

Tungsten Fabric Operator User APIs

Purpose 1

Custom Controllers 2

Manager 2

CommonConfiguration 2

Services 2

KeystoneSecretName 3

Config 3

CommonConfiguration 3

ServiceConfiguration 4

Controls 5

Kubemanagers 6

Vrouters 6

Webui 7

Cassandras 7

Zookeepers 8

Rabbitmq 8

Purpose

This document provides an aggregated overview of the user exposed APIs of the Tungsten Fabric Operator. The code can be viewed at:

<https://github.com/Juniper/contrail-operator/blob/master/pkg/apis/contrail/v1alpha1/>

Custom Controllers

The Spec of each Custom Controller has a CommonConfiguration and a Service specific configuration field. CommonConfiguration defines scale, host network, tolerations, node selector and image pull secrets. The CommonConfiguration can be set centrally in the Manager and used by all other services. However, each service can individually override the CommonConfiguration.

Manager

The Manager Custom Controller is the main interface to users. All subsequent Custom Controllers can be managed via the Manager Custom Controller.

```
type ManagerSpec struct {  
    CommonConfiguration ManagerConfiguration `json:"commonConfiguration,omitEmpty"`  
    Services             Services    `json:"services,omitEmpty"`  
    KeystoneSecretName string      `json:"keystoneSecretName,omitEmpty"`  
}
```

CommonConfiguration

These configurations are used subsequently by all services IF not overridden by the service.

```
type ManagerConfiguration struct {  
    NodeSelector map[string]string `json:"nodeSelector,omitEmpty" protobuf:"bytes,7,rep,name=nodeSelector"`  
    HostNetwork  *bool `json:"hostNetwork,omitEmpty" protobuf:"varint,11,opt,name=hostNetwork"`
```

```

ImagePullSecrets []string `json:"imagePullSecrets,omitempty"`

Tolerations []corev1.Toleration `json:"tolerations,omitempty" protobuf:"bytes,22,opt,name=tolerations"`
}

```

Services

Services define the individual services with their service specific configurations.

```

type Services struct {
    Config      *Config      `json:"config,omitempty"`
    Controls    []*Control    `json:"controls,omitempty"`
    Kubemanagers []*Kubemanager `json:"kubemanagers,omitempty"`
    Webui       *Webui       `json:"webui,omitempty"`
    Vrouters    []*Vrouter    `json:"vrouters,omitempty"`
    Cassandras []*Cassandra  `json:"cassandras,omitempty"`
    Zookeepers  []*Zookeeper  `json:"zookeepers,omitempty"`
    Rabbitmq    *Rabbitmq    `json:"rabbitmq,omitempty"`
    ProvisionManager *ProvisionManager `json:"provisionManager,omitempty"`
    Command     *Command     `json:"command,omitempty"`
    Postgres    *Postgres    `json:"postgres,omitempty"`
    Keystone    *Keystone    `json:"keystone,omitempty"`
    Swift       *Swift       `json:"swift,omitempty"`
    Memcached   *Memcached   `json:"memcached,omitempty"`
    Contrailmonitor *Contrailmonitor `json:"contrailmonitor,omitempty"`
    ContrailCNIs []*ContrailCNI `json:"contrailCNIs,omitempty"`
}

```

KeystoneSecretName

The name for the keystone secret

Config

Config defines the expected state of the Config service

```

type ConfigSpec struct {
    CommonConfiguration PodConfiguration `json:"commonConfiguration,omitempty"`
    ServiceConfiguration ConfigConfiguration `json:"serviceConfiguration"`
}

```

CommonConfiguration

```

type PodConfiguration struct {

    // NodeSelector is a selector which must be true for the pod to fit on a node.

    // Selector which must match a node's labels for the pod to be scheduled on that node.

    // More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/.

    // +optional

    NodeSelector map[string]string `json:"nodeSelector,omitempty" protobuf:"bytes,7,rep,name=nodeSelector"`
}

```

```

// Host networking requested for this pod. Use the host's network namespace.

// If this option is set, the ports that will be used must be specified.

// Default to false.

// +k8s:conversion-gen=false

// +optional

HostNetwork *bool `json:"hostNetwork,omitempty" protobuf:"varint,11,opt,name=hostNetwork"`

// ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec.

ImagePullSecrets []string `json:"imagePullSecrets,omitempty"`

// If specified, the pod's tolerations.

// +optional

Tolerations []corev1.Toleration `json:"tolerations,omitempty" protobuf:"bytes,22,opt,name=tolerations"`

// Number of desired pods. This is a pointer to distinguish between explicit

// zero and not specified. Defaults to 1.

// +optional

Replicas *int32 `json:"replicas,omitempty" protobuf:"varint,1,opt,name=replicas"`
}

```

ServiceConfiguration

Exposes the configuration of the Config service to the user. All settings are **optional**, in case they are not specified, defaults will be used or the setting is retrieved from run time state of other services/

```

type ConfigConfiguration struct {

    Containers          []*Container    `json:"containers,omitempty"`

    APIPort             *int           `json:"apiPort,omitempty"`

    AnalyticsPort       *int           `json:"analyticsPort,omitempty"`

    CollectorPort       *int           `json:"collectorPort,omitempty"`

    RedisPort           *int           `json:"redisPort,omitempty"`

    ApiIntrospectPort   *int           `json:"apiIntrospectPort,omitempty"`

    SchemaIntrospectPort *int           `json:"schemaIntrospectPort,omitempty"`

    DeviceManagerIntrospectPort *int           `json:"deviceManagerIntrospectPort,omitempty"`

    SvcMonitorIntrospectPort *int           `json:"svcMonitorIntrospectPort,omitempty"`

    AnalyticsApiIntrospectPort *int           `json:"analyticsMonitorIntrospectPort,omitempty"`

    CollectorIntrospectPort *int           `json:"collectorMonitorIntrospectPort,omitempty"`

    CassandraInstance   string         `json:"cassandraInstance,omitempty"`

    ZookeeperInstance   string         `json:"zookeeperInstance,omitempty"`

    NodeManager         *bool          `json:"nodeManager,omitempty"`

    RabbitmqUser        string         `json:"rabbitmqUser,omitempty"`

    RabbitmqPassword    string         `json:"rabbitmqPassword,omitempty"`

    RabbitmqVhost       string         `json:"rabbitmqVhost,omitempty"`

    LogLevel            string         `json:"logLevel,omitempty"`

    KeystoneSecretName   string         `json:"keystoneSecretName,omitempty"`

    KeystoneInstance     string         `json:"keystoneInstance,omitempty"`
}

```

```

AuthMode           AuthenticationMode `json:"authMode,omitempty"`
AAAMode            AAAMode           `json:"aaaMode,omitempty"`
Storage            Storage           `json:"storage,omitempty"`
FabricMgmtIP       string            `json:"fabricMgmtIP,omitempty"`

// Time (in hours) that the analytics object and log data stays in the Cassandra database. Defaults to 48 hours.
AnalyticsDataTTL *int `json:"analyticsDataTTL,omitempty"`

// Time (in hours) the analytics config data entering the collector stays in the Cassandra database. Defaults to 2160 hours.
AnalyticsConfigAuditTTL *int `json:"analyticsConfigAuditTTL,omitempty"`

// Time to live (TTL) for statistics data in hours. Defaults to 4 hours.
AnalyticsStatisticsTTL *int `json:"analyticsStatisticsTTL,omitempty"`

// Time to live (TTL) for flow data in hours. Defaults to 2 hours.
AnalyticsFlowTTL *int `json:"analyticsFlowTTL,omitempty"`
}

```

Controls

```

type ControlConfiguration struct {
    Containers    []*Container `json:"containers,omitempty"`
    CassandraInstance string      `json:"cassandraInstance,omitempty"`
    BGPPort       *int        `json:"bgpPort,omitempty"`
    ASNNumber     *int        `json:"asnNumber,omitempty"`
    XMPPPort      *int        `json:"xmppPort,omitempty"`
    DNSPort       *int        `json:"dnsPort,omitempty"`
    DNSIntrospectPort *int      `json:"dnsIntrospectPort,omitempty"`
    NodeManager   *bool       `json:"nodeManager,omitempty"`
    RabbitmqUser  string      `json:"rabbitmqUser,omitempty"`
    RabbitmqPassword string    `json:"rabbitmqPassword,omitempty"`
    RabbitmqVhost string      `json:"rabbitmqVhost,omitempty"`

    // DataSubnet allow to set alternative network in which control, nodemanager
    // and dns services will listen. Local pod address from this subnet will be
    // discovered and used both in configuration for hostip directive and provision
    // script.
    // +kubebuilder:validation:Pattern=`^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)((/([0-2]|2[0-9]|1[0-9]|[0-9]))?)$`
    DataSubnet string `json:"dataSubnet,omitempty"`
}

```

Kubemanagers

```

type KubemanagerConfiguration struct {
    Containers    []*Container `json:"containers,omitempty"`
    CassandraInstance string      `json:"cassandraInstance,omitempty"`
    ZookeeperInstance string    `json:"zookeeperInstance,omitempty"`
    UseKubeadmConfig *bool      `json:"useKubeadmConfig,omitempty"`
}

```

```

ServiceAccount    string    `json:"serviceAccount,omitempty"`
ClusterRole       string    `json:"clusterRole,omitempty"`
ClusterRoleBinding string    `json:"clusterRoleBinding,omitempty"`
CloudOrchestrator string    `json:"cloudOrchestrator,omitempty"`
KubernetesAPIServer string    `json:"kubernetesAPIServer,omitempty"`
KubernetesAPIPort  *int      `json:"kubernetesAPIPort,omitempty"`
KubernetesAPISSLPort *int      `json:"kubernetesAPISSLPort,omitempty"`
PodSubnets       string    `json:"podSubnets,omitempty"`
ServiceSubnets   string    `json:"serviceSubnets,omitempty"`
KubernetesClusterName string    `json:"kubernetesClusterName,omitempty"`
IPFabricSubnets  string    `json:"ipFabricSubnets,omitempty"`
IPFabricForwarding *bool     `json:"ipFabricForwarding,omitempty"`
IPFabricSnat      *bool     `json:"ipFabricSnat,omitempty"`
KubernetesTokenFile string    `json:"kubernetesTokenFile,omitempty"`
HostNetworkService *bool     `json:"hostNetworkService,omitempty"`
RabbitmqUser      string    `json:"rabbitmqUser,omitempty"`
RabbitmqPassword  string    `json:"rabbitmqPassword,omitempty"`
RabbitmqVhost     string    `json:"rabbitmqVhost,omitempty"`
}

```

Vrouters

```

type VrouterConfiguration struct {
    Containers    []*Container    `json:"containers,omitempty"`
    ControllInstance string          `json:"controllInstance,omitempty"`
    CassandraInstance string          `json:"cassandraInstance,omitempty"`
    Gateway       string          `json:"gateway,omitempty"`
    PhysicalInterface string        `json:"physicalInterface,omitempty"`
    MetaDataSecret string          `json:"metaDataSecret,omitempty"`
    NodeManager   *bool           `json:"nodeManager,omitempty"`
    Distribution   *Distribution    `json:"distribution,omitempty"`
    ServiceAccount string           `json:"serviceAccount,omitempty"`
    ClusterRole    string           `json:"clusterRole,omitempty"`
    ClusterRoleBinding string        `json:"clusterRoleBinding,omitempty"`
    VrouterEncryption bool           `json:"vrouterEncryption,omitempty"`
    ContrailStatusImage string        `json:"contrailStatusImage,omitempty"`
    StaticConfiguration *VrouterStaticConfiguration `json:"staticConfiguration,omitempty"`
}

```

Webui

```

type WebuiConfiguration struct {
    Containers    []*Container `json:"containers,omitempty"`

```

```

CassandraInstance string `json:"cassandraInstance,omitempty"`
ServiceAccount   string `json:"serviceAccount,omitempty"`
ClusterRole      string `json:"clusterRole,omitempty"`
ClusterRoleBinding string `json:"clusterRoleBinding,omitempty"`
KeystoneSecretName string `json:"keystoneSecretName,omitempty"`
KeystoneInstance string `json:"keystoneInstance,omitempty"`
}

```

Cassandras

```

type CassandraConfiguration struct {
    Containers []Container `json:"containers,omitempty"`
    ClusterName string `json:"clusterName,omitempty"`
    ListenAddress string `json:"listenAddress,omitempty"`
    Port *int `json:"port,omitempty"`
    CqlPort *int `json:"cqlPort,omitempty"`
    SslStoragePort *int `json:"sslStoragePort,omitempty"`
    StoragePort *int `json:"storagePort,omitempty"`
    JmxLocalPort *int `json:"jmxLocalPort,omitempty"`
    MaxHeapSize string `json:"maxHeapSize,omitempty"`
    MinHeapSize string `json:"minHeapSize,omitempty"`
    StartRPC *bool `json:"startRPC,omitempty"`
    Storage Storage `json:"storage,omitempty"`
}

```

Zookeepers

```

type ZookeeperConfiguration struct {
    Containers []Container `json:"containers,omitempty"`
    ClientPort *int `json:"clientPort,omitempty"`
    ElectionPort *int `json:"electionPort,omitempty"`
    ServerPort *int `json:"serverPort,omitempty"`
    AdminEnableServer *bool `json:"adminEnabled,omitempty"`
    AdminPort *int `json:"adminPort,omitempty"`
    Storage Storage `json:"storage,omitempty"`
}

```

Rabbitmq

```

type RabbitmqConfiguration struct {
    Containers []Container `json:"containers,omitempty"`
    Port *int `json:"port,omitempty"`
    SSLPort *int `json:"sslPort,omitempty"`
    ErlangCookie string `json:"erlangCookie,omitempty"`
}

```

```
Vhost    string    `json:"vhost,omitempty"`
User     string    `json:"user,omitempty"`
Password string    `json:"password,omitempty"`
Secret   string    `json:"secret,omitempty"`
}
```