

# Kubernetes 101

Pranshu Saxena  
Ruchi Singh

# Agenda

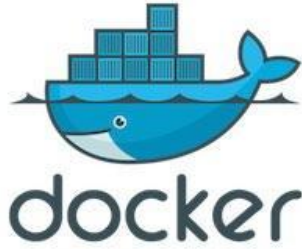
- Where Paytm is heading
- Docker
- Kubernetes
  - Architecture
  - Concepts
  - Security
  - Monitoring
  - Logging

# Aim

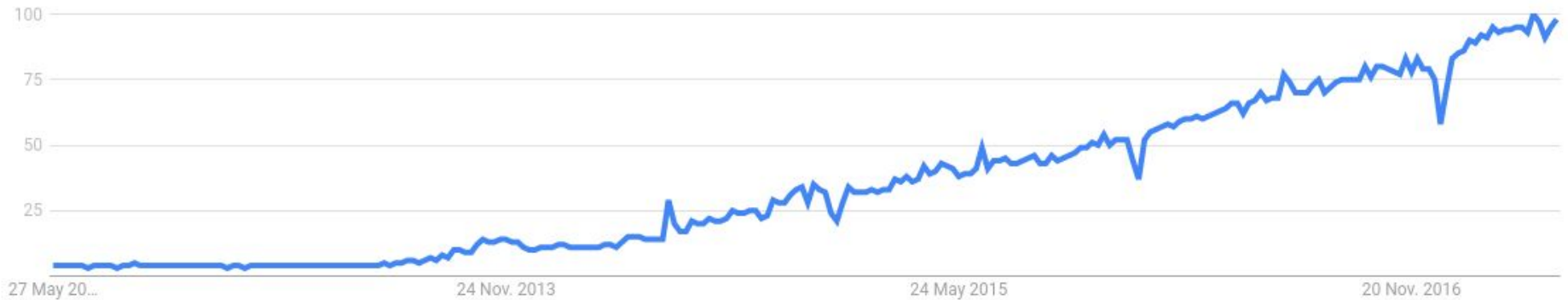
Everything at Paytm runs in containers.



# Docker



Interest over time ?

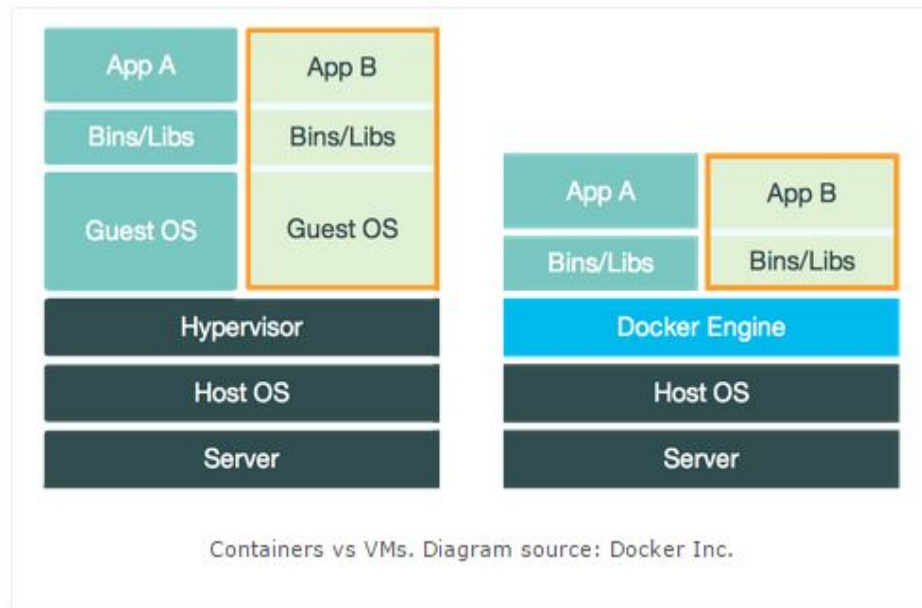


Source: Google Trends

# What is Docker?

- An implementation of the container idea.
- A package format.
- Resource isolation.
- An Ecosystem.

“build once, run anywhere”



# We need more

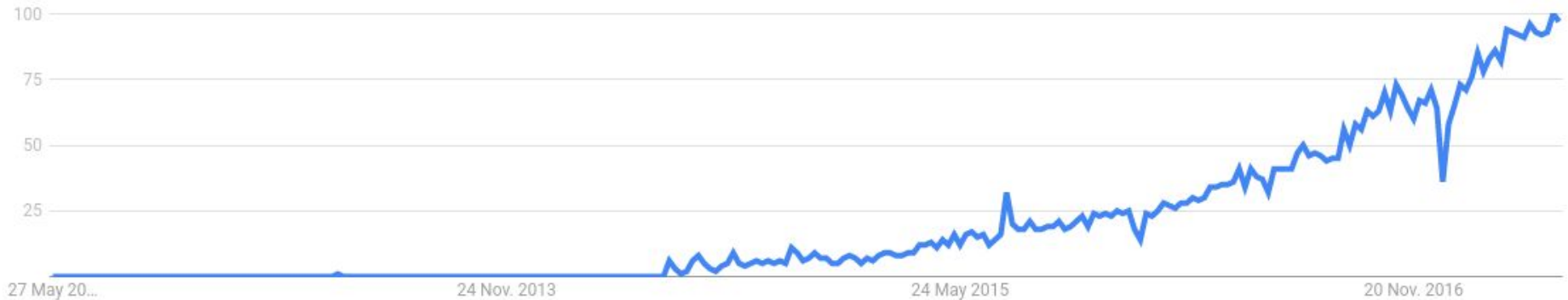
- **Scheduling:** Where should my containers run?
- **Monitoring:** What's happening with my containers?
- **Authorization:** Control who can do things to my containers.
- **Aggregates:** Compose sets of containers into jobs.
- **Scaling:** Making jobs bigger or smaller.
- **Seamless rolling update:** Handle multiple versions of containers without affecting applications.

# Kubernetes



kubernetes  
by Google™

Interest over time ?



Source: Google Trends

# Kubernetes

- Container orchestrator.
- Builds on Docker containers.
  - Also supporting other container technologies.
- Multiple cloud and bare-metal environments.
- 100% Open source, written in Go.

Let users manage applications, not machines.



# Kubernetes Architecture



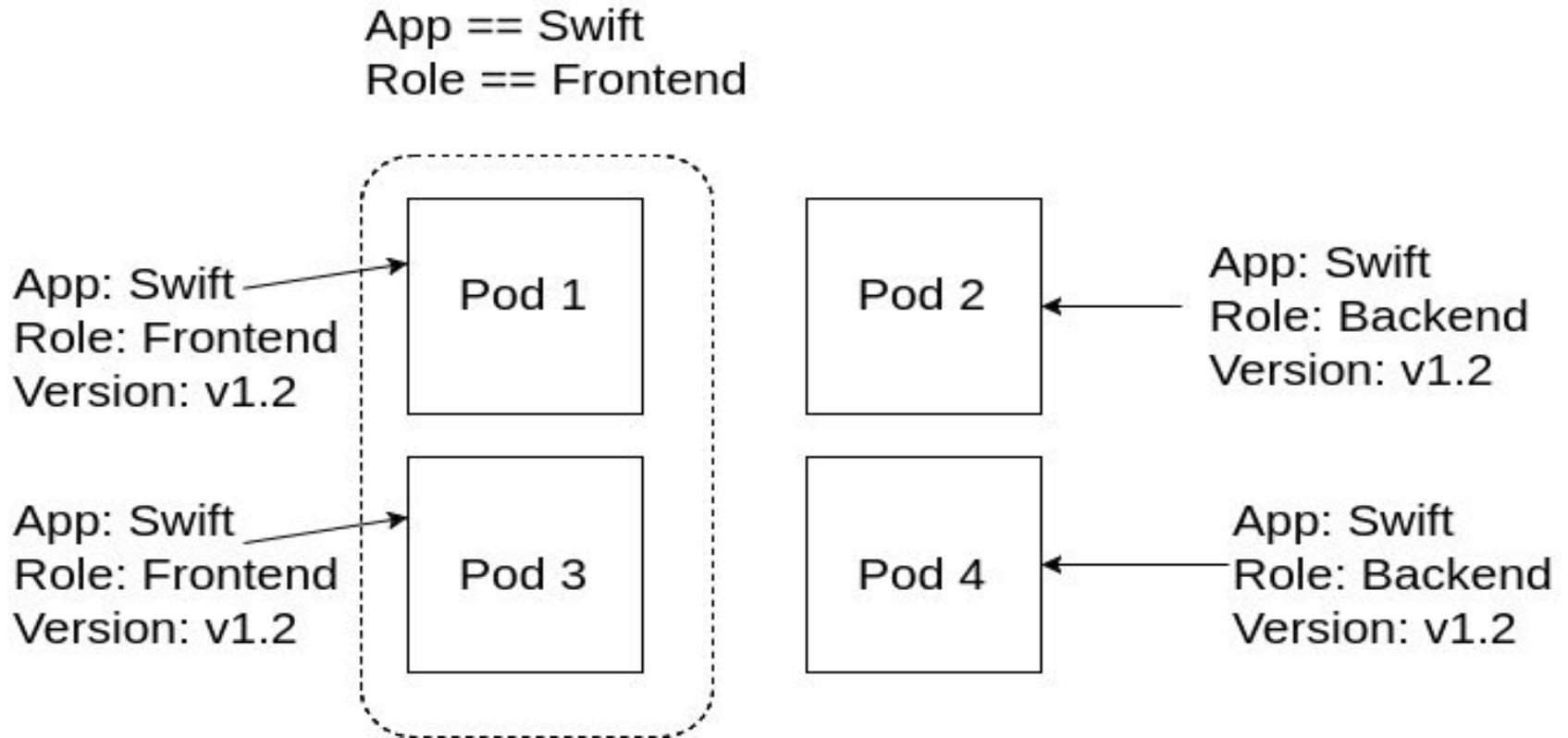
# Primary Concepts

- Container: A sealed application package (Docker).
- Pod: A small group of tightly coupled Containers.
- Labels: Identifying metadata attached to objects.
- Selector: A query against labels, producing a set result.
- Controller: A reconciliation loop that drives current state towards desired state.
- Service: A set of pods that work together.

# Labels

- Arbitrary metadata, represent Identity of Pods.
- Attached to any API object.
- Query-able by Selectors:
  - Think SQL 'select... where...'
- Few usages:
  - Select pods under a ReplicationController.
  - Select pods in a Service.

# Selectors



# Replication Controller

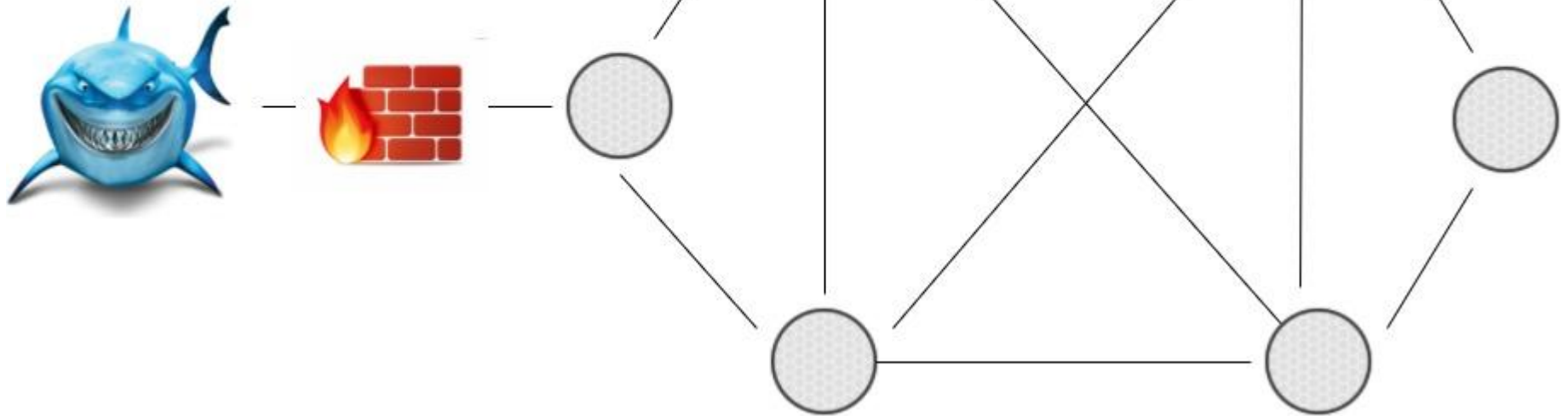
- A Replication Controller ensures that a specified number of pod 'replicas' are running at any one time.
- Replication Controller uses Pod Templates to create Pods.
- Replication controller uses Pod Labels to monitor and maintain the number of Pods to the desired level.

# Container Networking with Policy

- Weave Net: Creates a virtual network that connects containers/Pods across multiple hosts and enables their automatic discovery.
- No external cluster store required.
- Automatically chooses the fastest path between two hosts.
- Supports Kubernetes NetworkPolicy API.

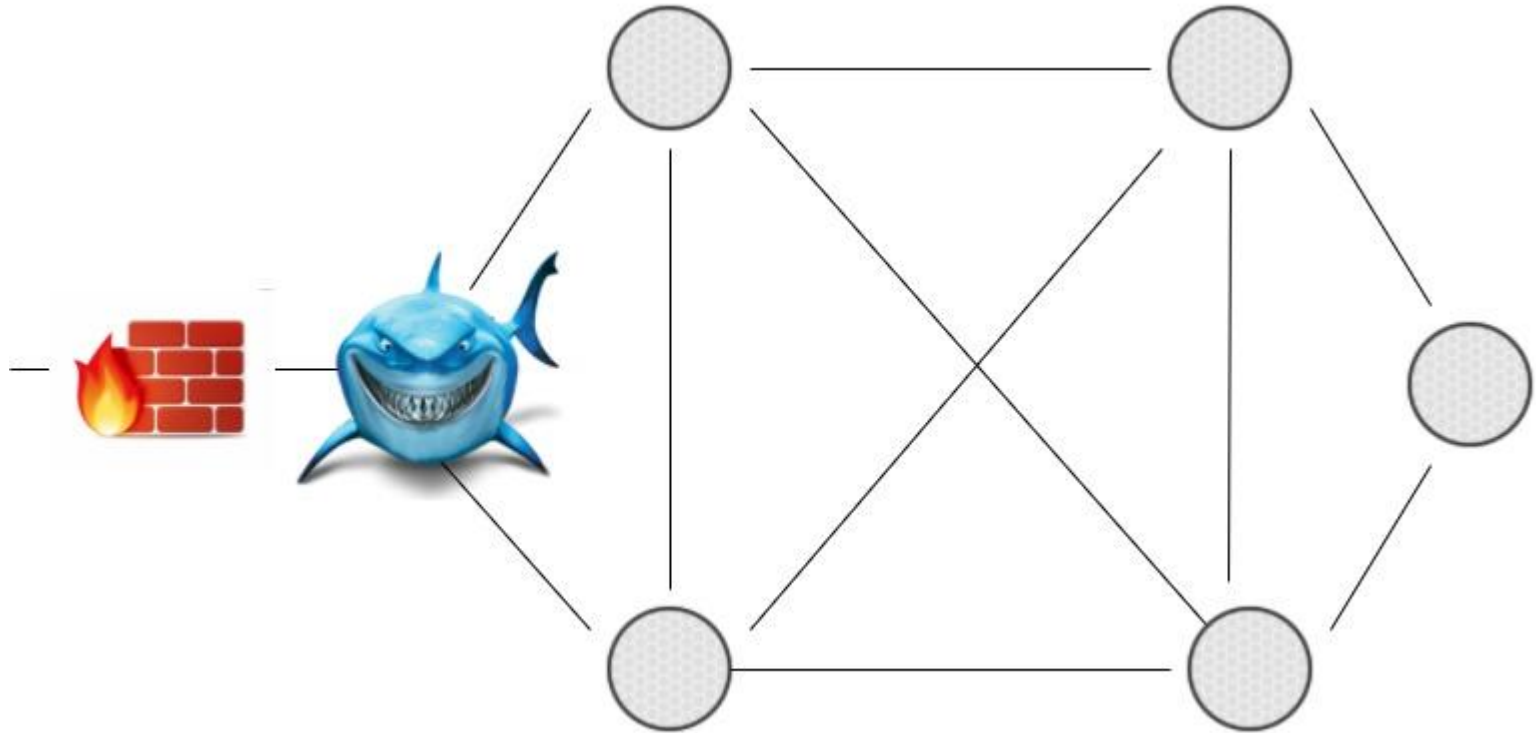
# K8s NetworkPolicy API is a game changer

- Traditional defense



# Kubernetes NetworkPolicy API

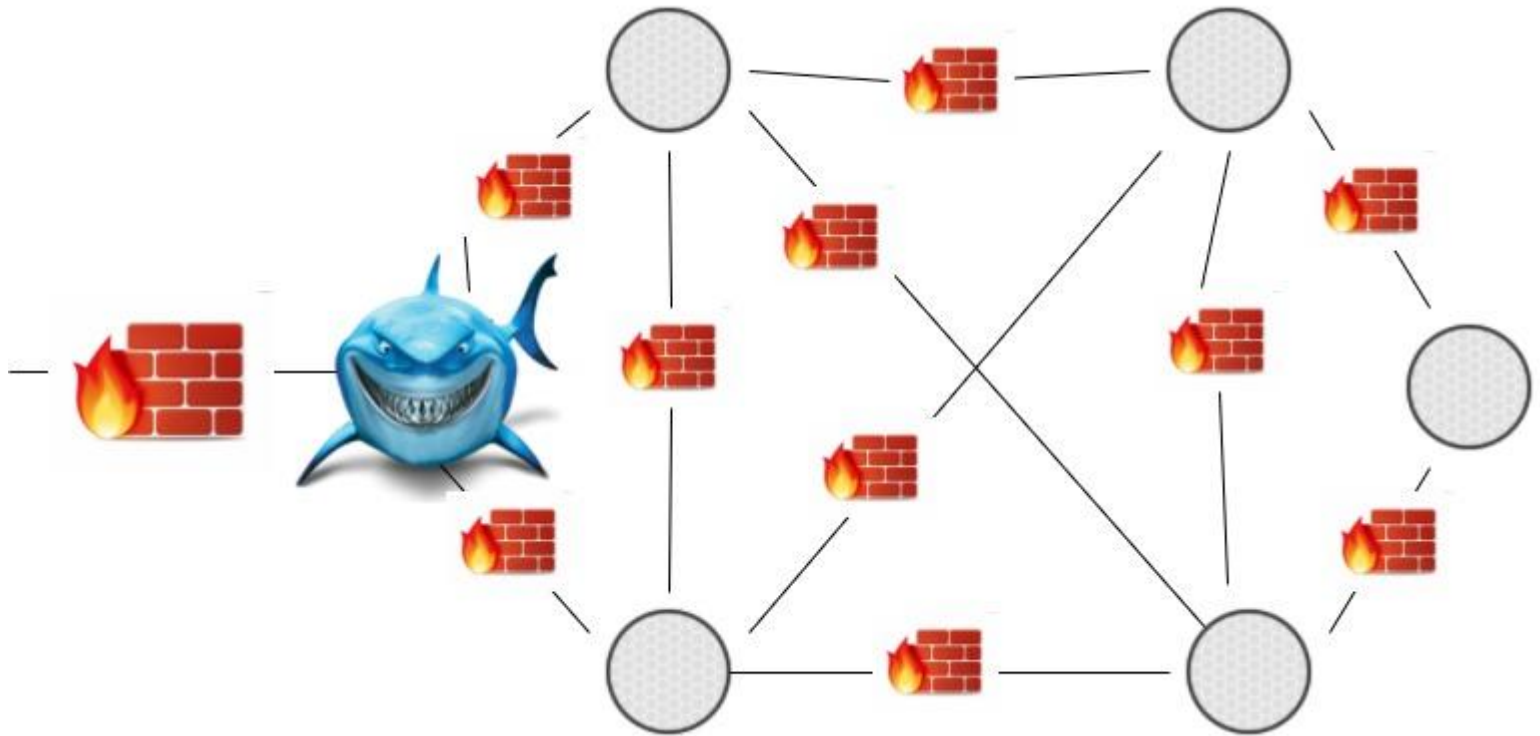
- Problem





# Kubernetes NetworkPolicy API

- Solution



# Monitoring

- Run Datadog Agent as a DaemonSets
  - Collect performance metrics for containers, pods, container namespaces.
  - Create monitors on the status of Kubelets.
  - Visualize Kubernetes cluster performance.
  - Ingest Kubernetes labels as tags in Datadog.



# Logging

- Run fluentd as a pod on each node.
- Gather logs from all containers.
- Export to kafka.
- Kafka to Elasticsearch using logstash.
- Kibana dashboard on top of Elasticsearch



# Sample Kubernetes deployment file

~/paytm-work/paytmall-infrastructure-as-code/kubernetes/urlshortener-api-staging.yaml (terraform) - Sublime Text (UNREGISTERED)

En (100%) 4:36 PM

```

1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    labels:
5      type: local
6    name: urlshortener-api-pv-volume
7  spec:
8    accessModes:
9      - ReadWriteOnce
10   capacity:
11     storage: 1Gi
12   hostPath:
13     path: /data/log/urlshortener-api
14   ---
15  apiVersion: v1
16  kind: PersistentVolumeClaim
17  metadata:
18    name: urlshortener-api-pv-claim
19  spec:
20    accessModes:
21      - ReadWriteOnce
22    resources:
23      requests:
24        storage: 1Gi
25    ---
26  apiVersion: v1
27  data:
28    URLSHORTENER_API_ENV: staging
29    URLSHORTENER_API_MONGO_HOST: 172.26.131.199,172.26.130.170,172.26.131.37
30    URLSHORTENER_API_MONGO_PORT: '27017'
31    URLSHORTENER_API_REDIS_HOST: 172.26.131.124
32    URLSHORTENER_API_REDIS_PORT: '6379'
33  kind: ConfigMap
34  metadata:
35    name: urlshortener-api-config
36    namespace: default
37  ---
38  apiVersion: v1

```

# Deployment file cont...

~/paytm-work/paytm-mall-infrastructure-as-code/kubernetes/urlshortener-api-staging.yaml (terraform) - Sublime Text (UNREGISTERED)

En (100%) 4:36 PM

FOLD init.yml x pmpipeline.groovy x urlshortener-api-prod.yaml x template.py x urlshortener-api-staging.yaml x pmpipelineDeploy.groovy x

```
37 ---
38 apiVersion: v1
39 kind: Service
40 metadata:
41   annotations:
42     service.beta.kubernetes.io/aws-load-balancer-backend-protocol: https
43     service.beta.kubernetes.io/aws-load-balancer-ssl-cert: /arn/
44     service.beta.kubernetes.io/aws-load-balancer-ssl-ports: '443'
45   labels:
46     service: urlshortener-api
47   name: urlshortener-api-extelb
48   namespace: default
49 spec:
50   ports:
51   - name: https
52     port: 443
53     protocol: TCP
54     targetPort: 80
55   selector:
56     service: urlshortener-api
57   type: LoadBalancer
58 ---
59 apiVersion: extensions/v1beta1
60 kind: Deployment
61 metadata:
62   name: urlshortener-api-deployment
63   namespace: default
64 spec:
65   minReadySeconds: 30
66   replicas: 1
67   revisionHistoryLimit: 1
68   selector:
69     matchLabels:
70       service: urlshortener-api
71   strategy:
72     rollingUpdate:
73       maxSurge: 1
74       maxUnavailable: 0
```

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: urlshortener-api-deployment
  namespace: default
spec:
  minReadySeconds: 30
  replicas: 1
  revisionHistoryLimit: 1
  selector:
    matchLabels:
      service: urlshortener-api
  strategy:
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 0
```

# Deployment file cont...

/paytm-work/paytmmall-infrastructure-as-code/kubernetes/urlshortener-api-staging.yaml (terraform) - Sublime Text (UNREGISTERED)

En (100%) 4:36 PM

FOLD init.yml x pmpipeline.groovy x urlshortener-api-prod.yaml x template.py x urlshortener-api-staging.yaml x pmpipelineDeploy.groovy x

```
112 containers:
113 - env:
114   - name: MY_POD_NAME
115     valueFrom:
116       fieldRef:
117         fieldPath: metadata.name
118   - name: MY_POD_IP
119     valueFrom:
120       fieldRef:
121         fieldPath: status.podIP
122   - name: URLSHORTENER_API_REDIS_HOST
123     valueFrom:
124       configMapKeyRef:
125         key: URLSHORTENER_API_REDIS_HOST
126         name: urlshortener-api-config
127   - name: URLSHORTENER_API_REDIS_PORT
128     valueFrom:
129       configMapKeyRef:
130         key: URLSHORTENER_API_REDIS_PORT
131         name: urlshortener-api-config
132   - name: URLSHORTENER_API_MONGO_PORT
133     valueFrom:
134       configMapKeyRef:
135         key: URLSHORTENER_API_MONGO_PORT
136         name: urlshortener-api-config
137   - name: URLSHORTENER_API_ENV
138     valueFrom:
139       configMapKeyRef:
140         key: URLSHORTENER_API_ENV
141         name: urlshortener-api-config
142   - name: URLSHORTENER_API_MONGO_HOST
143     valueFrom:
144       configMapKeyRef:
145         key: URLSHORTENER_API_MONGO_HOST
146         name: urlshortener-api-config
147   - name: URLSHORTENER_API_MONGO_USER
148     valueFrom:
149       configMapKeyRef:
```

# What and Why Service definition file?

A yaml file contains all the information related to any microservice.

Information such as -

- Component & Subcomponent name
- Organization, team email, lead email, team slack or pagerduty
- Kubernetes pod/container, service and environment vars definition
- Other services on which this microservice depends
- Info related to number of containers, CPU & Memory info
- Build & Deploy related info i.e. repo\_url, language, version, jenkins deploy users list
- Database and any storage info that microservice uses and their infra info.



# Sample Service definition file

init.yml

```
1 ---
2 organization: paytm
3 component: urlshortener
4 lead_email: lead_urlshortener@paytm.com
5 team_email: team_urlshortener@paytm.com
6 team_slack: urlshortener
7 team_pagerduty: urlshortener
8 business_unit: urlshortener
9 subcomponents: api
10 containerspec:
11   - name: urlshortener-api
12     build:
13       language: java|node
14       language_version: 1.8|8.5.0
15     dockerregistry_url: dockerregistry.paytmall.io/paytmall
16     port:
17       - 8080
18     ping_api:
19       url: /ping
20       port: 80
21       initial_delay_seconds: 30
22     health_api:
23       url: /health
24       port: 80
25       initial_delay_seconds: 15
26     logs:
27       - /log/urlshortener-api
28     volumes: #custom volume mount
29       - name: urlshortener-api-files-storage
30         host_path: /data/files/urlshortener-api
31         container_path: /apps/urlshortener-api/files/
32 dependencies:
33   - component: catalogplus
34     subcomponent: adapter
35
```

Infrastructure  
as  
code



# Environment specific service definition file



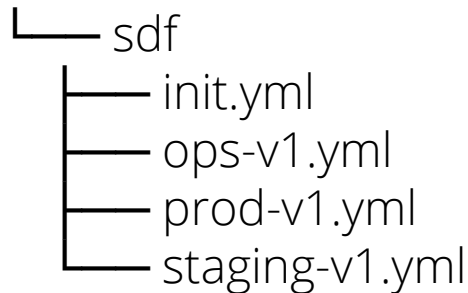
```
staging-v1.yml x
1 k8s:
2   service:
3     - name: urlshortener-api-2.paytmall.com
4       loadbalancerprotocol: https
5       type: internetloadbalancer
6     ports:
7       - name: https
8         port: 443
9         targetPort: 80
10        protocol: "TCP"
11      cert_path: /arn/
12   scale:
13     min_instance_count: 1
14     max_instance_count: 10
15     min_memory: 0.5      #request memory for container
16     max_memory: 1        #limit memory
17     min_cpu: 0.5         #request cpu for container
18     max_cpu: 2           #limit cpu
19   envvars:
20     v1:
21       non_secrets:
22         URLSHORTENER_API_REDIS_HOST: "172.26.131.124"
23         URLSHORTENER_API_REDIS_PORT: "6379"
24       secrets:
25         - URLSHORTENER_API_MONGO_USER
26         - URLSHORTENER_API_MONGO_PASSWORD
27   users:
28     approvers:
29       - animesh.ray
30       - ruchis2.singh
31       - abhishek11.singh
32     deploy_users:
33       - aditya.raj
34   infra:
35     databases:
36       - name: redisapiurlshortener
37         type: redis
38         port: 6379
39         version:
40           v1:
41             scale:
42               min_instance_count: 1
43               max_instance_count: 2
44               min_memory: 4
45               min_cpu: 2
46             instancetype: t2.medium
```

## Current usage of service definition file

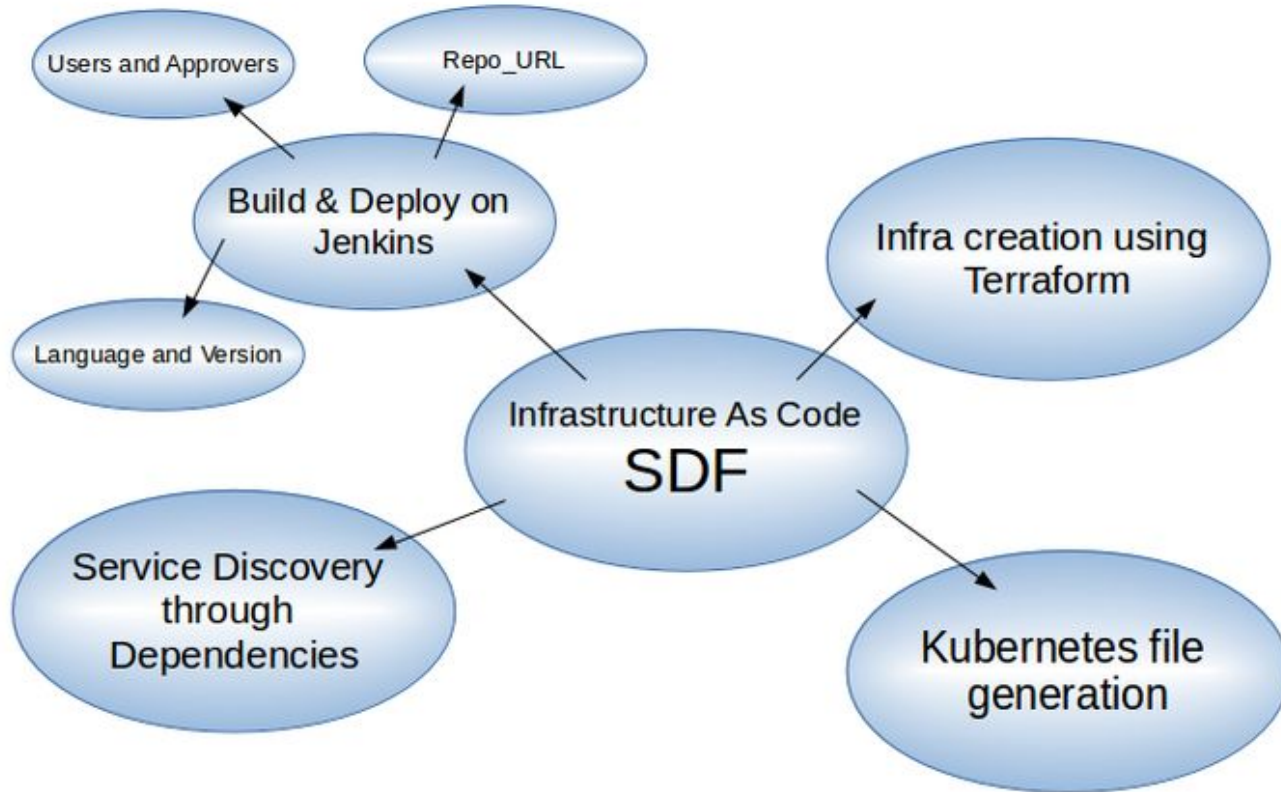
Right now, It is used to generate kubernetes deployment file for each microservice to deploy it on any environment in paytm mall new infrastructure.

### Where SDF files has been placed?

Inside each service repo, at root location -  
infra/



# Scope of service definition file



# Thank you!

Pranshu Saxena  
Ruchi Singh

[pranshu.saxena@paytm.com](mailto:pranshu.saxena@paytm.com)

GO **BIG** OR  
GO Home