Transcript

good afternoon and welcome everyone to the Cs 470 conference presentation in Cloud development my name is Scott I am a fourth year computer science major and as many of you are just about to receive my degree today I'm excited to present this talk about the many facets of cloud development whether you're a season Tech Enthusiast or someone just beginning in the world of technology I'll do my best to make this presentation fun informative and engaging for all the purpose of today's presentation is twofold first off I'd like to introduce myself and give a bit of background on who I am and why I'm interested in Cloud development in the first place secondly I would like to talk about the many aspects of cloud development and explain any complexities in a way that both Technical and non-technical audiences can understand so let's lock the hatch and blast off to explore the fasinating world of cloud development together in this presentation we'll talk

about development Concepts such as
serverless versus local storage
advantages of a serverless API
differences between mongodb and Dynamo
DB cloud-based development and also
Security in a cloud-based
application migrating fullstack
applications to the cloud offers
numerous benefits including scalability
and cost efficiency by understanding the
different migration models and best
practices a business can navigate the
cloud migration task with ease unlocking
the full potential of cloud computing
for their applications to explain what
is involved in taking an application to
the cloud I will explore models of
containerization first and also
orchestration first I would like to
speak about
containerization describing what a
container is how containers work the
benefits of using containers and the key
aspects of
containerization what is a container you
might ask a container simplifies the
process of packaging and deploying
applications by encapsulating each
component and its dependencies into

lightweight containers think of a
container as a portable storage basket
that holds everything your application
needs to run from the code to libraries
and all of the configurations
necessary a secondary aspect of of
containerization is
orchestration think of orchestration as
a superhero movers for your
applications just like movers help to
handle all the heavy lifting and
transportation during the move tools
such as Docker compose manage the
nitty-gritty details of deploying
scaling and taking care of your
containers the orchestration tools like
Docker compose automate the deployment
scaling and management of containers Mak
Mak sure that they're in the right place
at the right time and running without a
hitch across the entire infrastructure
so you can sit back relax and let these
superhero movers take care of your
application next I'd like to talk about
the serverless cloud using a serverless
API offers advantages such as
scalability cost Effectiveness and
simplified infrastructure management
serverless platforms automatically scale

resources based on demand so you don't need to worry about provisioning or managing servers this ensures your API can handle sudden spikes in traffic without any downtime also an advantage of serverless you only pay for the resources you use typically measured in function calls and execution time this can result in significant cost savings compared to traditional server-based architectures where you pay for idle resources serverless platforms hide the complexities of the underlying infrastructure allowing developers to focus solely on writing code this reduces the operational overhead associated with managing server and network roles the scripts produced to make this happen are essentially the code for the Lambda functions these scripts contain the logic for handling API requests and generating responses they can include any necessary dependencies such as sdks for interacting with AWS services or third-party libraries for specific

functionality next I'd like to talk
about the data model for the serverless
cloud mongodb utilizes a flexible
document based data model where data is
stored in Json like
documents documents are grouped into
collections and each document
encompasses its structure and
schema mongodb supports complex queries
and
indexes Dynamo DB on the other hand
utilizes a key value and document-based
data
model data is organized to tables which
each item can have a different structure
Dynamo DB requires specifying a primary
key for each item which can be a simple
partition key or composite partition and
sort key queries in mongodb perform
operations such as find update and
delete while in dynamodb which supports
key based operations queries for
retrieving updating and deleting items
based on their primary keys or
used scripts that were written included
operations to connect DB server
and perform crud operations while Dynamo
DB included operations to authenticate
with Amazon web service and interact

with Dynamo DB tables execute queries
and handle
responses next I will describe
elasticity and the pay as you go Concept
in cloudbased development elasticity in
cloud computing refers to the ability of
a system to to dynamically shift
Computing resources based on workload
essentially it's about scaling resources
up or down automatically in response to
changes in
demand the pay for use model is a
billing model commonly employed by a
cloud service provider it requires
customers to pay only for the cloud res
resources and services they use rather
than making them pay for upfront
investments in
infrastructure this seems to be an
efficient and cost-effective matter to
host an application in a cloud-based
World preventing unauthorized access to
Cloud information in AWS involves a
strategy that includes multiple layers
of protection this can be accomplished
by using identity access management to
control access to AWS services and
resources create am users groups and
roles within the principle of lease

privilege granting only permissions
necessary for users to perform their
tasks you can also enable multiactor
authentication MFA for added security
the difference between IAM roles and
policies in AWS is that a role is a type
of I am identity that can be
authenticated and authorized to utilize
an AWS resource whereas a policy defines
the permissions of the I am
identity to secure the connection
between Lambda and API Gateway Lambda
and the database and Lambda and S3
bucket it is important to implement
various security
measures use API Gateway resour resource
policies to control access at the API
level and restrict access based on IP
address VP endpoints or Amazon resource
names also known as
Arns I'd like to thank everyone for
attending the Cs 470 talk about Cloud
development I would like to leave you
with three takeaways from our conference
number one Cloud security i' like to
highlight the importance of cloud
Security in today's digital landscape
with the increasing use of cloud-based
Services organizations need to

prioritize security measures to protect their data applications and infrastructure from cyber threats and breaches number two best practices I'd like to mention the importance of best practices and strategies for securing Cloud environments including implementing strong identity and access management controls encrypting sensitive data monitoring and auditing Cloud resources and adopting a proactive approach to threat detection number three continuous learning it's very important and we like to emphasize the need for continuous learning and adaptation in the field of cloud security as technology evolves and new threats emerge Security Professionals need to stay updated on the latest trends tools and techniques to effectively Safeguard Cloud environments and mitigate strategy risks thank you all for attending