Kubernetes is written in **Go** language

1. What is Kubernetes?

Kubernetes (K8s) არის ღია კოდის სისტემა კონტეინერირებული აპლიკაციების განლაგების, მასშტაბირებისა და მართვის ავტომატიზაციისთვის. ის აჯგუფებს კონტეინერებს, რომლებიც ქმნიან აპლიკაციას ლოგიკურ ერთეულებად მარტივი მართვისა და აღმოჩენისთვის. K8s გთავაზობთ უამრავ მახასიათებელს, რაც მას პოპულარულ არჩევანს ხდის კონტეინერირებული აპლიკაციების მართვისთვის, მათ შორის: კონტეინერის ორკესტრირება, K8 ავტომატიზირებს კონტეინერების განლაგებას, მასშტაბირებას და მართვას. K8s აადვილებს სერვისების აღმოჩენას და წვდომას, რომლებიც გაშვებულია კლასტერზე. K8-ებს შეუძლიათ ავტომატურად დაამონტაჟონ შენახვის მოცულობა კონტეინერებში. ეს შეიძლება გამოყენებულ იქნას აპლიკაციების მუდმივი შენახვის უზრუნველსაყოფად.

2. What is Kubectl?

kubectl არის ბრძანების ხაზის ინსტრუმენტი Kubernetes კლასტერებთან ურთიერთობისთვის. ის საშუალებას გაძლევთ განათავსოთ, მართოთ და გააფართოვოთ კონტეინერირებული აპლიკაციები Kubernetes-ზე. kubectl ასევე გამოიყენება Kubernetes-ის რესურსებთან ურთიერთობისთვის, როგორიცაა pods, deployments და სერვისები. kubectl მუშაობს HTTP მოთხოვნების გაგზავნით Kubernetes API სერვერზე. API სერვერი პასუხისმგებელია Kubernetes კლასტერის მდგომარეობისა და მისი რესურსების მართვაზე. როდესაც თქვენ აწარმოებთ kubectl ბრძანებას, ის აგზავნის HTTP მოთხოვნას API სერვერზე, რომელიც შემდეგ ასრულებს მოთხოვნილ ოპერაციას.

3. What is Control Plane?

ის პასუხისმგებელია კლასტერის მართვაზე და მისი შეუფერხებლად მუშაობის უზრუნველსაყოფად. Control plane შედგება კომპონენტების ნაკრებისგან, რომლებიც ერთად მუშაობენ ისეთი ამოცანების შესასრულებლად, როგორიცაა: დაგეგმა: control plane წყვეტს, რომელ worker nodeზე გაუშვათ პოდები. Control plane ანაწილებს ტრაფიკს პოდებს შორის ისე, რომ ეფექტური და საიმედო იყოს.

4. What is Kube-apiserver?

Kube-apiserver არის Kubernetes-ის მნიშვნელოვანი კომპონენტი და აუცილებელია ნებისმიერი Kubernetes კლასტერის მუშაობისთვის. Kubernetes API არის RESTful API, რომელიც საშუალებას აძლევს კლიენტებს შექმნან, წაიკითხონ, განაახლონ და წაშალონ Kubernetes ობიექტები, როგორიცაა pods, სერვისები და განლაგება. Kube-apiserver პასუხისმგებელია Kubernetes ობიექტების მონაცემების დამოწმებასა და კონფიგურაციაზე. ის ასევე უზრუნველყოფს ავთენტიფიკაციას და ავტორიზაციას Kubernetes API-ის მოთხოვნებისთვის. Kube-apiserver არის პირველადი ინტერფეისი Kubernetes-ის საკონტროლო სიბრტყესა და Kubernetes კლასტერის სხვა კომპონენტებს შორის.

5. What is etcd?

etcd არის ღია კოდის განაწილებული გასაღების მნიშვნელობის სთორი საერთო კონფიგურაციისთვის, სერვისების აღმოჩენისთვის და განაწილებული სისტემების ან მანქანების კლასტერების კოორდინაციისთვის. ეს არის მრავალი განაწილებული სისტემის ძირითადი კომპონენტი, მათ შორის Kubernetes. etcd შექმნილია საიმედო, მასშტაბირებადი და მარტივი გამოსაყენებლად. ის აგებულია რაფტის კონსენსუსის ალგორითმის თავზე, რაც უზრუნველყოფს მონაცემების თანმიმდევრულობას კლასტერის ყველა node-ში etcd ასევე ძალიან მასშტაბირებადია და მისი გამოყენება შესაძლებელია ათასობით node-ის კლასტერების სამართავად.

6. What is kube-scheduler?

kube-scheduler არის Kubernetes-ის კომპონენტი, რომელიც პასუხისმგებელია ნოუდებისთვის პოდსების მინიჭებაზე. ის ამას აკეთებს სხვადასხვა ფაქტორების გათვალისწინებით, მათ შორის პოდს-ის მოთხოვნების, ნოუდებზე არსებული რესურსების და კონფიგურირებული დაგეგმვის პოლიტიკის გათვალისწინებით. kube-scheduler არის Kubernetes-ის კრიტიკული კომპონენტი და ის მნიშვნელოვან როლს ასრულებს იმის უზრუნველსაყოფად, რომ pods განლაგებულია იმ node-ში სადაც მათ შეუძლიათ იმუშაონ ყველაზე ეფექტურად. kube-scheduler არის ძლიერი ინსტრუმენტი Kubernetes კლასტერებისა და აპლიკაციების სამართავად. ეს აუცილებელია ნებისმიერი ორგანიზაციისთვის, რომელიც ამუშავებს Kubernetes-ს წარმოებაში.

8. What is Kubelet?

Kubelet არის აგენტი, რომელიც მუშაობს კუბერნეტის კლასტერში თითოეულ node-ზე. ის პასუხისმგებელია კონტეინერების მართვაზე node-ზე, მათ შორის გაშვებასა და გაჩერებაზე, წარუმატებელი კონტეინერების გადატვირთვაზე და მათ მონიტორინგზე. Kubelet ასევე დაუკავშირდება Kubernetes API სერვერს, რათა ის ინფორმირებული იყოს ნოუდისა და მისი კონტეინერების მდგომარეობის შესახებ.

9. What is Kube-proxy?

Kube-proxy არის ქსელის პროქსი, რომელიც მუშაობს კუბერნეტის კლასტერში თითოეულ კვანძზე. ის პასუხისმგებელია სერვისებსა და პოდებს შორის ქსელური კავშირის შენარჩუნებაზე. Kube-proxy არის Kubernetes-ის კრიტიკული კომპონენტი და ის მნიშვნელოვან როლს ასრულებს ქსელის ტრაფიკის სწორად გადაყვანის უზრუნველსაყოფად. Kube-proxy ინარჩუნებს ქსელურ კავშირს სერვისებსა და პოდებს შორის, იმის უზრუნველსაყოფად, რომ სწორი ქსელის წესები მოქმედებს თითოეულ კვანძზე. ეს საშუალებას აძლევს პოდებს დაუკავშირდნენ სერვისებს, მაშინაც კი, თუ პოდები ან სერვისები გადატანილია სხვადასხვა კვანძში.

10. What is Nodes?

node Kubernetes-ის აუცილებელი კომპონენტია და ისინი მნიშვნელოვან როლს ასრულებენ აპლიკაციების შეუფერხებლად მუშაობის უზრუნველსაყოფად. Nodes are the workhorses of Kubernetes. ისინი პასუხისმგებელნი არიან პოდების გაშვებაზე, რომლებიც Kubernetes აპლიკაციების ძირითადი სამშენებლო ბლოკია. Pods არის ერთი ან მეტი კონტეინერის ჯგუფები, რომლებიც განლაგებულია ერთად node-ზე. node ასევე პასუხისმგებელნი არიან პოდებისთვის შენახვისა და ქსელის უზრუნველყოფაზე. kubelet აგენტი უზრუნველყოფს, რომ პოდებს აქვთ წვდომა იმ რესურსებზე, რომლებიც მათ გასაშვებად სჭირდებათ.

11. What is Pods?

ოდი არის ერთი ან მეტი კონტეინერის ჯგუფი, საერთო საცავებით (მოცულობებით) და ქსელის რესურსებით და კონტეინერების გაშვების სპეციფიკაციით. Pods შექმნილია Kubernetes კლასტერის ერთ კვანძზე გასაშვებად და ისინი ძალიან პორტატულია. თქვენ შეგიძლიათ მარტივად გადაიტანოთ nodeები, ერთი nodeდან მეორეზე მათი შესრულების შეფერხების გარეშე. Pods არის ძლიერი ინსტრუმენტი Kubernetes კლასტერებზე კონტეინერირებული აპლიკაციების მართვისთვის. ისინი უზრუნველყოფენ უამრავ სარგებელს, მათ შორის: იზოლირება, პორტაბელურობა, მასშტაბურობა,

12. Troubleshooting with kubectl

პრობლემების მოგვარება kubectl-ით

ყველაზე გავრცელებული ოპერაციები შეიძლება შესრულდეს შემდეგი kubectl ბრძანებებით:

● kubectl get - რესურსების სია

● kubectl describe - რესურსის შესახებ დეტალური ინფორმაციის ჩვენება

● kubectl logs - ამობეჭდეთ ჟურნალები კონტეინერიდან პოდში

● kubectl exec - შეასრულეთ ბრძანება კონტეინერზე პოდში

თქვენ შეგიძლიათ გამოიყენოთ ეს ბრძანებები, რათა ნახოთ, როდის განთავსდა აპლიკაციები, როგორია მათი ამჟამინდელი სტატუსი,

სად მუშაობენ და როგორია მათი კონფიგურაციები

13. What is a Cluster?

კლასტერი არის ერთი სისტემა, რომელიც შედგება სერვერებისა და სხვა რესურსებისგან, Kubernetes კლასტერი არის set of nodes, რომლებიც აწარმოებენ კონტეინერიზებულ აპლიკაციებს. კლასტერები გამოიყენება სხვადასხვა მიზნებისთვის, მათ შორის მაღალი ხარისხის გამოთვლებისთვის, დიდი მონაცემების დამუშავებისა და ვებ ჰოსტინგისთვის. Kubernetes კლასტერები შედგება ერთი ძირითადი master და რამდენიმე workerისგან. ეს nodeები შეიძლება იყოს ფიზიკური კომპიუტერები ან ვირტუალური მანქანები, დამოკიდებულია კლასტერზე. კლასტერები გამოიყენება სხვადასხვა მიზნებისთვის, მათ შორის მაღალი ხარისხის გამოთვლებისთვის, დიდი მონაცემების დამუშავებისა და ვებ ჰოსტინგისთვის.

14. What is a container in Kubernetes?

Kubernetes-ის კონტეინერი არის მსუბუქი, იზოლირებული გარემო, რომელსაც შეუძლია აპლიკაციის გაშვება. კონტეინერები ვირტუალური მანქანების მსგავსია, მაგრამ ისინი უფრო ეფექტური და პორტატულია. კონტეინერები იზიარებენ ძირითად ოპერაციულ სისტემას, მაგრამ მათ აქვთ საკუთარი იზოლირებული მომხმარებლის სივრცე. ეს ნიშნავს, რომ თითოეულ კონტეინერს შეუძლია საკუთარი აპლიკაციის გაშვება სისტემის სხვა კონტეინერებზე გავლენის გარეშე. კონტეინერები შეფუთულია ყველაფრით, რაც მათ გასაშვებად სჭირდება, კოდის, ბიბლიოთეკების და გაშვების გარემოს ჩათვლით. ეს აადვილებს კონტეინერების განლაგებას და გაშვებას Kubernetes-ის ნებისმიერ კლასტერზე.

15. kubectl expose

kubectl expose ბრძანება გამოიყენება Kubernetes-ის სერვისის შესაქმნელად deployment, replica set, და pod. თქვენ ასევე შეგიძლიათ გამოიყენოთ `kubectl expose` ბრძანება, რომ შექმნათ სერვისი ასლის ნაკრებისთვის ან პოდისთვის. სერვისის შესაქმნელად `kubectl expose` ბრძანების გამოყენებით, უნდა მიუთითოთ სერვისის სახელი, სერვისის ტიპი და გამოსაჩენი პორტი. თქვენ ასევე შეგიძლიათ მიუთითოთ დამატებითი პარამეტრები, როგორიცაა სამიზნე პორტი და სელექტორი

16. deployments

Kubernetes-ის განლაგება არის კუბერნეტის კლასტერში აპლიკაციების განლაგების მართვის საშუალება. განლაგება არის Kubernetes-ის რესურსი, რომელიც განსაზღვრავს პოდების ნაკრების სასურველ მდგომარეობას. Deployment უზრუნველყოფს დეკლარაციულ განახლებებს Pods-ისა და ReplicaSets-ისთვის.

თქვენ აღწერთ სასურველ მდგომარეობას Deployment-ში და Deployment Controller ცვლის რეალურ მდგომარეობას სასურველ მდგომარეობამდე კონტროლირებადი სიჩქარით. თქვენ შეგიძლიათ განსაზღვროთ Deployments ახალი ReplicaSets-ის შესაქმნელად, ან არსებული განლაგების წასაშლელად და მათი ყველა რესურსის ახალი განლაგებით.

16. What is Ingress?

Ingress არის Kubernetes API ობიექტი, რომელიც საშუალებას გაძლევთ განსაზღვროთ ტრაფიკის მარშრუტის წესები თქვენს კლასტერში არსებულ სერვისებზე. ეს არის დეკლარაციული გზა თქვენს აპლიკაციებზე გარე წვდომის კონფიგურაციისთვის. Ingress შეიძლება გამოყენებულ იქნას: უზრუნველყოთ დატვირთვის დაბალანსება: Ingress-ს შეუძლია ტრაფიკის განაწილება სერვისის მრავალ ნაწილზე. ეს დაგეხმარებათ გააუმჯობესოთ თქვენი აპლიკაციების შესრულება და საიმედოობა.

SSL/TLS ტრაფიკის შეწყვეტა: Ingress-ს შეუძლია შეწყვიტოს SSL/TLS ტრაფიკი და უზრუნველყოს HTTPS წვდომა თქვენს აპლიკაციებზე. ეს დაგეხმარებათ თქვენი აპლიკაციების უსაფრთხოების გაუმჯობესებაში. Ingress არის ძლიერი ინსტრუმენტი, რომელიც დაგეხმარებათ მართოთ გარე წვდომა თქვენს Kubernetes აპლიკაციებზე. ეს არის გარე წვდომის კონფიგურაციის დეკლარაციული გზა, რაც აადვილებს მის მართვას და შენარჩუნებას.

17. What is the Kubernetes controller manager?

Kubernetes controller manager, რომელიც მუშაობს Kubernetes-ის control plane-ზე. ის პასუხისმგებელია კონტროლერების ნაკრების გაშვებაზე, რომლებიც მონიტორინგს უწევენ კლასტერულ მდგომარეობას და იღებენ ზომებს იმის უზრუნველსაყოფად, რომ ის შეესაბამება სასურველ მდგომარეობას. კონტროლერის მენეჯერი Kubernetes-ის კრიტიკული კომპონენტია და აუცილებელია Kubernetes-ის ნებისმიერი კლასტერის მუშაობისთვის. კონტროლერის მენეჯერი არის ძლიერი ინსტრუმენტი, რომელიც დაგეხმარებათ მართოთ თქვენი Kubernetes კლასტერი უფრო ეფექტურად. ეს არის კლასტერული მდგომარეობის მართვის დეკლარაციული გზა, რაც აადვილებს მის მართვას და შენარჩუნებას. კონტროლერის მენეჯერი არის მოქნილი და შეიძლება მორგებული იყოს ორგანიზაციის სპეციფიკური საჭიროებების დასაკმაყოფილებლად.