## DevOps interview questions:

## <https://www.udemy.com/devops-interview-questions-preparation-course/?couponCode=KPOWER10>

Projects:...walk through layesr, what you warking , automation in cloud, deployment scripts, collaborating with teams, scrum methodology,

Team size

Role.. devops role,

Programming :

Dev side ops side

How quickly you can learn

Architect in application

Given the chance the for team lead

Linux:...command to view crontab...CRONTAB

ALIAs: shortcuts

Ssh port forwarding...

What is zombie process—terminate state, child is there

How to check CPU consuming process: TOP

## Deployment

what is blue green deployment..:

Blue-green deployment is a technique that reduces downtime and risk by running two identical production environments called Blue and Green.

At any time, only one of the environments is live, with the live environment serving all production traffic. For this example, Blue is currently live and Green is idle.

As you prepare a new version of your software, deployment and the final stage of testing takes place in the environment that is *not* live: in this example, Green. Once you have deployed and fully tested the software in Green, you switch the router so all incoming requests now go to Green instead of Blue. Green is now live, and Blue is idle.

This technique can eliminate downtime due to application deployment. In addition, blue-green deployment reduces risk: if something unexpected happens with your new version on Green, you can immediately roll back to the last version by switching back to Blue.

Hot deployment: **What is Hot Deployment ?**

Hot-deployable services are those which can be added to or removed from the running server. It is the ability to change ON-THE-FLY what’s currently deployed without redeploying it.

Hot deployment is VERY hot for development. The time savings realized when your developers can simply run their build and have the new code auto-deploy instead of build, shutdown, startup is massive.

**What is cold Deployment ?**

Cold deployment are defined as tasks that require one or more instances of the server to restart to reflect the changes.

## DevOPS

1. What is devops :

Devops is group of practices focused on automating the software delivery process and improve collaboration between development and operations team to work towards the main goal build test release reliable software faster

Faster release and high quality software,

Devops practices: version control, CI CD, infratsutre as code

1. What is the need for devops,

**A Common Pre-DevOps Scenario**  
The Dev team that has a goal to ship as many features as possible, kicks a new release “over the wall” to QA. Then the tester’s goal is to find as many bugs as possible. When the testers bring their findings to Dev, the developers become defensive and blame the testers that are testing the environment for the bugs. The testers respond that it isn’t their testing environment, but the developer’s code that is the problem.

Eventually the issues get worked out and QA kicks the debugged new release “over the wall” to Ops. The Ops team’s goal is to limit changes to their system, but they apprehensively release the code and the system crashes. The finger pointing resumes.  
  
Ops says that Dev provided them faulty artifacts. Dev says everything worked fine in the test environment. The fire drills begin to debug the system and get production stable. The production environment isn’t Dev’s and QA’s responsibility, so they keep hands off while Ops spends all night fixing the production issues.

Devops addresses he below challenges:

Conflicts between the team working on various phases of development as they will have separate goals..devops makes these team to collaborate and involved in each phase of the software developemt.

Each day new code is deployed as the developers complete it. Automated testing ensures the code is ready to be deployed. After the code passes all the automated testing it is deployed to a small number of users. The new code is monitored for a short period to ensure there are no unforeseen problems and it is stable. The new code is then proliferated to the remaining users once the monitoring shows that it is stable. Many, if not all, of the steps after planning and development are done with no human intervention..

1. Devops reduces the deployment time for developers , faster testing sppedy feedback
2. Tools to automate the infrattutre managemnt
3. It makes sure that the environment si avaialable , environment and code tested together
4. Bring developers to closr to the cutomer experience
5. Create resuabale deployment procedure

Increased deployment frequncey

Lower failure rate

Shorten lead time between fixes..continuous testing makes it possible to detect defects earlier.

1. benefits of devops---Technical>continuous software delivery,improved defect detetction,shorter development cycle,rediced deployment failures, shorten lead time to fixes,

Business: faster delivery of features, more time available to add value

1. What is agile

Agile is a software development methodology which is based on iterative developments that is a workable product is delivered in 1 to 4 weeks iterations to get the frequent feedback to align with the changing business needs.

Main features of agile includes... the cutomer statisfaction, welcome changes, deliver a working software frequntely, collaboration between business people and develpers

Development tasks are broken down into smaller tasks and are planned to be completed in short iteration of 1 to 4 weeks .Each iteration includes planning requirement analysis, design coding testing.

Most popular agile methodologies includes below:

**Scrum**: it’s a well-defined process set of rules and responsibilities ..to structuring the software development into small iterations. Big and complex tasks are split into small user stories to be completed within a sprint. That approach includes sprint planning before the start of each sprint, stand-up meetings to track the progress of the stories, sprint reviews to show ta demo to get the feedback from the business. Below are the roles includes in scrum:

Product owner...mnages the product logs, splits the features into stories and set the priorties f the tasks

Scrum master...managesthe scrum board helps the team to foloow scrum frammewprk conduts the eeting standups

Development team...completes the work

The metrics and measurement of work is done through story points to measure how much work a scrum team can do in a sprint

**Kanban**: kanban is non structured way of developing software its like a workflow . its not time based and poduct can be released whenverit ready it is tracked by current stage in the process by manting a to dolist

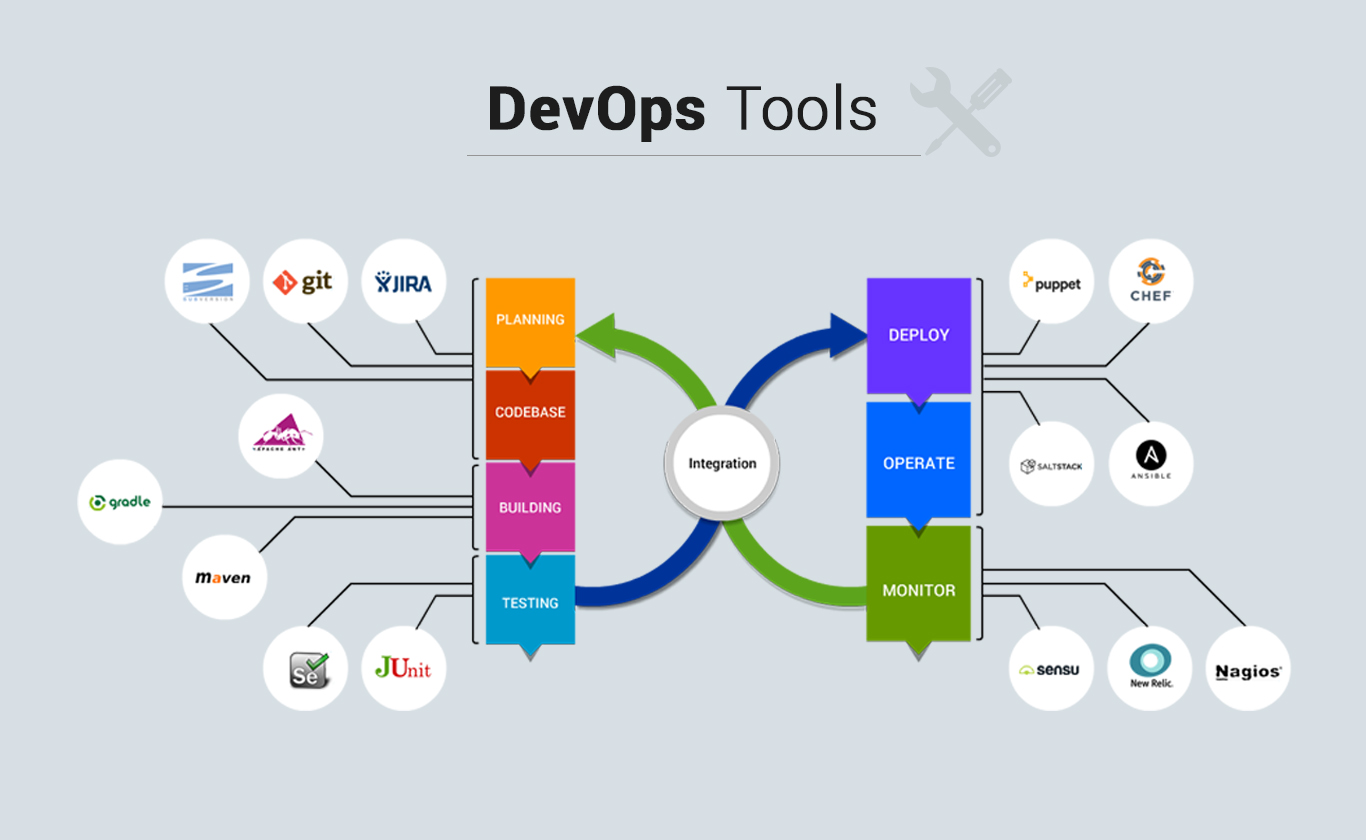
<https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>

1. Difference between agile and devops

Agile fills gap between developer and customer where as devops fills gap between developers and opes team .devops helps in implementing agility on the siftware develoepmnt .

| **Parameter** | **Agile** | **DevOps** |
| --- | --- | --- |
| What is it? | Agile refers to an iterative approach which focuses on collaboration, customer feedback, and small, rapid releases. | DevOps is considered a practice of bringing development and operations teams together. |
| Purpose | Agile helps to manage complex projects. | DevOps central concept is to manage end-to-end engineering processes. |
| Task | Agile process focusses on constant changes. | DevOps focuses on constant testing and delivery. |
| Implementation | Agile method can be implemented within a range of tactical frameworks like a sprint, safe and scrum. | The primary goal of DevOps is to focus on collaboration, so it doesn't have any commonly accepted framework. |
| Team skill set | Agile development emphasizes training all team members to have a wide variety of similar and equal skills. | DevOps divides and spreads the skill set between the development and operation teams. |
| Team size | Small Team is at the core of Agile. As smaller is the team, the fewer people on it, the faster they can move. | Relatively larger team size as it involves all the stack holders. |
| Duration | Agile development is managed in units of "sprints." This time is much less than a month for each sprint. | DevOps strives for deadlines and benchmarks with major releases. The ideal goal is to deliver code to production DAILY or every few hours. |
| Feedback | Feedback is given by the customer. | Feedback comes from the internal team. |
| Target Areas | Software Development | End-to-end business solution and fast delivery. |
| Shift-Left Principles | Leverage shift-left | Leverage both shifts left and right. |
| Emphasis | Agile emphasizes on software development methodology for developing software. When the software is developed and released, the agile team will not care what happens to it. | DevOps is all about taking software which is ready for release and deploying it in a reliable and secure manner. |
| Cross-functional | Any team member should be able to do what's required for the progress of the project. Also, when each team member can perform every job, it increases understanding and bonding between them. | In DevOps, development teams and operational teams are separate. So, communication is quite complex. |
| Communication | Scrum is most common methods of implementing Agile software development. Daily scrum meeting is carried out. | DevOps communications involve specs and design documents. It's essential for the operational team to fully understand the software release and its hardware/network implications for adequately running the deployment process. |
| Documentation | Agile method is to give priority to the working system over complete documentation. It is ideal when you're flexible and responsive. However, it can hurt when you're trying to turn things over to another team for deployment. | In the DevOps, process documentation is foremost because it will send the software to the operational team for deployment. Automation minimizes the impact of insufficient documentation. However, in the development of complex software, it's difficult to transfer all the knowledge required. |
| Automation | Agile doesn't emphasize on automation. Though it helps. | Automation is the primary goal of DevOps. It works on the principle to maximize efficiency when deploying software. |
| Goal | It addresses the gap between customer need and development & testing teams. | It addresses the gap between development + testing and Ops. |
| Focus | It focuses on functional and non-function readiness. | It focuses more on operational and business readiness. |
| Importance | Developing software is inherent to Agile. | Developing, testing and implementation all are equally important. |
| Speed vs. Risk | Teams using Agile support rapid change, and a robust application structure. | In the DevOps method, the teams must make sure that the changes which are made to the architecture never develop a risk to the entire project. |
| Quality | Agile produces better applications suites with the desired requirements. It can easily adapt according to the changes made on time, during the project life. | DevOps, along with automation and early bug removal, contributes to creating better quality. Developers need to follow Coding and Architectural best practices to maintain quality standards. |
| Tools used | JIRA, Bugzilla, Kanboard are some popular Agile tools. | Puppet, Chef, TeamCity OpenStack, AWS are popular DevOps tools. |
| Challenges | The agile method needs teams to be more productive which is difficult to match every time. | DevOps process needs to development, testing and production environments to streamline work. |
| Advantage | Agile offers shorter development cycle and improved defect detection. | DevOps supports Agile's release cycle. |

1. Devops toolchain



1. Experience building brdges between IT OPS QA and developers

Devops is about communication and colalbaration between ops and develoeprs.

I have brideged the gap between op and devoplers by implementing by creating self- service code continuous deployment process through Jenkins, in obiee odi daar project and in other project I supported issues from development as well ops side in Environments . like ofsaa and deployemnts like.and fixed the dirrences between production and testing environments. I successfully automated processes to support continuous software deployments.

Agile only deals with development ,

Where devops , starting from development, test and release

1. Agile methodologies

* Scrum:--focused on mamaging the tasks , it consists of set of roles and responsibilies with in a team , defines a devleopemnt actvties into repeatable SPRINTS, framework and process are already defined
* Crystal :..based on three concepts
  + **Chartering:** Various activities involved in this phase are creating a development team, performing a preliminary feasibility analysis, developing an initial plan and fine-tuning the development methodology
  + **Cyclic delivery:** The main development phase consists of two or more delivery cycles, during which the
    - Team updates and refines the release plan
    - Implements a subset of the requirements through one or more program test integrate iterations
    - Integrated product is delivered to real users
    - Review of the project plan and adopted development methodology
  + **Wrap Up:** The activities performed in this phase are deployment into the user environment, post- deployment reviews and reflections are performed.
* DSDM:dynamic software development model:---its rapid applcatoion development approach..users are required to be involved actvitly :

1. Time Boxing
2. MoSCoW Rules
3. Prototyping

The DSDM project consists of 7 phases

1. Pre-project
2. Feasibility Study
3. Business Study
4. Functional Model Iteration
5. Design and build Iteration
6. Implementation
7. Post-project

* FDD: feature driven development:---designing and building feature, very specifc and sghort phases of work per feature
  + - Domain object Modeling
    - Development by feature
    - Component/ Class Ownership
    - Feature Teams
    - Inspections
    - Configuration Management
    - Regular Builds
    - Visibility of progress and results

Lean software development--- Lean software development method is based on the principle "Just in time production". It aims at increasing speed of software development and decreasing cost. Lean development can be summarized in seven steps.

1. Eliminating Waste
2. Amplifying learning
3. Defer commitment (deciding as late as possible)
4. Early delivery
5. Empowering the team
6. Building Integrity
7. Optimize the whole

* Extreme programming---it aims to produce higer quality software, introducing g dynamically changing requirements ..phases of xp:

|  |  |
| --- | --- |
| Planning | * Identification of stakeholders and sponsors * Infrastructure Requirements * [Security](https://www.guru99.com/ethical-hacking-tutorials.html) related information and gathering * Service Level Agreements and its conditions |
| Anlysis | * Capturing of Stories in Parking lot * Prioritize stories in Parking lot * Scrubbing of stories for estimation * Define Iteration SPAN(Time) * Resource planning for both Development and QA teams |
| Design | * Break down of tasks * Test Scenario preparation for each task * Regression Automation Framework |
| Execution | * Coding * Unit Testing * Execution of Manual test scenarios * Defect Report generation * Conversion of Manual to Automation regression test cases * Mid Iteration review * End of Iteration review |
| Wrapping | * Small Releases * Regression Testing * Demos and reviews * Develop new stories based on the need * Process Improvements based on end of iteration review comments |
| Closure | * Pilot Launch * Training * Production Launch * SLA Guarantee assurance * Review SOA strategy * Production Support |

Bords: story cardboard, oline storyboard

1. Scrum and kanban diffrences:

|  |  |
| --- | --- |
| **Scrum** | **Kanban** |
| * In scrum technique, test must be broken down so that they can be completed within one sprint | * No particular item size is prescribed |
| * Prescribes a prioritized product backlog | * Prioritization is optional |
| * Scrum team commits to a particular amount of work for the iteration | * Commitment is optional |
| * Burndown chart is prescribed | * No particular item size is prescribed |
| * Between each sprint, a scrum board is reset | * A Kanban board is persistent. It limits the number of items in workflow state |
| * It cannot add items to ongoing iteration | * It can add items whenever capacity is available |
| * WIP limited indirectly | * WIP limited directly |
| * Timeboxed iterations prescribed | * Timeboxed iterations optional |

1. Differences between agile andsdlc:

|  |  |
| --- | --- |
| **Agile Model** | **Waterfall Model** |
| * Agile method proposes incremental and iterative approach to software design | * Development of the software flows sequentially from start point to end point. |
| * The **agile process** is broken into individual models that designers work on | * The design process is not broken into an individual models |
| * The customer has early and frequent opportunities to look at the product and make decision and changes to the project | * The customer can only see the product at the end of the project |
| * Agile model is considered unstructured compared to the waterfall model | * Waterfall model are more secure because they are so plan oriented |
| * Small projects can be implemented very quickly. For large projects, it is difficult to estimate the development time. | * All sorts of project can be estimated and completed. |
| * Error can be fixed in the middle of the project. | * Only at the end, the whole product is tested. If the requirement error is found or any changes have to be made, the project has to start from the beginning |
| * Development process is iterative, and the project is executed in short (2-4) weeks iterations. Planning is very less. | * The development process is phased, and the phase is much bigger than iteration. Every phase ends with the detailed description of the next phase. |
| * Documentation attends less priority than software development | * Documentation is a top priority and can even use for training staff and upgrade the software with another team |
| * Every iteration has its own testing phase. It allows implementing regression testing every time new functions or logic are released. | * Only after the development phase, the testing phase is executed because separate parts are not fully functional. |
| * In agile testing when an iteration end, shippable features of the product is delivered to the customer. New features are usable right after shipment. It is useful when you have good contact with customers. | * All features developed are delivered at once after the long implementation phase. |
| * Testers and developers work together | * Testers work separately from developers |
| * At the end of every sprint, user acceptance is performed | * User acceptance is **performed** at the end of the project. |
| * It requires close communication with developers and together analyze requirements and planning | * Developer does not involve in requirement and planning process. Usually, time delays between tests and coding |

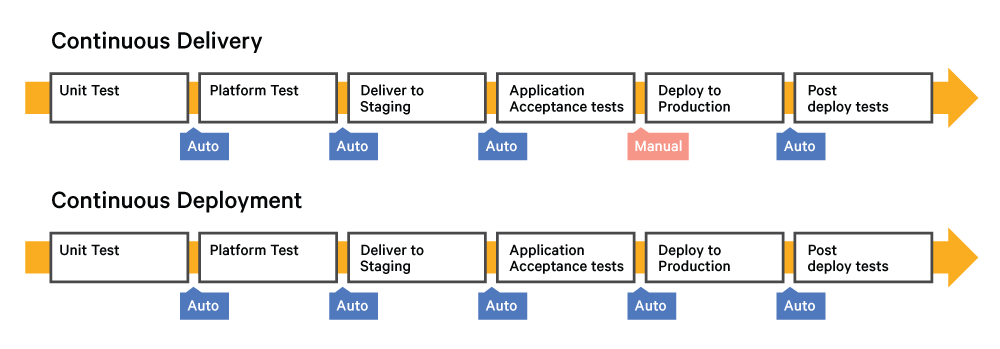
1. Devops core operations:

Developement: code develop, code coverage,unit test,packaging

Infra: provisioning,configuration,orchestration,Deployment

<https://www.atlassian.com/blog/devops/the-4-phases-of-devops>

1. Continuous delivey : devops practice where code is buil tested and deployed to QA enviromnt and made ready for prodytion deployment . it is tested in production like environment .



1. Continuous integration:

CI a process to monitor the code changes, when evr code change is detected a automated build and test is triggered Can aslo include code quality, health, provides faster feedback.

Every commit should be immediately compiled integrated the code to existing system or other developers code and tested and feedback should be given to the committer

Best practice

* To maintain a proper code repository
* 2. To automate the build process of the code
* Ensuring code check-ins on a daily basis as against batch commits
* Ensure that the commit made should always be buildable
* Results of the build should be transparent, made available to developers so as to keep them aware of the software quality of the product on a daily basis
* Testing the build on a Production like environment always
* A build broken by a commit should be fixed immediately
* Latest build should be available to everyone from the main stream
* To automate the deployment process

1. : What is scrum

it’s a well-defined process set of rules and responsibilities ..to structuring the software development into small iterations. Big and complex tasks are split into small user stories to be completed within a sprint. That approach includes sprint planning before the start of each sprint, stand-up meetings to track the progress of the stories, sprint reviews to show ta demo to get the feedback from the business. Below are the roles includes in scrum:

* Each iteration of a scrum is known as Sprint
* Product backlog is a list where all details are entered to get end product
* During each Sprint, top items of Product backlog are selected and turned into Sprint backlog
* Team works on the defined sprint backlog
* Team checks for the daily work
* At the end of the sprint, team delivers product functionality

1. Devops engineer responsibility with respect to agile development

* Understand the needs and challenges of a client across operations and development, and partner to formulate solutions that support their business and technical strategies and goals
* Develop solutions encompassing technology, process and people for:
* Continuous Delivery
* Infrastructure strategy & operations (including cloud)
* Build and release management
* Basic understanding of Networking
* Security (fair understanding of application and infrastructure security)
* Recommend and Implement solutions. Be totally hands on and have the ability to work independently
* Ensure delivery of exceptional technical solutions
* Maintain strong expertise and knowledge of current and emerging processes, techniques and tools
* Build the DevOps practice within ThoughtWorks and drive our thought-leadership externally
* Identifies and resolves problems in a timely manner
* Design, build and maintain the CI/CD infrastructure and tools to deliver Horizon Cloud Service
* Work closely with development teams to ensure that solutions are designed with customer user experience, scale/performance and operability in mind

1. Exaplain a use case for devops :

Can describe ..obi and odi use case deployment ..

Can describe rome ci pipeline

Can describe gdw env creation

Edw refresh and build

Ci pipleline for a java web application …Take an example from infy docker course ,devops course

1. What are tools you are familiar with that can help in role

GIT

JENKINS

SELENIUM

DOCKER

PUPPET

CHEF

ANSIBLE

NAGIOS

ELK STACK

SPLUNK

1. Descibe linux env

The Linux Operating System’s architecture primarily has these components: the Kernel, Hardware layer, System library, Shell and System utility.

The kernel is the core part of the operating system, which is  responsible for all the major activities of the LINUX operating system.

<https://www.tutorialspoint.com/operating_system/os_linux.htm>

1. Descibe your expericne implementing cd

Explain ci/cd for gdw env build .

Explain ci for obiee code review…

1. Greatest achievement on a project ..

Explain ci for obiee code review…

1. What have you been doing over the last 1-2 years?
2. How do you deploy software?
3. How have you handled failed deployments?---used automated tvts to check if deployments are successful
4. If something breaks in production, how do you know about it?
5. Most challenging situation and how do you fix it
6. Vagrant

## Testing :

<http://webappsuccess.com/testing-and-deployment.html>

1. Different types of testing needs to be done in software cycle

<https://www.geeksforgeeks.org/types-software-testing/>

1. What is sanity check

<https://www.guru99.com/smoke-sanity-testing.html>

1. What is continuous testing

Continuous testing generally refers to the process of executing automated tests throughout the delivery pipeline, thereby giving developers the earliest possible feedback about the quality of their code, and managers timely information about the risks associated with deploying that code to customers.

One can, and I think one should, also take continuous testing to mean all of the measures that an organization takes to build testing into the development and delivery process, from specification to design to coding to integration to production. Contrast this with the old-fashioned notion of a “testing phase” that occurs when development is done, after extensive test plans have been written.

While continuous testing is often taken to be coterminous with test automation, the more expansive definition includes activities that cannot be automated but can be “shifted left,” such as the integration of testability into the design process or the earliest possible acquisition of user or customer feedback.

1. How to automate testing in Devopslifecycle

<https://www.capgemini.com/2017/02/automated-test-environments-for-devops/>

Include testing in continuous delivery pipeline ..deploying code to pre prod or staging environments and provide continuous feedback for the developed code

* Provides fast and continuous feedback
* Excludes errors at early stages
* Enhances the checking process efficiency
* Accelerates the rate of change
* Allows flawless continuity throughout SDLC (system development life cycle)
* Allows developers to bring new features to market faster
* Ensures safety of the system
* Allows managers to make better trade-off decisions

1. Key elements continuous testing …

<https://devops.com/continuous-testing-what-exactly-is-it/>

1. Testing tool familiar with

Junit selenium testng

1. Functional testing

Functional testing is a type of testing which verifies that each **function** of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application.

Each and every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results. This testing involves checking of User Interface, APIs, Database, security, client/ server applications and functionality of the Application Under Test. The testing can be done either manually or using automation

<https://www.guru99.com/functional-testing.html>

1. Code coverage tools

<https://dzone.com/articles/an-introduction-to-code-coverage>

how much of the source code is tested

<https://confluence.atlassian.com/clover/comparison-of-code-coverage-tools-681706101.html>

1. Code review tools

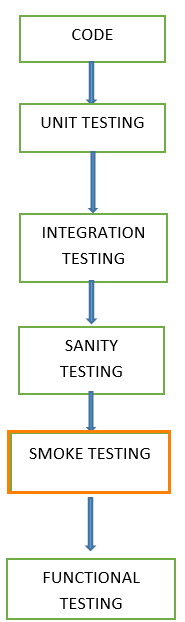
**Code Review** is nothing but testing the Source Code. Generally, it is used to find out the bugs at early stages of the development of software.

With this code review, the quality of the software gets improved and the bugs/errors in the program code decrease.

<https://www.softwaretestinghelp.com/code-review-tools/>

1. Regression testing : previously developed and tested code is working perfectly after the new change. Old testcases are run against the new version of the software .
2. Smoke test:

Smoke testing is a build verification testing to check if the build id stable and cab be tested further.these are minimal test to check the application build

[](https://cdn.guru99.com/images/3-2016/032816_1308_SmokeTestin1.png)

* All the show stoppers in the build will get identified by performing smoke testing.
* Smoke testing is done after the build is released to QA. With the help of smoke testing, most of the defects are identified at initial stages of software development.
* With smoke testing, we simplify the detection and correction of major defects.
* By smoke testing, QA team can find defects to the application functionality that may have surfaced by the new code.
* Smoke testing finds the major severity defects.

1. Testing tools

Configure selenium with jenkins.

## Config management:

* 1. Configuration management:

is a process for maintaining the configuration of systems throughout their life cycle. A system under configuration management control can have deployed artifacts, configuration files, system packages, user configurations, and services, all defined in source control. This ensures a repeatable process, which is the foundation of automation.

Configuration management has the ability to manage infrastructure deployments through the design, implementation, testing, building, release, and maintenance phases.

Configuration identification: information of assets and product, how many cpu, memory, package installed,..CMDB

Puppet an dhcef can fetch the iformation automically

Change mngemnt: control chghanges a product and its config doc

Config status accounting: provide status and info about the product and its config

Confg audit: verify config status

Use vcs with cms, give ability to rolback and rebuild the system

1. What is cm and benafits

There are several components in a configuration management system. Managed systems can include servers, storage, networking, and software. These are the targets of the configuration management system. The goal is to maintain these systems in known, determined states. Another aspect of a configuration management system is the description of the desired state for the systems. The third major aspect of a configuration management system is automation software, which is responsible for making sure that the target systems and software are maintained in the desired state.

Configuration management is important because it enables the ability to scale infrastructure and software systems without having to correspondingly scale administrative staff to manage those systems. This can make it possible to scale where it previously wasn’t feasible to do so.

Benefits:

The primary benefit of configuration management is consistency of systems and software. With configuration management, you no longer must guess or hope that a configuration is current. It is correct because the configuration management system makes sure that it is correct.

When combined with [automation](https://www.netapp.com/us/solutions/it-automation/index.aspx), configuration management can improve efficiency because manual configuration processes are replaced with automated processes. This also makes it possible to manage more targets with the same or even fewer resources.

1. Diffence between asset management and cm

Asset managemnt: T asset management maintains and develops standards, policies, processes, measurements and systems which enable the organization to properly manage its IT assets with respect to risk, control, governance, costs and business compliance and the performance objectives that have been set by the organization.concerned with finances, licensing

IT asset management is an integral part of an organization’s IT strategy, as it involves intensive data gathering of detailed software and hardware inventory information. This data is then used for informed decision making regarding future software and hardware redistribution and procurement. IT asset management enables organizations to manage their IT resources more effectively, saving money and time by having the ability to distinguish unnecessary purchases and the knowledge to leverage existing resources. This also helps in the minimization of risks that are related to the costs of advancing IT infrastructure portfolio projects based on old and incomplete or inaccurate information.

Concerned with deployemnts and operations, life cycle from deploy to retirement , ITIL based

1. What is IaC, purpose and where it stands in devops life cycle

Treating infrasturture as a code like software codes, Code to manage configurations; automate provision of infrastructure and deployments .its not limited to sysadmin , developers also can wirte codes to mage the servers.

IAC automation of IT operations build deploy manage by proviosing code and manual

It allows us to control the current state of our managed systems.

* aC requires applying DevOps practices to automation scripts to ensure they're free of errors, are able to be redeployed on multiple servers, can be rolled back in case of problems, and can be engaged by both operations and development teams. The use of modern coding systems like Ansible or Puppet is designed to make IaC environments accessible to anyone with basic knowledge of modern coding techniques and structures. **Manage infrastructure via source control**, thus providing a detailed audit trail for changes.
* **Apply testing to infrastructure** in the form of unit testing, functional testing, and integration testing.
* **Avoid written documentation,** since the code itself will document the state of the machine. This is particularly powerful because it means, for the first time, that infrastructure documentation is always up to date.
* **Enable collaboration** around infrastructure configuration and provisioning, most notably between dev and ops.

**Idempotence** means that if we run a script multiple times over the same resources, we will get the same results.ansible and chef are idempotent, it will not break existing resources

**Converge**: ability to successfully modify or restore the resources.when new changes are introduces and convergence means the resources should be succsffullybe able to restore to previous state

**Immutability**: concept the resources should not be modified oly created and destroyed

Similar to blue green deployment, AWS autoscaling group, cms in teraaform and cloudfromation

<https://github.com/Artemmkin/infrastructure-as-code-tutorial/blob/master/docs/00-introduction.md>

1. CM tools and their feature and diffrerences

|  |  |  |
| --- | --- | --- |
| **Feature** | **Puppet** | **Ansible** |
| Server Nodes | PuppetMaster Servers | Any machine with inventory can serve |
| Workflow | PULLs resources from PuppetMaster | PUSH workflow to work with resources |
| Resource management | Top-down or Resources applied randomly | Top-to-Bottom approach |
| Language Exensibility | Built and tested with Ruby | Built on Python |
| Syntax | Ruby's subset of DSL is used | YAML Files |
| Template Language | Based on RUBY's ERB | Based on Jinja2 |
| DevOps tool Support | Puppet supports DevOps Tools | Ansible supports DevOps Tools |

<https://www.edureka.co/blog/chef-vs-puppet-vs-ansible-vs-saltstack/>

1. PUSH and pull:

Check the screenshot,

1. What is puppet..puppet architecture

Puppet is a configuration Tool which is use to automate administration tasks.Puppet Agent(Client) sends request to Puppet Master (Server) and Puppet Master Push Configuration on Agent.

Puppet is a configuration management tool that is extremely powerful in deploying, configuring, managing, maintaining, a server machine.

In other words you can say that puppet can be used for an entire life of a server, starting from bootstrapping a server, to shredding and giving up a server.

To give you an overview let me say that you can define distinct configuration for each and every host using puppet, and continuously check and confirm whether the required configuration is in place and is not altered(if altered puppet will revert back to the required configuration ) on the host.

Puppet keeps the configuration of your hosts under check, and can be used in one shot to configure a machine from scratch (like installing package, editing and configuring,create and mange users required) etc.

The main added advantage is that you can manage the configuration of almost all open-source tools available out there, using puppet.

* Normally most of the configuration management tool, deploy the required configuration on a machine, and leave them as it is. But puppet keeps on verifying the configuration at a specified interval(which you can modify as per requirement).
* Puppet defines the configurations for a host with the help of a language which is very easy to learn and is only used for that purpose.
* Puppet is used by major players in the industry like Google,red hat etc.
* Larger open-source developer base
* Wide number of platforms are supported in puppet.
* Its works very smooth, even when deployed in a large infrastructure(thousands of hosts to manage)

<https://www.tutorialspoint.com/puppet/puppet_overview.htm>

Fetch the configuaration from a master

Apply catalog to target machine

clients

again target node will send the report to master

ow there

* In Puppet, the first thing what the Puppet master does is to collect the details of the target machine. Using the factor which is present on all Puppet nodes (similar to Ohai in Chef) it gets all the machine level configuration details. These details are collected and sent back to the Puppet master.
* Then the puppet master compares the retrieved configuration with defined configuration details, and with the defined configuration it creates a catalog and sends it to the targeted Puppet agents.
* The Puppet agent then applies those configurations to get the system into a desired state.
* Finally, once one has the target node in a desired state, it sends a report back to the Puppet master, which helps the Puppet master in understanding where the current state of the system is, as defined in the catalog

are multiple steps involved whenever a puppet agent of any node connects to a puppet master server for fetching data. These steps are mentioned below.

**Step 1:** Whenever a client node connects to the master, the master server analyzes the configuration to be applied to the node, and how to apply that configs on the node.

**Step 2:**Puppet master server Takes and collects all the resources and configurations to be applied to the node, and compiles it and make it a catalog. This catalog is given to the puppet agent of the node.

**Step 3:** Puppet agent will apply the configuration on the node, according to the catalog, and then reply back, and submit the report of the configuration applied to the puppet master server.

See the screenshot

All the master slave sa re connected from ssl,

1. How to enable auto signing in puppet

<https://www.edureka.co/blog/puppet-tutorial/>

1. What is chef and its architecture

<https://www.edureka.co/blog/what-is-chef/>

## Ansible…

## What is configuration management:

1. Beefits of ansible:
   1. Idempotent model
   2. Ability to provision cloud infratstruture
   3. Stateless no agent required
   4. Required only ssh and python
2. Chef and puppet:
   1. Idempotent
   2. Clent server
   3. Ruby based and dsls

Cloudinit:

Cloud agnostic,immutable resource management

Terraform

Cloudformation

Cloud formation is an AWS native configuration management tool. It is very similar in ability to Terraform, however it can only be used on AWS. CloudFormation scripts are JSON instructions that provide the necessary information for AWS to create resources and configure instances. Like Terraform, CloudFormation leverages cloudinit and userdata for instance configuration management, so it can be thought of as an immutable technology for instance configuration.

Docker

Configuration management is accomplished via the Dockerfile, which specifies the base container image to launch and any additional configurations you apply on top of the base image. Docker is considered immutable as once the image is built from Dockerfile, it is not changed.

SaltStack

it's Python-based and does not require a client on the target system for communication. However, it does support the classic client/server architecture, similar to Chef. SaltStack additionally supports provisioning of basic virtual machine infrastructures from many popular Cloud providers.

1. Ansible: autmtion engine, it useds yaml syntax, you provision env, configure env , deploy aplictions, speciy how infratsure looks, in   
   AWS autoscalling groups database, load balancer,can be automated

VMs: state of the services and settings ,

Deploy : deploy to multiple envs

Compliance checks: create task for desired state , enforce that state,report any chnages

1. Ansible concepts

Adhoc commands..i f only wants to do once

Playbooks:

1. What is ansible module

Modules (also referred to as “task plugins” or “library plugins”) are the ones that do the actual work in ansible, they are what gets executed in each playbook task. But you can also run a single one using the ‘ansible’ command.modules are individual unit of work, Wrappers of executables code

Each module supports taking arguments. Nearly all modules take key=value arguments, space delimited. Some modules take no arguments, and the command/shell modules simply take the string of the command you want to run.

Custom modules:can be created sing python,shell,ruby which returns json output laos can use ansible module to create custom module in python

Ansible provide test module script

ansible webservers -m service -a "name=httpd state=started"

ansible webservers -m ping

ansible webservers -m command -a "/sbin/reboot -t now"

<http://codeheaven.io/15-things-you-should-know-about-ansible/>

1. What are the modules you have used in Ansible

* File:
* Template:
* Yum
* Service
* Ping

1. Playbooks in ansible

<https://www.edureka.co/blog/what-is-ansible/>

Allows you to execute multiple tasks on multiple servers, (YAML)yaml aint markup language ,powerful format to represent data,mulitline | to ntroduce new line in yaml

if a string is a value and includes ‘ then it should be included in quotes

foo: “{{variable}} “

1. How to see the list of all variable and facts

<https://tekslate.com/ansible-interview-questions/>

1. How to deploy satic contents of an application without deploying entire application

<https://semaphoreci.com/community/tutorials/continuous-deployment-for-static-sites-with-docker-aws-and-ansible>

Handlers,: tasks which executed when it is notified in mail playbook though notify property.

Variables: varaiables ahs precedence when same name is used all more than places,

Template : allow a preprocessing before copied to the server,uses jija2 engien to generate the templates

Facts: gathers usefull varaibles sof the remote hosts,canbe used in playbooks

{{ ansible\_env [‘<name of the prop>’]}}

Debug : nsg…print message into standard output

1. Roles

Bundle common functionality, better reuse code, specific folder structure, ansible galaxy command

1. Basic debugging :ansible stops when error encountered. Ignore-error can be used to make the playbook running even after the error is occurs .
2. Deploy a lamp application in ansible:
   1. You created a reusable role to install a LAMP stack
   2. You created a role to install a Python web application on top of the LAMP stack
      1. You created a task to create a database table and a database user
      2. You downloaded some source code from a Github repo
      3. You installed some dependencies with pip - a python package manager
      4. You created a template to handle the Apache config file with some variables that pull from both Ansible and the defaults for the role
      5. You triggered a notification to restart Apache
3. Anisble inventory

The **Ansible inventory** file defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate. The file can be in one of many formats depending on your **Ansible** environment and plugins. ... If necessary, you can also create project-specific **inventory** files in alternate locations.

Groups

Groups of groups and variales :children

Static inventory: normal inventory manully add hosts

Dynamic inventory… get inventory from other locations like autoscalling in AWS.

If a code produces can be used for inventory creation,AWS has inventory scripts ..1.executable 2.ini files

Ansible tagnamedemo –I /vagrant/ec2.py –m ping

1. Install and configure ansible

Install in centos7

1. How do you secure secrest in ansible

## OPS and dev:

1. Explain devops role in software dev side and ops side

Dev:..plan code test build

Ops: operate deploy release monitor

Youtube simplelearn

1. What testing is necessary to ensure its ready for production—Devops is all about the continuous testing from development to prod..this ensures that code doesn’t have many error and .
2. Role of devops engineer : support environment to ensure continuous tetsng development deliveey
3. How can you reduce load time of dynamic website

<https://howtogetonline.com/how-to-fix-a-slow-website.php>

1. Talk about Webpage optimization, cached web pages, quality web hosting , compressed text files, Apache  fine tuning.

**1)** Also called [search engine optimization (SEO)](https://www.webopedia.com/TERM/S/SEO.html),  website optimization is a phrase that describes the procedures used to optimize – or to design from scratch – a website to rank well in [search engines](https://www.webopedia.com/TERM/S/search_engine.html).  Website optimization includes processes such as adding relevant [keyword](https://www.webopedia.com/TERM/K/keyword.html) and phrases on the website, editing [meta tags](https://www.webopedia.com/TERM/M/meta_tag.html), image tags, and optimizing other components of your website to ensure that it is accessible to a search engine and improve the overall chances that the website will be indexed by search engines.

**2)** A phrase used to describe the procedures to optimize the speed at which your website loads in a Web browser.  This type of optimization generally involves editing your website to optimize scripts, [HTML](https://www.webopedia.com/TERM/H/HTML.html) or [CSS](https://www.webopedia.com/TERM/C/CSS.html) code for faster loading.  It's also reduces the number of components such as images, scripts, or video components that are needed to [render](https://www.webopedia.com/TERM/R/rendering.html) the webpage.

A **web cache** (or *HTTP cache*) is an [information technology](https://en.wikipedia.org/wiki/Information_technology) for the temporary storage ([caching](https://en.wikipedia.org/wiki/Cache_(computing))) of [web documents](https://en.wikipedia.org/wiki/Web_document), such as [HTML pages](https://en.wikipedia.org/wiki/Webpage) and [images](https://en.wikipedia.org/wiki/Digital_image), to reduce [server](https://en.wikipedia.org/wiki/Bandwidth_(computing)) lag. A web cache system stores copies of documents passing through it; subsequent requests may be satisfied from the cache if certain conditions are met.[[1]](https://en.wikipedia.org/wiki/Web_cache#cite_note-1) A web cache [system](https://en.wikipedia.org/wiki/System) can refer either to an [appliance](https://en.wikipedia.org/wiki/Server_appliance), or to a computer program.

<https://winningwp.com/what-is-website-caching-and-why-is-it-so-important/>

<https://javarevisited.blogspot.com/2017/01/12-essential-apache-web-server-interview-questions-answers-java-linux.html?_sm_au_=i6VsDWV14ttJSZrM>

<http://www.linuxforfreshers.com/p/blog-page_15.html>

<https://www.tecmint.com/apache-interview-questions/>

## Career objective:

1. Why you chossed devops ---Its an opportunity to be involved in end to end cycle of delivey system, learning man tools on the way
2. What tools you would choose:

Version control –git

Build : maven,

storing ..nexus, artifactory

Test tools: selenium functional testing and web interfaces

Deployment app in a env: fr provising system..puppet chef or ansible

Docker for microservices

## Scripting languages:

## Python:

## Jenkins:

## How to configure user access in jenkins

# Jenkins versions using : 2.107.1.2

## <https://www.udemy.com/jenkins-quick-start/learn/v4/content>

1. Feature of pipeline:

**Extensibility**----pipeline plugin supports integration and extensions for other plugins

**Code**: code can be edit review and modify as per the requirements   
 **Pausability**---pipeline can be made to wait for human input and run again   
 **Durability**—can survive during planned and unplanned Jenkins restart  
**Versatility** -- Pipelines support complex real-world CD requirements, including the ability to fork/join, loop, and perform work in parallel.

1. What is jenkins:

Jenkins is CI tool,written in java , it helps automate software devleopemnt process , building testing , can be easily integrated with verison control systems, build tools, testing tools and deployemnts technologies. life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis can be automated

Sample workflow :



1. Jenkisn distributed architecture:

It has master slave architecture , can be confgigured , jobs and tools, you can run the job in differet servsers, like may be one job can be run to linux and there to windows, so slaves are created .

Jenkins master handles, shduling of jobs, monitor the slave,dispatching build to the slaves,

Slaves: can be run on operating systems,executing builds,worked on absis of requests recived from master.Like in our proeect for diffren t assets we different slaves .

1. what are the various ways in which build can be scheduled in Jenkins ?

Ans. Builds can be triggered by source code management commits.

Can be triggered after completion of other builds.

Can be scheduled to run at specified time ( crons )----

|  |  |
| --- | --- |
| MINUTE | Minutes within the hour (0–59) |
| HOUR | The hour of the day (0–23) |
| DOM | The day of the month (1–31) |
| MONTH | The month (1–12) |
| DOW | The day of the week (0–7) where 0 and 7 are Sunday. |

* \* specifies all valid values
* M-N specifies a range of values
* M-N/X or \*/X steps by intervals of X through the specified range or whole valid range
* A,B,...,Z enumerates multiple values

Manual Build Requests

1. benafits:

Advantage of Jenkins includes:

* Bugs tracking are easy at early stage in development environment.
* Provides a large numbers of plugin support.
* Iterative improvement to the code.
* Build failures are cached at integration stage.
* For each code commit changes an automatic build report notification generates.
* To notify developers about build report success or failure, it is integrated with LDAP mail server.
* Achieves continuous integration agile development and test driven development.
* With simple steps, maven release project is automated.

1. What is pipeline:

A cd pipeline is automated version of continuous delivery process, getting the code from version control to build test and deploy via series of stages Jenkins pipeline are created using DSL, into a Jenkins file. That a pipeline as a code and can be stored into version control systems

1. Benefits of using jenkinsfile pipeline:

* Automatically creates a Pipeline build process for all branches and pull requests.
* Code review/iteration on the Pipeline (along with the remaining source code).
* Audit trail for the Pipeline.
* Single source of truth [[3](https://jenkins.io/doc/book/pipeline/#_footnote_3)] for the Pipeline, which can be viewed and edited by multiple members of the project.

1. Jenkinsfile types and properties:
2. Declarative pipeline: ..a structure of build process pipeline ..without describing the control flow..starts with pipeline includes stages

Steps.

*Jenkinsfile (Declarative Pipeline)*

pipeline {

agent any

stages {

stage('Build') {

steps {

//

}

}

stage('Test') {

steps {

//

}

}

stage('Deploy') {

steps {

//

}

}

}

}

1. Scripted pipeline..groovy syntax.. starts with node element and define stage .this creates a worksace as well

*Jenkinsfile (Scripted Pipeline)*

node { ----wheneever the node is free it will start executing the stages..stages are executed in a qeueue

stage('Build') {

//

}

stage('Test') {

//

}

stage('Deploy') {

//

}

}

Env.<envvariable>--buildid, job url,job\_name..setting environment {

CC = 'clang'

}

Params.<param>

currentBuild.<varaibale>

handling credentials:.. credentials() to define a credential as env varaivbale from credentials id

withCredentials

..to generate the credentials in pipeline it self.

Handling failure: post {

always {

junit '\*\*/target/\*.xml'

}

failure {

mail to: team@example.com, subject: 'The Pipeline failed :('

}

}

}

Parallel execution:

stage('Test') {

parallel linux: {

node('linux') {

checkout scm

**try** {

unstash 'app'

sh 'make check'

}

**finally** {

junit '\*\*/target/\*.xml'

}

}

},

windows: {

node('windows') {

/\* .. snip .. \*/

}

}

}

Running docker..

* 1. Run script inside a docker container:

pipeline {

agent {

docker { image 'node:7-alpine' }

}

stages {

stage('Test') {

steps {

sh 'node --version'

}

}

}

}

*Jenkinsfile (Declarative Pipeline)*

pipeline {

agent {

docker {

image 'maven:3-alpine'

args '-v $HOME/.m2:/root/.m2'

}

}

stages {

stage('Build') {

steps {

sh 'mvn -B'

}

}

}

}

Using dockerfile:

1. How is Jenkins different from other tools in same category

Jenkisn open source .. supports more tools , inetgarate and extensible through plugins,

1. Benefits of using Jenkins:
2. Features of Jenkins—CICD, easy installation,easy config,plugins,extensible,distributed
3. Important plugins atleast 5 –

Build Pipeline

Maven

Git,ssh

Parametrised trigger

Blueocean

Html plubisher

Sonarqube

Artifactory

Email ext

Cobertura

AWS, docker plugins

Dashboards

Job dsl

1. How to move Jenkins between servers

<https://support.cloudbees.com/hc/en-us/articles/216241937-Migration-Guide-CloudBees-Jenkins-Platform-and-CloudBees-Jenkins-Team->

1. How to restart Jenkins

* http://localhost:8080/safeRestart  
  Error message: Jenkins cannot restart itself as currently configured.
* *Manage Jenkins → Restart Safely Plugin*.  
  Error message: Jenkins cannot restart itself as currently configured.
* Not able to find any Jenkins Service in services.msc.
* Navigate to jenkins-cli directory in CMD mode
* java -jar jenkins-cli.jar -s http://[jenkins-server]/ restart

1. Explain how to create a backup and copy files in Jenkins?

Answer to this question is really direct.

To create a backup all you need to do is to periodically back up your JENKINS\_HOME directory. This contains all of your build jobs configurations, your slave node configurations, and your build history. To create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory.

1. What is function of ci server
2. 17**. How will you secure Jenkins?**

The way I secure Jenkins is mentioned below, if you have any other way to do it than mention that:

* Ensure global security is on.
* Ensure that Jenkins is integrated with my company’s user directory with appropriate plugin.
* Ensure that matrix/Project matrix is enabled to fine tune access.
* Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
* Limit physical access to Jenkins data/folders.
* Periodically run security audits on same.

1. <https://wiki.jenkins.io/display/JENKINS/Plugin+tutorial#Plugintutorial-Deployingacustombuildofacoreplugin>

Stop Jenkins.

Copy the custom HPI to $Jenkins\_Home/plugins.

Delete the previously expanded plugin directory.

Make an empty file called <plugin>.hpi.pinned.

Start Jenkins.

1. Jenkisns supports following scm tools:
2. AccuRev  
   2. CVS  
   3. Subversion  
   4. Git  
   5. Mercurial  
   6. Perforce  
   7. Clearcase  
   8. RTC
3. Blue ocean pipelines

Blue Ocean rethinks the user experience of Jenkins. Designed from the ground up for [Jenkins Pipeline](https://jenkins.io/doc/book/pipeline/), but still compatible with freestyle jobs, Blue Ocean reduces clutter and increases clarity for every member of the team. Blue Ocean’s main features include:

* **Sophisticated visualizations** of continuous delivery (CD) Pipelines, allowing for fast and intuitive comprehension of your Pipeline’s status.
* **Pipeline editor** - makes creation of Pipelines approachable by guiding the user through an intuitive and visual process to create a Pipeline.
* **Personalization** to suit the role-based needs of each member of the team.
* **Pinpoint precision** when intervention is needed and/or issues arise. Blue Ocean shows where in the pipeline attention is needed, facilitating exception handling and increasing productivity.
* **Native integration for branch and pull requests**, enables maximum developer productivity when collaborating on code with others in GitHub and Bitbucket.
* Sophisticated visualization
* Pipeline editor
* Personalization
* Pinpoint precision
* Native integration for branch and pull requests.

1. Jenkins pipeline development tools:

<https://jenkins.io/blog/2017/05/18/pipeline-dev-tools/>

blueocean ,

classic ui ..jenkisn file stored injenkins home directory—scripted can be generated from pipelinesyntax.., declarative can be generated from declarative directive generator

or jenkinsfile

1. Fingerprints in Jenkins

<http://rodrigozrusso.com/2016/11/09/jenkins-fingerprinting.html>

1. As a user, when you will use the Jenkins with the Selenium and what is the scope of the same?

* Running Selenium tests in Jenkins allows you to run your tests every time your software changes and deploy the software to a new environment when the tests pass.
* Jenkins can schedule your tests to run at specific time.
* You can save the execution history and Test Reports.
* Jenkins supports Maven for building and[Testing](https://www.guru99.com/software-testing.html)a project in continuous integration.

<http://learn-automation.com/selenium-integration-with-jenkins/>

<https://www.softwaretestinghelp.com/integration-of-jenkins-with-selenium-webdriver/>

can check the selenium obiee reports job

1. CI tools like Jenkins:

Bamboo, Go CD, TeamCity, CodeShip, GitLab CI, Travis CI, Circle CI.

1. Simple Jenkins pipeline code for java:

<https://github.com/jenkins-docs/simple-java-maven-app/blob/master/jenkins/Jenkinsfile>

<https://www.tutorialspoint.com/jenkins/jenkins_setup_build_jobs.htm>

1. What is declarative jenkins pipeline---

declarative jekins pieline.isa structure build process pipeline script..with out a control flow

1. Agents in Jenkins

Jenkins supports the "master/agent" mode, where the workload of building projects are delegated to multiple "agent" nodes, allowing a single Jenkins installation to host a large number of projects, or to provide different environments needed for builds/tests. This document describes this mode and how to use it.

<https://wiki.jenkins.io/display/JENKINS/Distributed+builds>

1. What is stage—

a stage is set subset of tasks in jenkin pipeline

<https://jenkins.io/doc/book/pipeline/syntax/>

1. What is environment directive

The environment directive specifies a sequence of key-value pairs which will be defined as environment variables for the all steps, or stage-specific steps, depending on where the environment directive is located within the Pipeline.

1. Triggers in Jenkins

cron pollscm upstream

<http://www.andyfrench.info/2015/03/automatically-triggering-jenkins-build.html>

1. Input directive

Getting input from user in pieline..

The input directive on a stage allows you to prompt for input, using the [input step](https://jenkins.io/doc/pipeline/steps/pipeline-input-step/#input-wait-for-interactive-input). The stage will pause after any options have been applied, and before entering the stage`s `agent or evaluating its when condition. If the input is approved, the stage will then continue. Any parameters provided as part of the input submission will be available in the environment for the rest of thestage

1. What is scripted pipeline

Scriptred pipeline is groovy scripted jekins pipeline

1. How to secure jenkins

* Ensure global security is on in configurations
* Ensure that Jenkins is integrated with my company’s user directory with appropriate plugin.
* Ensure that matrix/Project matrix is enabled to fine tune access. Fro who can access which job
* Automate the process of setting rights/privileges in Jenkins with custom version controlled script.
* Limit physical access to Jenkins data/folders.
* Periodically run security audits on same.

1. How to install jenkins
2. How to configure jenkins, installation,
3. Configure executor,
4. Agents in jenkins

## AWS:

Provides: ..Secuority, reliability,performance,cost optimisation,operationa execlelence

* 1. What is the role of AWS in devops?—
  2. What is an AMI

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

* 1. How is IAC implemented using AWS
  2. What is cloud computing
  3. Saas paas iass
  4. What is difference between shutdown and terminate ec2 instance
  5. Private cloud public hybrid and community
  6. How autoscalling works
  7. How to make highly available fault tolrent architecture for web application
  8. What is security group

A security group is a set of firewall rules that control the traffic for your instance.

* 1. Explain AWS infrastructure
  2. How is IAC implemented in AWS
  3. DNS
* Dns hostname attribute in ec2 instance..is used if dns is enabled
  1. IAM roles
  2. **VPCs .. sunets, interntegateways, nacls**---

### VPC:

### is virtual private netwok within aws cloud associated with the aws accont, spans all the AZs in the regions and must specify a range of ip address in the form cidr notation.logically isolated network,

Benefits: public private subnets,conectiong to data center,vpn connections, control routing through route tables,assign multiple ip addresses to ec2 instance,

Has ability to control internet in and internet out,expose and isolate resources,deploy services to vpc.There are no charges to use vpc but biling is there for services associated withit

The default VPC is a logically isolated virtual network in the AWS cloud that is configured and connected to an internet gateway, which means that your instances will automatically receive public IP addresses when they're provisioned into the default VPC.

A vpc can span between AZ not regions

### Subnets

* Further a VPC is subdivided into subnets , can add one or more subnets in each AZs,subnets should have ip addresses with the subset of vpcs range.
* A subnet cannot span zones
* Divide Azs are divided into public and priavte subnets for security.
* VPC peering is a network connection between two or more VPCs in the same region, belong to either yourself or another AWS account that allows you to route traffic between them, using private IP addresses.
* Public subnets: resources will have direct connect ot outside internet
* Private: doent have direct access to internet
* Any subnet must be associated with any one route table, a route table can be associate with many subnets
* Internet gateway allows communication between vpc and internet..

### Routing tables: ..communication between assets in subnets within a vpc is doe through RT.set of rules called routes where network traffic is directed

Default RT ..conenct evry assets to otejr assets within vpc

Public RT: 0.0.0.0/0..any packet can cme to this particular subnet or asset, will path the IGW

Private RT:

**Connectivity**: baston host

If want to connect internet and send packet from internet to private assets subnets…a wayis , to create baston or jumpbox host in public subnet for the purpose od ssh/rdp, resource outside the vpc can login to bston or jumpbox to login to private network where boston host connection to.

create an EC2 instance that will serve as both an observer instance that you can run various tests from and a bastion host.

Note: Once you access a bastion host (for example, by using SSH to log into it), in order to access other instances you must either setup SSH port forwarding, or copy your SSH key material to the bastion host. The latter is not ideal for security reasons in a production environment. If you require Windows connectivity, then setting up Remote Desktop Gateway instead of SSH port forwarding is recommended. This Lab Step assumes SSH connectivity to Linux instances.

**NAT**: is used to connect private sibnet assets to internet as in send packets from private subnet to internet , network access interface is used,use private RT,0.0.0.0/0 to NAT, NAT has access to the internet

A NAT instance enables instances in the private subnet to initiate outbound traffic to the Internet. This scenario is common when you have a public-facing web application, while maintaining back-end servers that aren't publicly accessible.

Another way ..to connect internrt to private subnet is to setup a private data center and connect to vpc via virtual gateway add ip of pdc to private RT going to VGW

**NACL**: control access to subnets stateless,security layer for vpc ,acts as firewall for traffic in and out of subnets

* Your VPC automatically comes with a modifiable default network ACL. By default, it allows all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic.
* You can create a custom network ACL and associate it with a subnet. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
* Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
* You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
* A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest number that you can use for a rule is 32766. We recommend that you start by creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.
* A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
* Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

Elastic network interfces: secuorty instance can talk to isolaited instances ovewr private vpc

**Internet** **Gateway:**

An **Internet Gateway** is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It imposes no availability risks or bandwidth constraints on your network traffic. An Internet gateway serves two purposes: to provide a target in your VPC route tables for Internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IP addresses.

o use an **Internet gateway** your subnet's **route table** must contain a route that directs Internet-bound traffic to the Internet gateway. You can scope the route to all destinations not explicitly known to the route table (0.0.0.0/0), or you can scope the route to a narrower range of IP addresses; for example, the public IP addresses of your company’s public endpoints outside of AWS, or the Elastic IP addresses of other Amazon EC2 instances outside your VPC. If your subnet is associated with a route table that has a route to an Internet gateway, it's known as a **public subnet**.

Thanks to the new route rule, all VPC external traffic will be routed to the Internet Gateway and then to the Internet.

An Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It imposes no availability risks or bandwidth constraints on your network traffic. An Internet gateway serves two purposes:

1. Provide a target in your VPC route tables for Internet-routable traffic
2. Perform network address translation (NAT) for instances that have been assigned public IP addresses. (Note: It does not do this for instances with private IP addresses.)
3. An Internet Gateway can only be attached to one VPC

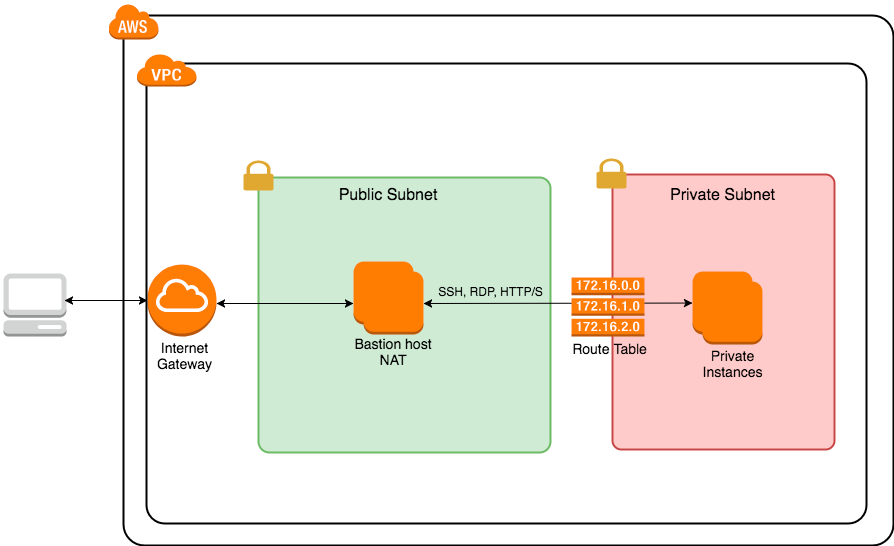
**Steps to create vpc:**

**Vpc🡪subnets->routetables🡪internet gateways-(there should be one to one relation between vpc and IDW)🡪create custom routetable🡪update route tables t include internet gateways🡪create subnet groups**

A simple webserver ad db structure:

## Highlights of Securing your VPC

Let's look at the high level lab environment diagram again, and go over some key points to tie this Lab together. Time and interest permitting, use the AWS Management Console to look up and/or confirm various settings as you read through this Lab Step. Realize that different organizations might configure things differently in accord with their security goals and policies. For example, increased or relaxed security on various inbound or outbound rules, etc.

[](https://assets.cloudacademy.com/bakery/media/uploads/blobid0-6f3335a2-0008-4652-9d80-afd7edd7e8b3.png)

The VPC has been configured with two subnets, a public subnet, and a private subnet. If a subnet's traffic is routed to an Internet gateway, the subnet is known as a *public subnet.*If a subnet doesn't have a route to the Internet gateway, the subnet is known as a *private subnet*. Instances launched in a private subnet do not have publicly routable internet addresses either.

Both subnets have a route table associated with them. Instances on the public subnet route internet traffic through the internet gateway. The private subnet routes internet traffic through the NAT instance.

Each instance launched in either subnet has it's own security group with inbound and outbound rules, to guarantee access is locked down to specific ports and protocols. For example, private instances on the private subnet allow any outbound traffic, but only allow SSH access from the bastion host. As another example, although the NAT instance is in the public subnet, it cannot be reached from the internet. It has an inbound rule that only grants instances from the private security group (private instances) access. Note that you might allow SSH access from your personal IP address or specific adminstrator's as well, or perhaps grant ICMP (ping) access during setup and troubleshooting efforts.

In addition to security groups, the private subnet also has a network access control list (NACL) as an added measure of security. NACL's allow for inbound and outbound rules, specified in a priority order. They are setup as implicit allow rules. If none of them are matched, all other traffic is denied. This private subnet NACL in this Lab allowed for SSH inbound traffic from the public subnet only. The outbound rules for the private NACL allowed for HTTP/S access to anywhere. This was proven to work in the Lab by performing operating system updates once the NAT instance was in place. The private route table sends the traffic from the instances in the private subnet to the NAT instance in the public subnet. The NAT instance sends the traffic to the Internet gateway for the VPC. The traffic is attributed to the Elastic IP address of the NAT instance.

* 1. VPC security :

NACL vs Security groups

* 1. Build and configure a NAT instance..
* NAT instance in our public subnet that will allow requests to come from the private subnet, which will provide it access to the internet. The first thing you need to do is to create a NAT security group, and we need to add an inbound rule for each of the private subnets that we want to be able to use the NAT, and specify the CIDR block and protocols that we want to allow to use the NAT instance.
  1. Difference in ec2 classic , default vpc,custom vpc
  2. What is the difference between start , shutdown and terminate of ec2 instances:
  3. What is reserve instances:

A **Reserved Instance** is a reservation of resources and capacity, for either one or three years, for a particular Availability Zone within a region. Unlike on-demand, when you purchase a reservation, you commit to paying for all of the hours of the 1- or 3-year term; in exchange, the hourly rate is lowered significantly.

* 1. Spot and reserve instance
  2. What are classic ec2 instances
  3. What is elastic ip:
* An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
* An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet; for example, to connect to your instance from your local computer

CloudFormation.., orchestration, infrastructure deployment

* 1. What is best feature in AWS
  2. Significance of sdf in EBS
  3. Manage data in AWS:

**EBS** ..Elastic block storage..cheaper and flexible..its like a usb that can be attached to EC2 instance.it is available When computer is off also

* 1. Volumes a hard device will be associated wth EC2 by default
  2. Create volmes ,select devices,
  3. Attch the volume to instance,device /dev/sdf (/dev virtual file containing information of all the devices associated with in all the linux system.
  4. Use lsblk to check the device
  5. Mount a device to aprticular directory..sudo mount –a /dev/sdforxvdf <target>
  6. Edit /ec/fstab ..mount on boot

**S3..Simple storage service…**

* + Click actions, Create bucket,create folder,upload a file,add security credentials,
  + Accessing the data from linux: WGet, then <http://s3.amazon.aws.com/bucketname/fodlername/file>
  + From aws CLI interface: install awscli, aws configure,fill in access key ,access the file through aws s3 cp <fullbucketfilename> <copiedfilename> --region=<which region>
  + Upload file from ec2 to s3 bucket..aws s3 cp filename <sbucketfilename> --region=

When you upload a folder from your local system or another machine, Amazon S3 uploads all the of files and subfolders from the specified folder to your bucket. It then assigns a key value that is a combination of the uploaded file name and the folder name. In this Lab Step you will upload a file to your bucket. The process is similar for uploading a single file, multiple files, or a folder with files in it.

All uploaded files are private by default and they can only be viewed, edited, or downloaded by you. In order to illustrate this point, complete the instructions below.

Note: The terms "file" and "object" are often interchanged when discussing S3. Techncially, S3 is an object store. S3 is not a block storage device, and does not contain a file system like your local host does. However, files such as images, movies or sound clips are often uploaded from your file system to S3.

Amazon S3 has a set of key/value pairs representing its metadata. There are two types of metadata: "System metadata" (for example, Content-Type and Content-Length) and custom "User metadata". User metadata is stored with the object and returned with it. As an example, you might have your own taxonomy for various images, such as “logo”, “screenshot”, “diagram”, "flowchart" and so on. In this Lab Step you will change the Content-Type of your image to "text/plain". You will also create custom user metadata.

**Glacier** is more cheaper, and its slow ---Java api to upload and download a arhive from eclipse has some libraries to add to aws Amazon Glacier is an extremely low-cost storage service that provides secure, durable, and flexible storage for data backup and archival.

* 1. Creating automated backups for S3:

S3 lifecycle..-->add a rule🡪configure rule🡪save it to glacier

* 1. Datapipeline mix for automated backups: AWS Data Pipeline helps you move, integrate, and process data across AWS compute and storage resources, as well as your on-premises resources. AWS Data Pipeline supports integration of data and activities across multiple AWS regions.
  2. Create datanodes🡪schdules🡪
  3. Can access datapieline---cli,sdk,webservice api
  4. How to access s3 from EC2

Accessing the data from linux: WGet, then <http://s3.amazon.aws.com/bucketname/fodlername/file>

From aws CLI interface: install awscli, aws configure,fill in access key ,access the file through aws s3 cp <fullbucketfilename> <copiedfilename> --region=<which region>

Upload file from ec2 to s3 bucket..aws s3 cp filename <sbucketfilename> --region=

* 1. What is dhcp—

The Dynamic Host Configuration Protocol (DHCP) provides a standard for passing configuration information to hosts on a TCP/IP network.

By default, all instances in a nondefault VPC receive an unresolvable host name that AWS assigns (for example, ip-10-0-0-202). You can assign your own domain name to your instances, and use up to four of your own DNS servers. To do that, you must specify a special set of DHCP options to use with the VPC.

* 1. AWS cli commands:

Aws ec2 create-vpc –cidr block 10.0.0.0/16---it will create vpc

Aws ec2 create-subnet –-vpc-id <vpcid> -cidr … availability-zone <AZ>

Aws ec2 create-internet-gateway

Aws ec2 attach-internet-gateway –igwid vpcid

Aws ec2 create-route-table

Aws ec2 associate-route-table

Aws ec2 modify-vpc-attribute –enable-dns-hostname

aws s3

* 1. Static Web hosting in aws:
  2. Cloudfront: it’s a content delivery network,speeds up distribution of static and deynmc content via worldeide network on edge locations..the request for url is routed to the edge locations not to the webserver,it will check for the cacahe in edge location, if not found,
* CloudFront will compare the request with the specifications in your distribution, and forward the request to the applicable origin service based on the file type
* the origin services will send the requested files back to the CloudFront edge location.
* as soon as the edge location receives the first byte from the origin, it will start to forward those files to the user. These files will also be added to the cache in the edge location for future requests.
* Amazon CloudFront is a content delivery network (CDN) service. You can speed up the delivery of static files using the HTTP or the HTTPS protocols. You can stream audio or video contents in real time using the RTMP protocol. Each CloudFront distribution has a unique *cloudfront.net* domain name that can be used to reference objects through the global network of edge locations.
* Store the original versions of your files on one or more origin servers. An origin server is the location of the definitive version of an object. Origin servers could be an Amazon S3 bucket, an Amazon EC2 instance, an Elastic Load Balancer or another remote server.
* Create a distribution to register the origin servers with Amazon CloudFront.
* Use your distribution’s domain name in your web pages, media player, or application. When end users request an object using this domain name, they are automatically routed to the nearest edge location for high-performance delivery of your content.
* Amazon CloudFront minimizes end user latency by delivering content from its entire global network of edge locations. Price Classes let you reduce your delivery prices by excluding Amazon CloudFront’s more expensive edge locations from your Amazon CloudFront distribution. In these cases, Amazon CloudFront will deliver your content from edge locations within the locations in the price class you selected and charge you the data transfer and request pricing from the actual location where the content was delivered.

Best practices:

Use s3 for static content, as data transfer from s3 to cloufront is free

Control access to content in s3

Control access to content include front

Edge caching

Versioning ,

Dynamic assets: ..

cache everything

streaming media

|  |  |  |  |
| --- | --- | --- | --- |
| DNS | Storage layer | Distribution layer | Backup archival system |
| Route53 | S3..All objects are uploaded static contents , simple landing page .   * Create a bucket with a unique name * Give public access to the bucket * Got to properties, static web hosting , provide index page and error page * Add a policy to bucket to provide permission to all the object s * Upload the files | CDN..use cloud front—create distribution.  Give bucket url , it wil genarwte a domain name  Viwers page can show the browsers informations can also give apache server name using apache as origin for website |  |
|  |  |  |  |

## Route53:

It’s a domain name system,convert domain anmes to ips, provides secure and reliable routing requests for Aws services o outside aws services

Hosted zne: is a container, routing information , collection as rresource records.

* Public hosted zone
* Private hosted zone for vpc

DNS Resource record types:

* CNAME, TXTSPF..many other

Routing policy..

* Simple--
* weighted ..multiple resource records,want traffic between
* latency .. resources in multiple datacenters
* failover..active passive failover..only be configured in public hosted zones
* geo-location

pricing based on

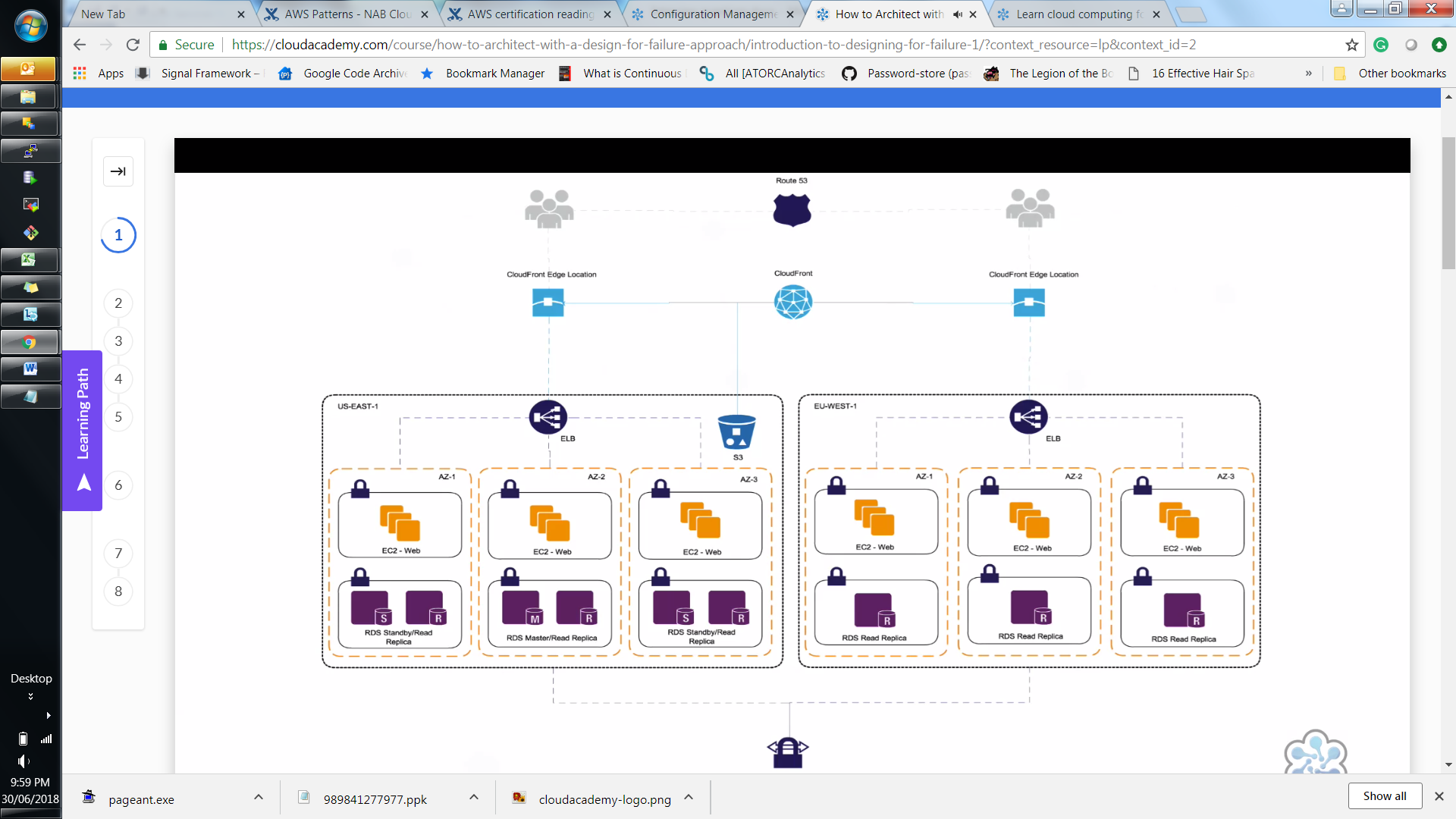
* hosted zones
* no of queries
* helath checks
* domain names
* Service level agreements

Register domain name using route53

Transfer domain name

1. Which route53 policy have you used
2. What is record sets in DNS
3. What is active passive failover
4. How to implement latency based routing
5. What are private hosted zones
6. How to impement health checks in route53, hight avavialibilty
7. How to design high availability architecture:

5 9s means 99.999 , 5 .5 mins per year downtime

* 1. Design to self heal—auto scalling
  2. Fault tolrent infrastructure
  3. Elimainate single point of failure
  4. Graceful degradation of instances—failover …if all the instances then s3 site should come,
  5. 

## Set up vpc , 6 subnets 2 for each 3 availability zones,

## Create rds , create a subnet group,launch db ,

## Create ELB , it periodicaly do health checks, pass the requetsto the healthy instance only ,create autoscalling policy, create launch configuration,create auto scaling group

## Add route 53 , …create a s3 bucket with same domain name,create failover site

## Add cloud fornt for dynamic content,

## Cloudwatch :

Metrics

A metric is the fundamental concept in CloudWatch and represents a time-ordered set of data points. These data points can be either your custom metrics or metrics from other services in AWS. You or AWS products publish metric data points into CloudWatch and you retrieve statistics about those data points as an ordered set of time-series data. Metrics exist only in the region in which they are created.

Think of a metric as a variable to monitor, and the data points represent the values of that variable over time. For example, the CPU usage of a particular Amazon EC2 instance is one metric, and the latency of an Elastic Load Balancing load balancer is another.

Namespaces

CloudWatch namespaces are containers for metrics. Metrics in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics.

Note: In this lab you will see namespaces that AWS has created for you, and a custom namespace created by the steps performed in this lab.

Alarms

You can use an alarm to automatically initiate actions on your behalf. An alarm watches a single metric over a specified time period, and performs one or more specified actions, based on the value of the metric relative to a threshold over time. The action is a notification sent to an Amazon SNS topic or an Auto Scaling policy. You can also add alarms to dashboards.

Alarms invoke actions for sustained state changes only. CloudWatch alarms will not invoke actions simply because they are in a particular state. The state must have changed and been maintained for a specified number of periods.

 installing some monitoring scripts on an EC2 instance. EC2 User Data is a simple way to run shell scripts during the first boot cycle of an instance. This is very useful and powerful when bootstrapping the instance. In fact, you can effectively customize the instance without even connecting to it via SSH or RDP. Although the monitoring scripts will be installed once when the instance boots, they will add an entry to crontab so that the custom metrics can be gathered and sent to CloudWatch every five minutes for you to view.

#!/bin/bash

yum install -y perl-DateTime perl-Sys-Syslog perl-LWP-Protocol-https

wget http://aws-cloudwatch.s3.amazonaws.com/downloads/CloudWatchMonitoringScripts-1.2.1.zip

unzip CloudWatchMonitoringScripts-1.2.1.zip

rm CloudWatchMonitoringScripts-1.2.1.zip

echo "\*/5 \* \* \* \* /aws-scripts-mon/mon-put-instance-data.pl --mem-util --disk-space-util --disk-path=/ --from-cron" > monitoring.txt

crontab monitoring.txt

rm monitoring.txt

**OK**—The metric is within the defined threshold  
**ALARM**—The metric is outside of the defined threshold  
**INSUFFICIENT\_DATA**—The alarm has just started, the metric is not available, or not enough data is available for the metric to determine the alarm state

## Elastic beanstalk

Deploy an application through elastic beanstalk..

## AWS CICD

Source code: codecommit,connec to code commit through ssh and https, can also integrate code commit to IAP , provide permisisons in form policy json files

Codecommit🡪codebuild(build and test)🡪code deploy

CodeCommit: AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly-scalable private Git repositories. CodeCommit eliminates the need to operate your own source control system or worry about scaling its infrastructure. You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.

Codebuild: compile build and test source code ,compile java and donet source code, build and compile docker images,trigger to build automatocaly on commits, supports docker container build cenvs.

Code build uses yaml file buildSpec file..

* Use a different build spec file for different builds in the same repository, such as buildspec\_debug.ymland buildspec\_release.yml.
* Store a build spec file somewhere other than the root of your source directory, such as config/buildspec.yml.

buildSpec file..

version:

phases:

install:

commands

build:

can also do this from aws cli : aws codebuild…

codedeploy : deploy code CodeDeploy can be used to take your Java or .NET libraries and/or executables and deploy them onto your web servers ec2 or on prem servers or take your updated lambda function and deploy it into the lambda servers, it sued sgents base method ,

* 1. Install codedeploy agent :into ec2…aws provides scripts
  2. It uses a json or yaml based appspec.yml to drive a deployment
  3. Deployment groups: group of ec2 instances deployment srvers, can be used in auto scalling groups
  4. Deployment speed can also be checked
  5. Can be integrated to cloudwatch,deployment triggers
* Code pipeline: CodePipeline is a fully managed, continuous delivery system used to automate the building, testing, and deployment of your code. CodePipeline provides visualization tools that give you the ability to understand the high level steps involved and orchestrated by CodePipeline.
* AWS CodeDeploy is a platform-agnostic automated code deployment service. You can run AWS CodeDeploy anywhere, not just on EC2 instances. The service works by installing a CodeDeploy Agent on servers which respond to deployment tasks. AWS CodeDeploy makes it easy to handle rollbacks and provides support for two deployment types: in-place and blue/green. AWS CodeDeploy deployment groups target a group of instances to deploy the application. You can specify the instances using tags or Auto Scaling groups. AWS CodeDeploy also leverages an Elastic Load Balancer (Classic) to balance requests between deployment group instances.
* You will set up deployment groups for both in-place and blue/green deployments in this Lab Step. The Cloud Academy Lab environment has created the Auto Scaling group and load balancer for you. The capacity of the Auto Scaling group is two. Following best practices, the Auto Scaling group and load balancer are using multiple Availability Zones for fault-tolerance. You will make use of the deployment groups in the following Lab Steps that work with the complete continuous deployment pipeline.
* After a few seconds, you will see a congratulatory message. It mentions one way to use AWS CodeDeploy. You can disregard most of the message since you will use AWS CodePipeline to automate the deployment. However, it is a good time to look at an **AppSpec** file. Every application needs an appspec.yml file at the root of the project directory in order to work with AWS CodeDeploy. This file instructs AWS CodeDeploy on how to deploy the application. It is similar to the buildspec.yml file that is used by AWS CodeBuild.
* AWS CodePipeline is a highly available continuous integration and continuous delivery service in the cloud. AWS CodePipeline defines a workflow to describe your software release process. The workflow is divided into stages and each stage is comprised of one or more actions. Actions Stages are connected by transitions. Actions can be anything from building or deploying, to publishing to an SNS Topic for manual approval or executing an AWS Lambda function. Transitions are visually represented as arrows from one stage to the next.
* When a stage succeeds, it creates an output artifact. These artifacts are kept in an S3 bucket that is managed by AWS CodePipeline. The following stage uses the artifact as its input artifact to perform actions on. If a stage fails, subsequent stages are not executed. For example, assume you have a build stage and a deploy stage. If the build stage fails, AWS CodePipeline will not attempt to deploy. The entire pipeline is executed when a new source code *revision* is detected.
* In this Lab Step, you will create an end-to-end deployment pipeline that uses the AWS Code Services you have set up: AWS CodeCommit, AWS CodeBuild, and AWS CodeDeploy. At first you will use the in-place deployment group defined in AWS CodeDeploy. You will use the pipeline you build throughout the remainder of this Lab.
* You have been using in-place deployments for your code up to this point. One disadvantage of in-place deployments is that an instance is out of service during an update to a new application application. While updates are happening your deployment group has fewer instances than you desire to serve traffic. You can minimize this impact by using the one at a time deployment configuration. However, the time to upgrade the deployment is longer than doing half at a time. These two strategies are the same for your deployment group because it only has two instances, but in general this is not the case. In-place deployments can also run the risk of subtle configuration issues. That is because the same instances are used for deploying every version of the application. Over time the server configuration may drift and cause unexpected problems.
* Blue/green deployments overcome the shortcomings of in-place deployments. To overcome in-place deployment issues, blue/green deployments utilize both the original *blue* environment and a second *green*environment. During a deployment, a green environment is created using new instances. By not reusing instances there is no risk of configuration drift. This is sometimes referred to as *immutable infrastructure*. The instances in the green environment install and run the new version of the application. While this is happening, all of the instances in the blue environment continue to serve traffic as usual. Once the green environment instances are ready and the application is validated, it is safe to switch traffic to the green environment. The switch is achieved using an Elastic Load Balancer in AWS CodeDeploy. When the traffic switches to the green environment, it becomes the blue environment. You can keep or destroy the original blue environment depending on whether or not you anticipate possibly needing to roll back. The blue and green environments are referred to as original and replacement in some areas of AWS CodeDeploy.
* In this Lab Step, you will see how AWS CodeDeploy seamlessly manages a blue/green deployment. You modify your code pipeline to use the blue/green deployment group you created earlier. You will commit a new feature to your AWS CodeCommit repository and observe the blue/green deployment procedure in AWS CodeDeploy.

## Programming questions devops:

1. String reversal
2. Implement linked list
3. Data structures
4. Algorithms

Notes…in systesm

## Git:…. <https://git-scm.com/book/en/v2>

1. Explain git repository: --

It contains the set of files of a project with the information on changes in the files.it has commit objects (set of files. Reflecting the state of a project at that time, references to parent commit object, SHA1 name for commit that uniquely identifies the commit; ), hooks set of references to commit objects

Git repo is called .git placed along with other project contents...that’s why it is different from cvcs cause the git repo is stored along with project there is no central servers.— stores that version of the file in the Git repository (Git refers to them as blobs),

├── HEAD---  
├── branches  
├── config----it has content run from git config, with git url ,remote branches   
├── description---description of repo  
├── hooks---set of scripts to run on git phases like commit push pull  
│ ├── pre-commit.sample  
│ ├── pre-push.sample  
│ └── ...  
├── info  
│ └── exclude—file that should be ignored   
├── objects---all the compressed objects with the hash names  
│ ├── info  
│ └── pack  
└── refs—references to the commit objects  
 ├── heads  
 └── tags

1. Difference between git and svn

|  |  |
| --- | --- |
| Git is distributed vcs—repsodiries can have multiple copies of repository along with source code  Speedy Source code management locally  Repository is copied on local server not on any centralised server  Or need any network access for source code management  Git branches are easier to work with  Git is secure as it encrypts the content into a hash code using algo SHA1  Content is stored as metadata  Git is not open source | Svn is non-distributed vcs  And client server system—each developer checkout the repository into the workspaces and commit back  A centralised server for repository , need to connect to server for any file changes  Content is stored as files  Svn is open source |

1. How does git commit works:---

Git commit: if the file didn’t change git adds the name of the compressed file into the snapshot. Else it compress the files and store to in objects folder and adds the compressed file hash name to the snapshot, once the snapshot is created it will also compressed and placed in object folder.. A commit file has

* 1. Hash name
  2. Comment
  3. Commiterinfo
  4. Hash of parent commit

git cat-file -p <diectryname><commitname>

git add will create a folder for the blob of the staged file

git commit..will create 2 folders

* 1. Commit fodler..has all the info to tree and commiter
  2. Tree folder or snapshot folder has link to all blobs includes all the files blobs or compressed files



1. State of file in git …
   1. Modify—files are modified..in working directory
   2. Staged---changes that are staged to commit…in staging area
   3. Commited ..changes are commited to the local database
2. Term cherry-pick in git

Cherry picking in git means to choose a commit from one branch and apply it onto another.

$ git checkout rel\_2.3

$ git cherry-pick dev~2 # commit F, above

Using git cherry-pick The command git cherry-pick commit applies the changes introduced by the named commit on the current branch. It will introduce a new, distinct commit. **Strictly speaking, using git cherry-pick doesn’t alter the existing history within a repository; instead, it adds to the history. As with other Git operations that introduce changes via the process of applying a diff, you may need to resolve conflicts to fully apply the changes from the given commit** . The command git cherry-pick is typically used to introduce particular commits from one branch within a repository onto a different branch. A common use is to forward- or back-port commits from a maintenance branch to a development branch.

1. What is git clone and git fork

<https://github.community/t5/Support-Series/The-difference-between-forking-and-cloning-a-repository/ba-p/1372>

1. What is a head in git

Git head is a pointer to the tip of branch .refrence to a commit objects last commited.. git commit –a, this is a pointer to the local branch you’re currently on

git checkout master

it will do 2 things:

moved the head to the master branch and changed all the files in wrking directory to the snapshot that master points to

1. What is git reset

Git reset is used to undo changes of staging and commit history..

1. Whenever git add is run it adds the file to the staging index meas it needs to be added to next commit..

Git reset –soft <commit> ..it will reset head of the branch to the specified commit removes comits made after that specified commit

Git reset –mixed <commit> it reset the head to specified commit remove sthe commit after that and removes all steged files

Git reset –hard..reset the head remoes commit, removes staged files and remove working directory chnages

1. Conflicts in git

Conflicts are the occurred when try to merge the changes in same lines between two branches, git is not able to merge them automatically

1. What is bare repository in git

A bare repository is created by git init –bare…this repo has only .git folder with no working directory or say no project source code. This repo ends in .git extension. Developer has to clone this directory take any changes and commit to it.

Bare repo is like a storage gilt repo from which data can be pulled and pushed to. But cannot be committed.it can be central repository for any organization

1. Difference between git pull and git fetch

Git pull---pull tge code from repo and rebase the local repo

Git fetch ..pull the code from repo .. do not do merging or rebase..

Needed to git rebase after git fetch

1. Git revert—revert a commit

**git revert** <commitname> revert the changes from the specified commit , create a new commit and point the head to the new one

List of file changed in a particular commit

**Git diff-tree –r(list individual files) <hash>**

Git diff-tree –no-commit-id --🡪 no commit id in output

1. Submit

Git clone

Git add

Git commit

Git push

1. Benefits of using git

* Distributed model: This means your work is your own. You can let others see only what is necessary. Not everything has to be public. There are other advantages to the distributed model, such as the speed (since most everything is local) and possibility of working offline
* Branching and merging are easy: Branching is a walk in the park. It feels like a natural part of the workflow. They are cheap (fast and consume very little space) so that you can branch whenever you want. This means you can sandbox your features and ideas till they are ready for the mainstream.
* Workflow is flexible: Compared to Centralized VCS, git has the qualities that allow to choose your own workflow. It can be as simple as a centralised workflow to as hierarchical as the dictator-lieutenant workflow. Use the process that best fits you.
* Data integrity is assured: Because git uses SHA1 trees, data corruption due to external reasons can be easily detected.

1. Objectives of git designs
2. What is git rebase

Taking changes from one branch and replay them on other..

Git checkout f1

Git rebase master------this will take the difference in between f1 head and starting commit of f1 and replay them to master head and creates a new commit in branch f1.

The new commit belongs to f1 branch now

Git checkout master

Git merge f1..will merge the new commit to the master and now master also has the newly created commit from rebase.

1. Branching strategies :

Git branch is movable pointer to commits..git log –graph will show all the commits included in that branch.

* 1. Feature branching—every feature has one branch once that feature is deployed in prod; the code is merged into master. Like gdw edw
  2. Task branching—like Rome... each issue has its own branch and once resolved will be merged into master
  3. Release branching—A branch per release like in dar... a branch is created for a release all the developemnt is done into release branch coz..the rpd compient beng used is signle for different project

1. What is vcs and benefits of vcs

VCS is a system that keeps tracks the modifications on files over a period of time so that it can be reverted to the previous state later if some mistaken happens.

CVCS: centralised... a single server that has all the files versions and clients or developers can checkout the files to modify and check in again—has disadvantage of central point of failure—cvs, svn

DVCs...Distributed version control. Clients checkout the snapshot files, fully clone or mirror the repos with history so that it can be copied back to the server—git , bazaar, mercurial,monotone

Benefits: It included creation deletion edits as history.

Team members can work concurrently on the same code

Tracebiltiy :changes to software are traceabale

1. Branching strategies that was used in project

Rome—task branching

Gdwedw..Feature branching

Dar—release branch

1. Explain architecture of git

Explain three tree architecture and how commit works

1. Some basic gitcommands

**git los:**

git log –stat

git log --graph

git log --pretty=oneline

git ls-files

1. How do you squash last n commit into a single commit

Combining last n commits through rebase..dont use on branches shared with there and the commits already pushed to remote repos

Git rebase –I HEAD~3…last 3 commits… p pick , s sqash

Git rebase –-interactive..

With git reset:

Git reset-soft HEAD~N &&

1. What is git bisect

Git bisect is used to debug the code in git. It used binary search algorithm on code. Gitbiset start will start the search and then you can mark a commit as good or bad .Bad commit is the one which contains the bug .and good which was working fine. Then git will get the commit in middle of the good and bad commit and select good or bad

Git bisect good

Git bisect bad

Git bisect reset. Sets to the commit before git bisect

Git bisect log

1. How to use git bisect to determine source of regression testing

Git bisect run <command>

1. Patching in git:

Patching is way to apply one branch commit changes to another branch

git format-patch master --stdout > fixemptyposter.patch --this will have all the commits which are thereon fixemtyposter.patch bt not on master..

git apply --stat fixemptyposter.patch

1. How to resolve conflicts in feature branch

git fetch origin # gets latest changes made to master

git checkout feature # switch to your feature branch

git merge master # merge with master

# resolve any merge conflicts here

git push origin feature # push branch and update the pull request

**Fixing via rebase:**

git fetch origin # gets latest changes made to master

git checkout feature # switch to your feature branch

git rebase master # rebase your branch on master

# complete the rebase, fix merge conflicts, etc.

git push --force origin feature

Note carefully that the --force option on the push after rebasing is probably required as the rebase effectively rewrites the feature branch. This means that you need to overwrite your old branch in GitHub by force pushing.

1. Smoke test before commit,--through pre-commit hooks..run a script

Smoke test using precommit hooks to ensure that no broken code going to git..

from the top-level working directory of your repository. When the script is executed (right before a commit), the exit code will be inspected. If it is zero, the commit will continue. If it is non-zero, the commit is **blocked** and the working directory is left in the “dirty” state (you can inspect it with git diff).

1. How to find alist of files changed in a commit

**git diff-tree -r {hash}**

**git diff-tree –no-commit-id –name-only -r {hash}**

git diff --name-only SHA1 SHA2

git diff --name-only HEAD~10 HEAD~5

git diff --name-only `git merge-base origin/master HEAD`

git diff --name-only HEAD...master

1. Pre-recive , update,post recive hooks

<https://www.digitalocean.com/community/tutorials/how-to-use-git-hooks-to-automate-development-and-deployment-tasks>

1. How merge works in git…

**Fastforward merge**: if the commit to be merged can be rechaed from the target branch commit history the it will just repoint the latest commit of target branch to new commit.

**Resursive strategy:** when the commit to merge is not on the direct ancestor directory ..three way merge happens..creates new snapshot and creates a new commit..refers to as the merge commit..

If the same file is changed in the above scnerio it will create amerge conflict..can use git mergetool for graphical

1. How to know that if a branch has already merged into master

git branch –v—to check the commit

git branch –merged to see what branches ar emerged to the current branch

git branch –no- merged

1. Rename a branch

git branch -m Automaterefresh Automaterefreshnew

1. Make utility
2. Git flow release process

## Maven:

1. How maven works
2. Configure and install maven
3. Maven build lifecycle

## Networking :

1. Types of http requests
2. How does http works
3. Describe two factor authentication
4. What is an mx record in dns
5. How does traceroute work
6. Load average
7. Difference in tcp and udp
8. Icmp
9. Ipv6
10. Steps to change the hostname
11. Ssl tunnel
12. What happens address is entere in browser
13. Ptr in dns
14. How measure the performance of a server

## Database:

1. NoSql database over rdbms

|  |  |
| --- | --- |
| SQL | NoSQL |
| Table based  Predefined schemas  Mysql, oracle,sqlite,postgres  For scalability: In most typical situations, SQL databases are vertically scalable. You can manage increasing load by increasing the CPU, RAM, SSD, etc, on a single server  SQL databases are best fit for heavy duty transactional type applications, as it is more stable and promises the atomicity as well as integrity of the data  SQL databases emphasizes on ACID properties ( Atomicity, Consistency, Isolation and Durability)  Uses structured query language | Document based,key value pair, graph based  Dynamic schemas  Mongodb,bigtable,redis,cassendra,hbase,coubase,couchdb  NoSQL databases are horizontally scalable. You can just add few more servers easily in your NoSQL database infrastructure to handle the large traffic.  you can use NoSQL for transactions purpose, it is still not comparable and sable enough in high load and for complex transactional applications  NoSQL database follows the Brewers CAP theorem ( Consistency, Availability and Partition tolerance )  Unstrutured query language   * Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents. * Graph stores are used to store information about networks of data, such as social connections. Graph stores include Neo4J and Giraph. * Key-value stores are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value. Examples of key-value stores are Riak and Berkeley DB. Some key-value stores, such as Redis, allow each value to have a type, such as 'integer', which adds functionality. * Wide-column stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.   Large volumes of rapidly changing structured, semi-structured, and unstructured data   * Agile sprints, quick schema iteration, and frequent code pushes * Object-oriented programming that is easy to use and flexible * Geographically distributed scale-out architecture instead of expensive, monolithic architecture |

|  |  |
| --- | --- |
| SQL Databases | NOSQL Databases |
| **Types** | One type (SQL database) with minor variations | Many different types including key-value stores, [document databases](https://www.mongodb.com/document-databases), wide-column stores, and graph databases |
| **Development History** | Developed in 1970s to deal with first wave of data storage applications | Developed in late 2000s to deal with limitations of SQL databases, especially scalability, multi-structured data, geo-distribution and agile development sprints |
| **Examples** | MySQL, Postgres, Microsoft SQL Server, Oracle Database | MongoDB, Cassandra, HBase, Neo4j |
| **Data Storage Model** | Individual records (e.g., 'employees') are stored as rows in tables, with each column storing a specific piece of data about that record (e.g., 'manager,' 'date hired,' etc.), much like a spreadsheet. Related data is stored in separate tables, and then joined together when more complex queries are executed. For example, 'offices' might be stored in one table, and 'employees' in another. When a user wants to find the work address of an employee, the database engine joins the 'employee' and 'office' tables together to get all the information necessary. | Varies based on database type. For example, key-value stores function similarly to SQL databases, but have only two columns ('key' and 'value'), with more complex information sometimes stored as BLOBs within the 'value' columns. Document databases do away with the table-and-row model altogether, storing all relevant data together in single 'document' in JSON, XML, or another format, which can nest values hierarchically. |
| **Schemas** | Structure and data types are fixed in advance. To store information about a new data item, the entire database must be altered, during which time the database must be taken offline. | Typically dynamic, with some enforcing data validation rules. Applications can add new fields on the fly, and unlike SQL table rows, dissimilar data can be stored together as necessary. For some databases (e.g., wide-column stores), it is somewhat more challenging to add new fields dynamically. |
| **Scaling** | Vertically, meaning a single server must be made increasingly powerful in order to deal with increased demand. It is possible to spread SQL databases over many servers, but significant additional engineering is generally required, and core relational features such as JOINs, referential integrity and transactions are typically lost. | Horizontally, meaning that to add capacity, a database administrator can simply add more commodity servers or cloud instances. The database automatically spreads data across servers as necessary. |
| **Development Model** | Mix of open-source (e.g., Postgres, MySQL) and closed source (e.g., Oracle Database) | Open-source |
| **Supports multi-record ACID transactions** | Yes | Mostly no. MongoDB 4.0, scheduled for Summer 2018\*, will add multi-document transactions. Sign up for the [beta today.](https://www.mongodb.com/transactions) |
| **Data Manipulation** | Specific language using Select, Insert, and Update statements, e.g. SELECT fields FROM table WHERE… | Through object-oriented APIs |
| **Consistency** | Can be configured for strong consistency | Depends on product. Some provide strong consistency (e.g., MongoDB, with tunable consistency for reads) whereas others offer eventual consistency (e.g., Cassandra). |

<https://www.tutorialspoint.com/mongodb/mongodb_environment.htm>

## Continuous monitoring:

<https://newrelic.com/devops/toolset>

<https://blog.chef.io/2018/04/05/continuous-monitoring/>

1. What is continus monitoring

* Complete operation monitoring (e.g. Up-time, Availability, Concurrency, Jobs, etc.)
* Aggregated and searchable alert and error log files
* Host level utilization trends (e.g. CPU, Memory, Storage)
* Pre and Post deployment baselines and comparisons
* Establish key performance metrics (KPI's)
* Profiling application database usage down to the statement level
* Tracking performance coincidental to changes in and around the application database

1. Any monitoring tool

Icinga

Sensu

Cloud watch

<https://www.blazemeter.com/blog/top-ten-monitoring-tools-every-devops-needs>

<http://cyborginstitute.org/projects/administration/monitoring-tactics/>

new relic

nagios

elk for log management

syslog

## Containers and virtualization

## <https://www.udemy.com/containers-101/>

1. What are containers

Container’s: it’s a runtime environment consists of applications and it dependent libraries,configuration files

1. Advantage of containers over virtualization

|  |  |
| --- | --- |
| Containers are lightweight, sit on top of the host os | Vms has it own OS hence, takes more resources |

1. How containers differ from hypervisor virtualisation
2. What is docker image , docker container, docker hub

Docker image is a basic unit of docker,

Docker container:

Docker hub: docker hub is cloud repository for images, it also allows to build images,test and stores,push to docker cloud so that it can be deployed to the host

1. What is docker swarm
2. What is docker file and docker-compose file

Dockerfile: dockerfile is a text document ,containes command to assemble an image

1. Use json for dockercomposefile

**docker-compose -f docker-compose.json up**

1. How to create docker container

**Docker run –t <imagename> <commandname>**

1. How to stop and start docekr container

Docker start <contaiener id>

Docker stop < contaiener id>

1. Docker container scale
2. Plateforms docker run on
3. What happens to data when container exists
4. How to link containers,
5. How will you share users in nodejs and mongodb
6. Without using Docker, can you see the processes running inside a container from the outside?
7. How does container linking works
8. How do you optimise dockerfile
9. What is the difference between trying to connect to a port that is not being listened to as opposed to one that is firewalled in terms of TCP?
10. Types of virtualisation
11. How did you use docker in past position

Explain how used docker wiith other tools jenkins, compare other tools with Docker functionality

## How far do Docker containers scale?

Large web deployments like Google and Twitter, and platform providers such as Heroku and dotCloud all run on container technology, at a scale of hundreds of thousands or even millions of containers running in parallel.

1. Is the data deleted when docker container exists:

You can answer this by saying, no I won’t loose my data when Dcoker container exits. Any data that your application writes to disk gets preserved in its container until you explicitly delete the container. The file system for the container persists even after the container halts.

## Change management

ALM application lifecycle management

OS installation provisioning

1. What all major Monitoring, Alerting, logging and Trending Tools?

## Linux:

1. mv a b c..

move a and b to directory c

1. Linux boot process
2. Centos .

. sshd\_config..to remove root persmiions

Restart ssh service

Edit sudoers file to give user permissions

Useradd Sakshi

Passwd for password

1. Package installation in centios

Yum: yellow dog updater, modifier

Rpm packages (red hat package management)

Yum search zip

Yum check-update

Yum update <paakgename>

Yum update

Yum info <package>

Yum list <package>

Yum search <package>

Yum remove <package>

Yum grouplist

Yum groupinfo <group name>

Yum groupinstall <groupname> packagesname

Yum groupremove

Packages not avalilabale in centos:

Download the rpm package.

Import the public key rpm –I <name of rpm file> to install rpm package

Yum install packagename

1. difference between **‘cmd1 ; cmd2’** and **‘cmd1 && cmd2’**
2. difference betweem linux and unix
3. command to delete all empty files
4. SFTP –simple file transfer protol
5. RAID
6. How to check dns records in linux
7. Difference between athenticationa nd authorisation
8. http vs https
9. ssl and certificates
10. SELinux…Securty enhanced linux..provide access control policies,tags files and resources ,processes inherit permssions,

DAC..standrad discretionary access control

MAC: Mandatory access control used by selinux, labels to resiources..

Sestatus..to see systsms selinux status

Security context:ls –z, user context,role context,type context,mlz context

1. Firewalld: firewalld is a software to manage firewalls:…

Firewall-cmd –get-zones

Firewall-cmd –get-active-zone

Firewall-cmd –zone=public –list-all

Firewall-cms –zone=public –add-port=80/tcp –permannt

Service firewalld restart

Firewall-cmd –zone=public –remove-port=80/tcp

1. Basic webserver: yum groupinstall

17 .An **SMB client** program for UNIX machines is included with the **Samba** distribution. It provides an ftp-like interface on the command line. You can use this utility to transfer files between a Windows 'server' and a Linux **client**.

Install samba,

**ITIL**

ITIL is a set of IT Service Management practices that focuses on aligning IT services with the needs of business.it help manage delivery,support,help imrpved service delivery and customer satisfaction, better management f business risk or failures

Stages:

Service strategy—finanacial management ,service portfolio management , demand strategy

Service design—SLAs,Service portfolio management ,Avaialability management , service catalog ,capacity management ,

1. People – Human resources and organizational structures required to support the service  
2. Processes – Service Management Processes required to support the service  
3. Products – Technology and other infrastructure required to support the service  
4. Partners – Third parties which offer additional support required to support the service

Service transition—change mgnmnt,release and deployment managmnt ,asset and configuration mgmnt,knowlwdge mgmnt

Service operation—incident mgmnt, problem managemnent,service desk,request fulfilment, application management ,IT operations management,

Continual service improvement

## Assignments:

**Requirement**

* Execute this task on Linux either RedHat family or Debian.
* Install Apache/PHP/Mysql using native packages.
* Configure the default index page to show php information.
* Also enable apache-status page to be accessible from your ip address.

**Requirement**

* Execute this task on Linux either RedHat family or Debian.
* Install Apache/Tomcat
* Configure JK module or proxy module to server the tomcat status page from apache.
* Create a self signed certificate
* Enable SSL on Apache with self signed certificate

**Requirement**

* Execute this task on Linux either RedHat family or Debian.
* Install mysql server on two machines
* Configure replication process for the database.

**Requirement**

* Execute this task on Linux either RedHat family or Debian.  
  Install nginx  
  Setup wordpress on it

**Requirement**

* Execute this task on Linux either RedHat family or Debian.
* Create user developer1
* Add him to the group project1
* Also add him to the sudo group sgroup1

**Requirement**

* Execute this task on Linux either RedHat family or Debian.
* Install apache on 2 virtual machines
* Install haproxy as a load balancer
* Setup sample website on apache web servers
* Configure the stats page to be available publicly
* Make sure the website is still available even if one web server goes down

<https://www.tadaay.com/blog/devops-interview-questions/>

https://github.com/spring-guides/gs-accessing-data-mysql/blob/master/complete/src/main/resources/application.properties

<http://www.springboottutorial.com/spring-boot-unit-testing-and-mocking-with-mockito-and-junit>

JAVA spring boot CI pipeline:

Through jenkins

Through AWS codepieline

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Source code | compile | Code review | Unit testing | Build | package | Deploy | Continous Testing | Continous monitoring |
| Docker |  |  |  |  | Build docker image | Push to docker registry | Run docker contianer statging using kubernates : <https://about.gitlab.com/2016/12/14/continuous-delivery-of-a-spring-boot-application-with-gitlab-ci-and-kubernetes/> | selenium |  |

The ideal candidate has the proven ability to manage and maintain cloud infrastructure such as AWS, Azure, Google Cloud; strong experience in technologies such as Docker, Kubernetes, Functions-As-A-Service; sound knowledge in infrastructure-as-code approaches with Ansible, Puppet, Chef or Terraform; hands-on knowledge of implementing multi-staged CI/CD with tools like Jenkins/Pipelines and and/other technologies; sound knowledge of the SDLC, and the ability to work with different project management methodologies (Agile, Waterfall, Hybrid).

of improving the daily lives of people, the world, and your career Next\_reimagined Job Title Specialist - Infrastructure DevOps Job Summary DevOps Engineer who is ideally a good programmer and knowledge of wide variety of open source technologies and tools Being comfortable with frequent, incremental code testing and deployment DevOps Engineer will drive the automation of multiple parts of infrastructure and deployment systems, striving to improve and shorten processes to enable engineering and operations teams work smarter and faster Design and develop infrastructure interfaces for complex business applications Contribute ideas for improvements in DevOps practices, delivering innovation through automation Primary Responsibilities The Specialist would bring hands-on technological expertise, passion and innovation to the table He will be responsible for designing and enabling Application support and handling Production farm and various Infrastructure platform for different delivery teams In the capacity of a subject matter experts will be responsible as a system architecture to design and build scalable and efficient Infrastructure Platforms At the same time, specialists will also be responsible for establishing best practices, cultivating thought leadership, and developing common practices solutions on Infrastructure Experience Guidelines 12-15 yearsGood Working Knowledge of Linux Operating SystemExtensive experience in Application Server administration like Tomcat

A DevOps Engineer is ideally a good programmer and has knowledge of wide variety of open source technologies and tools Being comfortable with frequent, incremental code testing and deployment, DevOps Engineer will drive the automation of multiple parts of infrastructure and deployment systems, striving to improve and shorten processes to enable engineering and operations teams work smarter and faster S he would design and develop infrastructure interfaces for complex business applications and contribute ideas for improvements in DevOps practices, delivering innovation through automation 160; 160; Primary Responsibilities 160; Responsible for implementation and support of application web server infrastructure for complex business applications Server configuration management, release management, deployments, automation troubleshooting Set-up and configure Development, Staging, UAT and Production server environment for projects and install configure all dependencies using the industry best practices Manage Code Repositories Manage, Document, Control and Innovate Development and Release procedure Configure automated deployment on multiple environment Hands-on working experience of Amazon Web Services, Azure or traditional hosting experience Knowledge Transfer the implementation to support team and until such time support any production issues Will be responsible for documentation of the implementation Will be expected to deliver as planned Will be expected to work in rotational shifts when required 160;Experience Guidelines 160; Required experience is 5-9 years Strong expertise in DevOps tools Deployment Chef Puppet Ansible

Nebula Nolio SCM TFS, GIT, ClearCase Build tools Ant,Maven, Make, Gradle Artifact repositories Nexes ,JFrog ArtiFactory CI tools Jenkins, TeamCity Experienced in scripting languages Python, Ant,Bash and Shell Extensive experience with TFS Git like distributed version control systems and their use in release management, branching, merging, and integration strategies Extensive understanding of CI best practices and tooling, preferably Jenkins Extensive experience in developing and automating deployment tools Expertise in managing Cloud VMWare resources and good exposure on Dockers Working knowledge of operating systems Unix, Linux, IBM AIX Experience in installation, configuration and managing apache webserver, Tomcat Jboss Good understanding of JVM, troubleshooting and performance tuning through thread dump and log analysis Backup and restoration of application web servers data Basic knowledge of network infrastructure for e g 160; TCP IP, DNS, Firewall, router, load balancer etc Understanding of clusters, Virtual Hosts, JDBC, auto-deployment, session replication, URL redirections, Re-write rules, http-plugins, security certificates, etc Experience in signing and designing highly available infrastructure Advanced scripting skills to automate jobs start, stop, recycle, deploy etc Security Hardening of application web server against known unknown vulnerabilities Hands on experience of AWS AZURE to create a network subnets, launch instances, create load balancer in a Production Environment 160;Education Full time Bachelor 8217;s Master 8217;s degree Science or Engineering preferred 160; Location Gurgaon and

 DevOps Engineer Automation Engineer 8211; Will drive the automation of multiple parts of infrastructure and deployment systems Design and develop infrastructure interfaces for complex business applications Senior DevOps Engineer Senior Automation Engineer - Architecting and designing high-end, cost effective infrastructure on AWS cloud 160;Managing platform 160;operations, 160;Production 160;Support 160;Coverage, 160;Management Reporting, Design 160;Documents, 160;Change Management 160;Process 160; DevOps Architect Specialist- Solution 160; 160;technical 160;architecture 160;and accountable 160;for 160;creating 160;the 160;architecture 160;definition, 160;NFRs, 160;Capacity 160;Planning, 160;Performance 160;tuning, 160;devops 160;pipeline 160; 160;release 160;management 160;strategyAbout SapientRazorfish SapientRazorfish is a new breed of transformation partner designed to help companies reimagine their businesses for the connected world With more than 12,000 people and 70 offices around the globe, our capabilities span growth and business model strategy, new product and service innovation, customer experience, enterprise digital transformation, IT modernization, omni-channel commerce, precision marketing, change management, digital operations, digital innovation, data strategy and advanced analytics 160; 160; Sapient is a registered service mark of Sapient Corporation 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; 160; Education : Other

Job Description

The future is infinitely wide and begging to be reimagined and SapientRazorfish is looking for people who have the courage to make their dent in what has always been. If you have an optimism that says there is no limit to what we can achieve in a time inviting those with vision to shake things up, then we should talk. Our work is fueled by connected thinking, challenging boundaries, and the power of digital redefining a connected world. Our people thrive because of the belief that it's both our privilege and responsibility to usher our clients and the world into what is next. We're seeking those who question the status quo and want to reimagine the way the world works while realizing a smarter way for business to help - all in service of improving the daily lives of people, the world, and your career. Next\_reimagined.  
  
Job Title:   
Senior Associate - Infrastructure DevOps  
  
Job Summary:   
DevOps Engineer who is ideally a good programmer and knowledge of wide variety of open source technologies and tools. Being comfortable with frequent, incremental code testing and deployment. DevOps Engineer will drive the automation of multiple parts of infrastructure and deployment systems, striving to improve and shorten processes to enable engineering and operations teams work smarter and faster. Design and develop infrastructure interfaces for complex business applications. Contribute ideas for improvements in DevOps practices, delivering innovation through automation.  
  
Primary Responsibilities:

 Responsible for implementation and support of application/web server infrastructure for complex business applications

 Server configuration management, release management, deployments, automation & troubleshooting

 Set-up and configure Development, Staging, UAT and Production server environment for projects and install/configure all dependencies using the industry best practices

 Manage Code Repositories

 Manage, Document, Control and Innovate Development and Release procedure

 Configure automated deployment on multiple environment

 Hands-on working experience of Amazon Web Services, Azure or traditional hosting experience

 Knowledge Transfer the implementation to support team and until such time support any production issues

 Will be responsible for documentation of the implementation

 Is expected to deliver as planned

 Expected to work in rotational shifts when required  
Experience Guidelines:

 4-8 years

 Strong expertise in Dev Ops tools:

 Deployment (Chef/Puppet/Ansible /Nebula/Nolio)

 SCM (TFS, GIT, ClearCase)

 Build tools (Ant,Maven, Make, Gradle)

 Artifact repositories (Nexes ,JFrog ArtiFactory)

 CI tools (Jenkins, TeamCity),

 Experienced in scripting languages: Python, Ant, Bash and Shell

 Extensive experience with TFS / Git like distributed version control systems and their use in release management, branching, merging, and integration strategies

 Extensive understanding of CI best practices and tooling, preferably Jenkins

 Extensive experience in developing and automating deployment tools

 Expertise in managing Cloud / VMWare resources and good exposure on Dockers

 Working knowledge of operating systems( Unix, Linux, IBM AIX)

 Experience in installation , configuration and managing apache webserver, Tomcat/Jboss

 Good understanding of JVM, troubleshooting and performance tuning through thread dump and log analysis

 **Backup and restoration of application/web servers data**

 Basic knowledge of network infrastructure for e.g. TCP/IP, DNS, Firewall, router, load balancer etc.

 **Understanding of clusters, Virtual Hosts, JDBC, auto-deployment, session replication, URL redirections, Re-write rules, http-plugins, security certificates, etc.**

 Experience in signing and designing highly available infrastructure

 Advanced scripting skills to automate jobs (start, stop, recycle, deploy etc.)

 Security Hardening of application/web server against known/unknown vulnerabilities

 Hands on experience of AWS/AZURE to create a network subnets, launch instances, create load balancer in a Production EnvironmentEducation:  
  
Full time Bachelor's /Master's degree (Science or Engineering preferred)  
  
Location:  
Gurgaon and Bangalore  
  
Projects this role will support:   
Projects for Clients who are Industry leaders in the fields on Banking, Insurance, Travel and Hospitality, Consumer Packed Good, Retail Sector, Consumer Electronic, Food and Beverages to name a few  
  
Possible career-paths:

 Dev Ops Engineer / Automation Engineer - will drive the automation of multiple parts of infrastructure and deployment systems. Design and develop infrastructure interfaces for complex business applications

 Senior Dev Ops Engineer / Senior Automation Engineer - Architecting and designing high-end, cost effective infrastructure on AWS cloud

 Managing platform operations, Production Support Coverage, Management

 Reporting, Design Documents, Change Management Process

 Dev Ops Architect / Specialist

 Solution & technical architecture and accountable for creating the architecture definition, NFRs, Capacity Planning, Performance tuning, Dev Ops pipeline & release management strategy  
About SapientRazorfish   
SapientRazorfish is a new breed of transformation partner purpose-built to help companies reimagine their business through radical customer-centricity. With more than 12,000 employees and 70 offices around the globe, our capabilities span growth and business model strategy, new product and service innovation, customer experience, enterprise digital transformation, IT modernization, omni-channel commerce, precision marketing, change management, digital operations, digital innovation, data strategy and advanced analytics.   
Sapient is a registered service mark of Sapient Corporation.