

SPE-CLO5

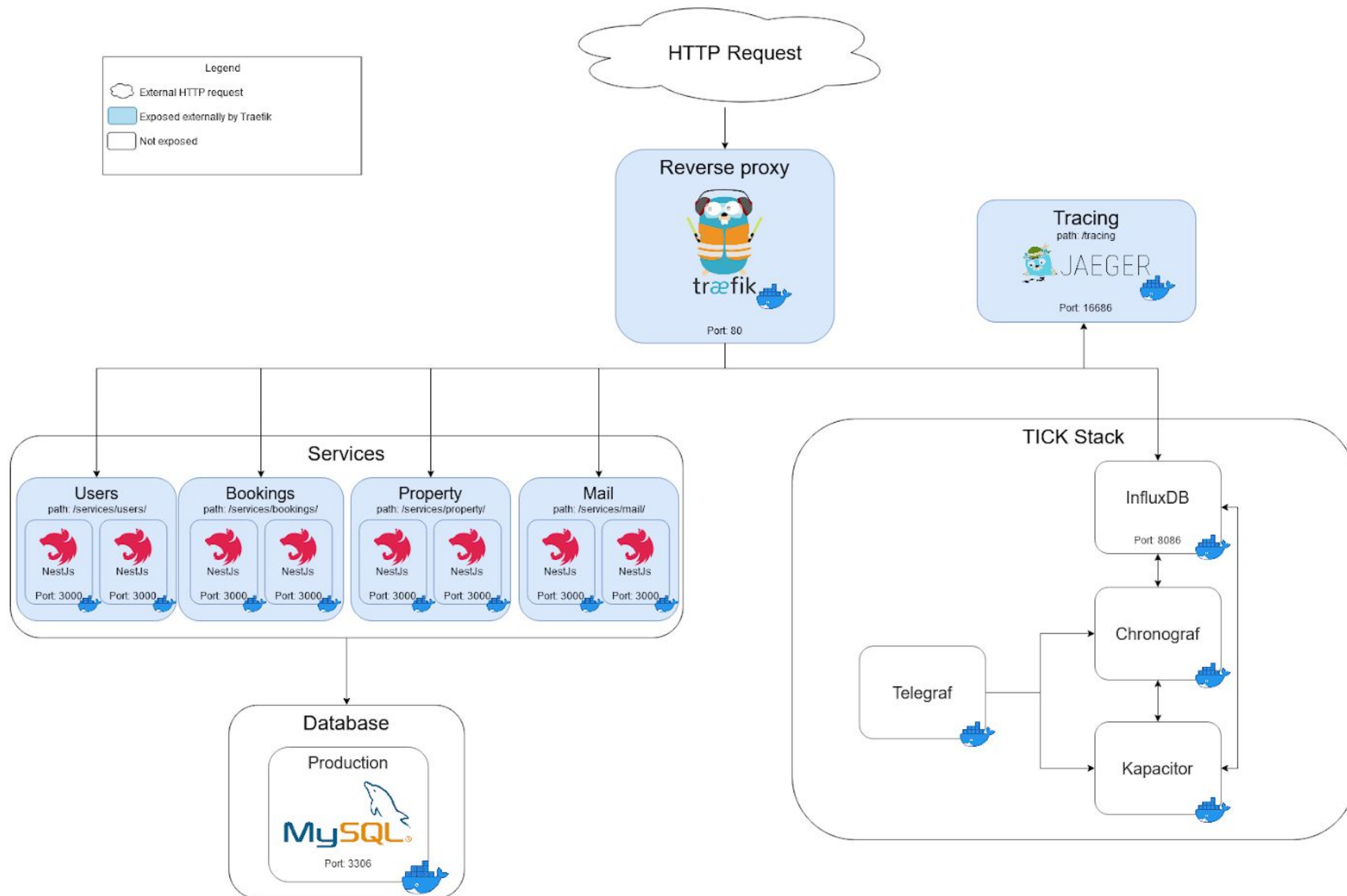
My Booking Services

-

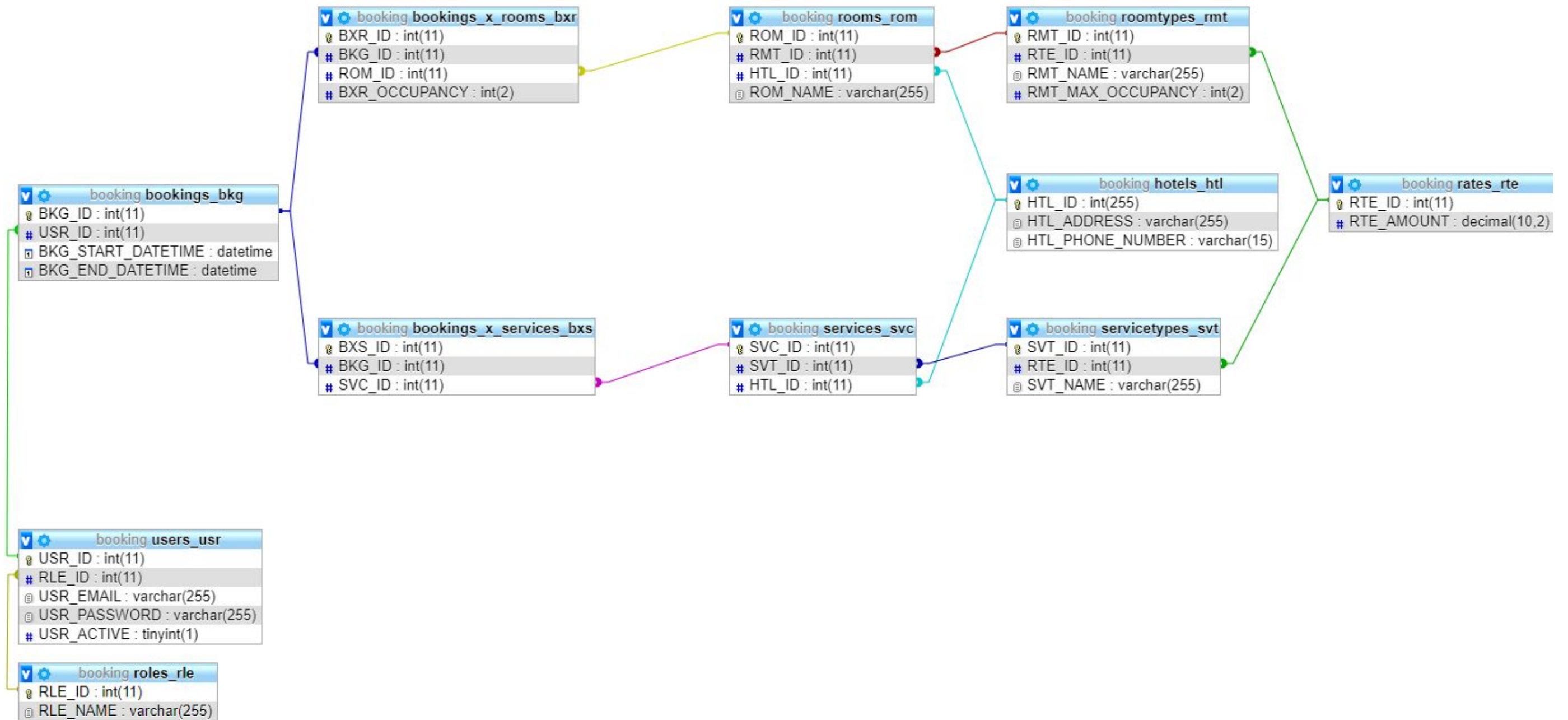
Technical Documentation

I.	Application Architecture	1
II.	Database Architecture	2
III.	System Architecture	3
IV.	Microservice Definition	4
	A. User Management	4
	B. Property Management	5
	C. Booking Management	7
	D. Mailing	8
V.	Project Management	9
	A. Milestones	9
	B. Issues	10

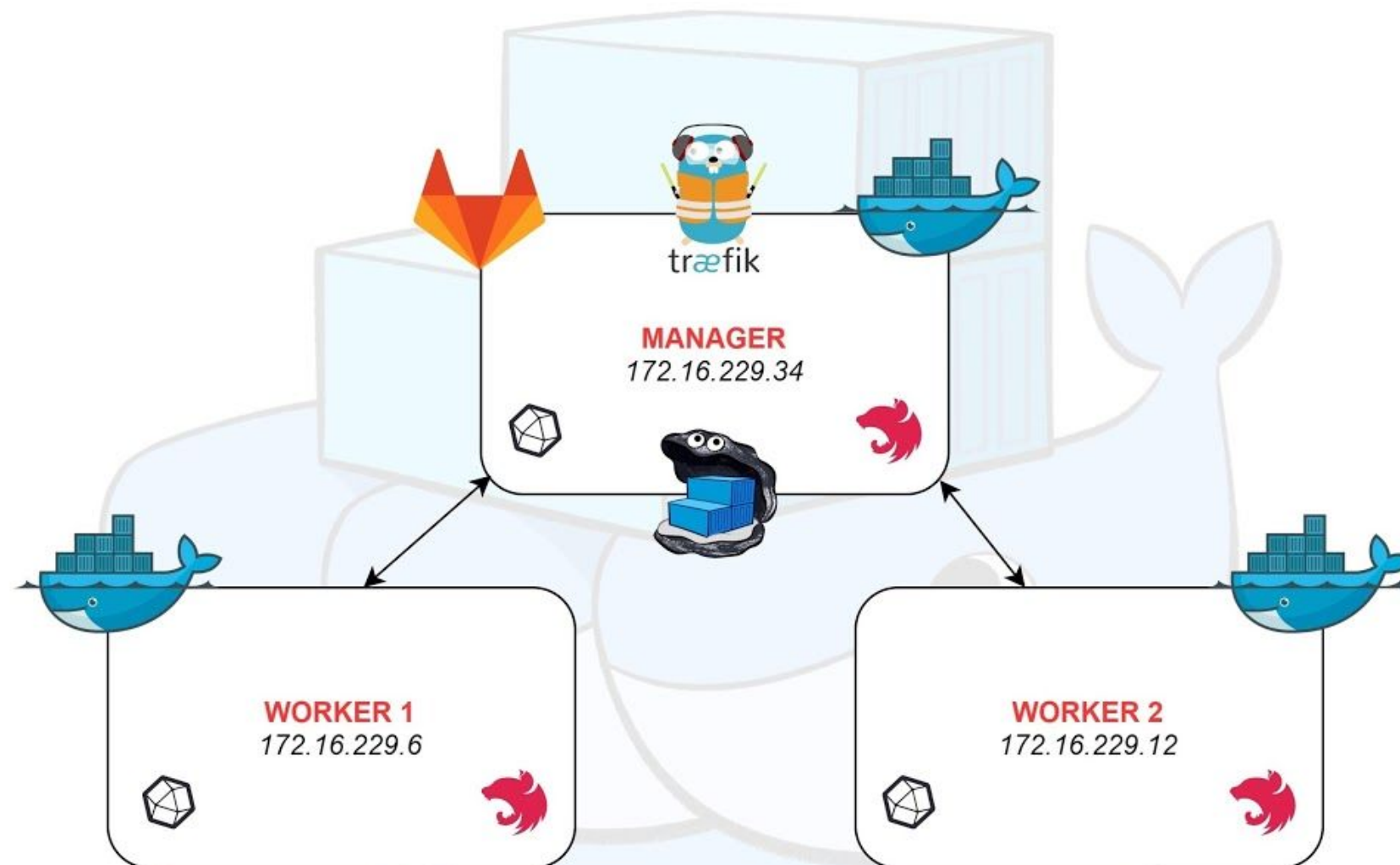
I. Application Architecture



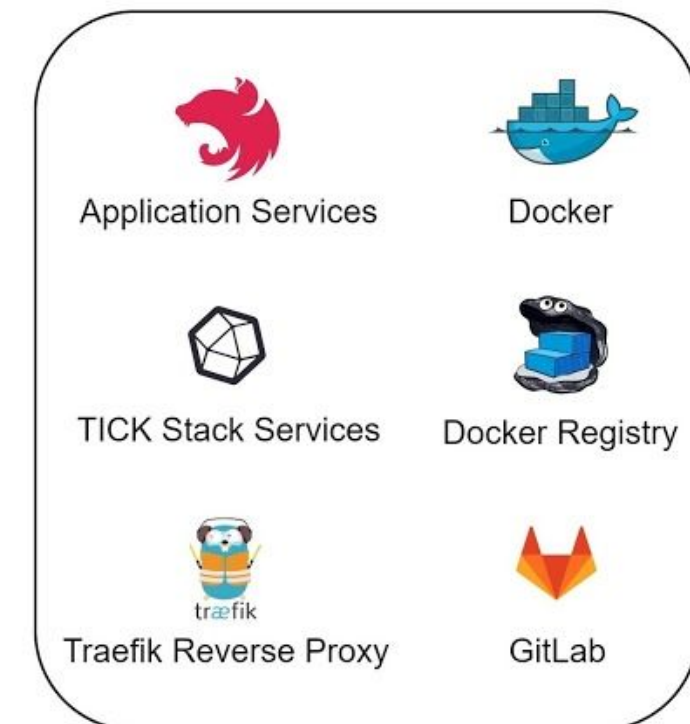
II. Database Architecture



III. System Architecture

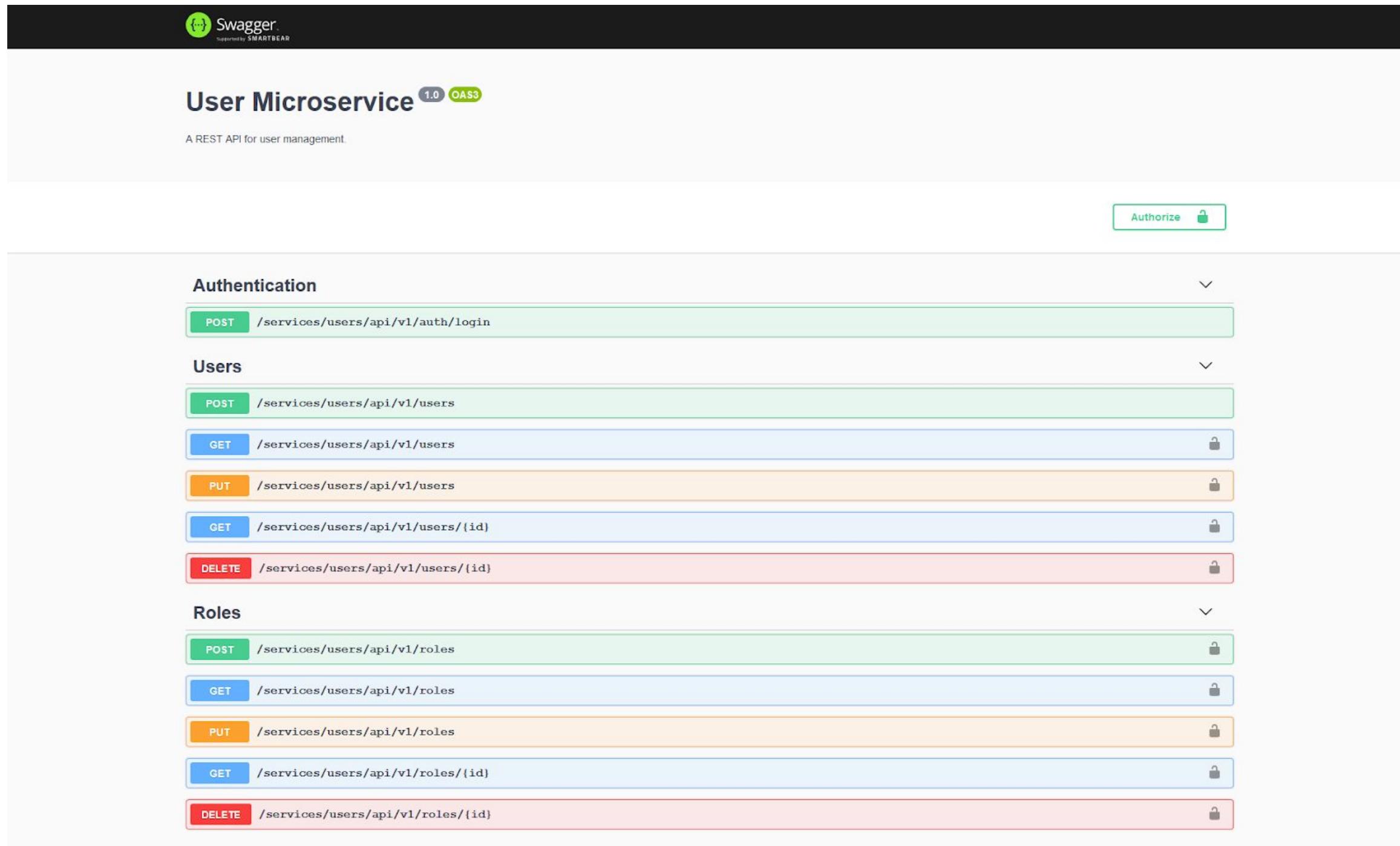


System Architecture - Docker Swarm



IV. Microservice Definition

A. User Management - Live version available at <http://172.16.229.6/services/users>



The image shows the Swagger UI for the 'User Microservice'. The header includes the Swagger logo and the text 'User Microservice 1.0 OAS3'. Below the header, there is a description: 'A REST API for user management.' and an 'Authorize' button. The main content is organized into three sections: Authentication, Users, and Roles. Each section contains a list of API endpoints with their respective HTTP methods and status indicators.

User Microservice 1.0 OAS3
A REST API for user management.

Authorize

Authentication

- POST /services/users/api/v1/auth/login


Users

- POST /services/users/api/v1/users
- GET /services/users/api/v1/users
- PUT /services/users/api/v1/users
- GET /services/users/api/v1/users/{id}
- DELETE /services/users/api/v1/users/{id}

Roles


- POST /services/users/api/v1/roles
- GET /services/users/api/v1/roles
- PUT /services/users/api/v1/roles
- GET /services/users/api/v1/roles/{id}
- DELETE /services/users/api/v1/roles/{id}

B. **Property Management** - Live version available at <http://172.16.229.6/services/property>






 **Swagger**
Supported by SMARTBEAR

Property Microservice 1.0 OAS3






A REST API for property management.

Authorize 

Hotels

POST	/services/users/api/v1/hotels	
GET	/services/users/api/v1/hotels	
PUT	/services/users/api/v1/hotels	
GET	/services/users/api/v1/hotels/{id}	
DELETE	/services/users/api/v1/hotels/{id}	

Rooms

POST	/services/users/api/v1/rooms	
GET	/services/users/api/v1/rooms	
PUT	/services/users/api/v1/rooms	
GET	/services/users/api/v1/rooms/{id}	
DELETE	/services/users/api/v1/rooms/{id}	

Services



POST	/services/users/api/v1/services	
GET	/services/users/api/v1/services	
PUT	/services/users/api/v1/services	
GET	/services/users/api/v1/services/{id}	
DELETE	/services/users/api/v1/services/{id}	

Room Types



POST	/services/users/api/v1/roomTypes	
GET	/services/users/api/v1/roomTypes	
PUT	/services/users/api/v1/roomTypes	
GET	/services/users/api/v1/roomTypes/{id}	
DELETE	/services/users/api/v1/roomTypes/{id}	

Service Types




POST	/services/users/api/v1/serviceTypes	
GET	/services/users/api/v1/serviceTypes	
PUT	/services/users/api/v1/serviceTypes	
GET	/services/users/api/v1/serviceTypes/{id}	
DELETE	/services/users/api/v1/serviceTypes/{id}	

Rates




POST	/services/users/api/v1/rates	
GET	/services/users/api/v1/rates	
PUT	/services/users/api/v1/rates	
GET	/services/users/api/v1/rates/{id}	
DELETE	/services/users/api/v1/rates/{id}	

C. **Booking Management** - Live version available at <http://172.16.229.6/services/bookings>







 Swagger
Supported by SMARTBEAR

Booking Microservice 1.0 OAS3





A REST API for booking management.

Authorize 


Bookings

POST	/services/users/api/v1/bookings	
GET	/services/users/api/v1/bookings	
GET	/services/users/api/v1/bookings/{startDate}/{endDate}	
GET	/services/users/api/v1/bookings/{id}	
PUT	/services/users/api/v1/bookings/{id}	
DELETE	/services/users/api/v1/bookings/{id}	

Rooms

POST	/services/users/api/v1/bookings-rooms	
GET	/services/users/api/v1/bookings-rooms	
GET	/services/users/api/v1/bookings-rooms/{BXR_ID}	
PUT	/services/users/api/v1/bookings-rooms/{BXR_ID}	


D. **Mailing** - Live version available at <http://172.16.229.6/services/mail>

 **Swagger**
Supported by SMARTBEAR


Mailing Microservice


1.0 OAS3

A REST API for mailing.

Authorize 

Mailing

POST /services/users/api/v1/mail 

POST /services/users/api/v1/mail/refresh 

V. Project Management

A. **Milestones** - Live version available at <http://172.16.229.34/root/my-booking-services/-/milestones>

The screenshot shows the GitLab interface for the 'My Booking Services' project. The left sidebar contains navigation links: Project overview, Repository, Issues (31), Merge Requests (0), CI / CD, Operations, Analytics, Wiki, Snippets, and Settings. The 'Issues' link is highlighted. The main content area is titled 'Milestones' and shows a list of five milestones. Each milestone includes a title, a date range, a progress bar, and a 'Close Milestone' button. The milestones are: 1. Define Project Architecture (Mar 16, 2020–Apr 24, 2020, 4 Issues, 0 Merge Requests, 0% complete), 2. Build Environment (Mar 16, 2020–Apr 24, 2020, 10 Issues, 0 Merge Requests, 0% complete), 5. Achieve Continuous Deployment (Apr 24, 2020–May 14, 2020, 1 Issue, 0 Merge Requests, 0% complete, Upcoming), 4. Achieve Continuous Integration (Apr 24, 2020–May 14, 2020, 4 Issues, 0 Merge Requests, 0% complete, Upcoming), and 3. Build Application & Tests (Mar 16, 2020–May 14, 2020, 12 Issues, 0 Merge Requests, 0% complete).

Milestone	Issues	Merge Requests	Completion	Status
1. Define Project Architecture	4	0	0% complete	
2. Build Environment	10	0	0% complete	
5. Achieve Continuous Deployment	1	0	0% complete	Upcoming
4. Achieve Continuous Integration	4	0	0% complete	Upcoming
3. Build Application & Tests	12	0	0% complete	

B. Issues (all but 'open' are assigned) - Live version available at <http://172.16.229.34/root/my-booking-services/-/boards>

The screenshot displays the GitLab Issue Boards interface for the 'My Booking Services' project. The top navigation bar includes the GitLab logo, project navigation links (Projects, Groups, More), a search bar, and user profile information. The left sidebar shows the project overview, repository, and a list of issues (31). The main area shows the 'MBS Kanban' board with four columns: Open, In Progress, Done, and Done & Deployed. Each column contains several issue cards with titles, labels, and status indicators.

Column	Issue Title	Labels	Status
Open	Deploy Services within Gitlab CD Pipeline	CD, Docker, Reverse Proxy, Services	Open
In Progress	Build User Service	Services	In Progress
In Progress	Build Booking Service	Services	In Progress
In Progress	Build User Service Tests	Services, Testing	In Progress
In Progress	Build Booking Service Tests	Services, Testing	In Progress
In Progress	Build Mailing Service Tests	Services, Testing	In Progress
In Progress	Secure all Services (JWT + Bcrypt + Roles)	Security, Services	In Progress
In Progress	Create Architecture Diagram	Architecture, Documentation, Project Management	In Progress
Done	Design Database	Architecture, Services	Done
Done	Define Micro-Service bounds	Architecture, Services	Done
Done	Build Property Service	Services	Done
Done	Build Property Service Tests	Services, Testing	Done
Done	Build Mailing Service	Services	Done
Done	Orchestrate Service Image Building	Docker, Services	Done
Done	Setup Swagger on all Services	Documentation, Services	Done
Done & Deployed	[ALL] Add ssh keys to Nodes	Ansible, Environment	Done & Deployed
Done & Deployed	[MANAGER] Install & configure Docker	Ansible, Docker, Environment	Done & Deployed
Done & Deployed	[MANAGER] Create Swarm	Ansible, Docker, Environment	Done & Deployed
Done & Deployed	[WORKERS] Make Workers join Swarm	Ansible, Docker, Environment	Done & Deployed
Done & Deployed	[MANAGER] Create and run Registry	Ansible, Docker, Environment	Done & Deployed
Done & Deployed	[MANAGER] Install & configure Gitlab	Ansible, Environment	Done & Deployed
Done & Deployed	[MANAGER] Create Gitlab users	Ansible, Environment	Done & Deployed