Lecture - 6

Mid-term

Bring your NUID for the exam if you forget you will not be allowed

the mid-term has 25% weightage

if you get caught cheating in the exam - 0 marks will be issued - you will be failed on the course

The exam will cover theory portion - no code during exam - not an open book exam

- From lecture, Assignments, the links professor has posted, slide decks

its not going to be anything that he has not taught in the lecture - The questions are designed to test your understanding - he will not try to trick you in the exam - OSI model

he will never ask for commands

but, this is a fair quesiton “what does grep do?”.

pen and paper bring your stuff

Do we have to know how VPC and subnets work?

if he has covered in the lecture then yes, else no

make sure your hand writing is clear

Virtual Machines

- When we have a VM we will have to always patch it (update packages and make changes to application - check for dependency changes) - **immutable machine images**
    - sometimes the deployment does not clear the old dependencies and the application does not start
    - when we do a in-place deployment (update application on the existing server) - what worked in dev, pre-prod might not work in prod (because of old server) - this is why we wil be building a new server when we deploy a new version of application (machine images)
        - This way we get consistency
        - never run manual commands on server
        - if something fails - create a new one from scratch
        - it is common that some instances will have failure might not be due to your application (hardware failure) when you are running the long enough at scale this failures will become frequent
    - Lot of things that are env specific will change (like region, network) this failures can happen because of this
    - improves security (something malicious happens this will be replaces), reliability (unhealthy → healthy)

Understanding Virualization

- VM boxes on your local machine (add more notes)
- Bare metal instance
    - our laptop
    - limitation of Bare metal - ~~Scalability~~ (he said no) actually its quite the opposite because we have control over hardware.
    - we will always prioritize reliability and stability and security over cool new features
    - that is why we will be using old version of OS all the time in enterprise

- when bare metal server support one OS - this is an disadvantage
- you will be spending a lot of money for servers - when you have bare metal instances
- this leads to increase in cost just to start deploying services on it
- Some servers will be ideal (not used) - when you have bare metal instances just staying on. So, we are essentialy spending on services that you are not using -
- How do we save money?
    - Some will say you can deploy multiple apps on one server
    - Some server vendors will just say is something is not working that will say we need you to run a specific version of software to support and troubleshoot issues in a server (but, applications might need to run a different version)
    - Servers are just expensive (like 50k for a server)
    - **The Answer -** Just virtualize - we create a virtual rep. of what you are interacting with (its a software layer over a big hardware)
    - Advantages (**Virtualization**)
        - Bin Bagging - get the most value of a given resource
        - So, this van be achieved by virtualizing the server. each VM wil have its own OS
        - Software defined data center - is the answer
        - Minimize or eliminate downtime -
            - The mainframe machine will redudentancy Hardware failures will be replaced with NO DOWNTIME
            - IRS has mainframe servers
            - Airline tickets - late confirmation message - reason - the processing of the booking happens by a system built by IBM on 70s on a mainframe server (sapier? - the name of the system)
            - When you buy a mainframe - the key reason is that

    - Virtualization has been the corner stone of mainframe (its been around for a long time)- cloud was born form that
    - Increase IT productivity - The server comes with a management engine , without a server
        - Provisioning a VM is much faster (because that are built from images) - 1 min time
        - This allows us to scale up and down quickly - optimize cost
        - Enable business continuity and disaster recovery

How does Virtu.. work?

- Hypervisor - a OS which is light weight with a purpose to manage Virtual machine
    - Two types of Hypervisor
        - Installed directly in the server
        - Installed on top of an existing OS
    - Popular choice - zen hypervisor
        - needs a linux OS - runs on top the OS
- if we have server we need to decide whch hypervisor to use, but in cloud we dont care
- Key properties of hypervisor: (refer slide 9)
    - Isolation:
        - Software will not always help us (it will always have bugs and holes)
        - Best we can do is we can add more layer of protection its really hard to exploit
        - the only one that can exploit (or try) is government (because they have lot of resources)
        - intel cpu security issue - meltdown - spectre

            - if else (processors will have something called branch prediction) Will process and keep it (wont persist it but will keep it with you
            - you will be running multiple application, an another process and look at the data, based on cpu cycle we can figureout what type of instructions are carried out by the cpu
            - patching such this are really hard - inter started adding hyper threading - 4 core cpu - each cpu running 4 threads - 16 virtual threads. and the thread on the same cpu can look at the data
            - This become a problem from cloud providers (this is not safe)
        - one suggestions was to disable hyper threading - Intel is mostly affected - AMD is little affected - ARM chips are not affected as f now
        - This anyhow is a software problem
    - Encapsulation:
    - Harware Independence:
        - VI software can simulate a different chip
- Host OS → Hypervisor → Guest OS → Apps
- Hypervisor → Guest OS → Apps

Amazon EC2 and Google Compute:

- there will be no specific question related to the cloud provider
- These are compute resources
- Time zones: a island skips a day because it decided to change its timezone
- now EC2 and Google compute - by default sticks to UTC

Benefits of VM and cloud (slide 17):

VM machine Types (Slide 19, 20,):

- we can buy any kind of machine based on the use case (we. have choices while selecting a EC2 instance)
- if you dont know what to select select General purpose - find out bottle necks - switch to an right one based on you analysis
- When running CPU intensive workloads - we need to have **compute optimized**
- Memory optimized (i.e redis one thread but large cache more RAM)
- Accelated GPUs- mining
- Storage Optimized - DB fast read and write

- Some times we need dedicated hardware because of complience/ security - Dedicated instance (slide 22):
  - like hospital servers
- Dedicated Hosts:
  - when we need licensing stuff - we can run this on the dedicated on the dedicated hosts
  - and we can see how the dedicated hosts and the underlying hardware
- vertical scaling involved downtime, but the DBs are designed with vertical scaling in mind
- Showed GCloud compute machine types (slide 24)
- Size of machine determine the throughput and bandwidth

Pricing (slide 27, 28, 29, 20) :

- various pricing models for EC2 instance (slide 27, 28,)
- Why are some companies moving back to onprem from cloud?

    - cloud is expensive but, when you are oncall then you need to have your laptop around you. but with on-prem oncall you need to be around in the DS all the time. with plumber on call and electrocution on problem
    - on-demand pricing - we only pay for what you use
    - cloud gives us flexibility - no comm itment - try stuff out - need it commit - else say no
    - Enterprise never pay the prices that we see, they will have a rep who will give you a enterprise discount
- Spot Instances (slide 30)
- NO COMPANY PAYS LIST PRICE (ONLY DEVELOPERS DO)
- GCE discounts (slide 33) - sustained use discount
- General note: more you are committed the more discounted prices

Cloud VM feature

- secure login using key pairs
- Digital ocean lets you login with password - (worst practice)
- Temporary and permanent storage volumes
- Have security groups
- Static IPv4 addresses (amazon is thinking about charging money for public IPV4)
    - Amazon is trying to push people towards IV6 and using private IPs
- tags
- Subnets (virtual networks)

Machine Images

- what is an AMI?

- on google its called machine image (custom image)
- when we build an image we always start with a **source image** thats the base OS
- on top the Source image we will build our desired image based of that
- we will install stuff on top of it and then take a image of that
- why are we doing this?
    - it is very tedious to do the setup work on every instance that we launch - this can error prone as well
    - So, having an image means we can launch the instance with the existing setup already present
    - i.e if we are using Java 18.x and we install Java 18.y (change in minor version) will result in an application failure
    - Building machine images with the exact version means that it is going to be very consistent in the dependency set up
    - First base image → then security team will make sure OS is patched and everything is safe → another team will install the java versions and all → dev team will then put their code here
    - This is a cycle every week a new base image will be cut and the whole process is repeated again
    - the end machine image that we create with the application is called the golden image
        - why?
          its all tested and proven with high level of confidence and can deploy in prof
    - How do we build machine images?
        - Different vendors will do things in different way
        - So, we dont want to run through all the three platform
        - The source image can come from the cloud platform itself
        - How do we achieve it?

        - There is tool called packer by HC
        - Packer user HCL
        - There is two places you need packer local machine and github runnder
        - **Do not try anything to do it yourself → explore github marketplace you will find one for packer**
    - Custom Machine Images use case:
        - Golden images
        - Environment parity
        - Auto-scaling Acceleration
            - The Auto scaler will bring up the VM but will not be ready on time with out Machine images
            - We can provide a shell script to Auto-scaler to configure the application before it starts
            - So, after switching to machine images this only takes 30Sec. So, this is really good in terms of time/cost. when we do this in larger scale and for enough time the value will be clear
    - Build an image - The template
    - Hands on:refer lecture page for resources
        - install packer
        - HCl is what is recommended to used with packer. the JSON wil be depreciated soon
        - Packer has a formatter command
        - terraform has a validate command
    - No reason to stick to specific version of provider
      always use > symbol
- Builder > Google Cloud Platform
    - says how the image will look like (pulls the source image)

    - the first thing to do when we pull the image sudo apt update (ubuntu) dnf update (centOS)
    - Then copy application code, install all the dependencies to run the project from scratch
    - we will not config env. variable because we are not yet sure which

  https://developer.hashicorp.com/packer/integrations/hashicorp/googlecompute

- on google cloud the tags can be used as a part of security
    - The tag will be part of firewall
    - attach the tag to VPC
    - and in kubernites its all labels
- we can use your default VPC to build Packer image (machine images) this is the only case we will allow this
    - in the next assignment create VPC and EC2 launches depends on VPC creation
- Provisioners:
    - like file copy provisioners
    - There are two ways to run shell script (inline, script)
    - a packer template can have more than one provisioners
    - they are executed on the order they are defined in the template
    - you will be having more than 1 shell script (copy files, set up packages etc..)
    - like creating a new user and creating a group, copy systemd file on to the server
    - DO NOT USER INLINE (WRITE SCRIPT)
- Post-processor:
    - we can tell where to save the image use the tmp directory for files that are going to be cleaned up (because when we build the machine image, the tmp directory will be ignored)

SHOWED AN DEMO WITH A PACKER FILE

points to note

- Don’t lock the version use >1.0 (something along this line)
- validate the Packer template
- why terraform plan will work and terraform apply may fail
- source_image vs source_image_family (will take the latest version of CentOs steam 8 -given that we said its CentOS8)
    - but, source_image will take the exact version
- check packer init -force <packer_file>

Launching the VM:

- Virtual machines are always zonal
- When we build out instance - we would have to pick the things that Amazon gives us. but on gcloud its very flexible
- we would define the custom image on the boot disk
- we will create firewall riles with we do with terraform
- we can select VPC and subnet
- Equivalent code on google there is a terraform you can get the terraform code there
    - this sometimes over complicate things so, be cautious while using it
    - you have to actually know the specs to figure out what needs to be removed
- Do I have to install nginx?
    - No, he just did it for demo
- fact: To allow http we have to add the tag http-server
- in google cloud there is a refresh option dont reload the whole window

- How to ssh into the instance?
    - you can use the os logiv command that you can copy from the instance page

on Amazon / Azure / google if we stop an instance we dont pay for compute but, only pay for storage. but in digital ocean we will pay for both even if we stop

when we build and deploy an instance the application should start and work NO SSHING in

for demo - Build process with happen on the Github runner

no firewall and not able to ssh and check and validate. you will loose lot of marks

the specs (storage) that we are giving to packer is to specify what kind of instance to launch for building an image

Writing Systemd files:

There is an example that is provided in the lecture refer the lecture 6 in webpage

Appendix:

read through all of the resources in digital ocean link can be on midterm

install postgres and realted stuff as part of the machine image

execstartpre - take a look

just get the service part right

local dev work do it in dev-project

Step 5:

- Tom cat is just an example not an requirement - you don’t unless you are using it

- have the Db locally
- packer must be in the webapp repo
- we will two packer actions
- packer fmt, packer validate
- Dont build an image when raising a PR only build it on merge
- You have to setup a service account to give the github runner the persmissions
- if we are using node create a zip file (dont do git clone on the build image)
- update the system
- Create a user and group without login shell
- Make sure you dont put all the provisioners in one script file
- once you have to user setup install dependences and application
- Provisioners that update permissions
- if any of the steps fail the build should fail
- DON NOT USE A SINGLE PROVISIONER TO ALL THE THINGS, NO INLINE SCRITPS
- four script → OS, group creation ,systemd, permission and owner ship
- very important dont allsow ssh traffic from the internet
- The instances values cannot be hardcoded (use tfvars only)
- no new repo
- in terrafrom we will use depends on for EC2 should only start when our VPC is ready
- Firewall, subnet and VPC all have to be created
- Make sure the repo remains clean after doing packer init
- https://github.com/google-github-actions/setup-gcloud
-