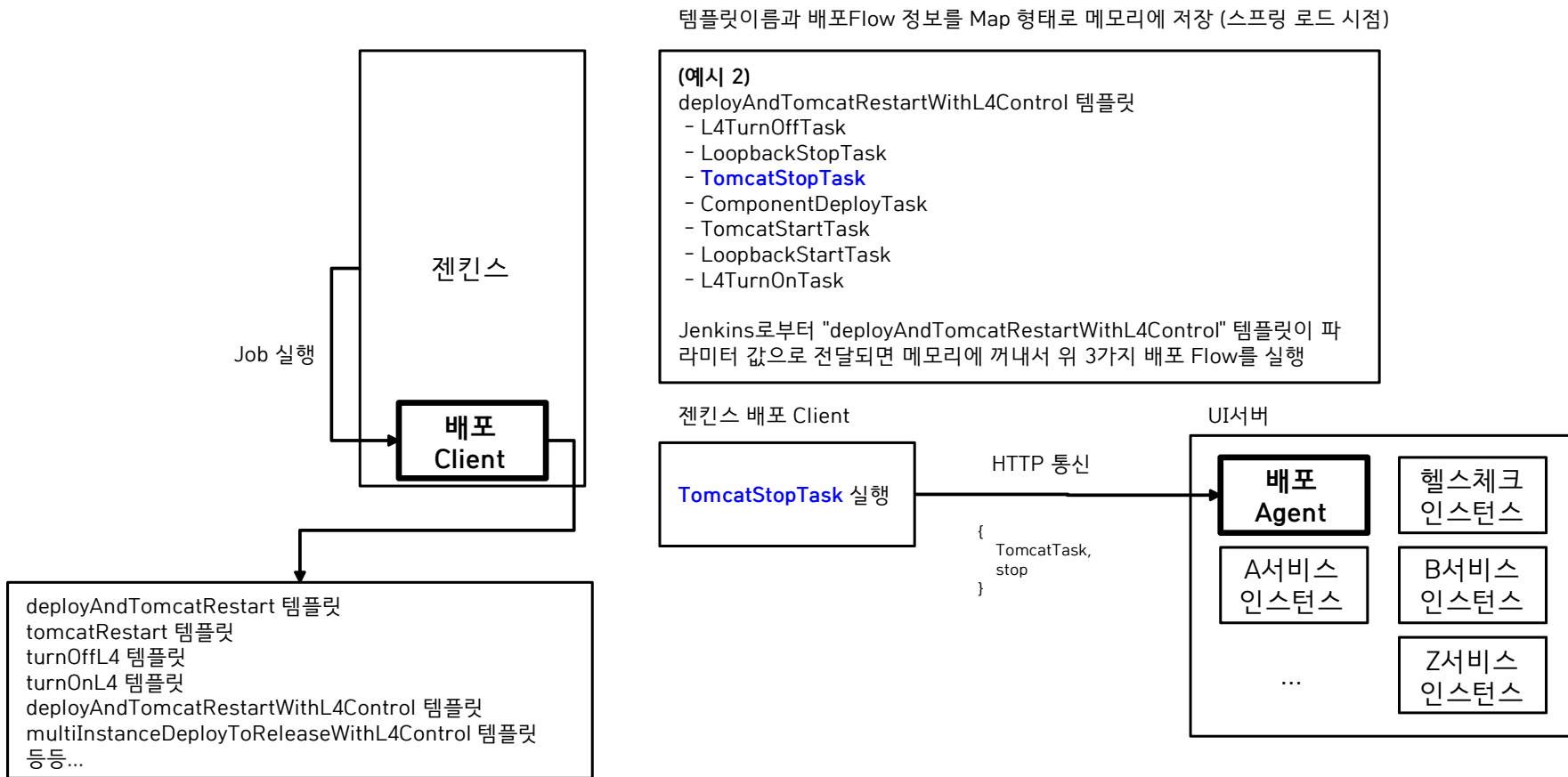
배포툴 관리

배포 Client 프로세스



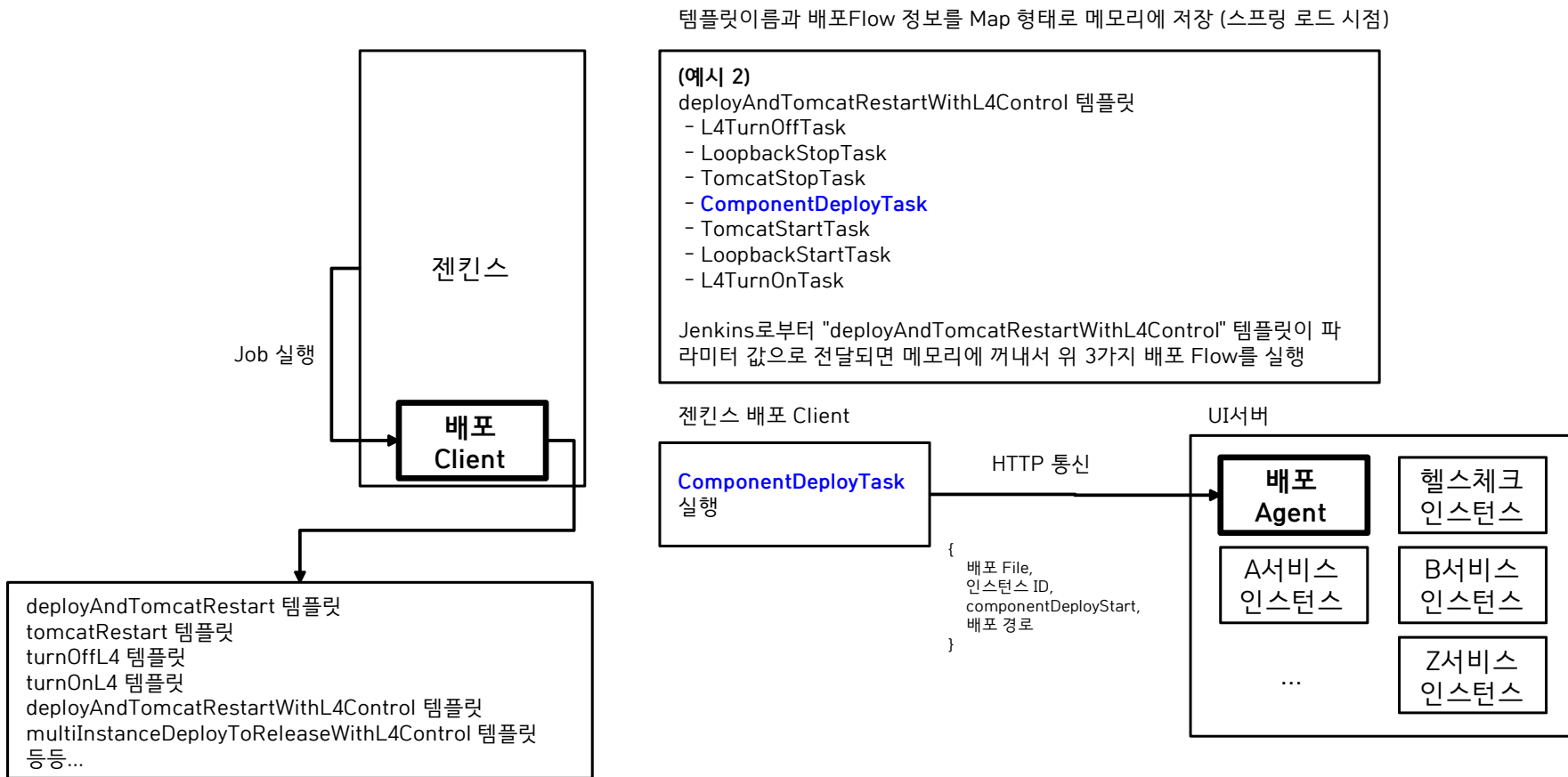
배포툴 관리

배포 Client 프로세스



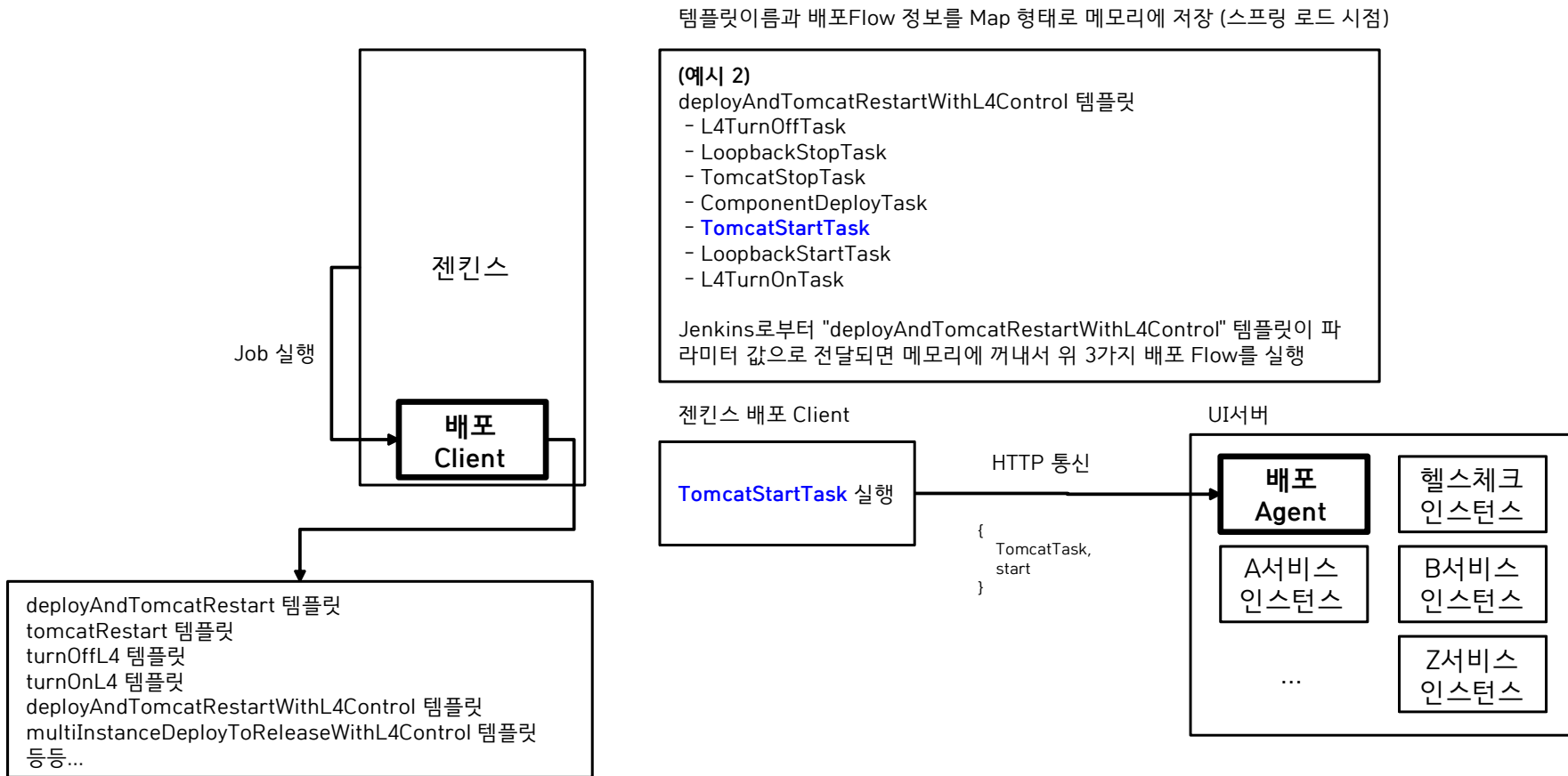
배포툴 관리

배포 Client 프로세스



배포툴 관리

배포 Client 프로세스



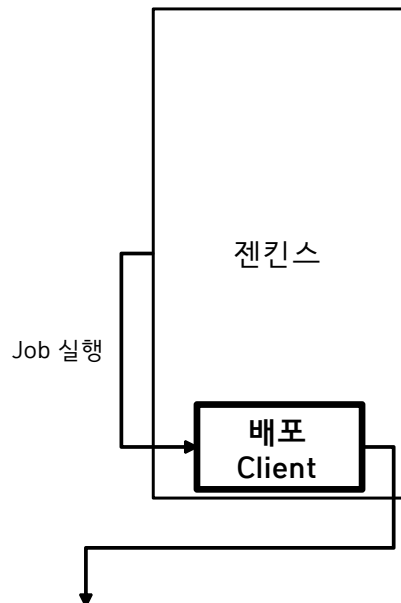
배포툴 관리

배포 Client 프로세스

템플릿이름과 배포Flow 정보를 Map 형태로 메모리에 저장 (스프링 로드 시점)

(예시 2)
deployAndTomcatRestartWithL4Control 템플릿
- L4TurnOffTask
- LoopbackStopTask
- TomcatStopTask
- ComponentDeployTask
- TomcatStartTask
- **LoopbackStartTask**
- L4TurnOnTask

Jenkins로부터 "deployAndTomcatRestartWithL4Control" 템플릿이 파라미터 값으로 전달되면 메모리에 꺼내서 위 3가지 배포 Flow를 실행



deployAndTomcatRestart 템플릿
tomcatRestart 템플릿
turnOffL4 템플릿
turnOnL4 템플릿
deployAndTomcatRestartWithL4Control 템플릿
multiInstanceDeployToReleaseWithL4Control 템플릿
등등...

jenkins 배포 Client

LoopbackStartTask
실행

HTTP 통신

```
{  
  LoopbackTask,  
  start  
}
```

UI서버

배포 Agent

A서비스
인스턴스

헬스체크
인스턴스

B서비스
인스턴스

Z서비스
인스턴스

...

배포툴 관리

배포 Client 프로세스

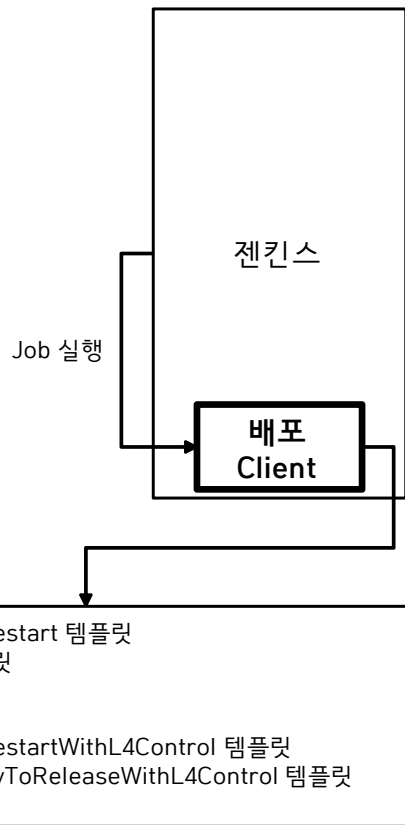
템플릿이름과 배포Flow 정보를 Map 형태로 메모리에 저장 (스프링 로드 시점)

(예시 2)

deployAndTomcatRestartWithL4Control 템플릿

- L4TurnOffTask
- LoopbackStopTask
- TomcatStopTask
- ComponentDeployTask
- TomcatStartTask
- LoopbackStartTask
- **L4TurnOnTask**

Jenkins로부터 "deployAndTomcatRestartWithL4Control" 템플릿이 파라미터 값으로 전달되면 메모리에 꺼내서 위 3가지 배포 Flow를 실행



jenkins 배포 Client

L4TurnOnTask 실행

HTTP 통신

```
{
  L4Task,
  serviceName,
  turnOn,
  port
}
```

"deployAndTomcatRestartWithL4Control"
템플릿 배포 종료

UI서버

배포
Agent

A서비스
인스턴스

헬스체크
인스턴스

B서비스
인스턴스

Z서비스
인스턴스

...

배포툴 관리

배포 Agent 프로세스

