

7 июня 2023 📍 Москва, МЦК ЗИЛ

# БЕКОН<sup>23</sup>

Первая в России конференция  
по БЕзопасности КОНтейнеров и контейнерных сред

## Концепция Cluster-API или как разворачивать безопасные кластера

Путилин Дмитрий Львович

Ведущий инженер VK (K8S Platform)



Путилин Дмитрий

Ведущий инженер



## Чем занимаюсь

Разнорабочий: Сопровождение K8S кластеров, разработка архитектурных решений в области автоматизации, NoOPS платформ на базе K8S in K8S и Host Based Firewall

## Опыт

**2019-2022 TechLead Devops в Сбер**

**2022-2023 Senior Devops в VK**



@dobry\_kot



github.com/fraima



1. Этапы сборки кластера
2. Terraform как единый инструмент сборки кластера
3. Что такое Cluster-API
4. Отличия Cluster-API и Terraform
5. Cluster-API ресурсы
6. Примеры валидации конфигурации кластера на примере Gatekeeper
7. Выводы

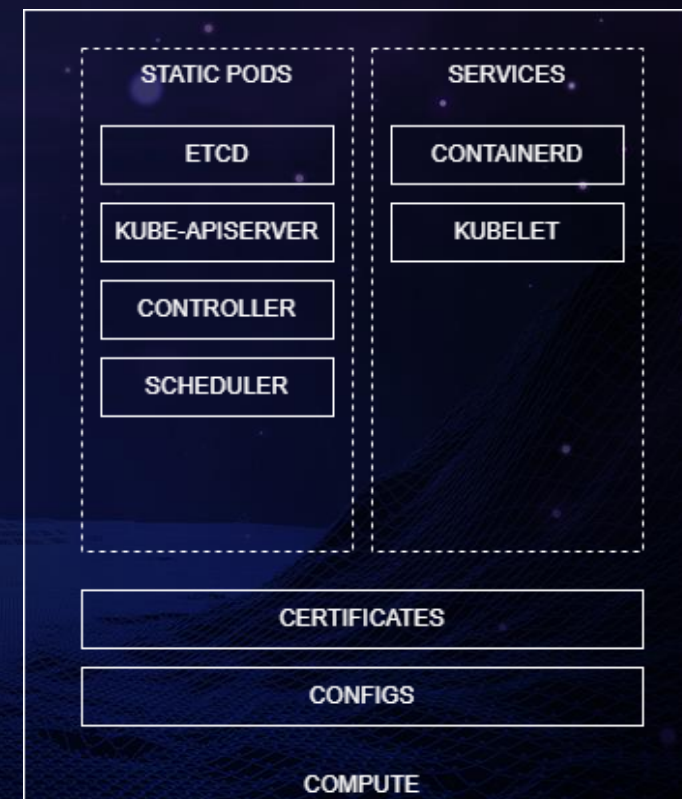
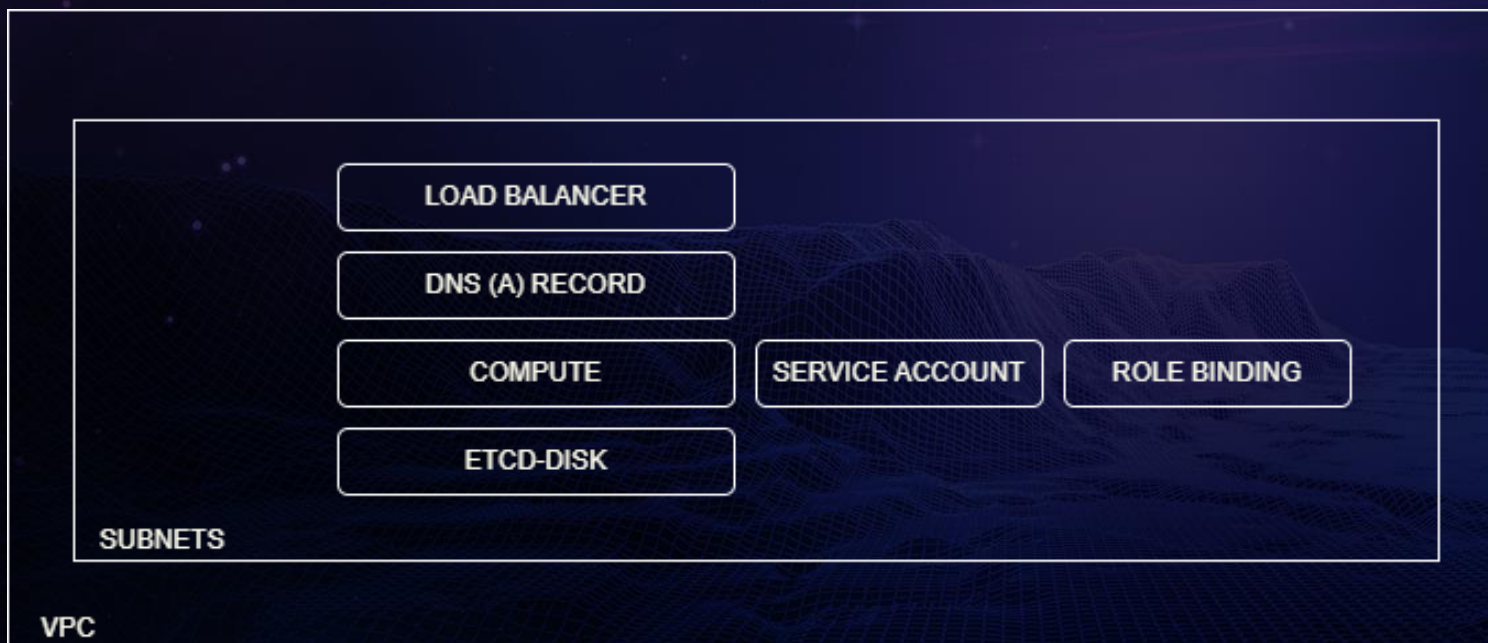
# Этапы сборки кластера



# Из каких этапов состоит сборка кластера?

## Этапы:

1. Подготовка шаблонов (конфигурационных файлов, сервисов и сертификатов)
2. Заказ инфраструктуры (VM, LB, Disk-etcd, DNS, SA, RoleBindings, VPC, SUBNET)
3. Доставка исполняемых компонент на конечные узлы (kubelet, runc, containerd, etc)
4. Сформировать из шаблонов файлы и доставить на конечные узлы



# Terraform как единый инструмент сборки кластера



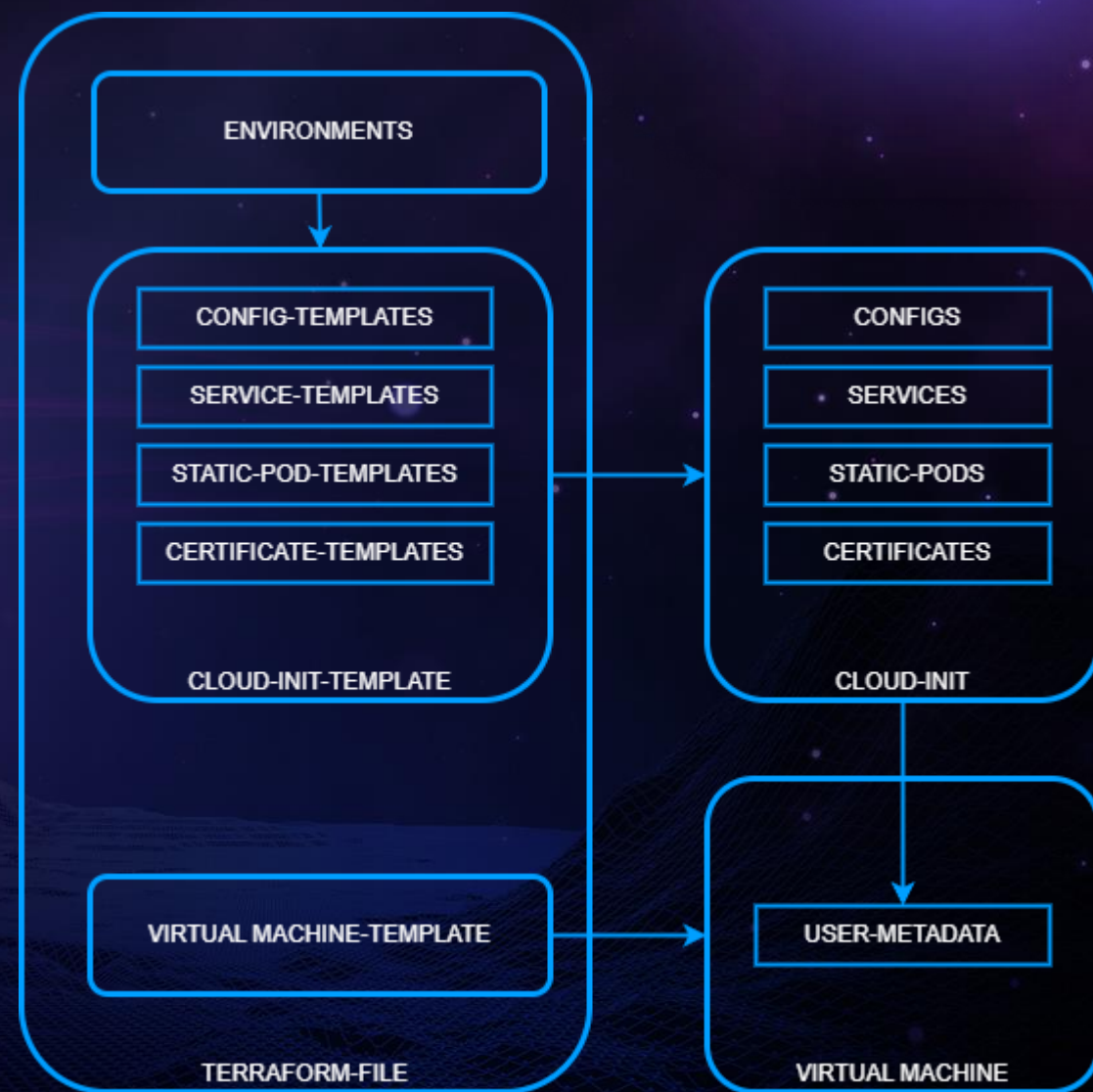
# Terraform как единый инструмент сборки кластера

## Плюсы:

1. Инструмент зарекомендовал себя в облачных средах
2. Множество готовых провайдеров и модулей под окружения
3. Большое комьюнити

## Реализация:

1. Описываем переменные окружения
2. Формируем шаблон cloud-init содержащий шаблоны конфигурационных файлов и сертификатов кластера
3. Формируем шаблон ресурса VM и передаем в user-metada шаблон cloud-init
4. Формируем шаблон ресурсов остальной инфраструктуры





# Terraform как единый инструмент сборки кластера

## Минусы:

1. Нестрогие правила написания структур ресурсов провайдера
2. Поддержка обратной совместимости конфигурации между облаками
3. Не все облака имеют одинаковый набор ресурсов
4. Неоднородность конфигурации увеличивает сложность разработки и сопровождения
5. Сложно проверять конфигурацию на соответствие требованиям

# Что такое Cluster-API?



# Cluster-API – это инструмент для создания K8S кластеров

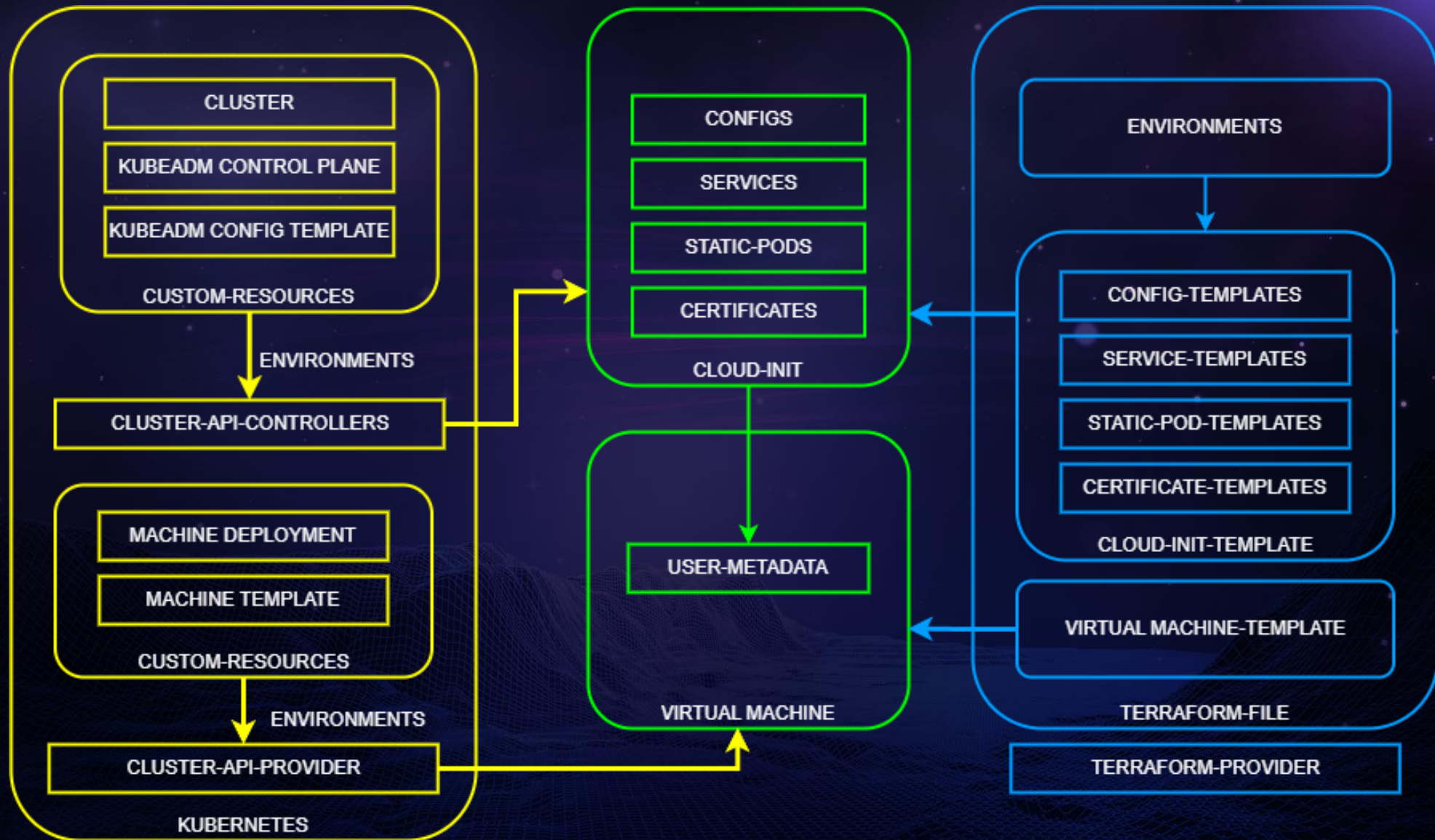
## Из чего состоит Cluster-API?

1. Kubernetes cluster типа minikube, kind, etc (основа)
2. Cert-manager (выписывание сертификатов)
3. Cluster-API controllers (основная логика для создания кластера)
4. Cluster-API provider (интерфейс для общения с облаком)
5. Cluster-API CRD (переменные окружения будущего кластера)



# Отличия Cluster-API и Terraform

# Чем отличается Cluster-API от Terraform?





## Чем отличается Cluster-API от Terraform?

1. Terraform ограничен в логических сценариях, Cluster-API ограничен воображением
2. Reconciliation loop
3. Строгий подход к написанию провайдеров
4. Повышенный уровень доверия к инструменту со стороны сообщества

# Cluster-API ресурсы



# Cluster-API ресурсы

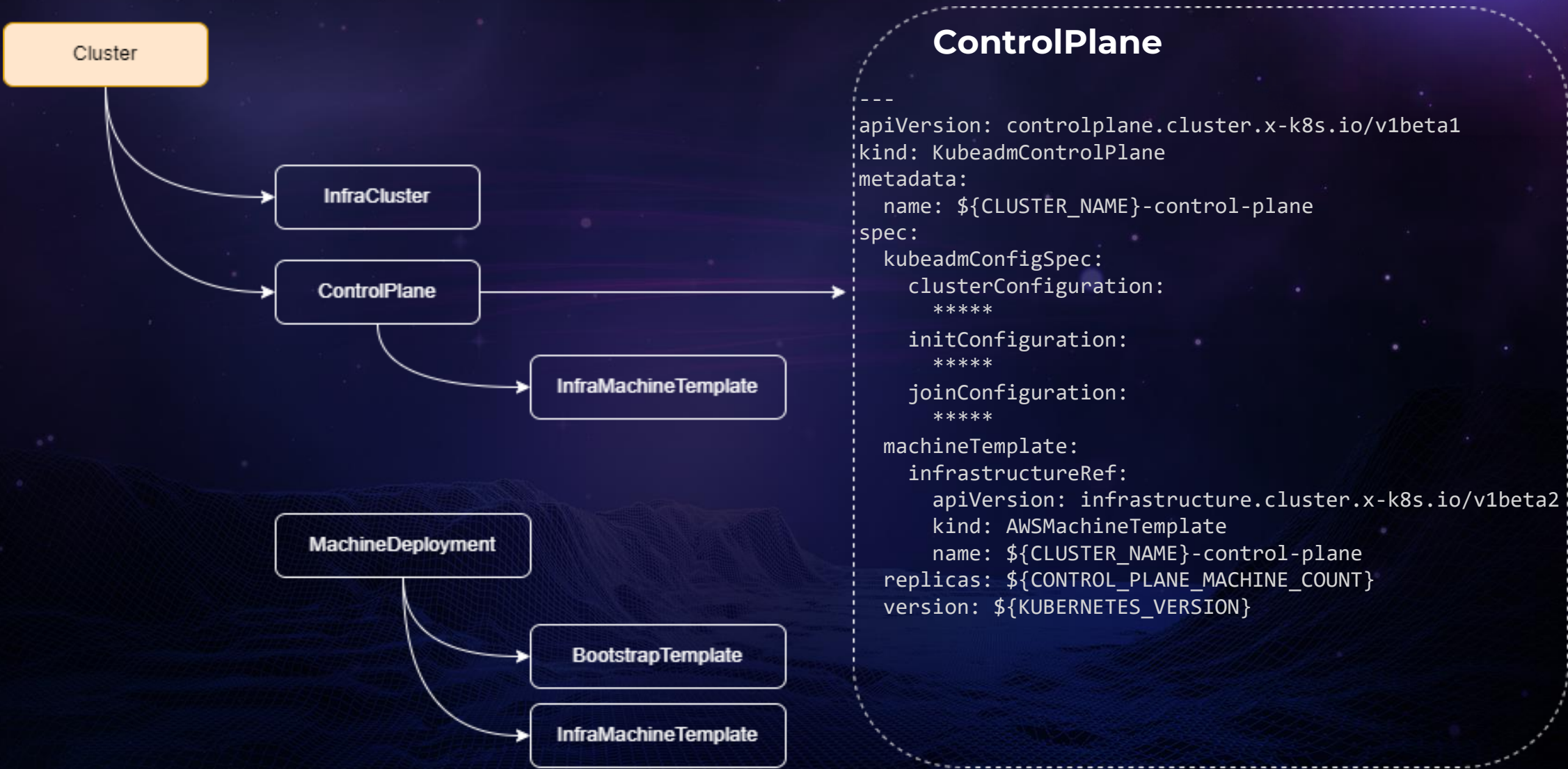
## Конфигурация:



### AWSMachineTemplate

```
---
apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
kind: AWSMachineTemplate
metadata:
  name: ${CLUSTER_NAME}-md-0
spec:
  template:
    spec:
      iamInstanceProfile: nodes.cluster-api-provider
      instanceType: ${AWS_NODE_MACHINE_TYPE}
      sshKeyName: ${AWS_SSH_KEY_NAME}
---
apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
kind: AWSMachineTemplate
metadata:
  name: ${CLUSTER_NAME}-control-plane
spec:
  template:
    spec:
      iamInstanceProfile: control-plane.cluster-api-provider
      instanceType: ${AWS_CONTROL_PLANE_MACHINE_TYPE}
      sshKeyName: ${AWS_SSH_KEY_NAME}
```

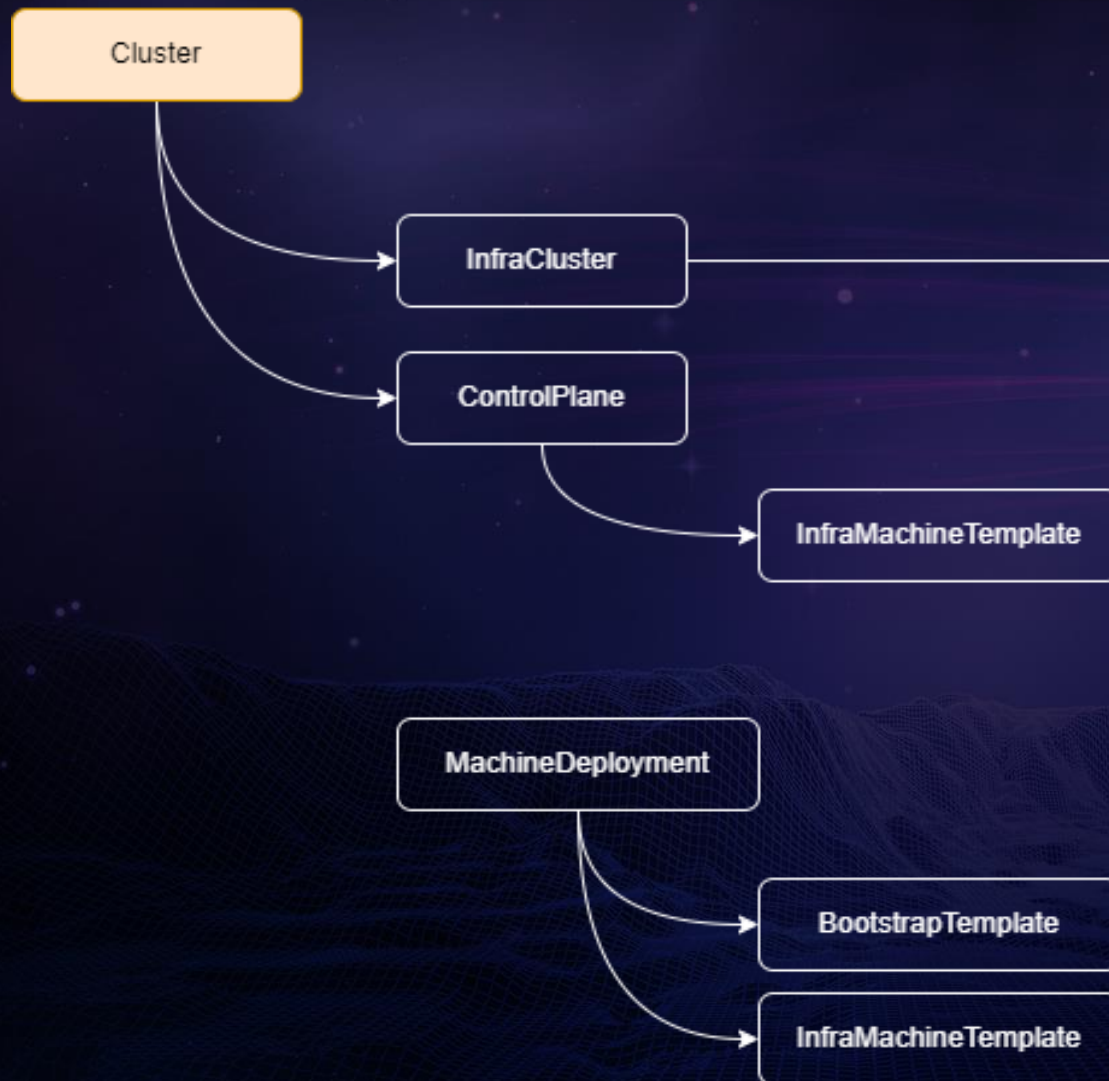
## Конфигурация:





# Cluster-API ресурсы

## Конфигурация:



### InfraCluster

```
apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
kind: AWSCluster
metadata:
  name: ${CLUSTER_NAME}
spec:
  network:
    vpc:
      availabilityZoneUsageLimit: 1
  region: ${AWS_REGION}
  sshKeyName: ${AWS_SSH_KEY_NAME}
```

# Cluster-API ресурсы

## Конфигурация:

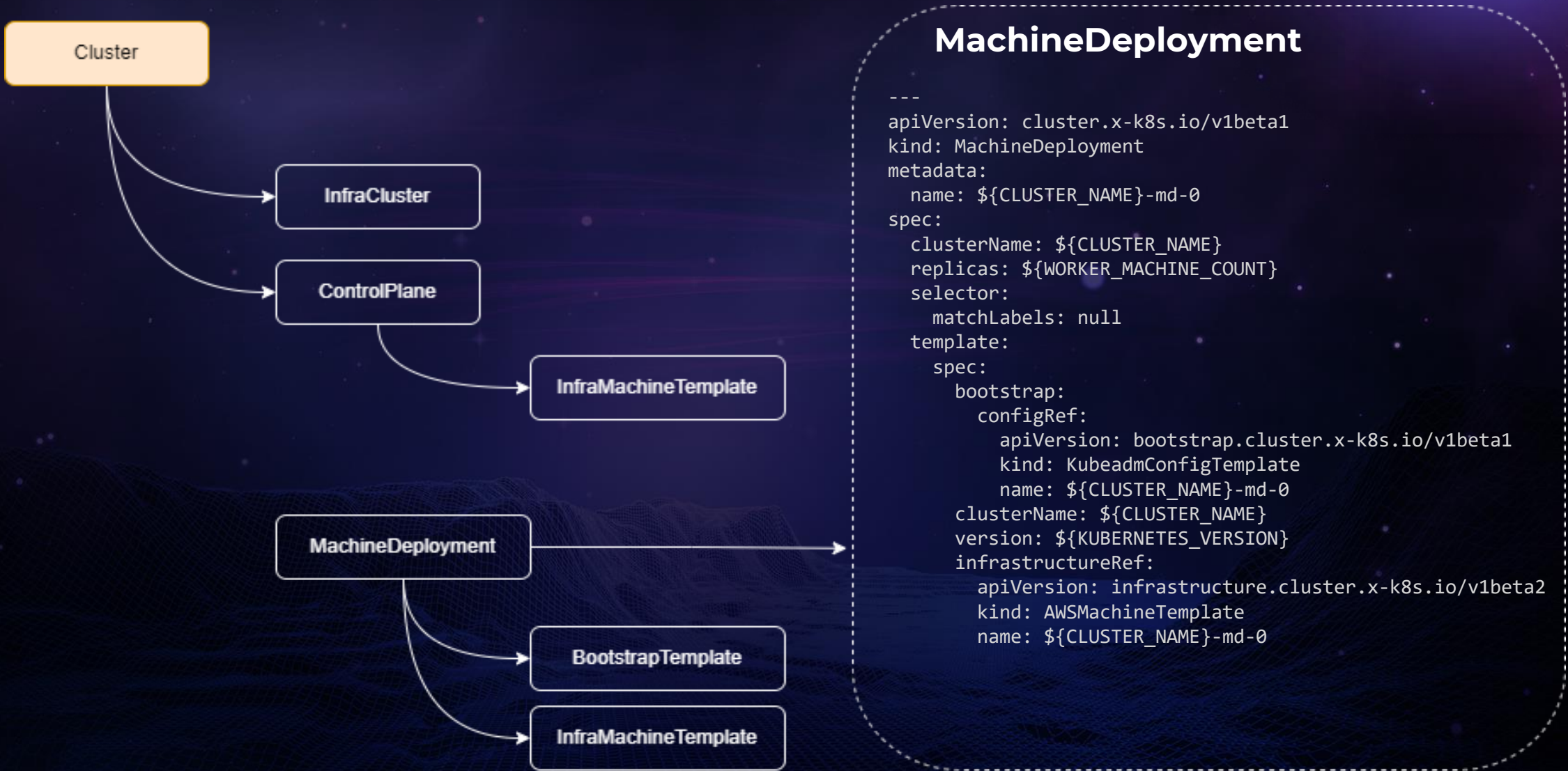


### Cluster

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  labels:
    ccm: external
    cni: ${CLUSTER_NAME}-crs-0
    csi: external
    name: ${CLUSTER_NAME}
spec:
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1beta1
    kind: KubeadmControlPlane
    name: ${CLUSTER_NAME}-control-plane
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
    kind: AWSCluster
    name: ${CLUSTER_NAME}
```



## Конфигурация:



# Cluster-api

## Конфигурация:

БЕКОН

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  labels:
    ccm: external
    cni: ${CLUSTER_NAME}-crs-0
    csi: external
  name: ${CLUSTER_NAME}
spec:
  clusterNetwork:
    pods:
      cidrBlocks:
        - 192.168.0.0/16
  controlPlaneRef:
    apiVersion: controlplane.cluster.x-k8s.io/v1beta1
    kind: KubeadmControlPlane
    name: ${CLUSTER_NAME}-control-plane
  infrastructureRef:
    apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
    kind: AWSCluster
    name: ${CLUSTER_NAME}
```

```
apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
kind: AWSCluster
metadata:
  name: ${CLUSTER_NAME}
spec:
  network:
    vpc:
      availabilityZoneUsageLimit: 1
      region: ${AWS_REGION}
      sshKeyName: ${AWS_SSH_KEY_NAME}
```

```
apiVersion: controlplane.cluster.x-k8s.io/v1beta1
kind: KubeadmControlPlane
metadata:
  name: ${CLUSTER_NAME}-control-plane
spec:
  kubeadmConfigSpec:
    clusterConfiguration:
      apiServer:
        extraArgs:
          cloud-provider: external
      controllerManager:
        extraArgs:
          cloud-provider: external
    initConfiguration:
      nodeRegistration:
        kubeletExtraArgs:
          cloud-provider: external
        name: '{{ ds.meta_data.local_hostname }}'
    joinConfiguration:
      nodeRegistration:
        kubeletExtraArgs:
          cloud-provider: external
        name: '{{ ds.meta_data.local_hostname }}'
  machineTemplate:
    infrastructureRef:
      apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
      kind: AWSMachineTemplate
      name: ${CLUSTER_NAME}-control-plane
      replicas: ${CONTROL_PLANE_MACHINE_COUNT}
      version: ${KUBERNETES_VERSION}
```

```
apiVersion: bootstrap.cluster.x-k8s.io/v1beta1
kind: KubeadmConfigTemplate
metadata:
  name: ${CLUSTER_NAME}-md-0
spec:
  template:
    spec:
      joinConfiguration:
        nodeRegistration:
          kubeletExtraArgs:
            cloud-provider: external
          name: '{{ ds.meta_data.local_hostname }}'
```

```
apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
kind: AWSMachineTemplate
metadata:
  name: ${CLUSTER_NAME}-control-plane
spec:
  template:
    spec:
      iamInstanceProfile: control-plane.cluster-api-provider-aws.sigs.k8s.io
      instanceType: ${AWS_CONTROL_PLANE_MACHINE_TYPE}
      sshKeyName: ${AWS_SSH_KEY_NAME}
```

```
apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
kind: AWSMachineTemplate
metadata:
  name: ${CLUSTER_NAME}-md-0
spec:
  template:
    spec:
      iamInstanceProfile: nodes.cluster-api-provider-aws.sigs.k8s.io
      instanceType: ${AWS_NODE_MACHINE_TYPE}
      sshKeyName: ${AWS_SSH_KEY_NAME}
```

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: MachineDeployment
metadata:
  name: ${CLUSTER_NAME}-md-0
spec:
  clusterName: ${CLUSTER_NAME}
  replicas: ${WORKER_MACHINE_COUNT}
  selector:
    matchLabels: null
  template:
    spec:
      bootstrap:
        configRef:
          apiVersion: bootstrap.cluster.x-k8s.io/v1beta1
          kind: KubeadmConfigTemplate
          name: ${CLUSTER_NAME}-md-0
          clusterName: ${CLUSTER_NAME}
        infrastructureRef:
          apiVersion: infrastructure.cluster.x-k8s.io/v1beta2
          kind: AWSMachineTemplate
          name: ${CLUSTER_NAME}-md-0
          version: ${KUBERNETES_VERSION}
```



## ИТОГ:

```
root@knode-master:~# kubectl get kubeadmcontrolplane -n aws-cluster
```

NAME	READY	INITIALIZED	REPLICAS	READY REPLICAS	UPDATED REPLICAS	UNAVAILABLE REPLICAS
capi-aws-control-plane	true	true	3	3	3	

```
root@knode-master:~# kubectl get machines -n aws-cluster
```

NAME	PROVIDERID	PHASE
capi-aws-control-plane-hwn7z	aws:///us-east-1c/i-0ec41f2979aba3f93	Running
capi-aws-control-plane-krdtk	aws:///us-east-1a/i-024bc67be29d2e0ff	Running
capi-aws-control-plane-vjhcm	aws:///us-east-1b/i-0480904cd62dc02f4	Running
capi-aws-md-0-69954c56d6-hbz77	aws:///us-east-1a/i-0894ba793dd2a41d0	Running
capi-aws-md-0-69954c56d6-q49vc	aws:///us-east-1a/i-05d8e5ea68987dfa9	Running
capi-aws-md-0-69954c56d6-vthgn	aws:///us-east-1a/i-0c940b78e9ee5d753	Running

```
root@knode-master:~# kubectl get clusters --all-namespaces
```

NAME	PHASE
capi-aws	Provisioned

```
root@knode-master:~# kubectl get machinedeployments -n aws-cluster
```

NAME	PHASE	REPLICAS	AVAILABLE	READY
capi-aws-md-0	Running	3	3	3

# Примеры валидации конфигурации кластера



# CIS Benchmark - kubelet

**C-0184 - Ensure that the Kubelet only makes use of Strong Cryptographic Ciphers**

**C-0183 - Verify that the RotateKubeletServerCertificate argument is set to true**

**C-0182 - Ensure that the --rotate-certificates argument is not set to false**

**C-0180 - Ensure that the --event-qps argument**

```
apiVersion: templates.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: kubeadmcontrolplane-validation
spec:
  crd:
    spec:
      names:
        kind: K8sManifest
  targets:
    - target: admission.k8s.gatekeeper.sh
      rego: |
        package kubeadmcontrolplane

        violation[{"msg": msg}] {
          input.kind.kind == "KubeadmControlPlane"
          not input.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.kubeletExtraArgs["event-qps"] == "5"
          msg = "Invalid value for event-qps. Allowed value is '5'."
        }

        violation[{"msg": msg}] {
          input.kind.kind == "KubeadmControlPlane"
          not input.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.kubeletExtraArgs["tls-cipher-suites"] == "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,*"
          msg = "Invalid value for tls-cipher-suites. Allowed value is 'TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,*'."
        }

        violation[{"msg": msg}] {
          input.kind.kind == "KubeadmControlPlane"
          not input.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.kubeletExtraArgs["rotate-certificates"] == "true"
          msg = "Invalid value for rotate-certificates. Allowed value is 'true'."
        }

        violation[{"msg": msg}] {
          input.kind.kind == "KubeadmControlPlane"
          not input.spec.kubeadmConfigSpec.initConfiguration.nodeRegistration.kubeletExtraArgs["feature-gates"] == "RotateKubeletServerCertificate=true"
          msg = "Invalid value for feature-gates. Allowed value is 'RotateKubeletServerCertificate=true'."
        }
```

```
---
apiVersion: controlplane.cluster.x-k8s.io/v1beta1
kind: KubeadmControlPlane
metadata:
  name: ${CLUSTER_NAME}-control-plane
spec:
  kubeadmConfigSpec:
    initConfiguration:
      nodeRegistration:
        kubeletExtraArgs:
          # C-0180
          event-qps: 5
          # C-0184
          tls-cipher-suites: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
          # C-0182
          rotate-certificates: true
          # C-0183
          feature-gates: RotateKubeletServerCertificate=true
```



## CIS Benchmark – kube-apiserver

C-0130 - Ensure that the API Server --audit-log-path argument is set

C-0117 - Ensure that the API Server --kubelet-certificate-authority argument

C-0120 - Ensure that the API Server --authorization-mode argument includes RBAC

C-0124 - Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used

```
---
apiVersion: constraints.gatekeeper.sh/v1beta1
kind: ConstraintTemplate
metadata:
  name: kubeadmcontrolplane-requirements
spec:
  crd:
    spec:
      names:
        kind: KubeadmControlPlane
  targets:
    - target: admission.k8s.gatekeeper.sh
      rego: |
        package kubeadmcontrolplane
        *****
        violation[{"msg": msg}] {
          input.review.kind.kind == "KubeadmControlPlane"
          not contains(input.request.object.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraArgs["authorization-mode"], "RBAC")
          not contains(input.request.object.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraArgs["authorization-mode"], "Node")
          let msg = "C-0120: Ensure that the API Server --authorization-mode argument includes RBAC"
        }

        violation[{"msg": msg}] {
          input.review.kind.kind == "KubeadmControlPlane"
          not contains(input.request.object.spec.kubeadmConfigSpec.clusterConfiguration.apiServer.extraArgs["enable-admission-plugins"], "SecurityContextDeny")
          let msg = "C-0124: Ensure that the admission control plugin SecurityContextDeny is set if PodSecurityPolicy is not used"
        }
}
```

```
---
apiVersion: controlplane.cluster.x-k8s.io/v1beta1
kind: KubeadmControlPlane
metadata:
  name: ${CLUSTER_NAME}-control-plane
spec:
  kubeadmConfigSpec:
    clusterConfiguration:
      apiServer:
        extraArgs:
          # C-0130
          audit-log-path: /var/log/apiserver/audit.log
          # C-0117
          kubelet-certificate-authority: /path/to/ca.crt
          # C-0120
          authorization-mode: "Node,RBAC"
          # C-0124
          enable-admission-plugins: SecurityContextDeny
```



# Выводы

## Cluster-API + OPA:

1. Поможет избавиться от тысяч строк кода на Ansible, Terraform, Puppet, Saltstack
2. Унифицирует ваше окружение и кластера
3. Добавит валидацию для конфигурации кластера
4. Предоставит достаточно ручек для дополнительной кастомизации

## Минусы:

1. Передача приватных ключей в cloud-init
2. Не для всех облаков есть провайдеры



7 июня 2023 📍 Москва, МЦК ЗИЛ  
Первая в России конференция  
по БЕзопасности КОНтейнеров и контейнерных сред



Contacts:

Email: [dlputilin@dobry-kot.ru](mailto:dlputilin@dobry-kot.ru)

Tg: [@dobry\\_kot](https://t.me/@dobry_kot)

git: [github.com/fraima](https://github.com/fraima)