

# LU09b - Deployment von Datenbanken & Backends

## AWS-Umgebung (Docker)

Der einfachste Weg, um Datenbank und Backend zu hosten, ist ein entsprechender Docker-Container zu erstellen. In der AWS-Buildumgebung ist der Zugriff auf die zugrundeliegende Docker-Instanz gewährt.

Damit man von ausserhalb der AWS-Umgebung auf diesen Backend-Container zugreifen kann wurde in der BZZ-AWS-Build-Umgebung ein Forwarding aktiviert, welches HTTP-Anfragen, die dem erwarteten Muster entsprechend an den Port 5000 des entsprechenden Containers weiterleitet.

Allgemein: `${BASE_URL}/api/${PROJECT_NAME}/${BRANCH_NAME}/${PATH} → ${PROJECT_NAME}_${BRANCH_NAME}_backend:5000/${PATH}`

Konkrete Beispiele:

- „<http://54.80.83.95/api/cicd/develop/polls>“ → „cicd\_develop\_backend:5000/polls“
- „<http://54.80.83.95/api/cicd/develop/votes>“ → „cicd\_develop\_backend:5000/votes“

```
location ~ ^/api/([^/]+)/([^/]+)/(.*)$ {
    proxy_pass http://$1_$2_backend:5000/$3;
}
```

Das bedeutet, dass im Frontend sichergestellt werden muss, dass Aufrufe des Backends die richtige URL (`/api/${PROJECT_NAME}/${BRANCH_NAME}/${PATH}`) verwenden.

Pipeline	Frontend-Code
<pre>npm install REACT_APP_API_BASE=/api/\${PROJECT_NAME}/\${BRANCH_NAME} \ npm run build</pre>	<pre>const API_BASE = process.env.REACT_APP_API_BASE; ... fetch(`\${API_BASE}/votes?poll_id=\${pollId}`) fetch(`\${API_BASE}/votes`, ... ) fetch(`\${API_BASE}/polls/\${code}`) fetch(`\${API_BASE}/polls`, ... )</pre>

TODO

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/cicd/learningunits/lu09/b?rev=1775485562>

Last update: **2026/04/06 16:26**

