PetStore API Performance Testing Report

1. **Test objective.**

These performance tests are performed with the objective of analyzing the behavior of the API with different conditions and resources.

2. **Test cases proposed.**

Because this API has 3 transactional sections, smoke, load and stress tests are created for these sections where the flows will be the ones that the users would normally do.

Pet section = Create pet, query it by ID, update pet, upload image to pet, delete pet.
Store section = Consult inventory, create order, query order, delete order
User section = Login and logout of users

3. **Environment**

These tests were created with K6 and run in a Docker container with the specifications of 2cpu and 2gb of memory.

4. **Parameterization of the execution**

Section	Smoke test	Load Test	Stress Test

Pet	vus= 1
	duration = 50s
		Stages: 3
		{ duration: "30s",
		  target: 3,
		},
		{ duration: "2m",
		  target: 7,
		},
		{ duration: "20s",
		  target: 5,
		}
			Stages: 4
			{ duration: "45s",
			  target: 6,
			},
			{ duration: "1m",
			  target: 7,
			},
			{ duration: "2m",
			  target: 20,
			}
			{ duration: "15s",
			  target: 10,
			}

Store	vus= 50
	duration = 10s
		Stages: 3
		{ duration: "50s",
		  target: 3,
		},
		{ duration: "90s",
		  target: 5,
		},
		{ duration: "30s",
		  target: 3,
		}
			Stages: 3
			{ duration: "30s",
			  target: 30 ,
			},
			{ duration: "45s",
			  target: 65,
			},
			{ duration: "15s",
			  target: 20,
			}

User	vus= 10
	duration = 10s
		Stages: 3
		{ duration: "10s",
		  target: 300,
		},
			Stages: 4
			{ duration: "45s",
			  target: 6,
			},

<table>
<tr><td></td><td></td><td>

```
{ duration: "20s",
   target: 5,
},
{ duration: "30s",
   target: 1000,
}
```

</td><td>

```
{ duration: "1m",
   target: 7,
},
{ duration: "2m",
   target: 20,
}
{ duration: "15s",
   target: 10,
}
```

</td></tr>
</table>

5. Results of the execution

Section	Smoke test	Load Test	Stress Test
Pet	Iterations= 9 Iterations duration Min = 6.04s Max = 6.42s Avg = 6.12s Request =36 Request failed= 0%	Iterations= 123 Iterations duration Min = 6.02s Max = 6.45s Avg = 6.07s Request = 488 Request failed= 0.81%	Iterations= 578 Iterations duration Min = 3s Max = 6.53s Avg = 4.02 Request =1127 Request failed= 36.29%
Store	Iterations= 654 Iterations duration Min = 38.45ms Max = 2.45s Avg = 782.88ms Request =654 Request failed= 0%	Iterations= 105 Iterations duration Min = 5.02s Max = 5.59s Avg = 5.06s Request = 420 Request failed= 0.81%	Iterations= 566 Iterations duration Min = 5s Max = 7.58 Avg = 5.97s Request = 1591 Request failed= 42.74%
User	Iterations= 90 Iterations duration Min = 1.04s Max = 1.79s Avg = 1.22s Request =90 Request failed= 0%	Iterations= 1150 Iterations duration Min = 1s Max = 59.46s Avg = 17.46s Request =2984 Request failed= 0%	Iterations= 13664 Iterations duration Min = 1s Max = 1m1s Avg = 3.2s Request =14249 Request failed= 93.15%

6. Conclusions

The endpoints by not having many validations such as correctly performing the login/logout, the creation with the same ID can generate false expectations of the tests because we cannot see how they respond to these events, it is also important to note that the creation/edition are the iterations that are most delayed in orders and pets.

7. Recommendations

Tests should be performed in environments with other conditions to know how much can be improved with the current results, as well as considering the login and logout requests because it can be something usual that can generate service crashes.