A Case-study On The value of Operators
What are Operators?

- Operators provide a way of packaging, deploying and managing Kubernetes applications using software.
- Comprises of CRD, a custom controller and operational business logic.
Evolution of Application Architectures
The Monolith
Distributed Session Management

Microservice

Microservice

Microservice

Load Balancer

Users
Shared Session Store
Stateless Tokens
A Peek into Doorman (An Auth Microservice)
Scenario

- Picture a company established in the early 2000s is attempting to transform their monolithic application into a set of microservices
- The first phase will involve creating an authentication microservice named, doorman
- The application will be hosted in a Container platform such as Openshift or Kubernetes
Doorman Feature List

- Manage users
- Creating and signing tokens
- Validating tokens
JSON Web Tokens (JWT)

- Pronounced as the English word “jot”
- Is compact and URL safe way to represent claims
- Have three sections - header, claims and signature verification
- Some supported algorithms are:
  - None
  - HMAC (HS256)
  - RSA (RS256)
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
JWT Header

{
    "alg": "RS256",
    "typ": "JWT"
}
JWT Claims

{
    "exp": 1571715259,
    "iat": 1571693659,
    "id": "cb13924f-e5da-4e5c-b5d3-566ed32faad7",
    "sub": "Edmund"
}

JWT Signature

● This portion of the JWT is used to determine if the contents of the JWT has been tampered with
HMAC

- Uses a symmetrical algorithm
- Designed to be fast
- Should only be used within the same microservice
- Also susceptible to brute force attacks
RS256 (RSA Key pair)

- Uses an asymmetrical algorithm
- Slower than HMAC
- Secure and can be used across microservices
- Disable HMAC if using RS256 to sign JWT tokens
verifyKey, err := jwt.ParseWithClaims(jwtString, &MyClaims{},
    func(token *jwt.Token) (interface{}, error) {
        if token.Method.Alg() != "RS256" {
            return nil, errors.New("Unsupported signing key")
        }
    }
)

...
POST  http://localhost:5000/user/login

Response headers 3  Request headers 1  Redirects 0  Timings

content-type: application/json
date: Fri, 25 Oct 2019 14:52:57 GMT
content-length: 860

```json
{
"token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiJEmzIzNzcsImd1dCI6MTU3MjAxNTE3
NywwaWQiOiJjYyZvMy01NjZlZDMyZmFhZDEiLCJzdWIiOiJFZG11bmQiLCJF

U0JxJ-jdKlaEXVqUk0500f--steTeNTVr0QKH2fEgtyr60kOeZGd5ok-UUr

Py9AnejF8prfegITxpiGdwfynlaM1cvMUEJjDii1SZN6t7M97q05diP8t0uleZd

DzJGkaR8R4nFH1Dkgse-Rdb782WpMY4t6-siJ08Z9FeVm5REAiWmor0hF0kuP3ExokS8kYTDS9qYSr20Q-

Ukbf0dRhmSmSR78WjiPRFwaKvR5rj-XMXj0cph856-

MF0z0tkmK_m86rL0kruoWJyBFshMLTbppszBDfITT7FssZeikLRYfkkvXy6FBed-KnYw3bM3jJVNZj7lgXwv6Wta

dFvDC35gavMi7isi9qPPiDWMh0egbubKFbq-fLpiL_pnc93G0JcKvMUPsrRKnBhxl_m6t-T8s4txoa8VfzG6AUxjv-
a0xpmLqpKMq29L984991v0TvhgVf5Cfd5uRh0AvVou-UjYBoK977cRF0cZ0P5b0qqIE20119rU-

klyF0x933gS5hyKNVbfUP9amRk3U4pYvB--

fwX4CYqF673L2z4x2rNrd3BTIT757NE7bJvPP7oMYXPawd0wjHl1pBsz8IKrbi5Fv0w552PgHydkBe70GGhb4HKT

Tw0gA7ar2gd0eC4xMkkbR491JSaN12PM"
}```
content-type: application/json

date: Fri, 25 Oct 2019 14:58:17 GMT
content-length: 287

[Array[2]]
  - 0: {
      "id": "2019-09-14 21:09:54",
      "created_at": "cb13924f-e5da-4e5c-b5d3-566ed32faad7",
      "firstname": "Edmund",
      "lastname": "0chieng",
      "username": "eochieng"
    },
  - 1: {
      "id": "2019-10-24T04:29:42Z",
      "created_at": "64d0ac74c-4aa6-4b59-a16b-4e128bff0fa9",
      "firstname": "Jane",
      "lastname": "Doe",
      "username": "jane.doe"
    }
Database Design

- Data security
- Scalability
- Client library support
- Performance
- Data replication
- High-availability
- Resilience
Persistent Volumes

- This volume would be required to store persistent data from the micro-service.
- Data that should be retained long-term should not be installed within a container.
Technical resources involved

- Database Administrators to review database design and tuning
- Network engineers/Security engineers to ensure application is secure
- Cloud Operations/Engineering team
- Backup team
Comparison of Install Processes
Install via Kubernetes Manifests

- Create Database credential secret
- Create tls secret to store RSA SSL certificate and private key
- Create Persistent volume used to hold the microservice data
- Install the Database statefulset
- Create database credentials
- Define the application database and tables
- Deploy the application
Installation using Operators

- Deploy the operator
- Deploy the application via a custom resource
Doorman Operator Design
Operator Maturity

**Phase I**
- **Basic Install**
  - Automated application provisioning and configuration management

**Phase II**
- **Seamless Upgrades**
  - Patch and minor version upgrades supported

**Phase III**
- **Full Lifecycle**
  - App lifecycle, storage lifecycle (backup, failure recovery)

**Phase IV**
- **Deep Insights**
  - Metrics, alerts, log processing and workload analysis

**Phase V**
- **Auto Pilot**
  - Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

---

**HELM**

---

**Ansible**

---

**GO**
Level 1 - Basic Install

- Generate/Acquire RSA SSL certificate for signing JWT tokens
- Create Database for storing user identities
- Create credentials for connecting to the database
- Deploy the authentication microservice
Level 2 - Seamless Upgrade

- Upgrade to the latest version of the application
- Rollback to previous version when deemed necessary
Level 3 - Full Lifecycle

● Create backups of the database
● Check integrity of backups
● Test functionality of the microservice
Level 4 - Deep Insights

- Expose custom metrics for the microservice
- Show graphs based on the metrics
- Setup alerting for only scenarios that need human intervention
curl -s http://localhost:5000/metrics | egrep '^doorman\n\ndoorman_login_rate_seconds_bucket{total_logins="200",le="0.1"} 0
\ndoorman_login_rate_seconds_bucket{total_logins="200",le="0.5"} 1
\ndoorman_login_rate_seconds_bucket{total_logins="200",le="1"} 1
\ndoorman_login_rate_seconds_bucket{total_logins="200",le="5"} 1
\ndoorman_login_rate_seconds_bucket{total_logins="200",le="+Inf"} 1
\ndoorman_login_rate_seconds_sum{total_logins="200"} 0.134216975
\ndoorman_login_rate_seconds_count{total_logins="200"} 1
\ndoorman_login_rate_seconds_bucket{total_logins="401",le="0.1"} 7
\ndoorman_login_rate_seconds_bucket{total_logins="401",le="0.5"} 7
\ndoorman_login_rate_seconds_bucket{total_logins="401",le="1"} 7
\ndoorman_login_rate_seconds_bucket{total_logins="401",le="5"} 7
\ndoorman_login_rate_seconds_bucket{total_logins="401",le="+Inf"} 7
\ndoorman_login_rate_seconds_sum{total_logins="401"} 0.642076606
\ndoorman_login_rate_seconds_count{total_logins="401"} 7
\ndoorman_total_failed_logins 7

shell$ 1
Level 5 - Auto Pilot

- Scale the auth service up/down based on traffic
- Ensure that the SSL certificates in use are valid
Conclusion
Conclusion

- Operators can save time for your skilled human assets and allow them to focus on enhancing the agenda of your organizations.
- You also get consistency within each operator version.
Thank you

Red Hat is the world’s leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.