Knowledge Transfer Document

Application : eGov

Technology : Devops

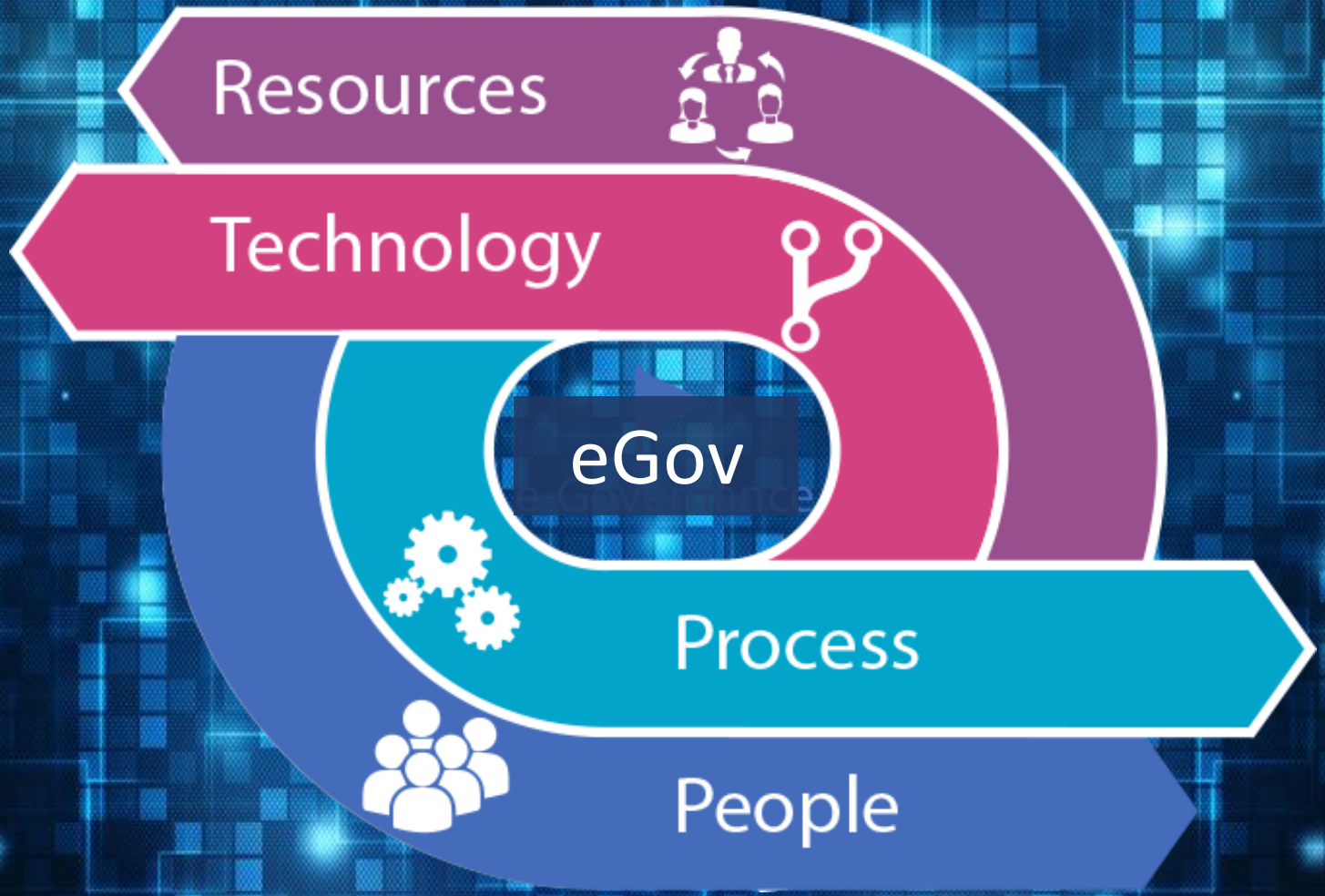
Christy Henitha JV

Technizant

Table of Contents

1. Brief about **eGov**
2. Brief about **Devops**
3. Brief about **Microservices**

1. Brief about eGov



Brief about eGov - Key Terms

eGovernments Foundation	**eGovernments Foundation** was established in 2003 by Nandan Nilekani (ex CEO - Infosys) and Srikanth Nadhamuni (CEO -Khosla Labs , ex CTO UIDAI)to partner with city administrators in their efforts to leverage technology for better service delivery and enhanced productivity. They developed **DIGIT**
DIGIT	**DIGIT** (Digital Infrastructure for Governance, Impact & Transformation) is India's largest platform for governance services.
Pura Seva	**Pura Seva** is a mobile app helping in **mGovernance.** It is a Citizen App to enable citizens to raise complaints and avail services digitally across 110 ULB (Urban Local bodies) in Andhra Pradesh
CityOS	**CityOS** platform is **OPEN SOURCE** microservices based API platform enabling quick rebundling of services as per specific needs. This repository contains various **infra, governance and municipal focused micro services.**
Societal Platform	**Societal Platforms** could create an environment that leverages the power of networks, and allows for the co-creation of solutions to resolve societal challenge. DIGIT is an example of societal platform

Brief about eGov – Need for DIGIT

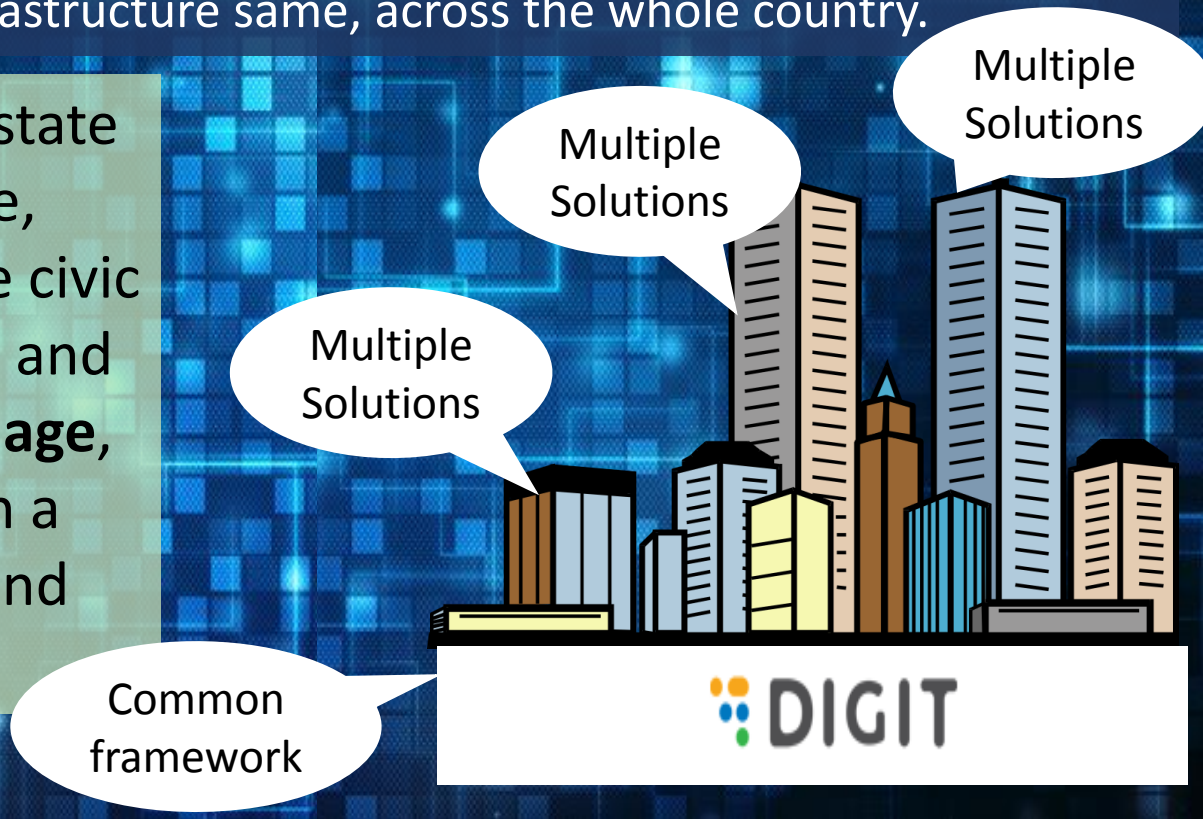
1. Indian Government focus is on **URBANIZATION / URBAN GOVERNANCE** for easy & efficient management of all public departments & drive state / country towards growth.
2. Urbanization refers to providing facilities of a good city like :
   Transparent process & immediate solution for prevailing problems in all areas including housing, education, tax, registration of land/birth/marriage etc. Providing good infra facilities, leading to growth of industries with its ancillaries with a favourable situation for employment & economic growth. Providing good health and medical facilities, drinking water supply and sanitation facilities
3. Already 30% urbanization done in India. In the decades to come, we are expecting 50% urbanization (half of the country).
4. Challenges in urbanization :
   - Integrated service delivery to citizens
   - Government process re-engineering
   - Mobile first approach
   - Cloud by default
   - Data security & preservation.

So…URBANIZATION IS A CHALLENGE AS WELL AS A OPPORTUNITY IN INDIA

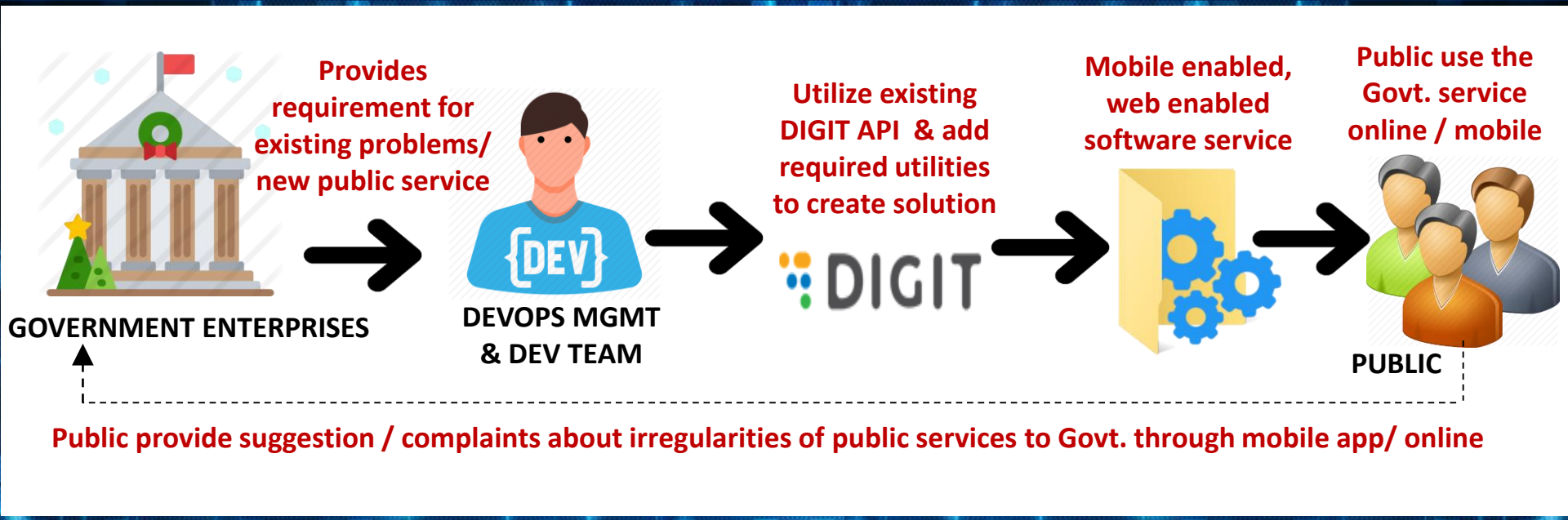
Brief about eGov – Need for DIGIT

5. We have to overcome the challenges of urbanization with speed & scale.
6. For achieving speed & scale in urbanization, we can use a shared digital infrastructure, which provides services online / mobile app.
7. The shared digital infrastructure is called as DIGIT (Digital Infrastructure Governance Impact Transformation)
8. DIGIT is a platform built on open API, inter-operable, open source.
9. DIGIT is available open source, in order to build multiple solutions, still keeping the basic framework & infrastructure same, across the whole country.

DIGIT is comprehensive, state of the art Open Source, interoperable and scalable civic tech platform for citizens and city governments to **manage**, **monitor** and **interact** in a transparent, effective and efficient manner.



Brief about eGov – DIGIT Work flow



1. DIGIT is a platform which is open source & open API powered for enabling ecosystem of partners, developers, enterprises and citizens to build new applications and solution.
2. The collaborative smart solutions empowers governments and citizens to meaningfully interact with each other and catalyze urban development.
3. The ready to use societal platform helps achieve quicker implementation time-frames and helps local governments achieve process improvements, accountability and transparency at various levels of administration
4. eGov is featured on the Open Forge portal, which maintains details of Govt. open source projects.(https://openforge.gov.in/project/showdetails.php?group_id=110)

Brief about eGov – Basic Operations

1. eGov solves hard to crack urban governance challenges using cutting edge open source technology DIGIT.
2. DIGIT helps governments become more citizen-centric, through user friendly cloud software, insightful data analytics and focus on mobile
3. It improves the urban citizen service delivery, increase the efficiency and effectiveness of the municipal staff and enable municipal executives to make data driven decisions to improve municipal operations.
5. The solutions include an ERP suite, Data Analytics, Visualization and Mobile apps. Together these enable efficient and effective operations, better decision making and contact-less, paperless urban service delivery
6. Ready to use, Configurable Urban Governance Platform.

- eGov has 15 years of experience in working with 450 ULB (Urban Local Body) & have 100+ employees for e-governance. It is funded by Nandan Nilekani (ex CEO - Infosys),Ratan Tata(Tata Trusts) & Omidyar Foundation.
- eGov is based in Bangalore with presence in Chennai, Andhra Pradesh & New Delhi.

Brief about eGov – Basic Operations

Fully Integrated with the **DIGIT** Ecosystem

Paperless

Cashless

Open Source

Automatic Fee Calculation

Integrated Workflow & Notifications

Highly Configurable

15	450	4 Cr
Year Focused on urban governance	Cities are being transformed on our platform	Citizens interact with our solutions

There are **4,000 cities** and towns in **India**. About 300 **cities** have population over 1,00,000. Seven **cities** have population more than 3 million

450 out of 4000 represents only 11%. We have scope for 3000+ cities

Brief about eGov – Basic Categories of Components

Revenue: Property Tax, Water Charges, Advertisement Tax,Trade License,Land and Estate,Sewerage

Administrative Efficiency: Financial Accounting System, Employee Management, Legal Case Management, Council Management, File Monitoring

Expenditure: Works Management, Payroll, Pension, Inventory/ Stores, Asset Management

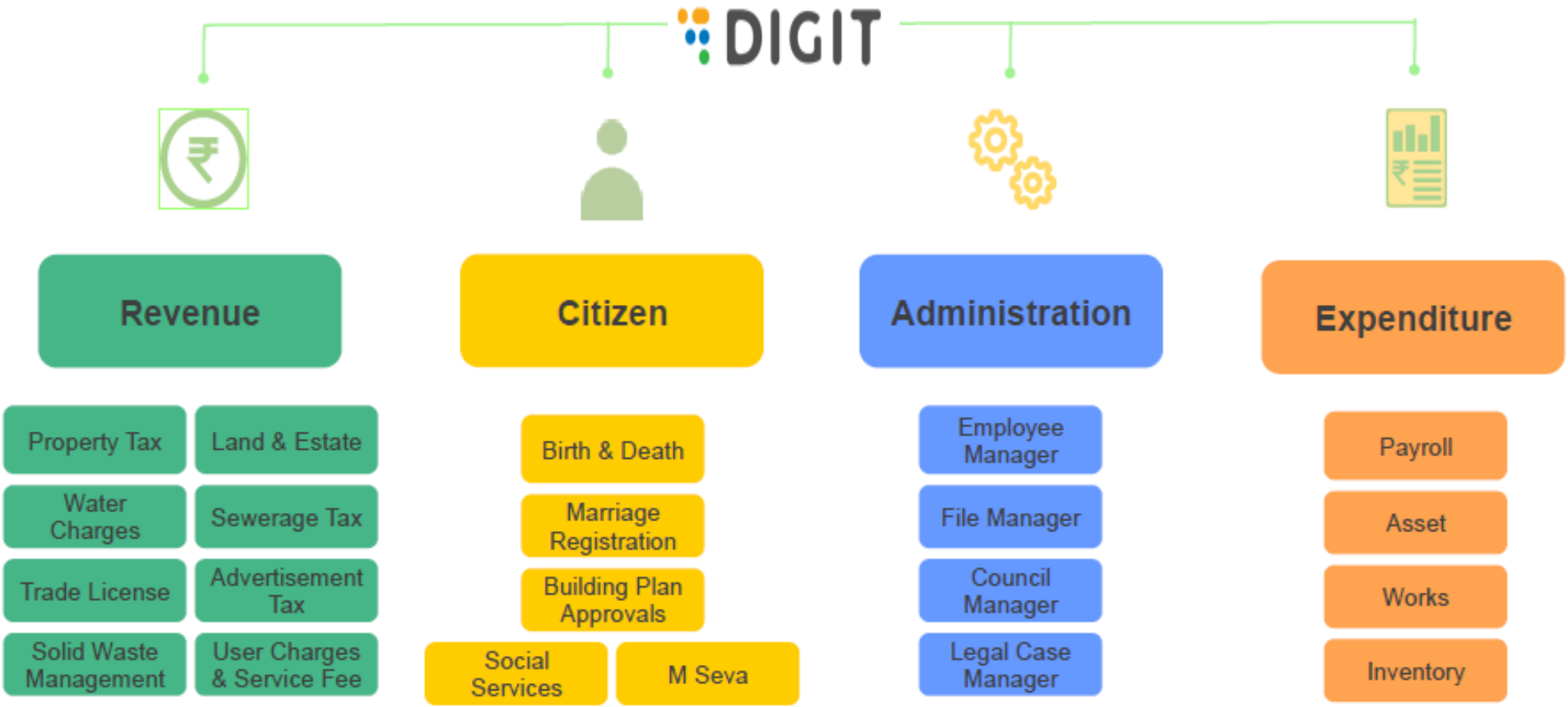
Mobile apps for the citizen and municipal employees, with service request, feedback and notification facility

Citizen Services: Public Grievance and Redressal, Birth and Death, Building Plan Approval, Marriage Registration, Citizen Portal

Integrated dashboard and analytics that would enable the municipal executives to take data driven decisions.

Brief about eGov – Basic Components

eGov DIGIT platform has 21 different apps.



Brief about eGov – Application/ outcome of DIGIT components

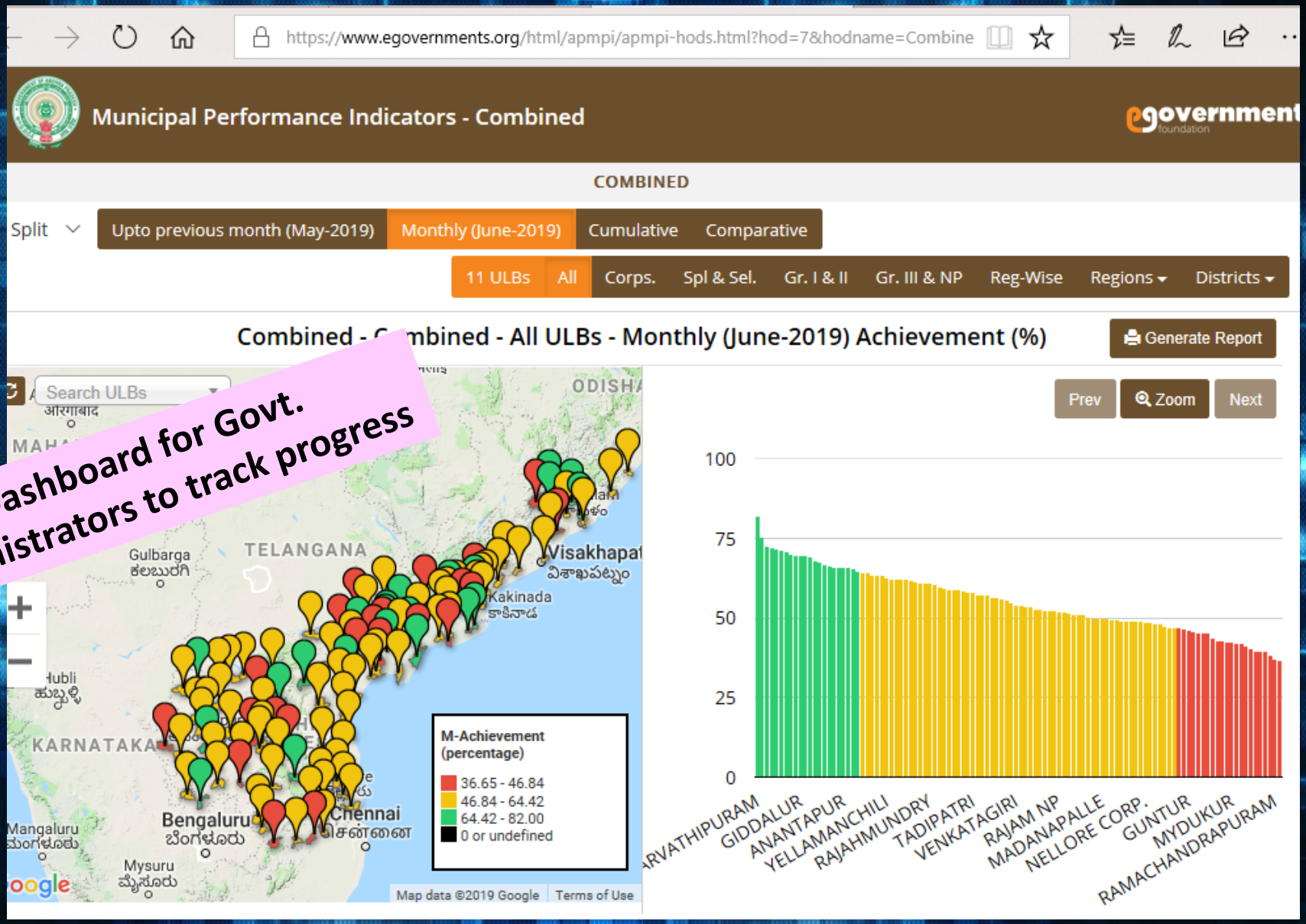
- **ERP for Municipalities** – to fully digitize civic body operations for both revenue and expenses management
- **Citizen Services** - digital service delivery to citizens for more than 30 services
- **Financial Management** – includes budgeting, forecasting and financial accounting that are compliant with National Municipal Accounting Standard
- **Digital Payment Systems** – enabling government offices to go cashless both internally as well as with citizens
- **Data Driven Governance** – Big Data Analytics Engine provides real time information& dashboards to administrator's and employees across various levels in government to drive data driven decisions making.
- **mGovernance** –a) Citizen App to enable citizens to raise complaints and avail services digitally across the state of AP and b) Employee App : A separate app that allows them to manage their workflow , upload pictures / evidence of the work done and provide status

Brief about eGov – Application/ outcome of DIGIT components

A short film on eGov Foundation, catalyzing urban governance and improving ease-of-living. Watch the 10-min feature film at https://youtu.be/Pn1KZrwOFzM

Mobile App for citizens to report problems & get quick services

Brief about eGov – Application/ outcome of DIGIT components



https://www.egovernments.org/html/apmpi/apmpi-hods.html?hod=7&hodname=Combined

Brief about eGov – Target Audience (End Users)

Citizens

Local Govt.

State Govt.

System Integrator

1 CITIZENS

- Access governments easily using your mobile phone
- Your requests automatically reach the appropriate person
- Stay up to date with inbuilt notifications

2 LOCAL GOVERNMENTS

- Realize the benefits of data driven governance
- Reduced paperwork for employees
- Efficient and prioritized use of resources

3 STATE GOVERNMENTS

- Enable all cities to become smart cities
- Drive best practice sharing across cities
- Improved urban budgeting and forecasting

4 SYSTEM INTEGRATORS

- Proven track record across India
- DIGIT reduces total project costs
- Ready to use & configurable governance platform

Brief about eGov – Partners

profsys Software Pvt. Ltd.

pwc

TATA
TATA CONSULTANCY SERVICES

TRIGYN technologies

UST Global®

wipro

BAHWAN CYBERTEK

भारत इलेक्ट्रॉनिक्स
BHARAT ELECTRONICS

EY

FLUENTGRID

IL&FS

InnoWave

KPMG

MapmyIndia

Mindtree
Welcome to possible

Massachusetts Institute of Technology

ncbs

NIUA
National Institute of Urban Affairs

Brief about eGov – Clients

NDMC

PMIDC

Nagpur Municipal Corporation

Greater Chennai Corporation

Government of Andhra Pradesh

Government of Karnataka

Guntur Municipal Corporation

https://www.digit.org/request-a-demo/

DIGIT

Request your DIGIT demo now

Full Name

Email

Phone No.

Organization Name

Message

For better understanding / queries, we can check with above ULB.
By requesting demo also, we will get better idea.
https://www.digit.org/request-a-demo/

For better understanding / queries, we can check with these options :

1. Co-Creator
2. Implementation/ deployment/ DIGIT partner

https://www.digit.org/partners/

Become a **Co Creator**

Co-Creators are integral to enriching the ecosystem and bring in a variety of new solutions which are not part of the standard platform. DIGIT will ease these Social Entrepreneurs through

- DIGIT specific Technical Documents
- Product Familiarisation Trainings
- Sandbox
- Access to a System Integrator Ecosystem

Become an **Implementation Partner**

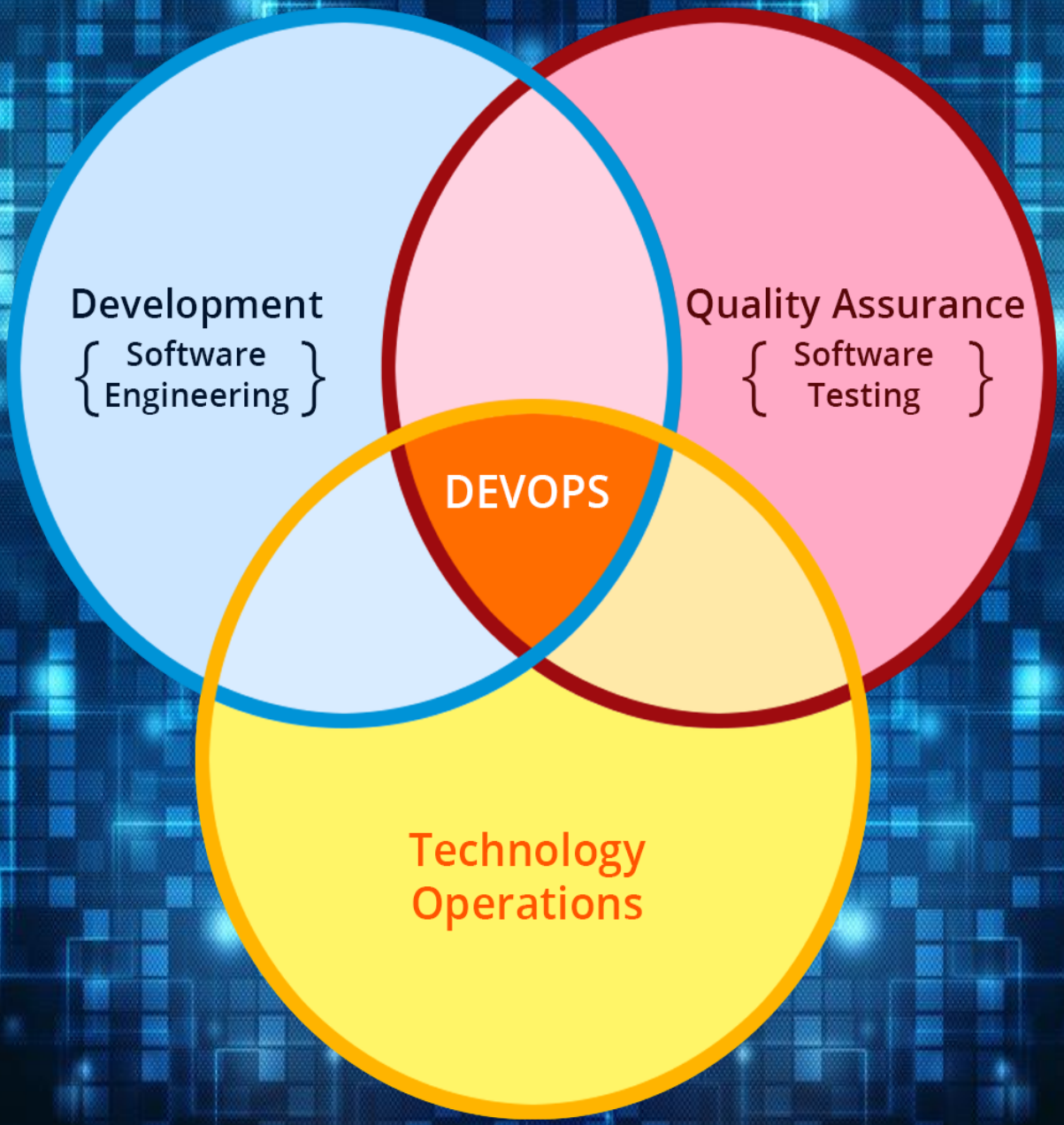
As key stakeholders in taking the platform to the end customers, Implementation Partners will be provided a host of services to empower them in customizing and driving implementation of DIGIT. Services will include:

- Capacity building documentation & materials
- Product Familiarisation Trainings & Certifications
- Digit Specific Technical Documents
- Consulting Support
- Configuration Consoles
- Access to a vibrant ecosystem
- Presales Support

Brief about eGov – Achievements

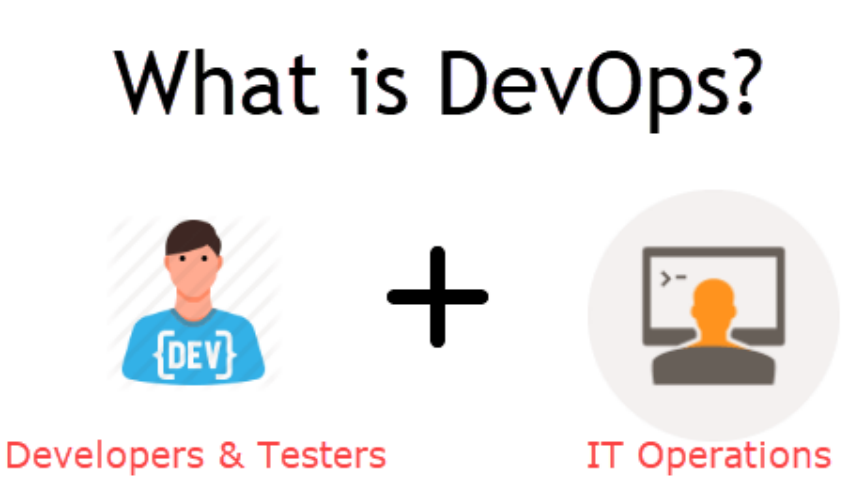
- eGov bags the prestigious "Order of Merit" at Business World Smart Cities Conclave & Awards, 2016. We were recognised for our implementation of "eGov Smart Platform" and “Puraseva” app across 110 ULBs in Andhra Pradesh.
- Chennai Corporation gets best e-governance award for implementing eGov SmartCity
- Nagpur Municipal Corporation gets Skoch Gold award 2014 for implementing eGov SmartCity
- eIndia 2008 - Best Government Initiative of the Year Award for implementing Property Tax Information System, a product in the eGov SmartCity suite.
- National Award for eGovernance 2010-11 - Excellence in Government Process Re-engineering for implementing property tax information system, a product from eGov SmartCity
- ICT Excellence 2010 Award for implementing three key products from the eGov SmartCity.
- Skoch 2010 - The world is open award for implementing public grievance redressal system, a product from eGov SmartCity
- Citizen’s Choice category of eINDIA Awards 2010 - Municipal IT Initiative of the year awarded to eGov SmartCity

2. Brief about Devops



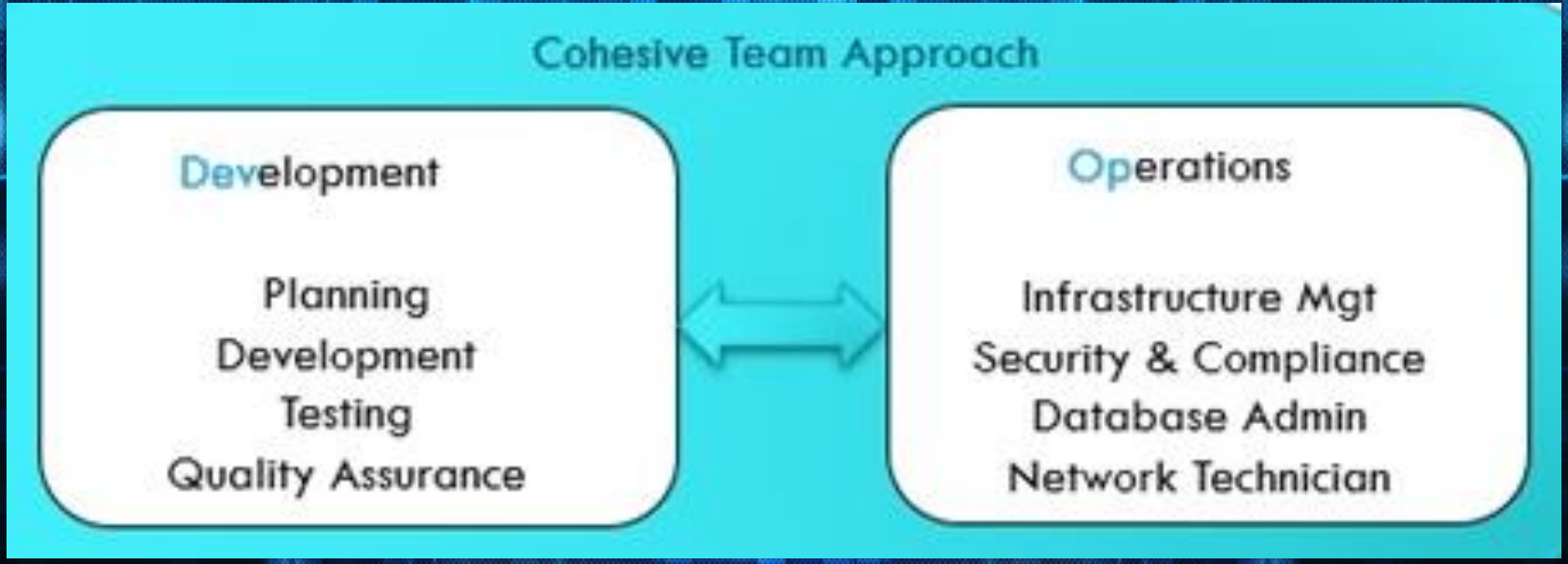
Brief about Devops – Basic meaning

- DevOps is a software development approach which involves Continuous Development, Continuous Testing, Continuous Integration, Continuous Deployment and Continuous Monitoring of the software throughout its development life cycle.
- DevOps is the preferred approach to develop high quality software in shorter dev cycles which results in greater customer satisfaction.
- This is why Facebook and other top companies have chosen DevOps as the way forward for their business goals.



Brief about Devops – Basic meaning

DevOps is a culture which promotes collaboration between Development, Testing and Operations Team to deploy code to production faster in a secure, automated & repeatable way.



The History of DevOps

Patrick Debois start's assessing IT Value Chain

Agile System Administrators Group is launched on Google

Inaugural "DevOps Days" are held in Ghent, Belgium

DEVOPS

Industry leading software vendors increase market presence with "Enterprise" class DevOps tools

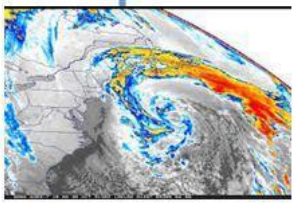
devX is born and Xceed launches the "12 days of DevOps"

2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015

Andrew Clay Shafer and Patrick Debois meet at Agile Conference 2008

John Allspaw and Paul Hammond present "10 Deploys per day" at Velocity

Open Source toolsets rip up the legacy playbook



The "Perfect Storm" of adjacent methodologies occurs

Cameron Haight predicts that DevOps will hit the big time in 2015 across Enterprise organisations

Gene Kim releases "The Phoenix Project"

DevOps begins to provide positive impact to "Enterprise" IT and experiences seismic adoption rates

What does the future of DevOps hold for your organisation?

Starting from 2009, it's a 10 year (a decade) journey for Deveops.

Brief about Devops – 6 Principles of Devops

1. Customer-Centric Action: DevOps team must take customer-centric action for that they should constantly invest in products and services.

2. End-To-End Responsibility: The DevOps team need to provide performance support until they become end-of-life. This enhances the level of responsibility and the quality of the products engineered.

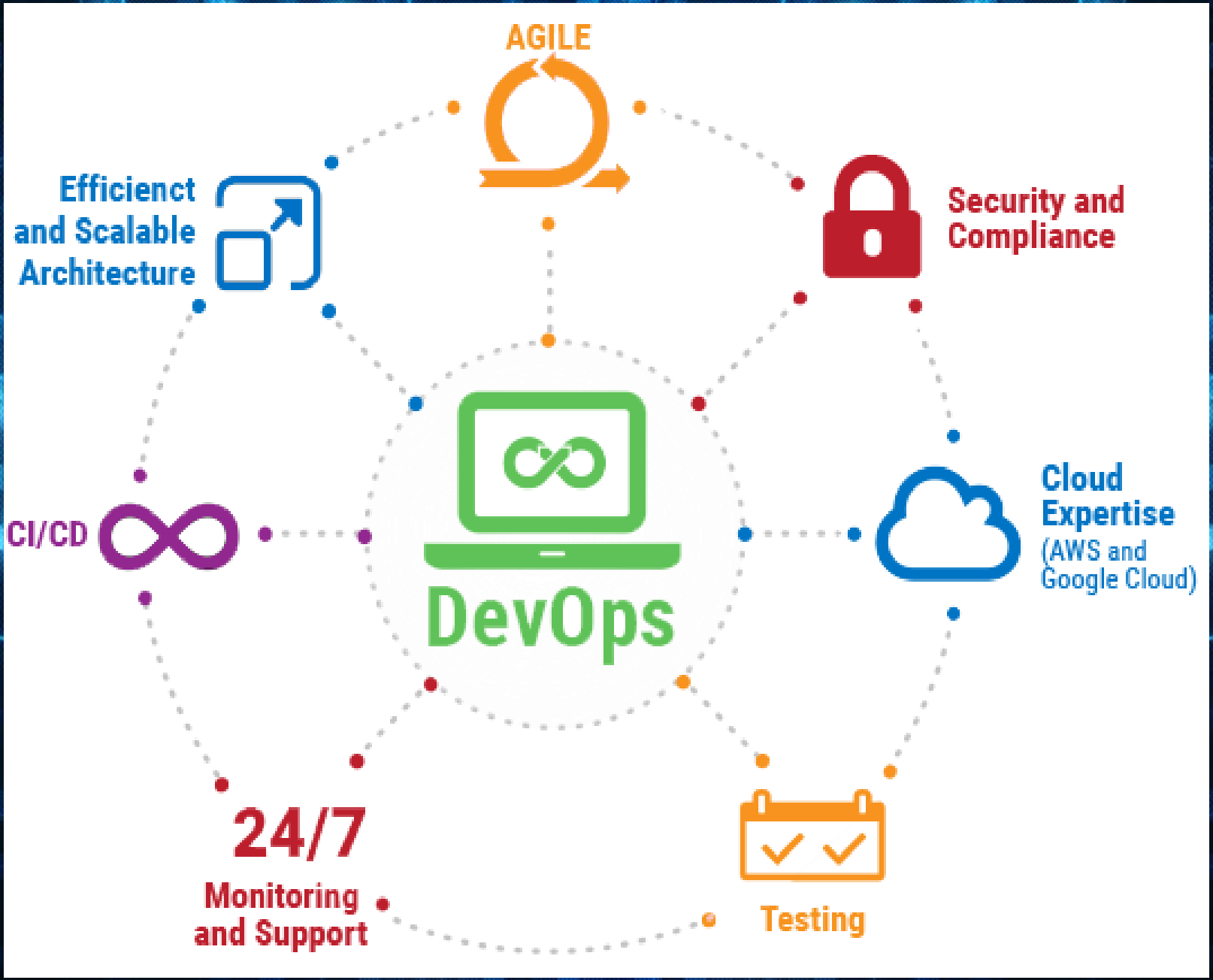
3. Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the improvement of product or services offered.

4. Automate everything: Automation is a vital principle of DevOps process. This is not only for the software development but also for the entire infrastructure landscape.

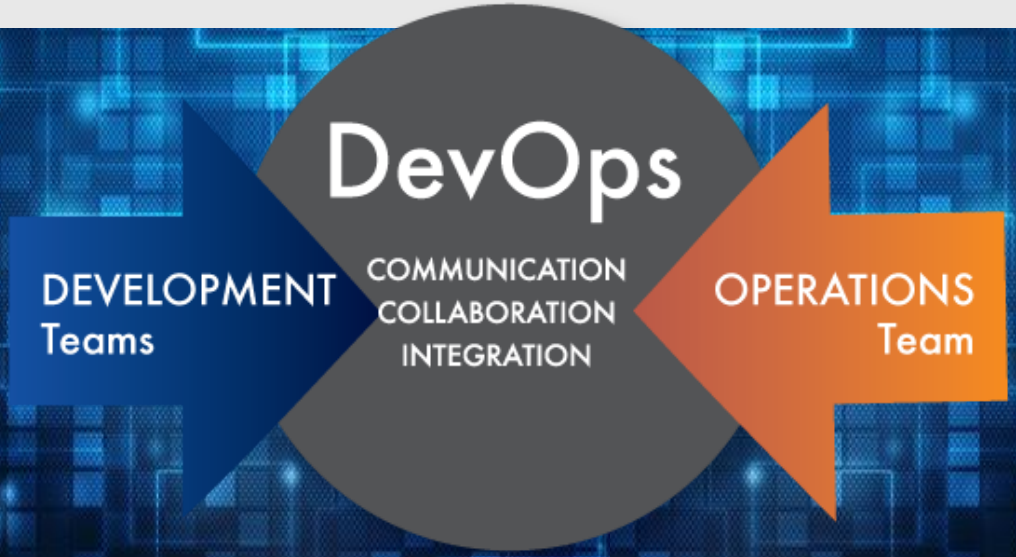
5. Work as one team: In the DevOps culture role of the designer, developer, and tester are already defined. All they needed to do is work as one team with complete collaboration.

6. Monitor and test everything: It is very important for DevOps team to have a robust monitoring and testing procedures.

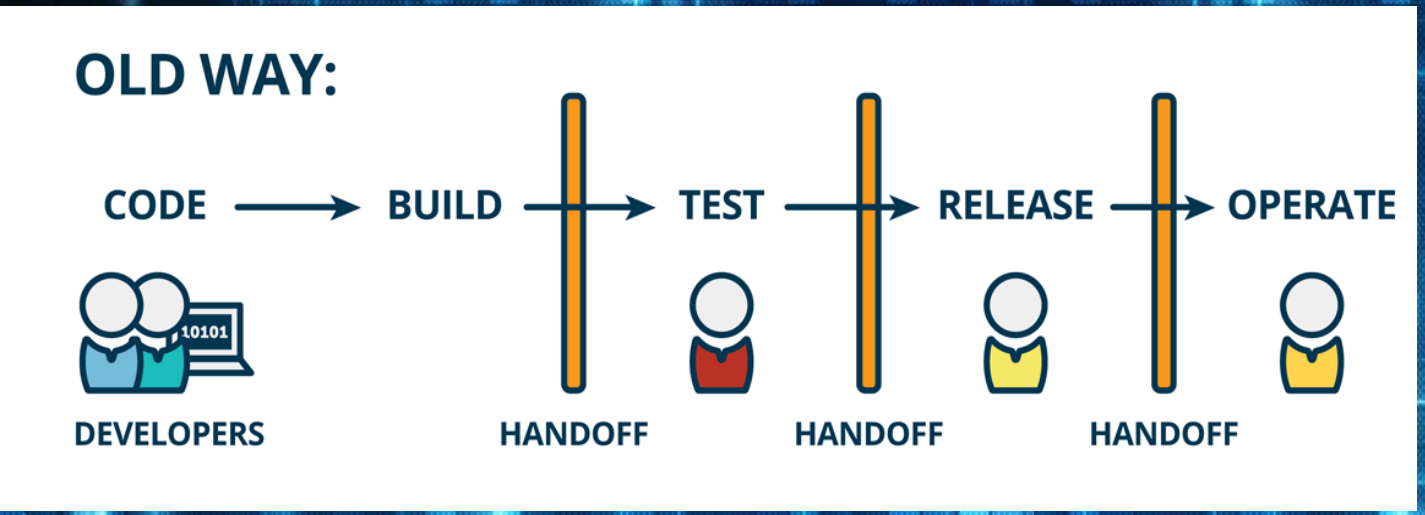
Brief about Devops – Main features of Devops

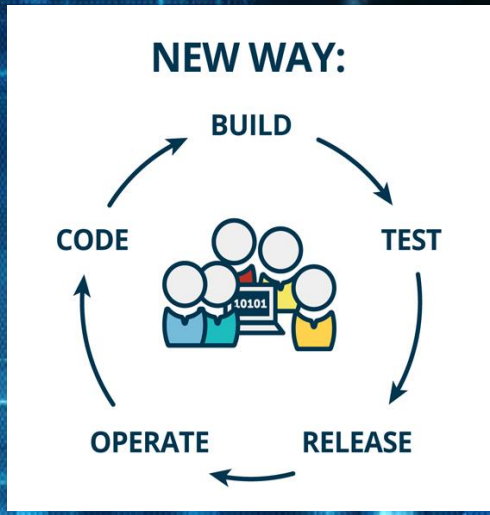


Brief about Devops – Traditional Vs Devops approach



Brief about Devops –Traditional Vs Devops approach





- Before DevOps, the development and operation team worked in complete isolation.
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in sync causing further delays.

There is a demand to increase the rate of software delivery by business stakeholders. As per **Forrester Consulting** Study, Only **17% of teams can use delivery software fast** enough. This proves the pain point in old approach & the demand of new approach **Devops** arises.

Brief about Devops – Traditional Vs Devops approach

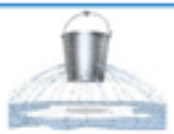
Let's compare traditional software waterfall model with DevOps to understand the changes DevOps bring.

We assume the application is scheduled to go live in 2 weeks and coding is 80% done. We assume the application is a fresh launch and the process of buying servers to ship the code has just begun

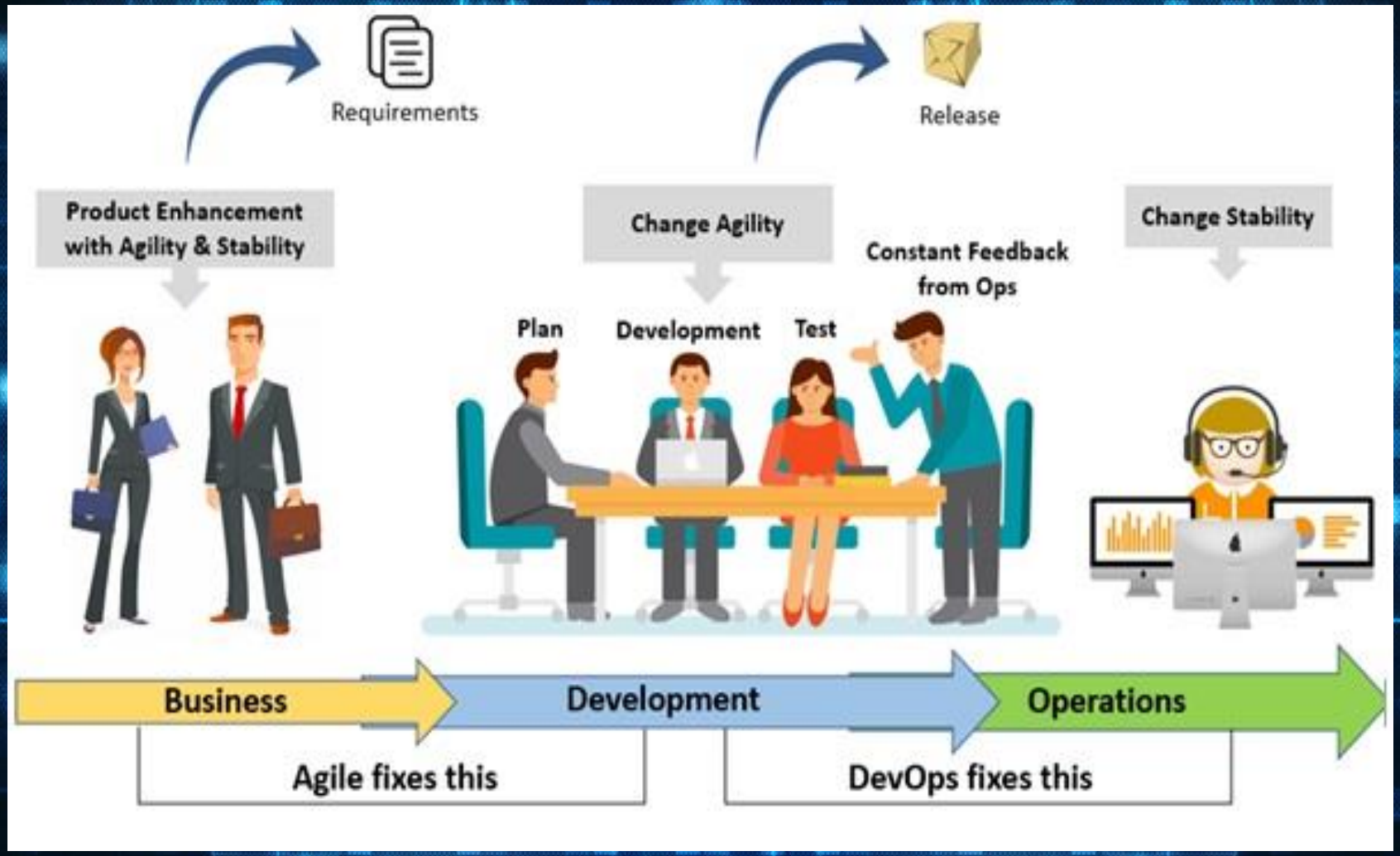
Old Process	DevOps
After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.	After placing an order for new servers Development and Operations team work together on the paperwork to set-up the new servers. This results in better visibility of infrastructure requirement.
Projection about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers who have deep knowledge of the application.	Projection about failover, redundancy, disaster recovery, data center locations, and storage requirements are pretty accurate due to the inputs from the developers.
Operations team has no clue on the progress of the Development team. Operations team develop a monitoring plan as per their understanding.	In DevOps, the Operations team is completely aware of the progress the developers are making. Operations team interact with developers and jointly develop a monitoring plan that caters to the IT and business needs. They also use advance Application Performance Monitoring (APM) Tools
Before go-live, the load testing crashes the application. The release is delayed.	Before go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks. The application is released on time.

Brief about Devops – Traditional Vs Devops approach

	Dev Challenges	DevOps Solution
	Waiting time for code deployment	• Continuous Integration ensures there is quick deployment of code, faster testing and speedy feedback mechanism • Thus there is no waiting time to deploy the code. Hence the developer focuses on building the current code
	Pressure of work on old, pending and new code	

	Ops Challenges	DevOps Solution
	Difficult to maintain uptime of the production environment	Containerization / Virtualization ensures there is a simulated environment created to run the software as containers offer great reliability for service uptime
	Tools to automate infrastructure management are not effective	Configuration Management helps you to organize and execute configuration plans, consistently provision the system, and proactively manage their infrastructure
	No. of servers to be monitored increases	Continuous Monitoring Effective monitoring and feedbacks system is established through Nagios Thus effective administration is assured
	Difficult to diagnose and provide feedback on the product	

Brief about Devops – Traditional Vs Devops approach



Brief about Devops – Outcome/ benefits of Devops

DevOps allows Agile Development Teams to implement **Continuous Integration** and **Continuous Delivery**. This helps them to launch products faster into the market.

1. Predictability: DevOps offers significantly lower failure rate of new releases
2. Reproducibility: Version everything so that earlier version can be restored anytime.
3. Maintainability: Effortless process of recovery in the event of a new release crashing or disabling the current system.
4. Time to market: DevOps reduces the time to market up to 50% through streamlined software delivery. This is particularly the case for digital and mobile applications.
5. Greater Quality: DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues.
6. Reduced Risk: DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle.
7. Resiliency: The Operational state of the software system is more stable, secure, and changes are auditable.
8. Cost Efficiency: DevOps offers cost efficiency in the software development process which is always an aspiration of IT companies' management.
9. Breaks larger code base into small pieces: DevOps is based on the agile programming method. Therefore, it allows breaking larger code bases into smaller and manageable chunks.

Devops emphasizes on 2 basic concepts

Continuous Integration (CI)

Continuous Delivery (CD)

Brief about Devops – Outcome/ benefits of Devops

Since then, Netflix has continued its commitment to automation and open source, and today engineers deploy code thousands of times per day. This year, Netflix was unanimously selected for the JAX Special Jury Award, with JAXenter editor Coman Hamilton declaring, "The rate at which this entertainment game-changer has adopted new technologies and implemented them into its DevOps approach is setting new standards in IT."

While Walmart is the king of big box retailers in the American heartland, online it has always struggled in the shadow of Amazon. To gain ground, it assembled a cutting-edge team through several tech acquisitions and founded WalmartLabs, the retailer's technology innovation and development arm, in 2011.

WalmartLabs has taken a decidedly DevOps approach to its mission. It incorporated OneOps cloud-based technology, which automates and accelerates application deployment.

DevOps gave us an edge

"DevOps has helped us do very frequent releases, giving us an edge on time to market. We are now able to make daily product releases as opposed to 6-month releases, and push fixes to our customers in a span of a few hours."

– Hamesh Chawla, VP of Engineering at Zephyr

Brief about Devops – Wnen to & When NOT to use Devops

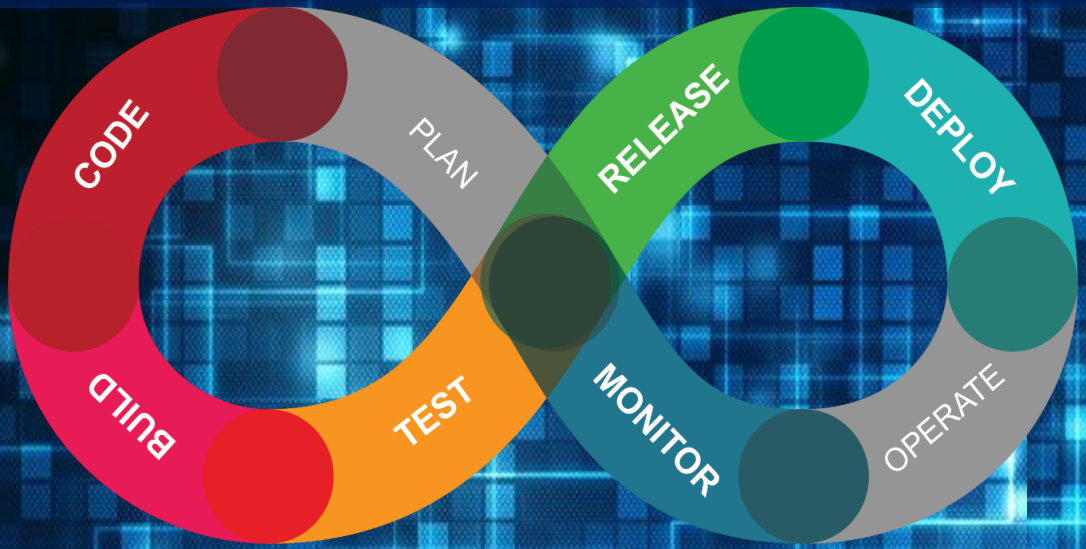
When to adopt DevOps?

DevOps should be used **for large distributed applications** such as eCommerce sites or applications hosted on a cloud platform.

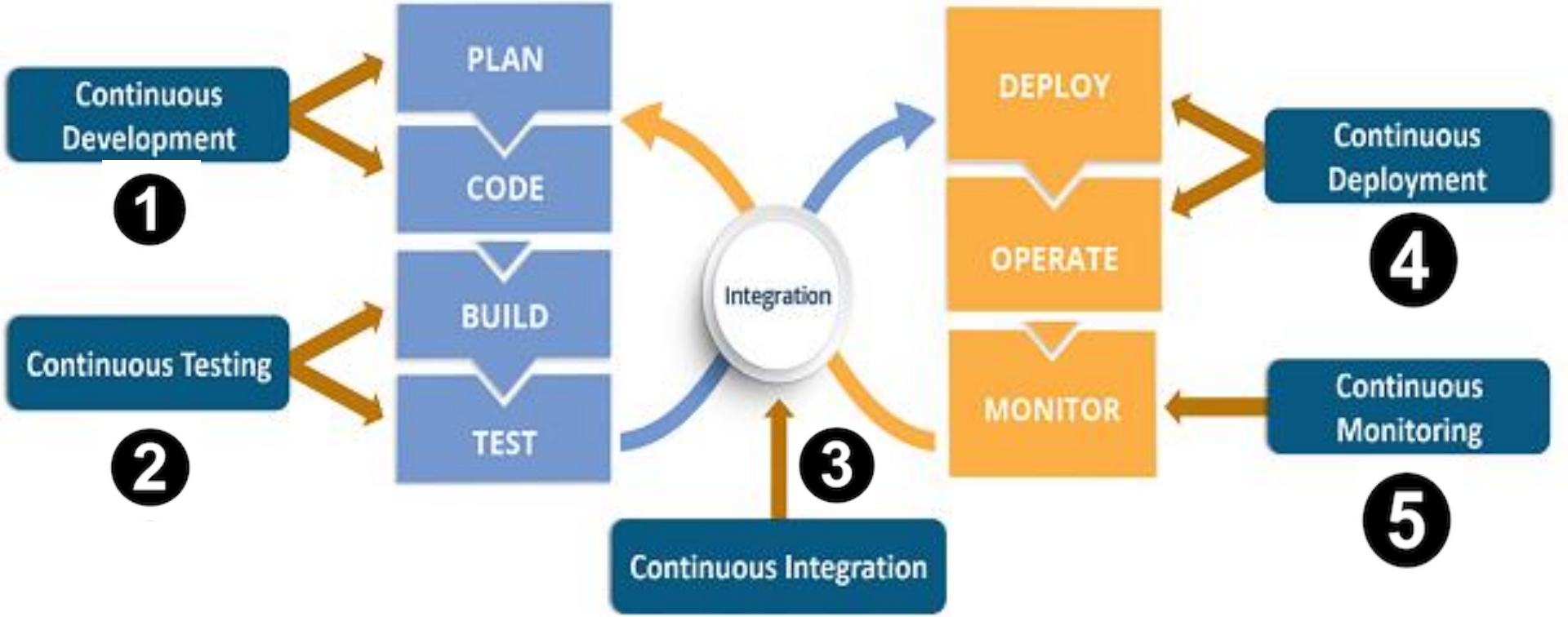
When NOT to adopt DevOps?

It should not be used in a mission-critical application like bank, power and other sensitive data sites. Such applications need strict access controls on the production environment, a detailed change management policy, access control policy to the data centers.

Brief about Devops – Devops Lifecycle

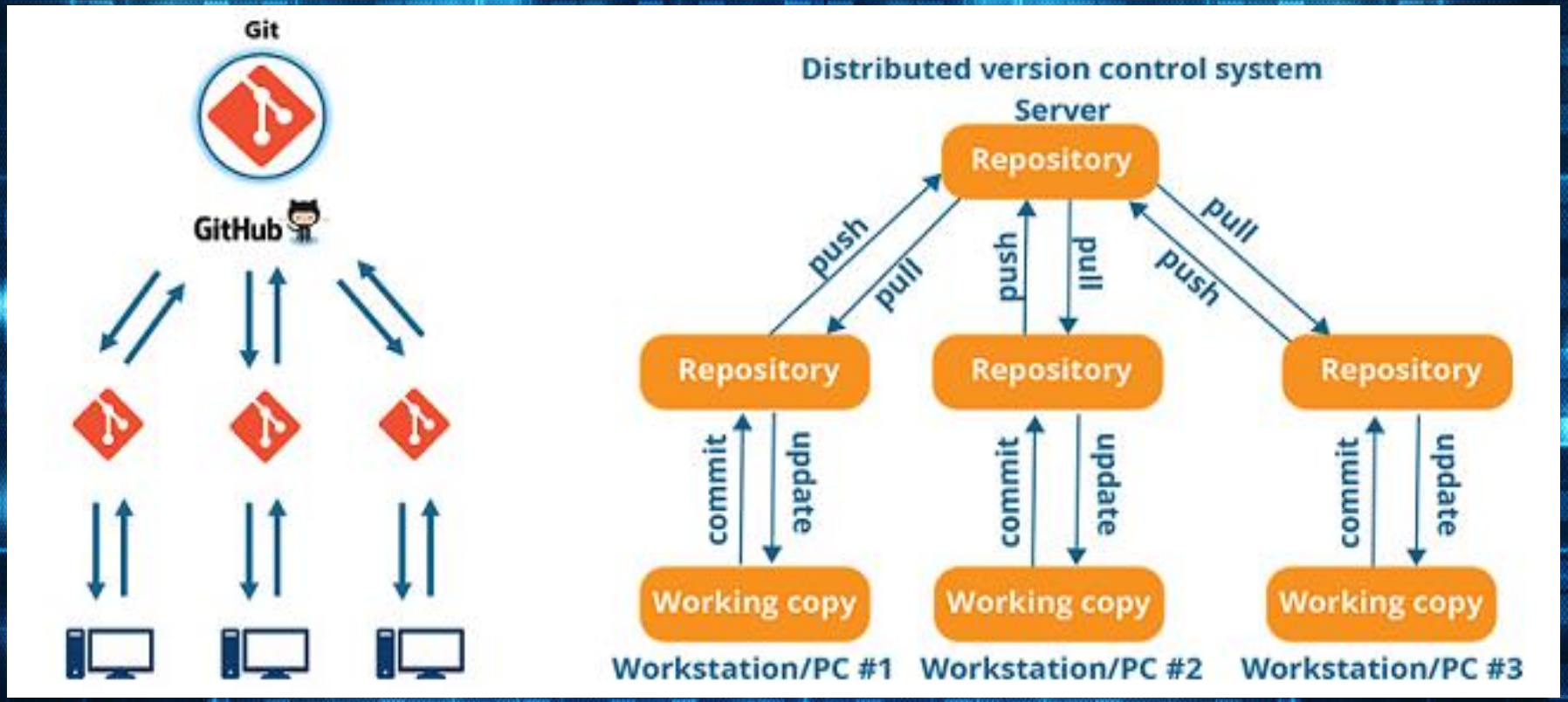


1. Continuous Development
2. Continuous Testing
3. Continuous Integration
4. Continuous Deployment
5. Continuous Monitoring



Continuous Development (Plan & Code)

1



Continuous Development (Plan & Code)

1

- This is the phase that involves ‘planning‘ and ‘coding‘ of the software. The vision of the project is decided during the planning phase and the developers begin developing the code for the application. There are tools like **JIRA** for planning requirements.
- The code can be written in any language, but it is maintained by using Version Control tools. Maintaining the code is referred to as Source Code Management. The most popular tools used are **Git, SVN, Mercurial, CVS**. Also tools like **Ant, Maven, Gradle** can be used in this phase for building/ packaging the code into an executable file that can be forwarded to any of the next phases.

Continuous Testing (Build & Test)

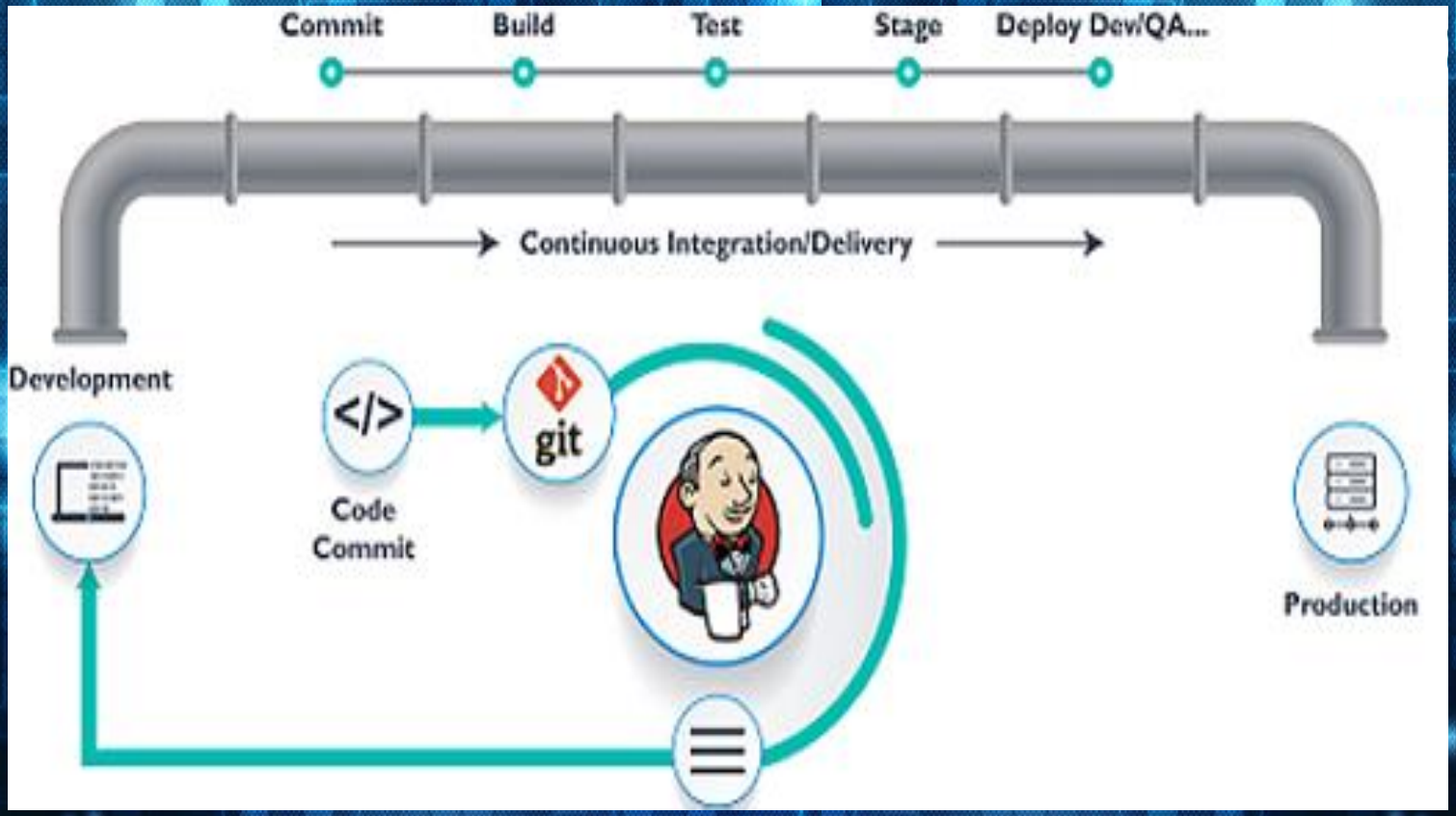
2

Continuous Testing (Build & Test)

2

- This is the stage where the developed software / build is continuously tested for bugs For continuous testing, automation testing tools like **Selenium, TestNG, JUnit**, etc are used. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there are no flaws in the functionality. In this phase, **Docker Containers** can be used for simulating the test environment.
- **Building** code not only involves compilation but it also includes code review, unit testing, integration testing, and packaging. Selenium does the automation testing, and the reports are generated by TestNG. This entire testing phase can be automated with the help of a Continuous Integration tool called Jenkins. Suppose you have written a selenium code in Java to test your application. Now you can build this code using **ant** or **maven**. Once the code is built, it is tested for User Acceptance Testing (UAT). This entire process can be automated using **Jenkins**.

Continuous Integration (Release)



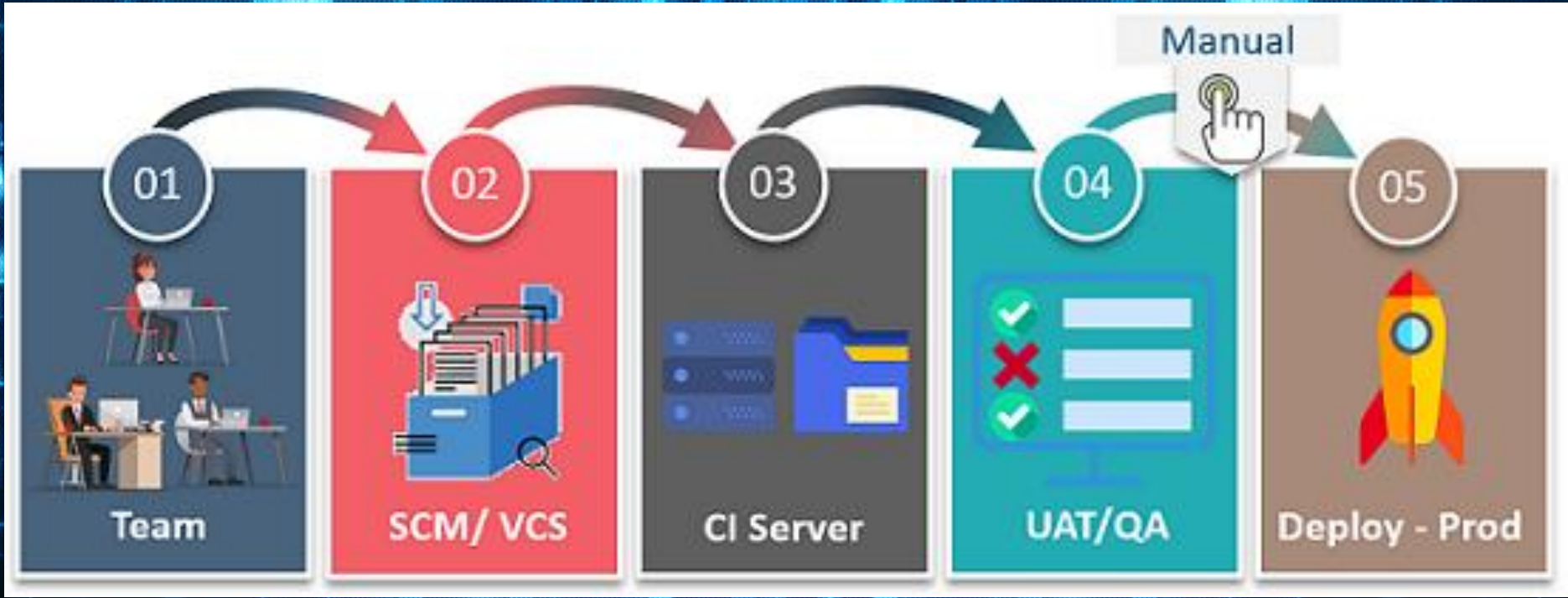
Continuous Integration (Release)

3

- This stage is the heart of the entire DevOps life cycle. The code supporting new functionality is **continuously integrated with the existing code**. Since there is continuous development of software, the updated code needs to be integrated continuously as well as smoothly with the systems to **reflect changes to the end-users**.
- **Jenkins** is a very popular tool used in this phase. Whenever there is a change in **the Git repository,** Jenkins fetches the updated code and it prepares a build of that code which is an executable file in **the form of a war or a jar.** This build is then forwarded to the **test server or the production server.**

Continuous Deployment (Deploy & Operate)

4



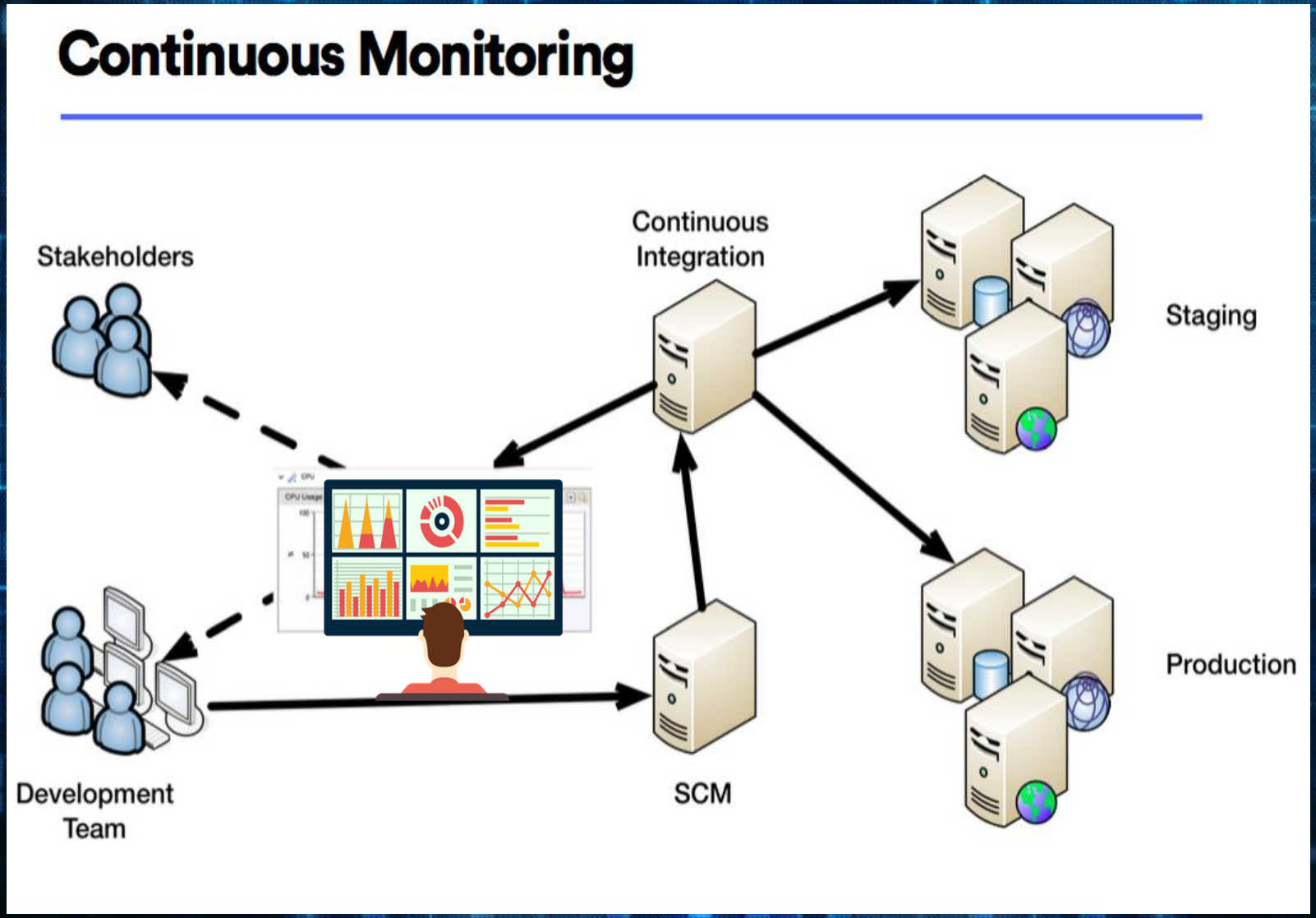
Continuous Deployment (Deploy & Operate)

4

- This is the stage where the code is deployed to the **production servers**. It is also important to ensure that the code is correctly deployed on all the servers.
- **Configuration management and containerization tools** help in achieving Continuous Deployment (CD).
- **Configuration management tools** help in releasing deployments to servers, scheduling updates on all servers and most importantly keeping the configurations consistent across all the servers. Since the new code is deployed on a continuous basis, configuration management tools play an important role in executing tasks quickly and frequently. Some popular tools that are used here are **Puppet, Chef, SaltStack, Ansible**
- **Containerization tools** help in maintaining consistency across the environments where the application is developed, tested and deployed. Using these tools, there is no scope of errors/ failure in the production environment as they package and replicate the same dependencies and packages used in the development/ testing/ staging environment. It makes your application easy to run on different computers. Besides this, they also help in scaling-up and scaling-down of instances swiftly **Docker & Vagrant** are the popular tools.

Continuous Monitoring (Monitor)

5



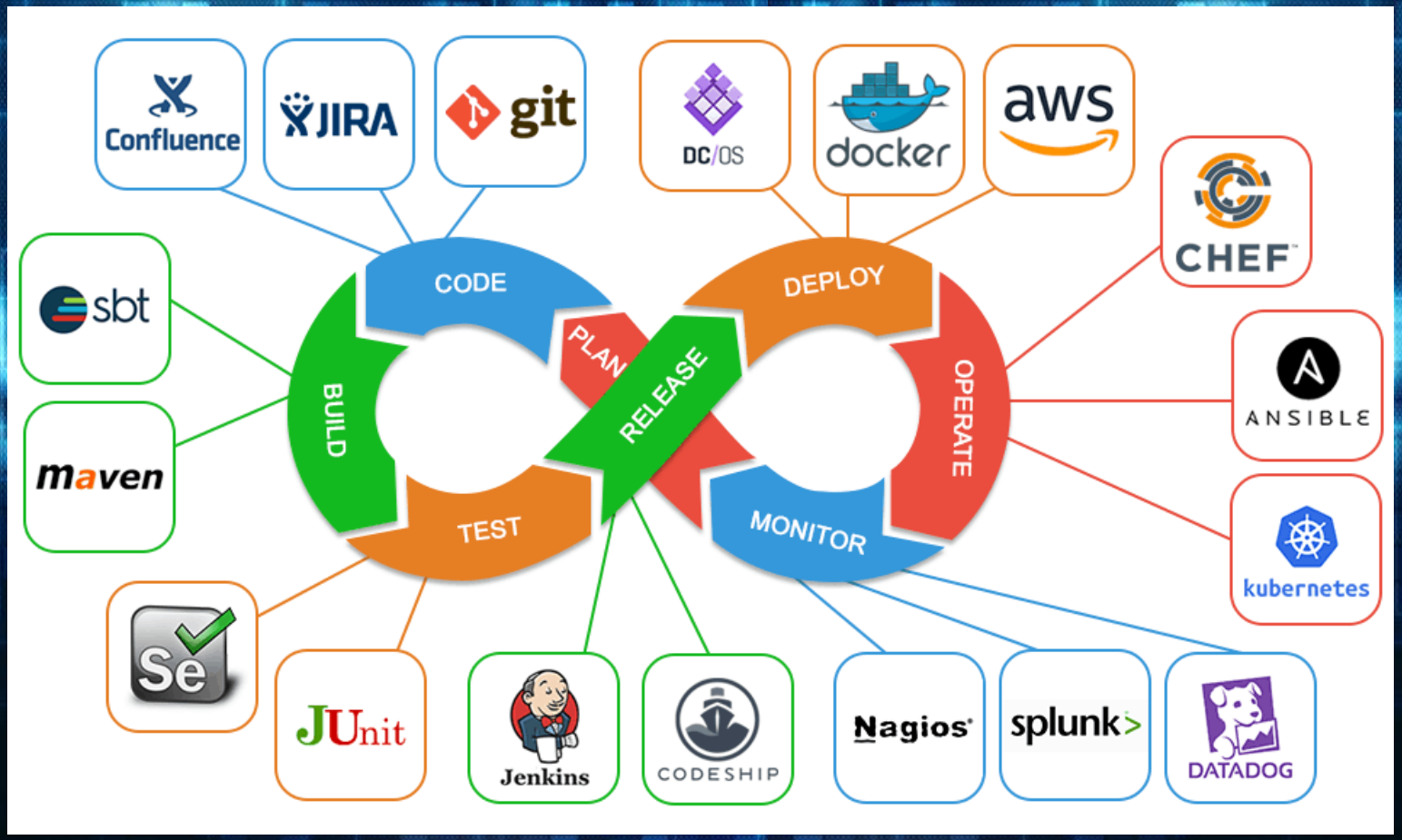
Continuous Monitoring (Monitor)

5

- This is a very crucial stage of the DevOps life cycle where you continuously **monitor & measure** the performance of application. Vital information about the use of the software is recorded. **System errors such as low memory, server not reachable, etc are resolved** in this phase.
- The root cause of any issue is determined in this phase. It **maintains the security and availability of the services**. Also if there are **network issues**, they are resolved in this phase. It helps us automatically fix the problem as soon as they are detected.
- The popular tools used for this are **Splunk, ELK Stack, Nagios, NewRelic, Sensu**. These tools help you monitor the application's performance and the servers closely and also enable you to check the health of the system proactively.
- They can **also improve productivity and increase the reliability** of the systems, which in turn **reduces IT support costs.** Any major issues if found are reported to the development team so that it can be fixed in the continuous development phase. This leads to **a faster resolution of issues**.

Brief about Devops – Tool chain & Automation

- Devops has a cross-functional mode of working, and in each phase, we use a different sets of tools, rather than a single one - referred to as **TOOL CHAIN**
- The output of one tool goes as the input of other tool.
- Devops automation involves **automation of automation tools**, by scripting the interaction among the tools.



Brief about Devops – How to get great results in Devops?

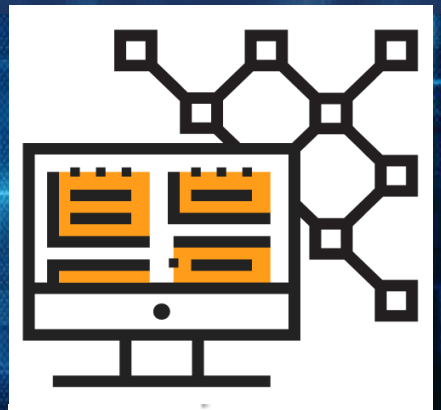
Agile Principles + Cloud Computing + DevOps Automation --> Great results

- Apart from team collaboration & knowledge sharing, **Agile principles** serve as a driver for DevOps implementation, as there are shorter sprints for releases and develops automation helps to achieve the deadlines. **Continuous Integration** and **Continuous Delivery** helps us to launch products faster into the market.
- In **cloud computing,** resources like memory, platform, application software are available on demand. The cloud helps to create a runtime environment in a short span of time.
- **DevOps automation** can be achieved by repackaging platforms, systems, and applications into reusable building blocks through the use of **virtualization, containerization, orchestration & Infrastructure as code**

VIRTUALIZATION

CONTAINERIZATION

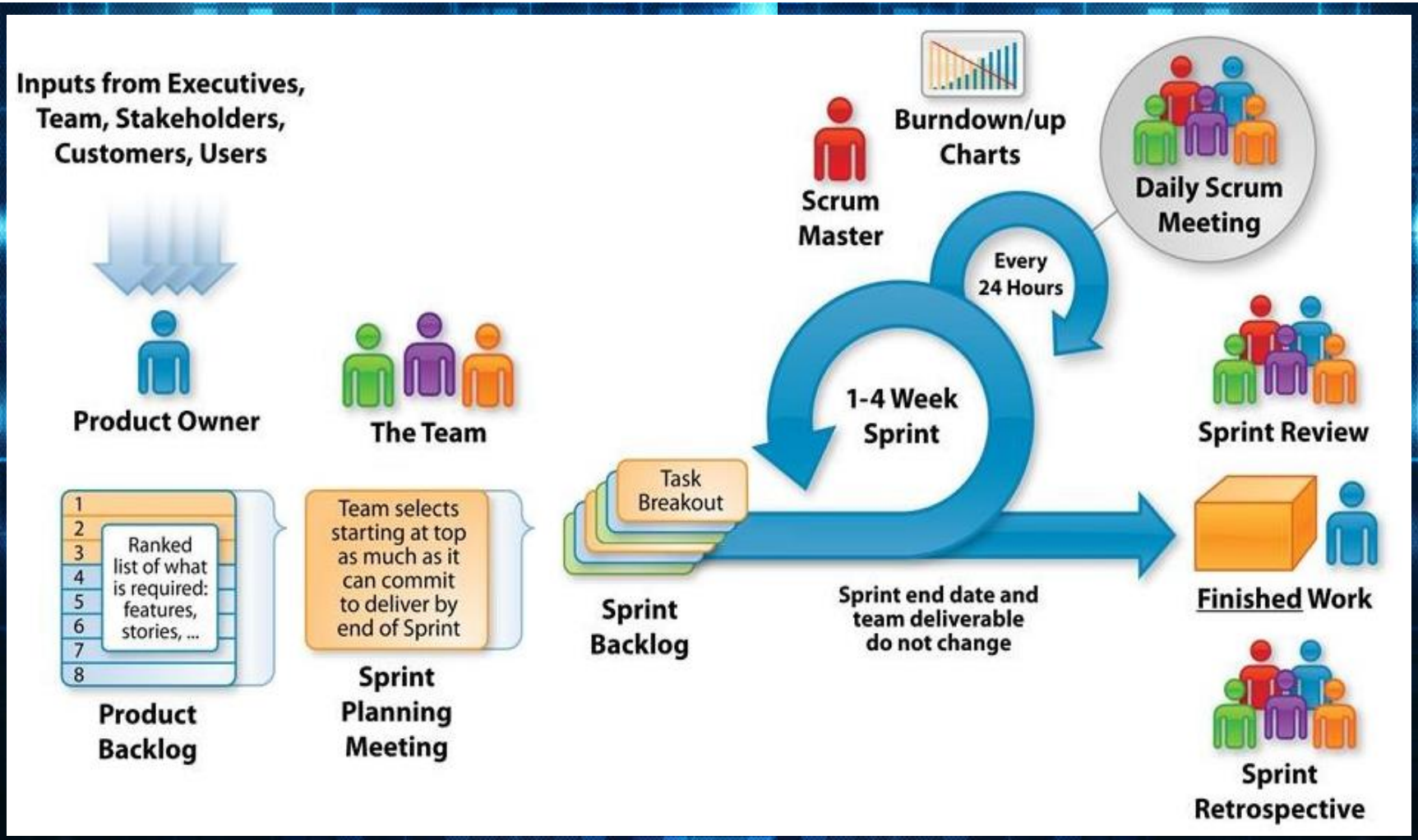
ORCHESTRATION



INFRASTRUCTURE AS CODE

Brief about Devops – Agile model of software development

1. **Individuals and interactions** over processes and tools
2. **Working software** over comprehensive documentation
3. **Customer collaboration** over contract negotiation
4. **Responding to change** over following a plan



Brief about Devops – Continuous Integration / Continuous Delivery

Continuous Integration (CI)

- **Continuous Integration (CI)** is the practice of developers ensuring that code is frequently checked in and integration tested with its dependencies with each check-in
- **It helps in reducing time & effort between Dev & Test team.**

Continuous Delivery (CD)

- **Continuous Delivery (CD)** is the practice of delivering reliable, low-risk releases to client with UAT (user acceptance test) done frequently to check user feedback to avoid surprises/failures in the final stage of release
- **It helps in reducing time & effort between Dev/Test team & Operations team/ client.**

With CI & CD…Development, Test, Support and Operations team work together as **one delivery team** to automate and streamline the build, test and release process.

CI / CD Tools:

Jenkins

CloudBees CodeShip

Bamboo

GitLab

circleci

Microsoft Visual Studio Team Services

Brief about Devops – Cloud computing

In **cloud computing**, resources like memory, platform, application software are available on demand. The cloud helps to create a runtime environment in a short span of time.

Purchasing own TV Antenna / Dish for watching TV channels involves high investment cost and maintenance is difficult.

Instead, we can pay cable TV operator a minimum amount for 'Pay as you use' channels. No setup costs & maintenance free

Similarly, in cloud computing, we can **use software, platform (operating system), memory also as 'pay as you use' service** for a minimum cost, instead of investing huge on high configuration servers, which we will not utilize to its maximum capacity & maintenance / server downtime is also a challenge.

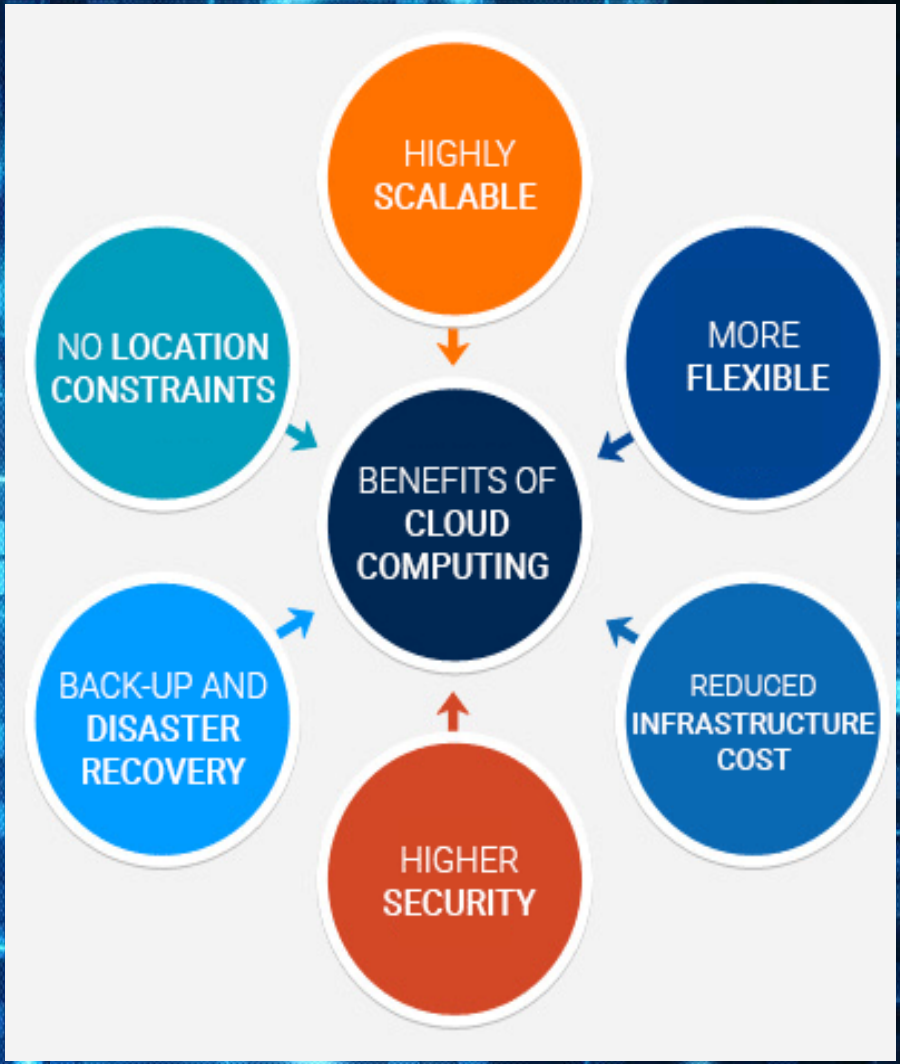
Brief about Devops – Cloud computing

Cloud service models

SaaS (Software as a service,
PaaS (Platform as a service)
IaaS (Infrastructure as a service)

Cloud deployment models

Public Cloud
Private Cloud
Hybrid Cloud
Community Cloud



Brief about Devops – Cloud computing

Cloud service models

- **Software as a Service:** We use SaaS day in day out. The software or application is available as a service and hence we don't need to manage or install it. The application is provided by the application owner, and all the responsibilities of managing the updates of that application are of the owner and not of the consumer. Google Docs is one example of SaaS.
- **Platform as a Service:** PaaS is a model where a platform or a runtime environment is provided by the service provider. It is the responsibility of the cloud consumer to upload the application and manage an up-to-date version of the application or software. Resources are managed by the cloud service provider. Backup and all other responsibilities also lie with the service provider. PaaS offerings provide features to maintain high availability and fault tolerance, but these need to be configured by cloud consumers. Amazon Elastic Beanstalk and Microsoft Azure App Services are two popular examples of Platform as a Service.
- **Infrastructure as a Service:** IaaS is a service model where the entire responsibility of managing the infrastructure, installing and configuring the runtime environment, and the configuration of high availability and fault tolerance is with the cloud consumer. Cloud consumers have no control over cloud resources (physical resources), but can configure or control operating systems, networks, storage, firewalls, and so on.

Brief about Devops – Cloud computing

Cloud Deployment models

Public cloud: Here, the cloud resources are made available to the public over the Internet. Since multiple organisations and individual users avail and share these resources, security is important while using the public cloud.

Private cloud: Here, the cloud resources are made available to specific units within the perimeter of an organisation. Security and governance is maintained by the organisation, and resources are shared by multiple business units of the same organisation.

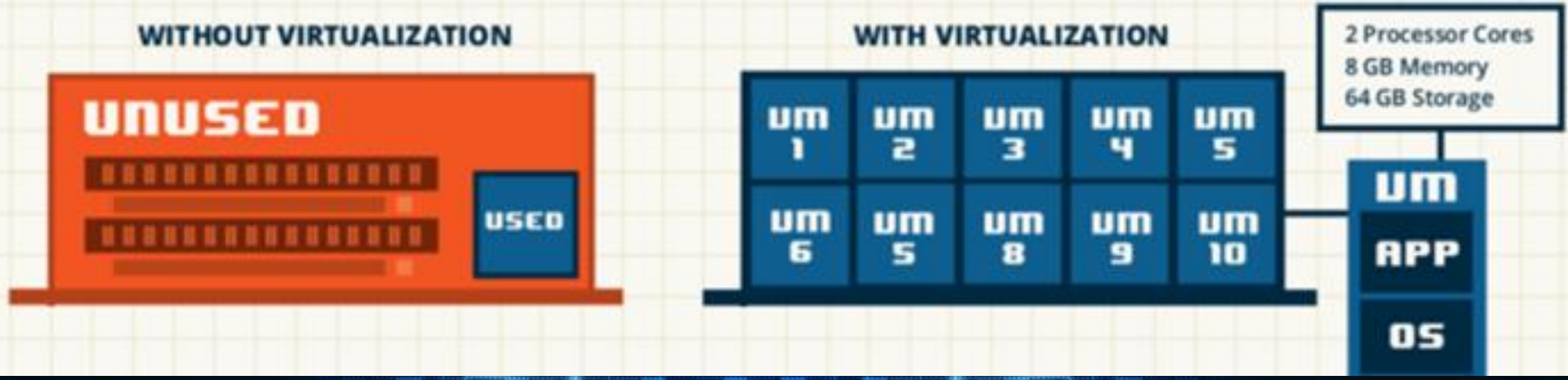
Hybrid cloud: Organisations where security is the main concern use the hybrid cloud. Different types of cloud deployment models are used to address security concerns and for effective utilisation of resources.

Community cloud: Here, the cloud resources are made available to a specific community of cloud consumers, such as life science based organisations or banks.

Brief about Devops – Virtualization

To better understand **virtualization** concept, consider that one giant room is available for one guest. Nobody needs that much space, so 90% of the building goes unused.

Similarly using virtualization, it will be more efficient to divide a computer's resources–like processors, memory and storage, then assign these resources to virtual machines and each one is capable of hosting its own OS and application.



Brief about Devops – Virtualization

- **Virtualization** is the process of creating software that mimics various hardware and software environments without having to change out the physical hardware itself. This allows a single piece of hardware to host to numerous VM configurations simultaneously.
- Virtualization enables DevOps teams to develop and test within simulated environments that run the full gamut of devices available to consumers while also testing deployment on virtual live environments.
- This enables development to occur alongside real-time testing of how the changes will impact the entire system.
- This level of accuracy in testing makes for vastly reduced deployment times and increased stability.

- **Virtual machine**: a virtual machine (VM) is an emulation of particular computer system, its resources such as CPU, RAM and DISK space.

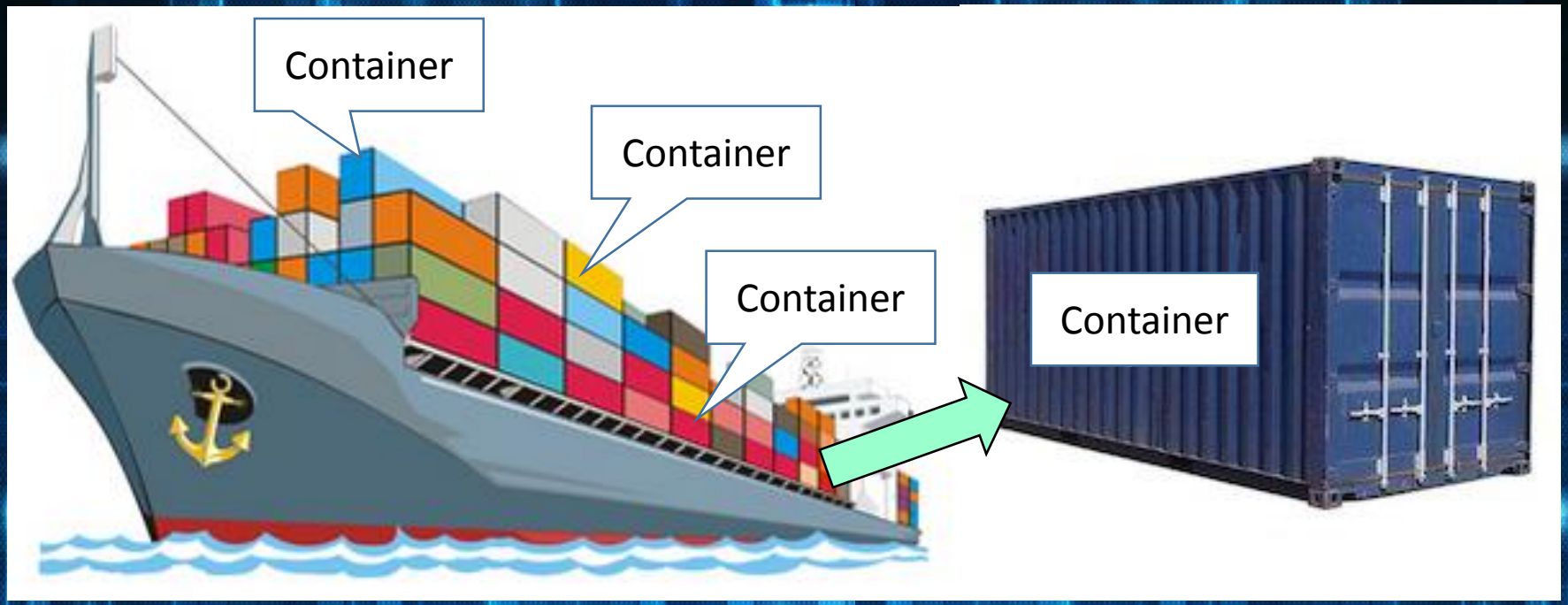
- **Hypervisor:** aka Virtual Machine Monitor (VMM) is a piece of computer software or hardware that creates and runs virtual machines. Popular hypervisors: VMWare (ESXi), AWS XEN, Microsoft Hyper V

Virtualization + cloud computing

- One of the important features of virtualization is that it allows sharing of applications to multiple customers and companies from a single machine.
- Virtualization in cloud computing can also be used to share the resources provided by the cloud environment like memory, platform and application.
- Since resources are available in cloud, the users utilizing virtualization can be in any location and use any compatible device, to avail the same service, which is not possible with virtualization done based on a server machine.
- With the help of virtualization in cloud computing, the customer can maximize the resources and reduces the physical system which is in need, which in turn increases productivity, efficiency and smooth delivery of software to clients.

Both Cloud computing & Virtualization are inter-related terms. But both are not same

Brief about Devops – Containerization



- The cost benefits and efficiency that **containerization** offers made it very compelling for IT organizations to adopt.
- Containerization is a lightweight alternative to a virtual machine that involves encapsulating an application in a container with its own operating system.
- A container takes its meaning from the logistics term, packaging container. When we refer to an application container, we mean packaging software.
- Containerization has recently gained hypes with an open source tool **Docker**.
- Docker containers are designed to run on every environment from physical computers to virtual machines, from bare-metal, Clouds, etc.

Brief about Devops – Containerization

The main benefits of using containers in ship are:

- Standardization on shape, size, volume and weight.
- The boxes that need to go to the same destination should stay together and loaded into the same shipping containers. Reduction in shipping costs and seamless movement across the sea, rail & road.

Containerization tools

HashiCorp Vagrant

docker

SHIP CONTAINER

DOCKER CONTAINER

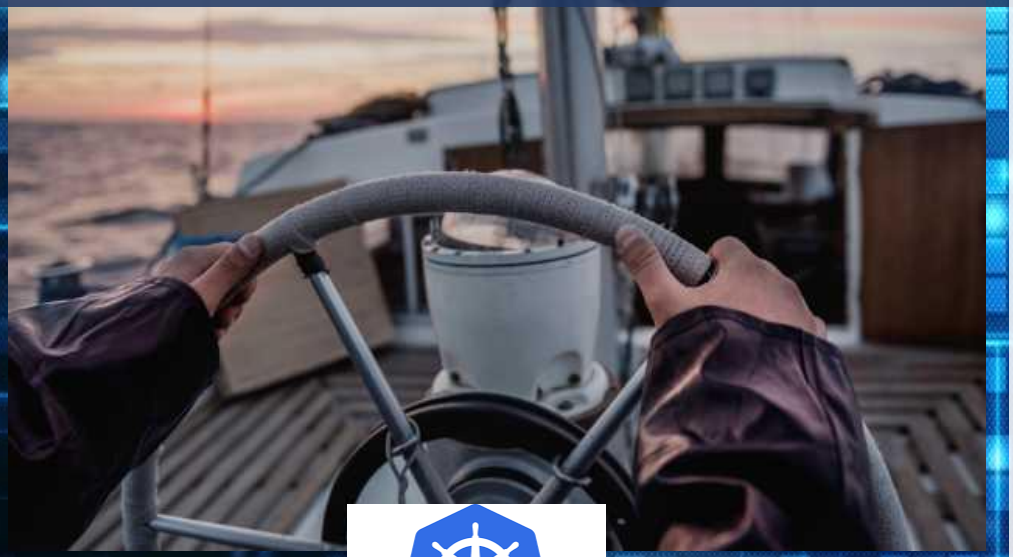
Brief about Devops – Containerization

Similarly, the main benefits of using containers in software are:

- Software containerization is encapsulating or packaging up software code, with related operating system, configuration files, libraries & all its dependencies, so that it can run uniformly and consistently on any infrastructure.
- Containerization allows applications to be "written once and run anywhere."
- With traditional methods, code is developed in a specific computing environment which, when transferred to a new location, often results in bugs and errors. But containerization offers **portability**, meaning that the application will run in the application developer's laptop, then to a test environment, then finally to a production environment without drastically changing the code.
- Hence, containerization ensure **consistency** & **standardization** by allowing the software to run reliably when moved from one computing environment to another. It also offers other benefits, like **fault isolation, ease of management and security.**
- Containers are **lightweight**, meaning they share the machine's operating system kernel and do not require the overhead of associating an operating system within each application. Containers are inherently smaller in capacity than a VM and require less start-up time, allowing more containers to run on the same compute capacity as a single VM. This drives higher server efficiencies and, in turn, reduces server and licensing costs.

Brief about Devops – Orchestration

- Implementing containers is not enough and its scalability is an operational challenge. A typical production implementation would require multiple containers, hosts, network configurations, load balancers, apps that need to be configured and ideally all these components should be deployed automatically. It requires planning, & most experts agree that an orchestration tool is a must.
- This is where an **orchestration** tool like **Kubernetes** comes in.
- Kubernetes is open source giving you the freedom to take advantage of on-premises, hybrid, or public cloud infrastructure, letting you **effortlessly move workloads** to where it matters to you.



kubernetes

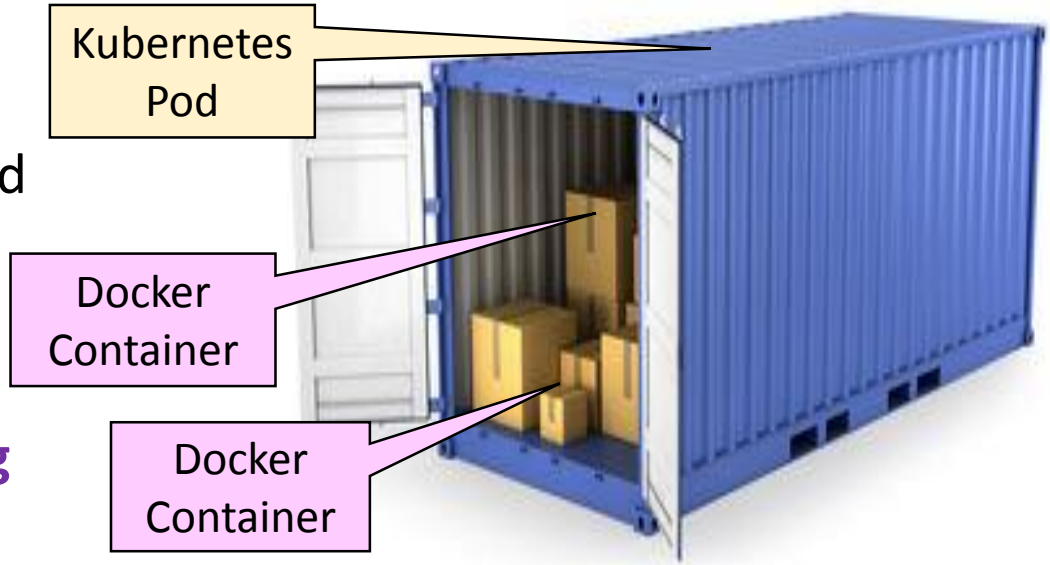
Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management.

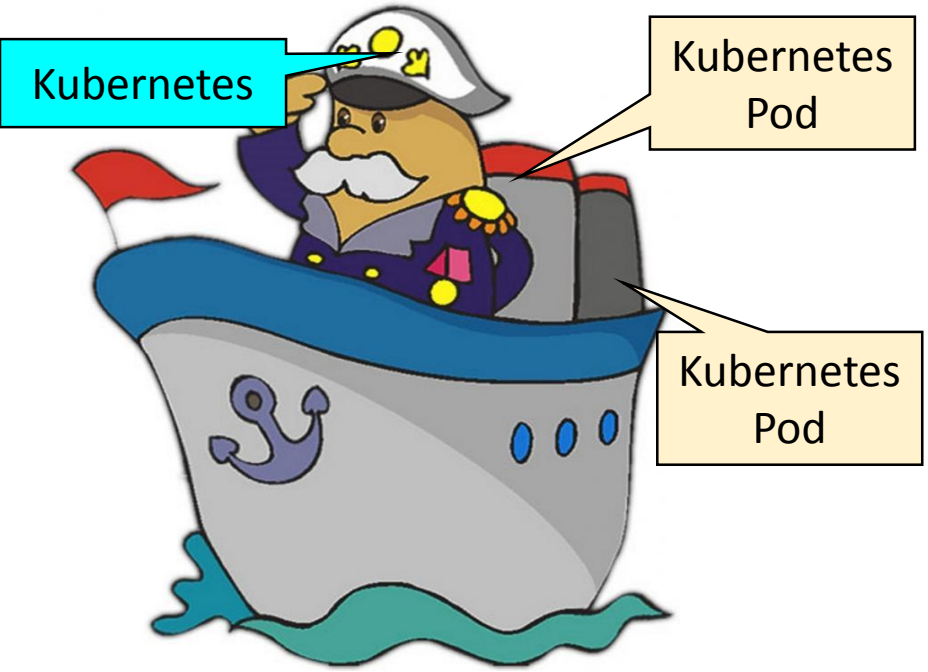
Kubernetes Objects

1. Pods
2. Services
3. Volumes
4. Namespaces
5. Secrets

Brief about Devops – Orchestration

Imagine Docker containers as packing boxes. The boxes that need to go to the same destination should stay together and loaded into the same shipping containers. In this analogy, the **packing boxes are Docker containers and the shipping containers are Kubernetes pods**.





All these Kubernetes pods make up the whole application. What you want for this cargo ship, is to arrive safely to its destination and avoid the stormy seas of the Internet. **Kubernetes**(Greek meaning is Captain or sailing master), **as the captain of the ship**, adeptly steers the ship along a smooth path which means ensuring all the pods under its supervision are managed effectively.

Brief about Devops – Infrastructure as code

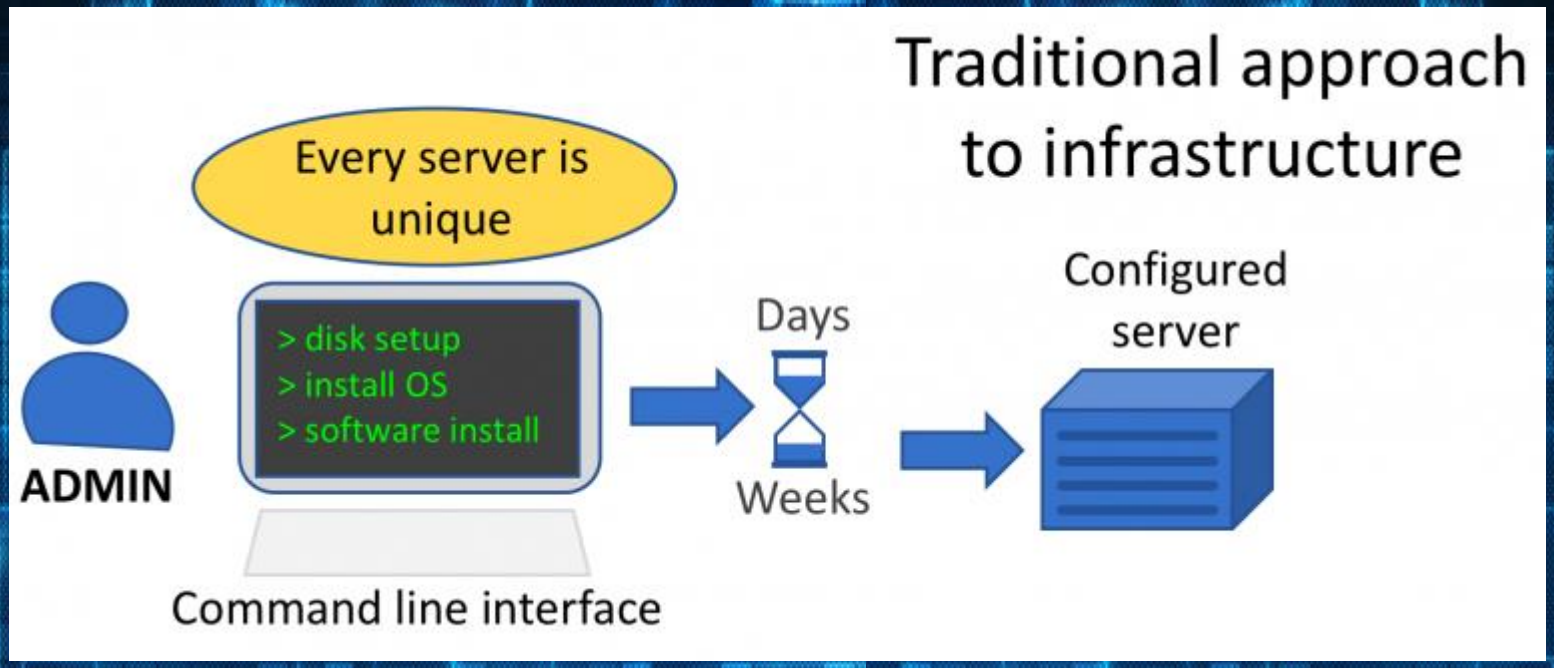
- **Infrastructure as code(IaC)**, is a method to provision and manage IT infrastructure through the use of source code, rather than through standard operating procedures & manual processes normally done by system /DB admin.
- Infrastructure as Code can simplify and accelerate your infrastructure provisioning process, help you avoid mistakes and comply with policies, keep your environments consistent, and save company a lot of time and money.
- Software engineers can be more productive on focus on higher-value tasks and can better serve customers.

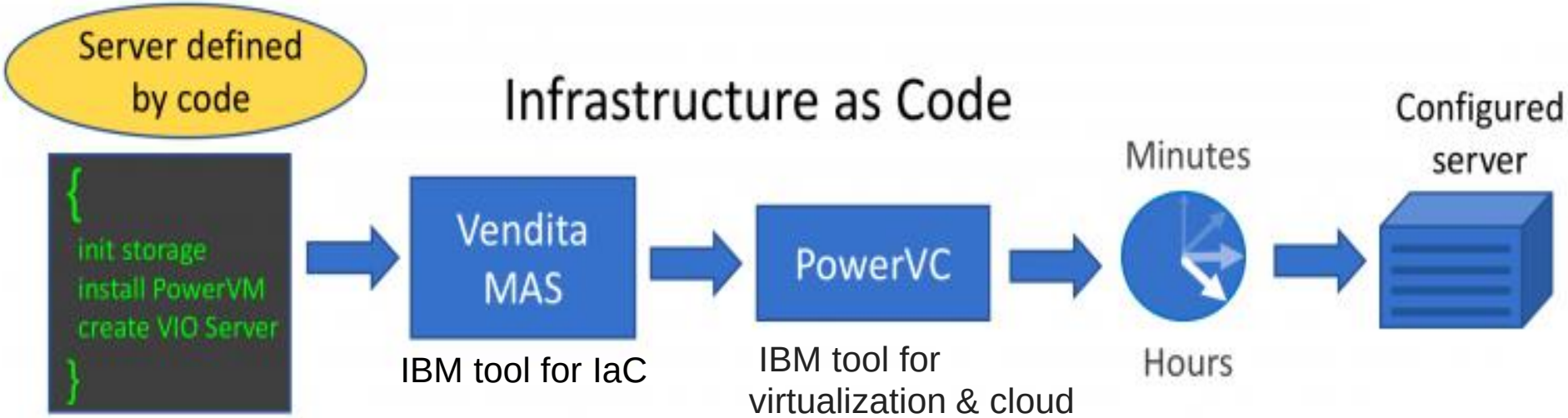
For example, w can install MySQL server, verify that MySQL is running properly, create a user account and password, set up a new database, and remove unneeded databases **through just software code, without an system admin** using infrastructure-as-code capabilities included with **Ansible**, an IT management and configuration tool.

Infrastructure as Code tools

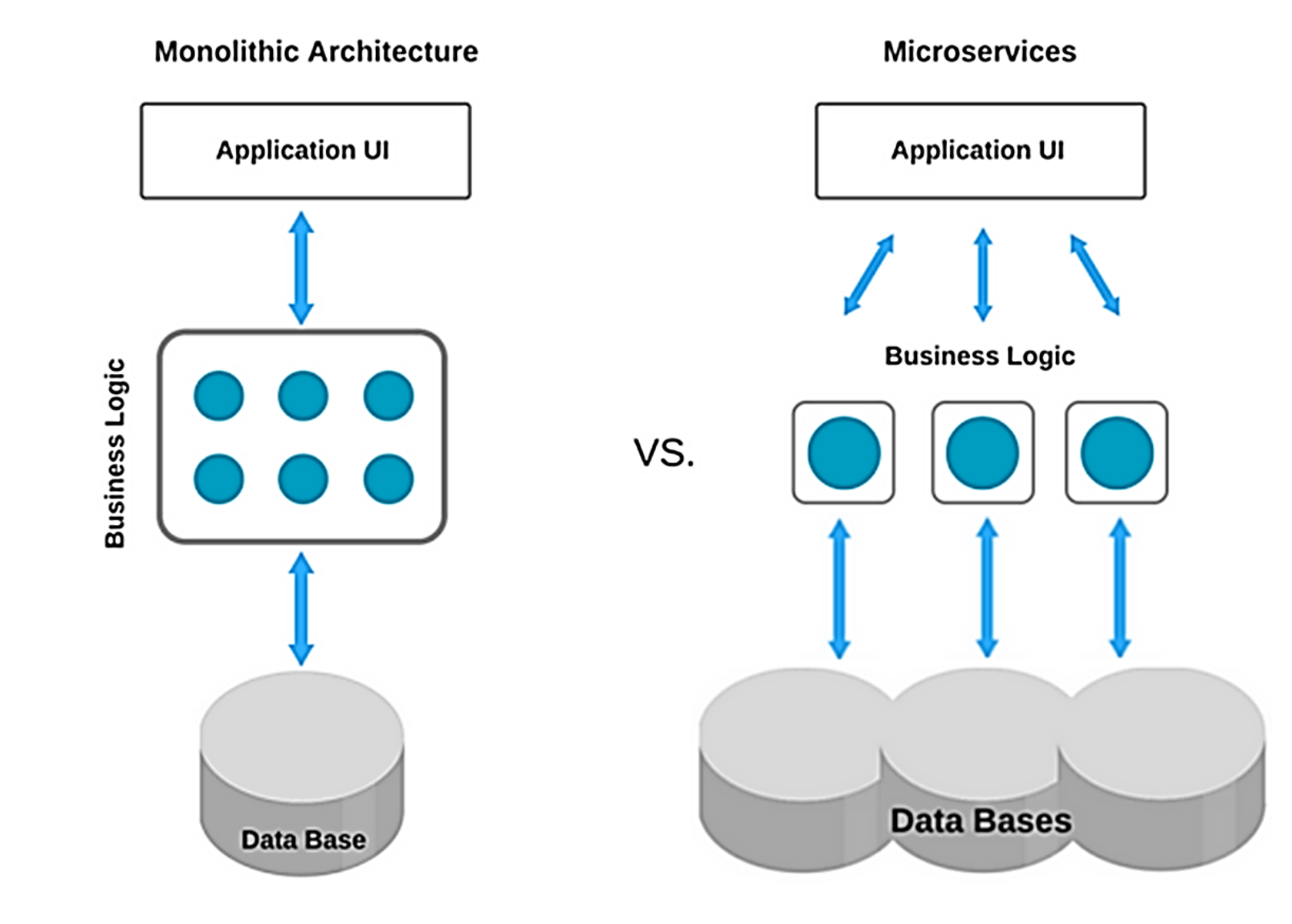
SALTSTACK ANSIBLE puppet labs CHEF

Brief about Devops – Infrastructure as code



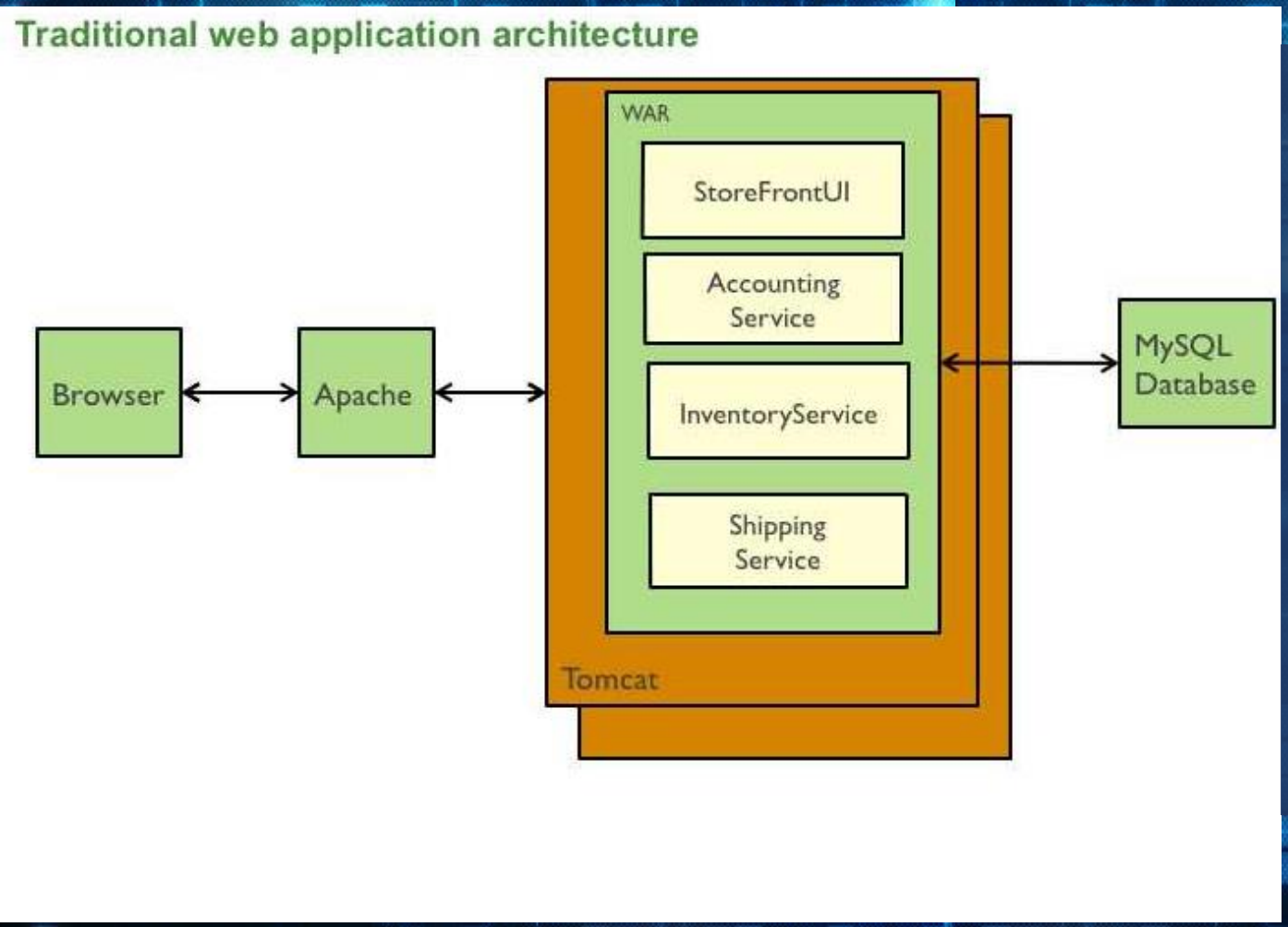


3. Brief about Microservices



Brief about Microservices – Monolithic Architecture

Suppose we are building an **e-commerce application** that takes orders from customers, verifies inventory and available credit, and ships them. It consists of several components including the StoreFrontUI, which implements the user interface, along with backend services for checking credit, maintaining inventory and shipping orders.



The application is deployed as a single monolithic (also called as legacy or traditional) application. For example, a Java web application consists of a **single WAR file** that runs on a web server like Tomcat.

Brief about Microservices – Monolithic Architecture

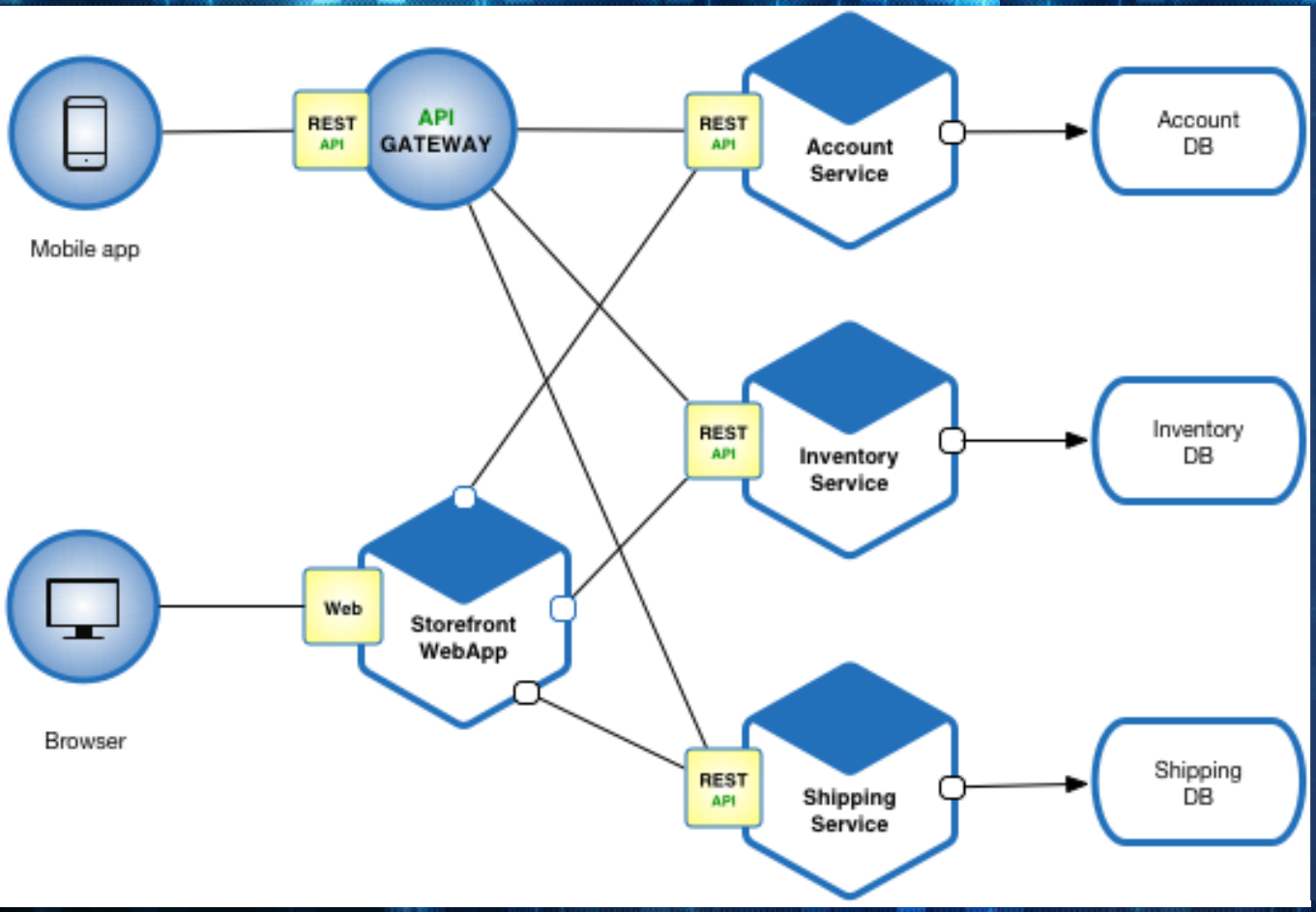
This solution has a number of benefits in the initial stage:

Simple to develop - the goal of current development tools and IDEs is to support the development of monolithic applications
Simple to deploy - you simply need to deploy the WAR file (or directory hierarchy) on the appropriate runtime
Simple to scale - you can scale the application by running multiple copies of the application behind a load balancer.

However, once the application becomes large and the team grows in size, this approach has a number of drawbacks that become increasingly significant:
The large monolithic code base can be difficult to understand and modify. As a result, development typically slows down.
Overloaded IDE - the larger the code base the slower the IDE and developers are less productive
Overloaded web container - the larger the application the longer it takes to start up.
Continuous deployment is difficult - a large monolithic application is also an obstacle to frequent deployments. In order to update one component you have to redeploy the entire application
Scaling the application can be difficult. Different application components have different resource requirements - one might be CPU intensive while another might memory intensive. With a monolithic architecture we cannot scale each component independently
Requires a long-term commitment to a technology stack which forces you to be married to the technology stack (and in some cases, to a particular version of that technology). In order to adopt a newer platform framework you have to rewrite the entire application

Brief about Microservices – Microservice Architecture

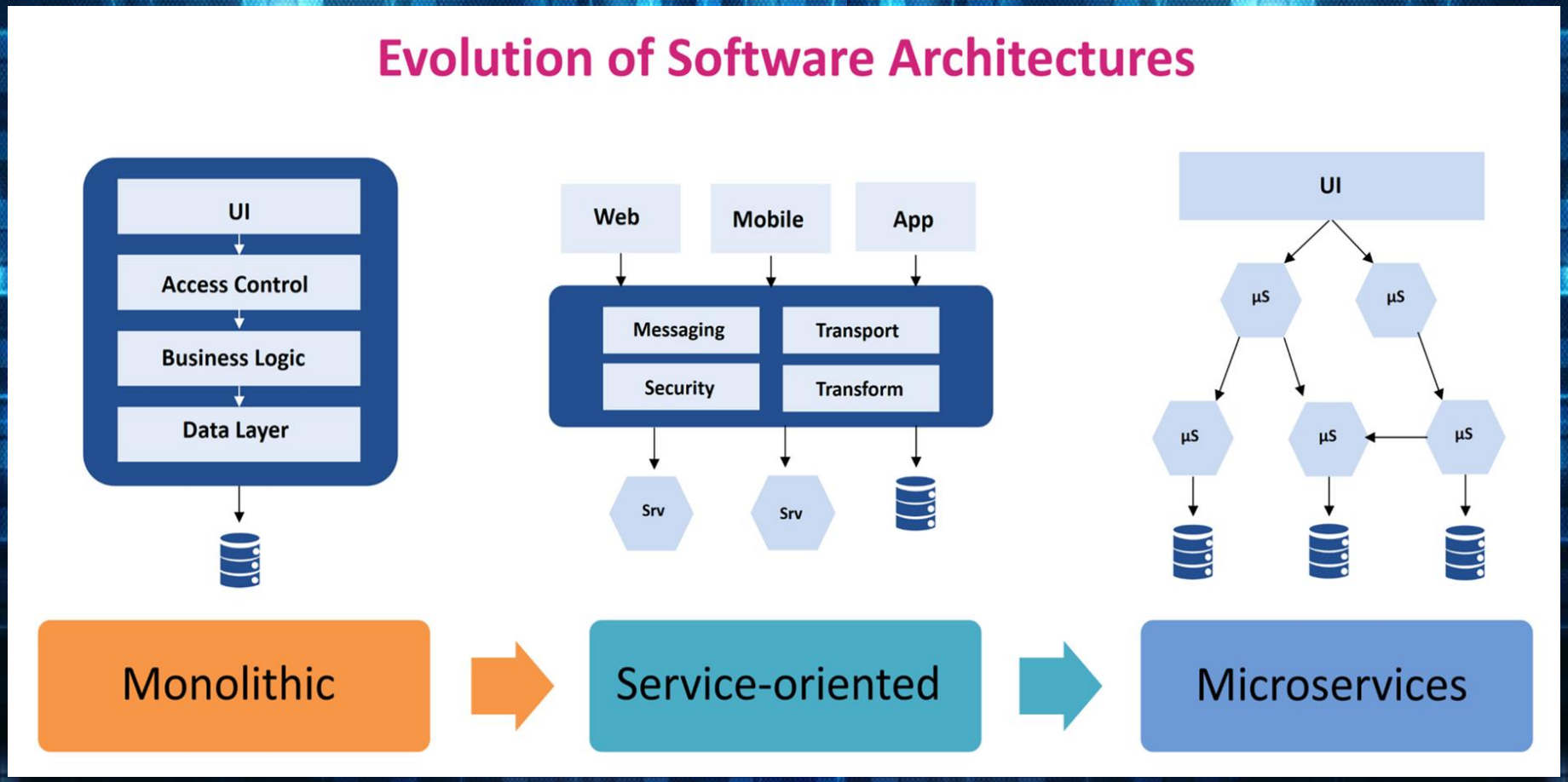
Suppose we are building an **e-commerce application** that takes orders from customers, verifies inventory and available credit, and ships them. It consists of several components including the StoreFrontUI, which implements the user interface, along with backend services for checking credit, maintaining inventory and shipping orders.



Services communicate using either synchronous protocols such as HTTP/REST or asynchronous protocols such as AMQP (Advanced Message Queuing Protocol). Services can be developed and deployed independently of one another. Each service has its own database.

Brief about Microservices – Microservices Architecture

Microservices are **independent entities** such that in the event of any failure of a particular entity, only that entity is affected and does not compromise the entire system. Microservices architecture enables companies to make deployments and **add new features much easier**. In order to increase runtime and efficient delivery, the companies need to shift to microservice architecture.



Brief about Microservices – Microservices + Containers

Containers are meant to act like a very lightweight encapsulation of any application (microservice / monolithic) with auto configuration system, so that when you install a container they run directly what's inside, without having to consider the local environment or OS.

With **Microservices**, a complex application is broken up into a series of smaller, more specialized services, each with its own database and its own business logic. Microservices then communicate with each other across common interfaces (like APIs) and REST interfaces (like HTTP). Using microservices, development teams can focus on updating specific areas of an application without impacting it as a whole, resulting in faster development, testing, and deployment.

Microservices + Containers

Microservices and containerization work well when used together. A microservice, developed within a container, then gains all of the inherent benefits of containerization—**portability** in terms of the development process and vendor **compatibility**, as well as developer **agility**, **fault isolation**, **server efficiencies**, **automation of installation**, **scaling** and **security**

References

Resources:
https://github.com/egovernments/egov-services
DIGIT:
https://www.digit.org
Ministry of housing & urban affairs :
http://104.215.151.14/
Open source e-government platforms :
https://openforge.gov.in/project/showdetails.php?group_id=110
Reports / Dashboard:
https://www.egovernments.org/html/apmpi/
Example of a eGov services provider:
http://psaitech.com/egov.html
---
https://www.egovernments.org/
https://societalplatform.org/
https://pppknowledgelab.org/sectors/e-government
https://egovstrategy.gov.ie/actions/
https://github.com/egovernments/egov-smartcity-suite
https://openforge.gov.in/project/showdetails.php?group_id=110
https://meity.gov.in/content/state-data-centre

eGov Awards (case study) References

- eGov bags the prestigious "Order of Merit" at Business World Smart Cities Conclave & Awards, 2016. "eGov Smart Platform" and “Puraseva” app across 110 ULBs in Andhra Pradesh. http://businessworld.in/article/BW-Businessworld-Presents-Smart-Cities-Conclave-And-Awards/22-12-2016-110193/
- Chennai Corporation gets best e-governance award for implementing eGov SmartCity http://m.timesofindia.com/city/chennai/Chennai-Corporation-gets-e-governance-award/articleshow/45550817.cms
- Nagpur Municipal Corporation gets Skoch Gold award 2014 for implementing eGov SmartCity http://timesofindia.indiatimes.com/city/nagpur/Nagpur-Municipal-Corporation-gets-award-for-e-governance/articleshow/43024144.cms
- National Award for eGovernance 2010-11 - Excellence in Government Process Re-engineering for implementing property tax information system, a product from eGov SmartCity http://darpg.gov.in/darpgwebsite_cms/document/file/award2011.pdf
- Skoch 2010 - The world is open award for implementing public grievance redressal system, a product from eGov SmartCity - http://www.thehindu.com/todays-paper/tp-national/tp-tamilnadu/corporation-receives-skoch-award/article778178.ece
- Citizen’s Choice category of eINDIA Awards 2010 - Municipal IT Initiative of the year awarded to eGov SmartCity - http://eindia.eletsonline.com/2010/award-winners/
- ICT Excellence 2010 Award for implementing three key products from the eGov SmartCity - http://www.mrc.gov.in/awards
- eIndia 2008 - Best Government Initiative of the Year Award for implementing Property Tax Information System, a product in the eGov SmartCity suite http://egov.eletsonline.com/2008/12/mcd-gets-the-best-e-governed-project

References

https://devops.com/
https://linoxide.com/linux-how-to/best-known-devops-tools/
https://www.iconagility.com/expertise/devops.php
https://www.information-age.com/ultimate-guide-devops-123482326/
https://tech.gsa.gov/guides/what_is_devops/
https://www.scalyr.com/blog/devopssec-challenges/
https://devops.com/doing-devops-in-an-interconnected-world/
https://www.atlassian.com/devops
https://techbeacon.com/devops/10-companies-killing-it-devops
https://www.bmc.com/blogs/devops-virtualization/
https://opensourceforu.com/2019/07/cloud-computing-and-devops-a-combination-that-can-transform-an-organisation/
https://www.ibm.com/cloud/learn/containerization
https://en.wikipedia.org/wiki/DevOps
https://www.udemy.com/topic/devops/
https://www.edureka.co/blog/top-10-devops-tools/
https://www.bitdegree.org/tutorials/devops-interview-questions/
https://www.guru99.com/devops-tutorial.html
http://www.techmarshals.com/best-devops-training-in-hyderabad/
https://intellipaat.com/blog/what-does-a-devops-engineer-do/

References

https://www.sumologic.com/blog/what-is-containerization/
https://jelastic.com/blog/containerization-devops/
https://www.linuxnix.com/what-is-dockerfile-and-how-to-create-docker-images/
https://www.linuxnix.com/what-is-containerization-in-devops/
https://apifriends.com/api-management/virtualization-containerization-devops/
https://apifriends.com/api-management/docker-and-kubernetes/
https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/
https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/
https://medium.com/google-cloud/kubernetes-101-pods-nodes-containers-and-clusters-c1509e409e16
https://medium.com/@ladislavGazo/building-docker-java-app-images-with-maven-bdd88305abb
https://stackoverflow.com/questions/21498832/in-docker-whats-the-difference-between-a-container-and-an-image
https://mindmajix.com/devops-interview-questions
https://kubernetes.io/case-studies/nokia/
https://www.quora.com/What-is-DevOps-What-are-the-duties-of-a-DevOps-Engineer
https://www.quora.com/What-are-the-best-tools-to-learn-DevOps-How-to-start-DevOps
https://www.linkedin.com/pulse/7-key-benefits-devops-certification-training-deepak-kumar?trk=mp-reader-card

Devops :

References

https://devops.com/devops-is-old-news-devprod-is-the-future/
https://success.docker.com/architectures
https://myopswork.com/maven-as-a-devops-tool-f5d04b0fc55c?gi=f593d5ea524e
https://www.reddit.com/r/devops/comments/cowe8e/is_it_important_to_learn_itil_for_a_devops_role/
https://hackernoon.com/the-roadmap-to-become-a-devops-dude-from-server-to-serverless-dd97420f640e

Microservices : References

https://microservices.io/
https://www.nginx.com/learn/microservices/
https://mherman.org/blog/developing-microservices-node-react-docker/
https://crunchytechbytz.wordpress.com/2018/02/09/microservice-with-apache-kafka-netflix-oss/
https://www.infoq.com/presentations/monolith-microservices-refactoring-analysis-tools/
https://www.smartspate.com/microservice-front-end/
https://devops.com/devops-com-containerization-microservices-survival-guide/
https://www.urolime.com/blogs/devops-in-2019-part-ii/
https://cloudplatform.googleblog.com/2015/01/what-makes-a-container-cluster.html?m=1
https://www.theregister.co.uk/2018/12/06/azurre_kubernetes_acs_axe/
https://jelastic.com/blog/containerization-devops/
https://docs.microsoft.com/en-us/dotnet/architecture/containerized-lifecycle/docker-application-lifecycle/containers-foundation-for-devops-collaboration
https://apifriends.com/api-management/microservices-governance/
https://www.capgemini.com/2019/06/practical-learnings-on-microservices/

References

https://www.techwalls.com/register-free-google-apps-standard-account-single-user/
http://appengine.google.com/
https://blogs.technet.microsoft.com/yungchou/2014/05/19/devops-automation-and-cloud-computing/

YET TO EXPLORE MORE…