[Amazon EKS] (CheatSheet)

1. EKS Cluster Management

- **Create an EKS Cluster**: `eksctl create cluster --name my-cluster`
- **List EKS Clusters**: `aws eks list-clusters`
- **Describe an EKS Cluster**: `aws eks describe-cluster --name my-cluster`
- **Delete an EKS Cluster**: `eksctl delete cluster --name my-cluster`
- **Update EKS Cluster Version**: `eksctl upgrade cluster --name my-cluster --version <new-version>`

2. Node Group Management

- **Create Node Group**: `eksctl create nodegroup --cluster my-cluster --name my-nodegroup`
- **List Node Groups in a Cluster**: `eksctl get nodegroup --cluster my-cluster`
- **Describe Node Group**: `aws eks describe-nodegroup --cluster-name my-cluster --nodegroup-name my-nodegroup`
- **Delete Node Group**: `eksctl delete nodegroup --cluster my-cluster --name my-nodegroup`
- **Scale Node Group**: `eksctl scale nodegroup --cluster my-cluster --name my-nodegroup --nodes <desired-count>`

3. Configuring kubectl for EKS

- **Update Kubeconfig for EKS Cluster**: `aws eks update-kubeconfig --name my-cluster`
- **Switch Kubectl Context to EKS Cluster**: `kubectl config use-context <context-name>`
- **Check Current Context**: `kubectl config current-context`

4. IAM Integration with EKS

- **Associate IAM Role to EKS Cluster**: `eksctl create iamidentitymapping --cluster my-cluster --arn <role-arn> --group system:masters --username admin`

By: Waleed Mousa

- **List IAM Identity Mappings**: `eksctl get iamidentitymapping --cluster my-cluster`
- **Remove IAM Identity Mapping**: `eksctl delete iamidentitymapping --cluster my-cluster --arn <role-arn>`

5. Network and VPC Configuration

- **Specify Subnets When Creating a Cluster**: `eksctl create cluster --name my-cluster --vpc-public-subnets subnet-12345,subnet-67890`
- **Create EKS Cluster with Private Networking**: `eksctl create cluster --name my-cluster --vpc-private-subnets subnet-12345,subnet-67890`
- **Update Cluster to Use Private Endpoint**: `eksctl utils update-cluster-endpoints --cluster my-cluster --private-access=true`

6. Security and Access Control

- **Enable Public and Private Access for EKS API Server**: `eksctl utils update-cluster-endpoints --cluster my-cluster --private-access=true --public-access=true`
- **Restrict Public Access to Certain IPs**: `eksctl utils set-public-access-cidrs --cluster my-cluster --approve --cidrs=<ip-range>`
- **Enable Logging for EKS Cluster**: `eksctl utils update-cluster-logging --enable-types=all --cluster my-cluster --approve`

7. Deploying Applications

- **Deploy Application**: `kubectl apply -f deployment.yaml`
- **List Deployments in EKS Cluster**: `kubectl get deployments`
- **Delete Deployment**: `kubectl delete deployment my-deployment`
- **Scale Deployment**: `kubectl scale deployment my-deployment --replicas=3`
- **Update Deployment Image**: `kubectl set image deployment/my-deployment my-container=my-image`

8. Service and Ingress

- **Expose Deployment as a Service**: `kubectl expose deployment my-deployment --port=80 --type=LoadBalancer`

- **Get Services**: `kubectl get services`
- **Create Ingress Resource**: `kubectl apply -f ingress.yaml`
- **List Ingress Resources**: `kubectl get ingress`

9. Working with Pods

- **List Pods in EKS Cluster**: `kubectl get pods`
- **Describe Pod**: `kubectl describe pod my-pod`
- **Delete Pod**: `kubectl delete pod my-pod`
- **Exec into a Running Pod**: `kubectl exec -it my-pod -- /bin/bash`

10. ConfigMaps and Secrets

- **Create a ConfigMap**: `kubectl create configmap my-config --from-literal=key1=value1`
- **Get ConfigMaps**: `kubectl get configmaps`
- **Create a Secret**: `kubectl create secret generic my-secret --from-literal=key1=value1`
- **Get Secrets**: `kubectl get secrets`

11. Monitoring and Logging

- **Deploy Kubernetes Metrics Server**: `kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml`
- **View Cluster Metrics**: `kubectl top nodes`
- **Install and Configure CloudWatch Agent**: `kubectl apply -f cloudwatch-agent.yaml`

12. Storage and Volumes

- **Create a Persistent Volume Claim (PVC)**: `kubectl apply -f pvc.yaml`
- **Get Persistent Volume Claims**: `kubectl get pvc`
- **Create a Storage Class**: `kubectl apply -f storage-class.yaml`
- **Get Storage Classes**: `kubectl get storageclass`

13. Autoscaling

- **Autoscale Deployment**: `kubectl autoscale deployment my-deployment --min=2 --max=5 --cpu-percent=80`
- **View Autoscaler Status**: `kubectl get hpa`
- **Edit Autoscaler Settings**: `kubectl edit hpa my-autoscaler`

14. Updating EKS Clusters and Node Groups

- **Upgrade EKS Cluster**: `eksctl upgrade cluster --name=my-cluster --approve`
- **Upgrade Node Group**: `eksctl upgrade nodegroup --name=my-nodegroup --cluster=my-cluster`

15. Managing EKS with eksctl

- **List Clusters with eksctl**: `eksctl get cluster`
- **Create Cluster with Specific Kubernetes Version**: `eksctl create cluster --name=my-cluster --version=1.18`
- **Delete Cluster with eksctl**: `eksctl delete cluster --name=my-cluster`

16. Advanced kubectl Commands for EKS

- **Rolling Restart of Deployments**: `kubectl rollout restart deployment/my-deployment`
- **Get Resource Usage by Pod**: `kubectl top pod`
- **Edit Resource on the Fly**: `kubectl edit svc/my-service`

17. Handling Jobs and CronJobs in EKS

- **Create a Job**: `kubectl apply -f job.yaml`
- **Get Jobs**: `kubectl get jobs`
- **Create a CronJob**: `kubectl apply -f cronjob.yaml`
- **Get CronJobs**: `kubectl get cronjobs`

18. EKS Cluster Add-ons

- **Install CoreDNS**: `eksctl create addon --name coredns --cluster my-cluster --force`
- **Install Kubernetes Dashboard**: `kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml`
- **Access Kubernetes Dashboard**: `kubectl proxy`

19. Backup and Restore

- **Install Velero for Backup/Restore**: `velero install --provider aws --bucket my-bucket --backup-location-config region=us-west-2 --snapshot-location-config region=us-west-2`
- **Create Backup with Velero**: `velero backup create my-backup`
- **Restore from Backup with Velero**: `velero restore create --from-backup my-backup`

20. EKS and IAM

- **Associate IAM OIDC Provider**: `eksctl utils associate-iam-oidc-provider --cluster=my-cluster --approve`
- **Create IAM Role for Service Account**: `eksctl create iamserviceaccount --name my-service-account --namespace default --cluster my-cluster --attach-policy-arn arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess --approve --override-existing-serviceaccounts`

21. Advanced Networking

- **Set up Calico for Network Policies**: `kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml`
- **Create a Network Policy**: `kubectl apply -f network-policy.yaml`
- **List Network Policies**: `kubectl get networkpolicy`

22. EKS Cluster Upgrades

- **Upgrade Cluster Control Plane**: `eksctl upgrade cluster --name my-cluster`
- **Upgrade Managed Node Groups**: `eksctl upgrade nodegroup --name=my-nodegroup --cluster=my-cluster`

23. Blue/Green and Canary Deployments

- **Deploy Blue/Green Using AWS App Mesh**: `kubectl apply -f appmesh-blue-green.yaml`
- **Implement Canary Deployments with Flagger**: `kubectl apply -f flagger-canary.yaml`

24. Managing Certificates and TLS

- **Install cert-manager for TLS**: `kubectl apply --validate=false -f https://github.com/jetstack/cert-manager/releases/download/v1.0.0/cert-manager.yaml`
- **Create a Certificate Resource**: `kubectl apply -f certificate.yaml`

25. EKS Cost Optimization

- **Use Spot Instances in Node Groups**: `eksctl create nodegroup --cluster my-cluster --name my-spot-nodegroup --spot`

26. Security and Compliance

- **Install kube-bench for CIS Benchmarks**: `kubectl apply -f kube-bench.yaml`
- **Run Security Scans with kube-bench**: `kubectl run --rm -i -t kube-bench --image=aquasec/kube-bench:latest --restart=Never -- kube-bench`
- **Install AWS Inspector for Cluster Security Assessment**: `aws inspector start-assessment-run --assessment-template-arn <template-arn>`

27. Integration with AWS Services

- **Integrate with AWS CloudWatch for Logging**: `kubectl apply -f aws-cloudwatch-logging.yaml`
- **Use AWS Load Balancer Controller for Ingress**: `kubectl apply -f aws-load-balancer-controller.yaml`

28. Pod Security Policies

- **Apply Pod Security Policy**: `kubectl apply -f pod-security-policy.yaml`
- **List Pod Security Policies**: `kubectl get psp`

29. Service Mesh

- **Install AWS App Mesh**: `kubectl apply -f aws-app-mesh.yaml`
- **Create a Mesh in AWS App Mesh**: `aws appmesh create-mesh --mesh-name my-mesh`

30. Disaster Recovery

- **Set Up Cross-Region EKS Cluster Replication**: `aws eks create-cluster --name secondary-cluster --region <secondary-region>`
- **Implement Disaster Recovery Plan with AWS Backup**: `aws backup start-backup-job --backup-vault-name my-vault --resource-arn <cluster-arn>`

31. EKS with Fargate

- **Create a Fargate Profile**: `eksctl create fargateprofile --cluster my-cluster --name my-fargate-profile --namespace my-namespace`
- **List Fargate Profiles**: `eksctl get fargateprofile --cluster my-cluster`

32. Managing EKS with AWS CLI

- **Create Cluster with AWS CLI**: `aws eks create-cluster --name my-cluster --role-arn <role-arn> --resources-vpc-config subnetIds=<subnet-ids>,securityGroupIds=<security-group-ids>`
- **Update EKS Cluster Config**: `aws eks update-cluster-config --name my-cluster --logging '{"clusterLogging":[{"types":["api","audit"],"enabled":true}]}'`

33. Advanced Resource Management

- **Taint EKS Nodes**: `kubectl taint nodes <node-name> key=value:effect`
- **Patch Kubernetes Resources**: `kubectl patch deployment my-deployment -p '{"spec":{"template":{"metadata":{"labels":{"date":"date +'%s'"}}}}}'`

34. High Availability and Fault Tolerance

- **Create Multi-AZ Node Groups**: `eksctl create nodegroup --cluster my-cluster --name multi-az-nodegroup --nodes-min=2 --nodes-max=4 --node-type=t3.medium --nodes=3 --region=us-west-2 --zones=us-west-2a,us-west-2b,us-west-2c`
- **Set Pod Disruption Budgets**: `kubectl apply -f pod-disruption-budget.yaml`

35. Performance Tuning

- **Optimize Cluster Autoscaler**: `kubectl apply -f cluster-autoscaler-autodiscover.yaml`
- **Implement Vertical Pod Autoscaler**: `kubectl apply -f vertical-pod-autoscaler.yaml`