Cloud Native Application Overview

SHAN
Cloud Architect

Cloud Native Application Introduction

01

Microservices

Cloud native applications use microservices architecture to break down complex applications into smaller, manageable services that can be developed, deployed, and scaled independently.

02

Containerization

Utilizing containers allows developers to package applications and their dependencies together, ensuring consistency across various environments and facilitating smoother deployment processes.

03

DevOps Practices

Implementing DevOps practices within cloud native applications enhances collaboration between development and operations teams, leading to faster delivery cycles and improved software quality.

Importance of Cloud Native Architecture

Scalability

Easily scale applications up or down based on user demand.

Resilience

Applications recover quickly from failures using built-in redundancy.

Speed

Accelerated development cycles with continuous integration and deployment.

Flexibility

Utilize multiple cloud services for optimal application performance.

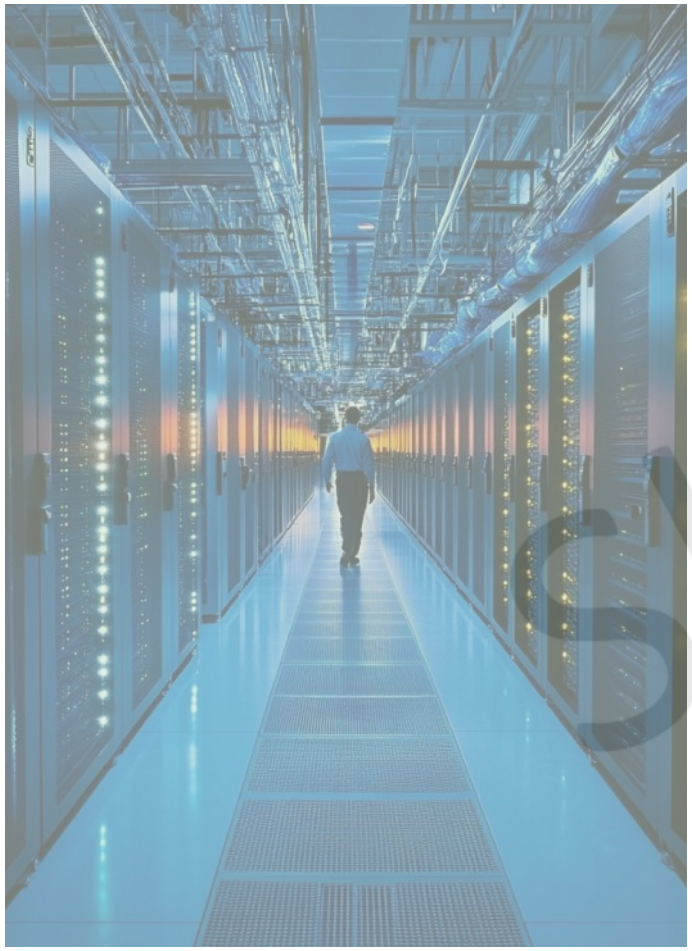
Cost Efficiency

Pay only for resources consumed rather than fixed server costs.

Innovation

Easier to adopt new technologies and improve application features.

Key Principles of Cloud Native Design



Microservices

Develop applications as independent, loosely coupled services.

Containers

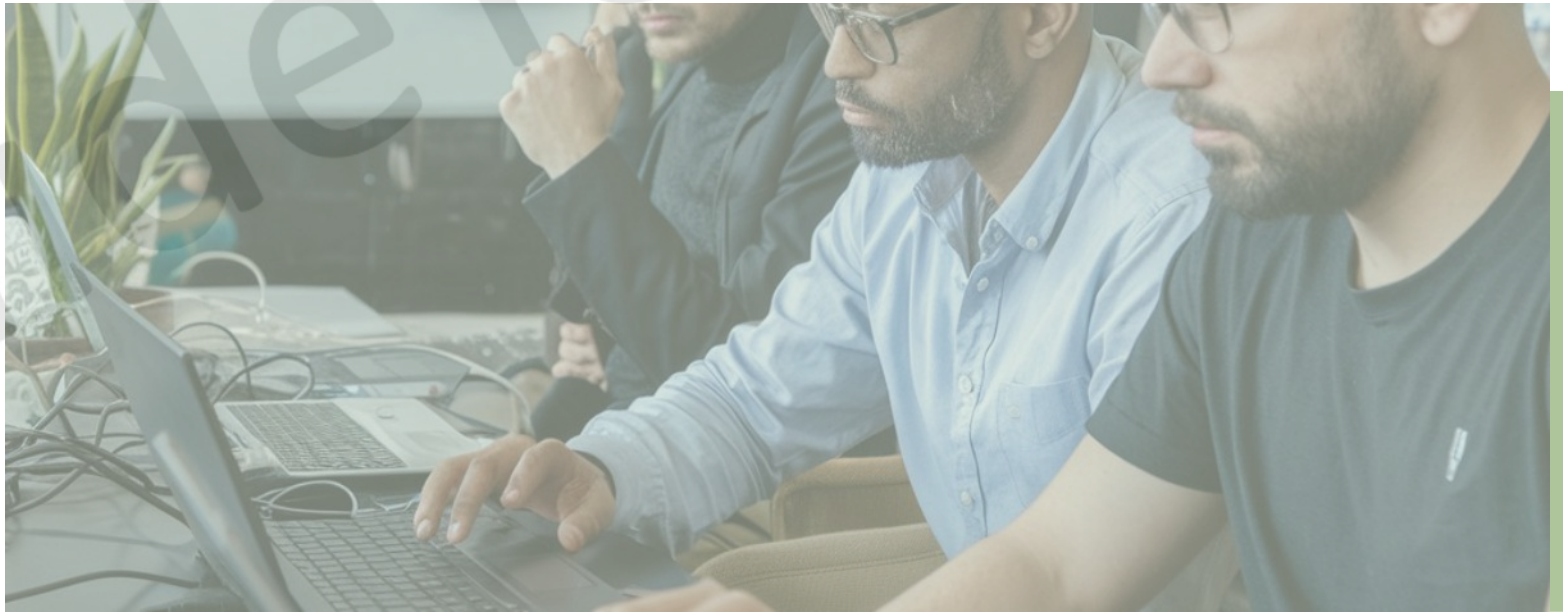
Package applications in lightweight, portable containers for consistency.

DevOps

Integrate development and operations to improve collaboration and efficiency.

Continuous Delivery

Automate deployment processes for faster and reliable releases.



Microservices and their Role

Independent

Microservices operate independently, promoting scalability and flexibility.

Deployment

Each microservice can be deployed without impacting others, ensuring continuous delivery.

Tech Diversity

Allows use of multiple technologies for different services, enhancing performance.

Resilience

Isolation of services improves system resilience against failures or outages.

Scaling

Microservices can be scaled independently based on specific service demand.

Containerization: Benefits and Practices

Utilize lightweight containers to streamline microservices deployment, enhancing scalability and portability across different environments. Implement CI/CD pipelines for automated testing and deployment to ensure rapid and consistent updates while minimizing downtime and operational risks.

Orchestration with Kubernetes Overview

01

Scalability

Kubernetes automates the scaling process of applications based on current demand, ensuring resource efficiency and optimal performance without manual intervention.

02

Service Discovery

Kubernetes provides built-in service discovery, allowing applications to automatically locate and connect to services, streamlining communication between different application components.

Cloud Provider Options and Choices

	Compute	Storage	Networking	Security	Pricing	Support
AWS	EC2	S3	VPC	IAM	$$	24/7
Azure	VMs	Blob Storage	Virtual Network	Azure Security	$$	24/7
Google Cloud	Compute Engine	Cloud Storage	VPC	IAM	$	Business hours
IBM Cloud	VMs	Cloud Object Storage	VPC	Cloud Security	$$$	Business hours

DevOps Practices for Cloud Native

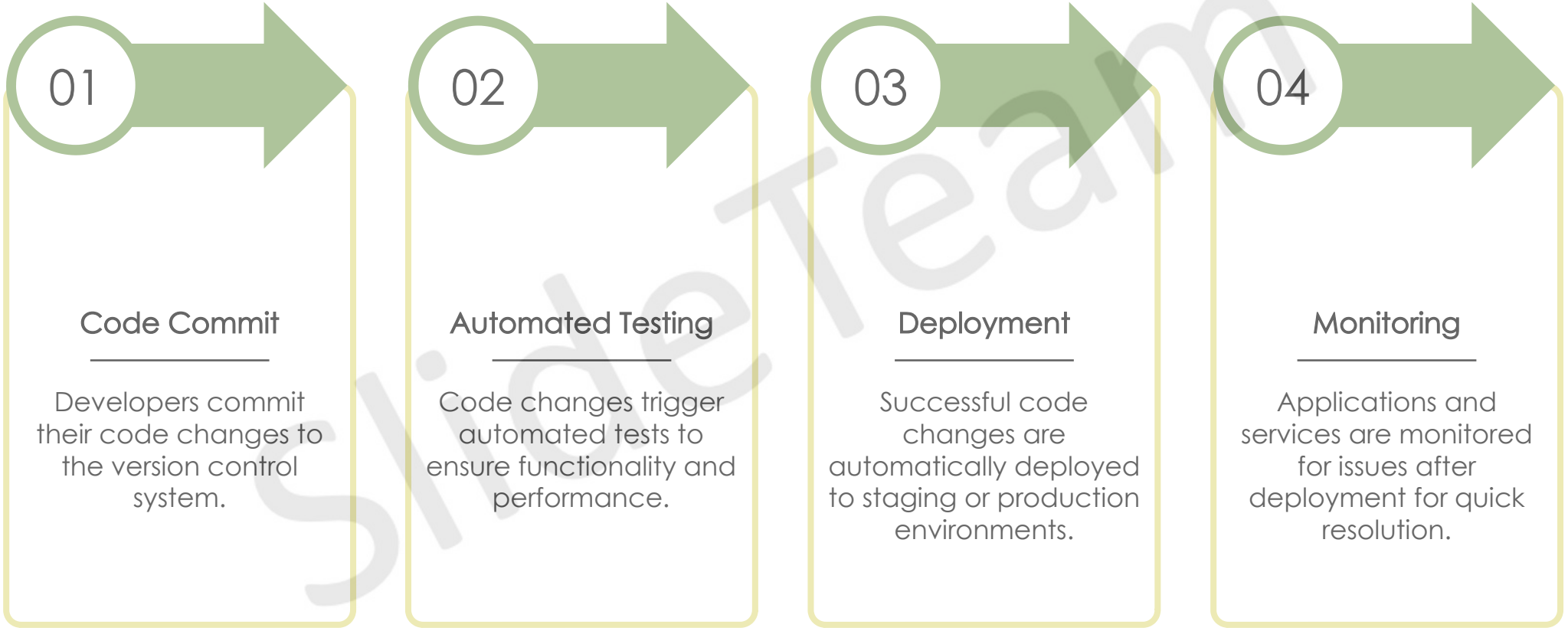
Continuous Delivery

Automate release processes to ensure that developers can deploy new features and bug fixes quickly and reliably without manual intervention in the cloud environment.

Infrastructure as Code

Manage and provision infrastructure using code to allow for versioning, consistency, and scalability, enabling teams to deploy cloud resources through scripts and configuration files.

Continuous Integration/Continuous Deployment



Challenges in Cloud Native Adoption

Pros

Scalability

01

Easily scale applications to meet changing user demands and expectations.

Flexibility

02

Allows developers to choose best tools and technologies for their applications.

Cons

Complexity

01

Managing multiple services can lead to increased operational challenges and difficulties.

Costs

02

Transitioning to cloud native can incur significant upfront and ongoing costs.

Security Considerations for Cloud Native

Access Control

Implement role-based access to limit user permissions effectively.

Data Encryption

Use encryption for data at rest and in transit to secure sensitive information.

Regular Audits

Conduct regular security audits to identify vulnerabilities and fix them promptly.

Network Segmentation

Segment networks to isolate services and minimize potential attack surfaces.

API Security

Enforce security measures to protect APIs from unauthorized access and misuse.

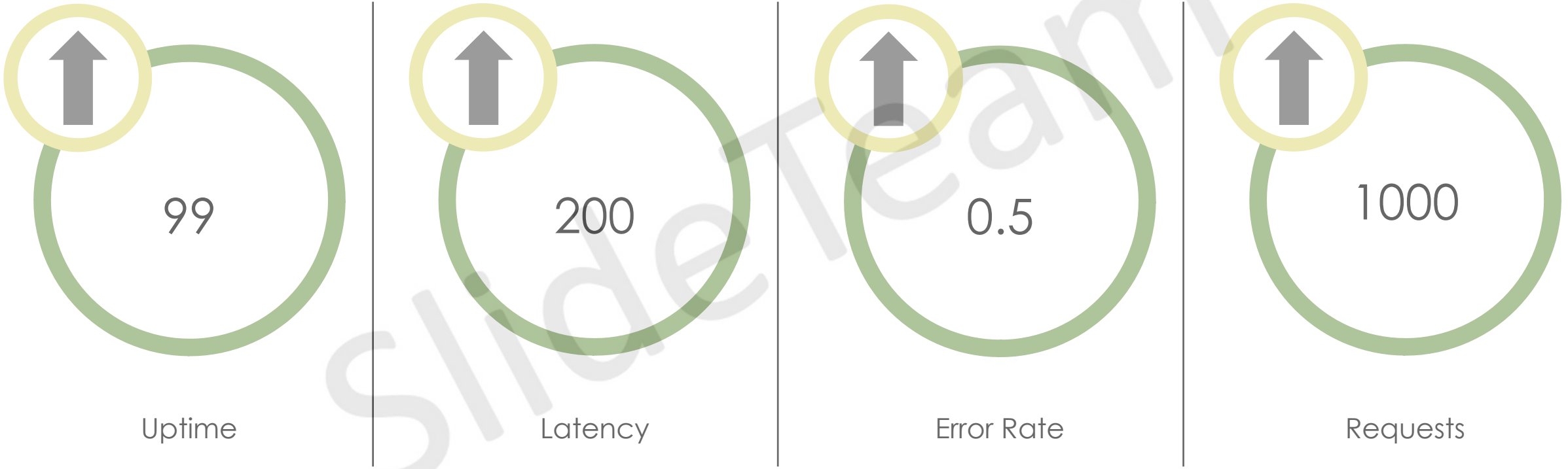
Monitoring Tools

Utilize monitoring tools to detect anomalies and respond to security incidents swiftly.

Incident Response

Establish a clear incident response plan to manage security breaches effectively.

Monitoring and Observability Techniques



This is a sample dashboard. Please edit the metrics according to your message.

Cost Management in Cloud Native Environments

Monitor

Continuously track usage to avoid unexpected costs.

Tag

Use tagging strategies to track and allocate costs by project.

Optimize

Regularly refine resource allocation based on performance needs.

Analyze

Conduct regular cost audits to identify inefficiencies and savings.

Automate

Use automation to shut down idle resources to save money.

Limit

Set budgets and alerts to manage spending proactively.

Forecast

Implement predictive analytics for budget planning and forecasting.

Review

Periodically assess cloud service providers for better pricing.

Real-World Cloud Native Case Studies

Problem Faced

Limited scalability and high latency in applications.

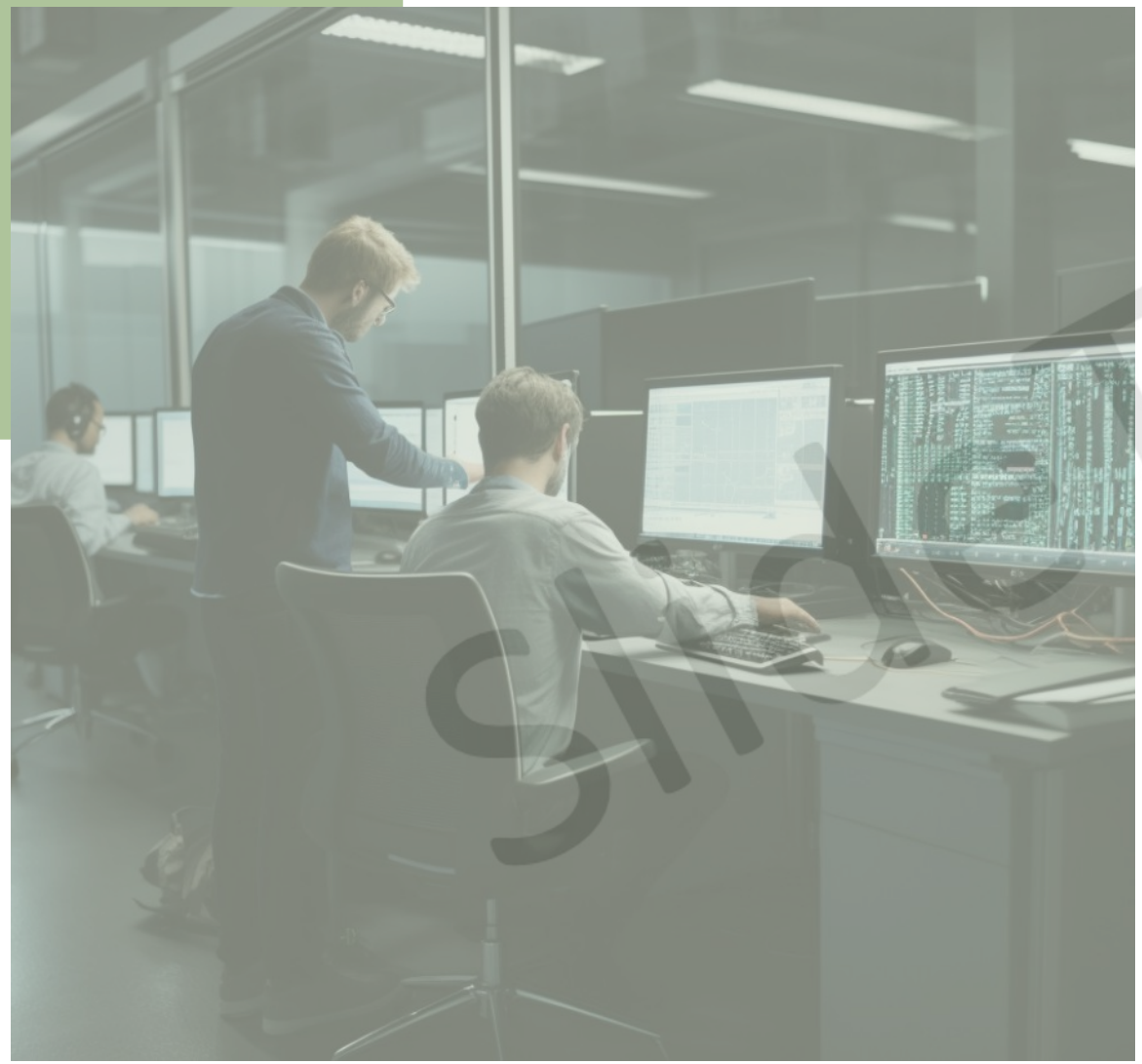
Solution Offered

Migrated to microservices architecture for efficient load.

Benefits

Improved performance and reduced deployment time.

Best Practices for Implementing Cloud Native



Use Microservices

Break applications into smaller services to improve scalability and maintainability effectively.

Automated Testing

Implement continuous integration and automated testing to ensure code quality and fast deployment.

Container Orchestration

Utilize orchestration tools for managing deployment, scaling, and operations of application containers.

Future Trends in Cloud Native Architecture

01

Microservices

Adoption of microservices architecture enhances scalability and allows for independent deployment of services.

02

Serverless Computing

Serverless models reduce infrastructure management, allowing developers to focus solely on code and functionality.

03

Kubernetes

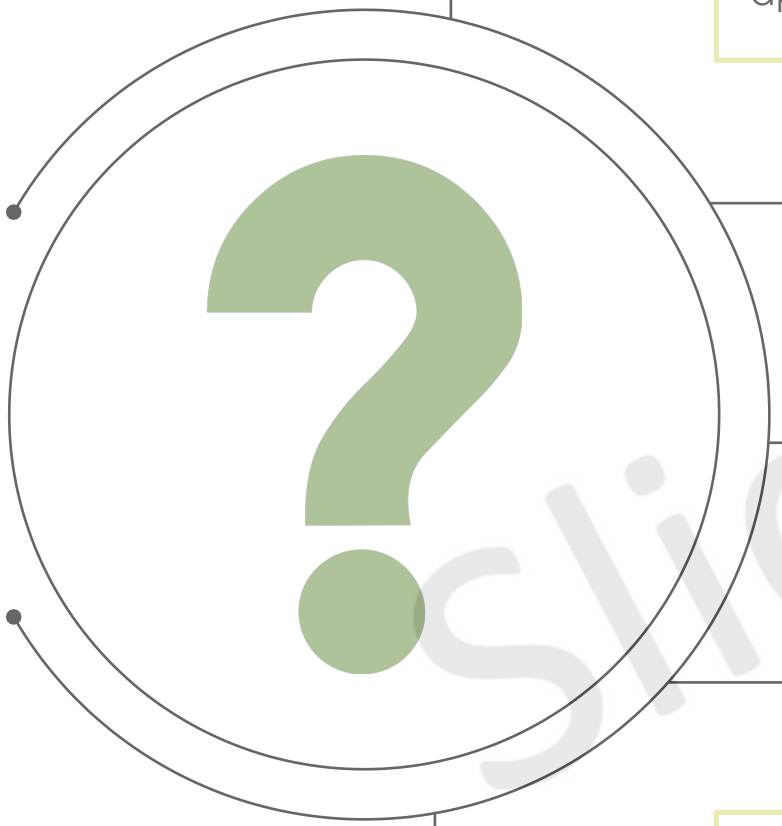
Kubernetes will become the standard for container orchestration, improving deployment and scaling strategies.

04

AI Integration

Integration of AI capabilities in cloud-native applications supports automation and enhances user experience.

Interactive Q&A Session



What challenges have you faced adopting cloud native applications?

Your Text Here

Your Text Here

Your Text Here

Your Text Here

Resources for Further Learning

Online Courses

Explore top-rated cloud native courses on platforms like Coursera.

Podcasts

Listen to podcasts discussing cloud native topics and trends.

Books & eBooks

Read essential literature on cloud native architecture and practices.

Documentation

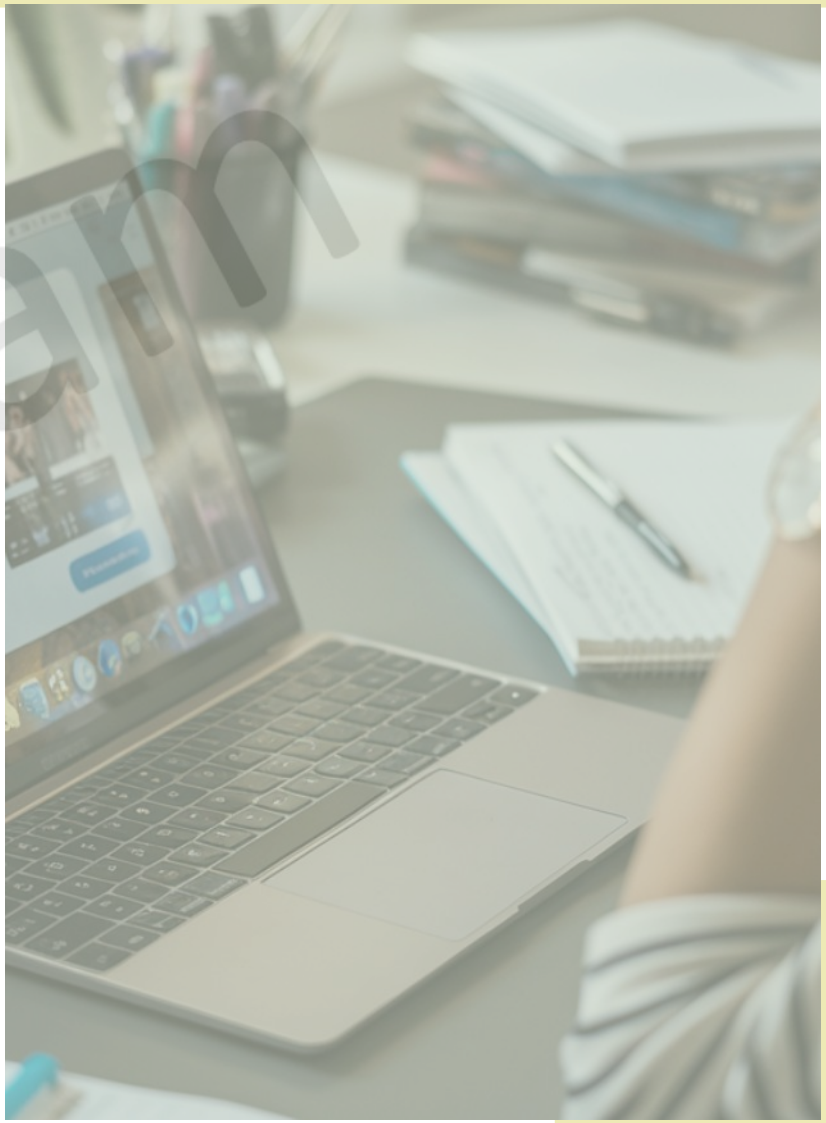
Access official documentation for tools and technologies in cloud native.

Webinars

Attend webinars hosted by industry experts to deepen your knowledge.

Community Forums

Join forums and communities to share and gain cloud native insights.



Summary and Key Takeaways

01

Scalability Benefits

Cloud native applications can scale efficiently to meet varying demands.

02

Microservices Architecture

Using microservices allows for easier updates and independent development cycles.

03

Resilience and Reliability

Cloud native design promotes high availability and fault tolerance features.

04

Continuous Delivery

Enables frequent updates and faster deployment through automation practices.

05

Resource Efficiency

Maximizes resource utilization, leading to cost savings and optimization.

06

Enhanced Collaboration

Facilitates better teamwork across development and operations teams.

Thanks for watching!

Address
123 Cloud St, Cloud City, CA

Email Address
contact@cloudnative.com

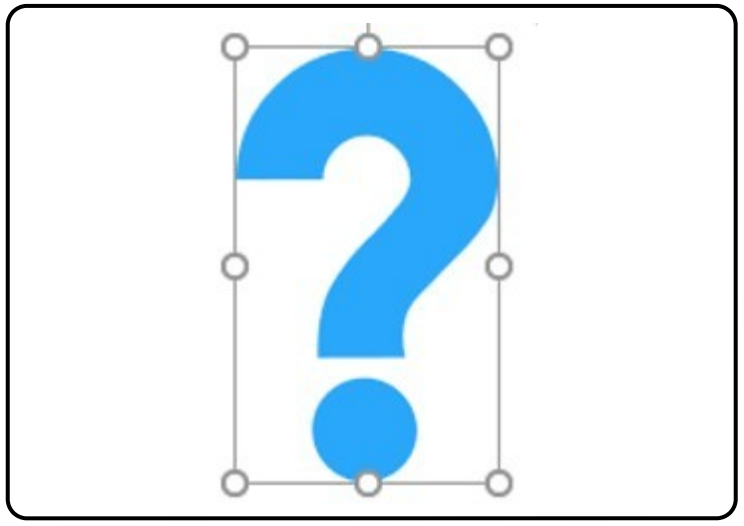
Contact Number
(123) 456-7890

Instructions to Change Color of Shapes

Some shapes in this deck need to be ungrouped to change colors

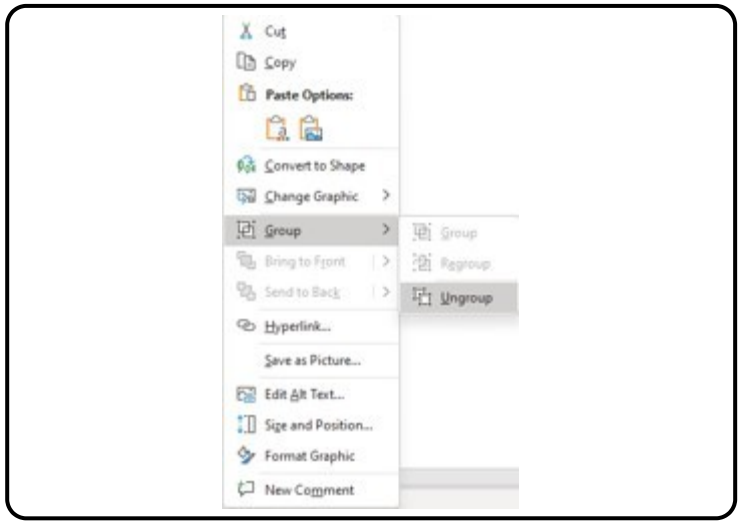
Step 1:

Select the shape, and right click on it



Step 2:

Select Group -> Ungroup.



Step 3:

Once ungrouped, you will be able to change colors using the "Format Shape" option