YouTube Video URL

https://youtu.be/p0tYcr4Z7eY

1. How would you deploy an application to AWS?

1. How would you deploy an application to AWS?

When a company wants to use AWS to deploy their workloads, they need to set up a landing zone (https://docs.aws.amazon.com/prescriptive-guidance/latest/migration-aws-environment/understanding-landing-zones.html). Here, you will setup VPCs including NAT gateways, Internet gateways, Security Groups etc. if you want to deploy an application on a VM, that is, an EC2 instance, then you will need to provision it. Depending on what kind of application, either it will have internet access or will not. So, you will have to choose the right VPC and within the VPC, right subnet (public or private), Security group. Also, if application needs autoscaling then you may use an AutoScaling group, attach with a Load Balancer and point the LB DNS to a DNS record in your hosted zone (example Route53). Once the instance is up, you will need to install application dependencies (here I am talking about a monolithic application) and deploy the application package (like a jar file). If you need a DB, then you may utilize AWS RDS service and so on.

2. What measures have you taken to secure your EKS clusters?

2. What measures have you taken to secure your EKS clusters?

- If you are using managed node groups then AWS is going to administer the worker node OS, kubelet, and AMIs.
- Don’t allow SSH access to nodes, instead use AWS SSM or Session Manager.
- Do not use Service Accounts for authentication as the credentials associated are static in nature. Instead, use RBAC or Role Based Access Control which follows the principle of least privileges to AWS resources.
- The cluster endpoint should be made private.
- Run your applications as a non-root user.
- Use CloudTrail for audit logs.
- Use AWS KMS for rotating the customer-managed keys.
- Have a robust security vulnerabilities management system within your EKS cluster.

3. What is the toughest challenge that you have faced with EKS clusters?

3. What is the toughest challenge that you have faced with EKS clusters?

Upgrading a cluster looks straight forward on paper but a lot of the things have to be taken care before you go for it. The first thing is to go through all the release notes in detail to understand the changes in the newer version. We need to be careful with any deprecated APIs that might already be used with some of our applications. So, its important to have a discussion with the developer team and understand the impact of the change. Always critical to have complete testing done in the lower environments before we go for production upgrade.

4. Have you ever worked on remediating security vulnerabilities?

4. Have you ever worked on remediating security vulnerabilities remediation?

Yes, we are using a tool called Tenable to run scans in our infrastructure. The scans are both internal and external. This is managed by the security team. So, they get the reports which has many details of the vulnerability like CVE ID, impact, remediation references etc. So, depending on the criticality of the vulnerability, we had SLAs to fix these.

For example, a new PHP version is available which is able to fix some vulnerabilities reported on an application in production. Then, we had to check with the application developer if we can go ahead with this change or not. If they say yes, then we will apply the change in the lower environment first, the developers will test it and share the results and then we can do the same change in production.

5. What monitoring tools have you used? Can you explain what components did you monitor?

5. What monitoring tools have you used? Can you explain what components did you monitor?

I have used Prometheus to scrape metrics data and used Grafana to visualize it. Generally we have multiple Kubernetes clusters to monitor. So, we can scrape metrics from Control-plane components like api-server, coredns, kube-scheduler and on the worker nodes, we can scrape container metrics from kubelet (cAdvisor). We can also access kube-state-metrics which means cluster-level metrics like deployments, pods etc. To get host-level metrics like cpu, memory, and network, we could use node-exporters.

For visualization, we use Grafana dashboards. For example, getting requests per minute from an application, getting the count of 500, 502, 504 errors etc.