6.2. Настройка внешнего доступа: Ingress-nginx

- TLS termination

also

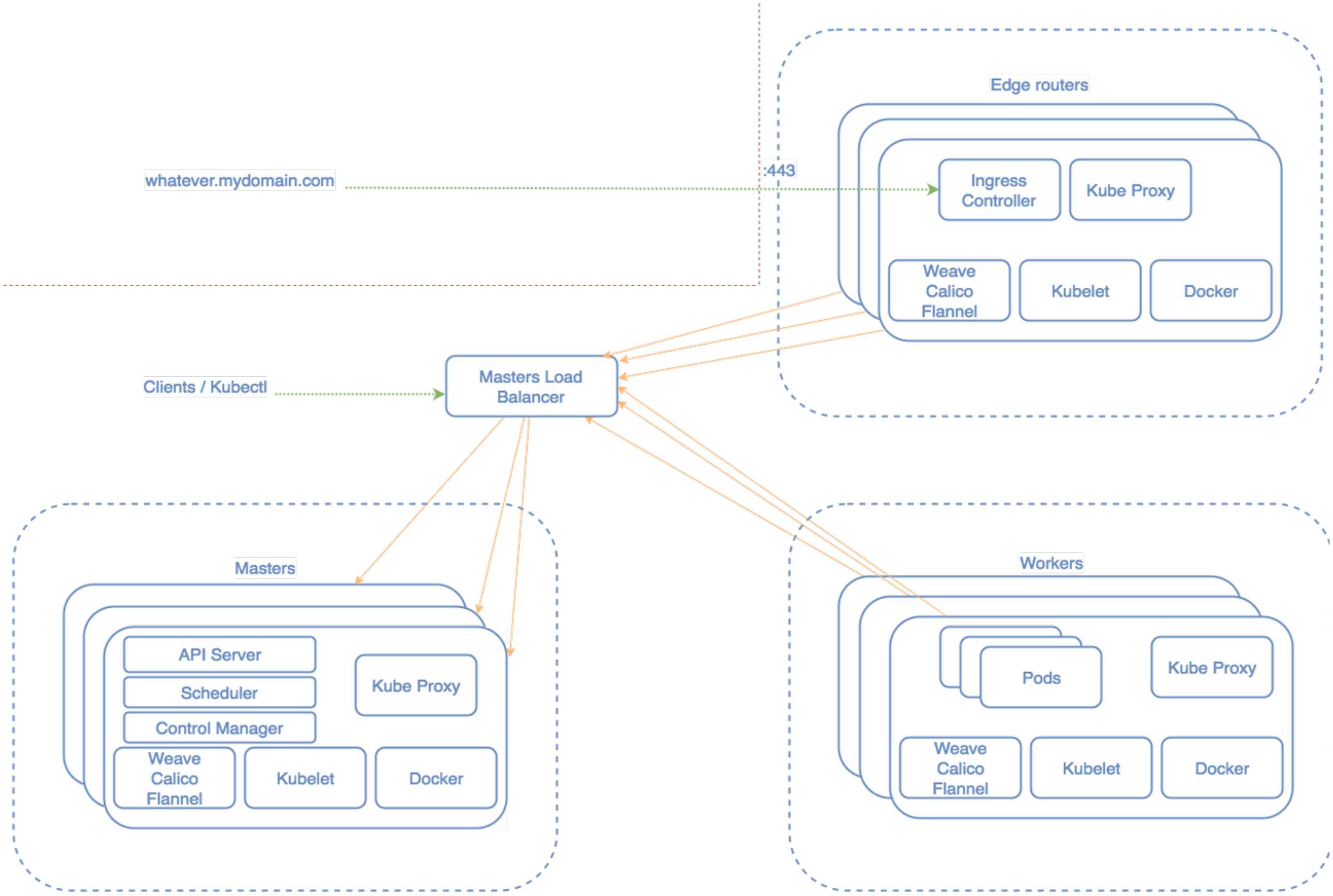
https://capgemini.github.io/kubernetes/kube-traefik/

https://kubernetes.github.io/ingress-nginx/deploy/

Ingress – это набор правил внутри вашего кластера, предназначенных для того, чтобы входящие подключения могли достичь сервисов (**Services**) ваших приложений.

В Kubernetes **Ingress** работает в паре с **IngressController**-ом следующим образом: в **Ingress** описываются правила балансировки трафика до сервисов ваших приложений, применяемые внутри балансировщика, который представляет собой ничто иное, как отдельный Pod с упакованным в него Nginx, Traefik и т.д.

IngressController - это deployment с внешним ПО: **nginx**, **traefik**, etc



Для этого нужно добавить в кластер отдельную ноду исключительно для запуска **IngressController** (k8sedge1.vsk.ru) и запретить на нем запуск других подов:

```
kubectl drain k8sedge1 --ignore-daemonsets
kubectl label nodes k8sedge1 role=edge-router
kubectl label nodes k8sn01 role=worker
```

После развертывания **IngressController** оно будет постоянно сканировать Kubernetes в поисках новых Ingress. При обнаружении будет добавлять в конфигурацию балансировщика новый endpoint из соответствующего Service.

Вот полная конфигурация всего, что нужно для IngressController (файл https://gitlab.vsk.ru/infra/kubernetes/blob/master/sample/nginx-ingress-controller.yaml):

```
---
apiVersion: v1
kind: Namespace
metadata:
  name: ingress-nginx
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: default-http-backend
  labels:
    app.kubernetes.io/name: default-http-backend
    app.kubernetes.io/part-of: ingress-nginx
  namespace: ingress-nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: default-http-backend
  template:
    metadata:
      labels:
        app.kubernetes.io/name: default-http-backend
        app.kubernetes.io/part-of: ingress-nginx
    spec:
      terminationGracePeriodSeconds: 60
      containers:
      - name: default-http-backend
        # Any image is permissible as long as:
        # 1. It serves a 404 page at /
        # 2. It serves 200 on a /healthz endpoint
        image: gcr.io/google_containers/defaultbackend:1.4
        livenessProbe:
          httpGet:
            path: /healthz
            port: 8080
            scheme: HTTP
          initialDelaySeconds: 30
          timeoutSeconds: 5
        ports:
        - containerPort: 8080
        resources:
          limits:
            cpu: 10m
            memory: 20Mi
          requests:
            cpu: 10m
            memory: 20Mi
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: default-http-backend
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: default-http-backend
    app.kubernetes.io/part-of: ingress-nginx
spec:
  ports:
  - port: 80
    targetPort: 8080
  selector:
    app.kubernetes.io/name: default-http-backend
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-configuration
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: tcp-services
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: udp-services
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress-serviceaccount
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
---
apiVersion: rbac.authorization.k8s.io/v1beta1
```

```
kind: ClusterRole
metadata:
  name: nginx-ingress-clusterrole
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
rules:
  - apiGroups:
      - ""
    resources:
      - configmaps
      - endpoints
      - nodes
      - pods
      - secrets
    verbs:
      - list
      - watch
  - apiGroups:
      - ""
    resources:
      - nodes
    verbs:
      - get
  - apiGroups:
      - ""
    resources:
      - services
    verbs:
      - get
      - list
      - watch
  - apiGroups:
      - "extensions"
    resources:
      - ingresses
    verbs:
      - get
      - list
      - watch
  - apiGroups:
      - ""
    resources:
        - events
    verbs:
        - create
        - patch
  - apiGroups:
      - "extensions"
    resources:
      - ingresses/status
    verbs:
      - update
```

```
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: Role
metadata:
  name: nginx-ingress-role
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
rules:
  - apiGroups:
      - ""
    resources:
      - configmaps
      - pods
      - secrets
      - namespaces
    verbs:
      - get
  - apiGroups:
      - ""
    resources:
      - configmaps
    resourceNames:
      # Defaults to "<election-id>-<ingress-class>"
      # Here: "<ingress-controller-leader>-<nginx>"
      # This has to be adapted if you change either parameter
      # when launching the nginx-ingress-controller.
      - "ingress-controller-leader-nginx"
    verbs:
      - get
      - update
  - apiGroups:
      - ""
    resources:
      - configmaps
    verbs:
      - create
  - apiGroups:
      - ""
    resources:
      - endpoints
    verbs:
      - get
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: RoleBinding
metadata:
  name: nginx-ingress-role-nisa-binding
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
```

```
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: nginx-ingress-role
subjects:
  - kind: ServiceAccount
    name: nginx-ingress-serviceaccount
    namespace: ingress-nginx
---
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRoleBinding
metadata:
  name: nginx-ingress-clusterrole-nisa-binding
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: nginx-ingress-clusterrole
subjects:
  - kind: ServiceAccount
    name: nginx-ingress-serviceaccount
    namespace: ingress-nginx
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx-ingress-controller
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app.kubernetes.io/name: ingress-nginx
  template:
    metadata:
      labels:
        app.kubernetes.io/name: ingress-nginx
      annotations:
        prometheus.io/port: '10254'
        prometheus.io/scrape: 'true'
    spec:
      nodeSelector:
        role: edge-router
      serviceAccountName: nginx-ingress-serviceaccount
      containers:
        - name: nginx-ingress-controller
          image:
quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.19.0
```

```
          args:
            - /nginx-ingress-controller
            -
--default-backend-service=$(POD_NAMESPACE)/default-http-backend
            - --configmap=$(POD_NAMESPACE)/nginx-configuration
            - --tcp-services-configmap=$(POD_NAMESPACE)/tcp-services
            - --udp-services-configmap=$(POD_NAMESPACE)/udp-services
            - --publish-service=$(POD_NAMESPACE)/ingress-nginx
            - --annotations-prefix=nginx.ingress.kubernetes.io
          securityContext:
            capabilities:
                drop:
                - ALL
                add:
                - NET_BIND_SERVICE
            # www-data -> 33
            runAsUser: 33
          env:
            - name: POD_NAME
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
            - name: POD_NAMESPACE
              valueFrom:
                fieldRef:
                  fieldPath: metadata.namespace
          ports:
          - name: http
            containerPort: 80
            hostPort: 80
          - name: https
            containerPort: 443
            hostPort: 443
          livenessProbe:
            failureThreshold: 3
            httpGet:
              path: /healthz
              port: 10254
              scheme: HTTP
            initialDelaySeconds: 10
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 1
          readinessProbe:
            failureThreshold: 3
            httpGet:
              path: /healthz
              port: 10254
              scheme: HTTP
            periodSeconds: 10
            successThreshold: 1
            timeoutSeconds: 1
---
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: ingress-nginx
  namespace: ingress-nginx
  labels:
    app.kubernetes.io/name: ingress-nginx
    app.kubernetes.io/part-of: ingress-nginx
spec:
  type: NodePort
  ports:
  - name: http
    port: 80
    targetPort: 80
    nodePort: 80
    protocol: TCP
  - name: https
    port: 443
    targetPort: 443
    nodePort: 443
    protocol: TCP
  selector:
    app.kubernetes.io/name: ingress-nginx
```

TLS termination

Сгенерировать сертификат:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out
tls.crt -subj "/CN=kubernetes.vsk.ru/O=nginx"
kubectl create secret tls tls-dashboard-secret --key tls.key --cert tls.crt
--namespace kube-system
```

И сохранить его для использования в yaml описании:

```
$ kubectl get secret tls-dashboard-secret --namespace kube-system -o yaml
apiVersion: v1
data:
  tls.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURLekNDQWhPZ0F3SUJBZ0lKQU9saHNxMD
FyUGRnTUEwR0NTcUdTSWIzRFFFQkN3VUFNQ3d4R2pBWUJnTlYKQkFNTUVXdDFZbVZ5Ym1WMFpY
TXVkbk5yTG5KMU1RNHdEQVlEVlFRS0RBVnVaMmx1ZURBZUZ3MHhPREE1TVRFdwpPRFEwTWpOYU
Z3MHhPVEE1TVRFd09EUTBNak5hTUN3eEdqQVlCZ05WQkFNTUVXdDFZbVZ5Ym1WMFpYTXVkbk5y
CkxuSjFNUTR3REFZRFZRUUtEQVZ1WjJsdWVEQ0NBU0l3RFFZSktvWklodmNOQVFFQkJRQURnZ0
VQQURDQ0FRb0MKZ2dFQkFMR24vbDk1VSthYnoxUjBLUEpZdkh2eHBMcFhhUCttdzAxdXQzcU8y
VlR2M0t4NGt2Um4xclkrbHFFNwpLNDNQTXNuazVIT09Jd09GcVJrRWwyQkRNVlA0aGxuUDIrbX
BJek94TkJVK0NoVGZlT3VzRHpsdWVwSmNXR3JQCkFwQmN0RXFRb3Y3eXgvOUIzdmVBRUI2NmxM
eHRpcGhDQk5YcXptRDFpZkpaVGlhSnlEQ1VLbGtjcjQ0d3pzdm4KVnRtMnVoaHF3dHRsNWtLUW
5CeVFtN0RPV0pHMTNPYU9xV1NkM0YxRU54SjZBR1BVdUJLTUhTSE5oWE1VTG93UAptNkJiVldG
Y3AzMTh2b2JSUCtUUVRkQlp5UUdLbzlBby85NFBtbVEvVE5mRm1SME9OYWFoZnRyVzJWdXFJOW
VVCllLZGxCM2pwTmRTekgxT0g5L2pWRmxvOGd4TUNBd0VBQWFOUU1FNHdIUVlEVlIwT0JCWUVG
Rm1jUjlmUENYc1YKZlcyQkdUbVlSV0NuYkY3Uk1COEdBMVVkSXdRWU1CYUFGRm1jUjlmUENYc1
ZmVzJCR1RtWVJXQ25iRjdSTUF3RwpBMVVkRXdRRk1BTUJBZjh3RFFZSktvWklodmNOQVFFTEJR
QURnZ0VCQUlOUjFsNjlRSTFaZjcwYVczRzkvYlNKClFwbk1KZy9MSnQ2TUhzN2lkS1pwQkNaSn
pLK0RlRnRoclUxYUVMM1c2U2V5czAwMVJ0NVJPbnlNaU9kUTBleHAKcjVIajBaaVJaVk9vQXRI
WmMwTUl3K2I5QjRsTXEzN3hsMW9NN2NOL1l4enJuTG1qUEExa3ZwdnJJQllJbHM4ZQpoRDhUNz
YxQUNpU2RUYjFSVlFwL2N4U01IVWFpcHZrYWNobjNrWk1sV2FpVVppcysrc3pQTk1FZGs4QXlG
dmZNCnNxUDdMWDRkREJ3M2psK0o3R0FER0VISVdZV2JSZE83WVhmWXZIMHVuS3ZnbTQ1MTJqNE
V2aGlrMkFjeWRLWHcKT0R0eXYrcGREOVRzUU40OE5raHM1b1cxQkFUd3l3WUF0UllmV2grZXdW
RHFSQ05VUVhLWWtnbWxUekgvT0IwPQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
  tls.key:
LS0tLS1CRUdJTiBQUklWQVRFIEtFWS0tLS0tCk1JSUV2UUlCQURBTkJna3Foa2lHOXcwQkFRRU
ZBQVNDQktjd2dnU2pBZ0VBQW9JQkFRQ3hwLzVmZVZQbW04OVUKZENqeVdMeDc4YVM2VjJqL3Bz
Tk5icmQ2anRsVTc5eXNlSkwwWjlhMlBwYWhPeXVOenpMSjVPUnpqaU1EaGFrWgpCSmRnUXpGVC
tJWlp6OXZwcVNNenNUUVZQZ29VMzNqcnJBODVibnFTWEZocXp3S1FYTFJLa0tMKzhzZi9RZDcz
CmdCQWV1cFM4YllxWVFnVFY2czVnOVlueVdVNG1pY2d3bENwWkhLK09NTTdMNTFiWnRyb1lhc0
xiWmVaQ2tKd2MKa0p1d3psaVJ0ZHptanFsa25keGRSRGNTZWdCajFMZ1NqQjBoellWekZDNk1E
NXVnVzFWaFhLZDlmTDZHMFQvawowRTNRV2NrQmlxUFFLUC9lRDVwa1Awelh4WmtkRGpXbW9YN2
ExdGxicWlQWGxHQ25aUWQ0NlRYVXN4OVRoL2Y0CjFSWmFQSU1UQWdNQkFBRUNnZ0VBUDZIK21N
```

```
L3FYMlZibzlKZmdENk1kSkNmQWllZzhrRVhIenNTK1JFRmcwak8KNysvWUl6YTJUMXluczJSd0
8wRUYzZzdnVFF1U093YTdCTDBDUTNhTnlwZnY5MmRlUzVsTGhJNXFLMW1uWHNhVQpsMFZ2eHBC
dDB0TEUwSzBKTEJBb2U3VEd6UzdFdUFjdUI5U1U4WXkvRGFibXg1YU9DTmtyeHNycitDUkl1OE
J4Cld5RnRmR0NCSFJ2aFdaNDJWZmpOWllKQ245a0YxVUJ0MFhKM0Q3b08xTG9RWU5VYmRaTDVk
ekN5T3Q0b3VlWmgKdUd1MnQyMDU2akNINXF3UUtCRHcvLzBERjVrWVI1YmxWTEFtcjlrRzU2SH
Y4T2Y3dzlob29iQ1NvME5BYTNGKwpNWHk2Q3RSQkhiNWVtS3hWT1UweTNhWXUzNElUd2xWTkpQ
OTNSbDMrZ1FLQmdRRGNXbHZjelBscTV3THUwbHBECml1eXluZ0R6bFBqdDgrNlVWOGZSZXpkV2
xlSmpEdUxJQ3Q1QVRCQTJNdkxKRjRDZTZoT3VBNnBaN0x2bEhZL1QKY1pwUTkwMmFUVGJWenIx
Z1U1NnA3WEs5ME85OTBCblJtdGp1aWtwU1lEZ3puUG5sV1RkNXdmWkdHRC9HRXoxaApKZldOQW
FLa1k5VHlob3JtQVY1SWFheWdNd0tCZ1FET1pXY3JOK3lYdHdaMDdoclpqSnlySUI0MXcvbkpv
THBoCnlSbVMyWmxoRFFhR09MYzlwaHIzVXZ3Rkcvd3NTR1B2L2NVY1JBWTd4OWJzTWwyWU1qdF
lkeFIrR1c4K01MbXMKVzg0cVZPUlBJQVV0bW10TittZkRjMEdDeWpKaGtSSkVjcmt0c1dhZzVU
MVlDbjh5a2V1VlRuU0UyKzN6M2x4VApmQmVqYy9ZeG9RS0JnUURM1JXSHVheUlJU3F5MEVzS2
wveWlLS0JteUVNckN5RGRCcE9xYVBDVEhaRktVSXUwCnBBZlBqSFI1dWF5Q3NNb0p2cThRb0NV
eHlPeVF2UWZIMFJyS3B5aDFYL3N5TU5YQUxiS09TM0hBOHRnWHhzUk8KWEVXakRhQ1VJSXNTK2
FyMGEyVm53Mk5xc1Y4VVowZE1kaWtrUFdIN1JsMjI2OHhCZ3N4QjlWRzNpUUtCZ0NpVApYdGpz
UWFNQjVFOFFrRTY4aWdCS2JDWDl4Um1TYzVITnl5MUhnaFBHVDhZTFdGclNVUExXYVoxSGh5Zm
RnMmdtCjU1MUs4ajdHWnNaQWtJM2ZNak9PRllWRTBzM1RIbTE0RHV2YmpKYlJvTEw2YktZRDJH
cGdPanpRSzE5cStjQmoKSFd5U042N2g0bEZ3bnRqK1B1WTNha2pFUTk5eGs0MEdGRStaK0lpQk
FvR0FZSVVQWm8yNStua1BSU1REODBGQQo2cHJHbE41SWZVeDdiSjUwN1dHcS9MNGZzUEdBWFRP
YkZoR1krYWhxcHVQWHlkRXVueDF3UjJVbkdldmVwTmZ1CmxkQ0doVWxpcVNTZHJZWHM2VnhuQV
BtQW42dmMvOXRvTjhGM3U5cnpmNCthZ2FMN21JN2h0dHkxSkFaS0tUdVkKYVhEYTVxOHBNaGJ0
bGVNK2wvZFNuZ289Ci0tLS0tRU5EIFBSSVZBVEUgS0VZLS0tLS0K
kind: Secret
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |

{"apiVersion":"v1","data":{"tls.crt":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t
Ck1JSURLekNDQWhPZ0F3SUJBZ0lKQU9saHNxMDFyUGRnTUEwR0NTcUdTSWIzRFFFQkN3VUFNQ3
d4R2pBWUJnTlYKQkFNTUVXdDFZbVZ5Ym1WMFpYTXVkbk5yTG5KMU1RNHdEQVlEVlFRS0RBVnVa
Mmx1ZURBZUZ3MHhPREE1TVRFdwpPRFEwTWpOYUZ3MHhPVEE1TVRFd09EUTBNak5hTUN3eEdqQV
lCZ05WQkFNTUVXdDFZbVZ5Ym1WMFpYTXVkbk5yCkxxuSjFNUTR3REFZRFZRUUtEQVZ1WjJsdWVE
Q0NBU0l3RFFZSktvWklodmNOQVFFQkJRQURnZ0VQQURDQ0FRb0MKZ2dFQkFMR24vbDk1VSthYn
oxUjBLUEpZdkh2eHBMcFhhUCttdzAxdXQzcU8yVlR2M0t4NGt2Um4xclkrbHFFNwpLNDNQTXNu
azVIT09Jd09GcVJrRWwyQkRNVlA0aGxuUDIrbXBJek94TkJVK0NoVGZlT3VzRHpsdWVwSmNXR3
JQCkFwQmN0RXFRb3Y3eXgvOUIzdmVBRUI2NmxMeHRpcGhDQk5YcXptRDFpZkpaVGlhSnlEQ1VL
bGtjcjQ0d3pzdm4KVnRtMnVoaHF3dHRsNWtLUW5CeVFtN0RPV0pHMTNPYU9xV1NkM0YxRU54Sj
ZBR1BVdUJLTUhTSE5oWE1VTG93UAptNkJiVldGY3AzMTh2b2JSUCtUUVRkQlp5UUdLbzlBby85
NFBtbVEvVE5mRm1SME9OYWFoZnRyVzJWdXFJOWVVCllLZGxCM2pwTmRTekgxT0g5L2pWRmxvOG
d4TUNBd0VBQWFOUU1FNHdIUVlEVlIwT0JCWUVGRm1jUjlmUENYc1YKZlcyQkdUbVlSV0NuYkY3
Uk1COEdBMVVkSXdRWU1CYUFGRm1jUjlmUENYc1ZmVzJCR1RtWVJXQ25iRjdSTUF3RwpBMVVkRX
dRRk1BTUJBZjh3RFFZSktvWklodmNOQVFFTEJRQURnZ0VCQUlOUjFsNjlRSTFaZjcwYVczRzkv
YlNKClFwbk1KZy9MSnQ2TUhzN2lkS1pwQkNaSnpLK0RlRnRoclUxYUVMM1c2U2V5czAwMVJ0NV
JPbnlNaU9kUTBleHAKcjVIajBaaVJaVk9vQXRIWmMwTUl3K2I5QjRsTXEzN3hsMW9NN2NOL1l4
enJuTG1qUEExa3ZwdnJJQllJbHM4ZQpoRDhUNzYxQUNpU2RUYjFSVlFwL2N4U01IVWFpcHZrYW
NobjNrWk1sV2FpVVppcysrc3pQTk1FZGs4QXlGdmZNCnNxUDdMWDRkREJ3M2psK0o3R0FER0VI
SVdZV2JSZE83WVhmWXZIMHVuS3ZnbTQ1MTJqNEV2aGlrMkFjeWRLWHcKT0R0eXYrcGREOVRzUU
40OE5raHM1b1cxQkFUd3l3WUF0UllmV2grZXdWRHFSQ05VUVhLWWtnbWxUekgvT0IwPQotLS0t
LUVORCBDRVJUSUZJQ0FURS0tLS0tCg==","tls.key":"LS0tLS1CRUdJTiBQUklWQVRFIEtFW
S0tLS0tCk1JSUV2UUlCQURBTkJna3Foa2lHOXcwQkFRRUZBQVNDQktjd2dnU2pBZ0VBQW9JQkF
RQ3hwLzVmZVZQbW04OVUKZENqeVdMeDc4YVM2VjJqL3BzTk5icmQ2anRsVTc5eXNlSkwwWjlhM
lBwYWhPeXVOenpMSjVPUnpqaU1EaGFrWgpCSmRnUXpGVCtJWlp6OXZwcVNNenNUUVZQZ29VMzN
qcnJBODVibnFTWEZocXp3S1FYTFJLa0tMKzhzZi9RZDczCmdCQWV1cFM4YllxWVFnVFY2czVnO
VlueVdVNG1pY2d3bENwWkhLK09NTTdMNTFiWnRyb1lhc0xiWmVaQ2tKd2MKa0p1d3psaVJ0ZHp
tanFsa25keGRSRGNTZWdCajFMZ1NqQjBoellWekZDNk1ENXVnVzFWaFhLZDlmTDZHMFQvawowR
TNRV2NrQmlxUFFLUC9lRDVwa1Awelh4WmtkRGpXbW9YN2ExdGxicWlQWGxHQ25aUWQ0NlRYVXN
4OVRoL2Y0CjFSWmFQSU1UQWdNQkFBRUNnZ0VBUDZIK21NL3FYMlZibzlKZmdENk1kSkNmQWllZ
zhrRVhIenNTK1JFRmcwak8KNysvWUl6YTJUMXluczJSd08wRUYzZzdnVFF1U093YTdCTDBDUTN
hTnlwZnY5MmRlUzVsTGhJNXFLMW1uWHNhVQpsMFZ2eHBCdDB0TEUwSzBKTEJBb2U3VEd6UzdFd
UFjdUI5U1U4WXkvRGFibXg1YU9DTmtyeHNycitDUkl1OEJ4Cld5RnRmR0NCSFJ2aFdaNDJWZmp
OWllKQ245a0YxVUJ0MFhKM0Q3b08xTG9RWU5VYmRaTDVkekN5T3Q0b3VlWmgKdUd1MnQyMDU2a
kNINXF3UUtCRHcvLzBERjVrWVI1YmxWTEFtcjlrRzU2SHY4T2Y3dzlob29iQ1NvME5BYTNGKwp
NWHk2Q3RSQkhiNWVtS3hWT1UweTNhWXUzNElUd2xWTkpQOTNSbDMrZ1FLQmdRRGNXbHZjelBsc
TV3THUwbHBECml1eXluZ0R6bFBqdDgrNlVWOGZSZXpkV2xlSmpEdUxJQ3Q1QVRCQTJNdkxKRjR
DZTZoT3VBNnBaN0x2bEhZL1QKY1pwUTkwMmFUVGJWenIxZ1U1NnA3WEs5ME85OTBCblJtdGp1a
WtwU1lEZ3puUG5sV1RkNXdmWkdHRC9HRXoxaApKZldOQWFLa1k5VHlob3JtQVY1SWFheWdNd0t
CZ1FET1pXY3JOK3lYdHdaMDdoclpqSnlySUI0MXcvbkpvTHBoCnlSbVMyWmxoRFFhR09MYzlwa
HIzVXZ3Rkcvd3NTR1B2L2NVY1JBWTd4OWJzTWwyWU1qdFlkeFIrR1c4K01MbXMKVzg0cVZPUlB
JQVV0bW10TittZkRjMEdDeWpKaGtSSkVjcmt0c1dhZzVUMVlDbjh5a2V1VlRuU0UyKzN6M2x4V
ApmQmVqYy9ZeG9RS0JnUURM1JXSHVheUlJU3F5MEVzS2wveWlLS0JteUVNckN5RGRCcE9xYVB
DVEhaRktVSXUwCnBBZlBqSFI1dWF5Q3NNb0p2cThRb0NVeHlPeVF2UWZIMFJyS3B5aDFYL3N5T
U5YQUxiS09TM0hBOHRnWHhzUk8KWEVXakRhQ1VJSXNTK2FyMGEyVm53Mk5xc1Y4VVowZE1kaWt
rUFdIN1JsMjI2OHhCZ3N4QjlWRzNpUUtCZ0NpVApYdGpzUWFNQjVFOFFrRTY4aWdCS2JDWDl4U
m1TYzVITnl5MUhnaFBHVDhZTFdGclNVUExXYVoxSGh5ZmRnMmdtCjU1MUs4ajdHWnNaQWtJM2Z
Nak9PRllWRTBzM1RIbTE0RHV2YmpKYlJvTEw2YktZRDJHcGdPanpRSzE5cStjQmoKSFd5U042N
2g0bEZ3bnRqK1B1WTNha2pFUTk5eGs0MEdGRStaK0lpQkFvR0FZSVVQWm8yNStua1BSU1REODB
GQQo2cHJHbE41SWZVeDdiSjUwN1dHcS9MNGZzUEdBWFRPYkZoR1krYWhxcHVQWHlkRXVueDF3U
jJVbkdldmVwTmZ1CmxkQ0doVWxpcVNTZHJZWHM2VnhuQVBtQW42dmMvOXRvTjhGM3U5cnpmNCt
hZ2FMN21JN2h0dHkxSkFaS0tUdVkKYVhEYTVxOHBNaGJ0bGVNK2wvZFNuZ289Ci0tLS0tRU5EI
FBSSVZBVEUgS0VZLS0tLS0K"},"kind":"Secret","metadata":{"annotations":{},"na
me":"tls-dashboard-secret","namespace":"kube-system"},"type":"kubernetes.i
o/tls"}
```

```
  creationTimestamp: 2018-09-11T09:05:58Z
  name: tls-dashboard-secret
  namespace: kube-system
  resourceVersion: "2000080"
  selfLink: /api/v1/namespaces/kube-system/secrets/tls-dashboard-secret
  uid: e60131de-b5a1-11e8-ae43-0050569156ef
type: kubernetes.io/tls
```

Описание Ingress должно выглядеть так:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  labels:
    k8s-app: kubernetes-dashboard
  name: dashboardingress
  namespace: kube-system
spec:
  tls:
  - secretName: tls-dashboard-secret
  rules:
  - host: kubernetes.vsk.ru
    http:
      paths:
      - path: /
        backend:
          serviceName: kubernetes-dashboard
          servicePort: 443
```

Вот так выглядит соединение:

```
$ curl -kvv https://kubernetes.vsk.ru
* About to connect() to kubernetes.vsk.ru port 443 (#0)
*   Trying 192.168.66.206...
* Connected to kubernetes.vsk.ru (192.168.66.206) port 443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
*  subject: O=nginx,CN=kubernetes.vsk.ru
*  start date:  11 08:44:23 2018 GMT
*  expire date:  11 08:44:23 2019 GMT
*  common name: kubernetes.vsk.ru
*  issuer: O=nginx,CN=kubernetes.vsk.ru
> GET / HTTP/1.1
> User-Agent: curl/7.29.0
> Host: kubernetes.vsk.ru
> Accept: */*
```